

# H. 264 编解码软件 JM8.6 的核心编码函数研究

陈 举, 赵 琦, 董金明

(北京航空航天大学 电子信息工程学院, 北京 100083)

**摘要:** H. 264 编解码软件 JM 通常被作为算法研究或者实际应用的平台。理解 JM 的程序结构是在其基础上进行进一步研究工作的前提。JM 中的编码函数 `encode_one_macroblock` 在程序实现编码算法的过程中起到核心的作用。通过对程序中开销计算, 关键全局变量及编码一个宏块的算法的分析, 查明了 JM 核心编码函数 `encode_one_macroblock()` 的结构, 给出了该函数的流程。

**关键词:** 高级视频编码; 运动估计; 率失真优化; 宏块; 模式选择

**中图分类号:** TN91    **文献标识码:** A    **文章编号:** 1000-7024 (2008) 17-4498-03

## Research on kernel encoding function of H. 264 CODEC JM8.6

CHEN Ju, ZHAO Qi, DONG Jin-ming

(College of Electronics Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

**Abstract:** JM, an open source CODEC of H. 264, is usually regarded as the platform used in algorithm research and application. Understanding the architecture of JM is a precondition for further research. The function `encode_one_macroblock()` is the center in the implementation of the encoding algorithm. With analyzing of calculation of the cost, the key global variables and the algorithm of encoding a macroblock in the program, the structure of the kernel encoding function `encode_one_macroblock()` and the flow of it is presented.

**Key words:** AVC; motion estimation; rate-distortion optimization; macroblock; mode decision

### 0 引言

H.264 视频编解码标准是由 ITU-T 的 VCEG(视频编码专家组)和 ISO/IEC 的 MPEG(活动图像编码专家组)的联合视频组 JVT 开发的<sup>[1]</sup>。和以前的视频编码标准相比, H.264 获得了更加出色的编码性能, 但需要更加复杂的运算。JM 模型是 JVT 发布的源代码公开的 H.264 编解码软件, 用 C 语言编写而成。JM8.6 是 JM 的一个版本, 尽管 JM8.6(下文简称 JM)没有引进最新的一些运动搜索以及快速模式选择算法, 但研究它的工作方式和程序结构对 H.264 算法研究和在 JM 基础上的代码剪裁仍然很有意义。为了简明起见, 本文只考虑 H.264 基本档次的特性。

### 1 宏块位置及完成的工作

这里介绍编码一个宏块(Macroblock), 在整个 H.264 编码系统中的位置以及负责完成的工作<sup>[2]</sup>, 如图 1 所示, 如果待编码的宏块属于 P slice(如果待编码的宏块属于 I slice, 那么其编码过程包含在下述过程中), 需要在帧内和帧间的几种编码模式中进行模式选择。如果选择的是帧内模式, 则预测值由当前帧已经编码重建的宏块(没有经过去块效应滤波器)

给出, 最佳的模式值经过熵编码输出到编码流。如果选择的是帧间模式, 则预测值由以前编码帧的重建图像给出, 选择的参考帧和运动向量等信息经过熵编码输出到码流。原始图像的值和帧内或者帧间预测得到的预测值相减得到残差数据, 这些残差数据经过变换, 量化后得到的残差系数也经过熵编码输出到码流。另外, 参加熵编码的残差系数经过反量化和反变换, 和预测值相加, 得到重建宏块, 存储在当前帧的重建图像中。当前帧的重建图像全部完成以后, 经过去块效应滤波器的滤波, 将作为参考帧存储起来成为以后编码图像的帧间运动估计的参考。图 1 中虚线包括起来的部分是 JM 的核心编码函数 `encode_one_macroblock()` 所完成的工作, 包括: 运动搜索, 参考帧选择, 帧间模式选择, 帧内模式选择, 帧内帧间判定, 残差数据的计算, 变换和量化, 反变换和反量化, 重建图像的计算。

### 2 开销的计算

开销(cost)是运动搜索, 最佳参考帧选择, 模式选择的依据。在 JM 中, 开销的计算有几种不同的方法<sup>[3]</sup>。

#### 2.1 运动搜索

$$\text{cost\_motion} = \text{SA}(T)D + \lambda \text{motion} * \text{Rate\_motion} \quad (1)$$

