

AVS-H.264 视频转码快速算法

尚 凯, 张万绪

(西北大学信息科学与技术学院, 西安 710127)

摘 要: 根据 AVS 标准和 H.264 标准的特点, 提出一种适用于 AVS 向 H.264 转码的快速算法。该算法利用 AVS 标准中帧间编码宏块的分割模式信息, 采用模式映射方法避免 H.264 编码过程中重新确定每个宏块分割模式, 加快了转码速度。仿真实验结果表明, 该算法在不影响视频质量的前提下, 极大降低了转码的计算复杂度, 提高了转码效率。

关键词: 视频转码; H.264 标准; 音视频编码标准; 模式选择

Fast Algorithm for AVS-H.264 Video Transcoding

SHANG Kai, ZHANG Wan-xu

(Institute of Information Science and Technology, Northwest University, Xi'an 710127)

[Abstract] According to the features of AVS and H.264 standard, this paper proposes a rapid algorithm for AVS to H.264 transcoding. This algorithm improves the speed of transcoding by taking advantage of AVS partition mode information of interframe coding macroblock and using mode mapping methods to avoid reconsidering partition mode for each macroblock during H.264 coding process. Simulation results show that this algorithm can decrease the complexity of calculation and improve coding efficiency without affecting the quality of video.

[Key words] video transcoding; H.264; Audio Video coding Standard(AVS); mode selection

视频转码是对压缩视频流进行端到端的处理, 转码器的输入/输出都是压缩数据, 转码后的压缩码流更能适应传输带宽和接收端的要求。

1 转码器

在传统级联式像素域转码器(Cascade Pixel Domain Transcoder, CPDT)中, 解码器和编码器相互独立, 用此类转码器进行 AVS 标准向 H.264 标准的转码时, 先解码 AVS 视频流, 然后重新编码, 最后压缩成 H.264 视频码流, 其转码器结构见图 1。在该转码器中, 由于没有利用 AVS 码流中的有效信息来加快 H.264 编码过程, 因此转码过程计算量大, 转码效率受到限制。本文通过分析 2 种标准帧间宏块模式选择, 有效利用了 AVS 码流中宏块的分割模式信息, 加快了转码的速度, 相应的转码器结构见图 2。其算法是将 AVS 与 H.264 中相同的宏块分割模式直接映射, 把 AVS 中没有的模式按 H.264 的模式进行进一步细分, 在细分过程中, 本文采用基于纹理的模式选择方法, 避免了在 H.264 编码过程中重新确定每个宏块的分割模式, 从而减少了计算复杂度。

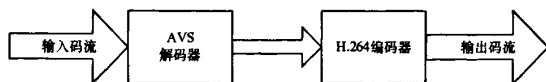


图1 传统转码器结构

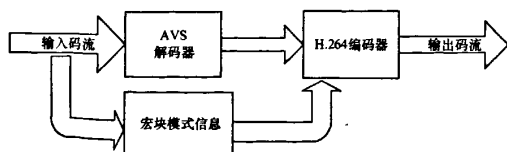


图2 改进的转码器结构

2 宏块分割模式

2.1 H.264 宏块分割模式

在 H.264 的帧间编码中采用宏块分割和亚分割的方法, 每个块可以按 16×16 、 16×8 、 8×16 、 8×8 分割, 在 8×8 宏块划分模式中, 还能进一步分割成 8×4 、 4×8 、 4×4 3 种模式^[1]。H.264 的宏块分割模式如图 3 所示。

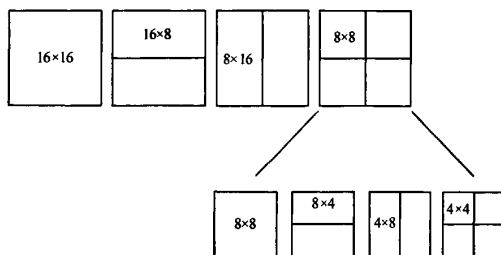


图3 H.264 宏块分割模式

每个宏块中每种分割都对应一个运动矢量, 在编解码过程中, 编码器对每个运动矢量和对应的宏块模式进行编码, 解码器根据每个宏块的分割模式和运动矢量信息进行解码。

除以上 7 种模式外, H.264 还支持 skip 模式, 该模式不包含预测残差和运动信息, 复杂度较小, 通常用于运动简单、背景变化不大的场景。H.264 编码时需要对每一种模式进行搜索, 并用率失真优化函数计算编码代价, 最后比较计算结果, 选择代价最小的模式作为帧间编码的宏块分割模式。

