转 深入了解 VP8

2013年10月15日 23:18:03 阅读数：12913

部分翻译： http://x264dev.multimedia.cx/?p=377

译者：delectate

- **问题一：vp8到底怎么样？**

难道他真的比x264拥有更高的压缩比率，是个优秀的编码器吗？他真的比h264优秀吗？似乎On2自己都羞于承认…拿vp7举例，On2宣称vp7比h264快15%，但事实是编码视频速度既不快，视频质量也不高。On2曾经把vp3开源，似乎想借助社区的力量debug，最终theora上当……他们花了6年时间，结果做出来的还是个鸡肋……

- **问题二：vp8真的没有专利问题吗？**

这个东西，还是google说了算……微软几年前发布VC-1，然后说没有专利问题……但是几个月后，就有众多公司认领了专利并成立了专利池（指几家公司把各自的专利放在一起，组成一个"池"。其他人如果要使用VC-1，就须向"池"的管理公司申请许可，一旦获得了许可，就可以使用"池"中的所有专利。）

- **问题三：vp8的代码情况如何**

首先你要知道：一个很好的编码器，可能是基于一个很烂的标准（divx？xvid？lol）；而一个很好的标准，也可能会出现n多很烂的编码程序（h264 vs coreavc？不知道耶）。

vp8的文档真的很烂，很多细节要不就是没有文字说明，要不就是拿一大堆c代码来填补空白。真的好痛苦……

我一直认为H.264的规格写得太啰嗦，但和vp8相比，至少他是精确而详细的，至少不会杀伤我这么多脑细胞。如果只靠vp8的这份文档，我想我死也写不出来vp8的解码器……（莫非他们就是一行一行读h264的文档然后写的ffh264解码器？牛！）

不吐槽了，把视线收回来，看vp8。一般处理视频的步骤都是：

编码：预测 -> 变换+量化处理 -> 熵编码 -> 环路过滤器

解码：熵编码 -> 预测 -> 反量化处理+变幻 -> 环路过滤器

- **p帧**

就是通过当前帧已有的部分预测其他区块的内容。可以是在当前帧进行，也可以通过运动补偿的方式在帧间进行。

- **帧内预测**

帧内预测是用来编码帧间不同，在空间域上进行的预测编码算法，可以除去相邻块之间的空间冗余度，取得更为有效的压缩。

vp8的帧内预测基本上都是照抄h264的。"subblock"几乎和h264一样（他们竟然用了同样的名字），还有h264的4×4模式……whole block预测也基本上一样使用16×16模式。色度预测模式也几乎没有区别，所以不可能拥有比h264更出色的效果了。但是值得一提的是，他们用TM_PRED替代了planar预测模式。在预测方式上看起来有些不同，但是实际上h264都提供了相似的实现方法。

所以我对vp8有点失望。虽然说h264的预测功能的确不错，但是这也是7年的老物了……需要改进，所以我真的希望类似real这样的公司赶紧灭亡吧，一个存在了十几年而从来没有任何改进的格式（rm，rmvb）……我希望on2能做一些更有创造性的工作，而不是照搬h264的东西，因为h264的专利问题就是一个定时炸弹。h264的帧内预测可是有专利的，真不知道on2怎么想的，我可不认为on2改个名字就能蒙混过关。

帧内预测对决：大部份照抄h264，甚至有的连名字都没改……但是效果貌似还不如h264。

- **帧间预测：**

帧内预测主要用在去除空间冗余上而帧间预测则主要用在去除时间冗余上。运动估计就是在参考帧中寻找与当前编码宏块最匹配的宏块，而运动补偿就是计算二者的差值。帧间预测主要由两个部分组成：参考帧和运动矢量。vp8支持3中参考帧：p帧，g帧（golden fream）和alt ref帧。运动矢量上，vp8支持比h264更多的可变大小区块，次像素精度上，他支持四分之一像素和6-tap插值过滤。简而言之就是：

