

DCT 영역에서의 POCS에 근거한 후처리

정회원 임 창 훈*

DCT Domain Post-Processing Based on POCS

Chang hoon Yim* *Regular Member*

요 약

POCS에 근거한 후처리 방식은 블록 결합 제거에 좋은 결과를 보여 주었지만 POCS는 반복적인 과정에 많은 연산을 요구하여 실시간 제약이 있는 실제적인 후처리를 위해서는 구현되기가 어렵다. 본 논문에서는 반복적인 기존의 POCS와 거의 비슷한 성능을 보이면서 계산 양을 획기적으로 감축한 DCT 영역에서의 POCS 방식을 제안한다. DCT 영역에서의 POCS 방식에서는 저역 통과 필터링을 DCT 영역에서 수행하여 역방향 DCT와 순방향 DCT 모듈을 제거하였다. 반복적인 POCS에서의 저역 통과 필터링을 분석하여 k 번째 반복 과정에서의 필터링과 동일한 효과를 갖는 k 차 저역 통과 필터링을 정의하고 여기에 해당하는 k 차 DCT 영역 POCS를 정의한다. 시뮬레이션 결과는 반복적 과정이 없는 k 차 DCT 영역 POCS는 k 번 반복적 과정을 거친 기존의 POCS 방식에 비하여 PSNR과 주관적 화질 면에서 거의 비슷한 성능을 보여 주면서 훨씬 작은 계산 양을 필요로 한다. 본 논문에서 제안하는 DCT 영역에서의 POCS 방식은 실제적인 실시간 동작을 필요로 하는 후처리 구현을 위해서 효과적으로 사용될 수 있다.

Key Words : POCS; post-processing; DCT domain filtering; blocking artifact; lowpass filtering

ABSTRACT

Even though post-processing methods based on projections onto convex sets (POCS) have shown good performance for blocking artifact reduction, it is infeasible to implement POCS for real-time practical applications. This paper proposes DCT domain post-processing method based on POCS. The proposed method shows very similar performance compared to the conventional POCS method, while it reduces tremendously the computational complexity. DCT domain POCS performs the lowpass filtering in the DCT domain, and it removes the inverse DCT and forward DCT modules. Through the investigation of lowpass filtering in the iterative POCS method, we define k th order lowpass filtering which is equivalent to the lowpass filtering in the k th iteration, and the corresponding k th order DCT domain POCS. Simulation results show that the k th order DCT domain POCS without iteration gives very similar performance compared to the conventional POCS with k iterations, while it requires much less computations. Hence the proposed DCT domain POCS method can be used efficiently in the practical post-processing applications with real-time constraints.

I. 서론

JPEG, MPEG-2, MPEG-4, H.261, H.263을 비롯한 많은 영상(image) 및 비디오(video) 표준 압축

(compression) 방식은 이산 코사인 변환(discrete cosine transform; DCT) 코딩 기술에 기반을 두고 있다. 영상과 비디오를 압축 영역(compressed domain)에서 처리를 하게 되면, 압축된 데이터와 압축

* 건국대학교 정보통신대학 인터넷미디어공학부(cyim@konkuk.ac.kr)
논문번호 : KICS2004-06-027, 접수일자 : 2004년 6월 8일

되지 않은 데이터 간의 변환(conversion) 과정을 줄일 수 있어서, 전체적인 계산량을 줄일 수 있다. 영상과 비디오를 압축 영역 처리(compressed domain processing) 관점에서 DCT 영역에서 직접 처리하는 방법들이 연구되어 왔다^[43].

블록 DCT 코딩 방식은 대체로 좋은 성능을 보이지만, 많은 경우에 있어서, 특히 낮은 비트율에서 블록 결함(blocking artifact)을 보인다. 블록 결함을 줄이기 위한 많은 연구가 수행되어 왔는데, 이 방법들은 크게 2가지의 부류로 나눌 수 있다^{[44][5]}. 그 중 한 가지는 겹쳐진(interleaved) 블록 변환 방법 등과 같은 변형된 형태의 변환 구조 방법이다^[7].

다른 방법은 필터링(filtering)이나 블록 집합으로의 투영(projection onto convex sets; POCS)에 근거한 후처리(post-processing) 방식이 있다^{[44][5][9][11]}. 후처리 방식은 부호화기(encoder)와 복호화기(decoder)를 변화시킬 필요가 없는 장점이 있다. 가장 간단한 후처리 방식은 저역통과 필터링(low-pass filtering) 방법인데, 간단한 저역통과 필터링 방법은 고주파수(high-frequency) 성분(component)이 심하게 손상을 받아서, 영상이 심하게 흐려지는 현상이 나타날 수 있다. 이에 비하여 POCS에 근거한 방식은 양자화 제한 집합(quantization constraint set; QCS)을 정의하여 후처리된 영상이 양자화된 영상과 같은 집합에 포함되도록 한다. 양자화 제한 집합은 양자화된 영상과 정규화 배열(normalization array)에 의하여 간단하게 구할 수 있다. 양자화 제한 집합으로의 투영 연산, 간단히 투영 연산은 clipping 연산에 의하여 수행된다. 투영 연산은 후처리된 영상이 발산(diverge)하는 것을 방지하는데 효과적이다. POCS에서는 저역통과 필터링이 공간 영역(spatial domain)에서 수행된다.

투영 연산과 저역통과 필터링이 POCS에서 반복적으로 수행되는데, 투영 연산은 DCT 영역에서 수행되어야 하고, 저역통과 필터링은 공간 영역에서 수행되어야 한다. CT 영역과 공간 영역을 오고 가기 위해서는 역방향 이산 코사인 변환(inverse discrete cosine transform; IDCT)과 순방향 이산 코사인 변환(forward DCT)이 반복적으로 수행되어야 한다. 따라서 일반적인 POCS 방식에서는 IDCT, 공간 영역 저역통과 필터링, DCT, 투영 연산들이 반복적으로 수행되어야 한다.

