

基于整型 DCT 变换的图象编码研究

韦长江 郝鹏威 石青云

(北京大学视觉与听觉信息处理实验室, 北京 100871)

摘要 为了既能保证图象无失真压缩, 又能提高编码效率, 给出了一种新的基于整型可逆 DCT 变换的嵌入式编码方法, 并实现了基于 DCT 的无失真编码和有失真编码方案的完美结合. 该算法是首先通过计算 DCT 浮点变换的整数可逆矩阵分解, 从而得到可以整数实现的可逆矩阵变换; 然后再利用得到的分解矩阵依次对图象样本进行变换; 最后将变换得到的系数用多种高效编码方法进行编码. 实验结果表明, 该新方法用于无损压缩编码时, 整型 DCT 明显好于浮点 DCT, 而且没有残差的编码问题; 用于有损压缩编码时, 两者的编码效果相近.

关键词 DCT(离散余弦变换) 整型变换 嵌入式编码 无失真

中图分类号: TN919.81 文献标识码: A 文章编号: 1006-8961(2002)03-0287-05

Image Coding Research Based on Integer DCT

WEI Chang-jiang, HAO Peng-wei, SHI Qing-yun

(National Laboratory on Machine Perception, Peking University, Beijing 100871)

Abstract In this paper the authors present a novel integer reversible DCT-based embedded image coding scheme, which can integrate lossy and lossless coding schemes perfectly. Image coding is of great interest in applications where efficiency with respect to data storage or transmission bandwidth is sought. The discrete cosine transform (DCT) has been applied extensively to the area of image coding. It has nice decorrelation and excellent energy-compaction properties and it can be easily implemented by hardware. In this paper the integer DCT transform is implemented by factoring the float DCT transform matrix into integer reversible transform series. Then the authors apply the series of matrices to image samples, and encode the coefficients by several effective coding algorithms. The simulation results illustrate that the integer DCT coding scheme is superior to the float DCT coding method in lossless coding, and the coding performance of integer DCT is close to that of float DCT in lossy coding.

Keywords DCT, Integer transform, Embedded coding, Lossless

0 引言

如今基于变换的编码方法已成为图象编码和视频编码的主流, 许多图象编码的国际标准都是基于变换编码的, 例如 JPEG, JPEG2000, MPEG-1, MPEG-2, H. 263, MPEG-4 等等. 图 1 给出了变换编码的通用框架.

在以上提到的编码标准中, 除 JPEG2000 采用的是小波变换外, 其他的都是采用 DCT 变换, 这是

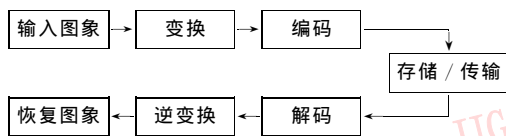


图 1 变换编码的通用框架

因为 DCT 变换是最接近 KL 变换的正交变换, 可有效去除图象空间域的相关性. DCT 变换可使图象的大部分能量集中到直流部分, 这对于图象编码非常重要. 另外, DCT 变换还有易于硬件并行实现的优点. 但是, 由于在这些方法中采用的都是浮点型 DCT 变换, 并不是整数可逆的, 因此, 利用有限计算精度的计算机来实现

DCT 变换域编码,只能是有损的编码.虽然这对于很多应用是可以允许的,但还有一些应用却不允许有任何的图象失真,比如医学图象和遥感图象.若一定要用浮点型 DCT 来实现无损压缩,就存在一个残差图象的编码问题,这必然导致无损编码的效率下降.此外,一些应用还要求有损和无损编码同时兼顾,即算法要统一.而对于这些应用,就需要实现可逆的整数变换.在文献[1]中,给出了一种使用 DCT 变换的无失真编码方法,其基本想法是在利用浮点 DCT 变换编码以后,先做一次解码,在得到重构图象后,再与原图象比较,以得到残差部分,再对残差部分利用预测的方法进行无失真编码,显然这样编码的效率较低.本文的方法,由于在变换部分就是无失真的整数可逆变换,因此只要保证编码支持无失真,就可达到图象的无失真压缩.

在变换编码中,大家知道,许多高效静止图象编码方法^[2~5]都是基于小波变换的,如 EZW^[2]、SPIHT^[3]、ZTE^[4]、EBCOT^[5]等编码方法.它们不仅都符合变换编码的通用框架,而且从这个框架中可以看到,变换和编码部分是分离的.且所有这些基于小波的算法,其高压缩性能很大一部分归功于它的编码部分,因为它们都采用了高效的位平面量化方法,并且都充分利用了变换系数的分布规律,但这些并不是小波变换所独有的性质,而其他的变换也可以使用这些编码方法.因为前面也提到了 DCT 变换有很多不

同于小波变换的优点,而且在视频压缩方面,现有的视频压缩标准采用的都是 DCT 变换,所以可以将笔者研究的整形 DCT 变换应用到视频压缩领域.本文就是将整型 DCT 变换与这些编码方法相结合进行编码的一次尝试,其编码结果表明,它不仅保留了 DCT 变换的优良特性,还获得了较好的编码效率.

