

## 利用变换域信息快速实现 H.264 帧内预测编码的新算法

苏睿<sup>①</sup> 刘贵忠<sup>①</sup> 张彤宇<sup>②</sup> 刘宝兰<sup>①</sup>

<sup>①</sup>(西安交通大学电子与信息工程学院 西安 710049)

<sup>②</sup>(西安电子科技大学通信工程学院 西安 710071)

**摘要:** 新的视频编码标准 H.264 中采用了帧内预测技术, 能够极大地减少空间冗余性, 从而进一步提高了对帧内宏块编码的效率, 但同时也增大了帧内编码的计算时间。为了减少帧内编码时间和编码延迟, 该文提出了一种能够快速实现帧内预测编码的新算法。该文算法先利用变换域的信息得到图像纹理的方向, 只在最可能的几个模式中进行模式选择, 从而降低了运算量。对不同的视频序列测试的结果表明, 使用本算法后帧内编码时间可以减少 70% 左右, 同时保持相近的图像压缩质量和码率水平。对于一些实时性要求苛刻的视频压缩应用, 采用该文的快速算法会比较有效。

**关键词:** H.264; 视频压缩编码; 帧内预测; 快速编码; 变换域; 先进视频编码

**中图分类号:** TN919.8

**文献标识码:** A

**文章编号:** 1009-5896(2007)01-0161-05

## Fast Mode Decision Algorithm for Intraprediction in H.264/AVC with Integer Transform

Su Rui<sup>①</sup> Liu Gui-zhong<sup>①</sup> Zhang Tong-yu<sup>②</sup> Liu Bao-lan<sup>①</sup>

<sup>①</sup>(School of Electron. & Info. Eng., Xi'an Jiaotong University, Xi'an 710049, China)

<sup>②</sup>(School of Telecommunications Eng., Xidian University, Xi'an 710071, China)

**Abstract:** A tool of intra prediction is introduced in the new video coding standard-H.264, which can cut down the spatial redundancy effectively. Though intra prediction can reduce the bit rate, it is undoubtedly computationally intensive. To overcome this drawback, a novel fast intra prediction algorithm is proposed in this paper. By make use of the information in the transform domain, the direction of texture is estimated at first and then the intra mode decision will be made among several most possible mode, which can diminish the number of calculation. To variable video sequences, the simulation results show that with the proposed algorithm the time of the intra coding can be reduce about 70% while keep the equivalent precision. It is conclusion that the proposed algorithm is suitable to the real-time implements with rigorous requirement on delay.

**Key words:** H.264; Video compression; Intra prediction; Fast encoding; Integer transform; AVC

### 1 引言

国际电信联盟(ITU-T)的视频编码专家组(VCEG)与国际标准化组织(ISO/IEC)的运动图像专家组(MPEG)在 2001 年合作成立了联合视频开发小组(JVT), 并于 2003 年 5 月颁布了视频压缩编码的新一代国际标准 H.264, 也称为 MPEG-4 标准的第 10 部分或先进视频编码(AVC)。与以往的标准相比, H.264 采用了最新开发的一些技术, 在相同的重建图像质量下, 输出的编码比特率显著降低(比 H.263 节省 50% 左右的码率<sup>[1,2]</sup>)。H.264 之所以被称为先进视频编码标准, 是因为它引入了许多新的编码技术, 帧内预测编码就是其中之一。帧内预测技术充分利用了图像的空间相关性, 用图像块周围的像素来进行适当的插值运算, 可以得到与原图像较为接近

的预测图像, 再对原图像和预测图像的残差进行变换编码。与以往标准中直接对帧内图像块做变换编码相比, 做帧内预测后能够明显提高编码效率和降低编码比特率。但是编码效率的提高也带来了计算复杂度的大幅度增加。

