

# 无失真 JPEG 图象压缩方案与 基于 S 变换的无失真图象压缩

许刚 (中国科学院软件研究所 100080)

廖斌 (武汉水利电力大学计算机科学系 430072)

李承毅 (长沙湖南省公安厅 410001)

**摘要:**基于 DCT 变换的 JPEG 图象编码方案是众所周知,但 ISO JPEG 另一种无失真的图象编码方案却很少介绍,本文将对无失真的 JPEG 方案作详细介绍,同时对图象的新一代无失真压缩 S+P 算法也给出详细的论述。实验结果表明,S+P 算法对无失真图象压缩有更好的压缩比。

**关键词:**无失真 JPEG 图象编码 预测编码 S+P 算法 图象压缩

## 1. 引言

在本文中,我们描述一种多分辨率变换用于图象的无失真压缩,称为 S+P 变换。这种新的图象编码方法类似于图象的子带编码,但只需通过简单的加法和移位操作就能完成计算。实验结果表明,在图象的压缩效率上,S+P 变换要比具有相同复杂度的单分辨率线性预测编码方法要高,而且计算量也非常小。同时,我们还提出一种熵编码方法,它利用了图象的多尺度的结构,能有效地压缩 S+P 变换后图象的系数,进行渐进分辨率的图象传输。这种图象编码方案表明,适当的图象变换可以融合在无损的图象压缩方案中,并且对于压缩效率和编码复杂度而言只产生可以忽略的影响。

下面将分别详细介绍这两种图象编码的原理及实现方法。

## 2. 无失真 JPEG 图象压缩方案

为了讨论的方便,可以将图象看作为一个二维样本数组。每个样本值代表一个象素值,对于彩色图象为 RGB 三个分量中的一个,对于灰度图象则只有一个分量。每个分量定义为具有 2~16 位精度的整数。点变换参数  $P_t$  能够用来降低源图象的精度。如果  $P_t$  不为零,则每个输入样本要右移  $P_t$  位,结果进入编码器形成新的样本。

ISO JPEG 标准定义了八个 PSVs,由表 1 给出。PSV 0 通常不用而保留给其他编码方法。对于每个象素  $P_{x,y}$ ,利用它的左边象素  $P_{x,y-1}$ ,上方象素  $P_{x-1,y}$  和左上方象素  $P_{x-1,y-1}$  的线性组合来计算  $P_{x,y}$  的预测值。PSV 1,2,3 是一维预测器(只用到一个相邻象素)。PSV 4,5,6,7 是二

维预测器(用到两个或多个相邻象素)。

表 1 无失真 JPEG 图象压缩方案的预测值

PSVs	预测器
0	无预测
1	$P_{x,y-1}$
2	$P_{x-1,y}$
3	$P_{x-1,y-1}$
4	$P_{x,y-1} + P_{x-1,y} - P_{x-1,y-1}$
5	$P_{x,y-1} + (P_{x-1,y} - P_{x-1,y-1})/2$
6	$P_{x-1,y} + (P_{x,y-1} - P_{x-1,y-1})/2$
7	$(P_{x,y-1} + P_{x-1,y})/2$

在进行无失真 JPEG 图象预测时,图象的第一行和第一列需要单独处理,PSV 1 用于第一行的预测,PSV 2 用于第一列的预测。对于图象左上角的样本,用  $2^{P-P_t-1}$  作为预测器,其中 P 为样本的精度。从象素值中减去该象素的预测值得到的差值用 Huffman 或算术熵编码器进行编码,这里我们使用的是 Huffman 编码。Huffman 熵编码模型将差值划分为 16 类,记为 SSSS,其中 SSSS 为 0~15 的整数。每一类差值用缺省的 Huffman 表(见[1])或根据图象定制 Huffman 表来分配一个码值。熵模型利用少量的差值符号集来减小定义和存储 Huffman 表所需的耗费,差值符号类如表 2 所示。

为了完整地定义一个差值,需要扩展 SSSS 的额外位数来表示差值的符号和幅度。扩展的规则如下:如果差值是正的,则扩展 SSSS 低位的位数来表示差值。如果差

表2 差值符号类的 Huffman 表

SSSS	差值
0	0
1	-1,1
2	-2,-1,1,2
3	-7,-3,3,7
4	-15,-8,8,15
5	-31,-16,16,31
6	-63,-32,32,63
7	-127,-64,64,127
8	-255,-128,128,255
9	-511,-256,256,511
10	-1023,-512,512,1023
11	-2047,-1024,1024,2047
12	-4095,-2048,2048,4095
13	-8191,-4096,4096,8191
14	-16383,-8192,8192,16383
15	-32767,-16384,16384,32767
16	32768

