

基于 DCT 变换的快速分形编码方法

何 佳, 刘政凯

(中国科学技术大学电子工程与信息科学系, 安徽合肥 230027)

摘 要: 本文提出了一种基于 DCT 变换加速相似块匹配的分形编码方法. 通过图像块与相似块在 DCT 变换域少数低频系数的比较, 来确定匹配的相似块. 并根据 DCT 变换的性质, 方便的确定对比度因子和亮度偏移量. 并采用将图像块与相似块的 DCT 变换系数预制成查找表的方法减少重复运算. 结果表明, 编码时间大幅度减少, 而 PSNR 只是略有下降.

关键词: 分形编码; DCT 变换; 查找表

中图分类号: TN919.81

文献标识码: A

文章编号: 0372-2112 (2001) 06-0748-03

A Fast Fractal Image Coding Method Based on the DCT Transform

HE Jia, LIU Zheng-kai

(Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, Anhui 230027, China)

Abstract: This paper presents a fractal image coding method based on the DCT transform that speeds up the matching procedure in the sets of domain blocks. This method compares the few low frequency DCT transform coefficients of the range blocks and domain blocks to find out the matched domain block. In accordance with the properties of the DCT transform, it calculates the contrast scaling and luminance shift conveniently. At the same time, it stores the DCT transform coefficients of the range blocks and domain blocks into a lookup table to reduce the redundancy operations. The result shows that the coding time is reduced evidently, while PSNR is only reduced very little.

Key words: fractal image coding; DCT transform; lookup table

1 引言

分形图像编码是图像编码的一个崭新领域. Barnsley 首先提出了以迭代函数系统 IFS (Iterated Function System) 对图像进行编码的思想^[1]. 其后, Jacquin 以局部 IFS 映射的思想实现了基于块的自动分形图像编码系统^[2]. 从此, 分形编码受到了广泛的关注与研究.

分形编码中, 设 I 是灰度取值 $[0, g]$ 的图像, 则 $I \subset R^3$. 首先将图像划分为不重叠的子块 R_i , $R_i \cap R_j = \emptyset, i \neq j, R_i \cup R_j = I$, R_i 称为图像块 (Range Block), $i = 1 \dots m$, 并设 D_j 是图像的一个子块, 称为相似块 (Domain Block), $j = 1 \dots n$. 设:

$F_j = \{f: D_j \rightarrow R^3; f \text{ 是一个仿射变换}\}$

则对于 R_i , 令 $f_{ij}: D_j \rightarrow F_j$ 使得 $d(R_i, f_{ij}(D_j)) \leq d(R_i, f(D_j)), \forall f \in F_j, \forall j$. 其中 d 表示图像块与相似块经过仿射变换后的距离.

设 k 满足 $d(R_i, f_{ik}(D_k)) = \min\{d(R_i, f_{ij}(D_j))\}, \forall j$. 分形编码则在于对 R_i , 寻找 $f_{ik}(D_k)$.

仿射变换 f 可以写成一个矩阵变换:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e \\ w \\ o \end{bmatrix}$$

其中 x, y 是每一相似块内某个像素的坐标. z 是该像素的灰度值. x, y 是变换后的像素坐标. z 是变换后像素的灰度值. a, b, c, d, s, e, w, o 是分形编码.

分形编码方法的基本特点是在图像中搜寻与当前编码的图像块最近似的相似块, 因此必然要进行大量的匹配运算. 一般学者都采取了块分类的方法来限制相似块的搜索范围. 如 Jacquin 提出将块分为三类: Shade、Edge、Midrange, 图像块与相似块仅在每一类集合的内部进行匹配. Fisher^[3]等人提出一种更精细的根据亮度及亮度变化来分类的方法: 把图像块和相似块等分成四个子块, 并分别计算每个子块的像素亮度平均值和相应的亮度方差值. 根据亮度平均值的分布情况将块分为 3 大类. 在此基础上, 再按方差值的 24 种排列分成 24 类. 这样就把所有图像块相似块分成了 72 类. 从而极大提高了编码速度. 以上方法的思想是通过抽取图像块与相似块的特征, 排除特征不相符的相似块来缩小搜索范围, 而对特征相同的图像块与相似块是否匹配的判定, 基本上都是基于逐点比较. 选取 MSE (mean square error) 最小的相似块作为搜索结果. 这个步骤仍然运算量很大. 匹配时能否通过少量的运算即判定当前图像块和当前相似块是否匹配? 本文通过 DCT 变换进行了一些探索. 严格讲, 相似块的尺寸要经过 Geometric 变换

才能等于图像块的尺寸. 在以后讨论中, 为简单起见, 相似块均指其经过 Geometric 变换后的结果.

