国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第14卷第1期　1992年3月　Vol. 14 No. 1

二维离散余弦变换与二维离散Fourier变换的快速算法

蒋增荣　成礼智
（系统工程与应用数学系）

摘　要　文中提出 $N\times M$2D－DCT(Ⅱ)的一种快速算法，其需实运算量为：$M_u=\frac{1}{2}NM\log_2N+\frac{1}{4}MN\log_2M$，$A_d=\frac{3}{2}NM\log_2NM-3MN-\frac{1}{2}M^2+M+N$（其中 N、M 为2的幂）。当 $N=M$ 时，与文[5]的结果一样，这是目前最好的结果。但文[5]算法不稳定，容易产生较大的误差。本文克服了这一缺点。并利用此2D－FCT（Ⅱ）导出了2D－DCT. 2D－DST和2D－DCST的快速算法及2D－DFT的一种快速算法。2D－DFT快速算法的运算量与文[1]中用FPT计算2D－DFT相近。

关键词　快速算法，离散余弦变换，离散富里叶变换
分类号　O241

在图象处理、编码与模式识别等领域中，二维离散余弦变换（2D-DCT）已成为有用的工具，其快速算法的研究受到人们的重视。由于2D-DCT核的分离性，可以首先建立1D-DCT的快速算法，然后利用行列法来计算2D-DCT. 但这样做的运算量比较大。1985年，M. A. Haque提出了一种2D-FCT算法[5]，所需运算量比文[2]用FFT计算2D-DCT要少得多。但从推导过程看，文[5]只提供了 $N\times N$（N 是2的幂）的2D-DCT算法，且其乘法运算主要是一个数与$1/\left(4\cos\frac{2k+1}{2N}\pi\cos\frac{2j+1}{2M}\pi\right)\left(0\leqslant k\leqslant\frac{N}{2}-1;\ 0\leqslant j\leqslant\frac{M}{2}-1\right)$相乘，容易产生较大的误差或溢出。本文对一般的（限制于2的幂）$N\times M$二维余弦变换提出了一种新的快速算法，本文算法中的乘法主要是一个数与$4\cos\frac{2k+1}{2N}\pi\cos\frac{2j+1}{2M}\pi$相乘，由于$\left|4\cos\frac{2k+1}{2N}\pi\cos\frac{2j+1}{2M}\pi\right|\leqslant 4$，从而克服了文[5]中算法的不稳定性。当 $N=M$ 时，本文算法所需运算量与文[5]相同。利用本文提出的2D-FCT算法，给出了2D-DFT新的快速算法，与文[1]比较，当 $N=M$ 时，乘法次数相近，加法次数略多于文[1]；但当 $N\geqslant M^2$ 时，乘法与加法次数均少于文[1].

1　二维离散正、余弦变换以及正余弦混合型变换的定义

设 $M=2^r$，$N=2^t(t\geq r)$，$\{y_{n,m}\}$、$\{Y_{n,m}\}$为二维实序列，表示为

$$y=(y_{n,m}),Y=(Y_{n,m})$$

* 1991年2月2日收稿　1991年8月19日收修改稿

下面用矩阵形式表示所有各类二维正弦、余弦以及正余混合变换：

$$\text{2D-DCT(Ⅱ)}: Y = [C_N^{\mathrm{I}}]y[C_M^{\mathrm{I}}]^T, y = [C_N^{\mathrm{I}}]^T Y[C_M^{\mathrm{I}}] \tag{1}$$

$$\text{2D-DCT(Ⅰ)}: Y = [C_{N+1}^{\mathrm{I}}]y[C_{M+1}^{\mathrm{I}}]^T, y = [C_{N+1}^{\mathrm{I}}]^T Y[C_{M+1}^{\mathrm{I}}] \tag{2}$$

