

Technisches Dokument: Licht-Saiter

von Nikolai Stecker (2240250) und Florian Völkers (2140739)

im Rahmen des Faches Audio-Video-Programmierung

des Studiengangs Media Systems

Projektbeschreibung

Licht-Saiter ist ein 6-saitiges Musikinstrument, bei dem der Nutzer Noten durch Unterbrechung von sechs verschieden farbigen Lichtstrahlen spielt. Jeder Lichtstrahl wird vertikal in 12 Bereiche unterteilt, die jeweils eine Note repräsentieren. Die insgesamt sechs Lichtstrahlen stellen jeweils eine Oktave (C2 bis H7) dar.

Außerdem hat der Nutzer die Möglichkeit in einer Webapplikation eines von 44 Instrumenten auszuwählen und folgende Effekte einzustellen: Volume, Attack, Release, Duration, Sustain, Decay und Reverb. Ein visuelles Feedback, welche der Saiten gespielt wird, ist durch aufleuchtende Flächen in der Webapplikation gewährleistet.

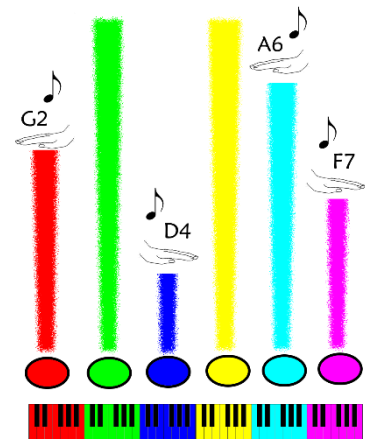


Abbildung 1: Funktionsweise des Licht-Saiters. Eigene Grafik vom 15.01.2018.

Eine Live-Demonstration des Instruments findet man [hier](#).

Technische Umsetzung

Die Licht-Saiten

Um die sechs Saiten aus Lichtstrahlen zu erhalten, haben wir LED-Lichtstrahler aus dem Lichtlabor ausgeliehen. Bei diesen konnten wir die RGB-Werte und die Lichtstreuung einstellen. Dies war nötig, um sechs eindeutig identifizierbare Lichtstrahlen zu erhalten. Die Strahler wurden in einem geringen Abstand nebeneinander aufgestellt. Bei der Einstellung der Farben wurde darauf geachtet, dass es einen größtmöglichen Kontrast zu den Nachbarn gibt. Daher entschieden wir uns dazu, die Primär- und Sekundärfarben zu nutzen. Wir stellten sie in der Reihenfolge Rot, Grün, Blau, Gelb, Cyan und Magenta auf.

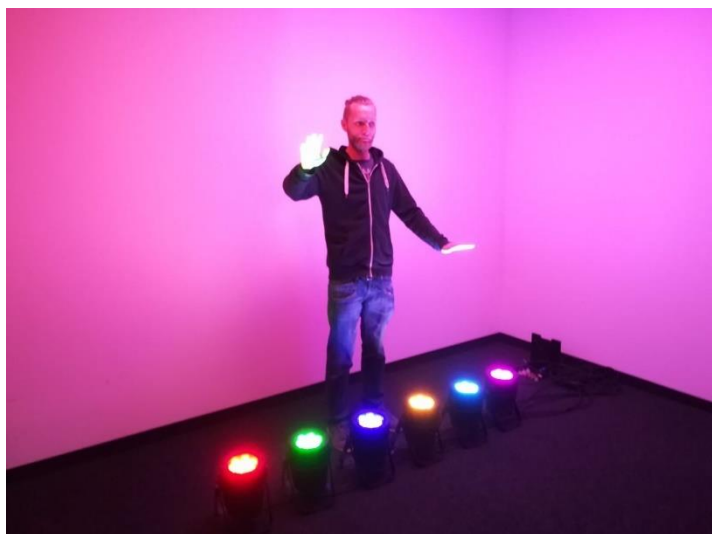


Abbildung 2: Nikolai Stecker beim Spielen des Licht-Saiters. Eigene Grafik vom 16.01.2018.

