

VVC 화면간 예측 부호화 기술

□ 강정원, 이하현 / 한국전자통신연구원

요약

VVC(Versatile Video Coding)는 ISO/IEC MPEG과 ITU-T VCEG으로 구성된 JVET(Joint Video Experts Team)에서 개발 중인 새로운 비디오 압축 표준 기술로 HEVC 대비 2배 압축률을 목표로 다양한 기술들이 채택되었다. 본 고에서는 VVC에 채택된 부호화 기술들 가운데 화면간 예측 기술에 대해 설명하고자 한다. VVC는 기존 부호화 표준인 HEVC의 화면간 예측 기술을 확장하고, 복호 과정에서의 움직임 벡터 탐색 및 계산 과정을 통해 전송받은 움직임 벡터를 보정하는 디코더 기반 움직임 벡터 보정 기술들을 채택하였다. 추가로, 확대, 축소, 회전 등과 같은 움직임을 예측할 수 있는 affine 움직임 모델 기반 움직임 예측 기술인 AFFINE 기반 움직임 벡터 예측 기술을 채택하였다. 이러한 기술들의 채택을 통해 VVC는 화면간 예측 정확도를 개선하였다.

1. 서론

국제 비디오 표준화 기구인 ISO/IEC SC29/WG11 MPEG(Moving Picture Experts Group)과 ITU-T VCEG(Video Coding Experts Group)이 공동으로 JVET(Joint Video Experts Team)을 설립하여 HEVC(High Efficiency Video Coding)[1] 대비 2배 압축률을 목표로 VVC(Versatile Video Coding) 표준화를 진행 중에 있다. VVC의 표준화 일정은 2019년 7월 CD(Committee Draft) 단계를 완료하였고, 2019년 10월 DIS(Draft International Standard) 단계를 거쳐 2020년 7월 FDIS(Final DIS) 단계로 표준화를 완료할 예정으로, 압축률 목표 달성을 위하여 영상의 분할 구조, 화면내 예측, 화면간 예측, 변환, 루프 필터 등에서 다양한 부호

※ 본 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2016-0-00572), 초고실감 미디어 서비스 실현을 위해 HEVC/3DA 대비 2배 압축을 제공하는 5세대비디오/오디오 표준 핵심 기술 개발 및 표준화

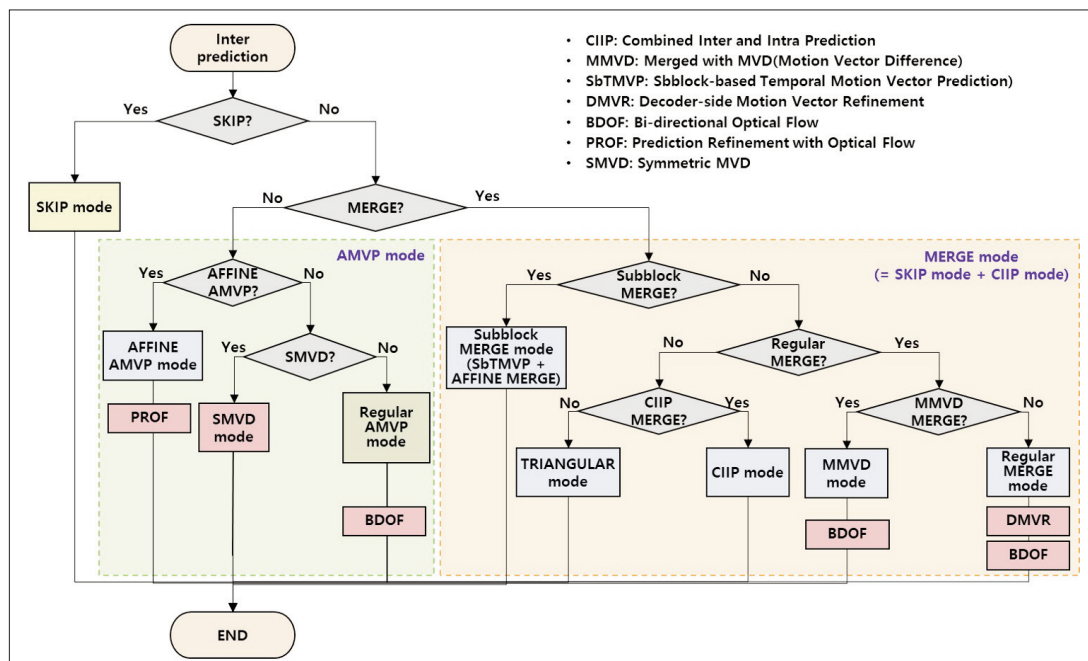
화 기술들을 채택하였다.

본 고는 VVC의 CD[2] 및 VVC의 참조 소프트웨어인 VTM(VVC Test Model) 6.0[3]에 채택된 다양한 부호화 기술들 가운데 화면간 예측 부호화 기술들에 대해 설명한다.

II. VVC 화면간 예측 기술

VVC는 <그림 1>에 나타난 바와 같이 다양한 화면간 예측 기술들을 채택하였다. SKIP 및 MERGE 모드 측면에서는 HEVC와 유사한 MERGE 모드인 일반 MERGE(Regular MERGE) 모드에 추가로 MMVD(Merge with Motion Vector Difference) 모드[4], CIIP(Combined Inter and Intra Prediction) 모드[5], TRIANGULAR 모드[6],

SbTMVP(Sbblock-based Temporal Motion Vector Prediction)[7]와 AFFINE MERGE 모드[8]를 포함하는 서브블록 기반 MERGE 모드가 채택되었다. AMVP(Advanced MVP) 모드 측면에서는 HEVC와 유사한 일반 AMVP(Regular AMVP) 모드에 SMVD(Symmetric MVD) 모드[9]와 AFFINE AMVP[8] 모드가 채택되었다. 또한, 이와 같은 MERGE 및 AMVP 모드 이외에 MERGE 혹은 AMVP에 추가로 적용하여 부호화 효율을 개선하는 디코더 기반 움직임 벡터 보정 기술들인 DMVR(Decoder-side MV Refinement)[10], BDOF(Bi-Directional Optical Flow)[11][12], PROF(Prediction Refinement with Optical Flow)[13]들이 추가되었다. 본 절에서는 앞에서 언급한 화면간 예측 기술을 포함한 VVC 주요 화면간 예측 기술에 대해 설명한다.



<그림 1> VVC 화면간 예측 흐름도

1. SKIP 및 MERGE 모드기반 부호화 기술

개념적으로 VVC의 SKIP 및 MERGE 모드는 HEVC의 그것들과 동일한 의미를 가지며, 전송되는 비트량을 최소화하면서 예측 정확도를 높이기 위해 HEVC에 채택된 SKIP/MERGE 모드를 확장한 다양한 SKIP/MERGE 모드 기반 부호화 기술들을 채택하였다.

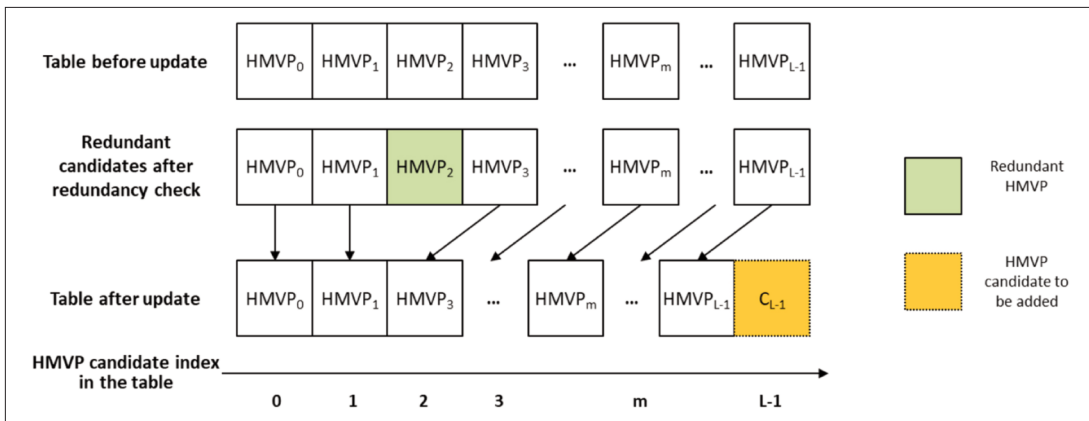
VVC의 MERGE 모드는 크게 블록 기반 MERGE 모드와 서브블록 기반 MERGE 모드로 구성되며, 블록 기반 MERGE 모드는 일반 MERGE 모드, MMVD 모드, CIIP 모드, TRIANGULAR 모드로 구분될 수 있다. 또한, VVC의 SKIP 모드는 CIIP 모드를 제외한 모든 MERGE 모드가 적용할 수 있다.

