

# 一种快速的像素级 AVS 环路滤波算法

焦梅<sup>1</sup>, 彭玉华<sup>1</sup>, 刘微<sup>2</sup>, 游娜<sup>1</sup>

(1. 山东大学信息科学与工程学院, 济南 250100; 2. 海信集团有限公司数字多媒体技术国家重点实验室, 青岛 266071)

**摘要:** 环路滤波器可以在编解码过程中降低块效应。针对现有环路滤波方法计算量大、耗时多的问题, 提出一种新的 AVS 环路滤波算法。该算法利用图像内  $8 \times 8$  块之间的相关性, 求得边界强度, 修改相应像素值。实验结果证明, 在保证图像质量基本不变的前提下, 新算法减少了环路滤波时间, 提高了编解码速度。

**关键词:** AVS 技术; 块效应; 环路滤波; 边界平滑度; 边界强度

## Fast AVS Loop-filtering Algorithm at Pixel Level

JIAO Mei<sup>1</sup>, PENG Yu-hua<sup>1</sup>, LIU Wei<sup>2</sup>, YOU Na<sup>1</sup>

(1. School of Information Science and Engineering, Shandong University, Jinan 250100;

2. State Key Laboratory of Digital Multi-media Technology, Hisense Group Ltd., Qingdao 266071)

**【Abstract】** Loop-filter can reduce the blocking effect in the codec process. Aiming at the problem that existing algorithms have high computational complexity and costing a lot of time, a new AVS loop-filtering algorithm is presented. This algorithm takes advantage of the correlation between  $8 \times 8$  bolcks of a picture to obtain boundary strength and modify the related pixels. Experimental results show that the new algorithm can reduce the cost time of loop-filtering and raise the codec speed while maintaining almost the same quality.

**【Key words】** AVS technology; blocking effect; loop-filtering; boundary flatness; boundary strength

### 1 概述

数字音视频编解码标准(Audio Video coding Standard, AVS)<sup>[1]</sup>和 H.264<sup>[2]</sup>采用基于宏块的编码框架, 将输入视频分割成宏块, 并对其进行预测、运动估计、变换、量化、熵编码等处理。在上述框架中, 因为做了分块处理, 且对每个  $8 \times 8$  的块进行整数离散余弦变换(Discrete Cosine Transform, DCT), 所以忽略了块与块之间的像素相关性。对每个块的 DCT 系数进行量化时, 要将 DCT 变换后的系数除以量化系数后取整, 因此, 丢弃了一些对图像质量影响不大的高频分量。当压缩率较高时, 会丢失块边缘的大量高频信息, 导致在重建图像中, 块的边界处出现不连续跳变, 即块效应<sup>[3]</sup>。为了去除块效应, 引入了环路滤波技术。

环路滤波以宏块为单位, 按光栅扫描顺序依次处理。对图像中每个宏块的亮度和色度(图 1)分别做环路滤波: (1)从左到右对垂直边界滤波; (2)从上到下对水平边界滤波。当前宏块上边或左边的样本值可能在以前的宏块环路滤波过程中已经被修改, 它们是当前宏块环路滤波的输入, 可能进一步被当前宏块环路滤波修改。当前宏块垂直边界滤波过程中修改的样本值作为水平边界滤波过程的输入。

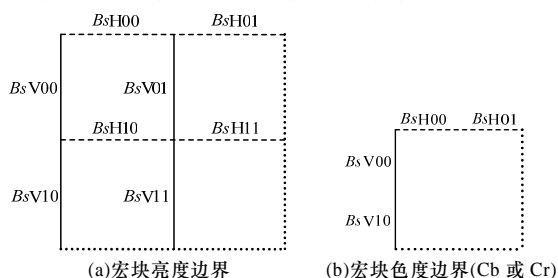


图 1 宏块中需要滤波的边界

### 2 AVS基准档次环路滤波算法

AVS 基准档次的环路滤波利用宏块类型、宏块中  $8 \times 8$  亮度块的运动矢量计算  $B_s$ , 并根据  $B_s$  的值进行滤波。

#### 2.1 边界强度 $B_s$ 的推导过程

$B_s$  的推导过程如下:

(1)如果边界两边的 2 个  $8 \times 8$  块中的 1 个或 2 个属于帧内预测宏块, 则  $B_s$  等于 2。

(2)当边界两边的 2 个  $8 \times 8$  块都不属于帧内预测宏块时, 如果 picture\_coding\_type 的值为“01”, 且满足以下 2 个条件中的一个, 则  $B_s$  等于 1; 如果 picture\_coding\_type 的值为“01”且以下 2 个条件都不满足, 则  $B_s$  等于 0。2 个条件如下:

1) 2 个块的参考图像不同。

2) 2 个块的参考图像相同, 但 2 个运动矢量分量中任意一个分量的差值大于等于一个整像素。

(3)当(1)和(2)都不满足时, 按以下方法计算  $B_s$ :

1) 当 2 个块  $L$  和  $R$  的前向索引参考值和后向索引参考值分别相等时, 若以下 2 个条件中任意一个成立, 则  $B_s$  等于 1; 否则,  $B_s$  等于 0。2 个条件如下:

① 2 个块的前向运动矢量分量中任意一个分量的差值大于等于一个整像素。

② 2 个块的后向运动矢量分量中任意一个分量的差值大于等于一个整像素。

**基金项目:** 山东省博士后科研专项基金资助项目(200602009); 山东省科技攻关基金资助项目(2005GG3201117)

**作者简介:** 焦梅(1982-), 女, 硕士, 主研方向: 视频编解码技术, 数字图像处理; 彭玉华, 教授、博士生导师; 刘微, 博士后; 游娜, 硕士

**收稿日期:** 2008-11-24 E-mail: jm726@163.com

2)否则,  $B_s$  等于 1。

## 2.2 滤波过程

图 2 描述了 2 个  $8 \times 8$  的块, 即  $L$  和  $R$  在水平或垂直边界两边的 6 个样本点(边界用较粗的线表示)。用  $L_0^-, L_1^-, R_0^-, R_1^-$  分别表示  $L_0, L_1, R_0, R_1$  滤波后的样本值。

$L_2$	$L_1$	$L_0$	$R_0$	$R_1$	$R_2$
-------	-------	-------	-------	-------	-------

图 2  $8 \times 8$  块垂直或水平边界两边的像素点

如果下式为真, 则对边界样本滤波。

$$B_s! = 0 \& \& Abs(L_0 - R_0) < \alpha$$

$$\& \& Abs(L_1 - L_0) < \beta \& \& Abs(R_1 - R_0) < \beta$$

其中,  $\alpha$ 和 $\beta$ 为块边界阈值, 可以通过与  $IndexA$  和  $IndexB$  的对应关系查表确定。本文定义

$$ap = Abs(L_2 - L_0), aq = Abs(R_2 - R_0),$$

$$Absdelta = Abs(L_0 - R_0) < ((\alpha \gg 2) + 2)$$

### 2.2.1 $B_s$ 等于 2 时的边界滤波过程

对亮度块边界两边的样本  $L_0, L_1, R_0, R_1$  的滤波过程如下:

若  $((ap < \beta) \& \& (Absdelta == 1))$  成立, 则修改  $L_0$  和  $L_1$ , 即

$$L_0^- = (L_1 + (L_0 \ll 1) + R_0 + 2) \gg 2$$

$$L_1^- = ((L_1 \ll 1) + L_0 + R_0 + 2) \gg 2$$

否则只修改  $L_0$ , 即

$$L_0^- = ((L_1 \ll 1) + L_0 + R_0 + 2) \gg 2$$

同理, 如果  $((aq < \beta) \& \& (Absdelta == 1))$  成立, 则修改  $R_0$  和  $R_1$ , 即

$$R_0^- = (R_1 + (R_0 \ll 1) + L_0 + 2) \gg 2$$

$$R_1^- = ((R_1 \ll 1) + R_0 + L_0 + 2) \gg 2$$

否则只修改  $R_0$ , 即

$$R_0^- = ((R_1 \ll 1) + R_0 + L_0 + 2) \gg 2$$

其中,  $L_0^-$  和  $R_0^-$  分别为  $L_0$  和  $R_0$  滤波后的值。如果在上述滤波过程中,  $L_1^-$  ( $R_1^-$ ) 不被赋值, 则不对  $L_1$  ( $R_1$ ) 滤波。否则,  $L_1^-$  或  $R_1^-$  为  $L_1$  或  $R_1$  滤波后的值。

色度块边界两边的样本  $L_0$  和  $R_0$  采用同样的方法滤波, 不对  $L_1$  和  $R_1$  滤波。

### 2.2.2 $B_s$ 等于 1 时的边界滤波过程

对  $L_0$  和  $R_0$  滤波的计算过程如下( $L_0^-$  和  $R_0^-$  分别为  $L_0$  和  $R_0$  滤波后的值):

$$\delta = Clip3(-C, C, (((R_0 - L_0) \times 3 + (L_1 - R_1) + 4) \gg 3))$$

$$L_0^- = Clip1(L_0 + \delta)$$

$$R_0^- = Clip1(R_0 - \delta)$$

判断是否需要到  $L_1$  和  $R_1$  滤波, 计算过程如下:

(1)如果为色度边界, 则不对  $L_1$  和  $R_1$  滤波。

(2)如果在亮度边界处有  $ap$  小于  $\beta$ , 则对  $L_1$  滤波, 滤波后的值为

$$L_1^- = Clip1(L_1 + Clip3(-C, C, (((L_0^- - L_1) \times 3 + (L_2 - R_0^-) + 4) \gg 3)))$$

(3)如果在亮度边界处有  $aq$  小于  $\beta$ , 则对  $R_1$  滤波, 滤波后的值为

$$R_1^- = Clip1(R_1 - Clip3(-C, C, (((R_1 - R_0^-) \times 3 + (L_0^- - R_2) + 4) \gg 3)))$$

在上述滤波过程中,  $C$  称为滤波裁减参数,  $C$  与  $IndexA$  之间的关系可以查表得到。

## 3 快速的像素级 AVS 环路滤波算法

本算法先判断当前宏块是否需要滤波, 如果需要滤波, 则对每个  $8 \times 8$  的亮度块进行如下处理: 判断第 1 行(对垂直边界滤波)或第 1 列(对水平边界滤波)的边界是否为真实边

界。如果不是真实边界, 则计算当前行或列的边界平滑度, 并根据边界平滑度计算滤波强度  $B_s$ 。根据  $B_s$  修改当前  $8 \times 8$  块边界两边的像素值。如果是真实边界, 则跳过当前  $8 \times 8$  块的滤波, 判断下一个  $8 \times 8$  亮度块第 1 行或第 1 列的边界是否为真实边界, 如图 3 所示, 其中, 粗线表示需要滤波的边界。

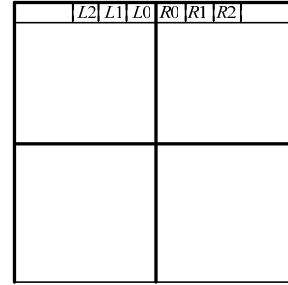


图 3 当前宏块

### 3.1 当前宏块边界的判断

当前宏块需要滤波的边界如果满足以下条件中的一个, 则不需要进行滤波, 直接跳到下一条边界:

- (1)当前帧为 B 帧或 P 帧, 边界为当前宏块的内部边界, 且亮度块的  $cbp$ (编码块模板)为 0。
- (2)当前帧为 B 帧或 P 帧, 边界为 2 个宏块的边界, 且 2 个宏块的亮度块的  $cbp$  都为 0。
- (3)当前帧为 B 帧或 P 帧, 2 个宏块的参考帧相同, 且 2 个宏块的运动向量的差小于一个整像素点。

### 3.2 新算法的 $B_s$ 推导过程

对宏块中  $8 \times 8$  亮度块和色度块计算  $B_s$ 。

如果第 1 行或第 1 列的边界两边的像素(图 2)满足

$$Abs(L_0 - R_0) < \alpha \& \& Abs(L_1 - L_0) < \beta$$

$$\& \& Abs(R_1 - R_0) < \beta$$

则进行下述处理, 否则令  $B_s=0$  并跳到下一个需要滤波的  $8 \times 8$  块的起始位置。 $\alpha$ 与 $\beta$ 依赖索引值  $IndexA$  和  $IndexB$  查表确定。

$$flatness = (abs(L_0 - R_0) < T_1) ? 1 : 0 + (abs(L_1 - L_0) < T_2) ? 1 : 0 +$$

$$(abs(L_2 - L_1) < T_2) ? 1 : 0 + (abs(R_1 - R_0) < T_2) ? 1 : 0 +$$

$$(abs(R_2 - R_1) < T_2) ? 1 : 0$$

其中,  $T_1$  的取值与  $\alpha$  有关;  $T_2$  的取值与  $\beta$  有关。

如果  $flatness \geq Th_1$ , 则  $B_s=2$ ; 如果  $flatness \geq Th_2$ , 且  $flatness < Th_1$ , 则  $B_s=1$ ; 如果  $flatness < Th_2$ , 则  $B_s=0$ 。 $Th_1$  和  $Th_2$  是固定值,  $Th_1$  可取 3, 4 或 5,  $Th_2$  可取 2 或 1。

如果当前  $8 \times 8$  亮度块第 1 行或第 1 列的  $B_s=0$ , 则当前整个  $8 \times 8$  亮度块的边界滤波强度为 0, 不需要滤波, 直接跳到下一个  $8 \times 8$  亮度块的起始位置, 重复上述步骤。如果当前  $8 \times 8$  亮度块第 1 行或第 1 列的  $B_s$  为 1 或 2, 则当前整个  $8 \times 8$  亮度块的边界滤波强度为 1 或 2。

