(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0128390
(43) 공개일자 2017년11월22일

(51) 국제특허분류(Int. Cl.)
H04N 19/103 (2014.01) *H04N 19/12* (2014.01)
H04N 19/159 (2014.01) *H04N 19/176* (2014.01)
H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/103 (2015.01)
H04N 19/12 (2015.01)
(21) 출원번호 10-2017-7027650
(22) 출원일자(국제) 2016년03월16일
심사청구일자 없음
(85) 번역문제출일자 2017년09월28일
(86) 국제출원번호 PCT/KR2016/002666
(87) 국제공개번호 WO 2016/148513
국제공개일자 2016년09월22일
(30) 우선권주장
62/135,176 2015년03월19일 미국(US)

(71) 출원인
엘지전자 주식회사
서울특별시 영등포구 여의대로 128 (여의도동)
(72) 발명자
전용준
서울특별시 서초구 양재대로11길 19 LG전자 특허센터
박승욱
서울특별시 서초구 양재대로11길 19 LG전자 특허센터
손은용
서울특별시 서초구 양재대로11길 19 LG전자 특허센터
(74) 대리인
김용인, 방해철

전체 청구항 수 : 총 15 항

(54) 발명의 명칭 **비디오 신호의 처리 방법 및 이를 위한 장치**

(57) 요 약

본 발명은 비디오 신호를 위한 비트스트림을 디코딩하는 방법 및 장치에 관한 것으로서, 현재 블록을 포함하는 픽처 내에서 상기 현재 블록과 인접하는 적어도 하나의 이웃 블록이 복수의 코딩 블록으로 분할되는지 여부를 판별하는 단계; 상기 적어도 하나의 이웃 블록이 분할되지 않는 경우: 상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록 중 하나로부터 유도되는지 여부를 지시하는 플래그 정보를 비트스트림으로부터 획득하는 단계, 및 상기 플래그 정보에 기초하여, 상기 적어도 하나의 이웃 블록 중에서 특정 이웃 블록으로부터 상기 현재 블록을 위한 파라미터 정보를 유도하는 단계; 상기 적어도 하나의 이웃 블록 모두가 분할되는 경우, 상기 현재 블록을 위한 파라미터 정보를 비트스트림으로부터 획득하는 단계; 및 상기 현재 블록을 위한 파라미터 정보를 기반으로 상기 현재 블록을 디코딩하는 단계를 포함하는 방법 및 이를 위한 장치에 관한 것이다.

(52) CPC특허분류
H04N 19/159 (2015.01)
H04N 19/176 (2015.01)
H04N 19/70 (2015.01)

명 세 서

청구범위

청구항 1

디코딩 장치에서 비디오 신호를 위한 비트스트림을 디코딩하는 방법으로서,

현재 블록을 포함하는 픽처 내에서 상기 현재 블록과 인접하는 적어도 하나의 이웃 블록이 복수의 코딩 블록으로 분할되는지 여부를 판별하는 단계, 상기 적어도 하나의 이웃 블록은 상기 현재 블록에 인접한 좌측 이웃 블록 및 상측 이웃 블록을 포함하며;

상기 적어도 하나의 이웃 블록이 분할되지 않는 경우:

상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록 중 하나로부터 유도되는지 여부를 지시하는 플래그 정보를 비트스트림으로부터 획득하는 단계, 및

상기 플래그 정보가 상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록으로부터 유도됨을 지시하는 경우, 상기 적어도 하나의 이웃 블록 중에서 특정 이웃 블록으로부터 상기 현재 블록을 위한 파라미터 정보를 유도하는 단계;

상기 적어도 하나의 이웃 블록 모두가 분할되는 경우, 상기 현재 블록을 위한 파라미터 정보를 비트스트림으로부터 획득하는 단계; 및

상기 현재 블록을 위한 파라미터 정보를 기반으로 상기 현재 블록을 디코딩하는 단계를 포함하는, 방법.

청구항 2

제1항에 있어서,

상기 좌측 이웃 블록이 분할되는 경우, 상기 특정 이웃 블록은 상기 우측 이웃 블록으로 결정되는, 방법.

청구항 3

제1항에 있어서,

상기 우측 이웃 블록이 분할되는 경우, 상기 특정 이웃 블록은 상기 좌측 이웃 블록으로 결정되는, 방법.

청구항 4

제1항에 있어서,

상기 좌측 이웃 블록 및 상기 우측 이웃 블록이 분할되지 않는 경우, 상기 방법은 상기 적어도 하나의 이웃 블록 중에서 하나를 지시하는 인덱스 정보를 비트스트림으로부터 획득하는 단계를 더 포함하되,

상기 특정 이웃 블록은 상기 적어도 하나의 이웃 블록 중에서 상기 인덱스 정보가 지시하는 블록으로 결정되는, 방법.

청구항 5

제1항에 있어서,

상기 적어도 하나의 이웃 블록은 상기 현재 블록을 포함하는 픽처 내에서 상기 현재 블록에 인접한 좌상측 이웃 블록, 우상측 이웃 블록과, 상기 현재 블록을 포함하는 픽처와 상이한 픽처에서 상기 현재 블록에 대응되는 위치에 있는 블록을 더 포함하는, 방법.

청구항 6

제5항에 있어서,

상기 방법은, 상기 플래그 정보가 상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록으로부터 유도됨을 지시하는 경우, 상기 적어도 하나의 이웃 블록 중에서 하나를 지시하는 인덱스 정보를 비트스트림

으로부터 획득하는 단계를 더 포함하되,

상기 특정 이웃 블록은 상기 적어도 하나의 이웃 블록 중에서 상기 인덱스 정보가 지시하는 블록으로 결정되는, 방법.

청구항 7

제1항에 있어서,

상기 현재 블록을 위한 파라미터 정보는 상기 현재 블록이 절반의 수평 및 수직 크기를 가지는 복수의 코딩 블록들로 분할되는지 여부를 지시하는 정보, 현재 블록 또는 현재 블록 내에 포함된 각 코딩 블록이 인트라 예측 모드로 코딩되는지 아니면 인터 예측 모드로 코딩되는지 여부를 지시하는 정보를 포함하는 방법.

청구항 8

제1항에 있어서,

상기 현재 블록을 위한 파라미터 정보는 상기 현재 블록 또는 현재 블록 내에 포함된 각 코딩 블록에 대한 인트라 예측 모드와 관련된 정보를 포함하는, 방법.

청구항 9

제1항에 있어서,

상기 현재 블록을 위한 파라미터 정보는 상기 현재 블록 또는 현재 블록 내에 포함된 각 예측 블록에 대한 인터 예측 파라미터 정보를 포함하고,

상기 인터 예측 파라미터 정보는 참조 픽처 인덱스 정보 및 움직임 벡터 정보를 포함하는, 방법.

청구항 10

제1항에 있어서,

상기 현재 블록을 디코딩하는 단계는 상기 현재 블록 또는 상기 현재 블록에 포함되는 각 블록에 대하여,

해당 블록이 0이 아닌 변환 계수를 포함하는지 여부를 지시하는 코딩 블록 플래그 정보를 상기 비트스트림으로부터 획득하는 것,

상기 코딩 블록 플래그 정보에 기초하여 상기 해당 블록에 대한 변환 계수 정보를 상기 비트스트림으로부터 획득하는 것, 및

상기 변환 계수 정보로부터 상기 해당 블록에 대한 레지듀얼을 획득하는 것을 포함하는, 방법.

청구항 11

제1항에 있어서,

상기 픽처는 상기 현재 블록의 크기와 동일한 크기를 가지는 복수의 블록을 포함하며,

상기 방법은 상기 복수의 블록 각각에 대하여 상기 단계들을 수행하는 것을 더 포함하는, 방법.

청구항 12

제10항에 있어서,

코딩 블록의 최소 크기를 지시하는 정보를 상기 비트스트림으로부터 획득하는 단계;

코딩 블록의 최소 크기와 코딩 블록의 최대 크기 간의 차이를 지시하는 정보를 상기 비트스트림으로부터 획득하는 단계; 및

상기 코딩 블록의 최소 크기를 지시하는 정보와 상기 차이를 지시하는 정보를 이용하여 상기 현재 블록의 크기를 결정하는 단계를 더 포함하는, 방법.

청구항 13

제1항에 있어서,

상기 특정 이웃 블록으로부터 상기 현재 블록을 위한 파라미터 정보를 유도하는 단계는,

상기 특정 이웃 블록의 파라미터 정보를 상기 현재 블록을 위한 파라미터 정보로 설정하는 것을 포함하는, 방법.

청구항 14

제1항에 있어서,

코딩 블록은 동일한 인트라 예측 모드 또는 인터 예측 모드가 적용되는 단위를 나타내는, 방법.

청구항 15

비디오 신호(video signal)를 위한 비트스트림을 디코딩하도록 구성된 디코딩 장치로서,

메모리; 및

상기 메모리에 동작시 연결되는(operatively connected) 프로세서를 포함하며, 상기 프로세서는

현재 블록을 포함하는 픽처 내에서 상기 현재 블록과 인접하는 적어도 하나의 이웃 블록이 복수의 코딩 블록으로 분할되는지 여부를 판별하고, 상기 적어도 하나의 이웃 블록은 상기 현재 블록에 인접한 좌측 이웃 블록 및 상측 이웃 블록을 포함하며;

상기 적어도 하나의 이웃 블록이 분할되지 않는 경우:

상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록 중 하나로부터 유도되는지 여부를 지시하는 플래그 정보를 비트스트림으로부터 획득하고, 및

상기 플래그 정보가 상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록으로부터 유도됨을 지시하는 경우, 상기 적어도 하나의 이웃 블록 들 중에서 특정 이웃 블록으로부터 상기 현재 블록을 위한 파라미터 정보를 유도하고;

상기 적어도 하나의 이웃 블록 모두가 분할되는 경우, 상기 현재 블록을 위한 파라미터 정보를 비트스트림으로부터 획득하고; 및

상기 현재 블록을 위한 파라미터 정보를 기반으로 상기 현재 블록을 디코딩하도록 구성된, 디코딩 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 비디오 처리 방법에 관한 것으로서, 보다 구체적으로는 블록 간의 병합을 이용한 비디오 신호의 처리 방법 및 이를 위한 장치에 관한 것이다.

배 경 기 술

[0002] 디지털 동영상 처리 기술이 급격히 발전함에 따라 고화질 디지털방송, 디지털 멀티미디어 방송, 인터넷 방송 등과 같은 다양한 매체를 이용한 디지털 멀티미디어 서비스가 활성화되고 있으며, 고화질 디지털 방송이 일반화되면서 다양한 서비스 애플리케이션이 개발되고 있고, 고화질, 고해상도의 영상을 위한 고속 동영상 처리 기술들이 요구되고 있다. 이를 위해, H.265/HEVC(High Efficiency Video Coding), H.264/AVC(Advanced Video Coding)와 같은 비디오 신호의 코딩에 관한 표준이 활발히 논의되고 있다.

발명의 내용

해결하려는 과제

[0003] 본 발명의 목적은 비디오 신호를 효율적으로 처리할 수 있는 방법 및 이를 위한 장치를 제공하는 데 있다.

[0004] 또한, 본 발명의 목적은 비디오 신호의 코딩에 필요한 비트 수를 줄임으로써 코딩 효율을 향상시키는 데 있다.

[0005] 또한, 본 발명의 목적은 비디오 코딩의 기본 처리 단위를 유연한 형태로 확장시킬 수 있게 함으로써 코딩 효율

을 향상시키는 데 있다.

[0006] 본 발명에서 이루고자 하는 기술적 과제들은 상기 기술적 과제로 제한되지 않으며, 언급하지 않은 또 다른 기술적 과제들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0007] 본 발명의 제1 양상으로서, 디코딩 장치에서 비디오 신호를 위한 비트스트림을 디코딩하는 방법이 제공되며, 상기 방법은 현재 블록을 포함하는 픽처 내에서 상기 현재 블록과 인접하는 적어도 하나의 이웃 블록이 복수의 코딩 블록으로 분할되는지 여부를 판별하는 단계, 상기 적어도 하나의 이웃 블록은 상기 현재 블록에 인접한 좌측 이웃 블록 및 상측 이웃 블록을 포함하며; 상기 적어도 하나의 이웃 블록이 분할되지 않는 경우: 상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록 중 하나로부터 유도되는지 여부를 지시하는 플래그 정보를 비트스트림으로부터 획득하는 단계, 및 상기 플래그 정보가 상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록으로부터 유도됨을 지시하는 경우, 상기 적어도 하나의 이웃 블록 중에서 특정 이웃 블록으로부터 상기 현재 블록을 위한 파라미터 정보를 유도하는 단계; 상기 적어도 하나의 이웃 블록 모두가 분할되는 경우, 상기 현재 블록을 위한 파라미터 정보를 비트스트림으로부터 획득하는 단계; 및 상기 현재 블록을 위한 파라미터 정보를 기반으로 상기 현재 블록을 디코딩하는 단계를 포함할 수 있다.

[0008] 본 발명의 제2 양상으로서, 비디오 신호(video signal)를 위한 비트스트림을 디코딩하도록 구성된 디코딩 장치가 제공되며, 상기 디코딩 장치는 메모리; 및 상기 메모리에 동작시 연결되는(operatively connected) 프로세서를 포함하며, 상기 프로세서는 현재 블록을 포함하는 픽처 내에서 상기 현재 블록과 인접하는 적어도 하나의 이웃 블록이 복수의 코딩 블록으로 분할되는지 여부를 판별하고, 상기 적어도 하나의 이웃 블록은 상기 현재 블록에 인접한 좌측 이웃 블록 및 상측 이웃 블록을 포함하며; 상기 적어도 하나의 이웃 블록이 분할되지 않는 경우: 상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록 중 하나로부터 유도되는지 여부를 지시하는 플래그 정보를 비트스트림으로부터 획득하고, 및 상기 플래그 정보가 상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록으로부터 유도됨을 지시하는 경우, 상기 적어도 하나의 이웃 블록 들 중에서 특정 이웃 블록으로부터 상기 현재 블록을 위한 파라미터 정보를 유도하고; 상기 적어도 하나의 이웃 블록 모두가 분할되는 경우, 상기 현재 블록을 위한 파라미터 정보를 비트스트림으로부터 획득하고; 및 상기 현재 블록을 위한 파라미터 정보를 기반으로 상기 현재 블록을 디코딩하도록 구성될 수 있다.

[0009] 바람직하게는, 상기 좌측 이웃 블록이 분할되는 경우, 상기 특정 이웃 블록은 상기 우측 이웃 블록으로 결정될 수 있다.

[0010] 바람직하게는, 상기 우측 이웃 블록이 분할되는 경우, 상기 특정 이웃 블록은 상기 좌측 이웃 블록으로 결정될 수 있다.

[0011] 바람직하게는, 상기 좌측 이웃 블록 및 상기 우측 이웃 블록이 분할되지 않는 경우, 상기 방법은 상기 적어도 하나의 이웃 블록 중에서 하나를 지시하는 인덱스 정보를 비트스트림으로부터 획득하는 단계를 더 포함하되, 상기 특정 이웃 블록은 상기 적어도 하나의 이웃 블록 중에서 상기 인덱스 정보가 지시하는 블록으로 결정될 수 있다.

[0012] 바람직하게는, 상기 적어도 하나의 이웃 블록은 상기 현재 블록을 포함하는 픽처 내에서 상기 현재 블록에 인접한 좌상측 이웃 블록, 우상측 이웃 블록과, 상기 현재 블록을 포함하는 픽처와 상이한 픽처에서 상기 현재 블록에 대응되는 위치에 있는 블록을 더 포함할 수 있다.

[0013] 바람직하게는, 상기 방법은, 상기 플래그 정보가 상기 현재 블록을 위한 파라미터 정보가 상기 적어도 하나의 이웃 블록으로부터 유도됨을 지시하는 경우, 상기 적어도 하나의 이웃 블록 중에서 하나를 지시하는 인덱스 정보를 비트스트림으로부터 획득하는 단계를 더 포함하되, 상기 특정 이웃 블록은 상기 적어도 하나의 이웃 블록 중에서 상기 인덱스 정보가 지시하는 블록으로 결정될 수 있다.

[0014] 바람직하게는, 상기 현재 블록을 위한 파라미터 정보는 상기 현재 블록이 절반의 수평 및 수직 크기를 가지는 복수의 코딩 블록들로 분할되는지 여부를 지시하는 정보, 현재 블록 또는 현재 블록 내에 포함된 각 코딩 블록이 인트라 예측 모드로 코딩되는지 아니면 인터 예측 모드로 코딩되는지 여부를 지시하는 정보를 포함할 수 있다.