최근에 인터넷이나 무선 네트워크를 통한 낮은 비트율에서의 비디오 스트리밍(video streaming) 등을 위한 실제적인 응용 분야에서 블록 결함을 감

소시키기 위한 기술이 필요해지고 있다. POCS에 근거한 방식은 블록 결함 감소에 뛰어난 성능을 보여 주고 있지만, 일반적인 POCS 방식은 반복적인 과정을 필요로 하여 실시간(real-time)으로 동작해야 되는 응용 분야를 위한 후처리에서는 적용되기 어려운 불가능하다. 실시간 동작을 필요로 하는 응용 분야에서는 후처리가 간단하거나 구현 가능 정도로만 복잡해야 한다.

본 논문에서는 DCT 영역에서의 POCS에 근거하여 반복적이지 않은 후처리 방식 즉 DCT 영역 POCS 방식을 제안한다. DCT 영역 POCS 방식에서는 저역통과 필터링이 DCT 영역에서 수행된다. 따라서 저역통과 필터링과 투영 연산 모두 DCT 영역에서 수행되어, 기존의 POCS 방식과는 달리 IDCT와 DCT 모듈이 필요가 없게 된다. 그리고 각 반복 과정에서의 저역통과 필터링을 분석하여 반복이 필요 없이 반복된 필터링과 동등한 k 차 저역통과 필터링을 정의한다. DCT 영역 필터링과 k 차 저역통과 필터링 개념을 결합하여 k 차 DCT 영역 필터링을 정의한다. 투영 연산은 기존의 POCS 방식에서 매 반복 과정마다 수행했던 것과는 달리 k 차 DCT 영역 필터링이 수행된 후 한번만 수행하는 것을 제안한다. 제안된 DCT 영역 POCS 방식은 기존의 POCS 방식과 거의 비슷한 성능을 보이면서 훨씬 적은 계산량을 필요로 한다. 따라서 제안된 DCT 영역 POCS 방식은 실제적인 실시간 후처리 응용 분야의 블록 결함을 줄이기 위한 효과적인 방식으로 채택될 수 있다.

본 논문의 구성은 다음과 같다. II장은 기존의 반복적인 POCS 방식을 배경 설명으로서 소개한다. III장에서는 POCS에서의 저역통과 필터링을 분석하고, k 번 반복된 필터링과 동일한 k 차 저역통과 필터링을 정의하고, IV장은 DCT 영역에서의 2차원 선형 필터링을 기술한다. V장에서 DCT 영역에서의 POCS 방식을 제안한다. VI장에서는 제안된 DCT 영역에서의 POCS 방식과 기존의 POCS 방식의 시뮬레이션 결과를 보여주고, VII장에서는 기존의 POCS 방식과 제안된 DCT 영역에서의 POCS 방식의 복잡도를 비교한다. 마지막으로 VIII장에서 결론을 내린다.

II. 기존의 반복적인 POCS 방식

그림 1은 기존의 반복적인 POCS 방식의 블록도를 나타낸다^{[9][11]}. 첫 번째로 DCT 계수(coefficients)

들로부터 공간 영역 영상을 얻기 위하여 역방향 이산 코사인 변환(IDCT)이 수행된다. 두 번째로 영상을 smoothing 하기 위하여 공간 영역 저역통과 필터링이 수행된다. 세 번째로 DCT 계수를 얻기 위하여 순방향 이산 코사인 변환(forward DCT)이 수행된다. 네 번째로 양자화 제한 조건을 만족시키기 위하여 clipping 연산이 투영 연산으로서 수행된다.

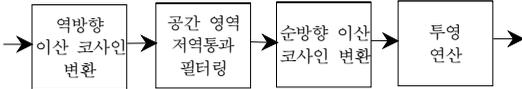


그림 1. 기존의 POCS 방식의 블록도한 번의 반복 과정

기존의 공간 영역에서의 POCS 방식에서는, 위의 네 가지 연산이 반복적으로 수행되어야 한다. 즉, k 번의 반복을 위해서는 그림 1의 4가지 연산이 k 번 반복되어야 한다. 따라서 기존의 공간 영역에서의 POCS 방식에서는 반복 횟수가 증가함에 따라서 복잡도가 선형적으로 비례하여 증가한다.

III. POCS에서의 저역통과 필터링

기존의 POCS 방식에서는 영상의 smoothing을 위하여 저역통과 필터링이 공간 영역에서 수행된다. $\{x(m,n)\}$ 과 $\{y(m,n)\}$ 은 각각 입력 영상과 필터링된 영상을 나타낸다. $\{f(m,n)\}$ 은 2차원 저역통과 필터링을 위한 필터 계수를 나타낸다. 여기에서는 대부분의 다른 POCS 방식^{[4],[5],[9]-[12]}에서와 마찬가지로 2차원 필터를 1차원 필터로 분리 가능(separable)하고, 또한 필터 계수가 대칭(symmetric)이라고 가정한다. 2차원 필터가 분리 가능이면 $\{f(m,n)\}$ 은 1차원 필터 $\{v(m)\}$, $\{h(n)\}$ 에 의하여 $f(m,n) = v(m)h(n)$ 과 같이 분리될 수 있다. 여기에서 1차원 필터 $\{v(m)\}$, $\{h(n)\}$ 의 구간을 각각 $-M \leq m \leq M$, $-N \leq n \leq N$ 이라고 가정한다.

기존의 POCS에서는 저역통과 필터링이 각 반복 과정마다 수행된다. k 번째 반복 과정에서는 $(k-1)$ 번째 반복 과정의 결과에 대하여 필터링이 수행된다. k 번째 반복 과정 후의 필터링된 결과를 $y^{(k)}(m,n)$ 이라 하면, $y^{(k)}(m,n) = f(m,n) * y^{(k-1)}(m,n)$ 이 된다.

