基于 IFS 系统的快速分形编码算法研究*

杨小远 赵 明

(北京航空航天大学理学院数学系 数学、信息与行为教育部重点实验室 北京 100083)

李 波

(北京航空航天大学 计算机学院 北京 100083)

摘 要

基于 Jacquin 提出的 IFS 系统, 本文提出了一种新的分形编码匹配策略. 这种新的匹配策略使编码时间大幅度缩短, 有效地克服了目前分形编码技术的缺点. 实验表明, 本文提出的这一新的策略可使编码时间缩短 80% 左右. 如果把本文提出的匹配算法与三行邻域搜索算法相结合, 则编码时间进一步缩短, 达 99.30% 左右, 而且峰值信噪比下降很少. 实验结果还表明: 该匹配策略在提高编码速度和重构图像质量方面同时具有优势, 有着广泛的应用前景.

关键词: 分形图像编码, 预搜索, 相关系数, 三行邻域

RESEARCH ON IFS-BASED FAST FRACTAL CODING ALGORITHM

Yang Xiaoyuan Zhao Ming

(*Department of Mathematics, School of Science, Beihang University, Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education, Beihang University and Peking University, Beijing* 100083)

Li Bo

(*School of Computer, Beihang University, Beijing,* 100083)

Abstract

Based on Jacquin's Iterated Function System, a new fractal coding matching strategy is proposed in this paper. The coding time can be reduced greatly by the new strategy. The disadvantage of present fractal coding technique can be overcome effectively. Experimental results show that the coding time can be reduced approximately by 80% by the new matching strategy. If combined with the 3-line neighborhood searching algorithm, the new matching strategy can further reduce the coding time approximately by 99.30%, with few dropping of PSNR. Experimental results also show that the new matching strategy has obvious advantage in increasing the coding speed and improving the quality of the reconstructed image, and it has wide application prospect.

Key words: fractal image coding, pre-searching, variance, correlation coefficient, 3-line neighborhood

* 2005 年 3 月 31 日收到.

§1. 引　　言

传统的分形图像编码是利用图像的自相似性对图像进行压缩的一种有效的算法. 它的理论基础是迭代函数系统 IFS (Iterated Function System) 理论和拼贴定理 (The Collage Theorem). 为了提高分形图像编码速度, Jacquin 将迭代函数系统的子映射定义域限制在图像的一个子块上 (称为局部迭代函数系统 LIFS), 提出了基于分块的自动分形图像编码系统 [1,2]. 自从 Jacquin 提出基于分块迭代变换理论及算法以来, 基于分形图像压缩算法得到广泛研究与应用. 主要的研究领域在三个方面: 值域块的划分方法, 映射方式的选择以及匹配策略的研究 [3,4].

传统的分形编码算法的主要问题是匹配计算复杂度大, 因而导致了编码时间过长. 为了解决这一问题, 本文提出一种新的分形编码匹配策略. 在进行值 - 域块匹配时, 利用图像块间的相似性, 对域块进行两次预搜索. 这种新的匹配策略避免了大量的匹配计算, 因而减少了编码时间. 这种快速算法可使编码时间普遍缩短 80% 左右, 而重构图像的 PSNR 下降不大, 有效地克服了分形编码的弊端.

以下两节我们将分别引进并讨论两次预搜索快速分形编码算法和三行邻域两次预搜索快速分形编码算法, 然后给出有关数值试验, 最后是有关结论.

§2. 快速分形编码算法

2.1. 几个定义

将原始图像 f 分割为互不重叠的 N_R 个图像值块 (Range)R_i, $i = 1, 2, \ldots, N_R$, 尺寸为 $k = n \times n$, 有

$$R_i \cap R_j = \phi, \quad i \neq j, \quad f = \bigcup_{i=1}^{N_R} R_i. \tag{1}$$

同时将原始图像 f 划分为可以相互重叠的 N_D 个图像域块 (Domain) D_j, 尺寸为 $l = 2n \times 2n$, $D_i \cap D_j \neq \phi$, $j = 1, 2, \ldots, N_D$, 所有域块构成域块池.

下面我们给出值 - 域快之间的匹配误差的定义.