- vp8参考帧：3
- h264：16
- vp8支持区块类型：6×16, 16×8, 8×16, 8×8, 4×4
- h264：16×16, 16×8, 8×16, 但是还有更灵活的子区块：例如 8×8 可以被分为 8×8, 8×4, 4×8, 或者 4×4
- VP8 chroma MV derivation: 4×4 色度均值处理 (有点类似于 MPEG-4 ASP)
- h264：直接使用，没有什么处理（没有均值处理，所以视觉效果比较好）
- H.264 拥有b帧和加权预测，但是vp8却没有

h264拥有更为灵活的架构，所以8×8并不常用，所以vp8弃用也没有太多影响。但是色度处理上，h264因为没有均值，所以拥有比vp8更好地效果，但是理论上它需要更多的时间来编码。但是实际中差别并不是很明显。

vp8的插值过滤器似乎优秀一些，但是他是以牺牲性能为代价的。竟然还用高达6的色度，真搞清楚他们在干什么……这只是无谓的浪费。

vp8没有使用b帧是个致命的缺陷。使用b帧可以提高10-20%压缩率并加快编码速度，但是on2在他们所有的视频编码格式中都没有应用b帧，但是即使如此也有可能引起专利问题。

帧间编码对决：部分与h264相似，但是架构上不如h264，虽然使用了更为优秀的过滤器，但是没有使用b帧，所以会严重导致压缩率降低。

- **变换与量化编码**

总体来说，VP8肯定比H.264弱。一个8 × 8变换缺乏细节，特别是在高的分辨率。很多转换也不是必要的，却在进行。所以比较慢。由于专利，变换也有可能产生纠纷。一个良好的新思路是采用 分层直流转换间块。

转换：类似H.264标准。慢一点，再稍微更准确的4 × 4变换。改进直流变换亮度（但色度不是）。没有8 × 8变换。总体而言，更糟。
量化方面，核心过程基本上和所有的MPEG 视频格式一样，所以VP8也不例外。一个视频格式，如果想体现自己与众不同，那么他们主要的方式是通过改变量化尺度因子。方法有两种，主要是：基础帧的偏移量，宏块级的偏移，对于视频来说整体适合或者部分适合。 VP8主要使用前者，但是远远低于H.264灵活的自定义量化矩阵，它允许调整亮度量化直流，交流亮度，色度直流，等等。后者（宏块级量化选择）可以在理论实现“图像分割”功能，虽然很hack，但是效果嘛……
vp8的致命错误是已经不使用宏块级量化作为其核心功能。该算法利用宏块级量化的优势被称为“自适应量化”，是绝对至关重要的。x264使用 执行方差的自适应量化 （算法对比：使用前 ， 使用后 ）后效果明显，至今仍然是x264视觉质量收益提供支持。
对量化对决：在成为杀手级视频格式前，缺少综合自适应量化将是绝对错误的。总体而言，非常糟糕。

- 熵编码

根据信息论的原理，可以找到最佳数据压缩编码的方法，数据压缩的理论极限是信息熵。如果要求编码过程中不丢失信息量，即要求保存信息熵，这种信息保持编码 叫熵编码，是根据消息出现概率的分布特性而进行的，是无损数据压缩编码。
在视频编码中，熵编码把一系列用来表示视频序列的元素符号转变为 一个用来传输或是存储的压缩码流.输入的符号可能包括量化的变换系数(像上面所说的运行级或零树),运动向量(对于每个运动补偿块的向量值x和y),标记 (在序列中用来表示重同步位的点),头(宏块头,图象头,序列的头等)以及附加信息(对于正确解码来说不重要的信息).
VP8使用的熵编码器有点类似于H.264，但有几个关键处又有所不同。首先，它忽略了范围/概率乘法表。第二，它是完全非自适应的：与H.264相比，它能够解码后的每一帧，概率值保持不变。
这种做法并不奇怪; VP5和VP6（也许是VP7）也用的非自适应算术编码器。更重要的是，我之所以对这个问题感兴趣，是因为它使自适应会增加单一的表查找来的算术解码功能 ——对性能的影响很小。
vp8这个方案的缺点是，像VP3/Theora（虽然没有那么严重），它偏向于以重新使用以前的运动矢量。这是相当危险的，因为正如开发员们最近发现，一些情况下，即选取一个运动矢量编码器是不是“真实”的运动矢量，以略去潜在的负面的视觉影响。在效率方面，我不知道是VP8比H.264的预测是否能做到更好。