1) 일반 MERGE 모드

일반 MERGE 모드는 HEVC의 MERGE 모드와 동일하게 후보 리스트를 구성하고 구성된 후보 리스트 가운데 수신된 인덱스가 가리키는 특정 후보의 움직임 정보를 해당 블록(CU, Coding Unit)의 움직임 정보로 사용하여 움직임 보상을 수행한다.

한편, 후보 리스트는 HEVC의 MERGE 모드와 동일한 공간적 MERGE 후보(spatial merge candidate), 시간적 MERGE 후보(temporal merge candidate)에 HMVP(History-based MVP)[14]와 PAMVP(Pairwise Average MVP)[15]를 추가하여 부호화 효율을 개선하였다.

HMVP는 이전 N개 CU의 움직임 정보를 테이블에 저장하여 현재 CU를 위한 MVP로 사용하는 방법으로, HEVC 표준화 이후 진행된 탐색 실험에서 부호화 효율 개선을 위해 제안되었던 공간적 MERGE 후보를 확장하는 기술들에 비해 단순하면서 부호화 효율이 높다는 장점으로 인해 채택되었다. VVC에 채택된 HMVP 테이블의 크기(L)는 5로 5개의 HMVP 후보를 저장하며, HMVP 테이블에 5개의 후보가 존재하는 경우에 새로운 움직임 정보를 저장하기 위하여 선입선출(FIFO, First-In-First-Out) 방법을 적용한다. 더불어, HMVP 테이블에 동일한 움직임 정보가 중복되어 저장되는 것을 방지하기 위해서 새로 추가하는 움직임 정보와 HMVP 테이블에 저장되어 있는 후보들과의 중복성을 확인하여 동일한 움직임 정보가 HMVP 테이블



〈그림 2〉 HMVP 테이블 갱신 과정

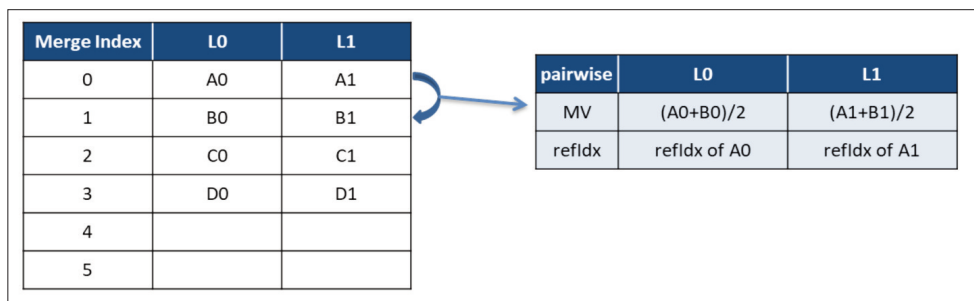
블에 존재하는 경우에는 해당 움직임 정보를 삭제하고 새로운 움직임 정보를 HMVP 테이블의 마지막 위치에 추가한다(〈그림 2〉). 또한, 영상의 새로운 CTU(Coding Tree Unit) 열이 시작되는 시점에 HMVP 테이블은 초기화되며, 서브블록 기반 화면간 예측 방법이 적용된 CU의 움직임 정보는 HMVP 후보로 사용하지 않는다.

PAMVP는 HEVC의 combined bi-predictive merge MVP를 대체하는 톨로써 combined bi-predictive merge MVP 대비 부호화 효율을 개선하여 채택되었다. PAMVP는 구성된 MERGE 후보 리스트에서 첫 번째 후보와 두 번째 후보의 움직임 벡터 평균값을 움직임 벡터로 설정하고 참조픽처는 첫 번째 후보와 동일하게 설정하여 얻어진다(〈그림 3〉).

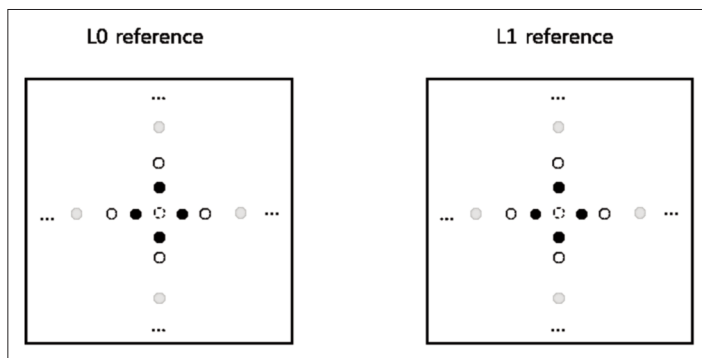
2) MMVD 모드

일반 MERGE 모드는 전송되는 비트량을 최소화하는 반면 예측신호가 정확하지 않을 수 있다는 단점이 있다. 이러한 단점을 보완하기 위하여 VVC는 일반 MERGE 모드 방법으로 유도된 움직임 벡터에 움직임 벡터 차분(MVD)을 더하여 보다 정확한 움직임 벡터를 유도할 수 있도록 하는 MMVD 모드를 MERGE 모드의 한가지 유형으로 채택하였다.

MMVD 모드는 부호화기 복잡도를 줄이기 위하여 일반 MERGE 모드의 후보 리스트 가운데 처음 2개 후보만 적용 가능하다. 2개 후보 가운데 전송받은 플래그가 가리키는 후보의 움직임 벡터를 초기 움직임 벡터로 하고, 추가로 전송받은 MVD 방향과 움직임 거리(MVD 절댓값) 정보로부터 유도된 MVD



〈그림 3〉 PAMVP 생성 과정



〈그림 4〉 MMVD 모드에서 벡터 차분 값 탐색 위치

〈표 1〉 움직임 방향 인덱스와 MVD 방향 관계

움직임 방향 인덱스	00	01	10	11
x-axis	+	-	N/A	N/A
y-axis	N/A	N/A	+	-

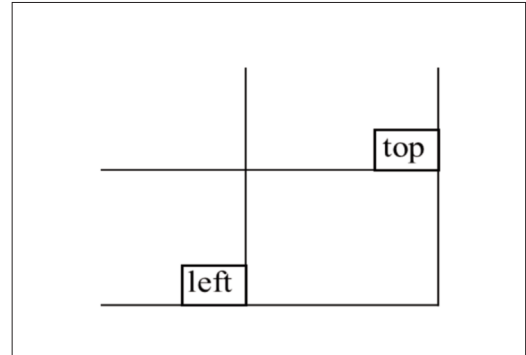
를 초기 움직임 벡터에 더하여 최종 움직임 벡터를 획득한다. MVD를 유도하기 위하여 전송되는 MVD 방향과 움직임 거리는 미리 정의된 테이블에 기반한다. MVD 방향은 〈그림 4〉와 같이 수평 또는 수직 방향만 가능하며, 이를 나타내는 인덱스와 MVD 방향의 관계는 〈표 1〉과 같다. 그리고, 움직임 거리는 〈표 2〉에 표현된 바와 같이 1/4-pel부터 32-pel의 범위를 가질 수 있다.

