



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0034644
(43) 공개일자 2020년03월31일

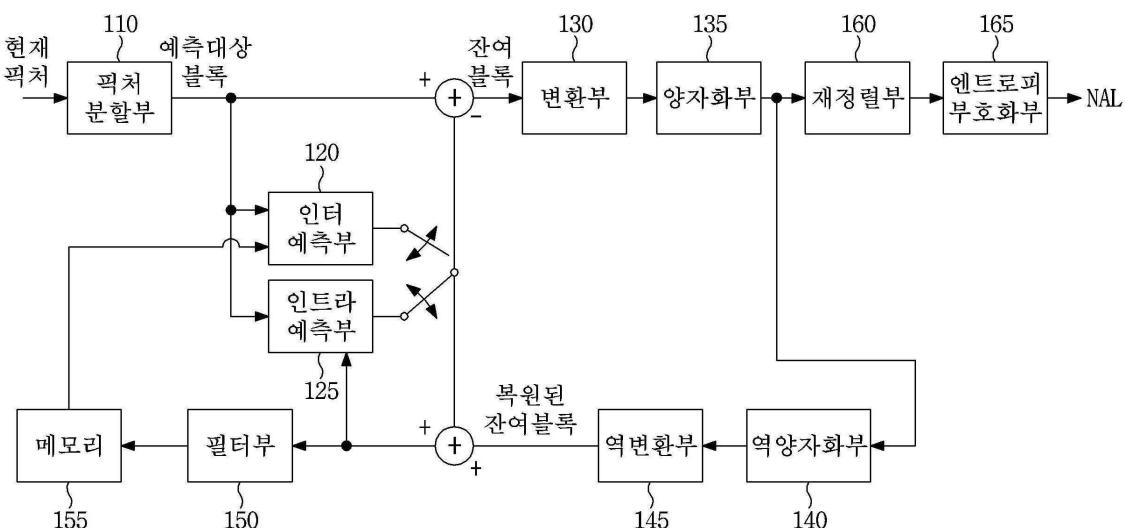
- (51) 국제특허분류(Int. Cl.)
H04N 19/11 (2014.01) *H04N 19/119* (2014.01)
H04N 19/132 (2014.01) *H04N 19/176* (2014.01)
H04N 19/186 (2014.01) *H04N 19/513* (2014.01)
H04N 19/593 (2014.01) *H04N 19/70* (2014.01)
H04N 19/86 (2014.01)
- (52) CPC특허분류
H04N 19/11 (2015.01)
H04N 19/119 (2015.01)
- (21) 출원번호 10-2019-0116383
(22) 출원일자 2019년09월20일
심사청구일자 없음
- (30) 우선권주장
1020180114342 2018년09월21일 대한민국(KR)
(뒷면에 계속)
- (71) 출원인
주식회사 엑스리스
경기도 성남시 분당구 분당내곡로 117, 4층(백현동, 백현알파돔4)
- (72) 발명자
이배근
경기도 성남시 분당구 판교로 393, 214동 1202호(삼평동, 봉들마을)
- (74) 대리인
최윤서

전체 청구항 수 : 총 14 항

(54) 발명의 명칭 영상 신호 부호화/복호화 방법 및 이를 위한 장치

(57) 요약

본 발명에 따른 영상 복호화 방법은, 현재 블록에 머지 후보 리스트를 생성하는 단계, 상기 머지 후보 리스트에 포함된 복수의 머지 후보들 중 어느 하나를 특정하는 단계, 상기 특정된 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 기초로, 상기 현재 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 유도하는 단계, 상기 현재 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 이용하여, 상기 현재 블록 내 서브 블록에 대한 어파인 벡터를 유도하는 단계, 및 상기 어파인 벡터를 기초로, 상기 서브 블록에 대한 움직임 보상 예측을 수행하는 단계를 포함한다.

대 표 도

(52) CPC특허분류

H04N 19/132 (2015.01)

H04N 19/176 (2015.01)

H04N 19/186 (2015.01)

H04N 19/513 (2015.01)

H04N 19/593 (2015.01)

H04N 19/70 (2015.01)

H04N 19/86 (2015.01)

(30) 우선권주장

1020180114343 2018년09월21일 대한민국(KR)

1020180114344 2018년09월21일 대한민국(KR)

명세서

청구범위

청구항 1

현재 블록에 머지 후보 리스트를 생성하는 단계;

상기 머지 후보 리스트에 포함된 복수의 머지 후보들 중 어느 하나를 특정하는 단계;

상기 특정된 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 기초로, 상기 현재 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 유도하는 단계;

상기 현재 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 이용하여, 상기 현재 블록 내 서브 블록에 대한 어파인 벡터를 유도하는 단계, 상기 서브 블록은 상기 현재 블록 보다 작은 크기의 영역임; 및

상기 어파인 벡터를 기초로, 상기 서브 블록에 대한 움직임 보상 예측을 수행하는 단계를 포함하되,

상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 현재 블록 주변에 인접하는 이웃 블록의 움직임 정보를 기초로 유도되는 것을 특징으로 하는, 영상 복호화 방법.

청구항 2

제1 항에 있어서,

상기 이웃 블록이 상기 현재 블록과 상이한 코딩 트리 유닛에 포함된 경우, 상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 이웃 블록의 좌측 하단 서브 블록 및 우측 하단 서브 블록의 움직임 벡터들을 기초로 유도되는 것을 특징으로 하는, 영상 복호화 방법.

청구항 3

제2 항에 있어서,

상기 좌측 하단 서브 블록은 상기 이웃 블록 내 좌측 하단 코너에 위치하는 좌측 하단 기준 샘플을 포함하고, 상기 우측 하단 서브 블록은 상기 이웃 블록 내 좌측 하단 코너에 위치하는 우측 하단 기준 샘플을 포함하는, 영상 복호화 방법.

청구항 4

제3 항에 있어서,

상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 좌측 하단 서브 블록 및 상기 우측 하단 서브 블록의 모션 벡터들의 차분 값을 스케일링 팩터를 이용한 비트 시프트 연산을 기초로 획득된 값을 기초로 유도되고,

상기 스케일링 팩터는 상기 좌측 하단 기준 샘플 및 상기 우측 하단 기준 샘플 사이의 수평 거리 및 오프셋을 가산한 값을 기초로 유도되는 것을 특징으로 하는, 영상 복호화 방법.

청구항 5

제3 항에 있어서,

상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 좌측 하단 서브 블록 및 상기 우측 하단 서브 블록의 모션 벡터들의 차분 값을 스케일링 팩터를 이용한 비트 시프트 연산을 기초로 획득된 값을 기초로 유도되고, 상기 스케일링 팩터는 상기 우측 하단 기준 샘플의 우측에 이웃하는 이웃 샘플과 상기 좌측 하단 기준 샘플 사이의 거리를 기초로 유도되는 것을 특징으로 하는, 영상 복호화 방법.

청구항 6

제1 항에 있어서,

상기 머지 후보 리스트는, 상기 현재 블록의 상단에 위치하는 상단 이웃 블록들 중 첫번째로 가용한 것으로 판단되는 상단 이웃 블록을 기초로 유도되는 제1 머지 후보, 및 상기 현재 블록의 좌측에 위치하는 좌측 이웃 블록들 중 첫번째로 가용한 것으로 판단되는 좌측 이웃 블록을 기초로 유도되는 제2 머지 후보를 포함하는, 영상 복호화 방법.

청구항 7

제1 항에 있어서,

상기 이웃 블록이 상기 현재 블록과 동일한 코딩 트리 유닛에 포함된 경우,

상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 이웃 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 기초로 유도되는 것을 특징으로 하는, 영상 복호화 방법.

청구항 8

현재 블록에 머지 후보 리스트를 생성하는 단계;

상기 머지 후보 리스트에 포함된 복수의 머지 후보들 중 어느 하나를 특정하는 단계;

상기 특정된 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 기초로, 상기 현재 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 유도하는 단계;

상기 현재 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 이용하여, 상기 현재 블록 내 서브 블록에 대한 어파인 벡터를 유도하는 단계, 상기 서브 블록은 상기 현재 블록 보다 작은 크기의 영역임; 및

상기 어파인 벡터를 기초로, 상기 서브 블록에 대한 움직임 보상 예측을 수행하는 단계를 포함하되,

상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 현재 블록 주변에 인접하는 이웃 블록의 움직임 정보를 기초로 유도되는 것을 특징으로 하는, 영상 부호화 방법.

청구항 9

제8 항에 있어서,

상기 이웃 블록이 상기 현재 블록과 상이한 코딩 트리 유닛에 포함된 경우, 상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 이웃 블록의 좌측 하단 서브 블록 및 우측 하단 서브 블록의 움직임 벡터들을 기초로 유도되는 것을 특징으로 하는, 영상 부호화 방법.

청구항 10

제9 항에 있어서,

상기 좌측 하단 서브 블록은 상기 이웃 블록 내 좌측 하단 코너에 위치하는 좌측 하단 기준 샘플을 포함하고, 상기 우측 하단 서브 블록은 상기 이웃 블록 내 좌측 하단 코너에 위치하는 우측 하단 기준 샘플을 포함하는, 영상 부호화 방법.

청구항 11

제10 항에 있어서,

상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 좌측 하단 서브 블록 및 상기 우측 하단 서브 블록의 모션 벡터들의 차분 값을 스케일링 팩터를 이용한 비트 시프트 연산을 기초로 획득된 값을 기초로 유도되고,

상기 스케일링 팩터는 상기 좌측 하단 기준 샘플 및 상기 우측 하단 기준 샘플 사이의 수평 거리 및 오프셋을 가산한 값을 기초로 유도되는 것을 특징으로 하는, 영상 부호화 방법.

청구항 12

제10 항에 있어서,

상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 좌측 하단 서브 블록 및 상기 우측

하단 서브 블록의 모션 벡터들의 차분 값을 스케일링 팩터를 이용한 비트 시프트 연산을 기초로 획득된 값을 기초로 유도되고,

상기 스케일링 팩터는 상기 우측 하단 기준 샘플의 우측에 이웃하는 이웃 샘플과 상기 좌측 하단 기준 샘플 사이의 거리를 기초로 유도되는 것을 특징으로 하는, 영상 부호화 방법.

청구항 13

제8 항에 있어서,

상기 머지 후보 리스트는, 상기 현재 블록의 상단에 위치하는 상단 이웃 블록들 중 첫번째로 가용한 것으로 판단되는 상단 이웃 블록을 기초로 유도되는 제1 머지 후보, 및 상기 현재 블록의 좌측에 위치하는 좌측 이웃 블록들 중 첫번째로 가용한 것으로 판단되는 좌측 이웃 블록을 기초로 유도되는 제2 머지 후보를 포함하는, 영상 부호화 방법.

청구항 14

제8 항에 있어서,

상기 이웃 블록이 상기 현재 블록과 동일한 코딩 트리 유닛에 포함된 경우,

상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 이웃 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 기초로 유도되는 것을 특징으로 하는, 영상 부호화 방법.

발명의 설명

기술 분야

[0001]

본 발명은 영상 신호 부호화/복호화 방법 및 이를 위한 장치에 관한 것이다.

배경 기술

[0002]

디스플레이 패널이 점점 더 대형화되는 추세에 따라 점점 더 높은 화질의 비디오 서비스가 요구되고 있다. 고화질 비디오 서비스의 가장 큰 문제는 데이터량이 크게 증가하는 것이며, 이러한 문제를 해결하기 위해, 비디오 압축율을 향상시키기 위한 연구가 활발하게 진행되고 있다. 대표적인 예로, 2009년에 MPEG(Motion Picture Experts Group)과 ITU-T(International Telecommunication Union-Telecommunication) 산하의 VCEG(Video Coding Experts Group)에서는 JCT-VC(Joint Collaborative Team on Video Coding)를 결성하였다. JCT-VC는 H.264/AVC에 비해 약 2배의 압축 성능을 갖는 비디오 압축 표준인 HEVC(High Efficiency Video Coding)를 제안하였으며, 2013년 1월 25일에 표준 승인되었다. 고화질 비디오 서비스의 급격한 발전에 따라 HEVC의 성능도 점차적으로 그 한계를 드러내고 있다.

발명의 내용

해결하려는 과제

[0003]

본 발명은 비디오 신호를 부호화/복호화함에 있어서, 어파인 모델을 이용한 인터 예측 방법 및 상기 방법을 위한 장치를 제공하는 것을 목적으로 한다.

[0004]

본 발명은 비디오 신호를 부호화/복호화함에 있어서, 서브 블록의 병진 움직임 벡터를 이용하여, 어파인 시드 벡터를 유도하는 방법 및 장치를 제공하는 것을 목적으로 한다.

[0005]

본 발명은 비디오 신호를 부호화/복호화함에 있어서, 이웃 블록과 현재 블록의 거리를 2의 역급수로 변환하여 어파인 시드 벡터를 유도하는 방법 및 장치를 제공하는 것을 목적으로 한다.

[0006]

본 발명에서 이루고자 하는 기술적 과제들은 이상에서 언급한 기술적 과제들로 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 발명에 따른 비디오 신호 복호화/부호화 방법은, 현재 블록에 머지 후보 리스트를 생성하는 단계, 상기 머지 후보 리스트에 포함된 복수의 머지 후보들 중 어느 하나를 특정하는 단계, 상기 특정된 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 기초로, 상기 현재 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 유도하는 단계, 상기 현재 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 이용하여, 상기 현재 블록 내 서브 블록에 대한 어파인 벡터를 유도하는 단계, 및 상기 어파인 벡터를 기초로, 상기 서브 블록에 대한 움직임 보상 예측을 수행하는 단계를 포함한다. 이때, 상기 서브 블록은 상기 현재 블록 보다 작은 크기의 영역이다. 또한, 상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 현재 블록 주변에 인접하는 이웃 블록의 움직임 정보를 기초로 유도될 수 있다.
- [0008] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 이웃 블록이 상기 현재 블록과 상이한 코딩 트리 유닛에 포함된 경우, 상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 이웃 블록의 좌측 하단 서브 블록 및 우측 하단 서브 블록의 움직임 벡터들을 기초로 유도될 수 있다.
- [0009] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 좌측 하단 서브 블록은 상기 이웃 블록 내 좌측 하단 코너에 위치하는 좌측 하단 기준 샘플을 포함하고, 상기 우측 하단 서브 블록은 상기 이웃 블록 내 좌측 하단 코너에 위치하는 우측 하단 기준 샘플을 포함할 수 있다.
- [0010] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 좌측 하단 서브 블록 및 상기 우측 하단 서브 블록의 모션 벡터들의 차분 값을 스케일링 팩터를 이용한 비트 시프트 연산을 기초로 획득된 값을 기초로 유도되고, 상기 스케일링 팩터는 상기 좌측 하단 기준 샘플 및 상기 우측 하단 기준 샘플 사이의 수평 거리 및 오프셋을 가산한 값을 기초로 유도될 수 있다.
- [0011] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 좌측 하단 서브 블록 및 상기 우측 하단 서브 블록의 모션 벡터들의 차분 값을 스케일링 팩터를 이용한 비트 시프트 연산을 기초로 획득된 값을 기초로 유도되고, 상기 스케일링 팩터는 상기 우측 하단 기준 샘플의 우측에 이웃하는 이웃 샘플과 상기 좌측 하단 기준 샘플 사이의 거리를 기초로 유도될 수 있다.
- [0012] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 머지 후보 리스트는, 상기 현재 블록의 상단에 위치하는 상단 이웃 블록들 중 첫번째로 가용한 것으로 판단되는 상단 이웃 블록을 기초로 유도되는 제1 머지 후보, 및 상기 현재 블록의 좌측에 위치하는 좌측 이웃 블록들 중 첫번째로 가용한 것으로 판단되는 좌측 이웃 블록을 기초로 유도되는 제2 머지 후보를 포함할 수 있다.
- [0013] 본 발명에 따른 비디오 신호 복호화/부호화 방법에 있어서, 상기 이웃 블록이 상기 현재 블록과 동일한 코딩 트리 유닛에 포함된 경우, 상기 머지 후보의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터는, 상기 이웃 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 기초로 유도될 수 있다.
- [0014] 본 발명에 대하여 위에서 간략하게 요약된 특징들은 후술하는 본 발명의 상세한 설명의 예시적인 양상일 뿐이며, 본 발명의 범위를 제한하는 것은 아니다.

발명의 효과

- [0015] 본 발명에 의하면, 어파인 모델을 이용한 인터 예측 방법을 통해 예측 효율을 향상시킬 수 있는 효과가 있다.
- [0016] 본 발명에 의하면, 서브 블록의 병진 움직임 벡터를 이용하여, 어파인 시드 벡터를 유도함으로써, 부호화 효율을 높일 수 있는 효과가 있다.
- [0017] 본 발명에 의하면, 이웃 블록과 현재 블록의 거리를 2의 역급수로 변환하여 어파인 시드 벡터를 유도함으로써, 부호화 효율을 높일 수 있는 효과가 있다. .
- [0018] 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

- [0019] 도 1은 본 발명의 일 실시예에 따른 영상 부호화기(인코더기)의 블록도이다.

도 2는 본 발명의 일 실시예에 따른 영상 복호화기(디코더기)의 블록도이다.

도 3은 본 발명의 일 실시예에 따른 기본 코딩 트리 유닛을 도시한 도면이다.

도 4는 코딩 블록의 다양한 분할 형태를 나타낸 도면이다.

도 5는 코딩 트리 유닛의 분할 양상을 예시한 도면이다.

도 6은 본 발명의 일 실시예에 따른 인터 예측 방법의 흐름도이다.

도 7은 오브젝트의 비선형 움직임을 예시한 도면이다.

도 8은 본 발명의 일실시예에 따른 어파인 모션에 기초한 인터 예측 방법의 흐름도이다.

도 9는 어파인 모션 모델 별 어파인 시드 벡터를 예시한 도면이다.

도 10은 4 파라미터 모션 모델 하에서 서브 블록들의 어파인 벡터들을 예시한 도면이다.

도 11은 머지 후보를 유도하는데 이용될 수 있는 이웃 블록을 도시한 도면이다.

도 12는 어파인 이웃 블록의 어파인 시드 벡터에 기초하여 현재 블록의 어파인 시드 벡터를 유도하는 예를 나타낸 도면이다.

도 13은 서브 블록의 모션 벡터를 어파인 이웃 블록의 어파인 시드 벡터로 설정하는 예를 나타낸 도면이다.

도 14 내지 도 16은 기준 샘플의 위치를 나타낸 도면이다.

도 17은 변형된 어파인 머지 벡터 유도 방법이 적용된 일 예를 나타낸 도면이다.

도 18은 논 어파인 이웃 블록을 기초로 현재 블록의 어파인 시드 벡터를 유도하는 예를 나타낸 도면이다.

도 19는 이웃 블록을 다른 이웃 블록으로 대체하는 예를 나타낸 도면이다.

도 20 및 도 21은 복수 머지 후보들을 이용한 움직임 보상 예측 방법을 나타낸 흐름도이다.

도 22는 본 발명의 일 실시예에 따른, 인트라 예측 방법의 흐름도이다.

도 23은 각 참조 샘플 라인이 포함하는 참조 샘플들을 나타낸 도면이다.

도 24는 인트라 예측 모드들을 나타낸 도면이다.

도 25 및 도 26은 참조 샘플들을 일렬로 배열하는 일차원 배열의 예시를 나타낸 도면이다.

도 27은 방향성 인트라 예측 모드들이 x축과 평행한 직선과 형성하는 각도를 예시한 도면이다.

도 28은 현재 블록이 비정방 형태인 경우, 예측 샘플이 획득되는 양상을 나타낸 도면이다.

도 29는 와이드 앵글 인트라 예측 모드들을 나타낸 도면이다.

도 30은 블록 강도를 결정하는 과정을 나타낸 흐름도이다.

도 31은 기 정의된 필터 후보들을 나타낸다.

도 32는 꾹처가 복수의 타일들로 분할된 예를 나타낸다.

도 33은 플렉서블 타일 기법에 따른 꾹처의 분할 양상을 나타낸 도면이다.

도 34는 각 코딩 트리 유닛에 타일 아이디가 할당되는 예를 나타낸 도면이다.

도 35는 타일별로 인루프 필터의 적용 여부가 선택적으로 결정되는 예를 도시한 것이다.

발명을 실시하기 위한 구체적인 내용

[0020] 이하에서는 도면을 참조하여 본 발명의 실시예를 상세히 설명한다.

[0021] 영상의 부호화 및 복호화는 블록 단위로 수행된다. 일 예로, 코딩 블록, 변환 블록, 또는 예측 블록에 대해, 변환, 양자화, 예측, 인루프 필터링, 또는 복원 등의 부호화/복호화 처리가 수행될 수 있다.

[0022] 이하, 부호화/복호화 대상인 블록을 '현재 블록'이라 호칭하기로 한다. 일 예로, 현재 블록은 현재 부호화/복호

화 처리 단계에 따라, 코딩 블록, 변환 블록, 또는 예측 블록을 나타낼 수 있다.

[0023] 아울러, 본 명세서에서 사용되는 용어 '유닛'은 특정 부호화/복호화 프로세스를 수행하기 위한 기본 단위를 나타내고, '블록'은 소정 크기의 샘플 어레이를 나타내는 것으로 이해될 수 있다. 별도의 설명이 없는 한, '블록'과 '유닛'은 동등한 의미로 사용될 수 있다. 일 예로, 후술되는 실시예에서, 코딩 블록과 코딩 유닛은 상호 동등한 의미를 갖는 것으로 이해될 수 있다.

[0024] 도 1은 본 발명의 일 실시예에 따른 영상 부호화기(인코더기)의 블록도이다.

[0025] 도 1을 참조하면, 영상 부호화 장치(100)는 꾹쳐 분할부(110), 예측부(120, 125), 변환부(130), 양자화부(135), 재정렬부(160), 엔트로피 부호화부(165), 역양자화부(140), 역변환부(145), 필터부(150) 및 메모리(155)를 포함할 수 있다.

[0026] 도 1에 나타난 각 구성부들은 영상 부호화 장치에서 서로 다른 특징적인 기능들을 나타내기 위해 독립적으로 도시한 것으로, 각 구성부들이 분리된 하드웨어나 하나의 소프트웨어 구성단위로 이루어짐을 의미하지 않는다. 즉, 각 구성부는 설명의 편의상 각각의 구성부로 나열하여 포함한 것으로 각 구성부 중 적어도 두 개의 구성부가 합쳐져 하나의 구성부로 이루어지거나, 하나의 구성부가 복수개의 구성부로 나뉘어져 기능을 수행할 수 있고 이러한 각 구성부의 통합된 실시예 및 분리된 실시예도 본 발명의 본질에서 벗어나지 않는 한 본 발명의 권리범위에 포함된다.

[0027] 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.

[0028] 꾹쳐 분할부(110)는 입력된 꾹쳐를 적어도 하나의 처리 단위로 분할할 수 있다. 이때, 처리 단위는 예측 단위(Prediction Unit: PU)일 수도 있고, 변환 단위(Transform Unit: TU)일 수도 있으며, 부호화 단위(Coding Unit: CU)일 수도 있다. 꾹쳐 분할부(110)에서는 하나의 꾹쳐에 대해 복수의 부호화 단위, 예측 단위 및 변환 단위의 조합으로 분할하고 소정의 기준(예를 들어, 비용 함수)으로 하나의 부호화 단위, 예측 단위 및 변환 단위 조합을 선택하여 꾹쳐를 부호화 할 수 있다.

[0029] 예를 들어, 하나의 꾹쳐는 복수개의 부호화 단위로 분할될 수 있다. 꾹쳐에서 부호화 단위를 분할하기 위해서는 쿼드 트리 구조(Quad Tree Structure)와 같은 재귀적인 트리 구조를 사용할 수 있는데 하나의 영상 또는 최대 크기 부호화 단위(largest coding unit)를 루트로 하여 다른 부호화 단위로 분할되는 부호화 유닛은 분할된 부호화 단위의 개수만큼의 자식 노드를 가지고 분할될 수 있다. 일정한 제한에 따라 더 이상 분할되지 않는 부호화 단위는 리프 노드가 된다. 즉, 하나의 코딩 유닛에 대하여 정방형 분할만이 가능하다고 가정하는 경우, 하나의 부호화 단위는 최대 4개의 다른 부호화 단위로 분할될 수 있다.

[0030] 이하, 본 발명의 실시예에서는 부호화 단위는 부호화를 수행하는 단위의 의미로 사용할 수도 있고, 복호화를 수행하는 단위의 의미로 사용할 수도 있다.

[0031] 예측 단위는 하나의 부호화 단위 내에서 동일한 크기의 적어도 하나의 정사각형 또는 직사각형 등의 형태를 가지고 분할된 것일 수도 있고, 하나의 부호화 단위 내에서 분할된 예측 단위 중 어느 하나의 예측 단위가 다른 하나의 예측 단위와 상이한 형태 및/또는 크기를 가지고 분할된 것일 수도 있다.

[0032] 부호화 단위를 기초로 인트라 예측을 수행하는 예측 단위를 생성시 최소 부호화 단위가 아닌 경우, 복수의 예측 단위 NxN 으로 분할하지 않고 인트라 예측을 수행할 수 있다.

[0033] 예측부(120, 125)는 인터 예측을 수행하는 인터 예측부(120)와 인트라 예측을 수행하는 인트라 예측부(125)를 포함할 수 있다. 예측 단위에 대해 인터 예측을 사용할 것인지 또는 인트라 예측을 수행할 것인지를 결정하고, 각 예측 방법에 따른 구체적인 정보(예컨대, 인트라 예측 모드, 모션 벡터, 참조 꾹쳐 등)를 결정할 수 있다. 이때, 예측이 수행되는 처리 단위와 예측 방법 및 구체적인 내용이 정해지는 처리 단위는 다를 수 있다. 예컨대, 예측의 방법과 예측 모드 등은 예측 단위로 결정되고, 예측의 수행은 변환 단위로 수행될 수도 있다. 생성된 예측 블록과 원본 블록 사이의 잔차값(잔차 블록)은 변환부(130)로 입력될 수 있다. 또한, 예측을 위해 사용한 예측 모드 정보, 모션 벡터 정보 등을 잔차값과 함께 엔트로피 부호화부(165)에서 부호화되어 복호화기에 전달될 수 있다. 특정한 부호화 모드를 사용할 경우, 예측부(120, 125)를 통해 예측 블록을 생성하지 않고, 원본 블록을 그대로 부호화하여 복호화부에 전송하는 것도 가능하다.

- [0034] 인터 예측부(120)는 현재 픽쳐의 이전 픽쳐 또는 이후 픽쳐 중 적어도 하나의 픽쳐의 정보를 기초로 예측 단위를 예측할 수도 있고, 경우에 따라서는 현재 픽쳐 내의 부호화가 완료된 일부 영역의 정보를 기초로 예측 단위를 예측할 수도 있다. 인터 예측부(120)는 참조 픽쳐 보간부, 모션 예측부, 움직임 보상부를 포함할 수 있다.
- [0035] 참조 픽쳐 보간부에서는 메모리(155)로부터 참조 픽쳐 정보를 제공받고 참조 픽쳐에서 정수 화소 이하의 화소 정보를 생성할 수 있다. 휘도 화소의 경우, 1/4 화소 단위로 정수 화소 이하의 화소 정보를 생성하기 위해 필터 계수를 달리하는 DCT 기반의 8탭 보간 필터(DCT-based Interpolation Filter)가 사용될 수 있다. 색차 신호의 경우 1/8 화소 단위로 정수 화소 이하의 화소 정보를 생성하기 위해 필터 계수를 달리하는 DCT 기반의 4탭 보간 필터(DCT-based Interpolation Filter)가 사용될 수 있다.
- [0036] 모션 예측부는 참조 픽쳐 보간부에 의해 보간된 참조 픽쳐를 기초로 모션 예측을 수행할 수 있다. 모션 벡터를 산출하기 위한 방법으로 FBMA(Full search-based Block Matching Algorithm), TSS(Three Step Search), NTS(New Three-Step Search Algorithm) 등 다양한 방법이 사용될 수 있다. 모션 벡터는 보간된 화소를 기초로 1/2 또는 1/4 화소 단위의 모션 벡터값을 가질 수 있다. 모션 예측부에서는 모션 예측 방법을 다르게 하여 현재 예측 단위를 예측할 수 있다. 모션 예측 방법으로 스kip(Skip) 방법, 머지(Merge) 방법, AMVP(Advanced Motion Vector Prediction) 방법, 인트라 블록 카피(Intra Block Copy) 방법 등 다양한 방법이 사용될 수 있다.
- [0037] 인트라 예측부(125)는 현재 픽쳐 내의 화소 정보인 현재 블록 주변의 참조 픽셀 정보를 기초로 예측 단위를 생성할 수 있다. 현재 예측 단위의 주변 블록이 인터 예측을 수행한 블록이어서, 참조 픽셀이 인터 예측을 수행한 픽셀일 경우, 인터 예측을 수행한 블록에 포함되는 참조 픽셀을 주변의 인트라 예측을 수행한 블록의 참조 픽셀 정보로 대체하여 사용할 수 있다. 즉, 참조 픽셀이 가용하지 않는 경우, 가용하지 않은 참조 픽셀 정보를 가용한 참조 픽셀 중 적어도 하나의 참조 픽셀로 대체하여 사용할 수 있다.
- [0038] 인트라 예측에서 예측 모드는 참조 픽셀 정보를 예측 방향에 따라 사용하는 방향성 예측 모드와 예측을 수행시 방향성 정보를 사용하지 않는 비방향성 모드를 가질 수 있다. 휘도 정보를 예측하기 위한 모드와 색차 정보를 예측하기 위한 모드가 상이할 수 있고, 색차 정보를 예측하기 위해 휘도 정보를 예측하기 위해 사용된 인트라 예측 모드 정보 또는 예측된 휘도 신호 정보를 활용할 수 있다.
- [0039] 인트라 예측을 수행할 때 예측 단위의 크기와 변환 단위의 크기가 동일할 경우, 예측 단위의 좌측에 존재하는 픽셀, 좌측 상단에 존재하는 픽셀, 상단에 존재하는 픽셀을 기초로 예측 단위에 대한 인트라 예측을 수행할 수 있다. 그러나 인트라 예측을 수행할 때 예측 단위의 크기와 변환 단위의 크기가 상이할 경우, 변환 단위를 기초로 한 참조 픽셀을 이용하여 인트라 예측을 수행할 수 있다. 또한, 최소 부호화 단위에 대해서만 NxN 분할을 사용하는 인트라 예측을 사용할 수 있다.
- [0040] 인트라 예측 방법은 예측 모드에 따라 참조 화소에 AIS(Adaptive Intra Smoothing) 필터를 적용한 후 예측 블록을 생성할 수 있다. 참조 화소에 적용되는 AIS 필터의 종류는 상이할 수 있다. 인트라 예측 방법을 수행하기 위해 현재 예측 단위의 인트라 예측 모드는 현재 예측 단위의 주변에 존재하는 예측 단위의 인트라 예측 모드로부터 예측할 수 있다. 주변 예측 단위로부터 예측된 모드 정보를 이용하여 현재 예측 단위의 예측 모드를 예측하는 경우, 현재 예측 단위와 주변 예측 단위의 인트라 예측 모드가 동일하면 소정의 플래그 정보를 이용하여 현재 예측 단위와 주변 예측 단위의 예측 모드가 동일하다는 정보를 전송할 수 있고, 만약 현재 예측 단위와 주변 예측 단위의 예측 모드가 상이하면 엔트로피 부호화를 수행하여 현재 블록의 예측 모드 정보를 부호화할 수 있다.
- [0041] 또한, 예측부(120, 125)에서 생성된 예측 단위를 기초로 예측을 수행한 예측 단위와 예측 단위의 원본 블록과 차이값인 잔차값(Residual) 정보를 포함하는 잔차 블록이 생성될 수 있다. 생성된 잔차 블록은 변환부(130)로 입력될 수 있다.
- [0042] 변환부(130)에서는 원본 블록과 예측부(120, 125)를 통해 생성된 예측 단위의 잔차값(residual) 정보를 포함한 잔차 블록을 DCT(Discrete Cosine Transform), DST(Discrete Sine Transform) 또는 변환 스kip과 같은 변환 방법을 사용하여 변환시킬 수 있다. 잔차 블록을 변환하기 위해 DCT를 적용할지, DST를 적용할지 또는 KLT를 적용할지는 잔차 블록을 생성하기 위해 사용된 예측 단위의 인트라 예측 모드 정보를 기초로 결정할 수 있다.
- [0043] 양자화부(135)는 변환부(130)에서 주파수 영역으로 변환된 값들을 양자화할 수 있다. 블록에 따라 또는 영상의 중요도에 따라 양자화 계수는 변할 수 있다. 양자화부(135)에서 산출된 값은 역양자화부(140)와 재정렬부(160)에 제공될 수 있다.

