(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0146872
(43) 공개일자 2021년12월06일

(51) 국제특허분류(Int. Cl.)
H04N 19/51 (2014.01) *H04N 19/139* (2014.01)
H04N 19/176 (2014.01)
(52) CPC특허분류
H04N 19/51 (2015.01)
H04N 19/139 (2015.01)
(21) 출원번호 10-2021-0163196(분할)
(22) 출원일자 2021년11월24일
심사청구일자 2021년11월24일
(62) 원출원 특허 10-2021-0052165
원출원일자 2021년04월22일
심사청구일자 2021년04월22일
(30) 우선권주장
1020120113851 2012년10월12일 대한민국(KR)

(71) 출원인
한국전자통신연구원
대전광역시 유성구 가정로 218 (가정동)
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
김휘용
대전광역시 유성구 가정로 218
임성창
대전광역시 유성구 가정로 218
(뒷면에 계속)
(74) 대리인
성병기

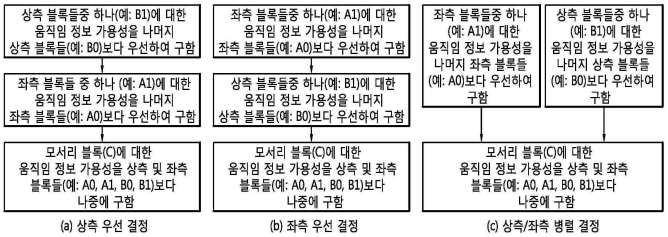
전체 청구항 수 : 총 3 항

(54) 발명의 명칭 **영상의 부호화/복호화 방법 및 이를 이용하는 장치**

(57) 요 약

본 발명에 따른 영상의 부호화 방법은 현재 블록의 주변 블록에 대한 가용성을 판단하는 단계와 상기 가용성 판단 결과에 기초하여 상기 현재 블록에 대한 머지 후보 리스트를 구성하는 단계와; 상기 머지 후보 리스트에 기초하여 상기 현재 블록에 대한 움직임 보상을 수행하는 단계를 포함하고, 상기 가용성을 판단하는 단계는, 상기 주변 블록에 대한 블록 가용성 정보를 유도하는 단계와; 상기 유도된 블록 가용성 정보를 이용하여 상기 주변 블록의 움직임 정보에 대한 가용성을 유도하는 단계를 포함하고, 상기 주변 블록에 대한 가용성을 판단하는 단계는 주변 블록이 복수인 경우 일정한 순서 제약에 따라 주변 블록들에 대한 가용성을 판단할 수 있다.

대 표 도



(52) CPC특허분류
H04N 19/176 (2015.01)
(72) 발명자
최진수
대전광역시 유성구 가정로 218
김진웅
대전광역시 유성구 가정로 218
박광훈
경기도 성남시 분당구 예원로 7, B동 302호(분당동, 동아빌라)
김경용
경기도 수원시 영통구 덕영대로 1725, 402호(영통동)

명 세 서

청구범위

청구항 1

영상의 복호화 방법에 있어서,

현재 블록의 예측 모드를 화면간 예측 모드로 결정하는 단계;

상기 현재 블록의 주변 블록에 대한 블록 가용성 정보를 유도하는 단계;

상기 유도된 블록 가용성 정보를 이용하여 상기 주변 블록의 움직임 정보에 대한 가용성을 유도하는 단계;

상기 움직임 정보에 대한 가용성에 기초하여 상기 현재 블록에 대한 움직임 정보 후보 리스트를 구성하는 단계;

상기 움직임 정보 후보 리스트에 기초하여 상기 현재 블록에 대한 움직임 벡터를 유도하는 단계; 및

상기 움직임 벡터를 이용하여 상기 현재 블록의 예측 블록을 생성하는 단계를 포함하되,

상기 움직임 정보에 대한 가용성을 유도하는 단계는 상기 주변 블록이 복수인 경우 일정한 순서 제약에 따라 수행되는 것을 특징으로 하고,

상기 움직임 정보 후보 리스트는,

공간적 후보 블록의 움직임 정보 및 시간적 후보 블록의 움직임 정보 중 적어도 하나를 포함하는 영상 복호화 방법.

청구항 2

영상의 부호화 방법에 있어서,

현재 블록의 예측 모드를 화면간 예측 모드로 결정하는 단계;

상기 현재 블록의 주변 블록에 대한 블록 가용성 정보를 유도하는 단계;

상기 유도된 블록 가용성 정보를 이용하여 상기 주변 블록의 움직임 정보에 대한 가용성을 유도하는 단계;

상기 움직임 정보에 대한 가용성에 기초하여 상기 현재 블록에 대한 움직임 정보 후보 리스트를 구성하는 단계; 및

상기 움직임 정보 후보 리스트에 관한 정보를 부호화하는 단계를 포함하되,

상기 움직임 정보에 대한 가용성을 유도하는 단계는 상기 주변 블록이 복수인 경우 일정한 순서 제약에 따라 수행되는 것을 특징으로 하고,

상기 움직임 정보 후보 리스트는,

공간적 후보 블록의 움직임 정보 및 시간적 후보 블록의 움직임 정보 중 적어도 하나를 포함하는 영상 부호화 방법.

청구항 3

비트스트림을 포함하는 비일시적 저장매체에 있어서, 현재 블록의 예측 모드를 화면간 예측 모드로 결정하는 단계;

상기 현재 블록의 주변 블록에 대한 블록 가용성 정보를 유도하는 단계;

상기 유도된 블록 가용성 정보를 이용하여 상기 주변 블록의 움직임 정보에 대한 가용성을 유도하는 단계;

상기 움직임 정보에 대한 가용성에 기초하여 상기 현재 블록에 대한 움직임 정보 후보 리스트를 구성하는 단계; 및

상기 움직임 정보 후보 리스트에 관한 정보를 부호화하는 단계를 포함하되,

상기 움직임 정보에 대한 가용성을 유도하는 단계는 상기 주변 블록이 복수인 경우 일정한 순서 제약에 따라 수행되는 것을 특징으로 하고,

상기 움직임 정보 후보 리스트는,

공간적 후보 블록의 움직임 정보 및 시간적 후보 블록의 움직임 정보 중 적어도 하나를 포함하는 영상 부호화 방법으로 생성된 비트스트림을 포함하는 비일시적 저장매체.

발명의 설명

기 술 분 야

[0001] 본 발명은 영상의 부호화 및 복호화 처리에 관한 것으로서, 보다 상세하게는 움직임 후보 목록을 구성하는 영상의 부호화 및 복호화 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 최근 HD(High Definition) 해상도(1280x1024 혹은 1920x1080)를 가지는 방송 서비스가 국내뿐만 아니라 세계적으로 확대되면서, 이제 많은 사용자들이 고해상도, 고화질의 영상에 많이 익숙해지고 있으며 그에 발맞춰서 많은 기관에서는 차세대 영상기기에 대한 개발에 박차를 가하고 있다. 또한 HDTV와 더불어 HDTV의 4배 이상의 해상도를 갖는 UHD(Ultra High Definition)에 대한 관심이 증대되면서 동영상 표준화 단체들은 보다 높은 해상도, 고화질의 영상에 대한 압축기술의 필요성을 인식하게 되었다. 또한 현재 HDTV, 휴대전화, 블루레이 플레이어에 사용되는 H.264/AVC보다 높은 압축 효율을 통해 동일한 화질을 유지하면서도 주파수 대역이나 저장 측면에서 많은 이득을 얻을 수 있는 새로운 표준이 절실한 실정이다. 현재 MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)는 공동으로 차세대 비디오 코덱인 HEVC(High Efficiency Video Coding)를 표준화하고 있으며, UHD 영상까지 포함한 영상을 H.264/AVC 대비 2배의 압축효율로 부호화를 하는 것을 목표로 하고 있다. 이는 HD, UHD 영상뿐만 아니라 3D 방송 및 이동통신망에서도 현재보다 낮은 주파수로 고화질의 영상을 제공할 수 있다.

[0003] HEVC는 현재 2010년 4월에 첫 JCT-VC(Joint Collaboration Team Video Coding) 회의가 열린 이후 각 기관의 기고를 통해 HM(HEVC Test Model)이라는 명칭의 코덱을 표준의 성능을 측정하기 위해 책정하였다.

발명의 내용

해결하려는 과제

[0004] 본 발명의 일 실시형태는 움직임 후보 목록을 구성하는데 있어서, 주변 블록의 사용 가능 여부를 판단하는 순서를 제공하는 방법 및 이를 이용하는 장치를 제공하는 것을 목적으로 한다.

[0005] 또한, 본 발명의 일 실시 형태는 움직임 후보 목록을 구성하는데 있어서, 부호화 단위 내의 예측 단위들 간의 병렬성을 제공하는 방법 및 이를 이용하는 장치를 제공하는 것을 목적으로 한다.

[0006] 이를 통하여 영상의 부, 복호화 효율을 향상시키는데 그 목적이 있다.

과제의 해결 수단

[0007] *본 발명의 일 실시예에 따른 영상의 부호화 방법은 현재 블록의 주변 블록에 대한 가용성을 판단하는 단계와 상기 가용성 판단 결과에 기초하여 상기 현재 블록에 대한 머지 후보 리스트를 구성하는 단계와; 상기 머지 후보 리스트에 기초하여 상기 현재 블록에 대한 움직임 보상을 수행하는 단계를 포함하고, 상기 가용성을 판단하는 단계는, 상기 주변 블록에 대한 블록 가용성 정보를 유도하는 단계와; 상기 유도된 블록 가용성 정보를 이용하여 상기 주변 블록의 움직임 정보에 대한 가용성을 유도하는 단계를 포함하고, 상기 주변 블록에 대한 가용성을 판단하는 단계는 주변 블록이 복수인 경우 일정한 순서 제약에 따라 주변 블록들에 대한 가용성을 판단할 수 있다.

[0008] 상기 현재 블록의 위치를 (xP, yP)라 할 경우, 상기 주변 블록은 (xP-1, yP+nPSH)에 존재하는 픽셀을 포함하는 제1 블록, (xP-1, yP+nPSH-1)에 존재하는 픽셀을 포함하는 제2 블록, (xP+nPSW, yP-1)에 위치하는 픽셀을 포함하는 제3 블록, (xP+nPSW-1, yP-1)에 위치하는 픽셀을 포함하는 제4 블록, (xP-1, yP-1)에 존재하는 픽셀을 포

함하는 제5 블록을 포함할 수 있다.

[0010] *상기 주변 블록에 대한 가용성을 판단하는 단계는, 상기 현재 블록과 상기 주변 블록에 대한 머지 후보 리스트를 병렬적으로 유도할 수 없는 경우 상기 블록 가용성 정보를 거짓(false, 0)으로 설정할 수 있다.

[0011] 상기 주변 블록에 대한 가용성을 판단하는 단계는, 상기 현재 블록의 파티션 형태가 PART_Nx2N, PART_nLx2N 및 PART_nRx2N 중 하나이고, 부호화 유닛 내의 상기 현재 블록에 대한 인덱스 partIdx가 '1' 이면, 상기 제2 블록에 대한 상기 블록 가용성 정보를 거짓(0)으로 설정할 수 있다.

[0012] 상기 주변 블록에 대한 가용성을 판단하는 단계는, 상기 현재 블록의 파티션 형태가 PART_2NxN, PART_2NxnU 및 PART_2NxnD 중 하나이고 부호화 유닛 내의 상기 현재 블록에 대한 인덱스 partIdx가 '1' 이면, 상기 제4 블록에 대한 상기 블록 가용성 정보를 거짓(false, 0)으로 설정할 수 있다.

[0013] 상기 주변 블록의 움직임 정보에 대한 가용성을 유도하는 단계에서, 상기 제2 블록의 움직임 정보에 대한 가용성은 상기 제 1 블록, 상기 제3 블록, 상기 제4 블록 및 상기 제5 블록의 움직임 정보에 대한 가용성보다 먼저 유도되고, 상기 제4 블록의 움직임 정보에 대한 가용성은 상기 제3 블록 및 상기 제5 블록의 움직임 정보에 대한 가용성보다 먼저 유도되며, 상기 제5 블록의 움직임 정보에 대한 가용성은 상기 제1 블록, 상기 제2 블록, 상기 제3 블록 및 상기 제4 블록의 움직임 정보에 대한 가용성보다 나중에 유도될 수 있다.

[0014] 상기 주변 블록의 움직임 정보에 대한 가용성을 유도하는 단계는 상기 제2 블록, 상기 제 4 블록, 상기 제3 블록, 상기 제1 블록, 상기 제5 블록 순으로 수행될 수 있다.

[0015] 상기 블록 가용성 정보를 유도하는 단계에서, 상기 주변 블록이 존재하지 않거나 인트라 모드로 부호화 되어 있으면 상기 블록 가용성 정보를 거짓(false, 0)으로 설정하고, 상기 블록 가용성 정보가 거짓(false, 0)이면 상기 움직임 정보에 대한 가용성은 0으로 설정될 수 있다.

[0016] 상기 움직임 정보에 대한 가용성을 유도하는 단계에서, 상기 제2 블록의 상기 블록 가용성 정보가 참(true, 1)이고, 상기 제2 블록의 움직임 정보와 상기 제4 블록의 움직임 정보가 동일하면, 상기 제4 블록의 움직임 정보에 대한 가용성은 0으로 설정될 수 있다.

[0017] 상기 움직임 정보에 대한 가용성을 유도하는 단계에서, 상기 제4 블록의 상기 블록 가용성 정보가 참(true, 1)이고, 상기 제4 블록의 움직임 정보와 상기 제3 블록의 움직임 정보가 동일하면, 상기 제3 블록의 움직임 정보에 대한 가용성은 0으로 설정될 수 있다.

[0018] 상기 움직임 정보에 대한 가용성을 유도하는 단계에서, 상기 제2 블록의 상기 블록 가용성 정보가 참(true, 1)이고, 상기 제2 블록의 움직임 정보와 상기 제1 블록의 움직임 정보가 동일하면, 상기 제1 블록의 움직임 정보에 대한 가용성은 0으로 설정될 수 있다.

[0019] 상기 움직임 정보에 대한 가용성을 유도하는 단계에서, 상기 제2 블록의 상기 블록 가용성 정보가 참(true, 1)이고, 상기 제2 블록의 움직임 정보와 상기 제5 블록의 움직임 정보가 동일하면 상기 제5 블록의 움직임 정보에 대한 가용성은 0으로 설정될 수 있다.

