

특집논문 (Special Paper)

방송공학회논문지 제22권 제3호, 2017년 5월 (JBE Vol. 22, No. 3, May 2017)

<https://doi.org/10.5909/JBE.2017.22.3.313>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

Corner Outlier Artifacts를 감소시키기 위한 HEVC 필터링 방법

고 경 환^{a)}, 이 영 렬^{a)†}

The Filtering Method to Reduce Corner Outlier Artifacts in HEVC

Kyung-hwan Ko^{a)} and Yung-Lyul Lee^{a)†}

요 약

HEVC 표준에 적용된 인루프 필터링 방법인 De-blocking filter와 SAO(Sample Adaptive Offset)은 영상의 블록화 현상(Blocking artifacts)과 ringing artifacts를 효과적으로 제거하여 부호화 효율 및 주관적 화질 향상을 달성하였다. 하지만, 인루프 필터링 사용에도 불구하고, Corner outlier artifacts라 불리는 블록 경계의 모서리 부분에서 발생하는 2차원 형태의 블록화 현상은 쉽게 제거하지 못하는 문제점을 보인다. 본 논문에서는 Corner outlier 픽셀 탐색 및 판단과정, 해당 outlier 픽셀의 필터링 방법을 통하여 이러한 artifacts를 감소시키는 방법을 제안한다. 실험결과, 제안하는 방법은 화면간 예측에서 약간의 압축 효율 향상을 보였으며, 특히 영상의 주관적 화질 측면에서 개선된 효과를 나타내었다.

Abstract

The In-loop filtering methods such as de-blocking filter and SAO(Sample Adaptive Offset) applied to the HEVC standard achieves coding efficiency and subjective quality improvement by reducing the blocking artifacts and the ringing artifacts. However, despite the use of In-loop filtering methods, the artifacts called a corner outlier occurring at the corner points of block boundaries are not removed. In this paper, the corner outlier artifacts are reduced by the detection, determination, and filtering processes on the corner outlier pixels. Experimental results show that the proposed method improves the subjective picture quality and slightly increases the coding efficiency in Inter prediction.

Keyword : HEVC, Corner Outlier artifact, filtering, artifacts reduction, subjective quality

1. 서 론

a) 세종대학교 컴퓨터공학부(Sejong University, Dept. of Computer Eng.)

† Corresponding Author : 이영렬(Yung-Lyul Lee)

E-mail: ylle@sejong.ac.kr

Tel: +82-2-3408-3753

ORCID: <http://orcid.org/0000-0003-2709-8282>

※ 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2015R1A2A2A01006085).

· Manuscript received March 6, 2017; Revised April 27, 2017; Accepted April 27, 2017.

HEVC(High Efficiency Video Coding)^[1]는 ITU-T의 VCEC(Video Coding Expert Group)과 ISO/IEC 산하 MPEG(Moving Picture Expert Group)이 JCT-VC(Joint Collaborative Team on Video Coding)^[1]이란 팀을 구성하여 개발한 동영상 압축 표준이다. HEVC 표준 기술에

서는 CTU(Coding Tree Unit)로부터 최소 8x8부터 최대 64x64 크기로 다양하게 분할될 수 있는 단위인 CU(Coding Unit)를 적용하였고, 이는 다시 예측단위인 PU(Prediction Unit)로 분할되어 인트라/인터 예측을 수행한다. 또한, CU 내에서 다시 변환단위인 TU(Transform Unit)로 분할되어 변환 및 양자화를 수행한다. 이와 더불어 HEVC 표준에서는 디블로킹 필터(De-blocking filter)^[2]와 SAO(Sample Adaptive Offset)^[3]를 포함한 인루프 필터링 방법을 적용하여, 영상 내 블록화 현상(Blocking artifacts) 및 링잉현상(Ringing artifacts)을 효과적으로 제거하는 기술을 채택하고 있다. 먼저 디블로킹 필터의 경우, 수직방향에서 수평방향 순으로, 인접한 두 블록의 경계상 발생하는 블록화 현상을 체크한 후 저대역통과필터(Low-pass filter)를 적용하여 블록화 현상을 제거하는 방법을 사용한다. SAO는 CTU 단위로 RD-cost기반으로 두가지 타입인 에지오프셋(Edge offset)이나 밴드오프셋(Band offset)을 결정하며, 결정된 오프셋 타입으로 계산된 4개의 오프셋 값들을 디코더로 전송하여 링잉현상을 제거한다.

