

vsfiltermod手册

简要介绍

VSFilterMod is modification of original **VSFilter** subtitle renderer by Gabest. This mod brings up new features and some minor bugfixes.

最初版修改者
VSFliterMod是Gabest对原版**VSFliter**字幕渲染器的修改版本。该mod带来了新的功能和一些小bug修复。特效标签

注：vsfiltermod已经经过很多代的修改，本版vsfiltermod手册介绍的是基于[domo](#)和[电脑迷](#)修改的版本[\[1\]](#)，与之前的版本相比最大的不同是——添加了更多特效标签。

目录

简要介绍

目录

特效标签说明

前要

字体缩放 Font scale

边缘模糊 Blur edges (高斯函数) (Gaussian kernel) (补充) (extended)

纵向偏移 Leading

横向偏移 Horizontal leading

渐变 Gradients color

渐变 Gradients opacity levels

抖动 Shaking

图片替代颜色填充 Images instead of color fills

边界变形 Boundaries deforming

扭曲 Distortion

基线倾斜 Baseline obliqu

Z 坐标 Z coordinate

极限移动 Polar move

曲线移动 Spline-move

矢量clip移动 Moveable vector clip

正交投影 Orthogonal projection

安装与使用 Installation And Usage

对于Aegisub

对于Avisynth

直接使用 DirectShow

使用VirtualDub

参考资料

尾注

特效标签说明

前 要

1. 如无特殊声明，所有标签都对绘图有效，尽管为了方便叙述在说明中使用的文字是**文本**而不是**文本和绘图**。
 2. 绘图(\p)不等同与挂图(\\$img)，挂图的概念与绘图完全不同，请前往挂图
Images instead of color fills
(图片替代颜色填充) 条目明晰清楚概念。
 3. 特殊声明
 1. 若无特殊声明，标签不可与\t配合使用。
 2. 在本文档的说明中，分有旧mod和新mod这两种情况，旧mod指之前所有版本的vsfiltermod，新mod特指本文档所述的domo和电脑迷所修改的vsfiltermod，其区别在于新mod添加了额外的特效标签。
 - 若无特殊声明，标签为旧mod标签。
 3. 在代码块示例中，有一些特殊字符表示一些特定含义，其本义是aegisub自动化会使用到的标签，在这里引用进行特指。
 - `$x`：表示原字幕行的x轴坐标 - [纵向偏移][横向偏移]条目
 - `$y`：表示原字幕行的y轴坐标 - [纵向偏移][横向偏移]条目
 - `line.width`：表示字幕的原宽度 - [扭曲]条目
 - `line.height`：表示字幕的原高度 - [扭曲]条目
 4. 本条目未声明且无批注的概念，为先前所应熟知的基础概念，本文档不会再重复阐述(有时间的话可能会抽时间将一些基础又不是很常见的概念在文末标注一下)。
 5.
 - 本文档并非完全客观的说明文档，其中包含了本人片面的主观臆断，不一定准确，也不一定完全，若想阅读客观的说明文档，请移步[参考资料2](#)和[3](#)。
 - 如果有什么建议或者对本文档有什么错漏之处能够补正，希望能够与[我](#)练习，本人会及时进行修改的。
- ### 4. 特殊标注
1. 所有的特效标签会使用阴影标识起来。
 2. 在下文的说明中，使用尖括号括注起来的，属于参数内容，需要填入具体参数。
 3. 在某些标签中，在参数标注内，会有中括号标注，中括号括起来的参数表示该参数是可选参数，如果不填会填入一个默认值作为参数。
 4. 在特殊标签中，会使用\$符号开头表示特殊含义，\$表示4个颜色作用类型，即主要颜色、次要颜色、边框颜色、阴影颜色。

5. 特效标签解释框架

在下文的特效标签说明中，会分3个部分进行说明，即特效标签框架及说明、示例和补充。

- part 1：由该特效标签的标准书写和一串应用说明组成；标准书写规定该标签的格式，应用说明是英文原版说明，通常只是简单的一段文字说明。
- part 2：由一个代码块和代码块中代码的示例图片组成，在代码块中，以`--`作为开头的话是注释。
- part 3：更为具体的特效标签说明，会详讲参数和一些注意事项。

6. **注意**：如果要使用`vsfiltermod`进行压制，最好请将压制软件中的`vsfiltermod`替换成本文档介绍的`vsfiltermod`，**特别是**使用了新标签的情况下。

在一些旧版`vsfiltermod`中，`\img` 标签是无法使用的，还有一些其他的零零散散的bug，本文档使用的`vsfiltermod`是相对bug更少的。

7. pdf无法填充动图，故一些动态效果并无法直接嵌入图片演示，还请亲自使用体会其效果。

字体缩放

`\fsc<scale>`

Similar to aggregate `\fscx<scale>\fscy<scale>`. Animatable by `\t`.

效果与同时使用 `\fscx<scale>\fscy<scale>` 相同。
可配合 `\t` 实现动态效果。

Example

示例

```
1 {\fsc200}TEST
2 --将文本 TEST 设置成200%大小，相当于{\fscx200\fscy200}
```

```
1 {\t(0,1000,\fsc200)\t(1000,2000,\fsc100)}TEST
2 --在0~1000ms时，将文本 TEST 放大到200%大小；在1000ms~2000ms时，将文本 TEST 缩小到原来的大小(100%)
```

Add

补充

- 参数
 - `<scale>` 的基础值是100，表示100%大小；
 - `<scale>` 值过大可能会因为渲染压力过大导致软件崩溃。
- 由于 `\fsc<scale>` 与 `\fscx<scale>\fscy<scale>` 等价，如果在一个字幕中，除了 `\fsc` 之外没有使用到其他的 `vsfiltermod` 标签，可以将 `\fsc ϕ` 整体替换成 `\fscx ϕ \fscy ϕ` (ϕ 表示同一个数值)，在压制中使用 `vsfilter` 而不是 `vsfiltermod`，从而减少压制时的渲染压力，提高压制速度。

边缘模糊(高斯函数)(补充)

`\xblur<strength>`

`\yblur<strength>`

Set the strength to blur the edges of the text, with X and Y position set separately.
Animatable by `\t`.

