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 $m^{(n)}_{s(n),e(n)}$ 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 $\mathrm{pin}(a,b,c,...)$ 表示将电影片段 a,b,c 等拼成一个拼接电影，用 $\mathrm{Chongjili}(a,b)$ 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

$$\min_{\mathcal{S}} -\mathrm{Chongjili}\left(\mathrm{pin}\left(m^{(I_1)}_{s(I_1),e(I_1)}, m^{(I_2)}_{s(I_2),e(I_2)}, \cdots\right), \mathrm{bgm}\right)$$

其中

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

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 $\mathrm{Chongjili}(a,b)$ 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，$\mathrm{Chongjili}(a,b)$ 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 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

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

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 $\{w_0,w_1,\cdots,w_{M-1}\}$ 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。

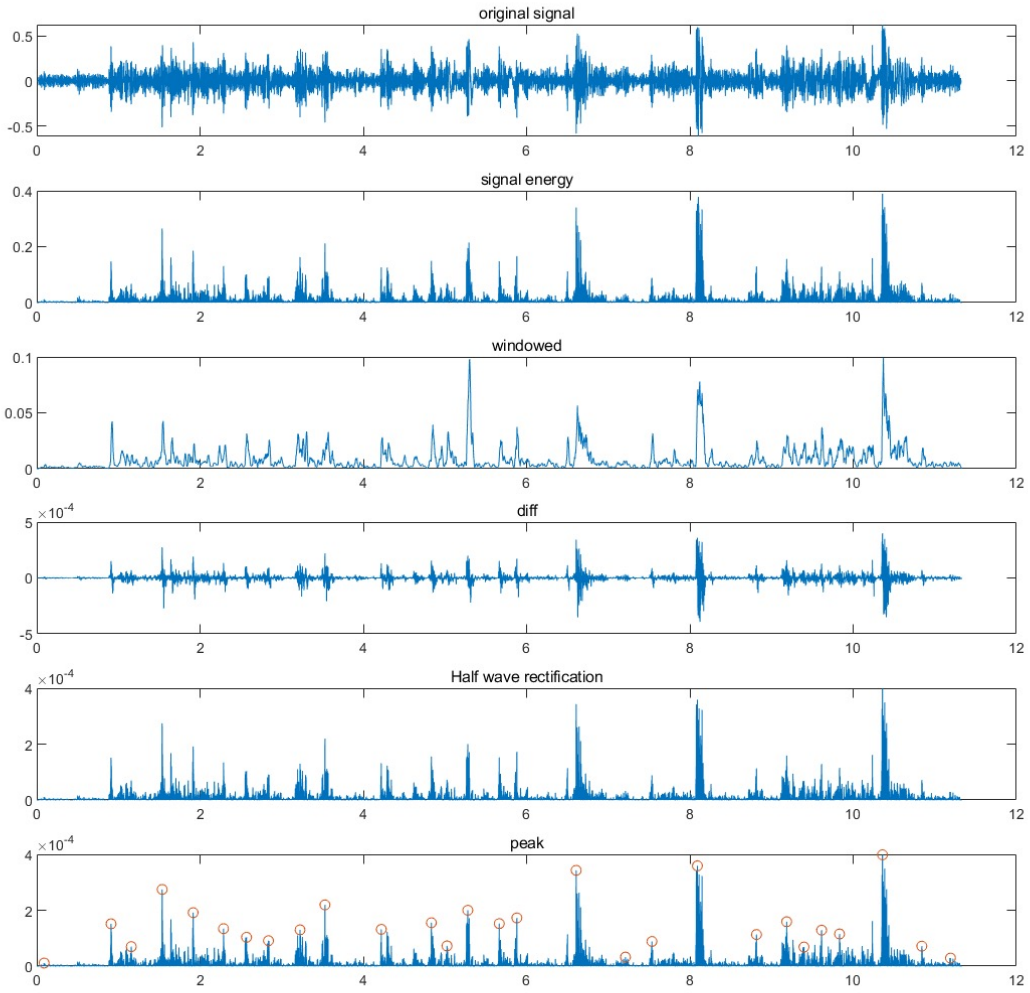

图 1: 音乐信号及其节奏点提取示例

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 $h_{-N/2}, h_{-N/2+1}, \cdots, h_{N/2}$ 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

