



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0030147
(43) 공개일자 2016년03월16일

- | | |
|--|---|
| <p>(51) 국제특허분류(Int. Cl.)
H04N 19/597 (2014.01) H04N 13/02 (2006.01)
H04N 13/04 (2006.01) H04N 5/222 (2006.01)</p> <p>(52) CPC특허분류
H04N 19/597 (2015.01)
H04N 13/0271 (2013.01)</p> <p>(21) 출원번호 10-2016-0023073(분할)
(22) 출원일자 2016년02월26일
심사청구일자 없음</p> <p>(62) 원출원 특허 10-2014-0059065
원출원일자 2014년05월16일
심사청구일자 2014년05월16일</p> <p>(30) 우선권주장
1020140037277 2014년03월28일 대한민국(KR)</p> | <p>(71) 출원인
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)</p> <p>(72) 발명자
박광훈
경기도 성남시 분당구 예원로 7, B동 302호(분당동, 동아빌라)
김태욱
경기도 수원시 영통구 청명로59번길 45-22(영통동)
이윤진
경기도 용인시 기흥구 흥덕2로 80, 309동 1502호(영덕동, 휴먼시아 흥덕마을3단지)</p> <p>(74) 대리인
김연권</p> |
|--|---|

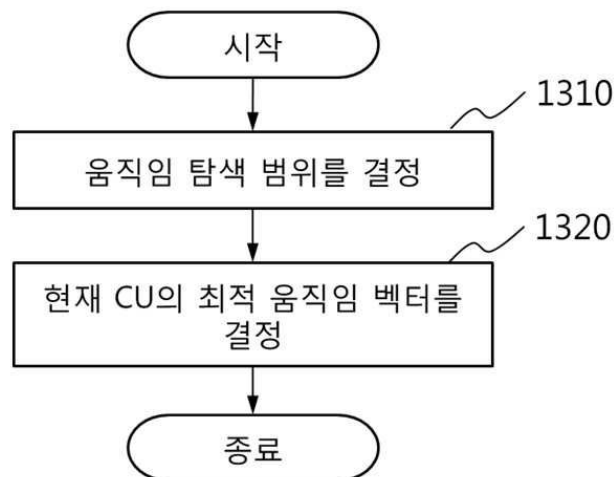
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 깊이 정보를 이용한 비디오 부호화 장치 및 그 방법

(57) 요약

깊이정보를 이용한 비디오 부호화 장치 및 그 방법을 개시한다. 일 실시예에 따른 깊이 정보를 이용한 영상 부호화 방법은, 현재 코딩 유닛(CU: Coding Unit)의 객체 정보에 기초하여, 상기 현재 CU의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역에서 움직임 탐색 범위를 결정하는 단계; 및 상기 움직임 탐색 범위 내의 움직임 정보에 기초하여 상기 현재 CU의 최적 움직임 벡터를 결정하는 단계를 포함한다.

대표도 - 도13



(52) CPC특허분류

H04N 13/0495 (2013.01)

H04N 5/2226 (2013.01)

명세서

청구범위

청구항 1

현재 코딩 유닛(CU: Coding Unit)의 객체 정보에 기초하여, 상기 현재 CU의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역에서 움직임 탐색 범위를 결정하는 단계; 및

상기 움직임 탐색 범위 내의 움직임 정보에 기초하여 상기 현재 CU의 최적 움직임 벡터를 결정하는 단계를 포함하는 깊이 정보를 이용한 영상 부호화 방법.

청구항 2

제1항에 있어서,

상기 움직임 탐색 범위는 상기 현재 CU를 구성하는 객체와 동일 객체 영역임을 특징으로 하는 깊이 정보를 이용한 영상 부호화 방법.

청구항 3

제1항에 있어서,

상기 움직임 탐색 범위를 결정하는 단계는,

깊이 영상으로부터 현재 코딩 유닛(CU: Coding Unit)의 깊이 정보를 확인하는 단계;

상기 깊이 정보에 기초하여 상기 현재 CU의 객체 정보를 결정하는 단계;

상기 CU의 객체 정보에 기초하여 상기 현재 CU가 단일객체로 이루어지는 지를 판단하는 단계; 및

상기 현재 CU가 단일객체로 이루어지는 경우, 상기 전체 움직임 탐색 영역 중 상기 CU의 단일 객체와 동일하지 않은 영역을 상기 움직임 탐색 범위에서 제외하는 단계를 포함하는

깊이 정보를 이용한 영상 부호화 방법.

청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 객체 정보는 깊이 영상을 라벨링 알고리즘에 의해 라벨링한 객체 정보 영상의 라벨 값 또는 상기 깊이 영상으로부터 획득되는 깊이 값 분포 정보임을 특징으로 하는

깊이 정보를 이용한 영상 부호화 방법.

청구항 5

제4항에 있어서,

상기 현재 CU의 깊이 값 분포 정보에 포함된 최대값 및 최소값의 차가 기 설정된 값 미만이면, 상기 현재 CU가 단일 객체로 이루어지는 것으로 판단되는 것을 특징으로 하는

깊이 정보를 이용한 영상 부호화 방법.

청구항 6

제1항에 있어서,

상기 움직임 탐색 범위를 결정하는 단계는,

상기 현재 CU의 크기 및 상기 현재 CU의 깊이 값 분포 정보에 포함된 최대값 및 최소값에 기초하여 상기 현재

CU가 단일 객체로 이루어지는 지를 판단하는 과정을 포함하는

깊이 정보를 이용한 영상 부호화 방법.

청구항 7

현재 코딩 유닛(CU: Coding Unit)의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역을 확인하는 단계;

깊이 영상으로부터 추출되는 상기 현재 CU의 깊이 정보에 따라 현재 움직임 탐색 위치에서의 움직임 탐색 여부를 결정하는 단계; 및

결정된 움직임 탐색 여부를 고려하여 상기 현재 CU의 최적 움직임 벡터를 결정하는 단계

를 포함하는 깊이 정보를 이용한 영상 부호화 방법.

청구항 8

제7항에 있어서,

상기 현재 움직임 탐색 위치에서의 움직임 탐색 여부를 결정하는 단계는,

상기 현재 CU의 크기 및 상기 현재 CU의 깊이 값 분포 정보에 포함된 최대값 및 최소값에 기초하여 상기 현재 CU가 단일 객체로 이루어지는 지를 판단하는 과정을 포함하는

깊이 정보를 이용한 영상 부호화 방법.

청구항 9

제7항에 있어서,

상기 현재 움직임 탐색 위치에서의 움직임 탐색 여부를 결정하는 단계는,

상기 현재 CU의 크기가 기설정된 값 이상이고 상기 현재 CU의 네 모서리 깊이 값들 중 최대값 및 최소값의 차가 기 설정된 기준값 이하이면, 상기 현재 CU의 네 모서리 깊이 값들 중 어느 하나를 저장하는 단계; 및

상기 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 같으면 현재 위치를 탐색하는 것으로 결정하고, 상기 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 다르면 현재 위치를 탐색하지 않는 것으로 결정하는 단계

를 포함하는 깊이 정보를 이용한 영상 부호화 방법.

청구항 10

제7항에 있어서,

상기 현재 움직임 탐색 위치에서의 움직임 탐색 여부를 결정하는 단계는,

상기 현재 CU의 크기가 기설정된 값 보다 작고 상기 현재 CU의 네 모서리 깊이 값들이 모두 같은 값이면, 상기 현재 CU의 네 모서리 깊이 값들 중 어느 하나를 저장하는 단계; 및

상기 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 같으면 현재 위치를 탐색하는 것으로 결정하고, 상기 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 다르면 현재 위치를 탐색하지 않는 것으로 결정하는 단계

를 포함하는 깊이 정보를 이용한 영상 부호화 방법.

청구항 11

현재 코딩 유닛(CU: Coding Unit)의 객체 정보에 기초하여, 상기 현재 CU의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역에서 움직임 탐색 범위를 결정하는 움직임 탐색 범위 결정부; 및

상기 움직임 탐색 범위 내의 움직임 정보에 기초하여 상기 현재 CU의 최적 움직임 벡터를 결정 움직임 벡터 결정부

를 포함하는 깊이 정보를 이용한 영상 부호화 장치.

청구항 12

제11항에 있어서,

상기 움직임 탐색 범위는 상기 현재 CU를 구성하는 객체와 동일 객체 영역임을 특징으로 하는

깊이 정보를 이용한 영상 부호화 장치.

청구항 13

제11항에 있어서,

상기 움직임 탐색 범위 결정부는,

상기 현재 CU가 단일객체로 이루어지는 경우, 상기 전체 움직임 탐색 영역 중 상기 CU의 단일 객체와 동일하지 않은 영역을 상기 움직임 탐색 범위에서 제외하는

깊이 정보를 이용한 영상 부호화 장치.

청구항 14

제13항에 있어서,

상기 움직임 탐색 범위 결정부는,

상기 현재 CU의 크기 및 상기 현재 CU의 깊이 값 분포 정보에 포함된 최대값 및 최소값에 기초하여 상기 현재 CU가 단일 객체로 이루어지는 지를 판단하는

깊이 정보를 이용한 영상 부호화 장치.

청구항 15

현재 코딩 유닛(CU: Coding Unit)의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역을 확인하고, 깊이 영상으로부터 추출되는 상기 현재 CU의 깊이 정보에 따라 현재 움직임 탐색 위치에서의 움직임 탐색 여부를 결정하는 움직임 탐색 여부 결정부; 및

상기 현재 CU의 최적 움직임 벡터를 결정하는 움직임 벡터 결정부

를 포함하는 깊이 정보를 이용한 영상 부호화 장치.

청구항 16

제15항에 있어서,

상기 움직임 탐색 범위 결정부는,

상기 현재 CU의 크기가 기설정된 값 이상이고 상기 현재 CU의 네 모서리 깊이 값들 중 최대값 및 최소값의 차이가 설정된 기준값 이하이면, 상기 현재 CU의 네 모서리 깊이 값들 중 어느 하나를 저장하고,

상기 현재 CU의 크기가 기설정된 값 보다 작고 상기 현재 CU의 네 모서리 깊이 값들이 모두 같은 값이면, 상기 현재 CU의 네 모서리 깊이 값들 중 어느 하나를 저장하는

깊이 정보를 이용한 영상 부호화 장치.

