

数据多次分组的图像无损压缩算法^①

高 健 陈 耀 刘 旦 (上海大学 自动化系 上海 200072)

摘 要: 对经典的图像无损压缩方法进行了比较全面的分析和研究。提出一种利用图像相邻像素之间的相关性对数据进行多次分组压缩的图像无损压缩算法,并提出了四种处理相同像素值或相近像素值的 RLE 改进算法和一种针对二值序列的长度编码表压缩算法,取得了较好的压缩效果。实验结果表明本算法的压缩比优于 WinZIP,与 WinRAR 基本相当。

关键词: 图像无损压缩; 数据多次分组; RLE; 压缩比; WinZIP; WinRAR

Image Lossless Compression Algorithm Based on Data Multi-Packet

GAO Jian, CHEN Yao, LIU Dan

(Automation Department, Shanghai University, Shanghai 200072, China)

Abstract: Several fundamental image lossless compression methods are explained and researched comprehensively. With the relevance of adjacent pixel, an image lossless compression algorithm based on data multi-packet is presented, and four RLE improved algorithms for pixels which has same or similar gray value and a length code table compression algorithm for bi-level sequence are proposed in this paper, there is very good effect. Experimental results shows the algorithm has a better compression ratio than WinZIP, almost the same effect as WinRAR.

Keywords: image lossless compression; data multi-packet; RLE; compression ratio; WinZIP; WinRAR

近年来,随着现代通讯技术和多媒体业务的发展,需要大量地存储和传输各类图像。为了解决数字图像数据量巨大的问题,必须对图像数据进行有效的压缩。因此,数字图像压缩编码技术已经成为现代信息社会中的一个热点研究方向。

图像压缩方法可以分为两大类:无损压缩和有损压缩。无损压缩,就是对图像数据进行压缩编码后在解码时能够无失真的恢复,也即在压缩和解压缩过程中数据没有丝毫损失,是一个可逆过程。有损压缩,就是对图像数据进行压缩编码时有数据损失,解码时不能无失真的恢复原来的数据,是一个不可逆过程。

一般来说,无损压缩的压缩比较低,但在许多领域却是迫切需要的,例如,图像需要做进一步的处理(如从图像中抽取特殊信息),重复压缩/解压缩,图像的获取代价昂贵,或者提供图像的要求质量未知等,特别是在医学图像、航空图像、卫星图像、遥感图像、

图像归档、高精度图像分析等领域^[1],高压缩比、可靠性强、实时性好的无损压缩方法具有很大的研究意义和实用价值。

1 常见无损压缩编码方法

1.1 Huffman 编码

哈夫曼编码(Huffman encoding)是根据信源中各种符号出现的概率进行编码,出现概率越高,分配的码字越短,从而达到较少的平均码长。哈夫曼编码是接近于信源熵的编码方法,得到了广泛应用。但是哈夫曼编码要对原始数据扫描两遍,数据压缩和还原速度都较慢。

1.2 游程编码(RLE)^[2]

游程编码(run-length encoding)是把一串连续的重复值(如图像的像素值)用一个单独的值和一个计数值来取代。比如序列 aaabbbbbccccc, 它的游程

^① 收稿时间:2010-03-26;收到修改稿时间:2010-05-06

编码就是 3a6b4c。对有大面积连续阴影或者颜色相同子块的图像,使用这种方法实现简单,压缩效果很好。

1.3 LZW 编码

LZW 编码的基本思想是建立一个符号串表,将输入字符串编码成定长的码流输出(通常为 12 位),并在编码过程中动态生成串表,算法是自适应的。LZW 编码原理的一个重要特征是串表中的代码不仅能取代一串同值的数据,也能够代替一串不同值的数据。在此方面,LZW 压缩原理是优于 RLE 的。

1.4 算术编码

算术编码完全抛弃了用特殊字符代替输入字符的思想。在算术编码中,输入的字符信息用 0 到 1 之间的数字进行编码,它用到两个基本的参数:符号的频率及其编码间隔。对于输入的字符信息,算术编码后形成一个唯一的浮点数。算术编码的效率一般要优于哈夫曼编码,但实现要比哈夫曼编码复杂^[3]。

