

基于分数盒维数的快速分形图像编码

何传江 黄娟娟 李高平

(重庆大学数理学院, 重庆 400030)

摘要 分形图像编码是一种很有前途的压缩技术,但编码时间长阻碍了它的广泛应用。分形编码的时间主要花费于在一个海量码本中搜索每个输入子块的最佳匹配块。针对这个问题,提出了一种快速分形编码算法,它基于图像块的分数盒维数特征,能够在较小的搜索范围内完成输入子块的最佳匹配。实验结果显示,该算法能够大大缩短编码时间,同时实现和全搜索分形编码算法相同或更好的图像质量。

关键词 分形 分形图像编码 分形图像压缩 分数盒维数

中图分类号: TN919.81 文献标识码: A 文章编号: 1006-8961(2007)02-0277-06

Fast Fractal Image Coding Based on Fractional Box-counting Dimension

HE Chuan-jiang, HUANG Juan-juan, LI Gao-ping

(College of Mathematics and Physics, Chongqing University, Chongqing 400030)

Abstract Fractal image coding is a very promising compression technique, but it suffers from long encoding time. The time to encode fractally an image is mostly spent on searching for the best-matched block for each of range blocks in a usually-large domain pool. This paper thus proposed a fast fractal encoding algorithm based on fractional box-counting dimension of an image block, which can find out the best-matched block for an input range block with a reduced search. Experimental results show that the proposed algorithm can significantly shorten the encoding time, while achieving the same or better decoded image quality as baseline fractal algorithm with full search.

Keywords fractal, fractal image coding, fractal image compression, fractional box-counting dimension

1 引言

图像的分形编码是一种利用现实图像中不同区域间存在的跨尺度相似性(局部自相似性)来减少图像数据冗余的新型编码技术,它突破了以往图像编码的框架,用一个函数来描述整幅图像,巧妙地将分形的自相似性应用到图像编码中,取得了很好的压缩效果。高压缩比、分辨率无关性和解码快等优点使得它有很好的发展前途。但是计算复杂性高、编码时间长是它的显著缺点,这严重阻碍了分形编码的广泛应用^[1]。

构造分形码的时间主要花费于在海量码本中寻

找每个输入子块的最佳匹配块。尽管全搜索法能够得到最优结果,但这个过程的计算成本很高,这限制了基于全搜索法的分形编码的应用范围。目前,人们提出了许多加快分形编码的方法,新的方案仍在不断出现^[2-5]。因为分形编码本身是限失真编码方法,所以牺牲一定的图像质量以换取编码时间减少的做法成为快速分形编码研究的主流。在实时性要求高、质量次之的场合,这种快速算法有较好的应用前景。但是在许多图像质量要求高的应用场合,相对于全搜索基本算法而言保持图像质量不变的快速算法是必需的。尽管也有一些相对于全搜索基本算法而言保持图像质量不变的加快算法,但缺点是加快倍数不高。因此,在保持图像质量不变或更好的

收稿日期:2005-08-10;改回日期:2005-11-03

第一作者简介:何传江(1964~),男,教授,副院长。2004年获得重庆大学自动化学院工学博士学位。主要从事分形图像编码等研究。

E-mail:chuanjianghe@sina.com

前提下,实现更快的编码是分形图像编码的一个重要课题。

分形维数理论是分形理论中一个十分重要的组成部分。分形维数是表征分形集的一个重要参数,在图像处理中已得到广泛的应用,如边缘检测、图像分割、纹理识别等。其应用的基础是:(1)自然界中不同种类的形态一般具有不同的分形维数;(2)Pentland的假设^[6],即自然界中的分形与图像的灰度表示之间有着一定的对应关系。分形维数是独立于图像分辨率、视角而稳定存在的物质表示量。此外,图像 3 维表面的分形模型不仅可以用来获得形状信息,还可以用来区分平滑或粗糙的纹理区域。因此,分形维数把图像的空间信息与灰度信息有机地结合起来,为图像处理提供了新的工具。

分形维数的种类很多,常用的有 Hausdorff 维数、相似维数和盒维数。分数盒维数(fractional box-counting dimension)^[7,8]是 Feng 在博士论文^[7]中提出的,它是在普通盒维数的基础上发展而来的,可以得到更为精确的维数估计值。它较好地反映了图像灰度变化的复杂度,视觉上相似的两个子块,分数盒维数也是近似相等的;反之,分数盒维数相差很大的两个子块,在视觉上一般是不相似的。因此,在分形图像编码中,把分数盒维数作为子块匹配的特征量就是非常自然的想法了。