收稿日期: 2007-08-16    E-mail: chenju@ee.buaa.edu.cn

**作者简介:** 陈举(1984—), 男, 湖北仙桃人, 硕士研究生, 研究方向为 H.264 视频压缩算法的实际应用; 赵琦, 女, 副教授, 研究方向为多路信息传输与处理、无人飞行器测控系统的可视化、模拟集成电路的设计、嵌入式系统设计等; 董金明(1941—), 男, 上海人, 教授, 研究方向为微波技术、滤波器设计、集成电路设计和机载雷达罩设计。

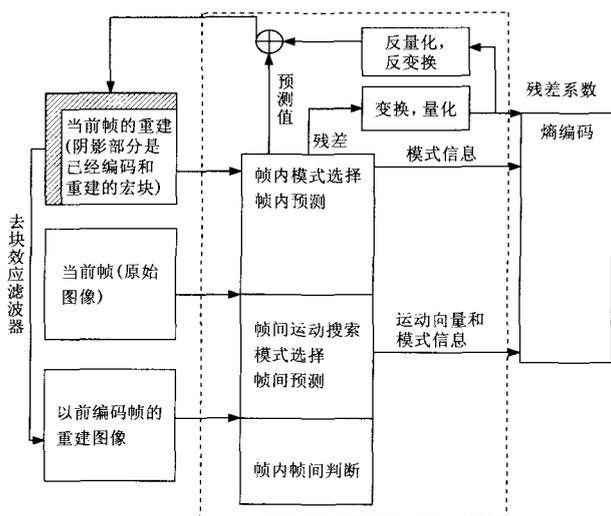


图1 H.264 编码器结构

式中 SA(T)D 根据配置文件(本文采用 JM 中的配置文件 encoder\_baseliencfg, 以下均相同)中的 UseHadamard 取值来进行选择。如果不使用 Hadamard 变换, SA(T)D 表示原始图像块和预测图像块之间的绝对差值和。如果使用 Hadamard 变换, SA(T)D 表示对原始图像和预测图像之间的差值矩阵进行  $4 \times 4$  的 Hadamard 变换, 然后对变换后的矩阵求绝对值和并除以 2。Rate 表示为了编码相应的运动向量所花费的比特数, 这个比特数在 JM 中用查表的方式获得: 编码的比特数事先经过计算存储在一维数组 mvbits[] 中, 使用的时候, 分别用 mv 的横纵两个分量的值(都是  $1/4$  像素精度)作为索引去查找 mvbits[]。lambda\_motion 和下文要提到的 lambda\_mode 称为乘法因子。在 JM 中, 它们由如下的公式计算获得

$$\lambda_{\text{mode}} = 0.85 \cdot 2^{(qp/3.00)} \quad (2)$$

$$\lambda_{\text{motion}} = \lambda_{\text{mode}}^{0.5} \quad (3)$$

式中, qp 表示量化参数。

## 2.2 最佳参考帧的选择

$$\text{cost}_{\text{ref}} = \text{cost}_{\text{motion}} + \lambda_{\text{motion}} \cdot \text{Rate}_{\text{ref}} \quad (4)$$

式中, cost\_motion 表示该在该参考帧中用式(1)计算的运动估计的开销, Rate\_ref 表示为了编码该参考帧的信息所需要的比特数。这个数值在 JM 中用查表的方式获得: 编码的比特数事先经过计算存储在一维数组 refbits[] 中。使用的时候, 用参考帧的编号作为索引去查找 refbits[]。

## 2.3 模式的选择

模式选择开销的计算, 根据配置文件中 RDOptimization 的取值的不同, 有两种方式, 如果没有使用率失真优化

$$\text{cost}_{\text{mode}} = \text{cost}_{\text{ref}} + \lambda_{\text{motion}} \cdot \text{Rate}_{\text{mode}} \quad (5)$$

式中, cost\_ref 表示由式(4)计算的, 在该模式下, 选择的最佳参考帧以及相应的运动估计的总开销。Rate\_mode 表示为了编码该参考帧的信息所需要的比特数。这个数值在 JM 中用查表的方式获得。值得注意的是, 在 JM 中, 查找将仍然使用 refbits[]。原因是, 模式信息和参考帧信息都是一个无符号整数信息。如果使用了率失真优化, 则

$$\text{cost}_{\text{mode}} = \text{SSD} + \lambda_{\text{mode}} \cdot \text{Rate} \quad (6)$$

