

对 H1264 中的多参考帧预测技术的评测

袁 昱, 孙立峰, 韩 丹, 钟玉琢

(清华大学计算机科学与技术系, 北京 100084)

摘 要: H1264 标准支持多参考帧预测技术. 为了考察这种基于多个前向参考帧的帧间运动补偿技术与编码效率的关系, 我们分别调节前向参考帧、跳过帧和 B 帧数这几个关键参数, 在它们取值的不同组合下进行了各种测试视频序列的编码实验. 实验结果表明, 多参考帧预测在绝大多数情况下不能改善编码效率, 反而引入了繁重的计算开销. 我们认为这是由于能体现多参考帧预测优势的视频序列的自相似性在实际应用中较为罕见, 因此建议尽可能不使用多参考帧预测技术.

关键词: 视频编码; H1264; 多参考帧; 预测

中图分类号: TP3 **文献标识码:** A **文章编号:** 0372-2112 (2004) 092-1572-04

Investigation of Multi-Reference-Frame Prediction in H1264

YUAN Yu, SUN Liefeng, HAN Dan, ZHONG Yuzhuo

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: H.264 supports a so-called multi-reference-frame prediction technique. We adjust its three relevant parameters (the number of previous reference frames, the number of skipped frames, and the number of B frames), and use their different combinations to encode various test video sequences respectively. Experimental results show that multi-reference-frame prediction does not improve coding performance but involve heavy computing cost in most cases. We propose that the advantage of this technique depends on a sort of video content self-similarity, which seldom occurs in practice. We then suggest not to use the multi-reference-frame prediction technique in most applications.

Key words: video coding; H1264; multi-reference-frame; prediction

1 引言

H1264^[1], 也被称为 MPEG2.4 AVC (Advanced Video Coding) 或 JVT (Joint Video Team), 是目前最先进的视频编码标准, 由 ISO/IEC 和 ITU-T 联合制订. 它集成了许多最新的视频编码技术. 毫无疑问, 它超出此前所有视频编码标准的杰出性能正是这些新技术的共同贡献. 然而, 这些新技术的性价比彼此之间却有着很大的差别. 其中的一些技术仅仅能轻微地提高编码效率, 却引入了繁重的计算开销. 对这一事实的勘查将有助于我们在实际应用中决定编码效率和计算复杂度之间的平衡, 尤其是在低端的应用中.

参考帧是帧间预测编码也就是运动补偿的基础, 根据其有待预测帧之间的位置关系, 可分为前向参考帧和后向参考帧. 例如, 在 MPEG2.4^[2] 中, P 帧是靠前向参考帧预测的, 而 B 帧可以靠前向参考帧和后向参考帧共同预测. 在以往的视频编码标准中, 一次预测里只允许一个前向参考帧^[2-4]. 作为一项改进, H1264 引入了多参考帧预测技术^[1], 允许前向参考帧

的数目从 1 到 5. 直觉地看来, 增加前向参考帧的数目将有助于提高在运动估计中找到最匹配的参考宏块的概率, 然而它也将带来显著的额外计算开销. 为了查明多参考帧预测究竟是不是一种具有良好性价比的视频编码技术, 我们把 H1264 JM 编解码器^[5] 作为实验平台, 测量了多参考帧预测技术的实际性能.

2 测试方法

在多参考帧预测中一共有三个关键的参数: 前向参考帧的数目, 跳过帧的数目, 和 B 帧的数目. 这三个参数在 H1264 JM 编解码器^[5] 中被表示为 NumberReferenceFrames, FrameSkip, 和 NumberBFrames. FrameSkip 表示在两个连续的参考帧 (例如, 两个连续的 P 帧) 之间的间隔 (跳过的帧数). NumberBFrames 表示在两个连续的参考帧之间应该插入并编码多少个 B 帧. 如果我们令这两个参数相等, 则我们可以把它们看作是一个联合的参数. 这个联合的参数的意思就是在两个连续的参考帧之间有多少个 B 帧. 以下, 我们用 NumberBFrames 来代表这

个联合参数.

