

pyrecplotanimation

January 27, 2017

0.1 Python Example:

Using Pyaudio, record sound from the audio device and plot, for 8 seconds, and display it live in a Window. Usage example: python pyrecplotanimation.py Gerald Schuller, October 2014

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
```

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

```
In [2]: 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))
```

Function for animation.

```
In [3]: 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,

```

```

In [4]: def init():
        line.set_ydata(np.ma.array(x, mask=True))
        return line,

```

Initialize the soundcard and print the sound card information.

```

In [5]: 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
```

Start recording and live plot of it.

```
In [6]: # When everything done, release the capture
```

```
print("* done")

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

```
* done
```