2021 春信号与系统大作业之“B 站，我来了 !”

谷源涛

2021 年 6 月 19 日初稿，26 日定稿

本次大作业研究混剪视频的自动制作方法。

混剪视频从一部或多部电影或游戏 CG 中节选若干片段组合并配以背景音乐，是艺术再创作。混剪视频往往采用节奏感强烈的背景音乐和精心选取的代表性素材片段，在短短几分钟内给观看者明显的感官冲击力。

1 问题建模

记背景音乐为 bgm ; 记第 i 个电影为 $m^{(i)}$，用 $m^{(i)}_{s(i),e(i)}$ 表示第 i 个电影从 $s(i)$ 秒开始到 $e(i)$ 秒结束的片段 ; 用 $m = \mathrm{Join}(a, b, c, \cdots)$ 表示将电影片段 $a, b, c, \cdots$ 按顺序组合成拼接电影 m ; 用 $\mathrm{Impact}(m, \mathrm{bgm})$ 表示将拼接电影 m 和背景音乐 bgm 叠加在一起形成的冲击力，则视频混剪问题可以建模为最优化问题 [1]

$$\min_{\mathcal{S}} -\mathrm{Impact}\left(\mathrm{Join}\left(m^{(I_1)}_{s(I_1),e(I_1)}, m^{(I_2)}_{s(I_2),e(I_2)}, \cdots\right), \mathrm{bgm}\right), \tag{1}$$

其中

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

表示所有未知变量构成的集合。显然问题(1)求解非常困难，所以 B 站上的混剪视频一般由 up 主手工制作。

困难之一在于函数 $\mathrm{Impact}(m, \mathrm{bgm})$ 没有解析形式的定义。一般来说，拼接电影 m 和背景音乐 bgm 的“节奏感”越吻合，$\mathrm{Impact}(m, \mathrm{bgm})$ 越大。常见的人工处理的混剪视频按“节奏感”的吻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，典型如 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，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，虽然两者往往是一致的。

[1] $\min_{x\in\mathcal{X}} f(x)$ 是最优化问题的数学记法，其含义是对所有 $x \in \mathcal{X}$ 寻找函数 $f(x)$ 的最小值 ; 另外，$x^* = \arg\min_{x\in\mathcal{X}} f(x)$ 表示当 $x = x^*$ 时 $f(x)$ 取得最小值。

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，后文将具体研究冲击力的解析定义。

除了函数 $\mathrm{Impact}(m, \mathrm{bgm})$ 太玄之外，问题(1)还是既有连续变量又有离散变量的混合优化问题，一般非凸，求解非常困难。

为方便大二同学解决，我们将（基本）作业简化为：给定一个背景音乐 bgm 和 K 个电影片段 $\{m^{(i)}\}_{i\in[1:K]}$，且满足

$$\sum_{k=1}^{K} \mathrm{Len}\left(m^{(i)}\right) \geq \mathrm{Len}(\mathrm{bgm})$$

其中 $\mathrm{Len}(\cdot)$ 表示取序列长度，只要选取最合适的片段按最合适的顺序拼在一起就好，即问题(1)简化为

$$\min_{I_1,I_2,\cdots\in[1:K]} -\mathrm{Impact}\left(\mathrm{Join}\left(m^{(I_1)}, m^{(I_2)}, \cdots\right), \mathrm{bgm}\right). \tag{2}$$

这是一个整数规划问题，有多种方法可以求得次优解。

注意问题(2)中，假设一共有 $\bar{K} \leq K$ 个电影片段被选中，一般不满足 $\sum_{k=1}^{\bar{K}} \mathrm{Len}\left(m^{(i)}\right)$ 严格等于 $\mathrm{Len}(\mathrm{bgm})$，但这个问题不严重，我们暂且忽略。