基于上述分数盒维数,提出一个快速分形编码算法。该算法把分数盒维数作为子块匹配的一个特征量,实验结果表明,这样做可以大量排除海量码本中不太可能匹配 R 块(range 块)的码块,从而大大减少了编码时间。

2 基本分形算法

在基本分形编码算法中,图像被分割成大小两类方块:range 块(R 块)和 domain 块(D 块),其中 R 块互不重叠且覆盖整幅图像,D 块可以重叠且边长通常为 R 块的两倍。D 块经 4-邻域像素平均或间距采样(decimation)收缩为 R 块的大小,这种子块的全体就构成码本(记为 Ω)。为改进图像质量,一般还要对码块 $D \in \Omega$ 施行 8 个等距变换(旋转与反射)。

编码阶段,对于每个 R 块 R_i ,为了寻求其最佳匹配块,需要求解下面的极小化问题:

$$\|R_i - (s_i D_{m(i)} + o_i I)\| = \min_{D \in \Omega} \left\{ \min_{\substack{s_i, o_i \in \mathbf{R} \\ |s_i| < 1}} \|R_i - (sD + oI)\| \right\} \quad (1)$$

其中, \mathbf{R} 是实数集, $\|\cdot\|$ 是向量 2-范数, $m(i)$ 表示 R_i 的最佳匹配块的序号,要求对比度因子满足约束 $|s| < 1$ 是为了理论上保证解码迭代序列收敛。问题式(1)的内层是约束极小化问题,为了减少计算复杂性,通常的做法是先忽略约束 $|s| < 1$,求解最小值问题:

$$E(R_i, D) = \min_{s, o \in \mathbf{R}} \|R_i - (sD + oI)\| \quad (2)$$

(记解为 s_i, o_i) 然后对不满足约束的 s_i 作截断处理。接着用全搜索方法求解问题式(1)的外层极小化问题:

$$E(R_i, D_{m(i)}) = \min_{D \in \Omega} E(R_i, D) \quad (3)$$

三元组 $(m(i), s_i, o_i)$ 就构成 R_i 的分形码,其中 s_i, o_i 是 s_i, o_i 的量化值。如果考虑等距变换,则分形码中还应包括等距变换序号。全体 R_i 的分形码就组成原始图像的分形码,它描述了一个使图像近似不变的压缩变换。

解码是相对简单的迭代过程,由分形码描述的压缩变换迭代作用于任何初始图像来完成。

3 快速编码算法

3.1 分数盒维数

设 A 是欧氏空间 \mathbf{R}^n 的一个集合,并设 A 被一个同维数的超方体所包围。设 q 是正整数,定义 $\rho_0 = 2^{-q}$ 为分辨率,且 $\rho(n) = 2^{-nq} = 2^n \rho_0$ 是计算尺度。用一组边长为 $\rho(n)$ 的盒子覆盖集合 A ,然后把每个边长为 $\rho(n)$ 的盒子进一步细分成边长为 ρ_0 的盒子。注意,每个边长为 $\rho(n)$ 的盒子由 2^{qn} 个边长为 ρ_0 的盒子组成。分别用 $N_n(A)$ 和 $N_0(A)$ 表示与 A 相交的边长为 $\rho(n)$ 和 ρ_0 的盒子的数目。如果

$$D' = \lim_{n \rightarrow \infty} \{ \ln(N_n(A)) / \ln(1/\rho_0) \} \quad (4)$$

存在,则称 D' 为 A 的分数盒维数^[7]。

下面讨论如何计算一幅图像的分数盒维数。

(1) 盒子内部分析(Intra-box Analysis)

在这个阶段,需要分析每个盒子内部的变化,然后计算出所需盒子的总数。以一个 2D 图像(大小为 $M \times M$)为例,图像的灰度视为第 3 维。

第 1 步,用大小为 $S \times S$ 的正方形格子划分图像,其中, S 是预设的分辨率,依据不同的应用需要,它可以取 2, 3 或 4 个像素。因此,图像的每边都被分割成 $\lfloor M/S \rfloor$ 个部分。如果图像的最大灰度是 I_c ,那么作为第 3 维的灰度也分为 $\lfloor M/S \rfloor$ 个部分。于

