

DrumEverywhere – Ein augmented reality drumkit mit Pure Data

Sebastian Merchel, Andreas Peetz, Silvio Ankeremann, Vincent Ridder, Ercan Altinsoy

Technische Universität Dresden, Lehrstuhl für Akustik und Haptik,

E-Mail: sebastian.merchel@tu-dresden.de

Einleitung

Dieser Beitrag diskutiert das Konzept und die Implementierung eines Pure Data Patches zur Erstellung eines AR-Schlagzeugs. Als Schlagfläche soll dabei ein beliebiges reelles Objekt dienen, z.B. eine Tischplatte, ein Kleiderständer oder eine Teekanne. Mit Hilfe eines Sensors (z.B. eines Piezos) wird ein Vibrationssignal am Objekt gemessen. Aus diesem sollen mit geringer Latenz durch Merkmalsextraktion und –klassifikation verschiedene Anschlagarten und/oder –positionen erkannt werden. Diesen können dann beliebige Klänge zugeordnet und über Kopfhörer oder Lautsprecher wiedergegeben werden. So lässt sich jeder physikalische Gegenstand in ein perkussives Musikinstrument verwandeln.

Setup

Als Sensor wurde ein handelsüblicher Tonabnehmer gewählt, welcher üblicherweise zur Ansteuerung von Stimmgeräten für akustischer Instrumente genutzt wird (Korg CM-300 BK). Dieses günstige Kontaktmikrofon besteht aus einem Piezosensor der mit Hilfe eines Klipps leicht an vielen Oberflächen montiert werden kann. Abbildung 1 zeigt den Sensor an einer Kaffeedose montiert. Das so aufgezeichnete Signal wird über eine Soundkarte zur Analyse an einen Rechner übertragen. Als Software-Framework wurde die quelloffene, grafische Programmierumgebung Pure Data [1] gewählt. Die Signalanalyse gliedert sich im Wesentlichen in drei Abschnitte: die Anschlagserkennung, die Merkmalsextraktion und die Klassifizierung. Weiterhin wird die Anschlagdynamik ausgewertet.



Abbildung 1: Klemmontage eines Piezosensors zur Aufnahme des Vibrationssignals, z.B. an einer Kaffeedose.

Anschlagerkennung

Ein üblicher Ansatz zur Anschlagserkennung ist die Auswertung der Einhüllenden eines Signals mit Hilfe eines Hüllkurvenfolgers (envelope follower). Überschreitet die Hüllkurve einen absoluten Schwellwert oder werden schnelle Anstiege beobachtet, wird ein Anschlag erkannt. Dieser Ansatz ist leider störanfällig, z.B. bei stark nachschwingenden Signalen. Verlässlichere Ergebnisse liefern Algorithmen welche die relativen Änderungen der Lautheit in separaten Frequenzbändern auswerten. Solch ein Algorithmus ist im Pure Data-Objekt `bonk~` implementiert [2]. Es erkennt Transienten in einem Audiosignal durch die Auswertung relativer Änderungen des Spektrums in 11 Frequenzbänder und soll in diesem Projekt zur Anschlagserkennung verwendet werden. Gleichzeitig liefert `bonk~` das grobe Leistungsspektrum sowie einen Intensitätswert zum Zeitpunkt des Anschlags. Damit kann bereits versucht werden unterschiedliche Anschlagssignale grob zu unterscheiden. Es erscheint jedoch sinnvoll weitere Klangmerkmale in die Klassifikation einzubinden.

Merkmalsextraktion

TimbreID ist ein Pure Data-Toolkit zur Audioanalyse und -klassifikation mit Fokus auf der Klangfarbe [3]. Es bietet eine Vielzahl von Pure Data-Objekten zur Berechnung von Merkmalen aus Audiosignalen. Es umfasst Analysen im Zeitbereich, wie z.B. die Zero Crossing Rate, Analysen im Frequenzbereich, z.B. Spectral Flatness, sowie rechenaufwendigere Analysen im Cepstralbereich. Hier sind besonders die Mel und Bark Frequency Cepstral Coefficients (BFCC) zu nennen. Diese Merkmale sind beliebig kombinierbar und können in einem Vektor an einen Klassifizierer übergeben werden.