청구항 17

제16항에 있어서,

상기 움직임 탐색 범위 결정부는,

상기 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 같으면 현재 위치를 탐색하는 것으로 결정하고, 상기 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 다르면 현재 위치를 탐색하지 않는 것으로 결정하는

깊이 정보를 이용한 영상 부호화 장치.

발명의 설명

기술 분야

[0001] 본 발명은 깊이 정보를 이용한 비디오 부호화에 관한 것으로, 깊이 정보를 이용하여 객체 정보를 유도하고 영상을 효율적으로 부호화하는 방법 및 장치에 관한 것이다.

배경 기술

[0002] 깊이 정보 영상은 3차원 비디오 부호화에서 널리 활용되고 있으며, Xbox 게임기에서의 키넥트(Kinect) 카메라, 인텔 SENZ3D 웹캠, iPad에서의 iSense 3D 스캐너, 구글 Tango 스마트폰 등과 같은 새로운 입력장치들에 구비된 깊이 정보 카메라는 여러 다양한 3D 및 2D 응용 어플리케이션에서 활용될 수 있다.

[0003] 한편, 깊이 정보 카메라의 대중화 및 보급으로 2D/3D 응용 어플리케이션은 더욱 다양한 2D/3D 응용 서비스를 통해 대중화가 진행 중에 있으며, 그에 따라 향후 멀티미디어 카메라 시스템에 깊이 정보 카메라가 포함되어 다양한 정보의 활용이 가능하다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 미국공개특허 제20140085416호(발명의 명칭: Method and apparatus of texture image compress in 3d video coding)

(특허문헌 0002) 한국공개특허 제2012-0137305호(발명의 명칭: 블록 분할 방법 및 이러한 방법을 사용하는 장치)

(특허문헌 0003) 한국공개특허 제2014-0048784호(발명의 명칭: 제한적 깊이정보값을 공유하여 움직임 정보를 유도하는 방법 및 장치)

발명의 내용

해결하려는 과제

[0005] 본 발명의 목적은 2차원 비디오 부호화시 깊이 정보를 이용함으로써 성능 열화가 없고 효율적인 부호화가 가능한 영상 부호화 방법 및 장치를 제공하는 것이다.

과제의 해결 수단

[0006] 일 실시예에 따른 깊이 정보를 이용한 영상 부호화 방법은, 현재 코딩 유닛(CU: Coding Unit)의 객체 정보에 기초하여, 상기 현재 CU의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역에서 움직임 탐색 범위를 결정하는 단계; 및 상기 움직임 탐색 범위 내의 움직임 정보에 기초하여 상기 현재 CU의 최적 움직임 벡터를 결정하는 단계를 포함한다.

[0007] 다른 일 실시예에 따른 깊이 정보를 이용한 영상 부호화 방법은, 현재 코딩 유닛(CU: Coding Unit)의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역을 확인하는 단계; 깊이 영상으로부터 추출되는 상기 현재 CU의 깊이 정보에 따라 현재 움직임 탐색 위치에서의 움직임 탐색 여부를 결정하는 단계; 및 결정된 움직임 탐색 여부를 고려하여 상기 현재 CU의 최적 움직임 벡터를 결정하는 단계를 포함한다.

[0008] 일 실시예에 따른 깊이 정보를 이용한 영상 부호화 장치는, 현재 코딩 유닛(CU: Coding Unit)의 객체 정보에 기초하여, 상기 현재 CU의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역에서 움직임 탐색 범위를 결정하는 움직임 탐색 범위 결정부; 및 상기 움직임 탐색 범위 내의 움직임 정보에 기초하여 상기 현재 CU의 최적 움직임 벡터를 결정 움직임 벡터 결정부를 포함한다.

[0009] 다른 일 실시예에 따른 깊이 정보를 이용한 영상 부호화 장치는, 현재 코딩 유닛(CU: Coding Unit)의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역을 확인하고, 깊이 영상으로부터 추출되는 상기 현재 CU의 깊이 정보에 따라 현재 움직임 탐색 위치에서의 움직임 탐색 여부를 결정하는 움직임 탐색 여부 결정부; 및 상기 움직임 탐색

색 범위 내의 움직임 정보에 기초하여 상기 현재 CU의 최적 움직임 벡터를 결정하는 움직임 벡터 결정부를 포함한다.

발명의 효과

[0010] 본 발명의 실시예에 따르면, 깊이 정보 카메라에서 획득한 깊이 정보 영상을 이용해 2차원 일반 영상을 부호화함으로써, 2D 영상에 대한 효율적인 부호화를 수행할 수 있다.

도면의 간단한 설명

[0011] 도 1은 일반 영상 및 일반 영상의 깊이 정보 맵에 대한 예시도이다.
 도 2는 키넥트 입력 장치의 예시도이다.
 도 3은 웹캠 제품을 나타낸다.
 도 4는 iSense 3D 스캐너 장치를 나타낸다.
 도 5는 구글 Tango 스마트폰을 나타낸다.
 도 6은 HEVC 부호화 장치를 설명하기 위한 도면이다.
 도 7은 스마트 폰에서 HEVC 인코더가 적용된 영상 부호화의 예를 설명하기 위한 도면이다.
 도 8은 스마트 폰에서 깊이 영상이 포함된 HEVC의 예를 설명하기 위한 도면이다.
 도 9는 영상의 복수의 유닛들로 분할하는 예를 나타낸다.
 도 10은 종래 기술에 따른 참조 블록으로부터 현재 블록의 움직임 정보를 구하는 방법의 예를 나타낸다.
 도 11은 종래 기술에 따른 움직임 벡터를 구하는 방법의 예를 나타낸다.
 도 12는 깊이 영상의 예를 나타낸다.
 도 13은 본 발명의 일 실시예에 따른 움직임 벡터 결정 방법을 설명하기 위한 흐름도이다.
 도 14는 본 발명의 다른 일 실시예에 따른 움직임 벡터 결정 방법을 설명하기 위한 흐름도이다.
 도 15는 본 발명의 일 실시예에 따른 움직임 탐색 방법의 예를 설명하기 위한 도면이다.
 도 16은 본 발명의 일 실시예에 따른 깊이 정보를 이용한 움직임 탐색 방법의 예를 설명하기 위한 도면이다.
 도 17은 본 발명의 일 실시예에 따른 영상 부호화 장치의 구성을 설명하기 위한 도면이다.
 도 18은 CU의 깊이 값 분포 예들을 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0012] 이하의 내용은 단지 본 발명의 원리를 예시한다. 그러므로 당업자는 비록 본 명세서에 명확히 설명되거나 도시되지 않았지만 본 발명의 원리를 구현하고 본 발명의 개념과 범위에 포함된 다양한 장치를 발명할 수 있는 것이다. 또한, 본 명세서에 열거된 모든 조건부 용어 및 실시예들은 원칙적으로, 본 발명의 개념이 이해되도록 하기 위한 목적으로만 명백히 의도되고, 이와 같이 특별히 열거된 실시예들 및 상태들에 제한적이지 않는 것으로 이해되어야 한다.

[0013] 예를 들어, 본 명세서의 블록도는 본 발명의 원리를 구체화하는 예시적인 회로의 개념적인 관점을 나타내는 것으로 이해되어야 한다. 이와 유사하게, 모든 흐름도, 상태 변환도, 의사 코드 등은 컴퓨터가 판독 가능한 매체에 실질적으로 나타낼 수 있고 컴퓨터 또는 프로세서가 명백히 도시되었는지 여부를 불문하고 컴퓨터 또는 프로세서에 의해 수행되는 다양한 프로세스를 나타내는 것으로 이해되어야 한다.

[0014] 이하, 첨부된 도면을 참조하여 본 발명에 따른 실시예들을 상세히 설명하기로 한다.

[0015] 도 1은 일반 영상 및 일반 영상의 깊이 정보 맵에 대한 예시도이다. 도 2는 키넥트 입력 장치의 예시도이다.

[0016] 도 1을 참조하면, 도 1의 (A)는 카메라를 통해 실제 촬영한 영상이고, (B)는 실제 영상에 대한 깊이 영상, 즉 깊이 정보 영상(또는 깊이 정보 맵)을 나타낸다. 깊이 정보(Depth Information)는 카메라와 실제 사물간의 거리

를 나타내는 정보를 의미한다.