分别设置边缘模糊在X方向和Y方向的强度。
可配合 `\t` 实现动态效果。

该标签为新mod标签。

Example

示例

```
1 {\xblur10}xblur
2 {\yblur10}yblur
3 {\xblur10\yblur10}xyblur
4 {\blur10}blur
5 --上代码效果即为下图表现效果(左1为有边框效果；中为无边框无阴影效果；右1为无边框有阴影效果)
```



Add

补充

1. 参数

- `<strength>` 为边框模糊强度，基础值为0；
- `<strength>` 可以不是整数，但不能是负数；
- `<strength>` 值过大可能会因为渲染压力过大导致软件崩溃。

2. 和 `\blur` 类似，当文本无边框的时候，该标签会模糊文字；否之，则只模糊边框。

Leading 纵向偏移

```
\fsvp<leading>
```

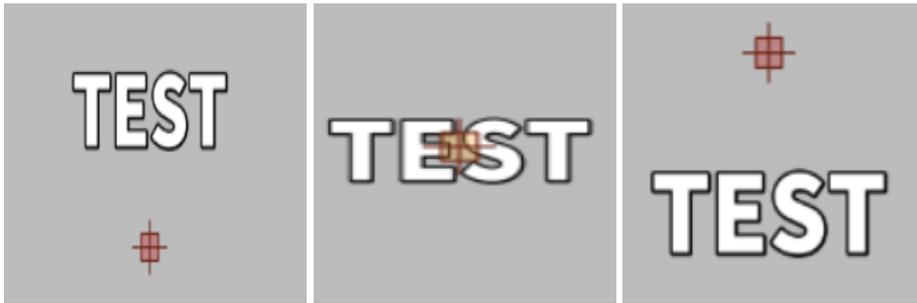
Changes text leading. Animatable by `\t`.

改变文字的纵向偏移。
可配合 `\t` 实现动态效果。

Example 示例

```
1 {\fsvp100}TEST
2 --将文本显示的位置向上平移100个像素。需要注意的是，文本的基点(pos点)仍然在原来的位置，见左1图（中图为无标签对比图）
```

```
1 {\fsvp-100}TEST
2 --将文本显示的位置向下平移100个像素。需要注意的是，文本的基点(pos点)仍然在原来的位置，见右1图（中图为无标签对比图）
```

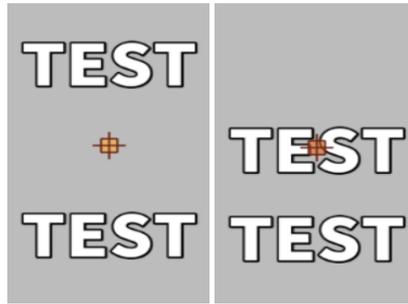


```
1 {\t(0,1000,\fsvp100)}TEST
2 --在0~2000ms时，将文本向上平移100个像素（在忽略基点的情况下，可以等价于{\move($x,$y,$x,!$y-100!,0,2000)}）
```

```
1 {\t(0,1000,\fsvp100)\t(1000,2000,\fsvp-100)}TEST
2 --在0~1000ms时，将文本向上平移100个像素；在1000ms~2000ms时，将文本从基点上面100像素位置平移到基点下方100像素的位置（相当于第二次实际上是移动了200像素）。（该效果无法通过\move实现）
```

```
1 {\t(0,1000,1.5,\fsvp200)}TEST
2 --在0~1000ms间，文本做加速直线运动
```

```
1 | {\fsvp100}TEST
2 | TEST
3 | --见左1（右1为无标签对比图）
```



Add 补充

1. 参数

1. `<leading>` 为偏移像素数，基础值为0；

2. `<leading>` 规定以y轴的正方向为正，负方向为负，即

当 $leading > 0$ 时，文本向上偏移；当 $leading < 0$ 时，文本向下偏移

2. 如上文在示例中所述，`\fsvp` 并不会对基点做出改变，也就是说，字幕的位置重叠规则仍然适用。
示例 6

3. 可以通过 `\fsvp` 实现一些 `\move` 无法实现的一些效果，比如变速直线运动，多段移动等等一些更复杂的**平移**运动。
示例 5 示例 4

横向偏移

`\fshp<leading>`

Changes horizontal text leading. Animatable by `\t`.

改变文字的横向偏移(效果同 `\fsvp` 相似)。

可配合 `\t` 制作变速直线运动。

该标签为新mod标签。

Example

示例

```
1 {\fshp100}TEST
2 --将文本显示的位置向左移动100个像素，需要注意的是，文本的基点(pos点)仍然在原来的位置，见左1图（中图为无标签对比图）
```

```
1 {\fshp-100}TEST
2 --将文本显示的位置向右移动100个像素，需要注意的是，文本的基点(pos点)仍然在原来的位置，见右1图（中图为无标签对比图）
```



```
1 {\t(0,2000,\fshp100)}TEST
2 --在0~2000ms时，将文本向左移动100个像素（在忽略基点的情况下，可以等价于{\move($x,$y,!$x-100!,!y,0,2000)}）
```

```
1 {\t(0,1000,\fshp100)\t(1000,2000,\fshp-100)}TEST
2 --在0~1000ms时，将文本向左平移100个像素；在1000ms~2000ms时，将文本从基点左方100像素位置平移到基点右方100像素的位置(相当于第二次实际上是移动了200像素)。（该效果无法通过\move实现）
```

Add

补充 同\fsvp

1. 参数

1. `<leading>` 为偏移像素数，基础值为0；

2. `<leading>` 规定以x轴的正方向为负，负方向为正，即

当 `leading > 0` 时，文本向左偏移；当 `leading < 0` 时，文本向右偏移

2. 如上文在示例中所述，`\fshp` 并不会对基点做出改变，也就是说，字幕的位置重叠规则仍然适用。

3. 可以通过 `\fsvp` 实现一些 `\move` 无法实现的一些效果，比如变速直线运动，多段移动等等一些更复杂的平移运动。

渐变 color

```
\$vc(left-top-color,right-top-color,left-bottom-color,right-bottom-color)
```

Creates gradients by using anchor colors. May be slow. Animatable by \t.