- [0044] 재정렬부(160)는 양자화된 잔차값에 대해 계수값의 재정렬을 수행할 수 있다.
- [0045] 재정렬부(160)는 계수 스캐닝(Coefficient Scanning) 방법을 통해 2차원의 블록 형태 계수를 1차원의 벡터 형태로 변경할 수 있다. 예를 들어, 재정렬부(160)에서는 지그-재그 스캔(Zig-Zag Scan) 방법을 이용하여 DC 계수부터 고주파수 영역의 계수까지 스캔하여 1차원 벡터 형태로 변경시킬 수 있다. 변환 단위의 크기 및 인트라 예측 모드에 따라 지그-재그 스캔 대신 2차원의 블록 형태 계수를 열 방향으로 스캔하는 수직 스캔, 2차원의 블록 형태 계수를 행 방향으로 스캔하는 수평 스캔이 사용될 수도 있다. 즉, 변환 단위의 크기 및 인트라 예측 모드에 따라 지그-재그 스캔, 수직 방향 스캔 및 수평 방향 스캔 중 어떠한 스캔 방법이 사용될지 여부를 결정할 수 있다.
- [0046] 엔트로피 부호화부(165)는 재정렬부(160)에 의해 산출된 값들을 기초로 엔트로피 부호화를 수행할 수 있다. 엔트로피 부호화는 예를 들어, 지수 골롬(Exponential Golomb), CAVLC(Context-Adaptive Variable Length Coding), CABAC(Context-Adaptive Binary Arithmetic Coding)과 같은 다양한 부호화 방법을 사용할 수 있다.
- [0047] 엔트로피 부호화부(165)는 재정렬부(160) 및 예측부(120, 125)로부터 부호화 단위의 잔차값 계수 정보 및 블록 타입 정보, 예측 모드 정보, 분할 단위 정보, 예측 단위 정보 및 전송 단위 정보, 모션 벡터 정보, 참조 프레임 정보, 블록의 보간 정보, 필터링 정보 등 다양한 정보를 부호화할 수 있다.
- [0048] 엔트로피 부호화부(165)에서는 재정렬부(160)에서 입력된 부호화 단위의 계수값을 엔트로피 부호화할 수 있다.
- [0049] 역양자화부(140) 및 역변환부(145)에서는 양자화부(135)에서 양자화된 값들을 역양자화하고 변환부(130)에서 변환된 값들을 역변환한다. 역양자화부(140) 및 역변환부(145)에서 생성된 잔차값(Residual)은 예측부(120, 125)에 포함된 움직임 추정부, 움직임 보상부 및 인트라 예측부를 통해서 예측된 예측 단위와 합쳐져 복원 블록(Reconstructed Block)을 생성할 수 있다.
- [0050] 필터부(150)는 디블록킹 필터, 오프셋 보정부, ALF(Adaptive Loop Filter) 중 적어도 하나를 포함할 수 있다.
- [0051] 디블록킹 필터는 복원된 픽쳐에서 블록간의 경계로 인해 생긴 블록 왜곡을 제거할 수 있다. 디블록킹을 수행할지 여부를 판단하기 위해 블록에 포함된 몇 개의 열 또는 행에 포함된 픽셀을 기초로 현재 블록에 디블록킹 필터 적용할지 여부를 판단할 수 있다. 블록에 디블록킹 필터를 적용하는 경우 필요한 디블록킹 필터링 강도에 따라 강한 필터(Strong Filter) 또는 약한 필터(Weak Filter)를 적용할 수 있다. 또한 디블록킹 필터를 적용함에 있어 수직 필터링 및 수평 필터링 수행시 수평 방향 필터링 및 수직 방향 필터링이 병행 처리되도록 할 수 있다.
- [0052] 오프셋 보정부는 디블록킹을 수행한 영상에 대해 픽셀 단위로 원본 영상과의 오프셋을 보정할 수 있다. 특정 픽쳐에 대한 오프셋 보정을 수행하기 위해 영상에 포함된 픽셀을 일정한 수의 영역으로 구분한 후 오프셋을 수행할 영역을 결정하고 해당 영역에 오프셋을 적용하는 방법 또는 각 픽셀의 에지 정보를 고려하여 오프셋을 적용하는 방법을 사용할 수 있다.
- [0053] ALF(Adaptive Loop Filtering)는 필터링한 복원 영상과 원래의 영상을 비교한 값을 기초로 수행될 수 있다. 영상에 포함된 픽셀을 소정의 그룹으로 나눈 후 해당 그룹에 적용될 하나의 필터를 결정하여 그룹마다 차별적으로 필터링을 수행할 수 있다. ALF를 적용할지 여부에 관련된 정보는 휘도 신호는 부호화 단위(Coding Unit, CU) 별로 전송될 수 있고, 각각의 블록에 따라 적용될 ALF 필터의 모양 및 필터 계수는 달라질 수 있다. 또한, 적용 대상 블록의 특성에 상관없이 동일한 형태(고정된 형태)의 ALF 필터가 적용될 수도 있다.
- [0054] 메모리(155)는 필터부(150)를 통해 산출된 복원 블록 또는 픽쳐를 저장할 수 있고, 저장된 복원 블록 또는 픽쳐는 인터 예측을 수행 시 예측부(120, 125)에 제공될 수 있다.
- [0055] 도 2는 본 발명의 일 실시예에 따른 영상 복호화기(디코더기)의 블록도이다.
- [0056] 도 2를 참조하면, 영상 복호화기(200)는 엔트로피 복호화부(210), 재정렬부(215), 역양자화부(220), 역변환부(225), 예측부(230, 235), 필터부(240), 메모리(245)가 포함될 수 있다.
- [0057] 영상 부호화기에서 영상 비트스트림이 입력된 경우, 입력된 비트스트림은 영상 부호화기와 반대의 절차로 복호화될 수 있다.
- [0058] 엔트로피 복호화부(210)는 영상 부호화기의 엔트로피 부호화부에서 엔트로피 부호화를 수행한 것과 반대의 절차로 엔트로피 복호화를 수행할 수 있다. 예를 들어, 영상 부호화기에서 수행된 방법에 대응하여 지수 골롬(Exponential Golomb), CAVLC(Context-Adaptive Variable Length Coding), CABAC(Context-Adaptive Binary

Arithmetic Coding)과 같은 다양한 방법이 적용될 수 있다.

[0059] 엔트로피 복호화부(210)에서는 부호화기에서 수행된 인트라 예측 및 인터 예측에 관련된 정보를 복호화할 수 있다.

[0060] 재정렬부(215)는 엔트로피 복호화부(210)에서 엔트로피 복호화된 비트스트림을 부호화부에서 재정렬한 방법을 기초로 재정렬을 수행할 수 있다. 1차원 벡터 형태로 표현된 계수들을 다시 2차원의 블록 형태의 계수로 복원하여 재정렬할 수 있다. 재정렬부(215)에서는 부호화부에서 수행된 계수 스캐닝에 관련된 정보를 제공받고 해당 부호화부에서 수행된 스캐닝 순서에 기초하여 역으로 스캐닝하는 방법을 통해 재정렬을 수행할 수 있다.

[0061] 역양자화부(220)는 부호화기에서 제공된 양자화 파라미터와 재정렬된 블록의 계수값을 기초로 역양자화를 수행할 수 있다.

[0062] 역변환부(225)는 영상 부호화기에서 수행한 양자화 결과에 대해 변환부에서 수행한 변환 즉, DCT, DST, 및 KLT에 대해 역변환 즉, 역 DCT, 역 DST 및 역 KLT를 수행할 수 있다. 역변환은 영상 부호화기에서 결정된 전송 단위를 기초로 수행될 수 있다. 영상 복호화기의 역변환부(225)에서는 예측 방법, 현재 블록의 크기 및 예측 방향 등 복수의 정보에 따라 변환 기법(예를 들어, DCT, DST, KLT)이 선택적으로 수행될 수 있다.

[0063] 예측부(230, 235)는 엔트로피 복호화부(210)에서 제공된 예측 블록 생성 관련 정보와 메모리(245)에서 제공된 이전에 복호화된 블록 또는 픽쳐 정보를 기초로 예측 블록을 생성할 수 있다.

[0064] 전술한 바와 같이 영상 부호화기에서의 동작과 동일하게 인트라 예측을 수행시 예측 단위의 크기와 변환 단위의 크기가 동일할 경우, 예측 단위의 좌측에 존재하는 픽셀, 좌측 상단에 존재하는 픽셀, 상단에 존재하는 픽셀을 기초로 예측 단위에 대한 인트라 예측을 수행하지만, 인트라 예측을 수행시 예측 단위의 크기와 변환 단위의 크기가 상이할 경우, 변환 단위를 기초로 한 참조 픽셀을 이용하여 인트라 예측을 수행할 수 있다. 또한, 최소 부호화 단위에 대해서만 NxN 분할을 사용하는 인트라 예측을 사용할 수도 있다.

[0065] 예측부(230, 235)는 예측 단위 판별부, 인터 예측부 및 인트라 예측부를 포함할 수 있다. 예측 단위 판별부는 엔트로피 복호화부(210)에서 입력되는 예측 단위 정보, 인트라 예측 방법의 예측 모드 정보, 인터 예측 방법의 모션 예측 관련 정보 등 다양한 정보를 입력 받고 현재 부호화 단위에서 예측 단위를 구분하고, 예측 단위가 인터 예측을 수행하는지 아니면 인트라 예측을 수행하는지 여부를 판별할 수 있다. 인터 예측부(230)는 영상 부호화기에서 제공된 현재 예측 단위의 인터 예측에 필요한 정보를 이용해 현재 예측 단위가 포함된 현재 픽쳐의 이전 픽쳐 또는 이후 픽쳐 중 적어도 하나의 픽쳐에 포함된 정보를 기초로 현재 예측 단위에 대한 인터 예측을 수행할 수 있다. 또는, 현재 예측 단위가 포함된 현재 픽쳐 내에서 기-복원된 일부 영역의 정보를 기초로 인터 예측을 수행할 수도 있다.

[0066] 인터 예측을 수행하기 위해 부호화 단위를 기준으로 해당 부호화 단위에 포함된 예측 단위의 모션 예측 방법이 스kip 모드(Skip Mode), 머지 모드(Merge 모드), 모션 벡터 예측 모드(AMVP Mode), 인트라 블록 카피 모드 중 어떠한 방법인지 여부를 판단할 수 있다.

[0067] 인트라 예측부(235)는 현재 픽쳐 내의 화소 정보를 기초로 예측 블록을 생성할 수 있다. 예측 단위가 인트라 예측을 수행한 예측 단위인 경우, 영상 부호화기에서 제공된 예측 단위의 인트라 예측 모드 정보를 기초로 인트라 예측을 수행할 수 있다. 인트라 예측부(235)에는 AIS(Adaptive Intra Smoothing) 필터, 참조 화소 보간부, DC 필터를 포함할 수 있다. AIS 필터는 현재 블록의 참조 화소에 필터링을 수행하는 부분으로써 현재 예측 단위의 예측 모드에 따라 필터의 적용 여부를 결정하여 적용할 수 있다. 영상 부호화기에서 제공된 예측 단위의 예측 모드 및 AIS 필터 정보를 이용하여 현재 블록의 참조 화소에 AIS 필터링을 수행할 수 있다. 현재 블록의 예측 모드가 AIS 필터링을 수행하지 않는 모드일 경우, AIS 필터는 적용되지 않을 수 있다.

[0068] 참조 화소 보간부는 예측 단위의 예측 모드가 참조 화소를 보간한 화소값을 기초로 인트라 예측을 수행하는 예측 단위일 경우, 참조 화소를 보간하여 정수값 이하의 화소 단위의 참조 화소를 생성할 수 있다. 현재 예측 단위의 예측 모드가 참조 화소를 보간하지 않고 예측 블록을 생성하는 예측 모드일 경우 참조 화소는 보간되지 않을 수 있다. DC 필터는 현재 블록의 예측 모드가 DC 모드일 경우 필터링을 통해서 예측 블록을 생성할 수 있다.

[0069] 복원된 블록 또는 픽쳐는 필터부(240)로 제공될 수 있다. 필터부(240)는 디블록킹 필터, 오프셋 보정부, ALF를 포함할 수 있다.

[0070] 영상 부호화기로부터 해당 블록 또는 픽쳐에 디블록킹 필터를 적용하였는지 여부에 대한 정보 및 디블록킹 필터를 적용하였을 경우, 강한 필터를 적용하였는지 또는 약한 필터를 적용하였는지에 대한 정보를 제공받을 수 있

다. 영상 복호화기의 디블록킹 필터에서는 영상 부호화기에서 제공된 디블록킹 필터 관련 정보를 제공받고 영상 복호화기에서 해당 블록에 대한 디블록킹 필터링을 수행할 수 있다.

[0071] 오프셋 보정부는 부호화시 영상에 적용된 오프셋 보정의 종류 및 오프셋 값 정보 등을 기초로 복원된 영상에 오프셋 보정을 수행할 수 있다.

[0072] ALF는 부호화기로부터 제공된 ALF 적용 여부 정보, ALF 계수 정보 등을 기초로 부호화 단위에 적용될 수 있다. 이러한 ALF 정보는 특정한 파라메터 셋에 포함되어 제공될 수 있다.

[0073] 메모리(245)는 복원된 픽쳐 또는 블록을 저장하여 참조 픽쳐 또는 참조 블록으로 사용할 수 있도록 할 수 있고 또한 복원된 픽쳐를 출력부로 제공할 수 있다.

[0075] 도 3은 본 발명의 일 실시예에 따른 기본 코딩 트리 유닛을 도시한 도면이다.

[0076] 최대 크기의 코딩 블록을 코딩 트리 블록이라 정의할 수 있다. 하나의 픽쳐는 복수개의 코딩 트리 유닛(Coding Tree Unit, CTU)으로 분할된다. 코딩 트리 유닛은 최대 크기의 코딩 유닛으로, LCU (Largest Coding Unit)라 호칭될 수도 있다. 도 3은 하나의 픽쳐가 복수개의 코딩 트리 유닛으로 분할된 예를 나타낸 것이다.

[0077] 코딩 트리 유닛의 크기는 픽쳐 레벨 또는 시퀀스 레벨에서 정의될 수 있다. 이를 위해, 코딩 트리 유닛의 크기를 나타내는 정보가 픽쳐 파라미터 세트 또는 시퀀스 파라미터 세트를 통해 시그널링될 수 있다.

[0078] 일 예로, 시퀀스 내 전체 픽쳐에 대한 코딩 트리 유닛의 크기가 128x128로 설정될 수 있다. 또는, 픽쳐 레벨에서 128x128 또는 256x256 중 어느 하나를 코딩 트리 유닛의 크기로 결정할 수 있다. 일 예로, 제1 픽쳐에서는 코딩 트리 유닛의 크기가 128x128로 설정되고, 제2 픽쳐에서는 코딩 트리 유닛의 크기가 256x256으로 설정될 수 있다.

[0079] 코딩 트리 유닛을 분할하여, 코딩 블록을 생성할 수 있다. 코딩 블록은 부호화/복호화 처리를 위한 기본 단위를 나타낸다. 일 예로, 코딩 블록 별로 예측 또는 변환이 수행되거나, 코딩 블록 별로 예측 부호화 모드가 결정될 수 있다. 여기서, 예측 부호화 모드는 예측 영상을 생성하는 방법을 나타낸다. 일 예로, 예측 부호화 모드는 화면 내 예측(Intra Prediction, 인트라 예측), 화면 간 예측(Inter Prediction, 인터 예측), 현재 픽쳐 참조(Current Picture Referencing, CPR, 또는 인트라 블록 카피(Intra Block Copy, IBC)) 또는 복합 예측(Combined Prediction)을 포함할 수 있다. 코딩 블록에 대해, 인트라 예측, 인터 예측, 현재 픽쳐 참조 또는 복합 예측 중 적어도 하나의 예측 부호화 모드를 이용하여, 코딩 블록에 대한 예측 블록을 생성할 수 있다.

[0080] 현재 블록의 예측 부호화 모드를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 상기 정보는 예측 부호화 모드가 인트라 모드인지 또는 인터 모드인지 여부를 나타내는 1비트 플래그일 수 있다. 현재 블록의 예측 부호화 모드가 인터 모드로 결정된 경우에 한하여, 현재 픽쳐 참조 또는 복합 예측이 이용 가능할 수 있다.

[0081] 현재 픽쳐 참조는 현재 픽처를 참조 픽쳐로 설정하고, 현재 픽쳐 내 이미 부호화/복호화가 완료된 영역으로부터 현재 블록의 예측 블록을 획득하기 위한 것이다. 여기서, 현재 픽쳐는 현재 블록을 포함하는 픽처를 의미한다. 현재 블록에 현재 픽쳐 참조가 적용되는지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 상기 정보는 1비트의 플래그일 수 있다. 상기 플래그가 참인 경우, 현재 블록의 예측 부호화 모드는 현재 픽쳐 참조로 결정되고, 상기 플래그가 거짓인 경우, 현재 블록의 예측 모드는 인터 예측으로 결정될 수 있다.

[0082] 또는, 참조 픽쳐 인덱스를 기초로, 현재 블록의 예측 부호화 모드가 결정될 수 있다. 일 예로, 참조 픽쳐 인덱스가 현재 픽처를 가리키는 경우, 현재 블록의 예측 부호화 모드는 현재 픽쳐 참조로 결정될 수 있다. 참조 픽쳐 인덱스가 현재 픽처가 아닌 다른 다른 픽처를 가리키는 경우, 현재 블록의 예측 부호화 모드는 인터 예측으로 결정될 수 있다. 즉, 현재 픽쳐 참조는 현재 픽쳐 내 부호화/복호화가 완료된 영역의 정보를 이용한 예측 방법이고, 인터 예측은 부호화/복호화가 완료된 다른 픽처의 정보를 이용한 예측 방법이다.

[0083] 복합 예측은 인트라 예측, 인터 예측 및 현재 픽쳐 참조 중 둘 이상을 조합된 부호화 모드를 나타낸다. 일 예로, 복합 예측이 적용되는 경우, 인트라 예측, 인터 예측 또는 현재 픽쳐 참조 중 어느 하나를 기초로 제1 예측 블록이 생성되고, 다른 하나를 기초로 제2 예측 블록이 생성될 수 있다. 제1 예측 블록 및 제2 예측 블록이 생성되면, 제1 예측 블록 및 제2 예측 블록의 평균 연산 또는 가중합 연산을 통해 최종 예측 블록이 생성될 수 있다. 복합 예측이 적용되는지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1

비트의 플래그일 수 있다.

[0084] 도 4는 코딩 블록의 다양한 분할 형태를 나타낸 도면이다.

[0085] 코딩 블록은 쿼드 트리 분할, 바이너리 트리 분할 또는 트리플 트리 분할을 기초로 복수의 코딩 블록들로 분할될 수 있다. 분할된 코딩 블록도 다시 쿼드 트리 분할, 바이너리 트리 분할 또는 트리플 트리 분할을 기초로 다시 복수의 코딩 블록들로 분할될 수 있다.

[0086] 쿼드 트리 분할은 현재 블록을 4개의 블록들로 분할하는 분할 기법을 나타낸다. 쿼드 트리 분할의 결과, 현재 블록은 4개의 정방 형태 파티션들로 분할될 수 있다 (도 4의 (a) 'SPLIT_QT' 참조).

[0087] 바이너리 트리 분할은 현재 블록을 2개의 블록들로 분할하는 분할 기법을 나타낸다. 수직 방향을 따라(즉, 현재 블록을 가로지르는 수직선을 이용) 현재 블록을 두개의 블록들로 분할하는 것을 수직 방향 바이너리 트리 분할이라 호칭할 수 있고, 수평 방향을 따라(즉, 현재 블록을 가로지르는 수평선을 이용) 현재 블록을 두개의 블록들로 분할하는 것을 수평 방향 바이너리 트리 분할이라 호칭할 수 있다. 바이너리 트리 분할 결과, 현재 블록은 2개의 비정방 형태 파티션들로 분할될 수 있다. 도 4의 (b) 'SPLIT_BT_VER'는 수직 방향 바이너리 트리 분할 결과를 나타낸 것이고, 도 4의 (c) 'SPLIT_BT_HOR'는 수평 방향 바이너리 트리 분할 결과를 나타낸 것이다.

[0088] 트리플 트리 분할은 현재 블록을 3개의 블록들로 분할하는 분할 기법을 나타낸다. 수직 방향을 따라(즉, 현재 블록을 가로지르는 두개의 수직선을 이용) 현재 블록을 세개의 블록들로 분할하는 것을 수직 방향 트리플 트리 분할이라 호칭할 수 있고, 수평 방향을 따라(즉, 현재 블록을 가로지르는 두개의 수평선을 이용) 현재 블록을 세개의 블록들로 분할하는 것을 수평 방향 트리플 트리 분할이라 호칭할 수 있다. 트리플 트리 분할 결과, 현재 블록은 3개의 비정방 형태 파티션들로 분할될 수 있다. 이때, 현재 블록의 중앙에 위치하는 파티션의 너비/높이는 다른 파티션들의 너비/높이 대비 2배일 수 있다. 도 4의 (d) 'SPLIT_TT_VER'는 수직 방향 트리플 트리 분할 결과를 나타낸 것이고, 도 4의 (e) 'SPLIT_TT_HOR'는 수평 방향 트리플 트리 분할 결과를 나타낸 것이다.

[0089] 코딩 트리 유닛의 분할 횟수를 분할 깊이(Partitioning Depth)라 정의할 수 있다. 시퀀스 또는 꾹처 레벨에서 코딩 트리 유닛의 최대 분할 깊이가 결정될 수 있다. 이에 따라, 시퀀스 또는 필치별로 코딩 트리 유닛의 최대 분할 깊이가 상이할 수 있다.

[0090] 또는, 분할 기법들 각각에 대한 최대 분할 깊이를 개별적으로 결정할 수 있다. 일 예로, 쿼드 트리 분할이 허용되는 최대 분할 깊이는 바이너리 트리 분할 및/또는 트리플 트리 분할이 허용되는 최대 분할 깊이와 상이할 수 있다.

[0091] 부호화기는 현재 블록의 분할 형태 또는 분할 깊이 중 적어도 하나를 나타내는 정보를 비트스트림을 통해 시그널링할 수 있다. 복호화기는 비트스트림으로부터 파싱되는 상기 정보에 기초하여 코딩 트리 유닛의 분할 형태 및 분할 깊이를 결정할 수 있다.

[0092] 도 5는 코딩 트리 유닛의 분할 양상을 예시한 도면이다.

[0093] 쿼드 트리 분할, 바이너리 트리 분할 및/또는 트리플 트리 분할 등의 분할 기법을 이용하여 코딩 블록을 분할하는 것을 멀티 트리 분할(Multi Tree Partitioning)이라 호칭할 수 있다.

[0094] 코딩 블록에 멀티 트리 분할을 적용하여 생성되는 코딩 블록들을 하위 코딩 블록들이라 호칭할 수 있다. 코딩 블록의 분할 깊이가 k인 경우, 하위 코딩 블록들의 분할 깊이는 k+1로 설정된다.

[0095] 반대로, 분할 깊이가 k+1인 코딩 블록들에 대해, 분할 깊이가 k인 코딩 블록을 상위 코딩 블록이라 호칭할 수 있다.

[0096] 현재 코딩 블록의 분할 타입은 상위 코딩 블록의 분할 형태 또는 이웃 코딩 블록의 분할 타입 중 적어도 하나를 기초로 결정될 수 있다. 여기서, 이웃 코딩 블록은 현재 코딩 블록에 인접하는 것으로, 현재 코딩 블록의 상단 이웃 블록, 좌측 이웃 블록, 또는 좌측 상단 코너에 인접하는 이웃 블록 중 적어도 하나를 포함할 수 있다. 여기서, 분할 타입은, 쿼드 트리 분할 여부, 바이너리 트리 분할 여부, 바이너리 트리 분할 방향, 트리플 트리 분할 여부, 또는 트리플 트리 분할 방향 중 적어도 하나를 포함할 수 있다.

[0097] 코딩 블록의 분할 형태를 결정하기 위해, 코딩 블록이 분할되는지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그 'split_cu_flag'로, 상기 플래그가 참인 것은, 머리 트리 분할 기법에 의해 코딩 블록이 분할됨을 나타낸다.

- [0098] `split_cu_flag`가 참인 경우, 코딩 블록이 퀘드 트리 분할되는지 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그 `split_qt_flag`로, 상기 플래그가 참인 경우, 코딩 블록은 4개의 블록들로 분할될 수 있다.
- [0099] 일 예로, 도 5에 도시된 예에서는, 코딩 트리 유닛이 퀘드 트리 분할됨에 따라, 분할 깊이가 1인 4개의 코딩 블록들이 생성되는 것으로 도시되었다. 또한, 퀘드 트리 분할 결과로 생성된 4개의 코딩 블록들 중 첫번째 코딩 블록 및 네번째 코딩 블록에 다시 퀘드 트리 분할이 적용된 것으로 도시되었다. 그 결과, 분할 깊이가 2인 4개의 코딩 블록들이 생성될 수 있다.
- [0100] 또한, 분할 깊이가 2인 코딩 블록에 다시 퀘드 트리 분할을 적용함으로써, 분할 깊이가 3인 코딩 블록을 생성할 수 있다.
- [0101] 코딩 블록에 퀘드 트리 분할이 적용되지 않는 경우, 코딩 블록의 크기, 코딩 블록이 꾹처 경계에 위치하는지 여부, 최대 분할 깊이 또는 이웃 블록의 분할 형태 중 적어도 하나를 고려하여, 상기 코딩 블록에 바이너리 트리 분할 또는 트리플 트리 분할을 수행할 것인지 여부를 결정할 수 있다. 상기 코딩 블록에 바이너리 트리 분할 또는 트리플 트리 분할이 수행되는 것으로 결정된 경우, 분할 방향을 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그 `mtt_split_cu_vertical_flag`일 수 있다. 상기 플래그에 기초하여, 분할 방향이 수직 방향인지 또는 수평 방향인지 여부가 결정될 수 있다. 추가로, 바이너리 트리 분할 또는 트리플 트리 분할 중 어느 것이 상기 코딩 블록에 적용되는지를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그 `mtt_split_cu_binary_flag`일 수 있다. 상기 플래그에 기초하여, 상기 코딩 블록에 바이너리 트리 분할이 적용되는지 또는 트리플 트리 분할이 적용되는지 여부가 결정될 수 있다.
- [0102] 일 예로, 도 5에 도시된 예에서는, 분할 깊이 1인 코딩 블록에 수직 방향 바이너리 트리 분할이 적용되고, 상기 분할 결과로 생성된 코딩 블록들 중 좌측 코딩 블록에는 수직 방향 트리플 트리 분할이 적용되고, 우측 코딩 블록에는 수직 방향 바이너리 트리 분할이 적용된 것으로 도시되었다.
- [0104] 인터 예측은 이전 꾹처의 정보를 이용하여, 현재 블록을 예측하는 예측 부호화 모드이다. 일 예로, 이전 꾹처 내 현재 블록과 동일한 위치의 블록(이하, 콜로케이티드 블록, Collocated block)을 현재 블록의 예측 블록으로 설정할 수 있다. 이하, 현재 블록과 동일한 위치의 블록을 기초로 생성된 예측 블록을 콜로케이티드 예측 블록(Collocated Prediction Block)이라 호칭하기로 한다.
- [0105] 반면, 이전 꾹처에 존재한 오브젝트가 현재 꾹처에서는 다른 위치로 이동하였다면, 오브젝트의 움직임을 이용하여 효과적으로 현재 블록을 예측할 수 있다. 예를 들어, 이전 꾹처와 현재 꾹처를 비교함으로써 오브젝트의 이동 방향 및 크기를 알 수 있다면, 오브젝트의 움직임 정보를 고려하여 현재 블록의 예측 블록(또는, 예측 영상)을 생성할 수 있다. 이하, 움직임 정보를 이용하여 생성된 예측 블록을 움직임 예측 블록이라 호칭할 수 있다.
- [0106] 현재 블록에서 예측 블록을 차분하여, 잔차 블록(residual block)을 생성할 수 있다. 이때, 오브젝트의 움직임이 존재하는 경우라면, 콜로케이티드 예측 블록 대신 움직임 예측 블록을 이용함으로써, 잔차 블록의 에너지를 줄이고, 이에 따라, 잔차 블록의 압축 성능을 향상시킬 수 있다.
- [0107] 위처럼, 움직임 정보를 이용하여 예측 블록을 생성하는 것을 움직임 보상 예측이라 호칭할 수 있다. 대부분의 인터 예측에서는 움직임 보상 예측에 기초하여 예측 블록을 생성할 수 있다.
- [0108] 움직임 정보는 모션 벡터, 참조 꾹처 인덱스, 예측 방향 또는 양방향 가중치 인덱스 중 적어도 하나를 포함할 수 있다. 모션 벡터는 오브젝트의 이동 방향 및 크기를 나타낸다. 참조 꾹처 인덱스는 참조 꾹처 리스트에 포함된 참조 꾹처들 중 현재 블록의 참조 꾹처를 특정한다. 예측 방향은 단방향 L0 예측, 단방향 L1 예측 또는 양방향 예측(L0 예측 및 L1 예측) 중 어느 하나를 가리킨다. 현재 블록의 예측 방향에 따라, L0 방향의 움직임 정보 또는 L1 방향의 움직임 정보 중 적어도 하나가 이용될 수 있다. 양방향 가중치 인덱스는 L0 예측 블록에 적용되는 가중치 및 L1 예측 블록에 적용되는 가중치를 특정한다.
- [0109] 도 6은 본 발명의 일 실시예에 따른 인터 예측 방법의 흐름도이다.
- [0110] 도 6을 참조하면, 인터 예측 방법은, 현재 블록의 인터 예측 모드를 결정하는 단계(S601), 결정된 인터 예측 모드에 따라 현재 블록의 움직임 정보를 획득하는 단계(S602) 및 획득된 움직임 정보에 기초하여, 현재 블록에 대

한 움직임 보상 예측을 수행하는 단계(S603)를 포함한다.

[0111] 여기서, 인터 예측 모드는 현재 블록의 움직임 정보를 결정하기 위한 다양한 기법들을 나타내는 것으로, 병진(Translation) 움직임 정보를 이용하는 인터 예측 모드와, 어파인(Affine) 움직임 정보를 이용하는 인터 예측 모드를 포함할 수 있다. 일 예로, 병진 움직임 정보를 이용하는 인터 예측 모드는, 머지 모드 및 모션 벡터 예측 모드를 포함하고, 어파인 움직임 정보를 이용하는 인터 예측 모드는 어파인 머지 모드 및 어파인 모션 벡터 예측 모드를 포함할 수 있다. 현재 블록의 움직임 정보는, 인터 예측 모드에 따라, 현재 블록에 이웃하는 이웃 블록 또는 비트스트림으로부터 파싱되는 정보를 기초로 결정될 수 있다.

[0112] 이하, 어파인 움직임 정보를 이용하는 인터 예측 방법에 대해 상세히 설명하기로 한다.

[0113] 도 7은 오브젝트의 비선형 움직임을 예시한 도면이다.

[0114] 영상 내 물체의 움직임이 선형적이지 않은 움직임이 발생할 수 있다. 일 예로, 도 7에 도시된 예에서와 같이, 카메라 줌인(Zoom-in), 줌 아웃(Zoom-out), 회전(Rotation) 또는 어파인 변환 등 오브젝트의 비선형 움직임이 발생할 수 있다. 오브젝트의 비선형 움직임이 발생한 경우, 병진 움직임 벡터로는 오브젝트의 움직임을 효과적으로 표현할 수 없다. 이에 따라, 오브젝트의 비선형 움직임이 발생하는 부분에서는 병진 움직임 대신 어파인 움직임을 이용하여 부호화 효율을 향상시킬 수 있다.

[0115] 도 8은 본 발명의 일실시예 따른 어파인 모션에 기초한 인터 예측 방법의 흐름도이다.

[0116] 현재 블록에 어파인 모션에 기초한 인터 예측 기법이 적용되는지 여부는, 비트스트림으로부터 파싱되는 정보에 기초하여 결정될 수 있다. 구체적으로, 현재 블록에 어파인 머지 모드가 적용되는지 여부를 나타내는 플래그 또는 현재 블록에 어파인 모션 벡터 예측 모드가 적용되는지 여부를 나타내는 플래그 중 적어도 하나에 기초하여, 현재 블록에 어파인 모션에 기초한 인터 예측 기법이 적용되는지 여부를 결정할 수 있다.

[0117] 현재 블록에 어파인 모션에 기초한 인터 예측 기법이 적용되는 경우, 현재 블록의 어파인 모션 모델을 결정할 수 있다(S801). 어파인 모션 모델은 6 파라미터 어파인 모션 모델 또는 4 파라미터 어파인 모션 모델 중 적어도 하나로 결정될 수 있다. 6 파라미터 어파인 모션 모델은 6개의 파라미터를 이용하여 어파인 모션을 표현한 것이고, 4 파라미터 어파인 모션 모델은 4개의 파라미터를 이용하여 어파인 모션을 표현한 것이다.

[0118] 수학식 1은 6 파라미터를 이용하여 어파인 모션을 표현한 것이다. 어파인 모션은 어파인 시드 벡터들에 의해 결정되는 소정 영역에 대한 병진 움직임을 나타낸다.

수학식 1

$$v_x = ax - by + e$$

$$v_y = cx + dy + f$$

[0119] [0120] 6개의 파라미터를 이용하여 어파인 모션을 표현하는 경우, 복잡한 움직임을 표현할 수 있으나, 각 파라미터들을 부호화하는데 필요한 비트 수가 많아져 부호화 효율이 저하될 수 있다. 이에 따라, 4개의 파라미터를 이용하여 어파인 모션을 표현할 수도 있다. 수학식 2는 4 파라미터를 이용하여 어파인 모션을 표현한 것이다.

수학식 2

$$v_x = ax - by + e$$

$$v_y = bx + ay + f$$

[0121] [0122] 현재 블록의 어파인 모션 모델을 결정하기 위한 정보가 부호화되어 비트스트림을 통해 시그널링될 수 있다. 일 예로, 상기 정보는 1비트의 플래그 'affine_type_flag'일 수 있다. 상기 플래그의 값이 0인 것은 4 파라미터 어파인 모션 모델이 적용됨을 나타내고, 상기 플래그의 값이 1인 것은 6 파라미터 어파인 모션 모델이 적용됨을 나타낼 수 있다. 상기 플래그는, 슬라이스, 타일 또는 블록(예컨대, 코딩 블록 또는 코딩 트리 유닛) 단위로 부호화될 수 있다. 슬라이스 레벨에서 플래그가 시그널링되는 경우, 상기 슬라이스 레벨에서 결정된 어파인 모션

모델이 상기 슬라이스에 속한 블록들 모두에 적용될 수 있다.

[0123] 또는, 현재 블록의 어파인 인터 예측 모드에 기초하여, 현재 블록의 어파인 모션 모델을 결정할 수 있다. 일 예로, 어파인 머지 모드가 적용되는 경우, 현재 블록의 어파인 모션 모델을 4 파라미터 모션 모델로 결정할 수 있다. 반면, 어파인 모션 벡터 예측 모드가 적용되는 경우, 현재 블록의 어파인 모션 모델을 결정하기 위한 정보를 부호화하여 비트스트림을 통해 시그널링될 수 있다. 일 예로, 현재 블록에 어파인 모션 벡터 예측 모드가 적용되는 경우, 1비트의 플래그 'affine_type_flag'에 기초하여, 현재 블록의 어파인 모션 모델을 결정할 수 있다.

[0124] 다음으로, 현재 블록의 어파인 시드 벡터를 유도할 수 있다(S802). 4 파라미터 어파인 모션 모델이 선택된 경우, 현재 블록의 두 컨트롤 포인트에서의 모션 벡터들을 유도할 수 있다. 반면, 6 파라미터 어파인 모션 모델이 선택된 경우, 현재 블록의 세 컨트롤 포인트에서의 모션 벡터들이 유도될 수 있다. 컨트롤 포인트에서의 모션 벡터를 어파인 시드 벡터라 호칭할 수 있다. 컨트롤 포인트는, 현재 블록의 좌상단 코너, 우상단 코너 또는 좌하단 코너 중 적어도 하나를 포함할 수 있다.

[0125] 도 9는 어파인 모션 모델 별 어파인 시드 벡터를 예시한 도면이다.

[0126] 4 파라미터 어파인 모션 모델에서는 좌상단 코너, 우상단 코너 또는 좌하단 코너 중 두개에 대한 어파인 시드 벡터들을 유도할 수 있다. 일 예로, 도 9의 (a)에 도시된 예에서와 같이, 4 파라미터 어파인 모션 모델이 선택된 경우, 현재 블록의 좌상단 코너(예컨대, 좌상단 샘플 (x1, y1))에 대한 어파인 시드 벡터 sv₀와 현재 블록의 우상단 코너(예컨대, 우상단 샘플 (x1, y1))에 대한 어파인 시드 벡터 sv₁을 이용하여 어파인 벡터를 유도할 수 있다. 좌상단 코너에 대한 어파인 시드 벡터 대신 좌하단 코너에 대한 어파인 시드 벡터를 사용하거나, 우상단 코너에 대한 어파인 시드 벡터 대신 좌하단 코너에 대한 어파인 시드 벡터를 사용하는 것도 가능하다.

[0127] 6 파라미터 어파인 모션 모델에서는 좌상단 코너, 우상단 코너 및 좌하단 코너에 대한 어파인 시드 벡터들을 유도할 수 있다. 일 예로, 도 9의 (b)에 도시된 예에서와 같이, 6 파라미터 어파인 모션 모델이 선택된 경우, 현재 블록의 좌상단 코너(예컨대, 좌상단 샘플 (x1, y1))에 대한 어파인 시드 벡터 sv₀, 현재 블록의 우상단 코너(예컨대, 우상단 샘플 (x1, y1))에 대한 어파인 시드 벡터 sv₁ 및 현재 블록의 좌상단 코너(예컨대, 좌상단 샘플 (x2, y2))에 대한 어파인 시드 벡터 sv₂를 이용하여 어파인 벡터를 유도할 수 있다.

