

中图法分类号: TN919.81 文献标志码: A 文章编号: 1006-8961(2010)12-1742-06

论文索引信息: 刘华, 吴云, 赵勇, 田伟森. N 阶整数 DCT 变换基通用生成算法[J]. 中国图象图形学报, 2010, 15(12): 1742-1747

N 阶整数 DCT 变换基通用生成算法

刘 华, 吴 云, 赵 勇, 田伟森

(北京大学深圳研究生院集成微系统科学与工程重点实验室, 深圳 518055)

摘 要: 为了使 DCT 变换能够通用, 首先通过对 DCT 变换原理进行研究, 发现了变换基系数的取值个数与阶数的关系, 并结合余弦函数的性质对其进行了证明; 然后以此为基础, 提出了一种 $N(N=2^k, k>0, \text{下同})$ 阶整数 DCT 变换基的通用生成算法(该算法无需对相应的浮点基进行具体分析); 接着通过巧妙排列系数的序号, 使得生成的中间多项式具有极强的规律性; 最后设计了一个 N 位 M 进制数, 用来实现 N 重循环, 以穷举所有的可能解, 并成功对任意 N 元多项式组进行了求解。实验结果表明, 只要计算机的能力足够强大, 应用此算法便可以发现任意 $N \times N$ 整数 DCT 变换的所有可用基。

关键词: DCT 变换; 整数; N 阶变换基; 系数个数; 阶数

Generic generating algorithm for N -order integer DCT transform radix

LIU Hua, WU Yun, ZHAO Yong, TIAN Weisen

(The Key Laboratory of Integrated Microsystems, Shenzhen Graduate School, Peking University, Shenzhen 518055)

Abstract: The relationship between amounts of coefficients and order in radix is discovered through research of DCT principle and proven in combination with the nature of cosine function. On this basis, a generic generating algorithm for N -order ($N=2^k, k>0, \text{sic passim}$) integer DCT transform radix is presented, which we do not need to analyze floating radix corresponding to integer's. Through rearrange variations of coefficient, the mid-polynomials are extremely regularity. The group of polynomials in arbitrary N -variable is resolved by designing a N -digits with M as radix implementing N -loops to exhaust all possible solutions. The experimental results show that the algorithm can find all available radix for arbitrary $N \times N$ integer DCT as long as the computing capacity is enough.

Keywords: DCT transform; integer; N -order radix; amounts of coefficients; order

0 引 言

离散余弦变换(DCT)^[1]算法自 1974 年由 Ahmed, Natrajan 和 Rao 首次提出以来, 便成了图像和视频编解码领域最流行的算法而得到了广泛应用。由于浮点 DCT 的固有缺陷, 从 H.264 开始的一代视频编解码标准, 包括先进音视频编解码标准(AVS-P2)^[2]等普遍采用整数 DCT, 这就使得对整数

DCT 变换基的寻找成为了一个重要的研究课题。国际上一些大公司也都纷纷为自己提出的变换基申请了专利保护, 这些变换基虽有较好的变换特性, 但大多都是通过盲试发现的。

目前, 由于现有的 DCT 变换基生成算法不能用于求解任意阶的变换基, 如在文献[3]中, 需要通过分析具体的浮点 DCT 变换基的特点, 才能求解出整数变换基, 即不能通用, 因此有很大的局限性, 而本文由于发现了变换基系数的取值个数与矩阵阶数的

收稿日期: 2009-04-02; 改回日期: 2009-06-25

第一作者简介: 刘 华(1977—), 男, 北京大学深圳研究生院 2006 级硕士研究生。主要研究方向为视频编解码。E-mail: liuhuahuai@sina.com。

关系,并对其进行了证明,因此本文以此为基础提出了一种 N 阶变换基的通用生成算法,成功解决了任意 N 阶变换基的发现问题。由于本文算法并不需要分析浮点变换基,因此做到了真正的通用。

1 整数 DCT 的相关知识

1.1 整数 DCT 原理

浮点 DCT 的变换过程为(\mathbf{X} 为待变换矩阵, \mathbf{Y} 为结果矩阵, \mathbf{C}_F 为浮点变换基, 下角 F 表示浮点, 下同):

$$\text{浮点正向变换: } \mathbf{Y} = \mathbf{C}_F \mathbf{X} \mathbf{C}_F^T \quad (1)$$

$$\text{浮点逆向变换: } \mathbf{X} = \mathbf{C}_F^T \mathbf{Y} \mathbf{C}_F \quad (2)$$

