

DCT 영역에서의 2차원 분리 대칭 선형 필터링

정회원 임 창 훈*

DCT Domain Separable Symmetric 2-D Linear Filtering

Changhoon Yim* *Regular Member*

요 약

본 논문에서는 DCT 영역에서의 효과적인 분리 대칭 선형 필터링 방법을 제안한다. 먼저 2차원 분리 선형 필터링을 DCT 영역에서의 1차원 필터링이 연결된 형태로 분해한다. 필터 계수가 대칭적일 때, DCT 영역의 필터링 행렬의 특별한 성질을 분석하였다. 이러한 성질을 이용하여 효과적인 DCT 영역에서의 2차원 필터링을 수행한다. 제안된 방법은 이전의 DCT 영역에서의 선형 필터링 방법에 비해서 일반적인 DCT 계수들이 가지고 있는 희소성을 고려할 때, 기존의 방법에 비하여 더 적은 수의 연산을 요구한다. 그리고 제안된 방식은 간단하고 규칙적인 연산 구조로 구성되어 있기 때문에, 효과적인 VLSI 구현에 적합하다.

Key Words : linear filtering; DCT domain filtering; compressed-domain processing, separable symmetric filtering.

ABSTRACT

We propose an efficient method for separable symmetric linear filtering in the DCT domain. First, separable 2-D linear filtering is decomposed into the cascade of 1-D filtering in the DCT domain. We investigate special characteristics of DCT domain filtering matrices when the filter coefficients are symmetric. Then we present the DCT domain 2-D filtering method using these characteristics. The proposed method requires smaller number of multiplications including typical sparseness of DCT coefficients compared to previous DCT domain linear filtering methods. Also, the proposed method is composed of simple and regular operations, which would be appropriate for efficient VLSI implementation.

I. 서론

JPEG, MPEG-2, MPEG-4, H.263, H.26L 등 많은 영상 (image)과 비디오 (video) 압축 (compression) 방식은 DCT (discrete cosine transform) 기술에 기반을 두고 있다. 압축 영역 (compressed domain)에서 영상과 비디오를 처리하게 되면, 더 적은 양의 데이터(data)를 처리하고 압축된 (compressed) 데이터와 비압축된 (uncompressed) 데이터 간의 변환 (conversion) 과정을 줄일 수 있으므로 계산량을 줄일 수 있다. 압축 영역 (compressed domain)에서의 처리 관점에서 영상과 비디오를

DCT 영역에서 직접 처리 하는 연구가 되어왔다^{[1]-[4]}.

필터링(filtering)은 신호, 영상, 비디오 처리에서 가장 중요한 연산 (operation) 중의 하나이다. 그리고 DCT 영역에서의 필터링에 관해서 많은 연구가 되어 왔다^{[2],[3],[5],[6]}. 공간 영역 (spatial domain)에서의 원형 콘벌루션 (circular convolution)은 이산 푸리에 변환 (discrete Fourier transform) 영역에서의 곱셈 (multiplication)에 해당된다는 것은 잘 알려져 있다. 이러한 콘벌루션 곱셈 성질 (convolution multiplication property: CMP)을 이용하여 DCT 영역에서 필터링을 하는 방법이 제안되었

* 삼성전자 디지털미디어연구소 IT Solution 팀 (ch.yim@samsung.com)
논문번호 : 020488-1111, 접수일자 : 2002년 11월 5일

다^[5]. 이러한 방식은 곱셈과 덧셈(addition) 수는 줄일 수 있으나, 근본적인 한계를 포함하고 있다. 공간 영역에서의 필터링은 원형 필터링이 아니고, 선형 (linear) 필터링이다. 참고문헌 [5]와 같이 원형 필터링 관계에 근거한 DCT 영역 필터링 방법은 블로킹 효과 (blocking artifacts)를 초래한다.

참고문헌 [6]에서는 선형 필터링을 이용한 변환 영역에서의 필터링을 제안하였다. 여기에서는 일반적인 의미에서의 1차원 필터링을 위한 행렬식을 표현하였다. 참고문헌 [2]에서는 DCT 영역에서의 2차원 필터링을 위한 근사적 (approximate) 선형 필터링 방법을 제안하였다. 여기에서는 복잡도 (complexity)를 줄이기 위하여 필터링을 위한 근사적 방법을 제시하였다. 그러나 이 방법으로는 근사적 컨벌루션 만을 얻을 수 있다. 참고문헌 [3]에서는 최근에 DCT와 DST(discrete sine transform)에 근거한 DCT 영역에서의 선형 필터링 방법을 제안하였다. 이 방법은 DCT 영역에서의 필터링을 위해서 DCT 계수와 아울러 DST 계수도 이용하였다. DST 계수는 미리 얻어질 수 없기 때문에, DCT 계수로부터 계산되어야 한다. 이와 같은 DCT/DST에 근거한 방법은 단지 곱셈 개수를 줄이는 데는 효과적이다. 그러나 이 방법은 cosine을 sine으로 변형하는데, sine을 cosine으로 변형하는데, 그리고 별도의 행렬을 뒤집는 (flipping) 연산을 위하여 매우 복잡한 신호의 흐름 (signal flow)을 필요로 하기 때문에, VLSI 구현이 용이하지 않다.