[0128] 후술되는 실시예에서는, 4 파라미터 어파인 모션 모델 하에서, 좌상단 컨트롤 포인트 및 우상단 컨트롤 포인트의 어파인 시드 벡터들을 각각 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터라 호칭하기로 한다. 후술되는 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 이용하는 실시예들에서, 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터 중 적어도 하나는 좌하단 컨트롤 포인트의 어파인 시드 벡터(제3 어파인 시드 벡터) 또는 우하단 컨트롤 포인트의 어파인 시드 벡터(제4 어파인 시드 벡터)로 대체될 수 있다.

[0129] 또한, 6 파라미터 어파인 모션 모델 하에서, 좌상단 컨트롤 포인트, 우상단 컨트롤 포인트 및 좌하단 컨트롤 포인트의 어파인 시드 벡터들을 각각 제1 어파인 시드 벡터, 제2 어파인 시드 벡터 및 제3 어파인 시드 벡터라 호칭하기로 한다. 후술되는 제1 어파인 시드 벡터, 제2 어파인 시드 벡터 및 제3 어파인 시드 벡터를 이용하는 실시예들에서, 제1 어파인 시드 벡터, 제2 어파인 시드 벡터 및 제3 어파인 시드 벡터 중 적어도 하나는 우하단 컨트롤 포인트의 어파인 시드 벡터(제4 어파인 시드 벡터)로 대체될 수 있다.

[0130] 어파인 시드 벡터들을 이용하여 서브 블록 별로 어파인 벡터를 유도할 수 있다(S803). 여기서, 어파인 벡터는 어파인 시드 벡터들을 기초로 유도되는 병진 모션 벡터를 나타낸다. 서브 블록의 어파인 벡터를 어파인 서브 블록 모션 벡터 또는 서브 블록 모션 벡터라 호칭할 수 있다.

[0131] 도 10은 4 파라미터 모션 모델 하에서 서브 블록들의 어파인 벡터들을 예시한 도면이다.

[0132] 서브 블록의 어파인 벡터는 컨트롤 포인트의 위치, 서브 블록의 위치 및 어파인 시드 벡터를 기초로 유도될 수 있다. 일 예로, 수학식 3은 어파인 서브 블록 벡터를 유도하는 예를 나타낸다.

수학식 3

$$v_x = \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)} (x - x_0) - \frac{(sv_{1y} - sv_{0y})}{(x_1 - x_0)} (y - y_0) + sv_{0x}$$

$$v_y = \frac{(sv_{1y} - sv_{0y})}{(x_1 - x_0)} (x - x_0) - \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)} (y - y_0) + sv_{0y}$$

[0133]

[0134] 상기 수학식 3에서 (x, y) 는 서브 블록의 위치를 나타낸다. 여기서, 서브 블록의 위치는 서브 블록에 포함된 기준 샘플의 위치를 나타낸다. 기준 샘플은 서브 블록의 좌상단 코너에 위치하는 샘플, 또는 x 축 또는 y 축 좌표 중 적어도 하나가 중앙 위치인 샘플일 수 있다. (x_0, y_0) 는 제1 컨트롤 포인트의 위치를 나타내고, (sv_{0x}, sv_{0y}) 는 제1 어파인 시드 벡터를 나타낸다. 또한, (x_1, y_1) 은 제2 컨트롤 포인트의 위치를 나타내고, (sv_{1x}, sv_{1y}) 는 제2 어파인 시드 벡터를 나타낸다.

[0135]

제1 컨트롤 포인트 및 제2 컨트롤 포인트가 각각 현재 블록의 좌상단 코너 및 우상단 코너에 대응하는 경우, $x_1 - x_0$ 는 현재 블록의 너비와 동일한 값으로 설정될 수 있다.

[0136]

[0136] 이후, 각 서브 블록의 어파인 벡터를 이용하여, 각 서브 블록에 대한 움직임 보상 예측을 수행할 수 있다 (S804). 움직임 보상 예측의 수행 결과, 각 서브 블록에 대한 예측 블록이 생성될 수 있다. 서브 블록들의 예측 블록들이, 현재 블록의 예측 블록으로 설정될 수 있다.

[0137]

[0137] 현재 블록의 어파인 시드 벡터는 현재 블록에 이웃하는 이웃 블록의 어파인 시드 벡터를 기초로 유도될 수 있다. 현재 블록의 인터 예측 모드가 어파인 머지 모드인 경우, 머지 후보 리스트에 포함된 머지 후보의 어파인 시드 벡터를 현재 블록의 어파인 시드 벡터로 결정할 수 있다. 또한, 현재 블록의 인터 예측 모드가 어파인 머지 모드인 경우, 현재 블록의 참조 픽처 인덱스, 특정 방향 예측 플래그 또는 양방향 가중치 중 적어도 하나를 포함하는 움직임 정보도 머지 후보와 동일하게 설정될 수 있다.

[0138]

[0138] 머지 후보는 현재 블록의 이웃 블록을 기초로 유도될 수 있다. 상기 이웃 블록은 현재 블록에 공간적으로 인접하는 공간적 이웃 블록 및 현재 픽처와 상이한 픽처에 포함된 시간적 이웃 블록 중 적어도 하나를 포함할 수 있다.

[0139]

[0139] 도 11은 머지 후보를 유도하는데 이용될 수 있는 이웃 블록을 도시한 도면이다.

[0140]

[0140] 현재 블록의 이웃 블록은, 현재 블록의 좌측에 인접하는 이웃 블록(A), 현재 블록의 상단에 이웃하는 이웃 블록(B), 현재 블록의 우측 상단 코너에 인접하는 이웃 블록(C), 현재 블록의 좌측 하단 코너에 인접하는 이웃 블록(D) 또는 현재 블록의 좌측 상단 코너에 인접하는 이웃 블록 중 적어도 하나를 포함할 수 있다. 현재 블록의 좌측 상단 샘플의 좌표가 (x_0, y_0) 인 경우, 좌측 이웃 블록 A는 (x_0-1, y_0+H-1) 위치의 샘플을 포함하고, 상단 이웃 블록 B는 (x_0+W-1, y_0-1) 위치의 샘플을 포함한다. 여기서, W 및 H 는 각각 현재 블록의 너비 및 높이를 나타낸다. 우측 상단 이웃 블록 C는 (x_0+W, y_0-1) 위치의 샘플을 포함하고, 좌측 하단 이웃 블록 D는 (x_0-1, y_0+H) 위치의 샘플을 포함한다. 좌측 상단 이웃 블록 E는 (x_0-1, y_0-1) 위치의 샘플을 포함한다.

[0141]

[0141] 이웃 블록이 어파인 인터 예측 모드로 부호화된 경우, 해당 이웃 블록의 어파인 시드 벡터를 기초로, 머지 후보의 어파인 시드 벡터를 유도할 수 있다. 이하, 어파인 인터 예측 모드로 부호화된 이웃 블록을 어파인 이웃 블록이라 호칭하기로 한다.

[0142]

[0142] 이웃 블록들을 기 정의된 스캔 순서로 탐색하여, 현재 블록에 대한 머지 후보를 생성할 수 있다. 상기 스캔 순서는 부호화기 및 복호화기에서 기 정의되어 있을 수 있다. 일 예로, A, B, C, D, E 순서로 이웃 블록을 탐색할 수 있다. 그리고, 탐색된 어파인 이웃 블록들로부터 순차적으로 머지 후보를 유도할 수 있다. 또는, 스캔 순서는 현재 블록의 크기, 형태 또는 어파인 모션 모델 중 적어도 하나에 기초하여 적응적으로 결정될 수 있다. 즉, 크기, 형태 또는 어파인 모션 모델 중 적어도 하나가 상이한 블록들에 대한 스캔 순서는 서로 상이할 수 있다.

[0143]

[0143] 또는, 현재 블록의 상단에 위치하는 블록들을 순차적으로 탐색하여, 첫번째로 발견된 어파인 이웃 블록으로부터 하나의 머지 후보를 유도하고, 현재 블록의 좌측에 위치하는 블록들을 순차적으로 탐색하여, 첫번째로 발견된 어파인 이웃 블록으로부터 하나의 머지 후보를 유도할 수 있다. 여기서, 현재 블록의 상단에 위치하는 이웃 블

록들은, 이웃 블록 E, 이웃 블록 B 또는 이웃 블록 C 중 적어도 하나를 포함하고, 현재 블록의 좌측에 위치하는 블록들은, 블록 A 또는 블록 D 중 적어도 하나를 포함할 수 있다. 이때, 이웃 블록 E를 현재 블록의 좌측에 위치하는 블록으로 분류하는 것도 가능하다.

[0144] 도시되지는 않았지만, 현재 블록의 시간적 이웃 블록으로부터 머지 후보를 유도할 수도 있다. 여기서, 시간적 이웃 블록은, 콜로케이티드 꽉쳐 내 현재 블록과 동일 위치의 블록 또는 이에 인접하는 블록을 포함할 수 있다. 구체적으로, 현재 블록의 시간적 이웃 블록이 어파인 인터 예측 모드로 부호화되었다면, 시간적 머지 후보의 어파인 시드 벡터에 기초하여, 머지 후보를 유도할 수 있다.

[0145] 머지 후보들을 포함하는 머지 후보 리스트를 생성하고, 머지 후보 리스트에 포함된 머지 후보들 중 하나의 어파인 시드 벡터를 현재 블록의 어파인 시드 벡터로 결정할 수 있다. 이를 위해, 머지 후보들 중 어느 하나를 식별하는 인덱스 정보가 부호화되어 비트스트림을 통해 전송될 수 있다.

[0146] 다른 예로, 이웃 블록들을 스캔 순서에 따라 탐색하되, 첫번째로 발견된 어파인 이웃 블록의 어파인 시드 벡터로부터 현재 블록의 어파인 시드 벡터를 유도할 수도 있다.

[0147] 위와 같이, 어파인 머지 모드 하에서, 현재 블록의 어파인 시드 벡터는 이웃 블록의 어파인 시드 벡터를 이용하여 유도될 수 있다.

[0148] 현재 블록의 인터 예측 모드가 어파인 모션 벡터 예측 모드인 경우, 모션 벡터 예측 후보 리스트에 포함된 모션 벡터 예측 후보의 어파인 시드 벡터를 현재 블록의 어파인 시드 벡터 예측값으로 결정할 수 있다. 상기 어파인 시드 벡터 예측값에 어파인 시드 벡터 차분값을 더하여, 현재 블록의 어파인 시드 벡터를 유도할 수 있다.

[0149] 어파인 시드 벡터 예측 후보는 현재 블록의 이웃 블록을 기초로 유도될 수 있다. 구체적으로, 현재 블록의 상단에 위치하는 이웃 블록들을 소정 스캔 순서에 따라 탐색하고, 첫번째로 발견된 어파인 이웃 블록으로부터 제1 어파인 시드 벡터 예측 후보를 유도할 수 있다. 또한, 현재 블록의 좌측에 위치하는 이웃 블록들을 소정 스캔 순서에 따라 탐색하고, 첫번째로 발견된 어파인 이웃 블록으로부터 제2 어파인 시드 벡터 예측 후보를 유도할 수 있다.

[0150] 어파인 시드 벡터 차분값을 결정하기 위한 정보가 부호화되어 비트스트림을 통해 전송될 수 있다. 상기 정보는 어파인 시드 벡터 차분값의 크기를 나타내는 크기 정보 및 어파인 시드 벡터 차분값의 부호를 나타내는 부호 정보를 포함할 수 있다. 각 컨트롤 포인트에 대한 어파인 시드 벡터 차분값은 동일한 값으로 설정될 수 있다. 또는, 컨트롤 포인트 별로 어파인 시드 벡터 차분값이 상이하게 설정될 수 있다.

[0151] 상술한 바와 같이, 어파인 이웃 블록의 어파인 시드 벡터로부터 머지 후보 또는 어파인 시드 벡터 예측 후보의 어파인 시드 벡터를 유도하고, 유도된 머지 후보 또는 어파인 시드 벡터 예측 후보의 어파인 시드 벡터를 이용하여, 현재 블록의 어파인 시드 벡터를 유도할 수 있다. 또는, 어파인 이웃 블록들을 소정의 스캔 순서에 따라 탐색한 뒤, 첫번째로 발견된 어파인 이웃 블록의 어파인 시드 벡터로부터 현재 블록의 어파인 시드 벡터를 유도할 수도 있다.

[0152] 이하, 어파인 이웃 블록의 어파인 시드 벡터로부터 현재 블록, 머지 후보 또는 어파인 시드 벡터 예측 후보의 어파인 시드 벡터를 유도하는 방법에 대해 상세히 설명하기로 한다. 후술되는 실시예들에서, 현재 블록의 어파인 시드 벡터를 유도하는 것은, 머지 후보의 어파인 시드 벡터 유도하는 것 또는 어파인 시드 벡터 예측 후보의 어파인 시드 벡터를 유도하는 것으로도 이해될 수 있다.

[0153] 도 12는 어파인 이웃 블록의 어파인 시드 벡터에 기초하여 현재 블록의 어파인 시드 벡터를 유도하는 예를 나타낸 도면이다.

[0154] 어파인 이웃 블록에 대해 좌상단 컨트롤 포인트에 대한 제1 어파인 시드 벡터 nv_0 및 우상단 컨트롤 포인트에 대한 제2 어파인 시드 벡터 nv_1 가 저장되어 있다면, 상기 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 기초로, 어파인 이웃 블록의 좌하단 컨트롤 포인트에 대한 제3 어파인 시드 벡터 nv_2 를 유도할 수 있다. 수학식 4는 제3 어파인 시드 벡터를 유도하는 예를 나타낸다.

수학식 4

$$[0155] \quad nv_{2x} = \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_{n2} - x_{n0}) - \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (y_{n2} - y_{n0}) + nv_{0x}$$

$$nv_{2y} = \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_{n2} - x_{n0}) - \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (y_{n2} - y_{n0}) + nv_{0y}$$

- [0156] 상기 수학식 4에서, (nv_{0x}, nv_{0y}) 은 제1 어파인 시드 벡터 nv_0 를, (nv_{1x}, nv_{1y}) 는 제2 어파인 시드 벡터 nv_1 를, (nv_{2x}, nv_{2y}) 는 제3 어파인 시드 벡터 nv_2 를 나타낸다. 또한, (x_{n0}, x_{n1}) 은 제1 컨트롤 포인트의 위치, (x_{n1}, x_{n2}) 는 제2 컨트롤 포인트의 위치를, (x_{n2}, x_{n0}) 는 제3 컨트롤 포인트의 위치를 나타낸다.

- [0157] 이후, 제1 어파인 시드 벡터, 제2 어파인 시드 벡터 및 제3 어파인 시드 벡터를 이용하여, 현재 블록의 어파인 시드 벡터를 유도할 수 있다. 수학식 5는 현재 블록의 제1 어파인 시드 벡터 v_0 를 유도하는 예를 나타내고, 수학식 6은 현재 블록의 제2 어파인 시드 벡터 v_1 를 유도하는 예를 나타낸다.

수학식 5

$$[0158] \quad v_{0x} = \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_0 - x_{n0}) - \frac{(nv_{2x} - nv_{0x})}{(y_{n2} - y_{n0})} (y_0 - y_{n0}) + nv_{0x}$$

$$v_{0y} = \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_0 - x_{n0}) - \frac{(nv_{2y} - nv_{0y})}{(y_{n2} - y_{n0})} (y_0 - y_{n0}) + nv_{0y}$$

수학식 6

$$[0159] \quad v_{1x} = \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_1 - x_{n0}) - \frac{(nv_{2x} - nv_{0x})}{(y_{n2} - y_{n0})} (y_1 - y_{n0}) + nv_{0x}$$

$$v_{1y} = \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_1 - x_{n0}) - \frac{(nv_{2y} - nv_{0y})}{(y_{n2} - y_{n0})} (y_1 - y_{n0}) + nv_{0y}$$

- [0160] 수학식 5 및 수학식 6에서, (v_{0x}, v_{0y}) 은 현재 블록의 제1 어파인 시드 벡터 sv_0 를 나타내고, (v_{1x}, v_{1y}) 는 현재 블록의 제2 어파인 시드 벡터 sv_1 를 나타낸다. 또한, (x_0, y_0) 은 제1 컨트롤 포인트의 위치를 나타내고, (x_1, y_1) 는 제2 컨트롤 포인트의 위치를 나타낸다. 일 예로, 제1 컨트롤 포인트는 현재 블록의 좌측 상단 코너를 나타내고, 제2 컨트롤 포인트는 현재 블록의 우측 상단 코너를 나타낸다.

- [0161] 상술한 예에서는 어파인 이웃 블록에 대한 3개의 어파인 시드 벡터들을 이용하여, 현재 블록의 어파인 시드 벡터들이 유도되는 것으로 설명하였다. 다른 예로, 어파인 이웃 블록의 어파인 시드 벡터들 중 2개만을 이용하여, 현재 블록의 어파인 시드 벡터들을 유도할 수 있다.

- [0162] 또는, 어파인 이웃 블록에 대한 좌측 상단 코너의 제1 어파인 시드 벡터, 우측 상단 코너의 제2 어파인 시드 벡터 또는 좌측 하단 코너의 제3 어파인 시드 벡터 대신 우측 하단 코너에 대한 제4 어파인 시드 벡터를 이용하여 현재 블록의 어파인 시드 벡터들을 유도할 수 있다.

- [0163] 특히, 현재 블록의 상단 경계가 코딩 트리 유닛의 상단 경계와 접하고, 현재 블록의 상단에 이웃하는 어파인 이웃 블록(이하, 상단 어파인 이웃 블록이라 함)의 상단 컨트롤 포인트(예컨대, 좌측 상단 코너 또는 우측 상단 코너)에 대한 어파인 시드 벡터를 이용하고자 하는 경우, 이들을 미리 메모리에 저장해야 하는 바, 라인 베퍼의

개수가 증가하는 문제가 발생할 수 있다. 이에 따라, 현재 블록의 상단 경계가 코딩 트리 유닛의 상단 경계에 접하는 경우, 상단 어파인 이웃 블록에 대해서는, 상단 컨트롤 포인트의 어파인 시드 벡터 대신 하단 컨트롤 포인트(예컨대, 좌측 하단 코너 또는 우측 하단 코너)의 어파인 시드 벡터를 이용하도록 설정할 수 있다. 일 예로, 상단 어파인 이웃 블록의 좌측 하단 코너에 대한 제3 어파인 시드 벡터 및 우측 하단 코너에 대한 제4 어파인 시드 벡터를 이용하여, 현재 블록의 어파인 시드 벡터들을 유도할 수 있다. 이때, 하단 코너에 대한 어파인 시드 벡터는 상단 코너에 대한 어파인 시드 벡터를 복사하거나, 상단 코너들에 대한 어파인 시드 벡터들로부터 유도될 수 있다. 일 예로, 제1 어파인 시드 벡터, 제2 어파인 시드 벡터 또는 제3 어파인 시드 벡터를 우측 하단 코너에 대한 제4 어파인 시드 벡터로 변환/치환하여 사용할 수 있다.

[0164] 수학식 7 및 수학식 8은 이웃 어파인 벡터의 좌측 하단 컨트롤 포인트에 대한 제3 어파인 시드 벡터 및 우측 하단 컨트롤 포인트에 대한 제4 어파인 시드 벡터를 이용하여, 현재 블록의 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 유도하는 예를 나타낸 것이다.

수학식 7

$$v_{0x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2x}$$

$$v_{0y} = \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2y}$$

[0165]

수학식 8

$$v_{1x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2x}$$

$$v_{1y} = \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2y}$$

[0166]

[0167] 수학식 7 및 8에서, (x_{n2}, y_{n2}) 는 어파인 이웃 블록의 좌측 하단 컨트롤 포인트의 좌표를 나타내고, (x_{n3}, y_{n3}) 는 어파인 이웃 블록의 우측 하단 컨트롤 포인트의 좌표를 나타낸다. (x_0, y_0) 는 현재 블록의 좌측 상단 컨트롤 포인트의 좌표를 나타내고, (x_1, y_1) 은 현재 블록의 우측 상단 컨트롤 포인트의 좌표를 나타낸다. (nv_{2x}, nv_{2y}) 는 어파인 이웃 블록의 좌측 하단 컨트롤 포인트의 어파인 시드 벡터(즉, 제3 어파인 시드 벡터)를 나타내고, (nv_{3x}, nv_{3y}) 는 어파인 이웃 블록의 우측 하단 컨트롤 포인트의 어파인 시드 벡터(즉, 제4 어파인 시드 벡터)를 나타낸다. (v_{0x}, v_{0y}) 는 현재 블록의 좌측 상단 컨트롤 포인트의 어파인 시드 벡터(즉, 제1 어파인 시드 벡터)를 나타내고, (v_{1x}, v_{1y}) 는 현재 블록의 우측 상단 컨트롤 포인트의 어파인 시드 벡터(즉, 제2 어파인 시드 벡터)를 나타낸다.

[0168] 수학식 7 및 수학식 8에 포함된 나누기 연산을 비트 시프트 연산으로 변경할 수도 있다. 비트 시프트 연산은 좌측 하단 컨트롤 포인트 및 우측 하단 컨트롤 포인트 사이의 너비(즉, $(x_{n3} - x_{n2})$)에 의해 유도되는 값을 기초로 수행될 수 있다.

[0169] 상술한 예에서와 같이, 현재 블록의 어파인 시드 벡터들은 부호화/복호화가 완료된 어파인 이웃 블록의 어파인 시드 벡터들로부터 유도될 수 있다. 이를 위해, 부호화/복호화가 완료된 어파인 이웃 블록의 어파인 시드 벡터들을 메모리에 저장해야 한다. 다만, 어파인 이웃 블록에 포함된 서브 블록들의 병진 모션 벡터들(즉, 어파인 벡터들)에 더하여 어파인 이웃 블록의 어파인 시드 벡터들을 메모리에 저장하게됨에 따라, 메모리 사용량이 증가하는 문제점이 있다. 이와 같은 문제점을 해소하기 위해, 어파인 이웃 블록의 어파인 시드 벡터 대신, 어파인 이웃 블록의 컨트롤 포인트에 인접한 서브 블록의 모션 벡터를 이용하여 현재 블록의 어파인 시드 벡터를 유도할 수 있다. 즉, 어파인 이웃 블록의 컨트롤 포인트에 인접한 서브 블록의 움직임 벡터를 어파인 이웃 블록의

어파인 시드 벡터로 설정할 수 있다. 여기서, 서브 블록은 부호화기 및 복호화기에서 기 정의된 크기/형태를 갖는 블록으로, 모션 벡터가 저장되는 기본 크기/형태를 갖는 블록일 수 있다. 일 예로, 서브 블록은 4x4 크기의 정방형 블록일 수 있다. 또는, 특정 샘플 위치에 대한 모션 벡터를 어파인 이웃 블록의 어파인 시드 벡터로 설정할 수 있다.

[0170] 도 13은 서브 블록의 모션 벡터를 어파인 이웃 블록의 어파인 시드 벡터로 설정하는 예를 나타낸 도면이다.

[0171] 컨트를 포인트에 인접하는 서브 블록의 움직임 벡터를 해당 컨트를 포인트의 어파인 시드 벡터로 설정할 수 있다. 일 예로, 도 13에 도시된 예에서, 어파인 이웃 블록의 좌측 하단 코너에 인접하는 서브 블록(좌측 하단 서브 블록)의 움직임 벡터 (nv_{4x} , nv_{4y})를 좌측 하단 컨트를 포인트의 어파인 시드 벡터 (nv_{2x} , nv_{2y})로 설정하고, 우측 하단 코너에 인접하는 서브 블록(우측 하단 서브 블록)의 움직임 벡터 (nv_{5x} , nv_{5y})를 우측 하단 코너의 컨트를 포인트의 어파인 시드 벡터 (nv_{3x} , nv_{3y})으로 설정할 수 있다. 여기서, 좌측 하단 서브 블록은 이웃 어파인 블록 내 좌측 하단 컨트를 포인트 (x_{n2} , y_{n2})에 인접하는 샘플(예컨대, $(x_{n2}, y_{n2}-1)$ 위치의 샘플)을 포함하는 서브 블록을 의미하고, 우측 하단 서브 블록은 이웃 어파인 블록 내 우측 하단 컨트를 포인트 (x_{n3} , y_{n3})에 인접하는 샘플(예컨대, $(x_{n3}-1, y_{n3}-1)$ 위치의 샘플)을 포함하는 블록을 의미할 수 있다. 수학식 7 및 수학식 8을 기초로 현재 블록의 어파인 시드 벡터들을 유도하는 경우, 어파인 이웃 블록의 제3 어파인 시드 벡터는 좌측 하단 서브 블록의 모션 벡터로 대체되고, 제4 어파인 시드 벡터는 우측 하단 서브 블록의 모션 벡터로 대체될 수 있다.

[0172] 이하, 후술되는 실시예에서는, 어파인 이웃 블록의 어파인 시드 벡터로 이용되는 서브 블록을, 어파인 서브 블록이라 호칭하기로 한다.

[0173] 본 발명의 일 실시예에 따르면, 특정 위치의 샘플을 기준으로 어파인 서브 블록을 결정할 수 있다. 일 예로, 특정 위치의 샘플을 포함하는 서브 블록을 어파인 서브 블록으로 설정할 수도 있다. 이하, 특정 위치의 샘플을 어파인 기준 샘플이라 호칭하기로 한다. 아울러, 좌측 하단 컨트를 포인트의 어파인 서브 블록을 결정하는데 이용되는 기준 샘플을 좌측 하단 기준 샘플이라 호칭하고, 우측 하단 컨트를 포인트의 어파인 서브 블록을 결정하는데 이용되는 기준 샘플을 우측 하단 기준 샘플이라 호칭하기로 한다.

[0174] 좌측 하단 기준 샘플 및 우측 하단 기준 샘플은 어파인 이웃 블록에 포함된 샘플들로부터 선택될 수 있다. 일 예로, 좌측 하단 서브 블록의 좌상단 샘플, 좌하단 샘플, 우상단 샘플 또는 좌하단 샘플 중 적어도 하나를 좌측 하단 기준 샘플로 설정하고, 우측 하단 서브 블록의 좌상단 샘플, 좌하단 샘플, 우상단 샘플 또는 좌하단 샘플 중 적어도 하나를 우측 하단 기준 샘플로 설정할 수 있다. 이에 따라, 좌측 하단 기준 샘플을 포함하는 좌측 하단 서브 블록 및 우측 하단 기준 샘플을 포함하는 우측 하단 서브 블록의 모션 벡터들이 각각 좌측 하단 컨트를 포인트에 대한 어파인 시드 벡터 및 우측 하단 포인트에 대한 어파인 시드 벡터로 설정될 수 있다.

[0175] 다른 예로, 좌측 하단 기준 샘플 또는 우측 하단 기준 샘플 중 적어도 하나가 어파인 이웃 블록의 바깥에 위치하는 샘플로 설정될 수 있다. 이에 대해서는, 도 14 내지 도 16을 참조하여 상세히 설명하기로 한다.

[0176] 도 14 내지 도 16은 기준 샘플의 위치를 나타낸 도면이다.

[0177] 도 14의 (a)에 도시된 예에서와 같이, 좌측 하단 컨트를 포인트에 대해서는 좌측 하단 서브 블록의 좌상단 샘플을 기준 샘플 (x_{n4} , y_{n4})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n4} , y_{n4})를 포함하는 좌측 하단 서브 블록이 좌측 하단 컨트를 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0178] 우측 하단 컨트를 포인트에 대해서는 우측 하단 서브 블록의 우상단 샘플의 우측에 위치하는 샘플을 기준 샘플 (x_{n5} , y_{n5})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n5} , y_{n5})를 포함하는 우측 하단 서브 블록의 우측에 이웃하는 서브 블록이 우측 하단 컨트를 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0179] 또는, 도 14의 (b)에 도시된 예에서와 같이, 좌측 하단 컨트를 포인트에 대해서는 좌측 하단 서브 블록의 좌상단 샘플의 좌측에 위치하는 샘플을 기준 샘플 (x_{n4} , y_{n4})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n4} , y_{n4})를 포함하는 좌측 하단 서브 블록의 좌측에 이웃하는 서브 블록이 좌측 하단 컨트를 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0180] 우측 하단 컨트를 포인트에 대해서는 우측 하단 서브 블록의 우상단 샘플을 기준 샘플 (x_{n5} , y_{n5})로 설정할 수 있

다. 이에 따라, 상기 기준 샘플 (x_{n5} , y_{n5})를 포함하는 우측 하단 서브 블록이 우측 하단 컨트롤 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0181] 또는, 도 15의 (a)에 도시된 예에서와 같이, 좌측 하단 컨트롤 포인트에 대해서는 좌측 하단 서브 블록의 좌하단 샘플을 기준 샘플 (x_{n4} , y_{n4})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n4} , y_{n4})를 포함하는 좌측 하단 서브 블록이 좌측 하단 컨트롤 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0182] 우측 하단 컨트롤 포인트에 대해서는 우측 하단 서브 블록의 우하단 샘플의 우측에 위치하는 샘플을 기준 샘플 (x_{n5} , y_{n5})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n5} , y_{n5})를 포함하는 우측 하단 서브 블록의 우측에 이웃하는 서브 블록이 우측 하단 컨트롤 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0183] 또는, 도 15의 (b)에 도시된 예에서와 같이, 좌측 하단 컨트롤 포인트에 대해서는 좌측 하단 서브 블록의 좌하단 샘플의 좌측에 위치하는 샘플을 기준 샘플 (x_{n4} , y_{n4})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n4} , y_{n4})를 포함하는 좌측 하단 서브 블록의 좌측에 이웃하는 서브 블록이 좌측 하단 컨트롤 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0184] 우측 하단 컨트롤 포인트에 대해서는 우측 하단 서브 블록의 우하단 샘플을 기준 샘플 (x_{n5} , y_{n5})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n5} , y_{n5})를 포함하는 우측 하단 서브 블록이 우측 하단 컨트롤 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0185] 또는, 도 16의 (a)에 도시된 예에서와 같이, 좌측 하단 컨트롤 포인트에 대해서는 좌측 하단 서브 블록의 좌상단 샘플 및 좌하단 샘플 사이에 위치하는 샘플(예컨대, 좌중간 샘플)을 기준 샘플 (x_{n4} , y_{n4})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n4} , y_{n4})를 포함하는 좌측 하단 서브 블록이 좌측 하단 컨트롤 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0186] 우측 하단 컨트롤 포인트에 대해서는 우측 하단 서브 블록의 우상단 샘플과 우하단 샘플 사이에 위치하는 샘플(예컨대, 우중간 샘플)의 우측에 위치하는 샘플을 기준 샘플 (x_{n5} , y_{n5})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n5} , y_{n5})를 포함하는 우측 하단 서브 블록의 우측에 이웃하는 서브 블록이 우측 하단 컨트롤 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0187] 또는, 도 16의 (b)에 도시된 예에서와 같이, 좌측 하단 컨트롤 포인트에 대해서는 좌측 하단 서브 블록의 좌상단 샘플 및 좌하단 샘플 사이에 위치하는 샘플의 좌측에 위치하는 샘플을 기준 샘플 (x_{n4} , y_{n4})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n4} , y_{n4})를 포함하는 좌측 하단 서브 블록의 좌측에 이웃하는 서브 블록이 좌측 하단 컨트롤 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0188] 우측 하단 컨트롤 포인트에 대해서는 우측 하단 서브 블록의 우상단 샘플 및 우하단 샘플 사이에 위치하는 샘플을 기준 샘플 (x_{n5} , y_{n5})로 설정할 수 있다. 이에 따라, 상기 기준 샘플 (x_{n5} , y_{n5})를 포함하는 우측 하단 서브 블록이 우측 하단 컨트롤 포인트에 대한 어파인 서브 블록으로 설정될 수 있다.

[0189] 수학식 7 및 수학식 8을 기초로 현재 블록의 어파인 시드 벡터들을 유도하는 경우, 어파인 이웃 블록의 제3 어파인 시드 벡터는 좌측 하단 컨트롤 포인트에 대한 어파인 서브 블록의 모션 벡터로 대체되고, 제4 어파인 시드 벡터는 우측 하단 컨트롤 포인트에 대한 어파인 서브 블록의 모션 벡터로 대체될 수 있다. 또한, 좌측 하단 컨트롤 포인트의 위치를 좌측 하단 기준 샘플의 위치로 대체하고, 우측 하단 컨트롤 포인트의 위치를 우측 하단 기준 샘플의 위치로 대체할 수 있다.

[0190] 도 14 내지 도 16을 통해 설명한 바와 달리, 기준 샘플에 인접하는 샘플을 포함하는 서브 블록을 어파인 서브 블록으로 설정할 수도 있다. 구체적으로, 어파인 이웃 서브 블록의 바깥에 위치하는 샘플을 기준 샘플로 설정되어, 어파인 이웃 블록에 포함된 서브 블록을 어파인 서브 블록으로 설정할 수 있다. 일 예로, 도 14의 (a)에 도시된 예에서, 우측 하단 서브 블록의 우상단 샘플의 우측에 위치하는 샘플을 기준 샘플 (x_{n5} , y_{n5})로 설정되어, 우측 하단 서브 블록을 우측 하단 코너에 대한 어파인 서브 블록으로 설정할 수 있다. 또는, 도 14의 (b)에 도시된 예에서, 좌측 하단 서브 블록의 좌상단 샘플의 좌측에 위치하는 샘플을 기준 샘플 (x_{n4} , y_{n4})로 설정되어, 좌측 하단 서브 블록을 좌측 하단 코너에 대한 어파인 서브 블록으로 설정할 수 있다.

[0191] 도 15 및 도 16에도 설명한 실시예가 동일하게 적용될 수 있다. 즉, 도 15의 (a) 또는 도 16의 (a)에 도시된 예에서, 우측 하단 서브 블록의 우하단 샘플 또는 우중단 샘플의 우측에 위치하는 샘플을 기준 샘플 (x_{n5} , y_{n5})로 설정하되, 우측 하단 서브 블록을 우측 하단 코너에 대한 어파인 서브 블록으로 설정할 수 있다. 또는, 도 15의 (b) 또는 도 16의 (b)에 도시된 예에서, 좌측 하단 서브 블록의 좌하단 샘플 또는 좌중단 샘플의 좌측에 위치하는 샘플을 기준 샘플 (x_{n4} , y_{n4})로 설정하되, 좌측 하단 서브 블록을 좌측 하단 코너에 대한 어파인 서브 블록으로 설정할 수 있다.

[0192] 상술한 예에서와 같이, 어파인 서브 블록의 모션 벡터를 이용하여, 어파인 이웃 블록의 어파인 시드 벡터를 유도할 수 있다. 이를 위해, 부호화/복호화가 완료된 블록의 경우, 서브 블록 단위로 모션 벡터를 저장할 수 있다.

[0193] 다른 예로, 어파인 이웃 블록에 대해 최소 개수의 어파인 시드 벡터들을 저장한 뒤, 저장된 어파인 시드 벡터들을 이용하여 어파인 서브 블록의 모션 벡터를 유도할 수도 있다.

[0194] 수학식 9 및 수학식 10은 어파인 이웃 블록의 어파인 시드 벡터들을 이용하여, 어파인 서브 블록의 모션 벡터를 유도하는 예를 나타낸 것이다.

