

다양한 해시 구성을 활용하는 HEVC 스크린 콘텐츠 코딩 화면 내 블록 카피 기법의 코딩 성능 분석

*이정현 **정제창

한양대학교 전자컴퓨터통신공학과

*mizali@hanyang.ac.kr **jjeong@hanyang.ac.kr

Performance Analysis of HEVC SCC Hash-based IntraBC through Modified Hash Structure

*Lee, Junghyun **Jeong, Jechang

Department of Electronics and Computer Engineering, Hanyang University

요약

기존의 자연영상에 초점이 맞춰져 있던 HEVC 동영상 압축 표준은 스크린 콘텐츠 코딩(SCC)을 제안하여 문자와 그래픽 효과가 많이 포함된 스크린 콘텐츠 영상의 압축 성능을 향상시켰다. SCC의 화면 내 블록 카피 기법은 스크린 콘텐츠 영상의 특성상 한 화면 내 반복되는 무늬들을 찾아 예측에 활용한다.

이에 본 논문은 화면 내 블록 카피 기법에서 사용되는 블록 탐색 방법에 대해 연구하였다. 특히, 전역 탐색에서 사용되는 해시 기반의 탐색기법의 해시 값 구성의 분석을 통해, 해시 구성 요소들의 비중을 감소와 새로운 해시 구성 요소를 활용하여 최대 0.330% 비트레이트율 감소 및 0.328% 비트레이트율 감소 효과를 확인할 수 있었다.

1. 서론

HD/UHD 고화질 영상의 효과적인 압축을 위해 국제 표준화 기구 JCT-VC (Joint Collaborative Team on Video Coding)에서는 HEVC (High Efficiency Video Coding) 동영상 압축 표준을 제정하였다. 하지만 자연 영상에 대하여 최적화된 HEVC 동영상 압축 표준의 요소 기술은 옛이나 색상의 높은 대조비가 포함되는 그래픽이나 글자가 포함되는 스크린 콘텐츠 영상 압축에는 최적화 되어있지 못하다.[1] 이에 따라 JCT-VC는 HEVC 확장 표준(HEVC Range Extension, HEVC/RExt)을 제안하여, HDR(High Dynamic Range) 영상과 스크린 콘텐츠 영상 등 다양한 동영상 포맷의 효율적인 압축방식의 표준을 포함하였다.[2] 이 중 HEVC/Screen Content Coding(HEVC/SCC)는 스크린 콘텐츠 압축 성능을 높이기 위하여 화면 내 블록 카피 기법 (Intra Block Copy, IBC)[3], 팔레트 부호화 모드(Palette Coding Mode, PCMD)[4], 적응적 색 공간 변환 기법(Adaptive Color Space Transform, ACST)[5] 등의 기술들을 포함하였다.

스크린 콘텐츠는 그래픽과 글자가 포함되는 특성상, 자연영상보다 더 많은 비슷한 무늬나 반복되는 무늬를 포함하고 있다. 이러한 특성을 고려하여 화면 내 블록 카피 기법이 제안되었다. 화면 내 블록 카피 기법은 지역적 탐색과 해시 기반의 전역적 탐색을 이용하여, 현재 압축하려는 블록에 대하여 화면 내에서 참조할 수 있는 예측 블록을 찾는다. 16x16 블록의 경우, 수평 또는 수직 방향의 1차원 지역적 탐색을 수행한다. 8x8 블록의 경우, 해시 기반 전역적 탐색을 수행하여, 화면 내에서 동일한 해시를 가진 블록들을 비교하여 예측 블록을 결정한다. 최종

적으로 화면 내 블록 카피 기법은 탐색을 통해 결정된 예측 블록의 위치를 블록 벡터(Block Vector)로 표현하게 된다.