最优的帧内预测模式与视频图像块的纹理方向是密切相关的, 利用这个特点, 文献[3,4]提出了一种快速帧内编码算法。先使用索贝尔(Sobel)算子对整幅图像做卷积运算, 求出每个像素亮度和色度梯度的幅度以及方向角。做 4×4 块的帧内模式选择时, 统计得出图像块的累计方向直方图, 据此判定图像块纹理最强的方向, 在这个方向和两个与此方向最接近的方向模式以及平均模式(DC)共 4 个模式中搜索最优模式, 而不是标准中默认的 9 种模式。对于 16×16 或者 8×8 大小的块, 也用相似的方法找到纹理最强的方向, 模式选择在这个方向模式和平均模式共 2 个模式中进行, 而不是默认的 4 种模式。这样做可以减少大约 50% 的运算时间, 同时保证了在同样的码率水平下恢复出解码图像的具有相近的峰

2005-05-16 收到, 2005-09-30 改回

国家自然科学基金(60272072), 国家教育部高等学校博士点基金(2000069828)和跨世纪优秀人才计划(2002 年度)资助课题

值信噪比(PSNR)。但此方法在求方向角时的需要除法运算(次数与图像的像素数相当)。文献[5]提出了一种适合执行帧内预测编码的超大规模集成电路(VLSI)结构,主要对率失真优化(RDO)算法中的哈达玛(Hadamard)变换进行了优化,但对算法本身并没有优化。

本文提出了一种快速帧内预测编码的新算法,用离散余弦变换(DCT)的核心矩阵先对原图像做整数变换,利用变换域的信息可以方便地求出图像块的纹理方向,同文献[3]一样,减少备选模式的数量以减少模式选择的计算量。同 Sobel 算子相比,整数变换的运算量要少很多(因为前者是逐点运算而后者是分块运算)。仿真试验说明本文的算法与文献[3]的算法具有同样的精度,而且所需的运算时间更少。

文章的第2节将简要介绍一下帧内编码过程,本文的算法在第3节中给出,接下来在第4节中将本文的算法与 H.264 中默认的算法以及文献[3]的算法进行对比试验,给出性能参数,最后在第5节中给出结论。

## 2 帧内编码过程

为了进一步提高帧内宏块的编码效率, H.264 使用帧内预测编码技术降低邻近宏块之间的空间相关性。对于亮度分量, H.264 在 9 种 4×4 大小的预测模式(适于纹理细节较复杂,空间相似度低的宏块使用)和 4 种 16×16 大小的预测模式(适于纹理较平坦,空间相似度高的宏块使用)中选择(使用 RDO 准则)一个最优模式。而对于纹理相对较不丰富的色度分量,使用 4 种 8×8 大小的预测模式。

### 2.1 4×4 大小的亮度预测模式

将每个 16×16 大小的宏块划分成 16 个 4×4 的子块,得到每个子块上面和左边的 13 个像素值(并不是原图像的像素,而是已经编码并重建了的像素),图 1 中 4×4 子块的 16 个像素 a, b, ..., p 的预测值可以由 13 个相邻的像素值 A, B, ..., M 计算得到。使用这些相邻像素,按 9 种备选模式插值出 9 幅预测图像,按一定准则(默认是 RDO)找到编码代价最小的模式并记录最小的编码代价。

H.264 中 4×4 亮度的预模式除平均模式(模式 2)以外有 8 种模式,它们具有不同的插值方向。图 2 显示了这 8 种模式的插值方向。其中垂直预测(模式 0)、水平预测(模式 1)和平均预测(相邻像素的均值作为预测值)模式总是有效的,如果上面像素或左边像素都不可用,可用 128 来代替。

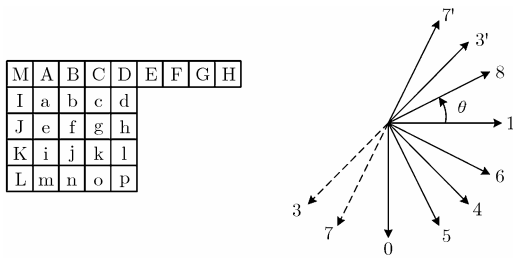


图 1 4×4 大小的子块及其邻近像素

图 2 4×4 子块的 8 种模式的预测方向

### 2.2 16×16 亮度块和 8×8 色度块预测模式

对于图像亮度变化比较平坦的区域, H.264 还提供了 4 种 16×16 大小的候选预测模式,以减少编码模式信息所占用的比特数,分别为:垂直预测(模式 0)、水平预测(模式 1)、平均预测(模式 2)和平面预测(模式 3)。使用这些模式时,整个宏块的亮度分量将作为一个整体来预测,其中前 3 种预测模式与 4×4 大小的预测方法基本相同,只是大小需要扩展,而平面预测(Plane)模式主要针对平缓变化的区域,它由边缘像素经 45 度插值得到。在这 4 种模式中找到使编码代价最小的模式,再与 16 个 4×4 子块的最小代价之和做比较,选择小的作为整个宏块的最优预测模式。

