

基于分类重排 LZW 的图像无损压缩算法

谢耀华 汤晓安 孙茂印 张永亮

(国防科技大学电子科学与工程学院, 长沙 410073)

摘要 在遥感、医学等许多应用领域中, 出于对图像质量的要求, 图像一般必须进行无损压缩。针对现有方法的局限, 提出了一种无损压缩算法。该算法利用图像灰度分布对压缩比的影响, 首先对像素进行灰度分类并用掩膜图记录类别信息, 然后采用 Hilbert 曲线将各类像素进行块间和块内重排, 最后采用 LZW (Lempel-Ziv-Welch) 算法对掩膜图与各类像素的数据流进行编码。经过对多幅标准测试图像以及遥感图像的实验结果表明, 本文算法在总体上具有比 LZW、行程编码 RLE (run length encoding) 和霍夫曼 (Huffman) 等方法更高的压缩比, 并且易于实现。

关键词 灰度分类 掩膜图 Hilbert 曲线 重排 LZW 无损压缩

中图分类号: TP391 文献标志码: A 文章编号: 1006-8961(2010)02-0236-06

A Lossless Image Compression Algorithm Based on Classification, Re-ordering and LZW

XIE Yao-hua TANG Xiao-an SUN Mao-yin, ZHANG Yong-liang

(School of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073)

Abstract In many fields such as remote sensing or medicine, lossless compression is usually required because of the demand for high image quality. A lossless image compression algorithm is proposed in this paper considering the limitation of existing methods. The algorithm takes advantage of the effects of the distribution of pixels on compression ratio. The pixels are classified firstly, and the classification results are recorded in mask images. After that, the pixels of each class are re-ordered using the Hilbert curve. Finally, the mask image and the data stream of each class are coded using the LZW algorithm. Experiment was conducted on both standard testing images and remote sensing images, and the result showed that the proposed algorithm produced higher compression ratio than the methods of LZW, RLE and Huffman. The proposed algorithm is also easy to implement.

Keywords classification, mask image, Hilbert curve, re-ordering, Lempel-Ziv-Welch (LZW), lossless compression

0 引言

随着医学、遥感、GIS 等领域中的成像设备高速发展或普及, 每时每刻都有大量数字图像产生^[1]; 比如, 以 1 m 分辨率的遥感影像覆盖广东省, 就将产生约 1 TB 的数据, 而广东省的面积仅占全国的 1/53 左右^[2]。急速增长的图像数据如果不加以压缩, 将会消耗巨大的存储器资源。因此, 人们提出了

许多压缩算法^[3], 如矢量量化、小波压缩、分形压缩等方法^[4]。在许多领域如医学、军事、科研中, 对图像质量的要求很高, 往往要求在不造成图像失真的前提下尽可能减少图像占用的存储空间; 这就对无损压缩技术提出了强烈的需求。

常见的无损压缩方法有比特平面编码、行程编码 (又称游程编码, RLE)、霍夫曼 (Huffman) 编码、LZW (Lempel-Ziv-Welch) 编码、算术编码与无损预测编码等^[5]。无损压缩方法利用算法消除图像像素

收稿日期: 2008-10-13 改回日期: 2008-12-31

第一作者简介: 谢耀华 (1979—), 男, 国防科学技术大学信号与信息处理专业博士研究生。主要研究方向为海量图像存储与处理。

Email: fjnxyh2000@163.com

间的相关性, 从而减小平均每像素占用的比特数 (bpp), 但传统的方法都未考虑像素空间分布顺序对压缩比的影响, 即, 相同的图像如果以不同的顺序来排列像素, 则在同种压缩算法处理之后, 所得的 bpp 不尽相同。有学者研究表明, 对于某些空间同质性较高的图像, 采用 Hilbert 曲线重排之后能够比简单的行扫描方式节约存储空间 60% 左右^[6]。Khuri 等人曾提出了一种利用 Hilbert 曲线和行程编码相结合的算法, 获得了压缩比的提升, 但该方法只是简单的算法组合^[7]。Liang 等人提出了几种压缩方案^[8], 均为 Hilbert 空间填充曲线和某些经典的无损压缩方法的结合, 其中性能最好的方案是先进行差分编码, 再利用 Hilbert 曲线重排数据, 最后用 LZW 和 Huffman 先后编码; 该方法较好地挖掘了经典编码方法的压缩能力, 但未考虑到将像素进行分类内聚以进一步提高压缩比。除了利用 Hilbert 曲线之外, 还有学者提出了另外一些基于数据重排的方法^[9-10], 也是通过将数据重新排列来提高压缩效果, 但采用的不是空间填充曲线。

