

H. 264 /AVC 中 1/4 精度内插算法的 硬件设计与实现

赵子梁¹⁾ 郑世宝²⁾

¹⁾(上海交通大学电子工程系图像通信与信息处理研究所, 上海 200240)

²⁾(上海交通大学上海市数字媒体处理与传输重点实验室, 上海 200240)

摘要 为了快速地进行 H. 264/AVC 中 1/4 精度像素内插, 提出并实现了一种适用于 H. 264 中 1/4 像素精度的内插算法的硬件设计。其中对亮度分量, 设计了一种将 2 维滤波转换为 1 维滤波, 4 个滤波器并行处理, 同时采用流水线技术和用移位加代替乘法运算的硬件架构; 对色度分量, 用移位加代替乘法运算进行 1/8 像素精度的内插。实践表明, 此架构可满足标清及高清要求, 且速度快, 面积小。

关键词 1/4 精度像素 硬件架构 滤波器 移位加

中图分类号: TN919.81 文献标识码: A 文章编号: 1006-8961(2007)10-1740-05

Design and Implementation of Quarter-pixel Interpolation in H. 264 /AVC

ZHAO Zi-liang¹⁾, ZHENG Shi-bao²⁾

¹⁾(Institute of Image Communication and Information Processing, Department of Electronic Engineering,
Shanghai Jiaotong University, Shanghai 200240)

²⁾(Shanghai Key Laboratory of Digital Media Processing and Transmissions, Shanghai Jiaotong University, Shanghai 200240)

Abstract In this paper, a hardware architecture for quarter-pixel interpolation in H. 264 is proposed. As for the luminance component, we adopt a design which is using four parallel filters which can turn 2-D filter into 1-D filter and using pipeline technique and shift&add instead of multiplication in the mean time. While for the chrominance component, we use a design comprising of only shift&add to calculate the eight-pixel interpolation. The architecture proposed in this paper can meet the requirement of SD and HD and it works fast and consumes less resources.

Keywords quarter-pixel interpolation, hardware architecture, filter, shift and add

1 引言

H. 264^[1]是 ITU-T 的视频编码专家组 (VCEG) 和 ISO/IEC 的活动图像编码专家组 (MPEG) 的联合视频组 (joint video team, JVT) 于 2004 年开发的一个先进的数字视频编码标准, 它既保留了以往压缩技术的优点和精华, 又具有其他压缩技术无法比拟的许多优点。

与以往标准不同, H. 264/AVC 采用 1/4 像素精度的内插算法 (亮度 1/4 精度, 色度 1/8 精度), 并

用 6 抽头滤波器实现所需位置像素的内插值, 其在提高精度的同时, 增加了实现的复杂度。而在解码端, 分数像素内插过程的时间复杂度平均占解码器的 26% ~ 30%^[2]。因此, 有必要设计一个运行速度快, 且相对消耗资源较少的内插器。文献[3]由于采用两个 1 维滤波, 并对数据进行了重用, 从而提高了数据的使用效率, 但每一行列转换都要消耗额外的 18 个时钟周期, 这不仅增加了总耗时, 也增加了控制复杂度; 对于 1/8 色度的内插, 其是采用组合逻辑一步完成, 这样不但增加了耗时, 而且也消耗了大量资源。文献[2]采用有损内插, 虽然提高了速度

收稿日期: 2007-06-15; 改回日期: 2007-06-25

第一作者简介: 赵子梁 (1982 -), 男, 硕士研究生。研究方向为视频处理及集成电路设计。E-mail: zhaoziliang@sjtu.edu.cn

和降低了复杂度,但应用于视频通信时,会使飘移误差越来越大,有可能会影响视频质量。本文采用流水线加并行处理技术来将 2 维滤波转换为两个 1 维滤波,并在输入数据的同时,进行相应位置的内插运算,以节省内插时间。

2 H.264/AVC 内插算法^[1]

亮度的分数像素样本如图 1 所示。

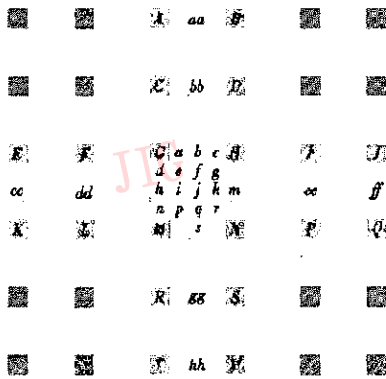


图 1 H.264 中各整数像素和分数像素的位置(亮度内插)^[1]

