

基于 JPEG2000 渐进过程数截断编码算法

孙文方, 赵亦工, 石自民, 宋蓓蓓

(西安电子科技大学模式识别与智能控制研究所, 西安 710071)

摘要:对JPEG2000的率失真优化截取的嵌入式码块编码算法的各编码块截断过程数进行了分析,对大量的图像进行实验后发现,图像的每个编码块是在相近的过程数截断,于是提出了一种同一过程数截断算法。在此基础上又提出了渐进过程数截断编码算法,最终这个算法舍去了复杂的率失真计算,降低了T₂编码器设计难度,提高了JPEG2000系统的处理速度,减少了存储量,对硬件实现是很好的新方法,处理512×512的8bits位深的图像16倍压缩为250fps。

关键词:JPEG2000; EBCOT; 率失真优化; 等编码过程

Progressive Passing-number Truncation Coding Algorithm Based on JPEG2000

SUN Wenfang, ZHAO Yigong, SHI Zimin, SONG Beibei

(Institute of Pattern Recognition and Intelligence Control, Xidian University, Xi'an 710071)

【Abstract】 Some analysis is focused on each code block passing number of EBCOT (embedded block coding with optimized truncation) in JPEG2000. Through experiments on lots of images, a conclusion can be made that each code block truncates at similar code block passing numbers. Hence this paper addresses a truncation algorithm in the equal-coding pass. Furthermore based on the above algorithm, a progressive passing-number truncation coding algorithm (PPNTC) is proposed which discards complicated rate-distortion calculation and makes it easy for the design of T₂ coder. This algorithm speeds up the processing of JPEG2000 and saves the memory cost. It can be realized more easily on hardware system. The system shows the developed scheme meets the high-speed requirement of the real-time coding 250fps.

【Key words】 JPEG2000; EBCOT; Rate-distortion Optimization; Equal-coding pass

JPEG2000^[1,2]作为新一代的静止图像压缩标准,它采用了离散小波变换,还有D.Taubman所提出率失真优化截取的內嵌码块编码算法(EBCOT)^[3,4]。EBCOT算法分为T₁编码器和T₂编码器两部分。T₁编码器由嵌入式比特平面编码器和MQ算术编码器组成,T₂编码器采用了率失真优化后压缩处理(PCRD),即按照各码块的码流对恢复图像的质量贡献,完成码流的率控制和组织。但是在大量的实验后发现,在要求的8~32倍压缩条件下,图像在0~4个位平面的率失真斜率基本一致,故每个编码块截断的位平面也基本一样。基于此,提出了按位平面渐进截断编码算法,大大简化了程序复杂性,提高了速度。

1 对 JPEG2000 的实验分析

按照 JPEG2000 第 1 部分方案编制的程序对 3 幅测试图像(512x512)进行测试。如图 1 所示,粗线所分割出来的是图像 3 层小波变换后的子带;每个小块是独立编码的编码块(Code Block);小块中的数据表示编码截断时所处的编码过程(索引为序列数,简称过程数);阴影表示该编码块没有需要的码流输出,其中的数据表示其最大的过程数。在 JPEG2000 编码中有 3 个编码过程:重要性传播过程(Significance propagation Pass, SpP),幅值精炼过程(Magnitude refinement, Pass, MrP),清理过程(Clean up Pass, CuP),除了最高位平面编码只用到清理过程,其它位平面编码依次用到以上的 3 个过程,假设一个编码块的最高位平面数为 N(0 表示最低位平面),则完成这个编码块总共需要经历 3N + 1 个编码过程,将这些编码过程按如表 1 的顺序进行标号索引。

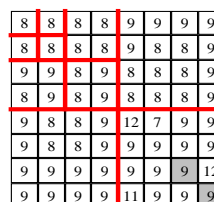


图 1 各编码块截断过程数示意图

表 1 编码过程数索引

位平面数	N	N - 1			...	0		
过程	CuP	SpP	MrP	CuP	...	SpP	MrP	CuP
过程数	3N	3N-1	3N-2	3(N-1)	...	2	1	0

当知道截断过程数 p,很容易计算出截断位平面数|p/3|,(符号|·|表示不大于‘·’的最大整数)。通过以上的一些定义和说明,可以得到一系列如图 2 所示(图 1 是 Lena 图像进行 8 倍压缩后的截断过程数示意图)的截断过程数示意图。