1 整型 DCT 变换

文献[6]仿照小波提升的方法,通过将变换的蝶形运算整型化而得到了可逆的整型 DCT 变换方法,并与零树编码算法^[2]结合得到了不错的结果.文献[7]给出了可逆线性变换整数实现的一般方法,其结论是:①线性变换只要是有限维的和可逆的,则都存在整数实现方式;②线性变换矩阵的行列式值为整数因子时,整数实现的基本操作除位置交换外,不超过 7 步.文献[8]中又得到了更优化的结果,对于 $N \times N$ 的非奇异矩阵,除排列矩阵外,可以有不超过 3 个单位三角矩阵分解,或者是不超过 $N+1$ 个单行基本可逆矩阵(SERM)分解,本文就是利用这个分解方法来计算得到整型 DCT 变换矩阵.

1.1 矩阵分解

对于如下一个 8×8 DCT 变换矩阵 A :

$$A = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1918 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

使用文献[8]中的方法,即可以得到矩阵 A 的很多形如

$$A = P_L S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0 P_R$$

的分解结果,其中, $S_m = I + e_m s_m^T$, $m = 1, 2, \dots, 8$, $S_0 = I + e_8 s_0^T$, $e_m (m = 1, 2, \dots, 8)$ 为 8 阶单位矩阵的第 m 列向量, s_m 是第 m 个元素为 0 的 8 维列向量, $m = 1, 2, \dots, 8$, s_0 是第 8 个元素为 0 的 8 维列向量.

下面将说明由这种矩阵导出的变换是整数可逆的.

考虑变换 $y = S_m x$, 其中, x 是整数信号,由公式

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \\ \vdots \\ y_N \end{bmatrix} = S_m x = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{m,1} & \cdots & 1 & \cdots & s_{m,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ \vdots \\ x_N \end{bmatrix}$$

可以得到相应的整数变换结果为

$$y_k = \begin{cases} x_k & k \neq m \\ x_m + \left[\sum_{i=1}^{m-1} s_{m,i} x_i + \sum_{i=m+1}^N s_{m,i} x_i \right] & k = m \end{cases}$$

其中, $k = 1, 2, \dots, N$, $[\]$ 表示取整,可以是四舍五入,

也可以是截尾取整.

显然,其逆变换为

$$x_k = \begin{cases} y_k & k \neq m \\ y_m - \left[\sum_{i=1}^{m-1} s_{m,i}x_i + \sum_{i=m+1}^N s_{m,i}x_i \right] & k = m \end{cases}$$

因为上面说明了与变换 $y = S_m x$ 对应的整数可逆变换,所以可以将变换 $y = Ax$ 分步来实现整数可逆变换.这里再简要说明一下变换的具体过程,即先对输入信号 x 进行行置换,再依次用 $y = S_m x$, $m = 0, 1, 2, \dots, 8$ 进行整型变换,然后再对得到的输出信号 y 进行列置换,这样得到的 y 就是整型 DCT 变换的结果,它是对原浮点 DCT 变换结果的一种逼近,但它却是整数可逆的.

现在的问题是,要从中找到最优的分解,其优化准则是选取使得整型变换的结果最接近浮点变换结

$$\begin{aligned} s_0 &= [1.1648, 1.2355, 1.2013, 1.0141, -0.3670, 0.4415, -1.9616, 0]^T \\ s_1 &= [0, 1.0327, 0.3636, 0.0065, -0.3609, 0.3953, -0.7148, -0.4619]^T \\ s_2 &= [-0.3768, 0, 0.5320, 0.1989, -0.4496, 0.6077, -0.8764, -0.2716]^T \\ s_3 &= [0.4243, -0.8360, 0, 0.7210, -0.7014, 0.4360, -0.8467, -0.1633]^T \\ s_4 &= [0.5885, -0.1595, 0.0272, 0, 0.4142, 0.3289, -0.8984, -0.2265]^T \\ s_5 &= [0.0670, 0.5601, 0.7594, -0.5370, 0, 0.3244, -0.1299, -0.3204]^T \\ s_6 &= [-0.3417, 0.2160, 0.2702, -0.1910, -1.0824, 0, 0.2929, -0.3468]^T \\ s_7 &= [-0.0579, -0.3055, -0.3821, 0.2702, 0.5307, 0.1077, 0, 0.4904]^T \\ s_8 &= [0.2721, -1.4708, -0.9780, -0.0156, 1.8032, 1.1615, -1.3182, 0]^T \end{aligned}$$