Die Bilderkennung

Um zu erkennen auf welcher Höhe der Nutzer die Lichtstrahlen unterbricht und damit die Note spielt, verwendeten wir eine Videokamera, die an einem Laptop angeschlossen war. Wir nutzten QT-Creator mit der Programmiersprache C++, um eine Applikation zu erstellen, mit der das erhaltene Videobild Frame für Frame nach bestimmten Faktoren abgesucht wird.

Zunächst muss die Applikation kalibriert werden. Dazu wählt man als erstes eine der sechs voreingestellten Farben aus. Nun stellt man die Minimal- und Maximalwerte für den Farbwert (hue), die Farbsättigung (saturation) und den Hellwert (value) ein. Um zu erkennen, welche Werte für die Kalibrierung richtig sind, sieht der Nutzer ein zweites Bild, bei dem alle Pixel, die sich innerhalb des Wertebereiches befinden, weiß und alle anderen schwarz gefärbt sind. Ist die Kalibrierung dieses Objektes abgeschlossen, so kann der Nutzer die Werte speichern und das Objekt wird anschließend im Realbild erkannt.

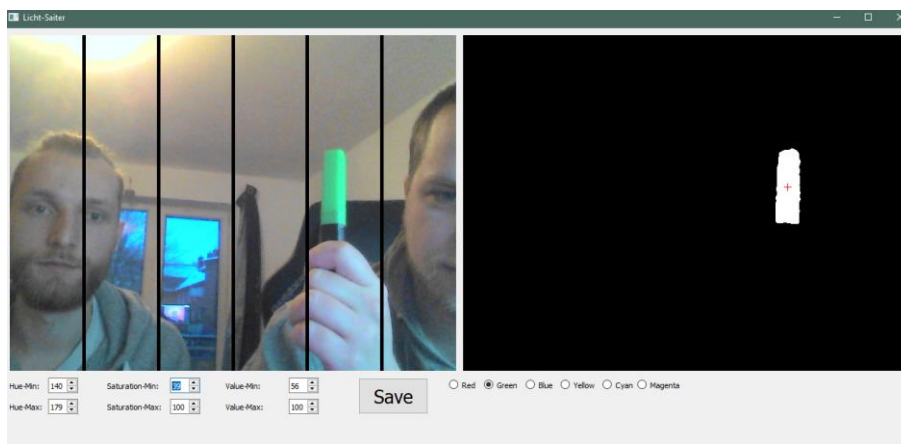


Abbildung 3: Die Videoerkennungs-Applikation bei der Kalibrierung eines grünen Textmarkers zu Testzwecken. Eigene Grafik vom 11.01.2018.

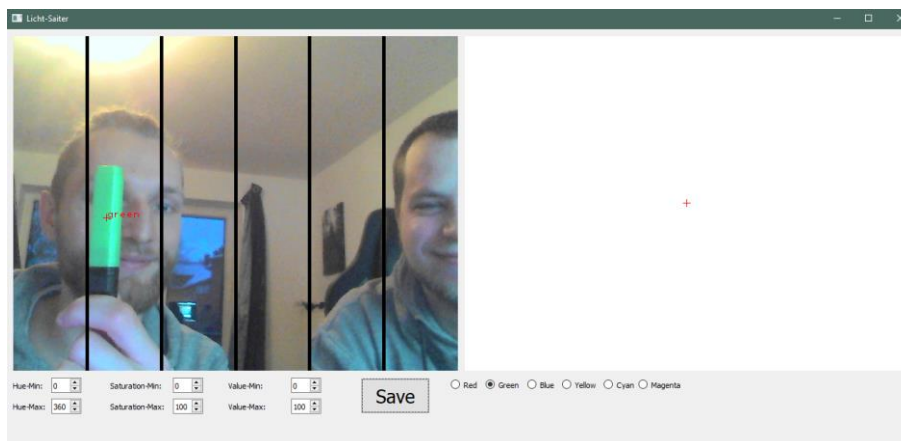


Abbildung 4: Die Videoerkennungs-Applikation nach der Kalibrierung. Das Objekt wird im Realbild wiedererkannt. Eigene Grafik vom 11.01.2018.