- [0017] 이러한 깊이 정보 영상은 주로 3차원 가상 시점 영상을 생성하는데 활용되며, 실제 이와 관련된 연구로 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 표준화가 현재 진행 중에 있다.
- [0018] 3차원 비디오 표준은 일반 영상과 그것의 깊이 정보 영상을 이용하여 스테레오스코픽 영상뿐만 아니라 오토스테레오스코픽 영상의 재생등을 지원할 수 있는 진보된 데이터 형식과 그에 관련된 기술에 대한 표준을 포함하고 있다.
- [0019] 2010년 11월 마이크로소프트는 XBOX-360 게임 디바이스의 새로운 입력장치로 키넥트(Kinect) 센서를 출시하였는데, 이 장치는 사람의 동작을 인지하여 컴퓨터 시스템에 연결하는 장치로 도 2에서 보듯 RGB 카메라뿐만 아니라 3D Depth 센서를 포함하여 이루어져 있다. 또한, 키넥트는 영상 장치로도 RGB 영상 및 최대 640x480 깊이 정보 맵(Depth Map)을 생성해 연결된 컴퓨터에 제공할 수 있다. 이외에도 2014년 인텔은 노트북용으로 320x240 Depth 센서가 장착된 720p CREATIVE SENZ3D 웹캠을 발표하였고, 애플은 RGB 카메라와 Depth 센서를 이용한 iPad 용 3D 스캐너로 iSense를 출시하였고, 구글은 Depth 센서가 장착된 Tango 스마트폰을 발표하였다.
- [0020] 도 3은 웹캠 제품을 나타낸다.
- [0021] 도 3을 참조하면, CREATIVE SENZ3D 웹캠을 나타내고 있으며, 도 3의 (A)는 SENZ3D 웹캠제품을 나타내고, (B)는 SENZ3D 웹캠 프로토타입을 나타낸다.
- [0022] 도 4는 iSense 3D 스캐너 장치를 나타내고, 도 5는 구글 Tango 스마트폰을 나타낸다.
- [0023] 도 4의 (A)는 iSense 제품을 나타내고, (B)는 iSense를 통한 스캐닝 처리의 예를 나타낸다. 도 5의 (A)는 구글 Tango 스마트폰 제품을 나타내고, (B)는 구글 Tango 스마트폰 프로토타입을 나타낸다.
- [0024] 키넥트, iSense 3D 스캐너, 인텔 SENZ3D 웹캠과 같은 영상 장비 및 구글 Tango 스마트폰의 출현은 고가의 2차원 및 3차원 게임이나 영상 서비스와 같은 다양한 응용 어플리케이션을 대중적으로 즐길 수 있게 되는 계기가 되었으며, 깊이 정보 카메라 또는 센서가 부착된 비디오 장치가 대중화가 되고 있음을 보여주고 있다.
- [0025] 이처럼 앞으로의 비디오 시스템이 2차원 일반 영상을 위한 서비스뿐만 아니라 일반 영상 카메라에 Depth 카메라가 결합되어 2차원과 3차원 응용 영상 서비스가 기본적으로 제공되는 형태 또는 핸드헬드(handheld) 시스템에서의 입력 보조장치의 형태로 발전할 것으로 예상된다.
- [0026] 일반 카메라와 Depth 카메라가 기본적으로 결합된 비디오 시스템은 3차원 비디오 코덱에서 깊이 정보를 이용하는 것뿐만 아니라, 2차원 비디오 코덱에서도 깊이 정보를 이용하는 새로운 방법이라 할 수 있겠다.
- [0027] 또한, 깊이 정보 카메라가 포함된 카메라 시스템에서 일반 영상의 부호화는 기존 비디오 코덱을 그대로 사용하여 부호화될 수 있다. 여기서 기존의 비디오 코덱의 일례로, MPEG-1, MPEG-2, MPEG-4, H.261, H.262, H.263, H.264/AVC, MVC, SVC, HEVC, SHVC, 3D-AVC, 3D-HEVC, VC-1, VC-2, VC-3 등으로 부호화 될 수 있으며, 그외 다양한 코덱으로 부호화될 수 있다.
- [0028] 도 6은 HEVC 부호화 장치를 설명하기 위한 도면이다.
- [0029] 실제 영상과 그것의 깊이 정보 맵을 부호화하는 방법의 일례로, 현재까지 개발된 비디오 부호화 표준 중에서 최고의 부호화 효율을 가지는 MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)에서 공동으로 표준화를 완료한 HEVC(High Efficiency Video Coding)를 이용하여 부호화를 수행할 수 있다. HEVC의 부호화 구조도의 일 예는 도 6과 같다. 도 6에서와 같이 HEVC에는 부호화 단위 및 구조, 화면 간(Inter) 예측, 화면 내(Intra) 예측, 보간(Interpolation), 필터링(filtering), 변환(Transform) 방법 등 다양한 새로운 알고리즘들을 포함하고 있다. 도 6은 영상 부호화 장치의 구성에 대한 일 예를 블록도로 도시한 것으로, HEVC 코덱에 따른 부호화 장치의 구성을 나타낸 것이다.
- [0030] 이때, 도 6의 필터부와 참조영상 버퍼 사이에는 SAO(Sample Adaptive Offset)이 구비될 수 있다. SAO는 코딩 에러를 보상하기 위해 화소값에 적정 오프셋(offset) 값을 더해줄 수 있다.
- [0031] HEVC는 인터 예측 부호화, 즉 화면 간 예측 부호화를 수행하므로, 현재 부호화된 영상은 참조 영상으로 사용되기 위해 복호화되어 저장될 필요가 있다. 따라서 양자화된 계수는 역양자화부에서 역양자화되고 역변환부에서

역변환된다. 역양자화, 역변환된 계수는 가산기(175)를 통해 예측 블록과 더해지고 복원 블록이 생성된다. 필터부를 거친 복원 블록은 참조 영상 버퍼에 저장된다.

- [0032] 도 7은 스마트 폰에서 HEVC 인코더가 적용된 영상 부호화의 예를 설명하기 위한 도면이다. 도 8은 스마트 폰에서 깊이 영상이 포함된 HEVC의 예를 설명하기 위한 도면이다.
- [0033] 도 7을 참조하면, HEVC 인코더가 적용된 스마트폰에서의 일반적인 형태는, 스마트폰을 통해 촬영된 영상을 HEVC 인코더를 이용해 부호화 하고, 부호화 된 영상을 가지고 서비스를 제공 받는 형태일 수 있다.
- [0034] 하지만 Depth 카메라가 포함된 스마트폰을 통해 영상을 촬영한다면 도 8과 같이 일반 영상(Texture)과 깊이 영상(Depth)이 독립적으로 생산되고, 깊이 정보를 이용하여 일반 영상과의 상관성을 이용하여 최적화를 통한 HEVC 인코더를 통해서 복잡도 감소를 통해 더욱 향상된 부호화 영상을 얻을 수 있다.
- [0035] 종래기술 특허문헌 1인 미국공개특허 제20140085416호는 깊이 맵(Depth map)으로부터 현재 블록의 객체에 대한 정보를 확인하여 블록을 병합(merge)하는 구성을 개시하고 있으나, 깊이 정보를 이용하여 어떻게 현재 CU의 움직임 벡터를 구할 것인가에 대해서는 전혀 개시하지 못하고 있다.
- [0036] 또한, 종래기술 특허문헌 2인 한국공개특허 제2012-0137305호 및 특허문헌 3인 한국공개특허 제2014-0048784호는 깊이 정보를 이용하는 내용을 개시하지 못하거나 CU의 분할 구조를 예측할 때 움직임 벡터를 구하기 위한 연산량을 줄이기 위한 구성을 명확하게 제시하지 못하고 있다.
- [0037] 도 9는 영상의 복수의 유닛들로 분할하는 예들을 나타낸다.
- [0038] 고효율 비디오 코딩 방식은 부호화를 수행할 때 영상을 코딩 유닛(CU: Coding Unit, 이하 'CU')의 기본 단위인 LCU(LCU: Largest Coding Unit) 단위로 나누어 부호화를 수행한다.
- [0039] 여기서, CU는 기존의 비디오 코덱인 H.264/AVC에서의 기본 블록인 매크로블록(MB: Macro Block, 이하 'MB')과 유사한 역할을 하지만, 16x16의 고정 크기를 갖는 MB와 달리 CU는 가변적으로 크기를 정할 수 있다. 또한 부호화를 위해 분할된 LCU는 영상의 효율적인 부호화를 위해 다시 LCU 보다 작은 크기를 갖는 여러 CU로 분할될 수 있다.
- [0040] 도 9를 참조하면, 64x64 크기의 LCU는 다양한 방식으로 복수의 CU들로 분할될 수 있다.
- [0041] 도 9의 (A)는 분할 깊이 값이 0인 64x64 크기의 LCU를 분할 깊이 1인 32x32 크기의 CU들로 분할한 예를 나타낸다.
- [0042] 도 9의 (B)는 32x32 크기의 CU들 중 하나를 분할 깊이 2로 분할한 예를 나타내고, (C)는 32x32 크기의 CU들 중 두개를 분할 깊이 2로 분할한 예를 나타낸다.
- [0043] 도 9의 (D)는 분할깊이 3으로 분할된 CU들을 포함하는 예를 나타낸다.
- [0044] 따라서, LCU 또는 CU의 분할구조 후보들은 다양한 방식으로 존재할 수 있다.
- [0045] LCU 분할 구조는 부호화 단위의 분할 정보이다. 위와 같이 다양한 LCU 분할 구조를 생성하여 LCU 분할 구조 후보에 저장한 뒤, 최적의 LCU 분할 구조를 결정하는 단계에서 LCU 단위로 LCU 분할 구조 후보 중 하나의 분할 구조를 최적의 LCU 분할 구조로 선택하게 된다. 이러한 방법을 사용함으로써 LCU 단위로 영상의 특성에 맞게 적응적인 LCU 분할 구조를 기초로 부호화를 수행함으로써 부호화 효율 및 화질 측면에서 효율적인 부호화를 수행할 수 있다는 장점이 있다.
- [0046] 종래기술에 따른 2차원 비디오 코덱은, 깊이 정보 이용을 전혀 반영하지 않고 알고리즘들이 설계되어 있다. 하지만, 실제 영상과 그것의 깊이 정보 영상은 큰 상관성을 가지므로, 깊이 정보를 2차원 영상을 부호화하는 데 활용할 수 있다는 것에 착안하여 깊이 정보를 고려한 알고리즘 개발로 2차원 비디오 부호화에서의 깊이 정보 이용방법을 고려할 수 있다.
- [0047] 본 발명의 기본적인 원리는 2차원 비디오 코덱에서 효율적인 부호화를 위해서, 깊이 정보 카메라에서 획득한 깊이 정보를 이용하여 실제 영상을 부호화하는 데 활용하기위해 움직임 예측 방법에서의 깊이 정보를 이용하는 것이다.
- [0048] 예를 들어, 깊이 정보 영상을 활용하여 일반 영상의 객체들을 구분하여 부호화할 경우, 일반 영상에 대한 부호화 복잡도를 크게 감소시킬 수 있다.

