

2021 春信号与系统大作业之“B 站，我来了 !”(初稿)

谷源涛

2021 年 6 月 19 日

本次作业研究混剪视频的自动制作方法。

1 问题建模

记背景音乐为 bgm, 记第 n 个电影 $m^{(n)}$, 用 $m_{s(n),e(n)}^{(n)}$ 表示第 n 个电影从 $s(n)$ 秒开始到 $e(n)$ 秒结束的片段, 用函数 $\text{pin}(a, b, c, \dots)$ 表示将电影片段 a, b, c 等拼成一个拼接电影, 用 $\text{Chongjili}(a, b)$ 表示将拼接电影 a 和背景音乐 b 叠加在一起的冲激力, 则混剪视频问题可以建模为

$$\min_S -\text{Chongjili} \left(\text{pin} \left(m_{s(I_1),e(I_1)}^{(I_1)}, m_{s(I_2),e(I_2)}^{(I_2)}, \dots \right), \text{bgm} \right)$$

其中

$$S = \{I_1, I_2, \dots, s(1), s(2), \dots, e(1), e(2), \dots\}$$

表示未知变量集合。显然这是一个非常困难的问题, 一般由人工解决。

困难之一在于 $\text{Chongjili}(a, b)$ 函数没有解析形式的定义。一般来说, 拼接电影 a 和背景音乐 b 的“节奏感”越符合, $\text{Chongjili}(a, b)$ 越大。常见的人工处理的混剪视频按“节奏感”及其符合方式大致可分为两类,

- 一类是电影转场 (镜头切换) 和背景音乐对齐, 典型样例包括

样例 1

- 另一类是电影的“激烈”程度和背景音乐的“激烈”程度对齐, 例如

样例 2

其中电影的“激烈”程度又可细分为电影视频的“激烈”程度和电影音频的“激烈”程度。

我们计划采用第二类里电影音频的“激烈”程度和背景音乐的“激烈”程度对齐的方式，后文将具体研究量化描述。

除了 $\text{Chongjili}(a, b)$ 函数太玄之外，这还是一个连续变量又有离散变量的混合优化问题，非常困难。我们将作业简化为：电影片段已经选好，只要按最优方式拼在一起就好。即省掉了 $\{s(n), e(n)\}_t$ 等变量，问题简化为

$$\min_{I_1, I_2, \dots} -\text{Chongjili}\left(\text{pin}\left(m^{(I_1)}, m^{(I_2)}, \dots\right), \text{bgm}\right)$$