1.5 无损预测编码

预测编码是根据数据之间存在相关性的特点,利用前面的一个或多个数据对下一个数据进行预测,然后对实际值与其预测值之差(预测误差)进行编码压缩。由于图像数据的相关性较强,从而可以用较少的数据对其差值进行量化编码,实现数据压缩的目的。现在常用的方法是差分脉冲编码调制(DPCM)^[4]。

2 数据多次分组压缩思想及算法实现

2.1 数据多次分组压缩思想

一般来说,图像像素值并非随机排列,而呈现一定的规律:除图像景物的边界以外,在图像的任意一个局部内图像像素值是渐变的,即每一点像素的值同它周围的其他像素的值大小相同或相近。图像像素的像素值范围为 0~255,每个像素值用 8bit 二进制数表示。从位平面编码方法我们可以知道,原始图像数据可以分成 8 个位平面来表示,随着位平面由低到高,高位平面数值相同的几率也逐渐增大,也即图像表现出更多相关性。

原始图像数据进行压缩处理后,压缩后的图像数据之间的相关性会下降,不能直接再进行压缩,需要对数据进行处理以增加其相关性。如果把压缩后的数据中每个数据都分为高位数据和低位数据,也即对数据重新分组,那么根据各个位平面数据的规律,高位

数据之间的相关性会较好,从而可以再进行编码压缩,而低位数据相关性较差,不能再压缩,如此反复多次处理,直到数据不能再分组为止。这种数据处理方法可以极大的提高数据压缩率,是数据多次分组压缩算法的基本思想。

上面所说的图像数据有多种分组方式,设所处理数据位数为 n ,如果分组后高位数据为 n_1 位,则低位数据位数为 $n-n_1$, n_1 的取值为 $1\sim n-1$,易知,高位数据的相关性与 n_1 的大小成反比。实验证明,每次分组时低位数据取两位时高位数据的平均压缩效果较好。因此本算法是通过对数据进行多次分组(直到数据不能再进行分组为止),并且对每次分组后的高位数据进行编码的逐级压缩方法,每次分组时低位数据定为 2 位。例如对灰度图像来说,原始图像数据(8bit)进行分组后的高位数据为 6bit;对压缩后的 6bit 高位数据继续分组,其高位数据为 4bit;再对压缩后的 4bit 数据继续分组,其高位数据为 2bit,此时分组结束(每次分组时低位数据均为 2bit)。

根据图像数据特点和为了方便算法描述,定义如下四种图像数据类型:等值序列、差 1 序列、循环交错序列和孤立点。

(1) 等值序列:是指图像数据序列中满足像素值相同的子序列(长度 $L \geq 2$),如 {155,155,155,155,155,155};

(2) 差 1 序列:是指图像数据序列中满足相邻像素值差值为 1 或 -1 的子序列(长度 $L \geq 4$),如 {125,126,125,124,123,124};

(3) 循环交错序列:是指图像数据序列中满足像素值间隔相等且相邻数据差值为 1 的子序列(长度 $L \geq 4$ 且为偶数),如 {165,166,165,166,165,166};

(4) 孤立点:不属于前三种序列的像素点均称为孤立点,如 {121,130,124,125,118}。

针对图像数据中不同序列类型的特点,对应提出了单、双表示游程编码方法、交错游程、差 1 序列编码方法和长度编码表压缩方法。

2.2 单、双表示游程编码方法

在一定条件下,RLE 是一种压缩效率很高的图像压缩方法。根据图像数据中等值序列与孤立点的分布特点,提出了两种基于 RLE 的压缩方法——单、双表示游程编码方法。

2.2.1 双表示游程编码方法

设有连续等值序列 $g_k g_k \dots g_k$ ，长度为 $n(n \geq 2)$ ，其编码方法为 $g_k g_k^1 11 \dots 10$ ，其中 1 的个数为 $n-2$ 个，0 为截止符号，1 和 0 的个数和再加 1 等于等值序列的长度；对于孤立点 g_{k+1} ，其编码方法为 g_{k+1}^1 ，对应的长度编码为空。同样把原始数据序列编码表示成一个新像素值序列和对应的长度编码表，这就是双表示游程编码方法。