式中, SSD 表示原始图像块和重建图像块之间的差值和。在

JM 使用式 6 对开销进行计算时, SSD 和 Rate 有多种含义。对于计算 SSD 来说, 有时程序只考虑了亮度分量, 例如 RDCost\_for\_8x8blocks() 函数, 有时候, 程序同时考虑了亮度分量和两个色度分量, 例如 RDCost\_for\_macroblock() 函数。Rate 表示编码所需要的比特数。程序将调用熵编码函数进行熵编码操作以确定 Rate 值。在不同情况下, Rate 有不同的含义。在 Rdcost\_for\_8x8blocks() 函数中, 为了进行帧间的模式选择, Rate 的计算过程将编码下列的信息: 模式, 运动矢量和亮度残差数据。在 RDCost\_for\_4x4IntraBlocks() 函数中, 为了进行亮度帧内预测模式的选择, Rate 只包括模式和亮度残差系数的编码比特数。而在 RDCost\_for\_macroblock() 函数中, 由于同时进行了帧内, 帧间, 亮度和色度的模式选择, Rate 将同时包含以下部分: 宏块头数据, 模式, 运动向量, 亮度残差数据以及色度残差数据。

## 3 编码宏块步骤

编码模式在以下的 7 种帧间, 17 种帧内模式中间选择。7 种帧间模式包括:  $16 \times 16$ 、 $16 \times 8$ 、 $8 \times 16$ 、 $8 \times 8$ 、 $8 \times 4$ 、 $4 \times 8$ 、 $4 \times 4$ , 分别对应模式编号 1~7。17 种帧内模式包括: 以  $4 \times 4$  块为单位进行亮度分量预测的 9 种模式(统称为 I4MB), 以宏块为单位进行亮度分量预测的 4 种模式(统称为 I16MB)和以宏块为单位进行色度分量预测的 4 种模式。值得注意的是, 在进行帧间预测的时候, 只有在选择了把一个宏块划分成 4 个  $8 \times 8$  的块之后, 才能进一步选择  $8 \times 4$ 、 $4 \times 8$  和  $4 \times 4$  的分块<sup>[4]</sup>, 因此, 在 JM 模型中, 把 4~7 的 4 种帧间模式统称为 P8x8。根据配置文件中的 RDOptimization 的取值, JM 采用两套不同的编码步骤。为了简明起见, 本文只讨论 RDOptimization=1(使用率失真)的情况, 并且假定待编码的宏块属于 P slice。编码过程大体可以分为以下 3 个阶段:

阶段 1: 对于帧间模式 mode=1, 2, 3, 分别按照式(1)和式(4)的开销计算标准进行运动搜索和参考帧的选择。

阶段 2: 对于当前宏块的 4 个  $8 \times 8$  块, 首先按照式(1)和式(4)的开销计算标准进行运动搜索和参考帧的选择。然后按照式(6)的标准进行 mode=4, 5, 6, 7 的模式选择。

阶段 3: 按照式(6)的标准, 进行以  $4 \times 4$  block 为单位的 9 种亮度预测模式的模式选择, 以最佳模式作为模式 I4MB 的值。然后以 SA(T)D 值作为开销值, 进行以 macroblock 为单位的 4 种亮度预测模式的模式选择, 以最佳模式作为模式 I16MB 的值。最后, 在帧间模式 1, 2, 3, P8x8 等 4 种模式, 帧内亮度分量预测的 I4MB, I16MB 两种模式, 以及色度预测的 4 种模式之间, 按照式(6)的标准进行模式选择。

最后, 程序将以上编码所得到的参数存储起来, 这些参数随后将进行熵编码以生成视频输出码流。

## 4 重要全局变量说明

JM 模型使用了大量的全局变量, 作为参数传递的媒介。理解这些全局变量的意义是理解程序工作方式的关键。下面对 encode\_one\_macroblock() 中的关键全局变量作一下说明。