本文方法仍然基于 Hilbert 曲线, 但在重排之前, 首先按照灰度的相似性将像素划分成不同的类别, 重排的过程在各类像素中分别进行, 这样就增加

了像素的内聚程度, 达到进一步提高压缩比的目的; 为了精简掩膜图的数据量, 像素在分类后进行了分块合并。由于本算法的主要工作集中于编码的前期, 因而能够与不同的压缩算法结合衍生出多种类似的方法。

1 基于分类重排 LZW 的压缩算法

Gaede 等学者研究的结果表明^[6], 对于不同的数字图像, 即使直方图相同, 在采用相同方法进行压缩之后得到的 bpp 值也不尽相同。这是因为, 当图像相邻像素间的灰度值越接近, 也就是像素间的相关性越大时, 压缩算法越容易通过去除相关性而减小数据量。更进一步, 如果将像素按照灰度值分类, 然后将灰度值相近的像素聚集, 就能够在概率意义上增大像素间的相关性; 本文提出的方法正是基于这一思想。

1.1 算法流程

本文算法的主要步骤包括: 灰度分类、Hilbert 重排以及 LZW 编码, 其中, 灰度分类还包含着 $M \times M$ 合并的步骤。算法的总体流程如图 1 所示。不失一般性, 本文所讨论的原始图像均为 $S \times S$ 的 256 级灰度图像, 其中 S 满足: $S = 2^n$ ($n \in \mathbf{N}$ 且 $n \geq 2$)。

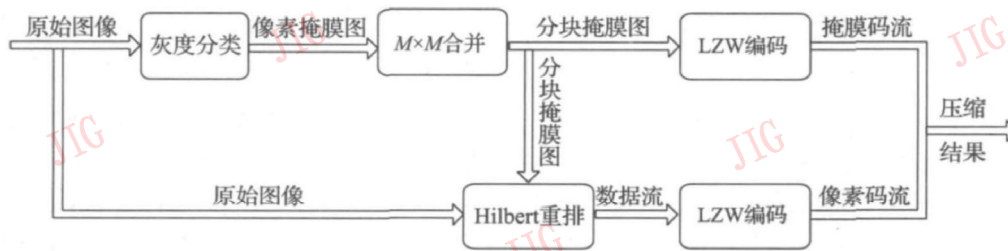


图 1 压缩编码流程

Fig 1 Flow of the proposed algorithm

1.2 灰度分类

灰度分类的任务是将图像中的像素划分为若干个类别, 使得同一类别中的像素灰度值互相接近。有不少技术可以用于实现像素分类; 为了降低算法的复杂度, 本文采用灰度阈值的方法。同时为了减少类别信息, 所划分的类别确定为两类。

由于原始图像的不确定性, 如果采用恒定阈值, 可能会导致某个类别的像素数太少或太多, 影响压缩效果。比较合理的方法是根据图像的直方图自适应的确定阈值, 使得两类像素数量接近。设原始图

像的直方图为

$$H = \{C_g \mid C_g \in \mathbf{Z} \text{ 且 } C_g \in [0, S^2], g = 0, 1, \dots, 255\}$$

容易找到使如下 $f(G)$ 取最小值的阈值 G :

$$f(G) = \left| \sum_{g < G} C_g - \sum_{h > G} C_h \right| \quad (1)$$

式中, $C_g, C_h \in H; g, h \in [0, 255]$. 在确定了阈值 G 之后, 即可将像素按如下准则划分为两类:

$$\begin{cases} I(i, j) \in C_1 & I(i, j) \geq G \\ I(i, j) \in C_2 & I(i, j) < G \end{cases} \quad (2)$$

$i, j = 0, 1, \dots, S - 1$

上式中, $I(i, j)$ 为原始图像在第 i 行第 j 列的像素值, C_1 与 C_2 分别代表高灰度与低灰度两个类别的像素集合。

根据上述准则, 可以得到一幅二值的像素掩膜图 M_p , 大小为 $S \times S$, 其像素按如下准则取值:

$$\begin{cases} M_p(i, j) = 1 & I(i, j) \in C_1 \\ M_p(i, j) = 0 & I(i, j) \in C_2 \end{cases} \quad (3)$$

