



**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(11) 공개번호 10-2025-0065732  
(43) 공개일자 2025년05월13일

- (51) 국제특허분류(Int. Cl.)  
 H04N 19/119 (2014.01) H04N 19/105 (2014.01)  
 H04N 19/109 (2014.01) H04N 19/132 (2014.01)  
 H04N 19/137 (2014.01) H04N 19/14 (2014.01)  
 H04N 19/176 (2014.01) H04N 19/44 (2014.01)  
 H04N 19/513 (2014.01) H04N 19/70 (2014.01)
- (52) CPC특허분류  
 H04N 19/119 (2015.01)  
 H04N 19/105 (2015.01)
- (21) 출원번호 10-2025-7014429(분할)
- (22) 출원일자(국제) 2020년11월12일  
 심사청구일자 없음
- (62) 원출원 특허 10-2021-7024025  
 원출원일자(국제) 2020년11월12일  
 심사청구일자 2021년07월28일
- (85) 번역문제출일자 2025년04월30일
- (86) 국제출원번호 PCT/US2020/060281
- (87) 국제공개번호 WO 2021/101791  
 국제공개일자 2021년05월27일
- (30) 우선권주장  
 62/938,894 2019년11월21일 미국(US)  
 17/091,253 2020년11월06일 미국(US)
- (71) 출원인  
 텐센트 아메리카 엘엘씨  
 미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747
- (72) 발명자  
 천, 렌-페이  
 미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내  
 리, 상  
 미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내  
 (뒷면에 계속)
- (74) 대리인  
 양영준, 김연송, 백만기

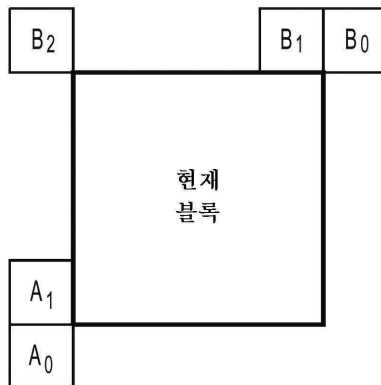
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 **비디오 코딩에서의 기하학적 파티셔닝 모드**

**(57) 요약**

비디오 디코딩 방법이, 기하학적 파티셔닝 모드에서 예측을 위해 파티션 에지를 따라 제1 부분 및 제2 부분으로 파티셔닝되는 현재 코딩 블록과 연관된 코딩된 비디오의 비트스트림으로부터 선택스 요소를 수신하는 단계, 및 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계를 포함한다. 현재 코딩 블록의 모션 필드에서 제1 모션 저장 유닛에 대한 저장된 모션 벡터 타입은 제1 모션 저장 유닛의 모션 인덱스에 기초하여 결정된다. 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스는 제1 모션 저장 유닛의 모션 인덱스로서 사용된다. 현재 코딩 블록의 모션 필드는 현재 코딩 블록의 모션 필드에서 제1 모션 저장 유닛에 대해 결정된 저장된 모션 벡터 타입에 기초하여 저장될 수 있다.

**대표도** - 도1a



(52) CPC특허분류

*HO4N 19/109* (2015.01)  
*HO4N 19/132* (2015.01)  
*HO4N 19/137* (2015.01)  
*HO4N 19/14* (2015.01)  
*HO4N 19/176* (2015.01)  
*HO4N 19/176* (2015.01)  
*HO4N 19/44* (2015.01)  
*HO4N 19/513* (2015.01)  
*HO4N 19/70* (2015.01)

(72) 발명자

리, 구이춘

미국 94306 캘리포니아주 팔로 알토 파크 블러바드  
2747 텐센트 아메리카 엘엘씨 내

류, 산

미국 94306 캘리포니아주 팔로 알토 파크 블러바드  
2747 텐센트 아메리카 엘엘씨 내

## 명세서

### 청구범위

#### 청구항 1

비디오 디코딩 방법으로서,

기하학적 파티셔닝 모드에서의 예측을 위해 파티션 에지를 따라 제1 부분 및 제2 부분으로 파티셔닝되는 현재 코딩 블록과 연관된 코딩된 비디오의 비트스트림으로부터 신택스 요소를 수신하는 단계;

상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계 - 상기 가중치 인덱스는 상기 현재 코딩 블록의 상기 제1 부분 및 상기 제2 부분에 대응하는 2개의 예측 신호를 블렌딩하기 위한 블렌딩 가중치를 도출하기 위해 사용됨 -;

제1 모션 저장 유닛의 모션 인덱스에 기초하여 상기 현재 코딩 블록의 모션 필드에서 상기 제1 모션 저장 유닛에 대한 저장된 모션 벡터 타입을 결정하는 단계 - 상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스는 상기 제1 모션 저장 유닛의 모션 인덱스로서 사용됨 -; 및

상기 현재 코딩 블록의 모션 필드에서 상기 제1 모션 저장 유닛에 대해 결정된 상기 저장된 모션 벡터 타입에 기초하여 상기 현재 코딩 블록의 모션 필드를 저장하는 단계를 포함하는 방법.

#### 청구항 2

제1항에 있어서,

상기 가중치 인덱스가 상기 제1 모션 저장 유닛의 모션 인덱스로서 사용되는 상기 현재 코딩 블록의 샘플 위치는 상기 코딩 블록의 모션 필드에서의 상기 제1 모션 저장 유닛 내의 샘플 위치인 방법.

#### 청구항 3

제1항에 있어서,

상기 가중치 인덱스가 상기 제1 모션 저장 유닛의 모션 인덱스로서 사용되는 상기 현재 코딩 블록의 샘플 위치는 상기 코딩 블록의 모션 필드에서의 상기 제1 모션 저장 유닛의 중심 위치에 이웃하는 샘플 위치인 방법.

#### 청구항 4

제1항에 있어서,

상기 제1 모션 저장 유닛은 4x4 샘플의 크기를 가지며, 상기 가중치 인덱스가 상기 제1 모션 저장 유닛의 모션 인덱스로서 사용되는 상기 현재 코딩 블록의 샘플 위치는 상기 코딩 블록의 모션 필드에서의 상기 제1 모션 저장 유닛 내의 좌표 (2, 2)의 샘플 위치인 방법.

#### 청구항 5

제1항에 있어서,

상기 제1 모션 저장 유닛은 4x4 샘플의 크기를 가지며 상기 현재 코딩 블록의 모션 필드에서  $(x_{sb}, y_{sb})$ 의 위치에 있고, 상기 현재 코딩 블록의 샘플 위치는 상기 현재 코딩 블록에서의  $((x_{sb} \ll 2) + 2, (y_{sb} \ll 2) + 2)$ 의 좌표를 가지며,  $((x_{sb} \ll 2) + 2, (y_{sb} \ll 2) + 2)$ 의 좌표에서의 가중치 인덱스는 상기 제1 모션 저장 유닛의 모션 인덱스로서 사용되는 방법.

#### 청구항 6

제1항에 있어서,

상기 2개의 예측 신호를 블렌딩하기 위해 상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 상기 블렌딩

가중치로 변환하기 위한 계산을 수행하는 단계 - 상기 변환은 룩업 테이블을 사용하지 않고 상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스의 선형 함수에 기초함 - 를 추가로 포함하는 방법.

**청구항 7**

제1항에 있어서,

상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계는:

상기 현재 코딩 블록의 코너로부터 상기 파티션 에지를 시프트하기 위해 사용되는 값  $\rho_{margin}$ 에 기초하여 상기 현재 코딩 블록의 샘플 위치에서 상기 가중치 인덱스를 결정하는 단계 - 상기 값  $\rho_{margin}$ 은 상기 현재 코딩 블록의 폭, 상기 현재 코딩 블록의 높이 및 상기 파티션 에지의 각도 중 적어도 하나에 따라 변함 - 를 포함하는 방법.

**청구항 8**

제1항에 있어서,

상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계는:

$$wIdx(x, y) = \left( x - \left( (w \times (4n + 3i)) \gg (3 + \log_2 n) \right) \right) \times \cos(\varphi) - \left( y - \left( (h \times (4n + 3i)) \gg (3 + \log_2 n) \right) \right) \times \cos\left(\varphi + \frac{\pi}{2}\right)$$
에 기초하여 상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계를 포함하고,

wIdx는 상기 가중치 인덱스를 나타내고, x 및 y는 상기 현재 코딩 블록에서의 샘플 위치의 좌표를 나타내고, w 및 h는 상기 현재 코딩 블록의 폭 및 높이를 나타내고,  $\varphi$ 는 상기 파티션 에지의 각도를 나타내고, n은 거리 양자화 스텝들의 수를 나타내고, i는 거리 양자화 스텝 인덱스를 나타내는 방법.

**청구항 9**

제1항에 있어서,

상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계는:

$$wIdx(x, y) = \left( \left( ((x \ll 1) + 1) \ll 3 \right) - \left( ((w \times (4n + 3i)) \ll 1) \gg \log_2 n \right) \right) \times \cos[\varphi] - \left( \left( ((y \ll 1) + 1) \ll 3 \right) - \left( ((h \times (4n + 3i)) \ll 1) \gg \log_2 n \right) \right) \times \cos\left[\varphi + \frac{\pi}{2}\right]$$
에 기초하여 상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계를 포함하고,

wIdx는 상기 가중치 인덱스를 나타내고, x 및 y는 상기 현재 코딩 블록에서의 샘플 위치의 좌표를 나타내고, w 및 h는 상기 현재 코딩 블록의 폭 및 높이를 나타내고,  $\varphi$ 는 상기 파티션 에지의 각도를 나타내고, n은 거리 양자화 스텝들의 수를 나타내고, i는 거리 양자화 스텝 인덱스를 나타내는 방법.

**청구항 10**

제1항에 있어서,

상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계는:

$$wIdx(x, y) = \begin{cases} \left( x - \frac{w \times (4n + 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n + 3i)}{8n} \right) \times \cos\left(\varphi + \frac{\pi}{2}\right), & 0 \leq \varphi < \frac{\pi}{2} \text{ 인 경우} \\ \left( x - \frac{w \times (4n - 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n + 3i)}{8n} \right) \times \cos\left(\varphi + \frac{\pi}{2}\right), & \frac{\pi}{2} \leq \varphi < \pi \text{ 인 경우} \\ \left( x - \frac{w \times (4n - 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n - 3i)}{8n} \right) \times \cos\left(\varphi + \frac{\pi}{2}\right), & \pi \leq \varphi < \frac{3\pi}{2} \text{ 인 경우} \\ \left( x - \frac{w \times (4n + 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n - 3i)}{8n} \right) \times \cos\left(\varphi + \frac{\pi}{2}\right), & \frac{3\pi}{2} \leq \varphi < 2\pi \text{ 인 경우} \end{cases}$$

에 기초하여 상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계를 포함하고, wIdx는 상기 가중치 인덱스를 나타내고, x 및 y는 상기 현재 코딩 블록에서의 샘플 위치의 좌표를 나타내고, w 및 h는 상기 현재 코딩 블록의 폭 및 높이를 나타내고,  $\varphi$ 는 상기 파티션 에지의 각도를 나타내고, n은 거리 양자화 스텝들의 수를 나타내고, i는 거리 양자화 스텝 인덱스를 나타내는 방법.

### 청구항 11

제1항에 있어서,

상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계는:

고정된 오프셋 값을 갖는 상기 현재 코딩 블록의 샘플 위치의 좌측 또는 상부 이웃 샘플의 가중치 인덱스에 기초하여 상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계를 포함하는 방법.

### 청구항 12

제1항에 있어서,

상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계는:

상기 현재 코딩 블록에서의 (0, 0) 좌표에서의 가중치 인덱스에 기초하여 상기 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하는 단계를 포함하는 방법.

### 청구항 13

제1항에 있어서,

$motionIdx(x_{sb}, y_{sb}) = \left( \left( \left( (x_{sb} \ll 3) + 1 \right) \ll 3 \right) - \left( \left( (w \times (4n + 3i)) \ll 1 \right) \gg \log_2 n \right) + 3 \right) \times \cos[\varphi] - \left( \left( \left( (y_{sb} \ll 3) + 1 \right) \ll 3 \right) - \left( \left( (h \times (4n + 3i)) \ll 1 \right) \gg \log_2 n \right) + 3 \right) \times \cos\left[\varphi + \frac{\pi}{2}\right]$ 에 기초하여 상기 현재 코딩 블록의 모션 필드에서 제2 모션 저장 유닛의 모션 인덱스를 결정하는 단계를 추가로 포함하고,

motionIdx는 상기 제2 모션 저장 유닛의 모션 인덱스를 나타내고,  $x_{sb}$  및  $y_{sb}$ 는 상기 현재 코딩 블록의 모션 필드에서의 상기 제2 모션 저장 유닛의 좌표를 나타내며, w 및 h는 상기 현재 코딩 블록의 폭 및 높이를 나타내고,  $\varphi$ 는 상기 파티션 에지의 각도를 나타내며, n은 거리 양자화 스텝들의 수를 나타내고, i는 거리 양자화 스텝 인덱스를 나타내는 방법.

### 청구항 14

비디오 인코딩 방법으로서,

현재 코딩 블록의 모션 필드에 기초하여 상기 현재 코딩 블록의 상기 모션 필드에서의 제1 모션 저장 유닛에 대

한 저장된 모션 벡터 타입을 결정하는 단계 - 상기 현재 코딩 블록은 기하학적 파티셔닝 모드에서의 예측을 위해 파티션 에지를 따라 제1 부분 및 제2 부분으로 파티셔닝됨 - ;

상기 현재 코딩 블록의 상기 모션 필드에서의 상기 제1 모션 저장 유닛에 대한 상기 저장된 모션 벡터 타입에 기초하여 상기 제1 모션 저장 유닛의 모션 인덱스를 결정하는 단계;

가중치 인덱스가 상기 제1 모션 저장 유닛의 모션 인덱스로서 사용되는, 상기 현재 코딩 블록의 샘플 위치를 결정하는 단계 - 상기 가중치 인덱스는 상기 현재 코딩 블록의 상기 제1 부분 및 상기 제2 부분에 대응하는 2개의 예측 신호를 블렌딩하기 위한 블렌딩 가중치를 도출하기 위해 사용됨 -; 및

상기 현재 코딩 블록과 연관된 코딩된 비디오의 비트스트림에서 선택 요소를 시그널링하는 단계를 포함하는 방법.

**청구항 15**

비디오 코딩 장치로서 회로를 포함하고, 상기 회로는 제1항 내지 제14항 중 어느 한 항에 따른 방법을 수행하도록 구성되는 비디오 코딩 장치.

**청구항 16**

비일시적 컴퓨터 판독가능 매체로서,

프로세서에 의해 실행될 때, 상기 프로세서로 하여금 제1항 내지 제14항 중 어느 한 항에 따른 방법을 수행하게 하는 명령어들을 저장하는 비일시적 컴퓨터 판독가능 매체.

**청구항 17**

비일시적 컴퓨터 판독가능 매체로서,

비트스트림을 저장하고,

상기 비트스트림은 제1항 내지 제13항 중 어느 한 항에 따른 방법에 의해 디코딩 가능하거나 상기 비트스트림은 제14항에 따른 방법을 통해 인코딩되는 비일시적 컴퓨터 판독가능 매체.

**발명의 설명**

**기술 분야**

[0001] [참조로 포함]

[0002] 본 출원은 2019년 11월 21일자로 출원된 미국 가출원 제62/938,894호, "SIMPLIFICATION FOR GEO/WEDGE INTER BLOCK"의 이익을 주장하는, 2020년 11월 6일자로 출원된 미국 특허 출원 제17/091,253호, "GEOMETRIC PARTITIONING MODE IN VIDEO CODING"의 이익을 주장한다. 이전 출원들의 개시내용들은 이로써 그 전체가 참조로 포함된다.

[0003] 본 개시내용은 일반적으로 비디오 코딩에 관련된 실시예들을 설명한다.

**배경 기술**

[0004] 본 명세서에 제공된 배경기술 설명은 본 개시내용의 맥락을 일반적으로 제시하기 위한 것이다. 현재 거명된 발명자들의 연구 - 그 연구가 이 배경기술 섹션에서 설명되는 한 - 뿐만 아니라 출원 시에 종래 기술로서의 자격이 없을 수 있는 설명의 양태들은 명시적으로도 암시적으로도 본 개시내용에 대한 종래 기술로서 인정되지 않는다.

[0005] 비디오 코딩 및 디코딩은 모션 보상(motion compensation)을 갖는 인터-픽처 예측(inter-picture prediction)을 사용하여 수행될 수 있다. 압축되지 않은 디지털 비디오는 일련의 픽처들을 포함할 수 있고, 각각의 픽처는, 예를 들어, 1920x1080 루미넌스 샘플들 및 연관된 크로미넌스 샘플들의 공간 차원(spatial dimension)을 갖는다. 일련의 픽처들은, 예를 들어, 초당 60개의 픽처 또는 60Hz의 고정 또는 가변 픽처 레이트(비공식적으로 프레임 레이트라고도 알려짐)를 가질 수 있다. 압축되지 않은 비디오는 상당한 비트레이트 요건들을 갖는다. 예를 들어, 샘플당 8 비트에서의 1080p60 4:2:0 비디오(60Hz 프레임 레이트에서의 1920x1080 루미넌스 샘플

플 해상도)는 1.5 Gbit/s에 근접한 대역폭을 요구한다. 1 시간 분량의 이러한 비디오는 600 기가바이트를 초과하는 저장 공간을 필요로 한다.

- [0006] 비디오 코딩 및 디코딩의 하나의 목적은, 압축을 통한, 입력 비디오 신호에서의 중복성(redundancy)의 감소일 수 있다. 압축은 전술한 대역폭 또는 저장 공간 요건들을, 일부 경우들에서, 2 자릿수 이상 감소시키는 데 도움이 될 수 있다. 무손실 및 손실 압축 둘 다뿐만 아니라 이들의 조합이 이용될 수 있다. 무손실 압축은 압축된 원래 신호로부터 원래 신호의 정확한 사본이 재구성될 수 있는 기법들을 지칭한다. 손실 압축을 이용할 때, 재구성된 신호는 원래 신호와 동일하지 않을 수 있지만, 원래 신호와 재구성된 신호 사이의 왜곡은 재구성된 신호를 의도된 응용에 유용하게 만들 정도로 충분히 작다. 비디오의 경우, 손실 압축이 널리 이용된다. 용인되는 왜곡의 양은 응용에 의존한다. 예를 들어, 특정 소비자 스트리밍 애플리케이션들의 사용자들은 텔레비전 배포 애플리케이션들의 사용자들보다 더 높은 왜곡을 용인할 수 있다. 달성가능한 압축비는 더 높은 허용가능한/용인가능한 왜곡이 더 높은 압축비를 산출할 수 있다는 것을 반영할 수 있다.
- [0007] 비디오 인코더 및 디코더는, 예를 들어, 모션 보상, 변환, 양자화, 및 엔트로피 코딩을 포함하여, 몇몇 광범위한 카테고리로부터의 기법들을 활용할 수 있다.
- [0008] 비디오 코덱 기술들은 인터 예측으로 알려진 기술들을 포함할 수 있다. 각각의 인터-예측된 코딩 유닛(CU)에 대해, 모션 파라미터들은 모션 벡터들, 참조 픽처 인덱스들 및 참조 픽처 리스트 사용 인덱스, 및 인터-예측된 샘플 생성을 위해 사용될 추가 정보를 포함한다. 모션 파라미터들은 명시적 또는 암시적 방식으로 시그널링될 수 있다. CU가 스킵 모드로 코딩될 때, CU는 하나의 예측 유닛(PU)과 연관되고, 유의 잔차 계수들을 갖지 않고, 코딩된 모션 벡터 델타 또는 참조 픽처 인덱스를 갖지 않는다. 병합 모드가 특정되는데, 그에 의해 공간 및 시간 후보들, 및 예를 들어, VVC에서 도입된 추가적인 스케줄들을 포함하는, 현재 CU에 대한 모션 파라미터들이 이웃 CU들로부터 획득된다. 병합 모드는 스킵 모드에 대해서뿐만 아니라 임의의 인터 예측된 CU에 적용될 수 있다. 병합 모드에 대한 대안은 모션 파라미터들의 명시적 송신이다. 모션 벡터, 각각의 참조 픽처 리스트 및 참조 픽처 리스트 사용 플래그에 대한 대응하는 참조 픽처 인덱스 및 다른 필요한 정보는 각각의 CU마다 명시적으로 시그널링된다.
- [0009] 일부 인터 예측 코딩 도구들은 확장된 병합 예측, MMVD(merge mode with motion vector difference), 대칭 MVD(motion vector difference) 시그널링을 가진 AMVP(advanced motion vector prediction mode), 아핀 모션 보상된 예측, SbTMVP(subblock-based temporal motion vector prediction), AMVR(adaptive motion vector resolution), 모션 필드 저장(1/16번째 루마 샘플 MV 저장 및 8x8 모션 필드 압축), BWA(bi-prediction with weighted averaging), BDOF(bi-directional optical flow), DMVR(decoder side motion vector refinement), TPM(triangular partitioning mode), 및 CIIP(combined inter and intra prediction)를 포함한다.
- [0010] 일부 경우들에서, 확장된 병합 예측이 사용된다. 병합 후보 리스트는 다음의 5개 타입의 후보들을 순서대로 포함함으로써 구성된다: (1) 공간 이웃 CU들로부터의 공간 MVP; (2) 병치된 CU들로부터의 시간적 MVP; (3) FIFO 테이블로부터의 이력 기반 MVP; (4) 쌍별 평균 MVP; 및 (5) 제로 MV들.
- [0011] 병합 리스트의 크기는 슬라이스 헤더에서 시그널링되고 병합 리스트의 최대 허용 크기는 일부 경우들에서 예를 들어 6이다. 병합 모드에서의 각각의 CU 코드에 대해, 최상의 병합 후보의 인덱스는 절단된 단항 이진화(truncated unary binarization, TU)를 사용하여 인코딩된다. 병합 인덱스의 제1 빈은 컨텍스트로 코딩되고 바이패스 코딩은 다른 빈들에 대해 사용된다.
- [0012] 도 1a는 공간 병합 후보들의 예시적인 위치들을 도시한다. 일부 경우들에서, 도 1a에 묘사된 위치들에 위치하는 후보들 중에서 최대 4개의 병합 후보가 선택될 수 있다. 선택 순서는 B1, A1, B0, A0, 및 B2이다. 위치 B2에서의 후보는 위치들 A0, B0, B1, 또는 A1에서의 임의의 CU가 이용가능하지 않거나(예를 들어, 위치 A0에서의 CU가 또 다른 슬라이스 또는 타일에 속함) 인터 코딩되지 않을 때에만 고려된다. 위치 A1에서의 후보가 병합 후보 리스트에 추가된 후에, 나머지 후보들의 추가는 코딩 효율이 개선되도록 동일한 모션 정보를 갖는 후보들이 병합 후보 리스트로부터 배제되는 것을 보장하는 중복 검사를 받는다.
- [0013] 도 1b는 공간 병합 후보들의 중복 검사를 위해 고려되는 후보 쌍들을 도시한다. 계산 복잡성을 줄이기 위해, 중복 검사에서 모든 가능한 후보 쌍들이 고려되지는 않는다. 대신에, 도 1b에서 화살표로 연결된 쌍들만이 고려되고 한 후보는 중복 검사를 위해 사용된 대응하는 후보가 동일한 모션 정보를 갖지 않으면 병합 후보 리스트에만 추가된다.
- [0014] 도 1c는 시간적 병합 후보에 대한 모션 벡터 스케일링을 도시한다. 일부 경우들에서, 하나의 시간적 병합 후보

만이 병합 후보 리스트에 추가될 수 있다. 특히, 이 시간적 병합 후보의 도출에서, 스케일링된 모션 벡터는 병치된 참조 픽처에 속하는 병치된 CU에 기초하여 도출된다. 공동 위치한 CU의 도출에 대해 사용되는 참조 픽처 리스트는 슬라이스 헤더에서 명시적으로 시그널링된다. 시간적 병합 후보에 대한 스케일링된 모션 벡터는 도 1c에서 점선으로 예시된 바와 같이 획득된다. 스케일링된 모션 벡터는 POC(picture order count) 거리들  $tb$  및  $td$ 를 사용하여 공동 위치한 CU의 모션 벡터로부터 도출되고, 여기서  $tb$ 는 현재 픽처의 참조 픽처와 현재 픽처 사이의 POC 차이로서 정의되고  $td$ 는 공동 위치한 픽처의 참조 픽처와 공동 위치한 픽처 사이의 POC 차이로서 정의된다. 시간적 병합 후보의 참조 픽처 인덱스는 0과 동일하게 설정될 수 있다.

[0015] 도 1d는 시간적 병합 후보에 대한 예시적인 위치들을 도시한다. 시간적 병합 후보는 위치들 C0 및 C1에서의 CU들 사이에서 선택된다. 위치 C0에서의 CU가 이용가능하지 않거나, 인터 코딩되지 않거나, 또는 CTU들의 현재 행의 외부에 있으면, 위치 C1에서의 CU가 사용된다. 그렇지 않으면, 위치 C0에서의 CU는 시간적 병합 후보의 도출에 사용된다.

### 발명의 내용

#### 과제의 해결 수단

[0016] 본 개시내용의 양태들은 비디오 인코딩 및 디코딩을 위한 방법들 및 장치를 제공한다. 일부 예들에서, 비디오 디코딩을 위한 장치는 처리 회로를 포함한다. 회로는 기하학적 파티셔닝 모드에서 예측을 위해 파티션 에지를 따라 제1 부분 및 제2 부분으로 파티셔닝되는 현재 코딩 블록과 연관된 코딩된 비디오의 비트스트림으로부터 선택스 요소를 수신하고, 현재 코딩 블록의 샘플 위치에서 가중치 인덱스를 결정하도록 구성될 수 있다. 가중치 인덱스는 현재 코딩 블록의 제1 부분 및 제2 부분에 대응하는 2개의 예측 신호를 블렌딩하기 위한 블렌딩 가중치(blending weight)를 도출하기 위해 사용된다. 회로는 제1 모션 저장 유닛의 모션 인덱스에 기초하여 현재 코딩 블록의 모션 필드에서 제1 모션 저장 유닛에 대한 저장된 모션 벡터 타입을 결정하고 - 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스는 제1 모션 저장 유닛의 모션 인덱스로서 사용됨 -, 및 현재 코딩 블록의 모션 필드에서 제1 모션 저장 유닛에 대해 결정된 저장된 모션 벡터 타입에 기초하여 현재 코딩 블록의 모션 필드를 저장하도록 추가로 구성될 수 있다.

