

Basic Usage of Wavetave

Rocaloid Development Team

Built on March 7, 2014

1 About Wavetave

Wavetave is an audio analysis and processing sandbox written in Octave Language and C++. The initiative of creating Wavetave is to test the feasibility of new algorithms, for example, the improved EpR Voice Model for CVE3.5.

Before any new algorithm is put into application, it should be implemented and tested out in Wavetave first.

Wavetave has two parts: SpectrumVisualizer and MinCVE. The former is a powerful audio analyzer based on gnuplot; the later is a minimal and **very incomplete** version of CVE3.5, being developed just for testing the feasibility of CVE3.5.

1.1 Source Code Quick Guide

The code repository for Wavetave:
<https://github.com/Rocaloid/Wavetave>

- All source codes of Wavetave are placed under `/src`.
- SpectrumVisualizer and all of its plugins are placed under `/src`.
- MinCVE is placed under `/src/MinCVE`.
- Some useful code snippets are placed under `/src/Util`.
- All C++ written modules are placed under `/src/Oct`.

1.1.1 Dependencies

You need gnuplot, octave, octave-dev and g++ to compile and run Wavetave.

1.1.2 Building Wavetave

```
1 $ git clone https://github.com/Rocaloid/Wavetave
2 $ cd Wavetave/src/Oct
3 $ for i in *.cc; do mkoctfile $i -s; done #This may take a while
```

2 SpectrumVisualizer

2.1 Basic Usage

SpectrumVisualizer can be activated in two ways: direct execution or activating from an octave console.

```
1 $ cd Wavetave
2 $ ./SpectrumVisualizer.m
```

Or

```
1 octave:1> SpectrumVisualizer
```

After starting up you will get two three windows: two plot windows and one terminal.

Figure 1 is used to display the time-domain signal.

Figure 2 displays the decibel-magnitude spectrum of the analyzed part of the signal.

The terminal shows hints and extra information of the analyzed signal.

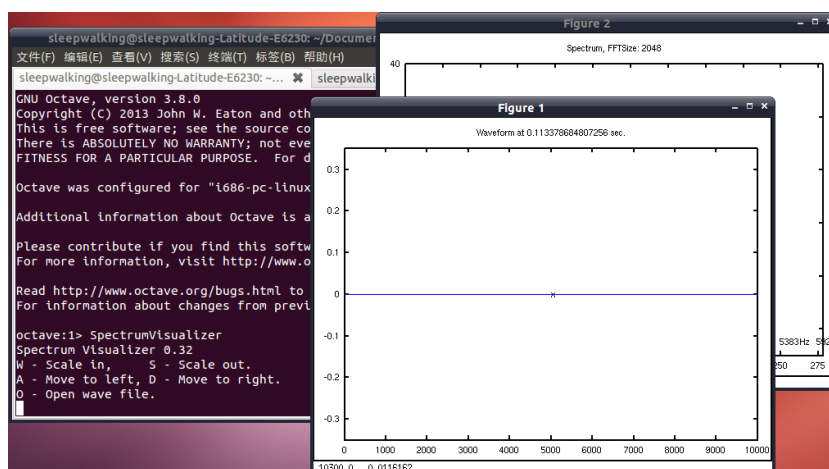


Figure 1: SpectrumVisualizer starts up

When interacting with SpectrumVisualizer, you should always keep Figure 1 focused because it is the figure listening for mouse and key events.

By default no wave is loaded when SpectrumVisualizer starts. Waves can be loaded by pressing O, then a prompt goes out in the terminal:

Wave to open(enclosed by quotes):

Input the file path with quotes. Currently SpectrumVisualizer only supports **single channel** wav files. The recommended sample rate is **44100Hz** and the recommended length is **1 to 5 seconds**. For example:

Wave to open(enclosed by quotes): '/tmp/test.wav'

If you are using gnome-terminal, this can be simply done by dragging the file into the terminal.

Then press enter.

Our example is a short pronunciation of 'A' that lasts for 1 second.

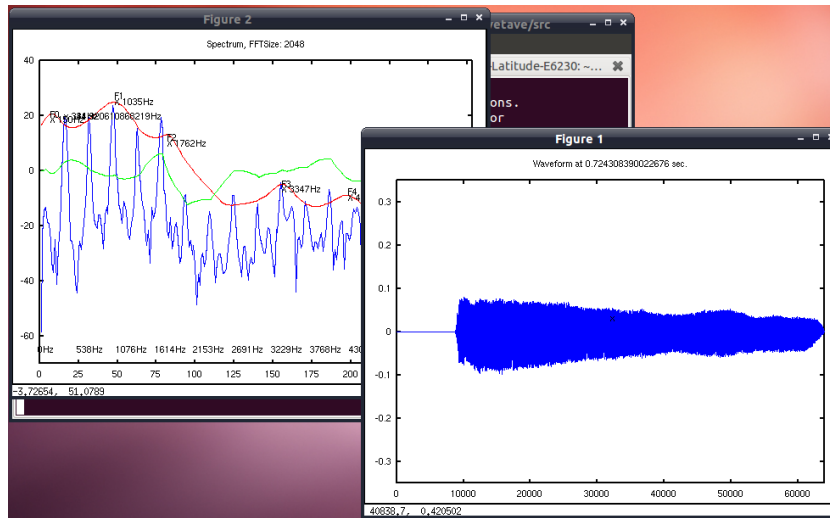


Figure 2: 'A'

Some plugins for Wavetave are activated by default, as shown in Figure 2 above. The red and green line and the labels come from Plugin_FormantFitting. We are not sure about which plugins are set to be activated by default because the settings are constantly changed.

2.1.1 Surfing through the Wave

You can change the position of the analysis window by clicking on Figure 1.

Press W to scale in; Press S to scale out.

Press A to move the window to left; Press D to move to right.

Depending on the speed of your device and the number of plugins that are activated, the response speed of SpectrumVisualizer may vary a lot. When it appears to be too slow, try to disable some plugins.

2.1.2 Configuring the Plugins

Open /src/SpectrumVisualizer.m with your favourite editor. Look for the following lines:

```

1 global Plugin_Load = [
2     "Empty"
3     "Plugin_Load_EpRInitialization"
4     # "Plugin_Load_PulseMarking"
5     # "Plugin_Load_PulseMarking_Stable"
6     # "Plugin_Load_PulseMarking_Naive"

```

```

7         ];
8     ...
9     global Plugin_Wave = [
10         "Empty"
11         #     "Plugin_UnvoicedDetection"
12         #     "Plugin_PulseMarking"
13         #     "Plugin_VOTMarking"
14     ];
15     ...
16     global Plugin_Spectrum = [
17         "Empty"
18         "Plugin_FOMarking"
19         "Plugin_FOMarking_ByPhase"
20         "Plugin_FormantFitting"
21         #     "Plugin_Freq2Pitch"
22         #     "Plugin_HarmonicMarking"
23         #     "Plugin_HarmonicMarking_Naive"
24         #     "Plugin_PhaseFigure"
25     ];

```

Activate or deactivate a plugin by removing or adding the comment mark before it.

- Plugin_Load stands for plugins that are called when a new wave file is loaded.
- Plugin_Wave stands for plugins that are called when Figure 1 is repainted.
- Plugin_Spectrum stands for plugins that are called when Figure 2 is repainted.

If you have implemented your own plugin, you can add its name to the above list. Make sure in each group of plugins there is at least an “Empty” because it keeps the list from being empty.