



# 一种改进的多分辨率分形解码方法

孙怀江 杨静宇

(南京理工大学计算机系 南京 210094)

(E-mail: sunhuaij@mail.njust.edu.cn)

**摘要** 为了减小分形解码方法的计算复杂性,基于对一种分形编码方法所使用 PIFS (Partitioned Iterated Function System) 不动点多分辨率特性的研究,提出一种改进的多分辨率分形解码方法,与 Oien 和 Lepsoy 的塔式方法相比,需要更少的计算量. 实验结果验证了有关结论的正确性.

**关键词** 分形, 图象压缩, 不动点, 多分辨率.

## AN IMPROVED MULTIREOLUTION FRACTAL DECODER

SUN Huaijiang YANG Jingyu

(Computer Department, Nanjing University of Science & Technology, Nanjing 210094)

**Abstract** Based on the multiresolution properties of PIFS (Partitioned Iterated Function System)'s fixed point used in the Oien and Lepsoy's fractal encoder, an improved multiresolution fractal decoder is proposed in this paper. Compared with the Oien and Lepsoy's pyramidal fractal decoder, the proposed decoder can further reduce computation complexity. Related conclusions are verified with experimental results.

**Key words** Fractal, image compression, fixed point, multiresolution.

## 1 引言

尽管分形图像解码过程只是对任意初始图像重复应用由 PIFS 所确定的映射<sup>[1,2]</sup>,但 PIFS 的有关参数在很大程度上影响着所需要的迭代次数,有时收敛速度较慢难以满足实时要求. Oien 和 Lepsoy 提出一种约束的编码方法<sup>[3]</sup>,引入了正交化运算,使得解码过程只需要较少的确定的迭代次数,克服了映射参数的影响,塔式结构进一步减少了计算量,但其减少计算量的方法仅仅是去掉了重复的计算. 本文在 Oien 和 Lepsoy 方法的基础上利用不动点的多分辨率特性,提出一种解码方法,需要更少的计算量.

## 2 一种约束的编码方法

本文将做如下假设：

1) 图像  $\mu$  为  $N \times N$  像素,  $\mu_{i,j}(i, j=0, \dots, N-1)$  代表其第  $(i, j)$  个像素值,  $N$  为 2 的整数次幂；

2) 令所有  $R_k = S_k(m_k, n_k, r_k)$  为方形子块, 其中  $(m_k, n_k)$  表示其左上顶点在图像  $\mu$  中的坐标,  $2^{r_k}$  为其水平或垂直方向像素数, 每个 range 块  $\mu|_{R_k}$  为  $2^{r_k} \times 2^{r_k}$  像素,

$$(\mu|_{R_k})_{i_k, j_k} = \mu_{m_k+i_k, n_k+j_k} \quad (i_k, j_k = 0, \dots, 2^{r_k} - 1) \quad (1)$$

代表 range 块  $\mu|_{R_k}$  的块内第  $(i_k, j_k)$  个像素值；

3) 令所有  $D_k = S_k(p_k, q_k, d_k)$  为方形子块且由整数个 range 块构成, 其中  $(p_k, q_k)$  表示其左上顶点在图像  $\mu$  中的坐标,  $2^{d_k}$  为其水平或垂直方向像素数, 每个 domain 块  $\mu|_{D_k}$  为  $2^{d_k} \times 2^{d_k}$  像素,

$$(\mu|_{D_k})_{i_k, j_k} = \mu_{p_k+i_k, q_k+j_k} \quad (i_k, j_k = 0, \dots, 2^{d_k} - 1) \quad (2)$$

代表 domain 块  $\mu|_{D_k}$  的块内第  $(i_k, j_k)$  个像素值,  $d_k = r_k + 1$ ；

4)  $w_k$  约束为如下形式

$$(w_k(\mu|_{D_k}))_{i_k, j_k} = a_k \cdot (s_k(\tau(\mu|_{D_k}) - \bar{\tau}(\mu|_{D_k})))_{i_k, j_k} + b_k \quad (i_k, j_k = 0, \dots, 2^{r_k} - 1), \quad (3)$$

其中  $\tau$  代表空间收缩变换

$$(\tau(\mu|_{D_k}))_{i_k, j_k} = \frac{1}{4} \cdot \{ (\mu|_{D_k})_{2i_k, 2j_k} + (\mu|_{D_k})_{2i_k+1, 2j_k} + (\mu|_{D_k})_{2i_k, 2j_k+1} + (\mu|_{D_k})_{2i_k+1, 2j_k+1} \} \quad (i_k, j_k = 0, \dots, 2^{r_k} - 1), \quad (4)$$

$\bar{\tau}(\mu|_{D_k})$  的大小与  $\tau(\mu|_{D_k})$  相同,  $\bar{\tau}(\mu|_{D_k})$  的每个元素都是  $\tau(\mu|_{D_k})$  所有像素的平均值. 式(3)中的减法运算便是正变化过程,  $S_k$  为由旋转和翻转构成的 8 种同构变换 (isometry).