2 快速分形编码

2.1 快速算法的原理

图 1 显示了图像子块进行 DCT 变换后, 低频系数表征图像子块特征的能力. (a) 是原图, (b) 是原图分成 8×8 的子块进行 DCT 变换后只取 3 个低频系数再进行 DCT 反变换所得图像. (c) 类似 (b) 的处理过程, 但是取 12 个低频系数. 可以观察到: (b) 已经基本反映了原图的概貌, (c) 图中细节部分得到加强. 这个例子说明对于 DCT 变换, 少数的低频系数就能表征图像的特征.



图 1 DCT 系数表征图像的作用比较

因此提出算法基本思想是: 仅用图像块与相似块变换域中少数几个低频 DCT 系数进行比较. 并且根据图 1 的结果认为: 只要所比较的 DCT 系数较为相似, 则对应的图像块与相似块也较为相似. 判断 DCT 系数相似的标准是根据 MSE 最小原则. 当得到匹配的相似块后再求取其它参数, 从而完成当前图像块的分形编码.

设 $f(j, k)$ 是尺寸为 $N \times N$ 图像中 (j, k) 点的亮度值, $j = 0 \dots N-1, k = 0 \dots N-1$; $F(u, v)$ 是 DCT 变换后 (u, v) 处的系数值, $u = 0 \dots N-1, v = 0 \dots N-1$; 则有

DCT 正变换:

$$F(u, v) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j, k) \sqrt{2} C(u) \cos\left[\frac{\pi}{N} u \left(j + \frac{1}{2}\right)\right] \sqrt{2} C(v) \cos\left[\frac{\pi}{N} v \left(k + \frac{1}{2}\right)\right] \quad (1)$$

DCT 反变换:

$$f(j, k) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \sqrt{2} C(u) \cos\left[\frac{\pi}{N} u \left(j + \frac{1}{2}\right)\right] \sqrt{2} C(v) \cos\left[\frac{\pi}{N} v \left(k + \frac{1}{2}\right)\right] \quad (2)$$

其中:

$$C(0) = 1/\sqrt{2}, C(w) = 1, w = 1, 2, \dots, N-1$$

设, 存在两个图像 R 和 D , 尺寸均为 $N \times N$. 像素 (i, j) 处的亮度值分别用 $r(i, j)$ 和 $d(i, j)$ 表示. 若对于所有的 $i = 0 \dots N-1, j = 0 \dots N-1$; 都有: $r(i, j) = sd(i, j) + g$ (3) 其中 s, g 是常量.

记 $r_{\text{dct}}(u, v)$ 为 R 经过 DCT 变换后 (u, v) 处的系数; $d_{\text{dct}}(u, v)$ 为 D 经过 DCT 变换后的 (u, v) 处的系数. 则有以下式 (4)、(5) 两条可以简化运算的性质:

$$g = [r_{\text{dct}}(0, 0) - sd_{\text{dct}}(0, 0)] / N \quad (4)$$

$$\text{当 } u, v \text{ 不同时为 } 0 \text{ 时: } r_{\text{dct}}(u, v) = sd_{\text{dct}}(u, v) \quad (5)$$

证明:

$$\begin{aligned} r_{\text{dct}}(u, v) &= \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} r(j, k) \sqrt{2} C(u) \cos\left[\frac{\pi}{N} u \left(j + \frac{1}{2}\right)\right] \sqrt{2} C(v) \cos\left[\frac{\pi}{N} v \left(k + \frac{1}{2}\right)\right] \\ &= \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} (sd(j, k) + g) \sqrt{2} C(u) \cos\left[\frac{\pi}{N} u \left(j + \frac{1}{2}\right)\right] \sqrt{2} C(v) \cos\left[\frac{\pi}{N} v \left(k + \frac{1}{2}\right)\right] \\ &= sd_{\text{dct}}(u, v) + \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} g \sqrt{2} C(u) \cos\left[\frac{\pi}{N} u \left(j + \frac{1}{2}\right)\right] \sqrt{2} C(v) \cos\left[\frac{\pi}{N} v \left(k + \frac{1}{2}\right)\right] \end{aligned}$$