[0017] 일 실시예에서, 가중치 인덱스가 제1 모션 저장 유닛의 모션 인덱스로서 사용되는 현재 코딩 블록의 샘플 위치는 코딩 블록의 모션 필드에서 제1 모션 저장 유닛 내의 샘플 위치이다. 일 실시예에서, 가중치 인덱스가 제1 모션 저장 유닛의 모션 인덱스로서 사용되는 현재 코딩 블록의 샘플 위치는 코딩 블록의 모션 필드에서 제1 모션 저장 유닛의 중심 위치에 이웃하는 샘플 위치이다. 일 실시예에서, 제1 모션 저장 유닛은 4x4 샘플의 크기를 가지며, 가중치 인덱스가 제1 모션 저장 유닛의 모션 인덱스로서 사용되는 현재 코딩 블록의 샘플 위치는 코딩 블록의 모션 필드에서의 제1 모션 저장 유닛 내의 좌표(2, 2)의 샘플 위치이다.

[0018] 일 실시예에서, 제1 모션 저장 유닛은 4x4 샘플들의 크기를 갖고 현재 코딩 블록의 모션 필드에서의  $(x_{sb}, y_{sb})$ 의 위치에 있고, 현재 코딩 블록의 샘플 위치는 현재 코딩 블록에서의  $((x_{sb} \ll 2) + 2, (y_{sb} \ll 2) + 2)$ 의 좌표를 갖고,  $((x_{sb} \ll 2) + 2, (y_{sb} \ll 2) + 2)$ 의 좌표에서의 가중치 인덱스는 제1 모션 저장 유닛의 모션 인덱스로서 사용된다.

[0019] 일 실시예에서, 회로는 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 2개의 예측 신호를 블렌딩하기 위한 블렌딩 가중치로 변환하기 위한 계산을 수행하도록 추가로 구성될 수 있다. 변환은 룩업 테이블을 사용하지 않고 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스의 선형 함수에 기초할 수 있다.

[0020] 일 실시예에서, 회로는 현재 코딩 블록의 코너로부터 파티션 에지를 시프트하기 위해 사용되는 값  $\rho_{margin_e}$ 에 기초하여 현재 코딩 블록의 샘플 위치에서 가중치 인덱스를 결정하도록 추가로 구성될 수 있고, 값  $\rho_{margin_o}$  현재 코딩 블록의 폭 또는 높이 및 파티션 에지의 각도에 따라 변한다.

[0021] 일 실시예에서, 회로는 다음에 기초하여 현재 코딩 블록의 샘플 위치에서 가중치 인덱스를 결정하도록 추가로 구성될 수 있다.

[0022] [수학식 1]

$$wIdx(x, y) = \left( x - \left( (w \times (4n + 3i)) \gg (3 + \log_2 n) \right) \right) \times \cos(\varphi) -$$

$$\left( y - \left( (h \times (4n + 3i)) \gg (3 + \log_2 n) \right) \right) \times \cos\left(\varphi + \frac{\pi}{2}\right)$$

[0023]

[0024] 여기서 wIdx는 상기 가중치 인덱스를 나타내고, x 및 y는 상기 현재 코딩 블록에서의 샘플 위치의 좌표를 나타내고, w 및 h는 상기 현재 코딩 블록의 폭 및 높이를 나타내고,  $\varphi$ 는 상기 파티션 에지의 각도를 나타내고, n은 거리 양자화 스텝들의 수를 나타내고, i는 거리 양자화 스텝 인덱스를 나타낸다. 일 실시예에서, 회로는 다음에 기초하여 현재 코딩 블록의 샘플 위치에서 가중치 인덱스를 결정하도록 추가로 구성된다.

[0025] [수학식 2]

$$wIdx(x, y) = \left( \left( (x \ll 1) + 1 \right) \ll 3 \right) - \left( \left( (w \times (4n + 3i)) \ll 1 \right) \gg \log_2 n \right) \times \cos[\varphi] -$$

$$\left( \left( (y \ll 1) + 1 \right) \ll 3 \right) - \left( \left( (h \times (4n + 3i)) \ll 1 \right) \gg \log_2 n \right) \times \cos\left[\varphi + \frac{\pi}{2}\right]$$

[0026]

[0027]

[0028] 여기서 wIdx는 상기 가중치 인덱스를 나타내고, x 및 y는 상기 현재 코딩 블록에서의 샘플 위치의 좌표를 나타내고, w 및 h는 상기 현재 코딩 블록의 폭 및 높이를 나타내고,  $\varphi$ 는 상기 파티션 에지의 각도를 나타내고, n은 거리 양자화 스텝들의 수를 나타내고, i는 거리 양자화 스텝 인덱스를 나타낸다. 일 실시예에서, 회로는 다음에 기초하여 현재 코딩 블록의 샘플 위치에서 가중치 인덱스를 결정하도록 추가로 구성된다.

[0029] [수학식 3]

$$wIdx(x, y) =$$

$$\begin{cases} \left( x - \frac{w \times (4n + 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n + 3i)}{8n} \right) \times \cos\left(\varphi + \frac{\pi}{2}\right), & 0 \leq \varphi < \frac{\pi}{2} \text{인 경우} \\ \left( x - \frac{w \times (4n - 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n + 3i)}{8n} \right) \times \cos\left(\varphi + \frac{\pi}{2}\right), & \frac{\pi}{2} \leq \varphi < \pi \text{인 경우} \\ \left( x - \frac{w \times (4n - 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n - 3i)}{8n} \right) \times \cos\left(\varphi + \frac{\pi}{2}\right), & \pi \leq \varphi < \frac{3\pi}{2} \text{인 경우} \\ \left( x - \frac{w \times (4n + 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n - 3i)}{8n} \right) \times \cos\left(\varphi + \frac{\pi}{2}\right), & \frac{3\pi}{2} \leq \varphi < 2\pi \text{인 경우} \end{cases}$$

[0030]

[0031] 여기서 wIdx는 상기 가중치 인덱스를 나타내고, x 및 y는 상기 현재 코딩 블록에서의 샘플 위치의 좌표를 나타내고, w 및 h는 상기 현재 코딩 블록의 폭 및 높이를 나타내고,  $\varphi$ 는 상기 파티션 에지의 각도를 나타내고, n은 거리 양자화 스텝들의 수를 나타내고, i는 거리 양자화 스텝 인덱스를 나타낸다.

[0032] 일 실시예에서, 회로는 고정된 오프셋 값을 갖는 현재 코딩 블록의 샘플 위치의 좌측 또는 상부 이웃 샘플의 가중치 인덱스에 기초하여 현재 코딩 블록의 샘플 위치에서 가중치 인덱스를 결정하도록 추가로 구성된다. 일 실시예에서, 회로는 현재 코딩 블록에서의 (0, 0)의 좌표에서의 가중치 인덱스에 기초하여 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스를 결정하도록 추가로 구성된다.

[0033] 일 실시예에서, 회로는 다음에 기초하여 현재 코딩 블록의 모션 필드에서 제2 모션 저장 유닛의 모션 인덱스를 결정하도록 추가로 구성된다.

[0034] [수학식 4]

[0035]  $motionIdx(x_{sb}, y_{sb})$

$$= \left( \left( (x_{sb} \ll 3) + 1 \right) \ll 3 \right) - \left( \left( (w \times (4n + 3i)) \ll 1 \right) \gg \log_2 n \right) + 3 \times \cos[\varphi] -$$

$$\left( \left( (y_{sb} \ll 3) + 1 \right) \ll 3 \right) - \left( \left( (h \times (4n + 3i)) \ll 1 \right) \gg \log_2 n \right) + 3 \times \cos \left[ \varphi + \frac{\pi}{2} \right]$$

[0036]

[0037] 여기서 motionIdx는 제2 모션 저장 유닛의 모션 인덱스를 나타내고,  $x_{sb}$  및  $y_{sb}$ 는 현재 코딩 블록의 모션 필드에서 제2 모션 저장 유닛의 좌표를 나타내며, w 및 h는 현재 코딩 블록의 폭 및 높이를 나타내고,  $\varphi$ 는 파티션 에지의 각도를 나타내며, n은 거리 양자화 스텝들의 수를 나타내고, i는 거리 양자화 스텝 인덱스를 나타낸다.

[0038] 본 개시내용의 양태들은 또한, 비디오 디코딩을 위해 컴퓨터에 의해 실행될 때, 컴퓨터로 하여금 비디오 디코딩을 위한 방법들 중 임의의 하나 또는 조합을 수행하게 야기하는 명령어들을 저장하는 비일시적 컴퓨터 판독가능 매체를 제공한다.

**도면의 간단한 설명**

[0039] 개시된 주제의 추가의 특징들, 본질 및 다양한 이점들이 다음의 상세한 설명 및 첨부 도면들로부터 더 명백할 것이다.

- 도 1a는 공간 병합 후보들의 예시적인 위치들을 도시한다.
- 도 1b는 공간 병합 후보들의 중복 검사를 위해 고려되는 예시적인 후보 쌍들을 도시한다.
- 도 1c는 시간적 병합 후보에 대한 모션 벡터 스케일링의 예를 도시한다.
- 도 1d는 시간적 병합 후보에 대한 예시적인 위치들을 도시한다.
- 도 2는 일 실시예에 따른 통신 시스템의 단순화된 블록도의 개략도를 도시한다.
- 도 3은 일 실시예에 따른 통신 시스템의 단순화된 블록도의 개략 예시를 도시한다.
- 도 4는 일 실시예에 따른 디코더의 단순화된 블록도의 개략도를 도시한다.
- 도 5는 일 실시예에 따른 인코더의 단순화된 블록도의 개략 예시를 도시한다.
- 도 6은 또 다른 실시예에 따른 인코더의 블록도를 도시한다.
- 도 7은 또 다른 실시예에 따른 디코더의 블록도를 도시한다.
- 도 8a 및 도 8b는 일부 실시예들에 따른 2개의 예시적인 삼각형 파티션을 도시한다.
- 도 9는 일부 실시예들에 따른 삼각형 파티션 모드에 대한 단방향 예측 모션 벡터 선택을 도시한다.
- 도 10a 및 도 10b는 일부 실시예들에 따른 루마 및 크로마에 대한 예시적인 가중치 맵들을 도시한다.
- 도 11은 본 개시내용의 일부 실시예들에 따른 예시적인 기하학적 파티셔닝 모드를 도시한다.
- 도 12는 일 실시예에 따른 예시적인 프로세스를 약술하는 흐름도를 도시한다.
- 도 13은 일 실시예에 따른 컴퓨터 시스템의 개략 예시를 도시한다.

**발명을 실시하기 위한 구체적인 내용**

[0040] I. 비디오 인코더 및 디코더

[0041] 도 2는 본 개시내용의 일 실시예에 따른 통신 시스템(200)의 단순화된 블록도를 예시한다. 통신 시스템(200)은, 예를 들어, 네트워크(250)를 통해, 서로 통신할 수 있는 복수의 단말 디바이스를 포함한다. 예를 들어, 통신 시스템(200)은 네트워크(250)를 통해 상호접속되는 제1 쌍의 단말 디바이스들(210 및 220)을 포

함한다. 도 2의 예에서, 제1 쌍의 단말 디바이스들(210 및 220)은 데이터의 단방향 송신을 수행한다. 예를 들어, 단말 디바이스(210)는 네트워크(250)를 통해 다른 단말 디바이스(220)로의 송신을 위해 비디오 데이터(예를 들어, 단말 디바이스(210)에 의해 캡처되는 비디오 픽처들의 스트림)를 코딩할 수 있다. 인코딩된 비디오 데이터는 하나 이상의 코딩된 비디오 비트스트림의 형식으로 송신될 수 있다. 단말 디바이스(220)는 네트워크(250)로부터 코딩된 비디오 데이터를 수신하고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처들을 복구하고 복구된 비디오 데이터에 따라 비디오 픽처들을 디스플레이할 수 있다. 단방향 데이터 송신은 미디어 서빙 응용들(media serving applications) 등에서 흔한 것일 수 있다.

[0042] 또 다른 예에서, 통신 시스템(200)은, 예를 들어, 영상 회의 동안 발생할 수 있는 코딩된 비디오 데이터의 양방향 송신을 수행하는 제2 쌍의 단말 디바이스들(230 및 240)을 포함한다. 데이터의 양방향 송신을 위해, 일 예에서, 단말 디바이스들(230 및 240) 중의 각각의 단말 디바이스는 네트워크(250)를 통해 단말 디바이스들(230 및 240) 중의 다른 단말 디바이스로의 송신을 위해 비디오 데이터(예를 들어, 단말 디바이스에 의해 캡처되는 비디오 픽처들의 스트림)를 코딩할 수 있다. 단말 디바이스들(230 및 240) 중의 각각의 단말 디바이스는 또한 단말 디바이스들(230 및 240) 중의 다른 단말 디바이스에 의해 송신된 코딩된 비디오 데이터를 수신할 수 있고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처들을 복구할 수 있고, 복구된 비디오 데이터에 따라 액세스 가능한 디스플레이 디바이스에서 비디오 픽처들을 디스플레이할 수 있다.

[0043] 도 2의 예에서, 단말 디바이스들(210, 220, 230 및 240)은 서버들, 개인용 컴퓨터들 및 스마트 폰들로서 예시될 수 있지만, 본 개시내용의 원리들은 그렇게만 제한되지는 않는다. 본 개시내용의 실시예들은 랩톱 컴퓨터들, 태블릿 컴퓨터들, 미디어 플레이어들 및/또는 전용 영상 회의 장비를 이용한 응용을 찾는다. 네트워크(250)는 예를 들어, 와이어라인(유선) 및/또는 무선 통신 네트워크들을 포함하여, 단말 디바이스들(210, 220, 230 및 240) 사이에 코딩된 비디오 데이터를 전달하는 임의의 수의 네트워크들을 나타낸다. 통신 네트워크(250)는 회선 교환 및/또는 패킷 교환 채널들에서 데이터를 교환할 수 있다. 대표적인 네트워크들은 원거리통신(telecommunications) 네트워크들, 로컬 영역 네트워크들, 광역 네트워크들 및/또는 인터넷을 포함한다. 본 논의의 목적을 위해, 네트워크(250)의 아키텍처 및 토폴로지는 아래에서 본 명세서에서 설명되지 않는 한 본 개시내용의 동작에 중요하지 않을 수 있다.

[0044] 도 3은, 개시된 주제를 위한 응용에 대한 예로서, 스트리밍 환경에서의 비디오 인코더 및 비디오 디코더의 배치를 예시한다. 개시된 주제는, 예를 들어, 영상 회의, 디지털 TV, CD, DVD, 메모리 스틱을 비롯한 디지털 미디어에 압축된 비디오를 저장하는 것, 및 등등을 비롯한 다른 비디오 지원 응용들에 똑같이 적용가능할 수 있다.

[0045] 스트리밍 시스템은 예를 들어, 압축되지 않은 비디오 픽처들의 스트림(302)을 생성하는 비디오 소스(301), 예를 들어, 디지털 카메라를 포함할 수 있는 캡처 서브시스템(313)을 포함할 수 있다. 일 예에서, 비디오 픽처들의 스트림(302)은 디지털 카메라에 의해 촬영되는 샘플들을 포함한다. 인코딩된 비디오 데이터(304)(또는 코딩된 비디오 비트스트림)와 비교할 때 많은 데이터 용량을 강조하기 위해 굵은 라인으로 묘사된 비디오 픽처들의 스트림(302)은 비디오 소스(301)에 결합된 비디오 인코더(303)를 포함하는 전자 디바이스(320)에 의해 처리될 수 있다. 비디오 인코더(303)는 아래에서 더 상세히 설명되는 바와 같이 개시된 주제의 양태들을 가능하게 하거나 구현하기 위해 하드웨어, 소프트웨어, 또는 이들의 조합을 포함할 수 있다. 비디오 픽처들의 스트림(302)과 비교할 때 적은 데이터 용량을 강조하기 위해 얇은 라인으로서 묘사된 인코딩된 비디오 데이터(304)(또는 인코딩된 비디오 비트스트림(304))는 미래의 사용을 위해 스트리밍 서버(305) 상에 저장될 수 있다. 도 3에서의 클라이언트 서브시스템들(306 및 308)과 같은 하나 이상의 스트리밍 클라이언트 서브시스템은 스트리밍 서버(305)에 액세스하여 인코딩된 비디오 데이터(304)의 사본들(307 및 309)을 검색할 수 있다. 클라이언트 서브시스템(306)은, 예를 들어, 전자 디바이스(330) 내에 비디오 디코더(310)를 포함할 수 있다. 비디오 디코더(310)는 인코딩된 비디오 데이터의 착신 사본(307)을 디코딩하고 디스플레이(312)(예를 들어, 디스플레이 스크린) 또는 다른 렌더링 디바이스(묘사되지 않음) 상에 렌더링될 수 있는 비디오 픽처들의 발신 스트림(311)을 생성한다. 일부 스트리밍 시스템들에서, 인코딩된 비디오 데이터(304, 307, 및 309)(예를 들어, 비디오 비트스트림들)는 특정 비디오 코딩/압축 표준들에 따라 인코딩될 수 있다. 해당 표준들의 예들은 ITU-T 권고안(Recommendation) H.265를 포함한다. 일 예에서, 개발 중인 비디오 코딩 표준은 VVC(Versatile Video Coding)로서 비공식적으로 알려져 있다. 개시된 주제는 VVC의 맥락에서 사용될 수 있다.

[0046] 전자 디바이스들(320 및 330)은 다른 컴포넌트들(도시되지 않음)을 포함할 수 있다는 점에 유의한다. 예를 들어, 전자 디바이스(320)는 비디오 디코더(도시되지 않음)도 포함할 수 있고 전자 디바이스(330)는 비디오 인코

더(도시되지 않음)도 포함할 수 있다.

- [0047] 도 4는 본 개시내용의 일 실시예에 따른 비디오 디코더(410)의 블록도를 도시한다. 비디오 디코더(410)는 전자 디바이스(430)에 포함될 수 있다. 전자 디바이스(430)는 수신기(431)(예를 들어, 수신 회로)를 포함할 수 있다. 비디오 디코더(410)는 도 3의 예에서의 비디오 디코더(310) 대신에 사용될 수 있다.
- [0048] 수신기(431)는 비디오 디코더(410)에 의해 디코딩될 하나 이상의 코딩된 비디오 시퀀스를 수신할 수 있다; 동일한 또는 다른 실시예에서, 한 번에 하나의 코딩된 비디오 시퀀스 - 각각의 코딩된 비디오 시퀀스의 디코딩은 다른 코딩된 비디오 시퀀스들과 독립적이다. 코딩된 비디오 시퀀스는, 인코딩된 비디오 데이터를 저장하는 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는, 채널(401)로부터 수신될 수 있다. 수신기(431)는 인코딩된 비디오 데이터를 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트림들과 함께 수신할 수 있고, 이들은 그것들 각각의 사용 엔티티들(묘사되지 않음)에 포워딩될 수 있다. 수신기(431)는 코딩된 비디오 시퀀스를 다른 데이터로부터 분리할 수 있다. 네트워크 지터를 방지하기 위해, 수신기(431)와 엔트로피 디코더/파서(420)(이후 "파서(420)") 사이 내에 버퍼 메모리(415)가 결합될 수 있다. 특정 응용들에서, 버퍼 메모리(415)는 비디오 디코더(410)의 일부이다. 다른 것들에서, 그것은 비디오 디코더(410)(묘사되지 않음) 외부에 있을 수 있다. 또 다른 것들에서, 예를 들어, 네트워크 지터를 방지하기 위해, 비디오 디코더(410) 외부의 버퍼 메모리(묘사되지 않음), 그리고 추가로, 예를 들어, 재생 타이밍을 핸들링하기 위해, 비디오 디코더(410) 내부의 또 다른 버퍼 메모리(415)가 존재할 수 있다. 수신기(431)가 충분한 대역폭 및 제어가능성의 저장/포워드 디바이스로부터, 또는 등시 동기식 네트워크(isosynchronous network)로부터 데이터를 수신하고 있을 때, 버퍼 메모리(415)는 필요하지 않을 수 있거나, 작을 수 있다. 인터넷과 같은 베스트 에포트 패킷 네트워크들 상에서의 사용을 위해, 버퍼 메모리(415)는 요구될 수 있고, 비교적 클 수 있고, 유리하게는 적응적 크기일 수 있고, 비디오 디코더(410) 외부의 운영 체제 또는 유사한 요소들(묘사되지 않음)에서 적어도 부분적으로 구현될 수 있다.
- [0049] 비디오 디코더(410)는 코딩된 비디오 시퀀스로부터 심벌들(421)을 재구성하기 위해 파서(420)를 포함할 수 있다. 그런 심벌들의 카테고리들은 비디오 디코더(410)의 동작을 관리하기 위해 사용되는 정보, 및 잠재적으로, 도 4에 도시된 바와 같이, 전자 디바이스(430)의 일체 부분(integral part)은 아니지만 전자 디바이스(430)에 결합될 수 있는 렌더링 디바이스(412)(예를 들어, 디스플레이 스크린)와 같은 렌더링 디바이스를 제어하기 위한 정보를 포함한다. 렌더링 디바이스(들)에 대한 제어 정보는 SEI 메시지(Supplemental Enhancement Information) 또는 VUI(Video Usability Information) 파라미터 세트 프래그먼트들(묘사되지 않음)의 형식일 수 있다. 파서(420)는 수신되는 코딩된 비디오 시퀀스를 파싱/엔트로피 디코딩할 수 있다. 코딩된 비디오 시퀀스의 코딩은 비디오 코딩 기술 또는 표준에 따를 수 있고, 가변 길이 코딩, 허프만 코딩(Huffman coding), 맥락 민감성(context sensitivity)을 갖거나 갖지 않는 산술 코딩 등을 포함하는 다양한 원리들을 따를 수 있다. 파서(420)는, 코딩된 비디오 시퀀스로부터, 그룹에 대응하는 적어도 하나의 파라미터에 기초하여, 비디오 디코더에서의 픽셀들의 서브그룹들 중 적어도 하나에 대한 서브그룹 파라미터들의 세트를 추출할 수 있다. 서브그룹들은 픽처 그룹들(Groups of Pictures, GOP들), 픽처들, 타일들, 슬라이스들, 매크로블록들, 코딩 유닛들(Coding Units, CU들), 블록들, 변환 단위들(Transform Units, TU들), 예측 단위들(Prediction Units, PU들) 등을 포함할 수 있다. 파서(420)는 또한 코딩된 비디오 시퀀스로부터 변환 계수, 양자화기 파라미터 값, 모션 벡터들 등과 같은 정보를 추출할 수 있다.
- [0050] 파서(420)는 버퍼 메모리(415)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/파싱 동작을 수행하여, 심벌들(421)을 생성할 수 있다.
- [0051] 심벌들(421)의 재구성은 코딩된 비디오 픽처 또는 그것의 부분들의 타입(예컨대: 인터 및 인트라 픽처, 인터 및 인트라 블록), 및 다른 인자들에 의존하여 다중의 상이한 유닛을 수반할 수 있다. 어느 유닛들이 수반되는지, 그리고 어떻게 되는지는 파서(420)에 의해 코딩된 비디오 시퀀스로부터 파싱된 서브그룹 제어 정보에 의해 제어될 수 있다. 파서(420)와 아래의 다중의 유닛 사이의 그러한 서브그룹 제어 정보의 흐름은 명확성을 위해 묘사되어 있지 않다.
- [0052] 이미 언급된 기능 블록들 이외에, 비디오 디코더(410)는 아래에 설명되는 바와 같이 개념적으로 다수의 기능 유닛으로 세분될 수 있다. 상업적 제약 하에서 동작하는 실제 구현에서, 이들 유닛 중 다수는 서로 밀접하게 상호작용하고, 적어도 부분적으로 서로 통합될 수 있다. 그러나, 개시된 주제를 설명하기 위한 목적으로, 아래의 기능 유닛들로의 개념적 세분이 적절하다.
- [0053] 제1 유닛은 스케일러/역변환 유닛(451)이다. 스케일러/역변환 유닛(451)은, 파서(420)로부터의 심벌(들)(421)

로서, 어느 변환을 사용할지, 블록 크기, 양자화 인자, 양자화 스케일링 행렬들(quantization scaling matrices) 등을 포함하여, 제어 정보뿐만 아니라 양자화된 변환 계수를 수신한다. 스케일러/역변환 유닛(451)은 집계기(aggregator)(455)에 입력될 수 있는 샘플 값들을 포함하는 블록들을 출력할 수 있다.

[0054] 일부 경우들에서, 스케일러/역변환(451)의 출력 샘플들은 인트라 코딩된 블록에 관련될 수 있다; 즉: 이전에 재구성된 픽처들로부터의 예측 정보를 이용하는 것이 아니라, 현재 픽처의 이전에 재구성된 부분들로부터의 예측 정보를 이용할 수 있는 블록. 그러한 예측 정보는 인트라 픽처 예측 유닛(452)에 의해 제공될 수 있다. 일부 경우들에서, 인트라 픽처 예측 유닛(452)은 현재 픽처 버퍼(458)로부터 폐치된 주위의 이미 재구성된 정보를 이용하여, 재구성 중인 블록의 동일한 크기 및 형상의 블록을 생성한다. 현재 픽처 버퍼(458)는, 예를 들어, 부분적으로 재구성된 현재 픽처 및/또는 완전히 재구성된 현재 픽처를 버퍼링한다. 집계기(455)는, 일부 경우들에서, 샘플당 기준으로, 인트라 예측 유닛(452)이 생성한 예측 정보를 스케일러/역변환 유닛(451)에 의해 제공된 출력 샘플 정보에 더한다.