[0020] 상기 움직임 정보에 대한 가용성을 유도하는 단계에서, 상기 제4 블록의 상기 블록 가용성 정보가 참(true, 1)이고, 상기 제4 블록의 움직임 정보와 상기 제5 블록의 움직임 정보가 동일하면 상기 제5 블록의 움직임 정보에 대한 가용성은 0으로 설정될 수 있다.

[0021] 상기 움직임 정보에 대한 가용성을 유도하는 단계에서, 상기 제 1 블록, 상기 제2 블록, 상기 제3 블록, 상기 제4 블록의 상기 움직임 정보에 대한 가용성이 모두 1이면 상기 제5 블록의 움직임 정보에 대한 가용성은 0으로 설정될 수 있다.

발명의 효과

[0022] 본 발명의 일 실시예에 따르면, 움직임 후보 목록을 구성하는데 있어서, 주변 블록의 사용 가능 여부를 판단하는 순서를 제공하는 방법 및 이를 이용하는 장치가 제공된다.

[0023] 또한 본 발명에 따르면 움직임 후보 목록을 구성하는데 있어서, 부호화 단위 내의 예측 단위들 간의 완벽한 병렬성을 방법 및 이를 이용하는 장치가 제공된다.

[0024] 이를 통하여 영상의 부, 복호화 효율을 향상시킬 수 있는 영상의 부호화/복호화 방법 및 장치가 제공된다.

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 병합 움직임 후보 목록으로 사용되는 현재 블록의 주변 블록들의 일예를 도시한 도면이다.

도 4는 화면 간 부호화에서 움직임 보상 과정을 설명하기 위한 제어 흐름도이다.

도 5는 일 예에 따라 하나의 CU가 2개의 PU로 나누어졌을 경우, 병합 움직임 후보 목록을 설정하는 일예를 나타낸 도면이다.

도 6은 다른 예에 따라 하나의 CU가 2개의 PU로 나누어졌을 경우, 병합 움직임 후보 목록을 설정하는 일예를 나타낸 도면이다.

도 7은 현재 블록의 주변 블록들의 일예를 도시한 도면이다.

도 8은 움직임 정보 가용성을 결정하는 순서를 제약하는 방법에 대한 여러 예를 도시한 도면이다.

도 9는 순차적으로 블록 움직임 정보의 가용성 정보를 판단하는 경우 주변 블록을 도시한 도면이다.

도 10은 하나의 CU가 복 수의 PU로 파티셔닝되는 예를 도시한 것이다.

도 11은 본 발명의 일 실시예에 따라 블록 가용성 정보를 재설정하는 방법을 설명하기 위한 제어 흐름도이다.

도 12는 현재 블록의 머지 후보 리스트 생성을 설명하기 위한 현재 블록 및 주변 블록을 도시한 도면이다.

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.

[0032] 도 1은 영상 부호화 장치의 일 실시예에 따른 구성을 나타내는 블록도이다.

[0033] 도 1을 참조하면, 상기 영상 부호화 장치(100)는 움직임 예측부(111), 움직임 보상부(112), 인트라 예측부(120), 스위치(115), 감산기(125), 변환부(130), 양자화부(140), 엔트로피 부호화부(150), 역양자화부(160), 역변환부(170), 가산기(175), 필터부(180) 및 참조영상 버퍼(190)를 포함한다.

[0034] 영상 부호화 장치(100)는 입력 영상에 대해 인트라(intra) 모드 또는 인터(inter) 모드로 부호화를 수행하고 비트스트림(bit stream)을 출력할 수 있다. 인트라 예측은 화면 내 예측, 인터 예측은 화면 간 예측을 의미한다. 인트라 모드인 경우 스위치(115)가 인트라로 전환되고, 인터 모드인 경우 스위치(115)가 인터로 전환된다. 영상 부호화 장치(100)는 입력 영상의 입력 블록에 대한 예측 블록을 생성한 후, 입력 블록과 예측 블록의 차분을 부호화할 수 있다.

[0035] 인트라 모드인 경우, 인트라 예측부(120)는 현재 블록 주변의 이미 부호화된 블록의 화소값을 이용하여 공간적 예측을 수행하여 예측 블록을 생성할 수 있다.

[0037] *인터 모드인 경우, 움직임 예측부(111)는, 움직임 예측 과정에서 참조 영상 버퍼(190)에 저장되어 있는 참조 영상에서 입력 블록과 가장 매치가 잘 되는 영역을 찾아 움직임 벡터를 구할 수 있다. 움직임 보상부(112)는 움직임 벡터와 참조 영상 버퍼(190)에 저장되어 있는 참조 영상을 이용하여 움직임 보상을 수행함으로써 예측 블록을 생성할 수 있다.

[0038] 감산기(125)는 입력 블록과 생성된 예측 블록의 차분에 의해 잔여 블록(residual block)을 생성할 수 있다. 변환부(130)는 잔여 블록에 대해 변환(transform)을 수행하여 변환 계수(transform coefficient)를 출력할 수 있다. 그리고 양자화부(140)는 입력된 변환 계수를 양자화 파라미터에 따라 양자화하여 양자화된 계수(quantized coefficient)를 출력할 수 있다.

[0039] 엔트로피 부호화부(150)는, 양자화부(140)에서 산출된 값들 또는 부호화 과정에서 산출된 부호화 파라미터 값 등을 기초로, 심볼(symbol)을 확률 분포에 따라 엔트로피 부호화하여 비트스트림(bit stream)을 출력할 수 있다. 엔트로피 부호화 방법은 다양한 값을 갖는 심볼을 입력 받아, 통계적 중복성을 제거하면서, 복호 가능한 2진수의 열로 표현하는 방법이다.

[0040] 여기서, 심볼이란 부호화/복호화 대상 구문 요소(syntax element) 및 부호화 파라미터(coding parameter), 잔여 신호(residual signal)의 값 등을 의미한다. 부호화 파라미터는 부호화 및 복호화에 필요한 매개변수로서, 구문 요소와 같이 부호화 장치에서 부호화되어 복호화 장치로 전달되는 정보뿐만 아니라, 부호화 혹은 복호화 과정에서 유추될 수 있는 정보를 포함할 수 있으며 영상을 부호화하거나 복호화할 때 필요한 정보를 의미한다. 부호화 파라미터는 예를 들어 인트라/인터 예측모드, 이동/움직임 벡터, 참조 영상 색인, 부호화 블록 패턴, 잔여 신호 유무, 변환 계수, 양자화된 변환 계수, 양자화 파라미터, 블록 크기, 블록 분할 정보 등의 값 또는 통계를 포함할 수 있다. 또한 잔여 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. 잔여 신호는 블록 단위에서는 잔여 블록이라 할 수 있다.

[0041] 엔트로피 부호화가 적용되는 경우, 높은 발생 확률을 갖는 심볼에 적은 수의 비트가 할당되고 낮은 발생 확률을 갖는 심볼에 많은 수의 비트가 할당되어 심볼이 표현됨으로써, 부호화 대상 심볼들에 대한 비트열의 크기가 감소될 수 있다. 따라서 엔트로피 부호화를 통해서 영상 부호화의 압축 성능이 높아질 수 있다.

[0042] 엔트로피 부호화를 위해 지수 골룸(exponential golomb), CAVLC(Context-Adaptive Variable Length Coding), CABAC(Context-Adaptive Binary Arithmetic Coding)과 같은 부호화 방법이 사용될 수 있다. 예를 들어, 엔트로피 부호화부(150)에는 가변 길이 부호화(VLC: Variable Lenghth Coding/Code) 테이블과 같은 엔트로피 부호화를 수행하기 위한 테이블이 저장될 수 있고, 엔트로피 부호화부(150)는 저장된 가변 길이 부호화(VLC) 테이블을 사용하여 엔트로피 부호화를 수행할 수 있다. 또한 엔트로피 부호화부(150)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.

[0043] 양자화된 계수는 역양자화부(160)에서 역양자화되고 역변환부(170)에서 역변환될 수 있다. 역양자화, 역변환된 계수는 가산기(175)를 통해 예측 블록과 더해지고 복원 블록이 생성될 수 있다.

[0044] 복원 블록은 필터부(180)를 거치고, 필터부(180)는 디블록킹 필터(deblocking filter), SAO(Sample Adaptive Offset), ALF(Adaptive Loop Filter) 중 적어도 하나 이상을 복원 블록 또는 복원 픽처에 적용할 수 있다. 필

터부(180)를 거친 복원 블록은 참조 영상 버퍼(190)에 저장될 수 있다.

[0046] 도 2는 영상 복호화 장치의 일 실시예에 따른 구성을 나타내는 블록도이다. 도 2를 참조하면, 상기 영상 복호화 장치(200)는 엔트로피 복호화부(210), 역양자화부(220), 역변환부(230), 인트라 예측부(240), 움직임 보상부(250), 필터부(260) 및 참조 영상 버퍼(270)를 포함한다.

[0047] 영상 복호화 장치(200)는 부호화 장치에서 출력된 비트스트림을 입력 받아 인트라 모드 또는 인터 모드로 복호화를 수행하고 재구성된 영상, 즉 복원 영상을 출력할 수 있다. 인트라 모드인 경우 스위치가 인트라로 전환되고, 인터 모드인 경우 스위치가 인터로 전환될 수 있다. 영상 복호화 장치(200)는 입력 받은 비트스트림으로부터 복원된 잔여 블록(residual block)을 얻고 예측 블록을 생성한 후 복원된 잔여 블록과 예측 블록을 더하여 재구성된 블록, 즉 복원 블록을 생성할 수 있다.

[0048] 엔트로피 복호화부(210)는, 입력된 비트스트림을 확률 분포에 따라 엔트로피 복호화하여, 양자화된 계수(quantized coefficient) 형태의 심볼을 포함한 심볼들을 생성할 수 있다. 엔트로피 복호화 방법은 2진수의 열을 입력 받아 각 심볼들을 생성하는 방법이다. 엔트로피 복호화 방법은 상술한 엔트로피 부호화 방법과 유사하다.

[0049] 양자화된 계수는 역양자화부(220)에서 역양자화되고 역변환부(230)에서 역변환되며, 양자화된 계수가 역양자화/역변환 된 결과, 복원된 잔여 블록(residual block)이 생성될 수 있다.

[0050] 인트라 모드인 경우, 인트라 예측부(240)는 현재 블록 주변의 이미 부호화된 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.

[0051] 복원된 잔여 블록과 예측 블록은 가산기(255)를 통해 더해지고, 더해진 블록은 필터부(260)를 거친다. 필터부(260)는 디블록킹 필터, SAO, ALF 중 적어도 하나 이상을 복원 블록 또는 복원 픽처에 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.

[0052] 상기 영상 복호화 장치(200)에 포함되어 있는 엔트로피 복호화부(210), 역양자화부(220), 역변환부(230), 인트라 예측부(240), 움직임 보상부(250), 필터부(260) 및 참조 영상 버퍼(270) 중 영상의 복호화에 직접적으로 관련된 구성요소들, 예컨대, 엔트로피 복호화부(210), 역양자화부(220), 역변환부(230), 인트라 예측부(240), 움직임 보상부(250), 필터부(260) 등을 다른 구성요소와 구분하여 복호화부 또는 디코딩부로 표현할 수 있다.

[0053] 또한, 영상 복호화 장치(200)는 비트스트림에 포함되어 있는 인코딩된 영상에 관련된 정보를 파싱하는 도시하지 않은 파싱부를 더 포함할 수 있다. 파싱부는 엔트로피 복호화부(210)를 포함할 수도 있고, 엔트로피 복호화부(210)에 포함될 수도 있다. 이러한 파싱부는 또한 디코딩부의 하나의 구성요소로 구현될 수도 있다.

[0055] 통상적으로 화면간 예측은 현재 픽처의 이전 픽처 또는 이후 픽처 중 적어도 하나를 참조 픽처로 하고, 참조 픽처를 기반으로 현재 블록에 대한 예측을 수행할 수 있다. 현재 블록의 예측에 이용되는 영상을 참조 픽처(reference picture) 또는 참조 프레임(reference frame)이라고 한다.

[0056] 참조 픽처 내의 영역은 참조 픽처를 지시하는 참조 픽처 인덱스(refIdx) 및 움직임 벡터(motion vector) 등을 이용하여 나타낼 수 있다.

[0057] 화면간 예측은 참조 픽처 및 참조 픽처 내에서 현재 블록에 대응하는 참조 블록을 선택해서, 현재 블록에 대한 예측 블록을 생성할 수 있다.

[0058] 화면 간 예측에서 부호화 장치 및 복호화 장치는 현재 블록의 움직임 정보를 도출한 후, 도출된 움직임 정보에 기반하여 화면 간 예측 및/또는 움직임 보상을 수행할 수 있다. 이 때, 부호화 장치 및 복호화 장치는 복원된 주변 블록(neighboring block) 및/또는 이미 복원된 콜(col) 픽처(collocated picture) 내에서 현재 블록에 대응되는 콜(col) 블록(collocated block)의 움직임 정보를 이용함으로써, 부호화/복호화 효율을 향상시킬 수 있다.

[0059] 여기서, 복원된 주변 블록은 이미 부호화 및/또는 복호화되어 복원된 현재 픽쳐 내의 블록으로서, 현재 블록에 인접한 블록 및/또는 현재 블록의 외부 코너에 위치한 블록을 포함할 수 있다. 또한 부호화 장치 및 복호화 장치는, 콜 픽쳐 내에서 현재 블록과 공간적으로 대응되는 위치에 존재하는 블록을 기준으로 소정의 상대적인 위치를 결정할 수 있고, 상기 결정된 소정의 상대적인 위치(상기 현재 블록과 공간적으로 대응되는 위치에 존재하는 블록의 내부 및/또는 외부의 위치)를 기반으로 상기 콜 블록을 도출할 수 있다. 여기서, 일례로 콜 픽쳐는 참조 픽쳐 리스트에 포함된 참조 픽쳐 중에서 하나의 픽쳐에 해당될 수 있다.

[0060] 화면간 예측은 현재 블록과의 레지듀얼(residual) 신호가 최소화되며 움직임 벡터 크기 역시 최소가 되도록 예측 블록을 생성할 수 있다.

