



**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(11) 공개번호 10-2013-0050149  
 (43) 공개일자 2013년05월15일

(51) 국제특허분류(Int. Cl.)  
*H04N 7/36* (2006.01)

(21) 출원번호 10-2011-0115348

(22) 출원일자 2011년11월07일

심사청구일자 없음

(71) 출원인

오수미

경기도 성남시 분당구 관교역로 98, 707동  
 1102호(백현동, 백현마을)

(72) 발명자

장민

서울특별시 서초구 잠원동 65-8 한신그린 아파트  
 A-1107

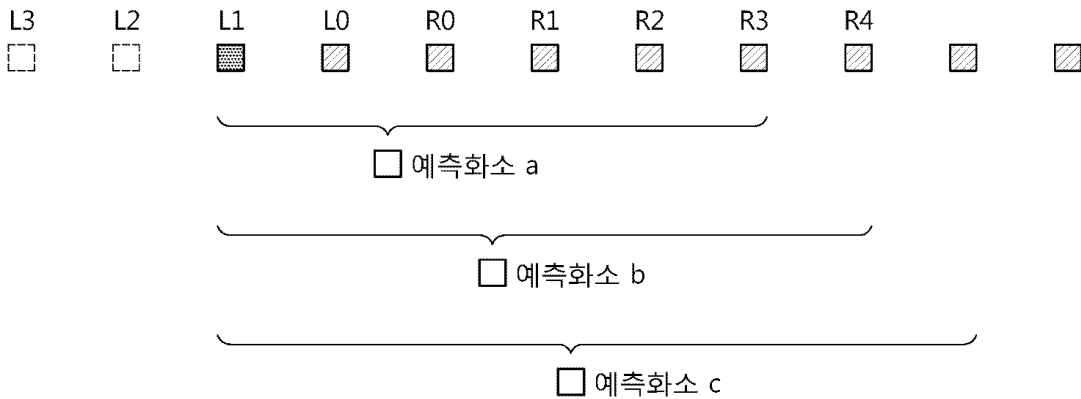
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 **인터 모드에서의 예측 블록 생성 방법**

**(57) 요약**

움직임 벡터가 정수화소 단위가 아닌 경우의 인터 예측 모드에서의 예측 블록을 생성하기 위해, 생성하고자 하는 예측 블록 내의 화소의 위치를 판단하고, 상기 화소의 위치에 따라 결정되는 필터를 이용하여 예측 화소를 생성하는 것을 특징으로 한다. 본 발명에 따르면, 예측 블록 생성을 위해 메모리로부터 읽어 들이는 데이터량을 줄임으로써, 복호기의 복잡도를 감소시킬 수 있다.

**대표도**



## 특허청구의 범위

### 청구항 1

움직임 벡터가 정수화소 단위가 아닌 경우의 인터 예측 모드에서의 예측 블록 생성 방법에 있어서,  
생성하고자 하는 예측 블록 내의 화소의 위치를 판단하는 단계;

상기 화소의 위치에 따라 결정되는 필터를 이용하여 예측 화소를 생성하는 단계를 포함하는 것을 특징으로 하는 예측 블록 생성 방법.

## 명세서

### 기술분야

[0001] 본 발명은 인터 모드로 부호화된 영상의 예측 블록을 생성하는 방법에 관한 것으로, 더욱 상세하게는 인터 모드로 부호화된 움직임 정보를 복호화하고, 이를 기초로 예측 블록을 생성하는 방법에 관한 것이다.

### 배경 기술

[0002] 영상의 품질을 유지하면서 동영상 신호를 효과적으로 압축하기 위한 많은 기술들이 제안되어 왔다. 특히, 현재 블록과 유사한 블록을 이전 화면에서 추출하여 그 차이값을 부호화하는 방식인 인터 예측 부호화 방식은 영상을 압축하는데 있어 가장 효과적인 방법 중의 하나이다.

[0003] 그러나, 상기 인터 예측 부호화 방식의 경우에는 잔차 블록을 부호화하여 보내는 대신에 각 블록들에 대응하는 움직임 정보를 별도로 전송해야 한다. 따라서, 움직임 정보를 효과적으로 부호화하여 데이터량을 줄이는 것이 또 하나의 영상 압축 방안 중의 하나이다.

[0004] 움직임 추정 부호화에서는 소정의 평가 함수를 이용하여 현재 블록과 가장 유사한 블록을 참조 픽처의 정해진 검색 범위에서 검색한다. 유사한 블록이 검색되면, 현재 블록과 참조 픽처 내의 유사한 블록 사이의 레지듀(residue)만 전송함으로써 데이터의 압축률을 높인다.

[0005] 이 때, 움직임 추정 부호화된 현재 블록을 복호화하기 위해서는 현재 블록과 참조 픽처 내의 유사한 블록 사이의 위치 차이를 나타내는 움직임 벡터에 대한 정보가 필요하다. 따라서, 부호화시에 움직임 벡터에 대한 정보를 부호화하여 비트스트림에 삽입하게 되는데, 움직임 벡터에 대한 정보를 그대로 부호화하여 삽입하면, 오버헤드(overhead)가 증가하게 되어 영상 데이터의 압축률이 낮아진다.

[0006] 따라서, 인터 예측 부호화 방법에서는 현재 블록의 움직임 벡터를 주변 블록들을 이용하여 예측하고, 예측의 결과로 생성된 예측 움직임 벡터와 원본 움직임 벡터와의 차분값만을 부호화하여 전송함으로써 움직임 벡터에 대한 정보도 압축한다.

[0007] 그러나, 움직임 벡터가 정수 단위가 아닌 경우에는 예측 블록을 생성해야 한다. 그러나, 예측 블록 생성은 복호기에서 많은 메모리와 대역폭을 요구하므로, 복잡도가 증가하게 된다.

[0008] 따라서, 메모리 점유량을 줄이고, 대역폭을 줄이면서, 예측 블록을 원본 블록과 유사하게 생성하기 위한 인터폴레이션 방법이 요구된다.

## 발명의 내용

### 해결하려는 과제

[0009] 본 발명은 인터 모드로 부호화된 영상의 예측 블록을 생성하는 방법에 관한 것으로, 더욱 상세하게는 인터 모드로 부호화된 움직임 정보를 복호화하고, 이를 기초로 예측 블록을 생성하는 방법에 관한 것이다.

**과제의 해결 수단**

[0010] 본 발명에 따른 예측 블록 생성 방법은, 예측 블록 내의 화소의 위치에 따라 서로 다른 필터를 사용하여 예측 블록을 생성하는 것을 특징으로 한다.

**발명의 효과**

[0011] 본 발명에 따르면, 예측 블록 생성을 위해 메모리로부터 읽어 들이는 데이터량을 줄임으로써, 복호기의 복잡도를 감소시킬 수 있다.

**도면의 간단한 설명**

- [0012] 도 1은 본 발명의 실시예에 따른 동영상 부호화 장치를 나타내는 블록도이다.
- 도 2는 본 발명에 따른 인터 예측 부호화 과정을 나타내는 블록도이다.
- 도 3은 본 발명의 실시예에 따른 머지 부호화 과정을 나타내는 블록도이다.
- 도 4는 본 발명의 제1 실시예에 따른 머지 후보자의 위치를 나타내는 도면이다.
- 도 5는 본 발명의 제2 실시예에 따른 머지 후보자의 위치를 나타내는 도면이다.
- 도 6은 본 발명의 실시예에 따른 AMVP 부호화 과정을 나타내는 블록도이다.
- 도 7은 본 발명의 실시예에 따른 동영상 복호화 장치를 나타내는 블록도이다.
- 도 8은 본 발명의 실시예에 따른 인터 예측 복호화 과정을 나타내는 도면이다.
- 도 9는 본 발명의 제1 실시예에 따른 머지 모드 움직임 벡터 복호화 과정을 나타내는 도면이다.
- 도 10은 본 발명의 제1 실시예에 따른 AMVP 모드 움직임 벡터 복호화 과정을 나타내는 도면이다.
- 도 11은 본 발명에 따른 인터 예측 복호화 장치를 나타내는 블록도이다.
- 도 12는 본 발명의 일 실시예에 따른 머지 모드 움직임 정보 복호화부를 나타내는 블록도이다
- 도 13은 본 발명의 일 실시예에 따른 AMVP 모드 움직임 정보 복호화부를 나타내는 블록도이다.
- 도 14는 본 발명의 일 실시예에 따른 참조 픽처에서의 시간 후보자를 저장하기 위한 후보의 위치를 나타낸다.
- 도 15는 본 발명의 일 실시예에 따른 시간 후보자의 위치를 나타내는 도면이다.
- 도 16은 본 발명에 따른 시간 후보자 선택을 위해 로딩되어야 하는 움직임 벡터 영역을 나타내고 있다.
- 도 17은 본 발명의 실시예에 따른 시간 후보자를 산출하는 방법을 나타낸 개념도이다.
- 도 18은 본 발명의 일 실시예에 따른 AMP에서의 공간 머지 후보자 위치를 나타내는 도면이다.
- 도 19는 본 발명의 다른 실시예에 따른 AMP에서의 공간 머지 후보자 위치를 나타내는 도면이다.
- 도 20은 본 발명의 또 다른 실시예에 따른 공간 머지 후보자 배제를 설명하기 위한 도면이다.
- 도 21은 본 발명의 다른 실시예에 따른 인터 예측 복호화 장치(300)를 나타내는 구성도이다.
- 도 22는 본 발명의 다른 실시예에 따른 머지 MVD 모드 움직임 정보 복호화부(480)를 나타내는 블록도이다.
- 도 23은 본 발명의 일 실시예에 따른 예측화소의 위치를 나타내는 도면이다.
- 도 24는 본 발명의 일 실시예에 따른 예측화소 생성을 위해 사용되는 정수화소의 위치를 나타내는 도면이다.
- 도 25는 본 발명의 다른 실시예에 따른 예측화소 생성을 위해 사용되는 정수화소의 위치를 나타내는 도면이다.
- 도 26은 본 발명의 또 다른 실시예에 따른 예측화소 생성을 위해 사용되는 정수화소의 위치를 나타내는 도면이다.

**발명을 실시하기 위한 구체적인 내용**

- [0013] 도 1은 본 발명의 실시예에 따른 동영상 부호화 장치를 나타내는 블록도이다.
- [0014] 도 1을 참조하면, 본 발명에 따른 동영상 부호화 장치(100)는 픽처 분할부(110), 변환부(120), 양자화부(130), 스캐닝부(131), 엔트로피 부호화부(140), 인트라 예측부(150), 인터 예측부(160), 역양자화부(135), 역변환부(125), 후처리부(170), 픽처 저장부(180), 감산부(190) 및 가산부(195)를 포함한다.
- [0015] 픽처 분할부(110)는 입력되는 비디오 신호를 분석하여 픽처를 가장 큰 코딩 유닛마다 소정 크기의 코딩 유닛으로 분할하여 예측 모드를 결정하고, 상기 코딩 유닛별로 블록의 크기를 결정한다. 그리고, 픽처 분할부(110)는 부호화할 블록을 예측 모드에 따라 인트라 예측부(150) 또는 인터 예측부(160)로 보낸다. 또한, 픽처 분할부(110)는 부호화할 블록을 감산부(190)로 보낸다.
- [0016] 변환부(120)는 입력된 원본 블록과 인트라 예측부(150) 또는 인터 예측부(160)에서 생성된 예측 블록의 잔차 신호인 잔차 블록을 변환한다. 상기 잔차 블록은 코딩 유닛으로 구성된다. 코딩 유닛으로 구성된 잔차 블록은 최적의 변환 단위로 분할되어 변환된다. 예측 모드(intra or inter)에 따라 서로 다른 변환 매트릭스가 결정될 수 있다. 또한, 인트라 예측의 잔차 신호는 인트라 예측 모드에 따라 방향성을 가지므로 인트라 예측 모드에 따라 적응적으로 변환 매트릭스가 결정될 수 있다. 변환 단위는 2개(수평, 수직)의 1차원 변환 매트릭스에 의해 변환될 수 있다. 예를 들어, 인터 예측의 경우에는 미리 결정된 1개의 변환 매트릭스가 결정된다. 반면에, 인트라 예측의 경우, 인트라 예측 모드가 수평인 경우에는 잔차 블록이 수직방향으로의 방향성을 가질 확률이 높아지므로, 수직방향으로는 DCT 기반의 정수 매트릭스를 적용하고, 수평방향으로는 DST 기반 또는 KLT 기반의 정수 매트릭스를 적용한다. 인트라 예측 모드가 수직인 경우에는 수직방향으로는 DST 기반 또는 KLT 기반의 정수 매트릭스를, 수평 방향으로는 DCT 기반의 정수 매트릭스를 적용한다. DC 모드의 경우에는 양방향 모두 DCT 기반 정수 매트릭스를 적용한다. 또한, 인트라 예측의 경우, 변환 단위의 크기에 의존하여 변환 매트릭스가 적응적으로 결정될 수도 있다.
- [0017] 양자화부(130)는 상기 변환 매트릭스에 의해 변환된 잔차 블록의 계수들을 양자화하기 위한 양자화 스텝 사이즈를 코딩 유닛별로 결정한다. 양자화 스텝 사이즈는 미리 정해진 크기 이상의 부호화 단위별로 결정된다. 상기 미리 정해진 크기는 8x8 또는 16x16일 수 있다. 그리고, 결정된 양자화 스텝 사이즈 및 예측 모드에 따라 결정되는 양자화 매트릭스를 이용하여 상기 변환 블록의 계수들을 양자화한다. 양자화부(130)는 현재 부호화 단위의 양자화 스텝 사이즈 예측자로서 현재 부호화 단위에 인접한 부호화 단위의 양자화 스텝 사이즈를 이용한다.
- [0018] 양자화부(130)는 현재 부호화 단위의 좌측 부호화 단위, 상측 부호화 단위, 좌상측 부호화 단위 순서로 검색하여 1개 또는 2개의 유효한 양자화 스텝 사이즈를 이용하여 현재 부호화 단위의 양자화 스텝 사이즈 예측자를 생성한다. 예를 들어, 상기 순서로 검색된 유효한 첫번째 양자화 스텝 사이즈를 양자화 스텝 사이즈 예측자로 결정할 수 있다. 또한, 상기 순서로 검색된 유효한 2개의 양자화 스텝 사이즈의 평균값을 양자화 스텝 사이즈 예측자로 결정할 수도 있고, 1개만이 유효한 경우에는 이를 양자화 스텝 사이즈 예측자로 결정할 수 있다. 상기 양자화 스텝 사이즈 예측자가 결정되면, 현재 부호화 단위의 양자화 스텝 사이즈와 상기 양자화 스텝 사이즈 예측자 사이의 차분값을 엔트로피 부호화부(140)로 전송한다.
- [0019] 한편, 현재 코딩 유닛의 좌측 코딩 유닛, 상측 코딩 유닛, 좌상측 코딩 유닛 모두가 존재하지 않을 가능성이 있다. 반면에 최대 코딩 유닛 내의 부호화 순서 상으로 이전에 존재하는 코딩 유닛이 존재할 수 있다. 따라서, 현재 코딩 유닛에 인접한 코딩 유닛들과 상기 최대 코딩 유닛 내에서는 부호화 순서상 바로 이전의 코딩 유닛이 후보자가 될 수 있다. 이 경우, 1) 현재 코딩 유닛의 좌측 코딩 유닛, 2) 현재 코딩 유닛의 상측 코딩 유닛, 3) 현재 코딩 유닛의 좌상측 코딩 유닛, 4) 부호화 순서상 바로 이전의 코딩 유닛 순서로 우선순위를 둘 수 있다. 상기 순서는 바뀔 수 있고, 상기 좌상측 코딩 유닛은 생략될 수도 있다.
- [0020] 상기 양자화된 변환 블록은 역양자화부(135)와 스캐닝부(131)로 제공된다.
- [0021] 스캐닝부(131)는 양자화된 변환 블록의 계수들을 스캐닝하여 1차원의 양자화 계수들로 변환한다. 양자화 후의 변환 블록의 계수 분포가 인트라 예측 모드에 의존적일 수 있으므로, 스캐닝 방식은 인트라 예측 모드에 따라 결정된다. 또한, 계수 스캐닝 방식은 변환 단위의 크기에 따라 달리 결정될 수도 있다. 상기 스캔 패턴은 방향성 인트라 예측 모드에 따라 달라질 수 있다. 양자화 계수들의 스캔순서는 역방향으로 스캔한다.
- [0022] 상기 양자화된 계수들이 복수개의 서브셋으로 분할된 경우에는 각각의 서브셋 내의 양자화 계수들에 동일한 스캔패턴을 적용한다. 상기 복수개의 서브셋은 하나의 메인 서브셋과 적어도 하나 이상의 잔여 서브셋으로 구성된

다. 메인 서브셋은 DC 계수를 포함하는 좌상측에 위치하고, 상기 잔여 서브셋은 메인 서브셋 이외의 영역을 커버한다.

- [0023] 서브셋 간의 스캔패턴은 지그재그 스캔을 적용한다. 스캔 패턴은 메인 서브셋으로부터 순방향으로 잔여 서브셋들로 스캔하는 것이 바람직하나, 그 역방향도 가능하다. 또한, 서브셋 내의 양자화된 계수들의 스캔패턴과 동일하게 서브셋 간의 스캔패턴을 설정할 수도 있다. 이 경우, 서브셋 간의 스캔패턴이 인트라 예측 모드에 따라 결정된다. 한편, 부호기는 상기 변환 유닛내의 0이 아닌 마지막 양자화 계수의 위치를 나타낼 수 있는 정보를 복호기로 전송한다. 각 서브셋 내의 0이 아닌 마지막 양자화 계수의 위치를 나타낼 수 있는 정보도 복호기로 전송한다. 상기 정보는 각각의 서브셋 내의 0이 아닌 마지막 양자화 계수의 위치를 나타내는 정보일 수 있다.
- [0024] 역양자화(135)는 상기 양자화된 양자화 계수를 역양자화한다. 역변환부(125)는 역양자화된 변환 계수를 공간 영역의 잔차 블록으로 복원한다. 가산기는 상기 역변환부에 의해 복원된 잔차블록과 인트라 예측부(150) 또는 인트라 예측부(160)로부터의 예측 블록을 합쳐서 복원 블록을 생성한다.
- [0025] 후처리부(170)는 복원된 픽처에 발생하는 블록킹 효과의 제거하기 위한 디블록킹 필터링 과정, 화소 단위로 원본 영상과의 차이값을 보완하기 위한 적응적 오프셋 적용 과정 및 코딩 유닛으로 원본 영상과의 차이값을 보완하기 위한 적응적 루프 필터링 과정을 수행한다.
- [0026] 디블록킹 필터링 과정은 미리 정해진 크기 이상의 크기를 갖는 블록 및 변환 단위의 경계에 적용하는 것이 바람직하다. 상기 크기는 8x8일 수 있다. 상기 디블록킹 필터링 과정은 필터링할 경계(boundary)를 결정하는 단계, 상기 경계에 적용할 경계 필터링 강도(boundary filtering strength)를 결정하는 단계, 디블록킹 필터의 적용 여부를 결정하는 단계, 상기 디블록킹 필터를 적용할 것으로 결정된 경우, 상기 경계에 적용할 필터를 선택하는 단계를 포함한다.
- [0027] 상기 디블록킹 필터의 적용 여부는 i) 상기 경계 필터링 강도가 0보다 큰지 여부 및 ii) 상기 필터링할 경계에 인접한 2개의 블록(P 블록, Q블록) 경계 부분에서의 화소값들이 변화 정도를 나타내는 값이 양자화 파라미터에 의해 결정되는 제1 기준값보다 작은지 여부에 의해 결정된다.
- [0028] 상기 필터는 적어도 2개 이상인 것이 바람직하다. 블록 경계에 위치한 2개의 화소들간의 차이값의 절대값이 제2 기준값보다 크거나 같은 경우에는 상대적으로 약한 필터링을 수행하는 필터를 선택한다. 상기 제2 기준값은 상기 양자화 파라미터 및 상기 경계 필터링 강도에 의해 결정된다.
- [0029] 적응적 오프셋 적용 과정은 디블록킹 필터가 적용된 영상내의 화소와 원본 화소간의 차이값(distortion)을 감소시키기 위한 것이다. 픽처 또는 슬라이스 단위로 상기 적응적 오프셋 적용 과정을 수행할지 여부를 결정할 수 있다. 픽처 또는 슬라이스는 복수개의 오프셋 영역들로 분할될 수 있고, 각 오프셋 영역별로 오프셋 타입이 결정될 수 있다. 오프셋 타입은 미리 정해진 개수(예를 들어, 4개)의 에지 오프셋 타입과 2개의 밴드 오프셋 타입을 포함할 수 있다. 오프셋 타입이 에지 오프셋 타입일 경우에는 각 화소가 속하는 에지 타입을 결정하여, 이에 대응하는 오프셋을 적용한다. 상기 에지 타입은 현재 화소와 인접하는 2개의 화소값의 분포를 기준으로 결정한다.
- [0030] 적응적 루프 필터링 과정은 디블록킹 필터링 과정 또는 적응적 오프셋 적용 과정을 거친 복원된 영상과 원본 영상을 비교한 값을 기초로 필터링을 수행할 수 있다. 적응적 루프 필터링은 상기 결정된 ALF는 4x4 크기 또는 8x8 크기의 블록에 포함된 화소 전체에 적용될 수 있다. 적응적 루프 필터의 적용 여부는 코딩 유닛별로 결정될 수 있다. 각 코딩 유닛에 따라 적용될 루프 필터의 크기 및 계수는 달라질 수 있다. 코딩 유닛별 상기 적응적 루프 필터의 적용 여부를 나타내는 정보는 각 슬라이스 헤더에 포함될 수 있다. 색차 신호의 경우에는, 픽처 단위로 적응적 루프 필터의 적용 여부를 결정할 수 있다. 루프 필터의 형태도 휘도와 달리 직사각형 형태를 가질 수 있다.
- [0031] 적응적 루프 필터링은 슬라이스별로 적용 여부를 결정할 수 있다. 따라서, 현재 슬라이스에 적응적 루프 필터링이 적용되는지 여부를 나타내는 정보는 슬라이스 헤더 또는 픽처 헤더에 포함된다. 현재 슬라이스에 적응적 루프 필터링이 적용됨을 나타내면, 슬라이스 헤더 또는 픽처 헤더는 추가적으로 적응적 루프 필터링 과정에 사용되는 휘도 성분의 수평 및/또는 수직 방향의 필터 길이를 나타내는 정보를 포함한다.
- [0032] 슬라이스 헤더 또는 픽처 헤더는 필터 세트의 수를 나타내는 정보를 포함할 수 있다. 이때 필터 세트의 수가 2 이상이면, 필터 계수들이 예측 방법을 사용하여 부호화될 수 있다. 따라서, 슬라이스 헤더 또는 픽처 헤더는 필터 계수들이 예측 방법으로 부호화되는지 여부를 나타내는 정보를 포함할 수 있으며, 예측 방법이 사용되는 경



우에는 예측된 필터 계수를 포함한다.

- [0033] 한편, 휘도 뿐만 아니라, 색차 성분들도 적응적으로 필터링될 수 있다. 따라서, 색차 성분 각각이 필터링되는지 여부를 나타내는 정보를 슬라이스 헤더 또는 픽처 헤더가 포함할 수 있다. 이 경우, 비트수를 줄이기 위해 Cr과 Cb에 대한 필터링 여부를 나타내는 정보를 조인트 코딩(즉, 다중화 코딩)할 수 있다. 이때, 색차 성분들의 경우에는 복잡도 감소를 위해 Cr과 Cb를 모두 필터링하지 않는 경우가 가장 빈번할 가능성이 높으므로, Cr과 Cb를 모두 필터링하지 않는 경우에 가장 작은 인덱스를 할당하여 엔트로피 부호화를 수행한다. 그리고, Cr 및 Cb를 모두 필터링하는 경우에 가장 큰 인덱스를 할당하여 엔트로피 부호화를 수행한다.
- [0034] 픽처 저장부(180)는 후처리된 영상 데이터를 후처리부(160)로부터 입력 받아 픽처(picture) 단위로 영상을 복원하여 저장한다. 픽처는 프레임 단위의 영상이거나 필드 단위의 영상일 수 있다. 픽처 저장부(180)는 다수의 픽처를 저장할 수 있는 버퍼(도시되지 않음)를 구비한다.
- [0035] 인터 예측부(160)는 상기 픽처 저장부(180)에 저장된 적어도 하나 이상의 참조 픽처를 이용하여 움직임 추정을 수행하고, 참조 픽처를 나타내는 참조 픽처 인덱스 및 움직임 벡터를 결정한다. 그리고, 결정된 참조 픽처 인덱스 및 움직임 벡터에 따라, 픽처 저장부(180)에 저장된 다수의 참조 픽처들 중 움직임 추정에 이용된 참조 픽처로부터, 부호화하고자 하는 블록에 대응하는 예측 블록을 추출하여 출력한다.
- [0036] 인트라 예측부(150)는 현재 블록이 포함되는 픽처 내부의 재구성된 화소값을 이용하여 인트라 예측 부호화를 수행한다. 인트라 예측부(150)는 예측 부호화할 현재 블록을 입력 받아 현재 블록의 크기에 따라 미리 설정된 개수의 인트라 예측 모드 중에 하나를 선택하여 인트라 예측을 수행한다. 인트라 예측부는 인트라 예측 블록을 생성하기 위해 참조 화소를 적응적으로 필터링한다. 참조 화소가 이용 가능하지 않은 경우에는 이용 가능한 참조 화소들을 이용하여 참조 화소들을 생성할 수 있다.
- [0037] 엔트로피 부호화부(140)는 양자화부(130)에 의해 양자화된 양자화 계수, 인트라 예측부(140)로부터 수신된 인트라 예측 정보, 인터 예측부(150)로부터 수신된 움직임 정보 등을 엔트로피 부호화한다.
- [0038] 도 2는 본 발명에 따른 인터 예측 부호화 과정을 나타내는 블록도이다.
- [0039] 인터 예측 부호화 과정은 현재 예측 유닛의 움직임 정보를 결정하는 단계, 예측 블록을 생성하는 단계, 잔차 블록을 생성하는 단계, 잔차 블록을 부호화하는 단계 및 움직임 정보를 부호화하는 단계를 포함한다. 이하에서는 예측 유닛을 블록으로 명명하여 설명한다.
- [0040] (1) 현재 예측 유닛의 움직임 정보를 결정하는 단계(S110)
- [0041] 현재 블록의 움직임 정보는 현재 블록이 참조해야 할 참조픽처 인덱스와 움직임 벡터를 포함한다.
- [0042] 따라서, 적어도 하나 이상의 참조 픽처 인덱스가 나타내는 부호화되어 복원된 픽처 내의 블록 들 중 현재 블록과의 차이 블록의 부호화량을 최소화시키는 참조 픽처 인덱스 및 움직임 벡터를 현재 예측 유닛의 움직임 정보로 결정한다.
- [0043] 현재 블록이 참조해야 할 참조픽처 인덱스는 현재 블록의 인터 예측 모드에 따라 달라질 수 있다. 예를 들어, 현재 블록이 단방향 예측 모드인 경우에는 리스트0(L0)에 속하는 참조 픽처들 중의 어느 하나를 나타내는 인덱스이다. 반면에, 현재 블록이 양방향 예측 모드인 경우에는 리스트 0(L0)에 속하는 참조 픽처를 나타내는 인덱스 및 리스트 1(L1)의 참조 픽처들 중의 어느 하나를 나타내는 인덱스를 포함할 수 있다. 또한, 현재 블록이 양방향 예측 모드인 경우에는 리스트 0과 리스트 1을 결합하여 생성된 복합 리스트(LC)의 참조 픽처들 중의 1개 또는 2개의 픽처를 나타내는 인덱스를 포함할 수 있다.
- [0044] 움직임 벡터는 각각의 참조픽처 인덱스가 나타내는 픽처내의 예측 블록의 위치를 나타낸다. 움직임 벡터는 정수 단위일수도 있으나, 1/8 또는 1/16 화소의 해상도를 가질 수 있다. 움직임 벡터가 정수단위가 아닐 경우에는 예측 블록은 정수 단위의 화소들로부터 생성된다.
- [0045] (2) 예측 블록을 생성하는 단계(S120)
- [0046] 움직임 벡터가 정수 단위일 경우에는, 참조픽처 인덱스가 나타내는 픽처 내의 움직임 벡터가 나타내는 위치의 대응 블록을 복사하여 현재 블록의 예측 블록을 생성한다.
- [0047] 그러나, 움직임 벡터가 정수 단위가 아닐 경우에는, 참조픽처 인덱스가 나타내는 픽처내의 정수 단위 화소들로

부터 예측 블록의 화소들을 생성한다. 이 경우, 휘도 화소의 경우에는 8탭의 보간 필터를 사용하여 예측 화소를 생성할 수 있다. 색차 화소의 경우에는 4탭 보간 필터를 사용하여 예측 화소를 생성할 수 있다.

