

基于 H.264 的复杂度可分级的 DCT 算法

张冬明^{1,2)} 林守勋¹⁾ 沈燕飞¹⁾ 张勇东¹⁾

¹⁾(中国科学院计算技术研究所网络与普适计算研究部 北京 100080)

²⁾(中国科学院研究生院 北京 100039)

(dmzhang@ict.ac.cn)

摘要 改进了一种 DCT 快速算法,使之可用于降低 H.264 中整数 4×4 DCT 的复杂度.考虑到 Laplacian 模型不能准确地描述实际的 H.264 DCT 系数的分布,定义了 3 类简化块,并用统计分析的方法分别建立了这 3 类块的系数分布模型,进而提出了一种基于新模型的快速 DCT 算法.设计了简化块的蝶形算法,并分析、比较了所提出的 2 种快速 DCT 算法的复杂度.以后者为基础,设计了一种 DCT 复杂度可分级机制,以调节不同计算能力平台上的 4×4 DCT 的复杂度.实验结果表明,该算法能够准确地按照设定目标控制 DCT 复杂度,且保证编码性能.

关键词 H.264 编码;整数 DCT;快速 DCT 算法;复杂度可分级算法
中图法分类号 TP391

Complexity Scalable Algorithm for DCT in H.264

Zhang Dongming^{1,2)} Lin Shouxun¹⁾ Shen Yanfei¹⁾ Zhang Yongdong¹⁾

¹⁾(Network and Pervasive Computing Research Department, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

²⁾(Graduate University of Chinese Academy of Sciences, Beijing 100039)

Abstract An improved fast DCT algorithm is set up to reduce the complexity of H.264 integer 4×4 DCT. Since the 4×4 DCT distribution does not fit Laplacian model well, three simplified blocks are defined and accordingly three coefficient models are created based on a statistical analysis. Furthermore, a fast DCT algorithm based on new models is proposed. Three butterfly algorithms are designed for those 3 simplified blocks respectively, and the complexities of the proposed algorithms are analyzed and compared. Then, a complexity scalable algorithm based on the latter is presented to adjust the 4×4 DCT complexity on different platforms with limited computation resource. Simulation results show that the algorithm can control H.264 4×4 DCT complexity within the target with negligible loss of encoding performance.

Key words H.264 encoding; integer DCT; fast DCT algorithm; complexity scalable algorithm

0 引言

媒体处理复杂度可分级技术^[1],能够自适应地调整视频编解码的复杂度,以适应不同计算能力的平台.最新的视频编码标准 H.264 采用 4×4 整数 DCT,而不是 H.263 和 MPEG-2 标准中普遍采用的 8×8 浮点 DCT,其变换计算量在整个编码所占比例有所下降.然而,对于一定大小的视频图像,DCT 的

计算复杂度是相对固定的,而运动估计的计算量则会随着编码参数变化.在手机等计算能力受限的平台上,往往采用简单的编码参数配置,运动估计的计算量减少,因而 DCT 计算比重增加,其重要性就凸显出来.所以研究 H.264 中的 DCT 复杂度可分级算法具有重要意义.

快速算法是复杂度分级实现的一种重要的实现方法^[2].早期的 DCT 快速算法侧重于结构优化,文献[3-4]通过对变换矩阵分解将变换过程分为多个

阶段,其间缓存中间结果以避免重复计算.文献 5-6 利用 DCT 系数的分布特点,只计算能量较集中的系数,可进一步降低 DCT 复杂度,但这种方法必须预先指定需要计算的系数,往往会降低编码效率.文献 7 提出了 SSAVT(statistical of sum of absolute value test)算法,可根据系数分布特征自适应地确定需要计算的系数.文献 8 提出的 Approx-Q 算法用低精度的变换矩阵替代浮点 DCT 的变换矩阵.文献 [9]认为 Approx-Q 算法较适合在高码率下降低 DCT 的运算复杂度,而 SSAVT 算法适用于中低码率,在综合这 2 类算法的特点之后,提出一种具有码率自适应性的 DCT 快速算法.我们认为,Approx-Q 算法是针对变换矩阵为浮点数这一特点来降低 DCT 复杂度,不适合整数 DCT;而 SSAVT 算法则是建立在系数分布模型基础上,与变换矩阵是否为浮点数无关.尽管如此,由于整数 4×4 变换与 8×8 浮点 DCT 存在较大的变化,SSAVT 算法不能直接用于 H.264 编码.本文以 SSAVT 算法为基础设计了一种改进的快速 DCT 算法——SSAVT-4;进而在新的系数分布模型基础上提出了一种快速 DCT 算法——ISSAVT-4;最后设计了一种复杂度可分级的 DCT 算法.