본 논문에서는 DCT 영역에서의 분리 (separable) 대칭 (symmetric) 선형 필터링을 위한 효과적인 방법을 제안한다. 이를 위하여 필터 계수가 대칭일 때, DCT 영역의 필터링 행렬의 특이한 성질을 분석하였다. 이러한 성질을 이용한 DCT 영역에서의 필터링 방법을 제안한다. DCT 영역의 입력 블록에서 흔히 나타나는 DCT 계수의 희소성 (sparseness)은 복잡도 감소를 위한 중요한 요소이다. 제안된 방법은 원형 필터링에 근거한 방법^[5]이나 근사적 필터링 방법^[2]에서와 같은 블로킹 효과가 없다. 제안된 방법은 입력 블록의 일반적인 희소성을 고려하면, 가장 적은 곱셈 수를 필요로 한다. 그리고 제안된 방법은 간단하고 규칙적인 연산을 필요로 하기 때문에 VLSI 구현 측면에서 적절한 방법이라고 할 수 있다.

본 논문의 구성은 다음과 같다. II장에서는 DCT 영역에서의 분리 대칭 필터링을 배경 설명으

로 기술한다. III장은 DCT 영역에서의 2차원 선형 필터링의 분해 (decomposition)을 보여준다. IV장에서는 필터 계수가 대칭적일 때의 DCT 영역의 필터링 행렬의 특별한 성질을 분석한다. V장에서는 이러한 성질을 이용하여 DCT 영역에서의 분리 대칭 선형 필터링을 위한 효과적인 방법을 제안한다. VI장에서는 DCT 계수의 희소성이 복잡도 감소에 기여하는 것을 기술한다. VII장에서는 기존의 방법과 복잡도를 비교한다. 마지막으로 VIII장에서 결론을 내린다.

II. DCT 영역에서의 분리 대칭 선형 필터링

$\{x(m, n)\}$ 과 $\{y(m, n)\}$ 은 각각 입력 영상과 필터링된 영상을 나타낸다. $\{f(k, l)\}$ 은 2차원 필터의 계수를 나타낸다. 2차원 필터 $\{f(k, l)\}$ 은 분리 (separable) 될 수 있다고 가정하는데, 그것은 $\{f(k, l)\}$ 이 어떤 1차원 필터 $\{v(k)\}$ 와 $\{h(l)\}$ 에 의하여 $f(k, l) = v_k h_l$ 과 같이 나타내어질 수 있는 것을 의미한다. 그리고 이러한 필터들은 대칭 (symmetric), 즉 $v_{-k} = v_k$, $h_{-l} = h_l$ 이 성립한다고 가정한다. 이러한 대칭 필터는 선형 위상 (linear phase) 특성을 위하여 대부분의 영상과 비디오 처리에 일반적으로 사용된다. 필터 $\{v(k)\}$ 와 $\{h(l)\}$ 의 구간을 각각 $-K \leq k \leq K$ 와 $-L \leq l \leq L$ 이라고 가정한다. 그리고 이전의 연구^{[2][3]}와 같이 K 와 L 은 8 이하인 것으로 가정한다.

$\{x(m, n)\}$ 과 $\{y(m, n)\}$ 이 겹쳐지지 않는 (non-overlapping) 8X8 행렬 $\mathbf{x}_{i,j}$ 와 $\mathbf{y}_{i,j}$ 로 구성되어 있다고 가정한다. 그리고 행렬 $\mathbf{v} \equiv [\mathbf{v}_{-1} \ \mathbf{v}_0 \ \mathbf{v}_1]$ 와 행렬 $\mathbf{h} \equiv [\mathbf{h}_{-1} \ \mathbf{h}_0 \ \mathbf{h}_1]$ 을 정의한다. 여기에서

$$\mathbf{v}_{-1} = \begin{bmatrix} v_8 & v_7 & \cdots & v_2 & v_1 \\ 0 & v_8 & \ddots & v_3 & v_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & v_8 & v_7 \\ 0 & 0 & \cdots & 0 & v_8 \end{bmatrix},$$

$$\mathbf{v}_0 = \begin{bmatrix} v_0 & v_1 & \cdots & v_6 & v_7 \\ v_1 & v_0 & \ddots & v_5 & v_6 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ v_6 & v_5 & \ddots & v_0 & v_1 \\ v_7 & v_6 & \cdots & v_1 & v_0 \end{bmatrix},$$

