



Compressed Compute-and-Forward mit korrelierten Audiosignalen

Lucas Weber

Geboren am: 10.01.1995

Matrikelnummer: xxxxx

Immatrikulationsjahr: 2013

Raphael Hildebrand

Geboren am: 01.08.1994

Matrikelnummer: xxxxxx

Immatrikulationsjahr: 2013

Florian Roth

Geboren am: 13.01.1994

Matrikelnummer: xxxxx

Immatrikulationsjahr: 2013

Orell Garten

Geboren am: 09.07.1993

Matrikelnummer: 3948375

Immatrikulationsjahr: 2013

13.07.2016

Betreuer

Carsten Hermann

Betreuender Hochschullehrer

Dr.-Ing. Rico Radeke

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit mit dem Titel *Compressed Compute-and-Forward mit korrelierten Audiosignalen* selbstständig und ohne unzulässige Hilfe Dritter verfasst habe. Es wurden keine anderen als die in der Arbeit angegebenen Hilfsmittel und Quellen benutzt. Die wörtlichen und sinngemäß übernommenen Zitate habe ich als solche kenntlich gemacht. Es waren keine weiteren Personen an der geistigen Herstellung der vorliegenden Arbeit beteiligt. Mir ist bekannt, dass die Nichteinhaltung dieser Erklärung zum nachträglichen Entzug des Hochschulabschlusses führen kann.

Dresden, 13.07.2016

Lucas Weber

Raphael Hildebrand

Florian Roth

Orell Garten

Inhaltsverzeichnis

1	Aufgabenstellung	4
2	Motivation	4
3	Theoretische Vorbetrachtung	4
3.1	Korrelationsfunktion	4
3.2	Parameter	4
3.3	Gauß-Regression	4
4	Verfügbare Technik	4
4.1	Software	4
4.2	Hardware	4
5	Programm	5
5.1	Aufbau des Programms	5
5.2	Mögliche Einstellungen	5
5.3	Probleme	6
6	Signalauswahl	6
6.1	Beispielsignal	6
6.2	Probleme bei der Signalauswahl	6
7	Zusammenfassung	6
8	Zusammenfassung	7

1 Aufgabenstellung

Es ist ein Programm zu entwerfen, welches Audiodateien einliest und dateiweise die beiden Stereo-Kanäle miteinander korreliert. Es sollen Maßzahlen entworfen und berechnet werden, die wesentliche charakteristische Eigenschaften der Korrelationsfunktion, insbesondere den Anteil dominanter Komponenten und deren Abklingverhalten, widerspiegeln. Dafür sind Audiosignale aufzunehmen, bezüglich der verwendeten Maße zu klassifizieren und entsprechend ihrer Klassifizierung systematisch abzuspeichern.

2 Motivation

Im Jahr 2022 werden 500 Milliarden internetfähige Geräte erwartet, die mit einander kommunizieren sollen. Das führt zu extrem hohen Datenmengen, die in kürzestmöglicher Zeit von A nach B transportiert werden müssen. Große Herausforderungen bestehen darin, dass man sehr kurze Verzögerungszeiten und eine hohe Widerstandsfähigkeit garantieren muss. Idealerweise benötigen die Geräte wenig Energie. Ein Ansatz zur Lösung dieses Problems ist die Netzwerkcodierung.

In bestimmten Szenarien ist eine große Anzahl an Geräten mit Sensorik zu Erfassung der Umgebung mit hohen Anforderungen an die Netzwerkkapazität zur Übertragung der erfassten Daten verbunden. Es stellt sich die Frage, wie viele Sensoren für eine ausreichend genaue Abbildung benötigt werden. Da die Quellen teilweise korrelierte Datenströme erzeugen, lässt sich die zu übertragende Gesamtdatenmenge reduzieren, was durch eine geeignete Kombination von Netzwerkcodierung mit Methoden des Compressed Sensing erreicht werden soll.

3 Theoretische Vorbetrachtung

3.1 Korrelationsfunktion

3.2 Parameter

3.3 Gauß-Regression

4 Verfügbare Technik

4.1 Software

Softwareseitig haben wir Octave benutzt. Als freie Alternative zu Matlab vereint Octave gute Performance, syntaktische Gleichheit zu Matlab, sowie kostenfreie Benutzung unter einem Dach. Im Vergleich mit Python haben wir festgestellt, dass die Geschwindigkeit aufwendiger Rechnungen, wie der Korrelation, bei Python schlechter ist. Somit haben wir uns für Octave entschieden. Um unseres selbstgeschriebenes Programm zu verifizieren haben wir eine Autokorrelation durchgeführt und diese mit der von Audacity berechneten AKF verglichen. Wir sind dabei zu dem Ergebnis gekommen, dass unser Programm funktioniert.

