

## Aufgabe 1: Filegesteuerte Spektralanalyse

Im Anhang sehen Sie in Abbildung 1 die Gesamtstruktur des Spracherkenners, der in diesem Semester programmiert werden soll.

Darin dient die Funktion `computeFeatures` zur Berechnung der Sprachmerkmale aus gegebenen Audiodaten. Um diese Funktion flexibel zu gestalten, bekommt sie als letzten Parameter ein Konfigurationsfile im HTK-Parameterformat. In dieser Übung soll diese Funktion, die entsprechend des Konfigurationsfiles verschiedene Sprachfeatures berechnen wird, zunächst für die Berechnung von Spektrogrammen implementiert werden.

### Hauptfunktion

**1.1** Schreiben Sie ein Hauptskript zum Testen, `uebung1.m`, das die Audiodatei `13zz637a.wav` einliest und

- den Verlauf des Signals über die Zeit plottet,
- die Abtastrate des Signals und die größte auftretende Amplitude ermittelt und
- das Signal über Kopfhörer ausgibt.

In diesem Skript können Sie anschließend alle weiteren Funktionen dieser Übung aufrufen und testen.

Hilfreiche Funktionen: `help`, `audioread`, `plot`, `linspace`, `length`, `soundsc`

### Konfigurationsfile

Gegeben ist eine Datei `configSPEC.con` im HTK-Parameterformat:

```
# coding parameters 'configSPEC' by D.K.
TARGETKIND = SPEC
TARGETRATE = 100000.0
SAVECOMPRESSED = F
SOURCEFORMAT = WAV
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
ENORMALISE = T
```

**1.2** Schreiben Sie eine Funktion

```
function out = getConfig(filename)
```

die eine Datei in der oben gezeigten Form in eine Variable der Struktur

```
out.preemcoef = 0.97 % als Zahl  
out.usehamming = 'T' % als char  
...
```

einliest. **Achtung:** die Reihenfolge der Einträge ist beliebig und die Konfigurationsparameter können von Config-File zu Config-File unterschiedlich sein. Zeiten werden im HTK-Format in 100ns-Einheiten dargestellt. Wandeln Sie deswegen `WINDOWSIZE` und `TARGETRATE` in Sekunden um.

Tipps: Wenn `str = 'feldname1'`, dann erzeugt `g.(str) = xyz` ein Strukturelement `g.feldname1` mit dem Wert `xyz`.

Hilfreiche Funktionen: `fopen`, `fgetl`, `strfind`, `str2num`

## Fensterung

**1.3** Schreiben Sie eine Funktion

```
function windowed = makeframes(audiodata, fs, parameters)
```

die eine Fensterung von Audiodaten entsprechend den Parametern `TARGETRATE`, `WINDOWSIZE` und `USEHAMMING` durchführt. Die verwendete Fenstergröße in Samples sollte die nächsthöhere Zweierpotenz über `WINDOWSIZE` sein. Das Verfahren ist in Abbildung 2 grafisch dargestellt. Berücksichtigen Sie bitte bei dieser Berechnung die Abtastfrequenz `fs` der Audiodaten. Ein Überblick über die verwendeten Parameter findet sich in Tabelle 1.

Tipps: Bei der Fensterung eines zeitdiskreten Signals mit  $Q$  Abtastwerten werden bei einer Fensterlänge  $L$  und einer Überlappung  $O$  mindestens

$$K = \left\lceil \frac{Q - O}{L - O} \right\rceil$$

Fenster benötigt. Darüber hinaus werden die folgenden Aufgaben einfacher, wenn Sie die Ergebnisse in einer Matrix speichern.

Hilfreiche Funktionen: `hamming`, `hanning`, `nextpow2`

---

WINDOWSIZE	Länge des Fensters in Sekunden
TARGETRATE	Zeitverschiebung zwischen zwei Fenstern in Sekunden
USEHAMMING	'F': Hanning-Fenster benutzen, sonst das Hamming-Fenster

---

Tabelle 1: Bedeutung der Konfigurationsparameter.

