
중첩 기반 연산과 Hanning Window를 이용한 블록 불연속 노이즈 방지 알고리즘

김주현* · 장원우** · 박정환*** · 양훈기**** · 강봉순*****

Algorithm to prevent Block Discontinuity by Overlapped Block and Hanning Window

Joohyun Kim* · Wonwoo Jang** · Junghwan Park*** · Hoongee Yang**** · Bongsoon Kang*****

이 논문은 2006년 삼성전기 지원을 받아 수행된 연구임

요 약

본 논문은 블록 처리 방법을 기반으로 하는 링잉 노이즈 감소 알고리즘을 사용할 때, 블록 불연속 노이즈(Block Discontinuity)를 방지 할 수 있는 중첩 기반(Overlapped Block) 연산과 Hanning Window에 관련된 것이다. 링잉 노이즈 감소 알고리즘은, 24bit RGB와 블록 기반 연산으로 하며, 수정된 K-means 알고리즘을 바탕으로 한다. 그래서 제안한 중첩 기반 연산은 입력 영상을 여러 단위 블록으로 조각낼 때, 단위 블록의 크기의 반을 중첩 시켜 선택하는 방법이다. 16×16 픽셀 크기의 데이터 블록을 데이터 유닛(Data Unit)이라고 정의 하였다. 그 후 처리된 데이터 유닛들을 등방성 분포를 지닌 Hanning Window를 사용하여 중첩된 데이터에서 원 이미지 형태로 복원하였다. 최종적으로 언급된 알고리즘의 성능을 확인하기 위해서 링잉 노이즈를 가진 이미지를 기존 방법(비 중첩 기반 연산)과 제안한 알고리즘으로 처리함으로써 각각의 결과를 비교하였다.

ABSTRACT

In this paper, we propose an Overlapped Block and an Hanning Window to prevent a Block Discontinuity when we use an algorithm to eliminate ringing artifact which is based on a block structure. The algorithm to eliminate ringing artifact operates with a block structure and 24-RGB data and is based on a modified K-means algorithm. The proposed overlapped block method is piled up one on another per an half of the size of unit-block when an input image is split into several unit-blocks. Therefore, we define a data unit as the unit-block of the block size, 16 × 16 pixels. We reconstruct the processed data units into the original form of input image by using an isotropic form of Hanning Window. Finally, in order to evaluate the performance of the abovementioned algorithms, we compare three image, an input image with ringing artifact, an image result obtained by conventional method (non-overlapped), and an image result obtained the proposed method.

키워드

Block Discontinuity, Ringing Artifact, Overlapped Block, Hanning Window, and K-means algorithm

* 삼성전기 중앙연구소 책임연구원

** 동아대학교 전자공학과 박사과정

*** 동아대학교 전자공학과 석사과정

**** 광운대학교 전자공학과 교수

***** 교신저자: 동아대학교 전자공학과 부교수

I. 서론

압축(Compression)과 보간법(interpolation)은 많은 영상 장치인 모바일 카메라 폰, 디지털 카메라, TV에 사용된다. 이러한 장치들은 압축된 영상 신호를 사용하기 때문에, 압축된 영상을 복원하기 위해서 보간법을 사용하게 된다. 보간법이라는 것이 주변의 값들을 이용하여 새로운 데이터를 생성하는 방법이다 [1]. 그러나 압축률이 높을수록, 원본 데이터의 정보가 작기 때문에, 이전의 이미지를 복원하기는 어려우며, 또한 이러한 과정에서 영상 노이즈가 발생하게 된다. 이러한 현상 중에 대표적인 것이 링잉 노이즈(Ring Artifact)이다 [2]. 데이터의 양자화(Quantization)에 의해 생성된 잡음 성분이 확대되어, 영상의 예리한 모서리 근방에서 진동성 무늬가 나타나게 된다. 이러한 링잉 노이즈를 제거하기 위한 방법을 16×16 크기의 블록 처리 방법을 이미지에 적용시켰다. 비디오 압축 표준 중에 하나인 H.26x와 MPEG-1/2/4는 블록 기반의 처리방법을 사용한다 [2]. 그러나 독립적으로 처리된 각 블록을 그대로 출력 시킬 경우에 JPEG 및 MPEG과 같은 블록 기반 압축 방법에서 발생하는 블록 간의 경계에서 Discontinuity 문제가 발생하였다. 링잉 노이즈는 제거 되었으나, 이미지에 블록 불연속(lock Discontinuity)이 나타나게 된 것이다. 그래서 본 논문에서 제안한 방법은 중첩 블록(Overlapped Block) 구조로 동작하며, Hanning Window를 이용한 이미지 복원 방법을 제안하고자 한다.