수학식 9

$$\begin{aligned} nv_{4x} &= \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_{n4} - x_{n0}) - \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (y_{n4} - y_{n0}) + nv_{0x} \\ nv_{4y} &= \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_{n4} - x_{n0}) - \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (y_{n4} - y_{n0}) + nv_{0y} \end{aligned}$$

[0195]

수학식 10

$$\begin{aligned} nv_{5x} &= \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_{n5} - x_{n0}) - \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (y_{n5} - y_{n0}) + nv_{0x} \\ nv_{5y} &= \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_{n5} - x_{n0}) - \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (y_{n5} - y_{n0}) + nv_{0y} \end{aligned}$$

[0196] [0197] 수학식 9 및 수학식 10에서, (nv_{4x}, nv_{4y}) 는 좌측 하단 컨트롤 포인트에 대한 어파인 서브 블록의 모션 벡터를 나타내고, (nv_{5x}, nv_{5y}) 는 우측 하단 컨트롤 포인트에 대한 어파인 서브 블록의 모션 벡터를 나타낸다. 어파인 서브 블록의 모션 벡터와 컨트롤 포인트의 어파인 시드 벡터가 동일하게 설정되므로, (nv_{4x}, nv_{4y}) 를 좌측 하단 컨트롤 포인트에 대한 어파인 시드 벡터 (nv_{2x}, nv_{2y}) 로 치환하거나, (nv_{5x}, nv_{5y}) 를 우측 하단 컨트롤 포인트에 대한 어파인 시드 벡터 (nv_{3x}, nv_{3y}) 로 치환할 수 있다.

[0198] (x_{n4}, y_{n4}) 는 좌측 하단 서브 블록에 대한 기준 샘플의 위치를 가리킨다. 또는, 이 대신 좌측 하단 서브 블록의 중심 위치 또는 좌측 하단 컨트롤 포인트의 위치를 사용할 수도 있다. (x_{n5}, y_{n5}) 는 우측 하단 서브 블록에 대한 기준 샘플의 위치를 가리킨다. 또는, 이 대신 우측 하단 서브 블록의 중심 위치 또는 우측 하단 컨트롤 포인트의 위치를 사용할 수 있다.

[0199] 수학식 9 및 수학식 10은 현재 블록이 코딩 트리 유닛의 경계와 접하지 않는 경우에 적용될 수 있다. 현재 블록이 코딩 트리 유닛의 상단 경계에 접하는 경우에는 수학식 9 및 수학식 10을 이용하는 대신, 좌측 하단 기준 샘플을 기초로 결정되는 어파인 서브 블록의 병진 모션 벡터를 제3 어파인 시드 벡터로 설정하고, 우측 하단 기준 샘플을 기초로 결정되는 어파인 서브 블록의 병진 모션 벡터를 제4 어파인 시드 벡터로 설정할 수 있다.

[0200] 수학식 7 및 수학식 8에서, $(x_{n3} - x_{n2})$ 는 좌측 하단 컨트롤 포인트와 우측 하단 컨트롤 포인트 사이의 너비를 나

타낸다. 상술한 바와 같이, x_{n3} 를 우측 하단 기준 샘플의 위치 x_{n5} 로 치환하고, x_{n2} 를 좌측 하단 기준 샘플의 위치 x_{n4} 로 치환할 수 있다. 이하, $(x_{n3}-x_{n2})$ 또는 위 수식의 위치를 기준 샘플들의 위치로 치환함으로써 획득되는 값(예컨대, $(x_{n5}-x_{n4})$)을 변수 W_{seed} 라 정의하고, 상기 변수를 서브 시드 벡터 너비라 호칭하기로 한다.

[0201] 기준 샘플들의 위치에 따라, 서브 시드 벡터 너비가 2의 멱급수(예컨대, 2^h)가 아닌 경우가 발생할 수 있다. 일 예로, 좌측 하단 서브 블록의 좌하단 샘플이 좌측 하단 기준 샘플로 설정되고, 우측 하단 서브 블록의 우하단 샘플이 우측 하단 기준 샘플로 설정된 경우, 서브 시드 벡터의 너비는 2의 배수가 아니다. 이처럼, 서브 시드 벡터 너비가 2의 멱급수가 아닌 경우, 2의 멱급수가 되도록 서브 시드 벡터 너비를 변환할 수 있다. 상기 변환은 서브 시드 벡터 너비에 오프셋을 가산/감산하는 것, 또는 기준 샘플의 위치 대신 상기 기준 샘플에 이웃하는 샘플의 위치를 이용하는 것을 포함할 수 있다. 일 예로, 좌측 하단 기준 샘플 및 우측 하단 기준 샘플 사이의 너비에 1을 가산하여 변환된 서브 시드 벡터 너비를 유도할 수 있다. 또는, 우측 하단 기준 샘플의 우측에 이웃한 이웃 기준 샘플과 좌측 하단 참조 샘플 사이의 너비를 변환된 서브 시드 벡터 너비로 설정할 수 있다. 이후, 변환된 서브 시드 벡터 너비를 수학식 7 및 수학식 8에 적용하여, 현재 블록의 어파인 시드 벡터를 유도할 수 있다.

[0202] 수학식 7 및 수학식 8에 포함된 나누기 연산을 비트 시프트 연산으로 변경할 수도 있다. 비트 시프트 연산은 변환된 서브 시드 벡터 너비(즉, 2의 멱급수로 표현되는 값)에 의해 유도되는 값을 기초로 수행될 수 있다.

[0203] 어파인 서브 블록을 결정하는데 이용되는 기준 샘플이 어파인 이웃 블록에 속하지 않는 경우, 어파인 이웃 블록에 포함된 샘플들 중 기준 샘플에 인접하는 샘플을 기초로, 어파인 이웃 블록의 어파인 시드 벡터를 유도할 수 있다. 구체적으로, 어파인 이웃 블록 내에서 상기 기준 샘플에 인접하는 샘플(이하, 이웃 기준 샘플이라 함)을 포함하는 서브 블록의 병진 모션 벡터를 어파인 이웃 블록의 어파인 시드 벡터로 설정할 수 있다. 이와 같이, 이웃 기준 샘플을 이용한 어파인 시드 벡터의 유도 방법을, 변형된 어파인 머지 벡터 유도 방법이라 정의할 수 있다.

[0204] 도 17은 변형된 어파인 머지 벡터 유도 방법이 적용된 일 예를 나타낸 도면이다.

[0205] 어파인 이웃 블록 E의 우측 하단 기준 샘플 (x_{n5}, y_{n5}) 가 어파인 주변 블록에 속하지 않는 경우, 어파인 이웃 블록에 포함된 샘플들 중 우측 하단 기준 샘플의 좌측에 인접하는 샘플 $(x_{n5}-1, y_{n5})$ 를 기초로 어파인 시드 벡터를 유도할 수 있다. 구체적으로, 이웃 기준 샘플 $(x_{n5}-1, y_{n5})$ 를 포함하는 서브 블록의 병진 모션 벡터를 우측 하단 컨트롤 포인트의 어파인 시드 벡터로 설정할 수 있다.

[0206] 도 17에 도시된 예에서는, 우측 하단 서브 블록의 우상단 샘플의 우측에 이웃하는 샘플이 우측 하단 기준 샘플인 것으로 도시하였다. 우측 하단 서브 블록의 우하단 샘플의 우측에 이웃하는 샘플 또는 우측 하단 서브 블록의 우중단 샘플의 우측에 이웃하는 샘플이 우측 하단 기준 샘플로 설정된 경우에도, 상기 이웃 기준 샘플의 좌측에 인접하는 샘플을 기초로 어파인 시드 벡터가 유도될 수 있다.

[0207] 또한, 좌측 하단 기준 샘플이 어파인 이웃 블록에 속하지 않는 경우에도, 설명한 실시예에 따라, 좌측 하단 기준 샘플이 우측에 인접하는 샘플을 기초로 어파인 시드 벡터를 유도할 수 있다.

[0208] 기준 샘플의 위치와 어파인 시드 벡터를 유도하기 위한 서브 블록을 상이하게 설정함으로써, 서브 시드 벡터 너비를 2의 멱급수로 설정할 수 있다.

[0209] 현재 블록 주변에 어파인 인터 모드로 부호화되지 않은 이웃 블록을 이용하여, 머지 후보, 어파인 시드 벡터 예측 후보 또는 현재 블록의 어파인 시드 벡터를 유도할 수 있다. 구체적으로, 어파인 인터 모드로 부호화되지 않은 블록들을 조합하고, 이들의 조합을 머지 후보 또는 어파인 시드 벡터 예측 후보로 설정할 수 있다. 일 예로, 현재 블록의 좌측 상단 코너에 인접하는 이웃 블록들 중 어느 하나의 모션 벡터, 현재 블록의 우측 상단 코너에 인접하는 이웃 블록들 중 어느 하나의 모션 벡터 또는 현재 블록의 좌측 하단 코너에 인접하는 이웃 블록들 중 어느 하나의 모션 벡터 중 적어도 하나의 조합을 머지 후보 또는 어파인 시드 벡터 예측 후보로 설정할 수 있다. 이때, 좌측 상단 코너에 인접하는 이웃 블록의 모션 벡터, 우측 상단 코너에 인접하는 이웃 블록의 모션 벡터 및 좌측 하단 코너에 인접하는 이웃 블록의 모션 벡터는 각각 좌측 상단 컨트롤 포인트에 대한 제1 어파인 시드 벡터, 우측 상단 컨트롤 포인트에 대한 제2 어파인 시드 벡터 및 좌측 하단 컨트롤 포인트에 대한 제3 어파인 시드 벡터로 설정될 수 있다.

[0210] 또는, 상술한 변형된 어파인 머지 벡터 유도 방법을 이용하여, 어파인 인터 모드로 부호화되지 않은 이웃 블록

을 이용하여, 머지 후보, 어파인 시드 벡터 예측 후보 또는 현재 블록의 어파인 시드 벡터를 유도할 수 있다. 이하, 어파인 인터 모드로 부호화되지 않은 이웃 블록을 논 어파인 이웃 블록이라 호칭하기로 한다.

[0211] 도 18은 논 어파인 이웃 블록을 기초로 현재 블록의 어파인 시드 벡터를 유도하는 예를 나타낸 도면이다.

[0212] 도 18에 도시된 예에서, 현재 블록에 이웃하는 이웃 블록들 모두가 논 어파인 주변 블록인 것으로 가정한다.

[0213] 현재 블록에 이웃하는 이웃 블록들 중 논 어파인 주변 블록 A로부터 현재 블록의 어파인 시드 벡터를 유도하고자 하는 경우, A에 대한 좌측 하단 기준 샘플 및 우측 하단 기준 샘플을 설정할 수 있다. 일 예로, 블록 A의 좌측 하단 샘플의 왼쪽에 이웃하는 샘플을 좌측 하단 기준 샘플로 설정하고, 블록 A의 우측 하단 샘플을 우측 하단 기준 샘플로 설정할 수 있다. 좌측 하단 기준 샘플은 블록 A를 벗어나므로, 상기 좌측 하단 기준 샘플의 우측에 인접하는 샘플을 포함하는 서브 블록의 모션 벡터를 블록 A의 제3 어파인 시드 벡터로 설정할 수 있다. 또한, 우측 하단 기준 샘플을 포함하는 서브 블록의 모션 벡터를 블록 A의 제4 어파인 시드 벡터로 설정할 수 있다. 이후, 수학식 9 및 10에 기초하여, 블록 A로부터 현재 블록에 대한 제1 어파인 시드 벡터 및 제2 어파인 시드 벡터를 유도할 수 있다.

[0214] 논 어파인 주변 블록에 대한 움직임 보상 예측이 서브 블록 단위로 수행된 경우에 한하여, 논 어파인 주변 블록으로부터 어파인 시드 벡터를 유도하는 방법이 이용될 수 있다. 여기서, 서브 블록 단위로 움직임 보상 예측을 수행하는 예측 기법은 STMVP, ATMVP, BIO (Bi-Directional Optical Flow), OBMC (Overlapped Block Motion Compensation) 또는 DMVR (Decoder side Motion Vector Refinement) 중 적어도 하나를 포함할 수 있다.

[0215] 상술한 실시예에서는, 현재 블록의 상단 경계가 코딩 트리 유닛의 경계에 접하면, 현재 블록의 상단에 위치하는 어파인 이웃 블록의 좌측 하단 컨트롤 포인트에 대한 제3 어파인 시드 벡터 및 우측 하단 컨트롤 포인트에 대한 제4 어파인 시드 벡터를 이용하여, 머지 후보, 어파인 시드 벡터 예측 후보 또는 현재 블록의 어파인 시드 벡터가 유도되는 것으로 설명하였다.

[0216] 다른 예로, 현재 블록의 상단 경계가 코딩 트리 유닛의 경계에 접하고, 현재 블록의 상단에 위치한 이웃 블록이 현재 블록과 상이한 코딩 트리 유닛에 속하면, 상기 이웃 블록 대신, 현재 블록이 속한 코딩 트리 유닛에 포함된 블록들 중 상기 이웃 블록과 가장 가까운 이웃 블록을 이용하여, 머지 후보, 어파인 시드 벡터 예측 후보 또는 현재 블록의 어파인 시드 벡터를 유도할 수 있다.

[0217] 도 19에 도시된 예에서는, 현재 블록이 코딩 트리 유닛의 상단 경계에 접하고, 현재 블록 상단에 위치한 블록들, B, C 및 E는 현재 블록과 상이한 코딩 트리 유닛에 속하는 것으로 도시되었다. 이에 따라, 블록 E를 이용하는 대신, 현재 블록이 속한 코딩 트리 유닛에 포함된 블록들 중 블록 E에 인접하는 블록 F를 이용하여, 현재 블록의 어파인 시드 벡터를 유도할 수 있다.

[0218] 현재 블록의 움직임 보상 예측을 위해, 복수 블록들의 어파인 시드 벡터들이 이용될 수 있다. 일 예로, 머지 후보 리스트로부터 복수개의 머지 후보들을 선택하고, 선택된 머지 후보들의 어파인 시드 벡터들을 기초로, 현재 블록에 대한 어파인 시드 벡터 또는 서브 블록 벡터를 유도할 수 있다. 복수 블록들의 어파인 시드 벡터들을 이용하여, 현재 블록을 부호화/복호화하는 것을 복수 어파인 머지 부호화 방법이라 호칭할 수 있다.

[0219] 현재 블록에 복수 어파인 머지 부호화 방법이 적용되는지 여부를 나타내는 정보가 부호화되어 비트스트림을 통해 전송될 수 있다. 또는, 현재 블록에 인접하는 이웃 블록들 중 어파인 이웃 블록의 개수, 머지 후보 리스트에 포함된 머지 후보의 개수 또는 현재 블록의 어파인 모션 모델 중 적어도 하나에 기초하여, 현재 블록에 복수 어파인 머지 부호화 방법을 적용할 것인지 여부가 결정될 수 있다.

[0220] 도 20 및 도 21은 복수 머지 후보들을 이용한 움직임 보상 예측 방법을 나타낸 흐름도이다.

[0221] 도 20은 복수 머지 후보들의 어파인 시드 벡터들을 이용하여, 현재 블록의 어파인 시드 벡터들을 유도하는 예를 나타낸 것이다. 도 21은 복수 머지 후보들의 어파인 시드 벡터들을 이용하여, 각 서브 블록의 모션 벡터를 유도하는 예를 나타낸 도면이다.

[0222] 현재 블록의 어파인 시드 벡터는 두 머지 후보들의 어파인 시드 벡터의 합, 차분, 평균 또는 가중합 연산에 기초하여 생성될 수 있다.

[0223] 다음의 수학식 11 및 12는 머지 후보들의 어파인 시드 벡터들의 합산을 통해 현재 블록의 어파인 시드 벡터가 유도되는 예를 나타낸 것이다.

수학식 11

$$[0224] (sv_{4x}, sv_{4y}) = (sv_{0x}, sv_{0y}) + (sv_{2x}, sv_{2y})$$

수학식 12

$$[0225] (sv_{5x}, sv_{5y}) = (sv_{1x}, sv_{1y}) + (sv_{3x}, sv_{3y})$$

[0226] 수학식 11 및 12에서, sv_4 는 현재 블록의 제1 어파인 시드 벡터를 나타내고, sv_0 는 제1 머지 후보의 제1 어파인 시드 벡터를 나타내며, sv_2 는 제2 머지 후보의 제1 어파인 시드 벡터를 나타낸다. 또한, sv_5 는 현재 블록의 제2 어파인 시드 벡터를 나타내고, sv_1 는 제1 머지 후보의 제2 어파인 시드 벡터를 나타내며, sv_3 는 제2 머지 후보의 제2 어파인 시드 벡터를 나타낸다.

[0227] 또한, 다음 수학식 13 및 14는 머지 후보들의 어파인 시드 벡터들의 가중합 연산을 통해 현재 블록의 어파인 시드 벡터가 유도되는 예를 나타낸 것이다.

수학식 13

$$[0228] (sv_{4x}, sv_{4y}) = (sv_{0x}, sv_{0y}) + ((sv_{2x}, sv_{2y}) >> 1)$$

수학식 14

$$[0229] (sv_{5x}, sv_{5y}) = (sv_{1x}, sv_{1y}) + ((sv_{3x}, sv_{3y}) >> 1)$$

[0230] 다른 예로, 제1 머지 후보의 어파인 시드 벡터들 및 제2 머지 후보의 어파인 시드 벡터들을 기초로, 각각 현재 블록 내 각각의 서브 블록에 대한 제1 서브 블록 모션 벡터 및 제2 서브 블록 모션 벡터를 생성할 수 있다. 이후, 제1 서브 블록 모션 벡터 및 제2 서브 블록 모션 벡터의 합, 차분, 평균 또는 가중합 연산에 기초하여 최종 서브 블록 모션 벡터를 생성할 수 있다.

[0231] 다음의 수학식 15는 제1 서브 블록 모션 벡터 및 제2 서브 블록 모션 벡터의 합산을 통해 최종 서브 블록 모션 벡터를 유도하는 예를 나타낸 것이다.

수학식 15

$$[0232] (V_{2x}, V_{2y}) = (V_{0x}, V_{0y}) + (V_{1x}, V_{1y})$$

[0233] 수학식 15에서, V_0 는 제1 서브 블록 모션 벡터를 나타내고, V_1 은 제2 서브 블록 모션 벡터를 나타내며, V_2 는 최종 서브 블록 모션 벡터를 나타낸다.

[0234] 또한, 다음 수학식 16은 제1 서브 블록 모션 벡터 및 제2 서브 블록 모션 벡터의 가중합 연산을 통해 최종 서브 블록 모션 벡터가 유도되는 예를 나타낸 것이다.

수학식 16

$$[0235] (V_{2x}, V_{2y}) = (V_{0x}, V_{0y}) + ((V_{1x}, V_{1y}) >> 1)$$

- [0236] 인트라 예측은 현재 블록 주변에 부호화/복호화가 완료된 복원 샘플을 이용하여, 현재 블록을 예측하는 것이다. 이때, 현재 블록의 인트라 예측에는, 인루프 필터가 적용되기 전의 복원 샘플이 이용될 수 있다.
- [0237] 인트라 예측 기법은 매트릭스(Matrix)에 기반한 인트라 예측 및 주변 복원 샘플과의 방향성을 고려한 일반 인트라 예측을 포함한다. 현재 블록의 인트라 예측 기법을 지시하는 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 1비트의 플래그일 수 있다. 또는, 현재 블록의 위치, 크기, 형태 또는 이웃 블록의 인트라 예측 기법 중 적어도 하나에 기초하여, 현재 블록의 인트라 예측 기법을 결정할 수 있다. 일 예로, 현재 블록이 꾹쳐 바운더리를 걸쳐 존재하는 경우, 현재 블록에는 매트릭트에 기반한 인트라 예측이 적용되지 않도록 설정될 수 있다.
- [0238] 매트릭스에 기반한 인트라 예측은, 부호화기 및 복호화기에서 기 저장된 매트릭스와, 현재 블록 주변의 복원 샘플 사이의 행렬 곱에 기반하여, 현재 블록의 예측 블록을 획득하는 방법이다. 기 저장된 복수개의 매트릭스들 중 어느 하나를 특정하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다. 복호화기는 상기 정보 및 현재 블록의 크기에 기초하여, 현재 블록의 인트라 예측을 위한 매트릭스를 결정할 수 있다.
- [0239] 일반 인트라 예측은, 비방향성 인트라 예측 모드 또는 방향성 인트라 예측 모드에 기초하여, 현재 블록에 대한 예측 블록을 획득하는 방법이다. 이하, 도면을 참조하여, 일반 인트라 예측에 기초한 인트라 예측 수행 과정에 대해 보다 상세히 살펴보기로 한다.
- [0240] 도 22는 본 발명의 일 실시예에 따른, 인트라 예측 방법의 흐름도이다.
- [0241] 현재 블록의 참조 샘플 라인을 결정할 수 있다(S2201). 참조 샘플 라인은 현재 블록의 상단 및/또는 좌측으로부터 k번째 떨어진 라인에 포함된 참조 샘플들의 집합을 의미한다. 참조 샘플은 현재 블록 주변 부호화/복호화가 완료된 복원 샘플로부터 유도될 수 있다.
- [0242] 복수의 참조 샘플 라인들 중 현재 블록의 참조 샘플 라인을 식별하는 인덱스 정보가 비트스트림을 통해 시그널링될 수 있다. 복수의 참조 샘플 라인들은, 현재 블록에 상단 및/또는 좌측 1번째 라인, 2번째 라인, 3번째 라인 또는 4번째 라인 중 적어도 하나를 포함할 수 있다. 표 1은 참조 샘플 라인들 각각에 할당되는 인덱스를 나타낸 것이다. 표 1에서는 1번째 라인, 2번째 라인 및 4번째 라인이 참조 샘플 라인 후보로 이용됨을 가정하였다.
- 표 1**
- | 인덱스 | 참조 샘플 라인 |
|-----|--------------|
| 0 | 1번째 참조 샘플 라인 |
| 1 | 2번째 참조 샘플 라인 |
| 2 | 4번째 참조 샘플 라인 |
- [0243] [0244] 현재 블록의 위치, 크기, 형태 또는 이웃 블록의 예측 부호화 모드 중 적어도 하나에 기초하여, 현재 블록의 참조 샘플 라인을 결정할 수도 있다. 일 예로, 현재 블록이 꾹쳐, 타일, 슬라이스 또는 코딩 트리 유닛의 경계에 접하는 경우, 첫번째 참조 샘플 라인을 현재 블록의 참조 샘플 라인으로 결정할 수 있다. 참조 샘플 라인에는 현재 블록의 상단에 위치하는 상단 참조 샘플들 및 현재 블록의 좌측에 위치하는 좌측 참조 샘플들을 포함할 수 있다. 상단 참조 샘플들 및 좌측 참조 샘플들은 현재 블록 주변의 복원 샘플들로부터 유도될 수 있다. 상기 복원 샘플들은 인루프 필터가 적용되기 이전 상태일 수 있다.
- [0245] 도 23은 각 참조 샘플 라인이 포함하는 참조 샘플들을 나타낸 도면이다.
- [0246] 현재 블록의 인트라 예측 모드에 따라, 참조 샘플 라인에 속한 참조 샘플들 중 적어도 하나를 이용하여, 예측 샘플을 획득할 수 있다.
- [0247] 다음으로, 현재 블록의 인트라 예측 모드를 결정할 수 있다(S2202). 현재 블록의 인트라 예측 모드는 비방향성 인트라 예측 모드 또는 방향성 인트라 예측 모드 중 적어도 하나가 현재 블록의 인트라 예측 모드로 결정될 수 있다. 비방향성 인트라 예측 모드는, 플래너 및 DC를 포함하고, 방향성 인트라 예측 모드는 좌하단 대각 방향부터 우상단 대각 방향까지 33개 또는 65개의 모드들을 포함한다.
- [0248] 도 24는 인트라 예측 모드들을 나타낸 도면이다.
- [0249] 도 24의 (a)는 35개의 인트라 예측 모드를 나타낸 것이고, 도 24의 (b)는 67개의 인트라 예측 모드들을 나타낸

것이다.

[0250] 도 24에 도시된 것보다 더 많은 수 혹은 더 적은 수의 인트라 예측 모드들이 정의될 수도 있다.

[0251] 현재 블록에 인접하는 이웃 블록의 인트라 예측 모드를 기초로, MPM(Most Probable Mode)을 설정할 수 있다. 여기서, 이웃 블록은, 현재 블록의 좌측에 인접하는 좌측 이웃 블록 및 현재 블록의 상단에 이웃하는 상단 이웃 블록을 포함할 수 있다. 현재 블록의 좌측 상단 샘플의 좌표를 $(0, 0)$ 이라 할 때, 좌측 이웃 블록은 $(-1, 0)$, $(-1, H-1)$ 또는 $(-1, (H-1)/2)$ 위치의 샘플을 포함할 수 있다. 여기서, H 는 현재 블록의 높이를 나타낸다. 상단 이웃 블록은 $(0, -1)$, $(W-1, -1)$ 또는 $((W-1)/2, -1)$ 위치의 샘플을 포함할 수 있다. 여기서, W 는 현재 블록의 너비를 나타낸다.

[0252] 이웃 블록이 일반 인트라 예측으로 부호화된 경우, 이웃 블록의 인트라 예측 모드에 기초하여 MPM을 유도할 수 있다. 구체적으로, 좌측 이웃 블록의 인트라 예측 모드를 변수 candIntraPredModeA로 설정하고, 상단 이웃 블록의 인트라 예측 모드를 변수 candIntraPredModeB로 설정할 수 있다.

[0253] 이때, 이웃 블록이 이용 불가능한 경우(예컨대, 이웃 블록이 아직 부호화/복호화되지 않은 경우 또는 이웃 블록의 위치가 꽉쳐 경계를 벗어난 경우), 이웃 블록이 매트릭스에 기반한 인트라 예측으로 부호화된 경우, 이웃 블록이 인터 예측으로 부호화된 경우 또는 이웃 블록이 현재 블록과 상이한 코딩 트리 유닛에 포함된 경우에 있어서, 이웃 블록의 인트라 예측 모드를 기초로 유도되는 변수 candIntraPredModeX (여기서, X는 A 또는 B)를 디폴트 모드로 설정할 수 있다. 여기서, 디폴트 모드는 플래너, DC, 수직 방향 모드 또는 수평 방향 모드 중 적어도 하나를 포함할 수 있다.

[0254] 또는, 이웃 블록이 매트릭스에 기반한 인트라 예측으로 부호화된 경우, 매트릭스들 중 어느 하나를 특정하기 위한 인덱스 값에 대응하는 인트라 예측 모드를 candIntraPredModeX로 설정할 수 있다. 이를 위해, 매트릭스를 특정하기 위한 인덱스 값들과 인트라 예측 모드들의 매핑 관계를 나타내는 톡업 테이블이 부호화기 및 복호화기에 기 저장될 수 있다.

[0255] 변수 candIntraPredModeA 및 변수 candIntraPredModeB에 기초하여, MPM들을 유도할 수 있다. MPM 리스트에 포함되는 MPM의 개수는 부호화기 및 복호화기에서 기 설정될 수 있다. 일 예로, MPM의 개수는, 3개, 4개, 5개 혹은 6개일 수 있다. 또는, MPM의 개수를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 또는, 이웃 블록의 예측 부호화 모드, 현재 블록의 크기 또는 형태 중 적어도 하나에 기초하여 MPM의 개수가 결정될 수 있다.

[0256] 후술되는 실시예들에서는 MPM의 개수가 3개인 것으로 가정하고, 3개의 MPM들을 MPM[0], MPM[1] 및 MPM[2]라 호칭하기로 한다. MPM들의 개수가 3개보다 많은 경우, MPM들은 후술되는 실시예들에서 설명하는 3개의 MPM을 포함하도록 구성될 수 있다.

[0257] candIntraPredA와 candIntraPredB가 동일하고, candIntraPredA가 플래너 또는 DC 모드인 경우, MPM[0] 및 MPM[1]는 각각 플래너 및 DC 모드로 설정될 수 있다. MPM[2]는 수직 방향 인트라 예측 모드, 수평 방향 인트라 예측 모드 또는 대각 방향 인트라 예측 모드로 설정될 수 있다. 대각 방향 인트라 예측 모드는, 좌측 하단 대각 방향 인트라 예측 모드, 좌측 상단 방향 인트라 예측 모드 또는 우측 상단 방향 인트라 예측 모드일 수 있다.

[0258] candIntraPredA와 candIntraPredB가 동일하고, candIntraPredA가 방향성 인트라 예측 모드인 경우, MPM[0]은 candIntraPredA와 동일하게 설정될 수 있다. MPM[1] 및 MPM[2]는 candIntraPredA와 유사한 인트라 예측 모드들로 설정될 수 있다. candIntraPredA와 유사한 인트라 예측 모드는 candIntraPredA와 인덱스 차분값이 ± 1 또는 ± 2 인 인트라 예측 모드일 수 있다. candIntraPredA와 유사한 인트라 예측 모드를 유도하기 위해 모듈로 연산(%) 및 오프셋이 이용될 수 있다.

[0259] candIntraPredA와 candIntraPredB가 상이한 경우, MPM[0]은 candIntraPredA와 동일하게 설정되고, MPM[1]은 candIntraPredB와 동일하게 설정될 수 있다. 이때, candIntraPredA와 candIntraPredB가 모두 비방향성 인트라 예측 모드인 경우, MPM[2]는 수직 방향 수직 방향 인트라 예측 모드, 수평 방향 인트라 예측 모드 또는 대각 방향 인트라 예측 모드로 설정될 수 있다. 또는, candIntraPredA와 candIntraPredB 중 적어도 하나가 방향성 인트라 예측 모드인 경우, MPM[2]는 플래너, DC, 또는 candIntraPredA 또는 candIntraPredB 중 더 큰 값을 갖는 것에 오프셋을 가산 또는 감산함으로써 유도되는 인트라 예측 모드로 설정될 수 있다. 여기서, 오프셋은 1 또는 2일 수 있다.

[0260] 복수의 MPM을 포함하는 MPM 리스트를 생성하고, 현재 블록의 인트라 예측 모드와 동일한 MPM이 MPM 리스트에 포함되어 있는지 여부를 나타내는 정보를 비트스트림을 통해 시그널링할 수 있다. 상기 정보는 1비트의 플래그로

MPM 플래그가 호칭될 수 있다. 상기 MPM 플래그가 현재 블록과 동일한 MPM이 MPM 리스트에 포함되어 있음을 나타내는 경우, MPM들 중 하나를 식별하는 인덱스 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 인덱스 정보에 의해 특정된 MPM이 현재 블록의 인트라 예측 모드로 설정될 수 있다. 상기 MPM 플래그가 현재 블록과 동일한 MPM이 MPM 리스트에 포함되어 있지 않음을 나타내는 경우, MPM들을 제외한 잔여 인트라 예측 모드들 중 어느 하나를 지시하는 잔여 모드 정보가 비트스트림을 통해 시그널링될 수 있다. 잔여 모드 정보는 MPM들을 제외한 잔여 인트라 예측 모드들에 인덱스를 재할당하였을 때, 현재 블록의 인트라 예측 모드에 대응되는 인덱스 값을 가리킨다. 복호화기에서는 MPM들을 오름차순으로 정렬하고, 잔여 모드 정보를 MPM들과 비교하여 현재 블록의 인트라 예측 모드를 결정할 수 있다. 일 예로, 잔여 모드 정보가 MPM과 같거나 작은 경우, 잔여 모드 정보에 1을 가산하여 현재 블록의 인트라 예측 모드를 유도할 수 있다.

[0261] 디폴트 모드를 MPM으로 설정하는 대신, 현재 블록의 인트라 예측 모드가 디폴트 모드인지 여부를 나타내는 정보를 비트스트림을 통해 시그널링할 수 있다. 상기 정보는 1비트의 플래그이고, 상기 플래그를 디폴트 모드 플래그라 호칭할 수 있다. 상기 디폴트 모드 플래그는 MPM 플래그가 현재 블록과 동일한 MPM이 MPM 리스트에 포함되어 있음을 나타내는 경우에 한하여 시그널링될 수 있다. 상술한 바와 같이, 디폴트 모드는, 플래너, DC, 수직 방향 모드 또는 수평 방향 모드 중 적어도 하나를 포함할 수 있다. 일 예로, 플래너가 디폴트 모드로 설정된 경우, 디폴트 모드 플래그는 현재 블록의 인트라 예측 모드가 플래너인지 여부를 지시할 수 있다. 디폴트 모드 플래그가 현재 블록의 인트라 예측 모드가 디폴트 모드가 아님을 가리키는 경우, 인덱스 정보에 의해 지시되는 MPM들 중 하나를 현재 블록의 인트라 예측 모드로 설정할 수 있다.

[0262] 복수개의 인트라 예측 모드들이 디폴트 모드들로 설정된 경우, 디폴트 모드들 중 어느 하나를 지시하는 인덱스 정보가 더 시그널링될 수 있다. 현재 블록의 인트라 예측 모드는 상기 인덱스 정보가 가리키는 디폴트 모드로 설정될 수 있다.

[0263] 현재 블록의 참조 샘플 라인의 인덱스가 0이 아닌 경우에는 디폴트 모드를 이용하지 못하도록 설정할 수 있다. 이에 따라, 참조 샘플 라인의 인덱스가 0이 아닌 경우에는 디폴트 모드 플래그를 시그널링하지 않고, 상기 디폴트 모드 플래그의 값을 기 정의된 값(즉, 거짓)으로 설정할 수 있다.

[0264] 현재 블록의 인트라 예측 모드가 결정되면, 결정된 인트라 예측 모드를 기초로, 현재 블록에 대한 예측 샘플들을 획득할 수 있다(S2203).

[0265] DC 모드가 선택된 경우, 참조 샘플들의 평균값을 기초로, 현재 블록에 대한 예측 샘플들이 생성된다. 구체적으로, 예측 블록 내 전체 샘플들의 값은 참조 샘플들의 평균값을 기초로 생성될 수 있다. 평균값은, 현재 블록의 상단에 위치하는 상단 참조 샘플들 및 현재 블록의 좌측에 위치하는 좌측 참조 샘플들 중 적어도 하나를 이용하여 유도될 수 있다.

[0266] 현재 블록의 형태에 따라, 평균값을 유도하는데 이용되는 참조 샘플들의 개수 또는 범위가 달라질 수 있다. 일 예로, 현재 블록이 너비가 높이보다 큰 비정방형 블록인 경우, 상단 참조 샘플들만을 이용하여 평균값을 계산할 수 있다. 반면, 현재 블록이 너비가 높이보다 작은 비정방형 블록인 경우, 좌측 참조 샘플들만을 이용하여 평균값을 계산할 수 있다. 즉, 현재 블록의 너비 및 높이가 상이한 경우, 길이가 더 긴 쪽에 인접하는 참조 샘플들만을 이용하여 평균값을 계산할 수 있다. 또는, 현재 블록의 너비와 높이 비율에 기초하여, 상단 참조 샘플들만을 이용하여 평균값을 계산할 것인지 여부 또는 좌측 참조 샘플들만을 이용하여 평균값을 계산할 것인지 여부를 결정할 수 있다.