用颜色创建渐变。可能导致渲染时比较慢。
可配合 \t 实现动态效果。

Example

示例

```
1 {\1vc(left_top_color,right_top_color,left_bottom_color,right_bottom_color)}
   --主要颜色渐变
2 {\2vc(left_top_color,right_top_color,left_bottom_color,right_bottom_color)}
   --次要颜色渐变
3 {\3vc(left_top_color,right_top_color,left_bottom_color,right_bottom_color)}
   --边框颜色渐变
4 {\4vc(left_top_color,right_top_color,left_bottom_color,right_bottom_color)}
   --阴影颜色渐变
```

```
1 {\1vc(&H000000&,&HFFFF00&,&HFF00FF&,&FFFFFF&)}TEST
2 --整体字幕渐变
```



```
1 {\1vc(&HC0C0C0&,&H46284C&,&&HAF2C80&&,&FFFFFF&)}T{\r\1vc(&HC6A4BA&,&H956648&,&HFF00FF&,&FFFFFF&)}E{\r\1vc(&HAF2C80&,&HC0C0C0&,&HFF00FF&,&HA327BA&)}S{\r\1vc(&H21229B&,&H956648&,&HFF00FF&,&H45775B&)}T
2 --单字符单渐变颜色
```



Add 补充

1. 参数

- $\$$ 表示4个颜色作用类型，即¹主要颜色、²次要颜色、³边框颜色、⁴阴影颜色
- 其4个参数表示为——左上颜色，右上颜色，左下颜色，右下颜色；其表现为，4个顶点颜色为设置颜色，然后依次向其他三个顶点的颜色^{渐变}过渡。
- 4个参数需要填入标准ASS颜色格式(同 \c 一样，非标准格式当然也能识别，但是最好不要)。

示例 3

示例 2

2. 能通过重置样式(\r)对单个字符进行渐变；否之，则是整条字幕进行整体渐变。

3. 4个参数可以是相同的，即不是只能制作4色渐变，2~4色渐变都是支持的。

渐变 opacity levels

```
\$vc(left_top_transparency,right_top_transparency,left_bottom_transparency,right_bottom_transparency)
```

Creates gradients by using opacity levels. May be slow. Animatable by `\t`.

用透明度创建渐变。可能导致渲染时比较慢。

可配合 `\t` 实现动态效果。

Example

示例

```
1 | {\1va(left_top_transparency,right_top_transparency,left_bottom_transparency,r
   | ight_bottom_transparency)} --主要颜色透明度渐变
2 | {\2va(left_top_transparency,right_top_transparency,left_bottom_transparency,r
   | ight_bottom_transparency)} --次要颜色透明度渐变
3 | {\3va(left_top_transparency,right_top_transparency,left_bottom_transparency,r
   | ight_bottom_transparency)} --边框颜色透明度渐变
4 | {\4va(left_top_transparency,right_top_transparency,left_bottom_transparency,r
   | ight_bottom_transparency)} --阴影颜色透明度渐变
```

```
1 | {\1va(&H00&,&HF2&,&HC2&,&H00&)}TEST
```



```
1 | {\1vc(&H000000&,&HFFFF00&,&HFF00FF&,&FFFFFF&)\bord0\1va(&H00&,&HFF&,&HFF&,&H0
   | 0&)}TEST
2 | --无边框、渐变透明度、渐变颜色
3 | {\1vc(&H000000&,&HFFFF00&,&HFF00FF&,&FFFFFF&)\bord0\4vc(&H5C5C7E&,&H8ABE80&,&
   | HA3965F&,&FFFFFF&)}TEST
4 | --无边框、无透明度、渐变主颜色、渐变阴影颜色
5 | {\1vc(&H000000&,&HFFFF00&,&HFF00FF&,&FFFFFF&)\bord0\1va(&H00&,&HFF&,&HFF&,&H0
   | 0&)\4vc(&H5C5C7E&,&H8ABE80&,&HA3965F&,&FFFFFF&)}TEST
6 | --无边框、渐变透明度、渐变主颜色、渐变阴影颜色
```



Add 补充

1. 参数

- `$` 表示4个作用颜色类型，即主要颜色¹、次要颜色²、边框颜色³、阴影颜色⁴
- 其4个参数表示为——左上透明度，右上透明度，左下透明度，右下透明度
- 4个参数需要填入标准ASS透明度格式 (同 `\a` 一样，非标准格式当然也能识别，但是最好不要)。

2. 能通过重置样式(`\r`) 对单个字符进行渐变；否之，则是整条字幕进行整体渐变。示 例 2

3. 4个参数可以是相同的，即不是只能制作4色渐变，2~4色渐变都是支持的。

4. 可以通过透明度渐变+阴影达到一些奇怪的颜色混合，合适地使用能够让字幕更有质感(示例里随便填的颜色)示 例 3

Shaking 抖动

```
\jitter(left,right,up,down,period[,seed])
```

It performs subtitle position shaking. First four parameters adjust maximum offset in every direction, fifth parameter sets shaking period in milliseconds, sixth parameter sets the initial seed for random number generator so the form shaking will not change upon calls. Animatable by `\t`.

这个标签使字幕产生抖动。

前四个参数调整在各个方向的最大偏移量，第五个参数设置在毫秒内的抖动周期，第六个参数设置为随机数发生器的初始种子，使之震动的形式保持一致。

可配合 `\t` 实现动态效果。

Example 示例

该标签为动态标签，无法进行图片演示，望下例能自行进行复制复习查看

```
1 {\jitter(5,5,5,5,50)}TEST
2 --进行偏移量为5像素，每50ms抖动1次的运动
```

```
1 {\jitter(5,5,5,5,50)}TEST_1
2 {\jitter(5,5,5,5,50)}TEST_2
3 --进行偏移量为5像素，每50ms抖动1次的运动，但是两行字幕分别进行不同的抖动
```

```
1 {\jitter(5,5,5,5,50,100)}TEST_1
2 {\jitter(5,5,5,5,50,100)}TEST_2
3 --进行偏移量为5像素，每50ms抖动1次的运动，两行字幕的抖动一致
```

```
1 {\jitter(5,5,5,5,50)\t(1000,1000,\jitter(0,0,0,0,50))}TEST
2 --在0~1000ms内，进行偏移量为5像素，每50ms抖动1次的运动
3 --进行上述相同的抖动，但是规定了时间
```

```
1 {\jitter(0,0,0,0,50)\t(0,1000,\jitter(5,5,5,5,50))}TEST
2 --在0~1000ms内，从不抖动步进到进行偏移量为5像素，每50ms抖动1次的运动，并在
1000ms~end进行偏移量为5像素，每50ms抖动1次的运动
```

```
1 {\t(0,1000,1.5,\jitter(5,5,5,5,50))}TEST
2 --在0~1000ms内，从不抖动加速步进到进行偏移量为5像素，每50ms抖动1次的运动
```

Add 补充

1. 本标签会让字幕以 `<period>` 周期，向左端 `<left>` 个像素，右端 `<right>` 个像素，上方 `<up>` 个像素，下方 `<down>` 个像素为最大偏移量进行随机抖动，抖动范围为——以文本主体为原点，在一个左上顶点为 $(!$left-<right>!,$stop-<up>!)$ ，右下顶点为 $(!$right+<right>!,$bottom+<down>!)$ 的矩形内随机移动。将整个文本作为对象

