第33卷 第10期 计 算 机 工 程 2007年5月
Vol.33 No.10 Computer Engineering May 2007
·多媒体技术及应用· 文章编号：1000—3428(2007)10—0193—03 文献标识码：A 中图分类号：TP37

MPEG-4 视频解码模块的设计与优化

张 石，张明亮，鲍喜荣，佘黎煌
（东北大学电子信息工程研究所，沈阳 110004）

摘 要：基于OMAP5910双核处理器设计了一个MPEG-4视频解码模块，对其进行了优化，使之具有实时解码处理的能力。介绍了MPEG-4简单框架的解码过程以及在DSP平台上代码优化过程的重点部分。算法的编写符合TI XDIAS算法标准，通用于C5000系列的DSP平台。该解码过程与优化方法对视频处理以及在OMAP平台开发其他应用有借鉴价值。

关键词：OMAP5910；MPEG-4；解码；优化

Design and Optimization of MPEG-4 Video Decoder

ZHANG Shi, ZHANG Mingliang, BAO Xirong, SHE Lihuang
(Institute of Electronic Information Engineering, Northeastern University, Shenyang 110004)

【Abstract】 This paper designs and optimizes a MPEG-4 video decoder based on OMAP5910 dual-core processor with the power of real-time decoding. It details the important parts of MPEG-4 simple profile decoding process and optimization process on DSP platform. The design of arithmetic is based on TI XDIAS standard and can be used on all the C5000 series of DSP platform. The process of decoder and the method of optimization are valuable as references on video process and other applications on OMAP platform.

【Key words】 OMAP5910; MPEG-4; Decoding; Optimization

视频压缩与解压缩一直是多媒体技术的焦点之一。利用MPEG-4技术压缩的视频以其图像失真小、压缩比大、互动交互等特点，近年来得到了广泛的应用。很多移动终端厂商都在自己的移动终端设备中加入了对MPEG-4视频解码的支持。TI公司近年来推出一款OMAP（Open Multimedia Application Platform）处理器，专门针对第3代移动通信网络以及移动多媒体终端的应用。目前国际上对这款处理器的应用越来越多，几乎所有的3G手机厂商都选择了OMAP平台作为他们未来产品的核心。本文在OMAP平台上设计了MPEG-4视频解码模块，并对其进行了优化，实现了MPEG-4视频的软解码。解码模块可以加入到未来多媒体移动终端的开发中，可扩展性较强。

1 开放式多媒体应用平台（OMAP）

本文采用的OMAP5910处理器是TI公司应用最为广泛的TMS320C55x DSP内核与低功耗、增强型ARM925微处理器组成的双核应用处理器。TMS320C55x系列可提供对低功耗应用的实时多媒体处理的支持，而ARM925微处理器可满足控制和接口方面的处理需要。OMAP5910处理器同时拥有2种产品的最佳性能，且通过优化处理器间的通信机制，使设计者可方便地同时享受这2种处理器的最大优点。基于双核结构，OMAP5910具有极强的运算能力和极低的功耗[1]。

2 MPEG-4解码模块的设计

设计一个MPEG-4 Simple Profile（简单框架）视频解码模块，解码算法的设计依照TMS320 DSP算法标准（XDAIS），通用于TI的C55x系列的DSP平台。

MPEG-4视频解码算法是基于对象编码的，更加注重多媒体系统的交互性与灵活性。基于对象编码使操控视频对象成为可能[2]。在有传输带宽的限制，必须对压缩比特率进行控制的情况下，MPEG-4的编码可以控制对象的比特率，根据对象的重要性进行比特的科学分配，避免整帧图像都受到影响，保证了图像的主观质量。MPEG-4的另一个特点是采用新的高效的压缩算法使得压缩比提高，它可以用相对小的比特率，获得高质量的画面效果[3]。线性MPEG-4视频就是没有对象交互的内容，仅包含视频流。它目前应用最为广泛，也是本文要解码的内容。MPEG-4解码过程如图1所示。

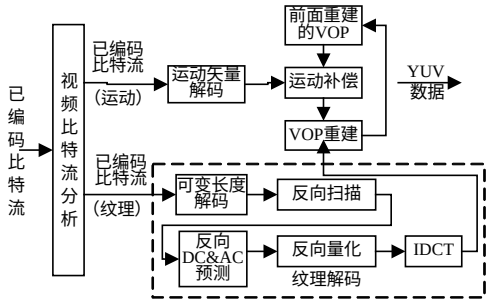

图1 MPEG-4解码模块结构

MPEG-4是基于视频对象平面（VOP）编码的，而VOP是视频对象的时间采样。理论上1帧视频图像可以由多个视频对象（VO）组成，但实际上，目前对于MPEG-4视频对象这个特性的应用不是很广泛。尤其是本文讨论的MPEG-4 Simple Profile编码的情况，1个VOP可以近似为1帧。采用宏块（Marco Block 16×16像素）编码，分为帧内编码（I-VOP）和帧间编码（P-VOP）。一个宏块包含4个亮度块（Block 8