Wie man im Screenshot erkennen kann, ist das Realbild in sechs Bereiche eingeteilt. Diese Bereiche dienen zur Unterscheidung der Farben. Außerdem helfen sie beim Aufbau und der Ausrichtung der Lichtstrahler, da diese in einem passenden Abstand voneinander und von der Kamera stehen müssen. Bei der Objekterkennung wird je nach abgespeichertem Farbwert nur einer der sechs Bereiche abgesucht. Dies bedeutet, dass ein als grün abgespeichertes Objekt nur im grünen Farbbereich erkannt werden kann. So werden zum einen eventuelle

Farbüberlagerungen ausgeschlossen und zum anderen wird die Laufzeit des Programms stark reduziert.

Um nun zu erkennen, welche Note gespielt wurde, wird die Position des Objektes als Referenz genommen. Die horizontale Position, also in welchem Farbbereich das Objekt ist, gibt an in welcher Oktave der Ton ist. Über die vertikale Position im Bild wird bestimmt, welche Note gespielt wird. Dazu wird die y-Koordinate des Objektes erkannt und einem von zwölf Notenbereichen zugewiesen. Die Notenbereiche werden durch MIDI-Noten bestimmt. Wir haben uns dazu entschieden unseren sechs verschiedenen Saiten die Noten von C2 bis H7 zuzuweisen. Dies entspricht den MIDI-Noten 36 bis 107. Die MIDI-Note wird dann über den Befehl `sendNoteOn(channel, note, velocity)` per LoopBe Internal MIDI an die Webapplikation gesendet.

Oktave #	MIDI Noten-Benennung											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	H
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

Abbildung 5: MIDI Noten-Benennung. Aufgerufen von <http://www.sengpielaudio.com/Rechner-notennamen.htm> am 18.01.2018.

Um einen Überblick über den Ablauf der Applikation zur Objekterkennung zu erhalten, haben wir die wichtigsten Abläufe in einem Flussdiagramm dargestellt.