定义 1. 对每一值块 R_i, 在域块池中搜索与其最优匹配的域块 D_j, 值 - 域块之间的匹配误差由式 (2) 给出, 并使式 (2) 最小而确定对比度比例因子 L_i, 对称旋转因子 T_i 以及灰度平移因子 O_i.

$$E(R_i, \widehat{D_j}) = \|R_i - (L_i \times T_i \times \widehat{D_j} + O_i I)\|^2, \tag{2}$$

其中 I 是与 R_i 同阶的单位矩阵, T_i 表示对图像块进行的八种变换中的一种, 即旋转 0°, 90°, 180°, 270° 以及垂直中线反射, 水平中线反射和对角线反射, $\widehat{\ }$ 表示 D 块的收缩变换 (通常采用四点平均法将 D 块收缩到和 R 块大小相同).

值块与域块方差的定义如下给出:

定义 2. 设值块 $R = \{a_{i,j}\}_{n\times n}$, 收缩变换后的域块 $\widehat{D} = \{b_{i,j}\}_{n\times n}$, 则值块, 域块的方

差分别定义为

$$var(R)=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(a_{i,j}-\overline{r})^2,\quad var(\widehat{D})=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(b_{i,j}-\overline{d})^2, \tag{3}$$

其中

$$\overline{r}=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}a_{i,j},\quad \overline{d}=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}b_{i,j}.$$

在我们的讨论中还需要引进值 - 域块之间的相关系数.

定义 3. 设值块 $R=\{a_{i,j}\}_{n\times n}$，收缩变换后的域块 $\widehat{D}=\{b_{i,j}\}_{n\times n}$，则值 - 域块之间的相关系数定义为

$$\rho(R,\widehat{D})=\frac{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}a_{i,j}b_{i,j}}{\sqrt{(\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}a_{i,j}^2)(\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}b_{i,j}^2)}}. \tag{4}$$

本文中, 我们采用下面的峰值信噪比作为评价重构图像质量的指标.

定义 4. 设原始的灰度图像为 $A=f(i,j)$, 重构图像为 $A'=f'(i,j)$, 其中 $i=1,2,\ldots,N$, $j=1,2,\ldots,M$, 则峰值信噪比为

$$PSNR=10\log\frac{255^2}{\frac{1}{MN}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{M}(f(i,j)-f'(i,j))^2}(dB). \tag{5}$$

2.2. 两次预搜索的快速分形编码算法

两次预搜索的快速分形编码算法是以 IFS 系统为基础. 在为值块寻找匹配域块时, 利用图像块的方差和图像块之间的相关系数在域块池中进行两次预搜索, 以排除大量的不相似域块, 减少大量的计算, 达到加速编码的目的.

在 Jacquin 的基本分形编码算法中, 值块 R 和域块 D 之间的匹配误差由式 (2) 所示. 设值块 $R=\{a_{i,j}\}_{n\times n}$, 收缩变换后的域块 $\widehat{D}=\{b_{i,j}\}_{n\times n}$, 记

$$\overline{r}=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}a_{i,j},\quad \overline{d}=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}b_{i,j}.$$

于是, 式 (2) 加入归一化因子就可以改写为 [5]

$$E_p(R,\widehat{D})=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(a_{i,j}-cb_{i,j}-h)^2. \tag{6}$$

我们称式 (6) 为归一化的匹配误差, 其中

$$c=\frac{n^2(\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}a_{i,j}b_{i,j})-(\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}a_{i,j})(\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}b_{i,j})}{n^2(\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}b_{i,j}^2)-(\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}b_{i,j})^2},\quad h=\overline{r}-c\overline{d}. \tag{7}$$

于是式 (6) 可进一步表示为

$$E_p(R,\widehat{D}) = var(R) - c^2 var(\widehat{D}). \tag{8}$$

当 $c^2 = 1$ 时, $E_p(R,\widehat{D})$ 取得最小值

$$E_{pmin}(R,\widehat{D}) = var(R) - var(\widehat{D}). \tag{9}$$

称式 (9) 为基于方差的匹配误差.