作者简介：张 石(1963－)，男，博士、教授，主研方向：嵌入式系统技术，实时信号处理与DSP技术；张明亮，硕士生；鲍喜荣，硕士、讲师；佘黎煌，硕士、助教

收稿日期：2006-05-30 **E-mail：**brightzml@126.com

×8 像素）和 2 个色度块，为 YUV 4：2：0 格式。解码出的 YUV 数据再经过 RGB 转换，交由 LCD 显示。下面对几个重点部分作说明。

2.1 可变长解码（VLD）

可变长解码的目的就是将视频比特流还原为块内的准 DCT 系数（经过量化以及预测的 DCT 系数）。DCT 系数中的 DC 系数与 AC 系数性质不同，对其的处理也不同。

由于每个块经过 DCT 变换去相关后，得到的 DC 系数保留了这 64 个像素图像的绝大部分信息，而相邻块图像的相似性很大，因此对于帧内编码的 DC 系数采用差分编码（DPCM），从码流中解出的是 DC 系数的差分值。再通过与帧内 DC 预测值相加，得到 DC 系数的量化值。但是当短头格式 short_video_header 等于 1 时，帧内的 DC 系数不是按照差分来编码的，而是以 8bits 无符号整数来编码的。

而 AC 系数经过量化以及预测之后，包含有大量的 0 系数。经过 Zig-Zag（或其他扫描方式）扫描之后，采用游程编码产生 3 个参数（LAST、RUN、LEVEL）组合的 EVENT，然后对 EVENT 进行可变长编码；从码流中解码出 EVENT 后，通过查表求出游程编码，再加上预测值就得到了 AC 系数的量化值。

帧间编码 AC 系数的处理采用与帧内编码相同的方式。对帧间编码 DC 系数，则不进行单独处理，而是与 AC 系数一起进行可变长编码。

可变长解码在解运动信息时也会用到。详细过程可以参考 JPEG 哈夫曼解码过程。

2.2 反向 DC/AC 预测

只有非短头格式的帧内编码才进行 DC/AC 系数预测。如果要做预测，DC 系数必须做预测，而 AC 系数是可选的。

8×8 像素块在经过 DCT 变换后，其 DC 系数和第 1 行或第 1 列的 AC 系数仍然有很强的相关性。所以可以通过预测编码来降低复杂度，使编解码简化。

首先要做的是确定预测方向，如图 2 所示，X、A、B、C分别表示当前块、当前块的前一块、左上块、正上块。预测方向是基于被编码块X周围DC系数的水平和垂直梯度比较的[2]。

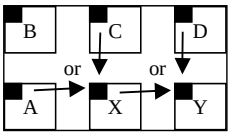

图 2 DC 预测

用如下方法来确定预测方向，其中 $F[0][0]$ 表示先前解码块的反量化 DC 值。

$$if\ (|F_A[0][0]-F_B[0][0]|<|F_B[0][0]-F_C[0][0]|)\ \text{ predict from block C}$$

$$else\ \text{ predict from block A}$$

如果某个预测块（A、B、C）超出了 VOP 边界，则此块的 $F[0][0]$ 用 $2^{(bits\_per\_pixel+2)}$ 即 1 024 代替。

每个块都必须进行 DC 预测，一个宏块中各个块的预测方向独立。根据前面决定的预测方向，预测方法如下：

$$if\ \text{ predict from block c}$$

$$QF_X[0][0]=PQF_X[0][0]+F_C[0][0]//dc\_scaler \quad (1)$$

$$else$$

$$QF_X[0][0]=PQF_X[0][0]+F_A[0][0]//dc\_scaler \quad (2)$$

其中，$QF[i][j]$ 表示系数的量化值；$PQF[i][j]$ 表示由 VLD 解码出的当前块系数差分值；$F[0][0]$ 定义如前；dc_scaler 根据量化参数的不同而取值不同；“ // ”表示取最临近的整数。

而 AC 预测不是必需的，只有在 ac_pred_flag 为 1 时才做，即 AC 预测只在块的第 1 行或者第 1 列进行。预测方向同该块 DC 的预测方向相同。如图 3 所示。