[0015] 바람직하게는, 상기 현재 블록을 위한 파라미터 정보는 상기 현재 블록 또는 현재 블록 내에 포함된 각 코딩 블

록에 대한 인트라 예측 모드와 관련된 정보를 포함할 수 있다.

[0016] 바람직하게는, 상기 현재 블록을 위한 파라미터 정보는 상기 현재 블록 또는 현재 블록 내에 포함된 각 예측 블록에 대한 인터 예측 파라미터 정보를 포함하고, 상기 인터 예측 파라미터 정보는 참조 픽처 인덱스 정보 및 움직임 벡터 정보를 포함할 수 있다.

[0017] 바람직하게는, 상기 현재 블록을 디코딩하는 것은 상기 현재 블록 또는 상기 현재 블록에 포함되는 각 블록에 대하여, 해당 블록이 0이 아닌 변환 계수를 포함하는지 여부를 지시하는 코딩 블록 플래그 정보를 상기 비트스트림으로부터 획득하는 것, 상기 코딩 블록 플래그 정보에 기초하여 상기 해당 블록에 대한 변환 계수 정보를 상기 비트스트림으로부터 획득하는 것, 및 상기 변환 계수 정보로부터 상기 해당 블록에 대한 레지듀얼을 획득하는 것을 포함할 수 있다.

[0018] 바람직하게는, 상기 픽처는 상기 현재 블록의 크기와 동일한 크기를 가지는 복수의 블록을 포함하며, 상기 방법은 상기 복수의 블록 각각에 대하여 상기 단계들을 수행하는 것을 더 포함할 수 있다.

[0019] 바람직하게는, 상기 방법은 코딩 블록의 최소 크기를 지시하는 정보를 상기 비트스트림으로부터 획득하는 단계; 코딩 블록의 최소 크기와 코딩 블록의 최대 크기 간의 차이를 지시하는 정보를 상기 비트스트림으로부터 획득하는 단계; 및 상기 코딩 블록의 최소 크기를 지시하는 정보와 상기 차이를 지시하는 정보를 이용하여 상기 현재 블록의 크기를 결정하는 단계를 더 포함할 수 있다.

[0020] 바람직하게는, 상기 특정 이웃 블록으로부터 상기 현재 블록을 위한 파라미터 정보를 유도하는 것은, 상기 특정 이웃 블록의 파라미터 정보를 상기 현재 블록을 위한 파라미터 정보로 설정하는 것을 포함할 수 있다.

[0021] 바람직하게는, 코딩 블록은 동일한 인트라 예측 모드 또는 인터 예측 모드가 적용되는 단위를 나타낼 수 있다.

발명의 효과

[0022] 본 발명에 의하면, 비디오 신호를 효율적으로 처리할 수 있다.

[0023] 또한, 본 발명에 의하면, 비디오 신호의 코딩에 필요한 비트 수를 줄임으로써 코딩 효율을 향상시킬 수 있다.

[0024] 또한, 본 발명에 의하면, 비디오 코딩의 기본 처리 단위를 유연한 형태로 확장시킬 수 있게 함으로써 코딩 효율을 향상시킬 수 있다.

[0025] 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0026] 첨부 도면은 본 발명에 관한 이해를 돕기 위해 상세한 설명의 일부로 포함되며, 본 발명에 대한 실시예를 제공하고, 상세한 설명과 함께 본 발명의 기술적 사상을 설명한다.

도 1은 인코딩 과정을 예시한다.

도 2는 디코딩 과정을 예시한다.

도 3은 코딩 트리 유닛(CTU)을 분할하는 방법의 순서도를 예시한다.

도 4는 CTU를 쿼드 트리 방식으로 분할하는 예를 예시한다.

도 5는 코딩 유닛을 위한 신택스 정보 및 동작을 예시한다.

도 6은 변환 트리에 대한 신택스 정보 및 동작을 예시한다.

도 7은 본 발명에 따른 CTU 병합 모드가 적용되는 예를 예시한다.

도 8은 본 발명에 따른 CTU 병합 모드와 크기가 확장된 CTU를 비교한 것이다.

도 9는 본 발명에 따른 CTU 병합 모드의 실시예를 예시한다.

도 10은 본 발명에 따른 방법의 순서도를 예시한다.

도 11은 본 발명이 적용될 수 있는 영상 처리 장치의 블록도를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0027] 이하의 기술은 비디오 신호(video signal)를 인코딩(encoding) 및/또는 디코딩하도록 구성된 영상 신호 처리 장치에서 사용될 수 있다. 일반적으로 비디오 신호는 눈으로 인지가능한 영상 신호(image signal) 또는 픽처들의 시퀀스를 지칭하지만, 본 명세서에서 비디오 신호는 코딩된 픽처(picture)를 나타내는 비트들의 시퀀스(sequence) 또는 비트 시퀀스에 해당하는 비트스트림을 지칭하는 데 사용될 수 있다. 픽처(picture)는 샘플들의 배열을 지칭할 수 있으며, 프레임(frame), 영상(image) 등으로 지칭될 수 있다. 보다 구체적으로, 픽처는 샘플들의 이차원 배열 또는 이차원 샘플 배열을 지칭할 수 있다. 샘플은 픽처를 구성하는 최소 단위를 지칭할 수 있고, 픽셀(pixel), 화소(picture element), 펠(pel) 등으로 지칭될 수 있다. 샘플은 휘도(luminance, luma) 성분 및/또는 색차(chrominance, chroma, color difference) 성분을 포함할 수 있다. 본 명세서에서, 코딩은 인코딩을 지칭하는 데 사용될 수도 있고, 혹은 인코딩/디코딩을 통칭할 수 있다.

[0028] 픽처는 적어도 하나의 슬라이스를 포함할 수 있으며, 슬라이스는 적어도 하나의 블록을 포함할 수 있다. 슬라이스는 병렬 처리 등의 목적, 데이터 손실 등으로 인해 비트스트림이 훼손된 경우 디코딩의 재동기화 등의 목적을 위해 정수 개의 블록을 포함하도록 구성될 수 있으며, 각 슬라이스는 서로 독립적으로 코딩될 수 있다. 블록은 적어도 하나의 샘플을 포함할 수 있으며, 샘플들의 배열을 지칭할 수 있다. 블록은 픽처보다 작거나 같은 크기를 가질 수 있다. 블록은 유닛으로 지칭될 수 있다. 현재 코딩되는 픽처를 현재 픽처라고 지칭하고, 현재 코딩되는 블록을 현재 블록이라고 지칭할 수 있다. 픽처를 구성하는 다양한 블록 단위가 존재할 수 있으며, 예를 들어 ITU-T H.265 표준(또는 HEVC(High Efficiency Video Coding) 표준)의 경우 코딩 트리 블록(CTB)(또는 코딩 트리 유닛(CTU)), 코딩 블록(CB)(또는 코딩 유닛(CU)), 예측 블록(PB)(또는 예측 유닛(PU)), 변환 블록(TB)(또는 변환 유닛(TU)) 등의 블록 단위가 존재할 수 있다.

[0029] 코딩 트리 블록은 픽처를 구성하는 가장 기본적인 단위를 지칭하며, 픽처의 텍스처(texture)에 따라 코딩 효율을 높이기 위해 쿼드-트리(quad-tree) 형태의 코딩 블록들로 분할될 수 있다. 코딩 블록은 코딩을 수행하는 기본 단위를 지칭할 수 있으며, 코딩 블록 단위로 인트라 코딩 또는 인터 코딩이 수행될 수 있다. 인트라 코딩은 인트라 예측을 이용하여 코딩을 수행하는 것을 지칭할 수 있으며, 인트라 예측은 동일한 픽처 또는 슬라이스 내에 포함된 샘플들을 이용하여 예측을 수행하는 것을 지칭할 수 있다. 인터 코딩은 인터 예측을 이용하여 코딩을 수행하는 것을 지칭할 수 있으며, 인터 예측은 현재 픽처와 서로 다른 픽처에 포함된 샘플들을 이용하여 예측을 수행하는 것을 지칭할 수 있다. 인트라 코딩을 이용하여 코딩되는 블록 또는 인트라 예측 모드로 코딩된 블록을 인트라 블록이라고 지칭할 수 있고, 인터 코딩을 이용하여 코딩되는 블록 또는 인터 예측 모드로 코딩된 블록을 인터 블록이라고 지칭할 수 있다. 또한, 인트라 예측을 이용한 코딩 모드를 인트라 모드라고 지칭할 수 있고, 인터 예측을 이용한 코딩 모드를 인터 모드라고 지칭할 수 있다.

[0030] 예측 블록은 예측을 수행하기 위한 기본 단위를 지칭할 수 있다. 하나의 예측 블록에 대해서는 동일한 예측이 적용될 수 있다. 예를 들어, 인터 예측의 경우 하나의 예측 블록에 대해서 동일한 움직임 벡터가 적용될 수 있다. 변환 블록은 변환을 수행하기 위한 기본 단위를 지칭할 수 있다. 변환은 픽셀 도메인(또는 공간 도메인 또는 시간 도메인)의 샘플들을 주파수 도메인(또는 변환 계수 도메인)의 변환 계수로 변환하는 동작을 지칭하거나, 그 반대의 동작을 통칭할 수 있다. 특히, 주파수 도메인(또는 변환 계수 도메인)의 변환 계수를 픽셀 도메인(또는 공간 도메인 또는 시간 도메인)의 샘플들로 변환하는 동작을 역변환이라고 지칭할 수 있다. 예를 들어, 변환은 이산 코사인 변환(DCT), 이산 사인 변환(DST), 푸리어 변환 등을 포함할 수 있다.

[0031] 본 명세서에서, 코딩 트리 블록(CTB)은 코딩 트리 유닛(CTU)과 혼용될 수 있고, 코딩 블록(CB)은 코딩 유닛(CU)과 혼용될 수 있고, 예측 블록(PB)은 예측 유닛(PU)과 혼용될 수 있고, 변환 블록(PB)은 변환 유닛(PU)과 혼용될 수 있다.

[0032] 도 1은 인코딩 과정을 예시한다.

[0033] 인코딩 장치(100)는 원영상(original image)(102)을 입력받아 인코딩을 수행한 다음 비트스트림(114)을 출력한다. 원영상(102)은 하나의 픽처에 해당할 수 있지만, 본 예에서 원영상(102)은 픽처를 구성하는 하나의 블록이라고 가정한다. 예를 들어, 원영상(102)은 코딩 블록에 해당할 수 있다. 인코딩 장치(100)는 원영상(102)에 대하여 인트라 모드로 코딩할지 인터 모드로 코딩할지 결정할 수 있다. 원영상(102)이 인트라 픽처 또는 슬라이스에 포함되는 경우, 원영상(102)은 인트라 모드로만 코딩될 수 있다. 하지만, 원영상(102)이 인터 픽처 또는 슬라이스에 포함되는 경우, 예를 들어 원영상(102)에 대하여 인트라 코딩 및 인터 코딩을 수행한 다음 RD(Rate-Distortion) 비용(cost)을 대비하여 효율적인 코딩 방법을 결정할 수 있다.

[0034] 원영상(102)에 대해 인트라 코딩을 수행하는 경우, 인코딩 장치(100)는 원영상(102)을 포함하는 현재 픽처의 복원 샘플들을 이용하여 RD 최적화를 보여주는 인트라 예측 모드를 결정할 수 있다(104). 예를 들어, 인트라 예측 모드는 DC(Direct Current) 예측 모드, 평면(planar) 예측 모드, 각도(angular) 예측 모드 중에서 하나로 결정될 수 있다. DC 예측 모드는 현재 픽처의 복원 샘플들 중에서 참조 샘플들의 평균값을 이용하여 예측을 수행하는 모드를 지칭하고, 평면 예측 모드는 참조 샘플들의 이중 선형 보간(bilinear interpolation)을 이용하여 예측을 수행하는 모드를 지칭하고, 각도 예측 모드는 원영상(102)에 대해 특정 방향에 위치한 참조 샘플을 이용하여 예측을 수행하는 모드를 지칭한다. 인코딩 장치(100)는 결정된 인트라 예측 모드를 이용하여 예측 샘플(predicted sample) 또는 예측값(prediction value)(또는 predictor)(107)을 출력할 수 있다.

[0035] 원영상(102)에 대해 인터 코딩을 수행하는 경우, 인코딩 장치(100)는 (디코딩된) 픽처 버퍼(122)에 포함된 복원 픽처(reconstructed picture)를 이용하여 움직임 추정(motion estimation, ME)을 수행하여 움직임 정보를 획득할 수 있다(106). 예를 들어, 움직임 정보는 움직임 벡터, 참조 픽처 인덱스 등을 포함할 수 있다. 움직임 벡터는 현재 픽처 내에서 원영상(102)의 좌표로부터 참조 픽처 내의 좌표까지의 오프셋을 제공하는 이차원 벡터를 지칭할 수 있다. 참조 픽처 인덱스는 (디코딩된) 픽처 버퍼(122)에 저장된 복원 픽처(reconstructed picture) 중에서 인터 예측을 위해 사용되는 참조 픽처들의 리스트(또는 참조 픽처 리스트)에 대한 인덱스를 지칭할 수 있으며, 참조 픽처 리스트에서 해당 참조 픽처를 가리킨다. 인코딩 장치(100)는 획득한 움직임 정보를 이용하여 예측 샘플 또는 예측값(107)을 출력할 수 있다.

[0036] 그런 다음, 인코딩 장치(100)는 원영상(102)과 예측 샘플(107) 간의 차이로부터 레지듀얼 데이터(108)를 생성할 수 있다. 인코딩 장치(100)는 생성된 레지듀얼 데이터(108)에 대해 변환을 수행할 수 있다(110). 예를 들어, 변환을 위해 이산 코사인 변환 (Discrete Cosine Transform, DCT), 이산 사인 변환 (Discrete Sine Transform, DST) 및/또는 웨이블릿 변환(Wavelet Transform) 등이 적용될 수 있다. 보다 구체적으로, 4×4 내지 32×32 크기의 정수 기반 DCT가 사용될 수 있으며, 4×4, 8×8, 16×16, 32×32 변환이 이용될 수 있다. 인코딩 장치(100)는 변환(110)을 수행하여 변환 계수 정보를 획득할 수 있다.

[0037] 인코딩 장치(100)는 변환 계수 정보를 양자화하여 양자화된 변환 계수 정보를 생성할 수 있다(112). 양자화는 양자화 파라미터(QP)를 이용하여 변환 계수 정보의 레벨을 스케일링하는 동작을 지칭할 수 있다. 따라서, 양자화된 변환 계수 정보는 스케일링된 변환 계수 정보라고 지칭될 수 있다. 양자화된 변환 계수 정보는 엔트로피 코딩(114)을 통해 비트스트림(116)으로 출력될 수 있다. 예를 들어, 엔트로피 코딩(114)은 고정 길이 코딩(fixed length coding, FLC), 가변 길이 코딩(variable length coding, VLC), 산술 코딩(arithmetic coding)을 기반으로 수행될 수 있다. 보다 구체적으로, 산술 부호화를 기반으로 한 문맥 기반 적응적 이진 산술 코딩(context adaptive binary arithmetic coding, CABAC), 가변 길이 코딩을 기반으로 한 Exp-Golomb 코딩, 및 고정 길이 코딩이 적용될 수 있다.

[0038] 또한, 인코딩 장치(100)는 양자화된 변환 계수 정보에 대해 역양자화(118) 및 역변환(120)을 수행하여 복원 샘플(121)를 생성할 수 있다. 도 1에 예시되지 않았지만, 하나의 픽처에 대하여 복원 샘플(121)을 획득하여 복원 픽처를 생성한 다음 복원 픽처에 대해 인루프 필터링이 수행될 수 있다. 인루프 필터링을 위해 예를 들어 디블록킹(deblocking) 필터, 샘플 적응적 오프셋(sample adaptive offset, SAO) 필터가 적용될 수 있다. 그런 다음, 복원 픽처(121)는 픽처 버퍼(122)에 저장되어 다음 픽처의 인코딩에 사용될 수 있다.

[0039] 도 2는 디코딩 과정을 예시한다.

[0040] 디코딩 장치(200)는 비트스트림(202)을 수신하여 엔트로피 디코딩(204)을 수행할 수 있다. 엔트로피 디코딩(204)은 도 1의 엔트로피 코딩(114)의 역방향 동작을 지칭할 수 있다. 디코딩 장치(200)는 엔트로피 디코딩(204)을 통해 예측 모드 정보, 인트라 예측 모드 정보, 움직임 정보 등을 포함하여 디코딩에 필요한 데이터 및 (양자화된) 변환 계수 정보를 획득할 수 있다. 디코딩 장치(200)는 획득된 변환 계수 정보에 대해 역양자화(206) 및 역변환(208)을 수행하여 레지듀얼 데이터(209)를 생성할 수 있다.