2. 参数

- `<left>`：字幕向左端移动的**最大偏移量**。
- `<right>`：字幕向右端移动的**最大偏移量**。
- `<up>`：字幕向上方移动的**最大偏移量**。
- `<down>`：字幕向下方移动的**最大偏移量**。
- `<period>`：字幕的抖动周期，数值越大抖动速度越慢。其关系对应为， $period = 1000 \div < period > \text{次} / s$
- `<seed>`：随机数种子。可选参数。

3. 注解

1. 前4个参数表示能够偏移的最大像素量，表示的是偏移范围，不代表每次移动就是输入的参数。
2. `<period>` 表示目标抖动周期，其算法为——`<period>` 毫秒抖动1次($\frac{1000}{<period>}$ 次/s)
示例 4、示例 5
3. `\jitter` 标签没有时间控制参数，但是能够通过 `\t` 标签实现时间控制。
可以观察到，在示例代码中，我们在进行 `\t` 时间控制的时候，会将周期统一，而不是直接去进行动态变化，这里面涉及到 `\t` 标签里面的参数具体实现，简单来说就是如果像那样没有控制周期就进行动态变化，会导致运算量过多从而可能导致软件崩溃，渲染压力也会相应地提高，所以最好是控制一下周期。
4. 同样，既然能够使用 `\t` 标签实现动画效果，那么当然也可以进行变速变化。

图片替代颜色填充

```
\img(path_to_png_file[,xoffset,yoffset])
```

Replaces color fill with repeated image pattern. Parameters are slashed (/) path to image and optional fill's base offset. Path may be relative to current subtitle file location. First VSFilterMod loads attached images and if fails tries to find them locally. VSFilterMod supports only 24 or 32 bit truecolor pngs with or without transparency channel. Offset is animatable by \t. Note that \be and \blur tag will not blur image but only mask which is used to place fill.

将颜色替换填充为重复的图片图案。

参数为图片路径（用(/)分隔）和可调整填充的基本偏移量。

路径可以是相对于当前字幕文件的位置。

首先VSFilterMod加载附加的图片，如果失败尝试在本地找到他们。

VSFilterMod只支持24或32位带或不带透明通道的真彩色png格式的图片。

偏移量可配合\t实现动态效果。

请注意 \be 和 \blur 不能模糊图片，只能模糊被图片填充的那个边框。

Example

示例 定制化 由于路径过于私人化，所以只有简单说明和图片演示

```
1 | {\1img(path_to_png_file[,xoffset,yoffset])} --将主要颜色替换成图片
2 | {\2img(path_to_png_file[,xoffset,yoffset])} --将次要颜色替换成图片
3 | {\3img(path_to_png_file[,xoffset,yoffset])} --将边框颜色替换成图片
4 | {\4img(path_to_png_file[,xoffset,yoffset])} --将阴影颜色替换成图片
```

```
1 | {\1img(D:\exam\gradients.png)}아절한 너의 숨소리는
2 | --使用绝对路径挂载图片作为文本背景颜色
```



```
1 | {\p1\an7 \bord0\shad0 \1img(gradients.png)}m 0 0 1 1024 0 1024 865 0 865
2 | --使用相对路径挂载图片
```



```

1 | {\p1\an7 \bord0\shad0 \img(logo-aegisub.png)}m 0 0 1 1024 0 1024 865 0 865
2 | --不使用符合图片分辨率的绘图挂载图片

```



Add 补充

1. 参数

- `<path_to_png_file>` : 填入图片路径，图片路径可以是绝对路径也可以是相对路径。
示例 3
- `<xoffset>` : x轴偏移，默认值为0像素。可选参数。
不偏移
- `<yoffset>` : y轴偏移，默认值为0像素。可选参数。
不偏移

注意 2. 技巧

- 通常挂图 示例 3 时都会将边框、阴影全部设置为0，然后使用 `\img`。
{bord0} {shad0}
- 挂图时，通常和绘图的基本规则一样，使用 `m 0 0` 点作为基点，然后绘制一个和挂载图片分辨率一样大小的矩形。
示例 3
- 可以通过挂载一张大的纯色背景图当做文字颜色。
渐变、花纹... 示例 2
- 不要使用 `<xoffset>` `<yoffset>` 这两个参数，如果想控制挂图的移动，请使用 `\fspv \fshp \move...` 移动标签。
大概率
 - 首先，这两个参数的功能非常的不直观，很难去控制，如果想进行图片的定位，直接使用 `\pos` 会更加直观；
 - 其次，这两个参数也无法控制图片的移动，还是需要移动标签辅助；
 - 最后，这两个参数可能会导致图片丢失。

3. 概念明晰

1. `\img` 这个标签的本质其实是——将颜色换成图片，加载进去的图片是承担了颜色的角色，而不是真正插入进入了一张图片，所以格式中有代表颜色对象的 `$`，但是这个外挂进去的图片只是渲染在最顶层，并不是真正变成其文字颜色，所以导致了说明里提到的第四点问题。
无法模糊图片

2. 由 1 可得，`\img` 并不一定只能用在绘图中，还可以当做文字背景。
示 例 2

3. `\img` 挂图只能加载png格式的图片，不能加载像jpg、svg之类的其他格式的图片。
24 或 32 位

4. `\img` 支持透明通道的png图片，但是挂载时需要将边框设置为0，否则其透明背景会被自动填充为边框颜色。
{\bord0}

5. `\img` 无法改变插入图片的大小、形状，它只会以图片原来的形式被渲染出来。
分辨率

当一张图片是200x200像素时，它只会以200x200的大小被渲染出来。

若文本小于200x200，则只会渲染出合适文本大小的那一部分；若文本大于200x200，超出部分会以额外图片的形式渲染出来。
多张无缝相同图片 示 例 4

6. 挂图的起始位置点规则不明，但是在单个绘图(图片分辨率与绘图大小一样)时，其位置是符合绘图的基点规则的。
(0,0)

4. bug说明

1. `\img` 这个标签并不完善，所以对`vsfiltermod`的版本要求特别高，所以当你使用`\img` 标签出错的时候，请首先检查`vsfiltermod`的版本(包括压制软件中的`vsfiltermod`)
需求 bug 更少的版本
下列问题