2 节奏点提取

节奏点的强弱是体现音乐“激烈”程度的重要表征，本节首先讨论节奏点的定量估算方法。

我们借鉴背景材料 [1] 第 24-27 页的提取节奏点的方法，分为五步处理。记音乐信号为 $x(n)$。

1. 幅度平方求能量：

$$y_1(n) = x^2(n). \tag{3}$$

2. 加窗平滑得包络：

$$y_2(n) = \sum_{j=0}^{M-1} w_j y_1(n-j), \tag{4}$$

其中 $\{w_0, w_1, \cdots, w_{M-1}\}$ 表示窗函数，即一个 M 阶 FIR 滤波器的系数，请参考小学期 MATLAB 教材和课件 [3]。

3. 差分提取变化点：

$$y_3(n) = y_2(n) - y_2(n-1). \tag{5}$$

4. 半波整流取正值：[2]

$$y_4(n) = \max\{y_3(n), 0\}. \tag{6}$$

[2]换成全波整流（即取绝对值）也差不多，同学们可以试试看。

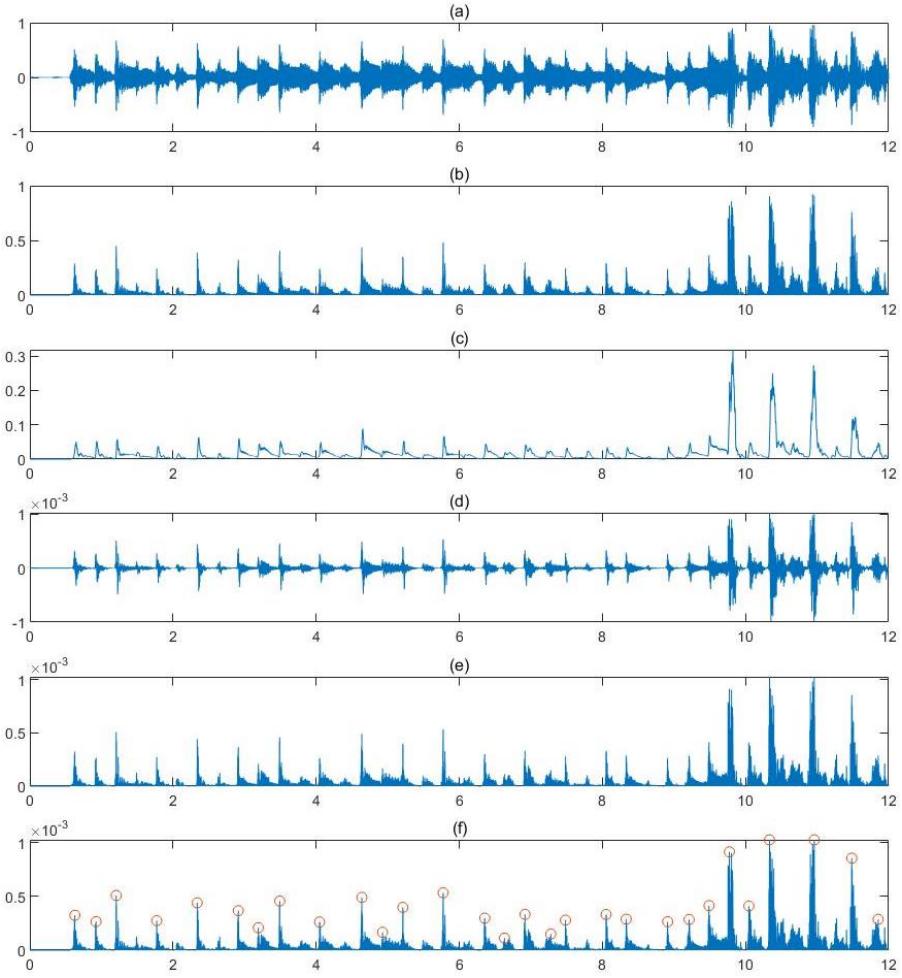

图 1: 音乐信号及其节奏点提取示例：(a) 原始音乐信号；(b) 幅度平方求能量；(c) 加窗平滑得包络；(d) 差分提取变化点；(e) 半波整流取正值；(f) 自动选峰定节奏。

