ISSN1000-9825 Journal of Software 软件学报 2000,11(5):620~627

# 可逆的DCT整型变换与无失真图像压缩*

闫宇松 石青云

(北京大学信息科学中心视觉与听觉信息处理国家重点实验室 北京 100871)

E-mail: yys@sxx0.math.pku.edu.cn

**摘要** 使用提升的方法,利用FFT(fast Fourier transform)的蝶型构造,完成了FFT与DCT(discrete cosine transform)的从整数到整数的变换.变换本身是可逆的,因此非常适合于无失真图像压缩.

**关键词** FFT,DCT,提升,整型变换,无失真图像压缩.

**中图法分类号** TP391

在文献[1]中,Daubechies采用提升的方法对小波变换进行了改造,使之成为从整数到整数的整型变换.这种整型变换克服了计算机浮点运算的精度误差问题,真正实现了变换的可逆性,为无失真图像压缩开辟了道路.但是,在传统的图像压缩方法以及信号处理中,FFT(fast Fourier transform)与DCT(discrete cosine transform)变换被广泛地使用.有人也曾经分析过,DCT变换在高信噪比的情况下比小波有更好的压缩性能.这一动机促使我们对DCT的整型变换进行研究.

本文第1节简述提升的基本原理和方法.第2节对FFT的蝶型算法进行矩阵分析,同时,运用提升的方法构造FFT的整型变换.第3节通过分析FFT与DCT的关系,进一步构造保持变换系数其鞍对称性的FFT整型变换.最终获得DCT的整数变换方法.整个算法流程是我们在文献[1]的思想基础上完成的.

## 1 提 升

Daubechies在文献[2]中叙述了提升的方法.该方法可以为任何主对角线元素为1的三角型矩阵构造出与其相对应的可逆整型变换.同样地,如果任何一个线性变换矩阵能够写成这种三角形矩阵的乘积的话,那么该变换也可以通过提升的方法构造出相应的可逆整型变换.

我们首先以上三角型矩阵为例来简述提升的思想.对于如下变换:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{1.1}$$

其中$x_1,x_2$为输入,$y_1,y_2$为输出,$\alpha$为浮点数.我们可以构造出它所对应的整型变换为

$$\begin{aligned} y_1 &= x_1 + \lfloor \alpha x_2 \rfloor, \\ y_2 &= x_2, \end{aligned} \tag{1.2}$$

其中$\lfloor x \rfloor$表示对$x$取整.我们从式(1.2)可以看出,如果$x_1,x_2$为整数,那么经过计算得到的$y_1,y_2$也必定为整数.因此,式(1.2)确实是一个从整数到整数的变换.同时,我们可以得到变换(1.2)的逆变换形式:

$$\begin{aligned} x_2 &= y_2, \\ x_1 &= y_1 - \lfloor \alpha x_2 \rfloor, \end{aligned} \tag{1.3}$$

式(1.3)所表示的是,如果我们获得了两个整数$y_1,y_2$,可以首先通过$y_2$重建$x_2$,再通过$y_1$与$x_2$,利用第2个式

* 本文研究得到国家自然科学基金(No.69735020)和国家863高科技项目基金(No.863-2-4)资助.作者闫宇松,1969年生,博士,主要研究领域为模式识别,图像处理,计算机视觉.石青云,女,1936年生,教授,博士生导师,中国科学院院士,主要研究领域为模式识别,图像处理,计算机视觉.

本文通讯联系人:闫宇松,北京 100871,北京大学视觉与听觉信息处理国家重点实验室

本文1998-12-29收到原稿,1999-03-11收到修改稿

予重建 $x_1$.容易看出,式(1.3)也是整型的变换,并且这种变换相对于式(1.2)来说是真正可逆的.类似地,下三角型矩阵也能够找到它所对应的可逆整型变换.

如果变换矩阵可以分解成多个主对角线元素为1的三角形矩阵的乘积,那么,只要对每一个三角形矩阵分别找到它所对应的可逆整型变换,然后按分解顺序依次变换,就可构造出整个变换的整型变换.文献[2]就是利用这种思想寻找小波变换所对应的可逆整型变换的.他们将这一类方法统称为提升,并将这种主对角线元素为1的三角形矩阵称为提升矩阵.

