图象块的分形近似和分块分形近似编码*

皮明红　彭嘉雄　刘华方

(华中理工大学图象识别与人工智能研究所,武汉　430074)

摘　要　提出了一种图象块的分形近似方法。该方法借助于 SAS 以达到对图象块的最小平方的分形近似,其运算量仅为 $2\times M\times N$ 次乘法和 $4\times M\times N$ 次加法,$M\times N$ 为图象块大小。由图象块分形近似量化而成的编码,称之为分块分形近似编码。其压缩比依赖于所选取图象块大小和迭代变换系数的量化。对 8×8 图象块,在不失真情况下,其压缩比达到 17.8 倍。通过对大量图象的实验结果表明,只需进行 8 次迭代,就能得到满意的重构图象。和其它分形编码相比,此方法简洁,编码速度快,对 220×220 的"Lena"图象进行编码,在 PC486/DX33 上,仅需时 30 秒。

关键字　IFS,SAS,不变函数,吸引子,分形近似,分形编码

1　引　言

自 A. Pentland[1]把分形引入到描述自然景象以来,分形被广泛地用于图象分析中。1988 年,M. F. Barnsley 在用仿射变换构造分形集的基础上[2],提出了分形图象压缩(Fractal Image Compression)的概念[3]。后来,A. E. Jacquin[4]和 J. M. Beaumont[5]系统地给出了图象分形编码,即分形块编码(Fractal Block Coding)。分形块编码的实质,在于利用了图象定义域块(局部)和值域块(局部)在不同尺度下的相似性。利用空间收缩算子(Spatial Contraction Operator)和作用在灰度上的变换(Massic Transformation)所合成的仿射变换把定义域块(局部)误差最小地贴到值域块(局部)上,从而建立了值域块与仿射变换之间的一一对应关系。进而用仿射变换和定义域块所在位置作为特征代表值域块图象,经量化后,构成了值域块的编码。然而在构造每个值域块对应的仿射变换时,在整幅图象的网格上全局搜索匹配,计算量极其庞大。对一幅 512×512 的图象进行编码,在 PC 486 计算机上要运行几小时,而且随着压缩倍数的提高,耗时更长。为了极大地减小运算量,必须改全局搜索匹配为类内搜索匹配。这就需要同时对定义域块和值域块进行分类。从而在构造值域块对应的仿射变换时,只需在同类定义域块中搜索匹配。A. E. Jacquin 把图象块分成 Shade Block、Midrange Block、Edge Block。虽然分类分形块编码不同程度地提高了分形编码的速度,但离实时性还相差甚远。

提高分形编码效率的另一条途径,是把图象块近似地看成一族仿射变换生成的分形吸引子。这些仿射变换不再是在整幅图象的网点上或类内搜索得到的,而是结合 SAS(Self Affine System)和某种误差最小准则向下求得的。此方法实质上是以图象块为处理单元,利用图象块的整体与局部存在尺度上的相似性,在误差最小的前提下,构造出把整体压缩到各个局部的仿射变换,从而把图象块与仿射变换族一一对应起来。以仿射变换族作为特征代表图象块,经量化后,构成了图象块的分块分形近似编码。

在文献[6]中,D. M. Monro 提出了对图象块分形近似的方法,此方法有二个缺点:(1)仿射变换中尺度因子{EMBED Equation |}的绝对值大于 0 占大多数,因此,此方法在迭代时不收敛。(2)随着图象块变大,解码图象的质量严重下降。为了克服以上二个缺点,本文改进简化了 D. M. Monro 的方法,得到了稳定的图象块分形近似方法。

* 图象信息处理与智能控制国家教委开放研究实验室资助课题。
收稿日期:1996-06-11;收到修改稿日期:1997-05-23

2 SAS和不变函数

IFS(Iterated Function System)是产生确定性分形图案的最重要的一种方法,其中最普遍的是SAS。SAS是一个R^2到R^2的仿射变换族$W\{w_k;k=1,2,\cdots\cdots,N\}$其中:

$$w_k(x,y)=\begin{bmatrix}a_{11}^{(k)} & a_{12}^{(k)}\\ a_{21}^{(k)} & a_{22}^{(k)}\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}+\begin{bmatrix}b_1^{(k)}\\ b_2^{(k)}\end{bmatrix}$$