求整数变换基就是将浮点变换基 \mathbf{C}_F 分解成整数矩阵 \mathbf{C}_I (下角 I 表示整型, 下同) 与浮点矩阵 \mathbf{G}_F , 使之满足 (\otimes 为矩阵点乘)

$$\mathbf{C}_F \approx \mathbf{C}_I \otimes \mathbf{G}_F \quad (3)$$

其中, \mathbf{G}_F 的第 m 行第 n 列元素值为

$$G_{m,n}^F = G_F(m,n) = \frac{1}{\sqrt{\sum_{n=0}^{N-1} (C_{m,n}^I)^2}} \quad (4)$$

$$(m, n = 0, 1, \dots, N-1)$$

将式(3)代入式(1)和式(2), 则整数 DCT 变换过程如下: 整数正向变换为

$$\mathbf{Y} = (\mathbf{C}_I \otimes \mathbf{G}_F) \mathbf{X} (\mathbf{C}_I \otimes \mathbf{G}_F^T) =$$

$$(\mathbf{C}_I \mathbf{X} \mathbf{C}_I^T) \otimes (\mathbf{G}_F \otimes \mathbf{G}_F^T) = \mathbf{C}_I \mathbf{X} \mathbf{C}_I^T \otimes \mathbf{E}_F \quad (5)$$

整数逆向变换为

$$\mathbf{X} = (\mathbf{C}_I \otimes \mathbf{G}_F)^T \mathbf{Y} (\mathbf{C}_I \otimes \mathbf{G}_F) =$$

$$\mathbf{C}_I^T (\mathbf{Y} \otimes (\mathbf{G}_F \otimes \mathbf{G}_F^T)) \mathbf{C}_I =$$

$$\mathbf{C}_I^T (\mathbf{Y} \otimes \mathbf{E}_F) \mathbf{C}_I \quad (6)$$

其中,

$$\mathbf{E}_F = \mathbf{G}_F \otimes \mathbf{G}_F^T \quad (7)$$

式(5)(6)中的点乘运算可分别在量化和逆量化中完成。

1.2 整数变换基需要满足的特性如下

1) 能归一化为正交矩阵^[4], 即 $\mathbf{C}_I \mathbf{C}_I^T$ 必须为对角阵;

2) 归一后的整数 DCT 变换基同浮点基相近^[4], 即要尽量满足式(3);

3) 整个变换过程的中间结果及最终结果都不能超出 -2^{15} 到 $2^{15} - 33$ 范围^[5] (可能会随阶数 N 的变化而调整), 又因目前的标准支持 8bits 深度的像

素值, 故整数变换基系数不能超出 4bits, 即

$$C_I(m, n) \in [0, 15]$$

2 变换基的系数取值个数与阶数的关系及其证明

通过大量实验及研究发现, 变换基的系数的取值个数与阶数是密切相关的, 这一点对于能否发现任意 N 阶变换基具有决定性作用。

根据余弦函数的性质及 DCT 核矩阵的特点, 对这一关系的证明如下:

由 DCT 核矩阵知, 浮点变换基第 m 行, 第 n 列的元素值为

$$C_{m,n}^F = C_F(m, n) =$$

$$\frac{\sqrt{2}}{N} \left(c(m) \cos \frac{(2n+1)m\pi}{2N} \right) \quad (8)$$

其中, $N = 2^k$, $k > 0$; $m, n = 0, 1, \dots, N-1$;

$$c(m) = \begin{cases} \frac{1}{\sqrt{2}} & m = 0 \\ 1 & m = 1, 2, \dots, N-1 \end{cases}$$

