

视频编码中算术编码的二进制化研究

喻莉 戴锐 金欣 朱光喜

(华中科技大学电子与信息工程系, 武汉 430074) (武汉光电国家实验室, 武汉 430074)

摘要 二进制化是自适应二进制算术编码中的重要步骤, 在二进制级别进行概率估计与编码, 能有效降低编解码计算复杂度。本文提出一种新型二进制化方法, 根据相邻变换块的非零系数个数存在的高度相关性, 对变换块的非零系数个数进行自适应二进制化; 在设计二进制化算法的过程中, 还针对概率分布近似对称的情况, 提出一种简化的将语法元素映射为码字序号的规则。实验结果表明, 与传统方法相比, 新的算法能在压缩效率相当的前提下, 有效降低计算复杂度。

关键词 算术编码 自适应 二进制化 码字映射规则

中图分类号: TP37 文献标识码: A 文章编号: 1006-8961(2007)09-1515-05

A Novel Binarization Scheme in Arithmetic Coding

YU Li, DAI Rui, JIN Xin, ZHU Guang-xi

(Electronics & Information Engineering Department, Huazhong University of Science and Technology, Wuhan 430074)

(Wuhan National Laboratory for Optoelectronics, Wuhan 430074)

Abstract Binarization is an elementary step in the framework of context-based adaptive binary arithmetic coding. It is introduced to simplify the context modeling and the subsequent arithmetic coding process. In this paper, we propose a novel binarization scheme for arithmetic coding. Based on the observation that the numbers of nonzero coefficients in neighboring blocks are highly correlated, the number of nonzero coefficients of a block is specifically binarized according to the numbers of nonzero coefficients of its top and left neighbors. In addition, a novel mapping rule is designed to map a syntax element to a code number, according to the observed features of probability distribution. Experimental results show that compared to the conventional scheme, this scheme can reduce decoding complexity with comparable coding efficiency.

Keywords arithmetic coding, adaptive, binarization, mapping rule

1 引言

算术编码(AC)^[1]在理论上可以达到无失真编解码定理给出的极限, 其概率模型与编码完全分离, 比变长编码(variable length coding, VLD)更能灵活适应信源的非平稳特性。算术编码在视频编码中的应用是当今编码研究的热点之一。视频编码标准H.264/AVC采用基于上下文的二进制自适应算术编码(CABAC)算法^[2]。中国正在制定的先进音视频编码标准AVS^[3]也将在其X Profile的熵编码部

分采用自适应的算术编码。

在基于块的混合视频编码领域, 最早的算术编码标准由H.263给出^[4]。其不足之处在于: 第一, H.263的算术编码沿用变长编码的语法元素, 因而算术编码能分配分数码长的特点得不到发挥; 第二, 采用固定的概率模型, 不能揭示数据统计特性的动态变化; 第三, 算术编码器的运算复杂度太高, 不利于实现。H.264中基于上下文的自适应二进制算术编码(CABAC)则克服了上述缺点。它的主要操作包括二进制化、上下文建模与二进制算术编码3个步骤。CABAC在二进制级别进行概率估计与编码,

基金项目: 湖北省自然科学基金项目(CGZ0223); 武汉青年科技晨光项目(20045006071-18)

收稿日期: 2006-06-20; 改回日期: 2006-09-18

第一作者简介: 喻莉(1970~), 女, 教授。1999年于华中理工大学获得通信与信息系统专业博士学位。一直从事计算机网络、移动通信、多媒体信息处理、数据压缩编码等方面的科研和开发工作。E-mail: lyuyxt@sina.com

能大大减少概率模型的建立以及更新的开销;利用上下文信息自适应选择概率模型,揭示了视频数据的非平稳特性;用查表代替乘法运算,能降低算术编码器的计算复杂度。

二进制化将所有非二进制的语法元素映射为二进制字符串。为了在最大程度上降低编码复杂度,二进制化的结果应尽可能的减少冗余。由于不同语法元素的概率分布特性不同,CABAC 为视频流中的每种语法元素设计了特定的二进制化方法^[5],例如,对非零残差系数的幅度用 UEGO 码二进制化,而对亮度残差块的编码模式(CBP)则用定长码进行二进制化。