3) CIIP 모드

일반적으로 한 개 이상 예측신호들의 가중합으로 얻어진 예측신호는 부호화 효율을 개선하는 효과가 있다. 쌍방향(bi-prediction) 예측에 의한 부호화 효율 개선도 이러한 맥락으로 설명할 수 있으며, 가중합되는 예측신호의 수를 늘릴수록 부호화 효율이 개선되는 경향을 보인다. 그러나, 메모리 대역폭 및 계산 복잡도의 한계로 인해 비디오 부호화 표준에서는 쌍방향 화면간 예측과 같이 2개 예측신호의 가중합 이상을 허용하지 않는다. 따라서 VVC는 메모리 대역폭을 증가시키지 않으면서 예측신호 가중합에 의한 부호화 효율 개선을 위해 화면간 예측신호와 화면내 예측신호를 결합하는 CIIP 모드를 채택하였다. CIIP 모드에서 화면간 예측신호는 일반 MERGE 모드와 동일한 과정으로 유도되고, 화면내 예측신호는 planar 모드를 적용하여 유도된다. 유도된 두 예측신호에 가중치를 적용하여 최종 예측신호를 구성하며, 적용되는 가중치인 (화면내 예측신호 가중치, 화면간 예측신호 가중치)는 〈그림 5〉의

〈표 2〉 움직임 거리 인덱스와 움직임 거리와의 관계

움직임 거리 인덱스	0	1	2	3	4	5	6	7
움직임 거리 (최소 샘플 단위)	1/4	1/2	1	2	4	8	16	32



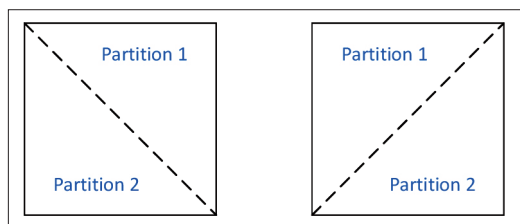
〈그림 5〉 CIIP 가중치 유도를 위해 사용되는 상단 및 좌측 주변 블록 위치

상단과 좌측 주변 블록의 화면내 예측 부호화 여부를 참고하여 다음과 같이 유도된다. 상단과 좌측 주변 블록이 모두 화면내 예측인 경우는 (3,1), 상단 혹은 좌측 주변 블록 중 하나의 블록만 화면내 예측인 경우는 (2,2), 상단과 좌측 주변 블록이 모두 화면내 예측이 아닌 경우는 (1,3)의 가중치를 적용한다.

4) TRIANGULAR 모드

VVC의 블록분할 구조는 삼각분할을 허용하지 않는 반면, 화면간 예측 시 삼각분할 블록과 유사한 효과를 주기 위하여 삼각분할 기반 화면간 예측 모드인 TRIANGULAR 모드를 블록 기반 MERGE 모드의 한가지 유형으로 채택하였다.

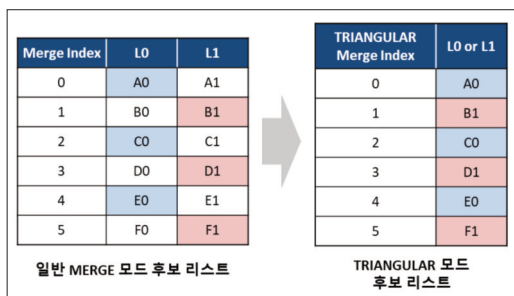
TRIANGULAR 모드인 경우, CU는 〈그림 6〉과 같이 두 가지 형태의 삼각형 모양으로 분할되고, 분할된 2개의 영역은 서로 다른 움직임 정보를 이용하여 독립적으로 화면간 예측을 수행한다. 이 때, 삼각분할된 2개 영역의 움직임 정보가 다름으로 인



〈그림 6〉 TRIANGULAR 모드에 적용되는 삼각분할 유형

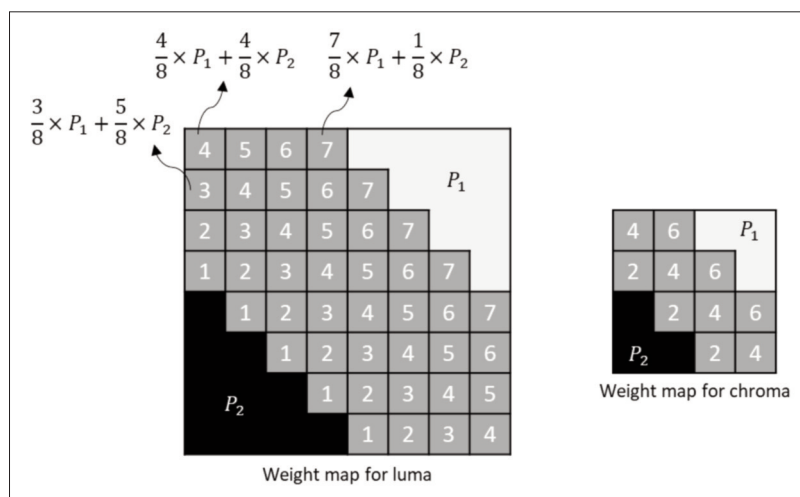
해 발생하는 삼각분할 경계영역에서의 예측신호 값들간의 불연속성을 최소화하기 삼각분할 경계영역의 예측신호는 2개 영역의 움직임 정보를 이용하여 유도된 예측신호 값들의 가중합으로 최종 예측신호를 유도한다.

TRIANGULAR 모드는 CU의 화면간 예측으로 인한 복잡도를 일반적인 쌍방향 예측과 동일하게 유지하기 위하여 각 분할 영역은 단방향 예측만 허용하며, 이를 위해 각 분할 영역의 단방향 움직임 정보는 일반 MERGE 모드 후보 리스트로부터 다음과 같이 유도된다. 〈그림 7〉과 같이 구성된 일반 MERGE 모드의 후보 리스트 가운데, 각 MERGE



〈그림 7〉 TRIANGULAR 모드 후보 리스트 생성 과정

인덱스가 짝수인 후보는 L0 리스트의 움직임 정보를, 홀수인 후보는 L1 리스트의 움직임 정보를 이용하여 단방향 움직임 정보만을 가지는 TRIANGULAR 모드 후보 리스트를 구성하며 만약 MERGE 인덱스가 짝수인 후보에 L0 리스트 움직임 정보가 존재하지 않는다면 L1 리스트 움직임 정보를 적용한다. 이렇게 구성된 TRIANGULAR 모드 후보 리스트 가운데, 각 분할 영역별로 적용할 MERGE 인덱스를 전송받아 각 분할 영역별로 화면간 예측을 수행하게 된다. 이때, 삼각분할 경계영역은 각 분할 영역



〈그림 8〉 삼각분할 경계에서 최종 예측신호 유도를 위한 가중치의 예

의 움직임 정보를 이용하여 두 개의 예측신호를 유도하고 유도된 예측신호들의 가중합으로 최종 예측 신호를 유도하며, 두 개의 예측신호에 적용하는 가중치는 <그림 8>과 같다.

5) 서브블록 기반 MERGE 모드

서브블록 기반 MERGE 모드는 CU를 sub-CU로 분할하고 sub-CU 단위로 다른 움직임 벡터를 유도하여 예측신호를 생성하는 모드로서 블록을 분할하지 않고 작은 블록단위의 움직임 벡터를 유도하여 예측을 수행함에 의해 예측 정확도를 높이는 효과를 얻는 것을 목적으로 하는 부호화 방법이다.