[0048] (3) 잔차 블록을 생성하는 단계(S130) 및 잔차 블록을 부호화하는 단계(S140)

[0049] 현재 블록의 예측 블록들이 생성되면, 현재 블록과 예측 블록의 차이값을 이용하여 잔차 블록을 생성한다. 변환 부호화 단위가 되는 잔차 블록의 크기는 현재 블록의 크기와 다를 수 있다. 예를 들어, 현재 블록이  $2N_x2N_y$ 인 경우에는 현재 블록과 잔차 블록의 크기가 동일하다. 그러나, 현재 블록이  $2N_xN_y$  또는  $N_x2N_y$ 인 경우에는, 변환 부호화를 위한 잔차블록은  $2N_x2N_y$ 일 수 있다. 예를 들어, 현재 블록이  $2N_xN_y$ 인 경우에는 잔차 블록은  $2N_x2N_y$ 을 구성하는 2개의  $2N_xN_y$  현재블록 각각에 대한 예측 블록을 구한 후, 각 현재 블록과 예측 블록의 차이값을 나타내는 2개의 차이 블록을 결합하여 잔차 블록을 생성할 수 있다. 또한,  $2N_xN_y$ 의 예측블록 2개의 경계부분의 불연속성을 해소하기 위해 경계 부분의 픽셀들을 오버랩 스무딩하여 형성된  $2N_x2N_y$  예측 블록을 생성한 후,  $2N_x2N_y$ 의 원본 블록(2개의 현재 블록으로 구성)과의 차이값을 잔차 블록으로 생성할 수도 있다.

[0050] 잔차 블록이 생성되면, 잔차 블록은 변환 부호화 크기 단위의 블록으로 부호화된다. 즉, 변환 부호화 크기 단위로 변환 부호화, 양자화 및 엔트로피 부호화된다. 이때, 변환 부호화의 크기는 잔차 블록의 크기에 따라 쿼드트리 방식으로 결정될 수 있다. 즉, 변환 부호화는 현재 블록의 예측모드 등에 따라 적응적으로 변환 기법이 적용될 수 있다.

[0051] 변환 부호화된 블록은 양자화 매트릭스를 이용하여 양자화된다. 양자화된 블록은 CABAC 또는 CAVLC로 엔트로피 부호화된다.

[0052] (4) 움직임 정보를 부호화하는 단계(S150)

[0053] 현재 블록의 움직임 정보는 현재 블록에 인접한 블록(예측 블록)들의 움직임 정보를 이용하여 부호화된다. 현재 블록의 움직임 정보는 먼저 부호화 또는 AMVP 부호화 된다. 따라서, 먼저 현재 블록의 움직임 정보를 먼저 부호화할지 또는 AMVP 부호화할지를 결정하고, 결정된 방식에 따라 부호화한다.

[0054] 먼저, 도 3을 참조하여 먼저 부호화 방식에 대해 설명한다.

[0055] 공간 머지 후보자와 시간 머지 후보자를 유도한다(S210, S220).

[0056] 여기서는 편의상 공간 머지 후보자를 유도하고 시간 머지 후보자를 구하는 것으로 예시하였으나, 공간 머지 후보자와 시간 머지 후보자를 구하는 순서는 이에 한정되지 않는다. 예를 들어, 반대의 순서로 구해도 되고, 병렬적으로도 구해도 된다.

[0057] 1) 공간 머지 후보자

[0058] 공간 머지 후보자 구성은 아래의 구성들 중의 어느 하나일 수 있다. 공간 머지 후보자 구성정보는 복호기로 전송될 수도 있고, 미리 정해진 어느 하나일 수도 있다.

[0059] (a) 공간 머지 후보자 구성 1

[0060] 복수개의 공간 머지 후보자는 도 4에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C) 및 현재 블록의 좌하측 블록(블록 D)이 될 수 있다. 이 경우, 유효한 블록이 모두 후보자가 되거나 A, B, C, D 순으로 스캔하여 유효한 2개가 후보자가 될 수도 있다. 현재 블록의 좌측에 복수개의 블록이 존재할 경우에는 유효한 가장 상측에 존재하는 블록 또는 가장 큰 면적을 갖는 유효한 블록이 좌측 블록으로 설정될 수 있다. 마찬가지로, 현재 블록의 상측에 복수개의 블록이 존재할 경우에는 유효한 가장 좌측에 존재하는 블록 또는 가장 큰 면적을 갖는 블록이 상측 블록으로 설정될 수 있다.

[0061] (b) 공간 머지 후보자 구성 2

[0062] 복수개의 공간 머지 후보자는 도 5에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 현재 블록의 좌하측 블록(블록 D) 및 현재 블록의 좌상측 블록(블록 E) 순으로 스캔하여 유효한 2개가 후보자가 될 수 있다. 여기서 좌측 블록은 블록 D에 인접하지 않고 블

록 E에 인접한 블록일 수 있다. 마찬가지로, 상측 블록은 블록 C에 인접하지 않고 블록 E에 인접한 블록일 수 있다.

[0063] (c) 공간 머지 후보자 구성 3

[0064] 복수개의 공간 머지 후보자는 도 5에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 현재 블록의 좌하측 블록(블록 D) 및 현재 블록의 좌상측 블록(블록 E) 중 4개가 후보자가 될 수 있다. 이 경우, 블록 E는 블록 A, B, C, D 중 어느 하나가 유효하지 않은 경우에 사용될 수 있다.

[0065] (d) 공간 머지 후보자 구성 4

[0066] 복수개의 공간 머지 후보자는 도 5에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 코너 블록(C, D, E 중 하나)로 설정될 수 있다. 코너 블록은 우상측 블록(블록 C), 현재 블록의 좌하측 블록(블록 D) 및 현재 블록의 좌상측 블록(블록 E) 순으로 스캔하여 유효한 첫번째 블록이다.

[0067] 2) 시간 머지 후보자

[0068] 시간 머지 후보자의 참조 픽처 인덱스와 움직임 벡터는 별도의 과정을 통해 얻어진다. 시간 머지 후보자의 참조 픽처 인덱스는 현재 블록에 공간적으로 인접한 블록들(예측 유닛들)의 참조 픽처 인덱스들을 이용하여 구할 수도 있다.

[0069] 구체적으로, 현재 블록의 시간 머지 후보자의 참조 인덱스들을 구하기 위해 현재 블록의 좌측 블록(A), 상측 블록(B), 우상측 블록(C), 좌하측 블록(D) 및 좌상측 블록(E)의 참조 픽처 인덱스들 중 일부 또는 전부가 사용될 수 있다. 예를 들어, 현재 블록의 좌측 블록(A), 상측 블록(B) 및 코너 블록(C, D, E 중 어느 하나)들의 참조 픽처 인덱스가 사용될 수 있다. 이 경우, 코너 블록의 참조 픽처 인덱스는 우상측 블록(C), 좌하측 블록(D), 좌상측 블록(E) 순으로 스캔하여 유효한 첫번째 블록의 참조 픽처 인덱스로 결정될 수 있다. 또한, 현재 블록의 좌측 블록(A), 상측 블록(B), 우상측 블록(C), 좌하측 블록(D) 및 좌상측 블록(E)의 순으로 스캔하여 유효한 홀수개(예를 들어, 3개)의 블록의 참조 픽처 인덱스가 사용될 수도 있다.

[0070] 먼저, 현재 블록의 시간 머지 후보자의 참조 인덱스들을 구하기 위해, 현재 블록의 좌측 블록(A), 상측 블록(B) 및 코너 블록(C, D, E 중 어느 하나)들의 참조 픽처 인덱스가 사용되는 경우에 대해 설명한다.

[0071] 현재 블록의 좌측 블록의 참조 픽처 인덱스(이하, 좌측 참조 픽처 인덱스), 상측 블록의 참조 픽처 인덱스(이하, 상측 참조 픽처 인덱스) 및 코너 블록의 참조 픽처 인덱스(이하, 코너 참조 픽처 인덱스)를 구한다. 여기에서는 코너의 예측 유닛들(C, D, E) 중 하나만을 후보자로 설정하였지만, 이에 한정되는 것은 아니며, 예측 유닛 C, D가 후보자로 설정되는 경우(후보자 4개), 예측 유닛 C, D, E가 모두 후보자로 설정되는 경우(후보자 5개)도 가능하다.

[0072] 상기 참조 픽처 인덱스들 중 유효한 참조 픽처 인덱스들 중 가장 많은 빈도를 갖는 참조 픽처 인덱스를 시간 스킵 후보자의 참조 픽처 인덱스로 설정된다. 유효한 후보자들 중 가장 많은 빈도를 갖는 참조 픽처 인덱스가 복수개 존재할 경우에는 상기 복수개의 참조 픽처 인덱스 중 최소값을 갖는 참조 픽처 인덱스를 시간 스킵 후보자를 위한 참조 픽처 인덱스로 설정한다.

[0073] 다음으로, 현재 블록의 시간 머지 후보자의 참조 인덱스들을 구하기 위해, 현재 블록의 좌측 블록(A), 상측 블록(B), 우상측 블록(C), 좌하측 블록(D) 및 좌상측 블록(E)의 순으로 스캔하여 유효한 3개의 블록의 참조 픽처 인덱스가 사용되는 경우에 대해 예시적으로 설명한다.

[0074] 상기 블록들의 스캔 순으로 스캔하여 유효한 3개의 참조 픽처 인덱스를 구한다. 여기에서는 3개의 유효한 참조 픽처 인덱스만을 이용하는 것으로 설정하였지만, 유효한 모든 개수를 이용할 수도 있다. 또한, 3개가 존재하지 않을 경우에는 유효한 개수들만을 이용하거나 인덱스 0의 값을 추가할 수도 있다.

[0075] 상기 참조 픽처 인덱스들 중 유효한 참조 픽처 인덱스들 중 가장 많은 빈도를 갖는 참조 픽처 인덱스를 시간 머지 후보자의 참조 픽처 인덱스로 설정된다. 유효한 후보자들 중 가장 많은 빈도를 갖는 참조 픽처 인덱스가 복수 개 존재할 경우에는 상기 복수개의 참조 픽처 인덱스 중 최소값을 갖는 참조 픽처 인덱스 또는 좌측 블록 또는 상측 블록의 참조 픽처 인덱스를 시간 머지 후보자의 참조 픽처 인덱스로 설정한다.



- [0076] 다음으로, 상기 시간 머지 후보자의 움직임 벡터를 구하는 과정을 설명한다.
- [0077] 먼저, 상기 시간 머지 후보자 블록이 속하는 픽처(이하, 시간 머지 후보자 픽처)를 결정한다. 시간 머지 후보자 픽처는 참조 픽처 인덱스가 0인 픽처로 설정될 수 있다. 이 경우, 슬라이스 타입이 P인 경우에는 리스트 0(list0)의 첫번째 픽처(즉 인덱스가 0인 픽처)가 시간 머지 후보자 픽처로 설정된다. 슬라이스 타입이 B인 경우에는 슬라이스 헤더내의 시간 머지 후보자 리스트를 나타내는 플래그가 나타내는 참조픽처 리스트의 첫번째 픽처가 시간 머지 후보자 픽처로 설정된다. 예를 들어, 상기 플래그가 1을 나타내면 list0로부터, 0을 나타내면 list1으로부터 시간 머지 후보자 픽처를 설정할 수 있다.
- [0078] 다음으로, 상기 시간 머지 후보자 픽처 내의 시간 머지 후보자 블록을 구한다. 상기 시간 머지 후보자 블록으로서, 상기 시간 머지 후보자 픽처 내의 현재 블록에 대응하는 복수개의 대응 블록 중 어느 하나가 선택될 수 있다. 이 경우, 복수개의 대응 블록들에 우선순위를 부여하고, 상기 우선순위에 기초하여 유효한 첫번째 대응 블록이 시간 머지 후보자 블록으로 선택될 수 있다.
- [0079] 예를 들어, 상기 시간 머지 후보자 픽처 내의 현재 블록에 대응하는 블록에 인접하는 좌하측 코너 블록(BR\_C) 또는 상기 시간 머지 후보자 픽처 내의 현재 블록에 대응하는 블록 내의 좌하측 블록(BR)을 제1 후보자 블록으로 설정하고, 상기 시간 머지 후보자 픽처 내의 현재 블록에 대응하는 블록의 중앙 위치의 좌상측 픽셀을 포함하는 블록(C1) 또는 우하측 픽셀을 포함하는 블록(C2)을 제 2 후보자 블록으로 설정될 수 있다.
- [0080] 상기 제1 후보자 블록이 유효하면 상기 제1 후보자 블록을 시간 머지 후보자 블록으로 설정하고, 상기 제1 후보자 블록이 유효하지 않고 상기 제2 후보자 블록이 유효하면, 상기 제2 후보자 블록을 시간 머지 후보자 블록으로 설정한다.
- [0081] 유효한 시간 머지 후보자 블록이 복수개 존재할 경우, 가장 큰 면적을 갖는 대응블록 또는 이들의 중앙값(median)을 시간 머지 후보자 움직임 벡터로 설정할 수도 있다.
- [0082] 또한, 시간 머지 후보자 블록은 미리 정해진 위치의 대응 블록으로만 정해질 수 있다. 이 경우, 대응 블록의 움직임 벡터가 존재하지 않으면 시간 머지 후보자는 존재하지 않는 것으로 판단할 수도 있다.
- [0083] 시간 머지 후보자 블록의 위치는 현재 예측 블록이 LCU 내에서 어느 위치에 있는지에 따라 달리 결정될 수도 있다. 즉, 현재 블록의 우측 경계 및/또는 아래쪽 경계가 LCU의 경계에 접하는지 여부에 따라 달리 결정될 수 있다.
- [0084] 상기 머지 후보자 예측 블록이 결정되면, 상기 시간 머지 후보자 예측 블록의 움직임 벡터를 시간 머지 후보자 움직임 벡터로 설정한다.
- [0085] 한편, 현재 블록의 크기에 따라 상기 시간 머지 후보자를 적응적으로 off 시킬 수도 있다. 예를 들어, 4x4 블록이 경우에는 복잡도 감소를 위해 상기 시간 머지 후보자를 off 시킬 수 있다.
- [0086] 다음으로 머지 후보자 리스트를 구축한다(S230)
- [0087] 유효한 머지 후보자를 이용하여 정해진 순서에 따라 머지 후보자 리스트를 구축한다. 이 경우, 복수개의 머지 후보자가 동일 움직임 정보(동일 움직임 벡터 및 동일 참조픽처 인덱스)를 가지면, 후순위의 머지 후보자를 리스트에서 삭제한다. 여기서, 움직임 벡터가 동일하지 않더라도 움직임 벡터만 소정 범위 차이가 나는 경우에도 후순위 머지 후보자를 리스트에서 삭제할 수도 있다. 상기 소정 범위는 미리 정해진 값이거나, 부호기에서 결정된 값일 수 있다. 부호기에서 결정된 값일 경우 복호기로 전송된다.
- [0088] 예를 들어, 공간 머지 후보자 구성 1의 경우, 정해진 순서는 A, B, Col, C, D 순이다. 여기서, Col은 시간 머지 후보자를 의미한다.
- [0089] 공간 머지 후보자 구성 2의 경우, A, B, C, D, E 순으로 스캔하여 유효한 2개, Col 순으로 구성될 수 있다.
- [0090] 공간 머지 후보자 구성 3의 경우, 정해진 순서는 A, B, Col, C, D의 순이되, A, B, C, D 중 적어도 하나가 유효하지 않을 경우에 E가 추가될 수 있다. 이 경우, E는 후순위로 추가될 수 있다. 또한, (A와 D 중 하나), (C, B, E 중 하나), Col 순으로 리스트가 구성될 수도 있다.
- [0091] 공간 머지 후보자 구성 4의 경우, 정해진 순서는 A, B, Col, Corner 순이거나, A, B, Corner, Col 순일 수 있다.
- [0092] 다음으로, 머지 후보자 생성이 필요한지 여부를 판단한다(S240). 상기한 머지 후보자 구성에서 머지 후보자 수

가 고정된 값으로 설정된 경우에는, 유효한 머지 후보자 수가 고정된 값보다 작으면 머지 후보자를 생성한다 (S250). 그리고, 생성된 머지 후보자를 머지 후보자 리스트에 추가한다. 이 경우, 생성된 머지 후보자는 리스트 내의 가장 후순위의 머지 후보자 다음 위치에 추가한다. 복수개의 머지 후보자가 추가될 경우에는 미리 정해진 순서에 따라 추가한다.

- [0093] 상기 머지 후보자 생성은 복수개의 방법들을 정해진 순서에 따라 적용할 수 있다. 현재 예측 블록이 속하는 슬라이스의 슬라이스 타입에 따라 서로 다른 방법들이 적용될 수 있다. 먼저, 머지 후보자를 생성하기 위한 방법들을 설명한다.
- [0094] 제1 방법은 유효한 머지 후보자 수가 2개 이상인 경우에 적용할 수 있다. 유효한 제1 머지 후보자의 움직임 정보가 리스트 A의 움직임 정보라고 할 때, 유효한 제2 머지 후보자의 움직임 정보가 상기 리스트와 다른 리스트의 움직임 정보이거나 이를 포함할 경우에 적용된다. 상기 제1 머지 후보자의 리스트 A 움직임 정보와 제2 머지 후보자의 리스트 B (B는 A와 다른 값임)의 움직임 정보를 결합하여 양방향 움직임 정보를 구성하고, 이를 후보자로 추가한다. 상기 추가될 수 있는 후보자가 복수인 경우에는 미리 정해진 순서에 따라 추가한다. 상기 미리 정해진 순서는 유효한 머지 후보자의 인덱스에 따라 결정된다.
- [0095] 제2 방법은 유효한 머지 후보자가 1개 이상 존재할 때 적용 가능하다. 상기 제2 방법에 의해 생성되는 후보자는 다음의 움직임 정보를 갖는다.
- [0096] - 유효한 머지 후보자의 참조 픽처 리스트와 다른 참조 픽처 리스트
- [0097] - 유효한 머지 후보자의 참조 픽처와 동일한 참조 픽처가 아니고, 현재 픽처와 머지 후보자 참조 픽처와의 시간적 거리와 동일한 거리만큼 떨어져 있는 참조 픽처
- [0098] - 유효한 머지 후보자의 움직임 벡터와 반대 방향의 동일한 크기의 움직임 벡터
- [0099] 제 3 방법은 움직임 벡터값이 0인 머지 후보자를 추가하는 것이다. 움직임 벡터값이 0인 머지 후보자는 3가지가 존재할 수 있다. 1) 움직임 벡터가 0이고, 참조픽처 리스트가 0이고, 참조픽처 인덱스가 0인 머지 후보자(단방향 L0 머지 후보자), 2) 움직임 벡터가 0이고, 참조픽처 리스트가 1이고, 참조픽처 인덱스가 0인 머지 후보자(단방향 L1 머지 후보자), 3) 상기 1), 2)의 머지 후보자를 결합한 양방향 머지 후보자가 존재할 수 있다.
- [0100] 슬라이스 타입이 P인 경우에는 상기한 단방향 머지 후보자들 중 하나가 추가될 수 있다. 또는 미리 정해진 순서로 후보자 수가 채워질때까지 추가될 수도 있다.
- [0101] 슬라이스 타입이 B인 경우에는 상기한 머지 후보자들 중 하나가 추가될 수 있다. 이 경우 양방향 머지 후보자가 추가될 수 있다. 또는 미리 정해진 순서(양방향 머지 후보자 -> 단방향 머지 후보자 순)로 후보자 수가 채워질때까지 추가될 수도 있다.
- [0102] 슬라이스 타입이 B이지만, 참조 픽처들이 현재 픽처보다 시간적으로 선행하는 경우(이하, 'low delay의 경우'라 함)에는 머지 후보자 추가시 상기 제2 방법을 적용하지 않는다.
- [0103] 결과적으로, 머지 후보자를 추가하는 방법은 현재 블록이 속하는 슬라이스 타입에 따라 결정된다.
- [0104] 슬라이스 타입이 P인 경우에는 제3 방법만이 적용된다. 제3 방법 중에서도 단방향 L0 머지 후보자가 추가된다.
- [0105] 슬라이스 타입이 B인 경우에는 제1 방법, 제2 방법, 제3 방법 순으로 머지 후보자 생성 순서를 적용할 수 있다. 제1 방법에 의해 머지 후보자 수가 채워지면, 제2, 3 방법을 적용하지 않는다. 그러나, 제1 방법에 의해 머지 후보자 수가 채워지지 않으면, 제2 방법을 적용하고, 제2 방법에 의해 추가되지 않을 경우에는 제3 방법을 적용할 수 있다. 제3 방법 적용시에는 양방향 머지 후보자를 우선적으로 채울 수 있다. 단방향 머지 후보자는 추가하지 않을 수도 있다.
- [0106] 한편, 슬라이스 타입이 P인 경우, 2개 이상의 머지 후보자가 존재하고, 동일 리스트에 속할 경우, 상기 2개 이상의 머지 후보자를 이용하여 새로운 머지 후보자를 생성할 수 있다. 예를 들어, 참조 픽처 인덱스가 동일할 경

우, 상기 참조 픽처 인덱스를 갖고, 2개의 머지 후보자의 움직임 벡터의 평균값 또는 3개의 머지 후보자의 미디언 값을 추가할 수도 있다. 슬라이스 타입이 B인 경우에도 적용 가능하다.

- [0107] 또는 상기 0 움직임 벡터를 추가한 후에도 후보자 수를 채우지 못할 경우, 상기 0 움직임 벡터 외에 영벡터 근처의 움직임 벡터를 추가할 수도 있다. 또는, 유효한 머지 후보자의 움직임 벡터를 조금 변형시켜 머지 후보자를 생성할 수도 있다.
- [0108] 다음으로, 구축된 머지 후보자 리스트에서 현재 블록의 움직임 정보와 동일한 움직임 정보를 갖는 머지 후보자를 머지 예측자로 결정한다(S260)
- [0109] 다음으로, 상기 머지 예측자의 인덱스(즉, 머지 인덱스)를 부호화한다(S270).
- [0110] 머지 인덱스는 고정길이 부호화 또는 CAVLC가 사용될 수 있다. CAVLC가 사용될 경우, 모드워드 매핑을 위한 머지 인덱스는 블록의 형태(PU shape) 및 블록의 인덱스(PU index)에 따라 조정될 수 있다.
- [0111] 머지 후보자 수는 고정될 수 있다. 이 경우에는 하나의 테이블을 이용하여 상기 머지 인덱스에 대응하는 코드워드를 선택한다.
- [0112] 도 6을 참조하여 AMVP 부호화 방식에 대해 설명한다.
- [0113] 공간 머지 후보자와 시간 머지 후보자를 유도한다(S310, S320).
- [0114] 여기서는 편의상 공간 머지 후보자를 유도하고 시간 머지 후보자를 구하는 것으로 예시하였으나, 공간 머지 후보자와 시간 머지 후보자를 구하는 순서는 이에 한정되지 않는다. 예를 들어, 반대의 순서로 구해도 되고, 병렬적으로도 구해도 된다.
- [0115] 1) 공간 AMVP 후보자
- [0116] 공간 AMVP 후보자 위치의 움직임 벡터는 현재 예측 블록의 움직임 벡터와 비교하여 4종류가 있다. 즉, 1) 동일 참조 픽처를 갖고 동일 리스트를 갖는 움직임 정보(mv1), 2) 동일 참조 픽처를 갖고 다른 리스트를 갖는 움직임 정보(mv2), 3) 다른 참조 픽처를 갖고 동일 리스트를 갖는 움직임 정보(mv3), 및 4) 다른 참조 픽처를 갖고 다른 리스트를 갖는 움직임 정보(mv4)가 존재할 수 있다. 이 경우, mv1, mv2는 동일 참조 픽처를 나타내므로, 움직임 벡터를 스케일링 할 필요가 없으나, mv3은 움직임 벡터를 스케일링 해야 하고, mv4는 스케일링을 해야 할 경우들이 발생한다. 공간 AMVP 후보자 구성 1
- [0117] 복수개의 공간 AMVP 후보자는 도 5에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A)과 좌하측 블록(블록 D)들 중 하나를 좌측 후보자로 택하고, 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 및 현재 블록의 좌상측 블록(블록 E)들 중 하나를 상측 후보자로 택할 수 있다. 이때 미리 정해진 순서로 스캔하여 유효한 첫번째 블록의 움직임 벡터가 후보자로 선택된다.
- [0118] <mv1, mv2만을 이용하는 경우>
- [0119] 스케일링 여부를 판단할 필요가 없는 mv1, mv2만을 이용하여 좌측 후보자 및 상측 후보자를 선택할 수 있다.
- [0120] 먼저, 좌하측 블록의 mv1, mv2순으로, 움직임 벡터가 존재하는지를 판단한다. 존재하면, 상기 움직임 벡터를 좌측 후보자로 설정한다. 그러나, 존재하지 않으면, 좌측 블록(A)의 mv1, mv2 순으로, 움직임 벡터가 존재하는지를 판단한다. 존재하면, 상기 움직임 벡터를 좌측 후보자로 설정한다.
- [0121] 다음으로, 우상측 블록(C), 상측 블록(B), 좌상측 블록(E) 순으로 mv1, mv2가 존재하는지를 판단하고, 존재하는 첫번째 움직임 벡터를 상측 후보자로 설정한다.
- [0122] <mv1, mv2, mv3, mv4를 이용하는 경우>
- [0123] 먼저, 좌측 후보자를 설정한다.
- [0124] 구체적으로, 좌하측 블록(D)의 움직임 벡터가 mv1, mv2순으로, 존재하는지를 판단한다. 존재하면, 상기 움직임 벡터를 좌측 후보자로 설정한다. 그러나, 존재하지 않으면, 좌측 블록(A)의 움직임 벡터가 mv1, mv2순으로 존재

하는지를 판단한다. 존재하면, 상기 움직임 벡터를 좌측 후보자로 설정한다. 그러나, 좌측 블록(A)의 mv1, mv2에 대응하는 움직임 벡터가 존재하지 않으면, 좌하측 블록(D)의 움직임 벡터가 존재하는지를 mv3, mv4 순으로 판단한다. 존재하면, 상기 존재하는 움직임 벡터를 현재 픽처와 참조 픽처간의 거리 및 상기 좌하측 블록(D)과 상기 좌상측 블록(D)의 참조 픽처간의 거리를 이용하여 스케일링하고, 좌측 후보자로 설정한다. 존재하지 않으면, 좌측 블록(A)의 움직임 벡터가 존재하는지를 mv3, mv4 순으로 판단하여 존재하면, 스케일링하여 좌측 후보자로 설정한다.

[0125] 다음으로, 상측 후보자는 좌측 후보자의 종류에 의존하여 설정될 수 있다. 구체적으로, 우상측 블록(C)의 움직임 벡터가 존재하는지를 mv1, mv2순서로 판단하고, 존재하면 상기 움직임 벡터를 상측 후보자로 설정한다. 존재하지 않으면, 상측 블록(B)의 움직임 벡터가 존재하는지를 mv1, mv2순서로 판단하고, 존재하면 상기 움직임 벡터를 상측 후보자로 설정한다. 존재하지 않으면, 좌상측 블록(E)의 움직임 벡터가 존재하는지를 mv1, mv2순서로 판단하고, 존재하면 상기 움직임 벡터를 상측 후보자로 설정한다.

[0126] 그러나, 존재하지 않으면, 좌측 후보자가 mv1 또는 mv2에 대응하거나 존재하지 않으면, 상기한 순서로 다시 mv3, mv4를 스캔하여 첫번째 만나는 움직임 벡터를 스케일링 한 후 상측 후보자로 설정한다. 그러나, mv3 또는 mv4에 대응하는 좌측 후보자가 존재하면, 상측 후보자는 이용 가능하지 않는 것으로 판단한다.

[0127] 미리 정해진 순서는 좌측 블록의 경우에는 블록 A, 블록 D순으로, 상측 블록의 경우에는 블록 B, 블록 C, 블록 E 순일 수도 있다.

[0128] (a) 공간 AMVP 후보자 구성 2

[0129] 복수개의 공간 AMVP 후보자는 도 4에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 및 현재 블록의 좌하측 블록(블록 D) 순으로 스캔하여 유효한 2개의 블록을 후보자로 선택할 수 있다. 이 경우, 유효한 블록이 모두 후보자가 되거나 A, B, C, D 순으로 스캔하여 유효한 2개가 후보자가 될 수도 있다. 현재 블록의 좌측에 복수개의 블록이 존재할 경우에는 유효한 가장 상측에 존재하는 블록 또는 가장 큰 면적을 갖는 유효한 블록이 좌측 블록으로 설정될 수 있다. 마찬가지로, 현재 블록의 상측에 복수개의 블록이 존재할 경우에는 유효한 가장 좌측에 존재하는 블록 또는 가장 큰 면적을 갖는 블록이 상측 블록으로 설정될 수 있다.

[0130] 상기한 (a)에서의 방식과 유사하게 2개의 후보자를 설정할 수 있다.

[0131] 먼저, 미리 정해진 순서로, 각 블록에 대하여 mv1, mv2에 대한 움직임 벡터가 존재하는지를 판단하고, 존재하면 그 움직임 벡터를 후보자로 설정한다. 2개의 후보자가 설정될때까지 검색한다.

