

# 정지영상의 Blocking Artifact를 제거하기 위한 후처리 알고리즘

## Postprocessing Algorithm for Reduction of Blocking Artifact in Still Images

장 동 식, 유 현 우, 손 용 준, 채 경 호  
(Dong-Sik Jang, Hun-Woo Yoo, Yong-Jun Sohn, Kyoung-Ho Chae)

**Abstract** : A new approach for reducing the blocking artifact is proposed in this paper. This method can be applied to conventional transform coding, without introducing additional information or significant blurring. Main drawbacks of the Discrete Cosine Transform(DCT) are visible block boundaries due to coarse quantization of the coefficients. Therefore, restoration techniques result in unnecessary blurring of the image. The proposed deblocking algorithm is based on the heuristic approach for edge detection. All edges of the standard still images are categorized, and the best edges are selected. Several images are used for experiments, and several other previous algorithms are compared with the proposed approach. The results show that the proposed algorithm works better than those of other previous researches. This algorithm can be used for JPEG, MPEG, and H.261 standard images.

**Keywords** : blocking artifact, deblocking, DCT, quantization, PSNR, JPEG

### I. 서론

지금까지 영상을 블록(block) 단위로 나누어서 부호화 하는 방법은 벡터 양자화, BTC (Block Truncation Coding)와 변환 부호화 같은 영상 압축 알고리즘에서 많이 사용했다. 그 가운데 DCT를 이용한 변환 부호화 방법은 주파수 영역에서 특성이 이상적인 변환 부호화 방법이 가장 가깝기 때문에 많이 쓰이는 방법이다. DCT를 이용한 알고리즘들은 영상을 8×8 픽셀 블록(block)으로 나누어서 각 블록에 대하여 독립적으로 DCT연산을 수행한다. 이러한 블록을 기본 단위로 하는 DCT는 블록 내부의 픽셀값은 상호 연관성이 제거되나 블록간의 연관성은 제거되지 않는 단점이 있다. 또한 주파수 영역에서 양자화과정으로 인하여 복원된 영상이 블록경계에서 블록킹(blocking) 현상이 발생한다. 이러한 블록킹 현상은 복원된 영상과 원 영상을 비교했을 때 시각적으로 왜곡된 영상으로 보이게 한다.

블록킹 현상을 줄이는 방법은 저주파 필터링, DPCM-DC 방법, 이웃 블록의 DC값들에서 AC값을 예측하는 방법, 불연속성을 최소화하는 방법, projection onto convex set (POCS) 등이 있다. 이러한 방법들은 이웃 블록들간의 연관성을 고려하지 못하는 DCT의 단점을 보완하는 방법이다. 먼저 저주파 필터링은 여분의 다른 정보 없이 복호기(decoder)에서 블록경계(block boundary)에서 발생하는 고주파 성분을 제거한다. 이 방법은 복원된 영상에서 발생하는 블록킹 현상은 줄일 수 있으나 영상이 퍼지개(blurring)되는 단점이 있다.

DPCM-DC방법은 DCT의 DC성분에 대해서 differential pulse code modulation(DPCM)을 적용하는 방법이다. 이 방법은 실제로 JPEG이나 MPEG, H.261, H.263에서 사용하고 있으나 DC성분 외에 다른 성분으로 인한 블록킹 현상은 제거

하지 못하고, 부호기(coder)와 복호기(decoder) 사이에 동기화가 이루어져야 하는 제약이 있다. 이웃 블록 DC값으로부터 잃어버린 AC값을 예측하는 방법은 효율이 떨어지는 단점이 있다. 최근에는 최적화 기법을 적용하여 이웃 블록과의 불연속성을 최소화함으로써 블록킹 현상을 최소로 하는 방법이 많이 소개되었다. 특히 이 방법은 최소화 과정에서 잃어버린 정보량을 복원한다는 점에서 다른 방법들과 다르다. 블록 사이의 불연속성은 블록간 경계에서 픽셀값들의 차이로 나타난다. [1]에서는 에지 성분으로 인한 불연속성을 수평방향의 이웃 픽셀간의 평균적인 분포를 이용하여 제거한 다음 불연속성을 줄이는 알고리즘이 제안되었다.

