



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0047414
(43) 공개일자 2015년05월04일

(51) 국제특허분류(Int. Cl.)
H04N 19/597 (2014.01)

(21) 출원번호 10-2014-0048066
(22) 출원일자 2014년04월22일
심사청구일자 없음

(30) 우선권주장
1020130126852 2013년10월24일 대한민국(KR)
1020130146600 2013년11월28일 대한민국(KR)

(71) 출원인

한국전자통신연구원

대전광역시 유성구 가정로 218 (가정동)

경희대학교 산학협력단

경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)

(72) 발명자

방건

대전광역시 유성구 송강로42번길 61, 310-1003 (송강동, 송강청솔아파트)

이광순

대전광역시 유성구 진잠로149번길 30, 208동303호 (교촌동)

(뒷면에 계속)

(74) 대리인

에스앤아이퍼특허법인

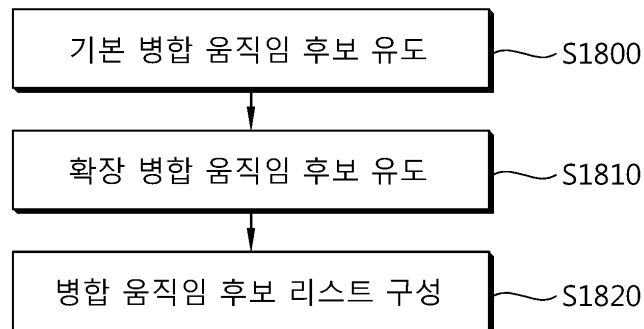
전체 청구항 수 : 총 15 항

(54) 발명의 명칭 비디오 부호화/복호화 방법 및 장치

(57) 요약

복수의 시점(view)을 포함하는 비디오 부호화/복호화 방법 및 장치가 개시된다. 상기 복수의 시점을 포함하는 비디오 복호화 방법은 현재 PU(Prediction Unit)에 대한 기본 병합 움직임 후보를 유도하여 병합 움직임 후보 리스트를 구성하는 단계, 상기 현재 PU가 깊이정보 맵 또는 의존적인 시점인 경우, 상기 현재 PU에 대한 확장 병합 움직임 후보를 유도하는 단계 및 상기 병합 움직임 후보 리스트에 상기 확장 병합 움직임 후보를 추가하는 단계를 포함한다.

대표도 - 도18



(72) 발명자

허남호

대전광역시 유성구 은구비남로 34, 801동1001호 (노은동, 열매마을8단지)

김경용

경기도 수원시 영통구 덕영대로 1725, 402호 (영통동)

박광훈

경기도 성남시 분당구 예원로 7, B동302호 (분당동, 동아빌라)

이윤진

경기도 용인시 기흥구 흥덕2로 80, 309동 1502호 (영덕동, 휴먼시아 흥덕마을3단지)

허영수

경기도 수원시 영통구 봉영로1492번길 20, B01호 (영통동)

이 발명을 지원한 국가연구개발사업

과제고유번호 11-921-02-001

부처명 미래창조과학부

연구관리전문기관 한국방송통신전파진흥원

연구사업명 방송통신ETRI연구개발지원사업

연구과제명 무안경 다시점 3D 지원 UHDTV 방송 기술 개발

기 여 율 1/1

주관기관 한국전자통신연구원

연구기간 2011.03.01 ~ 2015.02.28

명세서

청구범위

청구항 1

복수의 시점(view)을 포함하는 비디오 복호화 방법에 있어서,

현재 PU(Prediction Unit)에 대한 기본 병합 움직임 후보를 유도하여 병합 움직임 후보 리스트를 구성하는 단계;

상기 현재 PU가 깊이정보 맵 또는 의존적인 시점인 경우, 상기 현재 PU에 대한 확장 병합 움직임 후보를 유도하는 단계; 및

상기 병합 움직임 후보 리스트에 상기 확장 병합 움직임 후보를 추가하는 단계를 포함하며,

상기 확장 병합 움직임 후보를 추가하는 단계에서는,

상기 확장 병합 움직임 후보가 상기 병합 움직임 후보 리스트 내 상기 기본 병합 움직임 후보와 동일하지 않을 경우, 상기 확장 병합 움직임 후보를 상기 병합 움직임 후보 리스트에 추가하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 2

제1항에 있어서,

상기 확장 병합 움직임 후보를 추가하는 단계에서는,

상기 확장 병합 움직임 후보를 상기 병합 움직임 후보 리스트 내 임의의 위치에 추가하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 3

제1항에 있어서,

상기 기본 병합 움직임 후보는,

상기 현재 PU의 공간적 병합 움직임 후보 및 시간적 병합 움직임 후보 중 적어도 하나를 포함하며,

상기 공간적 병합 움직임 후보는,

상기 현재 PU에 공간적으로 인접하여 위치한 좌(left) 블록, 상(above) 블록, 우상(above-right) 블록, 좌하(bottom-left) 블록 및 좌상(above-left) 블록 중 적어도 하나의 블록을 포함하며,

상기 시간적 병합 움직임 후보는,

상기 현재 PU에 대한 동일 위치 픽처(co-located picture) 내 동일 위치 블록(co-located block)을 포함하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 4

제1항에 있어서,

상기 확장 병합 움직임 후보는,

시점 간 병합 후보(inter-view merge candidate: IvMC), 시점 합성 병합 후보(view synthesis prediction merge candidate) 및 텍스처 병합 후보(texture merge candidate) 중 적어도 하나를 포함하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 5

제4항에 있어서,

상기 확장 병합 움직임 후보를 유도하는 단계에서는,

상기 현재 PU가 시점 간 예측을 수행하는지 여부에 따라 상기 시점 간 병합 후보를 유도하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 6

제4항에 있어서,

상기 확장 병합 움직임 후보를 유도하는 단계에서는,

상기 현재 PU가 시점 합성 예측을 수행하는지 여부에 따라 상기 시점 합성 병합 후보를 유도하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 7

제4항에 있어서,

상기 확장 병합 움직임 후보를 유도하는 단계에서는,

상기 현재 PU의 깊이정보 맵이 텍스처 블록으로부터 움직임 예측을 수행하는지 여부에 따라 상기 텍스처 병합 후보를 유도하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 8

제2항에 있어서,

상기 임의의 위치는 상기 병합 움직임 후보 리스트 내 첫번째 항목(index)인 것을 특징으로 하는 비디오 복호화 방법.

청구항 9

제1항에 있어서,

상기 확장 병합 움직임 후보를 추가하는 단계에서는,

상기 병합 움직임 후보 리스트에 추가된 확장 병합 움직임 후보 개수와 상기 기본 병합 움직임 후보 개수를 더한 값이 상기 병합 움직임 후보 리스트의 최대 후보 개수보다 작은 경우, 상기 확장 병합 움직임을 상기 병합 움직임 후보 리스트에 추가하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 10

제7항에 있어서,

상기 현재 PU의 깊이정보 맵이 텍스처 블록으로부터 움직임 예측을 수행하는 경우,

상기 확장 병합 움직임 후보를 추가하는 단계에서는,

상기 텍스처 병합 후보가 상기 병합 움직임 후보 리스트 내 상기 기본 병합 움직임 후보와 동일하지 않을 경우, 상기 텍스처 병합 후보를 상기 병합 움직임 후보 리스트 내 첫번째 항목에 추가하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 11

제5항에 있어서,

상기 현재 PU가 시점 간 예측을 수행하는 경우,

상기 확장 병합 움직임 후보를 추가하는 단계에서는,

상기 시점 간 병합 후보가 상기 병합 움직임 후보 리스트 내 상기 기본 병합 움직임 후보와 동일하지 않을 경우, 상기 시점 간 병합 후보를 상기 병합 움직임 후보 리스트 내 첫번째 항목에 추가하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 12

제6항에 있어서,

상기 현재 PU가 시점 합성 예측을 수행하는 경우,

상기 확장 병합 움직임 후보를 추가하는 단계에서는,

상기 병합 움직임 후보 리스트에 추가된 확장 병합 움직임 후보 개수와 상기 기본 병합 움직임 후보 개수를 더한 값이 상기 병합 움직임 후보 리스트의 최대 후보 개수보다 작은 경우, 상기 시점 합성 병합 후보를 상기 병합 움직임 후보 리스트에 추가하는 것을 특징으로 하는 비디오 복호화 방법.

청구항 13

복수의 시점(view)을 포함하는 비디오 복호화 장치에 있어서,

현재 PU(Prediction Unit)에 대한 기본 병합 움직임 후보를 유도하여 병합 움직임 후보 리스트를 구성하는 기본 병합 움직임 리스트 구성 모듈; 및

상기 현재 PU가 깊이정보 맵 또는 의존적인 시점인 경우, 상기 현재 PU에 대한 확장 병합 움직임 후보를 유도하고, 상기 병합 움직임 후보 리스트에 상기 확장 병합 움직임 후보를 추가하는 추가 병합 움직임 리스트 구성 모듈을 포함하며,

상기 추가 병합 움직임 리스트 구성 모듈은,

상기 확장 병합 움직임 후보가 상기 병합 움직임 후보 리스트 내 상기 기본 병합 움직임 후보와 동일하지 않을 경우, 상기 확장 병합 움직임 후보를 상기 병합 움직임 후보 리스트에 추가하는 것을 특징으로 하는 비디오 복호화 장치.

청구항 14

복수의 시점(view)을 포함하는 비디오 부호화 방법에 있어서,

현재 PU(Prediction Unit)에 대한 기본 병합 움직임 후보를 유도하여 병합 움직임 후보 리스트를 구성하는 단계;

상기 현재 PU가 깊이정보 맵 또는 의존적인 시점인 경우, 상기 현재 PU에 대한 확장 병합 움직임 후보를 유도하는 단계; 및

상기 병합 움직임 후보 리스트에 상기 확장 병합 움직임 후보를 추가하는 단계를 포함하며,

상기 확장 병합 움직임 후보를 추가하는 단계에서는,

상기 확장 병합 움직임 후보가 상기 병합 움직임 후보 리스트 내 상기 기본 병합 움직임 후보와 동일하지 않을 경우, 상기 확장 병합 움직임 후보를 상기 병합 움직임 후보 리스트에 추가하는 것을 특징으로 하는 비디오 부호화 방법.

청구항 15

복수의 시점(view)을 포함하는 비디오 부호화 장치에 있어서,

현재 PU(Prediction Unit)에 대한 기본 병합 움직임 후보를 유도하여 병합 움직임 후보 리스트를 구성하는 기본 병합 움직임 리스트 구성 모듈; 및

상기 현재 PU가 깊이정보 맵 또는 의존적인 시점인 경우, 상기 현재 PU에 대한 확장 병합 움직임 후보를 유도하고, 상기 병합 움직임 후보 리스트에 상기 확장 병합 움직임 후보를 추가하는 추가 병합 움직임 리스트 구성 모듈을 포함하며,

상기 추가 병합 움직임 리스트 구성 모듈은,

상기 확장 병합 움직임 후보가 상기 병합 움직임 후보 리스트 내 상기 기본 병합 움직임 후보와 동일하지 않을 경우, 상기 확장 병합 움직임 후보를 상기 병합 움직임 후보 리스트에 추가하는 것을 특징으로 하는 비디오 부호화 장치.

발명의 설명

기술 분야

[0001] 본 발명은 비디오 부호화/복호화 방법 및 장치에 관한 것으로, 보다 상세하게는 3차원 비디오(3D video) 코딩을 위한 병합 움직임 후보 리스트(merge motion candidate list)를 구성하는 방법 및 장치에 관한 것이다.

배경 기술

[0002] 3차원 비디오는 3차원 입체 디스플레이 장치를 통해 사용자에게 실세계에서 보고 느끼는 것과 같은 입체감을 생성하게 제공한다. 이와 관련된 연구로써 ISO/IEC의 MPEG(Moving Picture Experts Group)과 ITU-T의 VCEG(Video Coding Experts Group)의 공동 표준화 그룹인 JCT-3V(The Joint Collaborative Team on 3D Video Coding Extension Development)에서 3차원 비디오 표준이 진행 중이다. 3차원 비디오 표준은 실제 영상(texture view)과 그것의 깊이정보 맵(depth map; depth view)을 이용하여 스테레오스코픽 영상뿐만 아니라 오토스테레오스코픽 영상의 재생 등을 지원할 수 있는 진보된 데이터 형식과 그에 관련된 기술에 대한 표준을 포함하고 있다.

발명의 내용

해결하려는 과제

[0003] 본 발명은 영상 부호화/복호화 효율을 향상시킬 수 있는 영상 부호화/복호화 방법 및 장치를 제공한다.

[0004] 본 발명은 부호화/복호화 효율을 향상시킬 수 있는 3차원 비디오 부호화/복호화 방법 및 장치를 제공한다.

[0005] 본 발명은 3차원 비디오 부호화/복호화 시 병합 움직임 후보 리스트를 구성하는 방법 및 장치를 제공한다.

과제의 해결 수단

[0006] 본 발명의 일 실시예에 따르면, 복수의 시점(view)을 포함하는 비디오 복호화 방법이 제공된다. 상기 비디오 복호화 방법은 현재 PU(Prediction Unit)에 대한 기본 병합 움직임 후보를 유도하여 병합 움직임 후보 리스트를 구성하는 단계, 상기 현재 PU가 깊이정보 맵 또는 의존적인 시점인 경우, 상기 현재 PU에 대한 확장 병합 움직임 후보를 유도하는 단계 및 상기 병합 움직임 후보 리스트에 상기 확장 병합 움직임 후보를 추가하는 단계를 포함한다.

[0007] 상기 확장 병합 움직임 후보를 추가하는 단계에서는, 상기 확장 병합 움직임 후보가 상기 병합 움직임 후보 리스트 내 상기 기본 병합 움직임 후보와 동일하지 않을 경우, 상기 확장 병합 움직임 후보를 상기 병합 움직임 후보 리스트에 추가할 수 있다.

[0008] 본 발명의 다른 실시예에 따르면, 복수의 시점(view)을 포함하는 비디오 복호화 장치가 제공된다. 상기 비디오 복호화 장치는 현재 PU(Prediction Unit)에 대한 기본 병합 움직임 후보를 유도하여 병합 움직임 후보 리스트를 구성하는 기본 병합 움직임 리스트 구성 모듈 및 상기 현재 PU가 깊이정보 맵 또는 의존적인 시점인 경우, 상기 현재 PU에 대한 확장 병합 움직임 후보를 유도하고, 상기 병합 움직임 후보 리스트에 상기 확장 병합 움직임 후보를 추가하는 추가 병합 움직임 리스트 구성 모듈을 포함한다.

[0009] 상기 추가 병합 움직임 리스트 구성 모듈은, 상기 확장 병합 움직임 후보가 상기 병합 움직임 후보 리스트 내 상기 기본 병합 움직임 후보와 동일하지 않을 경우, 상기 확장 병합 움직임 후보를 상기 병합 움직임 후보 리스트에 추가할 수 있다.

[0010] 본 발명의 또 다른 실시예에 따르면, 복수의 시점(view)을 포함하는 비디오 부호화 방법이 제공된다. 상기 비디오 부호화 방법은 현재 PU(Prediction Unit)에 대한 기본 병합 움직임 후보를 유도하여 병합 움직임 후보 리스트를 구성하는 단계, 상기 현재 PU가 깊이정보 맵 또는 의존적인 시점인 경우, 상기 현재 PU에 대한 확장 병합 움직임 후보를 유도하는 단계 및 상기 병합 움직임 후보 리스트에 상기 확장 병합 움직임 후보를 추가하는 단계를 포함한다.

[0011] 상기 확장 병합 움직임 후보를 추가하는 단계에서는, 상기 확장 병합 움직임 후보가 상기 병합 움직임 후보 리스트 내 상기 기본 병합 움직임 후보와 동일하지 않을 경우, 상기 확장 병합 움직임 후보를 상기 병합 움직임 후보 리스트에 추가할 수 있다.

[0012] 본 발명의 또 다른 실시예에 따르면, 복수의 시점(view)을 포함하는 비디오 부호화 장치가 제공된다. 상기 비디오 부호화 장치는 현재 PU(Prediction Unit)에 대한 기본 병합 움직임 후보를 유도하여 병합 움직임 후보 리스트를 구성하는 기본 병합 움직임 리스트 구성 모듈 및 상기 현재 PU가 깊이정보 맵 또는 의존적인 시점인 경우, 상기 현재 PU에 대한 확장 병합 움직임 후보를 유도하고, 상기 병합 움직임 후보 리스트에 상기 확장 병합 움직임 후보를 추가하는 추가 병합 움직임 리스트 구성 모듈을 포함한다.

[0013] 상기 추가 병합 움직임 리스트 구성 모듈은, 상기 확장 병합 움직임 후보가 상기 병합 움직임 후보 리스트 내 상기 기본 병합 움직임 후보와 동일하지 않을 경우, 상기 확장 병합 움직임 후보를 상기 병합 움직임 후보 리스트에 추가할 수 있다.

발명의 효과

[0014] 역방향 호환성을 제공하는 독립적인 시점(Independent View; 시점 0(View 0))에 대한 일반 영상의 부호화에 사용되는 모듈을 의존적인 시점(dependent view; 시점 1(View 1)과 시점 2(View 2))에 대한 일반 영상과 깊이정보 맵들에 그대로 적용하여 구현 복잡도를 감소시킬 수 있다.

[0015] 또한, 의존적인 시점(dependent view; 시점 1(View 1)과 시점 2(View 2))에 대한 일반 영상과 깊이정보 맵들에 부분 인코더를 추가적으로 적용함으로써 부호화 효율을 향상시킬 수 있다.

도면의 간단한 설명

[0016] 도 1은 3차원 비디오 시스템의 기본 구조와 데이터 형식을 개략적으로 나타내는 일례이다.

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

도 3은 3D 비디오 코덱(codec)에서 시점간 예측(inter view prediction)의 구조를 나타내는 일례이다.

도 4는 3D 비디오 인코더/디코더에서 실제 영상(texture view)과 깊이정보 맵(depth view)을 부호화/복호화하는 과정을 나타낸 일례이다.

도 5는 3D 비디오 코덱의 예측 구조의 일례를 나타낸다.

도 6은 3차원 비디오 코덱(3D Video Codec)의 인코더를 개략적으로 나타낸 구조도이다.

도 7은 HEVC 기반 3차원 비디오 코덱(3D-HEVC)에서 사용되는 병합 움직임 방법을 설명하기 위해 도시된 도면이다.

도 8은 현재 블록에 대한 병합 움직임 리스트를 구성하는데 사용되는 주변 블록들을 나타낸 일례이다.

도 9는 병합 움직임 후보 리스트를 구성하는 방법을 하드웨어로 구현한 일례를 나타내는 도면이다.

도 10은 본 발명의 일 실시예에 따른 3차원 비디오 코덱을 개략적으로 나타내는 도면이다.

도 11은 본 발명의 일 실시예에 따른 병합 움직임 방법을 개략적으로 설명하기 위한 개념도이다.

도 12는 본 발명의 일 실시예에 따른 도 11의 병합 움직임 방법을 하드웨어로 구현한 일례를 나타내는 도면이다.

도 13은 본 발명의 일 실시예에 따라 도 11 및 도 12의 병합 움직임 후보 리스트 구성 방법을 설명하기 위한 개념도이다.

도 14는 본 발명의 일 실시예에 따른 확장 병합 움직임 후보 리스트를 구성하는 방법을 설명하기 위한 도면이다.

도 15는 본 발명의 다른 실시예에 따른 확장 병합 움직임 후보 리스트를 구성하는 방법을 설명하기 위한 도면이다.

도 16은 본 발명의 일 실시예에 따른 병합 움직임 후보 리스트를 구성하는 방법을 개략적으로 나타내는 순서도이다.

도 17a 내지 도 17f는 본 발명의 일 실시예에 따른 확장 병합 움직임 후보를 병합 움직임 후보 리스트에 추가하는 방법을 나타내는 순서도이다.

도 18은 본 발명의 일 실시예에 따라 복수의 시점을 포함하는 비디오 부호화/복호화 시 병합 움직임 후보 리스트를 구성하는 방법을 개략적으로 나타내는 순서도이다.