[0041] 엔트로피 디코딩(204)을 통해 획득되는 예측 모드 정보는 현재 블록이 인트라 모드로 코딩되는지 인터 모드로 코딩되는지 여부를 지시할 수 있다. 예측 모드 정보가 인트라 모드를 지시하는 경우, 디코딩 장치(200)는 엔트로피 디코딩(204)을 통해 획득된 인트라 예측 모드에 기초하여 현재 픽처의 복원 샘플들로부터 예측 샘플(또는 예측값)(213)을 획득할 수 있다(210). 예측 모드 정보가 인터 모드를 지시하는 경우, 디코딩 장치(200)는 엔트로피 디코딩(204)을 통해 획득된 움직임 정보에 기초하여 픽처 버퍼(214)에 저장된 참조 픽처로부터 예측 샘플(또는 예측값)(213)을 획득할 수 있다(212).

[0042] 디코딩 장치(200)는 레지듀얼 데이터(209)와 예측 샘플(또는 예측값)(213)을 이용하여 현재 블록에 대한 복원 샘플(216)을 획득할 수 있다. 도 2에 예시되지 않았지만, 하나의 픽처에 대하여 복원 샘플(216)을 획득하여 픽처를 복원한 다음 복원 픽처에 대해 인루프 필터링이 수행될 수 있다. 그런 다음, 복원 픽처(216)는 다음 픽처의 디코딩을 위해 픽처 버퍼에 저장되거나 디스플레이를 위해 출력될 수 있다.

[0043] 비디오 인코딩/디코딩 프로세스는 소프트웨어(SW)/하드웨어(HW) 처리시 매우 높은 복잡도가 요구된다. 따라서, 제한된 자원(resource)을 이용하여 복잡도가 높은 작업을 수행하기 위해 픽처(또는 영상)을 최소의 처리 단위인 기본 처리 단위(processing unit)로 분할하여 처리할 수 있다. 따라서, 하나의 슬라이스는 적어도 하나의 기본 처리 단위를 포함할 수 있다. 이 경우, 하나의 픽처 또는 슬라이스에 포함되는 기본 처리 단위는 동일한 크기를 가질 수 있다.

[0044] HEVC(High Efficiency Video Coding) 표준(ISO/IEC 23008-2 또는 ITU-T H.265)의 경우 앞서 설명한 바와 같이 기본 처리 단위는 CTB(Coding Tree Block) 또는 CTU(Coding Tree Unit)로 지칭될 수 있으며, 64×64 픽셀의 크기를 가질 수 있다. 따라서, HEVC 표준의 경우 하나의 픽처는 기본 처리 단위인 CTU로 분할(partitioning)되어 인코딩/디코딩을 수행할 수 있다. 보다 구체적인 예로, 8192×4096 픽처를 인코딩/디코딩하는 경우 픽처는 128×64=8192개의 CTU로 나누어 8192개의 CTU에 대해 도 1에 예시된 인코딩 절차 또는 도 2에 예시된 디코딩 절차를 수행할 수 있다.

[0045] 비디오 신호 또는 비트스트림은 시퀀스 파라미터 세트(SPS), 픽처 파라미터 세트(PPS), 적어도 하나의 액세스 유닛을 포함할 수 있다. 시퀀스 파라미터 세트는 (픽처들의) 시퀀스 레벨의 파라미터 정보를 포함하며, 시퀀스 파라미터 세트의 파라미터 정보는 픽처들의 시퀀스에서 각 픽처에 적용될 수 있다. 픽처 파라미터 세트는 픽처 레벨의 파라미터 정보를 포함하며, 픽처 파라미터 세트의 정보는 픽처에 포함되는 각 슬라이스에 적용될 수 있다. 액세스 유닛은 하나의 픽처에 대응되는 유닛을 지칭하며, 적어도 하나의 슬라이스를 포함할 수 있다. 슬라이스는 정수 개의 CTU를 포함할 수 있다. 신택스 정보는 비트스트림에 포함된 데이터를 지칭하고, 신택스 구조는 특정 순서로 비트스트림에 존재하는 신택스 정보의 구조를 지칭한다.

[0046] 코딩 트리 블록의 크기는 SPS의 파라미터 정보를 이용하여 결정될 수 있다. SPS는 코딩 블록의 최소 크기를 지시하는 제1 정보와 코딩 블록의 최소 크기와 최대 크기 간의 차이를 지시하는 제2 정보를 포함할 수 있다. 일반적으로 블록의 크기는 2의 거듭제곱으로 표현될 수 있으므로 각 정보는 실제 값의 log2 값으로 표현될 수 있다. 따라서, 코딩 블록의 최소 크기의 log2 값은 제1 정보의 값에 특정 오프셋(예, 3)을 더하여 구할 수 있고, 코딩 트리 블록의 크기의 log2 값은 코딩 블록의 최소 크기의 log2 값에 제2 정보의 값을 더하여 구할 수 있다. 코딩 트리 블록의 크기는 1을 log2 값만큼 좌측 시프트하여 구할 수 있다.

[0047] 도 3은 코딩 트리 유닛(CTU)을 분할하는 방법의 순서도를 예시한다.

[0048] HEVC 표준에서는 압축효율 제고를 위해 CTU를 쿼드 트리 방식으로 적어도 하나의 코딩 유닛(CU)으로 분할한 후 코딩 유닛에 대해 인트라 예측 모드 또는 인터 예측 모드를 결정할 수 있다. CTU가 분할되지 않는 경우 CTU는 CU에 해당할 수 있으며, 이 경우 CU는 CTU와 동일한 크기를 가질 수 있으며 해당 CTU에 대해 인트라 예측 모드 또는 인터 예측 모드가 결정될 수 있다.

[0049] CTU가 쿼드 트리 방식으로 분할될 때 재귀적으로 분할될 수 있다. CTU는 4개의 유닛으로 분할된 다음 각 분할된 유닛은 쿼드 트리 방식으로 하위 유닛으로 다시 추가적으로 분할될 수 있다. CTU를 쿼드 트리 방식으로 재귀적으로 분할하여 최종적으로 생성되는 각 유닛이 코딩 유닛이 될 수 있다. 예를 들어, CTU가 제1, 2, 3, 4 블록을 분할된 다음, 제1 블록이 제5, 6, 7, 8 블록으로 분할되고, 제2, 3, 4 블록이 분할되지 않는 경우 제2, 3, 4, 5, 6, 7, 8 블록이 코딩 블록으로 결정될 수 있다. 이 예에서, 제2, 3, 4, 5, 6, 7, 8 블록 각각에 대해 인트라 예측 모드 또는 인터 예측 모드가 결정될 수 있다.

[0050] CTU가 코딩 유닛으로 분할되는지 여부는 RD(rate distortion) 효율을 고려하여 인코더 측에서 결정될 수 있으며, 분할 여부를 지시하는 정보를 비트스트림에 포함시킬 수 있다. 본 명세서에서, CTU 또는 코딩 유닛이 절반의 수평/수직 크기를 가지는 코딩 유닛으로 분할되는지 여부를 지시하는 정보는 split_cu_flag라고 지칭될 수 있다. 설명의 편의를 위해, CTU 내에서 블록이 분할되는지 여부를 지시하는 정보는 코딩 유닛을 위한 분할 지시 정보라고 지칭될 수 있다. 디코더 측에서는 코딩 쿼드 트리 내에서 각 코딩 유닛에 대해서 분할 여부를 지시하는 정보를 비트스트림으로부터 획득하여 코딩 유닛의 분할 여부를 결정하고 쿼드 트리 방식으로 코딩 유닛을 재귀적으로 분할할 수 있다. CTU가 재귀적으로 분할하여 형성되는 코딩 유닛의 트리 구조를 코딩 트리 또는 코딩 쿼드 트리라고 지칭한다. 코딩 트리 내에서 각 코딩 유닛이 더 이상 분할되지 않는 경우 해당 유닛은 최종

적으로 코딩 유닛으로 지정될 수 있다.

[0051] 앞서 설명한 바와 같이, 코딩 유닛은 예측을 수행하기 위해 적어도 하나의 예측 유닛으로 분할될 수 있다. 또한, 코딩 유닛은 변환을 수행하기 위해 적어도 하나의 변환 유닛으로 분할될 수 있다. CTU와 유사한 방식으로, 코딩 유닛은 쿼드 트리 방식으로 재귀적으로 변환 유닛으로 분할될 수 있다. 코딩 유닛을 쿼드 트리 방식으로 재귀적으로 분할하여 형성되는 구조를 변환 트리 또는 변환 쿼드 트리라고 지칭할 수 있으며, 분할 지시 정보와 유사하게 변환 트리 내에서 각 유닛이 분할되는지 여부를 지시하는 정보가 비트스트림에 포함될 수 있다. 본 명세서에서 변환을 위해 유닛이 절반의 수평/수직 크기를 가지는 유닛으로 분할되는지 여부를 지시하는 정보는 split_transform_flag라고 지칭될 수 있다. 설명의 편의를 위해, 변환 트리에서 각 유닛이 분할되는지 여부를 지시하는 정보는 변환 유닛을 위한 분할 지시 정보라고 지칭될 수 있다.

[0052] 도 4는 CTU를 쿼드 트리 방식으로 분할하는 예를 예시한다.

[0053] 도 4를 참조하면, CTU는 블록 1-7을 포함하는 제1 코딩 유닛, 블록 8-17을 포함하는 제2 코딩 유닛, 블록 18에 해당하는 제3 코딩 유닛, 블록 19-28을 포함하는 제4 코딩 유닛으로 분할될 수 있다. 제1 코딩 유닛은 블록 1에 해당하는 코딩 유닛, 블록 2에 해당하는 코딩 유닛, 블록 3-6을 포함하는 제5 코딩 유닛, 블록 7에 해당하는 코딩 유닛으로 분할될 수 있다. 제2 코딩 유닛은 코딩 쿼드 트리 내에서는 더 이상 분할되지 않지만, 변환을 위해서는 추가적인 변환 유닛으로 분할될 수 있다. 제4 코딩 유닛은 블록 19-22를 포함하는 제6 코딩 유닛, 블록 23에 해당하는 코딩 유닛, 블록 24에 해당하는 코딩 유닛, 블록 25-28을 포함하는 제7 코딩 유닛으로 분할될 수 있다. 제6 코딩 유닛은 블록 19에 해당하는 코딩 유닛, 블록 20에 해당하는 코딩 유닛, 블록 21에 해당하는 코딩 유닛, 블록 22에 해당하는 코딩 유닛으로 분할될 수 있다. 제7 코딩 유닛은 코딩 쿼드 트리 내에서는 더 이상 분할되지 않지만, 변환을 위해서는 추가적인 변환 유닛으로 분할될 수 있다.

[0054] 앞서 설명된 바와 같이, CTU 또는 코딩 유닛 각각에 대해 분할 여부를 지시하는 정보(예, split_cu_flag)가 비트스트림에 포함될 수 있다. 분할 여부를 지시하는 정보가 제1 값(예, 1)을 가지는 경우 CTU 또는 각 코딩 유닛이 분할될 수 있고, 분할 여부를 지시하는 정보가 제2 값(예, 0)을 가지는 경우 CTU 또는 각 코딩 유닛은 분할되지 않는다. 분할 여부를 지시하는 정보의 값은 달라질 수 있다.

[0055] 도 4의 예에서, CTU, 제1 코딩 유닛, 제4 코딩 유닛, 제6 코딩 유닛에 대한 분할 지시 정보(예, split_cu_flag)는 제1 값(예, 1)을 가질 수 있으며, 디코더는 비트스트림으로부터 해당 유닛에 대한 분할 지시 정보를 획득하고 이 값에 따라 해당 유닛을 4개의 하위 유닛으로 분할할 수 있다. 반면, 다른 코딩 유닛들(블록 1, 2, 7, 18, 19, 20, 21, 22, 23, 24, 및 블록 3-6에 해당하는 코딩 유닛, 블록 8-17에 해당하는 코딩 유닛, 블록 25-28에 해당하는 코딩 유닛)에 대한 분할 지시 정보(예, split_cu_flag)는 제2 값(예, 0)을 가질 수 있으며, 디코더는 비트스트림으로부터 해당 유닛에 대한 분할 지시 정보를 획득하고 이 값에 따라 해당 유닛을 더 이상 분할하지 않는다.

[0056] 앞서 설명된 바와 같이, 각 코딩 유닛은 변환을 위해 변환 유닛을 위한 분할 지시 정보에 따라 쿼드 트리 방식으로 적어도 하나의 변환 유닛으로 분할될 수 있다. 도 4를 다시 참조하면, 블록 1, 2, 7, 18, 19, 20, 21, 22, 23, 24에 해당하는 코딩 유닛은 변환을 위해 분할되지 않으므로 변환 유닛은 코딩 유닛에 해당할 수 있지만, 다른 코딩 유닛(블록 3-4, 8-17, 25-28에 대응되는 코딩 유닛)은 변환을 위해 추가적으로 분할될 수 있다. 각 코딩 유닛(예, 블록 3-4, 8-17, 25-28에 대응되는 코딩 유닛)으로부터 형성되는 변환 트리 내에서 각 유닛에 대한 분할 지시 정보(예, split_transform_flag)를 획득하고 분할 지시 정보의 값에 따라 변환 유닛으로 분할할 수 있다. 도 4에 예시된 바와 같이, 블록 3-6에 대응되는 코딩 유닛은 깊이(depth) 1의 변환 트리를 형성하도록 변환 유닛들로 분할될 수 있고, 블록 8-17에 대응되는 코딩 유닛은 깊이 3을 가지는 변환 트리를 형성하도록 변환 유닛들로 분할될 수 있으며, 블록 25-28에 대응되는 코딩 유닛은 깊이 1을 가지는 변환 트리를 형성하도록 변환 유닛들로 분할될 수 있다.

[0057] 도 5는 코딩 유닛을 위한 신택스 정보 및 동작을 예시하고, 도 6은 변환 트리에 대한 신택스 정보 및 동작을 예시한다. 도 5에 예시된 바와 같이, 현재 코딩 유닛에 대해 변환 트리 구조가 존재하는지 여부를 지시하는 정보가 비트스트림을 통해 시그널링될 수 있으며, 본 명세서에서 이 정보는 변환 트리 코딩 지시 정보 또는 rqt_root_cbf라고 지칭될 수 있다. 디코더는 변환 트리 코딩 지시 정보를 비트스트림으로부터 획득하고 변환 트리 코딩 지시 정보가 해당 코딩 블록에 대해 변환 트리가 존재함을 지시하는 경우 도 6에 예시된 동작을 수행할 수 있다. 만일 변환 트리 코딩 지시 정보가 해당 코딩 유닛에 대해 변환 트리가 존재하지 않음을 지시하는 경우 해당 코딩 유닛에 대한 변환 계수 정보는 존재하지 않으며 해당 코딩 유닛에 대한 (인트라 또는 인터) 예측값을 이용하여 코딩 유닛을 복원할 수 있다.

[0058] 코딩 유닛은 인트라 예측 모드 또는 인터 예측 모드로 코딩되는지 여부를 결정하는 기본 단위이다. 따라서, 각 코딩 유닛에 대해 예측 모드 정보가 비트스트림을 통해 시그널링될 수 있다. 예측 모드 정보는 해당 코딩 유닛이 인트라 예측 모드를 이용하여 코딩되는지 아니면 인터 예측 모드를 이용하여 코딩되는지를 지시할 수 있다.

[0059] 예측 모드 정보가 해당 코딩 유닛이 인트라 예측 모드로 코딩됨을 지시하는 경우, 인트라 예측 모드를 결정하는 데 사용되는 정보들이 비트스트림을 통해 시그널링될 수 있다. 예를 들어, 인트라 예측 모드를 결정하는 데 사용되는 정보는 인트라 예측 모드 참조 정보를 포함할 수 있다. 인트라 예측 모드 참조 정보는 현재 코딩 유닛의 인트라 예측 모드가 이웃 (예측) 유닛으로부터 유도되는지 여부를 지시하며, 예를 들어 prev_intra_luma_pred_flag라고 지칭될 수 있다.