这是一个整数规划问题，我们可以用贪婪方法求解。

2 节奏点提取

本节讨论一种“激烈”程度的定量刻画方法，即节奏感。

从音乐中提取节奏点还是一个尚未全部解决的开放问题，请参阅背景材料。

我们参考背景材料第 24-27 页的提取节奏点的方法，大致可分为五步。记音乐信号为 $x(n)$ 。

1. 幅度平方求能量：

$$y_1(n) = x^2(n).$$

2. 加窗平滑得到包络：

$$y_2(n) = \sum_{i=0}^{M-1} w_i y_1(n-i),$$

其中 $\{w_0, w_1, \dots, w_{M-1}\}$ 表示窗函数。

3. 做差分得到能量变化点：

$$y_3(n) = y_2(n) - y_2(n-1).$$

4. 半波整流：

$$y_4(n) = \max\{y_3(n), 0\}$$

5. 自动选峰。这是最开放的环节，可以采用很多启发性方法。例如，首先寻找所有局部极大值，然后按照满足两峰间隔不小于某阈值、能量变化不小于某阈值等选取。

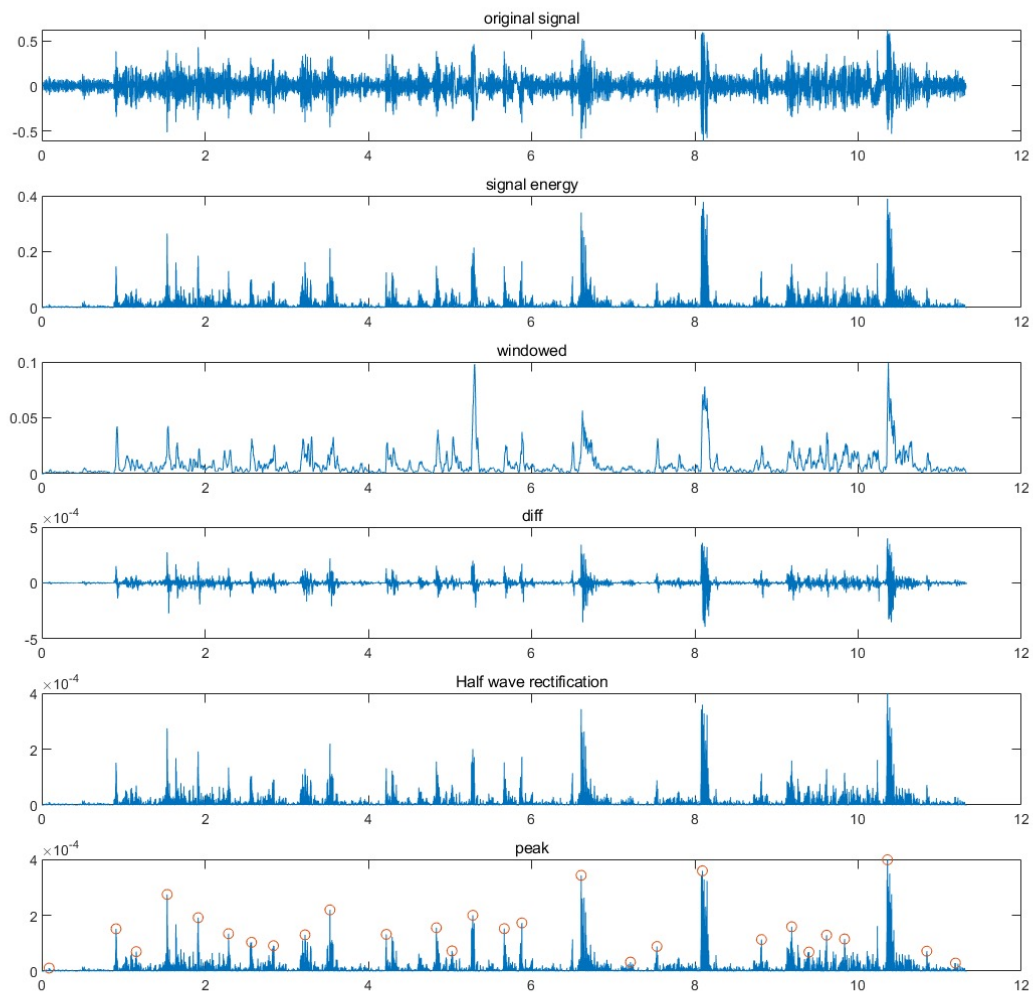


图 1: 音乐信号及其节奏点提取示例

我们选取一段背景音乐，对其依次进行上述处理，结果如图1所示。可见最后一行的圆圈，确实准确提取到了大部分节奏点。

节奏点找到后，接下来就可以转换音频的“激烈”程度。我们用 $y(n)$ 表示节奏点序列，其中 $y(n) = 0$ 表示 n 时刻非节奏点， $y(n)$ 非零表示 n 时刻节奏点的强度，那么

$$z(n) = \sum_{i=-N/2}^{N/2} h_i y(n-i)$$

其中 $h_{-N/2}, h_{-N/2+1}, \dots, h_{N/2}$ 表示低通滤波器系数，即可得到“激烈度”序列 $z(n)$ 。图1所示背景音乐的激烈度如图2所示。