II. 기본 알고리즘

이미지의 링잉 노이즈 감소를 위하여, CRA 알고리즘을 사용할 때 블록화 현상을 방지할 수 있는 중첩기반 연산과 Hanning Window 적용 시, 블록 불연속 노이즈가 없 는 영상을 위한 알고리즘을 제안한다.

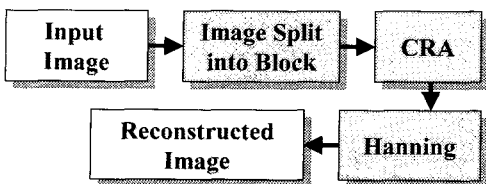


그림 1. 제안한 알고리즘의 블록도
Fig. 1. Block Diagram of Proposed Algorithm

이 알고리즘은 3가지 단계로 구성되며, 24bit RGB와 블록 기반 연산으로 한다. 이미지를 조각내어 블록화 시키는 Image Split into Block 단계는 이미지 분할 방법인 비 중첩 블록(Non-overlapped Block) 연산과 중첩 블록(Overlapped Block) 연산 방법이 있다. 비 중첩 블록 연산은 독립 블록 기반 압축 방법에서 블록 간 경계에서 Discontinuity 문제가 발생할 수 있다 [3]. Clear Ring Artifact(CRA) 단계는 이미지가 퍼져 보이는 링잉 노이즈를 제거하기 위한 단계이다. 그리고 Hanning 단계는 블록화 되어 있는 이미지를 다시 복원하는 단계이다. 중첩 블록 연산과 블록 크기와 같은 등방성(Isotropic) 형태의 계수 분포를 가지는 Hanning Window 이용한다.

2.1 Overlapped Block

본 논문에서 제안하는 기법은 중첩 블록(Overlapped Block) 구조 기반이다. 입력 영상에서 데이터 블록으로 조각내고 그것의 절반을 지속적으로 중첩시켜 선택 후 독립적으로 처리 한다. CRA를 수행하기 위해선 입력 데이터들을 16×16 픽셀 크기의 블록으로 나누어야 한다. 그래서 16×16 픽셀 크기의 데이터 블록을 데이터 유닛(Data Unit) 이라고 우선 정의한다. 그림 2는 비 중첩 블록(Non-overlapped Block)과 중첩 블록(Overlapped Block) 연산이 적용되는 과정이다. 그림 2-(a)는 그림 2-(b), Lenna 이미지의 가운데 직사각형 영역이 확대 시켜서 비 중첩 블록 연산 과정 적용 과정을 보여주는 개념도이다. 널리 사용되는 JPEG, MPEG과 유사한 블록기반 연산을 하는 알고리즘이기 때문에 데이터 유닛마다 독립적으로 처리된다. 서로 간에 중첩이 되는 부분이 없이 왼쪽에서 오른쪽으로 위에서 아래로 가면서 이미지를 분할 선택한다. 데이터 유닛의 크기가 16×16 픽셀이기 때문에, 그림 2-(a)의 ①, ②, ③, ④, ⑤, ⑥은 이미지에서 분할로 분할 선택되었다.

그림 2-(c)는 그림 2-(b)의 확대 된 일부분을 사용하여 중첩 블록 연산이 적용되는 과정의 개념도이다. 데이터 유닛의 크기가 16×16 픽셀이기 때문에, 인접 블록과는 8 픽셀 만큼 중첩시킨다. ① 데이터 유닛을 선택한 후, 다음 처리 영역을 ③ 데이터 유닛 아니라, ② 데이터 유닛을 선택한다. 인접 8 픽셀 중첩된 영역, 즉 데이터 유닛의 크기의 1/2 크기를 중복 하여 처리한다. [4] 가로 부분으로 이미지 데이터가 다 처리되면 ⑥ 데이터 유닛처럼 위 아래로 8 픽셀 중첩된 영역이 선택된다. 서로 간에 중첩