$i, j = 0, 1, \dots, S-1$

M_p 记录了所有像素的类别信息, 但它本身所占的比特数也较多。可以将其分块合并, 每块大小为 $M \times M$, ($M = 2^m, m = 2, 3, \dots$), 得到一张分块掩膜图 M_b 。对于每一个 $M_b(i, j)$, $i, j = 0, 1, \dots, SM-1$ 在 M_p 中均有一个像素块 B_{ij} 与之对应, 即

$$B_{ij} = \{M_p(u, v) \mid u = i \cdot M, i \cdot M + 1, \dots, i \cdot M + M - 1, v = j \cdot M, j \cdot M + 1, \dots, j \cdot M + M - 1\}$$

统计 B_{ij} 中的像素值, 若值为 1 的像素数不少于值为 0 的像素数, 则将对应的 $M_b(i, j)$ 置为 1 并将 B_{ij} 中所有像素划归 C_1 ; 否则将 $M_b(i, j)$ 置 0 并将 B_{ij} 中所有像素划归 C_2 。特别地, 当 $M = 1$ 时, M_p 与 M_b 相同。

通过灰度分类, 实质上将像素划分为两个互不重叠的子集, 每个子集中出现的符号(灰度值)种数远小于原始图像, 从而缩减了编码所需的理论上限。但分类数不能太多, 否则掩膜图将占用过多的存储空间。

1.3 Hilbert重排

为了使灰度值相近的像素在编码序列中聚集, 在对像素进行分类之后, 还要将两类像素重新排列。要通过重排使相近的像素聚集, 同时又不必存储额外的信息, 可以利用 Hilbert 曲线来实现。该曲线由德国数学家 Hilbert 于 1891 年提出^[11], 其特点是能够遍历 $2^n \times 2^n$, ($n = 2, 3, \dots$) 的点阵, 并使得点阵中相邻

的点被排列为 1 维上位置相近的点。Jagadish 的研究表明, Hilbert 曲线是具有最佳内聚特性的空间填充曲线^[12]。图 2 为 16 阶 Hilbert 曲线示意图。

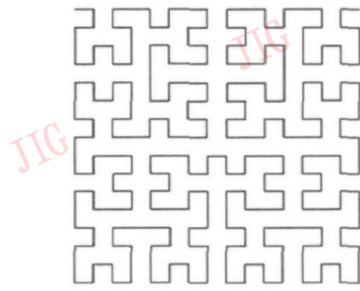


图 2 16 阶 Hilbert 曲线
Fig 2 Hilbert curve with a rank of 16

如 1.2 节所述, 为了减少像素掩膜图占用的空间, 将其分块合并, 每块大小为 $M \times M$, ($M = 2^m, m = 2, 3, \dots$), 总的块数为 $(SM) \times (SM)$ 。因此, 首先应进行块间重排, 即确定 $(SM) \times (SM)$ 个块之间的排列顺序。按照 SM 阶 Hilbert 曲线的顺序遍历分块掩膜图, 将所有块排成一个 1 维序列:

$$B_{ex} = \{M_b(u_0, v_0), \dots, M_b(u_{\lfloor \frac{SM}{M} \rfloor - 1}, v_{\lfloor \frac{SM}{M} \rfloor - 1})\},$$

$$u_m, v_m \in \left\{ 0, 1, \dots, \frac{S}{M} - 1 \right\},$$

$$m = 0, 1, \dots, \left\lfloor \frac{S}{M} \right\rfloor^2 - 1 \quad (4)$$

每个块中还含有 $M \times M$ 个像素, 则又利用 M 阶 Hilbert 曲线对其进行块内重排, 所有的块按照相同的方式进行; 设图像块为 B_{lk} , 则重排后得到一个 1 维序列为

$$B_{in} = \{B_{lk}(s_0, t_0), \dots, B_{lk}(s_{M^2-1}, t_{M^2-1})\},$$

$$s_n, t_n \in \{0, 1, \dots, M-1\}, n = 0, 1, \dots, M^2-1 \quad (5)$$

结合块间重排与块内重排, 即可将整个图像的所有像素排列成一个 1 维序列, 如式 (6) 所示:

$$D = \left[\begin{array}{l} I(u_0M + s_0, v_0M + t_0), \dots, I(u_0M + s_{M^2-1}, v_0M + t_{M^2-1}), \\ \dots \\ I(u_mM + s_0, v_mM + t_0), \dots, I(u_mM + s_{M^2-1}, v_mM + t_{M^2-1}), \\ \dots \\ I(u_{\lfloor \frac{SM}{M} \rfloor - 1}M + s_0, v_{\lfloor \frac{SM}{M} \rfloor - 1}M + t_0), \dots, I(u_{\lfloor \frac{SM}{M} \rfloor - 1}M + s_{M^2-1}, v_{\lfloor \frac{SM}{M} \rfloor - 1}M + t_{M^2-1}) \end{array} \right] \quad (6)$$

由于像素已被分成了两个类别, 只需顺序遍历序列 D 中的数据, 按式 (7) 中的准则即可将其分解为两个子序列 D_1 与 D_2 , 分别对应于高、低灰度值。 $\forall I(i, j) \in D, i, j = 0, 1, \dots, S-1$ 分解的准则如下:

$$\begin{cases} I(i, j) \in D_1 & M_b(i', j') = 1 \\ I(i, j) \in D_2 & M_b(i', j') = 0 \end{cases} \quad (7)$$

$$i' = \left\lfloor \frac{i}{M} \right\rfloor; \quad j' = \left\lfloor \frac{j}{M} \right\rfloor$$

1.4 LZW 编码

LZW 编码是一种基于“字典”的压缩方法。它使用固定长度的码字对相继出现的、由单个信源符号所构成的、长度可变的符号序列进行编码, 而不依赖于待编码信源符号出现的先验知识。LZW 是一种重要的无损压缩编码方法, 已经成功的应用于 GIF, TIFF, PDF 等多种文件格式中^[5]。

本文采用 LZW 算法对 1.3 节得到的数据流 D_1 与 D_2 进行压缩编码。对于分块掩膜图 M_b , 则首先将连续 8 个像素合并成一个字节, 然后对所得的字节流进行 LZW 编码。原图像 I 的信息完全包含在 D_1 , D_2 与 M_b 中。

解码的过程基本上是编码的逆过程, 先利用 LZW 解码算法恢复出 D_1 , D_2 与 M_b ; 然后, 按照块间 Hilbert 曲线的顺序遍历 M_b , 每次根据当前点的 M_b 值, 从 D_1 或 D_2 取出 $M \times M$ 个像素, 并按照块内 Hilbert 曲线顺序填充到原图像的当前块上。遍历完成之后, 即得到原始图像 I 。

2 实验与分析

为了检验本文算法的压缩性能, 利用 Matlab 7.0 和 ArkadiKagan 提供的 C++ 源程序 compression utility 实现了文中算法。选取分块合并的尺寸 $M = 1$, 因此, 像素掩膜图与分块掩膜图完全相同。此时, 像素重排的过程也随之而略为简化。LZW, RLE 和 Huffman 算法是 3 种经典而优秀的无损压缩方法, 至今仍在广泛使用, 并且能够根据文献准确重复其实现; 因此, 实验中将与这 3 种方法进行对比分析。

用于测试的图像共有 9 幅, 如图 3(a)、图 4 所示, 内容涵盖人像、静物、景观等。其中 7 幅为业内广泛使用的标准测试图像, 分别为 Cameraman, Lake, Lena, Livingroom, Peppers, Walkbridge, Woman。余下 2 幅 Lanier 与 DragonLake 为遥感影像图像。以上图像均为 256×256 灰度级为 8bpp。

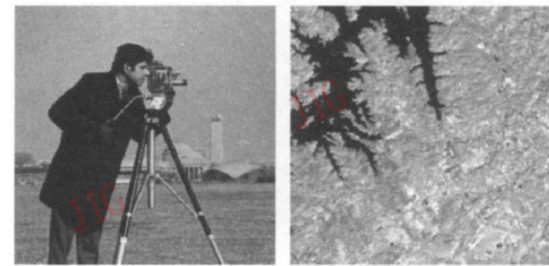
所有的测试图像首先经过灰度分类, 得到像素掩膜图 (亦即分块掩膜图); 然后采用 256 阶 Hilbert 曲线进行像素重排, 最终得到高、低灰度的两个数据流。图 3 显示了 Lena 的原始图像与重排后的图像 (此图像的上半部分由序列 D_1 中的数据构成, 下半部分由序列 D_2 中的数据构成)。可以看出, 经过重排之后, 同一扫描行的相邻像素值更加接近, 对后续的压缩过程更加有利。