[0061] 한편, 움직임 정보 도출 방식은 현재 블록의 예측 모드에 따라 달라질 수 있다. 인터 예측을 위해 적용되는 예측 모드에는 AMVP(Advanced Motion Vector Predictor), 머지(merge) 등이 있을 수 있다.

[0062] 일례로, AMVP(Advanced Motion Vector Predictor)가 적용되는 경우, 부호화 장치 및 복호화 장치는 복원된 주변 블록의 움직임 벡터 및/또는 콜 블록의 움직임 벡터를 이용하여, 예측 움직임 벡터 후보 리스트를 생성할 수 있다. 즉, 복원된 주변 블록의 움직임 벡터 및/또는 콜 블록의 움직임 벡터는 예측 움직임 벡터 후보로 사용될 수 있다. 부호화 장치는 상기 리스트에 포함된 예측 움직임 벡터 후보 중에서 선택된 최적의 예측 움직임 벡터를 지시하는 예측 움직임 벡터 인덱스를 복호화 장치로 전송할 수 있다. 이 때, 복호화 장치는 상기 예측 움직임 벡터 인덱스를 이용하여, 예측 움직임 벡터 후보 리스트에 포함된 예측 움직임 벡터 후보 중에서, 현재 블록의 예측 움직임 벡터를 선택할 수 있다.

[0063] 부호화 장치는 현재 블록의 움직임 벡터와 예측 움직임 벡터 간의 움직임 벡터 차분(MVD: Motion Vector Difference)을 구할 수 있고, 이를 부호화하여 복호화 장치로 전송할 수 있다. 이 때, 복호화 장치는 수신된 움직임 벡터 차분을 복호화할 수 있고, 복호화된 움직임 벡터 차분과 예측 움직임 벡터의 합을 통해 현재 블록의 움직임 벡터를 도출할 수 있다.

[0064] 부호화 장치는 또한 참조 픽처를 지시하는 참조 픽처 인덱스 등을 복호화 장치에 전송할 수 있다.

[0065] 복호화 장치는 주변 블록의 움직임 정보들을 이용하여 현재 블록의 움직임 벡터를 예측하고, 부호화 장치로부터 수신한 레지듀얼을 이용하여 현재 블록에 대한 움직임 벡터를 유도할 수 있다. 복호화 장치는 유도한 움직임 벡터와 부호화 장치로부터 수신한 참조 픽처 인덱스 정보를 기반으로 현재 블록에 대한 예측 블록을 생성할 수 있다.

[0066] 화면 간 예측에 이용되는 다른 모드 중 하나 인 스킵 모드의 경우에, 주변 블록의 정보를 그대로 현재 블록에 이용할 수 있다. 따라서 스킵 모드의 경우에, 부호화 장치는 현재 블록의 움직임 정보로서 어떤 블록의 움직임 정보를 이용할 것인지를 지시하는 정보 외에 레지듀얼 등과 같은 신택스 정보를 복호화 장치에 전송하지 않는다.

[0067] 다른 예로, 병합 움직임, 즉 머지(merge)가 적용되는 경우, 부호화 장치 및 복호화 장치는 복원된 주변 블록의 움직임 정보 및/또는 콜 블록의 움직임 정보를 이용하여, 병합 움직임 후보 목록(머지 후보 리스트, merge cadidate list)를 생성할 수 있다. 이때 움직임 정보란, 움직임 벡터, 참조 영상에 대한 인덱스 그리고 예측 방향(단방향, 양방향, 등) 중에서 적어도 하나를 포함한다.

[0068] 병합 움직임은 부호화 유닛(Coding Unit: CU, 이하 'CU'라 함) 단위의 병합 움직임과 예측 유닛(Prediction Unit: PU, 이하 'PU'라 함) 단위의 병합 움직임이 가능하다. CU 또는 PU(이하, 설명의 편의를 위해 '블록'이라 함) 단위로 병합 움직임을 수행하는 경우에는, 블록 파티션(partition)별로 병합 움직임을 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 병합 움직임을 할 것인지에 대한 정보를 전송할 필요가 있다.

[0069] 병합 움직임 후보 목록은 움직임 정보들이 저장된 목록을 나타내며, 병합 움직임이 수행되기 전에 생성된다. 여기서 병합 움직임 후보 목록에 저장되는 움직임 정보는 현재 블록에 인접한 주변 블록의 움직임 정보이거나 참조 영상에서 현재 블록에 대응되는(collocated) 블록의 움직임 정보일 수 있다. 또한 병합 움직임 후보 목록에 저장되는 움직임 정보는 이미 병합 움직임 후보 목록에 존재하는 움직임 정보들을 조합하여 만든 새로운 움직임 정보일 수 있다.

[0070] 도 3은 병합 움직임 후보 목록으로 사용되는 현재 블록의 주변 블록들의 일예를 도시한 도면이다.

[0071] 병합 움직임 후보 목록은 도 3의 주변 블록(A, B, C, D, E)과 그리고 동일 위치의 후보 블록(H(혹은 M))에 대하여 해당 블록의 움직임 정보가 현재 블록의 병합 움직임에 이용될 수 있는지를 판단하여, 이용 가능한 경우에는 해당 블록의 움직임 정보를 병합 움직임 후보 목록에 입력할 수 있다.

[0072] 시간적 후보 예측 블록으로 사용되는 동일 위치 후보 블록은 현재 블록을 포함하는 픽쳐 내의 픽셀 위치 (xP, yP)를 기초로 현재 블록의 콜 픽쳐(Colocated Picture)에서 (xP+nPSW, yP+nPSH) 위치의 픽셀을 포함하는 예측 블록(H)이거나 (xP+nPSW, yP+nPSH) 위치의 픽셀을 포함하는 예측 블록이 가용하지 않은 경우, (xP+(nPSW>>1), yP+(nPSH>>1)) 위치의 픽셀을 포함하는 예측 블록(M)이 될 수 있다.

[0073] 각 주변 블록은 서로 동일한 움직임 정보를 가지는지 확인하여 동일한 움직임 정보일 경우, 해당 주변 블록의 움직임 정보는 병합 움직임 후보 목록에 포함되지 않는다.

[0074] 실시 일 예로, 도 3에서 X 블록에 대한 병합 움직임 후보 목록을 생성할 때, 주변 블록 A가 사용 가능하여 병합 움직임 후보 목록에 포함되면, 주변 블록 B는 주변 블록 A와 동일한 움직임 정보가 아닐 경우에만 병합 움직임 후보 목록에 포함될 수 있다. 동일한 방법으로 주변 블록 C는 주변 블록 B와 동일한 움직임 정보가 아닐 경우에만 병합 움직임 후보 목록에 포함될 수 있다. 동일한 방법으로 주변 블록 D와 주변 블록 E에 적용될 수 있다.

[0075] 여기서 동일한 움직임 정보라는 것은 움직임 벡터가 동일하고 동일한 참조 픽쳐를 사용하고 동일한 예측 방향(단방향(정방향, 역방향), 양방향)을 사용함을 의미할 수 있다.

[0076] 마지막으로 도 3에서 X 블록에 대한 병합 움직임 후보 목록은 소정의 순서, 예컨대, A→B→C→D→E→H(혹은 M) 블록 순서로 목록에 추가될 수 있다.

[0078] 도 4는 화면 간 부호화에서 움직임 보상 과정을 설명하기 위한 제어 흐름도이다.

[0079] 도 4와 같이, 움직임 보상 과정은 현재 블록 및 주변 블록에 대한 정보가 입력되면, 주변 블록들에 대한 가용성을 판단하고(S410), 움직임 후보 목록을 구성하고(S420), 구성된 움직임 후보 목록을 통해 현재 블록에 대한 움직임 정보를 출력하여, 최종적으로 움직임 보상된 영상을 출력하는 단계(S430)로 구성된다.

[0080] 도 4에서 현재 블록에 대한 움직임 후보 목록을 구성하기 위해 사용되는 주변 블록들은 복 수개로 다양하게 설정될 수 있으며, 일예로 도 3의 주변 블록(A, B, C, D, E)과 그리고 동일 위치의 후보 블록(H 또는 M))일 수 있다.

[0081] 도 4의 첫 번째 단계인 주변 블록의 가용성 판단 단계(S411)는 해당 주변 블록이 존재할 경우 해당 블록이 가용하다고 판단하는 과정이며, 가용성 판단의 결과로 블록 가용성 정보가 출력된다. 여기서 주변 블록이 존재한다는 것은 주변 블록이 현재 블록이 속한 동일한 영상(픽쳐), 슬라이스, 타일 내부에 존재하여 복호 과정에서 접근이 가능하다는 것을 의미할 수 있다. 본 발명에서 판단이라 함은 결정 혹은 설정의 의미로 사용될 수 있다. 가용성 판단시 추가적으로 해당 주변 블록이 화면내 부호화 되어 있지 않은 경우에만 가용하다고 판단 할 수도 있다.

[0082] 판단 결과, 블록 가용성 정보가 ‘1’ 이면, 해당 블록은 사용 가능(일예로, 영상 내에 존재)하며, ‘0’ 이면 사용 불가능(일예로, 영상 밖에 위치하여 존재하지 않을 수 있음)한 것을 나타낼 수 있으며, 그 역도 가능하다.

[0083] 다음으로 블록 가용성 정보를 바탕으로 주변 블록의 움직임 정보에 대한 가용성이 판단(S413)된다. 이를 통하여 해당 주변 블록의 움직임 정보를 현재 블록의 움직임 후보 목록에 포함시킬수 있는지 여부를 나타내는 블록 움직임 정보의 가용성 정보가 출력된다.

[0084] 여기서 블록 움직임 정보의 가용성 정보가 ‘1’ 이면, 해당 블록은 사용 가능하며, ‘0’ 이면 사용 불가능하며 그 역도 가능한다.

[0085] 본 발명에 따르면 주변 블록들에 대한 움직임 정보의 가용성 판단은 임의의 순서로 수행될 수 없으며 주변 블록 간에 일정한 순서 제약을 준수해야 한다.

[0086] 주변 블록들에 대한 움직임 정보의 가용성 판단이 완료 되면, 블록 움직임 정보의 가용성 정보에 기초하여 초기 움직임 후보 목록이 구성되며(S421), 도 3의 주변 블록은 A→B→C→D→E→H→M 순서로 움직임 후보 목록에 추가될 수 있다.

[0087] 그 후 움직임 후보 목록은 재구성될 수 있으며, 재구성 과정을 거쳐 최종 움직임 후보 목록이 구성될 수 있다

(S423).

[0088] 최종 움직임 후보 목록에 포함된 움직임 후보 중에서 현재 블록의 움직임 정보가 선택되고(S431), 선택된 움직임 정보를 이용하여 현재 블록에 대한 움직임 보상이 수행된다(S423). 도 4의 과정을 거쳐 움직임 보상된 영상을 출력된다.

[0089] 부호화 장치는 병합 움직임 후보 리스트에 포함된 병합 움직임 후보 중에서 최적의 부호화 효율을 제공할 수 있는 병합 움직임 후보를 현재 블록에 대한 움직임 정보로 선택할 수 있다. 병합 움직임 후보 목록 중에서 현재 블록이 병합 움직임 할 후보에 대한 인덱스를 병합 움직임 인덱스(merge_idx, 이하 'merge_idx'라 함)라하며, 상기 선택된 머지 인덱스가 비트스트림에 포함되어 복호화 장치로 전송될 수 있다.

[0090] 병합 움직임 후보 목록 내에 동일한 움직임 정보가 존재하는 경우에는, 동일한 움직임 정보들 중에서 하나만 병합 움직임 후보 목록에 남겨둔다. 병합 움직임 후보 목록 내의 움직임 정보의 수는 소정의 개수로 설정할 수 있다. 예컨대, 병합 움직임 후보 목록 내의 움직임 정보의 수를 최대 5개(혹은 임의의 수)가 되도록 설정할 수 있다.

[0091] 복호화 장치는 전송된 머지 인덱스를 이용하여, 병합 움직임 후보 리스트에 포함된 머지 후보 중에서 하나를 선택할 수 있으며, 상기 선택된 병합 움직임 후보를 현재 블록의 움직임 정보로 결정할 수 있다. 따라서, 병합 움직임 모드가 적용되는 경우, 복원된 주변 블록 및/또는 콜 블록의 움직임 정보가 현재 블록의 움직임 정보로 그대로 사용될 수 있다. 복호화 장치는 예측 블록과 부호화 장치로부터 전송되는 레지듀얼을 더하여 현재 블록을 복원할 수 있다.

[0092] 병합 움직임 인덱스 값이 지시되지 않는다면, '0'을 갖는 것으로 추정(infer)될 수 있다.

[0094] 도 5는 일 예에 따라 하나의 CU가 2개의 PU로 나누어졌을 경우, 병합 움직임 후보 목록을 설정하는 일예를 나타낸 도면이다. 도시된 바와 같이, CU는 Nx2N으로 파티셔닝되었으며, 왼쪽에 있는 첫 번째 예측 블록, PU 0은 PartIdx가 0이고, 오른쪽에 있는 두 번째 예측 블록 PU 1은 PartIdx가 1로 식별된다.

[0095] PU에 대한 주변 블록의 움직임 정보는 현재 블록에 인접한 주변 블록(A0, A1, B0, B1, B2)의 움직임 정보이거나 참조 영상에서 현재 블록에 대응되는(collocated) 블록(미도시)의 움직임 정보일 수 있다. 도 5의 (a)는 PU 0에 대한 병합 움직임 후보 목록을 구성하기 위해 사용되는 주변 블록들을 나타내고, 도 5의 (b)는 PU 1에 대한 병합 움직임 후보 목록을 구성하기 위해 사용되는 주변 블록들을 나타낸다.

[0096] PU 1의 병합 움직임 후보 목록 구성 시, 동일 CU내의 다른 PU의 움직임 정보를 참조하지 않도록 함으로써다른 PU와 병렬적으로 병합 움직임 후보 목록을 구성할 수 있으며, 이를 위해 도 5의 (b)에서 PU 0에 위치한 A1 블록은 PU 1의 병합 움직임 후보 목록에 포함되지 않을 수 있다. 즉, 도 5의 (b)에서 PU 0에 위치한 A1 블록은 PU 1의 병합 움직임 후보 목록 구성에 사용 가능하지 않으며, 병합 움직임 후보 목록 구성 시 A1 블록의 움직임 정보에 접근할 수 없다는 것을 의미할 수 있다(혹은 움직임 정보가 존재하지 않는다는 것을 의미할 수 있으며, 혹은 블록 움직임 정보의 가용성이 없음을 의미할 수 있음).

