(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0133927
(43) 공개일자 2021년11월08일

(51) 국제특허분류(Int. Cl.)
H04N 19/597 (2014.01) *H04N 19/176* (2014.01)
H04N 19/50 (2014.01)
(52) CPC특허분류
H04N 19/597 (2015.01)
H04N 19/176 (2015.01)
(21) 출원번호 10-2021-0145772(분할)
(22) 출원일자 2021년10월28일
심사청구일자 2021년10월28일
(62) 원출원 특허 10-2020-0149041
원출원일자 2020년11월10일
심사청구일자 2020년11월10일
(30) 우선권주장
1020140000527 2014년01월03일 대한민국(KR)
1020140001531 2014년01월06일 대한민국(KR)

(71) 출원인
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
박광훈
경기도 성남시 분당구 예원로 7, B동 302호(분당동, 동아빌라)
이민성
경기도 수원시 영통구 봉영로1482번길 7-3, B03호(영통동)
(뒷면에 계속)
(74) 대리인
성병기

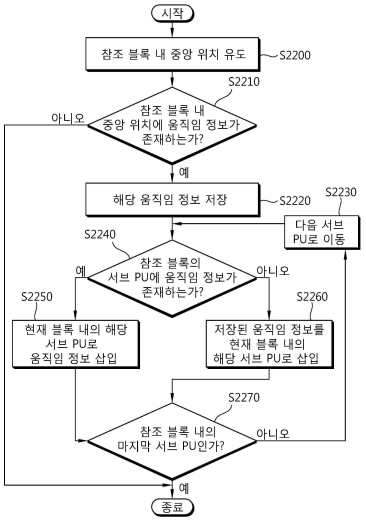
전체 청구항 수 : 총 3 항

(54) 발명의 명칭 **서브 예측 유닛 단위의 움직임 정보 유도의 방법 및 장치**

(57) 요 약

본 발명에 따르면, 현재 블록에 대한 예측 모드를 인터 예측 모드로 결정하는 단계, 참조 픽처에서 상기 현재 블록에 대응하는 참조 블록에 움직임 정보가 존재하는지를 판단하는 단계, 상기 참조 블록에 움직임 정보가 존재하는 경우에, 상기 현재 블록에 대한 움직임 정보를 상기 현재 블록 내 서브 예측 블록 단위로 유도하는 단계 및 상기 현재 블록에 대한 움직임 정보를 기반으로, 상기 현재 블록에 대한 예측 샘플을 유도하는 단계를 포함하는 3차원 영상 부호화 방법을 제공한다.

대 표 도 - 도22



(52) CPC특허분류
H04N 19/50 (2015.01)
(72) 발명자
허영수
서울특별시 송파구 올림픽로 435, 308동 1701호(신천동, 파크리오아파트)
이윤진
경기도 용인시 기흥구 흥덕2로 80, 309동 1502호(영덕동, 휴먼시아 흥덕마을3단지)

명 세 서

청구범위

청구항 1

현재 블록의 움직임 정보를 서브 블록 단위로 유도하는 단계; 및

상기 서브 블록 단위로 유도된 움직임 정보를 기반으로 상기 현재 블록에 대한 인터 예측을 수행하는 단계를 포함하고,

상기 현재 블록의 움직임 정보를 서브 블록 단위로 유도하는 단계는,

현재 서브 블록에 대응하는 참조 블록내의 대응 서브 블록에 움직임 정보가 존재하는지 여부를 판단하는 단계; 여기서, 상기 현재 서브 블록은 상기 현재 블록 내의 서브 블록임,

상기 대응 서브 블록에 움직임 정보가 존재하는 경우, 상기 대응 서브 블록의 움직임 정보로부터 상기 현재 서브 블록의 움직임 정보를 유도하는 단계; 및

상기 대응 서브 블록에 움직임 정보가 존재하지 않는 경우, 상기 참조 블록 내의 중앙 위치의 움직임 정보로부터 상기 현재 서브 블록의 움직임 정보를 유도하는 단계를 포함하고,

상기 참조 블록은 상기 현재 블록의 위치에 기초하여 결정되는 것을 특징으로 하는 영상 복호화 방법.

청구항 2

현재 블록의 움직임 정보를 서브 블록 단위로 유도하는 단계; 및

상기 서브 블록 단위로 유도된 움직임 정보를 기반으로 상기 현재 블록에 대한 인터 예측을 수행하는 단계를 포함하고,

상기 현재 블록의 움직임 정보를 서브 블록 단위로 유도하는 단계는,

현재 서브 블록에 대응하는 참조 블록내의 대응 서브 블록에 움직임 정보가 존재하는지 여부를 판단하는 단계; 여기서, 상기 현재 서브 블록은 상기 현재 블록 내의 서브 블록임,

상기 대응 서브 블록에 움직임 정보가 존재하는 경우, 상기 대응 서브 블록의 움직임 정보로부터 상기 현재 서브 블록의 움직임 정보를 유도하는 단계; 및

상기 대응 서브 블록에 움직임 정보가 존재하지 않는 경우, 상기 참조 블록 내의 중앙 위치의 움직임 정보로부터 상기 현재 서브 블록의 움직임 정보를 유도하는 단계를 포함하고,

상기 참조 블록은 상기 현재 블록의 위치에 기초하여 결정되는 것을 특징으로 하는 영상 부호화 방법.

청구항 3

부호화 방법에 의해 생성된 비트스트림을 저장하는 비 일시적 컴퓨터 판독 가능한 기록 매체에 있어서,

상기 부호화 방법은,

현재 블록의 움직임 정보를 서브 블록 단위로 유도하는 단계; 및

상기 서브 블록 단위로 유도된 움직임 정보를 기반으로 상기 현재 블록에 대한 인터 예측을 수행하는 단계를 포함하고,

상기 현재 블록의 움직임 정보를 서브 블록 단위로 유도하는 단계는,

현재 서브 블록에 대응하는 참조 블록내의 대응 서브 블록에 움직임 정보가 존재하는지 여부를 판단하는 단계;

여기서, 상기 현재 서브 블록은 상기 현재 블록 내의 서브 블록임,

상기 대응 서브 블록에 움직임 정보가 존재하는 경우, 상기 대응 서브 블록의 움직임 정보로부터 상기 현재 서브 블록의 움직임 정보를 유도하는 단계; 및

상기 대응 서브 블록에 움직임 정보가 존재하지 않는 경우, 상기 참조 블록 내의 중앙 위치의 움직임 정보로부터 상기 현재 서브 블록의 움직임 정보를 유도하는 단계를 포함하고,

상기 참조 블록은 상기 현재 블록의 위치에 기초하여 결정되는 것을 특징으로 하는 것을 특징으로 하는 기록 매체.

발명의 설명

기 술 분 야

[0001] 본 발명은 영상 부호화/복호화 장치 및 방법에 관한 발명으로, 더 구체적으로는 서브 예측 유닛 단위(Sub-Prediction Unit)에 따라 시간적 혹은 시점 간 움직임 정보를 유도하는 영상 부호화/복호화 방법 및 장치에 관한 발명이다.

배 경 기 술

[0002] 정보 통신 산업의 지속적인 발달은 HD(High Definition) 해상도를 가지는 방송 서비스의 세계적인 확산을 야기시켰으며, 이로 인해 많은 사용자들은 HD 영상에 익숙해지게 되었다.

[0003] HD화질에 익숙해진 사용자들은 더욱 더 높은 화질과 높은 해상도를 가지는 영상을 원하게 되었으며, 사용자들의 수요를 충족시키기 위해 많은 기관들이 차세대 영상 기기에 대한 개발에 박차를 가하였다. 이로 인하여, 오늘날에는 FHD(Full HD) 및 UHD(Ultra High Definition)가 지원되는 영상을 접할 수 있게 되었다.

[0004] 사용자들은 높은 화질과 높은 해상도를 가지는 영상뿐만 아니라, 입체감을 느낄 수 있는 3차원 영상을 원하게 되었다. 다양한 기관들은 사용자들의 수요를 충족시켜 주기 위하여 3차원 영상을 개발하기에 이르렀다.

[0005] 하지만, 3차원 영상은 실제 영상(texture) 정보뿐만 아니라 깊이 맵(depth map) 정보까지 포함하기 때문에, 2차원 영상보다 많은 데이터를 포함한다. . 따라서, 기존의 영상 부호화/복호화 과정에 따라 3차원 영상을 부호화/복호화 할 경우에는 충분한 부호화/복호화 효율이 나오지 않은 문제점이 있다.

발명의 내용

해결하려는 과제

[0006] 본 발명이 해결하고자 하는 과제는 부호화/복호화 대상 블록의 움직임 정보를 유도하는 장치 및 방법을 제공하는 것이다.

[0007] 본 발명이 해결하고자 하는 다른 과제는 부호화/복호화 대상 블록의 움직임 정보를 유도하는데 있어서, 데이터 의존성을 제거하는 장치 및 방법을 제공하는 것이다.

[0008] 본 발명이 해결하고자 하는 또 다른 과제는 서브 예측 유닛 단위로 부호화/복호화 대상 블록의 움직임 정보를 유도하는데 있어서, 데이터 의존성을 제거하여 영상 부호화/복호화 효율을 향상시키는 장치 및 방법을 제공하는 것이다.

[0009] 본 발명이 해결하고자 하는 또 다른 과제는 서브 예측 유닛 단위로 부호화/복호화 대상 블록의 움직임 정보를 유도할 때, 참조 블록의 움직임 정보를 이용하여 영상 부호화/복호화 효율을 향상시키는 장치 및 방법을 제공하는 것이다.

과제의 해결 수단

[0010] 본 발명의 일 실시예에 따르면, 현재 블록에 대한 예측 모드를 인터 예측 모드로 결정하는 단계, 참조 픽처에서 상기 현재 블록에 대응하는 참조 블록에 움직임 정보가 존재하는지를 판단하는 단계, 상기 참조 블록에 움직임 정보가 존재하는 경우에, 상기 현재 블록에 대한 움직임 정보를 상기 현재 블록 내 서브 예측 블록 단위로 유도

하는 단계 및 상기 현재 블록에 대한 움직임 정보를 기반으로, 상기 현재 블록에 대한 예측 샘플을 유도하는 단계를 포함하는 것을 특징으로 하는 3차원 영상 부호화 방법을 제공할 수 있다.

[0011] 이때, 상기 현재 블록과 상기 참조 블록은 예측 불록일 수 있다.

[0012] 이때, 상기 참조 블록의 움직임 정보는 상기 참조 블록의 중앙에 위치할 수 있다.

[0013] 이때, 상기 현재 블록에 대한 움직임 정보를 상기 현재 블록 내 서브 예측 블록 단위로 유도하는 단계에서는, 상기 현재 블록 내 서브 예측 블록에 대응하는 상기 참조 블록 내 서브 예측 블록에 움직임 정보가 존재하는 경우, 상기 참조 블록 내 서브 예측 블록에 존재하는 움직임 정보를 상기 현재 블록 내 서브 예측 블록의 움직임 정보로 유도할 수 있다.

[0014] 이때, 상기 현재 블록 내 서브 예측 블록에 대응하는 상기 참조 블록 내 서브 예측 블록에 움직임 정보가 존재하지 않는 경우, 상기 참조 블록의 움직임 정보를 상기 현재 블록 내 서브 예측 블록의 움직임 정보로 유도할 수 있다.

[0015] 본 발명의 다른 실시예에 따르면, 현재 블록에 대한 예측 모드를 인터 예측 모드로 결정하고, 참조 픽처에서 상기 현재 블록에 대응하는 참조 블록에 움직임 정보가 존재하는지를 판단하는 저장부 및 상기 참조 블록에 움직임 정보가 존재하는 경우에, 상기 현재 블록에 대한 움직임 정보를 상기 현재 블록 내 서브 예측 블록 단위로 유도하고, 상기 현재 블록에 대한 움직임 정보를 기반으로, 상기 현재 블록에 대한 예측 샘플을 유도하는 유도부를 포함하는 3차원 영상 부호화 장치를 제공할 수 있다.

[0016] 이때, 상기 현재 블록과 상기 참조 블록은 예측 불록일 수 있다.

[0017] 이때, 상기 참조 블록의 움직임 정보는 상기 참조 블록의 중앙에 위치할 수 있다.

[0018] 이때, 상기 유도부에서는, 상기 현재 블록 내 서브 예측 블록에 대응하는 상기 참조 블록 내 서브 예측 블록에 움직임 정보가 존재하는 경우, 상기 참조 블록 내 서브 예측 블록에 존재하는 움직임 정보를 상기 현재 블록 내 서브 예측 블록의 움직임 정보로 유도할 수 있다.

[0019] 이때, 상기 현재 블록 내 서브 예측 블록에 대응하는 상기 참조 블록 내 서브 예측 블록에 움직임 정보가 존재하지 않는 경우, 상기 참조 블록의 움직임 정보를 상기 현재 블록 내 서브 예측 블록의 움직임 정보로 유도할 수 있다.

[0020] 본 발명의 또 다른 실시예에 따르면, 현재 블록에 대한 예측 모드를 인터 예측 모드로 결정하는 단계, 참조 픽처에서 상기 현재 블록에 대응하는 참조 블록에 움직임 정보가 존재하는지를 판단하는 단계, 상기 참조 블록에 움직임 정보가 존재하는 경우에, 상기 현재 블록에 대한 움직임 정보를 상기 현재 블록 내 서브 예측 블록 단위로 유도하는 단계 및 상기 현재 블록에 대한 움직임 정보를 기반으로, 상기 현재 블록에 대한 예측 샘플을 유도하는 단계를 포함하는 3차원 영상 복호화 방법을 제공할 수 있다.

[0021] 이때, 상기 현재 블록과 상기 참조 블록은 예측 불록일 수 있다.

[0022] 이때, 상기 참조 블록의 움직임 정보는 상기 참조 블록의 중앙에 위치할 수 있다.

[0023] 이때, 상기 현재 블록에 대한 움직임 정보를 상기 현재 블록 내 서브 예측 블록 단위로 유도하는 단계에서는, 상기 현재 블록 내 서브 예측 블록에 대응하는 상기 참조 블록 내 서브 예측 블록에 움직임 정보가 존재하는 경우, 상기 참조 블록 내 서브 예측 블록에 존재하는 움직임 정보를 상기 현재 블록 내 서브 예측 블록의 움직임 정보로 유도할 수 있다.

[0024] 이때, 상기 현재 블록 내 서브 예측 블록에 대응하는 상기 참조 블록 내 서브 예측 블록에 움직임 정보가 존재하지 않는 경우, 상기 참조 블록의 움직임 정보를 상기 현재 블록 내 서브 예측 블록의 움직임 정보로 유도할 수 있다.

[0025] 본 발명의 또 다른 실시예에 따르면, 현재 블록에 대한 예측 모드를 인터 예측 모드로 결정하고, 참조 픽처에서 상기 현재 블록에 대응하는 참조 블록에 움직임 정보가 존재하는지를 판단하는 저장부 및 상기 참조 블록에 움직임 정보가 존재하는 경우에, 상기 현재 블록에 대한 움직임 정보를 상기 현재 블록 내 서브 예측 블록 단위로 유도하고, 상기 현재 블록에 대한 움직임 정보를 기반으로, 상기 현재 블록에 대한 예측 샘플을 유도하는 유도부를 포함하는 3차원 영상 복호화 장치를 제공할 수 있다.

[0026] 이때, 상기 현재 블록과 상기 참조 블록은 예측 불록일 수 있다.

[0027] 이때, 상기 참조 블록의 움직임 정보는 상기 참조 블록의 중앙에 위치할 수 있다.

[0028] 이때, 상기 유도부에서는, 상기 현재 블록 내 서브 예측 블록에 대응하는 상기 참조 블록 내 서브 예측 블록에 움직임 정보가 존재하는 경우, 상기 참조 블록 내 서브 예측 블록에 존재하는 움직임 정보를 상기 현재 블록 내 서브 예측 블록의 움직임 정보로 유도할 수 있다.

[0029] 이때, 상기 현재 블록 내 서브 예측 블록에 대응하는 상기 참조 블록 내 서브 예측 블록에 움직임 정보가 존재하지 않는 경우, 상기 참조 블록의 움직임 정보를 상기 현재 블록 내 서브 예측 블록의 움직임 정보로 유도할 수 있다.

발명의 효과

[0030] 본 발명에 따르면, 부호화/복호화 대상 블록의 움직임 정보를 유도하는 효과를 지닌다.

[0031] 본 발명에 따르면, 부호화/복호화 대상 블록의 움직임 정보를 유도하는데 있어서, 데이터 의존성을 제거하는 효과를 지닌다.

[0032] 본 발명에 따르면, 서브 예측 유닛 단위로 부호화/복호화 대상 블록의 움직임 정보를 유도하는데 있어서, 데이터 의존성을 제거하여 영상 부호화/복호화 효율을 향상시키는 효과를 지닌다.

[0033] 본 발명에 따르면, 서브 예측 유닛 단위로 부호화/복호화 대상 블록의 움직임 정보를 유도할 때, 참조 블록의 움직임 정보를 이용하여 영상 부호화/복호화 효율을 향상시키는 효과를 지닌다.

도면의 간단한 설명

[0034] 도 1은 3차원 비디오 시스템의 기본 구조를 개략적으로 도시한 것이다.

도 2는 “balloons” 영상의 실제 영상과 깊이 정보 맵 영상의 일례를 나타내는 도면이다.

도 3은 영상을 부호화 및 복호화할 때 영상의 분할 구조를 개략적으로 나타내는 도면이다.

도 4는 부호화 유닛(CU)이 포함할 수 있는 예측 유닛(PU)의 형태를 도시한 것이다.

도 5는 3차원 비디오 코덱(codec)에서 시점간 예측(inter view prediction)의 구조를 나타내는 일례다.

도 6은 3차원 비디오 인코더 및/또는 디코더에서 실제 영상(texture view)과 깊이 정보 맵(depth view)을 부호화 및/또는 복호화하는 과정을 나타낸 일례다.

도 7은 비디오 인코더의 일 실시예에 따른 구성을 나타내는 블록도이다.

도 8은 비디오 디코더의 일 실시예에 따른 구성을 나타내는 블록도이다.

도 9는 3차원 비디오 코덱에 대한 예측 구조의 일례를 나타내는 도면이다.

도 10은 현재 블록에 대한 병합 움직임(merge) 리스트(list)를 구성하는 데 사용되는 주변 블록들을 나타낸 일례다.

도 11은 인접한 시점의 움직임 정보를 이용하여, 현재 블록의 움직임 정보를 유도하는 과정의 일례를 나타내는 도면이다.

도 12는 하나의 예측 유닛(Prediction Unit, PU)이 여러 개의 서브 예측 유닛(Sub-Prediction Unit)으로 나누어지는 일례를 나타내는 도면이다.

도 13은 참조 블록을 이용하여 현재 블록의 움직임 정보를 유도하는 과정의 일례를 나타내는 도면이다.

도 14는 현재 블록의 움직임 정보를 유도하는데 사용되는 참조 블록의 일례를 도시한 것이다.

도 15a 내지 도 15e는 저장 공간에 저장된 움직임 정보를 이용하여, 움직임 정보를 유도하는 과정의 일례를 개략적으로 설명하는 도면이다.

도 16a 내지 도 16g는 저장 공간에 저장된 움직임 정보를 이용하여, 움직임 정보를 유도하는 과정의 다른 예를 개략적으로 설명하는 도면이다.

도 17은 일례에 따른, 참조 블록의 서브 예측 유닛을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보

를 유도하는 방법의 순서도다.

도 18은 참조 블록의 서브 예측 유닛을 이용하여 현재 블록의 서브 예측 유닛의 정보를 병렬적으로 유도하는 과정의 일례를 나타내는 도면이다.

도 19는 참조 블록 내에서 사용 가능한 서브 예측 유닛이 우측 최하단에 위치한 경우, 사용 가능한 서브 예측 유닛을 탐색하는 과정의 일례를 나타내는 도면이다.

도 20은 서브 예측 유닛 단위로 움직임 정보를 유도할 때 소요되는 시간을 개략적으로 도시한 것이다.

도 21은 본 발명이 적용되는 인터 예측부의 구조를 나타내는 블록도이다.

도 22는 본 발명의 일 실시예에 따른, 참조 블록의 임의의 위치에 존재하는 서브 예측 유닛을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 방법의 순서도이다.

도 23은 본 발명의 다른 실시예에 따른, 참조 블록의 임의의 위치에 존재하는 서브 예측 유닛을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 방법의 순서도이다.

도 24는 임의 위치의 움직임 정보를 이용하여, 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 과정의 일례를 나타내는 도면이다.

도 25는 본 발명의 또 다른 실시예에 따른, 임의의 움직임 정보 값을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 방법의 순서도이다.

도 26은 임의의 움직임 정보 값을 이용하여, 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 과정의 일례를 나타내는 도면이다.

도 27은 본 발명을 적용하여 움직임 정보를 유도할 때 소요되는 시간을 개략적으로 도시한 것이다.

발명을 실시하기 위한 구체적인 내용

[0035] 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.

[0036] 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.

[0037] 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.

[0038] 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.

[0040] *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 구성부 만을 포함하여 구현될 수 있고, 단지 성능 향상을 위해 사용되는 선택적 구성 요소를 제외한 필수 구성 요소만을 포함한 구조도 본 발명의 권리범위에 포함된다.

[0042] 3차원 비디오는 3차원 입체 디스플레이 장치를 통해 사용자에게 실 세계에서 보고 느끼는 것과 같은 입체감을 제공한다. 이와 관련하여, ISO/IEC의 MPEG(Moving Picture Experts Group)와 ITU-T의 VCEG(Video Coding Experts Group)의 공동 표준화 그룹인 JCT-3V(The Joint Collaborative Team on 3D Video Coding Extension Development)에서는 3차원 비디오 표준화가 진행 중이다.

[0044] 도 1은 3차원 비디오 시스템의 기본 구조를 개략적으로 도시한 것이다.

[0045] 도 1을 참조하면, 3차원 비디오(3D video; 3DV) 시스템은 송신측(sender) 및 수신측(receiver)을 포함할 수 있다. 이때, 도 1의 3차원 비디오 시스템은 3차원 비디오 표준에서 고려하고 있는 기본적인 3차원 비디오 시스템일 수 있으며, 3차원 비디오 표준은 실제 영상과 실제 영상에 대응되는 깊이 정보 맵을 이용하여, 스테레오스코픽 영상뿐만 아니라 오토스테레오스코픽(autostereoscopic) 영상의 재생을 지원할 수 있는 진보된 데이터 형식과 그에 관련된 기술에 대한 표준을 포함할 수 있다.

[0046] 송신측은 다시점(multi-view)의 영상 콘텐츠를 생성할 수 있다. 구체적으로, 송신측에서는 스테레오 카메라 및 다시점 카메라를 이용하여 비디오 정보를 생성하고, 깊이 정보 카메라를 이용하여 깊이 정보 맵(depth map, 또는 depth view)을 생성할 수 있다. 또한, 송신측에서는 변환기를 이용하여 2차원 영상을 3차원 영상으로 변환할 수도 있다. 송신측에서는 생성된 비디오 정보와 깊이 정보 맵 등을 이용하여 N(N≥2) 시점(view)(즉, 다시점)의 영상 콘텐츠를 생성할 수 있다. 이때, N 시점의 영상 콘텐츠에는 N 시점의 비디오 정보와 그것의 깊이 정보 맵(Depth-map) 정보 그리고 카메라 관련 부가 정보 등이 포함될 수 있다. N 시점의 영상 콘텐츠는 3차원 비디오 인코더에서 다시점 비디오 부호화 방법을 사용하여 압축될 수 있으며, 압축된 영상 콘텐츠(비트스트림)는 네트워크를 통해 수신측의 단말로 전송될 수 있다.

[0047] 수신측(receiver)은 송신측으로부터 수신된 영상 콘텐츠를 복호화하여 다시점(multi-view)의 영상을 제공할 수 있다. 구체적으로, 수신측에서는 비디오 디코더(예를 들어, 3차원 비디오 디코더, 스테레오 비디오 디코더, 2차원 비디오 디코더 등)에서 다시점 비디오 복호화 방법을 사용하여 전송 받은 비트스트림을 복호화함으로써 N 시점의 영상을 복원할 수 있다. 이때, 복원된 N 시점의 영상과 깊이 정보 맵 기반 랜더링(DIBR; Depth-Image-Based Rendering) 과정을 이용하여 N 시점 이상의 가상 시점 영상들이 생성될 수도 있다. 생성된 N 시점 이상의 가상 시점 영상들은 다양한 입체 디스플레이 장치(예를 들어, N-시점 디스플레이, 스테레오 디스플레이, 2차원 디스플레이 등)에 맞게 재생되어 사용자에게 입체감이 있는 영상을 제공하게 된다.

[0049] 도 2는 “balloons” 영상의 실제 영상과 깊이 정보 맵 영상의 일례를 나타내는 도면이다.

[0050] 도 2의 (a)는 국제 표준화 기구인 MPEG의 3차원 비디오 부호화 표준에서 사용 중인 “balloons” 영상을 나타낸 것이다. 도 2의 (b)는 도 2의 (a)에 도시된 “balloons” 영상에 대한 깊이 정보 맵 영상을 나타낸 것이다. 도 2의 (b)에 도시된 깊이 정보 맵 영상은 화면에 보이는 깊이 정보를 화소당 8비트로 표현한 것이다.

[0051] 깊이 정보 맵(depth map)은 가상 시점 영상을 생성하는데 사용되며, 깊이 정보 맵은 실세계에서 카메라와 실제 사물(object) 간의 거리(실사 영상과 동일한 해상도로 각 화소에 해당하는 깊이 정보)를 일정한 비트수로 표현한 것이다. 이때, 깊이 정보 맵은 깊이 정보 맵 카메라를 이용하거나, 실제 일반 영상(Texture)을 이용하여 획득할 수 있다.

[0052] 깊이 정보 맵 카메라를 이용하여 획득한 깊이 정보 맵은 주로 정지된 객체나 장면에서 신뢰성 높은 깊이 정보를 제공하나, 일정한 거리 이내에서만 깊이 정보 맵 카메라가 작동한다는 문제점이 있다. 이때, 깊이 정보 맵 카메라는 레이저(laser)나 구조화된 광선(structured light) 기법 또는 TFL(Time-of-Flight of Light)에 기초한 깊이 측정기법을 사용할 수 있다.

[0053] 깊이 정보 맵은 실제 일반 영상(Texture)과 변이 벡터(Disparity Vector)를 이용하여 생성될 수도 있다. 변이 벡터는 두 일반 영상 간의 시점 차이를 나타내는 정보를 의미한다. 변이 벡터는 현재 시점에서의 임의의 한 화소를 다른 시점에서의 화소들과 비교하여 가장 유사한 화소를 찾았을 때, 현재 시점에서의 임의의 한 화소와 다른 시점에서의 화소(현재 시점에서의 임의의 한 화소와 가장 유사한 화소) 간의 거리를 통해 획득할 수 있다.

[0054] 실제 영상과 그것의 깊이 정보 맵은 하나의 카메라뿐만 아니라 여러 개의 카메라에서 획득된 영상일 수 있다. 여러 개의 카메라에서 획득된 영상은 독립적으로 부호화될 수 있고, 일반적인 2차원 비디오 부호화/복호화 코덱을 사용하여 부호화/복호화될 수도 있다. 또한, 여러 개의 카메라에서 획득된 영상은 시점 간의 상관성이 존재

하므로, 여러 개의 카메라에서 획득된 영상은 부호화 효율을 높이기 위하여 서로 다른 시점들 간의 예측을 사용하여 부호화될 수 있다.

[0056] 도 3은 영상을 부호화 및 복호화할 때 영상의 분할 구조를 개략적으로 나타내는 도면이다.

[0057] 영상을 효율적으로 분할하기 위해 부호화 유닛(Coding Unit: CU)별로 부호화 및 복호화를 수행할 수 있다. 유닛은 구문 요소(syntax element)와 영상 샘플들이 포함된 블록을 합쳐서 일컫는 말이다. 유닛이 분할된다는 것은 유닛에 해당하는 블록을 분할한다는 것을 의미할 수 있다.