영상향상은 원래 영상을 처리하여 그 결과를 특별한 응용 목적에 맞게 고치는 것이다. 영상향상 접근 방식은 크게 공간 영역(spatial domain)방법과 주파수 영역(frequency domain) 방법으로 나눌 수 있다. 공간 영역은 영상 평면 그 자체를 의미하고, 접근 방식들은 영상에서 픽셀 단위의 처리를 기초로 한다. 주파수 영역의 처리기술은 영상의 푸리에 변환 수정을 근간으로 한다. 이 논문에서 제안한 블록킹제거(deblocking) 알고리즘은 공간 영역 방법이다.

이 논문에서 성능측정은 기존의 후처리방법(postprocessing)을 이용한 blocking artifact 감소 알고리즘을 이 논문에서 제안하는 blocking artifact 알고리즘과 비교 실험을 하여, 제안된 알고리즘의 성능은 복원된 영상이 시각적으로 얼마나 향상되었는가를 PSNR을 통하여 측정했다.

논문의 구성은 다음과 같다. 2장에서는 일반적인 영상압축 기법에 대해 알아보고 에지검출과 제안된 블록킹제거 알고리즘을 3장에서 살펴본다. 4장은 제안된 블록킹제거 알고리즘의 모수 추출과정에 대해 기술하고, 5장에서는 실제 영상이 시각적으로 향상되었는지 PSNR을 이용하여 기존 연구들에서 개발했던 알고리즘과 비교하여 살펴본다. 6장에서는 결론을 맺는다.

### II. 영상압축 기법

지금까지 개발된 방법들은 이웃 블록들간의 연관성을 고

러하지 못하는 DCT의 단점을 보완하는 방법들이다. 이러한 방법들은 부분적으로 시각적으로나 PSNR향상에서 성능이 좋아졌지만 고주파에서 퍼짐(blurring) 현상이 발생한다거나 계산 시간이 오래 걸린다거나 블록내의 영상이 일그러진다거나 하는 문제가 발생하여 한계점을 드러내왔다.

이 논문에서는 단순하면서도 시각적으로나 PSNR면에서 기존 방법들보다 뛰어난 성능을 나타내는 휴리스틱(heuristic) 기법을 활용한 새로운 알고리즘을 제안한다. 이 알고리즘은 간단한 개념에서 출발하여 휴리스틱기법으로 가장 효과적으로 에지를 찾아내고, 결과적으로 좋은 성능을 보였다.

영상 변환(image transformation)은 영상처리 분야에서 핵심적인 부분으로 영상 개선과 영상 복원, 부호화 과정에 기초가 된다. 먼저 이 절에서 JPEG 영상 압축 방식과 DCT, 양자화(quantization)과정에 대해서 알아보자.

가. JPEG 알고리즘[4]

JPEG 압축 전송 방식은 크게 압축·신장의 과정을 거쳐도 원래의 정보를 보존하는 무손실방법과 압축·신장의 과정에서 왜곡이 생겨 완전히 원래대로 재생되지 않는 손실로 나눌 수 있다.

나. DCT 부호화방식[5]

압축하는 측(encoder)에서 먼저 입력영상을 8×8 블록으로 분해한 수 블록 단위의 DCT 연산을 수행하고, 여기서 얻어진 DCT 계수를 DC(직류)성분과 AC(교류)성분으로 나누어 독립적으로 양자화테이블을 통해 양자화한다.

양자화된 DCT 계수 중 DC계수는, 바로 앞 블록의 DC계수를 예측치로 한 차분치를 부호화(encoding)한다. 그 나머지 AC 성분은 블록마다 지그재그 스캔에 의해 일렬로 나열한 후 부호화한다. 복호(decoding)는 부호화와 반대과정으로 영상을 재구성한다. 그림 1은 DCT 부호화, 복호화 방식 단계도이다.