以上假设和编码方法与 Oien 和 Lepsoy 的基本相同, 只是本文将空间收缩变换约束为  $2 \times 2$  平均, 而 Oien 和 Lepsoy 对此没有限制, 但本文这种约束是目前最常用的. 图 1 对以上有关符号做了更直观说明.

文[3]给出了如下结论: 对于任意给定的初始图像, 得到准确不动点所需迭代次数为

$$2 + \max_k \left[ \frac{r_k}{d_k - r_k} \right]. \quad (5)$$

如果初始图像取为 range 块的每个像素值等于相应的  $b_k$ , 则得到准确不动点所需迭代次数为

$$1 + \max_k \left[ \frac{r_k}{d_k - r_k} \right]. \quad (6)$$

其中初始图像算为一次迭代.

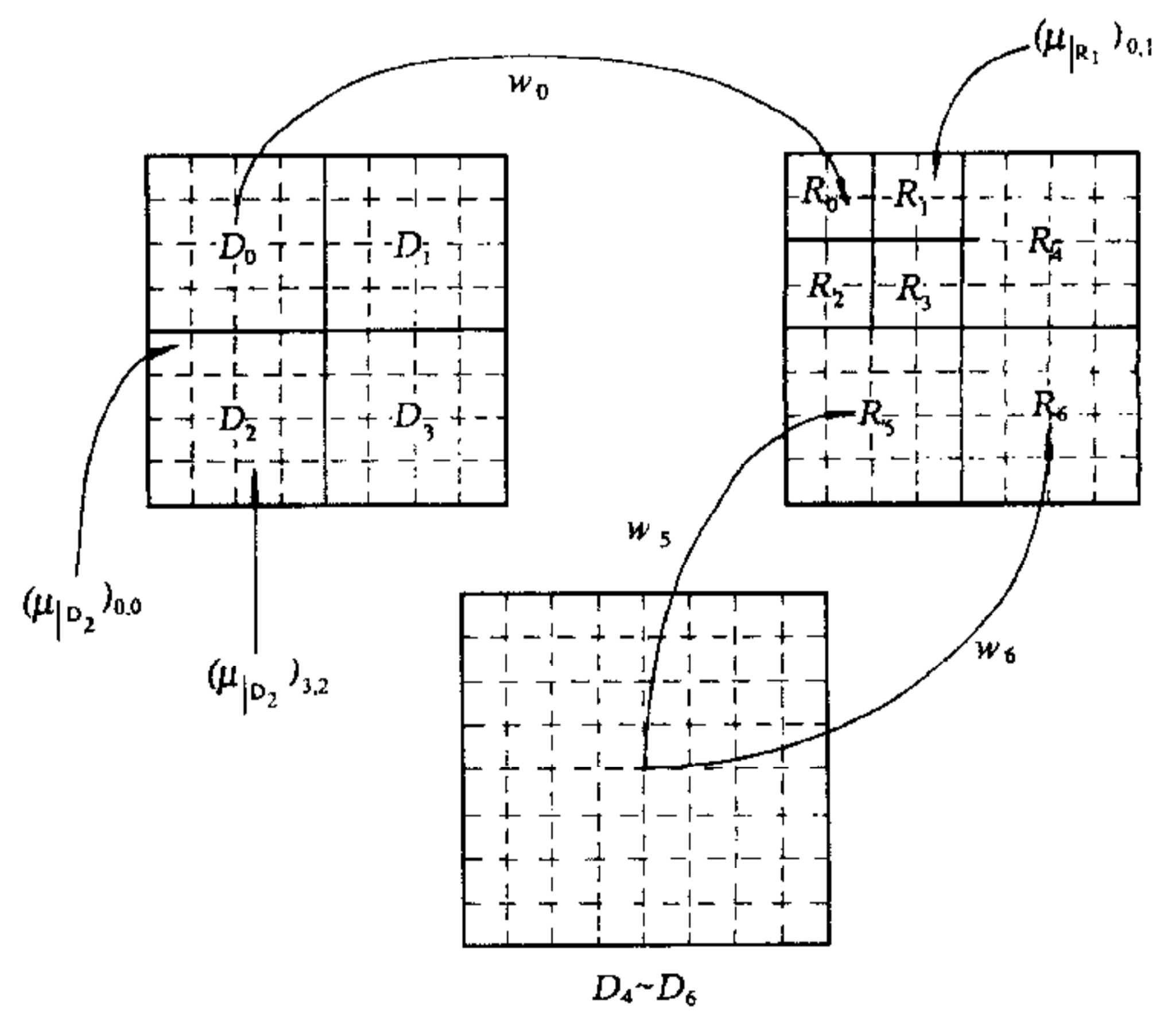


图 1 一个  $8 \times 8$  图像的 domain 块和 range 块

### 3 多分辨快速解码

前述约束编码方法所确定的 PIFS 仍然具有独立于分辨率的特性,对于任意  $N \times N$  像素初始图像,重复应用 PIFS 就得到变换  $W^1$  的不动点图像  $f^1(N \times N$  像素),对于任意  $\frac{N}{2} \times \frac{N}{2}$  像素初始图像,重复应用相同参数的 PIFS 就得到变换  $W^{\frac{1}{2}}$  的不动点图像  $f^{\frac{1}{2}}(\frac{N}{2} \times \frac{N}{2}$  像素).此时,与  $W^{\frac{1}{2}}$  相对应的 range 块大小为  $2^{r_k-1}(k=0, \dots, N_R-1)$ .文[4]中证明了如下几个性质.

**性质1.** 令  $f^1$  为  $W^1$  的不动点,则  $W^{\frac{1}{2}}$  的不动点为

$$f^{\frac{1}{2}} = \tau(f^1). \quad (7)$$

**性质2.** 令  $f^{\frac{1}{2}}$  为  $W^{\frac{1}{2}}$  的不动点,则  $W^1$  的不动点  $f^1$  为

$$(f^1_{|R_k^1})_{i_k, j_k} = a_k \cdot (s_k(f^{\frac{1}{2}}_{|D_k^{\frac{1}{2}}} - m(f^{\frac{1}{2}}_{|D_k^{\frac{1}{2}}}))_{i_k, j_k} + b_k \quad (i_k, j_k = 0, \dots, 2^{r_k} - 1). \quad (8)$$