이고 $\mathbf{v}_1 = \mathbf{v}_{-1}^t$ 이다. \mathbf{h}_{-1} , \mathbf{h}_0 , \mathbf{h}_1 행렬은 \mathbf{v}_{-1} ,

$\mathbf{v}_0, \mathbf{v}_1$ 행렬에서 각각 v_k 를 h_l 로 대치하여 유사하게 정의된다. 2차원 분리 대칭 필터링은 블록에 근거한 행렬 연산으로 다음과 같이 나타내어질 수 있다^{[2][3]}.

$$y_{i,j} = \sum_{m=-1}^1 \sum_{n=-1}^1 v_m x_{i+m,j+n} h_n^t$$

여기에서 $y_{i,j}$, $x_{i+m,j+n}$, v_m , h_n 은 크기가 8X8인 행렬이다.

\mathbf{C} 를 DCT 변환행렬이라 하면, \mathbf{C} 는 $\mathbf{C}^{-1} = \mathbf{C}^t$ 인 성질을 가지고 있다. $\mathbf{X}_{m,n}$ 과 $\mathbf{Y}_{i,j}$ 를 $\mathbf{x}_{m,n}$ 과 $\mathbf{y}_{i,j}$ 의 DCT라고 정의한다. 즉, $\mathbf{X}_{m,n} = \mathbf{C}\mathbf{x}_{m,n}\mathbf{C}^t$, $\mathbf{Y}_{i,j} = \mathbf{C}\mathbf{y}_{i,j}\mathbf{C}^t$ 이다. $\mathbf{Y}_{i,j}$ 는 $\mathbf{X}_{i+m,j+n}$ 들에 관한 식으로 나타내어질 수 있다^[2].

$$\mathbf{Y}_{i,j} = \sum_{m=-1}^1 \sum_{n=-1}^1 \mathbf{V}_m \mathbf{X}_{i+m,j+n} \mathbf{H}_n^t \quad (1)$$

\mathbf{V}_m 과 \mathbf{H}_n 을 DCT 영역 필터링 행렬이라고 한다. 여기에서 DCT 영역 필터링 행렬 \mathbf{V}_m 과 \mathbf{H}_n 은 필터 계수 $\{v_k\}$ 와 $\{h_l\}$ 이 주어지면 미리 계산되어질 수 있다.

III. DCT 영역 2차원 선형 필터링의 분해

식 (1)의 DCT 영역 2차원 선형 필터링은 수직과 수평 방향으로 분리 수행될 수 있다. 식 (1)의 DCT 영역 2차원 선형 필터링은 다음과 같이 1차원 선형 필터링으로 분해 (decompose) 될 수 있다.

$$\mathbf{Z}_{i,j} = \mathbf{V}_{-1}\mathbf{X}_{i-1,j} + \mathbf{V}_0\mathbf{X}_{i,j} + \mathbf{V}_1\mathbf{X}_{i+1,j} \quad (2)$$

$$\mathbf{Y}_{i,j} = \mathbf{Z}_{i,j-1}\mathbf{H}_{-1}^t + \mathbf{Z}_{i,j}\mathbf{H}_0^t + \mathbf{Z}_{i,j+1}\mathbf{H}_1^t \quad (3)$$

여기에서 $\mathbf{Z}_{i,j}$ 는 DCT 영역의 입력 행렬 $\mathbf{X}_{i,j}$ 에 대하여 수직 방향으로 1차원 선형 필터링된 행렬을 나타낸다. 식 (2), (3)과 같은 DCT 영역의 2차원 선형 필터링은 하나의 8X8 DCT 블록에 대하여 $3 \cdot 8^3 + 3 \cdot 8^3 = 3072$ 번의 곱셈을 요구한다.

IV. DCT 영역 필터링 행렬의 특징

이 장에서는 필터 계수가 대칭일 때 DCT 영역 필터링 행렬의 특별한 성질에 대하여 분석한다. 이

러한 성질들은 다음 장에서 복잡도 감소를 위하여 사용된다. 8X8 DCT 영역 변환 행렬 \mathbf{C} 는 다음과 같이 나타내어질 수 있다^[7].

$$\mathbf{C} = \begin{bmatrix} s_0 & s_0 & s_0 & s_0 & s_0 & s_0 & s_0 & s_0 \\ s_1 & s_3 & s_4 & s_6 & -s_6 & -s_4 & -s_3 & -s_1 \\ s_2 & s_5 & -s_5 & -s_2 & -s_2 & -s_5 & s_5 & s_2 \\ s_3 & -s_6 & -s_1 & -s_4 & s_4 & s_1 & s_6 & -s_3 \\ s_0 & -s_0 & -s_0 & s_0 & s_0 & -s_0 & -s_0 & s_0 \\ s_4 & -s_1 & s_6 & s_3 & -s_3 & -s_6 & s_1 & -s_4 \\ s_5 & -s_2 & s_2 & -s_5 & -s_5 & s_2 & -s_2 & s_5 \\ s_6 & -s_4 & s_3 & -s_1 & s_1 & -s_3 & s_4 & -s_6 \end{bmatrix}$$