- 无法使用相对路径加载图片
- 参数丢失、图片丢失、图片无法加载
- 压制报错、压制失败

2. 虽然不清楚本版本的`vsfiltermod`对于相对路径的实现兼容到了哪一部分，但是如果使用了相对路径而且出现上述错误的情况，请将相对路径更换为绝对路径之后再查看是否为相对路径导致的错误。
目前本人使用没出错过

3. 尽管如此，很 少仍然不能避免某些奇怪原因导致的bug，它还可能与你的电脑里的某些进程冲突导致bug，maybe如果还是无法解决，请将你的任务移交给其他人。
上述方法尝试之后

边界变形

`\rnd<arg>`

`\rndx<arg>`

`\rndy<arg>`

`\rndz<arg>`

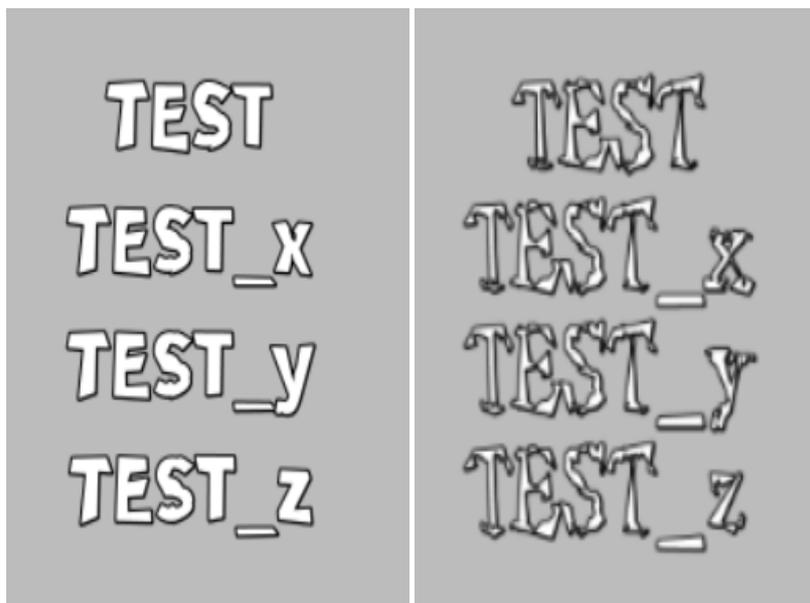
Moves border points on random number of pixels from $(-arg, arg)$ interval in selected direction. Animatable by `\t`.

在选定的区间内 $(-arg, arg)$ 随机移动边界上的像素。
可配合`\t`实现动态效果。

Example

示例

```
1 {\rnd30}TEST
2 {\rndx30}TEST_x
3 {\rndy30}TEST_y
4 {\rndz30}TEST_z
5 --左1图为思源黑体Heavy 100字号；右1图为仿宋 50字号
```



```
1 {\t(0,2000,\rnd20)\t(2000,4000,\rndy100)}TEST
2 --动态扭曲效果
```

Add
补充

这个标签没什么需要补充的

1. 可以简单模拟一些边界扭曲的日语字体。
示例 1 右图
2. 使用细一些的字体效果会更好，反之，字体越粗效果越不明显。
示例 1 左图

Distortion

扭曲

```
\distort(right_top_x,right_top_y,right_bottom_x,right_bottom_y,left_bottom_x,left_bottom_y)
```

Distorts text by moving corner pins to specified relative coordinates. Animatable by `\t`.

扭曲文字是通过移动角上坐标具体确定相对的坐标。

可配合 `\t` 实现动态效果。

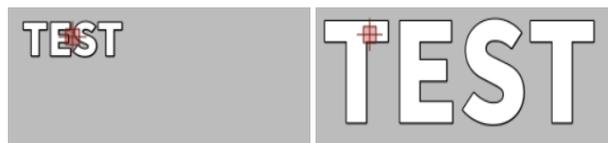
Example

示例

- 1 `{\distort(1,0,1,-1,0,1)}TEST`
- 2 --右上左下顶点不动，将右下顶点移动到右上顶点的上方(保持原高度)，输出一个'拧'着的文本，属于两个全等三角形



- 1 `{\distort(3,0,3,3,0,3)}TEST`
- 2 --将文本放大3倍（左1图为无标签对比图，右1图为效果图）



Add 补充

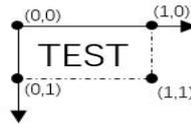
1. 该特效标签能使文字变形。

`\distort` 将一行字幕当做一个矩形对象，通过修改矩形的三个顶点以使字幕变形（类似于 ASSDraw3 中的 `bilinear` 功能）。

右上、右下、左下

2. 参数

1. `\distort` 将字幕对象视为一个矩形，以左上顶点为原点，规定左上顶点水平向右方向为x轴的正方向，左上顶点竖直向下方向为y轴的正方向，即左上角的顶点为(0,0)，右上顶点为(1,0)，右下顶点(1,1)，左下顶点(0,1)。



■ 即其规定为，x轴方向的1个单位长度为 `!line.width!`，y轴的一个单位长度为 `!line.height!`，虽然在坐标点表示上好像是个正方形，但是x轴和y轴两个方向的计量单位是不同的。

- 左上顶点无法被修改，也就是说，只能修改右上、右下、左下的顶点。
- 参数里面的数值为倍数关系，即1表示的是100%，2表示的是200%
- 参数可以是负数，也可以是小数

2. 前两个参数为右上顶点的x坐标和y坐标，中间两个参数为右下顶点的x坐标和y坐标，最后两个参数为左下顶点的x坐标和y坐标，左上顶点无法修改，即

`{\distort(1,0,1,1,0,1)}` 等于原字幕/图形。

- `<right_top_x>`：右上顶点x坐标
- `<right_top_y>`：右上顶点y坐标
- `<right_bottom_x>`：右下顶点x坐标
- `<right_bottom_y>`：右下顶点y坐标
- `<left_bottom_x>`：左下顶点x坐标
- `<left_bottom_y>`：左下顶点y坐标

3. `\distort` 变形对文本的对齐方式没有任何改变，即如果将一行字幕等比例

放大3倍 `{\an5\distort(3,0,3,3,0,3)}`，原来定位点在字幕最中间，由于变形后定位点还是在原来位置，但是字幕向右下角延展放大了，即定位点实际显示起来是在字幕的左上角。

无法修改左上顶点，故只能向其他三个方向延长

Baseline obliqu

基线倾斜

`\frs<angle>`

Character's baseline obliquity. Rotation anchor depends from style definition and `\an` tag. Animatable by `\t`.