我们使用了 10 个 CIF 文件(分辨率为 352@288)和 10 个 QCIF 文件(分辨率为 176@144)作为我们的测试数据集. 这些文件都是标准的视频测试序列^[3,4], 例如 Foreman 和 Coastguard. 在保持其他设置不变的同时, 我们调整 NumberReferenceFrames 和 NumberBFrames 来观察这些参数是如何影响码率和 PSNR 的. NumberReferenceFrames 的取值从 1 到 5, NumberBFrames 的取值从 1 到 10, 所以我们对每个测试文件将得到 50 个结果. 每一个结果都含有测试文件的每一帧被编码后的 PSNR. Y, PSNR. U, PSNR. V 和 Bits 信息.

3 测试结果

根据这些结果, 我们生成了每帧 Bits 的曲线和每帧 PSNR 的曲线. 分别固定 NumberReferenceFrames 和 NumberBFrames, 我们就把这些曲线组织成了不同的图. 在对这些图进行仔细的观察之后, 我们发现 PSNR. U 和 PSNR. V 的图与 PSNR. Y 的图非常相似, 不能为我们提供更多的信息, 所以我们选择 PSNR. Y 的图来展示 PSNR 所受的影响.

篇幅所限, 我们只列出一部分的图. 这部分图是由 Foreman CIF 的实验结果生成的, 按照其反映的是码率曲线还是 PSNR 曲线分别归类为图 1 和图 2. 实际上, 多参考帧预测技术在其他测试文件上的表现也非常类似.

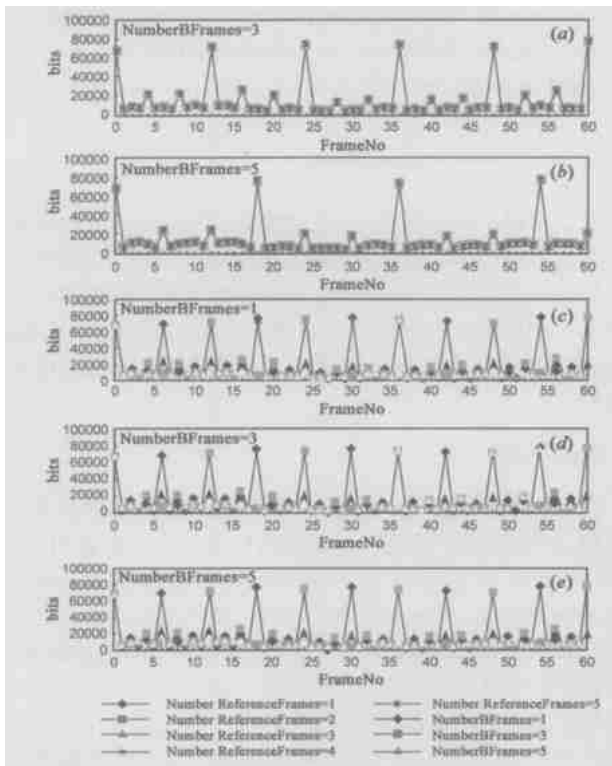


图 1

4 分析与解释

4.1 对测试结果的分析

图 1 中的前两幅图是我们分别把 NumberBFrames 固定为

3 和 5 时, 码率曲线随 NumberReferenceFrames 的变化情况. 我们可以看到, 每幅图中的样点都呈现三个层次: 最高的尖峰, 次高的尖峰和低谷. 由于 I 帧的 Bits 比 P 帧多, P 帧的 Bits 比 B 帧多, 因此最高的尖峰实际上就是 I 帧, 次高的尖峰是 P 帧, 而低谷是 B 帧.