[0058] 도 3을 참조하면, 영상(300)을 최대 부호화 유닛(Largest Coding Unit; LCU)(이하, LCU라 함) 단위로 순차적으로 분할한 후, LCU 마다 분할 구조를 결정한다. 본 명세서에서, LCU는 코딩 트리 유닛(Coding Tree Unit, CTU)과 동일한 의미로 사용될 수 있다. 분할 구조는 LCU(310) 내에서 영상을 효율적으로 부호화하기 위한 부호화 유닛(이하, CU라 함)의 분포를 의미하며, 이러한 분포는 하나의 CU를 그 가로 크기 및 세로 크기의 절반으로 감소된 4개의 CU로 분할할지 여부에 따라 결정될 수 있다. 분할된 CU는 동일한 방식으로 분할된 CU에 대해서 그 가로 크기 및 세로 크기가 절반씩 감소된 4개의 CU로 재귀적으로 분할될 수 있다.

[0059] 이때, CU의 분할은 미리 정의된 깊이까지 재귀적으로 분할될 수 있다. 깊이 정보는 CU의 크기를 나타내는 정보로서, 각 CU마다 저장될 수 있. 예컨대, LCU의 깊이는 0이고, SCU(Smallest Coding Unit)의 깊이는 미리 정의된 최대 깊이일 수 있다. 여기서, LCU는 상술한 바와 같이 최대 크기를 가지는 부호화 유닛이며, SCU(Smallest Coding Unit)는 최소 크기를 가지는 부호화 유닛이다.

[0060] LCU(310)로부터 가로 및 세로 크기의 절반으로 분할을 수행할 때마다 CU의 깊이는 1씩 증가한다. 예를 들어, 특정 깊이 L에서 CU의 크기가 2Nx2N라면, 분할이 수행되지 않는 경우에 CU의 크기는 여전히 2Nx2N이며, 분할이 수행되는 경우에 CU의 크기는 NxN이 된다. 이때, NxN 크기 CU의 깊이는 깊이 L+1이 된다. 즉, CU의 크기에 대응하는 N의 크기는 깊이가 1씩 증가할 때마다 절반으로 감소한다.

[0061] 도 3을 참조하면, 최소 깊이가 0인 LCU의 크기는 64x64 화소들이고, 최대 깊이가 3인 SCU의 크기는 8x8 화소들일 수 있다. 이때, 64x64 화소들의 CU(LCU)의 깊이는 0, 32x32 화소들의 CU의 깊이는 1, 16x16 화소들의 CU의 깊이는 2, 8x8 화소들의 CU(SCU)의 깊이는 3으로 표현될 수 있다.

[0062] 또한, 특정 CU를 분할할지에 대한 정보는 CU마다 1비트의 분할 정보를 통해 표현될 수 있다. 이 분할 정보는 SCU를 제외한 모든 CU에 포함될 수 있으며, 예컨대 CU를 분할하지 않을 경우에는 분할 정보에 0을 저장할 수 있고, CU를 분할할 경우에는 분할 정보에 1을 저장할 수 있다.

[0064] 도 4는 부호화 유닛(CU)이 포함할 수 있는 예측 유닛(PU)의 형태를 도시한 것이다.

[0065] LCU로부터 분할된 CU 중 더 이상 분할(split)되지 않는 CU는 하나 이상의 예측 유닛으로 분할 또는 파티셔닝(partitioning)될 수 있다.

[0066] 예측 유닛(이하, PU라 함)은 예측을 수행하는 기본 단위로서, 스킵(skip) 모드, 인터(inter) 모드, 인트라(intra) 모드 중 어느 하나로 부호화 및 복호화되며, 각 모드에 따라서 다양한 형태로 파티션될 수 있다.

[0067] 도 4를 참조하면, 스킵 모드의 경우, CU의 파티션 없이, CU와 동일한 크기를 갖는 2Nx2N 모드(410)를 지원할 수 있다.

[0068] 인터 모드의 경우, CU에 대하여 8가지의 파티션된 형태, 예컨대 2Nx2N 모드(410), 2NxN 모드(415), Nx2N 모드(420), NxN 모드(425), 2NxnU 모드(430), 2NxnD 모드(435), nLx2N 모드(440), nRx2N 모드(445)를 지원할 수 있다.

[0069] 인트라 모드의 경우, CU에 대하여 2Nx2N 모드(410), NxN 모드(425)를 지원할 수 있다.

[0071] 도 5는 3차원 비디오 코덱(codec)에서 시점 간 예측(inter view prediction)의 구조를 나타내는 일례다.

[0072] 시점 1(View 1)과 시점 2(View 2)는 시점 0(View 0)을 참조 영상으로 사용하여 시점 간 예측을 수행할 수 있으며, 부호화 순서는 시점 1(View 1)과 시점 2(View 2)보다 시점 0(View 0)이 먼저 부호화되어야 한다.

[0073] 이때, 시점 0(View 0)은 다른 시점과 상관없이 독립적으로 부호화될 수 있으므로 독립적인 시점(Independent View)이라고 한다. 반면, 시점 1(View 1)과 시점 2(View 2)는 시점 0(View 0)을 참조 영상으로 사용하여 부호화되므로 의존적인 시점(Dependent View)라고 한다. 독립적인 시점 영상은 일반적인 2차원 비디오 코덱을 사용하여 부호화 될 수 있다. 반면, 의존적인 시점 영상은 시점 간 예측을 수행하여야 하므로, 시점 간 예측 과정이 포함된 3차원 비디오 코덱을 사용하여 부호화될 수 있다.

[0074] 또한, 시점 1(View 1)과 시점 2(View 2)의 부호화 효율을 증가시키기 위하여 시점 1(View 1)과 시점 2(View 2)는 깊이 정보 맵을 이용하여 부호화될 수 있다. 예를 들어, 실제 영상과 그것의 깊이 정보 맵을 부호화할 때, 실제 영상과 깊이 정보 맵은 서로 독립적으로 부호화 및/또는 복호화될 수 있다. 또는 실제 영상과 깊이 정보 맵을 부호화할 때, 실제 영상과 깊이 정보 맵은 도 6과 같이 서로 의존적으로 부호화 및/또는 복호화될 수 있다.

[0076] 도 6은 3차원 비디오 인코더 및/또는 디코더에서 실제 영상(texture view)과 깊이 정보 맵(depth view)을 부호화 및/또는 복호화하는 과정을 나타낸 일례다.

[0077] 도 6을 참조하면, 3차원 비디오 인코더는 실제 영상(texture view)을 부호화하는 실제 영상 인코더(texture encoder)와 깊이 정보 맵(depth view)을 부호화하는 깊이 정보 맵 인코더(depth encoder)를 포함할 수 있다.

[0078] 이때, 실제 영상 인코더는 깊이 정보 맵 인코더에 의해 부호화된 깊이 정보 맵을 이용하여 실제 영상을 부호화할 수 있다. 반대로, 깊이 정보 맵 인코더는 실제 영상 인코더에 의해 부호화된 실제 영상을 이용하여 깊이 정보 맵을 부호화할 수 있다.

[0079] 3차원 비디오 디코더는 실제 영상을 복호화하는 실제 영상 디코더(texture decoder)와 깊이 정보 맵을 복호화하는 깊이 정보 맵 디코더(depth decoder)를 포함할 수 있다.

[0080] 이때, 실제 영상 디코더는 깊이 정보 맵 디코더에 의해 복호화된 깊이 정보 맵을 이용하여 실제 영상을 복호화할 수 있다. 반대로, 깊이 정보 맵 디코더는 실제 영상 디코더에 의해 복호화된 실제 영상을 이용하여 깊이 정보 맵을 복호화할 수 있다.

[0082] 도 7은 비디오 인코더의 일 실시예에 따른 구성을 나타내는 블록도이다.

[0083] 도 7은 멀티 뷰 구조에 대해 적용 가능한 비디오 인코더의 일 실시예를 나타내며, 멀티 뷰 구조에 대한 비디오 인코더는 단일 뷰 구조에 대한 비디오 인코더를 확장(extension)하여 구현될 수 있다. 이때, 도 7의 비디오 인코더는 도 6의 실제 영상 인코더 및/또는 깊이 정보 맵 인코더에서 사용될 수 있으며, 인코더는 부호화 장치를 의미할 수 있다.

[0084] 도 7을 참조하면, 비디오 인코더(700)는 인터 예측부(710), 인트라 예측부(720), 스위치(715), 감산기(725), 변환부(730), 양자화부(740), 엔트로피 부호화부(750), 역양자화부(760), 역변환부(770), 가산기(775), 필터부(780) 및 참조 픽처 버퍼(790)를 포함한다.

[0085] 비디오 인코더(700)는 입력 영상에 대해 인트라(intra) 모드 또는 인터(inter) 모드로 부호화를 수행하고 비트스트림(bitstream)을 출력할 수 있다.

[0086] 인트라 예측(intra prediction)은 화면 내 예측(intra picture prediction)을 의미하며, 인터 예측(inter prediction)은 화면 간 (inter picture) 예측 또는 뷰 간 (inter-view) 예측을 의미한다. 인트라 모드인 경우 스위치(715)가 인트라 모드로 전환되고, 인터 모드인 경우 스위치(715)가 인터 모드로 전환된다.

[0087] 비디오 인코더(700)는 입력 픽처의 블록(현재 블록)에 대한 예측 블록을 생성한 후, 현재 블록과 예측 블록의 차분을 부호화할 수 있다.

[0088] 인트라 모드인 경우, 인트라 예측부(720)는 현재 블록 주변의 이미 부호화된 블록의 화소값을 참조 화소로 이용할 수 있다. 인트라 예측부(720)는 참조 화소를 이용하여 현재 블록에 대한 예측 샘플들을 생성할 수 있다.

[0089] 인터 모드인 경우, 인터 예측부(710)는, 참조 픽처 버퍼(790)에 저장되어 있는 참조 픽처에서 입력 블록(현재 블록)에 대응하는 참조 블록을 특정하는 움직임 벡터를 구할 수 있다. 인터 예측부(710)는 움직임 벡터와 참조 픽처 버퍼(790)에 저장되어 있는 참조 픽처를 이용하여 움직임 보상을 수행함으로써 현재 블록에 대한 예측 블

록을 생성할 수 있다.

[0091] *멀티 뷰 구조에서, 인터 모드에서 적용되는 인터 예측은 인터 뷰 예측을 포함할 수 있다. 인터 예측부(710)는 참조 뷰의 픽처를 샘플링하여 인터 뷰 참조 픽처를 구성할 수도 있다. 인터 예측부(710)는 인터 뷰 참조 픽처를 포함하는 참조 픽처 리스트를 이용하여 인터 뷰 예측을 수행할 수 있다. 뷰 간의 참조 관계는 뷰 간의 의존성을 특정하는 정보를 통해 시그널링될 수 있다.

[0092] 한편, 현재 뷰의 픽처와 참조 뷰의 픽처가 동일 사이즈인 경우에 상기 참조 뷰 픽처에 적용되는 샘플링은 참조 뷰 픽처로부터의 샘플 복사 또는 보간에 의한 참조 샘플의 생성을 의미할 수 있다. 현재 뷰 픽처와 참조 뷰 픽처의 해상도가 상이한 경우에 상기 참조 뷰 픽처에 적용되는 샘플링은 업샘플링 또는 다운샘플링을 의미할 수 있다. 예컨대, 뷰 간 해상도가 다른 경우에는 참조 뷰의 복원된 픽처를 업샘플링하여 인터 뷰 참조 픽처가 구성될 수도 있다.

[0093] 어떤 뷰의 픽처를 이용하여 인터 뷰 참조 픽처를 구성할 것인지는 부호화 비용(coding cost) 등을 고려하여 결정될 수 있다. 인코더는 인터 뷰 참조 픽처로 사용될 픽처가 속하는 뷰(view)를 특정하는 정보를 복호화 장치로 전송할 수 있다.

[0094] 또한, 인터 뷰 예측에 있어서 참조되는 뷰, 즉 참조 뷰 내에서 현재 블록의 예측에 이용되는 픽처는 현재 픽처(현재 뷰 내 예측 대상 픽처)와 동일 AU(Access Unit)의 픽처일 수 있다.

[0095] 감산기(725)는 현재 블록과 예측 블록의 차분에 의해 레지듀얼 블록(residual block, 레지듀얼 신호)을 생성할 수 있다.

[0096] 변환부(730)는 레지듀얼 블록에 대해 변환(transform)을 수행하여 변환 계수(transform coefficient)를 출력할 수 있다. 변환 생략(transform skip) 모드가 적용되는 경우에, 변환부(730)는 레지듀얼 블록에 대한 변환을 생략할 수도 있다.

[0097] 양자화부(740)는 변환 계수를 양자화 파라미터에 따라 양자화하여 양자화된 계수(quantized coefficient)를 출력할 수 있다.

[0098] 엔트로피 부호화부(750)는, 양자화부(740)에서 산출된 값들 또는 부호화 과정에서 산출된 부호화 파라미터 값 등을 확률 분포에 따라 엔트로피 부호화하여 비트스트림을 출력할 수 있다. 엔트로피 부호화부(750)는 비디오의 화소 정보 외에 비디오 디코딩을 위한 정보(예컨대, 신택스 엘리먼트(syntax element) 등)을 엔트로피 부호화할 수도 있다.

[0099] 부호화 파라미터는 부호화 및 복호화에 필요한 정보로서, 신택스 엘리먼트와 같이 인코더에서 부호화되어 복호화 장치로 전달되는 정보뿐만 아니라, 부호화 혹은 복호화 과정에서 유추될 수 있는 정보를 포함할 수도 있다.

[0100] 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. 블록 단위에서, 레지듀얼 신호는 레지듀얼 블록이라 할 수 있다.

[0101] 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.

[0102] 엔트로피 부호화를 위해 지수 골룸(exponential Golomb), CAVLC(Context-Adaptive Variable Length Coding) 또는 CABAC(Context-Adaptive Binary Arithmetic Coding)과 같은 부호화 방법이 사용될 수 있다. 예를 들어, 엔트로피 부호화부(750)는 가변 길이 부호화(VLC: Variable Lenghth Coding/Code) 테이블을 이용하여 엔트로피 부호화를 수행할 수 있다. 또한 엔트로피 부호화부(750)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.

[0103] 양자화된 계수는 역양자화부(760)에서 역양자화되고 역변환부(770)에서 역변환될 수 있다. 역양자화, 역변환된 계수는 가산기(775)를 통해 예측 블록과 더해지고 복원 블록이 생성될 수 있다.

[0104] 복원 블록은 필터부(780)를 거치고, 필터부(780)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. 필터부(780)를 거친 복원 블록은 참조 영상 버퍼(790)에 저장될 수 있다.

[0106] 도 8은 비디오 디코더의 일 실시예에 따른 구성을 나타내는 블록도이다.

[0107] 도 8은 멀티 뷰 구조에 대해 적용 가능한 비디오 디코더의 일 실시예를 나타내며, 멀티 뷰 구조에 대한 비디오 디코더는 단일 뷰 구조에 대한 비디오 디코더를 확장(extension)하여 구현될 수 있다.

[0108] 이때, 도 8의 비디오 디코더는 도 6의 실제 영상 디코더 및/또는 깊이 정보 맵 디코더에서 사용될 수 있다. 설명의 편의를 위해, 본 명세서에서는 '복호화'와 '디코딩'가 혼용되거나 '복호화 장치'와 '디코더'가 혼용될 수도 있다.

[0109] 도 8을 참조하면, 상기 비디오 디코더(800)는 엔트로피 복호화부(810), 역양자화부(820), 역변환부(830), 인트라 예측부(840), 인터 예측부(850), 필터부(860) 및 참조 픽처 버퍼(870)를 포함한다.

[0110] 비디오 디코더(800)는 인코더에서 출력된 비트스트림을 입력 받아 인트라 모드 또는 인터 모드로 복호화를 수행하고 재구성된 영상, 즉 복원 영상을 출력할 수 있다.

[0111] 인트라 모드인 경우 스위치가 인트라 예측을 위해 전환되고, 인터 모드인 경우 스위치가 인터 예측을 위해 전환될 수 있다.

[0112] 비디오 디코더(800)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.

[0113] 엔트로피 복호화부(810)는, 입력된 비트스트림을 확률 분포에 따라 엔트로피 복호화하여, 양자화된 계수(quantized coefficient)와 신택스 엘리먼트 등의 정보를 출력할 수 있다.

[0114] 양자화된 계수는 역양자화부(820)에서 역양자화되고 역변환부(830)에서 역변환 된다. 양자화된 계수를 역양자화/역변환하여, 복원된 레지듀얼 블록(residual block)이 생성될 수 있다.

[0115] 인트라 모드인 경우, 인트라 예측부(840)는 현재 블록 주변의 이미 부호화된 블록의 화소값을 이용하여 현재 블록에 대한 예측 블록을 생성할 수 있다.

[0116] 인터 모드인 경우, 인터 예측부(850)는 움직임 벡터 및 참조 픽처 버퍼(870)에 저장되어 있는 참조 픽처를 이용하여 움직임 보상을 수행함으로써 현재 블록에 대한 예측 블록을 생성할 수 있다.

[0117] 멀티 뷰 구조의 경우, 인터 모드에서 적용되는 인터 예측은 인터 뷰 예측을 포함할 수 있다. 인터 예측부(850)는 참조 뷰의 픽처를 샘플링하여 인터 뷰 참조 픽처를 구성할 수 있다. 인터 예측부(850)는 인터 뷰 참조 픽처를 포함하는 참조 픽처 리스트를 이용하여 인터 뷰 예측을 수행할 수 있다. 뷰 간의 참조 관계는 뷰 간의 의존성을 특정하는 정보를 통해 시그널링될 수 있다.

[0118] 한편, 현재 뷰 픽처(현재 픽처)와 참조 뷰 픽처가 동일 사이즈인 경우에 상기 참조 뷰 픽처에 적용되는 샘플링은 참조 뷰 픽처로부터의 샘플 복사 또는 보간에 의한 참조 샘플의 생성을 의미할 수 있다. 현재 뷰 픽처와 참조 뷰 픽처의 해상도가 상이한 경우에 상기 참조 뷰 픽처에 적용되는 샘플링은 업샘플링 또는 다운샘플링을 의미할 수 있다.

[0119] 예컨대, 뷰 간 해상도가 다른 경우에 뷰 간에 인터 뷰 예측이 적용된다면, 참조 뷰의 복원된 픽처를 업샘플링하여 인터 뷰 참조 픽처가 구성될 수 있다.

[0120] 이때, 인터 뷰 참조 픽처로 사용될 픽처가 속하는 뷰를 특정하는 정보는 인코더로부터 디코더로 전송될 수 있다.

[0121] 또한, 인터 뷰 예측에 있어서 참조되는 뷰, 즉 참조 뷰 내에서 현재 블록의 예측에 이용되는 픽처는 현재 픽처(현재 뷰 내 예측 대상 픽처)와 동일 AU(Access Unit)의 픽처일 수 있다.

[0122] 복원된 레지듀얼 블록과 예측 블록은 가산기(855)에서 더해져서, 복원 블록이 생성된다. 다시 말하면, 레지듀얼 샘플과 예측 샘플이 더해져서 복원된 샘플 또는 복원된 픽처가 생성된다.

[0123] 복원된 픽처는 필터부(860)에서 필터링 된다. 필터부(860)는 디블록킹 필터, SAO, ALF 중 적어도 하나 이상을 복원 블록 또는 복원 픽처에 적용할 수 있다. 필터부(860)는 재구성된(modified) 혹은 필터링된(filtered) 복원

픽처(reconstructed picture)를 출력한다. 복원 영상은 참조 픽처 버퍼(870)에 저장되어 인터 예측에 사용될 수 있다.

[0124] 도 7과 도 8에서는 각각의 모듈들이 서로 다른 기능을 수행하는 것으로 설명하였으나, 본 발명은 이에 한정되지 않으며, 하나의 모듈에서 둘 이상의 기능을 수행할 있다. 예컨대, 도 7과 도 8에서 인트라 예측부와 인터 예측부의 동작은 하나의 모듈(예측부)에서 수행될 수도 있다.

[0125] 한편, 도 7 및 도 8에서는 하나의 인코더/디코더가 멀티 뷰에 대한 부호화/복호화를 모두 처리하는 것으로 설명하였으나, 이는 설명의 편의를 위한 것으로서, 인코더/디코더는 뷰별로 구성될 수도 있다.

[0126] 이 경우, 현재 뷰의 인코더/디코더는 다른 뷰의 정보를 이용하여 현재 뷰의 부호화/복호화를 수행할 수 있다. 예컨대, 현재 뷰의 예측부(인터 예측부)는 다른 뷰의 픽셀 정보 또는 복원된 픽처 정보를 이용하여 현재 블록에 대한 인트라 예측 또는 인터 예측을 수행할 수도 있다.

[0127] 여기서는, 뷰 간의 예측만을 예로서 설명하였으나, 인코더/디코더는 뷰별로 구성되든, 하나의 장치가 멀티 뷰를 처리하든 상관없이, 다른 뷰의 정보를 이용하여 현재 레이어에 대한 부호화/복호화를 수행할 수 있다.

[0128] 본 발명에서 뷰에 관한 설명은 스케일러빌리티를 지원하는 레이어(layer)에 대해서도 동일하게 적용될 수 있다. 예컨대, 본 발명에서 뷰(view)는 레이어(layer)일 수도 있다.

[0130] 도 9는 3차원 비디오 코덱에 대한 예측 구조의 일례를 나타내는 도면이다. 설명의 편의상 도 9에서는 3대의 카메라에서 획득한 실제 영상과 실제 영상에 대응되는 깊이 정보 맵을 부호화하기 위한 예측 구조를 나타낸다.

[0131] 도 9에서는 3대의 카메라에서 획득된 3개의 실제 영상은 시점(view)에 따라 T0, T1, T2로 나타내었고, 실제 영상에 대응되는 3개의 깊이 정보 맵은 시점(view)에 따라 각각 D0, D1, D2로 나타내었다. 여기서, T0와 D0는 시점 0(View 0)에서 획득한 영상이고, T1와 D1는 시점 1(View 1)에서 획득한 영상이며, T2와 D2는 시점 2(View 2)에서 획득한 영상이다. 이때, 도 9에 도시된 사각형은 영상(픽처)를 나타낸다.

[0132] 각 영상(픽처)는 부호화/복호화 타입에 따라 I 픽처(Intra Picture), P 픽처(Uni-prediction Picture), B 픽처(Bi-prediction Picture)로 나뉘어지며, 각 픽처는 각 픽처의 부호화/복호화 타입에 따라 부호화/복호화될 수 있다. I 픽처에서는 인터 예측 없이 영상 자체를 부호화하고, P 픽처에서는 단방향에 존재하는 참조 영상만을 이용하여 인터 예측을 하며, B 픽처에서는 양방향에 존재하는 참조 영상을 이용하여 인터 예측을 수행할 수 있다. 이때, 도 9의 화살표는 예측 방향을 나타낸다. 즉, 예측 방향에 따라 실제 영상과 그것의 깊이 정보 맵은 서로 의존적으로 부호화/복호화될 수 있다.

[0134] 인터 예측을 통해 영상의 부호화/복호화를 수행하기 위해서는 현재 블록의 움직임 정보가 필요하다. 현재 불록의 움직임 정보를 유추하는 방법으로는 현재 블록과 인접한 블록의 움직임 정보를 이용하는 방법, 동일한 시점 내에서 시간적 상관성을 이용하는 방법, 또는 인접한 시점에서 시점 간 상관성을 이용하는 방법이 있으며, 상술한 인터 예측 방법은 한 픽처에서 혼용되어 사용될 수 있다. 여기서, 현재 블록은 예측이 수행되는 블록을 말한다. 움직임 정보는 움직임 벡터, 참조 영상 번호 및/또는 예측 방향(예컨대, 단방향 예측인지 양방향 예측인지, 시간적 상관성을 이용하는지, 또는 시점 간 상관성을 이용하는지 여부 등)을 의미할 수도 있다.

[0135] 이때, 예측 방향은 참조 픽처 목록(Reference Picture List: RefPicList)의 사용에 따라 크게 단방향 예측과 양방향 예측으로 나누어 질 수 있다. 단방향 예측은 순 방향 참조 픽처 목록 (LIST 0, L0)을 사용하는 순방향 예측(Pred_L0: Prediction L0)과 역 방향 참조 픽처 목록 (LIST 1, L1)을 사용하는 역방향 예측(Pred_L1: Prediction L1)으로 구분된다. 또한, 양방향 예측(Pred_BI: Prediction BI)은 순방향 참조 픽처 목록(LIST 0)과 역방향 참조 픽처 목록(LIST 1)을 모두 사용하여, 순방향 예측과 역방향 예측이 모두 존재하는 것을 말할 수 있으며, 순방향 참조 픽처 목록 (LIST 0)을 역방향 참조 픽처 목록 (LIST 1)에 복사하여 순방향 예측이 두 개 존재하는 경우도 양 방향 예측에 포함할 수 있다.

[0136] 예측 방향 여부는 predFlagL0, predFlagL1를 사용하여 정의할 수 있다. 이때, predFlagL0는 순방향 참조 픽처 목록(List 0)의 사용 여부를 지시하는 지시자이며, predFlagL1는 역방향 참조 픽처 목록(List 1)의 사용 여부를 지시하는 지시자에 해당된다. 예컨대, 단방향 예측이고 순방향 예측일 경우에는 predFlagL0가 '1'이 되고 predFlagL1은 '0'이 될 수 있고, 단방향 예측이고 역방향 예측일 경우에는 predFlagL0가 '0'이 되고

predFlagL1이 '1'이 될 수 있으며, 양방향 예측일 경우에는 predFlagL0가 '1'이 되고 predFlagL1이 '1'이 될 수 있다.

[0138] 도 10은 현재 블록에 대한 머지 후보 리스트(merge candidate list)를 구성하는 데 사용되는 주변 블록들을 나타낸 일례다.

[0139] 머지 모드(merge mode)는 인터 예측을 수행하는 방법 중 하나이며, 머지 모드에서는 현재 블록의 움직임 정보(예를 들어, 움직임 벡터, 참조 픽처 리스트, 참조 픽처 인덱스 중 적어도 하나)로서 현재 블록의 주변 블록의 움직임 정보를 이용할 수 있다. 이때, 주변 블록의 움직임 정보를 현재 블록의 움직임 정보로 사용하는 것을 머징(merging), 움직임 병합(motion merging) 또는 병합 움직임 (merging motion)이라고 한다.

[0140] 머지 모드에서는 부호화 유닛(Coding Unit: CU, 이하 'CU'라 함) 단위의 병합 움직임(merging)과 예측 유닛(Prediction Unit: PU, 이하 'PU'라 함) 단위의 병합 움직임이 가능하다.

[0141] 블록 단위(예컨대, CU 또는 PU)(이하, 설명의 편의를 위해 '블록'이라 함) 로 병합 움직임을 수행하는 경우에는, 블록 파티션(partition)별로 병합 움직임을 수행할지 여부에 관한 정보와 현재 블록에 인접한 주변 블록 중 어떤 블록과 병합 움직임을 할 것인지에 대한 정보가 필요하다.

[0142] 병합 움직임을 수행하기 위해서 머지 후보 리스트(merging candidate list)를 구성할 수 있다.

[0143] 머지 후보 리스트는 움직임 정보들의 리스트를 나타내며, 머지 모드가 수행되기 전에 생성될 수 있다. 여기서, 머지 후보 리스트의 움직임 정보는 현재 블록에 인접한 주변 블록의 움직임 정보이거나, 이미 머지 후보 리스트에 존재하는 움직임 정보들을 조합하여 만든 새로운 움직임 정보일 수도 있다. 주변 블록의 움직임 정보(예컨대, 움직임 벡터 및/또는 참조 픽처 인덱스)는 주변 블록에 의해 특정되는 혹은 주변 블록에 저장된(주변 블록의 디코딩에 사용된) 움직임 정보일 수 있다.

