

MPEG-2 到 H.264 转码中的运动矢量优化算法

刘 昱¹, 王 立², 王 磊¹

(1. 天津大学电子信息工程学院, 天津 300072; 2. 深圳中兴通讯股份有限公司, 深圳 518057)

摘 要: 在视频转换编码技术中, 通常利用输入码流中的运动矢量等信息来降低转码处理的计算量。该文介绍一种 MPEG-2 到 H.264 的转码器, 针对 MPEG-2 和 H.264 标准中运动估计算法的不同特点, 提出用于转码处理的运动矢量优化算法, 并对不同算法进行分析比较。结果表明该优化算法可以在保证转码质量的前提下, 降低转码处理的计算量。

关键词: MPEG-2 标准; H.264 标准; 转码; 运动矢量

Motion Vector Refine Algorithms of MPEG-2 to H.264 Transcoding

LIU Yu¹, WANG Li², WANG Lei¹

(1. School of Electronic and Information Engineering, Tianjin University, Tianjin 300072; 2. ZTE Corporation, Shenzhen 518057)

【Abstract】 Information of coded video streams is always used by video transcoders to improve the efficiency. This paper suggests a MPEG-2 to H.264 video transcoder. The motion estimation difference between MPEG-2 and H.264 is analyzed and several Motion Vector(MV) refine algorithms are proposed to improve the transcoding quality. Using the suggested MV refine algorithms, the transcoder can generate high quality H.264 videos with low complexity.

【Key words】 MPEG-2; H.264; transcoding; Motion Vector(MV)

1 概述

目前, 数字多媒体压缩编码标准 MPEG-2 已获得了广泛应用, 成为多媒体资源的一种主要编码格式。H.264/AVC 作为新一代的视频压缩编码标准, 将逐步进入实际应用。为了共享节目资源, 将 MPEG-2 视频转换为 H.264/AVC 格式的需求日益强烈。运动估计处理在视频编码过程中计算量巨大, 其中用于帧间宏块编码的运动估计处理占用了整个编码过程约 70% 的运算量。因此在转换编码器中利用解码端获得的 MPEG-2 中现有运动矢量(MV)可以提高 H.264 编码部分的运动估计速度, 进而提高整个转换编码器的效率。

转换编码中通常以输入 MV 为中心进行优化运动估计, 优化处理可以参考现有的 H.264 快速运动估计算法, 文献[1]提出一种两步法, 通过限制搜索面积进行快速运动估计。文献[2]建议了一种零子块检测的方法来降低运动搜索的计算量, 提高编码速度, 在实时低码率编码器中可以得到很好的应用。文献[3]利用了 EPZS 方法来实现 H.264 中快速运动估计。文献[4]将 EPZS 方法应用到了 MPEG-2 到 H.264 转换编码中, 在编码效率上有一定的提高。在这些算法的基础上, 本文提出并比较了 4 种用于 MPEG-2 到 H.264 转码中的快速运动矢量优化算法, 可以在保证视频质量的情况下, 大幅度地节省计算量。

2 运动矢量重用转换编码器

为了获得较好的运动估计性能, 在 MPEG-2 到 H.264 转码处理中需要考虑 2 种标准在运动估计方面的差异, 主要的差异如下:

(1) MPEG-2 中 MV 是 1/2 像素精度的, 而 H.264 标准中规定, MV 是 1/4 像素精度的, 这样能够得到更精确的预测块, 从而降低残差数据的能量, 提高压缩效率。

(2) 半像素内插公式不同。在 MPEG-2 中, 半像素的内插是用相邻 2 个像素的算数平均值得到的。而在 H.264 中, 亮度的半像素是用一个六阶有限冲击响应滤波器对相邻整数位置的像素值进行内插得到的, 1/4 的像素值用线性插补得到。文献[5]中给出的算法可以对转换编码中由于 2 个标准内插公式不同引起的误差进行校正。