[0132] 그러나, 상기 블록들에 대해 모두 검사를 하였으나, 2개의 후보자가 설정되지 않으면, mv3, mv4에 대하여 미리 정해진 순서로 존재하는 움직임 벡터를 구한 후 스케일링 하여 후보자로 설정할 수 있다. 이 경우, mv1, mv2에 대한 후보자가 하나도 존재하지 않을 경우에는, mv3, mv4에 대하여 하나의 후보자만을 설정한다.

[0133] 복잡도를 더 감소시키기 위해 mv1, mv2에 대해서만 후보자가 존재하는지를 판단할 수도 있다.

[0134] (b) 공간 AMVP 후보자 구성 3

[0135] 복수개의 공간 AMVP 후보자는 도 5에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 현재 블록의 좌하측 블록(블록 D), 현재 블록의 좌상측 블록(블록 E)들 순으로 스캔하여 유효한 2개의 블록을 후보자로 선택할 수 있다. 여기서 좌측 블록은 블록 D에 인접하지 않고 블록 E에 인접한 블록일 수 있다. 마찬가지로, 상측 블록은 블록 C에 인접하지 않고 블록 E에 인접한 블록일 수 있다.

[0136] (a)의 경우와 동일하게 후보자를 선택할 수 있다. 따라서, 구체적인 내용은 생략한다.

[0137] (c) 공간 AMVP 후보자 구성 4

[0138] 복수개의 공간 AMVP 후보자는 도 5에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 현재 블록의 좌하측 블록(블록 D) 및 현재 블록의 좌상측 블록

(블록 E) 중 4개가 후보자가 될 수 있다. 이 경우, 블록 E는 블록 A, B, C, D 중 어느 하나가 유효하지 않은 경우에 사용될 수 있다.

[0139] 미리 정해진 블록 순서로 mv1, mv2에 대하여 후보자를 검색하고, 후보자가 채워지지 않을 경우, 미리 정해진 블록 순서로 mv3, mv4에 대하여 후보자를 검색한다.

[0140] (d) 공간 AMVP 후보자 구성 5

[0141] 복수개의 공간 AMVP후보자는 도 5에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 코너 블록(C, D, E 중 하나)로 설정될 수 있다. 코너 블록은 우상측 블록(블록 C), 현재 블록의 좌하측 블록(블록 D) 및 현재 블록의 좌상측 블록(블록 E) 순으로 스캔하여 유효한 첫번째 블록이다.

[0142] 미리 정해진 블록 순서로 mv1, mv2에 대하여 후보자를 검색하고, 후보자가 채워지지 않을 경우, 미리 정해진 블록 순서로 mv3, mv4에 대하여 후보자를 검색한다.

[0143] **2) 시간 AMVP 후보자**

[0144] 시간 AMVP 후보자는 움직임 정보만을 필요로 하므로, 머지 AMVP 후보자와 달리 참조픽처 인덱스를 구할 필요가 없다. 즉, 시간 AMVP 후보자의 움직임 벡터를 구하는 과정을 설명한다.

[0145] 먼저, 상기 시간 AVMP 후보자 블록이 속하는 픽처(이하, 시간 AVMP 후보자 픽처)를 결정한다. 시간 AVMP 후보자 픽처는 참조 픽처 인덱스가 0인 픽처로 설정될 수 있다. 이 경우, 슬라이스 타입이 P인 경우에는 리스트 0(list0)의 첫번째 픽처(즉 인덱스가 0인 픽처)가 시간 AVMP 후보자 픽처로 설정된다. 슬라이스 타입이 B인 경우에는 슬라이스 헤더내의 시간 AVMP 후보자 리스트를 나타내는 플래그(a flag indicating whether the temporal skip candidate picture is included in list0 or list1)가 나타내는 리스트의 첫번째 픽처가 시간 머지 후보자 픽처로 설정된다.

[0146] 다음으로, 상기 시간 AVMP 후보자 픽처 내의 시간 AVMP 후보자 블록을 구한다. 이 과정은 상기한 상기 시간 머지 후보자 블록을 구하는 과정과 동일하므로 생략한다.

[0147] 한편, 현재 블록의 크기에 따라 상기 시간 AVMP 후보자를 적응적으로 off 시킬 수도 있다. 예를 들어, 4x4 블록 이 경우에는 복잡도 감소를 위해 상기 시간 AVMP 후보자를 off 시킬 수 있다.

[0148] 다음으로, AMVP 후보자 리스트를 구축한다(S330).

[0149] 유효한 AVMP 후보자를 이용하여 정해진 순서에 따라 AVMP 후보자 리스트를 구축한다. 이 경우, 복수개의 AVMP 후보자가 동일 움직임 벡터를 가지면(참조픽처가 동일할 필요는 없음), 후순위의 AVMP 후보자를 리스트에서 삭제한다. 여기서, 움직임 벡터가 동일하지 않더라도 움직임 벡터만 소정 범위 차이가 나는 경우에도 후순위 머지 후보자를 리스트에서 삭제할 수도 있다. 상기 소정 범위는 미리 정해진 값이거나, 부호기에서 결정된 값일 수 있다. 부호기에서 결정된 값일 경우 복호기로 전송된다.

[0150] 공간 AMVP 후보자 구성 1의 경우, 정해진 순서는 A와 D 중 하나(A->D 순 또는 D->A순), B, C, E(B->C->E순 또는 C->B->E순) 중 하나, Col 순이거나, Col, A와 D 중 하나, B, C, E 중 하나의 순일 수 있다. 여기서 Col은 시간 AMVP 후보자를 나타낸다.

[0151] 공간 AMVP 후보자 구성 2의 경우, 정해진 순서는 A, B, Col, C, D 순이거나, C, D, Col, A, B 순일 수 있다.

[0152] 공간 AMVP 후보자 구성 3의 경우, 정해진 순서는 (A, B, C, D, E 순으로 유효한 2개), Col 순이거나, Col, (A, B, C, D, E 순으로 유효한 2개)의 순일 수 있다.

[0153] 공간 AMVP 후보자 구성 4의 경우, 정해진 순서는 A, B, Col, C, D의 순이되, A, B, C, D 중 적어도 하나가 유효하지 않는 경우, E가 후순위로 추가될 수 있다. 또는 Col, (A, B, C, D, E 순으로 유효한 4개)의 순일 수 있다.

[0154] 공간 AMVP 후보자 구성 5의 경우, 정해진 순서는 A, B, col, corner 순일 수 있다.

[0155] 다음으로, AVMP 후보자 생성이 필요한지 여부를 판단한다(S340). 상기한 AVMP 후보자 구성에서 AVMP 후보자 수가 고정된 값으로 설정된 경우에는, 유효한 AVMP 후보자 수가 고정된 값보다 작으면 AVMP 후보자를 생성한다



(S350). 그리고, 상기 생성된 AMVP 후보자를 리스트 내의 가장 후순위의 AMVP 후보자 다음 위치에 추가한다. 복수개의 AMVP 후보자가 추가될 경우에는 미리 정해진 순서에 따라 추가된다. 상기 추가되는 후보자는 움직임 벡터가 0 인 후보자가 추가될 수 있다.

[0156] 또는 상기 0 움직임 벡터를 추가한 후에도 후보자 수를 채우지 못할 경우, 상기 0 움직임 벡터 외에 영벡터 근처의 움직임 벡터를 추가할 수도 있다. 또는, 유효한 머지 후보자의 움직임 벡터를 조금 변형시켜 머지 후보자를 생성할 수도 있다.

[0157] 그러나, AMVP후보자 수가 가변이고, 유효한 AMVP후보자 만을 이용하는 경우에는 상기 단계들(S340, S350)은 생략 가능하다.

[0158] 다음으로, 구축된 AMVP 후보자 리스트에서 현재 블록의 움직임 벡터 예측자를 결정한다(S360). 그리고, 상기 예측자를 나타내는 AMVP 인덱스를 생성한다.

[0159] 다음으로, 현재 블록의 움직임 벡터와 상기 움직임 벡터 예측자 사이의 차분 움직임 벡터를 생성한다(S370)

[0160] 다음으로, 현재 블록의 참조 픽처 인덱스, 차분 움직임 벡터 및 AMVP 인덱스를 부호화한다(S380). 상기 AMVP 후보자가 한 개일 경우에는 AMVP 인덱스를 생략한다. 그러나, AMVP 후보자가 둘 이상일 경우에는 AMVP 인덱스를 부호화한다.

[0161] AMVP 인덱스는 고정길이 부호화 또는 CAVLC가 사용될 수 있다. CAVLC가 사용될 경우, 코드워드 매핑을 위한 AMVP 인덱스는 블록의 형태(PU shape) 및 블록의 인덱스(PU index)에 따라 조정될 수 있다.

[0162] AMVP 후보자 수는 고정인 경우에는 하나의 테이블을 이용하여 상기 AMVP 인덱스에 대응하는 코드워드를 선택한다.

[0163] 한편, 상기 머지 후보자 블록과 AMVP 후보자 블록을 동일하게 설정할 수 있다. 상기한 머지 후보자 구성과 동일한 구성을 갖는 AMVP 후보자 구성의 경우가 이에 해당한다. 이 경우에는 부호화기의 복잡도를 감소시킬 수 있다.

[0164] 도 7은 본 발명의 실시예에 따른 동영상 복호화 장치를 나타내는 블록도이다.

[0165] 도 7을 참조하면, 본 발명에 따른 동영상 복호화 장치는, 엔트로피 복호부(210), 역스캐닝부(220), 역양자화부(230), 역변환부(240), 인트라 예측부(250), 인터 예측부(260), 후처리부(270), 픽처 저장부(280), 가산부(290), 및 인트라/인터전환 스위치(295)를 구비한다.

[0166] 엔트로피 복호부(210)는, 수신된 부호화 비트 스트림을 복호하여, 인트라 예측 정보, 인터 예측 정보, 양자화 계수 정보 등으로 분리한다. 엔트로피 복호부(210)는 복호된 인터 예측 정보를 인터 예측부(260)에 공급한다. 엔트로피 복호부(210)는 인트라 예측 정보를 복호하여 인트라 예측부(250)로 공급한다. 또한, 상기 엔트로피 복호화(210)는 상기 역양자화 계수 정보를 역스캐닝부(220)로 공급한다.

[0167] 역스캐닝부(220)는 상기 양자화 계수 정보를 2차원 배열의 역양자화 블록으로 변환한다. 상기 변환을 위해 복수개의 역스캐닝 패턴 중에 하나를 선택한다. 계수 역스캐닝 패턴은 인트라 예측 모드에 따라 결정될 수 있다. 현재 변환 유닛의 크기가 미리 정해진 크기보다 큰 경우에는 미리 정해진 크기의 서브셋 단위로 역스캐닝하여 양자화된 변환 유닛을 구성한다. 그리고, 현재 변환 유닛의 크기가 미리 정해진 크기인 경우와 동일한 경우에는 변환 유닛 단위로 역스캐닝하여 양자화된 변환 유닛을 구성한다. 상기 양자화된 계수들이 복수개의 서브셋 단위로 역스캐닝된 경우에는 각각의 서브셋 내의 양자화 계수들에 동일한 역스캐닝패턴을 적용한다. 상기 복수개의 서브셋은 하나의 메인 서브셋과 적어도 하나 이상의 잔여 서브셋으로 구성된다. 메인 서브셋은 DC 계수를 포함하는 좌상측에 위치하고, 상기 잔여 서브셋은 메인 서브셋 이외의 영역을 커버한다.

[0168] 서브셋 간의 역스캐닝패턴은 역지그재그 스캔을 적용한다. 스캔 패턴은 잔여 서브셋들로부터 메인 서브셋으로 역스캐닝하는 것이 바람직하나, 그 역방향도 가능하다. 또한, 서브셋 내의 양자화된 계수들의 역스캐닝패턴과 동일하게 서브셋 간의 역스캐닝패턴을 설정할 수도 있다. 역스캐닝부(220)는 현재 변환 유닛 내의 0이 아닌 마지막 양자화 계수의 위치를 나타내는 정보를 이용하여 역스캐닝을 수행한다.

- [0169] 역양자화부(230)는 현재 코딩 유닛의 양자화 스텝 사이즈 예측자를 결정한다. 상기 예측자의 결정과정은 도 1의 양자화부(130)의 예측자 결정 과정과 동일하므로 생략한다. 역양자화부(230)는 결정된 양자화 스텝 사이즈 예측자와 수신한 잔차 양자화 스텝 사이즈를 더하여 현재 역양자화 블록에 적용된 양자화 스텝 사이즈를 얻는다. 역양자화부(230)는 양자화 스텝 사이즈가 적용된 양자화 매트릭스를 이용하여 역양자화 계수를 복원한다. 복원하고자 하는 현재 블록의 크기에 따라 서로 다른 양자화 매트릭스가 적용되며, 동일 크기의 블록에 대해서도 상기 현재 블록의 예측 모드 및 인트라 예측 모드 중 적어도 하나에 기초하여 양자화 매트릭스가 선택된다.
- [0170] 역변환부(240)는 역양자화 블록을 역변환하여 잔차 블록을 복원한다. 그리고, 상기 복원된 양자화 계수를 역변환하여 잔차 블록을 복원한다. 상기 역양자화 블록에 적용할 역변환 매트릭스는 예측 모드(인트라 또는 인터) 및 인트라 예측 모드에 따라 적응적으로 결정될 수 있다. 도 1의 변환부(120)에 적용된 변환 매트릭스의 역변환 매트릭스가 결정되므로 구체적인 기재는 생략한다.
- [0171] 가산부(290)는 역변환부(240)에 의해 복원된 잔차 블록과 인트라 예측부(250) 또는 인터 예측부(260)에 의해 생성되는 예측 블록을 가산함으로써, 영상 블록을 복원한다. 인트라 예측부(250)는 엔트로피 복호화부(210)로부터 수신된 인트라 예측 정보에 기초하여 현재 블록의 인트라 예측 모드를 복원한다. 그리고, 복원된 인트라 예측 모드에 따라 예측 블록을 생성한다.
- [0172] 인터 예측부(260)는 엔트로피 복호화부(210)로부터 수신된 인터 예측 정보에 기초하여 참조 픽처 인덱스와 움직임 벡터를 복원한다. 그리고, 상기 참조 픽처 인덱스와 움직임 벡터를 이용하여 현재 블록에 대한 예측 블록을 생성한다. 소수 정밀도의 움직임 보상이 적용될 경우에는 선택된 보간 필터를 적용하여 예측 블록을 생성한다.
- [0173] 후처리부(270)의 동작은 도 1의 후처리부(160)의 동작과 동일하므로 생략한다.
- [0174] 픽처 저장부(280)는 후처리부(270)에 의해 후처리된 복호 영상을 픽처 단위로 저장한다.
- [0175] 도 8은 본 발명에 따른 인터 예측 복호화 과정을 설명한다.
- [0176] 먼저, 복원하고자 하는 현재 블록이 SKIP 부호화 되었는지를 판단한다(S405). 이는 코딩 유닛의 skip\_flag를 통해 확인할 수 있다.
- [0177] 현재 블록이 SKIP 부호화된 경우에는 현재 블록의 skip 모드 움직임 정보 복호화 과정을 통해 현재 블록의 움직임 정보를 복호화한다(S410). skip 모드 움직임 정보 복호화 과정은 merge 모드 움직임 정보 복호화 과정과 동일하다.
- [0178] 그리고, 유도된 현재 블록의 움직임 정보가 나타내는 참조픽처의 대응 블록을 복사하여 복원블록을 생성한다(S415).
- [0179] 현재 블록이 SKIP 부호화되지 않았으면, 현재 블록의 움직임 정보가 merge 모드로 부호화되었는지를 판단한다(S420).
- [0180] 현재 블록의 움직임 정보가 merge 모드로 부호화 되었으면, 머지 모드 움직임 정보 복호화 과정을 통해 현재 블록의 움직임 정보를 복호화한다(S425).
- [0181] 그리고, 복호화된 현재 블록의 움직임 정보를 이용하여 예측 블록을 생성한다(S430).
- [0182] 현재 블록의 움직임 정보가 merge 모드로 부호화 되었으면, 잔차블록을 복호화한다(S435).
- [0183] 그리고, 상기 예측 블록과 잔차 블록을 이용하여 복원블록을 생성한다(S440)
- [0184] 한편, 현재 블록의 움직임 정보가 merge 모드로 부호화되지 않았으면, AMVP 모드 움직임 정보 복호화 과정을 통해 현재 블록의 움직임 정보를 복호화한다(S445).
- [0185] 그리고, 상기 복호화된 현재 블록의 움직임 정보를 이용하여 예측블록을 생성한다(S450). 그리고, 잔차블록을 복호화하고(S455), 상기 예측 블록과 잔차 블록을 이용하여 복원블록을 생성한다(S460).
- [0186] 움직임 정보 복호화 과정은 현재 블록의 움직임 정보의 부호화 패턴에 따라 달라진다. 현재 블록의 움직임 정보 부호화 패턴은 머지 모드, AMVP모드 중 하나일 수 있다. 스킵 모드의 경우에는 머지 모드와 동일한 움직임 정보 복호화 과정을 가지므로 생략한다.

- [0187] 먼저, 현재 블록의 움직임 정보 부호화 패턴이 머지 모드인 경우의 움직임 정보 부호화 과정에 대해 설명한다.
- [0188] 도 9은 머지 후보자 수가 고정일 경우의 머지 모드 움직임 벡터 부호화 과정을 도시한다.
- [0189] 먼저, 머지 코드워드가 존재하는지를 판단한다(S610).
- [0190] 머지 코드워드가 존재하지 않으면, 현재 블록의 머지 후보자가 1개인 것으로 판단하고, 유효한 머지 후보자를 검색한다(S620). 머지 후보자 구성 및 머지 후보자 검색 순서(즉, 리스트 구축순서)는 상기 도 3과 관련된 상세한 설명에 도시된 내용과 동일하다.
- [0191] 유효한 머지 후보자가 검색되면, 상기 머지 후보자의 움직임 정보, 즉 참조픽처 인덱스와 움직임 벡터를 현재 블록의 참조픽처 인덱스와 움직임 벡터로 설정한다(S630).
- [0192] 머지 코드워드가 존재하면, 머지 코드워드에 대응하는 머지 인덱스를 VLC 테이블을 이용하여 구한다(S640).
- [0193] 상기 머지 인덱스에 대응하는 머지 후보자를 구하기 위해 미리 정해진 순서에 따라 머지 후보자를 검색한다(S650). 머지 후보자 구성 및 머지 후보자 검색 순서(즉, 리스트 구축순서)는 상기 도 3과 관련된 상세한 설명에 도시된 내용과 동일하다.
- [0194] 인덱스에 대응하는 유효한 머지 후보자가 검색되면, 상기 머지 후보자의 움직임 정보를 현재 블록의 움직임 정보로 설정한다(S660).
- [0195] 다음으로, 현재 블록의 움직임 정보 부호화 패턴이 AMVP인 경우의 움직임 정보 부호화 과정에 대해 설명한다.
- [0196] 도 10은 AMVP 후보자 수가 고정일 경우의 움직임 벡터 부호화 과정을 도시한다.
- [0197] 현재 블록의 참조픽처 인덱스와 차분 움직임 벡터를 파악한다(S810)
- [0198] AMVP 코드워드가 존재하는지를 판단한다(S820).
- [0199] AMVP 코드워드가 존재하지 않으면, 현재 블록의 AMVP 후보자 수가 1인 것으로 판단하고, 유효한 AMVP 후보자를 검색한다(S830). AMVP 후보자 구성 및 AMVP 후보자 검색 순서(즉, 리스트 구축순서)는 상기 도 6과 관련된 상세한 설명에 도시된 내용과 동일하다.
- [0200] 유효한 AMVP 후보자가 검색되면, 상기 AMVP 후보자의 움직임 벡터를 현재 블록의 예측 움직임 벡터로 설정한다(S840).
- [0201] AMVP 코드워드가 존재하면, AMVP 코드워드에 대응하는 AMVP 인덱스를 VLC 테이블을 이용하여 구한다(S850).
- [0202] 상기 AMVP 인덱스에 대응하는 AMVP 후보자를 구하기 위해 미리 정해진 순서에 따라 AMVP 후보자를 검색한다(S860). AMVP 후보자 구성 및 AMVP 후보자 검색 순서(즉, 리스트 구축순서)는 상기 도 6과 관련된 상세한 설명에 도시된 내용과 동일하다.
- [0203] AMVP 인덱스에 대응하는 유효한 AMVP 후보자가 검색되면, 상기 AMVP 후보자의 움직임 벡터를 현재 블록의 예측 움직임 벡터로 설정한다(S870).
- [0204] 상기 S810단계에서 구한 차분 움직임 벡터 및 상기 S840 또는 S870 단계에서 구한 예측 움직임 벡터를 더하여 현재 블록의 최종 움직임 벡터로 설정한다(S880).
- [0205] 도 11은 본 발명에 따른 인터 예측 부호화 장치(300)를 나타내는 구성도이다.
- [0206] 본 발명에 따른 인터 예측 부호화 장치(300)는 디멀티플렉서(310), 움직임 정보 부호화 모드 판단부(320), 머지 모드 움직임 정보 부호화부(330), AMVP 모드 움직임 정보 부호화부(340), 예측블록 생성부(350), 잔차 블록 부호화부(360) 및 복원블록 생성부(370)를 포함한다.
- [0207] 디멀티플렉서(310)는 수신된 비트스트림으로부터 현재 부호화된 움직임 정보와 부호화된 잔차 신호들을 역다중화한다. 디멀티플렉서(310)는 상기 역다중화된 움직임 정보를 움직임 정보 부호화 모드 판단부(320)로 전송하고, 역다중화된 잔차신호를 잔차블록 부호화부(360)로 전송한다.

- [0208] 움직임 정보 부호화 모드 판단부(320)는 현재 블록의 움직임 정보 부호화 모드를 판단한다. 움직임 정보 부호화 모드 판단부(320)는 수신된 비트스트림의 skip\_flag가 1의 값을 갖는 경우에는 현재 블록의 움직임 정보 부호화 모드가 스킵 부호화 모드로 부호화된 것으로 판단하여 스킵 인덱스를 추출한다. 스킵 후보자와 머지 후보자는 동일하다. 따라서, 스킵 인덱스는 머지 인덱스와 동일하다. 움직임 정보 부호화 모드 판단부(320)는 수신된 비트스트림의 skip\_flag가 0의 값을 갖고, 디멀티플렉서(310)로부터 수신된 움직임 정보 중 merge\_flag가 1이면, 현재 블록의 움직임 정보 부호화 모드가 머지 모드로 부호화된 것으로 판단한다. 그러나, merge\_flag가 0이면, 현재 블록의 움직임 정보 부호화 모드가 AMVP 모드로 부호화된 것으로 판단한다.
- [0209] 머지 모드 움직임 정보 복호화부(330)는 움직임 정보 부호화 모드 판단부(320)가 현재 블록의 움직임 정보 부호화 모드를 스킵 또는 머지 모드로 판단한 경우에 활성화된다.
- [0210] AMVP 모드 움직임 정보 복호화부(340)는 움직임 정보 부호화 모드 판단부(320)가 현재 블록의 움직임 정보 부호화 모드를 AMVP 모드로 판단한 경우에 활성화된다.
- [0211] 예측블록 생성부(350)는 상기 머지 모드 움직임 정보 복호화부(330) 또는 AMVP 모드 움직임 정보 복호화부(340)에 의해 복원된 움직임 정보를 이용하여 현재 블록의 예측 블록을 생성한다. 움직임 벡터가 정수 단위일 경우에는, 참조픽처 인덱스가 나타내는 픽처 내의 움직임 벡터가 나타내는 위치에 대응하는 블록을 복사하여 현재 블록의 예측 블록을 생성한다. 그러나, 움직임 벡터가 정수 단위가 아닐 경우에는, 참조픽처 인덱스가 나타내는 픽처내의 정수 단위 화소들로부터 예측 블록의 화소들을 생성한다. 이 경우, 휘도 화소의 경우에는 8탭의 보간 필터를 사용하여 예측 화소를 생성할 수 있다. 색차 화소의 경우에는 4탭 보간 필터를 사용하여 예측 화소를 생성할 수 있다.
- [0212] 잔차 블록 복호화부(360)는 잔차신호를 엔트로피 복호화한다. 그리고, 엔트로피 복호화된 계수들을 역스캐닝하여 2차원의 양자화된 계수 블록을 생성한다. 역스캐닝 방식은 엔트로피 복호화 방식에 따라 달라질 수 있다. 즉, CABAC 기반으로 복호화된 경우와 CAVLC 기반으로 복호화된 경우의 인터 예측 잔차신호의 역스캐닝 방식이 달라질 수 있다. 예를 들어, CABAC 기반으로 복호화된 경우에는 대각선 방향의 래스터 역스캔 방식으로, CAVLC 기반으로 복호화된 경우에는 지그재그 역스캔 방식이 적용될 수 있다. 또한, 예측 블록의 크기에 따라 역스캐닝 방식이 달리 결정될 수도 있다.
- [0213] 잔차블록 복호화부(360)는 생성된 계수블록을 역양자화 매트릭스를 이용하여 역양자화한다. 상기 양자화 매트릭스를 유도하기 위해 양자화 스텝 사이즈가 코딩 유닛별로 복원된다. 양자화 스텝 사이즈는 미리 정해진 크기 이상의 코딩 유닛별로 복원된다. 상기 미리 정해진 크기는 8x8 또는 16x16일 수 있다.
- [0214] 상기 미리 정해진 크기 이상의 코딩 유닛별로 결정된 양자화 스텝 사이즈를 복원하기 위해 현재 코딩 유닛에 인접한 코딩 유닛의 양자화 스텝 사이즈를 이용한다. 현재 코딩 유닛의 좌측 코딩 유닛, 상측 코딩 유닛, 좌상측 코딩 유닛 순서로 검색하여 1개 또는 2개의 유효한 양자화 스텝 사이즈를 이용하여 현재 코딩 유닛의 양자화 스텝 사이즈 예측자를 생성한다. 예를 들어, 상기 순서로 검색된 유효한 첫번째 양자화 스텝 사이즈를 양자화 스텝 사이즈 예측자로 결정할 수 있다. 또한, 좌측 예측 유닛, 부호화 순서상 바로 이전의 코딩 유닛 순으로 검색하여 유효한 첫번째 양자화 스텝 사이즈를 양자화 스텝 사이즈 예측자로 결정할 수 있다. 상기 결정된 예측자와 차분 양자화 스텝 사이즈를 이용하여 현재 예측 유닛의 양자화 스텝 사이즈를 복원한다.
- [0215] 잔차블록 복호화부(360)는 상기 역양자화된 계수 블록을 역변환하여 잔차블록을 복원한다.
- [0216] 복원블록 생성부(370)는 상기 예측블록 생성부(350)에 의해 생성된 예측블록과 상기 잔차블록 복호화부(260)에 의하여 생성된 잔차블록을 더하여 복원블록을 생성한다.
- [0217] 도 12는 본 발명의 일 실시예에 따른 머지 모드 움직임 정보 복호화부(330)를 나타내는 블록도이다.
- [0218] 본 발명의 일 실시예에 따른 머지 모드 움직임 정보 복호화부(330)는 머지 예측자 인덱스 복호화부(331), 공간

머지 후보자 유도부(332), 시간 머지 후보자의 참조 픽처 인덱스 유도부(333), 시간 머지 후보자의 움직임 벡터 유도부(334), 시간 머지 후보자 구성부(335), 머지 리스트 구축부(336), 머지 후보자 생성부(337), 머지 예측자 선택부(338) 및 움직임 정보 생성부(339)를 포함한다.