文字的基线倾斜。

选择方向取决于style设定和 `\an` 标签(即对齐方式)。

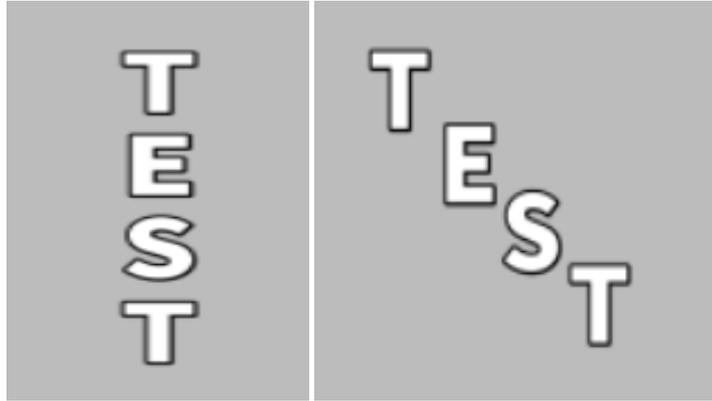
可配合 `\t` 实现动态效果。

Alignment

Example

示例

```
1 | {\frs90\frz-90\fsp20}TEST
2 | {\frs45\frz-45\fsp30}TEST
```



```
1 | {\t(\frs360\frz-360)}TEST
2 | --虽然移动的轨迹很奇怪，但其文字确切是始终维持在正方向的。
```

Add

补充

1. 实际上并没有感觉对齐方式对基线的旋转有影响。
2. 通常 `\frs` 会与 `\fsp` 连用，使修改后的字幕间距更为合适。
3. `\frsφ` 与 `\frz-φ` 同时使用，可以使字幕基线维持在正方向。

Z coordinate

Z 坐标

`\z<arg>`

Sets z coordinate. It may be signified as a distance from screen to text. It's noticeable in case of using `\frx` and `\fry` tags. Animatable by `\t`.

设置Z坐标。它可能表示文字到屏幕的距离(模拟Z轴)。

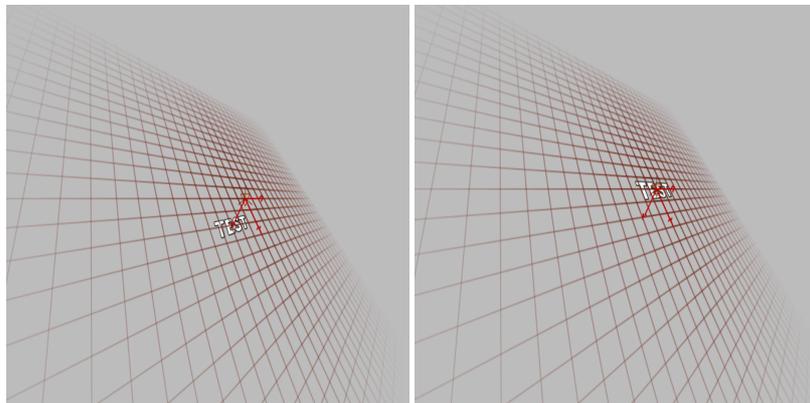
结合 `\frx` 和 `\fry` 效果更显著。

可配合 `\t` 实现动态效果。

Example

示例

```
1 | {\frx45\fry45\z90}TEST
2 | --左1为效果图（右1为无z标签对比图）
```



```
1 | {\z100\t(\z0\fry360)}TEST
```

Add

补充

1. 当单独使用 `\z` 标签时，其效果与 `\fsc` 类似。

2. 通过 `\z` 标签可以实现视觉逼近的效果。
示例 2

极限移动

```
\mover(x1,y1,x2,y2,angle1,angle2,radius1,radius2[,t1,t2])
```

It works like `\move` but now it's possible to use rounded, oval or spiral trajectories.

这个标签与 `\move` 的工作原理一样，但是可以实现圆形、椭圆形或者螺旋形轨迹运动。

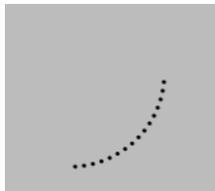
Example

示例

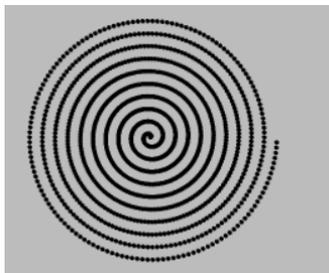
```
1 {\move(100,500,1000,500)}
2 {\mover(100,600,1000,600,0,0,0,0)}
3 --两个标签等效
```



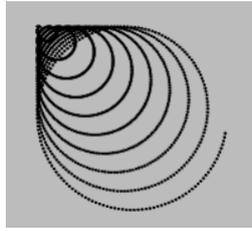
```
1 {\mover(500,500,500,500,-90,0,150,150)}
2 --以点(500,500)作为圆心，沿着一个半径为150像素的圆转过(-90,0)角度
```



```
1 {\mover(500,500,500,500,0,3600,0,200)}
2 --以点(500,500)作为圆心，从圆心开始向外旋转扩散3600°，半径从0不断延长至200像素
```



- 1 `{\mover(500,500,700,700,0,3600,0,200}`
- 2 `--`圆心从点(500,500)移动到(700,700),其中不断向外旋转扩散3600°,半径从0不断延长至200像素



Add 补充

1. 参数

- `<x1>` `<x1>`: 开始时**圆心坐标**, 单位是像素
- `<x2>` `<y2>`: 结束时**圆心坐标**, 单位是像素。
- `<angle1>`: 开始时**旋转角度**, 单位是度。
- `<angle2>`: 结束时**旋转角度**, 单位是度。
- `<radius1>`: 开始时**旋转半径**, 单位是像素。
- `<radius2>`: 结束时**旋转半径**, 单位是像素。
- `<t1>`: 开始时间, 默认值为行开始时间。可选参数。
- `<t2>`: 结束时间, 默认值为行结束时间。可选参数。

与 \move 不同

2. vsmo^d提供的移动标签(`\mover` `\moves` `\movevc`), 都无法在aeg中渲染出路径。

本条目示例中显示的路径, 是通过其他手段²获取的。

3. 注解

1. 开始坐标和结束坐标可以填入相同的坐标; 也可以填入不同坐标。
示例 2、示例 3 示例 4
2. 旋转半径可以填入相同的长度; 也可以填入不同的长度。
示例 2 示例 3、示例 4

Spline-move 曲线移动

```
\moves3(x1,x1,x2,y2,x3,y3[,t1,t2])
```

```
\moves4(x1,x1,x2,y2,x3,y3,x4,y4[,t1,t2])
```

It moves subtitle by spline curve trajectory. Functions with three or four base points are available, they produce cubic or bicubic Bezier curve trajectory.