值是负的,则从差值中减去1得到的结果来扩展 SSSS 低位的位数。最有意义的位放在扩展位数的前面。下面这个例子图示编码过程,如图1。设源图象为一个 2x3 维的灰度图象,我们使用  $PSV7((P_{x,y-1} + P_{x-1,y})/2)$  作为预测器和一个优化的 Huffman 表来编码差值类, Pt 设为 0。

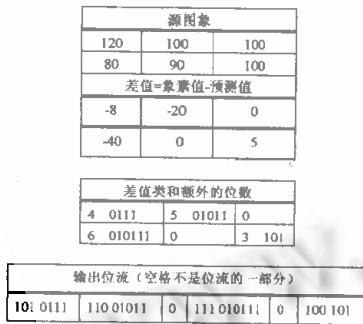


图1 2x3 图像 JPEG 的图示

以上解码经过编码的位流, Huffman 码字 101 首先取出并且经过 Huffman 解码得到符号类 4, 然后 4 的额外位数 0111 从位流中取出, 首位的零表示差值为负数, 于是差值的幅值等于 1 加上二进制值 0111, 再将重构的差值 -8 加上它的预测值 128 就得到它的真实值 120。重复这样的解码过程, 直到处理完所有的位流, 完成解码

过程。

### 3.S+P 变换方法

我们现在提出 S (Sequentail) 变换, 它类似于 Harr 多分辨率图象表示, 关于 S 变换存在各种各样的定义方法, 但大多数只在具体实现的细节方面略有差别。

对于一个整数序列  $c[n], n=0, \dots, N-1$ , 其中为偶数, 它可以用两个序列表示:

$$l[n] = \lfloor c[2n] + c[2n+1] \rfloor / 2, n=0, \dots, N/2-1$$

$$h[n] = c[2n] - c[2n+1], n=0, \dots, N/2-1 \quad (1)$$

其中  $\lfloor \cdot \rfloor$  表示向下截断。序列  $l[n]$  和  $h[n]$  形成  $c[n]$  的 S 变换。由于两个整数的和与差要么是两个奇数要么是两个偶数, 截断用来消除最无意义的位所带来的冗余, 除法和向下截断可以通过一个简单的移位操作来完成, 同时  $l[n]$  和  $h[n]$  可以使用  $c[n]$  所占用的内存空间。

逆变换为:

$$c[2n] = l[n] + \lfloor (h[n] + 1) / 2 \rfloor,$$

$$c[2n+1] = l[2n] - \lfloor h[n] / 2 \rfloor. \quad (2)$$

二维变换可以通过利用变换公式(1)对图象的行和列分别进行一维变换来实现, 如图2所示。

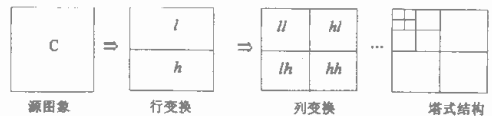


图2 图象的 S 变换

图2中 ll 对应的系数是 2x2 象素块的均值, 它们能够以原分辨率的一半来形成另外一幅图象。同样的变换方法可以用于这些降低了分辨率的均值图象来形成一个分层的金字塔结构。注意到用来表示 ll 图象中每个象素所需最大位数并没有随着每次变换而改变。例如如果原始灰度图象每个象素的精度为 8bpp, 则 ll 图象也有 8bpp。另一方面, 其他象素却需要有更多的位数来表示一个有符号的数。除了(1)和(2)式中的截断以外, 这种变换方法就相当于信号的子带分解<sup>[2]</sup>和图象的金字塔分解<sup>[3]</sup>。

S 变换非常简单, 计算效率高, 而且在一定程度上减小了图象的一阶熵。然而, 它在高通分量中残留了一定的相关性, 这种相关性是由于原始图象低频分量混叠产生的。因此, 我们可以使用更好的滤波器在更大范围内消除图象的相关性。然而整数的算术操作产生了对最低有效位统计上的依赖性, 这对于有损压缩而言不是个问

题,但对于无失真的图象压缩,这个问题必须解决。这意味着对于无失真图象压缩我们必须注意截断问题。

为了解决这个问题,我们可以利用这样一个事实:预测编码对于完全的重构不必是线性的。因此预测值能够被截断为一个整数。从而,我们可以用预测编码来改进S变换。然而,我们在一维变换中使用S+P变换(S变换+预测),而不是在最后的金字塔中进行预测, $l[n]$ 和 $h[n]$ 的一些值用来估计给定的 $h[n_0]$ 的值。将估计值记为 $\hat{h}[n]$ ,差值

$$h_d[n] = h[n] - \lfloor \hat{h}[n] + 1/2 \rfloor, n = 0, 1, \dots, N/2 - 1, (3)$$