하지만, HEVC에 적용된 인루프 필터링 방법은 블록화 현상과 링잉 현상은 효과적으로 제거할 수 있으나, 변환(Transform) 및 양자화(Quantization) 이후 잔차신호의 오류에 의해 블록의 모서리 부분에서 주로 발생하는 Corner outlier artifacts이란 현상을 제거하지 못하는 문제점을 나타낸다. Corner outlier artifacts는 주로 변환 및 양자화된 블록의 경계 부근에 존재하는 4개의 인접한 블록의 Corner 부분 픽셀 값들 중 하나가 다른 이웃한 블록의 픽셀 값들보다 더 크거나 작은 경우를 말하며, 이는 변환 및 양자화를 통하여 생성된 잔차신호의 오류에 의해 블록 내부의 AC성분들의 제거로 인하여 발생한다. 그림 1은 QP(Quantization parameter)값을 32로 설정하여 부호화한 basketballPass 영상에서 나타나는 Corner outlier의 예를 보인 그림으로서, 그림 1의 왼쪽인 원본 영상과 비교하였을 때 오른쪽의 복원된 영상에서 잔차신호의 오류로 인하여 블록의 Corner부분에서 outlier 픽셀값이 주변의 다른 픽셀 값들보다 차이가 나는 것을 쉽게 알아볼 수 있다. 이는 특히 현재 영상을 부호화/복호화 하기 위하여 사용하는 참조영상의 화질을 떨어뜨려 인터예측에서 부호화 및 복호화 성능의 저하를 야기할 수 있다.

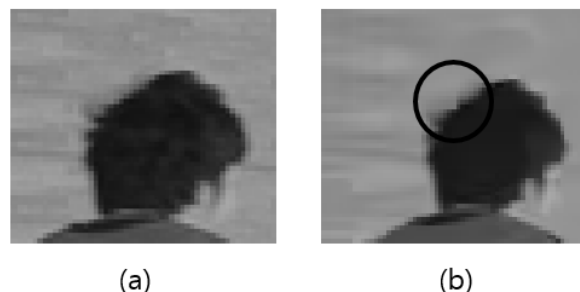


그림 1. Corner outlier artifacts의 예시: (a)원본영상, (b): Corner outlier artifacts를 포함한 복원영상

Fig. 1. Example of the Corner outlier artifacts: (a) original picture, (b) reconstructed picture including the Corner outlier artifacts

이러한 Corner outlier artifacts 현상을 영상에서 감소시키기 위하여 JPEG(Joint Picture Expert Group) 표준을 이용하여 Corner outlier 탐색 및 필터링 방법^{[4][5]}을 적용하여 PSNR 향상 및 주관적 화질을 향상시키는 방법이 제안되었다. 또한, MPEG-4 표준에서 Corner outlier artifacts를 줄이는 방법을 적용한 방법^[6]이 소개되었으며, H.264/AVC 표준의 인트라예측(Intra prediction)에서 디블로킹 필터링 방법과 Corner outlier artifacts reduction 방법을 함께 사용하는 방법^[7]도 제안되었다. 하지만, 그동안 HEVC를 기반으로 Corner outlier artifacts를 감소시키는 방법은 연구가 활발히 진행되지 않았다. 따라서, 본 논문에서는 인트라예측 및 인터예측으로 부호 및 복호화된 영상에서 Corner outlier artifacts를 줄이는 방법을 제안한다. 본 논문의 구성으로서 2장에서 Corner outlier 탐색 및 결정, 선택된 outlier 픽셀과 주변 픽셀을 필터링 방법을 제안하며, 3장에서 실험 결과 및 분석을 언급하고 4절에서 결론으로 마무리 짓는다.

II. Corner outlier artifacts reduction 방법

Corner outlier artifacts를 줄이는 방법은 총 3가지 단계로 구분되어 진행된다. 첫 번째 단계는 블록 경계의 corner 위치에서 해당 픽셀이 Corner outlier 픽셀인지 아닌지 체크하는 탐색과정이며, 두 번째 단계는 탐색과정에서 얻은 outlier 픽셀이 Corner outlier artifacts로 인하여 얻어진 픽셀인지 혹은 영상에서의 실제 에지인지를 판별하는 결정과

정이다. 앞의 두 가지 단계를 모두 충족한 경우에만 Corner outlier 필터링 단계가 수행되어 최종적으로 Corner outlier artifacts를 감소시킨다.

