(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0044917
(43) 공개일자 2012년05월08일

(51) 국제특허분류(Int. Cl.)
H04N 7/32 (2006.01)
(21) 출원번호 10-2011-0111661
(22) 출원일자 2011년10월28일
심사청구일자 없음
(30) 우선권주장
1020100105789 2010년10월28일 대한민국(KR)

(71) 출원인
한국전자통신연구원
대전광역시 유성구 가정로 218 (가정동)
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732, 국제캠퍼스 내 (서천동, 경희대학교)
(72) 발명자
정세윤
대전광역시 대덕구 우암로 463, 101동 1203호 (비래동, 금성백조아파트)
김휘용
대전광역시 유성구 지족동 열매마을아파트 601-201
(뒷면에 계속)
(74) 대리인
양문옥

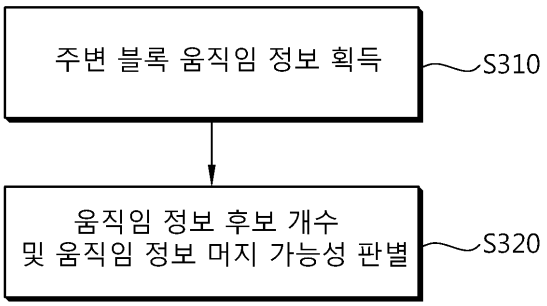
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **영상 정보 부호화 방법 및 복호화 방법**

(57) 요 약

본 발명은 영상 정보의 부호화 및 복호화 방법과 장치에 관한 것으로서, 본 발명에 따른 영상 정보 부호화 방법으로서, 현재 블록에 대한 예측을 수행하는 단계 및 상기 예측된 정보를 엔트로피 부호화하여 전송하는 단계를 포함하며, 상기 예측을 수행하는 단계는, 상기 현재 블록의 주변 블록으로부터, 상기 주변 블록의 움직임 정보를 획득하는 단계 및 상기 획득한 움직임 정보를 기반으로 상기 현재 블록의 움직임 정보를 결정하는 단계를 포함하고, 상기 전송 단계에서는 상기 현재 블록의 예측에 필요한 정보를 함께 전송할 수 있으며, 본 발명에 의하면, 영상 압축 처리의 복잡도를 낮추고 전송 정보량을 줄일 수 있다.

대 표 도 - 도3



(72) 발명자

임성창

대전광역시 유성구 신성로58번길 54, 세종빌라 201호 (신성동)

조숙희

대전광역시 유성구 봉산동 휴먼시아아파트 103동 802호

이진호

대전광역시 유성구 신성남로 65-12, 우노빌 302호 (신성동)

이하현

서울특별시 중랑구 동일로102길 34-8 (면목동)

김종호

대전광역시 유성구 신성로84번길 41, 파인하우스 205호 (신성동)

최진수

대전광역시 유성구 반석서로 98, 609동 1605호 (반석동, 반석마을6단지아파트)

김진웅

대전광역시 유성구 엑스포로 448, 305동 1603호 (전민동, 엑스포아파트)

안치득

대전광역시 유성구 도룡동 과기원아파트 2-504

박광훈

경기도 성남시 분당구 예원로 7, B동 302호 (분당동, 동아빌라)

김경용

경기도 수원시 영통구 덕영대로 1725, 402호 (영통동)

이한수

경기도 용인시 기흥구 탑실로 15, 탑실마을 대주피오레 1단지 103-902 (공세동, 탑실마을 대주피오레1단지)

이 발명을 지원한 국가연구개발사업

과제고유번호	10912-02001
부처명	방송통신위원회
연구사업명	방송통신기술개발사업(**원천기술개발사업)
연구과제명	지상파 양안식 3DTV 방송시스템 기술개발 및 표준화
주관기관	한국전자통신연구원
연구기간	2010.03.01 ~ 2013.02.28

특허청구의 범위

청구항 1

현재 블록에 대한 예측을 수행하는 단계; 및

상기 예측된 정보를 엔트로피 부호화하여 전송하는 단계를 포함하며,

상기 예측을 수행하는 단계는,

상기 현재 블록의 주변 블록으로부터, 상기 주변 블록의 움직임 정보를 획득하는 단계; 및

상기 획득한 움직임 정보를 기반으로 상기 현재 블록의 움직임 정보를 결정하는 단계를 포함하고,

상기 전송 단계에서는 상기 현재 블록의 예측에 필요한 정보를 함께 전송하는 것을 특징으로 하는 영상 정보 부호화 방법.

청구항 2

제1항에 있어서, 상기 주변 블록은,

상기 현재 블록과 동일한 부호화 유닛에 속하는 블록이고,

상기 주변 블록의 움직임 정보는 상기 부호화 유닛의 파티션 정보를 포함하는 것을 특징으로 하는 영상 정보 부호화 방법.

청구항 3

제1항에 있어서,

상기 현재 블록의 움직임 정보를 결정하는 단계에서는,

상기 주변 블록이 상기 현재 블록과 동일한 부호화 유닛에 속하는 경우에, 상기 주변 블록의 움직임 정보 및 상기 주변 블록의 움직임 정보와 동일한 움직임 정보를 상기 현재 블록의 머지 후보에서 제외하는 것을 특징으로 하는 영상 정보 부호화 방법.

청구항 4

제1항에 있어서,

상기 예측을 수행하는 단계에서는,

상기 현재 블록에 대한 움직임 정보는 상기 현재 블록과 같은 부호화 유닛에 속하는 다른 블록의 움직임 정보와 동일한 움직임 정보에는 머지하지 않는 것을 특징으로 하는 영상 정보 부호화 방법.

청구항 5

제1항에 있어서,

상기 현재 블록의 주변 블록들의 움직임 정보 중 상기 현재 블록과 같은 부호화 유닛에 속하는 다른 블록의 움직임 정보와 동일하지 않은 움직임 정보 후보가 존재하고,

상기 현재 블록의 움직임 정보를 머지 모드에 의해 예측하는 경우에,

상기 현재 블록의 움직임 정보 결정 단계에서는,

상기 현재 블록의 움직임 정보를 상기 움직임 정보 후보에 머지하는 것을 특징으로 하는 영상 정보 부호화 방법.

청구항 6

제1항에 있어서, 상기 현재 블록이 PART_NxN이고,

상기 현재 블록과 동일한 부호화 유닛에 속하는 세 블록들의 움직임 정보가 동일한 경우에,

상기 예측 단계에서는,

상기 현재 블록의 움직임 정보를 상기 현재 블록과 동일한 부호화 유닛에 속하는 세 블록들의 움직임 정보에 머지 하지 않는 것을 특징으로 하는 영상 정보 부호화 방법.

청구항 7

비트 스트림을 수신하는 단계;

상기 비트 스트림에 포함된 영상 정보를 엔트로피 복호화 하는 단계; 및

상기 엔트로피 복호화된 영상 정보를 기반으로 현재 블록에 대한 예측을 수행하는 단계를 포함하며,

상기 예측을 수행하는 단계는,

상기 현재 블록의 주변 블록으로부터, 상기 주변 블록의 움직임 정보를 획득하는 단계; 및

상기 획득한 움직임 정보를 기반으로 상기 현재 블록의 움직임 정보를 결정하는 단계를 포함하는 것을 특징으로 하는 영상 정보 복호화 방법.

청구항 8

제7항에 있어서, 상기 주변 블록은,

상기 현재 블록과 동일한 부호화 유닛에 속하는 블록이고,

상기 주변 블록의 움직임 정보는 상기 부호화 유닛의 파티션 정보를 포함하는 것을 특징으로 하는 영상 정보 복호화 방법.

청구항 9

제7항에 있어서, 상기 현재 블록의 움직임 정보를 결정하는 단계에서는,

상기 주변 블록이 상기 현재 블록과 동일한 부호화 유닛에 속하는 경우에, 상기 주변 블록의 움직임 정보 및 상기 주변 블록의 움직임 정보와 동일한 움직임 정보를 상기 현재 블록의 머지 후보에서 제외하는 것을 특징으로 하는 영상 정보 복호화 방법.

청구항 10

제7항에 있어서,

상기 예측을 수행하는 단계에서는,

상기 현재 블록에 대한 움직임 정보는 상기 현재 블록과 같은 부호화 유닛에 속하는 다른 블록의 움직임 정보와 동일한 움직임 정보에는 머지하지 않는 것을 특징으로 하는 영상 정보 복호화 방법.

청구항 11

제7항에 있어서, 상기 현재 블록의 주변 블록들의 움직임 정보 중 상기 현재 블록과 같은 부호화 유닛에 속하는 다른 블록의 움직임 정보와 동일하지 않은 움직임 정보 후보가 존재하고,

상기 현재 블록의 움직임 정보를 머지 모드에 의해 예측하는 경우에,

상기 현재 블록의 움직임 정보 결정 단계에서는,

상기 현재 블록의 움직임 정보를 상기 움직임 정보 후보에 머지하는 것을 특징으로 하는 영상 정보 복호화 방법.

청구항 12

제7항에 있어서, 상기 현재 블록이 PART_NxN이고,

상기 현재 블록과 동일한 부호화 유닛에 속하는 세 블록들의 움직임 정보가 동일한 경우에,

상기 예측 단계에서는,

상기 현재 블록의 움직임 정보를 상기 현재 블록과 동일한 부호화 유닛에 속하는 세 블록들의 움직임 정보에 머지 하지 않는 것을 특징으로 하는 영상 정보 복호화 방법.

명 세 서

기 술 분 야

[0001] 본 발명은 비디오 부호화 기술에 관한 것으로서, 더 구체적으로는 현재 블록의 움직임 정보를 주변 블록의 움직임 정보를 이용하여 예측하는 방법에 관한 것이다.

배 경 기 술

[0002] 최근 HD(High Definition) 해상도를 가지는 방송 서비스가 국내뿐만 아니라 세계적으로 확대되면서, 많은 사용자들이 고해상도, 고화질의 영상에 익숙해지고 있으며 이에 따라 많은 기관들이 차세대 영상기기에 대한 개발에 박차를 가하고 있다. 또한 HDTV와 더불어 HDTV의 4배 이상의 해상도를 갖는 UHD(Ultra High Definition)에 대한 관심이 증대되면서 보다 높은 해상도, 고화질의 영상에 대한 압축기술이 요구되고 있다.

[0003] 영상 압축 기술에는 시간적으로 이전 및/또는 이후의 픽쳐로부터 현재 픽쳐에 포함된 화소값을 예측하는 인터(inter) 예측 기술, 현재 픽쳐 내의 화소 정보를 이용하여 현재 픽쳐에 포함된 화소값을 예측하는 인트라(intra) 예측 기술, 조명 변화 등에 의한 화질의 열화를 방지하기 위한 가중치 예측 기술, 출현 빈도가 높은 심볼(symbol)에 짧은 부호를 할당하고 출현 빈도가 낮은 심볼에 긴 부호를 할당하는 엔트로피 부호화 기술 등이 있다. 특히, 생략 모드(skip mode)에서 현재 블록에 대한 예측이 수행되는 경우, 이전 부호화된 영역으로부터 예측된 값만을 이용하여 예측 블록이 생성되고 별도의 움직임 정보(motion information)나 레지듀얼(residual) 신호가 부호화기로부터 복호화기로 전송되지 않는다. 이러한 영상 압축 기술들에 의해 영상 데이터가 효율적으로 압축될 수 있다.

[0004] 한편, 움직임 정보의 부호화 방법 중 하나로 머지(merge) 방법이 이용되고 있다. 머지는 코딩 유닛(Coding Unit: CU, 이하 'CU'라 함) 단위의 머지와 예측 유닛(Prediction Unit: PU, 이하 'PU'라 함) 단위의 머지가 가능하다. CU또는 PU(이하, 설명의 편의를 위해 '블록'이라 함) 단위로 머지를 수행하는 경우에는, 블록 파티션(partition)별로 머지를 수행할지에 관한 정보와 현재 블록에 인접한 주변 블록(현재 블록의 좌측 인접 블록, 현재 블록의 상측 인접 블록, 현재 블록의 시간적(temporral) 인접 블록 등) 중 어떤 블록과 머지를 할 것인지에 대한 정보를 전송할 필요가 있다. 따라서, 머지에 대한 정보량이 증가하는 문제가 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명의 기술적 과제는 머지(merge) 방법을 이용한 영상 정보의 부호화 방법 및 장치를 제공하는 것이다.

[0006] 본 발명의 다른 기술적 과제는 부호화 유닛의 파티션 정보를 이용하여 현재 블록의 움직임 정보를 유추하는 방법 및 장치를 제공하는 것이다.

[0007] 본 발명의 다른 기술적 과제는 동일한 부호화 유닛 내 다른 블록의 움직임 정보를 이용하여 현재 블록의 움직임 정보를 유추하는 방법 및 장치를 제공하는 것이다.

[0008] 본 발명의 다른 기술적 과제는 현재 블록의 움직임 정보를 직접 전송하지 않고, 현재 블록의 움직임 정보를 유도함으로써 영상 압축 처리의 복잡도를 낮추고 전송 정보량을 줄이는 방법 및 장치를 제공하는 것이다.

과제의 해결 수단

[0009] (1) 본 발명의 일 실시형태는 영상 정보 부호화 방법으로서, 현재 블록에 대한 예측을 수행하는 단계 및 상기 예측된 정보를 엔트로피 부호화하여 전송하는 단계를 포함하며, 상기 예측을 수행하는 단계는, 상기 현재 블록의 주변 블록으로부터, 상기 주변 블록의 움직임 정보를 획득하는 단계 및 상기 획득한 움직임 정보를 기반으로 상기 현재 블록의 움직임 정보를 결정하는 단계를 포함하고, 상기 전송 단계에서는 상기 현재 블록의 예측에 필요한 정보를 함께 전송할 수 있다.

[0010] (2) (1)에서, 상기 주변 블록은, 상기 현재 블록과 동일한 부호화 유닛에 속하는 블록이고, 상기 주변 블록의 움직임 정보는 상기 부호화 유닛의 파티션 정보를 포함할 수 있다.

[0011] (3) (1)에서, 상기 현재 블록의 움직임 정보를 결정하는 단계에서는, 상기 주변 블록이 상기 현재 블록과 동일한 부호화 유닛에 속하는 경우에, 상기 주변 블록의 움직임 정보 및 상기 주변 블록의 움직임 정보와 동일한 움직임 정보를 상기 현재 블록의 머지 후보에서 제외할 수 있다.

[0012] (4) (1)에서, 상기 예측을 수행하는 단계에서는, 상기 현재 블록에 대한 움직임 정보는 상기 현재 블록과 같은 부호화 유닛에 속하는 다른 블록의 움직임 정보와 동일한 움직임 정보에는 머지하지 않을 수 있다.

[0013] (5) (1)에서, 상기 현재 블록의 주변 블록들의 움직임 정보 중 상기 현재 블록과 같은 부호화 유닛에 속하는 다른 블록의 움직임 정보와 동일하지 않은 움직임 정보 후보가 존재하고, 상기 현재 블록의 움직임 정보를 머지 모드에 의해 예측하는 경우에, 상기 현재 블록의 움직임 정보 결정 단계에서는, 상기 현재 블록의 움직임 정보를 상기 움직임 정보 후보에 머지할 수 있다.

[0014] (6) (1)에서, 제1항에 있어서, 상기 현재 블록이 PART_NxN이고, 상기 현재 블록과 동일한 부호화 유닛에 속하는 세 블록들의 움직임 정보가 동일한 경우에, 상기 예측 단계에서는, 상기 현재 블록의 움직임 정보를 상기 현재 블록과 동일한 부호화 유닛에 속하는 세 블록들의 움직임 정보에 머지 하지 않을 수 있다.

[0015] (7) 본 발명의 다른 실시형태는 영상 정보 복호화 방법으로서, 비트 스트림을 수신하는 단계, 상기 비트 스트림에 포함된 영상 정보를 엔트로피 복호화 하는 단계 및 상기 엔트로피 복호화된 영상 정보를 기반으로 현재 블록에 대한 예측을 수행하는 단계를 포함하며, 상기 예측을 수행하는 단계는, 상기 현재 블록의 주변 블록으로부터, 상기 주변 블록의 움직임 정보를 획득하는 단계 및 상기 획득한 움직임 정보를 기반으로 상기 현재 블록의 움직임 정보를 결정하는 단계를 포함할 수 있다.

[0016] (8) (7)에서, 상기 주변 블록은, 상기 현재 블록과 동일한 부호화 유닛에 속하는 블록이고, 상기 주변 블록의 움직임 정보는 상기 부호화 유닛의 파티션 정보를 포함할 수 있다.

[0017] (9) (7)에서, 상기 현재 블록의 움직임 정보를 결정하는 단계에서는, 상기 주변 블록이 상기 현재 블록과 동일한 부호화 유닛에 속하는 경우에, 상기 주변 블록의 움직임 정보 및 상기 주변 블록의 움직임 정보와 동일한 움직임 정보를 상기 현재 블록의 머지 후보에서 제외할 수 있다.

