

# 对 H.264/AVC 中自适应滤波技术的评测

陈亚菲, 陈生潭, 董元平

(西安电子科技大学 机电工程学院, 陕西 西安 710071)

**摘要:** H.264/AVC 标准中采用自适应滤波技术消除方块效应, 根据具体量化参数、宏块编码模式自适应调整滤波过程中的修正参数, 详细分析了为提高图像主观质量所采取的关键措施: 阈值判决条件、修正参数计算公式、在解码之后随即进行滤波以免误差通过参考图像帧扩散等, 并通过标准测试模型 JM86 实验验证了该滤波方法的优越性。

**关键词:** 自适应滤波 块效应 宏块编码模式 H.264/AVC

H.264/AVC 视频编解码标准有三个档次: 基本档次、主要档次和增强档次, 其中基本档次主要用于会议电视、可视电话等极低码率系统<sup>[1]</sup>。方块效应(Blocking Effect)即块噪声, 是分块变换编码的主要缺点, 限制了低比特率视频系统的性能。

H.264/AVC 中方块效应产生的原因有两个: 最重要的一个原因是来自基于块的帧内和帧间预测残差的 DCT 变换, 若变换系数在量化过程中量化相对粗糙, 则反量化恢复的变换系数带有误差, 造成重构图像在图像块边界上的视觉不连续; 第二个原因来自运动补偿, 运动估计中块匹配不可能绝对准确, 因此复制块的边界上就会产生数据不连续。

消除视频图像的块效应, 按原理可分为块重叠、滤波、自适应预测三类方法<sup>[2]</sup>。块重叠法计算量大且视觉效果不理想, 较优的方法是自适应与滤波相结合。滤波方式可分后滤波和环滤波, H.264/AVC 采用更优的环滤波, 它参与编码环节, 在重构帧时做滤波, 再进行运动补偿, 使编码效率更高。而后滤波在解码端结果输出之前才进行, 没有参与编码过程。

块噪声的大小与图像内容本身及其编码方式有很大关系, 因此 H.264/AVC 采用的自适应滤波系统, 根据帧编码类型、方式及量化参数等因素来确定滤波强度、滤波范围和修正参数<sup>[3-5]</sup>, 体现了自适应性; 因环滤波参与了编码环节, 重构帧须经过滤波之后才可存入帧存储器作为下一待解码帧的参考帧, 这样避免了累积误差的扩散, 提高了图像的主观质量。

## 1 边界分析和自适应性

### 1.1 边界分析

H.264/AVC 的去方块滤波器使去除编解码算法带来了方块效应, 以  $16 \times 16$  像素大小的宏块为单位, 按光栅扫描方式对重构帧图像的边界做滤波。

需要滤波的边界是除图像边界和片边界之外的所

有宏块边界, 有水平和垂直方向两种, 如图 1 中虚线所示。定义两个相邻  $4 \times 4$  块中一条直线带上的采样点为  $p_3 p_2 p_1 p_0 q_0 q_1 q_2 q_3$ , 假设在  $p_0$  和  $q_0$  之间是图像的真实边界, 用加粗黑线表示。

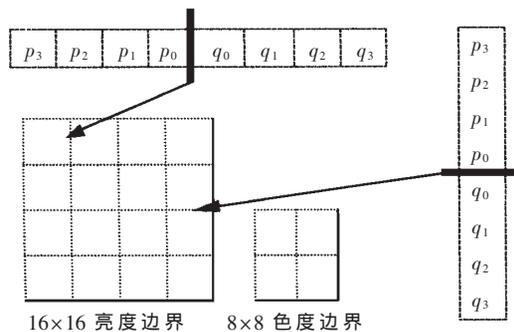


图 1 宏块中的边界

滤波要先判断边界是真实边界还是方块效应下的虚假边界。对真实边界不滤波, 对虚假边界则要根据实际情况来选择不同强度的滤波。

### 1.2 自适应性

去方块滤波器通过特定参数的调整以适应不同水平的需要, 分别满足对视频序列片级、图像块边界级和采样点级的滤波。

#### 1.2.1 对块边界级的自适应性

边界强度 BS (Boundary Strength) 决定对滤波参数的选择, 并控制滤波程度。色度块与相应亮度块的 BS 一致, 不用额外单独计算; 对  $4 \times 4$  子块, BS 一般取值在 0~4 之间, 与边界性质有关。滤波强度与编码模式的关系如表 1 所示。BS 的取值依据表 1 来判断, 设 A 块和 B 块为两相邻图像宏块, 表中条件是从上至下依次判断, 直到某一条件满足即给 BS 相应赋值。BS=4, 采用最强滤波; BS=0, 不采用滤波; BS=1~3, 采用普通滤波。