1. Corner outlier 탐색과정

Corner outlier artifacts는 주로 블록 경계의 현재 코딩할 블록을 포함한 4개의 블록이 인접해 있는 Corner 위치에서 주로 발생한다. 그림 2는 4개의 인접한 $2N \times 2N$ 블록들의 경계부근 Corner 위치에서 outlier 픽셀을 탐색하기 위해 사용하는 픽셀들의 예시를 나타낸다. Corner 위치는 그림에서 두 줄로 표시되어있는 4개의 블록들이 서로 마주친 위치이다. 이는, outlier의 중복 탐색을 방지하기 위함이며, 현재 코딩할 블록의 왼쪽 상단에서만 수행한다. 따라서 현재 코딩할 블록은 항상 그림 2처럼 픽셀 'D'를 포함하는 오른쪽 아래에서만 위치한다. 그림 2에서 표시된 사각형은 각각 픽셀을 의미한다. 대문자로 표기된 픽셀들이 Corner outlier 탐색과정에 사용되는 후보픽셀들이며, 소문자로 표기된 픽셀들은 세 번째 단계인 필터링 단계에서 사용되어 후보픽셀과 함께 필터링되는 픽셀들을 나타낸다.

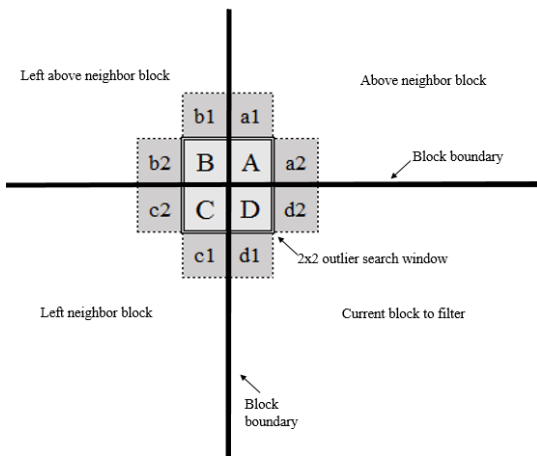


그림 2. 4개의 $2N \times 2N$ 블록 경계에 위치한 Corner 위치와 후보픽셀들 A,B,C,D

Fig. 2. An example of the corner points of the four $2n \times 2n$ blocks: A,B,C,D is one of the corner outlier candidates

그림에서 나타나듯이 4개의 후보 픽셀들 중 어느 픽셀이 Corner outlier 픽셀인지 알기 위하여 참조논문인 JPEG에

Corner outlier artifacts 제거방법을 적용하였던 방법^[4]에서 사용하였던 식(1)을 이용한다. 식(1)에서 사용된 $| \bullet |$ 연산자는 절대값 연산을 나타내며, 식의 threshold 값은 HEVC에서 사용하는 양자화 스텝 계수(QStep : Quantization step value)의 절반 값으로 설정하였다. QStep 값은 사용하는 QP에 따라 변경되는 값이며, HEVC에 테이블 형식으로 미리 저장되어 있는 값이다. 만약 QP 값이 32라면, threshold 값은 약 12.75로 계산되며, 이를 반올림 한 값을 사용하게 된다. QStep 값을 threshold로 선택한 이유는, 양자화 단계에서 transform 된 잔차신호를 나누는 값이 QStep 값이기 때문이다. Corner outlier artifacts를 비롯한 다양한 blocking artifacts들은 양자화 단계에서 잔차신호의 오류로 인하여 발생되기 때문에, 이 값은 QStep 값과 매우 밀접한 연관을 가지고 있다. 따라서 QStep 값의 절반 값을 threshold 값으로 이용하여, 양자화 오차 범위 내에서 outlier 픽셀을 탐색하는 방법을 선택하였다. 이러한 이유로, QStep값의 절반을 이용하여 식(1)을 통해 반환된 픽셀 위치는 Corner 위치에 존재하는 4개의 후보 픽셀 중 Corner outlier 픽셀로 탐색되어진 픽셀을 나타내게 된다. 식(1)의 첫 번째 조건은 4개의 후보 픽셀의 대각 방향을 고려하여 Corner outlier를 탐색하는 과정을 나타낸다. 식(1)의 첫 번째 조건을 만족한다면, 두 번째 단계로서 후보 픽셀들의 수평 방향을 고려하여 outlier 픽셀을 탐색한다. 위 두 가지 조건을 만족하게 되면, 현재 픽셀과 나머지 후보 픽셀들의 값이 threshold보다 모두 큰 경우, 해당 픽셀이 outlier 픽셀임을 확인한다.

```

If ( $|A-C| > |B-D|$ )
If ( $|A-B| > |C-D|$ )
If ( $|A-B| > \text{threshold} \ \&\& \ |A-C| > \text{threshold} \ \&\& \ |A-D| > \text{threshold}$ )
    Return A;
Else
If ( $|C-A| > \text{threshold} \ \&\& \ |C-B| > \text{threshold} \ \&\& \ |C-D| > \text{threshold}$ )
    Return C;
Else
    (1)

If ( $|B-C| > |A-D|$ )
If ( $|B-A| > \text{threshold} \ \&\& \ |B-C| > \text{threshold} \ \&\& \ |B-D| > \text{threshold}$ )
    Return B;
Else
If ( $|D-A| > \text{threshold} \ \&\& \ |D-B| > \text{threshold} \ \&\& \ |D-C| > \text{threshold}$ )
    Return D;

