2003년도 대한전자공학회 하계종합학술대회 제26권 제1호

복소연산이 없는 Polynomial 변환을 이용한 고속 2 차원 DCT

최환석, 김원하*
명지대학교, 정보통신공학부
전화: 031-330-6346, *011-9910-6175
전자우편: andxor@mju.ac.kr, wonha@mju.ac.kr

Fast two dimensional DCT by Polynomial Transform without complex operations

Hwan-Serk Choi, Won-Ha Kim*
Electronic, Information & Communication Eng, Myongji University
Email: andxor@mju.ac.kr. wonha@mju.ac.kr
Tel: 031-330-6346, *011-9910-6175, Fax : 031-321-0271

Abstract

본 논문은 Polynomial 변환을 이용하여 2차원 Discrete Cosine Transform (2D-DCT)의 계산을 1차원 DCT로 변환하여 계산하는 알고리즘을 개발한다. 기존의 일반적인 알고리즘인 row-column이 $N \times M$의 2D-DCT에서 $3/2NM\log_2(NM)-2NM+N+M$의 합과 $1/2NMlog_2(NM)$의 곱셈이 필요한데 비하여 본 논문에서 제시한 알고리즘은 $3/2\,NM\log_2 M\ +NM\log_2 N$ - $M-N/2+2$의 합과 $1/2NM\log_2 M$의 곱셈 수를 필요로 한다. 기존의 polynomial 변환에 의한 2D DCT는 Euler 공식을 적용하였기 때문에 복소 연산이 필요하지만 본 논문에서 제시한 polynomial 변환은 DCT의 modular 규칙을 이용하여 2D DCT를 1D DCT의 합으로 직접 변환하므로 복소 연산이 필요하지 않다. 또한 본 논문에서 제시한 알고리즘은 각 차원에서 데이터 크기가 다른 임의 크기의 2차원 데이터 변환에도 적용할 수 있다.

I. Introduction

The discrete cosine transform(DCT) has a wide range of applications such as image coding, feature, extraction, and so on. Since DCT produces excellent spatial frequency of images, most of applications adopting DCT are on image-related process. Thus, the dimension of DCT must be 2 in most of DCT applications. Algorithms for fast calculating DCT as well as for reducing computational complexity of two-dimensional calculations are necessary. The polynomial transform (PT) were reported to require the lowest operations of two-dimensional DCT. Compared with the row-column method, Polynomial transforms (PT) require only one half of the number of multiplications and a smaller number of additions. Therefore, the polynomial transform have the greatest potential toward lowest arithmetic complexity, but very little was done on the best way of implementing polynomial transforms. In this paper, we present the polynomial transform algorithm that can be easily implemented.

II. Computation of 2D-DCT-II

The 2D-DCT-II of the input sequences $x(n,m)$ is defined by

$$X(k,l)=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}x(n,m)\cos\frac{\pi(2n+1)k}{2N}\cos\frac{\pi(2m+1)l}{2M}$$
$$k=0,1,\cdots,N-1;\quad l=0,1,\cdots,M-1 \tag{1}$$

The constant scaling factors in the DCT definition are ignored for simplicity. We assume that M and N are powers of 2 and $M \geq N$. We can write $N=2^t$ and $M=2^J N$, where $t>0$ and $J \geq 0$, respectively.

The input sequences $x(n,m)$ can be decomposed into $y(n,m)$ likewise. So,

* 이 논문은 2001년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2001-003-E00158).

$$\begin{aligned} y(n,m) &= x(2n,2m) \\ y(N-1-n,m) &= x(2n+1,2m) \\ y(n,M-1-m) &= x(2n,2m+1) \\ y(N-1-n,M-1-m) &= x(2n+1,2m+1) \\ n &= 0,1,\cdots,N/2-1 \\ m &= 0,1,\cdots,M/2-1 \end{aligned} \qquad (2)$$

,and

$$X(k,l)=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} y(n,m)\cos\frac{\pi(4n+1)k}{2N}\cos\frac{\pi(4m+1)l}{2M}$$
$$k=0,1,\cdots,N-1;\ l=0,1,\cdots,M-1 \qquad (3)$$