발명을 실시하기 위한 구체적인 내용

- [0017] 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 해당 설명을 생략할 수도 있다.
- [0018] 본 명세서에서 어떤 구성 요소가 다른 구성 요소에 “연결되어” 있다거나 “접속되어” 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있는 것을 의미할 수도 있고, 중간에 다른 구성 요소가 존재하는 것을 의미할 수도 있다. 아울러, 본 명세서에서 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.
- [0019] 제1, 제2 등의 용어는 다양한 구성들을 설명하는데 사용될 수 있지만, 상기 구성들은 상기 용어에 의해 한정되지 않는다. 상기 용어들은 하나의 구성을 다른 구성으로부터 구별하는 목적으로 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성은 제2 구성으로 명명될 수 있고, 유사하게 제2 구성도 제1 구성으로 명명될 수 있다.
- [0020] 또한 본 발명의 실시예에 나타나는 구성부들은 서로 다른 특징적인 기능을 나타내기 위해 독립적으로 도시되는 것으로, 각 구성부들이 분리된 하드웨어나 하나의 소프트웨어 구성 단위로 이루어짐을 의미하지 않는다. 즉, 각 구성부는 설명의 편의상 각각의 구성부로 나열하여 포함한 것으로 각 구성부 중 적어도 두 개의 구성부가 하나의 구성부를 이루거나, 하나의 구성부가 복수 개의 구성부로 나뉘어져 기능을 수행할 수 있다. 각 구성부의 통합된 실시예 및 분리된 실시예도 본 발명의 본질에서 벗어나지 않는 한 본 발명의 권리 범위에 포함된다.
- [0021] 또한, 일부의 구성 요소는 본 발명에서 본질적인 기능을 수행하는 필수적인 구성 요소는 아니고 단지 성능을 향상시키기 위한 선택적 구성 요소일 수 있다. 본 발명은 단지 성능 향상을 위해 사용되는 구성 요소를 제외한 본 발명의 본질을 구현하는데 필수적인 구성부만을 포함하여 구현될 수 있고, 단지 성능 향상을 위해 사용되는 선택적 구성 요소를 제외한 필수 구성 요소만을 포함한 구조도 본 발명의 권리범위에 포함된다.
- [0022] 도 1은 3차원 비디오 시스템의 기본 구조와 데이터 형식을 개략적으로 나타내는 일예이다. 도 1의 3차원 비디오 시스템은 3차원 비디오 표준에서 고려하고 있는 기본적인 3차원 비디오 시스템일 수 있다.
- [0023] 도 1을 참조하면, 3차원 비디오(3D video; 3DV) 시스템은 다시점(multi-view)의 영상 콘텐츠를 생성하는 송신측(sender)과 송신측으로부터 수신된 영상 콘텐츠를 복호화하여 다시점의 영상을 제공하는 수신측(receiver)을 포함할 수 있다.
- [0024] 송신측에서는 스테레오 카메라 및 다시점 카메라를 이용하여 비디오 정보를 생성하고, 깊이정보 카메라를 이용하여 깊이정보 맵(depth map, 또는 depth view)을 생성할 수 있다. 또한, 송신측에서는 변환기를 이용하여 2차원 영상을 3차원 영상으로 변환할 수 있다. 송신측에서는 상기 생성된 비디오 정보와 깊이정보 맵 등을 이용하여 $N(N \geq 2)$ 시점(view)의 영상 콘텐츠를 생성할 수 있다.
- [0025] N 시점의 영상 콘텐츠에는 N 시점의 비디오 정보와 그것의 깊이정보 맵(Depth-map) 정보 그리고 카메라 관련 부가정보 등이 포함될 수 있다. N 시점의 영상 콘텐츠는 3D 비디오 인코더에서 다시점 비디오 부호화 방법을 사용하여 압축될 수 있으며, 압축된 영상 콘텐츠(비트스트림)는 네트워크를 통해 수신측의 단말로 전송될 수 있다.
- [0026] 수신측에서는 전송 받은 비트스트림을 비디오 디코더(예를 들어, 3D 비디오 디코더, 스테레오 비디오 디코더, 2D 비디오 디코더 등)에서 다시점 비디오 복호화 방법을 사용하여 복호화하여 N 시점의 영상을 복원할 수 있다.
- [0027] 복원된 N 시점의 영상은 깊이정보 맵 기반 렌더링(DIBR; Depth-Image-Based Rendering) 과정에 의해 N 시점 이상의 가상시점 영상들로 생성될 수 있다. 생성된 N 시점 이상의 가상시점 영상들은 다양한 입체 디스플레이 장치(예를 들어, N -시점 디스플레이, 스테레오 디스플레이, 2D 디스플레이 등)에 맞게 재생되어 사용자에게 입체감이 있는 영상을 제공하게 된다.

- [0028] 도 2는 “balloons” 영상의 실제 영상과 깊이정보 맵 영상의 일례를 나타내는 도면이다.
- [0029] 깊이정보 맵(depth map)은 가상시점 영상을 생성하는데 사용되며, 실세계에서 카메라와 실제 사물(object) 간의 거리(실사 영상과 동일한 해상도로 각 화소에 해당하는 깊이정보)를 일정한 비트수로 표현한 것이다.
- [0030] 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 깊이정보 맵 영상을 나타낸 것이다. 도 2의 (b)에 도시된 깊이정보 맵 영상은 화면에 보이는 깊이 정보를 화소당 8비트로 표현한 것이다.
- [0031] 실제 영상과 그것의 깊이정보 맵을 부호화하는 방법은, 예컨대 H.264/AVC(MPEG-4 Part 10 Advanced Video Coding)를 이용할 수 있으며, 또는 MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)에서 공동으로 표준화를 진행한 HEVC(High Efficiency Video Coding) 국제 동영상 표준을 사용할 수도 있다.
- [0032] 실제 영상과 그것의 깊이정보 맵은 하나의 카메라뿐만 아니라 여러 개의 카메라에서 획득된 영상일 수 있다. 여러 개의 카메라에서 획득된 영상은 독립적으로 부호화될 수 있으며, 일반적인 2차원 비디오 부호화 코덱을 사용하여 부호화될 수 있다. 또한, 여러 개의 카메라에서 획득된 영상은 시점 간의 상관성이 존재하므로, 여러 개의 카메라에서 획득된 영상은 부호화 효율을 높이기 위하여 서로 다른 시점간 예측을 사용하여 부호화될 수 있다.
- [0033] 도 3은 3D 비디오 코덱(codec)에서 시점간 예측(inter view prediction)의 구조를 나타내는 일례이다.
- [0034] 도 3을 참조하면, 시점 1(View 1)은 시점 0(View 0)을 기준으로 왼쪽에 위치한 카메라에서 획득한 영상이고, 시점 2(View 2)는 시점 0(View 0)을 기준으로 오른쪽에 위치한 카메라에서 획득한 영상이다.
- [0035] 또한, 시점 1(View 1)과 시점 2(View 2)는 시점 0(View 0)을 참조 영상으로 사용하여 시점간 예측을 수행하며, 부호화 순서는 시점 1(View 1)과 시점 2(View 2)보다 시점 0(View 0)이 먼저 부호화되어야 한다.
- [0036] 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 시점간 예측을 수행하여야 하므로, 시점간 예측 과정이 포함된 3차원 비디오 코덱을 사용하여 부호화될 수 있다.
- [0037] 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 깊이정보 맵을 이용하여 부호화될 수 있다. 예를 들어, 실제 영상과 그것의 깊이정보 맵을 부호화할 때, 실제 영상과 깊이정보 맵은 서로 독립적으로 부호화/복호화될 수 있다. 또는 실제 영상과 깊이정보 맵을 부호화할 때, 실제 영상과 깊이정보 맵은 도 4와 같이 서로 의존적으로 부호화/복호화될 수 있다.
- [0038] 도 4는 3D 비디오 인코더/디코더에서 실제 영상(texture view)과 깊이정보 맵(depth view)을 부호화/복호화하는 과정을 나타낸 일례이다.
- [0039] 도 4를 참조하면, 3D 비디오 인코더는 실제 영상(texture view)을 부호화하는 실제 영상 인코더(texture encoder)와 깊이정보 맵(depth view)을 부호화하는 깊이정보 맵 인코더(depth encoder)를 포함할 수 있다.
- [0040] 예를 들어, 실제 영상 인코더는 깊이정보 맵 인코더에 의해 이미 부호화된 깊이정보 맵을 이용하여 실제 영상을 부호화할 수 있다. 반대로, 깊이정보 맵 인코더는 실제 영상 인코더에 의해 이미 부호화된 실제 영상을 이용하여 깊이정보 맵을 부호화할 수 있다.
- [0041] 3D 비디오 디코더는 실제 영상을 복호화하는 실제 영상 디코더(texture decoder)와 깊이정보 맵을 복호화하는 깊이정보 맵 디코더(depth decoder)를 포함할 수 있다.
- [0042] 예를 들어, 실제 영상 디코더는 깊이정보 맵 디코더에 의해 이미 복호화된 깊이정보 맵을 이용하여 실제 영상을 복호화할 수 있다. 반대로, 깊이정보 맵 디코더는 실제 영상 디코더에 의해 이미 복호화된 실제 영상을 이용하여 깊이정보 맵을 복호화할 수 있다.

- [0043] 도 5는 3D 비디오 코덱의 예측 구조의 일례를 나타낸다.
- [0044] 도 5는, 설명의 편의 상, 3대의 카메라에서 획득한 실제 영상과 상기 실제 영상에 대한 깊이정보 맵을 부호화하기 위한 부호화 예측 구조를 나타낸 것이다.
- [0045] 도 5를 참조하면, 3대의 카메라에서 획득된 3개의 실제 영상은 시점에 따라 T0, T1, T2로 나타내었고, 실제 영상과 동일한 위치의 3개의 깊이정보 맵은 시점에 따라 D0, D1, D2로 나타내었다. 여기서, T0와 D0는 시점 0(View 0)에서 획득한 영상이며, T1와 D1는 시점 1(View 1)에서 획득한 영상이며, T2와 D2는 시점 2(View 2)에서 획득한 영상이다.
- [0046] 도 5에 도시된 사각형은 영상(픽처)를 나타낸다.
- [0047] 각 영상(픽처)는 부호화 타입에 따라 I 픽처(Intra Picture), P 픽처(Uni-prediction Picture), B 픽처(Bi-prediction Picture)로 나뉘어지며, 각 픽처의 부호화 타입에 따라 부호화될 수 있다. I 픽처는 픽처간 예측 없이 영상 자체를 부호화하고, P 픽처는 순방향으로만 참조 영상을 이용하여 픽처간 예측 부호화하고, B 픽처는 순방향과 역방향 양측으로 참조 영상을 이용하여 픽처간 예측 부호화한다.
- [0048] 도 5에서 화살표는 예측 방향을 나타낸다. 즉, 예측 방향에 따라 실제 영상과 그것의 깊이정보 맵은 서로 의존적으로 부호화/복호화될 수 있다.
- [0049] 실제 영상에서 현재 블록을 예측하기 위해서, 현재 블록의 움직임 정보를 유추하기 위한 방법은 크게 시간적 예측(temporal prediction)과 시점간 예측(inter view prediction)으로 나뉜다. 시간적 예측은 동일한 시점 내에서 시간적 상관성을 이용한 예측 방법이고, 시점간 예측은 인접한 시점에서 시점간 상관성을 이용한 예측 방법이다. 이러한 시간적 예측과 시점간 예측은 한 픽처에서 서로 혼용되어 사용될 수 있다.
- [0050] 여기서, 현재 블록은 실제 영상 내에서 현재 예측이 수행되는 블록을 말한다. 움직임 정보는 움직임 벡터만을 의미할 수 있으며, 혹은 움직임 벡터, 참조 영상 번호, 단방향 예측인지 양방향 예측인지, 시점간 예측인지 시간적 예측인지, 또 다른 예측인지를 의미할 수도 있다.
- [0051] 한편, 대용량의 3차원 영상 콘텐츠는 비트스트림의 양을 감소시키기 위하여 효율적으로 압축되어야 한다. 부호화 효율을 증가시키기 위하여 서로 다른 시점간 상관성을 이용할 수 있으며, 일반 영상(Texture View)과 깊이정보 맵(Depth View, 혹은 Depth Map) 간의 상관성을 이용할 수 있다. 하지만 이로 인해, 2차원 영상을 부호화할 때보다 더 많은 부호화 알고리즘이 요구되며, 이를 구현하기 위한 하드웨어 및 소프트웨어의 구현 복잡도가 증가되고 계산 복잡도가 증가되는 문제점이 존재한다.
- [0052] 도 6은 3차원 비디오 코덱(3D Video Codec)의 인코더를 개략적으로 나타낸 구조도이다.
- [0053] 도 6을 참조하면, 3차원 비디오 코덱(600)은 서로 다른 시점 영상들(예를 들어, 시점 0(view 0), 시점 1(view 1), 시점 2(view 2))을 입력으로 받아 부호화하고, 부호화된 통합된 하나의 비트스트림을 출력할 수 있다.
- [0054] 이때, 영상들에는 일반 영상(Texture View)뿐만 아니라 깊이정보 맵(Depth View)도 포함될 수 있다.
- [0055] 3차원 비디오 코덱(600)은 시점 정보(View ID 정보)에 따라 서로 다른 인코더에서 입력 영상들을 부호화할 수 있다.
- [0056] 예를 들어, 시점 0(View 0)의 영상은 역방향 호환성(Backward compatibility)를 위하여 기존 2차원 비디오 코덱으로 부호화되어야 하므로, 기본 계층 인코더(610, 시점 0 인코더)로 부호화될 수 있다. 시점 1(View 1)과 시점 2(View 2)의 영상들은 시점간 예측 알고리즘과 일반 영상과 깊이정보 맵 간의 상관성을 이용한 알고리즘이 포함된 3차원 비디오 코덱으로 부호화되어야 하므로, 향상 계층(enhancement layer) 인코더(620, 시점 1 혹은 시점 2 인코더)로 부호화될 수 있다.
- [0057] 또한, 일반 영상이 아닌 깊이정보 맵의 경우, 일반 영상의 부호화된 정보를 활용하여 부호화될 수 있으므로, 향상 계층 인코더(620)로 부호화될 수 있다. 따라서, 시점 0(View 0)의 영상을 부호화할 때에 비해, 시점 1(View 1)과 시점 2(View 2)의 영상을 부호화할 때 더욱 복잡한 인코더가 요구되며, 또한 기본 계층(base layer)의 일반 영상을 부호화할 때에 비해 깊이정보 맵을 부호화할 때 더욱 복잡한 인코더가 요구된다.

- [0058] 한편, HEVC에서는 영상 부호화/복호화 시 인터 예측(inter prediction)에 사용되는 움직임 정보의 부호화 방법 중 하나로 병합 움직임(merge, 혹은 merge motion) 방법을 이용한다. 여기서, 향상 계층(enhancement layer)에서의 부호화 효율을 증가시키기 위해서, 향상 계층에서는 기본 계층(base layer)에서의 병합 움직임 방법을 수정하여 향상된 병합 움직임 방법을 사용한다.
- [0059] 도 7은 HEVC 기반 3차원 비디오 코덱(3D-HEVC)에서 사용되는 병합 움직임 방법을 설명하기 위해 도시된 도면이다.
- [0060] 도 7을 참조하면, 3D-HEVC(700)에서는 시점 0(View 0)에 대한 병합 움직임 구성 방법(710)과 나머지 다른 시점들(시점 1(View 1)과 시점 2(View 2))에 대한 병합 움직임 구성 방법(720)으로 서로 구분되어 수행된다.
- [0061] 현재 PU(Prediction Unit, 혹은 PB(Prediction Block), 혹은 임의 크기의 블록)를 포함하는 영상이 3D-HEVC(700)에 입력되면, 3D-HEVC(700)는 입력 영상이 일반 영상인지 혹은 깊이정보 맵 영상인지에 대한 정보(Texture/Depth 정보)와 입력 영상의 시점 정보(ViewID 정보)를 기반으로, 시점 0(View 0)에 대한 병합 움직임 구성 방법(710)과 나머지 다른 시점들(시점 1(View 1)과 시점 2(View 2))에 대한 병합 움직임 구성 방법(720) 중 하나를 선택할 수 있다. 그리고 3D-HEVC(700)는 선택된 병합 움직임 구성 방법을 이용하여 현재 PU에 대한 병합 움직임 후보 리스트(merge candidate list)를 출력할 수 있다.
- [0062] 여기서, 현재 PU는 현재 영상의 부호화/복호화를 위해 현재 영상 내 예측이 수행되는 현재 블록을 말한다.
- [0063] 시점 0(View 0)에 대한 일반 영상은 역방향 호환성을 위해 기본 계층을 위한 병합 움직임 구성 방법을 이용하여 병합 움직임 후보 리스트를 구성한다. 반면, 시점 1(View 1)과 시점 2(View 2)에 대한 일반 영상과 깊이정보 맵은 향상 계층을 위한 병합 움직임 구성 방법을 이용하여 병합 움직임 후보 리스트를 구성한다.
- [0064] 향상 계층을 위한 병합 움직임 구성 방법은 기본 계층을 위한 병합 움직임 구성 방법에 새로운 후보를 추가하거나 후보 리스트 순서를 수정하는 방법으로 수행된다. 즉, 도 7에 도시된 바와 같이, 향상 계층(나머지 다른 시점들(시점 1(View 1)과 시점 2(View 2))과 깊이정보 맵)을 위한 병합 움직임 구성 방법에는 기본 계층을 위한 병합 움직임 구성 방법이 포함되어 있다.
- [0065] 따라서, 향상 계층을 위한 병합 움직임 구성 방법은 기본 계층을 위한 병합 움직임 구성보다 복잡하게 구성되어 있으며, 계산 복잡도가 크다는 것을 알 수 있다. 또한, 하드웨어 혹은 소프트웨어의 구현 측면에서도 기본 계층을 위한 병합 움직임 구성 방법과 향상 계층을 위한 병합 움직임 구성 방법을 모두 구현해주어야 하므로, 구현 복잡도가 2배 이상 증가하는 것을 알 수 있다.
- [0066] 도 8은 현재 블록에 대한 병합 움직임 리스트를 구성하는데 사용되는 주변 블록들을 나타낸 일례이다.
- [0067] 병합 움직임 방법(merge mode)은 현재 블록(현재 PU)의 움직임 정보(예를 들어, 움직임 벡터, 참조 픽처 리스트, 참조 픽처 인덱스 등)로서 현재 블록의 주변 블록의 움직임 정보를 이용하는 방법을 말하며, 주변 블록의 움직임 정보를 기반으로 현재 블록에 대한 병합 움직임 후보 리스트를 구성한다.
- [0068] 주변 블록은 도 8에 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 동일 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 결정한다.
- [0069] 병합 움직임 후보 리스트를 구성하기 위해서, 먼저 주변 블록(A, B, C, D, E)과 동일 위치의 후보 블록(H 혹은 M)의 움직임 정보가 현재 블록의 병합 움직임 후보 리스트를 구성하는 병합 움직임 후보(merge candidate)로 이용될 수 있는지를 판단하고, 다음으로 이용 가능한 블록의 움직임 정보를 병합 움직임 후보로 결정한다. 그리고 병합 움직임 후보를 병합 움직임 후보 리스트에 추가할 수 있다.
- [0070] 도 9는 병합 움직임 후보 리스트를 구성하는 방법을 하드웨어로 구현한 일례를 나타내는 도면이다.
- [0071] 도 9를 참조하면, 시점 0(View 0)에 대한 일반 영상에서 사용되는 병합 움직임 리스트를 구성하기 위한 입력 파

라미터와, 시점 1(View 1)과 시점 2(View 2)에 대한 일반 영상과 깊이정보 맵에서 사용되는 병합 움직임 리스트를 구성하기 위한 입력 파라미터는 동일하다. 단지 다른 점은 시점 1(View 1)과 시점 2(View 2)에 대한 일반 영상과 깊이정보 맵에서 사용되는 병합 움직임 리스트를 구성하기 위한 입력 파라미터(“추가 움직임 F”와 “추가 움직임 G”)가 추가된다는 점이다.