$$\text{2D-DCT(Ⅰ、Ⅱ)}: Y = [C_{N+1}^{\mathrm{I}}]y[C_M^{\mathrm{I}}], y = [C_{N+1}^{\mathrm{I}}]^T Y[C_M^{\mathrm{I}}]^T \tag{3}$$

$$\text{2D-DCST(Ⅰ)}: Y = [C_{N+1}^{\mathrm{I}}]y[S_{M-1}^{\mathrm{I}}]T, y = [C_{N+1}^{\mathrm{I}}]^T Y[S_{M-1}^{\mathrm{I}}] \tag{4}$$

$$\text{2D-DCST(Ⅰ、Ⅱ)}: Y = [C_N^{\mathrm{I}}]^T y[S_{M-1}^{\mathrm{I}}]^T, y = [C_N^{\mathrm{I}}]Y[S_{M-1}^{\mathrm{I}}] \tag{5}$$

$$\text{2D-DST(Ⅰ)}: Y = [S_{N-1}^{\mathrm{I}}]y[S_{M-1}^{\mathrm{I}}]^T, y = [S_{N-1}^{\mathrm{I}}]^T Y[S_{M-1}^{\mathrm{I}}] \tag{6}$$

其中

$$[C_N^{\mathrm{I}}] = \sqrt{\frac{2}{N}}\left[e_n\cos\frac{(2k+1)n}{2N}\pi\right] \qquad (n,k = 0,1,2,\cdots,N-1)$$

$$[C_{N+1}^{\mathrm{I}}] = \sqrt{\frac{2}{N}}\left[e_i e_j\cos\frac{ij}{N}\pi\right] \qquad (i,j = 0,1,2,\cdots,N)$$

$$[S_{N-1}^{\mathrm{I}}] = \sqrt{\frac{2}{N}}[\sin\frac{ij}{N}\pi] \qquad (i,j = 1,2,\cdots,N-1)$$

当 $i=0$ 或 N 时，$e_i=1/\sqrt{2}$，否则 $e_i=1$.

由于逆变换矩阵是正变换矩阵的转置，如果导出了正变换的快速算法，只需反方向进行就可得到逆变换的快速算法。因此我们只需讨论各类正变换的快速算法。

2 2D-FCT(Ⅱ)的算法推导

2.1 几个矩阵

$$[D_l] = (d_{ij})_{l\times l} \qquad d_{ij} = \begin{cases} 0 & i<j \\ (-1)^{i+j} & 1\leqslant j\leqslant i\leqslant l \end{cases} \tag{7}$$

$$[\bar{I}_l] = (a_{ij})_{l\times l} \qquad a_{ij} = \begin{cases} 1 & i+j=l+1 \\ 0 & \text{其 它} \end{cases} \tag{8}$$

$$[P_k] = (P_{ij})_{k\times k} \qquad p_{ij} = \begin{cases} 1 & i+1=2j \text{ 或 } 2j=i+2\left[\frac{k+1}{2}\right] \\ 0 & \text{其 它} \end{cases} \tag{9}$$

(8)、(9)式都是置换矩阵。

$$[A_l^l] = \begin{bmatrix} I_{l/2} & \bar{I}_{l/2} \\ I_{l/2} & -\bar{I}_{l/2} \end{bmatrix} \qquad (l\text{ 为偶数}) \tag{10}$$

$[A_{l/2}^{\mathrm{I}}]=\mathrm{diag}\{2C_{2l}^1, 2C_{2l}^3, \cdots, 2C_{2l}^{l-1}\}$ （l 为偶数），diag 表示对角矩阵，

$$C_{2l}^i = \cos\frac{i\pi}{2l} \tag{11}$$

$$J_l = \mathrm{diag}\left\{I_{l/2}, \frac{1}{2}, I_{l/2-1}\right\} \qquad (l\text{ 为偶数}) \tag{12}$$

2.2 算法推导

设 $[C_l^{\mathrm{I}}]=\sqrt{\frac{2}{l}}\mathrm{diag}\left\{\frac{\sqrt{2}}{2}, I_{l-1}\right\}[C_l]$ （$l=N$ 或 M），并令 $\bar{Y}=[C_N]y[C_M]^T$.

