IZMIR UNIVERSITY OF ECONOMICS
FACULTY OF ENGINEERING
SOFTWARE ENGINEERING

FENG 497 PROJECT PROPOSAL

Theremin.AI: CNN Based Instrument with Monocular RGB Camera

Authors:

Bedirhan Ziran ELBAN, Meriç LOMLU, Merve Nur TELLİ

Supervisor:

Alper DEMİR

OUTLINE

1.Abstract

These days, solutions that are based on Computer Vision and Neural Network made life easier, also they are opening new doors to Human-Machine Interfaces. With Neural Networks, low-cost consumer cameras are not a problem anymore CNN to get input as a command and the need for computational power is less than before. Those improvements have shown that recreating a unique instrument is viable. In the music industry, there are fewer works that are using Computer Vision and Neural Network as an input device. With the power of Neural Network, a cheap, monocular RGB camera is a solution to our work. Theremin.AI aims to show a way to input in a different way, without touching any hardware. Also, our main approach is showing an input way that opens a dimension to artists and developers.

Keywords-- hand detection, hand depth estimation, distance estimation, computer vision, synthetic oscillation, convolutional neural network, human-machine interface

2.Introduction

Theremin (or Thermenvox, or Aetherphone) is an analog instrument that is classified as Hornbostel-Sachs 531.1 (Electrophone), invented by a Russian scientist named Léon Theremin. It is also creating a sound but the invention is different from other instruments. Without touching or hitting any keys, strings, or percussive elements, the player must use a different interface to create a sound.

In the system, antennas are designed to operate the instrument as an interface that creates electric fields. The instrument has two Antennas that are differently **positioned**. One is horizontally and the other is vertically positioned on the system. The horizontal antenna controls the Amplitude or Volume and the vertical antenna controls the Pitch or Frequency. The sound of the instrument is based on electronic oscillators. To create an oscillator sound, it uses a specified resonant circuit that specializes to sense objects and it's resonance value. Objects are used as capacitors to create an analog signal. Also, the output sound is most likely a fretless instrument.

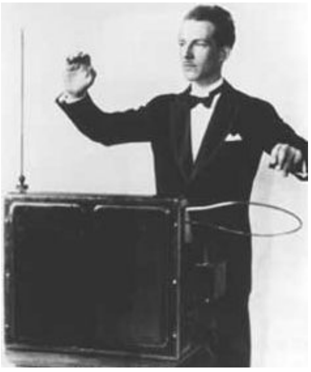


Figure 1: Léon Theremin, plays with his Theremin [1].

After the invention of Theremin, there is an inherited idea of "Gesture-Controlled" sound synthesis and created instruments such as TheremUS [2], Laser Koto [3], and Steim's BigEye [4]. Our project, Theremin.AI not going to be in that hyper instrument list, we planned to recreate the instrument that can create synthetic oscillation sounds by using a single, moderate RGB camera, with specialized neural network algorithms to catch hands and compute them in three dimensional way then creates a sound in real-time with that data within a playable latency (below 15 ms of input-to-sound reaction speed) also can be chained with the effect stomp-boxes. We inherited the idea of "Gesture-Controlled" sound synthesis and wanted to show a way to use this idea on Digital Audio Workstations, Artists, and Sound Designers. For example, with the ChucK language [5], there is a new way to create sound from different peripherals to make music.

2.1. Problem Statement

Today's technological development speed shows us that controlling systems with different ways are sometimes more useful and make room for the creativity of artists. In time, there were too many attempts to control software or generation of sound to make music like tons of MIDI workstations, analog Euro racks, samplers, randomizers, gyroscopic controllers, etc. But there is not a Theremin like “Gesture-Controlled” device in the industry. That made us step into that field.

Also, in the industry, there are machine learning and deep learning solutions made with Digital Signal Processing, like Kemper Profiling [7], but there is no “Gesture-Controlled” input using those solutions.

2.2. Why is The Project Worth Doing

In daily life, there are so many uses of Neural Network solutions, like weather forecasting, identifying cancers in early stages in medicine, currency exchange rates in finance, etc. Also, there are many projects in the field of Hand Detection, Hand Gesture Detection, and Sign Language. Most of them are specialized to control short commands, real-time tracking, motion capture. Those solutions are made to make life easier.

Capturing and getting inputs from Gestures can be used in projects that can eliminate many limitations on the physical interfaces. Right now, many of us think it's unhygienic to use touchable devices, due to the pandemic. On the other hand, one important characteristic of touchless interfaces will be gesture functionality. Limited availability of studies and projects are created for the musical industry. Theremin.AI opens a new dimension to instrument players and developers with the usefulness of Computer Vision and Neural Network.