色度分量的变化一般较平缓,所以只使用 8×8 大小的块为预测单位。与 16×16 大小的亮度预测类似,它也有 4 种预测模式,只是预测模式的顺序不同,分别是:平均预测(模式 0)、水平预测(模式 1)、垂直预测(模式 2)、平面预测(模式 3)。同一宏块的两个色度块(U 和 V 分量)采用相同的预测模式。

### 2.3 模式选择

每个宏块进行帧内编码时,亮度信号都要完成 9 种 4×4 预测模式和 4 种 16×16 预测模式,色度信号也要完成 4 种 8×8 大小的预测模式。然后通过模式选择得出一种最佳预测模式使编码的效率最高,实际上这是一种最优化过程。默认采用 RDO 方法进行模式选择过程,按(1)式计算:

$$RDcost = D + \lambda R \tag{1}$$

式中 RDcost (率失真代价)是实际编码比特数和失真度的综合测度,最小 RDcost 对应的预测模式为最优模式。D 采用残差图像像素的累计平方误差(SSE)度量。R 是编码这种预测模式所需的真实比特数。 $\lambda$ 表示 1 比特大约可以编码 SSE 是  $\lambda$  的残差像素,与量化参数(QP)有关,按式(2)取值:

$$\lambda = 0.85 \times 2^{QP/3} \tag{2}$$

由上面的讨论可知,帧内预测编码需要的运算量巨大。因此,有必要使用快速帧内预测算法来降低编码器的运算复杂度,满足实时视频通信的要求。

## 3 快速帧内预测编码算法

图像块变换后的交流(AC)系数也可以很好地反映出纹理方向性,由此本文提出了一种新的快速算法。采用整数变换的核心矩阵对原图像块做变换,用 6 个 AC 系数估计纹理方向,再使用估计出的方向减少候选模式集的大小,以此减少运算时间。

### 3.1 用整数变换估计纹理方向

H.264 采用了整数变换,它是 4×4 的 DCT 的近似,式(3)是反 DCT 的函数形式:

$$f(i, j) = \sum_{u=0}^3 \sum_{v=0}^3 C(u)C(v)F(u, v) \cos \frac{(2i+1)u\pi}{8} \cos \frac{(2j+1)v\pi}{8}, \tag{3}$$

$$0 \leq i, j, u, v \leq 3$$

式中  $f(i, j)$  和  $F(u, v)$  分别是残差像素和 DCT 系数。对式(3)

求  $x$  方向的偏导数得到

$$f'_x(i, j) = \sum_{u=0}^3 \sum_{v=0}^3 \frac{u\pi}{4} C(u)C(v)F(u, v) \sin \frac{(2i+1)u\pi}{8} \cdot \cos \frac{(2j+1)v\pi}{8} \quad (4)$$

定义  $\bar{f}'_x$  是一个块的水平偏导数的平均值，可以按照式(5)计算：

$$\bar{f}'_x = \sum_{i=0}^3 \sum_{j=0}^3 \lambda_i \lambda_j \sum_{u=0}^3 \sum_{v=0}^3 \frac{u\pi}{4} C(u)C(v)F(u, v) \sin \frac{(2i+1)u\pi}{8} \cdot \cos \frac{(2j+1)v\pi}{8} \quad (5)$$