여기에서 $f^{(1)}(m,n) \equiv f(m,n)$, $f^{(k)}(m,n) \equiv f^{(1)}(m,n) * f^{(k-1)}(m,n)$ 이라 하면, $y^{(k)}(m,n) \equiv f^{(k)}(m,n) * x(m,n)$ 임을 보일 수 있다. $f^{(k)}(m,n)$ 을 k 차(order) 저역통과 필터로 부르기로 하면, k 차 저역통과 필터 $f^{(k)}(m,n)$ 은 기존의 POCS 방법에서 k

번째 반복 과정에서의 저역 통과 필터와 동등한 역할을 수행하게 된다.

k 차 저역통과 필터 계수 $\{f^{(k)}(m,n)\}$ 도 역시 분리 가능한데, 1차원 필터 계수 $\{v(m)\}$ 과 $\{h(n)\}$ 으로부터 구해질 수 있다. $v^{(1)}(m) = v(m)$, $v^{(k)}(m) = v^{(1)}(m) * v^{(k-1)}(m)$, $h^{(1)}(n) = h(n)$, $h^{(k)}(n) \equiv h^{(1)}(n) * h^{(k-1)}(n)$ 이라 하면, $f^{(k)}(m,n) = v^{(k)}(m)h^{(k)}(n)$ 과 $y^{(k)}(m,n) = v^{(k)}(m)h^{(k)}(n) * x(m,n)$ 이 성립함을 보일 수 있다. 대부분의 기존 POCS 연구^{[9],[12]}에서 사용되었던 2차원 필터 계수는 3×3 으로서, $f(0,0) = 0.2042$, $f(0,1) = f(0,-1) = f(1,0) = f(-1,0) = 0.1239$, $f(1,1) = f(1,-1) = f(-1,1) = f(-1,-1) = 0.0751$ 이다. 이와 같은 2차원 필터는 분리 가능하고 대칭적인데, 여기에 해당되는 1차원 필터 계수는 $v(0) = h(0) = 0.4518$, $v(1) = v(-1) = h(1) = h(-1) = 0.2741$ 이다.

1차(order) 1차원(1-dimensional) 필터 계수 $\{v^{(1)}(m)\}$ 으로부터 k 차 1차원 필터 계수들을 계산할 수 있다. 표 1은 주어진 1차 필터 계수 $\{v^{(1)}(m)\}$ 으로부터 계산된 2차, 5차, 8차 필터 계수 $\{v^{(2)}(m)\}$, $\{v^{(5)}(m)\}$, $\{v^{(8)}(m)\}$ 을 나타낸다. 모든 k 와 m 에 대하여 $v^{(k)}(-m) = v^{(k)}(m)$ 이고 $h^{(k)}(-m) = h^{(k)}(m)$ 이 성립한다. 그림 2는 $\{v^{(1)}(m)\}$, $\{v^{(2)}(m)\}$, $\{v^{(5)}(m)\}$, $\{v^{(8)}(m)\}$ 에 대한 주파수 응답(frequency response)을 나타낸다. 그림 2에서 보면, 필터의 차수(order)가 커질수록 저역통과 대역폭(bandwidth)이 좁아지는 것을 알 수 있다. 이와 같은 이유로 기존의 POCS에서 반복 횟수가 증가할수록 필터링된 영상이 더욱 smoothing 된다.

표 1. k 차(order) 필터 $\{v^{(k)}(m)\}$ 의 1차원 필터 계수 ($k = 1, 2, 5, 8$)

m	$v^{(1)}(m)$	$v^{(2)}(m)$	$v^{(5)}(m)$	$v^{(8)}(m)$
0	0.4518	0.3544	0.2339	0.1870
1	0.2741	0.2477	0.1987	0.1682
2	0	0.0751	0.1203	0.1219
3	0	0	0.0498	0.0705
4	0	0	0.0128	0.0319
5	0	0	0.0015	0.0109
6	0	0	0	0.0027
7	0	0	0	0.0004
8	0	0	0	0.0000

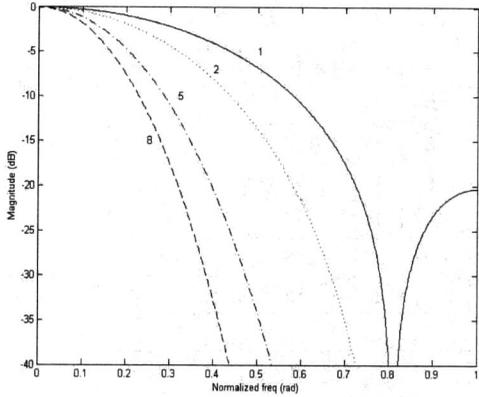


그림 2. $\{v^{(k)}(m)\}$ 에 대한 주파수 응답 ($k = 1, 2, 5, 8$)

