第16卷 第6期　　电子科学学刊　　Vol. 16, No. 6
1994年11月　　JOURNAL OF ELECTRONICS　　Nov. 1994

二维离散余弦变换的一种新的快速算法

卢颉　王新成　朱维乐
（清华大学电子工程系　北京　100084）

摘要　本文介绍了二维离散余弦变换（DCT）的一种新的快速算法.对于 $N\times N$ DCT($N=2^m$)，只需用 N 个一维 DCT 和若干加法运算. 与常规的行-列法相比，所需的乘法运算量减少了一半，也比其它的快速算法的乘法运算量要少，而加法运算量基本上是相同的.

关键词　图象处理，离散余弦变换，快速算法

1 引　言

离散余弦变换（DCT）广泛用于数字信号处理，特别是语音和图象数据压缩[1,2]. 因此不断有新的 DCT 的快速算法被提出[1,2]. 二维 DCT 的常规方法是行-列法. 对于计算 $N\times N$ DCT，需要计算 $2N$ 个一维 DCT. 为了更有效地计算二维 DCT，提出了直接作用于二维数据集的算法[3-5]. 迄今为止，现有文献中最有效的二维 DCT 算法是由 Duhamel[3] 提出的直接多项式方法. 该算法所需乘法次数减少到常规算法的50%. 另一方面，文献[4]提出的算法要求的乘法运算量是常规算法乘法运算量的75%. 由 Vetterbi[5] 提出的利用多项式变换 FFT 和旋转的间接方法要求的乘法运算量是常规算法的50%到75%之间. 采用递归二维 DCT 技术计算大尺寸的二维 DCT 非常有效[4]. 在 4×4 DCT 情况，该算法要求的乘法次数与文献[3,5]的算法相同. 然而对于更大尺寸的变换，该算法的乘法运算量要比文献[3,5]提出的算法多.

本文介绍了一种新的快速二维 DCT 算法. 该算法对于计算 $N\times N$ DCT（这里 $N=2^m$）只需计算 N 个一维 DCT（这里 $N=2^m$）. 所以所需乘法次数只是常规算法的一半. 这与文献[3]的算法一样. 但与文献[3]的算法相比，本文算法的计算结构具有高度规则性，并只要求实数运算.

2 算法原理

2.1 快速二维 DCT 算法

对于给定的二维数据序列 $\{x_{ij}; i,j=0,1,\cdots,N-1\}$，其二维 DCT 序列 $\{Y_{mn}$;

1993-05-08 收到，1994-04-13 定稿
卢　颉　男，1970年生，助理工程师，现从事多媒体技术开发，现在四川鼎天微电子有限公司工作，邮政编码 610054.
王新成　男，1965年生，博士后，现从事三维计算机视觉和体成象技术研究.
朱雅乐　男，1942年生，教授，现从事图文电视和 HDTV 研究，现在电子科技大学电子工程系工作，邮政编码 610054.

$m,n=0,1,\cdots,N-1\}$ 由下式给定.

$$Y_{mn}=\frac{4}{N^2}u(m)u(n)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x_{ij}\cos\frac{(2i+1)m}{2N}\pi\cos\frac{(2j+1)n}{2N}\pi$$

利用三角函数的积化和差公式,定义新的变换 A_{mn} 和 B_{mn} 如下式所示

$$\left.\begin{aligned}A_{mn}&=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x_{ij}\cos\frac{(2i+1)m+(2j+1)n}{2N}\pi,\\B_{mn}&=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x_{ij}\cos\frac{(2i+1m)-(2j+1)n}{2N}\pi.\end{aligned}\right\}\qquad(1)$$

这样, Y_{mn} 的归一化形式 y_{mn} 为: $y_{mn}=(A_{mn}+B_{mn})/2$

(1)式中的变换核等效于一维 DCT 变换核的条件是把 $\{(2i+1)m\pm(2j+1)n\}$ 表示为 $(2i+1)$ 的某个整数倍. 要满足这一条件, $(2j+1)$ 就应为 $(2i+1)$ 的倍数模 $2N$ 的同余数或者是 $2N$ 减去 $(2i+1)$ 的倍数模 $2N$ 的同余数. 即

