

音声データの基本操作

川村 新

1-1 : wav 音声ファイルの読み込み

収録フォルダ : 44_WAV_reading

リスト1 wavファイルの読み込みプログラムDD_wav_reading.c (抜粋)

```
(a) wavファイル読み込み用変数の宣言部
unsigned short tmp1; // 2バイト変数
unsigned long tmp2; // 4バイト変数

(b) wavヘッダ・ファイルの表示部
printf("Wave data is\n");
fseek(f1, 22L, SEEK_SET); // チャンネル情報位置に移動
fread (&tmp1, sizeof(unsigned short), 1, f1);
// チャンネル情報読み込み 2バイト
ch=tmp1; // 入力チャンネル数の記録
fread (&tmp2, sizeof(unsigned long), 1, f1);
// サンプリング周波数の読み込み 4バイト
Fs = tmp2; // サンプリング周波数の記録
fseek(f1, 40L, SEEK_SET); // サンプル数情報位置に移動
fread (&tmp2, sizeof(unsigned long), 1, f1);
// データのサンプル数取得 4バイト
len=tmp2/2/ch; // 音声の長さの記録 (2バイトで1サンプル)

printf("Channel = %d ch\n", ch);
// 入力チャンネル数の表示
printf("Sample rate = %d Hz \n", Fs);
// 入力サンプリング周波数の表示
printf("Sample number = %d\n", len);
// 入力信号の長さの表示
printf("\nPush any key\n");
getchar();
fseek(f1, 44L, SEEK_SET); // 音声データ開始位置に移動

(c) メイン・ループ内 Signal Processing 部
y[t]=s[t]; // 出力=入力
printf("16bit= %d\t Normalized= %f\n",input,s[t]);
// 観測信号を表示
```

```
C:\data\2015_sp100\44_WAV_reading>DD_wav_reading speech.wav
Wave data is
Channel = 1 ch
Sample rate = 16000 Hz
Sample number = 33000
Push any key
16bit= 18 Normalized= -0.00649
16bit= -1 Normalized= -0.00031
16bit= 7 Normalized= 0.000214
16bit= 18 Normalized= 0.000549
16bit= -13 Normalized= -0.000397
16bit= 2 Normalized= 0.000061
16bit= 4 Normalized= 0.000122
16bit= 21 Normalized= 0.000641
16bit= 26 Normalized= 0.000793
16bit= 29 Normalized= 0.000885
16bit= 15 Normalized= 0.000458
C:\data\2015_sp100\44_WAV_reading>
```

図1 wavファイルの読み込みプログラムの実行

● プログラム

wavファイルを読み込むプログラムをリスト1に示します。

wavファイルのヘッダは、2バイトで書かれている部分と4バイトで書かれている部分があります。このため、読み込み用の変数として、2バイト用のtmp1と、4バイト用のtmp2を準備しています。

ヘッダ情報の場所は規格で決まっているので、fseek関数で該当箇所移动到し、データを読み込んで表示します。後半のデータ部も開始場所が決まっているので、そこから読み出しを開始します。

今回は16ビット・データが並んでいるという前提で読み出しを行っています。

● 実行結果

実行結果を図1に示します。ヘッダ情報として、1チャンネルの音声で、サンプリング周波数が16000Hz、サンプル数が33000であることが表示されています。よって、音声信号は、 $33000/16000 = 2.0625$ 秒であることが分かります。

続いて、各データについて、16ビット整数値(-32768~32767)と正規化した数(-1~1)の両方が表示されています。

音声処理でよく利用されるwavファイルを読み込み、テキスト・データとして表示するプログラムです。wavファイルを扱うための基本処理になります。

● 仕組み

wavファイルは、最初にサンプリング周波数やデータの数などのヘッダ情報が書かれています。その次に、音のデータが並んでいます。これらはもちろん全てバイナリ・データ(0か1)で表現されています。

ここでは、ヘッダ情報を読み出して、主要部分を表示します。次に、バイナリ・データで書かれたデータを読み出し、テキスト・データとして表示します。