

## Android-Based Real-Time Audio Signal Processing

Durch den massiven Anstieg der Rechenleistung in den letzten Jahren liegt die Echtzeit-Audioverarbeitung nun im Bereich der Möglichkeiten der meisten Handheld-Geräte für Endanwender. Komplexe Signalmanipulationen wie z. B. Time-Stretching und Pitch-Shifting testen jedoch immer noch die Grenzen moderner Verarbeitungseinheiten. Wenn solche Algorithmen für Echtzeitanwendungen implementiert werden, geschieht dies aus Effizienzgründen meist in Low-Level-Programmiersprachen. Infolgedessen fehlt es an echtzeitfähigen Bibliotheken, die eine einfache Integration oder Erweiterung in High-Level-Programmiersprachen ermöglichen. Diese Arbeit konzentriert sich auf die Entwicklung und Implementierung eines hochwertigen Pitch-Shifting-Algorithmus, der eine gute Echtzeit-Performance auf Android-Handheld-Geräten zeigt.

Da es keinen Goldstandard für die Entwicklung von Pitch-Shifting-Algorithmen gibt, bestand der erste Schritt in einer gründlichen Literaturrecherche über die besten verfügbaren Time-Stretching und Pitch-Shifting-Algorithmen. Anschliessend wurden die besten Algorithmen implementiert und anhand von Klangqualität und Komplexität bewertet. In einem nächsten Schritt wurde untersucht, wie gut sich diese Algorithmen für Echtzeitanwendungen auf einem Android-Smartphone eignen.

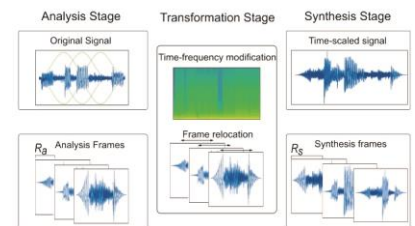
Aufgrund ihrer Prominenz und der Menge an verfügbaren Informationen haben wir uns entschieden, uns auf Block-Wise Phase-Vocoder-Algorithmen zu konzentrieren. Während der Fokus dieser Arbeit auf Pitch-Shifting lag, können die Hauptideen leicht auf Time-Stretching angepasst werden. Für eine eingehende Analyse wurden die Algorithmen zunächst in Python implementiert. Die vielversprechendsten Algorithmen wurden dann in Java implementiert und auf verschiedenen Android-Geräten getestet. Die Kombination eines Phase-Locked Vocoders mit Transienten Erkennung lieferten hierfür die besten Ergebnisse.

Die Endergebnisse der Arbeit sind eine Python-Anwendung mit fünf auf Phase-Vocoder basierenden Pitch-Shifting-Algorithmen, eine Android-App mit den drei vielversprechendsten Algorithmen und eine Wissensdatenbank, die helfen soll, Menschen in das Gebiet der Time-Frequency-Audioverarbeitung einzuführen. Unsere beiden besten Implementierungen können in Echtzeit auf einem Android-Smartphone ausgeführt werden und erreichen dort die Qualität der Rubber Band Library. Der gesamte Quellcode, der für diese Arbeit erstellt wurde, ist mit der Open-Source-Software-Lizenz (OSS) auf GitHub öffentlich verfügbar.

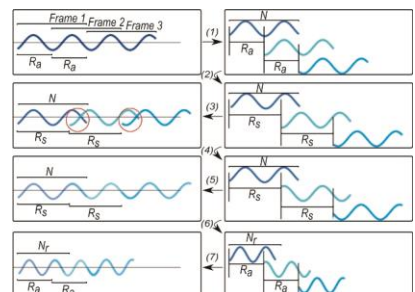


Diplomierende  
Alex Shasa Lopez  
Simon Elias Schneider

Dozierende  
Thomas Müller  
Martin Loeser



Die drei Stufen, die jeder Time-Frequency-Audioverarbeitungsalgorithmus implementiert. Zwischen den Algorithmen ändert sich nur die Transformationsstufe, welche definiert, ob er eine Modifikation der Time-Domain oder Time-Frequency-Domain durchführt.



Visualisierung verschiedener TSM-Algorithmen in der Time-Domain. 1) Signal framing; 2) Signal stretching; 3) Time-stretched OLA; 4) Phase alignment; 5) Phase-aligned, time-stretched OLA; 6) Resampling; 7)