如果每一个w_k是压缩映射或整体效果是压缩映射,那么一定存在唯一的分形吸引子A,且满足如下的不变性:

$$A=W(A)=\bigcup_{k=1}^{N}w_k(A)$$

SAS所产生的分形吸引子具有极好的保形性。什么样的仿射变换族,将产生什么样的分形图案,只是尺度大小不一样。而由分形图案反过来找相应的仿射变换族这一反问题,在文献[7]中作了详细的研究。本文意在把反问题与图象块的灰度函数结合起来,以达到对图象块的分形近似。

设$f(p)$是定义在分形吸引子A上的正实函数,图象块为$M\times N$,$[0,M-1]\times[0,N-1]$为矩形区域。吸引子$A\subset[0,M-1]\times[0,N-1]$,若$B=\{(i,j)|i=0,1,\cdots\cdots,M-1;\ j=0,1\cdots\cdots N-1\}$,那么把$f(p)$限制在$B$上且数字化,便是图象块的灰度矩阵。下面给出不变函数的定义,从而把函数f与SAS联系起来。

定义:函数$f:A\to R^+$是不变的,如果存在与$W=\{w_k:k=1,2,\cdots\cdots N\}$对应的压缩映射函数族$V=\{v_k:f(p)\to f(p);k=1,2\cdots\cdots,N\}$,使对任意$p\in A$,至少可以找到一组$(w_k,v_k)$,满足:

$$f(p)=Vf(p)=v_k(f(w_k^{-1}(p))) \tag{1}$$

若把f看成从B到A的实延拓,对于定义在A上满足不变性的灰度函数f,随着点p在w_k作用下随机的在分形吸引子A上变动,灰度值$f(p)$在v_k作用下同步地在灰度曲面上跳动。并且后一步的灰度值可以由当前灰度值和v_k给出。

3 图象块的不变分形近似

下面首先给出A上的函数f与g的误差准则,然后用最小二乘法,求出函数v_k

定义:函数f与g在A上的误差为

$$d(f,g)=\int_A(f-g)^2\mathrm{d}L \tag{2}$$

式中L为Lebesgue测度。

现在的问题是:给定了g,如何找出g的不变近似Vg,使式(2)定义的误差达到最小。为了导出与W对应的压缩映射函数族$V=\{v_k:g(p)\to g(p);k=1,2,\cdots,N\}$,使$g$在该压缩映射族下具有不变性,不妨让所有的压缩映射$v_k$为如下线性映射:

$$Vg(p)=v_k(g(w_k^{-1}(p)))=S^{(k)}\cdot g(w_k^{-1}(p))+b^{(k)} \tag{3}$$

那么

$$d(g,Vg)=\int_A(g-Vg)^2\mathrm{d}L\leqslant\sum_{k=1}^{N}\int_{w_k(A)}(g-v_k(g(w_k^{-1})))^2\mathrm{d}L \tag{4}$$

布满$[0,M-1]\times[0,N-1]$的分形星云(即分形吸引子A)产生的方法很多,最简单的方法。就是下面的4个仿射变换(见图1):

$$W_0(x,y)=\begin{bmatrix}0.5 & 0\\ 0 & 0.5\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}$$

$$W_1(x,y)=\begin{bmatrix}0.5 & 0\\ 0 & 0.5\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}+\begin{bmatrix}\frac{M}{2}\\ 0\end{bmatrix}$$

$$W_2(x,y)=\begin{bmatrix}0.5 & 0\\ 0 & 0.5\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}+\begin{bmatrix}0\\ \frac{N}{2}\end{bmatrix}$$

$$W_3(x,y)=\begin{bmatrix}0.5 & 0\\ 0 & 0.5\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}+\begin{bmatrix}\frac{M}{2}\\ \frac{N}{2}\end{bmatrix}$$