[0060] 인트라 예측 모드 참조 정보가 현재 코딩 유닛의 인트라 예측 모드가 이웃 유닛으로부터 유도됨을 지시하는 경우, 이웃 유닛의 인트라 예측 모드를 이용하여 인트라 예측 모드 후보 리스트를 구성하고 구성된 후보 리스트 중에서 현재 유닛의 인트라 예측 모드를 지시하는 인덱스 정보가 비트스트림을 통해 시그널링될 수 있다. 예를 들어, 인트라 예측 모드 후보 리스트 중에서 현재 유닛의 인트라 예측 모드로 사용되는 후보 인트라 예측 모드를 지시하는 인덱스 정보는 mpm_idx라고 지칭될 수 있다. 디코더는 인트라 예측 모드 참조 정보를 비트스트림으로부터 획득하고 획득된 인트라 예측 모드 참조 정보에 기초하여 인덱스 정보를 비트스트림으로부터 획득할 수 있다. 또한, 디코더는 획득된 인덱스 정보가 지시하는 인트라 예측 모드 후보를 현재 유닛의 인트라 예측 모드로 설정할 수 있다.

[0061] 인트라 예측 모드 참조 정보가 현재 코딩 유닛의 인트라 예측 모드가 이웃 유닛으로 유도됨을 지시하지 않는 경우, 현재 유닛의 인트라 예측 모드를 가리키는 정보가 비트스트림을 통해 시그널링될 수 있다. 비트스트림을 통해 시그널링되는 정보는 예를 들어 rem_intra_luma_pred_mode라고 지칭될 수 있다. 비트스트림으로부터 획득된 정보는 인트라 예측 모드 후보 리스트의 후보들의 값들과 비교하여 크거나 같은 경우 특정값(예, 1) 만큼 증가시키는 과정을 통해 현재 유닛의 인트라 예측 모드를 획득할 수 있다.

[0062] 픽처가 크로마 성분(또는 색차 성분)을 포함하는 경우 크로마 코딩 블록에 대한 인트라 예측 모드를 지시하는 정보가 비트스트림을 통해 시그널링될 수 있다. 예를 들어, 크로마 인트라 예측 모드를 지시하는 정보는 intra_chroma_pred_mode라고 지칭될 수 있다. 크로마 인트라 예측 모드는 크로마 인트라 예측 모드를 지시하는 정보 및 앞서 설명된 바와 같이 획득된 인트라 예측 모드(또는 루마 인트라 예측 모드)를 이용하여 표 1을 기반으로 획득할 수 있다. 표 1에서 IntraPredModeY는 루마 인트라 예측 모드를 가리킨다.

표 1

[0063]

intra_chroma_pred_mode	IntraPredModeY				
	0	26	10	1	X (0 <= X <= 34)
0	34	0	0	0	0
1	26	34	26	26	26
2	10	10	34	10	10
3	1	1	1	34	1
4	0	26	10	1	X

[0064] 인트라 예측 모드는 값에 따라 다양한 예측 모드를 나타낸다. 앞에서 설명한 과정을 통해 인트라 예측 모드의 값은 표 2에 예시된 바와 같이 인트라 예측 모드와 대응될 수 있다.

표 2

[0065]

Intra prediction mode	Associated name
0	INTRA_PLANAR
1	INTRA_DC
2..34	INTRA_ANGULAR2..INTRA_ANGULAR34

[0066] 표 2에서 INTRA_PLANAR는 평면 예측 모드(planar prediction mode)를 나타내며, 현재 블록에 인접한 상측(upper) 이웃 블록의 복원 샘플(reconstructed sample), 좌측(left) 이웃 블록의 복원 샘플, 좌하측(lower-left) 이웃 블록의 복원 샘플, 우상측(right-upper) 이웃 블록의 복원 샘플에 대해 보간을 수행하여 현재 블록의 예측값을 획득하는 모드를 나타낸다. INTRA_DC는 DC(Direct Current) 예측 모드를 나타내며, 좌측 이웃 블록의 복원 샘플들과 상측 이웃 블록의 복원 샘플들의 평균을 이용하여 현재 블록의 예측값을 획득하는 모드를 나타낸다. INTRA_ANGULAR2 내지 INTRA_ANGULAR34는 각도 예측 모드(angular prediction mode)를 나타내며, 현재 블록 내의 현재 샘플에 대해 특정 각도의 방향에 위치한 이웃 블록의 복원 샘플을 이용하여 현재 샘플의 예측값을 구하는 모드를 나타낸다. 특정 각도의 방향에 실제 샘플이 존재하지 않는 경우 이웃 복원 샘플들에 대해 보간을 수행하여 해당 방향에 대한 가상 샘플을 생성하여 예측값을 구할 수 있다.

[0067] 인트라 예측 모드는 코딩 유닛 별로 구할 수 있지만, 인트라 예측은 변환 유닛 단위로 수행될 수 있다. 따라서, 앞서 설명한 이웃 블록의 복원 샘플은 현재 변환 블록의 이웃 블록 내에 존재하는 복원 샘플을 지칭할 수 있다. 인트라 예측 모드를 이용하여 현재 블록에 대한 예측값을 구한 다음 현재 블록의 샘플값과 예측값 간의 차이를 구할 수 있다. 현재 블록의 샘플값과 예측값 간의 차이를 레지듀얼(또는 레지듀얼 정보 또는 레지듀얼 데이터)이라고 지칭할 수 있다. 디코더 측에서는 현재 블록에 대한 변환 계수 정보를 비트스트림으로부터 획득한 다음, 획득한 변환 계수 정보에 대해 역양자화 및 역변환을 수행하여 레지듀얼을 구할 수 있다. 역양자화는 양자화 파라미터(QP) 정보를 이용하여 변환 계수 정보의 값을 스케일링하는 것을 지칭할 수 있다. 변환 유닛은 변환을 수행하는 기본 단위이므로, 변환 유닛 단위로 변환 계수 정보가 비트스트림을 통해 시그널링될 수 있다.

[0068] 인트라 예측을 수행하는 경우 레지듀얼이 0일 수 있다. 예를 들어, 현재 블록의 샘플과 인트라 예측을 위한 참조 샘플이 동일한 경우 레지듀얼의 값이 0일 수 있다. 현재 블록에 대한 레지듀얼 값이 모두 0일 경우 변환 계수 정보의 값도 모두 0이므로 변환 계수 정보를 비트스트림을 통해 시그널링할 필요가 없다. 따라서, 비트스트림을 통해 해당 블록에 대한 변환 계수 정보가 시그널링되는지 여부를 지시하는 정보를 비트스트림을 통해 시그널링할 수 있다. 해당 변환 블록이 0이 아닌 변환 계수 정보를 가지는지 여부를 지시하는 정보는 코딩 블록 지시 정보(coded block indication information) 또는 코딩 블록 플래그 정보(coded block flag information)라고 지칭하며, 본 명세서에서 cbf로 지칭될 수 있다. 루마 성분에 대한 코딩 블록 지시 정보는 cbf_luma로 지칭될 수 있고, 크로마 성분에 대한 코딩 블록 지시 정보는 cbf_cr 또는 cbf_cb로 지칭될 수 있다. 디코더는 해당 변환 블록에 대한 코딩 블록 지시 정보를 비트스트림으로부터 획득하고, 코딩 블록 지시 정보가 해당 블록이 0이 아닌 변환 계수 정보를 포함함을 지시하는 경우 해당 변환 블록에 대한 변환 계수 정보를 비트스트림으로부터 획득하고 역양자화 및 역변환을 거쳐 레지듀얼을 획득할 수 있다.

[0069] 현재 코딩 블록이 인트라 예측 모드로 코딩되는 경우, 디코더는 변환 블록 단위로 예측값을 구하여 현재 코딩 블록에 대한 예측값을 구하고 및/또는 변환 블록 단위로 레지듀얼을 구하여 현재 코딩 블록에 대한 레지듀얼을 구할 수 있다. 디코더는 현재 코딩 블록에 대한 예측값 및/또는 레지듀얼을 이용하여 현재 코딩 블록을 복원할 수 있다.

[0070] 변환/역변환 기법으로서 이산 코사인 변환(discrete cosine transform, DCT)이 널리 이용되고 있다. DCT를 위한 변환 기저들은 적은 메모리와 빠른 연산을 위해 정수 형태로 근사화될 수 있다. 정수로 근사화된 변환 기저들은 행렬 형태로 표현될 수 있는데 행렬 형태로 표현된 변환 기저들을 변환 행렬이라고 지칭할 수 있다. H.265/HEVC 표준에서는 4×4 내지 32×32 크기의 정수 변환이 사용되며 4×4 또는 32×32 변환 행렬이 제공된다. 4×4 변환 행렬은 4×4 변환/역변환에 이용되고, 32×32 변환 행렬은 8×8, 16×16, 32×32 변환/역변환에 이용될 수 있다.

[0071] 한편, 현재 코딩 블록에 대한 예측 모드 정보가 현재 코딩 블록이 인터 예측을 이용하여 코딩됨을 지시하는 경우, 현재 코딩 블록의 파티셔닝 모드(partitioning mode)를 지시하는 정보가 비트스트림을 통해 시그널링될 수 있다. 현재 코딩 블록의 파티셔닝 모드를 지시하는 정보는 예를 들어 part_mode로 나타낼 수 있다. 현재 코딩 블록이 인터 예측을 이용하여 코딩되는 경우, 현재 코딩 블록의 파티셔닝 모드에 따라 현재 코딩 블록을 적어도 하나의 예측 블록을 분할할 수 있다.

[0072] 예를 들어, 현재 코딩 블록이 2N×2N 블록이라고 가정하면, 파티셔닝 모드는 PART_2Nx2N, PART_2NxN, PART_Nx2N, PART_2NxnU, PART_2NxnD, PART_nLx2N, PART_nRx2N, PART_NxN을 포함할 수 있다. PART_2Nx2N는 현재 코딩 블록과 예측 블록이 동일한 모드를 나타낸다. PART_2NxN는 현재 코딩 블록이 2개의 2N×N 예측 블록으로 분할되는 모드를 나타낸다. PART_Nx2N는 현재 코딩 블록이 2개의 N×2N 예측 블록으로 분할되는 모드를 나타낸다. PART_2NxnU는 현재 코딩 블록이 상측의 2N×n 예측 블록과 하측의 2N×(N-n) 예측 블록으로 분할되는 모

드를 나타낸다. PART_2NxnD는 현재 코딩 블록이 상측의 2N×(N-n) 예측 블록과 하측의 2N×n 예측 블록으로 분할되는 모드를 나타낸다. PART_nLx2N는 현재 코딩 블록이 좌측의 n×2N 예측 블록과 우측의 (N-n)×2N 예측 블록으로 분할되는 모드를 나타낸다. PART_nRx2N는 현재 코딩 블록이 좌측의 (N-n)×2N 예측 블록과 우측의 n×2N 예측 블록으로 분할되는 모드를 나타낸다. PART_NxN은 현재 코딩 블록이 4개의 N×N 예측 블록으로 분할되는 모드를 나타낸다. 예를 들어, n은 N/2이다.

[0073] 현재 코딩 블록이 인트라 코딩 모드인 경우에도 part_mode가 비트스트림을 통해 시그널링될 수 있다. 다만, 현재 코딩 블록이 인트라 코딩 모드이면, 현재 코딩 블록의 크기가 코딩 블록의 최소 크기인 경우에만 part_mode가 시그널링되며 part_mode의 값에 따라 현재 코딩 블록이 4개의 블록으로 추가 분할되는지 여부를 지시할 수 있다.

[0074] 예측 유닛은 움직임 추정 및 움직임 보상을 수행하는 단위이다. 따라서, 예측 유닛 단위로 인터 예측 파라미터 정보가 비트스트림을 통해 시그널링될 수 있다. 인터 예측 파라미터 정보는 예를 들어 참조 픽처 정보, 움직임 벡터 정보를 포함할 수 있다. 인터 예측 파라미터 정보는 이웃 유닛으로부터 유도되거나 비트스트림을 통해 시그널링될 수 있다. 인터 예측 파라미터 정보를 이웃 유닛으로부터 유도하는 경우를 병합 모드(merge mode)라고 지칭한다. 따라서, 현재 예측 유닛에 대한 인터 예측 파라미터 정보가 이웃 유닛으로부터 유도되는지 여부를 지시하는 정보가 비트스트림을 통해 시그널링될 수 있으며, 해당 정보는 병합 지시(merge indication) 정보 또는 병합 플래그 정보라고 지칭될 수 있다. 병합 지시 정보는 예를 들어 merge_flag로 나타낼 수 있다.

[0075] 병합 지시 모드가 현재 예측 유닛의 인터 예측 파라미터 정보가 이웃 유닛으로부터 유도됨을 지시하는 경우, 이웃 유닛을 이용하여 병합 후보 리스트를 구성하고 병합 후보 리스트 중에서 현재 유닛의 인터 예측 파라미터 정보를 유도할 병합 후보를 지시하는 정보가 비트스트림을 통해 시그널링될 수 있으며, 해당 정보는 병합 인덱스 정보라고 지칭될 수 있다. 예를 들어, 병합 인덱스 정보는 merge_idx로 나타낼 수 있다. 이웃 블록은 현재 블록을 포함하는 픽처 내에서 현재 블록과 인접한 좌측 이웃 블록, 상측 이웃 블록, 좌상측 이웃 블록, 좌하측 이웃 블록, 우상측 이웃 블록을 포함하는 공간적 이웃 블록과 현재 블록을 포함하는 픽처와 상이한 픽처 내에서 현재 블록에 대응하는 위치에 위치한(또는 co-located) 시간적 이웃 블록을 포함할 수 있다. 디코더는 상기 이웃 블록들을 이용하여 병합 후보 리스트를 구성하고 병합 인덱스 정보를 비트스트림으로부터 획득하고 병합 후보 리스트 중에서 병합 인덱스 정보가 지시하는 이웃 블록의 인터 예측 파라미터 정보를 현재 블록의 인터 예측 파라미터 정보로 설정할 수 있다.

[0076] 한편, 예측 블록이 코딩 블록에 대응되고 예측 블록에 대해 인터 예측을 수행한 결과 인터 예측 파라미터 정보가 특정 이웃 블록과 동일하고 레지듀얼도 모두 0인 경우, 인터 예측 파라미터 정보 및 변환 계수 정보 등이 비트스트림을 통해 시그널링될 필요가 없다. 이 경우, 코딩 블록에 대한 인터 예측 파라미터 정보는 이웃 블록으로부터 유도하면 되므로 병합 모드가 적용될 수 있다. 따라서, 해당 코딩 블록이 인터 예측을 이용하여 코딩되는 경우 해당 코딩 블록에 대해서는 병합 인덱스 정보만을 비트스트림을 통해 시그널링할 수 있는데, 이러한 모드를 병합 스킵 모드(merge skip mode)라고 지칭한다. 즉, 병합 스킵 모드에서는 병합 인덱스 정보(예, merge_idx)를 제외하고 코딩 블록에 대한 신택스 정보는 시그널링되지 않는다. 다만, 해당 코딩 블록에 대하여 병합 인덱스 정보(예, merge_idx)를 제외하고 더 이상 신택스 정보를 획득할 필요가 없다는 것을 지시하기 위해 스킵 플래그 정보가 비트스트림을 통해 시그널링될 수 있으며, 본 명세서에서 스킵 플래그 정보는 cu_skip_flag라고 지칭될 수 있다. 디코더는 인트라 코딩 모드가 아닌 슬라이스에서는 코딩 블록에 대해 스킵 플래그 정보를 획득하고 스킵 플래그 정보에 따라 병합 스킵 모드에서 코딩 블록을 복원할 수 있다.

[0077] 병합 지시 모드가 현재 예측 블록의 인터 예측 파라미터 정보가 이웃 블록으로부터 유도됨을 지시하지 않는 경우, 현재 예측 블록의 인터 예측 파라미터는 비트스트림을 통해 시그널링될 수 있다. 현재 예측 블록의 L0 예측인지 및/또는 L1 예측인지에 따라 참조 픽처 리스트 0에 대한 참조 픽처 인덱스 정보 및/또는 참조 픽처 리스트 1에 대한 참조 픽처 인덱스 정보가 비트스트림을 통해 시그널링될 수 있다. 움직임 벡터 정보는 움직임 벡터 차이(motion vector difference)를 나타내는 정보와 움직임 벡터 예측값(motion vector predictor)을 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 움직임 벡터 예측값을 나타내는 정보는 이웃 블록들의 움직임 벡터들로 구성되는 움직임 벡터 예측값 후보 리스트 중에서 현재 블록의 움직임 벡터 예측값으로 사용되는 후보를 지시하는 인덱스 정보이며, 움직임 벡터 예측값 지시 정보라고 지칭될 수 있다. 움직임 벡터 예측값 지시 정보는 예를 들어 mvp_l0_flag 또는 mvp_l1_flag으로 나타낼 수 있다. 디코더는 움직임 벡터 예측값 지시 정보에 기초하여 움직임 벡터 예측값을 획득하고 비트스트림으로부터 움직임 벡터 차이에 관련된 정보를 획득하여 움직임 벡터 차이를 구한 다음 움직임 벡터 예측값과 움직임 벡터 차이를 이용하여 현재 블록에 대한 움직임 벡터

정보를 구할 수 있다.

