

## 分形插值和分形图象压缩编码相结合进行图象数据压缩<sup>1</sup>

杨绍国 尹忠科 罗炳伟

(电子科技大学十四系 成都 610054)

**摘 要** 本文首先研究了分形图象压缩编码和分形插值图象压缩方法。这两种方法在压缩比很高时图象效果均不是很理想,为此我们提出了综合利用分形插值和分形图象编码方法进行图象数据高效压缩的思想。实验结果验证了这种思想的正确性和可行性。新的方法在恢复图象信噪比为 28.5dB 时,压缩倍数可高达 76 倍,速度也大大提高。

**关键词** 分形,插值,图象编码,图象压缩

**中图分类号** TN911.73

### 1 前 言

自然界具有标度不变的自相似结构,而图象是自然界的反映,因此图象也存在分形特征<sup>[1]</sup>。图象的分形特征表明图象数据存在着以前尚未认识和利用的信息冗余。为了利用这种信息冗余进行图象的数据压缩, M. Barnsley 于 1988 年提出了分形图象压缩编码方法<sup>[2]</sup>, 1990 年 A. Jacquin 提出了全自动的可行的分形图象压缩编码方法<sup>[3]</sup>。分形图象压缩编码的物理依据是自然界具有标度不变性的自相似结构,即分形特征。分形图象压缩编码的数学基础是迭代函数系统 (IFS), 分块迭代函数系统 (PIFS), 不动点原理 (The Fixed Point Theorem) 和拼贴原理 (The Collage Theorem)<sup>[2-4]</sup>。

应用分形图象压缩编码方法对图象进行压缩编码时,其算法复杂性和压缩比与图象的大小有关<sup>[4]</sup>。一方面算法复杂性随图象尺寸的增大而急骤增高。另一方面,一般分形图象编码方法是基于分块迭代函数系统的,当图象子块大小选定后,图象尺寸的大小决定图象含有图象子块的多少,因此在每一图象子块的代码变化不大的情况下,图象尺寸的大小决定了整幅图象压缩后的代码长度,即压缩比。从这两点出发,自然想到用插值方法<sup>[5]</sup>与分形编码方法结合起来对图象进行先抽样再编码,从而进行快速的高压缩比的图象压缩。恢复图象时先进行分形图象解码,得到小的恢复图象,然后利用插值得到原图象的恢复图象。进行通常的内插时,常会丢失纹理特征;利用分形插值方法插值可以产生高分辨率图象,并能保持原图象的纹理特征。对于方块效应等人工干扰,由于不具备分形特征,分形内插对其有良好的抑制作用。

本文把 DCT 域分形图象压缩编码方法与分形插值方法相结合,提出了一种压缩速度快、压缩比高的分形综合图象压缩方法。

### 2 快速 DCT 域分形图象压缩编码方法

**2.1 DCT 域分形图象压缩编码方法** 空间域分形图象压缩编码的过程为:对于原图象上任一子块  $f_{i,j}$  ( $i,j$  代表该子块的位置),在收缩图象上寻找某一同样大小的子块  $g_{k,l}$  ( $k,l$  代表子块  $g$  在收缩图象上的位置),使得  $g_{k,l}$  经过 (1) 式的变换后得到  $\hat{f}_{i,j}$  与  $f_{i,j}$  最相似。

$$\hat{f}_{i,j} = a \cdot g_{k,l} + b \cdot 1. \quad (1)$$

<sup>1</sup> 1997-04-17 收到, 1997-11-20 定稿

(1) 式中  $a$ 、 $b$  为待求实数，“1”代表象素值均为 1 的一个子块。对于原图象子块  $f_{i,j}$  来说，量化后的  $a$ 、 $b$  和  $g_{k,l}$  的位置即为该子块的压缩后的代码。

把 (1) 式变换到 DCT 域得 (子块大小取  $8 \times 8$ )：

$$\begin{bmatrix} \hat{F}_{i,j}^{0,0} & \hat{F}_{i,j}^{0,1} & \dots & \dots \\ \hat{F}_{i,j}^{1,0} & \hat{F}_{i,j}^{1,1} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}_{8 \times 8} = a \begin{bmatrix} \hat{G}_{k,l}^{0,0} & \hat{G}_{k,l}^{0,1} & \dots & \dots \\ \hat{G}_{k,l}^{1,0} & \hat{G}_{k,l}^{1,1} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}_{8 \times 8} + \begin{bmatrix} b_0 & 0 & \dots & \dots \\ 0 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}_{8 \times 8} \quad (2)$$

(2) 式中  $\hat{F}_{i,j}^{x,y}$  是  $\hat{f}_{i,j}$  子块经过 DCT 变换后的系数， $G_{k,l}^{x,y}$  是  $g_{k,l}$  子块经过 DCT 变换后的系数，子块  $b \cdot 1$  经过 DCT 变换后的系数，只有直流项不为 0，即  $b_0$ 。