[0072] 따라서, 추가된 움직임 정보들로 인해 병합 움직임 후보 리스트를 구성하는 부분이 변경된다. 즉, 추가된 움직임 정보들을 병합 움직임 후보 리스트에 포함시키기 위해서(부호화 효율을 증가시키기 위해서), 시점 1(View 1)과 시점 2(View 2)에 대한 일반 영상과 깊이정보 맵을 위한 병합 움직임 리스트 구성 모듈을 새로 구현하여야 한다. 이는 하드웨어의 구현 복잡도를 증가시킬 수 있다.

[0073] 이러한 문제점들을 해결하기 위해서, 본 발명에서는 항상 계층(실시 예로, 시점 1(View 1)과 시점 2(View 2)에 대한 일반 영상과 깊이정보 맵)을 위한 부호화 알고리즘 및 비디오 코덱의 구현 복잡도 및 계산 복잡도를 감소시키는 방법을 제안한다. 일례로, 본 발명에서는 이미 하드웨어 칩 형태로 구현이 완료된 기본 계층(시점0(View 0)에 대한 일반 영상)을 위한 “병합 움직임 후보 리스트 구성” 모듈을 그대로 재사용하여 항상 계층(실시 예로, 시점 1(View 1)과 시점 2(View 2)에 대한 일반 영상과 깊이정보 맵)에 적용함으로써, 하드웨어 구현 복잡도를 감소시킬 수 있다. 본 발명에 따르면, 2차원 영상 서비스에 사용되는 기본 계층을 위한 부호화기/복화화기(구체적으로, “병합 움직임 후보 리스트 구성” 모듈)를 가지고 있는 소비자가 3차원 영상 서비스를 받고자 한다면, 단지 추가적인 모듈(구체적으로, 항상 계층을 위한 “병합 움직임 후보 리스트 구성” 모듈)만 부착하면 손쉽게 3차원 영상 서비스를 받을 수 있게 된다.

[0074] 이하, 본 발명에서는 비디오 코덱의 구현 복잡도 및 계산 복잡도를 감소시켜서 데이터 처리량을 향상시키는 방법에 대해 설명한다.

[0075] [기본 방법]

[0076] 도 10은 본 발명의 일 실시예에 따른 3차원 비디오 코덱을 개략적으로 나타내는 도면이다.

[0077] 도 10을 참조하면, 3차원 비디오 코덱(1000)은 서로 다른 시점 영상들(예를 들어, 시점 0(view 0), 시점 1(view 1), 시점 2(view 2))을 입력으로 받아 부호화하고, 부호화된 하나의 비트스트림을 출력할 수 있다.

[0078] 이때, 영상들에는 일반 영상(Texture View)뿐만 아니라 깊이정보 맵(Depth View)도 포함될 수 있다. 영상들에는 다른 시점과 상관없이 독립적으로 부호화될 수 있는 독립적인 시점(Independent View)의 영상과, 독립적인 시점의 영상을 참조 영상으로 사용하여 부호화되는 의존적인 시점(Dependent View)의 영상을 포함할 수 있다. 예를 들어, 시점 0(View 0)은 독립적인 시점일 수 있으며, 시점 1(view 1)과 시점 2(view 2)는 시점 0(View 0)을 참조하여 부호화되는 의존적인 시점일 수 있다.

[0079] 3차원 비디오 코덱(1000)은 모든 시점(예를 들어, 시점 0, 시점 1 및 시점2)에 대한 일반 영상 및 깊이정보 맵을 부호화할 수 있는 인코더(부호화기)(1010)를 포함할 수 있다. 예컨대, 모든 시점에 대한 일반 영상 및 깊이정보 맵을 부호화할 수 있는 인코더(1010)는 MPEG-1, MPEG-2, MPEG-4 Part 2 Visual, H.264/AVC, VC-1, AVS, KTA, HEVC(H.265/HEVC) 등이 될 수 있다.

[0080] 3차원 비디오 코덱(1000)은 독립적인 시점이 아닌 의존적인 시점에 대한 일반 영상 및 깊이정보 맵에 대하여, 부호화 효율을 증가시키기 위하여, 부분 인코더(1020)를 포함할 수 있다. 예컨대, 부분 인코더(1020)는 시점 1(view 1)과 시점 2(view 2)에 대한 일반 영상 및 깊이정보 맵을 부호화할 수 있으며, 또는 모든 시점에 대한 깊이정보 맵을 부호화할 수 있다.

[0081] 3차원 비디오 코덱(1000)은 각 인코더(1010, 1020)에 의해 부호화된 영상들을 다중화하는 다중화기(1030)를 포함할 수 있다. 다중화기(1030)는 시점 0(View 0)의 일반 영상에 대한 비트스트림과, 다른 시점(시점 1(View 1)과 시점 2(View 2))들의 일반 영상 및 깊이정보 맵들에 대한 비트스트림을 다중화하여 하나의 비트스트림으로 출력할 수 있다.

[0082] 상술한 바와 같이, 본 발명의 일 실시예에 따른 3차원 비디오 코덱(1000)은 역방향 호환성을 제공하는 독립적인 시점(예컨대, 시점 0(View 0))에 대한 일반 영상의 부호화에 사용되는 모듈(1010)을 의존적인 시점(예컨대, 시점 1(View 1)과 시점 2(View 2))에 대한 일반 영상과 깊이정보 맵들에 그대로 적용함으로써, 구현 복잡도를 감소시킬 수 있다. 그리고, 본 발명의 일 실시예에 따른 3차원 비디오 코덱(1000)은 의존적인 시점(예컨대, 시점 1(View 1)과 시점 2(View 2))에 대한 일반 영상과 깊이정보 맵들에 부분 인코더(1020)를 추가적으로 적용함으로

써, 부호화 효율을 향상시킬 수 있다.

- [0083] 도 10에서 상술한 3차원 비디오 코덱은 부호화/복호화 전체 과정에 대하여 적용할 수 있으며, 부호화/복호화의 각 단계마다 각각 적용할 수도 있다.
- [0084] [세부 방법]
- [0085] 도 11은 본 발명의 일 실시예에 따른 병합 움직임 방법을 개략적으로 설명하기 위한 개념도이다. 도 11에 도시된 병합 움직임 방법은 HEVC 기반 3차원 비디오 코덱(3D-HEVC)에 의해 수행될 수 있으며, 3D-HEVC은 상술한 도 10의 3차원 비디오 코덱을 기반으로 구현될 수 있다.
- [0086] 본 발명의 일 실시예에 따른 병합 움직임 방법은 현재 PU에 대한 공간적 병합 움직임 후보(spatial merge candidate)와 시간적 병합 움직임 후보(temporal merge candidate)를 유도하고, 그리고 현재 PU에 대한 정보(예컨대, 현재 PU의 시점 정보, 현재 PU의 영상 타입 정보 등)를 기반으로 추가 병합 움직임 후보를 추가적으로 유도하고, 상기 유도된 병합 움직임 후보들을 기반으로 현재 PU에 대한 병합 움직임 후보 리스트(merge candidate list)를 구성할 수 있다.
- [0087] 도 11을 참조하면, 본 발명의 일 실시예에 따른 병합 움직임 방법에서, 입력은 현재 PU 정보(혹은 현재 영상 정보), 현재 PU의 영상이 일반 영상인지 혹은 깊이정보 맵 영상인지에 대한 정보(Texture/Depth 정보), 현재 PU의 시점 정보(ViewID 정보)이며, 출력은 현재 PU에 대한 병합 움직임 후보 리스트이다.
- [0088] 본 발명의 병합 움직임 방법은 현재 PU에 대하여 “기본 병합 움직임 리스트 구성” (1110) 단계가 기본적으로 수행되어 “기본 병합 움직임 후보 리스트”를 출력할 수 있다. 일례로, “기본 병합 움직임 리스트 구성” (1110)은 HEVC에서의 병합 움직임 후보 리스트 구성 방법을 그대로 사용할 수 있다.
- [0089] 다음으로, 본 발명의 병합 움직임 방법은 현재 PU의 영상이 일반 영상인지 깊이정보 맵 영상인지에 대한 정보(Texture/Depth 정보)와, 현재 PU에 대한 시점 정보(ViewID 정보)에 따라 “추가 병합 움직임 리스트 구성” (1120) 단계가 추가적으로 수행될 수 있다. 이때, “추가 병합 움직임 리스트 구성” (1120) 단계에서, 입력은 “기본 병합 움직임 리스트 구성” (1110) 단계에서 출력된 “기본 병합 움직임 후보 리스트”이며, 출력은 “확장 병합 움직임 후보 리스트”이다. “추가 병합 움직임 리스트 구성” (1120) 단계는 의존적인 시점(예컨대, 시점 1(View 1)과 시점 2(View 2))에 대한 일반 영상과 깊이정보 맵들에 대해 수행될 수 있다.
- [0090] 본 발명에 따른 병합 움직임 방법에 대한 보다 구체적인 내용은 후술하도록 한다.
- [0091] 도 12는 본 발명의 일 실시예에 따른 도 11의 병합 움직임 방법을 하드웨어로 구현한 일례를 나타내는 도면이다.
- [0092] 도 12를 참조하면, 본 발명의 일 실시예에 따른 병합 움직임 방법을 수행하는 장치(이하, 병합 움직임 장치)(1200)는 기본 병합 움직임 리스트 구성 모듈(1210)과 추가 병합 움직임 리스트 구성 모듈(1220)을 포함한다.
- [0093] 병합 움직임 장치(1200)의 입력은 공간적 병합 움직임 후보, 시간적 병합 움직임 후보, 추가 병합 움직임 후보이다. 병합 움직임 장치(1200)의 출력은 독립적인 시점에 대한 일반 영상의 경우에는 기본 병합 움직임 후보 리스트이고, 의존적인 시점에 대한 일반 영상과 깊이정보 맵의 경우에는 확장 병합 움직임 후보 리스트이다.
- [0094] 상술한 바와 같이, 독립적인 시점은 다른 시점과 상관없이 독립적으로 부호화될 수 있는 시점을 말하며, 기본 시점(base view)일 수 있다. 의존적인 시점은 독립적인 시점을 참조하여 부호화되는 시점을 말한다. 본 명세서에서는 설명의 편의를 위하여, 예컨대, 독립적인 시점은 시점 0(View 0)일 수 있으며, 의존적인 시점은 시점 1(view 1)과 시점 2(view 2)를 포함하는 것으로 기술하였다.
- [0095] 기본 병합 움직임 리스트 구성 모듈(1210)은 현재 PU에 대한 공간적 병합 움직임 후보 및 시간적 병합 움직임 후보를 유도하여 기본 병합 움직임 후보 리스트를 구성할 수 있다.
- [0096] 공간적 병합 움직임 후보는, 도 8에 도시된 바와 같이, 현재 PU와 공간적으로 인접한 주변 블록(A, B, C, D, E)으로부터 유도될 수 있다.
- [0097] 기본 병합 움직임 리스트 구성 모듈(1210)은 주변 블록(A, B, C, D, E)이 가용한지(available) 여부를

판단하고, 가용한 주변 블록의 움직임 정보를 현재 PU에 대한 공간적 병합 움직임 후보로 결정할 수 있다. 이때, 주변 블록(A, B, C, D, E)이 가용한지 여부를 판단할 때, 미리 정해진 순서 혹은 임의의 순서에 따라 주변 블록(A, B, C, D, E)의 가용성 여부를 판단할 수 있다. 예컨대, A, B, C, D, E 순으로 진행될 수 있다.

[0098] 시간적 병합 움직임 후보는, 도 8에 도시된 바와 같이, 현재 PU에 대한 동일 위치의 픽처(co-located picture; col picture) 내 동일 위치의 블록(co-located block; col block)(H, M)으로부터 유도될 수 있다. 동일 위치의 블록 H는 동일 위치의 픽처 내 현재 PU와 대응하는 위치의 블록(X')을 기준으로 하단 우측(bottom right)에 위치하는 PU 블록일 수 있다. 동일 위치의 블록 M은 동일 위치의 픽처 내 현재 PU와 대응하는 위치의 블록(X')을 기준으로 X' 블록의 중앙에 위치하는 PU 블록일 수 있다.

[0099] 기본 병합 움직임 리스트 구성 모듈(1210)은 동일 위치의 블록(H, M)이 가용한지 여부를 판단하고, 가용한 동일 위치의 블록의 움직임 정보를 현재 PU에 대한 시간적 병합 움직임 후보로 결정할 수 있다. 이때, 동일 위치의 블록(H, M)의 가용성 여부를 판단하는 순서는 H 블록, M 블록 순일 수 있고, 또는 그 반대 순서일 수도 있다.

[0100] 추가 병합 움직임 리스트 구성 모듈(1220)은 현재 PU의 영상이 일반 영상인지 혹은 깊이정보 맵 영상인지에 대한 정보(Texture/Depth 정보), 현재 PU에 대한 시점 정보(ViewID 정보)를 기반으로 현재 PU에 대한 추가 병합 움직임 후보를 유도하여 확장 병합 움직임 후보 리스트를 구성할 수 있다.

[0101] 현재 PU의 영상이 의존적인 시점(예컨대, 시점 1(view 1)과 시점 2(view 2))에 대한 일반 영상과 깊이정보 맵일 경우, 추가 병합 움직임 리스트 구성 모듈(1220)은 추가적으로 현재 PU의 의존적인 시점에 대한 일반 영상과 깊이정보 맵을 위한 병합 움직임 후보 리스트를 구성하는 과정을 수행할 수 있다.

[0102] 이때, 추가 병합 움직임 리스트 구성 모듈(1220)의 입력은, 기본 병합 움직임 리스트 구성 모듈(1210)에 의해 구성된 기본 병합 움직임 후보 리스트와 추가 병합 움직임 후보(F, G)이다. 추가 병합 움직임 리스트 구성 모듈(1220)의 출력은 확장 병합 움직임 후보 리스트이다.

[0103] 의존적인 시점(예컨대, 시점 1(view 1)과 시점 2(view 2))에 대한 일반 영상과 깊이정보 맵을 위한 병합 움직임 후보 리스트를 구성하기 위해서, 상술한 바와 같이 본 발명의 실시예에 따른 병합 움직임 장치는, 새로운 모듈을 구현하지 않고 단지 추가적인 부분 모듈만 구현함으로써 하드웨어의 구현 복잡도를 감소시킬 수 있다. 즉, 이미 하드웨어 칩 형태로 구현이 완료된 기본 계층(예컨대, 시점0(View 0)에 대한 일반 영상)을 위한 “병합 움직임 후보 리스트 구성” 모듈을 그대로 재사용하여 항상 계층(예컨대, 시점 1(View 1)과 시점 2(View 2)에 대한 일반 영상과 깊이정보 맵)에 적용함으로써, 하드웨어 구현 복잡도를 감소시킬 수 있다.

[0104] 도 13은 본 발명의 일 실시예에 따라 도 11 및 도 12의 병합 움직임 후보 리스트 구성 방법을 설명하기 위한 개념도이다. 도 13의 병합 움직임 후보 리스트 구성 방법은 상술한 도 10의 3차원 비디오 코덱 혹은 도 12의 병합 움직임 장치에서 수행될 수 있다.

[0105] 도 13을 참조하면, 본 발명의 실시예에 따른 병합 움직임 후보 리스트를 구성하기 위해서, 입력은 현재 PU 정보, 현재 PU의 영상이 일반 영상인지 깊이정보 맵 영상인지에 대한 정보(Texture/Depth 정보), 현재 PU에 대한 시점 정보(ViewID 정보)이며, 출력은 현재 PU에 대한 병합 움직임 후보 리스트이다.

[0106] 우선, 현재 PU에 대하여 기본 병합 움직임 후보 리스트(1310)가 구성된다. 예를 들어, 기본 병합 움직임 후보 리스트는 HEVC에서의 병합 움직임 후보 리스트 구성 방법을 그대로 사용할 수 있으며, 상술한 바와 같이 현재 PU에 대한 공간적 병합 움직임 후보 및 시간적 병합 움직임 후보를 기반으로 구성될 수 있다.

[0107] 다음으로, 현재 PU 영상이 일반 영상인지 깊이정보 맵 영상인지에 대한 정보(Texture/Depth 정보) 및 현재 PU 영상에 대한 시점 정보(ViewID 정보)를 기반으로 확장 병합 움직임 후보 리스트(1320)가 구성된다. 이때, 확장 병합 움직임 후보 리스트는 의존적인 시점(예컨대, 시점 1(view 1)과 시점 2(view 2))에 대한 일반 영상과 깊이정보 맵들에 대하여 구성될 수 있으며, 상술한 바와 같이 추가 병합 움직임 후보가 추가될 수 있다.

[0108] 만일, 현재 PU가 독립적인 시점(예컨대, 시점0(View 0))의 일반 영상일 경우, 기본 병합 움직임 후보 리스트가 출력될 수 있다. 그렇지 않고, 현재 PU가 의존적인 시점(예컨대, 시점 1(view 1)과 시점 2(view 2))에 대한 일반 영상과 깊이정보 맵들일 경우, 확장 병합 움직임 후보 리스트가 출력될 수 있다.

[0109] 이때, 확장 병합 움직임 후보 리스트의 후보 개수는 기본 병합 움직임 후보 리스트의 후보 개수보다 많을 수 있다.

- [0110] 도 14는 본 발명의 일 실시예에 따른 확장 병합 움직임 후보 리스트를 구성하는 방법을 설명하기 위한 도면이다.
- [0111] 도 14를 참조하면, 본 발명의 실시예에 따른 확장 병합 움직임 후보 리스트는, 추가적인 움직임 정보인 추가 병합 움직임 후보(예컨대, 움직임 정보 F)를 확장 병합 움직임 후보 리스트의 첫번째 항목(index)(혹은 임의의 위치에 해당하는 항목)에 삽입할 수 있다.
- [0112] 이때, 추가 병합 움직임을 후보를 삽입하기 전에, 추가 병합 움직임 후보(예컨대, 움직임 정보 F)와 기본 병합 움직임 후보 리스트의 첫번째 병합 움직임 후보(예컨대, 움직임 정보 A)를 서로 비교하고, 두 후보가 동일하지 않을 경우에 확장 병합 움직임 후보 리스트의 첫번째 항목에 추가 병합 움직임 후보(예컨대, 움직임 정보 F)를 삽입할 수 있으며, 그 역도 가능하다. 예를 들어, 두 후보(예컨대, 움직임 정보 F와 A)의 움직임 정보를 비교할 때, 두 후보의 움직임 벡터의 차이가 임의의 임계값 이내일 경우에는 추가 병합 움직임 후보(예컨대, 움직임 정보 F)를 확장 병합 움직임 후보 리스트에 삽입하지 않을 수 있으며, 그 역도 가능하다. 또는, 두 후보의 참조 영상이 동일하지 않을 경우에는 추가 병합 움직임 후보(예컨대, 움직임 정보 F)를 확장 병합 움직임 후보 리스트에 삽입할 수 있으며, 그 역도 가능하다.
- [0113] 도 15는 본 발명의 다른 실시예에 따른 확장 병합 움직임 후보 리스트를 구성하는 방법을 설명하기 위한 도면이다.
- [0114] 도 15를 참조하면, 본 발명의 실시예에 따른 확장 병합 움직임 후보 리스트는, 추가적인 움직임 정보인 추가 병합 움직임 후보(예컨대, 움직임 정보 F)를 확장 병합 움직임 후보 리스트의 첫번째 항목에 삽입하고, 또 다른 추가적인 움직임 정보인 추가 병합 움직임 후보(예컨대, 움직임 정보 G)를 확장 병합 움직임 후보 리스트의 세번째 항목(혹은 임의의 위치에 해당하는 항목)에 삽입할 수 있다.
- [0115] 이때, 추가 병합 움직임을 후보를 삽입하기 전에, 기본 병합 움직임 후보 리스트 내의 원래 항목(첫번째 항목 및 세번째 항목)과 추가 병합 움직임 후보(예컨대, 움직임 정보 F와 G)를 서로 비교하고, 두 후보(예컨대, 움직임 정보 A와 F, 혹은 움직임 정보 C와 G)가 동일하지 않을 경우에 확장 병합 움직임 후보 리스트의 첫번째 항목과 세번째 항목에 추가 병합 움직임을 후보를 삽입할 수 있으며, 그 역도 가능하다. 예를 들어, 두 후보(예컨대, 움직임 정보 A와 F, 혹은 움직임 정보 C와 G)의 움직임 정보를 비교할 때, 두 후보의 움직임 벡터의 차이가 임의의 임계값 이내일 경우에는 추가 병합 움직임 후보(예컨대, 움직임 정보 F 혹은 G)를 확장 병합 움직임 후보 리스트에 삽입하지 않을 수 있으며, 그 역도 가능하다. 또는, 두 후보의 참조 영상이 동일하지 않을 경우에는 추가 병합 움직임 후보(예컨대, 움직임 정보 F 혹은 G)를 확장 병합 움직임 후보 리스트에 삽입할 수 있으며, 그 역도 가능하다.
- [0116] [추가 방법들]
- [0117] 상술한 도 10 내지 도 13의 방법들은 다음과 같이 다양하게 적용될 수 있다.
- [0118] 1. 일 실시예로, 기본 인코더(혹은 기본 모듈)를 시점 0(View 0)에 대한 일반 영상뿐만 아니라 시점 1(View 1)과 시점 2(View 2)에 대한 일반 영상과 깊이정보 맵들에 적용할 수 있다.
- [0119] 2. 다른 실시예로, 기본 인코더(혹은 기본 모듈)를 복잡도가 높은 작은 블록단위(일예로, 8x8 단위 혹은 임의의 블록 크기)에만 적용할 수 있다. 이때, 시점 1(View 1)과 시점 2(View 2)에 대한 일반 영상과 깊이정보 맵들에 대하여, 작은 블록 크기 이하에서는 기본 인코더(혹은 기본 모듈)를 사용하여 부호화되고, 작은 블록 크기보다 클 경우에는 기본 인코더(혹은 기본 모듈)와 부분 인코더(혹은 확장 모듈)를 사용하여 부호화될 수 있다. 여기서, 기본 인코더(혹은 기본 모듈)은 도 11과 도 13에서 “기본 병합 움직임 리스트 구성” 단계를 수행할 수 있으며, 부분 인코더(혹은 확장 모듈)은 도 11과 도 13에서 “추가 병합 움직임 리스트 구성” 단계를 수행할 수 있다.
- [0120] 도 16은 본 발명의 일 실시예에 따른 병합 움직임 후보 리스트를 구성하는 방법을 개략적으로 나타내는 순서도이다. 도 16의 방법은 상술한 도 10 및 도 12에서 도시된 장치에 의해 수행될 수 있으며, 또는 3D-HEVC에 적용

하여 수행될 수도 있다. 설명의 편의 상, 도 16의 방법은 병합 움직임 장치에 의해 수행되는 것으로 설명한다.

