基于行列式的快速分形图像编码算法

何传江　刘维胜　申小娜
(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限失真压缩方法,然而,它存在计算量大的缺点,导致其编码时间过长。分形编码的时间主要花费在一个通常较大的码本中搜索每个输入子块的最佳匹配块。针对此问题,提出了一个加快编码的方案,它基于图像块的规范化行列式,能够在相对小的搜索邻域内找到输入子块的最佳匹配块。对 3 幅 512 × 512 测试图像的实验结果显示,与全搜索基本分形算法比较,依赖于搜索邻域的大小,该算法既能在峰值信噪比相同的情况下实现编码速度加快 30 倍左右,也能在主观质量略有下降的情况下实现编码速度加快1 000倍以上。

关键词　图像压缩　分形　分形图像编码　行列式
中图法分类号: TN919.81　　**文献标识码**: A　　**文章编号**: 1006-8961(2008)03-0435-05

Fast Fractal Image Encoding Algorithm Based on Local Determinants

HE Chuan-jiang, LIU Wei-sheng, SHEN Xiao-na
(*College of Mathematics and Physic*, *Chongqing University*, *Chongqing* 400030)

Abstract　Fractal image coding is a promising lossy compression technique in terms of achievable compression ratios and decoded image quality; However, it has the primary disadvantage of high computational demands resulting in unacceptably long encoding times. Most of the encoding times are spent on searching for the best-matched block to each of range blocks in a usually-large domain pool. This paper thus proposed an accelerating scheme by the determinants of normalized range and domain blocks, which can find out the best-matched block to an input range block in a relatively-small search neighborhood. Experimental results on three popular 512 × 512 test images showed that, depending on the search neighborhood size, the proposed algorithm not only can achieve the speed-up of about 30 times with the same PSNR(peak signal-to-noise ratio) as the baseline fractal algorithm with the full search, but also can obtain the speed-up of 1 000 times or more at the cost of tolerable degradation of the decoded image quality.

Keywords　image compression, fractal, fractal image coding, determinant

1　引　言

近十余年来,分形压缩编码技术受到了国内外众多专家和学者的关注,并广泛应用于图像特征提取、数字水印、图像签名、图像恢复和纹理分割等图像处理中。因为具有高压缩比、多分辨率和快速解码等特性,使得分形编码成为最有前途的压缩方法之一,也成为多媒体应用领域中很具吸引力的压缩方法(如微软公司的 CD-ROM 百科全书"Microsoft Encarta"就利用了这种压缩方法)。

分形图像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,用一个迭代函数系统(IFS)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。尽管高压缩比、分辨率无关性和解码快等优点使得它有很好的发展前景,但是也存在计算复杂性高、编码时间长的显著缺点,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

收稿日期:2006-04-21;**改回日期**:2006-10-17

第一作者简介:何传江(1964～　),男,博士,教授。1985 年、1988 年先后获得四川大学数学系理学学士、硕士学位,2004 年获得重庆大学自动化学院工学博士学位,现为重庆大学数理学院副院长。主要从事分形图像编码等研究。E-mail:chuanjianghe@ sina. com

得到最优结果,但这个过程的计算成本很高,从而限制了它的应用范围。目前,提出了许多加快分形编码的方法[2~8]。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也是必需的。因此,在保持图像质量不变或更好的前提下,实现更快的编码是分形图像编码的一个重要课题。

本文提出一个基于子块行列式的快速分形图像编码算法,并对 Lena 等 3 幅 512×512 测试图像,与全搜索基本分形算法比较,依赖于搜索邻域的大小,实验结果表明,该算法既能在峰值信噪比(PSNR)相同的情况下实现编码速度加快 30 倍左右,也能在主观质量略有下降的情况下实现编码速度加快 1 000倍以上。这说明与形态特征[5]、叉迹[6]一样,行列式也是子块匹配的一个较好的特征量。

2 基本分形算法

在分形图像编码中,一幅图像$\{\boldsymbol{\mu}_{\text{orig}}\}$被分割成互不重叠的子块(Range 块,简称 R 块)的集合$\{\boldsymbol{R}_i\}$,它们合起来覆盖整幅图像,即$\boldsymbol{\mu}_{\text{orig}}=\cup_i \boldsymbol{R}_i$。同时,在同一图像中确定另一种子块(Domain 块,简称 D 块)的集合,D 块可以重叠且不必覆盖整幅图像,其大小通常为 R 块的 4 倍。通常,D 块通过 4-邻域像素值平均收缩为 R 块的大小,收缩子块的集合构成匹配搜索所需的码本(记为$\boldsymbol{\Omega}$)。