[0078] 현재 코딩 블록이 인터 예측을 이용하여 코딩되는 경우, 인터 예측이 예측 블록 단위로 수행되는 것을 제외하고 변환 블록에 대해서는 동일/유사한 원리가 적용될 수 있다. 따라서, 현재 코딩 블록이 인터 예측을 이용하여 코딩되는 경우, 현재 코딩 블록을 쿼드 트리 방식으로 적어도 하나의 변환 블록으로 분할하고, 분할된 변환 블록 각각에 대하여 코딩 블록 지시 정보(예, cbf_luma, cbf_cb, cbf_cr)에 기초하여 변환 계수 정보를 획득하고 획득된 변환 계수 정보에 대해 역양자화 및 역변환을 수행하여 레지듀얼을 획득할 수 있다.

[0079] 현재 코딩 블록이 인트라 예측 모드로 코딩되는 경우, 디코더는 예측 블록 단위로 예측값을 구하여 현재 코딩 블록에 대한 예측값을 구하고 및/또는 변환 블록 단위로 레지듀얼을 구하여 현재 코딩 블록에 대한 레지듀얼을 구할 수 있다. 디코더는 현재 코딩 블록에 대한 예측값 및/또는 레지듀얼을 이용하여 현재 코딩 블록을 복원할 수 있다.

[0080] 한편, HEVC 표준은 4K 이상의 초고해상도(예, 4096×2160) 영상을 지원한다. 4K 이상의 초고해상도 영상에서는 균일(homogeneous)한 영역이 많아질 수 있다. 특히, 8K(예, 8192×4320) 이상의 해상도를 가지는 영상의 경우 4K 영상에 비해 균일한 영역의 크기는 더욱 커질 수 있다. 또한, 카메라에 입력되는 장면은 그대로 인데 해상도를 증가시킬 경우 기존 해상도에서 CTU가 커버하는 영역의 블록 크기는 증가하는 반면 CTU 크기는 그대로이므로 기존 CTU의 크기(예, 64×64)가 상대적으로 작아지는 효과를 가질 수 있다. 따라서, 기존 HEVC 표준에서 지원하는 CTU 크기(예, 64×64)보다 더 큰 CTU 사이즈를 사용하는 것이 코딩 효율을 더욱 높일 수 있다.

[0081] 예를 들어, 픽처 해상도가 4K에서 8K로 증가하는 경우 4K 영상에서 특정 영역을 나타내는 해상도는 4배로 증가할 수 있다. 따라서, 4K 이상의 초고해상도 영상에서와 같이 128×128 CTU를 가지는 것이 유리할 수 있다. 하지만, HEVC 표준에서는 CTU 사이즈가 예를 들어 64×64로 제한되기 때문에 4개의 CTU로 분할하여 처리될 수 밖에 없다. 앞서 살펴본 바와 같이, CTU마다 다양한 신택스 정보들이 기본적으로 제공되므로(예, 도 3, 5, 6 참조), 특정 영역의 코딩을 위해 4개의 64×64 CTU를 할당할 경우 128×128 CTU를 1개 할당하는 경우에 비해 기본 신택스 정보들을 위한 비트 수가 4배 가량 증가할 수 있으므로 코딩 효율 측면에서 매우 비효율적이다.

[0082] 이러한 기술적 문제를 해결하기 위한 방법으로서 CTU 크기를 확장하는 것을 고려할 수 있다. 예를 들어, CTU 크기를 128×128로 증가시키는 것을 고려할 수 있다. 이 경우, 코딩 효율을 향상시킬 수는 있지만 기존 HEVC 표준과의 호환성을 고려할 때 CTU 모양은 정사각형 방식으로만 확장할 수 있는 한계가 있다. 영상의 특성에 따라서는 CTU가 가로 크기가 더 크거나 세로 크기가 더 큰 직사각형 형태를 가지는 것이 유리할 수도 있다.

[0083] 이를 위한 방법으로서, 직사각형 형태의 CTU(예, 128×64 또는 64×128)를 고려할 수 있지만, 동일 영상 내에서 다른 영역에 대해서는 비효율적일 수 있는 한계가 있다. 구체적으로, 동일한 영상에서 특정 영역은 정사각형 CTU가 유리할 수 있는 반면 다른 영역은 직사각형 CTU가 유리할 수 있다. 또한, CTU는 영상의 기본 처리 단위이므로 CTU가 직사각형 형태를 가질 경우 하나의 픽처는 일정한 크기의 직사각형 블록들로 분할되어 처리된다. 영상에 따라 정사각형 CTU가 효과적인 영역이 많을 수도 있고 직사각형 CTU가 효과적인 영역이 많을 수도 있으므로 기본 처리 단위를 직사각형 형태로 분할하여 얻을 수 있는 효과는 보장될 수 없는 한계가 있다. 또한, 앞서 설명한 바와 같이, 기존 HEVC 표준의 경우 쿼드 트리 방식의 코딩 트리 및 변환 트리를 지원하므로 직사각형 형태의 CTU는 기존 HEVC 표준과의 호환성이 문제될 수 있다.

[0084] 이에, 본 발명에서는 CTU 크기를 확장시키면서도 동일 영상 내에서 CTU 형태를 유연(flexible)하게 할 수 있는 방법을 제안한다.

[0085] CTU 병합 모드(CTU merge mode)

[0086] CTU 크기를 확장함으로써 얻을 수 있는 효과는 CTU마다 기본적으로 제공되는 신택스 정보의 중복을 방지함으로써 비트 개수를 감소시키고 코딩 효율을 향상시키는 것이다. 따라서, 현재 CTU의 신택스 정보를 이웃 CTU로부터 유도하게 할 경우 CTU 크기를 확장하는 것과 동일한 효과를 얻을 수 있다. 또한, 직사각형은 연속된 복수의 정사각형으로 근사화될 수 있으므로 이웃 CTU로부터 신택스 정보를 유도할 때 현재 CTU에 인접한 좌측 이웃 CTU 또는 상측 이웃 CTU를 참조함으로써 직사각형 형태의 CTU를 가지는 것과 동일한 효과를 얻을 수 있다.

[0087] 본 발명에서는 기존 HEVC 표준에서처럼 정사각형 형태의 CTU 및 CTU 크기(예, 16×16, 32×32, 또는 64×64)를 유지하고 픽처 특성에 따라 현재 CTU를 이웃 CTU와 병합(merge)시키는 것을 제안한다. 현재 CTU를 이웃 CTU와 병합시킨다는 것은 현재 CTU의 정보를 비트스트림을 통해 전송하지 않고 이웃 CTU로부터 유도하는 것을 의미할 수 있다. 본 명세서에서, 본 발명에 따라 현재 CTU의 정보를 이웃 CTU로부터 유도하는 방식을 CTU 병합 모드라

고 지칭할 수 있다.

[0088] 본 발명에 따른 CTU 병합 모드에서 이웃 CTU로부터 유도되는 현재 CTU의 정보는 CTU 이하 레벨에서 비트스트림을 통해 시그널링되는 신택스 정보를 포함할 수 있다. 및/또는, 현재 CTU의 정보는 이웃 CTU의 인코딩/디코딩에 사용된 파라미터 정보를 포함할 수 있다. 현재 CTU의 정보를 이웃 CTU로부터 유도한다는 것은 이웃 CTU의 해당 정보를 현재 CTU의 정보로 설정하는 것을 지칭할 수 있다.

[0089] 예를 들어, 도 3을 참조하면, 이웃 CTU로부터 유도되는 정보는 코딩 유닛을 위한 분할 지시 정보(예, split_cu_flag)를 포함할 수 있다. 앞서 설명한 바와 같이, CTU는 분할 지시 정보(예, split_cu_flag)에 기초하여 쿼드 트리 방식으로 코딩 유닛들로 분할된다. 따라서, 이웃 CTU의 코딩 유닛 분할과 관련된 분할 지시 정보로부터 현재 CTU를 위한 분할 지시 정보가 유도되므로, 현재 CTU의 코딩 쿼드 트리는 이웃 CTU의 코딩 쿼드 트리와 동일한 구조를 가질 수 있다. 즉, 본 발명에 따른 CTU 병합 모드를 적용할 경우 현재 CTU는 이웃 CTU와 동일한 개수, 형태, 구조의 코딩 유닛들로 분할될 수 있다.

[0090] 또한, CTU 병합 모드가 적용되는 경우, 현재 CTU에 포함된 코딩 유닛에 대한 정보도 이웃 CTU로부터 유도될 수 있다. 따라서, CTU 병합 모드가 적용되는 현재 CTU의 정보는 코딩 유닛에 대한 정보를 포함할 수 있다.

[0091] 구체적으로, 현재 CTU에 포함된 코딩 유닛에 대한 정보는 이웃 CTU의 대응되는 코딩 유닛으로부터 유도될 수 있다. 예를 들어, 도 5를 참조하면, 이웃 CTU의 대응되는 코딩 유닛으로부터 유도되는 정보는 예측 모드 정보(예, pred_mode_flag), 코딩 유닛의 파티셔닝 모드(예, part_mode), 인트라 예측 모드와 관련된 정보(예, prev_intra_luma_pred_flag, mpm_idx, rem_intra_luma_pred_mode, intra_chroma_pred_mode), 변환 트리의 존재 여부를 지시하는 정보(예, rqt_root_cbf)를 포함할 수 있다. 또한, CTU 병합 모드가 적용되는 경우, 코딩 유닛들에 대한 파티셔닝 모드(예, part_mode)도 이웃 유닛으로부터 유도되므로 코딩 유닛 내에서 예측 유닛들도 이웃 유닛의 예측 유닛들과 동일한 개수와 형태로 분할될 수 있다. 혹은, 현재 CTU의 정보 중에서 인트라 예측 모드와 관련된 정보(예, prev_intra_luma_pred_flag, mpm_idx, rem_intra_luma_pred_mode, intra_chroma_pred_mode) 대신 인트라 예측 모드가 이웃 CTU로부터 유도될 수 있다.

[0092] 또한, CTU 병합 모드가 적용되는 경우, 코딩 유닛 내에서 각 예측 유닛에 대한 정보도 이웃 CTU로부터 유도될 수 있다. 구체적으로, 현재 CTU에 포함된 예측 유닛에 대한 정보는 이웃 CTU의 대응되는 예측 유닛으로부터 유도될 수 있다. 예를 들어, 현재 CTU의 특정 예측 유닛에 대해 병합 모드가 적용되는 경우, 이웃 CTU로부터 유도되는 정보는 해당 예측 유닛에 대한 병합 지시 정보(예, merge_flag), 병합 인덱스 정보(예, merge_idx)를 포함할 수 있다. 또한, 예를 들어, 현재 CTU의 특정 예측 유닛에 대해 병합 모드가 적용되지 않는 경우, 이웃 CTU로부터 유도되는 정보는 인터 예측 파라미터 정보(예, 참조 픽처 인덱스 정보, 움직임 벡터 정보)를 포함할 수 있다.

[0093] 유사하게, CTU 병합 모드가 적용되는 경우, 현재 CTU에서 변환 트리 또는 변환 유닛과 관련된 정보들도 이웃 CTU로부터 유도될 수 있다. 예를 들어, 도 3 및 도 6을 참조하면, 이웃 CTU로부터 유도되는 변환 유닛에 대한 정보는 변환 유닛에 대한 분할 지시 정보(예, split_transform_flag), 코딩 블록 지시 정보(예, cbf, cbf_luma, cbf_cb, cbf_cr), 변환 계수 정보를 포함할 수 있다. 이 경우, 변환 유닛을 위한 분할 지시 정보가 이웃 CTU로부터 유도되므로 현재 CTU의 변환 트리는 이웃 CTU의 변환 트리와 동일한 구조를 가질 수 있다.

[0094] 다른 예로, 이웃하는 CTU들 간에 텍스처가 미세하게 다른 경우에도 CTU 병합 모드를 적용하는 것이 현재 CTU에 대한 비트 개수를 감소시키는 데 유리할 수 있다. 이를 위해, CTU 병합 모드가 적용되더라도 레지듀얼 데이터와 관련되는 변환 계수 정보는 이웃 CTU로부터 유도되지 않고 비트스트림을 통해 시그널링될 수 있다. 이 경우, 디코더는 현재 CTU에서 변환 유닛을 위한 분할 지시 정보(예, split_transform_flag)를 이웃 CTU로부터 유도하여 변환 트리를 형성하고 변환 트리 내에서 각 변환 유닛에 대한 변환 계수 정보는 비트스트림으로부터 획득할 수 있다.

[0095] 변환 계수 정보가 비트스트림으로부터 획득되는 경우, 해당 변환 블록에 대한 코딩 블록 지시 정보(예, cbf) 역시 비트스트림을 통해 시그널링되는 것이 바람직할 수 있다. 따라서, CTU 병합 모드가 적용되더라도 각 변환 블록에 대한 변환 계수 정보 및 코딩 블록 지시 정보는 이웃 CTU로부터 유도되지 않고 비트스트림을 통해 시그널링될 수 있다.

[0096] 또 다른 예로, CTU 병합 모드가 적용되더라도 변환 트리와 관련된 정보(예, split_transform_flag, cbf, cbf_luma, cbf_cb, cbf_cr, 변환 계수 정보)는 이웃 CTU로부터 유도되지 않고 비트스트림을 통해 시그널링할 수 있다. 이 경우, 현재 CTU와 이웃 CTU는 서로 다른 변환 트리 구조를 가질 수 있으므로 이웃하는 CTU들간에 텍스

처가 다소 상이하더라도 CTU 병합 모드를 적용할 수 있다. 따라서, CTU 병합 모드가 적용되는 범위가 확대될 수 있으므로 비트 개수 감소 및 코딩 효율 증가에 유리할 수 있다. 본 예에서, 디코더는 CTU 병합 모드가 적용되는 경우 CTU 및 코딩 유닛과 관련된 정보(예, 도 3의 코딩 트리 및 도 5 참조)를 이웃 CTU로부터 유도하고 변환 트리와 관련된 정보(예, 도 3의 변환 트리 및 도 6 참조)를 비트스트림으로부터 획득할 수 있다.

[0097] 한편, 코딩 유닛을 위한 분할 지시 정보(예, split_cu_flag)에 따라 CTU는 코딩 유닛으로 분할되지 않을 수 있다. 이 경우, CTU는 코딩 유닛과 동일한 크기를 가질 수 있다. 또한, 해당 코딩 유닛이 병합 스킵 모드인 경우, CTU 크기에서 병합 스킵 모드를 수행하여 현재 CTU의 인터 예측 파라미터 정보(예, 참조 픽처 인덱스 정보, 움직임 벡터 정보)를 이웃 CTU로부터 획득하는 것과 유사할 수 있다. 하지만, CTU가 코딩 유닛과 동일한 크기인지 여부는 CTU에 대한 분할 지시 정보(예, split_cu_flag)를 획득함으로써 알 수 있다. 반면, 본 발명에 따른 CTU 병합 모드의 경우 코딩 유닛을 위한 분할 지시 정보(예, split_cu_flag)까지 이웃 CTU로부터 유도되므로 병합 스킵 모드에 비해 비트 개수를 줄일 수 있으며 코딩 효율을 향상시킬 수 있다.

[0098] 또한, 현재 CTU가 인트라 슬라이스에 포함되지 않는 경우에 병합 스킵 모드가 적용될 수 있다. 구체적으로, 코딩 유닛이 인트라 예측을 이용하여 코딩되지 않는 경우(또는 인터 예측을 이용하여 코딩되는 경우)에만 병합 스킵 모드가 적용될 수 있다. 따라서, 병합 스킵 모드는 인터 예측이 이용되는 경우에 제한적으로 적용될 수 있다. 이에 반해, 본 발명에 따른 CTU 병합 모드는 코딩 유닛이 인트라 예측을 이용하여 코딩되더라도 관련 정보를 이웃 CTU로부터 유도하기 때문에 인터 예측이 이용되는 경우에 제한되지 않고 적용될 수 있다.

[0099] CTU 병합 모드에 따른 신택스 정보

[0100] 본 발명에 따른 CTU 병합 모드를 지원하기 위해, 본 발명에서는 CTU 병합 지시 정보와 CTU 병합 인덱스 정보를 비트스트림을 통해 시그널링할 것을 제안한다. 본 발명에 따른 CTU 병합 지시 정보와 CTU 병합 인덱스 정보는 비트스트림 내에서 CTU를 위한 분할 지시 정보(예, split_cu_flag) 이전에 위치할 수 있다. 따라서, 디코더는 CTU를 위한 분할 지시 정보(예, split_cu_flag)를 획득하기 전에 본 발명에 따른 CTU 병합 지시 정보 및/또는 CTU 병합 인덱스 정보를 비트스트림으로부터 획득할 수 있다(예, 도 3 참조).