现在来求出w_k对应的v_k,使下式(5)最小

$$\int_{w_k(A)}(g-v_k(g(w_k^{-1})))^2\mathrm{d}L=\int_{(x,y)\in A}(g(w_k(x,y))-S^{(k)}\cdot g(x,y)-b^{(k)})\mathrm{d}L \tag{5}$$

对(5)关于$S^{(k)}$和$b^{(k)}$求导,得到

$$\frac{\partial}{\partial S^{(k)}}\int_A[\cdots]^2\mathrm{d}L=2\int_A[S^{(k)}\cdot(g(x,y))^2+b^{(k)}\cdot g(x,y)-g(w_k(x,y))\cdot g(x,y)]\mathrm{d}L \tag{6}$$

$$\frac{\partial}{\partial S^{(k)}}\int_A[\cdots]^2\mathrm{d}L=2\int_A[S^{(k)}\cdot g(x,y)^2+b^{(k)}-g(w_k(x,y))]\mathrm{d}L \tag{7}$$

由式(6)、(7)为0,得到方程组(8)。

$$\begin{bmatrix}\int_A(g(x,y))^2\mathrm{d}L & \int_A g(x,y)\mathrm{d}L\\ \int_A g(x,y)\mathrm{d}L & \int_A 1\mathrm{d}L\end{bmatrix}\begin{bmatrix}S^{(k)}\\ b^{(k)}\end{bmatrix}=$$

$$\begin{bmatrix} \int_A g(x,y)\cdot g(w_k(x,y))\mathrm{d}L \\ \int_A g(w_k(x,y))\mathrm{d}L \end{bmatrix} \tag{8}$$

把式(8)离散化,得到方程组(9)

$$\begin{bmatrix} \sum\limits_{(i,j)\in B} g(i,j)^2 & \sum\limits_{(i,j)\in B} g(i,j) \\ \sum\limits_{(i,j)\in B} g(x,y) & \sum\limits_{(i,j)\in B} 1 \end{bmatrix} \begin{bmatrix} S^{(k)} \\ b^{(k)} \end{bmatrix} = \begin{bmatrix} \sum\limits_{(i,j)\in B} g(x,y)\cdot g(w_k(x,y)) \\ \sum\limits_{(i,j)\in B} g(w_k(x,y)) \end{bmatrix} \tag{9}$$

这里,$B=\{(i,j)\mid i=0,1,\cdots,M-1;j=0,1\cdots N-1\}$。由(9),解出 $S^{(k)}$和 $b^{(k)}$。

$w_0(A)$	$w_1(A)$
$w_2(A)$	$w_3(A)$

图 1　$M\times N$ 的图象块和 4 个产生分形星云的仿射变换

Fig. 1　An M by N image block and four affine transformation to produce fractal nebulae

4　图象块的分形近似编码和重构

给出 8×8 的图象块和 4 个产生分形星云的仿射变换,*SAS* 把 8×8 正方形区域分别压缩到 4 个 4×4正方形区域(见图 1)。由于压缩映射的系数 S 和 b 是实数,在储存和传输前必须量化。其量化方法:首先计算 $f=S\times 64+b$,$g=S\times 128+b$,由于 f 和 g 在 0 到 255 之间,除以 4 后,把 f 和 g 量化为 6 bits,为了得到更高的压缩比,在每一块中,只对其中一个 f 和 g 按 6bits 进行量化,其余 f 和 g 以差值形式进行量化、存贮和传输。并利用 Huffman 编码对其差值进行编码,从而产生了图象块的分形近似编码。

在解码时,首先按 $f=S\times 64+b$,$g=S\times 128+b$ 解出 S 和 b。然后用如下两种方法之一进行图象重构,第一种方法,任给一初始图象块 V_0,经过迭代,得到图象块的重构系列$\{V_n=V^n g(W)\}$。第二种方法,任给一点 $P_0=(x,y)$,任赋该点一灰度值,经过空间上和灰度上的随机迭代:$\{p_n=W^n(p_0)\}$,$\{g_{n+1}=Vg(p_n)\}$。当 n 充分大时,就得到重构图象。