Using the trigonometric formulae, (3) can be computed by $X(k,l)=\frac{1}{2}[A(k,l)+B(k,l)]$, where

$$A(k,l)=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} y(n,m)\cos\left[\frac{\pi(4n+1)k}{2N}+\frac{\pi(4m+1)l}{2M}\right]$$
$$k=0,1,\cdots,N-1;\ l=0,1,\cdots,M-1 \qquad (4)$$

and

$$B(k,l)=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} y(n,m)\cos\left[\frac{\pi(4n+1)k}{2N}-\frac{\pi(4m+1)l}{2M}\right]$$
$$k=0,1,\cdots,N-1;\ l=0,1,\cdots,M-1. \qquad (5)$$

In order to simplify (4) and (5), we define $p(m)$ as . $p(m)=[(4p+1)m+p]\bmod N$, and then

$$4p(m)+1\equiv(4m+1)(4p+1) \bmod 4N$$
$$where\ p=0,1,\cdots,N-1;\ m=0,1,\cdots,M-1.$$

By plugging $p(m)$ into (4) and (5), we obtain

$$\begin{aligned} A(k,l) &= \sum_{p=0}^{N-1}\sum_{m=0}^{M-1} y(p(m),m)\cos\left[\frac{\pi(4p(m)+1)k}{2N}+\frac{\pi(4m+1)l}{2M}\right] \\ &= \sum_{p=0}^{N-1}\sum_{m=0}^{M-1} y(p(m),m)\cos\left[\frac{\pi(4m+1)(4p+1)k}{2N}+\frac{\pi(4m+1)l}{2M}\right] \\ &= \sum_{p=0}^{N-1}\sum_{m=0}^{M-1} y(p(m),m)\cos\left[\frac{\pi(4m+1)(2^J(4p+1)k+l)}{2M}\right]. \end{aligned} \qquad (6)$$

Also,

$$\begin{aligned} B(k,l) &= \sum_{p=0}^{N-1}\sum_{m=0}^{M-1} y(p(m),m)\cos\left[\frac{\pi(4p(m)+1)k}{2N}-\frac{\pi(4m+1)l}{2M}\right] \\ &= \sum_{p=0}^{N-1}\sum_{m=0}^{M-1} y(p(m),m)\cos\left[\frac{\pi(4m+1)(2^J(4p+1)k-l)}{2M}\right]. \end{aligned} \qquad (7)$$

$A(k,l)$ and $B(k,l)$ can be expressed via their common factor of cosine, which is defined by

$$V_p(j)=\sum_{m=0}^{M-1} y(p(m),m)\cos\left[\frac{\pi(4m+1)j}{2M}\right]$$
$$p=0,1,\cdots,N-1;\qquad j=0,1,\cdots,M-1 \qquad (8)$$

Then,

$$A(k,l)=\sum_{p=0}^{N-1} V_p[2^J(4p+1)k+l] \qquad (9)$$

and

$$B(k,l)=\sum_{p=0}^{N-1} V_p[2^J(4p+1)k-l] \qquad (10)$$

$$k=0,1,\cdots,N-1;\qquad l=0,1,\cdots,M-1$$

III. Polynomial Representation of 2D-DCT-II

$A(k,l)$ and $B(k,l)$ are expressed by $V_p(j)$, therefore $V_p(j)$'s properties are $A(k,l)$'s and $B(k,l)$'s. We define

$$\begin{cases} \hat{y}_p(2m)=y(p(m),m) \\ \hat{y}_p(2m+1)=y(p(M-1-m),M-1-m) \end{cases}$$
$$where\ m=0,1,\cdots,M/2-1$$

Therefore, (8) can be expressed into one dimensional DCT-II

$$V_p(j)=\sum_{m=0}^{M-1}\hat{y}_p(m)\cos\left[\frac{\pi(2m+1)j}{2M}\right]$$
$$where\ p=0,1,\cdots,N-1; j=0,1,\cdots,M-1 \qquad (8')$$

This means that 2D-DCT-II can be computed by the sum of 1D-DCT-II, $V_p(j)$.