- [0049] 여기서 객체들이란, 여러 개의 객체를 의미하여 배경 영상을 포함할 수 있으며, 블록기반 부호화 코덱에서 블록 내에는 여러 개의 객체가 존재할 수 있으며, 깊이 정보 영상으로 기반으로 해당 객체마다 각각 다른 부호화 방법들이 적용될 수 있다.
- [0050] 도 10은 종래 기술에 따른 참조 블록으로부터 현재 블록의 움직임 정보를 구하는 방법의 예를 나타낸다.
- [0051] 도 10을 참조하면, 움직임 정보인 MV(Motion Vector; 움직임 벡터)는 도 10의 (A)에 도시한 바와 같이 현재 예측하려는 "Current block"과 참조 영상에서 제일 근접한 블록과의 거리를 구해 획득될 수 있다.
- [0052] 이때, 도 10의 (B)를 참조하면, 움직임 정보인 MVP(Motion Vector Predictor)는 현재 블록의 주변 블록들(A, B, C, D)을 통해 결정될 수 있고, MVD(Motion Vector Difference)는 MVP를 통해 구해 질 수 있다.
- [0053] 이때, 구해진 MVD는 비디오 부호화 과정에 이용될 수 있다.
- [0054] 도 11은 종래 기술에 따른 움직임 벡터를 구하는 방법의 예를 나타낸다.
- [0055] 도 11은 HEVC 부호화 과정에서 현재 코딩 유닛(CU: Coding Unit, 이하 CU)에 대한 화면 간 예측의 움직임 정보의 부호화에서 MV를 구하는 방법을 예시한다.
- [0056] 도 11의 (A)는 HEVC에서 MV를 구하는 흐름도이고, (B)는 움직임 탐색 방법을 예시한다.
- [0057] 도 11을 참조하면, 참조 프레임으로부터 움직임 탐색 영역 안의 위치들에 다이아몬드 탐색 방법을 이용하여 1차로 현재 CU와 제일 근접한 블록이 존재하는 위치를 찾고 2차로 2포인트 탐색 방법을 이용하여 해당 위치를 중심으로 세밀하게 주변 위치들과 비교하여 최적의 MV를 구할 수 있다.
- [0058] 도 11의 (B)에서, 현재 CU의 탐색 시작 위치를 1번으로 하여 다이아몬드 탐색 방법에 의해 4번의 최적의 위치를 찾고, 해당 지점으로 2포인트 탐색 방법을 이용하여 5번 위치를 이용하여 최적의 MV를 구할 수 있다.
- [0059] HEVC에서 움직임 탐색 방법의 경우 움직임 탐색 영역 안의 위치에 대해서 일률적으로 탐색 알고리즘에 방법에 의해 해당 위치로부터의 MV를 모두 구해 그 중에 최적의 MV를 선택한다. 이러한 방법은 MV를 구하는 블록과 탐색 영역 안의 위치들과의 상관성을 모르기 때문에 모든 위치에 대해 MV를 구하는 방법을 이용하는 것이다. 또한 MV를 구하는 CU와 탐색 영역에서의 상관성이 없는 위치들에 대해서 불필요한 계산을 증가시켜 부호화 복잡도를 증가시키는 원인이 된다.
- [0060] 상기의 MV를 구하는 이러한 과정은 주변 블록에 대해 객체 정보를 고려하지 않고 있다. 이는 당연한데, 2차원 영상의 경우, 깊이 카메라가 없다면 2차원 영상 분석을 통하여 영상내의 객체정보를 추출해내야 하므로, 현존하는 2차원 비디오 부호화 방법에는 객체정보를 이용하는 방법이 전혀 탑재되어 있지 않기 때문이다.
- [0061] 같은 이유로, HEVC 경우, 화면 내 움직임 예측에서의 MV를 구하는 방법에서도 객체정보를 이용하는 부호화 방법이 전혀 탑재되어 있지 않다. 하지만 Depth 카메라를 이용하여 객체정보를 고려할 수 있다면, 움직임 예측에 있어서 해당 CU와 탐색 영역에서의 상관성을 알 수 있어 탐색 하는 영역을 설정할 수 있다면, 부호화 복잡도를 효율적으로 감소시킨다.
- [0062] 따라서, 깊이 정보를 움직임 탐색에 이용하는 경우 해당 CU와 동일한 객체 영역을 판단하여 해당 영역에 대해서 움직임 탐색 범위를 제한하는 경우 부호화 복잡도를 효율적으로 감소할 수 있다.
- [0063] 도 12는 깊이 영상의 예들을 나타낸다.
- [0064] 도 12의 (A)는 예를 들어 깊이 카메라로부터 획득된 84번째 영상 프레임의 깊이 영상이고, (B)는 예를 들어 깊이 카메라로부터 획득된 83번째 영상 프레임의 깊이 영상일 수 있다.
- [0065] 도 12의 (A)를 현재 영상이라 하면, 현재 CU가 단일 객체를 가질 때, 도 12의 (B)와 같은 참조 프레임으로부터 움직임 탐색을 하는 데 있어서 깊이 정보를 이용하여 움직임 탐색 영역에 있는 동일 객체 영역을 판단할 수 있다.
- [0066] 현재 CU와 동일 객체 영역의 경우에는 최적의 MV를 가질 확률이 높고, 그렇지 않은 경우에는 최적의 MV를 가지지 않을 확률이 높다. 따라서 현재 CU의 동일 객체 영역이 판단되면, 움직임 탐색 영역 중 현재 CU와 동일 객체 영역이 아닌 영역에 대해서는 움직임 탐색을 수행하지 않음으로써 움직임 탐색에 필요한 연산량을 줄일 수 있다.
- [0067] 본 발명에서 제안되는 방법은 블록에 대한 움직임 예측에서 탐색 범위 결정에서 깊이 정보를 이용하여 동일 객

체 영역을 판단하여 움직임 정보를 부호화하는 것이다.

- [0068] 도 13은 본 발명의 일 실시예에 따른 움직임 벡터 결정 방법을 설명하기 위한 흐름도이다.
- [0069] 도 13을 참조하면 1310단계에서 영상 부호화 장치는 현재 코딩 유닛(CU: Coding Unit)의 객체 정보에 기초하여, 상기 현재 CU의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역에서 움직임 탐색 범위를 결정한다.
- [0070] 예를 들어, 현재 CU의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역은 도 12의 (B)에 도시된 "탐색 영역"일 수 있다.
- [0071] 이때, 움직임 탐색 범위는 현재 CU를 구성하는 객체와 동일 객체 영역일 수 있다. 따라서, 예를 들어, 움직임 탐색 범위는 도 12의 (B)에서 "탐색 X" 부분을 제외한 "탐색 0" 부분일 수 있다.
- [0072] 움직임 탐색 범위를 결정하는 1310단계는, 깊이 영상으로부터 현재 코딩 유닛(CU: Coding Unit)의 깊이 정보를 확인하는 과정을 포함할 수 있다.
- [0073] 또한, 움직임 탐색 범위를 결정하는 1310단계는, 깊이 정보에 기초하여 상기 현재 CU의 객체 정보를 결정하고, CU의 객체 정보에 기초하여 상기 현재 CU가 단일객체로 이루어지는 지를 판단하고, 현재 CU가 단일객체로 이루어지는 경우 상기 전체 움직임 탐색 영역 중 상기 CU의 단일 객체와 동일하지 않은 영역을 상기 움직임 탐색 범위에서 제외하는 과정을 포함할 수 있다.
- [0074] 이때, 객체 정보는 깊이 영상을 라벨링 알고리즘에 의해 라벨링한 객체 정보 영상의 라벨 값 또는 상기 깊이 영상으로부터 획득되는 깊이 값 분포 정보일 수 있다. 예를 들어, 라벨링 알고리즘은 Watersherd 라벨링 알고리즘일 수 있다.
- [0075] 이때, 깊이 값 분포는 CU는 예를 들어, 도 18에 도시한 바와 같이 획득될 수 있다.
- [0076] 예를 들어, 현재 CU의 깊이 값 분포 정보에 포함된 최대값 및 최소값의 차가 기 설정된 값 미만이면, 상기 현재 CU가 단일 객체로 이루어지는 것으로 판단될 수 있다.
- [0077] 움직임 탐색 범위를 결정하는 1310단계는, 현재 CU의 크기 및 상기 현재 CU의 깊이 값 분포 정보에 포함된 최대값 및 최소값에 기초하여 상기 현재 CU가 단일 객체로 이루어지는 지를 판단하는 과정을 포함할 수 있다. 예를 들어, 1310단계는 도 16의 1610단계 및 1620단계를 포함할 수 있다.
- [0078] 1320단계에서 영상 부호화 장치는 움직임 탐색 범위 내의 움직임 정보에 기초하여 상기 현재 CU의 최적 움직임 벡터를 결정한다.
- [0079] 이때, 현재 CU의 최적 움직임 벡터를 결정하는 방법은 도 12의 (B)에서 "탐색 X" 부분을 제외한 "탐색 0" 부분에 대해 도 10 내지 도 11에서 설명한 방법일 수 있다.
- [0080] 도 14는 본 발명의 다른 일 실시예에 따른 움직임 벡터 결정 방법을 설명하기 위한 흐름도이다.
- [0081] 도 14를 참조하면, 1410단계에서 영상 부호화 장치는 현재 코딩 유닛(CU: Coding Unit)의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역을 확인한다.
- [0082] 1420단계에서 깊이 영상으로부터 추출되는 상기 현재 CU의 깊이 정보에 따라 현재 움직임 탐색 위치에서의 움직임 탐색 여부를 결정한다.
- [0083] 이때, 움직임 탐색 여부를 결정하는 1420단계는 현재 CU의 크기 및 상기 현재 CU의 깊이 값 분포 정보에 포함된 최대값 및 최소값에 기초하여 상기 현재 CU가 단일 객체로 이루어지는 지를 판단하는 과정을 포함할 수 있다.
- [0084] 또한, 움직임 탐색 여부를 결정하는 1420단계에서 영상 부호화 장치는, 현재 CU의 크기가 기설정된 값 이상이고 상기 현재 CU의 네 모서리 깊이 값들 중 최대값 및 최소값의 차가 기 설정된 기준값 이하이면, 상기 현재 CU의 네 모서리 깊이 값들 중 어느 하나를 저장한 후, 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 같으면 현재 위치를 탐색하는 것으로 결정하고, 상기 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 다르면 현재 위치를 탐색하지 않는 것으로 결정할 수 있다.
- [0085] 또한, 움직임 탐색 여부를 결정하는 1420단계에서 영상 부호화 장치는, 현재 CU의 크기가 기설정된 값 보다 작고 상기 현재 CU의 네 모서리 깊이 값들이 모두 같은 값이면, 상기 현재 CU의 네 모서리 깊이 값들 중 어느 하나를 저장한 후, 상기 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 같으면 현재 위치를 탐색하는 것으로 결정하고, 상기 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 다르면 현재 위치를 탐색하지 않는 것

으로 결정할 수 있다.