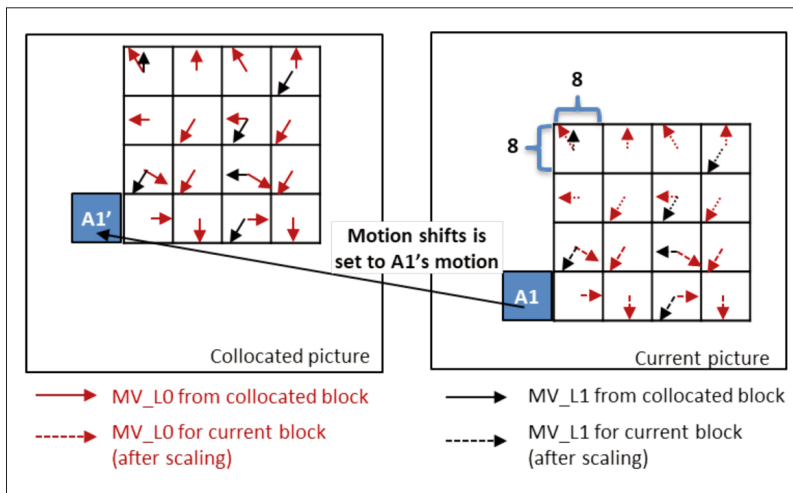
서브블록 기반 MERGE 후보 리스트는 블록 기반 MERGE 모드 가운데 시간적 MERGE 및 AMVP 후보인 TMVP(Temporal MVP)와 유사한 서브블록 기반 TMVP(SbTMVP)와 affine 움직임 모델을 적용한 AFFINE 기반 움직임 벡터 예측 후보들로 구성된다. SbTMVP는 MERGE 모드에만 적용되는 기술로 본 절에서 설명하고, AFFINE 기반 움직임

벡터 예측 기술은 MERGE 모드와 AMVP에 모두 적용되는 부호화 기술로 “3 AFFINE 기반 움직임 벡터 예측 기술” 절에서 자세히 다루도록 한다.

SbTMVP는 현재 CU의 대응블록을 찾는 과정과 대응블록의 움직임 벡터로부터 현재 CU내의 sub-CU 단위 움직임 벡터를 유도하는 과정으로 이루어지며, 각 과정은 다음과 같다.

현재 CU의 대응블록은 슬라이스 헤더에서 전송되는 대응참조픽처(collocated picture)에 존재하는 블록으로 <그림 9>에서 나타내는 바와 같이 현재 CU의 좌하단 위치(A1)의 참조픽처가 대응참조픽처와 동일한 경우에 A1 위치의 움직임 벡터를 이용하고, 동일하지 않은 경우에는 zero MV를 이용하여 대응블록 위치를 찾는다.

대응블록으로부터 현재 CU의 sub-CU(8x8 크기) 단위 움직임 벡터를 유도하기 위해서, 현재 CU를 8x8 sub-CU로 분할하고, 대응블록에서 각 sub-CU의 중심 위치에 대응되는 움직임 정보를 이용하여 각 sub-CU의 움직임 벡터를 유도한다.



<그림 9> Sub-CU 단위 움직임 벡터 유도 과정의 예

SbTMVP의 참조픽처는 TMVP 경우와 동일하게 L0/L1 참조픽처 리스트 인덱스 0이 가리키는 픽처이므로, 대응블록에서 유도된 움직임 벡터가 sbTMVP 참조픽처에 적용될 수 있도록 스케일링 과정을 거쳐서 sub-CU단위 움직임 벡터인 SbTMVP로 설정된다.

2. AMVP 모드 기반 화면간 부호화 기술

HEVC의 AMVP는 SKIP 및 MERGE 모드와 비교하여 보다 정확한 움직임 벡터를 복호 과정에서 유도할 수 있도록 주변 블록으로부터 얻어진 움직임 벡터 예측 값과 최종 움직임 벡터 값의 차이 값인 MVD 값을 전송하는 모드이다. VVC의 AMVP 모드는 HEVC의 AMVP 모드와 개념적으로 동일한 의미를 가지나 부호화 효율을 개선하기 위하여 HEVC와 유사한 일반 AMVP 모드에 SMVD 모드와 affine 움직임 모델을 적용한 AFFINE AMVP 모드를 추가하였다.

본 절에서는 일반 AMVP 및 SMVD 모드에 추가로, VVC AMVP 모드에 적용되는 부호화 틀인 AMVR (Adaptive Motion Vector Resolution)[16][17]과 BCW(Bi-prediction with CU-level Weight)[18]에 대해 다루고, AFFINE AMVP 모드는 “3. AFFINE 기반 움직임 벡터 예측 기술” 절에서 자세히 다루도록 한다.

1) 일반 AMVP 모드

일반 AMVP 모드는 예측신호 생성을 위해 예측 유형(단방향 혹은 쌍방향), 참조픽처, 움직임 벡터 예측 후보를 나타내는 플래그, MVD 정보를 부호화하고 복호 과정에서 부호화된 정보를 이용하여 유도된 최종 움직임 벡터와 참조픽처를 이용하여 예

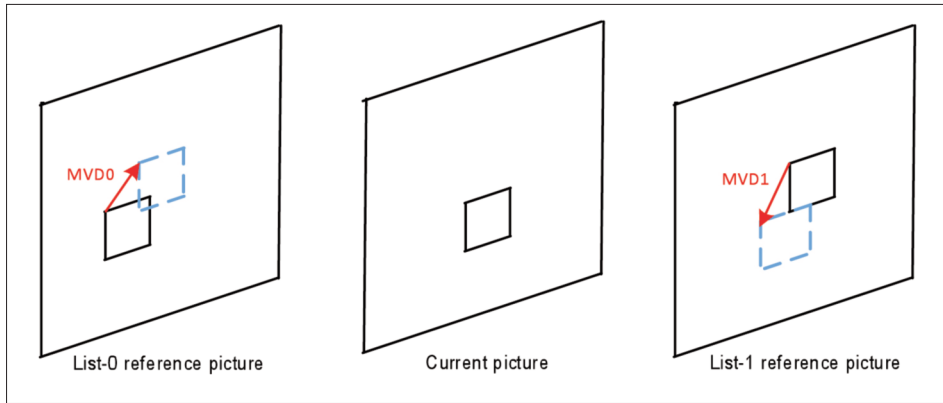
측신호를 생성한다는 점에서 HEVC의 AMVP 모드와 동일하다. 반면, AMVP 후보 리스트 구성 시 HEVC와 동일한 공간적 AMVP 후보와 시간적 AMVP 후보에 HMVP 후보를 추가하여 최대 2개의 움직임 벡터 예측 후보를 구성한다는 점에서 차이가 있다. 이때 사용되는 HMVP 후보는 SKIP 및 MERGE 모드에서 사용한 HMVP와 동일하다.

2) SMVD 모드

SMVD는 명칭에서 알 수 있듯이 양방향 예측(bi-directional prediction)의 경우에 L0 방향과 L1 방향의 MVD 값이 대칭을 이루도록 하는 모드로서 일반 AMVP 모드에 비해 전송되는 비트량을 줄이면서 SKIP 및 MERGE 모드에 비해 예측신호 정확도를 높이기 위한 방법이다. 전송되는 비트량을 줄이기 위하여, L0 방향과 대칭을 이루는 L1 방향 MVD 정보는 전송하지 않으며, 더불어 L0 및 L1 참조픽처 정보도 전송하지 않고 다음과 같이 복호 과정에서 유도하도록 한다.

SMVD는 양방향 예측에 제한적으로 적용되는 모드이기 때문에 현재 픽처를 기준으로 디스플레이 순서상 서로 반대방향에 위치하는 L0 참조픽처와 L1 참조픽처 조합을 찾아 SMVD 모드의 L0 및 L1 참조픽처로 설정한다. 따라서, 현재 픽처를 기준으로 디스플레이 순서상 서로 반대방향에 위치하는 L0 참조픽처와 L1 참조픽처 조합이 존재하지 않는 경우에는 SMVD 모드를 적용하지 못한다.