[0144] 이때, 주변 블록은 도 10에 도시된 바와 같이, 현재 블록과 공간적(spatial)으로 인접하여 위치한 주변 블록(neighboring block)(A, B, C, D, E)과, 현재 블록과 시간적(temporal)으로 대응하는 동일 위치의 후보 블록(co-located block)(H 혹은 M)을 포함할 수 있다. 동일 위치의 후보 블록은 현재 블록을 포함하는 현재 픽처와 시간적으로 대응하는 동일 위치의 픽처(co-located picture) 내 대응 위치의 블록을 말한다. 동일 위치의 픽처 내 H 블록이 가용하면 H 블록을 동일 위치의 후보 블록으로 결정하고, H 블록이 가용하지 않으면 동일 위치의 픽처 내 M 블록을 동일 위치의 후보 블록으로 결정할 수 있다.

[0145] 머지 후보 리스트의 구성시, 주변 블록(A, B, C, D, E) 및 동일 위치의 후보 블록(H 혹은 M)의 움직임 정보가 현재 블록의 머지 후보 리스트를 구성하는 머지 후보(merge candidate)로 이용될 수 있는지 여부가 판단된다. 즉, 현재 블록의 인터 예측에 이용 가능한 블록의 움직임 정보가 머지 후보로서 머지 후보 리스트에 추가될 수 있다.

[0146] 예컨대, X 블록에 대한 머지 후보 리스트를 구성하는 방법으로, 1) 우선 주변 블록 A가 사용 가능한 경우, 주변 블록 A를 머지 후보 리스트에 포함한다. 2) 이후, 주변 블록 B의 움직임 정보가 주변 블록 A의 움직임 정보와 동일하지 않을 경우에만 주변 블록 B를 머지 후보 리스트에 포함한다. 3) 동일한 방법으로 주변 블록 C의 움직임 정보가 주변 블록 B의 움직임 정보와는 다를 경우에만 주변 블록 C를 머지 후보 리스트에 포함하고, 4) 주변 블록 D의 움직임 정보가 주변 블록 C의 움직임 정보와는 다를 경우에만 주변 블록 D를 머지 후보 리스트에 포함한다. 아울러, 5) 주변 블록 E의 움직임 정보가 주변 블록 D의 움직임 정보와 다를 경우에만 주변 블록 E를 머지 후보 리스트에 포함하고, 6) 마지막으로, 머지 후보 리스트에는 주변 블록 H(혹은 M)를 포함시킨다. 즉, 각각의 주변 블록은 A→B→C→D→E→H(혹은 M) 블록 순서로 머지 후보 리스트에 추가될 수 있다. 여기서, 동일한 움직임 정보는 동일한 움직임 벡터, 동일한 참조 픽처, 및 동일한 예측 방향(단방향, 양방향)을 사용하는 것을 의미할 수 있다.

[0147] 여기서는, 주변 블록을 머지 후보로서 머지 후보 리스트에 추가한다는 표현과 주변 블록의 움직임 정보를 머지 후보로서 머지 후보 리스트에 추가한다는 표현을 혼용하였으나, 이는 설명의 편의를 위한 것으로서, 두 표현은 실질적으로 상이하지 않다. 예컨대, 머지 후보로서의 주변 블록은 해당 블록의 움직임 정보를 의미한다고 할 수 있다.

[0149] 도 11은 인접한 시점의 움직임 정보를 이용하여, 현재 블록의 움직임 정보를 유도하는 과정의 일례를 나타내는 도면이다.

[0150] 도 11에서는 설명의 편의를 위하여 하나의 인접한 시점만을 이용하여 현재 블록의 움직임 정보를 유도하는 과정을 예시로 들고 있으나, 인접한 시점은 2개 이상일 수도 있다.

[0151] 도 11을 참조하면, 3차원 비디오 시스템에서는 움직임 정보를 효율적으로 부호화/복호화 하기 위하여, 인접한 시점의 움직임 정보를 이용할 수 있다. 구체적으로, 도 11의 현재 블록(현재 위치 X에 대한 블록)은 현재 블록에 대한 움직임 정보를 유도하기 위하여, 인접한 시점에 위치한 대상 블록(참조 위치 XR)을 찾는다. 이때, 인접한 시점에 위치한 대상 블록은 현재 블록에 대응하는 블록을 의미하며, 현재 시점에서의 현재 픽처와 참조 시점에서의 현재 픽처는 카메라 위치의 차이만이 있을 뿐이므로, 상술한 바와 같이 변이 벡터(Disparity Vector, DV)를 이용하여 인접한 시점에 위치한 대상 블록을 유도할 수 있다.

[0153] 도 12는 하나의 예측 유닛(Prediction Unit, PU)이 여러 개의 서브 예측 유닛(Sub-Prediction Unit)으로 나누어지는 일례를 나타내는 도면이다.

[0154] 도 12에서는 64x64 크기를 가지는 예측 유닛(Prediction Unit, PU)이 8x8 크기의 서브 예측 유닛으로 나누어지는 일례를 나타낸다. 도 12에서는 설명의 편의를 위하여, 예측 유닛이 64x64의 크기를 가지는 것으로 설명하나, 예측 유닛은 64x64 크기뿐만 아니라, 32x32, 16x16, 8x8, 4x4 등의 크기를 가질 수도 있다. 3차원 비디오 시스템에서는 하나의 예측 유닛이 여러 개의 서브 예측 유닛(Sub-Prediction Unit)으로 분할될 수 있다. 이때, 변이 벡터를 이용한 움직임 정보의 유도는 서브 예측 유닛 단위로 수행된다. 서브 예측 유닛의 크기는 기 설정된 크기(예컨대, 4x4, 8x8, 16x16 등)를 가질 수 있으며, 서브 예측 유닛의 크기는 부호화 시에 지정될 수 있다. 서브 예측 유닛의 크기에 대한 정보는 비디오 파라미터 세트 확장 신텍스(Video Parameter Set Extension syntax, VPS Extension syntax) 내에 포함되어 시그널링될 수 있다.

[0156] 도 13은 참조 블록을 이용하여 현재 블록의 움직임 정보를 유도하는 과정의 일례를 나타내는 도면이다.

[0157] 현재 블록에 대한 움직임 정보의 유도 과정은 참조 블록에 존재하는 움직임 정보를 현재 블록의 움직임 정보로 설정하는 것을 의미한다. 하지만, 3차원 비디오 시스템에서는 움직임 정보를 효율적으로 부호화/복호화 하기 위하여, 현재 시점(view)에서 현재 픽쳐에 위치한 현재 블록(X)에 대해 각각의 서브 예측 유닛 단위로 움직임 정보를 유도할 수 있다.

[0158] 즉, 3차원 비디오 시스템에서는 참조 블록(XR)의 서브 예측 유닛에 존재하는 움직임 정보를 현재 블록(X)의 서브 예측 유닛에 대한 움직임 정보로 설정할 수 있다. 이때, 참조 블록(XR)은 참조 시점에서의 현재 픽쳐에 위치한 참조 블록(XR)을 의미할 수 있으며, 구체적인 움직임 정보 유도 과정은 후술하도록 한다.

[0160] 도 14는 현재 블록의 움직임 정보를 유도하는데 사용되는 참조 블록의 일례를 도시한 것이다.

[0161] 도 14에 따르면, 참조 블록은 PU를 의미할 수 있으며, 하나의 참조 블록은 총 16개의 서브 예측 유닛을 포함할 수 있다. 이때, 현재 블록의 각각의 서브 예측 유닛은 참조 블록의 서브 예측 유닛에 존재하는 움직임 정보를 이용하여, 현재 블록의 서브 유닛에 대한 움직임 정보를 유도할 수 있다.

[0162] 이하, 도 15a 내지 도 15e와, 도 16a 내지 도 16g에서는 참조 블록을 이용하여 현재 블록의 서브 예측 유닛에 대한 움직임 정보를 유도하는 방법을 도면을 통해 구체적으로 설명하도록 한다.

[0164] 도 15a 내지 도 15e는 저장 공간에 저장된 움직임 정보를 이용하여, 움직임 정보를 유도하는 과정의 일례를 개략적으로 설명하는 도면들이다. 이때, 도 15a 내지 도 15e에서 사용되는 참조 블록은 도 14의 참조 블록일 수 있다.

[0165] 현재 블록의 서브 예측 유닛이 참조 블록의 서브 예측 유닛에 대한 움직임 정보를 가져올 때, 참조 블록의 모든 서브 예측 유닛이 움직임 정보를 가지고 있는 것은 아니다. 즉, 참조 블록의 서브 예측 유닛 중에서는 움직임 정보를 가져올 수 없는 서브 예측 유닛이 있을 수 있다. 따라서, 움직임 정보를 가져올 수 없는 서브 예측 유닛

이 존재할 경우에는 현재 블록의 서브 예측 유닛이 움직임 정보를 유도하지 못하는 것을 보완하기 위하여, 현재 참조되고 있는 서브 예측 유닛의 이전 또는 이후에 존재하는 서브 예측 유닛의 움직임 정보를 이용할 수 있다. 예컨대, 참조 블록 내에서 사용 가능한 서브 예측 유닛의 움직임 정보를 미리 저장해두었다가, 움직임 정보 유도가 불가능한 참조 블록의 서브 예측 유닛이 있을 경우에는 미리 저장된 움직임 정보를 현재 블록의 서브 예측 유닛에 삽입하여, 현재 블록에 대한 움직임 정보의 유도를 수행할 수 있다.

[0166] 이하, 상술한 방법에 대한 원활한 이해를 돕기 위하여 참조 블록의 첫 번째 서브 예측 유닛은 움직임 정보를 가지고 있고, 참조 블록의 두 번째 서브 예측 유닛 이후에 움직임 정보가 존재하지 않는 서브 예측 유닛이 발생할 경우, 현재 블록의 서브 유닛에 대한 움직임 정보를 유도하는 방법의 일례를 도면을 통해 단계별로 설명하도록 한다.

[0168] 도 15a는 현재 블록의 서브 예측 유닛과, 저장 공간의 초기 상태를 도시한 도면이다.

[0169] 도 15a를 참조하면, Ref는 참조 블록을 의미하며, Ref 0, 1, 2, 3은 참조 블록 내 각 서브 예측 유닛을 의미한다. 즉, Ref 0은 참조 블록의 서브 예측 유닛 0(참조 블록의 첫 번째 서브 예측 유닛), Ref 1은 참조 블록의 서브 예측 유닛 1(참조 블록의 두 번째 서브 예측 유닛), Ref 2는 참조 블록의 서브 예측 유닛 2(참조 블록의 세 번째 서브 예측 유닛), Ref 3은 참조 블록의 서브 예측 유닛 3(참조 블록의 네 번째 서브 예측 유닛)을 의미한다. Cur은 현재 블록을 의미하며, Cur 0, 1, 2, 3은 현재 블록 내 각 서브 예측 유닛을 의미한다. 즉, Cur 0은 현재 블록의 서브 예측 유닛 0(현재 블록의 첫 번째 서브 예측 유닛), Cur 1은 현재 블록의 서브 예측 유닛 1(현재 블록의 두 번째 서브 예측 유닛), Cur 2는 현재 블록의 서브 예측 유닛 2(현재 블록의 세번째 서브 예측 유닛), Cur 3은 현재 블록의 서브 예측 유닛 3(현재 블록의 네 번째 서브 예측 유닛)을 의미한다.

[0170] 이때, 도 15a의 Ref 2의 'X' 는 참조 블록의 서브 예측 유닛 2를 이용해서는 움직임 정보 유도가 불가능하다는 것을 나타낸다.

[0172] 도 15b는 참조 블록의 서브 예측 유닛으로부터 움직임 정보를 유도하는 제1 단계를 나타낸다.

[0173] 도 15b를 참조하면, 현재 블록의 서브 예측 유닛 0은 참조 블록의 서브 예측 유닛 0으로부터 움직임 정보를 유도한다. 이때, 참조 블록의 서브 예측 유닛 0으로부터 움직임 정보 유도가 가능하므로, 참조 블록의 서브 예측 유닛 0의 움직임 정보를 저장 공간에 저장한다. 이때, 저장 공간에 저장된 움직임 정보는 움직임 정보 0이라고 정의될 수 있으며, 움직임 정보 0은 움직임 정보 유도가 불가능한 참조 블록의 서브 예측 유닛이 발생할 경우에 사용된다.

[0175] 도 15c는 참조 블록의 서브 예측 유닛으로부터 움직임 정보를 유도하는 제2 단계를 나타낸다.

[0176] 도 15c를 참조하면, 현재 블록의 서브 예측 유닛 1은 참조 블록의 서브 예측 유닛 1로부터 움직임 정보를 유도한다. 이때, 참조 블록의 서브 예측 유닛 1로부터 움직임 정보 유도가 가능하므로, 참조 블록의 서브 예측 유닛 1의 움직임 정보를 저장 공간에 저장한다. 이때, 저장되는 서브 예측 유닛 1의 움직임 정보는 움직임 정보 1이라고 정의될 수 있으며, 움직임 정보 1은 움직임 정보 0 대신에 저장 공간에 저장될 수 있다. 움직임 정보 1은 움직임 정보 유도가 불가능한 참조 블록의 서브 예측 유닛이 발생할 경우에 사용될 수 있다.

[0178] 도 15d는 참조 블록의 서브 예측 유닛으로부터 움직임 정보를 유도하는 제3 단계를 나타낸다.

[0179] 도 15d를 참조하면, 현재 블록의 서브 예측 유닛 2는 참조 블록의 서브 예측 유닛 2로부터 움직임 정보의 유도를 시도한다. 하지만, 참조 블록의 서브 예측 유닛 2는 움직임 정보의 유도가 불가능한 서브 예측 유닛이므로, 현재 블록의 서브 예측 유닛 2는 저장 공간에 저장된 움직임 정보를 이용하여, 현재 블록의 서브 예측 유닛 2의 움짐임 정보를 유도한다. 이때, 저장 공간에 저장된 움직임 정보는 움직임 정보 1일 수 있다.

[0181] 도 15e는 참조 블록의 서브 예측 유닛으로부터 움직임 정보를 유도하는 제4 단계를 나타낸다.

[0182] 도 15e를 참조하면, 현재 블록의 서브 예측 유닛 3은 참조 블록의 서브 예측 유닛 3으로부터 움직임 정보를 유

도한다. 이때, 참조 블록의 서브 예측 유닛 3으로부터 움직임 정보 유도가 가능하므로, 참조 블록의 서브 예측 유닛 3의 움직임 정보를 저장 공간에 저장한다. 이때, 저장되는 서브 예측 유닛 3의 움직임 정보는 움직임 정보 3이라고 정의될 수 있으며, 움직임 정보 3은 움직임 정보 1 대신에 저장 공간에 저장될 수 있다. 움직임 정보 3은 움직임 정보 유도가 불가능한 참조 블록의 서브 예측 유닛이 발생할 경우에 사용될 수 있다.

[0184] 도 16a 내지 도 16g는 저장 공간에 저장된 움직임 정보를 이용하여, 움직임 정보를 유도하는 과정의 다른 예를 개략적으로 설명하는 도면이다.

[0185] 도 16a 내지 도 16g에서는 저장된 움직임 정보가 없는 상태에서, 움직임 정보 유도가 불가능한 참조 블록의 서브 예측 유닛이 등장하고, 이후에 움직임 정보 유도가 가능한 참조 블록의 서브 예측 유닛이 등장하는 경우의 움직임 정보 유도 과정의 예를 나타낸다.

[0187] 도 16a는 현재 블록의 서브 예측 유닛과, 저장 공간의 초기 상태를 도시한 도면이다.

[0188] 도 16a를 참조하면, Ref는 참조 블록을 의미하며, Ref 0, 1, 2, 3은 참조 블록 내 각 서브 예측 유닛을 의미한다. 즉, Ref 0은 참조 블록의 서브 예측 유닛 0, Ref 1은 참조 블록의 서브 예측 유닛 1, Ref 2는 참조 블록의 서브 예측 유닛 2, Ref 3은 참조 블록의 서브 예측 유닛 3을 의미한다. Cur은 현재 블록을 의미하며, Cur 0, 1, 2, 3은 현재 블록 내 각 서브 예측 유닛을 의미한다. 즉, Cur 0은 현재 블록의 서브 예측 유닛 0, Cur 1은 현재 블록의 서브 예측 유닛 1, Cur 2는 현재 블록의 서브 예측 유닛 2, Cur 3은 현재 블록의 서브 예측 유닛 3을 의미한다. 이때, 도 16a의 Ref 0 및 1의 'X' 는 참조 블록의 서브 예측 유닛 0 및 참조 블록의 서브 예측 유닛 1을 이용해서는 움직임 정보 유도가 불가능하다는 것을 나타낸다.

[0190] 도 16b는 참조 블록의 서브 예측 유닛으로부터 움직임 정보를 유도하는 제1 단계를 나타낸다.

[0191] 도 16b를 참조하면, 현재 블록의 서브 예측 유닛 0은 참조 블록의 서브 예측 유닛 0으로부터 움직임 정보의 유도를 시도한다. 하지만, 상술한 바와 같이 참조 블록의 서브 예측 유닛 0으로부터는 움직임 정보의 유도가 불가능하고, 저장 공간에 저장된 움직임 정보 또한 없으므로, 제2 단계를 수행한다.

[0193] 도 16c는 참조 블록의 서브 예측 유닛으로부터 움직임 정보를 유도하는 제2 단계를 나타낸다.

[0194] 도 16c를 참조하면, 현재 블록의 서브 예측 유닛 1은 참조 블록의 서브 예측 유닛 1로부터 움직임 정보의 유도를 시도한다. 하지만, 상술한 바와 같이 참조 블록의 서브 예측 유닛 1로부터도 움직임 정보의 유도가 불가능하고, 저장 공간에 저장된 움직임 정보 또한 없으므로, 제3 단계를 수행한다.

[0196] 도 16d는 참조 블록의 서브 예측 유닛으로부터 움직임 정보를 유도하는 제3 단계를 나타낸다.

[0197] 도 16d를 참조하면, 현재 블록의 서브 예측 유닛 2는 참조 블록의 서브 예측 유닛 2로부터 움직임 정보를 유도한다. 이때, 참조 블록의 서브 예측 유닛 2로부터 움직임 정보 유도가 가능하므로, 참조 블록의 서브 예측 유닛 2의 움직임 정보를 저장 공간에 저장한다. 이때, 저장 공간에 저장된 움직임 정보는 움직임 정보 2라고 정의할 수 있으며, 움직임 정보 2는 움직임 정보 유도가 불가능한 참조 블록의 서브 예측 유닛이 발생할 경우에 사용된다.

[0199] 도 16e는 참조 블록의 서브 예측 유닛으로부터 움직임 정보를 유도하는 제4 단계를 나타낸다.

[0200] 도 16e를 참조하면, 현재 블록의 서브 예측 유닛 0은 저장 공간에 저장된 움직임 정보 2를 이용하여 현재 블록에 대한 서브 예측 유닛 0의 움직임 정보를 유도한다.

[0202] 도 16f는 참조 블록의 서브 예측 유닛으로부터 움직임 정보를 유도하는 제5 단계를 나타낸다.

[0203] 도 16f를 참조하면, 현재 블록의 서브 예측 유닛 1은 저장 공간에 저장된 움직임 정보 2를 이용하여 현재 블록에 대한 서브 예측 유닛 0의 움직임 정보를 유도한다.

[0205] 도 16g는 참조 블록의 서브 예측 유닛으로부터 움직임 정보를 유도하는 제6 단계를 나타낸다.

[0206] 도 16g를 참조하면, 현재 블록의 서브 예측 유닛 3은 참조 블록의 서브 예측 유닛 3으로부터 움직임 정보를 유도한다. 이때, 참조 블록의 서브 예측 유닛 3으로부터 움직임 정보 유도가 가능하므로, 참조 블록의 서브 예측 유닛 3의 움직임 정보를 저장 공간에 저장한다. 이때, 저장되는 서브 예측 유닛 3의 움직임 정보는 움직임 정보 3이라고 정의할 수 있으며, 움직임 정보 3은 움직임 정보 2 대신에 저장 공간에 저장될 수 있다. 움직임 정보 3은 움직임 정보 유도가 불가능한 참조 블록의 서브 예측 유닛이 발생할 경우에 사용될 수 있다.

[0208] 도 17은 일례에 따른, 참조 블록의 서브 예측 유닛을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 방법의 순서도다. 도 17 과정의 동작은 인코더 및/또는 디코더에서 수행될 수 있으며, 인코더 및/또는 디코더 내의 인터 예측부, 예컨대 도 7의 인터 예측부(720) 또는 도 8의 인터 예측부(850)에서 수행될 수도 있다.

[0209] 우선, 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하는 경우의 과정을 도 17을 참조하여 설명하면, 인터 예측부는 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하는지 여부에 대해서 판단한다(S1700).

[0210] 인터 예측부는 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하는 경우, 참조 블록의 서브 예측 유닛에 존재하는 움직임 정보를, 움직임 정보 유도 대상인 현재 블록의 서브 예측 유닛에 삽입한다(S1710).

[0211] 이후, 인터 예측부는 저장 공간에 움직임 정보가 저장되어 있는지 여부를 판단한다(S1720). 만약 저장 공간에 저장돼 있는 움직임 정보가 있을 경우에는 단계 S1750을 수행한다. 이때, 저장 공간에 대한 구체적인 설명은 상술한 바와 같으며, 움직임 정보 또한 상술한 바와 같다.

[0212] 만약, 저장 공간에 저장돼 있는 움직임 정보가 없는 경우, 인터 예측부는 움직임 정보 유도 대상인 현재 블록의 서브 예측 유닛이 현재 블록의 첫 번째 서브 예측 유닛인지 여부를 판단한다(S1730). 움직임 정보 유도 대상인 현재 블록의 서브 예측 유닛이 현재 블록의 첫 번째 서브 예측 유닛에 해당하는 경우에, 인터 예측부는 단계 S1750을 수행한다.

[0213] 단계 S1730에서, 현재 블록의 서브 예측 유닛이 첫 번째 서브 예측 유닛이 아닌 경우, 인터 예측부는 현재 블록의 첫 번째 서브 예측 유닛부터 움직임 정보 유도 대상인 현재 블록의 서브 예측 유닛 직전까지 존재하는 서브 예측 유닛에, 참조 블록의 서브 예측 유닛에 존재하는 움직임 정보를 삽입한다. 예컨대 움직임 정보 유도 대상인 현재 블록의 서브 예측 유닛이 세 번째 서브 예측 유닛이라면, 인터 예측부는 참조 블록의 서브 예측 유닛에 대한 움직임 정보를 현재 블록의 첫 번째 서브 예측 유닛부터 두 번째 서브 예측 유닛까지 삽입한다.

[0214] 인터 예측부는 저장 공간에 참조 블록의 서브 예측 유닛에 대한 움직임 정보를 저장하고, 갱신한다(S1750). 이때, 움직임 정보의 저장 및 갱신에 관한 구체적인 내용은 상술한 바와 같다.

[0215] 인터 예측부는 움직임 정보 유도 대상인 참조 블록의 서브 예측 유닛이 참조 블록에서의 마지막 서브 예측 유닛인지 여부를 판단한다(S1790). 움직임 정보 유도 대상인 참조 블록의 서브 예측 유닛이 참조 블록에서의 마지막 서브 예측 유닛인 경우에는 움직임 정보 유도 과정을 종료한다. 만약, 움직임 정보 유도 대상인 참조 블록의 서브 예측 유닛이 참조 블록에서의 마지막 서브 예측 유닛이 아닌 경우에, 인터 예측부는 참조 블록의 다음 서브 예측 유닛으로 처리 대상을 이동한다(S1780). 이후, 인터 예측부는 단계 S1700 내지 S1790 과정을 다시 수행한다.

[0217] 만약, 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하지 않는 경우에는 다음과 같은 과정이 수행된다.

[0218] 인터 예측부는 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하는지 여부에 대해서 판단한다(S1700).

[0219] 만약 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하지 않는다면, 인터 예측부는 저장 공간에 저장된 움직임 정보가 있는지 여부를 판단한다(S1770). 만약 저장 공간에 저장된 움직임 정보가 없다면, 인터 예측부는 단계 S1790 과정을 수행한다.

[0220] 저장 공간에 저장된 움직임 정보가 있는 경우, 인터 예측부는 움직임 정보 유도 대상인 서브 예측 유닛에 저장 공간에 저장된 움직임 정보를 삽입한다(S1750).

[0221] 인터 예측부는 상술한 과정을 수행한 후에, 움직임 정보 유도 대상인 참조 블록의 서브 예측 유닛이 참조 블록에서의 마지막 서브 예측 유닛인지 여부를 판단한다(S1790). 움직임 정보 유도 대상인 참조 블록의 서브 예측 유닛이 참조 블록에서의 마지막 서브 예측 유닛인 경우에는 움직임 정보 유도 과정을 종료한다. 만약, 움직임 정보 유도 대상인 참조 블록의 서브 예측 유닛이 참조 블록에서의 마지막 서브 예측 유닛이 아닌 경우에 인터 예측부는 참조 블록의 다음 서브 예측 유닛으로 처리 대상을 이동한다(S1780). 이후, 인터 예측부는 단계 S1700 내지 S1790 과정을 수행한다.

[0222] 이후, 상술한 과정을 통해 유도된 현재 블록에 대한 움직임 정보를 기반으로 하여, 인터 예측부는 현재 블록에 대한 예측 샘플을 유도한다. 예측 샘플은 상술한 예측 신호를 의미할 수 있으며, 상술한 바와 같이 예측 신호는 원 신호와 레지듀얼 신호의 차이를 의미할 수 있다.

[0223] 상술한 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 과정을 3D 비디오에 대하여 구체적으로 적용하면 아래 표 1과 같다. 상술한 바와 같이, 표 1의 동작은 인코더/디코더 혹은 인코더/디코더의 인터 예측부에서 수행될 수 있다.

[0224] <표 1>

이 프로세스의 입력은 다음과 같다. - 현재 예측 유닛의 좌상단 위치 (xPb, yPb) - 현재 예측 유닛의 너비(nPbW)와 높이(nPbH) - 참조 시점 인덱스 refViewIdx - 변이 벡터 mvDisp 이 프로세스의 출력은 다음과 같다. - 시간적 시점 간 움직임 후보가 사용 가능한지의 여부를 판단하는 플래그 availableFlagLXInterView, 이 때 LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. 본 명세서에서 '시간적 시점 간(temporal inter-view)'이란, 현재 픽처와 다른 시점(view)에 있는 픽처를 현재 픽처와 동일 시점에 있는 다른 시간(즉, 다른 POC)의 픽처처럼 참조할 수 있다는 것을 의미한다. - 시간적 시점 간 움직임 벡터 후보 mvLXInterView, 이 때 LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. - 참조 픽처 리스트 RefPicListLX에 존재하는 참조 픽처를 지정하는 참조 인덱스 refIdxLXInterView. 이 때 LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. LX에 대해 다음을 적용한다. - 플래그인 availableFlagLXInterView는 0으로 초기화한다. - 움직임 벡터 mvLXInterView는 (0, 0)으로 초기화한다.

[0225]

- 참조 인덱스 refIdxLXInterView는 -1로 초기화한다.

변수 nSbW와 nSbH는 다음과 같이 초기화한다.

nSbW = Min(nPbW, SubPbSize)

nSbH = Min(nPbH, SubPbSize)

변수 ivRefPic은 현재 액세스 유닛에서 refViewIdx의 값과 같은 ViewIdx를 가지는 픽처로 초기화된다. 변수 curSubBlockIdx는 0으로 초기화하고 변수 lastAvalableFlag도 0으로 초기화한다.

0부터 (nPbH / nSbH - 1)의 범위를 가지는 yBlk와, 0부터 (nPbW/nSbW - 1)의 범위를 가지는 xBlk에 대하여 다음을 적용한다.