[0101] CTU 병합 지시 정보는 현재 CTU에 대해 본 발명에 따른 CTU 병합 모드가 적용되는지 여부를 지시하며, 본 명세서에서 ctu_merge_flag라고 지칭될 수 있다. 구체적으로, CTU 병합 지시 정보는 현재 CTU와 관련된 정보가 이웃 CTU로부터 유도되는지 여부를 지시할 수 있다. 예를 들어, CTU 병합 지시 정보가 1의 값을 가지는 경우 현재 CTU와 관련된 정보가 이웃 CTU로부터 유도됨을 지시하고, CTU 병합 지시 정보가 0의 값을 가지는 경우 현재 CTU와 관련된 정보가 이웃 CTU로부터 유도되지 않음을 지시할 수 있다. 다른 예로, CTU 병합 지시 정보가 0의 값을 가지는 경우 현재 CTU와 관련된 정보가 이웃 CTU로부터 유도됨을 지시하고, CTU 병합 지시 정보가 1의 값을 가지는 경우 현재 CTU와 관련된 정보가 이웃 CTU로부터 유도되지 않음을 지시할 수 있다.

[0102] CTU 병합 지시 정보는 현재 CTU 내에서 가장 먼저 시그널링될 수 있다. 예를 들어, 도 3을 참조하면, CTU 병합 지시 정보는 split_cu_flag 전에 시그널링될 수 있다. 따라서, 디코더는 현재 CTU에 대한 CTU 병합 지시 정보(예, ctu_merge_flag)를 비트스트림으로 획득하고, CTU 병합 지시 정보에 기초하여 현재 CTU에 대한 정보를 비트스트림으로부터 획득하거나 이웃 CTU로부터 유도할 수 있다. 구체적으로, CTU 병합 지시 정보가 현재 CTU의 정보가 이웃 CTU로부터 유도되지 않음을 지시하는 경우 도 3 내지 도 6를 참조하여 설명된 절차를 수행하여 현재 CTU를 디코딩할 수 있다. 반면, CTU 병합 지시 정보가 현재 CTU의 정보가 이웃 CTU로부터 유도됨을 지시하는 경우, 디코더는 현재 CTU의 디코딩에 필요한 모든 정보를 해당 이웃 CTU로부터 유도하거나, 혹은 현재 CTU 내에 포함된 코딩 유닛에 대한 정보만을 해당 이웃 CTU로부터 유도하고 변환 트리 이하의 정보는 이웃 CTU와 무관하게 비트스트림으로부터 획득하거나, 혹은 현재 유닛의 변환 계수 정보만을 비트스트림으로부터 획득하고 나머지 현재 CTU의 정보는 이웃 CTU로부터 유도할 수 있다.

[0103] CTU 병합 인덱스 정보는 이웃 CTU들을 포함하는 후보들 중에서 현재 CTU를 병합할 후보 CTU를 지시하며, 본 명세서에서 ctu_merge_from_left_flag 또는 ctu_merge_idx라고 지칭될 수 있다. 구체적으로, CTU 병합 인덱스 정보는 이웃 CTU들을 포함하는 병합 후보들 중에서 특정 병합 후보를 지시할 수 있으며, CTU 병합 인덱스 정보가 지시하는 병합 후보로부터 현재 CTU의 정보가 유도될 수 있다. 본 명세서에서, 현재 CTU의 이웃 CTU들을 포함하는 병합 후보들을 병합 후보 리스트라고 지칭할 수 있고, CTU 병합 인덱스 정보가 지시하는 병합 후보를 병합 대상 CTU라고 지칭할 수 있다.

[0104] 예를 들어, CTU 크기를 직사각형 형태로 확장하기 위해 이웃 CTU들은 현재 CTU와 동일한 픽처 내에서 현재 CTU에 인접한 좌측 이웃 CTU, 상측 이웃 CTU를 포함할 수 있다. 이 경우, CTU 병합 인덱스 정보가 1의 값을 가지는

경우 좌측 이웃 CTU를 지시하고, CTU 병합 인덱스 정보가 0인 경우 상측 이웃 CTU를 지시할 수 있다. 혹은, CTU 병합 인덱스 정보가 0인 경우 좌측 이웃 CTU를 지시하고, CTU 병합 인덱스 정보가 1인 경우 상측 이웃 CTU를 지시할 수 있다.

[0105] 다른 예로, 보다 자유로운 형태로 CTU를 확장하고 코딩 효율 향상을 극대화하기 위해 이웃 CTU들은 현재 CTU와 동일한 픽처 내에서 현재 CTU에 인접한 좌측 이웃 CTU, 상측 이웃 CTU, 좌상측 이웃 CTU, 우상측 이웃 CTU, 현재 CTU를 포함하는 픽처와 상이한 픽처에서 현재 CTU에 대응되는 위치에 있는 CTU를 포함할 수 있다. 이 경우, CTU 병합 인덱스 정보는 상기 이웃 CTU들로 구성된 병합 후보 리스트 중에서 특정 이웃 CTU 또는 병합 후보에 대응되는 인덱스 값을 가질 수 있다. 예를 들어, CTU 병합 인덱스 정보는 0 내지 4의 값을 가질 수 있다.

[0106] 도 7은 본 발명에 따른 CTU 병합 모드가 적용되는 예를 예시한다. 도 7의 예에서, 픽처는 동일한 크기의 CTU들을 포함할 수 있다. 도 7의 예에서, 하나의 픽처가 5×5=25개의 CTU들로 구성되는 것으로 도시하였지만, 본 발명은 이에 제한되지 않으며 임의의 크기의 픽처에 대해서도 동일/유사하게 적용될 수 있다.

[0107] 도 7을 참조하면, 화살표는 병합 대상 CTU를 나타낸다. 가로 방향 화살표는 현재 CTU의 정보가 좌측 이웃 CTU로부터 유도됨을 나타내고, 세로 방향 화살표는 현재 CTU의 정보가 상측 이웃 CTU로부터 유도됨을 나타낸다. 도 7에 예시된 바와 같이, 본 발명에 따른 CTU 병합 모드를 적용할 경우, 가로로 긴 직사각형 형태의 CTU 혹은 세로로 긴 직사각형 형태의 CTU 혹은 L자 또는 지그재그 형태의 다각형 형태를 지원하는 것과 동일한 효과를 가질 수 있다. 따라서, 본 발명에 따른 CTU 병합 모드를 적용할 경우, 동일 영상 내에서 유연한 형태의 CTU를 함께 지원하는 것과 동일한 효과를 가질 수 있다.

[0108] 도 8은 본 발명에 따른 CTU 병합 모드와 크기가 확장된 CTU를 비교한 것이다. 도 8(a)는 본 발명에 따른 CTU 병합 모드를 예시하고, 도 8(b)는 크기가 확장된 CTU를 예시한다.

[0109] 도 8(a)를 참조하면, 좌상측 CTU의 경우 CTU 정보가 비트스트림을 통해 시그널링되지만, 우상측 CTU의 정보는 좌상측 CTU로부터 유도되고, 좌하측 CTU의 정보는 좌상측 CTU로부터 유도되고, 우하측 CTU의 정보는 좌하측 CTU로부터 유도될 수 있다. 따라서, 3개의 64×64 CTU에 대한 정보를 비트스트림을 통해 시그널링하지 않더라도 좌상측 CTU로 유도될 수 있으므로 128×128에 대해 하나의 CTU 정보를 비트스트림을 통해 시그널링한 것과 동일한 효과를 가질 수 있다. CTU 병합 모드를 이용할 경우 128×128 CTU를 사용한 것과 동일한 효과를 가질 수 있다.

[0110] 도 8(b)를 참조하면, 128×128 CTU가 사용되므로 전송 정보 측면에서 더 효율적일 수 있지만 영상 특성과 상관없이 항상 정사각형 형태의 CTU로만 제한된다는 단점이 있다.

[0111] 한편, 본 발명에 따른 CTU 병합 모드는 영상 특성이 균일한 영역에 대해 적용될 가능성이 높다. 또한, 영상 특성이 균일한 경우 CTU는 코딩 유닛들로 분할되지 않을 가능성이 높다. 따라서, 본 발명에 따른 CTU 병합 모드는 이웃 CTU들 중 적어도 하나가 코딩 유닛들로 분할되지 않는 경우에 적용하는 것을 고려할 수 있다. 현재 CTU의 이웃 CTU들 중 적어도 하나가 코딩 유닛들로 분할되지 않는 경우, CTU 병합 지시 정보(예, ctu_merge_flag)를 비트스트림을 통해 시그널링할 수 있다. 현재 CTU의 이웃 CTU들 모두가 코딩 유닛들로 분할되는 경우, CTU 병합 지시 정보(예, ctu_merge_flag)는 비트스트림을 통해 시그널링되지 않고, CTU 병합 모드는 현재 CTU를 적용하지 않는다. 따라서, 현재 CTU의 이웃 CTU들 모두가 코딩 유닛들로 분할되는 경우, 현재 CTU는 도 3 내지 도 6을 참조하여 설명된 절차에 따라 디코딩될 수 있다.

[0112] 이웃 CTU의 분할 여부에 따라 CTU 병합 모드의 적용 여부를 결정함으로써 CTU 병합 모드가 적용되지 않는 경우 CTU 병합 모드를 지시하는데 필요한 비트 수를 절감할 수 있으며, 코딩 효율을 향상시킬 수 있다.

[0113] 도 9는 본 발명에 따른 CTU 병합 모드의 실시예를 예시한다. 도 9에 예시된 절차는 CTU를 위한 분할 지시 정보(예, split_cu_flag)를 획득하기 전에 수행될 수 있다(예, 도 3 참조).

[0114] 도 9를 참조하면, 예를 들어, 이웃 CTU는 현재 CTU를 포함하는 픽처 내에서 현재 CTU와 인접한 좌측 이웃 CTU 및 상측 이웃 CTU를 포함할 수 있다. 이 경우, 좌측 이웃 CTU 또는 상측 이웃 CTU가 복수의 코딩 유닛으로 분할되지 않는 경우 CTU 병합 지시 정보(예, ctu_merge_flag)가 비트스트림을 통해 시그널링될 수 있다. 따라서, 디코더는 좌측 이웃 CTU 또는 상측 이웃 CTU가 복수의 코딩 유닛으로 분할되지 않는 경우 현재 CTU에 대한 CTU 병합 지시 정보(예, ctu_merge_flag)가 비트스트림으로부터 획득할 수 있다.

[0115] 반대로, 좌측 이웃 CTU 및 상측 이웃 CTU 모두가 복수의 코딩 유닛으로 분할되는 경우 CTU 병합 지시 정보(예, ctu_merge_flag)는 비트스트림을 통해 시그널링되지 않는다. 따라서, 좌측 이웃 CTU 및 상측 이웃 CTU 모두가 복수의 코딩 유닛으로 분할되는 경우 디코더는 CTU 병합 지시 정보(예, ctu_merge_flag)를 비트스트림으로부터

획득함이 없이 도 3, 5, 6에 예시된 절차대로 현재 CTU를 디코딩할 수 있다.

[0116] 또한, 현재 CTU와 병합할 이웃 CTU를 선정할 때, 좌측 이웃 CTU가 복수의 코딩 유닛으로 분할되지 않는 경우, 병합 대상 CTU는 상측 이웃 CTU로 결정되고, 상측 이웃 CTU로부터 현재 CTU의 정보를 유도할 수 있다. 및/또는, 상측 이웃 CTU가 복수의 코딩 유닛으로 분할되지 않는 경우, 병합 대상 CTU는 좌측 이웃 CTU로 결정되고, 좌측 이웃 CTU로부터 현재 CTU의 정보를 유도할 수 있다. 및/또는, 좌측 이웃 CTU 및 우측 이웃 CTU가 모두 복수의 코딩 유닛으로 분할되지 않는 경우, CTU 병합 인덱스 정보(예, ctu_merge_from_left_flag 또는 ctu_merge_idx)가 비트스트림을 통해 시그널링될 수 있다. 따라서, 좌측 이웃 CTU 및 우측 이웃 CTU가 모두 복수의 코딩 유닛으로 분할되지 않는 경우에만, CTU 병합 인덱스 정보를 시그널링함으로써 CTU 병합 인덱스 정보를 시그널링하는데 필요한 비트 개수를 절감할 수 있으며, 코딩 효율을 향상시킬 수 있다.

[0117] 다른 예로, 이웃 CTU는 현재 CTU를 포함하는 픽처 내에서 현재 CTU와 인접한 좌측 이웃 CTU, 상측 이웃 CTU, 좌상측 이웃 CTU, 우상측 이웃 CTU를 포함하는 공간적 이웃 CTU와, 현재 CTU를 포함하는 픽처와 상이한 픽처 내에서 현재 CTU에 대응하는 위치에 위치한(또는 co-located) CTU를 포함할 수 있다. 이 경우, 이웃 CTU 중 적어도 하나가 복수의 코딩 유닛으로 분할되지 않는 경우 CTU 병합 지시 정보(예, ctu_merge_flag)가 비트스트림을 통해 시그널링될 수 있다. 또한, 이웃 CTU 중 하나의 CTU만이 복수의 코딩 유닛으로 분할되지 않는 경우, CTU 병합 인덱스 정보(예, ctu_merge_from_left_flag 또는 ctu_merge_idx)를 비트스트림을 통해 시그널링하지 않고 분할되지 않는 CTU로부터 현재 블록의 정보를 유도할 수 있다. 만일 이웃 CTU 중 둘 이상의 CTU가 복수의 코딩 유닛으로 분할되지 않는 경우, CTU 병합 인덱스 정보(예, ctu_merge_from_left_flag 또는 ctu_merge_idx)가 비트스트림을 통해 시그널링될 수 있으며, CTU 병합 인덱스 정보가 지시하는 이웃 CTU로부터 현재 CTU의 정보를 유도할 수 있다.

[0118] 도 10은 본 발명에 따른 방법의 순서도를 예시한다.

[0119] 도 10에 예시된 방법은 적어도 하나의 픽처(또는 코딩된 픽처)를 포함하는 비트스트림에 대해 수행될 수 있다. 하나의 픽처는 적어도 하나의 슬라이스를 포함할 수 있고, 각각의 슬라이스는 일련의 CTU들을 포함할 수 있다. 따라서, 하나의 픽처는 동일한 크기의 CTU로 분할될 수 있고 CTU 단위로 인코딩/디코딩을 수행하여 픽처를 처리할 수 있다. 앞서 설명된 바와 같이, CTU의 크기는 시퀀스 파라미터 세트(SPS)로부터 구할 수 있으며, SPS를 이용하여 결정된 CTU의 크기에 따라 픽처를 동일한 크기의 CTU들로 분할한 다음 각각의 CTU에 대해 도 10에 예시된 방법을 적용할 수 있다.

[0120] 설명의 편의를 위해, 현재 CTU는 현재 블록이라고 지칭될 수 있고, CTU 병합 지시 정보(예, ctu_merge_flag)는 플래그 정보라고 지칭될 수 있고, CTU 병합 인덱스 정보(예, ctu_merge_from_left_flag 또는 ctu_merge_idx)는 인덱스 정보라고 지칭될 수 있다.

[0121] S1002 단계에서, 현재 CTU에 대해 CTU 병합 모드가 적용되는지 여부를 판별할 수 있다. 일 예로, 디코더는 CTU 병합 모드가 적용되는지 여부를 지시하는 CTU 병합 지시 정보(예, ctu_merge_flag)를 비트스트림으로부터 획득하고, CTU 병합 지시 정보에 기초하여 CTU 병합 모드가 적용되는지 판별할 수 있다. 예를 들어, CTU 병합 지시 정보가 1의 값을 가지는 경우 CTU 병합 모드가 적용됨을 지시할 수 있고, CTU 병합 지시 정보가 0의 값을 가지는 경우 CTU 병합 모드가 적용되지 않음을 지시할 수 있다. CTU 병합 지시 정보의 값은 서로 뒤바뀔 수 있다.

[0122] 다른 예로, S1002 단계에서, 현재 CTU를 포함하는 픽처 내에서 현재 CTU와 인접하는 적어도 하나의 이웃 CTU가 복수의 CU로 분할되는지 여부에 기초하여 CTU 병합 모드가 적용되는지를 판별할 수 있다. 보다 구체적으로, S1002 단계에서, 도 9를 참조하여 설명된 절차를 수행할 수 있다.