```

2. Corner outlier 결정과정

두 번째 과정은 Corner outlier 탐색과정을 통해 얻은 픽셀이 진짜 Corner outlier artifacts로 인하여 생긴 outlier 픽셀인지 아니면 실제 영상에서의 경계인지를 판단하는 Corner outlier 결정과정^[8]이다. Corner outlier 탐색 과정을 통해 얻은 outlier 픽셀을 바로 필터링 하게 된다면 Corner outlier artifacts가 아닌 영상 내부의 실제 에지(Edge)까지 Corner outlier로 오인하여 필터링 하는 문제점이 발생할 수 있다. 그림 3은 basketballPass 영상에서 실제 Corner outlier artifacts가 아니지만, Corner outlier 결정과정을 거치지 않고 필터링 과정을 수행하여 outlier로 오인하게 되어 의도치 않은 잡음(noise)을 생성할 때의 예시를 나타낸 것이다.

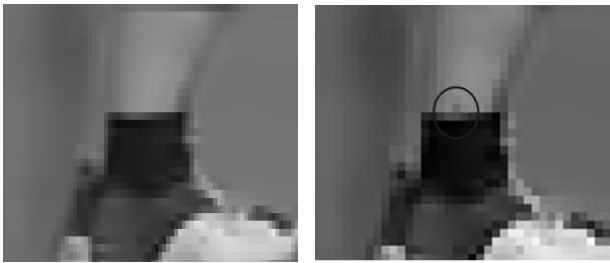


그림 3. Corner outlier 결정과정 없이 Corner outlier 필터링 한 경우: (왼쪽) 필터링 전, (오른쪽) 필터링 후

Fig. 3. Corner outlier filtering without Corner outlier confirmation method: (left) before filtering, (right) after filtering

앞의 과정인 Corner outlier 탐색과정은 블록 Corner 부분의 4개의 픽셀만을 이용하여 outlier 후보를 결정하였기 때문에 현재 corner 픽셀이 outlier인지 아닌지 판별할 수 있는 확실한 정보를 가지지 못한다. 따라서, Corner outlier 결정과정을 통하여 최대한 실제 에지와 Corner outlier를 구별해야 한다. 결정방법은 디블로킹 필터에서 사용하는 에지 판별 방법을 응용하여 실제 에지와 outlier를 구별한다. 그림 3은 Corner outlier 결정 과정에 사용되는 픽셀들을 나타낸 그림이며, 후보 픽셀 중 'C' 픽셀이 Corner outlier로 탐색된 경우를 나타낸다. 식(2)는 픽셀변화량을 계산하여 변화량이 디블로킹 필터에서 사용되는 $iBeta$ 값에 특정 임계값인 $threshold2$ 를 더한 값보다 작은 경우에만 true를 반환한다.

이 과정은 수직방향과 수평방향 모두 true로 반환된 경우만 필터링 과정으로 진행되게 구성하였으며, 두 방향 중 하나라도 만족하지 않으면 Corner outlier 필터링 과정을 수행하지 않는다. 수식의 $threshold2$ 값은 실험적으로 얻은 결과값인 7로 설정하여 사용하였다.