해시기반 탐색에 사용되는 해시 값은 블록의 특성에 기반하여 계산된다. 16비트로 이루어지는 해시 값은 영상의 CU(Coding Unit) 내 4비트의 그라디언트 값과 CU 4분할 영역의 각 DC 값을 3비트로 표현하여 결정된다.[4]. 화면 내 먼저 부호화 및 복원된 영역에서 현재 CU와 같은 해시 값을 갖는 블록들을 비교함으로써, 현재 부호화 하려는 CU와 가장 유사한 블록을 탐색한다. 기존의 해시 값은 CU의 특징을 그라디언트 값과 분할된 영역의 DC값이 대표하고 있다. 영상의 블록 단위의 특징을 표현할 수 있는 새로운 정량적 수치를 해시 값 구성에 활용하거나 블록의 특징을 나타내는 여러 정량적 수치들의 비중의 재분배를 통해 새로운 해시 값 구성을 제안할 수 있고, 개선된 해시기반 탐색 결과를 기대할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 HEVC/SCC 테스트 모델에 소개된 기존 해시기반 탐색 방법을 설명하고, 3장에서는 해시 값 계산에 이용하기 위한 CU의 특징을 표현할 수 있는 새로운 정량적 수치를 제안하고, 여러 정량적 수치들의 가중치를 달리하여 구성할 수 있는 새로운 해시값 구성을 소개한다. 4장에서는 새로 제안한 해시값을 활용하는 해시기반 탐색기법을 포함한 화면 내 예측 기법의 성능을 실험하고 분석한다. 마지막 5장에서는 결론을 제시한다.

2. 기존 화면 내 블록 카피 기법

기존의 HEVC/SCC에서 제안된 화면 내 블록 카피 기법을 위해

사용되는 블록 벡터 탐색 방법은 CU와 PU 분할 방법에 따라 제한되어 결정된다. 블록 벡터 탐색은 탐색 범위에 따라 지역적 탐색과 전역적 탐색으로 나뉜다. CU와 PU 분할 방법에 따라 사용할 수 있는 탐색 방법을 모두 이용하여 블록 벡터 후보들이 결정된다.[3]

2.1 지역적 블록 벡터 탐색

CU의 크기가 16x16이고 PU 분할이 2Nx2N으로 이루어질 경우, 큰 블록 자체가 블록 내 단조롭게 반복되어 나타날 확률이 적다. 때문에 넓은 범위를 제한적으로 탐색하는 그림 1 (a)의 1D 수평·수직 탐색 방법을 이용한다.

CU의 크기가 8x8이고 PU 분할이 2Nx2N, 2NxN, 또는 NxN으로 이루어질 경우, 화면 내에서 앞서 복호화된 영역 중 왼쪽 CTU (Coding Tree Unit)과 현재 CTU에 제한하여 그림 1 (b)의 수평·수직 탐색 방법과 그림 1 (c)의 지역적 2D 탐색 방법을 이용한다. PU 분할이 Nx2N으로 이루어질 경우, 그림 1 (a)의 탐색 방법을 이용한다.



그림 1. (a) 전체 1D 수평·수직 탐색 방법 (b) 제한된 영역 내 1D 수평·수직 탐색 방법 (c) 제한된 영역 2D 탐색

2.2 해시 기반의 전역적 블록 벡터 탐색

해시 기반의 전역적 블록 벡터 탐색은 CU의 크기가 8x8이고 PU 분할이 2Nx2N으로 이루어질 경우에만 사용한다. CU 단위로 해시 값을 계산하고, 복호화된 영역에서 계산되었던 해시 값들 중에 동일한 해시 값을 갖는 CU들을 예측 블록의 후보로 결정한다. 이는 현재 CU보다 앞서 복호화된 전체 영역을 대상으로 간단하고 빠르게 탐색을 할 수 있는 방법이다.

기존의 화면 내 블록 카피 기법에서 사용되는 해시 값은 16비트의 값으로 계산되며, CU의 특징을 표현할 수 있는 DC값과 그라디언트 등을 이용하여 구성된다. 그 중 DC값은 현재 CU를 4분할 하여, 4x4 크기의 블록의 DC를 계산한 4개의 DC값을 사용한다. 부분 블록의 DC값을 계산하는 방법은 식 (1)과 같다.

$$DC_k = \frac{1}{4 \times 4} \sum_{j=0}^3 \sum_{i=0}^3 p(i, j) \quad (1)$$

그라디언트 값은 8x8 블록 전체를 통해 계산하며, 그 값은 식 (2)와 같이 계산된다.

$$Grad. = \sum_{j=1}^7 \sum_{i=1}^7 \frac{|p(i, j) - p(i-1, j)| + |p(i, j) - p(i, j-1)|}{2} \quad (2)$$