〈그림 10〉에 나타난 바와 같이 설정된 L0 및 L1 참조픽처와 AMVP 후보 리스트로부터 유도된 L0 및 L1 방향 움직임 벡터 예측 값과 전송받은 L0 방향 MVD 값(MVD0)과 MVD0를 이용하여 유도한 L1 방향 MVD 값(MVD1=-MVD0)을 적용하여 SMVD 모드의 최종 예측신호를 획득한다.



〈그림 10〉 SMVD 모드의 움직임 벡터 예시

3) AMVR

HEVC의 경우 MVD가 항상 1/4 화도 샘플 해상도를 가지는 반면, VVC는 MVD를 부호화하는데 소요되는 비트량을 절감하기 위하여 MVD 해상도를 적응적으로 선택하는 AMVR 기술을 채택하였다. 〈표 3〉에 나타난 바와 같이 부호화 모드에 따라 적용 가능한 MVD 해상도는 다르며, CU에 적용하는 MVD 해상도는 두 개의 선택스(amvr_flag, amvr_precision_idx) 형태로 전송된다.

유도된 움직임 벡터 예측 값에 MVD를 더하여 얻어지는 최종 움직임 벡터가 MVD와 동일한 해상도를 가지도록 하기 위하여 움직임 벡터 예측 값은 MVD와 동일한 해상도를 가지도록 라운딩을 수행한다. 유도된 최종 움직임 벡터를 이용하여 움직임 보상 수행 시 1/2 샘플 위치에 대해서는 6탭 보간

필터를 적용하고 1/2 샘플 위치 이외의 샘플 위치에 대해서는 8탭 보간 필터를 적용한다.

4) BCW

HEVC는 쌍방향 예측의 경우 두 개의 움직임 벡터에 의해 유도된 두 예측신호를 평균하여 최종 예측신호를 획득하나, VVC는 부분적 밝기 변화 등의 경우에 발생할 수 있는 부호화 효율 저하를 개선하기 위하여 쌍방향 예측의 경우 다음의 식과 같이 두 예측신호의 가중합을 최종 예측신호로 유도한다.

$$P_{bi-pred} = ((8-w) * P_0 + w * P_1 + 4) \gg 3 \quad (1)$$

저지연 부호화시에는 5개 가중치($w \in \{-2, 3, 4, 5, 10\}$), 저지연 부호화가 아닌 경우에는 3개의 가중치

〈표 3〉 부호화 모드에 따른 MVD 해상도

amvr_flag	amvr_precision_idx	AFFINE AMVP 모드	IBC 모드	일반 AMVP 모드
0	-	1/4 화도 샘플	-	1/4 화도 샘플
1	0	1/16 화도 샘플	1-정수 화도 샘플	1/2 화도 샘플
1	1	1-정수 화도 샘플	4-정수 화도 샘플	1-정수 화도 샘플
1	2	-	-	4-정수 화도 샘플

($w \in \{3, 4, 5\}$)를 적용할 수 있으며, CU단위로 적용하는 가중치는 AMVP 모드의 경우 CU단위로 전송되는 가중치 인덱스에 의해 결정되며, MERGE 모드의 경우 MERGE 후보 인덱스에 기반하여 결정된다.

3. AFFINE 기반 움직임 벡터 예측 기술

HEVC를 포함한 기존 비디오 부호화 표준들은 상하좌우의 평행이동만을 고려한 움직임 보상을 수행하기 때문에, 현실에서 일반적으로 접하는 확대, 축소, 회전 등과 같은 움직임을 포함하고 있는 비디오들의 부호화 시 부호화 효율이 저하된다. VVC는 이러한 문제를 해결하기 위하여 4개 또는 6개 파라미터를 이용하는 affine 움직임 모델 기반 움직임 예측 기술을 채택하였다.

〈그림 11〉은 CPMV(Control Point MV) 기반 affine 움직임 모델을 표현한 것으로 두 개의 CPMV를 사용하는 4-파라미터 affine 움직임 모델과 세 개의 CPMV를 사용하는 6-파라미터 affine 움직임 모델을 나타낸다.

4-파라미터 affine 움직임 모델의 경우, CU내

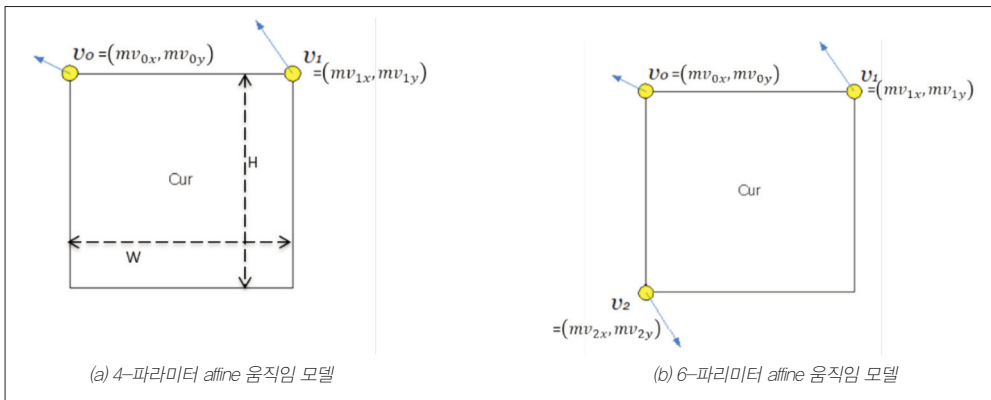
(x, y)위치에서의 움직임 벡터는 다음과 같이 유도된다.

$$\begin{cases} mv_x = \frac{mv_{1x} - mv_{0x}}{W}x + \frac{mv_{1y} - mv_{0y}}{W}y + mv_{0x} \\ mv_y = \frac{mv_{1y} - mv_{0y}}{W}x + \frac{mv_{1x} - mv_{0x}}{W}y + mv_{0y} \end{cases} \quad (2)$$

6-파라미터 affine 움직임 모델의 경우, CU내 (x, y)위치에서의 움직임 벡터는 다음과 같이 유도된다.

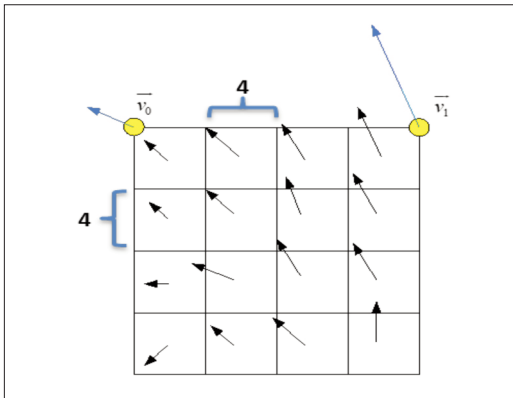
$$\begin{cases} mv_x = \frac{mv_{1x} - mv_{0x}}{W}x + \frac{mv_{2x} - mv_{0x}}{H}y + mv_{0x} \\ mv_y = \frac{mv_{1y} - mv_{0y}}{W}x + \frac{mv_{2y} - mv_{0y}}{H}y + mv_{0y} \end{cases} \quad (3)$$

앞의 두 식이 보여주듯이 CPMV로부터 affine 움직임 모델을 유도할 수 있고, 유도된 affine 움직임 모델에 기반하여 CU내 모든 샘플 위치에서의 움직임 벡터를 계산할 수 있다. 그러나, VVC는 픽셀 단위 움직임을 적용한 움직임 예측과 보상 과정의 복잡도를 줄이기 위하여 4x4 단위 블록 기반 움



〈그림 11〉 CPMV 기반 affine 움직임 모델

직임 예측과 보상을 수행한다. 즉, CU를 4x4 sub-CU 단위로 분할하고 각 sub-CU 중심 위치에서의 움직임 벡터를 affine 움직임 모델 기반으로 유도하여 각 sub-CU 단위 움직임 예측 및 보상을 수행한다(〈그림 12〉).



