

PROJEKTDOKUMENTATION

DER GESICHTSMUSIKANT

Audio-Video-Programmierung
WiSe 16/17

29.01.2017

LINH DO #2131896
MARCEL PLAGA (MT) #2202225
MARVIN STEINMETZ #0000000
ALINA BÖTTCHER #0000000

1 AUFGABE DES PROJEKTES

Mit der Software „Gesichtsmusikant“ wird über Gesichtstracking die Bewegung des Gesichtes eingefangen und mit Geräuschen wiedergegeben.

Dabei kann zwischen zwei Modi entschieden werden. Zum einen ist da der Sample Modus. Darin werden abgeschlossene Bewegungen gedeutet und dann ein kurzer Sound abgespielt. So ruft z.B. das Öffnen des Mundes das Trompeten eines Elefanten hervor.

Zum anderen ist da der Synthesizer Modus. Dieser ermöglicht es mithilfe des Gesichtes mit einem Ton zu spielen. Dazu wird auf dem Bildschirm ein Synthesizer eingeblendet, dessen Hebel sich mit der Gesichtsbewegung verändern.

2 INSTALLATIONSANLEITUNG

a) Pure Data:

Zunächst wird „Pure Data Extended Portable“ benötigt. Dieses kann unter folgendem Link heruntergeladen werden:

<http://sourceforge.net/projects/pure-data/files/pd-extended/0.43.4/Pd-0.43.4-extended-windowsxp-i386.zip/download>

Die heruntergeladene Datei muss entpackt werden. Der Ordner „Pd-0.43.4-extended“ muss dann in den Ordner „Gesichtsmusikant“ kopiert/verschoben werden. Später kann die Software dann über die Batchdatei „GesichtsmusikantSynthesizer.bat“ gestartet werden.

b) Öffnen des Projekts mit Qt Creator

Im Projektordner befindet sich ein Ordner, der ebenfalls „FEAS Konsole“ heißt, dort kann man die Datei „FEASKonsole.pro“ mit Qt Creator öffnen. Beim Ausführen des Programms, sollte in der linken Leiste vom Debug-Modus in den Release-Modus gewechselt werden (Im Debug-Modus funktioniert das Programm nämlich sehr langsam). Kompiliert haben wir mit der Qt Version 5.7.0.