블록 형태의 행렬식으로 나타내기 위하여 $\{x(m, n)\}$ 과 $\{y(m, n)\}$ 이 겹쳐지는 부분이 없는 8×8 행렬 $X_{i,j}$ 와 $Y_{i,j}$ 로 각각 구성되어 있다고 하고 $V \equiv [V_1 \ V_2 \ V_3]$ 과 $H \equiv [H_1 \ H_2 \ H_3]$ 을 정의한다. 여기에서

$$V_1 \equiv \begin{bmatrix} v(8) & v(7) & v(6) & v(5) & v(4) & v(3) & v(2) & v(1) \\ 0 & v(8) & v(7) & v(6) & v(5) & v(4) & v(3) & v(2) \\ 0 & 0 & v(8) & v(7) & v(6) & v(5) & v(4) & v(3) \\ 0 & 0 & 0 & v(8) & v(7) & v(6) & v(5) & v(4) \\ 0 & 0 & 0 & 0 & v(8) & v(7) & v(6) & v(5) \\ 0 & 0 & 0 & 0 & 0 & v(8) & v(7) & v(6) \\ 0 & 0 & 0 & 0 & 0 & 0 & v(8) & v(7) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & v(8) \end{bmatrix}$$

$$V_2 = \begin{bmatrix} v(0) & v(1) & v(2) & v(3) & v(4) & v(5) & v(6) & v(7) \\ v(1) & v(0) & v(1) & v(2) & v(3) & v(4) & v(5) & v(6) \\ v(2) & v(1) & v(0) & v(1) & v(2) & v(3) & v(4) & v(5) \\ v(3) & v(2) & v(1) & v(0) & v(1) & v(2) & v(3) & v(4) \\ v(4) & v(3) & v(2) & v(1) & v(0) & v(1) & v(2) & v(3) \\ v(5) & v(4) & v(3) & v(2) & v(1) & v(0) & v(1) & v(2) \\ v(6) & v(5) & v(4) & v(3) & v(2) & v(1) & v(0) & v(1) \\ v(7) & v(6) & v(5) & v(4) & v(3) & v(2) & v(1) & v(0) \end{bmatrix}$$

이고 $V_3 = V_1$ 이다. H_1, H_2, H_3 행렬들은 모든 k 에 대해서 v_k 를 h_k 로 바꾸어줌으로써 V_1, V_2, V_3 행렬들과 비슷하게 정의된다. 2차원 분리 대칭 선형 필터링은 $Y_{i,j} = \sum_{m=1}^3 \sum_{n=1}^3 V_m X_{i+m} H_n^t$ 과 같이 나타낼 수 있다^[2].

IV. DCT 영역 분리 대칭 선형 필터링

이번 장에서는 DCT 영역에서의 필터링을 위한 DCT 영역 분리 대칭 선형 필터링을 기술한다. $Y_{i,j}, X_{i+m,j+n}, V_m, H_n$ 행렬들을 각각 공간 영역 행렬 $Y_{i,j}, X_{i+m,j+n}, V_m, H_n$ 행렬들의 2차원 DCT 영역 행렬들이라고 정의한다. DCT는 unitary한 특성을 가지기 때문에, $Y_{i,j}$ 는 $X_{i+m,j+n}$ 에 관한 식으로 다음과 같이 나타내어질 수 있다^{[2],[15]}.

$$Y_{i,j} = \sum_{m=1}^3 \sum_{n=1}^3 V_m X_{i+m,j+n} H_n^t \quad (1)$$

여기에서 V_m 과 H_n 은 DCT 영역 필터링 행렬을 나타낸다.

DCT 영역에서의 2차원 분리 선형 필터링도 공간 영역에서의 마찬가지로 수직과 수평 방향의 1차원 필터링으로 분리되어 수행될 수 있다. DCT 영역에서의 (1)과 같은 2차원 분리 선형 필터링은 다음과 같은 2개의 DCT 영역에서의 1차원 필터링으로 분해(decomposition)될 수 있다^[16].

$$Z_{i,j} = V_1 X_{i-1,j} + V_2 X_{i,j} + V_3 X_{i+1,j} \quad (2)$$

$$Y_{i,j} = Z_{i,j-1} H_1^t + Z_{i,j} H_2^t + Z_{i,j+1} H_3^t \quad (3)$$

여기에서 $Z_{i,j}$ 는 DCT 영역 입력 행렬에 대해서 수직 방향으로의 1차원 선형 필터링 된 행렬을 나타낸다. DCT 영역에서 (2)와 (3)과 같은 2차원 분리 선형 필터링은 하나의 8×8 DCT 블록에 대해서 3072번의 곱셈(multiplication) 연산을 필요로 한다.

기존의 POCS 연구^{[9]-[12]}에서 사용되었던 2차원 필터 계수는 분리 대칭이다. 분리 대칭인 2차원 필터 계수에 대해서는 하나의 8×8 DCT 블록에 대한 DCT 영역에서의 2차원 필터링은 1536번의 곱셈 연산을 통하여 계산할 수 있다^[16].

V. DCT 영역에서의 POCS

이번 장에서는 DCT 영역에서의 POCS 방식을 제안한다. 그림 3은 제안된 DCT 영역에서의 POCS 방법의 블록도를 나타낸다. 여기에서 보면, 제안된 방식은 반복적인 과정이 없이, DCT 영역 저역통과 필터링과 투영 연산의 단 2가지 연산으로 수행되는 것을 알 수 있다.

그림 1의 기존의 POCS 방식에서는 역방향 이산 코사인 변환, 순방향 이산 코사인 변환을 포함한 4가지의 연산이 반복적으로 수행되어, 계산 양이 반복 회수에 선형적으로 증가한다. 그림 3의 단 한 가지의 DCT 영역 저역통과 필터링 연산은 그림 1의 역방향 이산 코사인 변환, 공간 영역 저역통과 필터링, 순방향 이산 코사인 변환의 3가지 연산을 연속적으로 수행하는 것과 동일하다.