- 변수 curAvailableFlag 는 0으로 초기화한다.
- 0부터 1의 범위를 가지는 X에 대해서 다음을 적용한다.
  - 플래그 spPredFlagL1[xBlk][yBlk]는 0으로 초기화한다.
  - 움직임 벡터 spMvLX는 (0, 0)으로 초기화한다.
  - 참조 인덱스 spRefIdxLX[xBlk][yBlk]는 -1로 초기화한다.
- 참조 블록 위치 (xRef, yRef)는 다음과 같이 유도된다.

```
xRef = Clip3( 0, PicWidthInSamplesL - 1,
        xPb + xBlk * nSbW + nSbW / 2 + ( ( mvDisp[ 0 ] + 2) >> 2 ) )
yRef = Clip3( 0, PicHeightInSamplesL - 1,
```

[0226]

yPb + yBlk * nSbH + nSbH / 2 + ((mvDisp[1] + 2) >> 2))

- 변수 ivRefPb는 ivRefPic이 가리키는 시점 간 참조 픽처 내의 (xRef, yRef) 위치에서의 휘도(luma) 예측 블록을 나타낸다.

- (xIvRefPb, yIvRefPb)는 ivRefPb가 가리키는 참조 블록의 좌상단 위치를 나타낸다.

- ivRefPb가 인트라 모드로 부호화 되지 않았다면 0부터 1의 범위를 가지는 X에 대해서 다음을 수행한다.

  - X가 0이거나 현재 슬라이스가 B 슬라이스 일 때 X에서 (1 - X)의 범위를 가지는 Y에 대해서 다음을 수행한다.

    - refPixListLYIvRef, predFlagLYIvRef[x][y], mvLYIvRef[x][y], refIdxLYIvRef[x][y]는 각각 ivRefPic가 가리키는 픽처 내의 RefPicListLY, PredFlagLY[x][y], MvLY[x][y], RefIdxLY[x][y]로 세팅된다.

    - predFlagLYIvRef[xIvRefPb][yIvRefPb]가 1이면, 0부터 num_ref_idx_lX_active_minus1(참조 픽처 리스트 내 참조 픽처 개수)의 범위를 가지는 i에 대해서 다음을 수행한다.

    - refPicListLYIvRef[refIdxLYIvRef[xIvRefPb][yIvRefPb]]의 POC가 RefPicListLX[i]와 같고 spPredFlagLX[xBlk][yBlk]가 0이라면, 다음을 적용한다.

      spMvLX[xBlk][yBlk] = mvLYIvRef[xIvRefPb][yIvRefPb]

      spRefIdxLX[xBlk][yBlk] = i

[0227]

```
            spPredFlagLX[ xBlk ][ yBlk ] = 1
            curAvailableFlag = 1


curAvailableFlag에 따라 다음을 적용한다
  - curAvailableFlag가 1이라면 다음의 순서를 적용한다.
    1. lastAvailableFlag가 0이라면 다음을 적용한다.
      - 0부터 1의 범위를 가지는 X에 대해서 다음을 적용한다
        mxLXInterView = spMvLX[ xBlk ][ yBlk ]
        refIdxLXInterView = spRefIdxLX[ xBlk ][ yBlk ]
        availableFlagLXInterview = spPredFlag[ xBlk ][ yBlk ]
      - curSubBlockIdx가 0보다 클 때, 0부터 (curSubBlockIdx - 1)의
      범위를 가지는 k에 대하여 다음을 적용한다.
        - 변수 i와 j는 다음과 같이 유도된다
          i = k % (nPSW / nSbW)
          j = k / (nPSW / nSbW)
        - 0부터 1의 범위를 가지는 X에 대해, 다음을 적용한다.
          spMvLX[ i ][ j ] = spMvLX[ xBlk ][ yBlk ]
          spRefIdxLX[ i ][ j ] = spRefIdxLX[ xBlk ][ yBlk ]
          spPredFlagLX[ i ][ j ] = spPredFlagLX[ xBlk ][ yBlk ]
    2. 변수 lastAvailableFlag에 1을 대입한다.
    3. 변수 xLastAvail과 yLastAvail에 각각 xBlk와 yBlk을 저장한다.
```

[0228]

```
  - curAvailableFlag가 0이고, lastAvailableFlag가 1이라면, 0부터 1의
  범위를 가지는 X에 대해서 다음을 적용한다.
    spMvLX[ xBlk ][ yBlk ] = spMvLX[ xLastAvail ][ yLastAvail ]
    spRefIdxLX[ xBlk ][ yBlk ] = spRefIdxLX[ xLastAvail ][ yLastAvail ]
    spPredFlagLX[ xBlk ][ yBlk ] =
    spPredFlagLX[ xLastAvail ][ yLastAvail ]
  - 변수 curSubBlockIdx는 curSubBlockIdx + 1로 세팅된다.
```

[0229]

[0230] 표 1을 구체적으로 설명하면 아래와 같다.

[0231] 표 1을 참조하면, 인터 예측부에는 현재 예측 블록의 좌상단의 위치, 현재 예측 블록의 너비와 현재 예측 블록의 높이, 참조 시점 인덱스, 및 변이 벡터가 입력된다. 이때, 현재 예측 블록의 좌상단의 위치는 (xPb, yPb)로 표시될 수 있으며, 'xPb'는 현재 예측 블록의 x축 좌표를, 'yPb'는 현재 예측 블록의 y축 좌표를 의미할 수 있다. 현재 예측 블록의 너비는 'nPbW'로 표시될 수 있으며, 현재 예측 블록의 높이는 'nPbH'로 표시될 수 있다. 참조 시점 인덱스는 'refViewIdx'로 표시될 수 있으며, 변이 벡터는 'mvDisp'로 표시될 수 있다. 이

때, 인터 예측부는 상술한 비디오 인코더/디코더의 인터 예측부에 대응될 수 있다.

[0232] 또한, 도 17을 참조하면, 참조 블록의 서브 예측 유닛을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 과정을 마친 경우, 인터 예측부에서는 시간적 시점 간 움직임 후보(temporal inter-view motion candidate)가 사용 가능한지의 여부를 판단하는 플래그와, 시간적 시점 간 움직임 벡터 후보(temporal inter-view motion vector candidate), 참조 픽처 리스트에 존재하는 참조 픽처를 지정하는 참조 인덱스를 출력한다. 이때, 시간적 시점 간 움직임 후보가 사용 가능한지의 여부를 판단하는 플래그는 'availableFlagLXInterView'로 정의될 수 있으며, 시간적 시점 간 움직임 벡터 후보는 'mvLXInterView'로 정의될 수 있다. 또한, 참조 픽처 리스트는 'RefPicListLX'로 표시될 수 있으며, 참조 픽처 리스트에 존재하는 참조 픽처를 지정하는 참조 인덱스는 'refIdxLXInterView'로 정의될 수 있다. 'availableFlagLXInterView', 'mvLXInterView', 'RefPicListLX'', 및 'refIdxLXInterView'에 존재하는 'LX'는 참조 픽처 리스트 0(List 0, L0) 또는 참조 픽처 리스트 1(List 1, L1)이 될 수 있다.

[0233] 이하, 인터 예측부에서 상술한 입력으로부터 상술한 출력을 도출하기 위한, 참조 블록의 서브 예측 유닛을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 방법을 구체적으로 설명한다.

[0234] 우선, 인터 예측부는 참조 블록의 서브 예측 유닛을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하기에 앞서, 초기화를 수행한다. 이때, availableFlagLXInterView는 0으로 설정되고, mvLXInterView는 (0,0)으로 설정되며, refIdxLXInterView는 -1로 설정된다. 인터 예측부에서 초기화를 수행할 때, 서브 예측 유닛의 너비와 서브 예측 유닛의 높이 또한 초기화 된다. 이때, 서브 예측 유닛의 너비는 'nSbW'로 표시될 수 있으며, 서브 예측 유닛의 높이는 'nSbH'로 표시될 수 있다. 변수 nSbW와 nSbH의 구체적인 초기화 방법은 아래 수학식 1과 같다.

[0235] <수학식 1>

$$nSbW = Min(nPbW, SubPbSize[\ nuh_layer_id\ ])$$

$$nSbH = Min(nPbH, SubPbSize[\ nuh_layer_id\ ])$$

[0236]

[0237] 이때, SubPbSize는 VPS(video pareameter set)에서 지정한 서브 예측 유닛의 높이, 너비의 크기를 의미하며, nuh_layer_id는 레이어를 식별(예컨대, 어느 참조 시점인지 여부 등)하는 인덱스를 의미한다. Min()은 입력되는 변수 중 작은 값을 가지는 변수를 출력하기 위한 것으로, 아래 수학식 2와 같이 정의될 수 있다.

[0238] <수학식 2>

$$Min(x,y) = \begin{cases} x; x \leq y \\ y; x > y \end{cases}$$

[0239]

[0240] 인터 예측부에서는 상술한 변수들뿐만 아니라, 시점 간 참조 픽처 내의 (xRef, yRef) 위치에서의 휘도(luma) 예측 블록과 현재 블록의 서브 예측 유닛을 식별하는 정보, 및 저장 공간에 저장된 움직임 정보의 사용 가능 여부를 식별하는 정보 또한 초기화될 수 있.

[0241] 이때, 시점 간 참조 픽처(inter-view reference picture) 내의 (xRef, yRef) 위치에서의 휘도 예측 블록은 현재 액세스 유닛에서 참조 뷰 인덱스의 값과 같은 뷰 인덱스를 가지는 픽처 내 블록으로 설정된다. 이때, 시점 간 참조 픽처 내의 (xRef, yRef) 위치에서의 휘도 예측 블록은 'ivRefPic'로 정의되며, 액세스 유닛은 영상이부/복호화 되는 단위를 의미한다. 액세스 유닛은 동일한 출력 순서(Picture Order Count: POC)를 가지는 서로 다른 시점의 영상들을 포함한다. 예를 들어 시점이 3개 라면, 하나의 액세스 유닛에는 첫 번째 시점의 일반 영상 및/또는 깊이 정보 영상, 두 번째 시점의 일반 영상 및/또는 깊이 정보 영상, 세 번째 시점의 일반 영상 및/또는 깊이 정보 영상이 포함될 수 있다. 참조 뷰 인덱스는 'refViewIdx'로 정의될 수 있고, 뷰 인덱스는 'ViewIdx'로 정의될 수 있다. 이때, ViewIdx는 현재 픽처의 시점을 의미할 수도 있다.

[0242] 이때, 초기화를 위해 현재 블록의 서브 예측 유닛을 식별하는 정보는 0으로 설정되며, 현재 블록의 서브 예측 유닛을 식별하는 정보는 'curSubBlockIdx'로 정의될 수 있다. 저장 공간에 저장된 움직임 정보의 사용 가능 여부를 식별하는 정보 또한 0으로 설정되어 초기화되며, 저장 공간에 저장된 움직임 정보의 사용 가능 여부를

식별하는 정보는 'lastAvalableFlag'로 설정될 수 있다.

[0244] 상술한 변수들의 초기화를 수행한 후, 인터 예측부에서는 0부터 (nPbH / nSbH - 1)의 범위를 가지는 yBlk와, 0부터 (nPbW/nSbW - 1)의 범위를 가지는 xBlk에 대하여는 아래와 같은 과정을 수행한다. 이때, xBlk는 블록의 x 좌표를 의미하며, yBlk는 블록의 y좌표를 의미한다.

[0245] 우선, 인터 예측부는 참조 블록의 서브 예측 유닛으로부터 움직임 정보의 예측 여부를 식별하는 정보, 서브 예측 유닛 예측 플래그, 서브 예측 유닛의 움직임 벡터, 및 서브 예측 유닛의 참조 인덱스를 초기화 한다. 구체적으로, 참조 블록의 서브 예측 유닛으로부터 움직임 정보의 예측 여부를 식별하는 정보는 0으로 설정되며, 이때, 참조 블록의 서브 예측 유닛으로부터 움직임 정보의 예측 여부를 식별하는 정보는 'curAvailableFlag'로 정의될 수 있다. 또한, 서브 예측 유닛 예측 플래그는 0으로 설정되며, 이때, 서브 예측 유닛 예측 플래그는 'spPredFlagL1'로 정의될 수 있고, 블록의 좌표를 표현하기 위해 서브 예측 유닛 플래그는 'spPredFlagL1[xBlk][yBlk]'로 정의될 수도 있다. 서브 예측 유닛의 움직임 벡터는 (0, 0)으로 설정되며, 서브 예측 유닛의 움직임 벡터는 'spMvLX'로 정의될 수 있다. 서브 예측 유닛의 참조 인덱스는 -1로 설정되며, 이때, 서브 예측 유닛의 참조 인덱스는 'spRefIdxLX'로 정의될 수 있고, 블록의 좌표를 표현하기 위해 서브 예측 유닛의 참조 인덱스는 'spRefIdxLX [xBlk][yBlk]'로 정의될 수도 있다.

[0246] 참조 블록의 위치 (xRef, yRef)는 아래 수학식 3과 같이 설정될 수 있다.

[0247] <수학식 3>

$$xRef = Clip3(0, PicWidthInSamplesL-1, xPb+xBlk*nSbW+nSbW/2+((mvDisp[0]+2)>>2)))$$

$$yRef = Clip3(0, PicHeightInSamplesL-1, yPb+yBlk*nSbH+nSbH/2+((mvDisp[1]+2)>>2)))$$

[0248]

[0249] 이때, xRef는 참조 블록의 위치에 대한 x 좌표를 의미하며, yRef는 참조 블록의 위치에 대한 y 좌표를 의미한다. PicWidthInSamplesL은 현재 픽처에서 너비를 의미하며, PicHeightInSamplesL는 현재 픽처에서 높이를 의미한다. 또한, Clip3()은 아래 수학식 4와 같이 정의될 수 있다.

[0250] <수학식 4>

$$Clip3(x,y,z) = \begin{cases} x; z < x \\ y; z > y \\ z; otherwise \end{cases}$$

[0251]

[0253] 시점 간 참조 블록이 인트라 모드로 부호화 되지 않았을 경우, 0부터 1의 범위를 가지는 X에 대해 아래의 과정을 수행한다. 시점 간 참조 블록은 ivRefPic이 가리키는 시점 간 참조 픽처 내의 (xRef, yRef) 위치에서의 휘도 예측 블록을 나타내며, 시점 간 참조 블록은 'ivRefPb'로 정의될 수 있다. 즉, 변수 ivRefPb는 ivRefPic이 가리키는 시점 간 참조 픽처 내의 (xRef, yRef) 위치에서의 휘도 예측 블록을 나타내며, ivRefPic는 시점 간 참조 픽처를 의미한다. ivRefPb가 가리키는 참조 블록의 좌상단 위치는 (xIvRefPb, yIvRefPb)로 설정될 수 있다.

[0254] 우선, X가 0이거나 현재 슬라이스가 B 슬라이스 일 때에는, Y(Y는 X에서 (1 - X)의 범위를 가진다.)에 대해 다음과 같이 각각의 변수를 재설정한다. refPicListLYIvRef는 ivRefPic가 가리키는 픽처 내의 RefPicListLY로 설정되며, 이때, RefPicListLY는 참조 픽처 리스트를 의미한다. predFlagLYIvRef[x][y]는 ivRefPic가 가리키는 픽처 내의 PredFlagLY[x][y]로 설정되며, 이때, PredFlagLY는 참조 픽처 리스트를 지시하는 식별자를 의미한다. mvLYIvRef[x][y]는 ivRefPic가 가리키는 픽처 내의 MvLY[x][y]로 설정되며, 이때, MvLY는 움직임 벡터를 의미한다. 마찬가지로, refIdxLYIvRef[x][y]는 ivRefPic가 가리키는 픽처 내의 RefIdxLY[x][y]로 설정되며, 이때, RefIdxLY는 참조 인덱스를 의미한다.

[0255] 이때, predFlagLYIvRef[xIvRefPb][yIvRefPb]가 1의 값을 가지는 경우, 0부터 num_ref_idx_lX_active_minus1(참조 픽처 리스트 내에 존재하는 참조 픽처 개수)의 범위를 가지는 i에 대해서는

아래 수학식 5가 적용될 수 있다.

[0256] <수학식 5>

$$spMvLX[xBlk][yBlk] = mvLYIvRef[xIvRefPb][yIvRefPb]$$

$$spRefIdxLX[xBlk][yBlk] = i$$

$$spPredFlagLX[xBlk][yBlk] = 1$$

$$curAvailableFlag = 1$$

[0257]

[0258] 한편, 표 1을 참조하면, curAvailableFlag가 1일 경우와, 0일 경우를 나누어 각각 아래와 같은 과정을 적용한다.

[0259] 우선, curAvailableFlag가 1이라면, 인터 예측부에서는 아래와 같은 과정을 수행한다.

[0260] 1. lastAvailableFlag가 0이라면, 0부터 1의 범위를 가지는 X에 대하여 아래 수학식 6이 적용될 수 있다.

[0261] <수학식 6>

$$mxLXInterView = spMvLX[xBlk][yBlk]$$

$$refIdxLXInterView = spRefIdxLX[xBlk][yBlk]$$

$$availableFlagLXInterview = spPredFlag[xBlk][yBlk]$$

[0262]

[0263] lastAvailableFlag가 0이고, curSubBlockIdx가 0보다 클 때, 0부터 (curSubBlockIdx - 1)의 범위를 가지는 k에 대하여, 변수 i와 j는 아래 수학식 7이 적용될 수 있다.

[0264] <수학식 7>

$$i = k\%(nPSW/nSbW)$$

$$j = k/(nPSW/nSbW)$$

[0265]

[0266] 이때, 0부터 1의 범위를 가지는 X에 대해서는, 아래 수학식 8이 적용된다.

[0267] <수학식 8>

$$spMvLX[i][j] = spMvLX[xBlk][yBlk]$$

$$spRefIdxLX[i][j] = spRefIdxLX[xBlk][yBlk]$$

$$spPredFlagLX[i][j] = spPredFlagLX[xBlk][yBlk]$$

[0268]

[0269] 2. 상술한 과정 이후, 인터 예측부에서는 변수 lastAvailableFlag에 1을 대입한다.

[0270] 3. 이후, 인터 예측부에서는 변수 xLastAvail과 yLastAvail에 각각 xBlk와 yBlk을 저장한다.

[0272] 만약, curAvailableFlag가 1이고, lastAvailableFlag가 1이라면, 인터 예측부에서는 0부터 1의 범위를 가지는 X에 대해서 아래 수학식 9를 적용한다.

[0274] *<수학식 9>

$$spMvLX[xBlk][yBlk] = spMvLX[xLastAvail][yLastAvail]$$

$$spRefIdxLX[xBlk][yBlk] = spRefIdxLX[xLastAvail][yLastAvail]$$

[0275] $$spPredFlagLX[xBlk][yBlk] = spPredFlagLX[xLastAvail][yLastAvail]$$

[0276] 마지막으로, 상술한 과정을 전부 수행한 경우, 변수 curSubBlockIdx는 curSubBlockIdx + 1로 설정된다.

[0278] 도 17에 따른 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 방법은 참조 블록 내의 서브 예측 유닛에서 움직임 정보를 유도할 수 없을 때, 이전(또는 이후)에 참조 했었던 참조 블록의 서브 예측 유닛에 대한 움직임 정보를 사용한다. 따라서, 도 17에 따른 움직임 정보 유도 방법에서는 다른 참조 블록의 서브 예측 유닛을 반드시 참조해야 하므로, 도 17에 따른 움직임 정보 유도 방법은 의존성을 가지게 된다. 의존성을 가지는 움직임 정보 유도 방법은 병렬적인 설계에 취약한 구조를 가지게 되며, 의존성을 가지는 움직임 정보 유도 방법이 병렬적인 설계에 취약하다는 점은 도 18을 통해 구체적으로 설명한다.

[0280] 도 18은 참조 블록의 서브 예측 유닛을 이용하여 현재 블록의 서브 예측 유닛의 정보를 병렬적으로 유도하는 과정의 일례를 나타내는 도면이다.

[0281] 도 18을 참조하면, Ref는 참조 블록을 의미하며, 이때 Ref 0, 1, 2, 3, 4, 5, 6, 7은 각각 참조 블록의 서브 예측 유닛 0, 1, 2, 3, 4, 5, 6, 7을 의미한다. Cur은 현재 블록을 의미하며, 이때, Cur 0, 1, 2, 3, 4, 5, 6, 7은 각각 현재 블록의 서브 예측 유닛 0, 1, 2, 3, 4, 5, 6, 7을 의미한다. Ref 2, 3, 4, 5의 X는 참조 블록의 서브 예측 유닛 2, 3, 4, 5 각각은 움직임 정보 유도 시 사용될 수 없다는 것을 의미한다.

[0282] 도 18에 따른 실시 일례로, 인터 예측부에서는 움직임 정보 유도가 불가능한 서브 예측 유닛으로부터 움직임 정보를 유도하기 위하여, 상술한 바와 같이 움직임 정보 유도가 가능한 서브 예측 유닛을 검출한다. 따라서, 인터 예측부에서는 현재 블록에 대한 각각의 서브 예측 유닛별로, 독립적으로 움직임 정보를 유도하는 것이 불가능하며, 상술한 과정에 따른 움직임 정보 유도는 병렬화가 어려운 문제점이 존재한다.

[0284] 도 19는 참조 블록 내에서 사용 가능한 서브 예측 유닛이 우측 최하단에 위치한 경우, 사용 가능한 서브 예측 유닛을 탐색하는 과정의 일례를 나타내는 도면이다.

[0285] 도 19를 참조하면, 각각의 사각형은 서브 예측 유닛을 의미하며, 굵은 실선으로 이루어진 사각형은 움직임 정보 유도 시 사용 가능한 서브 예측 유닛을 의미하고, 가는 실선으로 이루어진 사각형은 움직임 정보 유도 시 사용 불가능한 서브 예측 유닛을 의미한다. 또한, 점선으로 이루어진 화살표는 움직임 정보를 탐색하는 순서를 나타낸다.

[0286] 도 19와 같이 참조 블록 내에서 움직임 정보를 유도할 수 있는 서브 예측 유닛이 오른쪽 최하단에만 위치하고 있을 경우에는, 참조 블록 내의 좌상단부터 점선으로 이루어진 화살표를 따라 차례로 움직임 정보 유도를 할 수 있는 서브 예측 유닛을 탐색해야만 한다. 일반적인 경우에는 어떤 참조 블록의 어느 서브 예측 유닛이 움직임 정보 유도에 사용 가능한지 모르기 때문에, 참조 블록의 첫번째 서브 예측 유닛부터 순차적으로 서브 예측 유닛이 움직임 정보 유도에 사용 가능한지 여부를 탐색하는 것이다.

[0287] 하지만, 도 19에 따른 움직임 정보의 유도는 사용 가능한 서브 예측 유닛을 찾기 위하여, 참조 블록에 존재하는 모든 서브 예측 유닛을 검사하기 때문에, 잦은 메모리의 접근을 가져온다. 이때, 참조 블록의 서브 예측 유닛 중에 움직임 정보를 가지고 있는 서브 예측 유닛이 얼마 없을 경우에는, 불필요한 서브 예측 유닛의 탐색을 수행하게 된다. 특히, 참조 블록의 모든 서브 예측 유닛이 움직임 정보 유도에 사용될 수 없는 경우에는, 사용 가능한 참조 블록의 서브 예측 유닛을 탐색하는 과정은 아무 이득이 없으며, 불필요한 메모리 접근만 가져오는 방법이다. 이때, 움직임 정보가 없다는 것은 현재 블록이 인접 프레임 내의 참조 블록에서 유사한 영역을 찾지 못했다는 것을 의미한다.

[0288] 따라서, 참조 블록 내에 움직임 정보를 가지고 있는 서브 예측 유닛이 얼마 없거나, 움직임 정보를 가지고 있는

서브 예측 유닛이 전혀 없을 경우에는 인트라 예측을 이용하여 현재 블록을 부호화/복호화 하는 것이 더 효율이 좋을 수 있다. 즉, 참조 블록 내에 움직임 정보를 가지고 있는 서브 예측 유닛이 얼마 없거나, 움직임 정보를 가지고 있는 서브 예측 유닛이 전혀 없을 경우에는 현재 블록의 주변 픽셀에서 유사한 영역을 찾아 현재 블록의 부호화/복호화를 수행하는 것이 더 효율적이다.

[0290] 도 20은 서브 예측 유닛 단위로 움직임 정보를 유도할 때 소요되는 시간을 개략적으로 도시한 것이다.

[0291] 도 20에 따르면, 하나의 서브 예측 유닛 내의 움직임 정보를 유도하는데 걸리는 시간이 T이고, 참조 블록 내의 서브 예측 유닛의 개수가 N일때는 참조 블록 내의 모든 움직임 정보를 유도하는데 걸리는 시간이 NxT가 된다. 상술한 움직임 정보 유도 방법에서는 데이터의 의존성과, 잦은 메모리 접근이라는 문제점을 가져온다. 데이터의 의존성을 가지는 움직임 정보 유도 방법에서는 각 서브 예측 유닛의 움직임 정보를 독립적으로 유도할 수 없기 때문에, 하나의 서브 예측 유닛에 대한 움직임 정보를 유도하기 위해서는 다른 서브 예측 유닛에 대한 움직임 정보가 유도 되는 것을 기다려야만 한다. 따라서, 데이터의 의존성을 가지는 움직임 정보 유도 방법에서는 부호화/복호화 지연 시간이 발생하게 된다.

[0292] 결과적으로, 상술한 움직임 정보 유도 방법에서는 동시에 움직임 정보를 유도하는 데이터 병렬화의 효과를 볼 수 없으며, 상술한 움직임 정보 유도 방법의 설계 구조상 발생하는 잦은 메모리 접근으로 인하여, 메모리 사용 효율이 낮아지는 문제점이 생기게 된다.

[0294] 본 발명에서는 상술한 문제점을 해결하기 위해 움직임 정보 유도시 의존성을 제거하는 장치 및 방법을 제안한다. 도 21에서는 본 발명이 적용되는 인터 예측부의 구조의 일례를 나타내고, 도 22 내지 도 26을 통해 본 발명이 적용되는 움직임 정보 유도 방법의 실시예를 구체적으로 설명한다.

[0296] 도 21은 본 발명이 적용되는 인터 예측부(2100)의 구조를 나타내는 블록도이다.

[0297] 도 21에 따르면, 인터 예측부(2100)는 저장부(2110) 및 유도부(2120)를 포함할 수 있다. 이때, 인터 예측부(2100)는 상술한 3차원 비디오 인코더에서의 인터 예측부(710) 또는 3차원 비디오 디코더에서의 인터 예측부(850)를 의미할 수 있다. 또한, 도 21의 인터 예측부(2100)는 상술한 3차원 비디오 부호화/복호화 과정에 적용될 수도 있다.

[0298] 저장부(2110)는 임의의 움직임 정보를 지정하여, 임의의 움직임 정보를 저장 공간에 저장한다. 저장부(2110)에 저장되는 임의의 움직임 정보를 구하기 위해, 참조 블록의 임의의 위치에 존재하는 움직임 정보를 사용할 수 있다. 이때, 임의의 위치는 참조 블록의 중앙을 의미할 수 있으며, 참조 블록의 중앙을 포함하는 (서브) 예측 유닛을 의미할 수도 있다. 또한, 저장부(2110)에 저장된 임의의 움직임 정보는 임의의 초기 값으로 설정될 수도 있다. 만약, 임의의 움직임 정보를 저장 공간에 저장할 수 없는 경우에는 서브 예측 유닛 단위로 움직임 정보를 유도하는 과정을 생략할 수 있다. 서브 예측 유닛 단위로 움직임 유도를 생략하는 경우에는 상술한 바와 같이 인트라 예측을 수행할 수도 있다. 저장부(2110)의 구체적인 실시예는 후술하도록 한다.