5. 自动选峰定节奏：

$$y(n) = \cdots , \tag{7}$$

其中 $y(n)$ 表示 n 时刻节奏点的强度，强度为零即非节奏点。可以由 $y_4(n)$ 的局部极大值判定节奏点，但由于音频长时相关特性以及各种非理想因素，$y_4(n)$ 中存在大量非节奏点的峰值必须予以剔除。这是最开放的环节，可以采用很多启发性方法，如按照相邻两峰间隔不小于某阈值、能量变化不小于某阈值等准则去除非节奏点峰值。

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。观察子图 (f) 中的红色圆圈，感觉确实正确提取到大部分节奏点。

从音乐中提取节奏点还是一个尚未全部解决的开放问题，感兴趣的同学可完整阅读背景材料 [1]。

3 激烈度度量

节奏点找到后，接下来就可以由节奏点强度序列 $y(n)$ 平滑为音乐的“激烈度”序列 $z(n)$，

$$z(n) = \sum_{j=-N/2}^{N/2} h_j y(n-j), \tag{8}$$

其中 $\{h_{-N/2}, h_{-N/2+1}, \cdots, h_{N/2}\}$ 表示低通滤波器系数。图1(a) 中音乐信号的激烈度序列如图2所示。

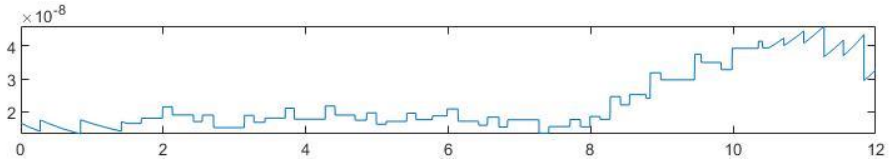

图 2: 音乐信号的“激烈度”序列示例。

上述方法可以评估背景音乐和电影片段里的音频的“激烈度”,分别记做 $Z(n)$ 和 $\{z_1(n), z_2(n), \cdots\}$，则问题(2)变为

$$\min_{I_1, I_2, \cdots \in [1:K]} -\text{Impact}\left([z_{I_1}(n), z_{I_2}(n), \cdots], Z(n)\right), \tag{9}$$

其中用 $[\cdot]$ 表示将多个序列按顺序拼在一起。注意 Impact 的定义发生了变化，但不影响继续讨论。

接下来我们假设 Impact 函数对序列长度满足可加性，即对任意序列 $a(n), b(n), c(n), d(n)$，如果

$$\text{Len}(a(n)) = \text{Len}(b(n)), \quad \text{Len}(c(n)) = \text{Len}(d(n)),$$

则有

$$\text{Impact}\left([a(n), c(n)], [b(n), d(n)]\right) = \text{Impact}\left(a(n), b(n)\right) + \text{Impact}\left(c(n), d(n)\right).$$

在这个假设下，问题(9)可以进一步变形为

$$\min_{I_1, I_2, \cdots \in [1:K]} -\sum_{k=1}^{\bar{K}} \text{Impact}\left(z_{I_k}(n), Z_{\mathcal{I}}^{(k)}(n)\right), \tag{10}$$

其中 $\mathcal{I}$ 表示候选下标序列 $I_1, I_2, \cdots$，序列集合 $\left\{Z_{\mathcal{I}}^{(k)}(n)\right\}_k$ 表示 $Z(n)$ 的 $\mathcal{I}$ 划分，即

$$Z(n) = \left[Z_{\mathcal{I}}^{(1)}(n), Z_{\mathcal{I}}^{(2)}(n), \cdots\right], \tag{11}$$

且 $\text{Len}\left(Z_{\mathcal{I}}^{(k)}(n)\right) = \text{Len}(z_{I_k}(n)), \forall k \in [1:\bar{K}]$。注意(10)中求和与(11)中划分的项数都是 $\bar{K} \leq K$，很快会发现存在不等号的事实将显著增加问题的复杂性。

继续简化问题，我们用 $\text{Intensity}(z(n))$ 表示从激烈度序列 $z(n)$ 到激烈度标量的映射。具体的，用

$$p_{I_k} = \text{Intensity}(z_{I_k}(n)), \tag{12}$$

$$q_k = \text{Intensity}\left(Z_{\mathcal{I}}^{(k)}(n)\right), \tag{13}$$

分别表示各个电影片段和背景音乐划分片段的激烈度序列的激烈度标量，则问题(10)可以进一步简化为

$$\min_{I_1,I_2,\cdots\in[1:K]} -\sum_{k=1}^{\bar{K}} \text{Impact}\left(p_{I_k}, q_k\right). \tag{14}$$

再次强调，我们知道 Impact 的含义发生了改变，但不介意混淆记法。

认真观察问题(14)，你想到了什么？

.....