毫无疑问, I 帧的编码与 NumberReferenceFrames 无关, 所以 I 帧的 Bits 应该不随 NumberReferenceFrames 变化, 图中的曲线也反映了这一点, 每幅图中的 5 条曲线里表示 I 帧的尖峰都是重合的. P 帧和 B 帧的编码与 NumberReferenceFrames 有关, 但是我们可以看出, 每幅图中的 5 条曲线里表示 P 帧的尖峰和表示 B 帧的低谷仍然基本是重合的, NumberReferenceFrames 的变化并未导致值得注意的区别. 在 NumberBFrames=3 的那幅图中, 仅有 3 处的 P 帧显得在 NumberReferenceFrames=1 时 Bits 比另 4 条曲线略高, 但是 NumberReferenceFrames 从 2 开始进一步增加并未导致 Bits 进一步降低. 在 NumberBFrames=5 的那幅图中, 仅有 1 处的 P 帧显得在 NumberReferenceFrames=1 时 Bits 比另 4 条曲线略高, 但是 NumberReferenceFrames 从 2 开始进一步增加并未导致 Bits 进一步降低. 而 B 帧的 Bits 随 NumberReferenceFrames 的变化就更小到可以忽略.

图 1 中的后三幅图是我们分别把 NumberReferenceFrames 固定为 1, 3, 5 时, Bits 曲线随 NumberBFrames 的变化情况. 每条曲线中最高尖峰是 I 帧, 次高的尖峰是 P 帧, 低谷是 B 帧. 因为每幅图中的 3 条曲线其 I, P, B 帧的分布并不一致, 所以曲线呈现相当的差异是完全符合我们预期的.

这三幅图都反映出, NumberBFrames 越大, 则 B 帧的 Bits 越高, 这是因为 NumberBFrames 越大, 则对 B 帧进行预测的误差就越大. 但是在 X 轴上相同位置上 I 帧的 Bits 远大于 P 帧和 B 帧, P 帧的 Bits 也明显大于 B 帧. 因此 NumberBFrames 越大, 整个序列总的 Bits (可以视作曲线的积分) 就越小. 这三幅图彼此之间并无可以察觉的区别, 说明 NumberReferenceFrames 不会影响 Bits 曲线随 NumberBFrames 的变化规律.

图 2 中的前两幅图是我们分别把 NumberBFrames 固定为 3 和 5 时, PSNR 曲线随 NumberReferenceFrames 的变化情况. 我们可以看到, 每幅图中的样点都呈现两个层次: 最高的尖峰, 起伏不平的低谷. 最高的尖峰是 I 帧, 而起伏不平的低谷是 P 帧和 B 帧. 理论上, I 帧、P 帧和 B 帧并不应该在 PSNR 上注有区别. 我们的曲线也表明, P 帧与 B 帧的 PSNR 并不像 Bits 那样呈现出 P 帧明显高于 B 帧的规律, 但是 I 帧的 PSNR 仍然明显高于 P 帧和 B 帧.

毫无疑问, I 帧的编码与 NumberReferenceFrames 无关, 所以 I 帧的 PSNR 应该不随 NumberReferenceFrames 变化, 图中的曲线也反映了这一点, 每幅图中的 5 条曲线里表示 I 帧的尖峰都是重合的. P 帧和 B 帧的编码与 NumberReferenceFrames 有关. 从每幅图中我们都可以看出, P 帧和 B 帧的 PSNR 曲线呈现出了比 Bits 图要明显得多的不重合现象. 更仔细地观察就可以看出, NumberReferenceFrames=1 时的 PSNR 曲线是最低的, 与 NumberReferenceFrames=2 的差别最大可以达到大约 0.2dB; 但是 NumberReferenceFrames=3 的曲线与 NumberRefer