CABAC 在二进制化这一步骤揭示了各语法元素总体上的统计特性,而视频信息的非平稳特性则由接下来的上下文建模过程来描述。虽然 CABAC 为各语法元素设计了特定的二进制化方法,但每种语法元素在整个编码过程中都采用固定的二进制化策略。本文提出一种新型的根据上下文信息调整的二进制化方法。利用相邻变换块中的非零残差系数个数的相关性,设计出一种对非零残差系数个数进行自适应二进制化的算法。与传统的编码方法相比,本文的方法能在编码效率相当的前提下,有效降低计算复杂度。

2 残差系数的算术编码

在典型的视频编码框架中,原始视频信息经过预测、变换与量化后,最后进行熵编码。基于 DCT 变换的变换块有如下特点:在量化之后,每个变换块中只存在少量的非零系数,它们大多数集中在变换块的左上方。根据这一特点,在对变换块进行算术编码之前,首先将其扫描为幅度信息(level information)与重要性图(significance map),其中幅度信息即为非零系数的幅度大小,重要性图表示了非零系数在变换块中的位置信息。

CABAC 对重要性图的编码算法如下:首先将每个块的所有残差系数映射到一张 1 维数组中。对每个系数设定一个重要性标识位(significant_flag),标识当前系数是否为非零;如果当前系数为非零,则另外传一个末尾重要性标识位(last_significant_flag),标识当前非零系数是否为扫描顺序中的最后一个非零系数。表 1 给出了一个编码重要性图的例子。以 χ_{SIC} 和 χ_{LAST} 分别表示 significant_flag 和 last_

significant_flag 的上下文模型,对扫描序号为 i 的系数 $\text{coeff}[i]$,其对应的两个标识位的模型号由扫描位置而定,如下式:

$$\chi_{\text{SIC}}(\text{coeff}[i]) = \chi_{\text{LAST}}(\text{coeff}[i]) = i \quad (1)$$

表 1 编码重要性图 significance map

Tab.1 The coding of significance map

	扫描位置								
	1	2	3	4	5	6	7	8	9
幅度信息 (level information)	9	0	-5	3	-1	0	-1	0	1
重要性标识位 (significant_flag)	1	0	1	1	1	0	1	0	1
末尾重要性标识位 (last_significant_flag)	0		0	0	0		0		1

3 自适应二进制化

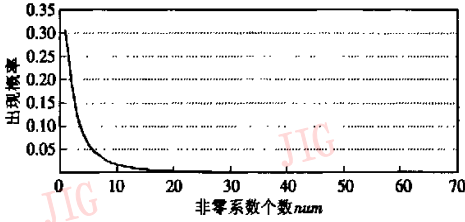
中国的音视频编码标准 AVS 正在如火如荼的制定之中。面向存储应用的 AVS 1.0 的视频部分已获批准成为国家标准。为进一步提升编码效率,AVS 工作组的专家正在制定 AVS 1.0 的增强版本 AVS X Profile。AVS X Profile 与 AVS 1.0 最主要的差别在于熵编码的方法:AVS 1.0 采用变长编码,而 AVS X Profile 将采用压缩率更高的自适应算术编码。AVS 1.0 和 AVS X Profile 都采用了 8×8 大小的变换块,由于熵编码算法的设计依赖于数据的统计特性,传统基于 4×4 变换块的熵编码方法可能不适用于 8×8 大小的变换块。本文正是要设计出适合 8×8 变换块概率分布特性的熵编码方法。

根据相邻变换块中非零系数个数的高度相关性,设计出一种新型的编码变换块中非零系数个数的算法,以替代上述重要性图中末尾重要性标识位(last_significant_flag)的编码。本文的二进制化算法包括两个关键步骤:首先利用上下文信息,为非零系数个数选择适当的码表;其次,根据一种映射规则,将非零系数个数映射为码字序号,进而得出其对应的二进制比特串。

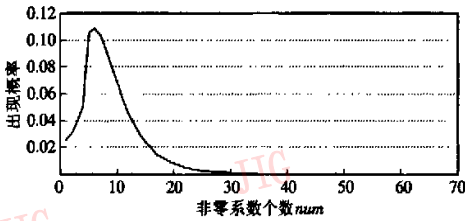
3.1 码表选择

对于一个特定的变换块,设 num 为该块的非零系数个数,leftnum 为其左邻块的非零系数个数,topnum 为其上邻块的非零系数个数。定义 pred 为 leftnum 与 topnum 的均值,利用 pred 来预测当前块的非零系数个数。实验结果表明,num 的概率分布