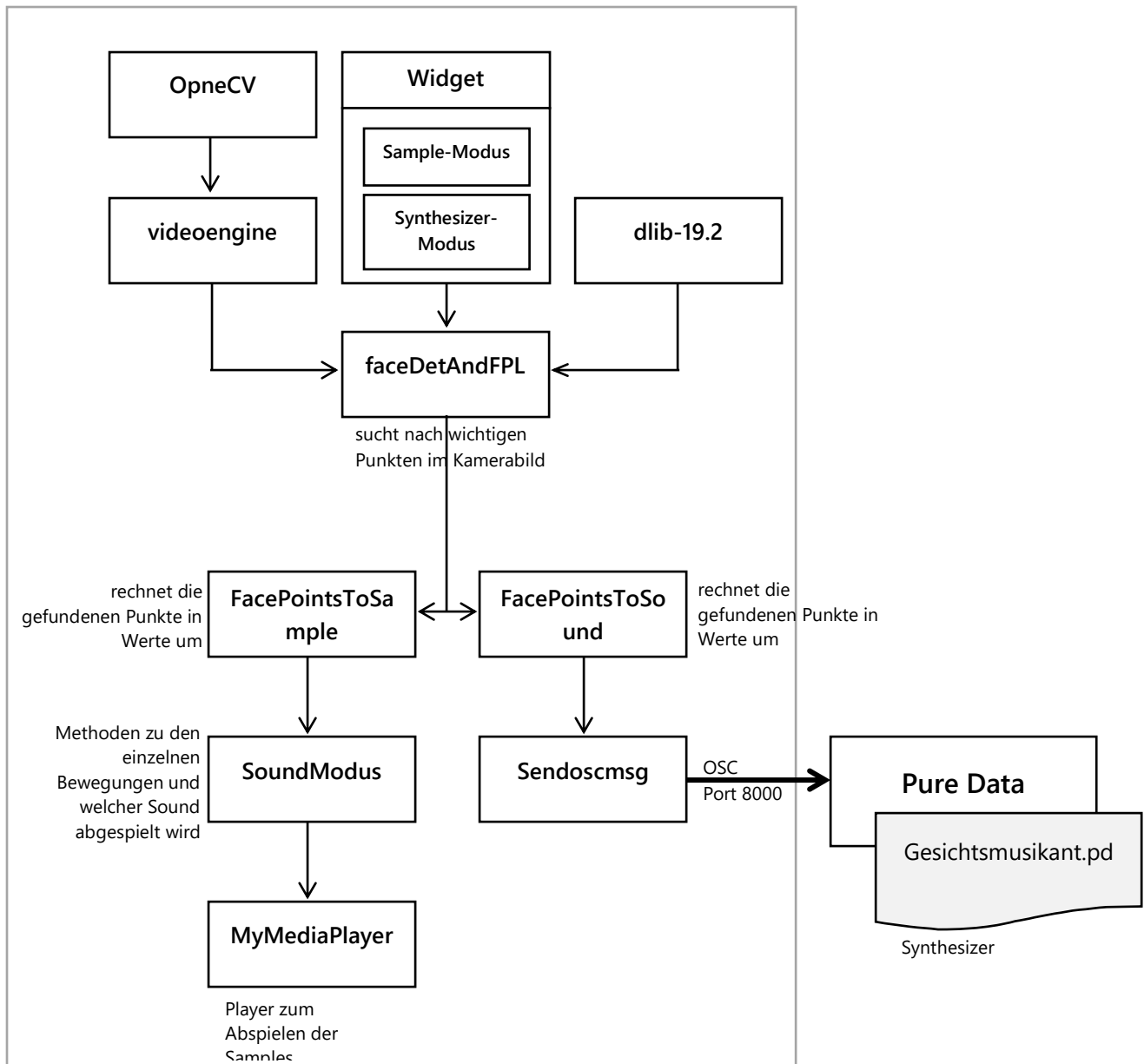
3 BEDIENUNGSANLEITUNG

Nach dem Start der Software erhält der Nutzer zwei Schaltflächen, über welche die entsprechenden Modi gestartet werden können. Im Sample-Modus kann der Nutzer dann nach der folgenden Zuteilung die Audiosamples abspielen:

Gesichtsbewegung	Audiosample
Augenbrauen hoch	Hupe
Mund auf	Adler
Kopf links/rechts rotieren	Mücke/Klirren
Kopf vorne/hinten bewegen	Elefant/Ente
Kopf links/rechts bewegen	Ozean/Blubbern

Nach dem Schließen des Videofensters kann der Synthesizernodus aufgerufen werden. Dieser erfordert zusätzlich, dass die Software PureData mit dem Gesichtsmusikanten-Patch läuft. Zusammen kann dies durch einen Doppelklick auf die Batchdatei „GesichtsmusikantSynthesizer.bat“, welche sich im Ordner Gesichtsmusikant befindet, gestartet werden. Dieser bietet dann Funktionen üblicher Synthesizer, welche zum einen per Maus voreingestellt werden können und zum anderen können die Slider und Regler über die Matrix im Bereich „OpenSoundControl“ auf der rechten Seite frei den Gesichtsbewegungen zugeordnet werden. Die Zeilen stellen dabei die Funktionen dar und die Spalten die unterschiedlichen Gesichtselemente. Es ist auch möglich mehrere Funktionen mit dem gleichen Gesichtselement zu steuern.

4 SYSTEMARCHITEKTUR



5 BESCHREIBUNG EINES TECHNISCHEN TEILASPEKTES

Gesichtsauswertung

In diesem Absatz möchte ich kurz darauf eingehen wie wir die Auswertung unserer Gesichtserkennung gelöst haben.

Für unseren Synthesizer müssen wir aus den Gesichtsmerkmalen, die wir auswerten wollen, Parameter berechnen, die in den Wertebereich von 0-100 fallen.

Durch die von Dlib gelieferten Landmark-Koordinaten, können wir über die Länge der Vektoren zwischen wichtigen Punkten eine grobe Form unserer Parameter berechnen. Diese Werte sind direkt abhängig von der Auflösung der benutzten Kamera, sowie der Kopffentfernung des Nutzers und somit nicht praktikabel für unsere Anwendung. Um diese Faktoren aus unseren Parametern zu entfernen, nutzen wir folgende Ansätze, um unseren Code zu optimieren.