行置换矩阵 P_L 和列置换矩阵 P_R 分别为:

$$P_L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$P_R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

果的变换,考虑整型变换 $[y] = [Ax]$,这里将 S_k 左乘取整的误差记为 d_k ,则有

$$\begin{aligned} [y] &= [Ax] = P_L (S_8 (\dots (S_1 (S_0 P_R x + d_0) + d_1) + \dots + d_7) + d_8) \\ &= Ax + P_L \left(\sum_{k=0}^7 \prod_{m=k+1}^8 S_m d_k + d_8 \right) \\ &= Ax + P_L \left(\sum_{k=0}^7 T_k d_k + d_8 \right) \end{aligned}$$

其中, $T_k = \prod_{m=k+1}^8 S_m, k = 0, 1, 2, \dots, N - 1$. 由 S_k 的特殊形式,通过分析就可得到除第 k 个元素外,误差为 $d_k (k = 1, 2, \dots, 7)$ 的其他元素都为 0 的结果,除第 8 个元素外,误差为 d_0 的其他元素都为 0,这样,要选取使得 $T_k (k = 1, 2, \dots, 7)$ 的第 k 列元素的绝对值尽可能小的分解,就必须使得 T_0 的第 8 列元素的绝对值尽可能小.经过优化筛选,选取了下面的分解:

1.2 变换系数重组

由于是采用类似于小波变换的 EBCOT 和 SPIHT 编码算法来进行整型 DCT 后编码,因此需要先对 DCT 的变换系数进行重组,以满足编码的条件.这里采用了如下两种重组方法:

第 1 种 将每个分块的 $8 \times 8 = 64$ 个系数看成是 64 个不同频率系数,即把不同块中相同频率的系数,按照块的排列顺序进行排列重组(图 2).

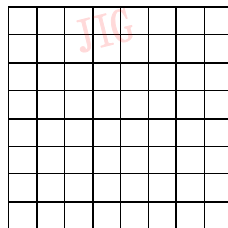


图 2 8×8 DCT 系数

设变换系数矩阵为 $D[M][N]$,重组后的矩阵为 $W[M][N]$,则有

$$W[i \% 8] \times M / 8 + i / 8 [(j \% 8) \times N / 8 + j / 8]$$

$$= D[i][j], i = 1, 2, \dots, M, j = 1, 2, \dots, N$$

这里的%是模运算符. 而且这种重组方式得到的系数矩阵, 对应于小波完全分解方式得到的系数矩阵. 所谓的小波完全分解方式, 就是在每一层分解的时候, 对上一层分解得到的 4 个子带都继续进行分解, 最后得到的所有子带都是一样大小.

第 2 种 首先把每个分块的 64 个系数划分成 10 个不同的频率子带(见图 3):

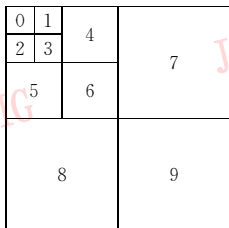


图 3 与 DCT 系数相对应的 3 层子带形式

然后把不同块中的相同频率的系数块, 按照块的排列顺序进行排列重组, 而这种重组方式则对应于小波的树形分解方式.

2 编码算法

2.1 支持完全分解的编码算法

文献 [5] 提出了 EBCOT (Embedded Block Coding with Optimized Truncation) 算法, 此算法已成为新一代静止图象压缩标准 JPEG2000 的基本框架. 其主要思想是, 首先对图象进行小波分解, 可以支持多种分解方式; 然后将变换得到的小波系数进

行分块, 通常是分成 64×64 大小, 最后对每一个系数块单独进行量化编码, 再将每个系数块编码成基本独立的码流. 它的优点之一就是依赖于某一种具体的分解方式, 本文结合 1.2 节中的第 1 种系数重组方法, 对整型 DCT 变换系数进行重组后, 再采用 EBCOT 算法进行编码.

2.2 支持树形分解的编码算法

Shapiro 首先提出了著名的 EZW 算法^[2], 此后的 SPIHT, ZTE 等算法^[3,4] 都是对 EZW 算法的改进和精细化. 它的主要思想是, 对图象进行树形小波分解, 并根据同一类子带的不同层系数的自相似性, 引入了零树的概念, 从而达到了高的压缩比和良好的重构图象质量. EZW 的另一个重要特点是, 它是系数的绝对大小为顺序和通过位平面量化来编码, 由于这样生成的码流具有嵌入式的特点, 从而为许多应用提供了一个很好的解决方法. 本文结合 1.2 节中的第 2 种系数重组方法, 对整型 DCT 变换系数进行重组后, 再采用 SPIHT 算法进行编码.