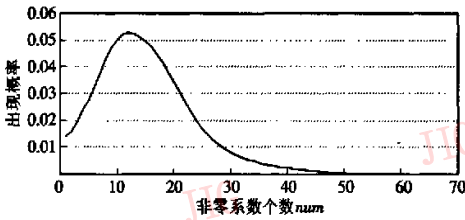
特性随变换块类型和预测值而不同。但不管对哪种变换块, num 的概率分布都有如下特性: 较小的 $pred$ 对应的概率分布较尖锐, 而与较大的 $pred$ 对应的概率分布则较平缓, 图1给出了3种典型的情况。



(a) Chroma 块 (预测值 $pred = 1$)



(b) Inter 块 (预测值 $pred = 7$)



(c) Intra 块 (预测值 $pred = 9$)

图1 非零系数个数的条件概率分布

Fig. 1 Distribution of the number of nonzero coefficients

根据这一特性, 可以为不同上下文预测信息下的 num 分配与其概率分布特性一致的二进制码。将每个预测值 $pred$ 下各种可能出现的 num 值按降序排列, 然后比较不同预测值下的分布特性, 将具有相似尖锐程度的预测值归为一类。本文把各种可能的上下文信息总共分为7类, 其中 Intra 8×8 块2类, Inter 8×8 块2类, Chroma 8×8 块3类, 如表2所示。

为每一类选择一种特定的码以适合该类中 num 的概率分布特性。预测值 $pred$ 越大, 概率分布特性越平缓, 对应表2中, $pred$ 的值越大, 码的阶数就越高。以图1中的3个典型上下文信息为例, 为 Chroma $pred = 1$ 的情况分配的码为 GR1; 为 Inter $pred = 7$ 分配的码为 GR2; 为 Intra $pred = 9$ 分配的码为 GR3。对比这3例可以看出, 本算法所分配

表2 码表选择

Tab. 2 Categories for binarization

块类型	预测值 $pred$	码字类型
Intra 8×8	$0 \leq pred \leq 2$	GR2
	$pred \geq 3$	GR3
Inter 8×8	$0 \leq pred \leq 9$	GR2
	$pred \geq 10$	GR3
Chroma 8×8	$pred = 0$	EG0
	$pred = 1$	GR1
	$pred \geq 2$	GR2

的码的种类能够反映出实际的概率分布特性。

3.2 码字映射规则

表3给出了本方法编码非零系数个数所用到的码字, 从这些码字的结构可知, 符号出现的概率越大, 为之分配的码长应该越短, 即对应的码字序号 ($code_num$) 应该越小。在某个特定的预测值 $pred$ 下, 为 num 分配的码字序号应该由 num 出现的概率大小决定。

表3 二进制化码表

Tab. 3 Codes for binarization

码字序号 ($code_num$)	EG0	GR1	GR2	GR3
0	0	1 0	1 00	1 000
1	0 10	1 1	1 01	1 001
2	0 11	0 1 0	1 10	1 010
3	00 100	0 1 1	1 11	1 011
4	00 101	00 1 0	0 1 00	1 100
5	00 110	00 1 1	0 1 01	1 101
6	00 111	000 1 0	0 1 10	1 110
7	000 1000	000 1 1	0 1 11	1 111
8	000 1001	0000 1 0	0 0 1 00	0 1 000
9	000 1010	0000 1 1	0 0 1 01	0 1 001
10	000 1011	00000 1 0	0 0 1 10	0 1 010
11	000 1100	00000 1 1	0 0 1 11	0 1 011
...