其中  $\lambda_i$  和  $\lambda_j$  是水平和垂直方向的加权值。对于  $\bar{f}'_x$ ，取垂直方向的权值  $\lambda_j \equiv 1$ 。推导式(5)得到

$$\begin{aligned} \bar{f}'_x &= \sum_{u=0}^3 \sum_{v=0}^3 \frac{u\pi}{4} C(u)C(v)F(u, v) \sum_{i=0}^3 \lambda_i \sin \frac{(2i+1)u\pi}{8} \\ &\quad \cdot \sum_{j=0}^3 \cos \frac{(2j+1)v\pi}{8} \\ &= \sum_{u=0}^3 C(u)F(u, 0) \frac{u\pi}{4} \sum_{i=0}^3 \lambda_i \sin \frac{(2i+1)u\pi}{8} \\ &= \eta_1 F(1, 0) + \eta_2 F(2, 0) + \eta_3 F(3, 0) \end{aligned} \quad (6)$$

同理得到垂直偏导数的平均值：

$$\bar{f}'_y = \eta_1 F(0, 1) + \eta_2 F(0, 2) + \eta_3 F(0, 3) \quad (7)$$

由式(6)和式(7)可以估计块的纹理方向角  $\theta$ ：

$$\begin{aligned} \theta &= \frac{180^\circ}{\pi} \arctan \left( \frac{\bar{f}'_y}{\bar{f}'_x} \right) \\ &= \frac{180^\circ}{\pi} \arctan \left( \frac{\eta_1 F(0, 1) + \eta_2 F(0, 2) + \eta_3 F(0, 3)}{\eta_1 F(1, 0) + \eta_2 F(2, 0) + \eta_3 F(3, 0)} \right), \\ &\quad |\theta| < 90^\circ \end{aligned} \quad (8)$$

式(9)是 H.264 中整数变换运算的矩阵表示：

$$\mathbf{F} = \mathbf{C}\mathbf{f}\mathbf{C}^T \otimes \mathbf{E} = \tilde{\mathbf{F}} \otimes \mathbf{E} \quad (9)$$

式中  $\mathbf{f}$  和  $\mathbf{F}$  分别是残差像素和整数变换的系数矩阵， $\mathbf{C}$  是变换矩阵， $\tilde{\mathbf{F}} = \mathbf{C}\mathbf{f}\mathbf{C}^T$  是整数变换的核心变换部分，它的计算只需要整数加法和移位运算。 $\mathbf{E}$  是缩放矩阵。符号  $\otimes$  表示矩阵的对应元素相乘。为了只使用核心变换的结果以减少乘法运算， $\theta$  可以按式(10)计算：

$$\begin{aligned} \theta &= \frac{180^\circ}{\pi} \\ &\quad \cdot \arctan \left( \frac{\eta_1 \tilde{F}(0, 1)E(0, 1) + \eta_2 \tilde{F}(0, 2)E(0, 2) + \eta_3 \tilde{F}(0, 3)E(0, 3)}{\eta_1 \tilde{F}(1, 0)E(1, 0) + \eta_2 \tilde{F}(2, 0)E(2, 0) + \eta_3 \tilde{F}(3, 0)E(3, 0)} \right) \\ &= \frac{180^\circ}{\pi} \arctan \left( \frac{\kappa_1 \tilde{F}(0, 1) + \kappa_2 \tilde{F}(0, 2) + \kappa_3 \tilde{F}(0, 3)}{\kappa_1 \tilde{F}(1, 0) + \kappa_2 \tilde{F}(2, 0) + \kappa_3 \tilde{F}(3, 0)} \right), \quad |\theta| < 90^\circ \end{aligned} \quad (10)$$

其中  $\kappa_i = \eta_i E(i, 0) = \eta_i E(0, i)$ 。为了减少计算量，可以将  $\kappa_i$  的取值限定为整数值。我们设定了 4 种备选方案，分别对几个

标准测试序列做了试验，结果它们的性能都相近，所以不妨选择  $\kappa_i \equiv 1$ ，式(10)变为

$$\theta = \frac{180^\circ}{\pi} \arctan \left( \frac{\tilde{F}(0, 1) + \tilde{F}(0, 2) + \tilde{F}(0, 3)}{\tilde{F}(1, 0) + \tilde{F}(2, 0) + \tilde{F}(3, 0)} \right), \quad |\theta| < 90^\circ \quad (11)$$