[0018] (10) (7)에서, 상기 예측을 수행하는 단계에서는, 상기 현재 블록에 대한 움직임 정보는 상기 현재 블록과 같은 부호화 유닛에 속하는 다른 블록의 움직임 정보와 동일한 움직임 정보에는 머지하지 않을 수 있다.

[0019] (11) (7)에서, 상기 현재 블록의 주변 블록들의 움직임 정보 중 상기 현재 블록과 같은 부호화 유닛에 속하는 다른 블록의 움직임 정보와 동일하지 않은 움직임 정보 후보가 존재하고, 상기 현재 블록의 움직임 정보를 머지 모드에 의해 예측하는 경우에, 상기 현재 블록의 움직임 정보 결정 단계에서는, 상기 현재 블록의 움직임 정보를 상기 움직임 정보 후보에 머지할 수 있다.

[0020] (12) (7)에서, 상기 현재 블록이 PART_NxN이고, 상기 현재 블록과 동일한 부호화 유닛에 속하는 세 블록들의 움직임 정보가 동일한 경우에, 상기 예측 단계에서는, 상기 현재 블록의 움직임 정보를 상기 현재 블록과 동일한 부호화 유닛에 속하는 세 블록들의 움직임 정보에 머지 하지 않을 수 있다.

발명의 효과

[0021] 본 발명에 의하면, 영상 정보의 압축 성능을 향상시키고, 정보의 전송량을 줄일 수 있다.

[0022] 본 발명에 의하면, 현재 블록에 머지를 적용하는 경우에도 머지 정보의 전송량을 줄일 수 있다.

[0023] 본 발명에 의하면, 부호화 유닛 혹은 예측 유닛의 파티션 정보를 이용하여 현재 블록의 움직임 정보를 유추함으로써, 영상 압축 처리의 복잡도를 낮추고 전송 정보량을 줄일 수 있다.

[0024] 본 발명에 의하면, 동일한 부호화 유닛 내 다른 블록의 움직임 정보를 이용하여 현재 블록의 움직임 정보를 유추함으로써 영상 압축 처리의 복잡도를 낮추고 전송 정보량을 줄일 수 있다.

도면의 간단한 설명

[0025] 도 1은 영상 부호화 장치의 일 실시예에 따른 구성을 나타내는 블록도이다.

도 2는 영상 복호화 장치의 일 실시예에 따른 구성을 나타내는 블록도이다.

도 3은 본 발명이 적용되는 시스템에서 부호화기와 복호화기의 예측에 관한 동작의 일 예를 개략적으로 설명하는 순서도이다.

도 4는 본 발명이 적용되는 시스템에서 부호화기와 복호화기의 예측에 관한 동작의 다른 예를 개략적으로 설명하는 순서도이다.

도 5는 하나의 CU가 여러 형태의 PU로 나뉘는 일 예를 개략적으로 설명하는 도면이다.

도 6은 하나의 CU로부터 나뉘어질 수 있는 PU의 여러 형태를 개략적으로 나타낸 도면이다.

도 7은 CU로부터 나뉜 두 PU의 머지 대상에 관하여 개략적으로 설명하는 도면이다.

도 8은 정해진 위치의 블록으로부터 후보 움직임 정보를 얻어내는 방법을 개략적으로 도시한 것이다.

도 9는 PU에 적용되는 머지의 개념을 개략적으로 설명하는 도면이다.

도 10은 CU가 4개의 PU로 나뉘어진 경우에, 각 PU의 움직임 정보에 관하여 개략적으로 설명하는 도면이다.

도 11은 CU로부터 나뉘어지는 PU 간의 관계를 개략적으로 나타낸 것이다.

도 12는 하나의 CU가 두 개의 PU로 나뉘어져 있을 때, 현재 PU의 주변 블록 중 현재 블록의 머지 후보 블록이 될 수 있는 블록을 개략적으로 도시한 것이다.

도 13 내지 15는 주변 블록이 현재 블록과 동일한 CU 내의 다른 블록인 경우에 현재 블록이 머지하는 여러 예를 도시한 것이다.

도 16은 동일 CU 내 두 블록에 대하여, 다른 블록의 머지 정보를 이용하여 현재 블록의 머지 정보를 결정하는 예를 개략적으로 설명하는 도면이다.

도 17 내지 19는 머지 후보 블록을 소정의 주변 블록으로 한정한 경우에, 다른 블록의 움직임 정보를 이용하여 현재 블록의 움직임 정보를 결정하는 여러 예를 개략적으로 설명하는 도면이다.

도 20은 본 발명이 적용되는 시스템에서 부호화기의 동작을 개략적으로 설명하는 순서도이다.

도 21은 본 발명이 적용되는 시스템에서 복호화기의 동작을 개략적으로 설명하는 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0026] 이하, 도면을 참조하여 본 발명의 실시 형태에 대하여 구체적으로 설명한다. 본 명세서의 실시예를 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 명세서의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다.

[0027] 어떤 구성 요소가 다른 구성 요소에 “연결되어” 있다거나 “접속되어” 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있으나, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 아울러, 본 발명에서 특정 구성을 “포함”한다고 기술하는 내용은 해당 구성 이외의 구성을 배제하는 것이 아니며, 추가적인 구성이 본 발명의 실시 또는 본 발명의 기술적 사상의 범위에 포함될 수 있음을 의미한다.

[0028] 제1, 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다.

[0029] 또한 본 발명의 실시예에 나타나는 구성부들은 서로 다른 특징적인 기능들을 나타내기 위해 독립적으로 도시되는 것으로, 각 구성부들이 분리된 하드웨어나 하나의 소프트웨어 구성단위로 이루어짐을 의미하지 않는다. 즉, 각 구성부는 설명의 편의상 각각의 구성부로 나열하여 포함한 것으로 각 구성부 중 적어도 두 개의 구성부가 합쳐져 하나의 구성부로 이루어지거나, 하나의 구성부가 복수 개의 구성부로 나뉘어져 기능을 수행할 수 있고 이러한 각 구성부의 통합된 실시예 및 분리된 실시예도 본 발명의 본질에서 벗어나지 않는 한 본 발명의 권리범위에 포함된다.

[0030] 또한, 일부의 구성 요소는 본 발명에서 본질적인 기능을 수행하는 필수적인 구성 요소는 아니고 단지 성능을 향상시키기 위한 선택적 구성 요소일 수 있다. 본 발명은 단지 성능 향상을 위해 사용되는 구성 요소를 제외한 본 발명의 본질을 구현하는데 필수적인 구성부만을 포함하여 구현될 수 있고, 단지 성능 향상을 위해 사용되는 선택적 구성 요소를 제외한 필수 구성 요소만을 포함한 구조도 본 발명의 권리범위에 포함된다.

[0031] 도 1은 영상 부호화 장치의 일 실시예에 따른 구성을 나타내는 블록도이다.

[0032] 도 1을 참조하면, 상기 영상 부호화 장치(100)는 움직임 예측부(110), 움직임 보상부(115), 인트라 예측부(120), 감산기(125), 변환부(130), 양자화부(135), 엔트로피 부호화부(140), 역양자화부(145), 역변환부(150), 가산기(155), 필터부(160) 및 참조영상 버퍼(165)를 포함한다.

[0033] 영상 부호화 장치(100)는 입력 영상에 대해 인트라(intra) 모드 또는 인터(inter) 모드로 부호화를 수행하고 비트 스트림(bit stream)을 출력할 수 있다. 인트라 모드인 경우에는 인트라 예측부(120)에서 예측이 수행되고, 인터 모드의 경우에는 움직임 예측부(110), 움직임 보상부(115) 등을 통해서 예측이 수행될 수 있다. 영상 부호화 장치(100)는 입력 영상의 입력 블록에 대한 예측 블록을 생성한 후, 입력 블록과 예측 블록의 차분을 부호화할 수 있다.

[0034] 인트라 모드인 경우, 인트라 예측부(120)는 현재 블록 주변의 이미 부호화된 블록의 화소값을 이용하여 공간적 예측을 수행하여 예측 블록을 생성할 수 있다.

[0035] 인터 모드인 경우, 움직임 예측부(110)는, 움직임 예측 과정에서 참조 영상 버퍼(165)에 저장되어 있는 참조 영상에서 입력 블록과 가장 매치가 잘 되는 영역을 찾아 움직임 벡터를 구할 수 있다. 움직임 보상부(115)는 움직임 벡터와 참조 영상 버퍼(165)에 저장되어 있는 참조 영상을 이용하여 움직임 보상을 수행함으로써 예측 블록을 생성할 수 있다.

[0036] 감산기(125)는 입력 블록과 생성된 예측 블록의 차분에 의해 레지듀얼 블록(residual block)을 생성할 수 있다. 변환부(130)는 레지듀얼 블록에 대해 변환(transform)을 수행하여 변환 계수(transform coefficient)를 출력할 수 있다. 레지듀얼 신호는 원신호와 예측 신호의 차이를 의미할 수 있고, 또한 원신호와 예측 신호의 차이가 변환(transform)된 형태의 신호 또는 원신호와 예측 신호의 차이가 변환되고 양자화된 형태의 신호를 의미할 수도 있다. 레지듀얼 신호를 블록 단위에서는 레지듀얼 블록이라 할 수 있다.

[0037] 양자화부(135)는 변환 계수를 양자화 파라미터에 따라 양자화한 양자화 계수(quantized coefficient)를 출력할 수 있다.

[0038] 엔트로피 부호화부(140)는, 양자화부(135)에서 산출된 값들 또는 부호화 과정에서 산출된 부호화 파라미터 값 등에 대응하는 심볼(symbol)을 확률 분포에 따라 엔트로피 부호화하여 비트 스트림(bit stream)을 출력할 수 있다.

[0039] 엔트로피 부호화가 적용되는 경우, 높은 발생 확률을 갖는 심볼에 적은 수의 비트가 할당되고 낮은 발생 확률을 갖는 심볼에 많은 수의 비트가 할당되도록 함으로써 영상 부호화의 압축 성능을 높일 수 있다.

[0040] 엔트로피 부호화를 위해 CAVLC(Context-Adaptive Variable Length Coding), CABAC(Context-Adaptive Binary Arithmetic Coding) 등과 같은 부호화 방법이 사용될 수 있다. 예를 들어, 엔트로피 부호화부(140)에는 가변 길이 부호화(VLC: Variable Lenghth Coding/Code) 테이블을 사용하여 엔트로피 부호화를 수행할 수 있다. 엔트로피 부호화부(140)는 대상 심볼의 이진화(binarization) 방법 및 대상 심볼/빈(bin)의 확률 모델(probability model)을 도출한 후, 도출된 이진화 방법 또는 확률 모델을 사용하여 엔트로피 부호화를 수행할 수도 있다.

[0041] 양자화된 계수는 역양자화부(145)에서 역양자화되고 역변환부(150)에서 역변환될 수 있다. 역양자화되고, 역변환된 계수는 복원된 레지듀얼 블록(reconstructed residual block)으로 생성 되며, 가산기(155)는 예측 블록과 복원된 레지듀얼 블록을 이용하여 복원 블록을 생성할 수 있다.

[0042] 필터부(160)는 디블록킹 필터(deblocking filter), SAO(Sample Adaptive Offset), ALF(Adaptive Loop Filter) 중 적어도 하나 이상을 복원 블록 또는 복원 픽쳐에 적용할 수 있다. 필터부(160)를 거친 복원 블록은 참조 영상 버퍼(165)에 저장될 수 있다.

[0043] 도 2는 영상 복호화 장치의 일 실시예에 따른 구성을 나타내는 블록도이다.

[0044] 도 2를 참조하면, 상기 영상 복호화 장치(200)는 엔트로피 복호화부(210), 역양자화부(220), 역변환부(230), 인트라 예측부(240), 움직임 보상부(250), 필터부(260), 참조 영상 버퍼(270) 및 가산기(280)를 포함한다.

[0045] 영상 복호화 장치(200)는 부호화기에서 출력된 비트 스트림을 입력 받아 인트라 모드 또는 인터 모드로 복호화를 수행하고 재구성된 영상, 즉 복원 영상을 출력할 수 있다. 인트라 모드인 경우에는 인트라 예측부(240)에서 예측이 수행되고, 인터 모드인 경우에는 움직임 보상부(250)를 통해서 예측이 수행될 수 있다. 영상 복호화 장치(200)는 입력 받은 비트스트림으로부터 복원된 레지듀얼 블록(residual block)을 얻고 예측 블록을 생성한 후 복원된 레지듀얼 블록과 예측 블록을 더하여 재구성된 블록, 즉 복원 블록을 생성할 수 있다.

[0046] 엔트로피 복호화부(210)는, 입력된 비트 스트림을 확률 분포에 따라 엔트로피 복호화하여 양자화된 계수(quantized coefficient) 형태의 심볼들을 생성할 수 있다. 엔트로피 복호화 방법은 상술한 엔트로피 부호화 방법에 대응하여 수행될 수 있다.

[0047] 양자화된 계수는 역양자화부(220)에서 역양자화되고 역변환부(230)에서 역변환되며, 양자화된 계수가 역양자화/역변환 된 결과 복원된 레지듀얼 블록(residual block)이 생성될 수 있다.

[0048] 인트라 모드인 경우, 인트라 예측부(240)는 현재 블록 주변의 이미 복호화된 블록의 화소값을 이용하여 공간적 예측을 수행하여 예측 블록을 생성할 수 있다. 인터 모드인 경우, 움직임 보상부(250)는 움직임 벡터 및 참조 영상 버퍼(270)에 저장되어 있는 참조 영상을 이용하여 움직임 보상을 수행함으로써 예측 블록을 생성할 수 있다.

[0049] 가산기(280)는 복원된 레지듀얼 블록과 예측 블록을 기반으로 복원 블록을 생성할수 있다. 필터부(260)는 디블록킹 필터, SAO, ALF 중 적어도 하나 이상을 복원 블록에 적용할 수 있다. 필터부(260)는 재구성된 영상, 즉 복원 영상을 출력한다. 복원 영상은 참조 영상 버퍼(270)에 저장되어 화면 간 예측에 사용될 수 있다.

[0050] 한편, 현재 블록에 대하여 인터 예측을 하는 경우에, 머지(merge) 모드, 스킵(skip) 등의 예측 모드를 이용할 수 있다.

[0051] 이 중 머지 모드에서는 현재 블록의 움직임 정보를 인접한 다른 블록의 움직임 정보와 머지할 수 있다. 머지(merge)한다는 것은, 인터 예측을 수행하는 경우에, 현재 블록의 주변 블록의 움직임 정보를 현재 블록의 움직임 정보로 이용하는 것을 말한다. 이때, 현재 블록의 움직임 정보로 이용될 수 있는 주변 블록의 움직임 정보를 설명의 편의를 위해 '움직임 정보 후보' 라고 표현 할 수도 있다. 또한, 현재 블록의 주변 블록으로서 움직임 정보 후보를 가지는 블록을 '(머지) 후보 블록' 으로 표시할 수도 있다. 또한, 후보 블록들 중에서 현재 블록의 움직임 정보로 사용할 움직임 정보를 가지는 후보 블록 또는 움직임 정보 후보 중 현재 블록의 움직임 정보로 사용되는 움직임 정보 후보를 '머지 대상' 이라고 표시할 수 있다. 따라서, 설명의 편의를 위해, 현재 블록이 후보 블록 중 선택된 블록에 머지한다는 표현을 사용할 수 있으나, 이는 당업자가 이해하는 바와 같이, 현재 블록의 움직임 정보가 후보 블록의 움직임 정보에 머지하여, 후보 블록의 움직임 정보 후보를 현재 블록의 움직임 정보로 사용한다는 의미임에 유의한다.

[0052] 이때, 움직임 정보는 움직임 벡터(motion vector), 참조 영상 색인(reference picture index), 단방향 예측(uni-prediction)인지 양방향 예측(bi-prediction)인지 그리고 단방향 예측 중 어떤 방향 예측인지를 나타내는 인터 예측 모드(inter prediction mode)를 포함할 수 있다. 즉, 인터 예측 모드는 참조 영상 리스트 중 어떤 참조 영상 리스트(reference picture list)에 포함된 영상들 중 어떤 조합으로 사용할지에 관한 것이다. 또한, 움직임 정보는 블록의 파티션 정보를 포함할 수도 있다.

[0053] 두 움직임 정보 예컨대, 제1 움직임 정보와 제2 움직임 정보가 동일하다는 것은 제1 움직임 정보의 움직임 벡터, 참조 영상, 인터 예측 모드가 제2 움직임 정보의 움직임 벡터, 참조 영상, 인터 예측 모드와 모두 동일하다는 것을 의미할 수 있다.

[0054] 현재 블록에 대하여 머지가 수행되는 경우에, 부호화기는 머지에 관한 정보를 복호화기에 전송할 수 있다. 머지에 관한 정보는 현재 블록이 머지 되었는지 여부를 나타내는 정보로서 머지 플래그(merge_flag), 현재 블록이 어떤 블록과 머지 하는지를 나타내는 정보로서 머지 인덱스(merge_idx) 등을 포함할 수 있다.

