第 8 卷(A 版) 第 3 期
2003 年 3 月

中国图象图形学报
Journal of Image and Graphics

Vol. 8(A), No. 3
Mar. 2003

基于拼贴误差拟合(补偿)的分形压缩算法

贺 欣[1)] 魏小鹏[2)] 欧宗瑛[1)] 张建明[1)]

[1)](大连理工大学机械工程学院,大连 116023) [2)](大连大学先进设计技术中心,大连 116622)

摘 要 采用分形压缩算法来提高压缩速度和压缩比的一种有效途径是降低对拼贴误差的要求,然而这往往导致图象失真.为克服该问题,在经典分形压缩算法的基础上,提出了一种新的基于拼贴误差拟合(补偿)的分形压缩算法.该算法可以提供两种不同拼贴误差的分形编码,由于通过对不同区块使用与之相适应的相似变换类型,可以有效地补偿拼贴误差,从而可降低压缩图象的失真度.理论研究和实际应用表明,该算法为分形压缩图象所面临的失真度和压缩比的矛盾提供了一种有效的解决方案.

关键词 通信图象处理(510.4050) 分形压缩 拼贴误差 数据拟合

中图法分类号:TN919.81 **文献标识码**:A **文章编号**:1006-8961(2003)03-0352-04

A Fractal Compression Algorithm Based on Collage Distance Fitting and Compensating

HE Xin[1)], WEI Xiao-peng[2)], OU Zong-ying[1)], ZHANG Jian-ming[1)]

[1)](*School of Mechanical Engineering, Dalian University of Technology, Dalian* 116023)

[2)](*Advanced Design Technology Center, Dalian University, Dalian* 116622)

Abstract In most of the fractal image compression algorithms, the most efficient method to speed up coding process and increase compression ratio is to adopt bigger collage distance. However this method will result in poor image. To overcome the disadvantage, this paper has presented a new fractal compression algorithm based on collage distance fitting and compensating. This algorithm has provided two kinds of different compression codes, which contain their own unique collage distance sets and different types of similar transforms. By applying different types of similar transforms to different range blocks, the collage distance has been compensated, and the image quality and compression speed have been improved. In the end, an experiment of 'Lena' image has provided the application and demonstration of this algorithm. Theoretical analysis and application in practice have indicated that the proposed algorithm can be regarded as a suitable alternative to ordinary fractal compression method for balancing between the distortion measure and compression ratio.

Keywords Fractal compression, Collage distance, Data fitting

0 引 言

众所周知,对所有分形压缩算法的一个根本要求是拼贴误差尽可能地小.目前,为了达到这个目的,已研究出了许多算法,但如果仅以此为目的,则代价是很大的,因为当拼贴误差的阈值很小时,搜索符合要求的变换会变得很困难,且算法复杂,运算量大.例如,在 Pentium133 个人计算机上,压缩典型的 Lena 图象,如果不对搜索算法作任何特殊处理,那么就需要大约 1 小时.

经典的分形压缩方法是由 Arnaud Jacquin 最先提出的[1];随后,文献[2]在此基础上又做了一些改进.这些改进主要是针对以下几个方面进行的:(1)针对寻找区块的最佳匹配搜索方式的改进,如采用四叉树法、基于图形的统计不变性等方法来对域块进行各种方式的分类搜索,包括采用其他学科的各种技术(如人工智能、模式识别等技术)来帮助缩

基金项目:国家自然科学基金项目(69774030);高等学校骨干教师资助计算项目
收稿日期:2002-03-08;改回日期:2002-10-10

小搜索空间；(2)采用更为复杂的变换方式来提高搜索成功率，如采用基于多项式的分形压缩方法，但由于复杂的变换需要更多的参数，因此将导致压缩比的降低和计算量的提高；(3)针对分形压缩的方块现象而提出的可重叠区块编码方案，很明显，这种编码方案会导致重复编码及运算量提高.