可逆整型变换保证变换对于整数信号是真正可逆的,同时,变换结果与浮点运算相差无几,能够有效地降低信号的冗余度,从而为数据的无损压缩做好准备.本文将利用提升的方法构造FFT与DCT所对应的整型变换,但在具体构造前,需首先引入两个基本矩阵的提升公式.

(1) 拉伸(scaling)矩阵的提升形式为

$$\begin{bmatrix} k & 0 \\ 0 & 1/k \end{bmatrix} = \begin{bmatrix} 1 & k-k^2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1/k & 1 \end{bmatrix} \begin{bmatrix} 1 & k-1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{1.4}$$

(2) 旋转(rotation)矩阵的提升形式为

$$\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} = \begin{bmatrix} 1 & -\mathrm{tg}(\alpha/2) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin\alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & -\mathrm{tg}(\alpha/2) \\ 0 & 1 \end{bmatrix}. \tag{1.5}$$

由式(1.4)、(1.5)我们可以看出,拉伸矩阵与旋转矩阵都可以写成提升的形式.因此,拉伸变换与旋转变换也都有其对应的可逆整型变换.

(1) 拉伸矩阵的整型变换如下所示:

$$\begin{aligned} &y_1^{(0)} = x_1, \\ &y_2^{(0)} = x_1 + x_2, \\ &y_1^{(1)} = y_1^{(0)} + \lfloor (k-1) y_2^{(0)} \rfloor, \\ &y_2^{(1)} = y_2^{(0)}, \\ &y_1^{(2)} = y_1^{(1)}, \\ &y_2^{(2)} = y_2^{(1)} + \lfloor (-1/k) y_1^{(1)} \rfloor, \\ &y_1 = y_1^{(2)} + \lfloor (k-k^2) y_2^{(2)} \rfloor, \\ &y_2 = y_2^{(2)}. \end{aligned}$$

(2) 旋转矩阵的整型变换如下所示:

$$\begin{aligned} &y_1^{(0)} = x_1 + \lfloor -\mathrm{tg}(\alpha/2) x_2 \rfloor, \\ &y_2^{(0)} = x_2, \\ &y_1^{(1)} = y_1^{(0)}, \\ &y_2^{(1)} = y_2^{(0)} + \lfloor \sin\alpha y_0^{(0)} \rfloor, \\ &y_1 = y_1^{(1)} + \lfloor -\mathrm{tg}(\alpha/2) y_2^{(1)} \rfloor, \\ &y_2 = y_2^{(1)}. \end{aligned}$$

以上是拉伸矩阵与旋转矩阵所对应的整型变换,其中,有上标的变量是中间结果,$x_1$,$x_2$为整数输入,$y_1$,$y_2$为整数输出.相应的反变换类似于式(1.3),也很容易获得.并且这两个矩阵的可逆整型变换都将在下文被引用.

## 2 FFT的整型变换

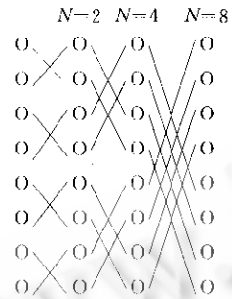
为了后面叙述的方便,我们首先引入复整数的概念.任何一个复数,如果它的实部和虚部都是整数,我们就称之为复整数.由于FFT变换是作用在复数域上的变换,所以本节所研究的FFT整型变换指的是从复整数到复整数的可逆变换.

离散傅里叶变换(DFT)在信号处理中具有十分广泛的应用,它的数学表达式如下:

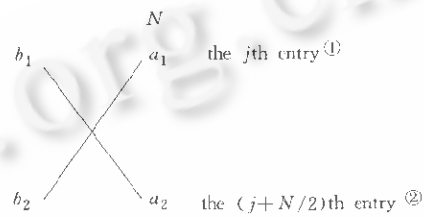
$$\hat{F}[u]=\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}F[n]\exp\left[\frac{-i2\pi nu}{N}\right], \tag{2.1}$$

$$F[n]=\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}\hat{F}[u]\exp\left[\frac{-i2\pi un}{N}\right]. \tag{2.2}$$

同时,人们为了克服傅里叶变换本身巨大的运算量,研究出一种快速算法,称为快速傅里叶变换(FFT).FFT 能处理离散信号长度为 2 的整数幂.它的具体方法就是将原始信号按逆序重排,对重排后的数据再进行几层蝶型变换,在最后一层获得变换结果,理论推导可参看文献[3,4].图 1(a)是 8 个数据的 FFT 示意图.