[0098] 도 6은 다른 예에 따라 하나의 CU가 2개의 PU로 나누어졌을 경우, 병합 움직임 후보 목록을 설정하는 일예를 나타낸 도면이다. 도시된 바와 같이, CU는 2NxN 으로 파티셔닝되었으며, 상단의 첫 번째 예측 블록, PU 0은 PartIdx가 0이고, 오른쪽에 있는 하단의 예측 블록 PU 1은 PartIdx가 1로 식별된다.

[0099] PU에 대한 주변 블록의 움직임 정보는 현재 블록에 인접한 주변 블록(A0, A1, B0, B1, B2)의 움직임 정보이거나 참조 영상에서 현재 블록에 대응되는(collocated) 블록(미도시)의 움직임 정보일 수 있다. 도 6의 (a)는 PU 0에 대한 병합 움직임 후보 목록을 구성하기 위해 사용되는 주변 블록들을 나타내고, 도 6의 (b)는 PU 1에 대한 병합 움직임 후보 목록을 구성하기 위해 사용되는 주변 블록들을 나타낸다.

[0100] PU 1의 병합 움직임 후보 목록 구성 시, 동일 CU내의 다른 PU의 움직임 정보를 참조하지 않도록 함으로써 다른 PU와 병렬적으로 병합 움직임 후보 목록을 구성할 수 있으며, 이를 위해 도 6의 (b)에서 PU 0에 위치한 B1 블록은 PU 1의 병합 움직임 후보 목록에 포함되지 않을 수 있다. 즉, 도 6의 (b)에서 PU 0에 위치한 B1 블록은 PU 1의 병합 움직임 후보 목록 구성에 사용 가능하지 않으며, 병합 움직임 후보 목록 구성 시 B1 블록의 움직임 정보에 접근할 수 없다는 것을 의미할 수 있다(혹은 움직임 정보가 존재하지 않는다는 것을 의미할 수 있으며, 혹

은 블록 움직임 정보의 가용성이 없음을 의미할 수 있음).

[0102] 한편, 본 발명은 움직임 예측을 위한 움직임 후보 목록을 구성함에 있어, 움직임 후보 목록이 구성되는 대상 블록(현재 블록이라 칭함)과 인접한 주변 블록들에 대한 움직임 후보 가용성을 유도하는 방법 및 장치에 관한 것이다.

[0103] 도 7은 현재 블록의 주변 블록들의 일예를 도시한 도면이다.

[0104] 도 7을 참조하면 현재 블록(X)과 인접한 주변 블록들은 좌측 블록들(A0, A1), 상측 블록들(B0, B1), 그리고 모서리 블록(C)으로 구분할 수 있다. 본 발명은 이러한 주변 블록들 대해 움직임 후보로 사용 가능한지에 대한 움직임 정보 가용성을 결정하는 순서를 제약함으로써 동일 부호화 유닛(CU)에 속한 예측 유닛(PU) 들의 움직임 후보 목록(움직임 후보 목록)을 병렬적으로 구성하는 것을 가능하게 한다.

[0106] 도 8은 움직임 정보 가용성을 결정하는 순서를 제약하는 방법에 대한 여러 예를 도시한 도면이다.

[0107] 도 8을 참조하면 본 발명에 의한 움직임 정보 가용성 결정 순서 제약 방식은 (a) 상측 우선 방식, (b) 좌측 우선 방식, (c) 상측/좌측 병렬 방식이 있을 수 있다.

[0108] 상측 우선 방식은 상측 블록들 중 하나에 대한 움직임 정보 가용성을 다른 주변 블록들 보다 가장 우선하여 결정하는 방식이고, 좌측 우선 방식은 좌측 블록들 중 하나에 대한 움직임 정보 가용성을 다른 주변 블록들 보다 가장 우선하여 결정하는 방식이다.

[0109] 움직임 정보 가용성 결정 과정의 병렬성을 높이기 위해 (c)와 같이 상측/좌측 병렬 결정 방식을 사용할 수도 있으며, 이때는 좌측 블록들에 대한 움직임 정보 가용성 결정과 상측 블록들에 대한 움직임 정보 가용성 결정을 병렬로 수행할 수 있는 장점이 있다.

[0110] 중복된 움직임 정보를 가진 주변 블록이 움직임 후보 목록에 포함되지 않도록 각 주변 블록의 움직임 정보 가용성 결정 과정에 움직임 정보의 동일성을 판단하는 움직임 정보 동일성 판단 과정이 포함될 수 있다.

[0111] 예컨대, 도 7의 B0에 대한 움직임 정보 가용성 결정 시 B0의 움직임 정보가 B1의 움직임 정보와 동일한 경우에는 B0에 대한 움직임 정보 가용하지 않다고 결정할 수 있다.

[0112] 본 발명에 따르면, 움직임 정보 가용성 결정 과정에 움직임 정보 동일성 판단 과정이 포함되는 경우에도 도 8과 같이 움직임 정보 가용성 결정 순서에 제약을 받아야만 한다. 예를 들어 주변 블록들에 대한 움직임 정보 가용성 결정 순서 상 B1이 B0 보다 우선하는 경우에만 B0의 움직임 정보가 B1의 움직임 정보와 동일한 지 여부에 따라 B0의 움직임 정보 가용성을 결정할 수 있다.

[0113] 반대의 경우, 즉 주변 블록들에 대한 움직임 정보 가용성 결정 순서 상 B0가 B1 보다 우선하는 경우에는 B0의 움직임 정보가 B1의 움직임 정보와 동일한 지에 따라 B0의 움직임 정보 가용성을 결정할 수 없다.

[0114] 본 발명에서는 도 8의 세 가지 움직임 정보 가용성 결정 순서 제약 방식들 중 (b) 좌측 우선 결정 방식을 예로 하여 설명된다. 하지만, 주변 블록의 가용성 판단 과정을 포함하는 상기 설명된 내용과 후술될 내용은 (a) 상측 우선 결정 방식 및 (c) 상측/좌측 병렬 결정 방식에 적용될 수 있으며, 이는 동 분야의 통상의 지식을 가진 사람들에게는 자명하다고 할 수 있다.

[0116] 본 발명에 의하면, 도 5의 (b)의 PU 1의 병합 움직임 후보 목록을 구성하는데 있어서, 주변 블록 B1은 주변 블록 A1 이 사용 가능하지 않다는 것을 미리 알 수 있다. 따라서, 도 5의 (b)의 PU 1의 병합 움직임 후보 목록을 구성하는데 있어서, 주변 블록 A1 의 가용성 정보를 확인하여 주변 블록 B1 과 주변 블록 A1 이 동일한 움직임 정보를 가지는지를 확인하지 않아도 된다.

[0117] 다시 말하면, 도 5의 (b)의 PU 1의 병합 움직임 후보 목록을 구성하기 위해서 PU 1과 동일 CU 내에 속한 주변 블록 A1 의 움직임 정보에 접근하지 않아도 되므로 PU 0와 PU 1의 병합 움직임 후보 목록을 병렬적으로 구성할 수 있다.

[0118] 유사하게, 도 6의 (b)의 PU 1의 병합 움직임 후보 목록을 구성하는데 있어서, 주변 블록 B0은 주변 블록 B1 이

사용 가능하지 않다는 것을 미리 알 수 있다. 따라서, 도 6의 (b)의 PU 1의 병합 움직임 후보 목록을 구성하는데 있어서, 주변 블록 B1 의 가용성 정보를 확인하여 주변 블록 B1 과 주변 블록 B0 이 동일한 움직임 정보를 가지는지를 확인하지 않아도 된다.

[0119] 다시 말하면, 도 6의 (b)의 PU 1의 병합 움직임 후보 목록을 구성하기 위해서 PU 1과 동일 CU 내에 속한 주변 블록 B1 의 움직임 정보에 접근하지 않아도 되므로 PU 0와 PU 1의 병합 움직임 후보 목록을 병렬적으로 구성할 수 있다.

[0121] PU의 병합 움직임 후보 목록을 병렬적으로 구성하면서 주변 블록들에 대한 움직임 정보 가용성 결정 순서를 제약하는 기 위한 구체적인 방법으로 순차적으로 블록 움직임 정보의 가용성 정보를 판단하는 방법과 주변 블록의 가용성 판단 과정 후 블록 가용성 정보를 재설정하는 방법이 존재할 수 있다.

[0123] [방법 1] 병합 움직임 후보 목록을 구성하는데 있어서, 순차적으로 블록 움직임 정보의 가용성 정보를 판단하는 방법의 일 예

[0124] 도 9는 순차적으로 블록 움직임 정보의 가용성 정보를 판단하는 경우 주변 블록을 도시한 도면으로, 현재 블록(X)의 주변 블록(A, B, C, D, E)을 도시하고 있다.

[0125] 본 발명에서 순차적이란, 현재 혹은 다음 순서에 해당하는 정보가 이전 순서에 해당하는 정보에 의존적인 것을 의미할 수 있다.

[0126] 주변 블록에 대한 움직임 정보로 사용 가능한지 여부(블록 움직임 정보의 가용성 정보)에 대한 판단은 기설정된 순서로 이루어질 수 있다. 좌측 블록들 중 하나에 대한 움직임 정보 가용성을 다른 주변 블록들 보다 가장 우선하여 결정하는 좌측 우선 결정 방식을 따를 경우, A→B→C→D→E 블록의 순서로 해당 블록의 움직임 정보가 현재 블록의 병합 움직임에 이용될 수 있는지를 판단하고 이용 가능한 경우에는 해당 블록의 움직임 정보를 병합 움직임 후보 목록에 입력할 수 있다.

[0127] 도 10은 하나의 CU가 복 수의 PU로 파티셔닝되는 예를 도시한 것이다.

[0128] 도 10에 도시된 바와 같이, CU는 파티셔닝 되지 않고 2Nx2N의 PU일 수도 있고, 동일한 크기인 2NxN, Nx2N, NxN으로 파티셔닝 될 수도 있다. 또한, CU는 2NxnU, 2NxnD, nLx2N, nRx2N 와 같이 비대칭적으로 파티셔닝 될 수도 있다.

[0130] [방법 2] 병합 움직임 후보 목록을 구성하는데 있어서, 블록 가용성 정보를 재설정하는 방법의 일 예

[0131] 도 11은 본 발명의 일 실시예에 따라 블록 가용성 정보를 재설정하는 방법을 설명하기 위한 제어 흐름도이다. 도 11은 도 4의 주변 블록들에 대한 가용성 판단 단계(S410)에서 블록 가용성 정보를 재설정하는 방법의 구조도의 일예를 나타낸다.

[0132] 도 11을 참조하면, 우선 주변 블록들이 존재하는지를 여부에 대한 주변 블록의 가용성이 판단된다(S1110). 판단 결과로 블록 가용성 정보가 출력된다.

[0133] 그런 후, 주변 블록의 블록 가용성 정보가 재설정된다(S1120)).

[0134] 일 예로, 도 5의 (b) PU 1에 대하여, 병합 움직임 후보 목록을 다른 PU와 병렬적으로 구성하기 위해서 도 5의 (b) PU 0에 위치한 A1 블록은 PU 1의 병합 움직임 후보 목록에 포함될 수 없다.

[0135] 따라서 도 5의 (b) PU 0에 위치한 A1 블록의 블록 가용성 정보를 '0'으로 설정하여 해당 블록을 사용할 수 없게 할 수 있다.

[0136] 다음으로, 설정된 블록 가용성 정보를 바탕으로 주변 블록의 움직임 정보에 대한 가용성을 판단한다(S1130). 단계 S1130을 통하여 주변 블록의 움직임 정보를 현재 블록의 움직임 후보 목록에 포함시킬수 있는지 여부를 나타내는 블록 움직임 정보의 가용성 정보가 출력된다.

[0137] 블록 움직임 정보의 가용성 정보를 바탕으로 초기 움직임 후보 목록이 구성된다.

[0138] 도 11의 블록 가용성 정보를 재설정하는 방법은 하나의 CU가 다수개의 PU로 분할 될 경우 (일예로, 도 10에서의 2NxN, Nx2N, NxN, 2NxnU, 2NxnD, nLx2N, nRx2N), 즉 하나의 CU에서 분할된 PU에 적용할 수 있다.

[0140] 상술한 방법 1을 이용한 복호화 과정의 일례를 나타내면 아래와 같다.

[0141] 이하, 본 발명에서 사용되는 용어에 대한 정의 다음과 같다.

[0142] PredMode는 현재 PU(Prediction Unit) 블록의 부호화 모드를 나타내며, MODE_SKIP, MODE_INTRA, MODE_INTER 가 존재한다.

[0143] Merge_flag는 현재 PU블록이 머지 모드로 부호화되었는지를 나타낸다. 현재 PU 블록이 머지 모드로 부호화되었다면, Merge_flag는 '1'로 설정되며, 그 반대로도 설정할 수 있다.

[0144] merge_idx는 움직임 정보 후보 목록에서 현재 PU 블록에 대한 움직임 정보를 가르키는 인덱스를 나타낸다.

[0145] log2_parallel_merge_level_minus2는 병렬 머지 스킵의 레벨을 정의한다. 본 발명에 따를 경우 머지 스킵은 화면간 예측에 적용되는 모드 중 머지모드와 스킵모드를 동시에 지칭하는 용어로 해석될 수 있으며, 병렬 머지 스킵은 머지 모드 및/또는 스킵 모드를 적용하는 경우 특정 영역 내의 블록들 간에 병합 움직임 후보 목록을 병렬적으로 구성할 수 있는 것을 의미할 수 있다.

[0146] 예를 들어, log2_parallel_merge_level_minus2의 값이 '0'이면, 병렬 머지 스킵이 수행되지 않는다. log2_parallel_merge_level_minus2의 값이 '1'이면, 8x8 블록 안의 있는 모든 PU에 대하여 병렬 머지 스킵이 수행된다. log2_parallel_merge_level_minus2의 값이 '2'이면, 16x16 블록 안의 있는 모든 PU에 대하여 병렬 머지 스킵이 수행되고, log2_parallel_merge_level_minus2의 값이 '3'이면, 32x32 블록 안의 있는 모든 PU에 대하여 병렬 머지 스킵이 수행된다. log2_parallel_merge_level_minus2의 값이 '4'이면, 64x64 블록 안의 있는 모든 PU에 대하여 병렬 머지 스킵이 수행된다.