1 SSAVT-4 算法

1.1 设计基础

SSAVT-4 算法以 SSAVT 为设计基础. SSAVT 算法利用残差系数和 DCT 系数分布特性来降低 DCT 的复杂度,具体特性包括:1)残差系数分布服从 Laplacian 分布^[7];2)DCT 系数也服从 Laplacian 分布^[10];3)DCT 系数分布的 Laplacian 参数可由 SAD(sum of absolute difference)来计算.依此建立 SAD 与量化参数的关系模型,可以获得量化后 DCT 系数 (u, v) 99% 可能为 0 的充分条件,即

$$SAD < \frac{QP \cdot q(u, v) \cdot N^2}{3\sqrt{2} \Gamma_N(u, v)} \quad (1)$$

其中, q 为量化矩阵, QP 为量化参数,二者乘积表示系数 (u, v) 的量化步长; N 为变换块的尺寸; SAD 为残差块中系数绝对值之和,该值在编码过程就可获得,不需要额外计算;

$$\Gamma_N(u, v) = [D_N R_N D_N^T]_{u, u} [D_N R_N D_N^T]_{v, v} \quad (2)$$

D_N 为 DCT 矩阵, R_N 为残差块系数的相关矩阵^[11],

$$R_N = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & & \rho^{N-2} \\ \rho^2 & \rho & 1 & & \rho^{N-3} \\ \vdots & & & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \rho^{N-3} & \dots & 1 \end{bmatrix}$$

ρ 表示空域系数在一维上的相关性.

根据式(1),就可检测残差块的任一系数经 DCT 和量化后是否为 0. 式(1)右边部分的值称为 SAD 阈值. Zig-zag 扫描将二维 DCT 系数按低频到高频排列为二维,可以认为某个系数检测为 0,其后所有系数均为 0. 如检测全零块时,只需要检测 DC 系数是否为 0. 最后一个非 0 系数的位置称为截断位置,截断位置之后的系数全部置 0,以降低复杂度.

1.2 算法

设 X 为残差矩阵, Y 为变换后的系数矩阵, $A \times 4$ DCT 定义为

$$Y = A \cdot X \cdot A^T \quad (3)$$

$$A = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix},$$

$$a = 1/2, b = \sqrt{1/2} \cos(\pi/8), c = \sqrt{1/2} \cos(3\pi/8).$$

H.264 中的变换并不是精确的 4×4 DCT,它是对式(3)作近似处理得到的一种完全的整数变换,定义为

$$Y = H \cdot X \cdot H^T \otimes S \quad (4)$$

其中,

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix};$$

$$S = \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}, a = \frac{1}{2}, b = \sqrt{\frac{2}{5}};$$

\otimes 表示矩阵的 Kronecker 乘积.

为了降低实现的复杂度,在 H.264 的量化表的设计中,巧妙地将缩放矩阵 S 融合到量化中,在量化阶段就可以近似地通过乘法和移位来实现,避免了除法.这样, H.264 协议中的变换描述为^[12] $Y = H \cdot X \cdot H^T$. 由此可知,式(4)才是 H.264 中实际的变换公式,因此,参照式(2),

$$\Gamma(u, v) = [HR_4 H^T \otimes S]_{u, u} [HR_4 H^T \otimes S]_{v, v}.$$

一旦将缩放矩阵 S 移回到变换中, H.264 的块内系数实际使用相同的量化步长, 它仅和量化参数 QP 有关, 这样式(1)中 $QP \cdot q(u, v)$ 可简化为 $q(QP)$. 因此, H.264 中判定量化后的 DCT 系数 (u, v) 99% 可能为 0 的条件就变化为

