

算术编码在分形图象压缩中的应用

王春梅 程乾生

(北京大学数学系, 北京 100871)

摘要 结合近年来对分形块编码的研究成果, 实现了其编码和解码过程; 并且通过研究存储数据之间的内在联系, 提出对不同的数据按不同的原则进行分类, 然后分别施行算术编码。分析数据及实验效果说明, 改进的编码算法在不影响信噪比的前提下得到了较好的压缩比。

关键词 分形块编码 二叉树分割 失真率 游程编码 算术编码

0 引言

自分形概念出现至今, 已有二十多年的历史, 而利用分形作图象压缩是在 1987 年由 Barnsley 首先提出来的^[1,2]。其基本思想就是寻找图象内部的自相似性, 并以一定的变换来模拟, 然后以变换的存储代替原图的存储; 解码时, 只需将变换重复作用于任一初始图, 利用迭代函数系的理论, 经过有限次迭代就可以得到原图的较好的近似。

这是一种完全崭新的压缩观点, 但 Barnsley 并未给出适于所有图象的自动生成变换算法。直到 1989 年, 他的学生 Jacquin 才在其博士论文中详尽阐述了一种自动生成分形变换的算法——分形块编码^[3]。该算法要寻找的是图象块之间的相似性, 它的提出为利用分形作图象压缩开辟了新的思路, 此后大量论文对它进行了改进^[4-6], 包括分块原则、分块形状、失真率计算和变换类选择等方面。

本文就是利用已有的研究成果, 提出了一些具体算法改进, 取得了较好的效果。

1 分形块编码及解码实现

利用分形作图象压缩的理论基础, Barnsley 在他的论文中已作了详尽的论述^[1,2,7], 集中体现在迭代函数系理论、粘贴定理和反迭代函数系理论。

分形块编码实现过程如下:

(1) 对待压缩图象进行分割;

(2) 对每一分割块(又称值域块), 按某种原则寻找匹配块(又称定义域块), 并建立最佳变换关系, 存储相应信息;

(3) 对相应的信息适当编码。

本文采用自适应二叉树分割算法^[5]分割图象, 具体采用三种图象块尺寸: 16×16 、 8×8 和 4×4 ; 也可以采用 32×32 和 2×2 的图象块, 但对每一 32×32 块的计算量过大, 而对 2×2 的图象块搜索过程较长, 所以均不宜采用。

两个图象块之间的变换关系 T 采用如下 3 类变换的复合:

$$T = ULS$$

其中 S 为一固定几何变换, L 为旋转变换(共 8 种), 而 U 为一类线性变换($Uf = af + b$), 其系数对比度 a 、亮度 b 待定。

两个图象块之间差异程度称为失真率, 其度量采用均方误差, 即

$$d(f, g) = \sum_{(i,j) \in R} (f(i, j) - g(i, j))^2$$

其中 $f(i, j)$ 为象素点 (i, j) 处的灰度值。这样在寻找最佳变换时, 可利用最小二乘法确定 a 、 b 的值, 使均方误差

$$d(f, g) = \sum_{(i,j) \in R} (f(i, j) - ag(i, j) - b)^2$$

最小。可分别对 a 、 b 给出一定的量化级别, 便于存储。

相对于编码过程,解码过程非常简单,具体如下:

- (1)任给一初始图象;
- (2)将编码图象所得信息恢复为相应的变换,按与编码过程中相同的二叉树搜索顺序作用到初始图象;
- (3)对(2)中所得图象实行平滑处理,并把该图象当作初始图象重复(2)、(3)操作,直到已迭代了一特定次数。

2 算术编码应用于分形块编码

经过分形块编码作用后,每一值域块需要存储的相关数据包括:值域块的大小、定义域块的位置(即其左下角坐标 (x, y))、变换 L 、对比度 a 、亮度 b ,我们还要对它们进行编码。

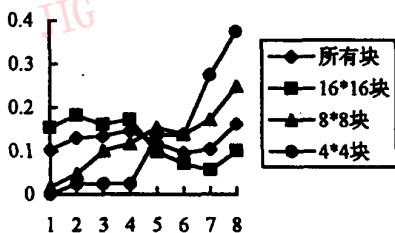
常用的编码算法有:

(1)以固定字长来存储;如值域块大小用2个比特,定义域块位置用14个比特,变换 L 用3个比特,对比度 a 用3个比特,亮度 b 用7个比特。

(2)利用熵编码对各类信息分别编码;

当然,利用上面两种算法就可以去除一定的冗余,得到较好的压缩比。

通过进一步研究这些数据,我们发现:

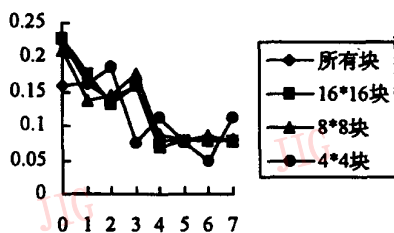


水平方向是对比度取值, 垂直方向为其分布
图1 值域块大小对对比度取值的影响

由信息熵的定义可知,分布曲线变化越缓和,所给信息的不确定程度越高,相应地熵就越大,从而编码时所需比特数就越多;分布曲线变化越剧烈,所给信息的不确定程度越低,熵就越小,从而编码时所需比特数就越少。因此,由图1可以得出结论,把所有块的对比值同时编码,比按块尺寸不同分别编码对比值需要更多的空间;类似地,对于定义域块的位置及亮度,也应按块尺寸不同分别编码。而对变换 L 来说,使用两种编码方式有差不多的效果。

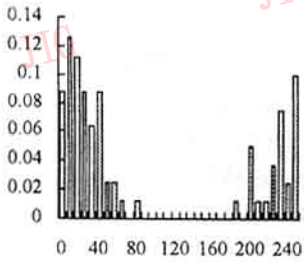
(1)值域块大小易采用游程编码。因为本文采用自适应二叉树分割算法,且各分割块信息是按照二叉树搜索顺序进行存储的;又相邻块之间相关性较大,因此相邻的值域块大小信息出现相同的概率较大,易采用游程编码。

(2)值域块的大小分别与定义域块的位置、变换 L 、对比度 a 、亮度 b 之间的关系;经过多次实验,我们发现它们之间并不是全无关联,而是有内在的联系。图1给出了值域块大小对对比度的影响。在我们的实验中,对比度 a 取8种值,值域块大小取 16×16 、 8×8 、 4×4 ;4条曲线分别为考虑所有值域块、只考虑 16×16 块、只考虑 8×8 块、只考虑 4×4 块时,对比度8种值的分布(即出现的概率)情况。不难看出,分块考虑时对比度分布曲线变化更剧烈,不分块考虑时对比度分布曲线较和缓。图2给出了值域块大小对变换 L 的影响。其中变换 L 取8种值,值域块大小取 16×16 、 8×8 、 4×4 ;四条曲线分别为考虑所有值域块、只考虑 16×16 块、只考虑 8×8 块、只考虑 4×4 块时,变换 L 8种值的分布(即出现的概率)情况。而与图1中情况不同,此时4条分布曲线变化程度都差不多。多次实验表明,值域块大小对其他数据(定义块的位置和亮度 b)的影响与其对对比度 a 的影响类似。

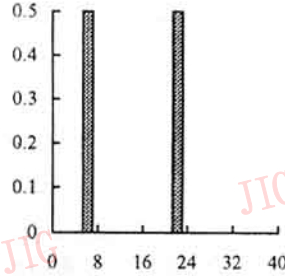


水平方向是变换L取值, 垂直方向为其分布
图2 值域块大小对变换L的影响

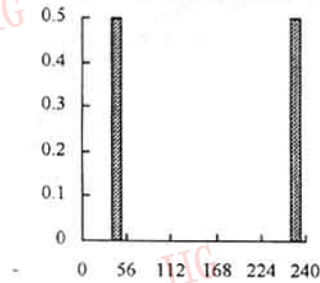
(3)值域块大小一定的条件下,对比度 a 对亮度 b 的影响;图3给出了某次实验中,当值域块为 4×4 时它们之间的联系。实验中,对比度 a 取8种值,亮度 b 取0~255之间的值;此次实验中,对比度 a 取1的块不存在,图3的8种情况分别为不考虑对比度、对比度取2~8时亮度 b 的分布。类似地,按对比度值的不同分别来考虑,亮度 b 的分布曲线变化更剧烈,而混在一起时考虑,亮度 b 的分布曲线较缓和。因此,对亮度 b 编码时,不考虑对比度比考虑对比度时需要更多的字节。