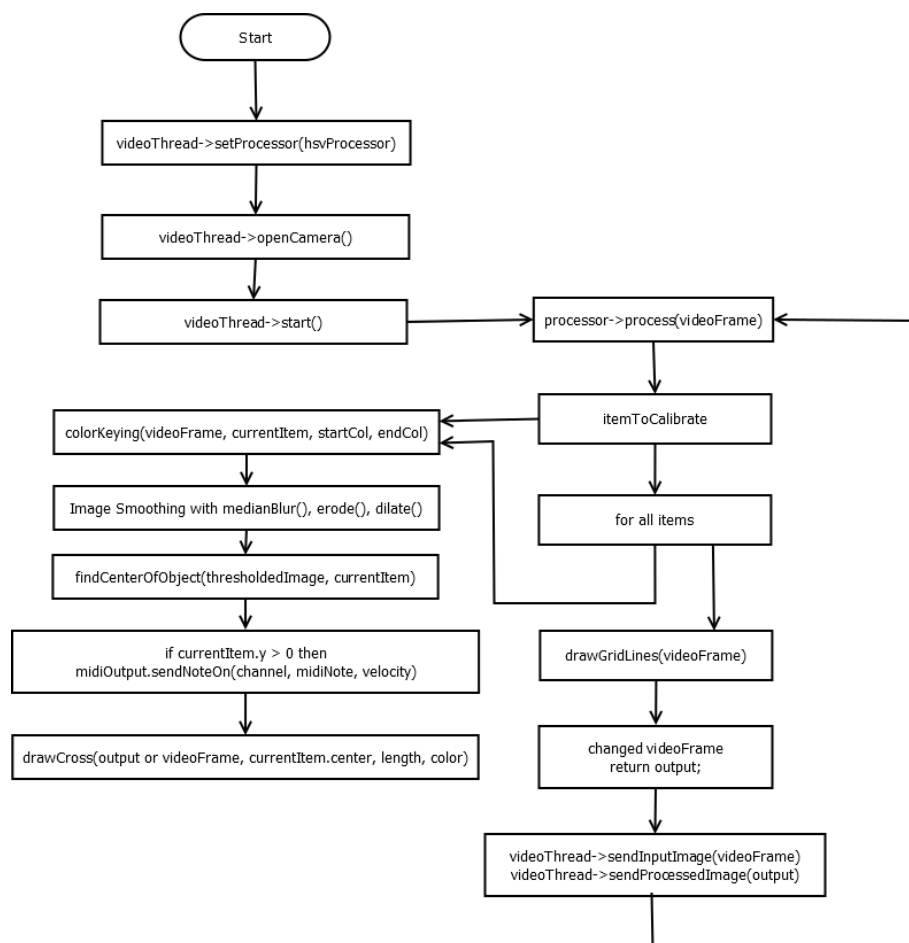


Abbildung 6: Vereinfachtes Flussdiagramm zur Darstellung des Ablaufs der Objekterkennung im Videobild mit openCV. Eigene Grafik vom 18.01.2018.

Zur Videobildererkennung wurde die bereitgestellte Bibliothek Videoengine genutzt. Diese wurde ein wenig verändert, damit auch das Eingangsbild durch den Prozessor verändert werden kann. Der Prozessor wurde selbst entwickelt. Nach dem Start der VideoEngine durchläuft dieser in einer Schleife immer wieder die folgenden Prozesse.

Zu Beginn wird das Eingabebild in den HSV-Farbraum übersetzt, weil damit die Farberkennung der Objekte besser zu gewährleisten ist. Anschließend durchlaufen sowohl das aktuell zu kalibrierende Objekt als auch die bereits gespeicherten Objekte den Prozess des Color Keyings. Um ungewolltes Rauschen herauszufiltern werden drei von openCV bereitgestellte Funktionen genutzt (medianBlur, erode, dilate). Da wir erfahren möchten an welcher Stelle sich das Objekt im Videobild befindet, wird im nächsten Schritt die Mitte des Objektes gesucht. Bei bereits kalibrierten Objekten findet an diesem Punkt die Übertragung der Noten per MIDI statt. Das Senden der MIDI Note findet jedoch nur statt, wenn sich die Note verändert hat. So wird einer Übersteuerung entgegengewirkt.

Die Webapplikation

In der Webapplikation wird mit Hilfe der Web-Audio-API und der Bibliothek [soundfont-player](#) aus den, von der Bildererkennung gesendeten, MIDI-Noten Musik erzeugt.