[0267] 플래너 모드가 선택된 경우, 수평 방향 예측 샘플과 수직 방향 예측 샘플을 이용하여, 예측 샘플을 획득할 수 있다. 여기서, 수평 방향 예측 샘플은, 예측 샘플과 동일한 수평선상에 위치하는 좌측 참조 샘플 및 우측 참조 샘플을 기초로 획득되고, 수직 방향 예측 샘플은, 예측 샘플과 동일한 수직선상에 위치하는 상단 참조 샘플 및 하단 참조 샘플을 기초로 획득된다. 여기서, 우측 참조 샘플은, 현재 블록의 우측 상단 코너에 인접하는 참조 샘플을 복사하여 생성되고, 하단 참조 샘플은, 현재 블록의 좌측 하단 코너에 인접하는 참조 샘플을 복사하여 생성될 수 있다. 수평 방향 예측 샘플은 좌측 참조 샘플 및 우측 참조 샘플의 가중합 연산을 기초로 획득되고, 수직 방향 예측 샘플은 상단 참조 샘플 및 하단 참조 샘플의 가중합 연산을 기초로 획득될 수 있다. 이때, 각 참조 샘플에 부여되는 가중치는 예측 샘플의 위치에 따라 결정될 수 있다. 예측 샘플은 수평 방향 예측 샘플 및 수직 방향 예측 샘플의 평균 연산 또는 가중합 연산을 기초로 획득될 수 있다. 가중합 연산이 수행되는 경우, 예측 샘플의 위치에 기초하여 수평 방향 예측 샘플 및 수직 방향 예측 샘플에 부여되는 가중치를 결정할 수 있다.

- [0268] 방향성 예측 모드가 선택되는 경우, 선택된 방향성 예측 모드의 예측 방향(또는 예측 각도)을 나타내는 파라미터를 결정할 수 있다. 하기 표 2는 인트라 예측 모드 별 인트라 방향 파라미터 intraPredAng를 나타낸 것이다.

표 2

PredModeIntra IntraPredAng	1 -	2 32	3 26	4 21	5 17	6 13	7 9
PredModeIntraIntra PredAng	8 5	9 2	10 0	11 -2	12 -5	13 -9	14 -13
PredModeIntraIntra PredAng	15 -17	16 -21	17 -26	18 -32	19 -26	20 -21	21 -17
PredModeIntraIntra PredAng	22 -13	23 -9	24 -5	25 -2	26 0	27 2	28 5
PredModeIntraIntra PredAng	29 9	30 13	31 17	32 21	33 26	34 32	

- [0270] 표 2는 35개의 인트라 예측 모드가 정의되어 있을 때, 인덱스가 2 내지 34 중 어느 하나인 인트라 예측 모드들 각각의 인트라 방향 파라미터를 나타낸다. 33개보다 더 많은 방향성 인트라 예측 모드가 정의되어 있는 경우, 표 2를 보다 세분화하여, 방향성 인트라 예측 모드 각각의 인트라 방향 파라미터를 설정할 수 있다. 현재 블록의 상단 참조 샘플들 및 좌측 참조 샘플들을 일렬로 배열한 뒤, 인트라 방향 파라미터의 값을 기초로, 예측 샘플을 획득할 수 있다. 이때, 인트라 방향 파라미터의 값이 음수인 경우, 좌측 참조 샘플들과 상단 참조 샘플들을 일렬로 배열할 수 있다.

- [0271] 도 25 및 도 26은 참조 샘플들을 일렬로 배열하는 일차원 배열의 예시를 나타낸 도면이다.

- [0272] 도 25는 참조 샘플들을 수직 방향으로 배열하는 수직 방향 일차원 배열의 예시를 나타낸 것이고, 도 26은 참조 샘플들을 수평 방향으로 배열하는 수평 방향 일차원 배열의 예시를 나타낸 것이다. 35개의 인트라 예측 모드가 정의된 경우를 가정하여, 도 25 및 도 26의 실시예를 설명한다.

- [0273] 인트라 예측 모드 인덱스가 11 내지 18 중 어느 하나인 경우, 상단 참조 샘플들을 반시계 방향으로 회전한 수평 방향 일차원 배열을 적용하고, 인트라 예측 모드 인덱스가 19 내지 25 중 어느 하나인 경우, 좌측 참조 샘플들을 시계 방향으로 회전한 수직 방향 일차원 배열을 적용할 수 있다. 참조 샘플들을 일렬로 배열함에 있어서, 인트라 예측 모드 각도를 고려할 수 있다.

- [0274] 인트라 방향 파라미터에 기초하여, 참조 샘플 결정 파라미터를 결정할 수 있다. 참조 샘플 결정 파라미터는 참조 샘플을 특정하기 위한 참조 샘플 인덱스 및 참조 샘플에 적용되는 가중치를 결정하기 위한 가중치 파라미터를 포함할 수 있다.

- [0275] 참조 샘플 인덱스 iIdx 및 가중치 파라미터 ifact는 각각 다음의 수학식 17 및 18을 통해 획득될 수 있다.

수학식 17

$$iIdx = (y+1) * P_{ang} / 32$$

수학식 18

$$ifact = [(y+1) * P_{ang}] \& 31$$

- [0278] 수학식 17 및 18에서 P_{ang} 는 인트라 방향 파라미터를 나타낸다. 참조 샘플 인덱스 iIdx에 의해 특정되는 참조 샘플은 정수 펠(Integer pe1)에 해당한다.

- [0279] 예측 샘플을 유도하기 위해, 적어도 하나 이상의 참조 샘플을 특정할 수 있다. 구체적으로, 예측 모드의 기울기를 고려하여, 예측 샘플을 유도하는데 이용되는 참조 샘플의 위치를 특정할 수 있다. 일 예로, 참조 샘플 인덱

스 iIdx를 이용하여, 예측 샘플을 유도하는데 이용되는 참조 샘플을 특정할 수 있다.

[0280] 이때, 인트라 예측 모드의 기울기가 하나의 참조 샘플로는 표현되지 않는 경우, 복수의 참조 샘플들을 보간하여 예측 샘플을 생성할 수 있다. 일 예로, 인트라 예측 모드의 기울기가 예측 샘플과 제1 참조 샘플 사이의 기울기 및 예측 샘플과 제2 참조 샘플 사이의 기울기 사이의 값인 경우, 제1 참조 샘플 및 제2 참조 샘플을 보간하여 예측 샘플을 획득할 수 있다. 즉, 인트라 예측 각도를 따르는 앵글러 라인(Angular Line)이 정수 펠에 위치한 참조 샘플을 지나지 않는 경우, 상기 앵글러 라인이 지나는 위치의 좌우 또는 상하에 인접 위치하는 참조 샘플들을 보간하여 예측 샘플을 획득할 수 있다.

[0281] 하기 수학식 19는 참조 샘플들을 기초로, 예측 샘플을 획득하는 예를 나타낸 것이다.

수학식 19

$$P(x,y) = ((32-i_{fact})/32) * Ref_1D(x+iIdx+1) + (i_{fact}/32) * Ref_1D(x+iIdx+2)$$

[0283] 수학식 19에서, P는 예측 샘플을 나타내고, Ref_1D은 일차원 배열된 참조 샘플들 중 어느 하나를 나타낸다. 이 때, 참조 샘플의 위치는 예측 샘플의 위치 (x, y) 및 참조 샘플 인덱스 iIdx에 의해 결정될 수 있다.

[0284] 인트라 예측 모드의 기울기가 하나의 참조 샘플로 표현 가능한 경우, 가중치 파라미터 ifact는 0으로 설정된다. 이에 따라, 수학식 19는 다음 수학식 20과 같이 간소화될 수 있다.

수학식 20

$$P(x,y) = Ref_1D(x+iIdx+1)$$

[0286] 복수의 인트라 예측 모드에 기초하여 현재 블록에 대한 인트라 예측을 수행할 수도 있다. 일 예로, 예측 샘플별로 인트라 예측 모드를 유도하고, 각각의 예측 샘플에 할당된 인트라 예측 모드에 기초하여 예측 샘플을 유도할 수 있다.

[0287] 또는, 영역 별로 인트라 예측 모드를 유도하고, 각각의 영역에 할당된 인트라 예측 모드에 기초하여 각 영역에 대한 인트라 예측을 수행할 수 있다. 여기서, 상기 영역은 적어도 하나의 샘플을 포함할 수 있다. 상기 영역의 크기 또는 형태 중 적어도 하나는 현재 블록의 크기, 형태 또는 인트라 예측 모드 중 적어도 하나에 기초하여 적응적으로 결정될 수 있다. 또는, 부호화기 및 복호화기에서 현재 블록의 크기 또는 형태와는 독립적으로 영역의 크기 또는 형태 중 적어도 하나가 기 정의되어 있을 수 있다.

[0288] 또는, 복수의 인트라 예측 각각을 기초로 인트라 예측을 수행하고, 복수회의 인트라 예측을 통해 획득된 복수의 예측 샘플들의 평균 연산 또는 가중합 연산을 기초로 최종 예측 샘플을 유도할 수 있다. 일 예로, 제1 인트라 예측 모드를 기초로 인트라 예측을 수행하여 제1 예측 샘플을 획득하고, 제2 인트라 예측 모드를 기초로 인트라 예측을 수행하여 제2 예측 샘플을 획득할 수 있다. 이후, 제1 예측 샘플 및 제2 예측 샘플 사이의 평균 연산 또는 가중합 연산을 기초로, 최종 예측 샘플을 획득할 수 있다. 이때, 제1 예측 샘플 및 제2 예측 샘플 각각에 할당되는 가중치는, 제1 인트라 예측 모드가 비방향성/방향성 예측 모드인지 여부, 제2 인트라 예측 모드가 비방향성/방향성 예측 모드인지 여부 또는 이웃 블록의 인트라 예측 모드 중 적어도 하나를 고려하여 결정될 수 있다.

[0289] 복수의 인트라 예측 모드들은 비방향성 인트라 예측 모드와 방향성 예측 모드의 조합, 방향성 예측 모드들의 조합 또는 비방향성 예측 모드들의 조합일 수 있다.

[0290] 도 27은 방향성 인트라 예측 모드들이 x축과 평행한 직선과 형성하는 각도를 예시한 도면이다.

[0291] 도 27에 나타난 예서와 같이, 방향성 예측 모드들은 좌측 하단 대각 방향부터 우측 상단 대각 방향 사이에 존재할 수 있다. x축과 방향성 예측 모드가 형성하는 각도로 설명하면, 방향성 예측 모드들은, 45도 (좌측 하단 대각 방향) 부터, -135도 (우측 상단 대각 방향) 사이에 존재할 수 있다.

[0292] 현재 블록이 비정방 형태인 경우, 현재 블록의 인트라 예측 모드에 따라, 인트라 예측 각도를 따르는 앵글러 라인 상에 위치하는 참조 샘플들 중 예측 샘플에 보다 가까운 참조 샘플 대신 예측 샘플에 보다 먼 참조 샘플을

이용하여 예측 샘플을 유도하는 경우가 발생할 수 있다.

[0293] 도 28은 현재 블록이 비정방 형태인 경우, 예측 샘플이 획득되는 양상을 나타낸 도면이다.

[0294] 일 예로, 도 28의 (a)에 도시된 예에서와 같이, 현재 블록이 너비가 높이보다 큰 비정방형이고, 현재 블록의 인트라 예측 모드가 0도부터 45도 사이의 각도를 갖는 방향성 인트라 예측 모드인 것으로 가정한다. 위 경우, 현재 블록의 우측 열 부근의 예측 샘플 A를 유도할 때, 상기 각도를 따르는 앵글러 모드 상에 위치하는 참조 샘플들 중 상기 예측 샘플과 가까운 상단 참조 샘플 T 대신 상기 예측 샘플과 면 좌측 참조 샘플 L을 이용하는 경우가 발생할 수 있다.

[0295] 다른 예로, 도 28의 (b)에 도시된 예에서와 같이, 현재 블록이 높이가 너비보다 큰 비정방형이고, 현재 블록의 인트라 예측 모드가 -90 도부터 -135도 사이인 방향성 인트라 예측 모드인 것으로 가정한다. 위 경우, 현재 블록의 하단 행 부근의 예측 샘플 A를 유도할 때, 상기 각도를 따르는 앵글러 모드 상에 위치하는 참조 샘플들 중 상기 예측 샘플과 가까운 좌측 참조 샘플 L 대신 상기 예측 샘플과 면 상단 참조 샘플 T를 이용하는 경우가 발생할 수 있다.

[0296] 위와 같은 문제점을 해소하기 위해, 현재 블록이 비정방형인 경우, 현재 블록의 인트라 예측 모드를 반대 방향의 인트라 예측 모드로 치환할 수 있다. 이에 따라, 비정방형 블록에 대해서는 도 24에 도시된 방향성 예측 모드들 보다 더 큰 혹은 더 작은 각도를 갖는 방향성 예측 모드들을 사용할 수 있다. 이와 같은, 방향성 인트라 예측 모드를 와이드 앵글 인트라 예측 모드라 정의할 수 있다. 와이드 앵글 인트라 예측 모드는 45도 내지 -135도 범위에 속하지 않는 방향성 인트라 예측 모드를 나타낸다.

[0297] 도 29는 와이드 앵글 인트라 예측 모드들을 나타낸 도면이다.

[0298] 도 29에 도시된 예에서, 인덱스가 -1부터 -14인 인트라 예측 모드들 및 인덱스가 67부터 80 사이인 인트라 예측 모드들이 와이드 앵글 인트라 예측 모드들을 나타낸다.

[0299] 도 29에서는 각도가 45도 보다 큰 14개의 와이드 앵글 인트라 예측 모드들(-1부터 -14) 및 각도가 -135도 보다 작은 14개의 와이드 앵글 인트라 예측 모드들(67부터 80)을 예시하였으나, 이보다 더 많은 수 또는 더 적은 수의 와이드 앵글 인트라 예측 모드들이 정의될 수 있다.

[0300] 와이드 앵글 인트라 예측 모드가 사용되는 경우, 상단 참조 샘플들의 길이는 2W+1로 설정되고, 좌측 참조 샘플들의 길이는 2H+1로 설정될 수 있다.

[0301] 와이드 앵글 인트라 예측 모드를 사용함에 따라, 도 28의 (a)에 도시된 샘플 A는 참조 샘플 T를 이용하여 예측되고, 도 28의 (b)에 도시된 샘플 A는 참조 샘플 L을 이용하여 예측될 수 있다.

[0302] 기존 인트라 예측 모드들과 N개의 와이드 앵글 인트라 예측 모드들을 더해, 총 67 + N개의 인트라 예측 모드들을 사용할 수 있다. 일 예로, 표 3은 20개의 와이드 앵글 인트라 예측 모드가 정의된 경우, 인트라 예측 모드들의 인트라 방향 파라미터를 나타낸 것이다.

표 3

PredModeIntra	-10	-9	-8	-7	-6	-5	-4	-3	-2
intraPredAngle	114	93	79	68	60	54	49	45	39
PredModeIntra	-1	2	3	4	5	6	7	8	9
intraPredAngle	35	32	29	26	23	21	19	17	15
PredModeIntra	10	11	12	13	14	15	16	17	18
intraPredAngle	13	11	9	7	5	3	2	1	0
PredModeIntra	19	20	21	22	23	24	25	26	27
intraPredAngle	-1	-2	-3	-5	-7	-9	-11	-13	-15
PredModeIntra	28	29	30	31	32	33	34	35	36
intraPredAngle	-17	-19	-21	-23	-26	-29	-32	-29	-26
PredModeIntra	37	38	39	40	41	42	43	44	45
intraPredAngle	-23	-21	-19	-17	-15	-13	-11	-9	-7
PredModeIntra	46	47	48	49	50	51	52	53	54
intraPredAngle	-5	-3	-2	-1	0	1	2	3	5
PredModeIntra	55	56	57	58	59	60	61	62	63
intraPredAngle	7	9	11	13	15	17	19	21	23
PredModeIntra	64	65	66	67	68	69	70	71	72

intraPredAngle	26	29	32	35	39	45	49	54	60
PredModeIntra	73	74	75	76					
intraPredAngle	68	79	93	114					

[0304] 현재 블록이 비정방형이고, S2202 단계에서 획득된 현재 블록의 인트라 예측 모드가 변환 범위에 속하는 경우, 현재 블록의 인트라 예측 모드를 와이드 앵글 인트라 예측 모드로 변환할 수 있다. 상기 변환 범위는 현재 블록의 크기, 형태 또는 비율 중 적어도 하나를 기초로 결정될 수 있다. 여기서, 상기 비율은 현재 블록의 너비 및 높이 사이의 비율을 나타낼 수 있다. 현재 블록이 너비가 높이보다 큰 비정방형인 경우, 변환 범위는 우측 상단 대각 방향의 인트라 예측 모드 인덱스(예컨대, 66)부터 (우측 상단 대각 방향인 인트라 예측 모드의 인덱스 - N)로 설정될 수 있다. 여기서, N은 현재 블록의 비율을 기초로 결정될 수 있다. 현재 블록의 인트라 예측 모드가 변환 범위에 속하는 경우, 상기 인트라 예측 모드를 와이드 앵글 인트라 예측 모드로 변환할 수 있다. 상기 변환은 상기 인트라 예측 모드에 기 정의된 값을 차감하는 것일 수 있고, 기 정의된 값은 와이드 앵글 인트라 예측 모드들을 제외한 인트라 예측 모드들의 총 개수(예컨대, 67)일 수 있다.

[0305] 상기 실시예에 의해, 66번부터 53번 사이의 인트라 예측 모드들은, 각각 -1번부터 -14번 사이의 와이드 앵글 인트라 예측 모드들로 변환될 수 있다.

[0306] 현재 블록이 높이가 너비보다 큰 비정방형인 경우, 변환 범위는 좌측 하단 대각 방향의 인트라 예측 모드 인덱스(예컨대, 2)부터 (좌측 하단 대각 방향의 인트라 예측 모드의 인덱스 + M)으로 설정될 수 있다. 여기서, M은 현재 블록의 비율을 기초로 결정될 수 있다. 현재 블록의 인트라 예측 모드가 변환 범위에 속하는 경우, 상기 인트라 예측 모드를 와이드 앵글 인트라 예측 모드로 변환할 수 있다. 상기 변환은 상기 인트라 예측 모드에 기 정의된 값을 가산하는 것일 수 있고, 기 정의된 값은 와이드 앵글 인트라 예측 모드들을 제외한 방향성 인트라 예측 모드들의 총 개수(예컨대, 65)일 수 있다.

[0307] 상기 실시예에 의해, 2번부터 15번 사이의 인트라 예측 모드들 각각은 67번부터 80번 사이의 와이드 앵글 인트라 예측 모드들로 변환될 수 있다.

[0308] 이하, 변환 범위에 속하는 인트라 예측 모드들을 와이드 앵글 인트라 대체 예측 모드로 호칭하기로 한다.

[0309] 변환 범위는 현재 블록의 비율에 기초하여 결정될 수 있다. 일 예로, 표 4 및 표 5는 각각 와이드 앵글 인트라 예측 모드 제외 35개의 인트라 예측 모드가 정의된 경우와 67개의 인트라 예측 모드가 정의된 경우, 변환 범위를 나타낸다.

표 4

Condition	Replaced Intra Prediction Modes
W/H = 2	Modes 2, 3, 4
W/H > 2	Modes 2, 3, 4, 5, 6
W/H = 1	None
H/W = 1/2	Modes 32, 33, 34
H/W < 1/2	Modes 30, 31, 32, 33, 34

표 5

Condition	Replaced Intra Prediction Modes
W/H = 2	Modes 2, 3, 4, 5, 6, 7
W/H > 2	Modes 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
W/H = 1	None
H/W = 1/2	Modes 61, 62, 63, 64, 65, 66
H/W < 1/2	Modes 57, 58, 59, 60, 61, 62, 63, 64, 65, 66

[0312] 표 4 및 표 5에 나타난 예에서와 같이, 현재 블록의 비율에 따라, 변환 범위에 포함되는 와이드 앵글 인트라 대체 예측 모드들의 개수가 상이할 수 있다. 기준의 인트라 예측 모드들에 추가로 와이드 앵글 인트라 예측 모드들이 사용됨에 따라, 와이드 앵글 인트라 예측 모드들을 부호화하는데 필요한 리소스가 증가하여, 부호화 효율이 낮아질 수 있다. 이에 따라, 와이드 앵글 인트라 예측 모드들을 그대로 부호화하는 대신, 와이드 앵글 인트

라 예측 모드들에 대한 대체 인트라 예측 모드들을 부호화하여, 부호화 효율을 향상시킬 수 있다.

[0313] 일 예로, 현재 블록이 67번의 와이드 앵글 인트라 예측 모드를 이용하여 부호화된 경우, 67번의 와이드 앵글 대체 인트라 예측 모드인 2번을 현재 블록의 인트라 예측 모드로 부호화할 수 있다. 또한, 현재 블록이 -1번의 와이드 앵글 인트라 예측 모드로 부호화된 경우, -1번의 와이드 앵글 대체 인트라 예측 모드인 66번을 현재 블록의 인트라 예측 모드로 부호화할 수 있다.

[0314] 복호화기에서는 현재 블록의 인트라 예측 모드를 복호화하고, 복호화된 인트라 예측 모드가 변환 범위에 포함되는지 여부를 판단할 수 있다. 복호화된 인트라 예측 모드가 와이드 앵글 대체 인트라 예측 모드인 경우, 인트라 예측 모드를 와이드 앵글 인트라 예측 모드로 변환할 수 있다.

[0315] 또는, 현재 블록이 와이드 앵글 인트라 예측 모드로 부호화된 경우, 와이드 앵글 인트라 예측 모드를 그대로 부호화할 수도 있다.

[0316] 인트라 예측 모드의 부호화는 상술한 MPM 리스트를 기초로 이루어질 수 있다. 이하, MPM 리스트의 구성 방법에 대해 상세히 살펴보기로 한다. 후술되는 실시예들에서는 각도가 45도 보다 큰 10개의 와이드 앵글 인트라 예측 모드들(-1부터 -10) 및 각도가 -135도보다 작은 10개의 와이드 앵글 인트라 예측 모드들(67부터 76)이 정의된 것으로 가정한다.

[0317] 이웃 블록이 와이드 앵글 인트라 예측 모드로 부호화된 경우, 상기 와이드 앵글 인트라 예측 모드에 대응하는 와이드 앵글 대체 인트라 예측 모드를 기초로, MPM을 설정할 수 있다. 일 예로, 이웃 블록이 와이드 앵글 인트라 예측 모드로 부호화된 경우, 변수 candIntraPredX(X는 A 또는 B)를 와이드 앵글 대체 인트라 예측 모드로 설정할 수 있다.

[0318] 또는, 현재 블록의 형태에 따라, MPM 유도 방법을 결정할 수 있다. 일 예로, 현재 블록이 너비와 높이가 동일한 정방형인 경우, candIntraPredX를 와이드 앵글 대체 인트라 예측 모드로 설정할 수 있다. 반면, 현재 블록이 비정방형인 경우, candIntraPredX를 와이드 앵글 인트라 예측 모드로 설정할 수 있다.

[0319] 또는, 이웃 블록의 와이드 앵글 인트라 예측 모드가 현재 블록에 적용 가능한 것인지 여부에 기초하여 candIntraPredX를 와이드 앵글 인트라 예측 모드로 설정할 것인지 여부가 결정될 수 있다. 일 예로, 현재 블록이 너비가 높이보다 큰 비정방형인 경우에는, 우측 상단 대각 방향의 인트라 예측 모드보다 인덱스가 큰 와이드 앵글 인트라 예측 모드는 그대로 candIntraPredX로 설정되나, 좌측 하단 대각 방향의 인트라 예측 모드보다 인덱스가 작은 와이드 앵글 인트라 예측 모드에 대해서는 이에 대응하는 와이드 앵글 대체 인트라 예측 모드가 candIntraPredX로 설정된다. 반면, 현재 블록이 높이가 너비보다 큰 비정방형인 경우에는, 좌측 하단 대각 방향의 인트라 예측 모드보다 인덱스가 작은 와이드 앵글 인트라 예측 모드는 그대로 candIntraPredX로 설정되나, 우측 상단 대각 방향의 인트라 예측 모드보다 인덱스가 큰 와이드 앵글 인트라 예측 모드에 대해서는 이에 대응하는 와이드 앵글 대체 인트라 예측 모드가 candIntraPredX로 설정된다.

[0320] 즉, 와이드 앵글 인트라 예측 모드로 부호화된 이웃 블록이 현재 블록과 동일/유사한 형태를 갖는지 여부에 따라, 와이드 앵글 인트라 예측 모드를 그대로 이용하여 MPM을 유도할 것인지 여부 또는 와이드 앵글 대체 인트라 예측 모드를 이용하여 MPM을 유도할 것인지 여부가 결정될 수 있다.

[0321] 또는, 현재 블록의 형태와 무관하게 이웃 블록의 와이드 앵글 인트라 예측 모드를 candIntraPredX로 설정할 수도 있다.

[0322] 요약하면, candIntraPredX는 이웃 블록의 와이드 앵글 인트라 예측 모드 또는 와이드 앵글 대체 인트라 예측 모드로 설정될 수 있다.

[0323] candIntraPredA 및 candIntraPredB를 기초로 MPM을 유도할 수 있다. 이때, MPM은 candIntraPredA 또는 candIntraPredB와 유사한 인트라 예측 모드로 유도될 수 있다. candIntraPredA 또는 candIntraPredB와 유사한 인트라 예측 모드는 모듈로 연산과 오프셋을 기초로 유도될 수 있다. 이때, 모듈로 연산에 이용되는 상수와 오프셋은 현재 블록의 형태에 따라 상이하게 결정될 수 있다.

[0324] 표 6은 현재 블록의 형태에 따른 MPM을 유도하는 예를 나타낸 것이다.

표 6

```

[0325]
i) If cuWidth is equal to cuHeight
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 61) % 64)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 64)
ii) otherwise (i.e., cuWidth is not equal to cuHeight)
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)

```

[0326] candIntraPredA 및 candIntraPredB가 동일하고, candIntraPredA는 방향성 인트라 예측 모드인 것으로 가정한다. 현재 블록이 정방형인 경우, candIntraPredA와 유사한 인트라 예측 모드는 와이드 앵글 인트라 예측 모드들을 제외한 방향성 인트라 예측 모드들의 총 개수에서 1을 차감한 값에 기초한 모듈러 연산을 기초로 획득될 수 있다. 일 예로, 와이드 앵글 인트라 예측 모드들 이외의 방향성 인트라 예측 모드들의 개수가 65개인 경우, candIntraPredA를 기초로 유도된 값과 64의 모듈로 연산에 기초하여 MPM을 유도할 수 있다. 반면, 현재 블록이 비정방형인 경우, candIntraPredA와 유사한 인트라 예측 모드는 와이드 앵글 인트라 예측 모드들을 포함한 방향성 인트라 예측 모드들의 총 개수에서 1을 차감한 값에 기초한 모듈러 연산을 기초로 획득될 수 있다. 일 예로, 와이드 앵글 인트라 예측 모드들의 개수가 20개인 경우, candIntrapredA를 기초로 유도된 값과 84의 모듈로 연산에 기초하여 MPM을 유도할 수 있다. 현재 블록의 형태에 따라 모듈로 연산에 이용되는 상수를 상이하게 설정함에 따라, 와이드 앵글 인트라 예측 모드가 candIntraPredA와 유사한 방향성 인트라 예측 모드로 설정될 수 있는지 여부가 결정될 수 있다. 일 예로, 64를 이용한 모듈로 연산에서는 와이드 앵글 인트라 예측 모드가 candIntraPredA와 유사한 방향성 인트라 예측 모드로 설정될 수 없는 반면, 84를 이용한 모듈로 연산에서는 와이드 앵글 인트라 예측 모드가 candIntraPredA와 유사한 방향성 인트라 예측 모드로 설정될 수 있다.

[0327] 또는, candIntraPredA와 candIntraPredB가 동일한 경우, 현재 블록의 형태 및 candIntraPredA가 와이드 앵글 인트라 예측 모드인지 여부를 고려하여 MPM을 유도할 수 있다.

[0328] 표 7은 현재 블록의 형태에 따른 MPM을 유도하는 예를 나타낸 것이다.

표 7

[0329]

```

- If candIntraPredModeB is equal to candIntraPredModeA, the following applies:
-- If candIntraPredModeA is less than 2 (i.e., equal to INTRA_PLANAR or INTRA_DC),
candModeList[x] with x = 0..2 is derived as follows:
candModeList[0] = INTRA_PLANAR
candModeList[1] = INTRA_DC
candModeList[2] = INTRA_ANGULAR50
-- otherwise candModeList[x] with x = 0..2 is derived as follows:
i) If cuWidth is equal to cuHeight
1. if candIntrapredA is smaller than 0 or candIntraPredModeA is greater than 66
candModeList[0] = INTRA_PLANAR
candModeList[1] = INTRA_DC
candModeList[2] = INTRA_ANGULAR50
2. otherwise if candIntraPredA >= 0 && candIntraPredModeA < 67)
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 61) % 64)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 64)
ii) otherwise if cuWidth is greater than cuHeight
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)
iii) otherwise (i.e., cuWidth is greater than cuHeight)
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)
candModeList[2] = -8 + ((candIntraPredModeA - 1) % 84)
- Otherwise (candIntraPredModeB is not equal to candIntraPredModeA), the following applies:
-- candModeList[0] and candModeList[1] are derived as follows:
candModeList[0] = candIntraPredModeA
candModeList[1] = candIntraPredModeB
-- If neither of candModeList[0] and candModeList[1] is equal to INTRA_PLANAR,
candModeList[2] is set to INTRA_PLANAR
-- Otherwise, if neither of candModeList[0] and candModeList[1] is equal to INTRA_DC,
candModeList[2] is set equal to INTRA_DC,
-- Otherwise, candModeList[2] is set to INTRA_ANGULAR 50.

```

[0330]

candIntraPredA와 candIntraPredB는 동일한 것으로 가정한다. 현재 블록이 정방형이고, candIntraPredA가 와이드 앵글 인트라 예측 모드인 경우, MPM들을 디폴트 모드들로 설정할 수 있다. 일 예로, MPM[0], MPM[1] 및 MPM[2]를 각각 플래너, DC 및 수직 방향 인트라 예측 모드로 설정할 수 있다.

[0331]

현재 블록이 정방형이고, candIntraPredA가 와이드 앵글 인트라 예측 모드가 아닌 방향성 인트라 예측 모드인 경우, candIntraPredA와 이에 유사한 방향성 인트라 예측 모드를 MPM들로 설정할 수 있다. 일 예로, MPM[0]를 candIntraPredA로 설정하고, MPM[1] 및 MPM[2]를 candIntraPredA와 유사한 방향성 인트라 예측 모드로 설정할 수 있다.

[0332]

현재 블록이 비정방형이고, candIntraPredA가 방향성 인트라 예측 모드인 경우, candIntraPredA와 이에 유사한 방향성 인트라 예측 모드를 MPM들로 설정할 수 있다. 일 예로, MPM[0]을 candIntraPredA로 설정하고, MPM[1] 및 MPM[2]를 candIntraPredA와 유사한 방향성 인트라 예측 모드로 설정할 수 있다.

[0333]

candIntraPredA와 유사한 방향성 인트라 예측 모드는 모듈로 연산 및 오프셋을 이용하여 유도될 수 있다. 이때, 모듈로 연산에 이용되는 상수는 현재 블록의 형태에 따라 상이할 수 있다. 또한, 현재 블록의 형태에 따라, candIntraPredA와 유사한 방향성 인트라 예측 모드를 유도하는데 이용되는 오프셋을 상이하게 설정할 수 있다. 일 예로, 현재 블록이 너비가 높이보다 큰 비정방형인 경우, 오프셋 2를 이용하여 candIntraPredA와 유사한 방향성 인트라 예측 모드를 유도할 수 있다. 반면, 현재 블록이 높이가 너비보다 큰 비정방형인 경우, 오프셋 2 및 -8을 이용하여 candIntraPredA와 유사한 방향성 인트라 예측 모드를 유도할 수 있다.

- [0334] 또는, candIntraPredX가 가장 큰 혹은 가장 작은 인덱스를 갖는 와이드 앵글 인트라 예측 모드인지 여부를 고려하여, MPM들을 유도할 수 있다.
- [0335] 표 8은 와이드 앵글 인트라 예측 모드 인덱스를 고려하여 MPM을 유도하는 일 예를 나타낸 것이다.

표 8

[0336]

```

- If candIntraPredModeB is equal to candIntraPredModeA, the following applies:
-- If candIntraPredModeA is less than 2 (i.e., equal to INTRA_PLANAR or INTRA_DC),
candModeList[x] with x = 0...2 is derived as follows:
candModeList[0] = INTRA_PLANAR
candModeList[1] = INTRA_DC
candModeList[2] = INTRA_ANGULAR50
-- otherwise candModeList[x] with x = 0..2 is derived as follows:
i) If (candIntraPredModeA < 0 && candIntraPredModeA != -10)
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA - 3) % 84)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)
ii) otherwise, if (candIntraPredModeA = -10)
candModeList[0] = candIntraPredModeA
candModeList[1] = -1
candModeList[2] = -9
iii) otherwise, if (candIntraPredModeA > 67 && candIntraPredModeA < 76)
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)
iv) otherwise, candIntraPredModeA is equal to 67
candModeList[0] = candIntraPredModeA
candModeList[1] = 66
candModeList[2] = 68
v) otherwise, candIntraPredModeA is equal to 77
candModeList[0] = candIntraPredModeA
candModeList[1] = 76
candModeList[2] = 67
vi) otherwise, if (candIntraPredModeA >= 0 && candIntraPredModeA < 67)
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 61) % 64)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 64)
- Otherwise (candIntraPredModeB is not equal to candIntraPredModeA), the following applies:
-- candModeList[0] and candModeList[1] are derived as follows:
candModeList[0] = candIntraPredModeA
candModeList[1] = candIntraPredModeB
-- If neither of candModeList[0] and candModeList[1] is equal to INTRA_PLANAR,
candModeList[2] is set to INTRA_PLANAR
-- Otherwise, if neither of candModeList[0] and candModeList[1] is equal to INTRA_DC,
candModeList[2] is set equal to INTRA_DC,
-- Otherwise, candModeList[2] is set to INTRA_ANGULAR 50.

```

- [0337] candIntraPredA와 candIntraPredB는 동일한 것으로 가정한다. 설명의 편의를 위해, 좌측 하단 대각 방향의 인트라 예측 모드보다 인덱스 값이 작은 와이드 앵글 인트라 예측 모드들을 하단 방향 와이드 앵글 인트라 예측 모드들이라 호칭하고, 우측 상단 대각 방향의 인트라 예측 모드보다 인덱스 값이 큰 와이드 앵글 인트라 예측 모드들을 우측 방향 와이드 앵글 인트라 예측 모드들이라 호칭하기로 한다. candIntraPredA가 하단 방향 와이드 앵글 인트라 예측 모드인 경우, candIntraPredA와 이에 유사한 방향성 인트라 예측 모드를 MPM들로 설정할 수 있다. 이때, candIntraPredA가 가장 작은 값을 갖는 하단 방향 와이드 앵글 인트라 예측 모드인 경우, 기 정의된 인덱스 값을 갖는 하단 방향 와이드 앵글 인트라 예측 모드를 MPM으로 설정할 수 있다. 여기서, 기 정의된 인덱스

스는 하단 방향 와이드 앵글 인트라 예측 모드들 중 값이 가장 큰 인덱스 일 수 있다. 일 예로, candIntraPredA가 -10인 경우, MPM[0], MPM[1] 및 MPM[2]는 각각 -10, -1, -9로 설정될 수 있다.