$$SAD_4 < \frac{q(QP) \cdot 4^2}{3\sqrt{2}\Gamma_4(u, v)} \quad (5)$$

其中 $SAD_4 = \sum_{j=0}^3 \sum_{i=0}^3 |X(j, i)|$. 利用式(5)可以检测 4×4 整数 DCT 系数的截断位置. 我们把改进后适用于 4×4 整数 DCT 的算法称为 SSAVT-4. ρ 的取值会直接影响 SSAVT-4 算法的性能, 本文采用与文献[7]相同的取值——0.6. 确定检测的系数位置后, $\Gamma_4(u, v)$ 就可计算出来, 因此式(5)右边部分在 QP 设定之后可提前获得.

2 ISSAVT-4 算法

由第 1.1 节可知, DCT 系数服从 Laplacian 分布是 SSAVT 算法建模的基础之一. 然而, H.264 中的 4×4 的整数变换是对实际 4×4 DCT 的近似, 其系数分布规律与 Laplacian 分布存在差别. 很自然地, 我们可通过统计 H.264 中实际 DCT 系数分布情况来寻找其分布特征. 下面建立 SAD 阈值与 QP 新的关系模型, 并利用该模型建立更快速的 DCT 算法 ISSAVT-4.

2.1 系数分布模型

在 ISSAVT-4 算法中, 首先要建立全零块的检测模型; 为进一步降低 DCT 复杂度, 我们还建立了仅有 DC 系数和仅有低频 3 个系数的简化块检测.

没有像文献[7]一样建立低频 2×2 系数检测, 是因为 2×2 块包含的 4 个系数按照 Zig-zag 扫描的顺序是不连续的, 而 4 个连续的系数在分布上是不规整的, 不便于蝶形算法的使用. 为便于说明, 本文中系数全 0、仅有 DC 系数、仅有低频 3 个系数的块分别称为 A 类块、B 类块和 C 类块. 对于有 3 个以上非 0 系数的块, 其所能节省的计算量很小, 故没有定义更多的简化块.

统计模型的建立过程分为 2 个步骤: 数据采样和曲线拟合. 下面以 A 类块为例, 介绍 ISSAVT-4 算法中检测模型的建立过程.

1) A 类块的数据采集

在数据采集实验中, 我们在 H.264 规定的量化参数 $[0, 51]$ 中, 对选择的测试序列使用不同的 QP 进行编码, 编码过程中统计 SAD 的取值与实际的全零块的分布情况. 比如, 对于某个 QP , 编码中统计到 $SAD < 50$ 的情况有 100 次, 而其中有 99 次残差块在变换和量化之后系数全 0, 则可认为在该 QP 下, 某块的 $SAD < 50$, 它就有 99% 的可能为全零块. 在统计过程中我们注意到, 不同的视频序列有类似的结果, 说明块的这一性质是稳定的.

2) A 类块的曲线拟合

对统计的散点数据进行拟合, 可得到 SAD 与 QP 的系数全 0 的关系曲线. 利用拟合的曲线函数, 可以计算出未采样 QP 的 SAD 阈值. 最终获得的 A 类块的阈值曲线如图 1a 所示. SSAVT-4 算法在检测是否全 0 时, 只需要检测 $(0, 0)$ 位置的系数是否为 0, $\Gamma_4(0, 0)$ 可预先计算. 图 1a 中同时给出了 SSAVT-4 算法根据式(5)绘制的全零块阈值曲线.