例如对像素值序列 {3,3,4,5,5,5,5,5,3} 进行单、双表示游程编码方法结果如表 1 所示。

表 1 单、双表示游程编码方法

	单表示游程	双表示游程
新像素值序列	3453	334553
长度编码表	1001111100	011110

从上面的定义可知，单、双表示游程编码方法都是把原始数据序列表示成一个新像素值序列和对应的长度编码表。但具体表示方法各有特点：(1)对某一具体的等值序列进行编码，前者只需要保留一个像素值和其长度编码；后者需要保留两个像素值和其长度编码。显然，此时单表示游程方法效率要高于双表示游程方法。(2)对某一图像数据序列进行编码，由于序列中存在孤立点，此时前者的新像素值序列需保留每个等值序列中的一个像素值和所有孤立点的值，长度编码表保留其对应的长度编码；后者的新像素值序列需要保留每个等值序列中的两个像素值和所有孤立点的值，长度编码表保留其对应的长度编码。此时两种编码方法的效率主要是由图像数据中等值序列与孤立点的分布特点决定的：当图像数据中等值序列较多，孤立点相对较少时，单表示游程方法效率高；反之，则双表示游程方法效率高。

总之，两种方法各有优势：前者的新像素值序列数据比后者要少，后者的长度编码表数据比前者少。所以在对图像数据序列进行编码时，具体选择哪种编码方法需要先进行分析判断。

2.2.2 交错游程和差 1 序列编码方法

交错游程编码方法与差 1 序列编码方法是分别针对图像像素值序列中的循环交错序列与差 1 序列提出的编码方法。

(1) 交错游程编码方法：设有循环交错序列 $g_k g_{k+1} \dots g_k g_{k+1}$ ，长度为 $n(n \geq 4$ 且为偶数)，其对应

编码方法为 $g_k g_k^1 11 \dots 10^1$ 或 $g_k g_k^1 11 \dots 10^0$ ，原始数据序列顺序表示成一个新像素值序列、对应的长度编码表和状态表。其中长度编码表中 1 的个数为 $n/2-1$ 个，0 为截止符号；状态表位值为 1，表示循环交错序列中 $g_k > g_{k+1}$ ，值为 0，则表示 $g_k < g_{k+1}$ 。这就是交错游程编码方法。

(2) 差 1 序列编码方法：设有差 1 序列 $g_k g_{k+1} g_{k+2} \dots g_{k+n}$ ，长度为 $n(n \geq 4)$ ，其对应的编码方法为 $g_k g_k^1 11 \dots 10^1 10 \dots 01$ ，原始数据序列顺序表示成一个新像素值序列、对应的长度编码表和状态表。其中长度编码表中 1 的个数为 $n-1$ 个，0 为截止符号；状态表记差 1 序列中每个值与其后一个值的差值，其长度为 $n-1$ ，若差值为 1，状态表位记 1，若差值为 -1，则记 0。这就是差 1 序列编码方法。

由定义可知，交错游程和差 1 序列编码方法使用时要求像素值序列中不存在等值序列(否则数据不能恢复)，所以一般只适用于单表示游程方法之后再对其新像素值序列进行二次编码压缩(因为用单表示游程方法后其编码表示中的新像素值序列必无等值序列)。

在数据多次分组压缩算法中，两种编码方法一般同时使用，一起编码时需要注意：(1)交错游程方法的压缩效率要高于差 1 序列方法，所以当差 1 序列中包含循环交错序列时，优先考虑使用交错游程方法。(2)为了减少码表数量，把两种方法各自的长度编码表合成一张表示，把长度编码按顺序记录即可。(3)由于两种编码方法的新像素值序列每次都是保存两个相同像素值，需要对其进行区分，否则无法解码。可以直接在交错游程方法的状态表中进行表示：第 1 位作为编码方法区分标志位，为 1 表示交错游程方法，为 0 表示差 1 序列方法；若第 1 位为 1，则需要再在后面加 1 位状态位，状态位为 1 表示循环交错序列中 $g_k > g_{k+1}$ ，为 0 则表示 $g_k < g_{k+1}$ ；若第 1 位为 0，则不需要在后面加状态位。所以对交错游程序列，交错游程方法的状态表中需要用 2 位表示，而差 1 序列只需 1 位即可。注意：差 1 序列方法表示的状态表不变。