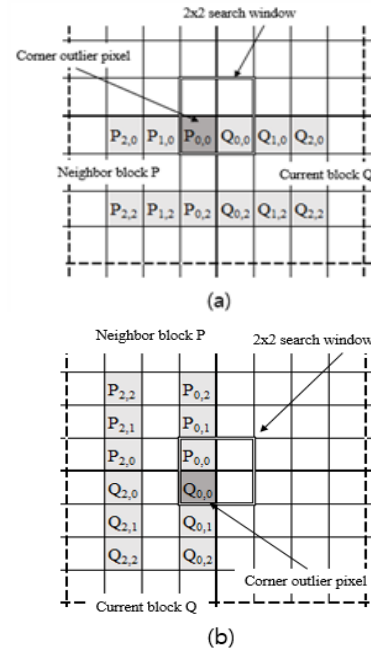


그림 4. Corner outlier를 결정하기 위해 사용되는 픽셀들의 위치 및 결정과정: (a)수평방향, (b)수직방향

Fig. 4. Pixels for corner outlier confirmation process: (a)vertical direction, (b)horizontal direction

$$\begin{aligned} dp0 &= |P_{2,0} - 2*P_{1,0} + P_{0,0}| \\ dp3 &= |P_{2,2} - 2*P_{1,2} + P_{0,2}| \\ dq0 &= |Q_{2,0} - 2*Q_{1,0} + Q_{0,0}| \\ dq3 &= |Q_{2,2} - 2*Q_{1,2} + Q_{0,2}| \end{aligned} \quad (2)$$

$$Diff = dp0 + dp3 + dq0 + dq3$$

$$\begin{aligned} \text{If}(Diff < iBeta + threshold2) \\ Ver(hor)outlier = true \end{aligned}$$

3. Corner outlier 필터링과정

Corner outlier 탐색 및 결정 과정이 완료되어 최종 Corner outlier 픽셀로 결정된 후보픽셀 및 그와 인접한 2개의 픽셀들은 저대역통과필터 기반의 필터링 과정을 통하여

필터링된다. 식(3)은 후보 픽셀 ‘C’가 최종 Corner outlier 픽셀로 판별되어 필터링 되는 예를 나타낸 것이다. Luma 컴포넌트는 식(3)의 모든 수식을 사용하며, Chroma 컴포넌트에서는 식(3)의 첫 번째 수식만을 사용한다.

$$\begin{aligned} C' &= ((2 * A) + B + (4 * C) + D + 4) \gg 3 \\ c1' &= (C' + (3 * c1) + 2) \gg 2 \\ c2' &= (C' + (3 * c2) + 2) \gg 2 \end{aligned} \quad (3)$$

선택된 Corner outlier 픽셀과 인접 픽셀을 필터링 한 후, 디블로킹 필터에서는 해당 Corner 위치에서 Corner outlier 필터링이 진행되었지만, 디블로킹 필터링이 진행되지 않은 블록 경계를 필터링 한다. 이는 outlier 필터링된 Corner 위치와 이를 포함한 블록 경계에서 약 41%의 경우 픽셀 변화량을 의미하는 변수인 Diff가 iBeta값보다 크기 때문에 디블로킹 필터가 동작하지 않아 생긴 블록화 현상을 고려한 것이다.

III. 실험 및 결과분석

본 논문에서 제안하는 Corner outlier 감소 방법은 HEVC의 참조 소프트웨어인 HM16.6^[9]버전을 통해 구현하였으며, JCT-VC의 CTC(Common Test Condition)^[10]의 실험 영상 A-F Class의 영상들을 사용하였다. 실험은 RA(Random

access main)와 LDB(Low-delay B main), LDP(Low-delay P main) 환경에서 테스트 하였으며, QP 값은 22, 27, 32, 37을 각각 사용하였다. Class A는 2560x1600 크기로 축소된 4K 영상이며, Class B는 1920x1080사이즈의 Full HD영상을 나타내며, Class C는 832x480크기를 갖는 영상이다. Class D는 416x240 사이즈의 영상이며, Class E는 화상용 영상으로서 1280x720 크기의 영상이다. Class F는 스크린 콘텐츠 영상으로서 Class E와 마찬가지로 1280x720 크기의 영상을 말한다. 본 실험에서는 CTC의 RA환경에서 실험하지 않는 Class E와, LD환경에서 사용하지 않는 Class A까지 함께 실험하여 다양한 영상에서의 성능평가를 비교하였다. 먼저 표 1은 각각 RA, LDB, LDP 환경에서 4가지 QP 값들을 통해 실험한 BD-rate의 성능 평가를 나타낸 것이다.