여기에서

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \cos \frac{\pi}{4} \\ \cos \frac{\pi}{16} \\ \cos \frac{\pi}{8} \\ \cos \frac{3\pi}{16} \\ \cos \frac{5\pi}{16} \\ \cos \frac{3\pi}{8} \\ \cos \frac{7\pi}{16} \end{bmatrix}$$

$c_{k,l}$ 을 8X8 DCT 변환 행렬 \mathbf{C} 의 (k,l) 번째 원소라 한다. 그러면 \mathbf{C} 의 짝수 행은 짝수 대칭 (even-symmetric)이고 \mathbf{C} 의 홀수 행은 홀수 대칭 (odd-symmetric)이다^[7]. 즉, $c_{k,l} = (-1)^k c_{k,7-l}$ 이 성립한다.

$p_{m,n}^-$ 과 $p_{m,n}^+$ 을 각각 \mathbf{v}_{-1} 과 \mathbf{v}_1 의 (m,n) 번째 원소라 정의한다. 여기에서 $m > n$ 일 때 $p_{m,n}^- = 0$ 이고, $m < n$ 일 때 $p_{m,n}^+ = 0$ 이다. $p_{m,n}^+$ 은 $p_{m,n}^-$ 에 관한 식으로 다음과 같이 나타내어질 수 있다.

$$p_{m,n}^+ = \begin{cases} p_{7-m,7-n}^-, & m \geq n \\ 0, & m < 0 \end{cases} \quad (4)$$

$a_{i,j}^-$ 와 $a_{i,j}^+$ 를 각각 DCT 영역 필터링 행렬 \mathbf{V}_{-1} 과 \mathbf{V}_1 의 (i,j) 번째 원소라 정의한다. 여기에서 $\mathbf{V}_{-1} = \mathbf{C}\mathbf{v}_{-1}\mathbf{C}^t$ 이기 때문에, $a_{i,j}^-$ 는 다음과 같이 $p_{m,n}^-$ 에 관한 식으로 나타내어질 수 있다.

$$a_{i,j}^- = \sum_{m=0}^7 \sum_{n=0}^7 c_{i,m} p_{m,n}^- c_{j,n} = \sum_{m=0}^7 \sum_{n=m}^7 c_{i,m} c_{j,n} p_{m,m}^- \quad (5)$$

이와 유사한 방법으로,

$$a_{i,j}^+ = \sum_{m=0}^7 \sum_{n=0}^7 c_{i,m} p_{m,n}^+ c_{j,n} = \sum_{m=0}^7 \sum_{n=0}^m c_{i,n} c_{j,n} p_{m,m}^+ \quad (6)$$

식 (4), (5), (6)에서, 다음 식이 성립하는 것을 알 수 있다.

$$a_{i,j}^+ = (-1)^{i+j} a_{i,j}^- \quad (7)$$

식 (7)에서 필터 계수가 대칭이면, DCT 영역 필터링 행렬 \mathbf{V}_1 의 원소는 \mathbf{V}_{-1} 의 원소에 관한 식으로 나타내어질 수 있음을 알 수 있다.

DCT 영역 필터링 행렬 \mathbf{V}_0 에 관한 유사한 식을 얻기 위하여, \mathbf{v}_0 을 2개의 행렬로 $\mathbf{v}_0 = \mathbf{v}_0^- + \mathbf{v}_0^+$ 와 같이 분해할 수 있다. 여기에서

$$\mathbf{v}_0^- = \begin{bmatrix} \frac{1}{2} v_0 & v_1 & \cdots & v_6 & v_7 \\ 0 & \frac{1}{2} v_0 & \ddots & \ddots & v_6 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \frac{1}{2} v_0 & v_1 \\ 0 & 0 & \cdots & 0 & \frac{1}{2} v_0 \end{bmatrix}$$

이고 $\mathbf{v}_0^+ = (\mathbf{v}_0^-)^t$ 이다. $q_{m,n}^-$ 과 $q_{m,n}^+$ 을 각각 \mathbf{v}_0^- 와 \mathbf{v}_0^+ 의 (m,n) 번째 원소라 정의한다. 행렬 \mathbf{V}_0^- 와 \mathbf{V}_0^+ 은 각각 행렬 행렬 \mathbf{v}_0^- 와 \mathbf{v}_0^+ 의 DCT를 나타낸다. $b_{i,j}$, $b_{i,j}^-$, $b_{i,j}^+$ 를 각각 행렬 \mathbf{V}_0 , \mathbf{V}_0^- , \mathbf{V}_0^+ 의 (i,j) 번째 원소라 정의하면 $b_{i,j} = b_{i,j}^- + b_{i,j}^+$ 이 성립한다. 위와 같은 방법으로 식을 전개하면, $b_{i,j}^+ = (-1)^{i+j} b_{i,j}^-$ 이 성립함을 보일 수 있다. 따라서

$$i+j \text{가 홀수이면, } b_{i,j} = 0 \quad (8)$$

이상에서 필터 계수가 대칭이면, DCT 영역 필터링

행렬 \mathbf{V}_0 원소의 반이 0인 성질을 보였다.

