

Примена на компјутерски генериран звук

1. Вовед

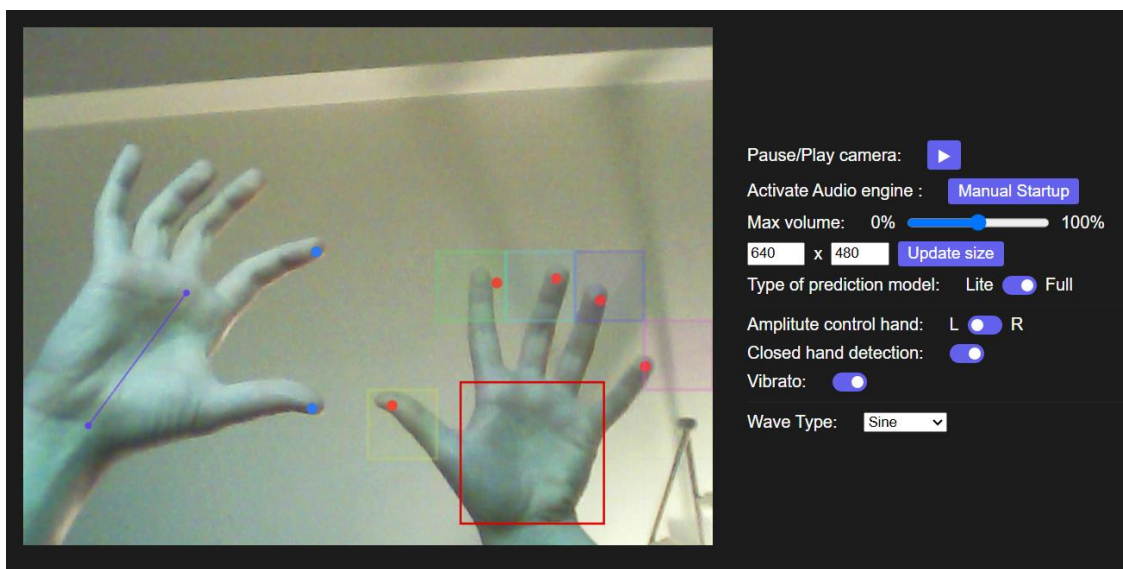
Целта на овој проект е да се создаде интерактивна апликација што применува компјутерски генериран звук. Оваа веб апликација функционира како дигитален музички инструмент и ја користи камерата да ги следи точките на раце. Врз основа на тие точки и нивна меѓусебна релација генерира одредени ноти.

Проектот е изработен со основни веб технологии (HTML, CSS и JavaScript), и со библиотеките ml5.js и p5.js. Оваа апликација е хостирана преку GitHub Pages и може да се пристапи на <https://veljkoadzic.github.io/audio-synthesis-project>.

Библиотеката ml5.js е надградена на TensorFlow.js и нуди едноставна интеграција со p5.js. Содржи повеќе алгоритми и модели за машинско учење, а во овој проект се користи моделот за детекција на раце HandPose.

Библиотеката p5.js овозможува лесен пристап до разните API на прелистувачот за едноставна работна околина за графика, звук и интеракции со корисник на веб. Во оваа апликација се користи за исцртување на платно, менаџирање со видео, и менаџирање со AudioContext API преку осцилатори елементи.

Во проектот дополнително се користи и обичен JavaScript код за: процесирање на претпоставки од HandPose моделот, управување со менито за поставки, и класи за интеракција помеѓу HandPose моделот и осцилаторите.



2. Како работи апликацијата?

Главната логика е имплементирана во sketch.js датотеката. Ова скрипта ги содржи функциите setup(), која се повикува еднаш кога се стартува апликацијата, и draw() која се повикува секое исцртување (до 30 пати во секунда).

Во setup() се поставува големината на платното, глобални осцилатори и се поставува формата за поставки. Наредно се креира објект за видео, се вчитува модел за детекција за раце, и се креира нова матрица од копчиња.

Во draw() прво на најдолниот слој се исцртува една рамка од видео објектот, и потоа ако има промена на типот на осцилатор се врши промена во матрицата. Следно се извршува логика за секоја од рацете.

Кога има детекција на главната рака се исцртуваат користените точки и се пресметува амплитуда за осцилаторите на матрицата. Ако е вклучено вибрато, според ротација на раката се пресметува нивото на кое вибрато имаат осцилаторите (за колку да отстапуваат од нивната фреквенција).

Кога има детекција на секундарна рака матрицата ги ќе притисне сите копчиња во кои врвовите на прстите се наоѓаат. Потоа се исцртуваат точките на прсти. Ако е поставено детекција на дланка се исцртува и квадрат околу дланката. Ова служи за визуелно да покаже каде прстите можат да бидат без да ги притискаат копчињата.

3. Ограничувања

Апликацијата нуди солидни брзина и прецизност на детекција, но бидејќи се извршува во прелистувач перформансите се ограничени од самиот прелистувач и можностите на JavaScript. Како резултат, апликацијата се соочува со одреден степен на доцнење, непрецизност и прекини во звукот.

Во следниот дел ќе се разгледаат алтернативни имплементации за ваква апликација, и предности и недостатоци за имплементацијата.

4. Алтернативни начини на имплементација

4.1. Имплементација со TensorFlow.js и WebGL

Еден начин да се подобрат перформансите на апликацијата е да не се користат ml5.js и p5.js за мултимедија и за машинско учење. Овие библиотеки се големи што влијае на перформанси. Може проектот да се имплементира со обичен JavaScript код кој ги користи API-та на прелистувачот, и TensorFlow.js за детекција на раце.

Исто така, сегашната имплементација се извршува само на процесорот и нема хардверско заврзување. Може да се искористи графичка карта за паралелно процесирање со WebGL.

Овој пристап има можност да ги подобри перформансите, но ја зголемува сложеноста на имплементација.

4.2. Имплементација со Processing

Processing е околина за визуелно и креативно програмирање базирана на Java. Таа е многу слична со p5.js бидејќи се креирани и одржувани од истата фондација.

Processing овозможува исцртување во прозорец, содржи библиотека за видео и библиотека за управување на звук со осцилатори, се со слична структура и синтакса како p5.js.

Библиотеката за машинско учење OpenCV за Processing содржи модел за детекција на раце со истите клучни точки и може да се користи за имплементација на апликацијата.

Предностите што доаѓаат со користење на Processing е зголемени перформанси и брзина бидејќи е оптимизирана околината за графика и мултимедија, и затоа ефективно ги користи ресурсите на машината.

4.3. Имплементација во Python

Python е одличен за апликации со машинско учење бидејќи има голем број пакети со отворен код за секоја намена. За имплементација на проектот со python би можело да се искористи OpenCV пакетот за детектирање на раце, Pyo пакетот за генерирање на звук, и Tkinter за визуелен интерфејс и исцртување на точките на дланките.

Предности со овој пристап се тоа што јазикот има интуитивна синтакса за лесна имплементација, и има библиотеки што се оптимизирани за машинско учење, графика и генерирање звук.