

# Python-多媒体

2022年5月24日 3:25

## 1. moviepy

moviepy.editor: 以下函数除读写操作外均有返回值

视频操作:

读取视频: `vid = VideoFileClip("XXX.mp4")`

视频剪辑:

`vid.subclip(10, 60)`: 截取00:00:10~00:00:60部分 (秒)

`vid.set_start(5)`: 从5秒开始 (前5秒是黑的)

`vid.crossfadein(1)`: 1秒渐入

画面调位:

`vid.set_pos((100, 100))`: 设置位置, 距左上角的像素距离

`set_pos("center")`: 正中

`set_pos(("center", "top"))`: 水平方向居中, 垂直方向居顶

`set_pos(("left", "center"))`: 水平方向居左, 垂直方向居中

`set_pos((0.4, 0.7), relative=True)`: 0.4倍宽, 0.7倍高处

`set_pos(lambda t: ('center', 50+t))`: 水平居中, (按时间) 向下移动! (滚动字幕!)

画面缩放:

`vid.resize(width="360")`: 按宽度调整

`vid.resize(0.6)`: 按比例缩放

保存画面&画面遮罩:

`vid.save_frame("first_frame.jpg")`

`vid.save_frame("frame.png", t=2)`

`vid_mask = ImageClip("processed_first_frame.png", ismask=True)`

`vid.set_mask(vid_mask)`

音频操作:

`vid.volumex(0.8)`: 音量×0.8

`vid.set_audio(aud)`: 设置音频

`vid.without_audio()`: 去掉音频

倍速播放:

`vid.speedx(2)`: 2倍速

混合fx: `vid.fx(effect1, args1).fx(effect2, args2).fx(effect3, args3)`

`vid.fx(vfx.black_white)`: 黑白变化

`vid.fx(vfx.mirror_x/vfx.mirror_y)`: 镜像翻转

输出文件:

`new_vid.write_videofile("new_XXX.avi", fps=25, codec="mpeg4",  
audio_bitrate="1000k", bitrate="4000k")`

音频操作:

读取音频: `aud = vid.audio`

输出文件:

`aud.write_audiofile(  
filename, fps=None, nbytes=2, buffersize=2000, codec=None, bitrate=None,  
ffmpeg_params=None, write_logfile=False, logger='bar'  
)`

fps: 若为None, 则设置为self.fps

nbytes: 若为16-bit音频, 设为2; 若为32-bit音频, 设为4

codec: 若为16-bit wav, 设为"pcm-s16le"; 若为32-bit wav, 设为"pcm\_s32le"; 根据后缀名推测

bitrate: 决定文件的大小和质量, 但只是一个目标, 实际上不一定实现, 如"50k", "500k", "5000k"

文本框操作:

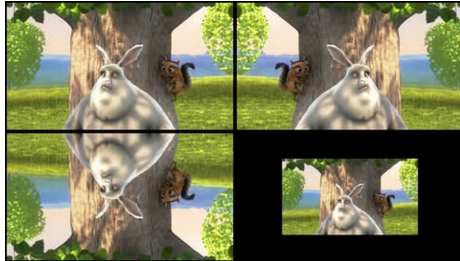
```
txt = TextClip("Wow!", fontsize=20, color="yellow")
txt.set_pos("center").set_duration(10)          # 显示10秒
```

混合操作:

画面混合: `new_vid = CompositeVideoClip([vid1, vid2, vid3])`

vid1在最下面, vid3在最上面

同屏排列: `new_vid = clips_array([[vid1, vid2], [vid3, vid4]])`



画面连接: `new_vid = concatenate_videoclips([vid, vid], transition=trans_vid)`

transition为过渡动画

音频混合: `new_aud = CompositeAudioClip([aud1, aud2, aud3])`

音频连接: `new_aud = concatenate_audioclips([aud1, aud1, aud1])`