Um unseren Algorithmus von der Kopffentfernung unabhängig zu machen, wird eine Bezugsgröße benötigt, an der wir den Rest der Berechnung orientieren. Für diesen Bezug nutzen wir die Breite des Gesichts und ermitteln das Größenverhältnis der Parameter zur Gesichtsbreite. Aus diesen Verhältnissen können wir in jedem Frame berechnen wie viel z.B die Länge des komplett geöffneten Mundes beträgt. Aus dieser errechneten Bezugslänge und der aktuellen Länge wird errechnet wieviel Prozent der maximalen Mundöffnung erreicht ist. Durch das Verwenden von Prozentwerten und Verhältnissen kann ebenfalls die Auflösung der verwendeten Kamera unsere Werte nicht verfälschen.

```
mouthOpen = (double)
((shapes[0]. part(62) - shapes[0]. part(66)). length())
* 100 /
((shapes[0]. part(0) - shapes[0]. part(16)). length() * ratioMouth);
```

Die Bezugsgrößen für einzelne wichtige Gesichtsmerkmale sind in unserer Version als Variablen hinterlegt und müssen auf einzelne Anwender angepasst werden, sollen aber in einer zukünftigen Version über Slider während des Programmablaufs angepasst werden können. Die errechneten Werte müssen am Ende nur noch auf den Bereich von 0-100 eingegrenzt werden, um zu den Parametern unseres PureDataPatches zu passen.

6 AUSWERTUNG:

a) Was hat funktioniert

Insgesamt hat die Umsetzung der geplanten Funktionen mit Sample- und Synthesizer-Modus schon mal funktioniert. Ziemlich zuverlässig hat dabei dlib gearbeitet. Nur bei sehr schlechten Lichtverhältnissen und bei zu üppigen Bärten hat es Erkennungsprobleme gegeben.

Ebenso hat die Implementierung der Übertragung der Steuerparameter an PureData über das Open Sound Control Protokoll ohne größere Probleme funktioniert.

Ebenfalls gut funktioniert hat die Zusammenarbeit der Computer. Wir hatten keine Versionsprobleme oder irgendwelche anderen Probleme die bei der Zusammenarbeit mit mehreren Computern und mehreren Programmen entstehen können.

Womit wir auch sehr zufrieden waren, war das Zusammenfügen zu einem ganzen Programm. So hatte jeder erstmal allein für sich seine Aufgabe programmiert und fertig gemacht. Zudem hatte jeder Funktionen geschrieben, sodass andere aus der Gruppe leicht die gewünschte Funktion verwenden konnten.

Die Kommunikation innerhalb verlief auch insgesamt ziemlich gut durch Treffen in der Uni, Skype und Telegram.

b) wo waren die Grenzen

Stellenweise waren tiefgreifendere C++ Kenntnisse eine Hürde. So war nicht wirklich nachvollziehbar, warum das Programm bei der Integrierung der zusätzlichen Klassen ohne genaue Fehlerangaben nicht gestartet ist und letztendlich die Lösung dafür war, die Klassen über Pointer einzubinden.

Bei der Einrichtung von dlib gab es auch zunächst größere Probleme, diese mit Qt Creator zu benutzen. Im Internet gab es zum Beispiel öfter den Vorschlag das Kompilieren mit „CMake“ durchzuführen, da dies performanter ist. Dies haben wir nach stundenlangem Herumprobieren jedoch nicht hingekriegt und mussten stattdessen die ganze source.cpp in das Projekt miteinbinden.

Ein anderes Beispiel wäre, dass die Überlegung da war, dass die Überwachung der Punkte in einem Thread läuft und das Umrechnen und Ausgeben in einem anderen, sodass beides sozusagen unabhängig voneinander läuft. Leider ist es nicht gelungen, das zu implementieren.

Weiterhin ist immer der Fehler aufgetaucht, dass er ein „&“ erwartet hat, obwohl eines vorhanden war.

Zudem sind im Synthesizernmodus die Echtzeitanforderungen gegebenenfalls etwas kritisch zu betrachten. Zwar ist hier nur eine äußerst minimale Verzögerung wahrnehmbar, für präzise Soundanwendungen könnte dies vielleicht aber schon zu viel sein. Auch ist die untereinander unabhängige Auswertung der Gesichtsp Parameter nicht immer gegeben. Beispielsweise wenn der Kopf nach oben oder unten bewegt wird, wirkt sich dies währenddessen auch merkbar auf Parameter wie Mund und Augenbrauen aus. Auch neigen einige Parameter etwas dazu unwillkürlich zu springen.

c) was würden wir beim nächsten Mal anders machen

Dadurch, dass wir eine Aufgabe hatten, die für uns alle absolutes Neuland war, war vorher schlecht abzuschätzen, wie die genaue Klassenstruktur aussehen wird. Das hat dazu geführt, dass wir zwischenzeitlich etwas Chaos hatten, welche Klasse für was zuständig ist.

Durch andere Projekte ist das Projekt zum Anfang hin etwas vernachlässigt worden. Eine bessere Zeitplanung wäre für das nächste Mal sicherlich ein gutes Ziel.