则

$$Y = \frac{2}{\sqrt{NM}}\operatorname{diag}\left\{\frac{\sqrt{2}}{2}, I_{N-1}\right\}\overline{Y}\operatorname{diag}\left\{\frac{\sqrt{2}}{2}, I_{M-1}\right\} \tag{13}$$

现在计算 $\overline{Y}$. 在文[4]中已建立了递推关系式：

$$[C_l] = L_l(1)\operatorname{diag}\{C_{l/2}, C_{l/2}\}V_l(1) \quad (l \geq 4) \tag{14}$$

其中

$$L_l(1) = [P_l]\operatorname{diag}\{I_{l/2}, D_{l/2}\}\cdot J_l, V_l(1) = \operatorname{diag}\{I_{l/2}, A^{\mathrm{I}}_{l/2}\}[A^{\mathrm{I}}_l]$$

因此，若记 $y_1 = V_N(1)y(V_M(1))^T$，就有

$$\overline{Y} = L_N(1)\operatorname{diag}\{C_{N/2}, C_{N/2}\}y_1\operatorname{diag}\{C^T_{M/2}, C^T_{M/2}\}L^T_M(1) \tag{15}$$

将 y_1 分为四个 $N/2\times M/2$ 的子阵块

$$y_1 = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix}$$

为计算 y_1，先计算

$$[A^{\mathrm{I}}_N]y[A^{\mathrm{I}}_M]^T = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

由式(10)直接作矩阵乘法，有

$$\begin{aligned}
[A_{11}] &= (a_{ij}) = [(y_{ij} + y_{N-1-i,j}) + (y_{i,M-1-j} + y_{N-1-i,M-1-j})] \\
[A_{12}] &= (b_{ij}) = [(y_{ij} + y_{N-1-i,j}) - (y_{i,M-1-j} + y_{N-1-i,M-1-j})] \\
[A_{21}] &= (c_{ij}) = [(y_{ij} - y_{N-1-i,j}) + (y_{i,M-1-j} - y_{N-1-i,M-1-j})] \\
[A_{22}] &= (d_{ij}) = [(y_{ij} - y_{N-1-i,j}) - (y_{i,M-1-j} - y_{N-1-i,M-1-j})] \\
i &= 0,1,\cdots,N/2-1; j = 0,1,2,\cdots,M/2-1.
\end{aligned} \tag{16}$$

直接计算得

$$y_1 = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12}A^{\mathrm{I}}_{M/2} \\ A^{\mathrm{I}}_{N/2}A_{21} & A^{\mathrm{I}}_{N/2}A_{22}A^{\mathrm{I}}_{M/2} \end{bmatrix} \tag{17}$$

故

$$\begin{gathered}
y_{11} = A_{11} = (a_{ij}), y_{12} = [A_{12}A^{\mathrm{I}}_{M/2}] = (2C^{2j+1}_{2M}b_{ij}) \\
y_{21} = [A^{\mathrm{I}}_{N/2}A_{21}] = (2C^{2i+1}_{2N}\cdot C_{ij}), y_{22} = (4C^{2i+1}_{2N}C^{2j+1}_{2M}d_{ij})
\end{gathered} \tag{18}$$

$$\overline{Y} = L_N(1)\begin{bmatrix} C_{N/2}y_{11}C^T_{M/2} & C_{N/2}y_{12}C^T_{M/2} \\ C_{N/2}y_{21}C^T_{M/2} & C_{N/2}y_{22}C^T_{M/2} \end{bmatrix}L^T_M(1) \tag{19}$$