以 Goldhill 图像为例进行分析,从图 2 中可以看出无论是低倍压缩还是高倍压缩,每个编码块的截断过程数相差在 0~4 之间,换算到位平面也就相差 0~1。在此绘制了 Goldhill 图像的第 1 个编码块在编码中的各个过程编码输出累积量(Byte)和位平面的关系如图 3 所示。

基金项目:“十五”国防预研项目

作者简介:孙文方(1979-),男,博士生,主研方向:图像压缩,小波信号处理;赵亦工,教授、博导;石自民、宋蓓蓓,博士生

收稿日期:2006-01-16 **E-mail:** swfile@163.com

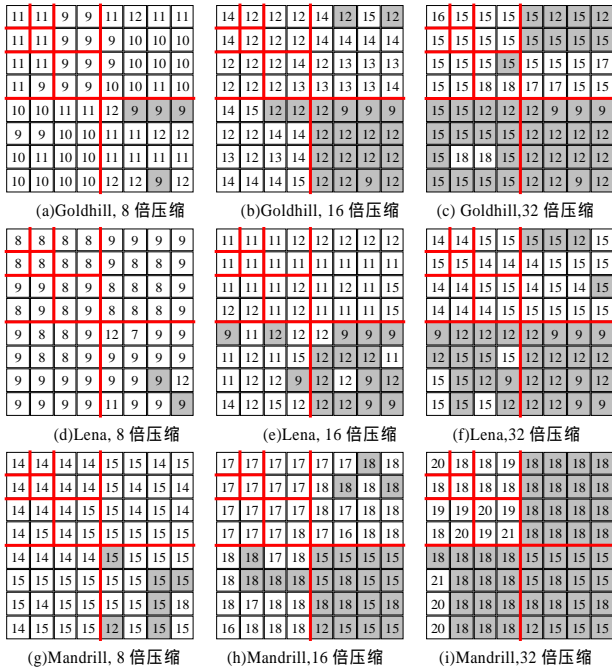


图 2 图像各编码块失真优化后截断过程数

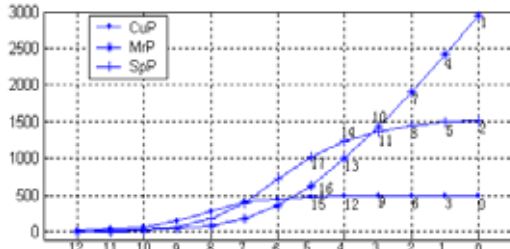


图 3 各过程编码输出累积量(纵轴)和位平面(横轴)的关系

其它编码块的这种关系和图 3 相似，图 3 中各曲线上所标注的数字表示过程数。随着编码的进行清理过程和幅值精炼过程的编码输出增量是逐渐减少的，如果图 3 的编码块在第 10 过程截断，它与在第 9 过程截断效果是一样的，因为在第 9 过程并没有码流输出，所以在图 2 所示的各图中截断过程数的差距实际有可能还要小。由此很容易考虑到如果所有的编码块在同一个过程数截断对于图像恢复的效果是否并没有太大影响？事实上是如此的。

仍以 Goldhill 图像为例进行实验，在按同一过程数截断的方法中独立编码块的先后顺序以“Z”字型进行，在这里是以“在同一个过程数编码中，各编码块对率失真的贡献按子带从低到高的顺序逐渐降低”为假设的。截断算法如下：

$m[i][j]$ ：用来记录编码块 i 在过程数 j 编码结束时所编码输出字节数总和(含以前过程数编码输出)；

$cut[i]$ ：编码块 i 的截取字节数； Cr ：压缩比； $Xsize$ ， $Ysize$ ：图像长和宽； BL ：实际编码长度。

把图像大小，压缩比，及截断信息作为编码后的头文件，假设长度为 Hfl 个字节。则实际编码长度 $BL=Xsize*Ysize/Cr-Hfl$ 。先把每个编码块编码出数据先进行缓存，在编码过程中是很容易得到 $m[i][j]$ 信息的。当所有的编码块编码结束后可以进行如下的同一过程数的截断算法。

程序段(1)

```

For j=0:45
    {Temp=0;
    For i=0:63
        Temp+=m[i][j];
    }

```

```

If (Temp<BL) break;
}
程序段(2)
For i=0:63
    cut[i]=m[i][j];
For i=0:63
    {Temp+=m[i][j-1]-cut[i];
    If (Temp>BL) break;
    Else cut[i]=m[i][j-1];}
cut[i]+=BL-Temp;