由于 $N = 2^k = 2 \times 2^{k-1}$, 所以 $m \neq 0$ 时, $\frac{(2n+1)m}{N} \neq l$

(注: l 为整数); 即 $\frac{(2n+1)m\pi}{2N} \neq \frac{l\pi}{2}$, 因此

$$\left| \cos \frac{(2n+1)m\pi}{2N} \right| \in \left(\cos \frac{(2l+1)\pi}{2}, \left| \cos \frac{2l\pi}{2} \right| \right) \cup$$

$$\left(\cos \frac{(2l+1)\pi}{2}, \left| \cos \frac{2(l+1)\pi}{2} \right| \right) \cong$$

$$\left(\cos \frac{(2l+1)N\pi}{2N}, \left| \cos \frac{2lN\pi}{2N} \right| \right) \cup$$

$$\left(\cos \frac{(2l+1)N\pi}{2N}, \left| \cos \frac{2(l+1)N\pi}{2N} \right| \right)$$

其中, $l = \frac{(2n+1)m}{2N}$ (x 表示不大于 x 的最大整数)。

由于 $|\cos x|$ 以 π 为周期, 且在 $x > 0$ 时, 第 l 个周期以 $x = \frac{2l+1}{2}\pi$ 为轴对称, 所以

$$\left| \cos \frac{(2n+1)m\pi}{2N} \right| \in \left(\cos \frac{(2l+1)N\pi}{2N}, \left| \cos \frac{2lN\pi}{2N} \right| \right)$$

其中 $l = 0, 1, 2, \dots$ 。

若取 $l = 0$, 则

$$\left| \cos \frac{(2n+1)m\pi}{2N} \right| \in \left(\cos \frac{N\pi}{2N}, \cos \frac{0\pi}{2N} \right) \cong$$

$$\left[\cos \frac{(N-1)\pi}{2N}, \cos \frac{\pi}{2N} \right]$$

又因

$$C_{0,n}^F = C_F(0, n) = \sqrt{\frac{2}{N}} \left(c(0) \cos \frac{(2n+1)0\pi}{2N} \right) = \sqrt{\frac{2}{N}} \sqrt{\frac{1}{2}} = \sqrt{\frac{2}{N}} \cos \frac{\pi}{4} = \sqrt{\frac{2}{N}} \cos \frac{N}{2} \frac{\pi}{2N}$$

当 $r = N/2$ 时, 则

$$\sqrt{\frac{2}{N}} \cos \frac{r\pi}{2N} = \sqrt{\frac{2}{N}} (c(0) \cos 0) = C_F(0, n) = C_{0,n}^F$$

$$\text{故 } |C_F(m, n)| \in \left\{ y_j(r) = \sqrt{\frac{2}{N}} \cos \frac{r}{2N} \pi \right\} \quad (9)$$

式中, $m, n = 0, 1, \dots, N-1$; $r, j = 1, 2, \dots, N-1$

即 $C_F(m, n)$ 的取值个数最多为 $N-1$ 个。

又因 $\cos x \left(0 < x < \frac{\pi}{2} \right)$ 单调减, 且当 $n=0$ 时, r 从 1 到 $N-1$ 均有取值, 所以 $C_F(m, n)$ 的取值个数最少为 $N-1$ 个。

综上所述, N 阶变换基系数的绝对值的取值个数有且仅有 $N-1$ 个。

特别地, 当 $m=0$ 时, $r = N/2$ 。

3 整数变换基生成算法

在文献[3]中需要结合具体的浮点变换基(如 8×8 变换基)通过对浮点变换系数进行具体分析来求解整数变换基。而本文由于有了系数的取值个数与矩阵阶数的关系, 从而无需知道具体的浮点变换基, 就可以求解出整数变换基。本文结合整数变换基需要满足的特性, 给出如下算法。

3.1 计算变换基系数变量矩阵 C_1

1) 计算变换基中的所有可能系数

由式(9)计算出 $\{y_j(r)\}$, 其中 $r=1, 2, \dots, N-1$, 令与其对应的整数变换基系数变量为 $\{x_j\}$; 此时, r 与 j 一致。

