

# 宏块级帧场自适应的去块效应滤波模块分析与优化

王军, 周开伦, 林涛

(同济大学 超大规模集成电路研究所, 上海 200331)

**摘要:**对支持宏块级帧场自适应的 H.264 去块效应滤波过程进行深入分析, 利用相邻 4×4 像素块间数据的依赖关系合理组织数据存储顺序, 提出了一种针对 H.264/AVC 标准的高性能、低复杂度的去块效应滤波系统的 VLSI 结构。

**关键词:**H.264/AVC 解码器; 宏块级帧场自适应; 去块滤波

中图分类号: TN911.73

文献标识码: A

## Analysis and optimization of deblocking filter in H.264/AVC MBAFF decoder

WANG Jun, ZHOU Kai Lun, LIN Tao

(Very Large Scale Integrated Circuits Laboratory, Tongji University, Shanghai 200092, China)

**Abstract:** In this paper, a high efficiency and low complexity deblocking filter architecture for H.264/AVC MBAFF decoder is proposed. Making good use of data dependence between neighboring 4×4 blocks, the design reduces the requirement of on-chip SRAM bandwidth and increases the throughput of the filter processing.

**Key words:** H.264/AVC decoder; MBAFF; deblocking filter

H.264/AVC 是由 ITU-T 视频编码专家组和 ISO/IEC 运动图像专家组联合制定的面向未来 IP 和无线环境的最新视频编码标准<sup>[1]</sup>。H.264/AVC 采用基于块的 DCT 变换、量化、运动补偿, 这些技术都会不可避免地在块边界引入块效应, 严重影响图像的主观质量, 因此, H.264/AVC 标准采用去块效应环路滤波(简称环路滤波)来去除块效应, 以提高图像的主观质量和编码效率。

但在大幅度改善视频图像质量和压缩效率的同时, 去块效应滤波系统也引入了极高的计算复杂度。其运算量大约占解码器计算总量的 1/3<sup>[2]</sup>, 使之成为解码器设计中的瓶颈之一。去块效应滤波系统对图像中每一个 4×4 块的内容都要进行横向和纵向 2 次滤波, 涉及到大量的存储器读写, 因此, 去块效应滤波系统的 VLSI 结构设计和中间数据缓冲器的选择成了硬件结构设计上的重大难题。

参考文献[2]是基于帧上的滤波结构, 参考文献[3-5]中列举了几种已经提出的硬件加速器结构, 这 2 种结构都是以宏块滤波为基础的。很明显, 基于帧上的滤波结构需要一帧的缓冲器和更长的系统延时。为了克服这个困难, 参考文献[5]引入了一个双口 SRAM 来同时对滤波数据进行读和写, 但是内部结构比较复杂而且处理延时仍然较大。

从现有的文献看, 讨论 H.264 去块效应算法和硬件实现的文章不少, 但对支持宏块级帧场自适应去块效应的文献几乎还是空白。本文分析并优化了支持宏块级帧场自适应的去块效应的存储器结构, 并给出了硬件实现。

### 1 算法

#### 1.1 去块效应滤波

去块滤波在整个视频解码过程中位于重建之后, 对每个解码完成的宏块按光栅扫描顺序进行滤波, 图像边界不用滤波。滤波是基于宏块的, 由于 H.264 中的整数变换是基于 4×4 点的, 那么块滤波系统在滤波的时候也是以 4×4 点的块为单元处理。先对垂直边界进行水平滤波, 再对水平边界进行垂直滤波。对宏块的 2 个方向滤波都完成后才能进行后面宏块的滤波。对图像中宏块的滤波按 raster 扫描方式进行。对帧场自适应编码帧, 它们在垂直方向上相邻的宏块对放在一起, 则滤波顺序按宏块对进行, 即在帧中对宏块对按 raster 扫描方式进行, 对每个宏块对先进行顶部宏块的滤波<sup>[2]</sup>。图 1 虚线为需要滤波的边界。对每个亮度宏块, 先滤波宏块最左边的边界(图 1 中序号 1), 然后依次从左到右宏块内 3 个垂直边界(图 1 中的序号 2~4)。类似地, 对水平边界先滤波宏块顶部的边界(图 1 中