두 식에서 $p(i, j)$ 는 CU 내 (i, j) 위치의 화소값을 의미한다. 식

(1)의 DC_k 값 ($k=0,1,2,3$)은 8-bit 영상에서 계산될 경우, 부분 블록의 화소들의 평균값을 나타내는 값으로, 최소 0에서 최대 255의 8비트의 값으로 계산된다. 식 (2)의 $Grad.$ 값은 최소 0에서 최대 12,495 ($= 7 \times 7 \times 255$)의 값으로 계산된다. 각 값을 MSB(Most Significant Bits) 함수를 사용하여 양자화시켜 16비트의 해시 값을 구성하는데 사용한다. 해시 값을 구성하는 방법을 식으로 나타내면 다음과 같다. :

$$Hash = (MSB_3(DC_0) \ll 13) + (MSB_3(DC_1) \ll 10) + (MSB_3(DC_2) \ll 7) + (MSB_3(DC_3) \ll 4) + (MSB_4(Grad.)) \quad (3)$$

식 (3)의 $MSB_m(X)$ 는 X 를 표현하는 2진법의 비트스트림 중 최상위 m 비트만 취하여 값으로 만들어내는 함수이며, \ll 는 left shift 연산을 나타낸다. 해시 값의 비트의 구성을 도식화하여 나타내면 그림 2와 같다.

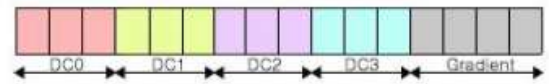


그림 2. 기존의 16비트 해시 값 구성

3. 제안하는 방법

해시 값은 CU의 특징을 표현 할 수 있는 부분 블록들의 DC값과 그라디언트 값을 사용한다. 해당 값들은 MSB 함수를 활용하여, 값 자체를 사용하는 것이 아닌 근사된 값을 사용한다. 근사하는 정도를 달리 하여 해시 구성을 새롭게 할 수 있다. 뿐만아니라, DC값, 그라디언트 값 외에 CU가 갖는 특징을 표현할 다른 값을 해시 구성을 위해 사용할 수 있다. 두 가지 변화를 통해 새로운 해시 값 구성을 제안하기 위해, 앞서 해시 구성시 특징 값들을 근사가 갖는 의미와 추가적으로 사용할 수 있는 특징 값을 다음과 같이 제안한다.

3.1 해시 구성 요소의 근사

예를 들어 완벽하게 4 분할 부분 블록들의 DC값이 일치하는 CU를 찾기 위해서는 각 4개의 DC값을 근사없이 사용하여 해시 값을 계산해야 한다. 하지만 해시 값을 구성하는 특징값들의 전체값을 저장하고, 비교하기에는 과도한 메모리 사용은 필요로 하게 된다. 따라서 해시 값을 16비트의 값으로 제한하여 CU의 특징을 표현할 수 있는 값들을 근사하여 해시 구성에 활용한다.

부분 블록의 DC값은 최소 0에서 최대 255인 8비트의 값이다. 해시 값 구성을 위해 MSB 함수를 사용하여 상위 3비트만 이용한다. 이는 하위 5비트 값이 표현하는 값은 근사시키는 효과를 갖고 있다. 예를 들어 MSB 함수를 사용하여 상위 4비트를 사용한다면, 기존의 3비트를 이용할 때보다 부분 블록의 DC값에 더 비중을 주었다는 의미이다.

기존의 해시 구성에서 사용하는 부분 블록의 DC값들과 그라디언트 값이 갖는 비중을 달리하여 아래와 같은 3가지 해시 값을 제안한다.

$$Hash = (MSB_3(DC_0) \ll 12) + (MSB_3(DC_1) \ll 9) + (MSB_3(DC_2) \ll 6) + (MSB_3(DC_3) \ll 3) + (MSB_3(Grad.)) \quad (4)$$

$$Hash = (MSB_2(DC_0) \ll 10) + (MSB_2(DC_1) \ll 8) + (MSB_2(DC_2) \ll 6) + (MSB_2(DC_3) \ll 4) + (MSB_4(Grad.)) \quad (5)$$

$$Hash = (MSB_2(DC_0) \ll 9) + (MSB_2(DC_1) \ll 7) + (MSB_2(DC_2) \ll 5) + (MSB_2(DC_3) \ll 3) + (MSB_3(Grad.)) \quad (6)$$

식 (4)는 기존의 해시 구성보다 그라디언트 값의 비중을 3비트로 낮추었다. 식 (5)는 기존의 해시 구성보다 부분 블록들의 DC값의 비중을 2비트로 낮추었다. 식 (6)은 기존의 해시 구성보다 부분 블록들의 DC 값의 비중을 2비트로, 그라디언트 값의 비중을 3비트로 낮추었다.