是,共得到 $\lfloor M/S \rfloor \times \lfloor M/S \rfloor \times \lfloor M/S \rfloor$ 个盒子。令 J 表示盒子的高度,那么

$$J = I_c / \lfloor M/S \rfloor$$

这样每个盒子的大小是 $S \times S \times J$ 。

第 2 步,得到每个小方格的灰度最大值和最小值,然后计算那些小格子所需要的盒子数。对每个小方格 (i, j) :

$$N_{\text{intra-box}} = \sum_{\text{all cells}} n(i, j)$$

$$n(i, j) = \lceil (I_{\text{max}}(i, j) - I_{\text{min}}(i, j)) / J \rceil$$

注意,盒子的大小是 $S \times S \times J$,而 $n(i, j)$ 是覆盖网格 (i, j) 处的灰度表面所需的盒子数。

第 3 步,把所有网格的小盒子的数量加起来,就得到整幅图像内部所需要的盒子数:

$$N_{\text{intra-box}} = \sum_{\text{all cells}} n(i, j)$$

(2) 盒子之间的分析 (Inter-box Analysis)

在普通盒维数的算法中,盒子之间的变化和小尺度下低估盒子数的结果被忽略了,而在分数盒维数算法中,把盒子之间的变化也考虑在内。

第 1 步,建立直接和对角相邻关系,如图 1 所示。有 4 个直接的相邻盒子:

I - II, II - IV, III - IV, I - III

另有两个对角相邻盒子:

I - IV, II - III

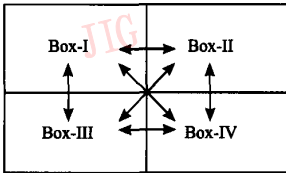


图 1 盒子间变化分析
Fig. 1 Inter-box analysis

第 2 步,分析邻近盒子间的变化范围。这可以通过比较图 1 所示的几组相邻的盒子的变化范围来确定。定义每个盒子的变化范围为

$$R_i = [I_{\text{min}}(i), I_{\text{max}}(i)]$$

其中, $i = \text{I}, \text{II}, \text{III}, \text{IV}$ 。这里的目标是计算出这两个范围之间的差别。规则如下:令 R_i 和 R_j 表示相邻盒子的变化范围。

(1) 如果 $R_i \cap R_j \neq \emptyset$,那么没有差别被考虑,不需要盒子。

$$R_{\text{gap}} = [I_{\text{gap-min}}, I_{\text{gap-max}}] = [0, 0]$$

(2) 如果 $R_i \cap R_j = \emptyset$ 且 $R_i < R_j$,那么 R_i 和 R_j

的变化范围 $R_{\text{gap}} = [I_{\text{gap-min}}, I_{\text{gap-max}}]$,其中

$$I_{\text{gap-max}} = \min(R_j), I_{\text{gap-min}} = \max(R_i)$$

第 3 步,分析垂直、水平和对角相邻的盒子之间的变化量 V 。在垂直与水平相邻盒子之间,计算所需盒子数和盒子内部方法相同,要注意的是,如果对角相邻盒子中的灰度差产生“粗糙的斑点”(rough spot),则指定一个盒子覆盖这个“粗糙的斑点”。

第 4 步,得到覆盖这些有盒子间差异的盒子总数如下:

$$N_{\text{inter-box}} = N_{\text{inter-box-vertical-horizontal}} + N_{\text{inter-box-diagonal}}$$

所以,需要的盒子的总数是

$$N = N_{\text{intra-box}} + N_{\text{inter-box}}$$

最后,在双对数坐标系中,对数据组 $((\ln(M/\rho_0), \ln(N(0)))$ 进行线性回归,得到一条固定的直线,其斜率就是前面定义的分数盒维数 D' 。

3.2 算法分析与描述

首先给出一个熟知的结果,它被许多文献从不同的角度引用(例如文献[3]、[5])。

定理 对于 $R, D \in \mathbb{R}^{n \times n}$, 设最小值问题

$$E(R, D) = \min_{p, q \in \mathbb{R}^n} \|R - pD - qI\|$$

的解为 s, o , 则有

$$\begin{cases} s = \frac{\langle R - \bar{R}I, D - \bar{D}I \rangle}{\|D - \bar{D}I\|^2} \\ o = \bar{R} - s\bar{D} \end{cases} \quad (5)$$

$$E(R, D)^2 = \|R - \bar{R}I\|^2 - s^2 \|D - \bar{D}I\|^2 \quad (6)$$

其中, \langle, \rangle 表示欧氏内积, \bar{R}, \bar{D} 表示均值。

本文提出的算法基于这样的假设:如果 D 块 D 和 R 块 R 的分数盒维数相差大,那么 D 不能匹配 R 。换言之,如果 D 匹配 R ,则 D 和 R 的分数盒维数相差不大。在这样的假设下, R 块 R 的最佳匹配 D 块 D 在分数盒维数意义下一定是 R 的近邻。因此,把码本 Ω 按分数盒维数大小赋予升序结构,然后使用二分搜索法,在赋序码本 Ω 中对在分数盒维数意义下与输入子块 R 最接近的码块 D 的 k 邻域内再进行匹配搜索。