Klassifizierung

Zur Klassifizierung kommt ein k-Nearest-Neighbor-Algorithmus (KNN) zum Einsatz, der vergleicht, welche der Klassen dem zu klassifizierenden Vektor im Merkmalsraum am nächsten liegen. Zwischen dem eingehenden Vektor und allen Klassen wird entweder eine euklidische Distanz oder eine Manhattan-Distanz (Taxicab) berechnet. Um die Trainings- Merkmalsvektoren in Klassen zu gruppieren, benutzt timbreID ein agglomeratives Clusterverfahren, dem als Parameter die Anzahl der vorhandenen Klassen übergeben wird. Bei diesem Verfahren bildet anfänglich jeder Vektor ein eigenes Cluster. Danach werden die Vektoren stückweise nach Ähnlichkeit zusammengefasst, bis nur noch die gewünschte Anzahl an Clustern vorhanden ist. TimbreID wurde bereits erfolgreich für die Klassifizierung von zwölf verschiedenen Schlaginstrumenten eingesetzt [4].

Implementierung

Die hier vorgestellte Implementierung basiert auf einem Beispiel von William Brent [3]. Der Prototyp verfügt über eine einfache grafische Benutzeroberfläche mit Kurzanleitung. Der Nutzer kann zwischen der Ausgabe von Audiosamples und MIDI-Signalen entscheiden. Der Prototyp besteht aus zwei PureData-Patches: Dem Main-Patch, der die Berechnungen enthält, und dem Control-Patch mit der Bedienoberfläche.

Der Programmablauf lässt sich aufgrund der Datenflussstruktur von PureData gut anhand von Abbildung 2 nachvollziehen. Ein Anschlag im eingehenden Audiosignal wird zunächst mit Hilfe von `bonk~` detektiert. Das resultierende Triggersignal löst die Extraktion von Merkmalen mit `bfcc~` (Bark Frequency Cepstral Coefficients) aus. Hier kann die auszuwertende Fensterlänge variiert werden. Längere Fenster verbessern die Erkennungsrate, vergrößern aber auch die notwendige Verzögerung. Im Beispiel ist eine Fenstergröße von 1024 Samples zu sehen. Dies entspricht einer Fenstergröße von 23 ms bei einer Abtastrate von 44,1 kHz. Insgesamt wurde damit eine Verzögerung zwischen Anschlag und Audioausgabe von etwa 40 ms erreicht. Dies ist für schnelle Schlagfolgen noch zu lang. An dieser Stelle können beliebige Merkmale ausgewählt und entsprechende Berechnungsalgorithmen eingefügt werden. Erlaubt ist was für den jeweiligen Anwendungsfall gut funktioniert. Bei ersten Versuchen mit Anschlägen auf einer Tischplatte konnten bei hinreichender Fenstergröße gute Ergebnisse mit Erkennungsraten von bis zu 94% unter Verwendung der Bark Frequency Cepstral Coefficients (`bfcc~`) erreicht werden. Die Kombination anderer „low level“, d.h. nicht-cepstraler Merkmale, erzielte Erkennungsraten bis zu 70%.

