

关于我两年前转身成为改进字幕菌后的那些事

两年前，突然产生了一个收藏动画 BDrip 的想法。论坛里下了个字幕，发现跟视频不适配。为了适配，我下海去学习了一下如何修改字幕，然后发到了论坛上。无独有偶，观看时候发现 10 多年前的字幕多多少少都会出现一些问题，导致体验不太好，然后就踏上了改进字幕的道路。作为社畜，平时也没太多时间去弄。能让我把兴趣转为源源动力去改进的，只因字幕的炫酷特效。

时光匆匆，改进字幕也改了两年了。接下来就说下这一年来我领悟到的一些小技巧。希望能对你有所帮助。

一、查找篇

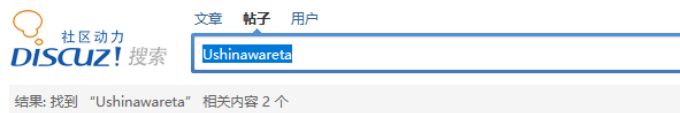
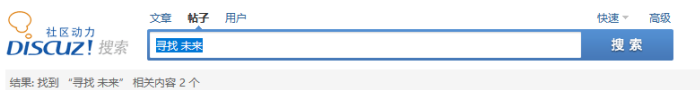
1. 字幕查找

a) 本论坛里找字幕

因为 acgrip.com 是论坛式的字幕分享网站，不是展现式的字幕下载网站，加上论坛里有来自境内外的各路字幕大佬，繁简字幕混杂，帖子标题并没有很好的格式统一。所以，**搜索时候不要通过全标题去搜，要用标题中的某个关键字去搜。搜不到就用繁体关键字、罗马音、英文等去搜。**

例如，搜寻“寻找失去的未来”，直接搜只会出现一个帖子。但是，搜索“寻找 未来”会出现 2 个帖子。搜索罗马音“Ushinawareta”，也会出现 2 个帖子。

所以，换个关键字再搜一下，你可能会发现更多的字幕任你挑选。



b) 字幕整合包

站内 Tonyhsie 大神提供了一个 B 站字幕整合包，里面有 1000 多部动画的字幕。[帖子在此](#)，他会不时季度更新，收藏一下那帖子吧。

还有一个是 DHR 字幕组的整合包，[帖子在此](#)。

2000 多部动画外挂字幕磁力链接(2014 年之前的, 很多断源的老番都有): magnet:?xt=urn:btih:73be4f4e39745bf8493d7bd16b131af579fb9a53
秒传: 79BF805ED26D141449D8E09621C0A262#B669DF597627E501210D976B036D0051#377344#动画字幕索引.xls
427D2D4D2EA61FF5237AED41B4DBA556#49495F19D2908F161558D1ED878FEAFB#1971969019#动画字幕.rar
网盘: <https://www.mediafire.com/file/zl82roiizqf0wpz>
<https://www.mediafire.com/file/dn212lteqjfe97b>

c) 其他容易进的中文动画字幕网站

因为论坛[某些原因](#), 论坛里不收录“漫游字幕组”的字幕。所以该组相关的可以到[该组的网站](#)去下。

漫游字幕组: <https://sub.popgo.org/>

喵喵字幕组: <https://github.com/orgs/NekomoeKissaten-SUB/projects/1>

北宇治字幕组: <https://github.com/orgs/Kitauji-Sub/projects/2>

SweetSub: <https://github.com/SweetSub/SweetSub>

✿拨雪寻春✿: https://github.com/HaruhanaSub/Haruhana-Fansub_Source

MingYSub: <https://github.com/MingYSub/SubsArchive>

冷番补完字幕组: <https://coolfanssub.gitbook.io/lfbw-sub>

蛇大(Snnake)参与过制作的字幕 (雪飘为多): <https://subbers.org/subtitles/>

咪梦动漫组 (貌似已停更): <https://github.com/DMYJS/MMSUB>

轻之国度字幕组 (貌似已停更): <https://github.com/qiusj759039257/LKSUB>

异域-11 番小队 (非官方库): <https://github.com/sommio/11Fans-Subtitles>

伪射手: <https://secure.assrt.net/>

SubHD: <https://subhd.tv/>

字幕库: <https://zimuku.org/>

天雪论坛字幕板块: <https://www.skyey2.com/forum.php?mod=forumdisplay&fid=75>

HDClub 精研视所 (繁体): <https://www.hd.club.tw/forum-102-1.html>

opensubtitles: <https://www.opensubtitles.org/en/search/sublanguageid-chi,zht,zhe>

kitsunekko.net (日文字幕库): <https://www.kitsunekko.net/dirlist.php?dir=subtitles%2Fjapanese%2F>

2. 字体查找

a) 超级字体整合包

字幕一般都会用到一些字体。如果你不使用对应的字体, 很可能就会发生字幕位置乱套, 严重影响体验。说真的, 不使用对应的字体去使用字幕, 是对字幕制作者的一种不尊重。人家辛苦做了一个美观的字幕出来分享, 不使用对应字体来看的话, 字幕可能就变成一坨翔了。某些人可能对一坨翔的字幕并不在意, 但是字幕制作者想分享的并不是一坨翔, 而是一份美观的字幕作品。请不要糟蹋字幕制作者的苦心的努力。

强力推荐下载[“超级字体整合包 XZ”](#)。一包在手, 字体不愁 (几乎)。磁力链接[在这里](#)。下载里面的“完整包”就行。虽然几十个 G 容量有点大, 但是, 字幕需要的字体绝大部分都能在里面找到, 不用每次重复下, 还是很值得的。下载下来的字体并不需要安装, 只需用字体加载工具 [FontLoaderSub](#) 之类的加载就行, 本文的“字体篇”会提及。

另外, 可以补充一下字体 X 包里有而 XZ 字体包没有的字体, 帖子[在这里](#)。里面含有一些老旧字幕需要的字体。

b) 字体下载网站

如果“超级字体整合包 XZ”里找不到，那就要到别的地方去找了。可以到论坛对应字幕的帖子内看看有没有热心群众提供了链接，如果没有，可以到字体下载网站去找。但是，找到了并不一定能够免费下载。最后的手段就是直接百度了。提供几个我常用的网站。

求字体: <https://www.giuziti.com/>

字客网: <https://www.likefont.com/> 或 <https://www.fontke.com/>

方正字库 (改版了, 需要用客户端才行):

<https://www.foundertype.com/index.php/FindFont/index.html>

c) 通过字幕组发布的种子里附带的字体包

因为字体网站多多少少会有这有那的限制, 导致下载不了。这时候, 可以通过字幕组发布种子附带的字体包来下载所需的字体。怎么找到某个种子里有某个字体呢? 这就有点技巧了。

(i) 通过论坛的全局搜索

先在论坛里随便搜一下, 出现搜索页面时候点击“高级”, 填入字体名称然后勾选全文进行论坛全局搜索。见下图。然后你会发现好几部动画都用到了这个字体, 然后你就可以去找该字幕组发布的外挂字幕的 BD 种子 (一般是 mkv 格式的), 里面就会有所需的字体。

例如, 全局找“HYXuanSong 85S”, 发现喵喵的“派对浪客诸葛孔明”里面有带这个字体, 然后去 BT 网站找该对应动画的种子把字体包下载下来就行。

The screenshot shows the Discuz! forum search interface. At the top, there are tabs for '文章' (Articles), '帖子' (Posts), and '用户' (Users). A search bar contains the text 'b站'. To the right of the search bar, there are buttons for '快速' (Quick) and '高级' (Advanced). The '高级' button is circled in red with a '1' next to it. Below the search bar, there is a section for '帖子高级搜索' (Post Advanced Search). In this section, the '关键词' (Keywords) field contains 'HYXuanSong 85S', which is circled in red with a '2' next to it. There is also a checkbox for '全文' (Full Text) which is checked. Below this, there are various filters for '主题范围' (Topic Range), '特殊主题' (Special Topics), '搜索时间' (Search Time), '排序类型' (Sorting Type), and '搜索范围' (Search Range). The '搜索范围' dropdown menu is open, showing a list of forum categories. The '全部版块' (All Sections) option is selected and circled in red with a '3' next to it. At the bottom of the search filters, there is a '搜索' (Search) button, which is also circled in red. Below the search filters, the search results are displayed. The first result is titled '金装的维尔梅~ 濒临堕落的魔法师和最强的灾厄一起冲入魔法世界~ / 金装のヴェルメイユ~ 摩っぶち魔術師は最強の厄災と魔法世界を突き進む~ / Kinsou no Vermeil'. The second result is titled '派对浪客诸葛孔明/派对咖孔明/Paripi Koumei/パリピ孔明'. This result is circled in red. Below this result, there is a link to '重啓人生: 大逃亡 / エスタブライフ グレイトエスケープ / Estab-Life: Great Escape'. At the bottom, there is a link to '扶桑花之舞/フラ・フラダンス/Hula Fulla Dance'.