[0123] S1004 단계에서, 현재 CTU에 대해 CTU 병합 모드가 적용되는 경우 S1006 단계로 진행할 수 있고, 현재 CTU에 대해 CTU 병합 모드가 적용되지 않는 경우 S1012 단계로 진행할 수 있다.

[0124] S1006 단계에서, 현재 CTU에 대한 병합 대상 CTU를 결정할 수 있다. 본 명세서에서, 현재 CTU의 정보를 유도할 CTU 또는 CTU 병합 인덱스 정보가 지시하는 CTU를 병합 대상 CTU라고 지칭할 수 있다. 일 예로, 디코더는 이웃 CTU들을 포함하는 병합 후보 리스트를 구성할 수 있다. 이웃 CTU들은 현재 CTU를 포함하는 픽처 내에서 현재 CTU와 인접한 좌측 이웃 CTU 및 상측 이웃 CTU를 포함할 수 있다. 혹은, 이웃 CTU들은 현재 CTU를 포함하는 픽처 내에서 현재 CTU를 포함하는 픽처 내에서 현재 CTU에 인접한 좌측 이웃 CTU, 상측 이웃 CTU, 좌상측 이웃 CTU, 우상측 이웃 CTU, 현재 CTU를 포함하는 픽처와 상이한 픽처에서 현재 CTU에 대응되는 위치에 있는 CTU를 포함할 수 있다.

[0125] 디코더는 비트스트림으로부터 CTU 병합 인덱스 정보를 획득하고, 병합 후보 리스트 중에서 CTU 병합 인덱스 정

보가 지시하는 CTU를 병합 대상 CTU로서 결정할 수 있다.

[0126] 혹은, 도 9를 참조하여 설명한 바와 같이, 이웃 CTU가 복수의 코딩 유닛으로 분할되는지 여부에 따라 병합 대상 CTU를 결정할 수 있다. 구체적인 예로, 좌측 이웃 CTU가 복수의 CU로 분할되는 경우, 병합 대상 CTU는 우측 이웃 CTU로 결정되고, CTU 병합 인덱스 정보는 비트스트림을 통해 시그널링되지 않는다. 우측 이웃 CTU가 복수의 CU로 분할되는 경우, 병합 대상 CTU는 좌측 이웃 CTU로 결정되고, CTU 병합 인덱스 정보는 비트스트림을 통해 시그널링되지 않는다. 좌측 이웃 CTU 및 우측 이웃 CTU가 분할되지 않는 경우, 비트스트림으로부터 CTU 병합 인덱스 정보를 획득하고 CTU 병합 인덱스 정보가 지시하는 병합 후보를 병합 대상 CTU로 결정할 수 있다.

[0127] S1008 단계에서, 병합 대상 CTU로부터 현재 CTU의 정보를 유도할 수 있다. 예를 들어, 현재 CTU의 정보를 유도하는 것은 병합 대상 CTU의 정보를 현재 CTU의 정보로 설정하는 것을 포함할 수 있다. 병합 대상 CTU로부터 유도되는 정보는 도 3, 5, 6을 참조하여 설명한 정보를 포함할 수 있다. 예를 들어, 현재 CTU의 모든 정보가 병합 대상 CTU로부터 유도될 수 있다. 다른 예로, 현재 CTU 중에서 CTU 및 CU에 대한 정보가 병합 대상 CTU로부터 유도될 수 있다. 이 경우, 도 3 및 도 5에 예시된 정보들이 병합 대상 CTU로부터 유도될 수 있다. 또 다른 예로, 현재 CTU 중에서 변환 계수 정보를 제외한 나머지 정보가 병합 대상 CTU로부터 유도될 수 있다.

[0128] 만일 CTU 병합 모드가 적용되지 않는 경우, S1010 단계에서, 현재 CTU에 대한 정보는 비트스트림으로부터 획득될 수 있다. 구체적으로, 디코더는 CTU 병합 모드를 적용하지 않고, 도 3, 5, 6을 참조하여 설명된 절차를 수행할 수 있다.

[0129] S1012 단계에서, 디코더는 현재 CTU에 대한 정보를 기반으로 현재 CTU를 복원할 수 있다. 앞서 설명한 바와 같이, 인트라 예측이 적용되는 경우, 인트라 예측 모드에 기초하여 각 변환 블록에 대한 예측값을 획득하고, 각 변환 블록에 대한 레지듀얼을 획득한 다음, 현재 CTU를 복원할 수 있다. 또한, 인터 예측이 적용되는 경우, 인터 예측 파라미터 정보에 기초하여 각 예측 블록에 대한 예측값을 획득하고, 각 변환 블록에 대한 레지듀얼을 획득한 다음, 현재 CTU를 복원할 수 있다.

[0130] 앞서 설명한 바와 같이, CU는 인트라 예측 모드 또는 인터 예측 모드를 결정하는 단위이며, CU 내에서는 동일한 인트라 예측 모드 또는 인터 예측 모드가 적용될 수 있다. PU는 인터 예측을 수행하는 단위이며, PU 내에서는 동일한 인터 예측 파라미터 정보(예, 움직임 벡터 정보, 참조 픽처 인덱스 정보)가 적용될 수 있다. TU는 변환을 수행하는 단위이며, TU 단위로 변환 계수 정보를 획득하고, 역양자화/역변환이 수행될 수 있다.

[0131] 도 11은 본 발명이 적용될 수 있는 영상 처리 장치의 블록도를 예시한다. 영상 처리 장치는 영상 신호의 인코딩 장치 및/또는 디코딩 장치를 포함할 수 있다. 예를 들어, 본 발명이 적용될 수 있는 영상 처리 장치는 스마트폰 등과 같은 이동 단말, 랩톱 컴퓨터 등과 같은 휴대용 기기, 디지털 TV, 디지털 비디오 플레이어 등과 같은 가전 제품 등을 포함할 수 있다.

[0132] 메모리(12)는 프로세서(11)의 처리 및 제어를 위한 프로그램을 저장할 수 있고, 부호화된 비트스트림, 복호화된 영상, 제어 정보 등을 저장할 수 있다. 또한, 메모리(12)는 각종 영상 신호를 위한 버퍼로서 활용될 수 있다. 메모리(12)는 ROM(Read Only Memory), RAM(Random Access Memory), EPROM(Erasable Programmable Read Only Memory), EEPROM(Electrically Erasable Programmable Read-Only Memory), 플래쉬(flash) 메모리, SRAM(Static RAM), HDD(Hard Disk Drive), SSD(Solid State Drive) 등과 같은 저장 장치로서 구현될 수 있다.

[0133] 프로세서(11)는 영상 처리 장치 내 각 모듈의 동작을 제어한다. 특히, 프로세서(11)는 본 발명에 따른 인코딩/디코딩을 수행하기 위한 각종 제어 기능을 수행할 수 있다. 프로세서(11)는 컨트롤러(controller), 마이크로 컨트롤러(microcontroller), 마이크로 프로세서(microprocessor), 마이크로 컴퓨터(microcomputer) 등으로도 불릴 수 있다. 프로세서(11)는 하드웨어(hardware) 또는 펌웨어(firmware), 소프트웨어, 또는 이들의 결합에 의해 구현될 수 있다. 하드웨어를 이용하여 본 발명을 구현하는 경우에는, 본 발명을 수행하도록 구성된 ASIC(application specific integrated circuit) 또는 DSP(digital signal processor), DSPD(digital signal processing device), PLD(programmable logic device), FPGA(field programmable gate array) 등이 프로세서(11)에 구비될 수 있다. 한편, 펌웨어나 소프트웨어를 이용하여 본 발명을 구현하는 경우에는 본 발명의 기능 또는 동작들을 수행하는 모듈, 절차 또는 함수 등을 포함하도록 펌웨어나 소프트웨어가 구성될 수 있으며, 본 발명을 수행할 수 있도록 구성된 펌웨어 또는 소프트웨어는 프로세서(11) 내에 구비되거나 메모리(12)에 저장되어 프로세서(11)에 의해 구동될 수 있다.

[0134] 또한, 장치(10)는 네트워크 인터페이스 모듈(network interface module, NIM)(13)을 선택적으로(optionally) 포함할 수 있다. 네트워크 인터페이스 모듈(13)은 프로세서(11)와 동작시 연결(operatively connected)되며, 프

로세서(11)는 네트워크 인터페이스 모듈(13)을 제어하여 무선/유선 네트워크를 통해 정보 및/또는 데이터, 신호, 메시지 등을 나르는 무선/유선 신호를 전송 또는 수신할 수 있다. 네트워크 인터페이스 모듈(13)은 예를 들어 IEEE 802 계열, 3GPP LTE(-A), Wi-Fi, ATSC(Advanced Television System Committee), DVB(Digital Video Broadcasting) 등과 같은 다양한 통신 규격을 지원하며, 해당 통신 규격에 따라 제어 정보 및/또는 부호화된 비트스트림과 같은 영상 신호를 송수신할 수 있다. 네트워크 인터페이스 모듈(13)은 필요에 따라 장치에 포함되지 않을 수 있다.

[0135] 또한, 장치(10)는 입출력 인터페이스(14)를 선택적으로(optionally) 포함할 수 있다. 입출력 인터페이스(14)는 프로세서(11)와 동작시 연결(operatively connected)되며, 프로세서(11)는 입출력 인터페이스(14)를 제어하여 제어 신호 및/또는 데이터 신호를 입력받거나 출력할 수 있다. 입출력 모듈(14)은 예를 들어 키보드, 마우스, 터치패드, 카메라 등과 같은 입력 장치와 디스플레이 등과 같은 출력 장치와 연결될 수 있도록 USB(Universal Serial Bus), Bluetooth, NFC(Near Field Communication), 직렬/병렬 인터페이스, DVI(Digital Visual Interface), HDMI(High Definition Multimedia Interface) 등과 같은 규격을 지원할 수 있다.

[0136] 이상에서 설명된 방법들 및 실시예들은 본 발명의 구성요소들과 특징들이 소정 형태로 결합된 것들이다. 각 구성요소 또는 특징은 별도의 명시적 언급이 없는 한 선택적인 것으로 고려되어야 한다. 각 구성요소 또는 특징은 다른 구성요소나 특징과 결합되지 않은 형태로 실시될 수 있다. 또한, 일부 구성요소들 및/또는 특징들을 결합하여 본 발명의 실시예를 구성하는 것도 가능하다. 본 발명의 실시예들에서 설명되는 동작들의 순서는 변경될 수 있다. 어느 실시예의 일부 구성이나 특징은 다른 실시예에 포함될 수 있고, 또는 다른 실시예의 대응하는 구성 또는 특징과 교체될 수 있다. 특허청구범위에서 명시적인 인용 관계가 있지 않은 청구항들을 결합하여 실시예를 구성하거나 출원 후의 보정에 의해 새로운 청구항으로 포함시킬 수 있음은 자명하다.

[0137] 본 발명에 따른 방법 및 실시예는 다양한 수단, 예를 들어, 하드웨어, 펌웨어(firmware), 소프트웨어 또는 그것들의 결합 등에 의해 구현될 수 있다. 하드웨어에 의한 구현의 경우, 본 발명의 일 실시예는 하나 또는 그 이상의 ASIC(application specific integrated circuit), DSP(digital signal processor), DSPD(digital signal processing device), PLD(programmable logic device), FPGA(field programmable gate array), 프로세서, 콘트롤러, 마이크로 콘트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다.

[0138] 펌웨어나 소프트웨어에 의한 구현의 경우, 본 발명은 이상에서 설명된 기능 또는 동작들을 수행하는 모듈, 절차, 함수 등의 형태를 포함하는 소프트웨어 코드 또는 명령어(instruction)로 구현될 수 있다. 소프트웨어 코드 또는 명령어는 컴퓨터 판독가능한 매체에 저장되어 프로세서에 의해 구동될 수 있으며 프로세서에 의해 구동될 때 본 발명에 따른 동작들을 수행할 수 있다. 상기 컴퓨터 판독가능한 매체는 상기 프로세서 내부 또는 외부에 위치하거나 원격으로 네트워크를 통해 상기 프로세서와 연결될 수 있으며, 상기 프로세서와 데이터를 주고받을 수 있다.

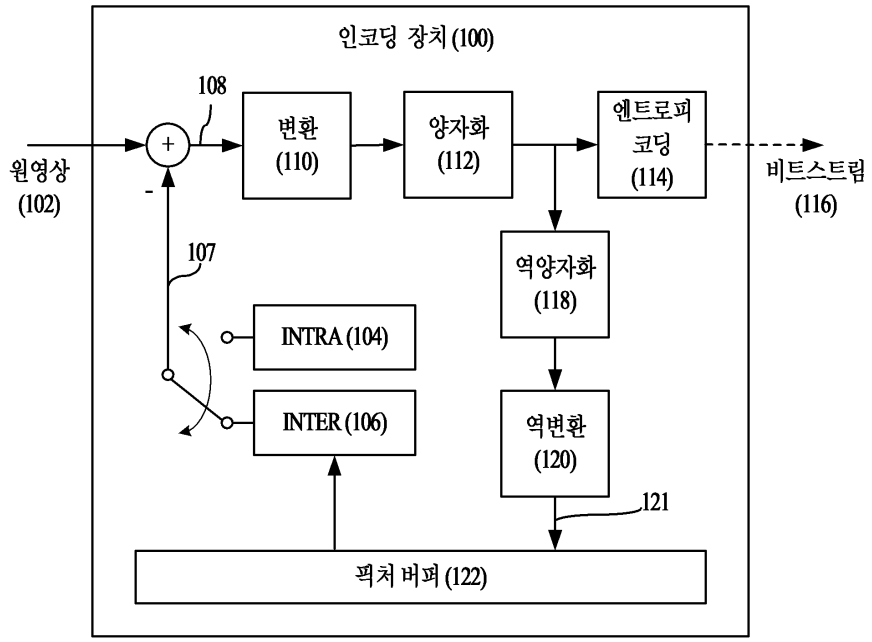
[0139] 본 발명은 본 발명의 특징을 벗어나지 않는 범위에서 다른 특정한 형태로 구체화될 수 있음은 당업자에게 자명하다. 따라서, 상기의 상세한 설명은 모든 면에서 제한적으로 해석되어서는 아니되고 예시적인 것으로 고려되어야 한다. 본 발명의 범위는 첨부된 청구항의 합리적 해석에 의해 결정되어야 하고, 본 발명의 등가적 범위 내에서의 모든 변경은 본 발명의 범위에 포함된다.

산업상 이용가능성

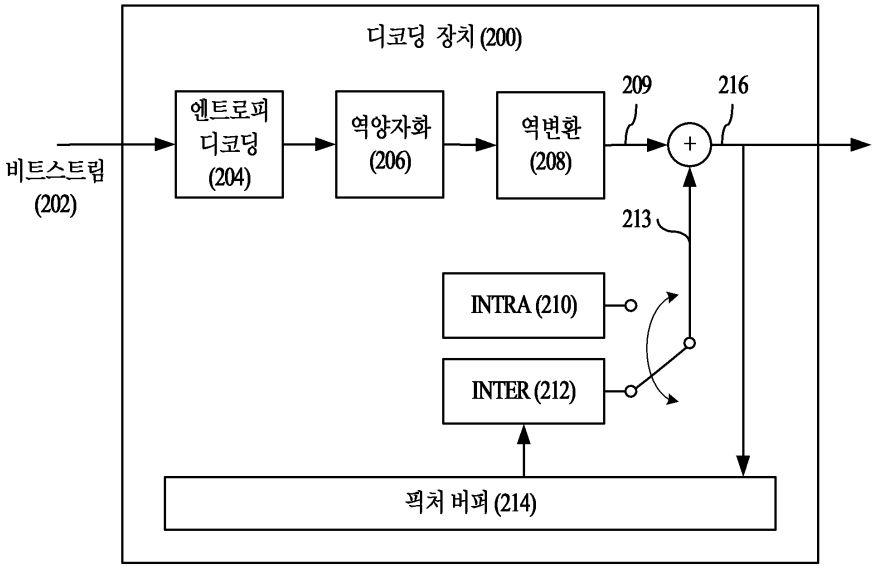
[0140] 본 발명은 디코딩 장치, 인코딩 장치와 같은 영상 처리 장치에 이용될 수 있다.