[0055] 다른 경우들에서, 스케일러/역변환 유닛(451)의 출력 샘플들은 인터 코딩되고, 잠재적으로 모션 보상된 블록에 관련될 수 있다. 이 경우에, 모션 보상 예측 유닛(453)은 참조 픽처 메모리(457)에 액세스하여 예측을 위해 사용되는 샘플을 폐치할 수 있다. 블록에 관련된 심벌들(421)에 따라 폐치된 샘플들을 모션 보상을 한 후에, 이들 샘플은 집계기(455)에 의해 스케일러/역변환 유닛(451)의 출력(이 경우 잔차 샘플들 또는 잔차 신호라고 불림)에 더해져서 출력 샘플 정보를 생성할 수 있다. 모션 보상 예측 유닛(453)이 그로부터 예측 샘플들을 폐치하는 참조 픽처 메모리(457) 내의 어드레스들은, 예를 들어, X, Y, 및 참조 픽처 컴포넌트들을 가질 수 있는 심벌들(421)의 형식으로 모션 보상 예측 유닛(453)에 이용가능한 모션 벡터들에 의해 제어될 수 있다. 모션 보상은 또한 서브샘플 정확한 모션 벡터들이 사용 중일 때 참조 픽처 메모리(457)로부터 폐치된 샘플 값들의 보간, 모션 벡터 예측 메커니즘 등을 포함할 수 있다.

[0056] 집계기(455)의 출력 샘플들에 대해 루프 필터 유닛(456)에서의 다양한 루프 필터링 기법들이 수행될 수 있다. 비디오 압축 기술들은, 파서(420)로부터의 심벌들(421)로서 루프 필터 유닛(456)에 이용가능하게 되고 코딩된 비디오 시퀀스(코딩된 비디오 비트스트림이라고도 지칭됨)에 포함된 파라미터들에 의해 제어되지만, 코딩된 픽처 또는 코딩된 비디오 시퀀스의 이전(디코딩 순서로) 부분들의 디코딩 동안 획득된 메타-정보에 응답할 뿐만 아니라, 이전에 재구성된 및 루프-필터링된 샘플 값들에 응답할 수도 있는 인-루프 필터 기술(in-loop filter technologies)들을 포함할 수 있다.

[0057] 루프 필터 유닛(456)의 출력은 렌더링 디바이스(412)에 출력될 뿐만 아니라 미래의 인터-픽처 예측에서 사용하기 위해 참조 픽처 메모리(457)에 저장될 수도 있는 샘플 스트림일 수 있다.

[0058] 특정 코딩된 픽처들은, 일단 완전히 재구성되면, 미래 예측을 위한 참조 픽처들로서 사용될 수 있다. 예를 들어, 일단 현재 픽처에 대응하는 코딩된 픽처가 완전히 재구성되고 코딩된 픽처가 참조 픽처로서 식별되면(예를 들어, 파서(420)에 의해), 현재 픽처 버퍼(458)는 참조 픽처 메모리(457)의 일부가 될 수 있고, 다음의 코딩된 픽처의 재구성을 개시하기 전에 새로운 현재 픽처 버퍼가 재할당될 수 있다.

[0059] 비디오 디코더(410)는 ITU-T Rec. H.265와 같은 표준에서의 미리 결정된 비디오 압축 기술에 따라 디코딩 동작들을 수행할 수 있다. 코딩된 비디오 시퀀스는 비디오 압축 기법 또는 표준의 선택스, 또는 비디오 압축 기법 또는 표준에서 문서화된 프로파일들 둘 다를 고수한다는 점에서, 코딩된 비디오 시퀀스는 사용 중인 비디오 압축 기법 또는 표준에 의해 특정된 선택스(syntax)를 준수할 수 있다. 구체적으로, 프로파일은 비디오 압축 기술 또는 표준에서 이용가능한 모든 도구들로부터 해당 프로파일 하에서 사용하기 위해 이용가능한 유일한 도구들로서 특정 도구들을 선택할 수 있다. 또한 준수를 위해 필요한 것은 코딩된 비디오 시퀀스의 복잡성이 비디오 압축 기술 또는 표준의 레벨에 의해 정의된 경계 내에 있는 것일 수 있다. 일부 경우들에서, 레벨들은 최대 픽처 크기, 최대 프레임 레이트, 최대 재구성 샘플 레이트(예를 들어, 초당 메가 샘플들로 측정됨), 최대 참조 픽처 크기 등을 제한한다. 레벨들에 의해 설정된 한계들은, 일부 경우들에서, HRD(Hypothetical Reference Decoder) 사양들 및 코딩된 비디오 시퀀스에서 시그널링된 HRD 버퍼 관리를 위한 메타데이터를 통해 추가로 제한될 수 있다.

[0060] 일 실시예에서, 수신기(431)는 인코딩된 비디오와 함께 추가적인(중복) 데이터를 수신할 수 있다. 이 추가적인 데이터는 코딩된 비디오 시퀀스(들)의 일부로서 포함될 수 있다. 이 추가적인 데이터는 데이터를 적절히 디코딩하고 및/또는 원래의 비디오 데이터를 더 정확하게 재구성하기 위해 비디오 디코더(410)에 의해 사용될 수 있다. 추가적인 데이터는 예를 들어, 시간적, 공간, 또는 신호 잡음 비(SNR) 향상 계층들, 중복 슬라이스들, 중복 픽처들, 순방향 오류 정정 코드들 등의 형식일 수 있다.

- [0061] 도 5는 본 개시내용의 일 실시예에 따른 비디오 인코더(503)의 블록도를 도시한다. 비디오 인코더(503)는 전자 디바이스(520)에 포함된다. 전자 디바이스(520)는 송신기(540)(예를 들어, 송신 회로)를 포함한다. 비디오 인코더(503)는 도 3의 예에서의 비디오 인코더(303) 대신에 사용될 수 있다.
- [0062] 비디오 인코더(503)는 비디오 인코더(503)에 의해 코딩될 비디오 이미지(들)를 캡처할 수 있는 비디오 소스(501)(도 5의 예에서는 전자 디바이스(520)의 일부가 아님)로부터 비디오 샘플들을 수신할 수 있다. 또 다른 예에서, 비디오 소스(501)는 전자 디바이스(520)의 일부이다.
- [0063] 비디오 소스(501)는, 임의의 적합한 비트 심도(예를 들어: 8 비트, 10 비트, 12 비트, ...), 임의의 컬러 공간(예를 들어, BT.601 Y CrCb, RGB, ...), 및 임의의 적합한 샘플링 구조(예를 들어, Y CrCb 4:2:0, Y CrCb 4:4:4)일 수 있는 디지털 비디오 샘플 스트림의 형식으로 비디오 인코더(503)에 의해 코딩될 소스 비디오 시퀀스를 제공할 수 있다. 미디어 서빙 시스템에서, 비디오 소스(501)는 이전에 준비된 비디오를 저장하는 저장 디바이스일 수 있다. 영상 회의 시스템에서, 비디오 소스(501)는 비디오 시퀀스로서 로컬 이미지 정보를 캡처하는 카메라일 수 있다. 비디오 데이터는 순차적으로 볼 때 모션을 부여하는 복수의 개별 픽처로서 제공될 수 있다. 픽처들 자체는 픽셀들의 공간 어레이로서 조직될 수 있고, 여기서 각각의 픽셀은 사용 중인 샘플링 구조, 컬러 공간 등에 의존하여 하나 이상의 샘플을 포함할 수 있다. 본 기술분야의 통상의 기술자는 픽셀들과 샘플들 사이의 관계를 용이하게 이해할 수 있다. 이하의 설명은 샘플들에 초점을 맞춘다.
- [0064] 일 실시예에 따르면, 비디오 인코더(503)는 소스 비디오 시퀀스의 픽처들을 실시간으로 또는 응용에 의해 요구되는 임의의 다른 시간 제약들 하에서 코딩된 비디오 시퀀스(543)로 코딩 및 압축할 수 있다. 적절한 코딩 속도를 시행하는 것이 컨트롤러(550)의 하나의 기능이다. 일부 실시예들에서, 컨트롤러(550)는 아래 설명되는 바와 같이 다른 기능 유닛들을 제어하고 다른 기능 유닛들에 기능적으로 결합된다. 결합은 명료성을 위해 묘사되지 않는다. 컨트롤러(550)에 의해 설정된 파라미터들은 레이트 제어 관련 파라미터들(픽처 스킵, 양자화기, 레이트-왜곡 최적화 기법들의 램다 값들,...), 픽처 크기, 픽처 그룹(GOP) 레이아웃, 최대 모션 벡터 허용 참조 영역 등을 포함할 수 있다. 컨트롤러(550)는 특정 시스템 설계에 대해 최적화된 비디오 인코더(503)에 관련된 다른 적합한 기능들을 갖도록 구성될 수 있다.
- [0065] 일부 실시예들에서, 비디오 인코더(503)는 코딩 루프에서 동작하도록 구성된다. 과도하게 단순화된 설명으로서, 일 예에서, 코딩 루프는 소스 코더(530)(예를 들어, 코딩될 입력 픽처, 및 참조 픽처(들)에 기초하여 심벌 스트림과 같은 심벌들을 생성하는 것을 담당함), 및 비디오 인코더(503)에 임베드된 (로컬) 디코더(533)를 포함할 수 있다. 디코더(533)는 (원격) 디코더가 또한 생성하는 것과 유사한 방식으로 샘플 데이터를 생성하기 위해 심벌들을 재구성한다(심벌들과 코딩된 비디오 비트스트림 사이의 임의의 압축이 개시된 주제에서 고려되는 비디오 압축 기술들에서 무손실이기 때문에). 재구성된 샘플 스트림(샘플 데이터)은 참조 픽처 메모리(534)에 입력된다. 심벌 스트림의 디코딩이 디코더 위치(로컬 또는 원격)와는 독립적으로 비트-정확한(bit-exact) 결과들을 야기하기 때문에, 참조 픽처 메모리(534) 내의 콘텐츠도 또한 로컬 인코더와 원격 인코더 사이에서 비트-정확한 상태이다. 다시 말해서, 인코더의 예측 부분은 디코딩 동안 예측을 사용할 때 디코더가 "보는(see)" 것과 정확히 동일한 샘플 값들을 참조 픽처 샘플들로서 "본다(sees)". 참조 픽처 동기성(reference picture synchronicity)의 이러한 기본적인 원리(그리고, 예를 들어, 채널 오류들 때문에, 동기성이 유지될 수 없는 경우, 결과적인 드리프트)는 일부 관련 기술에서도 사용된다.
- [0066] "로컬" 디코더(533)의 동작은 도 4와 관련하여 위에서 이미 상세히 설명된 비디오 디코더(410)와 같은 "원격" 디코더와 동일할 수 있다. 그러나, 또한 도 4를 잠시 참조하면, 심벌들이 이용가능하고 엔트로피 코더(545) 및 파서(420)에 의한 코딩된 비디오 시퀀스로서의 심벌들의 인코딩/디코딩이 무손실일 수 있기 때문에, 버퍼 메모리(415) 및 파서(420)를 포함하는, 비디오 디코더(410)의 엔트로피 디코딩 부분들은 로컬 디코더(533)에서 완전히 구현되지 않을 수 있다.
- [0067] 이 시점에서 이루어질 수 있는 관찰은, 디코더에 존재하는 파싱/엔트로피 디코딩을 제외한 임의의 디코더 기술이 또한 필연적으로, 대응하는 인코더에서, 실질적으로 동일한 기능 형식으로 존재할 필요가 있다는 점이다. 이러한 이유로, 개시된 주제는 디코더 동작에 초점을 맞춘다. 인코더 기술들은 포괄적으로 설명된 디코더 기술들의 역(inverse)이기 때문에 그것들에 대한 설명은 축약될 수 있다. 특정 영역들에서만 더 상세한 설명이 요구되고 아래에 제공된다.
- [0068] 동작 동안, 일부 예들에서, 소스 코더(530)는, "참조 픽처들"로서 지정된 비디오 시퀀스로부터의 하나 이상의 이전에 코딩된 픽처를 참조하여 예측적으로 입력 픽처를 코딩하는, 모션 보상된 예측 코딩을 수행할 수 있다. 이러한 방식으로, 코딩 엔진(532)은 입력 픽처의 픽셀 블록들과 입력 픽처에 대한 예측 참조(들)로서 선택될 수

있는 참조 픽처(들)의 픽셀 블록들 사이의 차이들을 코딩한다.

- [0069] 로컬 비디오 디코더(533)는, 소스 코더(530)에 의해 생성된 심벌들에 기초하여, 참조 픽처들로서 지정될 수 있는 픽처들의 코딩된 비디오 데이터를 디코딩할 수 있다. 코딩 엔진(532)의 동작들은 유리하게는 손실 프로세스들일 수 있다. 코딩된 비디오 데이터가 비디오 디코더(도 5에 도시되지 않음)에서 디코딩될 수 있는 경우, 재구성된 비디오 시퀀스는 전형적으로 일부 오류들을 갖는 소스 비디오 시퀀스의 복제본(replica)일 수 있다. 로컬 비디오 디코더(533)는 참조 픽처들에 대해 비디오 디코더에 의해 수행될 수 있는 디코딩 프로세스들을 복제하고 재구성된 참조 픽처들이 참조 픽처 캐시(534)에 저장되게 야기할 수 있다. 이러한 방식으로, 비디오 인코더(503)는 (송신 오류들이 없이) 원단(far-end) 비디오 디코더에 의해 획득될 재구성된 참조 픽처로서 공통 콘텐츠를 갖는 재구성된 참조 픽처들의 사본들을 저장할 수 있다.
- [0070] 예측기(535)는 코딩 엔진(532)에 대한 예측 검색들을 수행할 수 있다. 즉, 코딩될 새로운 픽처에 대해, 예측기(535)는 새로운 픽처들에 대한 적절한 예측 참조로서 역할할 수 있는 참조 픽처 모션 벡터들, 블록 형상들 등과 같은 특정 메타데이터 또는 샘플 데이터(후보 참조 픽셀 블록들로서)에 대해 참조 픽처 메모리(534)를 검색할 수 있다. 예측기(535)는 적절한 예측 참조들을 찾기 위해 샘플 블록 바이 픽셀 블록(sample block-by-pixel block) 기준으로 동작할 수 있다. 일부 경우들에서, 예측기(535)에 의해 획득된 검색 결과들에 의해 결정된 바와 같이, 입력 픽처는 참조 픽처 메모리(534)에 저장된 다중의 참조 픽처로부터 인출된 예측 참조들을 가질 수 있다.
- [0071] 컨트롤러(550)는, 예를 들어, 비디오 데이터를 인코딩하기 위해 사용되는 파라미터들 및 서브그룹 파라미터들의 설정을 포함하여, 소스 코더(530)의 코딩 동작을 관리할 수 있다.
- [0072] 진술한 모든 기능 유닛들의 출력은 엔트로피 코더(545)에서 엔트로피 코딩을 겪을 수 있다. 엔트로피 코더(545)는 다양한 기능 유닛들에 의해 생성된 심벌들을, 허프만 코딩, 가변 길이 코딩, 산술 코딩 등과 같은 기술들에 따라 심벌들을 무손실 압축함으로써, 코딩된 비디오 시퀀스로 변환한다.
- [0073] 송신기(540)는, 인코딩된 비디오 데이터를 저장할 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는, 통신 채널(560)을 통한 송신을 준비하기 위해 엔트로피 코더(545)에 의해 생성된 코딩된 비디오 시퀀스(들)를 버퍼링할 수 있다. 송신기(540)는 비디오 코더(503)로부터의 코딩된 비디오 데이터를 송신될 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트림(소스들이 도시되지 않음)과 병합할 수 있다.
- [0074] 컨트롤러(550)는 비디오 인코더(503)의 동작을 관리할 수 있다. 코딩 동안, 컨트롤러(550)는, 각자의 픽처에 적용될 수 있는 코딩 기법들에 영향을 미칠 수 있는, 특정 코딩된 픽처 타입을 각각의 코딩된 픽처에 할당할 수 있다. 예를 들어, 픽처들은 종종 다음 픽처 타입들 중 하나로서 할당될 수 있다:
- [0075] 인트라 픽처(Intra Picture)(I 픽처)는 예측의 소스로서 시퀀스에서의 임의의 다른 픽처를 사용하지 않고 코딩되고 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱들은, 예를 들어, "IDR"(Independent Decoder Refresh) 픽처들을 포함하는, 상이한 타입의 인트라 픽처들을 허용한다. 본 기술분야의 통상의 기술자는 I 픽처들의 해당 변형들 및 그것들 각자의 응용들 및 특징들을 인식한다.
- [0076] 예측 픽처(Predictive picture)(P 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 하나의 모션 벡터 및 참조 인덱스를 사용하여 인트라 예측(intra prediction) 또는 인터 예측(inter prediction)을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다.
- [0077] 양방향 예측 픽처(Bi-directionally Predictive Picture)(B 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 2개의 모션 벡터 및 참조 인덱스를 사용하여 인트라 예측 또는 인터 예측을 사용하여 코딩되고 디코딩될 수 있는 것일 수 있다. 유사하게, 다중-예측 픽처들은 단일 블록의 재구성을 위해 2개보다 많은 참조 픽처 및 연관된 메타데이터를 사용할 수 있다.
- [0078] 소스 픽처들은 흔히 복수의 샘플 블록(예를 들어, 각각 4x4, 8x8, 4x8, 또는 16x16 샘플들의 블록들)으로 공간적으로 세분되고 블록 바이 블록(block-by-block) 기준으로 코딩될 수 있다. 블록들은 블록들의 각자의 픽처들에 적용되는 코딩 할당에 의해 결정된 다른 (이미 코딩된) 블록들을 참조하여 예측적으로 코딩될 수 있다. 예를 들어, I 픽처들의 블록들은 비예측적으로 코딩될 수 있거나 또는 그것들은 동일한 픽처의 이미 코딩된 블록들을 참조하여 예측적으로 코딩될 수 있다(공간 예측 또는 인트라 예측). P 픽처의 픽셀 블록들은, 하나의 이전에 코딩된 참조 픽처를 참조하여 공간 예측을 통해 또는 시간적 예측을 통해, 예측적으로 코딩될 수 있다. B 픽처들의 블록들은, 하나 또는 2개의 이전에 코딩된 참조 픽처를 참조하여 공간 예측을 통해 또는 시간적 예측

을 통해, 예측적으로 코딩될 수 있다.

- [0079] 비디오 인코더(503)는 ITU-T Rec. H.265와 같은 표준 또는 미리 결정된 비디오 코딩 기술에 따라 코딩 동작들을 수행할 수 있다. 그것의 동작 중에, 비디오 인코더(503)는, 입력 비디오 시퀀스에서 시간적 및 공간 중복성을 이용하는 예측 코딩 동작들을 포함하여, 다양한 압축 동작들을 수행할 수 있다. 따라서, 코딩된 비디오 데이터는 사용 중인 비디오 코딩 기술 또는 표준에 의해 특정된 신택스(syntax)를 따를 수 있다.
- [0080] 일 실시예에서, 송신기(540)는 인코딩된 비디오와 함께 추가적인 데이터를 송신할 수 있다. 소스 코더(530)는 코딩된 비디오 시퀀스의 일부로서 그러한 데이터를 포함할 수 있다. 추가적인 데이터는 시간적/공간/SNR 향상 계층들, 중복 픽처들 및 슬라이스들과 같은 다른 형식의 중복 데이터, SEI 메시지들, VUI 파라미터 세트 프래그먼트들 등을 포함할 수 있다.
- [0081] 비디오는 시간적 시퀀스에서 복수의 소스 픽처(비디오 픽처들)로서 캡처될 수 있다. 인트라-픽처 예측(중중 인트라 예측으로 축약됨)은 주어진 픽처에서 공간 상관을 이용하고, 인터-픽처 예측은 픽처들 사이의 (시간적 또는 다른) 상관을 이용한다. 일 예에서, 현재 픽처라고 지칭되는, 인코딩/디코딩 중인 특정 픽처가 블록들로 파티셔닝된다. 현재 픽처 내의 블록이 비디오 내의 이전에 코딩되고 여전히 버퍼링된 참조 픽처 내의 참조 블록과 유사할 때, 현재 픽처 내의 블록은 모션 벡터라고 지칭되는 벡터에 의해 코딩될 수 있다. 모션 벡터는 참조 픽처 내의 참조 블록을 가리키고, 다수의 참조 픽처가 사용 중인 경우, 참조 픽처를 식별하는 제3의 차원을 가질 수 있다.
- [0082] 일부 실시예들에서, 인터-픽처 예측에서 양방향 예측(bi-prediction) 기법이 사용될 수 있다. 양방향 예측 기법에 따르면, 둘 다 비디오 내의 현재 픽처에 디코딩 순서에서 앞서서 (그러나, 디스플레이 순서에서, 과거 및 미래에 제각기 있을 수 있는) 제1 참조 픽처 및 제2 참조 픽처와 같은 2개의 참조 픽처가 사용된다. 현재 픽처 내의 블록은 제1 참조 픽처 내의 제1 참조 블록을 가리키는 제1 모션 벡터, 및 제2 참조 픽처 내의 제2 참조 블록을 가리키는 제2 모션 벡터에 의해 코딩될 수 있다. 블록은 제1 참조 블록과 제2 참조 블록의 조합에 의해 예측될 수 있다.
- [0083] 또한, 코딩 효율을 개선하기 위해 인터-픽처 예측에서 병합 모드(merge mode) 기법이 사용될 수 있다.
- [0084] 본 개시내용의 일부 실시예들에 따르면, 인터-픽처 예측들 및 인트라-픽처 예측들과 같은 예측들이 블록들의 단위로 수행된다. 예를 들어, HEVC 표준에 따르면, 비디오 픽처들의 시퀀스 내의 픽처는 압축을 위해 코딩 트리 유닛(coding tree unit, CTU)들로 파티셔닝되고, 픽처 내의 CTU들은 64x64 픽셀들, 32x32 픽셀들, 또는 16x16 픽셀들과 같은 동일한 크기를 갖는다. 일반적으로, CTU는 3개의 코딩 트리 블록(coding tree block, CTB)을 포함하는데, 이것은 하나의 루마 CTB 및 2개의 크로마 CTB이다. 각각의 CTU는 하나 또는 다중의 CU로 재귀적으로 쿼드-트리 스플릿될 수 있다. 예를 들어, 64x64 픽셀들의 CTU는 64x64 픽셀들의 하나의 CU, 또는 32x32 픽셀들의 4개의 CU, 또는 16x16 픽셀들의 16개의 CU로 스플릿될 수 있다. 일 예에서, 각각의 CU는, 인터 예측 타입 또는 인트라 예측 타입과 같은, CU에 대한 예측 타입을 결정하기 위해 분석된다. CU는 시간적 및/또는 공간 예측성에 의존하여 하나 이상의 예측 유닛(prediction unit, PU)으로 스플릿된다. 일반적으로, 각각의 PU는 하나의 루마 예측 블록(prediction block, PB), 및 2개의 크로마 PB를 포함한다. 일 실시예에서, 코딩(인코딩/디코딩)에서의 예측 동작은 예측 블록의 단위로 수행된다. 예측 블록의 예로서 루마 예측 블록을 사용하면, 예측 블록은, 8x8 픽셀들, 16x16 픽셀들, 8x16 픽셀들, 16x8 픽셀들 등과 같은, 픽셀들에 대한 값들(예를 들어, 루마 값들)의 행렬을 포함한다.
- [0085] 도 6은 본 개시내용의 또 다른 실시예에 따른 비디오 인코더(603)의 도면을 도시한다. 비디오 인코더(603)는 비디오 픽처들의 시퀀스에서 현재 비디오 픽처 내의 샘플 값들의 처리 블록(예를 들어, 예측 블록)을 수신하고, 처리 블록을 코딩된 비디오 시퀀스의 일부인 코딩된 픽처로 인코딩하도록 구성된다. 일 예에서, 비디오 인코더(603)는 도 3의 예에서의 비디오 인코더(303) 대신에 사용된다.
- [0086] HEVC 예에서, 비디오 인코더(603)는 8x8 샘플들 등의 예측 블록과 같은 처리 블록에 대한 샘플 값들의 행렬을 수신한다. 비디오 인코더(603)는 처리 블록이, 예를 들어, 레이트-왜곡 최적화를 이용하여 인트라 모드, 인트라 모드, 또는 양방향 예측 모드를 이용하여 최선으로 코딩되는지를 결정한다. 처리 블록이 인트라 모드로 코딩되어야 할 때, 비디오 인코더(603)는 인트라 예측 기법을 이용하여 처리 블록을 코딩된 픽처 내에 인코딩할 수 있다; 그리고 처리 블록이 인터 모드 또는 양방향 예측 모드로 코딩되어야 할 때, 비디오 인코더(603)는 처리 블록을 코딩된 픽처 내에 인코딩하기 위해 인터 예측 또는 양방향 예측 기법을 제각기 이용할 수 있다. 특정 비디오 코딩 기술들에서, 병합 모드는 예측자들 외부의 코딩된 모션 벡터 성분의 선택 없이 하나 이상의 모션 벡

터 예측자로부터 모션 벡터가 도출되는 인터 픽처 예측 서브모드일 수 있다. 특정 다른 비디오 코딩 기술들에서, 대상 블록에 적용가능한 모션 벡터 성분이 존재할 수 있다. 일 예에서, 비디오 인코더(603)는 처리 블록들의 모드를 결정하기 위한 모드 결정 모듈(도시되지 않음)과 같은 다른 컴포넌트들을 포함한다.