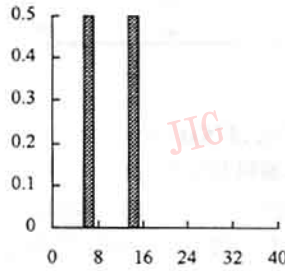
(a) 不考虑对比度 a 取值,所有亮度 b 的分布



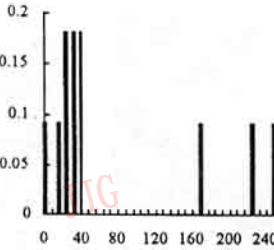
(b) 对比度 a 取 1 时,亮度 b 的分布



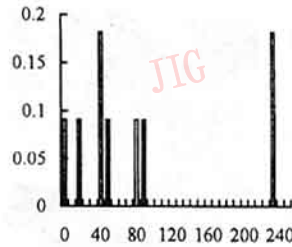
(c) 对比度 a 取 3 时,亮度 b 的分布



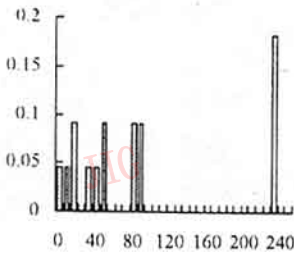
(d) 对比度 a 取 4 时,亮度 b 的分布



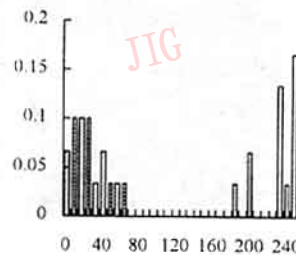
(e) 对比度 a 取 5 时,亮度 b 的分布



(f) 对比度 a 取 6 时,亮度 b 的分布



(g) 对比度 a 取 7 时,亮度 b 的分布



(h) 对比度 a 取 8 时,亮度 b 的分布

图 3 对比度 a 取值对亮度 b 的影响

由上面数据分析中的(2)、(3)可以知道,分形块编码得到的数据有着内在的联系。如果很好地利用种种联系,就能更好地去除冗余度。而作为作为一种很好的统计编码算法的算术编码,可以很容易地提取出数据间的关系,从而达到更好压缩比。

表 1 和表 2 给出了考虑某种联系和不考虑该联

系的情况下,分别施行算术编码后存储相应信息所需字节比较。二表中的数据验证了图 1、图 2 和图 3 中的情况:分布曲线变化剧烈的数据可以用更少的字节来存储。

表 1

存储信息	不考虑值域块大小的影响(bytes)	考虑值域块大小的影响(bytes)	提高百分比(%)
定义域块位置	1 022	926	9.73
变换 L	520	519	0.2
对比度 a	1 199	1 153	3.8
亮度 b	1 109	1 036	6.5

表 2

值域块尺寸(块)	存储亮度时不考虑对比度的影响(bytes)	存储亮度时考虑对比度的影响(bytes)	提高百分比(%)
16 × 16	749	635	15.22
8 × 8	354	295	16.67
4 × 4	54	40	25.9

3 图例与分析

利用以上提出的算法,我们就可以对任何一幅待压缩图象实施分形块编码和解码过程。但是编码过程中有一个很大的缺点,定义域块的个数很多,搜索过程中要做大量的数学运算,因此编码时间极为漫长。为此提出了很多聚类算法,搜索只对特征相似的定义域块进行,从而可以大大减少编码时间。

其中比较简单但十分有效的是 Fisher 提出的按均值和方差排序不同来分类的算法^[7]。

本文结合 1、2 节提出的算法和 Fisher 的聚类算法,对 4 幅标准图象 Lena、Zelda、Girl 和 Goldhill 实施了分形块编码和解码。这些图象尺寸均为 512 × 512,有 256 种灰度级,图 4 给出了原图和相应的解码图象。表 3 给出了图 4 中各解码图象的信噪比、压缩比以及相应的编码时间。本文中的实验均在 Pentium 166 微机上实现。

表 3 各解码图象(图 4 中)的信噪比、压缩比和编码时间

解码图	信噪比(PSNR)/dB	压缩比(CPR)	编码时间(s)
lena(a1)	30.73	63.15	134
lena(a2)	38.16	15.2	460
zelda(b1)	31.5	68.99	64
zelda(b2)	37.33	15.77	427
girl(c1)	30.97	36.3	116
girl(c2)	37.72	8.92	705
goldhill(d1)	27.96	41.24	115
goldhill(d2)	35.12	6.32	1033