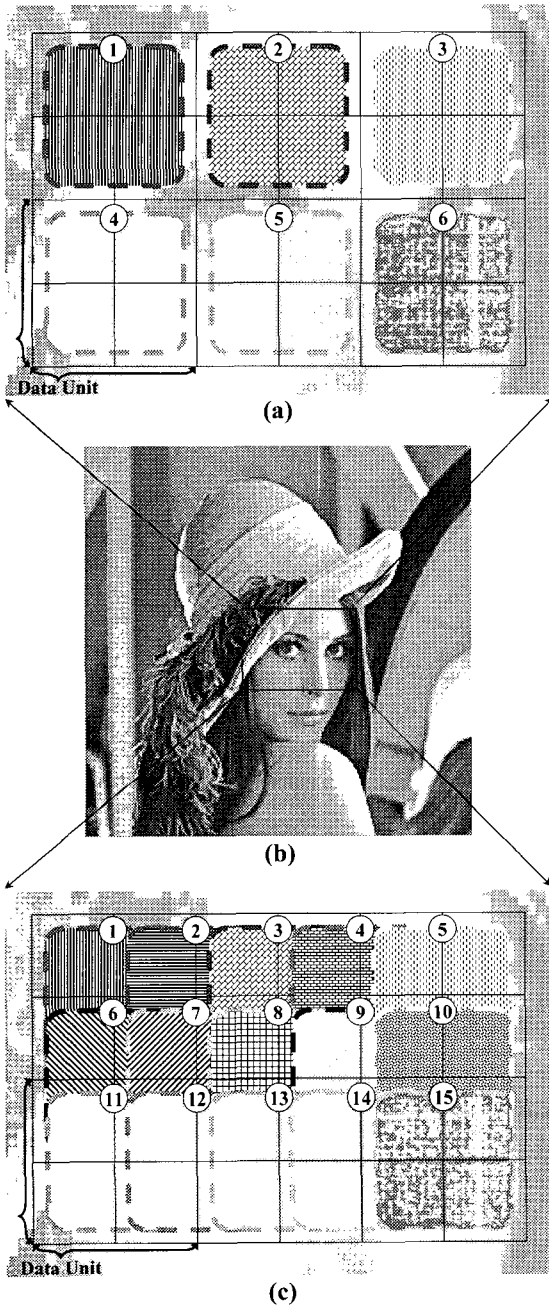


그림 2. (a) Non-overlapped Block 연산 개념도, (b) Lenna 이미지 (256x256), (c) Overlapped Block 연산 개념도

Fig. 2. (a) Diagram of the Non-overlapped Block Processing, (b) Original Image of Lenna (256x256), (c) Diagram of the Overlapped Block Processing

이 되는 부분을 가지면서, 이미지의 위에서부터 아래로, 왼쪽으로부터 오른쪽으로 가면서 이미지를 분할 선택한다.

2.2 CRA

CRA은 수정된 K-means 알고리즘을 이용하여 16x16 픽셀 크기의 각 데이터 유닛 마다 링잉 노이즈(Ringing Artifact)를 제거한다. K-means 알고리즘은 군집화(Clustering)를 위한 알고리즘으로서, 정해진 영역의 데이터들을 K개의 군집으로 나눈다. 이를 분류하기 위한 기준으로서 임의의 K개의 초기 중심점들을 선정한다. 선정된 K개의 초기 중심점들과 영역내의 데이터들 간의 거리 값 가운데서 가장 작은 초기 중심점으로 데이터를 할당한다. 그리고 영역내의 데이터 점들이 모두 다 처리되고 나면 각 K개의 그룹이 생성되며, 각 그룹의 소속된 데이터들의 좌표 값의 산술 평균(Arithmetic Average)를 하여 K개의 새로운 중심점을 만들어 낸다. 위에서 언급된 중심점과 데이터와의 최소 거리 비교 방법을 반복적으로 수행하여, 이전 단계에 생성된 K개의 중심점과 거리의 차이가 없을 때 까지 수행한다 [5-6]. K-means 알고리즘을 사용하기 위해서 반복 횟수를 1회로 한정하였으며, 군집의 개수를 2개 (K=2)로 수정하였다. 왜냐하면 최종점을 찾을 때 까지 많은 시간이 소요되기 때문이다. 또한 초기 중심점을 데이터 유닛 내의 픽셀 중에서 최대 및 최소의 휘도 값을 가지는 픽셀로 하였다. 이러한 수정된 K-means 알고리즘을 이용하여 데이터 유닛 내에 픽셀들을 두 그룹으로 나누고, 각 그룹의 산술 평균 값인 2개의 대표 값을 구한다. 데이터 유닛 안의 모든 픽셀들을 최단 거리에 있는 새로운 중심점으로 이동시켜서, 링잉 노이즈를 제거하게 된다 [7].

2.3 Hanning Window

CRA의해 독립적으로 처리된 16x16 픽셀 크기의 데이터 유닛을 Hanning Window를 이용하여 이웃 데이터 블록 경계에서 smooth transition이 가능하도록 한다. 이 과정은 이미지 전체를 일반적인 필터와 개념이 다르다. 그림 3는 인접한 두 개의 데이터 유닛 간 경계에서 Hanning Window를 이용한 개념도이다.