网盘: <https://www.mediafire.com/file/lnb34ph99vzy9ok>
论坛里还有推荐 nexusfont、FontExpert 的, 我没用过。
nexusfont: <https://bbs.acgrip.com/thread-3437-1-1.html>
FontExpert: <https://bbs.acgrip.com/thread-6153-1-1.html>

3. 字体识别

能用文字图片识别字体字型的, 暂时发现只有这 3 家。

求字体: <https://www.qiuziti.com/>
字客网: <https://www.likefont.com/>
站长字体: <https://font.chinaz.com/zhaozi/>

4. 字体修正

因为 Win7 及以上版本操作系统问题, 会导致某些老旧的字体安装或使用不了。站内苏辰汐大佬做了个字体修正包。建议用修正过的字体替换掉 XZ 包里面对应的字体。

字体修正包: <https://bbs.acgrip.com/thread-8793-1-2.html>

5. 日文字体对应字型相近的中文字体

很有用的对应字型对应表, 做字幕的话建议收藏。(看看巨佬 [Jalapeño](#) 发的主题帖, 你就会自觉 Orz)

Fontworks 日文字体 中日字体匹配方案推荐: <https://bbs.acgrip.com/thread-3059-1-1.html>

DynaFont 日文字体 中日字体匹配推荐表: <https://bbs.acgrip.com/thread-3824-1-1.html>

Morisawa 日文字体 中日字体匹配推荐表: <https://bbs.acgrip.com/thread-3823-1-1.html>

日本三大厂圆黑宋楷仿 中日字体匹配: <https://bbs.acgrip.com/thread-5052-1-1.html>

新人 OPED 中日字体样式推荐: <https://bbs.acgrip.com/thread-5358-1-1.html>

6. 字体相关的其他问题

同一字体的不同版本可能存在字型及大小差异: <https://bbs.acgrip.com/thread-9396-1-1.html>

相对与 XZ 包里的, 新版的方正准圆会把日文也囊括, 建议更新一下: <https://bbs.acgrip.com/thread-9331-1-1.html>

三、字幕工具篇

1. Aegisub - 字幕编辑工具

官网停更好几年了, 最后一版是 2014 年的发布的。老版有一些奇怪的 bug, 所以建议使用其他人持续维护的版本。

通常对于非字幕组的人而言, 我建议使用非定制非魔改版, 这样做出来的字幕通用性比较好。做出来含自动化代码的字幕别人也可以正常运行。建议使用 wangqr 维护的版本:

<https://github.com/wangqr/Aegisub/releases>

这里会有一个问题, 就是新版的 Aegisub 更新了 Lua 的解析引擎, 导致某些语法有更改。最容易出现坑人的地方就是[括号问题](#)。

自动化代码大部分都包含这一句:

```
cn = _G.unicode.len(orgline.text_stripped:gsub(" ", ""))
```

旧版的 Aegisub 运行没有问题, 而用了新 Lua 引擎的就会报错。

需要加个括号修改一下, 就能兼容运行:

```
cn = _G.unicode.len((orgline.text_stripped:gsub(" ", "")))
```


我的回帖也引用了 Aegisub 维护者的解决方案：

<https://bbs.acgrip.com/forum.php?mod=redirect&goto=findpost&ptid=9237&pid=87877>

楼主说老版的 Aegisub 会有内存泄漏问题，往上翻就能看到主楼的说明。

教程的话，百度或 B 站搜一下，一大堆，随便找个学习一下就好。我只是懂个皮毛，现阶段来说足够了。

老版的 Aegisub 自带的 AviSynth 有 bug，导致各种奇怪的崩溃问题。如果使用 AviSynth 做 Aegisub 的处理模块的话，建议更新一下，把对应系统版本的 AviSynth.dll 扔到 Aegisub 的目录就行了。下载地址：

<https://github.com/AviSynth/AviSynthPlus/releases>

2. ListAssFonts - 字幕字体分析工具

论坛里大佬 tonyhsie 的力作，集列出字幕所需字体、检查缺字、还原子集化字体名等多种功能为一体的牛逼小工具。自己看帖内说明吧：<https://bbs.acgrip.com/thread-1894-1-1.html>

3. Sushi - 频谱比对型的字幕调轴工具

Sushi 是通过比对两个不同音频的频谱的相似度来进行字幕调轴的工具。简单来说就是通过音频频谱 A 去偏移匹配音频频谱 B，得到一个偏移量，然后通过这个偏移量重新计算字幕的时间轴。调轴的精准度相当高，虽然说程序有点 bug 会导致偶尔会出现调轴失败。

配合 tonyhsie 大佬写的 SushiHelper，使用时候会更方便。SushiHelper 的操作详情请看帖子：

<https://bbs.acgrip.com/thread-7987-1-1.html>

扔一个打包好的 [sushi+sushiHelper+Alass+ffsubsync+Alass FFsubsync Helper](#)，x64 的 Windows 开箱即用。

秒传：

D6A82CEAF78054FE90760692A456D7D2#ECE00BB76A297F1B3D680D53008DA510#91950616#SS_2023.z

网盘：<https://www.mediafire.com/file/5rn75tcstnt86i>

4. Alass、ffsubsync - 频谱波形分析型的字幕调轴工具

先说明一点，如果你处理后不进行完整地字幕检查（完整地观看一遍）和后续处理，我不建议使用这两个工具进行调轴，因为一般说来，这两个工具调出来的时间轴位置卡的太紧，没有提前和延后的缓冲时间。导致一行字幕出现时候会有闪动现象，消失时候也略显过快，观看的体感很不好。

Alass 和 ffsubsync 的原理应该是类似的。通过分析频谱的波形，前后卡时间点的位置，然后把某行字幕塞进这个时间位。Alass 是分区段进行平移调轴的，字幕行数要匹配频谱分析出时间段的个数，否则就会乱套。ffsubsync 是全局分析的，只会全字幕进行相同的平移调整。因为这两个工具不需要一个参照音频进行比对，所以对那种缺少参照视频或音频的字幕来说，可能是一个方便的选择（这段原理说明是我在使用过程中分析发现的，并没有去研究其源代码，有可能不对）。

建议处理完用 Aegisub 的时间后续处理器处理一下。用 Alass 的话，开始提前 50~100ms（试了几个，体感 70ms 比较好），结束延后 50ms。用 ffsubsync 的话，一般来说，对于本来匹配精度较高的轴，不调也能接受，但也有些需要根据调出来的偏移时间来进行较大幅度的调整。

Alass 还有一个小功能，用来做带时间轴的空白字幕。下面的帖里有介绍。

Alass 以及 Alass FFsubsync Helper 相关的使用说明：<https://bbs.acgrip.com/thread-8753-1-1.html>

上一点的秒传已包含了这两个工具。

5. Mocha - 摄像机反求跟踪工具

这中文说明看上去感觉怪怪的，简单地说，就是通过影片来反向计算摄像机的运行轨迹的工具。

知道摄像机运行轨迹有什么用？啊，这可有用了。当前我用得最多的是用来做字幕屏幕字的动态跟随效果。

要把摄像机轨迹数据运用到字幕中，需要搭配 Aegisub 的插件使用。插件请看下一小节。

Mocha 跟踪字幕简单教程，看一遍就基本会用：<https://www.bilibili.com/video/av19838657/>

用 Aegisub-Motion 插件先将视频切成图片再扔到 Mocha 里跟踪，某些情况下比上面 Mocha 里打开视频跟踪成功率更高（不会出现错帧、花屏等情况）：

<https://cnlang.org/forum.php?mod=viewthread&tid=85161>

Mocha 高级点的技巧（12 集）：<https://www.bilibili.com/video/BV1BY41177EB/>

Mocha v10.0.0.934 单独版+汉化（解压码：vcb）：

秒传：

24558A3092BED1F3346FF015209BE444#76B26141328DAB52A076E38726BF797C#215972174#MP_v10.z

网盘：<https://www.mediafire.com/file/bq0pdr6eh5ohfjl>

6. Aegisub-Motion - 解析运动轨迹数据的 Aegisub 插件

这工具一般是配合 Mocha 导出的轨迹数据使用的。上一点 Mocha 相关的第一个视频就有教操作。

下载链接：<https://github.com/TypesettingTools/Aegisub-Motion/releases>

与之相关的，有一个叫 Aegisub-Perspective-Motion 的插件（这两个插件使用场景、功能不一样）。这个插件可以实现透视(perspective)变换。但是，导出的数据格式跟用 Aegisub-Motion 插件不同，这个需要输入 After Effects CC POWER PIN 格式的数据。

Aegisub-Perspective-Motion 的具体用法请看：<https://zahuczky.com/aegisub-perspective-motion/>