### 3.2 预测模式的划分

对于  $4 \times 4$  亮度预测，8 种模式与图像子块的纹理方向有关，它们可以用  $\theta$  来划分。为了将  $\theta$  的取值范围限定为  $[-\pi/2, \pi/2]$ ，可以把模式 3 和模式 7 翻转为 3' 和 7' 模式(见图 2)。于是  $\theta$  的取值范围被划分为 8 个互不交叠的区域，对应 8 种模式。估计纹理方向  $\theta$  后可以按照公式(12)求出最可能的帧内编码模式。同样式(13)和式(14)分别界定了  $16 \times 16$  和  $8 \times 8$  大小模式的纹理方向：

$$\text{mode}_{4 \times 4} = \begin{cases} 0 & \text{while } \theta \in [-90^\circ, -76.6^\circ] \cup [76.6^\circ, 90^\circ] \\ 1 & \text{while } \theta \in [-13.3^\circ, 13.3^\circ] \\ 3' & \text{while } \theta \in [35.8^\circ, 54.2^\circ] \\ 4 & \text{while } \theta \in [-54.2^\circ, -35.8^\circ] \\ 5 & \text{while } \theta \in [-76.7^\circ, -54.2^\circ] \\ 6 & \text{while } \theta \in [-35.8^\circ, -13.3^\circ] \\ 7' & \text{while } \theta \in [54.2^\circ, 76.6^\circ] \\ 8 & \text{while } \theta \in [13.3^\circ, 35.8^\circ] \end{cases} \quad (12)$$

$$\text{mode}_{16 \times 16} = \begin{cases} 1 & \text{while } \theta \in [-22.5^\circ, 22.5^\circ] \\ 0 & \text{while } \theta \in [67.5^\circ, 90^\circ] \cup [-90^\circ, -67.5^\circ] \\ 3 & \text{while } \theta \in \text{else} \end{cases} \quad (13)$$

$$\text{mode}_{8 \times 8} = \begin{cases} 1 & \text{while } \theta \in [-22.5^\circ, 22.5^\circ] \\ 2 & \text{while } \theta \in [67.5^\circ, 90^\circ] \cup [-90^\circ, -67.5^\circ] \\ 3 & \text{while } \theta \in \text{else} \end{cases} \quad (14)$$

### 3.3 本文算法的流程

对于  $4 \times 4$  大小的亮度块，按公式  $\tilde{\mathbf{F}} = \mathbf{C}\mathbf{f}\mathbf{C}^T$  做变换，再按式(11)计算出  $\theta$  后，设置  $\theta$  对应的模式(按式(12))和邻近的两个模式以及平均模式共 4 个模式为候选模式集，在此模式集中进行模式选择(按式(1))。对于  $16 \times 16$  的亮度度块，先每 16 个像素抽取一个像素组成  $4 \times 4$  大小的块，计算出  $\theta$ ，设置  $\theta$  对应的模式(按式(13))和平均模式共 2 个模式为候选模式集，在此模式集中进行模式选择，之后再与先前计算的 16 个  $4 \times 4$  的最小 RDcost 之和比较，RDcost 最小的作为最终选择的模式，按此模式进行编码。对于  $8 \times 8$  的色度块，先每 4 个像素抽取一个像素组成  $4 \times 4$  大小的块，对于  $U$  和  $V$  分

量两个块都做一遍变换,按式(15)估计 $\theta$ :

$$\theta = \arctan[(\tan \theta_1 + \tan \theta_2)/2] \quad (15)$$

式中 $\theta_1$ 和 $\theta_2$ 是对 $U$ 和 $V$ 两个分量按式(11)估计的纹理方向角,设置 $\theta$ 对应的模式和平均模式共2个模式为候选模式集,在此模式集中进行色度分量的模式选择。

表1中列出了计算每个宏块的纹理方向时,文献[3]算法和本文算法需要的计算次数。可以看出使用文献[3]方法需要大量的除法运算和浮点加法运算,而本文算法减少了这些运算。