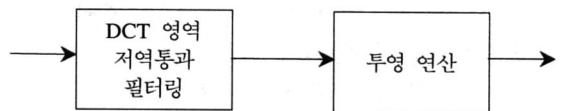


그림 3. 제안된 DCT 영역에서의 POCS 방식의 블록도

III장에서 POCS를 위한 k 차 저역통과 필터링을 정의하였다. k 차 저역통과 필터링은 기존의 POCS에서 k 번째 반복 과정에서의 저역통과 필터와 동일한 효과를 갖는다. 만약 DCT 영역 필터링을 위해서 k 차 저역통과 필터 계수가 사용되었으면, 이 경우를 k 차 DCT 영역 필터링이라고 정의한다. 필터 계수가 주어지면, DCT 영역 필터링을 위한 행렬 ($V_n, H_n, n = 1, 2, 3$)들은 미리 계산되어질 수 있다. k 차 DCT 영역 필터링이 사용될 경우의 DCT 영역 POCS를 k 차 DCT 영역 POCS라 정의한다. 그림 3에서의 투영 연산은 기존의 POCS에서의 투영 연산과 동일하다. k 차 DCT 영역 POCS를 위해서는 k 차 DCT 영역 필터링이 수행된 후 투영 연산을 단 한 번만 수행하면 된다. 그림 3의 POCS 블록도는 DCT 영역에서의 POCS를 위한 전체의 연산을 나타낸 것이고, 그림 1의 기존의 POCS 블록도는 한 번의 반복 과정만을 나타낸 것이다.

제안된 DCT 영역에서의 POCS 방식은 기존의 POCS 방식에 비하여 훨씬 적은 연산을 필요로 하는데 그 이유는 다음과 같다. 첫째, 제안된 DCT 영역에서의 POCS 방식은 어떠한 반복 과정도 필요로 하지 않는다. 둘째, 제안된 DCT 영역에서의 POCS 방식은 역방향 이산 코사인 변환(IDCT)과 순방향 이산 코사인 변환(forward DCT)을 필요로 하지 않는다. 셋째, DCT 계수들은 양자화 이후에, 특히 낮은 비트율 코딩에서, 일반적으로 매우 희박한(sparse) 특성을 갖는다. 제안된 DCT 영역에서의 POCS 방식과 기존의 POCS 방식에 대한 자세한 복잡도 비교는 VII장에서 기술한다.

VI. 시뮬레이션 결과

이번 장에서는 제안된 DCT 영역에서의 POCS 방식과 기존의 공간 영역에서의 POCS 방식의 성능을 평가하기 위한 시뮬레이션 결과를 제시한다. 표 2는 DCT 계수의 양자화(quantization)를 위해서 시뮬레이션에서 사용된 양자화 테이블^{[9],[10],[11]}을 나타낸다.

그림 4는 'Barbara' 영상에 대한 기존의 POCS 방식의 반복 회수(iteration number)와 제안된 DCT 영역에서의 POCS 방식의 차수(order)에 대한 후처리된 영상의 peak signal-to-noise ratio (PSNR) 값 비교를 나타낸다. 그림 4에서 '+' 점선은 기존의 POCS 방식의 PSNR 값, 'o' 실선은 제안된 DCT 영역 POCS 방식의 PSNR 값을 나타낸다. 여기에서

표 2. 양자화 테이블^{[9],[10],[11]}

50	60	70	70	90	120	255	255
60	60	70	96	130	255	255	255
70	70	80	120	200	255	255	255
70	96	120	145	255	255	255	255
90	130	200	255	255	255	255	255
120	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

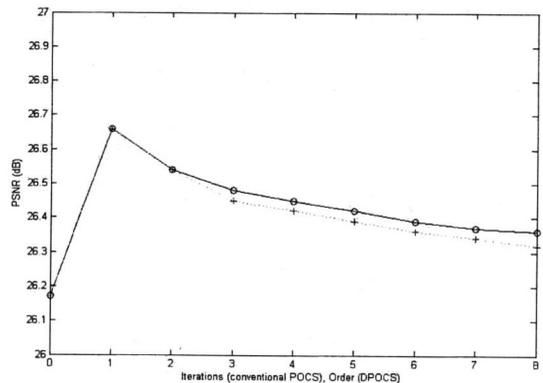


그림 4. 'Barbara' 영상에 대한 기존의 POCS 방식의 반복 회수(iteration number)와 제안된 DCT 영역에서의 POCS 방식의 차수(order)에 대한 후처리된 영상의 PSNR 값 (기존의 POCS 방식: +점선, 제안된 DCT 영역에서의 POCS 방식: o 실선)

는 기존의 POCS 방식과 제안된 DCT 영역 POCS 방식을 8번까지의 반복 회수와 거기에 해당하는 8차까지의 차수에 대하여 각각 비교하였다. 기존의 POCS 방식에서는 PSNR 값이 8번 정도의 반복 회수에서 거의 수렴(converge)하는 결과를 보여주고 있다^{[9],[11]}. 그림 4의 결과는 제안된 DCT 영역 POCS 방식이 기존의 POCS 방식과 거의 비슷한 PSNR 성능을 보여주고 있는 것을 알 수 있다. 좀 더 구체적으로는 8차의 제안된 DCT 영역 POCS 방식이 8번째 반복에서의 기존의 POCS 방식에 비하여 0.05 dB 정도 높은 PSNR 값을 보여주고 있는데, 이 정도의 PSNR 차이는 거의 무시할 수 있는 차이이다.

비슷하게 그림 5, 그림 6, 그림 7은 각각 'Lena', 'Baboon', 'Peppers' 영상에 대한 기존의 POCS 방식의 반복 회수와 제안된 DCT 영역에서의 POCS 방식의 차수에 대한 후처리된 영상의 PSNR 값 비교를 나타낸다. 이 결과들에서 보면 제안된 DCT 영역 POCS 방식이 기존의 POCS 방식과 매우 비

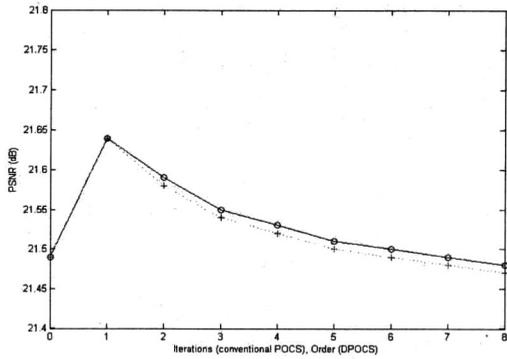


그림 5. 'Lena' 영상에 대한 기존의 POCS 방식의 반복 회수와 제안된 DCT 영역에서의 POCS 방식의 차수에 대한 후처리된 영상의 PSNR 값 (기존의 POCS 방식: +점선, 제안된 DCT 영역에서의 POCS 방식: o 실선)