(1) img->all\_mv[block\_x][block\_y][list][ref][mode][direction]; 该变量是一个 6 维数组, 存储了一个宏块的 16 个  $4 \times 4$  块在全部参考帧, 全部帧间预测模式下的运动向量。direction=0 表示

运动向量的 x 分量, direction=1 表示 y 分量。函数 BlockMotionSearch() 的主要任务就是进行运动搜索, 并更新该数组的值。

(2) best8x8fwref[mode][block8x8]: 该 2 维数组存储了一个宏块的 4 个 8x8block 在各个帧间预测模式下的最佳前向参考帧。在第 3 节描述的编码过程的第一阶段和第二阶段中, 进行完最佳参考帧的选择后, 会将最佳参考帧的信息存储在这个数组中。编码过程的第 3 阶段, 计算预测值时, 程序会引用该数组作为参考。

(3) best8x8mode[block8x8]: 该维数组存储了一个宏块的 4 个 8x8block 的最佳帧间预测模式。它的作用和意义 and best8x8fwref 类似。

(4) enc\_picture->imgY[pix\_y][pix\_x], enc\_picture->imgUV[uv][pix\_y][pix\_x]: 这两个数组存储了当前编码帧的重建图像的亮度分量和色度分量。其中, pix\_y, pix\_x 表示像素点在整个图像中的纵横坐标。uv 则指示了 U 或 V 分量。这两个数组在计算 SSD 值时被引用。

(5) motion\_cost[mode][list][ref][block]: 该数组存储了一个宏块的分块在不同帧间模式和参考帧下的运动估计的开销值。函数 PartitionMotionSearch() 的主要任务就是更新该数组。同时, 该数组也在决定最佳参考帧时被引用。

(6) img->mpr, img->mprr, img->mprr\_2 和 img->mprr\_c。这 4 个数组分别存储了一个宏块在如下 4 种预测模式下的预测值: 亮度分量的帧间预测, 亮度分量的 4x4 的帧内预测, 亮度分量的 16x16 的帧内预测, 色度分量的 8x8 预测。

### 5 代码分析

按照第 3 节描述的编码步骤对 encode\_one\_macroblock() 函数进行简要的分析。

(1) 对帧间模式 1, 2, 3 进行运动搜索和参考帧的选择。

首先, 程序调用 PartitionMotionSearch() 函数分别对每个分块进行运动搜索。PartitionMotionSearch() 主要进行了 3 项工作: 第一, 选择一个参考帧和分块(如果选择了 P8x8 模式, 则每个 8x8block 还要进行继续的划分), 调用 BlockMotionSearch() 进行每个分块的运动搜索。其中, BlockMotionSearch() 函数找到运动向量并更新 img->all\_mv 的值, 并将对应的运动开销作为函数返回值; 第二, 用 BlockMotionSearch() 的返回值更新 motion\_cost, 作为接下来进行参考帧选择的依据; 第三, 更新 enc\_picture->ref\_idx, enc\_picture->mv 两个全局变量, 这两个变量将作为 BlockMotionSearch() 中运动搜索时运动向量预测的依据。程序从 PartitionMotionSearch() 返回之后, 继续进行最佳参考帧的选择, 选择依据为式(4)。最后, 程序将最佳参考帧的信息存储在数组 best8x8fwref 中。

(2) 对当前编码宏块的 4 个 block8x8 分别进行运动搜索, 参考帧的选择和模式的选择。

运动搜索和参考帧选择的过程与第一阶段基本相同。而模式的选择将按照式 (6) 的标准进行。程序将调用 RDCost\_for\_8x8blocks() 来进行模式开销的计算, 该函数主要进行了以下几项工作。第一, 调用 LumaResidualCoding8x8() 对该 8x8 块进行亮度值的预测, 残差数据的变换量化以及反变换反量化, 以及重建图像的计算。第二, 调用熵编码函数对待编码信息

进行熵编码, 得到 rate 值。第三, 计算 SSD 值并使用式 6 计算开销, 将其作为函数的返回值。RDCost\_for\_8x8block() 的调用关系如图 2 所示。