2) 查找 $C_{m,n}^F$ 在 $\{x_j\}$ 中的序号 j

由式(8)知, 因 $|\cos x|$ 以 π 为周期, 故令 $t = \frac{(2n+1)m\pi}{2N} \% \pi = (2n+1)m \% (2N)$, 使 $|C_{m,n}^F|$ 映射到 $\sqrt{\frac{2}{N}} |\cos x|$, $x \in (0, \pi)$ 区间, 则 $t/(2N) \in (0, \pi)$; 因 $|\cos x|$, $x \in (0, \pi)$ 以 $\pi/2$ 为轴对称, 故若 $t > N$, 则再取 $t = 2N - t$, 此时, $|C_{m,n}^F|$ 对应于 $\sqrt{\frac{2}{N}} \cos x$, $x \in (0, \pi/2)$;

在 $\{y_j(r)\}$ 中查找 $r=t$ 时的序号 j , 以 r 首次出现为序调整 $\{y_j(r)\}$ 中的序号 j , 同时相应调整 $\{x_j\}$ 中的序号 j 。这一顺序调整可以极大降低后续求解过程的复杂度;

3) 计算 $C_{m,n}^F$ 的符号 s

再由式(8)知, 因 $\cos x$ 以 2π 为周期, 故令 $q = \frac{(2n+1)m\pi}{2N} \% (2\pi) = (2n+1)m \% (4N)$, 使 $C_{m,n}^F$ 映

射到 $\sqrt{\frac{2}{N}} \cos x$, 其中 $x \in (0, 2\pi)$, 以确定其正负, 若 $N < q < 3N$, 即 $q/(4N) \in (\pi/2, 3\pi/2)$, 则 $\cos x < 0$, $x \in (\pi/2, 3\pi/2)$, 即 $C_1(m, n) < 0$, 符号 $s = 0 - 1$; 否则若 $q < N$ 或 $q > 3N$, 即 $q/(4N) \in (0, \pi/2) \cup (3\pi/2, 2\pi)$, 则 $\cos x > 0$, $x \in (0, \pi/2) \cup (3\pi/2, 2\pi)$, 即 $C_1(m, n) > 0$, 符号 $s = +1$;

4) 确定 C_1 第 m 行、第 n 列的带符号系数变量

$$C_{m,n}^1 = C_1(m, n) = s \times x_j$$

按 m, n 从 0 到 $N-1$ 的顺序重复步骤 2)~4)

逐个计算 $C_1(m, n)$, 特别地, $m=0$ 时, $t = N/2$, $s = +1$ 。

3.2 计算多项式矩阵 $P = C_1 C_1^T$

根据性质 1) 计算 $C_1 C_1^T$, 其结果矩阵 P 中各元素均为二次多项式。由于在生成系数矩阵的过程中, 对 $\{y_j(r)\}$ 及 $\{x_j\}$ 中的顺序进行过调整, 故矩阵 P 中各元素(多项式)具有如下性质:

- 1) 所有多项式中的系数 x_j 的序号 j 连续;
- 2) 将所有多项式按是否包含某个系数进行分组, 共有 $\lfloor N \rfloor$ 组, 则同一组中的每个多项式都包含本组中的所有系数, 且每个系数出现的次数均相同;
- 3) 任意两组中所包含的系数互不重叠;
- 4) 若矩阵 P 中各元素(多项式)按行、列顺序排列, 则每组中均是对角线多项式(即全平方项)首先出现, 非对角线多项式(即无平方项)紧随其后(若存在), 由性质 1) 知, 其值为 0;
- 5) 若将恒值为 0 的元素省略, 且各二次项均成比例的只保留一个, 则每组多项式的个数及每个多项式所包含的系数的个数与二次项数(相同的重复计算)的关系如表 1 所示。

6) 结果为如下形式:

$$p_1 : Nx_1^2$$

$$p_2 : 2 \sum_{j=2}^{\frac{N}{2}+1} x_j^2 \quad (\text{当 } N \geq 16 \text{ 时})$$

$p_3: x_2 x_3 + \dots - x_{\frac{N}{2}} x_{\frac{N}{2}+1}$ (共 $\frac{N}{2}$ 项)
 \dots , (为 $x_2, \dots, x_{\frac{N}{2}+1}$ 的二次多项式, 包括 p_3 , 共 $\frac{N}{4} - 1$ 个)

$$p_{\frac{N}{4}+2}: 4 \sum_{j=\frac{N}{2}+2}^{\frac{3N}{4}+1} x_j^2 \quad (\text{当 } N \geq 32 \text{ 时})$$