7. Extrapolate Tracking - 利用推断来补全轨迹数据的 Aegisub 插件

有时候会遇到某些运动是移出屏幕或淡出而导致无法成功跟踪前几帧或最后几帧，这时候 Extrapolate Tracking 插件就能发挥作用了。先用 Mocha+Aegisub-Motion 插件处理一遍，然后删掉前/后几帧跟踪失败的字幕行，接着选择正确跟踪的字幕行，运行 Extrapolate Tracking 插件，就能根据所选字幕行来推断补全前/后几帧的跟踪字幕。

具体用法请看：<https://phoscity.github.io/Aegisub-Scripts/Extrapolate%20Tracking/>

8. Hyperdimensional Relocator - 多功能重定位 Aegisub 插件

Hyperdimensional Relocator 插件功能很丰富强大，可以实现简单手动跟踪、字幕逐帧切割、多行字幕整体变换移动等功能。如果要跟踪的只是简单的几帧字幕，可以选择用这个插件去手动弄而不必打开 Mocha。某些 Mocha 跟踪不上的情况也可以用这个插件去手工弄。使用方法文字不好说明，看下面视频教程。

官方用法说明：<http://unanimated.hostfree.pw/ts/scripts-manuals.htm?i=1#relocator>

用法说明的中文翻译：<https://git.io/JYfFF>

常用几项功能的视频教程：<https://www.bilibili.com/video/av544823192/>

9. HYDRA - 多功能标签变换 Aegisub 插件

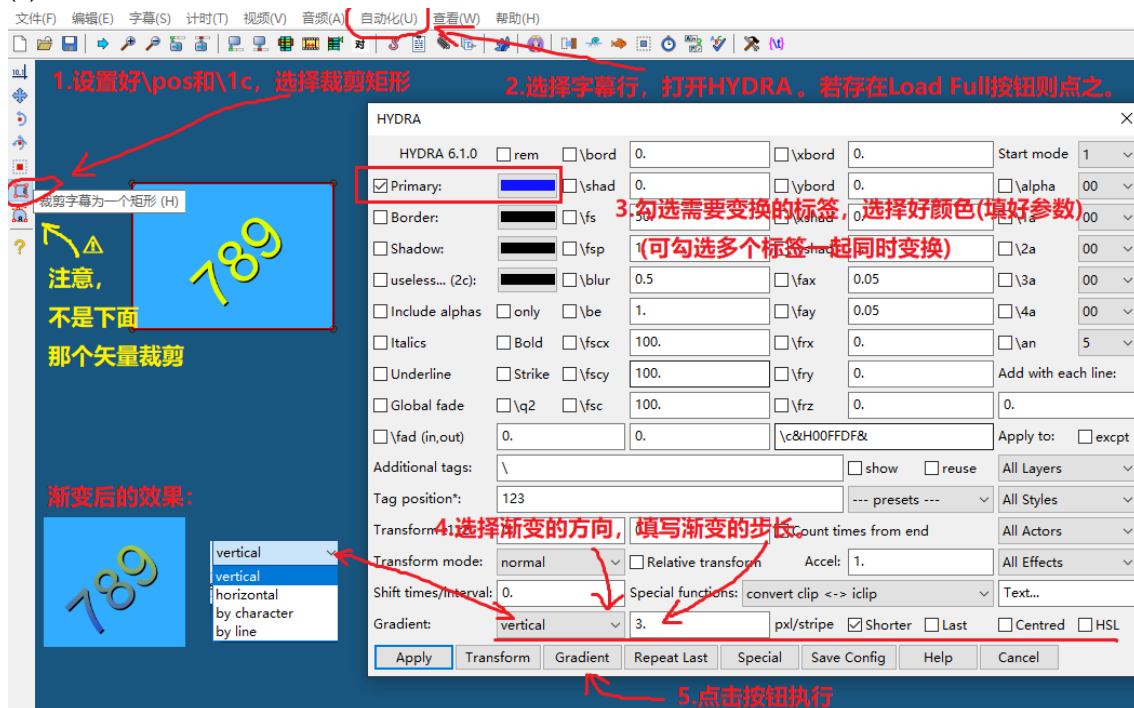
HYDRA，意思是多头蛇，看名字就知道又是一款功能很丰富的插件。可以处理标签变换、位置变换、颜色渐变、角度渐变等。Hyperdimensional Relocator 处理以位置为主的变换，而 HYDRA 处理的是颜色、角度等非位置相关的变换。

没发现有什么详细的中文视频介绍如何使用这个，这里简单说一下如何用 HYDRA 插件处理渐变。

- (i) 选择一行字幕，设置好位置\pos 和颜色\c，然后用矩形裁剪工具把需要渐变的位置裁剪出一个矩形的 \clip (\clip 的参数是 4 个逗号分隔的数值)。
- (ii) Aegisub 菜单栏选“自动化->HYDRA”，然后弹出对话框如果存在 Load Full 按钮则点 Load Full 加载全部功能，不存在则说明当前已经是全功能模式。
- (iii) 勾选需要进行变换的目标标签，当前做渐变，所以勾选 Primary（即\c），然后选择目标颜色。当然你

也可以再勾其它标签一起做变换，例如旋转的话就勾选\frz，然后填写目标角度。

- (iv) “Gradient:”旁边的下拉框可以选择渐变的方向，vertical 是纵向，horizontal 是横向，也可以逐字(by character)或按行(by line)，第一步需选择多行字幕来处理渐变。填写渐变的步长。
- (v) 点击“Gradient”按钮执行结果。



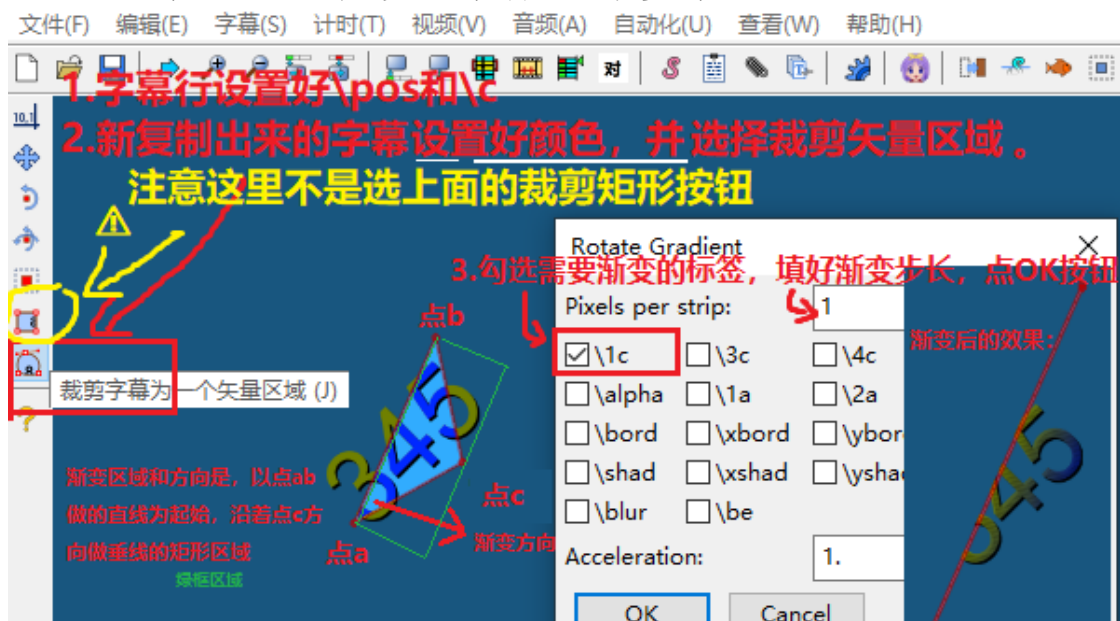
官方用法说明: <https://unanimated.github.io/ts/scripts-manuals.htm#hydra>

视频教程 (英文, 从这一节点开始看即可): <https://www.youtube.com/watch?v=d7ds9QG4Kfl&t=321s>

10. Rotate Gradient - 可任意方向进行渐变的 Aegisub 插件

Aegisub 有很多种渐变插件，但大部分都只能进行纵向或横向渐变，比如上一点用 HYDRA，就只能横向或纵向渐变。利用 Rotate Gradient 插件可以实现任意方向进行渐变。

- (i) 把需要处理的那行字幕设置好位置\pos 和颜色\c，然后复制多一行。
- (ii) 选择新复制的那行字幕，设置好目标颜色\c。然后用矢量裁剪工具沿顺时针方向绘制 3 个点。其中，前两个点决定了渐变的开始位置，最后一个点决定了渐变延续的距离和渐变走向。
- (iii) 选择需要处理的那行字幕和新复制的那行字幕（共两行），然后 Aegisub 菜单栏选“自动化->Rotate Gradient”，勾选需要渐变的标签\c，填写好渐变步长，点 OK 按钮执行。