V. DCT 영역 분리 대칭 선형 필터링을 위한 효과적인 방법

이 장에서는 IV장의 성질을 이용하여 DCT 영역 분리 대칭 선형 필터링을 위한 효과적인 방법을 제안한다. 식 (2)의 효과적인 계산을 위하여 $\mathbf{Z}_{i,j} \equiv \mathbf{Z}_{i,j}^{(1)} + \mathbf{Z}_{i,j}^{(0)}$ 로 분해한다. 여기에서 $\mathbf{Z}_{i,j}^{(1)} \equiv \mathbf{V}_{-1} \mathbf{X}_{i-1,j} + \mathbf{V}_1 \mathbf{X}_{i+1,j}$, $\mathbf{Z}_{i,j}^{(0)} \equiv \mathbf{V}_0 \mathbf{X}_{i,j}$ 로 정의된다.

먼저 식 (7)에 의하여 $\mathbf{Z}_{i,j}^{(1)}$ 의 계산 양을 감소시킨다. $x_{k,l}^-$, $x_{k,l}^+$, $z_{k,l}^{(1)}$ 을 각각 행렬 $\mathbf{X}_{i-1,j}$, $\mathbf{X}_{i+1,j}$, $\mathbf{Z}_{i,j}^{(1)}$ 의 (k,l) 번째 원소로 정의하면

$$\begin{aligned} z_{k,l}^{(1)} &= \sum_{m=0}^7 a_{k,m}^- x_{m,l}^- + \sum_{m=0}^7 a_{k,m}^+ x_{m,l}^+ \\ &= \sum_{n=0}^3 a_{k,2n}^- (x_{2n,l}^- + (-1)^{k+2n} x_{2n,l}^+) \\ &\quad + \sum_{n=0}^3 a_{k,2n+1}^- (x_{2n+1,l}^- + (-1)^{k+2n+1} x_{2n+1,l}^+) \end{aligned} \quad (9)$$

식 (9)는 짝수와 홀수의 경우를 나누어서 다음과 같이 계산될 수 있다.

k 가 짝수(even)일 때,

$$z_{k,l}^{(1)} = \sum_{n=0}^3 a_{k,2n}^- (x_{2n,l}^- + x_{2n,l}^+) + \sum_{n=0}^3 a_{k,2n+1}^- (x_{2n+1,l}^- - x_{2n+1,l}^+)$$

k 가 홀수(odd)일 때,

$$z_{k,l}^{(1)} = \sum_{n=0}^3 a_{k,2n}^- (x_{2n,l}^- - x_{2n,l}^+) + \sum_{n=0}^3 a_{k,2n+1}^- (x_{2n+1,l}^- + x_{2n+1,l}^+) \quad (10)$$

식 (10)과 같은 행렬 $\mathbf{Z}_{i,j}^{(1)}$ 의 계산은 512번의 곱셈을 필요로 한다.

다음으로 식 (8)에 의하여 $\mathbf{Z}_{i,j}^{(0)}$ 의 계산 양을 감소시킨다. $x_{k,l}$ 과 $z_{k,l}^{(0)}$ 을 각각 $\mathbf{X}_{i,j}$ 과 $\mathbf{Z}_{i,j}^{(0)}$ 의 (k,l) 번째 원소라 하면,

$$z_{k,l}^{(0)} = \sum_{m=0}^7 b_{k,m} x_{m,l} = \sum_{n=0}^3 b_{k,2n} x_{2n,l} + \sum_{n=0}^3 b_{k,2n+1} x_{2n+1,l}$$

식 (8)에서 k 가 홀수일 때는 $b_{k,2n} = 0$ 이고, k 가 짝수일 때는 $b_{k,2n+1} = 0$ 인 것을 알 수 있다. 따라서 $\mathbf{Z}_{i,j}^{(0)}$ 를 위한 행렬 연산은 다음과 같이 수행할 수 있다.

$$z_{k,l}^{(0)} = \begin{cases} \sum_{n=0}^3 b_{k,2n} x_{2n,l} & \text{for even } k \\ \sum_{n=0}^3 b_{k,2n+1} x_{2n+1,l} & \text{for odd } k \end{cases} \quad (11)$$

식 (11)과 같은 행렬 $\mathbf{Z}_{i,j}^{(0)}$ 의 계산은 256번의 곱셈을 요구한다.