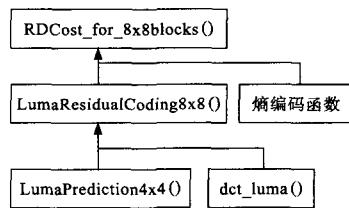


图 2 函数 RDCost\_for\_8x8blocks 的调用关系

其中, LumaPrediction4x4() 首先调用 OneComponentLumaPrediction4x4() 函数, 按照给定的预测模式, 参考帧以及运动向量, 得到一个(或者两个, 对于 B slice 的双向预测而言) 4x4block 的亮度矩阵。然后对该亮度矩阵进行加权处理, 将预测值存储在 img->mpr 中。随后, 程序调用 dct\_luma() 对残差数据进行变换量化以及反量化反变换, 并进行重建块的计算, 将重建块矩阵存储在 enc\_picture->imgY 中, 并将经过重排序和游程编码的系数存储在 img->cofAC 中。程序从 LumaResidualCoding8x8() 函数返回以后, 将紧接着调用熵编码函数计算 Rate 值, 在计算了 SSD 之后, 程序按照式(6)得到编码开销作为函数的返回值。

(3) 最终编码模式的确定。该部分的程序流程如图 3 所示。

完成以上模式选择的操作后, 程序调用 store\_macroblock\_parameters() 函数和 set\_stored\_macroblock\_parameters() 函数存储相关的编码信息。程序从 encode\_one\_macroblock() 函数返回以后, 会紧接着调用 write\_one\_macroblock() 进行熵编码和码流生成工作。

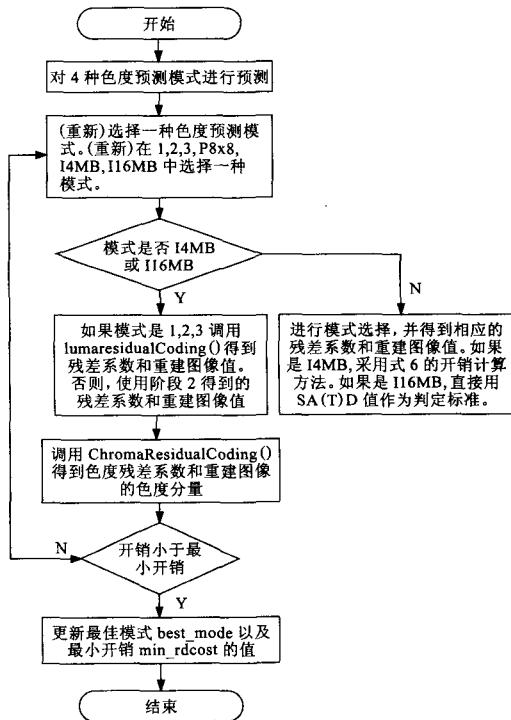
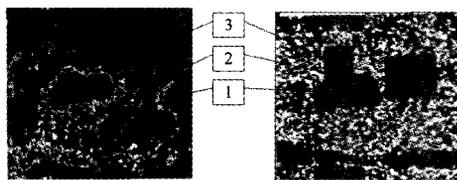


图 3 最终编码模式的决定

(下转第 4506 页)

结构显得不重要,因此宜选用尺度 $2^3$ 下的边缘图像,在该图像中反映出裂缝3的整体轮廓清晰。可见,根据小波变换多尺度边缘检测的特点,可以对同一幅锚杆钻孔图像中的各种地质特征在不同尺度下的边缘图像中进行分析。



(a) Sobel 算子的图像边缘检测 (b) Roberts 算子的图像边缘检测

图2 传统方法的图像边缘提取

比较图2和图1(b)、(c)、(d)的边缘检测结果发现:采用传统算子边缘检测获得的图2中的裂缝1、2、3的边缘特征明显不如采用基于小波变换检测获得的图1中的裂缝1、2、3的边缘特征清晰和完整,并且有许多细微结构没有被检测出来。图2(b)中Roberts算子边缘检测将裂缝1、2、3周围的许多非边缘特征也当作边缘特征检测出来。因此基于小波变换的边缘检测对于有噪声干扰的图像,对噪声有很强的抑制作用。因此,采用小波变换边缘检测可以检测出很弱的边缘特征,从而可以更为完整地反映图像的基本特征,同时也去除了部分噪声。