[0299] 유도부(2120)는 현재 블록의 서브 예측 유닛에서 움직임 정보를 유도하는 과정을 수행한다. 이때, 유도부(2120)에서는 기본적으로 앞서 설명한 움직임 유도 과정이 수행될 수 있다. 다만, 본 발명에서 제안하는 유도부(2120)에서는, 현재 블록의 첫번째 서브 예측 유닛에 대응되는 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하지 않을 경우, 움직임 정보가 존재하는 참조 블록의 서브 예측 유닛까지 검색을 수행한 후, 움직임 정보가 존재하는 참조 블록의 서브 예측 유닛으로부터 현재 블록의 첫 번째 서브 예측 블록의 움직임 정보를 유도하는 대신에, 저장부에 저장된 움직임 정보로부터 현재 블록의 첫 번째 서브 예측 블록의 움직임 정보를 유도할 수 있다. 유도부(2120)의 구체적인 실시예는 후술하도록 한다.

[0301] 이하, 본 발명의 구체적인 실시예들을 도면과 함께 구체적으로 설명하도록 한다.

[0303] 실시예 1

[0304] 도 22는 본 발명의 일 실시예에 따른, 참조 블록을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 방법을 개략적으로 설명하는 순서도이다.

[0305] 실시예 1에서는 참조 블록의 중앙 위치에 대한 움직임 정보에 기반하여, 현재 블록의 서브 예측 유닛(현재 서브 유닛)에 대한 움직임 정보를 유도한다. 실시예 1은 인코더 및 디코더에서 수행되거나 인코더 및 디코더의 예측부 또는 인터 예측부에서 수행될 수 있다. 여기서는 설명의 편의를 위해, 도 21의 인터 예측부(2100)가 실시예 1의 동작을 수행하는 것으로 설명한다.

[0306] 도 22를 참조하면, 인터 예측부(2100)는 참조 블록 내 중앙 위치를 유도할 수 있다(S2200). 참조 블록 내 중앙 위치는 수학식 10과 같이 유도될 수 있다. 이때, 참조 블록은 참조 픽처 내에서 현재 블록과 동일한 위치의 블록으로서, 현재 블록과 동일한 크기를 갖는 블록일 수 있다.

[0307] <수학식 10>

[0308]
$$X\text{의 위치} = xPb + (nPbW \gg 1)$$

$$Y\text{의 위치} = yPb + (nPbH \gg 1)$$

[0309] 이때, xPb, yPb 는 현재 PU의 좌상단 위치를 나타내며, nPbW는 현재 PU의 너비, nPbH는 현재 PU의 높이를 의미한다.

[0310] 인터 예측부(2100)는 참조 블록 내 중앙 위치에 움직임 정보가 존재하는지 여부를 판단할 수 있다 (S2210). 참조 블록 내 중앙 위치는 상술한 바와 같이 특정될 수 있다.

[0311] 참조 블록 내 중앙 위치에 이용 가능한(available) 움직임 정보가 존재하지 않으면, 인터 예측부(2100)는 움직임 정보 유도 과정을 종료할 수 있다. 예컨대, 참조 블록의 중앙에 이용 가능한 움직임 정보가 없다면, 인터 예측부(2100)는 현재 블록에 대한 움직임 정보를 유도하지 않을 수도 있다.

[0312] 참조 블록 내 중앙 위치에 움직임 정보가 존재하면, 인터 예측부(2100)는 참조 블록의 중앙 위치에 존재하는 움직임 정보를 저장 공간에 저장할 수 있다(S2220). 참조 블록의 중앙에 위치하는 움직임 정보는, 참조 블록의 중앙에 가장 인접한 풀샘플(full sample) 위치를 포함하는 예측 블록에 대한 움직임 정보일 수 있다. 인터 예측부(2100)에서의 움직임 정보에 대한 구체적인 저장 과정은 상술한 바와 같다. 인터 예측부(2100)는 저장된 참조 블록의 움직임 정보를 기반으로 현재 서브 예측 유닛의 움직임 정보를 유도할 수 있다.

[0313] 인터 예측부(2100)는 현재 서브 예측 유닛에 대응되는 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하는지 여부를 판단할 수 있다(S2240).

[0314] 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하는 경우, 인터 예측부(2100)는 현재 서브 예측 유닛에, 참조 블록의 서브 예측 유닛에 대한 움직임 정보를 삽입할 수 있다(S2250). 즉, 인터 예측부(2100)는 참조 블록의 서브 예측 블록에 대한 움직임 정보(예컨대, 움직임 벡터, 참조 픽처 인덱스)를 대응하는 현재 서브 예측 유닛에 대한 움직임 정보로 설정할 수 있다.

[0315] 참조 블록의 서브 예측 유닛에 이용 가능한 움직임 정보가 존재하지 않는 경우, 인터 예측부(2100)는 저장 공간에 저장된 참조 블록의 움직임 정보를 현재 서브 예측 유닛에 삽입한다(S2260). 즉, 현재 서브 예측 블록에 대응하는 참조 블록 내 서브 예측 블록의 움직임 정보가 이용 가능하지 않은 경우에, 인터 예측부(2100)는 S2200 단계에서 저장한 참조 블록의 중앙에 대한 움직임 정보를 현재 서브 예측 블록에 대한 움직임 정보로 설정할 수 있다.

[0317] *인터 예측부(2100)는 현재 서브 예측 유닛에 대응하는 참보 블록 내 서브 예측 블록이 참조 블록 내 마지막 서브 예측 유닛인지 (혹은 동일한 의미로 현재 서브 예측 블록이 현재 블록 내의 마지막 서브 예측 블록인지) 여부에 대해 판단할 수 있다(S2270). 인터 예측부(2100)는 참조 블록의 서브 예측 유닛이 마지막 서브 예측 유닛인 경우에는 움직임 정보 유도 과정을 종료할 수 있다.

[0318] 참조 블록의 서브 예측 유닛이 마지막 서브 예측 유닛이 아니면, 인터 예측부(2100)는, 움직임 정보의 유도를 계속 수행하기 위하여, 현재 블록의 다음 서브 예측 유닛에 대하여 움직임 정보의 유도를 진행한다(S2230).

[0320] 상술한 실시예 1의 움직임 정보 유도 과정을 3D 비디오의 디코딩에 적용하면 아래 표 2와 같다.

[0321] <표 2>

[0322]

이 프로세스의 입력은 다음과 같다. - 현재 예측 유닛의 좌상단 위치 (xPb, yPb) - 현재 예측 유닛의 너비(nPbW)와 높이(nPbH) - 참조 시점 인덱스 refViewIdx - 변이 벡터 mvDisp 프로세스의 출력은 다음과 같다. - 시간적 시점 간 움직임 후보가 사용 가능한지의 여부를 판단하는 플래그 availableFlagLXInterView, 이 때 LX는 참조 픽처 리스트 L0와 L1이 될 수 있다. - 시간적 시점 간 움직임 벡터 후보 mvLXInterView, 이 때 LX는 참조 픽처 리스트 L0와 L1이 될 수 있다. - 참조 픽처 리스트 RefPicListLX에 존재하는 참조 픽처를 지정하는 참조 인덱스 refIdxLXInterView. 이 때 LX는 참조 픽처 리스트 L0와 L1이 될 수 있다. LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. LX에 대해 다음을 적용한다. - 플래그(flag)인 availableFlagLXInterView는 0으로 초기화한다. - 움직임 벡터 mvLXInterView는 (0, 0)으로 초기화한다. - 참조 인덱스 refIdxLXInterView는 -1로 초기화한다. 변수 nSbW와 nSbH는 다음과 같이 초기화한다.

- nSbW = Min(nPbW, SubPbSize)

- nSbH = Min(nPbH, SubPbSize)

이 때, SubPbSize는 VPS에서 지정한 서브 예측 단위의 높이, 너비 크기이다.

변수 ivRefPic은 현재 액세스 유닛에서 refViewIdx의 값과 같은 ViewIdx를 가지는 픽처로 초기화된다.

변수 curSubBlockIdx는 0으로 초기화한다.

참조 위치 (xRef, yRef)는 다음과 같이 유도될 수 있다.

xRefFull = xPb + (nPbW >> 1) + ((mvDisp[0] + 2) >> 2)

yRefFull = yPb + (nPbH >> 1) + ((mvDisp[1] + 2) >> 2)

xRef = Clip3(0, PicWidthInSamplesL - 1, (xRefFull >> 3) << 3)

yRef = Clip3(0, PicHeightInSamplesL - 1, (yRefFull >> 3) << 3)

ivRefPic은 현재 액세스 유닛에서 refViewIdx와 같은 ViewIdx를 가지는 픽처로 설정된다. 참조 픽처에서 움직임 정보는 8x8 픽셀 블록 단위로 저장될 수 있다. 보정 팩터 xRefFull와 yRefFull는 mvDisp를 이용하여 특정되는 참조 블록 중앙의 풀 샘플에 대한 위치일 수 있다.

ivRefPb는 ivRefPic에서 (xRef, yRef) 위치를 포함(cover)하는 예측 블록일 수 있다.

(xIvRefPb, yIvRefPb)는 ivRefPb 의 좌상단 위치를 특정한다.

[0323]

만약 ivRefPb가 인트라모드로 부호화되지 않았다면, X부터 (1 - X)의 범위를 가지는 Y에 대하여 다음을 적용할 수 있다.

- refPicListLYIvRef, predFlagLYIvRef[x][y], mvLYIvRef[x][y], refIdxLYIvRef[x][y]는 각각 시점 간 참조 픽처 ivRefPic 내에서 그에 상응하는 변수, 즉, RefPicListLY, PredFlag[x][y], MvLY[x][y], RefIdxLY[x][y]로 설정된다.
- 만약 predFlatLYIvRef[xIvRefPb][yIvRefPb]가 1과 같다면, 0부터 num_ref_idx_lX_active_minus1(참조 픽처 리스트 X에 존재하는 참조 픽처 개수)의 범위를 가지는 i에 대하여 다음을 적용할 수 있다
  - 만약 refPicListLYIvRef[refIdxLYIvRef[xIvRefPb][yIvRefPb]]의 POC(Picture Order Count: 픽처의 출력 순서)가 RefPicListLX[i]와 같고 availableFlagLXInterView가 0이라면 다음을 적용한다.

    availableFlagLXInterView = 1

    mvLXInterView = mvLYIvRef[xIvRefPb][yIvRefPb]

    refIdxLX = i

만약 availableFlagL0InterView, 또는 availableFlagL1InterView가 1이라면 다음을 수행한다

- 0부터 (nPbH / nSbH - 1)의 범위를 가지는 yBlk와, 0부터 (nPbW/nSbW - 1)의 범위를 가지는 xBlk에 대하여 다음을 적용한다.

[0324]

[0325]

- 변수 curAvailableFlag 는 0으로 초기화한다.
- 0부터 1의 범위를 가지는 X에 대해서 다음을 적용한다.
  - 플래그 spPredFlagL1[xBlk][yBlk]는 0으로 초기화한다.
  - 움직임 벡터 spMvLX는 (0, 0)으로 초기화한다.
  - 참조 인덱스 spRefIdxLX[xBlk][yBlk]는 -1로 초기화한다.
- 참조 블록 위치 (xRef, yRef)는 다음과 같이 유도된다.

  xRef = Clip3(0, PicWidthInSamplesL - 1,

  xPb + xBlk * nSbW + nSbW / 2 + ((mvDisp[0] + 2) >> 2))

  yRef = Clip3(0, PicHeightInSamplesL - 1,

  yPb + yBlk * nSbH + nSbH / 2 + ((mvDisp[1] + 2) >> 2))
- 변수 ivRefPb는 ivRefPic이 가리키는 시점 간 참조 픽처 내의 (xRef, yRef) 위치에서의 휘도 예측 블록을 나타낸다.
- (xIvRefPb, yIvRefPb)는 ivRefPb가 가리키는 참조 블록의 좌상단 위치를 나타낸다.
- ivRefPb가 인트라 모드로 부호화 되지 않았다면 0부터 1의 범위를 가지는 X에 대해서 다음을 수행한다.
  - X가 0이거나 현재 슬라이스가 B 슬라이스 일 때 X에서 (1 - X)의 범위를 가지는 Y에 대해서 다음을 수행한다.
    - refPixListLYIvRef, predFlagLYIvRef[x][y], mvLYIvRef[x][y], refIdxLYIvRef[x][y]는 각각 ivRefPic가 가리키는 픽처 내의 RefPicListLY, PredFlagLY[x][y],

```
        MvLY[ x ][ y ], RefIdxLY[ x ][ y ]로 세팅된다.
        - predFlagLYIvRef[ xIvRefPb ][ yIvRefPb ]가 1이면, 0부터
        num_ref_idx_lX_active_minus1(참조 픽처 리스트 내 참조 픽처 개수)의
        범위를 가지는 i에 대해서 다음을 수행한다.
          - refPicListLYIvRef[ refIdxLYIvRef[ xIvRefPb ][ yIvRefPb ] ]의
          POC가 RefPicListLX[ i ] 와 같고 spPredFlagLX[ xBlk ][ yBlk ]가
          0이라면, 다음을 적용한다.
            spMvLX[ xBlk ][ yBlk ] = mvLYIvRef[ xIvRefPb ][ yIvRefPb ]
            spRefIdxLX[ xBlk ][ yBlk ] = i
            spPredFlagLX[ xBlk ][ yBlk ] = 1
            curAvailableFlag = 1
      - curAvailableFlag에 따라 다음을 적용한다.
        - curAvailableFlag가 0이면, 0부터 1의 범위를 가지는 X에 대해서
        다음을 적용한다.
          spMvLX[ xBlk ][ yBlk ] = mvLXInterView
          spRefIdxLX[ xBlk ][ yBlk ] = refIdxLX
          spPredFlagLX[ xBlk ][ yBlk ] = availableFlagLXInterView
    - 변수 curSubBlockIdx는 curSubBlockIdx + 1로 세팅된다.

만약 availableFlagL0InterView와 availableFlagL1InterView가 0이라면 이
프로세스는 종료된다.
```

[0326]

[0328] 표 2에 기반해서, 실시예 1을 다시 설명하면 아래와 같다.

[0329] 표 2를 참조하면, 인터 예측부(2100)에는 현재 예측 블록의 좌상단의 위치, 현재 예측 블록의 너비와 현재 예측 블록의 높이, 참조 시점 인덱스, 변이 벡터가 입력된다. 이때, 현재 예측 블록의 좌상단의 위치는 (xPb, yPb)로 정의될 수 있다. 현재 예측 블록의 너비는 'nPbW'로 정의될 수 있으며, 현재 예측 블록의 높이는 'nPbH'로 정의될 수 있다. 참조 시점 인덱스는 'refViewIdx'로 정의될 수 있으며, 변이 벡터는 'mvDisp'로 표시될 수 있다.

[0330] 참조 블록의 서브 예측 유닛을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 과정을 마친 경우, 인터 예측부(2100)에서는 시점 간 예측이 가능한지의 여부를 판단하는 플래그와, 시점 간 움직임 벡터, 참조 픽처 리스트에 존재하는 참조 픽처를 지정하는 참조 인덱스를 출력할 수 있다. 이때, 시점 간 움직임 후보가 사용 가능한지의 여부를 판단하는 플래그는 'availableFlagLXInterView'로 정의될 수 있으며, 시간적 시점 간 움직임 벡터 후보는 'mvLXInterView'로 정의될 수 있다. 또한, 참조 픽처 리스트는 'RefPicListLX'로 정의될 수 있으며, 참조 픽처 리스트에 존재하는 참조 픽처를 지정하는 참조 인덱스는 'refIdxLXInterView'로 표시될 수 있다. 'availableFlagLXInterView', 'mvLXInterView', 'RefPicListLX', 및 'refIdxLXInterView'에 존재하는 'LX'는 참조 픽처 리스트 0(List 0, L0) 또는 참조 픽처 리스트 1(List 1, L1)이 될 수 있다.

[0331] 이하, 상술한 입력으로부터 상술한 출력을 도출하여, 현재 블록의 서브 예측 유닛에 대한 움직임 정보를 유도하는 방법을 구체적으로 설명한다.

[0332] 우선, 인터 예측부(2100)는 참조 블록의 서브 예측 유닛을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하기에 앞서, 초기화를 수행한다. 이때, availableFlagLXInterView는 0으로 설정되고, mvLXInterView는 (0,0)으로 설정되며, refIdxLXInterView는 -1로 설정될 수 있다. 또한, 인터 예측부(2100)에서 초기화를 수행할 때, 서브 예측 유닛의 너비와 서브 예측 유닛의 높이 또한 초기화 될 수 있다. 이때, 서브 예측 유닛의 너비는 'nSbW'로 표시될 수 있으며, 서브 예측 유닛의 높이는 'nSbH'로 표시될 수 있다. 수학식 11은 변수 nSbW와 nSbH의 초기화 방법에 대한 일 예이다.

[0333] <수학식 11>

$$nSbW = Min(nPbW, SubPbSize[\ nuh_layer_id\ ])$$

$$nSbH = Min(nPbH, SubPbSize[\ nuh_layer_id\ ])$$

[0334]

[0335] 이때, SubPbSize는 VPS에서 지정한 서브 예측 유닛의 높이, 너비의 크기를 의미하며, nuh_layer_id는 레이어를 식별(예컨대, 어느 참조 시점인지 여부 등)하는 인덱스를 의미한다. Min()은 입력되는 변수 중 작은 값을 가지는 변수를 출력하는 연산자이다.

[0336] 인터 예측부(2100)에서는 상술한 변수들뿐만 아니라, 시점 간 참조 픽처 내의 (xRef, yRef) 위치에서의 휘도(luma) 예측 블록과 현재 블록의 서브 예측 유닛을 식별하는 정보, 및 저장 공간에 저장된 움직임 정보의 사용 가능 여부를 식별하는 정보 또한 초기화할 수 있다.

[0337] 이때, 시점 간 참조 픽처는 현재 액세스 유닛에서 참조 뷰 인덱스의 값과 같은 뷰 인덱스를 가지는 픽처로 설정될 수 있다. 이때, 시점 간 참조 픽처는 'ivRefPic'라 표시하고, 시점 간 참조 픽처 내의 (xRef, yRef) 위치에서의 휘도 예측 블록은 'ivRefPb'라고 표시할 수 있다. 하나의 액세스 유닛은 동일한 출력 순서(Picture Order Count: POC)를 가지는 서로 다른 시점의 영상들을 포함한다. 참조 시점 인덱스는 'refViewIdx'로 정의될 수 있고, 시점 인덱스는 'ViewIdx'로 정의될 수 있다.

[0339] 참조 위치는 실시예 1에 따른 참조 블록의 중앙을 포함(cover)하는 예측 블록을 특정하는 위치일 수 있다. 이때의 참조 위치에 대한 움직임 정보가 현재 서브 예측 유닛에 대한 움직임 정보를 유도하기 위해 저장될 수 있다. 수학식 12는 참조 위치(xRef, yRef)를 유도하는 방법의 일 예이다.

[0340] <수학식 12>

$$xRefFull = xPb + (nPbW \gg 1) + ((mvDisp[0] + 2) \gg 2)$$

$$yRefFull = yPb + (nPbH \gg 1) + ((mvDisp[1] + 2) \gg 2)$$

$$xRef = Clip3(0, PicWidthInSamplesL\ -\ 1, (xRefFull \gg 3) \ll 3)$$

$$yRef = Clip3(0, PicHeightInSamplesL\ -\ 1, (yRefFull \gg 3) \ll 3)$$

[0341]

[0342] 이때, xRefFull과 yRefFull은 참조 블록의 중앙에 근접한 풀 샘플의 위치를 나타낸다. 즉, xRefFull과 yRefFull는 정수 위치의 샘플에 대한 x좌표와 y좌표를 나타낸다.

[0344] ivRefPb는 (xRef, yRef) 를 포함(cover)하는 예측 블록 또는 서브 예측 유닛일 수 있다. 휘도 샘플의 위치(xIvRefPb, yIvRefPb)가 ivRefPb의 좌상단을 특정할 수 있다.

[0346] 만약 ivRefPb가 인트라 모드로 부호화/복호화되지 않았다면, X부터 (1 - X)의 범위를 가지는 Y에 대하여 아래 (1)과 (2)의 과정을 적용할 수 있다.

[0347] (1) refPicListLYIvRef는 시점 간 참조 픽처 ivRefPic 내의 RefPicListLY로 설정되며, predFlagLYIvRef[x][y]는 시점 간 참조 픽처 ivRefPic 내의 PredFlag[x][y]로 설정되고, refIdxLYIvRef[x][y]는 시점 간 참조 픽처 ivRefPic 내의 RefIdxLY[x][y]로 설정될 수 있다.

[0349] *(2) 만약 predFlagLYIvRef[xIvRefPb][yIvRefPb]의 값이 1이라면, 0부터 num_ref_idx_lX_active_minus1(참조 픽처 리스트 X에 존재하는 참조 픽처 개수)의 범위를 가지는 i에 대해 다음 과정을 적용한다: 만약 refPicListLYIvRef[refIdxLYIvRef[xIvRefPb][yIvRefPb]]의 POC(Picture Order Count: 픽처의 출력 순서)가 RefPicListLX[i]와 같고 availableFlagLXInterView가 0이라면 수학식 13을 적용할 수 있다.

[0350] <수학식 13>

$$availableFlagLXInterView = 1$$

$$mvLXInterView = mvLYIvRef[xIvRefPb][yIvRefPb]$$

$$refIdxLX = i$$

[0351]

[0353] 이어서, availableFlagL0InterView, 또는 availableFlagL1InterView가 1인 경우, 인터 예측부(2100)는 0부터 (nPbH / nSbH - 1)의 범위를 가지는 yBlk와, 0부터 (nPbW/nSbW - 1)의 범위를 가지는 xBlk에 대하여는 아래와 같은 과정을 수행한다. 이때, xBlk는 x좌표를 의미하며, yBlk는 y좌표를 의미한다. 다시 말하면, 참조 블록의 중앙에서 가용한 움직임 정보가 유도된 경우에, 인터 예측부(2100)는 서브 예측 블록 단위로 움직임 정보를 유도할 수 있다.

[0355] *우선, 인터 예측부(2100)는 참조 블록의 서브 예측 유닛으로부터 움직임 정보의 예측 여부를 식별하는 정보, 서브 예측 유닛 예측 플래그, 서브 예측 유닛의 움직임 벡터, 및 서브 예측 유닛의 참조 인덱스를 초기화할 수 있다..

[0356] 이때, 참조 블록의 서브 예측 유닛으로부터 움직임 정보의 예측 여부를 식별하는 정보는 'curAvailableFlag'로, 서브 예측 유닛 예측 플래그는 'spPredFlagLX1'로 서브 예측 유닛 플래그는 'spPredFlagLX[xBlk][yBlk]'로, 서브 예측 유닛의 움직임 벡터는 'spMvLX'로, 서브 예측 유닛의 참조 인덱스는 'spRefIdxLX'로 정의될 수 있고, 서브 예측 유닛의 참조 인덱스는 'spRefIdxLX [xBlk][yBlk]'로 정의될 수 있다.

[0357] 참조 블록의 위치 (xRef, yRef)는 아래 수학식 14와 같이 서브 블록 단위로 재설정된다.

[0358] <수학식 14>

$$xRef = Clip3(0, PicWidthInSamplesL-1, xPb + xBlk*nSbW + nSbW/2 + ((mvDisp[0]+2) >> 2)))$$

$$yRef = Clip3(0, PicHeightInSamplesL-1, yPb + yBlk*nSbH + nSbH/2 + ((mvDisp[1]+2) >> 2)))$$

[0359]

[0360] PicWidthInSamplesL은 현재 픽처의 너비를 의미하며, PicHeightInSamplesL는 현재 픽처의 높이를 의미한다. 또한, Clip3()은 상술한 바와 같다.

[0362] 이후, 시점 간 참조 블록이 인트라 모드로 부호화 되지 않았을 경우, 0부터 1의 범위를 가지는 X에 대해 아래의 과정을 수행한다:

[0364] X가 0이거나 현재 슬라이스가 B 슬라이스 일 때에는, Y(Y는 X에서 (1 - X)의 범위를 가진다.)에 대해 다음과 같이 각각의 변수를 재설정한다. refPicListLYIvRef는 ivRefPic에 의해 특정되는 픽처 (즉, 시점 간 참조 픽

처)에 대한 참조 픽처 리스트 RefPicListLY로 설정될 수 있다. predFlagLYIvRef[x][y]는 PredFlagLY[x][y]로 설정된다. PredFlagLY[x][y]는 ivRefPic가 특정하는 픽처 내 (x, y) 위치에서 적용되는 참조 픽처 리스트를 지시한다. mvLYIvRef[x][y]는 MvLY[x][y]로 설정된다. MvLY[x][y]는 ivRefPic가 특정하는 픽처 내 (x, y)에서의 움직임 벡터를 의미한다. refIdxLYIvRef[x][y]는 RefIdxLY[x][y]로 설정된다. RefIdxLY[x][y]는 ivRefPic가 가리키는 픽처 내 (x,y)에서의 참조 픽처를 지시한다.

[0365] predFlagLYIvRef[xIvRefPb][yIvRefPb]가 1의 값을 가지는 경우, 0부터 num_ref_idx_lX_active_minus1(참조 픽처 리스트 내에 존재하는 참조 픽처 개수)의 범위를 가지는 i에 대해서, refPicListLYIvRef[refIdxLYIvRef[xIvRefPb][yIvRefPb]]의 POC가 RefPicListLX[i] 와 같고 spPredFlagLX[xBlk][yBlk]가 0이라면 아래 수학식 15가 적용될 수 있다.

[0366] <수학식 15>

$$spMvLX[xBlk][yBlk] = mvLYIvRef[xIvRefPb][yIvRefPb]$$

$$spRefIdxLX[xBlk][yBlk] = i$$

$$spPredFlagLX[xBlk][yBlk] = 1$$

$$curAvailableFlag = 1$$

[0367]

[0368] 상술한 과정을 수행한 후에도 설정된 curAvailableFlag가 0일 경우(즉, spRefIdxLX=i가 아니고(예컨대, spRefIdxLx=-1), spPredFlagLX=1이 아닌(예컨대, spPredFlagLX=-1)인 경우)에는 서브 블록 단위로 움직임 정보를 유도할 수 없는 경우라고 할 수 있다. 따라서, 인터 예측부(2100)는 0부터 1의 범위를 가지는 X에 대해서 수학식 16을 적용할 수 있다.

[0369] 즉, 참조 블록의 서브 예측 유닛으로부터 움직임 정보의 유도가 불가능할 경우, 인터 예측부(2100)는 참조 블록의 중앙 위치에 대한 움직임 정보로부터 현재 블록의 서브 예측 유닛에 대한 움직임 정보를 유도할 수 있다.

[0371] <수학식 16>

$$spMvLX[xBlk][yBlk] = mvLXInterView$$

$$spRefIdxLX[xBlk][yBlk] = refIdxLX$$

$$spPredFlagLX[xBlk][yBlk] = availableFlagLXInterView$$

[0372]

[0374] 마지막으로, 상술한 과정을 전부 수행한 경우, 변수 curSubBlockIdx는 curSubBlockIdx + 1로 설정되며, 만약 availableFlagL0InterView와 availableFlagL1InterView가 0이라면, 본 발명의 실시예 1에 따른 움직임 정보 유도 과정은 종료된다.

[0376] 실시예 2

[0377] 도 23은 본 발명의 다른 실시예에 따라서, 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 방법을 개략적으로 설명하는 순서도이다. 도 23의 예에서는, 참조 블록의 임의의 위치에 존재하는 서브 예측 유닛을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보가 유도될 수 있다.

[0378] 실시예 2에서는 참조 블록의 중앙 부분을 포함하는 서브 예측 유닛의 움직임 정보에 기초하여, 현재 블록의 서브 예측 유닛에 대한 움직임 정보를 유도할 수 있다.

[0379] 도 23의 예는 인코더 및 디코더에서 수행될 수도 있고, 인코더 및 디코더의 예측부 또는 도 21의 인터 예측부(2100)에서 수행될 수도 있다. 여기서는 설명의 편의를 위해, 인터 예측부(2100)가 도 23의 각 단계를 수행하는 것으로 설명한다.