(a) A length-8 FFT. The symbol O indicates a complex value
(a) 8个数据的FFT变换,其中O代表复数




① 第 $j$ 个数据,② 第 $j+N/2$ 个数据

(b) Single butterfly, $a_1$, $a_2$ are output, $b_1$, $b_2$ are input
(b) 单个蝶形变换示意图,$a_1$,$a_2$为输入,$b_1$,$b_2$为输入

Fig. 1
图1

图 1(b)代表了单个蝶形的变换,具体的运算形式如下:

$$\begin{bmatrix}a_1\\a_2\end{bmatrix}=\begin{bmatrix}1 & W_N^j\\1 & W_N^{j+N/2}\end{bmatrix}\begin{bmatrix}b_1\\b_2\end{bmatrix}=\begin{bmatrix}1 & W_N^j\\1 & -W_N^j\end{bmatrix}\begin{bmatrix}b_1\\b_2\end{bmatrix},\quad \text{其中 } W_N^j=\exp\left[\frac{-i2\pi j}{N}\right]. \tag{2.3}$$

式(2.3)是一般的 FFT 蝶形变换公式.但是,由于我们的整型变换要与数据压缩联系起来,而压缩需要变换后数据的动态范围尽可能地小,所以在这里我们将式(2.1)和式(2.2)中的前缀 $1/\sqrt{N}$ 融于每一个蝶形变换中,也就是在每一层变换中多乘以一个 $1/\sqrt{2}$,这样,蝶形变换的结果与式(2.1)和式(2.2)的计算结果实际上是等价的,相应的运算形式如下:

$$\begin{bmatrix}a_1\\a_2\end{bmatrix}=\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{W_N^j}{\sqrt{2}}\\\frac{1}{\sqrt{2}} & \frac{W_N^{j+N/2}}{\sqrt{2}}\end{bmatrix}\begin{bmatrix}b_1\\b_2\end{bmatrix},\quad \text{其中 } W_N^j=\exp\left[\frac{-i2\pi j}{N}\right]. \tag{2.4}$$

下面,我们将以式(2.4)为基础来推导可逆的 FFT 整型变换.

由以上讨论可知,FFT 变换可以表示成多个蝶形变换的组合,如果我们能够保证每一个蝶形变换都有对应的可逆整型变换,那么就可以获得 FFT 的整型变换.如何保证蝶形变换有对应的整型变换呢?由第 2 节可知,只要蝶形变换矩阵能够表示成提升矩阵的乘积,就可以构造出它的可逆整型变换.显然,对于变换(2.4),我们可以得到以下公式:

$$\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{W_N^j}{\sqrt{2}}\\\frac{1}{\sqrt{2}} & -\frac{W_N^j}{\sqrt{2}}\end{bmatrix}=\begin{bmatrix}1 & 0\\1 & 1\end{bmatrix}\begin{bmatrix}1 & -1/2\\0 & 1\end{bmatrix}\begin{bmatrix}\frac{\sqrt{2}}{2} & 0\\0 & \sqrt{2}\end{bmatrix}\begin{bmatrix}1 & 0\\0 & W_N^j\end{bmatrix}. \tag{2.5}$$

在式(2.5)中,我们可以看出,矩阵 $\begin{bmatrix}1 & 0\\1 & 1\end{bmatrix}$ 与 $\begin{bmatrix}1 & -\frac{1}{2}\\0 & 1\end{bmatrix}$ 满足提升的要求.而 $\begin{bmatrix}\frac{\sqrt{2}}{2} & 0\\0 & \sqrt{2}\end{bmatrix}$ 是一个拉伸矩阵,

可以通过式(1.4)表示成提升的形式.因此,下面我们只需要证明矩阵$\begin{bmatrix}1 & 0\\ 0 & -W_N^k\end{bmatrix}$可以找到它所对应的整型变换就可以了.

