

Sprachsynthese und -erkennung Übung 4

LPC-Vocoder/Sprachkompression

4.1 Einleitung

In dieser Übung beschäftigen wir uns mit der Modell der Spracherzeugung nach dem Quelle-Filter-Prinzip und seiner Nutzung bei der Sprachcodierung und -kompression. Bei diesem Modell wird das Sprachsignal in Anteile zerlegt, die von den Stimmbändern als Oszillator (bei stimmhaften Lauten) oder einer rauschhaften Anregung (bei stimmlosen Lauten) herrühren, d.h. der "Quelle", und solche, die aus der akustischen Filterung der "Quelle" durch den Vokaltrakt entstehen, dem "Filter". Mit Hilfe der sogenannten Linearen Prädiktion (LPC) lässt sich das Sprachsignal in der beschriebenen Weise in Quelle und Filter aufspalten. Die Software PRAAT stellt den LPC-Algorithmus zur Verfügung, den wir für diese Übung nutzen wollen. Das Filter, das den Vokaltrakt beschreibt, ist ein digitales Filter, wie wir es in der Vorlesung besprochen haben und wird durch seine Ordnung gekennzeichnet, d.h. die Anzahl seiner Verzögerungsglieder.

4.2 Vorbereitungen

Als erstes lesen wir mit [Read] in PRAAT unsere Sprachdatei ein. Dann wählen wir "Formants&LPC->To LPC (autocorrelation)". Die anderen Verfahren, die PRAAT noch anbietet, unterscheiden sich nur geringfügig in der Methode der Berechnung der Filter-Koeffizienten. Im Einstelldialog sind besonders die Punkte "Prediction Order" (Ordnung des Filters) und "Analysis width" (Framebreite bei der Analyse) von Interesse. Diese wollen wir in dieser Aufgabe variieren. Zusätzlich zu den von PRAAT angebotenen Standardwerten von Order=16/Analysis Width=25ms (kurz: 16/25) werden wir Analysen bei Ordnungen von 10, 6 und 1 durchführen, sowie Framelängen von 50 und 100 ms untersuchen. "Time step" (die "Vorschubbreite" der Analyse) setzen wir auf denselben Wert wie "Analysis width", d.h. wir erlauben kein Überlappen von benachbarten Frames.

Um die verschiedenen Varianten nicht zu verwechseln, geben wir den einzelnen LPC-Objekten mit 'Rename' entsprechende Namen.

Nachdem die verschiedenen LPC-Objekte kreiert wurden, führen wir zunächst eine inverse Filterung des Sprachsignals durch, mit der wir theoretisch die "Quelle" zurückgewinnen (Sound und LPC (16/25) gemeinsam anwählen, "Filter (inverse)"). Das entstehende Signal, das auch als "Residual"-Signal bezeichnet wird, betrachten wir im Edit-Fenster. Wie unterscheiden sich die Spektren vom Original, wie ist die Restverständlichkeit der ursprünglichen Sprache? (Spektrogramme von Original und Residual-Signal ins Protokoll) Nun gewinnen wir das ursprüngliche Signal wieder zurück (Rekonstruktion), indem wir das Residuum mit dem dazugehörigen LPC-Objekt (16/25) filtern (Residuum und LPC gemeinsam anwählen, "Filter", "Use LPC Gain" nicht ankreuzen!).

4.3 Variation des Filters

Das Residual-Signal dient uns jetzt zum Testen der anderen LPC-Objekte mit den verschlechterten Parametern. Auch hier wählen wir das Residuum und das gewünschte LPC-Objekt an und filtern wieder. Vergleiche die Qualität der einzelnen Varianten mit dem Original! Wie unterscheiden sich die Beispiele, bei denen die Filter-Ordnung variiert wurde, wie solche, bei denen die Framebreite variiert wurde? Bei welcher Kombination ist die Sprachverständlichkeit nicht mehr akzeptabel?

Für die 12 Qualitäten (also Filter-Ordnungen 16, 10, 6, 1 kombiniert mit Framebreiten 25, 50 und 100ms) berechnen wir die Datenraten. Dazu machen wir uns Folgendes klar: Übertragen werden müssen mindestens je Analyseblock (also z.B. je 25ms) die Koeffizienten (Zahl entspricht der Filterordnung) und die Grundfrequenz f_0 . Setze hier jeweils pro Zahlenwert 16bit an. Vergleiche diese Datenrate mit der ursprünglichen PCM-Datenrate der Sprachdatei.

4.4 Variation der Quelle

Nun wollen wir die Anregung (Quelle) verändern. Dafür habe ich zwei Signale bereitgestellt, weißes Rauschen und einen 100Hz-Pulse Train, einen obertonreiches Signal mit Grundfrequenz 100Hz

- [Weißes Rauschen](#)
- [Pulse Train](#)

Wenn wir das Rauschen mit dem LPC-Objekt filtern (jetzt "Use LPC Gain" anwählen), bekommen wir so etwas wie Flüstersprache, beim Pulse Train eine Roboterstimme. Abschließend werden wir auch noch eine Musikdatei als Anregung benutzen, was einen Vocoder-effekt bewirken sollte. Am besten eignen sich hier flächige, eher statische Sounds wie Synthesizerakkorde, da der Rhythmus vom Sprachsignal kommen soll.

Ins Protokoll: (1) Wave-Dateien: Original, Residual-Signal, rekonstruierte Signale mit Ordnung 16/10/6/1 sowie Analysis Width=25/50/100ms und Rekonstruktionssignal mit manipulierter Anregung ("Pulse Train", "Noise", "Vocoder"), Qualitätsvergleich der einzelnen Varianten, Datenraten und Spektrogramme.