当 u, v 同时为 0 时 $r_{\text{dct}}(u, v) = sd_{\text{dct}}(u, v) + Ng$, 则式 (4) 成立. 当 u, v 不同时为 0 时, 因为

$$\begin{aligned} &\sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \cos\left[\frac{\pi}{N} u \left(j + \frac{1}{2}\right)\right] \cos\left[\frac{\pi}{N} v \left(k + \frac{1}{2}\right)\right] \\ &= \sum_{j=0}^{N-1} \cos\left[\frac{\pi}{N} u \left(j + \frac{1}{2}\right)\right] \sum_{k=0}^{N-1} \cos\left[\frac{\pi}{N} v \left(k + \frac{1}{2}\right)\right] = 0 \end{aligned}$$

所以

$$r_{\text{dct}}(u, v) = sd_{\text{dct}}(u, v)$$

式 (5) 成立.

证毕

当匹配的相似块确定后, 若将 R 看成图像块, D 看成相似块, s 看成对比度因子 (Contrast scaling), g 看成亮度偏移量 (Luminance shift), 则图像块与相似块的仿射变换关系和式 (3) 完全相同. 因此可以认为式 (4)、(5) 成立. 式 (4) 的作用使得亮度偏移值的计算将非常简单. 式 (5) 说明, 根据 MSE 最小原则进行匹配前, 需对相似块的低频 DCT 系数乘以对比度因子, 然后进行的比较才有意义 (直流系数不进行比较).

另外, DCT 变换后能量集中在低频分量, 量化时一般采用 JPEG 标准的 Zig-Zag 形排序方法, 具体细节不再赘述.

2.2 查找表加速

通过研究发现, 匹配过程中存在着大量的重复运算. 如: 每次比较时都要进行 DCT 变换. 因此考虑将图像块与相似块的 DCT 变换结果预先制成查找表, 每次匹配比较时, 只需查表即可. 另外, 由于并不需要计算所有的 DCT 系数, 所以从运算时间考虑没有必要使用 DCT 快速算法, 只需简单的直接根据式 (1) 计算所需的系数, 从而简化了运算的复杂性.

具体实现时, 定义两个多维数组:

$DATA_{\text{DCTDOMAIN}} : \text{Array}[0..3, 0..SUBIMGDOMAINPNUM-1, 0..SUBIMGRANGE-1, 0..DCTCOEFFNUM-1]$ of real;
 $DATA_{\text{DCTRANGE}} : \text{Array}[0..3, 0..SUBIMGRANGE-1, 0..SUBIMGRANGE-1, 0..DCTCOEFFNUM-1]$ of real;
 其中, $DATA_{\text{DCTDOMAIN}}$ 记录相似块集合的 DCT 变换系数, $DATA_{\text{DCTRANGE}}$ 记录图像块集合的 DCT 变换系数. 类似 Jacquin 的方法, 将原图分成四个子图. 数组第一维 $[0..3]$ 用来区分这四个子图. 第二维和第三维用来确定子块的位置, 其中 $SUBIMGDOMAINPNUM$ 是每个子图相似块总个数的平方根. $SUBIMGRANGE-1$ 是每个子图图像块总个数的平方根. 第四维用来区分特定子块中不同的 DCT 系数. $DCTCOEFFNUM$ 是 DCT 变换后用来匹配的系数个数.

2.3 算法实现

图 2 是算法实现的流程图.

在实践中, Jacquin 的分类方法使图像块与相似块很少会被分到 Midrange 类, 这一结果也被其他学者的研究所证实^[4]. 因此, 在分类中仅将块分类按亮度变化程度分成均匀块与非均匀块. 对均匀块按 Shade 处理. 实践中, 当尝试对非均匀块按 Edge 处理时, 压缩比有所下降, 但 PSNR 却不一定提高. 因此, 对非均匀块按 Midrange 处理.