Fig.1 Interpolation position in H.264(luminance)

在图 1 的方块中,用大写字母表示整数像素的亮度样本,小写字母表示分数像素的亮度样本。半像素处的亮度样本预测值使用 6 抽头滤波器来获得,其抽头系数为 1, -5, 20, 20, -5, 1。而 1/4 像素处的预测值则使用半像素和整数像素的平均值来得到。每个位置的计算过程如下:

(1) 半像素位置样本 b 和 h 的计算过程如下:

先用 6 抽头滤波器计算中间变量

$$b_1 = E - 5 \times F + 20 \times G + 20 \times H - 5 \times I + J \quad (1)$$

$$h_1 = A - 5 \times C + 20 \times G + 20 \times M - 5 \times R + T \quad (2)$$

最终的预测值 b 和 h 使用下式得到:

$$b = \text{Clip1}((b_1 + 16) \gg 5) \quad (3)$$

$$h = \text{Clip1}((h_1 + 16) \gg 5) \quad (4)$$

(2) 半像素位置样本 j 可通过两次使用 6 抽头滤波器得到,即

$$j_1 = cc - 5 \times dd + 20 \times h_1 + 20 \times m_1 - 5 \times ee + ff \quad (5)$$

$$j_2 = aa - 5 \times bb + 20 \times b_1 + 20 \times s_1 - 5 \times gg + hh \quad (6)$$

两个式子任取其一,因它们的结果是相同的,然后即可得到以下的最终预测值:

$$j = \text{Clip1}((j_1 + 512) \gg 10) \quad (7)$$

(3) 1/4 像素处的样点 a, c, d, n, f, i, k 和 g 可通过与之最近的整数和半像素样点进行平均得到,即

$$a = (G + b + 1) \gg 1, f = (b + j + 1) \gg 1 \quad (8)$$

$$c = (H + b + 1) \gg 1, i = (h + j + 1) \gg 1 \quad (9)$$

$$d = (G + h + 1) \gg 1, k = (j + m + 1) \gg 1 \quad (10)$$

$$n = (M + h + 1) \gg 1, q = (j + s + 1) \gg 1 \quad (11)$$

(4) 1/4 像素处的样点 e, g, p 和 r 可通过将与之最近的半像素样点进行平均得到,即

$$e = (h + b + 1) \gg 1, p = (h + s + 1) \gg 1 \quad (12)$$

$$g = (m + b + 1) \gg 1, r = (m + s + 1) \gg 1 \quad (13)$$

色度的分数像素样本内插如图 2 所示。

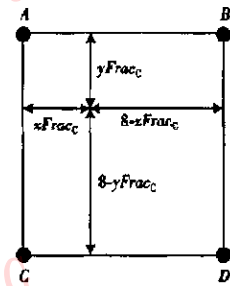


图 2 色度整数像素位置(色度内插)^[1]

Fig.2 Integer position of chroma pixels(chrominance)

在图 2 中, $xFrac_C$ 和 $yFrac_C$ (下角 C 代表 chroma,下同) 分别代表以 1/8 像素为单位色度运动矢量的分数偏移,其中

$$xFrac_C = mvCLX[0] \& 7, yFrac_C = mvCLX[1] \& 7 \quad (14)$$

$mvCLX$ 可看成存放运动矢量小数部分的数组。

对于给定的整数像素样本 A, B, C, D , 其分数精度的色度样本预测值 $predPartLX_C[X_C, Y_C]$ 可通过以下计算过程得到:

$$\begin{aligned} predPartLX_C[X_C, Y_C] = & ((8 - xFrac_C) \times (8 - yFrac_C) \times \\ & A + xFrac_C \times (8 - yFrac_C) \times B + \\ & (8 - xFrac_C) \times yFrac_C \times C + \\ & xFrac_C \times yFrac_C \times D + 32) \gg 6 \end{aligned} \quad (15)$$

3 硬件架构设计与实现

为方便说明,下面对内插位置,用 0~15 代替图 1 中的英文字母 G, a, b, c 等(顺序从左到右,从上到下)。由于 H.264 宏块分块为 $16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4$ 等 7 种分块之一,其中 4×4 为最小情况,而其他大小的块都可拆为几个 4×4 块,所以在设计中,采用基于 4×4 块的内插设计,即所有不