#### 1.2.2 对采样点级的自适应性

去方块滤波中, 重要的是区分真实边界和由 DCT 及

表 1 滤波强度与编码模式的关系

图像块模式和编码条件	BS
A 块或 B 块为帧内预测且该边界为宏块边界	4
A 块或 B 块为帧内预测	3
A 块或 B 块对残差编码	2
A 块与 B 块的运动矢量之差不小于 1 个亮度采样点的距离	1
A 块与 B 块的运动补偿的参考帧不同	1
其他	0

量化造成的虚假边界,通过分析每个待滤波边界两侧的采样点来区分这两种情况。一般真实边界两侧的像素值梯度差要比虚假边界两侧的像素值梯度差大,定义与量化有关的因子  $\alpha$ 、 $\beta$  来判断: $\alpha$  表示两块间的边界阈值, $\beta$  表示各块内的边界阈值,依据图 1 有:

$$|p_0 - q_0| < \alpha \quad (1)$$

$$|p_1 - p_0| < \beta \quad (2)$$

$$|q_1 - q_0| < \beta \quad (3)$$

仅当上述三个条件同时满足时,采样点才需要滤波。 $\alpha$  和  $\beta$  可由经验公式计算得到:

$$\alpha(x) = 0.8(2^{x/6} - 1) \quad (4)$$

$$\beta(x) = 0.5x - 7 \quad (5)$$

$x$  是量化参数  $qp(0 \sim 51)$ , (1) 式、(2) 式反映了滤波强度和待滤波图像质量的关系:  $qp$  较大时, 编码误差大, 较多的边界需要被滤波, 而阈值随  $qp$  增加而增大。

### 1.2.3 对片级的自适应性

在片级, 编码器用片头句法元素传递偏移量加给  $qp$  来调整  $\alpha$  和  $\beta$ : 用正的偏移增加滤波强度, 去除次优运动估计、编码模式选择等引起的额外方块, 改善低分辨率图像的视觉质量; 用负的偏移来减少滤波强度, 保护高分辨率图像的细节不被滤波作用模糊。

## 2 滤波过程

滤波操作基于宏块进行。依据图 1 对当前宏块每八个相邻采样点组成一个条带, 依次对垂直/水平边界做水平/垂直滤波, 之后滤波下一宏块; 对亮度宏块, 需要判定是用强滤波来修正边界上的采样点及两个内部采样点, 还是用普通滤波来修正边界上的采样点及一个内部采样点; 色度宏块滤波仅修正边界采样点。

### 2.1 BS 值为 1~3 时的普通滤波

滤波分为基本滤波运算和限幅两个阶段。

#### 2.1.1 基本滤波运算

满足(1)~(3)式时, 采用一个 4-tap 线性滤波器, 输入为  $p_1 p_0 q_0 q_1$ , 输出为修正后的  $p'_0 q'_0$ :

$$p'_0 = p_0 + \Delta_0 \quad (6)$$

$$q'_0 = q_0 + \Delta_0 \quad (7)$$

$$\Delta_0 = (4(q_0 - p_0) + (p_1 - q_1) + 4) \gg 3 \quad (8)$$

相应脉冲响应为  $(1, -4, 4, -1)/8$ , “ $\gg$ ”表示右移位运算, 等效于除法。亮度滤波时若还满足:

$$|p_2 - p_0| < \beta \quad (9)$$

$$\text{或满足 } |q_2 - q_0| < \beta \quad (10)$$

则说明块内部存在虚假边界, 要设计另一滤波器修正  $p_1$  或  $q_1$ , 二者滤波相似。以  $p_1$  为例:

$$p'_1 = p_1 + \Delta_{p1} \quad (11)$$

$$\Delta_{p1i} = (p_2 + ((p_0 + q_0 + 1) \gg 1) - 2p_1) \gg 1 \quad (12)$$

相应脉冲响应为  $(1, -2, 0.5, 0.5)/2$ 。

#### 2.1.2 限幅

$\Delta_0$  是根据初始值  $\Delta_{0i}$  限幅得到, 若  $\Delta_{0i}$  直接用于滤波, 则可能导致滤波频率过低以及出现图像模糊:

$$\Delta_0 = \begin{cases} -c_0 & \Delta_{0i} \leq -c_0 \\ \Delta_{0i} & -c_0 < \Delta_{0i} < c_0 \\ c_0 & \Delta_{0i} \geq c_0 \end{cases} \quad (13)$$

$c_0$  由表 2 查找得到。

表 2 滤波限幅变量  $c_0$  的查找表<sup>[1]</sup>

$c_0$	qp 与偏移量相加后的和							
	<17	.....	21	22	.....	41	.....	51
BS=1	0	.....	0	0	.....	4	.....	13
BS=2	0	.....	1	1	.....	5	.....	17
BS=3	0	1	1	1	.....	8	.....	25

$\Delta_{p1}$  由  $\Delta_{p1i}$  从式(13)限幅得到。 $q$  点操作与其相似, 但在式(12)中用  $q_2 q_1$  代替  $p_2 p_1$ 。对色度滤波, 只有  $p_0 q_0$  被修正, 其判断、修正方法类似于亮度滤波, 只是其限幅变量需要在表 2 的变量值上再加 1。

### 2.2 BS 值为 4 时的强滤波

$$\text{设约束条件 } |p_0 - q_0| < (\alpha \gg 2) + 2 \quad (14)$$

当式(9)和式(14)都成立时, 则进行亮度强滤波, 修正边界上采样点及两个内部采样点:

$$p'_0 = (p_2 + 2p_1 + 2p_0 + 2q_0 + q_1 + 4) \gg 3 \quad (15)$$

$$p'_1 = (p_2 + p_1 + p_0 + q_0 + 2) \gg 2 \quad (16)$$

$$p'_2 = (2p_3 + 3p_2 + p_1 + p_0 + q_0 + 4) \gg 3 \quad (17)$$

否则只要(9)式和(14)式中有一个不成立或对色度点只由下式修正:

$$p_0: p'_0 = (2p_1 + p_0 + q_1 + 2) \gg 2 \quad (18)$$

$q$  点修正方法相同, 只是选择亮度滤波器时用(10)式和(14)式作判断。

## 3 实验仿真和结果分析

基于 H.264/AVC JM86 标准模型做性能评测, 其实验条件是: Pentium(R)4 CPU 2.60GHz, 256MB 内存, Windows XP, 测试序列为 100 帧的 Foreman.QCIF, YUV 采样格式 4:2:0。