```

对以上的伪代码作一简要解释。由于图像 3 层小波变换后最高位平面 $N=15$ ，因此过程数不会大于 $3 \times 15 = 45$ 。程序段(1)是用来确定一个大致同一截断过程数的，由于在这个过程数的所有以前各编码块编码输出长度总和小于或等于目标码流长度，因此在较先的编码块中需要多截取一些数据进行补偿，直到与目标编码长度相等为止，这就是程序段(2)所能达到的。正因为是这样的策略，所以如果靠前的编码块在第 j 个过程数截断，那么靠后的编码块有可能在第 $j+1$ 个过程数截断，如图 4 所示。

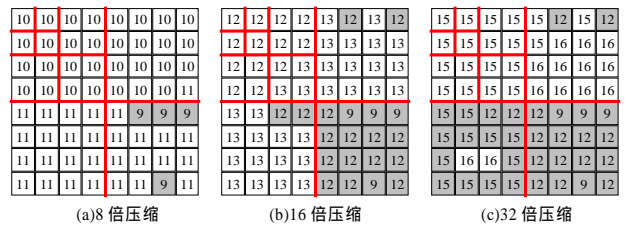


图 4 Goldhill 图像各编码块按同一过程数截断

把标准的 JPEG2000 算法称为 A，把本算法称为 B，对表 2 中的 6 幅标准测试图像进行压缩。进行 3 层 CDF97 小波变换，采用 64×64 的编码块。表 2 中列出了对每幅图像压缩后重建图像的峰值信噪比(PSNR)值(单位 dB)以及 B 方法相对 A 方法 PSNR 的变化($\Delta PSNR$)值。

表 2 两种方法在图像恢复质量上的比较

压缩倍数	图像 算法	Goldhill	Lena	Aieplane	Boat	Mandrill	shapes
		8	A	36.370	40.068	40.897	39.087
	B	36.344	40.004	40.798	39.071	28.957	38.023
	PSNR	-0.026	-0.064	-0.099	-0.016	-0.035	-0.007
16	A	33.066	37.092	36.402	34.322	25.508	35.578
	B	32.938	37.075	36.348	34.286	25.417	35.533
	PSNR	-0.128	-0.017	-0.054	-0.036	-0.091	-0.045
32	A	30.388	33.930	32.281	30.692	23.097	33.210
	B	30.356	33.892	32.241	30.625	23.079	33.200
	PSNR	-0.032	-0.038	-0.040	-0.067	-0.018	-0.010

从表 2 的数据可以看到这个方案相对 EBCOT 来说是相当的：对于小波变换后的数据进行 3 个过程扫描压缩时，不同的编码块在相同的过程数有相近的率失真斜率，用率失真截断的方法也会截断到相近的过程数，而且小波变换会产生一定的鲁棒性，因此在同一个过程数截断只有很小的质量降低。这个方案相对于 EBCOT 来说少了比较复杂的率失真斜率的计算。

2 渐进过程数截断算法

B 算法中仍需要将所有的编码块完全编码后进行截断，但是这样将产生很多最终并不需要的编码输出，有时难以达到实时性和存储要求。如何减少不必要的码流输出，进而加快处理速度，提出了基于 B 算法的渐进过程数截断算法(PPNTC，算法 C)。

程序段(1)

```

end_j=0;
For (i=0;i<=63;i++)
{Code Block (i,end_j)
for(j=end_j;j++)
{Temp=0;
For(q=0;q<=i;q++)
Temp+=m[q][j];
If (Temp<BL) break;
Else end_j=j; }
}

```

程序段(2)

```

Temp=0;
For (i=0; i<=63;i++)
{cut[i]=m[i][end_j+1];
Temp+=m[i][end_j+1];
}
For (i=0; i<=63;i++)
{Temp+=m[i][end_j]-cut[i];
If (Temp>BL) break;
Else cut[i]=m[i][end_j];
}
cut[i]+=BL-Temp;

```

上面的伪代码中 Code Block(i,end_j)是指在对编码块 i 编码时,只需编到第 end_j 个过程数结束为止,后面的过程不再编码。每当一个编码块结束编码后就对 end_j 更新,更新的原理是因为如果前 i 个编码块在第 j 个过程数的 m[q][j], q=0,1,...,i-1 的和大于 BL 则前 i+1 编码块有 m[q][j] q=0,1,..., i 的和仍大于 BL, 所以第 i+1 编码块的第 j 个过程数以后的过程不再需要编码。

图 5(a)所示为 Goldhill 图像在 16 倍压缩时每个编码块在编码过程中的停止编码过程数,即上面提到的 end_j,可见它是在不断提升的。其最后一个附加方框中的数据即为最终的截断过程数,然后利用程序段(2),得到如图 5(b)所示的编码块过程数截断图,当然其左上角的数据必然和图 5(a)附加框的数据一致。