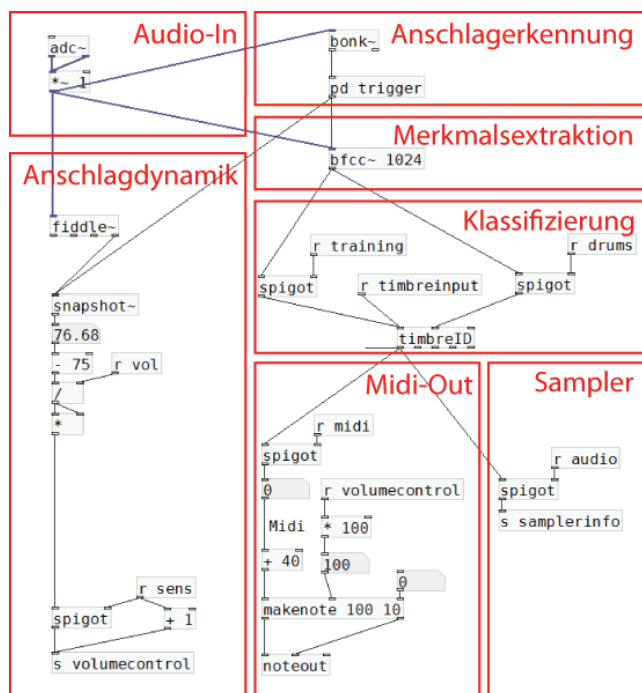


Abbildung 1: Hauptpatch des PureData-Programms zur Erstellung eines AR-Schlagzeuges.

Die Klassifizierung mit `timbreID~` ist ebenfalls in Abbildung 2 erkennbar. Das detektierte Cluster wird als Zahl ausgegeben und triggert entweder die Wiedergabe eines Audiosamples oder die Ausgabe einer Midi-Note, z.B. zur Verwendung eines externen Synthesizers. Gleichzeitig wird durch den Anschlagstrigger die Erkennung der Anschlagsdynamik mit Hilfe des Objektes `fiddle~` ausgelöst. Der durch `snapshot~` gepufferte Wert kann anschließend zur Manipulation der Signalamplitude verwendet werden.

Abbildung 3 zeigt das Kontrollpatch mit dessen Hilfe zwischen den verschiedenen Modi zur Aufzeichnung von Trainingsamples, Clustern und Wiedergabe von Drumsounds umgeschaltet werden kann. Ein einfacher Audiomixer steht ebenfalls zur Verfügung.

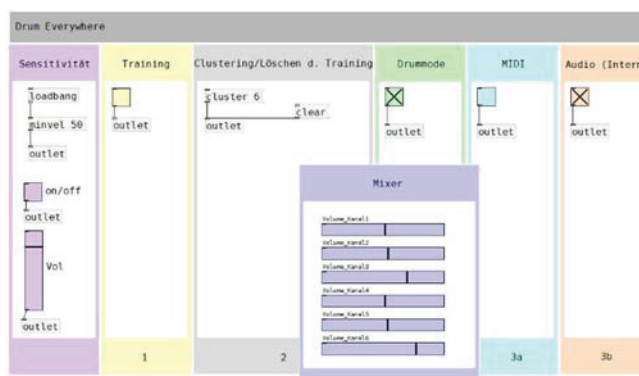


Abbildung 3: Kontrollpatch mit einfacher Benutzeroberfläche und Audiomixer.

Zusammenfassung

Mit Hilfe des beschriebenen Frameworks können verschiedene Merkmale und Merkmalskombinationen ausgewählt und hinsichtlich der Erkennungsrate für einen konkreten Anwendungsfall verglichen werden. Im Rahmen des Projektes wurde ein Prototyp implementiert und getestet. Die Gesamtverzögerung der derzeitigen Implementierung ist für schnelle Schlagfolgen jedoch bisher zu lang so dass hier Optimierungsbedarf besteht. Zum einen könnten weitere oder andere Merkmale extrahiert werden, zum anderen kann der angewendete Klassifizierungsalgorithmus ersetzt werden. Ein weiterer offener Punkt ist das gleichzeitige Erkennen mehrerer Anschlagorte was mit der derzeitigen Implementierung noch nicht möglich ist.

Literatur

- [1] Miller Puckette, "Pure Data," in *International Computer Music Conference*, 1997.
- [2] M. Puckette, T. Apel, and D. Zicarelli, "Real-time audio analysis tools for Pd and MSP," in *International Computer Music Conference*, 1998.
- [3] W. Brent, "A timbre analysis and classification toolkit for Pure Data," in *International Computer Music Conference*, 2010.
- [4] W. Brent, "Cepstral analysis tools for percussive timbre identification," in *3rd International Pure Data Convention*, 2009.