pyrecfftanimation

May 3, 2017

1 Python Example: *pyrecfftanimation*

Using Pyaudio, record sound from the audio device and plot the fft magnitude spectrum life, for 8 seconds.

Usage example: `python pyrecfftanimation.py`
`- Gerald Schuller, October 2014.`

1.1 Import the modules and define the variables.

```
In [1]: import pyaudio
        import struct
        #import math
        #import array
        import numpy as np
        #import sys
        #import wave
        import matplotlib.pyplot as plt
        import matplotlib.animation as animation
        #import pylab
        #import cv2

In [2]: CHUNK = 2048 #Blocksize
        WIDTH = 2 #2 bytes per sample
        CHANNELS = 1 #2
        RATE = 32000  #Sampling Rate in Hz
        RECORD_SECONDS = 70

        fftlen=CHUNK/2
```

1.2 Initialize the plots:

```
In [3]: [fig, ax] = plt.subplots()
        plt.ylabel('dB')
        plt.xlabel('FFT bins/Subbands')
        plt.title('Live FFT Magnitude Spectrum of Microphone Signal')

        x = np.arange(0, fftlen)        # x-array
```

```
        #Set scale on y-axis and generate line object with it:
        [line, ]= ax.plot(x, 100.0**np.sin(x))
```

1.3 Function for animation:

```
In [4]: def animate(i):
            # update the data
            #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);

            #plt.plot(samples)  #<-- here goes the signal processing.
            line.set_ydata(20.0*np.log((np.abs(np.fft.fft(samples[0:fftlen])/np.sqrt(fftlen))+1)
            #line.set_ydata(samples)
            return line,

In [5]: def init():
            line.set_ydata(np.ma.array(x, mask=True))
            return line,
```

1.4 Initialize the soundcard:

```
In [6]: 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
```

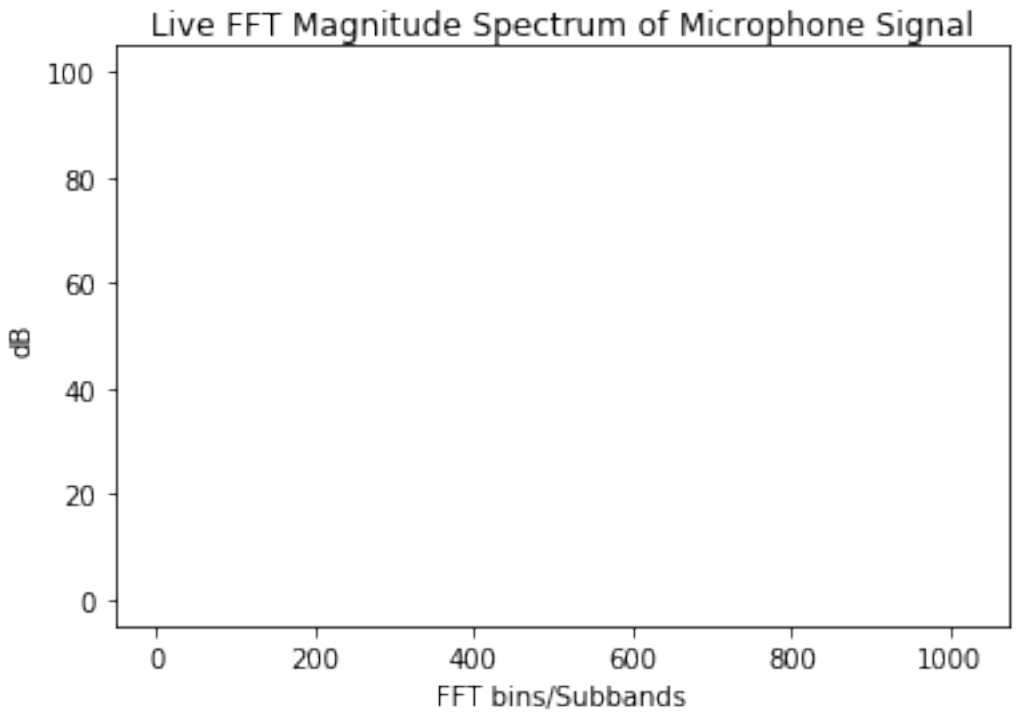
```
44100.0
('i = ', 1)
2
44100.0
('i = ', 2)
0
44100.0
('i = ', 3)
0
44100.0
('i = ', 4)
2
44100.0
('i = ', 5)
2
44100.0
('i = ', 6)
0
44100.0
('i = ', 7)
0
44100.0
('i = ', 8)
0
44100.0
('i = ', 9)
2
44100.0
('i = ', 10)
2
44100.0
('i = ', 11)
0
44100.0
```

1.5 Start recording plot the live animation:

In [7]:
```
print("* recording")

        #ani = animation.FuncAnimation(fig, animate, np.arange(1, 200), init_func=init,
        #    interval=25, blit=True)
        ani = animation.FuncAnimation(fig, animate,  init_func=init, interval=25, blit=True)
        plt.show()


        # When everything done, release the capture
```

```
        print("* done")

        #f.close()
        stream.stop_stream()
        stream.close()
```

```
* recording
```




```
* done
```