[0055] 머지를 수행하는 경우에는, 주변 블록의 파티션(partition) 정보 혹은 현재 블록의 파티션 정보로부터 현재 블록의 머지 가능성을 확인한다. 확인 결과를 기반으로, 현재 블록에 머지를 수행할 지와 머지를 수행한다면 어떤 주변 블록에 머지할 것인지를 결정할 수 있다.

[0056] 현재 블록에 대해서 머지를 수행하는지에 관한 정보와 현재 블록을 어느 블록에 머지할 것인지에 관한 정보 등

머지에 관한 정보를 모두 복호화기에 전송하기 위해서는 많은 량의 정보를 처리할 필요가 있으며, 정보 전달량도 크다.

[0057] 이와 관련하여, 본 명세서에서는 머지에 관한 정보 중 일부 또는 전부를 전송하지 않아도 되는 경우에 이를 생략하는 방법과 머지 대상의 우선 순위를 변경하여 머지 정보를 표현/전송하는데 필요한 비트를 줄이는 방법을 개시한다.

[0058] 본 명세서에서 사용하는 용어의 정의는 아래와 같다.

[0059] CU(Coding Unit)는 비디오 부호화/복호화(영상 정보 부호화/복호화)의 기본 단위이다. PU(Prediction Unit)는 예측(Prediction)의 기본 단위로, PU 파티션(partition)들로 구성된다. 각 PU 파티션들은 인터 예측일 경우 움직임 정보를 담고 있다.

[0060] 머지 플래그(merge_flag)는 현재 블록과 주변 블록의 움직임 정보가 일치하는지를 지시한다. 예컨대, merge_flag는 현재 블록과 주변 블록의 움직임 정보가 일치하는 경우 논리값 1을 갖고, 현재 블록과 주변 블록의 움직임 정보가 일치하지 않는 경우 논리값 0을 갖는다. 혹은 현재 블록과 주변 블록의 움직임 정보가 일치하는 경우 논리값 0을 갖고, 현재 블록과 주변 블록의 움직임 정보가 일치하지 않는 경우 논리값 1을 갖도록 merge_flag 값을 설정할 수 있다.

[0061] 머지 레프트 플래그(merge_left_flag)는 머지(Merge) 가능한 움직임 정보를 갖는 주변 블록 대상이 2개(좌측 블록, 상측 블록) 존재할 때, 현재 블록과 움직임 정보가 동일한 블록을 지시한다. 예컨대, 현재 블록의 움직임 정보가 왼쪽 블록의 움직임 정보와 같은 경우 논리값 1을 갖고, 현재 블록의 움직임 정보가 왼쪽 블록의 움직임 정보와 같지 않은 경우(위쪽 블록과 움직임 정보가 일치하는 경우) 논리값 0을 갖는다. 혹은 현재 블록의 움직임 정보가 왼쪽 블록의 움직임 정보와 같은 경우 논리값 0을 갖고, 현재 블록의 움직임 정보가 왼쪽 블록의 움직임 정보와 같지 않은 경우(위쪽 블록과 움직임 정보가 일치하는 경우) 논리값 1을 갖도록 merge_left_flag 값을 설정할 수 있다.

[0062] 머지 인덱스(merge_idx, 이하 'merge_idx'라 함)는 머지 후보 리스트 중에서 현재 블록이 머지할 후보에 대한 인덱스를 나타낸다. 머지 인덱스 값이 지시되지 않는다면, '0'을 갖는 것으로 추정(infer)될 수 있다.

[0063] 머지(merge) 후보 리스트는 움직임 정보들이 저장된 리스트를 나타낸다. 여기서 머지 후보 리스트에 저장되는 움직임 정보는 현재 블록에 인접한 주변 블록의 움직임 정보이거나 참조 영상에서 현재 블록에 대응되는(collocated) 블록의 움직임 정보일 수 있다. 또한 머지 후보 리스트에 저장되는 움직임 정보는 이미 머지 후보 리스트에 존재하는 움직임 정보들을 조합하여 만든 새로운 움직임 정보일 수 있다.

[0064] 본 명세서에서 블록(block)은 상술한 바와 같이, CU, PU, PU 파티션(Partition) 또는 매크로 블록을 의미할 수 있으며, 여러 픽셀로 구성되어 있다. 따라서, 당업자라면 본 명세서에서 기재되는 발명의 내용에 따라 필요로 하는 정보 또는 대상 프로세스 등에서 블록이 의미하는 대상을 유추할 수 있다.

[0065] 이하, 상술한 용어들과 도면을 참조하여, 머지에 관한 정보 중 일부 또는 전부의 전송을 생략하는 방법과 머지 대상의 우선 순위를 변경하는 방법에 관해 구체적으로 설명한다.

[0066] 머지에 관한 정보는 현재 블록이 머지하는지를 지시하는 정보(merge_flag), 현재 블록이 머지할 수 있는 주변 블록의 움직임 정보로서 좌측 블록의 움직임 정보와 상측 블록의 움직임 정보가 있을 때 어느 주변 블록의 움직임 정보에 머지할 것인지를 지시하는 정보(merge_left_flag), 머지 후보 리스트의 움직임 정보 중에서 어느 움직임 정보를 이용하여 머지할 것인지를 지시하는 정보(merge_idx), 머지 후보 리스트 등을 포함한다.

[0067] 머지에 관한 정보 중에서 merge_flag는 머지를 적용하는 경우에 반드시 필요한 정보이다. 머지에 관한 정보 중에서 merge_left_flag, merge_idx 등은 항상 필요한 정보는 아니며 조건부로 필요한 정보라고 할 수 있다.

[0068] 예컨대, 현재 블록의 머지에 이용할 수 있는 주변 블록의 움직임 정보로서 서로 다른 2개의 움직임 정보(현재 블록의 좌측에 있는 블록의 움직임 정보와 현재 블록의 상측에 있는 블록의 움직임 정보)가 존재하는 경우에만 merge_left_flag를 사용하여 현재 블록의 움직임 정보가 좌측 블록의 움직임 정보와 일치하는지 상측 블록의 움직임 정보와 일치하는지를 지시할 수 있다.

[0069] 또한, 머지 후보 리스트에 존재하는 움직임 정보의 개수가 2개 이상인 경우에만 merge_idx를 이용하여, 현재 블록의 움직임 정보로서 어떤 움직임 정보를 이용할 것인지를 지시할 수 있다.

[0070] 머지를 수행하는 경우에는 상술한 바와 같이 많은 머지 정보가 전달되게 된다. 또한, 현재 PU에 대하여 머지를

수행하는 경우에는 현재 CU에 대하여 머지를 수행하는 경우보다 더 많은 머지 정보를 필요로 하기도 한다.

[0071] 이 경우에, 머지 정보를 모두 명시적으로 전달하지 않고, 머지 정보의 일부 또는 전부를 묵시적으로 유추할 수 있다면, 머지 정보의 전송량을 줄일 수 있다. 이와 관련하여, 다음과 같은 두 가지 방법 중 적어도 하나를 이용하여 전송 정보량을 줄일 수 있다.

[0072] ① 머지 정보의 일부 또는 전부의 전송을 생략하여 정보량을 줄이는 방법 - 현재 블록에 머지가 적용될 가능성이 없는 경우에는 머지 정보의 전송을 생략할 수 있다.

[0073] ② 머지 정보의 부호화 효율을 향상시키는 방법 - 블록의 모양과 위치에 따라서 merge_left_flag의 의미를 변경해서 엔트로피 부호화의 성능을 높일 수 있다. 예컨대, PU 파티션의 모양과 PU의 위치에 따라서 merge_left_flag의 의미를 변경할 수 있다. 또한, 블록의 모양과 위치에 따라서 merge_idx의 의미를 변경하여 엔트로피 부호화의 성능을 높일 수 있다. 예컨대, PU 파티션의 모양과 PU의 위치에 따라서 merge_idx의 의미를 변경할 수 있다.

[0074] 도 3은 본 발명이 적용되는 시스템에서 부호화기와 복호화기의 예측에 관한 동작의 일 예를 개략적으로 설명하는 순서도이다.

[0075] 도 3을 참조하면, 부호화기 또는 복호화기는 우선 주변 블록의 움직임 정보를 획득한다(S310). 현재 블록의 주변 블록 중 현재 블록의 움직임 정보를 예측하는데 이용될 수 있는 움직임 정보를 가지는 주변 블록들을 탐색하고 움직임 정보를 획득하도록 할 수 있다. 또한, 현재 블록이 움직임 정보를 이용할 수 있는 주변 블록의 위치를 미리 정해두고, 상기 정해진 주변 블록의 움직임 정보만을 현재 블록의 움직임 정보를 예측하는데 이용하도록 할 수도 있다.

[0076] 부호화기 또는 복호화기는 현재 블록의 움직임 정보로서 이용되거나 현재 블록의 움직임 정보를 예측하는데 이용될 수 있는 움직임 정보의 후보 개수를 판별하고, 현재 블록이 움직임 정보의 후보와 머지할 가능성을 판별한다(S320). 부호화기 또는 복호화기는 현재 블록의 주변 블록들에 대해서, 현재 블록의 움직임 정보를 예측하는데 이용할 수 있는 주변 블록의 움직임 정보 개수를 판별한다. 또한, 부호화기 또는 복호화기는 현재 블록에 대하여 머지를 적용할 가능성을 판단한다.

[0077] 상술한 단계를 기반으로 부호화기는 현재 블록의 머지를 결정하고, 머지를 수행하며, 필요한 머지 정보를 부호화하여 복호화기에 전송한다. 복호화기는 부호화기로부터 수신한 머지 정보를 기반으로 상술한 바와 같이 현재 블록의 머지를 결정하고, 머지를 수행한다

[0078] 상술한 움직임 정보의 획득과 판별 단계 그리고 이를 이용한 현재 블록의 머지 등은 부호화기와 복호화기 내의 특정 모듈들, 예컨대, 움직임 예측부, 움직임 보상부 등에서 수행될 수 있다.

[0079] 도 4는 본 발명이 적용되는 시스템에서 부호화기와 복호화기의 예측에 관한 동작의 다른 예를 개략적으로 설명하는 순서도이다.

[0080] 도 4를 참조하면, 부호화기 또는 복호화기는 우선 주변 블록의 움직임 정보를 획득한다(S410). 현재 블록의 주변 블록 중 현재 블록의 움직임 정보를 예측하는데 이용될 수 있는 움직임 정보를 가지는 주변 블록들을 탐색하고 움직임 정보를 획득하도록 할 수 있다. 또한, 후보 블록으로서 이용할 수 있는 주변 블록의 위치를 미리 정해두고, 상기 정해진 주변 블록의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 이용하도록 할 수도 있다.

[0081] 부호화기 또는 복호화기는 현재 블록에 대해 움직임 정보의 머지 가능성을 판별한다(S420).

[0082] 상술한 단계를 기반으로 부호화기는 현재 블록의 머지를 결정하고, 머지를 수행하며, 필요한 머지 정보를 부호화하여 복호화기에 전송한다. 복호화기는 부호화기로부터 수신한 머지 정보를 기반으로 상술한 바와 같이 현재 블록의 머지를 결정하고, 머지를 수행한다

[0083] 상술한 움직임 정보의 획득과 판별 단계 그리고 이를 이용한 현재 블록의 머지 등은 부호화기와 복호화기 내의 특정 모듈들, 예컨대, 움직임 예측부, 움직임 보상부 등에서 수행될 수 있다.

[0084] 한편, 도 3 및 도 4에서 상술한 머지 방법은 블록 단위로 적용될 수 있다. 따라서, 현재 블록에 대하여 머지를 적용할 것인지를 결정한 후, 결정에 따라 머지를 수행한다. 예컨대, 머지는 현재 CU에 적용될 수도 있고, 현재 PU에 적용될 수도 있다. 하나의 CU는 2 개 또는 4 개의 PU로 나뉘어 부호화될 수 있으며, CU 자체로 부호화될 수 있다.

[0085] 도 5는 하나의 CU가 PU로 나뉘는 일 예를 개략적으로 설명하는 도면이다.

[0086] 도 5(a)는 하나의 CU(510)를 개략적으로 도시한 것이다.

[0087] 도 5(b)는 하나의 CU가 두 개의 PU로 분할되는 일 예를 도시하고 있다. 예컨대, CU(520)는 CU(520)와 폭이 같고 높이가 절반인 두 개의 PU(525, 530)로 나뉠 수 있다. 또한, CU(535)는 CU(535)와 높이가 같고 폭이 절반인 두 개의 PU(540, 545)로 나뉠 수 있다.

[0088] 도 5(c)는 하나의 CU가 네 개의 PC로 분할되는 일 예를 도시하고 있다. 예컨대, CU(550)는 폭과 높이가 CU(550)의 절반인 네 개의 PU(555, 560, 565, 570)로 나뉠 수 있다.

[0089] 도 6은 하나의 CU로부터 나뉘어질 수 있는 PU의 여러 형태를 개략적으로 나타낸 도면이다.

[0090] 도 6을 참조하면, 하나의 CU 블록이 그 자체로 부호화되는 경우의 PU(610)를 “PART_2Nx2N”으로 나타낸다. 여기서 N은 0보다 큰 정수로서, 2N은 픽셀의 개수이며, 해당 블록의 폭 및/또는 높이를 나타낸다.

[0091] 하나의 CU가 2개의 PU로 부호화될 경우, PU의 형태에 따라 “PART_2NxN”인 두 개의 PU(620), “PART_Nx2N”인 두 개의 PU(630), “PART_2NxnU”인 두 개의 PU(650), “PART_2NxnD”인 두 개의 PU(660), “PART_nLx2N”인 두 개의 PU(670), “PART_nRx2N”인 두 개의 PU(680) 등으로 나뉘어 질 수 있다. 여기서 U, D, L, R은 각각 비대칭형 모양을 대칭형과 구분하기 위해 N 대신 사용된 표현이다.

[0092] 하나의 CU가 4개의 PU로 부호화될 경우에는 “PART_NxN”인 네 개의 PU(640)로 나뉠 수 있다.

[0093] 한편, CU가 PU로 나뉜 상술한 바와 같은 파티션(partition) 형태에 따라서, 현재 블록의 머지 대상을 결정하는 방법을 생각할 수 있다.

[0094] 도 7은 CU로부터 나뉜 두 PU의 머지 대상에 관하여 개략적으로 설명하는 도면이다. 도 7에서는, 현재 CU(700)가 두 PU(710, 720)로 나뉘며, 현재 블록(700)의 인접 블록으로서, 좌측의 두 블록(740, 750)과 상측의 블록(730)이 존재하는 경우를 예로서 설명하고 있다.

[0095] 도 7의 예에서 주변 블록으로부터 획득할 수 있는 움직임 정보의 개수는 2개일 수도 있고, 1개일 수도 있고, 얻을 수 있는 정보가 없을 수도 있다.

[0096] 움직임 정보를 얻을 수 없는 경우는, 주변의 블록(730, 740, 750)이 모두 인트라 예측 블록인 경우로서, 주변 블록으로부터 얻을 수 있는 가용 움직임 정보의 개수가 0이다. 이 경우에는 현재 블록의 예측을 위해 머지를 적용할 수 없으며, 따라서 복호화기에 머지 정보를 전송할 필요가 없게 된다.

[0097] 주변 블록으로부터 얻을 수 있는 움직임 정보가 1개인 경우는, 좌측의 블록(740 750)과 상측 블록(730) 중 어느 한 쪽이 인트라 예측 블록인 경우이다. 인트라 예측 블록으로부터는 움직임 정보를 획득할 수 없으므로, 가용 움직임 정보는 하나가 된다. 따라서, 머지 정보로서 요구되는 것은 머지 여부를 나타내는 merge_flag가 된다. 현재 블록이 머지를 하게 된다면, 현재 블록의 움직임 정보로서 이용할 수 있는 움직임 정보는 하나밖에 없기 때문에 어떤 움직임 정보가 머지의 대상이 되는지를 별도로 지시할 필요가 없다. 따라서, merge_idx, merge_left_flag 등의 다른 정보는 사용하거나 전송될 필요가 없다.

[0098] 주변 블록으로부터 얻을 수 있는 움직임 정보가 2개인 경우로서, 좌측 블록(740, 750) 중 한 블록의 움직임 정보와 상측 블록(730)의 움직임 정보를 얻는 경우를 생각할 수 있다. 이 경우에 얻어낸 두 움직임 정보가 서로 같은 경우는 주변 블록으로부터 얻을 수 있는 움직임 정보가 1개인 경우와 동일하게 처리될 수 있다. 주변 블록으로부터 얻어낸 두 움직임 정보가 서로 다른 경우에는 merge_flag와 함께 merge_left_flag 등의 정보가 머지 정보로서 요구된다.

[0099] 한편, 도 7에 도시한 바와 같이 현재 블록(700)의 좌측 전체 영역(740, 750)과 상측 전체 영역(730)으로부터 머지 대상이 되는 움직임 정보를 얻어내지 않고, 현재 블록의 머지에 이용할 움직임 정보(후보 움직임 정보)를 주변 영역들 중에서 정해진 블록으로부터만 얻어내는 방법을 고려할 수도 있다.