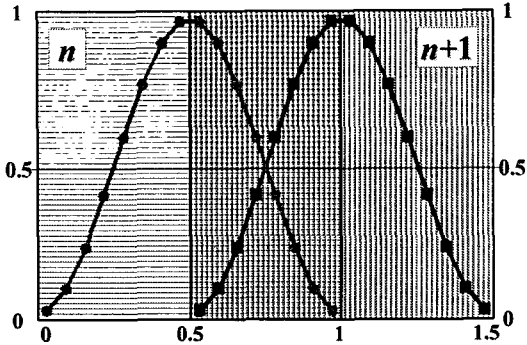


그림 3. Hanning Window를 Discontinuity 개념도
Fig. 3. Diagram to eliminate the Discontinuity by using Hanning Window

그림 3의 원:○그래프와 정사각형:□ 그래프는 Hanning Window 수식에 의해 만들어진 계수 값들의 분포이며 이것들은 동일하다. 각 데이터 유닛에 곱해지는 Window 계수는 중첩된 부분을 중심으로 한쪽은 경계 방향으로 점차 낮아지고, 다른 한쪽은 경계 방향에서부터 점차 높아지는 형태로 분포하게 된다. 그리고 중첩된 부분에서 두 개의 Window 계수들의 합은 항상 1이 되도록 한다. 독립적으로 처리된 데이터 유닛 n과 n+1에 Hanning Window의 계수를 곱한 후, 데이터 유닛에서 같은 위치에 있는 픽셀들을 서로 더하여 최종 결과를 구하게 된다.

영상 이미지의 특성상 2차원 Window가 필요하다. 아래의 내용은 2차원 Hanning Window를 만드는 과정을 보여준다. 우선 수식 (1)을 사용하여 1차원 Hanning Window 계수들을 만들어 낸다.[8]

$$Wini = 8 \times \left(1 - \cos\left(\frac{2\pi}{N} \times i + k\right)\right) \quad (1)$$

$$i = 1, 2, \dots, N$$

이미 데이터 유닛이라는 블록의 크기를 16×16으로 정하였기 때문에, 수식 (1)에서 데이터 유닛의 크기와 동일한 크기의 계수들의 분포를 위해 N=16으로 한다. 또한 중첩된 부분에서 계수들의 합이 같도록 하기 위해서, 대칭의 구조를 이루는 cosine 함수를 사용하였으며, 상수 k를 사용하여 계수들을 균일하게 하였다. 1차원 16개의 Hanning Window 계수들을 2차원 모양의 16×16으로 만들기 위해서 수식 (2)의 방법을 사용한다.

$$Window = Wini^T \times Wini \quad (2)$$

수식 (2)의 결과 행렬 16×16 Hanning Window의 값은 그림 4와 같다.

Hanning Window =

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	4	6	10	12	15	16	16	15	12	10	6	4	1	0
0	4	16	24	40	48	60	64	64	60	48	40	24	16	4	0
0	6	24	36	60	72	90	96	96	90	72	60	36	24	6	0
0	10	40	60	100	120	150	160	160	150	120	100	60	40	10	0
0	12	48	72	120	144	180	192	192	180	144	120	72	48	12	0
0	15	60	90	150	180	225	240	240	225	180	150	90	60	15	0
0	16	64	96	160	192	240	256	256	240	192	160	96	64	16	0
0	16	64	96	160	192	240	256	256	240	192	160	96	64	16	0
0	15	60	90	150	180	225	240	240	225	180	150	90	60	15	0
0	12	48	72	120	144	180	192	192	180	144	120	72	48	12	0
0	10	40	60	100	120	150	160	160	150	120	100	60	40	10	0
0	6	24	36	60	72	90	96	96	90	72	60	36	24	6	0
0	4	16	24	40	48	60	64	64	60	48	40	24	16	4	0
0	1	4	6	10	12	15	16	16	15	12	10	6	4	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

그림 4. 16×16 행렬 크기의 Hanning Window
Fig. 4. The Hanning Window of the 16×16 matrix

Hanning Window의 기본 공식이 cosine을 사용하기 때문에 수식 (1)에서 의해 생성된 1×16의 데이터들은 서로 대칭을 이루는 구조이다. 또한 1차원을 바탕으로 2차원 16×16을 만들었기 때문에 그림 4의 계수들의 값의 크기는 등방성(isotropic)의 분포 형태를 지닌다. 그림 5은 중첩 블록 처리로 발생한 데이터 유닛을 Hanning Window를 이용하여 복원하는 과정을 보여 준다.