首先,我们知道,FFT变换都是针对复数操作的.这使我们有可能将$W_N^k$本身写成提升的形式.对于复数操作,$b=W_N^k\times a$,其中$b=b_x+ib_y$,$a=a_x+ia_y$,我们可以写成实数的变换形式如下:

$$\begin{bmatrix}b_x\\ b_y\end{bmatrix}=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}a_x\\ a_y\end{bmatrix},\quad \text{其中 } \theta=\frac{-i2\pi j}{N}, \tag{2.6}$$

再利用旋转矩阵的提升式(1.5),就得到了相应的从复整数$a$到复整数$b$的可逆整型变换.进一步地,$b=-W_N^k\times a$也有可逆整型变换,从而说明了$\begin{bmatrix}1 & 0\\ 0 & -W_N^k\end{bmatrix}$可以找到所对应的整型变换.这使我们能够构造出蝶形变换(2.4)的可逆整型变换.相应地也就得到了FFT的可逆整型变换.

我们按照以上算法在计算机上实现了FFT的可逆整型变换.其优点是:(1)真正做到了从整数到整数的可逆变换,从而克服了浮点变换的误差问题;(2)与浮点变换的计算结果相差很小,能够达到与FFT浮点变换相同的效果;(3)动态范围保持不变.

因此,FFT整型变换对于复数数据的处理是具有优势的.但是,它也有一个缺点.众所周知,对于实数信号而言,经过FFT变换后,变换结果应该具有对称共轭性,这一特性使得我们只要记录一半的复数,就可以把原始实信号恢复出来.但是,整型变换由于其本身的固有特点,使得这一特性丧失了.而这一特性对于数据压缩是至关重要的,并且也直接关系到DCT整型变换的实现,这促使我们继续研究对实信号具有共轭对称性的FFT整型变换,并在下一节给出结果.

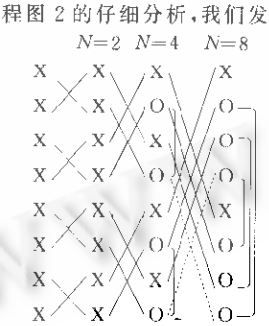
## 3 DCT的整型变换

这一节,我们将讨论DCT的整型变换问题.但首先我们将研究对实信号具有共轭对称性的FFT整型变换.在此基础上,再进一步讨论DCT整型变换的问题.

### 3.1 对实信号保持共轭对称性的FFT整型变换

图2是对实信号进行FFT变换的流程示意图[5].其中符号X代表实数,O代表复数.弧线相连的两个O代表互为共轭的两个复数.

通过对流程图2的仔细分析,我们发现FFT变换中的蝶形运算共有4类.

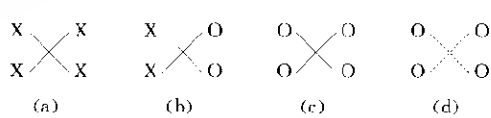



The symbol X indicates a real value, and O indicates complex value. Arcs show complex values that are conjugates of each other.①

①X代表实数,O代表复数,弧线连接一对共轭复数.

Fig. 2 FFT for real value

图2 实信号的FFT




X indicates real value, O indicates complex value.①

①X代表实数,O代表复数.

Fig. 3 Four kinds of butterfly for FFT

图3 FFT中的4类蝶形运算

(1) 所有的图 3(a)形式的蝶形变换矩阵都具有如下的实数形式:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad \text{其中 } a_1, a_2, b_1, b_2 \text{ 都是实数.}$$

因而可以根据式(2.5)写成实整数的可逆变换形式,并不影响后面的复共轭.

(2) 所有的图 3(b)形式的蝶形变换矩阵都具有如下的形式:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{i}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad \text{其中 } a_1, a_2 \text{ 是共轭复数, } b_1, b_2 \text{ 是实数.}$$

这个变换是本节问题的关键.只有当它所对应的整型变换的输出结果是共轭复整数时,才能保证整个 FFT 整型变换对实信号保持共轭对称性.在第 2 节中,由于不要求 $a_1, a_2$ 是共轭复整数,所以除以 $\sqrt{2}$ 是可行的.但是,当要求变换结果是共轭复整数的时候,除以 $\sqrt{2}$ 则是行不通的,我们目前的处理办法是对变换乘以 $\sqrt{2}$.变换的形式如下:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad \text{其中 } a_1, a_2 \text{ 是共轭复数, } b_1, b_2 \text{ 是实数.} \tag{3.1}$$

容易看出,此公式在 $b_1, b_2$ 是实整数的时候,$a_1, a_2$ 肯定是共轭复整数,并且变换是可逆的.这一点保证了 FFT 整型变换最后结果的对称复共轭性.