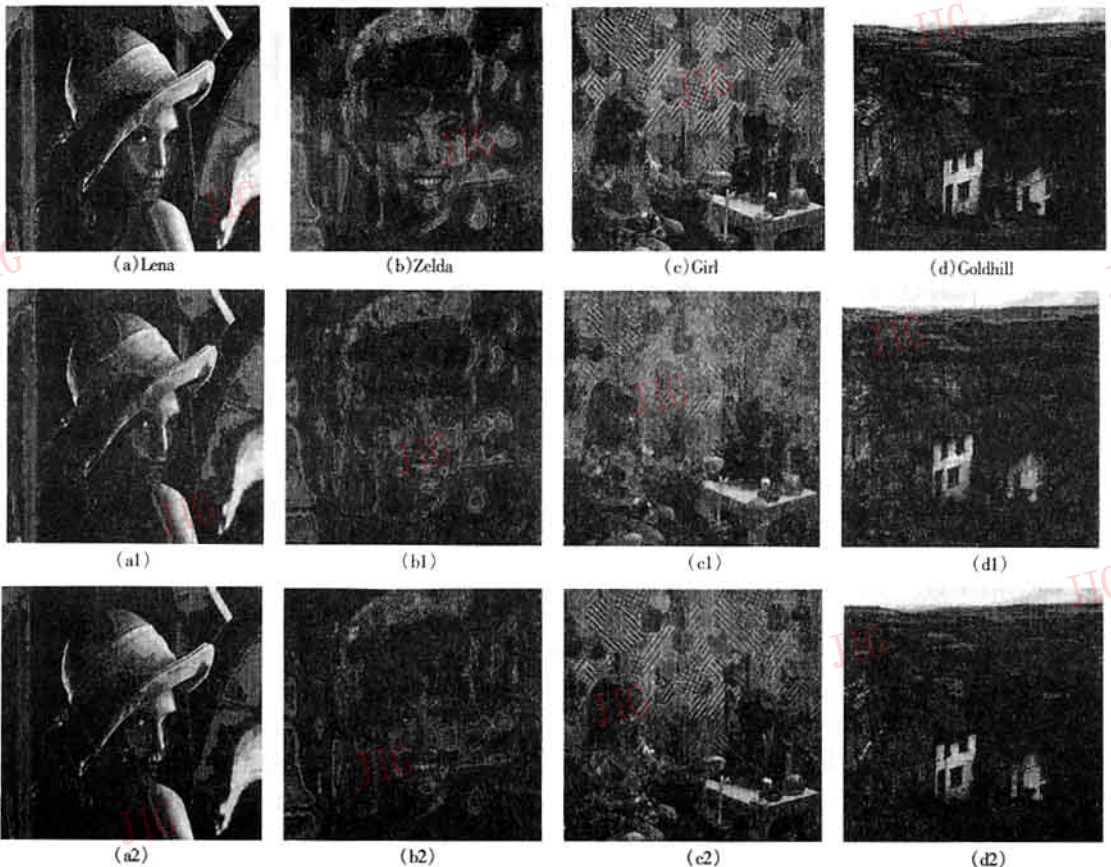


图 4 四幅标准图象及其相应的解码图象

从上面各幅解码图象的效果来看,分形块编码还是一种可行的图象压缩算法,但仍有许多有待解决的问题:

(1)由于实际上的分形图象形状极不规则,即相似性表现的形式多种多样,只利用规则的划分得到相似性,未免差强人意,需要更进一步讨论相似性的提取;

(2)用均方误差的形式作为失真率研究相似性很不全面,应给出更好地度量相似性的方法;

(3)搜索过程太复杂,应考虑如何利用图象块本身的特点,更快更好地寻找到匹配块,节省编码时间;

(4)分形块编码与其它图象压缩方法相结合的前景。



王春梅 1993年毕业于南开大学数学系,1996年获北京大学应用数学专业硕士学位,现在攻读博士学位,主要从事图象编码,模式识别等领域的研究工作。



程乾生 1963年毕业于北京大学数学系,现为北京大学数学科学学院信息科学系教授、博士生导师,研究领域为信号与信息处理,时间序列分析,模式识别,决策分析,信息融合。

参考文献

- 1 Barnsley M F, Demko S G. Iterated Function Schemes and the Global Construction of Fractals. Proc. R. Soc. Lond. A 399, 1985, 243 ~ 275.
- 2 Barnsley M F, Elton J H, Hardin D P. Recurrent Iterated Function Systems. Constr. Approx., 1989, 5(1): 3 ~ 31.
- 3 Jacquin A E. Fractal Image Coding: A Review. Proc. of the IEEE, 1993, 1451 ~ 1461.
- 4 Fisher Y. Fractal Image Compression. Fractals, 1994, (3): 347 ~ 366.
- 5 Fisher Y. Fractal Image Compression, in Chaos and Fractals: New Frontiers of Science. Peitgen H O, Jurgens H, saupe D, Springer-Verlag, New York, 1992.
- 6 Bedford T, Dekking F M. Fractal Coding of Monochrome Images. Signal Processing: Image communication, 1994, (6): 405 ~ 419.
- 7 Fisher Y, ed. Fractal Image Compression: Theory and Application. Springer-Verlag, New York, 1994.

Arithmetic Coding's Application in Fractal Image Compression

Wang Chunmei, Cheng Qiansheng

(Department of Mathematics, Peking University, Beijing 100871)

Abstract Fractal Image Compression is a valid algorithm. On the foundation of recent research about Fractal Block Coding, this paper completes both the encoding and decoding processes. Furthermore, by analyzing the relation between stored data, the authors first catalog different data according to different rules, then respectively apply Arithmetic Coding on them. The analyses and results show a better compression ratio on the basis of a fixed PSNR.

Keywords Fractal block coding, Quadtree partition, Distortion ratio, Run-length coding, Arithmetic coding