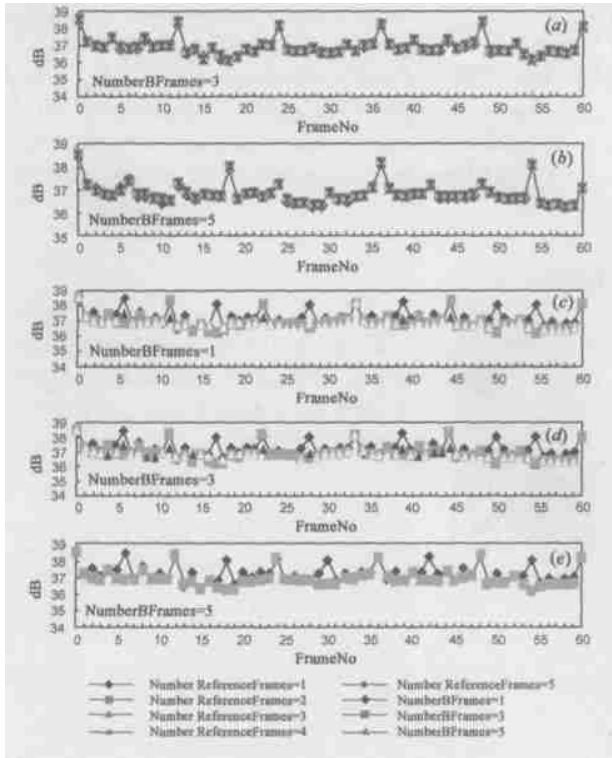


图 2

enceFrames= 2 的曲线就只有极小的差别, 绝大部分地方几近重合, 差别最大的地方也不超过 0.1 dB; 而 NumberReferenceFrames 从 3 开始进一步增加则根本看不到 PSNR 曲线有可以察觉的变化。

图 2 中的后三幅图是我们分别把 NumberReferenceFrames 固定为 1, 3, 5 时, PSNR 曲线随 NumberBFrames 的变化情况。每条曲线中最高的高峰是 I 帧, 起伏不平的低谷是 P 帧和 B 帧。因为每幅图中的 3 条曲线其 I、P、B 帧的分布并不一致, 所以曲线呈现相当的差异是完全符合我们预期的。

这三幅图都反映出, NumberBFrames 越大, 则 B 帧的 PSNR 越低, 这是因为 NumberBFrames 越大, 则对 B 帧进行预测的误差就越大。但是在 X 轴上相同位置上 I 帧的 PSNR 明显大于 P 帧和 B 帧, P 帧的 PSNR 也略大于 B 帧。因此 NumberBFrames 越大, 整个序列的平均 PSNR 就越小。这三幅图彼此之间并无可以察觉的区别, 说明 NumberReferenceFrames 不会影响 PSNR 曲线随 NumberBFrames 的变化规律。

综上, H1264 所引入的多参考帧预测技术, 并不能对编码效率产生值得注意的提高。NumberReferenceFrames= 2 与 NumberReferenceFrames= 1 相比, 编码产生的总 Bits 数只会降低几乎可以忽略的一点点, 而少数 P 帧或 B 帧的 PSNR 会略为提高一点, 但最大不超过 0.1 dB。在 2 以上进一步提高 NumberReferenceFrames 则不会带来任何值得注意的变化。换言之, NumberReferenceFrames 从 1 提高到 2, 只会极细微地提高编码效率, 而在 2 以上的进一步提高则没有任何意义。

4.1.2 进一步的解释

直觉也许会告诉我们, 在预测时采用多参考帧, 必然有利

于作出更准确的预测, 从而预测误差更小, 进而提高编码效率。但是上面的测试结果却并非如此理想。这是为什么呢? 下面是我们对此的一种解释:

我们可以拿 B 帧的原理来类比。直觉上, B 帧是双向预测, 必然有利于作出更准确的预测, 从而提高编码效率。一定是这样吗? 让我们来看两个例子。

为了便于说明, 我们用数值来表示待编码的序列。假定原始序列 a 为:

1 2 3 4 5 6 7 8 9

如果编码方案为 IPPPIPPPI, 则我们得到

1 + 1 + 1 + 1 5 + 1 + 1 + 1 9

如果编码方案为 IBPBI BPBI, 则我们得到

1 + 1 + 2 + 1 5 + 1 + 2 + 1 9

可以看出, 对原始序列 a, 采用 B 帧不但没有带来好处, 反而降低了编码效率。

再看另一个原始序列 b:

5 1 9 1 5 1 9 1 5

如果编码方案为 IPPPIPPPI, 则我们得到

5 - 4 + 8 - 8 5 - 4 + 8 - 8 5

如果编码方案为 IBPBI BPBI, 则我们得到

5 - 4 + 4 - 4 5 - 4 + 4 - 4 5

可以看出, 对原始序列 b, 采用 B 帧明显提高了编码效率。

由此可见, 采用 B 帧并不总是会提高编码效率, 有时反而会降低编码效率。采用 B 帧是否会提高编码效率, 跟原始序列的内容有关。只不过, 实际应用中, 类似于原始序列 a 这样的特殊情形并不多, 就统计规律而言, 大多数情况下, 采用 B 帧都会提高编码效率。

采用多参考帧的情形与上面很类似。多参考帧是否会提高编码效率, 同样是跟原始序列的内容有关的。我们可以认为, 实际应用中, 大多数原始序列都不能体现出多参考帧的优越性, 仅仅是在少数情况下 NumberReferenceFrames= 2 会比 NumberReferenceFrames= 1 略好, 而 NumberReferenceFrames= 3 以上就更难找到能体现其优势的原始序列了。

我们还可以从另一个角度来理解。假定我们有 3 个帧, 记为 f_1, f_2, f_3 。如果 f_3 是 P 帧, 把 f_1 和 f_2 当作参考帧来预测 f_3 (NumberReferenceFrames= 2), 实际上和把 f_2 当作 B 帧, 由 f_1 和 f_3 来预测 f_2 (NumberReferenceFrames= 1) 并无多大区别, 都是从两个参考帧来预测一个帧。NumberReferenceFrames= 2 时, P 帧是由两个参考帧来预测的, 我们可以将其看作是退化成了 B 帧, 因此也就相当于 NumberReferenceFrames= 1 时增大了 NumberBFrames。所以编码效率会略有提高。但是 NumberReferenceFrames= 3 以上就基本无法再提高编码效率。

5 结论

H1264 中的多参考帧预测是一种性价比非常低的视频编码技术, 在绝大多数情况下, 它都不能使编码效率有所改善, 但却会引入繁重的额外计算开销。我们建议尽可能不使用该技术。

然而,确实有可能存在一些视频片段,在这些视频片段上使用多参考帧预测将有助于提高视频的客观质量(由 PSNR 所度量)。因此,在某些不需要实时编码的应用中,对视频序列允许进行两趟乃至更多趟扫描,于是我们就可以通过对视频序列的预先分析,基于内容地决定编码的参数设置^[6,7],从而在能体现多参考帧预测的优势的视频片段上使用多参考帧预测。

参考文献:

- [1] Recommendation ITU-T H1264. Advanced video coding for generic audiovisual services [S] .
- [2] International Standard ISO/IEC 13818. Information technology—Generic coding of moving pictures and associated audio[S].
- [3] Wang Y, Ostermann J, Zhang Y Q. Video Processing and Communications[M]. USA: Prentice Hall, 2002.
- [4] Bhaskaran V, Konstantinides K. Image and Video Compression Standards[M]. London: Kluwer Academic Publishers, 1997.
- [5] ITU-T. H1264 JM Reference Software[CP] . ITU-T, 2003.

- [6] Lan A Y. Scene Context Dependent Reference Frame Placement for MPEG Video Coding[D]. Seattle: University of Washington, 1996.
- [7] Yuan Y, Feng D, Zhong Y Z. Three fast methods for adaptive keyframe setting and dynamic frame rate adjusting in video coding[A]. Proc of International Conference on Signal Processing (ICSP 2003) [C] . Canakkale: International Computational Intelligence Society, 2003. 342 - 347.

作者简介:



袁昱男, 1976 年生于重庆, 1999 年毕业于清华大学计算机科学与技术系, 现为清华大学计算机科学与技术系博士研究生, 主要研究方向为视频编码与视频分析。