当然，一个最直接的减少运算量、提高压缩速度和压缩比的办法是降低对拼贴误差的要求. 虽然这个办法对上述提到的各种技术都是适用的，可是采用这种方法的唯一缺点在于它会导致图象失真. 但可以采用某种适当的补救措施来解决这个问题.

1　经典的分形压缩技术

首先介绍分形压缩算法的基本原理：一幅图象可以看成是空间 $L^2(D)$ 的某一元素，D 是图象的矩形支撑. 分形压缩的目的在于寻找一个压缩($\|F(g)\| \le s\|g\|$ 对任何 s)映射 $F: L^2(D) \to L^2(D)$，并使其具有一个与图象 g 近似的不动"点" $\tilde{g}$

$$F(\tilde{g}) = \tilde{g}, \tilde{g} \approx g \tag{1}$$

由于映射 F 是压缩的，因此由压缩映射理论可以知道，它的不动点是唯一的，并且可以通过迭代的方法来得到这个不动点. 这个过程可以表示为

$$\lim_{n \to \infty} F^{(n)}(h) = \tilde{g}, \text{对于任何初始值 } h \in L^2(D) \tag{2}$$

式中，n 为迭代次数.

一般来讲，满足条件式(1)和式(2)的函数 F 是相当复杂的，因为要表示一个这样的函数并不一定比直接表示该函数所代表的图象简单，所以 Jacquin 和 Barnsley 的第 1 次研究使用了一种分片仿射变换的方法，它表示了相对于区块的基本相似变换. 虽然表达 F 所需要的参数相对较少，但其不动点 $\tilde{g}$ 却能表达自然图象及其细节. 分片仿射变换定义为

$$F_g(r) = s_i g \boldsymbol{T}_i^{-1}(r) + c_i, \text{对于任何 } r \in \boldsymbol{R}_i \tag{3}$$

其中：$\boldsymbol{T}_i: \boldsymbol{R}^2 \to \mathbf{R}^2$ 是相似变换阵；s_i 为灰度对比度调节量；c_i 为亮度偏移量.

$\boldsymbol{R}_i$ 为对支撑 $\boldsymbol{D}$ 进行分割得到的某一部分，叫做区块或子块，$\boldsymbol{D}_i = \boldsymbol{T}_i^{-1}(\boldsymbol{R}_i)$ 被叫做域块或父块. 父块和子块一般取为平行四边形.

式(3)表达了如下操作：

(1)在图象 g 中提取域块 $\boldsymbol{D}_i$；

(2)用 $\boldsymbol{T}_i$ 对域块 $\boldsymbol{D}_i$ 进行仿射变换；

(3)根据 s_i 和 c_i 来调整变换结果的亮度和对比度；

(4)用调整后的结果来代替在区块 $\boldsymbol{R}_i$ 上的那部分图象函数 $F(g)$.

由于 F 是分片定义的，并且可以通过迭代的方法来得到其不动点，因此它可以叫做迭代函数系(IFS).

如果从广义测度来理解，只要使 $|s_i| < 1$，即可满足 F 收敛的要求，并不需要变换矩阵 $\boldsymbol{T}_i$ 也收敛. 但在算法的实际使用上，一般取 $|\boldsymbol{T}_i| < 1$，这是与以下实验结果是一致的：对于绝大部分图象，当从大像素块到小像素块变换时，往往容易得到满足 $|\boldsymbol{T}_i| < 1$ 的结果. 另外，从简化的目的考虑，一般取 $\boldsymbol{T}_i$ 的旋转角度为 90°的倍数.