- [0219] 머지 예측자 인덱스 복호화부(331)는 미리 정해진 복호화 테이블을 이용하여 수신된 머지 예측자 코드워드에 대응하는 머지 예측자 인덱스를 복원한다.
- [0220] 공간 머지 후보자 유도부(332)는 현재 블록에 인접한 블록의 유효한 움직임 정보를 공간 머지 후보자로 설정한다. 공간 머지 후보자는 도 5에 도시된 바와 같이 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 현재 블록의 좌하측 블록(블록 D) 및 현재 블록의 좌상측 블록(블록 E) 중 최대 4개가 후보자가 될 수 있다. 이 경우, 블록 E는 블록 A, B, C, D 중 하나 이상이 유효하지 않은 경우에 사용될 수 있다.
- [0221] 또한, 공간 머지 후보자는 현재 블록의 좌측 블록(블록 A'), 현재 블록의 상측 블록(블록 B'), 현재 블록의 코너 블록(C, D, E 중 하나)로 설정될 수 있다. 코너 블록은 우상측 블록(블록 C), 현재 블록의 좌하측 블록(블록 D) 및 현재 블록의 좌상측 블록(블록 E) 순으로 스캔하여 유효한 첫번째 블록이다. 여기서 블록 A'은 현재 블록의 좌측 블록 중 E에 접한 블록일 수 있다. 블록 B'은 현재 블록의 상측 블록 중 E에 접한 블록일 수 있다.
- [0222] 또한, 공간 머지 후보자는 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 현재 블록의 좌하측 블록(블록 D) 및 현재 블록의 좌상측 블록(블록 E) 순으로 스캔하여 유효한 2개가 후보자가 될 수 있다.
- [0223] 시간 머지 후보자의 참조 픽처 인덱스 유도부(333)는 현재 블록의 시간 머지 후보자를 위한 참조 픽처 인덱스를 구한다. 시간 머지 후보자의 참조 픽처 인덱스는 현재 블록에 공간적으로 인접한 블록들(예측 유닛들)의 참조 픽처 인덱스들을 이용하여 구할 수도 있다.
- [0224] 구체적으로, 현재 블록의 시간 머지 후보자의 참조 인덱스들을 구하기 위해 현재 블록의 좌측 블록(A), 상측 블록(B), 우상측 블록(C), 좌하측 블록(D) 및 좌상측 블록(E)의 참조 픽처 인덱스들 중 일부 또는 전부가 사용될 수 있다. 예를 들어, 현재 블록의 좌측 블록(A), 상측 블록(B) 및 코너 블록(C, D, E 중 어느 하나)의 참조 픽처 인덱스가 사용될 수 있다. 이 경우, 코너 블록의 참조 픽처 인덱스는 우상측 블록(C), 좌하측 블록(D), 좌상측 블록(E) 순으로 스캔하여 유효한 첫번째 블록의 참조 픽처 인덱스로 결정될 수 있다. 또한, 현재 블록의 좌측 블록(A), 상측 블록(B), 우상측 블록(C), 좌하측 블록(D) 및 좌상측 블록(E)의 순으로 스캔하여 유효한 홀수개(예를 들어, 3개)의 블록의 참조 픽처 인덱스가 사용될 수도 있다.
- [0225] 여기서 좌측 블록(A)은 블록 D에 인접하지 않고 블록 E에 인접한 블록일 수도 있다. 마찬가지로, 상측 블록(B)은 블록 C에 인접하지 않고 블록 E에 인접한 블록일 수도 있다.
- [0226] 현재 블록의 시간 머지 후보자의 참조 인덱스들을 구하기 위해, 현재 블록의 좌측 블록(A), 상측 블록(B) 및 코너 블록(C, D, E 중 어느 하나)들의 참조 픽처 인덱스가 사용되는 경우를 예로 들어 설명한다.
- [0227] 현재 블록의 좌측 블록의 참조 픽처 인덱스(이하, 좌측 참조 픽처 인덱스), 상측 블록의 참조 픽처 인덱스(이하, 상측 참조 픽처 인덱스) 및 코너 블록의 참조 픽처 인덱스(이하, 코너 참조 픽처 인덱스)를 구한다. 상기 참조 픽처 인덱스들 중 유효한 참조 픽처 인덱스들 중 가장 많은 빈도를 갖는 참조 픽처 인덱스를 시간 스킵 후보자의 참조 픽처 인덱스로 설정된다. 유효한 후보자들 중 가장 많은 빈도를 갖는 참조 픽처 인덱스가 복수개 존재할 경우에는 상기 복수개의 참조 픽처 인덱스 중 최소값을 갖는 참조 픽처 인덱스를 시간 스킵 후보자를 위한 참조 픽처 인덱스로 설정한다.
- [0228] 다음으로, 현재 블록의 시간 머지 후보자의 참조 인덱스들을 구하기 위해, 현재 블록의 좌측 블록(A), 상측 블록(B), 우상측 블록(C), 좌하측 블록(D) 및 좌상측 블록(E)의 순으로 스캔하여 유효한 3개의 블록의 참조 픽처 인덱스가 사용되는 경우에 대해 예시적으로 설명한다. 상기 블록들의 스캔 순으로 스캔하여 유효한 3개의 참조 픽처 인덱스를 구한다. 여기에서는 3개의 유효한 참조 픽처 인덱스만을 이용하는 것으로 설정하였지만, 유효한 모든 개수를 이용할 수도 있다. 또한, 3개가 존재하지 않을 경우에는 유효한 개수들만을 이용하거나 인덱스 0의



값을 추가할 수도 있다.

- [0229] 상기 참조 픽처 인덱스들 중 유효한 참조 픽처 인덱스들 중 가장 많은 빈도를 갖는 참조 픽처 인덱스를 시간 머지 후보자의 참조 픽처 인덱스로 설정된다. 유효한 후보자들 중 가장 많은 빈도를 갖는 참조 픽처 인덱스가 복수 개 존재할 경우에는 상기 복수개의 참조 픽처 인덱스 중 최소값을 갖는 참조 픽처 인덱스 또는 좌측 블록 또는 상측 블록의 참조 픽처 인덱스를 시간 머지 후보자의 참조 픽처 인덱스로 설정한다.
- [0230] 시간 머지 후보자의 움직임 벡터 유도부(334)는 상기 시간 머지 후보자 블록이 속하는 픽처(이하, 시간 머지 후보자 픽처)를 결정한다. 시간 머지 후보자 픽처는 참조 픽처 인덱스가 0인 픽처로 설정될 수 있다. 이 경우, 슬라이스 타입이 P인 경우에는 리스트 0(list0)의 첫번째 픽처(즉 인덱스가 0인 픽처)가 시간 머지 후보자 픽처로 설정된다. 슬라이스 타입이 B인 경우에는 슬라이스 헤더내의 시간 머지 후보자 리스트를 나타내는 플래그가 나타내는 참조픽처 리스트의 첫번째 픽처가 시간 머지 후보자 픽처로 설정된다. 예를 들어, 상기 플래그가 1을 나타내면 list0로부터, 0을 나타내면 list1으로부터 시간 머지 후보자 픽처를 설정할 수 있다.
- [0231] 다음으로, 상기 시간 머지 후보자 픽처 내의 시간 머지 후보자 블록을 구한다. 상기 시간 머지 후보자 블록으로서, 상기 시간 머지 후보자 픽처 내의 현재 블록에 대응하는 복수개의 대응 블록 중 어느 하나가 선택될 수 있다. 이 경우, 복수개의 대응 블록들에 우선순위를 부여하고, 상기 우선순위에 기초하여 유효한 첫번째 대응 블록이 시간 머지 후보자 블록으로 선택될 수 있다. 그러나, 상기 시간 머지 후보자 블록의 위치는 1개일 수 있다. 상기 시간 머지 후보자 블록의 위치는 현재 블록의 LCU 내에서의 위치에 따라 달라질 수 있다. 즉, 현재 블록의 오른쪽 경계가 LCU 경계에 접하는지 여부 및 현재 블록의 아래쪽 경계가 LCU 경계에 접하는지에 따라 달라질 수 있다.
- [0232] 한편, 현재 블록의 크기에 따라 상기 시간 머지 후보자를 적응적으로 off 시킬 수도 있다. 예를 들어, 4x4 블록 이 경우에는 복잡도 감소를 위해 상기 시간 머지 후보자를 off 시킬 수 있다.
- [0233] 시간 머지 후보자 구성부(335)는 상기 시간 머지 후보자의 참조 픽처 인덱스 유도부(333)로부터 구한 참조 픽처 인덱스와 상기 시간 머지 후보자의 움직임 벡터 유도부(334)로부터 구한 움직임 벡터를 시간 머지 후보자의 참조픽처 인덱스와 움직임 벡터로 구성한다.
- [0234] 머지 리스트 구축부(336)는 유효한 머지 후보자를 이용하여 머지 후보자 리스트를 구축한다. 머지 후보자는 유효한 공간 머지 후보자에 작은 번호를 먼저 부여하고, 그 이후에 시간 머지 후보자(이하, Col로 표현)에 번호를 부여할 수 있다. 공간 머지 후보자들 사이의 우선순위는 도 5의 A, B, C, D, E 순서일 수 있다. 이 경우, 블록 E는 블록 A, B, C, D 중 어느 하나가 유효하지 않은 경우에 사용될 수 있다.
- [0235] 이 경우, 복수개의 머지 후보자가 동일 움직임 정보(동일 움직임 벡터 및 동일 참조픽처 인덱스)를 가지면, 후 순위의 머지 후보자를 리스트에서 삭제한다. 여기서, 움직임 벡터가 동일하지 않더라도 움직임 벡터만 소정 범위 차이가 나는 경우에도 후순위 머지 후보자를 리스트에서 삭제할 수도 있다. 상기 소정 범위는 미리 정해진 값이거나, 부호기에서 결정된 값일 수 있다. 부호기에서 결정된 값일 경우 복호기로 전송된다.
- [0236] 상기한 머지 후보자의 우선순위가 A, B, Col, C, D, E 순서일 수도 있다. 또한, A, B, C, D 중 하나 이상이 유효하지 않은 경우에는 상기 유효하지 않은 최초 블록의 위치에 상기 유효한 E의 움직임 정보를 삽입할 수도 있다.
- [0237] 한편, 상기 유효한 머지 후보자 수가 미리 정해진 수(예를 들어, 5)보다 작으면, 머지 후보자 생성부(337)에 머지 후보자 생성을 요청한다. 그리고, 상기 머지 후보자 생성부(337)로부터 수신된 머지 후보자를 상기 유효한 머지 후보자에 후순위로 추가한다. 이 경우, 머지 예측자 인덱스 복호화부로부터 수신되는 머지 예측자 인덱스에 대응하는 머지 후보자가 머지 후보자 리스트에 포함되면, 그 이후의 머지 후보자 추가를 하지 않을 수도 있다.
- [0238] 머지 후보자 생성부(337)는 유효한 머지 후보자 수(중복 제거)가 미리 정해진 수보다 작을 경우에 활성화된다. 머지 후보자 생성부(337)는 현재 슬라이스의 타입에 따라 머지 후보자 생성 방법을 달리한다.
- [0239] 슬라이스 타입이 P인 경우에는 참조픽처가 리스트 0의 참조픽처 인덱스 0인 참조픽처를 선택하고, 움직임 벡터

가 0인 머지 후보자를 추가한다.

- [0240] 슬라이스 타입이 B인 경우에는 복수개의 머지 후보자 생성 방법을 미리 정해진 순서에 따라 이용하여 머지 후보자를 생성한다. 제1, 제2, 제3 방법이 존재할 수 있다. 머지 후보자 수가 미리 정해진 수만큼 채워질때까지 진행한다.
- [0241] 제1 방법은 유효한 머지 후보자 수가 2개 이상이고, 양방향 움직임 정보가 존재하는 경우에 적용할 수 있다. 유효한 제1 머지 후보자의 움직임 정보가 리스트 A의 움직임 정보라고 할 때, 유효한 제2 머지 후보자의 움직임 정보가 상기 리스트와 다른 리스트의 움직임 정보이거나 이를 포함할 경우에 적용된다. 상기 제1 머지 후보자의 리스트 A 움직임 정보와 제2 머지 후보자의 리스트 B (B는 A와 다른 값임)의 움직임 정보를 결합하여 양방향 움직임 정보를 구성하고, 이를 후보자로 추가한다. 상기 추가될 수 있는 후보자가 복수인 경우에는 미리 정해진 순서에 따라 추가한다. 상기 미리 정해진 순서는 유효한 머지 후보자의 인덱스에 따라 결정된다.
- [0242] 제2 방법은 유효한 머지 후보자가 1개 이상 존재할 때 적용 가능하다. 상기 제2 방법에 의해 생성되는 후보자는 다음의 움직임 정보를 갖는다.
- [0243] - 유효한 머지 후보자의 참조 픽처 리스트와 다른 참조 픽처 리스트
- [0244] - 유효한 머지 후보자의 참조 픽처와 동일한 참조 픽처가 아니고, 현재 픽처와 머지 후보자 참조 픽처와의 시간적 거리와 동일한 거리만큼 떨어져 있는 참조 픽처
- [0245] - 유효한 머지 후보자의 움직임 벡터와 반대 방향의 동일한 크기의 움직임 벡터
- [0246] 제 3 방법은 움직임 벡터값이 0인 머지 후보자를 추가하는 것이다. 움직임 벡터값이 0인 머지 후보자는 3가지가 존재할 수 있다. 1) 움직임 벡터가 0이고, 참조픽처 리스트가 0이고, 참조픽처 인덱스가 0인 머지 후보자(단방향 L0 머지 후보자), 2) 움직임 벡터가 0이고, 참조픽처 리스트가 1이고, 참조픽처 인덱스가 0인 머지 후보자(단방향 L1 머지 후보자), 3) 상기 1), 2)의 머지 후보자를 결합한 양방향 머지 후보자가 존재할 수 있다.
- [0247] 결과적으로, 슬라이스 타입이 B인 경우에는 제1 방법, 제2 방법, 제3 방법 순으로 머지 후보자 생성 순서를 적용할 수 있다. 제1 방법에 의해 머지 후보자 수가 채워지면, 제2, 3 방법을 적용하지 않는다. 그러나, 제1 방법에 의해 머지 후보자 수가 채워지지 않으면, 제2 방법을 적용하고, 제2 방법에 의해 추가되지 않을 경우에는 제3 방법을 적용할 수 있다. 제3 방법 적용시에는 양방향 머지 후보자만을 추가할 수도 있고, 단방향 머지 후보자를 후순위로 추가할 수도 있다.
- [0248] 슬라이스 타입이 B이지만, 참조 픽처들이 현재 픽처보다 시간적으로 선행하는 경우(이하, 'low delay의 경우'라 함)에는 머지 후보자 추가시 상기 제2 방법을 적용하지 않는다.
- [0249] 한편, 슬라이스 타입이 P인 경우, 2개 이상의 머지 후보자가 존재하고, 동일 리스트에 속할 경우, 상기 2개 이상의 머지 후보자를 이용하여 새로운 머지 후보자를 생성할 수 있다. 예를 들어, 참조 픽처 인덱스가 동일할 경우, 상기 참조 픽처 인덱스를 갖고, 2개의 머지 후보자의 움직임 벡터의 평균값 또는 3개의 머지 후보자의 미디언 값을 추가할 수도 있다. 슬라이스 타입이 B인 경우에도 적용 가능하다.
- [0250] 또는 상기 0 움직임 벡터를 추가한 후에도 후보자 수를 채우지 못할 경우, 상기 0 움직임 벡터 외에 영벡터 근처의 움직임 벡터를 추가할 수도 있다. 또는, 유효한 머지 후보자의 움직임 벡터를 조금 변형시켜 머지 후보자를 생성할 수도 있다.
- [0251] 머지 예측자 선택부(338)는 상기 머지 예측자 인덱스 복호화부(331)에 의해 복원된 머지 예측자 인덱스에 대응하는 머지 후보자를 머지 후보자 리스트에서 선택하여 현재 블록의 머지 예측자로 선택한다
- [0252] 움직임 정보 생성부(338)는 상기 머지 후보자 인덱스에 대응하는 머지 예측자를 상기 머지 리스트 구축부(336)에 의해 구축된 리스트에서 선택하고, 상기 선택된 머지 예측자의 움직임 정보(즉, 움직임 벡터 및 참조픽처 인덱스)를 현재 블록의 움직임 정보로 결정한다.

- [0253] 도 13은 본 발명의 일실시예에 따른 AMVP 모드 움직임 정보 복호화부를 나타내는 구성도이다.
- [0254] 본 발명의 실시예에 따른 AMVP 모드 움직임 정보 복호화부는 AMVP 예측자 인덱스 복호화부(341), 잔차 움직임 정보 복호화부(342), 공간 AMVP 후보자 유도부(343), 시간 AMVP 후보자 유도부(344), AMVP 리스트 구축부(345), AMVP 후보자 생성부(346), AMVP 예측자 선택부(347) 및 움직임 정보 생성부(349)를 포함한다.
- [0255] AMVP 예측자 인덱스 복호화부(341)는 AMVP 예측자 인덱스를 복호화한다. 상기 인덱스가 CAVLC로 부호화된 경우에는 복호화 후, unsigned integer 곱셈 코드에 대응하는 인덱스를 이용하여 현재 블록의 AMVP 예측자 인덱스로 결정할 수 있다.
- [0256] 잔차 움직임 정보 복호화부(342)는 현재 블록의 참조 픽처 인덱스와 차분 움직임 벡터를 복호화한다. CAVLC로 부호화된 경우에는 복호화 후, 참조픽처 인덱스는 unsigned integer 곱셈 코드에 대응하는 값을 이용해 참조 픽처 인덱스를 구하고, 차분 움직임 벡터는 unsigned integer 곱셈 코드에 대응하는 값을 이용해 복원한다.
- [0257] 공간 AMVP 후보자 유도부(343)는 현재 블록에 인접한 미리 정해진 위치의 블록들의 움직임 정보를 검색하여, 공간 AMVP 후보자로 선택한다. 예를 들어, 도 5에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A)과 좌하측 블록(블록 D)들 중 하나를 좌측 후보자로 택하고, 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 및 현재 블록의 좌상측 블록(블록 E)들 중 하나를 상측 후보자로 택할 수 있다. 이때 미리 정해진 순서로 스캔하여 유효한 첫번째 블록의 움직임 벡터가 후보자로 선택된다. 움직임 벡터가 존재하지 않는 경우에는 이용 가능하지 않은 것으로 설정한다.
- [0258] 공간 AMVP 후보자 위치의 움직임 벡터는 현재 예측 블록의 움직임 벡터와 비교하여 4종류가 있다. 즉, 1) 동일 참조 픽처를 갖고 동일 리스트를 갖는 움직임 정보(mv1), 2) 동일 참조 픽처를 갖고 다른 리스트를 갖는 움직임 정보(mv2), 3) 다른 참조 픽처를 갖고 동일 리스트를 갖는 움직임 정보(mv3), 및 4) 다른 참조 픽처를 갖고 다른 리스트를 갖는 움직임 정보(mv4)가 존재할 수 있다. 이 경우, mv1, mv2는 동일 참조 픽처를 나타내므로, 움직임 벡터를 스케일링 할 필요가 없으나, mv3은 움직임 벡터를 스케일링 해야 하고, mv4는 스케일링을 해야 할 경우들이 발생한다.
- [0259] 이하, 공간 AMVP 후보자를 선택하는 방법에 대해 설명한다.
- [0260] <mv1, mv2만을 이용하는 경우>
- [0261] 스케일링 여부를 판단할 필요가 없는 mv1, mv2만을 이용하여 좌측 후보자 및 상측 후보자를 선택할 수 있다. 먼저, 좌하측 블록의 mv1, mv2순으로, 움직임 벡터가 존재하는지를 판단한다. 존재하면, 상기 움직임 벡터를 좌측 후보자로 설정한다. 그러나, 존재하지 않으면, 좌측 블록(A)의 mv1, mv2 순으로, 움직임 벡터가 존재하는지를 판단한다. 존재하면, 상기 움직임 벡터를 좌측 후보자로 설정한다. 다음으로, 우상측 블록(C), 상측 블록(B), 좌상측 블록(E) 순으로 mv1, mv2가 존재하는지를 판단하고, 존재하는 첫번째 움직임 벡터를 상측 후보자로 설정한다.
- [0262] <mv1, mv2, mv3, mv4를 이용하는 경우>
- [0263] 먼저, 좌측 후보자를 설정한다. 구체적으로, 좌하측 블록(D)의 움직임 벡터가 mv1, mv2순으로, 존재하는지를 판단한다. 존재하면, 상기 움직임 벡터를 좌측 후보자로 설정한다. 그러나, 존재하지 않으면, 좌측 블록(A)의 움직임 벡터가 mv1, mv2순으로 존재하는지를 판단한다. 존재하면, 상기 움직임 벡터를 좌측 후보자로 설정한다. 그러나, 좌측 블록(A)의 mv1, mv2에 대응하는 움직임 벡터가 존재하지 않으면, 좌하측 블록(D)의 움직임 벡터가 존재하는지를 mv3, mv4 순으로 판단한다. 존재하면, 상기 존재하는 움직임 벡터를 현재 픽처와 참조 픽처간의 거리 및 상기 좌하측 블록(D)과 상기 좌하측 블록(D)의 참조 픽처간의 거리를 이용하여 스케일링하고, 좌측 후보자로 설정한다. 존재하지 않으면, 좌측 블록(A)의 움직임 벡터가 존재하는지를 mv3, mv4 순으로 판단하여 존재하면, 스케일링하여 좌측 후보자로 설정한다.
- [0264] 다음으로, 상측 후보자는 좌측 후보자의 종류에 의존하여 설정될 수 있다. 구체적으로, 우상측 블록(C)의 움직

임 벡터가 존재하는지를 mv1, mv2순서로 판단하고, 존재하면 상기 움직임 벡터를 상측 후보자로 설정한다. 존재하지 않으면, 상측 블록(B)의 움직임 벡터가 존재하는지를 mv1, mv2순서로 판단하고, 존재하면 상기 움직임 벡터를 상측 후보자로 설정한다. 존재하지 않으면, 좌상측 블록(E)의 움직임 벡터가 존재하는지를 mv1, mv2순서로 판단하고, 존재하면 상기 움직임 벡터를 상측 후보자로 설정한다.

[0265] 그러나, 존재하지 않으면, 좌측 후보자가 mv1 또는 mv2에 대응하거나 존재하지 않으면, 상기한 순서로 다시 mv3, mv4를 스캔하여 첫번째 만나는 움직임 벡터를 스케일링 한 후 상측 후보자로 설정한다. 그러나, mv3 또는 mv4에 대응하는 좌측 후보자가 존재하면, 상측 후보자는 이용 가능하지 않는 것으로 판단한다.

[0266] 미리 정해진 순서는 좌측 블록의 경우에는 블록 A, 블록 D순으로, 상측 블록의 경우에는 블록 B, 블록 C, 블록 E 순일 수도 있다.

[0267] 한편, 복수개의 공간 AMVP 후보자는 도 4에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 및 현재 블록의 좌하측 블록(블록 D) 순으로 스캔하여 유효한 2개의 블록을 후보자로 선택할 수 있다. 이 경우, 유효한 블록이 모두 후보자가 되거나 A, B, C, D 순으로 스캔하여 유효한 2개가 후보자가 될 수도 있다. 현재 블록의 좌측에 복수개의 블록이 존재할 경우에는 유효한 가장 상측에 존재하는 블록 또는 가장 큰 면적을 갖는 유효한 블록이 좌측 블록으로 설정될 수 있다. 마찬가지로, 현재 블록의 상측에 복수개의 블록이 존재할 경우에는 유효한 가장 좌측에 존재하는 블록 또는 가장 큰 면적을 갖는 블록이 상측 블록으로 설정될 수 있다.

[0268] 이 경우, 상기한 (a)에서의 방식과 유사하게 2개의 후보자를 설정할 수 있다. 먼저, 미리 정해진 순서로, 각 블록에 대하여 mv1, mv2에 대한 움직임 벡터가 존재하는지를 판단하고, 존재하면 그 움직임 벡터를 후보자로 설정한다. 2개의 후보자가 설정될 때까지 검색한다. 그러나, 상기 블록들에 대해 모두 검사를 하였으나, 2개의 후보자가 설정되지 않으면, mv3, mv4에 대하여 미리 정해진 순서로 존재하는 움직임 벡터를 구한 후 스케일링 하여 후보자로 설정할 수 있다. 이 경우, mv1, mv2에 대한 후보자가 하나도 존재하지 않을 경우에는, mv3, mv4에 대하여 하나의 후보자만을 설정한다. 복잡도를 더 감소시키기 위해 mv1, mv2에 대해서만 후보자가 존재하는지를 판단할 수도 있다.

[0269] 복수개의 공간 AMVP 후보자는, 도 5에 도시된 바와 같이, 현재 블록의 좌측 블록(블록 A), 현재 블록의 상측 블록(블록 B), 현재 블록의 우상측 블록(블록 C), 현재 블록의 좌하측 블록(블록 D), 현재 블록의 좌상측 블록(블록 E)들 순으로 스캔하여 유효한 2개의 블록을 후보자로 선택할 수 있다. 여기서 좌측 블록은 블록 D에 인접하지 않고 블록 E에 인접한 블록일 수 있다. 마찬가지로, 상측 블록은 블록 C에 인접하지 않고 블록 E에 인접한 블록일 수 있다. 이 경우에는 상기 정해진 순서의 블록들 순서로, 각 블록에 대하여mv1, mv2 순으로, 움직임 벡터가 존재하는지를 판단한다. 존재하면, 상기 움직임 벡터를 상기 블록들 순서로 후보자로 설정한다. 후보자 수가 채워지면 멈춘다. 그러나, 움직임 벡터가 존재하지 않거나, 후보자 수가 채워지지 않으면, mv3, mv4에 대하여 동일한 방식으로 후보자 수가 채워질 때까지 진행한다.

[0270] 시간 AMVP 후보자 유도부(344)는 먼저, 상기 시간 AVMP 후보자 블록이 속하는 픽처(이하, 시간 AVMP 후보자 픽처)를 결정한다. 시간 AVMP 후보자 픽처는 참조 픽처 인덱스가 0인 픽처로 설정될 수 있다. 이 경우, 슬라이스 타입이 P인 경우에는 리스트 0(list0)의 첫번째 픽처(즉 인덱스가 0인 픽처)가 시간 AVMP 후보자 픽처로 설정된다. 슬라이스 타입이 B인 경우에는 슬라이스 헤더내의 시간 AVMP 후보자 리스트를 나타내는 플래그(a flag indicating whether the temporal skip candidate picture is included in list0 or list1)가 나타내는 리스트의 첫번째 픽처가 시간 머지 후보자 픽처로 설정된다. 다음으로, 상기 시간 AVMP 후보자 픽처 내의 시간 AVMP 후보자 블록을 구한다. 이 과정은 상기한 상기 시간 머지 후보자 블록을 구하는 과정과 동일하므로 생략한다.

[0271] 한편, 현재 블록의 크기에 따라 상기 시간 AVMP 후보자를 적응적으로 off 시킬 수도 있다. 예를 들어, 4x4 블록 이 경우에는 복잡도 감소를 위해 상기 시간 AVMP 후보자를 off 시킬 수 있다. 이 경우에는 시간 AMVP 후보자 유도부(344)는 활성화되지 않는다.

[0272] AMVP 리스트 구축부(345)는 유효한 공간 AMVP 후보자 및 시간 AMVP 후보자를 이용하여 미리 정해진 우선순위에

따라 AMVP 리스트를 구축한다. 공간 AMVP 후보자가 높은 우선순위를 가진다. 복수개의 AVMP 후보자가 동일 움직임 벡터를 가지면(참조픽처가 동일할 필요는 없음), 후순위의 AVMP 후보자를 리스트에서 삭제한다. 여기서, 움직임 벡터가 동일하지 않더라도 움직임 벡터만 소정 범위 차이가 나는 경우에도 후순위 머지 후보자를 리스트에서 삭제할 수도 있다. 상기 소정 범위는 미리 정해진 값이거나, 부호기에서 결정된 값일 수 있다. 부호기에서 결정된 값일 경우 복호기로 전송된다.

[0273] AMVP 리스트 구축부(345)는 상기 유효한 AMVP 후보자 수가 미리 정해진 수(예를 들어, 2)보다 작으면, AMVP 후보자 생성부(346)에 AMVP 후보자 생성을 요청한다. 그리고, AMVP 후보자 생성부(346)로부터 수신된 AMVP 후보자를 상기 유효한 AMVP 후보자에 후순위로 추가한다. 이 경우, AMVP 예측자 인덱스 복호화부(341)로부터 수신되는 AMVP 예측자 인덱스에 대응하는 AMVP 후보자가 AMVP 후보자 리스트에 포함되면, 그 이후의 AMVP 후보자 추가를 하지 않을 수도 있다.

[0274] AMVP 후보자 생성부(347)는 AVMP 후보자 유효한 AMVP 후보자 수가 미리 정해진 수보다 작을 경우에 활성화된다. 추가되는 AMVP 후보자는 움직임 벡터가 0 인 후보자가 추가될 수 있다.

[0275] 또는 상기 0 움직임 벡터를 추가한 후에도 후보자 수를 채우지 못할 경우, 상기 0 움직임 벡터 외에 영벡터 근처의 움직임 벡터를 추가할 수도 있다. 또는, 유효한 머지 후보자의 움직임 벡터를 조금 변형시켜 머지 후보자를 생성할 수도 있다.

[0276] AMVP 예측자 선택부(347)는 상기 AMVP 예측자 인덱스 복호화부(341)에 의해 복원된 AMVP 예측자 인덱스에 대응하는 AMVP 후보자를 AMVP 후보자 리스트에서 선택하여 현재 블록의 AMVP 예측자로 결정한다. 그리고, 상기 선택된 AMVP 예측자를 현재 블록의 예측 움직임 벡터로 설정한다.

[0277] 움직임 정보 생성부(348)는 AMVP 예측자 선택부(347)에 의해 생성된 예측 움직임 벡터와 잔차 움직임 정보 복호화부(342)에 의해 판독된 잔차 움직임 벡터를 더하여 현재 블록의 움직임 벡터를 생성한다. 그리고, 잔차 움직임 정보 복호화부(342)에 의해 복원된 참조 픽처 인덱스를 현재 블록의 참조 픽처 인덱스로 설정한다.