(3) 我们将所有的如图 3(c)所示的蝶形变换称为主蝶.主蝶变换的处理完全按照第 2 节中所给出的算法.这是因为在该变换中,$a_1, a_2, b_1, b_2$ 都是复整数,而且这些复整数之间并没有什么共轭约束,因而也不影响变换结果的对称共轭性.

(4) 我们将所有的如图 3(d)所示的蝶形变换称为虚蝶.这是因为,对于实信号 FFT 变换来说,这个变换根本不必要.如图 2 所示,每一个虚蝶两边的 4 个复值都有弧线和另一个主蝶相连.而弧线代表了两个复值的共轭特性.所以只要在正变换的时候,先计算主蝶变换的结果,再对其值取复共轭,赋给虚蝶;反变换的时候,先重建主蝶,再对主蝶的重建值取共轭赋给虚蝶.这种机制对浮点型复数和整型复数的变换都是一样的,既依靠主蝶变换保证了它的整型变换可逆性,又维持了变换结果的对称共轭性.

按照以上对 4 类蝶形变换的处理,对 FFT 一层一层地加以改造,最终将达到所要求的特性,即具有对称共轭特性的整型可逆变换.而上述第(2)类形式所造成的动态范围的增长也不是很大,它最多使图 2 中的每一个复值 $O$ 一致地增长 $\sqrt{2}$ 倍,而不管 FFT 变换了几层.通过实验,这一结果在计算机上得到了验证.

同时,我们将会看到,此处复值 $O$ 增长的 $\sqrt{2}$ 倍与 DCT 的定义、式(3.2)中的 $S[u]$ 相一致,从而使 DCT 所对应的整型变换与 DCT 浮点型变换保持动态范围的一致.

### 3.2 DCT 的整型变换

DCT 变换在图像压缩中起着重要的作用.JPEG 图像压缩方法就是采用 DCT 变换来降低图像数据的冗余度并提高压缩比的.快速 DCT 变换和 FFT 变换有着密切的关系[11].下面我们将利用这种关系来推导 DCT 的可逆整型变换.

对于 $N$ 点实信号 $F_N[n], n=0,1,\ldots,N-1$,DCT 变换的具体运算形式[9]如下:

$$C[u] = S[u]\sum_{n=0}^{N-1} F_N[n]\cos\left[\frac{(2n+1)u\pi}{2N}\right],$$

其中

$$S[u] = \begin{cases} \frac{1}{\sqrt{N}}, & u=0, \\ \frac{\sqrt{2}}{\sqrt{N}}, & u\neq 0. \end{cases} \tag{3.2}$$

如何用FFT的形式表示DCT变换呢[3,4]?首先,以$(2N-1)/2$为对称点,将$N$点信号$F_N[n]$,$n=0,1,\ldots,N-1$,扩展为$2N$点信号$F_{2N}[n]$,$n=0,1,\ldots,2N-1$,使之满足

$$F_{2N}[n]=\begin{cases}F_N[n], & n=0,1,\ldots,N-1\\ F_N[2N-n-1], & n=N,\ldots,2N-1\end{cases},$$

同时定义$F_N^e[n]=F_{2N}[2n]$,$F_N^o[n]=F_{2N}[2n+1]$.可以证明,DCT系数$\hat{C}[u]$满足如下公式(详细证明请见附录):

$$\hat{C}[u]=\frac{\sqrt{N}}{2}S[u]\{\hat{F}_N^e[u]\cdot W_{2N}^{u/2}+\overline{\hat{F}_N^e[u]}\cdot W_{2N}^{-u/2}\}. \tag{3.3}$$

下面我们将通过式(3.3)来构造DCT的可逆整数变换.首先我们对实整数信号$F_N^e[n]$作保持共轭对称性的FFT整型变换.

当$u=0$时,$\hat{F}_N^e[0]$是一个实整数,所以$\hat{C}[0]=\frac{\sqrt{N}}{2}S[0]\{\hat{F}_N^e[0]+\overline{\hat{F}_N^e[0]}\}=\hat{F}_N^e[0]$.