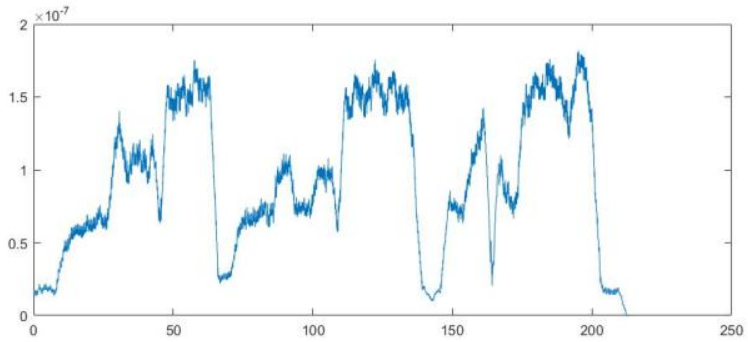

图 2: 音乐信号的激烈度示意

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

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

我们的任务就是确定下标集 $I_1, I_2, \cdots$。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，可以想到以下启发式的贪婪算法。

1. 首先确定 I_1，即求解

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

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

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

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。

4 作业和评分规则

4.1 作业

提供'test.wav' 音频文件与视频素材文件'1.mp4'...'53.mp4'。

4.1.1 节奏点提取

1. 读入'test.wav' 文件，对音频前 12 秒进行章节2中 1-4 步所述处理。调整窗函数的长度与类型，对比能量包络的差异，并在报告中给出对比结果。固定一种窗函数与长度，得到半波整流后结果。
2. 在上一问的基础上，由于存在各种非理想因素及音频特点，第 4 步得到的结果中存在大量非节奏点的峰值。设计自适应选峰算法提取音频峰值，在报告中阐述设计的选峰算法，并给出音频前 12 秒的选峰结果。

4.1.2 激烈度衡量

设计合适窗长与类型的低通滤波器对'test.wav' 的激烈度进行评估，得到音频的 "激烈度" $Z(n)$ 类似图2所示，在报告中给出低通滤波器设计与类似图2的音频 "激烈度" 示意图。

以上得到的 "激烈度" 指标是针对音频单点的激烈度，如何衡量一段音频的 "激烈度" 呢? 我们可以采用 $\frac{\sum_{n=1}^{N} z_i(n)}{N}$ 作为一段视频 "激烈度" 的衡量指标。

用如上指标或者自己设定一段音频 "激烈度" 指标得到'1.mp4'...'5.mp4' 五个视频的音乐 "激烈度" $m_i(n)$，在报告中给出 5 个视频的 "激烈度" 排序。

4.1.3 混剪视频生成

设计算法求解 $I_1, I_2, ...$ 其中 bgm 为'test.wav' 音频文件可供选择的视频素材文件为'1.mp4'...'53.mp4'。一个简单的 "冲击力" 定义是

$$\text{Chongjili}(\text{pin}(z_{(I_1)}, z_{(I_2)}, \cdots), Z(n)) = \sum_i |\frac{\sum_{n=1}^{N} z_{I_i}(n)}{N} - \frac{\sum_{n=1}^{N} Z_{trunc_i}(n)}{N}|$$

其中 Z_{trunc_i} 表示 bgm 中与第 z_{I_i} 段视频对应的音频。

将剪辑视频生成大于 80 秒的混剪视频，上传混剪视频。

$I_1, I_2, ...$ 需由程序自动计算生成，后续拼接视频可由人手动完成。拼接过程中可以执行的操作有且仅有调整导入视频片段顺序与调整 bgm 音量大小。在附录中给出手动拼接视频示例。

混剪视频保留视频声音并加上'test.wav' 的音频，由于'test.wav' 较视频素材声音较大，混剪时可适当降低'test.wav' 音量。

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：

- 本作业满分 100 分（第一题 50 分，第二题 25 分，第三题 25 分），计入总评成绩时加 3 分。
- 请在提交时把如下所有内容放在一个名为 "学号 _ 姓名" 的文件夹里，将文件夹压缩后上传到学堂作业区。
  - 读我：1 个名为 readme.txt 的文件，介绍当前目录下的所有内容；
  - 报告：1 个名为 report.pdf 的文档，描述你的答案、理由和解答过程、程序的运行方法、以及实验结果；
  - 代码：一个名为 code 的文件夹，内含解答过程中必要的代码和数据文件，包括源代码和可执行代码（如果有的话）；
  - 支持库：一个名为 support 的文件夹，内含在你的代码中用到但不是你亲自实现的支持性素材，包括开源代码或库等；如果这个文件夹下的内容超过 10M，请不要提供素材本身，而是提供下载地址，并注明安装和操作方法。
- 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
audioread	MATLAB 标准	将音频文件中的数据读入内存
plot	MATLAB 标准	绘制波形
scatter	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. 北邮人 bt

