

Straßenmusik

Projektdokumentation

Audio-Video-Programmierung

Media Systems/Medientechnik WiSe 2020/21

HAW Hamburg

Jakob Sudau

Prof. Dr. Andreas Plaß

Fabian Hensel

#2364627

Daniel Friesen

#2340482

Jakob l'Etienne

#2365009



Videoverarbeitung mit Python/OpenCV

Der Originalframe des Videos wird zuerst in ein Schwarz-Weiß-Bild umgewandelt aus welchem dann die durchschnittlichen Helligkeitswerte innerhalb der drei Erkennungszonen berechnet werden. Diese Werte entsprechen später den Tonhöhen.

Mit Hilfe des Backgroundsubtractor MOG2 wird aus dem Originalframe eine Binäre-Differenzmaske erstellt. Auf dieser sind die Fahrzeuge grob erkennbar in Form von weißen Blobs auf schwarzem Hintergrund dargestellt. Genauer: Die sich bewegenden Fahrzeuge werden mit weißen Pixeln maskiert, während alles was sich nicht bewegt dem Hintergrund zugeordnet und Schwarz maskiert wird.

Diese Maske wird anschließend mit einem Medianfilter bearbeitet um kleine Objekte, welche kein Fahrzeug sind, herauszufiltern.

Danach wird die Maske noch mit einem Morphologischen-Filter (Closing) bearbeitet, um Lücken innerhalb der Fahrzeug-Blobs zu schließen.

Bevor die Maske ausgewertet wird, werden noch Rechtecke um die Erkennungszonen auf dem Originalframe gezeichnet.

Ob sich ein Fahrzeug in einer Erkennungszone befindet oder nicht, wird folgendermaßen ausgewertet:

Zuerst wird geprüft ob in dem kompletten, horizontalen Bildausschnitt in welchem die Erkennungszonen liegen, der Durchschnittswert von schwarzen und weißen Pixeln unter einem Schwellenwert (80) liegt. Dadurch soll verhindert werden, dass bei einer plötzlichen Aufhellung des Bildes (z.B. durch das Vorbeifahren eines weißen LKWs), fälschlicherweise ein Fahrzeug erkannt wird.

Nun wird geprüft ob innerhalb einer Erkennungszone der Durchschnittswert von schwarzen und weißen Pixeln unter 77 liegt. Dieser Wert entspricht einem Verhältnis von 30% weißen zu 70% schwarzen Pixeln. Wenn dem so ist und kein Ton bereits gespielt wird, wird ein Midi-Signal für das Starten eines Tons gesendet. Hier wird die bereits oben berechnete durchschnittliche Helligkeit als Note und die für diese Spur festgelegte Oktave als Anschlagstärke gesendet. Zusätzlich wird der Noten-Wert als letzter abgespielter Ton gespeichert.

Wenn kein Fahrzeug erkannt wird und ein Ton bereits abgespielt wird, wird ein Midi-Signal für das Stoppen eines Tons gesendet. Der zuvor gespeicherte Ton wird nun als Note übermittelt.

Am Ende wird der Originalframe mit den gekennzeichneten Erkennungszonen angezeigt.

Klangerzeugung in Javascript

Als Grundlage für die Klangerzeugung in Javascript wurde ein einfacher Synthesizer über die Web Audio API erstellt.

Die eingehenden MIDI Signale repräsentieren nicht direkt die gespielten Noten, sondern die Helligkeit des erkannten Autos (MIDI Note) und die Oktavlage des zu spielenden Tons (Velocity).

Um diesen Helligkeitswert in einen Notenwert umzuwandeln wurde eine Funktion erstellt, die diesen Wert in eine ganze Zahl zwischen 0 und N konvertiert, wobei N die Anzahl der Töne in der vorgewählten Tonleiter ist.

Die Tonleiter ergibt sich aus einem Array, in dem ausgehend von einem Grundton die musikalischen Intervalle gelistet sind. Aus diesen ergibt sich die jeweilige diatonische Tonleiter. Auf diese Weise ist es auch sehr einfach den Grundton in andere Oktavlagen zu übertragen.

Dur:

Note	1 (Grundton	2	3	4	5	6	7
Halbtöne zum nächsten Ton	2	2	1	2	2	2	1

Moll

Note	1 (Grundton	2	3	4	5	6	7
Halbtöne zur nächsten Note	2	1	2	2	1	2	2

Pentatonik

Note	1 (Grundton	2	3	4	5
Halbtöne zur nächsten Note	3	2	2	3	2

Zeitaufwand

Tätigkeit	Tatsächliche Zeit in h	Geschätzte Zeit in h
Ideenfindung	5	
Konzeption	10	
Programmierung	30	20
Projektdokumentation	5	
Saldo	50	20