当$u=N/2$时,$\hat{F}_N^e[u]$是一个实整数,所以

$$\hat{C}[u]=\frac{\sqrt{N}}{2}S\left[\frac{N}{2}\right]\{\hat{F}_N^e[u]W_{2N}^{u/2}+\overline{\hat{F}_N^e[u]W_{2N}^{u/2}}\}=\frac{\sqrt{2}}{2}\hat{F}_N^e[u]\{W_{2N}^{N/4}+W_{2N}^{-N/4}\}=\hat{F}_N^e[u].$$

当$u\neq 0$且$u\neq N/2$时,由于我们已经在前面获得了保持共轭对称性的FFT整型变换,即满足

$$\sqrt{2}\,\hat{F}_N^e[u]=\overline{\sqrt{2}\,\hat{F}_N^e[N-u]}=x+yi,$$

其中$x$,$y$是已经得到的整数.

那么,令$\alpha=\frac{\pi u}{2N}$,则$W_{2N}^{u/2}=\cos\alpha+i\sin\alpha$,并且有

$$\hat{C}[u]=\frac{\sqrt{N}}{2}S[u]\{\hat{F}_N^e[u]W_{2N}^{u/2}+\overline{\hat{F}_N^e[u]W_{2N}^{u/2}}\}=x\cdot\cos\alpha-y\cdot\sin\alpha.$$

同理有

$$\hat{C}[N-u]=x\cdot\sin\alpha+y\cdot\cos\alpha,$$

写成矩阵形式为

$$\begin{bmatrix}\hat{C}[u]\\ \hat{C}[N-u]\end{bmatrix}=\begin{bmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}. \tag{3.4}$$

同样地,利用旋转矩阵的提升式(1.5),式(3.4)中的$\hat{C}[N-u]$与$\hat{C}[u]$的整数解可以统一获得.

这样,我们就解决了DCT的可逆整数变换问题,并且保持了变换的动态范围,完美地达到了目的.而且,我们猜想变换本身的动态范围已经不可能继续降低了.

### 3.3 DCT的整型变换算法总结

(1) 首先对$N$点实信号$F_N[n]$进行扩展,扩展为$2N$点对称实信号$F_{2N}[n]$.

(2) 通过$F_{2N}[n]$获得$F_N^e[n]$,使之满足$F_N^e[n]=F_{2N}[2n]$.

(3) 对$F_N^e[n]$进行保持共轭对称性的FFT可逆整型变换,获得$\hat{F}_N^e[0]$与$\hat{F}_N^e[N/2]$实整数和其他$\sqrt{2}\,\hat{F}_N^e[u]$的复整数形式.

(4) 利用所获得的数据进行DCT整数变换,获得$\hat{C}[u]$.

(5) 由于每一步变换都有相应的反变换,所以,整个DCT整数变换的逆变换可以相应地获得.

(6) 对于二维的数据,我们可以先逐行地进行一维DCT变换,再逐列地进行一维DCT变换.反变换的时候,应先逐列地进行一维DCT反变换,再逐行地进行一维DCT反变换,从而保证整数变换对二维数据的完全重建.

## 4 用DCT整型变换进行无失真图像压缩

我们利用上节所获得的DCT整型变换,对几幅标准测试图像进行了无失真压缩.作实验采用的是8×8的二维DCT整型可逆变换,编码方法是简单的算术编码,没有采用任何形式的条件熵.实验结果见表1.结果显

示,经 DCT 变换后,数据的冗余度降低,对变换后的数据进行算术编码比直接对原始数据进行算术编码具有更好的压缩性能.

DCT 整型变换不仅可以用于无失真图像压缩,也可以用于有失真图像压缩.如果我们将二者与渐进性编码方法结合起来,就可以达到渐进性直至无失真的灰度图像压缩.进一步地,文献[6]可以做到从 RGB 空间到 YCbCr 颜色空间的可逆整型变换,这样,如果将其和 DCT 整型变换相结合,我们就可以完成渐进性直至无失真的彩色图像压缩.