[0338] candIntraPredA가 우측 방향 와이드 앵글 인트라 예측 모드인 경우, candIntraPredA와 이에 유사한 방향성 인트라 예측 모드를 MPM들로 설정할 수 있다. 이때, candIntraPredA가 가장 큰 값을 갖는 우측 방향 와이드 앵글 인트라 예측 모드인 경우, 기 정의된 인덱스 값을 갖는 우측 방향 와이드 앵글 인트라 예측 모드를 MPM으로 설정할 수 있다. 여기서, 기 정의된 인덱스는 우측 방향 와이드 앵글 인트라 예측 모드들 중 값이 가장 작은 인덱스 일 수 있다. 일 예로, candIntraPredA가 77인 경우, MPM[0], MPM[1] 및 MPM[2]는 각각 77, 76, 67일 수 있다.

[0339] 또는, candIntraPredA의 인덱스에 1을 차분 또는 가산하여 유도된 인덱스가 인트라 예측 모드들의 인덱스 중 가장 작은값보다 작은 경우 또는 가장 큰 값보다 큰 경우, 디폴트 모드를 MPM으로 설정할 수 있다. 여기서, 디폴트 모드는 플래너, DC, 수직 방향 인트라 예측 모드, 수평 방향 인트라 예측 모드 또는 대각 방향 인트라 예측 모드 중 적어도 하나를 포함할 수 있다.

[0340] 또는, candIntraPredA의 인덱스에 1을 차분 또는 가산하여 유도된 인덱스가 인트라 예측 모드들의 인덱스 중 가장 작은값보다 작은 경우 또는 가장 큰 값보다 큰 경우, candIntraPredA와 반대 방향인 인트라 예측 모드 또는 상기 반대 방향인 인트라 예측 모드에 유사한 인트라 예측 모드를 MPM으로 설정할 수 있다.

[0341] 또는, 현재 블록의 형태와 주변 블록의 형태를 고려하여, MPM 후보를 유도할 수 있다. 일 예로, 현재 블록과 이웃 블록이 모두 비정방형인 경우와 현재 블록은 정방형이나 이웃 블록은 비정방형인 경우의 MPM 유도 방법은 상이할 수 있다.

[0342] 현재 블록의 크기, 현재 블록의 형태, 이웃 블록의 크기 또는 이웃 블록의 형태 중 적어도 하나를 고려하여, MPM 리스트 내 MPM들을 재정렬(또는 리오더링)할 수 있다. 여기서, 재정렬은 MPM들 각각에 할당된 인덱스를 재할당하는 것을 나타낸다. 일 예로, 현재 블록과 동일한 크기 또는 동일한 형태인 이웃 블록의 인트라 예측 모드와 동일한 MPM에 더 작은 인덱스를 할당할 수 있다.

[0343] 좌측 이웃 블록의 인트라 예측 모드 candIntraPredA 및 상단 이웃 블록의 인트라 예측 모드 candIntraPredB가 각각 MPM[0] 및 MPM[1]로 설정되는 것으로 가정한다.

[0344] 현재 블록 및 상단 이웃 블록이 너비가 높이보다 큰 비정방형인 경우, 상단 이웃 블록의 인트라 예측 모드 candIntraPredB가 더 작은 인덱스를 갖도록 MPM들을 재정렬할 수 있다. 즉, candIntraPredB를 MPM[0]로 재정렬하고, candIntraPredA를 MPM[1]로 재정렬할 수 있다.

[0345] 또는, 현재 블록 및 상단 이웃 블록이 높이가 너비보다 큰 비정방형인 경우, 상단 이웃 블록의 인트라 예측 모드 candIntraPredB가 더 작은 인덱스를 갖도록 MPM들을 재정렬할 수 있다. 즉, candIntraPredB를 MPM[0]로 재정렬하고, candIntraPredA를 MPM[1]로 재정렬할 수 있다.

[0346] 또는, 현재 블록 및 상단 이웃 블록이 정방형인 경우, 상단 이웃 블록의 인트라 예측 모드 candIntraPredB가 더 작은 인덱스를 갖도록 MPM들을 재정렬할 수 있다. 즉, candIntraPredB를 MPM[0]로 재정렬하고, candIntraPredA를 MPM[1]로 재정렬할 수 있다.

[0347] MPM들을 재정렬하는 대신, 최초 candIntraPredX를 MPM에 할당할 때, 현재 블록의 크기, 현재 블록의 형태, 이웃 블록의 크기 또는 이웃 블록의 형태 중 적어도 하나를 고려할 수도 있다.

[0348] 현재 블록의 크기 또는 형태에 기초하여, MPM들을 재정렬할 수도 있다. 일 예로, 현재 블록이 너비가 높이보다 큰 비정방형인 경우, 내림 차순으로 MPM들을 재정렬할 수 있다. 반면, 현재 블록이 높이가 너비보다 큰 비정방형인 경우, 오름 차순으로 MPM들을 재정렬할 수 있다.

[0350] 원본 영상에서 예측 영상을 차분하여 유도된 잔차 영상을 유도할 수 있다. 이때, 잔차 영상을 주파수 도메인으로 변경하였을 때, 주파수 성분들 중 고주파 성분들을 제거하더라도, 영상의 주관적 화질은 크게 떨어지지 않는다. 이에 따라, 고주파 성분들의 값을 작게 변환하거나, 고주파 성분들의 값을 0으로 설정한다면, 시각적 왜곡이 크게 발생하지 않으면서도 압축 효율을 증가시킬 수 있는 효과가 있다. 위 특성을 반영하여, 잔차 영상을 2차원 주파수 성분들로 분해하기 위해 현재 블록을 변환할 수 있다. 상기 변환은 DCT(Discrete Cosine Transform) 또는 DST(Discrete Sine Transform) 등의 변환 기법을 이용하여 수행될 수 있다.

[0351] DCT는 코사인 변환을 이용하여, 잔차 영상을 2차원 주파수 성분으로 분해(또는 변환)하는 것이고, DST는 사인

변환을 이용하여, 잔차 영상을 2차원 주파수 성분으로 분해(또는 변환)하는 것이다. 잔차 영상의 변환 결과, 주파수 성분들은 기저 영상으로 표현될 수 있다. 일 예로, NxN 크기의 블록에 대해 DCT 변환을 수행하는 경우, N2 개의 기본 패턴 성분이 획득될 수 있다. 변환을 통해 NxN 크기 블록에 포함된 기본 패턴 성분들 각각의 크기가 획득될 수 있다. 이용된 변환 기법에 따라, 기본 패턴 성분의 크기를 DCT 계수 또는 DST 계수라 호칭할 수 있다.

- [0352] 변환 기법 DCT는 0이 아닌 저주파 성분들이 많이 분포하는 영상을 변환하는데 주로 이용된다. 변환 기법 DST는 고주파 성분들이 많이 분포하는 영상에 주로 이용된다.
- [0353] DCT 또는 DST 이외의 변환 기법을 사용하여 잔차 영상을 변환할 수도 있다.
- [0354] 이하, 잔차 영상을 2차원 주파수 성분들로 변환하는 것을 2차원 영상 변환이라 호칭하기로 한다. 아울러, 변환 결과 획득된 기본 패턴 성분들의 크기를 변환 계수라 호칭하기로 한다. 일 예로, 변환 계수는 DCT 계수 또는 DST 계수를 의미할 수 있다. 후술될 제1 변환 및 제2 변환이 모두 적용된 경우, 변환 계수는 제2 변환의 결과로 생성된 기본 패턴 성분의 크기를 의미할 수 있다.
- [0355] 변환 기법은 블록 단위로 결정될 수 있다. 변환 기법은 현재 블록의 예측 부호화 모드, 현재 블록의 크기 또는 현재 블록의 크기 중 적어도 하나를 기초로 결정될 수 있다. 일 예로, 현재 블록이 인트라 예측 모드로 부호화되고, 현재 블록의 크기가 NxN보다 작은 경우에는 변환 기법 DST를 사용하여 변환이 수행될 수 있다. 반면, 상기 조건을 만족하지 않는 경우, 변환 기법 DCT를 사용하여 변환이 수행될 수 있다.
- [0356] 잔차 영상 중 일부 블록에 대해서는 2차원 영상 변환이 수행되지 않을 수도 있다. 2차원 영상 변환을 수행하지 않는 것을 변환 스킵(Transform Skip)이라 호칭할 수 있다. 변환 스킵이 적용된 경우, 변환이 수행되지 않는 잔차값들을 대상으로 양자화가 적용될 수 있다.
- [0357] DCT 또는 DST를 이용하여 현재 블록을 변환한 뒤, 변환된 현재 블록을 다시 변환할 수 있다. 이때, DCT 또는 DST에 기초한 변환을 제1 변환이라 정의하고, 제1 변환이 적용된 블록을 다시 변환하는 것을 제2 변환이라 정의할 수 있다.
- [0358] 제1 변환은 복수개의 변환 코어 후보들 중 어느 하나를 이용하여 수행될 수 있다. 일 예로, DCT2, DCT8 또는 DCT7 중 어느 하나를 이용하여 제1 변환이 수행될 수 있다.
- [0359] 수평 방향 및 수직 방향에 대해 상이한 변환 코어가 사용될 수도 있다. 수평 방향의 변환 코어 및 수직 방향의 변환 코어의 조합을 나타내는 정보가 비트스트림을 통해 시그널링될 수도 있다.
- [0360] 제1 변환 및 제2 변환의 수행 단위가 상이할 수 있다. 일 예로, 8x8 블록에 대해 제1 변환을 수행하고, 변환된 8x8 블록 중 4x4 크기의 서브 블록에 대해 제2 변환을 수행할 수 있다. 이때, 제2 변환이 수행되지 않는 잔여 영역들의 변환 계수를 0으로 설정할 수도 있다.
- [0361] 또는, 4x4 블록에 대해 제1 변환을 수행하고, 변환된 4x4 블록을 포함하는 8x8 크기의 영역에 대해 제2 변환을 수행할 수도 있다.
- [0362] 제2 변환의 수행 여부를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다.
- [0363] 복호화기에서는 제2 변환의 역변환(제2 역변환)을 수행하고, 그 수행 결과에 제1 변환의 역변환(제1 역변환)을 수행할 수 있다. 상기 제2 역변환 및 제1 역변환의 수행 결과, 현재 블록에 대한 잔차 신호들이 획득될 수 있다.
- [0364] 양자화는 블록의 에너지를 줄이기 위한 것으로, 양자화 과정은 변환 계수를 특정 상수값으로 나누는 과정을 포함한다. 상기 상수값은 양자화 파라미터에 의해 유도될 수 있고, 양자화 파라미터는 1부터 63사이의 값으로 정의될 수 있다.
- [0365] 부호화기에서 변환 및 양자화를 수행하면, 복호화기는 역양자화 및 역변환을 통해 잔차 블록을 획득할 수 있다. 복호화기에서는 예측 블록과 잔차 블록을 더하여, 현재 블록에 대한 복원 블록을 획득할 수 있다.
- [0366] 현재 블록의 복원 블록이 획득되면, 인루프 필터링(In-loop filtering)을 통해 양자화 및 부호화 과정에서 발생하는 정보의 손실을 줄일 수 있다. 인루프 필터는 디블록킹 필터(Deblocking filter), 샘플 적응적 오프셋 필터 (Sample Adaptive Offset filter, SAO) 또는 적응적 루프 필터(Adaptive Loop Filter, ALF) 중 적어도 하나를 포함할 수 있다. 이하, 인루프 필터가 적용되기 전의 복원 블록을 제1 복원 블록이라 호칭하고, 인루프 필터가

적용된 이후의 복원 블록을 제2 복원 블록이라 호칭하기로 한다.

[0367] 제1 복원 블록에 디블록킹 필터, SAO 또는 ALF 중 적어도 하나를 적용하여 제2 복원 블록을 획득할 수 있다. 이 때, SAO 또는 ALF는 디블록킹 필터가 적용된 이후에 적용될 수 있다.

[0368] 디블록킹 필터는 블록 단위로 양자화를 수행함에 따라 발생하는 블록의 경계에서의 화질 열화(Blocking Artifact)를 완화시키기 위한 것이다. 디블록킹 필터를 적용하기 위해, 제1 복원 블록과 이웃 복원 블록 사이의 블록 강도(Blocking Strength, BS)를 결정할 수 있다.

[0369] 도 30은 블록 강도를 결정하는 과정을 나타낸 흐름도이다.

[0370] 도 30에 도시된 예에서, P는 제1 복원 블록을 나타내고, Q는 이웃 복원 블록을 나타낸다. 여기서, 이웃 복원 블록은 현재 블록의 좌측 또는 상단에 이웃하는 것일 수 있다.

[0371] 도 30에 도시된 예에서는, P 및 Q의 예측 부호화 모드, 0이 아닌 변환 계수가 포함되어 있는지 여부, 동일한 참조 픽처를 이용하여 인터 예측되었는지 여부 또는 움직임 벡터들의 차분값이 문턱값 이상인지 여부를 고려하여 블록 강도가 결정되는 것으로 도시되었다.

[0372] 블록 강도에 기초하여, 디블록킹 필터의 적용 여부가 결정될 수 있다. 일 예로, 블록 강도가 0인 경우에는 필터링이 수행되지 않을 수 있다.

[0373] SAO는 주파수 영역에서 양자화를 수행함에 따라 발생하는 링잉 현상(Ringing Artifact)을 완화시키기 위한 것이다. SAO는 제1 복원 영상의 패턴을 고려하여 결정되는 오프셋을 가산 또는 감산함으로써 수행될 수 있다. 오프셋의 결정 방법은 에지 오프셋(Edge Offset, EO) 또는 밴드 오프셋(Band Offset)을 포함한다. EO는 주변 화소들의 패턴에 따라, 현재 샘플의 오프셋을 결정하는 방법을 나타낸다. BO는 영역 내 비슷한 밝기 값을 갖는 화소들의 집합에 대해 공통의 오프셋을 적용하는 방법을 나타낸다. 구체적으로, 화소 밝기를 32개의 균등한 구간으로 나누고, 유사한 밝기 값을 갖는 화소들을 하나의 집합으로 설정할 수 있다. 일 예로, 32개의 밴드를 중 인접한 4개의 밴드를 하나의 그룹으로 설정하고, 4개 밴드에 속한 샘플들에는 동일한 오프셋 값을 적용할 수 있다.

[0374] ALF는 제1 복원 영상 또는 디블록킹 필터가 적용된 복원 영상에 기 정의된 크기/모양의 필터를 적용하여 제2 복원 영상을 생성하는 방법이다. 하기 수학식 21은 ALF의 적용 예를 나타낸다.

수학식 21

$$R'(i, j) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}} f(k, l) \cdot R(i + k, j + l)$$

[0375]

[0376] 픽처, 코딩 트리 유닛, 코딩 블록, 예측 블록 또는 변환 블록 단위로, 기 정의된 필터 후보들 중 어느 하나를 선택할 수 있다. 각각의 필터 후보들은 크기 또는 모양 중 어느 하나가 상이할 수 있다.

[0377] 도 31은 기 정의된 필터 후보들을 나타낸다.

[0378] 도 31에 도시된 예에서와 같이, 5x5, 7x7 또는 9x9 크기의 다이아몬드 형태 중 적어도 하나를 선택할 수 있다.

[0379] 크로마 성분에 대해서는 5x5 크기의 다이아몬드 형태만이 사용될 수 있다.

[0381] 파노라믹 비디오, 360도 비디오 또는 4K/8K UHD(Ultra High Definition) 영상 등 고해상도 영상의 실시간 또는 저지연 부호화를 위해 하나의 픽처를 복수개의 영역으로 나누고, 복수개의 영역을 병렬로 부호화/복호화하는 방안을 고려할 수 있다. 이를 위해, 픽처를 병렬 부호화/복호화의 기본 단위인 타일들로 분할하고, 타일들 각각을 병렬처리할 수 있다.

[0382] 타일은 직사각형 형태를 갖도록 제한될 수 있다. 타일을 부호화/복호화함에 있어서, 다른 타일의 데이터는 이용하지 않는다. 타일 단위로 CABAC(Context adaptive Binary Arithmetic Coding) 컨텍스트의 확률 테이블을 초기화 할 수 있으며, 타일들의 경계에서는 인루프 필터가 적용되지 않도록 설정할 수 있다.

- [0383] 도 32는 픽처가 복수의 타일들로 분할된 예를 나타낸다.
- [0384] 타일은 적어도 하나의 코딩 트리 유닛을 포함하고, 타일의 경계는 코딩 트리 유닛의 경계와 일치한다.
- [0385] 도 32에 도시된 예에서와 같이, 픽처를 복수개의 타일 세트로 분할할 수 있다. 픽처를 복수개의 타일 세트로 분할하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다.
- [0386] 픽처의 분할 형태에 따라, 픽처 바운더리를 제외한 모든 영역에서 타일들이 동일한 크기를 가질 수 있다.
- [0387] 또는, 수평 방향으로 인접하는 타일들이 갖은 높이를 갖도록 픽처를 분할하거나, 수직 방향으로 인접하는 타일들이 갖은 너비를 갖도록 픽처를 분할할 수 있다.
- [0388] 픽처를 가로지르는 수직선 또는 수평선 중 적어도 하나를 이용하여 픽처를 분할함에 따라, 타일들 각각은 상이한 열(Column) 및/또는 행(Row)에 속한다. 후술되는 실시예에서는, 타일이 속한 열을 타일 열이라 호칭하고, 타일이 속한 행을 타일 행이라 호칭하기로 한다.
- [0389] 픽처가 타일로 분할되는 양상을 결정하기 위한 정보가 비트스트림을 통해 시그널링될 수 있다. 상기 정보는 픽처 파라미터 세트 또는 시퀀스 파라미터 세트를 통해 부호화되어 시그널링될 수 있다. 상기 정보는 픽처 내 타일의 개수를 결정하기 위한 것으로서, 타일 행의 개수를 나타내는 정보 및 타일 열의 개수를 나타내는 정보를 포함할 수 있다. 일 예로, 신택스 요소 `num_tile_columns_minus1`은 타일 열의 개수에서 1을 차감한 값을 나타내고, 신택스 요소 `num_tile_rows_minus1`은 타일 행의 개수에서 1을 차감한 값을 나타낸다.
- [0390] 도 32에 도시된 예에서, 타일 열의 개수는 4개이고, 타일 행의 개수는 3개 이므로, `num_tile_columns_minus1`은 3을 나타내고, `num_tile_rows_minus1`은 2를 나타낼 수 있다.
- [0391] 픽처가 복수의 타일들로 분할되는 경우, 타일의 크기를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 픽처가 복수의 타일 열로 분할되는 경우, 각 타일 열의 너비를 나타내는 정보가 비트스트림을 통해 시그널링되고, 픽처가 복수의 타일 행으로 분할되는 경우, 각 타일 행의 높이를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 각 타일 열에 대해 타일 열의 너비를 나타내는 신택스 요소 `column_width_minus1`이 부호화되어 시그널링되고, 각 타일 행에 대해 타일 행의 높이를 나타내는 신택스 요소 `row_height_minus1`이 부호화되어 시그널링될 수 있다.
- [0392] `column_width_minus1`은 타일 열의 너비에서 1을 차감한 값을 나타내고, `row_height_minus1`은 타일 행의 높이에서 1을 차감한 값을 나타낼 수 있다.
- [0393] 마지막 타일 열에 대해서는 `column_width_minus1`의 부호화가 생략되고, 마지막 타일 행에 대해서는 `row_height_minus1`의 부호화가 생략될 수 있다. 마지막 타일 열의 너비 및 마지막 행의 높이는 픽처 크기를 고려하여 유도될 수 있다.
- [0394] 복호화기는 `column_width_minus1` 및 `row_height_minus1`을 기초로, 타일의 크기를 결정할 수 있다.
- [0395] 표 9는 픽처를 타일들로 분할하기 위한 신택스 테이블을 나타낸 것이다.

표 9

...	
<code>if (tile_enabled_flag) {</code>	
<code> num_tile_columns_minus1</code>	<code>ue(v)</code>
<code> num_tile_rows_minus1</code>	<code>ue(v)</code>
<code> uniform_spacing_flag</code>	<code>u(1)</code>
<code> if (!uniform_spacing_flag) {</code>	
<code> for (i=0; i < num_tile_columns_minus1; i++)</code>	
<code> column_width_minus1[i]</code>	<code>ue(v)</code>
<code> for (i=0; i < num_tile_rows_minus1; i++)</code>	
<code> row_height_minus1[i]</code>	<code>ue(v)</code>
<code> }</code>	
<code> loop_filter_across_tiles_enabled_flag</code>	<code>u(1)</code>
<code> }</code>	
...	

- [0397] 표 9를 살펴보면, 타일 열의 개수를 나타내는 신택스 요소 `num_tile_columns_minus1` 및 타일 행의 개수를 나타내는 신택스 요소 `num_tile_rows_minus1`이 시그널링될 수 있다. 다음으로, 픽처가 균등한 크기의 타일들로 분할되는지 여부를 나타내는 신택스 요소 `uniform_spacing_flag`가 시그널링될 수 있다. `uniform_spacing_flag`가 참인 경우, 픽처 경계를 제외한 잔여 영역에서 타일들은 균등한 크기로 분할될 수 있다.
- [0398] `uniform_spacing_flag`가 거짓인 경우, 각 타일 열의 너비를 나타내는 신택스 요소 `column_width_minus1` 및 각 타일 행의 높이를 나타내는 신택스 요소 `row_height_minus1`이 시그널링될 수 있다.
- [0399] 신택스 요소 `loop_filter_across_tiles_enabled_flag`는 타일 경계에서 루프 필터를 사용하는 것이 허용되는지 여부를 나타낸다.
- [0400] 타일 열들 중 너비가 가장 작은 타일 열을 최소 너비 타일이라 호칭하고, 타일 행들 중 높이가 가장 작은 타일 행을 최소 높이 타일이라 호칭할 수 있다. 최소 너비 타일의 너비를 나타내는 정보 및 최소 높이 타일의 높이를 나타내는 정보가 비트스트림을 통해 시그널링될 수 있다. 일 예로, 신택스 요소 `min_column_width_minus1`은 최소 너비 타일의 너비에서 1을 차감한 값을 나타내고, 신택스 요소 `min_row_height_minus1`은 최소 높이 타일의 높이에 1을 차감한 값을 나타낸다.
- [0401] 타일 열 별로, 최소 타일 너비와의 차분값을 나타내는 정보가 시그널링될 수 있다. 일 예로, 신택스 요소 `diff_column_width`는 현재 타일 열과 최소 타일 열 사이의 너비 차분값을 나타낸다. 상기 너비 차분값은 코딩 트리 유닛 열 개수의 차분값으로 표현될 수도 있다. 복호화기는 `min_column_width_minus1`을 기초로 유도된 최소 너비 타일의 너비와 `diff_column_width`를 기초로 유도된 너비 차분값을 더하여 현재 타일의 너비를 유도할 수 있다.
- [0402] 또한, 타일 행 별로, 최소 타일 높이와의 차분값을 나타내는 정보가 시그널링될 수 있다. 일 예로, 신택스 요소 `diff_row_height`는 현재 타일 행과 최소 타일 행 사이의 높이 차분값을 나타낸다. 상기 높이 차분값은 코딩 트리 유닛 행 개수의 차분값으로 표현될 수도 있다. 복호화기는 `min_row_height_minus1`을 기초로 유도된 최소 높이 타일의 높이와 `diff_row_height`를 기초로 유도된 높이 차분값을 더하여 현재 타일의 높이를 유도할 수 있다.
- [0403] 표 10은 크기 차분 정보를 포함하는 신택스 테이블을 나타낸 것이다.

표 10

...	
if (<code>tile_enabled_flag</code>) {	
<code>num_tile_columns_minus1</code>	ue(v)
<code>num_tile_rows_minus1</code>	ue(v)
<code>uniform_spacing_flag</code>	u(1)
if (! <code>uniform_spacing_flag</code>) {	
<code>min_column_width_minus1</code>	
<code>min_row_height_minus1</code>	
for (i=0; i < <code>num_tile_columns_minus1</code> ; i++)	
<code>diff_column_width</code> [i]	ue(v)
for (i=0; i < <code>num_tile_rows_minus1</code> ; i++)	
<code>diff_row_height</code> [i]	ue(v)
}	
<code>loop_filter_across_tiles_enabled_flag</code>	u(1)
}	
...	

- [0405] 좌우로 인접하는 타일들의 높이가 상이하도록 픽처를 분할하거나, 상하로 인접하는 타일들의 너비가 상이하도록 픽처를 분할할 수 있다. 위와 같은 픽처 분할 방식을 플렉서블 타일(Flexible Tile) 분할 기법이라 호칭할 수 있고, 플렉서블 타일 분할 기법을 통해 분할된 타일들을 플렉서블 타일이라 호칭할 수 있다. 도 33은 플렉서블 타일 기법에 따른 픽처의 분할 양상을 나타낸 도면이다.
- [0406] 픽처를 분할하여 생성된 타일들 간의 탐색 순서는 소정의 스캔 순서를 따를 수 있다. 또한, 소정의 스캔 순서에

따라 타일들 각각에 인덱스가 할당될 수 있다.

[0407] 타일들의 스캔 순서는 래스터 스캔(Raster Scan), 대각 스캔(Diagonal Scan), 수직 방향 스캔 또는 수평 방향 스캔 중 어느 하나일 수 있다. 도 33의 (a) 내지 (d)는 각각 래스터 스캔, 대각 스캔, 수직 방향 스캔 및 수평 방향 스캔에 따라 타일들 각각에 인덱스가 할당된 예를 나타낸다.

[0408] 현재 타일의 크기 또는 위치에 따라, 다음 스캔 순서를 결정할 수도 있다. 일 예로, 현재 타일의 높이와 현재 타일의 우측에 이웃하는 타일의 높이가 상이한 경우(예컨대, 우측 이웃 타일의 높이가 현재 타일의 높이보다 더 큰 경우), 현재 타일의 하단에 이웃하는 타일과 동일한 수직 선상에 놓인 타일들 중 최좌측에 위치하는 것을 현재 타일 다음의 스캔 대상으로 결정할 수 있다.

[0409] 타일들의 스캔 순서는 꾹처 또는 시퀀스 단위로 결정될 수 있다.

[0410] 또는, 꾹처 내 첫번째 타일의 크기를 고려하여 타일들의 스캔 순서를 결정할 수 있다. 일 예로, 첫번째 타일의 너비가 높이보다 큰 경우, 타일들의 스캔 순서를 수평 스캔으로 설정할 수 있다. 첫번째 타일의 높이가 너비보다 큰 경우, 타일들의 스캔 순서를 수직 스캔으로 설정할 수 있다. 첫번째 타일의 너비가 높이와 동일한 경우, 타일들의 스캔 순서를 래스터 스캔 또는 대각 스캔으로 설정할 수 있다.

[0411] 비트스트림을 통해 타일의 총 개수를 나타내는 정보가 시그널링될 수 있다. 일 예로, 플렉서블 타일 기법이 적용되는 경우, 꾹처 내 타일들의 총 개수에 2를 차감하여 유도되는 신택스 요소 number_of_tiles_in_picture_minus2를 시그널링할 수 있다. 복호화 장치는 number_of_tiles_in_picture_minus2를 기초로, 현재 꾹처에 포함된 타일들의 개수를 인식할 수 있다.

[0412] 표 11은 타일 개수 정보를 포함하는 신택스 테이블을 나타낸 것이다.

표 11

pic_parameter_set_rbsp() {	Descriptor
...	
<u>tile_enabled_flag</u>	u(1)
...	
if(tile_enabled_flag) {	
<u>number_of_tiles_in_picture_minus2</u>	ue(v)
<u>subtile_width_minus1</u>	ue(v)
<u>subtile_height_minus1</u>	ue(v)
for (i=0; i < number_of_tiles_in_picture_minus2+2 ; i++) {	
if (i > 0)	
<u>use_previous_tile_size_flag</u>	u(1)
if (use_previous_tile_size_flag == 0) {	
<u>tile_width_minus1[i]</u>	ue(v)
<u>tile_height_minus1[i]</u>	ue(v)
}	
}	
}	
}	
}	
<u>loop_filter_across_tiles_enabled_flag</u>	u(1)
}	
...	

[0414] 타일의 크기를 부호화하는데 필요한 비트수를 줄이기 위해, 서브 타일의 크기를 나타내는 정보를 부호화하여 시그널링할 수 있다. 서브 타일은 타일을 구성하는 기본 단위로, 각 타일은 적어도 하나의 서브 타일을 포함하여 구성될 수 있다. 서브 타일은 하나 이상의 코딩 트리 유닛을 포함할 수 있다. 일 예로, 신택스 요소 subtile_width_minus1은 서브 타일의 너비에 1을 차감한 값을 나타낸다. 신택스 요소 subtile_height_minus1은 서브 타일의 높이에 1을 차감한 값을 나타낸다.

[0415] 첫번째 타일을 제외한 잔여 타일에 대해, 이전 타일과 동일한 크기를 갖는지 여부를 나타내는 정보를 부호화하여 시그널링할 수 있다. 일 예로, 신택스 요소 use_previous_tile_size_flag는 현재 타일의 크기가 이전 타일의 크기와 동일한지 여부를 나타낸다. use_previous_tile_size_flag가 참인 것은 현재 타일의 크기가 이전 타일의

크기가 동일함을 나타낸다. `use_previous_tile_size_flag`가 거짓인 경우, 현재 타일의 크기를 나타내는 정보를 부호화하여 시그널링할 수 있다. 첫번째 타일에 대해서는 `use_previous_tile_size_flag`의 부호화를 생략하고, 상기 플래그의 값을 거짓으로 설정할 수 있다.

[0416] 타일 크기를 나타내는 정보는, i번째 타일의 너비를 나타내는 선택스 요소 `tile_width_minus1[i]` 및 i번째 타일의 높이를 나타내는 선택스 요소 `tile_height_minus1[i]`를 포함할 수 있다.

[0417] 타일 크기를 나타내는 정보는 서브 타일의 크기와의 차분값을 나타낼 수 있다. 서브 타일의 크기 정보를 이용함으로써, 각 타일의 크기를 부호화하는데 필요한 비트 수를 줄여 부호화/복호화 효율을 증가시킬 수 있다. 일 예로, i번째 타일의 너비 `tileWidth`는 다음의 수학식 22를 기초로 유도되고, i번째 타일의 높이 `tileHeight`는 다음의 수학식 23을 기초로 유도될 수 있다.

수학식 22

$$tileWidth = (subtile_width_minus1 + 1) * (tile_width_minus1[i] + 1)$$

수학식 23

$$tileHeight = (subtile_height_minus1 + 1) * (tile_height_minus1[i] + 1)$$

[0419] 또는, 서브 타일 크기 정보의 부호화를 생략하고, i번째 타일의 크기를 그대로 타일 크기 정보로 부호화할 수도 있다. 서브 타일의 크기 정보는 선택적(Optional)으로 부호화될 수 있다. 서브 타일의 크기 정보가 부호화되었는지 여부를 나타내는 정보가 비디오 파라미터 세트, 시퀀스 파라미터 세트 또는 픽처 파라미터 세트를 통해 시그널링될 수 있다.

[0420] 상술한 타일 크기와 관련된 정보들을 코딩 트리 유닛의 개수를 나타내는 것으로 부호화하여 시그널링할 수도 있다. 일 예로, `column_width_minus1`, `min_column_width_minus1`, `subtile_width_minus1` 또는 `tile_width_minus1` 등은 타일에 포함된 코딩 트리 유닛 열의 개수를 나타낼 수 있다. 아울러, `diff_column_width`은 최소 너비 타일에 포함된 코딩 트리 유닛 열의 개수와 현재 타일에 포함된 코딩 트리 유닛 열의 개수와의 차분값을 나타낼 수 있다.

[0421] 또한, `row_height_minus1`, `min_row_height_minus1`, `subtile_height_minus1` 또는 `tile_height_minus1` 등은 타일에 포함된 코딩 트리 유닛 행의 개수를 나타낼 수 있다. 아울러, `diff_row_height`은 최소 높이 타일에 포함된 코딩 트리 유닛 행의 개수와 현재 타일에 포함된 코딩 트리 유닛 행의 개수와의 차분값을 나타낼 수 있다.

[0422] 복호화기는 상기 선택스 요소들을 기초로 유도된 코딩 트리 유닛 열의 개수 및/또는 코딩 트리 유닛 행의 개수와 코딩 트리 유닛의 크기를 기초로, 타일의 크기를 결정할 수 있다. 일 예로, i번째 타일의 너비는 $(tile_width_minus1[i]+1) * (코딩 트리 유닛의 너비)$ 로 설정되고, i번째 타일의 높이는 $(tile_height_minus1[i]+1) * (코딩 트리 유닛의 높이)$ 로 설정될 수 있다.

[0423] 한편, 코딩 트리 유닛의 크기를 나타내는 정보는 시퀀스 파라미터 세트 또는 픽처 파라미터 세트를 통해 시그널링될 수 있다.

[0424] 표 11에서는 현재 타일의 크기가 이전 타일과 동일한지 여부를 나타내는 선택스 요소 `use_previous_tile_size_flag`가 이용되는 것으로 설명하였다. 다른 예로, 현재 타일의 너비가 이전 타일의 너비와 동일한지 여부를 나타내는 정보 또는 현재 타일의 높이가 이전 타일의 높이와 동일한지 여부를 나타내는 정보를 부호화하여 시그널링할 수도 있다.

[0425] 표 12는 현재 타일의 너비가 이전 타일의 너비와 동일한지 여부를 나타내는 정보를 포함하는 선택스 테이블을 나타낸 것이다.

표 12

<code>pic_parameter_set_rbsp() {</code>	<code>Descriptor</code>
<code>...</code>	

<u>tiles_enabled_flag</u>	u(1)
...	
CoveredWidthByTile = 0;	
CoveredHeightByTile = 0 ;	
PrevWidthByTile = 0;	
PrevHeightByTile = 0 ;	
if(tiles_enabled_flag) {	
<u>number_of_tiles_in_picture_minus2</u>	ue(v)
for (i=0; i < number_of_tiles_in_picture_minus2+2 ; i++) {	
if (i > 0)	
<u>use_previous_tile_size_flag</u>	u(1)
if (use_previous_tile_size_flag == 0) {	
<u>use_previous_tile_width_flag</u>	ue(v)
if (use_previous_tile_width_flag == 0) {	ue(v)
<u>tile_width_minus1[i]</u>	ue(v)
<u>tile_height_minus1[i]</u>	ue(v)
} else {	
<u>tile_height_minus1[i]</u>	ue(v)
}	
}	
<u>loop_filter_across_tiles_enabled_flag</u>	u(1)
}	
...	

[0428] 선택스 요소 `use_previous_tile_width_flag`는 현재 타일의 너비가 이전 타일과 동일한지 여부를 나타낸다. `use_previous_tile_width_flag`가 참인 경우, 현재 타일의 너비는 이전 타일의 너비와 동일하게 설정될 수 있다. 이 경우, 현재 타일의 너비를 나타내는 정보의 부호화를 생략하고, 이전 타일의 너비로부터 현재 타일의 너비를 유도할 수 있다. `use_previous_tile_width_flag`가 거짓인 경우, 현재 타일의 너비를 나타내는 정보를 시그널링 할 수 있다. 일 예로, `tile_width_minus1[i]`는 i 번째 타일의 너비에 1을 차감한 값을 나타낼 수 있다.