다. DCT 변환[2]

2차원 DCT는  $u, v = 0, 1, 2, \dots, N-1$ 에 대해 다음과 같이 정의한다.  $u$ 는 블록의 행(row)을 나타내고,  $v$ 는 열(column)을 나타낸다.

$$C(u, v) = a(u)a(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)AB \quad (1)$$

여기서,

$$A = \cos\left[\frac{(2x+1)u\pi}{2N}\right]$$

$$B = \cos\left[\frac{(2v+1)v\pi}{2N}\right]$$

$$a(u) = \sqrt{\frac{1}{N}} \quad u = 0$$

$$a(u) = \sqrt{\frac{2}{N}} \quad u = 1, 2, 3, \dots, N-1$$

$$a(v) = \sqrt{\frac{1}{N}} \quad v = 0$$

$$a(v) = \sqrt{\frac{2}{N}} \quad v = 1, 2, 3, \dots, N-1$$

또한 역 DCT는  $x, y = 0, 1, 2, \dots, N-1$ 에 대해 아래와 같이 정의한다.  $x$ 는 block의 row를 나타내고,  $y$ 는 block의 column

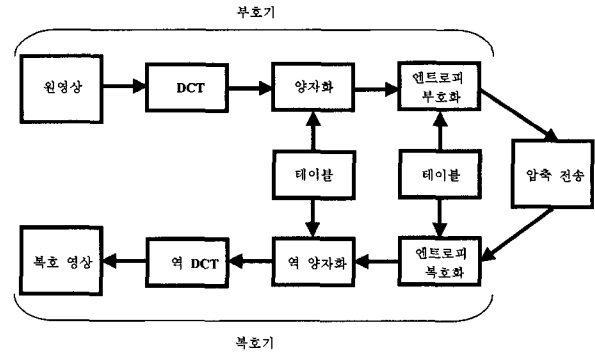


그림 1. DCT 부호, 복호화 방식 단계도.  
Fig. 1. Procedure of DCT encoding and decoding.

을 나타낸다.

$$f(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} a(u)a(v)C(u, v)AB \quad (2)$$

여기서,

$$A = \cos\left[\frac{(2x+1)u\pi}{2N}\right]$$

$$B = \cos\left[\frac{(2v+1)v\pi}{2N}\right]$$

III. 제안한 알고리즘

이 논문에서 제시하는 블록경계 알고리즘의 기본 개념은 다음과 같다.

- 1) JPEG 표준으로 네트워크를 통해 전송한 영상의 복원한 픽셀값을 추출한다.
- 2) 복원한 영상의 8×8 블록별로 인접한 블록 경계 픽셀값을 추출한다.
- 3) 각 블록의 column, row 별로 인접 경계의 픽셀값의 차이를 구한다.

$$d_r = |f(i, j+7) - f(i, j+8)|$$

$$i = 0, 1, 2, \dots, n-1$$

$$j = 0, 1, 2, \dots, \frac{n-1}{8}$$

$$d_c = |f(i+7, j) - f(i+8, j)|$$

$$i = 0, 1, 2, \dots, \frac{n-1}{8}$$

$$j = 0, 1, 2, \dots, n-1$$

여기서  $d_r$ 과  $d_c$ 는 각각 블록 경계의 차이값 행벡터와 열벡터를 나타낸다.  $f(i, j+7), f(i, j+8)$ 은 차이값 행벡터의 왼쪽과 오른쪽 8×8 블록의 픽셀값을 나타내고,  $f(i+7, j), f(i+8, j)$ 는 차이값 열벡터의 위와 아래의 8×8 블록의 픽셀값을 나타낸다.

4) 인접 블록 경계 픽셀값의 차이가 휴리스틱 기법을 적용하여 찾은값 가운데 하한( $d_l$ ) ~ 상한( $d_h$ ) 사이에 있어야 한다.

$$\begin{aligned} d_l \leq d_r \leq d_h \\ d_l \leq d_c \leq d_h \end{aligned} \quad (5)$$

5) 두 인접 경계 픽셀값 가운데 작은 쪽에 두 픽셀값 차이에 대해  $\alpha$  등분한 값을 더해 준다.

만약  $f(i, j+7) < f(i, j+8)$  이고  $i=0, 1, 2 \dots n-1$   
 $j=0, 1, 2 \dots \frac{n-1}{8}$  이면

$$f'(i, j+7) = f(i, j+7) + \frac{d_r}{\alpha} \quad (6)$$

이고,  
 $f(i, j+7) > f(i, j+8)$  이고  $i=0, 1, 2 \dots n-1$   
 $j=0, 1, 2 \dots \frac{n-1}{8}$  이면

$$f'(i, j+8) = f(i, j+8) + \frac{d_r}{\alpha} \quad (7)$$

이다.