표 1에서 나타난 것처럼 Corner outlier artifacts를 줄이는 방법에서 BD-rate 성능 향상은 거의 나타나지 않았다. 그 이유로서는 첫 번째, 영상 전체 Corner 위치에서 Corner outlier artifacts가 발생하는 빈도가 현저히 낮은 점을 들 수 있다. 표 2는 All Intra 환경과 Low-delay P 환경에서 각 Class에서 영상 2개씩을 선정하여 Corner outlier 결정과정 없이 원본 영상과 Corner outlier 필터링 한 영상과의 MSE (Mean Square Error)를 계산하여 Corner outlier 필터링을 실행하였을 때 원본영상과의 비교하여 픽셀들의 에러가 감소하는 블록들의 비율을 나타낸 표이다.

표 1. Inter Prediction 에서의 BD-rate 성능평가: Random access, Low-delay B main, Low-delay P main
Table 1. Experimental results of Random access, Low-delay B main, Low-delay P main

	Random Access main			Low-delay B main			Low-delay P main		
	BD-rate Y	BD-rate U	BD-rate V	BD-rate Y	BD-rate U	BD-rate V	BD-rate Y	BD-rate U	BD-rate V
Class A	-0.1%	0.0%	0.0%	-0.1%	-0.2%	0.0%	-0.1%	0.1%	0.1%
Class B	0.0%	0.2%	-0.1%	-0.1%	0.0%	-0.2%	-0.1%	-0.1%	0.0%
Class C	-0.1%	-0.1%	0.0%	0.0%	-0.2%	-0.2%	-0.2%	-0.3%	-0.3%
Class D	0.1%	0.0%	0.1%	0.1%	0.3%	0.3%	0.0%	0.3%	0.1%
Class E	0.0%	-0.4%	-0.2%	0.0%	0.2%	0.1%	-0.1%	0.0%	0.0%
Class F	0.4%	0.3%	0.1%	0.1%	-0.3%	0.0%	0.1%	0.1%	0.1%
Overall	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	-0.1%	0.0%	0.0%
Enc Time	100%			100%			100%		
Dec Time	109%			110%			113%		

표 2. Corner outlier 필터링 시 원본영상과 비교하여 에러가 감소하는 블록들의 비율

Table 2. The ratio of the blocks whose error is reduced compared to the original images and the filtered images

Sequence	All Intra				Low delay P			
	QP=22	QP=27	QP=32	QP=37	QP=22	QP=27	QP=32	QP=37
PeopleOnStreet	5.5%	15.3%	31.1%	46.6%	9.4%	27.1%	48.0%	65.0%
Traffic	5.5%	18.2%	34.6%	53.2%	6.1%	12.0%	34.0%	54.9%
Kimono	7.9%	25.3%	42.4%	56.6%	11.3%	33.6%	53.9%	72.5%
Cactus	5.1%	18.0%	32.7%	45.4%	7.7%	19.4%	31.4%	44.2%
BasketballDrill	6.4%	20.3%	32.3%	42.2%	8.8%	23.6%	39.5%	55.8%
BQMall	5.7%	17.0%	29.4%	41.9%	6.9%	18.9%	34.0%	49.9%
BasketballPass	6.5%	21.0%	34.8%	43.0%	10.1%	25.3%	39.0%	49.4%
BlowingBubbles	4.5%	15.6%	28.4%	39.2%	4.5%	12.2%	23.3%	35.3%
Johnny	6.4%	21.0%	34.9%	48.7%	4.4%	12.1%	26.1%	43.8%
KristenAndSara	6.7%	19.8%	32.4%	45.5%	4.9%	14.4%	29.1%	48.3%
Average	6.0%	19.2%	33.3%	46.2%	7.4%	20.4%	35.8%	51.9%

표 3. BD-rate 성능평가: [6] 과 제안하는 Corner outlier 감소기법

Table 3. Experimental results of [4], [6], Proposed method

	[6]			Proposed Method		
	BD-rate Y	BD-rate U	BD-rate V	BD-rate Y	BD-rate U	BD-rate V
Class A						
Class B	0.7%	0.7%	0.5%	-0.1%	-0.1%	0.0%
Class C	0.3%	0.1%	0.6%	-0.2%	-0.3%	-0.3%
Class D	0.6%	0.0%	0.6%	0.0%	0.3%	0.1%
Class E	0.4%	0.3%	0.9%	-0.1%	0.0%	0.0%
Class F	0.3%	-0.5%	-0.1%	0.1%	0.1%	0.1%
Overall	0.5%	0.1%	0.5%	-0.1%	0.0%	0.0%