取代 $h[n]$ ,形成了一幅新的经过变换的图象,它具有较小的一阶熵。我们不从序列 $l[n]$ 中抽取估计值,因为 $l[n]$ 形成降低分辨率的图象,而这幅图象稍后将用同样的方法进行变换。

定义:

$$\Delta l[n] = l[n-1] - l[n], (4)$$

其估计的一般形式为:

$$\hat{h}[n] = \sum_{i=-L_1}^{L_2} \alpha_i \Delta l[n+i] - \sum_{j=1}^H \beta_j h[n+j]. (5)$$

我们用 $\Delta l[n]$ 取代 $l[n]$ 使之具有零均值估计,这样就没有必要从 $c[n]$ 中减去均值。由于 $l[n]$ 没有被预测误差所取代,因此索引 $i$ 可以为负值。为了简化说明,目前可以不考虑图象的边界问题。

在一维的逆变换中,预测值按照逆序加上

$$h[n] = h_d[n] + \lfloor \hat{h}[n] + 1/2 \rfloor, n = N/2 - 1, N/2 - 2, \dots, 0 (6)$$

这样为计算当前 $n$ 的预测值所需的 $h[n]$ 已经复原了。在序列 $h[n]$ 复原之后,一维逆S变换公式(2)则可以进行计算。二维S+P变换可以通过先后对图象的行和列进行一维S+P变换来完成。然而应该注意到由于截断的原因,(3)式是非线性的,这就使得变换的顺序很重要。比如说,如果首先对图象的列进行变换再对行进行变换,则逆变换应先进行变换再进行列变换。

在(5)式中预测系数 $\alpha_i$ 和 $\beta_j$ 的确定有三种方案:熵最小,方差最小和频域设计。选择系数使得一阶熵最小可以参考Nelder-Mead方法[4],但这种方法对于实际应用来说其计算量太大,只能作为一种指标来评价其他方法。选择系数使得 $h_d[n]$ 的方差最小可以通过求解Yule-Walker方程组[5]。但是,我们没有必要使得经过S+P变换的图象熵最小,数值结果表明即使使用基于最小方差策略的高度自适应预测器方法,它的性能还不如在频域中设计固定的预测器,下面就说明这种方法。

如果不考虑截断,联合(1),(3),(4)和(5),我们将 $h_d[2n]$ 可以看作作为一个输入序列 $c[n]$ 经过一个非因果FIR滤波器所得到的输出,然后用2合1子抽样得到的结果。该滤波器响应的z变换是:

$$F(z) = (z-1) \left[ \frac{(z^{-1}+1)^2}{2} \sum_{i=-L_1}^{L_2} \alpha_i z^{2i} - \sum_{j=1}^H \beta_j z^{2j} - 1 \right]. (7)$$

预测编码具有非因果响应是不寻常的,正如前面解释的,由于 $l[n]$ 的值没有被预测值所取代,因此这种非因果响应是有可能的。

表3 S+P变换的预测器系数

预测器	参数			
	$\alpha_{-1}$	$\alpha_0$	$\alpha_1$	$\beta$
A	0	1/4	1/4	0
B	0	2/8	3/8	2/8
C	-1/16	4/16	8/16	6/16

在图象经过S变换后,其图象大部分的能量集中在低频部分,为了减少 $h_d[n]$ 的方差,我们应选择一个在低频部分快速衰减的滤波器。然而,由于公式(7)的结构和逆变换的需求,低频区域的迅速衰减不可避免地导致高频区域具有较大的增益。在理论上最佳预测器的选择依赖于图象本身的特点:光滑、无噪声的图象应使用在低频部分衰减较快的滤波器,这样可以使图象的压缩效果较好。而对于含有噪声并且细节信号比较多的图象,则宜用高频区域具有较低增益的滤波器。但是根据观察,熵对预测器参数的选择不是很敏感,而且预测器的参数也不是唯一的。也就是说,存在通用的预测器对大多数图象(如肖像,风景,医疗图象等)效率都比较高。经过对不同类型的图象进行广泛的测试,我们选择两组预测器,其系数在表3中给出。预测器A计算复杂度最小,B用于自然图象,而C用于非常光滑的医疗图象。它们能表现出很好的性能,而且在计算上可以用移位来代替乘法或除法。例如预测器B可以用下式来计算:

$$\hat{h}[n] = \frac{1}{8} \{ 2(\Delta l[n] + \Delta l[n+1] - h[n+1]) + \Delta l[n+1] \}. (8)$$