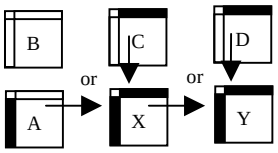

图 3 AC 预测

其过程描述如下，其中 QP 表示宏块的量化参数，其他参数定义同上。

$$if\ \text{ predict from block A}$$

$$QF_X[i][0]=PQF_X[i][0]+(QF_A[i][0]\times QP_A)//QP_X \quad (3)$$

$$else$$

$$QF_X[0][j]=PQF_X[0][j]+(QF_A[0][j]\times QP_C)//QP_X \quad (4)$$

$$i=1\ \text{to}\ 7$$

$$j=1\ \text{to}\ 7$$

对于 AC 预测来说，如果块 A 或者块 C 在 VOP 外部，则对应的预测系数（QF 值）用 0 代替。

2.3 IDCT

经过可变长解码以及预测等过程之后，DCT 系数 $F[i][j]$ 再经过 IDCT 变换就会得到图像 $f[y][x]$。变换的结果需要限制在 $[-2^{bits\_per\_pixel},2^{bits\_per\_pixel}-1]$ 之间。这个解码过程使用了 TMS320C55x 内核自带的 IDCT 硬件加速功能，具体过程不详细说明。

2.4 运动补偿

运动补偿是 MPEG-4 视频解码的核心之一。只有帧间编码时才会用到运动补偿。在编码过程中，通过块的匹配得出运动矢量信息，同时也会得出当前编码块的残差信息（匹配块与编码块差值）。残差信息带有的信息量较少，通过纹理编码。因而解码时，运动补偿通过从码流中获取运动信息并解码出运动矢量。然后根据运动矢量从参考帧中得出预测值。最后将预测值与解码的纹理信息（残差信息）相加，即得到实际的图像。由于 MPEG-4 Simple Profile（简单框架）处理的都是基于块对象的，因此不涉及形状编码，无需填充过程。

要解码出运动矢量信息(MV_x,MV_y)，先由可变长解码解出运动矢量差分信息(MVD_x,MVD_y)。然后与运动矢量预测值(P_x,P_y)相加，得到运动矢量。

4MV 运动矢量预测就是用来获取运动矢量预测值(P_x,P_y)。它通过对 3 个候选矢量(MV_1,MV_2,MV_3）进行中值滤波来实现。这 3 个候选矢量是由与当前解码块邻近的块的运动矢量组成的。对于当前解码块在宏块中不同的位置，选择如图 4 所示。

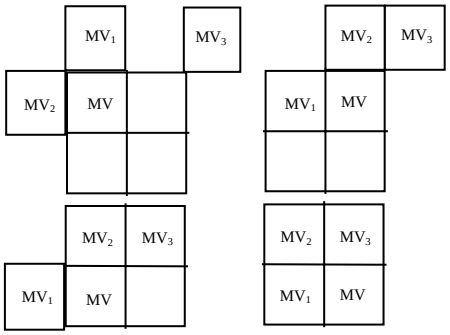

图 4 MV1、MV2、MV3 的选择

图中 MV 是待预测值。上述情况是基于 4MV 而定的，一个宏块中的 4 个块可以各自拥有自己运动矢量。当宏块只有一个运动矢量（1MV）时，使用左上面的情况。另外还有以下 4 个原则：

(1)如果某个候选矢量在 VOP 外部，则该矢量无效；

(2)如果有且仅有 1 个候选矢量无效，则将其置 0；

(3)如果有且仅有 2 个候选矢量无效，则将它们设置为第 3 个候选矢量的值；

(4)如果 3 个候选矢量均无效，则均置 0。

以上为亮度块的情况，对于色度块，其运动矢量没有编码，而是通过对 4 个亮度块运动矢量的运算求得。将 4 个亮度运动矢量求和除以 16，整数部分为运动矢量，小数部分根据查表求得插值单位。

无限制运动补偿允许运动矢量指向参考 VOP 的边界外部，而实际上参考 VOP 的外部像素是不存在的，因而可以通过坐标点转换使其指向参考 VOP 的边界点。但是这增加了额外的运算。另一个方法是将参考 VOP 的边界扩充，然后利用边界点来填充扩出来的边界，称作“贴边”。这么做的目的是省去了坐标点转换的计算，但是增加了参考 VOP 的存储空间，尤其是当参考 VOP 图像很大时。边界的扩充一般选择 32 点和 16 点，分别针对亮度信息和色度信息。在经过运动补偿得出图像预测值之后，再与纹理解码出的信息相加，得出最后图像。

3 MPEG-4 解码模块的优化

优化是模块设计过程中必须考虑的问题。解码模块仅做到正常解码播放还不够，还需要有效率的解码播放。从总体上来说，优化主要是 2 大问题：对存储器操作的优化和对数据运算以及结构的优化。