[0100] 도 8은 정해진 위치의 블록으로부터 후보 움직임 정보를 얻어내는 방법을 개략적으로 도시한 것이다. 도 8을 참조하면, 현재 블록(X)는 상측 주변 영역 중 좌상단 블록(E), 상측 블록(B), 우상측 블록(C), 좌측 주변 영역 중 좌측 블록(A), 좌하단 블록(D)으로부터 후보 움직임 정보를 얻어 내서 머지 할 수 있다. 또한, 동일 위치 블록(X’) 내 소정의 블록(M)과 동일 위치 블록(X’)의 인접 블록(H)로부터 후보 움직임 정보를 얻어내서 머지할 수도 있다. 이때, 동일 위치 블록은 참조 영상 내의 블록일 수 있다.

[0101] 정해진 위치의 주변 블록을 현재 블록의 머지에 대한 후보 블록으로 사용하는 경우에, 부호화기는 머지 모드에서 사용할 머지 후보 리스트를 구성하고 현재 블록에 대한 움직임 정보를 머지 후보 리스트의 움직임 정보 후보를 기반으로 설정할 수 있다.

[0102] 예컨대, 현재 블록(PU 또는 CU)의 주변 블록으로부터 움직임 정보 후보를 설정할 수 있다. 이때, 시간적(temporal) 움직임 정보 후보를 사용하기 위한 참조 인덱스를 설정하고, 시간적으로 유추된 움직임 정보를 머지 움직임 정보로 설정할 수 있다. 여기서 시간적으로 유추된 움직임 정보는 동일 위치(co-located) 블록의 움직임 정보를 의미한다.

[0103] 머지 후보 리스트는 도 8의 주변 블록(A, B, C, D, E)과 동일 위치의 후보 블록에 대하여 소정의 순서, 예컨대, A→B→C→D→E→동일위치 블록의 순서로 해당 블록의 움직임 정보가 현재 블록의 머지에 이용될 수 있는지를 판단한다. 이용 가능한 경우에는 해당 블록의 움직임 정보를 머지 후보 리스트에 입력한다.

[0104] 머지 후보 리스트 내에 동일한 움직임 정보가 존재하는 경우에는, 동일한 움직임 정보들 중에서 하나만 머지 후보 리스트에 남겨둔다.

[0105] 머지 후보 리스트 내의 움직임 정보의 수는 소정의 개수로 설정할 수 있다. 예컨대, 머지 후보 리스트 내의 움직임 정보의 수를 최대 5개가 되도록 설정할 수 있다.

[0106] 현재 픽처가 B 픽처이거나 현재 슬라이스가 B 슬라이스인 경우에는, 양 방향 머지 후보를 설정하거나 스케일되지 않은(non-scaled) 양 방향 머지 후보를 설정할 수 있다.

[0107] 또한, 제로 (0, 0) 움직임 정보를 머지 후보로 설정할 수 있다.

[0108] 머지 후보 리스트에서, 머지 인덱스(merge_idx)가 지시하는 움직임 정보를 현재 블록의 움직임 정보로 설정할 수 있다.

[0109] 현재 블록의 주변 블록으로부터 움직임 정보 후보를 설정하는 방법을 좀 더 구체적으로 설명한다. 움직임 정보 후보를 설정하기 위해, 현재 블록의 좌상단 위치, 현재 블록의 너비와 높이에 관한 정보, 현재 블록을 지시하는 정보가 입력된다. 이때, 현재 블록은 PU일 수 있다. 이하, 설명의 편의를 위해 현재 블록이 PU인 경우를 예로서 설명한다.

[0110] 현재 PU의 좌상단 위치는 (xP, yP)와 같이 주어질 수 있다. 현재 PU의 높이와 너비를 지시하는 변수는 (nPSW, nPSH)와 같이 나타낼 수 있다. 또한, 현재 CU 내에 있는 PU 파티션 중에서 현재 PU를 지시하는 인덱스는 PartIdx와 같이 나타낼 수 있다.

[0111] 하나의 CU가 2개의 PU로 분리될 경우, 각각의 PU에 대한 PartIdx는 '0' 또는 '1'의 값을 가질 수 있다. 예컨대, 좌측 또는 상측 PU에 대한 PartIdx의 값을 0, 우측 또는 하측 PU에 대한 PartIdx의 값을 1로 설정할 수 있다. 하나의 CU가 4개의 PU로 분리될 경우, 각각의 PU에 대한 PartIdx는 '0', '1', '2' 또는 '3'의 값을 가질 수 있다. 예컨대, 좌상의 PU에 대한 PartIdx 값을 0, 우상의 PU에 대한 PartIdx 값을 1, 좌하의 PU에 대한 PartIdx 값을 2, 우하의 PU에 대한 PartIdx 값을 3으로 설정할 수 있다.

[0112] 움직임 정보 후보를 설정하는 단계에서는 다음과 같은 네 가지 정보-주변 PU의 사용 가능 여부를 알려주는 변수(availableFlagN), 주변 PU의 참조 인덱스(refIdxLXN), 주변 PU에 대한 예측 리스트 이용 가능 여부를 알려주는 변수(predFlagLXN) 그리고 주변 PU에 대한 움직임 정보(예컨대, 움직임 벡터, mvLXN)-가 출력될 수 있다. 여기서, N는 도 8에서 주변 PU A, B, C, D, E 중 어느 하나를 지시하는 값으로서, 상기 각 변수 또는 인덱스는 N이 지시하는 PU에 대한 정보를 나타낸다. 여기서 X는 0 혹은 1로 대체될 수 있다. 예를 들어 X가 0일 경우에는 상기 네 가지 정보 중 세 가지 정보는 mvL0N, refIdxL0N, predFlagL0N가 되며 이는 L0 움직임 정보를 의미할 수 있다. 또한, X가 1일 경우에는 mvL1N, refIdxL1N, predFlagL1N가 되며 이는 L1 움직임 정보를 의미할 수 있다. 이때, L0는 참조 영상 리스트 0 (reference picture list 0)을 의미하며, L1은 참조 영상 리스트 1 (reference picture list 1)을 의미한다.

[0113] 현재 블록과 현재 PU 주변의 PU(공간적 PU, 도 8의 A, B, C, D, E)가 merge 움직임 후보로 사용 가능한 지를 판단하기 위해서 아래와 같은 순서가 수행된다. (아래에서, N은 도 8의 A, B, C, D, E로 대체될 수 있으며, (xN, yN) 은 (xP-1, yP+nPSH), (xP-1, yP+nPSH-1), (xP+nPSW, yP-1), (xP+nPSW-1, yP-1), (xP-1, yP-1)로 대체될 수 있다.)

[0114] 아래 조건 중에서 하나라도 '참' 이면, availableFlagN 이 '0'으로 설정되고 mvLXN은 '0,0'로 설정될 수

있다.

[0115] (1) N이 E이고 availableFlagD + availableFlagA + availableFlagC + availableFlagB이 '4'이다.

[0116] (2) [xN, yN]의 위치에 있는 PU가 사용 불가능하거나 혹은 인트라 모드로 부호화되었다.

[0117] (3) 현재 PU가 "PART_2NxN" 혹은 "PART_2NxnU" 혹은 "PART_2NxnD"이고, PartIdx가 '1'이고, PartIdx가 '0'에 해당하는 PU(xP, yP-1)와 주변 PU(xN, yN)가 아래 (a)~(c)의 조건과 같이 동일한 움직임 정보를 가진다.

[0118] (a) mvLX[xP, yP-1] == mvLX[xN, yN]

[0119] (b) refIdxLX[xP, yP-1] == refIdxLX[xN, yN]

[0120] (c) predFlagLX[xP, yP-1] == predFlagLX[xN, yN]

[0121] (4) 현재 PU가 "PART_Nx2N" 혹은 "PART_nLx2N" 혹은 "PART_nRx2N"이고, PartIdx가 '1'이며, PartIdx가 '0'에 해당하는 PU(xP-1, yP)와 주변 PU(xN, yN)가 아래 (a)~(c)의 조건과 같이 동일한 움직임 정보를 가진다.

[0122] (a) mvLX[xP-1, yP] == mvLX[xN, yN]

[0123] (b) refIdxLX[xP-1, yP] == refIdxLX[xN, yN]

[0124] (c) predFlagLX[xP-1, yP] == predFlagLX[xN, yN]

[0125] (5) 현재 PU가 "PART_NxN"이고, PartIdx가 '3'이고, PartIdx가 '2'에 해당하는 PU(xP-1, yP)와 PartIdx가 '0'에 해당하는 PU(xP-1, yP-1)가 아래 (a)~(c)의 조건과 같이 동일한 움직임 정보를 가지고 있고,

[0126] (a) mvLX[xP-1, yP] == mvLX[xP-1, yP-1]

[0127] (b) refIdxLX[xP-1, yP] == refIdxLX[xP-1, yP-1]

[0128] (c) predFlagLX[xP-1, yP] == predFlagLX[xP-1, yP-1]

[0129] PartIdx가 '1'에 해당하는 PU(xP, yP-1)와 주변 PU(xN, yN)가 아래 (a)~(c)의 조건과 같이 동일한 움직임 정보를 가진다.

[0130] (a) mvLX[xP, yP-1] == mvLX[xN, yN]

[0131] (b) refIdxLX[xP, yP-1] == refIdxLX[xN, yN]

[0132] (c) predFlagLX[xP, yP-1] == predFlagLX[xN, yN]

[0133] (5) 현재 PU가 "PART_NxN"이고, PartIdx가 '3'이며, PartIdx가 '1'에 해당하는 PU(xP, yP-1)와 PartIdx가 '0'에 해당하는 PU(xP-1, yP-1)가 아래 (a)~(c)의 조건과 같이 동일한 움직임 정보를 가지고 있고,

[0134] (a) mvLX[xP, yP-1] == mvLX[xP-1, yP-1]

[0135] (b) refIdxLX[xP, yP-1] == refIdxLX[xP-1, yP-1]

[0136] (c) predFlagLX[xP, yP-1] == predFlagLX[xP-1, yP-1]

[0137] PartIdx가 '2'에 해당하는 PU(xP-1, yP)와 주변 PU(xN, yN)가 아래 (a)~(c)의 조건과 같이 동일한 움직임 정보를 가진다.

[0138] (a) mvLX[xP-1, yP] == mvLX[xN, yN]

[0139] (b) refIdxLX[xP-1, yP] == refIdxLX[xN, yN]

[0140] (c) predFlagLX[xP-1, yP] == predFlagLX[xN, yN]

[0141] 상술한 (1) 내지 (5)가 '참(true)'에 해당하는 경우가 아니면, availableFlagN은 '1'로 설정되며, mvLXN, refIdxLXN, predFlagLXN에는 각각 mvLX[xN, yN], refIdxLX[xN, yN], predFlagLX[xN, yN]이 설정된다.

[0142] 도 9는 PU에 적용되는 머지의 개념을 개략적으로 설명하는 도면이다.

[0143] 도 9를 참조하면, 하나의 CU로부터 도 6에서 설명한 파티션 형태를 가지는 PU들이 나뉘어지는 것을 볼 수 있다.

각각의 PU들은 도 9에 도시된 바와 같이 머지될 수 있다. 예컨대, PART_2Nx2N의 경우에는 전체 블록이 인터 모드로 예측되어 부호화될 수도 있고, 전체 블록에 머지가 적용되어 부호화될 수도 있다. 여기서 인터 모드란, 인터 예측 모드 중 머지 모드를 제외한 나머지 방법에 의해 인터 예측이 수행되는 모드를 말한다.

[0144] 또한, 하나의 CU가 복수의 PU로 나뉜 나머지 경우에 대해서도, 도시된 바와 같이 각 파티션, 즉 CU로부터 나뉜 각 블록에 대해 머지가 적용될 수 있다. 이때, PU로 나뉜 경우에도 도시된 바와 같이 CU 레벨에서 머지가 수행될 수 있다.

[0145] 앞서 언급한 바와 같이, 머지에 이용될 수 있는 후보 움직임 정보의 개수와 머지가 이용될 수 있는지를 CU로부터 분할된 PU의 파티션 형태를 기반으로 판별할 수 있다.

[0146] 구체적으로, 현재 블록과 주변 블록이 동일한 CU 내에 있는 경우에, 현재 블록의 움직임 정보는 주변 블록의 움직임 정보에 머지하지 않는다. 여기서, 현재 블록과 주변 블록이 동일한 CU 내에 있다는 것은, 현재 블록과 주변 블록이 하나의 CU 내에 있는 두 파티션이라는 것을 의미한다.

[0147] 동일한 CU 내에 두 PU(제1 PU(PartIdx=0) 및 제2 PU(PartIdx=1))가 있는 경우에, 제1 PU와 제2 PU는 서로 동일한 움직임 정보를 가지지 않는다. 따라서, 제1 PU와 제2 PU 사이에 머지가 적용되지 않게 된다. 즉, 제1 PU 움직임 정보 및 제1 PU와 동일한 움직임 정보는 제2 PU의 머지 후보 리스트에 추가되지 않도록 할 수 있다.

[0148] 예컨대, 도 5(B)와 같이, 하나의 CU가 2개의 PU로 나누어지는 경우, 두 PU는 서로 다른 움직임 정보를 가진다. 도 5(B)와 같은 경우에, 하나의 CU로부터 2개의 PU를 만들어 냈음에도 2개의 PU가 동일한 움직임 정보를 가지게 된다면, 차라리 2개의 PU가 아닌 하나의 CU로 부호화 과정을 진행하는 것이 더 효율적이라고 할 수 있다. 따라서, 하나의 CU 내 2개의 PU가 있는 경우에는, 이 2개의 PU가 서로 다른 움직임 정보를 가진다.

[0149] CU가 네 개의 파티션으로 나뉘는 경우, 예컨대 CU가 네 개의 PU로 나뉘는 경우에도 각 PU 간에 동일한 움직임 정보를 가지지 않는 PU 사이에는 머지가 적용되지 않게 된다.

[0150] 예컨대, 도 5(C)와 같이, 하나의 CU가 4개의 PU(제1 PU(PartIdx=0), 제2 PU(PartIdx=1), 제3 PU(PartIdx=2), 및 제4 PU(PartIdx=3))로 나누어지는 경우, 4 개의 PU가 모두 동일한 움직임 정보를 가질 수는 없다. 만약 하나의 CU 내에 있는 4개의 PU들이 동일한 움직임 정보를 가지게 된다면, 상술한 바와 같이 차라리 CU를 PU로 나누지 말고 CU 레벨에서 부호화하는 것이 더 효율적이다. 따라서, 하나의 CU가 4개의 PU로 나누어진다면, 이 4개의 PU 중 적어도 하나는 상이한 움직임 정보를 가지게 된다. 즉, 제1 PU, 제2 PU, 제3 PU가 모두 동일한 움직임 정보를 가지면 해당 움직임 정보와 동일한 움직임 정보는 제4 PU의 머지 후보 리스트에 추가되지 않도록 할 수 있다.

[0151] 도 10은 CU가 4개의 PU로 나뉘어진 경우에, 각 PU의 움직임 정보에 관하여 개략적으로 설명하는 도면이다. 도 10을 참조하면, 하나의 CU가 네 개의 PU(1010, 1020, 1030, 1040)로 나뉘어 있다.

[0152] 도 10에서 PU(1010)의 움직임 정보와 PU(1030)의 움직임 정보가 같을 경우에, PU(1040)는 PU(1020)의 움직임 정보와 동일한 움직임 정보를 가질 수 없다. 만일 PU(1010)의 움직임 정보와 PU(1030)의 움직임 정보가 같고, PU(1040)와 PU(1020)의 움직임 정보가 같다면, CU를 4개의 PU가 아니라, PU(1010)와 PU(1030)의 블록과 PU(1020)와 PU(1040)의 블록으로 나뉘는 PU 파티션을 가지고 부호화하는 게 더 효율적이다. 따라서, 도 10에서 PU(1010)의 움직임 정보와 PU(1030)의 움직임 정보가 같을 경우, PU(1040)는 PU(1020)의 움직임 정보와는 다른 움직임 정보를 가져야만 한다. 즉, PU(1020)의 움직임 정보는 PU(1240)의 머지 후보 리스트에 추가되지 않도록 할 수 있다.

[0153] 마찬가지로 도 10에서 PU(1010)의 움직임 정보와 PU(1020)의 움직임 정보가 같을 경우에, PU(1040)는 PU(1030)의 움직임 정보와 동일한 움직임 정보를 가질 수 없다. 만일 PU(1010)의 움직임 정보와 PU(1020)의 움직임 정보가 같고, PU(1040)와 PU(1030)의 움직임 정보가 같다면, CU를 4개의 PU가 아니라, PU(1010)와 PU(1020)의 블록과 PU(1030)와 PU(1040)의 블록으로 나뉘는 두 PU 파티션을 가지고 부호화하는 게 더 효율적이다. 따라서, 도 10에서 PU(1010)의 움직임 정보와 PU(1020)의 움직임 정보가 같을 경우, PU(1040)는 PU(1030)의 움직임 정보와는 다른 움직임 정보를 가져야만 한다. 즉, PU(1030)의 움직임 정보는 PU(1040)의 머지 후보 리스트에 추가되지 않도록 할 수 있다.