我们构造的两次预搜索快速分形编码算法的具体步骤如下:

1. 把原始图像 f 分割为互不重叠的 N_R 个图像值块 (Range)R_i, $i = 1,2,\ldots,N_R$, 尺寸为 $k = n \times n$, 有

$$R_i \cap R_j = \phi, \quad i \neq j, \quad f = \bigcup_{i=1}^{N_R} R_i.$$

2. 把原始图像 f 划分为可以相互重叠的 N_D 个图像域块 (Domain)D_j, 尺寸为 $l = 2n \times 2n$, $D_i \cap D_j \neq \phi$, $j = 1,2,\ldots,N_D$, 所有域块构成域块池.

3. 对于每一值块 R_i, 在域块池中从左到右、从上到下进行扫描, 搜索与之最佳匹配的域块 D_j. 在匹配过程中, 首先对每一域块 D_j 都用四点平均法进行收缩变换得收缩域块 $\widehat{D_j}$. 两次预搜索的快速算法匹配过程如下:

   (a) 将域块 $\widehat{D_1}$ 设置为初始最佳匹配域块, 对 $\widehat{D_1}$ 进行八种对称旋转变换, 分别通过式 (6) 计算 R_i 与八种变换后的 $\widehat{D_1}$ 的归一化匹配误差, 取其中最小误差作为初始匹配误差, 记为 E_p^0.

   (b) 对域块 $\widehat{D_2}$, 用式 (9) 计算 R_i 与 $\widehat{D_2}$ 之间的基于方差的匹配误差 $E_{pmin}(R_i,\widehat{D_2})$, 进行首次预搜索, 即将它和 E_p^0 进行比较.

   - 如果 $E_{pmin}(R_i,\widehat{D_2}) \geq E_p^0$, 拒绝域块 $\widehat{D_2}$, 不再进行匹配计算.
   - 否则, 用式 (4) 计算 R_i 与 $\widehat{D_2}$ 之间相关系数 $\rho(R_i,\widehat{D_2})$, 进行第二次预搜索.
   - 如果 $\rho(R_i,\widehat{D_2}) < \rho_0$($\rho_0$ 是根据实际需要而事先设定的相关系数门限), 拒绝域块 $\widehat{D_2}$, 不再进行匹配计算.
   - 否则, 对 $\widehat{D_2}$ 进行八种对称旋转变换, 分别通过式 (6) 计算 R_i 与八种变换后 $\widehat{D_2}$ 的归一化匹配误差, 取其中最小误差记为 $E_p(R_i,\widehat{D_2})$.
   - 如果 $E_p(R_i,\widehat{D_2}) \geq E_p^0$, 拒绝域块 $\widehat{D_2}$.
   - 否则 E_p^0 被 $E_p(R_i,\widehat{D_2})$ 替换, 并将 $\widehat{D_2}$ 设置为当前的最佳匹配域块.

   (c) 按照上一步扫遍域块池中的所有域块, 寻找与值块 R_i 的最佳匹配域块.

4. 对每一个值块 R_i, 通过第三步搜索与之匹配的域块 D_j, 同时得到变换 $\omega(X_i,Y_i,L_i,T_i,O_i)$, 并记录为编码, 其中 X_i 和 Y_i 表示 D_i 块的位置.

5. 每一个值块 R_i 都对应着一个变换 ω_i, 由这些搜索结果构成的 PIFS(Partitioned Iteration Function System), 即为图像的编码结果 $\omega = (\omega_1, \omega_2, \ldots, \omega_{N_R})$.

6. 如果所有的值 – 域块都匹配结束, 记录所有的参数, 编码结束.

根据不动点定理和拼贴定理, 解码时, 从任意起始图像开始, 经过多次迭代, 即可重构图像.

§3. 三行邻域两次预搜索快速分形编码算法

将本文两次预搜索快速分形算法与三行邻域搜索法 [8] 相结合, 我们可构造一种三行邻域两次预搜索快速分形编码算法, 为此我们首先介绍值块的三行邻域定义.

定义 5. 以值块为邻域中心, 由值块所在行及其邻近两行所确定的邻域称为值块的三行邻域. 各种情况由图 1 所示.