$$(2j+1)[p(2i+1)]\bmod 2N \text{ 或 } (2j+1)=2N-[p(2i+1)]\bmod 2N,$$

式中 p 是从 1 到 $N-1$ 范围内的奇数. 因为超出这一范围的 p 值将产生重复的 j 值, 于是上两式又可等效为

$$j=[pi+(p-1)/2]\bmod N,\qquad(2a)$$

$$j=N-1-[pi+(p-1)/2]\bmod N,\qquad(2b)$$

$$p=1,3,\cdots,N-1.$$

可以证明,当 i 处在 0 到 $N-1$ 的范围内时,由上式所获得的 j 的 N 个序列是互不相同的. 因此,二维输入数据集可被划分为 N 个不同的数据集,它的每一个下标都满足上式所示的关系. 那么,我们可以看出这些数据集中的每一个变换核都等效于一维 DCT 的变换核. 为了区分当 $p=1,3,5,\cdots,N-1$ 时由上式所获得的 j 序列中的每一个, 我们将它表示为

$$j(p;a)=[pi+(p-1)/2]\bmod N$$

或

$$j(p;b)=N-1-[pi+(p-1)/2]\bmod N,$$

$$p=1,3,\cdots,N-1;\quad i=0,1,2,\cdots,N-1,$$

即对于给定的 $p,\{j(p;a):i=0,1,\cdots,N-1\}$ 是从 (2a) 式获得的 j 的序列,而 $\{j(p;b):i=0,1,\cdots,N-1\}$ 是由 (2b) 式获得的 j 的序列. 因此, 通过把二维输入序列 $\{x_{ij};i,j=0,1,2,\cdots,N-1\}$ 划分为 N 个一维序列 $\{x_{ij(p;a)}:i=0,1,2,\cdots,N-1\}$ 和 $\{x_{ij(p;b)}:i=0,1,2,\cdots N-1\}$, $p=1,3,5,\cdots,N-1$. 这样,(1)式的一维变换就可表示为一维 DCT 之和. 将这些一维数据序列分别表示为 R_p^a 和 R_p^b, 即

$$R_p^a=\{x_{ij(p;a)}:i=0,1,2,\cdots,N-1;\ j(p;a)=[pi+(p-1)/2]\bmod N\},\qquad(3a)$$

$$R_p^b=\{x_{ij(p;b)}:i=0,1,2,\cdots,N-1;j(p;b)=N-1-[pi+(p-1)/2]\bmod N\},\qquad(3b)$$

$$p=1,3,5,\cdots,N-1.$$

通过引入新的整数序列 q_{pi}, 即 $q_{pi}=[pi-(p-1)/2]\bmod N$, 可以把 (3a) 式和 (3b) 式改写为不用模的运算. 利用数据分组,可以把 A_{mn} 和 B_{mn} 表示为

$$A_{mn}=\sum_{\substack{p=1\\(p:\mathrm{odd})}}^{N-1}\{T_p^a(m,n)+T_p^b(m,n)\},$$

$$B_{mn}=\sum_{\substack{p=1\\(p:\mathrm{odd})}}^{N-1}\{S_p^a(m,n)+S_p^b(m,n)\}.$$

然后，将 n 区分为偶数和奇数两种情况来讨论，即

$$T_p^a(m,n)=\begin{cases}\sum_{i=0}^{N-1}x_{ij(p:a)}\cos\frac{(2i+1)(m+np)}{2N}\pi, & n\ \text{为偶数时；} \quad (4a)\\ \sum_{i=0}^{N-1}(-1)^{q_{pi}}x_{ij(p:b)}\cos\frac{(2i+1)(m+np)}{2N}\pi, & n\ \text{为奇数时。} \quad (4b)\end{cases}$$

同理可得 $T_p^b(m,n),S_p^a(m,n),S_p^b(m,n)$ 的相应表示，则 y_{mn} 可表示为

$$y_{mn}=\begin{cases}\frac{1}{2}\sum_{\substack{p=1\\(p:\mathrm{odd})}}^{N-1}\left[\sum_{i=0}^{N-1}(x_{ij(p:a)}+x_{ij(p:b)})\cos\frac{(2i+1)(m+np)}{2N}\pi\right.\\ \left.+\sum_{i=0}^{N-1}(x_{ij(p:a)}+x_{ij(p:b)})\cos\frac{(2i+1)(m-np)}{2N}\pi\right], \quad n\ \text{为偶数时；}\\ \frac{1}{2}\sum_{\substack{p=1\\(p:\mathrm{odd})}}^{N-1}\left[\sum_{i=0}^{N-1}(-1)^{q_{pi}}(x_{ij(p:a)}-x_{ij(p:b)})\cos\frac{(2i+1)(m+np)}{2N}\pi\right.\\ \left.+\sum_{i=0}^{N-1}(-1)^{q_{pi}}(x_{ij(p:a)}-x_{ij(p:b)})\cos\frac{(2i+1)(m-np)}{2N}\pi\right],\\ \qquad n\ \text{为奇数时。}\end{cases}$$