[0121] 도 16을 참조하면, 병합 움직임 장치는 현재 PU를 위한 병합 움직임 후보 리스트에 기본 병합 움직임 후보들을 추가한다(S1600).

[0122] 여기서, 기본 병합 움직임 후보들은, 상술한 바와 같이 현재 PU에 대한 공간적 병합 움직임 후보 및 시간적 병합 움직임 후보를 포함할 수 있으며, 독립적인 시점의 일반 영상에 대한 후보들일 수 있다.

[0123] 병합 움직임 장치는 현재 PU를 포함하는 현재 픽처가 깊이정보 맵인지 또는 의존적인 시점인지를 판단한다(S1610).

[0124] 만일, 현재 PU를 포함하는 현재 픽처가 깊이정보 맵 또는 의존적인 시점이라면, 병합 움직임 장치는 병합 움직임 후보 리스트에 확장 병합 움직임 후보를 추가한다(S1620).

[0125] 여기서, 확장 병합 움직임 후보는 깊이정보 맵 또는 의존적인 시점의 영상(일반 영상 및 깊이정보 맵)에 대한 후보들일 수 있다.

[0126] 표 1 내지 표 6의 기술 형식은 현재 MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)에서 공동으로 표준화를 진행 중인 JCT-3V(Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11)의 기술 형식이다.

[0127] 표 1은 기존 확장 병합 움직임 후보를 추가하는 과정을 포함하는 프로세스의 입출력의 일예를 나타내고, 표 2는 본 발명의 실시예에 따른 확장 병합 움직임 후보를 추가하는 과정을 포함하는 프로세스의 입출력의 일예를 나타낸다.

표 1

H.8.5.3.2 Derivation process for motion vector components and reference indices

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a luma location (x_{Bl} , y_{Bl}) of the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block,
- a variable n_{CbS} specifying the size of the current luma coding block,
- two variables n_{PbW} and n_{PbH} specifying the width and the height of the luma prediction block,
- a variable $partIdx$ specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are:

- the luma motion vectors $mvL0$ and $mvL1$,
- the chroma motion vectors $mvCL0$ and $mvCL1$,
- the reference indices $refIdxL0$ and $refIdxL1$,
- the prediction list utilization flags $predFlagL0$ and $predFlagL1$.
- the flag $subPbMotionFlag$, specifying, whether the motion data of the current PU has sub prediction block size motion accuracy.

Let (x_{Pb} , y_{Pb}) specify the top-left sample location of the current luma prediction block relative to the top-left luma sample of the current picture where $x_{Pb} = x_{Cb} + x_{Bl}$ and $y_{Pb} = y_{Cb} + y_{Bl}$.

Let the variable $currPic$ and $ListX$ be the current picture and $RefPicListX$, with X being 0 or 1, of the current picture, respectively.

The function $LongTermRefPic(aPic, aPb, refIdx, LX)$, with X being 0 or 1, is defined as follows:

- If the picture with index $refIdx$ from reference picture list LX of the slice containing prediction block aPb in the picture $aPic$ was marked as "used for long term reference" at the time when $aPic$ was the current picture, $LongTermRefPic(aPic, aPb, refIdx, LX)$ is equal to 1.
- Otherwise, $LongTermRefPic(aPic, aPb, refIdx, LX)$ is equal to 0.

The variables $vspModeFlag$, $ivpMvFlagL0$, $ivpMvFlagL1$ and $subPbMotionFlag$ are set equal to 0.

For the derivation of the variables $mvL0$ and $mvL1$, $refIdxL0$ and $refIdxL1$, as well as $predFlagL0$ and $predFlagL1$, the following applies:

- If $merge_flag[x_{Pb}][y_{Pb}]$ is equal to 1, the derivation process for luma motion vectors for merge mode as specified in subclause H.8.5.3.2.1 is invoked with the luma location (x_{Cb} , y_{Cb}), the luma location (x_{Pb} , y_{Pb}), the variables n_{CbS} , n_{PbW} , n_{PbH} , and the partition index $partIdx$ as inputs, and the output being the luma motion vectors $mvL0$, $mvL1$, the reference indices $refIdxL0$, $refIdxL1$, and the prediction list utilization flags $predFlagL0$ and $predFlagL1$, the disparity vector availability flags $ivpMvFlagL0$ and $ivpMvFlagL1$, the flag $vspModeFlag$, and the flag $subPbMotionFlag$.

[0128]

표 2

H.8.5.3.2 Derivation process for motion vector components and reference indices Inputs to this process are: – a luma location (x_{Cb}, y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture, – a luma location (x_{Bl}, y_{Bl}) of the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block, – a variable n_{CbS} specifying the size of the current luma coding block, – two variables n_{PbW} and n_{PbH} specifying the width and the height of the luma prediction block, – a variable $partIdx$ specifying the index of the current prediction unit within the current coding unit. Outputs of this process are: – the luma motion vectors $mvL0$ and $mvL1$, – the chroma motion vectors $mvCL0$ and $mvCL1$, – the reference indices $refIdxL0$ and $refIdxL1$, – the prediction list utilization flags $predFlagL0$ and $predFlagL1$. – the flag $subPbMotionFlag$, specifying, whether the motion data of the current PU has sub prediction block size motion accuracy. Let (x_{Pb}, y_{Pb}) specify the top-left sample location of the current luma prediction block relative to the top-left luma sample of the current picture where $x_{Pb} = x_{Cb} + x_{Bl}$ and $y_{Pb} = y_{Cb} + y_{Bl}$. Let the variable $currPic$ and $ListX$ be the current picture and $RefPicListX$, with X being 0 or 1, of the current picture, respectively. The function $LongTermRefPic(aPic, aPb, refIdx, LX)$, with X being 0 or 1, is defined as follows: – If the picture with index $refIdx$ from reference picture list LX of the slice containing prediction block aPb in the picture $aPic$ was marked as "used for long term reference" at the time when $aPic$ was the current picture, $LongTermRefPic(aPic, aPb, refIdx, LX)$ is equal to 1. – Otherwise, $LongTermRefPic(aPic, aPb, refIdx, LX)$ is equal to 0. The variables $vspModeFlag$, $ivpMvFlagL0$, $ivpMvFlagL1$ and $subPbMotionFlag$ are set equal to 0. For the derivation of the variables $mvL0$ and $mvL1$, $refIdxL0$ and $refIdxL1$, as well as $predFlagL0$ and $predFlagL1$, the following applies: – If $merge_flag[x_{Pb}][y_{Pb}]$ is equal to 1, the following applies: – The derivation process for base luma motion vectors for merge mode as specified in subclause H.8.5.3.2.17 is invoked with the luma location (x_{Cb}, y_{Cb}), the luma location (x_{Pb}, y_{Pb}), the variables n_{CbS}, n_{PbW}, n_{PbH}, and the partition index $partIdx$ as inputs, and the output being the luma motion vectors $mvL0$, $mvL1$, the reference indices $refIdxL0$, $refIdxL1$, and the prediction list utilization flags $predFlagL0$ and $predFlagL1$, the additional output parameters being the merge candidate lists $mergeCandList$, the availability flags $availableFlagN$ of the neighbouring prediction units (with N being replaced by $A0$, $A1$, $B0$, $B1$ or $B2$). – If $DepthFlag$ is equal to 1 or $ViewIdx$ is not equal to 0, the derivation process for luma motion vectors for merge mode as specified in subclause H.8.5.3.2.1 is invoked with the luma location (x_{Cb}, y_{Cb}), the luma location (x_{Pb}, y_{Pb}), the variables n_{CbS}, n_{PbW}, n_{PbH}, and the partition index $partIdx$, the merge candidate lists $mergeCandList$, the availability flags $availableFlagN$ of the neighbouring prediction units (with N being replaced by $A0$, $A1$, $B0$, $B1$ or $B2$) as inputs, and the output being the luma motion vectors $mvL0$, $mvL1$, the reference indices $refIdxL0$, $refIdxL1$, and the prediction list utilization flags $predFlagL0$ and $predFlagL1$, the disparity vector availability flags $ivpMvFlagL0$ and $ivpMvFlagL1$, the flag $vspModeFlag$, and the flag $subPbMotionFlag$.
--

[0129]

[0130]

표 2에 도시된 본 발명의 실시예에 따른 확장 병합 움직임 후보를 추가하는 과정은 추가적인 입력으로서, 병합 움직임 후보 리스트($mergeCandList$), 기본 병합 움직임 후보가 추가 되었는지를 지시하는 플래그 ($availableFlagN$)가 사용 된다.

[0131]

표 2에서 N 은 좌(left), 상(above), 우상(above-right), 좌하(bottom-left), 좌상(above-left) 위치의 후보임을 의미하는 $A0$, $A1$, $B0$, $B1$, $B2$ 로 대체될 수 있다. 입력 받은 병합 움직임 후보 리스트($mergeCandList$)에는 기존 방법에 의하여 기본 병합 움직임 후보들이 임의의 우선 순위대로 저장되어 있다. 일 예로, 좌 후보, 상 후보, 우상 후보, 좌하 후보, 좌상 후보, 시간적 (예측) 후보, 합성 양방향 예측 후보(combined bi-predictive candidate), 움직임이 0인 후보 순서로 저장 되어 있을 수 있다. 출력은 확장 병합 움직임 후보들에 대한 추가 작업이 완료된 병합 움직임 후보 리스트이다.

[0132] 표 3은 기존 확장 병합 움직임 후보 추가 과정을 나타낸 것이고, 표 4는 본 발명의 실시예에 따른 확장 병합 움직임 후보 추가 과정을 나타낸 것이다.

[0133] 표 4에서는 기본 병합 움직임 후보들이 이미 추가되어 있는 리스트를 처리하므로, 확장 병합 움직임 후보들에 대한 과정들만을 처리하게 된다. 따라서, 기존 3D-HEVC에서, HEVC에서 사용하던 병합 움직임 후보들에 대한 처리가 중복 구현되지 않도록 생략할 수 있다.

표 3

H.8.5.3.2.3 Derivation process for luma motion vectors for merge mode

1. The derivation process for merging candidates from neighbouring prediction unit partitions in subclause 8.5.3.2.2 is invoked with the luma coding block location (x_{Cb} , y_{Cb}), the coding block size n_{CbS} , the luma prediction block location (x_{Pb} , y_{Pb}), the luma prediction block width n_{PbW} , the luma prediction block height n_{PbH} , and the partition index $partIdx$ as inputs, and the output being the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$, and $availableFlagB_2$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$, and $refIdxLXB_2$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$, and $predFlagLXB_2$, and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$, and $mvLXB_2$, with X being 0 or 1.
2. The reference indices for the temporal merging candidate, $refIdxLXCol$, with X being 0 or 1, are set equal to 0.
3. The derivation process for temporal luma motion vector prediction in subclause H.8.5.3.2.7 is invoked with the luma location (x_{Pb} , y_{Pb}), the luma prediction block width n_{PbW} , the luma prediction block height n_{PbH} , and the variable $refIdxL0Col$ as inputs, and the output being the availability flag $availableFlagL0Col$ and the temporal motion vector $mvL0Col$. The variables $availableFlagCol$, $predFlagL0Col$ and $predFlagL1Col$ are derived as follows:

$$availableFlagCol = availableFlagL0Col$$

$$predFlagL0Col = availableFlagL0Col$$

$$predFlagL1Col = 0$$
4. When `slice_type` is equal to B, the derivation process for temporal luma motion vector prediction in subclause H.8.5.3.2.7 is invoked with the luma location (x_{Pb} , y_{Pb}), the luma prediction block width n_{PbW} , the luma prediction block height n_{PbH} , and the variable $refIdxL1Col$ as inputs, and the output being the availability flag $availableFlagL1Col$ and the temporal motion vector $mvL1Col$. The variables $availableFlagCol$ and $predFlagL1Col$ are derived as follows:

$$availableFlagCol = availableFlagL0Col \mid\mid availableFlagL1Col$$

$$predFlagL1Col = availableFlagL1Col$$
5. Depending on `iv_mv_pred_flag[ nuh_layer_id ]`, the following applies.
 - If `iv_mv_pred_flag[ nuh_layer_id ]` is equal to 0, the flags $availableFlagIvMC$, $availableIvMCShift$ and $availableFlagIvDC$ are set equal to 0.
 - Otherwise (`iv_mv_pred_flag[ nuh_layer_id ]` is equal to 1), the derivation process for the inter-view merge candidates as specified in subclause H.8.5.3.2.10 is invoked with the luma location (x_{Pb} , y_{Pb}), the variables n_{PbW} and n_{PbH} , as the inputs and the output is assigned to the availability flags $availableFlagIvMC$, $availableIvMCShift$ and $availableFlagIvDC$, the reference

[0134]

indices refIdxLXIvMC , refIdxLXIvMCShift and refIdxLXIvDC , the prediction list utilization flags predFlagLXIvMC , $\text{predFlagLXIvMCShift}$ and predFlagLXIvDC , and the motion vectors mvLXIvMC , mvLXIvMCShift and mvLXIvDC (with X being 0 or 1, respectively)..

6. Depending on $\text{view_synthesis_pred_flag[nuh_layer_id]}$, the following applies.
 - If $\text{view_synthesis_pred_flag[nuh_layer_id]}$ is equal to 0, the flag availableFlagVSP is set equal to 0.
 - Otherwise ($\text{view_synthesis_pred_flag[nuh_layer_id]}$ is equal to 1), the derivation process for a view synthesis prediction merge candidate as specified in subclause H.8.5.3.2.13 is invoked with the luma locations (x_{Cb} , y_{Cb}) as input and the outputs are the availability flag availableFlagVSP , the reference indices refIdxL0VSP and refIdxL1VSP , the prediction list utilization flags predFlagL0VSP and predFlagL1VSP , and the motion vectors mvL0VSP and mvL1VSP .
7. Depending on $\text{mpi_flag[nuh_layer_id]}$, the following applies.
 - If $\text{mpi_flag[nuh_layer_id]}$ is equal to 0, the variable availableFlagT is set equal to 0.
 - Otherwise ($\text{mpi_flag[nuh_layer_id]}$ is equal to 1), the derivation process for the texture merging candidate as specified in subclause H.8.5.3.2.14 is invoked with the luma location (x_{Pb} , y_{Pb}), the variables n_{PbW} and n_{PbH} as the inputs and the outputs are the flag availableFlagT , the prediction utilization flags predFlagL0T and predFlagL1T , the reference indices refIdxL0T and refIdxL1T , and the motion vectors mvL0T and mvL1T .
8. The merge candidate lists mergeCandList and $\text{mergeCandIsVspFlag}$ are constructed as specified by the following ordered steps:
 - a. The variable numMergeCand is set equal to 0.
 - b. When availableFlagT is equal to 1, the entry $\text{mergeCandList[numMergeCand]}$ is set equal to T, the entry $\text{mergeCandIsVspFlag[numMergeCand]}$ is set equal to 0 and the variable numMergeCand is increased by 1.
 - c. When availableFlagIvMC is equal to 1,
 - When the following condition is true,
 - $\text{availableFlagT} == 0$,
 - or one or more of the following conditions are true:
 - $\text{availableFlagT} == 1 \ \&\& \ \text{predFlagLXT} \neq \text{predFlagLXIcMv}$ (with X being replaced by 0 and 1),
 - $\text{availableFlagT} == 1 \ \&\& \ \text{mvLXT} \neq \text{mvLXIcMv}$ (with X being replaced by 0 and 1),
 - $\text{availableFlagT} == 1 \ \&\& \ \text{refIdxLXT} \neq \text{refIdxLXIcMv}$ (with X

[0135]

being replaced by 0 and 1), the entry mergeCandList[numMergeCand] is set equal to IvMC, the entry mergeCandIsVspFlag[numMergeCand] is set equal to 0 and the variable numMergeCand is increased by 1. d. When availableFlagA_i is equal to 1, the following applies, with N being replaced by (DepthFlag ? T : IvMC): – When the following condition is true, – availableFlagN == 0, or one or more of the following conditions are true: – availableFlagN == 1 && predFlagLXN != predFlagLXA_i (with X being replaced by 0 and 1), – availableFlagN == 1 && mvLXN != mvLXA_i (with X being replaced by 0 and 1), – availableFlagN == 1 && refIdxLXN != refIdxLXA_i (with X being replaced by 0 and 1), the entry mergeCandList[numMergeCand] is set equal to A_i, the entry mergeCandIsVspFlag[numMergeCand] is set equal to VspModeFlag[xPb - 1][yPb + nPbH - 1] and the variable numMergeCand is increased by 1. e. When availableFlagB_i is equal to 1, the following applies, with N being replaced by (DepthFlag ? T : IvMC): – When the following condition is true, – availableFlagN == 0, or one or more of the following conditions are true: – availableFlagN == 1 && predFlagLXN != predFlagLXB_i (with X being replaced by 0 and 1), – availableFlagN == 1 && mvLXN != mvLXB_i (with X being replaced by 0 and 1), – availableFlagN == 1 && refIdxLXN != refIdxLXB_i (with X being replaced by 0 and 1), the entry mergeCandList[numMergeCand] is set equal to B_i, the entry mergeCandIsVspFlag[numMergeCand] is set equal to 0 and the variable numMergeCand is increased by 1. f. When availableFlagB₀ is equal to 1, the entry mergeCandList[numMergeCand] is set equal to B₀, the entry mergeCandIsVspFlag[numMergeCand] is set equal to 0 and the variable numMergeCand is increased by 1.
--

[0136]