(3) 预测帧的数量不同。在 MPEG-2 中, 只能使用前面一帧和后面一帧的数据进行预测, 最大范围是 2 帧。而在 H.264 中, 可以使用多帧预测, 最大范围是 16 帧。

(4) 预测块大小不同。在 MPEG-2 中是按照 16×16 的宏块或 16×8 的块进行预测的, 而在 H.264 中, 可以将宏块划分成 16×16, 16×8, 8×16 和 8×8 4 种子块, 而 8×8 的子块又可以进一步划分成 8×8, 8×4, 4×8 和 4×4 4 种更小的块来进行预测, 较小的块可能使运动估计更精确, 产生较小的运动残差, 降低码率, 这就大大提高了运动估计的灵活性, 从而提高压缩效率。

针对上述差异, 在 MPEG-2 到 H.264 转换编码中, 也应当充分利用 H.264 的优点来获得更好的压缩效率。由于 H.264 中的 B 条带属于主档次而不是基本档次, 最基本的解码器并不支持, 因此从实用性考虑, 本文主要研究的是 P 条带中的运动估计算法。此外, 根据实验比较及参考文献[6], 多帧预测技术对于提高 H.264 编码效率的贡献不明显, 而计算量增加很多, 因此, 本文的算法没有考虑多参考帧的情况。

基金项目: 国家自然科学基金资助项目(60602033); 天津市自然科学基金资助重点项目(07JCZDJC6800)

作者简介: 刘 昱(1976—), 男, 讲师, 主研方向: 数字多媒体处理; 王 立, 博士; 王 磊, 硕士研究生

收稿日期: 2007-12-20 **E-mail:** liuyu@tju.edu.cn

3 MPEG-2 到 H.264 转换编码的运动矢量优化算法

在转换编码中, 变字长解码之后可以得到 MPEG-2 中每个宏块的 MV, 将该 MV 重新应用到 H.264 编码器部分就能够有效提高运动估计的速度, 得到运动矢量重用级联转换编码结构, 功能框架如图 1 所示。图中运动估计处理是影响转码性能的主要因素, MPEG-2 和 H.264 编码处理中, 均采用如下流程完成运动估计: 先是进行整像素全局搜索, 得到整像素的运动矢量, 之后经过内插得到半像素和 1/4 像素内插图像, 再进行运动矢量优化, 得到最终的半像素精度或 1/4 像素精度(H.264)的运动矢量, 转码中可以采用下列方法进行 MV 重用。

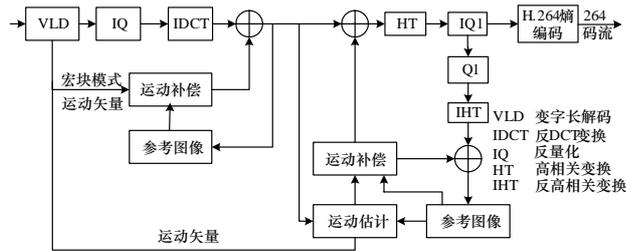


图 1 MPEG-2 到 H.264 运动矢量重用级联型转码器功能框架

3.1 整像素运动估计算法

由于 MPEG-2 的运动矢量也是经过全局运动搜索得到的, 得到的参考块与当前待编码块相关性最高, 因此可用 MPEG-2 的全局运动搜索结果来替代 H.264 中整像素全局运动搜索, 节省大量的计算量。方法是从 MPEG-2 码流经过变字长解码得到半像素精度的运动矢量, 将这个运动矢量换算成整像素精度, 送到 H.264 编码部分作为整像素搜索的运动矢量, 在这个运动矢量基础上, 按照 JM10.1^[7] 中的算法进行 H.264 的半像素和 1/4 像素精度的运动搜索, 得到亮度信号的运动矢量。按照这种方法每个块需要进行半像素精度和 1/4 像素精度的搜索, 共 18 个点的搜索。