[0278] 한편, 시간 머지 후보자 또는 시간 AMVP 후보자를 구하기 위해서는 시간 머지 후보자 픽처 또는 시간 AMVP 후보자 픽처(이하, co-픽처 또는 참조픽처라 함) 내의 움직임 벡터들을 저장하고 있어야 한다. 이에 많은 메모리가 소모되므로, 적절한 방법으로 움직임 벡터들을 저장할 필요가 있다. 이하, 시간 머지 후보자 또는 시간 AMVP 후보자를 통칭하여 시간 후보자라 한다. 또한, 시간 후보자의 위치가 미리 고정된 위치로 설정되어 있었으나, 현재 블록의 LCU 내에서의 위치에 따라 적응적으로 선택할 수도 있다. 또한, 시간 후보자를 구하기 위해 미리 정해진 이하 구체적으로 설명한다.

[0279] **[시간 후보자 움직임 저장 방법]**

[0280] 도 14는 참조 픽처에서 시간 후보자를 저장하기 위해 16개의 4x4로 구성된 16x16 영역을 나타내고 있다. 참조 픽처의 모든 PU의 MV를 저장하고 있을 경우에 메모리가 너무 많이 들기 때문에, 소정 크기의 움직임 벡터 저장 유닛(MVSU : motion vector storing unit) 단위로 움직임 벡터를 저장하는 것이 바람직하다. MVSU는 미리 정해진 하나의 크기, 예를 들어, 8x8 또는 16x16일 수 있다. 또는 부호기가 복호기로 상기 크기를 전송할 수도 있다.

[0281] MVSU가 16x16일 경우, 그 안에 복수개의 PU가 들어 있는 경우, 하나의 MVSU마다 대표되는 하나의 움직임 벡터를 저장하고 있어야 한다. 예를 들어, 도 C에서와 같이, C3 또는 BR 영역 중 어느 하나를 포함하는 예측 유닛의 MV가 MVSU의 MV가 되도록 설정할 필요가 있다. 이 경우, 바람직하게는 BR을 포함하는 예측 유닛의 MV가 MVSU의 MV가 되도록 설정할 수 있다.

[0282] **[시간 후보자 움직임 벡터 유도시의 문제점]**



- [0283] HM3.0에서는 시간 후보자(Temporal motion vector predictor)를 유도하기 위해 미리 저장되어 있는 참조 픽처의 co-located PU의 움직임 벡터(상기한 저장방법으로 저장된 움직임 벡터)를 가져오기 위해, 현재 PU가 포함되어 있는 LCU에 대응하는 참조 픽처의 LCU 영역 내의 움직임 정보 뿐만 아니라, LCU 경계 외의 움직임 벡터 정보도 포함하여야 하는 문제가 있다. 즉, co-located MV를 읽어 들일 영역이 LCU와 align되지 않는다. 구체적으로, co-located MV가 도 15의 H에 존재할 경우, co-located MV를 읽어 들일 영역이 LCU 경계를 넘어설 수 있다. 시간 후보자 움직임 벡터 위치가 H 또는 C0인 경우에는, 도 16의 (a)와 같이, 현재 LCU 내의 PU들의 시간 후보자를 구하기 위해 읽어 들여야 하는 MV가 4개의 LCU에 걸칠 수 있고, H만을 이용할 경우에도 도 16의 (b)와 같이 4개의 LCU 영역을 걸칠 수 있다.
- [0284] 따라서, LCU boundary 밖에 있는 영역의 MV들도 읽어 들여야 하므로, 결국 memory bandwidth를 증가 시키는 결과를 가져오게 된다. 이를 극복하기 위해서, LCU boundary에서는 motion data를 가져올 때, boundary를 넘어가지 않도록, 시간 후보자의 위치를 적응적으로 조정하는 것이 바람직하다.
- [0285] 이하 구체적인 방법을 설명한다.
- [0286] 도 17은 본 발명의 실시예에 따른 시간 후보자를 산출하는 방법을 나타낸 개념도이다.
- [0287] 현재 예측 유닛의 시간 후보자를 유도하기 위한 참조 픽처 내의 현재 예측 유닛의 위치에 대응하는 후보자들을 나타낸다. 현재 예측 유닛의 좌상측에 위치한 화소의 위치를 (xP, yP), 현재 예측 유닛의 폭의 nW, 높이를 nH라고 할 경우, 참조 픽처 내에서의 동일한 위치를 (xP, yP)로 표시할 수 있다.
- [0288] 현재 예측 유닛의 시간 후보자는 BR, H, LH, AH 중에서 선택될 수 있다. BR(상단 좌측 시간적 후보)은 (xP+nW-1, yP+nH-1)에 위치한 픽셀을 포함하는 MVSU(움직임 벡터 저장 유닛), AH는 (xP+nW, yP+nH-1)에 위치한 픽셀을 포함하는 MVSU, LH는 (xP+nW-1, yP+nH)에 위치한 픽셀을 포함하는 MVSU, H는 (xP+nW, yP+nH)에 위치한 픽셀을 포함한 MVSU 이다.
- [0289] 여기서, 현재 예측 유닛의 시간 후보자는 현재 예측 유닛이 LCU의 경계에 접하는지 여부에 따라, BR, H, LH, AH 중에서 적응적으로 선택된다. 구체적으로, 우측 경계에만 접하는지 여부, 아래쪽 경계에만 접하는지 여부, 및 우측 및 아래쪽 경계 모두에 접하는지에 따라 결정될 수 있다. 여기서, 현재 예측 유닛이 LCU 경계에 접하는지 여부는 현재 예측 유닛에 대응하는 참조 픽처 내의 대응 유닛이 LCU 경계에 접하는지 여부와 동일하다.
- [0290] 먼저, 현재 예측 유닛이 LCU의 우측 경계 및 아래쪽 경계 모두에 접하지 않는 경우에는 H의 MV를 시간 후보자로 결정한다. 그러나, H가 존재하지 않는 경우에는 BR 또는 C3를 시간 후보자로 결정할 수도 있다.
- [0291] 현재 예측 유닛이 LCU의 우측 경계에만 접하는 경우에는 H 및 AH가 LCU 경계 밖에 존재하므로, LCU 내부에 속하는 LH의 MV를 시간 후보자로 결정한다.
- [0292] 현재 예측 유닛이 LCU 우측 및 아래쪽 경계에 모두 접하는 경우에는 BR만이 LCU 내부에 존재하기 때문에, BR의 MV를 시간 후보자로 결정한다.
- [0293] 위에서 설명한 바와 같이, 현재 예측 유닛의 시간 후보자를 현재 예측 유닛을 포함하는 LCU에 대응하는 참조 픽처의 LCU 내부에서 선택하도록 함으로써, 복호기의 memory bandwidth를 줄일 수 있다. 즉, H 위치가 현재 LCU 내에 존재하면 이용하고, H 위치가 현재 LCU 내에 존재하지 않으면 다른 위치(BR, AH 또는 LH)의 시간 후보자를 사용할 수 있다.
- [0294] 한편, 상기한 TMVP(H, BR, LH, AH)가 이용 가능하지 않는 경우에는 다른 위치의 MV를 시간 후보자로 설정할 수도 있다. 이 경우, C0 또는 C3가 이용될 수도 있다.
- [0295] 또한, 상기한 시간 후보자(H, BR, LH, AH)가 이용 가능하지 않는 경우에는 현재 예측 유닛의 LCU 내에서의 위치에 따라 시간 후보자로 설정되는 후보자의 위치가 변할 수도 있다.
- [0296] - 현재 예측 유닛이 LCU의 우측 경계 및 아래쪽 경계 모두에 접하지 않고, H의 MV가 이용 가능하지 않은 경우에는 BR의 MV를 시간 후보자로 결정할 수 있다. 또는 AH나 LH의 MV를 시간 후보자로 결정할 수도 있다.
- [0297] - 현재 예측 유닛이 LCU의 우측 경계에만 접하고, LH가 이용 가능하지 않은 경우, BR의 MV를 시간 후보자로 결정할 수 있다.
- [0298] - 현재 예측 유닛이 LCU의 아래쪽 경계에만 접하고, AH가 이용 가능하지 않은 경우, BR의 MV를 시간 후보자로

결정할 수 있다.

- [0299] - 현재 예측 유닛이 LCU의 우측 경계 및 아래쪽 경계 모두에 접하고, BR의 MV가 이용 가능하지 않은 경우에는 시간 후보자는 존재하지 않는 것으로 결정할 수 있다. 또는, C0, 또는 C3의 MV를 시간 후보자로 결정할 수도 있다.
- [0300] - 상기한 4가지의 경우 모두, 시간 후보자는 존재하지 않는 것으로 결정할 수도 있다.
- [0301] 상기 방법은 merge 모드에서 적용할 수도 있다. AMVP 모드에서는 시간 후보자를 선택적으로 이용하거나, 이용하지 않을 수도 있다.
- [0302] Merge 시간 후보자는 이용하되, AMVP 시간 후보자는 없는 것으로 설정할 수도 있다.
- [0303] AMVP에서는 현재 예측 유닛의 LCU 내에서의 위치에 관계없이, C0 또는 BR 위치의 MV를 시간 후보자로 설정할 수도 있다.
- [0304] 다음으로, 머지 후보자 구성에서의 현재 블록의 형태에 따라 정해지는 특정 위치의 공간 머지 후보자를 제외시키는 방법에 대해 설명한다. 이 방법은 상술한 방법 및 장치에 모두 적용될 수 있다. 현재 블록의 공간 머지 후보자의 위치가 도 5의 A, B, C, D, E의 위치에 존재할 경우를 예로 들어 설명한다. 그러나 도 4의 경우에도 동일한 방식이 적용될 수 있다.
- [0305] 현재 예측 블록이 도 18의 (a)와 같이  $kN \times 2N$  ( $k=3/2$ ) 크기의 P1 블록일 경우에는 도 18의 (a)에 도시된 A, B, C, D, E의 위치에 존재하게 된다. 이 경우, A로 표시된 블록은 P0가 된다.
- [0306] 현재 예측 블록이 도 18의 (a)와 같이  $kN \times 2N$  크기의 P1일 경우에는 현재 블록의 움직임 정보가 P0의 움직임 정보와 동일하지 않아야 한다. 동일한 경우에는  $2N \times 2N$  블록을 P0와 P1으로 분리할 필요가 없다.
- [0307] 따라서, 현재 블록이 도 18의 (a)의 P1일 경우에는 P0(즉, A)를 이용 가능하지 않은 머지 후보자로 설정한다. 마찬가지로, P0와 동일한 움직임 정보를 갖는 다른 머지 후보자들도 이용 가능하지 않은 머지 후보자로 설정한다. 이 과정은 도 3, 9의 유효한 공간 머지 후보자를 검색하는 과정, 도 12의 공간 머지 후보자 유도부 등에서 수행될 수 있다. 또는 P0는 상기 과정 또는 상기 유도부에서 수행하고, P0와 동일한 움직임 정보를 갖는 머지 후보자를 제외시키는 과정이 머지 리스트 구축 과정에서 수행할 수도 있다.
- [0308] 마찬가지로, 현재 예측 블록이 도 18의 (b)와 같이  $hN \times 2N$  ( $h=1/2$ ) 크기의 P1 블록일 경우에는 P0(즉, A)를 이용 가능하지 않은 머지 후보자로 설정한다. 마찬가지로, P0와 동일한 움직임 정보를 갖는 다른 머지 후보자들도 이용 가능하지 않은 머지 후보자로 설정한다. 이 과정은 도 3, 9의 유효한 공간 머지 후보자를 검색하는 과정, 도 12의 공간 머지 후보자 유도부 등에서 수행될 수 있다. 또는 P0는 상기 과정 또는 상기 유도부에서 수행하고, P0와 동일한 움직임 정보를 갖는 머지 후보자를 제외시키는 과정이 머지 리스트 구축 과정에서 수행할 수도 있다.
- [0309] 현재 예측 블록이 도 19의 (a)와 같이  $2N \times kN$  크기의 P1일 경우에는 현재 블록의 움직임 정보가 P0(즉, B)의 움직임 정보와 동일하지 않아야 한다. 동일한 경우에는  $2N \times 2N$  블록을 P0와 P1으로 분리할 필요가 없다.
- [0310] 따라서, 현재 블록이 도 19의 (a)의 P1일 경우에는 P0(즉, A)를 이용 가능하지 않은 머지 후보자로 설정한다. 마찬가지로, P0와 동일한 움직임 정보를 갖는 다른 머지 후보자들도 이용 가능하지 않은 머지 후보자로 설정한다. 이 과정은 도 3, 9의 유효한 공간 머지 후보자를 검색하는 과정, 도 12의 공간 머지 후보자 유도부 등에서 수행될 수 있다. 또는 상기 과정 중 P0를 이용 가능하지 않은 것으로 설정하는 과정은 상기 과정 또는 상기 유도부에서 수행하고, P0와 동일한 움직임 정보를 갖는 머지 후보자를 이용 가능하지 않은 것으로 설정하는 과정이 머지 리스트 구축 과정에서 수행할 수도 있다.
- [0311] 마찬가지로, 현재 예측 블록이 도 19의 (b)와 같이  $2N \times hN$  ( $h=1/2$ ) 크기의 P1 블록일 경우에는 P0(즉, B)를 이용

가능하지 않은 머지 후보자로 설정한다. 마찬가지로, P0와 동일한 움직임 정보를 갖는 다른 머지 후보자들도 이용 가능하지 않은 머지 후보자로 설정한다. 이 과정은 도 3, 9의 유효한 공간 머지 후보자를 검색하는 과정, 도 12의 공간 머지 후보자 유도부 등에서 수행될 수 있다. 또는 상기 과정 중 P0를 이용 가능하지 않은 것으로 설정하는 과정은 상기 과정 또는 상기 유도부에서 수행하고, P0와 동일한 움직임 정보를 갖는 머지 후보자를 이용 가능하지 않은 것으로 설정하는 과정이 머지 리스트 구축 과정에서 수행할 수도 있다.

[0312] 현재 예측 블록이  $N \times N$ 이고, 도 20에서의 P3 블록일 경우에 이용 가능하지 않은 머지 후보자 설정에 대하여 설명한다.

[0313]  $2N \times 2N$ 이 4개의  $N \times N$ 으로 split될 경우, P0 블록의 움직임 정보를  $\alpha$ , P1 블록의 움직임 벡터를  $\beta$ , P2의 움직임 벡터를  $\gamma$ 라고 할 때,  $\alpha$ 와  $\beta$ 가 동일하면 A 머지 후보자 (즉, P2 블록 또는  $\gamma$ )을 이용 가능하지 않도록 설정한다. 그리고, 상기  $\gamma$ 와 동일한 움직임 정보를 갖는 머지 후보자들도 이용 가능하지 않은 것으로 설정한다. 이 과정은 도 3, 9의 유효한 공간 머지 후보자를 검색하는 과정, 도 12의 공간 머지 후보자 유도부 등에서 수행될 수 있다. 또는 상기 과정 중 A를 이용 가능하지 않은 것으로 설정하는 과정은 상기 과정 또는 상기 유도부에서 수행하고, A와 동일한 움직임 정보를 갖는 머지 후보자를 이용 가능하지 않은 것으로 설정하는 과정이 머지 리스트 구축 과정에서 수행할 수도 있다.

[0314]  $\alpha$ 와  $\gamma$ 가 동일하면 B 머지 후보자 (즉, P1 블록 또는  $\beta$ )을 이용 가능하지 않도록 설정한다. 그리고, 상기  $\beta$ 와 동일한 움직임 정보를 갖는 머지 후보자들도 이용 가능하지 않은 것으로 설정한다.

[0315] 이 과정은 도 3, 9의 유효한 공간 머지 후보자를 검색하는 과정, 도 12의 공간 머지 후보자 유도부 등에서 수행될 수 있다. 또는 상기 과정 중 B를 이용 가능하지 않은 것으로 설정하는 과정은 상기 과정 또는 상기 유도부에서 수행하고, B와 동일한 움직임 정보를 갖는 머지 후보자를 이용 가능하지 않은 것으로 설정하는 과정이 머지 리스트 구축 과정에서 수행할 수도 있다.

[0316] 다음으로 잔차 블록을 복원하는 과정에 대해 설명한다. 상술한 생성된 예측 블록과 이 과정에서 구한 잔차 블록을 더하여 복원블록을 생성한다. 각 과정은 도 7, 11의 대응하는 장치에서도 동일하게 동작한다.

[0317] 1. 엔트로피 복호화

[0318] 현재 복원하고자 하는 잔여 블록의 계수정보를 엔트로피 복호화한다.

[0319] 2. 역스캐닝

[0320] 상기 엔트로피 복호화된 잔여 블록을 역스캐닝하기 위해 스캔 패턴을 결정한다. 그리고, 결정된 스캔 패턴을 이용하여 2차원 블록을 생성한다. 스캔 패턴은 복원하고자 하는 현재 블록의 부호화 모드에 따라 달라질 수 있다. 인트라 예측 부호화된 경우에는 현재 블록의 인트라 예측 모드, 예측 블록의 크기 등에 따라 적응적으로 스캔 패턴이 결정될 수 있다.

[0321] 인트라 예측 부호화된 경우의 스캔 패턴은 다음의 방법 중 하나를 사용할 수 있다.

[0322] [제1예]

[0323] 현재 변환 블록이 휘도 블록이면, 현재 변환 블록을 포함하는 현재 예측 블록(이하, 현재 블록이라 함)의 크기와 관계없이 지그재그 스캔을 수행할 수 있다. CABAC의 경우에는 상기한 지그재그 스캔을 상기 지그재그 스캔과 다른 diagonal 스캔으로 대체할 수 있다. 즉, 엔트로피 부호화 모드(또는 복호화 모드)에 따라 스캔 패턴이 결정될 수도 있다.

[0324] [제2예]

[0325] 현재 변환블록의 형태에 따라 적응적으로 스캔타입을 결정할 수 있다. 예를 들어, 현재 변환블록이  $2N \times N$  또는  $2N \times hN$ 인 경우에는 수직스캔을,  $N \times 2N$  또는  $hN \times 2N$ 인 경우에는 수평스캔을 수행할 수 있다. 정사각형인 경우에는

지그재그 스캔을 수행할 수도 있다. 이 경우, 변환 블록의 크기가 소정 크기 이상인 경우에는 지그재그 스캔만을 수행할 수도 있다.

[0326] 또는 블록의 크기가 소정 크기보다 큰 경우에는 서브블록 단위로 스캔을 수행할 수 있다. 이 경우, last non-zero 변환 계수의 위치를 이용하여 스캔해야 할 서브블록들을 알 수 있게 된다. 또한, 각 서브블록들의 경우, 변환계수를 갖지 않는 경우에는 이를 나타내는 정보(1bit의 플래그)가 비트스트림으로 전송될 수 있다. 따라서, 상기 정보들을 이용하여 서브블록을 역스캔하여 변환 블록을 생성할 수 있다.

[0327] [예 1]

[0328] 블록의 크기가 미리 정해진 크기(예를 들어, 4x4 또는 8x8)보다 큰 경우에는 상기 블록을 복수개의 NxN의 서브블록으로 나누어 스캔할 수 있다. 이 경우, 변환 블록의 형태에 따라 결정되는 스캔패턴을 상기 서브블록 내의 변환 계수를 스캔할 때 사용할 수 있다. 서브블록들 간의 스캔 순서는 상기 결정되는 스캔패턴 또는 지그재그 스캔순서로 할 수 있다. 또는 래스터 스캔 순서로 할 수도 있다.

[0329] [예 2]

[0330] 블록의 크기가 미리 정해진 크기보다 큰 경우에는 상기 변환 블록을 미리 정해진 개수(예를 들어, 4)의 서브블록으로 나누어 스캔할 수 있다. 이 경우, 변환 블록의 형태에 따라 결정되는 스캔패턴을 상기 서브블록 내의 변환 계수를 스캔할 때 사용할 수 있다. 서브블록들 간의 스캔 순서는 상기 결정되는 스캔패턴 또는 지그재그 스캔순서로 할 수 있다. 또는 래스터 스캔 순서로 할 수도 있다.

[0331] 색차 블록의 경우에는 하나의 스캔패턴(예를 들어, 지그재그 스캔)만을 적용할 수 있다. 그러나, 상기한 휘도 블록과 동일하게 변환 블록의 형태에 따라 미리 정해진 스캔 패턴을 적용할 수도 있다.

[0332] 한편, 상기한 내용들은 변환 계수(또는 변환 계수의 크기)를 스캔하는 방식에 관한 것이다. 그러나, CABAC으로 엔트로피 부호화를 수행할 경우에는 Significant Map, Level, signs을 부호화하여야 한다. 이때, level과 마찬가지로 Significant Map도 스캔이 되어 부호화되어야 하고, 복호기에서는 역스캔되어 2차원의 significant map을 구성하여야 한다. 다음의 방법들이 존재할 수 있다.

[0333] [예1]

[0334] Sig. Map과 level coding에 동일한 스캔패턴을 적용할 수 있다. 또한, 스캔 방향도 동일하게 할 수 있다. 상기 스캔 패턴은 상기한 "현재 변환 블록이 인터 예측인 경우" 기술된 내용과 동일한 방식을 수행할 수 있다. 또는 미리 정해진 크기의 변환 블록들에 대해서만 스캔 패턴을 변화시킬 수도 있다.

[0335] Sig.Map은 블록의 크기와 관계없이 last significant coefficient로부터 DC로 스캔 패턴에 따라 스캔을 수행할 수 있다. 그러나, level의 경우에도 동일한 방식으로 스캔을 수행하되, 복수개의 서브셋으로 나누어 스캐닝할 수 있다. 상기 복수개의 서브셋은 정사각형의 형태일 수도 있지만, 스캔 패턴에 따라 결정되는 미리 정해진 level 개수(예를 들어, 16개)의 집합일 수도 있다. 예를 들어, 지그재그 스캔시의 미리 정해진 개수의 계수들이 하나의 서브셋을 구성할 수도 있다.

[0336] 이 경우, 상기 서브셋의 위치에 따라 서로 다른 컨텍스트 set들이 결정될 수 있다.

[0337] 종합하면, 스캔 패턴은 다음의 방법들의 다양한 조합에 의해 결정될 수 있다.

[0338] - 현재 블록 또는 서브 블록의 예측 모드에 따라 스캔 패턴이 달라진다. 인트라인 경우에는 3가지 스캔 방식 중 하나를, 인터인 경우에는 1가지 스캔 방식만 사용할 수도 있다.

[0339] - 현재 블록 또는 서브 블록의 예측 모드가 인트라인 경우, 현재 블록 또는 서브 블록의 인트라 예측 모드 및 변환블록(현재 블록 또는 서브블록일 수 있음)의 크기에 따라 스캔 패턴이 달라질 수 있다.

[0340] - 현재 블록 또는 서브 블록의 예측 모드가 인터인 경우, 변환블록(현재 블록 또는 서브블록일 수 있음)의 모양 및/또는 변환블록의 크기에 따라 스캔 패턴이 달라질 수 있다.

[0341] - 현재 블록 또는 서브 블록의 변환 계수의 엔트로피 부호화 방식에 따라 스캔패턴이 달라질 수 있다.

- [0342] 한편, 현재 예측 유닛의 형태에 따라 스캐닝 패턴이 달리 결정될 수도 있다.
- [0343] 예측 유닛의 형태에 따라 scanning pattern이 결정될 수 있다. 예를 들어, 인터 예측이고, 예측 유닛이 2NxN이면 vertical scan을 사용하고, Nx2N이면, horizontal scan을 사용할 수 있다. 그리고, 예측 유닛이 정사각형이면 지그재그 스캔을 사용할 수 있다.
- [0344] AMP에서도 동일한 방식을 적용할 수 있다. h=0.5이다.
- [0345] - 즉, 2NxhN의 예측 유닛에는 vertical scan을 사용하고, hNx2N의 경우에는 horizontal scan을 사용할 수 있다.
- [0346] - 2Nx3N/2에 대해서는 vertical과 지그재그 스캔 중 하나를 사용할 수 있다. 3N/2x2N의 경우에는 horizontal과 지그재그 스캔 중 하나를 사용할 수 있다. 또는 스캔타입을 시그널링 해줄 수도 있다.
- [0347] **3. 역양자화**
- [0348] 상기 2차원의 잔여계수 변환 블록을 역양자화한다. 구체적인 방법은 다음과 같다.
- [0349] **[QU 결정]**
- [0350] 현재 변환블록에 적용된 양자화 파라미터를 구하기 위해 QP를 복원하기 위한 최소 단위(QU)를 결정한다.
- [0351] QP는 수신된 cu\_qp\_delta\_enabled\_flag를 이용하여 구할 수 있다. cu\_qp\_delta\_enabled\_flag는 QP<sub>Y</sub>를 LCU 단위로 변경할지 또는 그 하위의 CU에서도 추가적으로 변경이 가능하도록 할지를 정의하고 있다. cu\_qp\_delta\_enabled\_flag의 값이 1(즉, LCU보다 작은 CU에서의 QP<sub>Y</sub>는 각변경을 허용)하는 경우에는, PPS(picture parameter set)에서 **max\_cu\_qp\_delta\_depth**를 통해 QP<sub>Y</sub> 변경을 허용하는 최소 크기의 CU size(이를 LCU를 기준으로 maximum allowed depth 형태로 표현)를 정의하고 있다. QP<sub>Y</sub> 변경을 허용하는 최소 크기의 CU size는 다음 수식에 의해 결정된다.
- [0352]  $\log_2\text{MinCUDQPSize} = \log_2\text{MaxCUSize} - \text{max\_cu\_qp\_delta\_depth}$
- [0353] 여기서, log2MinCUDQPSize가 최소 크기의 CU size를 나타낸다.
- [0354] 따라서, cu\_qp\_delta\_enabled\_flag와 **max\_cu\_qp\_delta\_depth**를 통해 상기 QP가 변경될 수 있는 최소 크기의 CU size(이하, QU:quantization unit이라 함)를 구한다.
- [0355] 한편, cu\_qp\_delta\_enabled\_flag와 **max\_cu\_qp\_delta\_depth**를 하나의 변수로 설정할 수도 있다. 예를 들어, cu\_qp\_delta\_enabled\_info를 기존의 cu\_qp\_delta\_enabled\_flag와 max\_cu\_delta\_depth를 합친 정보로 수신할 수 있다. 즉, cu\_qp\_delta\_enabled\_info의 값을 복원하여, 상기 값이 0인 경우에는 QP 고정, 1인 경우에는 LCU 단위의 QP 변경(즉, QU=LCU), 2인 경우에는 QU=(LCU/2)x(LCU/2), 3인 경우에는 QU=(LCU/4)x(LCU/4) 등으로 설정할 수도 있다.
- [0356] 상기한 경우, cu\_qp\_delta\_enabled\_info를 SPS에서만 정의하는 것이 아니라 PPS마다 on/off 하도록 정의하여 더 flexible하게 QP를 조정할 수도 있다.
- [0357] **[dQP 복원]**
- [0358] QU 단위로 dQP를 복원한다.
- [0359] <슬라이스마다 QP range가 정해지는 경우>
- [0360] 슬라이스마다 MaxQP와 MinQP 또는 이를 나타내는 정보(예를 들어, QP range)가 정의될 수 있다. 상기 QP range는 slice QP값을 기준으로 대칭적으로 허용되는 값으로 전송하는 것이 효과적임. 예를 들어 전송하는 파라미터는 (MaxQP-sliceQP)값일 수 있음. 이 경우, 상기 파라미터가 2이고, slice QP가 25라면 허용되는 QP range는 23~27로 설정될 수 있다.
- [0361] QP range가 정해지면, 허용가능한 dQP 값을 알 수 있다. 따라서, 엔트로피 복호화된 dQP 인덱스를 통해 dQP 값



을 구할 수 있다.

[0362] 예를 들어, dQP indexing은 다음과 같을 수 있다. dQP는 현재 QP와 예측 QP와의 차이값을 임. QP range가 전송 되면, 가능한 dQP 값은 MinQP-predQP to MaxQP-predQP로 제한되므로, dQP의 인덱스를 기존 방법과 달리 설정할 수 있음. 예를 들어, MaxQP=27, MinQP=23 and predQP=24라면 인덱싱은 다음의 표 1과 같이 될 수 있다.

표 1

Index	dQP value
0	0
1	1
2	-1
3	2
4	3

[0364] 허용 가능한 QP range가 전송되면, 허용 가능한 인덱스도 정해지게 되므로, CABAC의 경우, dQP index를 부호화 하기 위한 binarization을 truncated unary binarization을 이용하는 것이 효과적이다.

[0365] <QP range를 전송하지 않거나 QP range가 클 경우>

[0366] dQP를 기존의 unary binarization을 이용하는 것이 아니라, 절대값과 부호를 구분할 수 있도록 binarization을 하여 부호화된 것일 수 있다. 예를 들어, dQP의 절대값을 unary binarization을 하고, 여기에 부호를 나타내는 bin을 마지막 위치 또는 두번째 위치에 위치시킬 수도 있다.

[0367] 한편, 부호화기에서 dQP가 0인지 아닌지를 플래그를 통해 알려줄 수도 있다.

[0368] 따라서, 이 경우, 상기 dQP를 엔트로피 복호화 후 얻어진 binarization 데이터에 대응하는 QP의 크기값과 부호를 추출하여 dQP를 복원할 수 있다.