同大小的分块都拆成 4×4 块来进行内插。

通过对每个位置内插所需的数据量进行的分析发现,可以分为以下几种情况:①内插位置为 0,则所需数据为 4×4;②内插位置为 1,2,3,则所需数据为 9×4;③内插位置为 4,8,12,则所需数据为 4×9;④其他内插位置,所需的数据为 9×9。再通过对每个位置内插所用的方法分析,可以得到以下结论:内插主要采用 6 抽头滤波器进行滤波及取均值这两种方法的有机组合来完成。

由以上两个结论可得:内插所需数据量最大为 9×9,通过滤波器与取平均的组合即可得到所需位置的插值。基于这个思路,设计了一个架构,其框图如图 3 所示。

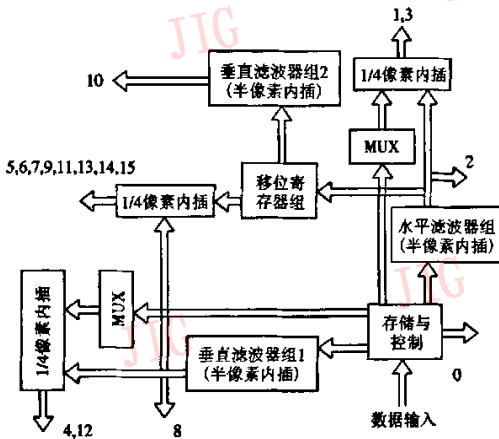


图 3 整体框架
Fig. 3 Architecture

图中 0~15 表示对应位置内插后的数据输出。

半像素及 1/4 像素内插模块如图 4 所示。举个例子,对于内插位置 2,要完成同一行内插需要 9 个点,即图 1 中的 E,F,G,H,I,J 及 J 后面的 3 个点(同

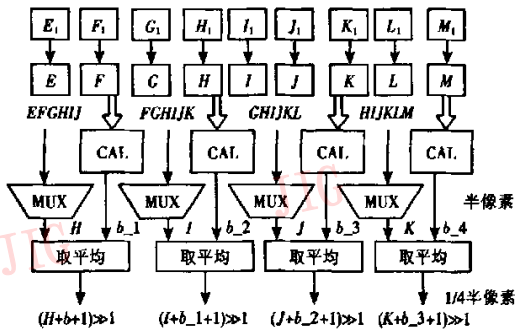


图 4 半像素及 1/4 像素内插
Fig. 4 Half and quarter pixel interpolation

一行中,设为 K,L,M)。由于设计采用移位寄存器,因此传送给 CAL 模块的数据无需经过选择即可直接进行计算(后面几个周期下一行的数据会自动移下来,即图 4 中 E₁,F₁,G₁,H₁,I₁,J₁,K₁,L₁,M₁,进行下一行的内插)。传送给 CAL(4 个 CAL 模块构成一个滤波器组)的数据依次为 EFGHIJ,FGHIJK,GHIJKL,HIJKLM(这里 EFGHIJ 表示 E,F,G,H,I,6 个数据,以下同),通过计算即可得到位置 2 上的内插值 b, b₋₁,b₋₂,b₋₃。半像素位置 8 的内插方法相同,只是计算所需的数据是列方向上的 9 个值。而对于 1/4 像素内插,例如内插位置 3,只需将内插位置 2 计算后的值与经过 MUX 的选择后的值取平均即可,MUX 的选择输出为 H,I,J,K。对于其他 1/4 像素,位置 1,4,12 的内插,方法相同,只是数据不同。

CAL(滤波器)采用两级流水设计(如图 5 所示),即先将乘法运算转换成移位加,然后通过加法器,再取整并移位,最后经 Clip1 模块计算,该滤波器框图如图 5 所示(其中模块 PE 用于计算中间值,如图 6 所示)。

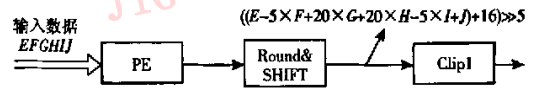


图 5 滤波器设计
Fig. 5 Filter design

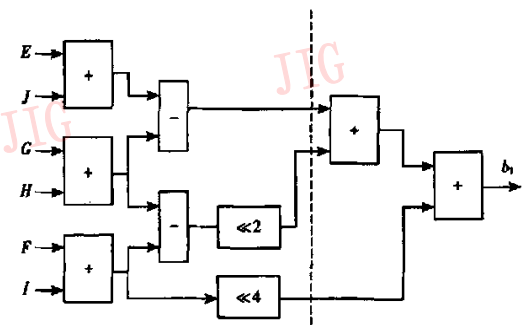


图 6 亮度 PE 模块(虚线为流水线分割)
Fig. 6 PE module for luminance (pipeline)

