THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Re-sample and Re-quantize Audio Signals with Matlab

How to load an audio file?

To read in an audio file, the following function can be used:

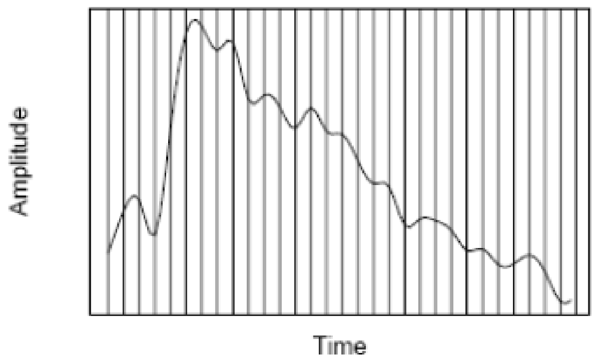
[*data*, *sampleRate*, *nBits*] = wavread(filename, fmt)

- filename: The filename input is a string enclosed in single quotes;
- fmt: specifies the data type format of *data* used to represent samples read from the file;
- data: contains the returned sampled data from the sound file; its value represents the volume (or intensity) of the sound signal. Data is a vector (or a matrix).
- sampleRate: represent the sample rate in Hertz;
- nBits: represents the number of bits per sample used to encode the data in the file.

How to load an audio file? (cont.)

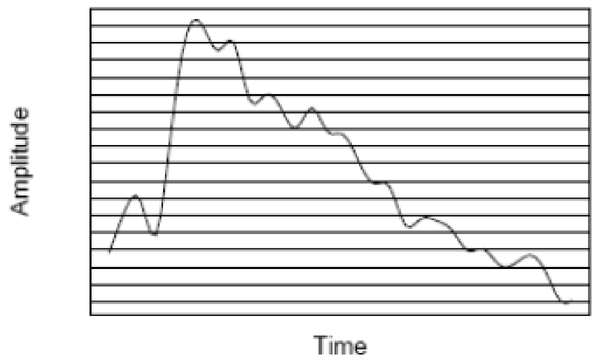
Pulse Code Modulation (PCM)

- The basic techniques for creating digital signals from analog signals are **sampling** and **quantization**.
- Quantization consists of selecting breakpoints in magnitude, and then remapping any value within an interval to one of the representative output levels.



Sampling

sampleRate



Quantization

nBits

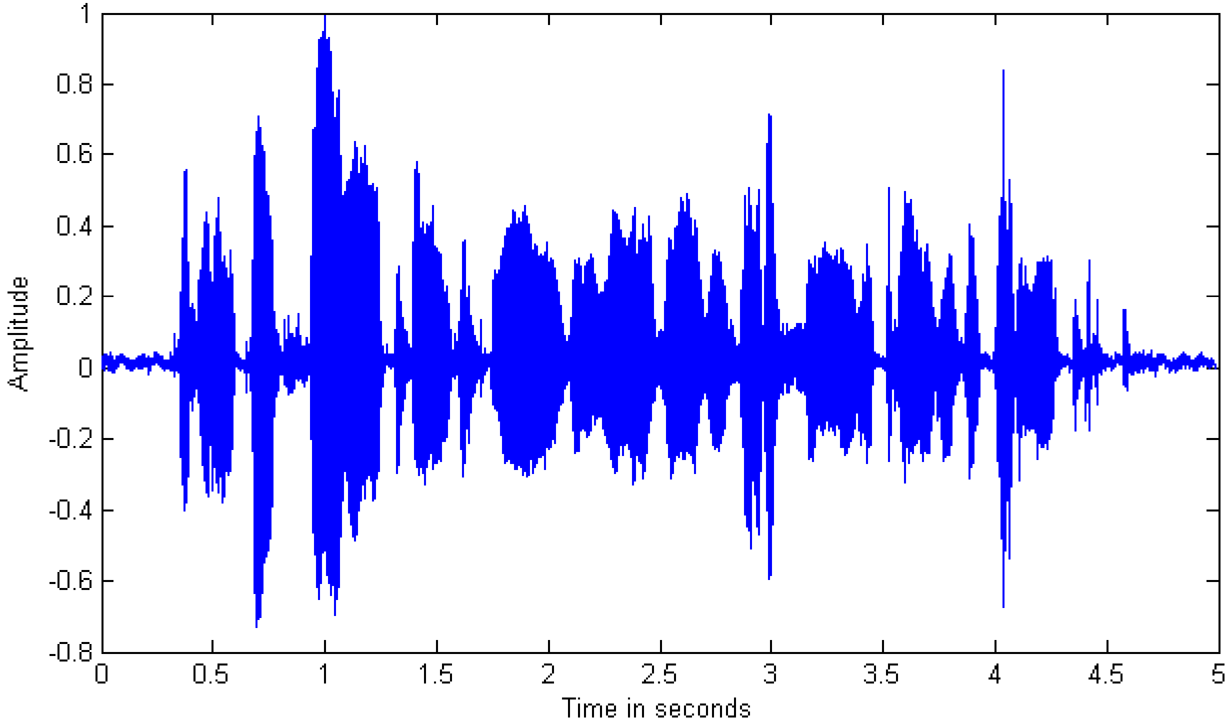
How to load an audio file? (cont.)