**Table 1** Experimental result for DCT integer transform

**表 1** DCT 整型变换的无失真图像压缩效果 (byte)

| Standard image① | Original size② | Arithmetic coding directly③ | DCT integer transform+arithmetic coding④ |
|---|---|---|---|
| Lena | 262 144 | 237 166 | 157 958 |
| Zelda | 262 144 | 229 142 | 146 500 |
| Girl | 262 144 | 216 736 | 153 388 |
| Barb | 262 144 | 239 026 | 170 803 |
| Plane | 262 144 | 203 268 | 155 801 |
| Flower | 262 144 | 220 888 | 132 934 |
| Goldhill | 262 144 | 222 413 | 172 176 |

①标准图像,②原始大小,③直接算术编码,④DCT 整型变换+算术编码.

当然,DCT 整型变换也可以用于其他形式数据的无损压缩,比如语音等一维信号.同时,DCT 变换被使用了很多年,具有大量成熟的快速算法.我们认为,考虑使用各种快速算法进行可逆整型变换也应是值得大家进一步深入探讨和研究的问题.事实上,本文所使用的方法正是借鉴了从 FFT 到 DCT 的快速算法[5],实验证明,这种方法具有与整型双正交小波变换基本相同的计算复杂度.文献[7]也可以实现 DCT 的整型变换,不过它使用的是另一种不同的快速算法.

## 参考文献


1 Calderbank R, Daubechies I, Sweldens W *et al*. Wavelet transforms that map integers to integers. Technical Report, Department of Mathematics, Princeton University, 1996. http://cm.bell labs.com/who/wim/papers/papers.html#integer

2 Daubechies I, Sweldens W. Factoring wavelet transforms into lifting steps. Technical Report, Bell Laboratories, Lucent Technologies, 1996

3 Xia De-shen, Fu De-sheng. Technology and Application for Modern Image Processing. Nanjing: Southeast Unversity Press, 1997
(夏德深,傅德胜.现代图像处理技术与应用.南京:东南大学出版社,1997)

4 Pennebaker W B, Mitchell J L. JPEG Still Image Data Compression Standard. New York: Van Nostrand Reinhold, 1993

5 Sorensen H V, Jones D L, Heideman M T *et al*. Real Value fast Fourier transform algorithms. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1987, ASSP-35(6): 849~863

6 Yan Yu-song, Shi Qing-yun. Reversible color space transform. Pattern Recognition and Artificial Intelligence, 1999, 12(1): 38~44
(闫宇松,石青云.颜色空间之间的可逆变换.模式识别与人工智能,1999,12(1):38~44)

7 Hong J. Discrete Fourier, Hartley and cosine transforms in signal processing [Ph.D. Thesis]. Columbia University, 1993


## 附 录

**定理**. 在 $N$ 点实信号 $F_N[n]$, $n=0,1,\ldots,N-1$, 如果定义

$$F_{2N}[n]=\begin{cases}F_N[n], & n=0,1,N-1\\ F_N[2N-n-1], & n=N,\ldots,2N-1\end{cases},$$

$$F'_N[n]=F_{2N}[2n],\quad F''_N[n]=F_{2N}[2n+1],$$

那么 $F_N[n]$ 的 DCT 系数 $\hat{C}[u]$ 满足下式:

$$\hat{C}[u]=\frac{\sqrt{N}}{2}S[u]\{\hat{F}'_N[u]\cdot W_{2N}^{u/2}+\overline{\hat{F}''_N[u]}\cdot W_{2N}^{u/2}\},$$

其中 $\hat{F}_N^e[u]$是 $F_N^e[n]$的 FFT 系数，$W_N=\exp\left[\frac{-i2\pi j}{N}\right]$，$S[u]$如式(3.2)所示.

证明：对 $F_{2N}[i]$进行 FFT 变换，得到下式：

$$\begin{aligned}\hat{F}_{2N}[u]&=\frac{1}{\sqrt{2N}}\sum_{n=0}^{2N-1}F_{2N}[n]W_{2N}^{nu}=\frac{1}{\sqrt{2N}}\left\{\sum_{n=0}^{N-1}F_{2N}[n]W_{2N}^{nu}+\sum_{n=N}^{2N-1}F_{2N}[n]W_{2N}^{nu}\right\}\\&=\frac{1}{\sqrt{2N}}\left\{\sum_{n=0}^{N-1}F_N[n]W_{2N}^{nu}+\sum_{n=N}^{2N-1}F_N[2N-n-1]W_{2N}^{nu}\right\}\\&=\frac{1}{\sqrt{2N}}\left\{\sum_{n=0}^{N-1}F_N[n]W_{2N}^{nu}+\sum_{n=0}^{N-1}F_N[n]W_{2N}^{(2N-1-n)u}\right\}\\&=\frac{1}{\sqrt{2N}}\left\{\sum_{n=0}^{N-1}F_N[n]W_{2N}^{nu}+\sum_{n=0}^{N-1}F_N[n]W_{2N}^{(-1-n)u}\right\},\end{aligned}$$