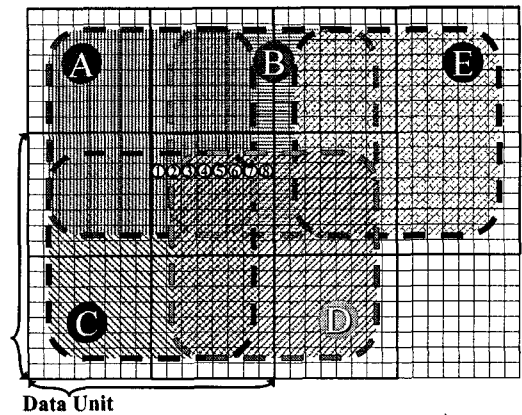


그림 5. Hanning Window를 이용한 Overlapped Block 연산의 개념도
Fig. 5. Diagram of the Overlapped Block by using Hanning Window

우선 16×16 데이터 유닛 ①, ②, ③, ④, ⑤ 5개를 처리한다고 가정해 보자. 8×8 영역이 4개의 Data Unit들이 서로 겹치게 된다. 그래서 이 겹쳐진 부분의 ①, ②, ③, ④, ⑤, ⑥, ⑦, ⑧ 위치의 픽셀들이 처리되는 과정을 서술하고자 한다. 수식 (2)에 의해 만들어진 Hanning Window의 계수들의 값을 해당 위치에 포개듯이 곱한다. 그림 6은 각각의 입력 블록마다 Hanning Window의 계수가 어떻게 곱해지는 과정을 보여준다.

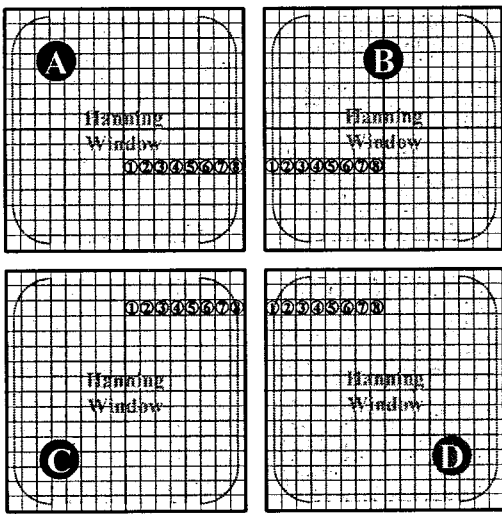


그림 6. Hanning Window가 각 데이터 유닛마다 곱의 연산 개념도

Fig. 6. Diagram of the product of each Data Unit with Overlapped Block and Hanning Window

그림 6의 4가지 경우처럼, Hanning Window의 계수들이 각 데이터 유닛에 곱해지는 위치가 서로 다르다. 그러나 Hanning Window의 계수의 등방성 분포 때문에, 곱해지는 계수의 합은 항상 256이 된다. 그래서 계수들을 각각의 픽셀마다 곱해서 서로 중첩되는 4개의 부분을 각각 더하고, 출력에서 256을 나누면 4개의 중첩되는 부분이 모두 고려된 최종 출력을 얻을 수 있다. 아래의 수식들은 그림 5의 중첩된 영역의 데이터가 저장되는 과정을 보여준다.

$$\textcircled{1} = A_1 \times \frac{192}{256} + B_1 \times \frac{0}{256} + C_1 \times \frac{64}{256} + D_1 \times \frac{0}{256} \quad (3)$$

$$\textcircled{2} = A_2 \times \frac{180}{256} + B_2 \times \frac{12}{256} + C_2 \times \frac{60}{256} + D_2 \times \frac{4}{256} \quad (4)$$

$$\textcircled{3} = A_3 \times \frac{144}{256} + B_3 \times \frac{48}{256} + C_3 \times \frac{48}{256} + D_3 \times \frac{16}{256} \quad (5)$$

$$\textcircled{4} = A_4 \times \frac{120}{256} + B_4 \times \frac{72}{256} + C_4 \times \frac{40}{256} + D_4 \times \frac{24}{256} \quad (6)$$

$$\textcircled{5} = A_5 \times \frac{72}{256} + B_5 \times \frac{120}{256} + C_5 \times \frac{24}{256} + D_5 \times \frac{40}{256} \quad (7)$$

$$\textcircled{6} = A_6 \times \frac{48}{256} + B_6 \times \frac{144}{256} + C_6 \times \frac{16}{256} + D_6 \times \frac{48}{256} \quad (8)$$

$$\textcircled{7} = A_7 \times \frac{12}{256} + B_7 \times \frac{180}{256} + C_7 \times \frac{4}{256} + D_7 \times \frac{60}{256} \quad (9)$$

$$\textcircled{8} = A_8 \times \frac{0}{256} + B_8 \times \frac{192}{256} + C_8 \times \frac{0}{256} + D_8 \times \frac{64}{256} \quad (10)$$

위의 수식에서, A1은 데이터 유닛 ①의 ① 위치의 픽셀을 나타낸다. 이렇게 각 블록마다 가중치를 달리 해주려는 이유는 데이터 유닛이 독립적으로 연산되었기 때문에 절대적 위치가 같은 픽셀의 값이 다르기 때문이다. 이러한 이유로 가중치 즉 계수들을 서로 달리 해주면서 그 합을 1에 맞게 유지한다면, 같은 위치의 4가지 픽셀들이 고려되기 때문에 왜곡 없이 discontinuity를 없앨 수 있다. 그림 7은 이를 3차원적 입체적 그래프로 나타낸 것이다.

