The Cameremin (Camera Theremin)

MUS207 Final

Andrew Kehler
Tyler Kettner

1 Introduction

The theremin is an instrument in its own league. Developed in 1928 by Russian inventor Leon Theremin[1], the theremin is one of the oldest electronic instruments. It is a ‘touchless’ instrument, controlled by two metal antennas that act as position sensors. These metal antennas sense the position of the player’s hands, with one hand controlling the frequency of an oscillator, and the other hand controlling the amplitude. While the ability to manipulate sound without touching the instrument already gives the theremin a unique and futuristic feel to it, we decided to take the theremin to the next level - by creating an instrument-less theremin. Enter, the ‘Cameremin.’

2 Concept

Our aptly-named ‘Cameremin’ was designed with ease-of-use in mind. Instead of a physical instrument, the Cameremin simply consists of a single camera, connected to a computer running a program. The camera tracks hand position and finger movements to control pitch and gain, as well as effects like vibrato. The lack of any physical component required to be held/played by the user allows for extremely flexible application. For example, a performer would be able to walk onto a stage empty-handed and immediately start playing the Cameremin, provided prior set up. In its current state, the Cameremin might not have much immediate use other than a quick tech demo, but with enough development it could easily be transformed into a useful tool for performances, instrument-less sound manipulation, or even gesture based audio synthesis (such as ‘text-to-speech’ for ASL).

3 Context

The original concept for the Cameremin was derived from a video posted by Matias Vilaplana [2] in which a performer dances in an open space where their body was tracked and various body positioning was used to control oscillators and sound files to create a soundscape that moved with the dancer. The hand controller aspect was heavily influenced by Imogen Heap’s Mi.Mu gloves[3] which the musician uses in various songs. The gloves allow Heap to control several aspects of a scape like pitch and filter. The Cameremin borrows the pitch element directly. The combination of these elements into a theremin design came from an article on PIER Journals describing a SRR based touch-pad for a microwave theremin.[4] It proposes an alternative interface using Split-Ring Resonators to detect finger proximity to change resonance frequency while remaining unaffected by other objects which, on a surface-level definition case, the Cameremin follows without the use of any physical components.

4 Implementation

The Cameremin involves two components: a Python program that tracks hand and finger positioning, and a Max/MSP patch where the sound is synthesized.

4.1 Python Code

Written in Python 3.11, the script utilizes OpenCV to capture frames from video input, essentially taking a screenshot multiple times per second. MediaPipe then analyzes these photos with machine learning[5] to detect any hands/fingers in view. Once it ‘sees’ a hand, it applies a skeletal overlay using 21 key points (mostly the joints in your hand), and uses it to determine the position of those points. In the case of the Cameremin, we only use two of those points - the tips of the index fingers and thumbs. These points are then mapped to pixel coordinates, where Pythagoras’ theorem is used to calculate the distance between them. The position of these points and the distance between them is then converted into data that is readable by Max, and is sent to the maxpatch file through a UDP server.

4.2 Max/MSP Patch

In Max/MSP, the data from the Python program is received through a UDPreceive object, where the data is then sent to different sub-patches. The distance between thumb and index on the right hand controls the gain. As the distance increases, gain levels decrease. Similarly, as the distance between finger tips decreases, the gain increases. The left index/thumb distance is scaled and used to control modulation depth within the vibrato sub-patch (pvibrato). The oscillator sub-patch, which contains pvibrato, combines a triangle wave with a sine wave with a frequency controlled by the position of the thumb. This signal is then modulated by the output from pvibrato before being sent to the audio output.

5 Strengths

One of the biggest strengths of the Cameremin is that there is no physical instrument needed to play. While it still requires physical components such as a computer and a camera, the lack of any need to manipulate an object by the user provides an ease of use and accessibility unlike most other instruments. The tracking is also very precise, allowing for a large amount of room for expressivity when playing.

Additionally, the flexibility of our program is quite vast. While the current version is basic, it provides a glimpse into future possibilities and use cases.

6 Weaknesses

While the Cameremin can provide a cool touch-less music experience, there are some barriers that would need to be overcome to ensure optimal usage. The speed at which the program updates data relies on the frame rate of the camera being used - this means that with lower quality cameras, a change in pitch can become choppy, instead of a smooth glide from one note to the next.

Along with changes in pitch, the accuracy of notes is limited by the camera's resolution. Since finger positions are mapped to pixels in the frame, lower resolutions means less pixels, meaning less available pitches to play.

6.1 Compatibility

During development, we encountered some compatibility issues that slowed things down. We quickly found out that the Python program would not run on some of the developer's computers. Whether this was an issue with the computer OS (Windows/Mac), Python version, or something completely different, we weren't able to figure it out. This meant that when the Max/MSP side of things needed to be tested with live data, the patch would need to be sent over to a computer that could run the Python program for testing and debugging.

7 Future Considerations

The Cameremin provides a solid demonstration of what can be achieved when combining hand tracking with sound synthesis. However, it is still simple in design and functionality. Other than showcasing how it works, there isn't yet much practical use for it. In order to make the Cameremin feel more complete, there are a few aspects that would need to be refined/built upon.

7.1 Note transitions

As mentioned before, the program controlling the synth can only update the frequency as fast as the frame rate of the camera being used. This can create a stutter when quickly changing pitch (especially if the pitches are more than a fifth apart), and cause unwanted frequencies to be played. While this obstacle can be overcome by utilizing a camera with a high frame rate, requiring a high-end camera decreases the accessibility of the Cameremin.

With that being said, this issue could also potentially be resolved by figuring out a way to smoothly transition between pitches on the program's side of things, thus removing the dependence of note transitions on the frame rate of the camera.

7.2 Gesture Clarity

One thing that becomes clear after using the Cameremin for a while is how easily it reacts to every little movement. Even when a user isn't trying to change anything, small shifts in the hand—like moving slightly while holding a note—can mess with the sound. This makes it hard to stay on a single pitch or keep a steady vibrato going, especially during slower or more controlled parts.

To fix this, we could implement some kind of way to ignore small movements unless they cross a certain threshold that would help filter out the stuff that's not intentional. Another idea would be adding a "lock-in" gesture that tells the program to hold a certain value until something else happens. Either of these changes would help the Cameremin feel more predictable and easier to use for people trying to play it.

References:

[1] *The London Mercury*.1928 Vol.17.Iss.99 https://archive.org/details/london_mercury_1928-01_17_99/mode/2up

[2] Matias Viliplana 2019. *Mocap Max/Msp Dance Interaction*. https://www.youtube.com/watch?v=Q8ODY2IPwqc

[3] Dezeen 2014. *Imogen Heap's Mi.Mu gloves will "change the way we make music"* https://www.youtube.com/watch?v=ci-yB6EgVW4

[4] Vladyslav Tkach, Mykola Khobzei, Serhii Haliuk, Ihor Safronov, Andrii Samila, Vjaceslavs Bobrovs, and Dmytro Vovchuk, *Microwave Theremin Piano: SRR-Based Touchpad*, Progress In Electromagnetics Research Letters, Vol. 123, 89-94, 2025doi:10.2528/PIERL24080404. https://www.jpier.org/PIERL/pier.php?paper=24080404

[5] Revision 574dad1. MediaPipe Hand Solutions:Docs https://mediapipe.readthedocs.io/en/latest/solutions/hands.html