Projektkonzept – Emotionsbasierte Audiosynthese (EbAS)

Corinna Bohnenberger (2217555)
Hennes Römmer (2217606)
Jan Heimann (2210236)
Raoul Bickmann (2217470)

Einleitung

Für die Prüfungsleistung im Fach Audio-Video-Programmierung wird unsere Gruppe, bestehend aus Corinna Bohnenberger, Jan Heimann, Hennes Römmer und Raoul Bickmann einen Facial Expression Audio Synthesizer entwickeln. Dieser soll in C++ umgesetzt und am Ende des Semesters, zusammen mit anderen Projekten, im Forum Finkenau ausgestellt werden. Konkret hat sich unsere Gruppe dazu entschlossen ein Programm zur emotionsbasierten Audiosynthese, genannt EbAS, zu entwickeln. Dafür soll bei der Ausstellung ein Videostream eingelesen werden. In diesem werden das Gesicht eines Nutzers und dessen Emotion erkannt. Abhängig welche Emotion erkannt wurde, werden verschiedene MIDI Files abgespielt. Im Folgenden werden wir Genaueres dazu und zur Umsetzung des Projektes erläutern.

Projektziel

Unser Projekt "Emotionsbasierte Audiosynthese" (EbAS) soll nach Fertigstellung am 24. Januar 2017 im Forum Finkenau ausgestellt werden. Dazu bauen wir neben einem Rechner und einem Bildschirm, eine Webcam und Lautsprecher auf und starten das Programm. Es soll nun im Folgenden alles weitere von allein funktionieren. Der Benutzer stellt sich so vor die Webcam, dass die Kamera sein Gesicht erfassen kann. Wenn ein Gesicht gefunden wurde, wird dieses auf den Bildschirm übertragen. Auf der grafischen Oberfläche befindet sich außerdem ein Button, mit dem der Benutzer optional die Linien der Gesichtserkennung ein- oder ausblenden kann. Wenn mehr als ein Gesicht erkannt werden, öffnet sich eine Fehlermeldung mit dem Hinweis, dass sich nur eine Person vor der Webcam befinden darf. Nachdem eine Emotion erkannt wurde, wird entsprechend ein kurzes Musikstück abgespielt und angezeigt, um welche Emotion es sich handelt. Solange die Musik abgespielt wird, ist es nicht möglich eine neue Emotion zu erkennen. Es soll neben den Emotionen "Glück", "Trauer", "Wut" und "Überraschung" auch eine neutraler Gesichtsausdruck erfasst werden können.

Anforderungsanalyse

Aufgabe der Software ist es, aus einem von der Webcam erhaltenen Videostream, das Gesicht des Benutzers zu erkennen und seine aktuelle Emotion auf Basis des Gesichtsausdrucks abzulesen. Anschließend wird ein musikalisches Motiv abgespielt, welches dem Gemütszustand des Benutzers entspricht.
Das Hauptaugenmerk wird darauf liegen, dass dem Anwender ein möglichst homogenes Erlebnis geboten. Es geht schließlich um eine Interaktion von Nutzer und Anwendung, die gerade auf den menschlichen Emotionen basiert. Dabei stellt sich das Programm dem Nutzer in drei Bereichen dar:

1. Die Verbindung zum Nutzer mittels der Webcam
2. Die grafische Oberfläche
3. Das erzeugte Klangerlebnis

Es soll der Eindruck erweckt werden, dass es zu einer Art Dialog zwischen Rechner und Mensch kommt. Um dies zu erreichen, darf nicht das Bild einer Maschine entstehen, welche einfach nur Schritt für Schritt ihren Algorithmen nachgeht. Vielmehr muss das Programm dem Nutzer durchgehend signalisieren, dass es auf ihn achtet, ihm gewissermaßen zuhört. Dies kann zum einen über das visuelle Feedback geschehen und zum anderen natürlich über die synthetisierten Melodien. So wird schon während der Erkennungsphase der Emotion klargestellt werden müssen, ob es der Anwendung möglich ist seinem Gegenüber den aktuellen Gemütszustand ablesen zu können. Ist die Emotion erst einmal erkannt, übernimmt der Klang die Interaktion. Daraus ergeben sich für die Melodien, welche erklingen sollen, bestimmte Anforderungen. Sie sollten die Emotion, welche sie repräsentieren, präzise und eindeutig widerspiegeln. Ob das gelingt hängt unter anderem davon ab, wie sie harmonisch und vom Melodieverlauf her aufgebaut sind. So sollte eine Melodie, die Wut darstellt, viele Dissonanzen enthalten und sehr energetisch und unruhig wirken. Freude hingegen sollte eher konsonant klingen, mit vielen Dur-Akkorden arbeiten und eine unbeschwerte und leichte Melodie enthalten. Insgesamt sollten die wiedergegebenen Motive kurz und eingängig sein. Auf diese Weise wird dem Anwender schnell aufgezeigt, zu welcher Emotion sein Gesichtsausdruck zugeordnet wurde, sodass gleich darauf der nächste Ausdruck gedeutet werden kann. Aber auch der gesamte Klangeindruck spielt eine Rolle. Dazu zählt eine passende Instrumentierung sowie ein wirkungsvoller Einsatz von Effekten wie Equalizer und Hall bzw. Reverb.
Für ein einheitliches und intuitives Erlebnis ist es zudem essentiell, die Bedienung der Anwendung so selbsterklärend und einfach wie möglich zu halten. Darum findet die Interaktion rein über die Webcam bzw. das von ihr aufgezeichnete Video vom Nutzer statt. Die Benutzung von Maus oder Tastatur soll weitestgehend vermieden werden um die Bedienung möglichst simpel zu halten.
Eine typischer Anwendungsfall könnte wie folgt aussehen: Der Nutzer startet das Programm. Das ist die einzige Situation in der die Maus benutzt werden muss. Für den Fall, dass sich mehrere Gesichter im Aufnahmebereich der Webcam befinden, wird dem Anwender in dem Anwendungsfenster ein Hinweis angezeigt, dass nur ein Gesicht zur Zeit gedeutet werden kann. Sobald nur noch ein Gesicht erkannt wird, geht der Ablauf normal weiter.
Um den Benutzer nicht unnötig von dem Erlebnis abzulenken, sollten noch ein paar weitere Punkte erfüllt sein. Fehldeutungen der Emotion sollten vermieden werden, der Übergang von der Erkennung zur Audioausgabe und zurück sollte fließend vonstatten gehen und die Erkennung sollte nach Möglichkeit kurz gehalten werden.

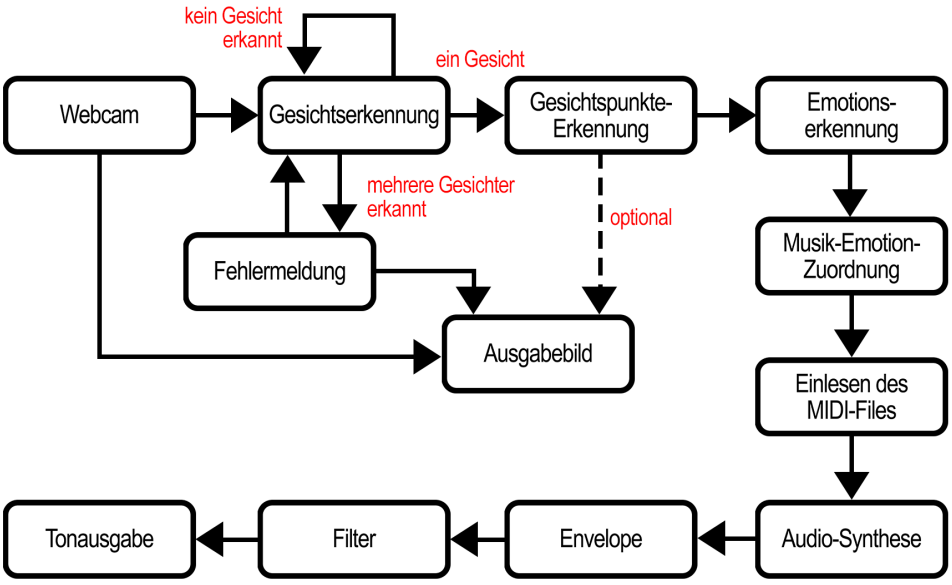
Technische Rahmenbedingungen

Als Framework für C++ benutzen wir QtCreator. Dies bietet sich an, da wir durch die Vorlesung schon mit dem Programm vertraut sind und wir damit einfach ein User Interface erstellen können.
Ebenfalls aus der Vorlesung ist OpenCv bekannt. Diese Open Source Bibliothek werden wir für den ersten Schritt, das Einlesen des Videostreams, verwenden. Anschließend werden die einzelnen Frames von Methoden von Dlib, einer weiteren Open Source Bibliothek, verarbeitet. Dlib bietet den Vorteil, dass es nicht nur Methoden zur Gesichtserkennung gibt, sondern es auch noch möglich ist die Mimik zu erkennen. Dlib ist in der Lage das Erkennen dieser Facial

Landmarks innerhalb von 1 ms durchzuführen. Dies wird es uns erleichtern die Emotionen des Nutzers zu bestimmen. Ein weiterer Vorteil von Dlib ist, dass es eine Header-only Library ist.
Außerdem werden wir OpenCV verwenden um die Facial Landmarks darzustellen und das aktuelle Bild der Webcam auszugeben.