The other properties of $V_p(j)$ are

$V_p(j+u2M)=(-1)^u V_p(j)$,

$V_p(2M-j)=-V_p(j)$, and

$V_p(M)=0,\quad where\ j=0,1,\cdots,M-1$.

These properties of $V_p(j)$'s period are interpreted by the 'mod' operation, because these properties of the periodicity are similar to the 'mod' operation in polynomial which divider is $z^{2M}+1$.

Based on the properties of $V_p(j)$, it can be proved that

$$\begin{aligned} A(k,2M-l) &= \sum_{p=0}^{N-1} V_p[2^J(4p+1)k+2M-l] \\ &= -\sum_{p=0}^{N-1} V_p[2^J(4p+1)k-l] \\ &= -B(k,l) \end{aligned} \qquad (11)$$

and,

$$A(k,0)=B(k,0),\qquad A(0,l)=B(0,l). \qquad (12)$$

Now, define a generating polynomial such as

$$B_k(z)=\sum_{l=0}^{M-1}B(k,l)z^l-\sum_{l=M}^{2M-1}A(k,2M-l)z^l \qquad (13)$$

By substituting (10) and (11) in (13), we have

$$B_k(z)=\sum_{l=0}^{2M-1}\sum_{p=0}^{N-1}V_p(2^J(4p+1)k-l)z^l$$

from which $A(k,l)$ and $B(k,l)$ can be derived. In order to use the polynomial transform, the $\bmod(z^{2M}+1)$ operation for $B_k(z)$ is expressed by

$$\begin{aligned} B_k(z) &\equiv \sum_{p=0}^{N-1}\sum_{l=0}^{2M-1} V_p(2^J(4p+1)k-l)z^l \bmod z^{2M}+1 \\ &\equiv \sum_{p=0}^{N-1}\sum_{l=0}^{2M-1} V_p(l-2^J(4p+1)k)z^l \bmod z^{2M}+1 \\ &\equiv \sum_{p=0}^{N-1}\sum_{l=0}^{2M-1} V_p(l)z^{\,l+2^J(4p+1)k} \bmod z^{2M}+1 \\ &\equiv \left[\sum_{p=0}^{N-1} U_p(z)\ \hat{z}^{pk}\right] z^{\,2^Jk} \bmod z^{2M}+1 \\ &\equiv C_k(z) z^{\,2^Jk} \bmod z^{2M}+1 \end{aligned} \tag{14}$$

where

$$U_p(z) \equiv \sum_{j=0}^{2M-1} V_p(j) z^j \bmod z^{2M}+1$$

$$C_k(z) \equiv \sum_{p=0}^{N-1} U_p(j)\hat{z}^{pk} \bmod z^{2M}+1 \tag{15}$$

$k=0,1,\cdots,N-1,\ \hat{z} \equiv z^{2^{J+2}} \bmod z^{2M}+1.$

We also generate $A_k(z)$ such as

$$A_k(z) = \sum_{l=0}^{2M-1}\sum_{p=0}^{N-1} V_p(2^J(4p+1)k+l)z^l \qquad k=0,1,\cdots,N-1 \tag{16}$$

which will be used in the Inverse-2D-DCT-Ⅱ.

In addition, $X_k(z)$ is defined as

$$\begin{aligned} X_k(z) &= \sum_{l=0}^{2M-1} X(k,l)z^l \\ &\equiv \frac{1}{2}(B_k(z)+A_k(z)) \bmod z^{2M}+1 \end{aligned} \tag{17}$$

where $k=0,1,\cdots,N-1.$

Ⅳ. Fast Polynomial Transform

$U_p(z)$ and $C_k(z)$ are polynomial transforms that can be computed by a fast algorithm. The coefficients of $U_p(z)$ are $V_p(j)$, which have the symmetric property,

$$V_p(2M-j) = -V_p(j),\ \ V_p(M)=0$$

It indicates that the only half of the coefficients are needed to express $U_p(z)$. This property can be expressed as

$$U_p(z) \equiv U_p(z^{-1}) \bmod z^{2M}+1. \tag{18}$$

$$U_p(z^{-1}) \equiv \sum_{j=0}^{2M-1} V_p(j) z^{-j} \bmod z^{2M}+1$$

Based on this property, it can be proved that the polynomial sequence $C_k(z)$ also has a symmetric property expressed as

$$C_{N-k}(z) \equiv C_k(z^{-1}) \bmod z^{2M}+1 \tag{19}$$

which indicates that about one half of the coefficients $C_k(z)$ are necessarily computed.