表1 计算每个宏块的纹理方向需要的计算次数

方法	整数加	浮点加	除法	移位
Sobel	3840	640	384	1536
DCT	1216	1	19	304

#### 4 性能分析和仿真结果

使用JVT提供的JM6.1程序,加入了本文以及文献[3]中的两种快速帧内编码算法,程序在P4 2.8GHz, 1G RAM, Window XP环境下执行。编码时间,编码比特数和重建图像的PSNR值由软件实时计算得出。性能对比试验参照了文献[6]和文献[7]中的部分规定进行设定,具体测试的条件如下:

- (1) 采用3个CIF格式(352×288)的图像,100帧,帧率15帧/秒。
- (2) 采用RDO进行模式选择。
- (3) 采用基于上下文的算术编码(CABAC)。
- (4) 图像组(GOP)的结构采用全I帧。

表2  $Q_p = 28$ 和 $32$ 时的试验数据

视频序列		码率(kbps)		编码时间(ms)		PSNR (dB) $Q_p=28$			PSNR (dB) $Q_p=32$		
		$Q_p=28$	$Q_p=32$	$Q_p=28$	$Q_p=32$	$Y$	$U$	$V$	$Y$	$U$	$V$
Foreman	RDO	947.73	604.10	2569	2459	37.61	41.20	43.95	35.11	40.06	42.34
	Sobel	974.16	626.94	1078	1016	37.59	41.11	43.81	35.09	40.00	42.20
	DCT	996.49	643.84	856	794	37.58	41.13	43.83	35.09	40.01	42.22
Coast guard	RDO	1615.14	1006.70	3016	2709	35.64	43.55	44.96	32.66	42.41	43.95
	Sobel	1643.72	1032.64	1266	1203	35.59	43.48	44.65	32.63	42.42	43.51
	DCT	1703.21	1061.95	956	879	35.58	43.47	44.60	32.61	42.38	43.41
Mother & Daughter	RDO	587.68	377.23	2438	2297	39.51	43.60	44.62	36.96	42.15	43.20
	Sobel	614.93	397.37	1110	1063	39.49	43.55	44.43	36.95	41.95	43.04
	DCT	631.77	406.74	827	762	39.48	43.58	44.51	36.94	42.02	43.08

表3  $Q_p = 36$ 和 $40$ 时的试验数据

视频序列		码率(kbps)		编码时间(ms)		PSNR (dB) $Q_p=36$			PSNR (dB) $Q_p=40$		
		$Q_p=36$	$Q_p=40$	$Q_p=36$	$Q_p=40$	$Y$	$U$	$V$	$Y$	$U$	$V$
Foreman	RDO	388.20	253.31	2335	2125	32.78	38.83	40.63	30.47	37.85	38.97
	Sobel	406.80	267.52	989	967	32.78	38.68	40.41	30.48	37.76	38.92
	DCT	417.03	272.18	762	776	32.78	38.70	40.38	30.48	37.74	38.89
Coast guard	RDO	593.63	337.35	2591	2375	30.10	41.39	42.89	27.77	40.57	41.77
	Sobel	614.48	354.87	1125	1047	30.08	40.91	42.11	27.76	40.10	41.07
	DCT	634.13	366.52	822	747	30.07	40.85	41.95	27.75	40.04	40.81
Mother & Daughter	RDO	230.78	141.06	2328	2094	34.67	40.48	41.51	32.50	39.44	40.51
	Sobel	249.27	158.71	1032	922	34.66	40.35	41.25	32.50	39.25	40.09
	DCT	253.14	159.97	747	715	34.66	40.38	41.27	32.48	39.33	40.14

本文采用峰值信噪比(PSNR)作为客观评价指标,按式(16),式(17)计算PSNR:

$$\text{PSNR} = 10 \lg \left( \frac{255^2}{\text{MSE}} \right) \quad (16)$$

其中MSE是源图像与重建图像之间的均方误差, MSE按下式计算:

$$\text{MSE} = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |x(m, n) - y(m, n)|^2 \quad (17)$$

将量化参数( $Q_p$ )的值分别取28, 32, 36和40,采用默认的方法(即不采用快速算法,记为RDO),文献[3]的算法(记为Sobel)和本文算法(记为DCT)这3种方法进行仿真试验,对码率、编码时间以及解码后图像的亮度 $Y$ 和两个色度 $U$ 、 $V$ 的PSNR这5项性能进行了对比测试,试验结果列于表2和表3。

