VCB-Studio教程22 30fps片源处理(1)

本教程旨在讲述常规的30fps片源处理。其中涉及到ivtc/deint两种基础操作。

1. NTSC vs PAL

众所周知，视频最初起源于电视节目。电视依赖交流电供电，最早一批的电视能周期性显示的帧率，与交流电频率相关。目前世界上主要存在两种交流电频率，一种是50Hz，比如中国大陆与欧洲，一种是美国与日本流行的60Hz。由此诞生出两种不同的视频制式，NTSC vs PAL。

这两个制式的区别详细展开非得说上两千个字；这里大家有兴趣自己去wikipedia拜读。今天要说的，主要是两者帧率上的区别。PAL依赖50fps的交流电，其帧率一般为50fps和25fps；NTSC依赖60Hz的交流电。其帧率多为24fps/30fps/48fps/60fps。我们接触到的日本动漫BD与DVD，几乎清一色都是NTSC制式的。

2. 30fps源的识别和读取

1920x1080的蓝光片源，对视频的分辨率和帧率有着严格的限制。在HEVC时代之前，NTSC制式的蓝光一般只支持3种帧率：

24000/1001 = 23.976 fps

24000/1000 = 24.000 fps

30000/1001 = 29.970 fps

前两者统称24fps，后者称为30fps。

24fps的视频，通常在mediainfo里面显示的信息是逐行扫描(Progressive):



如果是逐行扫描的视频，通常无需做任何针对扫描方式的处理。这种视频我们称之为24p(p=progressive)。

但是30fps视频，mediainfo里面多显示隔行扫描(Interlaced):



Interlaced的视频，在支持隔行扫描的tv显示器上播放是没有问题的，但是在不支持的PC显示器上，如果不作处理，播放的时候很可能会出现拉丝现象：



为了避免这种场景的产生，Interlaced的片源必须非常谨慎的处理。但是30fps的视频，又不一定是特定的类型，其类型的多样性，让针对隔行扫描的片源处理非常复杂。本教程将分类讲述每个类型的识别和处理。

需要注意的是，不少30 fps片源采用的是soft pulldown的形式，即24p的原生视频，通过加注一个“pulldown flag”，让播放器在播放的时候事实做成30fps。这类视频在蓝光和DVD里，经常与真·30fps视频混合，使得处理起来格外困难。我们会在下一章额外介绍它。

3、片源的载入

30fps片源载入，依旧可以使用LWLibavVideoSource。不过参数里必须增加一项：repeat=True

repeat=True可以强制soft pulldown的片源，在解码后完整的输出30fps。

没有这个标签，soft pulldown的部分就会被直接解码为24p。如果视频中有非soft pulldown的30fps部分，会使得整个视频以vfr的形式载入。而avs和vs均没有原生的vfr支持，这样，你会得到一个帧率完全随机，帧数完全不可靠的视频。

而强制这个标签，最坏的结果也是得到了全程30fps，但是内容不太相同的混合clip。这样虽然棘手，至少可以做。

只有一个情况可以拿掉repeat=True：全程24p，soft pulldown。判断条件为：

开启repeat=True，帧数为x

不开启，帧数为y

| 0.8 \* x – y| < 1

这个意思是30fps直接变24fps，帧数误差在1帧以内（这个周期是每5帧变为4帧，如果帧数总和不是5的倍数，那么有些周期无法完整变换，不可能完美切合），那么可以判定这是个全程soft pulldown的视频。这时候可以直接当做24p制作。

30fps视频有个“场序”(field order)的概念，指的是每一帧的上下两个场，在视频中，是顶场先于底场显示，还是底场先于顶场显示。绝大多数视频都是顶场优先，Top Field First，TFF。

4、Progressive伪装成的Interlaced: 30p

如果你对于手上这个视频左看右看看不到拉丝，也没有规律性的重复帧，说明这个视频是Progressive的。它是原生30fps的逐行扫描视频。只不过蓝光规定，30fps只能以interlaced的标示呈现，所以伪装成了Interlaced而已。30p是最简单的30fps视频；不需要做任何针对性的处理。