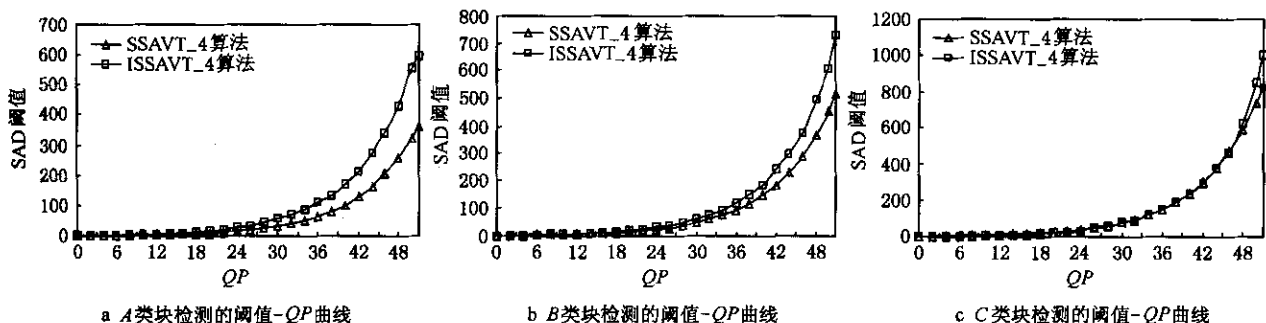


图 1 检测 3 类块的 SAD 阈值与 QP 曲线图

类似地, 可建立 B 类块和 C 类块的检测模型, 相应的阈值曲线如图 1b, 1c 所示. 根据式(5)计算 $\Gamma_4(0, 1)$ 和 $\Gamma_4(2, 0)$, 可以获得 SSAVT-4 算法用于检测 B 类块和 C 类块的 SAD 与 QP 的阈值曲线.

SSAVT-4 算法对亮度和色度采用相同的模型判决需要计算的系数. 其中考虑到亮度分量的分辨率与色度分量的分辨率往往是不一致的(如 $4:2:0$ 格式中亮度分量的分辨率是色度分量的 2 倍), 同时

H.264 的亮度和色度分量的量化参数也不一样。因此,有必要建立独立于亮度的色度块检测模型,建模方法与亮度分量相同,本文略。

2.2 算法

在 SAD 阈值模型基础上即可建立 ISSAVT-4 算法。设 T_A, T_B, T_C 分别为 A、B 和 C 类块的检测阈值,具体步骤如下:

Step1. 初始化。根据初始 QP 计算 T_A, T_B, T_C 。

Step2. 编码一帧。

for all macroblocks in one frame

{

QP 发生变化时,按照线性模型更新 T_A, T_B, T_C

if ($SAD < T_A$)

不计算 DCT;

else if ($SAD < T_B$)

只计算 DC 系数;

else if ($SAD < T_C$)

计算低频 3 个系数;

else

完整计算 4×4 系数;

}

该过程对于亮度分量和色度分量是完全相同的。

2.3 复杂度分析

蝶形算法通过缓存中间结果来避免重复计算,提高变换的计算速度。在算法实现中,将 DCT 分为行变换和列变换,每次变换可看作向量 $x = (x(0), x(1), x(2), x(3))$ 变换到 $X = (X(0), X(1), X(2), X(3))$ 的过程。完整的 4×4 变换的计算过程中行变换和列变换都需要 4 个如图 2 所示的计算过程^[13],因此一个完整的 4×4 DCT 的计算量记为 DCT_{full} ,它包括 64 次加法和 16 次移位。

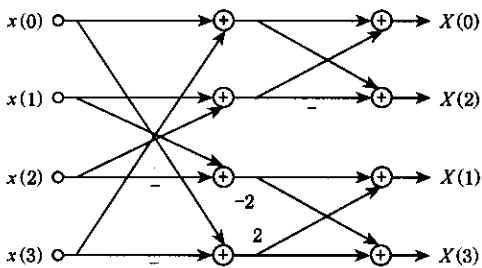


图 2 H.264 整数变换的蝶形运算过程

B 类块中,仅有 DC 系数时的变换过程实质上就是将 16 个系数相加,需要的计算量为 15 次加法,计算量记为 DCT_B ,则 $DCT_B \approx 18.75\% \cdot DCT_{full}$ 。

对于 C 类块的 DCT,在行变换阶段只需要计算上 2 行的变换值,共需要 16 次加法和 4 次移位;而

在列变换阶段,只要对得到的 2 个向量进行变换,相应地,蝶形算法可简化为图 3 所示,虚线部分表示的计算在第 2 个向量变换中也是不需要的,则 C 类块的变换共需要 9 次加法和 1 次移位,计算量记为 DCT_C ,则 $DCT_C \approx 37.5\% \cdot DCT_{full}$ 。