\dots

$$p_{\frac{N}{2}-2}: \frac{N}{4} \sum_{j=N-6}^{N-3} x_j^2 \quad (\text{当 } N \geq 8 \text{ 时})$$

$$p_{\frac{N}{2}-1}: x_{N-6} x_{N-5} - x_{N-6} x_{N-4} - x_{N-5} x_{N-3} - x_{N-4} x_{N-3} \quad (\text{当 } N \geq 8 \text{ 时})$$

$$p_{\frac{N}{2}}: \frac{N}{2} (x_{N-2}^2 + x_{N-1}^2) \quad (\text{当 } N \geq 4 \text{ 时})$$

7) 解的特点, $N \geq 4$ 时, 各阶变换基的第 1 组可采用 2 阶变换基的解; $N \geq 8$ 时, 各阶的变换基第 3 组及以后各组可采用 $N/2$ 阶变换基的解。

表 1 每组多项式的个数、每个多项式所包含的系数个数与二次项数的关系

Tab. 1 The relationship between quadratic numbers and each group the number of polynomial, coefficient numbers in each polynomial

组序号	1	2	3	4	\dots	$\text{lb}N - 1$	$\text{lb}N$
每组多项式个数	1	$\frac{N}{4}$	$\frac{N}{8}$	$\frac{N}{16}$	\dots	2	1
本组多项式的系数个数与二次项个数(相同者不重复计算)	1	$\frac{N}{2}$	$\frac{N}{4}$	$\frac{N}{8}$	\dots	4	2
存在条件($N \geq ?$)	2	16	32	64	\dots	8	4

3.3 解矩阵 P 中各多项式组

根据 3.2 节中列出的多项式特点, 可采用穷举法求解 P 中各组多项式, 详细流程如图 1 所示。为此, 本文设计了一个 N 位 M 进制数 k , 为便于理解, 图 1 中称之为计数器: 因所有系数的最大可能取值为 15, 故其进制 M 为 16; $N = j_{\max} - j_{\min} + 2$ (j_{\max}, j_{\min} 为当前组多项式系数的最大、最小序号); x_j 对应 k 的第 $j - j_{\min}$ 位; 第 $N - 1$ 位用作哨兵位, 初值为 0, 若变为 1, 即最高位发生进位, 则表示所有的可能取值都已试探完毕。若当前组多项式中没有非对角线多项式, 则 k 的所有可能取值都是解, 不必试探。否则, 将 k 从 $0 \underbrace{1 \dots 1}_{N-1 \text{ 个 } 1}$ 到 $0 \underbrace{15, \dots, 15}_{N-1 \text{ 个 } 15}$, x_j ($j \in [j_{\min}, j_{\max}]$) 各取其中一位) 逐一代入本组所有非对角线多项式