对于分形压缩，首先要回答的是：如何选择 $\boldsymbol{R}_i$，$\boldsymbol{T}_i$ 和 c_i？拼贴原理部分地回答了这个问题：

如果 $\|F(g) - g\| < \varepsilon$，则

$$\|\tilde{g} - g\| < \frac{\varepsilon}{1-s}, \text{其中}, s = \sup_{h \in L^2(D)} \frac{\|F(h)\|}{h} \tag{4}$$

根据拼贴原理，压缩映射的任务就将"寻找一个不动点接近于图象函数 F 的问题"转变为"寻找一个可以将图形映射到接近于其自身的函数"的问题. 其接近程度可以用拼贴距离 $\|F(g) - g\|$ 来表示，一般用均方差表示. 在算法中可通过如下步骤实现：

(1)将图象以某种方式分割成若干区块 $\{\boldsymbol{R}_i\}$；

(2)考察各种面积较大的域块 $\{\boldsymbol{D}_i\}$ 到区块的可能变换 $\{\boldsymbol{T}_i\}$. 具体地讲，对于每一个确定的 $\boldsymbol{T}_i$，首先计算在块 $\boldsymbol{R}_i$ 上，$\|F(g) - g\|$ 取最小值时的 $s \in (0,1)$ 和 c；然后在 $\boldsymbol{T}_i$ 中，找到最小的 $\|F(g) - g\|$ 取值及其对应的 s 和 c，并将它们保留在 $T_{j,i}$，c_i 和 s_i 中.

所有与 $\boldsymbol{R}_i$ 对应的 $T_{j,i}$、c_i 和 s_i 就构成分形压缩算法的输出 $\{T_{i,j}, c_i, s_i\}$，它们完全定义了 F，并可以根据压缩映射原理来重建图象：

$$g \approx \tilde{g} = \lim_{n \to \infty} F^{(n)}(h) \approx F^{(N)}(h) \tag{5}$$

式中，h 是某一初始元，一般取与图象相同的支持；N 是迭代次数，根据收敛速度，该值应取足够大，以满足精度要求.

2　基于拼贴误差拟合(补偿)的分形压缩算法

基于拼贴误差拟合(补偿)的分形压缩算法主要包括如下两个步骤：

(1) 与经典的分形压缩方法基本相同，经过这步的操作，实际已经得到了可以直接使用的高压缩

比的压缩编码,而不同的是,首先,采用的分片仿射中的亮度偏移量 c_i 等于 0,即在算法中,偏移量不被考虑,参见式(3),基本的相似变换为

$$F_g(r) = s_i \boldsymbol{g} \boldsymbol{T}_i^{-1}(r) + c_i, \text{对于任何 } r \in \boldsymbol{R}_i \tag{6}$$

很显然,这样会使匹配成功的可能性降低,并将增加搜索时间,但通过提高拼贴误差的阈值 $e_{\min 1}$,即降低对拼贴误差的要求,即可求得平衡;其次,与一般的分形压缩技术不同,本算法中,是根据另一个匹配误差的阈值 $e_{\min}$ 来保留一部分区块的拼贴误差信息 e_i,即当 $e_{\min} \leq e_i \leq e_{\min 1}$ 时,则保留该区块的拼贴误差信息 e_i 于编码中,这些信息将在算法的第 2 步中用到.在本步处理中,目前任何已经存在的分形算法都可以采用,而且由于省去了偏移量的计算和采用了较大拼贴误差阈值 $e_{\min 1}$,因此,编码过程会相当快.其唯一的不足之处是解码图象可能局部失真严重,但代码中却包含了这些局部失真严重地方的位置及失真度大小的信息,这些信息将会在算法的第 2 步中被用于对分形编码进行修正,以便对拼贴误差加以补偿;

(2) 找出在第 1 步编码中,保留有拼贴误差的区块,再根据区块的尺度,选择适用于该区块的基本相似变换:

$$F_g(r) = s_i \boldsymbol{g} \boldsymbol{T}_i^{-1}(r) + c_i + f_{e_i}(r), \text{对于任何 } r \in \boldsymbol{R}_i \tag{7}$$

其中,$f_{e_i}(r)$ 可以看成是拼贴误差 e_i 的拟合(补偿)函数,因此可以用拟合像素灰度误差数据的最小二乘法来计算出这个多项式[3].本步不存在匹配搜索计算,仅仅需要计算拟合多项式,并用它来取代第 1 步编码中保留的拼贴误差,其得到的代码将是经过修正的分形压缩代码.这种代码与目前所有的分形压缩算法得到的代码相比,其不同之处在于:对于每一个区块,所采用的基本相似变换类型是不同的,它反映了图象局部的自相似程度.