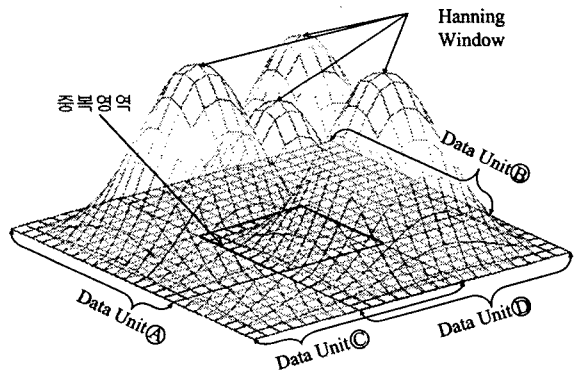


그림 7. Hanning Window를 이용한 Overlapped Block 연산의 3차원 투시도

Fig. 7. Perspective plot of the Overlapped Block by using Hanning Window in the three dimension

그림 7의 가운데 굵은 선 부분이 위에서 언급한 중복된 영역이다. 그리고 등방성(isotropic) 분포 형태를 지닌 것은 Hanning Window의 각 계수들의 값을 일정한 높이로 하여 서로 이어서 나타난 형태이다.

III. 실험 및 결과

그림 8은 16×16 픽셀 크기의 데이터 유닛에 CRA를 적용시켜 나타난 결과이다. 그림 8-(a)는 ISO 해상도 차트의 일부분이며 입력 이미지이다. 그림 8-(b)는 CRA의 결과이다. 그림 8-(a)에서 뭉개진 부분이 그림 8-(b)에서는 명확하게 되었다. 그림 8-(c)는 8-(a)와 8-(b)의 점선의 1-D 단면 픽셀 값 그래프이다. 그림 8-(c)의 원:○으로 그려져 있는 선은 그림 8-(a)의 점선 부분이며, 정사각형:□으로 그려져 있는 선은 그림 8-(b)의 점선 부분이다.

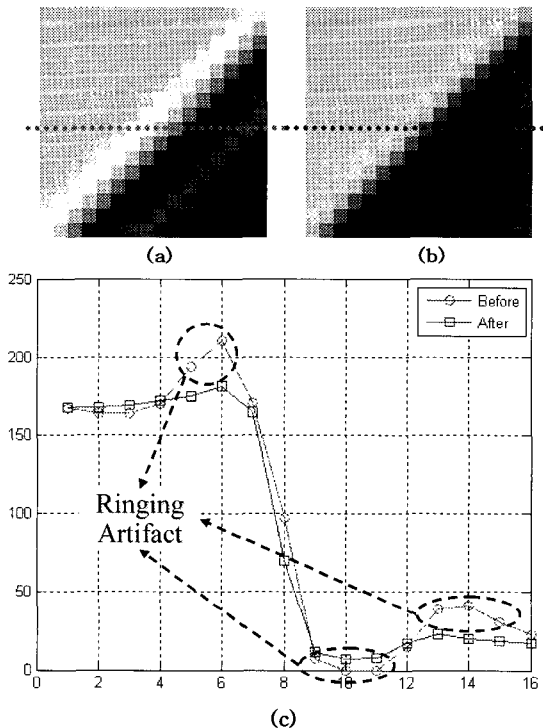


그림 8. (a) 입력 이미지 (16×16), (b) CRA 알고리즘의 결과 이미지 (16×16), (c) 1-D에서 통한 CRA 성능 전후 성능비교
 Fig. 8. (a) Original image from a portion of the ISO resolution chart (16×16), (b) Image obtained in CRA algorithm (16×16), (c) Comparison between the original image and the image obtained in CRA algorithm on the 1-D

그림 8-(c)의 원:○ 그래프에서 이미지의 에지(Edge)를 중심으로 하여 overshoot 및 undershoot 현상인 링잉 노이즈 현상이 나타난다. 그러나 그림 8-(c)의 정사각형