OMAP5910 DSP内核的内部存储器分为 2 部分：DARAM（双访问）和SARAM（单访问）[4]。DARAM可以在 1 个时钟周期内完成 2 次读或写操作，大小为 64KB。而SARAM1 个时钟周期内只能完成 1 次读或写操作，大小为 96KB。因为 160KB的空间除去存放必要代码外，留给解码用的空间非常少，所以必须借助由DSP MMU映射过来的SDRAM空间。和内部存储器相比其存储速度差了很多，因而对存储器的操作占了很大一部分周期。由ARM端映射过来的待解码的数据也是放在SDRAM中的，在可变长解码过程中，这部分消耗的时间也比较大。另外还包括参考帧的存取、可变长解码的查表等。

在比特流分析以及可变长解码部分，需要频繁读取 ARM 映射过来的、在 SDRAM 中的待解码数据。读取主要是按位读取，因此采用一次读取多位，利用临时变量保存起来的办法来减少对内存的读取次数。另外还通过 DMA 通道将一部分数据导入到内部 RAM 中，提高一次读取的速度。查表过程也是如此。

当前解码块数组（8×8）由于调用极其频繁，因此将其放在 DARAM 中。另外应该将其数据按内存地址对齐，以方便汇编读取。对参考帧的读取优化也很重要，在预测与补偿的时候要用到。优化的方法也是用 DMA 通道将一部分数据导入到内部 RAM。总之，对存储器操作进行优化的原则就是尽量减少读取次数，并灵活利用 DMA 通道的作用。

对运算以及结构的优化的目的是使代码更适合于在 DSP 平台运行。利用 DSP 的结构特点来进行专门的优化。

由于解码出的DCT系数含有大量的0系数，因此将预测、反扫描、反量化嵌入到可变长解码过程中以减少对 0 元素做不必要的运算。

尽量拆分大的循环，用代码量较少的小循环代替。这样可以尽量保证循环体代码整体放入到指令cache中，提高执行效率。尽量保证数据 32 位字对齐，充分利用DSP对 32 位的处理能力，对数据的操作也尽量以 32 位为单位[5]。

运算方面，C55x是定点处理器。所以应该将浮点运算改为定点运算，还要尽量少用除法，用乘法或移位运算代替。调整代码，充分利用DSP桶型移位器以及双MAC乘法器等优势[5]。DSP自带了对IDCT的硬件加速运算[4]，执行效率很高，为优化提供了方便。对于简单运算，使用intrinsics指令代替。对于复杂运算，通过分析尽量将其拆分简化。必要时可以考虑利用一小部分存储空间，采用查表方式替代。

最后，应该考虑将频繁调用以及运算复杂的代码用汇编语言编写，不但能提高执行效率，还可以充分利用 DSP 流水线并行处理的优势。

4 评测与总结

OMAP5910 处理器双核都工作在 150MHz，解码模块工作时需要额外用到 1MB 的 SDRAM 空间，其中包括了存放待解码数据、解码用查表以及参考帧数据。待解码数据由 ARM 端负责不断更新。在代码功能实现后，未进行优化前，播放一段 QCIF MPEG-4 视频明显有停顿现象，平均不到 10 帧/s。优化之后，播放流畅，播放速度在 25 帧/s 以上。达到了实时播放的要求。

OMAP 平台是 TI 面向第 3 代移动通信网络以及多媒体应用推出的高性能、低功耗、具有开放式结构的平台。目前已被多家国际厂商采用，未来应用更广泛。本文基于 OMAP5910 平台设计一个 MPEG-4 视频解码模块，对将来在 OMAP 平台上开发其他应用有很大的参考作用。

另外，文中提到的关于 MPEG-4 解码流程以及代码优化方法，对视频处理以及在其他 DSP 平台的算法优化也有很多借鉴意义。

参考文献

1 肖金锃, 殷小贡. 基于 ARM 核和 DSP 核的 OMAP5910 嵌入式系统[J]. 微型机与应用, 2004, 12(12): 23-25.

2 International Standard. ISO/IEC 14496-2-98 Information Technology Coding of Audio-Visual Objects: Visual[S]. 1998.

3 钟玉啄, 王　琪, 贺玉文. 基于对象的多媒体数据压缩编码国际标准 MPEG-4 及其校验模型[M]. 北京: 科学出版社, 2000.

4 OMAP5910 Dual-Core Processor Data Manual (sprs197d)[Z]. Texas Instruments Incorporated, 2004.

5 TMS320C55x Optimizing C C++ Compiler User's Guide (Rev. E) (spru281e)[Z]. Texas Instruments Incorporated, 2003.