- [0087] 도 6의 예에서, 비디오 인코더(603)는 도 6에 도시된 바와 같이 함께 결합된 인터 인코더(inter encoder)(630), 인트라 인코더(intra encoder)(622), 잔차 계산기(residue calculator)(623), 스위치(626), 잔차 인코더(624), 일반 제어기(621), 및 엔트로피 인코더(625)를 포함한다.
- [0088] 인터 인코더(630)는 현재 블록(예를 들어, 처리 블록)의 샘플들을 수신하고, 블록을 참조 픽처들 내의 하나 이상의 참조 블록(예를 들어, 이전 픽처들 및 나중 픽처들 내의 블록들)과 비교하고, 인터 예측 정보(예를 들어, 인터 인코딩 기법에 따른 중복 정보의 설명, 모션 벡터들, 병합 모드 정보)를 생성하고, 임의의 적합한 기법을 이용하여 인터 예측 정보에 기초하여 인터 예측 결과들(예를 들어, 예측된 블록)을 계산하도록 구성된다. 일부 예들에서, 참조 픽처들은 인코딩된 비디오 정보에 기초하여 디코딩되는 디코딩된 참조 픽처들이다.
- [0089] 인트라 인코더(622)는 현재 블록(예를 들어, 처리 블록)의 샘플들을 수신하고, 일부 경우들에서 블록을 동일한 픽처 내의 이미 코딩된 블록들과 비교하고, 변환 후 양자화된 계수들을 생성하고, 일부 경우들에서 또한 인트라 예측 정보(예를 들어, 하나 이상의 인트라 인코딩 기법에 따른 인트라 예측 방향 정보)를 수신하도록 구성된다. 일 예에서, 인트라 인코더(622)는 또한 동일한 픽처 내의 참조 블록들 및 인트라 예측 정보에 기초하여 인트라 예측 결과들(예를 들어, 예측된 블록)을 계산한다.
- [0090] 일반 컨트롤러(621)는 일반 제어 데이터를 결정하고 일반 제어 데이터에 기초하여 비디오 인코더(603)의 다른 컴포넌트들을 제어하도록 구성된다. 일 예에서, 일반 컨트롤러(621)는 블록의 모드를 결정하고, 모드에 기초하여 스위치(626)에 제어 신호를 제공한다. 예를 들어, 모드가 인트라 모드일 때, 일반 컨트롤러(621)는 잔차 계산기(623)에 의한 사용을 위해 인트라 모드 결과를 선택하도록 스위치(626)를 제어하고, 인트라 예측 정보를 선택하고 인트라 예측 정보를 비트스트림에 포함시키도록 엔트로피 인코더(625)를 제어한다; 그리고 모드가 인터 모드일 때, 일반 컨트롤러(621)는 잔차 계산기(623)에 의한 사용을 위해 인터 예측 결과를 선택하도록 스위치(626)를 제어하고, 인터 예측 정보를 선택하고 인터 예측 정보를 비트스트림에 포함시키도록 엔트로피 인코더(625)를 제어한다.
- [0091] 잔차 계산기(623)는 수신된 블록과 인트라 인코더(622) 또는 인터 인코더(630)로부터 선택된 예측 결과들 사이의 차이(잔차 데이터)를 계산하도록 구성된다. 잔차 인코더(624)는 잔차 데이터에 기초하여 동작하여 잔차 데이터를 인코딩하여 변환 계수들을 생성하도록 구성된다. 일 예에서, 잔차 인코더(624)는 공간 도메인에서 주파수 도메인으로 잔차 데이터를 변환하고, 변환 계수들을 생성하도록 구성된다. 그 후 변환 계수들에 대해 양자화 처리를 수행하여 양자화된 변환 계수들을 획득한다. 다양한 실시예들에서, 비디오 인코더(603)는 잔차 디코더(628)를 또한 포함한다. 잔차 디코더(628)는 역변환을 수행하고, 디코딩된 잔차 데이터를 생성하도록 구성된다. 디코딩된 잔차 데이터는 인트라 인코더(622) 및 인터 인코더(630)에 의해 적합하게 사용될 수 있다. 예를 들어, 인터 인코더(630)는 디코딩된 잔차 데이터 및 인터 예측 정보에 기초하여 디코딩된 블록들을 생성할 수 있고, 인트라 인코더(622)는 디코딩된 잔차 데이터 및 인트라 예측 정보에 기초하여 디코딩된 블록들을 생성할 수 있다. 디코딩된 블록들은 디코딩된 픽처들을 생성하기 위해 적합하게 처리되고 디코딩된 픽처들은 메모리 회로(도시되지 않음)에 버퍼링되고 일부 예들에서 참조 픽처들로서 사용될 수 있다.
- [0092] 엔트로피 인코더(625)는 인코딩된 블록을 포함하기 위해 비트스트림을 포맷하도록 구성된다. 엔트로피 인코더(625)는 HEVC 표준과 같은 적합한 표준에 따라 다양한 정보를 포함하도록 구성된다. 일 예에서, 엔트로피 인코더(625)는 일반 제어 데이터, 선택된 예측 정보(예를 들어, 인트라 예측 정보 또는 인터 예측 정보), 잔차 정보, 및 다른 적합한 정보를 비트스트림에 포함시키도록 구성된다. 개시된 주제에 따르면, 인터 모드 또는 양방향 예측 모드의 병합 서브모드에서 블록을 코딩할 때, 잔차 정보가 존재하지 않는다는 점에 유의한다.
- [0093] 도 7은 본 개시내용의 또 다른 실시예에 따른 비디오 디코더(710)의 도면을 도시한다. 비디오 디코더(710)는 코딩된 비디오 시퀀스의 일부인 코딩된 픽처들을 수신하고, 코딩된 픽처들을 디코딩하여 재구성된 픽처들을 생성하도록 구성된다. 일 예에서, 비디오 디코더(710)는 도 3의 예에서의 비디오 디코더(310) 대신에 사용된다.
- [0094] 도 7의 예에서, 비디오 디코더(710)는 도 7에 도시된 바와 같이 함께 결합된 엔트로피 디코더(771), 인터 디코더(780), 잔차 디코더(773), 재구성 모듈(774), 및 인트라 디코더(772)를 포함한다.
- [0095] 엔트로피 디코더(771)는, 코딩된 픽처로부터, 코딩된 픽처가 구성되는 선택스 요소들을 나타내는 특정 심벌들을

재구성하도록 구성될 수 있다. 그러한 심벌들은, 예를 들어, 블록이 코딩되는 모드(예컨대, 예를 들어, 인트라 모드, 인터 모드, 양방향 예측(bi-predicted) 모드, 후자의 둘은 병합 서브모드 또는 또 다른 서브모드에서), 인트라 디코더(772) 또는 인터 디코더(780) 제각기에 의한 예측을 위해 사용되는 특정 샘플 또는 메타데이터를 식별할 수 있는 예측 정보(예컨대, 예를 들어, 인트라 예측 정보 또는 인터 예측 정보), 예를 들어, 양자화된 변환 계수들의 형식으로 된 잔차 정보 등을 포함할 수 있다. 일 예에서, 예측 모드가 인터 또는 양방향 예측 모드일 때, 인터 예측 정보가 인터 디코더(780)에 제공된다; 그리고 예측 타입이 인트라 예측 타입일 때, 인트라 예측 정보가 인트라 디코더(772)에 제공된다. 잔차 정보에 대해 역양자화가 수행될 수 있고 이것은 잔차 디코더(773)에 제공된다.

[0096] 인터 디코더(780)는 인터 예측 정보를 수신하고, 인터 예측 정보에 기초하여 인터 예측 결과들을 생성하도록 구성된다.

[0097] 인트라 디코더(772)는 인트라 예측 정보를 수신하고, 인트라 예측 정보에 기초하여 예측 결과들을 생성하도록 구성된다.

[0098] 잔차 디코더(773)는 역양자화를 수행하여 탈양자화된(de-quantized) 변환 계수들을 추출하고, 탈양자화된 변환 계수들을 처리하여 잔차를 주파수 도메인으로부터 공간 도메인으로 변환하도록 구성된다. 잔차 디코더(773)는 또한 (양자화기 파라미터(QP)를 포함하도록) 특정 제어 정보를 요구할 수 있고, 그 정보는 엔트로피 디코더(771)에 의해 제공될 수 있다(이는 단지 저용량 제어 정보일 수 있으므로 데이터 경로가 묘사되지 않음).

[0099] 재구성 모듈(774)은, 공간 도메인에서, 잔차 디코더(773)에 의해 출력된 잔차와 예측 결과들(경우에 따라 인터 또는 인트라 예측 모듈에 의해 출력된 것)을 조합하여 재구성된 블록을 형성하도록 구성하고, 재구성된 블록은 재구성된 픽처의 일부일 수 있고, 재구성된 픽처는 결국 재구성된 비디오의 일부일 수 있다. 시각적 품질을 개선하기 위해 디블로킹 동작 등과 같은 다른 적합한 동작들이 수행될 수 있다는 점에 유의한다.

[0100] 비디오 인코더들(303, 503, 및 603), 및 비디오 디코더들(310, 410, 및 710)은 임의의 적합한 기법을 이용하여 구현될 수 있다는 점에 유의한다. 일 실시예에서, 비디오 인코더들(303, 503, 및 603), 및 비디오 디코더들(310, 410, 및 710)은 하나 이상의 집적 회로를 이용하여 구현될 수 있다. 또 다른 실시예에서, 비디오 인코더들(303, 503, 및 603), 및 비디오 디코더들(310, 410, 및 710)은 소프트웨어 명령어들을 실행하는 하나 이상의 프로세서를 사용하여 구현될 수 있다.

[0101] **II. 인터 예측을 위한 삼각형 파티션 모드(Triangle Partition Mode, TPM)**

[0102] 일부 경우들에서, 인터 예측을 위해 TPM이 지원될 수 있다. TPM은 8x8 이상인 CU들에만 적용될 수 있다. TPM은 정규 병합 모드, MMVD 모드, CIIP 모드, 및 서브블록 병합 모드와 같은 다른 병합 모드들과 함께, CU-레벨 플래그를 한 종류의 병합 모드로서 사용하여 시그널링될 수 있다.

[0103] TPM이 사용될 때, CU는 예를 들어, 도 8a 및 8b에 도시된 바와 같이 대각선 스플릿 또는 반-대각선(anti-diagonal) 스플릿을 이용하여 2개의 삼각형 형상의 파티션으로 균일하게 스플릿될 수 있다. CU 내의 각각의 삼각형 파티션은 그 자신의 모션 파라미터들을 이용하여 인터-예측될 수 있다. 각각의 파티션에 대해 단방향 예측만이 허용될 수 있다. 즉, 각각의 파티션은 하나의 모션 벡터 및 하나의 참조 인덱스를 가진다. 단방향 예측 모션 제약은 삼각형 파티션이 종래의 양방향 예측과 동일한 것을 보장하기 위해 적용된다. 즉, 각각의 CU에 대해 2개의 모션 보상된 예측만이 필요하다. 각각의 파티션에 대한 단방향 예측 모션은 도 1a 내지 도 1d에 기술된 프로세스를 사용하여 도출될 수 있다.

[0104] TPM이 현재 CU에 대해 사용되는 경우, TPM의 방향(대각선 또는 반-대각선) 및 2개의 병합 인덱스(각각의 파티션에 대해 하나씩)를 표시하는 플래그가 추가로 시그널링될 수 있다. 최대 TPM 후보 크기는 슬라이스 레벨에서 명시적으로 시그널링될 수 있으며, TPM 병합 인덱스들에 대한 선택스 이진화를 지정할 수 있다. 삼각형 파티션들 각각을 예측한 후에, 대각선 또는 반-대각선 에지를 따른 샘플 값들이 적응 가중 값들을 갖는 블렌딩 프로세스를 사용하여 조절될 수 있다. 전체 CU에 대한 예측 신호를 도출한 후에, 다른 예측 모드들에서와 같이 변환 및 양자화 프로세스가 전체 CU에 추가로 적용될 수 있다. 마지막으로, TPM을 이용하여 예측되는 CU의 모션 필드가 저장될 수 있다.

[0105] 일부 예들에서, TPM은 서브블록 변환(subblock transform, SBT)과 조합되어 사용될 수 없다. 즉, 시그널링된 삼각형 모드가 1과 동일할 때, cu\_sbt\_flag는 시그널링 없이 0인 것으로 추론된다.

[0106] **II.1 단방향 예측 후보 리스트 구성**

[0107] 단방향 예측 후보 리스트는 확장된 병합 예측 프로세스에 따라 구성된 병합 후보 리스트로부터 직접 도출될 수 있다. 삼각형 단방향 예측 후보 리스트에서의 단방향 예측 모션의 인덱스로서 N을 표시한다. X가 N의 패리티와 동일한, N번째 확장된 병합 후보의 LX 모션 벡터는 TPM을 위한 N번째 단방향 예측 모션 벡터로서 사용된다. 이러한 모션 벡터들은 도 9에서 "X"로 마킹된다. N번째 확장된 병합 후보의 대응하는 LX 모션 벡터가 존재하지 않는 경우, 동일한 후보의 L(1-X) 모션 벡터가 TPM에 대한 단방향 예측 모션 벡터로서 대신 사용된다.

[0108] **II.2 삼각형 파티션 에지를 따른 블렌딩**

[0109] 각각의 삼각형 파티션을 그 자신의 모션 파라미터들을 이용하여 예측한 후에, 2개의 예측 신호에 블렌딩 프로세스를 적용하여 대각선 또는 반-대각선 에지 주변의 샘플들을 도출할 수 있다. 도 10a 및 도 10b에 도시된 바와 같이, 다음의 가중 값들: 루마에 대한 {7/8, 6/8, 5/8, 4/8, 3/8, 2/8, 1/8} 및 크로마에 대한 {6/8, 4/8, 2/8}이 블렌딩 프로세스에서 사용된다.

[0110] 일부 경우들에서, TPM에서 코딩된 CU의 모션 벡터들은 다음의 프로세스를 사용하여 생성될 수 있다. Mv1 및 Mv2가 상이한 참조 픽처 리스트들로부터의 것이면(예를 들어, 하나는 L0로부터의 것이고 다른 하나는 L1로부터의 것임), Mv1 및 Mv2는 양방향 예측 모션 벡터를 형성하기 위해 간단히 조합된다. 그렇지 않고, Mv1 및 Mv2가 동일한 리스트로부터 온 것이면, 단방향 예측 모션 Mv2만이 저장된다.

[0111] **III. 기하학적 병합 모드(Geometric Merge Mode, GEO)**

[0112] 기하학적 병합 모드(GEO)와 관련된 기법들은 (i) Han Gao, Semih Esenlik, Elena Alshina, Anand Meher Kotra, Biao Wang, Max Blaser, Johannes Sauer, "CE4: CE4-1.1, CE4-1.2 and CE4-1.14: Geometric Merge Mode (GEO)", JVET-P0068, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 16th Meeting: Geneva, CH, 1-11 October 2019; (ii) Han Gao, Semih Esenlik, Elena Alshina, Anand Meher Kotra, Biao Wang, Max Blaser, Johannes Sauer, "Simplified GEO without multiplication and minimum blending mask storage", JVET-P0884, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 16th Meeting: Geneva, CH, 1-11 October 2019; (iii) Kevin Reuzel, Chun-Chi Chen<sup>2</sup>, Han Huang<sup>3</sup>, Wei-Jung Chien, Vadim Seregin, Marta Karczewicz, Ru-Ling Liao<sup>4</sup>, Jie Chen<sup>5</sup>, Yan Ye, Jiancong Luo, Max Blaser, Johannes Sauer of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 16th Meeting: Geneva, CH, 1-11 October 2019에 설명되어 있다. JVET-P0068, JVET-P0884, 및 JVET-P0085의 문헌들은 본 명세서에 참조로 통합된다.

[0113] GPM(geometric partitioning mode)이라고도 지칭되는 GEO(geometric merge mode)는 복수의 상이한 파티셔닝 방식을 지원할 수 있다. 파티셔닝 방식들은 각도들 및 에지들에 의해 정의될 수 있다. 예를 들어, 140개의 상이한 파티셔닝 방식이 CU의 중심에 대해 32개의 각도(11.25°의 동일한 간격으로 0과 360° 사이에서 양자화됨) 및 5개의 에지에 의해 구별될 수 있다.

[0114] 도 11은 예시적인 GEO를 도시한다. 도 11에서, 각도  $\Phi_i$ 는 0도와 360도 사이의 양자화된 각도를 나타내고, 거리 오프셋  $\rho_i$ 는 최대 거리  $\rho_{max}$ 의 양자화된 오프셋을 나타낸다.  $\rho_{max}$ 의 값은 수학적 식 5에 의해 8과 동일하고  $\log_2$  스케일링된 짧은 에지 길이로 스케일링된 w 또는 h에 대해 기하학적으로 도출될 수 있다. 변수들 h 및 w는 현재 블록의 높이 및 폭을 나타낸다.  $\Phi$ 가 0°와 같을 때,  $\rho_{max}$ 는 w/2와 동일하다.  $\Phi$ 가 90°와 동일할 때,  $\rho_{max}$ 는 h/2와 동일하다. 수학적 식 1에서,  $\rho_{margin} = 1.0$ 인 것은 스플릿 경계가 현재 블록의 코너에 너무 가까워지는 것을 방지하기 위한 것이다.

[0115] [수학적 식 5]

[0116] 
$$\rho_{max}(\varphi, w, h) = \cos(\varphi) \left( \frac{h}{2 \tan(\frac{\pi}{2} - \varphi)} + \frac{w}{2} \right) - \rho_{margin}, 0 < \varphi < \frac{\pi}{2}$$

[0117] GEO에서의 각각의 파티션 모드(즉, 각도 인덱스와 에지 인덱스의 쌍)는 2개의 파티셔닝된 부분에서의 샘플들을 블렌딩하기 위해 픽셀-적용 가중치 테이블을 할당받을 수 있다. 샘플의 가중 값은 예를 들어, 0 내지 8의 범위일 수 있고 픽셀의 중심 위치로부터 에지까지의 L2 거리에 의해 결정될 수 있다. 가중 값들이 할당될 때 유닛-이득 제약을 따를 수 있다. 예를 들어, 작은 가중 값이 GEO 파티션에 할당될 때, 큰 상보적인 것이 다른 파티

선에 할당되어, 합산하여 8이 된다.

[0118] **IV. GEO에 대한 가중된 샘플 예측 프로세스**

[0119] **IV.1 블렌딩 가중치 계산**

[0120] GEO에서, 최종 샘플 예측자  $P_B$ 는 수학적 식 6에 따라 2개의 3-비트 블렌딩 마스크(즉, 가중 값들 또는 가중 인자들)  $W_0$  및  $W_1$  및 2개의 예측자  $P_0$  및  $P_1$ 로 도출될 수 있다.

[0121] [수학적 식 6]

[0122] 
$$P_B = (W_0 P_0 + W_1 P_1 + 4) \gg 3$$

[0123] 블렌딩 마스크들  $W_0$  및  $W_1$ 는 그들의 가중 인덱스들에 기초하여 룩업 테이블로부터 도출될 수 있다. (일부 상황들에서 블렌딩 가중치라고도 지칭되는) 블렌딩 가중치 인덱스는 수학적 식 7에 도시된 바와 같이 샘플 위치(x, y)와 스플릿 경계 간의 거리에 기초하여 도출될 수 있다.

[0124] [수학적 식 7]

[0125] 
$$wIdx(x, y) = x \times \cos(\varphi) + y \times \cos\left(\varphi + \frac{\pi}{2}\right) - \left(\rho + \frac{w}{2} \times \cos(\varphi) + \frac{h}{2} \times \cos\left(\varphi + \frac{\pi}{2}\right)\right)$$

[0126] 여기서,

[0127] [수학적 식 8]

[0128] 
$$\rho = i \times \frac{\rho_{\max}(\varphi, w, h)}{n}$$

[0129] [수학적 식 9]

[0130] 
$$= i \times \left(\left(\frac{h}{2} \times \tan(\varphi) + \frac{w}{2}\right) \times \cos(\varphi) - \rho_{\text{margin}}\right) / n$$

[0131] [수학적 식 10]

[0132] 
$$= i \times \left(\left(\frac{w}{2} \times \cos(\varphi) + \frac{h}{2} \times \cos\left(\varphi + \frac{\pi}{2}\right)\right) - \rho_{\text{margin}}\right) / n$$

[0133] 수학적 식 10에서, n은 총 거리 양자화 스텝들을 나타내고, i는 각도  $\varphi$ 를 갖는 GEO에 대한 거리 스텝 인덱스를 나타내고,  $i < n$ 이다. 마지막으로, 샘플의 블렌딩 마스크들  $W_0$  및  $W_1$ (또는 가중 값들)은 수학적 식 11에 도시된 바와 같이, GeoFilter로 표시되는 표 3을 사용하여 설정될 수 있다.

[0134] [수학적 식 11]

[0135] 
$$weight(x, y) = wIdx(x, y) \leq 0 ? GeoFilter[[wIdx(x, y)]] : 8 - GeoFilter[[wIdx(x, y)]]$$

[0136] 예시적인 가중된 샘플 예측 프로세스는 다음과 같이 설명된다. 이 프로세스에 대한 입력들은 현재 코딩 블록의 폭 및 높이를 지정하는 2개의 변수 nCbW 및 nCbH, 2개의 (nCbW)x(nCbH) 어레이 predSamplesLA 및 predSamplesLB, 기하학적 파티션의 각도 인덱스를 지정하는 변수 angleIdx, 기하학적 파티션의 거리 idx를 지정하는 변수 distanceIdx, 및 컬러 성분 인덱스를 지정하는 변수 cIdx를 포함한다. 이 프로세스의 출력들은 예측 샘플 값들의 (nCbW)x(nCbH) 어레이 pbSamples 및 (nCbW>>2)x(nCbH>>2) 어레이 motionIdx를 포함한다.

[0137] 변수 bitDepth는 다음과 같이 도출될 수 있다: cIdx=0일 경우, bitDepth=BitDepthY; 그렇지 않으면, bitDepth=BitDepthC.

[0138] 변수들 shift1=Max(5, 17-bitDepth)이고 offset1=1<<(shift1-1)이다.

[0139]  $x=0..nCbW-1$  및  $y=0..nCbH-1$ 을 갖는 루마에 대한 가중치 어레이 sampleWeightL[x][y] 및 크로마에 대한 sampleWeightC[x][y]는 다음과 같이 도출될 수 있다:

- [0140] 변수들  $wIdx = \log_2(nCbW)$  및  $hIdx = \log_2(nCbH)$ .
- [0141] 변수  $whRatio = (wIdx \geq hIdx) ? wIdx - hIdx : hIdx - wIdx$ ,  $scaleIdx = (wIdx \geq hIdx) ? hIdx - 3 : wIdx - 3$ .
- [0142] 변수들  $displacementX = angleIdx$  및  $displacementY = (displacementX + 8) \% 32$ .
- [0143] 변수  $angleN = (wIdx \geq hIdx) ? ((angleIdx >> 3 \& 1) ? angleIdx \% 8 : 8 - angleIdx \% 8 : (angleIdx >> 3 \& 1) ? 8 - angleIdx \% 8 : angleIdx \% 8$ .
- [0144] 변수 rho는 표 1 및 표 2에 지정된 stepDis 및 Dis로 표시된 룩업 테이블들을 사용하여 다음의 값에 설정될 수 있다.  
 $rho = distanceIdx * (stepDis[whRatio][angleN] \ll scaleIdx) + (Dis[displacementX] \ll wIdx) + (Dis[displacementY] \ll hIdx)$ .
- [0145] 변수 weightIdx 및 weightIdxAbs는  $x = 0 \dots nCbW - 1$  및  $y = 0 \dots nCbH - 1$ 인 룩업 테이블 2를 사용하여 계산될 수 있다.  
 $weightIdx = ((x \ll 1) + 1) * Dis[displacementX] + ((y \ll 1) + 1) * Dis[displacementY] - rho$ .  $weightIdxAbs = Clip3(0, 26, (abs(weightIdx) + 4) \gg 3)$ .
- [0146]  $x = 0$ 이고  $y = nCbH - 1$ 인 경우, 변수 partIdx는  $weightIdx > 0$ 에 설정될 수 있다.
- [0147]  $x = 0 \dots nCbW - 1$  및  $y = 0 \dots nCbH - 1$ 인 sampleWeightL[x][y]의 값은 GeoFilter로서 표시되는 표 3에 따라 설정될 수 있다.  
 $sampleWeightL[x][y] = weightIdx \leq 0 ? GeoFilter[weightIdxAbs] : 8 - GeoFilter[weightIdxAbs]$ .
- [0148]  $x = 0 \dots nCbW - 1$  및  $y = 0 \dots nCbH - 1$ 인 값 sampleWeightC[x][y]는 다음과 같이 설정될 수 있다:  
 $sampleWeightC[x][y] = sampleWeightL[(x \ll (SubWidthC - 1))][y \ll (SubHeightC - 1)]$ .

표 1

<b>whRatio</b>	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
<b>angleN</b>	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8
<b>stepDis [whRatio] [angleN]</b>	77	95	108	116	119	116	108	95	77	77	115	147	173	192	202	203	195	179
<b>whRatio</b>	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
<b>angleN</b>	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8
<b>stepDis [whRatio] [angleN]</b>	77	155	226	287	336	372	392	396	384	77	235	382	515	626	712	770	798	794

[0149]

표 2

<b>idx</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>Dis[idx]</b>	64	63	59	53	45	36	24	12	0	-12	-24	-36	-45	-53	-59	-63
<b>idx</b>	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
<b>Dis[idx]</b>	-64	-63	-59	-53	-45	-36	-24	-12	0	12	24	36	45	53	59	63

[0150]

표 3

<b>idx</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13
<b>GeoFilter[idx]</b>	4	4	4	4	5	5	5	5	5	5	5	6	6	6
<b>idx</b>	14	15	16	17	18	19	20	21	22	23	24	25	26	
<b>GeoFilter[idx]</b>	6	6	6	6	7	7	7	7	7	7	7	7	8	

[0151]

[0152] IV.2 최소 블렌딩 가중치 마스크 저장

[0153] 사전 계산된 블렌딩 마스크들의 저장 요구를 줄이기 위하여, 최소 블렌딩 마스크 저장 방법은 블렌딩 가중치 저장에 대한 84-91% 메모리 요구 감소를 달성할 수 있다.