- g. When availableFlagIvDC is equal to 1, and one or more of the following conditions is true,
- availableFlagA1 == 0,
 - predFlagLXA1 != predFlagLXIvDC, (with X being replaced by 0 and 1),
 - mvLXA1 != mvLXIvDC (with X being replaced by 0 and 1),
 - reldxLXA1 != reldxLXIvDC (with X being replaced by 0 and 1),
- and one or more of the following conditions is true,
- availableFlagB1 == 0,
 - predFlagLXB1 != predFlagLXIvDC, (with X being replaced by 0 and 1),
 - mvLXB1 != mvLXIvDC (with X being replaced by 0 and 1),
 - reldxLXB1 != reldxLXIvDC (with X being replaced by 0 and 1),
- the entry mergeCandList[numMergeCand] is set equal to IvDC, the entry mergeCandIsVspFlag[numMergeCand] is set equal to 0 and the variable numMergeCand is increased by 1.
- h. When availableFlagVSP is equal to 1, ic_flag is equal to 0 and iv_res_pred_weight_idx is equal to 0, the entry mergeCandList[numMergeCand] is set equal to VSP, the entry mergeCandIsVspFlag[numMergeCand] is set equal 1 and the variable numMergeCand is increased by 1.
- i. When availableFlagA0 is equal to 1 and numMergeCand is less than 5 + NumExtraMergeCand, the entry mergeCandList[numMergeCand] is set equal to A0, the entry mergeCandIsVspFlag[numMergeCand] is set equal to VspModeFlag[xPb - 1][yPb + nPbH] and the variable numMergeCand is increased by 1.
- j. When availableFlagB2 is equal to 1 and numMergeCand is less than 4 + NumExtraMergeCand, the entry mergeCandList[numMergeCand] is set equal to B2, the entry mergeCandIsVspFlag[numMergeCand] is set equal to 0 and the variable numMergeCand is increased by 1.
- k. When availableFlagIvMCShift is equal to 1 and numMergeCand is less than 5 + NumExtraMergeCand, and one or more of the following conditions are true,
- availableFlagIvMC == 0,
 - predFlagLXMC != predFlagLXMCShift (with X being replaced by 0 and 1),

[0137]

- $mvLXMC \neq mvLXiVMCShift$ (with X being replaced by 0 and 1),
 - $refIdxLXMC \neq refIdxLXMCSHIFT$ (with X being replaced by 0 and 1),
- the entry $mergeCandList[numMergeCand]$ is set equal to $IvMCSHIFT$, the entry $mergeCandIsVspFlag[numMergeCand]$ is set equal to 0 and the variable $numMergeCand$ is increased by 1.
1. A variable $availableFlagIvDCShift$ is set to 0 and when all of the following conditions are true
- $DepthFlag$ is equal to 0, $availableFlagIvMCSHIFT$ is equal to 0,
 - $numMergeCand$ is less than $5 + NumExtraMergeCand$,
- the derivation process for the shifted disparity merging candidate as specified in subclause H.8.5.3.2.15 is invoked with the availability flags $availableFlagN$, the reference indices $refIdxL0N$ and $refIdxL1N$, the prediction list utilization flags $predFlagL0N$ and $predFlagL1N$, the motion vectors $mvL0N$ and $mvL1N$, of every candidate N being in $mergeCandList$, $mergeCandList$, $mergeCandIsVspFlag$, and $numMergeCand$ as the inputs and the outputs are the flag $availableFlagIvDCShift$, the prediction utilization flags $predFlagL0IvDCShift$ and $predFlagL1IvDCShift$, the reference indices $refIdxL0IvDCShift$ and $refIdxL1IvDCShift$, and the motion vectors $mvL0IvDCShift$ and $mvL1IvDCShift$. When $availableFlagIvDCShift$ is equal to 1, the entry $mergeCandList[numMergeCand]$ is set equal to $IvDCShift$, the entry $mergeCandIsVspFlag[numMergeCand]$ is set equal to 0 and the variable $numMergeCand$ is increased by 1.
- m. When $availableFlagCol$ is equal to 1 and $numMergeCand$ is less than $5 + NumExtraMergeCand$, the entry $mergeCandList[numMergeCand]$ is set equal to Col , the entry $mergeCandIsVspFlag[numMergeCand]$ is set equal to 0 and the variable $numMergeCand$ is increased by 1.
9. The variable $numOrigMergeCand$ is set equal to $numMergeCand$.
10. When $slice_type$ is equal to B, the derivation process for combined bi-predictive merging candidates specified in subclause H.8.5.3.2.3 is invoked with $mergeCandList$, $mergeCandIsVspFlag$, the reference indices $refIdxL0N$ and $refIdxL1N$, the prediction list utilization flags $predFlagL0N$ and $predFlagL1N$, the motion vectors $mvL0N$ and $mvL1N$ of every candidate N in $mergeCandList$, $numCurrMergeCand$, and $numOrigMergeCand$ as inputs, and the output is assigned to $mergeCandList$, $numCurrMergeCand$, the reference indices $refIdxL0combCand_k$ and $refIdxL1combCand_k$, the prediction list utilization flags $predFlagL0combCand_k$ and $predFlagL1combCand_k$, and the motion vectors $mvL0combCand_k$ and $mvL1combCand_k$ of every new candidate $combCand_k$ being added into $mergeCandList$. The number of candidates being added, $numCombMergeCand$, is set equal to $(numCurrMergeCand - numOrigMergeCand)$. When $numCombMergeCand$ is greater than 0, k ranges from 0 to

- $numCombMergeCand - 1$, inclusive, and $mergeCandIsVspFlag[numOrigMergeCand + k]$ is set equal to 0.
11. The derivation process for zero motion vector merging candidates specified in subclause 8.5.3.2.4 is invoked with the $mergeCandList$, the reference indices $refIdxL0N$ and $refIdxL1N$, the prediction list utilization flags $predFlagL0N$ and $predFlagL1N$, the motion vectors $mvL0N$ and $mvL1N$ of every candidate N in $mergeCandList$, and $numCurrMergeCand$ as inputs, and the output is assigned to $mergeCandList$, $numCurrMergeCand$, the reference indices $refIdxL0zeroCand_m$ and $refIdxL1zeroCand_m$, the prediction list utilization flags $predFlagL0zeroCand_m$ and $predFlagL1zeroCand_m$, and the motion vectors $mvL0zeroCand_m$ and $mvL1zeroCand_m$ of every new candidate $zeroCand_m$ being added into $mergeCandList$. The number of candidates being added, $numZeroMergeCand$, is set equal to $(numCurrMergeCand - numOrigMergeCand - numCombMergeCand)$. When $numZeroMergeCand$ is greater than 0, m ranges from 0 to $numZeroMergeCand - 1$, inclusive, and $mergeCandIsVspFlag[numOrigMergeCand + numCombMergeCand + m]$ is set equal to 0.
-

표 4

H.8.5.3.2.3 Derivation process for luma motion vectors for merge mode

1. Depending on `iv_mv_pred_flag[ nuh_layer_id ]`, the following applies.
 - If `iv_mv_pred_flag[ nuh_layer_id ]` is equal to 0, the flags `availableFlagIvMC`, `availableIvMCShift` and `availableFlagIvDC` are set equal to 0.
 - Otherwise (`iv_mv_pred_flag[ nuh_layer_id ]` is equal to 1), the derivation process for the inter-view merge candidates as specified in subclause H.8.5.3.2.10 is invoked with the luma location (`xPb`, `yPb`), the variables `nPbW` and `nPbH`, as the inputs and the output is assigned to the availability flags `availableFlagIvMC`, `availableIvMCShift` and `availableFlagIvDC`, the reference indices `refIdxLXIvMC`, `refIdxLXIvMCShift` and `refIdxLXIvDC`, the prediction list utilization flags `predFlagLXIvMC`, `predFlagLXIvMCShift` and `predFlagLXIvDC`, and the motion vectors `mvLXIvMC`, `mvLXIvMCShift` and `mvLXIvDC` (with `X` being 0 or 1, respectively)..
2. Depending on `view_synthesis_pred_flag[ nuh_layer_id ]`, the following applies.
 - If `view_synthesis_pred_flag[ nuh_layer_id ]` is equal to 0, the flag `availableFlagVSP` is set equal to 0.
 - Otherwise (`view_synthesis_pred_flag[ nuh_layer_id ]` is equal to 1), the derivation process for a view synthesis prediction merge candidate as specified in subclause H.8.5.3.2.13 is invoked with the luma locations (`xCb`, `yCb`) as input and the outputs are the availability flag `availableFlagVSP`, the reference indices `refIdxLOVSP` and `refIdxLIVSP`, the prediction list utilization flags `predFlagLOVSP` and `predFlagLIVSP`, and the motion vectors `mvLOVSP` and `mvLIVSP`.
3. Depending on `mpi_flag[ nuh_layer_id ]`, the following applies.
 - If `mpi_flag[ nuh_layer_id ]` is equal to 0, the variable `availableFlagT` is set equal to 0.
 - Otherwise (`mpi_flag[ nuh_layer_id ]` is equal to 1), the derivation process for the texture merging candidate as specified in subclause H.8.5.3.2.14 is invoked with the luma location (`xPb`, `yPb`), the variables `nPbW` and `nPbH` as the inputs and the outputs are the flag `availableFlagT`, the prediction utilization flags `predFlagLOT` and `predFlagLIT`, the reference indices `refIdxLOT` and `refIdxLIT`, and the motion vectors `mvLOT` and `mvLIT`.
4. The merge candidate lists `mergeCandList` and `mergeCandIsVspFlag` are constructed as specified by the following ordered steps:
 - a. The variables `numMergeCand`, `numA1B1B0`, and `numA0B2` are set equal to 0.
 - ~~b.~~ When `availableFlagAi` is equal to 1, the entry `mergeCandIsVspFlag[ numA1B1B0 ]` is set equal to

[0140]

VspModeFlag[$xPb - 1$][$yPb + nPbH - 1$] and variable numA1B1B0 is increased by 1.

- e. When availableFlagB₁ is equal to 1, the entry mergeCandIsVspFlag[numA1B1B0] is set equal to VspModeFlag[$xPb + nPbW - 1$][$yPb - 1$] and the variable numA1B1B0 is increased by 1.
- d. When availableFlagB₀ is equal to 1, the entry mergeCandIsVspFlag[numA1B1B0] is set equal to VspModeFlag[$xPb + nPbW$][$yPb - 1$] and the variable numA1B1B0 is increased by 1.
- e. When availableFlagA₀ is equal to 1, the entry mergeCandIsVspFlag[numA1B1B0] is set equal to VspModeFlag[$xPb - 1$][$yPb + nPbH$] and the variable numA0B2 is increased by 1.
- f. When availableFlagB₂ is equal to 1, the entry mergeCandIsVspFlag[numA1B1B0 + numA0B2] is set equal to VspModeFlag[$xPb - 1$][$yPb - 1$] and the variable numA0B2 is increased by 1.
- g. When availableFlagT is equal to 1, the following applies:
 - The following applies for N being replaced by A1 and B1:
 - The variable pruneFlagN is set equal to 0.
 - When all of the following conditions are true,
 - availableFlagN == 1,
 - predFlagLXN == predFlagLXT, (with X being replaced by 0 and 1),
 - mvLXN == mvLXT (with X being replaced by 0 and 1),
 - refIdxLXN == refIdxLXT (with X being replaced by 0 and 1),
 pruneFlagN is set equal to 1.
 - The variables mergeCandList, MergeCandIsVspFlag, numMergeCand and numA1B1B0 are modified as specified in the following:
 - Depending on pruneA1Flag and pruneB1Flag, the following applies:
 - If pruneFlagA1 is equal to 0 and pruneFlagB1 is equal to 0, the following applies for k in the range of MaxNumMergeCand - 1 to 1, inclusive,

[0141]

- mergeCandList[k] = mergeCandList[k - 1]
- mergeCandIsVspFlag[k] = mergeCandIsVspFlag[k - 1]
- Otherwise, the following applies:
 - numA1B1B0 = numA1B1B0 - 1
 - When availableFlagA1 + availableFlagB1 is greater than 1 and pruneFlagA1 is equal to 0, the following applies:
 - mergeCandList[1] = mergeCandList[0]
 - mergeCandIsVspFlag[1] = mergeCandIsVspFlag[0]
 - The entry mergeCandList[numMergeCand] is set equal to T, the entry mergeCandIsVspFlag[numMergeCand] is set equal to 0.
 - The variable numMergeCand is increased by 1.

h. When availableFlagIvMC is equal to 1, the following applies:

The variable pruneFlagA1, pruneFlagB1, pruneFlagT, addIvMC is set equal to 0.

If DepthFlag is equal to 0, the following applies for N being replaced by A1, B1:

- When all of the following conditions are true,
 - availableFlagNe followin
 - predFlagLXNagNe following conditions are true, 1, B1:pplies:Flag[numM
 - mvLXNlagLXNagNe following conditions are true, 1, B1:ppli
 - refldxLXNXNagNe following conditions are true, 1, B1:pplies:Flag[
- pruneFlagN is set equal to 1.

Otherwise, if DepthFlag is equal to 1, following applies:

- When all of the following conditions is true,
 - availableFlagT =following conditions is true, laced by 0 and 1),d 1),1B1B0 are modified as sp
 - availableFlagT =following conditions is true, laced by 0 and 1),d 1),1B1B0 are mo
 - availableFlagT =following conditions is true,dxLXIvMC (with X being replaced by 0 and 1),
- pruneFlagT is set equal to 1.
- The variables mergeCandList, MergeCandIsVspFlag, numMergCand and numA1B1B0 are modified as specified in the following:
 - Depending on pruneA1Flag and pruneB1Flag, and pruneFlagT, the following applies:

[0142]

- If `pruneFlagA1` is equal to 0 and `pruneFlagB1` is equal to 0, and `pruneFlagT` is equal to 0, the following applies for `k` in the range of `MaxNumMergeCand` + 1 to `numMergeCand` + 1, inclusive,
 - `addIvMC = 1`
 - `mergeCandList[and + 1, inclusive,pr k – 1 ]`
 - `mergeCandIsVspFlag[ 1, inclusive,pruneFlagB1` is equal
- Otherwise, if `DepthFlag` is equal to 0 and one of the following conditions is true,
 - `pruneFlagA1 == 1`,
 - `pruneFlagB1 == 1`,
 the following applies:
 - `numA1B1B0 = numA1B1B0 g i`
 - `addIvMC = 1`
 - When `availableFlagA10 g` is equal to 0 and one of the following conditions is true, is
 - `mergeCandList[ 1 ]` is set equal to `mergeCandList[ 0 ]`,
`mergeCandIsVspFlag[ 1 ]` is set equal to `mergeCandIsVspFlag[ 0 ]`.
- If `addIvMC` is equal to 1, the following applies:
 - The entry `mergeCandList[ numMergeCand ]` is set equal to `IvMC`,
 the entry `mergeCandIsVspFlag[ numMergeCand ]` is set equal to 0.
 - The variable `numMergeCand` is increased by 1.
- i. The variable `numMergeCand` is set equal to `numMergeCand + numA1B1B0`.
- j. When `availableFlagIvDC` is equal to 1, and one or more of the following conditions is true,
 - `availableFlagA1 == 0`,
 - `predFlagLXA1 != predFlagLXIvDC`, (with `X` being replaced by 0 and 1),
 - `mvLXA1 != mvLXIvDC` (with `X` being replaced by 0 and 1),
 - `refIdxLXA1 != refIdxLXIvDC` (with `X` being replaced by 0 and 1),
 and one or more of the following conditions is true,
 - `availableFlagB1 == 0`,
 - `predFlagLXB1 != predFlagLXIvDC`, (with `X` being replaced by 0 and 1),
 - `mvLXB1 != mvLXIvDC` (with `X` being replaced by 0 and 1),

[0143]

- $\text{refIdxLXB}_i \neq \text{refIdxLXIvDC}$ (with X being replaced by 0 and 1),

k ranges from $\text{MaxNumMergeCand} - 1$ to $\text{numMergeCand} + 1$, inclusive, and $\text{mergeCandList}[k]$ is set equal to $\text{mergeCandList}[k - 1]$, $\text{mergeCandIsVspFlag}[k]$ is set equal to $\text{mergeCandIsVspFlag}[k - 1]$, the entry $\text{mergeCandList}[\text{numMergeCand}]$ is set equal to IvDC, the entry $\text{mergeCandIsVspFlag}[\text{numMergeCand}]$ is set equal to 0 and the variable numMergeCand is increased by 1.

- k. When availableFlagVSP is equal to 1, ic_flag is equal to 0 and $\text{iv_res_pred_weight_idx}$ is equal to 0 and numMergeCand is less than $5 + \text{NumExtraMergeCand}$, k ranges from $\text{MaxNumMergeCand} - 1$ to $\text{numMergeCand} + 1$, inclusive, and $\text{mergeCandList}[k]$ is set equal to $\text{mergeCandList}[k - 1]$, $\text{mergeCandIsVspFlag}[k]$ is set equal to $\text{mergeCandIsVspFlag}[k - 1]$, the entry $\text{mergeCandList}[\text{numMergeCand}]$ is set equal to VSP, the entry $\text{mergeCandIsVspFlag}[\text{numMergeCand}]$ is set equal to 1 and the variable numMergeCand is increased by 1.

- l. The variable numMergeCand is set equal to $\text{numMergeCand} + \text{numA0B2}$.

- m. When $\text{availableFlagIvMCShift}$ is equal to 1 and numMergeCand is less than $5 + \text{NumExtraMergeCand}$, and one or more of the following conditions are true,

- $\text{availableFlagIvMC} == 0$,
- $\text{predFlagLXMC} \neq \text{predFlagLXMCShift}$ (with X being replaced by 0 and 1),
- $\text{mvLXMC} \neq \text{mvLXIvMCShift}$ (with X being replaced by 0 and 1),
- $\text{refIdxLXMC} \neq \text{refIdxLXMCShift}$ (with X being replaced by 0 and 1),

k ranges from $\text{MaxNumMergeCand} - 1$ to $\text{numMergeCand} + 1$, inclusive, and $\text{mergeCandList}[k]$ is set equal to $\text{mergeCandList}[k - 1]$, $\text{mergeCandIsVspFlag}[k]$ is set equal to $\text{mergeCandIsVspFlag}[k - 1]$, the entry $\text{mergeCandList}[\text{numMergeCand}]$ is set equal to IvMCShift, the entry $\text{mergeCandIsVspFlag}[\text{numMergeCand}]$ is set equal to 0.

- n. A variable $\text{availableFlagIvDCShift}$ is set to 0 and when all of the following conditions are true

- DepthFlag is equal to 0,
- $\text{availableFlagIvMCShift}$ is equal to 0,
- numMergeCand is less than $5 + \text{NumExtraMergeCand}$,

the derivation process for the shifted disparity merging candidate as specified in subclause H.8.5.3.2.15 is invoked with the availability flags availableFlagN , the

[0144]

reference indices refIdxL0N and refIdxL1N , the prediction list utilization flags predFlagL0N and predFlagL1N , the motion vectors mvL0N and mvL1N , of every candidate N being in mergeCandList , mergeCandList , $\text{mergeCandIsVspFlag}$, and numMergeCand as the inputs and the outputs are the flag $\text{availableFlagIvDCShift}$, the prediction utilization flags $\text{predFlagL0IvDCShift}$ and $\text{predFlagL1IvDCShift}$, the reference indices refIdxL0IvDCShift and refIdxL1IvDCShift , and the motion vectors mvL0IvDCShift and mvL1IvDCShift . When $\text{availableFlagIvDCShift}$ is equal to 1, k ranges from $\text{MaxNumMergeCand} - 1$ to $\text{numMergeCand} + 1$, inclusive, and $\text{mergeCandList}[k]$ is set equal to $\text{mergeCandList}[k - 1]$, $\text{mergeCandIsVspFlag}[k]$ is set equal to $\text{mergeCandIsVspFlag}[k - 1]$, the entry $\text{mergeCandList}[\text{numMergeCand}]$ is set equal to IvDCShift , the entry $\text{mergeCandIsVspFlag}[\text{numMergeCand}]$ is set equal to 0 and the variable numMergeCand is increased by 1.

....