从式(19)看出，$N\times M$ 的 2D-DCT(Ⅱ)可通过 4 个 $N/2\times M/2$ 点的 2D-DCT(Ⅱ)，然后左、右各乘以 $L_N(1)$和 $L^T_M(1)$来实现。$N/2\times M/2$ 的 2D-DCT(Ⅱ)可重复进行(19)式的运算，如此递推下去，共做 $r-1$ 次，得到 4^{r-1}个 $N_1\times 2(N_1=2^{t-r+1})$点的 2D-DCT(Ⅱ). 上述递推过程可表示如下。

设 $L_N(m) = \operatorname{diag}\{L_{N/2}(m-1), L_{N/2}(m-1)\}, V_N(m) = \operatorname{diag}\{V_{N/2}(m-1), V_{N/2}(m-1)\}$ $(2\leqslant m\leqslant r-1)$；

$$y_0 = y, y_i = V_N(i)y_{i-1}V^T_M(i), 1\leqslant i\leqslant r-1, y_{r-1} = (y^{(r-1)}_{ij}), (y^{(r-1)}_{ij}\text{为 } N_1\times 2 \text{ 阶矩阵}) \tag{20}$$

则

$$\overline{Y} = L_N(1)L_N(2)\cdots L_N(r-1)(C_{N_1}y^{(r-1)}_{ij}C^T_2)L^T_M(r-1)\cdots L^T_M(1) \tag{21}$$

又设 $y_{ij}^{(r-1)}=(a_{mn}^{(i,j)})_{N_1\times 2},\overline{Y}_{ij}^{(r-1)}=V_N(1)y_{ij}^{(r-1)}C_2^T$，则由式(21)计算 $\overline{Y}$ 的步骤为：

(1) 从 $i=1$ 到 $r-1$ 计算 $y_i=V_N(i)y_{i-1}V_M^T(i)$

(2) 计算

$$\overline{Y}_{ij}^{(r-1)}=\begin{bmatrix}V_1 & V_2\\ V_3 & V_4\end{bmatrix}=\left[\begin{array}{c:c} a_{00}^{(i,j)}+a_{01}^{(i,j)} & 2C_4'(a_{00}^{(i,j)}-a_{01}^{(i,j)})\\ \vdots & \vdots\\ a_{N_1/2-1,0}^{(i,j)}+a_{N_1/2-1,1}^{(i,j)} & 2C_4'(a_{N_1/2-1,0}^{(i,j)}-a_{N_1/2-1,1}^{(i,j)})\\ \hdashline a_{N_1/2,0}^{(i,j)}+a_{N_1/2,1}^{(i,j)} & 2C_4'(a_{N_1/2,0}^{(i,j)}-a_{N_1/2,1}^{(i,j)})C_{2N_1}'\\ \vdots & \vdots\\ a_{N_1-1,0}^{(i,j)}+a_{N_1-1,1}^{(i,j)} & 2C_4'(a_{N_1-1,0}^{(i,j)}-a_{N_1-1,1}^{(i,j)})C_{2N_1}' \end{array}\right] \tag{22}$$

(V_1，V_2，V_3，V_4 是长为 $N_1/2$ 的向量)

$$Y_{ij}^{(r-1)}=\begin{bmatrix}I_N & \\ & D_{N_1/2}\end{bmatrix}L_{N_1}(1)\begin{bmatrix}C_{N_1/2}V_1 & C_{N_1/2}V_2\\ C_{N_1/2}V_3 & C_{N_1/2}V_4\end{bmatrix}=C_{N_1}y_{ij}^{(r-1)}C_2^T \tag{23}$$

从式(23)知，求 $Y_{ij}^{(r-1)}$ 可归结为 4 个 $\frac{N_1}{2}$ 的 1D-DCT(Ⅱ)的计算，而 1D-DCT(Ⅱ)可采用文[4]中方法进行，然后作矩阵乘法。

(3) 令 $T_0=(C_{N_1}y_{ij}^{(r-1)}C_2^T)$，$T_i=L_N(r-i)T_{i-1}L_M^T(r-i)$，$(i=1,2,\cdots,r-1)$，依次由 $i=1$ 到 $r-1$ 计算 T_i，则 $\overline{Y}=Y_{r-1}$.

