

EveryDrum

Autoren: Tom Ziegler¹, Lukas Peschel¹
Betreuer: Dr. Sebastian Merchel¹

¹ Technische Universität Dresden

Erstellt am 9. Juli, 2018

Abstract

Drums | Android | Arduino | PureData |

Einleitung

Diese Projektarbeit entstand im Rahmen des Modules "Virtuelle Realität" im Sommersemester 2018. Die Zielstellung war die Entwicklung eines Systems, bestehend aus Hard- und Software, mithilfe dessen eine beliebige Oberfläche als Schlagzeug verwendet werden kann. Eine angeschlagene Position soll dafür in Echtzeit klassifiziert und einem der vorher erlernten Schlaginstrumente zugeordnet werden.

Die Vergabe der Projektthemen und erste Besprechung dieser erfolgte am 16.04.2018. Am 28.05. erfolgte eine Zwischenpräsentation, im Rahmen dieser die bisherigen Konzepte der verschiedenen Teams gegenseitig vorgestellt und diskutiert wurden. Die Abgabe und abschließende Präsentation der Ergebnisse erfolgte am 09.07.2018. Der Zeitraum der Bearbeitung erstreckte sich damit über 12 Wochen.

Im Rahmen einer ersten Konzeption wurden folgende Maßgaben für die Umsetzung aufgestellt:

- Das entstandene System soll portabel, einfach anzuwenden und wenig spezielle Hardware erfordern
- Um ein realistisches Spielgefühl zu erzeugen, soll die maximale Latenz zwischen Anschlag und Abspielen des Sounds 100ms betragen
- Das System soll robust funktionieren und dem Anwender ein Feedback über die Zuverlässigkeit der Erkennung liefern, sodass dieser eventuell Verbesserungsmaßnahmen durchführen kann (z.Bsp. Reduzierung von Hintergrundrauschen)

Um die Realisierung der Projektidee in diesem Zeitrahmen möglich zu machen, wurden zu Beginn einige Vereinbarungen getroffen:

- Als Oberfläche sollen vorerst nur Tische mit Holz- oder Kunststoffplatten verwendet werden
- Maximal sollen drei verschiedene Positionen unterschieden werden
- Im Sinne einer Prototypentwicklung sollen Funktionalität und Beweis der Konzeptplausibilität höhere Priorität als Benutzerfreundlichkeit haben
- In diesem Projekt soll die Eigenschwingung der angeschlagenen Oberfläche zur Auswertung genutzt werden.

Fragestellungen

- Welche Oberflächen eignen sich am besten?
- Wie groß muss das Zeitfenster für einen Schlag gewählt werden, um eine robuste Erkennung sowie eine geringe Latenz zu erreichen?
- Welche Informationen könne aus den Beschleunigungsdaten gewonnen werden, und welche eignen sich zur Echtzeit-Klassifizierung?
- Wie kann die Überlagerung zweier Anschläge verhindert/verringert werden?

1. Konzept

Ein naiver Ansatz wäre, mehrere Sensoren auf der Oberfläche zu verteilen und eine ein- oder zweidimensionale Lokalisation mittels Bestimmung der Laufzeitunterschiede durchzuführen. Dies setzt allerdings die Verwendung mindestens zweier (eindimensional, max. 3 Positionen) bzw. dreier (zweidimensional) Sensoren. Die Verwendung mehrerer Sensoren bedeutet jedoch auch steigende Anforderungen an die verarbeitende Hardware, höherer Realisierungsaufwand, höhere Anforderungen an den Anwender und nicht zuletzt höheren Hardwareaufwand und -abhängigkeit, was die Portabilität und Verwendbarkeit deutlich einschränkt.

Daher soll lediglich ein Sensor verwendet werden. Dies bedeutet jedoch, dass die Unterscheidung der Anschlagpositionen lediglich anhand der Schwingungscharakteristiken erfolgen muss.

Wird eine Tischplatte an einer Position angeschlagen, führt diese eine Eigenschwingung aus. Je nach Material, Lagerung, Objekte auf der Oberfläche, etc... unterscheiden sich verschiedene Tischplatten in ihrer Impulsantwort. Außerdem unterscheidet sich die ausgeführte Schwingung je nach Position und Art der Erregung: Ein Schlag mit den Fingerknochen erzeugt eine andere Schwingung als ein Schlag mit der flachen Hand - dies lässt sich leicht anhand des entstehenden Geräusches verifizieren. Ebenso erzeugt ein Schlag auf die Tischkante eine andere Schwingung als ein Schlag in die Tischmitte - dies zeigt sich vor allem in unterschiedlichen Nachhallzeiten.

Gewisse charakteristische Eigenschaften sind bauartbedingt durch den Tisch vorgegeben. Diese können vom Anwender nicht beeinflusst werden, folglich muss die Software in der Lage sein, diese Unterschiede in der Verarbeitung zu berücksichtigen. Trotzdem hat der Anwender gewisse Möglichkeiten, die Schwingung der Tischplatte zu beeinflussen, beispielsweise durch Anbringen von Massen, Dämpfern, etc... Dies kann und soll bewusst eingesetzt werden, um die Schwingungscharakteristiken der Anschlagpositionen möglichst unterschiedlich zu gestalten.