표 2에서 보이듯이 QP값이 22인 경우는 Corner outlier 필터링을 실행하였을 때 원본영상과 비교하였을 때, 에러가 감소하는 비율이 All Intra의 경우 약 6%, Low-delay의 경우 약 7%로 현저히 낮았다. 따라서, Corner outlier 필터링 시 BD-rate 향상의 효과가 크지 않을 수 있음을 알 수 있다. 두 번째 이유로, Corner outlier filtering을 통해 outlier 픽셀과 인접한 2개의 픽셀, 총 3개의 픽셀만을 필터링 하기 때문에 전체적으로 BD-rate 성능 향상에는 큰 영향을 미치지 않은 것으로 분석되었다.

표 3은 Low-delay P 환경에서 기존 Corner outlier 감소 방법인 [6]과 제안하는 방법과 비교한 BD-rate 성능평가를 나타낸 것이다. [6]은 outlier 결정과정에서 QP값을 threshold로 사용하여 블록 경계 주변의 픽셀값 차이와, outlier 픽셀이 속한 블록의 평탄성을 검토하여 수행하는 방법이다.

제안하는 방법은 디블로킹필터에서 사용하는 edge detection 방법을 이용한 Corner outlier 결정과정을 통해 명확한 outlier 픽셀을 선택하기 때문에 BD-rate 성능저하를 최소화 할 수 있었다.

비록 Corner outlier 제거방법이 BD-rate 성능 향상에 많은 효과를 얻지는 못하였지만, 영상의 주관적 화질에는 개선된 효과를 보여주었다. 그림 5는 그림 1의 예시인 basketballPass 영상에서 Corner outlier 필터링을 통해 개선된 주관적인 화질을 보여준다. 제거되었던 높은 주파수 성분을 필터링하여 그림 5의 오른쪽과 같이 개선된 영상을 생성할 수 있었다. 그림 6은 Kimono 영상에서 Corner outlier 필터링 된 부분을 나타내며, 그림 7은 PeopleOnStreet 영상에서 제안하는 방법으로 필터링 된 부분을 나타낸다. 그림 6과 그림 7은 Low-delay B 환경에서 인터예측으로 생성된 2번

째 영상으로서, Corner outlier artifacts를 줄이는 방법이 이전 인트라 영상과 인터예측으로 생성된 참조영상에 사용되어 참조영상의 화질 상승으로 인하여 Corner outlier 현상 및 블록화 현상이 사라진 것을 볼 수 있다.

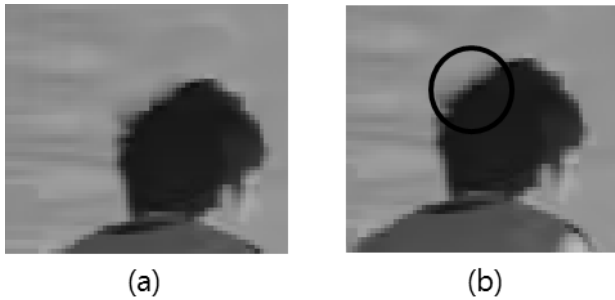


그림 5. 복호화된 실험영상: (a)HM16.6, (b): 제안하는 방법
Fig. 5. Subjective quality comparison: (a) HM16.6, (b) proposed method

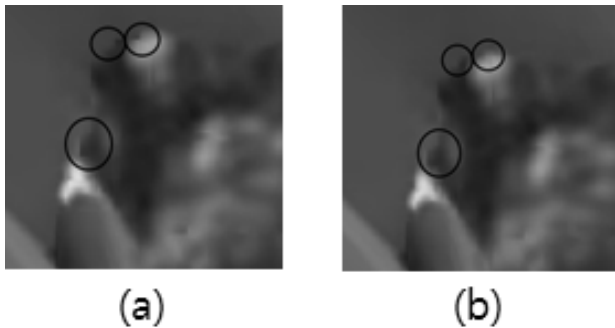


그림 6. 복호화된 실험영상: (a)HM16.6, (b): 제안하는 방법
Fig. 6. Subjective quality comparison: (a) HM16.6, (b) proposed method



그림 7. 복호화된 실험영상: (a)HM16.6, (b): 제안하는 방법
Fig. 7. Subjective quality comparison: (a) HM16.6, (b) proposed method