[0154]  $g\_sampleWeight_L[]$ 가 가중치들을 블렌딩하기 위한 미리 정의된 마스크들을 나타낸다고 하자.  $N$ 이 각각의 세트에서 미리 정의된 마스크들의 수를 나타내고,  $N$ 이  $NA \gg 1$ 로 설정되는 것으로 가정하고, 여기서  $NA$ 는 GEO에서 지원되는 각도들의 수이다.  $M \times M$ 은 블렌딩 가중치들에 대한 미리 정의된 마스크들의 크기를 나타내고,  $M$ 은  $128 + ((ND-1) \times (128 \gg S)) \ll 1$ 로 설정되며,  $ND$ 는 GEO에서 지원되는 스텝들의 수이고,  $S$ 는  $ND-1$ 에 설정된다. 5 스텝 설정을 갖는 32개의 각도에 대해,  $N$ 은 16에 설정되고  $M$ 은 192에 설정된다. 4 스텝 설정을 갖는 24개의 각도에 대해,  $N$ 은 12에 설정되고  $M$ 은 224에 설정된다.

[0155] 기하학적 파티셔닝 인덱스  $K$ 를 갖는 크기  $W \times H$ 의 블록에 대해, 루마 샘플들에 대한 블렌딩 가중치들은 다음과 같이 도출된다. 변수들 각도  $\phi$  및 거리  $\rho$ 는 기하학적 파티셔닝 인덱스  $K$ 를 이용하여 룩업 테이블로부터 획득된다. 변수들  $offsetX$  및  $offsetY$ 는 다음과 같이 계산될 수 있다:

[0156] [수학식 12]

$$offsetX = \begin{cases} (M - W) \gg 1, \phi \% N = 8 \text{ 또는 } (\phi \% N \neq 0 \text{ 및 } H \geq W) \\ ((M - W) \gg 1) + \phi < N ? (\rho \times W) \gg S : -((\rho \times W) \gg S), \text{ 다른 경우} \end{cases}$$

[0157]

[0158] [수학식 13]

$$offsetY = \begin{cases} ((M - H) \gg 1) + \phi < N ? (\rho \times H) \gg S : -((\rho \times H) \gg S), \\ \phi \% N = 8 \text{ 또는 } (\phi \% N \neq 0 \text{ 및 } H \geq W) \\ (M - H) \gg 1, \text{ 다른 경우} \end{cases}$$

[0159]

[0160] [수학식 14]

$$sampleWeight_L[x][y] = g\_sampleWeight_L[\phi \% N][x + offsetX][y + offsetY]$$

[0161]

[0162] GEO에 대한 최소 블렌딩 가중치 마스크 저장을 갖는 예시적인 가중된 샘플 예측 프로세스는 다음과 같이 설명된다. 이 프로세스에 대한 입력들은 현재 코딩 블록의 폭 및 높이를 지정하는 2개의 변수  $nCbW$  및  $nCbH$ , 2개의  $(nCbW) \times (nCbH)$  어레이  $predSamplesLA$  및  $predSamplesLB$ , 기하학적 파티셔닝의 각도 인덱스를 지정하는 변수  $angleIdx$ , 기하학적 파티셔닝의 거리  $idx$ 를 지정하는 변수  $distanceIdx$ , 및 컬러 성분 인덱스를 지정하는 변수  $cIdx$ 를 포함한다. 이 프로세스의 출력들은 예측 샘플 값들의  $(nCbW) \times (nCbH)$  어레이  $pbSamples$  및 변수  $partIdx$ 를 포함한다.

[0163] 변수  $bitDepth$ 는 다음과 같이 도출될 수 있다:  $cIdx=0$ 일 경우,  $bitDepth=bitDepthY$ ; 그렇지 않으면,  $bitDepth=bitDepthC$ .

[0164] 변수들  $shift1=Max(5, 17-bitDepth)$ 이고  $offset1=1 \ll (shift1-1)$ 이다.

[0165]  $x=0..nCbW-1$  및  $y=0..nCbH-1$ 을 갖는 루마에 대한 가중치 어레이  $sampleWeightL[x][y]$  및 크로마에 대한  $sampleWeightC[x][y]$ 는 다음과 같이 도출될 수 있다:

[0166] 변수  $hwRatio=nCbH/nCbW$ .

[0167] 변수들  $displacementX=angleIdx$  및  $displacementY=(displacementX+8)\%32$ .

[0168] 변수  $partIdx=(angleIdx>13 \&\& angleIdx \leq 27) ? 1:0$ .

[0169] 변수  $\rho$ 는 룩업 테이블 2를 사용하여 다음의 값에 설정될 수 있다.  $\rho=(Dis[displacementX] \ll 8) + (Dis[displacementY] \ll 8)$ .

[0170] 다음 조건들 중 하나가 참이면, 변수  $shiftHor=0$ 이다: (1)  $angleIdx \% 16=8$ ; 및 (2)  $angleIdx \% 16! = 0$  및  $hwRatio \geq 1$ . 그렇지 않으면,  $shiftHor=1$ 이다.

[0171]  $shiftHor=0$ 이면,  $offsetX=(256-nCbW) \gg 1$ ,  $offsetY=(256-nCbH) \gg 1 + angleIdx < 16 ? (distanceIdx * nCbH) \gg 3 : -$

((distanceIdx\*nCbH)>>3).

[0172] 그렇지 않고 shiftHor=1이면,  $offsetX=(256-nCbW)>>1+angleIdx<16?(distanceIdx*nCbW)>>3:-((distanceIdx*nCbW)>>3),$

[0173]  $offsetY=(256-nCbH)>>1.$

[0174] 변수 weightIdx 및 weightIdxAbs는 다음과 같이  $x=0..nCbW-1$  및  $y=0..nCbH-1$ 인 룩업 테이블 2를 사용하여 계산될 수 있다:  
 $weightIdx(((x+offsetX)<<1)+1)*Dis[displacementX]+(((y+offsetY)<<1)+1)*Dis[displacementY]-rho,$   
 $weightIdxAbs=Clip3(0, 26, abs(weightIdx)).$

[0175]  $x=0..nCbW-1$  및  $y=0..nCbH-1$ 인 sampleWeightL[x][y]의 값은 GeoFilter로서 표시되는 표 3에 따라 설정될 수 있다.  $sampleWeightL[x][y]=weightIdx<=0?GeoFilter[weightIdxAbs]:8-GeoFilter[weightIdxAbs].$

[0176]  $x=0..nCbW-1$  및  $y=0..nCbH-1$ 인 값 sampleWeightC[x][y]는 다음과 같이 설정될 수 있다:  
 $sampleWeightC[x][y]=sampleWeightL[(x<<(SubWidthC-1))][(y<<(SubHeightC-1))].$

[0177] **V. GEO를 위한 모션 벡터 저장 프로세스**

[0178] **V.1 TPM-유사 모션 벡터 도출**

[0179] 일부 경우들에서, 4x4 모션 저장 유닛의 4개의 코너에서의 루미넌스 샘플 가중치들이 합산될 수 있다. 합은 2개의 단방향 예측 모션 정보 및 양방향 예측 모션 정보 중 하나가 저장되는지를 결정하기 위해 2개의 임계값과 비교될 수 있다. 양방향 예측 모션 정보는 TPM과 동일한 프로세스를 이용하여 도출될 수 있다.

[0180] GEO에 대한 예시적인 모션 벡터 저장 프로세스는 다음과 같이 설명된다.

[0181]  $xSbIdx=0..(nCbW>>2)-1$  및  $ySbIdx=0..(nCbH>>2)-1$ 인 어레이 motionIdx[xSbIdx][ySbIdx]는 다음과 같이 도출될 수 있다:

[0182] 변수들  $threshScaler=(wIdx+hIdx)>>1)-1,$   $threshLower =32>>threshScaler,$   $threshUpper=32-threshLower,$   
 $Cnt=sampleWeightL[(xSbIdx<<2)][(ySbIdx<<2)]+sampleWeightL[(xSbIdx<<2)+3][(ySbIdx<<2)]+sampleWeightL[(xSbIdx<<2)+3][(ySbIdx<<2)+3],$   $Cnt=partIdx?32-Cnt:Cnt,$   
 $motionIdx[xSbIdx][ySbIdx]=Cnt<=threshLower?0:Cnt>=threshUpper?1:2.$

[0183]  $merge\_geo\_flag[xCb][yCb]=1$ 인 경우,  $xSbIdx=0..numSbX-1,$  및  $ySbIdx=0..numSbY-1$ 인  $sType=motionIdx[xSbIdx][ySbIdx].$

[0184] **V.2 단순화된 모션 벡터 저장 프로세스**

[0185] 일부 경우들에서, 모션 벡터 저장 프로세스는 추가로 단순화된다. 4x4 모션 저장 유닛의 중심 위치와 스플릿 경계 간의 거리가 계산되고 고정 임계값과 비교되어, 4x4 모션 저장 유닛에 대해 단방향 또는 양방향 예측 모션 정보가 저장되는지를 결정할 수 있다. 거리의 부호는 어느 단방향 예측 모션 정보가 단방향 예측 저장 경우에 저장되어야 하는지를 나타낸다. 블렌딩 마스크 및 모션 저장의 의존성이 제거될 수 있다.

[0186]  $merge\_geo\_flag[xCb][yCb]=1$ 인 경우,  $wIdx=log2(cbWidth),$   $hIdx=log2(cbHeight),$   $whRatio=(wIdx >=hIdx)?wIdx-hIdx:hIdx-wIdx,$   $scaleIdx=(wIdx>=hIdx)?hIdx-3:wIdx-3,$   $displacementX=angleIdx,$   
 $displacementY=(displacementX+8)\%32,$   $angleN=(wIdx>=hIdx)?(angleIdx>>3&\&1)?angleIdx\%8:8-angleIdx\%8:(angleIdx>>3&\&1)?8-angleIdx\%8:angleIdx\%8.$

[0187] 변수 rho는 표 1 및 표 2에 지정된 stepDis 및 Dis로 표시된 룩업 테이블들을 사용하여 다음의 값에 설정된다.  $rho=distanceIdx*(stepDis[whRatio][angleN]<<scaleIdx)+(Dis[displacementX]<<wIdx)+(Dis[displacementY]<<hIdx).$

[0188] 변수 motionOffset은 표 1 및 표 2에 지정된 Dis로 표시된 룩업 테이블들을 사용하여 다음의 값에 설정된다.  $motionOffset=3*Dis[displacementX]+3*Dis[displacementY].$

[0189] 변수 motionIdx는 다음과 같이 룩업 테이블 2를 사용하여 계산된다:  $motionIdx=((xSbIdx<<3)+1)*Dis[displacementX]+((xSbIdx<<3)+1)*Dis[displacementY]-rho+motionOffset.$

[0190] 변수 sType은 다음과 같이 도출된다:  $partIdx=1$ 인 경우,  $sType=abs(motionIdx)<256?2:motionIdx<=0?1:0;$  그렇지

지 않으면,  $sType = \text{abs}(\text{motionIdx}) < 256 ? 2 : \text{motionIdx} \leq 0 ? 0 : 1$ .

[0191] **V.3 최소 모션 벡터 저장 프로세스**

[0192] 모션 필드 저장을 위한 마스크들을 저장하는 데 필요한 메모리를 줄이기 위해, 프로세스에서, 미리 정의된 마스크로부터의 모든 정보가 모션 필드 저장의 마스크들에 대해 도출될 수 있다. 이 프로세스는 MergeWedgeFlag[xCb][yCb]=1로 코딩 유닛을 디코딩할 때 호출된다. 이 프로세스에 대한 입력들은 현재 픽처의 상단 좌측 루마 샘플에 대한 현재 코딩 블록의 상단 좌측 샘플을 특징하는 루마 위치 (xCb, yCb), 루마 샘플들에서 현재 코딩 블록의 폭을 특징하는 변수 cbWidth, 루마 샘플들에서 현재 코딩 블록의 높이를 특징하는 변수 cbHeight, 1/16 분수-샘플 정확도 mvA 및 mvB에서의 루마 모션 벡터들, 참조 인덱스들 refIdxA 및 refIdxB, 및 예측 리스트 플래그들 predListFlagA 및 predListFlagB를 포함한다.

[0193] 수평 및 수직 방향에서의 현재 코딩 블록 내의 4x4 블록의 수를 특징하는 변수들 numSbX 및 numSbY는  $\text{numSbX} = \text{cbWidth} \gg 2$  및  $\text{numSbY} = \text{cbHeight} \gg 2$ 에 동일하게 설정된다.

[0194] 변수들 displacementX=angleIdx, displacementY=(displacementX+8)%32, hwRatio=nCbH/ nCbW.

[0195] 다음 조건들 중 하나가 참이면, 변수 shiftHor=0이다: (1) angleIdx%16=8; 및 (2) angleIdx%16!=0 및 hwRatio ≥ 1. 그렇지 않으면, shiftHor=1이다.

[0196] 변수 partIdx=(angleIdx>=13&&angleIdx<=27)?1:0.

[0197] shiftHor=0인 경우, offsetX=(64-numSbX)>>1, offsetY=(64-numSbY)>>1+angleIdx<16?(distanceIdx\*nCbH)>> 5:-((distanceIdx\*nCbH)>>5). 그렇지 않고 shiftHor=1인 경우, offsetX=(64-numSbX)>>1+angleIdx<16?(distanceIdx\*nCbW)>>5:-((distanceIdx\*nCbW)>> 5), offsetY=(64- numSbY)>>1.

[0198] 변수 rho의 값은 이하의 수학적 및 표 2에 지정된 Dis 룩업 테이블에 따라 도출된다.  $\text{rho} = (\text{Dis}[\text{displacementX}] \ll 8) + (\text{Dis}[\text{displacementY}] \ll 8)$ .

[0199] 변수 motionOffset는 표 1 및 표 2에 지정된, Dis로 표시된 룩업 테이블들을 사용하여 다음의 값에 동일하게 설정된다.  $\text{motionOffset} = 3 * \text{Dis}[\text{displacementX}] + 3 * \text{Dis}[\text{displacementY}]$ .

[0200] xSbIdx=0..numSbX-1, 및 ySbIdx=0..numSbY-1인 서브블록 인덱스(xSbIdx, ySbIdx)에서의 각각의 4x4 서브블록에 대해, 변수 motionIdx는 다음과 같이 룩업 테이블 2를 사용하여 계산된다:  $\text{motionIdx} = (((\text{xSbIdx} + \text{offsetX}) \ll 3) + 1) * \text{Dis}[\text{displacementX}] + (((\text{xSbIdx} + \text{offsetY} \ll 3) + 1) * \text{Dis}[\text{displacementY}] - \text{rho} + \text{motionOffset}$ .

[0201] **VI. 블렌딩 프로세스 및 모션 필드 저장**

[0202] 섹션 III, IV 및 V에 기술된 GEO에서, 가중된 샘플 예측 프로세스 및 모션 필드 저장 결정은 계산상 비용이 많이 든다. (예를 들어, 섹션 IV.1 및 V.1에서 기술된 바와 같은) 일부 예들에서, 블렌딩 가중치들 및 저장된 모션 벡터 타입이 온-더-플라이(on-the-fly)로 계산되는 경우, 크기 WxH 샘플들의 블록에 대한 곱셈들, 시프트들 및 덧셈들의 수는 다음과 같을 수 있다:

[0203] - 곱셈:  $3 + 2xWxH + 0x(WxH/16)$

[0204] - 시프트:  $6 + 3xWxH + 8x(WxH/16)$

[0205] - 덧셈:  $10 + 6xWxH + 7x(WxH/16)$

[0206] 예를 들어, 8x8 블록에 대해, 샘플당 2.04 곱셈, 3.09 시프트 및 6.16 덧셈이 요구된다. 또한, 일부 예들에서 온-더-플라이 계산들을 위해 3개의 룩업 테이블이 사용된다. 수학적 4에서의 p는 부동 소수점 연산들을 갖는 복잡한 함수이고, 룩업 테이블을 사용하여 실현된다. 다른 2개의 룩업 테이블은 블렌딩 가중치 인덱스를 블렌딩 가중치 값으로 변환하기 위한 cos[.] 값 룩업 테이블 및 변환 테이블이다. 따라서, 각각의 GEO 코딩된 블록들에 대해 블렌딩 가중치 마스크들 및 모션 벡터 타입 마스크들을 온-더-플라이로 계산하는 것은 계산적으로 부담이 된다.

[0207] 계산 복잡도를 감소시키기 위해, 블렌딩 가중치 마스크들 및 모션 벡터 저장 타입 마스크들은 일부 예들에서 미리 계산되고 메모리에 저장될 수 있다. 그러나, 거대한 메모리 저장 크기는 설계 과제이다. GEO의 140-모드 변형을 예로 들면, 그 정보를 저장하는 데 필요한 메모리는 다음과 같이 계산될 수 있다:

[0208] - 블렌딩 가중치들에 대해:  $(8x8+8x16+8x32+8x64+16x8+16x16+16x32+16x64+32x8+32x16+32x32+32x64+64x8+64x16+64x32+64x64+64x128+128x64+128x128) \times 140 \times 4 = 26,414,080 \text{ bits} = 3,301,760 \text{ bytes} \approx 3.3 \text{ Mbytes}$

[0209] - 모션 필드 저장에 대해:  $(2x2+2x4+2x8+2x16+4x2+4x4+4x8+4x16+8x2+8x4+8x8+8x16+16x2+16x4+16x8+16x16+16x32+32x16+32x32) \times 140 \times 2 = 825,440 \text{ bits} = 103,180 \text{ bytes} \approx 103 \text{ Kbytes}$

[0210] 위의 계산에 기초하여, GEO의 3개의 변형에 대한 메모리 저장 요건이 표 4에 열거된다:

표 4

	140 파티셔닝 모드	108 파티셔닝 모드	80 파티셔닝 모드
블렌딩 가중치들	3,301,760 bytes	2,547,072 bytes	1,886,720 bytes
모션 저장	103,180 bytes	79,596 bytes	58,960 bytes
총	3,404,940 bytes ≈ 3.4 Mbytes	2,626,668 bytes ≈ 2.6 Mbytes	1,945,680 bytes ≈ 1.9 Mbytes

[0211]

[0212] 많은 수의 파티셔닝 모드로 인해, 실제 구현에서 각각의 블록 크기 및 파티셔닝 모드에 대한 모션 필드 저장을 위해 모든 적응적 가중치들 및 모션 벡터 타입 마스크들을 저장하는 것은 어려울 수 있다. 모션 필드 저장을 위한 모션 벡터 타입 마스크들 및 적응적 가중치들을 저장하는 데 필요한 메모리를 줄이기 위해, 미리 정의된 마스크들의 2개의 세트, 블렌딩 가중치 도출을 위한 하나 및 모션 필드 저장의 마스크들을 위한 다른 하나가 섹션들 IV.2 및 V.3에서 설명된 기술들을 이용하여 채택될 수 있다. 블렌딩 가중치들을 위해 필요한 메모리 및 모션 필드 저장을 위한 마스크가 감소될 수 있다. 그러나, 요구되는 메모리의 양은 실제 구현들에 대해 여전히 크다.  $N$ 이  $N_A \gg 1$ 로 설정되고,  $N_A$ 가 GEO에서 지원되는 각도들의 수이고,  $M$ 이  $128 + ((N_b - 1) \times (128 \gg S)) \ll 1$ 로 설정되고,  $N_b$ 가 GEO 모드에서 지원되는 스텝들의 수이고,  $S$ 가  $N_b - 1$ 로 설정된다고 가정한다. 미리 정의된 마스크들을 저장하는 데 필요한 비트들의 수는 다음과 같이 열거된다:

[0213] - 블렌딩 가중치들의 경우:  $(M \times M) \times N \times 4$

[0214] - 모션 필드 저장의 경우:  $(M \times M) / 16 \times N \times 2$

[0215] GEO의 3개의 변형에 대한 메모리 저장 요건이 아래의 표 5에 열거된다:

표 5

	140 파티셔닝 모드	108 파티셔닝 모드	80 파티셔닝 모드
블렌딩 가중치들	294,912 bytes	401,408 bytes	301,056 bytes
모션 저장	9,216 bytes	12,544 bytes	9,408 bytes
총	304,128 bytes	413,952 bytes	310,464 bytes
이전 설계	3,404,940 bytes	2,626,668 bytes	1,945,680 bytes
감소 %	91%	84%	84%

[0216]

[0217] GEO에서의 가중된 블렌딩 프로세스 및 모션 필드 저장 프로세스를 더 단순화하기 위한 실시예들이 아래에 설명된다. 이들 실시예에서의 기법은 삼각 값들(예를 들어, 코사인 값들)의 작은 룩업 테이블로 온-더-플라이 가중 인자 계산 및 모션 벡터 저장 타입 결정을 가능하게 한다.

[0218] 설명된 바와 같이, 기하학적 병합 모드의 블렌딩 가중치 마스크들은 이하에 표현된 수학적 식 7 및 수학적 식 9로 룩업 테이블들을 이용하여 스플릿 경계 및 샘플 위치의 거리로부터 도출될 수 있다.

[0219] [수학식 7]

$$wIdx(x, y) = x \times \cos(\varphi) + y \times \cos\left(\varphi + \frac{\pi}{2}\right) - \left(\rho + \frac{w}{2} \times \cos(\varphi) + \frac{h}{2} \times \cos\left(\varphi + \frac{\pi}{2}\right)\right)$$

[0220]

[0221] 여기서,

[0222] [수학식 9]

$$\rho = i \times \frac{\left(\frac{h}{2} \times \tan(\varphi) + \frac{w}{2}\right) \times \cos(\varphi) - \rho_{\text{margin}}}{n}, \text{ 여기서 } 0 \leq \varphi < \frac{\pi}{2}$$

[0223]

[0224] 수학식 3에서,  $\varphi$  는 파티션 에지에 대응하는 각도 값을 나타내고,  $\rho$  는 현재 코딩 블록의 중심과 스플릿 경계 간의 거리이다. 수학식 5에서,  $n$  은 거리 양자화 스텝들의 총 수를 나타내고,  $i$  는 각도  $\varphi$  를 갖는 GEO 파티션에 대한 거리 스텝 인덱스를 나타낸다. 수학식 5에서 뒤로 시프트된  $\rho_{\text{margin}}$  값은 스플릿 경계가 현재 코딩 블록의 코너에 너무 가까운 것을 피하기 위한 것이다. 일 예로서, 8x8 블록에 대한  $\rho_{\text{margin}}$  의 전형적인 값은 일부 예들에서 1.0일 수 있다.

[0225] **블록 및 각도 의존  $\rho_{\text{margin}}$**

[0226] 일 실시예에서, 스플릿 경계(파티션 에지)가 코딩 블록의 코너에 너무 가까운 것을 회피하기 위해 사용되는, 수학식 5에서의 값  $\rho_{\text{margin}}$  은 상수가 아니다. 예를 들어, 값  $\rho_{\text{margin}}$  은 블록 폭 및/또는 높이 및/또는  $\varphi$  (스플릿 경계의 각도)에 좌우되어 변할 수 있다.

[0227] 값  $\rho_{\text{margin}}$  은 GEO 파티션 각도  $\varphi$  와 현재 코딩 블록의 폭 및 높이의 함수인 것으로 정의될 수 있다. 예를 들어,  $\rho_{\text{margin}}$  은 다음의 수학식들로부터 도출될 수 있다:

[0228] [수학식 15]

$$\rho_{\text{margin}} = ((w \gg 3) \times \cos(\varphi) + (h \gg 3) \times \tan(\varphi)) \times \cos(\varphi)$$

[0229]

[0230] [수학식 16]

$$= (w \gg 3) \times \cos(\varphi) + (h \gg 3) \times \cos\left(\varphi + \frac{\pi}{2}\right)$$

[0231]

[0232] 수학식 17 및 수학식 18에 따르면, 거리 오프셋  $\rho$  는 아래에 도시된 바와 같이 단순화될 수 있다.

[0233] [수학식 17]

$$\rho = i \times \frac{\cos(\varphi) \times \left(\frac{w}{2} - (w \gg 3)\right) - \cos\left(\varphi + \frac{\pi}{2}\right) \times \left(\frac{h}{2} - (h \gg 3)\right)}{n}$$

[0234]

[0235] [수학식 18]

$$= \frac{3iw}{8n} \times \cos(\varphi) - \frac{3ih}{8n} \times \cos\left(\varphi + \frac{\pi}{2}\right)$$

[0236]

[0237] 따라서, 거리 오프셋  $\rho$  의 계산은 코사인 값 룩업 테이블 연산들 외에 시프트, 곱셈, 및 덧셈 연산들에 의해 달

성될 수 있다. 계산은 상수  $\rho_{\text{margin}}$  값을 사용하는 것과 비교하여 단순화되었다.

[0238] 단일 룩업 테이블에 의한 온-더-플라이 블렌딩 가중치 인덱스 계산

[0239] 일 실시예에서, 코사인 계산을 위한 룩업 테이블만이 블렌딩 가중치 인덱스 계산 동안 사용된다. 그 결과, 모든 가중 인덱스 계산은 블렌딩 가중치 마스크들을 저장하기 위한 큰 테이블을 체크하지 않고서 온-더-플라이로 구현될 수 있다.

[0240] 상기 수학식 18에 기초하여, 수학식 3에 나타낸 바와 같이, 샘플 위치로부터 스플릿 경계까지의 거리로부터 도출된 GEO의 블렌딩 마스크들은 다음과 같이 더 단순화될 수 있다.

[0241] [수학식 19]

$$wIdx(x, y) = \left(x - \frac{w}{2}\right) \times \cos(\varphi) + \left(y - \frac{h}{2}\right) \times \cos\left(\varphi + \frac{\pi}{2}\right) - \rho$$

[0242]

[0243] [수학식 20]

$$= \left(x - \frac{w}{2}\right) \times \cos(\varphi) + \left(y - \frac{h}{2}\right) \times \cos\left(\varphi + \frac{\pi}{2}\right) - \left(\frac{3iw}{8n} \times \cos(\varphi) - \frac{3ih}{8n} \times \cos\left(\varphi + \frac{\pi}{2}\right)\right)$$

[0244]

[0245] [수학식 21]

$$= \left(x - \frac{w \times (4n + 3i)}{8n}\right) \times \cos(\varphi) - \left(y - \frac{h \times (4n + 3i)}{8n}\right) \times \cos\left(\varphi + \frac{\pi}{2}\right)$$

[0246]

[0247] 거리 스텝 수  $n$ 이 2의 거듭제곱의 값이면, 수학식 21에서의 나눗셈 연산들은 수학식 22에 나타낸 바와 같이 우측 시프트들 전에 라운딩 오프셋을 갖거나 갖지 않고서 우측 시프트 연산들을 사용하여 실현될 수 있다.

