pyrecspecwaterfall

May 3, 2017

1 Python Example

Using Pyaudio, record sound from the audio device and plot a waterfall spectrum display, for 8 seconds.

Usage example: `python pyrecspecwaterfall.py`

— `Gerald Schuller, November 2014`

1.1 Input:

As the program runs the recording starts and wait for the user inputs from the keyboard.

** 'q' - quit the recording and hence the program (program will not end by clicking close on the window, so instead press 'q')**

1.2 Output:

Real time audio spectrogram for the processed blocks of the recording moving upward(like an upside down waterfall).

1.2.1 Importing relevant modules:

```
In [1]: import pyaudio
        import struct
        import numpy as np
        import cv2
```

1.2.2 Defining the variables:

```
In [2]: CHUNK = 1024 #Blocksize
        WIDTH = 2 #2 bytes per sample
        CHANNELS = 1 #2
        RATE = 32000  #Sampling Rate in Hz
```

1.2.3 Initialize the sound card and print out the detail specs about the inputs:

```
In [3]: p = pyaudio.PyAudio()

        a = p.get_device_count()
        print("device count=",a)
```

```
        for i in range(0, a):
            print("i = ",i)
            b = p.get_device_info_by_index(i)['maxInputChannels']
            print(b)
            b = p.get_device_info_by_index(i)['defaultSampleRate']
            print(b)


        stream = p.open(format=p.get_format_from_width(WIDTH),
                        channels=CHANNELS,
                        rate=RATE,
                        input=True,
                        output=True,
                        #input_device_index=3,
                        frames_per_buffer=CHUNK)
```

```
('device count=', 12L)
('i = ', 0)
2
44100.0
('i = ', 1)
2
44100.0
('i = ', 2)
0
44100.0
('i = ', 3)
0
44100.0
('i = ', 4)
2
44100.0
('i = ', 5)
2
44100.0
('i = ', 6)
0
44100.0
('i = ', 7)
0
44100.0
('i = ', 8)
0
44100.0
('i = ', 9)
2
44100.0
```

```
('i = ', 10)
2
44100.0
('i = ', 11)
0
44100.0
```

1.2.4 Start recording and simultaneouly plotting the waterfall(going upwards). The colours in each row shows freuency intensities horizontally:

```
In [4]: print("* recording")
        #Size of waterfall diagramm:
        #max CHUNK/2 rows:
        rows=500
        cols=512
        fftlen=cols*2
        frame=0.0*np.ones((rows,cols,3))

        while(True):

            #Reading from audio input stream into data with block length "CHUNK":
            data = stream.read(CHUNK)
            #Convert from stream of bytes to a list of short integers (2 bytes here) in "samples
            #shorts = (struct.unpack( "128h", data ))
            shorts = (struct.unpack( 'h' * CHUNK, data ));
            samples=np.array(list(shorts),dtype=float);

            #shift "frame" 1 up:
            frame[0:(rows-1),:]=frame[1:rows,:];
            #compute magnitude of 1D FFT of sound
            #with suitable normalization for the display:
            #frame=np.abs(np.ffqt.fft2(frame[:,:,1]/255.0))/512.0
            #write magnitude spectrum in lowes row of "frame":
            R=0.25*np.log((np.abs(np.fft.fft(samples[0:fftlen])[0:(fftlen/2)]/np.sqrt(fftlen))+1
            #Color mapping:
            #Red:
            frame[rows-1,:,2]=R
            #Green:
            frame[rows-1,:,1]=np.abs(1-2*R)
            #Blue:
            frame[rows-1,:,0]=1.0-R
            #frame[rows-1,:,0]=frame[rows-1,:,1]**3
            # Display the resulting frame
            cv2.imshow('frame',frame)
            #Keep window open until key 'q' is pressed:
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
```

```
        # When everything done, release the capture

        cv2.destroyAllWindows()

        stream.stop_stream()
        stream.close()
        p.terminate()
```

* recording