这种算法的优点在于节省了编码端整像素搜索部分的计算量, 运算速度提高很多。由于整像素部分的搜索 H.264 和 MPEG-2 差异很小, 对于半像素和 1/4 像素搜索部分没有任何影响, 因此获得的编码质量很好, 与采用 H.264 编码器进行重编码相比, PSNR 差异很小。PSNR 产生差异主要是由于 MPEG-2 中进行的整像素搜索是基于 16×16 的宏块进行的, 而在 H.264 中, 如果宏块分成 16×8 , 8×16 , 8×8 甚至更小的 8×4 , 4×8 和 4×4 , 得到的整像素搜索结果可能是不同的, 这就可能引起最终预测结果与 JM 中的标准预测结果不同, 造成编码效率变化。

3.2 半像素运动估计算法

半像素精度运动搜索快速算法主要是为了充分利用 MPEG-2 的半像素精度运动矢量, 将 MPEG-2 的半像素运动矢量送到 H.264 编码部分作为半像素搜索的运动矢量, 在该运动矢量基础上, 按照 JM 中的算法再进行 H.264 的 1/4 像素精度的运动搜索, 得到亮度信号的运动矢量。该方法对每个块仅需要进行 1/4 像素精度的搜索, 共 9 个点的搜索。

这种方法的优点在于充分利用了 MPEG-2 的运动矢量, 节省了 H.264 部分的整像素和半像素搜索计算, 进一步提高转码的速度。然而, MPEG-2 的半像素内插是用相邻两点的均值得到, 而 H.264 中是用一个六阶有限冲击响应滤波器来得到的, 因此相同图像用不同的内插公式会得到不同的结果, 这就会造成运动估计的不同, 使得 MPEG-2 的运动矢

量在 H.264 的半像素图像中并不一定是最优的, 会引起编码效率变化。通过实验, 发现这种方法质量会有所下降, 相比于计算量的节省, 代价比较大, 因此需要对 MPEG-2 的运动矢量在 H.264 编码部分进行再搜索, 得到更好的运动估计结果, 本文按照这一想法进一步提出了运动矢量重用重搜索算法, 能够得到更好的编码效率。

3.3 运动矢量重搜索算法

由于 HMSA 对于编码质量影响很大, 因此需要在重用 MPEG-2 半像素精度运动矢量时, 进行再搜索, 得到高质量的运动估计。对 MPEG-2 运动矢量的重搜索是在运动矢量指向位置的周围按照 1/4 像素的精度开一个搜索窗, 在这部分内进行全搜索, 得到最匹配的运动矢量作为 H.264 的 1/4 像素精度的运动矢量送到编码部分。实验发现, 在兼顾考虑转码编码质量和速度的条件下, 窗的大小选择 MPEG-2 半像素运动矢量指向位置的上下左右各半个像素, 以 1/4 像素为单位的话就是上下左右各 2 个点, 效果较好, 如图 2 所示。其中, m 点是 MPEG-2 运动矢量指向的半像素点, 以这个点为中心进行 2 个点的搜索, a, c, e, k, o, u, w, y 都是半像素点, 其他的是 1/4 像素点。在窗内进行全局搜索, 就可以得到 1/4 像素精度的运动矢量。此算法需要进行 25 个点的搜索。

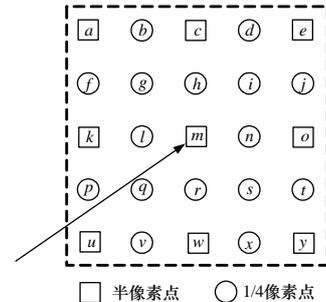


图 2 运动矢量重搜索优化

相比于 HMSA, MVRSA 算法在计算量上有一定增加, 但是编码质量提高很多。同采用 JM 的重编码转码器相比, 质量上损失很小, 而计算量节省很多。同 PMSA 相比计算量上也有一定增加, 每个块需要多进行 7 个点的搜索, 如果要提高编码质量, 需要增加窗的大小, 计算量会随之增加。