[0248] [수학식 22]

$$wIdx(x, y) = \left(x - \left((w \times (4n + 3i)) \gg (3 + \log_2 n)\right)\right) \times \cos(\varphi) - \left(y - \left((h \times (4n + 3i)) \gg (3 + \log_2 n)\right)\right) \times \cos\left(\varphi + \frac{\pi}{2}\right)$$

[0249]

[0250] 수학식 22에 따라서, 각각의 샘플에 대한 블렌딩 가중치 인덱스는 곱셈기들, 덧셈기, 및 하나의  $\cos[.]$  룩업 테이블을 이용하여 획득될 수 있다. 또한, 수학식 20은 단 하나의  $\cos[.]$  테이블이 현재 CU 내의 모든 샘플들의 블렌딩 가중치 인덱스를 위해 사용된다는 것을 나타낸다. 그것은 또한 이  $\cos[.]$  값이 한 번 액세스될 수 있고 그 후 가중치 인덱스 계산을 위해 현재 CU 내의 모든 샘플들에 전파될 수 있다는 힌트를 준다.

[0251] 정수 전용 계산

[0252] 일 실시예에서, 부동 소수점 연산들을 피하기 위해, 블렌딩 가중치 인덱스를 도출하기 위한 성분들이 스케일 업되어, 가중치 인덱스의 스케일 업 값을 낳을 수 있다. 스케일 업 값은 나중에 그의 정상 값 범위로 다시 스케일 업될 수 있다.

[0253] 예를 들어,  $\cos[.]$  룩업 테이블에서의 각각의 코사인 값은  $m$ -비트 고정 소수점 값으로서 실현될 수 있고, 여기서  $m$ 은 6 또는 3과 같은 양의 값일 수 있다. 게다가, 샘플 중심은 또한 블렌딩 가중치 인덱스 계산을 위해 이산 샘플링을 갖는 디지털 이미지에서 채택된다. 가중치 인덱스  $wIdx(x, y)$ 는 부동 소수점 연산을 회피하고 이산 샘플링 상황을 보상하기 위해  $2^{m+4}$ 의 스케일링 인자로 스케일링될 수 있다. 수학식 22는 따라서 수학식 23으로 다시 쓰여질 수 있다. 비교적 높은 정밀도를 유지하기 위해, 라운딩 오프셋이 적용될 수 있다.

[0254] [수학식 23]

$$wIdx(x, y) = \left( \left( ((x \ll 1) + 1) \ll 3 \right) - \frac{\left( (w \times (4n + 3i)) \ll 1 \right)}{n} \right) \times \cos[\varphi] - \left( \left( ((y \ll 1) + 1) \ll 3 \right) - \frac{\left( (h \times (4n + 3i)) \ll 1 \right)}{n} \right) \times \cos \left[ \varphi + \frac{\pi}{2} \right]$$

[0255]

[0256] 여기서,  $\cos[\varphi]$  및  $\cos \left[ \varphi + \frac{\pi}{2} \right]$  은 m-비트 룩업 테이블로서 구현된다.

[0257] 거리 스텝 수 n이 2의 거듭제곱의 값이면, 수학식 23에서의 나눗셈 연산들은 수학식 24에 나타낸 바와 같이 우측 시프트 연산들을 사용하여 실현될 수 있다. 라운딩 오프셋이 또한 적용될 수 있다.

[0258] [수학식 24]

$$wIdx(x, y) = \left( \left( ((x \ll 1) + 1) \ll 3 \right) - \left( (w \times (4n + 3i)) \ll 1 \right) \gg \log_2 n \right) \times \cos[\varphi] - \left( \left( ((y \ll 1) + 1) \ll 3 \right) - \left( (h \times (4n + 3i)) \ll 1 \right) \gg \log_2 n \right) \times \cos \left[ \varphi + \frac{\pi}{2} \right]$$

[0259]

[0260] 4개의 사분면에서의 각도들

[0261] 일 실시예에서, 가중 인덱스 계산은 스플릿 경계의 각도  $\Phi$  의 사분면에 의존하여 상이한 공식들을 사용하여 구현될 수 있다. 예를 들어, 스플릿 경계 계산에 있어서,  $\cos[\varphi]$  및  $\cos \left[ \varphi + \frac{\pi}{2} \right]$  의 값 둘 모두는 양일 수 있다. 따라서, 4개의 상이한 블렌딩 가중치 마스크 공식이 4개의 상이한 사분면에 대응하는 파티션 각도 인덱스에 대해 사용될 수 있다. 4개의 상이한 사분면에 대한 블렌딩 가중치 인덱스 수학식이 아래에 나타내어진다.

[0262] [수학식 25]

$$wIdx(x, y) =$$

$$\begin{cases} \left( x - \frac{w \times (4n + 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n + 3i)}{8n} \right) \times \cos \left( \varphi + \frac{\pi}{2} \right), & 0 \leq \varphi < \frac{\pi}{2} \text{ 인 경우} \\ \left( x - \frac{w \times (4n - 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n + 3i)}{8n} \right) \times \cos \left( \varphi + \frac{\pi}{2} \right), & \frac{\pi}{2} \leq \varphi < \pi \text{ 인 경우} \\ \left( x - \frac{w \times (4n - 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n - 3i)}{8n} \right) \times \cos \left( \varphi + \frac{\pi}{2} \right), & \pi \leq \varphi < \frac{3\pi}{2} \text{ 인 경우} \\ \left( x - \frac{w \times (4n + 3i)}{8n} \right) \times \cos(\varphi) - \left( y - \frac{h \times (4n - 3i)}{8n} \right) \times \cos \left( \varphi + \frac{\pi}{2} \right), & \frac{3\pi}{2} \leq \varphi < 2\pi \text{ 인 경우} \end{cases}$$

[0263]

[0264] 행별 및 열별 기반 블렌딩 가중치 인덱스 계산

[0265] 일 실시예에서, 가중치 인덱스  $wIdx(x, y)$ 는 행별로 계산될 수 있는데, 즉, 현재 행의 값  $wIdx(x, y)$ 는 이전 행의 값  $wIdx(x, y-1)$ 에 기초하여 도출된다. 대안적으로, 가중치 인덱스  $wIdx(x, y)$ 는 열별로 계산될 수 있는데, 즉, 현재 열의 가중치 인덱스들  $wIdx(x, y)$ 는 이전 열의 가중치 인덱스들  $wIdx(x-1, y)$ 에 기초하여 도출된다.

[0266] 샘플 위치 (x,y)에 대해, 블렌딩 마스크는 수학적식 23에 기초하여 수학적식 26-28을 이용하여 도출될 수 있다.

[0267] [수학적식 26]

$$wIdx(x,y) = \left( \left( \left( \left( (x-1) + 1 \right) \ll 1 \right) + 1 \right) \ll 3 \right) - \frac{\left( (w \times (4n + 3i)) \ll 1 \right)}{n} \right) \times \cos[\varphi] \\ - \left( \left( \left( (y \ll 1) + 1 \right) \ll 3 \right) - \frac{\left( (h \times (4n + 3i)) \ll 1 \right)}{n} \right) \times \cos \left[ \varphi + \frac{\pi}{2} \right]$$

[0268]

[0269] [수학적식 27]

$$= (1 \ll 4) \times \cos[\varphi] + \left( \left( \left( (x-1) \ll 1 \right) + 1 \right) \ll 3 \right) - \frac{\left( (w \times (4n + 3i)) \ll 1 \right)}{n} \right) \\ \times \cos[\varphi] - \left( \left( (y \ll 1) + 1 \right) \ll 3 \right) - \frac{\left( (h \times (4n + 3i)) \ll 1 \right)}{n} \right) \\ \times \cos \left[ \varphi + \frac{\pi}{2} \right]$$

[0270]

[0271] [수학적식 28]

$$= (1 \ll 4) \times \cos[\varphi] + wIdx(x-1,y)$$

[0272]

[0273] 수학적식 26에 나타낸 바와 같이, 가중치 인덱스 wIdx(x,y)는 고정된 오프셋 값을 갖는 이전 열의 이웃하는 좌측 픽셀의 블렌딩 가중치 인덱스로부터 도출될 수 있다. 고정된 오프셋 값은  $\cos(\varphi)$ 이고, 이것은 스케일링 인자를 갖는, 대응하는 각도 인덱스를 갖는 간단한 룩업 테이블 값이다. 따라서, 블렌딩 가중치 인덱스 값들의 하나의 열만이 마스크 저장을 위해 요구된다. 가중치 인덱스들의 다른 열들은 고정된 오프셋 값에 의해 이전 열의 블렌딩 가중치 인덱스 값들을 사용함으로써 결정될 수 있다.

[0274] 샘플 위치 (x,y)에 대해, 이 픽셀 샘플의 블렌딩 마스크 또는 블렌딩 가중치 인덱스는 또한 수학적식 23에 기초하여 수학적식 29-31을 사용하여 도출될 수 있다.

[0275] [수학적식 29]

$$wIdx(x,y) = \left( \left( \left( (x \ll 1) + 1 \right) \ll 3 \right) - \frac{\left( (w \times (4n + 3i)) \ll 1 \right)}{n} \right) \times \cos[\varphi] \\ - \left( \left( \left( \left( (y-1) + 1 \right) \ll 1 \right) + 1 \right) \ll 3 \right) - \frac{\left( (h \times (4n + 3i)) \ll 1 \right)}{n} \right) \\ \times \cos \left[ \varphi + \frac{\pi}{2} \right]$$

[0276]

[0277] [수학식 30]

$$\begin{aligned}
 &= (1 \ll 4) \times \cos \left[ \varphi + \frac{\pi}{2} \right] + \left( \left( ((x \ll 1) + 1) \ll 3 \right) - \frac{((w \times (4n + 3i)) \ll 1)}{n} \right) \\
 &\quad \times \cos[\varphi] - \left( \left( ((y - 1) \ll 1) + 1 \right) \ll 3 \right) - \frac{((h \times (4n + 3i)) \ll 1)}{n} \\
 &\quad \times \cos \left[ \varphi + \frac{\pi}{2} \right]
 \end{aligned}$$

[0278]

[0279] [수학식 31]

$$= (1 \ll 4) \times \cos \left[ \varphi + \frac{\pi}{2} \right] + wIdx(x, y - 1)$$

[0280]

[0281] 수학식 29에서, 가중치 인덱스  $wIdx(x, y)$ 는 고정된 오프셋 값에 의해 상기 행의 상부 이웃 픽셀의 블렌딩 가중

치 인덱스로부터 도출될 수 있다. 고정된 오프셋 값,  $(1 \ll 4) \times \cos \left[ \varphi + \frac{\pi}{2} \right]$ 은  $\frac{\pi}{2}$  위상 시프트를 갖는 주어진 각도 인덱스를 사용함으로써 간단한 룩업 테이블  $\cos[\cdot]$ 로부터 도출될 수 있다. 따라서, 블렌딩 가중치 값들의 한 행만이 마스크 저장을 위해 요구된다. 다른 행들은 고정된 오프셋 값에 의해 이전 행의 블렌딩 가중치 값들을 이용함으로써 구현될 수 있다.

[0282] **가중 인덱스들의 독립적인 도출**

[0283] 일 실시예에서, 코딩 블록에서의 위치  $(x, y)$ 에서의 가중치 인덱스는, 좌표  $(x, y)$  및 코사인 테이블에 의존하여, 특정의 가중치 인덱스 값  $wIdx(0, 0)$ (이것은 블록 크기, 거리 오프셋  $\rho$ , 파티션 각도  $\varphi$ 에 기초하여 결정될 수 있음)으로부터 도출될 수 있다. 이러한 방식으로, 코딩 블록에서의 각각의 샘플의 가중 인덱스는 독립적으로 도출될 수 있다.

[0284] 수학식 7로부터 아래의 수학식이 도출될 수 있다:

[0285] [수학식 32]

$$wIdx(x, y) = wIdx(0, 0) + x \times \cos(\varphi) + y \times \cos \left( \varphi + \frac{\pi}{2} \right)$$

[0286]

[0287] 여기서,

[0288] [수학식 33]

$$wIdx(0, 0) = - \left( \rho + \frac{w}{2} \times \cos(\varphi) + \frac{h}{2} \times \cos \left( \varphi + \frac{\pi}{2} \right) \right)$$

[0289]

[0290] 따라서,  $wIdx(0, 0)$ 의 값을 결정한 후에, 블록에서의 각각의 샘플의 가중치 인덱스는 병렬로 수학식 32에 의해 계산될 수 있다.

[0291] **가중치 인덱스로부터 블렌딩 가중치로의 룩업 테이블 자유 변환**

[0292] 일 실시예에서, 룩업 테이블 기반 가중치 인덱스 대 최종 블렌딩 가중치 변환이 계산 기반 변환에 의해 대체된다. 룩업 테이블을 체크하는 동작들이 제거됨에 따라, 각각의 샘플의 최종 가중 인자가 병렬로 도출될 수 있다.

[0293] 섹션 VI에서 설명된 GEO의 가중된 샘플 프로세스와 같은 일부 예들에서, 샘플 가중치의 값은 룩업 테이블(예를 들어,  $GeoFilter[\ ]$  룩업 테이블, 표 3)에 따라 설정된다. 룩업 테이블과 연관된 제한을 제거하고 가중 도출을 병렬로 가능하게 하기 위해, 샘플의 가중치 값이 가중치 인덱스의 선형 함수에 기초하여 온-더-플라이로 도출될 수 있다. 선형 함수는 룩업 테이블의 도움 없이 가중치 인덱스를 블렌딩 가중치로 변환할 수 있다.

[0294] 일 예로서, 아래의 수학식 (수학식 34)은 가중치 인덱스를 블렌딩 가중치로 변환하기 위해 채택될 수 있다.

- [0295] [수학식 34]
- [0296]  $weight(x, y) = (2^{idx2wShiftBit+2} - wIdx(x, y)) \gg idx2wShiftBit$
- [0297] 부동 소수점 연산 기반 계산(예를 들어, 수학식 19 및 수학식 20)에 대해  $wIdx(x, y)$ 를 계산하기 위해 어떤 스케일링 업도 사용되지 않으며 수학식 32에서의  $idx2wShiftBit$ 는 "0"일 수 있다. 샘플 가중치는 다음과 같이 도출될 수 있다.
- [0298] [수학식 35]
- [0299]  $weight(x, y) = 4 - wIdx(x, y)$
- [0300] 고정 소수점 연산 기반 계산(예를 들어, 수학식 19 및 수학식 20)에 대해, 수학식 30에서의  $idx2wShiftBit$ 가 수학식 36에서와 같이 기술될 수 있다.
- [0301] [수학식 36]
- [0302]  $idx2wShiftBit = m + 3$
- [0303] 샘플 가중치는 수학식 37에 나타낸 바와 같이 우측 시프트 연산을 사용하여 온-더-플라이로 도출될 수 있다.
- [0304] [수학식 37]
- [0305]  $weight(x, y) = (2^{m+5} - wIdx(x, y)) \gg (m + 3)$
- [0306]  $m$ 의 전형적인 값은 코사인 테이블의 정밀도 비트 수일 수 있다. 예를 들어,  $m$ 은 일반적으로 6 또는 3일 수 있다.
- [0307] **록업 테이블 자유 모션 인덱스 계산**
- [0308] 일 실시예에서, 모션 인덱스를 결정하기 위해 록업 테이블 자유 계산 방법이 채택될 수 있다. 예를 들어, 현재 코딩 블록에서의 모션 저장 유닛의 모션 인덱스는, 예를 들어, 모션 저장 유닛의 좌표  $(x_{sb}, y_{sb})$  및 파티션 거리 오프셋  $\rho$  ( $n$  및  $i$ 로 표현됨), 파티션 각도  $\phi$ , 및 현재 코딩 블록의 블록 크기에 기초하여 결정될 수 있다. 그 결과, 현재 코딩 블록에서의 모션 인덱스들  $motionIdx(x_{sb}, y_{sb})$ 은 독립적으로 (병렬로) 동시에 도출될 수 있다.
- [0309] 일부 예들에서, TPM-유사 모션 저장 프로세스의 활용 없이, 모션 벡터 저장 프로세스는 모션 저장 유닛(서브블록)의 중심 위치와 스플릿 경계 사이의 거리를 사용함으로써 온-더-플라이로 도출될 수 있다. 모션 저장 유닛은  $4 \times 4$  샘플들의 크기, 또는 다른 서브블록 크기들(예를 들어,  $8 \times 8$  샘플들)을 가질 수 있다.  $w \times h$  서브블록의 중심 위치는  $(W/2, H/2)$ , 또는  $(W/2-1, H/2-1)$ , 또는 그것의 이웃 위치들에서의 정수 위치로서 정의될 수 있다. 대안적으로, 서브-펠 위치  $((W-1)/2.0, (H-1)/2.0)$ 이 중심 위치로서 정의될 수 있다. 상기 거리 값은 단방향 예측 또는 양방향 예측 모션 정보가 모션 저장 유닛에 대해 저장되는지를 결정하기 위해 고정된 임계값과 비교된다. 거리의 부호는 어느 단방향 예측 모션 정보가 단방향 예측 저장 경우에 저장될 수 있는지를 나타낸다. 따라서, 블렌딩 마스크 및 모션 저장의 의존성이 제거될 수 있다.
- [0310] 일 예에서, 모션 저장 유닛(서브블록)은  $4 \times 4$  샘플들의 크기를 갖는다. 변수들  $x_{sb}$  및  $y_{sb}$ 은 수평 및 수직 방향들에서 현재  $w \times h$  코딩 블록에서의  $4 \times 4$  서브블록의 위치를 지정한다. 변수들  $x_{sb}$  및  $y_{sb}$ 은 수학식  $0 \leq x_{sb} < (w \gg 2)$  및  $0 \leq y_{sb} < (h \gg 2)$ 을 만족한다. 수학식 21의 개념 및 수학식 19에서의 스케일링에 의한 이산 샘플링에 대한 상세한 도출을 이용하면,  $4 \times 4$  모션 저장 유닛의 중심 위치와 스플릿 경계 사이의 거리가 수학식 38에서 도출될 수 있다.

[0311] [수학식 38]

$$\begin{aligned}
 &= \left( \left( (x_{sb} \ll 3) + 1 \right) \ll 3 \right) - \frac{\left( (w \times (4n + 3i)) \ll 1 \right)}{n} + 3 \right) \times \cos[\varphi] \\
 \text{motionIdx}(x_{sb}, y_{sb}) &= \left( \left( (y_{sb} \ll 3) + 1 \right) \ll 3 \right) - \frac{\left( (h \times (4n + 3i)) \ll 1 \right)}{n} \\
 &+ 3 \right) \times \cos \left[ \varphi + \frac{\pi}{2} \right]
 \end{aligned}$$

[0312]

[0313] 거리 스텝 수  $n$ 이 2의 거듭제곱 값인 경우, 수학식 38에서의 나눗셈 연산들이 수학식 39에 나타난 바와 같이 우측 시프트 연산들을 사용하여 실현될 수 있다. 비교적 높은 정밀도를 유지하기 위해, 라운딩 오프셋이 또한 적용될 수 있다.

[0314] [수학식 39]

$$\begin{aligned}
 &= \left( \left( (x_{sb} \ll 3) + 1 \right) \ll 3 \right) - \left( \left( (w \times (4n + 3i)) \ll 1 \right) \gg \log_2 n \right) + 3 \right) \\
 &\quad \times \cos[\varphi] \\
 \text{motionIdx}(x_{sb}, y_{sb}) &= \left( \left( (y_{sb} \ll 3) + 1 \right) \ll 3 \right) \\
 &- \left( \left( (h \times (4n + 3i)) \ll 1 \right) \gg \log_2 n \right) + 3 \right) \\
 &\quad \times \cos \left[ \varphi + \frac{\pi}{2} \right]
 \end{aligned}$$

[0315]

[0316] **블렌딩 가중치 인덱스로부터 도출된 모션 인덱스**

[0317] 일 실시예에서, 코딩 블록에서의 모션 저장 유닛(서브블록)의  $\text{motionIdx}(x_{sb}, y_{sb})$ 로 표시된 모션 인덱스는 서브블록 내의 샘플 위치의 블렌딩 가중치 인덱스로부터 직접 도출된다. 이러한 방식으로, 모션 인덱스를 결정하기 위한 계산 비용이 절약되거나 감소될 수 있다.

[0318] 섹션 V에서 설명된 바와 같이, 모션 인덱스는 모션 저장 유닛의 중심 위치와 스플릿 경계 사이의 거리에 기초하여 도출될 수 있다. 블렌딩 가중치 인덱스는 또한 샘플과 스플릿 경계 간의 거리로부터 도출될 수 있다. 따라서, 모션 인덱스는 대응하는 서브블록 내에 있는 샘플의 블렌딩 가중치 인덱스로부터 근사화될 수 있고, 서브블록의 중심에 위치하지 않을 수 있다.

[0319] 예에서, 근사화는 수학식 40에 따라 구현될 수 있다.

[0320] [수학식 40]

$$\text{motionIdx}(x_{sb}, y_{sb}) = wIdx((x_{sb} \ll 2) + i, (y_{sb} \ll 2) + j), \text{ 여기서 } \begin{cases} 0 \leq i < 4 \\ 0 \leq j < 4 \end{cases}$$

[0321]

[0322] 수학식 36에서, 코딩 블록의 모션 필드에서의 모션 저장 유닛은 4x4 샘플들의 크기를 가지는 것으로 가정된다. 모션 저장 유닛 내의 샘플 위치의 좌표는 (i, j)에 의해 표현된다. 예를 들어, 모션 필드에서 (1, 0)의 좌표를 갖는 모션 저장 유닛, 및 모션 인덱스를 계산하기 위해 지정되는 모션 저장 유닛에서 (2, 2)의 좌표를 갖는 샘플 위치에 대해, 코딩 블록에서 샘플 위치  $((1 \ll 2) + 2, (0 \ll 2) + 2)$ , 즉 (6, 2)에서의 가중치 인덱스가 모션 저장 유닛의 모션 인덱스로서 사용될 수 있다. 계산 프로세스는 다음과 같이 보여질 수 있다.

[0323] [수학식 41]

$$\text{motionIdx}(1,0) = wIdx((1 \ll 2) + 2, (0 \ll 2) + 2) = wIdx(6, 2)$$

[0324]

[0325] 다양한 예들에서,  $(i, j)$ 는 대응하는 서브블록의 중심 위치  $(1.5, 1.5)$ 에 이웃하는 샘플 위치들에 대응하는  $(1,1)$ ,  $(1,2)$ ,  $(2,1)$ , 및  $(2,2)$ 와 동일할 수 있다. 더 일반적으로,  $2^n \times 2^n$  샘플들의 크기를 갖는 모션 저장 유닛의 경우, 근사화는 수학적 식 42에 따라 구현될 수 있다.

[0326] [수학적 식 42]

[0327] 
$$\text{motionIdx}(x_{sb}, y_{sb}) = \text{wIdx}((x_{sb} \ll n) + i, (y_{sb} \ll n) + j), \text{ 여기서 } \begin{cases} 0 \leq i < 2^n \\ 0 \leq j < 2^n \end{cases}$$

[0328] 수학적 식 40-42에서,  $\text{wIdx}(x, y)$ 로 표시되는 코딩 블록 내의 샘플 위치  $(x, y)$ 에서의 블렌딩 가중치 인덱스 값은 다양한 예들에서 다양한 방식들로 획득될 수 있다. 블렌딩 가중치 인덱스 값들이 가중된 블렌딩 프로세스 동안 계산된 후에, 각각의 모션 벡터 저장 유닛 내의 지정된 위치들에서의 블렌딩 가중치 인덱스들은 각각의 모션 저장 유닛에 대한 저장될 모션 벡터 타입들의 결정을 위한 모션 인덱스 값들로서 직접 사용될 수 있다. 모션 저장 유닛 내의 중심 샘플 위치의 파티션 에지까지의 거리에 기초하여 모션 인덱스를 계산하는 것과 비교하여, 계산 복잡도가 감소될 수 있다.

[0329] **VII. 기하학적 파티셔닝 모드의 예시적인 프로세스들**

[0330] 도 12는 본 개시내용의 실시예들에 따른 예시적인 프로세스(1200)를 요약하는 흐름도를 도시한다. 다양한 실시예들에서, 프로세스(1200)는 단말 디바이스들(210, 220, 230 및 240) 내의 처리 회로, 비디오 디코더(310)의 기능들을 수행하는 처리 회로, 비디오 디코더(410)의 기능들을 수행하는 처리 회로 등과 같은 처리 회로에 의해 실행된다. 일부 실시예들에서, 프로세스(1200)는 소프트웨어 명령어들로 구현되고, 따라서 처리 회로가 소프트웨어 명령어들을 실행할 때, 처리 회로는 프로세스(1200)를 수행한다. 프로세스(1200)는 (S1201)으로부터 시작하여 (S1210)로 진행할 수 있다.

[0331] (S1210)에서, 인덱스 요소가 코딩된 비디오의 비트스트림으로부터 수신될 수 있다. 인덱스 요소는 기하학적 파티셔닝 모드(geometric partitioning mode, GPM)에서의 예측을 위해 파티션 에지를 따라 제1 부분 및 제2 부분으로 파티셔닝되는 현재 코딩 블록과 연관될 수 있다. 예를 들어, 다음의 인덱스 요소들이 비트스트림에서 시그널링될 수 있다: GPM이 현재 코딩 블록에 대해 인에이블되는 것을 표시하는 CU 레벨 GPM 인에이블 플래그; 현재 코딩 블록에 적용되는 파티션 모드를 표시하는 GPM 인덱스(테이블에서의 파티션 각도 인덱스 및 거리 오프셋 인덱스에 대응함); ?? 현재 코딩 블록의 제1 부분 및 제2 부분에 대응하는 2개의 GPM 병합 인덱스.