DCT 域分形图象压缩编码的过程为：对于 DCT 域图象上的任一子块  $F_{i,j}$ ，在 DCT 域收缩图象上寻找一同样大小的子块  $G_{k,l}$ ，使得  $G_{k,l}$  经过 (2) 式的变换后得到的  $\hat{F}_{i,j}$  与  $F_{i,j}$  最相似。

事实上，无论怎样选取  $G_{k,l}$  和  $a$ 、 $b_0$ ，最相似的  $\hat{F}_{i,j}$  与  $F_{i,j}$  仍不够足够相似，这是造成分形图象压缩编码方法恢复图象质量不高的原因。为此进行如下改进，使  $\hat{F}_{i,j}$  与  $F_{i,j}$  足够相似，使相似子块的搜索更加容易，从而大大提高搜索速度。

$$\begin{bmatrix} \hat{F}_{i,j}^{0,0} & \hat{F}_{i,j}^{0,1} & \dots & \dots \\ \hat{F}_{i,j}^{1,0} & \hat{F}_{i,j}^{1,1} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}_{8 \times 8} = a \begin{bmatrix} 0 & Z_1 G_{k,l}^{0,1} & G_{k,l}^{0,2} & \dots \\ Z_2 G_{k,l}^{1,0} & Z_3 G_{k,l}^{1,1} & G_{k,l}^{1,2} & \dots \\ G_{k,l}^{2,0} & G_{k,l}^{2,1} & G_{k,l}^{2,2} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}_{8 \times 8} + \begin{bmatrix} F_{i,j}^{0,0} & T_1 F_{i,j}^{0,1} & 0 & \dots & \dots \\ T_2 F_{i,j}^{1,0} & T_3 F_{i,j}^{1,1} & 0 & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \quad (3)$$

(3) 式中  $T_1$ 、 $T_2$ 、 $T_3$ 、 $Z_1$ 、 $Z_2$ 、 $Z_3$  可取 0 或 1， $T_1 \neq Z_1$ ， $T_2 \neq Z_2$ ， $T_3 \neq Z_3$ ， $F_{i,j}^{x,y}$  为子块  $f_{i,j}$  经过 DCT 变换后的系数。对于任一 DCT 域子块，自适应地取以上八种形式中的一种，这样就可以利用 DCT 域能量集中的特点，从而利用子块的信息冗余，提高压缩倍数。同时，子块  $f_{i,j}$  的 DCT 变换系数  $F_{i,j}^{x,y}$  的应用，可使相似子块很容易找到，使编码的速度大大提高。

2.2 快速 DCT 域分形图象压缩编码方法 为方便起见，在这一部分中，我们所说的子块大小定为  $8 \times 8$ 。对于 DCT 域的任一子块  $F_{i,j}$  (值域块)，在 DCT 域收缩图象上寻找其相似子块  $G_{k,l}$  (定义域块) 的过程如图 1 所示。寻找相似子块时，先研究不搜寻时所对应的块  $G_{k,l}^{(0)}$  是否经 (3) 式运算后得到的  $\hat{F}_{i,j}^{(0)}$  满足要求。如果满足要求，则寻找这一值域块  $F_{i,j}$  的 PIFS 码的过程结束；如果不满足要求，再在  $G_{k,l}^{(0)}$  相邻小区域内寻找。如果存在某一个  $G_{k,l}^{(1)}$  在这个小区域内满足  $G_{k,l}^{(1)}$ ，经 (3) 式运算后得到的  $\hat{F}_{i,j}^{(0)}$  与  $F_{i,j}$  足够相似，则寻找

这一值域块  $F_{i,j}$  的 PIFS 码的过程结束; 否则, 在下一更大的区域寻找。如果总满足不了要求, 则直至寻找遍整幅收缩图象。

对于现实世界的图象和分块迭代函数系统而言, 一个值域块  $F_{i,j}$  的最佳相似子块往往存在于收缩图象 ( $\text{int}(i/16) \times 8, \text{int}(j/16) \times 8$ ) 的位置上或附近 ( $\text{int}$  代表取整运算)。所以我们认为在上述区域内寻找到的最佳相似子块, 可以近似地认为是整幅收缩图象上的最佳相似子块。这种近似所造成的恢复图象的质量下降是很小的。基于这种认识, 在整幅收缩图象上的搜寻不再是必要的。这将极大地提高编码的速度, 因为对于文献 [4] 提出的编码过程而言, 大部分计算时间是花费在整幅收缩图象的搜寻上。

### 3 综合利用分形技术进行图象数据压缩

单独用本文第 2 节提出的分形图象编码方法对 Lenna 图象进行图象压缩, 在压缩比为 73 时, 恢复图象的信噪比为 25.3。单独使用文献 [5] 所述的分形内插方法进行图象压缩, 在压缩比为 64 时, 恢复图象的信噪比为 25.7。由此可见, 这两种方法在压缩比很高时, 恢复图象的效果均不是很理想, 为此我们提出了综合利用分形插值和分形图象编码方法进行图象数据高效压缩的思想。