这两个性质说明了 PIFS 不动点在不同分辨率下的关系.当  $r_k-1 > 0(k=0, \dots, N_R-1)$  时,前面的假设1)~4)仍然被满足,所以  $W^{\frac{1}{4}}$  的不动点  $f^{\frac{1}{4}}$  与  $W^{\frac{1}{2}}$  的不动点  $f^{\frac{1}{2}}$  之间的类似关系仍然成立,同时可以得出如下性质.

**性质3.** 在假设1)~4)的前提下,令  $r_{\min} = \min\{r_k, k=0, \dots, N_R-1\}$ ,则 PIFS 不动点构成的多分辨塔共有  $r_{\min} + 1$  级.

令  $r_{\max} = \max\{r_k, k=0, \dots, N_R-1\}$ ,根据 Oien 和 Lepsoy 的结论和本文假设,有如下结论.

**性质4.** 任意给定不同分辨率的初始图像,得到不动点  $f^{\frac{1}{2^k}}(k=0, \dots, r_{\min})$  所需要的迭代次数为

$$2 + r_{\max} - k, (i=0, \dots, N_R-1), \quad (9)$$

如果初始图像取为  $R_i^{\frac{1}{2^k}}(i=0, \dots, N_R-1; k=0, \dots, r_{\min})$  的每个像素值都等于  $b_k$ ,则得到不动点  $f^{\frac{1}{2^k}}(k=0, \dots, r_{\min})$  所需要的迭代次数为

$$1 + r_{\max} - k, (i=0, \dots, N_R-1), \quad (10)$$

其中初始图像算为一次迭代.

根据前面的性质,可以得到一种快速解码方法,即首先做  $2 + r_{\max} - r_{\min}$  次或  $1 + r_{\max} - r_{\min}$  次迭代得到  $f^{\frac{1}{2^{r_{\min}}}}$ ,然后利用性质2得到  $f^1$ .

### 4 与 Oien 和 Lepsoy 塔式方法的比较

在本文的假设下,上一节给出的快速解码方法与 Oien 和 Lepsoy 塔式方法最直接的区别是,前者适用于任意给定的初始图像,而后者只适用于初始图像取为  $R_i^{\frac{1}{2^{r_{\min}}}}(i=0, \dots, N_R-1)$  的每个像素值都等于  $b_k$  的情况.