我想到了“信号与系统”第六章 [2] 讲的“相关系数”或者“内积”!

即用排列后电影片段的“激烈度”标量构成的序列和背景音乐的“激烈度”序列的相关性的负值度量两者叠加所形成的冲击力，从而将问题(14)具体化为

$$\min_{I_1,I_2,\cdots\in[1:K]} -\frac{\sum_{k=1}^{\bar{K}} p_{I_k} q_k}{\left(\sum_{k=1}^{\bar{K}} p_{I_k}^2\right)^{1/2}\left(\sum_{k=1}^{\bar{K}} q_k^2\right)^{1/2}}, \tag{15}$$

其中激烈度标量采用激烈度序列的二范数，即

$$\text{Intensity}(z(n)) = \left(\sum_n z^2(n)\right)^{1/2}. \tag{16}$$

继续努力化简：结合(13)和(16)，发现

$$\left(\sum_{k=1}^{\bar{K}} q_k^2\right)^{1/2} = \left(\sum_{k=1}^{\bar{K}} \left(Z_{\mathcal{I}}^{(k)}(n)\right)^2\right)^{1/2} \approx \left(\sum_n Z^2(n)\right)^{1/2} = \text{constant}.$$

利用这个性质最后一次简化问题(15)，得到

$$\min_{I_1,I_2,\cdots\in[1:K]} -\sum_{k=1}^{\bar{K}} p'_{I_k} q_k, \tag{17}$$

其中

$$p'_{I_k} = \frac{p_{I_k}}{\left(\sum_{j=1}^{\bar{K}} p_{I_j}^2\right)^{1/2}} \tag{18}$$

表示候选子集上的归一化激烈度标量。

模型很难再简化了，下面就要求解问题(17)。

4 求解方法

直观的看，问题(17)是从 p' 序列中找一个子集的排列，希望和 q 序列的内积最大。这个简化的问题似乎没有多项式解法，注意同时还存在困难：子集的选取决定 p' 序列和 q 序列的值。

4.1 全局搜索法

Step0 预处理：计算 $Z(n)$ 和 $p_1, p_2, \cdots, p_k$；

Step1 从 K 个视频片段中选择若干个；

Step2 若选中的视频总长度和 bgm 的长度相差过大，返回 Step1 重新选择；

Step3 记选中的视频片段数量为 $\bar{K}$，选择这些视频下标集的一个排列 $\mathcal{I}$；

Step4 计算 $p'_{I_1}, p'_{I_2}, \cdots$ 和 $q_1, q_2, \cdots$；

Step5 记录 $-\sum_{k=1}^{\bar{K}} p'_{I_k} q_k$，返回 Step3 遍历下一个排列；

Step6 返回 Step1 遍历下一个选择；

Step7 从全部记录值中找到最小值，其对应的排列即为最优的 $I_1, I_2, \cdots$。

这种方法遍历所有选择和所有排列，视频片段太多的话复杂度过大不可取。

4.2 次优解法

观察问题(17)，可见一个难点是：只有下标集 $\mathcal{I}$ 确定了，才能计算 $q_1, q_2, \cdots$，原因是各个视频片段不等长。为了绕过这个难点，我们首先假设各个视频片段的长度不会差别太大，然后考虑用均值近似，记

$$\bar{L} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{Len}(z_k(n))$$

表示视频片段的平均长度，然后按 $\bar{L}$ 对 $Z(n)$ 进行均匀划分，然后可计算出平均意义下 bgm 的激烈度变化趋势

$$q_1, q_2, \cdots.$$

然后再从可直接计算得到的 $p_1, p_2, \cdots, p_k$ 中选择一个子集排列使得和 q 序列的内积最大。

如何选，随机挑 100 次，找一次最好的？

4.3 其它解法

肯定还有其它（更好的）方法，留待同学们探索。

5 作业和评分规则

5.1 作业

我们为大家精心准备了背景音乐“test.wav”和 $K = 53$ 段视频素材“1.mp4”, “2.mp4”, $\cdots$, “53.mp4”, 请同学们按顺序解答下述题目。