식 (3)의 계산 양 감소는 이상에서 기술한 식 (2)의 계산 양 감소와 같은 방법으로 수행할 수 있다. 식 (3)의 효과적인 계산을 위하여 $\mathbf{Y}_{i,j}$ 는 $\mathbf{Y}_{i,j} \equiv \mathbf{Y}_{i,j}^{(1)} + \mathbf{Y}_{i,j}^{(0)}$ 와 같이 분해한다. 여기에서 $\mathbf{Y}_{i,j}^{(1)} \equiv \mathbf{Z}_{i,j-1} \mathbf{H}_{-1}^t + \mathbf{Z}_{i,j+1} \mathbf{H}_1^t$, $\mathbf{Y}_{i,j}^{(0)} \equiv \mathbf{Z}_{i,j} \mathbf{H}_0^t$ 로 정의된다. 여기에서 $d_{i,j}^-$ 와 $d_{i,j}^+$ 를 각각 행렬 \mathbf{H}_{-1}^t 와 \mathbf{H}_1^t 의 (i,j) 번째 원소를 나타내기로 한다. IV장에서와 같은 방법으로 식을 전개하면, $d_{i,j}^+ = (-1)^{i+j} d_{i,j}^-$ 가 성립하는 것을 보일 수 있다. $z_{k,l}^-$, $z_{k,l}^+$, $y_{k,l}^{(1)}$ 을 각각 행렬 $\mathbf{Z}_{i,j-1}$, $\mathbf{Z}_{i,j+1}$, $\mathbf{Y}_{i,j}^{(1)}$ 의 (k,l) 번째 원소를 나타내기로 하면, 행렬 $\mathbf{Y}_{i,j}^{(1)}$ 는 다음과 같이 계산될 수 있다.

k 가 짝수일 때,

$$y_{k,l}^{(1)} = \sum_{n=0}^3 ((z_{k,2n}^- + z_{k,2n}^+) d_{2n,l}^- + (z_{k,2n+1}^- - z_{k,2n+1}^+) d_{2n+1,l}^-)$$

k 가 홀수일 때,

$$y_{k,l}^{(1)} = \sum_{n=0}^3 ((z_{k,2n}^- - z_{k,2n}^+) d_{2n,l}^- + (z_{k,2n+1}^- + z_{k,2n+1}^+) d_{2n+1,l}^-)$$

(12)

식 (12)와 같은 행렬 $\mathbf{Y}_{i,j}^{(1)}$ 의 계산은 512번의 곱셈을 요구한다.

$e_{i,j}$ 를 \mathbf{H}_0^t 의 (i,j) 번째 원소라 하고 위에서와 같은 방법으로 식을 전개하면, 식 (8)에서와 같이 $i+j$ 가 홀수일 때 $e_{i,j} = 0$ 인 것을 보일 수 있다. $z_{k,l}$ 과 $y_{k,l}^{(0)}$ 을 각각 행렬 $\mathbf{Z}_{i,j}$ 와 $\mathbf{Y}_{i,j}^{(0)}$ 의 (k,l) 번째

원소라 하면, 행렬 $\mathbf{Y}_{i,j}^{(0)}$ 는 다음과 같이 계산될 수 있다.

$$y_{k,l}^{(0)} = \begin{cases} \sum_{n=0}^3 z_{k,2n} e_{2n,l} & \text{for even } l \\ \sum_{n=0}^3 z_{k,2n+1} e_{2n+1,l} & \text{for odd } l \end{cases} \quad (13)$$

식 (13)과 같은 행렬 $\mathbf{Y}_{i,j}^{(0)}$ 의 계산은 256번의 곱셈을 요구한다.

전체적으로 이 장에서 제안된 DCT 영역 필터링 방법은 하나의 8×8 DCT 블록에 대하여 $512+256+512+256=1536$ 번의 곱셈을 필요로 한다. 제안된 DCT 영역의 필터링 방법은 반올림 에러 (rounding error)를 제외하면, 공간 영역 (spatial domain)의 필터링 방법과 같은 결과를 얻는다.

VI. DCT 계수의 희소성에 의한 계산 양 감소

DCT 계수의 희소성 (sparseness)은 DCT 영역 필터링을 위해서 매우 중요한 요소이다. DCT는 우수한 에너지 압축 효과가 있기 때문에 JPEG, MPEG-2, MPEG-4, H.263, H.26L 등의 영상, 비디오 표준에 채택이 되었다. 실제적으로 양자화 (quantization) 후에는 일반적 DCT 블록에서 높은 비율의 계수가 0 값을 갖는데, 특히 낮은 전송율 (low bit-rate) 압축에서 이러한 현상이 강하게 나타난다.