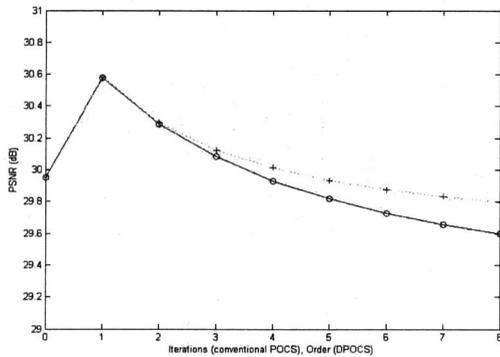


그림 6. 'Baboon' 영상에 대한 기존의 POCS 방식의 반복 회수와 제안된 DCT 영역에서의 POCS 방식의 차수에 대한 후처리된 영상의 PSNR 값 (기존의 POCS 방식: +점선, 제안된 DCT 영역에서의 POCS 방식: o 실선)

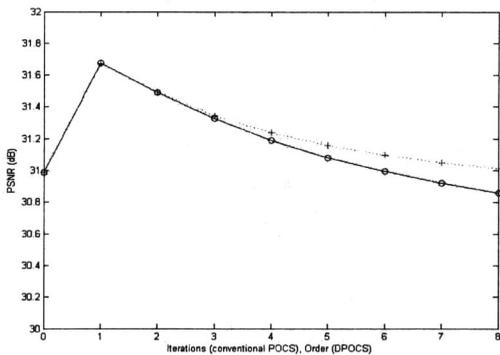


그림 7. 'Peppers' 영상에 대한 기존의 POCS 방식의 반복 회수와 제안된 DCT 영역에서의 POCS 방식의 차수에 대한 후처리된 영상의 PSNR 값 (기존의 POCS 방식: +점선, 제안된 DCT 영역에서의 POCS 방식: o 실선)

슷한 PSNR 성능을 보여주고 있는 것을 알 수 있다. 영상의 특성에 따라서 'Barbara'와 'Baboon' 영상의 경우에는 제안된 DCT 영역에서의 POCS 방식이, 'Lena'와 'Peppers' 영상의 경우에는 기존의

POCS 방식이 약간 높은 PSNR을 보이고 있는데, 그 차이가 0.2 dB 이내로서 거의 비슷한 PSNR 성능을 보인다는 사실을 알 수 있다.

그림 8은 'Barbara' 영상에 대한 시뮬레이션 결과를 나타낸다. 그림 8(a)는 원영상을 나타내며, 그림 8(b)는 DCT 부호화되어 블록 결합(blocking artifact)을 보이는 영상을 나타낸다. 그림 8(c)는 8차의 반복을 하여 기존의 POCS 방식으로 후처리된 영상을 나타내며, 그림 8(d)는 8차의 DCT 영역 저역 통과 필터링을 적용하여, 8차의 제안된 DCT 영역 POCS 방식으로 후처리된 영상을 나타낸다.

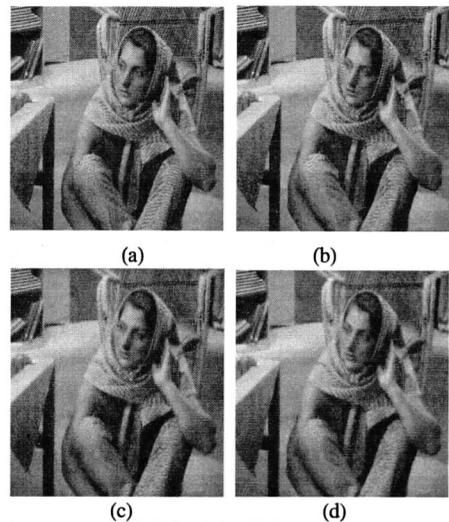


그림 8. 'Barbara' 영상에 대한 결과 (a) 원영상 (b) DCT 부호화되어 블록 효과를 보이는 영상 (c) 기존의 POCS 방식으로 후처리된 영상 (8번 반복) (d) 제안된 DCT 영역에서의 POCS 방식으로 후처리된 영상 (8차)

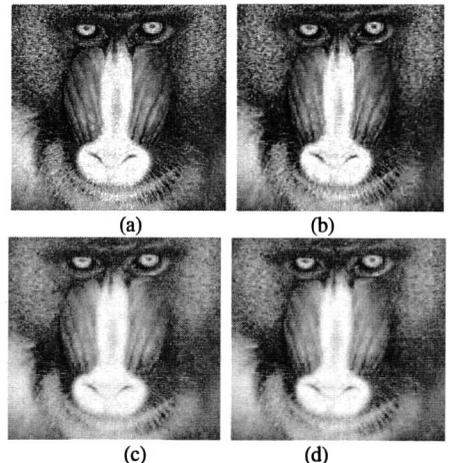


그림 9. 'Baboon' 영상에 대한 결과 (a) 원영상 (b) DCT 부호화되어 블록 효과를 보이는 영상 (c) 기존의 POCS 방식으로 후처리된 영상 (8번 반복) (d) 제안된 DCT 영역에서의 POCS 방식으로 후처리된 영상 (8차)

그림 9는 'Baboon' 영상에 대한 시뮬레이션 결과를 나타낸다. 그림 9(a)는 원영상을 나타내며, 그림 9(b)는 DCT 부호화되어 블록 결함을 보이는 영상을 나타낸다. 그림 9(c)는 8차의 반복을 하여 기존의 POCS 방식으로 후처리된 영상을 나타내며, 그림 9(d)는 8차의 제안된 DCT 영역 POCS 방식으로 후처리된 영상을 나타낸다.