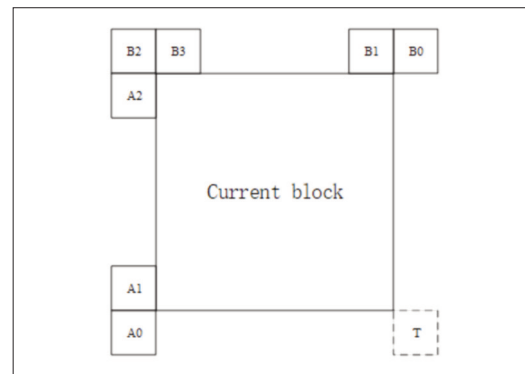
〈그림 12〉 Sub-CU단위 AFFINE 기반 움직임 벡터의 예

VVC에는 AFFINE 기반 움직임 벡터 예측 기술을 적용한 AFFINE MERGE 모드와 AFFINE AMVP 모드가 존재한다.

AFFINE MERGE 모드는 서브블록 기반 MERGE 모드의 후보 리스트에 AFFINE 기반 움직임 벡터 예측 후보들을 포함시켜서 현재 CU의 움직임 보상에 적용할 수 있도록 하는 방법으로, AFFINE 기반 움직임 벡터 예측 후보들은 상속 AFFINE MERGE 후보들과 조합 AFFINE MERGE 후보들로 구성된다. 상속 AFFINE MERGE 후보(inherited affine merge candidate)는 현재 CU의 상단 혹은 좌측 주변 CU가 AFFINE 기반 움직임 벡터 예측 기술을 적용하여 부호화된 경우에 해당 CU의 CPMV로부터 유도한 현재 CU의 CPMV를 의미한다. 조합 AFFINE MERGE 후보(constructed affine merge candidate)는 주변의 움직임 벡터를 결합하여 유

도한 현재 CU의 CPMV를 의미한다. 구체적으로, 〈그림 13〉에서 B2→B3→A2 순서로 움직임 벡터 존재여부를 확인하여 첫 번째로 가용한 움직임 벡터를 CPMV1로 설정하고, 유사하게 B1→B0 순서로 움직임 벡터를 확인하여 CPMV2, A1→A0 순서로 움직임 벡터를 확인하여 CPMV3, TMVP를 CPMV4로 각각 설정한다. 설정된 CPMV들을 다음과 같이 조합하여 조합 AFFINE MERGE 후보로 사용한다. 이 때, 3개의 CPMV들로 구성된 후보는 6-파라미터 AFFINE MERGE 후보이며, 2개의 CPMV들로 구성된 후보는 4-파라미터 AFFINE MERGE 후보이다.

{CPMV1, CPMV2, CPMV3},
 {CPMV1, CPMV2, CPMV4},
 {CPMV1, CPMV3, CPMV4},
 {CPMV2, CPMV3, CPMV4},
 {CPMV1, CPMV2}, {CPMV1, CPMV3}



〈그림 13〉 조합 AFFINE MERGE 후보 유도를 위한 CPMV 후보 위치

AFFINE AMVP 모드는 상속 AFFINE AMVP 후보, 조합 AFFINE AMVP 후보, 평행이동 MV, zero MV들로 후보 리스트를 구성한다. 상속

AFFINE AMVP 후보는 상속 AFFINE MERGE 후보와 동일하나 상속받는 CU의 참조픽처가 현재 CU의 참조픽처와 동일한 경우에만 해당 CU의 CPMV로부터 유도한 현재 CU의 CPMV를 후보 리스트에 추가한다. 조합 AMVP 후보도 조합 AFFINE MERGE 후보와 동일하나 모든 CPMV들의 참조픽처가 현재 CU의 참조픽처와 동일한 경우에만 후보 리스트에 추가한다. 상속 AFFINE AMVP 후보와 조합 AFFINE AMVP 후보로 구성된 AFFINE AMVP 모드의 후보들이 2개 이하인 경우에 평행이동만 가능한 평행이동 MV를 후보로 추가하며, 이 때, 평행이동 MV로는 CPMV1, CPMV2, CPMV3가 각각 사용될 수 있다. 평행이동 MV를 후보 리스트에 추가하였음에도 후보가 2개 이하인 경우에는 zero MV를 추가한다.

4. 디코더 기반 움직임 벡터 보정 기술

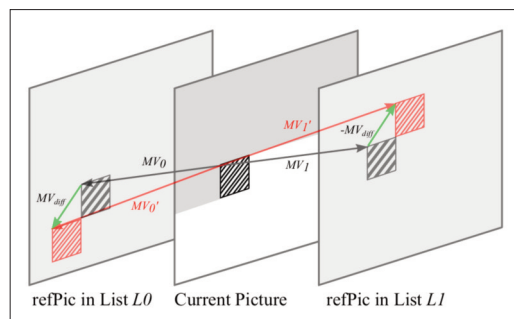
VVC에는 복호 과정에서 유도된 움직임 벡터를 기반으로 추가적인 움직임 벡터 탐색 혹은 움직임 벡터 계산 과정을 통해 보정된 움직임 벡터를 획득하여 예측신호의 정확도를 높이는 기술인 DMVR, BDOF, AFFINE 모드를 위한 PROF가 채택되었다. 본 절에서는 이러한 기술들에 대해 다루도록 한다.

1) DMVR

DMVR은 일반 MERGE 모드에서 유도한 양방향 움직임 벡터를 복호 과정에서 추가정보 전송 없이 BM(Bi-lateral Matching) 기반 움직임 벡터 탐색 과정을 통해 보정하여 일반 MERGE 모드의 예측 정확도를 높이는 기술이다.

BM 기반 움직임 벡터 보정은 L0 방향 움직임 벡

터와 L1 방향 움직임 벡터를 대칭으로 탐색하면서 L0 방향 예측신호와 L1 방향 예측신호 사이의 왜곡을 최소화하는 L0 및 L1 방향 움직임 벡터를 최종 움직임 벡터로 유도하는 방법이다. <그림 14>에서 MV_0 와 MV_1 은 일반 MERGE 모드에서 유도한 L0 및 L1 방향 움직임 벡터이며, MV_0' 는 L0 방향 움직임 벡터인 MV_0 를 MV_{diff} 만큼 보정한 움직임 벡터이고, MV_1' 는 L1 방향 움직임 벡터인 MV_1 을 L0 방향 보정 움직임 벡터와 대칭으로 보정하기 위해 $-MV_{diff}$ 만큼 보정한 움직임 벡터이다. 이 때, L0 참조픽처에서 MV_0' 가 가리키는 영역과 L1 참조픽처에서 MV_1' 가 가리키는 영역 간의 왜곡(SAD, Sum of Absolute Difference)이 최소가 되면 MV_0' 와 MV_1' 을 최종 움직임 벡터로 설정하게 된다.



<그림 14> Bi-lateral matching 기반 움직임 벡터 보정

DMVR은 움직임 벡터 보정에 의한 부호화 효율 대비 탐색 과정에서 발생하는 디코더 복잡도를 고려하여 움직임 벡터 탐색 범위를 수평 및 수직 방향에 대해 정수 샘플 단위를 기준으로 ± 2 샘플 크기로 제한하고, 주어진 탐색 범위내에서 정수 샘플 단위 탐색 및 부샘플 단위 탐색의 2단계 탐색 과정을 통해 최종 움직임 벡터를 구한다. 정수 샘플 단위 탐색은 ± 2 탐색 범위내의 25개 탐색 위치 각각에

대해 SAD 값을 계산하여 최소 SAD 값을 가지는 위치를 찾는다. 부샘플 단위 탐색은 계산 복잡도를 줄이기 위해 부샘플 위치에서의 SAD 값을 계산하여 비교하는 대신 정수 샘플 단위 탐색 과정에서 계산한 SAD 값을 이용한 2D parametric error surface equation에 의해 탐색된 정수 샘플 주변 부샘플 위치들 가운데 SAD 값이 최소가 되는 부샘플 위치를 추정한다. 이렇게 추정된 위치를 가리키는 움직임 벡터를 최종 움직임 벡터로 설정하게 된다.