经过对多个典型测试序列的帧间模式划分的统计分析得出, skip 模式和较大尺寸的分割模式占整个模式划分的 90%

作者简介: 尚 凯(1984—), 男, 硕士研究生, 主研方向: 视频图像压缩; 张万绪, 副教授

收稿日期: 2010-01-22

E-mail: shangkai1984.good@163.com

以上,而 8×4 、 4×8 、 4×4 这3种模式只占不足10%^[2]。根据H.264的编码算法,在编码过程中要对每种模式进行搜索,并计算每种模式的率失真代价,由于所占比例非常小的模式依然要进行搜索和计算,因此会增加计算复杂度和编码时间。

2.2 AVS宏块分割模式

AVS的帧间模式划分种类比H.264少,只有4种模式,即 16×16 、 16×8 、 8×16 、 8×8 ^[3],去掉了所占比例不大但占用了编码计算过程大量时间的小块划分模式(8×4 、 4×8 、 4×4),因此,极大降低了编码复杂度。AVS的宏块分割模式如图4所示。

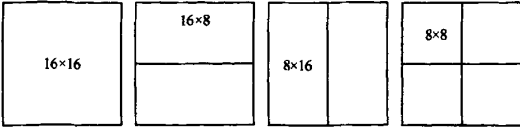


图4 AVS宏块分割模式

3 基于纹理的模式判别

对于 $M\times N$ 的块,DCT公式如下:

$$F(u,v) = c(u)c(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)u\pi}{16}\right] \times \cos\left[\frac{(2y+1)v\pi}{16}\right] \quad (1)$$

其中, $c(u), c(v) = \sqrt{\frac{1}{M}}, \sqrt{\frac{1}{N}}$, $u, v = 0$; $c(u), c(v) = \sqrt{\frac{2}{M}}, \sqrt{\frac{2}{N}}$, $u, v \neq 0$ 。

经变换后得到的交流分量系数的总能量为

$$E = \sum_{u=1}^{M-1} \sum_{v=1}^{N-1} F^2(u,v) \quad (2)$$

由于一个 $M\times N$ 的块在空间域的总能量与DCT变换后全部系数的总能量相等,因此 E 还可以表示为

$$E = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f^2(x,y) - \frac{1}{M \times N} \left[\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \right]^2 \quad (3)$$

其中, $\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f^2(x,y)$ 为此 $M\times N$ 块在空间域的总能量; $\frac{1}{M \times N} \left[\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \right]^2$ 为平均亮度具有的能量,即直流系数的能量。在一个 $M\times N$ 的块中,当相邻2个像素的值分别取得最大值和最小值时,此块就具有最大的交流系数总能量,可以表示为

$$E_{\max} = (M \times N) \frac{f_{\max}^2(x,y) + f_{\min}^2(x,y)}{2} - \frac{M \times N}{4} [f_{\max}(x,y) + f_{\min}(x,y)]^2 \quad (4)$$

其中, $f_{\max}(x,y)$ 和 $f_{\min}(x,y)$ 分别为 $f(x,y)$ 的最大值和最小值。得到 E 和 E_{\max} 后,本文用参数 β 来表示宏块的细节程度。

$$\beta = \frac{E}{E_{\max}} \quad (5)$$

β 取值范围为 $[0,1]$,纹理细节程度越高, E 越趋近于 E_{\max} ,从而 β 越趋近于1。所以,根据 β 的值可以判断宏块的细节程度。在本文中,当 $\beta > 0.9$ 时,认为宏块具有很高的细节,对于该块选用 4×4 的分割模式;当 $0.7 < \beta < 0.9$ 时,认为宏块具有较高细节,对于该块采用 8×4 或 4×8 分割模式;否则采用 8×8 宏块分割模式。通过以上方法可以预先排除一些模式,避免了对所有模式都进行搜索,降低了计算复杂度。

由于 16×8 、 8×4 模式比较适合水平纹理明显的区域,而

8×16 、 4×8 模式较适合垂直纹理明显的区域,因此若 $0.7 < \beta < 0.9$,则判断宏块采用 8×4 分割模式还是 4×8 分割模式时,首先要判断出该块的纹理方向,当块的纹理方向趋于水平方向时,采用 8×4 分割模式,否则采用 4×8 分割模式。判别宏块纹理方向的方法如下:

$$D_h = \sum_{x=0}^{M-1} \sum_{y=1}^{N-1} [f(x,y) - f(x,y-1)]^2 \quad (6)$$

$$D_v = \sum_{x=1}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - f(x-1,y)]^2 \quad (7)$$