H.8.5.3.2.17 Derivation process for base luma motion vectors for merge mode

The specifications in subclause 8.5.3.2.1 apply, with the following changes:

The following paragraph

- “When slice_type is equal to B, the derivation process for combined bi-predictive merging candidates” is replaced by “When slice_type is equal to B and numMergeCand is less than 5, the derivation process for combined bi-predictive merging candidates”
- “temporal luma motion vector prediction in subclause 8.5.3.2.7 is invoked” is replaced by “temporal luma motion vector prediction in subclause H.8.5.2.3.7 is invoked”

[0145]

[0146]

표 4에 기술된 본 발명의 실시예에 따른 확장 병합 움직임 후보 추가 과정에서는 혼합 양방향 예측 후보(combined bi-predictive candidate)를 유도할 시, 확장 움직임 병합 후보들을 추가로 이용하는 기존 방법을 사용하지 않고, HEVC 표준에서 사용하는 기본 움직임 병합 후보들만을 혼합하여 유도하는 방법을 사용하여 기존 방법보다 연산을 줄이면서 거의 동일한 부호화 효율을 얻을 수 있다.

[0147]

표 4에 기술된 본 발명의 실시예에 따른 확장 병합 움직임 후보 추가 과정은 도 17a 내지 도 17f를 참조하여 구체적으로 설명하도록 한다.

[0148]

도 17a, 도 17b, 도 17c, 도 17d, 도 17e 및 도 17f는 본 발명의 일 실시예에 따른 확장 병합 움직임 후보를 병합 움직임 후보 리스트에 추가하는 방법을 나타내는 순서도이다.

[0149]

도 17a 내지 도 17f의 방법은 상술한 표 4의 확장 병합 움직임 후보 추가 과정을 기반으로 구성된 것이다. 도 17a 내지 도 17f의 방법은 상술한 도 10 및 도 12에서 도시된 장치에 의해 수행될 수 있으며, 또는 3D-HEVC에 적용하여 수행될 수도 있다.

[0150]

1. 플래그 $\text{iv_mv_pred_flag}[\text{nuh_layer_id}]$ 는 현재 PU가 시점 간 예측(inter view prediction)을 시도 할 수 있는가를 의미한다. 상기 플래그 $\text{iv_mv_pred_flag}[\text{nuh_layer_id}]$ 가 1이라면, 시점 간 병합 후보(inter-view merge candidate: IvMC), 시차 병합 후보(inter-view disparity merge candidate; IvDC), 이동된 시점 간 병합 후보(shifted inter-view merge candidate: IvMCShift)의 사용 가능 여부를 확인하여 각각 플래그 availableFlagIvMC , $\text{availableIvMCShift}$, availableFlagIvDC 에 저장하고, 사용 가능한 후보들의 움직임 정보를 유도한다.

[0151]

2. 플래그 $\text{view_synthesis_pred_flag}[\text{nuh_layer_id}]$ 는 현재 PU가 시점 합성 예측(view synthesis prediction)을 시도 할 수 있는가를 의미한다. 상기 플래그 $\text{view_synthesis_pred_flag}[\text{nuh_layer_id}]$ 가 1이라면, 시점 간

합성 병합 후보의 사용 가능 여부를 플래그 availableFlagVSP에 저장하고, 사용 가능하다면 움직임 정보를 유도한다.

- [0152] 3. 플래그 mpi_flag[nuh_layer_id]는 현재 PU가 깊이정보 맵이며, 텍스처 블록으로부터 움직임 예측을 시도 할 수 있는가를 의미한다. 상기 플래그 mpi_flag[nuh_layer_id]가 1이면, 텍스처 병합 후보의 사용 가능 여부를 플래그 availableFlagT에 저장하고, 사용 가능하다면 움직임 정보를 유도한다.
- [0153] 4. 현재까지 기본 병합 움직임 후보들로만 구성된 병합 움직임 후보 리스트(mergeCandList)와 각 후보에 대한 시점 간 예측 플래그(mergeCandIsVspFlag)는 다음과 같은 과정으로 재구성 된다.
- [0154] a. numMergeCand은 병합 움직임 후보의 총 개수, numA1B1B0는 기본 병합 움직임 후보 중 좌(left), 상(above), 우상(above-right) 위치에 해당하는 후보들의 개수, numA0B2는 기본 병합 움직임 후보 중 좌하(bottom-left), 좌상(above-left)의 위치에 해당하는 후보들의 개수이다. numMergeCand, numA1B1B0, numA0B2을 0으로 초기화한다.
- [0155] b. 좌 후보(A1)가 사용 가능한지 판단한다. 좌 후보가 사용 가능하다면, numA1B1B0을 1 증가시킨다. 또한, 좌 후보가 시점 간 합성 예측(View Synthesis Prediction, 이하 VSP)을 사용하였는지를 플래그로 저장한다.
- [0156] c. 상 후보(B1)가 사용 가능한지 판단한다. 상 후보가 사용 가능하다면, numA1B1B0을 1 증가시킨다. 또한, 상 후보가 VSP를 사용하였는지를 플래그로 저장한다.
- [0157] d. 우상 후보(B0)가 사용 가능한지 판단한다. 우상 후보가 사용 가능하다면, numA1B1B0을 1 증가시킨다. 또한, 우상 후보가 VSP를 사용하였는지를 플래그로 저장한다.
- [0158] e. 좌하 후보(A0)가 사용 가능한지 판단한다. 좌하 후보가 사용 가능하다면, numA0B2을 1 증가시킨다. 또한, 좌하 후보가 VSP 사용하였는지를 플래그로 저장한다.
- [0159] f. 좌상 후보(B2)가 사용 가능한지 판단한다. 좌상 후보가 사용 가능하다면, numA0B2을 1 증가시킨다. 또한, 좌상 후보가 VSP를 사용하였는지를 플래그로 저장한다.
- [0160] g. 플래그 availableFlagT가 1이면 다음 과정을 수행한다.
- [0161] - pruneFlagA1, pruneFlagB1을 0으로 설정한다.
- [0162] - 좌 후보가 사용 가능하고, 텍스처 후보의 움직임 정보가 좌 후보와 같다면 pruneFlagA1을 1로 설정한다.
- [0163] - 상 후보가 사용 가능하고, 텍스처 후보의 움직임 정보가 상 후보와 같다면 pruneFlagB1을 1로 설정한다.
- [0164] - 만일 pruneFlagA1과 pruneFlagB1이 모두 0이면, 리스트의 numMergeCand 위치에 새로운 공간을 하나 만든다. 이 때, 새로운 공간을 하나 만든다는 것은 리스트 내의 numMergeCand 위치에서부터의 모든 값들을 오른쪽 값으로 한 칸씩 이동 시킨다는 것을 의미한다.
- [0165] - 그 밖의 경우, 다음 과정을 수행한다.
- [0166] numA1B1B0를 1 감소시킨다.
- [0167] 좌, 상 후보가 모두 사용 가능하고 pruneFlagA1이 0이라면 리스트의 두 번째 값을 첫 번째 값으로 설정한다.
- [0168] - 리스트의 첫 번째 값을 텍스처 후보로 설정하고 numMergeCand을 1 증가시킨다.
- [0169] h. 플래그 availableFlagIvMC가 1이면 다음 과정을 수행한다.
- [0170] - pruneFlagA1, pruneFlagB1, pruneFlagT, addIvMC를 0으로 설정한다.
- [0171] - 현재 픽처가 텍스처이면(DepthFlag=1), 다음 과정을 수행한다.
- [0172] 좌 후보가 사용 가능하고, 시점 간 후보의 움직임 정보가 좌 후보와 같다면 pruneFlagA1을 1로 설정한다.
- [0173] 상 후보가 사용 가능하고, 시점 간 후보의 움직임 정보가 상 후보와 같다면 pruneFlagB1을 1로 설정한다.
- [0174] - 현재 픽처가 깊이맵이고(DepthFlag=0), 텍스처 후보가 사용 가능하고, 시점 간 후보의 움직임 정보가 텍스처 후보와 같다면 pruneFlagT을 1로 설정한다.
- [0175] - 만일 pruneFlagA1, pruneFlagB1, pruneFlagT가 모두 0이면, 리스트의 numMergeCand 위치에 새로운 공간을 하

나 만들고, addIvMC를 1로 설정한다.

[0176] - 그렇지 않고, 현재 픽처가 텍스처이며(DepthFlag=0) 시점 간 후보의 움직임 정보가 좌 후보 또는 상 후보와 같다면, 다음 과정을 수행한다.

[0177] numA1B1B0를 1 감소시킨다.

[0178] addIvMC를 1로 설정한다.

[0179] 좌, 상 후보가 모두 사용 가능하고 pruneFlagA1이 0이라면 리스트의 두 번째 값을 첫 번째 값으로 설정한다.

[0180] - addIvMC가 1이라면, 리스트의 첫 번째 값을 텍스처 후보로 설정하고 numMergeCand을 1 증가시킨다.

[0181] i. numMergeCand에 numA1B1B0을 더한다.

[0182] j. 플래그 availableFlagIvDC가 1이면 다음 과정을 수행한다.

[0183] - 시차 병합 후보(IvDC)의 움직임 정보와 사용 가능한 좌 후보 및 상 후보를 비교한다. 그 결과, 좌 후보 및 상 후보의 움직임 정보와 모두 다르다면 병합 리스트의 numMergeCand 위치에 새로운 공간을 하나 만든 후 시차 병합 후보(IvDC)를 추가한다.

[0184] - numMergeCand을 1 증가시킨다.

[0185] k. 플래그 availableFlagVSP가 1이고, 밝기보상 프래그(ic_flag)가 0이고, 잔여 오차 신호 예측 계수(iv_res_pred_weight_idx)가 0이고, numMergeCand가 5 + 추가 병합후보 개수(NumExtraMergeCand)보다 작으면 시점 합성 병합 후보를 리스트에 추가하고 numMergeCand3DV, numMergeCand를 1 증가시킨다.

[0186] l. numMergeCand에 numA0B2를 더한다.

[0187] m. 플래그 availableFlagIvMCShift가 1이고 numMergeCand가 리스트의 최대 길이(일예로 6)보다 작으면, 다음 과정을 수행한다.

[0188] - 시점 간 병합 후보(IvMC)가 사용 가능하다면, 시점 간 병합 후보와 이동된 시점 간 병합 후보(IvMCShift)를 비교하여 서로 다르다면 리스트의 numMergeCand 위치에 새로운 공간을 하나 만든 후 이동된 시점 간 병합 후보를 추가한다.

[0189] - numMergeCand을 1 증가시킨다.

[0190] n. 플래그 availableFlagIvMCShift가 0이고, 현재 PU가 깊이정보 맵 상에 위치하지 않고, numMergeCand가 리스트의 최대 길이(일예로 6)보다 작으면, 다음 과정을 수행한다.

[0191] - 이동된 시차 예측 후보(IvDCShift)가 사용 가능하다면, 리스트의 numMergeCand 위치에 새로운 공간을 하나 만든 후 이동된 시점 간 병합 후보를 추가한다.

[0192] - numMergeCand을 1 증가시킨다.

[0193] 상기 과정 h, j에서 시점 간 병합 후보와 시차 병합 후보를 리스트의 기존 후보들과 비교 시, 리스트 내의 모든 후보들과 비교하지 않고 그 중 일부와 비교하는 방법을 사용하여 복잡도를 낮출 수 있다. 일 예로 좌 후보와 상 후보만을 비교에 사용할 수 있다.

[0194] 표 5는 기존 혼합 양방향 예측 후보를 유도하는 과정의 일예를 나타낸 것이고, 표 6은 본 발명의 실시예에 따른 3D-HEVC에서 HEVC 혼합 양방향 예측 후보 유도 과정을 재사용하는 일예를 나타낸 것이다.

표 5

H.8.5.3.2.3 Derivation process for combined bi-predictive merging candidates

Inputs to this process are:

a merging candidate list mergeCandList,

a list mergeCandIsVspFlag,

the reference indices refIdxL0N and refIdxL1N of every candidate N in mergeCandList,

the prediction list utilization flags predFlagL0N and predFlagL1N of every candidate N in mergeCandList,

the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList,

the number of elements numCurrMergeCand within mergeCandList,

the number of elements numOrigMergeCand within the mergeCandList after the spatial and temporal merge candidate derivation process.

Outputs of this process are:

the merging candidate list mergeCandList,

the number of elements numCurrMergeCand within mergeCandList,

the reference indices refIdxL0combCandk and refIdxL1combCandk of every new candidate combCandk added into mergeCandList during the invocation of this process,

the prediction list utilization flags predFlagL0combCandk and predFlagL1combCandk of every new candidate combCandk added into mergeCandList during the invocation of this process,

the motion vectors mvL0combCandk and mvL1combCandk of every new candidate combCandk added into mergeCandList during the invocation of this process.

When numOrigMergeCand is greater than 1 and less than MaxNumMergeCand, the variable numInputMergeCand is set equal to numCurrMergeCand, the variable combIdx is set equal to 0, the variable combStop is set equal to FALSE, and the following steps are repeated until combStop is equal to TRUE:

5. The variables l0CandIdx and l1CandIdx are derived using combIdx as specified in Table 8-6.

6. The following assignments are made, with l0Cand being the candidate at position l0CandIdx and l1Cand being the candidate at position l1CandIdx in the merging candidate list mergeCandList:

l0Cand = mergeCandList[l0CandIdx]

l1Cand = mergeCandList[l1CandIdx]

7. When all of the following conditions are true:

mergeCandIsVspFlag[l0CandIdx] = 0,

mergeCandIsVspFlag[l1CandIdx] = 0,

predFlagL0l0Cand == 1

[0195]

```

predFlagL111Cand == 1
( DiffPicOrderCnt( RefPicList0[ refIdxL010Cand ], RefPicList1[ refIdxL111Cand ] ) != 0
) || ( mvL010Cand != mvL111Cand )
the candidate combCandk with k equal to ( numCurrMergeCand - numInputMergeCand ) is
added at the end of mergeCandList, i.e. mergeCandList[ numCurrMergeCand ] is set equal to
combCandk, and the reference indices, the prediction list utilization flags, and the motion
vectors of combCandk are derived as follows and numCurrMergeCand is incremented by 1:

refIdxL0combCandk = refIdxL010Cand
refIdxL1combCandk = refIdxL111Cand
predFlagL0combCandk = 1
predFlagL1combCandk = 1
mvL0combCandk[ 0 ] = mvL010Cand[ 0 ]
mvL0combCandk[ 1 ] = mvL010Cand[ 1 ]
mvL1combCandk[ 0 ] = mvL111Cand[ 0 ]
mvL1combCandk[ 1 ] = mvL111Cand[ 1 ]
numCurrMergeCand = numCurrMergeCand + 1

```

8. The variable combIdx is incremented by 1.

When combIdx is equal to (cremented by 1.and is incremented by 1:- 1)) or numCurrMergeCand is equal to MaxNumMergeCand, combStop is set equal to TRUE.

[0196]

표 6

H.8.5.3.2.17 Derivation process for base luma motion vectors for merge mode

– “When slice_type is equal to B, the derivation process for combined bi-predictive merging candidates” is replaced by “When slice_type is equal to B and numMergeCand is less than 5, the derivation process for combined bi-predictive merging candidates”

[0197]

[0198]

표 6에 기술된 본 발명의 실시예에 따른 혼합 양방향 예측 후보 유도 과정에서는 표 5에서처럼 의존적인 시점이 나 깊이정보 맵만을 위한 새로운 모듈을 제작하지 않고, 기존 HEVC의 과정을 그대로 사용한다. 따라서 표 6의 방법에서는 표 5의 과정이 모두 제거 된다.

[0199]

표 6에 기술된 본 발명의 실시예에 따른 혼합 양방향 예측 후보 유도 과정을 구현하면 표 7과 같은 결과를 얻을 수 있다. 실험에 사용 된 비디오 시퀀스들은 JCT-3V 표준화에서 공식적으로 사용하는 테스트용 비디오 시퀀스들이다.

[0200]

표 7은 기존 방법(표 5의 방법)과 본 발명에서 제안하는 방법(표 6의 방법)의 부호화 효율 및 부호화 시간 비교 실험 결과를 나타낸다.

표 7

	시점 0	시점 1	시점 2	video PSNR / video bitrate	synth PSNR / total bitrate
Balloons	0.0%	0.0%	0.2%	0.0%	0.0%
Kendo	0.0%	0.1%	0.0%	0.0%	0.1%
Newspaper_CC	0.0%	0.0%	0.1%	0.0%	0.0%
GT_Fly	0.0%	0.0%	-0.1%	0.0%	0.0%
Poznan_Hall2	0.0%	0.1%	0.2%	0.0%	-0.1%
Poznan_Street	0.0%	0.0%	-0.2%	0.0%	0.0%
Undo_Dancer	0.0%	0.0%	-0.2%	0.0%	0.1%
1024x768	0.0%	0.0%	0.1%	0.0%	0.0%
1920x1088	0.0%	0.0%	-0.1%	0.0%	0.0%
average	0.0%	0.0%	0.0%	0.0%	0.0%

Shark	0.0%	-0.2%	0.1%	0.0%	0.0%
-------	------	-------	------	------	------

[0201]

[0202]

표 7에 도시된 바와 같이, 본 발명의 실시예에 따른 3D-HEVC에서 HEVC 혼합 양방향 예측 후보 유도 과정을 재사용하였음에도 비교 결과, 기존 방법에 비하여 PSNR 대비 비트율(bitrate) 증가가 0.1% 미만으로 거의 동일한 부호화 효율을 보인다.

[0203]

상술한 방법은 현재 MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)에서 공동으로 표준화를 진행 중인 HEVC(High Efficiency Video Coding)를 사용할 수 있다. 따라서 상술한 방법은 표 8의 예와 같이 블록 크기 혹은 CU(Coding Unit) 깊이 혹은 TU(Transform Unit) 깊이 등에 따라 적용 범위를 달리할 수 있다. 이렇게 적용 범위를 결정하는 변수(즉, 크기 혹은 깊이 정보)는 부호화기 및 복호화기가 미리 정해진 값을 사용하도록 설정할 수도 있고, 프로파일 또는 레벨에 따라 정해진 값을 사용하도록 할 수도 있고, 부호화기가 변수 값을 비트스트림에 기재하면 복호화기는 비트스트림으로부터 이 값을 구하여 사용할 수도 있다. CU 깊이에 따라 적용 범위를 달리하는 할 때는 표 8에 예시한 바와 같이, 방법A) 주어진 깊이 이상의 깊이에만 적용하는 방식, 방법B) 주어진 깊이 이하에만 적용하는 방식, 방법C) 주어진 깊이에만 적용하는 방식이 있을 수 있다.

[0204]

표 8에 따르면, 주어진 CU(혹은 TU) 깊이가 2인 경우, 본 발명의 방법들을 적용할 수 있다. 표 8에서 0 표시는 해당 깊이에 적용함을 나타내고 X 표시는 해당 깊이에 적용하지 않음을 나타낸다.

표 8

적용범위를 나타내는 CU(혹은 PU 혹은 TU) 깊이	방법A	방법B	방법C
0	X	O	X
1	X	O	X
2	O	O	O
3	O	X	X
4 이상	O	X	X

[0205]

[0206] 모든 값에 대하여 본 발명의 방법들을 적용하지 않는 경우는 임의의 지시자(flag)를 사용하여 나타낼 수도 있고, CU 값의 최대값보다 하나 더 큰 값을 적용범위를 나타내는 CU 값 값으로 시그널링 함으로써 표현할 수도 있다.

[0207] 본 발명의 추가적인 특징으로, 상술한 본 발명의 방법들의 적용 여부는 비트스트림에 포함되어 시그널링될 수 있다. 예컨대, 상술한 본 발명의 방법들의 적용 여부에 대한 정보는 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), 슬라이스 헤더(Slice Header)의 선택스(Syntax)에 포함되어 시그널링될 수 있다.

[0208] 표 9는 상술한 본 발명의 방법들의 적용 여부를 SPS를 이용하여 시그널링하는 방법의 일예를 나타낸다.

표 9

seq_parameter_set_rbsp() {	Descriptor
profile_idc	u(8)
reserved_zero_8bits /* equal to 0 */	u(8)
level_idc	u(8)
...	ue(v)
reuse_enabled_flag	u(1)
if(reuse_enabled_flag)	
reuse_disabled_info	ue(v)
...	

[0209]

[0210] 표 10은 상술한 본 발명의 방법들의 적용 여부를 PPS를 이용하여 시그널링하는 방법의 일예를 나타낸다.