- [0086] 1430단계에서 결정된 움직임 탐색 여부를 고려하여 상기 현재 CU의 최적 움직임 벡터를 결정한다.
- [0087] 도 15는 본 발명의 일 실시예에 따른 움직임 탐색 방법의 예를 설명하기 위한 도면이다.
- [0088] 1510단계에서 영상 부호화 장치는 움직임 탐색 영역으로부터 탐색을 시작하고, 1520단계에서 현재 CU가 단일 객체 영역으로 구성되었는지를 판단하여, 단일 구성 객체 영역으로 구성된 경우 1530단계를 수행하고, 그렇지 않은 경우 1550단계에서 종래 기술에 따른 기존의 움직임 탐색 방법을 적용하여 움직임 벡터를 계산할 수 있다.
- [0089] 1530단계에서 영상 부호화 장치는 참조 프레임 위치에서의 객체와 현재 CU의 객체가 동일 객체 영역일 경우 또는 동일 객체로 판단되는 경우에는 1550단계에서 해당 위치에서 움직임 벡터를 계산하고, 그렇지 않은 경우에는 1540단계에서 현재 위치에서 움직임 벡터를 구하지 않는다는 정보를 저장하고 움직임 탐색 과정을 종료할 수 있다.
- [0090] 도 16은 본 발명의 일 실시예에 따른 깊이 정보를 이용한 움직임 탐색 방법의 예를 설명하기 위한 도면이다.
- [0091] 도 16에 도시된 방법은 동일 객체 여부 판단 및 현재 위치에서의 움직임 탐색 여부를 결정하는 예를 나타낸다.
- [0092] 1610단계에서 영상 부호화 장치는, 현재 CU의 크기가 64x64 또는 32x32인지 확인하고 32x32 이상이면 1620단계를 수행하고, 그렇지 않은 경우 1640단계를 수행한다.
- [0093] 1620단계에서 깊이 값 분포 정보로부터 현재 CU의 네 모서리의 깊이 값 중 최대값과 최소값의 차이가 5 미만인지를 판단한다. 이때, 최대값과 최소값의 차이가 5 미만이면 1630단계를 수행하고, 그렇지 않은 경우 1680단계를 수행한다.
- [0094] 1630단계에서 영상부호화 장치는 현재 CU의 네 모서리 중 좌상단 깊이 값을 저장하고 1670단계를 수행한다.
- [0095] 1640단계에서 영상 부호화 장치는 현재 CU의 크기가 16x16인지 확인하고 16x16이면 1650단계를 수행하고, 그렇지 않은 경우 1680단계를 수행한다.
- [0096] 1650단계에서 현재 CU의 네 모서리의 깊이 값이 모두 같은지를 확인하고 모두 같은 경우 1660단계를 수행하고, 그렇지 않은 경우 1680단계를 수행한다.
- [0097] 1660단계에서 영상부호화 장치는 현재 CU의 네 모서리 중 어느 한 값을 저장하고 1670단계를 수행한다.
- [0098] 1670단계에서 영상 부호화 장치는 현재 움직임 탐색 위치의 깊이 값이 이전(1630단계 또는 1660단계)에 저장된 값과 같은지를 확인한다.
- [0099] 이때, 현재 움직임 탐색 위치, 즉 현재 움직임 탐색 포인트에서의 깊이 값이 이전에 저장된 값과 같은 경우 1680 단계에서 현재 위치의 움직임 정보를 탐색하고, 그렇지 않은 경우 1690단계에서 현재 위치의 움직임 정보를 탐색하지 않는 것으로 결정하고 움직임 탐색 과정을 종료할 수 있다.
- [0100] 도 17은 본 발명의 일 실시예에 따른 영상 부호화 장치의 구성을 설명하기 위한 도면이다.
- [0101] 도 17을 참조하면, 영상 부호화 장치(1700)는 도 13 내지 도 16의 방법을 수행할 수 있다.
- [0102] 영상 부호화 장치(1700)는 움직임 탐색 범위 결정부(1710), 움직임 탐색 여부 결정부(1720) 및 움직임 벡터 결정부(1730)를 포함할 수 있다.
- [0103] 예를 들어, 영상 부호화 장치(1700)는 움직임 탐색 범위 결정부(1710) 및 움직임 탐색 여부 결정부(1720)로 구성될 수 있다. 또한, 영상 부호화 장치(1700)는 움직임 탐색 여부 결정부(1720) 및 움직임 벡터 결정부(1730)로 구성될 수도 있다.
- [0104] 움직임 탐색 범위 결정부(1710)는 현재 코딩 유닛(CU: Coding Unit)의 객체 정보에 기초하여, 상기 현재 CU의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역에서 움직임 탐색 범위를 결정한다.
- [0105] 이때, 움직임 탐색 범위는 상기 현재 CU를 구성하는 객체와 동일 객체 영역일 수 있다.
- [0106] 움직임 탐색 범위 결정부(1710)는 현재 CU가 단일객체로 이루어지는 경우, 상기 전체 움직임 탐색 영역 중 상기 CU의 단일 객체와 동일하지 않은 영역을 상기 움직임 탐색 범위에서 제외할 수 있다.
- [0107] 움직임 탐색 범위 결정부(1710)는 현재 CU의 크기 및 상기 현재 CU의 깊이 값 분포 정보에 포함된 최대값 및 최

소값에 기초하여 상기 현재 CU가 단일 객체로 이루어지는 지를 판단할 수 있다.

- [0108] 움직임 탐색 여부 결정부(1720)는 현재 코딩 유닛(CU: Coding Unit)의 움직임 벡터 계산을 위한 전체 움직임 탐색 영역을 확인하고, 깊이 영상으로부터 추출되는 상기 현재 CU의 깊이 정보에 따라 현재 움직임 탐색 위치에서의 움직임 탐색 여부를 결정한다.
- [0109] 움직임 탐색 여부 결정부(1720)는 현재 CU의 크기가 기설정된 값 이상이고 상기 현재 CU의 네 모서리 깊이 값들 중 최대값 및 최소값의 차가 기 설정된 기준값 이하이면, 상기 현재 CU의 네 모서리 깊이 값들 중 어느 하나를 저장한다.
- [0110] 또한, 움직임 탐색 여부 결정부(1720)는 현재 CU의 크기가 기설정된 값 보다 작고 상기 현재 CU의 네 모서리 깊이 값들이 모두 같은 값이면, 상기 현재 CU의 네 모서리 깊이 값들 중 어느 하나를 저장한다.
- [0111] 움직임 탐색 여부 결정부(1720)는 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 같으면 현재 위치를 탐색하는 것으로 결정하고, 상기 현재 움직임 탐색 위치의 깊이 값이 상기 저장된 값과 다르면 현재 위치를 탐색하지 않는 것으로 결정할 수 있다.
- [0112] 움직임 벡터 결정부(1730)는 움직임 탐색 범위 내의 움직임 정보에 기초하여 상기 현재 CU의 최적 움직임 벡터를 결정한다.
- [0113] 도 18은 CU의 깊이 값 분포 예들을 나타내는 도면이다.
- [0114] 일 실시예에 따른 영상 부호화 방법에 있어서, CU 또는 블록이 동일한 단일 객체로 이루어지는지를 판단하는 방법의 예로써, CU 또는 블록의 네 모서리 위치의 깊이 값을 이용할 수 있다.
- [0115] 도 18을 참조하면, 영상 부호화 장치는 (A)와 같은 CU 깊이 값 분포를 깊이 값의 변화가 크지 않고 고른 것으로 판단할 수 있다.
- [0116] 반면, 영상 부호화 장치는 도 18의 (B)와 같은 CU 깊이 값 분포를 깊이 값의 변화가 크기 때문에 단일 객체로 이루어지지 않는 것으로 판단할 수 있다.
- [0117] 도 18의 (B)를 참조하면, CU 내에 중간 부분과 모서리 부분이 깊이 값의 변화가 매우 크다는 것을 볼 수 있는데, 이러한 경우 네 모서리의 깊이 값 중 최대값과 최소값의 차이가 크다는 것을 알 수 있다. 따라서 이러한 CU는 단일 객체로 이루어져 있을 확률이 낮기 때문에 CU 분할을 수행할 수 있다.
- [0118] 영상 부호화 장치는 CU의 네 모서리 깊이 값들 중 최대값 및 최소값의 차가 기 설정된 기준값 이하이면 상기 CU가 단일 객체로 이루어지는 것으로 판단할 수 있다.
- [0119] 이때, 예를 들어 도 18의 (A)에서 M1의 정규화된 깊이 값은 9이고, M2 및 M3의 정규화된 깊이 값은 7이고, M4의 정규화된 깊이 값은 7일 수 있다.
- [0120] 또한, 도 18의 (B)에서 M1의 정규화된 깊이 값은 9이고, M2 및 M4의 정규화된 깊이 값은 7이고, M3의 정규화된 깊이 값은 1일 수 있다.
- [0121] 이때, 도 18의 (A)의 경우 CU 네 모서리의 깊이 값의 최대값 및 최소값의 차가 2이고, 도 18의 (B)의 경우 CU 네 모서리의 깊이 값의 최대값 및 최소값의 차가 8이다.
- [0122] 따라서, 기 설정된 기준값이 5인 경우, 도 18의 (A)는 단일 객체로 이루어지는 것으로 결정되고, 도 18의 (B)는 단일 객체로 이루어지지 않는 것으로 결정될 수 있다.
- [0123] [표 1]은 도 16에 도시된 실시예를 HEVC에 적용한 실험 결과를 나타낸다.
- [0124] 실험 결과를 통하여 주관적 화질에서의 동일한 품질로 영상 품질의 큰 열화 없이 부호화 복잡도가 감소됨을 확인할 수 있다.