[0380] 도 23을 참조하면, 인터 예측부(2100)는 참조 블록 중앙의 서브 예측 유닛(중앙 서브 예측 유닛)의 위치를 도할 수 있다(S2300). 참조 블록 내에 위치한 중앙 서브 예측 유닛은 참조 블록의 중앙 부분에 있는 서브 예측 유닛을 의미하며, 참조 블록의 중앙은 상술한 바와 같다. 수학식 17은 참조 블록 내 중앙 서브 예측 유닛의 위치를 유도하는 일 예이다.

[0381] <수학식 17>

$$\text{중앙 부속 예측 유닛의 } X\text{값} = xPb + (nPbW/nSbW/2)*nSbW + nSbW/2$$

$$\text{중앙 부속 예측 유닛의 } Y\text{값} = yPb + (nPbH/nSbH/2)*nSbH + nSbH/2$$

[0382]

[0383] 이때, (xPb, yPb)는 현재 예측 유닛의 좌상단 위치를 나타내며, nPbW는 현재 예측 유닛의 너비, nPbH는 현재 예측 유닛의 높이를 의미한다.

[0384] 인터 예측부(2100)는 참조 블록 내 중앙 서브 예측 유닛에 움직임 정보가 존재하는지 여부를 판단하며(S2310), 참조 블록 내 중앙 서브 예측 유닛의 위치는 상술한 바와 같다. 만약, 참조 블록 내 중앙 서브 예측 유닛의 위치에서 움직임 정보가 존재하지 않을 경우에는 인터 예측부(2100)는 움직임 정보 유도 과정을 종료할 수도 있다.

[0385] 참조 블록 내 중앙 서브 예측 유닛에 움직임 정보가 존재하는 경우, 인터 예측부(2100)는 중앙 위치에 존재하는 움직임 정보를 저장할 수 있다(S2320). 인터 예측부(2100)가 움직임 정보를 저장하는 구체적인 과정은 상술한 바와 같다.

[0386] 인터 예측부(2100)는 현재 서브 예측 유닛의 움직임 정보 유도를 수행한다. 인터 예측부(2100)는 현재 서브 예측 유닛에 대응되는 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하는지 여부를 판단할 수 있다(S2340).

[0387] 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하는 경우, 인터 예측부(2100)는 참조 블록의 서브 예측 유닛에 존재하는 움직임 정보를 현재 서브 예측 유닛에 삽입할 수 있다(S2350). 만약, 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하지 않는 경우, 인터 예측부(2100)는 S2320에서 저장한 움직임 정보를 현재 서브 예측 유닛에 삽입할 수 있다(S2360).

[0388] 인터 예측부(2100)는 움직임 정보 유도의 대상이 되는 참조 블록의 서브 예측 유닛이, 마지막 서브 예측 유닛인지를 판단할 수 있다(S2370). 참조 블록의 서브 예측 유닛이 마지막 서브 예측 유닛인 경우에, 인터 예측부(2100)는 현재 블록에 대한 움직임 정보 유도 과정을 종료할 수 있다. 만약, 참조 블록의 서브 예측 유닛이 마지막 서브 예측 유닛이 아닌 경우에는, 움직임 정보의 유도를 계속 수행하기 위하여, 현재 블록 내 다음 서브 예측 유닛으로 대상을 이동한다(S2330).

[0390] 상술한 실시예 2의 움직임 정보 유도 과정을 3D 비디오에 적용하면 표 3과 같다.

[0391] <표 3>

이 프로세스의 입력은 다음과 같다. - 현재 예측 유닛의 좌상단 위치 (xPb, yPb) - 현재 예측 유닛의 너비(nPbW)와 높이(nPbH) - 참조 시점 인덱스 refViewIdx - 변이 벡터 mvDisp 프로세스의 출력은 다음과 같다. - 시간적 시점 간 움직임 후보가 사용 가능한지의 여부를 판단하는 플래그 availableFlagLXInterView, 이 때 LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. - 시간적 시점 간 움직임 벡터 후보 mvLXInterView, 이 때 LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. - 참조 픽처 리스트 RefPicListLX에 존재하는 참조 픽처를 지정하는 참조 인덱스 refIdxLXInterView. 이 때 LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. LX에 대해 다음을 적용한다. - 플래그인 availableFlagLXInterView는 0으로 초기화한다 - 움직임 벡터 mvLXInterView는 (0, 0)으로 초기화한다. - 참조 인덱스 refIdxLXInterView는 -1로 초기화한다. 변수 nSbW와 nSbH는 다음과 같이 초기화한다

[0392]

- nSbW = Min(nPbW, SubPbSize)

- nSbH = Min(nPbH, SubPbSize)

이 때, SubPbSize는 VPS에서 지정한 서브 예측 단위의 높이, 너비 크기이다.

변수 ivRefPic은 현재 액세스 유닛에서 refViewIdx의 값과 같은 ViewIdx를 가지는 픽처로 초기화된다.

변수 curSubBlockIdx는 0으로 초기화한다.

참조 위치 (xRef, yRef)는 다음과 같이 유도될 수 있다

xRef = Clip3(0, PicWidthInSamplesL - 1, xPb + (nPbW / nSbW / 2) * nSbW + nSbW/2)

yRef = Clip3(0, PicHeightInSamplesL - 1, yPb + (nPbH / nSbH / 2) * nSbH + nSbH/2)

ivRefPic은 현재 액세스 유닛에서 refViewIdx와 같은 ViewIdx를 가지는 픽처로 설정된다.

ivRefPb는 ivRefPic에서 (xRef, yRef) 위치를 포함하는 예측 블록으로 설정된다. (xIvRefPb, yIvRefPb)는 ivRefPb가 가리키는 참조 블록의 좌상단 위치로 설정된다.

[0393]

[0394]

만약 ivRefPb가 인트라모드로 부호화되지 않았다면, 0부터 1의 범위를 가지는 X에 대하여 다음을 적용한다.

- X가 0이거나 현재 슬라이스 B슬라이스 일 때, X부터 (1 - X)의 범위를 가지는 Y에 대하여 다음을 적용한다
  - refPicListLYIvRef, predFlagLYIvRef[x][y], mvLYIvRef[x][y], refIdxLYIvRef[x][y]는 각각 시점 간 참조 픽처 ivRefPic 내에서 그에 상응하는 변수, 즉, RefPicListLY, PredFlag[x][y], MvLY[x][y], RefIdxLY[x][y]로 설정된다.
  - 만약 predFlatLYIvRef[xIvRefPb][yIvRefPb]가 1과 같다면, 0부터 num_ref_idx_lX_active_minus1(참조 픽처 리스트 X에 존재하는 참조 픽처 개수)의 범위를 가지는 i에 대하여 다음을 적용한다: 만약 refPicListLYIvRef[refIdxLYIvRef[xIvRefPb][yIvRefPb]]의 POC(Picture Order Count: 픽처의 출력 순서)가 RefPicListLX[i]와 RefPicListLX[i]가 같고 centerPredFlag가 0이라면 다음을 적용한다
    - centerAvailableFlag = 1
    - centerMvLX = mvLYIvRef[xIvRefPb][yIvRefPb]
    - centerRefIdxLX = i
    - centerPrdeFlagLX = 1

만약 centerAvailableFlag가 1이라면 다음을 수행한다

- 0부터 (nPbH / nSbH - 1)의 범위를 가지는 yBlk와, 0부터 (nPbW/nSbW - 1)의

[0395]

범위를 가지는 xBlk에 대하여 다음을 적용한다.

- 변수 curAvailableFlag 는 0으로 초기화한다
- 0부터 1의 범위를 가지는 X에 대해서 다음을 적용한다
  - 플래그 spPredFlagL1[xBlk][yBlk]는 0으로 초기화한다.
  - 움직임 벡터 spMvLX는 (0, 0)으로 초기화한다.
  - 참조 인덱스 spRefIdxLX[xBlk][yBlk]는 -1로 초기화한다
- 참조 블록 위치 (xRef, yRef)는 다음과 같이 유도된다

```
xRef = Clip3( 0, PicWidthInSamplesL - 1,
xPb + xBlk * nSbW + nSbW / 2 + ( ( mvDisp[ 0 ] + 2) >> 2 ) )
yRef = Clip3( 0, PicHeightInSamplesL - 1,
yPb + yBlk * nSbH + nSbH / 2 + ( ( mvDisp[ 1 ] + 2) >> 2 ) )
```

- 변수 ivRefPb는 ivRefPic이 가리키는 시점 간 참조 픽처 내의 (xRef, yRef) 위치에서의 휘도 예측 블록을 나타낸다
- (xIvRefPb, yIvRefPb)는 ivRefPb가 가리키는 참조 블록의 좌상단 위치를 나타낸다.
- ivRefPb가 인트라 모드로 부호화 되지 않았다면 0부터 1의 범위를 가지는 X에 대해서 다음을 수행한다.
  - X가 0이거나 현재 슬라이스가 B 슬라이스 일 때 X에서 (1 - X)의 범위를 가지는 Y에 대해서 다음을 수행한다.
    - refPixListLYIvRef, predFlagLYIvRef[x][y], mvLYIvRef[x][y], refIdxLYIvRef[x][y]는 각각 ivRefPic가

가리키는 픽처 내의 RefPicListLY, PredFlagLY[x][y],

MvLY[x][y], RefIdxLY[x][y]로 세팅된다.

- predFlagLYIvRef[xIvRefPb][yIvRefPb]가 1이면, 0부터

num_ref_idx_lX_active_minus1(참조 픽처 리스트 내 참조 픽처 개수)의

범위를 가지는 i에 대해서 다음을 수행한다:

refPicListLYIvRef[refIdxLYIvRef[xIvRefPb][yIvRefPb]]의 POC가

RefPicListLX[i] 와 같고 spPredFlagLX[xBlk][yBlk]가 0이라면,

다음을 적용한다.

spMvLX[xBlk][yBlk] = mvLYIvRef[xIvRefPb][yIvRefPb]

spRefIdxLX[xBlk][yBlk] = i

spPredFlagLX[xBlk][yBlk] = 1

curAvailableFlag = 1

- curAvailableFlag에 따라 다음을 적용한다

- curAvailableFlag가 0이면, 0부터 1의 범위를 가지는 X에 대해서 다음을

적용한다.

spMvLX[xBlk][yBlk] = centerMvLX

spRefIdxLX[xBlk][yBlk] = centerRefIdxLX

spPredFlagLX[xBlk][yBlk] = centerPredFlagLX

- 변수 curSubBlockIdx는 curSubBlockIdx + 1로 세팅된다

그렇지 않다면, 즉, centerAvailableFlag가 0이라면, 이 프로세스는 종료된다.

[0396]

[0398] 표 3에 기반해서 실시예 2를 다시 설명하면 아래와 같다.

[0399] 표 3에서 사용하는 변수들은 표 2에서 설명한 바와 동일하다.

[0400] 인터 예측부(2100)에서는 참조 블록의 서브 예측 유닛을 이용하여 현재 서브 예측 유닛의 움직임 정보를 유도하기에 앞서, 초기화를 수행한다. 초기화의 내용은 표 2에서 설명한 바와 동일하다.

[0402] 인터 예측부는 참조 블록의 중앙 서브 예측 유닛의 위치를 특정할 수 있다. 이때의 참조하는 블록의 위치는 참조 위치를 기반으로 결정될 수 있으며, 참조 위치 (xRef, yRef)는 수학식 18과 같이 유도된다.

[0403] <수학식 18>

$$xRef = Clip3(0, PicWidthInSamplesL - 1, xPb + (nPbW / nSbW / 2) * nSbW + nSbW/2)$$

$$yRef = Clip3(0, PicHeightInSamplesL - 1, yPb + (nPbH / nSbH / 2) * nSbH + nSbH/2)$$

[0404]

[0406] ivRefPic은 현재 액세스 유닛에서 refViewIdx와 같은 ViewIdx를 가지는 픽처이며, ivRefPb는 ivRefPic 내에서 수학식 19를 통해 유도된 (xRef, yRef) 위치를 포함하는 예측 블록 또는 서브 예측 유닛이다.

[0407] (xIvRefPb, yIvRefPb)는 ivRefPb 의 좌상단 위치를 특정한다.

[0409] 만약 ivRefPb가 인트라 모드로 부호화/복호화되지 않았고, X가 0이거나 현재 슬라스가 B 슬라이스 일 경우에는 X부터 (1 - X)의 범위를 가지는 Y에 대하여 아래와 같은 과정을 적용한다.

[0410] 표 2에서 설명한 바와 같이, refPicListLYIvRef는 RefPicListLY로 설정되며, predFlagLYIvRef[x][y]는 PredFlag[x][y]로 설정되고, refIdxLYIvRef[x][y]는 RefIdxLY[x][y]로 설정된다.

[0411] 만약 predFlagLYIvRef[xIvRefPb][yIvRefPb]가 1과 같다면, 0부터 num_ref_idx_lX_active_minus1(참조 픽처 리스트 X에 존재하는 참조 픽처 개수)의 범위를 가지는 i에 대해서, refPicListLYIvRef[refIdxLYIvRef[xIvRefPb][yIvRefPb]]의 POC(Picture Order Count: 픽처의 출력 순서)가 RefPicListLX[i]와 같고 availableFlagLXInterView가 0인 경우에 수학식 19를 적용한다.

[0412] <수학식 19>

$$centerAvailableFlag = 1$$

$$centerMvLX = mvLYIvRef[xIvRefPb][yIvRefPb]$$

$$centerRefIdxLX = i$$

$$centerPredFlagLX = 1$$

[0413]

[0414] 수학식 19에서, centerAvailableFlag는 참조 블록 내의 중앙 서브 예측 유닛에 대한 이용 가능 여부를 나타내며, centerMvLX는 참조 블록 내의 중앙 서브 예측 유닛에 대한 움직임 벡터를 의미한다. 또한, centerRefIdxLX는 참조 블록 내의 중앙 서브 예측 유닛에 대한 참조 인덱스를 나타내고, centerPredFlagLX는 중앙 서브 예측 유닛의 참조 픽처 리스트를 지시한다. 이때, centerAvailableFlag, centerMvLX, centerRefIdxLX, 및/또는 centerPredFlagLX는 중앙 서브 예측 유닛의 움직임 정보를 의미한다. 즉, 인터 예측부(2100)는 수학식 19를 통해 설정한 참조 블록의 중앙 서브 예측 유닛의 움직임 정보를 저장 공간에 저장할 수 있다.

[0416] 상술한 바와 같이 변수들의 값을 설정한 후, centerAvailableFlag가 1인 경우, 인터 예측부(2100)에서는 0부터 (nPbH / nSbH - 1)의 범위를 가지는 yBlk와, 0부터 (nPbW/nSbW - 1)의 범위를 가지는 xBlk에 대하여는 아래와 같은 과정을 수행한다. 이때, xBlk는 블록의 x좌표를 의미하며, yBlk는 블록의 y좌표를 의미한다. 다시 말하면, 참조 블록 중앙의 서브 블록에서 가용한 움직임 정보가 유도된 경우에, 인터 예측부(2100)는 서브 예측 블록 단위로 현재 블록의 움직임 정보를 유도할 수 있다.

[0417] 우선, 인터 예측부(2100)는 참조 블록의 서브 예측 유닛으로부터 움직임 정보의 예측 여부를 식별하는 정보, 서브 예측 유닛 예측 플래그, 서브 예측 유닛의 움직임 벡터, 및 서브 예측 유닛의 참조 인덱스를 초기화 한다. 구체적인 내용은 표 2에서 설명한 바와 동일하다.

[0418] 참조 블록의 위치 (xRef, yRef)는 서브 예측 블록 단위로 수학식 20과 같이 재설정된다.

[0419] <수학식 20>

$$xRef = Clip3(0, PicWidthInSamplesL-1, xPb + xBlk*nSbW + nSbW/2 + ((mvDisp[0]+2) >> 2)))$$

$$yRef = Clip3(0, PicHeightInSamplesL-1, yPb + yBlk*nSbH + nSbH/2 + ((mvDisp[1]+2) >> 2)))$$

[0420]

[0421] 이때, xRef는 참조 블록의 위치에 대한 x좌표를 의미하며, yRef는 참조 블록의 위치에 대한 y좌표를 의미한다.

PicWidthInSamplesL은 현재 픽처의 너비를 의미하며, PicHeightInSamplesL는 현재 픽처의 높이를 의미한다. Clip3()은 상술한 바와 같다.

[0423] 시점 간 참조 블록이 인트라 모드로 부호화 되지 않았을 경우, 인터 예측부(2100)는 0부터 1의 범위를 가지는 X에 대해 아래의 과정을 수행한다.

[0424] 우선, X가 0이거나 현재 슬라이스가 B 슬라이스 일 때에는, Y(Y는 X에서 (1 - X)의 범위를 가진다.)에 대해 다음과 같이 각각의 변수를 재설정한다. 구체적인 내용은 표 2에서 설명한 바와 같다.

[0425] 이때, predFlagLYIvRef[xIvRefPb][yIvRefPb]가 1의 값을 가지는 경우, 0부터 num_ref_idx_lX_active_minus1(참조 픽처 리스트 내에 존재하는 참조 픽처 개수)의 범위를 가지는 i에 대해서, , refPicListLYIvRef[refIdxLYIvRef[xIvRefPb][yIvRefPb]]의 POC가 RefPicListLX[i] 와 같고 spPredFlagLX[xBlk][yBlk]가 0이라면 아래 수학식 21가 적용될 수 있다.

[0426] <수학식 21>

$$spMvLX[xBlk][yBlk] = mvLYIvRef[xIvRefPb][yIvRefPb]$$

$$spRefIdxLX[xBlk][yBlk] = i$$

$$spPredFlagLX[xBlk][yBlk] = 1$$

[0427] $$curAvailableFlag = 1$$

[0429] 상술한 과정을 수행한 후에도 설정된 curAvailableFlag가 0일 경우(즉, spRefIdxLX=i가 아니고(예컨대, spRefIdxLx=-1), spPredFlagLX=1이 아닌(예컨대, spPredFlagLX=-1)인 경우)에는 서브 블록 단위로 움직임 정보를 유도할 수 없는 경우라고 할 수 있다. 따라서, 인터 예측부(2100)는 0부터 1의 범위를 가지는 X에 대해서 수학식 22를 적용할 수 있다.

[0430] 즉, 참조 블록의 서브 예측 유닛으로부터 움직임 정보의 유도가 불가능할 경우, 인터 예측부(2100)는 중앙 서브 유닛의 움직임 정보로부터 현재 블록의 서브 예측 유닛에 대한 움직임 정보를 유도할 수 있다.

[0431] <수학식 22>

$$spMvLX[xBlk][yBlk] = centerMvLX$$

$$spRefIdxLX[xBlk][yBlk] = centerRefIdxLX$$

[0432] $$spPredFlagLX[xBlk][yBlk] = centerPredFlagLX$$

[0434] 마지막으로, 상술한 과정을 전부 수행한 경우, 변수 curSubBlockIdx는 curSubBlockIdx + 1로 설정되며, 만약 availableFlagL0InterView와 availableFlagL1InterView가 0이라면, 본 발명의 실시예 2에 따른 움직임 정보 유도 과정은 종료된다.

[0436] 도 24는 임의 위치의 움직임 정보를 이용하여, 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 과정의 일례를 나타내는 도면이다.

[0437] 도 24를 참조하면, 도 24의 상단에 위치한 블록들은 참조 블록의 서브 예측 유닛들을 의미하며, 도 24의 하단에 위치한 블록들은 현재 블록의 서브 예측 유닛들을 의미한다. 또한, X는 임의의 위치를 의미하며, X의 움직임 정보는 저장 공간에 저장되게 된다. 이때, 도 24의 임의 위치의 움직임 정보는 실시예 1의 참조 블록의 중간 위치의 움직임 정보를 의미할 수 있으며, 도 24의 임의 위치의 움직임 정보는 실시예 2의 참조 블록의 중앙 서브 예측 유닛의 움직임 정보를 의미할 수도 있다.

[0438] 임의 위치의 움직임 정보를 이용하여, 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도할 경우, 참조 블록의 서브 예측 유닛 각각은 임의 위치의 움직임 정보를 이용할 수 있다. 즉, 현재 블록의 복수의 서브 예측 유닛은 동시에 임의 위치의 움직임 정보를 이용하여 움직임 정보의 유도를 할 수 있으며, 임의 위치의 움직임 정보를 이용한 움직임 정보 유도는 데이터 의존성을 극복할 수 있다. 따라서, 임의 위치의 움직임 정보를 이용할 경우, 인터 예측부(2100)는 병렬적인 움직임 정보 유도가 가능하게 된다.

[0439] 상술한 바와 같이, 실시예 1 및 실시예 2는 임의 위치에 존재하는 움직임 정보를 이용하여, 움직임 정보를 유도한다. 따라서, 실시예 1 및 실시예 2에 따른 움직임 정보 유도 방법은 참조 블록 내에 존재하는 각 서브 예측 유닛이 서로 독립적으로 유도될 수 있게 한다. 즉, 실시예 1 및 실시예 2에서는 움직임 정보 유도가 가능한 서브 예측 유닛을 찾기 위하여, 순차적으로 움직임 정보 유도 가능한 서브 예측 유닛을 탐색할 필요가 없으며, 참조 블록의 첫 번째 서브 예측 유닛이 움직임 정보 유도에 사용 불가능한 경우, 실시예 1 및 실시예 2에서는 기설정된 움직임 정보를 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도한다. 이로 인하여, 실시예 1 및 실시예 2를 통한 움직임 정보 유도는 데이터 의존성을 제거하기 때문에, 각 서브 예측 유닛의 움직임 정보를 병렬적으로 유도할 수 있다는 이점이 있다. 또한, 실시예 1 및 실시예 2를 통한 움직임 정보 유도는, 기존의 움직임 정보 유도에 비하여 추가적인 메모리 접근을 막아 메모리 접근 회수를 줄이는 이점을 가져온다.

[0441] 실시예 3

[0442] 도 25는 본 발명의 또 다른 실시예에 따른, 임의의 움직임 정보 값을 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 방법의 순서도이다.

[0443] 도 25를 참조하면, 실시예 3은 초기(Default) 움직임 정보를 설정하고, 참조 블록의 서브 예측 유닛으로부터 움직임 정보 유도가 불가능할 경우, 설정된 초기 움직임 정보로부터 현재 서브 예측 유닛에 대한 움직임 정보를 유도하는 방법을 제공한다. 이때, 설정된 초기 움직임 정보는 영벡터 등을 의미할 수 있다. 실시예 3에 따른 구체적인 움직임 정보 유도 방법은 아래와 같다.

[0444] 인터 예측부(2100)는 초기 움직임 정보를 저장 공간에 저장할 수 있 (S2500). 인터 예측부(2100)가 움직임 정보를 저장하는 구체적인 과정은 상술한 바와 같다.

[0445] 이어서, 인터 예측부(2100)는 현재 서브 예측 유닛의 움직임 정보 유도를 수행할 수 있다. 인터 예측부(2100)는 현재 서브 예측 유닛에 대응되는 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하는지 여부를 판단할 수 있다(S2520).

[0446] 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하는 경우, 인터 예측부(2100)는 현재 서브 예측 유닛에 참조 블록 서브 예측 유닛의 움직임 정보를 삽입할 수 있다(S2530). 참조 블록의 서브 예측 유닛에 움직임 정보가 존재하지 않는 경우, 인터 예측부(2100)는 저장 공간에 저장된 움직임 정보를 현재 서브 예측 유닛에 삽입할 수 있다(S2540).

[0447] 인터 예측부(2100)는 움직임 정보 유도의 대상이 되는 참조 블록의 서브 예측 유닛이, 마지막 서브 예측 유닛인지를 판단할 수 있다(S2550). 참조 블록의 서브 예측 유닛이 마지막 서브 예측 유닛인 경우에, 인터 예측부(2100) 움직임 정보 유도 과정을 종료할 수 있다. 참조 블록의 서브 예측 유닛이 마지막 서브 예측 유닛이 아닌 경우에, 인터 예측부(2100)는 움직임 정보의 유도를 계속 수행하기 위하여, 참조 블록의 다음 서브 예측 유닛의 움직임 정보를 탐색할 수 있다(S2510).

[0449] 상술한 실시예 3의 움직임 정보 유도 과정을 3D-HEVC Draft Text 2에 적용하면 아래 표 4와 같다.

[0450] <표 4>

이 프로세스의 입력은 다음과 같다. - 현재 예측 유닛의 좌상단 위치 (xPb, yPb) - 현재 예측 유닛의 너비(nPbW)와 높이(nPbH) - 참조 시점 인덱스 refViewIdx - 변이 벡터 mvDisp 프로세스의 출력은 다음과 같다. - 시간적 시점 간 움직임 후보가 사용 가능한지의 여부를 판단하는 플래그 availableFlagLXInterView, 이 때 LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. - 시간적 시점 간 움직임 벡터 후보 mvLXInterView, 이 때 LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. - 참조 픽처 리스트 RefPicListLX에 존재하는 참조 픽처를 지정하는 참조 인덱스 refIdxLXInterView. 이 때 LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. LX는 참조 픽처 리스트 L0과 L1가 될 수 있다. LX에 대해 다음을 적용한다. - 플래그인 availableFlagLXInterView는 0으로 초기화한다. - 움직임 벡터 mvLXInterView는 (0, 0)으로 초기화한다. - 참조 인덱스 refIdxLXInterView는 -1로 초기화한다. 변수 nSbW와 nSbH는 다음과 같이 초기화한다.

[0451]

- nSbW = Min(nPbW, SubPbSize)

- nSbH = Min(nPbH, SubPbSize)

이 때, SubPbSize는 VPS에서 지정한 서브 예측 단위의 높이, 너비 크기이다.

변수 ivRefPic은 현재 액세스 유닛에서 refViewIdx의 값과 같은 ViewIdx를 가지는 픽처로 초기화된다.

변수 curSubBlockIdx는 0으로 초기화한다.

변수 availableFlagL0InterView와 availableFlagL1InterView는 다음과 같이 초기화된다.

availableFlagL0Zero= 1

mvL0Zero = (0, 0)

refIdxL0Zero = 0

현재슬라이스가 B슬라이스라면

availableFlagL1Zero = 1

mvL1Zero = (0, 0)

refIdxL1Zero = 0

0부터 (nPbH / nSbH - 1)의 범위를 가지는 yBlk와, 0부터 (nPbW/nSbW - 1)의 범위를 가지는 xBlk에 대하여 다음을 적용한다.

[0452]

- 변수 curAvailableFlag 는 0으로 초기화한다.

- 0부터 1의 범위를 가지는 X에 대해서 다음을 적용한다.

  - 플래그 spPredFlagL1[xBlk][yBlk]는 0으로 초기화한다.

  - 움직임 벡터 spMvLX는 (0, 0)으로 초기화한다.

  - 참조 인덱스 spRefIdxLX[xBlk][yBlk]는 -1로 초기화한다.

  - 참조 블록 위치 (xRef, yRef)는 다음과 같이 유도된다.

    xRef = Clip3(0, PicWidthInSamplesL - 1,

    xPb + xBlk * nSbW + nSbW / 2 + ((mvDisp[0] + 2) >> 2))

    yRef = Clip3(0, PicHeightInSamplesL - 1,

    yPb + yBlk * nSbH + nSbH / 2 + ((mvDisp[1] + 2) >> 2))

- 변수 ivRefPb는 ivRefPic이 가리키는 시점 간 참조 픽처 내의 (xRef, yRef) 위치에서의 휘도 예측 블록을 나타낸다.

- (xIvRefPb, yIvRefPb)는 ivRefPb가 가리키는 참조 블록의 좌상단 위치를 나타낸다.

- ivRefPb가 인트라 모드로 부호화 되지 않았다면 0부터 1의 범위를 가지는 X에 대해서 다음을 수행한다.

  - X가 0이거나 현재 슬라이스가 B 슬라이스 일 때 X에서 (1 - X)의 범위를 가지는 Y에 대해서 다음을 수행한다.