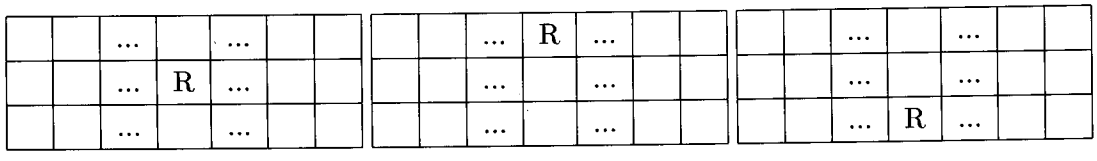

图 1 值块与之三行邻域

三行邻域搜索法是以 Jacquin 全搜索法为基础, 算法步骤与全搜索算法基本相同, 只是在为值块搜索匹配域块时, 搜索范围不是全部域块池, 而是限制在值块的三行邻域内.

三行邻域两次预搜索分形编码算法具体步骤如下:

1. 把原始图像 f 分割为互不重叠的 N_R 个图像值块 (Range)R_i, $i = 1, 2, \ldots, N_R$, 尺寸为 $k = n \times n$, 满足
$$R_i \cap R_j = \phi, \quad i \neq j, \quad f = \bigcup_{i=1}^{N_R} \overline{R}_i.$$

2. 把原始图像 f 划分为可以相互重叠的 N_D 个图像域块 (Domain)D_j, 尺寸为 $l = 2n \times 2n$, $D_i \cap D_j \neq \phi$, $j = 1, 2, \ldots, N_D$, 所有域块构成域块池.

3. 对于每一值块 R_i, 在值块 R_i 的三行邻域内, 利用本文快速算法的匹配过程, 即两次预搜索, 搜索 R_i 的最佳匹配域块 D_j.

4. 对每一个值块 R_i, 通过第三步搜索与之匹配的域块 D_j, 同时得到变换 $\omega(X_i, Y_i, L_i, T_i, O_i)$, 并记录为编码, 其中 X_i 和 Y_i 表示 D_i 块的位置.

5. 每一个值块 R_i 都对应着一个变换 ω_i, 由这些搜索结果构成的 PIFS, 即为图像的编码结果 $\omega = (\omega_1, \omega_2, \ldots, \omega_{N_R})$.

6. 如果所有的值 – 域块都匹配结束, 记录所有的参数, 编码结束.

§4. 实验结果与分析

为了检测本文提出编码方法的效果, 我们针对原始图像为 256×256(8bitpixel) 的 Lena, Oldhouse 和 Camera 三幅标准图像进行了计算机模拟. 在模拟中, 采取值块尺寸为 4 × 4, 域块尺寸为 8 × 8, 搜索步长为 1, 解码迭代 10 次. 压缩算法的编程语言是 C 语言, 运行环境. 中文 WindowsXP, Pentium4　2.8GHz, RAM 是 256M.

4.1. 两次预搜索快速分形编码算法

实验结果由表 1 给出, 在压缩比保持不变的情况下, 表 1 给出了 Jacquin 全搜索算法与本文提出的两次预搜索快速分形编码算法在相关系数门限 ρ_0 分别取 0.7, 0.75, 0.8, 0.85 时的 PSNR 和编码时间, 单位分别是 dB 和 s(秒).

表 1　两次预搜索快速分形编码算法与 Jacquin 算法的实验结果

图像	Jacquin 算法		两次预搜索快速分形编码算法							
	PSNR	时间	$\rho_0 = 0.7$		$\rho_0 = 0.75$		$\rho_0 = 0.8$		$\rho_0 = 0.85$	
	(dB)	(s)	PSNR	时间	PSNR	时间	PSNR	时间	PSNR	时间
Lena	33.21	914.66	31.31	354.61	30.64	251.89	29.77	170.50	29.08	123.58
Olhouse	33.33	911.76	35.24	298.26	34.62	252.28	33.83	127.52	33.37	90.06
Camera	29.73	917.01	26.87	103.19	26.72	99.36	26.51	94.75	26.35	90.00

图 2, 图 3, 图 4 分别是对三幅标准图像进行实验的重构图像比较.