在图象的边界上,可以使用预测器:

$$\hat{h}[0] = \Delta l[1]/4, \hat{h}[N/2-1] = \Delta l[N/2-1]/4. (9)$$

渐进分辨率的传输策略能够很容易用多分辨率变换来实现,这是因为编码器是从金字塔的最高层开始编码。

译码器在接收了从顶层到1层的数据后,就能恢复一幅尺寸比原始图象小 $2^l$ 的图象。

为了对S+P变换后的系数进行熵编码,我们可以利用这样一个事实:变换后图象的象素之间存在统计上的依赖,而且这种依赖性是不可能由线性预测编码来进一步的减少。事实上,我们也必须注意到编码方法的复杂度,特别是变换后的图象的某些成分是不可能有效地压缩,因此也许需要不加编码直接传送。

在JPEG方法中,整数值被分解成三个部分:bits的长度,符号和幅度差值。幅度差值是实际幅值与一个特定的预定义集合的下限的差值。长度是指用来表达符号和幅度差值所需位数,它经过熵编码形成可变长码(VLC)。符号和幅度差值不进行编码而以可变长整数(VLI)的形式来传送。由于VLI没有经过熵编码,因此对编码效率会有一定影响。但它也有个优点:VLC符号的数量较少,简化了熵编码过程。也就是说,使用这种表示方法,所获得的位率接近于对整数进行熵编码所得到的位率,但复杂度却得以减小。

表4 幅度集变长整数表示的定义

幅度集合(MS)	幅度区间	符号位	幅度位数
0	{0}	无	0
1	[-1,1]	有	0
2	[-2,2]	有	0
3	[-3,3]	有	0
4	[-5,-4],[4,5]	有	1
5	[-7,-6],[6,7]	有	1
6	[-11,-8],[8,11]	有	2
7	[-15,-12],[12,15]	有	2
8	[-23,-16],[16,23]	有	3
9	[-31,-24],[24,31]	有	3
⋮	⋮	⋮	⋮

我们使用同样的方法来对S+P变换的结果进行熵编码。然而,为了减少部分数据没有经过编码所带来的影响,我们推荐一个稍微复杂一点的整数表示方法,见表4。在这种表示方法中,幅度集合索引首先进行熵编码,然后紧跟着符号位和幅度差值。例如,数字15和-16可以分别以序列(7,+3)和(8,-0)来进行传输。

#### 4. 实验结果分析(略)

#### 5. 结论

本文详细描述了两种无失真静态图象编码方法,即ISO JPEG静态无失真图象编码和无失真S+P图象编码。尽管JPEG静态无失真图象编码标准的制定已经有很多年了,但是对这种类型的编码方案的介绍却很少,因此本文从应用的角度出发,给出了这类编码方法的详细介绍,同时也给出了标准图象的编码结果和分析,在实际应用中JPEG所定义的七种预测器可以自动选取,这种自动选取需要建立一个频率统计表来计数每一不同显示时的符号数。从实验结果上来看,二维预测器要比一维预测器的图象编码的结果要好。S+P算法是目前国际上出现的一种新的无失真图象编码方法,这种图象编码的新颖之处在于引入了变换编码的思想,而通常变换编码是用于有损的图象编码。此外S+P算法中的变换源于子带编码和金字塔编码,因此这种算法可以在更大范围内消除图象的冗余度,从实验结果上来看,这种新的图象编码优于JPEG无失真图象编码方案,同时这种编码算法也可以用于有损的图象编码中,是一种很有前途的图象编码方法。

致谢:本文得到中国科学院软件研究所杨学平研究员和王裕国研究员的悉心指导和有益的讨论,作者深表谢意。

#### 参考文献

- [1] ISO/IEC Information technology - Digital compression and coding of continuous-tone still image. ISO/IEC DIS 10918-1.
- [2] J. W. Woods et al, Subband Coding of Images, IEEE Transaction on Acoustics Speech and Signal Processing, 1986, 34(5):1278-1288.
- [3] P. J. Burt et al, The Laplacian Pyramid as a Compact Image Code, IEEE Transaction on Communications, 1983, 31(4):532-540.
- [4] W. H. Press et, Numerical Recipes: the Art of Scientific Programming, Cambridge University Press, Cambridge, New York, 1996.

(来稿时间:1998年5月)

(编者注:原文篇幅较长,本刊作了部份删略,读者若有需要可与作者联系。)