[0154] 도 11은 도 10에서 설명한, CU로부터 나뉘어지는 PU 간의 관계를 개략적으로 나타낸 것이다.

[0155] 도 11(A)을 참조하면, CU(1100)가 네 개의 PU(1~4)로 나누었는데, PU(1)와 PU(3)의 움직임 정보가 같고, PU(2)와 PU(4)의 움직임 정보가 같다면, 부호화 효율을 고려할 때, 상술한 바와 같이 CU(1100)를 두 개의 PU(1110, 1120)로 나누는 것이 더 효율적이다.

[0156] 도 11(B)을 참조하면, CU(1130)가 네 개의 PU(1~4)로 나누었는데, PU(1)와 PU(2)의 움직임 정보가 같고, PU(3)와 PU(4)의 움직임 정보가 같다면, 부호화 효율을 고려할 때, 상술한 바와 같이 CU(1130)를 두 개의 PU(1140, 1150)로 나누는 것이 더 효율적이다.

[0157] 또한, 도 11(C)을 참조하면, CU(1160)가 네 개의 PU(1~4)로 나누었는데, PU(1), PU(2), PU(3), PU(4)의 움직임 정보가 모두 같다면, 부호화 효율을 고려할 때 하나의 CU(1160), 즉 하나의 PU(1170)로 부호화를 진행하는 것이 더 효율적이다.

[0158] 이처럼, 하나의 CU로부터 나뉘어지는 PU들은 서로 다른 움직임 정보를 가지게 되므로, 이를 이용하여 머지를 수행하는데 필요한 정보량을 결정할 수 있다. 즉, 하나의 CU가 2개의 PU로 나누어져 있을 경우, 각 PU 파티션들은 서로 다른 움직임 정보를 가지게 된다. 다시 말하면, 두 PU 파티션(PartIdx=0인 PU 파티션과 PartIdx=1인 PU 파티션)은 동일한 움직임 정보를 가질 수 없다. 따라서, PartIdx=1인 PU 파티션의 머지 후보 리스트에 PartIdx=0인 PU 파티션의 움직임 정보와 동일한 움직임 정보는 추가되지 않도록 할 수 있다.

[0159] 도 12는 하나의 CU(1200)가 두 개의 PU(1210, 1220)로 나뉘어져 있을 때, 현재 PU의 주변 블록 중 현재 블록의 머지 후보 블록이 될 수 있는 블록을 개략적으로 도시한 것이다. 도 12를 참조하면, 상술한 바와 같이, 현재 PU(1220)를 PartIdx=1인 PU 파티션이라고 할 때, 현재 PU(1220)의 주변 블록들인 A1, B1, B0, A0, B2의 움직임 정보들 중에서 PartIdx=0인 좌측의 PU(1210)와 동일한 움직임 정보를 가지는 블록의 움직임 정보는 현재 PU(1220)의 움직임 정보 후보로 사용될 수 없다.

[0160] 따라서, PartIdx=1인 PU 파티션의 머지 후보 리스트에 PartIdx=0인 PU 파티션의 움직임 정보와 동일한 움직임 정보는 추가되지 않도록 할 수 있다. 예를 들어, B1의 움직임 정보가 PartIdx=0인 PU 파티션의 움직임 정보와 동일할 경우, B1의 움직임 정보는 PartIdx=0인 PU 파티션의 머지 후보 리스트에 추가되지 않는다.

[0161] 표 1은 본 발명이 적용되는 시스템에서 머지의 정보량을 결정하는 방법의 일 예를 나타낸 것이다.

표 1

[0162]

상측 / 좌측	동일 CU	다른 CU
동일 CU	머지 불가능 (Merge 정보 불필요)	후보가 하나일 경우 Merge_idx 정보 사용 안 함
다른 CU	후보가 하나일 경우 Merge_idx 정보 사용 안 함	종래의 방식 Merge flag Merge_idx

[0163] 구체적으로, 표 1은 하나의 CU에서 나뉘어진 PU들이 동일한 움직임 정보를 가질 수 있는지에 관해서 상술한 내용을 기반으로 현재 블록에 머지를 적용하는 경우에 전송될 정보의 종류와 정보량을 결정하는 일 예를 나타내고 있다.

[0164] 표 1의 예에서, 주변 블록이 현재 블록(예컨대, PU)과 동일한 CU 내에 있는 블록(예컨대, PU)인 경우에, 해당 주변 블록은 현재 블록에 대한 머지 후보로 사용되지 않는다. 즉, 해당 주변 블록의 움직임 정보가 현재 블록의 움직임 정보로 사용되지 않도록, 해당 주변 블록의 움직임 정보를 현재 블록의 머지 후보로 사용하지 않는다. 이때, 현재 블록의 움직임 정보를 머지할 움직임 정보를 가지는 후보 블록은 상술한 바와 같이 현재 블록의 파티션 정보를 기반으로 결정될 수 있다.

[0165] 도 13은 주변 블록이 현재 블록과 동일한 CU 내의 다른 블록인 경우에 현재 블록이 머지하는 일 예를 도시한 것이다. 도 13에서는, 현재 블록과 주변 블록이 동일한 CU 내의 서로 다른 PU인 경우를 예로서 설명한다.

[0166] 도 13을 참조하면, 동일한 CU 내에 두 개의 PU가 존재하는 경우(1310)는, PART_2NxN인 PU의 경우, PART_Nx2N인 PU의 경우, PART_2NxnU인 PU의 경우, PART_2NxnD인 PU의 경우, PART_nLx2N인 PU의 경우, PART_nRx2N인 PU의 경

우가 있다.

[0167] 각 PU의 후보 블록이 되는 주변 블록은 CU 내 PU의 파티션 정보로부터 결정될 수 있다. 예컨대, PART_2NxN인 PU의 경우에, 상측의 PU는 상측 주변 블록과 머지할 수 있고, 하측의 PU는 좌측 주변 블록과 머지할 수 있다. PART_Nx2N인 PU의 경우에, 좌측의 PU는 좌측 주변 블록과 머지할 수 있고, 우측의 PU는 상측의 주변 블록과 머지할 수 있다. PART_2NxnU인 PU의 경우에, 상측의 PU는 상측 주변 블록과 머지할 수 있고, 하측의 PU는 좌측 주변 블록과 머지할 수 있다. PART_2NxnD인 PU의 경우 에, 상측의 PU는 상측 주변 블록과 머지할 수 있고, 하측의 PU는 좌측 주변 블록과 머지할 수 있다. PART_nLx2N인 PU의 경우에, 좌측의 PU는 좌측 주변 블록과 머지할 수 있고, 우측의 PU는 상측의 주변 블록과 머지할 수 있다. PART_nRx2N인 PU의 경우에, 좌측의 PU는 좌측 주변 블록과 머지할 수 있고, 우측의 PU는 상측의 주변 블록과 머지할 수 있다.

[0168] 동일 CU 내의 각 PU에 머지가 수행될 것인지가 결정된다. PU가 머지하는지에 관한 정보는 merge_flag에 의해 지시될 수 있다. PU에 머지를 수행할 것인지는 부호화기 혹은 부호화기 내 소정의 모듈(예컨대, 움직임 예측부)에서 결정될 수 있다. 머지를 수행하는 경우에, 머지의 대상 블록은 상술한 바와 같이 CU 내 PU의 파티션 정보에 의해 결정될 수 있다.

[0169] 도 13의 경우에는 CU 내 PU 파티션 정보로부터 판단할 때, PU의 후보 블록이 하나 존재하게 되므로, PU가 머지하는 경우에도 머지할 후보 블록을 지시하는 merge_left_flag 등의 정보를 전송할 필요가 없다. 복호화기는 현재 CU의 파티션 정보를 기반으로 현재 블록의 머지 대상을 특정할 수 있다.

[0170] 도 13에서 PART_2Nx2N인 PU와 PART_NxN인 PU는 머지에 관한 정보, 예컨대 머지 대상에 관한 정보까지 복호화기에 전송한다.

[0171] 도 14는 주변 블록이 현재 블록과 동일한 CU 내의 다른 블록인 경우에 현재 블록이 머지하는 다른 예를 도시한 것이다. 도 14에서는, 현재 블록과 주변 블록이 동일한 CU 내의 서로 다른 PU인 경우를 예로서 설명한다.

[0172] 도 14의 예에서는, 도 13의 예와 달리, PART_2Nx2N인 PU와 PART_NxN인 PU의 경우에는 머지를 적용하지 않도록 함으로써, 복잡도를 줄이고 전송 정보량도 줄일 수 있다.

[0173] 도 14의 예에서도, 동일한 CU 내에 두 개의 PU가 존재하는 경우(1410), 즉 PART_2NxN인 PU의 경우, PART_Nx2N인 PU의 경우, PART_2NxnU인 PU의 경우, PART_2NxnD인 PU의 경우, PART_nLx2N인 PU의 경우, PART_nRx2N인 PU의 경우에는 도시된 바와 같이, 도 13과 동일하게 CU의 파티션 정보를 기반으로 PU의 움직임 정보로 사용될 움직임 정보 후보를 결정할 수 있다.

[0174] 도 15는 주변 블록이 현재 블록과 동일한 CU 내의 다른 블록인 경우에 현재 블록이 머지하는 또 다른 예를 도시한 것이다. 도 14에서는, 현재 블록과 주변 블록이 동일한 CU 내의 서로 다른 PU이고, CU 단위의 머지가 수행되는 경우를 예로서 설명한다.

[0175] 도 15를 참조하면, CU 단위 머지의 경우에는, PU 파티션(PART_2Nx2N, PART_NxN, PART_2NxN, PART_Nx2N, PART_2NxnU, PART_2NxnD, PART_nLx2N, PART_nRx2N) 별로 2개의 머지 유형을 고려할 수 있다. 도 15의 경우에도, CU 내 PU 파티션에 따른 후보 블록은 도 13 및 도 14의 예와 동일하게 결정될 수 있다.

[0176] 일반적인 예측 모드에서 현재 CU는 PU 단위(Variable block size)로 분리되어 예측이 수행될 수 있지만, CU 단위 머지를 수행하는 경우에는 CU 단위로 merge_flag를 통해 머지를 수행할지 여부를 결정하고, 해당 CU 내의 각 PU에서 해당 CU의 파티션 정보를 통해 머지를 수행할 주변 블록을 결정한다. 이때, merge_flag는 CU 단위로 하나만 전송된다. CU 단위로 merge를 수행하는 것을 결정된 경우, CU 내의 모든 PU에 대해 머지가 수행된다. CU 단위로 머지를 수행하지 않기로 결정된 경우, CU 내의 모든 PU에 대해 머지를 수행하지 않고 일반적인 부호화 방법(머지 이외의 인터 예측)을 수행한다.

[0177] CU 단위 머지의 경우, 해당 CU와 CU 내 PU에 대해서는 Merge_left_flag를 전송하지 않는다. 복호화기는 CU 내의 PU 파티션 정보를 기반으로 CU 내의 각 PU가 머지할 주변 블록을 결정할 수 있다.

[0178] PART_2Nx2N와 PART_NxN인 PU의 경우에는 머지 정보를 복호화기에 전송하고, 복호화기는 수신한 머지 정보를 기반으로 머지를 수행하도록 할 수 있다.

[0179] 한편, 하나의 CU 내에 4개의 PU가 존재하며, 4개의 PU 중에 하나가 현재 블록이고, 4개의 PU 중 일부가 동일한 움직임 정보를 가질 수 있는 경우에는, 현재 블록의 머지 대상을 결정하기 위해 이전 PU(현재 CU 내 다른 PU)의 머지 정보도 함께 고려할 필요가 있다.

[0180] 상술한 바와 같이, CU가 네 개의 PU로 나뉘는 경우에, 4개의 PU에 대한 움직임 정보가 모두 같을 수는 없다. 따라서, 4개의 PU 중 일부가 동일한 움직임 정보를 가질 수 있는 경우에는, 현재 블록의 머지 대상을 결정하기 위해 이전 PU 파티션의 머지 정보도 함께 고려할 필요가 있다.

[0181] 도 10을 참조하면, PU(1010), PU(1020), PU(1030)이 서로 머지되었을 때, PU(1040)은 PU(1010), PU(1020), PU(1030)과 머지할 수 없다. 즉, PU(1040)은 주변 블록과 머지 할 수 없게 되며, PU(1040)에 대한 머지 정보를 전송하지 않아도 복호화기 측에서는 PU(1040)가 PU(1010), PU(1020), PU(1030)과 머지할 수 없다는 것을 판단할 수 있다. 또한, 현재 블록이 속하는 CU(현재 CU)의 파티션 정보(예컨대, PU 파티션 정보)와 주변 블록의 파티션 정보(예컨대, PU 파티션 정보)를 이용해서 현재 블록의 머지 대상을 결정할 수 있다.

[0182] 도 16은 동일 CU 내 두 블록(예컨대, PU)에 대하여, 다른 블록의 머지 정보를 이용하여 현재 블록의 머지 정보를 결정하는 예를 개략적으로 설명하는 도면이다.

[0183] 동일 CU 내에 두 개 이상의 PU가 있는 경우에, 두 번째 이후의 PU에 대한 머지 정보를 부호화할 때는, 동일 CU 내에서 이전에 부호화된 PU의 머지 정보를 이용할 수 있다. 여기서 두 번째 이후의 블록(예컨대, PU)라 함은, 현재 PU가 PART_2NxN 또는 PART_N_2N과 같이 두 개의 PU로 나뉘는 경우에는 우측의 PU 혹은 하측의 PU, 즉 PartIdx=1인 PU일 될 수 있다. 현재 PU가 PART_NxN인 경우에는, PartIdx 값이 1, 2, 3 에 해당하는 PU를 두 번째 이후를 PU라고 할 수 있다.

[0184] 예컨대, 도 16(A)와 같이 CU(1300) 내 두 개의 PU(1610, 1620) 중 PU(PartIdx=0, 1610)가 좌측의 블록(1630)과 머지된 경우에, 현재 블록(PartIdx=1, 1620)에는 머지가 적용되지 않는다. PU(1610)이 좌측 블록(1630)과 머지된 것은 PU(1610)의 움직임 정보가 좌측 블록(1630)의 움직임 정보와 동일하다는 것을 의미한다. 따라서, 현재 블록(1620)의 움직임 정보는 상술한 바와 같이 좌측 블록(1630)의 움직임 정보와 상이하고, 현재 블록은 좌측 블록(1630)에 머지할 수 없다.

[0185] 또한, 도 16(A)와 같이 CU(1600) 내 두 개의 PU(1610, 1620) 중 PU(PartIdx=0, 1610)이 좌측의 블록(1630)과 머지된 경우에, 현재 블록(PartIdx=1, 1620)을 머지 후보 리스트에 좌측의 블록(1630)의 움직임 정보를 추가하지 않을 수 있다. PU(1610)이 좌측 블록(1630)과 머지된다는 것은 PU(1610)의 움직임 정보가 좌측 블록(1630)의 움직임 정보와 동일하다는 것을 의미한다. 따라서, 현재 블록(1620)의 움직임 정보는 상술한 바와 같이 좌측 블록(1630)의 움직임 정보와 상이하고, 현재 블록은 좌측 블록(1630)에 머지할 수 없다.

[0186] 또한, 도 16(B)의 예에서 CU(1640) 내 두 개의 PU(1650, 1660) 중 PU(13650)이 좌측의 블록(1670)과 머지되었다고 한다면, 현재 블록(1660)이 머지할 수 있는 주변 블록은 좌측 블록(1680)이 된다. 이 경우에, 현재 블록(1660)이 머지할 수 있는 공간적인 대상은 좌측 블록(1680) 하나이다. 따라서, 부호화기측에서는, 두 개의 주변 블록이 존재하지만, merge_left_flag 등으로 머지 대상을 지시하지 않고, 단순히 merge_flag만을 복호화기에 전달할 수 있다.

[0187] 이 경우에, 복호화기는 수신한 merge_flag가 현재 블록(1660)이 머지한다는 것을 지시하면, 동일한 CU 내 다른 블록(1650)의 머지 정보를 확인할 수 있다. 다른 블록(1650)이 좌측 블록(1370)에 머지한다는 것을 확인하면, 현재 블록(1660)의 머지 대상을 지시하는 정보(예컨대, merge_left_flag)가 없더라도 현재 블록(1660)의 머지 대상을 특정할 수 있다.

[0188] 한편, 상술한 바와 같이, 현재 블록의 주변 블록 중 특정 위치의 블록들을 현재 블록의 머지 후보 블록으로 미리 정하고, 머지 후보 블록의 움직임 정보를 이용하여 현재 블록의 움직임 정보를 예측하도록 할 수 있다.

[0189] 도 17은 머지 후보 블록을 소정의 주변 블록으로 한정한 경우에, 다른 블록의 움직임 정보를 이용하여 현재 블록의 움직임 정보를 결정하는 일 예를 개략적으로 설명하는 도면이다. 도 17에서는 현재 CU(1700) 내에 두 개의 블록(1705, 1710)이 존재하는 경우를 예로서 설명하고 있다. 도 17에서 두 개의 블록(1705, 1710)은 PART_Nx2N, PART_nLx2N 혹은 PART_nRx2N인 블록이며, PU일 수 있다.