(a) Lena 的原始图像 (b) 重排后的数据

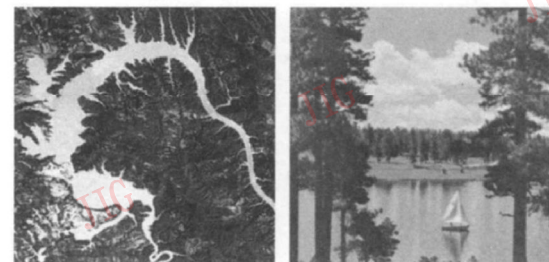
图 3 重排结果

Fig. 3 The result of re-ordering



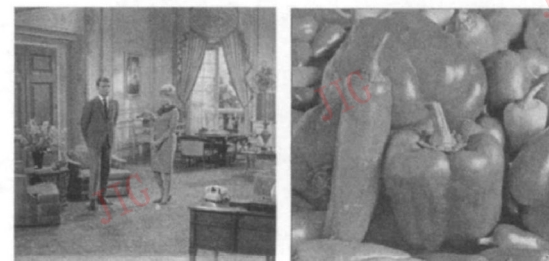
(a) Cameraman

(b) Lanier



(c) DragonLake

(d) Lake



(e) Livingroom

(f) Peppers



(g) Walkbridge

(h) Woman

图 4 其余原始测试图像

Fig. 4 The rest of the test image

将 Lena 在内的所有测试图像编号如下:

- 1 - Cameraman
- 2 - Lanier
- 3 - DragonLake

- 4 - Lake
- 5 - Lena
- 6 - Livingroom
- 7 - Peppers
- 8 - Walkbridge
- 9 - Woman

通过实验测得所有测试图像在不同算法下的单位像素占用的比特数。如图 5 所示:

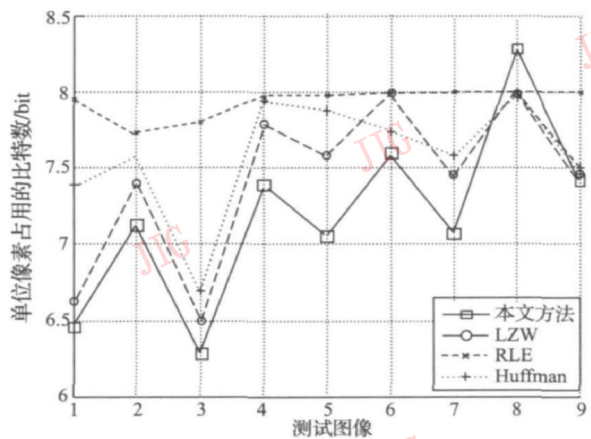


图 5 测试图像在不同算法下单位像素占用的比特数比较

Fig 5 Bpp comparison of images by different algorithms

从图中可以看出,在 3 种现有的无损算法中, LZW 算法的压缩性能最为优秀,在对大多数图像的压缩结果中单位像素占用最少的比特数;而本文改进之后的压缩结果在绝大多数情况下单位像素占用的比特数比 LZW 更少,是所有算法中最小的。本算法平均每像素所占用的比特数比起 RLE 算法减小了 10.46%,获得了较大的性能提升。所有的实验图像中,只有极少数图像 (Walkbridge) 经过压缩,反而单位像素占用的比特数较多;观察 Walkbridge 可以看出,该图像的内容凌乱,高频分量过于丰富,相邻像素之间的相关性非常小,图像的信号特征在多个局部范围内与噪声图像接近,这在实际应用所涉及的图像中较不常见。

为了检验本算法的运行效率,在处理器为 AMD Athlon(tm) 64 × 21.7 GHz、内存为 1 G 的便携式计算机上运行算法的 Matlab 程序 (代码未做任何优化),得到压缩与解压过程所需的时间 (不含 LZW 编码步骤),如表 1 所示。可以看出,编解码过程耗时接近并且都不多,但在后续的研究中还是有必要进一步改善。