可以看出，

$$\sum_{i=0}^{N-1}(x_{ij(p:a)}+x_{ij(p:b)})\cos\frac{(2i+1)(m+np)}{2N}\pi \tag{5a}$$

和

$$\sum_{i=0}^{N-1}(x_{ij(p:a)}+x_{ij(p:b)})\cos\frac{(2i+1)(m-np)}{2N}\pi \tag{5b}$$

相应于数据序列 $\{x_{ij(p:a)}+x_{ij(p:b)}\}$ 的一维 DCT 之一，该数据序列依赖于 m 和 n。即是，通过定义

$$f_{pl}=\sum_{i=0}^{N-1}(x_{ij(p:a)}+x_{ij(p:b)})\cos\frac{(2i+1)l}{2N}\pi, \tag{6}$$

可以看出，(5a) 式和 (5b) 式相应于 f_{pl} 或 $-f_{pl}$ 中的一个。这里，$l=0,1,\cdots,N-1$。

通过定义

$$g_{pl}=\sum_{i=0}^{N-1}(-1)^{q_{pi}}(x_{ij(p:a)}-x_{ij(p:b)})\cos\frac{(2i+1)l}{2N}\pi. \tag{7}$$

可以看出，

$$\sum_{i=0}^{N-1}(-1)^{q_{pi}}(x_{ij(p;a)}-x_{ij(p;b)})\cos\frac{(2i+1)(m+np)}{2N}\pi$$

和

$$\sum_{i=0}^{N-1}(-1)^{q_{pi}}(x_{ij(p;a)}-x_{ij(p;b)})\cos\frac{(2i+1)(m-np)}{2N}\pi$$

相应于 $\pm g_{pl}$ 中的一个．这里，$l=0,1,2,\cdots,N-1$．因此，为了计算 $N\times N$ DCT 序列 $\{y_{mn}:m,n=0,1,2,\cdots,N-1\}$，我们只需要 $\{f_{pl}: l=0,1,2,\cdots,N-1\}$ 和 $\{g_{pl}:l=0,1,2,\cdots,N-1\},p=1,3,5,\cdots,N-1$．这就意味着计算 $N\times N$ DCT 只需计算N个一维 DCT．

2.2 快度二维逆离散余弦变换（IDCT）算法

在进行 DCT 逆变换时，也就是其正变换过程的逆转．首先，我们需要将输入 Y_{mn} 进行归一化，亦即

$$y_{mn}=\frac{1}{2}[Y_{mn}\cdot u(m)\cdot u(n)],\ u(l)=\begin{cases}\sqrt{2}, & l=0;\\ 1, & l\neq 0,\end{cases} \tag{8}$$

再将 n 分为偶数和奇数两种情况，由 y_{mn} 分别求出 f_{pl} 和 g_{pl}．其具体过程是通过：$m=l\pm pn$，$n=0,1,2,\cdots,N-1$，求得 m；再取此m的绝对值，对 $4N$ 求模，接下来与正变换类似地分六种情况讨论

(1) $0\leqslant m<N$ 时，$m=n,y_{mn}=y_{mn}$;

(2) $m=N$ 时，$y_{mn}=0$;

(3) $N<m\leqslant 2N$ 时，$m=2N-m,y_{mn}=-y_{mn}$;

(4) $2N<m<3N$ 时，$m=4N-m,y_{mn}=-y_{mn}$;

(5) $m=3N$ 时，$y_{mn}=0$;

(6) $3N<m<4N$ 时，$m=4N-m,y_{mn}=y_{mn}$.

与正变换不同的是，不能把上面求得的若干个 y_{mn} 单纯相加．当求 f_{pl} 时，$n=0$，则m只取 $l+pn$，不取 $l-pn$；求 g_{pl} 时，$l=0$，则m只取 $l+pn$，不取 $l-pn$．得到 f_{pl} 和 g_{pl} 后，通过一维 IDCT 就可求得N组一维输出．这些还不是 IDCT 的结果，还需要重新排序．如果令这些一维 IDCT 输出为 R_{mn}，$n=0,1,2,\cdots,N-1$，则 $i=n$，$j=[(m+1)n+i/2]\mathrm{mod}N$（当$m$为偶数时）；$j=N-[m\cdot n+(i+1)/2]\mathrm{mod}N$（当$m$为奇数时）．这样求出的 x_{ij} 就是二维 IDCT 的输出结果．上面是求 IDCT 的一般公式．在对于定点的 IDCT 运算时，还可以有更快的算法．在正交变换情况下，如果不考虑变换中间变量的数值，那么正向 DCT 的信号流程图反转过来，就可以获得归一化 IDCT 的信号流程图．只是在系数上还需要作些修改．