[0190] 앞서 설명한 바와 같이, 현재 블록에 대한 공간적인 머지 후보는 현재 블록의 상측에 있는 소정의 주변 블록과 현재 블록의 좌측에 있는 소정의 주변 블록이 될 수 있다.

[0191] 예컨대, 블록(1705)이 머지하는 경우에는, 도 8에서 설명한 바와 같이, 블록(1705)의 좌상측에 있는 주변 블록(1715), 블록(1705)의 상측에 있는 주변 블록(1720), 블록(1705)의 우상측에 있는 주변 블록(1725), 블록(1705)의 좌측에 있는 주변 블록(1740), 블록(1705)의 좌하측에 있는 주변 블록(1745)가 블록(1705)에 대한 공

간적 머지 후보 블록이 될 수 있다.

[0192] 도 16에서 설명한 바와 같이, 현재 CU 내에의 블록들(예컨대, PU들) 중에서 두 번째 이후의 블록에 대해서는 해당 블록(현재 블록)의 머지 정보를 결정하는데 다른 블록의 머지 정보를 이용할 수 있다.

[0193] 예컨대, 도 17에서 블록(1710)이 현재 블록이라고 가정하면, 현재 블록(1710)의 머지 정보는 다른 블록(1705)의 머지 정보를 이용해서 결정할 수 있다. 도 17에서, 현재 블록(1710)의 좌상측에 있는 주변 블록(1720), 현재 블록의 상측에 있는 주변 블록(1730), 현재 블록의 우상측에 있는 주변 블록(1735), 현재 블록의 좌측에 있는 주변 블록(1750), 현재 블록의 좌하측에 있는 주변 블록(1755)이 현재 블록(1710)에 대한 공간적 머지 후보 블록이 될 수 있다.

[0194] 이때, 동일한 CU(1700) 내의 다른 블록(1705)이 머지하는 블록은 현재 블록의 머지 후보 블록이 될 수 없다. 따라서, 도 17에서 블록(1705)가 주변 블록(1720)과 머지한다면, 주변 블록(1720)은 현재 블록(1710)의 머지 후보 블록에서 제외된다. 현재 블록(1710)에 관한 머지 정보(예컨대, merge_idx)는, 다른 블록(1705)가 주변 블록(1720)에 머지한다는 정보를 기반으로 주변 블록(1720)을 현재 블록(1710)의 머지 후보 블록에서 제외해서 생성될 수 있다.

[0195] 도 18은 머지 후보 블록을 소정의 주변 블록으로 한정한 경우에, 다른 블록의 움직임 정보를 이용하여 현재 블록의 움직임 정보를 결정하는 다른 예를 개략적으로 설명하는 도면이다. 도 18에서는 현재 CU(1800) 내에 두 개의 블록(1805, 1810)이 존재하는 경우를 예로서 설명하고 있다. 도 18에서 두 개의 블록(1805, 1810)은 PART_2NxN, PART_2NxnN 혹은 PART_2NxnD인 블록이며, PU일 수 있다.

[0196] 도 18의 예에서 블록(1805)이 머지하는 경우에는, 도 8에서 설명한 바와 같이, 블록(1805)의 주변 블록 중 블록(1815, 1820, 1825, 1830, 1835)가 블록(1805)에 대한 공간적 머지 후보 블록이 될 수 있다.

[0197] 도 18에서 블록(1810)이 현재 블록이라고 가정하면, 현재 블록(1810)의 머지 정보를 다른 블록(1805)의 머지 정보를 이용해서 결정할 수 있다. 도 18에서, 현재 블록(1810)의 좌상측에 있는 주변 블록(1830), 현재 블록의 상측에 있는 주변 블록(1840), 현재 블록의 우상측에 있는 주변 블록(1845), 현재 블록의 좌측에 있는 주변 블록(1850), 현재 블록의 좌하측에 있는 주변 블록(1855)이 현재 블록(1810)에 대한 공간적 머지 후보 블록이 될 수 있다.

[0198] 이때, 동일한 CU(1800) 내의 다른 블록(1805)이 머지하는 블록은 현재 블록(1810)의 머지 후보 블록이 될 수 없다. 따라서, 도 18에서 블록(1805)가 주변 블록(1830)과 머지 한다면, 주변 블록(1830)은 현재 블록(1810)의 머지 후보 블록에서 제외된다. 현재 블록(1810)에 관한 머지 정보(예컨대, merge_idx)는, 다른 블록(1805)이 주변 블록(1830)에 머지한다는 정보를 기반으로 주변 블록(1830)을 현재 블록(1810)의 머지 후보 블록에서 제외해서 생성될 수 있다.

[0199] 도 19는 머지 후보 블록을 소정의 주변 블록으로 한정한 경우에, 다른 블록의 움직임 정보를 이용하여 현재 블록의 움직임 정보를 결정하는 또 다른 예를 개략적으로 설명하는 도면이다. 도 19에서는 현재 CU(1900) 내에 네 개의 블록(1905, 1910, 1915, 1920)이 존재하는 경우를 예로서 설명하고 있다. 도 19에서 네 개의 블록(1905, 1910, 1915, 1920)은 PART_NxN인 블록이며, PU일 수 있다.

[0200] 도 19의 예에서 블록(1905)이 머지하는 경우에는, 도 8에서 설명한 바와 같이, 블록(1905)의 공간적 주변 블록 중 블록(1925, 1930, 1935, 1950, 1955)가 블록(1905)에 대한 머지 후보 블록이 될 수 있다. 블록(1910)이 머지하는 경우에는, 블록(1910)의 공간적 주변 블록 중 블록(1930, 1940, 1945, 1970, 2000)이 블록(1910)에 대한 머지 후보 블록이 될 수 있다. 또한, 블록(1915)이 머지하는 경우에는, 블록(1915)의 공간적 주변 블록 중 블록(1950, 1970, 1975, 1960, 1965)가 블록(1915)에 대한 머지 후보 블록이 될 수 있다. 블록(1920)이 머지하는 경우에는, 블록(1920)의 주변 블록 중 블록(1970, 1980, 1985, 1990, 1995)이 블록(1920)에 대한 공간적 머지 후보 블록이 될 수 있다. 이 중 블록(1985)와 블록(1995)는 해당 블록이 복호화되었다면, 블록(1920)에 대한 공간적 머지 후보 블록이 될 수 있다.

[0201] 도 19에서도 현재 블록의 머지 정보를 다른 블록의 머지 정보를 이용해서 결정할 수 있다.

[0202] 도 19에서, 블록(1920)이 현재 블록이라고 가정하면, 도 10의 경우와 같이 블록(1905), 블록(1910), 블록(1915)가 머지하는 경우에, 현재 블록(1920)의 머지 정보는 블록(1905), 블록(1910), 블록(1915)의 머지 정보를 이용하여 결정될 수 있다. 예컨대, 블록(1905)와 블록(1910)이 주변 블록(1930)에 머지하고, 블록(1915)가 주변 블록(1970) 또는 주변 블록(1975)에 머지하는 경우를 고려할 수 있다. 이 경우에, 동일한 CU(1900) 내의 다른

세 블록(1905, 1910, 1915)는 동일한 움직임 정보를 가지게 된다. 따라서, 현재 블록(1920)은 주변 블록(1970, 1980, 1990)에 머지할 수 없고, 주변 블록(1995)과 주변 블록(1985) 중에서 블록(1905, 1910, 1915)의 움직임 정보와 다른 움직임 정보를 가지는 블록에 머지할 수 있다. 이 중 블록(1985)와 블록(1995)는 해당 블록이 복호화되었다면, 블록(1920)에 대한 공간적 머지 후보 블록이 될 수 있다.

[0203] 이 경우에는 현재 블록(1920)이 머지할 수 있는 후보 블록이 주변 블록(1995)와 주변 블록(1985)의 2개이므로, merge_left_flag로 현재 블록(1920)의 머지 후보 블록을 지시할 수도 있다.

[0204] 도 20은 본 발명이 적용되는 시스템에서 부호화기의 동작을 개략적으로 설명하는 순서도이다.

[0205] 도 20을 참조하면, 부호화기는 현재 블록에 대한 예측을 수행한다(S2010). 현재 블록(현재 PU)에 대한 예측 모드가 인트라가 아닌 경우에, 현재 블록에 대하여 머지 모드를 적용할 수도 있다. 이때, 현재 블록의 움직임 정보, 예컨대 머지 모드를 적용할 것인지, 어떤 주변 블록의 움직임 정보에 머지할 것인지 등을 현재 블록과 동일한 CU 내에 있는 다른 블록(예컨대, 다른 PU)의 움직임 정보를 기반으로 결정할 수도 있다. 구체적인 내용은 도 3 등을 이용하여 상술한 바와 같다.

[0206] 부호화기는 현재 블록에 대한 예측을 수행한 후, 예측 결과에 관한 정보를 엔트로피 부호화한다(S2020). 예컨대, 부호화기는 현재 블록에 대한 예측 블록 및/또는 현재 블록에 대한 예측을 수행하기 필요한 정보 등을 엔트로피 부호화할 수 있다.

[0207] 부호화기는 엔트로피 부호화된 정보를 비트스트림을 통해 복호화기로 전송할 수 있다(S2030).

[0208] 도 21은 본 발명이 적용되는 시스템에서 복호화기의 동작을 개략적으로 설명하는 순서도이다.

[0209] 도 21을 참조하면, 복호화기는 부호화기로부터 수신한 영상 정보를 엔트로피 복호화한다(S2110). 영상 정보는 부호화기로부터 수신되는 비트스트림에 포함되어 복호화기에 전송된다. 영상 정보는 현재 블록에 대한 예측 블록, 레지듀얼 블록, 예측을 위한 정보 등을 포함한다.

[0210] 복호화기는 엔트로피 복호화된 정보를 기반으로 현재 블록에 대한 예측을 수행한다(S2120). 현재 블록이 인터 예측 모드인 경우에 현재 블록의 움직임 정보를 현재 CU의 파티션 정보 등으로부터 유도할 수 있다. 예컨대, 복호화기는 현재 블록에 머지가 적용될 수 있는지, 얼마나 많은 움직임 정보 후보가 있는지, 어떤 움직임 정보 후보에 머지하는지 등과 같은 정보의 일부 또는 전부를 유도할 수 있다. 이 경우, 복호화기는 현재 블록과 동일한 CU 내의 다른 PU에 관한 움직임 정보를 이용하여 현재 블록의 움직임 정보를 유도할 수도 있다. 구체적인 내용은 도 4 등을 이용하여 상술한 바와 같다.

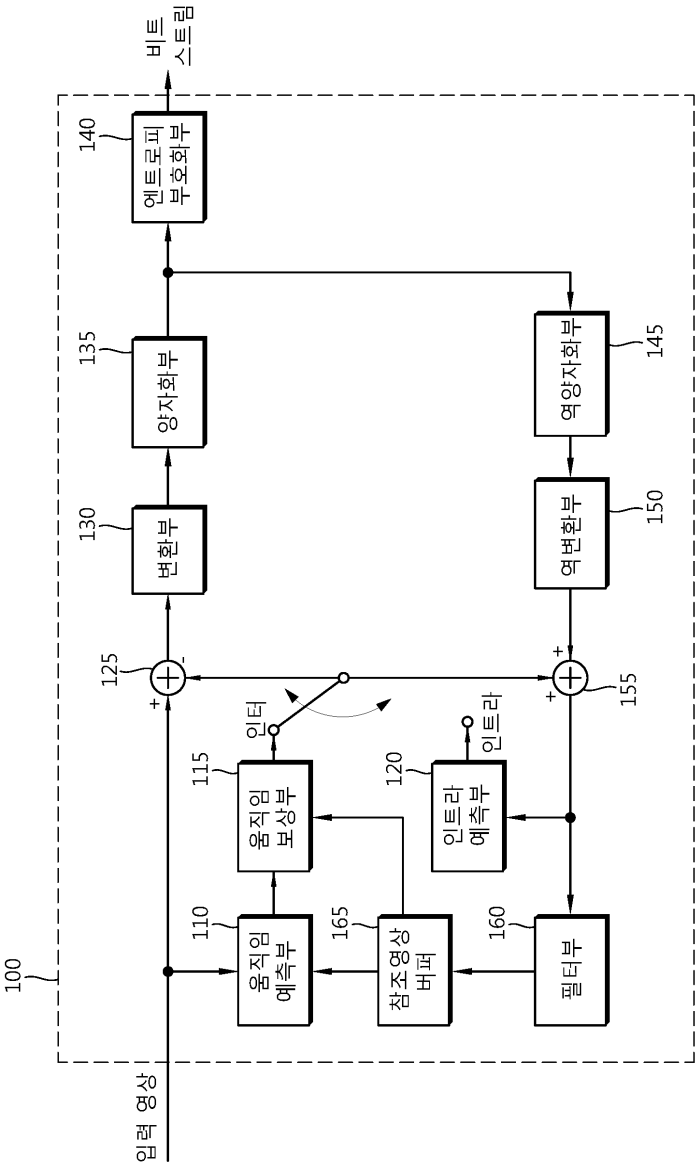
[0211] 복호화기는 예측의 결과를 기반으로 재생 신호를 생성하여 영상을 재생할 수 있다(S2130).

[0212] 상술한 예시적인 시스템에서, 방법들은 일련의 단계 또는 블록으로써 순서도를 기초로 설명되고 있지만, 본 발명은 단계들의 순서에 한정되는 것은 아니며, 어떤 단계는 상술한 바와 다른 단계와 다른 순서로 또는 동시에 발생할 수 있다. 또한, 상술한 실시예들은 다양한 양태의 예시들을 포함한다. 따라서, 본 발명은 이하의 특허청구범위 내에 속하는 모든 다른 교체, 수정 및 변경을 포함한다고 할 것이다.

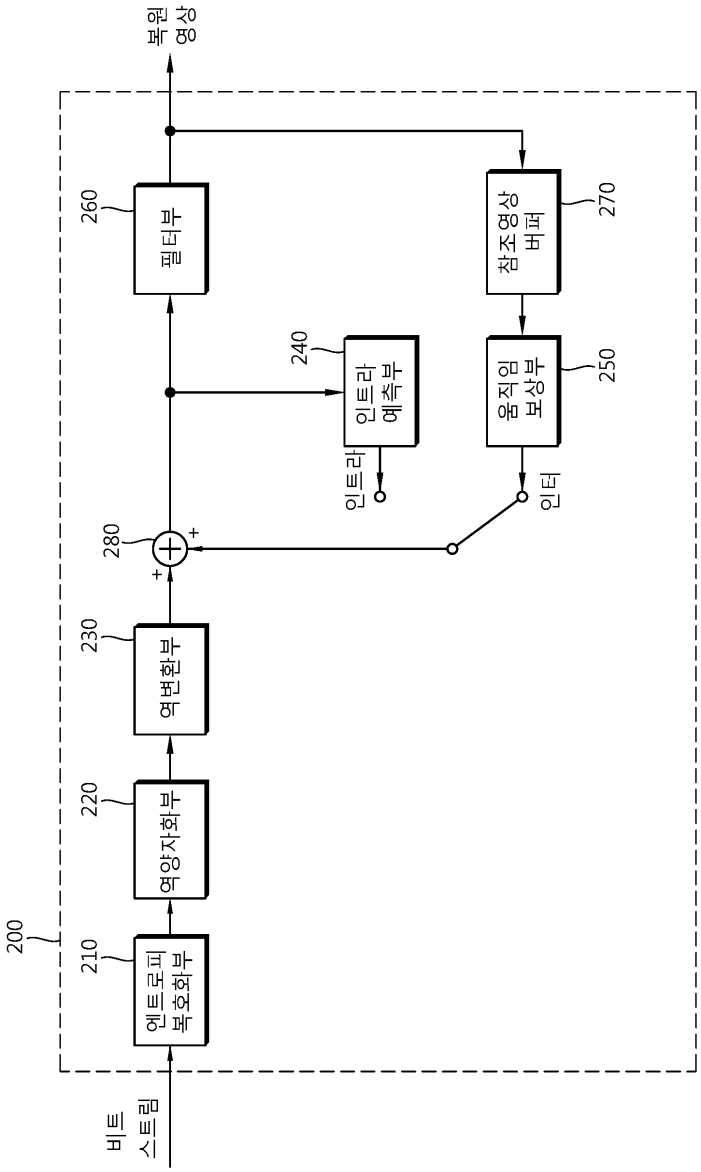
[0213] 지금까지 본 발명에 관한 설명에서 일 구성 요소가 타 구성 요소에 "연결되어" 있다거나 "접속되어"있다고 언급된 때에는, 상기 일 다른 구성 요소가 상기 타 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 상기 두 구성 요소 사이에 다른 구성 요소가 존재할 수도 있다고 이해되어야 한다. 반면에, 일 구성 요소가 타 구성 요소에 "직접 연결되어"있다거나 "직접 접속되어"있다고 언급된 때에는, 두 구성 요소 사이에 다른 구성요소가 존재하지 않는 것으로 이해되어야 한다.