求值。若全都为 0, 则 x_j ($j \in [j_{\min}, j_{\max}]$) 的当前值是解。

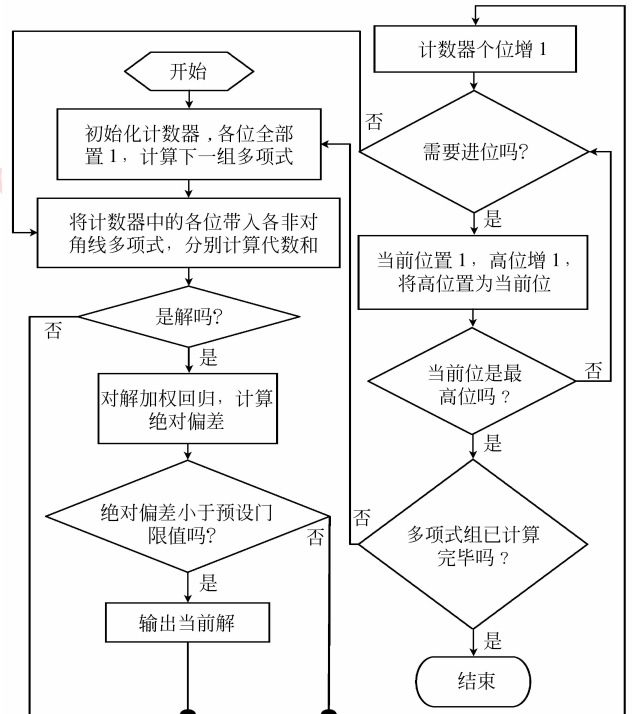


图 1 多项式求解流程图

Fig. 1 The flow chart for solving polynomial

为检验对性质 2) 的满足程度, 应对当前解进行回归。因系数越大对变换后的能量贡献越大, 故

$$T_F = \sum_{j=j_{\min}}^{j_{\max}} \left| \frac{x_j}{\sqrt{c \sum_{l=j_{\min}}^{j_{\max}} x_l^2}} - y_j \right| \times \frac{y_j}{\sum_{i=j_{\min}}^{j_{\max}} y_i} \text{ 应加权回归}^{[3]},$$

并使门限值 T_F 尽量小, T_F 应根据阶数 N 的大小调整。

其中, y_j 为与 x_j 相应的浮点变换基系数, c 为本组对角线多项式二次项的系数。需要指出的是, 由于文献[3]中做加权时, 用的是每个系数在整个变换基所有系数中的权重, 而本文则采用系数在当前组多项式所有系数中的权重, 即本文在计算偏差时的分母部分比文献[3]中的分母小, 因此表现为计算出的偏差比文献[3]中的偏差要大。但经分组后, 各组多项式在变换系数上各自独立, 组间没有交叉重叠, 这样更能真实反映对性质 2) 的满足程度。

将变换基系数矩阵中的变量系数用对应解替换即可得整数变换基 C_I , 最后即可根据式(4)(7)计算出 E_F 。

4 算法复杂度分析

各阶的中间多项式个数及二次项的个数中最多的均为第 2 组,是存储及计算的关键所在。由表 1 可知, N 阶变换基的中间多项式中第 2 组共有 $N/4$ 个多项式,每个多项式中共有 $N/2$ 个二次项及系数。

1) 存储空间上的复杂度主要体现在保存中间多项式上。其中,多项式中的每个二次项包括系数、两个变量的下标及指向下一项的指针,共需 16 Bytes 存储空间,因此保存第 2 组多项式共需 $16 \times \frac{N}{2} \times \frac{N}{4} = 2N^2$ Bytes 存储空间。另外,还需要一个变量矩阵及一个残差单位阵的存储空间,即 $2 \times 4 \times N^2 = 8N^2$ 。故其存储复杂度为 $10N^2$ 。

2) 由于每个变量的可能取值有 15 个,需要对每个变量进行 15 次试探,故采用穷举法的计算复杂度为 $(\frac{N}{4} - 1) \times 15^{\frac{N}{2}}$ 。

5 实验结果

表 2—表 4 中的变换基为第 2 组多项式在门限值 $T_F = 0$ 时的解,按照同浮点 DCT 系数的加权偏差从小到大的顺序排列。表中, x_j 为整数变换基系数; z_j 为加权回归后的变换基系数;偏差(全)为按文献 [3] 的方法计算得到的加权偏差;偏差门限值 T_F 为

按本文方法计算得到的加权偏差。与 4×4 整数变换基(第 2 组)相对应的浮点变换基为 0.652 28, 0.270 6,其中 3 号基是 JM 参考软件^[6]中用到的。与 8×8 整数变换基(第 2 组)相对应的浮点变换基为 0.490 39, 0.415 73, 0.277 79, 0.097 545,其中 1 号基是 AVS1-P2 参考软件^[7]中用到的,3 号基是 JM 参考软件^[6]中用到的。与 16×16 整数变换基(第 2 组)相对应的浮点变换基为 0.351 85, 0.338 33, 0.311 81, 0.273 3, 0.224 29, 0.166 66, 0.102 63, 0.034 654。限于篇幅,本文只列出 4、8、16 阶第 2 组多项式的解(2 阶只有一个变换系数,省略)。上述结果的测试环境为 intel Pentium 4 主频 2.8GHz,内存 1GBytes。结果表明, 8×8 DCT 变换的所有可用整数基在 5s 内即可以全部求解出;对于 16×16 DCT 变换的所有的可用整数基也可以在 7min 内全部求解出。实际上只要计算能力足够强大,应用此算法便可以发现任意 $N \times N$ 整数 DCT 的所有可用变换基。