		Low-delay Main				Random-access Main			
		Y BD-rate	U BD-rate	V BD-rate	Encoding Time	Y BD-rate	U BD-rate	V BD-rate	Encoding Time
3DV Test Sequences	Balloons	0.0%	0.2%	0.2%	94.5%	0.0%	0.0%	0.1%	93.2%
	Kendo	0.0%	0.3%	0.9%	89.6%	0.1%	0.5%	0.3%	93.1%
	Newspaper_CC	-0.1%	-0.2%	0.2%	94.9%	0.0%	-0.1%	-0.1%	96.3%
	Poznan_Hall2	0.1%	1.5%	0.8%	94.8%	0.0%	-0.1%	0.1%	96.3%
	Poznan_Street	0.0%	-0.5%	0.7%	98.8%	0.3%	0.6%	0.6%	101.2%
	GT_Fly	0.2%	0.2%	0.4%	91.7%	0.9%	1.5%	1.8%	96.8%
	Undo_Dancer	0.0%	0.0%	0.5%	96.0%	0.1%	0.4%	0.6%	97.9%
	Shark	0.0%	0.0%	0.1%	92.3%	0.5%	0.8%	0.8%	98.1%
1024x768 Average		0.0%	0.1%	0.4%	93.0%	0.0%	0.2%	0.1%	94.2%
1920x1088 Average		0.1%	0.3%	0.5%	94.7%	0.4%	0.6%	0.8%	98.1%
Average		0.0%	0.2%	0.5%	94.1%	0.3%	0.5%	0.5%	96.6%

본 발명의 실시예들에 있어서, 대상 범위 또는 적용 범위는 블록 크기 혹은 CU의 분할 깊이 등에 따라 가변될 수 있다.

이때, 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 분할 깊이에 따라 적용 범위를 달리하는 할 때는 [표 2]에 예시한 바와 같다. 방식A는 미리 설정된 깊이값 이상의 깊이에만 적용하는 방식일 수 있으며, 방식B는 미리 설정된 깊이값 이하에만 적용하는 방식일 수 있고, 방식C는 미리 설정된 깊이값에만 적용하는 방식이 있을 수 있다.

[표 2]

적용범위를 나타내는 CU 깊이	방법 A	방법 B	방법 C
0	X	0	X
1	X	0	X
2	0	0	0
3	0	X	X
4	0	X	X

[표 2]는 주어진 CU 분할 깊이가 2인 경우, 본 발명의 방법들을 적용하는 적용 범위 결정 방식의 예를 나타낸다. (0: 해당 깊이에 적용, X: 해당 깊이에 적용하지 않음.)

모든 깊이에 대하여 본 발명의 방법들을 적용하지 않는 경우는 임의의 지시자(flag)를 사용하여 비트스트림에 나타낼 수도 있고, CU 깊이의 최대값보다 하나 더 큰 값을 적용범위를 나타내는 CU 깊이 값으로 시그널링 함으로써 표현할 수도 있다.

또한 상술한 각 방법들은 휘도 블록의 크기에 따라 색차 블록에 다르게 적용될 수 있으며, 휘도 신호영상 및 색차 영상에도 다르게 적용될 수 있다.

[표 3]은 각 방법을 조합하는 경우, 휘도 블록의 크기 및 색차 블록에 따라 다르게 적용되는 예를 나타낸다.

[0134]

[표 3]

휘도블록크기	색차블록크기	휘도적용	색차적용	방법들
4(4x4, 4x2, 2x4)	2(2x2)	0 or X	0 or X	가 1, 2, ..
	4(4x4, 4x2, 2x4)	0 or X	0 or X	나 1, 2, ..
	8(8x8, 8x4, 4x8, 2x8 등)	0 or X	0 or X	다 1, 2, ..
	16(16x16, 16x8, 4x16, 2x16 등)	0 or X	0 or X	라 1, 2, ..
	32(32x32)	0 or X	0 or X	마 1, 2, ..
8(8x8, 8x4, 2x8 등)	2(2x2)	0 or X	0 or X	바 1, 2, ..
	4(4x4, 4x2, 2x4)	0 or X	0 or X	사 1, 2, ..
	8(8x8, 8x4, 4x8, 2x8 등)	0 or X	0 or X	아 1, 2, ..
	16(16x16, 16x8, 4x16, 2x16 등)	0 or X	0 or X	자 1, 2, ..
	32(32x32)	0 or X	0 or X	카 1, 2, ..
16(16x16, 8x16, 4x16 등)	2(2x2)	0 or X	0 or X	타 1, 2, ..
	4(4x4, 4x2, 2x4)	0 or X	0 or X	파 1, 2, ..
	8(8x8, 8x4, 4x8, 2x8 등)	0 or X	0 or X	하 1, 2, ..
	16(16x16, 16x8, 4x16, 2x16 등)	0 or X	0 or X	개 1, 2, ..
	32(32x32)	0 or X	0 or X	내 1, 2, ..

[0135]

[0136]

[표 3]의 변형된 방법들 중에서 방법 사 1 을 살펴보면, 휘도블록의 크기가 8(8x8, 8x4, 2x8 등)인 경우이고, 그리고 색차 블록의 크기가 4(4x4, 4x2, 2x4)인 경우에 본 발명의 실시 예에 따른 머지 리스트 구성 방법을 휘도 신호 및 색차 신호에 적용할 수 있다.

[0137]

위의 변형된 방법들 중에서 방법 파 2 을 살펴보면, 휘도블록의 크기가 16(16x16, 8x16, 4x16 등)인 경우이고, 그리고 색차 블록의 크기가 4(4x4, 4x2, 2x4)인 경우에 본 발명의 실시 예에 따른 머지 리스트 구성 방법을 휘도 신호에 적용하고 색차 신호에는 적용하지 않을 수 있다.

[0138]

또 다른 변형된 방법들로 휘도 신호에만 본 발명의 실시 예에 따른 머지 리스트 구성 방법이 적용되고 색차 신호에는 적용되지 않을 수 있다. 반대로 색차 신호에만 본 발명의 실시 예에 따른 머지 리스트 구성 방법이 적용되고 휘도 신호에는 적용되지 않을 수 있다.

[0139]

이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설

명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소 (processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서 (parallel processor)와 같은, 다른 처리 구성 (processing configuration)도 가능하다.

[0140]

[0141]

소프트웨어는 컴퓨터 프로그램 (computer program), 코드 (code), 명령 (instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로 (collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소 (component), 물리적 장치, 가상 장치 (virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파 (signal wave)에 영구적으로, 또는 일시적으로 구체화 (embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0142]

[0143]

실시에에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체 (magnetic media), CD-ROM, DVD와 같은 광기록 매체 (optical media), 플롭티컬 디스크 (floptical disk)와 같은 자기-광 매체 (magneto-optical media), 및 롬 (ROM), 램 (RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0144]

[0145]

이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

[0146]

[0147]