### 3.3 新算法的滤波过程

当  $B_s=2$  时, 仅修改  $L_0, R_0$  的值, 即

$$L_0^- = (L_1 + (L_0 \ll 1) + R_0 + 2) \gg 2$$

$$R_0^- = (R_1 + (R_0 \ll 1) + L_0 + 2) \gg 2$$

当  $B_s=1$  时, 仅修改  $L_0, R_0$  的值, 此时滤波方式发生改变, 即

$$\delta = Clip3(-C, C, (((R_0 - L_0) \times 3 + (L_1 - R_1) + 4) \gg 3))$$

$$L_0^- = Clip1(L_0 + \delta)$$

$$R_0^- = Clip1(R_0 - \delta)$$

当  $B_s=0$  时, 不进行滤波。

$L0^*$ ,  $R0^*$  是对  $L0$ ,  $R0$  滤波后的值。色度边界和亮度边界采用相同的滤波方式。

## 4 测试与结果分析

本实验在配置为 P4 1.7 GHz CPU、512 MB 内存、Win XP 操作系统的 PC 上进行，编译环境为 VC++6.0。实验时取  $T1 = (\alpha \gg 3) + 2$ ,  $T2 = (\beta + 2) / 4$ ,  $Th1 = 4$ ,  $Th2 = 2$ 。

### 4.1 性能测试

在 AVS-P2 的基准档次下，采用 IBBPBBP... 的编码结构，I 帧周期为 17，即  $Intraperiod=17$ 。对每个序列编码 100 帧，进行客观性能测试。表 1 给出了对 cif 格式图像在帧编码的情况下进行处理时，本文提出的新算法相对 RM52j\_r1 anchor 算法的性能差异。表 2 给出了对 576i 格式的图像在 PAFF(帧场自适应)的情况下进行处理时，本文算法相对 RM52j\_r1 anchor 算法的性能差异。

表 1 对 cif 格式图像的测试结果

Cif	信噪比的降低量/dB	平均滤波时间减少幅度/(%)
foreman_cif	0.05	42.34
mobile_cif	0.01	26.19
news_cif	0.04	36.57
paris_cif	0.03	49.47
tempete_cif	0.02	26.88
average	0.03	36.29

表 2 对 576i 格式图像的测试结果

576i PAFF	信噪比的降低量/dB	平均滤波时间减少幅度/(%)
basketball	0.02	22.92
flowergarden	0.05	34.17
horseriding	0.02	46.71
average	0.03	34.60

由表 1 和表 2 可以看出，本文提出的环路滤波算法对多数测试序列的环路滤波时间平均减少 30% 以上，且信噪比平均降低 0.03dB。

### 4.2 算法效果比较

算法效果如图 4~图 7 所示，其中，图 4 和图 6 是经 RM52j\_r1 源程序编码重建后的图像；图 5 和图 7 是使用本文提出的新算法得到的重建图像。



图 4 foreman\_cif 原算法得到的重建图像



图 5 foreman\_cif 新算法得到的重建图像



图 6 mobile\_cif 原算法得到的重建图像



图 7 mobile\_cif 新算法得到的重建图像

## 5 结束语

AVS 基准档次<sup>[4]</sup>的环路滤波利用宏块类型、宏块中  $8 \times 8$  亮度块的运动矢量计算滤波强度  $B_s$ 。它在修改边界两边的像素值时，要对对应像素之差进行阈值判断，并决定是否进行滤波。本算法根据图像的相关性，先利用当前  $8 \times 8$  亮度块边界两边的 6 个像素值得得边界平滑度，然后根据边界平滑度求得  $B_s$ ，再修改相应像素值，在修改像素值的过程中去除了大量条件判断，且只修改边界两边各一个像素值，因此，减小了计算量和环路滤波耗时。

### 参考文献

- [1] 高文, 王强, 马思伟. AVS 数字音视频编解码标准[J]. 中兴通讯技术, 2006, 12(3): 6-9.
- [2] ITU. T Recommendation and Final Draft International Standard of Joint Video Specification[S]. Tech. Rec.: H.264 ISO/IEC 14496-10 AVC, 2003.
- [3] 生滨, 高文, 吴迪. AVS 环路滤波器设计及实现[J]. 高技术通讯, 2006, 16(7): 666-670.
- [4] AVS 工作组. GB/T 20090.2-2006 信息技术先进音视频编码第 2 部分: 视频[S]. 2006.

编辑 陈 晖