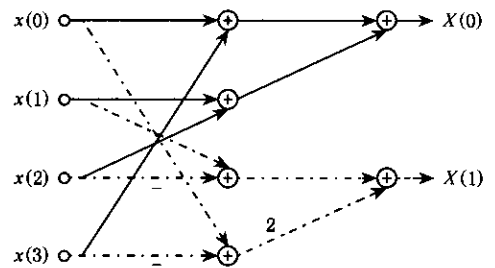


图 3 简化的 C 类块列变换的蝶形运算过程

快速算法对 DCT 复杂度的降低程度用 $C_{DCT} =$

$\frac{CL_{real}}{CL_{total}} \cdot 100\%$ 来衡量,其中 CL_{total} 表示不使用快速

算法时所需要的计算量, CL_{real} 表示使用快速算法后当前帧所有块 DCT 实际需要的计算量。考虑到每个块之间可能有较大的差异,我们在帧级计算 DCT 的复杂度。

设不采用快速算法时需要计算的 DCT 的次数为 a , $CL_{total} = DCT_{full} \cdot a$ 。设 $\lambda_A, \lambda_B, \lambda_C$ 分别为每帧中检测到的系数 A 类块、B 类块、C 类块所占的比例,则

$$CL_{real} = (1 - \lambda_A - \lambda_B - \lambda_C) \cdot DCT_{full} \cdot a + DCT_B \cdot \lambda_B \cdot a + DCT_C \cdot \lambda_C \cdot a,$$

最终可得

$$C_{DCT} = 1 - \lambda_A - 81.25\% \lambda_B - 62.5\% \lambda_C \quad (6)$$

2.4 ISSAVT-4 与 SSAVT-4 算法性能比较

为了具体地比较 ISSAVT-4 和 SSAVT-4 算法的性能,实验平台选择 JM73^[14],编码参数设置如下:量化参数选择 28,32,36,40;参考帧数为 1;运动搜索范围为 ± 16 ;运动矢量精度为 1/4 像素;所有模式有效;熵编码为 CAVLC 编码;环内滤波有效;每个序列编码 200 帧,仅第 1 帧编码为 I 帧,其他帧编码为 P 帧。

从图 1 中可以发现 ISSAVT-4 算法的曲线都位于 SSAVT-4 算法之上,这表示将有更多的块被检测为 A、B 或 C 类块,可以预测 ISSAVT-4 算法的计算复杂度更低。实验中,对使用 SSAVT-4 和 ISSAVT-4 算法时的 λ_A, λ_B 和 λ_C 分别进行统计,并根据式(6)计算相应的 DCT 复杂度。表 1、2 分别给出

了 SSAVT-4 和 ISSAVT-4 算法的复杂度统计数据 (200 帧数据的均值)。实验结果表明,与 SSAVT-4 相

比,ISSAVT-4 能够节约 20% 以上的变换计算量,这与预测结果是一致的。

表 1 使用 SSAVT-4 算法的 C_{DCT}

%

量化参数	序列										
	foreman		carphone		stefan		news		salesman		
	λ	C_{DCT}	λ	C_{DCT}	λ	C_{DCT}	λ	C_{DCT}	λ	C_{DCT}	
28	A	11.2		26.4		16.3		29.5		8.7	
	B	14.7	58.2	15.0	45.7	18.7	68.0	14.1	45.3	10.4	63.5
	C	29.8		25.1		13.7		22.0		30.9	
32	A	22.3		34.2		21.6		34.6		13.8	
	B	17.5	45.7	15.8	37.4	7.9	62.4	15.0	38.8	13.1	53.5
	C	28.6		24.9		15.5		23.1		35.2	
36	A	34.0		44.8		26.9		44.2		21.8	
	B	18.0	33.5	15.9	27.8	8.1	54.7	14.1	30.5	18.0	39.8
	C	28.6		23.1		19.0		22.1		38.1	
40	A	45.5		54.6		31.6		53.3		31.7	
	B	19.6	23.5	16.9	19.7	9.7	45.3	15.0	22.8	25.9	27.8
	C	24.1		19.1		24.4		18.7		31.2	