3. Objectives of the Work

Theremin.AI is a system that will have three parts. The first part is about utilizing a single, monocular RGB camera to get vision. We want to position the camera below input hands. After getting a vision, in the second part, we need to detect and calculate the positions of the hands in real-time. This part is going to be our main aim. After detection of the hand, we have to send hand visions to two different Neural Networks (NN) to calculate the Height of the AmplitudeHand (Left Hand) and Position of the PitchHand for the horizontal vector. Calculating the Height of the AmplitudeHand is the “Challenge” of our project because we have to calculate an accurate depth with a non-specialized, simple RGB camera to get AmplitudeParameter. For this part, we have to create a fast and accurate NN to calculate the depth of AmplitudeHand. Then we will limit the AmplitudeParameter then send it to the oscillation part. Also, the other simultaneous part will calculate the Distance of the PitchHand. The distance is limited by virtual boundaries. This time the specialized NN part will create and send PitchParameter concerning virtual boundaries. That PitchParameter also will be sent to the oscillation part of the system.

Finally, the oscillation part is the output part. In this part, we will use AmplitudeParameter and PitchParameter and create a simple oscillation signal as output. Parameters will be processed using two different algorithms that scale with the two-octave frequencies and a listenable amplitude.

As written above, we aim to achieve a new input solution using Computer Vision and Neural Network Techniques. Also, we wanted to prepare our work to publish in a reputable Academic Journal and after the pandemic, we wanted to show our works to the IEEE and digitIZmir conferences. In the future, we may port the system to the modern Digital Audio Workstations as a plug-in or a plug-and-play MIDI controller hardware to use for everyone.

4. Scope of The Work

In the project, using Computer Vision to get input from the user. From a single RGB camera, the system gets input from two hands. After that, to recognize hands, the usage of neural networks will be our first work to start. Convolutional Neural Networks (CNN) are used heavily to recognize hands. There are so many CNN works made in the past by groups and researchers are doing "Arms Race" in this field. After getting hands as input, we need to create a simultaneous system that can define the hand positions in real-time with Neural Networks. This will be our main aim to compute an accurate calculation. We will work both of the hands in a single real-time vision. Then we will create a basic synthetic oscillation signal like the original Theremin. These days, most of the modern Theremin's created by different companies, like the Moog Theremin series [7], are using multiple oscillation signals to create compound and different sounds. We wanted to show an accurate output, no point to design cool synthesizer sounds in the scope of this work.

5. Literature Survey

We searched for an article written by the Audio Engineering Society about making a Theremin [1] in a different way, made by project-specific hardware to achieve a hyper-instrument version of the Theremin prototype using UltraSound technology.

There are too many projects about detecting hands and depth estimation, mostly they have worked on Stereoscopic or RGB-D cameras or not real-time. We surveyed only the way they used, not the hardware. One is HandyDepth: Example-based Stereoscopic Hand Depth Estimation using Eigen Leaf Node Features [8]. This article is about calculating the hand pose estimation from a cheap, stereo video recording peripheral. In this work, they tested Random Forest Tree, Distance Transform, and Eroded techniques and they choose Random Forest for accuracy. Then they used the Eigen Leaf Node technique to get an accurate hand depth estimation. This is a machine learning project. Another article is Hand PointNet: 3D Hand Pose Estimation using Point Sets [9]. This time, researchers used 3D hand images to create an estimated clustered hand simulation in three dimensions. Also, they created more accurate fingertip examples in the project. With Hand PointNet, users can create three-dimension clustered hand simulation with two dimension hand images.

Then we find some interesting and useful articles for our Theremin.AI. The first one is Learning to Estimate 3D Hand Pose from Single RGB Images [10]. This research is about Sign Language. To achieve this, they worked on two-dimension RGB images, also they got a two-thirds accuracy score to create words from hand pose. They contributed a large dataset with key points for the field.

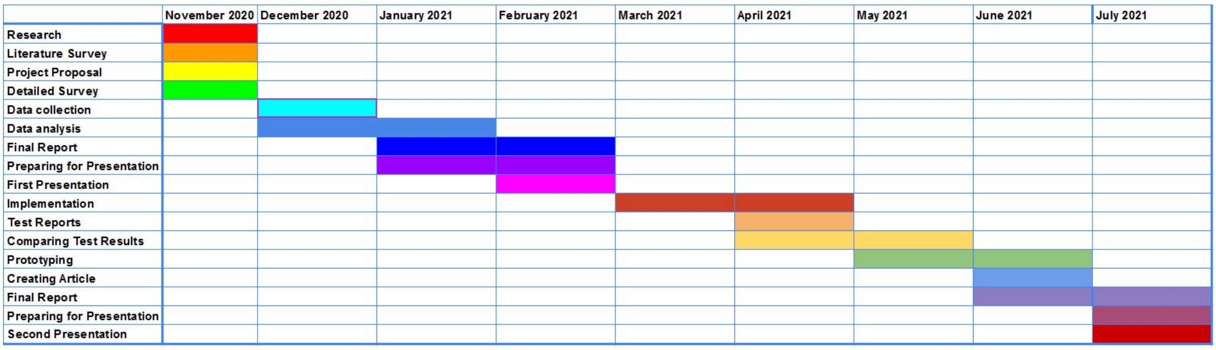
Then GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB [11] article we worked on. This article is interesting to us because they wanted better accuracy scores than Zimmermann and Brox's Estimated 3D Hand Pose from Single RGB Images to work. In this work, their start point is the failure of monocular RGB cameras. They used kinematic model insertion to estimate the skeleton of the hand. They used multiple techniques and architecture to make calculations. 3D Hand Shape and Pose Estimation from a Single RGB Image [12] research are interesting to us. They used a monocular RGB camera to calculate hand pose and shape then they created a three-dimension artificial hand with Graph Convolutional Neural Network. Also, this research achieved accurate scores like works that are using RGB-D cameras. They focused on Chebyshev Spectral Graph CNN to create a polygonal hand simulation.