所以

$$\begin{aligned}\frac{\sqrt{2N}}{2}S[u]W_{2N}^{u/2}\hat{F}_{2N}[u]&=\frac{S[u]}{2}\left\{\sum_{n=0}^{N-1}F_N[n]W_{2N}^{(n+\frac{1}{2})u}+\sum_{n=0}^{N-1}F_N[n]W_{2N}^{-(n+\frac{1}{2})u}\right\}\\&=S[u]\sum_{n=0}^{N-1}F_N[n]\cos\left[\frac{(2n+1)u\pi}{2N}\right]=\hat{C}[u].\end{aligned}$$

另外，

$$\begin{aligned}\hat{F}_{2N}[u]&=\frac{1}{\sqrt{2N}}\sum_{n=0}^{2N-1}F_{2N}[n]W_{2N}^{nu}=\frac{1}{\sqrt{2N}}\left\{\sum_{n=0}^{N-1}F_{2N}[2n]W_{2N}^{2nu}+\sum_{n=0}^{N-1}F_{2N}[2n+1]W_{2N}^{(2n+1)u}\right\}\\&=\frac{1}{\sqrt{2N}}\left\{\sum_{n=0}^{N-1}F_N^e[n]W_N^{nu}+\sum_{n=0}^{N-1}F_N^o[n]W_N^{nu}W_{2N}^u\right\}=\frac{1}{\sqrt{2}}\{\hat{F}_N^e[u]+\hat{F}_N^o[u]W_{2N}^u\},\end{aligned}$$

其中 $F_N^e[n]=F_{2N}[2n]$，$F_N^o[n]=F_{2N}[2n+1]$.

又因为 $F_{2N}$是对称的，满足 $F_{2N}[2N-1-i]=F_{2N}[i]$，所以有

$$F_N^o[N-n-1]=F_{2N}[2(N-n-1)+1]=F_{2N}[2N-2n-1]=F_{2N}[2n]=F_N^e[n].$$

由此可以证明：$\hat{F}_N^o[u]=\overline{\hat{F}_N^e[u]}\cdot W_N^u$. 所以有 DCT 系数 $\hat{C}[u]$满足下式：

$$\begin{aligned}\hat{C}[u]&=\frac{\sqrt{2N}}{2}S[u]W_{2N}^{u/2}\hat{F}_{2N}[u]=\frac{\sqrt{N}}{2}S[u]W_{2N}^{u/2}\{\hat{F}_N^e[u]+\hat{F}_N^o[u]W_{2N}^u\}\\&=\frac{\sqrt{N}}{2}S[u]\{\hat{F}_N^e[u]\cdot W_{2N}^{u/2}+\overline{\hat{F}_N^e[u]\cdot W_N^u}W_{2N}^uW_{2N}^{u/2}\}\\&=\frac{\sqrt{N}}{2}S[u]\{\hat{F}_N^e[u]\cdot W_{2N}^{u/2}+\overline{\hat{F}_N^e[u]}W_{2N}^{u/2}\}.\end{aligned}$$

□

## Reversible DCT Mapping Integers to Integers and Lossless Image Compression

YAN Yu-song　SHI Qing-yun

(*National Laboratory on Machine Perception　Center for Information Science　Beijing University　Beijing*　100871)

**Abstract**　In this paper, the authors describe the precedures of DCT (discrete cosine transform) and FFT (fast Fourier transform) which map integers to integers by using lifting scheme and the butterfly configuration of FFT. The transform is reversible, fast and suitable for the lossless image compressions.

**Key words**　FFT (fast Fourier transform), DCT (discrete cosine transform), lifting scheme, transform from integer to integer, lossless image compression.