转 x264编码指南——码率控制

2013年10月14日 23:20:00 阅读数：22919

x264是一个 H.264/MPEG4 AVC 编码器，本指南将指导新手如何创建高质量的H.264视频。

对于普通用户通常有两种码率控制模式：crf（Constant Rate Factor)和Two pass ABR。码率控制是一种决定为每一个视频帧分配多少比特数的方法，它将决定文件的大小和质量的分配。

如果你在编译和安装libx264 方面需要帮助，请查看ffmpeg和x264编译指南：

http://ffmpeg.org/trac/ffmpeg/wiki/CompilationGuide

crf（Constant Rate Factor)：

该方法在输出文件的大小不太重要的时候，可以使整个文件达到特定的视频质量。该编码模式在单遍编码模式下提供了最大的压缩效率，每一帧可以按照要求的视频质量去获取它需要的比特数。不好的一面是，你不能获取一个特定大小的视频文件，或者说将输出位率控制在特定的大小上。

1 选择一个CRF值

量化比例的范围为0~51，其中0为无损模式，23为缺省值，51可能是最差的。该数字越小，图像质量越好。从主观上讲，18~28是一个合理的范围。18往往被认为从视觉上看是无损的，它的输出视频质量和输入视频一模一样或者说相差无几。但从技术的角度来讲，它依然是有损压缩。

若Crf值加6，输出码率大概减少一半；若Crf值减6，输出码率翻倍。通常是在保证可接受视频质量的前提下选择一个最大的Crf值，如果输出视频质量很好，那就尝试一个更大的值，如果看起来很糟，那就尝试一个小一点值。

注释：本文所提到的量化比例只适用于8-bitx264（10-bit x264的量化比例 为0~63），你可以使用x264 --help命令在Output bit depth选项查看输出位深，在各种版本中，8bit是最常见的。

2 选择一个预设

预设是一系列参数的集合，这个集合能够在编码速度和压缩率之间做出一个权衡。一个编码速度稍慢的预设会提供更高的压缩效率（压缩效率是以文件大小来衡量的)。这就是说，假如你想得到一个指定大小的文件或者采用恒定比特率编码模式，你可以采用一个较慢的预设来获得更好的质量。同样的，对于恒定质量编码模式，你可以通过选择一个较慢的预设轻松地节省比特率。

如果你很有耐心，通常的建议是使用最慢的预设。目前所有的预设按照编码速度降序排列为： ultrafast,superfast,veryfast,faster,fast,medium,slow,slower,veryslow,placebo

缺省预设为medium,请忽略 placebo因为它是毫无用处的（参看下面的问答）。你可以使用--preset来查看预设列表，也可以通过x264 --fullhelp来查看预设所采用的参数配置。

你可以基于输入内容的独特性通过使用--tune来改变参数设置。当前的 tune包括：film,animation,grain,stillimage,psnr,ssim,fastdecode,zerolantency。假如你的压制内容是动画，你可以使用animation，或者你想保留纹理，那就是用grain。如果你不确定使用哪个选项或者说你的输入与所有的tune皆不匹配，你可以忽略--tune 选项。你可以使用--tune来查看tune列表，也可以通过x264 --fullhelp来查看tune所采用的参数配置。

另外一个可选的参数是--profile，它可以将你的输出限制到一个特定的 H.264 profile,该选项可以被忽略除非你的播放设备只支持某种profile。当的所有profile 包括：baseline,main.high,high10,high422,high444 。注意使用--profile选项和无损编码是不兼容的。

如下所示，作为一种快捷方式，你可以通过不声明 preset和tune得内容来为ffmpeg罗列所有可能的内部preset和tune。

ffmpeg -i input -c:v libx264 -preset -tune dummy.mp4

3 使用你的预设

一旦你选择了一个预设，请把它应用到你的剩余的尚未编码的视频，这样可以确保它们有同样的视频质量。

CRF例子：

接下来将使用x264编码一个视频，使用一个比普通预设稍慢的预设，这样可以得到比默认设置稍好一点的视频质量。

ffmpeg -i input -c:v libx264 -preset slow -crf 22-c:a copy output.mkv

注意在这个例子中，输入文件的音频流被简单地拷贝到输出，并没有重编码。

两遍模式：

如果你的目标是一个确定大小的文件而且帧与帧之间的视频质量并不重要，这个方法很适用。这通过一个例子可以得到很好地解释。你的视频有10分钟（600秒）的时长同时要求输出为50MB，因为比特率=文件大小/时长，

50MB*8192（MB转kilobits)/600秒=683 kbps（全局比特率）

,683kbps-128kbps(音频比特率）=555kbps（视频比特率），

两边编码的例子：

ffmpeg -y -i input -c:v libx264 -preset medium -b:v 555k -pass 1 -an -f mp4 /dev/null &&

ffmpeg -i input -c:v libx264 -preset medium -b:v 555k -pass 2 -c:a libfdkaac -b:a 128k mp4 output.mp4

注意 windows 用户应该使用NUL来取代/dev/null

当使用CRF时，请选择使用你所能容忍的最慢预设。

同时建议你看一下《制作高质量的 MPEG4 DVD电影视频剪辑》,这是一篇 MPEG4 编码器编码指南,它会让你深刻的了解当你面临存储空间受限时,两边编码模式对于有效的使用每一个bit是多么的重要。