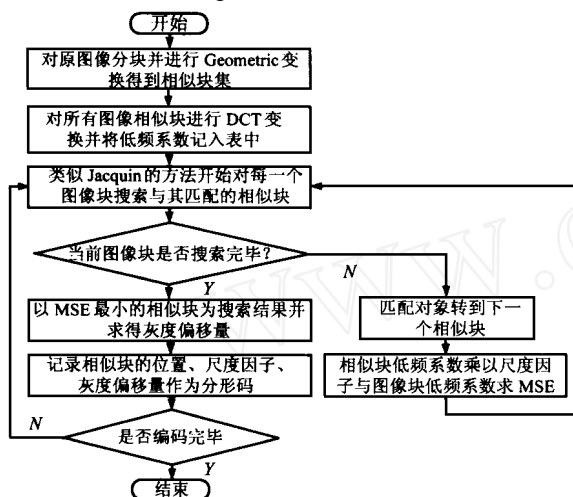


图2 快速算法流程图

3 实验结果

实验用 256 × 256 的 8bit 灰度图像 Lena, CPU: PII266, 内存: 32M. 为使算法简明, 图像块、相似块取一种尺寸.

表 1 和表 2 显示了 Jacquin 原始算法和快速算法的数值结果. 图 3 是实际的编码结果.

表 1 Jacquin 原始算法结果

图像块尺寸(像素)	编码时间(秒)	PSNR(dB)	压缩比
4 × 4	178	30.17	8.63
8 × 8	63	25.62	34.00

表 2 快速算法结果

图像块尺寸(像素)	DCTCOEFFNUM	编码时间(秒)	PSNR(dB)	压缩比	查找表大小(KBytes)
4 × 4	4	2.5	27.76	10.44	160
4 × 4	8	3.5	28.73	10.44	320
4 × 4	16	5.5	29.07	10.44	640
8 × 8	4	1.0	24.03	42.20	40
8 × 8	8	1.5	24.50	42.20	80
8 × 8	16	2.5	24.79	42.20	160



(a) 4×4子块DCTCOEFFNUM=16的快速算法
信噪比 29.07dB 压缩比 10.44



(b) 4×4子块Jacquin的方法
信噪比 30.17dB 压缩比 8.63

图3 快速算法与Jacquin算法结果比较

从实验结果可以看出:

(1) 快速算法的编码时间(表 2 编码时间包含预制查找表的时间)大幅度减少.

(2) 随着 DCTCOEFFNUM 的增加, PSNR 有所增加, 但总是略小于 Jacquin 的结果. 这是因为快速算法的匹配结果是次最优匹配, 而不是 Jacquin 的最优匹配.

(3) 压缩比略有增加. 这是因为按 Midrange 处理非均匀块时舍去了 Isometric 变换的缘故.

(4) 随着 DCTCOEFFNUM 的增加, 查找表的大小也成比例增加. 但是 PSNR 的增加并不明显. 因此对于 4 × 4 子块, DCTCOEFFNUM 取 8 较为合适. 对于更大尺寸子块, DCTCOEFFNUM 取 12 到 16 之间的值较为合适.

4 结论

对于本文提出的压缩算法, 因为搜索到的相似块不一定是图像域 MSE 最小意义上的最佳匹配, 图像质量有所下降, 但速度极大提高. 观察图 3 的结果, 快速算法对于 4 × 4 子块与 Jacquin 所提出算法压缩图像主观上几乎不能分辨优劣. 因此本文所提出的算法可以认为是一种可供选择的折衷算法. 另外, 在编码中由于将 DCT 变换结果预制成查找表, 因此要占用一定的内存, 但是随着计算机硬件的迅速发展, 用内存空间换取编码时间的方法也将成为可行的选择. 并且本算法也容易与其它块分类算法结合使用, 以期得到更高的速度.

参考文献:

- [1] Barnsley M F. Fractal Everywhere [M]. New York: Academic Press, 1988.
- [2] Jacquin A. Fractal image coding: A Review [A]. Proc of the IEEE [C]. 1993, 81(10): 1451 - 1465.
- [3] Jacobs E W, Fisher Y, Boss R D. Image compression: A study of iterated transform method [J]. Signal Processing 1992, 29: 251 - 263.
- [4] 何爱军, 马争鸣. 分形图像编码. 中国图象图形学报 [J]. Mar. 1999, 4(A) (3): 188 - 192.

作者简介:



何佳男. 1976 年生于河北藁城. 中国科学技术大学电子工程与信息科学系信号与信息处理专业博士研究生. 目前主要研究方向为图像压缩、分形几何及小波分析.



刘政凯. 男. 1940 年生于辽宁沈阳. 中国科学技术大学教授, 博士生导师. 主要从事遥感图像处理, 人工神经网络及模式识别方面的研究. 已发表论文 60 余篇, 出版专著 6 本.