[0369] <dQP binarization>

[0370] HM 3.0에서는 unary binarization을 이용하여 이진화하였다. 그러나, 이 경우, 비트수가 많이 소모되는 문제가 있으므로, dQP의 절대값 또는 dQP-1의 절대값을 unary binarization하고, 여기에 dQP의 부호를 추가하여 bin string을 구성할 수 있다. 이 경우, 추가되는 dQP의 부호는 dQP의 절대값의 unary binarization된 bin string의 맨 마지막에 위치할 수 있다. 또는 dQP의 절대값 또는 dQP-1의 절대값을 truncated unary binarization하고, 여기에 dQP의 부호를 추가하여 bin string을 구성할 수 있다. 이 경우에는 복호기의 복잡도 감소를 위해 dQP의 절대값의 unary binarization된 bin string의 두번째 bin에 위치하도록 할 수도 있다.

[0371] 따라서, 엔트로피 복호화된 bin string으로부터 상기한 binarization 방식에 대응하는 인덱스를 복원하여 상기 인덱스에 대응하는 dQP를 복원할 수 있다.

[0372] [QP 예측자 결정]

[0373] QU 단위로 QP 예측자를 결정한다. QP 예측자는 다음의 방법 중 하나로 결정될 수 있다.

[0374] <실시예 1>

[0375] 왼쪽의 QU가 존재하면, 왼쪽의 QU의 QP(즉, 왼쪽 QU의 가장 오른쪽?)가 QP 예측자가 된다. 그러나, 왼쪽의 QU가 존재하지 않는 경우에는 부호화/복호화 순서상 바로 이전의 QU의 QP가 QP 예측자로 정의된다.

[0376] <실시예 2>

[0377] 현재 CU가 QU보다 작고, QU 내에 선행하는 CU가 존재하는 경우에는 QP 예측자를 구할 필요가 없다. 이 경우, 선행하는 CU의 또는 현재 CU가 포함된 QU의 양자QP를 현재 CU의 QP로 설정하여 이용한다. 그러나, 현재 CU가 QU보다 작지 않거나, QU내의 좌상측 CU이면 다음의 과정을 통해 QP 예측자를 유도한다.

- [0378] - 현재 CU의 cu\_qp\_delta 및 현재 cu의 좌상측 블록의 예측 모드가 입력된다. Cu\_qp\_delta는 cu syntax를 통해 수신한다.
- [0379] - 현재 CU의 예측 모드와 동일한 예측 모드를 갖는 좌측 CU(또는 QU) 및 상측 CU(또는 QU)의 QP들 중 적어도 하나를 이용하여 QP 예측자를 유도한다.
- [0380] A. 현재 CU가 인터 예측인 경우에는 좌측 CU 및 상측 CU가 모두 인터 예측이면 좌측 CU 및 상측 CU의 QP들의 평균값이 QP 예측자가 된다.
- [0381] 
$$QP \text{ 예측자} = ( QP_{Y,ABOVE} + QP_{Y,LEFT} + 1 ) / 2$$
- [0382] 또는 좌측 CU의 QP를 QP 예측자로 결정할 수 있다.
- [0383] B. 현재 CU가 인터 예측이고, 좌측 CU 및 상측 CU 중 하나만이 인터 예측이면 인터 예측인 CU의 QP를 QP 예측자로 결정한다.
- [0384] F. 현재 CU가 인터 예측이고, 좌측 CU 및 상측 CU 모두 인트라 예측이면, 부호화/복호화 순서상 바로 이전의 CU의 QP를 QP 예측자로 정의할 수 있다.
- [0385] H. 여기에서, 좌측 또는 상측 CU의 잔차신호가 0이 아닌 경우에 상기 방법을 적용하고, 잔차 신호가 0인 경우에는 그 CU를 non available로 하여 처리할 수 있다.
- [0386] <실시예 3>
- [0387] 현재 CU가 QU보다 작고, QU 내에 선행하는 CU가 존재하는 경우에는 QP 예측자를 구할 필요가 없음. 그러나, 현재 CU가 QU보다 작지 않거나, QU내의 좌상측 CU이면 다음의 과정을 통해 QP 예측자를 유도한다.
- [0388] - 현재 CU의 cu\_qp\_delta 및 현재 cu의 좌상측 블록의 예측 모드가 입력된다. Cu\_qp\_delta는 cu syntax를 통해 수신한다.
- [0389] - 예측 모드가 인터인 경우에는 다음의 과정을 수행한다.
- [0390] A. 현재 CU의 상측 QU의 현재 CU와 인접하는 상측 CU의 예측모드가 현재 CU의 예측 모드와 동일하고, 현재 CU의 좌측 QU의 현재 CU와 인접하는 좌측 CU의 예측 모드가 현재 CU와 동일한 경우에는 다음과 같다.
- [0391] 
$$QP \text{ 예측자} = ( ( QP_{Y,ABOVE} + QP_{Y,LEFT} + 1 ) / 2$$
- [0392] B. 현재 CU의 상측 CU의 예측 모드 및 좌측 CU의 예측 모드들 중 하나만이 현재 CU의 예측 모드와 동일한 경우에는 동일한 CU의 QP를 QP 예측자로 정의한다.
- [0393] C. 현재 CU의 상측 CU 및 좌측 CU 중 하나만이 존재할 경우에는 존재하는 CU의 QP를 QP 예측자로 정의 또는 모드가 다를 경우 오프셋(2 또는 3) 값을 더해준다.
- [0394] D. 모두 non available할 경우에는 부호화/복호화 순서상 이전에 부호화/복호화된 CU의 QP를 이용하여 QP 예측자를 결정한다.
- [0395] <실시예 4>
- [0396] QP가 CU 내에서 변화도록 설정할 경우, 즉 PU 또는 TU 단위로도 적응적으로 설정할 수 있도록 할 경우에는, 상기한 CU들의 QP가 상기 CU들에 대응하는 PU 또는 TU의 QP로 변경된다. 이때 상측 QP는 상측 CU의 좌측에 위치하는 PU 또는 TU의 QP, 좌측 QP는 좌측 CU의 상측에 위치하는 PU 또는 TU의 QP를 의미할 수 있다.
- [0397] 예를 들어, QP가 TU 단위로 변경되도록 설정할 경우, 소정 크기 이하의 TU에 대해서는 동일한 QP가 적용되도록 하는 파라미터를 부호기가 설정하여 복호기로 전송한다. 그리고, 상기 파라미터에 기초하여 소정 크기보다 작은 TU의 QP는 동일하게 설정하는 것은 앞서 CU의 경우와 동일하게 진행한다.
- [0398] [QP 복원]

- [0399] dQP와 QP 예측자를 이용하여 QP를 구한다. 즉, 현재 CU의  $QP=dQP+QP$  예측자가 된다.
- [0400] **[역양자화 수행]**
- [0401] 다음으로, QP를 이용하여 현재의 변환 블록의 역양자화를 수행한다.
- [0402] 역양자화를 위해 상기 QP를 이용하여 역양자화 매트릭스를 구하여 역양자화를 수행한다. 양자화 매트릭스는 예측 모드, 변환블록의 크기에 따라 미리 정해진 것일 수 있다.
- [0403] 그러나, SPS 또는 PPS마다 양자화 매트릭스를 변형시켜 부호화효율을 증대시킬 수 있다. 이 경우에는 양자화 매트릭스가 부호화되어 전송될 수 있다. 따라서, 복호기는 전송된 양자화 매트릭스 정보를 복원한 후에, 역양자화를 수행할 수도 있다.
- [0404] 구체적으로, SPS 또는 PPS에서 양자화 매트릭스를 부호화하여 전송할 수도 있고, 별도의 NAL unit을 통해 양자화 매트릭스를 전송할 수도 있다.
- [0405] 양자화 매트릭스는 부호화되어 전송되는 경우, 부호화 순서는 다음과 같다.
- [0406] 1) 먼저 현재 양자화 매트릭스를 입력 받는다.
- [0407] 2) 다음으로, 현재 양자화 매트릭스의 예측 매트릭스를 이용하여 잔차 신호를 생성한다.
- [0408] 3) 다음으로, 잔차 신호를 부호화한다.
- [0409] 양자화 매트릭스의 예측 방식은 다음과 같다.
- [0410] - 이전에 부호화된 참조 양자화 매트릭스 중 하나를 예측값으로 사용할 수 있다. 이 경우, 잔차 신호는 현재 양자화 매트릭스와 이전 양자화 매트릭스의 차이값으로 결정된다. 만약 이전에 부호화된 참조 양자화 매트릭스의 크기가 현재의 양자화 매트릭스 크기와 다를 경우에는 미리 정해진 방식으로 다운 샘플링 또는 업샘플링을 한다. 업샘플링의 경우 선형 보간을 사용할 수 있고, 다운 샘플링의 경우에는 4:1의 경우에는 4개의 평균값을, 16:1의 경우에는 16:1의 평균값을 사용하거나, 미리 정해진 2개 또는 4개의 값의 평균값을 이용할 수도 있다.
- [0411] - 현재 양자화 매트릭스의 성분을 DPCM을 사용하여 잔차신호를 생성할 수도 있다. 이 경우에는 양자화 매트릭스의 계수들은 지그재그 스캔순서로 DPCM이 적용된다. 그리고, 양자화가 적용될 수 있다
- [0412] - 한편, 상기 ①의 잔차 신호 또는 ②의 DPCM 신호를 양자화하여 신호를 압축하는 방법을 이용할 수도 있다. 이 경우, 양자화 매트릭스의 계수들 중 저주파 성분보다 고주파 성분쪽의 양자화를 크게 하는 것이 바람직하다. 따라서, 양자화 매트릭스를 지그재그 순서상으로 복수개의 영역으로 나누고, 각 영역마다 서로 다른 양자화 파라미터로 양자화하는 것이 바람직하다. 즉, DC를 포함하는 N번째 계수(지그재그 스캔순서상)까지는 양자화를 하지 않고, N+1~M번째 계수까지는 제1 양자화 파라미터(예를 들어 2), M+1~L번째 계수까지는 제2 양자화 파라미터를 이용하는 식으로 잔차신호 또는 DPCM신호를 양자화하여 전송하는 것이 바람직하다.
- [0413] 복호기에서는 상기 과정의 역과정을 진행하면 된다.
- [0414] 양자화된 또는 양자화되지 않은 잔차신호 또는 DPCM 신호를 CAVLC 또는 CABAC를 이용하여 부호화할 수 있다.
- [0415] CAVLC를 이용할 경우에는, 상기 잔차신호 또는 DPCM 신호를 지그재그 스캔 순서로 런과 레벨의 조합을 이용하여 부호화할 수 있다. 즉, 기존의 H.264 방식 또는 현재 HEVC 4.0에서 제정되어 있는 방식을 이용하여 부호화할 수 있다.
- [0416] 한편, 색차 QP는 Luma QP로부터 구한다. 다만 luma QP가 소정값 이내인 경우에는 luma QP값을 chroma QP값으로 설정하고, 소정값 이상인 경우에는 별도의 mapping 을 이용할 수도 있다. 이 경우, 픽처 또는 슬라이스 헤더에서 chroma QP와 luma QP 사이의 맵핑 관계를 알려주는 정보(chroma\_qp\_index\_offset)를 시그널링 해주는 것이 좋다. 영상의 특성에 따라 복수개의 맵핑 테이블이 존재할 수도 있다. Chroma\_qp\_index\_offset은 PPS로 보내 줘도 된다.
- [0417] **4. 역변환**

- [0418] **[변환 복호화 과정]**
- [0419] 1. 먼저, 역양자된 잔차 블록을 수신한다.
- [0420] 2. 그리고, 수신된 잔차블록이 현재 예측 블록이 인트라 예측 잔차 블록인지 인터 예측 잔차 블록인지를 판단한다.
- [0421] 3. 현재 예측 블록의 예측 모드가 인트라 예측이면, 잔차 블록의 크기에 관계없이 각 잔차 블록의 크기에 대응하는 역변환을 수행할 수 있다. 상기 역변환은 DCT 기반 역변환 또는 KLT 기반 역변환일 수 있다.
- [0422] 이 경우, 상기 예측 블록의 형태가 정사각형이면, 정사각형 역변환을 사용하여 변환한다.
- [0423] 상기 예측 블록의 형태가  $2N \times N$ ,  $N \times 2N$ 이면, 1) 잔차 블록이 2개 이상의 정사각형 블록으로 나누어져 정사각형 역변환이 수행되거나, 2) non-square transform이 수행될 수도 있다. 2)의 경우, 예를 들어,  $2N \times N$  역변환은  $N \times N$  역변환과  $2N \times 2N$  역변환을 사용하여 역변환을 할 수도 있다.
- [0424] 상기 예측 블록의 형태가  $2N \times 0.5N$ ,  $2N \times 1.5N$ 으로 나누어지는 경우에는, 변환블록의 크기는  $2N \times 2N$ , 4개의  $2N \times 0.5N$ 으로 나누어질 수 있고(이 경우, split transform flag가 사용됨), 상기  $2N \times 0.5N$ 은 4개의  $N \times 0.25N$ 으로 나누어지거나 그대로 변환될 수 있다(이 경우에는 split transform flag가 사용됨). 따라서, 상기 split transform flag를 이용하여, 상기 크기에 대응하는 역변환을 수행할 수 있다.
- [0425] 상기한 경우의  $2N \times 0.5N$  역 변환은  $2N \times 2N$  역변환과  $0.5N \times 0.5N$  역변환을 이용하여 구현될 수 있다.
- [0426] 5. 현재 예측 블록의 예측 모드가 인트라 예측이고, 상기 예측 블록의 변환 블록의 크기가 제1 크기보다 큰 경우에는 다음의 2가지 방법 중 하나를 적용할 수 있다.
- [0427] <방법 1>
- [0428] 상기 인트라 예측 모드에 따라 적응적으로, 상기 수신된 역양자화 블록의 변환 계수들 중 제1 서브셋을 제2 역변환기법을 통해 역변환한다. 상기 제2 역변환기법은 상기한 “현재 예측 블록의 예측 모드가 인트라 예측이고, 상기 예측 블록의 변환 블록의 크기가 제1 크기보다 작거나 같은 경우”에 적용되는 역변환 방법과 동일할 수 있다.
- [0429] 그리고, 상기 제2 역변환된 블록 또는 상기 수신된 역양자화 블록 전체에 제1 역변환 기법을 적용한다. 상기 제1 역변환 기법은 KLT 기반 정수 역변환 또는 DCT 기반 정수 역변환일 수 있다.
- [0430] 상기 제1 서브셋은 DC를 포함하는 소정 계수의 포함한다. 상기 서브셋은 순방향 지그재그 스캔순서로 DC를 포함하는 소정 계수(예를 들어, 16개)를 포함할 수도 있고, DC를 포함하는  $4 \times 4$  또는  $8 \times 8$  블록일 수도 있다. 상기 서브셋의 크기는 변환 블록의 크기에 따라 변할 수도 있다.
- [0431] <방법 2>
- [0432] “현재 예측 블록의 예측 모드가 인트라 예측이고, 상기 예측 블록의 변환 블록의 크기가 제1 크기보다 작거나 같은 경우”에 적용되는 역변환 방법과 동일할 수 있다.
- [0433] <방법 3>
- [0434] 인트라 예측 모드에 관계없이, 변환 블록 전체에 제1 역변환 기법만을 적용할 수 있다. 상기 제1 역변환 기법은 KLT 기반 정수 역변환 또는 DCT 기반 정수 역변환일 수 있다.
- [0435] 한편, 상기한 split transform flag가 수신된 비트스트림 내에 존재하지 않으면, 현재 변환 블록의 크기가 허용된 변환 블록의 최대 크기(MaxTrafoSize)보다 크면, split\_tranform\_flag의 값은 1로 설정하고, 상기 split transform flag를 이용하여 변환 블록을 역변환할 수 있다.

- [0436] [색차 블록의 변환 복호화 과정]
- [0437] 1. 먼저, 역양자된 잔차 블록을 수신한다.
- [0438] 2. 그리고, 수신된 잔차블록이 현재 예측 블록이 인트라 예측 잔차 블록인지 인터 예측 잔차 블록인지를 판단한다.
- [0439] 3. 현재 예측 블록의 예측 모드가 인터 예측이면, 잔차 블록의 크기에 관계없이 각 잔차 블록의 크기에 대응하는 역변환을 수행한다. 상기 역변환은 DCT 기반 역변환 또는 KLT 기반 역변환일 수 있다.
- [0440] 이 경우, 상기 예측 블록의 형태가 정사각형이면, 정사각형 역변환을 사용하여 변환한다.
- [0441] 상기 예측 블록의 형태가  $2N \times N$ ,  $N \times 2N$ 이면, 1) 잔차 블록이 2개 이상의 정사각형 블록으로 나누어져 정사각형 역변환이 수행되거나, 2) non-square transform이 수행될 수도 있다. 2)의 경우, 예를 들어,  $2N \times N$  역변환은  $N \times N$  역변환과  $2N \times 2N$  역변환을 사용하여 역변환을 할 수도 있다.
- [0442] 상기 예측 블록의 형태가  $2N \times 0.5N$ ,  $2N \times 1.5N$ 으로 나누어지는 경우에는, 변환블록의 크기는  $2N \times 2N$ , 4개의  $2N \times 0.5N$ 으로 나누어질 수 있고(이 경우, split transform flag가 사용됨), 상기  $2N \times 0.5N$ 은 4개의  $N \times 0.25N$ 으로 나누어지거나 그대로 변환될 수 있다(이 경우에는 split transform flag가 사용됨). 따라서, 상기 split transform flag를 이용하여, 상기 크기에 대응하는 역변환을 수행할 수 있다.
- [0443] 상기한 경우의  $2N \times 0.5N$  역 변환은  $2N \times 2N$  역변환과  $0.5N \times 0.5N$  역변환을 이용하여 구현될 수 있다.
- [0444] 한편, 상기한 split transform flag가 수신된 비트스트림 내에 존재하지 않으면, 현재 변환 블록의 크기가 허용된 변환 블록의 최대 크기(MaxTrafoSize)보다 크면, split\_tranform\_flag의 값은 1로 설정하고, 상기 split transform flag를 이용하여 변환 블록을 역변환할 수 있다.
- [0445] 한편, 새로운 움직임 정보 부호화 모드에 대해서 설명한다.
- [0446] Merge mode에서는 전송하여야 할 움직임 정보로서 머지 플래그와 머지 인덱스만이 존재한다.
- [0447] 반면에, AMVP mode에서는 현재 블록의 인터 예측 모드가 단방향 예측이면, 전송하여야 할 움직임 정보는 참조픽처 인덱스, AMVP 후보자 인덱스, 차분 움직임 벡터(MVD)를 포함한다. 그리고, 현재 블록의 인터 예측 모드가 양방향 예측이면, 전송하여야 할 움직임 정보는 리스트0, 1 각각에 대한 참조픽처 인덱스, AMVP 후보자 인덱스, 차분 움직임 벡터(MVD)를 포함한다.
- [0448] 그러나, 후보자와 움직임 벡터만이 조금 차이가 나는 움직임 정보가 현재 블록의 최적 움직임 정보일 경우에는 머지가 되지 못하고, AMVP 방식으로 부호화가 되어야 하기 때문에 전송효율이 떨어지게 된다.
- [0449] 따라서, 머지 후보자와 참조 픽처 인덱스는 동일하고, 움직임 벡터만이 조금 차이가 나는 움직임 정보가 현재 블록의 최적의 움직임 정보이면, 상기 머지 모드 인덱스와 차분 움직임 정보를 전송하는 것이 더욱 효과적이다.
- [0450] 이러한 모드를 머지 MVD 모드로 새롭게 도입할 필요가 있다. 이하에서는 머지 MVD 방법 및 그 방법에서의 머지 MVD 부호화 및 복호화 동작에 대해 설명한다. 상기 방법은 머지 MVD 예측자의 인터 예측 모드에 따라 달라질 수 있다. 머지 MVD 후보자는 머지 후보자와 동일할 수 있다. 또는 다를 수도 있다. 머지 MVD 예측자는 머지 MVD 후보자들로부터 구해질 수 있다.
- [0451] [머지 MVD 예측자의 인터 예측 모드가 단방향 예측 모드인 경우]
- [0452] 머지 MVD 예측자의 인터 예측 모드가 단방향 예측 모드인 경우에는 다음의 어느 하나의 방식을 이용한다.
- [0453] <실시예 1>
- [0454] 현재 블록의 움직임 정보도 상기 머지 MVD 예측자와 동일한 단방향 움직임 정보를 갖는다. 전송해야 할 머지 MVD 인덱스는 머지 MVD 후보자들 중 어느 하나를 나타낸다. 전송해야 할 차분 움직임 벡터는 상기 머지 MVD 예측자와 현재 블록의 움직임 벡터의 차이값을 나타낸다.
- [0455] 따라서, 복호기에서 복원하는 현재 블록의 움직임 정보는 단방향 움직임 정보로서, 참조픽처 인덱스는 상기 머지 MVD 인덱스가 나타내는 머지 MVD 예측자의 움직임 정보가 나타내는 참조픽처 인덱스가 되고, 움직임 벡터는



상기 상기 머지 MVD 인덱스가 나타내는 머지 MVD 예측자의 움직임 벡터에 상기 차분 움직임 벡터를 더한 값이 된다.

[0456] <실시예 2>

[0457] 현재 블록의 움직임 정보는 양방향 움직임 정보를 갖는다. 전송해야 할 머지 MVD 인덱스는 머지 MVD 후보자들 중 어느 하나를 나타낸다. 전송해야 할 차분 움직임 벡터는 상기 머지 MVD 예측자와 현재 블록의 움직임 벡터의 차이값을 나타낸다.

[0458] 현재 블록의 다른 방향의 움직임 정보는 상기 머지 MVD 예측자의 참조 픽처 리스트와 다른 리스트의 참조픽처 인덱스 0을 갖는 픽처(또는 동일 거리만큼 떨어져 있는 픽처)이고, 움직임 정보는 상기 머지 MVD 예측자와 방향이 반대인 움직임 벡터가 된다.

[0459] 따라서, 복호기에서 복원하는 현재 블록의 움직임 정보 중 상기 머지 MVD 후보자와 동일 방향 움직임 정보의 참조픽처 인덱스는 상기 머지 MVD 인덱스가 나타내는 머지 MVD 예측자의 움직임 정보가 나타내는 참조픽처 인덱스가 되고, 움직임 벡터는 상기 머지 MVD 인덱스가 나타내는 머지 MVD 예측자의 움직임 벡터에 상기 차분 움직임 벡터를 더한 값이 된다.

[0460] 그리고, 현재 블록의 움직임 정보 중 상기 머지 MVD 후보자와 다른 방향 움직임 정보의 참조픽처 인덱스는 0(또는 동일 거리만큼 떨어져 있는 픽처)가 되고, 움직임 벡터는 상기 머지 MVD 인덱스가 나타내는 머지 MVD 예측자와 방향이 반대인 움직임 벡터(거리가 다를 경우, 스케일링 된 것이어야 함)가 된다.

[0461] 스케일링을 해야 할 경우에는, 복잡도 감소를 위해 머지 MVD를 허용하지 않을 수도 있다.

[0462] <실시예 3>

[0463] 현재 블록의 움직임 정보는 양방향 움직임 정보를 갖는다. 전송해야 할 머지 MVD 인덱스는 머지 MVD 후보자들 중 어느 하나를 나타낸다. 전송해야 할 차분 움직임 벡터는 상기 머지 MVD 예측자와 다른 방향의 생성된 움직임 벡터와 현재 블록의 움직임 벡터의 차이값을 나타낸다.

[0464] 현재 블록의 다른 방향의 움직임 정보는 상기 머지 MVD 예측자의 참조 픽처 리스트와 다른 리스트의 참조픽처 인덱스 0을 갖는 픽처(또는 동일 거리만큼 떨어져 있는 픽처)이고, 움직임 정보는 상기 머지 MVD 예측자와 방향이 반대인 움직임 벡터가 된다.

[0465] 따라서, 복호기에서 복원하는 현재 블록의 움직임 정보 중 상기 머지 MVD 후보자와 동일 방향 움직임 정보의 참조픽처 인덱스는 상기 머지 MVD 인덱스가 나타내는 머지 MVD 예측자의 움직임 정보가 나타내는 참조픽처 인덱스가 되고, 움직임 벡터는 상기 머지 MVD 인덱스가 나타내는 머지 MVD 예측자의 움직임 벡터가 될 수 있다.

[0466] 그리고, 현재 블록의 움직임 정보 중 상기 머지 MVD 후보자와 다른 방향 움직임 정보의 참조픽처 인덱스는 0(또는 동일 거리만큼 떨어져 있는 픽처)가 되고, 움직임 벡터는 상기 머지 MVD 인덱스가 나타내는 머지 MVD 예측자와 방향이 반대인 움직임 벡터(거리가 다를 경우, 스케일링 된 것이어야 함)에 MVD를 더한 값이 된다.

[0467] 스케일링을 해야 할 경우에는, 복잡도 감소를 위해 머지 MVD를 허용하지 않을 수도 있다.

[0468] <복호화 순서>

[0469] - 현재 블록의 부호화 모드가 머지 MVD 모드인지를 판단하여, 머지 MVD 모드인 경우에 적용한다.

[0470] - 머지 MVD 인덱스를 복호화한다.

[0471] - 머지 MVD 후보자 리스트를 구축한다. 이 방법은 머지 후보자 리스트 구축과 동일할 수 있다. 즉, 공간 후보자, 시간 후보자도 머지 후보자와 동일하게 구할 수 있다. 추가도 가능..

[0472] - 머지 MVD 인덱스가 나타내는 움직임 정보를 구한다.

[0473] - MVD를 복호하여 구한다.

[0474] - MVD를 더할 참조 픽처를 결정하고(실시예 1의 경우에는 필요없음), 상기 MVD를 결정된 참조 픽처를 나타내는

움직임 벡터에 더한다.

[0475] [머지 MVD 예측자의 인터 예측 모드가 양방향 예측 모드인 경우]

[0476] 머지 MVD 예측자의 인터 예측 모드가 양방향 예측 모드인 경우에는 다음의 어느 하나의 방식을 이용한다.

[0477] 현재 블록의 움직임 정보는 양방향 움직임 정보를 갖는다. 전송해야 할 머지 MVD 인덱스는 머지 MVD 후보자들 중 어느 하나를 나타낸다. 현재 블록의 L0, L1 참조 픽처 인덱스는 머지 MVD 예측자의 참조 픽처 인덱스들과 동일하다.

[0478] <전송할 차분 움직임 벡터가 1개인 경우>

[0479] 이 경우에는 L0 방향 차분 움직임 벡터인지, L1 방향 차분 움직임 벡터인지가 정해져야 한다. 따라서, 부호기에서 이를 나타내는 정보를 보낼 수 있다. 또 다른 방법으로는 차분 움직임 벡터를 L0, L1 방향 중 하나로 고정할 수 있다. 또는 L0 방향 참조 픽처와 현재 픽처와의 거리와 L1 방향 참조 픽처와 현재 픽처와의 거리를 비교하여 거리가 큰 쪽의 차분 움직임 벡터가 되도록 할 수 있다. 이 경우, 양쪽 거리가 같을 경우, L1 방향 차분 움직임 벡터가 되도록 설정할 수 있다.

[0480] <전송할 차분 움직임 벡터가 2개인 경우>

[0481] 이 경우에는 전송해야 할 차분 움직임 벡터가 2개인 경우도 가능하다. 이 경우에는 L0, L1순으로 차분 움직임 벡터를 전송한다.

[0482] <복호화 순서>

[0483] - 현재 블록의 복호화 모드가 머지 MVD 모드인지를 판단하여, 머지 MVD 모드인 경우에 적용한다.

[0484] - 머지 MVD 인덱스를 복호화한다.

[0485] - 머지 MVD 후보자 리스트를 구축한다. 이 방법은 머지 후보자 리스트 구축과 동일할 수 있다. 즉, 공간 후보자, 시간 후보자도 머지 후보자와 동일하게 구할 수 있다. 추가도 가능..

[0486] - 머지 MVD 인덱스가 나타내는 움직임 정보를 구한다.

[0487] - MVD를 복호하여 구한다.

[0488] - MVD를 더할 참조 픽처를 결정하고, 상기 MVD를 결정된 참조 픽처를 나타내는 움직임 벡터에 더한다.

[0489] 상기 머지 MVD 방법은 슬라이스 타입이 B인 경우에만 적용할 수도 있다.

[0490] 슬라이스 타입이 P인 경우에는 상술한 머지 MVD 예측자의 인터 예측 모드가 단방향 예측 모드인 경우의 실시예 1의 방법만을 적용한다.

[0491] 도 21은 본 발명의 다른 실시예에 따른 인터 예측 복호화 장치(300)를 나타내는 구성도이다.

[0492] 본 발명에 따른 인터 예측 복호화 장치(400)는 디멀티플렉서(410), 움직임 정보 부호화 모드 판단부(420), 머지 모드 움직임 정보 복호화부(430), AMVP 모드 움직임 정보 복호화부(440), 예측블록 생성부(450), 잔차 블록 복호화부(460), 복원블록 생성부(470) 및 머지 MVD 모드 움직임 정보 복호화부(480)를 포함한다.