3 与其他快速算法的比较

将本算法所需乘法和加法运算次数分别记作M和 A．因为计算 $N\times N$ DCT 中只需要N个一维的 DCT，因此，在实现一维 DCT 时，可以用任意一个现存的一维 DCT 算法．在用文献[1]或文献[2]的一维 DCT 算法时，对于N点一维 DCT，需要的乘法次数

为 $(N/2)\log_2 N$，需要的加法次数为 $(3N/2)\log_2 N - N + 1$. 那么，$N \times N$ DCT 需要的乘法次数为

$$M = (N^2/2)\log_2 N \tag{9a}$$

这里需要指出的是，需要对 $(1 + \log_2 N)$ 个蝶形级和对一维 DCT 级的加法进行计算. 然而，在最后一级，不需要对 y_{0j} 和 $y_{(N/2)(N/2)}$ 进行加法运算。

再有，除了 f_{10} 和 f_{30} 外，g_{p0} 和 f_{p0} 不需要蝶形对. 那么，除了对一维 DCT 级所需要的加法运算外，所需的加法次数是: $N^2(1 + \log_2 N) - 2N - (N - 2)$，因此，$N\times N$ DCT 所需的加法运算次数为

$$\begin{aligned} A &= N^2(1 + \log_2 N) - 2N - (N - 2) + N[(3N/2)\log_2 N - N + 1] \\ &= (5N^2/2)\log_2 N - 2N + 2 \end{aligned} \tag{9b}$$

表 1 乘法和加法次数比较

		常规算法	文献[4]	文献[5]	文献[3]	本文算法
乘法	4×4	32	24	16	16	16
	8×8	192	144	104	96	96
	16×16	1024	768	568	512	512
加法	4×4	72	72	70	68	74
	8×8	464	464	462	484	466
	16×16	2592	2592	2558	2531	2530

图 1 对 GIRL 图象进行 DCT 及 IDCT 变换

(a) 原图象 (b) 4×4 DCT 图象 (c) 8×8 DCT 图象 (d) 16×16 DCT 图象 (e) 16×16 IDCT 还原图象

这样，根据式（9a）和式（9b），我们可以把本算法所需乘法和加法运算次数与文献[3—5]以及常规算法进行比较。表 1 示出了比较结果。

从表 1 可以看出，本算法的乘法次数最少，只有常规算法的一半；加法次数与其它算法相当。同文献[3]相比，其运算次数几乎相同。但文献[3]的算法需要进行复数运算，而本文算法只需进行实数运算，这就大大减少了运算的复杂性。此外，通过对正变换和逆变换的结果归一化后显示出来的图象可以看出，图象还原误差很小。对一幅 256 × 256 的 girl 图象进行 16 × 16 DCT 和 IDCT 变换，在 IBM PC386 微机（主频 33MHz，带 80387 协处理器）上，只需 8s。图 1 显示了这一结果。

应用本算法，可以达到既减少了运算量又具有较好的推导算法和较简单的运算结构。因此，本算法在科学计算和工程应用中都将是有效的。

参　考　文　献

[1] Lee B C. IEEE Trans. on ASSP, 1984, ASSP-32(12): 1243—1245.

[2] Yang Jar-Ferr, Shaih Shih-Chang, Bai Bor-Long. IEEE Trans. on CE, 1993, CE-39(4): 934—940.

[3] Duhamel P, Guillemot C. Polynomial Transform Computation of 2-D DCT, in Proc. ICASSP'90, New York, USA: 1990, 1515—1518.

[4] Chau L-P, Sui W-C. Electron. Lett., 1994, 30(3): 197—198.

[5] Lee M H, Crebbin G. IEE Proc.-F, Radar Image and Signal Processing, 1994, 141(1): 39—48.

A NEW FAST ALGORITHM OF 2-D DISCRETE COSINE TRANSFORM

Lu Jie　　Wang Xincheng　　Zhu Weile

(Department of Electronic Engineering, Tsinghua University, Beijing 100084)

Abstract　A new algorithm for the fast computation of a 2-D discrete cosine transform (DCT) is presented. It is shown that the $N \times N$ DCT, where $N = 2^n$, can be computed using only N 1-D DCT's and additions, instead of using $2N$ 1-D DCT's, as in the conventional row-column approach. Hence the total number of multiplications for the proposed algorithm is only half of that required for the row-column approach, and is also less than that of most of other fast algorithms, while the number of additions is almost comparable to that of others.

Key words　Image processing, Discrete cosine transform, Fast algorithm