pyrecplotanimation

January 25, 2017

1 This program prints out the info about the audio devices and its properties(e.g., default sampling rate). Also it shows a live plot of waveform of the recording it does, while the program runs.

1.0.1 Input:

As you run the program it starts recording thorugh the selecetd microphone device.

1.0.2 Output:

The outputs are as following: 1. In the console it prints out the counts of the devices, their input channels and the sampling rate it uses. 2. A live plot of the waveform being recorded(The output is a bit delayed as the processing takes some time)

Import the relevant modules

In [ ]:
```
"""
Using Pyaudio, record sound from the audio device and plot, for 8 seconds, and display i
Usage example: python pyrecplotanimation.py test.wav
Gerald Schuller, October 2014
"""
%matplotlib inline
import pyaudio
import struct
import numpy as np
import sys
import wave
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import pylab
import cv2
```

1.0.3 Note: Exclude '%matplotlib inline' when running the code in your IDE.

Define the variables

In [ ]:
```
CHUNK = 1024 #Blocksize
WIDTH = 2 #2 bytes per sample
CHANNELS = 1 #2
RATE = 32000  #Sampling Rate in Hz
RECORD_SECONDS = 70
```

In [ ]:
```
fig, ax = plt.subplots()

x = np.arange(0, CHUNK)        # x-array
#Scale axis as this sine function:
line, = ax.plot(x, 20000.0*np.sin(x))

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(np.log((np.abs(pylab.fft(samples))+0.1))/np.log(10.0))
    line.set_ydata(samples)
    return line,

def init():
    line.set_ydata(np.ma.array(x, mask=True))
    return line,


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)


print("* recording")

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


# When everything done, release the capture

print("* done")

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