还有一点要注意的是VP8文件的结构。这个有点像VP3/Theora，涉及组件的每个码流的语法元素。这个不幸的是，它是硬件加速的噩梦，需要极大提高存储效能以及带宽。
熵编码对决：它在某些方面出现好转，在某些方面恶化，实在怪异。我的直觉是，它对H.264非自适应算术编码有一些轻微的优势，但是硬件加速方面，真是杯具。

- **环路滤波器**

通常是在编码/解码帧后使用，通常是为了消除基于DCT的视频格式的方块现象。
VP8的环路滤波器与H.264依然相似，但有一些区别。首先，它有两种模式（即可以由编码器选择）：快速模式和普通模式。快速模式比H.264 的简单，而正常模式较为复杂。第二，宏块之间的滤波，VP8的过滤器比h264的宏块滤波器有更广泛的作用范围，h264的则仅限于边缘。
环路滤波器对比：绝对比H.264的差，因缺乏自动适应性。虽然选择快速模式可以加快速度，但是成像结果惨不忍睹，十分模糊。

- **附录A：**

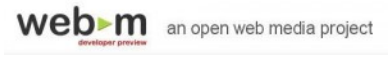
总体而言，很明显，VP8明显比H.264差。上面提到的主要弱点是缺乏适当的自适应量化，没有B帧，缺少8 × 8变换，以及非自适应性环路滤波器。其实我真的希望VP8能超越H.264，VC – 1或者H.264 Baseline Profile。当然，vp8现在性能明显优于theora和Dirac。
解码的速度我不太清楚条件，目前大约是FFmpeg的H.264解码器的16%（比自称最先进的CoreAVC慢25%-35%） 。当然，全面优化后能达到多少还难以预料，但是目前的确需要大幅优化。
最后，还是专利问题。 VP8与H.264太相似了：一个精辟的，有些不太准确的描述说VP8将是基于H.264 Baseline Profile的最好的编码器。虽然我不精通法律，但是我不相信他们能够做到这点。即使VC – 1的不同于VP8和H.264，但是VC – 1也无法从软件专利这个魔掌中逃脱。
如果google运气够好，专利之战能够胜利，也许能给theora带来一丝曙光。

> *对比：*
> *VP8 (On2 VP8 rc8) (source)*
> *H.264 (Recent x264) (source)*
> *H.264 Baseline Profile (Recent x264) (source)*
> *Theora (Recent ptalabvorm nightly) (source)*
> *Dirac (Schroedinger 1.0.9) (source)*
> *VC-1 (Microsoft VC-1 SDK) (source)*
> *MPEG-4 ASP (Xvid 1.2.2) (source)*

附录B：
谷歌之所以选择Matroska作为封装格式，这是不足为奇的：的Matroska是目前使用最广泛使用的视频封装格式之一，也许MP4（又名ISOmedia）可能是一个更好的设计格式，但不是很灵活，而且是一个封闭格式。

另一个优点是Matroska可用于视频流：虽然不常用，它的确可以。请注意，我不是说渐进式下载（a'la YouTube）的，而是实际的流，其中编码器是实时工作的。这样做的唯一方法是用MP4通过发送视频的“片段”，非常hacky的方法。

真搞不懂为什么google非要给 Matroska 起 WebM 这么个愚蠢的名字



音频选择Vorbis格式，是明智之举，没有专利问题，而且性能优秀。而且libvorbis是最好的开源通用音频编码器。虽然AAC是在低的比特率表现较好，但是没有好的开源的AAC编码器：FAAC的FFmpeg的AAC编码器是更糟。此外，FAAC分部是不是免费软件，它包含了非免费编码器的代码。因为专利的问题，估计google也别无可选……

- 参考资料:

http://x264dev.multimedia.cx/?p=377

http://www.ruanyifeng.com/blog/2010/05/the_first_view_of_vp8.html

http://baike.baidu.com/view/587027.htm

http://hi.baidu.com/hnu_lina/blog/item/f13a59b1fd7eb9520823023e.html

http://zh.wikipedia.org/zh/%E7%86%B5%E7%BC%96%E7%A0%81

原文地址： https://www.deleak.com/blog/2010/05/22/the-first-in-depth-technical-analysis-of-vp8/

文章标签： vp8 h.264 编码 滤波 对比

个人分类： 视频编码