利用上述过程计算 $\overline{Y}$ 的运算量为：

$$M_\mu=\frac{1}{2}MN\log_2 N+\frac{1}{4}MN\log_2 M$$

$$A_d=\frac{3}{2}MN\log_2(MN)-3MN-\frac{M^2}{2}+(M+N) \tag{24}$$

3 利用 2D-FCT(Ⅱ)计算其它二维离散正、余弦变换以及 2D-DFT

3.1 2D-DCT(Ⅰ、Ⅱ)的计算

设 $y=(y_{ij}),\hat{y}=\frac{2}{\sqrt{NM}}(e_j\cdot y_{ij})$，$[A_{N+1}]=(C_N^{ij})$，$\overline{Y}=[A_{N+1}]\hat{y}[C_M]$

由(3)有

$$Y=\operatorname{diag}\left\{\frac{\sqrt{2}}{2},I_{N-1},\frac{\sqrt{2}}{2}\right\}\overline{Y} \tag{25}$$

由文[4]有

$$[A_{N+1}]=[P_{N+1}]\operatorname{diag}\{A_{N/2+1},C_{N/2}^T\}[B_{N+1}^I] \tag{26}$$

其中 $$[B_{N+1}^I]=\begin{bmatrix}I_{N/2} & \bar{I}_{N/2}\\ I_{N/2} & -\bar{I}_{N/2}\end{bmatrix}$$

由式(14)与式(26)就得到

$$\overline{Y}=[P_{N+1}]\operatorname{diag}\{A_{N/2+1},C_{N/2}^T\}([B_{N+1}^I]\hat{y}L_M(1))\operatorname{diag}\{C_{M/2},C_{M/2}\}V_M(1) \tag{27}$$

记
$$\hat{y}^{(1)} = [B^{I}_{N+1}]\hat{y}L_M(1) = \begin{bmatrix} \hat{y}_{11} & \hat{y}_{12} \\ \hat{y}_{21} & \hat{y}_{22} \end{bmatrix} \tag{28}$$

其中 $\hat{y}_{11}$与 $\hat{y}_{12}$是$\left(\frac{N}{2}+1\right)\times\frac{M}{2}$阶矩阵，$\hat{y}_{21}$与 $\hat{y}_{22}$是$\frac{N}{2}\times\frac{M}{2}$阶矩阵。

直接计算(28)式只需$\frac{3}{2}MN$ 次加法，于是(27)式可化为

$$\overline{Y} = [P_{N+1}]\begin{bmatrix} A_{\frac{N}{2}+1}\hat{y}_{11}C_{M/2} & A_{N/2+1}\hat{y}_{12}C_{M/2} \\ C^T_{N/2}\hat{y}_{21}C_{M/2} & C^T_{N/2}\hat{y}_{22}C_{M/2} \end{bmatrix}V_M(1) \tag{29}$$

从(29)式可以看出，一个$(N+1)\times M$ 的 2D-DCT(Ⅰ、Ⅱ)可以化为两个$\left(\frac{N}{2}+1\right)\times\frac{M}{2}$的 2D-DCT(Ⅰ、Ⅱ)与两个$\frac{N}{2}\times\frac{M}{2}$的 2D-DCT(Ⅱ)。对于$\left(\frac{N}{2}+1\right)\times\frac{M}{2}$的 2D-DCT(Ⅰ、Ⅱ)继续利用(29)式递推下去，共 $r-1$ 步，最后得到 2^{r-1}个$(N_1+1)\times 2(N_1=2^{t-r+1})$的 2D-DCT(Ⅰ、Ⅱ)与 2^{r-1}个 $N_1\times 2$ 的 2D－DCT(Ⅱ)。在上述递推过程中，2D-DCT(Ⅱ)均采用本文§2 的方法计算。下面计算$(N_1+1)\times 2$ 的 2D-DCT(Ⅰ、Ⅱ)。