[0332] (S1220)에서, 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스가 결정될 수 있다. 가중치 인덱스는, 예를 들어, 샘플 위치의 파티션 에지까지의 거리에 대응할 수 있다. 가중치 인덱스는 현재 코딩 블록의 제1 부분 및 제2 부분에 대응하는 2개의 예측 신호를 블렌딩하기 위한 블렌딩 가중치를 도출하기 위해 후속하여 사용될 수 있다. 일 예에서, 가중치 인덱스는 가중치 인덱스의 선형 함수를 사용하여 블렌딩 가중치로 변환될 수 있다.

[0333] (S1230)에서, 현재 코딩 블록의 모션 필드에서의 제1 모션 저장 유닛에 대한 저장된 모션 벡터 타입은 제1 모션 저장 유닛의 모션 인덱스에 기초하여 결정될 수 있다. (S1220)에서 결정된 현재 코딩 블록의 샘플 위치에서의 가중치 인덱스는 제1 모션 저장 유닛의 모션 인덱스로서 사용된다. 예를 들어, (S1220)에서 결정된 가중치 인덱스는 저장된 모션 벡터 타입을 결정하기 위해 제1 모션 저장 유닛의 중심 위치에서의 모션 인덱스를 근사화하기 위해 사용될 수 있다. 이러한 방식으로, 제1 모션 저장 유닛의 중심 위치에서 모션 인덱스를 계산하기 위한 계산 비용이 절약되거나 감소될 수 있다. GPM을 이용한 디코딩 성능이 개선될 수 있다.

[0334] (S1240)에서, 현재 코딩 블록의 모션 필드는 현재 코딩 블록의 모션 필드에서 제1 모션 저장 유닛에 대해 결정된 저장된 모션 벡터 타입에 기초하여 저장될 수 있다. 예를 들어, 2개의 GPM 병합 인덱스에 대응하는 2개의 모션 벡터는 MV1 및 MV2로 표시될 수 있다. 저장된 모션 벡터 타입에 의존하여, 기하학적 파티션의 제1 부분으로부터의 MV1, 기하학적 파티션의 제2 부분으로부터의 MV2, 및 MV1과 MV2의 조합된 MV가 제1 모션 저장 유닛에 대한 모션 필드에 저장된다. 프로세스(1200)는 (S1299)으로 진행하고, (S1299)에서 종료할 수 있다.

[0335] **VIII. 컴퓨터 시스템**

[0336] 위에서 설명된 기법들은 컴퓨터 관독가능 명령어들을 이용하여 컴퓨터 소프트웨어로서 구현되고 하나 이상의 컴퓨터 관독가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 13은 개시된 주제의 특정 실시예들을 구현하기에 적합한 컴퓨터 시스템(1300)을 도시한다.

[0337] 컴퓨터 소프트웨어는, 하나 이상의 컴퓨터 중앙 처리 유닛(CPU), 그래픽 처리 유닛(GPU) 등에 의해,

직접적으로, 또는 해석, 마이크로-코드 실행 등을 통해 실행될 수 있는 명령어들을 포함하는 코드를 생성하기 위해 어셈블리, 컴파일(compilation), 링킹(linking), 또는 유사한 메커니즘들을 겪을 수 있는 임의의 적합한 머신 코드 또는 컴퓨터 언어를 사용하여 코딩될 수 있다.

- [0338] 명령어들은, 예를 들어, 개인용 컴퓨터, 태블릿 컴퓨터, 서버, 스마트폰, 게이밍 디바이스, 사물 인터넷 디바이스 등을 포함하여, 다양한 타입의 컴퓨터들 또는 그것의 컴포넌트들 상에서 실행될 수 있다.
- [0339] 컴퓨터 시스템(1300)에 대한 도 13에 도시된 컴포넌트들은 사실상 예시적인 것이고, 본 개시내용의 실시예들을 구현하는 컴퓨터 소프트웨어의 사용 또는 기능성의 범위에 대한 임의의 제한을 시사하도록 의도되지 않는다. 컴포넌트들의 구성이 컴퓨터 시스템(1300)의 예시적인 실시예에서 예시된 컴포넌트들 중 임의의 하나 또는 이들의 조합과 관련하여 임의의 종속성 또는 요건을 갖는 것으로 해석해서도 안 된다.
- [0340] 컴퓨터 시스템(1300)은 특정 휴먼 인터페이스 입력 디바이스들을 포함할 수 있다. 그러한 휴먼 인터페이스 입력 디바이스는, 예를 들어, 촉각 입력(예컨대: 키스트로크, 스와이프, 데이터 글러브 모션), 오디오 입력(예컨대: 음성, 손뼉), 시각적 입력(예컨대, 제스처), 후각적 입력(묘사되지 않음)을 통한 하나 이상의 인간 사용자에게 의한 입력에 응답할 수 있다. 휴먼 인터페이스 디바이스들은 또한 오디오(예컨대: 음성, 음악, 주변 사운드), 이미지들(예컨대: 스캐닝된 이미지들, 스틸 이미지 카메라로부터 획득된 사진 이미지들), 비디오(예컨대 2차원 비디오, 입체적 비디오를 포함하는 3차원 비디오)와 같은, 인간에 의한 의식적인 입력과 반드시 직접적으로 관련되는 것은 아닌 특정 미디어를 캡처하기 위해 사용될 수 있다.
- [0341] 입력 휴먼 인터페이스 디바이스들은: 키보드(1301), 마우스(1302), 트랙패드(1303), 터치 스크린(1310), 데이터-글러브(도시되지 않음), 조이스틱(1305), 마이크로폰(1306), 스캐너(1307), 카메라(1308) 중 하나 이상(각각의 하나만이 묘사됨)을 포함할 수 있다.
- [0342] 컴퓨터 시스템(1300)은 특정 휴먼 인터페이스 출력 디바이스들을 또한 포함할 수 있다. 그러한 휴먼 인터페이스 출력 디바이스들은, 예를 들어, 촉각 출력, 사운드, 광, 및 냄새/맛을 통해 하나 이상의 인간 사용자의 감각들을 자극하고 있을 수 있다. 그러한 휴먼 인터페이스 출력 디바이스들은 촉각 출력 디바이스들(예를 들어, 터치-스크린(1310), 데이터-글러브(도시되지 않음), 또는 조이스틱(1305)에 의한 촉각 피드백이지만, 입력 디바이스들로서 역할하지 않는 촉각 피드백 디바이스들도 있을 수 있음), 오디오 출력 디바이스들(예컨대: 스피커들(1309), 헤드폰들(묘사되지 않음)), 시각적 출력 디바이스들(예컨대 CRT 스크린들, LCD 스크린들, 플라즈마 스크린들, OLED 스크린들을 포함하는 스크린들(1310), 각각은 터치-스크린 입력 능력이 있거나 없고, 각각은 촉각 피드백 능력이 있거나 없고 - 이들 중 일부는 스테레오그래픽 출력과 같은 수단을 통해 2차원 시각적 출력 또는 3개보다 많은 차원의 출력을 출력할 수 있음 -); 가상 현실 안경(묘사되지 않음), 홀로그래픽 디스플레이 및 연기 탱크(묘사되지 않음), 및 프린터(묘사되지 않음)를 포함할 수 있다. (스크린(1310) 등의) 이들 시각적 출력 디바이스는 그래픽 어댑터(1350)를 통해 시스템 버스(1348)에 접속될 수 있다.
- [0343] 컴퓨터 시스템(1300)은 인간 액세스가능한 저장 디바이스들 및 그것들과 연관된 매체들, 예컨대 CD/DVD 등의 매체(1321)를 갖는 CD/DVD ROM/RW(1320)를 포함하는 광학 매체, 썸-드라이브(1322), 이동식 하드 드라이브 또는 솔리드 스테이트 드라이브(1323), 테이프 및 플로피 디스크(묘사되지 않음)와 같은 레거시 자기 매체, 보안 동글(묘사되지 않음)과 같은 특수화된 ROM/ASIC/PLD 기반 디바이스들 등을 또한 포함할 수 있다.
- [0344] 본 기술분야의 통상의 기술자들은 현재 개시된 주제와 관련하여 사용되는 용어 "컴퓨터 판독가능 매체(computer readable media)"가 송신 매체들, 반송파들, 또는 다른 일시적 신호들을 포함하지 않는다는 점을 또한 이해할 것이다.
- [0345] 컴퓨터 시스템(1300)은 또한 하나 이상의 통신 네트워크(1355)에 대한 네트워크 인터페이스(1354)를 포함할 수 있다. 하나 이상의 통신 네트워크(1355)는 예를 들어, 무선, 유선, 광학적일 수 있다. 하나 이상의 통신 네트워크(1355)의 예들은 로컬 영역 네트워크들 예컨대 이더넷, 무선 LAN들, GSM, 3G, 4G, 5G, LTE 등을 포함하는 셀룰러 네트워크들, 케이블 TV, 위성 TV, 및 지상파 방송 TV를 포함하는 TV 유선 또는 무선 광역 디지털 네트워크들, CANBus를 포함하는 차량 및 산업 등을 포함한다. 특정 네트워크들은 흔히 특정 범용 데이터 포트들 또는 주변 버스들(1349)(예컨대, 예를 들어, 컴퓨터 시스템(1300)의 USB 포트들)에 부착된 외부 네트워크 인터페이스 어댑터들을 요구한다; 다른 것들은 흔히 아래에 설명되는 바와 같은 시스템 버스로의 부착에 의해 컴퓨터 시스템(1300)의 코어에 통합된다(예를 들어, PC 컴퓨터 시스템으로의 이더넷 인터페이스 또는 스마트폰 컴퓨터 시스템으로의 셀룰러 네트워크 인터페이스). 이들 네트워크들 중 임의의 것을 사용하여, 컴퓨터 시스템(1300)은 다

른 엔티티들과 통신할 수 있다. 그러한 통신은 단방향성 수신 전용(예를 들어, 방송 TV), 단방향성 송신 전용(예를 들어, CANbus 대 특정 CANbus 디바이스들), 또는 예를 들어, 로컬 또는 광역 디지털 네트워크들을 사용하는 다른 컴퓨터 시스템들과의 양방향성일 수 있다. 위에서 설명된 바와 같은 네트워크들 및 네트워크 인터페이스들 각각에 대해 특정 프로토콜들 및 프로토콜 스택들이 사용될 수 있다.

[0346] 전술한 휴먼 인터페이스 디바이스들, 인간-액세스가능한 저장 디바이스들, 및 네트워크 인터페이스들은 컴퓨터 시스템(1300)의 코어(1340)에 부착될 수 있다.

[0347] 코어(1340)는 하나 이상의 중앙 처리 유닛(CPU)(1341), 그래픽 처리 유닛(GPU)(1342), 필드 프로그래머블 게이트 영역(FPGA)(1343)의 형식으로 특수화된 프로그래머블 처리 유닛, 특정 태스크에 대한 하드웨어 가속기(1344) 등을 포함할 수 있다. 이들 디바이스는, 판독 전용 메모리(ROM)(1345), 랜덤 액세스 메모리(1346), 내부 비-사용자 액세스 가능 하드 드라이브들, SSD들 등과 같은 내부 대용량 저장소(1347)와 함께, 시스템 버스(1348)를 통해 접속될 수 있다. 일부 컴퓨터 시스템들에서, 시스템 버스(1348)는 추가적인 CPU들, GPU들 등에 의한 확장을 가능하게 하기 위해 하나 이상의 물리적 플러그의 형식으로 액세스가능할 수 있다. 주변 디바이스들은 코어의 시스템 버스(1348)에 직접적으로, 또는 주변 버스(1349)를 통해 부착될 수 있다. 주변 버스를 위한 아키텍처들은 PCI, USB 등을 포함한다.

[0348] CPU들(1341), GPU들(1342), FPGA들(1343), 및 가속기들(1344)은, 조합하여, 전술한 컴퓨터 코드를 구성할 수 있는 특정 명령어들을 실행할 수 있다. 그 컴퓨터 코드는 ROM(1345) 또는 RAM(1346)에 저장될 수 있다. 과도적인 데이터가 또한 RAM(1346)에 저장될 수 있는 반면, 영구 데이터는, 예를 들어, 내부 대용량 저장소(1347)에 저장될 수 있다. 메모리 디바이스들 중 임의의 것에 대한 고속 저장 및 검색은, 하나 이상의 CPU(1341), GPU(1342), 대용량 저장소(1347), ROM(1345), RAM(1346) 등과 밀접하게 연관될 수 있는, 캐시 메모리의 사용을 통해 가능하게 될 수 있다.

[0349] 컴퓨터 판독가능 매체는 다양한 컴퓨터 구현 동작들을 수행하기 위한 컴퓨터 코드를 그 위에 가질 수 있다. 매체 및 컴퓨터 코드는 본 개시내용의 목적을 위해 특별히 설계되고 구성된 것들일 수 있거나, 또는 그것들은 컴퓨터 소프트웨어 기술분야의 기술자들에게 잘 알려져 있고 이용가능한 종류의 것일 수 있다.

[0350] 제한이 아니라 예로서, 아키텍처를 갖는 컴퓨터 시스템(1300), 및 구체적으로 코어(1340)는 프로세서(들)(CPU들, GPU들, FPGA, 가속기들 등을 포함함)가 하나 이상의 유형의(tangible) 컴퓨터 판독가능 매체에 구현된 소프트웨어를 실행하는 결과로서 기능성을 제공할 수 있다. 그러한 컴퓨터 판독가능 매체는 위에 소개된 바와 같은 사용자-액세스가능한 대용량 저장소뿐만 아니라, 코어-내부 대용량 저장소(1347) 또는 ROM(1345)과 같은 비일시적인 본질의 것인 코어(1340)의 특정 저장소와 연관된 매체일 수 있다. 본 개시내용의 다양한 실시예들을 구현하는 소프트웨어가 그러한 디바이스들에 저장되고 코어(1340)에 의해 실행될 수 있다. 컴퓨터 판독가능 매체는 특정 필요에 따라 하나 이상의 메모리 디바이스 또는 칩을 포함할 수 있다. 소프트웨어는 코어(1340) 및 구체적으로 그 내부의 프로세서들(CPU, GPU, FPGA 등을 포함함)로 하여금, RAM(1346)에 저장된 데이터 구조들을 정의하는 것 및 소프트웨어에 의해 정의된 프로세스들에 따라 그러한 데이터 구조들을 수정하는 것을 포함하여, 본 명세서에 설명된 특정 프로세스들 또는 특정 프로세스들의 특정 부분들을 실행하게 야기할 수 있다. 추가로 또는 대안으로서, 컴퓨터 시스템은, 본 명세서에 설명된 특정 프로세스들 또는 특정 프로세스들의 특정 부분들을 실행하기 위해 소프트웨어 대신에 또는 그와 함께 동작할 수 있는, 회로(예를 들어: 가속기(1344))에 하드와이어링되거나 다른 방식으로 구현된 로직의 결과로서 기능성을 제공할 수 있다. 소프트웨어에 대한 참조는, 적절한 경우, 로직을 포함할 수 있고, 그 반대도 가능하다. 컴퓨터 판독가능 매체에 대한 참조는, 적절한 경우, 실행을 위한 소프트웨어를 저장하는 회로(예컨대 집적 회로(IC)), 또는 실행을 위한 로직을 구현하는 회로, 또는 둘 다를 포함할 수 있다. 본 개시내용은 하드웨어 및 소프트웨어의 임의의 적합한 조합을 포함한다.

[0351] 본 개시내용이 여러 예시적인 실시예들을 설명하였지만, 본 개시내용의 범위 내에 속하는 변경들, 치환들, 및 다양한 대체 균등물들이 존재한다. 따라서, 본 기술분야의 통상의 기술자들은, 비록 본 명세서에 명시적으로 도시되거나 설명되지는 않았지만, 본 개시내용의 원리들을 구현하고 따라서 그것의 사상 및 범위 내에 있는, 다수의 시스템 및 방법들을 고안할 수 있을 것이라는 점이 인정될 것이다.

[0352] 부록 A: 두문자어들

[0353] ASIC: Application-Specific Integrated Circuit

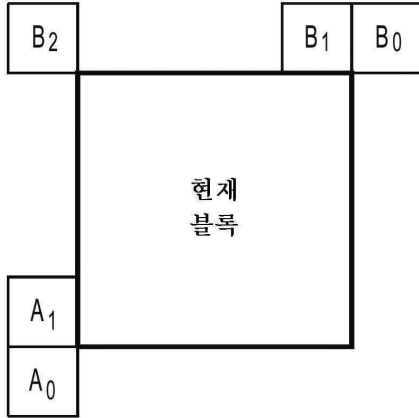
[0354] BMS: Benchmark Set

- [0355] CANBus: Controller Area Network Bus
- [0356] CD: Compact Disc
- [0357] CPUs: Central Processing Units
- [0358] CRT: Cathode Ray Tube
- [0359] CTBs: Coding Tree Blocks
- [0360] CTUs: Coding Tree Units
- [0361] CU: Coding Unit
- [0362] DVD: Digital Video Disc
- [0363] FPGA: Field Programmable Gate Areas
- [0364] GEO: Geometric Merge Mode:
- [0365] GOPs: Groups of Pictures
- [0366] GPUs: Graphics Processing Units
- [0367] GSM: Global System for Mobile communications
- [0368] HDR: High dynamic range
- [0369] HEVC: High Efficiency Video Coding
- [0370] HRD: Hypothetical Reference Decoder
- [0371] IC: Integrated Circuit
- [0372] JEM: Joint Exploration Model
- [0373] JVET: Joint Video Exploration Team
- [0374] LAN: Local Area Network
- [0375] LCD: Liquid-Crystal Display
- [0376] LTE: Long-Term Evolution
- [0377] MV: Motion Vector
- [0378] OLED: Organic Light-Emitting Diode
- [0379] PBs: Prediction Blocks
- [0380] PCI: Peripheral Component Interconnect
- [0381] PLD: Programmable Logic Device
- [0382] PUs: Prediction Units
- [0383] RAM: Random Access Memory
- [0384] ROM: Read-Only Memory
- [0385] SDR: Standard dynamic range
- [0386] SEI: Supplementary Enhancement Information
- [0387] SNR: Signal Noise Ratio
- [0388] SSD: Solid-State Drive
- [0389] TPM: Triangular partitioning mode
- [0390] TUs: Transform Units,

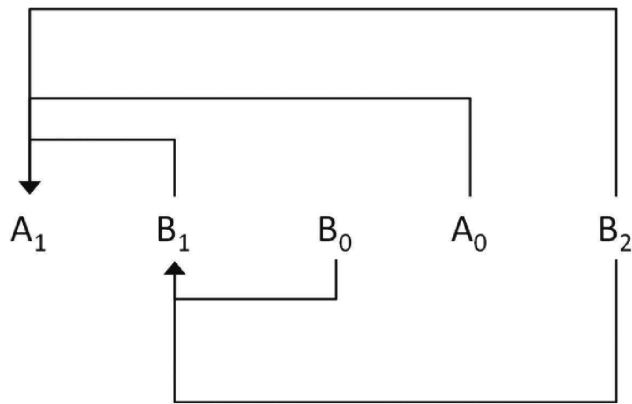
- [0391] USB: Universal Serial Bus
- [0392] VUI: Video Usability Information
- [0393] VVC: Versatile Video Coding