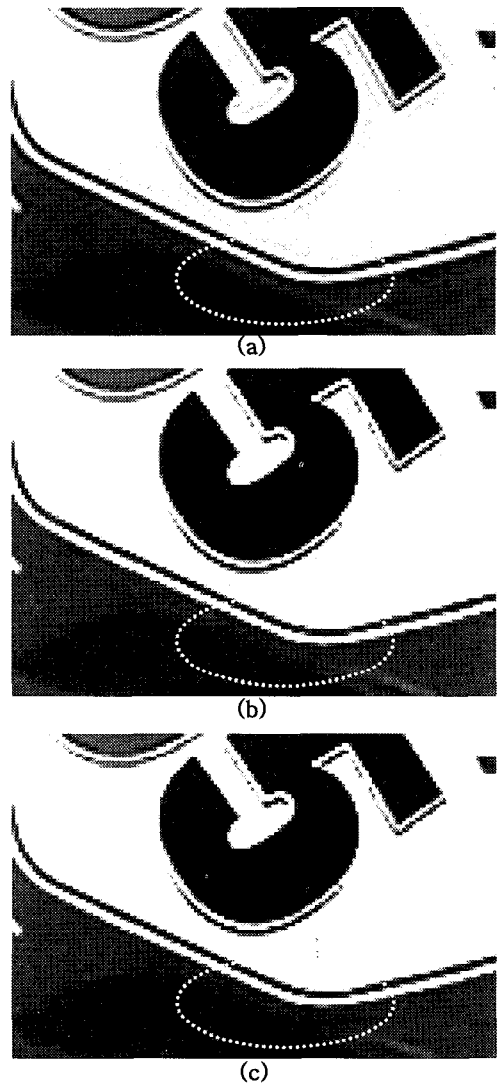


그림 9. (a) 입력 이미지 (96×144), (b) Non-overlapped에서 CRA 알고리즘의 결과 이미지 (96×144), (c) Overlapped에서 CRA 알고리즘의 결과 이미지 (96×144)
 Fig. 9. (a) Original image (96×144), (b) Image obtained in Non-overlapped with CRA algorithm (96×144), (c) Image obtained in Overlapped with CRA algorithm

□ 그래프는 알고리즘을 통해서 이러한 값들이 제거가 되었음을 보여준다.

제안한 알고리즘 전체의 성능을 검증하기 위해서 입력 영상으로 링잉 노이즈를 가진 그림 9-(a)를 사용하였

다. 비 중첩 블록 기반과 CRA 알고리즘을 수행한 결과를 그림 9-(b)에 나타내었다. 그림의 위 쪽 중앙의 숫자 '5'를 중심으로 흰 부분에 링잉 노이즈가 감소되었다. 그러나 이미지의 아랫부분의 에지 영역에서 블록 불연속 노이즈가 나타난다. 중첩 블록 기반과 CRA 알고리즘 수행하고, Hanning Window를 이용하여 복원한 결과를 그림 9-(c)에 나타내었다. 그림 9-(b)와 동일하게 링잉 노이즈가 감소시켰다. 그러나 이미지의 아랫부분의 에지 영역에서 블록 불연속 노이즈는 없다. 그림 10은 그림 9의 이미지에서 타원 부분의 1-D 단면의 픽셀 값 그래프이다.

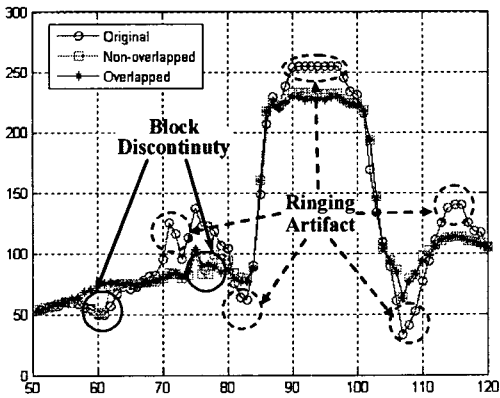


그림 10. 1-D에서 통한 Non-overlapped와 Overlapped에서 CRA 성능비교

Fig. 10. Comparison among the original image, the image obtained in Non-overlapped with CRA algorithm, and the image obtained in Overlapped with CRA algorithm on the 1-D

그림 10의 원:○으로 그려져 있는 선은 그림 9-(a)의 타원 부분이며, 정사각형:□으로 그려져 있는 선은 그림 9-(b)의 타원 부분이며, 별:*으로 그려져 있는 선은 그림 9-(c)의 타원 부분이다. 원:○의 그래프는 이미지의 에지(Edge)를 중심으로 하여 overshoot 및 undershoot 현상인 링잉 노이즈 현상이 존재함을 보여준다. 정사각형:□과 별:*은 CRA 알고리즘에 의해 링잉 노이즈가 감소가 되었다. 하지만 비 중첩 기반인 정사각형:□ 그래프가 다른 2개의 그래프들에 비해서, 16개씩 픽셀의 값이 급격하게 변하는 블록 불연속 노이즈가 있음을 보여준다. 반면에 중첩 기반과 Hanning Window연산 결과인 별:* 그래프는 언급된 2가지의 노이즈가 나타나지 않는다.