[0148] Z-스캔 가용성 유도 과정

[0149] [1] Z-스캔 가용성 유도 과정에서 입력은 아래와 같다.

[0150] (xCurr, yCurr): 현재 픽쳐의 좌상단 위치와 관련된 현재 블록의 좌상단 위치;

[0151] (xN, yN): 현재 픽쳐의 좌상단 위치와 관련된 주변 블록의 좌상단 위치

[0152] Z-스캔 가용성 유도 과정에서 출력은 아래와 같다.

[0153] availableN: (xN, yN) 위치의 주변 블록의 가용성 정보

[0155] [2] minBlockAddrCurr을 현재 블록에 대한 z-스캔 순서로 최소 휘도 블록의 주소라고 설정한다. 이 때 minBlockAddrCurr는 아래와 같다.

[0156] minBlockAddrCurr= MinTbAddrZS[xCurr>>Log2MinTrafoSize][yCurr>>Log2MinTrafoSize]

[0157] (여기서 MinTbAddrZS는 최소 변환 블록(transform block)에서의 z-스캔 순서 배열을 의미하며, log2MinTrafoSize는 최소 변환 블록의 크기를 의미한다.)

[0159] [3] minBlockAddrN을 (xN, yN) 위치를 가지는 주변 블록에 대한 z-스캔 순서로 최소 휘도 블록의 주소로 설정한다.

[0160] 만일 아래 조건((1) 내지 (4)) 중에 하나라도 '참'이면, minBlockAddrN을 '-1'로 설정한다.

[0161] (1) xN이 '0'보다 작다.

[0162] (2) yN이 '0'보다 작다.

[0163] (3) xN이 pic_width_in_luma_sample(영상의 너비)보다 크다.

[0164] (4) yN이 pic_height_in_luma_sample(영상의 너비)보다 크다.

[0165] 그렇지 않다면(즉, (xN, yN) 위치가 픽쳐 경계 안이라면), 아래를 적용한다.

[0166] minBlockAddrN = MinTbAddrZS[xN >> Log2MinTrafoSize][yN >> Log2MinTrafoSize]

[0168] [4] 주변 블록 가용성 정보 availableN은 아래 과정을 통해 유도된다.

[0169] 만일 아래 조건((1) 내지 (4))중에 하나라도 '참(true)'이면, availableN은 '거짓(false)'으로 설정한다.

[0170] (1) minBlockAddrN이 '0'보다 작다.

[0171] (2) minBlockAddrN이 minBlockAddrCurr 보다 크다.

[0172] (3) minBlockAddrN을 가지는 주변 블록을 포함하고 있는 슬라이스의 주소 인덱스가 minBlockAddrCurr을 가지는 현재 블록을 포함하고 있는 슬라이스의 주소 인덱스와 다르다.

[0173] (4) minBlockAddrN을 가지는 주변 블록이 minBlockAddrCurr을 가지는 현재 블록과 다른 타일(Tile) 에 위치한다. (여기서 타일 이란, 영상을 슬라이스로 나누는 것과 동일한 개념이며, 타일 간에는 블록의 부호화 정보를 유도하지 않는다.)

[0174] 그렇지 않다면, availableN은 '참'으로 설정한다.

[0176] 예측 블록(prediction block)의 가용성 유도 과정

[0177] [1] 예측 블록(prediction block)의 가용성 유도 과정에서 입력은 아래와 같다.

[0178] (xC, yC): 현재 픽처의 좌상단 위치와 관련된 현재 휘도 부호화 블록의 좌상단 위치;

[0179] (nCS): 현재 CU의 크기를 가르키는 변수;

[0180] (xP, yP): 현재 픽처의 좌상단 위치와 관련된 현재 휘도 예측 블록의 좌상단 위치;

[0181] (nPbW, nPbH): 현재 CU블록 내에서 현재 PU의 너비와 높이를 가르키는 변수;

[0182] (partIdx): 현재 CU블록 내에서 현재 PU의 인덱스;

[0183] (xN, yN): 현재 픽처의 좌상단 위치와 관련된 주변 블록의 위치;

[0184] 예측 블록(prediction block)의 가용성 유도 과정에서 출력은 아래와 같다.

[0185] availableN: (xN, yN) 위치를 가지는 주변 블록의 가용성 정보

[0187] [2] 변수 sameCb를 현재 휘도 예측 블록과 주변 휘도 예측 블록이 동일한 휘도 부호화 블록인지를 나타내는 것으로 정의한다.

[0188] 만일 아래 조건들((1) 내지 (4))이 모두 '참'이면, sameCb를 '참'으로 설정한다.

[0189] (1) xC가 xN보다 작거나 혹은 같다.

[0190] (2) yC가 yN보다 작거나 혹은 같다.

[0191] (3) (xC + nCbS)가 xN보다 크다.

[0192] (4) (yC + nCbS)가 yN보다 크다.

[0193] 그렇지 않다면, sameCb는 '거짓'으로 설정한다.

[0194] 여기서 sameCb가 '참'일 경우, 현재 휘도 예측 블록과 주변 휘도 예측 블록이 동일한 휘도 부호화 블록에 속하는지를 의미할 수 있으며, 또는 그 역도 가능하다.

[0196] [3] 주변 예측 블록의 가용성 정보 availableN은 다음과 같이 유도된다.

[0197] 만일 sameCb가 ‘거짓’이라면, z-스캔 가용성 유도 과정을 적용하여, availableN을 설정한다.

[0198] 그렇지 않고, 만일 아래 조건((1) 내지 (5))이 이 모두 ‘참’이면, availableN을 ‘거짓’으로 설정한다.

[0200] *(1) (nPbW << 1)이 nCbS와 같다.

[0201] (2) (nPbH << 1)이 nCbS와 같다.

[0202] (3) partIdx 이 ‘1’이다.

[0203] (4) (yC + nPbH) 이 yN 보다 작거나 같다.

[0204] (5) (xC + nPbW) 이 xN보다 크다.

[0205] 그렇지 않다면, availableN을 ‘참’으로 설정한다.

[0207] availableN이 ‘참’이고 PredMode[xN][yN]이 MODE_INTRA일 경우, availableN을 ‘거짓’으로 설정한다.

[0209] 움직임 벡터와 참조 픽쳐 인덱스를 유도하는 과정

[0210] [1] 움직임 벡터와 참조 픽쳐 인덱스를 유도하는 과정에서 입력은 아래와 같다.

[0211] (xC yC): 현재 CU(Coding Unit) 블록의 좌상단 위치;

[0212] (xB, yB): 현재 PU의 좌상단 위치;

[0213] (nCS): 현재 CU의 크기를 가르키는 변수;

[0214] (nPSW, nPSH): 현재 PU의 너비와 높이를 가르키는 변수;

[0215] (partIdx): 현재 CU블록 내에서 현재 PU의 인덱스

[0216] 움직임 벡터와 참조 픽쳐 인덱스를 유도하는 과정에서 출력은 아래와 같다.

[0217] (mvL0와 mvL1): 휘도 샘플에 대한 움직임 벡터;

[0218] (mvCL0와 mvCL1): 색차 샘플에 대한 움직임 벡터;

[0219] 참조 픽쳐 인덱스; refIdxL0와 refIdxL1

[0220] predFlagL0와 predFlagL1: 예측 목록 이용 가능 여부(L0혹은 L1예측을 수행하는지)를 알려주는 변수

[0222] [2] 현재 PU블록의 위치를 xP = xC + xB, yP = yC + yB 로 설정한다.

[0224] [3] mvL0, mvL1, refIdxL0, refIdxL1, PredFlagL0, PredFlagL1은 아래 과정을 통하여 유도된다.

[0225] 만일 PredMode가 MODE_SKIP이라면, 머지 모드를 위한 휘도 움직임 벡터 유도과정을 수행한다. 여기서 입력은 (xC, yC), (xP, yP), nCS, nPSW, nPSH, partIdx 이며, 출력은 mvL0, mvL1, refIdxL0, refIdxL1, PredFlagL0, PredFlagL1이다.

[0226] 그렇지 않고, 만일 PredMode가 MODE_INTER이고 현재 PU블록에 대한 merge_flag가 ‘1’이라면, 머지 모드를 위한 휘도 움직임 벡터 유도 과정을 수행한다. 여기서 입력은 (xC, yC), (xP, yP), nCS, nPSW, nPSH, partIdx 이며, 출력은 mvL0, mvL1, refIdxL0, refIdxL1, PredFlagL0, PredFlagL1이다.

[0227] 그렇지 않다면, mvLX, refIdxLX, PredFlagLX는 신택스(syntax) 요소 ref_idx_lx, mvd_lX에 의해 설정된다. 여기서 X는 ‘0’ 혹은 ‘1’로 교체될 수 있다.

[0229] 머지 모드를 위한 휘도 움직임 벡터 유도 과정

[0230] [1] 머지 모드를 위한 휘도 움직임 벡터 유도 과정에서 입력은 아래와 같다.

[0231] (xC yC): 현재 CU(Coding Unit) 블록의 좌상단 위치;

[0232] (xP, yP): 현재 PU의 좌상단 위치;

[0233] (nCS): 현재 CU의 크기를 가르키는 변수;

[0234] (nPSW, nPSH): 현재 PU의 너비와 높이를 가르키는 변수;

[0235] (partIdx): 현재 CU블록 내에서 현재 PU의 인덱스;

[0236] 머지 모드를 위한 휘도 움직임 벡터 유도 과정에서 출력은 아래와 같다.

[0237] (mvL0와 mvL1): 휘도 샘플에 대한 움직임 벡터;

[0238] refIdxL0와 refIdxL1: 참조 픽쳐 인덱스;

[0239] predFlagL0와 predFlagL1: 예측 목록 이용가능 여부(L0혹은 L1예측을 수행하는지)를 알려주는 변수

[0241] [2] 변수 singleMCLFlag(공통의 머지 후보 목록 플래그)는 아래와 같이 유도된다.

[0242] 만일 log2_parallel_merge_level_minus2 의 값이 '0' 보다 크고 nCS 값이 '8' 이면, singleMCLFlag 에 '1' 을 설정한다.

[0243] 그렇지 않다면, singleMCLFlag 에 '0' 을 설정한다.

[0245] [3] 현재 PU의 좌상단 위치(xP, yP), 현재 PU의 너비와 높이를 가르키는 변수(nPSW, nPSH)는 아래와 같이 유도된다.

[0246] 만일 singleMCLFlag 가 '1' 이면, xP에 xC를 입력하고 yP에 yC를 입력하고 nPSW와 nPSH에 nCS를 입력한다.

[0247] 부연 설명: singleMCLFlag 의 값이 '1' 로 설정되면, 하나의 CU안의 모든 PU블록은 파티션 형태에 상관없이 2Nx2N PU에 대한 머지 후보 목록을 공유한다.

[0249] [4] 머지 부호화 모드에서 사용할 머지 후보 목록을 구성하고 현재 PU에 대한 움직임 정보(mvL0, mvL1, refIdxL0, refIdxL1, PredFlagL0, PredFlagL1)를 설정하는 단계는 아래와 같은 순서로 수행된다.

[0250] [4-1] 현재 PU을 포함하는 현재 CU의 주변 PU으로부터 머지 움직임 후보를 설정하는 과정을 수행한다.

[0251] 여기서 입력은 (xP, yP), nPSH, nPSW, singleMCLFlag 이며, 출력은 availableFlagN(주변 PU의 사용 가능 여부를 알려주는 변수), mvL0N, mvL1N, refIdxL0N, refIdxL1N, PredFlagL0N, PredFlagL1N이다. (여기서 N은 도 12의 A1, B1, B0, A0, B2으로 교체될 수 있으며, X는 '0' 혹은 '1' 로 교체될 수 있다.)

[0252] [4-2] 시간적 머지 움직임 후보를 사용하기 위한 참조 픽쳐 인덱스 refIdxLX 는 '0' 으로 설정된다. 여기서 X는 '0' 혹은 '1' 로 교체될 수 있다.

[0253] [4-3] 시간적으로 유추된 움직임 정보를 머지 움직임 후보로 설정하는 과정을 수행한다. 여기서 입력은 (xP, yP), nPSH, nPSW, refIdxLX이며, 출력은 availableFlagLXCol(시간적으로 유추된 움직임 정보의 사용 가능 여부를 알려주는 변수), mvLXCol(시간적으로 유추된 움직임 정보)이다.

[0254] availableFlagCol(시간적으로 유추된 움직임 정보의 사용 가능 여부를 알려주는 변수)과 PredFlagLXCol(시간적으로 유추된 움직임 정보에 대한 예측 목록 이용가능 여부(L0혹은 L1예측을 수행하는지)를 알려주는 변수)는 아래와 같이 설정된다.

[0255] availableFlagCol = availableFlagL0Col || availableFlagL1Col

[0256] PredFlagLXCol = availableFlagLXCol

[0258] [4-4] 다음으로, 머지 후보 목록을 아래와 같이 구성한다.

[0260] *도 12는 현재 블록의 머지 후보 리스트 생성을 설명하기 위한 현재 블록 및 주변 블록을 도시한 도면이다.

[0261] 현재 블록, 즉 예측 대상 블록의 좌측 상단에 존재하는 픽셀의 위치를 (xP, yP)라고 하고 예측 대상 블록의 너비를 nPSW, 예측 대상 블록의 높이를 nPSH라는 변수로 정의한다.

[0262] 이하, 본 발명의 실시예에서는 현재 블록의 공간적 주변 예측 블록은 (xP-1, yP+nPSH-1)에 존재하는 픽셀을 포함하는 블록을 좌측 A1 블록, (xP-1, yP+nPSH)에 존재하는 픽셀을 포함하는 블록을 A0 블록, (xP+nPSW, yP-1)에 위치하는 픽셀을 포함하는 블록을 B0 블록, (xP+nPSW-MinPuSize, yP-1)에 위치하는 픽셀을 포함하는 블록을 B1 블록, (xP-MinPuSize, yP-1)에 위치하는 픽셀을 포함하는 블록을 B2 블록을 포함할 수 있다.