例如对像素值序列 {4,3,2,3,2,3,2,7,5,4,3,4,5} 同时进行交错游程和差 1 序列编码方法结果如表 2 所示。

表2 交错游程与差1序列编码方法

	交错游程方法	差1序列方法
新像素值序列	433755	
长度编码表	11011110	
状态表	110	1100

2.2.3 长度编码表压缩方法

前面介绍的几种编码方法对图像数据进行编码表示后,都包含了一张长度编码表。长度编码表中数据(1bit)为二值序列,为了对二值序列进行有效压缩,提出了一种对长度编码表进行编码的二值序列升位压缩方法。

二值序列升位压缩方法原理:先把二值序列中每两位一组按顺序重新组合成一个新数据,处理后得到的新数据序列(2bit)大小为0~3;由于由于图像数据的相关性,长度编码表中存在较多的等值序列,所以新数据序列中各数值出现的概率差别较大,可以用 Huffman 编码对新数据序列进行处理,实现数据压缩。

编码时需要注意的是:如果二值序列数据个数为奇数,则在序列最后补1个0,使其变成偶数,同时记下二值数据个数的奇偶状态(解码时需要)。

对二值序列{1,1,0,0,0,1,1,1,1,1,0,0,1,1,1,1}进行升位处理得到的新数据序列为{3,0,1,3,3,0,3,3},对新数据序列进行 Huffman 编码后,只需要12bit就可以表示16bit原序列(序列越长,压缩效果越好)。

2.3 图像数据多次分组压缩算法实现

2.3.1 图像数据预处理

图像的信源数据不是互相独立的,是一种有记忆的信源,即相邻像素之间具有相关性。图像无损压缩的目的是要最大限度且无失真的去除相邻像素间的冗余度,为了提高图像压缩效率,先对图像数据进行预处理,即先对图像进行横向或纵向 Z 字形扫描。这里采用统计图像相邻像素点的平均方差作为判断图像像素相关性的标准:平均方差越小,表示图像像素相关性好,反之则相关性差。如果图像像素横向相关性比较好,采用横向 Z 字形扫描;如果图像像素纵向相关性比较好,则采用纵向 Z 字形扫描。

2.3.2 算法实现

基于上述编码方法,图像数据多次分组压缩算法的具体步骤为:

(1) 对图像进行 Z 字形扫描预处理,得到图像像素值序列。

(2) 判断图像数据(8bit)序列是否适合进行双表示游程编码处理,适合则采用双表示游程方法编码,否则不做处理。

(3) 对剩余的图像数据进行分组,得到高位数据(6bit)和低位数据(2bit)两组数据。对高位数据序列先进行单表示游程方法编码,再对编码后的新数据序列同时进行交错游程和差1序列方法编码;低位数据不作处理,直接保存。

(4) 重复步骤(3)。

(5) 对剩余的图像数据继续分组,得到高位数据(2bit)和低位数据(2bit)两组数据。对高位数据序列用双表示游程方法编码;低位数据不作处理,直接保存。

(6) 把前面步骤中保存的所有长度编码表用长度编码表压缩方法进行二次编码,至此算法结束。

图像数据多次分组压缩方法流程如图1所示:

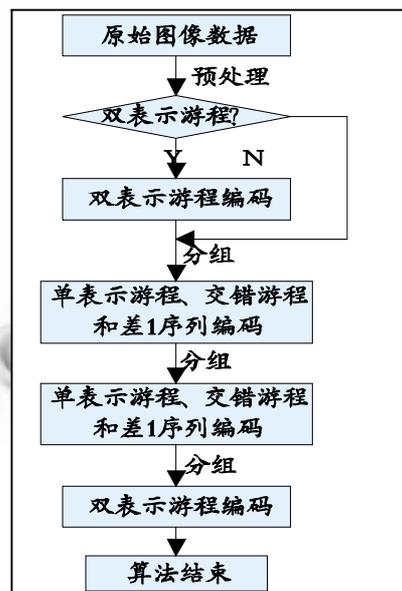


图1 图像数据分组压缩方法流程图

压缩文件只需要保存编码处理后的图像数据序列、每次分组时的低位数据和各算法中产生的编码表。根据这些数据和编码表可以逐级反向无失真的恢复原始图像数据,这里不再详细叙述。