여기서,  $f'(i, j+7), f'(i, j+8)$ 은 블록의 새로운 인접 경계의 픽셀값이다.

이런 방법으로 열벡터에 대해서도 똑같이 적용한다.

6) 두 인접 경계 픽셀값 가운데 큰 쪽에 두 픽셀값 차이에 대해  $\alpha$  등분한 값을 빼 준다.

만약  $f(i, j+7) < f(i, j+8)$  이고  $i=0, 1, 2 \dots n-1$   
 $j=0, 1, 2 \dots \frac{n-1}{8}$  이면

$$f'(i, j+8) = f(i, j+8) - \frac{d_c}{\alpha} \quad (8)$$

이고,  $f(i, j+7) > f(i, j+8)$  이고  $i=0, 1, 2 \dots n-1$   
 $j=0, 1, 2 \dots \frac{n-1}{8}$  이면

$$f'(i, j+7) = f(i, j+7) - \frac{d_c}{\alpha} \quad (9)$$

이다.

여기서,  $f'(i, j+7), f'(i, j+8)$ 은 블록의 새로운 인접 경계의 픽셀값이다.

이런 방법으로 열벡터에 대해서도 똑같이 적용한다.

7) 수정된 픽셀값으로 영상을 재구성한다.

이 논문에서 제시하는 알고리즘은  $8 \times 8$  블록 단위로 DCT와 양자화를 함으로서 발생하는 블록킹현상을 가장 단순하게 줄이는 방법을 사용하였다. 먼저 휴리스틱 기법을 적용하여 매개변수(parameter)를 추정하여 원 영상에서 예지부분을 찾아냈다. 실험결과 인접 픽셀간 차이 가운데 상한( $d_h$ )이 70 이상일 때 알고리즘 결과 값이 최대였다. 따라서  $d_h$ 가 70 이상이면 예지로 인식했다. 또한 인접 블록경계 사이의 차를

삼등분해서 블록간의 차이를 줄이기 때문에 하한( $d_l$ )이 3 미만일 때는 의미가 없어 두 블록 인접 픽셀간 차이가 3 이상일 때만 적용했다. 이러한 각 행벡터와 열벡터값 차이( $d_r, d_c$ )가  $d_l$ 에서  $d_h$  사이에 있으면 블록킹제거 알고리즘을 적용하고 그렇지 않으면 원래 복원한 영상의 픽셀값을 그대로 적용한다.

그림 2는 블록킹 제거 알고리즘을 적용했을 때 인접 경계 픽셀값의 변화를 그래프로 표현한 것이다.

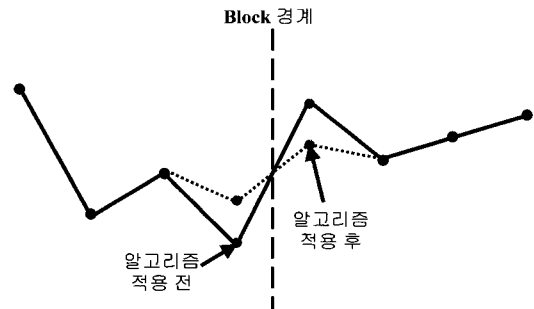


그림 2. Blocking Artifact 감소 개념도.  
 Fig. 2. Reducing of blocking Artifact.

그림 3은 설명한 블록킹제거 알고리즘의 개요도이다.