3 算法的实际运用结果

为了解本文算法的效果,采用 256×256 大小的 256 级标准 Lena 灰度图象进行了实验,实验采用 VisualC++ 6.0 编程,平台为 P300 联想微机 Windows98 操作系统,实验结果如图 1 所示.本实验中,算法第 2 步对所有的拼贴误差均采用如下同一个拟合函数:

$$f_{e_i}(r) = a_{1i}x + a_{2i}y + a_{3i}x^2 + a_{4i}y^2 + a_{5i}xy + a_{6i}, r = r_x, y = r_y \tag{8}$$

(a) 256×256 大小的原始 Lena 图象 (b) 算法第 1 步的压缩结果(压缩比为 31.5:1,均方差为 287.4) (c) 全部算法的压缩结果(压缩比为 15.6:1,均方差为 45.7)

图 1 实验结果

实验结果表明,本文算法的第 1 步可以实现较高的压缩比,但图象失真度高;而全部算法虽然降低了压缩比,但图象失真度低.

在相同的计算机上,采用经典的分形压缩算法(4×4 分块)搜索时间为 57s,而用本算法全部两部分搜索的消耗时间为 23s.由此可见,与经典的分形压缩算法相比较,本算法有效地降低了搜索时间.

4 结论和展望

综上所述,如果拟合补偿函数(偏移量)固定为零次多项式,则本文提出的算法与经典的分形压缩算法类似;如果拟合补偿函数固定为 n 次多项式($n>0, n \in \mathbf{N}$),则与基于多项式的分形压缩方法类

似[4]. 注意，这里只能认为是“类似”，因为本算法在计算最佳匹配时，始终没有考虑偏移量，而且在“近似”IFS 的编码中，也没有偏移量的信息. 这里偏移量的信息，或者称之为细节信息，是分开存储的. 本算法的第 1 步给出了图象在高压缩比下的近似分形压缩编码，虽失真度比较大，但可以单独使用，比如用于对图象数据的检索；第 2 步可以看成是对第 1 步得到的分形编码的误差修正，但它必须与第 1 步的代码一起使用，才可得到低失真的压缩图象.

在本文算法中，拟合函数类型的选择是最重要的，因为如果选择 n 次多项式，则多项式的次数应该使拼贴误差的大小和对应区块的尺度相适应，对于这个问题，还有待于进一步研究.

参 考 文 献

1 Jacquin A E. Image coding based on a fractal theory of iterated contractive image transformations systems [J]. IEEE Transaction on Image Processing, 1992, 1(1): 18～30.

2 Reusens E. Overlapped adaptive partitioning for image coding based on the theory of iterated functions systems [A]. In: Proceedings of International Conference on Acoustic, Speech and Signal Processing[C], Adelaide, Australia, 1994, 557～560.

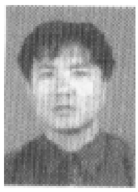
3 高望东，刘淑珍. 数值计算方法[M]. 大连：大连理工大学出版社，1992：211～220.

4 徐佩霞，孙功宪. 小波分析与应用实例[M]. 合肥：中国科技大学出版社，1996：138～156.

贺　欣　1966 年生，2002 年获大连理工大学工学博士学位. 主要研究领域为计算机图形学与图象处理.

魏小鹏　1959 年生，博士，大连大学教授，博士生导师. 主要研究领域为计算机图形学、图象处理和智能 CAD.

欧宗瑛　1939 年生，大连理工大学教授，博士生导师. 主要研究领域为计算机视觉和智能 CAD.



张建明　1973 年生，大连理工大学机械工程学院博士生. 主要研究领域为计算机图形学与图象处理.