# 图形、图像与多媒体

Image Processing and Multimedia Technology

的序号 5)，然后依次从上到下宏块内 3 个水平边界（图 1 中的序号 6~8）。色度滤波次序类似。

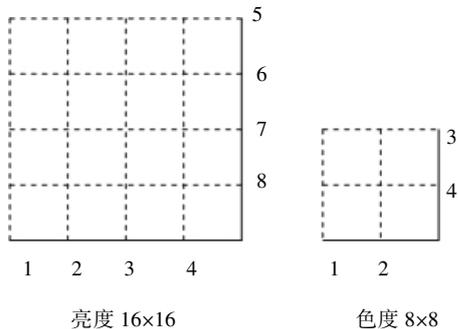


图 1 宏块滤波顺序

H.264/AVC 的自适应环路滤波包括如下几个部分：  
 (1)边界滤波像素的输入输出；(2)边界滤波强度的计算，算法如图 2 所示，必须先分清图像的真实边界和虚假边界，对真实边界要加以保持，而对虚假边界进行平滑滤波；(3)计算边界阈值  $\alpha$ 、 $\beta$ 、 $c_0$ 。最后对滤波强度不为 0 的虚假边界进行平滑滤波。整个处理过程如图 3 所示。

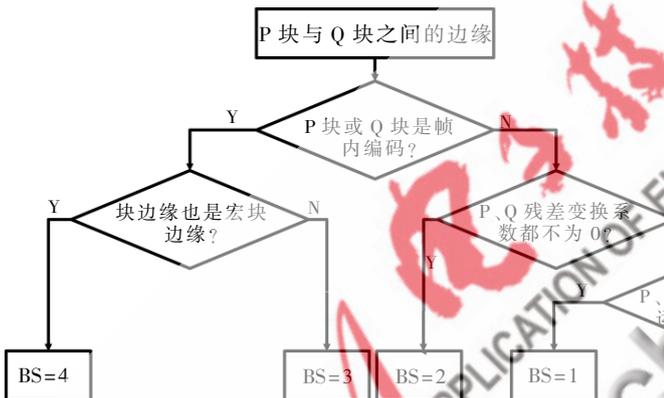


图 2 滤波强度计算算法

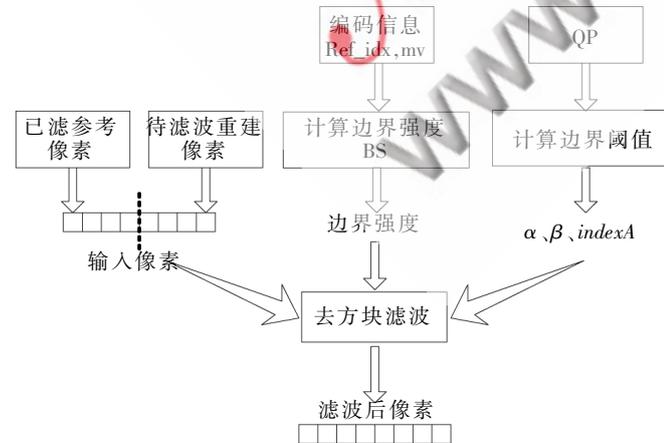


图 3 去块效应滤波过程

## 1.2 宏块级帧场自适应

视频帧可以分为 2 种类型：连续帧或隔行视频帧<sup>[9]</sup>。通

常为了减少大面积的闪烁现象，把一帧分为 2 个隔行的场。这时场内相邻行的时间相关性较强，帧内相邻行的空间相关性较强。在有物体移动或摄像机镜头移动的隔行扫描帧中，与逐行扫描帧相比，临近的两行的统计依赖性下降。在这种情况下，也许对每个场单独进行压缩的效率更高。为此 H.264 的设计中允许编码器在对一帧进行编码时做如下决定：

- (1)2 个场结合在一起，对其按一个独立的帧进行编码（帧模式）。
- (2)2 个场结合，对其按单独的场独立进行编码（场模式）。
- (3)将 2 个场结合在一起，并将其作为一个单独的帧压缩，但编码时需要首先将垂直相邻的 2 个宏块分成 2 场或 2 帧中的一对宏块。

对一个序列中的每一帧都可以自适应地选择这 3 种选项中的一种。H.264 采用宏块自适应帧场编码（MB-AFF）模式中，帧场编码的选择在宏块级中指定。且当前片通常由 16 亮度像素宽和 32 亮度像素高的单元组成，并以宏块对的形式编码，如图 4 所示。编码器可按 2 个帧宏块或者 2 个场宏块来编码每个宏块对<sup>[7]</sup>。