Um eine hohe Flexibilität in der Anwendung zu liefern, soll auf die Verwendung eines PCs/Laptops verzichtet werden. Stattdessen soll die softwareseitige Umsetzung mittels eines Smartphones erfolgen.

Zur Aufnahme der Schwingung wurden verschiedene Möglichkeiten erprobt:

- Verwendung eines Piezosensors
- Verwendung des internen Beschleunigungssensors des Smartphones
- Verwendung eines Arduino Mikrocontrollers mit Beschleunigungssensoren

2. Umsetzung

Für die Umsetzung wurde ein Android-Smartphone, ein Lenovo K6, mit der Android-Version 7.0 verwendet. Hingegen zu älteren Smartphones unterstützen neuere den USB On-The-Go (OTG) Modus, womit die Stromversorgung des Arduinos über USB-Anschluss gewährleistet wird.

Desweiteren wurde ein Arduino CH340 chip verwendet und als Beschleunigungssensoren das Modell MPU-6050. Die Android-App wurde in Java geschrieben. Zur Kommunikation mit dem Arduino und abgreifen der Beschleunigungssensoren des Sensors wurden die Bibliotheken *usb-serial-for-android* [mwm18] und *UsbSerial* [Her18] verwendet.

Die synthetisch erzeugten Schlagzeuggeräusche für Hihat, Snare und Bass wurden mit PureData [uIoEMA18] erzeugt. Zur Nutzung der PureData-Patches auf dem Android Gerät, wurde die Bibliothek *pd-for-android* [lu18] von *libpd* verwendet. Diese unterstützt keine erweiterten Funktionen, wie sie in PureData Extended vorkommen.

3. Arduino

Arduino Mikrocontroller

Für die Umsetzung wurde ein "Arduino Nano" Mikrocontroller mit Prozessor ATmega328P verwendet. Dieser führt selbst keine Datenverarbeitung aus, sondern bietet lediglich die Kommunikationsschnittstelle zwischen Smartphone und Sensoren. Die Kommunikation zum Smartphone erfolgt über die Serielle Schnittstelle, die Kommunikation zum Sensor über den I²C-Bus.

CH340

Bei dem Arduino Chip CH340 handelt es sich nach dem Datenblatt [htt18] um einen BUS Konvertierungs Chip, der eine serielle USB Schnittstelle ansprechen kann. Es unterstützt sowohl 5V, als auch 3.3V Spannungsquellen zum Betrieb. Die Baudraten reichen von 50bps bis zu 2Mbps.

MPU-6050

Nach dem Datenblatt [Inc] besitzt der MPU-6050 Sensor ein Gyroskop und ein Beschleunigungssensor mit X-, Y- und Z-Achsen. Der Beschleunigungssensor kann dabei verwendet werden, um Erschütterungen zu erkennen.

Signalanalyse

We use simple highpass filtering to sort out most of the noise and only let stronger vibrations from the surface pass the sensors...

To further smooth our data, we use boxfilter/ Gauss filter.

After basic filtering, we use ... to decide where the bang is coming from and what sound to play.

Instead of timbreID, which clusters and order features, we use ... [Bre]

DBSCAN is a density based cluster algorithm, which "cluster based on core and" In contrary to algorithm, such as K-Means, it has the advantage to detect arbitrary forms of cluster and filter out noise[EA13]

In fact, that we have much noise in our velocity data, we decided to use DBSCAN, which is easy to implement. To further increase accuracy and the unknown fact of epsilon, we further used ARCADE, which is a ... to [cite]

Using 3 sensors, we had results, while 2and 1 was unsatisfactory.

For 3 sensors we used barycentric interpolation, for 2 linear interpolation, for one we approximated by time/force?

Graph shows comparison between and the number of sensors used.

TODO

4. Diskussion

Literaturnachweis

- [Bre] William Brent. A timbre analysis and classification toolkit for pure data. University of California, San Diego, Center for Research in Computing and the Arts.
- [EA13] Mohammed T. H. Elbatta and Wesam M. Ashour. A dynamic method for discovering density varied clusters. In *International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 6, No. 1, February, 2013*, 2013.
- [Her18] Felipe (felHR85) Herranz. Usbserial. <https://github.com/felHR85/UsbSerial>, 2018.
- [htt18] <http://wch.cn/>. Usb to serial chip ch340. <https://www.olimex.com/Products/Breadboarding/BB-CH340T/resources/CH340DS1.PDF>, 2018.
- [Inc] TDK Inc. Mpu 6000 and mpu 6050 product specification revision 3.4. <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [lu18] libpd (<http://libpd.cc/>). pd-for-android. <https://github.com/libpd/pd-for-android>, 2018.
- [mwm18] mike w (mik3y). usb-serial-for-android. <https://github.com/mik3y/usb-serial-for-android>, 2018.
- [uloEMA18] IEM (<http://iem.at/>) Institute of Electronic Music and Acoustics. Pure data. <https://puredata.info/>, 2018.