7 附录

以下给出手动拼接视频示例，当然你可以选择自动拼接或其它软件进行拼接。

1. 从`https://lv.ulikecam.com/?_s=1&keyword=JianYingex`下载剪映并安装，选择剪映是因为剪映是一款轻量级的视频剪辑软件。
2. 打开剪映如图3所示，选择开始创作后如图4所示。

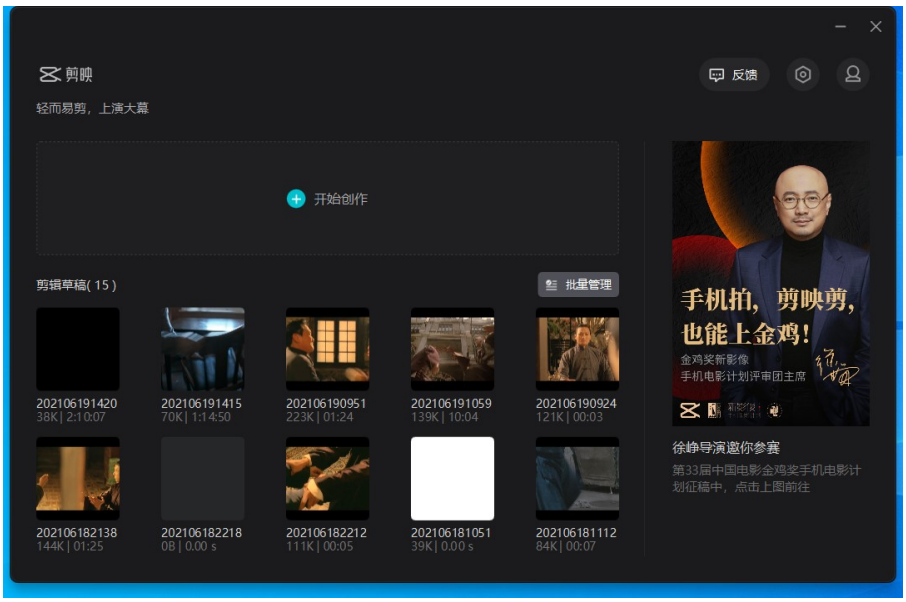

图 3: 开始界面

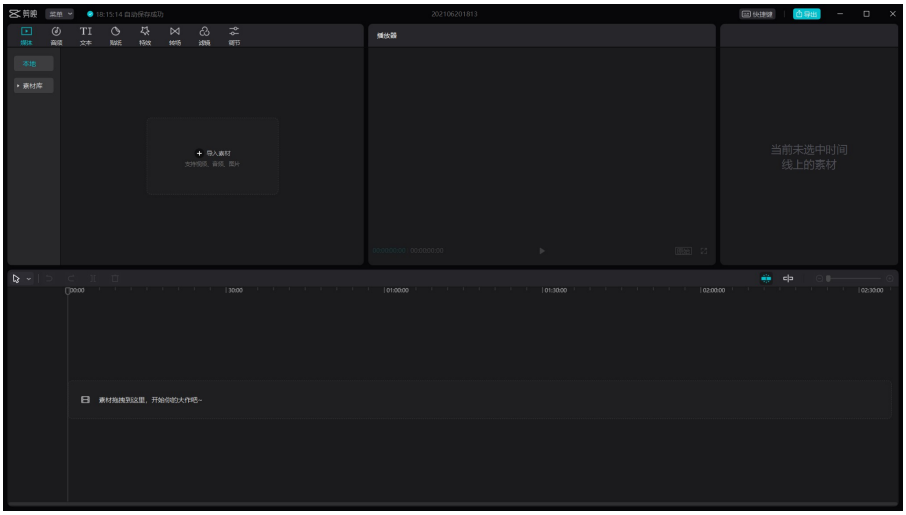

图 4: 创作界面

3. 选择导入素材，导入视频素材与音频文件后如图5所示。

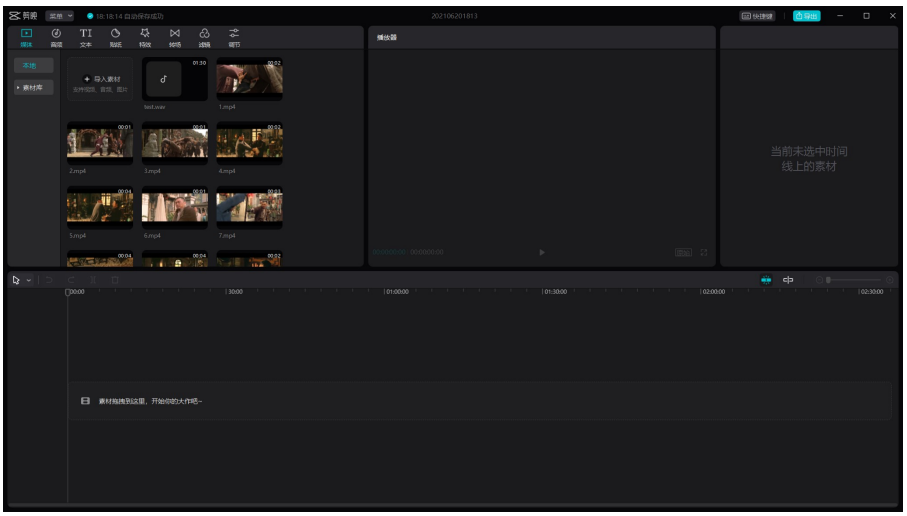

