MuSpaghetti Studio

06.04.2020

Omar Mohamed Ibrahim

Omar Magdy Emam

Omar Nagah Sayed

Description

MuSpaghetti Studio is an application that offers the user a virtual experience of a **musical studio** through image processing and computer vision techniques. The app uses the front-facing camera of the hosting device to start a real time video capturing where the user is surrounded with a number of virtual different musical instruments, placed at certain positions on the video screen. The user can interface virtually with these instruments by hitting them with a tracked object (stick for example) to produce the assigned sound to this musical tool.

The user can use the **GUI** to import the desired sound and image then assign them to one of the available positions on screen. There are two ways of musical instrument interfacing user can decide between, either **ON/OFF** (i.e. once the tool is hit it goes on until it is hit back again to go off) or **Just On Hit** (i.e. the tool produces the assigned sound only on being hit). The user also is able import a full sound track to run in the background with his own remix**.**

Once starting the video capturing, the app control is switched to the **Hand Gestures**, the user can test the instruments sounds until he/she gives the hand sign of record starting. The GUI keeps recording the played music until the user gives the stop recording sign, then the camera is turned off and switched back to the GUI to playback the recorded sound track and decide whether to save or delete it as he/she wishes.

Project Implementation Plan

- We will use some **image processing** techniques to **pre-process** the real-time video frames and **remove noise** to be able to detect the required object.
- We will use **motion tracking** algorithms in **OpenCV** library to keep detecting the moving object (ex: hand/stick/etc ...) through real time.
- We will implement a **"Hand Gesture Recognition"** module using **contour** detection, to be able to give start/stop instructions to the sound recorder.
- We will build a **multithreading** infra-structure to run all the project features in parallel at the same time.
- We will create a desktop **GUI** to act as the workstation, compiling all required controls to interface all the features.

Team Members Roles

Member responsible for implementation of:

1. Motion tracking algorithms and musical band video interface**.**
2. Hand Gesture Recognition module for recording control.
3. Multithreading infra-structure and GUI packaging.

Image Samples

Virtual Music Studio

Hand Gesture Recognition

References

[1] Object tracking algorithms in OpenCV
https://www.pyimagesearch.com/2018/07/30/opencv-object-tracking/

[2] Hand gesture recognition techniques
https://www.hindawi.com/journals/tswj/2014/267872/