도면

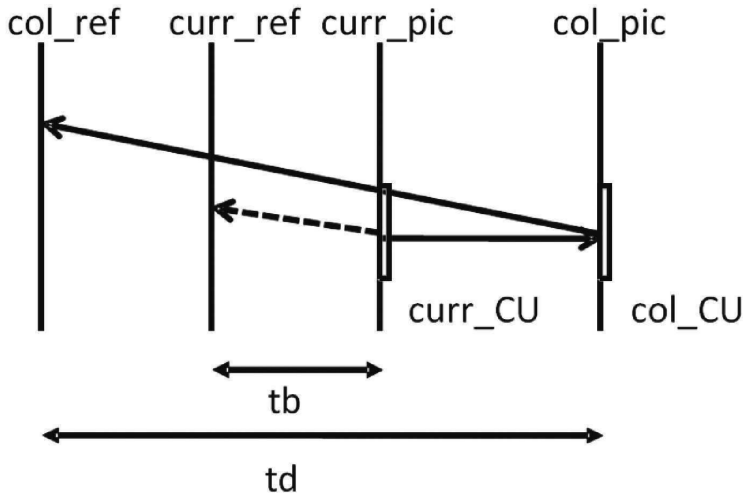
도면1a



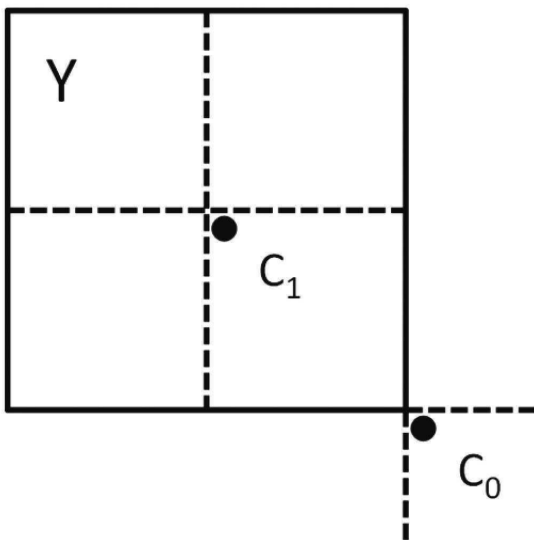
도면1b



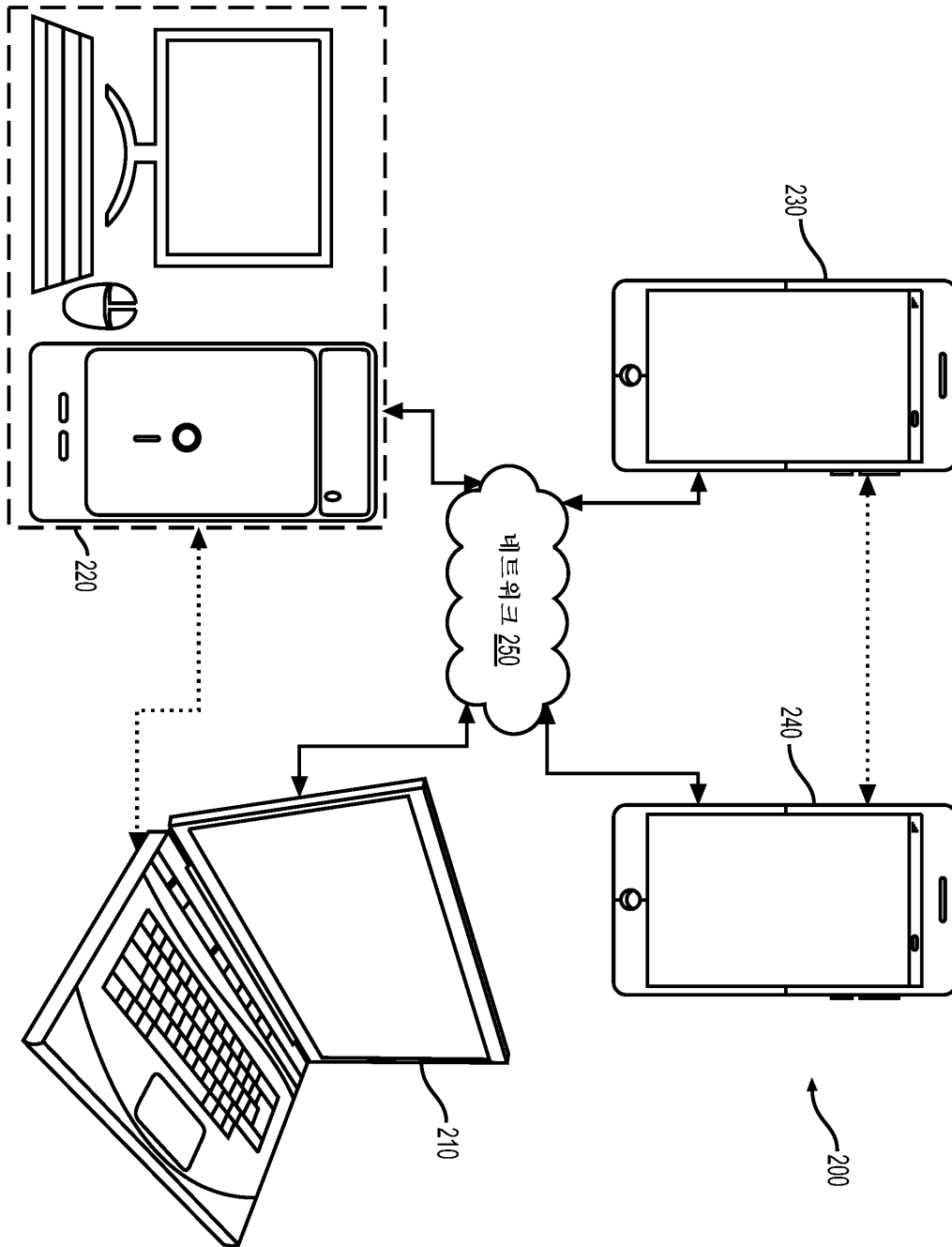
도면1c



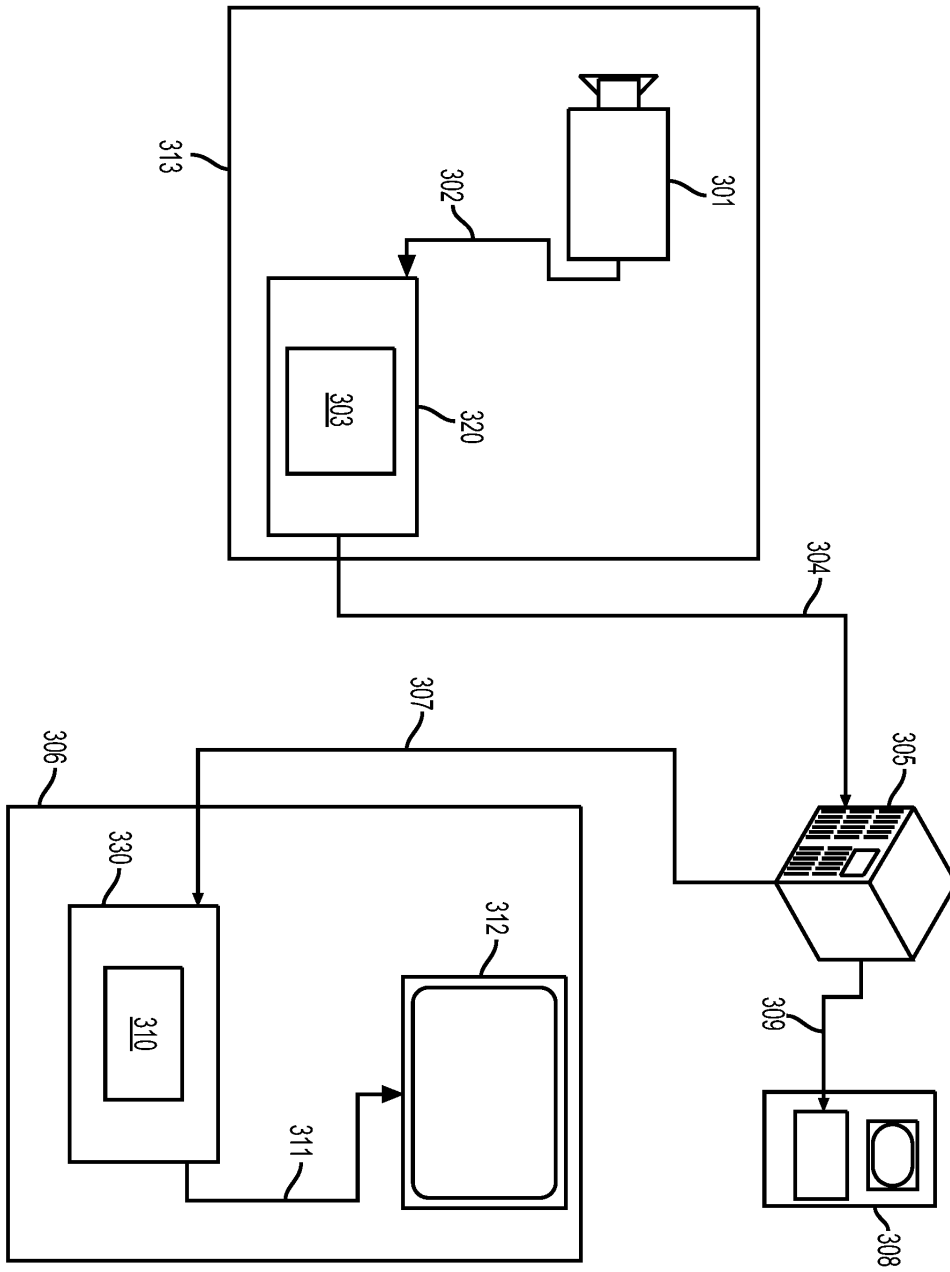
도면1d



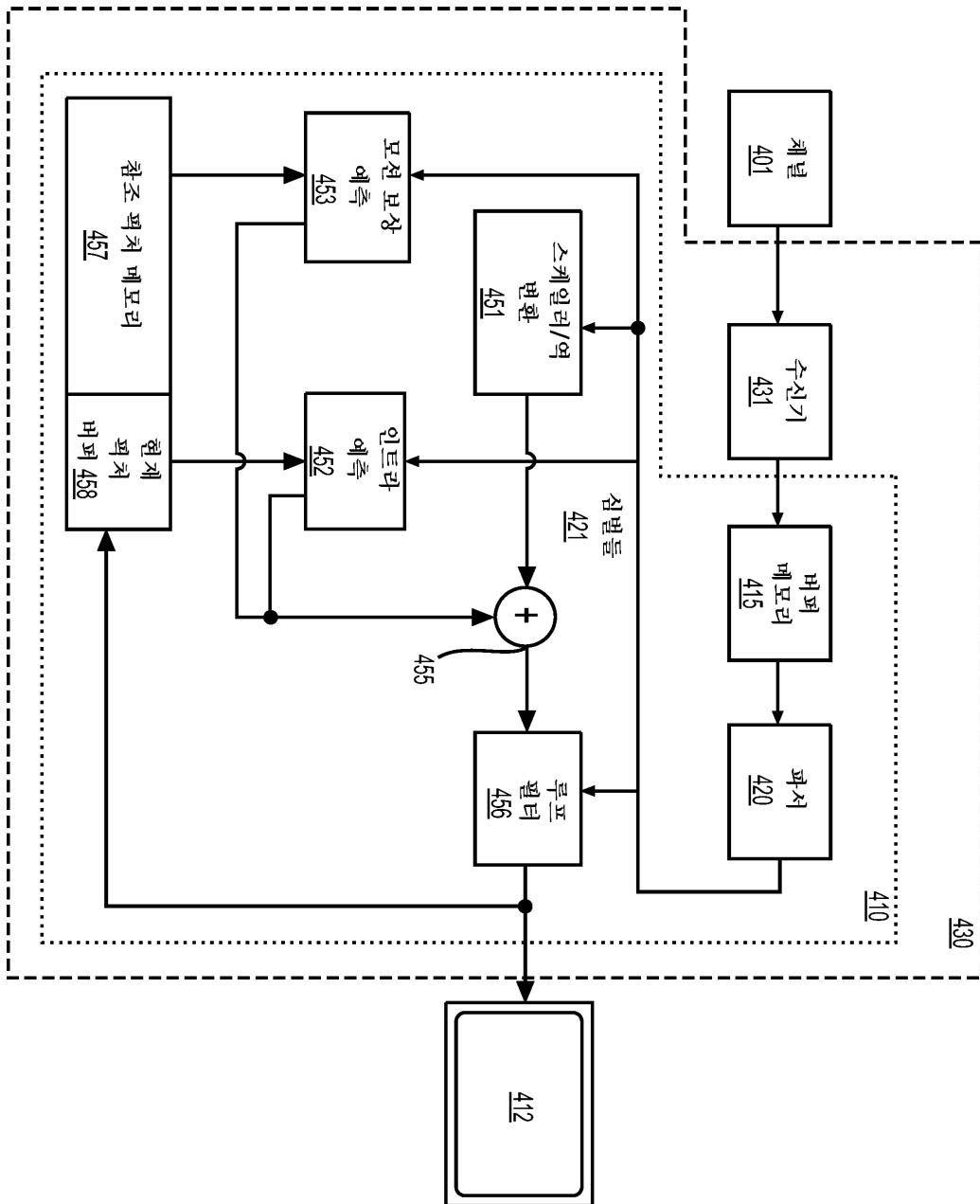
도면2



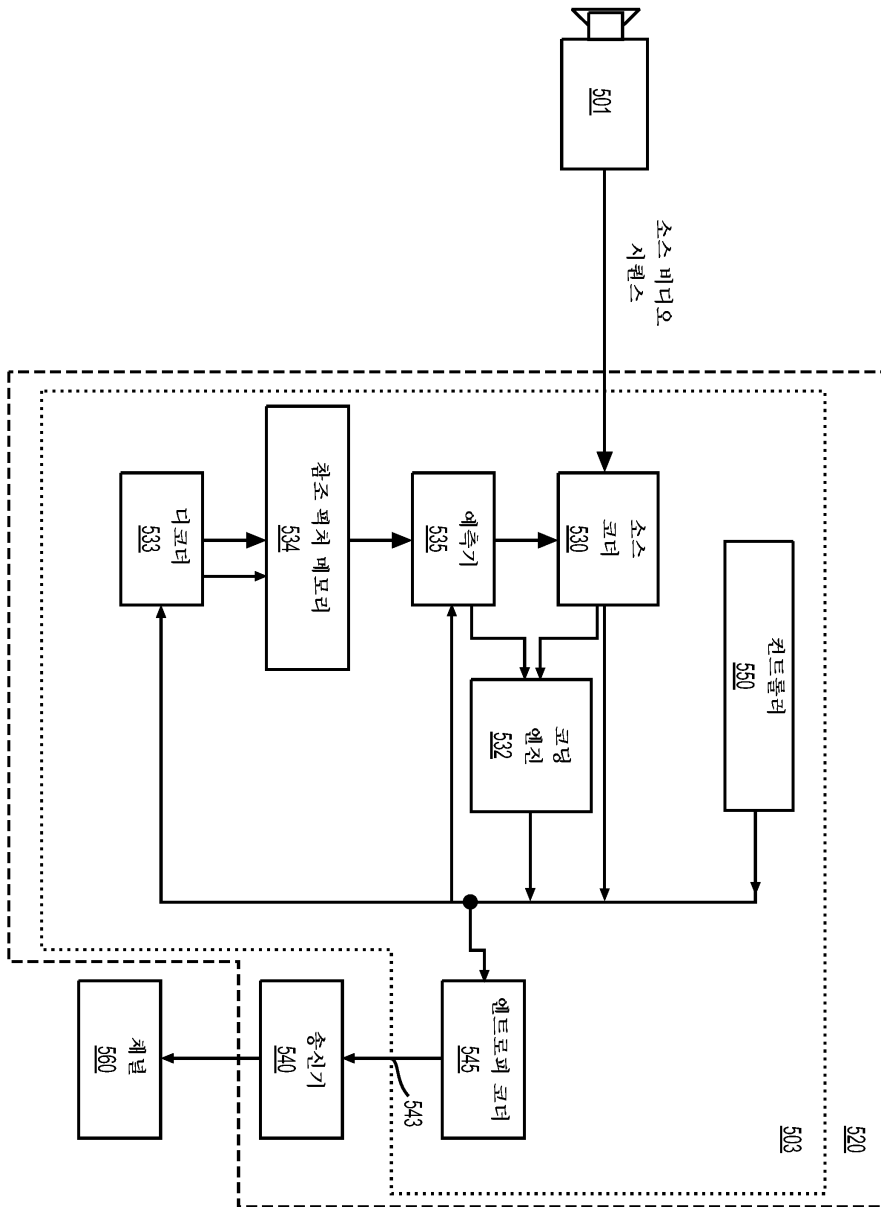
도면3



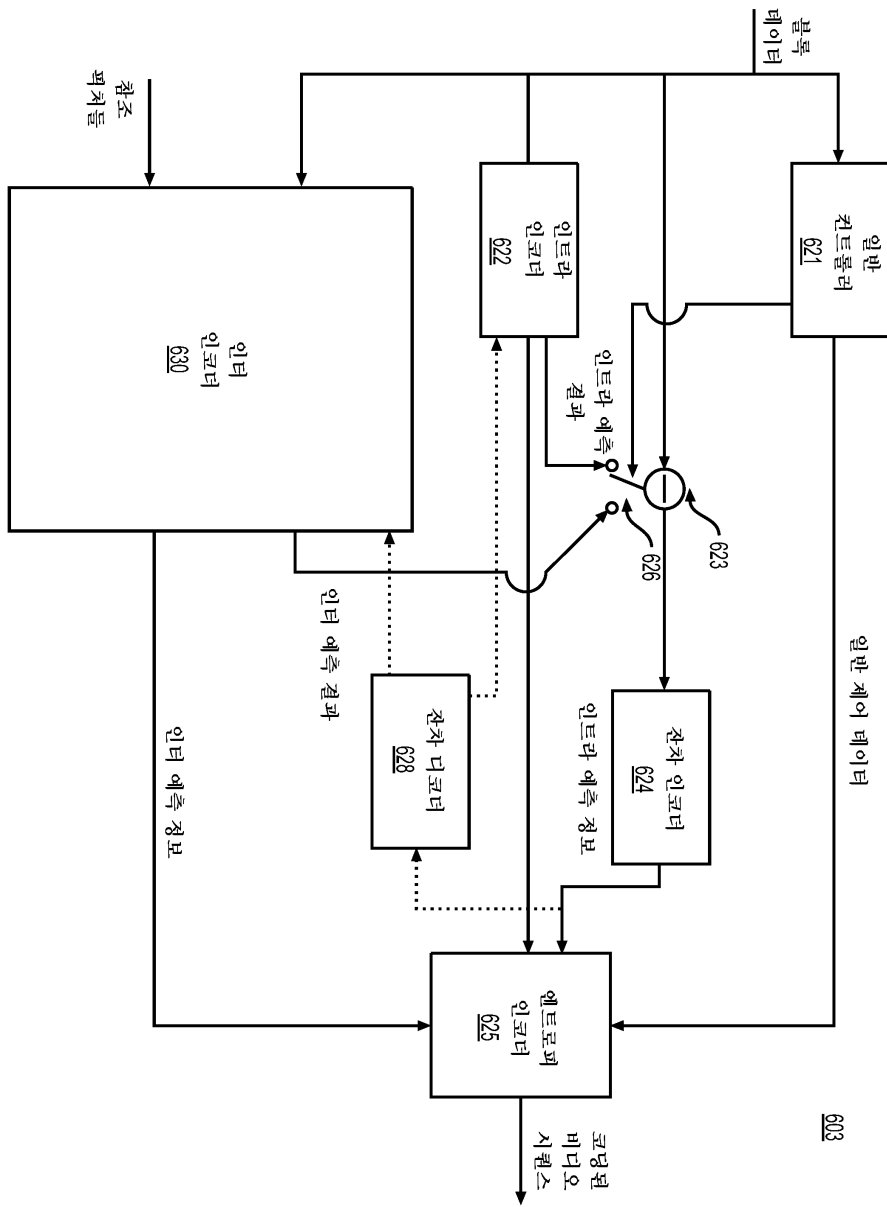
도면4



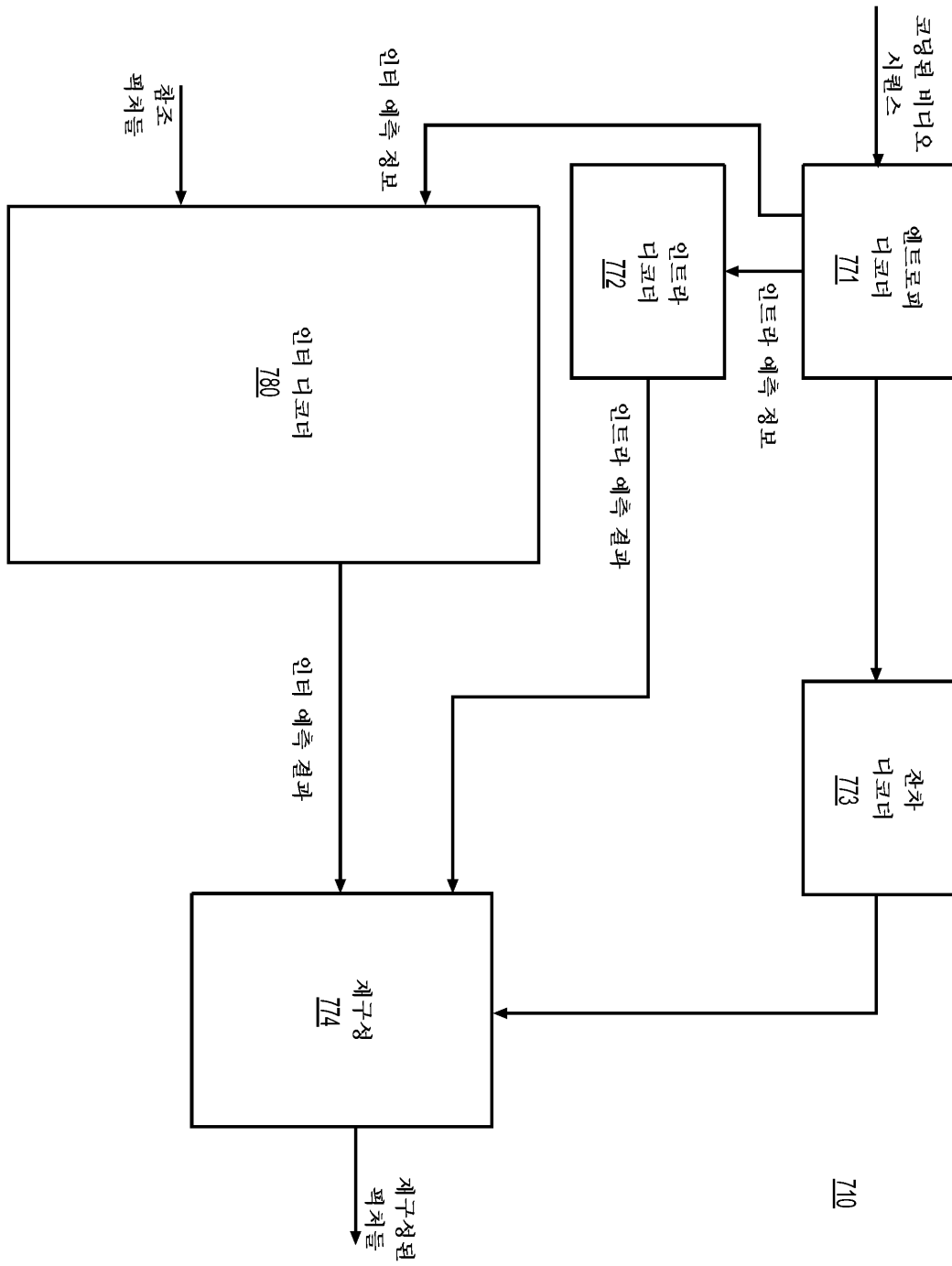
도면5



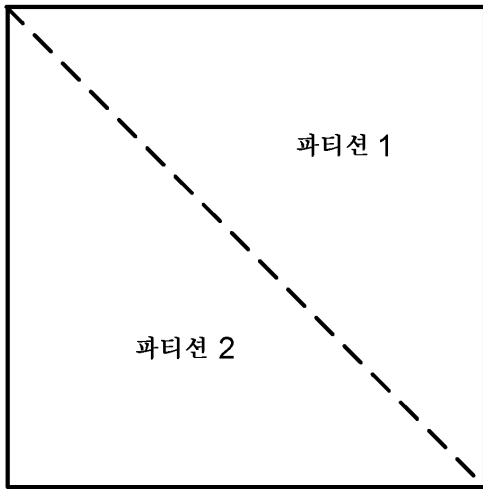
도면6



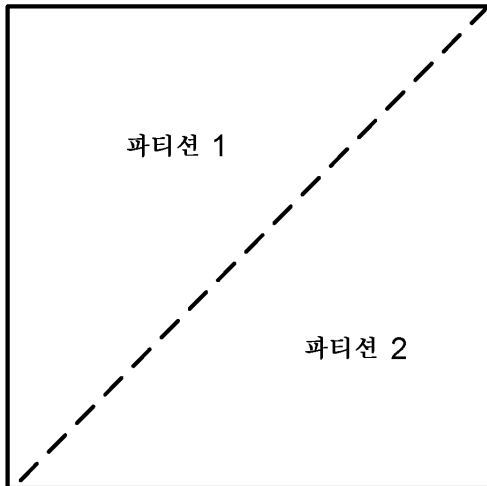
도면7



도면8a



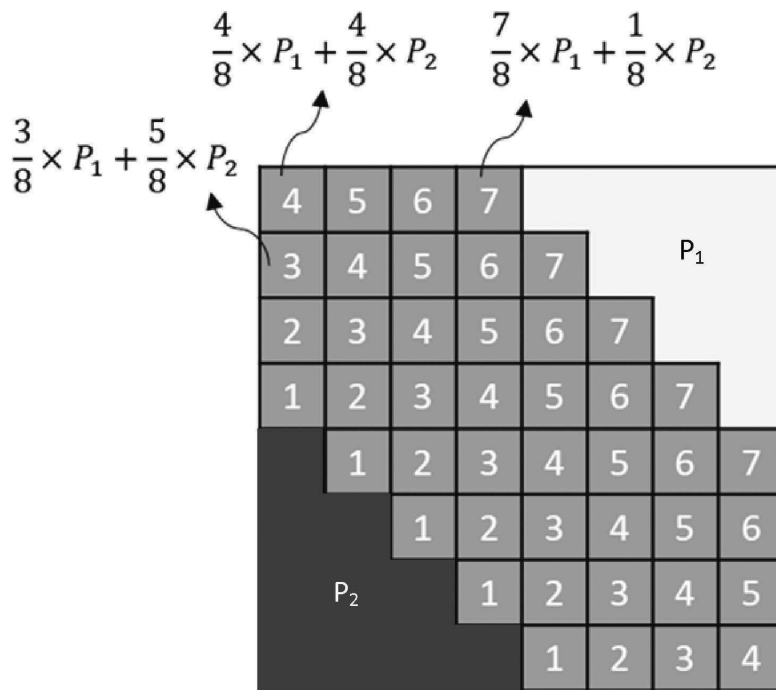
도면8b



도면9

병합 인덱스	L0 MV	L1 MV
0	X	
1		X
2	X	
3		X
4	X	

도면10a



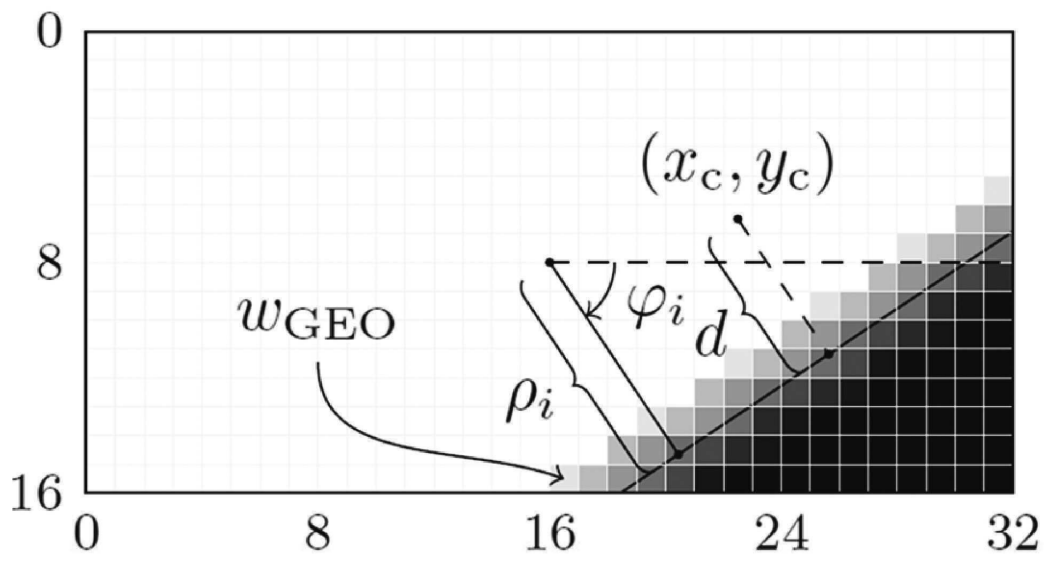
루마에 대한 가중치 맵

도면10b

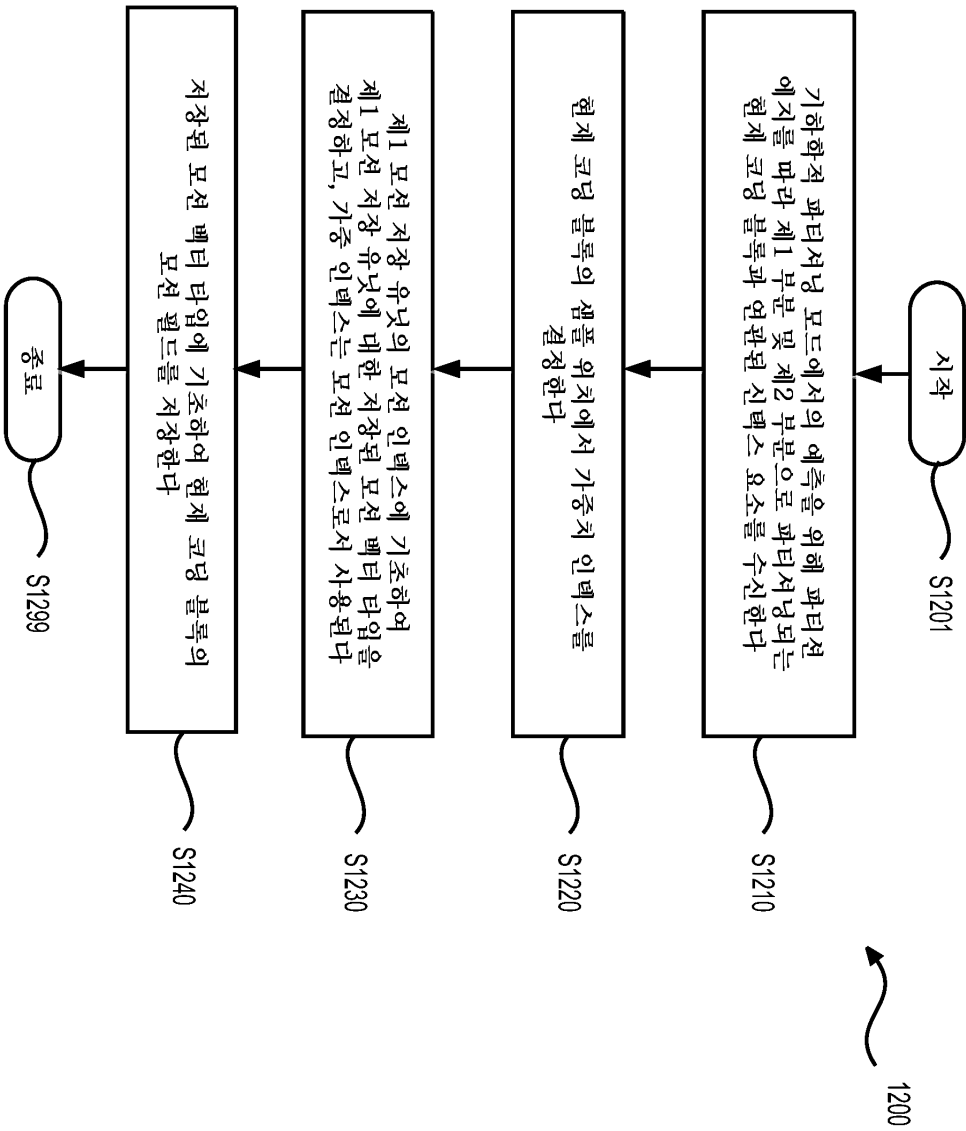
4	6	P <sub>1</sub>	
2	4	6	
	2	4	6
P <sub>2</sub>		2	4

크로마에 대한 가중치 맵

도면11



도면12



도면13