[0493] 디멀티플렉서(410), 움직임 정보 부호화 모드 판단부(420), 머지 모드 움직임 정보 복호화부(430), AMVP 모드 움직임 정보 복호화부(440), 예측블록 생성부(450), 잔차 블록 복호화부(460), 복원블록 생성부(470)의 동작은 도 11의 디멀티플렉서(310), 움직임 정보 부호화 모드 판단부(320), 머지 모드 움직임 정보 복호화부(330), AMVP 모드 움직임 정보 복호화부(340), 예측블록 생성부(350), 잔차 블록 복호화부(360) 및 복원블록 생성부(370)의 동작과 동일하다.

[0494] 머지 MVD 모드 움직임 정보 복호화부(480)는 움직임 정보 부호화 모드 판단부(420)가 현재 블록의 움직임 정보 부호화 모드를 머지 MVD모드로 판단한 경우에 활성화된다. 머지 MVD 인덱스와 MVD를 이용하여 움직임 정보를 복

원한다.

- [0495] 도 22는 본 발명의 다른 실시예에 따른 머지 MVD 모드 움직임 정보 복호화부(480)를 나타내는 블록도이다.
- [0496] 본 발명의 일 실시예에 따른 머지 MVD 모드 움직임 정보 복호화부(480)는 머지 MVD 예측자 인덱스 복호화부(481), 공간 머지 MVD 후보자 유도부(482), 시간 머지 MVD 후보자의 참조 픽처 인덱스 유도부(483), 시간 머지 MVD 후보자의 움직임 벡터 유도부(484), 시간 머지 MVD 후보자 구성부(485), 머지 MVD 리스트 구축부(486), 머지 MVD 후보자 생성부(487), 머지 MVD 예측자 선택부(488), 움직임 정보 생성부(489) 및 잔차 움직임 벡터 복호화부(490)를 포함한다.
- [0497] 머지 MVD 예측자 인덱스 복호화부(481), 공간 머지 MVD 후보자 유도부(482), 시간 머지 MVD 후보자의 참조 픽처 인덱스 유도부(483), 시간 머지 MVD 후보자의 움직임 벡터 유도부(484), 시간 머지 MVD 후보자 구성부(485), 머지 MVD 리스트 구축부(486), 머지 MVD 후보자 생성부(487), 머지 MVD 예측자 선택부(488), 움직임 정보 생성부(489)의 동작은 도 12의 머지 예측자 인덱스 복호화부(331), 공간 머지 후보자 유도부(332), 시간 머지 후보자의 참조 픽처 인덱스 유도부(333), 시간 머지 후보자의 움직임 벡터 유도부(334), 시간 머지 후보자 구성부(335), 머지 리스트 구축부(336), 머지 후보자 생성부(337), 머지 예측자 선택부(338) 및 움직임 정보 생성부(339)의 동작과 동일하므로 생략한다.
- [0498] 다만, 움직임 정보 생성부(489)는 잔차 움직임 벡터 복호화부(490)로부터 수신된 MVD와 머지 MVD 예측자가 나타내는 움직임 정보를 이용하여 현재 블록의 움직임 정보를 생성한다.
- [0499] 잔차 움직임 벡터 복호화부(490)는 현재 블록의 차분 움직임 벡터를 복호한다. 차분 움직임 벡터가 CAVLC로 부호화된 경우에는 복호한 후, unsigned integer 곱셈 코드에 대응하는 값을 이용해 복원한다.
- [0500] 움직임 정보를 이용하여 예측 블록 생성하는 과정에서, 움직임 벡터가 integer이면 참조 픽처로부터 예측 블록을 복사하고, 움직임 벡터가 non-integer이면, 예측블록 생성을 위해 참조 픽처 내의 화소들을 이용하여 예측 블록의 화소들을 보간한다. 여기서는 예측 블록의 화소들을 보간하는 또 다른 방법에 대해 설명한다.
- [0501] **[휘도 화소 보간 방법]**
- [0502] 복호된 움직임 정보 중에서 참조 픽처 인덱스가 나타내는 픽처를 선택한다.
- [0503] 선택된 픽처 내에서의 움직임 벡터가 나타내는 블록을 생성하기 위해 화소를 보간한다. 상기 움직임 벡터는 quarter-pel 단위를 움직임 벡터이다.
- [0504] 도 23을 참조하여 설명한다.
- [0505] 생성하여야 할 화소의 위치가 a, b, c에 위치하는 경우에는 수평방향의 가장 가까운 위치에 존재하는 필터 탭수 만큼의 정수 화소들에 필터를 적용하여, a, b, c를 생성한다.
- [0506] 생성하여야 할 화소의 위치가 d, h, n에 위치하는 경우에는 수직방향의 가장 가까운 위치에 존재하는 정수 화소들에 필터를 적용하여 d, h, n을 생성한다.
- [0507] 생성하여야 할 화소의 위치가 e, i, p에 위치하는 경우에는 수직 방향으로 가장 가까운 위치에 존재하는 a 위치의 화소들을 상기한 방법으로 구한 후에, 수직방향으로 가장 가까운 위치에 존재하는 소정 개수의 a 위치의 quarter-pel 화소들에 필터를 적용하여 예측 화소를 생성한다.
- [0508] 생성하여야 할 화소의 위치가 g, k, r에 위치하는 경우에는 수직 방향으로 가장 가까운 위치에 존재하는 c 위치의 화소들을 상기한 방법으로 구한 후에, 수직방향으로 가장 가까운 위치에 존재하는 소정 개수의 c 위치의 quarter-pel 화소들에 필터를 적용하여 예측 화소를 생성한다.
- [0509] 생성하여야 할 화소의 위치가 f, j, q에 위치하는 경우에는 수직 방향으로 가장 가까운 위치에 존재하는 b 위치의 화소들을 상기한 방법으로 구한 후에, 수직방향으로 가장 가까운 위치에 존재하는 소정 개수의 b 위치의 half-pel 화소들에 필터를 적용하여 예측 화소를 생성한다.
- [0510] 한편, e, f, g, i, j, k, p, q, r을 구할때, a, b, c 위치의 화소들을 이용하여 구했으나, d, h, n 위치의 화

소들을 이용하여 구할 수도 있다. 이 경우, e, f, g 위치의 화소는 d 위치의 quarter-pel 화소들을 이용하고, l, j, k 위치의 화소는 h 위치의 half-pel 화소들을 이용하고, p, q, r 위치의 화소들은 n 위치의 quarter-pel 화소들을 이용하여 구할 수 있다.

[0511] 그리고, 상기 생성에 적용되는 필터는 각 방향에서의 sub-pel 위치에 따라 달라질 수 있다.

[0512] 표 2은 HM 3.0에서의 필터를 나타낸다.

표 2

[0513]

화소 위치	필터 계수
H	{-1, 4, -11 40, 40, -11, 4, -1}
F1	{-1, 4, -10 57, 19, -7, 3, -1}
F2	{-1, 3, -7 19, 57, -10, 4, -1}

[0514] Half-pel 위치의 H에 대해서는 8-tap symmetry filter를 사용하고, quarter-pel 위치의 F1, F2에 대해서는 8-tap non symmetry filter를 사용한다. a, h는 H의 위치에 대응된다. a, d는 F1의 위치에 대응된다. c, n은 F2의 위치에 대응된다. e, f, g는 F1에, i, j, k는 H에, p, q, r은 F2의 위치에 대응된다. 따라서, a의 경우, L3~L0 위치의 정수화소들과, R0~R3 위치의 정수화소들에 표 1의 필터를 적용하여 예측 화소를 생성한다. 마찬가지로, h의 경우에는 수직방향으로 동일하게 상기 필터를 적용한다. e, f, g, i, j, k, p, q, r의 경우에는 수직 방향으로 가장 가까운 sub-pel 위치의 화소들을 상기 필터를 사용하여 생성한 후, 수직방향으로 상기 생성된 화소들을 이용하여 동일한 방식으로 예측 화소를 생성한다

[0515] 상기한 바와 같이, HM 3.0에서는 생성하여야 할 위치에 따라서만 필터 계수가 변하게 된다. 상기한 8-tap 필터는 효과적으로 화소를 예측할 수 있으나, 탭수가 크기 때문에 부호화기 및 복호화기에서의 연산량이 크다는 문제점이 있다. 따라서, 상기 연산량을 줄이면서, 예측 화소를 효과적으로 보간할 수 있는 새로운 필터를 제안한다.

[0516] <실시에 1>

[0517] 예측 하고자 하는 화소의 위치뿐만 아니라, 현재 블록의 예측 방법이 uni-prediction인지 bi-prediction인지에 따라서 서로 다른 필터를 사용하는 것이 예측 블록을 생성하는데 효과적일 수 있다.

[0518] (예 1)

[0519] 표 3에서와 같이, 예측 모드가 단방향 예측인지, 양방향 예측인지에 따라 서로 탭수를 갖는 필터를 사용할 수 있다. 또한, 화소의 위치에 따라 서로 다른 탭수를 갖는 필터를 적용할 수 있다. 상기 예측 모드 및 화소의 위치에 따라 서로 다른 탭수를 갖는 필터를 적용할 수도 있다. 표 2에서는 단방향 예측시 half-pel 화소에 대해서는 6 탭, quarter-pel 화소에 대해서는 5 탭을 사용하고 있다. 그리고, 양방향 예측시에는 HM 3.0에서와 동일하게 8 탭을 사용하고 있다.

표 3

[0520]

예측 모드	화소 위치	탭 수	필터 계수
단방향	H	6	{2, -8, 36, 36, -8, 2}
	FL	5	{-3 51 20, -7, 2}
	FR	5	{2 -7 20 51 -3}
양방향	H	8	{-1, 4, -11 40, 40, -11, 4, -1}
	FL	8	{-1, 4, -10 57, 19, -7, 3, -1}
	FR	8	{-1, 3, -7 19, 57, -10, 4, -1}

- [0521] (예 2)
- [0522] 표 2와 달리, 양방향 예측에서는 2종류의 예측 블록을 생성하여야 하므로, 양방향에서의 필터 탭수를 작게 설정하는 것이 복잡도를 줄이는데 훨씬 효과적이다. 또한, 양방향 예측의 경우에는 비교적 예측이 용이하므로, 탭수를 줄여도 잔차신호에는 큰 변화가 없을 수 있다.
- [0523] 따라서, 양방향에서의 필터 탭수를 단방향에서의 필터 탭수보다 작게 설정한다. 예를 들어, 단방향에서의 탭수는 8로 하고, 양방향에서의 탭수는 6 또는 7로 설정할 수 있다.
- [0524] 또는, 단방향에서의 탭수를 화소 위치에 따라 달리 설정할 수 있다. 예를 들어, half-pel 위치의 탭수는 8로 고정하되, quarter-pel 위치의 탭수는 7 또는 6으로 설정할 수 있다. 이 경우, 양방향에서의 필터 탭수는 quarter-pel 위치의 탭수와 동일하거나 작게 설정할 수 있다.
- [0525] <실시예 2>
- [0526] 복수개의 필터 세트를 구비하여, 복수개의 필터 세트 중 하나를 선택하도록 할 수 있다. 상기 선택 방법은 슬라이스 단위 또는 LCU 단위로 할 수도 있다. 필터 세트를 특정하는 필터 세트 인덱스가 복호기로 슬라이스 또는 LCU 단위로 전송될 수 있다. 상기 각각의 필터 세트는 화소 위치에 따라 적용되어야 하는 필터들을 구비한다.
- [0527] 상기 실시예 1의 방법과 결합하여 적용될 수 있다.
- [0528] <실시예 3>
- [0529] 예측 블록의 크기에 따라 서로 다른 필터 탭을 사용할 수 있다. 4x4 예측 블록을 생성하기 위해 8탭을 쓸 경우에는 11x11의 데이터를 로딩해야 하고, 16x16 예측 블록을 생성하기 위해 8탭을 쓸 경우에는 23x23의 데이터를 로딩해야 한다. 따라서, 예측 블록의 크기가 작아질수록 실제 예측해야 할 블록의 크기보다 더 많은 비율의 데이터를 로딩하게 된다.
- [0530] 따라서, 예측 블록의 소정 크기보다 작은 경우에는 작은 탭수(예를 들어, 4 또는 6)의 필터를 사용하고, 예측 블록의 크기가 소정 크기보다 크거나 같은 경우에는 큰 탭수(예를 들어, 6 또는 8)의 필터를 사용할 수 있다. 소정 크기는 8x8인 것이 바람직하다.
- [0531] 상기한 예측 블록에 따라 다른 탭수의 필터를 적용하는 방법은 양방향 예측에서만 적용할 수 있다. 이는, 단방향 예측에서는 예측 성능을 높이기 위해 탭수를 유지하고, 양방향 예측에서는 탭수를 줄여도 성능 저하가 적기 때문이다.
- [0532] <실시예 4>
- [0533] 예측 블록과 참조 픽처 내의 대응 블록의 관계를 고려하여 탭수를 조정할 수도 있다. 먼저, 예측 블록과 참조 픽처 내의 대응 블록의 양자화 파라미터를 비교하여 탭수를 조절할 수도 있다. 즉, 참조 픽처 내의 대응 블록의 양자화 파라미터가 현재 예측 블록의 양자화 파라미터보다 소정 크기 만큼 더 큰 경우에는 작은 탭수의 필터를 사용하고, 그렇지 않은 경우에는 큰 탭수의 필터를 사용할 수 있다. 상기 소정 크기는 0 또는 음수일 수도 있다.
- [0534] 또는 현재 블록을 포함하는 슬라이스 타입과 예측 블록을 포함하는 슬랑이스 타입을 비교하여 탭수를 조정할 수도 있다. 예를 들어, 슬라이스 타입이 같은 경우에는 작은 탭수를 적용하고, 슬라이스 타입이 다를 경우에는 큰 탭수를 적용할 수도 있다. 이 경우, 양방향 예측에서만 상기 방법을 적용할 수도 있다.
- [0535] <실시예 5>
- [0536] 단방향 예측과 양방향 예측에 사용되는 필터 탭수를 동일하게 유지하되, 동일한 화소 위치에 서로 다른 필터 계수를 갖는 필터를 적용할 수 있다. 단방향 예측과 양방향 예측 각각에 보다 적합한 필터들을 따로 설정하여 적용할 수 있다.



- [0537] 한편, 상기한 방법들에 의해 생성된 예측 블록은 현재 예측 블록에 인접한 복원된 화소들을 이용하여 필터링 될 수도 있다. 예를 들어, 예측 블록들 중 좌측 및 상측 경계에 위치하는 화소는 상기 화소에 접하는 복원 화소를 이용하여 필터링 할 수 있다. 예측 블록의 좌상측 화소는 좌측과 상측에서 접하므로 2개의 복원 화소를 이용하여 필터링 할 수 있다. 1개의 복원 화소를 이용하여 필터링할 경우의 필터는 [1, 3]을 적용하고, 2개의 복원 화소를 이용하여 필터링할 경우에는 [1, 2, 1] 필터를 적용할 수 있다.
- [0538] 이 경우, 변환 부호화에 적용되는 방법은 상기한 필터링 여부에 따라 다른 변환 방법을 사용할 수 있다. 예를 들어, 필터링이 적용되지 않는 경우에는 horizontal DCT + vertical DCT를 적용하고, 필터링이 적용되는 경우에는 다른 변환 방법, 예를 들어, horizontal DST + vertical DST 등을 적용할 수도 있다.
- [0539] [회도 화소 보간 방법]
- [0540] 도 24는 4x4 블록의 예측 블록을 생성하기 위해 로딩해야 하는 화소들을 x축 상으로만 표현한 것이다. 도 24에 도시된 바와 같이, 8-tap 보간필터를 사용할 경우, 4x4 블록의 예측 화소를 생성하기 위해 참조 픽처내의 11x11의 화소들을 로딩해야 하므로 메모리 대역폭이 많이 소모되게 된다.
- [0541] 따라서, 메모리 대역폭을 줄이기 위해서 도 25에서와 같이 로딩되는 데이터량을 줄일 수 있다. 도 25는 4x4 블록의 예측 블록을 생성하기 위해 로딩되어야 하는 참조 픽처 내의 화소수를 7x7로 줄인 경우를 예시하고 있다.
- [0542] 이 경우, 예측 화소 a를 생성하기 위한 종전의 참조 화소 2개(L2, L3)를 이용할 수 없게 된다. 따라서, 이 경우에는 다음의 방식들 중 하나를 사용할 수 있다.
- [0543] <예 1>
- [0544] 도 25에 도시된 바와 같이, 좌측의 L1, L1 화소와 우측의 R0-R3 화소를 이용하여 예측 화소 a를 생성할 수 있다. 이 경우 6탭 필터를 사용하게 된다. 또한, 보다 예측을 잘 하기 위해 R4를 포함하여 7탭 필터를 이용해 예측 화소 a를 생성할 수 있다. 즉, 예측화소를 중심으로 양방향으로 동일하지 않은 참조 화소를 이용하여 예측 화소를 생성할 수 있다.
- [0545] 마찬가지로, 예측화소 b의 경우에도 예측 화소의 좌측 참조 화소를 더 많이 이용하여 예측 화소를 생성할 수 있다. 6탭 또는 7탭 필터를 사용하여 예측 화소를 생성할 수 있다.
- [0546] 예측화소 c의 경우에는 우측의 경우에는 기존의 참조 화소들을 이용하고, 좌측의 경우에는 3개의 참조 화소를 이용하여 예측 화소를 생성할 수 있다.
- [0547] 도 26은 폭이 8 이상인 예측 블록의 경우에서의 예측 화소를 생성하기 위해 이용되는 참조화소들을 나타낸다. 도 26에 도시된 바와 같이 예측화소 a는 좌측의 2개의 참조 화소 및 우측의 4개 또는 5개의 참조화소들을 이용하여 생성된다. 즉, 6탭 또는 7탭 필터를 사용한다. 예측화소 b는 좌측의 3개의 참조 화소 및 우측의 4개 참조 화소들을 이용하여 생성된다. 즉, 7탭 필터를 사용한다. 예측화소 c는 좌측의 4개의 참조 화소 및 우측의 4개의 참조화소들을 이용하여 생성된다. 즉, 8탭 필터를 사용한다
- [0548] 상기한 바와 같이, 예측 화소를 생성하기 위해 사용되는 참조 화소의 수 및 위치가 예측화소의 위치에 따라 달라질 수 있다. 즉, 블록 경계면에 인접한 참조 화소들의 경우에는 블록 내부의 예측 화소 생성을 위한 필터와 다른 필터를 사용하여 예측 화소를 생성할 수 있다. 또한 필터의 탭수가 달라질 수도 있다. 또한, 상기한 바와 같이, 이용되는 양방향의 참조화소 수가 예측 화소의 위치에 따라 달라질 수 있다.
- [0549] 도 25, 26에 도시된 바와 같이, 로딩되지 않은 L2, L3를 외삽하여 생성하여 종전의 방법을 사용할 수도 있다. 단, 이 경우의 필터탭수는 동일하지만 계수는 달라질 수 있다.
- [0550] 한편, 이하에서는 AMVP 움직임 벡터 부호화/복호화 방법의 또 다른 실시예에 대하여 설명한다.
- [0551] 현재 HEVC 3.0에서는 현재 블록이 양방향 예측을 할 경우에 부호화해야 하는 움직임 정보는 List0의 참조 픽처 인덱스 및 움직임 벡터, List1의 참조 픽처 인덱스 및 움직임 벡터를 포함한다.
- [0552] 상기 정보를 부호화하기 위해 각 List별로 움직임 벡터 예측자를 결정하고, 상기 움직임 벡터 예측자를 이용하

여 각 list별로 차분 움직임 벡터를 구한다. 그리고, 각 리스트별로 참조픽처 인덱스와 차분 움직임 벡터를 전송하게 된다.

[0553] 각 리스트별 차분 움직임 벡터를 구하기 위해 공간 및 시간 AMVP 후보자를 구하여 AMVP 후보자 리스트를 구성하는 과정 등은 상술한 바와 같다.

[0554] 그러나, 상기와 같이 움직임 벡터 예측자를 구할 경우, 각 리스트마다 움직임 벡터 예측자를 구해야 하므로, 복잡도가 증가하게 되어 부호화기 및 복호화기에 많은 부하를 유발시킨다.

[0555] 따라서, 새로운 AMVP 부호화 방식이 도입될 필요가 있다. 이하에서는 새로운 AMVP 부호화 방식에 대해 설명한다. 특히, 양방향 예측모드에서의 움직임 벡터 부호화 방식에 대하여 설명한다.

[0556] **[인터 예측 모드가 단방향 예측 모드인 경우]**

[0557] AMVP 인터 예측 모드가 단방향 예측 모드인 경우에는 상술한 방법들을 사용하여 부호화/복호화할 수 있다.

[0558] **[인터 예측 모드가 단방향 예측 모드인 경우]**

[0559] AMVP 인터 예측 모드가 양방향 예측 모드인 경우에 대해 설명한다. 즉, AMVP 후보자 리스트 구성에 대해 설명한다. 나머지 부분은 상술한 방법들과 동일하다.

[0560] <실시예 1>

[0561] 먼저, list0에 대한 AMVP 후보자 리스트를 구성한다. 이는 상술한 방법들과 동일한 방식으로 구한다. 상기한 후보자 리스트로부터 list0의 움직임 벡터 예측자를 선택한다.

[0562] 다음으로, List1에 대한 AMVP 후보자 리스트는 list0의 후보자 리스트를 이용한다.

[0563] 예를 들어, 상기 list0의 AMVP 후보자 리스트 중 현재 블록의 list1 참조픽처와 동일한 시간적 거리를 갖는 후보자 list1의 움직임 벡터 예측자로 결정할 수 있다.

[0564] 또한, 상기 list0의 AMVP 후보자들 중 하나를 선택하여 list1의 움직임 벡터 예측자로 결정할 수도 있다. 이 경우, 현재 블록의 list1 참조픽처와 동일한 시간적 거리를 갖지 않는 list0의 AMVP 후보자가 후보자 list1의 움직임 벡터 예측자로 선택된 경우에는 상기한 시간적 거리를 이용하여 스케일링을 수행한다.

[0565] 이 경우의 복호기에서의 동작은, 현재 블록의 list0의 참조픽처 인덱스, 차분움직임 벡터, 및 AMVP 인덱스 및 list0의 참조픽처 인덱스, 차분움직임 벡터, 및 AMVP 인덱스를 복원한다. 복원하는 방법은 상술한 방법과 동일하다.

[0566] 다음으로, list0의 AMVP 후보자 리스트를 구축하여, 상기 list0의 AMVP 인덱스가 나타내는 후보자를 예측자로 결정하고, 상기 AMVP 예측자의 움직임 벡터와 복원된 차분 움직임 벡터를 더하여 현재 블록의 list0 움직임 벡터를 복원한다.

[0567] 다음으로, list1의 AMVP 예측자로, list0의 AMVP 후보자 리스트를 이용한다. 즉, 상기 list1의 AMVP 인덱스가 나타내는 list0의 AMVP 후보자를 List1의 AMVP 예측자로 결정하고, 상기 list1의 AMVP 예측자의 움직임 벡터와 복원된 list1의 차분 움직임 벡터를 더하여 list1의 움직임 벡터를 복원한다.

[0568] <실시예 2>

[0569] 먼저, list0에 대한 AMVP 후보자 리스트를 구성한다. 이는 상술한 방법들과 동일한 방식으로 구한다. 상기한 후보자 리스트로부터 list0의 움직임 벡터 예측자를 선택한다.

[0570] List1의 AMVP 예측자로 상기 list0의 움직임이 벡터 예측자를 이용한다.

[0571] 이 경우, 현재 블록의 list1 참조픽처와 동일한 시간적 거리를 갖지 않는 list0의 AMVP 후보자가 후보자 list1의 움직임 벡터 예측자로 선택된 경우에는 상기한 시간적 거리를 이용하여 스케일링을 수행할 수 있다. 또는 이 경우에, list1에 대한 AMVP 리스트를 별도로 작성하여 List1의 AMVP 예측자를 결정할 수도 있다.

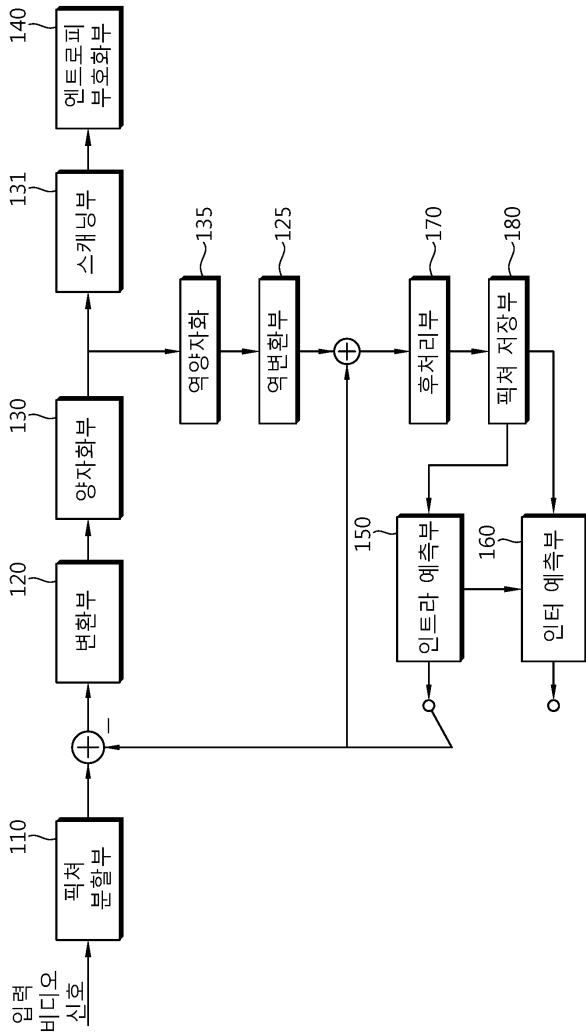
- [0572] 이 경우의 복호기에서의 동작은, 현재 블록의 list0의 참조픽처 인덱스, 차분움직임 벡터, 및 AMVP 인덱스 및 list0의 참조픽처 인덱스, 차분움직임 벡터, 및 AMVP 인덱스를 복원한다. 복원하는 방법은 상술한 방법과 동일하다.
- [0573] 다음으로, list0의 AMVP 후보자 리스트를 구축하여, 상기 list0의 AMVP 인덱스가 나타내는 후보자를 예측자로 결정하고, 상기 AMVP 예측자의 움직임 벡터와 복원된 차분 움직임 벡터를 더하여 현재 블록의 list0 움직임 벡터를 복원한다.
- [0574] 다음으로, list1의 AMVP 예측자로, list0의 AMVP 예측자를 이용하는 경우, 상기 list0의 AMVP 예측자를 이용하여 list 1의 AMVP 예측자를 생성한다. 이 경우, 현재 블록의 list1 참조픽처와 동일한 시간적 거리를 갖지 않는 list0의 AMVP 후보자가 후보자 list1의 움직임 벡터 예측자로 선택된 경우에는 상기한 시간적 거리를 이용하여 스케일링을 수행할 수 있다.
- [0575] <실시예 3>
- [0576] 먼저, list0에 대한 AMVP 후보자 리스트를 구성한다. 이는 상술한 방법들과 동일한 방식으로 구한다. 상기한 후보자 리스트로부터 list0의 움직임 벡터 예측자를 선택한다.
- [0577] 다음으로, List1에 대한 AMVP 후보자 리스트에는 list0의 AMVP 예측자 또는 이를 이용하여 생성되는 움직임 벡터 포함된다. 상기 움직임 벡터는 list1 방향의 참조 픽처 인덱스를 갖는 움직임 벡터로서 list0의 AMVP 예측자와 동일 방향일수도 있고, 반대 방향일 수도 있다. 상기 list0의 AMVP 예측자를 list1 AMVP 후보자들보다 낮은 인덱스를 갖는 후보자로 설정한다. 그러나, 상기 list0의 AMVP 예측자의 참조 픽처와 현재 픽처와의 시간적 거리가 현재 블록의 list1 참조픽처와 동일한 시간적 거리를 갖는 경우에 한해, list1 후보자로 추가할 수도 있다.
- [0578] 이 경우의 복호기에서의 동작은, 현재 블록의 list0의 참조픽처 인덱스, 차분움직임 벡터, 및 AMVP 인덱스 및 list0의 참조픽처 인덱스, 차분움직임 벡터, 및 AMVP 인덱스를 복원한다. 복원하는 방법은 상술한 방법과 동일하다.
- [0579] 다음으로, list0의 AMVP 후보자 리스트를 구축하여, 상기 list0의 AMVP 인덱스가 나타내는 후보자를 예측자로 결정하고, 상기 AMVP 예측자의 움직임 벡터와 복원된 차분 움직임 벡터를 더하여 현재 블록의 list0 움직임 벡터를 복원한다.
- [0580] 다음으로, list1의 복호화된 AMVP 예측자로 특정 인덱스(예를 들어, 가장 낮은 인덱스)를 갖는 경우, 즉, list0의 AMVP 예측자 또는 이를 이용하여 생성된 움직임 벡터가 list1의 움직임 벡터가 된다. 따라서, 이 경우에는 list1의 AMVP 후보자 리스트를 구축할 필요가 없다. 그러나, list1의 복호화된 AMVP 예측자로 특정 인덱스(예를 들어, 가장 낮은 인덱스)를 갖지 않는 경우에는 list1에 들어갈 AMVP 후보자를 생성하여야 한다. 생성 방법은 상술한 방법과 동일할 수 있다.
- [0581] 상기에서는 본 발명의 바람직한 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