当宏块的纹理趋于水平方向时, $D_h > D_v$ 。当宏块纹理趋于垂直方向时, $D_v > D_h$ 。当水平纹理宏块的纵向相邻像素值分别为最大值和最小值时, D_h 达到最大值而 D_v 为0。当垂直纹理宏块的横向相邻像素值分别为最大值和最小值时, D_v 达到最大值而 D_h 为0。如果令 D_h 的最大值为 $D_{h\max}$, D_v 的最大值为 $D_{v\max}$,则:

$$D_{h\max} = M \times (N-1) \times d^2 \quad (8)$$

$$D_{v\max} = N \times (M-1) \times d^2 \quad (9)$$

其中, d 为 $f(x,y)$ 最大值和最小值差的绝对值。通过以下2个等式判断纹理方向:

$$\alpha_h = \frac{D_h}{D_{h\max}} \quad (10)$$

$$\alpha_v = \frac{D_v}{D_{v\max}} \quad (11)$$

若 $\alpha_h > \alpha_v$,则说明当前块具有较多水平纹理,选择 8×4 分割模式。若 $\alpha_h < \alpha_v$,则说明当前块具有较强的垂直纹理,选择 4×8 分割模式。

4 算法流程

在转码之前先提取出以AVS标准编码的码流中的帧间宏块分割模式信息,然后判断每个宏块的分割模式是否为 16×16 、 16×8 、 8×16 这3种模式之一,若是,则采用直接映射的方式把宏块模式映射到H.264编码模式中,否则用基于纹理的模式判别法对其他模式进行判定,将最终的分割模式作为H.264编码的宏块分割模式。

5 仿真实验结果

为了验证本文算法的有效性,将它与传统级联式转码器进行对比实验。使用的AVS参考软件为rm52g,H.264参考软件为JM86,视频测试序列使用akiyo、news、foreman。

在仿真实验中,输入的视频序列格式为QCIF,画面组(Group Of Pictures, GOP)格式为IPPP, GOP长度为10,帧率为30 f/s,测试序列长度为100帧。而在H.264中,设置JM86参考软件的编码参数为QP=28,最大搜索范围为16,帧率为30 f/s,参考帧为2帧。

实验结果如表1所示,可以看出,本文提出的算法与传统的转码器算法相比,在峰值信噪比平均仅下降约0.25 dB、比特率(Bit Rate, BR)平均仅增加4%的情况下,平均运动估计消耗的时间MVTC减少了80%以上,说明本算法有效。

表1 2种算法的性能比较

| 算法 | 指标 | akiyo | news | foreman |
|------|--------------------------|---------|---------|---------|
| CPDT | PSNR/dB | 38.734 | 37.281 | 36.283 |
| 本文算法 | | 38.491 | 37.030 | 36.019 |
| CPDT | BR/(Kb·s ⁻¹) | 75.790 | 136.330 | 180.490 |
| 本文算法 | | 78.570 | 142.080 | 186.260 |
| CPDT | MVTC/ms | 675.390 | 695.070 | 711.430 |
| 本文算法 | | 131.700 | 136.230 | 140.590 |

(下转第244页)

```

; Clear Timer 0 Interrupt Response Bit
CALL    OSIntExit
%RestoreContext(); Restore new task's context
BSET IEN    ; Enable interrupts
RETI      ; Return to interrupted task
OSTickISR   ENDP

```

4 应用实例

在编写完所有的移植代码之后，可以编写任务程序进行测试。以轿车车门控制器设计为例，将其应用到轿车车门的控制系统设计中，该系统包括 4 个任务：车窗控制，按键输入，后视镜调整和看门狗。系统流程如图 2 所示。