VII. 복잡도 비교

이번 장에서는 기존의 반복적인 POCS 방식과 제안된 반복적이지 않은 DCT 영역에서의 POCS 방식의 복잡도를 비교한다. 그림 1에 나타난 바와 같이, 기존의 POCS 방식에서는 역방향 이산 코사인 변환(IDCT), 공간 영역 필터링, 순방향 이산 코사인 변환(DCT), 투영 연산의 4가지 과정이 필요하다. 이상의 4가지 과정이 각 반복 과정마다 필요로 하기 때문에, 기존의 POCS 방식의 복잡도는 반복 회수에 선형적으로 비례한다.

먼저 공간 영역에서의 저역 통과 필터링의 복잡도를 구하는데, 여기에서는 [9]-[12]에서 사용되었던 3x3 필터 커널(kernel)을 가정한다. 공간 영역 저역 통과 필터링을 1차원 필터링으로 분리하여 계산하면, 하나의 8x8 블록에 대하여 384번의 곱셈을 필요로 한다. 그 다음으로 DCT와 IDCT의 계산량을 구하는데, 가장 일반적인 2차원 DCT와 IDCT 구현 방법은 row-column 방법이다^[13]. 일반적인 row-column 방법에 의한 2차원 DCT 계산 방법은 하나의 8x8 블록에 대하여 192번의 곱셈을 필요로 한다^[14]. 그리고 IDCT를 위해서는 같은 수의 곱셈을 필요로 한다.

전체적으로 기존의 POCS 방식에서는 한번의 반복 과정을 위해서 768번의 곱셈을 필요로 한다. 하나의 8x8 블록에 대하여 필요한 곱셈 회수를 m 이라 하면, 기존의 POCS 방식에서 k 번의 반복 과정을 수행하기 위해서는 $m = 768k$ 가 된다. 표 3의 왼쪽 부분은 기존의 POCS 방식에서 1, 2, 5, 8번의 반복 과정을 거치기 위한 곱셈의 회수를 나타낸다. 기존의 POCS 방식에서는 반복 회수가 증가함에 따라서, 곱셈 회수가 선형적으로 비례하여 증가한다.

이제 제안된 DCT 영역에서의 POCS 방식의 복잡도를 생각해 보기로 한다. DCT 영역에서의 저역 통과 필터링에 필요한 곱셈 수는 DCT 계수의 희박한 특성(sparseness)과 관계가 있다. 실제적으로 일반적인 DCT 계수들 중 많은 부분이 양자화 후에 0

표 3. 반복 회수 1, 2, 5, 8에 대한 기존의 POCS 방식에 필요한 곱셈 개수(왼쪽 부분)와 각각에 해당하는 차수에 대한 제안된 DCT 영역에서의 POCS 방식에 필요한 곱셈 개수(오른쪽 부분)

기존의 POCS 방식		제안된 DCT 영역에서의 POCS 방식		
반복 회수	곱셈 개수	차수	희박한 특성을 고려하지 않은 곱셈 개수	전형적으로 희박한 상태를 가정한 곱셈 개수
1	768	1	1536	384
2	1536	2	1536	384
5	3840	5	1536	384
8	6144	8	1536	384

이 되는데, 특히 압축률(compression ratio)이 커질수록 이러한 경향은 심하게 나타난다. 입력 DCT 행렬 $X_{i,j}$ 와 1차원 필터링된 DCT 행렬 $Z_{i,j}$ 의 계수들 중 0이 아닌 계수의 비율을 각각 α 와 β 라 하면, DCT 영역 필터링을 (2)와 (3)과 같이 계산할 때, 곱셈 회수 m 은 다음과 같다.

$$m = 768\alpha + 768\beta \tag{4}$$

DCT 계수의 희박한 특성은 양자화 이후에 얻어 지는데, DCT 계수의 희박한 정도는 영상의 성질과 양자화 과정 등 여러 가지 요인에 의하여 달라질 수 있다. [15]에서는 8x8 블록 내에서 왼쪽 위 4x4 부분의 계수만이 0이 아닐 때, DCT 블록이 희박한 특성을 가지고 있다고 정의하였다. 이 경우에는 $\alpha = \beta = 0.25$ 이며, (4)에 의한 곱셈 회수는 384가 된다. 이 경우를 전형적으로 희박한 상태라고 정의할 수 있는데, 실제적인 낮은 비트율 코딩에서는 α 와 β 값은 대체적으로 0.25보다 작다.

표 3의 오른쪽 부분은 1, 2, 5, 8 차수에 대한 제안된 DCT 영역에서의 POCS 방식에 필요한 곱셈 수를 나타낸다. 희박한 특성을 고려하지 않았을 때는 모든 차수에 대해서 곱셈 수가 1536인데, 이는 기존의 POCS 방식에서 2번의 반복을 하였을 때와 같은 곱셈 수이다. 8차의 DCT 영역에서의 POCS 방식에 필요한 곱셈과 8번의 반복을 사용한 기존의 POCS 방식의 곱셈 수를 비교해 보면, 6144번이 1536번으로 줄어든 것을 알 수 있는데, 이는 25%의 곱셈 수에 해당된다. DCT 계수의 희박한 특성을 고려하면, 곱셈 수는 더욱 줄어든다. 전형적으로

희박한 상태($\alpha = \beta = 0.25$)를 가정한 DCT 영역에서의 POCS 방식에 필요한 곱셈 수는 모든 차수에 대하여 384이다. 전형적으로 희박한 상태를 가정한 8차 DCT 영역에서의 POCS에 필요한 곱셈 수와 8번 반복한 기존의 POCS에 필요한 곱셈 수를 비교해 보면, 6144가 384번으로 줄어든 것을 알 수 있는데, 이는 단지 6.3%의 곱셈 수에 해당한다.