设 $\overline{Y}'=[A_{N_1+1}]y'[C_2]$，$\overline{Y}'$与 y'均为$(N_1+1)\times 2$ 矩阵。首先计算 $Z=y'[C_2]$

$$Z = \begin{bmatrix} y'_{00} + C^1_4 y'_{01} & y'_{00} - C^1_4 y'_{01} \\ \vdots & \vdots \\ y'_{N_1 0} + C^1_4 y'_{N_1 1} & y'_{N_1 0} - C^1_4 y'_{N_1 1} \end{bmatrix} = (Z_1, Z_2) \tag{30}$$

其中 Z_1，Z_2 是两个长为 N_1+1 的向量。

由式(30)得到$\overline{Y}'=(A_{N_1+1}Z_1, A_{N_1+1}Z_2)$。这样，$\overline{Y}'$可化为两个 N_1+1 的 1D-DCT(Ⅰ)。采用文[4]方法计算 $\overline{Y}'$需乘法 $N_1\log_2 N_1-2N_1+2$ 次，加法 $3N_1\log_2 N_1-4N_1+2\log_2 N_1+8$ 次。

按上述过程计算$(N+1)\times M$ 的 2D-DCT(Ⅰ、Ⅱ)总的运算量为

$$\begin{cases} M_u = \frac{1}{4}NM\log_2 M + \frac{NM}{2}\log_2 N - \frac{3}{2}MN + \frac{5}{2}N \\ A_d = \frac{3MN}{2}\log_2 NM - \frac{15}{2}MN - \frac{M^2}{2} + M\log_2 N + N\log_2 M + 9N + 5M \end{cases} \tag{31}$$

类似于 2D-DCT(Ⅰ、Ⅱ)，可得到 2D-DCT(Ⅰ)、2D-DST(Ⅰ)、2D-DCST(Ⅰ)、2D-DCST(Ⅰ、Ⅱ)的快速算法，所需的实乘与实加运算量见表 1.

3.2 2D-DFT 的快速算法

设

$$H_1(N) = \begin{bmatrix} I_{N/2} & & -i\dot{\bar{I}}_{N/2-1} \\ & 1 & \\ 0\ \bar{I}_{N/2-1} & & i\bar{I}_{N/2-1} \end{bmatrix} \qquad H_2(N) = \begin{bmatrix} I_{N/2} & & \dot{\bar{I}}_{N/2-1} \\ & 1 & \\ 0\ I_{N/2-1} & & -\bar{I}_{N/2-1} \end{bmatrix}$$

N 阶 Fourier 变换矩阵为

$$[F_N] = \left[\exp\left\{-\frac{2mn\pi}{N}i\right\}\right], i = \sqrt{-1} \tag{32}$$

由文[4]有

$$[F_N] = H_1(N)\mathrm{diag}\{A_{N/2+1}, S_{N/2-1}\}H_2(N) \tag{33}$$

设 Y、y 为两个 $N\times M$ 点的复常数序列，y 的实部、虚部分别为 $\mathrm{Re}y$、$\mathrm{Im}y$. 将 2D-DFT 表示为

$$\begin{aligned} Y &= [F_N]y[F_M]^T \\ &= H_1(N)\mathrm{diag}\{A_{N/2+1}, S_{N/2-1}\}(H_2(N)yH_2^T(M)\mathrm{diag}\{A_{M/2+1}^T, S_{N/2-1}^T\}H_1^T(M) \end{aligned} \tag{34}$$

设

$$\begin{aligned} \tilde{y} = \tilde{y}_1 + i\tilde{y}_2 &= H_2(N)\mathrm{Re}yH_2^T(M) + iH_2(N)\mathrm{Im}yH_2^T(M) \\ &= \begin{bmatrix} \tilde{y}_{11}^{(1)} & \tilde{y}_{12}^{(1)} \\ \tilde{y}_{21}^{(1)} & \tilde{y}_{22}^{(1)} \end{bmatrix} + i\begin{bmatrix} \tilde{y}_{11}^{(2)} & \tilde{y}_{12}^{(2)} \\ \tilde{y}_{21}^{(2)} & \tilde{y}_{22}^{(2)} \end{bmatrix} \end{aligned} \tag{35}$$