从这些数据可以看出采用不同的 $Q_p$ 值,本文算法都可以大大减少帧内编码的时间,是RDO方法的31.4%~36.5%,是Sobel方法的71.4%~80.2%。保持了相近的图像质量,亮度 $Y$ 分量的PSNR值只比RDO方法最多下降0.16%,比Sobel方法最多下降0.06%。编码的码率比RDO方法略有增加,增加的比例为5.36%~11.81%,与Sobel方法基本相当,只增加了0.78%~2.67%。在通常情况下,GOP的结构有1个I帧和十几个P帧组成,I帧增加的码率被整个GOP平均后会很低。

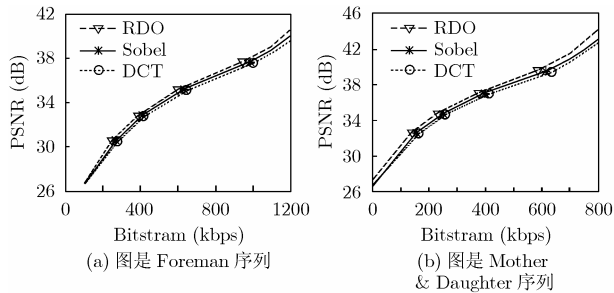


图3 Qp从28增大到40的码率-PSNR图

图3绘出Foreman和Mother & Daughter两个序列的率失真曲线,采用文献[7]中建议的曲线拟和方法。可以看出,采用本文的快速算法得到的率失真曲线与RDO方法得到的曲线具有相同的趋势,而且比较接近,尤其在低码率时。

## 5 结束语

本文介绍了一种利用变换域信息进行H.264帧内预测编码的快速算法,利用6个AC系数可以快速判定原图像的纹理方向,据此得到一个较小的候选模式集,在此模式集中进行模式选择可以减少运算时间。仿真试验数据说明了在相同的解码图像质量和大体相当的码率水平下,本文算法能够大大缩短编码时间。与文献[3]中采用的Sobel算子和直方图的方法,本文的方法更快速简单。对于一些实时性要求苛刻的视频压缩应用,采用本文的快速算法会比较有效。

## 参考文献

- [1] Richardson Iain. H.264 and MPEG-4 video compression: video coding for next generation multimedia. England: John Wiley & Sons Ltd, 2004: Chapter 6.
- [2] Sullivan G J and Wiegand T. Video compression—from concepts to the H.264/AVC standard. *Proceedings of IEEE*, Jan. 2005, Vol. 93: 18–31.

- [3] Pan Feng and Lin Xiao, *et al.*. Fast mode decision for intra prediction. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, 7th meeting, Pattaya, Thailand, Mar. 2004, JVT-G013.
- [4] Pan Feng and Lin Xiao, *et al.*. Fast mode decision algorithm for intraprediction in H.264/AVC video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2005, 15(7): 813–822.
- [5] Huang Yu Wen and Hsieh Bing Yu, *et al.*. Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame code. *IEEE Transactions on Circuits and Systems for Video Technology*, 2005, 15(3): 378–401.
- [6] Sullivan G and Bjontegarrd G. Recommended simulation common conditions for H.26L coding efficiency experiments on low-resolution progressive-scan source material. Presented at the 14<sup>th</sup> VCEG-N81 Meeting, Santa Barbara, CA, Sep. 2001.
- [7] Bjontegarrd G. Calculation of average PSNR differences between RD-curves. Presented at the 13<sup>th</sup> VCEG-M33 Meeting, Austin, TX, Apr. 2001.

- 苏睿: 男, 1974年生, 博士生, 研究方向包括视频压缩编码算法研究、视频压缩系统的VLSI结构设计等。
- 刘贵忠: 男, 1962年生, 博士, 教授, 博士生导师, 主要从事非平稳信号的分析与处理、音视频数据压缩、模式识别、反演等方面的理论、方法与应用研究。
- 张彤宇: 女, 1972年生, 硕士生, 研究方向为视频数据压缩和现代移动通信理论研究。
- 刘宝兰: 女, 1980年生, 硕士生, 研究方向为视频数据压缩和模式识别。