    - refPixListLYIvRef, predFlagLYIvRef[x][y], mvLYIvRef[x][y], refIdxLYIvRef[x][y]는 각각 ivRefPic가 가리키는 픽처 내의 RefPicListLY, PredFlagLY[x][y],

[0453]

MvLY[x][y], RefIdxLY[x][y]로 세팅된다.

- predFlagLYIvRef[xIvRefPb][yIvRefPb]가 1이면, 0부터 num_ref_idx_lX_active_minus1(참조 픽처 리스트 내 참조 픽처 개수)의 범위를 가지는 i에 대해서 다음을 수행한다.

- refPicListLYIvRef[refIdxLYIvRef[xIvRefPb][yIvRefPb]]의 POC가 RefPicListLX[i] 와 같고 spPredFlagLX[xBlk][yBlk]가 0이라면, 다음을 적용한다.

spMvLX[xBlk][yBlk] = mvLYIvRef[xIvRefPb][yIvRefPb]

spRefIdxLX[xBlk][yBlk] = i

spPredFlagLX[xBlk][yBlk] = 1

curAvailableFlag = 1

- curAvailableFlag에 따라 다음을 적용한다

- curAvailableFlag가 0이면, 0부터 1의 범위를 가지는 X에 대해서 다음을 적용한다.

spMvLX[xBlk][yBlk] = mvLXZero

spRefIdxLX[xBlk][yBlk] = refIdxLXZero

spPredFlagLX[xBlk][yBlk] = availableFlagLXZero

- 변수 curSubBlockIdx는 curSubBlockIdx + 1로 세팅된다.

[0454]

[0455] 표 4에 기반해서 실시예 3을 다시 설명하면 아래와 같다. 표 3에서 사용하는 변수들은 표 2에서 설명한 바와 동일하다.

[0456] 인터 예측부(2100)에서는 참조 블록의 서브 예측 유닛을 이용하여 현재 서브 예측 유닛의 움직임 정보를 유도하기에 앞서, 초기화를 수행한다. 초기화의 내용은 표 2에서 설명한 바와 동일하다.

[0458] 또한, 변수 availableFlagLXZero, mvLXZero, 및 refIdxLXZero는 각각 수학식 23 및 수학식 24와 같이 설정된다. 이때, X는 0 또는 1이다.

[0459] <수학식 23>

$$availableFlagL0Zero = 1$$

$$mvL0Zero = (0,0)$$

$$refIdxL0Zero = 0$$

[0460]

[0461] <수학식 24>

$$availableFlagL1Zero = 1$$

$$mvL1Zero = (0,0)$$

$$refIdxL1Zero = 0$$

[0462]

[0463] 이때, availableFlagLXZero는 초기 움직임 정보가 사용 가능한지 여부에 대한 식별자를 의미하며, mvLXZero는 초기 움직임 정보를 의미하고, refIdxLXZero는 초기 움직임 정보의 참조 인덱스를 의미한다.

[0464] 상술한 바와 같이 변수들의 값을 설정한 후, 인터 예측부(2100)에서는 0부터 (nPbH / nSbH - 1)의 범위를 가지는 yBlk와, 0부터 (nPbW/nSbW - 1)의 범위를 가지는 xBlk에 대하여는 아래와 같은 과정을 수행한다. 이때, xBlk는 블록의 x좌표를 의미하며, yBlk는 블록의 y좌표를 의미한다.

[0465] 우선, 인터 예측부(2100)는 참조 블록의 서브 예측 유닛으로부터 움직임 정보의 예측 여부를 식별하는 정보, 서브 예측 유닛 예측 플래그, 서브 예측 유닛의 움직임 벡터, 및 서브 예측 유닛의 참조 인덱스를 초기화 한다. 구체적인 내용은 표 2에서 설명한 바와 동일하다.

[0466] 참조 블록의 위치 (xRef, yRef)는 서브 예측 블록 단위로 수학식 25와 같이 재설정된다.

[0467] <수학식 25>

$$xRef = Clip3(0, PicWidthInSamplesL-1, xPb + xBlk*nSbW + nSbW/2 + ((mvDisp[0]+2) >> 2)))$$

$$yRef = Clip3(0, PicHeightInSamplesL-1, yPb + yBlk*nSbH + nSbH/2 + ((mvDisp[1]+2) >> 2)))$$

[0468]

[0470] 시점 간 참조 블록이 인트라 모드로 부호화 되지 않았을 경우, 인터 예측부(2100)는 0부터 1의 범위를 가지는 X에 대해 아래의 과정을 수행할 수 있다.

[0471] 우선, X가 0이거나 현재 슬라이스가 B 슬라이스 일 때에는, Y(Y는 X에서 (1 - X)의 범위를 가진다.)에 대해 표 2에서 설명한 바와 같이 각각의 변수를 재설정한다.

[0472] predFlagLYIvRef[xIvRefPb][yIvRefPb]가 1의 값을 가지는 경우, 0부터 num_ref_idx_lX_active_minus1(참조 픽처 리스트 내에 존재하는 참조 픽처 개수)의 범위를 가지는 i에 대해서는 아래 수학식 26이 적용될 수 있다.

[0473] <수학식 26>

$$spMvLX[xBlk][yBlk] = mvLYIvRef[xIvRefPb][yIvRefPb]$$

$$spRefIdxLX[xBlk][yBlk] = i$$

$$spPredFlagLX[xBlk][yBlk] = 1$$

$$curAvailableFlag = 1$$

[0474]

[0476] 상술한 과정을 수행한 후 curAvailableFlag가 0일 경우, 인터 예측부(2100)는 0부터 1의 범위를 가지는 X에 대해서 수학식 27을 적용할 수 있다.

[0477] 즉, 참조 블록의 서브 예측 유닛으로부터 움직임 정보의 유도가 불가능할 경우, 인터 예측부(2100)는 임의로 설정된 초기 움직임 정보로부터 현재 블록의 서브 예측 유닛에 대한 움직임 정보를 유도할 수 있다.

[0478] <수학식 27>

$$spMvLX[xBlk][yBlk] = mvLXZero$$

$$spRefIdxLX[xBlk][yBlk] = refIdxLXZero$$

[0479] $$spPredFlagLX[xBlk][yBlk] = availableFlagLXZero$$

[0481] 마지막으로, 상술한 과정을 전부 수행한 경우, 변수 curSubBlockIdx는 curSubBlockIdx + 1로 설정되며, 만약 availableFlagL0InterView와 availableFlagL1InterView가 0이라면, 본 발명의 실시예 3에 따른 움직임 정보 유도 과정은 종료된다.

[0483] 도 26은 임의의 움직임 정보 값을 이용하여, 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도하는 과정의 일례를 나타내는 도면이다.

[0484] 도 26을 참조하면, 도 26의 상단에 위치한 블록들은 참조 블록의 서브 예측 유닛들을 의미하며, 도 26의 하단에 위치한 블록들은 현재 블록의 서브 예측 유닛들을 의미한다. 또한, 초기 움직임 정보는 저장 공간에 저장된다. 이때, 도 26의 초기 움직임 정보는 실시예 3의 임의로 설정된 초기 움직임 정보를 의미할 수 있다.

[0485] 초기 움직임 정보를 이용하여, 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도할 경우, 참조 블록의 서브 예측 유닛 각각은 임의의 값인 초기 움직임 정보를 이용할 수 있다. 즉, 현재 블록의 복수의 서브 예측 유닛은 임의의 초기 움직임 정보를 이용하여 동시에 움직임 정보의 유도를 할 수 있으며, 이로써, 현재 블록의 복수의 서브 예측 유닛은 데이터 의존성을 극복할 수 있다. 따라서, 임의의 값을 가지는 초기 움직임 정보를 이용할 경우, 인터 예측부(2100)는 병렬적인 움직임 정보 유도가 가능하게 된다.

[0486] 상술한 바와 같이, 실시예 3에서는 인터 예측부(2100)가 임의의 값을 가지는 초기 움직임 정보를 이용하여, 움직임 정보를 유도한다. 따라서, 실시예 3에 따른 움직임 정보 유도 방법은 참조 블록 내에 존재하는 각 서브 예측 유닛이 서로 독립적으로 유도될 수 있게 한다. 즉, 실시예 3에서는 움직임 정보 유도가 가능한 서브 예측 유닛을 찾기 위하여, 순차적으로 움직임 정보 유도 가능한 서브 예측 유닛을 탐색할 필요가 없으며, 참조 블록의 첫 번째 서브 예측 유닛이 움직임 정보 유도에 사용 불가능한 경우, 실시예 3에서는 기 설정된 움직임 정보를 이용하여 현재 블록에 대한 서브 예측 유닛의 움직임 정보를 유도한다. 이로 인하여, 실시예 3을 통한 움직임 정보 유도는 데이터 의존성을 제거하기 때문에, 각 서브 예측 유닛의 움직임 정보를 병렬적으로 유도할 수 있다는 이점이 있다. 또한, 실시예 3을 통한 움직임 정보 유도는, 기존의 움직임 정보 유도에 비하여 추가적인 메모리 접근을 막아 메모리 접근 회수를 줄이는 이점을 가져온다.

[0488] 도 27은 본 발명을 적용하여 움직임 정보를 유도할 때 소요되는 시간을 개략적으로 도시한 것이다.

[0489] 도 20에 따르면, 하나의 서브 예측 유닛 내의 움직임 정보를 유도하는데 걸리는 시간이 T이고, 참조 블록 내의 서브 예측 유닛의 개수가 N일때는 참조 블록 내의 모든 움직임 정보를 유도하는데 걸리는 시간이 NxT가 된다. 하지만, 본 발명의 실시예에 따른 움직임 정보 유도시에는, 움직임 정보 유도의 병렬화를 이룰 수 있기 때문에, 움직임 정보 유도시간은 T에 해당하게 되며, 3차원 비디오 부호화/복호화 지연 시간이 감소된다.

[0491] 상술한 본 발명의 실시예들은 블록 크기, 부호화 유닛(Coding Uint, CU) 깊이, 또는 변환 유닛(Transform Unit, TU) 깊이 등에 따라 적용 범위를 달리할 수 있다. 적용 범위를 결정하는 변수는 인코더/디코더에서 미리 정해진 값을 사용하도록 설정될 수도 있고, 프로파일 또는 레벨에 따라 정해진 값을 사용하도록 할 수도 있으며, 인코더가 변수 값을 비트스트림에 기재하면 디코더는 비트스트림으로부터 변수 값을 구하여 사용할 수도 있다.

[0492] 예컨대, CU 깊이에 따라 적용 범위를 달리하는 할 때는 표 5에서 예시된 바와 같이, 주어진 깊이 이상의 깊이에만 적용하는 방식(방법 A), 주어진 깊이 이하에만 적용하는 방식(방법 B), 또는 주어진 깊이에만 적용하는 방식(방법 C)이 있을 수 있다. 모든 깊이에 대하여 본 발명의 방법들을 적용하지 않는 경우는 임의의 지시자(flag)

를 사용하여 나타낼 수도 있고, CU 깊이 값으로 본 발명의 발법들을 적용하지 않음을 지시할 수도 있으며, 이때, CU 깊이 값은 CU가 가질 수 있는 최대의 깊이 값보다 더 큰 값으로 설정될 수 있다.

[0493] <표 5>

[0494]

적용 범위를 나타내는 CU(또는 PU, 또는 TU)의 깊이	방법 A	방법 B	방법 C
0	X	O	O
1	X	O	X
2	O	O	O
3	O	X	X
4 이상	O	X	X

[0496] 상술한 실시예들에서, 방법들은 일련의 단계 또는 유닛으로서 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.

[0497] 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.

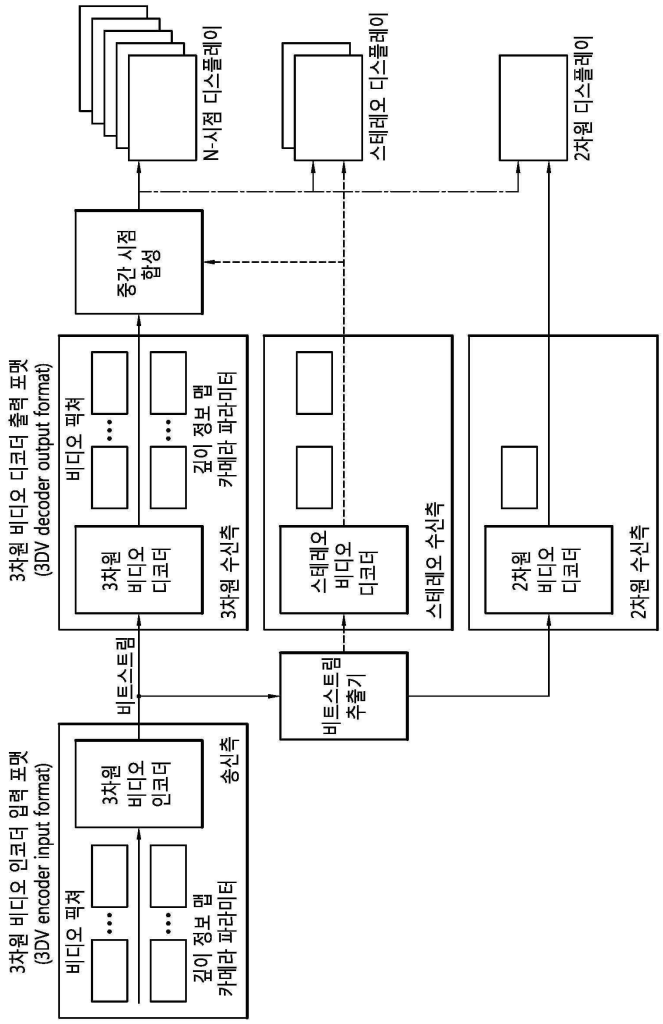
[0498] 상술한 본 발명에 따른 방법은 컴퓨터에서 실행되기 위한 프로그램으로 제작되어 컴퓨터가 읽을 수 있는 기록 매체에 저장될 수 있으며, 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다.

[0499] 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다. 그리고, 상기 방법을 구현하기 위한 기능적인(function) 프로그램, 코드 및 코드 세그먼트들은 본 발명이 속하는 기술분야의 프로그래머들에 의해 용이하게 추론될 수 있다.

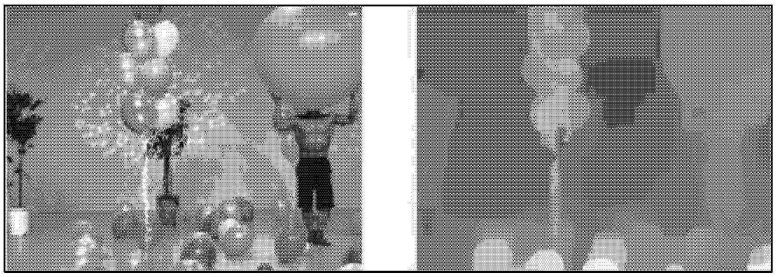
[0500] 또한, 본 발명의 바람직한 실시예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특정의 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변형 실시가 가능한 것은 물론이고, 이러한 변형 실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해 되어서는 안 될 것이다.

도면

도면1

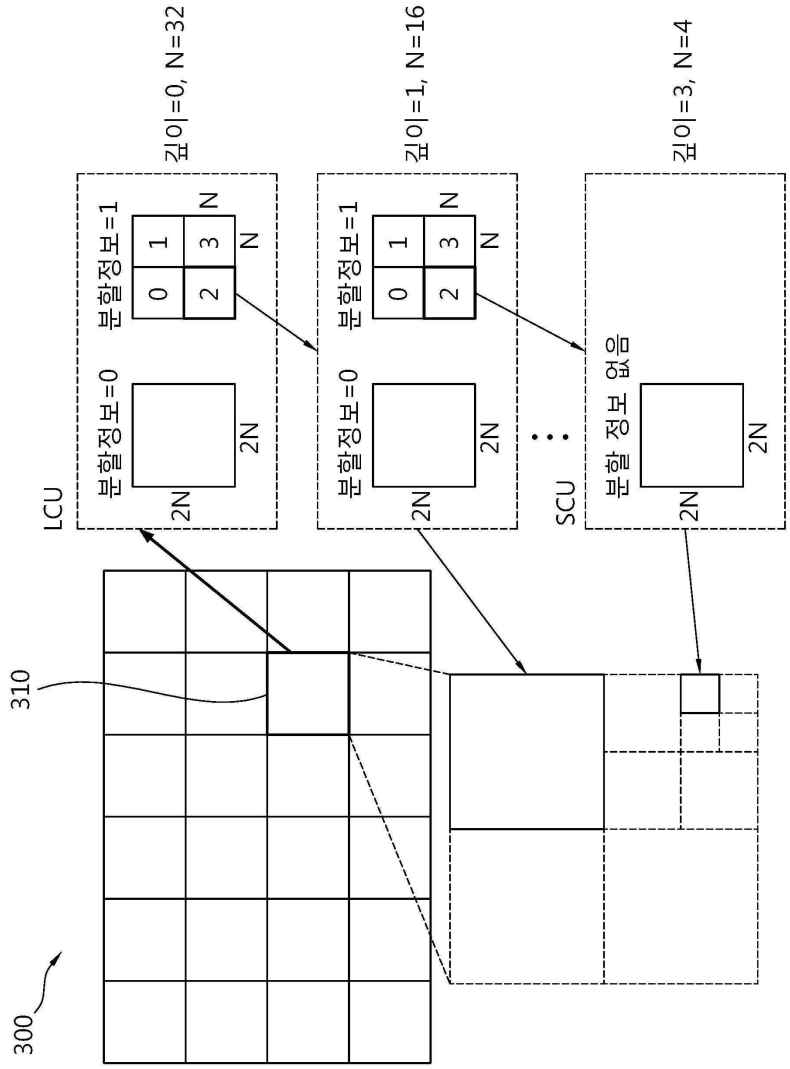


도면2

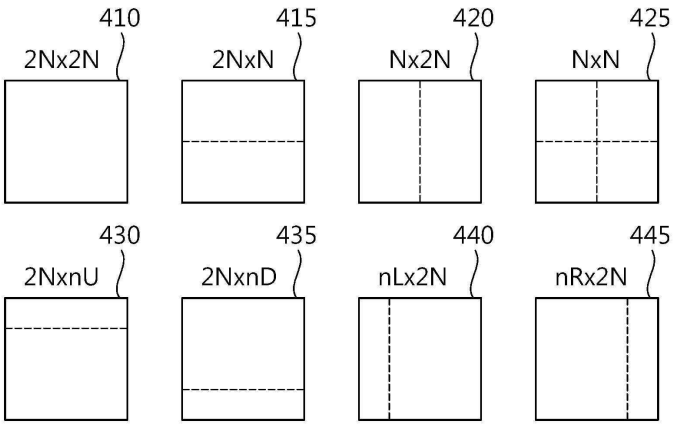


(a) 실제 영상 (b) 깊이 정보 맵

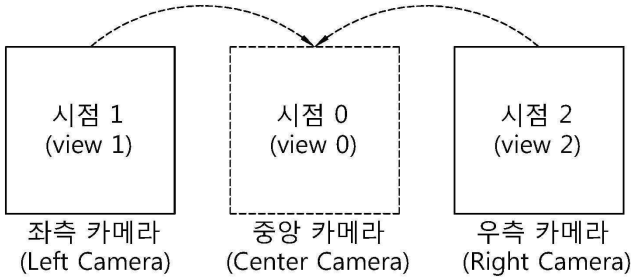
도면3


깊이=0, N=32
깊이=1, N=16
깊이=3, N=4
분할정보=1
분할정보=0
분할 정보 없음
LCU
SCU
2N
N
0
1
2
3
310
300

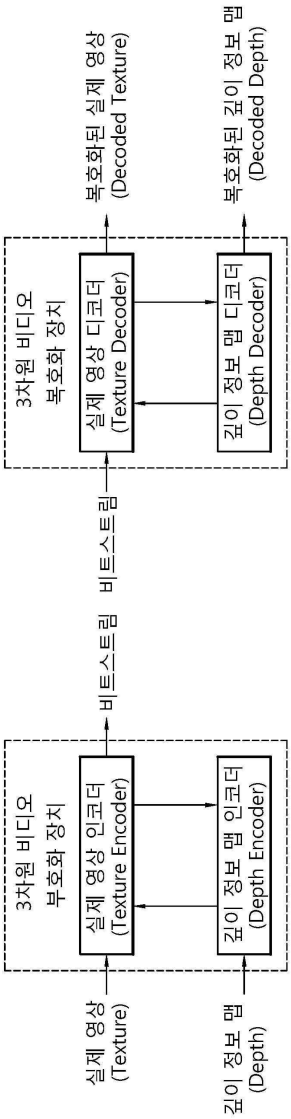
도면4


410
2Nx2N
415
2NxN
420
Nx2N
425
NxN
430
2NxnU
435
2NxnD
440
nLx2N
445
nRx2N

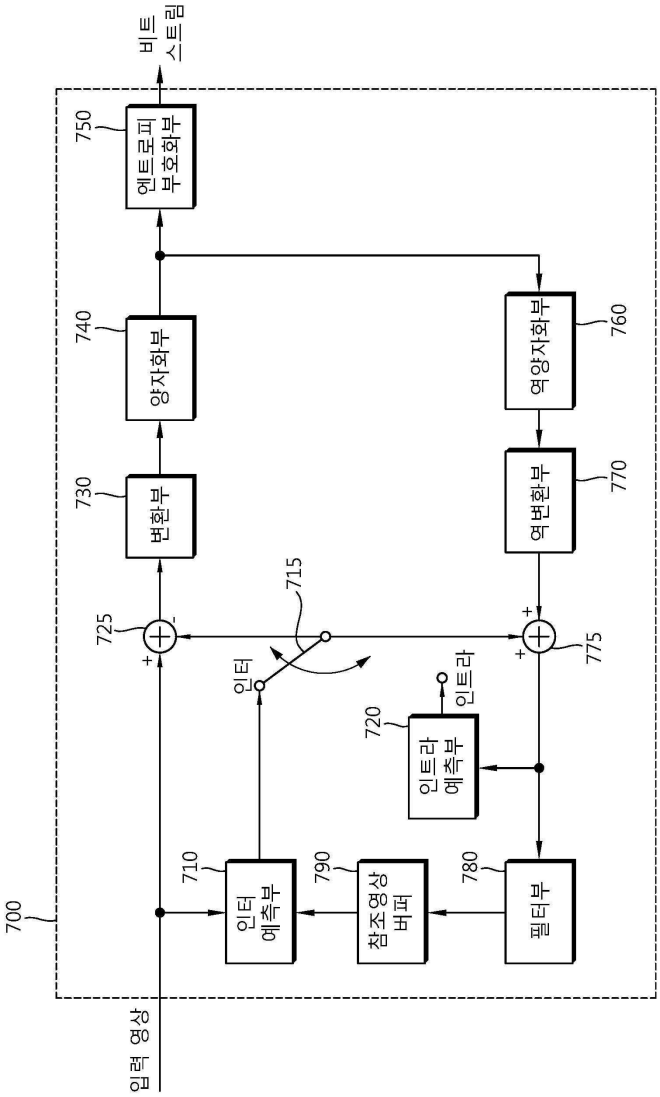
도면5


시점 1
(view 1)
시점 0
(view 0)
시점 2
(view 2)
좌측 카메라
(Left Camera)
중앙 카메라
(Center Camera)
우측 카메라
(Right Camera)

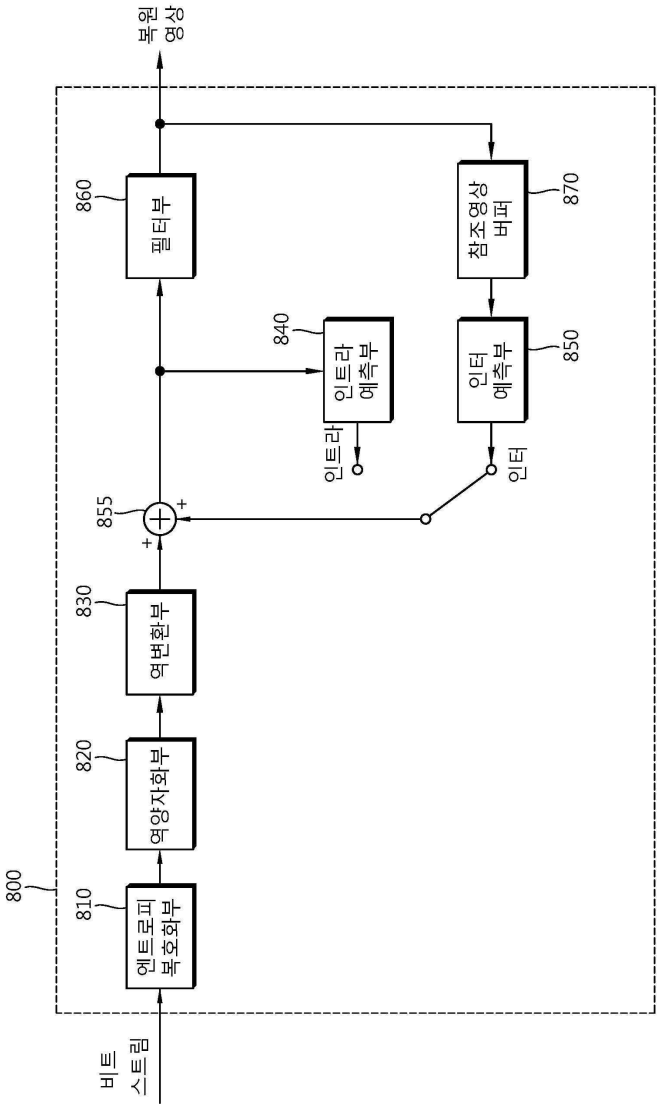
도면6


복호화된 실제 영상
(Decoded Texture)
복호화된 깊이 정보 맵
(Decoded Depth)
3차원 비디오
복호화 장치
실제 영상 디코더
(Texture Decoder)
깊이 정보 맵 디코더
(Depth Decoder)
비트스트림
비트스트림
3차원 비디오
부호화 장치
실제 영상 인코더
(Texture Encoder)
깊이 정보 맵 인코더
(Depth Encoder)
실제 영상
(Texture)
깊이 정보 맵
(Depth)

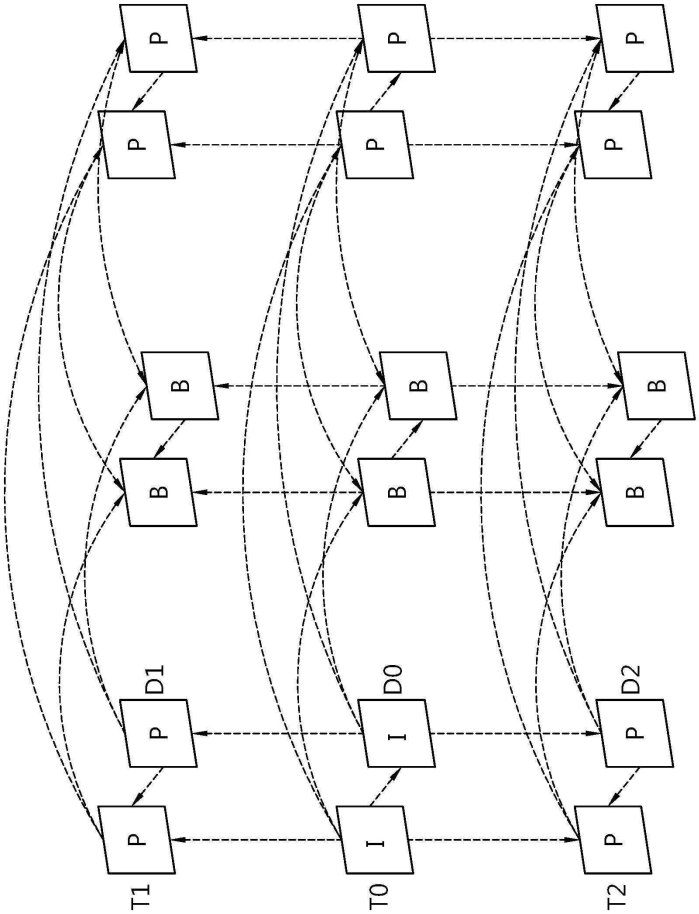
도면7


비트 스트림
750
엔트로피 부호화부
740
양자화부
730
변환부
760
역양자화부
770
역변환부
715
725
775
인터
인트라
720
인트라 예측부
710
인터 예측부
790
참조영상 버퍼
780
필터부
700
입력 영상