下面讨论在初始图像取为  $R_i^{\frac{1}{2^{r_{\min}}}}(i=0, \dots, N_R-1)$  的每个像素值都等于  $b_k$  的情况下两种方法的差别.两种方法的多分辨塔都共有  $r_{\min} + 1$  级,如果  $r_{\max} = r_{\min} = r$ ,即所有 range

块大小相等,则两种方法所完成的运算过程完全相同,此时初始图像就是本文方法的  $f_{2^{r_{\min}}}^1$ ,而利用性质2每提升一级分辨率对应于 Oien 和 Lepsoy 方法在相应分辨率上的一次迭代. 如果  $r_{\max} > r_{\min}$ ,则本文方法首先在最低分辨率上做  $1+r_{\max}-r_{\min}$  次迭代得到  $f_{2^{r_{\min}}}^1$  (初始图像算为一次迭代),然后剩下的任务就是利用性质2提升分辨率  $r_{\min}$  次得到  $f^1$ . 而 Oien 和 Lepsoy 方法在最低分辨率上的任务仅仅是给出初始图像,除了在  $r_{\min}$  个更高的分辨率上进行  $r_{\min}$  次迭代以外,还要在最高分辨率上进行  $r_{\max}-r_{\min}$  次重复迭代,分辨率每提高一级,迭代一次所需计算量提高4倍. 因此,本文方法减少计算量主要体现为在最低分辨率上迭代  $r_{\max}-r_{\min}$  次(不算初始图像),而 Oien 和 Lepsoy 方法在最高分辨率上进行  $r_{\max}-r_{\min}$  次迭代,其余的计算对于两种方法是相同的. 这些差别的根源在于,Oien 和 Lepsoy 方法仅仅是去除了计算过程中的重复计算,而本文方法利用了不动点的多分辨率特性.

对于图1中  $8 \times 8$  图像及相应的块划分方法,利用 Oien 和 Lepsoy 塔式方法平均每个像素需要2次乘法和6次加法运算,而用本文方法平均每个像素需要1.25次乘法和3.75次加法运算,节省的计算量超过三分之一.

对一幅  $256 \times 256$  的256灰度级图像<sup>[4]</sup>,range 块由两级 quadtree 划分得到,其大小分别为  $8 \times 8$  (448个)和  $4 \times 4$  (2144个),domain 块大小分别为  $16 \times 16$  和  $8 \times 8$ ,所有 domain 块的左上顶点位于原图像中  $8 \times 8$  的网格上. 采用 Oien 和 Lepsoy 塔式方法解码,首先给出  $64 \times 64$  初始图像(range 块的每个像素值都等于相应的  $b_k$ ),然后各进行一次迭代分别得到  $128 \times 128$  和  $256 \times 256$  图像,最后再进行一次  $256 \times 256$  分辨率的迭代得到解码结果. 采用本文解码方法,用同样方法给出  $64 \times 64$  初始图像后,首先进行一次  $64 \times 64$  分辨率的迭代,然后利用性质2进行两次分辨率提升得到解码结果. 两种解码结果完全相同,前者用0.914秒,而后者需0.553秒. 用其他方法给出  $64 \times 64$  初始图像,只需在  $64 \times 64$  分辨率上再多进行一次迭代,得到相同结果.

## 5 结束语

对于引入了正交化运算的分形编码方法,其映射不动点在不同多分辨率下的关系与文[5]中结果类似,只是在由低分辨率不动点得到高分辨率不动点时也要加上正交化运算. 在此基础上,提出了一种快速分形图像解码方法,与 Oien 和 Lepsoy 方法相比,需要更少的计算量,计算量减少的程度取决于具体的块划分方式. 实验结果表明,对于 range 块的划分为二级 quadtree 的情况,本文方法比 Oien 和 Lepsoy 方法节省超过三分之一的计算量.

## 参 考 文 献

- 1 Jacquin A E. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. Image Processing*, 1992, **IP-1**(1):18~30
- 2 Fisher Y *et al.* Fractal Image Compression: Theory and Application, Berlin: Springer-Verlag, 1995
- 3 Oien G E, Lepsoy S. A class of fractal image coders with fast decoder convergence. *Fractal Image Compression: Theory and Application*, Berlin: Springer-Verlag, 1995. 153~176
- 4 孙怀江. 分形图像压缩的多分辨理论. 南京理工大学博士后工作报告, 1997
- 5 孙怀江, 杨静宇. PIFS 多分辨率特性研究. *计算机学报*, 1998, **21**(11):1042-1046

孙怀江 1968年生,博士,副教授. 研究兴趣包括图象处理、模式识别等.

杨静宇 1941年生,教授,博士生导师,计算机系主任. 研究兴趣包括人工智能、机器人等.