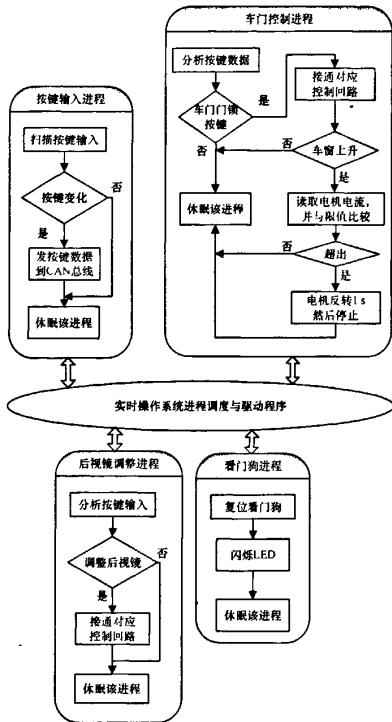


图 2 系统流程

轿车车门控制系统硬件示例如图 3 所示。

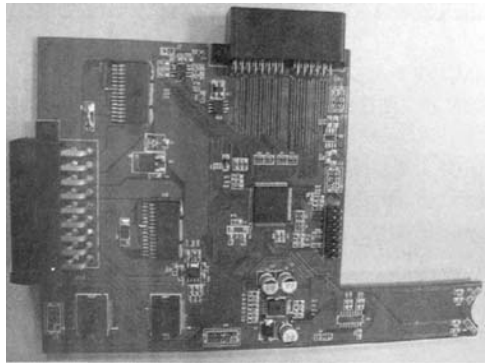


图 3 轿车车门控制系统硬件示例

长时间运行后，系统工作正常，由此证明， $\mu\text{C}/\text{OS-II}$ 被成功地移植到了 XC164 控制器上，实现了轿车车门的控制。

5 结束语

本文介绍了 $\mu\text{C}/\text{OS-II}$ 实时操作系统移植到 XC164CS 微控制器上的方法，并将其应用到轿车车门控制系统的设计中，证明了移植的正确性。移植后的软件开发平台可作为一种通用的基于 $\mu\text{C}/\text{OS-II}$ 实时操作系统的嵌入式软件开发平台，尤其适用于各种汽车电子的嵌入式软件开发。

参考文献

- [1] 季 虹, 付少锋, 车向泉, 等. 实时嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 内核的分析与改进[J]. 计算机工程, 2007, 33(16): 246-247.
- [2] 沈金荣, 刘 翔. $\mu\text{C}/\text{OS-II}$ 内核结构分析及多任务调度实现[J]. 计算机工程, 2006, 32(23): 85-87.
- [3] Labrosse J J. 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ [M]. 邵贝贝, 译. 北京: 北京航空航天大学出版社, 2003.
- [4] Infineon Inc.. XC164 用户手册 V2.1[Z]. 2004-03.
- [5] 程晓宇, 毕笃彦, 蔡 晖. $\mu\text{C}/\text{OS}$ 在龙芯处理器上的移植及性能分析[J]. 计算机工程, 2009, 35(1): 213-215.

编辑 顾姣健

(上接第 235 页)

图 5、图 6 分别为 akiyo 和 foreman 2 个标准测试序列使用 2 种算法后，前 30 帧图像的峰值信噪比。可以看出，本文提出的快速算法的峰值信噪比下降不多，在人眼视觉可接受的范围内。

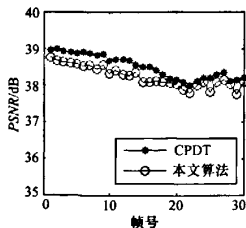


图 5 akiyo 序列峰值信噪比

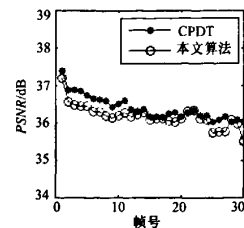


图 6 foreman 序列峰值信噪比

6 结束语

本文算法充分利用 AVS 帧间运动估计的宏块分割模式


信息，采用直接映射的方式把 AVS 中 16×16 、 16×8 、 8×16 模式直接映射到 H.264 编码模式中。当 AVS 为 8×8 宏块模式时，采用基于纹理的模式判别法决定下一步分割模式。该算法以峰值信噪比的少量下降和比特率的少量上升为代价，极大降低了编码消耗的时间。

参考文献

- [1] 毕厚杰. 新一代视频压缩编码标准——H.264/AVC[M]. 北京: 人民邮电出版社, 2005.
- [2] 董海燕, 张其善. 低复杂度的 H.264 帧间宏块编码模式选择算法[J]. 计算机工程, 2006, 32(12): 221-222.
- [3] 王蕾睿, 沈礼权, 张兆扬. 一种基于 AVS 视频编码的快速帧间预测模式选择算法[J]. 中国图象图形学报, 2007, 12(10): 1794-1797.

编辑 陈 晖

AVS-H. 264视频转码快速算法

作者: [尚凯](#), [张万绪](#), [SHANG Kai](#), [ZHANG Wan-xu](#)
作者单位: [西北大学信息科学与技术学院, 西安, 710127](#)
刊名: [计算机工程](#) 
英文刊名: [COMPUTER ENGINEERING](#)
年, 卷(期): 2010, 36(12)

参考文献(3条)

1. [王蕾睿;沈礼权;张兆扬](#) [一种基于AVS视频编码的快速帧间预测模式选择算法](#)[期刊论文]-[中国图象图形学报](#) 2007(10)
2. [董海燕;张其善](#) [低复杂度的H. 264帧间宏块编码模式选择算法](#)[期刊论文]-[计算机工程](#) 2006(12)
3. [毕厚杰](#) [新一代视频压缩编码标准—H. 264/AVC](#) 2005

本文链接: http://d.g.wanfangdata.com.cn/Periodical_jsjgc201012081.aspx