[0263] 도 12의 A1, B1, B0, A0, B2, Col 순으로 해당블록의 움직임 정보가 사용가능할 경우, 해당 블록의 움직임 정보를 머지 후보 목록에 차례대로 입력한다. 여기서 Col은 시간적으로 유추된 움직임 정보를 말한다.

[0264] [4-5] 머지 후보 목록 내에서 동일한 움직임 정보가 존재할 경우, 해당 움직임 정보를 하나만 남겨두는 단계를 수행한다.

[0265] [4-6] 머지 후보 목록의 움직임 정보 개수를 설정하고 최대 머지 후보 목록의 개수를 '5' 로 설정하는 단계를 수행한다.

[0266] [4-7] B 픽쳐일때, 결합 양예측 머지 후보를 추가하는 단계를 수행한다.

[0267] [4-8] 영(0,0) 움직임 정보를 머지후보로 추가하는 단계를 수행한다.

[0268] [4-9] 머지 후보 목록에서 merge_idx에 존재하는 움직임 정보를 현재 PU의 움직임 정보로 설정하는 단계를 수행한다.

[0270] 현재 PU의 주변 PU(공간적 PU)으로부터 머지 움직임 후보를 설정하는 단계

[0271] [1] 현재 PU의 주변 PU(공간적 PU)으로부터 머지 움직임 후보를 설정하는 단계에서 입력은 아래와 같다.

[0272] (xC yC): 현재 CU(Coding Unit) 블록의 좌상단 위치;

[0273] (xP, yP): 현재 PU의 좌상단 위치;

[0274] (nCS): 현재 CU의 크기를 가르키는 변수;

[0275] (nPSW, nPSH): 현재 PU의 너비와 높이를 가르키는 변수;

[0276] singleMCLFlag: 현재 CU에 대한 공통의 머지 후보 목록의 사용여부를 나타내는 변수;

[0277] (partIdx): 현재 CU블록 내에서 현재 PU의 인덱스;

[0278] 현재 PU의 주변 PU(공간적 PU)으로부터 머지 움직임 후보를 설정하는 단계에서 출력은 아래와 같다. (여기서 N은 도 12의 A1, B1, B0, A0, B2으로 교체될 수 있으며, X는 '0' 혹은 '1' 로 교체될 수 있다.)

[0279] availableFlagN: 주변 PU N의 움직임 정보 가용성 여부를 알려주는 변수;

[0280] refIdxLXN: 주변 PU N의 참조 인덱스;

[0281] predFlagLXN: 주변 PU N에 대한 예측 목록 이용가능 여부를 알려주는 변수;

[0282] mvLXN: 주변 PU N에 대한 움직임 벡터

[0284] [2] 아래 [3]의 과정을 통해 주변 PU A0, A1, B0, B1, 혹은 B2의 availableFlagN을 유도하기 위해서, 아래의 좌측 우선 결정 방식에 따른 유도 순서(제한)를 적용한다.

[0285] A1 은 A0, B0, B1 그리고 B2 보다 먼저 availableFlagN(여기서 N = A1)을 유도하여야 한다.

[0286] B1 은 B0와 B2 보다 먼저 availableFlagN (여기서 N = B1)을 유도하여야 한다.

[0287] B2 는 A0, A1, B0 그리고 B1 다음에 availableFlagN (여기서 N = B2)을 유도하여야 한다.

[0288] 상기한 좌측 우선 결정 방식의 순서 제약에 부합하는 주변 블록의 움직임 정보 가용성 유도 순서의 예는 다음과 같다:

[0289] A1 → B1 → A0 → B0 → B2,

[0290] A1 → B1 → B0 → A0 → B2

[0291] A1 → B1 → A0 및 B0 (병렬 유도) → B2

[0292] A1 → A0 → B1 → B0 → B2

[0293] A1 → A0 및 B1 (병렬 유도) → B0 → B2

[0295] [3] 현재 PU의 주변 PU(공간적 PU)이 머지 움직임 후보로 사용 가능한지를 판단하기 위한 과정은 아래의 내용을 적용한다. (여기서 주변 PU은 도 12의 A1, B1, B0, A0, B2을 말하며, N은 도 12의 A1, B1, B0, A0, B2으로 교체될 수 있으며, (xN, yN) 은 (xC-1, yC+nCS), (xC-1, yC+nCS-1), (xC+nCS, yC-1), (xC+nCS-1, yC-1), (xC-1, yC-1)으로 교체될 수 있다.)

[0296] [3-1] 주변 PU의 가용성을 판단하는 과정을 수행하며, 입력은 (xC, yC), (xP, yP), nPbW, nPbH, (xN, yN) 그리고 partIdx 이며, 출력은 (xN, yN)에 위치한 PU의 가용성 정보 availableN 이다.

[0297] [3-3] 만일 아래 조건((1) 내지 (10))중에서 하나라도 '참' 이면, availableFlagN 이 '0'으로 설정되고 mvLXN은 '0,0'으로 설정된다.

[0298] (1) (xP >> (log2_parallel_merge_level_minus2 +2))가 (xN>>(log2_parallel_merge_level_minus2+2))와 같고 (yP>>(log2_parallel_merge_level_minus2+2))가 (yN>>(log2_parallel_merge_level_minus2 +2))와 같다.

[0299] (2) N이 B2이고 availableFlagA0 + availableFlagA1 + availableFlagB0 + availableFlagB1이 '4'이다.

[0300] (3) [xN, yN]의 위치에 있는 PU가 사용 불가능하거나 혹은 화면 내 모드로 부호화되었다. 즉, availableN 이 '0'이다.

[0301] (4) singleMCLFlag가 '0'이고 현재 PU의 파티션 형태가 PART_2NxN 혹은 PART_2NxnU 혹은 PART_2NxnD이고 PartIdx가 '1'이고 N이 'B1'이다.

[0302] (5) singleMCLFlag가 '0'이고 현재 PU의 파티션 형태가 PART_Nx2N 혹은 PART_nLx2N 혹은 PART_nRx2N 이고 PartIdx가 '1'이고 N이 'A1'이다.

[0303] (6) 움직임 정보 동일성 판단 과정으로, N이 B1이고, (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0304] (7) 움직임 정보 동일성 판단 과정으로, N이 B0 이고, (xB1, yB1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0305] (8) 움직임 정보 동일성 판단 과정으로, N이 A0 이고, (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0306] (9) 움직임 정보 동일성 판단 과정으로, N이 B2 이고, (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0307] (10) 움직임 정보 동일성 판단 과정으로 N이 B2 이고, (xB1, yB1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0308] 그렇지 않으면, availableFlagN은 '1'로 설정되며, mvLXN, refIdxLXN, predFlagLXN은 각각 mvLX[xN, yN], refIdxLX[xN, yN], predFlagLX[xN, yN]으로 설정된다.

[0310] 방법 1의 현재 PU의 주변 PU(공간적 PU)으로부터 머지 움직임 후보를 설정하는 단계는 본 발명의 일 예에 따라 아래와 같이 변형될 수 있다.

[0311] [1] 현재 PU의 주변 PU(공간적 PU)으로부터 머지 움직임 후보를 설정하는 단계의 입력은 아래와 같다.

[0312] (xC yC): 현재 CU(Coding Unit) 블록의 좌상단 위치;

[0313] (xP, yP): 현재 PU의 좌상단 위치;

[0314] (nCS): 현재 CU의 크기를 가르키는 변수;

[0315] (nPSW, nPSH): 현재 PU의 너비와 높이를 가르키는 변수;

[0316] singleMCLFlag: 현재 CU에 대한 공통의 머지후보목록의 사용여부를 나타내는 변수;

[0317] (partIdx): 현재 CU블록 내에서 현재 PU의 인덱스

[0318] 이 단계에서는 출력은 아래와 같다. (여기서 N은 도 102의 A1, B1, B0, A0, B2으로 교체될 수 있으며, X는 '0' 혹은 '1'로 교체될 수 있다.)

[0319] availableFlagN: 주변 PU N의 움직임 정보 가용성 여부를 알려주는 변수;

[0320] refIdxLXN: 주변 PU N의 참조 인덱스;

[0321] predFlagLXN: 주변 PU N에 대한 예측 목록 이용가능 여부를 알려주는 변수;

[0322] mvLXN: 주변 PU N에 대한 움직임 벡터;

[0324] [2] 아래 [3]의 과정을 통해 주변 PU A0, A1, B0, B1, 혹은 B2의 availableFlagN을 유도하기 위해서, 아래의 유도 순서(제한)를 적용한다.

[0325] B2 는 A0, A1, B0 그리고 B1 다음에 availableFlagN (여기서 N = B2)을 유도 한다.

[0327] [3] 현재 PU의 주변 PU(공간적 PU)이 머지 움직임 후보로 사용 가능한지를 판단하기 위한 과정은 아래와 같다. (여기서 주변 PU은 도 12의 A1, B1, B0, A0, B2일 수 있으며, N은 도 12의 A1, B1, B0, A0, B2으로 교체될 수 있으며, (xN, yN) 은 (xC-1, yC+nCS), (xC-1, yC+nCS-1), (xC+nCS, yC-1), (xC+nCS-1, yC-1), (xC-1, yC-1)으로 교체될 수 있다.)

[0328] [3-1] 주변 PU의 가용성을 판단하며, 이 단계에서 입력은 (xC, yC), (xP, yP), nPbW, nPbH, (xN, yN) 그리고 partIdx 이며, 출력은 (xN, yN)에 위치한 PU의 가용성 정보 availableN 이다.

[0329] [3-2] 만일 아래 조건((1) 내지 (3)) 중에서 하나라도 '참'이면, availableN은 '0'으로 설정한다.

[0330] (1) 만일 (xP >> (log2_parallel_merge_level_minus2 +2))가 (xN >> (log2_parallel_merge_level_minus2 +2))와 같고 (yP >> (log2_parallel_merge_level_minus2 +2))가 (yN >> (log2_parallel_merge_level_minus2 +2))와 같다.

[0331] (2) singleMCLFlag가 '0'이고 현재 PU의 파티션 형태가 PART_2NxN 혹은 PART_2NxnU 혹은 PART_2NxnD이고 PartIdx가 '1'이고 N이 'B1'이다.

[0332] (3) singleMCLFlag가 '0'이고 현재 PU의 파티션 형태가 PART_Nx2N 혹은 PART_nLx2N 혹은 PART_nRx2N 이고 PartIdx가 '1'이고 N이 'A1'이다.

[0333] [3-2] 만일 아래 조건 중에서 하나라도 '참' 이면, availableFlagN 이 '0'으로 설정되고 mvLXN은 (0,0)으로 설정된다.

[0334] (1) N이 B2이고 availableFlagA0 + availableFlagA1 + availableFlagB0 + availableFlagB1이 '4'이다.

[0335] (2) [xN, yN]의 위치에 있는 PU이 사용불가능하거나 혹은 인트라로 부호화되었다. 즉, availableN 이

'0' 이다.

[0336] (3) 움직임 정보 동일성 판단 과정으로, N이 B1이고, (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0337] (4) 움직임 정보 동일성 판단 과정으로, N이 B0이고, (xB1, yB1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0338] (5) 움직임 정보 동일성 판단 과정으로, N이 A0이고, (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0339] (6) 움직임 정보 동일성 판단 과정으로, N이 B2이고, (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0341] *(7) 움직임 정보 동일성 판단과정): N이 B2과 같고 그리고 (xB1, yB1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0342] [3-3] 그렇지 않으면, availableFlagN은 '1' 로 설정되며, mvLXN, refIdxLXN, predFlagLXN에는 각각 mvLX[xN, yN], refIdxLX[xN, yN], predFlagLX[xN, yN]이 설정된다.

[0344] 방법 1의 현재 PU의 주변 PU(공간적 PU)으로부터 머지 움직임 후보를 설정하는 단계는 본 발명의 다른 예에 따라 아래와 같이 변형될 수 있다.

[0346] *[1] 현재 PU의 주변 PU(공간적 PU)으로부터 머지 움직임 후보를 설정하는 단계에서 입력은 아래와 같다.

[0347] (xC yC): 현재 CU(Coding Unit) 블록의 좌상단 위치;

[0348] (xP, yP): 현재 PU의 좌상단 위치;

[0349] (nCS): 현재 CU의 크기를 가르키는 변수;

[0350] (nPSW, nPSH): 현재 PU의 너비와 높이를 가르키는 변수;

[0351] singleMCLFlag: 현재 CU에 대한 공통의 머지 후보목록의 사용여부를 나타내는 변수;

[0352] (partIdx): 현재 CU블록 내에서 현재 PU의 인덱스:

[0353] 이 단계에서 출력은 아래와 같다. 여기서 N은 도 102의 A1, B1, B0, A0, B2으로 교체될 수 있으며, X는 '0' 혹은 '1' 로 교체될 수 있다.

[0354] availableFlagN: 주변 PU N의 움직임 정보 가용성 여부를 알려주는 변수;

[0355] refIdxLXN: 주변 PU N의 참조 인덱스;

[0356] predFlagLXN: 주변 PU N에 대한 예측 목록 이용가능 여부를 알려주는 변수;

[0357] mvLXN: 주변 PU N에 대한 움직임 벡터

[0359] [2] 현재 PU의 주변 PU(공간적 PU)이 머지 움직임 후보로 사용 가능한지를 판단하기 위한 과정은 아래의 내용을 적용한다. 여기서 주변 PU은 도 12의 A1, B1, B0, A0을 말하며, N은 도 12의 A1, B1, B0, A0, 로 교체될 수 있으며, (xN, yN) 은 (xC-1, yC+nCS), (xC-1, yC+nCS-1), (xC+nCS, yC-1), (xC+nCS-1, yC-1)로 교체될 수 있다.

[0360] [2-1] 주변 PU의 가용성을 판단하는 과정을 수행하며, 여기서 입력은 (xC, yC), (xP, yP), nPbW, nPbH, (xN, yN) 그리고 partIdx 이며, 출력은 (xN, yN)에 위치한 PU의 가용성 정보 availableN 이다.

[0361] [2-2] 만일 아래 조건 중((1) 내지 (10))에서 하나라도 '참' 이면, availableN은 '0' 으로 설정한다.