编码阶段,对于每个 R 块$\boldsymbol{R}_i$,在码本$\boldsymbol{\Omega}$中搜索其最佳匹配码块$\boldsymbol{D}_{m(i)}\in\boldsymbol{\Omega}$,使得误差$E(\boldsymbol{R}_i,\hat{\boldsymbol{R}}_i)$最小。其中,$\hat{\boldsymbol{R}}_i=st(\boldsymbol{D}_{m(i)})+o\mathbf{I}$,$\mathbf{I}$是亮度值均为 1 的常值块,$t$是等距变换(4 个旋转与 4 个反射),因子$s$、$o$分别调整$t(\boldsymbol{D}_{m(i)})$的对比度和亮度。

具体说,对于每个 R 块$\boldsymbol{R}_i$,为了寻求其最佳匹配块,需要求解极小化问题(未考虑等距变换):

$$\|\boldsymbol{R}_i-(s_i\boldsymbol{D}_{m(i)}+o_i\mathbf{I})\|=\min_{D\in\boldsymbol{\Omega}}\{\min_{\substack{s,o\in\mathbf{R}\\|s|<1}}\|\boldsymbol{R}_i-(s\boldsymbol{D}+o\mathbf{I})\|\} \tag{1}$$

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

$$E(\boldsymbol{R}_i,\boldsymbol{D})=\min_{s,o\in\mathbf{R}}\|\boldsymbol{R}_i-(s\boldsymbol{D}+o\mathbf{I})\| \tag{2}$$

(记解为s_i,o_i)然后对不满足约束的s_i按照函数$s=\text{sign}(s)\min(|s|,1)$作截断处理。接着用全搜索方法求解问题(1)的外层极小化问题

$$E(\boldsymbol{R}_i,\boldsymbol{D}_{m(i)})=\min_{D\in\boldsymbol{\Omega}}E(\boldsymbol{R}_i,\boldsymbol{D}) \tag{3}$$

三元组$(m(i),\hat{s}_i,\hat{o}_i)$就构成$\boldsymbol{R}_i$的分形码,其中$\hat{s}_i,\hat{o}_i$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$\boldsymbol{R}_i$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(迭代次数预设)。具体地说,分形解码按下面的方式进行:

$$\boldsymbol{R}_i^{(k)}=s_i\tau_l(\boldsymbol{D}_{m(i)}^{(k-1)})+o_i\mathbf{I}\qquad \boldsymbol{\mu}^{(k)}=\cup_i\boldsymbol{R}_i^{(k)} \tag{4}$$

式中,$\boldsymbol{R}_i^{(k)}$是第k次迭代图像$\boldsymbol{\mu}^{(k)}$的 R 块,$\boldsymbol{D}_{m(i)}^{(k-1)}$是源于第$(k-1)$次迭代图像$\boldsymbol{\mu}^{(k-1)}$的最佳匹配码块,$\tau_l$是最佳的等距变换。不难看出,分形解码中每次迭代图像所需的码本都是由上一次迭代图像提供的,而第一次迭代图像所需的码本则由初始图像提供。由此可见,分形编码类似于但又不同于矢量量化(VQ)编码。

3 快速编码算法

在海量码本中搜索每个 R 块的最佳匹配码块占了分形编码的主要时间。如果能够按照某种方式尽可能排除不太可能匹配 R 块的码块,或者变全搜索为局部搜索,编码时间将会大大减少。事实证明,这种做法是很有效的[2~8]。本文算法也是基于这种思想。

3.1 算法分析

首先给出一个简单的结果:给定$\boldsymbol{R},\boldsymbol{D}\in\mathbf{R}^{n\times n}$,以及最小值问题

$$E(\boldsymbol{R},\boldsymbol{D})=\min_{s,o\in\mathbf{R}}\|\boldsymbol{R}-s\boldsymbol{D}-o\mathbf{I}\| \tag{5}$$