直接计算(35)式只需 $4MN-4(M+N)$ 次实数加，将(35)式代入(34)式得到

$$\begin{aligned} Y &= H_1(N)\begin{bmatrix} A_{N/2+1}\tilde{y}_{11}^{(1)}(A_{M/2+1})^T & A_{N/2+1}\tilde{y}_{12}^{(1)}S_{M/2-1}^T \\ S_{N/2-1}\tilde{y}_{21}^{(1)}A_{M/2+1}^T & S_{N/2-1}\tilde{y}_{22}^{(1)}S_{M/2-1}^T \end{bmatrix}H_1^T(M) \\ &\quad + iH_1(N)\begin{bmatrix} A_{N/2+1}\tilde{y}_{11}^{(2)}A_{M/2+1}^T & A_{N/2+1}\tilde{y}_{12}^{(2)}S_{M/2-1}^T \\ S_{N/2-1}\tilde{y}_{21}^{(2)}A_{M/2+1}^T & S_{N/2-1}\tilde{y}_{22}^{(2)}S_{M/2-1}^T \end{bmatrix}H_1^T(M) \end{aligned} \tag{36}$$

由(36)式知道，$N\times M$ 的 2D-DFT 可分解为 2 个 2D-DCT(Ⅰ)，4 个 2D-DCST(Ⅰ)，2 个 2D-DST(Ⅰ)。对于这些二维离散正、余弦及其混合型变换，均可采用前面方法快速计算，然后左、右分别乘上 $H_1(N)$ 与 $H_1^T(M)$。左、右乘只需 $4NM-4(N+M)$ 次实数加，计算(36)式所需的实运算量为：

$$\begin{cases} M_u = NM\log_2 N + \dfrac{NM}{2}\log_2 M - \dfrac{9}{2}NM + 4(N+M) \\ A_d = 3NM\log_2 NM - 5NM + (2N+M)\log_2 N + (2M+N)\log_2 M \end{cases} \tag{37}$$

当输入序列 y 为实数时，实运算量为

$$\begin{cases} M_u = \dfrac{1}{2}NM\log_2 N + \dfrac{1}{4}NM\log_2 M - \dfrac{9}{4}NM + 2(N+M) \\ A_d = \dfrac{3}{2}NM\log_2 NM - \dfrac{5}{2}NM + \left(N + \dfrac{M}{2}\right)\log_2 N + \left(M + \dfrac{N}{2}\right)\log_2 M \end{cases} \tag{38}$$

由式(37)、(38)不难看出，利用 2D-FCT 算法来计算 2D-DFT，所需实乘比基 2 *FFT* 减少 50%左右，实加减少 15%左右。与文[1]的快速多项式变换 (FPT) 计算 2D-DFT 比较，当 $N=M$ 时，乘法次数相同，加法次数略多于文[1]. 但当 $N\geqslant M^2$ 时，乘法与加法次数均少于 DFT 的 FPT 算法。

表 1 列出了上述各类 $N\times M$ 的 DCT、DST 以及 DFT 所需的实运算量公式，其中 $N=2^t$，$M=2^r$，$t\geqslant r\geqslant 2$.