[0362] (1) (xP >> (log2_parallel_merge_level_minus2 +2))가 (xN >> (log2_parallel_merge_level_minus2 +2))와 같

고 (yP >> (log2_parallel_merge_level_minus2 +2))가 (yN >> (log2_parallel_merge_level_minus2 +2))와 같다.

[0363] (2) singleMCLFlag가 '0'이고 현재 PU의 파티션 형태가 PART_2NxN 혹은 PART_2NxnU 혹은 PART_2NxnD이고 PartIdx가 '1'이고 N이 'B1'이다.

[0364] (3) singleMCLFlag가 '0'이고 현재 PU의 파티션 형태가 PART_Nx2N 혹은 PART_nLx2N 혹은 PART_nRx2N 이고 PartIdx가 '1'이고 N이 'A1'이다.

[0365] (4) 만일 아래 조건 중에서 하나라도 '참' 이면, availableFlagN 이 '0'으로 설정되고 mvLXN은 '0,0'으로 설정된다.

[0366] (5) [xN, yN]의 위치에 있는 PU이 사용불가능하거나 혹은 인트라로 부호화되었다. (즉, availableN 이 '0'이다)

[0367] (6) 움직임 정보 동일성 판단 과정으로, N이 B1이고, availableA1=1이고 그리고 (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0368] (7) 움직임 정보 동일성 판단 과정으로, N이 B0이고, availableB1=1이고 그리고 (xB1, yB1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0369] (8) 움직임 정보 동일성 판단 과정으로, N이 A0 이고, availableA1=1이고 그리고 (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0370] (9) 움직임 정보 동일성 판단 과정으로, N이 B2 이고, availableA1=1이고 그리고 (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0371] (10) 움직임 정보 동일성 판단 과정으로, N이 B2 이고, availableB1=1이고 그리고 (xB1, yB1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0372] 그렇지 않으면, availableFlagN은 '1'로 설정되며, mvLXN, refIdxLXN, predFlagLXN에는 각각 mvLX[xN, yN], refIdxLX[xN, yN], predFlagLX[xN, yN]이 설정된다.

[0374] [3] A0, A1, B0 그리고 B1 다음에 B2의 availableFlagN (여기서 N = B2)이 유도된다. 현재 CU의 주변 PU(공간적 PU) 중 B2가 머지 움직임 후보로 사용 가능한지를 판단하기 위한 과정은 아래의 내용을 적용한다. 여기서 (xB2, yB2) 는 (xC?1, yC?1)로 교체될 수 있다.

[0375] [3-1] 주변 PU의 가용성을 판단하는 과정을 수행하며, 여기서 입력은 (xC, yC), (xP, yP), nPbW, nPbH, (xB2, yB2) 그리고 partIdx 이며, 출력은 (xB2, yB2)에 위치한 PU의 가용성 정보 availableB2 이다.

[0376] [3-2] 만일 아래 조건이 '참' 이면, availableFlagB2 이 '0'으로 설정되고 mvLXB2은 '0,0'으로 설정된다.

[0377] availableFlagA0 + availableFlagA1 + availableFlagB0 + availableFlagB1이 '4'이다.

[0378] 그렇지 않으면, availableFlagB2은 '1'로 설정되며, mvLXB2, refIdxLXB2, predFlagLXB2에는 각각 mvLX[xB2, yB2], refIdxLX[xB2, yB2], predFlagLX[xB2, yB2]이 설정된다.

[0380] *방법 1의 현재 PU의 주변 PU(공간적 PU)으로부터 머지 움직임 후보를 설정하는 단계는 본 발명의 또 다른 예에 따라 아래와 같이 변형될 수 있다.

[0381] [1] 현재 PU의 주변 PU(공간적 PU)으로부터 머지 움직임 후보를 설정하는 단계에서는 입력은 아래와 같다.

[0382] (xC yC): 현재 CU(Coding Unit) 블록의 좌상단 위치;

[0383] (xP, yP): 현재 PU의 좌상단 위치;

[0384] (nCS): 현재 CU의 크기를 가르키는 변수;

[0385] (nPSW, nPSH): 현재 PU의 너비와 높이를 가르키는 변수;

[0386] singleMCLFlag: 현재 CU에 대한 공통의 머지 후보목록의 사용 여부를 나타내는 변수;

[0387] 현재 CU블록 내에서 현재 PU의 인덱스: (partIdx)

[0388] 이 단계에서는 출력은 아래와 같다. (여기서 N은 도 102의 A1, B1, B0, A0, B2으로 교체될 수 있으며, X는 '0' 혹은 '1'로 교체될 수 있다.)

[0389] availableFlagN: 주변 PU N의 움직임 정보 가용성 여부를 알려주는 변수;

[0390] refIdxLXN: 주변 PU N의 참조 인덱스;

[0391] predFlagLXN: 주변 PU N에 대한 예측 목록 이용가능 여부를 알려주는 변수;

[0392] mvLXN: 주변 PU N에 대한 움직임 벡터

[0394] [2] 현재 PU의 주변 PU(공간적 PU)이 머지 움직임 후보로 사용 가능한지를 판단하기 위한 과정은 좌측 우선 결정 방식에 따라 A1→B1→B0→A0→B2 블록의 순서로 수행된다. 여기서 주변 PU은 도 12의 A1, B1, B0, A0, B2을 말하며, N은 도 102의 A1, B1, B0, A0, B2으로 교체될 수 있으며, (xN, yN) 은 (xC-1, yC+nCS), (xC-1, yC+nCS-1), (xC+nCS, yC-1), (xC+nCS-1, yC-1), (xC-1, yC-1)으로 교체될 수 있다.

[0395] [2-1] 주변 PU의 가용성을 판단하는 과정을 수행하며, 여기서 입력은 (xC, yC), (xP, yP), nPbW, nPbH, (xN, yN) 그리고 partIdx 이며, 출력은 (xN, yN)에 위치한 PU의 가용성 정보 availableN 이다.

[0396] [2-2] 만일 아래 조건 중((1) 내지 (10))에서 하나라도 '참' 이면, availableFlagN 이 '0'으로 설정되고 mvLXN은 '0,0'으로 설정된다.

[0397] (1) (xP >> (log2_parallel_merge_level_minus2 +2))가 (xN >> (log2_parallel_merge_level_minus2 +2))와 같고 (yP >> (log2_parallel_merge_level_minus2 +2))가 (yN >> (log2_parallel_merge_level_minus2 +2))와 같다.

[0398] (2) N이 B2이고 availableFlagA0 + availableFlagA1 + availableFlagB0 + availableFlagB1이 '4'이다.

[0399] (3) [xN, yN]의 위치에 있는 PU이 사용 불가능하거나 혹은 인트라로 부호화되었다. 즉, availableN 이 '0'이다.

[0400] (4) singleMCLFlag가 '0'이고 현재 PU의 파티션 형태가 PART_2NxN 혹은 PART_2NxnU 혹은 PART_2NxnD이고 PartIdx가 '1'이고 N이 'B1'이다.

[0401] (5) singleMCLFlag가 '0'이고 현재 PU의 파티션 형태가 PART_Nx2N 혹은 PART_nLx2N 혹은 PART_nRx2N 이고 PartIdx가 '1'이고 N이 'A1'이다.

[0402] (6) 움직임 정보 동일성 판단 과정으로, N이 B1과 같고 그리고 (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0403] (7) 움직임 정보 동일성 판단 과정으로, N이 B0이고, (xB1, yB1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0404] (8) 움직임 정보 동일성 판단 과정으로, N이 A0 이고, (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0406] *(9) 움직임 정보 동일성 판단 과정으로, N이 B2 이고, (xA1, yA1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0407] (10) 움직임 정보 동일성 판단 과정으로, N이 B2 이고, (xB1, yB1)에 위치한 PU의 움직임 정보와 (xN, yN)에 위치한 움직임 정보가 동일하다.

[0408] 그렇지 않으면, availableFlagN은 '1'로 설정되며, mvLXN, refIdxLXN, predFlagLXN에는 각각 mvLX[xN, yN], refIdxLX[xN, yN], predFlagLX[xN, yN]이 설정된다.

[0410] 상술한 방법들은 모두 블록 크기 혹은 CU 깊이 등에 따라 적용 범위를 달리할 수 있다. 이렇게 적용 범위를 결정하는 변수(즉, 크기 혹은 깊이 정보)는 부호화기 및 복호화기가 미리 정해진 값을 사용하도록 설정할 수도 있고, 프로파일 또는 레벨에 따라 정해진 값을 사용하도록 할 수도 있고, 부호화기가 변수 값을 비트스트림에 기재하면 복호화기는 비트스트림으로부터 이 값을 유도하여 사용할 수도 있다.

[0411] CU 깊이에 따라 적용 범위를 달리하는 할 때는 아래 표 1에 예시한 바와 같이, 방법 (A) 주어진 깊이 이상의 깊이에만 적용하는 방식, 방법 (B) 주어진 깊이 이하에만 적용하는 방식, 방법 (C) 주어진 깊이에만 적용하는 방식이 있을 수 있다.

표 1

[0412]

적용범위를 나타내는 CU 깊이	방법A	방법B	방법C
0	X	O	X
1	X	O	X
2	O	O	O
3	O	X	X
4	O	X	X

[0413] 표 1의 “0” 은 해당 깊이에 가용성 여부를 판단하는 방법을 적용하는 경우를 나타내고, “X” 는 해당 깊이에 가용성 여부를 판단하는 방법을 적용하지 않는 것을 나타낸다.

[0414] 표 1을 참조하면, CU 깊이가 2인 경우 상술한 방법들이 모두 적용될 수 있다. 표 1은 하나의 예를 도시한 것으로, 이는 변경 가능하다.

[0416] 상술한 실시예에서, 방법들은 일련의 단계 또는 블록으로서 순서도를 기초로 설명되고 있으나, 본 발명은 단계들의 순서에 한정되는 것은 아니며, 어떤 단계는 상술한 바와 다른 단계와 다른 순서로 또는 동시에 발생할 수 있다. 또한, 당해 기술 분야에서 통상의 지식을 가진 자라면 순서도에 나타난 단계들이 배타적이지 않고, 다른 단계가 포함되거나, 순서도의 하나 또는 그 이상의 단계가 본 발명의 범위에 영향을 미치지 않고 삭제될 수 있음을 이해할 수 있을 것이다.

[0417] 상술한 실시예는 다양한 양태의 예시들을 포함한다. 다양한 양태들을 나타내기 위한 모든 가능한 조합을 기술할 수는 없지만, 해당 기술 분야의 통상의 지식을 가진 자는 다른 조합이 가능함을 인식할 수 있을 것이다. 따라서, 본 발명은 이하의 특허청구범위 내에 속하는 모든 다른 교체, 수정 및 변경을 포함한다고 할 것이다.

부호의 설명

[0418]