表 1 压缩与解压过程花费的时间

Tab 1 Time consumption of compression and decompression

过程	图 像								
	1	2	3	4	5	6	7	8	9
压缩	1.902 5	1.853 9	1.905 1	1.906 6	1.853 7	1.830 7	1.886 9	1.905 3	1.863 1
解压	1.765 3	1.810 8	1.731 7	1.743 9	1.750 3	1.749 9	1.790 4	1.757 2	1.749 9

3 结 论

本文利用灰度分类技术和 Hilbert 曲线,扩展了 LZW 编码技术,得到一种新的图像无损压缩算法。首先通过灰度分类,缩小了像素的取值空间,提高了像素间的内聚程度;进而通过块间、块内 Hilbert 重排使得灰度相近的像素进一步聚集;最后采用 LZW 方法对图像数据流和掩膜数据流进行编码。对标准测试图像以及普通遥感图像的实验结果表明,本算法易于实现并能够以更小的 bpp 压缩各类图像,其压缩性能总体上优于 LZW, RLE 和 Huffman 方法;对于实验中的图像,本算法平均每像素所占用的比

特数比起 RLE 算法减小了 10% 以上。为了达到提高压缩性能的目的,算法的压缩和解压过程增加了一定的时间开销,但并不显著。

本算法具有较强的可扩展性,其最后一个步骤除了采用 LZW 之外,也可以采用其他的编码算法,从而衍生出多种算法;当然,对于不同的编码算法分类及重排的最佳方案也应不同,需要进一步研究。虽然数据重排对于自然图像压缩的积极作用已经得到前人的证明,但对不同图像的效果有所不同;具备怎样特点的图像适合进行怎样的重排,其内在的机理还应进一步深入研究。此外,还可进行的工作是在保证压缩性能的前提下优化流程,进一步提高运行效率。

参考文献 (References)

- [1] Enrico Magli, David Taubman. Image compression practices and standards for geospatial information systems [C] // Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS'03). Toulouse, France: IEEE Press, 2003: 654-656.
- [2] Li De-ren. Digital earth and "3S" technology [J]. China surveying, 2003(2): 28-31. [李德仁. 数字地球与"3S"技术 [J]. 中国测绘, 2003(2): 28-31.]
- [3] An Xiao-dong, Chen Jing. A summarization of image compression methodology [J]. Computer development and application, 2006, 19(12): 24-26. [安晓东, 陈静. 图像压缩方法综述 [J]. 电脑开发与应用, 2006, 19(12): 24-26.]
- [4] Chen Chuan-Chun. On the selection of image compression algorithm [C] // Proceedings of the 14th IEEE International Conference on Pattern Recognition, Brisbane, Australia: IEEE Press, 1998, 2: 1500-1504.
- [5] Zhang Chun-tian, Su Yu-ting, Zhang Jing. Digital Image Compression [M]. Beijing: Tsinghua Publishing House, 2006. [张春田, 苏育挺, 张静. 数字图像压缩编码 [M]. 北京: 清华大学出版社, 2006.]
- [6] Gaede V, Günther O. Multidimensional access methods [J]. ACM Computing Surveys, 1998, 30(2): 170-231.
- [7] Khuri S, Hsu H C. Interactive packages for learning image compression algorithms [C] // Proceedings of the 5th Annual Conference on Innovation and Technology in Computer Science Education. New York, NY, USA: ACM, 2000: 73-76.
- [8] Liang JY, Chen C S, Huang C H, et al. Lossless compression of medical images using hilbert space-filling curves [J]. Computerized Medical Imaging and Graphics, 2008, 32(3): 174-182.
- [9] Ng K S, Cheng L M. Image compression system using data reordering technique [J]. Electronics Letters, 2003, 39(14): 1046-1047.
- [10] Slobodan Vucetic. A fast algorithm for lossless compression of data tables by reordering [C] // Proceedings of the Data Compression IEEE Conference (DCC'06). Snowbird, UT, USA: IEEE Computer Society, 2006: 469-469.
- [11] Qi Dong-xu. Fraction and Its Generation by Computer [M]. Beijing: Science Publishing House, 1994. [齐东旭. 分形及其计算机生成 [M]. 北京: 科学出版社, 1994.]
- [12] Jagdish H V. Linear clustering of objects with multiple attributes [C] // Proceedings of ACM SIGMOD International Conference on Management of Data. Atlantic City, New Jersey, USA: ACM Press, 1990: 332-342.