这个标签用曲线轨迹移动字幕。

函数可以带三或四个基点，它们产生二次或三次Bezier曲线轨迹。

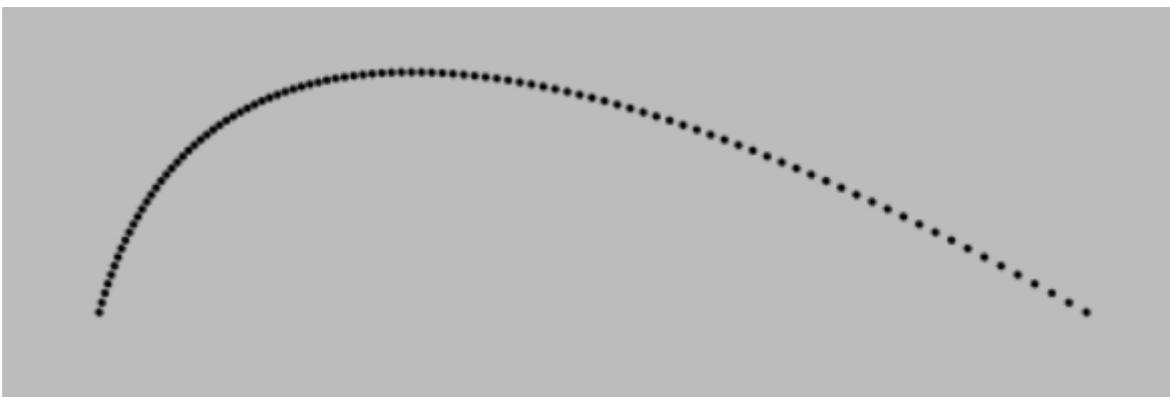
Example

示例

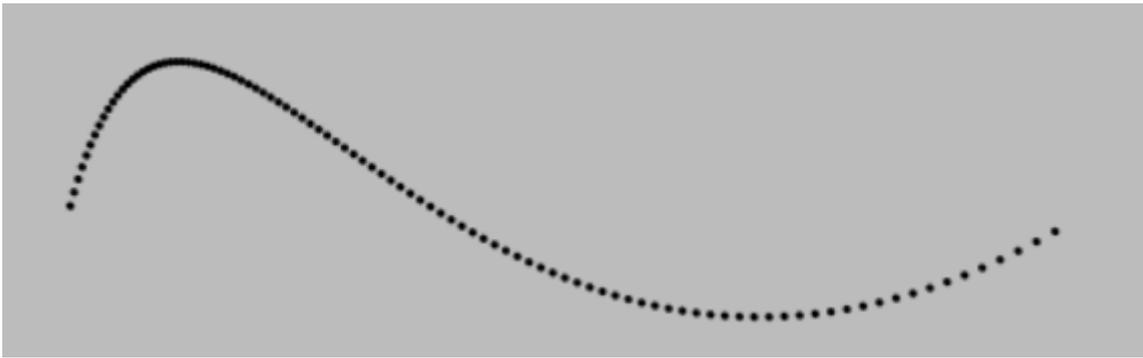
```
1 {\move(200,500,1000,500)}  
2 {\moves3(200,600,300,600,1000,600)}  
3 {\moves4(200,700,600,700,700,700,1000,700)}  
4 --使用曲线运动实现变速直接运动
```



```
1 {\moves3(200,700,300,300,1000,700)}  
2 --二阶贝塞尔曲线移动
```



```
1 | {\moves4(200,700,300,300,500,1000,1000,700)}
2 | --三阶贝塞尔曲线移动
```



Add 补充

1. vsmo^d提供的移动标签(`\mover` `\moves` `\movevc`), 都无法在aeg中渲染出路径。
与 \move 不同
本条目示例中显示的路径, 是通过其他手段²获取的。
2. 和 `\move` 类似, 在一行字幕内, 只能同时生效一个标签, 且与 `\pos` `\move` `\mover` 不能同时使用。
Spline-move 示 例 1
3. 曲线移动能够实现变速直线运动。
不能填入两个相同坐标
4. `\moves` 填入的坐标不能是同一个坐标, 否则会无法正常移动。

矢量clip移动

`\movevc(x1,y1)``\movevc(x1,y1,x2,y2[,t1,t2])`

It moves vector clips (`\clip`, `\iclip`) independently to subtitles (unaffected by `\move` or `\pos`). Parameters are same to `\move`. Pixel precision.

它可以独立移动字幕上的矢量clip(`\clip`, `\iclip`), 可同时使用 `\move` 或 `\pos`。
参数和 `\move` 相同。
精准到像素。

Example

示例

```

1 {\clip(m 100 100 1 200 100 200 200 100 200)\p1\an7\pos(0,0)}m 0 0 1 1920 0
  1920 1080 0 1080
2 {\clip(m 100 100 1 200 100 200 200 100 200)\p1\an7\pos(0,0)\movevc(100,100)}m
  0 0 1 1920 0 1920 1080 0 1080

```



```

1 {\an5\clip(m 100 100 1 200 100 200 200 100
  200)\move(100,150,200,250)\movevc(0,0,100,100)}TEST
2 --只显示「ST」的移动

```

Add

补充

1. 参数

- 当只有2个参数时, 相当于 格式 1 矢量clip的 `\pos`
- 当有4个参数(及更多)时, 相当于 格式 2 矢量clip的 `\move` (但是使用的是相对坐标)

2. 注解

- 无法移动普通遮罩(普通遮罩使用 `\t` 进行移动), 只能移动矢量遮罩。
- 虽然说明写着参数与 `\move` 相同, 但是其本质是完全不同的。
`\move` 使用的坐标是绝对坐标 (或者说其他的所有使用到坐标的移动标签) 使用的都是绝对坐标, 只有 `\movevc` 使用的是相对坐标, 即其参数表示的不是 (x_1, y_1) 这一个坐标, 实际上表示的是 $(\Delta x, \Delta y)$, 是原矢量遮罩的坐标距移动后位置的距离。
示例 1
- 和 [安装使用](#) 说的一样, 「Aegisub **不识别新选项标签**」, 也就是说, 如果同时使用了 `\pos` 或 `\move` 标签, 则需要手动输入 `\pos` 参数或 `\move` 参数。否则, `\movevc` 标签会被替换消失掉。