Sobald eine Emotion erkannt wurde wird die entsprechende MIDI-Datei eingelesen. Dies wird mit der MIDI-Datei-Bibliothek “jdksmidi” für C++ geschehen. Die aus diesem File entnommenen MIDI-Events werden dann in einen Vektor, samt ihrer Zeitinformation gespeichert. Eine Sequenzer-Klasse übernimmt nun das zeitliche Einordnen dieser Events und triggert sie an der richtigen Stelle. Die MIDI-Events enthalten verschiedene Informationen, welche für die anschließende Audio-Synthese wichtig sind. So geben sie etwa die Note-On und Note-Off Befehle, geben Auskunft über die Tonhöhe, sowie die Dynamik der einzelnen Töne und ihre Instrumentierung. Mit diesen Daten kann nun die Oszillator-Klasse, welche mittels additiver Synthese die Klangfarben realer Instrumente nachbildet, die Audiodaten generieren. Diese werden dann durch eine Envelope-Klasse auch noch im Zeitbereich bearbeitet, was ebenfalls zum Klangcharakter beiträgt. Zuletzt werden die Audiodaten verschiedenen Effekten (FIR-Filtern) übergeben. Auf diese Weise werden Hall und Equalizer implementiert. Die fertig verarbeiteten Audiodaten landen dann in dem Buffer der AudioEngine, welche die Schnittstelle zum Audio-Interface ist, wo dann der Ton abgespielt wird.
Dieses Blockdiagramm stellt den geplanten Ablauf unseres Programms dar.

Flussdiagram Systemarchitektur



Bedienkonzept

Unser Programm wird aus einem Fenster bestehen. Auf diesem ist das Bild der Webcam zu sehen. In der oberen linken Ecke wird die aktuelle Emotion bzw. der entsprechende Gesichtsausdruck dargestellt. Hierdurch soll es ermöglicht werden, zu kontrollieren, ob der Algorithmus den Gesichtsausdruck richtig erkannt hat.
Für den Benutzer wird es die Möglichkeit geben, über eine Checkbox auszuwählen, ob er die Linien des Gesichts angezeigt bekommen möchte, oder nicht.

Konzeptbilder

Zeitplan

Unseren ersten Prototypen werden wir bis zum 13.12. fertigstellen. Dieser soll in der Lage sein neben einem glücklichen und traurigen auch einen neutralen Gesichtsausdruck zu erkennen. Zu diesen Emotionen sollen bereits die entsprechenden Midi–Dateien abgespielt werden.
Nach Fertigstellung des Prototypen werden wir uns um die Erkennung weiterer Emotionen und der Erstellung der dafür benötigten Midi–Dateien kümmern.
Zusätzlich werden wir das von uns bereits beschriebene User Interface umsetzen.
Bis zur Präsentation am 24. Januar soll das Programm funktionstüchtig und die Arbeit an der Dokumentation fast abgeschlossen sein, sodass diese am 28. Januar abgegeben werden kann.

Aufwand

Der von uns geschätzte Aufwand wird sich auf ungefähr 35 Personentage, also knapp 290 Arbeitsstunden belaufen. Diese setzen sich zusammen aus gemeinsamen Besprechungen über die bisher erledigten Aufgaben und kommende Aufgaben. Der Großteil dieser Zeit ist allerdings die Umsetzung. Hierfür rechnen wir noch mit in etwa 50 Stunden Aufwand pro Person.

Teamplanung

Jedes Mitglied unseres Teams soll in der Lage sein, jeden bei allen anfallenden Aufgaben zu unterstützen. Dennoch haben wir uns entschieden, zwei Teams, je eins für Video und Audio, zu bilden. Raoul und Hennes werden sich um die Umsetzung des Videoteils kümmern. Hierunter fallen die Ausgabe des Bildes der Webcam, das Erkennen der Emotionen und deren Ausgabe. Raoul und Hennes werden sich auch um die Umsetzung des User Interfaces kümmern.
Jan und Corinna beschäftigen sich mit der Audiosynthese zur Erzeugung der entsprechenden MIDI-Dateien und deren Implementierung. Außerdem entwickeln sie die Schnittstelle, die zur Zuordnung der Musik zu den Emotionen notwendig ist. Schlussendlich kümmern sich Jan und Corinna um das Hinzufügen von Effekten und die Ausgabe der Töne.