100 : 영상 부호화 장치 111: 움직임 예측부

112: 움직임 보상부 120 : 인트라 예측부

115 : 스위치 125 : 감산기

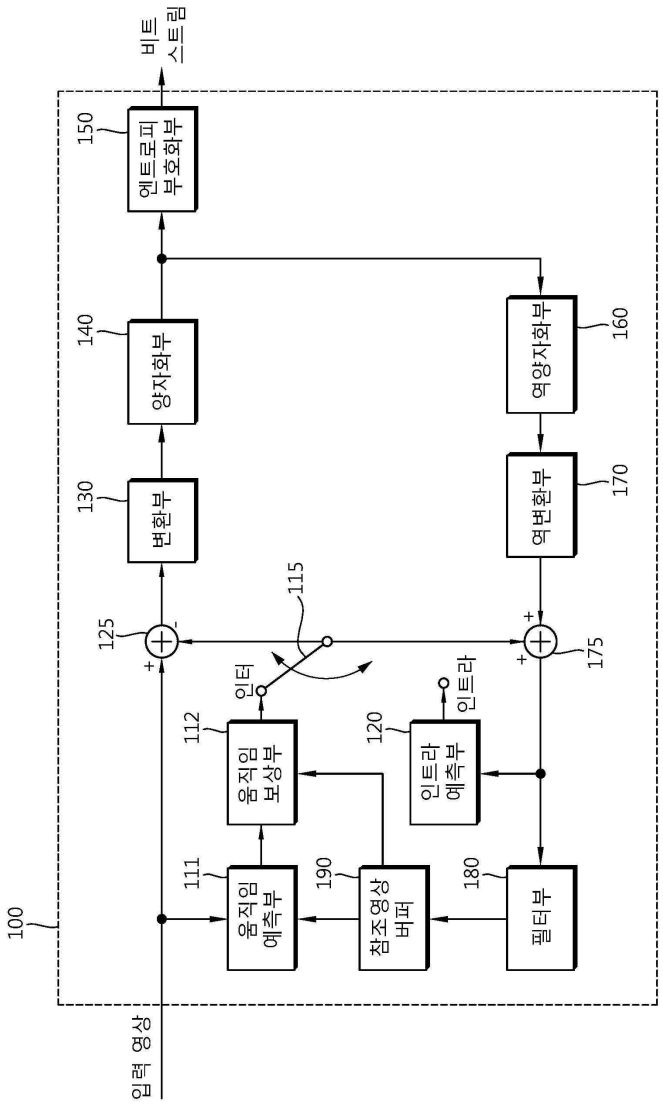
130 : 변환부 140 : 양자화부

150 : 엔트로피 부호화부 160 : 역양자화부

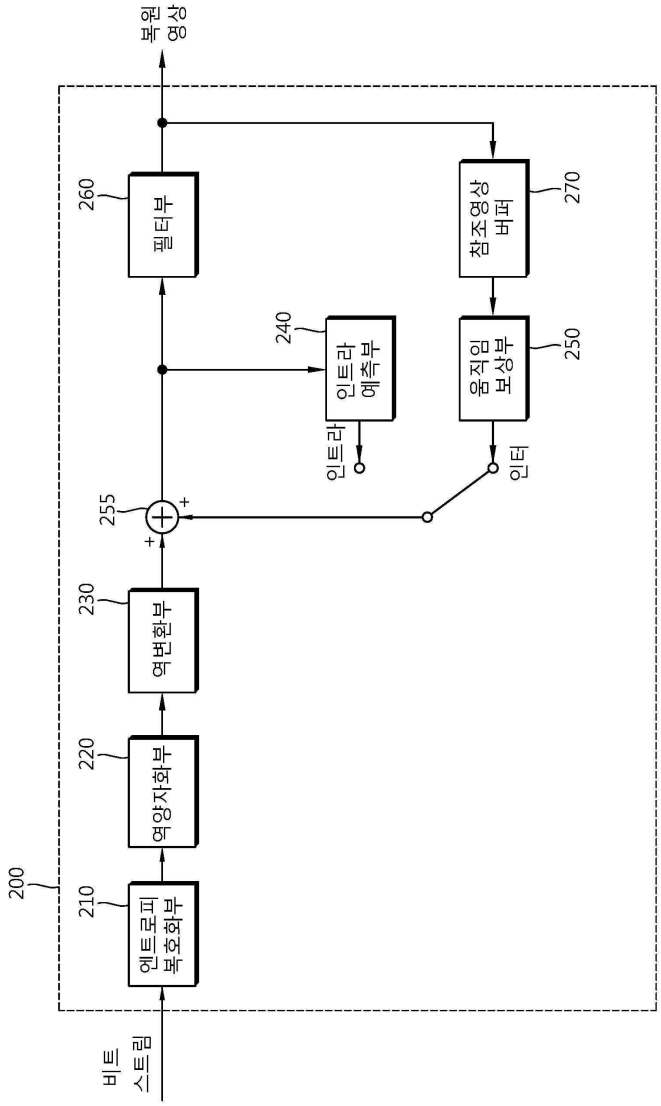
170 : 역변환부 180 : 필터부

도면

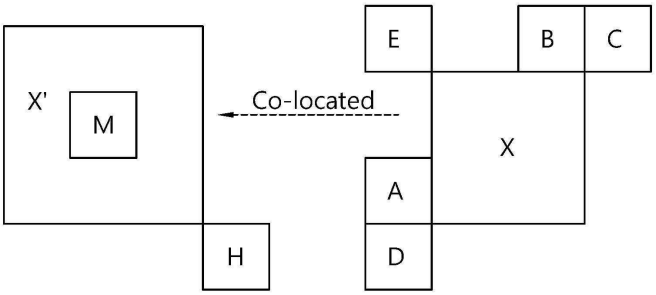
도면1


비트 스트림
엔트로피 부호화부
150
양자화부
140
변환부
130
역양자화부
160
역변환부
170
115
125
175
인터
인트라
움직임 보상부
112
인트라 예측부
120
움직임 예측부
111
참조영상 버퍼
190
필터부
180
100
입력 영상

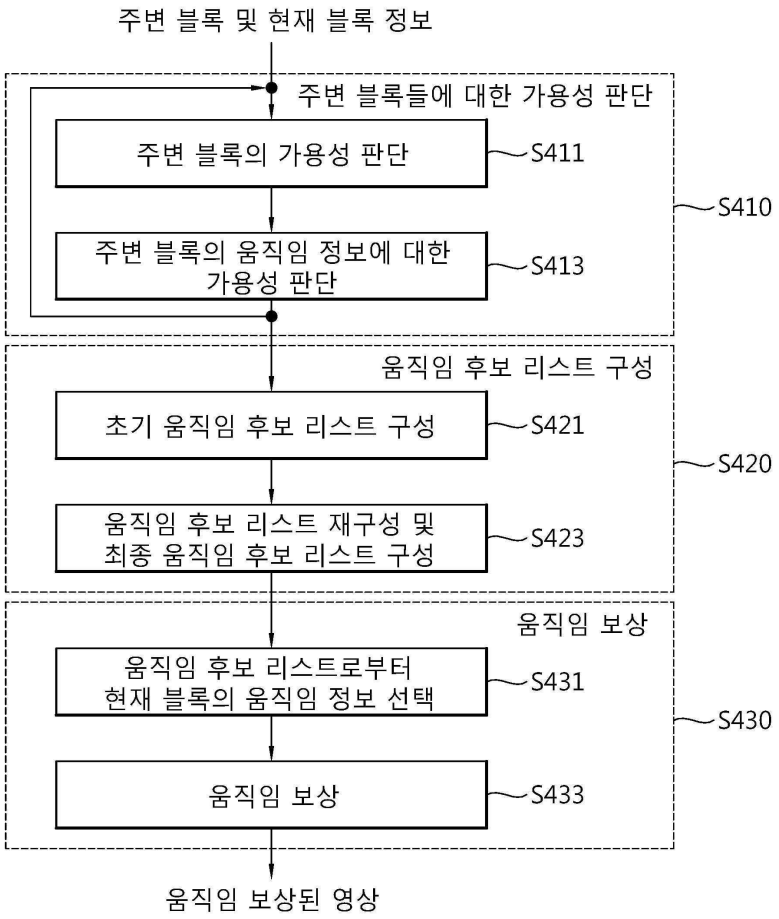
도면2


복원 영상
260
필터부
270
참조영상 버퍼
240
인트라 예측부
250
움직임 보상부
인트라
인터
255
+
+
230
역변환부
220
역양자화부
210
엔트로피 복호화부
200
비트 스트림

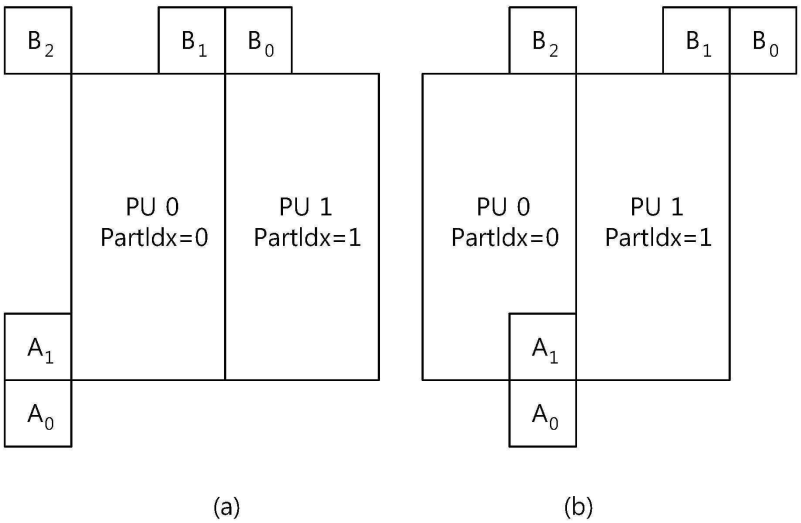
도면3


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

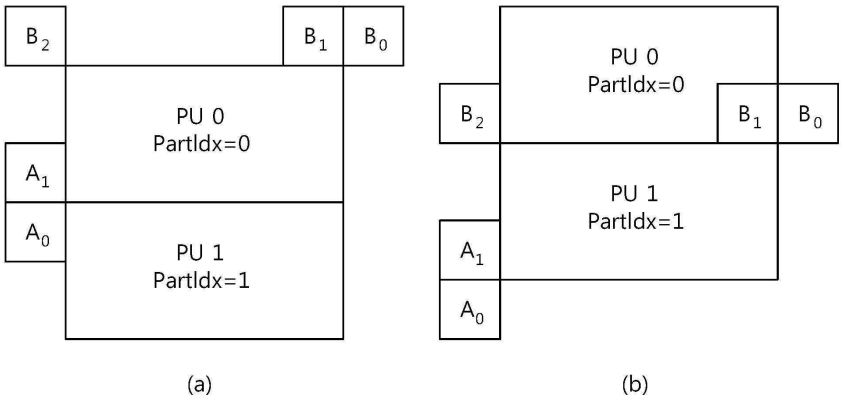
도면4


주변 블록 및 현재 블록 정보
주변 블록들에 대한 가용성 판단
주변 블록의 가용성 판단
S411
주변 블록의 움직임 정보에 대한 가용성 판단
S413
S410
움직임 후보 리스트 구성
초기 움직임 후보 리스트 구성
S421
움직임 후보 리스트 재구성 및 최종 움직임 후보 리스트 구성
S423
S420
움직임 보상
움직임 후보 리스트로부터 현재 블록의 움직임 정보 선택
S431
움직임 보상
S433
S430
움직임 보상된 영상

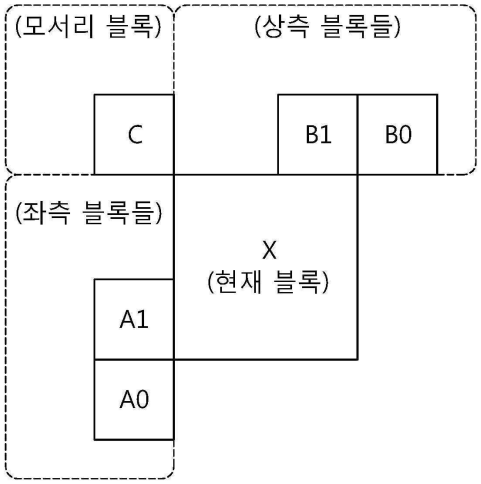
도면5


B2
B1
B0
PU 0
PartIdx=0
PU 1
PartIdx=1
A1
A0
(a)
B2
B1
B0
PU 0
PartIdx=0
PU 1
PartIdx=1
A1
A0
(b)

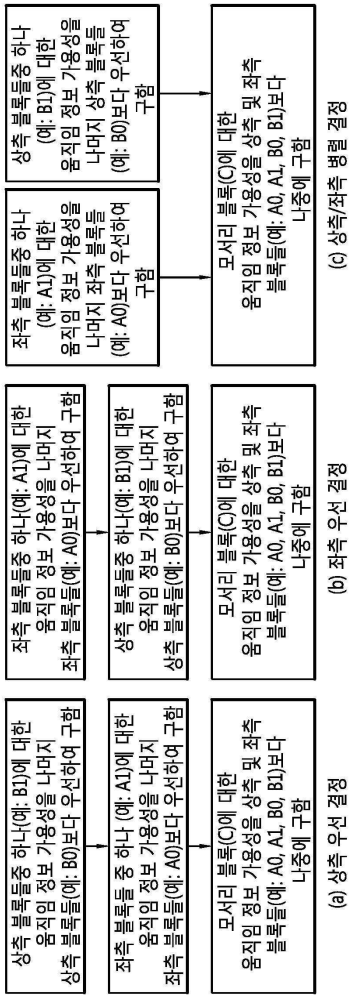
도면6


B2
B1
B0
PU 0
PartIdx=0
A1
A0
PU 1
PartIdx=1
(a)
PU 0
PartIdx=0
B2
B1
B0
PU 1
PartIdx=1
A1
A0
(b)

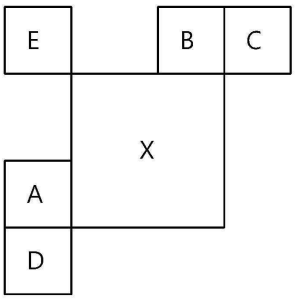
도면7


(모서리 블록)
(상측 블록들)
C
B1
B0
(좌측 블록들)
X
(현재 블록)
A1
A0

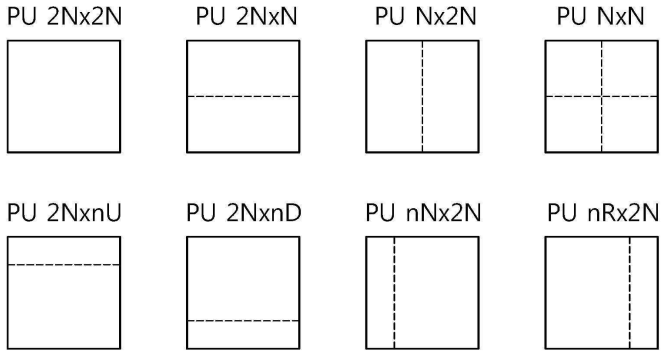
도면8


상측 블록들중 하나(예: B1)에 대한 움직임 정보 가용성을 나머지 상측 블록들(예: B0)보다 우선하여 구함
좌측 블록들중 하나(예: A1)에 대한 움직임 정보 가용성을 나머지 좌측 블록들(예: A0)보다 우선하여 구함
모서리 블록(C)에 대한 움직임 정보 가용성을 상측 및 좌측 블록들(예: A0, A1, B0, B1)보다 나중에 구함
(c) 상측/좌측 병렬 결정
좌측 블록들중 하나(예: A1)에 대한 움직임 정보 가용성을 나머지 좌측 블록들(예: A0)보다 우선하여 구함
상측 블록들중 하나(예: B1)에 대한 움직임 정보 가용성을 나머지 상측 블록들(예: B0)보다 우선하여 구함
모서리 블록(C)에 대한 움직임 정보 가용성을 상측 및 좌측 블록들(예: A0, A1, B0, B1)보다 나중에 구함
(b) 좌측 우선 결정
상측 블록들중 하나(예: B1)에 대한 움직임 정보 가용성을 나머지 상측 블록들(예: B0)보다 우선하여 구함
좌측 블록들 중 하나 (예: A1)에 대한 움직임 정보 가용성을 나머지 좌측 블록들(예: A0)보다 우선하여 구함
모서리 블록(C)에 대한 움직임 정보 가용성을 상측 및 좌측 블록들(예: A0, A1, B0, B1)보다 나중에 구함
(a) 상측 우선 결정

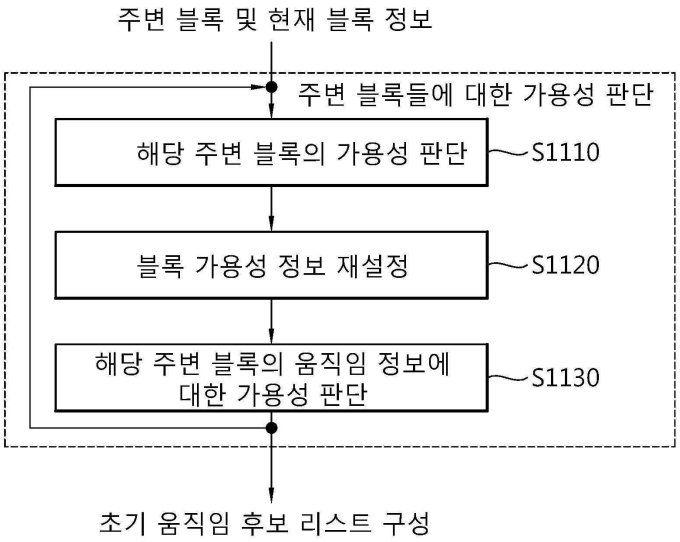
도면9


E
B
C
X
A
D

도면10



도면11



도면12