For example, with the following codes, some basic information about the audio file can be extracted.

```
[data, sampleRate,nbits] = wavread('speech_dft.wav');

fprintf('audio length = %g s\n', length(data)/sampleRate);
fprintf('sample rate = %g Hz\n', sampleRate);
fprintf('number of bits persample = %g \n', nbits);
%plot the amplitude.
plot((1:length(data))/sampleRate, data);
xlabel('Time in seconds');
ylabel('Amplitude');
```

How to load an audio file? (cont.)



How to play an audio file?

After you have load the audio data into the memory, you can use the following method to play it:

```
sound (data, sampleRate)
```

Note that: values in data should be between -1~1

Example:

```
[data, sampleRate,nbits] = wavread('speech_dft.wav');
sound(data, sampleRate);
```

How to write an audio file?

You can use the following method to write audio data back to a file:

```
wavWrite(data, sampleRate, nBits, filename)
```

Audio signal re-sampling and re-quantization

- Re-sampling: Re-sample the original audio signal with a lower rate. For example, after such an operation, we can change the sampling rate of an audio signal from 80KHZ to 40KHZ.
- Re-quantization: Quantize the original audio signal to a lower precision. That is using less bits to represent a sample. For example, if the original audio data range is from -32768~32768 (16bit persample), after re-quantization, we can map it to 0~255 (8bit persample)

Audio signal re-sampling and re-quantization (cont.)

```
%%%%%%
%%%%%%1. load an audio segment.
[data, sampleRate,nbits] = wavread('speech_dft.wav');

%print out some information about this audio segment.
fprintf('audio length = %g s\n', length(data)/sampleRate);
fprintf('sample rate = %g Hz\n', sampleRate);
fprintf('number of bits persample = %g \n', nbits);
%plot the amplitude.
plot((1:length(data))/sampleRate, data);
xlabel('Time in seconds');
ylabel('Amplitude');
%play this audio segment.
sound(data, sampleRate);
```

Audio signal re-sampling and re-quantization (cont.)

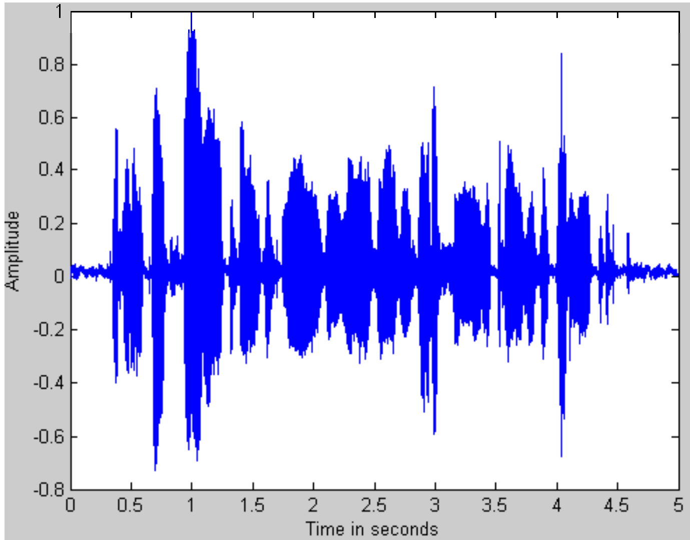
```
%%%%%%
%%%%%%2. Re-quantize the original audio
%%%%%%%%%%segment to a lower precision.
%get the native sound data.
[nativedata]= wavread('speech_dft.wav','native');
nativedata = double(nativedata);
newNBits = 4; %set new number of bits persample

newQData = (nativedata + 2^(nbits-1)) *
          (2^newNBits-1) / (2^(nbits-1)*2 - 1);
newQData = round(newQData);
newQData = 2*newQData / (2^newNBits-1) - 1;
%plot the amplitude after re-quantization.
figure;
plot((1:length(newQData))/sampleRate, newQData);
xlabel('Time in seconds');
ylabel('Amplitude');
%play the re-quantized sound data.
sound(newQData, sampleRate);
```