5　实验与结论

表 1 给出了本文方法和 Monro 方法对应的尺度因子统计表,从表 1 可以看出,Monro 方法的尺度因子,其绝对值大于 1 占绝大多数,从而导致了迭代不收敛。而本文方法的尺度因子,其绝对值小于 1 占 96%以上,并且每一图象块 4 个仿射变换从整体上看是收敛的。我们在很多不同的图象上进行了实验,都得到了类似的结果。

表 1　本文方法和 Monro 方法的尺度因子统计

Table 1　Scalar factor's statistics of our and Monro's method

实验图象	编码方法	尺度因子 $s<1$	尺度因子 $s\geqslant 1$	编码方法	尺度因子 $a_3<1$	尺度因子 $a_3\geqslant 1$
220×220 Lena	本文方法	2 904(96%)	12(4%)	Monro 方法	458(15.7%)	2 458(84.3%)
256×256 Plane	本文方法	4 075(99.5%)	21(0.5%)	Monro 方法	793(19.3%)	3 304(80.7%)

表 2 给出本文分形近似编码对 220×220 "Lena"图象和 256×256"Plane"图象的实验数据以及与 Jackquin 方法、Monro 方法的比较;图 3 给出本文方法、Jackquin 方法、Monro 方法的的重构图象;图 4 给出本文方法和 JPEG 的重构图象。从表 2 可以看出,尽管未对仿射变换系数作量化,Monro 方法重构图象的峰值信噪比还是不高。由于 Monro 方法的不收敛性,导致重构图象上存在着块状的斑点(图 3(e));并且随着图象块的变大,重构图象的质量严重地下降(图 3(f))。而本文方法的重构图象峰值信噪比较高,重构图象上无任何斑点,视觉效果好(图 3(b));但随着图象块变大,重构图象的质量略有下降(图 3(c))。另外我们将本文方法压缩 17.83倍、经 1 000 次随机迭代的重构图象(图 4(c))与 JEPG 压缩 16 倍的重构图象(图 4(d))进行了比较,本文方法的 PSNR 略高。重构效果相近。

总之,本文提出的分形近似编码克服了 Monro 方法的 2 个缺点,使压缩比从 8 倍提高到 17.8 倍。

表 2 本文方法的实验结果以及与 Monro 方法和 Jackquin 方法对照

Table 2 The result of our fractal coding and comparsion with Monro's and Jackquin's method

实验图象	编码方法	分形近似图象块的大小	原图象字节数(bytes)	压缩后字节数(bytes)	压缩比(倍)	编码时间(秒)	PSNR(dB)
Lena 220×220	本文方法	8×8	48 400	2 720	17.8	30	29.56
		16×8		1 458	33.2	20	26.92
Lena 220×220	Monro 方法	8×8	48 400			46	27.37
		16×8				32	22.43
Lena 220×220	Jackquin 方法(2 级)		48 400	3 636	13.3	1158	31.2
Plane 256×256	本文方法	8×8	65 536	3 675	17.83	29	33.52
	JPEG		65 536	3 096	16	13	33.14

a)原图象 b)本文方法(8×8) c)本文方法(16×8)

d)jackquin 方法(2 级) e)Monro 方法(,系数未量化) f)Monro 方法(,系数未量化)