DMVR은 VVC의 다른 부호화 기술들과 유사하게 하드웨어 측면의 구현 복잡도를 줄이기 위해 다음과 같은 방법들이 적용되었다. DMVR은 CU를 16x16 크기의 서브블록으로 분할하여 서브블록 단위 움직임 벡터 보정을 수행한다. 이로 인해 DMVR의 16x16 서브블록 단위 병렬처리가 가능하며, 서브블록 단위로 보다 정확하게 움직임 벡터를 보정할 수 있게 됨에 따라 부호화 효율도 개선되었다. 또한, 하드웨어 구현 시 복호 과정에서 움직임 벡터 보정으로 인해 지연이 발생하지 않도록 하기 위하여 보정된 최종 움직임 벡터는 현재 CU의 예측신호 생성 및 시간적 움직임 벡터 후보(TMVP 및 SbTMVP)에만 사용되고, 공간적 MERGE 및 AMVP 후보로는 보정되지 않은 일반 MERGE 모드에서 유도된 초기 움직임 벡터가 사용된다.

2) BDOF

일반적인 쌍방향 예측에서는 두 개의 CU단위 움직임 벡터로부터 생성한 예측신호를 가중합하여 현재 CU의 최종 예측신호를 생성한다. 이때 적용하는 움직임 벡터는 CU단위의 최적 움직임 벡터일 수 있지만, CU내 모든 샘플 위치에서 최적의 움직임 벡터는 아닐 수 있다. 이러한 CU단위 움직임 예측으로 인한 샘플단위 예측오차를 개선하기 위하여,

VVC의 BDOF는 광-흐름(optical flow) 기반 움직임 벡터 예측 기술을 적용하여 CU 단위로 획득된 양방향 움직임 벡터를 기준으로 4x4 서브블록단위로 최적 움직임 벡터를 찾고, 해당 최적 움직임 벡터 기반으로 CU내 각 휘도 샘플 위치에서의 샘플 값 변화량을 추정하여 최종 예측신호를 획득한다.

일반적으로 광-흐름 기반 움직임 예측 방법은 예측하고자 하는 영역이 광-흐름을 따라 밝기 값이 일정하고, 움직임이 느리며, 주변 샘플들과 동일한 움직임을 가진다고 가정한다. 이와 같은 조건하에 주어진 영역내에서 임의의 위치의 휘도 샘플 값이 변화하는 요인은 해당 영역 내의 움직임 때문일 것이다. 이러한 원리를 적용하여 VVC의 BDOF는 CU가 양방향 예측인 경우에 4x4 서브블록 단위로 L0 방향 예측신호와 L1 방향 예측신호 사이의 왜곡이 최소가 되는 보정된 움직임 벡터 (v_x , v_y)를 다음 식과 같이 계산에 의해 획득한다.

여기서, s_1 , s_2 , s_3 , s_4 , s_5 는 4x4 서브블록내 L0 방향 예측신호(I^0), L1 방향 예측신호(I^1), L0 방향

$$v_x = \frac{s_3}{s_1} \quad , \quad v_y = \frac{s_4 - v_x \times s_5}{s_2}$$

예측신호의 수평방향 기울기(I_x^0) 및 수직방향 기울기(I_y^0), L1 방향 예측신호의 수평방향 기울기(I_x^1) 및 수직방향 기울기(I_y^1)로부터 유도된 파라미터들로서 다음과 같이 계산된다.

$$s_1 = \sum_{[i,j] \in \Omega} \text{abs}(I_x^1 + I_x^0):$$

$$s_2 = \sum_{[i,j] \in \Omega} \text{abs}(I_y^1 + I_y^0)$$

$$s_3 = \sum_{[i,j] \in \Omega} (-\text{sign}(I_x^1 + I_x^0) \times (I^0 - I^1))$$

$$s_4 = \sum_{[i,j] \in \Omega} (-\text{sign}(I_y^1 + I_y^0) \times (I^0 - I^1))$$

$$s_5 = \sum_{[i,j] \in \Omega} \text{sign}(I_y^1 + I_y^0) \times (I_x^1 + I_x^0)$$

마지막으로 획득된 움직임 벡터를 이용하여 다음 식에 의해 (i,j) 샘플 위치에서의 예측신호 샘플 값 변화량($BDOF_{offset}(i,j)$) 및 최종 예측신호 샘플 값 ($Pred_{BDOF}(i,j)$)을 계산한다.

$$BDOF_{offset}(i,j) = \left(v_x \times (I_x^0(i,j) - I_x^1(i,j)) \right) / 2 + \left(v_y \times (I_y^0(i,j) - I_y^1(i,j)) \right) / 2$$

$$pred_{BDOF}(i,j) = (I^0(i,j) + I^1(i,j)) / 2 + BDOF_{offset}(i,j)$$

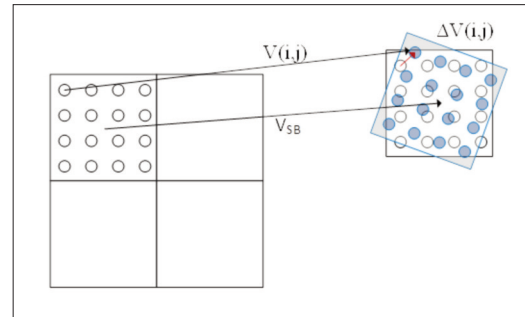
BDOF는 DMVR과 동일하게 병렬처리가 가능하도록 CU를 16x16 크기의 서브블록으로 분할하여 서브블록 단위로 독립적으로 BDOF를 수행한다. 그리고 양방향 예측 시 동일하지 않은 가중치를 적용하여 예측신호를 획득하는 BCW의 경우 광-흐름을 따라 밝기 값이 일정하다는 BDOF 가정에 어긋나므로, BCW 적용 CU에 대해 BDOF를 적용하지 않는다.

3) AFFINE 모드를 위한 PROF

VVC의 AFFINE 움직임 예측/보상 기술은 서브블록단위로 움직임 벡터를 유도하여 보상을 수행함에 의해 픽셀단위 움직임을 벡터를 유도하고 보상하는 방식에 비해 메모리 대역폭 및 계산 복잡도가 낮다는 장점이 있는 반면, 픽셀단위 움직임 예측/보상에 비해 예측 정확도가 떨어진다. VVC의 PROF는 서브블록단위 AFFINE 움직임 예측/보상의 예측 정확도를 픽셀단위 움직임 예측/보상의 예측 정확도와 유사하도록 개선하기 위한 기술로서 BDOF와 유사하게 광-흐름에 기반하여 서브블록단위로

AFFINE 움직임 보상된 픽셀 값들에 대해 픽셀단위로 보정 값을 계산하여 최종 예측신호를 획득하는 기술이다.

PROF는 L0 방향과 L1 방향에 대한 픽셀단위 보정을 독립적으로 수행하여 단방향 예측 혹은 쌍방향 예측에도 적용 가능하다는 점에서 BDOF와 차이가 있다. 각 픽셀 위치에서의 움직임 벡터 보정 값은 <그림 15>에 나타난 바와 같이 4x4 서브블록 중심 위치의 움직임 벡터(V_{SB})와 4x4 서브블록내 픽셀 위치 (i,j) 에서의 움직임 벡터($V(i,j)$) 간의 차이인 $\Delta V(i,j) = (\Delta V_x(i,j), \Delta V_y(i,j))$ 로 유도된다. 그리고, 수평방향 및 수직방향 기울기($g_x(i,j)$, $g_y(i,j)$)는 BDOF와 동일하게 3탭 필터 $[-1, 0, 1]$ 를 사용하여 계산하며, 기울기 계산 시 필요한 블록 경계 밖의 예측샘플 값은 참조픽처에서 가까운 정수 샘플 값을 사용한다.



