

Project 1 编程实现音乐节奏或旋律的可视化

使用 python 编程实现音频可视化。使用到了 pyaudio^[1], wave^[2], librosa^[3], numpy^[4] 库。

1. 文件目录

```
.
├── demo.mov           // 截屏演示视频，文件太大没有放在邮件附件中，参见GitHub链接
├── main.py            // 源代码
├── readme.md          // markdown报告
├── report_PJ1.pdf     // pdf报告
└── swan_lake.wav      // .wav格式音乐样例
```

2. 算法说明

2.1 读取和播放音频

pyaudio 和 wave 库用于读取和播放 .wav 格式音频。

```
# 创建播放器
p = pyaudio.PyAudio()
with wave.open(music_file, 'rb') as wf:
    # 打开音乐文件
    stream = p.open(
        format=p.get_format_from_width(wf.getsampwidth()),
        channels=wf.getnchannels(),
        rate=wf.getframerate(),
        output=True # 设置为True表示输出音频
    )
```

2.2 获取音频节奏

调用了 librosa 库提供的 `librosa.beat.beat_track()` 函数获取音频的节奏帧。

```
y, sr = librosa.load(music_file)
tempo, beat_frames = librosa.beat.beat_track(y=y, sr=sr)
```

2.3 获取音频频谱

获取音频每一帧的频谱图。采用快速傅里叶变换 (FFT) 将每一个音频帧的信号变换到频域，得到这一帧的不同频率的振幅。调用了 numpy 库提供的快速傅里叶变换函数 `np.fft.fft()` 实现。

```
# 当音乐还未结束
while data != '':
    stream.write(data)
    # 继续读取音频数据
    data = wf.readframes(nframes)
    # 傅里叶变换
    data_fft = np.real(np.fft.fft(np.fromstring(data, dtype=np.int16)))
```

2.4 可视化

使用 pygame 库创建可视化窗口。调用 `pygame.draw.rect()` 函数，绘制每个音频帧的频谱，并通过 `color` 参数和帧计数控制颜色变换。根据 `beat_frames` 在节奏点时刻增大可视化的bar高度，强调节奏

点，达到节奏可视化的效果。

```
for n in range(0, data_fft.size, step):
    bar_height = min(abs(int(data_fft[n] * 1e-4)), 0.4 * window_height)
    # 根据节奏点增大bar_height，达到可视化节奏效果，iter为音频帧计数
    if iter in beat_frames:
        bar_height = min(bar_height + 5, 0.4 * window_height)
    # 画bar，上下对称，控制颜色变换
    pygame.draw.rect(screen, colors[int(n / data_fft.size * len(colors) - iter / 2) % len(colors)], p)
    pygame.draw.rect(screen, colors[int(n / data_fft.size * len(colors) - iter / 2) % len(colors)], p)
```

3. 运行方式

- 安装 pyaudio, wave, numpy, pygame, librosa 库
- 音频文件必须是 **.wav** 格式
- 在Terminal中输入命令：`python main.py swan_lake.wav` , `swan_lake.wav` 可替换成其他音频文件路径

4. GitHub 链接

https://github.com/jiayunz/Computer_Graphics/tree/master/PJ1

-
1. <https://www.opengl.org/> ↗
 2. <https://pythonhosted.org/Wave/> ↗
 3. <http://librosa.github.io/librosa/> ↗
 4. <https://numpy.org/> ↗