其次,根据式(6),有

$$\begin{aligned} E(R, D)^2 &= \|R - \bar{R}I\|^2 - s^2 \|D - \bar{D}I\|^2 \\ &= \|R - \bar{R}I\|^2 \left(1 - \frac{\langle R - \bar{R}I, D - \bar{D}I \rangle^2}{\|R - \bar{R}I\|^2 \cdot \|D - \bar{D}I\|^2} \right) \\ &\leq \|R - \bar{R}I\|^2 = (n\sigma_R)^2 \end{aligned}$$

上式表明,如果子块 R 的标准差 σ_R 很小,那么对于任意子块 $D \in \mathbb{R}^{n \times n}$, $E(R, D)$ 也小,因此任意子

块 D 匹配 R 的误差都小。于是,对于标准差足够小的输入子块 R ,任意子块 $D \in \mathbf{R}^{n \times n}$ 都可以作为其匹配块。但是,标准差足够小的子块 R 的亮度变化很小,这样的子块通常是图像的平滑区域,可近似看成常值块,用其均值版本 $\bar{R} \cdot I$ 近似最合适。于是,对于标准差足够小的输入子块 R ,无需再搜索其匹配块,从而可以进一步加快编码。

前面已经提到,解码图像质量依赖于对不满足约束的对比度因子 s 做截断处理的数目。显然,这个数目越少,解码图像质量越高。为此,对因子 s 的表达式进行一些分析。根据 s 的表达式(5)和 Cauchy-Schwarz 不等式,有

$$|s| \leq \frac{\|R - \bar{R}I\| \|D - \bar{D}I\|}{\|D - \bar{D}I\|^2} = \sigma_R / \sigma_D \quad (7)$$

不难看出,对于给定的输入子块 R ,码块 D 的标准差 σ_D 越大, s 的绝对值越小,从而 s 满足约束 $|s| < 1$ 的可能性越大,需要作截断处理的可能性越小,从而导致解码图像质量的提高。因此,从 Ω 中预先排除小方差码块有望提高重构图像质量。此外, Ω 中码块数目的减少也能够加快编码速度。

最后,因为对码块 D 的 8 种等距变换(4 个 90° 倍数旋转和水平方向、垂直方向、主次对角线方向对称反射)不改变其分数盒维数,所以在赋序码本中码块 D 与其等距变换版本是近邻。于是编码器在最后进行 8 个等距变换,以进一步改进图像质量。

基于上述分析,本文算法的具体步骤如下:

(1) 分割图像为不重叠 $n \times n$ 方块 (R 块),记为 R 。以纵横方向步长均为 Δ 生成尺寸为 $2n \times 2n$ 的 D 块池,即相邻 D 块间在水平或垂直方向有 Δ 个像素的重叠。对于每个 D 块,采用 4-邻域像素值平均得到 $n \times n$ 块,这样的子块集合构成码本 Ω 。

(2) 设定 R 块的标准差阈值 τ 以及 D 块的标准差阈值 η ,定义容许码本 $\Omega_\eta = \{D \in \Omega; \sigma_D \geq \eta\}$ 。

(3) 计算每个子块 R 和码块 $D \in \Omega_\eta$ 的分数盒维数,按分数盒维数大小赋予码本 Ω_η 升序结构。

(4) 对于每个输入子块 R :

若 $\sigma_R < \tau$,则直接用 $\bar{R}I$ 近似之;

若 $\sigma_R \geq \tau$,则按二分法在赋序容许码本 Ω_η 中搜索与子块 R 分数盒维数相差最小的码块 D_c 。

(5) 在赋序容许码本 Ω_η 中定义以 D_c 为中心的 k -邻域,从中选择与子块 R 有最小 $E(R, D)$ 的码块 D_m 。然后考虑 D_m 的 8 个等距变换,并选出有最小

$E(R, D)$ 的等距变换序号 l 。序号 m 和 l 、参数 s 和 o 就组成输入子块 R 的分形码 (m, l, s, o) 。其中, s, o 是参数 s 和 o 的量化值。