3 实验结果及分析

对图像压缩质量的客观评价通常采用压缩比 C_r 和

峰值信噪比 PSNR 这两个指标来衡量。而无损压缩是没有 PSNR 这个指标的。

设图像的尺寸为 $M \times N$ 个像素, 每个像素为 B_p 比特, 压缩后的总比特数为 B_c 。

压缩比定义为: $B_r = B_p / B_c$

平均码长定义为: $B_r = B_p / C_r$

本文的实验结果均采用 B_r 表示, 单位是 **bpp**(bit per pixel)^[5]。

为了测试本算法的性能, 用 VC2008 对其进行了实现^[6]。并将结果与 WinZIP、WinRAR 两种压缩方法进行了比较, 测试对象是从数字图像经典测试图片中选择的四张图片, 规格为 256×256 的 8 位灰度连续色调图像。表 3 给出了具体的数据。

表 3 实验结果比较(bpp)

	WinZIP	WinRAR	Proposed
lena	6.57	5.29	5.17
camera	5.94	5.32	5.42
bookshelf	6.10	5.93	5.35
columbia	6.01	4.71	5.59
平均	6.16	5.31	5.38

实验得出以下结论与分析:

(1) 从表 3 实验数据可以看出, 对所有测试图片, 本算法压缩效果均优于 WinZIP, 压缩性能与 WinRAR 基本相当。

(2) 对比了压缩的有效性后, 来看压缩的复杂度(使用压缩时的 CPU 运算时间来衡量复杂度)。本算法在空域中对图像数据进行处理, 没有复杂的数学运算, 只是在数据分组时有整数乘法运算, 所选实验图像在编解码时每个像素值的平均乘法运算次数为 1.5 次, 压缩时的 CPU 运算时间均在 0.3s 左右, 与 WinZIP、WinRAR 的基本相当, 优于经典图像无损压缩算法, 所以本算法具有高效、高压缩比等特点。

(3) 图像数据预处理部分只是简单的横向或纵向 Z 字形扫描, 还可以进一步改进预处理方法: 可以先将原始图像进行分块, 然后再对每个子块进行横向或纵向 Z 字形扫描, 可以更加充分利用图像相邻像素

的相关性, 提高图像的压缩比, 当然这也会相应增加算法的复杂性。

(4) 低位数据部分由于其相关性低, 不能对其进行进一步的压缩处理, 是本算法的不足之处。如果能解决这一问题, 算法的压缩性能会得到很大提高。

4 结束语

图像无损压缩作为图像压缩的一个分支, 在很多领域具有不可替代的作用。通过对图像像素分布特性的研究并对传统的几种图像无损压缩方法进行了比较全面的分析后, 提出了基于数据分组的图像无损压缩算法, 取得了较好的压缩效果, 并与传统算法进行了比较, 文中算法提供了一个独特的思路, 并指出了改进的方向。

由于每种无损压缩都有自己的适用范围, 压缩比受不失真要求的限制, 真正意义上高压缩比的通用无损压缩算法目前还没有。因此在选用算法之前需要对图像数据进行分析, 使用时根据数据表现出的特点, 利用算法的思想, 灵活使用算法是提高压缩比的有效手段^[7]。我们相信, 随着研究工作的深入, 必将带来更大的技术突破。

参考文献

- 冯希. 几种图像无损压缩与编码方法的比较研究[硕士学位论文]. 北京: 中国科学院研究生院, 2008.
- 黄贤武, 王加俊, 李家华. 数字图像处理与压缩编码技术. 成都: 电子科技大学出版社, 2000. 211-253.
- 孟宪伟, 晏磊. 图像压缩编码方法综述. 影像技术, 2007, (1): 6-8.
- Rafael C. Gonzalez, Richard E. Woods 著, 阮秋琦, 阮宇智等译. 数字图像处理(第二版). 北京: 电子工业出版社. 2007: 326-364.
- 张晓玲. 图像无损压缩算法初步研究[硕士学位论文]. 北京: 北京工业大学, 2001.
- 王占全, 徐慧. 精通 visual c++ 数字图像处理技术与工程案例. 北京: 人民邮电出版社, 2009. 1-86.
- 籍俊伟. 无损图像压缩技术的研究与应用[硕士学位论文]. 北京: 北京化工大学, 2004.