그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

도면1



도면2



도면3



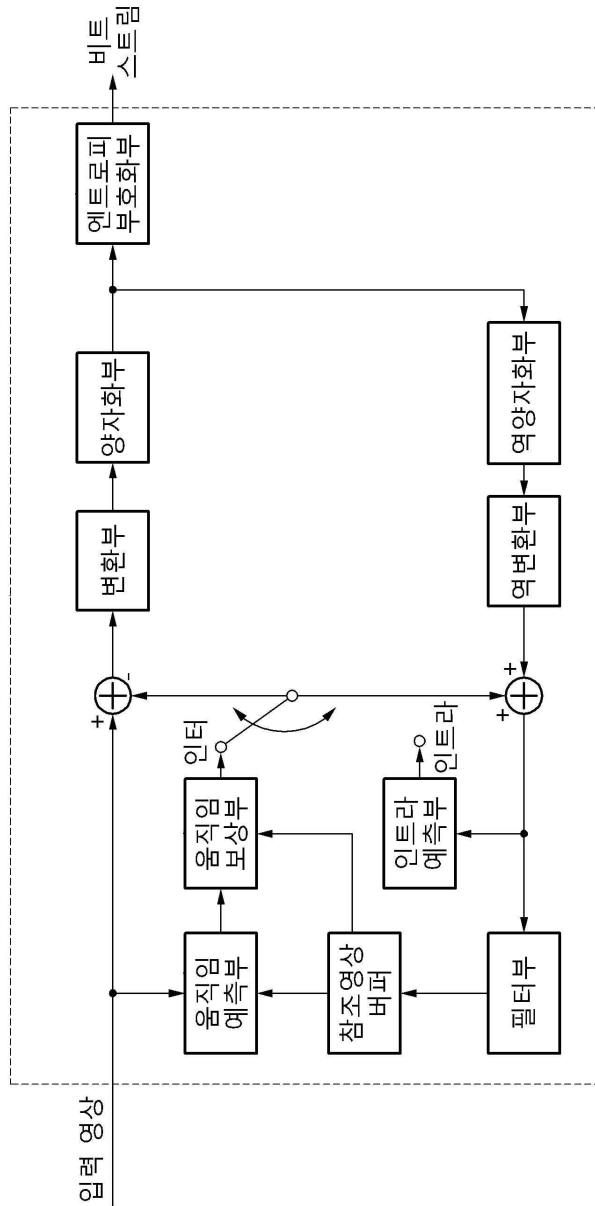
도면4



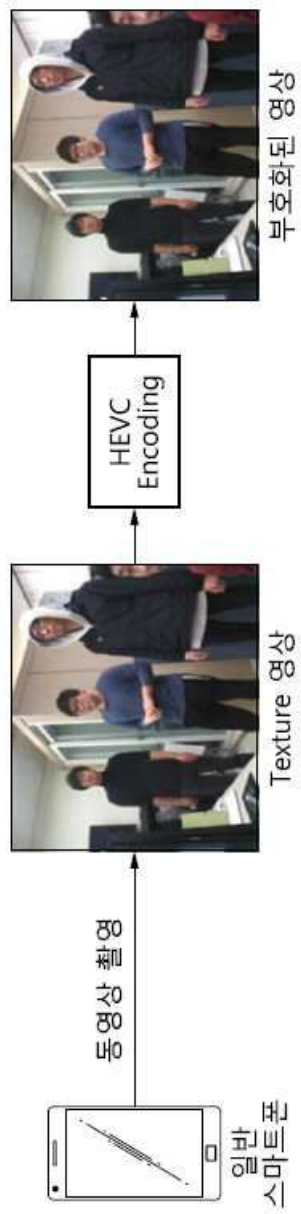
도면5



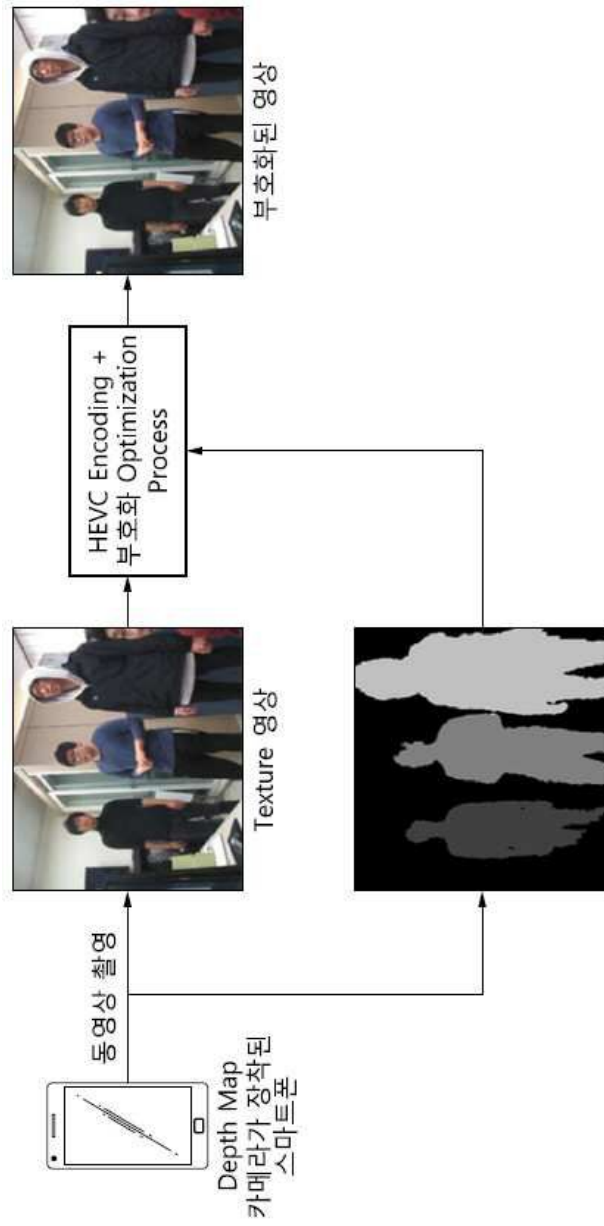
도면6



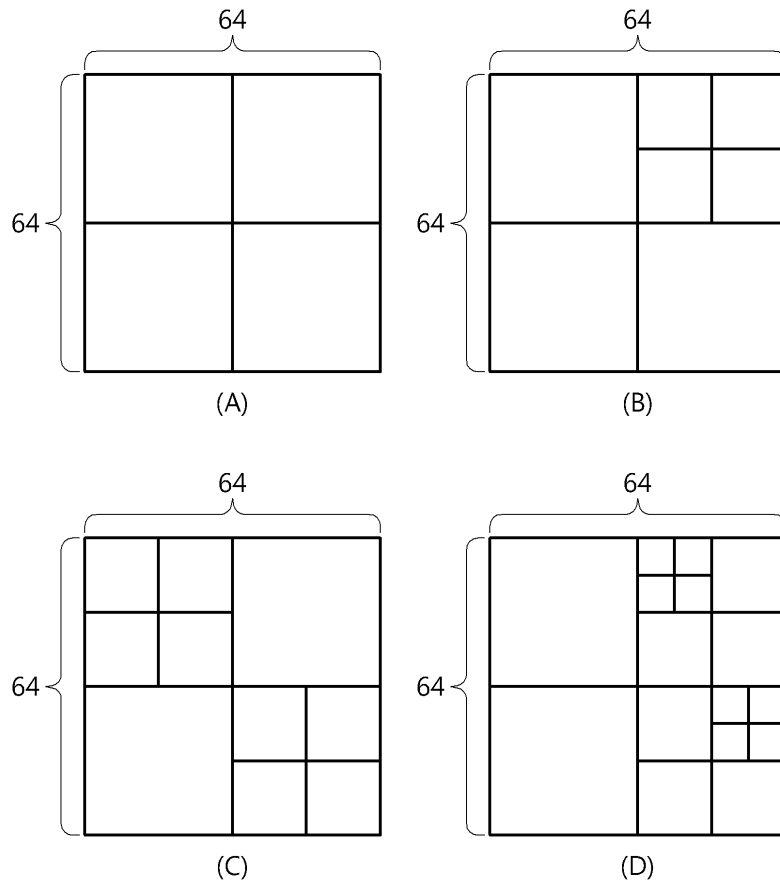
도면7



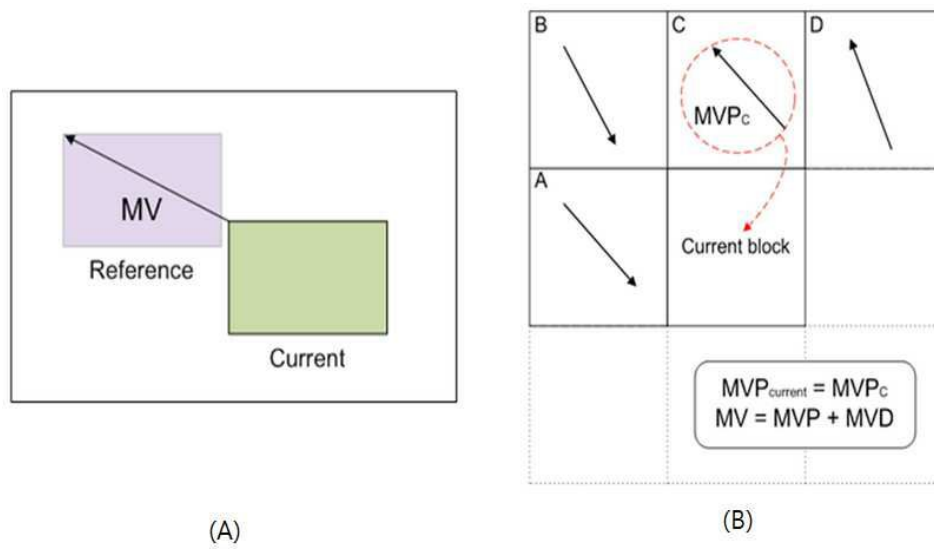
도면8



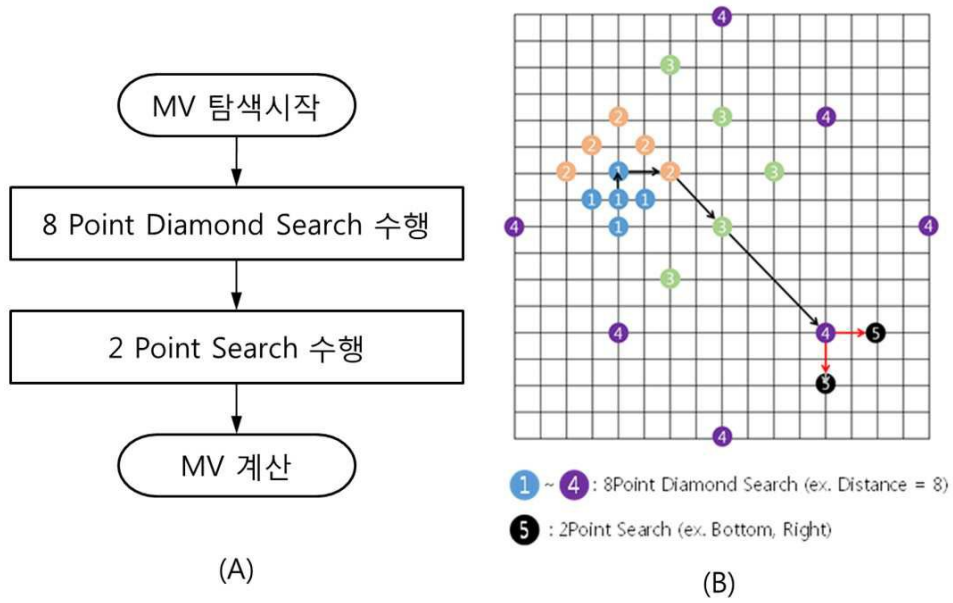
도면9



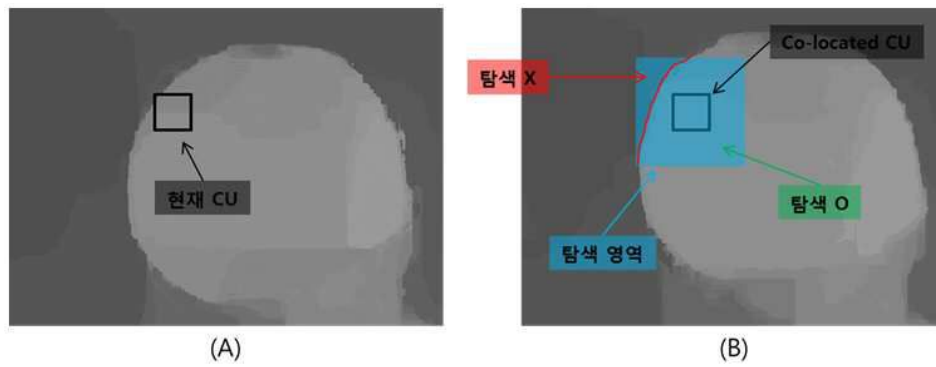