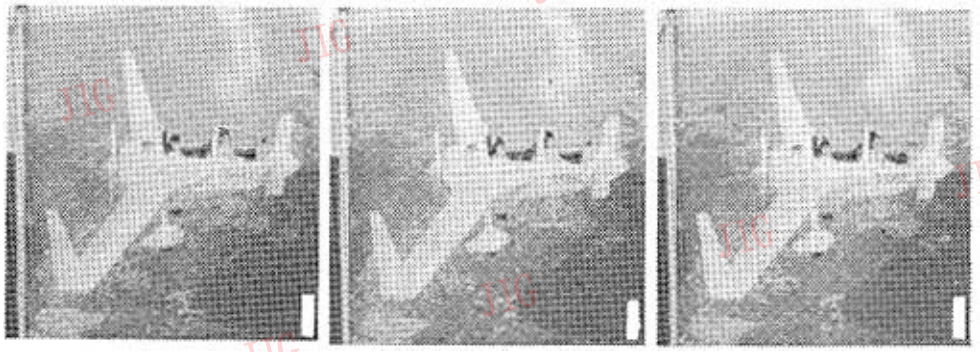
图 3 Lena 原图象与 8 次迭代后的解码图象

Fig. 3 original image and decoded image of Lena after 8 times iteration

a)original Lena image b)our decoded Lena by image block c)our decoded Lena by image block d)Jackquin's decoded Lena by two levels e)Monro's decoded Lena by image block and unquantized coefficients e)Monro's decoded Lena by image block and unquantized coefficients

分块分形近似编码以块为单位对图象进行分形近似后,对线性映射的系数进行量化、编码。它不象 Jackquin 算法,费时地在同类内搜索,从而找到与其匹配的定义域块。由于分块分形近似编码的运算量仅为 $2\times M\times N$ 次乘法和 $4\times M\times N$ 次加法,其计算量与图象块的大小成线性关系,因此,编码速度极快。完全可以对图象进行实时编码和传输。

分块分形近似编码和其它块编码一样,存在方块效应。其次分块分形近似编码对高频部分和 Edge Block 处理效果较差,而且随着图象块变大,重构效果下降。解决这一问题,是一项极有意义的工作。



(a)原图象　　(b)1000 次随机迭代后重构图象　　(c)用 JPEG 压缩 16 倍后重构图象

图 4　256×256 的飞机图象和解码图象

Fig. 4　Original "Plane" image and decoded image by image block

(a)original Plane image(b)our decoded Plane after 1000 random iteration,17.83 times (c)JPEG's decoded Plane,16 times.

参 考 文 献

1　Pentland A. Fractal Based Description of Natural Scenes. IEEE PAMI, 1984,6:661～674.

2　Barnsley M F, Demko S G. Iterated Function Schemes and the Global Construction of Fractals. Proc. R. Soc. A399,1985:243～275

3　Barnsley M F, Sloan A D. A Better Way to Compress Images, Byte,1988,13:215～233.

4　Jacquin A E. Fractal Image Coding:A Review. Proceedings of the IEEE,1993,81(10):1451～1465.

5　Beaumont J M. Image Data Compression Using Fractal Techniques. Proc. IEEE ICASSP,713～747.

6　Monro D M, Dudbridge F. Fractal Approximation of Image Block,Eleetron. Lett,1992,28:1053～1055.

7　Monro D M,Dudbridge F,wilson A. Deterministic Rendering of Self-affine fractals. IEEE Colloquium on the Application of Fractal Techniques in Image Processing,1990.

皮明红,博士,现在华中理工大学图象识别与人工智能研究所工作。从事分形与小波应用,图象处理,数据压缩等研究。

Fractal Approximation of Image Blocks and Block Fractal Approximation Code

Pi Minghong, Peng Jiaxiong, Liu huafang

(Institute of Pattern Recognition and Artificial Intelligence
Huazhong University of Science and Technology, Wuhan 430074)

Abstract　A method for approximation of image block is presented, based on a least squares fractal approximation by a Self Affine System. The computational cost is only equal to $2\times M\times N$ products and $4\times M\times N$ additions. The code of image block consists of the quantized fractal cofficients. This block coding of images is called as block fractal approximation code(BFAC). Its compression ratio depends on the size of image block and quantization of fractal cofficients. BFAC for "Lena" image by 8×8 blocks is quantized to six bits and encodes to a compression ratio of 17.8 : 1, its encoding and decoding time is about 30 seconds in PC486/DX33. After eight iterations of decoding, the reconstructive image look like the origin. Its greater speed and simplicity compared to other fractal transforms suggest its immediate applicability.

Keywords: IFS, SAS, Invariant function, Attractor, Fractal approximation, Fractal code