由于宏块级帧场自适应是按照垂直宏块对的形式进行编码，因此其解码顺序与普通帧/场中宏块的解码顺序有很大的不同，如图 5、图 6 所示。在非帧场自适应模式下，解码器按照水平扫描的顺序依次解码宏块；而在宏块级帧场自适应时，解码顺序演变为锯齿形顺序。

当采用宏块级自适应编码时，相邻宏块位置的获取就比较复杂了，在以宏块对为单位对图像进行解码时，除了需要判断当前宏块是上宏块还是下宏块，

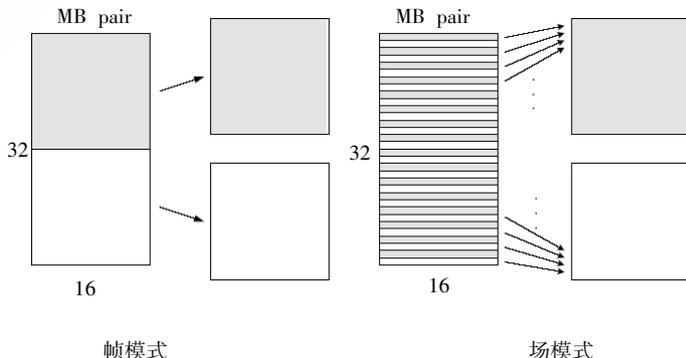


图 4 宏块自适应帧场编码

还要判断当前宏块对和参考宏块对的帧场属性，因为对于计算所需的像素位置是不同的。宏块级帧场自适应取值方法，具体可参见 H.264 协议<sup>[1]</sup>。相邻宏块对间的关系如图 7 所示。

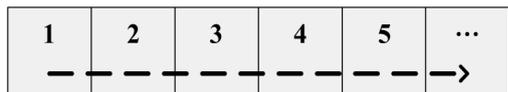


图5 非 MBAFF 码流解码顺序

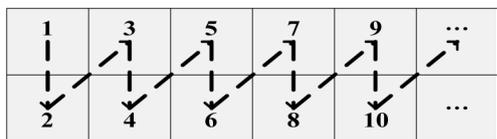


图6 MBAFF 码流解码顺序

mbAddrD	mbAddrB	mbAddrC
mbAddrD+1	mbAddrB+1	mbAddrC+1
mbAddrA	CurrMbAddr or	
mbAddrA+1	CurrMbAddr	

图7 MBAFF 时相邻宏块间的关系

## 2 去块效应滤波系统结构

### 2.1 滤波器结构

针对影响去块效应滤波系统速度的因素,提出了一种高效的滤波结构,如图8所示。该结构利用了数据重用机制和并行计算来确保达到速度要求。整个结构由中央滤波处理单元、阈值查询单元、数据输入输出单元和同步控制单元组成。

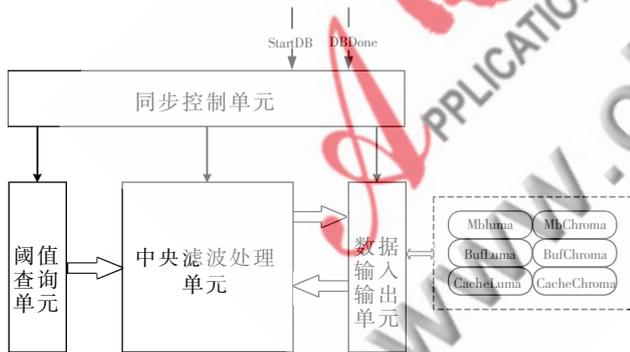


图8 滤波器结构

中央滤波处理单元是去块效应滤波系统的核心部分。考虑到 SRAM 存取数据的时间耗费,不能采用参考软件中的逐个滤波集合依次处理的方法,而一定要采用并行处理的方式进行。但出于芯片面积和功耗的考虑,本设计中采用以一个块边界(Block Edge)为单元的滤波处理方案。

阈值查询单元负责接收上一模块传递进来的  $indexA$ 、 $indexB$ , 然后实现查表功能,输出阈值  $\alpha$ 、 $\beta$ 、 $c_0$  以及滤波强度  $Strength$ 。