3.2 새로운 해시 구성 요소

3.1 절의 방법대로 해시를 구성하는 특징값들의 비중을 낮춘다면, 동일한 해시 값으로 계산되는 CU들이 더 많아지고, 해당 CU들에 대해 RD-cost를 비교하여 최적의 CU를 찾는 계산량도 많아지게 된다. CU의 특징으로 사용할 수 있는 새로운 해시 구성 요소를 포함한다면, 동일한 해시 값으로 탐색되는 CU를 더 제한하는 효과를 가질 수 있다.

화면 내 블록 카피 기법이 이용될 경우, 현재 블록과 예측 블록은 거의 비슷한 패턴을 가지고 있다. 따라서 블록 내 화소들이 갖는 히스토그램도 비슷한 양상을 갖고 있다. 따라서 블록의 히스토그램 정보도 CU의 특징을 표현할 수 있는 특징값으로 활용할 수 있다. 블록의 히스토그램에서 가장 많은 화소들이 포함된 히스토그램 구간을 새로운 해시 구성 요소로서 사용하고자 한다.

해시 기반 전역적 탐색은 8x8 크기의 CU에서 사용되기 때문에 총 64개의 화소가 나타내는 히스토그램을 비교하게 된다. 8비트 영상에서 화소 값은 최소 0부터 최대 255까지의 256개의 값의 히스토그램 구간을 갖고 있고, 이를 표현하기 위해서는 총 8비트가 필요하다. 히스토그램 최다 구간 값 또한 근사하여 사용하는 것이 필요하다. 예를 들어, 3비트로 근사하자면, 0부터 255까지의 화소값을 32개의 값씩 묶어 한 구간으로 결정한다면 총 8개의 구간이 되어, 3비트로 히스토그램 최다 구간을 표현할 수 있다.

식 (6)에서 제안된 해시 구성 방법에 히스토그램 최다 구간을 포함하여 아래와 같은 해시 값을 제안한다.

$$Hash = (MaxHisto(4) \ll 11) + (MSB_2(DC_0) \ll 9) + (MSB_2(DC_1) \ll 7) + (MSB_2(DC_2) \ll 5) + (MSB_2(DC_3) \ll 3) + (MSB_3(Grad.)) \quad (7)$$

$$Hash = (MaxHisto(3) \ll 11) + (MSB_2(DC_0) \ll 9) + (MSB_2(DC_1) \ll 7) + (MSB_2(DC_2) \ll 5) + (MSB_2(DC_3) \ll 3) + (MSB_3(Grad.)) \quad (8)$$

$$Hash = (MaxHisto(2) \ll 11) + (MSB_2(DC_0) \ll 9) + (MSB_2(DC_1) \ll 7) + (MSB_2(DC_2) \ll 5) + (MSB_2(DC_3) \ll 3) + (MSB_3(Grad.)) \quad (9)$$

$$Hash = (MaxHisto(1) \ll 11) + (MSB_2(DC_0) \ll 9) + (MSB_2(DC_1) \ll 7) + (MSB_2(DC_2) \ll 5) + (MSB_2(DC_3) \ll 3) + (MSB_3(Grad.)) \quad (10)$$

$MaxHisto(n)$ 은 히스토그램 최다 구간을 n 비트로 근사하여 나타낸 값을 의미한다.

새로 제안된 해시 구성 요소에 대하여 과한 비중을 둘 경우, 동일한 해시를 갖는 CU가 적어질 수 있기 때문에, 최적의 비중을 찾아야 한다. 따라서 식 (7)부터 식 (10)과 같이 다양한 비중을 통해 적절한 비중 값을 찾도록 한다.

4. 실험 및 실험결과

식 (4)부터 식(10)까지 제안된 새로운 해시 값 구성의 성능을 측정하기 위해 HEVC 참조 소프트웨어 SCM-6.1을 사용한다. [7] 실험을 위해 자연 영상(Class B, Class C) 4개, 스크린 콘텐츠 영상(Class F 및 CTC 영상) 4개, 총 8개의 실험 영상을 이용한다. 실험에 사용하는 양자화 계수(QP) 값은 {22, 27, 32, 37}을 이용하며, SCC CTC(Common Test Conditions)에서 정의된 AI(All Intra) configuration을 사용한다.