一般来说, 为语法元素分配码字序号的最佳方法是建立 VLC 表。但由于通过实验发现, 非零系数个数的分布还有如下规律: 预测值 $pred$ 近似等于该预测值下最有可能出现的 num 值; 每种预测值下系数的概率分布相对最有可能出现的值近似对称。根据这两个特点, 设计出一种简化的将 num 映射为码字序号的算法。码字映射规则示意图如图2所示。

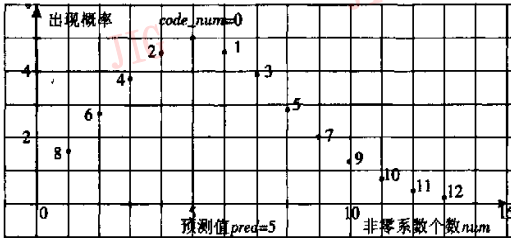


图 2 码字映射规则示意图

Fig. 2 Proposed mapping method

具体描述如下:将最有可能出现的 num (可用 $pred$ 表示) 映射为码字序号 0; $pred + 1$ 与 $pred - 1$ 分别映射为码字序号 1 和 2; $pred + 2$ 与 $pred - 2$ 分别映射为码字序号 3 和 4。也就是说,为每个 num 选择的码字序号由 num 与 $pred$ 的距离而定。

在编码端,码字序号由以下步骤计算:

```

If  $num \leq pred$ 
     $code\_num = 2 \times (pred - num)$ 
Else if  $num < 2 \times pred$ 
     $code\_num = 2 \times (num - pred) - 1$ 
Else
     $code\_num = num - 1$ 

```

相应的解码步骤如下:

```

If  $code\_num \geq 2 \times pred - 1$ 
     $num = code\_num - 1$ 
Else if  $code\_num$  is even
     $num = pred - code\_num / 2$ 
Else
     $num = pred + (code\_num + 1) / 2$ 

```

3.3 自适应二进制化流程

对每个变换块而言,对该块的非零系数个数的二进制化算法总结如下:

(1) 计算预测值 $pred$

如果当前块的上邻块和左邻块都存在,设其上邻块的非零系数个数为 $topnum$,左邻块的非零系数个数为 $leftnum$,则 $pred = (leftnum + topnum) / 2$;如果当前块只有一个上邻块或只有一个左邻块,则将 $pred$ 设置为该邻块的非零系数个数;否则 $pred = 0$ 。

(2) 选择二进制化方法

根据当前块类型以及预测值 $pred$,查找表 2 得到二进制化码的类型。

(3) 计算码字序号 $code_number$

计算当前块中实际的非零系数个数 num ,已知

预测值 $pred$,可根据 3.2 节介绍的方法将 num 映射为特定的码字序号。

(4) 对非零系数个数二进制化

最终的二进制化结果由码的类型和码字序号唯一确定,如表 3 所示。

3.4 自适应二进制化流程

对语法二进制化之后,需要对每个二进制位分配概率模型^[3]。本文对非零系数个数进行二进制化之后,对其模型的分配基于码字的分类与二进制码的位数。具体而言,为表 2 所示的每种分类分配 7 个模型,前缀的前 4 位每位对应一个模型,前缀的剩下所有位对应一个模型;为后缀分配 2 个模型,后缀的前 2 位每位对应一个模型,后缀的剩下位采用旁路方式编码。本算法一共分了 7 种不同的编码类型(如表 2 所示),每种分类对应 7 个模型,所以编码非零系数所需总的模型开销为 $7 \times 7 = 49$ 个模型。

4 实验结果

算术编码主要应用于编码高码率的视频序列,故本文选择一组 AVS 的标准高清测试序列 720P (分辨率为 1280×720) 来测试算法性能。将本文提出的自适应二进制算法与传统编码 $last_significant_flag$ 的算法(简称 LAST 算法)在编码性能和复杂度方面进行综合比较。

实验中采用通用的 PSNR(峰值信噪比)计算公式为

$$PSNR = 10 \lg \frac{(2^n - 1)^2}{MSE} \quad (2)$$

其中, n 为图像样本的位数, MSE 为编码后图像与原始图像像素的均方误差。表 4 给出了两算法的率失真性能比较。其中,“PSNR 增益”一栏给出了本算法相对 LAST 算法的 PSNR 增益,“码率增加”一栏则是将 PSNR 的变化折算成码率变化的结果。本算法的 PSNR 较 LAST 算法的 PSNR 只有 0.03dB 的下降,相当于码率在原来的基础上仅提升了 1.02%。如此小的差别表明,本文算法与 LAST 算法的编码效率是相当的。