数据输入输出单元负责给中央滤波单元传送数据和存储数据。在新的视频标准中,进行水平滤波(垂直边界滤波)和垂直滤波(水平边界滤波)所用的数据格式有所不同,水平滤波取2个4×4块的行数据进行滤波,而垂直滤波是对2个块的列数据进行滤波。由于存储数据时,都是一行4个数据进行存储的,因此在进行水平滤波过程时,每个块的数据都要经过  $T$  (矩阵转换单元)进行一次矩阵的行列转换,垂直滤波准备数据。同时由于 MBAFF 模式下参考像素的获取有了更大的灵活性,因此片内数据的读写控制需要更加周密的设计。

同步控制单元用一个有限状态机来实现,控制整个去块效应滤波的数据处理过程包括接收启动信号、宏块参数信息的读取、阈值查询、滤波像素值的输入、滤波操作和滤波后数据处理等;控制中央滤波单元的流水线操作,完成宏块图像数据去块效应滤波功能。

### 2.2 数据流控制

去块效应滤波在大幅度改善视频图像质量和压缩效率的同时,增加了对外部存储器的访问频率。而频繁的存储器存取会增大整个芯片的功耗,所以设计出合理的存储器访问策略对本设计尤为重要。

由于滤波过程中不仅需要当前宏块重建之后的像素数据,还需要相邻宏块的已经滤波完的像素。对于1080像素这样解析度的图像来说,在 MBAFF 模式下,需要存储3个Block行的数据( $1920 \times 4 \times 12 \times 3 = 270$  KB),这么多的数据用片内 SRAM 来存储肯定是不现实的,而 DRAM 具有较低的成本和不错的访存速度,它被广泛用来存储整个图像帧的数据。同时在片内开辟一块很小的 SRAM (LineCache) 用来和 DRAM 交互,增加数据的重复使用率和访存效率,在解码完若干个宏块后进行更新。这样可以提高系统的整体性能。

作为沟通寄存器和 DRAM 数据流向的中间层的 SRAM 在设计中显得尤为重要。片内存储器分为单口 SRAM 和 Register Files 两部分。本设计采用6块片内 SRAM,如图8所示。下面以亮度信号为例,具体介绍 SRAM 的用途。MbLuma 用来存放运动补偿单元处理后的宏块数据,由去块效应单元处理完毕后的数据也将放在这里,然后写入到 SDRAM 中。当宏块的第一条边界两边像素点滤波完成后,右边的4个4×4块被存入寄存器中,这部分数据在对下一边界滤波时可被重用,这样就减少了存储器读取的频率,从而降低处理的延迟。重复上述过程,直到最后一条垂直边界滤波完成之后,将垂直边界滤波后的数据存入 BufLuma 中。BufLuma 作为片内的存储器用来缓冲进行完纵向滤波后的数据。之前首先进行了一次变换(将横向排列的像素变为纵向),以满足接下来将要进行的横向滤波的要求。在进行垂直边界滤波时,数据主要由 MbLuma 读出,处理后主要写入 BufLuma。接下来再进行水平边界滤波时,数据主要由 BufLuma 读出,处理后主要写入

## 图形、图像与多媒体

Image Processing and Multimedia Technology

MbLuma, 然后写入 SDRAM。CacheLuma 中存放有去块效应需要暂存的参考像素数据, 主要包括当前宏块左侧和上边的若干个 Block 像素。

MBAFF 下宏块边界参考像素数据存取要复杂很多。MBAFF 下都是按照宏块对的顺序滤波的, 宏块对内部的帧场属性是一致的, 但是不同宏块对之间可能有不同的帧场属性, 对于宏块内部边界的滤波与非 MBAFF 时边界滤波没有区别, 但是对于宏块边界的滤波, 需要根据当前宏块在宏块对中的位置, 以及当前宏块和参考宏块的帧场属性来综合判断, 确定滤波所需的滤波强度以及滤波参考像素的取值。对于水平边界滤波也就形成了 8 种可能的组合, 如表 1 所示。

表 1 MBAFF 模式下组合方式

CurrentMB	RefMB	ID.	
Frame	Top	Frame	1
		Field	2
	Bottom	Frame	3
		Field	4
Field	Top	Frame	5
		Field	6
	Bottom	Frame	7
		Field	8