3.4 快速运动矢量优化算法

为进一步降低计算量, 提出一种用于 MPEG-2 到 H.264 转码处理中的快速运动矢量优化算法, 具体算法如图 3 所示。

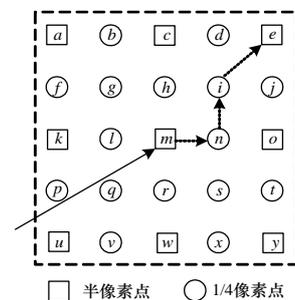


图 3 快速运动矢量优化

(1) 比较 MPEG-2 运动矢量指向的半像素点和上下左右相邻的 1/4 像素点, 即 m, h, l, n, r 5 个点, 从这 5 个点中找到最匹配的点。如果是中间的 m 点, 说明在窗内其他方向的运动搜索都不会好于原有的 MPEG-2 运动矢量, 那么就使用 m 点作为运动搜索的结果。否则, 找到最匹配的点, 以 n 为例, 进行下一步搜索。

(2)从(1)搜索中得到的最优点作为第2次搜索的中心,进一步比较相邻的1/4像素点,例如 n 周围的 i, o, s 3个点,同(1)一样,从这4个点中找到最优的,如果是 n 或 o 最优,那么就停止搜索,以这个点为运动搜索的结果,否则继续进行下一步。

(3)以(2)中的最优点为中心,比较周围的点。假设(2)的结果是 i 点,那么就继续搜索 d, e, j 3个点,从这4个点中找到最优的,作为运动搜索的结果。

上述方法最多仅需要计算11个点,同MVRSA相比每个块可以节省14个点,大约节省56%的计算量,同PMSA相比每个块节省7个点,大约节省39%的计算量,而编码质量略有下降,相比于HMSA,每个块需要多搜索2个点,但是编码质量明显提高。

4 实验结果及分析

实验所采用的测试码流为CIF(352×288)格式的Football, Paris, Flower和Mobile, QCIF(176×144)格式的Foreman, Mother和Akio,对文中提出的运动矢量优化算法和标准级联转换编码器进行比较,标准级联转换编码器采用标准TM5解码器和JM10.1编码器级联构成,在实验结果中记做REF。本文提出的快速算法用TM5解码器和相应编码器级联构成。输入码流是用TM5压缩的标准MPEG-2视频序列,长度均为30帧,帧率为30帧/s, GOP长度是12,帧结构为IPPP…。计算PSNR的参考图像是MPEG-2码流解码得到的原始图像。

4.1 不同算法的R-D曲线比较

在图4中给出了对于Mobile和Forman序列采用不同算法得到的R-D曲线。

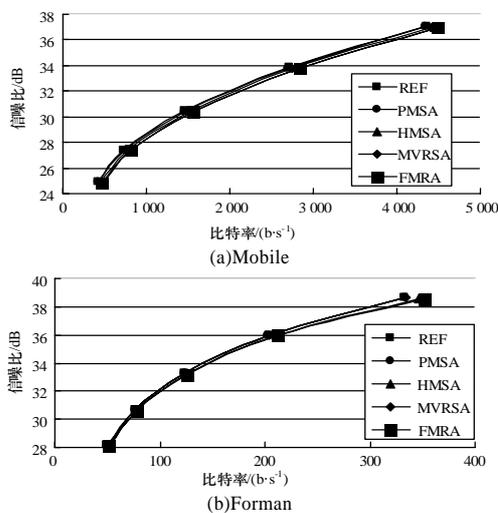


图4 不同算法比较R-D曲线

从图4中可以看出,对于不同序列,所比较的5个算法会有不同的表现,但是总体来说,采用PMSA算法的质量最接近标准算法,R-D曲线基本一致,这主要是由于PMSA算法最接近标准算法,在半像素搜索中利用了H.264中的六阶有限冲击响应滤波器来进行内插,使得半像素搜索的结果好于其他优化算法。HMSA算法质量最差,主要是由于在重用了MPEG-2的运动矢量后,HMSA算法对运动矢量进行的修正范围最小,仅仅是开了3×3的窗,而MVRSA和MVRSTA是在5×5的窗内进行的修正,PMSA更是在7×7的窗内进行的搜索,因此HMSA算法质量最差。MVRSA算法的搜索窗范围介于PMSA和HMSA之间,因此,其编码质量也介于两者之间,但是更接近于PMSA和REF的质量,与PMSA