3 实验结果

实验图象是 Lena, Peppers 和 Mandrill 等几幅通用的压缩编码测试灰度图象, 其大小均为 $512 \times 512 = 262\,144$ pixels (1pixel=1Byte). 本文对这 3 幅图象进行了变换后的熵、无损和有损压缩等 3 项实验和测试.

首先进行了图象熵的测试, 其结果见表 1.

表 1 图象信息熵

图象名	原始熵	本文 DCT 变换后的熵	浮点 DCT 变换取整后的熵	浮点变换误差熵	取整熵与误差熵之和
Lena	7.45	4.74	4.47	2.58	7.05
Peppers	7.57	4.95	5.09	2.53	7.62
Mandrill	7.36	6.44	6.35	2.66	9.01

从表 1 可以看出, 本文 DCT 变换后的熵接近于浮点 DCT 变换后的熵, 但是在用浮点 DCT 变换来实现无损编码的时候, 还要编码其反变换后得到的误差信息, 表中给出的误差熵都大于 2, 实际上, 用

浮点 DCT 变换来实现无损编码所要处理的数据熵应为取整后的熵与误差熵之和, 显然它们都大于相应整型 DCT 变换的熵.

其次, 应用 3 种压缩算法作无损压缩, 其结果见表 2.

表 2 无失真压缩结果

图象名	文献[1]算法		整型 DCT+SPIHT 算法		整型 DCT+EBCOT 算法	
	压缩后字节数 (Byte)	bpp	压缩后字节数 (Byte)	bpp	压缩后字节数 (Byte)	bpp
Lena	164 484	5.02	142 497	4.35	147 053	4.49
Peppers	179 056	5.46	151 851	4.63	157 232	4.80
Mandrill	242 780	7.41	198 732	6.06	204 619	6.24

从表 2 可以看出,由于需要编码的信息熵具有一定的差别,因此通过浮点 DCT 变换实现的无损压缩编码结果与通过本文整型 DCT 变换实现的无损压缩编码结果相差很多.这充分表现出了整型 DCT 的优越性.

最后,采用两种不同的变换对 Lena 图象进行了有损压缩性能测试,其结果见表 3.

表 3 采用不同 DCT 变换的 JPEG 算法结果(PSNR)比较

压缩比	采用浮点 DCT 的 JPEG 算法	采用本文 DCT 的 JPEG 算法
2.8	43.62	43.16
12.5	35.78	35.72
32.5	30.40	30.25

从表 3 可以看出,对于有损编码,它们的编码结果非常接近,这是由于本文整型 DCT 变换后的熵非常接近浮点变换后熵的缘故.同样从对 Peppers 和 Mandrill 图象的测试结果也能得到与 Lena 图象压缩结果相同的结论.

参 考 文 献

- 1 Giridhar Mandyam, Nasir Ahmed, Neeraj Magotra. Lossless image compression using the discrete cosine transform [J]. Journal of Visual Communication and Image Representation, 1997,8(1):21~26.
- 2 Shapiro J M. Embedded image coding using zerotrees of wavelet coefficients[J]. IEEE Trans. Signal Processing, 1993,41(12): 3445~3463.
- 3 Said A, Pearlman M. A new, fast and efficient image codec based on set partitioning in hierarchical trees[J]. IEEE Trans. Circuits and Systems for Video Technology, 1996, 6 (3): 243~250.
- 4 Iraj Sodagar, Hung-Ju Lee, Paul Hatrack *et al.* Scalable wavelet coding for synthetic/natural hybrid images [J]. IEEE Trans. Circuits and Systems for Video Technology, 1999, 9 (2): 244~254.

- 5 David Taubman. High performance scalable image compression with EBCOT[J]. IEEE Trans. Image Processing, 2000,9(7): 1158~1170.
- 6 闫宇松,石青云. 可逆 DCT 整型变换与无失真图象压缩[J]. 软件学报, 2000,11(5):620~627.
- 7 郝鹏威,石青云. 可逆线性变换的整数实现[J]. 中国科学(E 辑), 2000,4(2):132~141.
- 8 Hao Pengwei, Shi Qingyun. Matrix factorizations for reversible integer mapping[J]. IEEE Trans. on Signal Processing, 2001, 49(10):2314~2324.

韦长江 1972 年生,1996 年获北京大学数学科学学院理学学士学位,现为北京大学数学科学学院博士生.主要从事计算机视觉、基于感兴趣区域的图象压缩等方面的研究.

郝鹏威 1966 年生,1997 年获中国科学院遥感应用研究所理学博士学位,现为北京大学信息科学中心副教授.研究方向包括信号变换、信号重构、图象压缩、信息隐藏.

石青云 1936 年生,中国科学院院士,北京大学数学科学学院教授、博士生导师,北京大学信息科学中心学术委员会主任,国际模式识别委员会理事.主要研究领域为模式识别、图象视频编码等.