显然,$\|\boldsymbol{R}-s\boldsymbol{D}-o\mathbf{I}\|^2$是$s$、$o$的二次多项式,通过对$s$、$o$求偏导,得到式(5)的解为

$$s=\begin{cases}\dfrac{\langle\boldsymbol{R}-\bar{r}\mathbf{I},\boldsymbol{D}-\bar{d}\mathbf{I}\rangle}{\|\boldsymbol{D}-\bar{d}\mathbf{I}\|^2} & \boldsymbol{D}\neq\bar{d}\mathbf{I}\\ 0 & \boldsymbol{D}=\bar{d}\mathbf{I}\end{cases} \tag{6}$$

$$o=\bar{r}-s\bar{d} \tag{7}$$

$$E(\boldsymbol{R},\boldsymbol{D})^2=\|\boldsymbol{R}-\bar{r}\mathbf{I}\|^2-s^2\|\boldsymbol{D}-\bar{d}\mathbf{I}\|^2 \tag{8}$$

式中,$\langle,\rangle$表示欧氏内积,$\bar{r}$、$\bar{d}$分别表示$\boldsymbol{R}$、$\boldsymbol{D}$的亮度均值。

从第 2 节知道,在分形编码中,每个 R 块 $\boldsymbol{R}$ 由其最佳匹配块 $\boldsymbol{D}\in\boldsymbol{\Omega}$ 的亮度变换来近似,即

$$\boldsymbol{R}\approx s\boldsymbol{D}+o\mathbf{I}\tag{9}$$

根据式(7),上式可变为

$$\boldsymbol{R}\approx s\boldsymbol{D}+(\bar{r}-s\bar{d})\mathbf{I}\tag{10}$$

于是,得到

$$\boldsymbol{R}-\bar{r}\mathbf{I}\approx s(\boldsymbol{D}-\bar{d}\mathbf{I})\tag{11}$$

这说明,如果 $\boldsymbol{D}$ 是 $\boldsymbol{R}$ 的匹配块,那么它们去均值后仅仅相差一个常数倍数(即对比度因子 s)。

为了消除对比度因子的影响,单位化它们的去均值版本,得到

$$\frac{\boldsymbol{R}-\bar{r}\mathbf{I}}{\|\boldsymbol{R}-\bar{r}\mathbf{I}\|}\approx\frac{s(\boldsymbol{D}-\bar{d}\mathbf{I})}{\|s(\boldsymbol{D}-\bar{d}\mathbf{I})\|}=\operatorname{sgn}(s)\frac{\boldsymbol{D}-\bar{d}\mathbf{I}}{\|\boldsymbol{D}-\bar{d}\mathbf{I}\|}\tag{12}$$

上式说明,如果 $\boldsymbol{D}$ 是 $\boldsymbol{R}$ 的匹配块,那么它们的去均值、单位化版本(称为规范化)近似相等或相差一个正负符号。因为矩阵的行列式是其元素的连续函数,所以式(12)表明,$\boldsymbol{R}$ 和 $\boldsymbol{D}$ 的规范化版本的行列式的绝对值(以下简称规范化行列式)也应该近似相等。于是得到这样的命题:如果 $\boldsymbol{D}$ 匹配 $\boldsymbol{R}$,则 $\boldsymbol{D}$ 和 $\boldsymbol{R}$ 的规范化行列式近似相等。换言之,如果 $\boldsymbol{D}$ 和 $\boldsymbol{R}$ 的规范化行列式相差较大,那么 $\boldsymbol{D}$ 一定不能匹配 $\boldsymbol{R}$。

基于上述命题,把码本 $\boldsymbol{\Omega}$ 按其码块的规范化行列式的大小赋予其序结构(升序或降序),然后使用二分搜索法或其他搜索法,在赋序码本 $\boldsymbol{\Omega}$ 中找出在规范化行列式意义下与输入子块 $\boldsymbol{R}$ 最接近的码块 $\boldsymbol{D}$,最后在这个码块 $\boldsymbol{D}$ 的 k 邻域(k 是搜索邻域大小)内再进行匹配搜索,找出输入子块 $\boldsymbol{R}$ 的最佳匹配块。

