

양방향 예측에서 LIC 복잡도를 줄이기 위한 방법

최한솔, 박준택, 이종석, 심동규, *방건, *김희용
광운대학교, *ETRI

{whiteblack4, juntaek, suk2080, dgsim}@kw.ac.kr, *{gbang, hykim5}@etri.re.kr

Modified LIC for Bi Prediction to Reduce Complexity

Hansol Choi Juntaek Park Jongseok Lee Donggyu Sim *Gun Bang *Huiyong Kim
Kwangwoon University, *ETRI

요 약

본 논문에서는 양방향 예측 모드에서 LIC(Local Illuminance Compensation)의 복잡도 감소를 위한 방법을 제안한다. 제안하는 방법은 현재 CU 가 양방향 예측 모드일 경우 움직임 보상 단계에서 예측 블록 두 개를 가중 평균을 낸 뒤 LIC 를 수행함으로써 양방향 예측모드에서 LIC 를 한번만 수행하여 기존 LIC 대비 낮은 복잡도를 보인다. 제안하는 방법의 성능 평가를 위하여 RA(Random Access) 조건에서 MPEG 표준 실험 영상의 B, C, D 클래스를 이용하여 BD-rate 를 VTM-2.0.1 과 비교한다. 실험결과로서 본 논문에서 제안하는 방법이 VTM2.0.1 대비 BD-rate 성능 관점에서 Y, U, V 각각에 대하여 평균 0.17, 0.17, 0.23 의 성능 향상을 보인다.

1. 서론

ISO/IEC MPEG(Moving Picture Expert Group)과 ITU-T VCEG(Video Coding Expert Group)은 JVET(Joint Video Exploration Team)을 구성하여 HEVC(High Efficiency Video Coding) [1] 대비 2 배 이상의 부호화 효율을 갖는 차세대 비디오 압축 표준화를 진행 중이다. 차세대 비디오 압축 소프트웨어인 VTM(Versatile Video Coding Test Model) [4]은 HEVC 대비 부호화 성능 향상을 위한 최소한의 기술을 포함하고 있다. 더 큰 부호화 성능 향상을 위한 추가 기술들이 포함된 소프트웨어가 BMS(Benchmark Set)이다. BMS 에 존재하는 화면간 참조 기술들 중 하나로 LIC(Local Illuminance Compensation)가 있다. 조명 변화 등에 의하여 현재 프레임과 참조 프레임 간에 부분적으로 조명 차가 있을 경우 현재 블록과 참조 블록 간의 픽셀 값 차이가 커질 수 있다[3]. LIC 는 현재 블록과의 픽셀 값의 분산의 차가 가장 작은 참조 블록에 현재 블록과의 밝기 차를 보상하는 기술이다. 기존의 LIC 는 양방향 예측을 통해 유도된 두개의 예측 블록에 각각 적용된다[2]. 현재 블록과 참조 블록의 주변 픽셀들 간의 선형 관계를 구해 밝기 보상 파라미터 scale 값과 offset 값을 유도한다. 하지만, LIC 기술은 큰 bitrate 이득이 있지만 하드웨어 구현상의 pipe-line 문제와 함께 계산 복잡도 문제가 해결되지 않은 상황이다. 따라서 본 논문에서는 LIC 의 복잡도를 줄이는 방법을 제안한다. 제안하는 방법은 현재 예측 블록이 화면간 예측의 양방향 예측일 경우 두 개의 예측 블록을 가중 합산하여 얻은 최종 예측 블록에 대하여 LIC 파라미터를 계산하고 LIC 를 적용한다. 또한 파라미터 유도 과정의 복잡도를 줄이기 위하여 LIC 파라미터는 offset 값만을 사용한다.

본 논문의 구성은 다음과 같다. 2 절에서는 제안한 방법을 설명하고, 3 절에서는 결과를 제시한다. 4 절에서는 본 논문에 대한 결론을 맺는다.

2. 본론

본 논문에서 제안하는 LIC 에서 양방향 예측일 때 예측 블록 생성 과정은 그림 1 과 같은 순서로 진행된다. 참조 프레임 리스트 0 과 1 로부터 2 개의 초기 예측 블록($Pred0$ 과 $Pred1$)이 유도된다. 초기 예측블록에 움직임 보상이 적용된다. 움직임 보상이 적용된 예측블록($Pred0'$ 과 $Pred1'$)에 가중 평균을 적용하여 하나의 예측 블록($Pred$)을 유도한다. 한 개로 합쳐진 예측 블록에 LIC 를 적용하여 최종 예측 블록을 유도한다.