도면

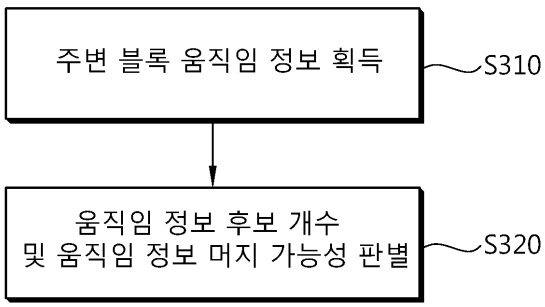
도면1


비트 스트림
엔트로피 부호화부
140
양자화부
135
변환부
130
125
-
+
인터
115
움직임 보상부
110
움직임 예측부
165
참조영상 버퍼
160
필터부
155
+
+
역변환부
150
역양자화부
145
인트라
120
인트라 예측부
100
입력 영상

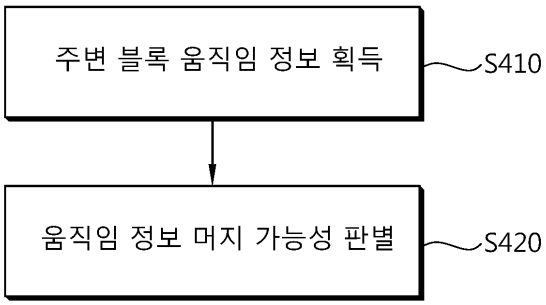
도면2


복원 영상
260
필터부
270
참조영상 버퍼
240
인트라 예측부
250
움직임 보상부
인트라
인터
280
+
+
230
역변환부
220
역양자화부
210
엔트로피 복호화부
200
비트 스트림

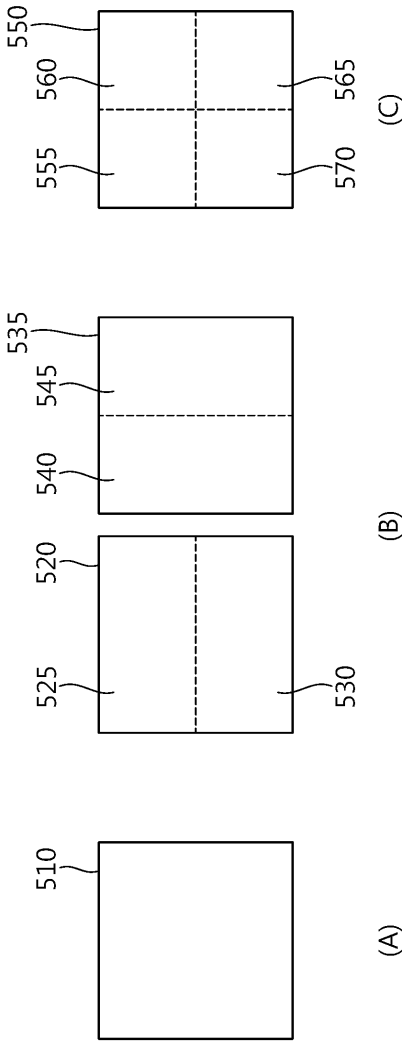
도면3


주변 블록 움직임 정보 획득
S310
움직임 정보 후보 개수
및 움직임 정보 머지 가능성 판별
S320

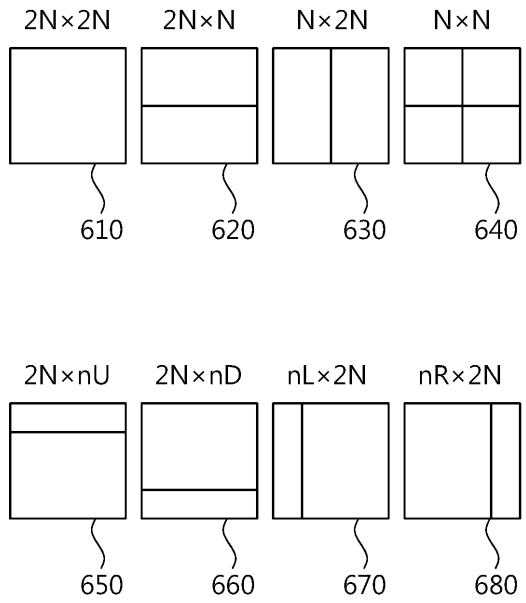
도면4


주변 블록 움직임 정보 획득
S410
움직임 정보 머지 가능성 판별
S420

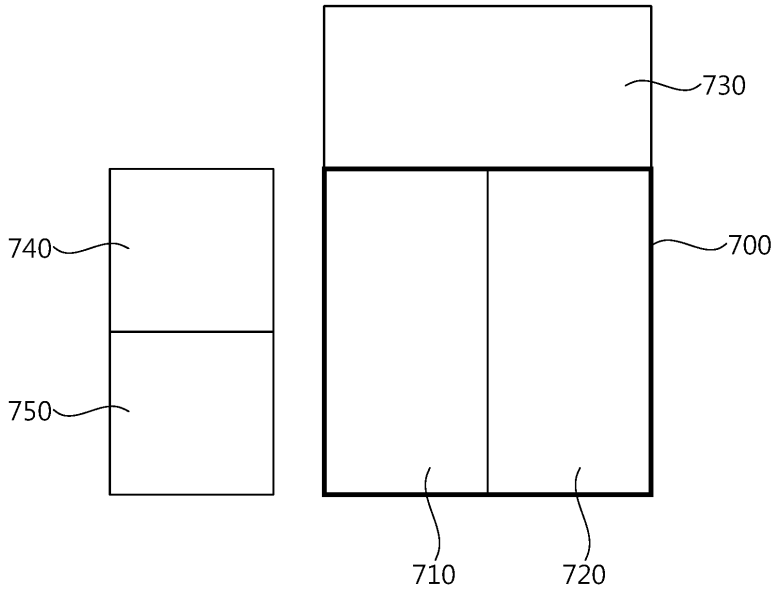
도면5


550
560
565
555
570
(C)
535
545
540
(B)
520
525
530
510
(A)

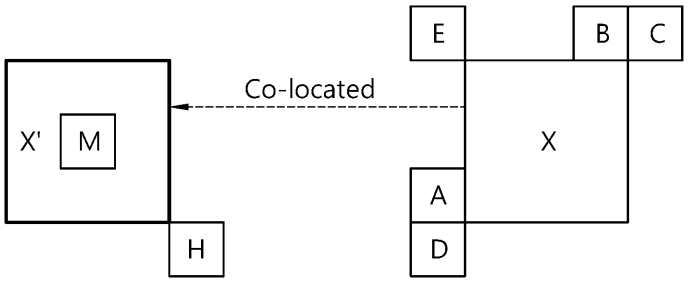
도면6


2N×2N
2N×N
N×2N
N×N
610
620
630
640
2N×nU
2N×nD
nL×2N
nR×2N
650
660
670
680

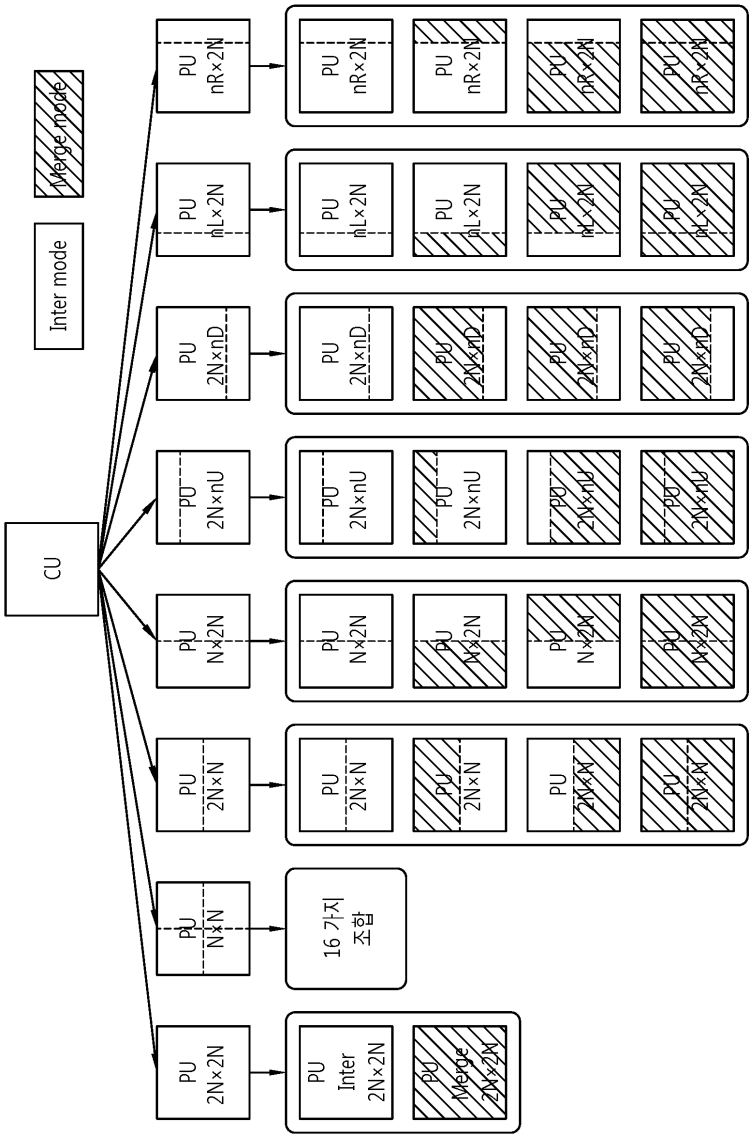
도면7


730
740
700
750
710
720

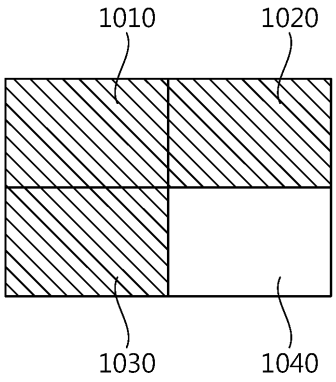
도면8


E
B
C
Co-located
X'
M
X
A
H
D

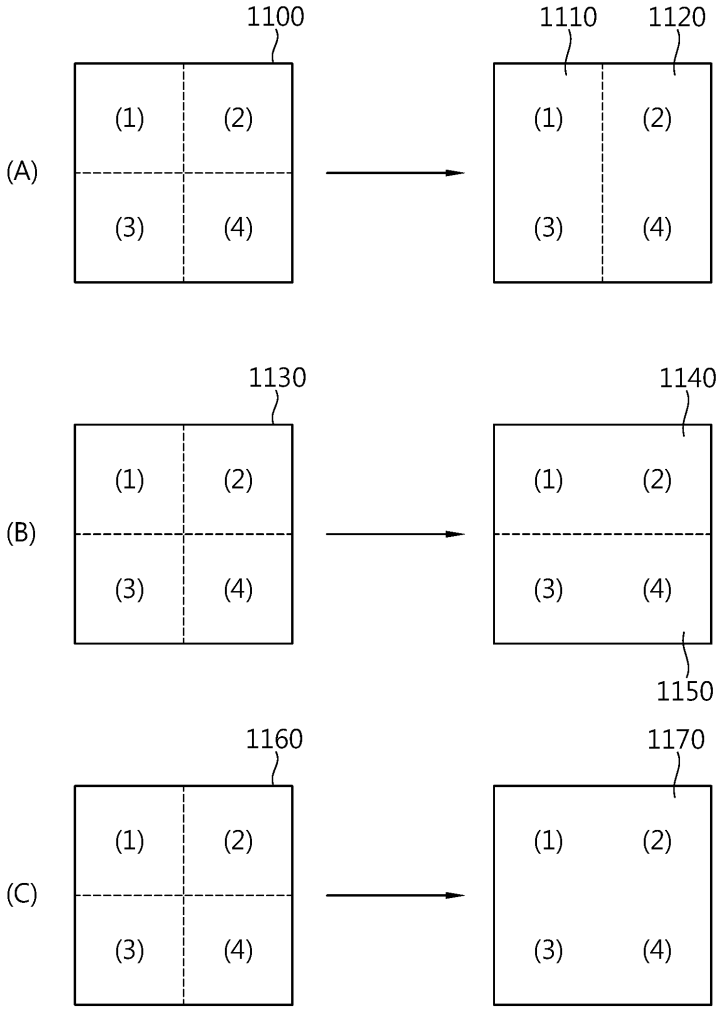
도면9


Merge mode
Inter mode
CU
PU 2N×2N
PU N×N
PU 2N×N
PU N×2N
PU 2N×nU
PU 2N×nD
PU nL×2N
PU nR×2N
16 가지 조합
PU Inter 2N×2N
PU Merge 2N×2N

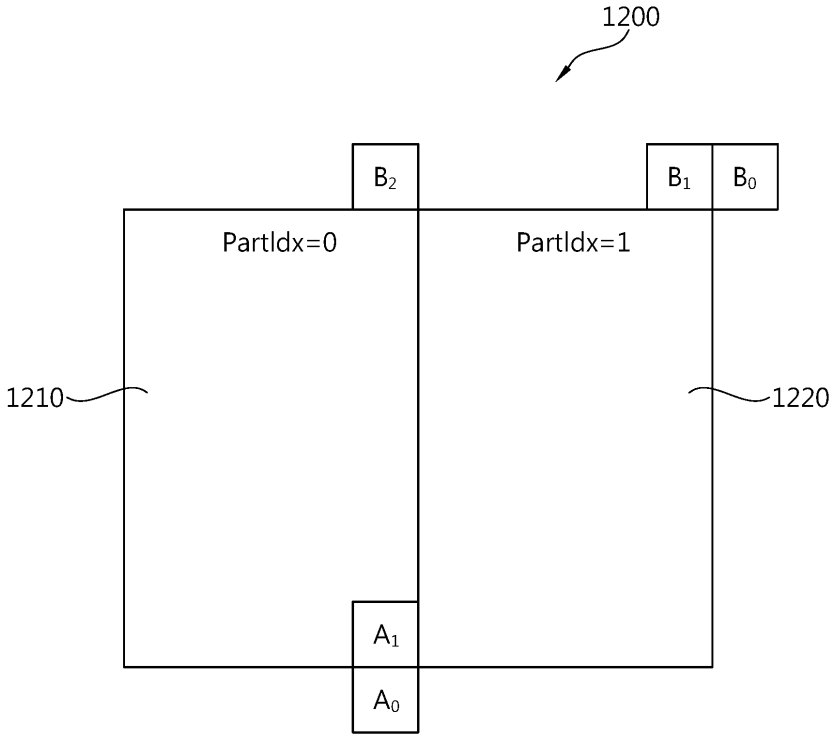
도면10


1010
1020
1030
1040

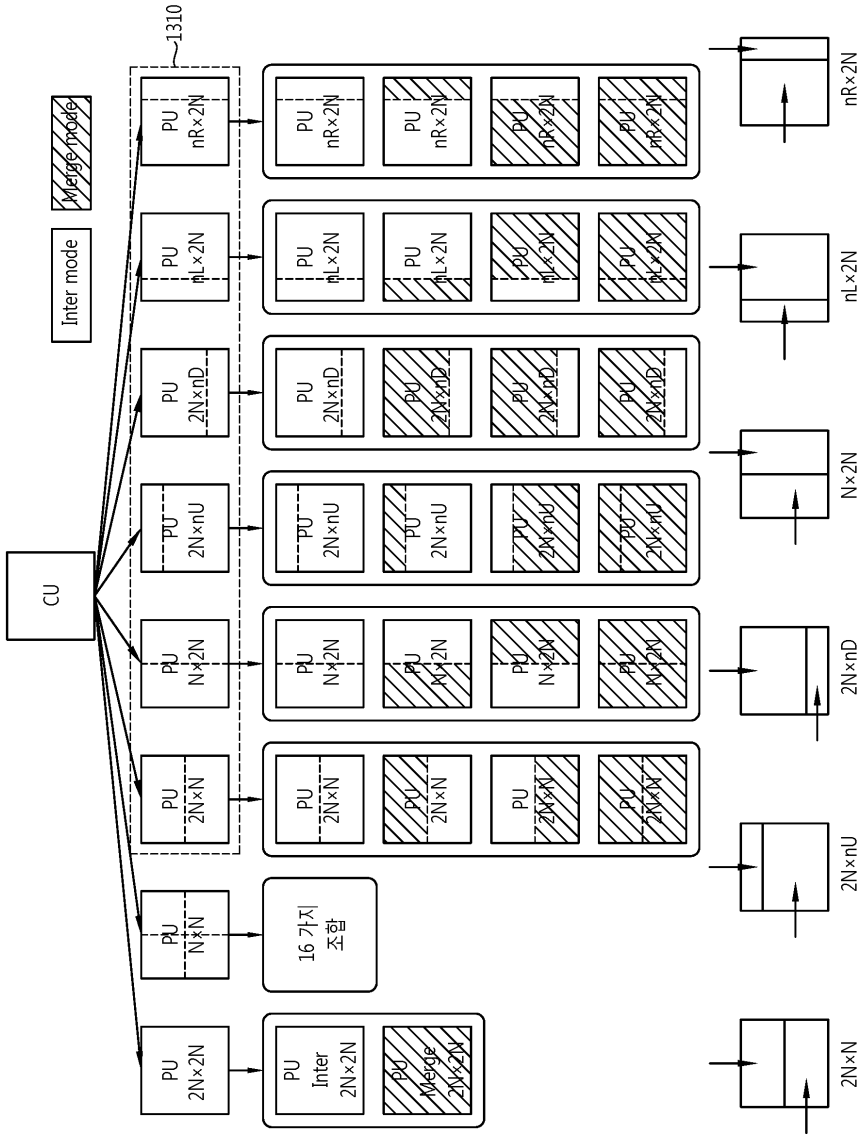
도면11


(A)
1100
(1)
(2)
(3)
(4)
1110
1120
(1)
(2)
(3)
(4)
(B)
1130
(1)
(2)
(3)
(4)
1140
(1)
(2)
(3)
(4)
1150
(C)
1160
(1)
(2)
(3)
(4)
1170
(1)
(2)
(3)
(4)