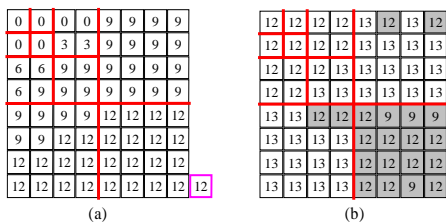


图 5 Goldhill 图像编码各子块截断过程数

把 A、C 两个算法的执行速度比较,只列出了编码耗时(单

(上接第 194 页)

根据以上的实验结果,可以知道使用改进过的叉点精细化算法,可以进一步提高叉点的准确率,为脱机手写汉字识别提供了一个非常好的基础。

参考文献

1 Lin F, Tang X. Off-line Handwritten Chinese Character Stroke Extraction[C]. International Conference on Pattern Recognition, Canada, 2002: 249-252.

位 s, 见表 3), 编制了 32 位的标准 C 语言程序, 在 DOS 下运行, CPU 为 PIII933。由于没有太多注重编程技巧, 因此这些数值有些偏大, 但是参考这些确切的数据可以肯定地说算法 C 在运行速度上有很大优势。

表 3 两种方法在运行耗时比较

压缩倍数	图像 算法	Goldhill	Lena	Aieplane	Boat	Mandrill	Shapes
		16	A	1.21	1.15	1.15	1.21
	C	0.55	0.55	0.55	0.50	0.44	0.61
	T	-0.66	-0.60	-0.60	-0.71	-0.93	-0.60
	T/A	-54.6%	-52.2%	-52.2%	-58.7%	-67.9%	-49.6%

当忽略头文件(一般很小), 则 16 倍压缩时的目标码流长度 BL=16 384, 在表 4 中 A-BL 对应的是每个编码块完全编码时产生的码流长度(单位 Bytes), 算法 A 就是这样的, C-BL 对应的是用算法 C 时产生的码流长度, 可见这个方案可以减少很多不必要的码流输出。算法 C 在硬件设计中将大大降低复杂度, 减少缓存空间和提高速度。

表 4 两种方法在输出码流长度的比较

压缩倍数	图像 算法	Goldhill	Lena	Aieplane	Boat	Mandrill	Shapes
		16	A-BL	158 501	140 034	130 478	143 464
	C-BL	43 117	39 134	39 706	40 260	43 407	42 083
	L	-115 324	-100 900	-90 772	-103 204	-159 341	-109 320
	L/A	-72.8%	-72.1%	-69.6%	-71.9%	-78.6%	-72.2%

3 结论

本文提出的渐进过程数截断算法降低了系统复杂性、提高了系统的处理速度, 减少了存储量的使用, 易于硬件实现。在大量的实验中这个算法是相当稳定的, 不会因为特殊的图像而产生很坏的效果。这是一个要用于硬件实现的算法评估方案, 从本文的分析可以看出, 这个方案在硬件实现上相对于 PCRD 来说在时间和空间复杂度上将有很大优势, 目前硬件研制已经完成, 处理 512 x 512 的 8bits 位深的图像 16 倍压缩为 250fps。

参考文献

1 ISO/IEC. FCD15444-1: Information Technology JPEG2000 Image Coding System-Part 1: Core Coding System[S]. 2000-03.
2 Christopoulos C, Skodras A, Ebrahimi A. The JPEG2000 Still Image Coding System: An Overview[J]. IEEE Trans. on Consumer Electronics, 2000, 46(4): 1103-1127.
3 Taubman D. High Performance Scalable Image Compression with EBCOT[J]. IEEE Trans. on Image Processing, 2000, 9(7): 1158-1170.
4 Taubman D, Ormentlich E, Weinberger M, et al. Embedded Block Coding in JPEG2000[C]. IEEE Int. Conf. on Image Processing, 2000-09, 12: 33-36.

2 Zhang T Y, Suen C Y. A Fast Parallel Algorithm for Thinning Digital Pattern[J]. Communication of the ACM, 1984, 27(3): 236-239.
3 Cao R, Tan C L. A Model of Stroke Extraction from Chinese Character Images[C]. International Conference on Pattern Recognition (ICPR), Spain, 2000: 368-371.
4 Chen Y S, Hsu W H. An Interpretive Model of Line Continuation in Human Visual Perception[J]. Pattern Recognition, 1989, 22(5): 619-639.