其次,对于亮度变化很小的子块 $\boldsymbol{R}$(本文用标准差 $\sigma_{\mathrm{R}}=\|\boldsymbol{R}-\bar{r}\mathbf{I}\|/n$ 度量亮度变化的程度),即 $\sigma_{\mathrm{R}}\leqslant\tau$($\tau>0$ 是预设阈值,实验测定),这样的子块通常是图像的平滑区域,可以视为常值块,即 $\boldsymbol{R}\approx\bar{r}\mathbf{I}$。于是,对于标准差足够小的输入子块 $\boldsymbol{R}$,直接用其均值版本 $\bar{r}\mathbf{I}$ 代替(相当于 $s=0,o=\bar{r}$),无需再搜索其匹配块,从而可以进一步加快编码。此外,实验表明,当阈值 τ 取某些值时(一般 $2\leqslant\tau\leqslant5$),这种做法还可提高质量,如表 1 所示。

表 1　关于参数 τ 的实验数据

Tab. 1　Experimental data about the parameter τ

算法性能	阈值 τ								测试图像
	0	1	2	3	4	5	6	7	
PSNR(dB)	34.39	34.39	34.45	34.54	34.60	34.47	34.29	34.06	L
时间(s)	482.1	475.9	402.1	313.2	244.2	200.3	171.3	147.6	
PSNR(dB)	32.17	32.18	32.20	32.28	32.65	32.63	32.53	32.34	G
时间(s)	482.4	477.7	451.8	423.5	381.7	332.9	284.1	244.1	
PSNR(dB)	33.79	33.79	33.85	33.98	34.10	33.97	33.78	33.58	P
时间(s)	477.8	474.8	452.9	374.8	274.8	209.3	168.3	140.4	

注:L: Lena,G: Goldhill, P: Peppers

最后,解码图像质量也依赖于对不满足约束的对比度因子 s 作截断处理的数目。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(6)和 Cauchy-Schwarz 不等式,有

$$|s|\leqslant\frac{|\boldsymbol{R}-\bar{r}\mathbf{I}|\|\boldsymbol{D}-\bar{d}\mathbf{I}\|}{\|\boldsymbol{D}-\bar{d}\mathbf{I}\|^2}=\frac{\|\boldsymbol{R}-\bar{r}\mathbf{I}\|}{\|\boldsymbol{D}-\bar{d}\mathbf{I}\|}=\sigma_R/\sigma_D\tag{13}$$

不难看出,对于给定的输入子块 $\boldsymbol{R}$,码块 $\boldsymbol{D}$ 的标准差 σ_{D} 越大,s 的绝对值越小,s 满足约束 $|s|<1$ 的可能性越大,需要作截断处理的 s 的数目减少,自然有望提高解码图像质量。此外,码本 $\boldsymbol{\Omega}$ 中码块数目的减少也能够加快编码速度。

3.2　算法描述

基于上述理论分析,本文编码算法的具体步骤如下(以固定块分割为例):

(1) 设定平滑性阈值 τ 和容许码本阈值 η。

(2) 分割待编码图像为 $B\times B$ 固定块(如 4×4),记为 $\boldsymbol{R}$(R 块)。从待编码图像的左上角开始,从左往右,自上而下,以步长 L(通常大于一个像素)滑动一个尺寸为 $2B\times2B$ 的滑窗来构成 D 块池。对于每个 D 块,采用 4-邻域像素平均收缩为 $B\times B$ 大小,它们构成码本 $\boldsymbol{\Omega}$。

(3) 定义容许码本 $\boldsymbol{\Omega}_\eta=\{\hat{\boldsymbol{D}}\in\boldsymbol{\Omega}\mid\sigma_{\hat{D}}\geqslant\eta\}$,其中,$\hat{\boldsymbol{D}}$ 为 $\boldsymbol{D}$ 的规范化版本。

(4) 计算每个 R 块和 $\boldsymbol{D}\in\boldsymbol{\Omega}_\eta$ 的亮度均值和方差,以及规范化行列式。并且按照规范化行列式的大小对容许码本 $\boldsymbol{\Omega}_\eta$ 赋序(升序或降序)。

(5) 对于每个 R 块 $\boldsymbol{R}$,按下面的步骤搜索其最佳匹配块 $\boldsymbol{D} \in \boldsymbol{\Omega}_{\eta}$:

① 若 $\sigma_{\mathrm{R}} < \tau$,则直接用 $\bar{r}\mathbf{I}$ 近似之;

② 若 $\sigma_{\mathrm{R}} \geqslant \tau$,则按二分法在赋序容许码本 $\boldsymbol{\Omega}_{\eta}$ 中搜索与子块 $\boldsymbol{R}$ 规范化行列式相差最小的码块 $\boldsymbol{D}_{l}$。

③ 在 $\boldsymbol{\Omega}_{\eta}$ 中定义以 $\boldsymbol{D}_{l}$ 为中心的 k 邻域(k 是搜索邻域大小)。在这个搜索邻域中,每个 D 块都经过 8 种等距变换,并从中选择与 $\boldsymbol{R}$ 有最小误差的子块 $\boldsymbol{D}_{m}$ 和相应的等距变换序号。记录 $\boldsymbol{R}$ 的相应参数:$\hat{s}$,$\hat{o}$ 及 $\boldsymbol{D}_{m}$ 在图像中的位置,还有等距变换序号。其中 $\hat{s}$,$\hat{o}$ 是由式(6)、式(7)计算得到并经量化后的值。

(6) 对于其余 R 块,重复步骤(5),直至所有 R 块为止。

4 实验结果

实验对象为 3 幅不同类型的 512×512 标准测试图像(8bpp),它们是 Lena、Goldhill 和 Peppers。实验平台为运行 Windows XP 的 PC(1.5G CPU/ 256M 内存),算法采用固定块分割,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。

因为对比度因子 s 必须满足约束 $|s| < 1$,以保证解码迭代序列收敛,但研究结果表明,按式(6)得到的 s 有一些不满足约束,因而需要作截断处理。综合考虑压缩比与图像质量,参数 s,o 分别按 5 倍和 7 倍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 $o > 255$,取 $o = 255$,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 1　参数 τ 的研究

为了研究参数 τ 的最佳取值,本文实验的算法仅仅是基本算法的微小修改,即若 $\sigma_{\mathrm{R}} < \tau$,则直接用 $\bar{r}\mathbf{I}$ 近似之,其余部分未改变。实验结果见表 1。

从表 1 可以看出,对于合适的阈值 τ,在编码时间减少的同时,图像质量还有所提高,从 1 增加到 4 时,3 幅图像的 PSNR 都逐渐增加,当 $\tau = 4$ 时达到最大,超过 4 时 PSNR 开始减小。因此,综合编码速度和图像质量,宜选取 $\tau = 4$ 作为缺省值。

实验 2　参数 η 的研究

实验所用的算法也仅仅是基本算法的微小修改,即把基本算法的码本 $\boldsymbol{\Omega}$ 换成容许码本 $\boldsymbol{\Omega}_{\eta}$,其余部分未改变。实验结果见表 2。

表 2　关于参数 η 的实验数据

Tab. 2　Experimental data about the parameter η

算法性能	阈值 η								测试图像
	5	10	15	20	25	30	35	40	
PSNR(dB)	34.58	34.80	34.83	34.81	34.73	34.63	34.44	34.10	L
时间(s)	226.2	143.9	105.9	72.47	48.47	32.88	21.72	12.67	
PSNR(dB)	32.24	32.69	32.76	32.75	32.61	32.31	32.12	31.56	G
时间(s)	334.6	195.4	113.3	65.73	34.94	18.23	9.59	5.93	
PSNR(dB)	34.10	34.35	34.36	34.34	34.30	34.21	34.11	34.0	P
时间(s)	224.9	143.0	104.4	81.20	62.36	43.92	31.50	22.99	

注:L: Lena,G: Goldhill, P: Peppers

从表 2 看出,对于合适的阈值 η,在编码时间减少的同时,图像质量还有所提高,当 $\eta \leqslant 30$ 时,3 幅图像的 PSNR 都超过基本算法的 PSNR。尽管 $\eta = 30$ 并不是达到最大 PSNR 的取值,但是,综合编码速度和图像质量,宜选取 $\eta = 30$ 作为缺省值。此外,上述结果也表明,从码本 $\boldsymbol{\Omega}$ 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 3.1 节的理论分析。