α 와 β 를 각각 입력 DCT 행렬 $\mathbf{X}_{i,j}$ 와 1차원 필터링된 DCT 행렬 $\mathbf{Z}_{i,j}$ 의 0이 아닌 계수의 비율을 나타내기로 한다. 제안된 DCT 영역 2차원 선형 필터링에 필요한 곱셈 수 m 는 다음과 같다.

$$m = 768\alpha + 768\beta \quad (14)$$

실제적으로 희소성의 정도는 영상 특성이나 양자화 연산 등의 많은 요소에 영향을 받는다. 따라서 α 와 β 는 다른 영상과 다른 경우에 대하여 값이 달라진다. 참고 문헌 [3]에서는, 왼쪽 위 4분의 1 부분면 4×4 부분의 계수만 0이 아니면, 즉 나머지 4분의 3 부분의 계수가 전부 0이면 DCT 블록이 희소 (sparse) 하다고 정의하였다. 이 경우에는 $\alpha = \beta = 0.25$ 이고, 식 (14)의 곱셈 수는 384가

된다.

VII. 복잡 정도 비교

이 장에서는 이전의 DCT 영역 필터링 방법과 제안된 DCT 영역 필터링 방법의 복잡 정도를 비교한다. 제안된 방식을 공간 영역 필터링 방법, DCT/DST에 근거한 방법^[3]과 비교 하였다. 원형 필터링에 근거한 방법^[5]과 근사 방법^[2]은 블록 효과 (blocking artifact)를 초래할 수 있기 때문에 비교 대상에 포함되지 않았다. 여기에서는 곱셈의 수와 VLSI 구현 관점을 비교하였다.

공간 영역 방법은 IDCT, 공간 영역 필터링, DCT를 필요로 한다. 공간 영역의 필터링은 II장에서 K 와 L 값에 영향을 받는다. 필터 계수가 대칭일 때 공간 영역의 필터링은 하나의 8×8 블록에 대하여 1차원으로의 분리 계산을 수행하면 $64K+64L+128$ 번의 곱셈을 필요로 한다. 이제 DCT와 IDCT 계산에 대하여 생각한다. 2차원 DCT/IDCT 구현을 위하여 많은 고속 알고리즘과 VLSI 구조가 제안되어 왔다. 2차원 DCT/IDCT VLSI 구현을 위하여 가장 많이 이용되는 방법은 행-열 방법 (row-column method)이다^[7]. 일반적인 행-열 방법에 의한 2차원 DCT/IDCT 계산은 하나의 8×8 블록에 대하여 192/192 번의 곱셈을 필요로 한다. 전체적으로 공간 영역 방법은 하나의 8×8 DCT 블록에 대하여 $64K+64L+512$ 번의 곱셈을 필요로 한다.

DCT/DST에 근거한 방법^[3]은 DCT/DST 컨벌루션 곱셈 성질 (convolution multiplication property)을 이용한 방법이다. 참고 문헌 [3]에 의하면 대칭 계수에 의하여 하나의 8×8 DCT 블록에 대하여 DCT 계수가 회소하지 않은 경우에는 736 번, 회소할 경우에는 448번의 곱셈을 필요로 한다. 이 방법은 DCT 영역 필터링을 위하여 DCT 계수 외에도 DST 계수를 이용한다. DST 계수는 미리 주어지는 것이 아니기 때문에 주어진 DCT 계수로부터 계산되어야 한다. 그리고 DST 계수는 DCT 영역 필터링의 완성을 위하여 다시 DCT 계수로 변환되어야 한다. 따라서 이 방법은 코사인의 사인 변환 (cosine-to-sine transform: CST) 부분과 사인의 코사인 변환 (sine-to-cosine transform: CST) 변환 부분을 필요로 한다. 그리고 이 방법은 행렬을 뒤집는 (flipping) 연산을 많이 필요로 한다. 행렬을 뒤집는 연산은 곱셈을 필요로 하지는

않지만, VLSI 구현 관점에서 무시될 수 없는 연산이다. 전체적으로 DCT/DST에 근거한 방법은 CST/SCT 계산과 별도의 행렬을 뒤집는 연산을 위해서 매우 복잡한 신호의 흐름 (signal flow)을 초래하여서 효과적인 VLSI 구현을 위해서는 적절하지가 않다. DCT 영역 필터링의 중요한 요소는 양자화된 입력 DCT 블록의 일반적인 회소성이다. 그런데 DCT 영역 계수를 DST 영역으로 변환하면 DCT 영역에서의 회소성이 일반적으로 잘 유지가 되지 않는다^[3]. 따라서 DCT/DST에 근거한 방법은 DCT 계수의 회소성을 잘 이용하지 못하는 방법이라고 할 수 있다.