官方用法说明 (里面有视频): <https://phoscity.github.io/Aegisub-Scripts/Rotated%20Gradient/>

另一种处理任意方向渐变方法是用 Gradient Cut 插件，这个插件比 Rotate Gradient 更简单，只要用矢量裁剪框起来，然后执行插件按照里面填参数即可，里面有 Angle 参数可以设置渐变的角度。但是，Gradient Cut 插件生成的渐变字幕是图形化后（\p）的字幕，文字已经不会在字幕中体现，这对字幕的后期修改比较麻烦（比如做繁化）。

11. DependencyControl - 用于管理 Aegisub 插件的插件

用来管理 Aegisub 插件的插件。每次打开都会去 GitHub 拖元数据，可能需要魔法上网才能使用。网络不好会把 Aegisub 卡住。用了后安装和升级一些 Aegisub 的插件是挺方便的。

下载链接：<https://github.com/TypesettingTools/DependencyControl/releases>

动手能力强的话，可以把 aegisub\config 里面的 IO.DependencyControl.json（运行一次后就会生成）改一下，最外层的 config 增加两个参数 updateInterval 和 updaterEnabled（检测更新的毫秒间隔和是否启用更新），例如，增加两行：

"updateInterval": 8640000,

"updaterEnabled": false,

```
.....},
....."config":
.....{
.....  "logMaxFiles": 30,
.....  "logLevel": 3,
.....  "maxToFileLevel": 2,
.....  "updateInterval": 8640000,
.....  "traceLevel": 3,
.....  "updaterEnabled": false,
.....  "updaterRunning":
.....  {
.....    "host": "arch.AegisubChain",
.....    "time": 1667757669
.....  }
.....}
```

12. 其他的 Aegisub 插件

站内 Seekladoo 大佬收集了一些在 DependencyControl 里没有的小插件，Add k Tags（自动打平均的 k 值）、Combine Overlaps（选择时间有重叠的字幕）、delete_fx（快速删除遇到的第一个特效字段为 fx 后的所有字幕【特效字段为 fx 后如果有非 fx 的字幕也会删掉，慎用】）等还是挺好用的：

https://github.com/seekladoo/aegisub_scripts

几个小插件【“函数库”那个挺好用的】：<https://github.com/RichardKiriko/Kiriko>

Unanimated 的插件合集介绍：<http://unanimated.hostfree.pw/ts/scripts-manuals.htm>

Unanimated 的插件合集下载（建议直接在 DependencyControl 里安装）：

<https://github.com/unanimated/luagisub>

PhosCity 的插件合集介绍（某些插件需要特定编译版本的 aegisub）：<https://phoscity.github.io/Aegisub-Scripts/>

PhosCityd 的插件合集下载（建议直接在 DependencyControl 里安装）：

<https://github.com/PhosCity/Aegisub-Scripts>

DependencyControl 中全部插件的列表：<https://typesettingtools.github.io/depctrl-browser/macros/>

GitHub 上还有很多 Aegisub 的插件，搜一下 Aegisub script 即可。

13. AdjustAssTime - 调整字幕时轴/FPS 和合并字幕工具

论坛里大佬 tonyhsie 的又一力作。之前没有注意到，后来遇到合并字幕需求时候用了一下，发现十分好用。可以平移调整字幕的时间轴、帧率、合并字幕，甚至可以利用合并功能来调分辨率，后文会说到。

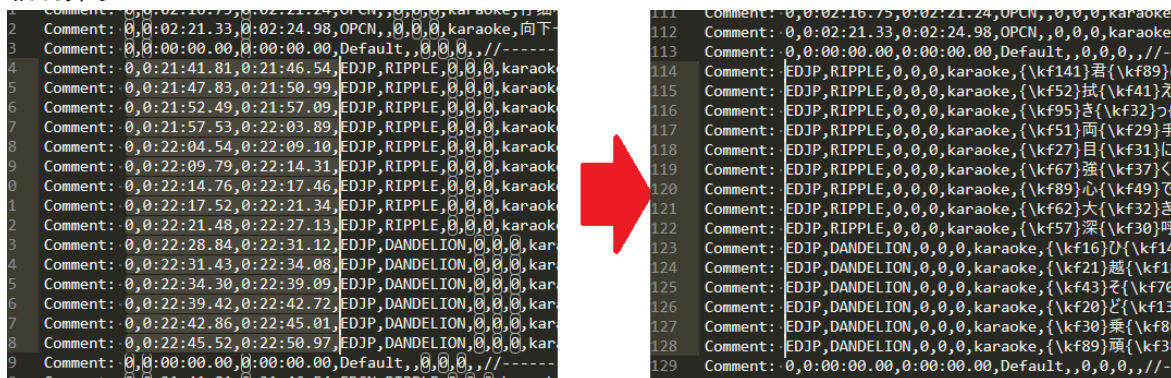
自己看帖内说明：<https://bbs.acgrip.com/thread-1936-1-1.html>

14. SublimeText3 - 文本编辑工具

为何推荐一个文本编辑工具？因为有个操作对直接编辑字幕很方便。当然，类似功能的编辑器还会有其

他，我是 Sublime 忠实粉，所以推荐这个。

SublimeText 支持块状选择，可以按着 **Ctrl+Alt** 再按上键/下键 或 按着鼠标滚轮进行拖动选择（对，鼠标滚轮是能按下的，可能有人会不知道），就能选择一片对齐的区域去修改。如下图，我把某片区域的时间轴删掉了：



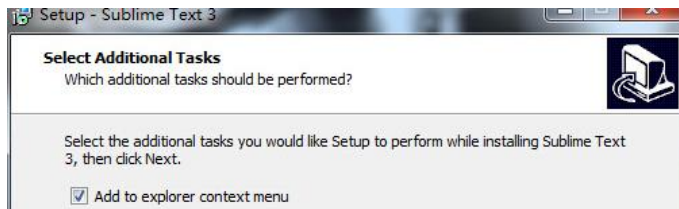
这块状选择的功能对直接处理字幕文本来说很方便，例如：改轴、更新整片 OPED 文字等。

Sublime Text v3.2.2b3211 x64 【含 ConvertToUTF8 插件和汉化】：

秒传：77354FF36E6CD464275F9B143A236D92#9E4F0FDEF165F6C1243392D60E43261E#11314261#ST.z

网盘：<https://www.mediafire.com/file/vn302ruvi8i0rtu>

安装时候记得勾选 Add to explorer context menu，这样在右键单击文件时就可以直接使用 Sublime Text 打开。



15. BeyondCompare - 文件比较工具

非常好用的一个文件比较工具。可以用于比对两个字幕，查看修改的差异。特别是在改错字范畴，十分方便。改错字相关的操作，下章会介绍。

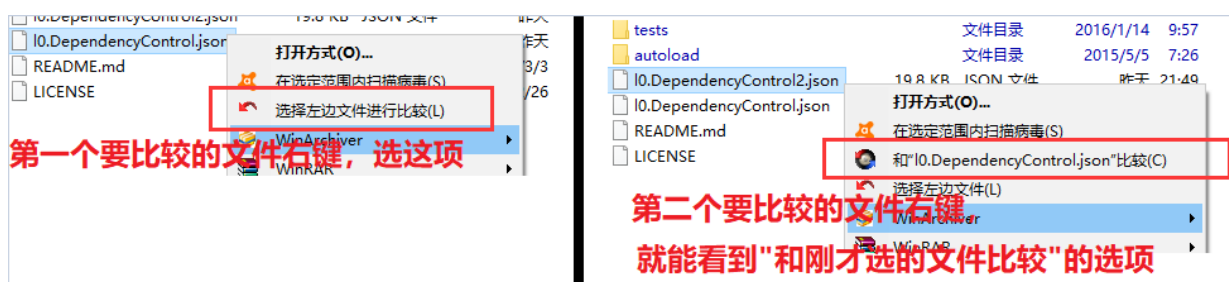
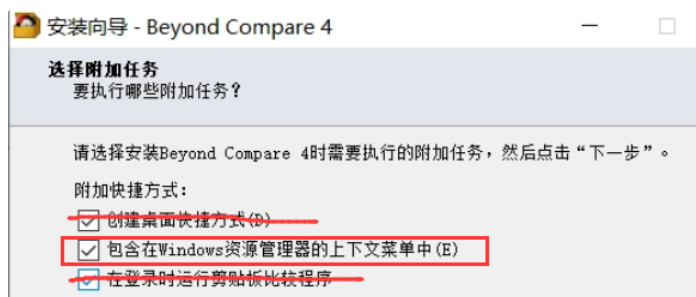
BeyondCompare v4.4.7.28397：

秒传：

354C7215AAAA11C35F929BC6F852C4B5#1DDF3725AB864B8489B7773F861E1462#23607287#BC_2023.z

网盘：<https://www.mediafire.com/file/npcyydyn9ff082>

安装时候记得勾选“包含在 Windows 资源管理器的上下文菜单中”。



16. SpleeterMsvcExe - 音频人声分离工具

Spleeter 是一个 Python 的机器学习的人声分离框架，可以将伴奏和人声分离成两个独立的音频。详情可以[点这里](#)查看。SpleeterMsvcExe 是 Spleeter 的一个 C 语言实现，不用安装 Python 的一大堆机器语言包，直接开箱即用，项目[点这里](#)查看。把人声分离后，可以利用干净的人声去做自动打轴之类的，准确度会高一点。下面提供的下载，包含一个脚本 spleeter_alass.bat，可以将 SpleeterMsvcExe 生成的人声音频，输入到 Alass 中，生成精确度比直接生成更好的带时间轴的空白字幕。内存小的注意，这工具有点吃内存，因为要实行机器学习计算，25 分钟的视频预留 4G 以上内存比较好。