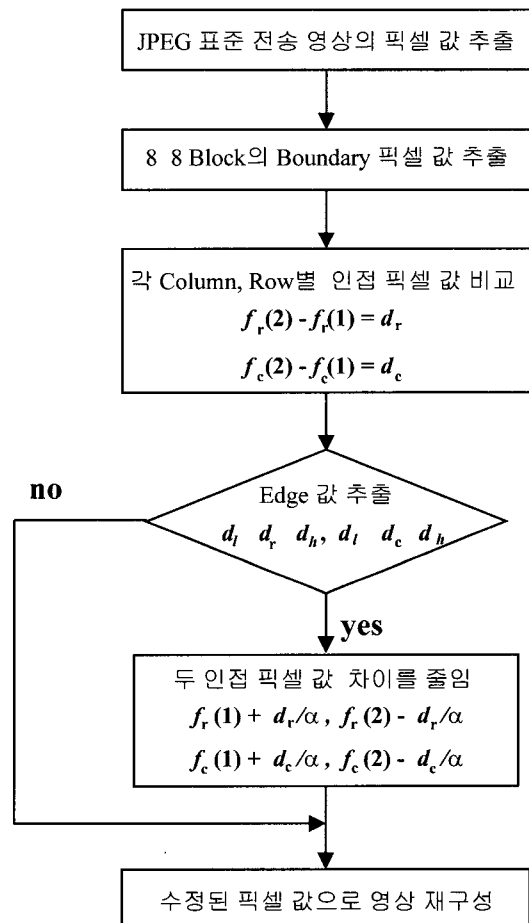


그림 3. DCT 부호, 복호화 방식 단계도.  
 Fig. 3. Deblocking algorithm.

**IV. 제안 알고리즘의 모수 도출**

이 논문에서 제안한 블록킹제거 알고리즘은 512×512 크기의 “Lena”, “Peppers”, “Airplane”을 대상으로 실험을 실시하였다. 실험에서 정지영상은 JPEG 표준에 의거하여 “Lena”는 0.232bpp(bit per pixel), “Peppers”는 0.218 bpp 그리고 “Airplane”은 0.280 bpp로 압축된 영상을 사용하였다. 각 영상은 크기에 따른 표현의 제약으로 중심 부분 256 x 256 만을 표시하였다.

우선 각 블록 경계 사이에서 에지를 배제하기 위해 휴리스틱 접근 방법을 통해 가장 적절한 에지를 찾아냈다. 이러한 에지에 제외한 실험결과는 표 1와 같다. 표에서 보듯이 각 에지 범위들 가운데 표준 영상 3개에 대해  $d_l$ 이 3에서  $d_h$  70사이까지 가장 좋은 실험 결과를 얻을 수 있었다. 현재 모든 종류의 표준 영상에서도 인접 픽셀값 차이가 3~70사이에서 알고리즘을 적용할 때 가장 좋은 결과를 얻을 수 있는 지를 연구중이다. 또한 각 블록 단위별로 블록킹 현상을 줄이기 위한 방법으로 인접 경계 1개, 2개, 3개에 대해 실험을 했다. 결과적으로 경계 픽셀값은 서로 인접한 1개 픽셀만 적용했을 때가 가장 좋은 복원영상을 만들었다. 그 결과는 그림 4와 같다.

그림 4에서 보듯이 row나 column 별로 알고리즘을 적용했을 때 인접한 하나의 row나 column을 적용하는 것이 가장 좋은 결과값을 나타냈다. 또한 이 알고리즘에서 두 인접 픽셀값의 차이를 삼등분하는 것이 사등분이나 이등분에 비해 더 좋은 결과를 나타냈다. 그 결과는 다음 그림 5와 같다.

표 1. Edge를 제외한 알고리즘 결과(PSNR).

Table 1. Algorithm result after eliminating of edge.

표준영상 에지 제외 범위(하한~상한)	Lena (0.232bpp)	Peppers (0.218bpp)	Airplane (0.280bpp)
JPEG 표준방식	30.88	31.29	29.84
40 ~ 50	30.83	31.32	29.82
40 ~ 70	30.88	31.32	29.87
40 ~ 90	30.90	31.31	29.88
30 ~ 50	30.91	31.37	29.90
30 ~ 70	30.96	31.37	29.95
30 ~ 90	30.98	31.35	29.96
20 ~ 50	31.02	31.45	30.01
20 ~ 70	31.07	31.45	30.07
20 ~ 90	31.10	31.44	30.07
10 ~ 50	31.17	31.59	30.17
10 ~ 70	31.21	31.59	30.22
10 ~ 90	31.23	31.58	30.23
3 ~ 50	31.26	31.68	30.19
3 ~ 70	31.32	31.68	30.25
3 ~ 90	31.32	31.67	30.25

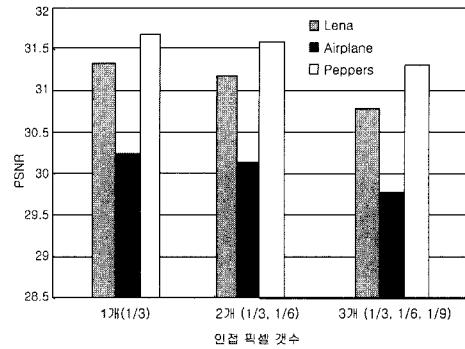


그림 4. 인접 Boundary별 PSNR 비교.