实验 3　参数 k 的研究

本实验有两个目的,一是规范化行列式是子块匹配的一个好的特征量;另一个是研究参数 k 对算法性能的影响。

根据实验 1、实验 2 的结果,选取 $\tau = 4$ 和 $\eta = 30$ 作为算法的缺省值。表 3 给出了对所选 3 幅图像的实验数据。

从表 3 可以看出,(1)当 k 分别取 280、150 和 370 时,3 幅图像的 PSNR 都不再变化(达到最大值),且都略高于基本算法;(2)当 $k = 200$ 时,3 幅图

表 3　关于参数 k 的实验数据($\tau=4,\eta=30$)

Tab. 3　Experimental data about the parameter k ($\tau=4,\eta=30$)

图像	性能	邻域大小 k							基本算法
		1	100	150	200	280	370	390	
L	PSNR(dB)	30.28	34.10	34.28	34.38	34.46	34.46	–	34.39
	时间(s)	0.25	9.39	12.64	15.14	17.21	–	–	482.1
G	PSNR(dB)	29.54	32.19	32.30	32.30	–	–	–	32.17
	时间(s)	0.36	11.80	14.59	14.73	–	–	–	482.4
P	PSNR(dB)	29.37	33.52	33.70	33.84	33.98	34.03	34.03	33.79
	时间(s)	0.28	10.17	14.16	17.67	22.23	26.87	27.47	477.8

注:L: Lena,G: Goldhill,P: Peppers

像的 PSNR 超过或接近基本算法,同时编码时间减少 30 倍左右;(3)当 $k=1$ 时,尽管 3 幅图像的 PSNR 下降了 10% 左右,但是编码时间减少1 000倍以上。

图 1 给出了 $k=1$ 时 Lena 图像的解码图。此时,编码时间为 0.25s(基本算法为 482.1s,编码时间减少近2 000倍),PSNR 达到 30.28dB,且主观质量也是不错的(肩膀、帽子边沿处有一点点“毛刺”现象)。这说明,对于输入子块 $\boldsymbol{R}$,即使直接把规范化行列式与之接近的码块 $\boldsymbol{D}$ 作为其最佳匹配块,解码图像的质量也能够达到较高的水平。

图 1　$k=1$ 时 Lena 图像的解码图($\tau=4,\eta=30$)

Fig. 1　Decoded image of Lena at $k=1$ ($\tau=4,\eta=30$)

5　结　论

本文把行列式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,行列式可以作为子块匹配的一个重要的特征量。

参数 k 决定了本文算法的主要编码时间,它是可调的。因为可通过选取较小的 k 值来实现快速的编码,所以,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。此外,当 $k=1$ 时,解码图像的质量也达到较好的效果(PSNR 为 30dB 左右),因此,本文算法也可以作为无搜索分形编码算法的一个候选。

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　Lai C M, Lam K M, Siu W C A fast fractal image coding based on kick-out and zero contrast conditions[J]. IEEE Transactions on Image Processing, 2003, **12**(11): 1398 ~ 1403.

3　He C, Yang S X, Huang X. Variance-based accelerating scheme for fractal image encoding[J]. IEE Electronics Letters, 2004, **40**(2): 115 ~ 116.

4　He C, Yang S X, Xu X. Fast fractal image compression based on one-norm of normalized block[J]. IEE Electronics Letters, 2004, **40**(17): 1052 ~ 1053.

5　He Chuan-jiang, Yang Jing. Fast fractal image encoding based on shape feature[J]. Journal of Image and Graphics, 2005, **10**(4): 411 ~ 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): 411 ~ 414.]

6　He chuang-jiang, Huang xi-yue. Fast fractal image coding based on local cross trace[J]. Chinese Journal of Computers, 2005, **28**(10): 1753 ~ 1759. [何传江, 黄席樾. 基于图像块叉迹的快速分形图像编码算法[J]. 计算机学报, 2005, **28**(10): 1753 ~ 1759.]

7　Wu X, Jackson D, Chen H. A fast fractal image encoding method based on intelligent search of standard deviation[J]. Computers and Electrical Engineering, 2005, **31**: 402 ~ 421.

8　He C, Xu X, Yang J. Fast fractal image encoding using one-norm of normalized block[J]. Chaos, Solitons & Fractals, 2006, **27**(5): 1178 ~ 1186.