Then we surveyed Estimating 3D Hand Pose from a Cluttered Image [13]. In this work, researchers used two different cluttered images of tolerant techniques. They used the Chamfer Distance Model to get a pattern of hand and they matched with the calculations. This work made to visualize palm, fingers, and joints with different colors also achieved a fast and accurate score.

Finally, there is a CNN Based Posture-Free Hand Detection [14] article. They used Convolutional Neural Network-based work without posture and sensitive data concern. Also, the interesting part is, they compared two different libraries and performance. Their benchmarking libraries are CUDA and TensorFlow. Also, they have great results with a 4th Generation Mobile Intel i7 CPU without using any enthusiastic Graphics Processing Unit like Nvidia Titan X. The literature survey we did here is only a part of our study. We filtered what we need in the future.

6. Project Plan and Schedule

Due to the Covid-19 situation, we created a flexible, nine-month-long Gantt Chart.



	November 2020	December 2020	January 2021	February 2021	March 2021	April 2021	May 2021	June 2021	July 2021
Research	🟥								
Literature Survey	🟧								
Project Proposal	🟨								
Detailed Survey	🟩								
Data collection		🟦							
Data analysis		🟦	🟦						
Final Report			🟦	🟦					
Preparing for Presentation			🟪	🟪					
First Presentation				🟪					
Implementation					🟥	🟥			
Test Reports						🟧			
Comparing Test Results						🟨	🟨		
Prototyping							🟩	🟩	
Creating Article								🟦	
Final Report								🟪	🟪
Preparing for Presentation									🟪
Second Presentation									🟥

7. Risk Analysis

There are few risks possible that can cause us to fail for this project. The greatest risk above them is the camera position and choosing a specific camera. For example, if we use a FOV camera like a fish-eye lens or locate the camera in the wrong position, it can cause pixel stretching for the image and we might not get accurate results. There is a risk of getting fast and hand recognition and getting accurate hand depth values. We need boundaries for accurate hand depth values. Our equipment may be insufficient for this project. Finally, there is always the risk of going beyond the scope of the project. In other words, we can deviate from our goal. For the last two risks if they are happening we make a mistake somewhere. Our plan B is listed as follows:

Camera Position and Specific Camera: Through trial and error we can find the right position and the right camera. For example, position the camera below AmplitudeHand or PitchHand, etc.

Hand Recognition and Accurate Hand Depth and Distance: We can train the NN more or we can change our dataset.

Going Beyond the Scope: We can go back to our original purpose. A quick recap or seek more works in the field.

8. REFERENCES

[1] Léon Theremin. (2016) Engineering and Technology History Wiki [Online]. Available: https://ethw.org/Leon_Theremin

[2] A. F. M. Gomes, D. Albuquerque, G. Campos, J. Vieira. (2009) ThertemUs: the Ultrasonic Theremin

[3] M. Masaoka. (2006) Lazer Koto [Online]. Available: https://futuremusic.com/2006/11/22/miya-masaoka-invents-the-laser-koto/

[4] Steim. Big Eye. [Online] Available: http://www.steim.org/steim/bigeye.html

[5] G. Wang. (2008) The ChucK Audio Programming Language: A Strongly-timed and On-the-fly Environ/mentality.

[6] Kemper [Online]. Available: https://www.kemper-amps.com/profiler/overview#a-profiling

[7] Moog Theremin Series [Online]. Available: https://www.moogmusic.com/synthesizers?type=52

[8] R. R. Basaru, G. G. Slabaugh, C. Child, A. Alonso. (2016) HandyDepth: Example-based Stereoscopic Hand Depth Estimation using Eigen Leaf Node Features

[9] L. Ge, Y. Cai, J. Weng, J. Yuan. (2018) Hand PointNet: 3D Hand Pose Estimation using Point Sets

[10] C. Zimmermann, T. Brox (2017) Learning to Estimate 3D Hand Pose from Single RGB Images

[11] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, D. Casas, C. Theobalt. (2018) GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB

[12] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, J. Yuan. (2019) 3D Hand Shape and Pose Estimation from a Single RGB Image

[13] V. Athitsos, S. Sclaroff. (2003) Estimating 3D Hand Pose from a Cluttered Image

[14] R. Adiguna, Y. E. Soelistio. (2018) CNN Based Posture-Free Hand Detection