표 10

pic_parameter_set_rbsp() {	Descriptor
pic_parameter_set_id	ue(v)
seq_parameter_set_id	ue(v)
entropy_coding_mode_flag	u(1)
...	
reuse_enabled_flag	u(1)
if(reuse_enabled_flag)	
reuse_disabled_info	ue(v)
...	

[0211]

[0212] 표 11은 상술한 본 발명의 방법들의 적용 여부를 슬라이스 헤더를 이용하여 시그널링하는 방법의 일예를 나타낸다.

표 11

slice_header() {	Descriptor
slice_type	ue(v)
pic_parameter_set_id	ue(v)
frame_num	u(v)
...	
reuse_enabled_flag	u(1)
if(reuse_enabled_flag)	
reuse_disabled_info	ue(v)
...	

[0213]

[0214]

표 12는 상술한 본 발명의 방법들의 적용 여부를 슬라이스 헤더를 이용하여 시그널링하는 방법의 다른 예를 나타낸다.

표 12

slice_header() {	Descriptor
lightweight_slice_flag	u(1)
if(!lightweight_slice_flag) {	
slice_type	ue(v)
pic_parameter_set_id	ue(v)
frame_num	u(v)
...	
}	
if(entropy_coding_mode_flag && slice_type != I)	
cabac_init_idc	ue(v)
first_slice_in_pic_flag	u(1)
...	
reuse_enabled_flag	u(1)
if(reuse_enabled_flag)	
reuse_disabled_info	ue(v)
...	

[0215]

[0216]

표 9 내지 표 12에서, “reuse_enabled_flag”는 상술한 본 발명의 방법들의 적용 여부를 알려준다. 여기서, 상술한 본 발명의 방법들이 적용되었을 경우 “reuse_enabled_flag”가 ‘1’이 되며, 상술한 본 발명의 방법들이 적용되지 않았을 경우 “reuse_enabled_flag”가 ‘0’이 된다. 그 반대도 가능하다.

[0217]

“reuse_disabled_info”는 상술한 본 발명의 방법들이 적용되었을 경우(혹은 “reuse_enabled_flag”이 참일 경우) 활성화되는 syntax로써, 이는 CU의 깊이(depth) (혹은 CU의 크기(size) 혹은 매크로 블록의 크기 혹은 서

브 매크로 블록의 크기 혹은 블록의 크기)에 따라 상술한 본 발명의 방법들의 적용 여부를 알려준다.

- [0218] 일례로, “reuse_disabled_info”가 ‘0’ 일 경우에는 상술한 본 발명의 방법들이 모든 블록 크기에 적용될 수 있다. “reuse_disabled_info”가 ‘1’ 일 경우에는 상술한 본 발명의 방법들이 4x4 블록 크기보다 큰 단위에만 적용할 수 있다.
- [0219] 다른 예로, “reuse_disabled_info”가 ‘2’ 일 경우에는 상술한 본 발명의 방법들이 8x8 블록 크기보다 큰 단위에만 적용할 수 있다. 혹은 그 역도 가능하다. 예를 들어 “reuse_disabled_info”가 ‘1’ 일 경우에는 상술한 본 발명의 방법들이 4x4 블록 크기보다 작은 단위에만 적용할 수 있다. 이러한 “reuse_disabled_info” 선택스(syntax)의 사용 방법은 다양하게 적용할 수 있다.
- [0220] 위의 방법을 이용하여 픽처(혹은 프레임) 단위로 적용 여부를 결정할 수 있다. 또한 P 픽처(혹은 프레임)에만 본 명세서의 방법이 적용될 수 있으며, B 픽처(혹은 프레임)에만 본 명세서의 방법이 적용될 수 있다.
- [0221] 상술한 본 발명의 방법들은 3차원 비디오 코덱뿐만 아니라 스케일러블 비디오 코덱에도 적용할 수 있다. 일례로, 스케일러블 비디오 코덱의 기본 계층에서 사용하는 부호화/복호화 모듈을 그대로 향상 계층에 적용한 후, 추가적으로 부분 부호화/복호화 모듈을 이용하여 향상 계층을 부호화/복호화할 수 있다. 다른 예로, 스케일러블 비디오 코덱의 기본 계층에서 사용하는 “기본 병합 움직임 리스트 구성” 모듈을 그대로 향상 계층에 적용하여 “기본 병합 움직임 후보 리스트”를 구성한 후, 추가적으로 “추가 병합 움직임 리스트 구성” 모듈을 이용하여 “기본 병합 움직임 후보 리스트”를 재설정(변경)하여 향상 계층을 위한 “확장 병합 움직임 후보 리스트”를 구성할 수 있다.
- [0222] 도 18은 본 발명의 일 실시예에 따라 복수의 시점을 포함하는 비디오 부호화/복호화 시 병합 움직임 후보 리스트를 구성하는 방법을 개략적으로 나타내는 순서도이다.
- [0223] 도 18의 방법은 상술한 도 10 및 도 12에서 도시된 장치에 의해 수행될 수 있으며, 또는 3D-HEVC에 적용하여 수행될 수도 있다. 설명의 편의 상, 도 18의 방법은 병합 움직임 장치에 의해 수행되는 것으로 설명한다.
- [0224] 도 18을 참조하면, 병합 움직임 장치는 현재 PU에 대한 기본 병합 움직임 후보를 유도하고, 유도된 기본 병합 움직임 후보를 기반으로 병합 움직임 후보 리스트를 구성한다(S1800).
- [0225] 기본 병합 움직임 후보는, 상술한 바와 같이, 현재 PU에 대한 공간적 병합 움직임 후보 및 시간적 병합 움직임 후보를 포함할 수 있다.
- [0226] 예컨대, 도 8에 도시된 바와 같이, 병합 움직임 장치는 현재 PU에 공간적으로 인접하여 위치한 좌(left) 블록, 상(above) 블록, 우상(above-right) 블록, 좌하(bottom-left) 블록 및 좌상(above-left) 블록 중 적어도 하나의 블록으로부터 공간적 병합 움직임 후보를 유도할 수 있다. 그리고 병합 움직임 장치는 현재 PU에 대한 동일 위치 픽처(co-located picture) 내 동일 위치 블록(co-located block)(예를 들어, 하단 우측(bottom right) 블록, 중앙(center) 블록)으로부터 시간적 병합 움직임 후보를 유도할 수 있다.
- [0227] 병합 움직임 장치는, 상술한 바와 같이, 공간적 병합 움직임 후보 및 시간적 병합 움직임 후보의 가용성 여부를 기반으로 병합 움직임 후보 리스트를 구성할 수 있다.
- [0228] 현재 PU가 깊이정보 맵 또는 의존적인 시점인 경우, 병합 움직임 장치는 현재 PU에 대한 확장 병합 움직임 후보를 유도한다(S1810).
- [0229] 확장 병합 움직임 후보는, 의존적인 시점의 영상 또는 깊이정보 맵 영상에 대한 예측을 위해 사용되는 병합 움직임 후보를 말한다. 확장 병합 움직임 후보는 시점 간 병합 후보(inter-view merge candidate: IvMC), 시점 합성 병합 후보(view synthesis prediction merge candidate) 및 텍스처 병합 후보(texture merge candidate) 중 적어도 하나를 포함할 수 있다.
- [0230] 예를 들어, 현재 PU가 시점 간 예측을 수행하는지 여부에 따라 시점 간 병합 후보, 시차 병합 후보(inter-view disparity merge candidate; IvDC), 이동된 시점 간 병합 후보(shifted inter-view merge candidate: IvMCShift), 이동된 시차 병합 후보(shifted inter-view disparity merge candidate; IvDCShift)를 유도할 수 있다. 현재 PU가 시점 합성 예측을 수행하는지 여부에 따라 시점 합성 병합 후보를 유도할 수 있다. 현재 PU의

깊이정보 맵이 텍스처 블록으로부터 움직임 예측을 수행하는지 여부에 따라 텍스처 병합 후보를 유도할 수 있다.

[0231] 병합 움직임 장치는 상기 유도된 확장 병합 움직임 후보를 병합 움직임 후보 리스트에 추가함으로써, 최종적으로 병합 움직임 후보 리스트를 재구성할 수 있다(S1820).

[0232] 이때, 추가할 확장 병합 움직임 후보가 병합 움직임 리스트 내 기본 병합 움직임 후보와 동일하지 않은 경우, 병합 움직임 장치는 확장 병합 움직임을 병합 움직임 후보 리스트에 추가한다. 확장 병합 움직임 후보는 병합 움직임 후보 리스트 내 임의의 위치(예컨대, 리스트의 첫번째 항목)에 추가될 수 있다.

[0233] 또한, 병합 움직임 후보 리스트에 추가된 확장 병합 움직임 후보 개수와 기본 병합 움직임 후보 개수를 더한 값이 병합 움직임 후보 리스트의 최대 후보 개수보다 작은 경우, 병합 움직임 장치는 확장 병합 움직임을 병합 움직임 후보 리스트에 추가한다.

[0234] 예를 들어, 현재 PU의 깊이정보 맵이 텍스처 블록으로부터 움직임 예측을 수행하는 경우, 텍스처 병합 후보가 유도될 수 있다. 이때, 유도된 텍스처 병합 후보가 병합 움직임 리스트 내 기본 병합 움직임 후보와 동일하지 않은 경우, 텍스처 병합 후보를 병합 움직임 리스트 내 첫번째 항목에 추가할 수 있다.

[0235] 현재 PU가 시점 간 예측을 수행하는 경우, 시점 간 병합 후보가 유도될 수 있다. 이때, 유도된 시점 간 병합 후보가 병합 움직임 리스트 내 기본 병합 움직임 후보와 동일하지 않은 경우, 시점 간 병합 후보를 병합 움직임 리스트 내 첫번째 항목에 추가할 수 있다.

[0236] 현재 PU가 시점 합성 예측을 수행하는 경우, 시점 합성 병합 후보가 유도될 수 있다. 이때, 병합 움직임 후보 리스트에 추가된 확장 병합 움직임 후보 개수와 기본 병합 움직임 후보 개수를 더한 값이 병합 움직임 후보 리스트의 최대 후보 개수보다 작은 경우, 유도된 시점 합성 병합 후보가 병합 움직임 후보 리스트에 추가될 수 있다.

[0237] 확장 병합 움직임을 병합 움직임 후보 리스트에 추가하는 구체적인 과정은 본 명세서의 실시예들에서 상세히 설명한 바 있으므로, 본 실시예에서는 구체적인 설명을 생략한다.

[0238] 상술한 병합 움직임 후보 리스트를 기반으로 현재 PU에 대한 움직임 정보를 획득할 수 있으며, 상기 움직임 정보를 이용하여 현재 PU에 대한 예측을 수행함으로써 현재 PU의 예측 샘플 값을 획득할 수 있다.

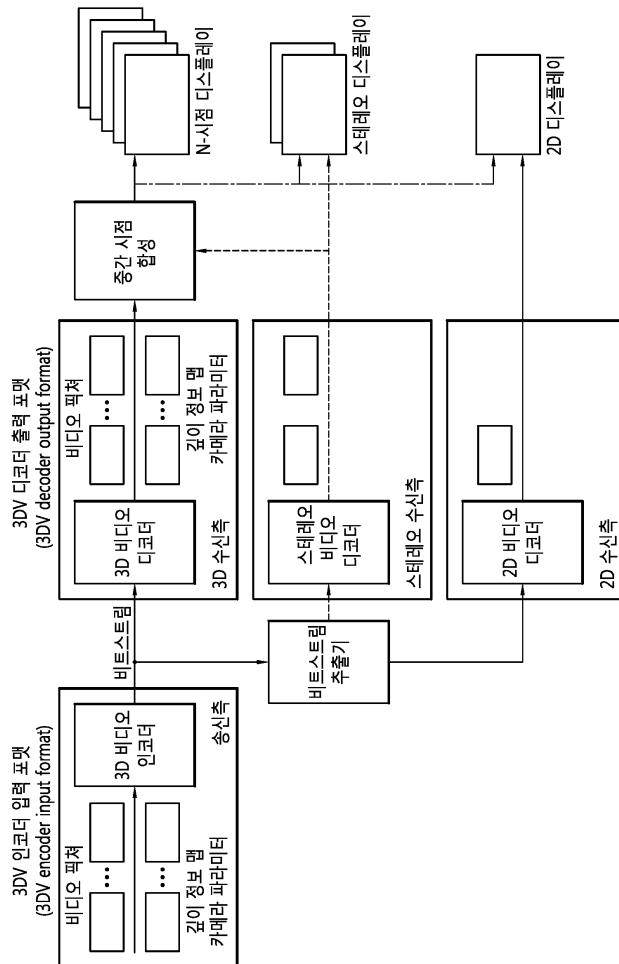
[0239] 따라서, 부호화기에서는 현재 PU의 예측 샘플 값을 기반으로 현재 PU의 잔차(residual) 샘플 값을 획득하고, 잔차 샘플 값을 변환/양자화 및 엔트로피 부호화하여 복호화기로 전송할 수 있다. 복호화기에서는 현재 PU의 예측 샘플 값과 부호화기에 의해 전송된 현재 PU의 잔차 샘플 값을 기반으로 현재 PU의 복원 샘플 값을 얻을 수 있다.

[0240] 상술한 실시예들에서, 방법들은 일련의 단계 또는 블록으로서 순서도를 기초로 설명되고 있으나, 본 발명은 단계들의 순서에 한정되는 것은 아니며, 어떤 단계는 상술한 바와 다른 단계와 다른 순서로 또는 동시에 발생할 수 있다. 또한, 당해 기술 분야에서 통상의 지식을 가진 자라면 순서도에 나타난 단계들이 배타적이지 않고, 다른 단계가 포함되거나, 순서도의 하나 또는 그 이상의 단계가 본 발명의 범위에 영향을 미치지 않고 삭제될 수 있음을 이해할 수 있을 것이다.

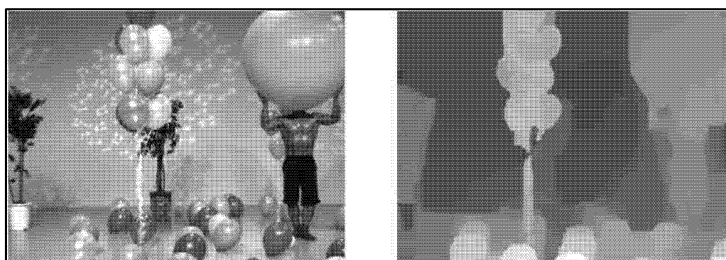
[0241] 이상의 설명은 본 발명의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명에 개시된 실시 예들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시 예에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호범위는 특허청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