以下是两算法的解码复杂度分析。在采用算术编码的系统中,对语法元素的解码过程分为两步:首先从算术解码器(arithmetic decoding engine)读取二

表 4 本文算法与 LAST 算法的性能比较
Tab. 4 A comparison of coding efficiency

序列	PSNR 增益 (dB)	码率增加 (%)
City	-0.027	0.96
Crew	-0.022	1.00
Harbour	-0.039	1.08
Night	-0.034	0.98
Spincalendar	-0.02	1.08
平均	-0.03	1.02

进制符号,再由二进制符号解析出语法元素的值。由于本文算法在二进制化的过程中利用了上下文信息,因此需要读取的二进制符号数少于 LAST 算法,这样就减轻了解码器的负担。LAST 算法的解析过程在代码实现上主要涉及到分支操作,而本文算法的解析过程在代码实现上包括解码 GR/EG 码所必需的和移位操作,以及执行映射操作时的少量分支操作。将两算法总体上的解码复杂度用 x86 处理器所需的时钟周期^[6]来度量。图 3 给出了在 3 种

不同的量化步长下,两算法平均每解码一帧图像所需的时钟周期。实验结果表明,本文算法能够明显降低解码的复杂度,特别是在高码率(QP 较小)的情况下,变换块非零系数的个数相对较多,本文算法的优势更加明显。所以本文算法特别适合于高码率的应用环境。

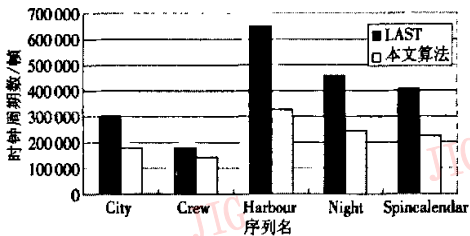
5 结 论

提出了一种算术编码中的自适应二进制化方法。根据上下文信息,对残差块中的非零系数个数选择合适的二进制化方法。实验结果表明,该方法能在不损失编码效率的前提下,有效降低解码端的计算复杂度,并特别适合于高码率的应用环境。

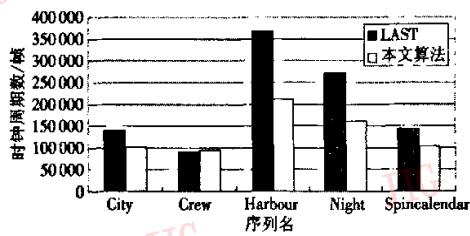
在设计整套二进制化算法的过程中,还提出了一种简化的将语法元素映射为码字序号的方法,此方法适用于语法元素的概率分布近似对称的情况。此外,利用上下文信息进行二进制化的方法也可推广于编码视频中的其他语法元素,如残差幅度信息,运动矢量等。

参考文献 (References)

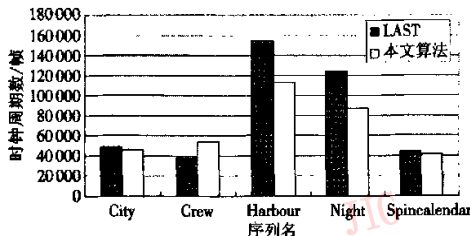
- 1 Witten I H, Neal R M, Cleary J G. Arithmetic coding for data compression [J]. Communications of the ACM, 1987, 30 (6): 520 ~ 540.
- 2 Wiegand T, Sullivan G J, Bjontegaard G. Overview of the H. 264/AVC video coding standard [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13 (7): 560 ~ 576.
- 3 GB/T20090.2-2006. Information Technology-Advanced Audio Video Coding Standard (Part 2): Video [S]. [GB/T20090.2-2006. 信息技术—先进音视频编码(第 2 部分):视频[S].]
- 4 ITU-T H. 263. Video Codec for Low Bit Rate Communication [S].
- 5 Marpe D, Schwarz H, Wiegand T. Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13 (7): 620 ~ 636.
- 6 Intel Instruction Set [EB/OL]. <http://www.xemean.net/resource/doc2.htm>.



(a) QP = 27



(b) QP = 30



(c) QP = 35

图 3 解码复杂度比较

Fig. 3 A comparison of decoding complexity