正交投影

`\ortho0`

`\ortho1`

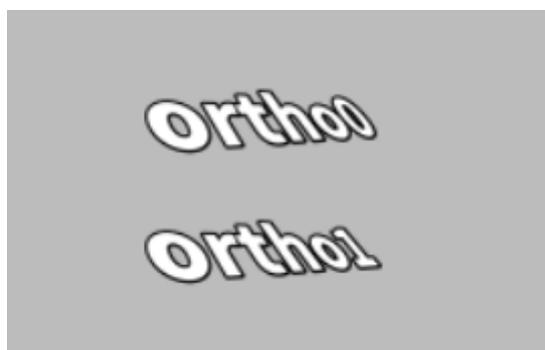
方便制作3d效果，配合`\frx` `\fry` 使用。

该标签为新mod标签。

Example

示例

```
1 | {\orhto0\frx45\fry45}ortho0  
2 | {\orhto1\frx45\fry45}ortho1
```

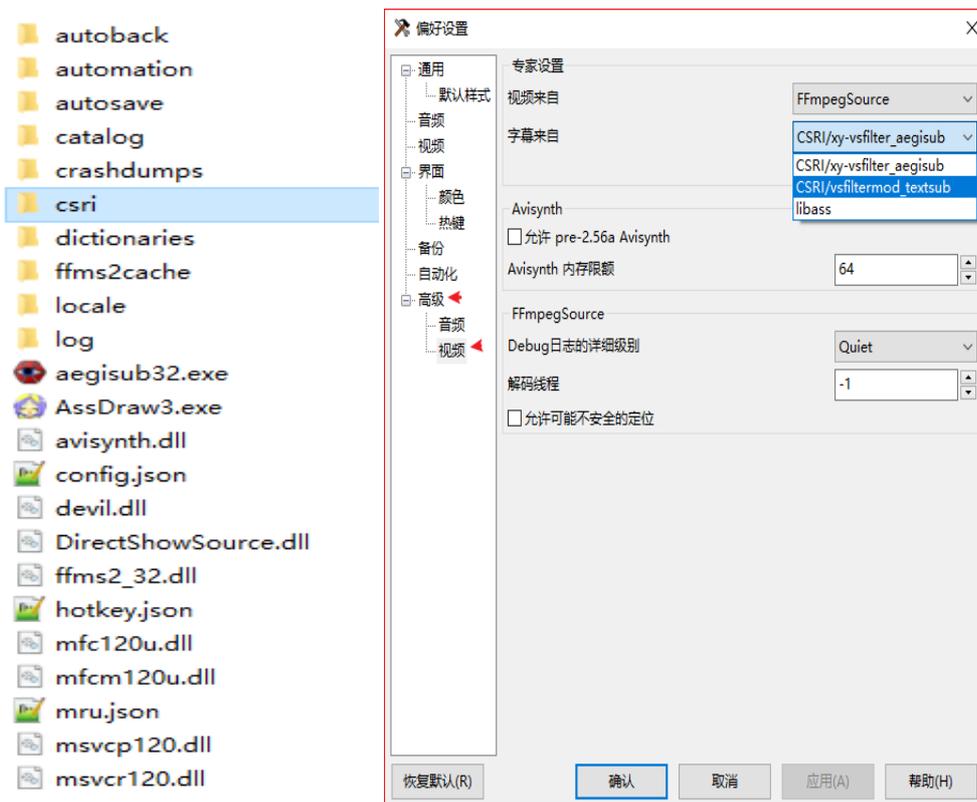


安装与使用

对于Aegisub

Copy *VFilterMod.dll* to `[..\Aegisub\csri]` directory, than start **Aegisub**, open Options dialog, select Video tab and change Subtitle provider to `csri/vsfiltermod_textsub`.

复制*VFilterMod.dll*到`[..\Aegisub\csri]`目录下, 然后运行Aegisub, 打开【查看】→【选项】, 选择【视频】, 然后修改【字幕来自】为`vsfiltermod_textsub`。



Important Note: Aegisub doesn't know about new override tags so it will delete or replace them with known ones when you will try to use Visual Typesetting tools.

文本编辑栏快捷键

注意: Aegisub不识别新选项标签, 所以当你使用【可视排版】时, 它可能会删除或把你输入的替换为已知项。

对于Avisynth

Copy *VFilterMod.dll* to default Avisynth plugins directory (`C:\Program Files\Avisynth 2.5\plugins`) or load it with `LoadPlugin()` function. Mod provides similar to `VFilter` functions but with "Mod" suffix e.g. `TextSubMod()`.

复制*VFilterMod.dll*到默认Avisynth插件目录(`C:\Program Files\Avisynth 2.5\plugins`)或者用`LoadPlugin()`函数载入

在使用ass字幕时用`TextSubMod()`函数。

直接使用

To use VSFilterMod with video players first of all you should unregister installed version of VSFilter. Find where it's located on your disc and execute "regsvr32 /u Path\to\VSFilter.dll". Then place VSFilterMod.dll to some suitable for you place and execute "regsvr32 Path\to \VSFilterMod.dll".

如果想要通过播放器使用 VSFilterMod，首先你应该卸载以前的版本（这一段可以忽略）.....直接把VSFilterMod.dll复制到（系统盘\windows\system32\）下就行了（注意，一些标签是无法在播放器里被直接渲染出来，只能通过压制方法查看）。

使用VirtualDub

Copy VSFilterMod.dll to VirtualDub plugins directory and change its extension to vdf (it should be VSFilterMod.vdf now). Then select TextSubMod from filters and work as usual.

复制VSFilterMod.dll到VirtualDub插件目录，改变它的扩展名.vdf。（也就是说，它的名字现在应该是VSFilterMod.vdf）。然后选择TextSubMod，像往常一样使用即可。

参考资料

1. VFilterMod中文说明文档
2. <https://code.google.com/archive/p/vsfiltermod/wikis/NewTags.wiki>
3. https://github.com/qwe7989199/aegisub_scripts/tree/master/VFilterMod_bin
4. https://aegi.vmoe.info/docs/3.2/ASS_Tags/

尾注

1. 本文档中所出现的「挂图」特指「想要挂载一张图片」，而不是将图片作为颜色对象使用的情况。 [↩](#)

2. 其他手段：^{特效}指使用模板/函数手段将移动的轨迹转化为多个点(点越密集表示速度越慢)，模拟移动轨迹。 [↩](#) [↩](#)