IV. 결 론

본 논문에서는 블록 경계의 Corner 위치에 나타나는 Corner outlier artifacts 현상을 감소시키는 방법을 제안하였다. 이를 위해 Corner 위치의 4개 후보 픽셀에 대하여 outlier를 탐색하는 과정과 선택된 픽셀이 artifacts인지 실제 에지인지 확인하는 결정과정, 최종 선택된 outlier 픽셀을 필터링 하는 과정을 통하여 Corner outlier artifacts를 줄이는 방법을 소개하였다. 제안하는 방법은 비록 BD-rate 성능향상은 크게 보장하지 못하였지만, 복원된 영상의 주관적 화질 측면에서 개선된 효과를 나타내었다.

참 고 문 헌 (References)

- [1] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Transactions On Circuits And Systems For Video Technology, Vol. 22, No. 12, December 2012.
- [2] Andrey Norkin, Gisle Bjøntegaard, Arild Fuldseth, Matthias Narroschke, Masaru Ikeda, Kenneth Andersson, Minhua Zhou, and Geert Van Der Auwera, "HEVC Deblocking Filter", IEEE Transactions On Circuits And Systems For Video Technology, Vol. 22, No. 12, December 2012.
- [3] Chih-Ming Fu, Elena Alshina, Alexander Alshin, Yu-Wen Huang, Ching-Yeh Chen, and Chia-Yang Tsai, Chih-Wei Hsu, Shaw-Min Lei, Jeong-Hoon Park, and Woo-Jin Han, "Sample Adaptive Offset in the HEVC Standard", IEEE Transactions On Circuits And Systems For Video Technology, Vol. 22, No. 12, December 2012.
- [4] Y. L. Lee, H. C. Kim, and H. W. Park, "Blocking effect reduction of JPEG images by signal adaptive filtering," IEEE Transactions On Image Processing, Vol. 7, pp. 229 - 234, February 1998.
- [5] A. Gandam and J. Sidhu, "A post-processing algorithm for detection and removal of corner outlier," International journal of computer and applications, vol. 4, pp. 9 - 16, Jul. 2010.
- [6] Yunpyo Hong, Byeungwoo Jeon, "Loop-filtering for Reducing Corner outlier", The Institute of Electronics Engineers of Korea - Signal Processing Vol.41, No.5, pp.217-223, September 2004.
- [7] Loung Pham Van, Jan De Cock, Clenn Van Wallendael, Byeungwoo Jeon, and Rik Van de Walle, "Adaptive Deblocking Filtering Scheme for Intra-Coded Slices in H.264/AVC", Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on, pp. 171-176, September 2013.
- [8] K. H. Ko, Y. L. Lee, "Corner Outlier reduction in HEVC", International Workshop on Advanced Image Technology 2017, Jan. 2017.
- [9] https://jvet.hhi.fraunhofer.de/svn/svn_HM/JEMSoftware/tags/HM-16.6/
- [10] Frank Bossen, "Common test conditions and software reference configurations". document JCTVC-I1100, May. 2012.

저 자 소 개



고 경 환

- 2016년 2월 : 세종대학교 디지털콘텐츠학과 (학사)
- 2016년 3월 ~ 현재 : 세종대학교 컴퓨터공학과 (석사과정)
- ORCID : <http://orcid.org/0000-0002-3096-8704>
- 주관심분야 : Future Video Coding, HEVC(High Efficiency Video Coding), Image compression, Image processing



이 영 렬

- 1985년 2월 : 서강대학교 전자공학과 (학사)
- 1987년 2월 : 서강대학교 전자공학과 (석사)
- 1999년 2월 : 한국 과학기술원 전기·전자공학과 (박사)
- 1987년 1월 ~ 1994년 2월 : 삼성전자 R&D 센터 Digital Media Lab.
- 1999년 3월 ~ 2001년 8월 : 삼성전자 R&D 센터 Digital Media Lab. 수석연구원
- 2001년 9월 ~ 현재 : 세종대학교 전자정보대학 컴퓨터공학부 컴퓨터공학과 교수
- ORCID : <http://orcid.org/0000-0003-2709-8282>
- 주관심분야 : Future Video Coding, HEVC(High Efficiency Video Coding), Scalable Video Coding(SVC), Multi-view video coding(MVC), H.264/AVC, Image processing, Frame rate up-conversion, Transcoding, image transform