We define

$$C^0_{n_0n_1\cdots n_{t-1}}(z) = U_{n_{t-1}n_{t-2}\cdots n_0}(z)$$

$$\begin{aligned} C^j_{n_0\cdots n_{t-j-1}k_{j-1}\cdots k_0}(z) &\equiv \sum_{n_{t-j}=0}^{1}\cdots\sum_{n_{t-1}=0}^{1} U_{n_{t-1}n_{t-2}\cdots n_0}(z) \\ &\cdot \hat{z}^{\,2^{t-1}n_{t-1}k_0+2^{t-2}n_{t-2}(2k_1+k_0)+\cdots+2^{t-j}n_{t-j}(2^{j-1}k_{j-1}+\cdots+k_0)} \\ &\bmod z^{2M}+1, \qquad \textit{where } j=1,2,\cdots,t. \end{aligned} \tag{20}$$

And, for $k_{j-1}=0$ *or* 1, (20) can be expressed as

$$C^j_{n_0\cdots n_{t-j-1}0k_{j-2}\cdots k_0}(z) \equiv C^{j-1}_{n_0\cdots n_{t-j-1}0k_{j-2}\cdots k_0}(z) + C^{j-1}_{n_0\cdots n_{t-j-1}1k_{j-2}\cdots k_0}(z)\hat{z}^q \bmod z^{2M}+1 \tag{21}$$

$$C^j_{n_0\cdots n_{t-j-1}1k_{j-2}\cdots k_0}(z) \equiv C^{j-1}_{n_0\cdots n_{t-j-1}0k_{j-2}\cdots k_0}(z) - C^{j-1}_{n_0\cdots n_{t-j-1}1k_{j-2}\cdots k_0}(z)\hat{z}^q \bmod z^{2M}+1 \tag{22}$$

where $j=1,2,\cdots,t,\ \ q=\sum_{l=0}^{j-2} k_l 2^{t-j+l}$.

Equation (21) and (22) are the computational steps of the fast polynomial transform algorithm. The temporary outputs of each stage, which index is j, are $C^j_{n_0\cdots n_{t-j-1}k_{j-1}\cdots k_0}(z)$ and each output has an important symmetric property of (19). If we replace the binary expression of the subscripts $(0k_{j-2}\cdots k_0)$ by $\bar{k}$ and $(n_0\cdots n_{t-j-1})$ by $\bar{n}$ for simplicity, (21) and (22) become

$$C^j_{\bar{n}2^j+\bar{k}}(z) \equiv C^{j-1}_{\bar{n}2^j+\bar{k}}(z) + C^{j-1}_{\bar{n}2^j+\bar{k}+2^{j-1}}(z)\ \hat{z}^q \bmod z^{2M}+1 \tag{21'}$$

$$C^j_{\bar{n}2^j+\bar{k}+2^{j-1}}(z) \equiv C^{j-1}_{\bar{n}2^j+\bar{k}}(z) - C^{j-1}_{\bar{n}2^j+\bar{k}+2^{j-1}}(z)\ \hat{z}^q \bmod z^{2M}+1 \tag{22'}$$

$\bar{k}=0,1,\cdots,2^{j-1}-1;\ \ \bar{n}=0,1,\cdots,2^{j-1}-1.$

Therefore, the output polynomials of each stage have the following properties, which leads to a reduction of the computational complexity for (21′) and (22′) by one half as shown in (19).