对于半像素位置 10,可先计算半像素位置 2 的中间值,如图 1 中的 aa,bb,b,s,gg,hh,先存入另一个移位寄存器(5×4,最后一个数据 hh 计算内插后存入,第 1 个数据 aa 移出,所以只需 5×4 而非 6×4),再 6 个一组送入垂直滤波器组 2 计算,其所得到的式(7)中 j 的值,即为位置 10 的内插值,垂直滤波器组 2 也由 4 个 CAL 模块构成,只是输入数据位宽

不同(如图 7 所示)。

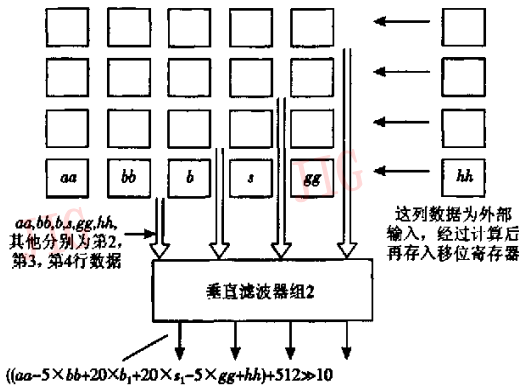


图 7 半像素位置 10 内插

Fig. 7 Position 10 interpolation

对于位置为 5,7,13,15 的 1/4 像素内插来说,其可先通过计算半像素内插位置为 2 和 8 的值,再通过 1/4 像素内插计算平均值得到。对于位置为 6,9,11,14 的 1/4 像素内插,可先通过计算半像素内插位置 10 的值及半像素内插位置 2 或 8 的值,再通过 1/4 像素内插得到。对于图 3 中内插位置为 0 的值,可直接将结果输出,无需计算。

对于色度,所用公式见式(15),经过变换,原式即改为

$$(8 - yFrac_c)[(8 - xFrac_c) \times A + xFrac_c \times B] + yFrac_c[(8 - xFrac_c) \times C + xFrac_c \times D] \quad (16)$$

可得通式 $(8 - P) \times X + P \times Y$,先求 $X = A, Y = B$ 时的通式的值 E ,再求 $X = C, Y = D$ 时的通式的值 F ,最后求 $X = E, Y = F$ 时的通式的值,就可得到式(16)的结果。

设计的模块结构如图 8、图 9 所示。

CAL₁ 和 CAL₂ 模块将乘法转换为移位加,其中

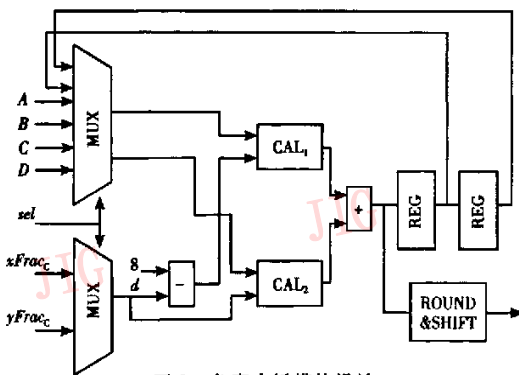


图 8 色度内插模块设计

Fig. 8 Module for chroma interpolation

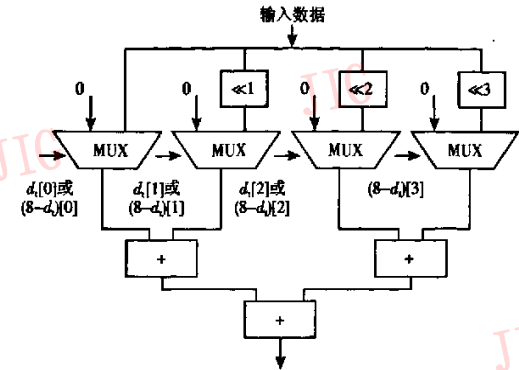


图 9 CAL(1,2) 模块设计

Fig. 9 Module CAL(1,2)