5.1.1 提取节奏点

1. 对“test.wav”的前 12 秒进行节奏点提取的前四步处理, 即公式(3)至(6)。调整(4)中窗函数的类型与长度, 对比能量包络的差异, 并**在报告中进行讨论**。固定一种窗函数与长度, **计算半波整流结果并在报告中绘出**。

2. 在上一问的基础上完成节奏点提取的最后一步处理, 设计自动选峰算法提取前 12 秒节奏点, **在报告中阐述设计思想和描述算法步骤, 并绘制前 12 秒的节奏点**。

5.1.2 度量激烈度

1. 参考(8)设计合适类型与长度的低通滤波器计算“test.wav”的激烈度 $Z(n)$, **在报告中介绍滤波器设计, 并绘制类似图2的“激烈度”序列**。

2. 参考(16)计算“1.mp4”、$\cdots$、“5.mp4”等五个视频的“激烈度”标量 $p_k, k = 1, \cdots, 5$, **在报告中给出 5 个视频的“激烈度”排序**。

5.1.3 生成混剪视频

参考前文设计一种算法求解(17),使用“test.wav”作为背景音乐,选择视频素材“1.mp4”, “2.mp4”, $\cdots$, “53.mp4”, 生成不短于 80 秒的混剪视频并**在报告中介绍算法并讨论其优缺点**。

由下标集生成混剪视频的方法请参考附录。

5.2 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），代码需包含必要的注释；
  - 实验结果：一个名为 result 的文件夹，内含最终完成的剪辑视频以及按顺序存放素材文件名的文件 input.txt；
  - 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。

6 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 [3]。

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。

7 致谢

1. 首先感谢大四学生金澄！他收集准备的素材，编程探索了提取节奏点、度量激烈度、拼接视频的完整过程，证实了本次大作业的可行性，还撰写了第二章和第五章的初稿、以及介绍手动拼接方法的附录；最重要的，他是设计大作业过程中的主要讨论者，没有金澄的参与就不会有本次大作业。

2. 感谢助教博士生孟令航、直博生张振威和助教博士生焦宇晨。令航和振威参与了设计大作业之初的讨论，调研了节奏点提取等技术进展；振威还实现了自动拼接视频的工具；宇晨帮助阅读了大作业终版的文本，检查建模合理性，并参与了解决方法的讨论。

