

gesichts.musik

Kurs: Audio-Video Programmierung

Dozent: Prof. Dr. Andreas Plaß

Projektdokumentation

Teilnehmer: Tim Zschage (2221141), Vincent Schnoor (2208819)

Aufgabe des Projektes

Das Programm erkennt über eine Kamera/Webcam menschliche Gesichter und deren Gesichtsausdrücke. Daraus berechnet es einen Ton und spielt diesen ab.

Bedienungsanleitung

Gesichts.musik startet wenn die .exe ausgeführt wird. Nach dem Start sieht man ein Programmfenster mit einer Menüleiste mit den Einträgen Video und Play. Unter *Video* kann man die Quelle für die Gesichtserkennung einstellen, entweder eine Videodatei oder über eine Kamera/Webcam. Unter *Play* wird das Video oder die Kameraübertragung gestartet und angezeigt.

Links sieht man das Originalvideo, während man rechts das veränderte Video sieht, in welchem die Merkmale des Gesichtes hervorgehoben wurden die von dem Programm erfasst werden.

Unter den Videofenstern gibt es Einstellungsmöglichkeiten. Die Slider und Checkboxen unter Kalibrierung funktionieren leider nicht, sollten aber dazu da sein, Feinabstimmungen der Gesichtserkennung zu bewirken. Unter Optionen kann mit einem Slider die Lautstärke der Ausgabe des Tones bestimmt werden.

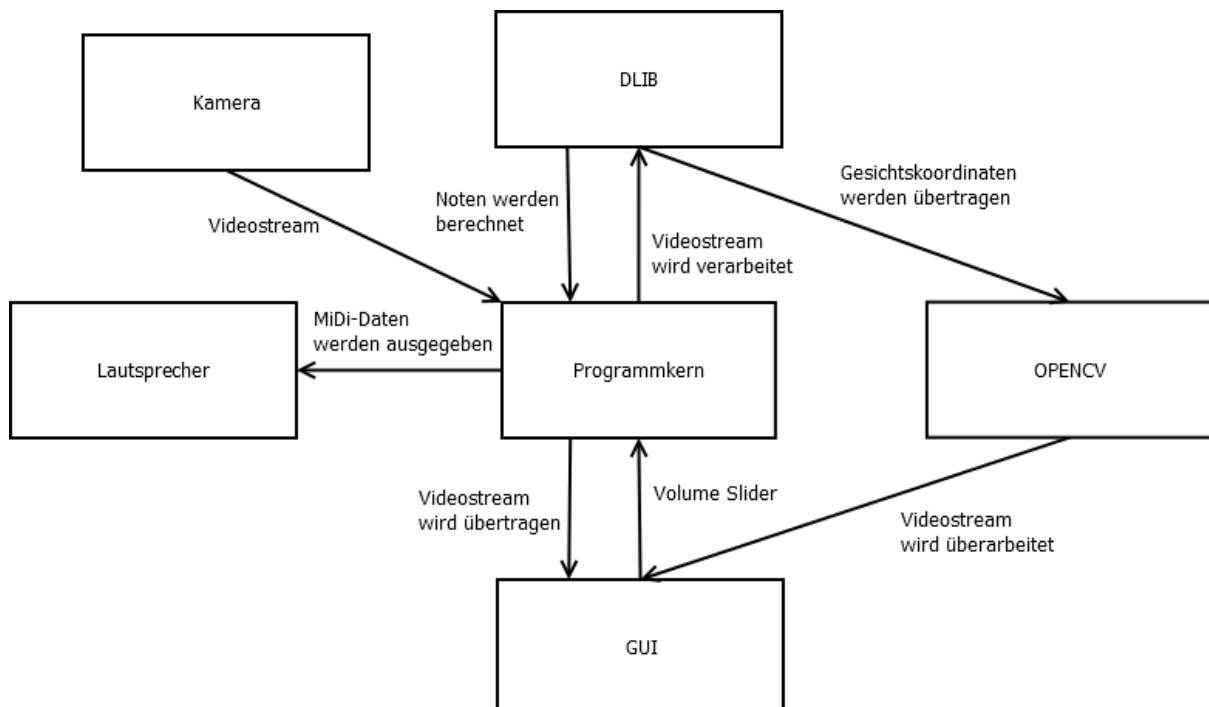
Installationsanleitung

Damit gesichts.musik erfolgreich gestartet werden kann, müssen einige Schritte getan werden.

1. Visual Studio 2015 Community muss installiert sein (<https://www.visualstudio.com/downloads/>), insbesondere der C++ Teil.
2. OpenCV 3.1.0 muss installiert (<http://opencv.org/downloads.html>) und an einen beliebigen Ort entpackt werden.
3. Umgebungsvariablen einstellen.
 - a. Start->Suchen nach:->Umgebungsvariablen für dieses Konto
 - b. Eine neue Variable **OPENCV_DIR** mit dem Pfad zum Build-Verzeichnis (dort wo OpenCV entpackt wurde) definieren, z.B.: C:\opencv\opencv-3.1.0\build\x64\vc14
 - c. Die Umgebungsvariable **PATH** (durch Semikolon abgetrennt) ergänzen mit:
%OPENCV_DIR%\bin
4. Dlib herunterladen (<http://dlib.net/>) und an eine beliebige Stelle entpacken.
5. CMake herunterladen (<https://cmake.org/download/>) und installieren.
6. Dlib mithilfe von CMake kompilieren.
 - a. Kommandozeile öffnen (Start->Suchen nach:->cmd)