对于 ID.1、3、6、8 的组合, 由于当前宏块和参考宏块帧场属性相同, 取值和非 MBAFF 模式下的一样; 而对于当前宏块和参考宏块具有不同的帧场属性, 情况就要复杂很多。为了更清楚地说明对宏块级帧场自适应滤波, 下面举例分析。当前宏块为宏块对中的顶宏块, 是场宏块, 现在做水平边界的滤波, 它需要参考其上面宏块对的信息, 其参考宏块为帧宏块对, 那么它需要的参考 Block 就是图 9 左边中浅灰 Block; 假设当前宏块为宏块对中的顶宏块, 是帧宏块, 现在做水平边界的滤波, 它需要参考其上面宏块对的信息, 其参考宏块为场宏块对, 那么它需要的参考 Block 为图 9 中右边浅灰 Block。

更多的宏块级帧场自适应在计算预测值时选取哪些值, 具体可参见 H. 264 协议<sup>[1]</sup>。

本文对 H.264/AVC 标准中 MBAFF 模式下的去块效应滤波过程进行了分析, 利用数据重用机制和并行计算来加速滤波的过程, 提高了整个滤波系统的速度, 对 H.264/AVC 的编解码有较好的实现。

## 参考文献

- [1] ITU-T. Advanced video coding for generic audio visual services. ITU-T Recommendation H.264 | ISO/IEC 14496-10 (MPEG-4 AVC), 2005.
- [2] SIMA M, ZHOU Yuan Hua, ZHANG Wei. An efficient architecture for adaptive deblocking filter of H.264/AVC video coding. 46

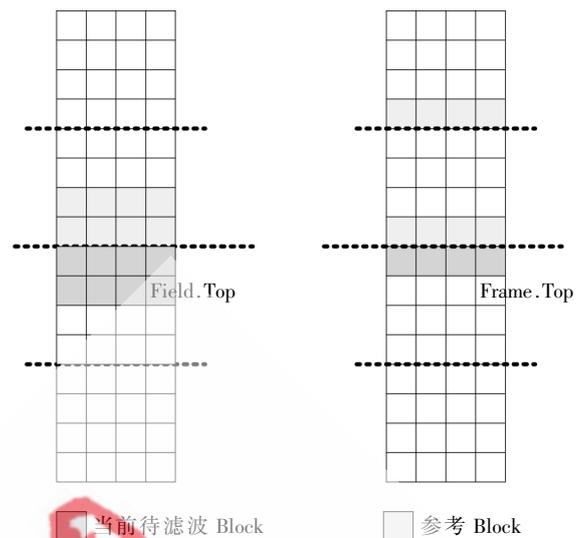


图 9 MBAFF 水平滤波取值

- [3] HUANG Y W, CHEN T W, HSIEH B Y, et al. Architecture design for deblocking filter in H.264/JVT/AVC. IEEE International Conference on Multimedia and Expo, 2003.
- [4] SHENG B, GAO W, WU D. An implemented architecture of deblocking filter for H.264/AVC. IEEE International Conference on Image Processing, 2004, 1:665-668.
- [5] CHENG Chao Chung, CHANG Tian Sheuan. An hardware efficient deblocking filter for H.264/AVC. Proc. of International Conference on Consumer Electronics (ICCE) 2005. [s.l.]: IEEE Press, 2005.
- [6] 冯晓茹, 陈晨, 杨海池, 等. 支持宏块级帧场自适应的 H.264 解码器帧内模块 FPGA 实现[J]. 空间电子技术, 2008(4).
- [7] 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.
- [8] KIM J, NA S, KYUNG C. A low-power deblocking filter architecture for H.264 advanced video coding. Very Large Scale Integration, 2007. VLSI-SoC 2007. IFIP International Conference on Volume, Issue, 15-17 Oct. 2007:190-193 Digital Object Identifier 10.1109/VLSISOC.2007.4402496
- [9] LIST P, JOCH A, LAINEMA J, et al. Adaptive deblocking filter [J]. IEEE Trans. on Circuit Syst. Video Technol, 2003, 13(7).
- [10] JVT H.264/AVC Reference Software JM 9.3.

(收稿日期: 2009-07-13)

(上接第 42 页)

- [8] 刘春庆. 空分装置计算机控制系统设计实现[D]. 山东科技大学工程硕士学位论文, 2005.05.
- [9] 邓李. ControlLogix 系统实用手册[M]. 北京: 机械工业出版社, 2008.
- [10] 黄宏中. 浅谈空分装置自动化[J]. 自控工程设计. 2000(5): 29-31.
- [11] 皮壮行. 可编程控制器的系统设计与应用实例[M]. 北京: 机械工业出版社, 2000.

(收稿日期: 2009-06-23)

《微型机与应用》2009 年第 20 期