표 1은 기존의 SCM-6.1과 제안된 새로운 해시 값 구성들의 성능을 BDBR 단위(%)로 비교한 결과이다. 실험 1부터 실험 7은 각 식 (4)부터 식 (10)에 제시된 해시 값 구성을 활용하여 화면 내 블록 카피 기술을 사용한다.

3.1절에서 제안한 바와 같이, 해시 구성 요소들의 근사 정도를 더

카테고리	실험 영상	해상도	BDBR (%) : SCM-6.1 비교						
			해시 구성 요소의 근사 변화 (3.1절)			히스토그램 최다 구간을 활용한 해시 구성 (3.2절)			
			실험 1	실험 2	실험 3	실험 4	실험 5	실험 6	실험 7
자연 영상	BQTerrace	1920x1080	-0.050	-0.037	-0.083	-0.038	-0.065	-0.061	-0.080
	Kimono1		0.009	0.009	0.010	0.000	-0.012	-0.005	-0.003
	PartyScene	832x480	-0.012	-0.023	-0.020	0.047	-0.012	-0.003	0.020
	RaceHorsesC		0.007	-0.026	-0.010	0.000	-0.013	-0.024	-0.004
자연 영상 평균			-0.012	-0.019	-0.026	0.002	-0.026	-0.023	-0.012
스크린 콘텐츠 영상	sc_desktop_8bit_444	1920x1080	-0.100	-0.430	-0.562	-0.360	-0.584	-0.569	-0.495
	sc_web_browsing_8bit_444		-0.159	-0.234	-0.364	-0.802	-0.300	-0.619	-0.529
	SlideEditing	1280x720	-0.474	-1.156	-1.525	0.171	-1.109	-1.215	-1.419
	SlideShow		-0.062	-0.006	-0.081	-0.182	-0.081	0.000	-0.133
스크린 콘텐츠 영상 평균			-0.199	-0.457	-0.633	-0.293	-0.519	-0.601	-0.644
전체 평균			-0.106	-0.238	-0.330	-0.146	-0.273	-0.312	-0.328

표 1. SCM-6.1 대비 제안된 새로운 해시 구성을 사용하는 실험 결과

크게하는 실험 1부터 실험 3의 경우, 자연 영상에서는 최대 0.010% 비트레이트율 증가나 최대 0.050%의 비트레이트율 감소의 효과를 확인할 수 있다. 실험 1부터 실험 3에서 자연 영상의 경우, 최저 평균 0.012% 비트레이트율 감소(실험 1) 및 최대 평균 0.026% 비트레이트율 감소(실험 3) 효과를 확인할 수 있었다. Kimono1 영상의 경우 비트레이트율 증가가 주로 이루어져 제안하는 방법을 사용할 경우 손해가 주로 나타났으나, 그 정도가 무시할 수 있을만큼 적다. 뿐만아니라, 다른 영상에서 나타나는 비트레이트 감소 효과도 매우 적게 나타난다. 스크린 콘텐츠 영상의 경우, 최저 0.006% 비트레이트율 감소의 미미한 효과를 얻을 때도 있지만, 최대 1.525% 비트레이트율 감소의 효과를 얻을 수 있었다. 실험 1부터 실험 3에서 스크린 콘텐츠 영상의 경우, 최저 평균 0.106% 비트레이트율 감소(실험 1) 및 최대 평균 0.330% 비트레이트율 감소(실험 3) 효과를 확인할 수 있다.

3.2절에서 제안한 히스토그램 최다 구간을 활용하는 헤시 구성을 활용하는 실험 4부터 실험 7의 경우, 자연 영상에서는 최대 0.047% 비트레이트율 증가와 최대 0.080% 비트레이트율 감소의 효과를 확인할 수 있었다. 실험 4부터 실험 7에서 자연 영상의 경우, 최대 평균 0.002% 비트레이트율 증가(실험 4) 및 최대 평균 0.026% 비트레이트율 감소(실험 5) 효과를 확인할 수 있다. 실험 4의 경우 비트레이트율이 증가되는 실험 결과들이 주로 확인되나 이 정도의 손해는 무시할 수 있을만큼 적으며, 다른 실험에서 얻어지는 비트레이트율 감소 효과도 적다. 스크린 콘텐츠 영상의 경우, 최대 0.171% 비트레이트율 증가나 최대 1.419% 비트레이트율 감소 효과를 확인할 수 있었다. 평균적으로 최저 0.146% 비트레이트율 감소(실험 4)와 최대 0.328% 비트레이트율 감소(실험 7)의 효과를 확인할 수 있다.