实验结果表明,基于小波变换的图像边缘检测方法在锚杆钻孔图像边缘检测中具有良好的应用效果。该方法具有多尺度检测的优点,可以利用不同尺度下的边缘图像中检测地质特征。小尺度下的检测结果用于分析图像的精细结构,大尺度下的检测结果用于分析图像的整体结构。在实际应用中,我们应该根据具体问题,选择适当的尺度。此外,从边缘图像中可以进一步计算出地质的某些特征参数。

(上接第4500页)

## 6 结束语

JM作为JVT官方推出的H.264软件编解码模型,主要为测试的目的而开发,它实现了H.264全部特性,通常作为算法研究的平台。本文通过对JM编码部分的核心函数`encode_one_macroblock()`的分析,查明了JM对一个宏块进行编码的程序结构,有利于算法研究工作的展开。例如,新的运动搜索算法可按照`BlockMotionSearch()`函数定义的接口,在该函数中实现,而不用改动程序的其它部分。另外,由于JM关心的是引进新的编码算法和特性来尽量提高编码性能,没有考虑算法复杂度的问题,造成了其源代码冗长,结构复杂。通过对其结构的分析,我们可以有目的裁减代码,去除不需要的特性,作为在目标处理器上(例如DSP处理器)实现算法的软件参考<sup>[5]</sup>。

## 参考文献:

- [1] Thomas Wiegand, Gary Sullivan. Overview of the H.264/AVC video coding standard[C]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 560-575.
- [2] Iain Richardson. H. 264 and MPEG-4 video compression [M].

## 3 结束语

本文利用基于小波变换的模极大值图像边缘检测方法,将多尺度分析、边缘检测和噪声平滑三者融为一体是基于小波变换的图像边缘检测方法的特点,由于该方法相当于利用一个低通平滑函数对图像平滑后再求偏导数,因此平滑的过程中消除了部分噪声。实验结果表明:在锚杆钻孔图像边缘检测中,可以从不同尺度下的边缘图像中检测裂缝特征,取得了良好的效果。在实际应用中,应该根据具体问题的需要进行适当尺度的选择。

## 参考文献:

- [1] 李金生.我国井工煤矿开采技术现状和发展展望[J].煤矿开采, 2002, 7(3): 1-5.
- [2] 袁小平.基于图像处理技术的煤矿岩层监视监测系统研究[D].徐州:中国矿业大学, 2004.
- [3] 沈庭芝.数字图像处理及模式识别[M].北京:北京理工大学出版社, 2003: 98-99.
- [4] 李捷, 唐星科, 蒋延军.几种边缘检测算法的比较研究[J].信息技术, 2007, 24(9): 106-108.
- [5] 李淀, 乔永强, 贺骊, 等.图像改造算法的实现[J].微电子学与计算机, 2006, 23(4): 45-47.
- [6] 胡智君, 张庆丰, 杜培明.一种灰度图像基本处理方法及实现方案[J].仪器仪表用户, 2006, 13(3): 116-117.
- [7] LIEW A W C, LAW N F. Reconstruction from 2-D wavelet transform modulus maxima using projection [J]. IEEE Proc Vision, Image Signal Process, 2000, 147(2): 176-184.
- [8] 彭玉华.小波变换与工程应用[M].北京:北京腾图电子出版社, 2005: 61-63.

England: John Wiley & Sons Ltd, 2004.

- [3] Keng-Pang Lim, Gary Sullivan, Thomas Wiegand. Text description of joint model reference encoding methods and decoding concealment methods[R]. Busan, Korea: Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, 2005-11-09.
- [4] ITU-T Recommendation H.264, Advanced video coding for generic audiovisual services [S].
- [5] 王仁龙.基于VxWorks和DM642视频采集压缩系统的技术研究[J].计算机工程与应用, 2007, 43(11): 216-218.
- [6] Huang Yu-Wen, Hsieh Bing-Yu, Chen Tung-Chien, et al. Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2005, 15(3): 378-401.
- [7] 陈春雁. H.264中基于多分辨率高效帧间预测算法[J].计算机工程与应用, 2007, 43(1): 83-85.
- [8] 周韬. H.264中基于全零块的编码模式快速选择[J].计算机工程与设计, 2007, 28(14): 3404-3406.
- [9] 李军.适用于H.264的直接二维离散余弦变换[J].计算机工程与设计, 2007, 28(14): 3401-3403.