图 5: 导入素材后

4. 将音频文件拖入下方时间轴中，将视频素材按得到的 $I_1, I_2, \ldots$ 顺序依次拖入下方时间轴中。结果如图6所示。

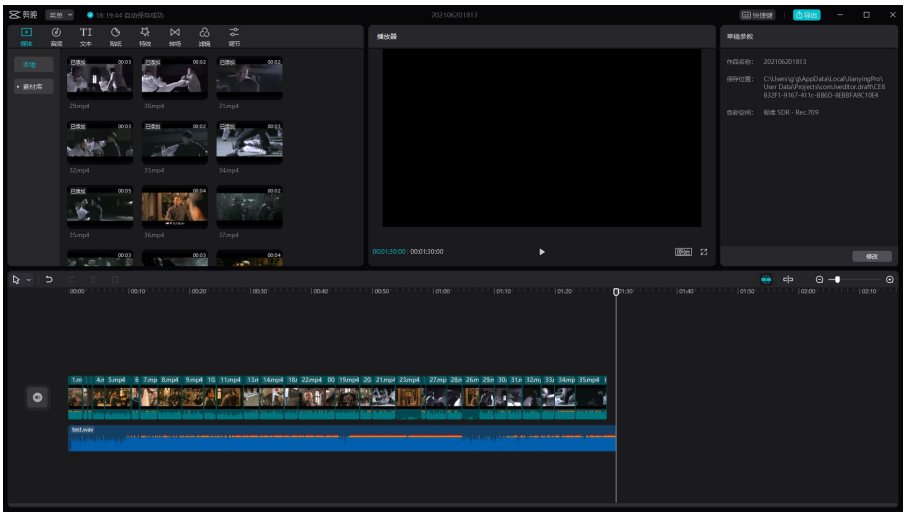

图 6: 剪辑界面

5. 在下方时间轴选中'test.wav'，拖动其最右方使音频与视频长度对齐并在整体界面的右上部分可以调整音量大小。处理后如图7所示.

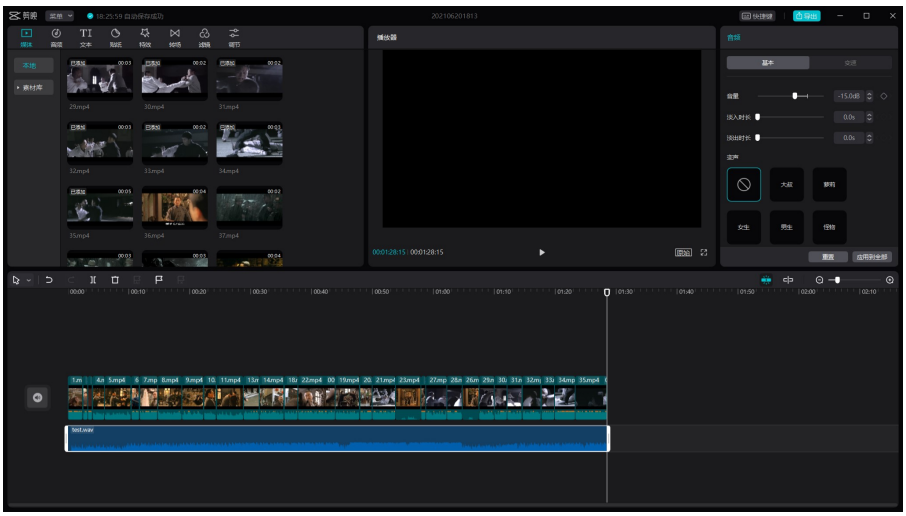

图 7: 对齐并调整音量大小后

6. 右上角导出文件。

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