SpleeterMsvcExe v1.0 打包【含 spleeter_alass.bat】：

秒传：

9F590F805DEF395373A0EE593E979AC7#A3BD1F9C8804BC4B5362EB01E6685EE5#182720967#SME.z

网盘：<https://www.mediafire.com/file/ggm8d3mv39azfu0>

17. DD 烤肉机 - 自动打轴工具

无意中搜索发现的，看介绍很牛逼的样子就试了试，但是还是不会用。

DD 烤肉机介绍：<https://www.bilibili.com/video/av457004292/>

DD 烤肉机下载：https://github.com/zhimingshenjun/DD_KaoRou2/releases

DD 烤肉机 v2.0.1mod【离线自改版，Spleeter、ffmpeg 等已升级，修正了一些 bug】：

秒传：

07C976771BEFC084416D152D9C88F3AD#C48C8333B78E4CA7451D79DBF2979C7B#226243255#DD.z

网盘：<https://www.mediafire.com/file/f4i56klrh5sm7t>

18. BatchSubtitlesConverter - 字幕批量处理工具

功能好像很多很强大，但是我还没用透。我只是用它简单地处理字幕格式转换。自己研究一下吧。

下载地址：<https://github.com/TSDARKNESS/BatchSubtitlesConverter/releases>

BatchSubtitlesConverter v2.9【Unlimited】：

秒传：CF65EE0C30E8717E5C2CAE52FC7859E9#CD167C2F0702807F0B71C236D9BDD9E7#1624103#BSC.z

网盘：<https://www.mediafire.com/file/luia5wea37nfdlp>

19. 繁化姬 - 字幕简繁转换

将繁体字幕转为简体 或 将简体字幕转为繁体，因为里面用了有语义分析、样式检测屏蔽、自定义词语替换等功能，简繁互转的效果比用 word 或其他一般的转换工具要好。另外，有一个常见的错字修正的神奇功能，下一章会提及用到。因为服务器在境外，可能要魔法上网使用起来才会比较稳。软件打开时或转换过程中的卡住话，大概率就是网络不通，拖不到元数据了。关掉重新打开尝试就行。

下载地址：<https://github.com/Fanhuaji/Fanhuaji-GUI-by-James1201/releases>

20. Textify - 复制窗体文字

这工具跟字幕有点不相关，但是我还是想推荐一下。因为我经常用来复制 FontLoaderSub 显示的字体路径。



Textify v1.10.4:

秒传: 6808BCD38F9B9783DF132BB60559CAE0#52F6A81FF900360C68A68C7AD4907E47#433116#T.z

网盘: <https://www.mediafire.com/file/jq39u8n4cl8fawf>

21. TranspWnds - 窗口置顶、设置透明度

这工具也跟字幕有点不相关，但是有时候跟其他软件组合起来使用还是不错的。有一个用法就是设置窗口透明度用来在 ASSDraw3 中临摹画图。

下载地址: <https://code.google.com/archive/p/transpwnds/downloads>

TranspWnds v1.4.1:

秒传: 4CA5AF87216032B95CF21BF7E7ACEEA9#4CA5AF87216032B95CF21BF7E7ACEEA9#156299#TW.z

网盘: <https://www.mediafire.com/file/fjd9yopvcwdwd9x>

四、字幕实战篇

1. ass 字幕基础

ASS 基本特效代码图文教程 by BBA 【总结得不错，看完基本就懂了】:

秒传: 423D6F30062B9DA631C7C7E31E6EB968#581AF8F769235BBDC9EAD8FDBCEA31A5#672155#教程.z

网盘: <https://www.mediafire.com/file/ih1hfg982npdits>

中文文档 (标签相关): https://aegi.vmoee.info/docs/3.2/ASS_Tags/

中文文档 (卡拉 OK 模板自动化相关): <https://aegi.vmoee.info/docs/3.2/Automation/>

2. 改错字

a) 繁化姬+比对软件

繁化姬自带常见错字修正功能，利用繁化姬和比对软件（例如 BeyondCompare），可以简单快速揪出字幕错字。

譬如，你手上有一个简体字幕 A，先用繁化姬把简体字体繁化成字幕 B，再用繁化姬把字幕 B 简化成字幕 C，接着用比对软件进行字幕 A 和字幕 C 的对比，人工检查不一样的地方进行修正。

为何不直接把简体的字幕 A 直接一次性简化到字幕 C，要多一步繁化呢？这里有点门道，我测试了多次发现的，简→繁→简 比直接 简→简 错字修正的概率高。因为第一次转换（繁化）过程中会产生错字，第二次转换（简化）会更容易命中错字修复规则修复更多的错字。

当然，如果本来字幕是繁体的，那么可以进行繁→简→繁这种操作，然后再比对前后两个繁体字幕。用 BeyondCompare 作比对的话，按上面菜单栏的“差别”按钮，就会只显示比对差异的部分。差异部分一般都是很少的，扫一眼就检查完了，然后可以直接在上面修改错字，然后框的右上角有保存按钮。

繁化姬转换出来的不一定就是对的，大部分情况都会存在几个错字，必须人工检查一次。

3. 日语语音转文字

我不太懂日文，但我看了几百部动画，对日文某些句子还是能大概理解其含义的。在改进字幕的过程中，对某些字幕的翻译我会产生一些疑惑，感觉翻译得不太对。这时候，我会使用语音转文字，把动画的对话转成日文，在把日文扔到多个翻译进行翻译比对。对于一些语速很快的句子，字幕组可能会听不清导致错译，这时候也可以借助语音转文字技术，分析出对话的日文，在进行翻译。

这里推荐一个网站 ConvertSpeech，单独几句的话，日文语音识别准确率挺高的，应该有 8 成以上。但是免费的有限制，1 个 IP1 天只能识别一次。魔法上网可以少量突破 IP 限制。其他的对日语的识别都不太行。这里又能吹一下之前章节提到的音频人声分离工具 SpleeterMsvcExe。对于非人声背景音较大的对话，可以先用 SpleeterMsvcExe 把人声分离出来，再扔到 ConvertSpeech 里做识别，识别正确率会有较大提升。

ConvertSpeech: <https://convertspeech.com/ja-JP/>
GoogleCloud: <https://cloud.google.com/speech-to-text/#put-speech-to-text-into-action>
IBM (需注册): <https://www.ibm.com/cloud/watson-speech-to-text>

上面推荐的是可以直接传音频文件去识别的，还有其他的网站，但不能传音频文件，只能麦克风实时录音，这种我就作不推荐了。手机 App 也有很多翻译软件可以实时录音翻译的，我也不作推荐。

4. 用 ASSDraw3 画图

ASSDraw3 是 Aegisub 自带的画矢量图工具，如果需要为字幕做个简单的图，可以用 AssDraw3 作画，而不必动用 Adobe 等那些大型的作画工具。ASSDraw3 只有简单的基本功能，生成整数的矢量点，精度不太高，但胜在便捷。可以直接拖一张图片进去 ASSDraw3 中当成背景进行矢量临摹绘制。

利用 ASSDraw3 对背景进行临摹的视频教程: <https://www.bilibili.com/video/av47054247/>

5. 动画插曲歌词翻译见解

我不太懂日文，以下仅是我改进字幕过程中，对日文歌词进行机翻比对和对动画剧情的理解后，得出的一点见解。不一定正确。

很多动画的歌词实际上是在讲述整季动画剧情里的一些故事，或对某些场面的描述。歌词翻译的话，可以想想为何会出现这句歌词，究竟歌曲想表达动画里的什么东西。不要直白地按照语义去翻译，这样翻译出来的歌词可能会成为一个四不像，东说一下，西扯一下，没有一个主题，不知道想表达啥，让人感觉很奇怪——为何歌词里会有这上下文无关的一句？。当然，想知道整季动画的剧情，要等播完看完才知道。没播完的话，对剧情里故事、场面的了解是不完整的，等做季度合集时候再改进就好。

下面举例说明一下，若对某字幕组冒犯了，我先给你说声对不起。我并没有贬低的意思。我的翻译也不是说就是正确的，只是想表达一下我的见解。

约会大作战 4 的片头曲《Over》:

某组的翻译	
目覚めても 目覚めても 届かない 結末に見慣れてしまいそうなほど また一つ また一つ 消えてゆく 増えすぎた犠牲 無機質な影 どこかに置き去りにした 正しさも 躊躇いも まるで何もなかったように 確実に仕留めるの 閉ざしかけた唇 触れて 繋いで 唯一の望み以外 何も惜しくはないの 優し過ぎるあなたの為なら 何度でも捧げましょう 諦めてくれるなら	苏醒 觉醒 传达不到 结局早已厌闻厌听 一个 又一个 化为泡影 过量的牺牲 无机质样的影子 舍弃在无觅之地的正确与踌躇 如同不曾存在 切实处决 紧闭的双唇 触碰 结合 除去唯一所求 毫无保留 若是为过于温柔的你 无数次奉献 只要你愿放弃
我的翻译	
目覚めても 目覚めても 届かない	一次又一次地醒来 依旧到达不了

結末に見慣れてしまいそうなほど また一つ また一つ 消えてゆく 増えすぎた犠牲 無機質な影 どこかに置き去りにした 正しさも 躊躇いも まるで何もなかったように 確実に仕留めるの 閉ざしかけた唇 触れて 繋いで 唯一の望み以外 何も惜しくはないの 優し過ぎるあなたの為なら 何度でも捧げましょう 諦めてくれるなら	这种结局对我来说已司空见惯 一人又一人 不断消失着 徒增的牺牲 成为了没有生命温暖的影子 义不容辞、毫无犹豫地自我遗弃在某处 我若无其事般地 使出了制胜一招 紧闭的双唇 触碰 合吻 除了绝无仅有的心愿 其余一切我皆毫不在乎 为了过甚温柔的你 无论多少次我都愿意献出一切 如果你能打消念头的話
--	---

五、字幕特效篇

1. 一些推荐

字幕特效这东西，别想太多复杂，实际上就是普通的那几个 ass 标签进行了不同位置、不同时间的组合。有代码经验的，可以写脚本去生成 ass 标签；不懂编码的，可以装特效插件，在特效插件里对应位置填空生成 ass 标签。论坛里的 Seekladoo 大佬分享了很多特效字幕制作相关的帖子，可以[点这里](#)查看。我这里就抽一些出来推荐一下：

模板收集相关：

Aegisub 447 例 ASS 特效模板：<https://bbs.acgrip.com/thread-7339-1-1.html>

TCAX 特效字幕模板 001-467：<https://bbs.acgrip.com/thread-8816-1-1.html>

TCAX 模板特效在线预览等：<https://space.bilibili.com/5423735/video?tid=36&keyword=&order=pubdate>

特效收集（油管，需魔法上网）：<https://www.youtube.com/channel/UCPx6IJDJ3FI7FXiH5vx82aA/videos>

Kirena Jang 的特效字幕（油管，需魔法上网）：<https://www.youtube.com/@KirenaJang/videos>

特效分类聚合（之前油管抓的，现在改成私密视频不能直接看了，缺 1 个字幕，网盘数据丢失下不了）：

秒传：2E89B2FCAA4A6D7E71DA6300E3AC8BAD#8B74BFE25D739B4DD773B04013FBEC0A#79832923#特效.z

网盘：<http://www.mediafire.com/file/49b0apmi42i7uy1>

教程相关：

Aegisub 中文 Wiki（前文发过一次，这里再发一次）：

https://aegi.vmo.einfo/docs/3.2/Automation/Karaoke_Templater/Code_execution_environment/

Aegisub 特效字幕自动化入门：<https://www.bilibili.com/video/BV1Z4411z7BH>

Aegisub 特效字幕高级教程（学完应该能成为特效大佬了）：

<https://space.bilibili.com/346816900/video?tid=36&keyword=&order=pubdate>

ASS 特效插件 Kara Effector 3.5：<https://bbs.acgrip.com/thread-6559-1-1.html>

TCAX 软件本体+视频教程+...：<https://bbs.acgrip.com/thread-6383-1-1.html>

2. 遇到的一些坑

a) Mocha 导出的跟踪数据处理后，文字抖动严重

因为 Ass 字幕放缩标签 \fscx、\fscy 的精度问题，某些小数点的放缩会用相近的放缩来显示，例如 \fscx60.15 可能会用 \fscx60.1 或 \fscx60 来显示。在运动过程中，这微小的放缩偏差，可能就会造成文字抖动现象。一个解决窍门是，使用小字号的字体，然后对字体进行放大处理。

例如，\fs60 实际上跟 \fs6\fscx1000\fscy1000（\fs6 放大 10 倍就是 \fs60）是相等的。我们选择用 \fs6\fscx1000\fscy1000。这样，Mocha 的跟踪数据通过 Aegisub-Motion 的放缩计算后，很大可能就不会产生小数点，或者只会产生小数点后 1 位。这样，放缩的精度问题就能解决了。

b) 自动化 loop 相关

如果自动化进行了 loop 操作，则每次循环都会重新算一次代码。要是代码中使用了随机函数值当参数，那么每次进行 loop 循环时候都会算一次随机数。如果需要只在第一次进入 loop 循环时候计算，可以用 `j==1` 判断然后用 `remember` 保存。例如：

3 行字幕可能会产生 1~3 种颜色：

```
!maxloop(3){\an5\pos($lcenter,$lmiddle-j*250!)\3c!colors[math.random(3)]!}
```

3 行字幕只会产生 1 种颜色：

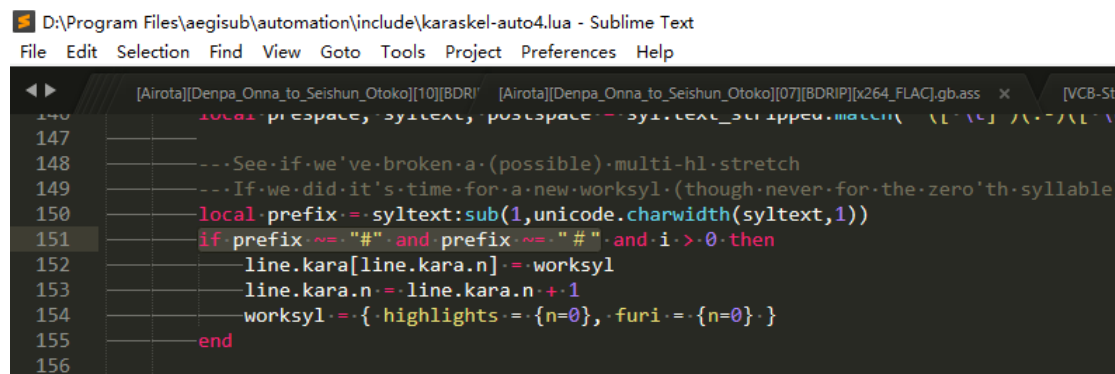
```
!maxloop(3){\an5\pos($lcenter,$lmiddle-j*250!)\3c!colors[j==1 and remember("cc", math.random(3)) or recall.cc]!}
```

c) #开头的字幕

之前处理一个标题，用 # 打头然后 EP、标题那样的。我写了行自动化来处理标题定位的问题。百思不得其解，执行模板后，发现生成的行并没有文字，就一个空白行。

开始以为 Aegisub 出什么 bug 了，后来看卡拉 OK 模板的源码实现才发现，原来，自动化会忽略掉 # 开头的文本的，就算前面有 { } 带着标签或空格字符，也会忽略掉。这种情况，在前面加 \h 或 \N 等字符，就能解决。

相关判断代码在 `karaskel-auto4.lua` 里，搜“`prefix ~= "#"`”，你会发现有 2 个位置有判断。



d) template char 是获取不到 \$swidth 和 \$sheight 的

template char 是获取不到 `$swidth` 和 `$sheight` 的，需要自己用 `$sright-$sleft` 和 `$sbottom-$stop` 来算。

e) 对精度要求不严格的情况下，尽量使用整数或只保留 1 位小数

对精度要求不严格的情况下，尽量使用整数或只保留 1 位小数，如果特效代码很多的话，可能减少字幕文件 10~30% 的大小。自动化代码生成后，用文本工具打开字幕，随意浏览一遍生成的代码就能大概知道哪个位置的数值可以进行精度优化。某些除不尽数的小数位数会多达 14 位，完全没必要留这么多。

f) table 变量是不能跨作用域被 remember 的

如果你在某作用域定义了一个 local 的 table 变量，然后 remember 到“A”了。一旦出了 table 的作用域，你再 recall.A 就会取不到 table 的值。在 loop 里面调用函数，函数里面如果有这种用法的话，下一次 loop 循环就会取不到上一次 loop 循环 remember 的 table 值，要小心。后文提供的 bezier_move 函数，正是因为这个问题，导致 bezier_points_x、bezier_points_y、bezier_factk 这 3 个 table 变量我使用了全局变量而不是将 local 变量 remember。

g) 建议使用 `math.randomseed` 确保每次处理的随机数都相同的值

使用 `math.randomseed` 可以确保每次运行自动化模板，再同一个位置的随机值都能取得固定的值。这样生成的字幕差异也方便，每集的随机特效看起来也比较统一。

h) 获取字幕标注的分辨率的方法

长、宽分别是 `meta.res_x` 和 `meta.res_y`。我觉得用字幕标注的分辨率比用视频的分辨率来计算更妥当。

i) 不建议使用思源系列的字体

不同字幕滤镜可能会对思源系列的字体产生不一样的效果，而且 `v1.x` 和 `v2.x` 字体加粗的调用方式也有些不同。论坛里也出现过很多使用思源字体出现各种的情况，其中比较详细介绍的一帖：