3. 金澄希望感谢北邮人 bt。

8 附录

8.1 自动拼接视频和叠加背景音乐

我们提供了自动拼接的脚本程序及其使用教程（GitHub）或者（代理），可根据待合并的视频片段名称列表文件 input.txt 自动生成混剪视频。

8.2 手动拼接视频和叠加背景音乐

以下给出手动拼接视频的方法示例，当然你可以选择其它软件进行拼接。

1. 从剪映官网下载剪映并安装。
2. 打开剪映如图3所示，选择开始创作后如图4所示。

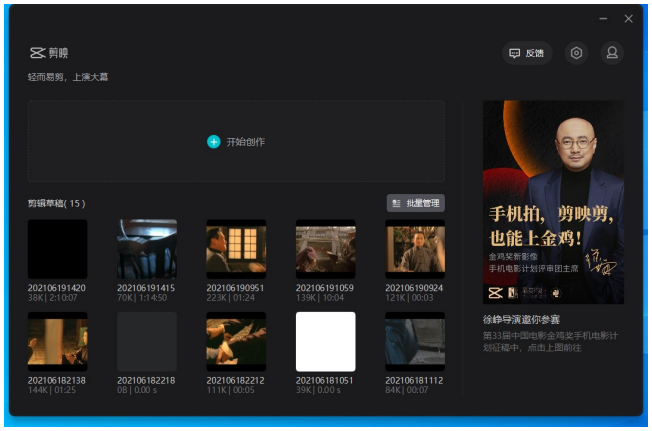

图 3: 开始界面

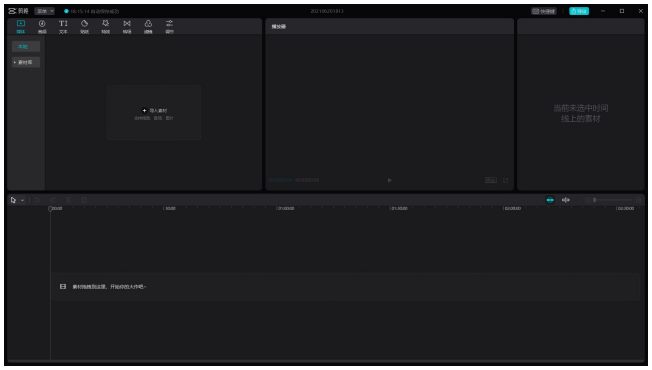

图 4: 创作界面

3. 选择导入素材，导入视频素材与音频文件后如图5所示。

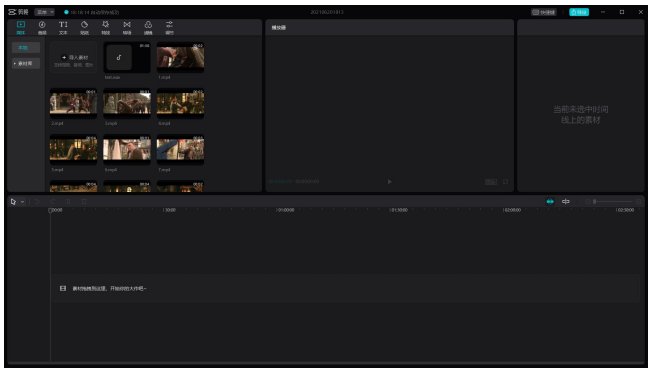

图 5: 导入素材后

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

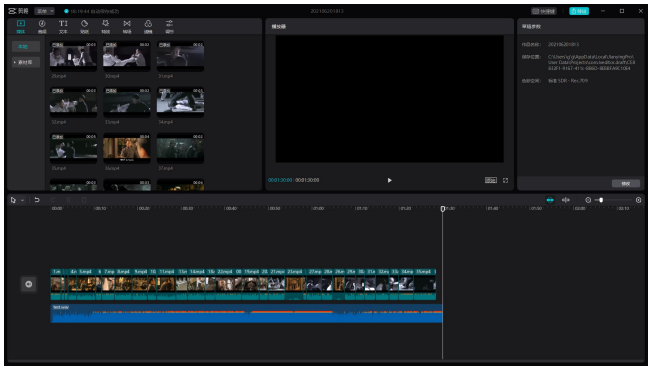

图 6: 剪辑界面

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.

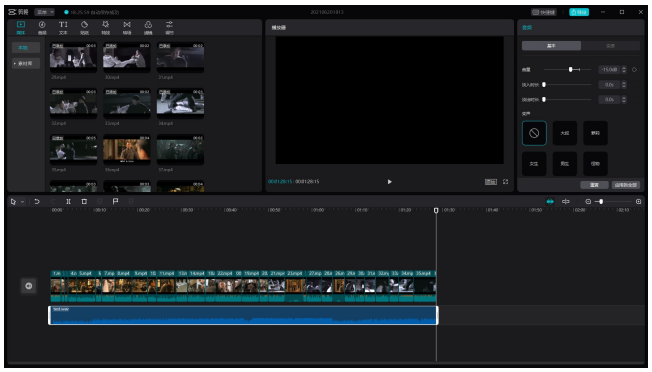

图 7: 对齐并调整音量大小后

6. 右上角导出文件。

参考文献

[1] Meinard Müller, Christof Weiss, Stefan Balke, Tempo and Beat Tracking, Tutorial: Automatisierte Methoden der Musikverarbeitung 47. Jahrestagung der Gesellschaft für Informatik

[2] 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

[3] 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