제안된 방식은 회소하지 않은 경우 1536번, 회소한 경우 384번의 곱셈을 필요로 한다. 입력 DCT 블록이 회소한 경우에는 제안된 방법이 곱셈의 수에 있어서 가장 좋은 방법이라고 할 수 있다. 회소한 경우는 DCT에 근거한 압축에 있어서 매우 일반적인 경우라고 할 수 있다. 실제로 DCT 블록은 더 회소하고, 즉 25%의 계수보다 더 적은 계수만이 0이 아니고, 이러한 현상은 저 전송율 압축일수록 더 현저하게 나타난다.

VLSI 구현을 위해서는 제안된 방식이 가장 효과적인 방식이라고 할 수 있다. 제안된 방식은 규칙적인 연산에 의한 간단한 신호의 흐름으로 구현될 수 있다. 즉 단순한 곱셈과 덧셈의 연속적인 연산이라 할 수 있다. 제안된 방법은 DCT/DST에 근거한 방법과 같은 별도의 행렬을 뒤집는 연산을 필요로 하지 않는다. 제안된 방법은 간단한 곱셈과 덧셈으로만 구성되어 있으므로, 효과적으로 VLSI를 구현하는 것이 가능하다. 이러한 효과적인 VLSI 구현은 향후에 연구되어질 수 있다.

VII. 결 론

본 논문에서는 DCT 영역에서의 효과적인 분리 대칭 선형 필터링 방법을 제안하였다. 먼저 2차원 분리 선형 필터링을 DCT 영역에서의 1차원 필터링이 연결된 형태로 분해하였다. 그리고 필터 계수가 대칭적일 때, DCT 영역 필터링 행렬의 특별한 성질을 분석하였다. 이러한 성질을 이용하여 효과적인 DCT 영역에서의 2차원 필터링을 수행 하였다. 제안된 방법은 기존의 방법에 비교하여 DCT 계수의 회소성을 고려할 때 더 적은 수의 곱셈을 요구한다. 또한 제안된 방법은 간단하고 규칙적인 구조로 구성되어 있기 때문에, 효과적인 VLSI 구

현에 적합하다.

참 고 문 헌

[1] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," IEEE Journal of Selected Areas in Comm., vol. 13, no. 1, pp.1-11, Jan. 1995.

[2] N. Merhav and R. Kresch, "Approximate convolution using DCT coefficient multipliers," IEEE Trans. Circ. Syst. Video Tech., vol. 8, no. 4, pp. 378-385, Aug. 1998.

[3] R. Kresch and N. Merhav, "Fast DCT domain filtering using the DCT and the DST," IEEE Trans. Image Processing, vol. 8, no. 6, pp. 821-833, June 1999.

[4] C. Yim and M. A. Isnardi, "An efficient method for DCT-domain image resizing with mixed field/frame mode macroblocks", IEEE Trans. Circ. Syst. Video Tech., vol. 9, no. 5, pp.696-700, Aug. 1999.

[5] B. Chitprasert and K. R. Rao, "Discrete cosine transform filtering," Signal Processing, vol. 19, pp.233-245, 1990.

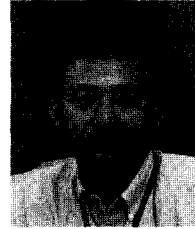
[6] J. B. Lee and B. G. Lee, "Transform domain filtering based on pipelining structure," IEEE Trans. Signal Proc., vol. 40, no. 8, pp.2061-2064, 1992.

[7] A. Madisetti and A. N. Wilson, Jr. "A 100 MHz 2-D 8x8 DCT/IDCT processor for HDTV applications," IEEE Trans. Circ. Syst. Video Tech., vol. 5, no. 2, pp. 158-165, Apr. 1995.

[8] N. I. Cho and S. U. Lee, "Fast algorithm and implementation of 2-D discrete cosine transform," IEEE Trans. Circ. Syst., vol. 38, no. 3, pp.297-305, Mar. 1991.

임 창 훈(Changhoon Yim)

정회원



1986년 2월 : 서울대학교 제어계측공학과 학사

1988년 2월 : 한국과학기술원 전기 및 전자공학과 석사

1996년 12월 : The University of Texas at Austin, Electrical & Computer Engineering, Ph.D

1988년 3월 ~ 1991년 6월 : 한국방송공사 기술연구소 연구원

1996년 12월 ~ 1999년 3월: Sarnoff Corporation, Member of Technical Staff

1999년 3월 ~ 2000년 7월: Lucent Technologies, Bell Labs, ASIC Engineer

2000년 8월 ~ 2002년 3월: KLA-Tencor Corporation, Sr. Software Engineer

2002년 5월 ~ : 삼성전자 디지털미디어연구소 수석연구원

<주관심분야> 멀티미디어 처리, 비디오/영상 압축, 영상/신호 처리