## Fouriertransformation

- 1.4** Transformieren Sie die gefensterten Daten mit dem Befehl `fft` in den Frequenzbereich. Stellen Sie das Ergebnis mittels `imagesc(log(abs(...)))` dar.
- 1.5** Wenn ein Signal mit der Abtastfrequenz `fs` via FFT der Länge `nfft` in den Frequenzbereich transformiert wird, welches sind dann die nicht-redundanten Stützstellen im Frequenzbereich? Schreiben Sie eine Funktion

```
function freqs = linfoqs(fs, nfft)
```

die diese Frequenzstützstellen aus der Abtastfrequenz `fs` und der FFT-Länge `nfft` ermittelt.

## Erste Feature Extraction

- 1.6** Schreiben Sie eine Funktion

```
function features = computeFeatures(audiodata, fs, configfile)
```

die unter Verwendung der bisher implementierten Funktionen den nicht-redundanten Teil des Spektrums der Audiodaten zurückliefert. Überprüfen Sie die Aufrufstruktur, indem Sie das Skript `testUebung1.m` aufrufen.

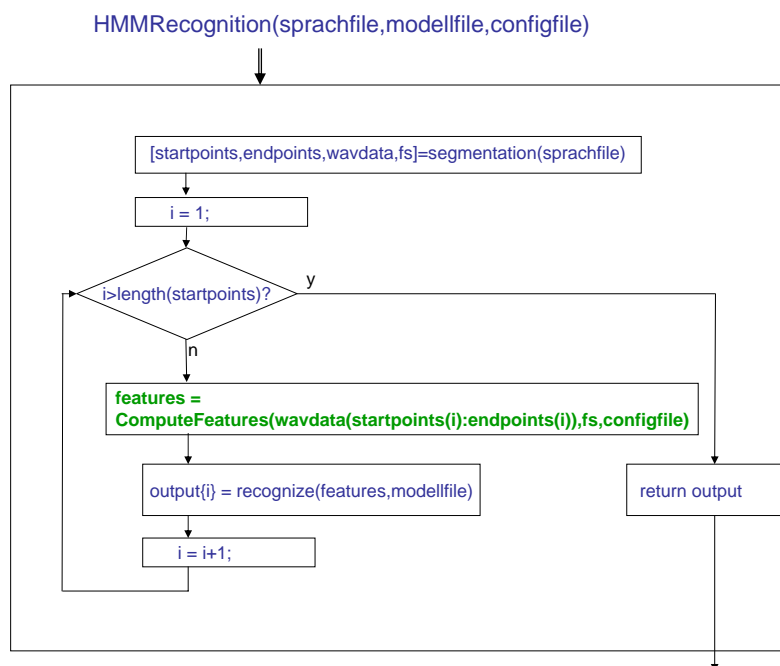


Abbildung 1: Gesamtstruktur des Spracherkenners.

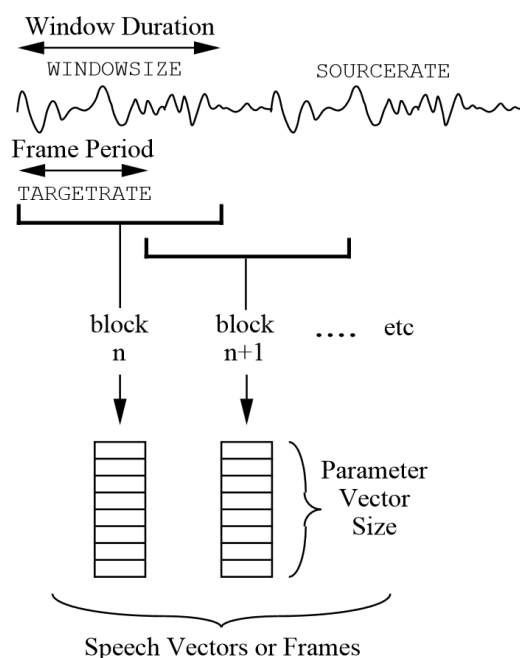


Abbildung 2: Bedeutung der Parameter in der Berechnung überlappender Sprach-Frames, aus: HTK-Book, Version 3.2.1, Cambridge University, 2002