表 2 使用 ISSAVT-4 算法的 C_{DCT}

%

量化参数	序列										
	foreman		carphone		stefan		news		salesman		
	λ	C_{DCT}	λ	C_{DCT}	λ	C_{DCT}	λ	C_{DCT}	λ	C_{DCT}	
28	A	36.3		50.6		29.7		51.1		28.2	
	B	1.5	51.3	1.3	39.2	0.7	64.5	1.1	39.6	1.4	58.0
	C	18.0		14.7		8.4		13.5		20.5	
32	A	51.8		60.9		35.0		59.2		39.7	
	B	1.8	37.6	1.6	30.0	0.9	58.7	1.3	32.1	2.1	45.8
	C	14.8		12.5		9.0		12.2		20.5	
36	A	63.1		69.7		40.5		66.8		53.4	
	B	5.0	25.3	4.1	20.8	3.0	50.7	4.0	24.0	6.4	30.3
	C	12.3		9.8		10.2		9.5		17.8	
40	A	71.1		76.0		44.6		72.2		65.2	
	B	7.3	16.3	5.8	13.9	5.9	41.4	5.2	17.7	9.4	18.4
	C	10.7		8.6		14.7		9.5		14.0	

为了测试快速算法对编码性能影响,我们还对多个测试序列的率失真 (rate distortion, RD) 数据进行了分析和比较。图 4 所示为 foreman 序列的编码性能测试结果。可以看出,ISSAVT-4 算法和 SSAVT-4 算法的曲线与原始曲线几乎完全重合。这表明采用快速算法后,编码性能几乎没有变化。其他序列的编码性能测试也有类似的结果,限于篇幅,不再一一列出。

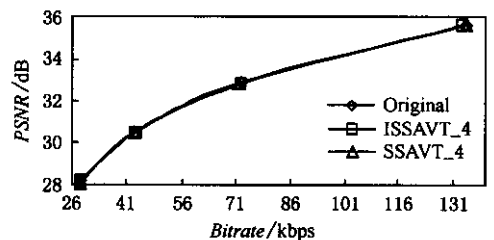


图 4 编码性能的 RD 曲线比较

ISSAVT-4 算法复杂度低于 SSAVT-4 算法的原因如下: 1) 由于 H.264 中 4×4 整数变换与标准的 4×4 DCT 之间存在差异, 变换系数用 Laplacian 分布描述不够准确, 而 ISSAVT-4 算法采用统计数据建模, 更符合系数的实际分布特征; 2) SSAVT-4 算法的性能受 ρ 取值的影响, 而 ISSAVT-4 算法与 ρ 的设置无关; 3) 考虑到亮度分量和色度分量的分布的不同特点, ISSAVT-4 算法针对亮度和色度分量建立不同的阈值检测机制, 而 SSAVT-4 算法对亮度和色度采用相同的阈值。

3 H.264 中变换的复杂度控制

我们设计了一种针对 H.264 中整数编码的复杂度控制机制, 其目标为: 1) 维持复杂度在设定的水平上; 2) 减少帧之间的 DCT 复杂度的波动; 3) 减少因采用快速算法带来的编码性能降低。依据 C_{DCT} 和目标复杂度 C_t 的关系, 复杂度控制可分为 2 种情况: 1) $C_t > C_{DCT}$, 充分利用计算资源尽可能地提高编码效率; 2) $C_t < C_{DCT}$, 降低 DCT 的复杂度。

在确定量化参数之后, C_{DCT} 的分布仅和 SAD 的分布特征有关, 而 SAD 的分布与视频序列特征有关, 统一的 SAD 分布模型难以建立。不过, 由于视频的时间相关性特征, 完全可以根据历史信息来预测 SAD 的分布情况。考虑到各宏块的复杂度之间存在较大差异, 我们在帧级进行 DCT 的复杂度控制。具体的做法是在编码器启动之后, 记录 SAD 值在各范围内的分布概率, 完成一帧编码后更新 λ_A, λ_B 和 λ_C , 计算该帧的 C_{DCT} 。根据 C_{DCT} 与 C_t 的差别, 利用设计的 2 个阈值 Th_Q 和 Th_t 来决定采用粗调还是细调。如果差别大于 Th_Q , 则调节编码的 QP ; 如果