IV. 결론

본 논문에서 제안하는 알고리즘은 링잉 노이즈 감소를 위한 CRA 알고리즘을 사용할 때 블록화 현상을 방지할 수 있는 중첩 기반(Overlapped Block) 연산과 Hanning Window에 관련된 것이다. 블록 간 경계에서 불연속 노이즈 문제를 해결하기 위해서 이미지를 블록 구조 동작인 16×16 픽셀 크기의 데이터 유닛을 설정하고 그것의 크기 절반을 중첩 시키는 방법인 중첩 블록 구조 기반을 사용하였다. CRA 알고리즘은 데이터 유닛을 수정된 K-means 알고리즘을 기반으로 하여 링잉 노이즈를 제거하였다 [9]. 그 후 처리된 결과를 Hanning Window를 사용하여 복원하였다. 링잉 노이즈와 블록 불연속 노이즈가 제거되는 테스트 결과로 알고리즘의 우수성을 확인할 수 있다. 따라서 제안된 알고리즘을 통해서 JPEG 및 MPEG과 같은 블록 기반 방법에 중첩 기반 연산과 Hanning Window를 적용시킨다면, 블록 불연속 노이즈가 없는 영상을 얻을 수 있다.

참고문헌

- [1] K. Jack, *Video Demystified: A Handbook for the Digital Engineer*, HighTextPub., 1996.
- [2] H. Kong, A. Vetro, and H. Sun, "Edge map guided adaptive post-filter for blocking and ringing artifacts removal," *ISCAS 2004*, Vol. 3, pp.929-932, May 2004.
- [3] R.C. Gonzalez, *Digital image processing 2nd edition*, 2002.
- [4] J.H Kim, *Design of Smart Zoom System with Boost-up Filter and ADEN Algorithm*, Ph.D. Thesis, Dong-A University, Busan, Korea, Feb. 2006.
- [5] B. Maliatski and O. Yadid-Pecht, "Hardware-driven adaptive K-means clustering for real-time video imaging," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 15, No. 1, pp. 164-166, Jan. 2005.
- [6] D. Charalampidis, "A modified K-means algorithm for circular invariant clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 12, pp. 1856-1865, Dec. 2005.

- [7] W. Jang, J. Kim, W. Choi, B. Kwak, and B. Kang, "Ringing Artifacts Removal System for Mobile Application Camera by Modified K-means Algorithm," Submitted to 2007 Advanced Video and Signal based Surveillance, Mar. 2007.
- [8] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, New Jersey, 1998.
- [9] J. Kim, W. Jang, B. Kwak, S. Kim, and B. Kang, "A New Image-scaling Algorithm Eradicating Blurring and Ringing to Apply to Camera Phones," International Conference on Consumer Electronics, p. 5.4-1, Jan. 2007.

저자소개



김 주 현(Joohyun Kim)

2002년 2월 동아대학교 전기전자컴퓨터공학부 전자공학과(공학사)
2004년 2월 동아대학교 전자공학과(공학석사)

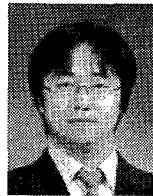
2007년 2월 동아대학교 전자공학과(공학박사)
2007년 3월~현재 삼성전기 중앙연구소 책임연구원
※ 관심분야 : VLSI algorithm/architecture design, image/video processing, and wireless communication



장 원 우(Wonwoo Jang)

2005년 2월 동아대학교 전기전자컴퓨터공학부 전자공학과(공학사)
2007년 2월 동아대학교 전자공학과(공학석사)

2007년 3월~현재 동아대학교 전자공학과 박사과정
※ 관심분야 : VLSI algorithm/architecture design, image/video processing, and wireless communication



박 정 환(Junghwan Park)

2006년 2월 동아대학교 전기전자컴퓨터공학부 전자공학과(공학사)
2006년 3월~현재 동아대학교 전자공학과 석사과정

※ 관심분야 : VLSI algorithm/architecture design, image/video processing, and wireless communication



양 훈 기(Hoongee Yang)

1985년 연세대학교 전자공학과(공학사)

1987년 미국 SUNY at Buffalo 전기 및 컴퓨터공학과(공학석사)

1992년 미국 SUNY at Buffalo 전기 및 컴퓨터공학과(공학박사)

1993년~현재 광운대학교 전파공학과 교수

※ 관심분야 : UWB and wireless communication



강 봉 순(Bongsoon Kang)

1985년 연세대학교 전자공학과(공학사)

1987년 미국 University of Pennsylvania 전기공학과(공학석사)

1990년 미국 Drexel University 전기 및 컴퓨터공학과(공학박사)

1989년~1999년 삼성전자 반도체 수석연구원

1999년~현재 동아대학교 전자공학과 부교수

2006년~현재 멀티미디어 연구센터 소장

2006년~현재 2단계 BK21 사업팀장

※ 관심분야 : VLSI algorithm/architecture design, image/video processing, and wireless communication