- b. Folgendes eingeben:
`cd (Verzeichnis in welches Dlib entpackt wurde)`
`mkdir build`
`cd build`
`cmake -G „Visual Studio 14 2015 Win64“ ..`
`cmake --build . --config Release`
7. Die .pro Datei im Projekt öffnen (FEAS->FEAS->VideoPlayer->videoplayer.pro)
8. Unter win32{..} müssen einige Pfade angepasst werden:
 - a. INCLUDEPATH += D:\Dlib\dlib-19.2 muss in das Verzeichnis geändert werden in welches Dlib entpackt wurde.
 - b. SOURCES += D:\Dlib\dlib-19.2\dlib\all\source.cpp
 D:\Dlib\dlib-19.2 muss in das Verzeichnis geändert werden in welches Dlib entpackt wurde.
 - c. Im Projekt unter VideoPlayer->videoengine->videoengine.cpp, videoformat.h und videoprocessor.h muss
`#include <D:\opencv\opencv\build\include\opencv2\opencv.hpp>` in den Pfad geändert werden an dem OpenCV entpackt wurde.

Systemarchitektur



Die am Rechner angeschlossene Kamera überträgt die Frames des Videostreams an den Programmkern.

Diese Frames werden dann auf dem GUI ausgegeben. Zeitgleich verarbeitet die API Dlib das

Bildmaterial mit seiner internen Gesichtserkennungsfunktion und überträgt die Koordinaten des erkannten Gesichtes an OPENCV. Außerdem berechnet Dlib MiDi-Noten aus den Koordinaten und übermittelt sie an den Programmkern. OPENCV wandelt die Koordinaten so um, dass es sie verwenden kann. Auf Basis der Koordinaten zeichnet OPENCV dann Linien der erkannten Gesichtsregionen in das Bild. Die überarbeiteten Frames werden dann an die GUI gesendet und ausgegeben. Der Lautstärkeregler der GUI schickt den Lautstärkewert an den Programmkern, welcher die MiDi-Daten über den Lautsprecher ausgibt.

Technischer Teilaspekt

Unser wichtigster Teilaspekt, ist die Berechnung der MiDi-Noten:

```
long mh = (upper-lower).length();

    noteNumber = (mh-10) / 8;
    if (noteNumber > 12) {
        noteNumber = 12;
    }
    else if (noteNumber < 0) {
        noteNumber = 0;
    }
}

point lowerA = point(0,500);

for(int i = 19; i < 25; i++) {
    if (d.part(i).y() < lowerA.y()){
        lowerA = d.part(i);
    }
}
int an = d.part(27).y() - lowerA.y();

octave = (an-35) / 3;
if (octave > 11) {
    octave = 11;
}
else if (octave < 0) {
    octave = 0;
}
```

Über die Variable *mh* wird die Öffnung des Mundes berechnet. *Upper* ist der höchste Punkt des Mundes und *lower* der Tiefste. *Notenumber* ergibt sich aus der Mundöffnung, von der wir 10 abziehen um die Werte von Null aufwärts zu haben. Durch eine Division durch 8 bekommen wir einen Wertebereich von 0 – 11. Die *if*-Bedingung darunter stellt sicher, dass man nicht aus dem Wertebereich rausgeht.

Die Variable *an* ergibt sich aus der Nasenmitte und dem tiefsten Punkt der Augenbrauen.

Aus dieser Variable wird dann die *octave* berechnet.

Am Ende wird die MiDi-Note aus der *notenumber* und der *octave* berechnet:

```
octave * 12 + notenumber;
```

Einige Werte sind Erfahrungswerte die bei uns am besten funktioniert haben. Um das Programm flexibler zu gestalten wollten wir Slider einbauen, mit deren Hilfe man diese Werte verändern kann. Aufgrund von Zeit – und Personalmangel hat es dafür leider nicht mehr gereicht.

Auswertung / Fazit

Das Projekt, welches im Kurs „Audio-Video Programmierung“ entstand, wurde am Ende nicht ganz so wie wir es uns vorgestellt haben. Anfangs bestand unsere Gruppe noch aus 4 Leuten, was sich dann auf 2 reduziert hat. Dadurch gestalteten sich die Arbeitsverteilung und die Umsetzbarkeit schwierig. Hinzu kommt, dass wir mit den Tücken der Programmiersprache C++ Probleme hatten und die Kombination verschiedener Frameworks und Bibliotheken sehr zeitaufwendig war.

Besonders die sehr grundlegenden Vorlesungen waren bei einem Projekt dieses Umfanges nicht gerade optimal. Wir hätten uns gewünscht, dass das Framework *QT* und die Bibliothek *OPENCV* umfangreicher erklärt worden wären.

Die Vorstellung der Projekte am Ende des Semesters war leider eher demotivierend.

Es wurden viele Projekte vorgestellt, von denen die meisten zu einem Medientechnikkurs gehörten.

Wir verstehen, dass es zur Fülle der Ausstellung beitrug, dennoch waren diese Projekte deutlich umfangreicher und weitaus aufregender als unseres.

Dennoch fanden wir, dass der Kurs an sich sehr interessant ist und gerade Gesichtserkennung sich als Thema hervorragend eignet. Auch die Erlernung einer weiteren Programmiersprache ist hilfreich.

Nur die Umsetzung war nicht vollends überzeugend.