[0429] 현재 타일의 크기가 이전 타일의 크기가 상이한 것으로 결정된 경우(예컨대, `use_previous_tile_size_flag` 값이 0인 경우)에 한하여, 선택스 요소 `use_previous_tile_width_flag`가 부호화되어 시그널링될 수도 있다.

[0430] `tile_width_minus1[i]`는 i 번째 타일에 포함된 코딩 트리 유닛 열의 개수에 1을 차분한 것일 수 있다. 복호화기에서는 `tile_width_minus1[i]`에 1을 더하여, i 번째 타일에 속한 코딩 트리 유닛 열의 개수를 유도하고, 유도된 값에 코딩 트리 유닛의 너비를 곱하여 타일 너비를 계산할 수 있다.

[0431] 표 13은 현재 타일의 높이가 이전 타일의 높이와 동일한지 여부를 나타내는 정보가 더 포함된 선택스 테이블을 나타낸 것이다.

표 13

pic_parameter_set_rbsp() {	Descriptor
...	
<u>titles_enabled_flag</u>	u(1)
...	
CoveredWidthByTile = 0;	
CoveredHeightByTile = 0 ;	
PrevWidthByTile = 0;	
PrevHeightByTile = 0 ;	
if(titles_enabled_flag) {	
<u>number_of_tiles_in_picture_minus2</u>	ue(v)
for (i=0; i < number_of_tiles_in_picture_minus2+2 ; i++) {	
if (i > 0)	
<u>use_previous_tile_size_flag</u>	u(1)

if (use_previous_tile_size_flag == 0) {	
use_previous_tile_width_flag	ue(v)
if (use_previous_tile_width_flag == 0) {	ue(v)
tile_width_minus1[i]	ue(v)
tile_width_minus1[i] +=1;	
use_previous_tile_height_flag	
if (use_previous_tile_height_flag ==0){	
tile_height_minus1[i]	ue(v)
tile_height_minus1[i] +=1;	
}	
} else {	
use_previous_tile_height_flag	
if (use_previous_tile_height_flag ==0){	
tile_height_minus1[i]	ue(v)
tile_height +=1;	
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
...	

- [0433] 신택스 요소 use_previous_tile_height_flag는 현재 타일의 높이가 이전 타일과 동일한지 여부를 나타낸다. use_previous_tile_height_flag가 참인 경우, 현재 타일의 높이는 이전 타일의 높이와 동일하게 설정될 수 있다. 이 경우, 현재 타일의 높이를 나타내는 정보의 부호화를 생략하고, 이전 타일의 높이로부터 현재 타일의 높이를 유도할 수 있다. use_previous_tile_height_flag가 거짓인 경우, 현재 타일의 높이를 나타내는 정보를 시그널링할 수 있다. 일 예로, tile_height_minus1[i]는 i번째 타일의 높이에 1을 차감한 값을 나타낼 수 있다.
- [0434] 현재 타일의 크기가 이전 타일의 크기가 상이한 것으로 결정된 경우(예컨대, use_previous_tile_size_flag 값이 0인 경우)에 한하여, 신택스 요소 use_previous_tile_height_flag가 부호화되어 시그널링될 수도 있다. 또한, 신택스 요소 use_previous_tile_height_flag는 use_previous_tile_width_flag가 거짓인 경우에 한하여 시그널링될 수 있다.
- [0435] 표 12는 use_previous_tile_width_flag가 이용되는 경우의 예를 나타내고, 표 13은 use_previous_tile_width_flag 및 use_previous_tile_height_flag가 이용되는 경우의 예를 나타낸다. 상기 표들에 나타나지는 않았지만, use_previous_tile_width_flag의 부호화를 생략하고, use_previous_tile_height_flag만을 사용할 수도 있다.
- [0436] use_previous_tile_height_flag 및 use_previous_tile_size_flag 중 어느 것을 사용할 것인지는 타일 스캔 순서, 첫 번째 타일의 너비/높이 또는 이전 타일의 너비/높이 중 적어도 하나를 기초로 결정될 수 있다. 일 예로, 타일 스캔 순서가 수직 방향인 경우에는 use_previous_tile_height_flag를 사용하는 한편, 타일 스캔 순서가, 수평 방향인 경우에는 use_previous_tile_width_flag를 사용할 수 있다. 또는, 첫 번째 타일 또는 이전 타일이 너비가 높이보다 큰 비정방형인 경우, use_previous_tile_width_flag를 사용하는 한편, 첫번째 타일 또는 이전 타일이 높이가 너비보다 큰 비정방형인 경우, use_previous_tile_height_flag를 사용할 수 있다.
- [0437] 픽처에 포함된 타일들의 개수를 시그널링하는 한편, 마지막 타일에 대해서는 타일 크기와 관련된 정보의 부호화를 생략할 수 있다.
- [0438] 표 14는 마지막 타일에 대한 타일 크기 정보의 부호화가 생략되는 예를 나타낸 것이다.

표 14

pic_parameter_set_rbsp() {	Descriptor
...	
titles_enabled_flag	u(1)
...	

if(tiles_enabled_flag) {	
number_of_tiles_in_picture_minus2	ue(v)
subtile_width_minus1	ue(v)
subtile_height_minus1	ue(v)
for (i=0; i < number_of_tiles_in_picture_minus2+1 ; i++) {	
if (i > 0)	
use_previous_tile_size_flag	u(1)
if (use_previous_tile_size_flag == 0) {	
tile_width_minus1[i]	ue(v)
tile_height_minus1[i]	ue(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
...	

- [0440] 마지막 타일을 제외한 타일들의 크기가 특정되면, 픽쳐 내 잔여 영역을 마지막 타일로 설정할 수 있다. 각각의 코딩 트리 유닛에 대해, 코딩 트리 유닛이 속하는 타일을 식별하기 위한 식별자(이하, 타일 아이디, TileID라 함)를 할당할 수 있다.
- [0441] 도 34는 각 코딩 트리 유닛에 타일 아이디가 할당되는 예를 나타낸 도면이다.
- [0442] 동일한 타일에 속하는 코딩 트리 유닛들에는 동일한 타일 아이디가 할당될 수 있다. 구체적으로, Tile N에 속한 코딩 트리 유닛 들에는 N번의 TileID가 할당될 수 있다.
- [0443] 각 코딩 트리 유닛에 할당되는 타일 아이디를 결정하기 위해, 픽쳐 내 코딩 트리 유닛의 위치를 나타내는 변수 x와 y를 결정할 수 있다. 여기서, x는 코딩 트리 유닛의 좌측 상단 샘플의 위치 (x0, y0) 중 x축 좌표를 코딩 트리 유닛의 너비로 나눈 값을 나타내고, y는 코딩 트리 유닛의 좌측 상단 샘플의 위치 (x0, y0) 중 y축 좌표를 코딩 트리 유닛의 높이로 나눈 값을 나타낸다. 구체적으로, x 및 y는 다음의 수학식 24 및 25에 의해 유도될 수 있다.

수학식 24

$$x = (x0 / (CTU \text{ 너비}))$$

수학식 25

$$y = (y0 / (CTU \text{ 높이}))$$

- [0446] 각 코딩 트리 유닛에 타일 아이디를 할당하는 것은 다음의 과정을 거쳐 수행될 수 있다.
- [0447] i) 타일 아이디 초기화
- [0448] 각 코딩 트리 유닛의 타일 아이디를 픽쳐 내 타일들의 개수에 1을 차분한 값으로 초기화할 수 있다.

표 15

[0449] If tiles_enabled_flag is equal to 1, the value of the variable tile_id and the value of the two-dimentional array TileId are specified as follows:

```
tile_id = 0
for (y=0; y<PicHeightInCtbsY; y++)
- for (x=0; x<PicWidthInCtbsY; x++)
-- TileId[x][y] = number_of_tiles_in_picture_minus2+1

tile_height_minus1[number_of_tiles_in_picture_minus2+1] = 0
tile_width_minus1[number_of_tiles_in_picture_minus2+1] = 0
```

[0450] ii) 타일 아이디 유도

표 16

[0451] The value of the variable tile_id and the value of the two-dimentional array CTU_tile assignment are derived as follows:

```
for(ctu_y=0;      ctu_y      <      (tile_height_minus1[i]+1)*(subtile_height_minus1+1)      &&
(tile_coordinate_y+ctu_y) < PicHeightInCtbsY; ctu_y++)
-   for(ctu_x=0;      ctu_x      <      (tile_width_minus1[i]+1)*(subtile_width_minus1+1)      &&
(tile_coordinate_x+ctu_x) < PicWidthInCtbsY; ctu_x++)
-- TileId[tile_coordinate_x + ctu_x] [tile_coordinate_y + ctu_y] = tile_id
tile_id++
```

[0452] 상술한 실시예에서는, 타일들의 경계에서 인루프 필터를 적용하는 것이 허용되는지 여부를 나타내는 플래그가 꾹쳐 파라미터 세트를 통해 시그널링되는 것으로 설명하였다. 다만, 모든 타일 경계에서 인루프 필터를 사용하지 않도록 설정하는 경우, 주관적 화질이 감소하고 부호화 효율이 떨어지는 문제가 발생할 수 있다. 이에, 타일 별로 인루프 필터 허용 여부를 나타내는 정보를 부호화하여 시그널링할 수 있다.

[0453] 도 35는 타일별로 인루프 필터의 적용 여부가 선택적으로 결정되는 예를 도시한 것이다.

[0454] 도 35에 도시된 예에서와 같이, 수평 경계 또는 수직 경계에서 인루프 필터(예컨대, 디블록킹 필터, SAO 및/또는 ALF)를 허용할 것인지 여부가 타일별로 결정될 수 있다.

[0455] 표 17은 인루프 필터의 허용 여부를 나타내는 정보가 타일 별로 부호화되는 예를 나타낸 것이다.

표 17

pic_parameter_set_rbsp() {	Descriptor
...	
tiles_enabled_flag	u(1)
...	
if(tiles_enabled_flag) {	
number_of_tiles_in_picture_minus2	ue(v)
subtile_width_minus1	ue(v)
subtile_height_minus1	ue(v)
for (i=0; i < number_of_tiles_in_picture_minus2+1 ; i++) {	
if (i > 0)	
use_previous_tile_size_flag	u(1)
if (use_previous_tile_size_flag == 0) {	
tile_width_minus1[i]	ue(v)
tile_height_minus1[i]	ue(v)
}	
loop_filter_across_tiles_flag[i]	u(1)

}	
}	
...	

[0457] 표 17의 예에서, 선택스 요소 `loop_filter_across_tiles_flag[i]`는 i 번째 타일에 대해 인루프 필터를 적용하는 것이 허용되는지 여부를 나타낸다. `loop_filter_across_tile_flag[i]`의 값이 1인 것은 타일 아이디가 i 인 타일의 수평 및 수직 바운더리에서 인루프 필터를 사용할 수 있음을 나타낸다. `loop_filter_across_tiles_flag[i]`의 값이 0인 것은 타일 아이디가 i 인 타일의 수평 및 수직 바운더리에서 인루프 필터를 사용할 수 없음을 나타낸다. 수평 방향 및 수직 방향 각각에 대해 인루프 필터의 허용 여부를 나타내는 정보를 부호화할 수도 있다.

[0458] 표 18은 인루프 필터의 허용 여부를 나타내는 정보가 수평 방향 및 수직 방향에 대해 개별적으로 부호화되는 예를 나타낸 것이다.

표 18

pic_parameter_set_rbsp() {	Descriptor
...	
<code>tiles_enabled_flag</code>	u(1)
...	
<code>if(tiles_enabled_flag) {</code>	
<code>number_of_tiles_in_picture_minus2</code>	ue(v)
<code>subtile_width_minus1</code>	ue(v)
<code>subtile_height_minus1</code>	ue(v)
<code>for (i=0; i < number_of_tiles_in_picture_minus2+1 ; i++) {</code>	
<code>if (i > 0)</code>	
<code>use_previous_tile_size_flag</code>	u(1)
<code>if (use_previous_tile_size_flag == 0) {</code>	
<code>tile_width_minus1[i]</code>	ue(v)
<code>tile_height_minus1[i]</code>	ue(v)
<code>}</code>	
<code>loop_filter_hor_across_tiles_flag[i]</code>	u(1)
<code>loop_filter_ver_across_tiles_flag[i]</code>	u(1)
<code>}</code>	
<code>}</code>	
...	

[0460] 표 18의 예에서, 선택스 요소 `loop_filter_hor_across_tiles_flag[i]`는 i 번째 타일에 대해 수평 방향을 가로지르는 인루프 필터를 적용하는 것이 허용되는지 여부를 나타낸다. 선택스 요소 `loop_filter_ver_across_tiles_flag[i]`는 i 번째 타일에 대해 수직 방향을 가로지르는 인루프 필터를 적용하는 것이 허용되는지 여부를 나타낸다. `loop_filter_hor_across_tile_flag[i]`의 값이 1인 것은 타일 아이디가 i 인 타일의 수평 바운더리에서 인루프 필터를 사용할 수 있음을 나타낸다. `loop_filter_ver_across_tile_flag[i]`의 값이 0인 것은 타일 아이디가 i 인 타일의 수평 바운더리에서 인루프 필터를 사용할 수 없음을 나타낸다.

[0461] `loop_filter_ver_across_tile_flag[i]`의 값이 1인 것은 타일 아이디가 i 인 타일의 수직 바운더리에서 인루프 필터를 사용할 수 있음을 나타내고, `loop_filter_ver_across_tile_flag[i]`의 값이 0인 것은 타일 아이디가 i 인 타일의 수직 바운더리에서 인루프 필터를 사용할 수 없음을 나타낸다.

[0462] 또는, 복수의 타일들을 포함하는 타일 그룹에 대해 인루프 필터가 허용되는지 여부를 나타내는 정보를 부호화하여 시그널링될 수도 있다. 타일 그룹에 포함된 복수의 타일들에 대한 인루프 필터의 허용 여부는 상기 정보에 의해 결정될 수 있다.

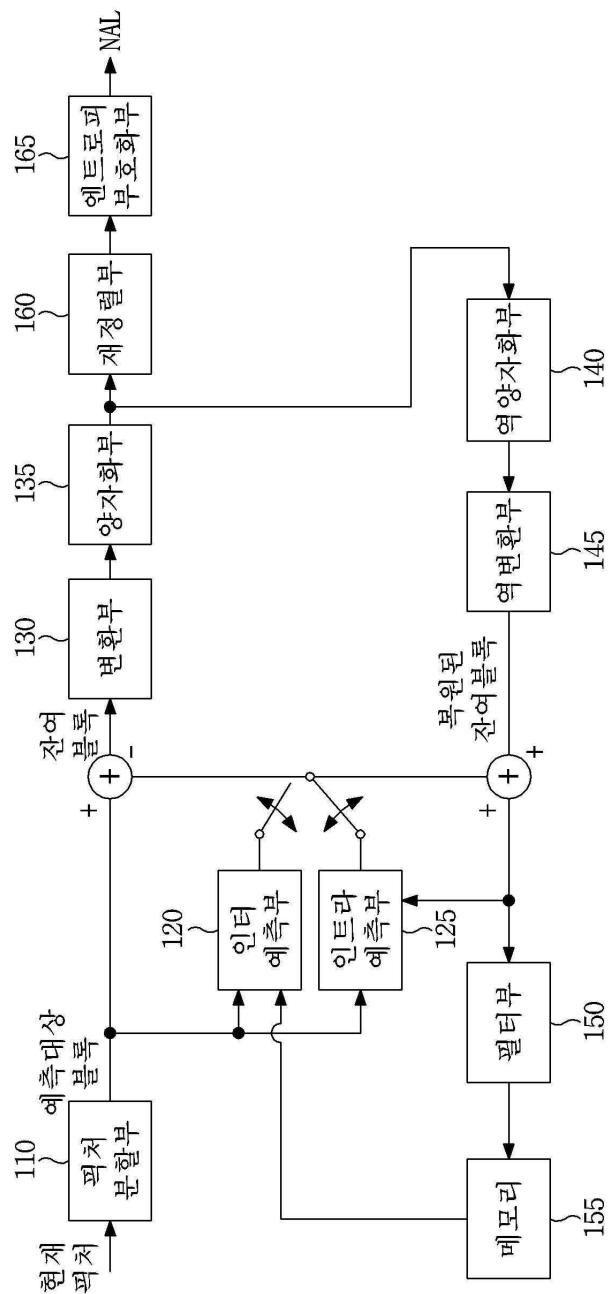
[0463] 타일 그룹을 결정하기 위해, 타일 그룹에 속한 타일의 개수, 타일 그룹의 크기 또는 꾹처의 분할 정보 중 적어도 하나가 비트스트림을 통해 시그널링될 수 있다. 또는, 부호화기 및 복호화기에서 기 정의된 크기의 영역이

타일 그룹으로 설정될 수 있다.

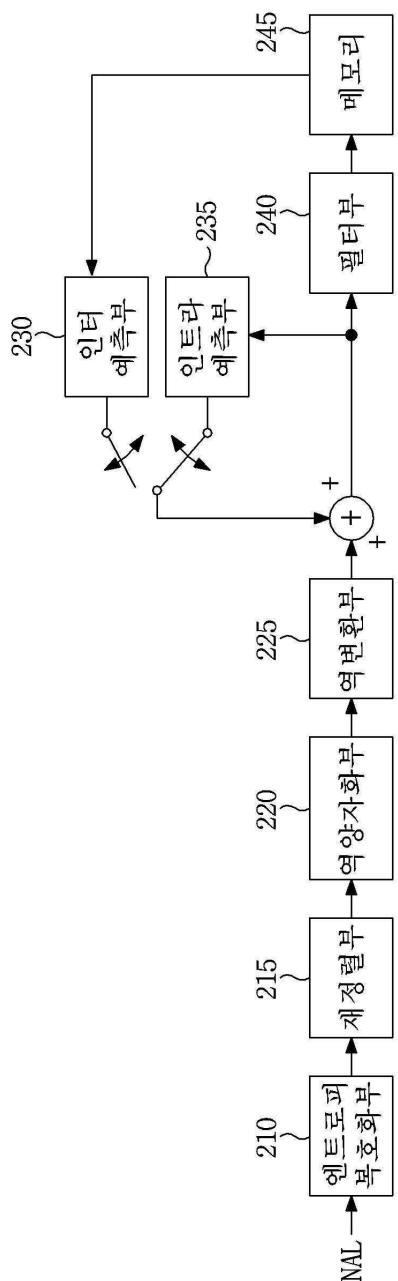
- [0464] 인루프 필터가 허용되는지 여부를 나타내는 정보의 부호화를 생략하고, 타일에 포함된 코딩 트리 유닛의 개수, 타일의 너비 또는 타일의 높이 중 적어도 하나를 기초로 인루프 필터의 허용 여부를 결정할 수 있다. 일 예로, 타일 너비가 기준값보다 작은 경우, 수평 방향에 대한 인루프 필터를 허용하고, 타일 높이가 기준값보다 작은 경우, 수직 방향에 대한 인루프 필터를 허용할 수 있다.
- [0465] 타일 경계에서 인루프 필터가 사용되는 경우, 타일에 포함된 데이터를 기초로 타일 외부의 복원 데이터를 생성할 수 있다. 이때, 타일 외부의 복원 영상은 타일에 포함된 데이터를 패딩 또는 보간하여 획득될 수 있다. 이후, 타일 외부의 복원 데이터를 이용하여, 인루프 필터를 적용할 수 있다.
- [0467] 복호화 과정 또는 부호화 과정을 중심으로 설명된 실시예들을, 부호화 과정 또는 복호화 과정에 적용하는 것은, 본 발명의 범주에 포함되는 것이다. 소정의 순서로 설명된 실시예들을, 설명된 것과 상이한 순서로 변경하는 것 역시, 본 발명의 범주에 포함되는 것이다.
- [0468] 상술한 실시예는 일련의 단계 또는 순서도를 기초로 설명되고 있으나, 이는 발명의 시계열적 순서를 한정한 것은 아니며, 필요에 따라 동시에 수행되거나 다른 순서로 수행될 수 있다. 또한, 상술한 실시예에서 블록도를 구성하는 구성요소(예를 들어, 유닛, 모듈 등) 각각은 하드웨어 장치 또는 소프트웨어로 구현될 수도 있고, 복수의 구성요소가 결합하여 하나의 하드웨어 장치 또는 소프트웨어로 구현될 수도 있다. 상술한 실시예는 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

도면

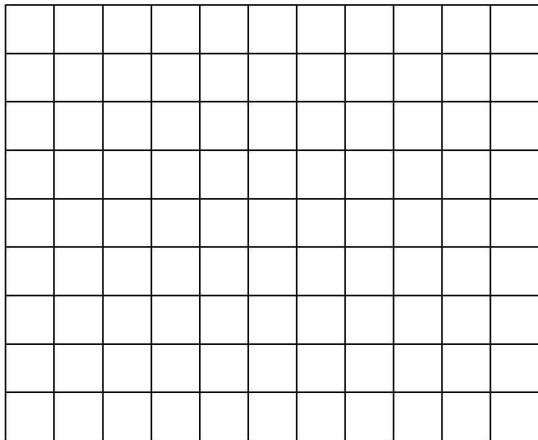
도면1



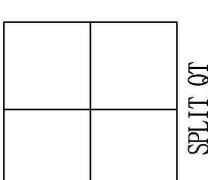
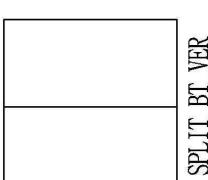
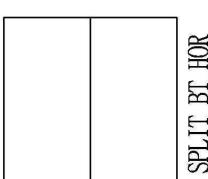
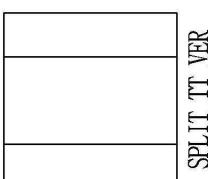
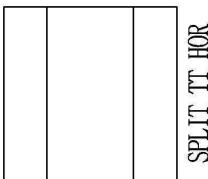
도면2



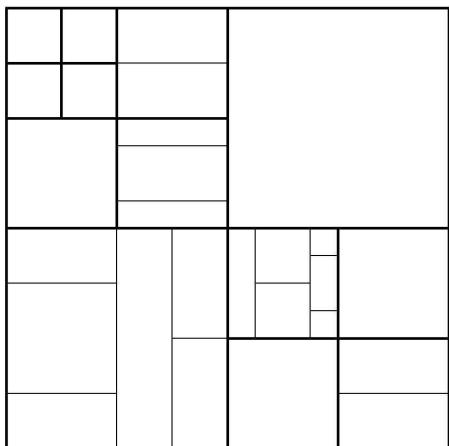
도면3



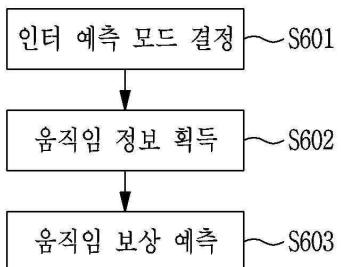
도면4



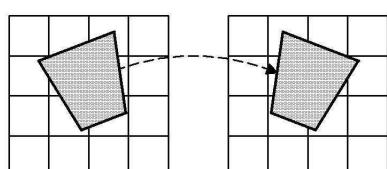
도면5



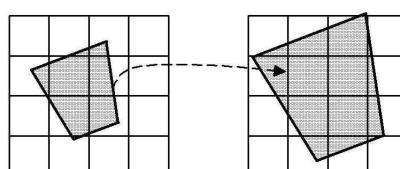
도면6



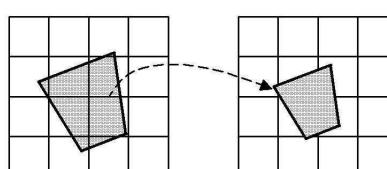
도면7



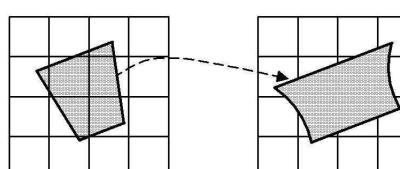
Rotation



Zoom-in

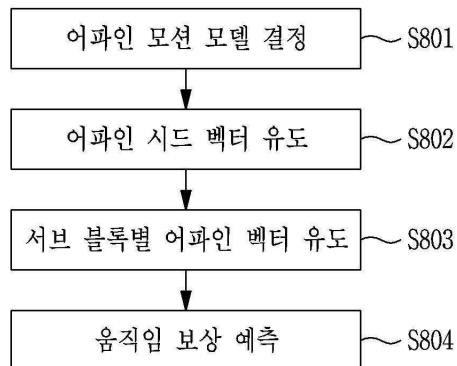


Zoom-out

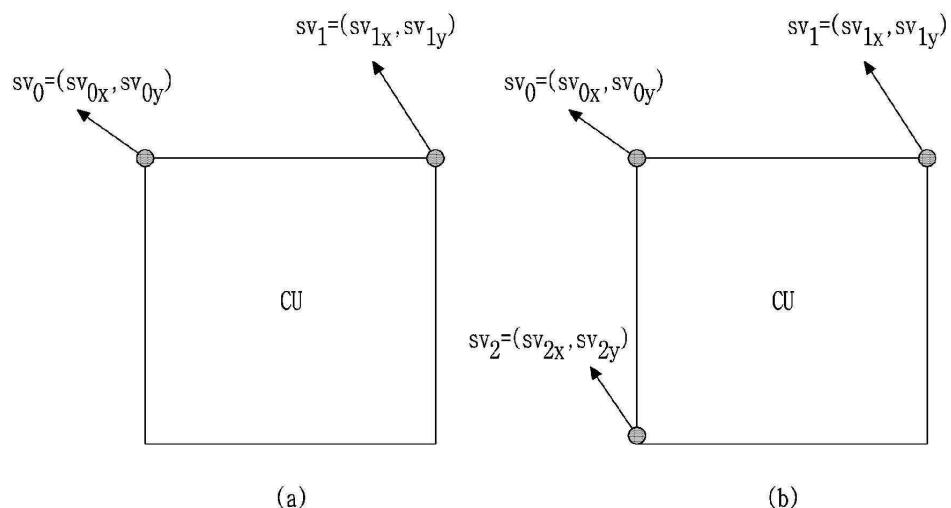


Affine 변환

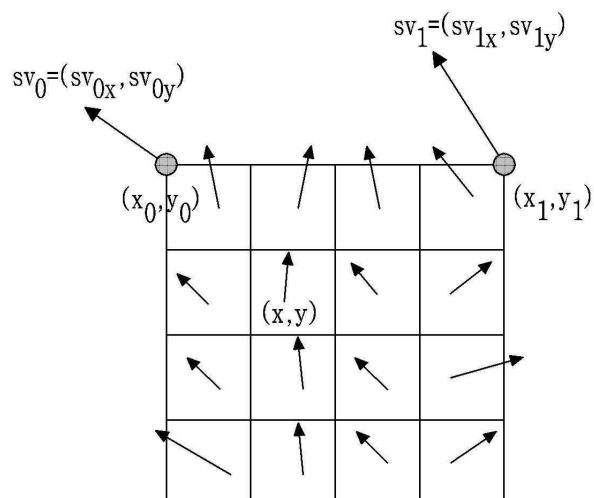
도면8



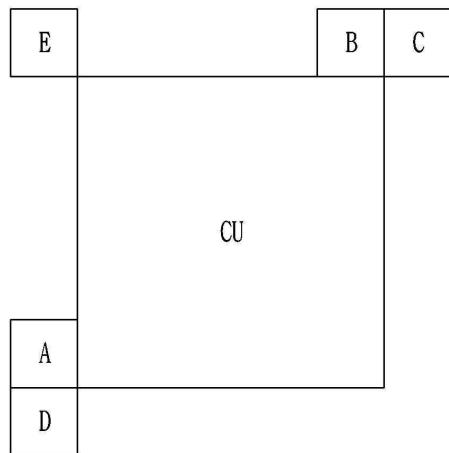
도면9



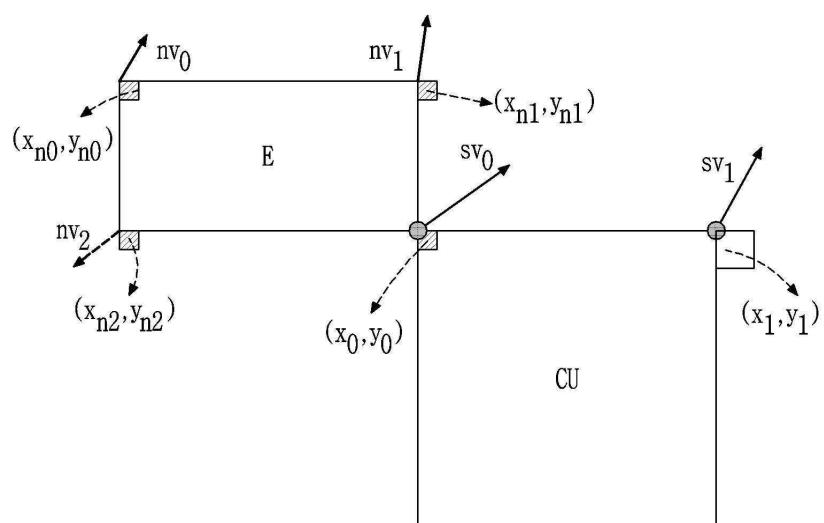
도면10



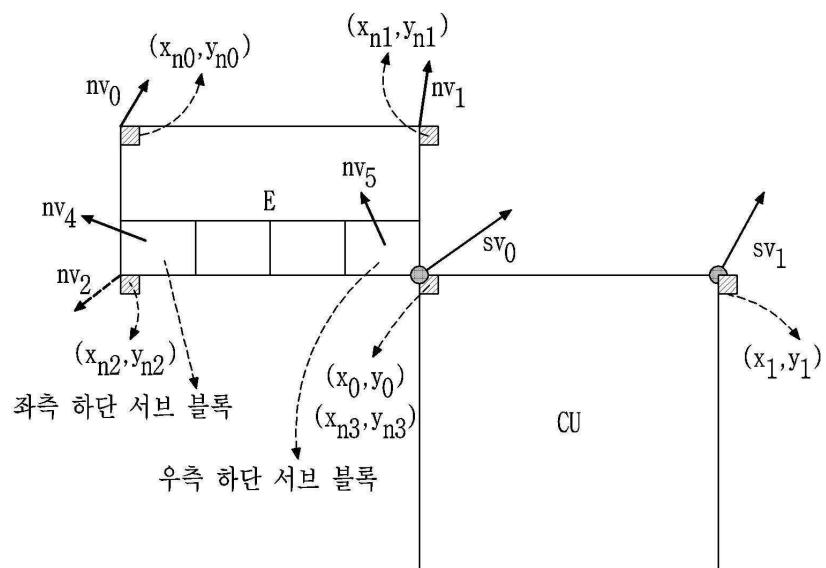
도면11



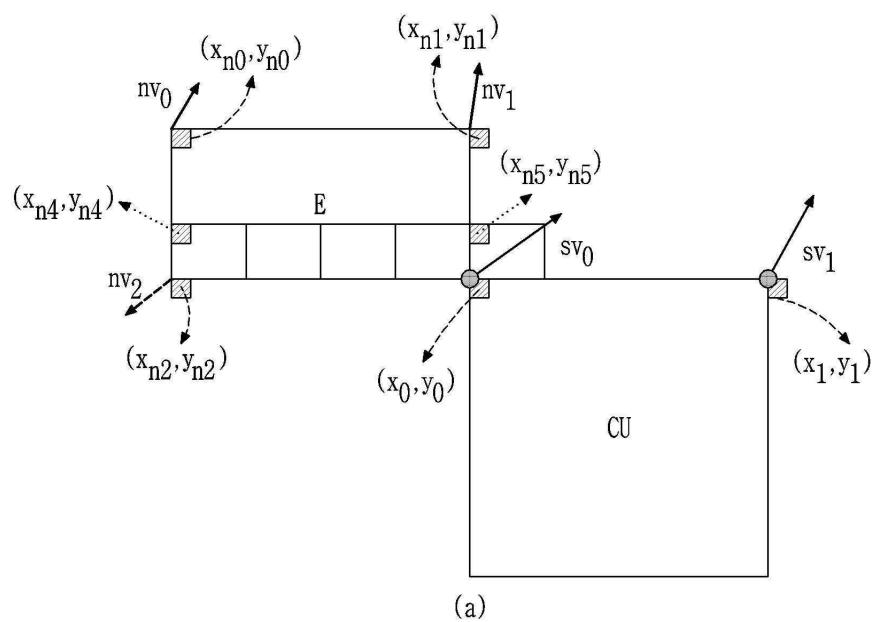
도면12



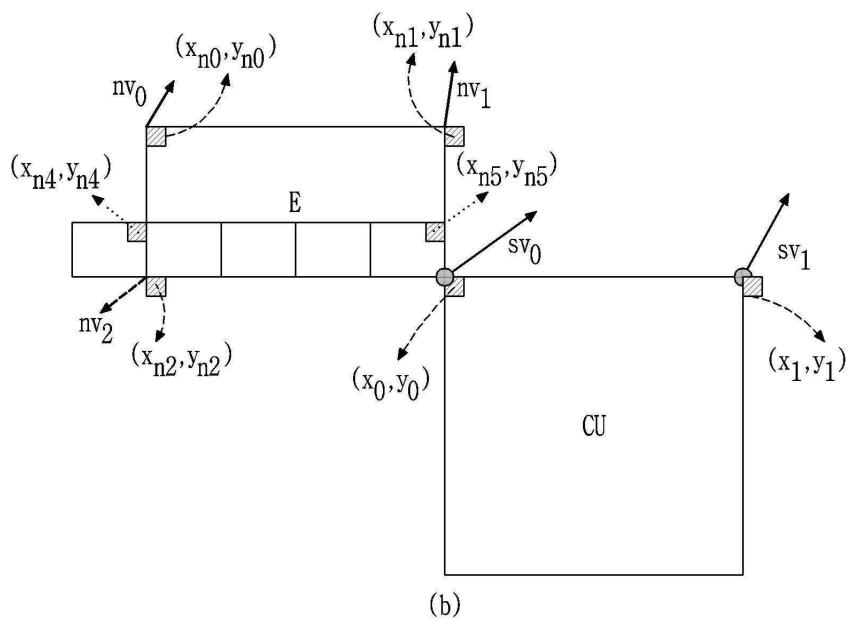
도면13



도면14

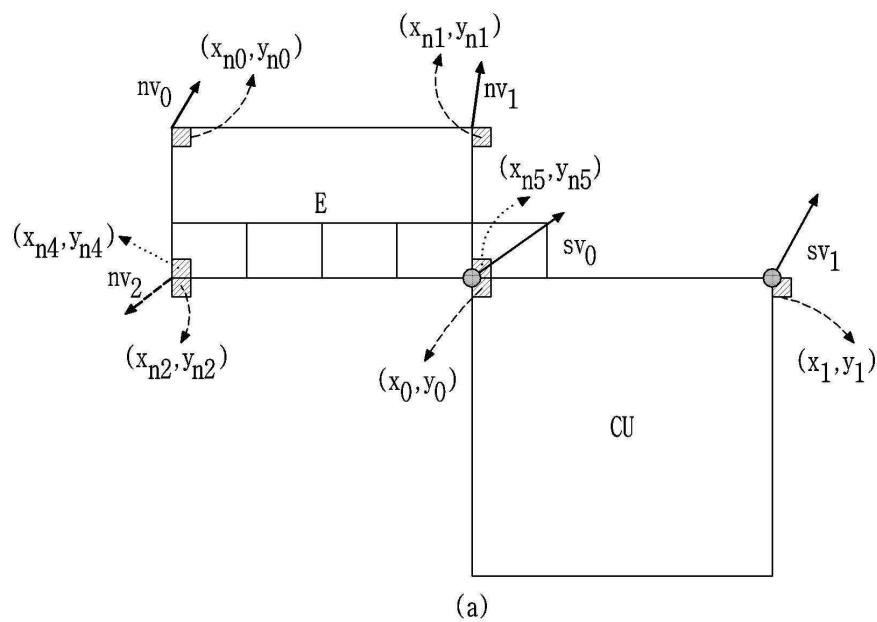


(a)