**부호의 설명**

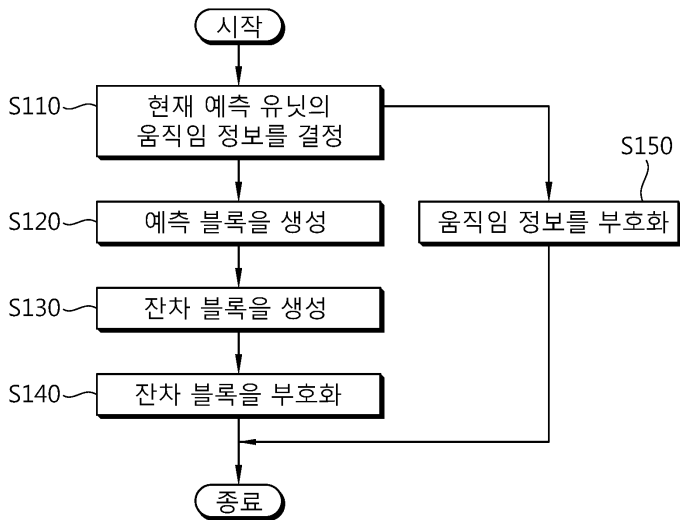
- [0582] 260 : 인터 예측 복호화부

도면

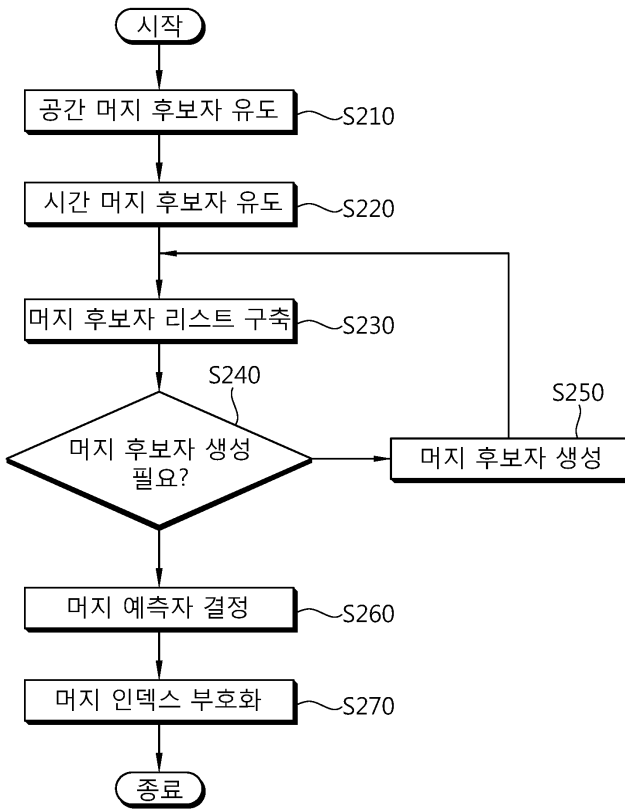
도면1



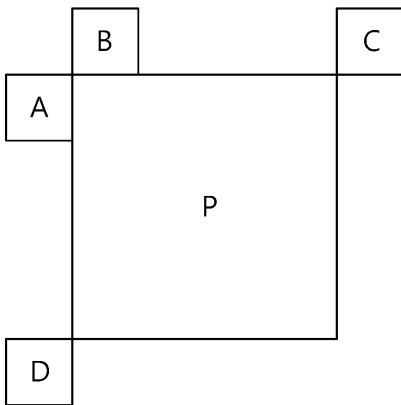
도면2



도면3

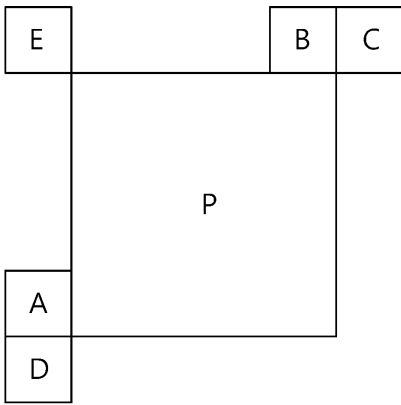


도면4

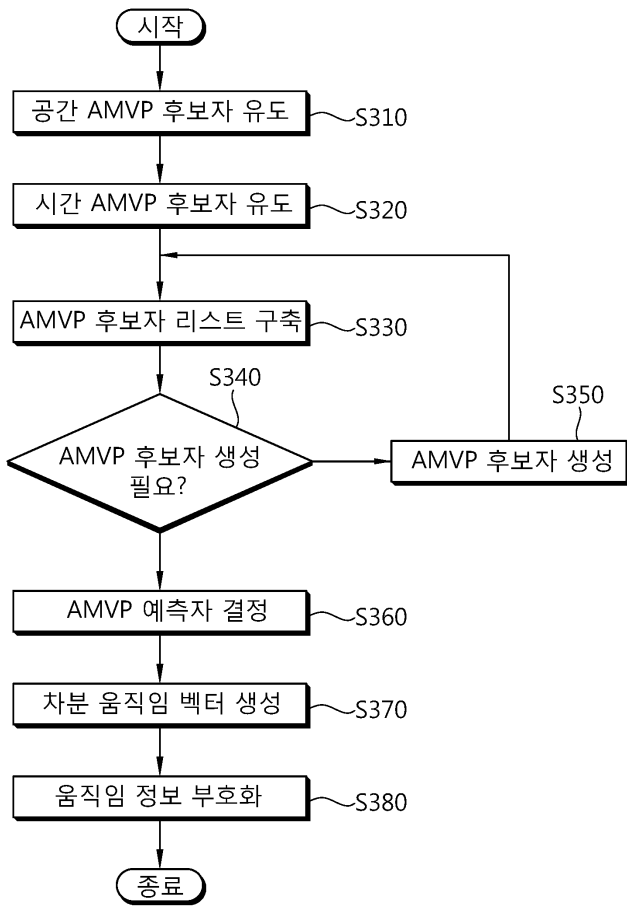




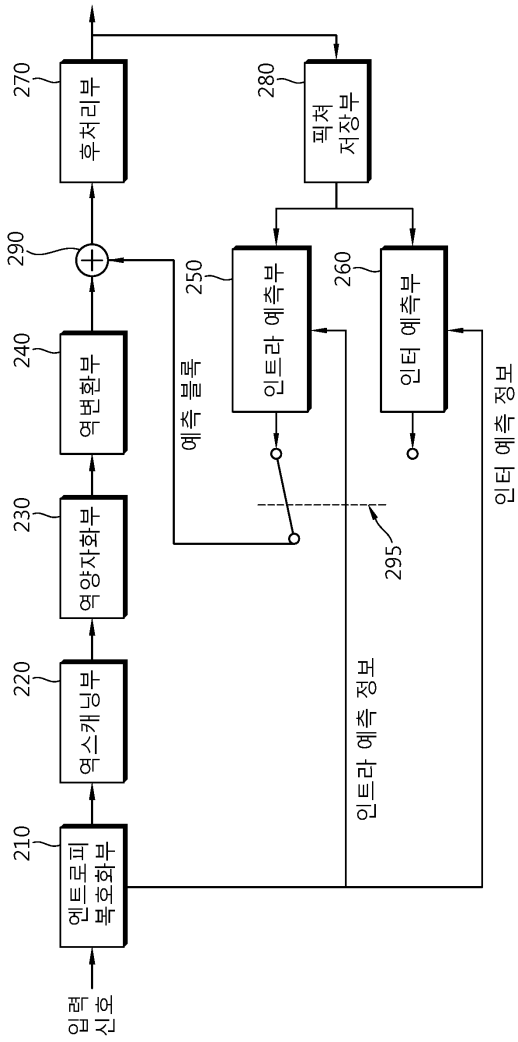
도면5



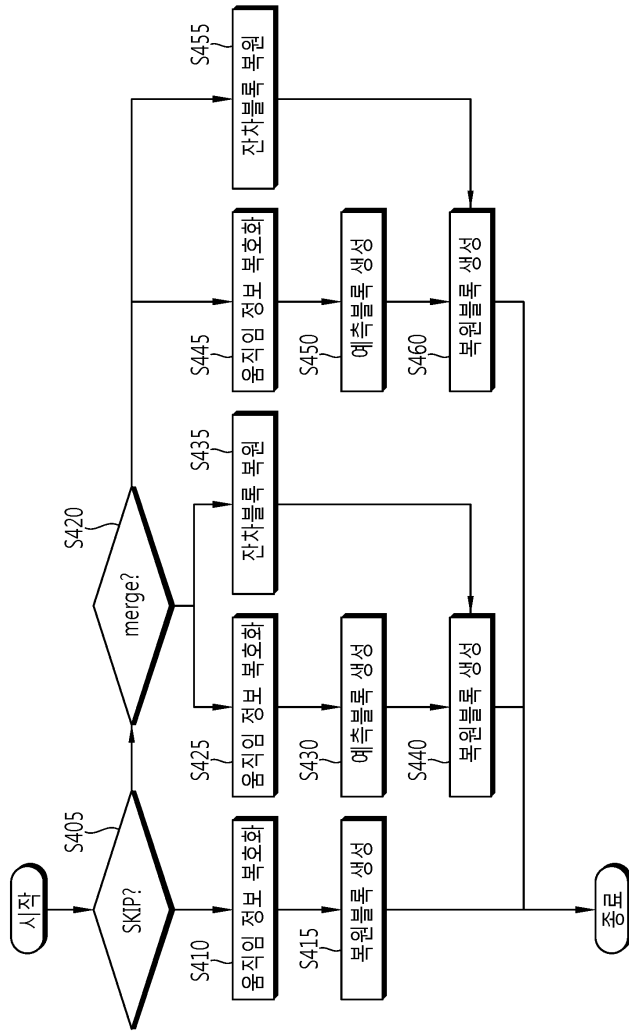
도면6



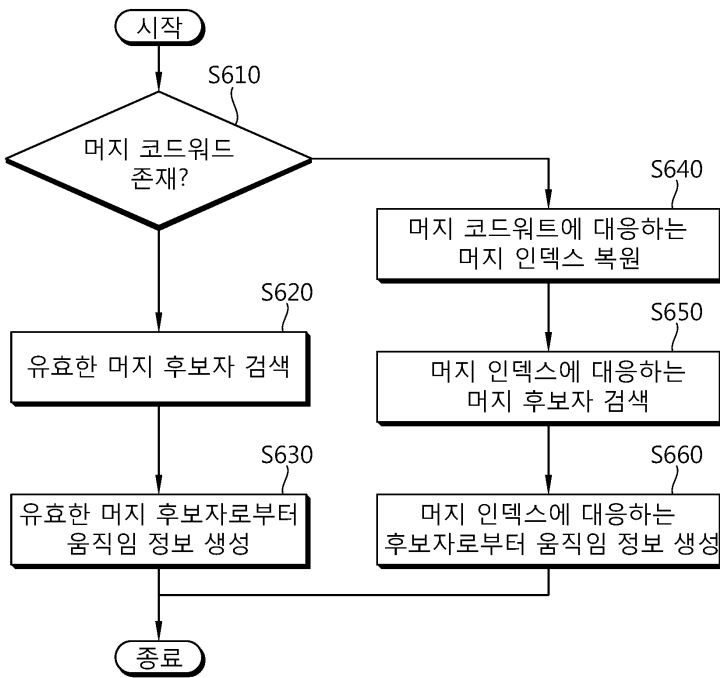
도면7



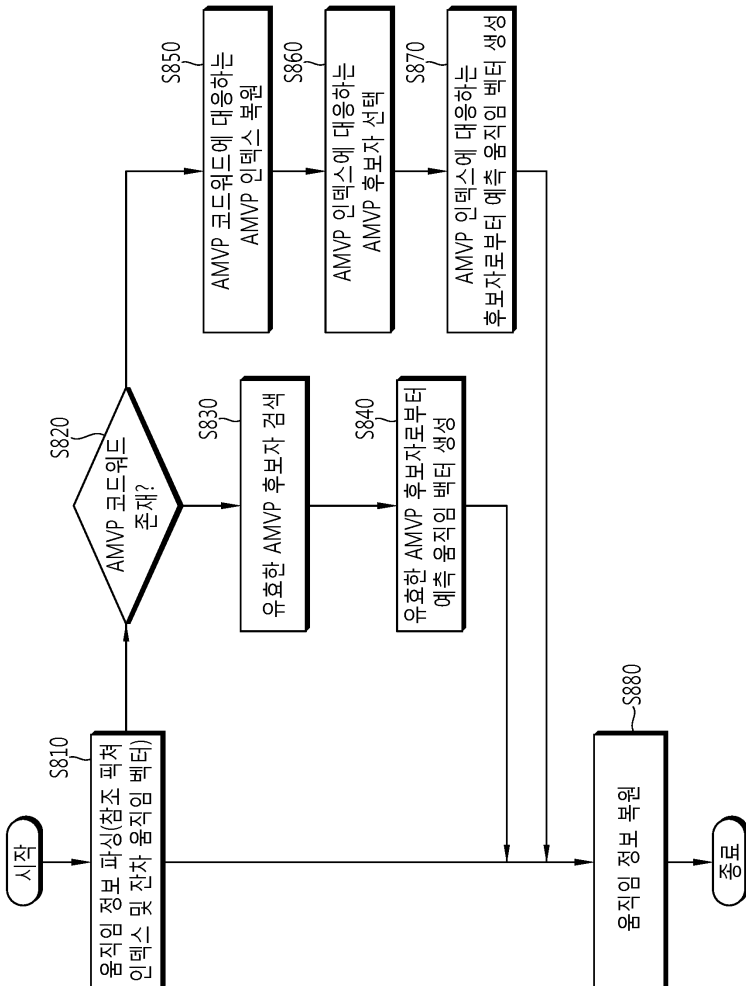
도면8



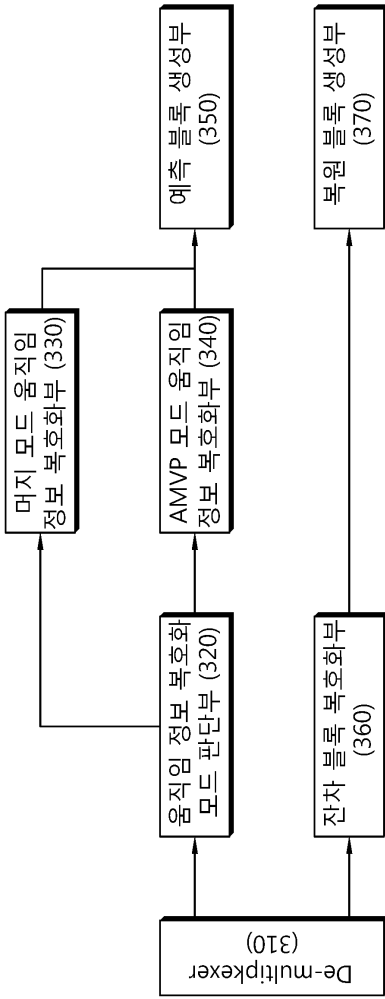
도면9



도면10

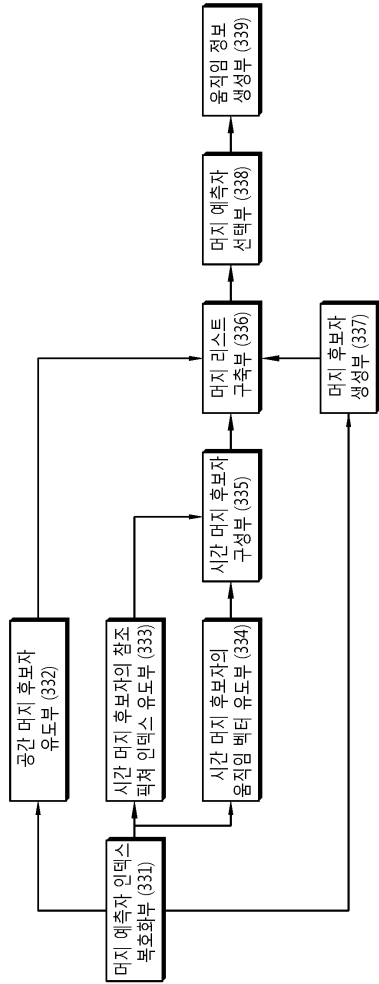


도면11

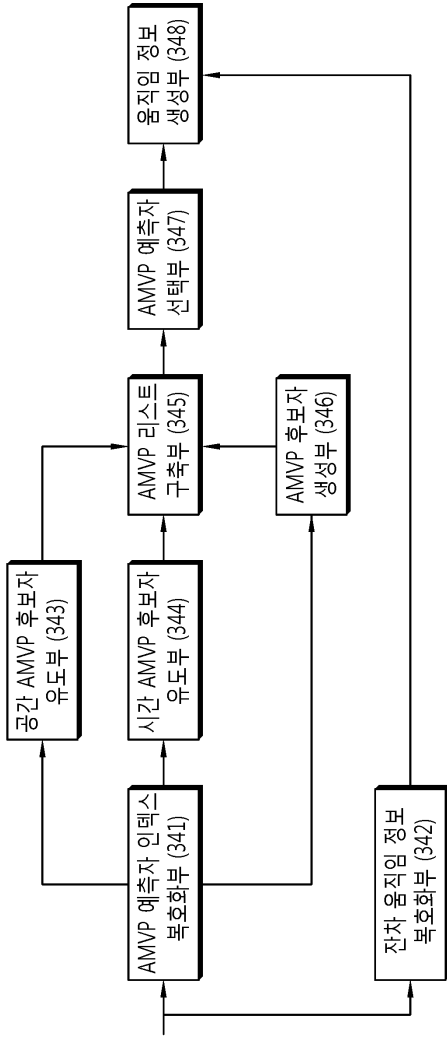




도면12



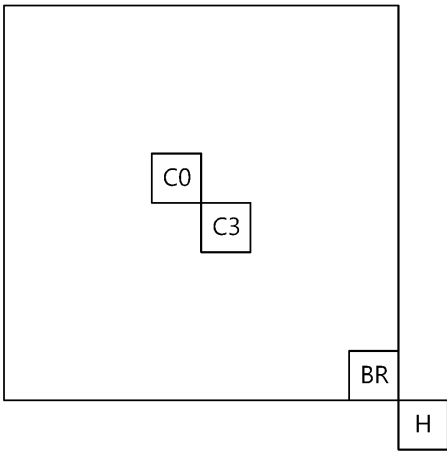
도면13



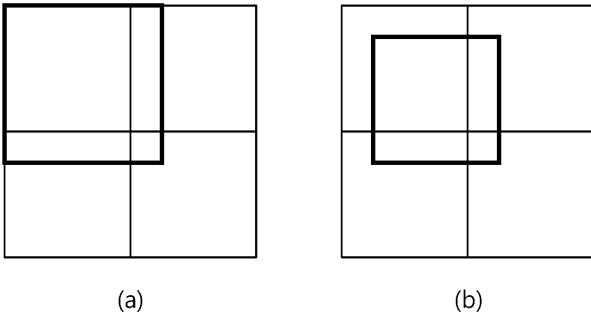
도면14

	C0		
		C3	
			BR

도면15



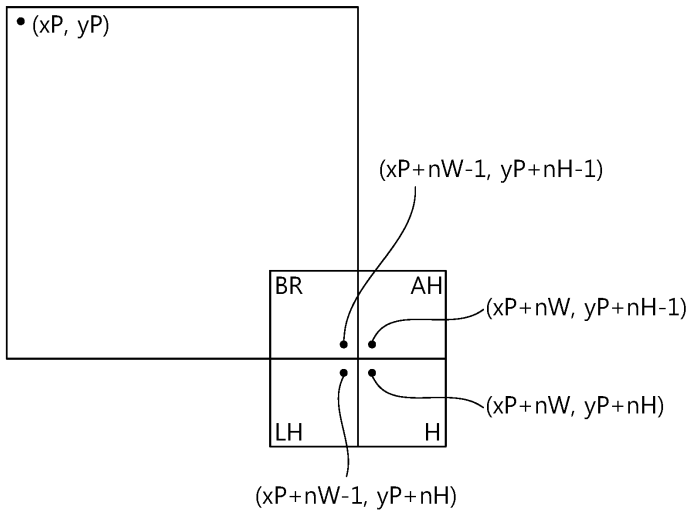
도면16



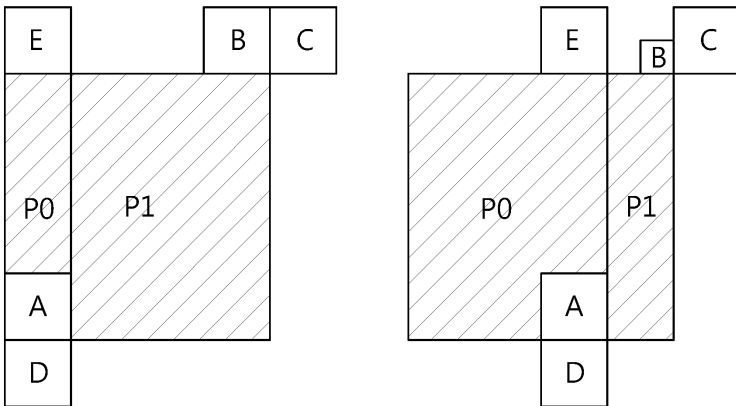
 LCUs

 Co-located motion data fetch area

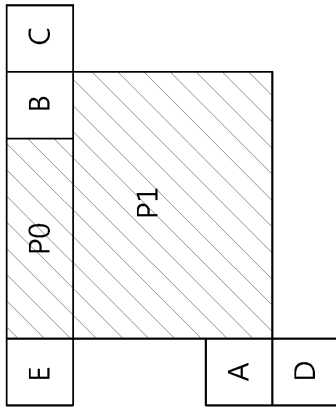
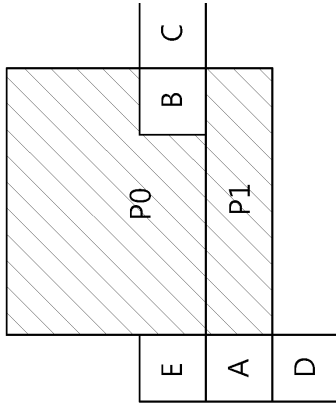
도면17



도면18



도면19

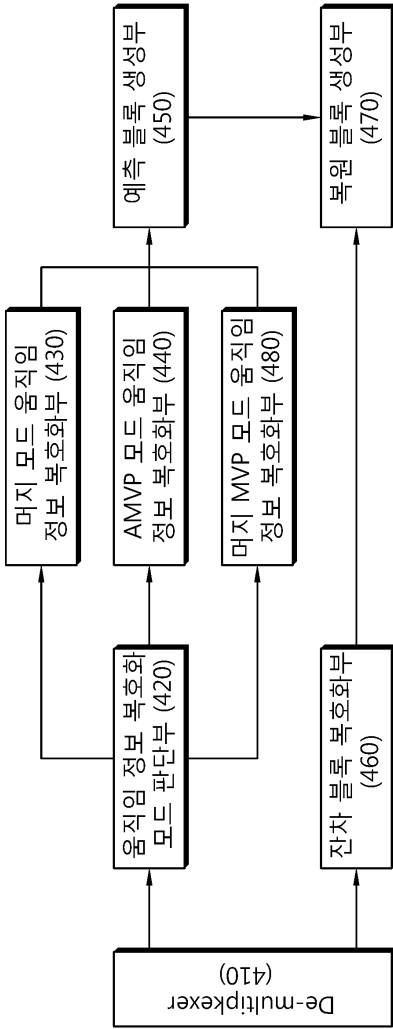


도면20

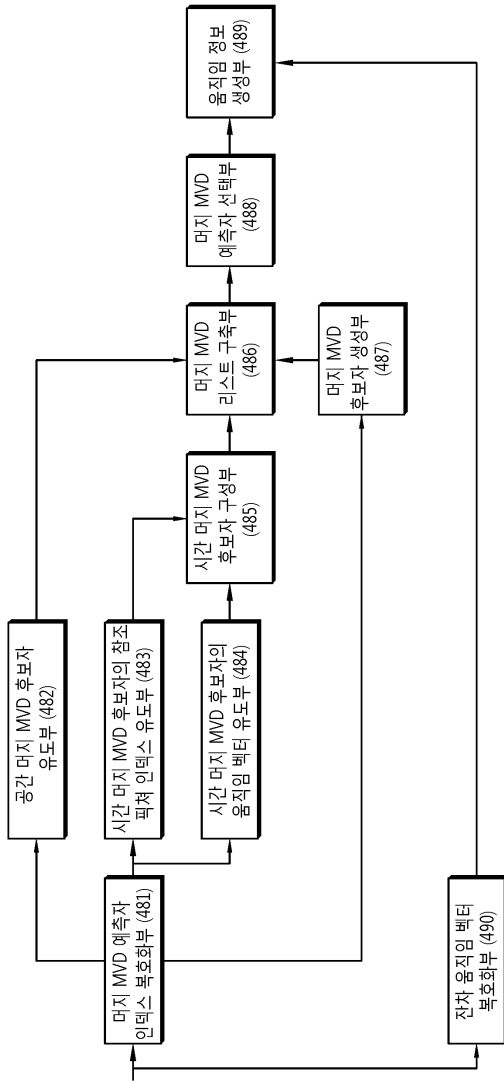
P0 ( $\alpha$ )	P1 ( $\beta$ )
P2 ( $\gamma$ )	P3



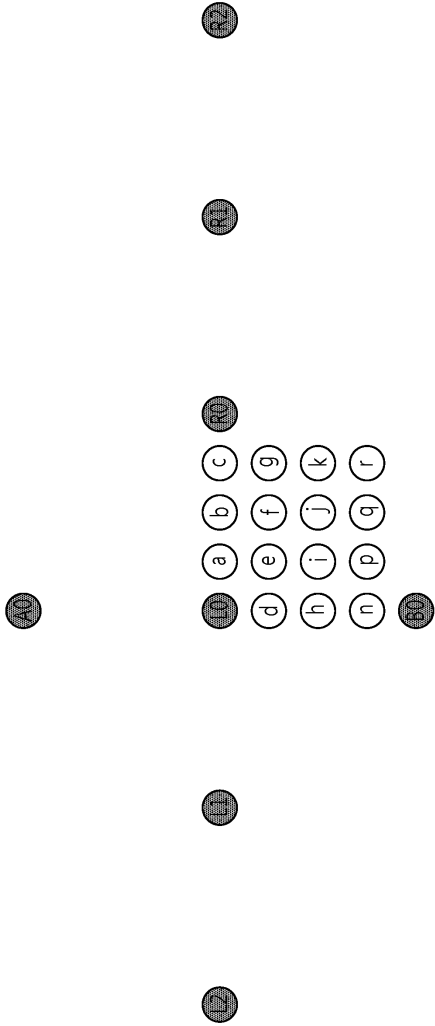
도면21



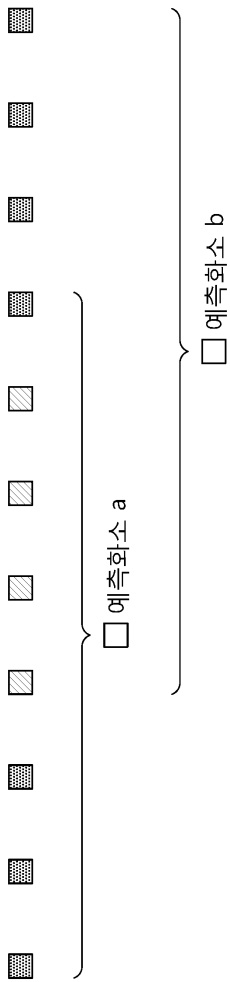
도면22



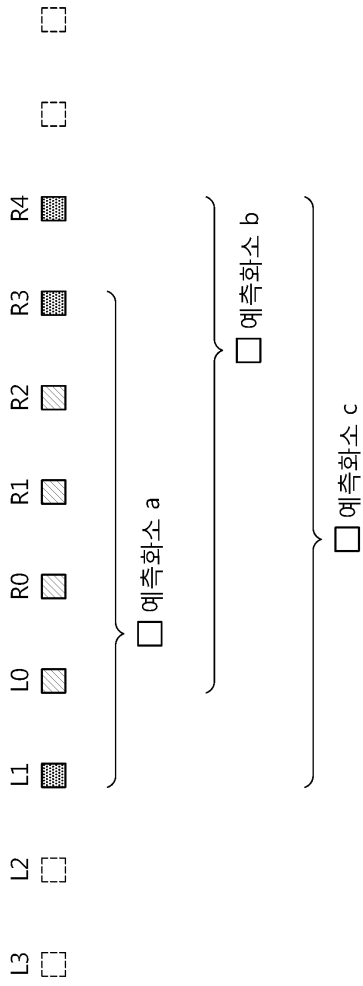
도면23



도면24



도면25





도면26