无损H.264

你可以使用-qp 0或者-crf 0 来编码一个无损输出,对于无损压缩我们提倡使用-qp 胜过-crf。因为8 bitx264和10 bitx264中的 crf 针对无损模式使用了不同的值。对此ultrafast和veryslow是两个非常有用的预设,因为飞快的编码速度和出色的压缩比通常是两个非常重要的因素。大部分的非 ffmpeg播放器不能播放无损模式,所以如果考虑到兼容性问题，你可能不能使用无损模式。

无损压缩的例子（快速编码）

ffmpeg -i input -c:v libx264 -preset ultrafast -qp 0 output.mkv

无损压缩的例子（高压缩比）

ffmpeg -i input -c:v libx264 -preset veryslow -qp 0 output.mkv

重写缺省预设

你可以使用-x264opts来重写预设或者使用 libx264的私有选项(可以通过ffmpeg -h来完整的查看 libx264选项)。我们并不建议你这么做除非你知道你在做什么。所有预设均是由x264的开发者创建的，想通过微调参数来提高输出质量通常是在浪费时间。

例子：

ffmpeg -i input -c:v libx264 -preset slow -crf 22 -x264opts keyint=123:min-keyint=20 -c:a copy output.mkv

附加信息 ：

ABR（Average Bit Rate)

ffmpeg -i input -c:v libx264 -b:v 1000k ....

它提供了某种“运行均值”的目标，终极目标是最终文件大小匹配这个“全局平均”数字（因此基本上来说，如果编码器遇到大量码率开销非常小的黑帧，它将以低于要求的比特率编码,但是在接下来几秒内的非黑帧它将以高质量方式编码方式使码率回归均值）使用两边编码模式是这个方法变得更加有效,你可以和“max bit rate ”配合使用来防止码率的波动。

CBR（Constant Bit Rate）

事实上根本就没有CBR这种模式，但是你可以通过补充ABR参数“模拟”一个恒定比特率设置，比如：

ffmpeg -i input -c:v libx264 -b:v 4000k -minrate 4000k -maxrate 4000k -bufsize 1835k out.m2v

在这个例子中，-bufsize是一个“码率控制缓冲区”,因此它会在每一个有用的1835k视频数据内强制一个你所要求的均值（此处为4000k），所以基本上我们会认为接收端/终端播放器会缓冲那么多的数据，因此在这个数据内部波动是没有问题的。

当然，如果只有黑帧或者空白帧，它所花费的的比特率将少于需求位率（但它会尽可能的提高质量水平，直到crf）。

最大比特率的CRF模式

你可以通过声明-crf和-maxrate设置来使用带有最大比特率crf模式，比如：

ffmpeg -i input -c:v libx264 -crf 20 -maxrate 400k -bufsize 1835k

这将会有效的将crf值锁定在20，但是如果输出码率超过400kbps，在这种情况下编码器会将质量降低到低于crf 20。

低延迟

x264提高了一个 -tune zerolatency 选项。

兼容性：

如果你想让你的视频最大化的和目标播放设备兼容(比如老版本的的ios或者所有的android 设备），那么你可以这做：

-profile:v baseline

这将会关闭很多高级特性，但是它会提供很好的兼容性。也许你可能不需要这些设置，因为一旦你用了这些设置，在同样的视频质量下与更高的编码档次相比会使比特率稍有增加。

关于profile列表和关于它们的描述，你可以运行x264 --fullhelp

要牢记apple quick time 对于x264编码的视频只支持 YUV 420颜色空间，而且不支持任何高于 mian profile编码档次。这样对于quick time 只留下了两个兼容选项baseline和 main。其他的编码档次qucik time均不支持，虽然它们均可以在其它的播放设备上回放。

使用-ss和-t选项可以编码一个段落而不是整个视频，这样可以快速的了解视频编码输出情况。

-ss 从起始值算起的偏移时间，这个值可以以秒为单位或者HH：MM：SS格式

-t 输出时延，这个值可以以秒为单位或者HH：MM：SS格式

问题与解答：

1 两遍编码模式能够比CRF模式提供更好的质量吗？

不能，但它可以更加精确地控制目标文件大小。

2 为什么 placebo 是一个浪费时间的玩意儿？

与 veryslow相比，它以极高的编码时间为代价换取了大概1%的视频质量提升，这是一种收益递减准则，veryslow 与 slower相比提升了3%；slower 与 slow相比提升了5%；slow 与 medium相比提升了5%~10%。

3 为什么我的无损输出看起来是无损的？

这是由于rgb->yuv的转换，如果你转换到yuv444,它依然是无损的。

4 显卡能够加速x264的编码吗？

不，x264没有使用（至少现在没有），有一些私有编码器使用了GPU加快了编码速度，但这并不意味着它们经过良好的优化。也有可能还不如x264，或许速度更慢。总的来说，ffmpeg到目前为止还不支持GPU。

翻译注释：x264在2013版中已经开始支持基于opencl的显卡加速，用于帧类型的判定。

5 为Quick time 播放器压制视频

你需要使用-pix_fmt yuv420p来是你的输出支持QT 播放器。这是因为对于H.264视频剪辑苹果的Quick time只支持 YUV420颜色空间。否则ffmpeg会根据你的视频源输出与Quick time 不兼容的视频格式或者不是基于ffmpeg的视频。

原文地址： http://blog.csdn.net/vblittleboy/article/details/8982857

文章标签： x264 码率控制 crf abr

个人分类： 视频编码