Manual CPSP environment

Table of contents

Introduction	1
Setup the environment	1
Making cochleograms	1
Using functionsTest	1
Using commands	1
The result	2
Analyzing cochleograms	2
Example Introduction	2

The CPSP environment is a toolset for sound annotation and sound processing. This manual focusses on the processing part of the environment. The main processing step included is a gammachirp filterbank.

Setup the environment

In matlab go the the cpsp directory and run startup there: cd /home/soundrecognition/cpsp/current/ startup Go back the a directory where you can write.

Making cochleograms

A cochleogram is a time-frequency-energy plane as computed by a model of the human ear. There is a GUI-way, functionsTest, and a command-line-way to compute these cochleograms.

Using functionsTest

The GUI is started with the functionsTest command. On the left is a list of functions that can be used. We'll mostly use calcEdB. You use calcEdB by selecting it and pressing the '>' button. It is now listed in the right listbox. Now press "solve dep.", this leads to the correct settings. Finally press "run wav" and select your wav file.

Using commands

If you have a wave-file use: processfile(wavfilename, 'require', 'EdB'); where wavfilename is the name of your wav file. If you have a variable in your matlab workspace:

00	functionsTest	
🖻 🖥		
a	Auditory Cognition Group	
Show hidden	Refresh Spectrogram 🛟	
HC Bits addNoise addReverb aggressionDetecto basic Texture calcBG calcBG calcBS calcNSC calcRS calcRS calcRSPulse dataCommunicatio	or > <	Run Live Run Var Run Wav Run Dataset Run Last
detPulse initCoCoSizes logToFile	x x	Stop
(Config TFProcessor Clear List Solve Dep.	
Author	Didion Vindom	Edit Function
Name:	Calculates EdB	
Description:	Calculates the logarithmic energy distribution	Edit Info
		Edit init
ld:	\$Id: calcEdB_info.m 324 2006-05-12 09:08:24Z dirkian \$	Edit shift
Provides:	EdB	
Order:	5	Edit cleanup
functionsTest_Ope	ningFcn, 104: FunctionsTest ready	Plot tools

processmem(x, 'sampleRate', fs, 'require', 'EdB');
where x is the variable and fs its sample frequency.

The 'require' argument takes 'EdB' as argument in this case. If you want other functions (see the left listbox of functionsTest) to run look at the 'Provides' field in the lower part of functionsTest when the desired function is selected. For example, if you want the regionBasedNSC to run use:

```
processmem(x, 'sampleRate', fs, 'require', 'NSC');
```

The result

The result of both ways of computations is a structure in the matlab workspace called D. The D structure has the following fields (their existence depends on the functions selected/ required):

Fieldname	Size	Description
audio.data	1xnSamples	The audio data used for the cochleogram
audio.sampleFrequency	1	The samplefrequency of D.audio.data
Е	nseg x nframes	The energy of the cochleogram on a linear scale

Fieldname	Size	Description
EdB	nseg x nframes	The energy of the cochleogram on a logarithmical scale
PAS_S	nseg x nframes	The peaks-above-surroundings in the frequency direction. Show the local tonality
PAS_T	nseg x nframes	The peaks-above-surroundings in the time direction. Show the local pulsality
NSC_S	nseg x nframes	The index map for the tonal signal components
NSC_T	nseg x nframes	The index map for the pulsal signal components

Analyzing cochleograms

To display a cochleogram on the screen use the following commands:

```
figure;
imageMask;
seg2freqaxis(gca);
```

The figure command opens a new figure window so the next commands don't overwrite the existing figures. ImageMask draws the cochleogram currently in the D structure with a proper dynamic range. The seg2freqaxis(gca) changes the segments/frames axis to frequency/time axis.

If you need detailed information about a point use the datatip(*) from the figure toolbar. The compute the frequency from the y value use seg2freq(y) to compute the time from the frame number(x-value) type x*env.frameSize/1000;

Example

```
The following code produces a cochleogram of 'hallo.wav'
[x fs] = wavread('/home/soundrecognition/cpsp/current/hallo.wav');
processmem(x, 'sampleRate', fs);
```

```
figure;
imageMask;
seg2freqaxis(gca);
```

The figure should look like:



The D structure should be:

D =

audio: [1x1 struct]
 E: [100x247 double]
 EdB: [100x247 double]`