도면

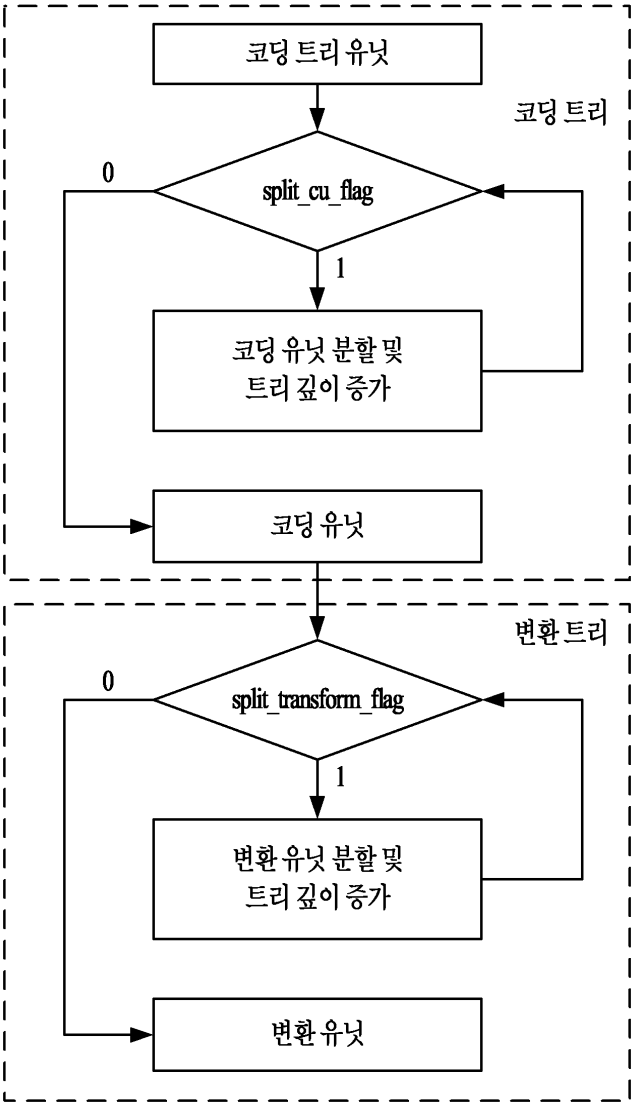
도면1


인코딩 장치 (100)
108
변환 (110)
양자화 (112)
엔트로피 코딩 (114)
원영상 (102)
비트스트림 (116)
107
역양자화 (118)
INTRA (104)
INTER (106)
역변환 (120)
121
픽처 버퍼 (122)

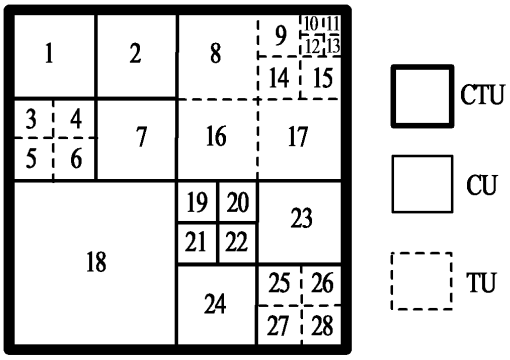
도면2

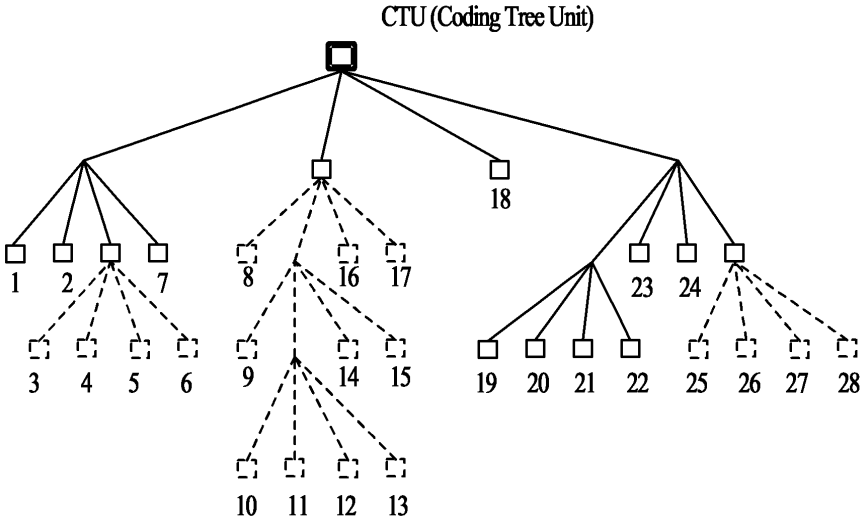

디코딩 장치 (200)
209
216
엔트로피 디코딩 (204)
역양자화 (206)
역변환 (208)
비트스트림 (202)
213
INTRA (210)
INTER (212)
픽처 버퍼 (214)

도면3

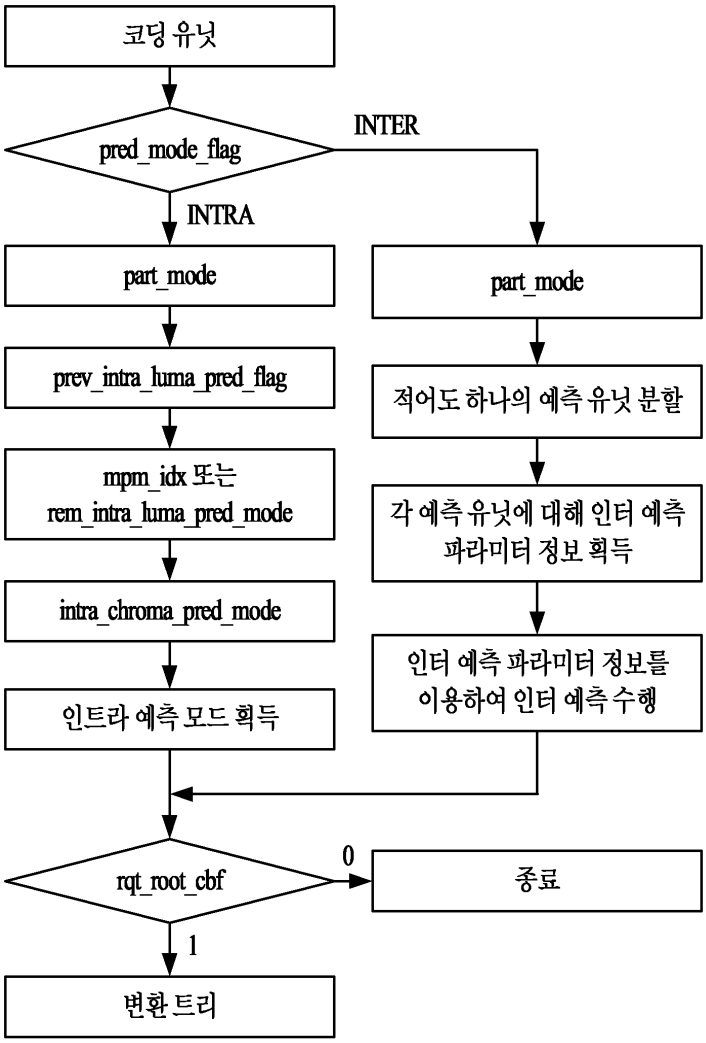

코딩 트리 유닛
코딩 트리
0
split_cu_flag
1
코딩 유닛 분할 및
트리 깊이 증가
코딩 유닛
변환 트리
0
split_transform_flag
1
변환 유닛 분할 및
트리 깊이 증가
변환 유닛

도면4

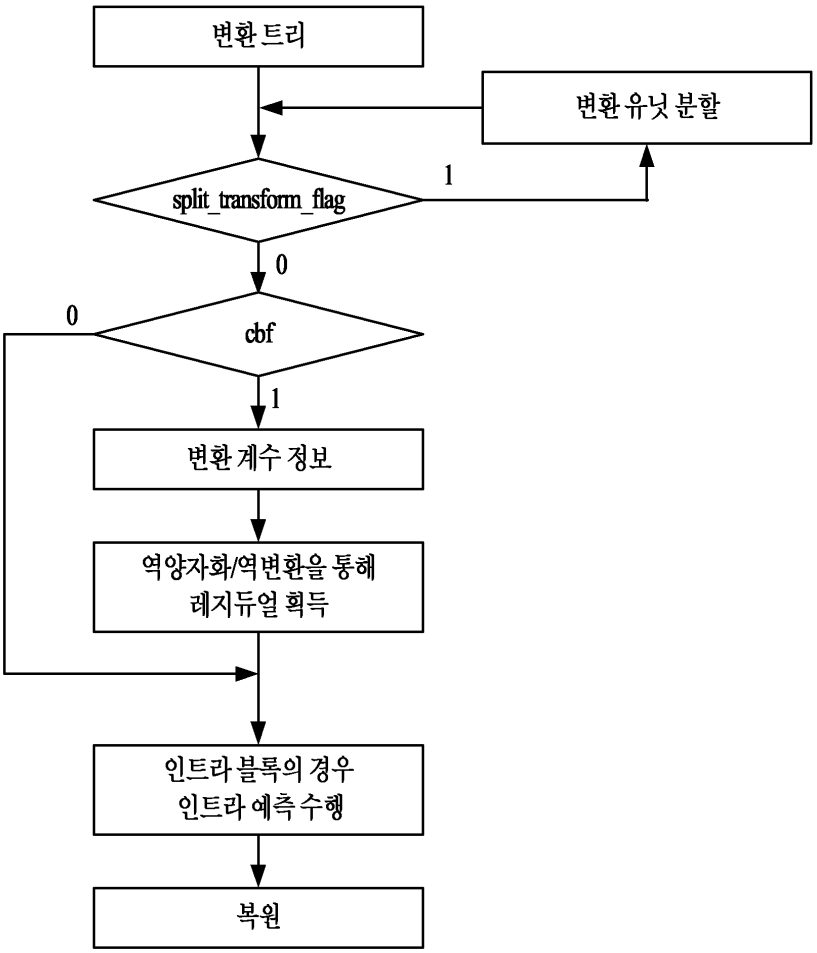

1
2
8
9
10
11
12
13
14
15
3
4
5
6
7
16
17
19
20
21
22
23
18
24
25
26
27
28
CTU
CU
TU


CTU (Coding Tree Unit)
1
2
7
8
16
17
18
23
24
3
4
5
6
9
14
15
19
20
21
22
25
26
27
28
10
11
12
13

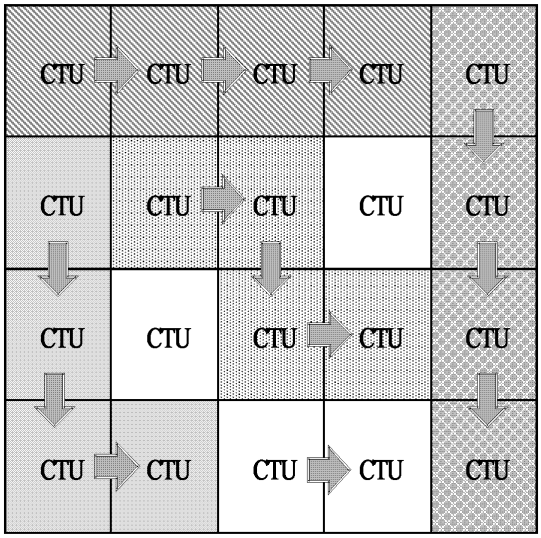
도면5


코딩 유닛
pred_mode_flag
INTER
INTRA
part_mode
prev_intra_luma_pred_flag
mpm_idx 또는
rem_intra_luma_pred_mode
intra_chroma_pred_mode
인트라 예측 모드 획득
part_mode
적어도 하나의 예측 유닛 분할
각 예측 유닛에 대해 인터 예측
파라미터 정보 획득
인터 예측 파라미터 정보를
이용하여 인터 예측 수행
rqt_root_cbf
0
종료
1
변환 트리

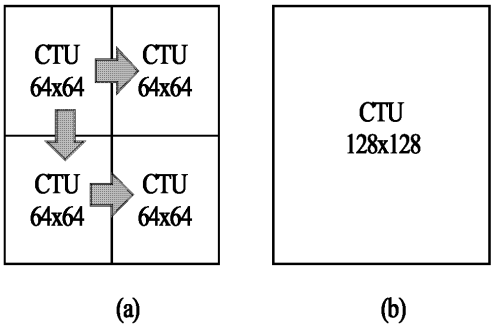
도면6


변환 트리
변환 유닛 분할
split_transform_flag
1
0
0
cbf
1
변환 계수 정보
역양자화/역변환을 통해
레지듀얼 획득
인트라 블록의 경우
인트라 예측 수행
복원

도면7


CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU

도면8

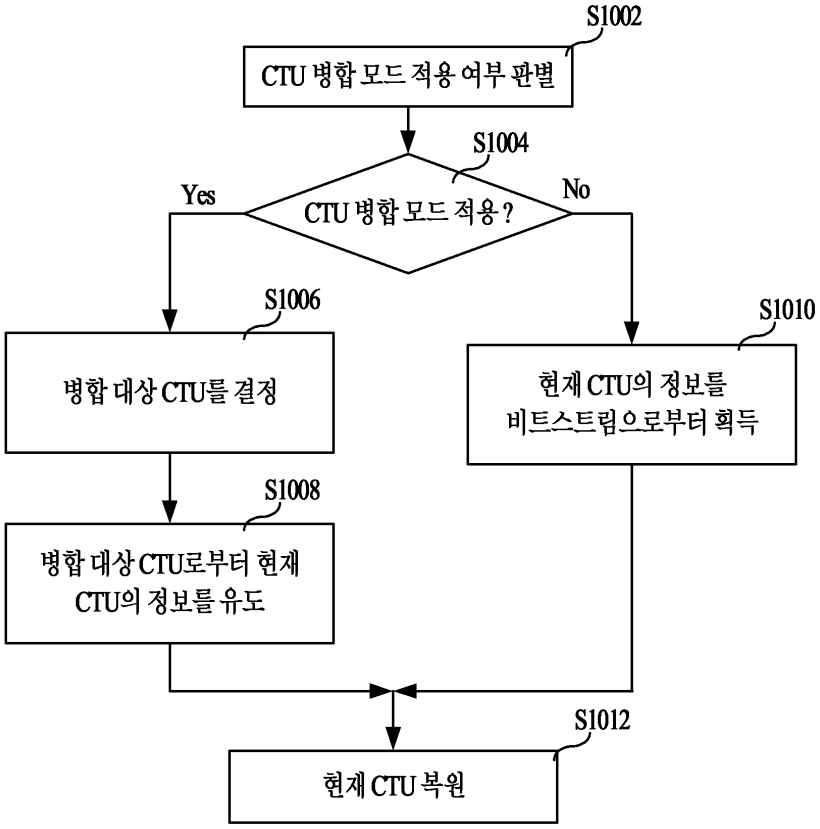


도면9

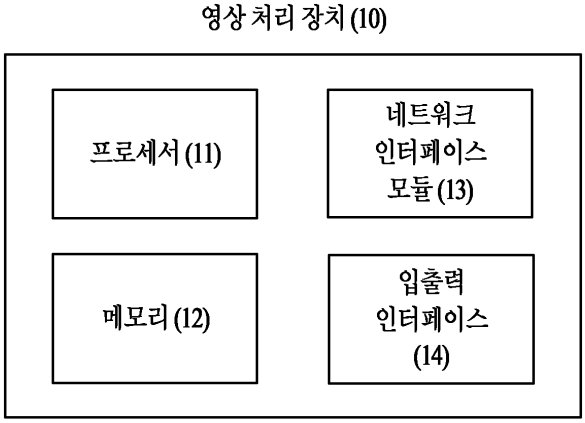
LeftCtuNotSplitFlag = 1 if the left neighboring CTU is NOT split.
AboveCtuNotSplitFlag = 1 if the left neighboring CTU is NOT split.
CtuMergeDir = left or above, determined by the following pseudo code.

```
if( LeftCtuNotSplitFlag || AboveCtuNotSplitFlag ) {
    ctu_merge_flag
    if( !LeftCtuNotSplitFlag ) //left is split but not above
        CtuMergeDir = Above
    if( !AboveCtuNotSplitFlag ) //above is split but not left
        CtuMergeDir = Left
    if( LeftCtuNotSplitFlag && AboveCtuNotSplitFlag ) {//both are not split
        ctu_merge_from_left_flag
        CtuMergeDir = (ctu_merge_from_left_flag==1) ? Left : Above
    }
}
```

도면10


S1002
CTU 병합 모드 적용 여부 판별
S1004
CTU 병합 모드 적용 ?
Yes
No
S1006
병합 대상 CTU를 결정
S1008
병합 대상 CTU로부터 현재 CTU의 정보를 유도
S1010
현재 CTU의 정보를 비트스트림으로부터 획득
S1012
현재 CTU 복원

도면11


영상 처리 장치 (10)
프로세서 (11)
네트워크 인터페이스 모듈 (13)
메모리 (12)
입출력 인터페이스 (14)