<https://bbs.acgrip.com/thread-3805-1-1.html>

思源主要也就宋体和黑体，代替这两种字体的字体多的是。可以用方正雅宋系列和方正兰亭黑系列、汉仪旗黑系列替代。这些系列都是多字重的 GBK 字集。

宋体	横比较细的宋体：		横竖都均等的宋体：
	方正特雅宋	方正博雅宋	方正屏显雅宋
	方正大雅宋	方正小标宋	方正宋黑
	方正粗雅宋	方正大标宋	方正悠宋
	方正中粗雅宋	方正粗宋	方正粗黑宋
	方正中雅宋	方正宋三	方正筑紫明朝宋
	方正准雅宋	方正宋体	
	方正标雅宋		
	方正细雅宋		
	方正纤雅宋		
黑体	方正兰亭特黑		汉仪旗黑-25S
	方正兰亭大黑		汉仪旗黑-30S
	方正兰亭粗黑		汉仪旗黑-35S
	方正兰亭中黑		汉仪旗黑-40S
	方正兰亭粗黑		汉仪旗黑-45S
	方正兰亭准黑		汉仪旗黑-50S
	方正兰亭黑		汉仪旗黑-55S
	方正兰亭细黑		汉仪旗黑-60S
	方正兰亭纤黑		汉仪旗黑-65S
	方正兰亭超细黑		汉仪旗黑-70S
			汉仪旗黑-75S
			汉仪旗黑-80S
			汉仪旗黑-85S
			汉仪旗黑-90S
			汉仪旗黑-95S
			汉仪旗黑-105 简繁

j) 对字幕进行分辨率缩放建议用 `AdjustAssTime`

有时候想将字幕的分辨率统一一下，会对字幕的分辨率进行调整。`Aegisub` 里自带分辨率调整功能，但是对 `\bord` 和 `\blur` 的缩放会有一些 bug，导致调整分辨率后的字幕跟原来的样子不一样。可以使用 `AdjustAssTime` 进行分辨率调整。

六、代码篇

写了几个函数，分享一下吧。大部分是参照各路大神提供的所修改来的，使用范围可以更广。觉得可用就直接拿去。扔进前面提到的 Aegisub 其他插件**函数库**中（这插件的保存位置在<aegisub 安装目录>\log\funcs.lua），就能生成单行格式化好的方法。如果你之前没用过这个插件，可以直接覆盖，否则，打开文件自己选择自行在菜单“自动化→Add function”录入吧。

1. bezier_move - 贝塞尔曲线移动

```
function bezier_move(s_time, e_time, afterimage_dur, x_blur, y_blur, accel, max_space, frame_dur, ...)
```

跟别人提供的 bezier_move 相比，这个可以配合模板声明时候的 loop（template ... loop N）来使用来实现多个图案进行贝塞尔曲线运动。仅在模板声明 loop 的循环中计算（准确来说应该是记录）输入参数。

参数的含义：

s_time，相对于行首的开始时间。

e_time，相对于行首的结束时间。

afterimage_dur，运动完成后的残影时间。

x_blur，x 抖动偏移（可以是函数）。

y_blur，y 抖动偏移（可以是函数）。

accel，移动的加速度（越小体感越快）。

max_space，单帧时间运动的最远距离。

frame_dur，一帧所需要的毫秒时间。

...，不定数量的控制点坐标，第奇数个是 x 坐标，第偶数个是 y 坐标，必须 x 跟 y 成对出现。

2. char_counter、word_counter - 字符计数器、单词计数器

要配下面这行 char_counter_init 来使用，这个要用 code line 修饰词来运行，是用来初始化计数器的。

```
ci = {0,0,0,0,0,0,0,0}; cn = _G.unicode.len((orgline.text_stripped:gsub("[%s ]+", ""))); con =  
_G.unicode.len((orgline.text_stripped)); cwn = _G.unicode.len((orgline.text_stripped:gsub("[A-Za-z0-9-]+",  
"A"):gsub("[%s ]+", "")));
```

cn 是单行非空白字符的总字数，con 是所有的总字数，cwn 是单词的总个数（总个数=英文单词数+其它符号的单字数）。当前只初始化了 8 个计数槽位，需要更多槽位的话，自己修改 ci 的数组长度。ci[N]是用来计数器的数值的。

```
function char_counter(ref, text, ignore_regex)
```

字符计数器。

参数的含义：

ref 是保存计数器使用的槽位编号，从 1 开始。

text 是用来判断的文本，非必要参数，可以不输入。一般用 syl.text_stripped 当参数输入。

ignore_regex 是忽略文本的正则表达式，命中这个正则表达式的话，计数器就不会计数。text 非 null 的话，默认会忽略空白和空格字符。

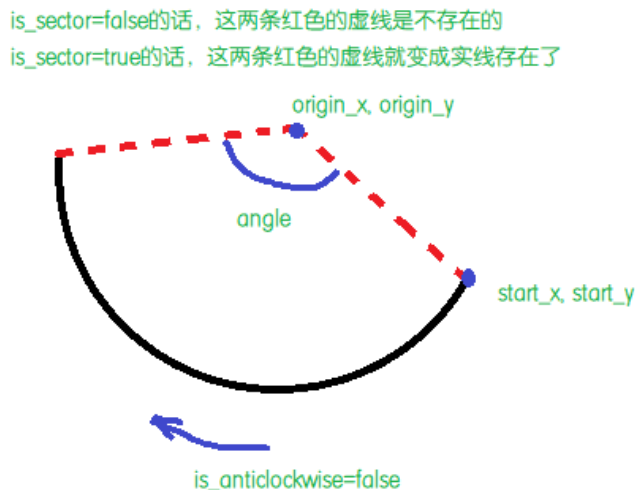
```
function word_counter(ref, text, ignore_regex)
```

单词计数器。参数跟 char_counter 一样。不同的一点是，带 text 参数时，这个对连续的英文单词（正则表示就是"[a-zA-Z.]+"）只会计数 1 次。这个方法在各个文字设置不同颜色的场景，对整个英文单词设置同一种颜色比较方便。

3. draw_bezier_circular_arc - 用贝塞尔曲线来画圆弧

用 4 段贝塞尔曲线可以模拟一个圆，原理[点这里](#)查看。只要不是特别小的圆，画出来的圆效果都很好。

几个参数的含义是：圆心 x 坐标，圆心 y 坐标，圆弧的起始位置 x 坐标，圆弧的起始位置 y 坐标，圆弧跨越的角度（非弧度），是否逆时针，是否扇形
用图来说明可能更清晰一些。



function draw_bezier_circular_arc(origin_x, origin_y, start_x, start_y, angle, is_anticlockwise, is_sector)
参数的含义：

origin_x, 圆心的 x 坐标。

origin_y, 圆心的 y 坐标。

start_x, 弧线起始点的 x 坐标。

start_y, 弧线起始点的 y 坐标。

angle, 弧线旋转的角度。

is_anticlockwise, 是否逆时针旋转, bool 类型。逆时针则为 true, 顺时针则为 false。

is_sector, 是否扇形, bool 类型。扇形的话, 则会把半径也画上, 否则返回的只有弧线。扇形则为 true, 弧线则为 false。

4. clip_move、clip_line_move - clip 区域移动

ass 标签中, clip 区域是不能移动的, 最多是进行放缩。这方法使用单帧处理的方式, 令 clip 区域可以“移动”。clip_line_move 是对 clip_move 一个子化的实现, 只能线性移动。clip_move 则可以根据 clip_data_func 的输入, 实现速度、轨迹、大小动态变化的效果。接口实现得比较底层, 可以比较自由地控制。

function clip_line_move(clip_data, x_distance, y_distance, x_calibration, y_calibration, s_time, e_time, scale, is_iclip, accel, frame_dur)

参数的含义：

clip_data, clip 标签的数据。

x_distance, x 坐标移动的距离 (可正负)。

y_distance, y 坐标移动的距离 (可正负)。

x_calibration, x 的起始坐标修正 (进行放缩时候有用)。

y_calibration, y 的起始坐标修正 (进行放缩时候有用)。

s_time, clip 开始移动时距离字幕行首的时间。

e_time, clip 结束移动时距离字幕行首的时间。

scale, 放大倍数, 可配合 \p2、\p3 等放缩绘画标签使用。

is_iclip, 是否为 iclip, bool 类型。Iclip 的话则为 true, clip 的话则为 false。

accel, 移动的加速度 (越小体感越快)。

frame_dur, 一帧所需要的毫秒时间。

```
function clip_move(clip_data_func, s_time, e_time, scale, is_iclip, frame_dur)
```