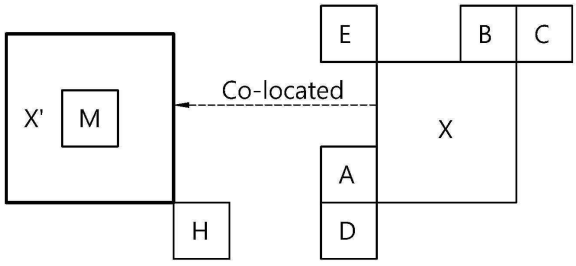
도면8


복원 영상
860
필터부
870
참조영상 버퍼
840
인트라 예측부
850
인터 예측부
인트라
인터
855
+
+
830
역변환부
820
역양자화부
810
엔트로피 복호화부
800
비트 스트림

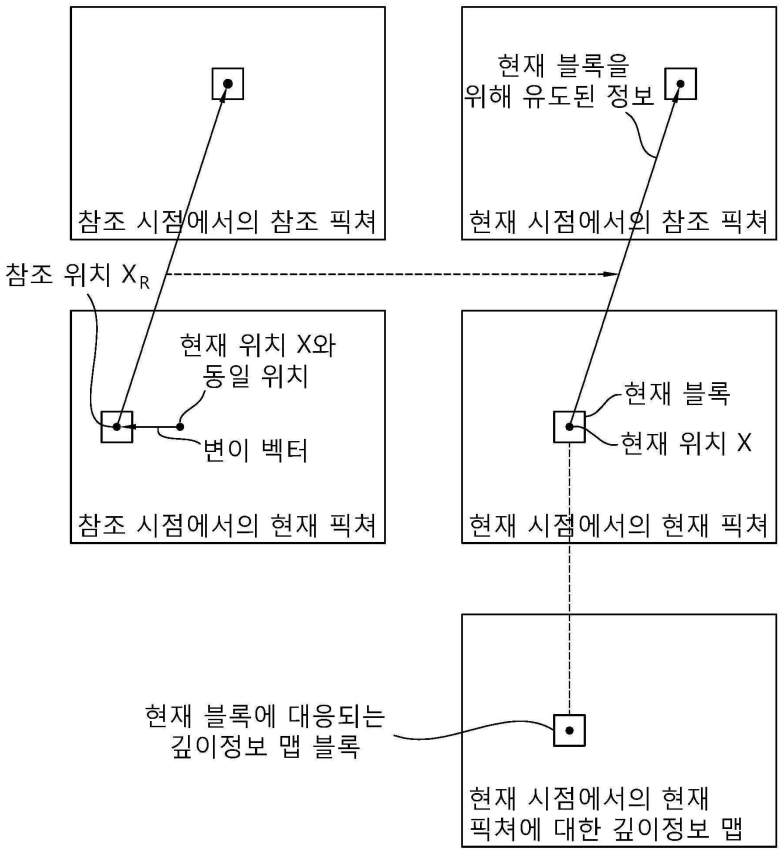
도면9



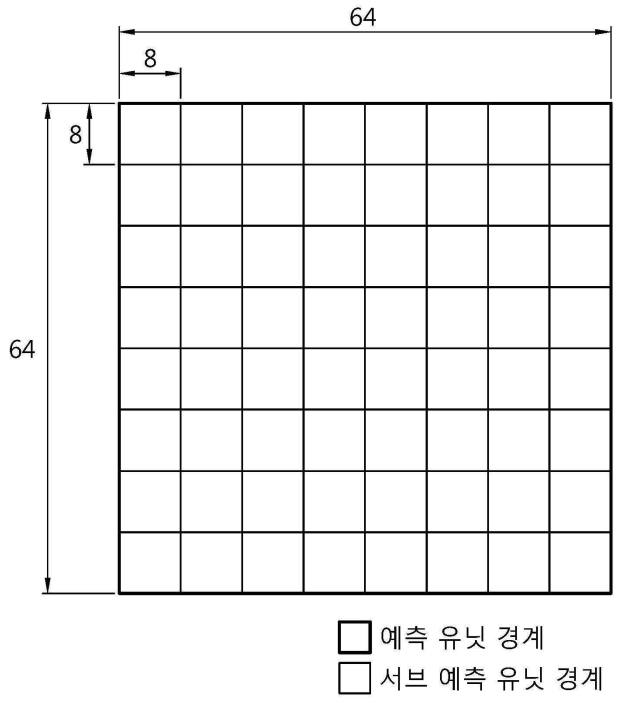
도면10



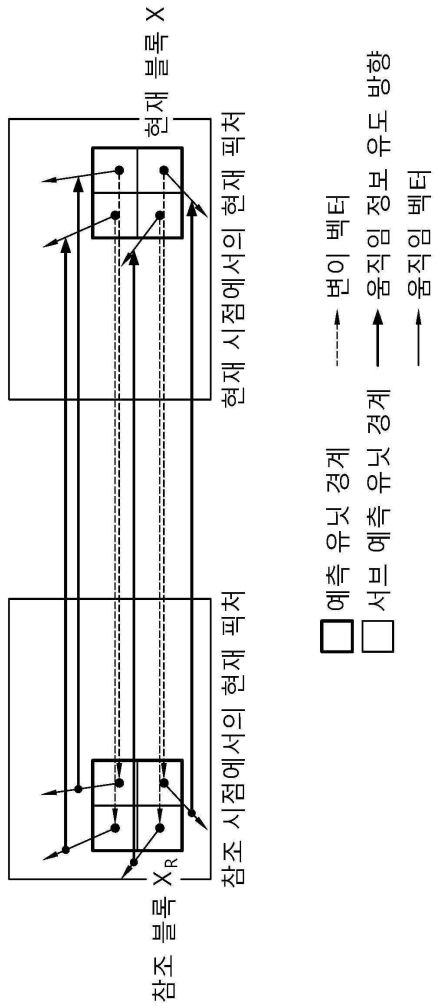
도면11


참조 시점에서의 참조 픽쳐
현재 블록을
위해 유도된 정보
현재 시점에서의 참조 픽쳐
참조 위치 X_R
현재 위치 X와
동일 위치
변이 벡터
참조 시점에서의 현재 픽쳐
현재 블록
현재 위치 X
현재 시점에서의 현재 픽쳐
현재 블록에 대응되는
깊이정보 맵 블록
현재 시점에서의 현재
픽쳐에 대한 깊이정보 맵

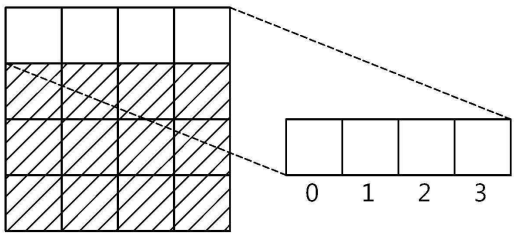
도면12


64
8
8
64
예측 유닛 경계
서브 예측 유닛 경계

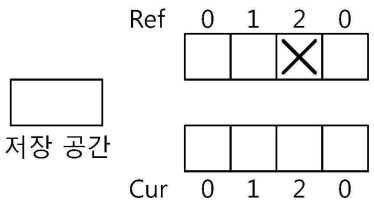
도면13



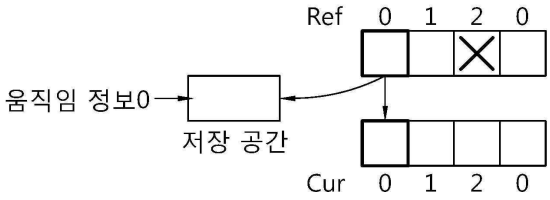
도면14



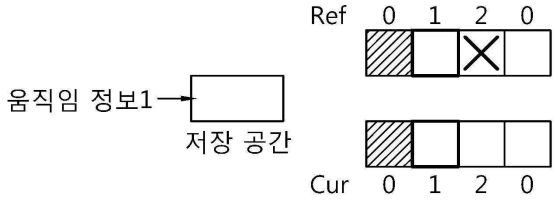
도면15a



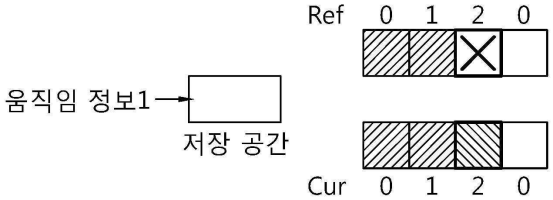
도면15b



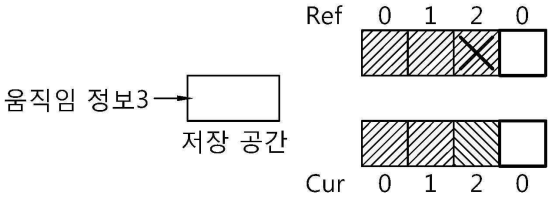
도면15c



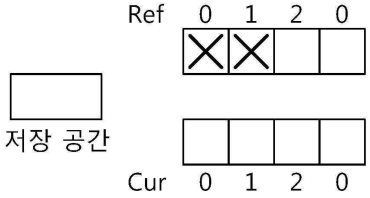
도면15d



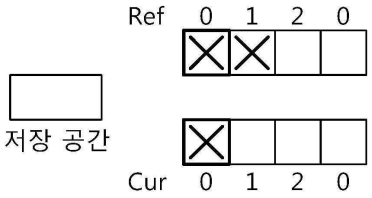
도면15e



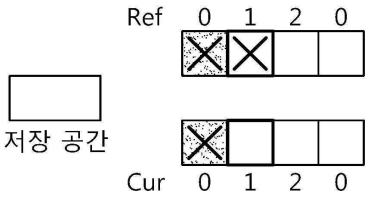
도면16a



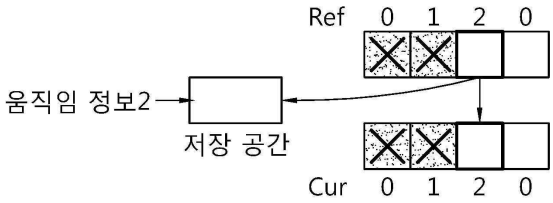
도면16b



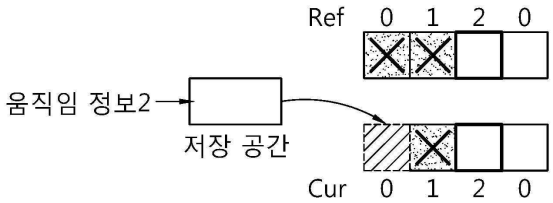
도면16c



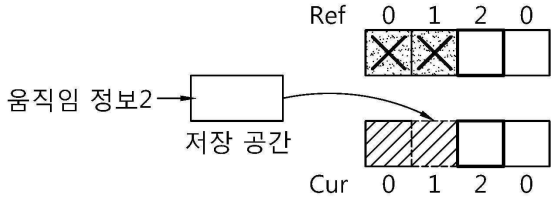
도면16d



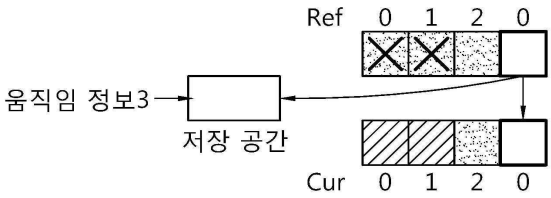
도면16e



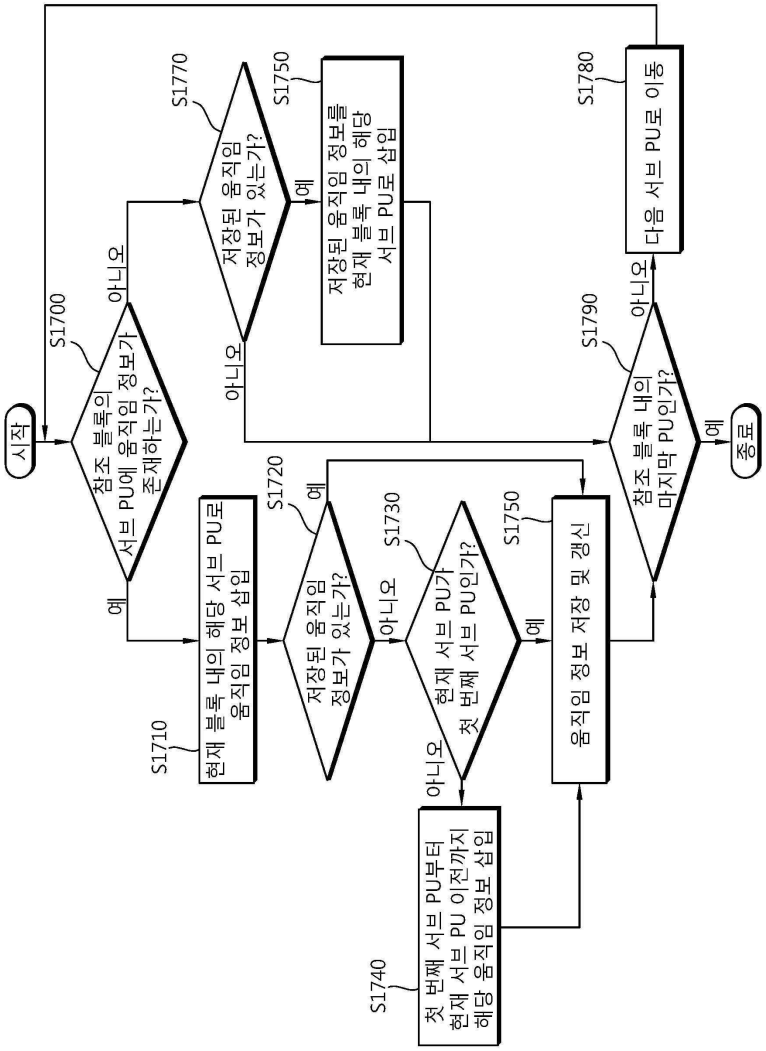
도면16f



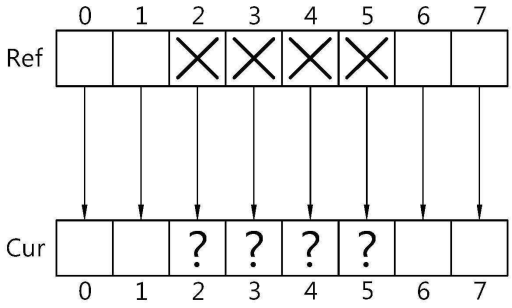
도면16g


Ref 0 1 2 0
움직임 정보3
저장 공간
Cur 0 1 2 0

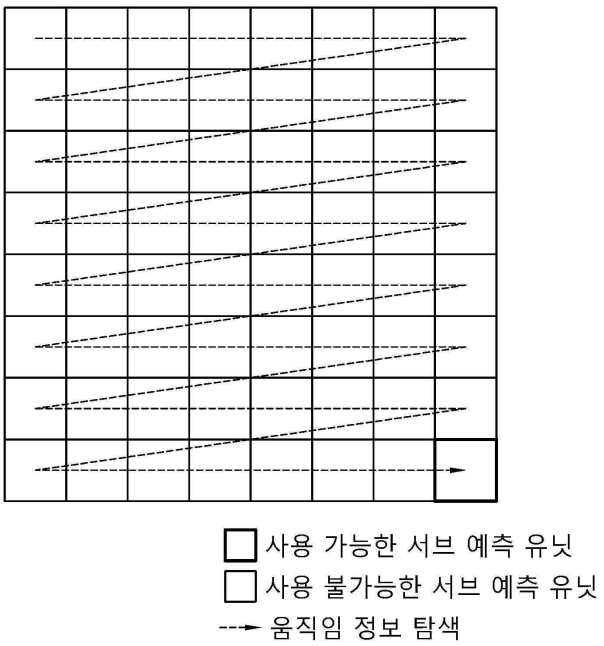
도면17


시작
S1700
참조 블록의 서브 PU에 움직임 정보가 존재하는가?
아니오
예
S1710
현재 블록 내의 해당 서브 PU로 움직임 정보 삽입
S1720
저장된 움직임 정보가 있는가?
아니오
예
S1730
현재 서브 PU가 첫 번째 서브 PU인가?
아니오
예
S1740
첫 번째 서브 PU부터 현재 서브 PU 이전까지 해당 움직임 정보 삽입
S1750
움직임 정보 저장 및 갱신
S1770
저장된 움직임 정보가 있는가?
아니오
예
S1750
저장된 움직임 정보를 현재 블록 내의 해당 서브 PU로 삽입
S1780
다음 서브 PU로 이동
S1790
참조 블록 내의 마지막 PU인가?
아니오
예
종료

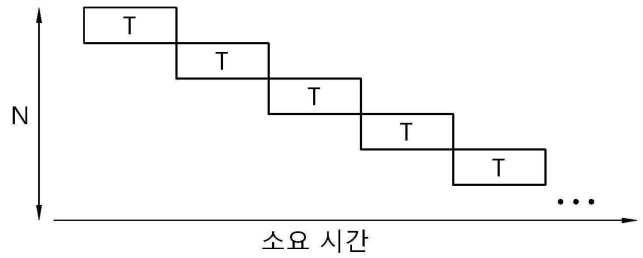
도면18



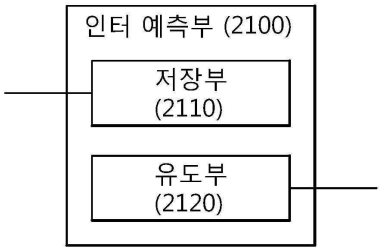
도면19



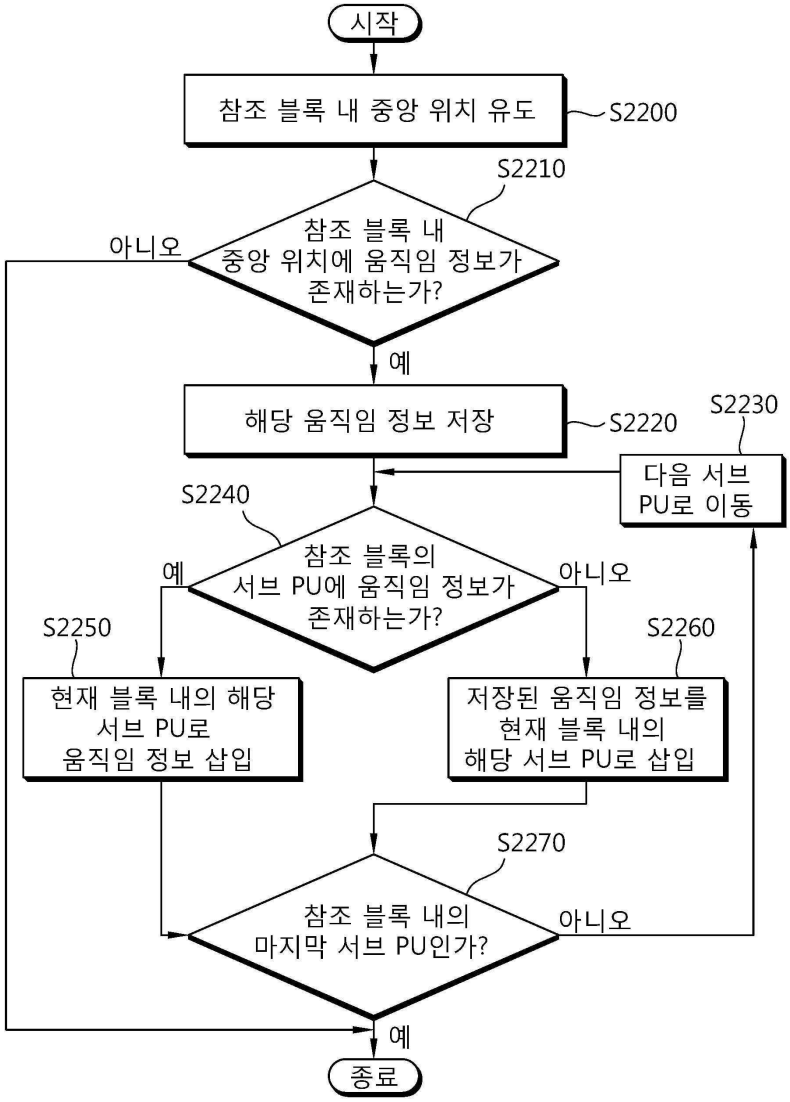
도면20



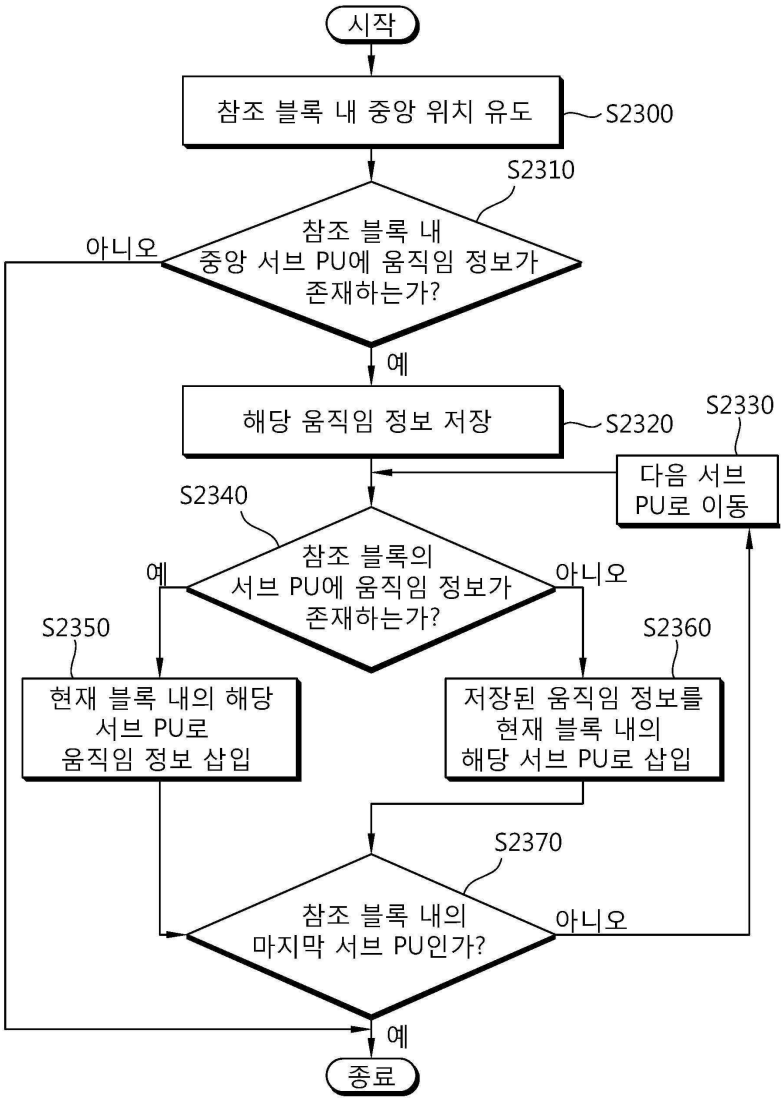
도면21


인터 예측부 (2100)
저장부
(2110)
유도부
(2120)

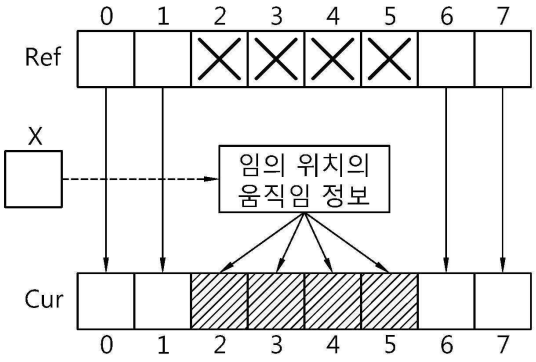
도면22


시작
참조 블록 내 중앙 위치 유도
S2200
S2210
참조 블록 내
중앙 위치에 움직임 정보가
존재하는가?
아니오
예
해당 움직임 정보 저장
S2220
S2230
다음 서브
PU로 이동
S2240
참조 블록의
서브 PU에 움직임 정보가
존재하는가?
예
아니오
S2250
현재 블록 내의 해당
서브 PU로
움직임 정보 삽입
S2260
저장된 움직임 정보를
현재 블록 내의
해당 서브 PU로 삽입
S2270
참조 블록 내의
마지막 서브 PU인가?
아니오
예
종료

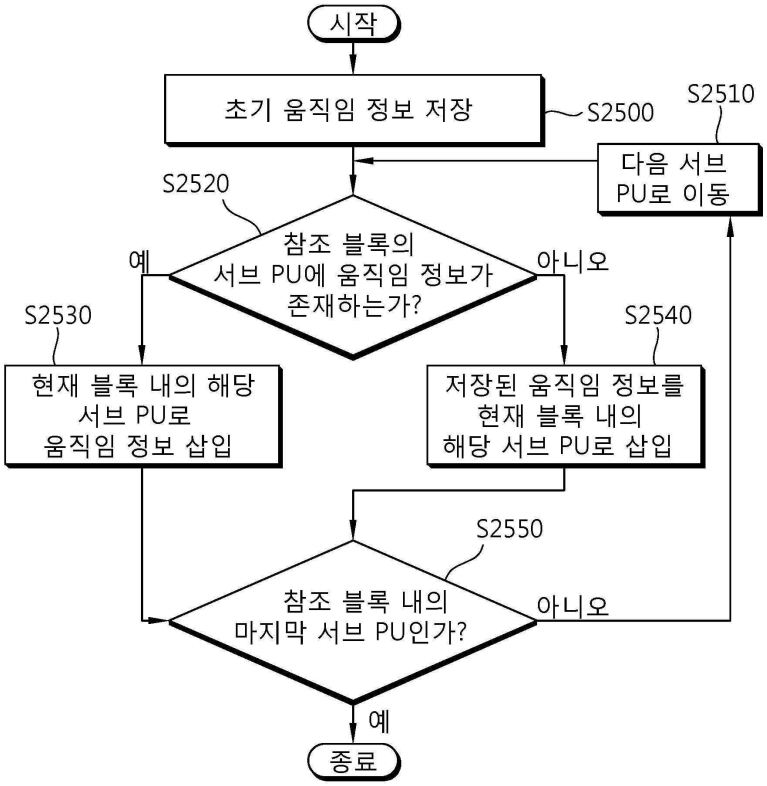
도면23


시작
참조 블록 내 중앙 위치 유도
S2300
S2310
참조 블록 내
중앙 서브 PU에 움직임 정보가
존재하는가?
아니오
예
해당 움직임 정보 저장
S2320
S2330
다음 서브
PU로 이동
S2340
참조 블록의
서브 PU에 움직임 정보가
존재하는가?
예
아니오
S2350
S2360
현재 블록 내의 해당
서브 PU로
움직임 정보 삽입
저장된 움직임 정보를
현재 블록 내의
해당 서브 PU로 삽입
S2370
참조 블록 내의
마지막 서브 PU인가?
아니오
예
종료

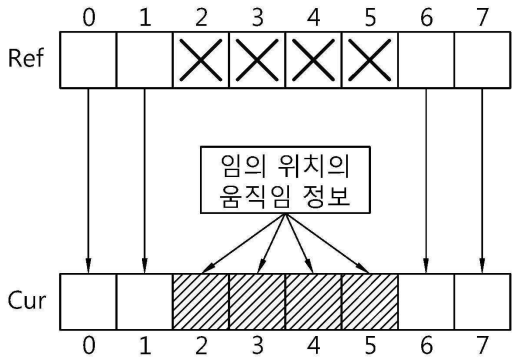
도면24


0 1 2 3 4 5 6 7
Ref
X
임의 위치의
움직임 정보
Cur
0 1 2 3 4 5 6 7

도면25



도면26



도면27