VS里最好加一个std.SetFieldBased(src,0)，这样是告诉vs，这是一个progressive的视频。否则有些滤镜可能会误认为这是interlace源，采用错误的算法。

需要注意的是，日本动漫基本都是24p制作，30p的动漫内容（特别是正片24p，特典里出现了30p动漫）极其罕见。这时候一定要再次确认你是不是开了repeat=True，以及是不是24d（见下文）。

5、每4帧插入一个重复帧构成的30fps: 24d

如果你对于手上这个视频左看右看看不到拉丝，但是有规律性的重复帧，表现在至少一段视频，5帧内周期性的出现一帧重复帧，说明这个视频是24d（d=duplicate, 重复）的。它是原生24p的视频，通过添加重复帧构成的逐行扫描视频。然后伪装成了Interlaced而已。24d的视频处理只需要删除重复帧就可以了：

tdecimate(mode=0/1)。tdecimate是用于删除重复帧的滤镜：<http://avisynth.org.ru/docs/english/externalfilters/tivtc_tdecimate.htm>

mode=0是选择每5帧里，最可能是重复帧（与前后相似度最高），mode=1是在每5帧里，选择最长连续的重复帧序列。它们的区别用于常规24d动漫几乎没有区别可言，这里推荐mode=0（默认）。

其实TDecimate还有其他很多参数（包括mode就有很多）可以微调，这里先略过。

vs当中，对应的滤镜是vfm.VDecimate()。<http://www.vapoursynth.com/doc/plugins/vivtc.html>

VDecimate没有那么多参数，基本相当于TDecimate(mode=0)。不过应付常规内容绰绰有余了。24d的内容读入，依旧需要先用std.SetFieldBased(src,0)标记这是progressive的源。

6、(多见于真人特典)每帧都有拉丝的30fps: 30i

如果你对于手上这个视频几乎每帧都有拉丝，说明这个视频就是Interlaced的。这种视频我们叫做30i. 对于Interlaced的视频，帧的概念不再适用（每帧都是烂的），而是用的场的概念。你看到的画面，奇数行构成上半场，偶数行构成下半场。

1080i的视频，每一场的分辨率是1920x540，每一场的画面就像是1080p压扁到1920x540一样。30i的视频，每秒钟30帧，就是60场。这样的视频需要做的操作是Deinterlace(反交错)，将每一个1920x540的场插值成1920x1080，再按照顺序先后显示:

QTGMC(preset=”slow”)

(QTGMC是最常用的反交错滤镜。通常通过preset来调节速度/质量的trade off)

vs里，还需要开启TFF = True/False，告诉滤镜这个视频的场序，一般的视频都是TFF=True，顶场优先。

完整的反交错后，30i的视频会被deint到60p。因为每秒钟有60场。部分软件为了保持前后帧率，会把一半的帧砍掉，变成30p。这种操作我们并不鼓励，但是全部60p编码时间极长，不需要保留“丝滑级动态效果”的场合下，砍到30p不失为非常合理的选择。

7、(多见于动漫特典)每5帧周期性的出现2帧烂帧:24t

如果你对于手上这个视频规律性的每5帧出现2帧左右（可能是1~3帧，但是基本是2帧）拉丝，说明这个视频是telecline的。这种视频我们叫做24t. 24t是24p的视频做了telecline操作，使得它可以在60Hz，支持Interlaced的电视显示器上播放。之所以会出现拉丝，是因为按照Progressive的显示方式，场之间出现了错位。

soft pulldown的视频，在开启repeat=True之后，读入的形式就是24t。

24t的视频需要做的操作叫做ivtc(inverse telecline,反交错)

tfm(order=-1, pp=6)

tdecimate(mode=0)

tfm这个滤镜的作用是场匹配。它可以把24t的视频场错位予以修复，变成24d，无拉丝的视频。然后再用tdecimate删除重复帧。

这里有两个参数，order指定场序，-1默认就是用avs里记载的场序（一般由原滤镜提供；pp表示如果出现没能匹配的帧，也就是30i部分，如何处理。pp=6是做一个反交错，这样可以免于显示拉丝。）

vs里对应的是vivtc.VFM()和vivtc.Vdecimate()