差别大于 Th_t , 调节阈值 T_B ; 差别更小的情况视为达到要求, 不作任何调节。在该复杂度控制算法中, 快速算法选择性能更高的 ISSAVT-4 算法, 具体步骤如下:

Step1. 初始帧编码, 编码过程中统计 λ_A, λ_B 和 λ_C , 计算得到该帧的 C_{DCT} , 记为 C_0 。

Step2. 编码第 n 帧前, 更新阈值 ($n > 0$)

```

if(( $C_{n-1} - C_t$ ) <  $-Th_{QP}$ )
{
     $QP = QP - 1$ ;
    根据新  $QP$  更新  $T_A, T_B$  和  $T_C$ ;
}
else if(( $C_{n-1} - C_t$ ) >  $Th_{QP}$ )
{
     $QP = QP + 1$ ;
    根据新  $QP$  更新  $T_A, T_B$  和  $T_C$ ;
}
else if(( $C_{n-1} - C_t$ ) >  $Th_t$ )
{
     $T_B = (T_B + 2 * T_C) / 3$ ;
}
else if(( $C_{n-1} - C_t$ ) <  $-Th_t$ )
{
     $T_B = (T_B + T_A) / 2$ ;
}

```

Step3. 快速算法编码第 n 帧。

以上只是制定了一种根据编码过程复杂度分布情况自适应更新 SAD 阈值的策略, 降低复杂度由 ISSAVT-4 算法来完成。我们通过一个实验来检验该策略对 DCT 复杂度控制的能力。在实验中, 设置 $Th_Q = 0.05$ 和 $Th_t = 0.01$, 初始 QP 设为 34, C_t 分别设置为 10%、30%、50% 和 70%。统计了 200 帧中 C_{DCT} 的波动情况。结果显示, 设计的复杂度控制策略在多个复杂度水平上都体现了较好的控制能力, 尤其是在 30% 和 50% 的情况下。图 5 所示为 foreman 序列使用该复杂度控制策略后的 C_{DCT} 曲线。

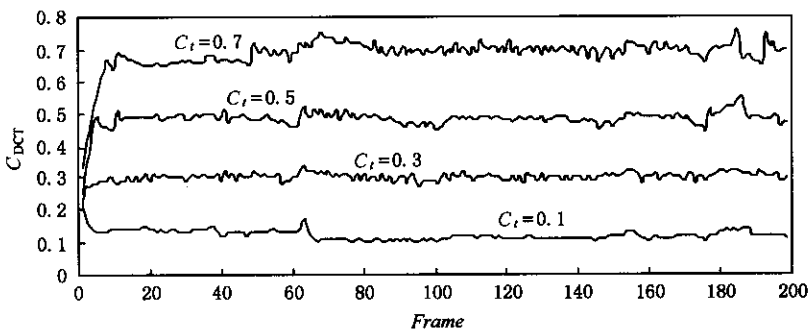


图 5 foreman 序列的 C_{DCT} 曲线

文献 [15] 中提出了一种用于 H.263 的 DCT 复杂度控制方法, 它建立在全零块检测的快速算法基础

上, 通过在帧级调整 QP 来调节 C_{DCT} 。与之相比, 本文控制算法建立在 ISSAVT-4 算法基础上, 能够进行

3 类块的检测. 相应地, 复杂度控制算法除了可调节 QP 外, 还可通过阈值调整更精细地调节 C_{DCT} .

4 结 论

本文提出的 SSAVT-4 算法改进了 SSAVT 的阈值计算方法, 适用于 H.264 的 4×4 整数变换. 进而根据 H.264 中变换系数分布的概率统计分析, 建立了新的系数分布模型, 提出了基于该模型的快速 ISSAVT-4 算法, 在保证相同的编码性能的情况下, 进一步减少了变换的计算量.