도면

도면1



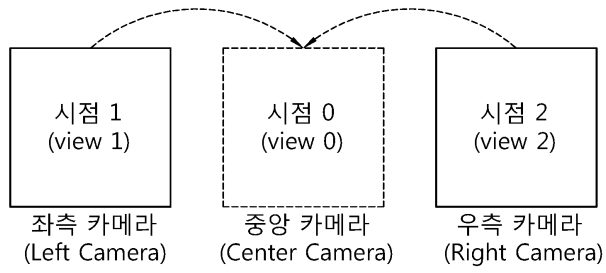
도면2



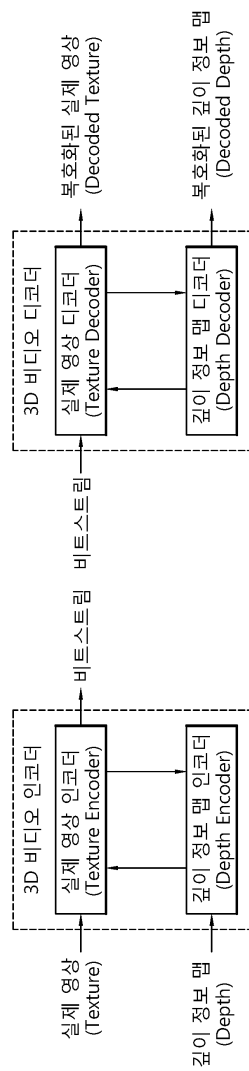
(a) 실제 영상

(b) 깊이 정보 맵

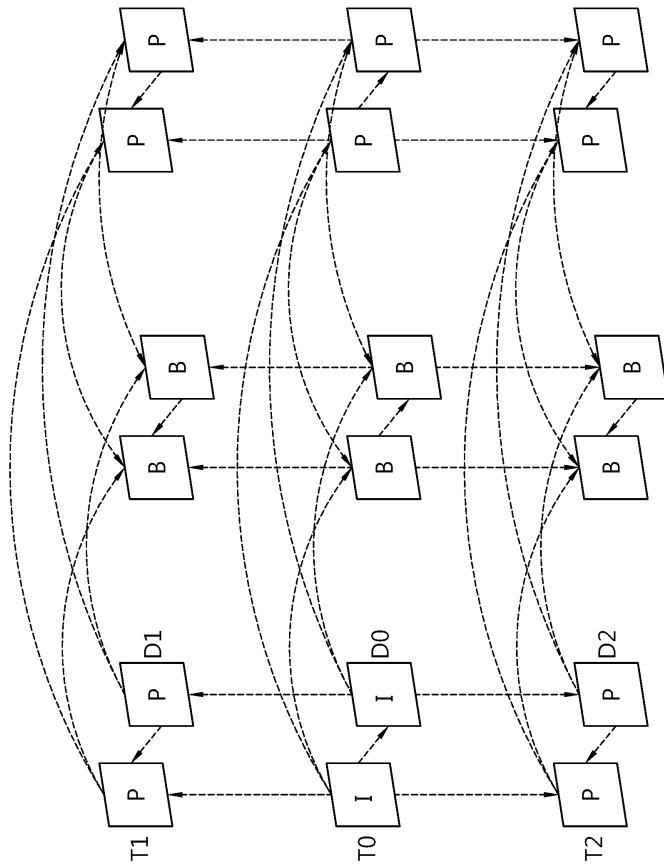
도면3



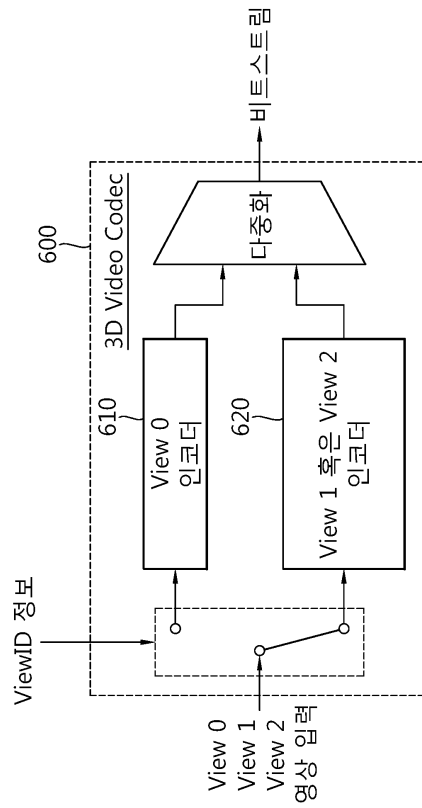
도면4



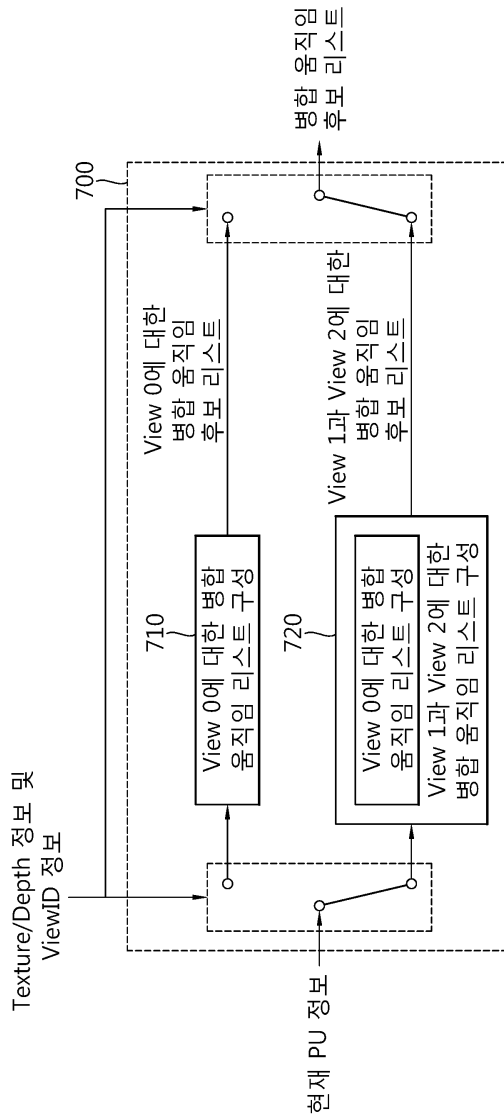
도면5



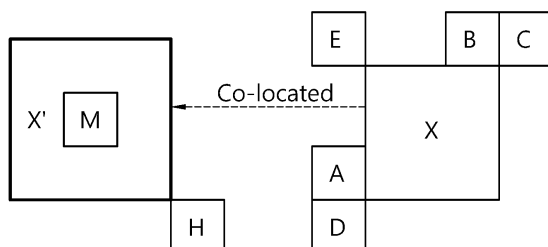
도면6



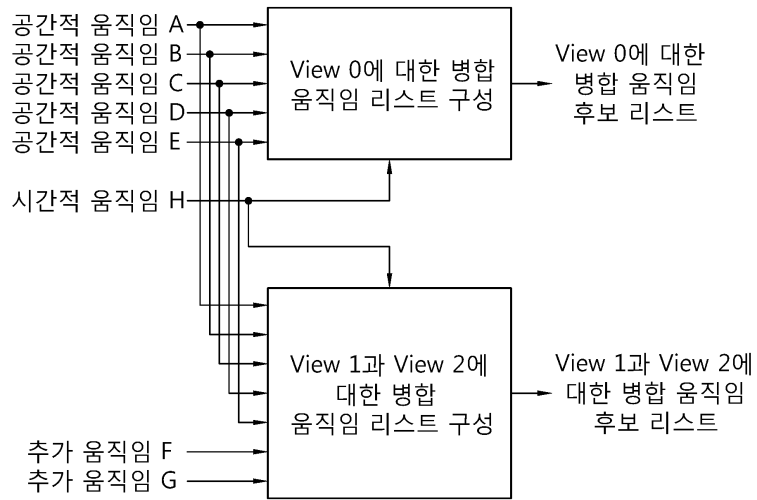
도면7



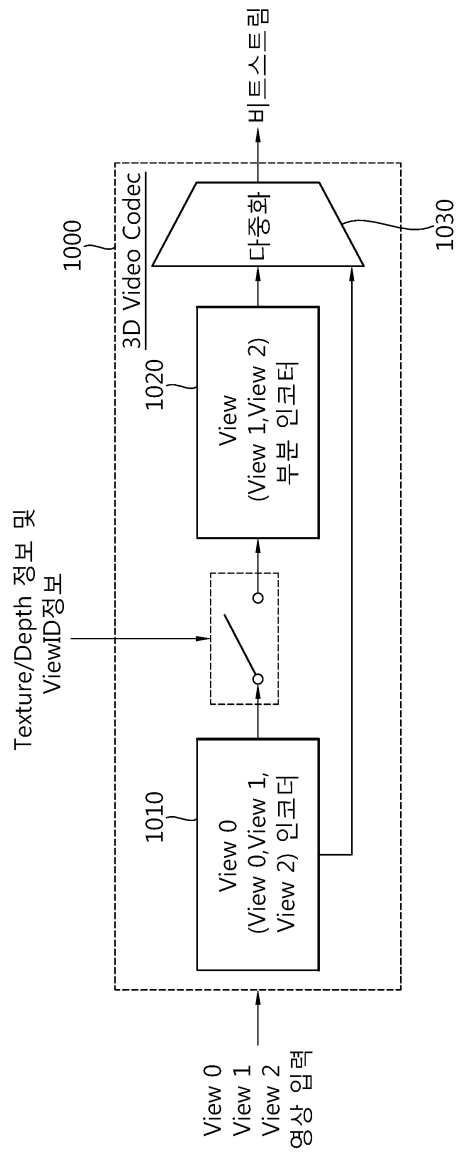
도면8



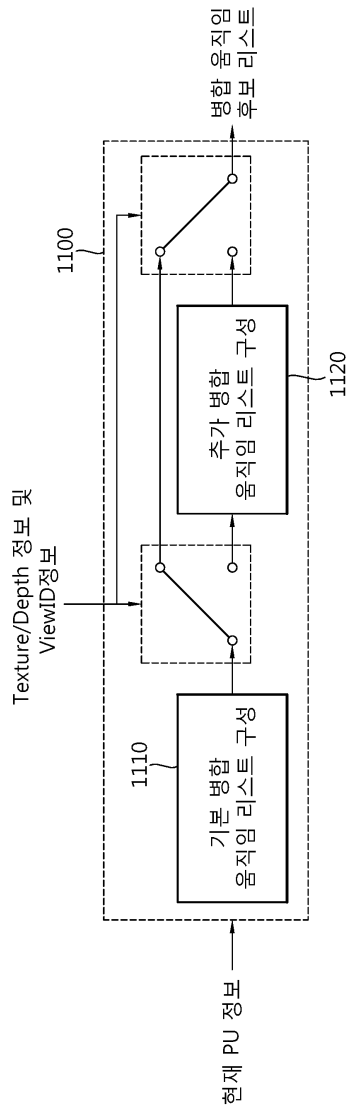
도면9



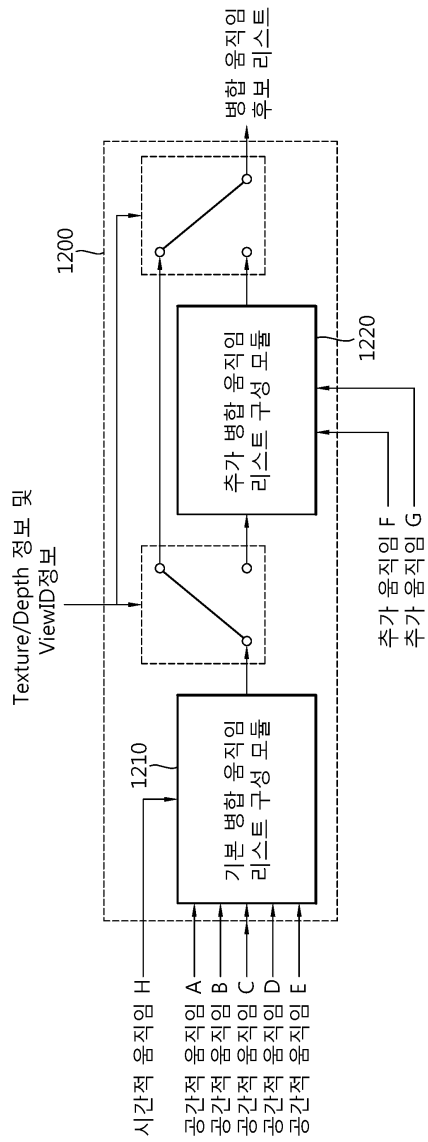
도면10



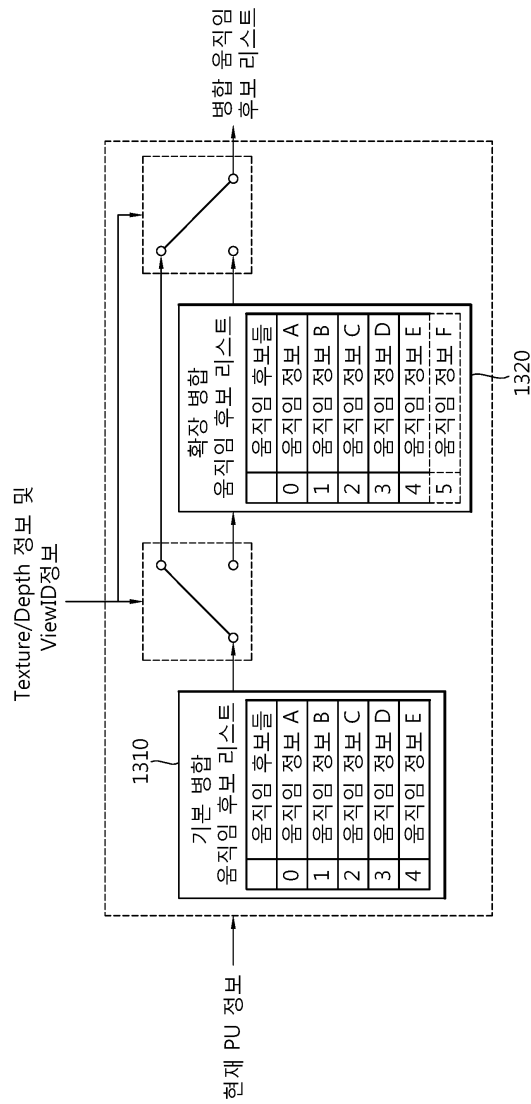
도면11



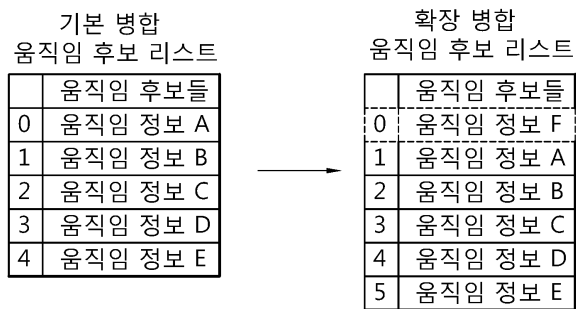
도면12



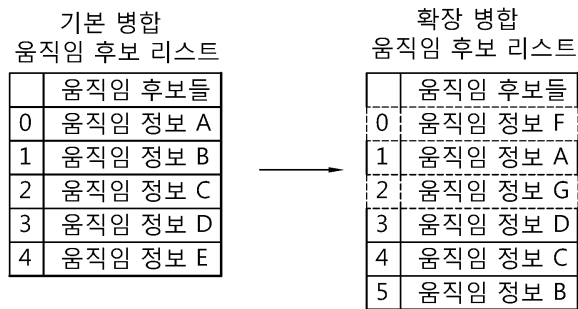
도면13



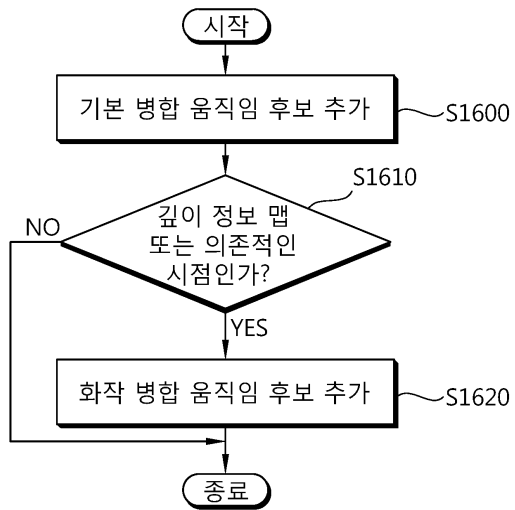
도면14



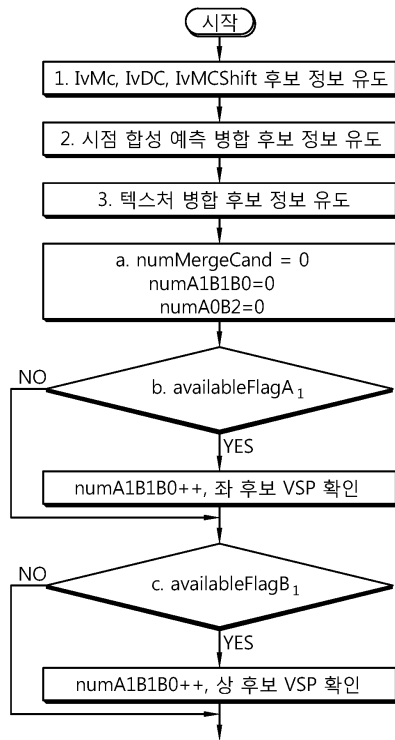
도면15



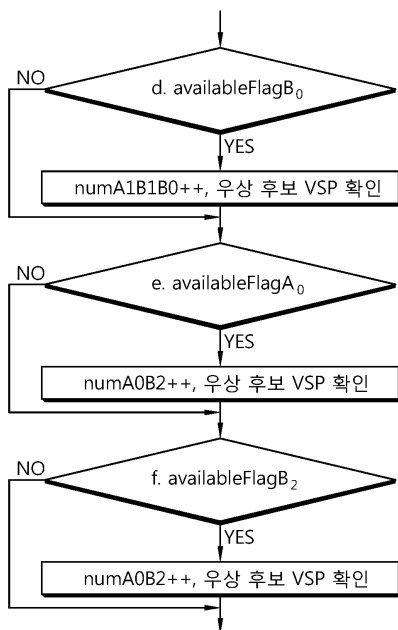
도면16



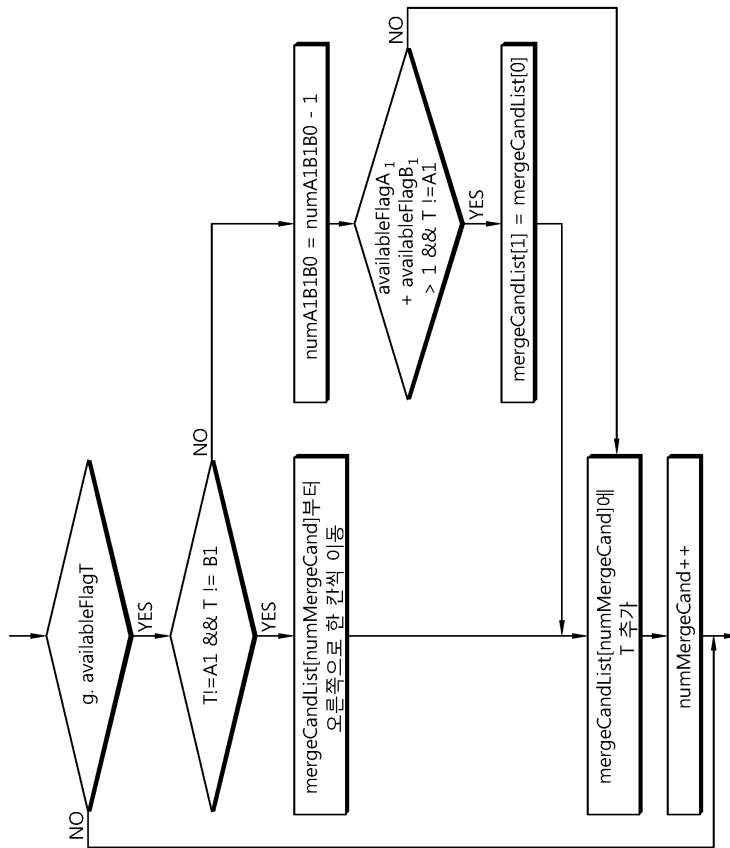
도면17a



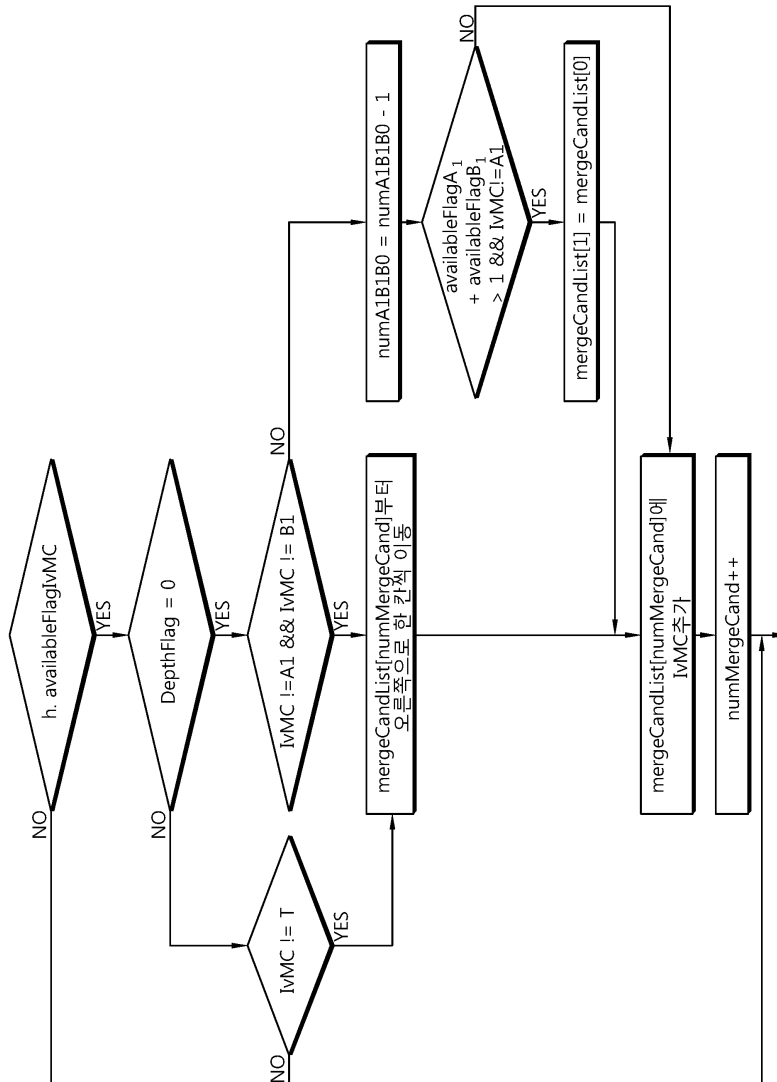
도면17b



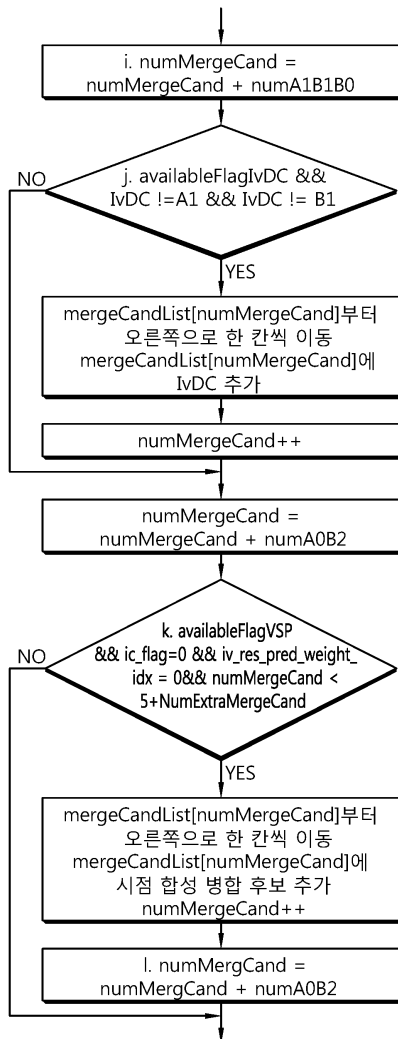
도면17c



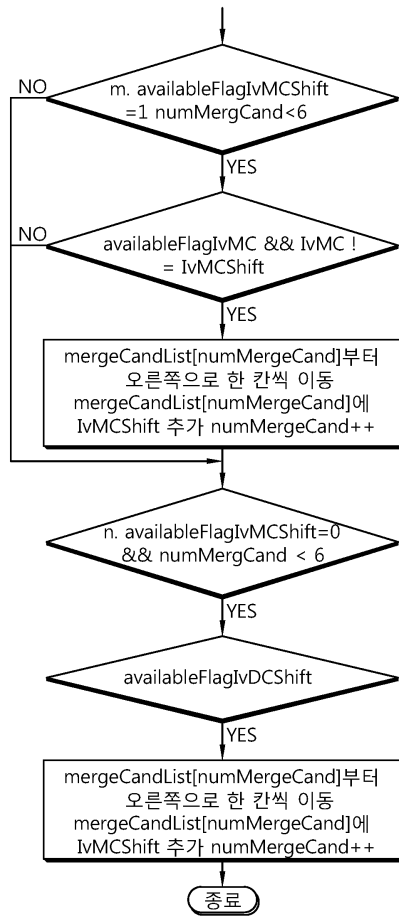
도면17a



도면17e



도면17f



도면18