8、复杂组合——时间前后的组合

一个m2ts，并不一定前后都是一致的。经常是前面24t，后面30i，这种复杂的组合。甚至4种基本类型都有出现。对于组合型的片源，一般原则都是deint到60p，但是也有例外的：

如果分段比较少，也比较好确定段落，可以直接分段讨论处理

如果是24t + 24d, 直接当做24d做成24t。

如果是24t/24d+30p，并且其中一个占了绝对多数（一般认为70%以上），可以直接当做24t/24d或者30p做。24t的部分，可以通过场匹配变成24d。去除掉拉丝场景后，接下来只要决定要不要做删除帧。

需要当心的多数场景是没有拉丝的24d/30p，极少数是混搭极少数30i和24t场景，翻片源的时候，不一定看得到。这时候可以在读取片源后，接一个daa()做后处理。daa()的操作是上下分场，各自拉升到1920x1080，再average blend。对于上下场一样的这样没有问题（事实上它这样做可以去除缟缟），对于24t和30i内容，这样做会造成frame blending。但是极少数的blending总要好过未处理的拉丝:

src = core.lsmas.LWLibavSource(“24d+30p.m2ts”, repeat=True)

src = haf.daa(src)

如果是24t/24d + 30p，可以有自动化的方法做成可变帧率(<https://www.nmm-hd.org/doc/TIVTC> 中的7)

用于蓝光源，可以这么写：

1-pass:

LWLibavVideoSource("00000.m2ts", repeat=true)

tfm(output="matches.txt", pp=6)

tdecimate(mode=4, output="metrics.txt")

2-pass:

LWLibavVideoSource("00000.m2ts", repeat=true)

tfm(input="matches.txt", pp=6)

tdecimate(mode=5,hybrid=2,vfrDec=1,input="metrics.txt",tfmIn="matches.txt",mkvOut="mkv-timecodesfile.txt")

1-pass可以直接用avsmeter空跑不输出视频，只输出matches.txt和metrics.txt。这两个文件随后用于2pass作为输入。2pass输出的视频应该用于压制，并且输出mkv-timecodesfile.txt用于最终成品的时间码封装：



9、复杂组合——空间叠加的组合

一段画面，并不一定是某种基本类型。经常是背景24t画面，然后背景上面打上30i/30p的字幕，这种复杂的组合。对于复合型的片源，一般原则都是deint到60p，但是也有例外的：

在片源叠加比较简单，易于分段处理，ivtc\_txt60mc可以实现对于24t/24d+30i/30p混合源的处理(https://www.nmm-hd.org/newbbs/viewtopic.php?f=7&t=1367)

24d+30p这种完全没有拉丝的组合，可以直接当做30p做而不用deint。如果是24t+30p混合，比如背景24t动漫，盖上30p字幕或者插入作画，一般可以试着场匹配，场匹配后多半可以转为24d+30p。但是要当心30p部分对场匹配干扰，使得场匹配不能成功，还有剩余拉丝的的情况。如果30p部分造成了干扰，只能硬着头皮deint到60p了。

10、复杂组合——时间和空间双重叠加

就是上文所有可能都出现了。这种当做30i处理就好了。deint到60p，实际上就是模仿TV播放画面的效果，只要这个视频制作是没问题的，deint到60p就不会出现目视效果上任何的问题。

由此你们应该了解为啥我一直说碰到30fps的视频一定要小心。碰到30fps的视频，最基础的一步，一定要仔细分析它的片源类型。有哪些基础类型，是否有时间或者空间上的叠加。只有判断正确了，你才能针对性的做出最好的应对。