#### (1) 分形内插法图象压缩编码

对原图象, 本文首先采用固定式内插法编码, 即被传输的那部分像素是固定的, 未被传输的像素在解码时利用已传输的像素经过分形内插计算而恢复。

#### (2) DCT 域分形图象压缩编码

对原图象经抽样得到的图象, 采用本文第 2 节提出的分形图象压缩编码方法进行进一步的压缩, 以便进一步提高压缩比。

分形内插编码和分形图象压缩编码的结合, 不仅可以极大地提高计算速度和压缩比, 而且还能有效地抑制方块效应, 提高恢复图象的质量。

### 4 实验结果分析

图 2 为  $512 \times 512 \times 8$  的 Lenna 原图象。编码时首先对 Lenna 图象进行 4:1 抽样即每隔一行抽取一行、每隔一列抽取一列像素。对于抽样后的 Lenna 图象, 采用本文第 2 节提出的分形图象压缩编码方法进行进一步的压缩。解码时, 先用分形图象压缩编码方法进行解码, 得到抽样的恢复图象, 然后再用分形内插的方法, 得到原始图象的恢复图象。图 3 为压缩后恢复图象。从恢复图象看, 没有明显的失真, 虽然, 头发等细节部分变得模糊了, 但我们认为在 76 倍压缩比的情况下, 这样的图象质量是令人满意的。

表 1 给出了 PC486 微机对  $512 \times 512$  的 Lenna 图象进行压缩和恢复的实验结果, 在信噪比大致相同的条件下, 本文提出的方法和文献中原分形方法 (局部迭代函数系统 LIFS) 相比较, 压缩比提高到 76 倍, 压缩速度提高 636 倍。

本文提出的方法, 其突出优点是压缩比高, 压缩速度快, 这对图象通信的实现是非常有益的。

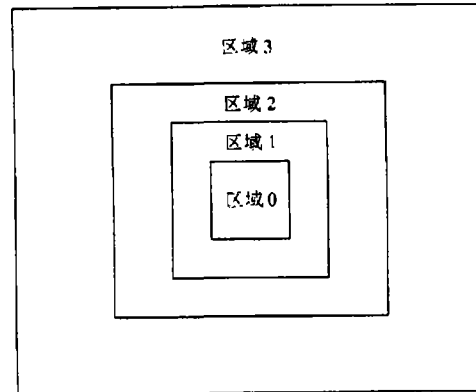


图 1 搜索示意图  
区域 0 代表无搜索时对应的子块, 区域 1 代表小的搜索区域, 区域 2 代表大的搜索区域, 区域 3 代表整个收缩图象。搜索次序为: 区域 0 → 区域 1 → 区域 2 → 区域 3。



图 2 512 × 512 × 8 的 Lena 原图象



图 3 恢复图象, 信噪比为 28.5dB, 压缩比为 76 倍

表 1 实验结果

	PSNR(dB)	压缩比 (倍)	压缩时间 (h)	备注
LIFS	30.9	24.8	25	引自文献 [4]
快速 LIFS	29.0	25.2	7	
新方法	28.5	76.4	0.011	

## 参 考 文 献

- [1] Mandelbrot B B. The Fractal Geometry of Nature. San Francisco: Freeman, 1982, 1-80.
- [2] Barnsley M F, Sloan A D. A better way to compress images. Byte, 1988, 13: 215-223.
- [3] Jacquin A E. Fractal image coding based on a theory of iterated contractive image transformations. SPIE, Vol.1360, Visual Communications and Image Processing, 1990, 227-239.
- [4] 房育栋, 余英林. 快速分形图象压缩编码. 电子学报, 1996, 24(1): 28-33.
- [5] 杨绍国, 尹忠科, 罗炳伟, 尧德中. 分形插值在图象处理中的应用. 电子科学学刊, 1997, 19(4): 562-565.

## FRACTAL IMAGE COMPRESSION WITH FRACTAL INTERPOLATION AND FRACTAL IMAGE CODING

Yang Shaoguo    Yin Zhongke    Luo Bingwei

(University of Electronic Science and Technology of China, Chengdu 610054)

**Abstract** In this paper, fractal image coding and fractal interpolation for image compression are studied at first. Because the reconstructed image quality using the two methods at high compression is not good, a new image compression idea combining these two methods is proposed. The coding results verified the new idea. The compression ratio is as high as 76 to 1 as PSNR being 28.5 dB and coding time is very short.

**Key words** Fractal, Interpolation, Compression coding, Image compression

杨绍国: 男, 1963年生, 副教授(博士后), 现从事图象处理与计算机视觉方面的科研和教学工作。

尹忠科: 男, 1969年生, 博士生, 现从事图象传输与处理方面的研究工作。

罗炳伟: 男, 1938年生, 教授, 现从事图象处理与计算机视觉方面的研究和教学工作。