Fig. 4. Comparison of PSNR according to different number of boundary pixels.

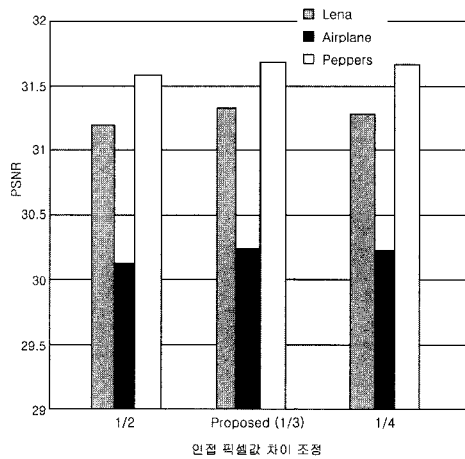


그림 5. 인접 픽셀값 차이 조정에 따른 PSNR 결과.

Fig. 5. PSNR result after turning number of boundary pixels.

**V. 성능평가**

제안한 알고리즘은 Yang[3]과 Zakhor[1]가 제안한 알고리즘과 성능을 비교하였다. 이 논문에서 제안한 블록킹제거 알고리즘으로 복원한 영상과 Yang과 Zakhor가 제안한 알고리즘으로 복원한 영상을 비교한 결과, 사람의 시각으로는 거의 구분하지 못할 정도로 비슷한 결과가 나왔다. 하지만 PSNR을 기준으로 비교한 결과는 이 논문에서 제안한 알고리즘이 뛰어난 결과를 나타내었다.

표 2는 이러한 실험 결과에 대해 PSNR을 기준으로 비교한 데이터로서 기존의 방법들과 비교한 결과를 나타냈다.

표 2. 기존 알고리즘과 PSNR 성능비교.

Table 2. PSNR comparison between proposed method and other related.

Image ( bpp )	PSNR(dB)			
	JPEG	Zakhor	Yang	Proposed
Lena ( 0.232 )	30.78	30.57	31.06	31.32
Airplane ( 0.280 )	29.77	29.09	29.74	30.24
Peppers ( 0.218 )	31.29	31.41	31.54	31.68

그림 6은 실험에 적용한 "Lena" 원 영상, 0.232 bpp로 압축한 JPEG 영상과 이 논문에서 제안한 블록킹제거 알고리즘을 이용하여 블록화 현상을 제거한 영상이다. 또한 그림 7은 실험에 적용한 "Airplane" 원 영상, 0.280 bpp로 압축한 JPEG 영상과 이 논문에서 제안한 블록킹제거 알고리즘을 적용한 영상이고, 그림 8은 실험에 적용한 "Peppers" 원 영상, 0.218 bpp로 압축한 JPEG 영상과 이 논문에서 제안한 블록킹제거 알고리즘을 적용한 영상이다. 그림에서 보듯이 JPEG 영상과 비교하여 블록킹 현상이 줄어들었고, PSNR 측면에서도 표에서 보듯 많이 향상됐다.

최근에 DWT(discrete wavelet transform)을 이용하여 압축 표준으로 각광받고 있는 JPEG2000에서는 "Lena" 영상(512×512)에 대해 0.125 bpp일 경우는 31.04 PSNR을, 0.25 bpp일 경우는 34.14 PSNR을 기록했다[6].

VI. 결론

본 논문은 정지영상을 압축 전송하는 과정에서 발생하는 블록킹 Artifact를 줄이는 알고리즘을 제시했다. 정지영상을 JPEG표준으로 discrete cosine transform (DCT)으로 전송했을 때 발생하는 블록킹 artifact를 줄이기 위해 휴리스틱 기법을 적용하여 에지를 찾아내고 블록의 인접 경계 픽셀값들에만 단순한 블록킹제거 알고리즘을 제시한다.