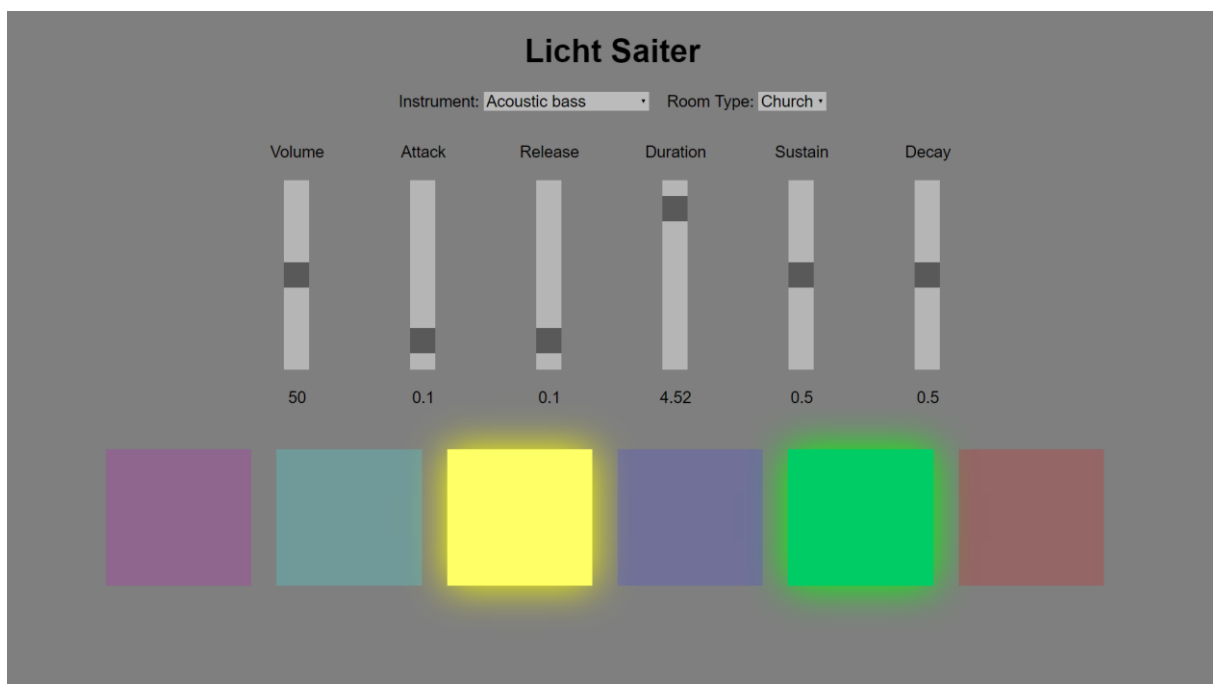


Abbildung 7: Die Webapplikation. Mit angespieltem gelben und grünen Lichtstrahler. Eigene Grafik vom 13.01.2018.

Auf der grafischen Oberfläche hat der Nutzer die Möglichkeit die erzeugten Töne zu beeinflussen. Er kann über ein Dropdown-Menü das aktuelle Instrument ändern. Die verschiedenen Instrumente werden mit SoundFonts umgesetzt, ein SoundFont enthält Muster und Einstellungen bezüglich Hüllkurve, Filter-Eigenschaften und andere Aspekte, die das jeweilige Instrument ausmachen.

Der SoundFont wird mit Hilfe des soundfont-players über die MIDI-Noten gelegt. Die zum gewählten Instrument benötigte Soundfont wird aus einem [GitHub-Repository](#) geladen.

Über ein weiteres Dropdown-Menü kann er einen von 4 Reverb-Effekten (room, church, cave, garage) einstellen. Diese werden mit Hilfe einer Convolver-Node, die zwischen den soundfont-player und die AudioContext-Destination geschaltet wird, umgesetzt.

Die anderen Parameter können über Slider eingestellt werden, die so festgelegten Werte werden direkt beim Spielen einer Note an den soundfont-player übergeben.

Die Webapplikation enthält 6 farbige Flächen, welche die 6 Lichtstrahler repräsentieren. Über die gespielten Noten kann der entsprechende Strahler ermittelt werden und das zugehörige Feld leuchtet auf. Die gewählte Notenlänge (duration) beeinflusst auch die Dauer des Aufleuchtens der Felder.

Die von der Bilderkennungs-Software gesendeten MIDI-Noten lösen in der Webapplikation einen Eventhandler aus, der auf den MIDI-Kanal horcht. Das empfangene MIDI-Signal wird überprüft. Die enthaltene Note wird zusammen mit den eingestellten Parametern an den soundfont-player übergeben, dieser erzeugt dann mit Hilfe des gewählten SoundFonts ein Audiosignal, das der Note auf dem gewählten Instrument entspricht. Das Audiosignal wird danach noch durch die Convolver-Node geleitet um den gewählten Reverb-Effekt über das Audiosignal zu legen.