输入为数据 d_i (d_i 为一中间变量,其等于 $xFrac_c$ 或 $yFrac_c$) 的 CAL2 模块,只要前 3 个 MUX 即可,因为 d_i 范围为 0~7;而输入为 $(8 - d_i)$ 的 CAL1 模块则要所有 4 个 MUX,因为 $(8 - d_i)$ 的范围为 1~8。

4 性能分析

以上架构采用 Verilog 实现,并用 Modelsim6.0 进行仿真,图 10 给出了 1/4 像素内插单元部分仿真波形。同时对该架构采用了 Synopsys 公司的 Design Compiler 进行了综合,该架构采用 Smic 0.18 μ m 工艺,最高频率可达 125MHz,面积为 0.43mm²。此架构处理一个宏块最多需要 560 个时钟周期,最少 64 个时钟周期。经过分析发现,这个架构可以处理 16 帧/s,720 x 480 大小 4:2:0 格式的标清图像。与文献[4]中结果相比(频率为 99.198MHz,处理一个宏块需要 787 个时钟周期),频率提高了约 26%,处理时间减少了 227 个时钟周期。文献[5]综合时,未包括存储单元,其最高频率为 120MHz,从给出的仿真波形看,数据输入一个时钟周期后就得到内插值。其可在一个时钟周期内完成滤波和取平均,但由于有可能这段会产生关键路径,因此不利于提

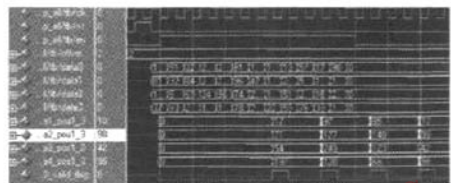


图 10 1/4 像素位置 3 仿真波形

Fig. 10 Quarter-pixel position 3 simulation

高速度。此外,由于最坏情况是所有宏块内插位置都是 6 或 9 或 11 或 14(这种情况下需要耗费最多的时钟周期),而通过对图像进行统计后发现,这种情况是小概率事件,几乎不可能发生,所以经过分析计算这个架构能达到 1080i@30fps 要求。对于图像大小为 352×288 ,传输速度为 15fps 的 3G 手机图像来说,其所需频率仅为 3.320MHz,因此,此架构很适合 3G 手机使用。

5 结 论

本文提出了一种 2 维滤波转 1 维滤波,4 个滤波器并行处理,并采用流水线技术和用移位加代替乘法运算的 H.264 1/4 像素精度的亮度内插器,同时设计了 1/8 精度的色度内插器。这样通过输入内插位置及其所需数据,这两个内插器都能在一定时间内完成内插运算。本设计充分考虑了速度和面积的平衡,不仅达到了处理标清和高清的要求,同时很适合 3G 手机使用。

参考文献 (References)

- 1 Joint Video Team (JVT) of ISO/IEC & ITU-T VCEG. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification [S]. H. 264/ISO/IEC 14496-10 AVC, ITU-T, Mar, 2002.
- 2 Fang Yan-long, Zhou Jun. Fast algorithm for H. 264 interpolation [J]. Computer Engineering, 2006, 32(1): 218 ~ 223. [方延龙,周军. H. 264 分数像素内插的快速算法[J]. 计算机工程, 2006, 32(1): 218 ~ 223.]
- 3 Wang Sheng-zen, Lin Ting-an, Liu Tsu-ming, et al. A new motion compensation design for H. 264_AVC decoder [A]. In: Proceedings of IEEE International Symposium on Circuits and Systems [C], Kobe Japan, 2005, 5: 4558 ~ 4561.
- 4 Ren Xiao-hui. Implementation of Interpolation and Weighted Prediction for H. 264/AVS Decoder [D]. Hefei: Hefei Industry University, 2006. [任晓慧. 插值与加权预测在 H. 264/AVS 视频解码器中的实现[D]. 合肥:合肥工业大学, 2006.]
- 5 Hu Li, Wang Feng, Zhen Shi-bao. A hardware architecture for quarter-pixel interpolation in H. 264 [J]. Digital Television and Video, 2005, (10): 14 ~ 17. [胡力,王峰,郑世宝. H. 264 中 1/4 精度像素插值算法的一种硬件实现架构[J]. 数字电视与视频, 2005, (10): 14 ~ 17.]

1 Joint Video Team (JVT) of ISO/IEC & ITU-T VCEG. Draft ITU-T