程序处理流程的 C 语言伪代码如下:

```
//滤波一个图像帧里的所有宏块, 按光栅扫描方式
for(i=0; i<PicSizeInMbs; i++) //宏块个数循环
{ get_mb_position(); //获取宏块地址
  LeftMbFlag, TopMbFlag, EdgeCond; //宏块边界状态
  CheckAvailabilityOfNeighbors(); //检查相邻像素点
  for(dir=0; dir<2; dir++); //先垂直边界再水平边界滤波
  { for(edge=0; edge<4; edge++) //每个边界方向上有 4 条
    { if(edge||EdgeCondition) //若是可以滤波的边界
```

```

{ GetStrength(); //获取 BS 值
  StrengthSum; //用于排除 BS 为全零的宏块,不滤波
  if(StrengthSum EdgeLoop(); //亮度,色度滤波
} }
//EdgeLoop()函数的伪代码如下:
PelNum=yuv? 8:16; //宏块的每行采样点数目
//yuv=0 对应亮度滤波,yuv=1 对应色度滤波
for(pel=0; pel<PelNum; pel++)
{ xQ,yQ; //由 idx 与 edge 计算得到
  (xP,yP)=getNeighbour(xQ,yQ); //p 点由 q 点计算得到
  MbQ,MbP; //获取 q 点和 p 点所在宏块地址
  bsIdx; //BS 值存入为矩阵后的索引值
  if(P 点存在||允许滤波标志为真)
  { incQ,incP,SrcQ,SrcP; //修正前后坐标矩阵初始化
    QP; //P、Q 块的平均量化参数,亮度/色度不同
    L0~L3,R0~R3; //8 个采样点
    if(Strength[bsIdx]) //BS 值不为零
    { if(abs(R0-L0)<Alpha) //式(1)成立
      { C0; //滤波限幅变量矩阵的指针
        if(abs(R0-R1)<Belta&&abs(L0-L1)<Belta)
        { if(! yuv) {aq,qp} //亮度滤波,判决条件赋给变量
          if(Strength[bsIdx]=4) //强滤波
            {if(yuv) SrcQ,SrcP; //色度,修正 p0,q0
              else{gap=absDelta<<((Alpha>>2)+2);
                aq&=gap,ap&=gap; //判决条件合成
                SrcQ=aq?():();SrcP=ap?():(); //修正
              } } //强滤波结束
            else{c0; //色度或亮度滤波时,调整限幅变量
              dif; //限幅得到式(13)中的 delta 值
              SrcQ[0],SrcP[0]; //修正 q0,p0
              if(! yuv) //亮度时,判断是否修正 q1,p1
                {if(aq) SrcQ[];if(ap) SrcP[];} }
            } } } //整个滤波过程结束
    函数中,复杂的条件判断用嵌套 if 语句实现,且阈值判断不等式变成二值变量及三目逻辑运算,使得整个程序较为简单实效地完成了滤波操作。
  
```

滤波效果分别如图 2、图 3 所示。图 2 说明 30 帧序

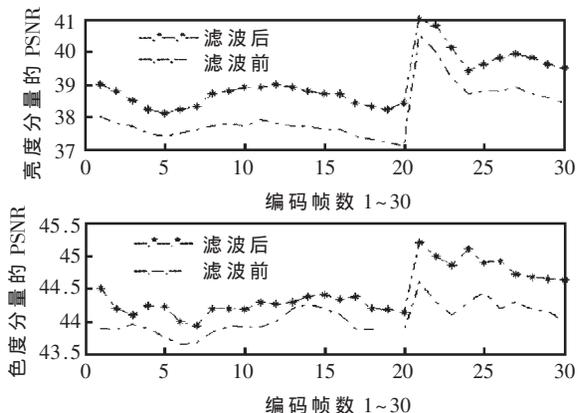


图 2 滤波提升序列图像的 PSNR



(a)滤波前



(b)滤波后

图 3 去方块效应滤波操作

列图像在解码过程中,其峰值信噪比(PSNR)受滤波影响而变化,在滤波后 PSNR 平均提升约 0.68 个 dB;图 3 是某一帧图像在滤波前后的画面,滤波后方块效应明显减少,图像质量有较明显的提高,码率节省约 7%。

H.264/AVC 的自适应滤波技术用于消除编解码算法带来的方块效应,是根据量化参数等来确定滤波参数,体现了滤波的自适应性,并可靠提高了图像质量。简单实效的阈值判决条件和修正参数算法,在标准测试模型 JM86 中得到验证。但是滤波增加了解码器端的计算复杂度,占用了约 1/3 的计算时间。因此寻求低复杂度、优化滤波算法的方法是需要继续研究的问题。

参考文献

- 1 ITU-T Rec.H.264(03/2005)-Pre-published version[S], ISO/IEC 14496-10.2005
- 2 王雁来.基于 H.264 的高清晰视频编码技术研究[D].北京工业大学硕士学位论文,2004
- 3 LIST P,JOCH A,LAINEMA J et al.Adaptive Deblocking Filter[J].IEEE Transactions on Circuits and Systems for Video Technology, 2003; 13(7)
- 4 WIEGAND T,SULLIVAN G J,BJONTEGAARD G et al. Overview of the H.264/AVC video coding standard[J].IEEE Transactions on Circuits and Systems for Video Technology, 2003;(13):560~576
- 5 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

(收稿日期:2006-07-19)