利用自适应的 SAD 分布模型, 本文建立了一种复杂度可分级的 DCT 控制策略, 它能够按设定的目标控制 DCT 计算复杂度, 满足设定计算复杂度的情况下, 通过自适应地调整 QP 和阈值 T_B 获得较高的编码效率. 下一步工作将考虑在宏块级调整阈值以加速复杂度的调节过程.

本文设计的复杂度控制策略与 ISSAVT-4 算法结合, 能够有效地实现 H.264 中的 DCT 复杂度分级. ISSAVT-4 算法也可以单独使用, 以最大限度地降低 H.264 的 DCT 复杂度. 此外, 由于量化、反量化、Zig-zag 扫描、反 DCT 等过程与 DCT 是密切相关的, 如果某些 DCT 系数为 0, 这些过程则可相应简化. 在一个实时的、简单参数配置的 H.264 编码系统中, 上述过程加上 DCT 计算所占比重可达 30% 左右, 这表明本文算法对于降低和控制整个 H.264 编码复杂度能够发挥重要作用.

参 考 文 献

- [1] Henschel C, Braspenning R, Gabrani M. Scalable algorithm for media processing [C] // Proceedings of International Conference on Image Processing, Singapore, 2001: 342-345
- [2] Braspenning R, de Haan G, Henschel C. Complexity scalable motion estimation [C] // Proceedings of Visual Communications and Image Processing, Lugano, 2002: 442-453
- [3] Chen W H, Smith C H, Fralick S C. A fast computational algorithm for the discrete cosine transform [J]. IEEE Transactions on Communications, 1977, 25(9): 1004-1009
- [4] Feig E, Winograd S. Fast algorithms for the discrete cosine transform [J]. IEEE Transactions on Spatial Processing, 1992, 40(9): 2174-2193
- [5] Wang Z. Pruning the fast discrete cosine transform [J]. IEEE Transactions on Communications, 1991, 39(5): 640-643
- [6] Skodras A. Fast discrete cosine transform pruning [J]. IEEE Transactions on Signal Processing, 1995, 43(2): 197-205

- [7] Pao I-M, Sun M-T. Modelling DCT coefficients for fast video encoding [J]. IEEE Transactions on Circuits System and Video Technology, 1999, 9(4): 608-616
- [8] Lengwehasatit K, Ortega A. DCT computation based on variable complexity fast approximations [C] // Proceedings of International Conference on Image Processing, Chicago, 1998: 95-99
- [9] Lengwehasatit K, Ortega A. Scalable variable complexity approximate forward DCT [J]. IEEE Transactions on Circuits System and Video Technology, 2004, 14(11): 1236-1248
- [10] Gormish M J. Source coding with channel, distortion and complexity constraints [D]. Stanford: Stanford University, 1994
- [11] Jain A K. Fundamentals of digital image processing [M]. London: Prentice-Hall, 1989
- [12] Richardson I E G. H.264/MPEG-4 part 10 white paper: transform and quantization [OL]. [2005-08-19]. <http://www.vcodex.com>
- [13] Hallapuro A, Karczewicz M, Malvar H S. Low-complexity transform and quantization in H.264/AVC [OL]. [2005-08-19]. <ftp://ftp3.itu.ch/av-arch/jvt-site>
- [14] JVT. JVT reference software JM73 [CP]. [2005-08-19]. <http://bs.hhi.de/suehring/tml/download/JM73.zip>
- [15] Richardson I E G, Zhao Yafan. Adaptive management of video encoder complexity [J]. Journal of Real-Time Imaging, 2002, 8(4): 291-301



张冬明 男, 1977 年生, 博士研究生, 主要研究方向为数字视频编码.



林守勋 男, 1948 年生, 博士, 研究员, 博士生导师, 主要研究方向为多媒体技术及应用系统.



沈燕飞 男, 1976 年生, 硕士, 助理研究员, 主要研究方向为多媒体数字信号处理.



张勇东 男, 1973 年生, 博士, 副研究员, 主要研究方向为视频内容分析、视频数据压缩等.