

Spec-Recorder の Read me

ライセンス:MIT ライセンス

動作環境:今のところ、Ubuntu12.04 以降と Windows7 で動作を確認しています。きっと MacOSX や他の Linux ディストリビューションでも動きます。HTTP 経由の画像閲覧のクライアントは、Chrome のみ動作を確認済み

必要なモジュールなど:Python2.7 と Python Imaging Library(PIL)1.1.7 と SoundExchange(SoX)14.4.1

概要

このソフトウェアは主に流星電波観測のための、コンピュータのデフォルトのオーディオデバイスの入力信号をスペクトログラムとして PNG ファイルに書き出し、CSV ファイルに変換し、エコーを自動でカウントするものです。

Ubuntu でのインストール方法

```
sudo apt-get install sox libsox-fmt-all python-imaging
```

を実行します。

spec-recorder の Zip ファイルをダウンロードして、展開します。

Windows でのインストール方法

spec-recorder の Zip ファイルをダウンロードして、展開します。

<http://www.python.jp/download/> から Python2.7.5 のインストーラをダウンロードし、インストールしてください。32bit の Windows を使っている場合は x86 を、64bit の場合は x86-64 を選んでください。

<http://www.pythonware.com/products/pil/> から Python Imaging Library 1.17 for Python2.7 をダウンロードし、インストールしてください。

<http://sourceforge.net/projects/sox/files/sox/14.4.1/> から sox-14.4.1a-win32.zip をダウンロードし、展開してください。展開したフォルダの中のすべてのファイルを spec-recorder のフォルダにコピーしてください。spec-recorder のフォルダの中にある sox.exe をどこか別の場所にコピー&ペーストして、そのファイル名を soxi.exe に変え、spec-recorder のフォルダにコピーしてください。spec-recorder のフォルダの中に sox.exe と soxi.exe が両方あるようにしてください。

ubuntu での使い方

無線機などのオーディオ出力をシステムのデフォルトのサウンドデバイスの入力につないでください

コマンドラインを開いて main.py があるフォルダに移動してください

```
python main.py
```

を端末から実行してください

適宜、ボリュームを調節してください

……待つ

5 分毎に一つずつ PNG ファイルと MP3 ファイルと CSV ファイルが生成されます

終了させるときは、Ctrl と C を同時に押してください。

Windows での使い方

PC のマイク入力に無線機からのオーディオ出力をつなぎ、コントロールパネルでそのオーディオデバイスをシステムのデフォルトに指定します。(これをしないと SoX が動かない可能性があります。)

main.py をダブルクリックします

(.py のファイルが Python に関連付けられていれば、これで起動するはずです)

起動できないときは、IDLE(Python と一緒にインストールできます)で main.py を開いて F5 を押して実行してください。

Windows での注意点

Windows での使用では今のところ、オーディオファイルの自動削除はできません。自動カウントはできるようになりました。

Config ファイルの書き方

server = off を server = on に書き換えると Web サーバーの機能が有効になります。他も同様です。

[sox]

high_color:ハイカラーの PNG ファイルを生成する。

mono_color:モノクロの PNG ファイルを生成する。#人がカウントするときは off のほうがいいです。

[server]

server:Python 内蔵の HTTP サーバーを使う。

ポートは 8000 番、localhost:8000/latest_image.html に WEB ブラウザでアクセスすると、最新の PNG ファイルが表示されます

[file]

del_ogg:Ogg Vorbis ファイルを自動で削除する。(PNG ファイルは残る)

write_filename:PNG ファイルの中に時刻を書き込む

write_csv:CSV ファイルとしてデータを書き出す

[analyze]

analyze:自動エコーカウントを有効にする

エコーの自動カウント機能について

この機能は、CSV ファイルを読み込み、その中からエコーを探し、エコーの始まりの PNG ファイルの X 軸のピクセルを echo の CSV 形式のファイルに書き出すものです。その列数がエコーの数となります

この機能はまだ開発途中で、飛行機や家電製品由来の信号があると、それをエコーとして数えてしまう可能性があります。あなたがよほどノイズの少ない環境に住んでいない限り、今のところは、この機能を全面的に信頼しないほうがいいと思われます。

この機能をより良い物にするために、実際にお使いいただき、フィードバックをいただけると幸いです。

更新履歴

beta1	基本的な機能の実装
beta2	HTTP サーバー機能などの実装 SoX の呼び出しに用いる関数の一部変更
Ver.1.0	wave.py のみの実行で freq_exec.py も実行されるように変更 画像の自動解析の開発中
Ver.1.01	wave.py を main.py に変更 (Python の WAVE モジュールと名前が被るため) エコーの自動カウントの開発進行 (まだ使える状態ではない) CSV 形式でのデータの書き出しの実装
Ver.1.1	流星の自動カウント機能のマージ 時々 PNG ファイルが生成されないという重大な欠陥の修正 Windows 版の開発 ファイルの保存先を各フォルダに分けるように変更
Ver.1.11	Windows 対応!!! (自動カウントは未対応 PNG ファイルの生成はできる) 時刻管理の改善 音声ファイルを Ogg Vorbis に変更

Ver.1.12	Windows での自動カウントに対応 Echo ファイルの書き出し先を echo フォルダに変更 エラーが起ころともとりあえずメインループを実行し続けるように変更
----------	--

謝辞:SoX の開発者の方に感謝いたします