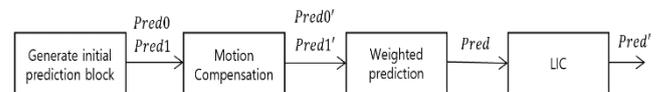


그림 1. 제안된 움직임 예측 블록 생성 흐름

양방향 예측일 경우 LIC 파라미터 유도 과정의 복잡도를 줄이기 위하여 offset 파라미터만 사용한다. offset 파라미터 (β)는 현재 블록의 이웃 픽셀과 참조 블록들의 이웃하는 픽셀들을 이용하여 수식(1)를 통해 유도된다.

$$\beta = \frac{\sum((R_0(n)+R_1(n)) \gg 1) - \sum C(n)}{N} \quad (1)$$

$R_0(n)$ 과 $R_1(n)$ 은 참조 블록들의 이웃 픽셀들이고, $C(n)$ 은 현재 블록의 이웃 픽셀들이고, N 은 현재 블록의 너비와 높이의 합이다. 두 예측블록을 평균 내어 합친 예측블록($Pred$)에 유도한 offset 파라미터를 더하여 LIC 가 적용된 최종 예측블록($Pred'$)을 생성한다.

$$Pred' = Pred + \beta \quad (2)$$

3. 실험결과

본 논문에서 제안하는 방법은 BMS2.0.1 기반으로 구현하였으며 성능 평가를 위한 실험 영상으로 MPEG 표준 실험 영상의 B, C, D 클래스를 RA(Random Access) 조건에서 진행하였다. 표 1 은 VTM2.0.1 대비 제안하는 방법의 BD-Rate 성능을 나타낸다. VTM2.0.1 대비 BD-rate 는 Y, U, V 각각 평균 0.17, 0.17, 0.23 의 BD-Rate 성능 향상을 보인다.

표 1. VTM2.0.1 대비 제안하는 방법의 BD-Rate

	BD-rate		
	Y (%)	U (%)	V (%)
ClassB	-0.45	-0.36	-0.39
ClassC	-0.17	-0.19	-0.17
ClassD	0.10	0.04	-0.15
Overall	-0.17	-0.17	-0.23

4. 결론

본 논문은 양방향 예측 모드에서 LIC 의 복잡도 감소를 위한 방법을 제안하였다. BMS2.0.1 에 구현되어 있는 LIC 는 양방향 예측에서 두 예측 블록에 각각 LIC 를 적용한 뒤 가중 평균을 낸다. 예측블록의 밝기 보상을 위한 LIC 파라미터로 scale 값과 offset 값을 사용한다. 본 논문에서는 LIC 의 복잡도 감소를 위하여 양방향 예측 모드일 경우 두 개의 예측 블록을 가중 평균을 낸 뒤 LIC 를 한번만 수행하였다. 양방향 예측의 경우 예측 블록의 밝기 보상을 위한 파라미터로는 offset 값 만을 사용하였다. 본 논문에서 제안하는 방법은 VTM2.0.1 대비 BD-rate 성능 관점에서 Y, U, V 각각에 대하여 평균 0.17, 0.17, 0.23 의 성능 향상이 있음을 보였다.

감사의글

이 논문은 IITP ETRI 지원사업인 이머시브 미디어 전문연구실사업(출연처과제번호 2018-0-00207)의 지원 및 2018 년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-018R1A2B2008238)의 지원을 받아 작성되었습니다.

참고문헌

- [1] Sullivan, G. J., Ohm, J. R., Han, W. J., and Wiegand, T., "Overview of the high efficiency video coding(HEVC) standard", IEEE Transactions on circuits and systems for video technology, 22(12), 1649-1668, Dec. 2012.
- [2] J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm and J. Boyce, JVET-G1001, "Algorithm description of Joint Exploration Test Model 7", Jul. 2017.
- [3] J. Hur, S. Cho and Y. Lee, "Adaptive local illumination change compensation method for H. 264/AVC-based

multiview video coding", IEEE Transactions on Circuits and Systems for Video Technology, 17(11), 1496-1505, Nov. 2007.

[4] J. Chen, Y. Ye and S. Kim, JVET-K1002, "Algorithm description for Versatile Video Coding and Test Model 2 (VTM 2)", Jul. 2018.