VIII. 결론

본 논문에서는 DCT 영역에서의 POCS 방식을 제안하였다. DCT 영역에서의 POCS에서는 저역 통과 필터링이 DCT 영역에서 수행되기 때문에, 역방향 이산 코사인 변환과 순방향 이산 코사인 변환을 필요로 하지 않는다. 본 논문에서는 또한 각 반복 단계에서의 저역 통과 필터링을 분석하여, 반복을 하지 않은 k 차 저역 통과 필터링을 정의하였다. DCT 영역에서의 필터링과 k 차 저역 통과 필터링 개념을 결합하여, DCT 영역에서의 k 차 저역 통과 필터링을 정의하였다. 제안된 DCT 영역에서의 POCS 방식에서는 투영 연산은 DCT 영역에서의 k 차 저역 통과 필터링 이후에 단 한 번만 수행된다.

기존의 POCS 방식에서는 역방향 이산 코사인 변환, 공간영역 저역통과 필터링, 순방향 이산 코사인 변환, 투영 연산의 4가지 연산이 반복적으로 이루어 계산 양이 반복 회수에 선형적으로 증가하는데 비하여, 본 논문에서 제안하는 DCT 영역에서의 POCS 방식은 DCT 저역통과 필터링, 투영 연산 2가지의 연산이 단 한 번만 수행된다.

시뮬레이션 결과는 반복적 과정을 하지 않는 DCT 영역에서의 POCS 방식이 반복적 과정을 필요로 하는 기존의 POCS 방식과 거의 유사한 성능을 보여 주었다. 객관적인 PSNR 값과 주관적인 영상 화질 모두, 각 반복 과정에서의 기존의 POCS 결과와 각각의 차수에 해당하는 DCT 영역에서의 POCS 결과가 거의 비슷한 결과를 보여 주었다. 제안된 DCT 영역에서의 POCS 방식은 기존의 POCS 방식에 비하여 훨씬 적은 복잡도를 요구한다.

제안된 DCT 영역에서의 POCS 방식은 실제적인 실시간 응용 분야에서 블록 결합을 줄이기 위한 효과적인 방식으로 채택될 수 있다. 인터넷이나 무선 네트워크 등에서의 낮은 비트율 실시간 영상과 비디오 전송을 위한 후처리 응용은 향후 연구로 진행될 수 있다.

참고 문헌

- [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. Circuits Syst. Video Technol.*, vol. 8, no. 1, pp. 468-476, Aug. 1998.
- [3] C. Yim and M. A. Isnardi, "An efficient method for DCT-domain image resizing with mixed field/frame mode macroblocks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 5, pp. 696-700, Aug. 1999.
- [4] H. Paek, R.-C. Kim, and S. U. Lee, "On the POCS-based postprocessing technique to reduce the blocking artifacts in transform coded images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 3, pp. 358-367, June 1998.
- [5] S. H. Park and D. S. Kim, "Theory of projection onto narrow quantization constraint set and its applications," *IEEE Image Processing*, vol. 8, no. 10, pp. 1361-1373, Oct. 1999.
- [6] D. Pearson and M. Whybray, "Transform coding of images using interleaved blocks," *Proc. Inst. Elect. Eng.*, vol. 131, pp. 466-472, Aug. 1984.
- [7] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking artifacts," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 553-559, Apr. 1989.
- [8] B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 1258-1267, Oct. 1986.
- [9] A. Zakhor, "Iterative procedure for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 1, pp. 91-95, Mar. 1992.
- [10] Y. Yang, N. P. Galatanos, and A. K. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed image," *IEEE Trans.*

Circuits Syst. Video Technol., vol. 3, no. 6, pp. 421-432, Dec. 1993.

- [11] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Projection-based spatially adaptive reconstruction of block-transform compressed images," *IEEE Image Processing*, vol. 4, no. 7, pp. 896-908, July 1995.
- [12] Y. Jeong, I. Kim, and H. Kang, "A practical projection-based postprocessing of block-coded images with fast convergence rate," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 10, no. 4, pp. 617-623, June 2000.
- [13] A. Madisetti and A. N. Wilson, Jr., "A 100 MHz 2-D DCT/IDCT processor for HDTV applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 2, pp. 158-165, 1995.
- [14] N. I. Cho and S. U. Lee, "Fast algorithm and implementation of 2-D discrete cosine transform," *IEEE Trans. Circuits and Syst.*, vol. 38, no. 3, pp. 297-305, Mar. 1991.
- [15] R. Kresch and N. Merhav, "Fast DCT domain filtering using the DCT and the DST," *IEEE Trans. Image Processing*, vol. 8, no. 6, pp. 378-384, June. 1999.
- [16] 임창훈, "DCT 영역에서의 2차원 분리 대칭 선형 필터링," 한국통신학회 논문지, 28권, 5C호, pp.461-467, 5월, 2003년.

임창훈(Chang hoon Yim)

정회원



1986년 2월 서울대학교 제어계측공학과 학사

1988년 2월 한국과학기술원 전기 및 전자공학과 석사

1996년 12월 The University of Texas at Austin, Electrical and Computer Engineer-

ing 박사

1988년 3월~1991년 6월 한국방송공사 기술연구소 연구원

1996년 12월~1999년 3월 Sarnoff Corporation, 연구원

1999년 3월~2000년 7월 Lucent Technologies, Bell Labs, 연구원

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

2002년 5월~2003년 8월 삼성전자 디지털미디어연구소, 수석연구원

2003년 9월~현재 건국대학교 인터넷미디어공학부, 조교수

<관심분야> 멀티미디어 통신, 비디오 압축, 디지털 영상처리, 멀티미디어 네트워크