需要强调的一点是,虽然加权偏差越小与浮点变换基的近似程度就越好,但是否最优,还要结合变换时需要的加法及移位次数进行考虑。

表 2 4×4 整数变换基
Tab. 2 4×4 integer transform radix

序号	x_2	x_3	z_2	z_3	偏差阈值 T_F	偏差(全)
1	12	5	0.652 71	0.271 96	0.000 801 4	0.000 519 99
2	5	2	0.656 53	0.262 61	0.004 637 4	0.003 009
3	2	1	0.632 46	0.316 23	0.028 091	0.018 227

表 3 8×8 整数变换基
Tab. 3 8×8 integer transform radix

序号	x_2	x_3	x_4	x_5	z_2	z_3	z_4	z_5	偏差阈值 T_F	偏差(全)
1	10	9	6	2	0.475 7	0.428 1	0.285 4	0.095 1	0.011 48	0.006 429
2	15	12	8	3	0.504 5	0.403 6	0.269 1	0.100 9	0.011 48	0.006 429
3	12	10	6	3	0.499 1	0.416 0	0.249 6	0.124 8	0.011 60	0.006 498

表 4 16×16 整数变换基
Tab. 4 16×16 integer transform radix

序号	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	z_2	z_3	z_4
1	15	13	11	11	5	6	5	1	0.394 5	0.341 9	0.289 3
2	10	12	6	9	7	8	2	2	0.322 1	0.386 5	0.193 3
3	11	14	11	11	15	9	1	1	0.264 2	0.336 2	0.264 2

序号	z_5	z_6	z_7	z_8	z_9	偏差阈值 T_F	偏差(全)
1	0.289 3	0.131 5	0.157 8	0.131 49	0.026 30	0.029 458	0.015 527
2	0.289 9	0.225 5	0.257 7	0.064 42	0.064 42	0.049 152	0.025 908
3	0.264 2	0.360 2	0.216 1	0.024 02	0.024 02	0.053 282	0.028 085

6 结 论

本文首先分析了整数 DCT 变换原理,然后根据 DCT 核矩阵及余弦函数的特点,对变换基系数的取值个数与矩阵阶数的关系进行了证明,并结合正交矩阵的特性,提出了一种 N 阶变换基的通用生成算法。在文献[3]中需要结合具体的变换矩阵,如 8 阶变换矩阵进行具体分析求解,并且只给出了 4 阶及 8 阶变换矩阵的求解结果。而本文算法由于有了系数的取值个数与矩阵阶数的关系,因此无需知道浮点变换基就可以解出任意 N 阶整数变换基,浮点系数只是在做加权回归时才需要使用。

参考文献 (References)

- [1] Watson Andrew B. Image compression using the discrete cosine transform [J]. *Mathematica Journal*, 1994, 4(1):81-88.
- [2] GB/T 20090.2—2006 Information Technology-Advanced Coding of Audio and Video - Part 2:Video[S]. [GB/T 20090.2—2006 信息技术先进音视频编码第 2 部分:视频[S]]
- [3] Wang Zhongyuan, Hu Ruimin, Tian Gang, et al. The generic generating algorithm for integer DCT transform radix [J]. *Journal of Image and Graphics*, 2008, 13(6):1061-1064. [王中元,胡瑞敏,田纲,等. 整数 DCT 变换中变换基的通用生成算法 [J]. *中国图象图形学报*, 2008, 13(6):1061-1064.]
- [4] Malvar H S, Hallapuro A, Karczewicz M, et al. Low-complexity transform and quantization in H.264/AVC [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003, 13(7): 598-603.
- [5] Wiegand T, Sullivan G, Luthra A. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H. 264 | ISO/IEC 14496-10 AVC) [EB/OL]. [2009-02-25]. <http://www.itu.int/rec/T-REC-H.264-201003-I>.
- [6] JVT. JM Reference Software Official Version JM15.0[CP/OL]. [2009-03-19]. <http://iphone.hhi.de/suehring/tml>.
- [7] AVS. AVS Reference Software Official Version rm52j[CP/OL]. [2009-03-19]. <http://www.avs.org.cn/fruits>.