上述方法可以评估背景音乐的“激烈度”，记做 $Z(n)$ ，也可以评估电影里的

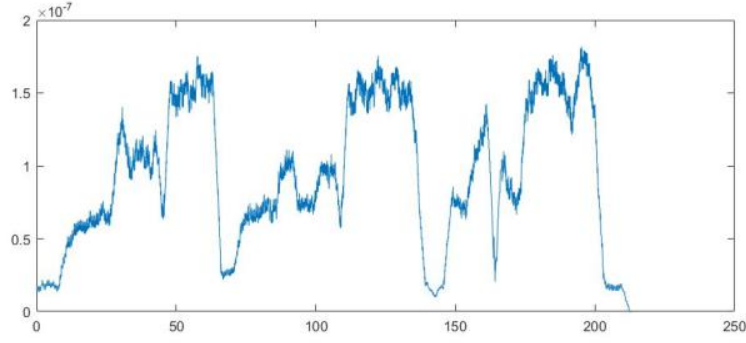


图 2: 音乐信号的激烈度示意

音频的“激烈度”，记做 $\{z_1(n), z_2(n), \dots\}$ ，接下来问题就变成了

$$\min_{I_1, I_2, \dots} -\text{Chongjili}(\text{pin}(z_{I_1}(n), z_{I_2}(n), \dots), Z(n)),$$

我们的任务就是确定下标集 I_1, I_2, \dots 。注意函数定义有些混乱，显然 Chongjili 和 pin 的自变量都已经变了，但应该不影响理解。

3 贪婪算法

贪婪算法是一种迭代算法，其基本思想是不考虑全局最优，仅就目前的状态尽力去做到最好。就本问题而言，可以想到两种启发式的贪婪算法。

1. 首先确定 I_1 ，即求解

$$\min_{I_1} -\text{Chongjili}(z_{I_1}(n), Z_{\text{trunc}}(n)),$$

其中 $Z_{\text{trunc}}(n)$ 表示从 $Z(n)$ 的头部截取和 $z_{I_1}(n)$ 等长的一段。记最优解为 I_1^* ，然后确定 I_2 ，即求解

$$\min_{I_2 \neq I_1^*} -\text{Chongjili}(\text{pin}(z_{I_1^*}(n), z_{I_2}(n)), Z_{\text{trunc}}(n)),$$

其中 $Z_{\text{trunc}}(n)$ 的长度变成了 $z_{I_1^*}(n)$ 和 $z_{I_2}(n)$ 的长度之和。如此往复，直至解出全部下标集。

2. 换一个思路，首先找出所有电影片段中音频最激烈的，即

$$\min_i - \sum_n z_i(n)$$

然后把它和背景音乐里最激烈的段落拼在一起，即.....

4 作业和评分规则

4.1 作业

4.2 评分规则

本次大作业采用“上传报告、代码和实验结果，由助教评判”的传统评分方式。具体流程如下：

- 要求独立完成。禁止任何形式的抄袭。任何抄袭、剽窃等学术不端行为必将受到严厉打击。
- 不限解决方法，不限开发环境；可以用任何工具软件或开源代码（不包括其他人专门为解决本作业开发的工具），但必须注明出处。

5 文件操作和绘图工具介绍

可以用 MATLAB 编程实现，可能用到的专业功能函数如下表所示 [2]。

函数名	类型	说明
wavread	MATLAB 标准	将 wav 文件中的数据读入内存
wavwrite	MATLAB 标准	将内存中的数据写入 wav 文件
plot	MATLAB 标准	绘制波形
sound	MATLAB 标准	播放声音
window	MATLAB 标准	窗函数

也可以用 Python 或其他任何语言实现。

6 致谢

1. 金澄
2. 孟令航、张振威
3.

参考文献

- [1] 郑君里、应启珩、杨为理,《信号与系统》第三版,北京:高等教育出版社, 2011.3
- [2] 谷源涛、应启珩、郑君里,《信号与系统——MATLAB 综合实验》,北京:高等教育出版社, 2008.1