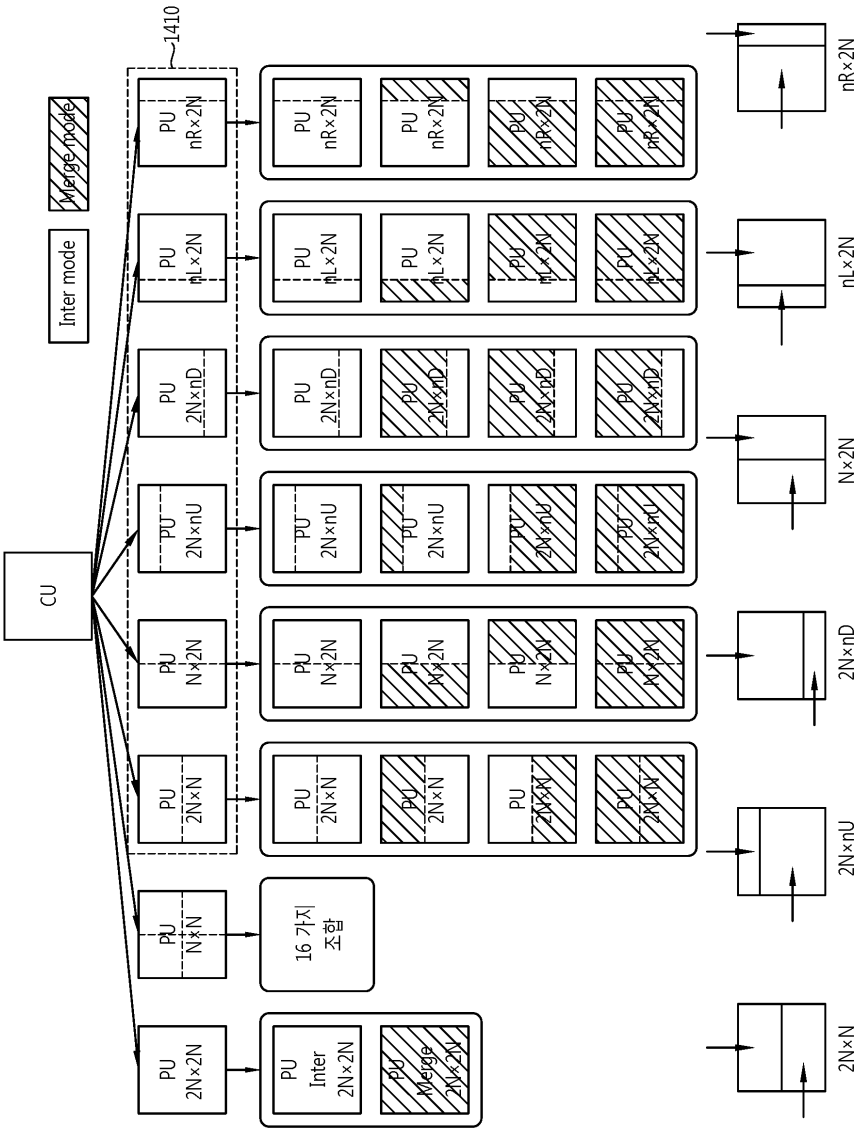
도면12



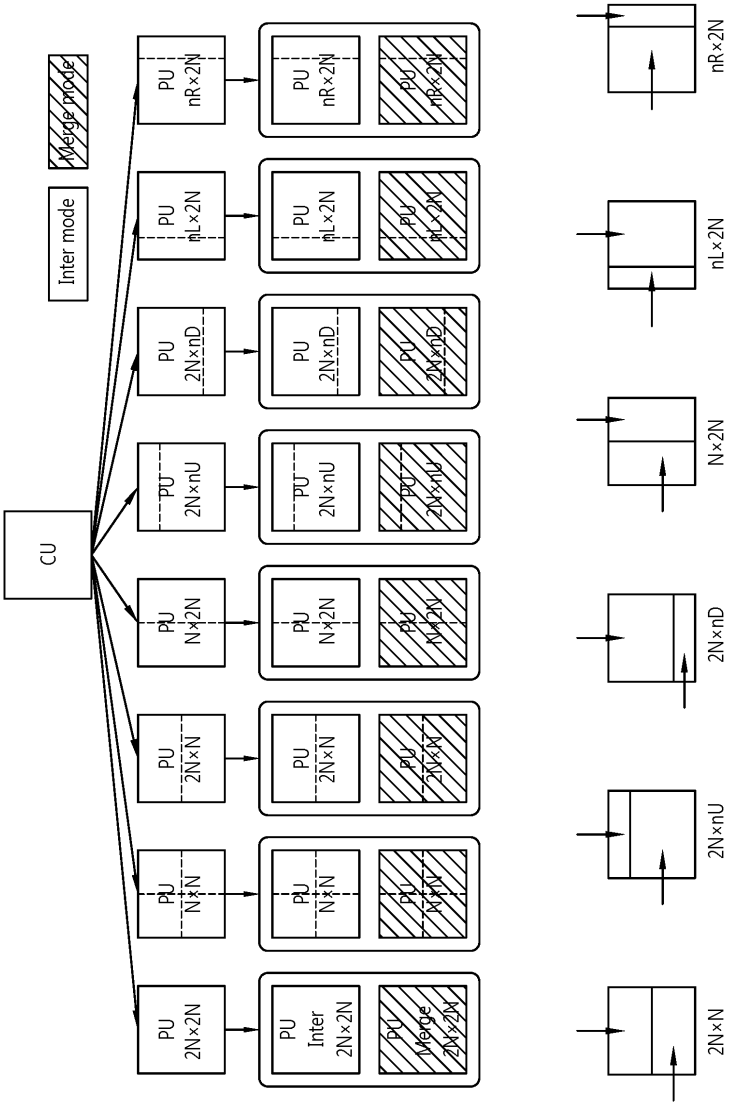
도면13


CU
Inter mode
Merge mode
1310
PU 2N×2N
PU N×N
PU 2N×N
PU N×2N
PU 2N×nU
PU 2N×nD
PU nL×2N
PU nR×2N
16 가지 조합
PU Inter 2N×2N
PU Merge 2N×2N
2N×N
N×2N
2N×nU
2N×nD
nL×2N
nR×2N

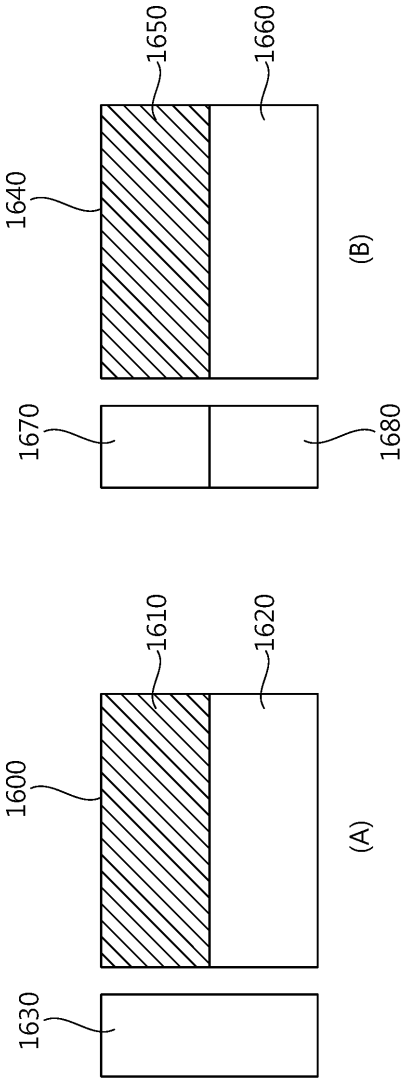
도면14


1410
Merge mode
Inter mode
CU
PU 2N×2N
PU N×N
PU 2N×N
PU N×2N
PU 2N×nU
PU 2N×nD
PU nL×2N
PU nR×2N
16 가지 조합
PU Inter 2N×2N
PU Merge 2N×2N
2N×N
N×2N
2N×nU
2N×nD
nL×2N
nR×2N

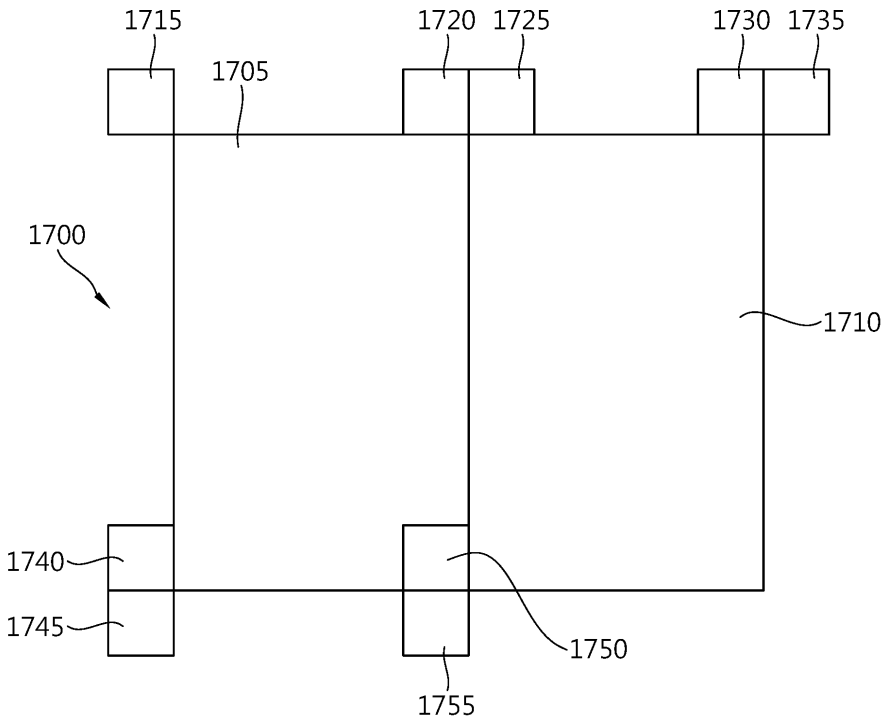
도면15


CU
Inter mode
Merge mode
PU 2N×2N
PU Inter 2N×2N
PU Merge 2N×2N
PU N×N
PU 2N×N
PU N×2N
PU 2N×nU
PU 2N×nD
PU nL×2N
PU nR×2N
2N×N
2N×nU
2N×nD
N×2N
nL×2N
nR×2N

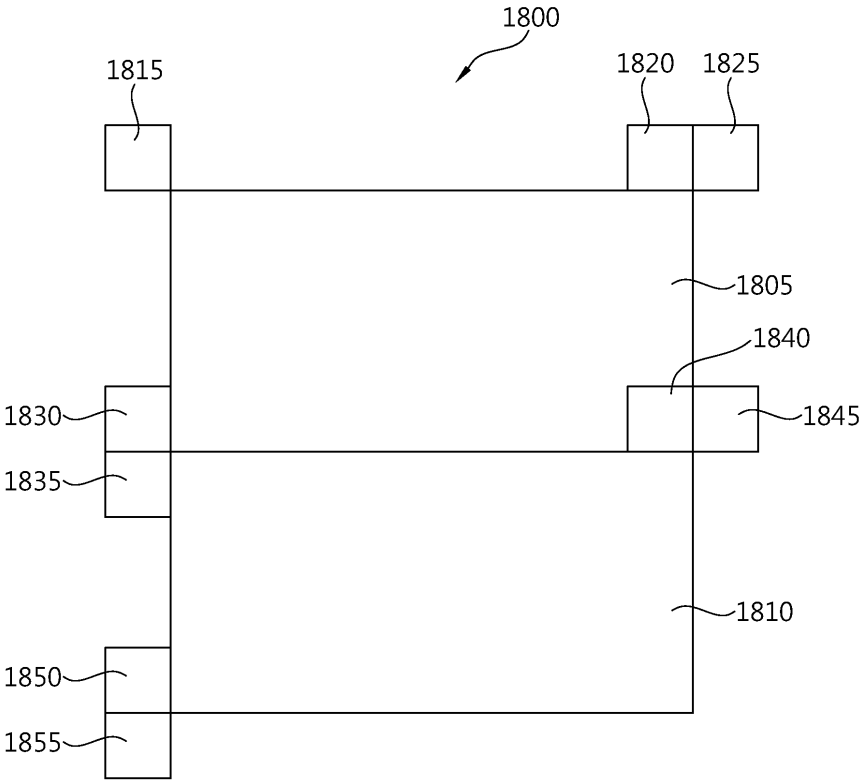
도면16



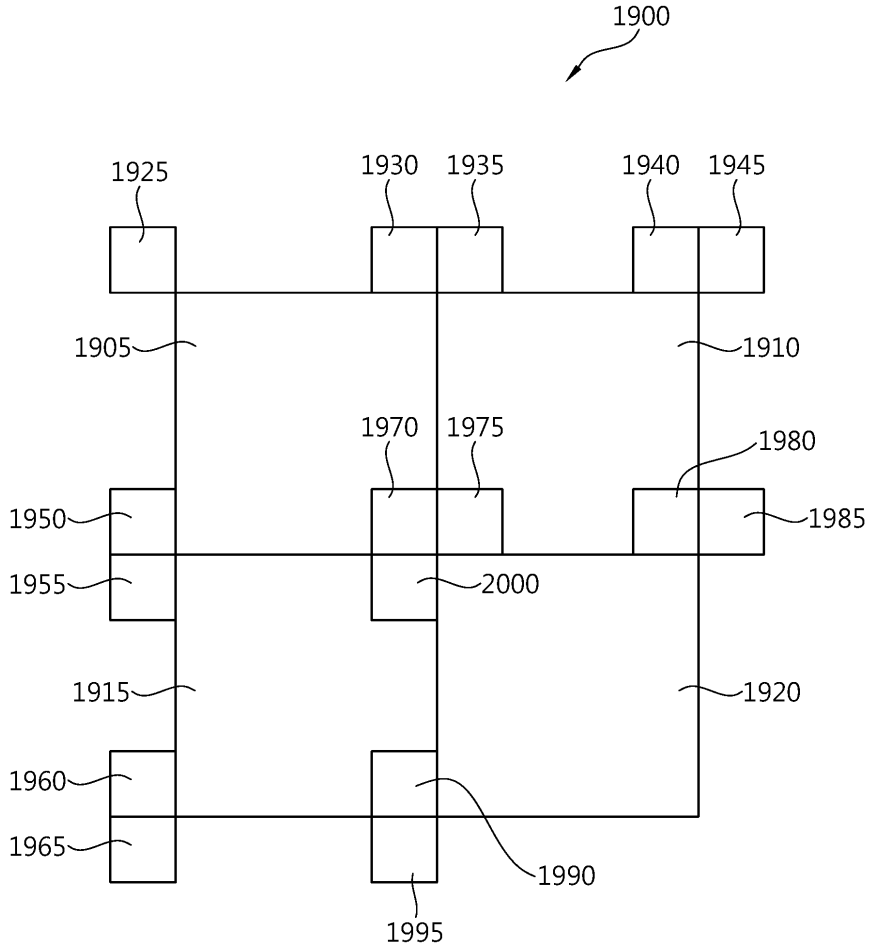
도면17


1715
1705
1720
1725
1730
1735
1700
1710
1740
1745
1750
1755

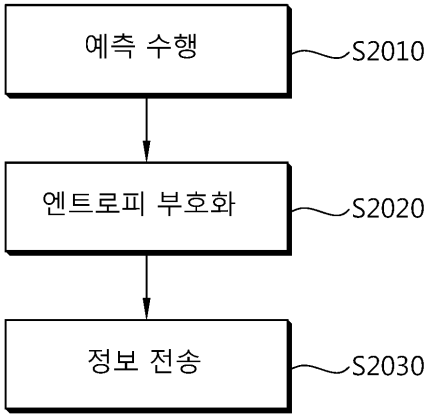
도면18


1800
1815
1820
1825
1805
1840
1830
1845
1835
1810
1850
1855

도면19


1900
1925
1930
1935
1940
1945
1905
1910
1970
1975
1980
1950
1985
1955
2000
1915
1920
1960
1965
1990
1995

도면20


예측 수행
S2010
엔트로피 부호화
S2020
정보 전송
S2030

도면21