$$\begin{aligned} C^j_{\bar{n}2^j+2^j-\bar{k}}(z^{-1}) &\equiv C^j_{\bar{n}2^j+\bar{k}}(z) \bmod z^{2M}+1 \\ C^j_{\bar{n}2^j}(z^{-1}) &\equiv C^j_{\bar{n}2^j}(z) \bmod z^{2M}+1 \\ C^j_{\bar{n}2^j+2^{j-1}}(z^{-1}) &\equiv C^j_{\bar{n}2^j+2^{j-1}}(z) \bmod z^{2M}+1 \\ \bar{k} &= 0,1,\cdots,2^{t-1}-1 \end{aligned}$$

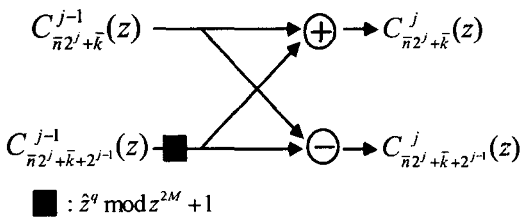

Fig.1. Butterfly operation

These properties enable us to obtain $C^{\bar{i}}_{\bar{n}2^{j}+\bar{k}}(z), 2^{j-1}+1 \le \bar{k} \le 2^{j}-1$, from $C^{\bar{i}}_{\bar{n}2^{j}+\bar{k}}(z), 1 \le \bar{k} \le 2^{j-1}$. Therefore, (21) and (22) need NM additions only. It is noted from (13) that the Mth coefficient of each output polynomial is $B(k, M)$, which is not required by our computation. Therefore, a further saving of $N/2-1$ additions is available. In total, the number of additions required by (15) for the fast polynomial transform is $NM \log_2 N - N/2 + 1$.

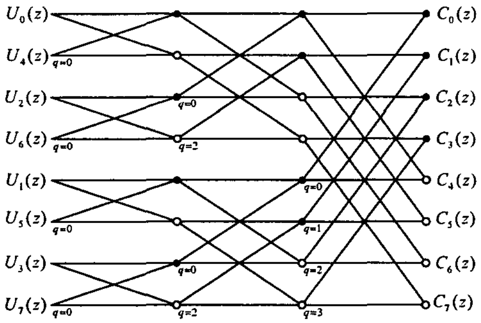

Fig. 2. where N=8, the flow graph of polynomial transform algorithm. Briefly, Fig. 1. is represented by .

Therefore, if it is assumed that our algorithm uses the most computationally efficient fast algorithm for 1D-DCT-II of length M, the total number of additions for 2D-DCT-II are $3/2\, NM \log_2 M + NM \log_2 N - M - N/2 + 2$ and the total number of multiplications are $1/2 NM \log_2 M$.

size	R.C.	Block factors	P.T
4×4	α=72 μ=32	α=70 μ=16	α=76 μ=16
8×8	α=464 μ=192	α=462 μ=104	α=470 μ=96
16×16	α=2592 μ=1024	α=2592 μ=768	α=2538 μ=512
32×32	α=13376 μ=5120	α=13376 μ=3840	α=12754 μ=2560

Table 3. Comparison of computational complexity of 2D-DCT-II for an N×N block input. (α : the number of additions, μ : the number of multiplications)

V. Conclusions

Based on a method of directly using the polynomial transform in the computation of the 2D-DCT, this paper presents a fast algorithm for 2D-DCT-II. Symmetric properties of the polynomial are used to reduce the computational complexity and the computational structure is much more simplified compared with other reported algorithms. So, the simplified structure can make the implement of the 2D-DCT be easer than what it used to be.

References

[1] K. R. Rao and P. Yip. *Discrete Cosine Transform : Algorithms, Advantages and Applications.* New York: Academic, 1990

[2] B. G. Lee, *"A New algorithm to compute the discrete cosine transform." IEEE Trans. Acoust. Speech, Signal Processing,* vol. ASSP-32, pp.1243-1245, Apr. 1984.

[3] P. Duhamel and C. Guillemot, *"Polynomial transform computation of 2-D DCT," in Proc.* ICASSP '90, pp.1515-1518, Apr. 1990.

[4] M. Vetterli, *"Fast 2-D discrete cosine transform." in Proc.* ICASSP'85, Mar, 1985