실험 결과 더 근사된 구성 요소들을 헤시 구성에 사용할 경우, 최저 0.106% 비트레이트율 감소 효과 및 최대 0.330% 비트레이트율 감소 효과를 확인할 수 있었다. 더 근사된 구성 요소를 사용할 경우, 동일한 헤시 값들 갖을 수 있는 블록을 많이 갖게 되고, 그 중 최적의 블록을 선택하게 되어 개선 효과가 나타난다.

블록의 특징을 나타내는 새로운 헤시 구성 요소로 히스토그램 최다 구간을 사용할 경우, 최저 0.146 % 비트레이트율 감소 및 최대 0.328% 비트레이트율 감소 효과를 확인할 수 있었다. 추가적인 헤시 구성 요소가 없는 실험 3에 비해 동일한 헤시가 나타나는 블록들의 후보가 줄어 효과가 더 좋게 되었다. 전체적으로는 새로 추가되는 헤시 구성 요소의 비중이 적을수록 좋은 효과를 보이고 있으나, 영상별로 비교해 보았을때에는 단순히 비중이 적어질수록 개선된 효과를 보이지 않기도 했다.

5. 결론

이 논문은 화면 내 블록 카피 기법에서 헤시 기반에 전역적 탐색 방법에 이용되는 헤시 값 구성을 달리하는 방법을 통해 개선된 효과를 확인할 수 있었다. 기존에 4비트로 근사된 그라디언트 값과 각 3비트로 근사된 부분 블록 DC값을 헤시 구성에 이용하는 SCM-6.1에 비해, 더 근사된 구성 요소들을 활용하는 방법과 히스토그램 최다 구간을 새로운 헤시 구성요소로 사용하는 방법을 제시했다. 실험 결과 더 근사된 구성 요소들을 헤시 구성에 활용하는 경우 0.330% 비트레이트율 감소 효과를 확인할 수 있으며, 히스토그램 최다 구간을 새로운 헤시 구성요

소로 사용하는 경우 0.328%의 비트레이트율 감소효과를 확인할 수 있었다.

헤시를 구성하는 요소들의 비중을 단순히 낮출 경우, 선택되는 블록이 많아져 더 좋은 예측을 할수 있지만, RD-cost를 비교할 대상이 증가하므로 복잡도가 증가하는 것을 예상할 수 있다. 그리고 추가적인 헤시 구성 요소를 사용하는 방법을 통해 선택되는 블록을 효과적으로 제한할 수 있는 방법을 기대할 수 있다. 이에 따라 인코딩 시간 등 복잡도를 고려하는 다양한 헤시 구성을 통해, 최적의 헤시 구성을 고려하는 실험을 진행해야 할 것이다.

감사의 글

“이 연구는 방위사업청 및 국방과학연구소의 재원에 의해 설립된 신호정보 특화연구센터 사업의 지원을 받아 수행되었음.”

참고문헌

- [1] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp.1649-1668, December 2012.
- [2] D. Flynn, M. Naccari, C. Rosewarne, K. Sharman, J. Sole, G. Sullivan, and T. Suzuki, “High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 7, Document JCTVC-Q1005,” in *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*, April 2014.
- [3] C. Pang, J. Sole, L. Guo, M. Karczewicz, and R. Joshi, “intra Motion Compensation with 2-D MVs, Document JCTVC-N0256,” in *ISO/IEC/JTC1/SC29/WG11 and ITU-T SG16 Q.6*, Aug. 2013.
- [4] Patrice Onno, “Suggested combined software and text for run-based palette mode,” Document JCTVC-R0348, July 2014.
- [5] L. Zhang, J. Chen, J. Sole, and M. Karczewicz, “JCTVC-Q0112: in-loop color-space transform,” in *ISO/IEC/JTC1/SC29/WG11 and ITU-T SG16 Q.6*, March 2014.
- [6] R. Joshi, J. Xu, R. Cohen, S. Liu, and Y. Ye, “Screen Content Coding Test Model 6 Encoder Description (SCM 6) Document JCTVC-V1014,” in *ISO/IEC/JTC1/SC29/WG11 and ITU-T SG16 Q.6*, February 2016.
- [7] HM-10.1+RExt-3.0 software, Developed by JCT-VC, https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.7+SCM-6