参数的含义：

clip_data_func, 生成 clip 数据的函数。是一个形如 clip_data_func(cur_time, clip_s_time, clip_e_time)的函数，其中 cur_time 是当前处理帧距离字幕行首的时间，clip_s_time 是 clip_move 操作的开始时间，即 clip_move 的输入参数 s_time，clip_e_time 是 clip_move 操作的结束时间，即 clip_move 的输入参数 e_time。

s_time, clip 开始移动时距离字幕行首的时间。

e_time, clip 结束移动时距离字幕行首的时间。

scale, 放大倍数，可配合 \p2、\p3 等放缩绘画标签使用。

is_iclip, 是否为 iclip, bool 类型。Iclip 的话则为 true, clip 的话则为 false。

frame_dur, 一帧所需要的毫秒时间。

5. reverse_draw_direction - 反向绘制图形

这个可以将描点的绘图反向进行绘制，通过一个正向的图形，再在其末尾追加一个偏移或稍微放缩后的反向绘制图形，就可以实现一个仅剩描边的图形。

含有特殊绘制代码的图形可能会有 bug。一般用 ASSDraw 绘制出来的图形是没问题的。

```
function reverse_draw_direction(draw_data)
```

参数的含义：

draw_data 是绘制的图案数据

6. loop_handle - 可以配合 template 修饰语 loop 进行双层循环的脚手架

这个函数可以配合 template 修饰语 loop 达到双层循环的效果。比如某行字幕想重复 3 遍，可以直接在 template 的修饰语中填 loop 3，但是，那行字幕又是一个内部循环的多行字幕的话，那就要么分开 3 行写，要么改内部循环的代码。这个 loop_handle 的函数就是为了处理这种情况而编写的。因为要用到内联函数作为参数传入，所以看起来可能会有点困难。

```
loop_handle(main_func, loop_condition_func, init_func)
```

参数的含义：

main_func, 主处理函数。需要填一个形如 main_func(idx, condition)的函数，其中 idx 是循环次数，condition 是由 loop_condition_func 返回的条件结果。返回值为字幕代码的字符串

loop_condition_func, 条件控制处理函数。需要填一个形如 loop_condition_func(idx)的函数，其中 idx 是循环次数。返回值为两个：condition, loop_next_round_flag。其中，condition 返回值用于传给 main_func 作输入参数；loop_next_round_flag 是一个 bool 变量，用于判断下轮循环是否还需要继续，继续就返回 true，终止就返回 false。

init_func, 初始化处理函数。可不填，或填入一个形如 init_func(lh_current_loop, lh_loop_times)的函数，其中 lh_current_loop 是外部 loop（即修饰语中的 loop）的当前循环次数，lh_loop_times 是外部 loop（即修饰语中的 loop）的总循环次数。用于初始化一些变量（记录传入参数等），单行字幕只会执行一遍。如果 template 的修饰语中填 loop 3，那么该行字幕就会处理 3 遍，则 init_func 也会根据每遍处理而执行 3 次。比如，如果传入参数中使用到了随机数，如果没有记录随机数的话，根据 aegisub karaoke 模板执行的流程，那么每次循环生成的字幕行就会传入不同的参数（因为每次循环都会执行一次随机数生成）。如果想固定某个传入随机数在循环中不变，就要用在 init_func 中用 remember 去记录，再在 main_func 或 loop_condition_func 中用 recall 去取出使用。

7. clip_gradient、color_line_gradient - 渐变颜色生成

如果用 VSFilterMod 字幕滤镜的话，用 vc 标签就能简单生成渐变颜色。但是其它字幕滤镜并不支持这个 tag，所以要做颜色渐变要另寻他法。简单的线性渐变可以用 color_line_gradient 函数，复杂的情况可以使用更加通用的 clip_gradient 函数，但会困难一些。

```
function color_line_gradient(stagger_gap, start_left, start_top, end_right, end_bottom, is_vertical,
start_color_tags, end_color_tags, RGB_or_HSV_or_HSL, accel, angle)
```

参数的含义：

stagger_gap, 切割间距, 整数。渐变是切成一行行或一列列去处理的。切割间距越小, 渐变得越细腻, 但是生成的字幕行数也越多, 渲染负荷也越重。

start_left, 左边沿的位置。

start_top, 上边沿的位置。

end_right, 右边沿的位置。

end_bottom, 下边沿的位置。

is_vertical, 渐变方向是横向还是纵向, bool 型, 纵向渐变则填 true, 横向渐变则填 false。

start_color_tags, 渐变开始的颜色标签字符串, 可以多个标签组合。

如: "\c&H001C2B&\1a&H40&\3c&H4D5E6F&"。

end_color_tags, 渐变结束的颜色标签字符串, 可以多个标签组合。

RGB_or_HSV_or_HSL, 颜色渐变的方式, 可以填"RGB"或"HSV"或"HSL"这 3 个值。渐变颜色从开始的颜色根据这个方式渐变到结束的颜色。

accel, 渐变的加速度, 可不填, 默认是 1。可以调整渐变的颜色比例。

angle, 调整渐变方向的角度度数。

```
function clip_gradient(tags_func, stagger_gap, start_left, start_top, end_right, end_bottom, is_vertical,
clip_func)
```

参数的含义：

tags_func, 生成标签字符串的函数。一个形如 color_tags(cur_idx, clip_tag)的函数, 其中 cur_idx 是当前生成渐变行数的计数器, clip_tag 是 clip_func 返回的当前渐变行的 clip 标签字符串。返回值是该切割间距的渐变行的标签字符串。

stagger_gap, 切割间距, 整数。

start_left, 左边沿的位置。

start_top, 上边沿的位置。

end_right, 右边沿的位置。

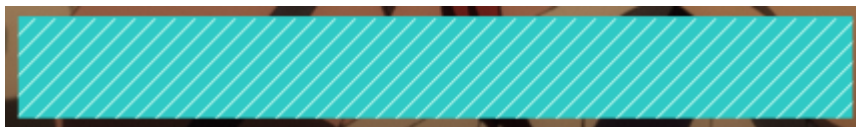
end_bottom, 下边沿的位置。

is_vertical, 渐变方向是横向还是纵向, bool 类型, 纵向渐变则填 true, 横向渐变则填 false。

clip_func, 生成当前的渐变行的 clip 标签字符串, 实际就是间距切割。是形如 clip clip_func(cur_idx, is_vertical, clip_start_left, clip_start_top, clip_end_right, clip_end_bottom)的内联函数, 其中 cur_idx 是当前生成渐变行数的计数器, is_vertical 是渐变方向, clip_start_left、clip_start_top、clip_end_right、clip_end_bottom 是用户输入的左上右下四个边沿。返回值是当前渐变行 (间距切割后) 的 clip 标签字符串。

8. interlace_strip_clip - 生成用于间隔条纹的 clip 数据

这是用来生成间隔条纹的, 因为没有对边沿修整处理, 所以只能用于 clip 中。也就是说生成的绘画数据会超过边界, 放在 clip 中, 超过部分不会显示所以一般没问题。直接做绘图数据的话, 超过边界的部分会展示出来, 所以就不太行了。



```
function interlace_strip_clip(start_left, start_top, end_right, end_bottom, gap1, gap2, angle, shift_x, shift_y)
```

参数的含义：

start_left, 左边沿的位置。

start_top, 上边沿的位置。

end_right, 右边沿的位置。

end_bottom, 下边沿的位置。
gap1, 第一种条纹的横向间距。
gap2, 第二种条纹的横向间距。
angle, 条纹角度度数。
shift_x, 生成数据后在 x 轴方向上移动的距离。
shift_y, 生成数据后在 y 轴方向上移动的距离。

七、后记

字幕就像一盒月饼。
文本就是月饼的馅料,
糕点师的工艺决定了馅料的咸甜;
时轴就是月饼的饼皮,
烘焙师的专注决定了饼皮的火候;
样式就是月饼的包装盒,
屏幕字特效就是月饼包装盒上的立体装饰;
歌词就是月饼的手提袋,
歌词特效就是月饼的手提袋上的烫金图腾。
可能这月饼平平无奇, 没有华丽的包装, 但味道却绝佳;
也有可能这月饼包装炫彩夺目, 但月饼却味异难啃, 不尽如人意。

v1.0 写了 2 个晚上, 洋洋洒洒码了快 1 万字, v2.0 的新内容又增加了快 5000 字。通篇又臭又长, 看似很多, 实际回想一下, 却没什么是我自己的东西, 基本都是转别人的链接来分享的。希望能带给你机缘, 让你在字幕这一块有所领悟吧。希望你的领悟, 能让字幕界有所进步, 哪怕只有零星的一丝。

v1.0 2022 年 12 月 01 日
v2.0 2023 年 12 月 01 日
发布于 bbs.acgrip.com