(a) 原始图像　(b)Jacquin 方法　(c)$\rho_0 = 0.7$　(d)$\rho_0 = 0.75$　(e)$\rho_0 = 0.8$　(f)$\rho_0 = 0.85$

图 2　对 Lena 图像进行的实验. 其中 (b) 是 Jacquin 方法重构图像,(c)(d)(e)(f) 是快速算法重构图像

(a) 原始图像　(b)Jacquin 方法　(c)$\rho_0 = 0.7$　(d)$\rho_0 = 0.75$　(e)$\rho_0 = 0.8$　(f)$\rho_0 = 0.85$

图 3　对 Oldhouse 图像进行的实验. 其中 (b) 是 Jacquin 方法重构图像,(c)(d)(e)(f) 是快速算法重构图像

(a) 原始图像 (b)Jacquin 方法 (c)$\rho_0 = 0.7$ (d)$\rho_0 = 0.75$ (e)$\rho_0 = 0.8$ (f)$\rho_0 = 0.85$

图 4 对 Camera 图像进行的实验. 其中 (b) 是 Jacquin 方法重构图像,(c)(d)(e)(f) 是快速算法重构图像

表 2 给出了两次预搜索快速分形编码算法与 Jacquin 全搜索算法的性能比较, 即给出了快速算法在相关系数门限 ρ_0 分别取 0.7, 0.75, 0.8, 0.85 时较 Jacquin 算法的重构图像的 PSNR 的升降情况及编码时间缩短的百分比.

表 2 两次预搜索快速分形编码算法与 Jacquin 算法的性能比较

图像	$\rho_0 = 0.7$		$\rho_0 = 0.75$		$\rho_0 = 0.8$		$\rho_0 = 0.85$	
	PSNR 下降	时间缩短	PSNR 下降	时间缩短	PSNR 下降	时间缩短	PSNR 下降	时间缩短
Lena	1.90	61.23%	2.57	72.46%	3.44	81.35%	4.13	86.49%
Olhouse	-1.91	67.29%	-1.29	72.33%	-0.50	86.01%	-0.04	90.12%
Camera	2.86	88.75%	3.01	89.17%	3.22	89.67%	3.38	90.19%

从表 2 可以看到, 本文提出的快速分形编码算法在相关系数门限取不同值时, 编码时间缩短程度不同, 但速度普遍提高很大, 在重构图像的质量可以接受的情况下, 编码时间大约可缩短 80%~90% 左右.Lena 和 Camera 的重构图像的 PSNR 略微下降, 而 Oldhouse 的重构图像的 PSNR 普遍有所提高.

为了验证本文算法的性能, 我们做了大量的模拟实验以进行统计分析, 表 3 给出了本文快速算法与 Jacquin 全搜索算法在重构图像的 PSNR 保持相同情况下的实验结果.

表 3 在 PSNR 保持不变的情况下两次预搜索快速分形编码算法与 Jacquin 算法的实验结果

图像	Jacquin 算法		两次预搜索快速分形编码算法			
	PSNR(dB)	编码时间 (s)	ρ_0	PSNR(dB)	编码时间 (s)	时间缩短
Lena	33.21	914.66	0.47	33.21	493.80	45.94%
Olhouse	33.33	911.76	0.87	33.33	80.06	91.22%
Camera	29.73	917.01	0.41	29.73	216.78	76.36%

从表 3 可以看出, 在重构图像的 PSNR 保持不变的情况下, 本文快速算法可使编码时间缩短 46%~91% 左右. 如果继续放宽相关系数门限, 重构图像的 PSNR 会不断提高, 直到收敛.

4.2. 三行邻域两次预搜索快速分形编码算法

实验结果如表 4 所示. 在压缩比保持不变的情况下, 表 4 给出了 Jacquin 全搜索算法与三行邻域两次预搜索快速分形编码算法在相关系数门限 ρ_0 分别取 0.7, 0.75, 0.8 时的 PSNR 和编码时间, 单位分别是 dB 和 s(秒).

表 4　三行邻域两次预搜索快速分形编码算法的实验结果

图像	Jacquin 算法		三行邻域两次预搜索快速分形编码算法					
	PSNR (dB)	时间 (s)	$\rho_0 = 0.7$		$\rho_0 = 0.75$		$\rho_0 = 0.8$	
			PSNR	时间	PSNR	时间	PSNR	时间
Lena	33.21	914.66	29.86	8.50	29.38	6.41	28.70	4.64
Olhouse	33.33	911.76	32.16	9.09	31.61	8.41	31.31	7.76
Camera	29.73	917.01	25.90	4.01	25.79	3.94	25.73	3.83

图 5, 图 6, 图 7 分别是对三幅标准图像进行实验的重构图像比较.