表 1　$N\times M$ 各类 DCT、DST 和 DFT 所需的运算量

类　别	实乘次数	实加次数
2-DCT(Ⅱ)	$\frac{MN}{2}\log_2 N+\frac{MN}{4}\log_2 M$	$\frac{3}{2}MN\log_2 MN-3MN-\frac{M^2}{2}+M+N$
2D-DCT(Ⅰ、Ⅱ)	$\frac{MN}{2}\log_2 N+\frac{NM}{4}\log_2 M-\frac{3}{2}MN+\frac{5}{2}N$	$\frac{3}{2}MN\log_2 MN-\frac{15}{2}MN-\frac{M^2}{2}+9N+5M+M\log_2 N+N\log_2 M$

续表

类别	实乘次数	实加次数
2D-DCT(Ⅰ)	$\frac{MN}{2}\log_2 N+\frac{NM}{4}\log_2 M-MN+M+2N$	$\frac{3}{2}MN\log_2 MN-6MN-M^2+4N+M+N\log_2 N$
2D-DCST(Ⅰ、Ⅱ)	$\frac{MN}{2}\log_2 N+\frac{NM}{4}\log_2 M-\frac{3}{2}MN$	$\frac{3}{2}MN\log_2 MN-6MN+4N-3N-\frac{M^2}{2}$
2D-DST(Ⅰ)	$\frac{MN}{2}\log_2 N+\frac{MN}{4}\log_2 M-2MN+N+M$	$\frac{3MN}{2}\log_2 NM-8MN-\frac{M^2}{2}+M\log_2 N+N\log_2 M$
2D-DCST(Ⅰ)	$\frac{MN}{2}\log_2 N+\frac{MN}{4}\log_2 M-\frac{3}{2}MN+M+\frac{N}{2}$	$\frac{3MN}{2}\log_2 NM-5MN-\frac{M^2}{2}+\frac{N}{2}\log_2 M+M\log_2 N$
2D-DFT(输入复序列)	$NM\log_2 N+\frac{1}{2}NM\log_2 M-\frac{9}{2}MN+4(N+M)$	$3MN\log_2 NM-5MN+(2N+M)\log_2 N+(2M+N)\log_2 M$
2D-DFT(输入实序列)	$\frac{1}{2}NM\log_2 N+\frac{1}{4}NM\log_2 M-\frac{9}{4}NM+2(N+M)$	$\frac{3}{2}NM\log_2 NM-\frac{5}{2}NM+\left(N+\frac{M}{2}\right)\log_2 N+\left(M+\frac{N}{2}\right)\log_2 M$

参考文献

1 蒋增荣. 用多项式变换 (FPT) 计算二维离散富里叶变换 (DFT). 高等学校计算数学学报, 1984(2)

2 Makhoul J. A Fast Consine Transform in One and Two Dimension. IEEE. Trans. 1980, Assp—28(1)

3 Lee B G. A New Algorithm to Compute the Discrete Consine Transform. IEEE. Trans. 1984, Assp—32(6)

4 蒋增荣,成礼智.离散余弦变换(DCT)与离散富里叶变换(DFT)的快速算法.高等学校计算数学学报,1990,(4)

5 Haque H A. A Two—Dimension Fast Consine Transform. IEEE. Trans. 1985, Assp-13

Fast Algorithm for Two-dimensional Discrete Cosine Transform and Two-dimensional Discrete Fourier Transform

Jiang Zengrong　Cheng Lizhi

(Department of System Engineering and Applied Mathematics)

Abstract

A fast algorithm for computing N×M 2D-DCT(Ⅱ) is developed. The amount of the real operations are: $M_u=\frac{1}{2}NM\log_2 N+\frac{1}{4}NM\log_2 M$, $A_d=\frac{3}{2}NM\log_2 NM-3MN-\frac{1}{2}M^2+M+N$. where $N=2^t, M=2^r$. When $N=M$, the result is the same as in [5], which is the best result at present. But the algorithm in [5] has instability leading to large errors easily. This limitation has been overcome in the paper. We also derive some algorithms about 2D-DCT, 2D-DST, 2D-DCST. A fast algorithm about 2D—DFT is developed in particular, the amount of the operations in paper is similar to that in [1] with FPT to calculate 2D-DFT

Key words　fast algorithm, discrete cosine transform, discrete Fourier transform