(6) 对于其余输入子块 R ,重复步骤 4 ~ 步骤 5。

4 实验研究

本文算法中,编码时间和重构图像质量依赖于 3 个控制参数:搜索邻域大小 k 、输入子块 R 的标准差阈值 τ 和容许码本阈值 η 。

因为 s 必须满足约束 $|s| < 1$ 以保证解码迭代序列收敛,但研究表明^[1],按式(5)得到的 s 有一些不满足约束,因而需要进行截断处理。综合考虑压缩比与图像质量,参数 s, o 分别按 5b 和 7b 量化可以得到最佳效果^[1]。鉴于此,实验中采用如下截断方案:若 $s > 1$,取 $s = 31/32$,若 $s < 0$,取 $s = 0$ 。此外,对 o 的截断方案如下:若 $o > 127$,取 $o = 127$,若 $o < 0$,取 $o = 0$ 。当然这并不是最优截断方案。基本算法和本文算法都采用这个方案,因此,这并不影响对比的公平性。

实验对象为 4 幅不同类型的 512×512 (8b 量化)标准测试图像,它们是 Lena、Goldhill、Boat 和 Peppers。实验平台为运行 Windows XP 的 PC (1700MHz CPU/256M 内存),程序用 C++ Builder 编写。测试性能参数是峰值信噪比 (PSNR) 和编码时间 (s)。在实验中,选取 R 块大小为 4×4 , D 块大小为 8×8 ,生成 D 块池的滑窗步长为 8 个像素。

参数 τ 的实验研究。不难想见,参数 τ 越大,被当作常值块的输入 R 块越多,需要匹配搜索的 R 块越少,自然加快了编码速度,但可能伴随图像质量的降低。实验结果表明(图 2):当 $\tau > 4$ 时,解码图像中某些平滑区域(如脸部)出现轻微的“块效应”; τ 值越大,这种块效应越明显,但 $\tau = 4$ 时,块效应基本消失。因此,综合编码速度和图像质量,宜选取 $\tau = 4$ 作为缺省值。

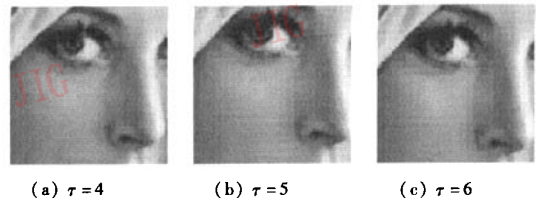


图 2 本文算法 ($\eta = 30, k = 100$) 的块效应

Fig. 2 Tiling effect of the proposed algorithm ($\eta = 30, k = 100$)

参数 η 的实验研究。显然,对于给定输入 R 块,参数 η 越大,容许码本越小。因此,取太大的参数 η ,完全有可能排除掉其最佳匹配块,从而引起解码图像质量的降低。关于参数 η 的不同取值对算法性能的影响,表 1 给出了实验结果:

(1) 从码本 Ω 中预先排除小方差子块,确实能提高重构图像质量,也能大大加快编码速度,这证实了 3.2 节的理论分析;

(2) 阈值 η 也不能太大,否则引起图像质量的降低;

(3) 综合质量与编码时间, $\eta = 30$ 是最佳值。

参数 k 的实验研究。根据上面的实验结果,兼顾质量与时间,宜取 $\tau = 4$ 和 $\eta = 30$ 作为缺省值。因此,下面设定 $\tau = 4, \eta = 30$ 来研究邻域大小 k 对算法性能的影响,表 2 给出了对所选 4 幅图像的实验结果:当 k 分别取 300、180、390 和 380 时, Lena、Goldhill、Boat 和 Peppers 的 PSNR 都不再变化(达到最大值),且都高于基本算法的 PSNR。当 $k = 5$ 时,

表 1 本文算法($\tau = 4, k = 100$)的编码结果

Tab. 1 Coding results of the proposed algorithm ($\tau = 4, k = 100$)

η	Lena		Goldhill		Boat		Peppers	
	PSNR	时间	PSNR	时间	PSNR	时间	PSNR	时间
0	31.2	7.64	29.85	12.0	27.58	8.14	30.44	8.61
10	32.29	7.67	30.47	12.0	27.75	8.01	31.86	8.58
20	32.66	7.76	31.54	11.94	30.80	8.05	32.35	8.55
30	33.28	7.78	31.61	11.9	31.46	8.09	32.91	8.59
40	33.41	7.95	31.19	5.61	31.38	8.06	32.98	8.56
50	32.86	4.47	30.35	1.42	30.98	3.84	32.87	7.94
60	32.18	1.67	29.1	0.50	27.92	0.25	32.03	2.61