도면10



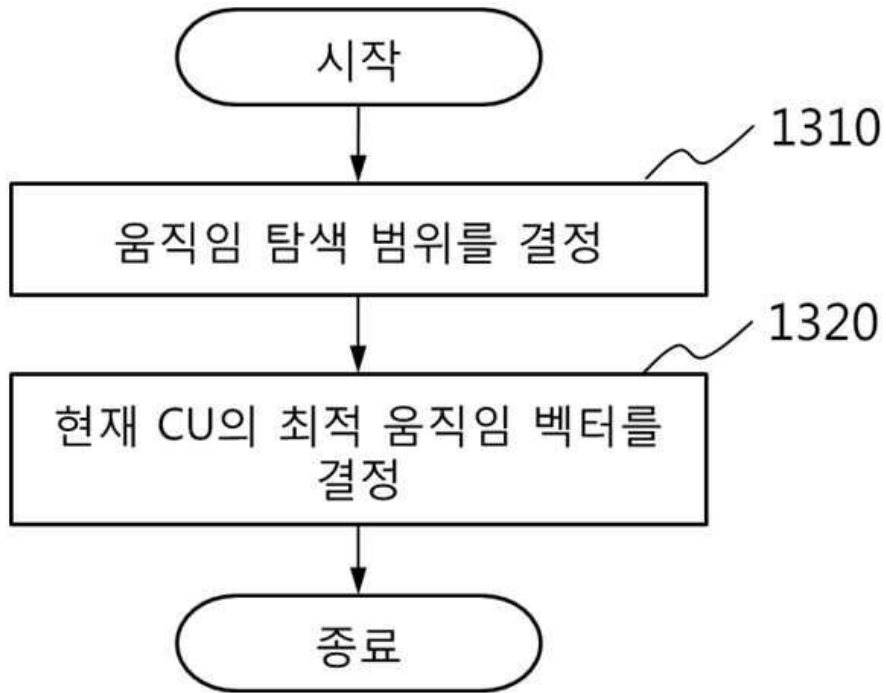
도면11



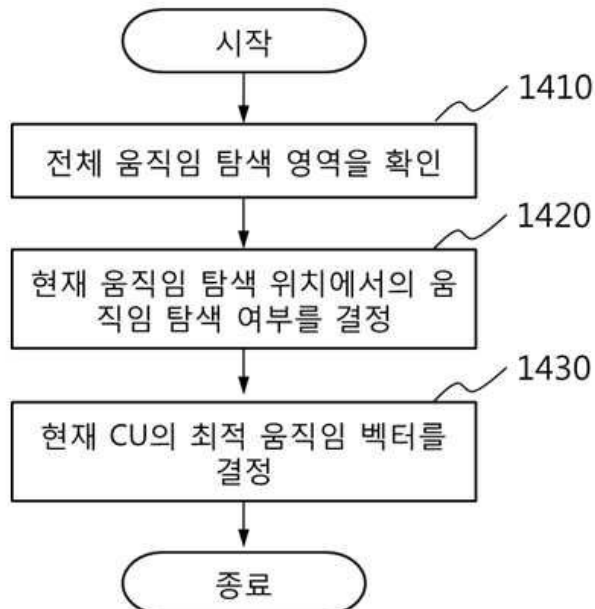
도면12



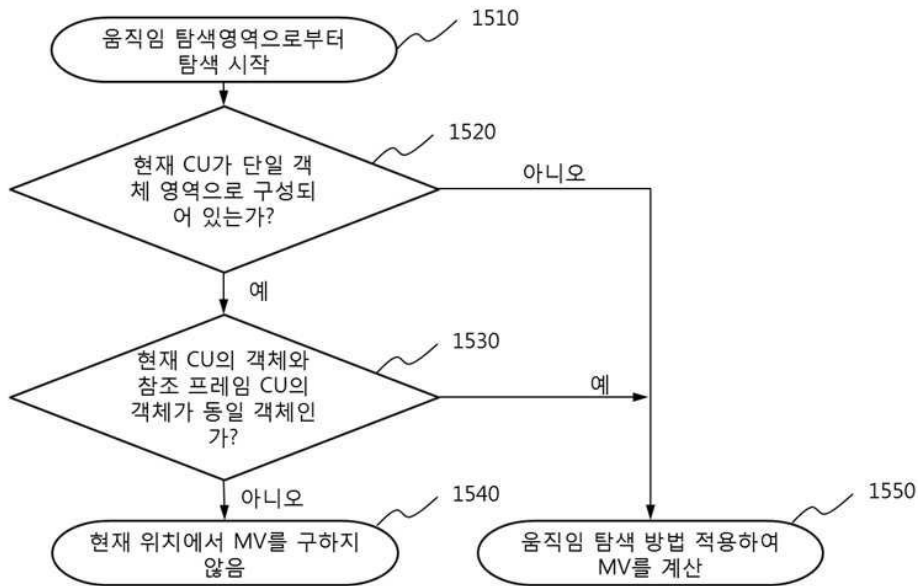
도면13



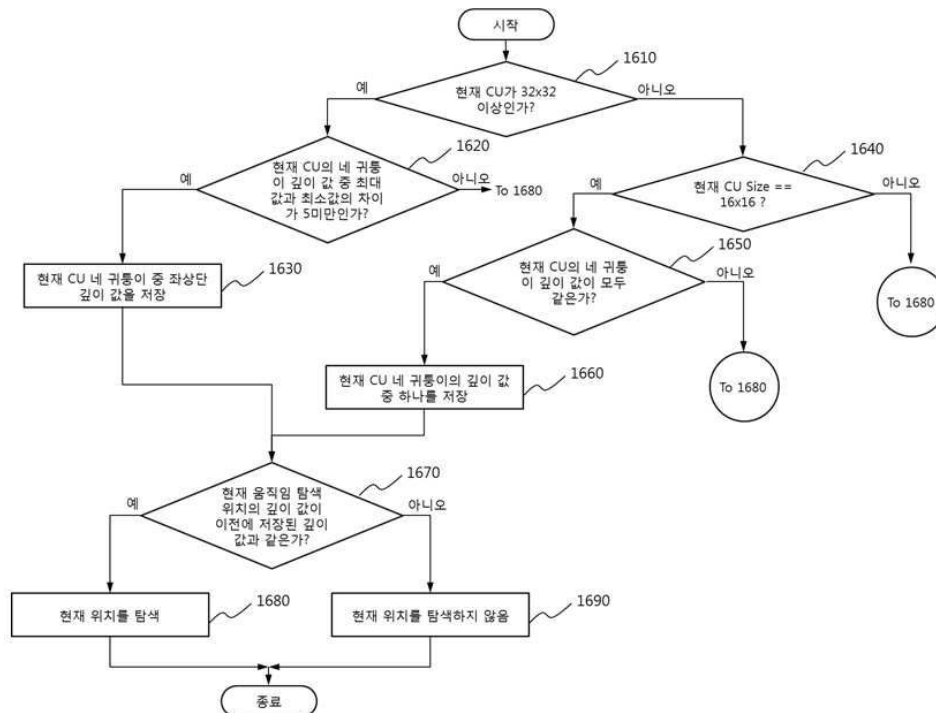
도면14



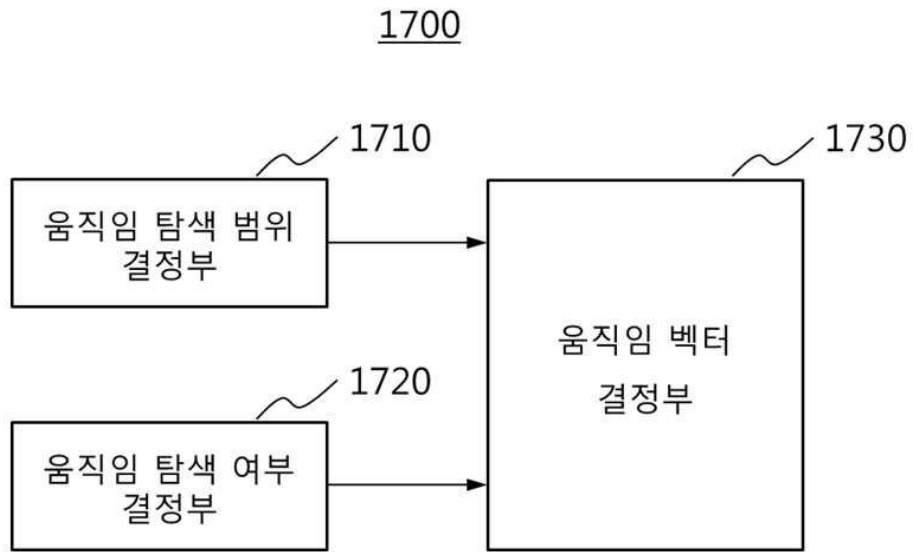
도면15



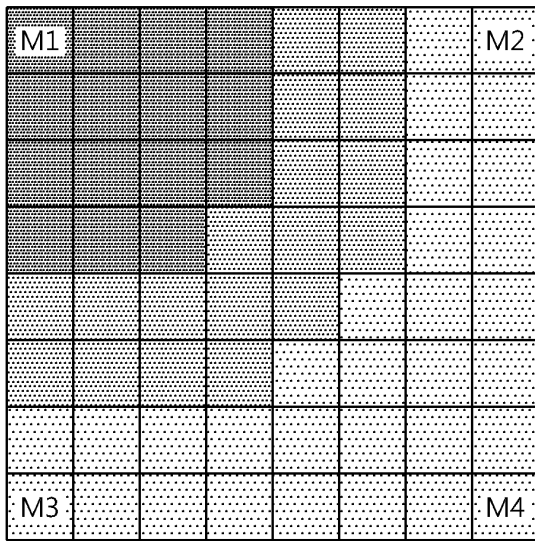
도면16



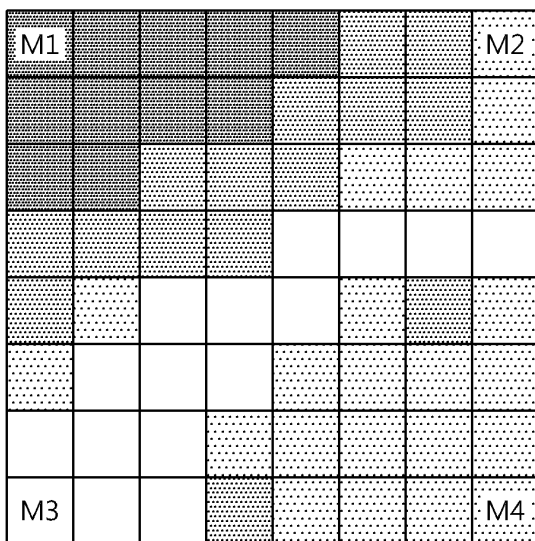
도면17



도면18



(A)



(B)