이 논문에서 제안한 알고리즘은 블록 경계에서 발생하는 블록킹 artifact를 row와 column 단위의 차이를 계산하여 각 경계 픽셀 단위로 보상함으로써, 무리한 양자화로 인하여 발생한 왜곡을 완화시켜 시각적인 측면에서 복원한 영상의 품질을 전반적으로 향상시킬 수 있었다. 또한 PSNR 면에서 제안한 알고리즘은 지금까지 기존에 개발했던 알고리즘들보다 뛰어난 성능을 나타냈다. 특히 제안한 블록킹제거 알고리



(a)



(b)



(c)

그림 6. Lena 영상 실험 결과 (a) Lena 512×512 원 영상 (b) Lena JPEG 영상(0.232 bpp) (c) 제안한 알고리즘 적용한 Lena 영상.

Fig. 6. Experimental results using Lena image (a) 512×512 Lena image (b) Lena JPEG image(0.232 bpp) (c) Lena image after applying proposed method.



(a)



(b)

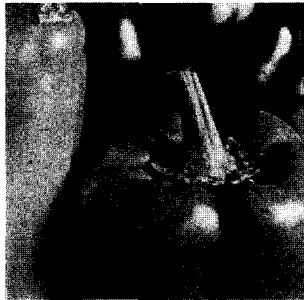


(c)

그림 7. Airplane 영상 실험 결과 (a) Airplane 512× 512 원 영상 (b) Airplane JPEG 영상(0.280 bpp) (c) 알고리즘 적용한 Airplane영상.

Fig. 7. Experimental results using Airplane image (a) 512×512 Airplane image (b) Airplane JPEG image (0.280 bpp) (c) Airplane image after applying proposed method.

즘은 간단한 아이디어에서 출발하여 구성 자체가 단순할 뿐만 아니라, DCT 계수와 같은 추가 정보 없이 압축 전송 후 복원된 영상만을 사용하기 때문에 계산 량이 적어 매우 효율적이다. 따라서 이 블록킹 제거 알고리즘은 JPEG은 물론 MPEG, H.261, H.263에 후처리(postprocessing) 기법으로 적용할 수 있다.



(a)



(b)



(c)

그림 8. Peppers 영상 실험 결과 (a) Peppers 512× 512 원 영상 (b) Peppers JPEG 영상(0.218 bpp) (c) 알고리즘 적용한 Peppers 영상.

Fig. 8. Experimental results using Peppers image (a) 512× 512 Peppers image (b) Peppers JPEG image(0.218 bpp) (c) Peppers image after applying proposed method.

참고문헌

[1] Shigenobu Minami and Avid Zakhor, "An Optimization Approach for Removing Blocking Effects in Transform Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol 5, no. 2, pp. 74-82, April, 1995.  
 [2] Gonzales & Woods, "Digital Image Processing," Addison Wesley Longman, 1992.  
 [3] Yongyi Yang, Nikolas P. Galatsanos, and Aggelos K. Katsaggelos, "Projection-Based Spatially Adaptive Reconstruction of Block-Transform Compressed Images," *IEEE Transaction on Image Processing*, vol 4, no.7, pp. 896-908, July, 1995.  
 [4] 후지와라 히로시, 정제창(역), "최신 MPEG," 교보문고, 1995.  
 [5] 천인국, 윤영택, "기초편 영상처리," 기한재, 1998.  
 [6] D. Taubman, E. Ordentlich, M. Weinberger, G. Serrousi, "Embedded Block Coding in JPEG2000," Tech Report; HPL-2001-35.



손 용 준

1957년 7월 7일생. 1980년 서울공대 자원 공학과 졸업, 동대학 산업공학 석사(1983), Texas A&M University 박사과정 수료(1988), 고려대학교 산업시스템정보 공학 박사과정 수료(2001). 관심분야는 컴퓨터비전, 스케줄링, 머신비전.



채 경 호

1967년 12월 2일생. 1990년 공군사관학교 산업공학과 졸업, 현재 고려대학교 석사과정, 관심분야는 통신망을 이용한 영상 전송과 영상처리

장 동 식

제어·자동화·시스템공학 논문지 제7권, 제4호, 참조.

유 현 우

제어·자동화·시스템공학 논문지 제7권, 제4호, 참조.