4.2 Hardware

Es wurden zwei hochwertige Kondensatormikrofone vom Telekom-Lehrstuhl der TU Dresden zur Verfügung gestellt. Außerdem konnten wir ein Stativ, eine Stereoschiene sowie eine USB-Soundkarte verwenden.

5 Programm

5.1 Aufbau des Programms

Das Programm ist in 3 große Teile gegliedert. Dazu zählt das Einlesen der Audiodateien, die benötigte Signalverarbeitung inklusiver Berechnung der gewünschten Parameter und das Speichern der gewonnenen Werte in Form einer Excel-Datei.

Einlesen der Audiodaten Die Audiodateien liegen im WAV-Format als Stereoaufnahme vor. Zunächst wird eine Liste mit allen Dateien in einem bestimmten Ordner erstellt, damit die Dateien nacheinander eingelesen werden können. Im nächsten Schritt werden die beiden Kanäle voneinander getrennt, um diese dann in die Signalverarbeitung zu übergeben.

Signalverarbeitung Das Kernstück der Signalverarbeitung ist eine periodische Korrelationsfunktion, die den linken und rechten Kanal miteinander korreliert. Die dabei entstandene Korrelationsfunktion wird dann weiter untersucht. Als nächstes wird eine Art Einhüllende berechnet, die ein Maß für die Steilheit der Kurve ist. Wie bereits im Abschnitt "Gauß-Regression" beschrieben, wird dann mit Hilfe der Methode der kleinsten Quadrate eine Gauß-Glocke so angepasst, dass sie den Verlauf der Hüllkurve der KKF möglichst gut abbildet. Die Parameter ripple, σ , Gleichanteil und Zeitverschiebung des Maximums aus dem Ursprung werden danach an eine Funktion übergeben, die diese Daten in einer Excel-Tabelle speichert.

Speicherung Die Speicherung der Daten erfolgt in einer Excel-Datei. Dabei wird zu erst der Dateiname des Samples und alles dazugehörigen Werte gespeichert. Außerdem wird noch ein Link zum Graphen der Korrelationsfunktion angegeben, damit man sich diese bei der Auswahl der Test-Signale anschauen kann.

5.2 Mögliche Einstellungen

Im Code gibt es diverse Einstellungen, die das Verhalten des Programms je nach Wunsch des Anwenders verändern. Diese sind am Beginn der main.m-Datei festgelegt und werden im folgenden beschrieben:

- *path* - Pfad zur Sammlung der WAV-Dateien
- *excel_path* - Pfad unter dem Excel-Datei mit Lösungen gespeichert wird
- *output* - Unterscheidung, ob Ergebnisse gespeichert oder angezeigt werden
- *calc* - Wechsel zwischen Berechnung im Zeit- und Frequenzbereich möglich.
- *x_axes* - Unterscheidung, ob Korrelation gegen Samples oder Zeit aufgetragen wird
- *priority* - Unterscheidung, ob angegebene Blocklänge oder Zeitdauer priorisiert wird
- *t_start* - Startzeitpunkt der Korrelation
- *t_dur* - Array *A1* mit Menge an Zeitdauern die korrelierten werden sollen, Korrelation beginnt immer bei *t_start*
- *Ncor_init* - Array, mit identischer Länge zu *A1*. Gibt an wie oft korrespondierender Eintrag in *A1* hintereinander korreliert wird.
- *Lcor* - Blocklänge der Korrelation

5.3 Probleme

6 Signalauswahl

Wir haben für die Aufnahme der Signale viel verschiedene Situationen ausgesucht, um ein möglichst breites Spektrum an Raum-Effekten zu erhalten. Aufnahmeorte waren beispielsweise der Platz vor dem HSZ, die Wiese zwischen Physik- und Mathematikgebäude, sowie der Trefftbau. Außerdem wurde in einer Wohnung gemessen, um Effekte von schallabsorbierenden Stoffen wie Teppich oder Bett zu erhalten. Soweit möglich, haben wir die natürliche Geräuschkulisse am jeweiligen Ort eingefangen. Zusätzlich dazu wurde ein definiertes Signal mittels eines Lautsprechers erzeugt, um Direktschall zu nutzen. Bei diesen Aufnahmen sollten die Effekte des Raumes am deutlichsten hervortreten.

6.1 Beispielsignal

6.2 Probleme bei der Signalauswahl

7 Zusammenfassung

8 Zusammenfassung