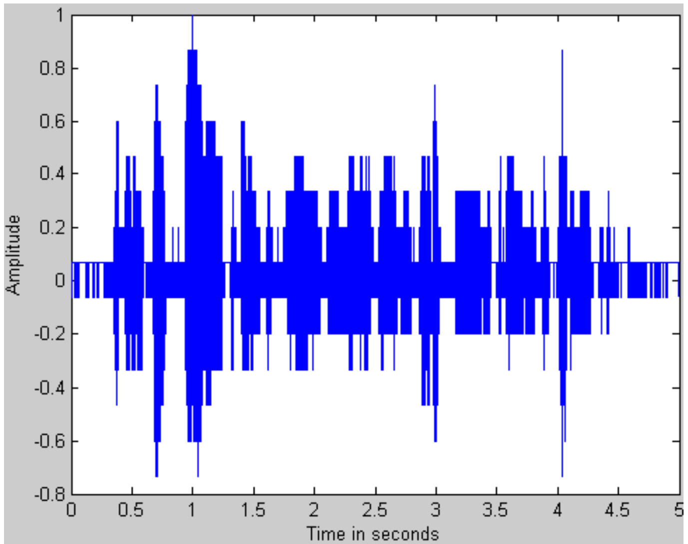
Audio signal re-sampling and re-quantization (cont.)

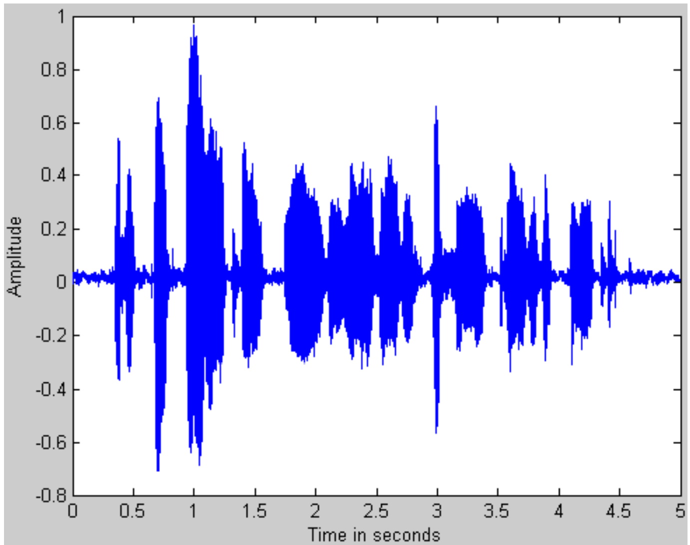
```
%%%%%%
%%%%%%3. Re-sample the original audio
%%%%%%segment with a lower sampling rate.
newSampleRate = 5000; %set the new sample rate.
newData = resample(data,newSampleRate,sampleRate);
figure;
plot((1:length(newData))/newSampleRate, newData);
xlabel('Time in seconds');
ylabel('Amplitude');
sound(newData, newSampleRate);
```

Audio signal re-sampling and re-quantization (cont.)



Original signal





After re-quantization

After re-sampling

- Please go to

  http://www4.comp.polyu.edu.hk/~cslzhang/comp5422/

  downloading the file “audio_lab.rar”.