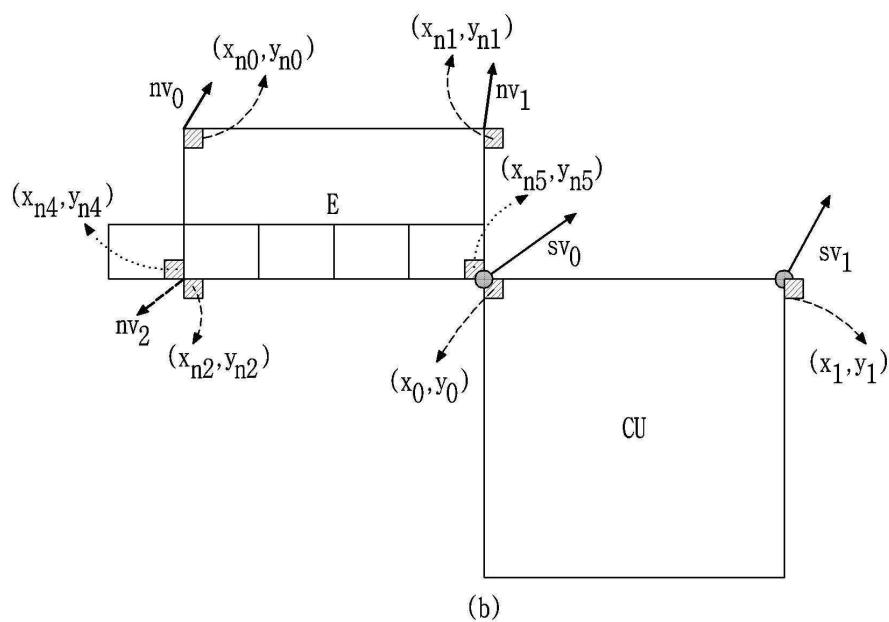


(b)

도면15

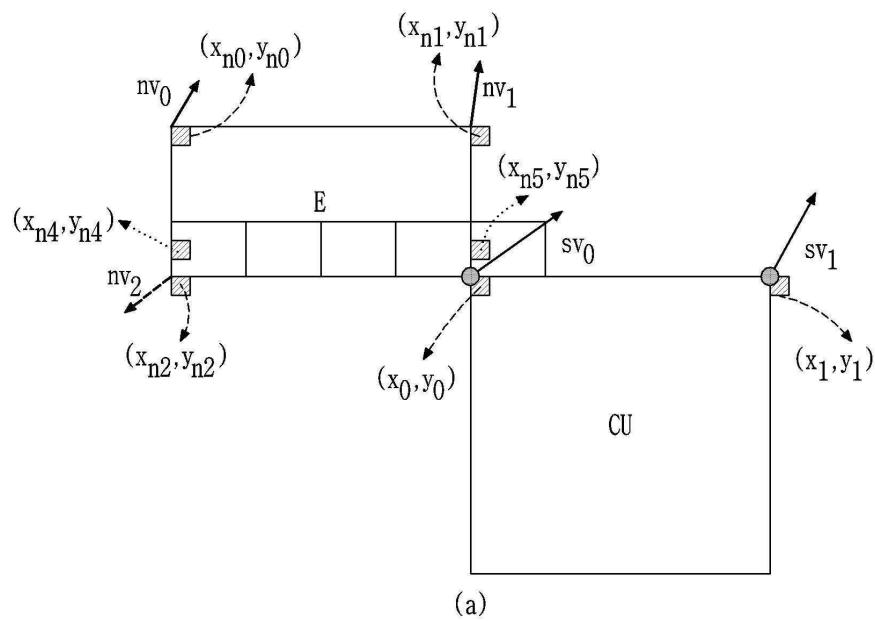


(a)

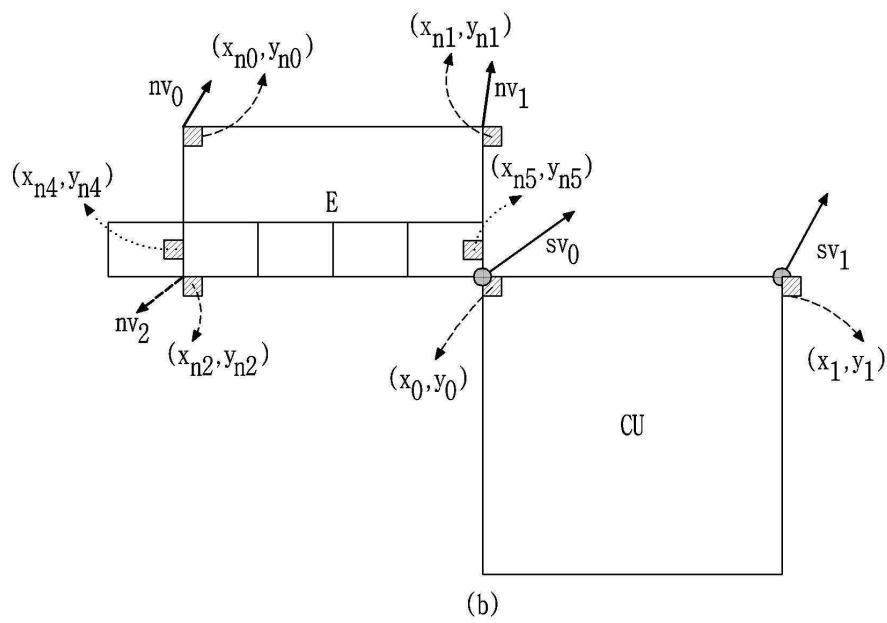


(b)

도면16

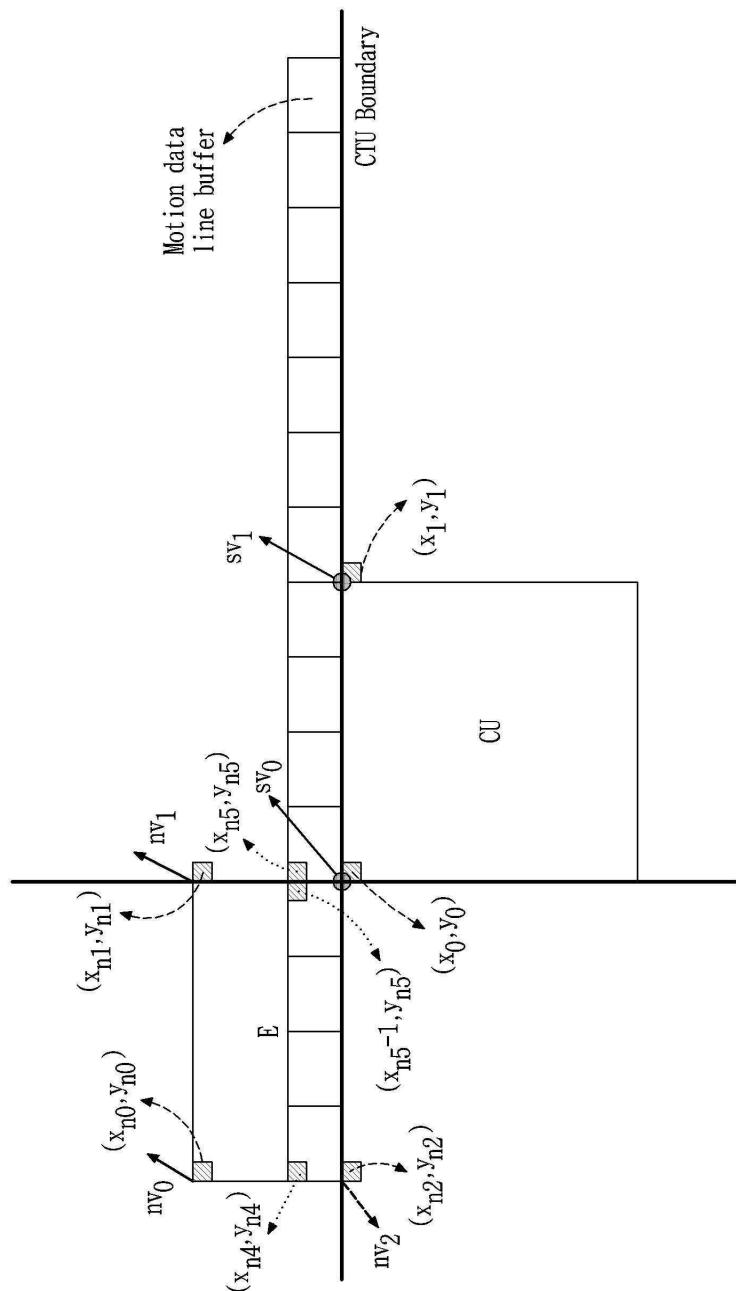


(a)

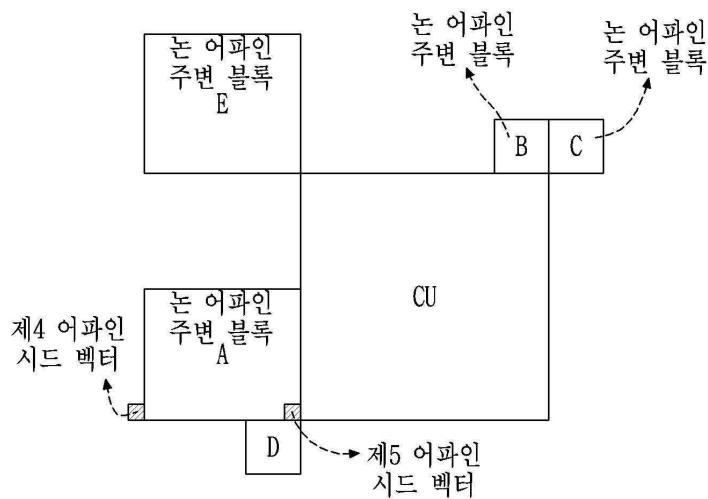


(b)

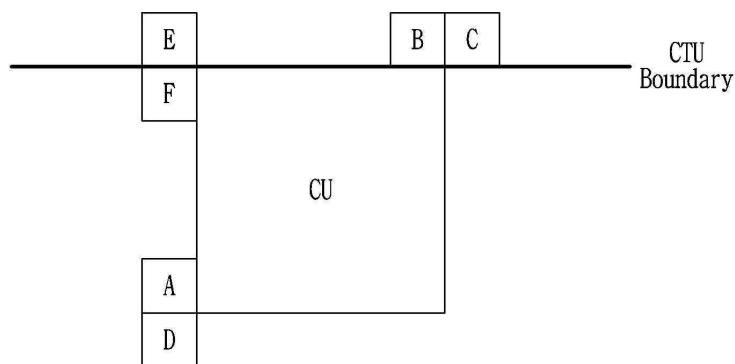
도면17



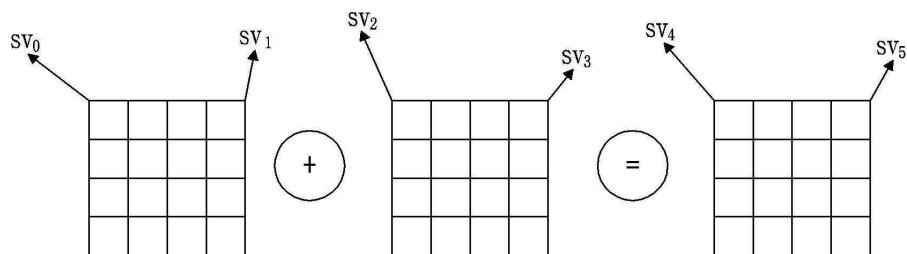
도면18



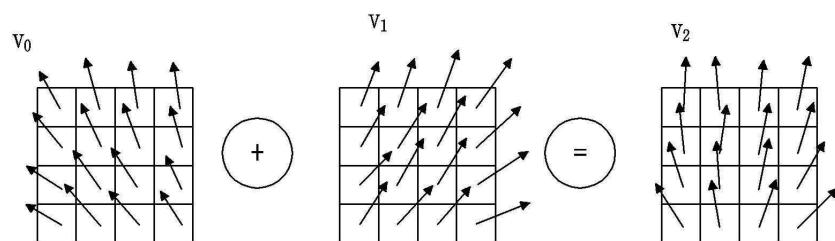
도면19



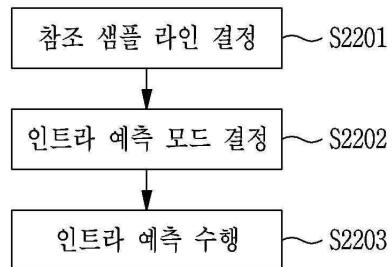
도면20



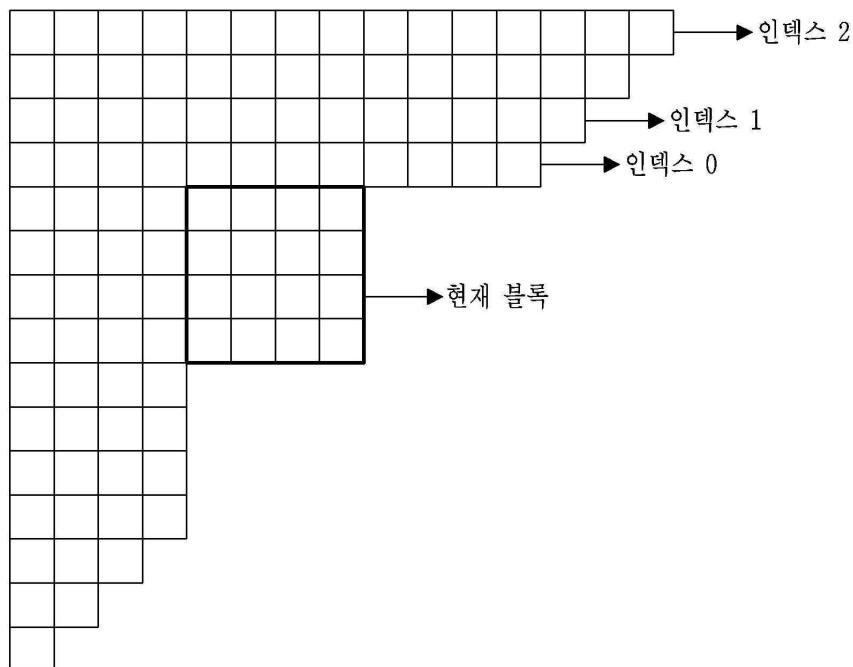
도면21



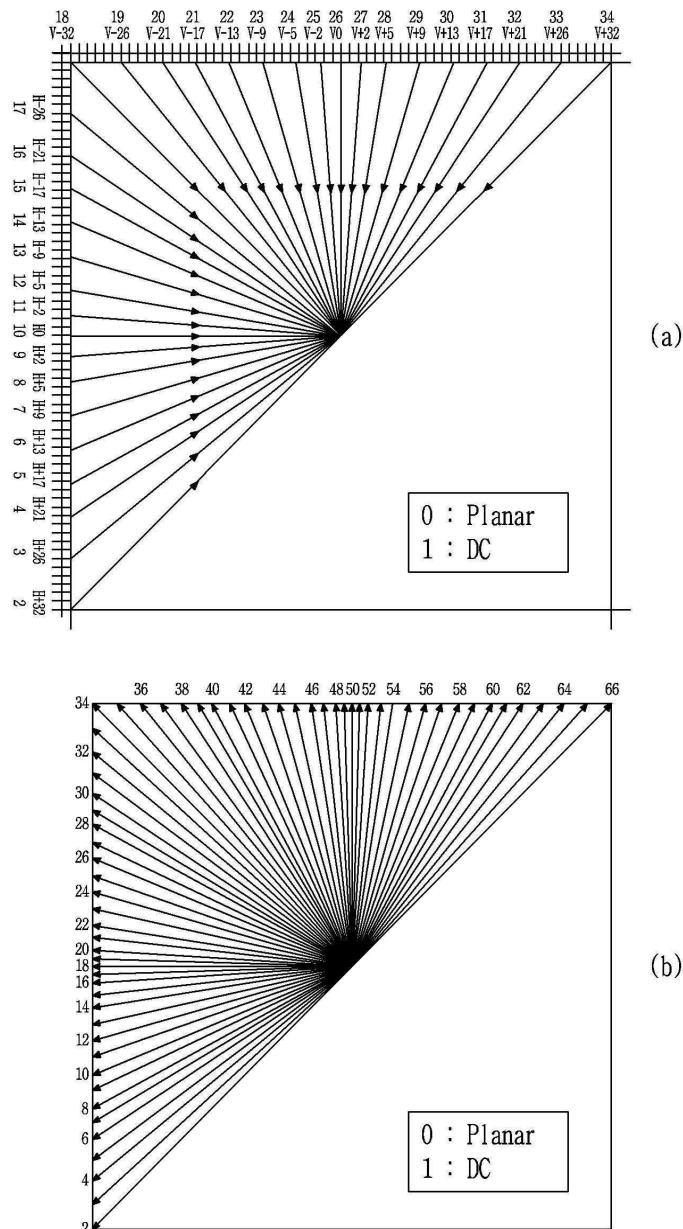
도면22



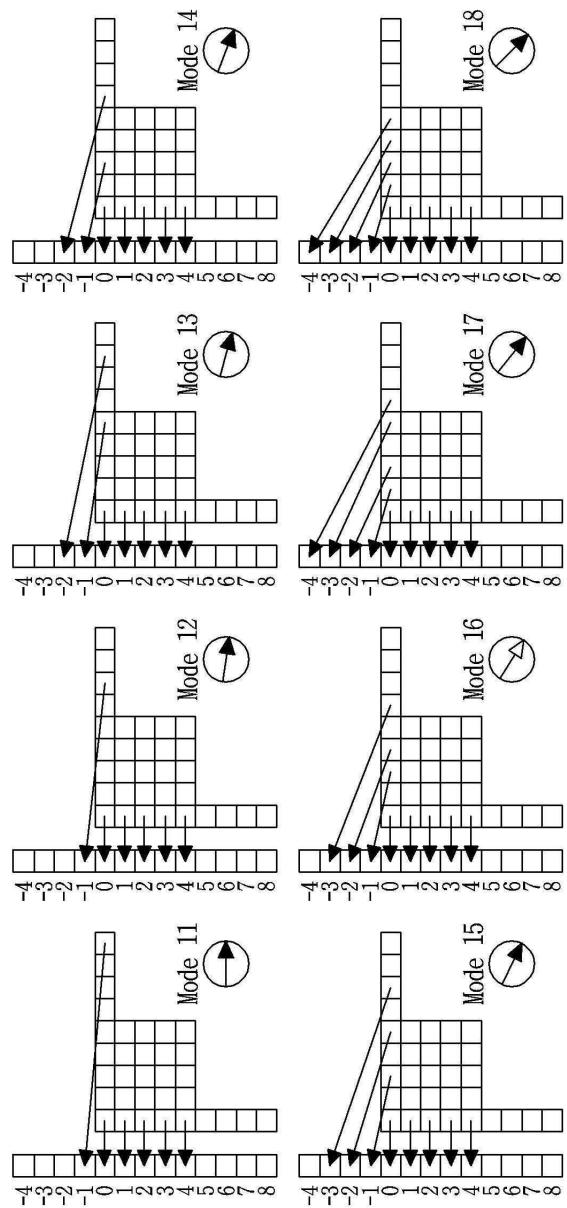
도면23



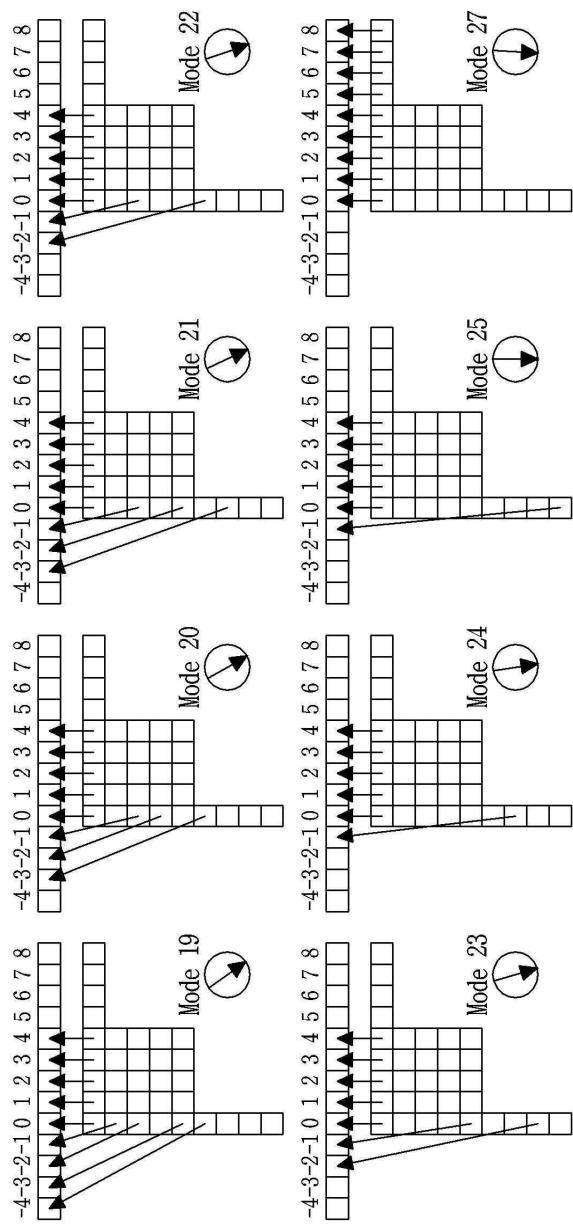
도면24



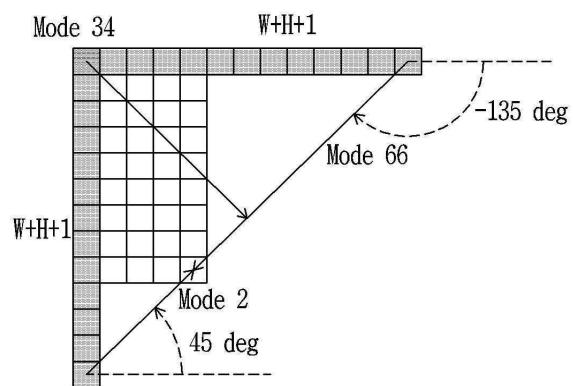
도면25



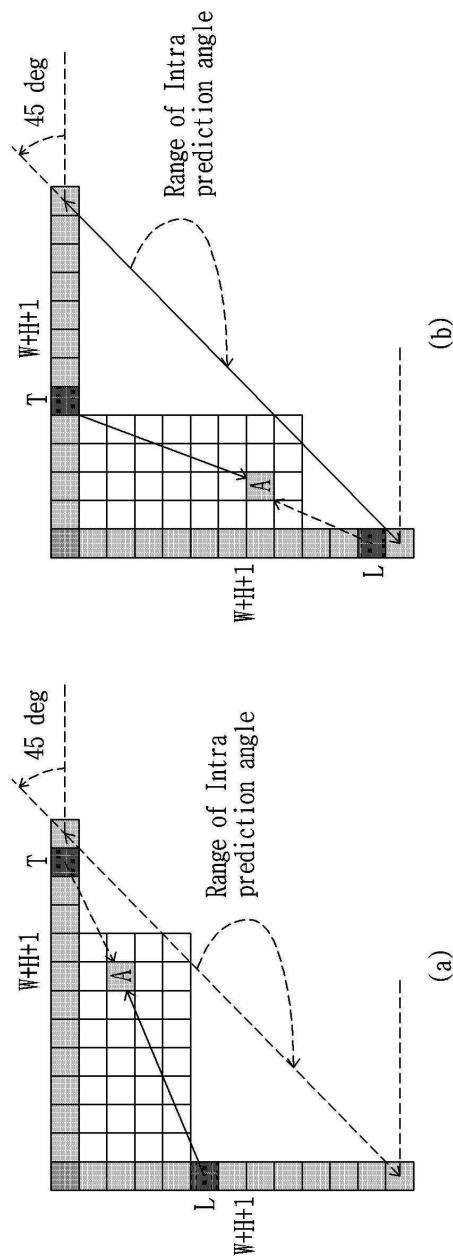
도면26



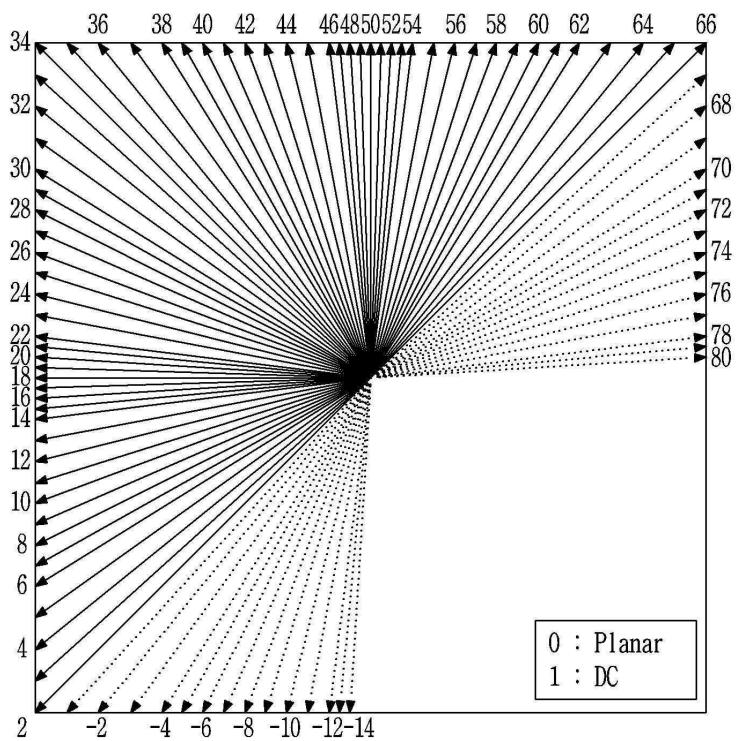
도면27



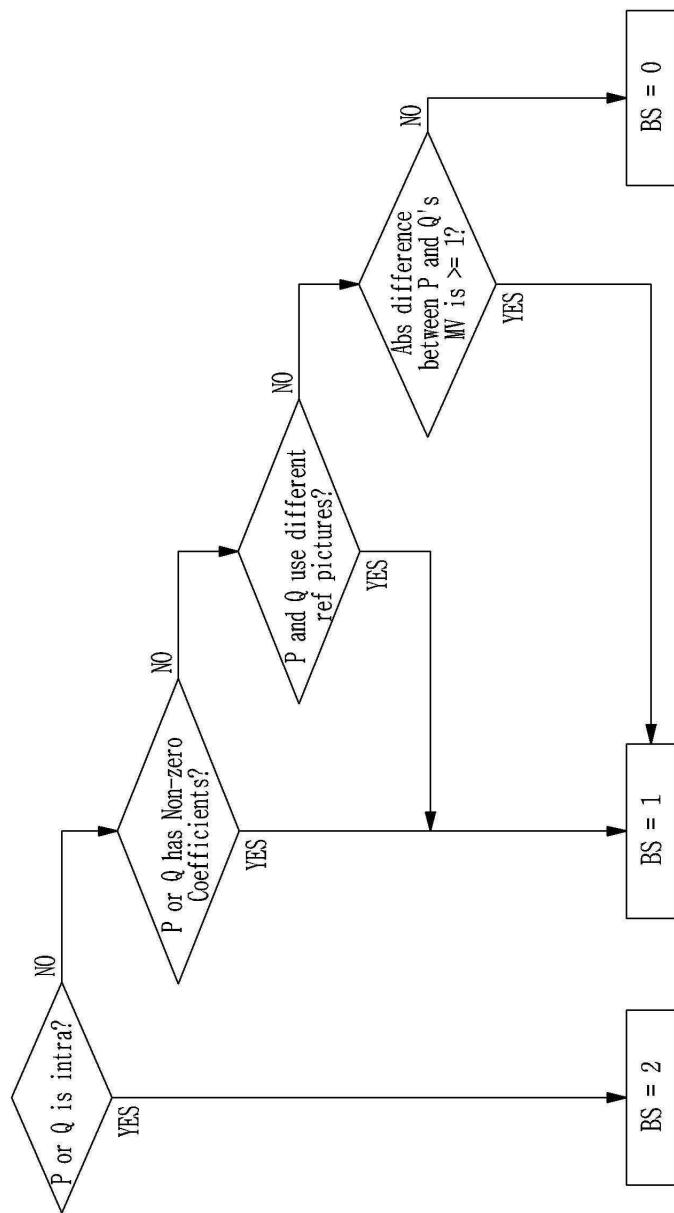
도면28



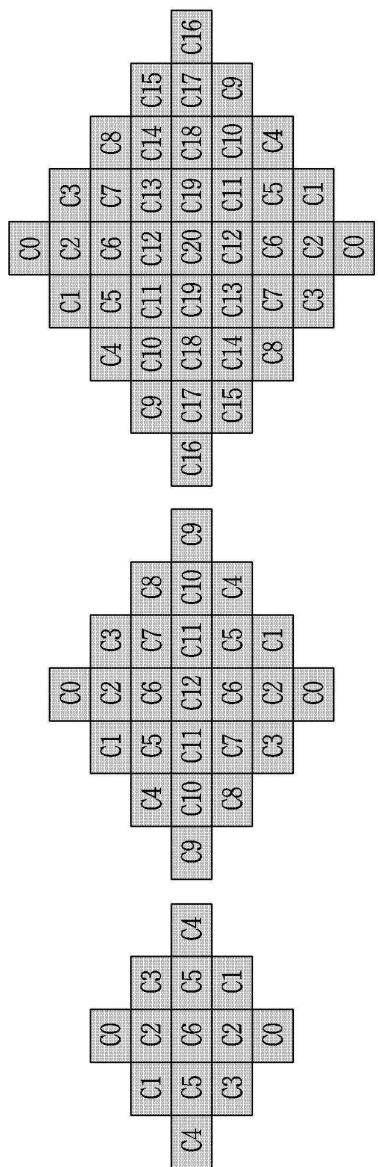
도면29



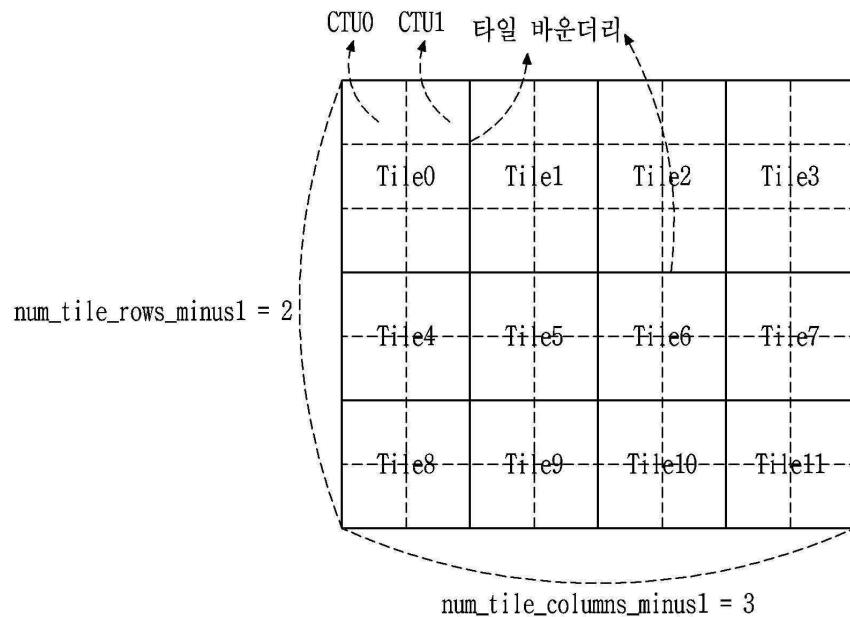
도면30



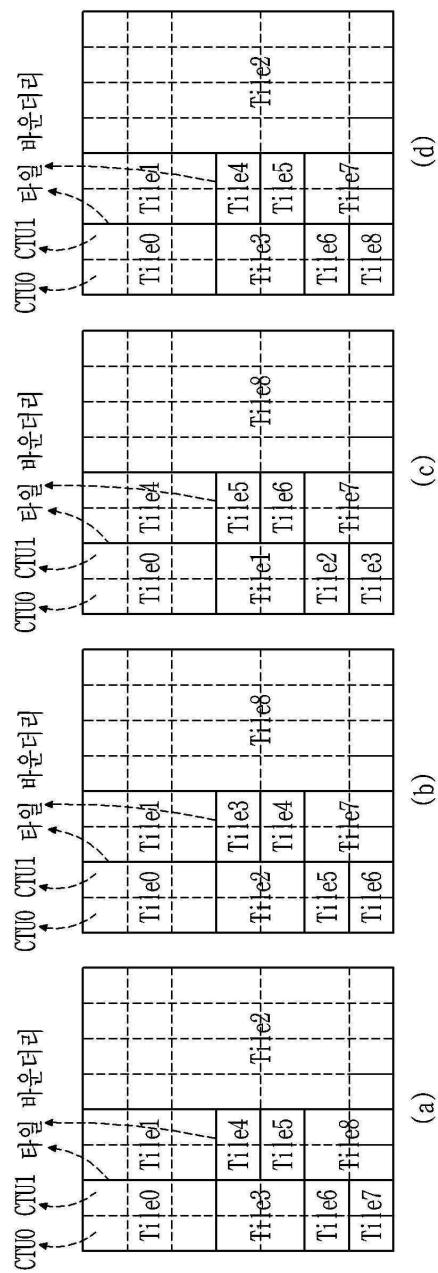
도면31



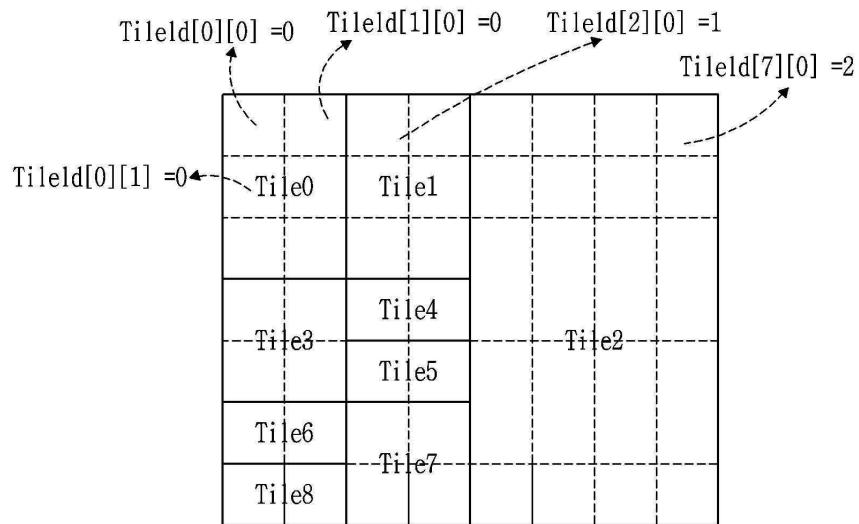
도면32



도면33



도면34



도면35