4 幅图像的平均 PSNR 与全搜索基本算法相差大约 2.65dB,但主观质量都是不错的(图 3 仅仅给出 Lena 的解码图)。这证实了 3.2 节的理论分析,即输入 R 块的最佳匹配块一定是在与这个输入子块的分数盒维数最接近的码块附近。此外,当 $k = 180$ 时,4 幅图像的 PSNR 超过或接近其最大 PSNR。因此,一般可以设定 $k \leq 200$ 。

表 2 本文算法($\tau = 4, \eta = 30$)的编码结果

Tab. 2 Coding results of the proposed algorithm ($\tau = 4, \eta = 30$)

k	Lena		Goldhill		Boat		Peppers	
	PSNR	时间	PSNR	时间	PSNR	时间	PSNR	时间
5	30.83	0.62	30.08	0.94	28.95	0.64	30.11	0.79
100	33.28	7.78	31.61	11.98	31.46	8.09	32.91	8.59
170	33.58	13.47	31.82	19.72	31.78	13.58	33.15	14.45
180	33.59	13.70	31.82	19.50	31.81	14.34	33.16	15.39
300	33.87	22.8	—	—	32.12	23.76	33.43	25.48
380	—	—	—	—	32.32	29.89	33.52	32.57
390	—	—	—	—	32.35	30.72	33.52	33.64
基本算法	33.58	776.5	31.57	783.5	32.32	771.7	33.16	784.3



图 3 $k = 5$ 时 Lena 图像的解码图($\tau = 4, \eta = 30$)

Fig. 3 Decoded image of Lena at $k = 5$ ($\tau = 4, \eta = 30$)

本文算法的编码时间主要由参数 k 控制(见表 2),这个参数是可调的。因此,在实时性要求高、质量次之的应用场合,本文算法具有较好的应用前景,因为该算法可通过选取较小的 k 值来实现快速的编码。图 3 给出了 $k = 5$ 时 Lena 图像的解码图(此处 $\tau = 4, \eta = 30$),编码时间为 0.62s(基本算法编码时间为 776s),尽管 PSNR 下降 2.65dB,但主观质量还是不错的(肩膀边沿处有轻微“锯齿”效应)。

5 结 论

本文把分数盒维数应用于加快分形图像编码,得到了很好的实验效果。这说明,在分形图像编码中,分数盒维数可以作为子块匹配的一个重要的特征量。

参考文献 (References)

1 Wohlberg B, Jager G. A review of the fractal image coding literature

- [J]. IEEE Transactions on Image Processing, 1999, 8(12): 1716 ~ 1729.
- 2 He C, Yang S X, Huang X. Variance-based accelerating scheme for fractal image encoding[J]. IEE Electronics Letters, 2004, 40(2): 115 ~ 116.
- 3 He C, Yang S X, Xu X. Fast fractal image compression based on one-norm of normalised block[J]. IEE Electronics Letters, 2004, 40(17): 1052 ~ 1053.
- 4 He Chuan-jiang, Jiang Hai-jun, Huang Xi-yue. Fast fractal image encoding based on mean deviation-ordered[J]. Journal of Image and Graphics, 2004, 9(9): 1130 ~ 1134. [何传江, 蒋海军, 黄席樾. 基于平均偏差排序的快速分形图像编码[J]. 中国图象图形学报, 2004, 9(9): 1130 ~ 1134.]
- 5 He Chuan-jing, Yang Jing. Fast fractal image encoding based on shape feature[J]. Journal of Image and Graphics, 2005, 10(4): 410 ~ 414. [何传江, 杨静. 基于形态特征的快速分形图像编码[J]. 中国图象图形学报, 2005, 10(4): 410 ~ 414.]
- 6 Pentland A P. Fractal-based description of nature scenes[J]. IEEE Transactions on Pattern Analysis Machine Intelligence, 1984, 6(6): 661 ~ 674.
- 7 Feng J. Fractional fractal geometry for image processing [D]. Evanston, Illinois, USA: Northwestern University, 2000.
- 8 Bisoi A K, Mishra J. On calculation of fractal dimension of images [J]. Pattern Recognition letters, 2001, 22: 631 ~ 637.