质量差异很小,而比HMSA质量增加较多。FMRA算法由于搜索点数比MVRSA少,因此得到的质量比MVRSA差,但由于搜索窗比HMSA大,因此质量优于HMSA,但与PMSA, MVRSA和REF的算法相比质量上有一定下降。

4.2 不同算法时间比较

在表1中给出了不同序列采用不同算法进行运动估计所用的时间,同一算法对不同QP值进行运动估计时搜索的时间是一样的,因此计算量不变,所用时间没有明显变化,表中的时间数据是对不同QP值所用时间进行平均的结果,能够抵消随机误差,这样可以表示不同算法间的运算速度。实验机器配置为:CPU是Pentium 4 1.6 GHz,内存为512 MB。所有时间以秒为单位。

表1 不同运动矢量优化算法时间比较

CIF	REF	PMSA	HMSA	MVRSA	FMRA
Football	12.436 2	0.476	0.294 6	0.684 2	0.366 6
Flower	14.087 2	0.425 2	0.312 8	0.629 8	0.256 6
Paris	14.471 4	0.425 6	0.363 2	0.618 6	0.281 8
Mobile	14.305 8	0.510 4	0.327 8	0.689 2	0.361 6
QCIF	REF	PMSA	HMSA	MVRSA	FMRA
Foreman	3.573 4	0.108 8	0.0772	0.157	0.062 4
Mother	3.513 8	0.109 6	0.094 4	0.119 8	0.050 4
Akio	3.572 2	0.115 8	0.058 8	0.133 4	0.062 4

从表中可以看出,在相同分辨率下,不同的序列进行运动估计的时间基本一样,这是由于帧间预测对每个块进行搜索的次数是一样的,图像尺寸大导致块的数量多,计算量大,预测时间长。而在同一序列中,标准算法进行运动估计搜索的时间特别长,主要是由于标准算法中对于整像素部分的搜索占用了大量的计算量,导致计算速度很慢。而本文提出的几种算法中,HMSA计算量最小,之后依次为MVRSTA, PMSA和FMRA,由于省略掉了标准算法中的整像素搜索,因此搜索时间上节省了超过85%的时间,整个时间的节省能够达到95%,而质量上没有太大损失。

5 结束语

根据数字视频转换编码的特点,将MPEG-2的运动矢量进行了重用,提出了4种快速算法,并对不同算法进行了实验分析和比较。针对不同的应用,可以采用不同的快速算法。如果要保证转换编码的输出视频质量,例如:可以采用PMSA和MVRSA 2种算法,这2种算法的编码质量同标准级联转换编码器相比没有明显差别,如果要求转换编码的计算速度,可以采用HMSA和MVRSTA 2种算法,这2种算法的计算量较低,时间上仅为JM的3%左右,质量上虽略有下降,但下降不会超过0.5 dB,主观质量上不会有明显差异。

参考文献

- [1] Shimizu T, Yoneyama A, Yanagihara H, et al. A Two-stage Variable Block Size Motion Search Algorithm for H.264 Encoder[C]//Proc. of 2004 International Conference on Image Processing. Singapore: [s. n.], 2004: 1481-1484.
- [2] Cheng Yun, Dai Kui, Wang Zhiying, et al. Motion Search Method Based on Zero-block Detection in H.264 AVC[C]//Proc. of the 8th International Conference on Computer Supported Cooperative Work in Design. Xiamen, China: [s. n.], 2004: 739-743.

(下转第253页)