(a) 原始图像　(b)Jacquin 方法　(c)$\rho_0 = 0.7$　(d)$\rho_0 = 0.75$　(e)$\rho_0 = 0.8$

图 5　对 Lena 图像进行的实验. 其中 (b) 是 Jacquin 方法重构图像,(c)(d)(e) 是结合算法重构图像

(a) 原始图像　(b)Jacquin 方法　(c)$\rho_0 = 0.7$　(d)$\rho_0 = 0.75$　(e)$\rho_0 = 0.8$

图 6　对 Oldhouse 图像进行的实验. 其中 (b) 是 Jacquin 方法重构图像,(c)(d)(e) 是结合算法重构图像

(a) 原始图像　(b)Jacquin 方法　(c)$\rho_0 = 0.7$　(d)$\rho_0 = 0.75$　(e)$\rho_0 = 0.8$

图 7　对 Camera 图像进行的实验. 其中 (b) 是 Jacquin 方法重构图像,(c)(d)(e) 是结合算法重构图像

表 5 给出了三行邻域两次预搜索快速分形编码算法与 Jacquin 全搜索算法性能的比较, 给出了三行领域两次预搜索算法在相关系数门限 ρ_0 分别取 0.7, 0.75, 0.8 时较 Jacquin 算法重构图像的 PSNR 的升降情况及编码时间缩短的百分比.

从表 5 可以看到, 三行邻域两次预搜索快速分形编码算法的编码时间可缩短 99% 以上.

表 5 三行邻域两次预搜索算法与 Jacquin 算法的性能比较

图像	$\rho_0 = 0.7$		$\rho_0 = 0.75$		$\rho_0 = 0.8$	
	PSNR 下降	时间缩短	PSNR 下降	时间缩短	PSNR 下降	时间缩短
Lena	3.35	99.07%	3.83	99.30%	4.51	99.49%
Olhouse	1.17	99.00%	1.72	99.08%	2.02	99.15%
Camera	3.83	99.56%	3.94	99.57%	4.00	99.58%

§5. 结 论

利用本文提出的两次预搜索快速分形编码算法进行编码时, 可根据重构图像的不同要求, 取不同的相关系数门限. 如果要求图像质量高, 可将相关系数门限确定为稍小些. 如果要求图像质量不高, 可将相关系数门限确定为稍大些. 这样在编码时间和重构图像质量两者的均衡方面上会取得更好的效果. 总体来讲, 本文提出的快速算法在缩短编码时间方面是非常有效的. 另外, 该算法在提高重构图像质量方面也具有优势, 如图像 Oldhouse, 在相关系数门限 $\rho_0 = 0.85$ 时, 编码时间缩短 90.12%, 重构图像的 PSNR 仍然有所提高.

该算法与三行邻域搜索算法相结合时, 编码速度能进一步得到提高, 编码时间可缩短 99% 以上. 可见, 本文提出的算法有效地克服了分形编码的弊端. 我们相信本文提出的算法有着广泛的应用前景.

参 考 文 献

[1] Arnaud. E. Jacquin. A novel fractal block-coding technique for digital images.Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, 4:4(1990) 2225–2228.

[2] Arnaud. E. Jacquin. Fractal image coding: A review. Proc. IEEE, 81:10(1993) 1451–1465.

[3] Brendt Wohlberg and Gerhard de Jager. A review of the fractal image coding literature. IEEE Trans. Image Processing, 8:12(1999) 1716–1729.

[4] Tong Chong-Sze and Pi Minghong. Fast Fractal Image Encoding Based on Adaptive Search. IEEE. Trans. Image Processing, 10:9(2001) 1269–1277.

[5] 陈守吉, 张立明, 分形与图像压缩, 上海科技教育出版社, 上海, 1998.

[6] Rafael C. Gonzalez and Richard E. Woods 著, 阮秋琦, 阮宇智等译, 数字图像处理 (第二版), 电子工业出版社, 北京, 2003.

[7] 王耀明, 图像的矩函数 – 原理、算法及应用, 华东理工大学出版社, 上海, 2002.