<그림 15> 4x4 서브블록의 중심 위치의 움직임 벡터(V_{SB})와 $\Delta V(i,j)$

획득된 움직임 벡터 보정 값 ($\Delta V_x(i,j)$, $\Delta V_y(i,j)$)과 수평방향 및 수직방향 기울기($g_x(i,j)$, $g_y(i,j)$)에 기반하여 다음과 같이 픽셀 위치 (i,j) 에서의 픽셀 보정 값($\Delta I(i,j)$)을 계산한다.

$$\Delta I(i,j) = g_x(i,j) * \Delta V_x(i,j) + g_y(i,j) * \Delta V_y(i,j)$$

최종 보정된 샘플 값 $I'(i,j)$ 은 다음 식과 같이 4x4 서브 블록 단위로 AFFINE 움직임 보상을 수행하여 얻은 샘플 값 $I(i,j)$ 에 보정된 샘플 값 $\Delta I(i,j)$ 을 더해 얻게 된다.

$$I'(i,j) = I(i,j) + \Delta I(i,j)$$

II. 결 론

본 고는 VVC의 화면간 예측 부호화 기술들에 대해 살펴보았다. VVC는 SKIP 및 MERGE 모드의 부호화 효율 개선을 위해 HEVC의 SKIP 및 MERGE 모드를 확장하여 MMVD 모드, CIIP 모드, TRIANGULAR 모드, SbTMVP와 AFFINE MERGE 모드를 포함하는 서브블록 기반 MERGE 모드를 추가하였으며, HEVC의 AMVP 모드를 확

장하여 SMVD 모드 및 AFFINE AMVP 모드를 추가하였다. 특히, AFFINE MERGE와 AMVP 모드에 적용되는 AFFINE 기반 움직임 예측 기술의 채택으로 인해 확대, 축소, 회전 등과 같은 움직임을 기존 비디오 부호화 표준과 비교하여 효율적으로 예측할 수 있게 되었다. 또한, 복호 과정에서의 움직임 벡터 탐색 및 계산 과정을 통해 전송받은 움직임 벡터를 보정하는 디코더 기반 움직임 벡터 보정 기술들을 채택하여 추가로 부호화 효율을 개선하였다. 본 고에서 다룬 VVC의 화면간 예측 기술들이 적용된 VTM6.0은 RA(Random Access) 조건에서 HEVC 대비 34.86% BD-rate 효율, 인코더 복잡도 954%, 디코더 복잡도 167%를 보인다. 2020년 7월 표준화 완료 일정과 HEVC대비 2배 압축률 달성이 라는 목표를 고려할 때, 추가적인 부호화 효율 향상 방법의 채택과 동시에 복잡도 감소를 위한 연구가 필요하다.

참 고 문 헌

- [1] High Efficiency Video Coding(HEVC), Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan. 2013 (and later editions).
- [2] B. Bross, J. Chen, S. Liu, "Versatile Video Coding (Draft 6)," JVET-O2001, 15th JVET meeting: Gothenburg, SE, 3-12 July 2019
- [3] J. Chen, Y. Ye, S. H. Kim, "Algorithm description for Versatile Video Coding and Test Model 6 (VTM 6)," JVET-O2002, 15th JVET meeting: Gothenburg, SE, 3-12 July 2019
- [4] S. Jeong, M. W. Park, Y. Piao, M. Park, K. Choi, "CE4 Ultimate motion vector expression (Test 4.5.4)," JVET-L0054, 12th JVET meeting: Macao, CN, 3-12 Oct. 2018
- [5] L. Pham Van, G. Van der Auwera, A. K. Ramasubramanian, V. Seregin, M. Karczewicz, "CE10: CIIP with position-independent weights (Test CE10-1.1)," JVET-N0302, 14th meeting: Geneva, CH, 19-27 March 2019
- [6] R.-L. Liao, C. S. Lim, "CE10.3.1.b: Triangular prediction unit mode," JVET-L0124, 12th JVET meeting: Macao, CN, 3-12 Oct. 2018
- [7] L. Zhang, K. Zhang, H. Liu, J. Xu, Y. Wang, P. Zhao, D. Hong "CE2-related: Early awareness of accessing temporal blocks in sub-block merge list construction," JVET-M0273, 13th JVET meeting: Marrakech, MA, 9-18 Jan. 2019
- [8] H. Yang, H. Chen, Y. Zhao, J. Chen, "Draft text for affine motion compensation," JVET-K0565, 11th JVET meeting: Ljubljana, SI, 11-18 July 2018
- [9] J. Luo, Y. He, "CE4-related: Simplified symmetric MVD based on CE4.4.3," JVET-M0444, 13th JVET meeting: Marrakech, MA, 9-18 Jan. 2019

참고문헌

- [10] S. Sethuraman, "CE9: Results of DMVR related Tests CE9.2.1 and CE9.2.2," JVET-M0147, 13th JVET meeting: Marrakech, MA, 9-18 Jan. 2019
- [11] X. Xiu, Y. He, Y. Ye, "CE9-related: Complexity reduction and bit-width control for bi-directional optical flow (BIO)," JVET-L0256, 12th JVET meeting: Macao, CN, 3-12 Oct. 2018
- [12] T. Chujoh, T. Ikai, "Non-CE9: An improvement of BDOF," JVET-M0063, 13th JVET meeting: Marrakech, MA, 9-18 Jan. 2019
- [13] J. Luo, Y. He, "CE4-2.1: Prediction refinement with optical flow for affine mode," JVET-O0070, 15th JVET meeting: Gothenburg, SE, 3-12 July 2019
- [14] L. Zhang, K. Zhang, H. Liu, Y. Wang, P. Zhao, and D. Hong, "CE4: History-based Motion Vector Prediction (Test 4.4.7)," JVET-L0266, 12th JVET meeting: Macao, CN, 3-12 Oct. 2018
- [15] Y.-L. Hsiao, T.-D. Chuang, C.-Y. Chen, C.-W. Hsu, Y.-W. Huang, S.-M. Lei, "CE4.4.12: Pairwise average candidates," JVET-L0090, 12th JVET meeting: Macao, CN, 3-12 Oct. 2018
- [16] Y. Zhang, Y. Han, C.-C. Chen, C.-H. Hung, W.-J. Chien, M. Karczewicz, "CE4.3.3: Locally adaptive motion vector resolution and MVD coding," JVET-K0357, 11th JVET meeting: Ljubljana, SI, 11-18 July 2018
- [17] H. Liu, K. Zhang, L. Zhang, J. Xu, Y. Wang, P. Zhao, D. Hong, "CE2: Adaptive Motion Vector Resolution for Affine Inter Mode (Test 2.1.2)," JVET-M0246, 13th JVET meeting: Marrakech, MA, 9-18 Jan. 2019
- [18] Y.-C. Su, C.-Y. Chen, Y.-W. Huang, S.-M. Lei, Y. He, J. Luo, X. Xiu, Y. Ye, "CE4-related: Generalized bi-prediction improvements combined from JVET-L0197 and JVET-L0296," JVET-L0646, 12th JVET meeting: Macao, CN, 3-12 Oct. 2018

필자소개



강정원

- 1993년 : 한국항공대학교 항공전자공학과 학사
- 1995년 : 한국항공대학교 항공전자공학과 공학석사
- 2003년 : Georgia Institute of Tech. ECE 공학박사
- 2003년 ~ 현재 : 한국전자통신연구원 책임연구원 및 표준전문위원
- 주관심분야 : 영상 압축, 영상신호처리



이하현

- 2002년 : 한국항공대학교 항공전자공학과 학사
- 2007년 : 과학기술연합대학원대학교(UST) 이동통신 및 디지털방송공학전공 공학석사
- 2003년 ~ 현재 : 한국전자통신연구원 선임연구원
- 주관심분야 : 2D/입체공간 비디오 압축 및 영상신호처리