

(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)(11) 공개번호 10-2022-0165776  
(43) 공개일자 2022년12월15일

- (51) 국제특허분류(Int. Cl.)  
H04N 19/117 (2014.01) H04N 19/132 (2014.01)  
H04N 19/157 (2014.01) H04N 19/176 (2014.01)  
H04N 19/186 (2014.01) H04N 19/70 (2014.01)  
H04N 19/82 (2014.01)
- (52) CPC특허분류  
H04N 19/117 (2015.01)  
H04N 19/132 (2015.01)
- (21) 출원번호 10-2022-7039257
- (22) 출원일자(국제) 2022년01월28일  
심사청구일자 2022년11월09일
- (85) 번역문제출일자 2022년11월09일
- (86) 국제출원번호 PCT/US2022/014255
- (87) 국제공개번호 WO 2022/197375  
국제공개일자 2022년09월22일
- (30) 우선권주장  
63/163,707 2021년03월19일 미국(US)  
17/568,565 2022년01월04일 미국(US)
- (71) 출원인  
텐센트 아메리카 엘엘씨  
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747
- (72) 발명자  
두 이신  
미국 94306 캘리포니아주 팔로 알토 2747 파크 블러바드 텐센트 아메리카 엘엘씨 내
- 차오 신  
미국 95129 캘리포니아주 새너제이 페어포드 웨이 1172
- 리우 산  
미국 95125 캘리포니아주 새너제이 네바다 애비뉴 1155
- (74) 대리인  
유미특허법인

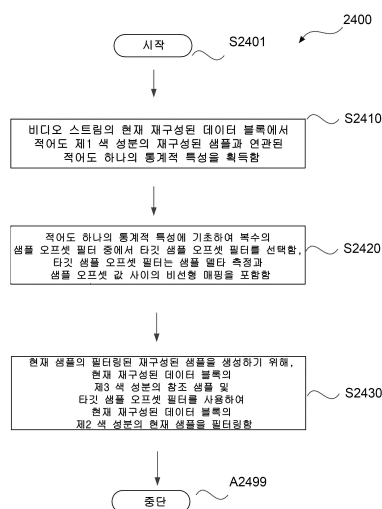
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 샘플 오프셋에 대한 적응적 비선형 매핑

## (57) 요약

비디오 디코더에서 루프 내 샘플 오프셋 필터링을 위한 방법이 개시된다. 이 방법은, 비디오 스트림의 현재 재구성된 데이터 블록에서 적어도 제1 색 성분의 재구성된 샘플과 연관된 적어도 하나의 통계적 특성을 획득하는 단계; 적어도 하나의 통계적 특성에 기초하여 복수의 샘플 오프셋 필터 중에서 타깃 샘플 오프셋 필터를 선택하는 단계 - 타깃 샘플 오프셋 필터는 샘플 델타 측정과 샘플 오프셋 값 사이의 비선형 매핑을 포함함 -; 및 현재 샘플의 필터링된 재구성된 샘플을 생성하기 위해, 현재 재구성된 데이터 블록의 제3 색 성분의 참조 샘플 및 타깃 샘플 오프셋 필터를 사용하여 현재 재구성된 데이터 블록의 제2 색 성분의 현재 샘플을 필터링하는 단계를 포함한다.

대표도 - 도24



(52) CPC특허분류

*H04N 19/157* (2015.01)

*H04N 19/176* (2015.01)

*H04N 19/186* (2015.01)

*H04N 19/70* (2015.01)

*H04N 19/82* (2015.01)

---

## 명세서

### 청구범위

#### 청구항 1

비디오 스트림(video stream)의 루프 내 필터링(in-loop filtering)을 위한 방법으로서,

상기 비디오 스트림의 현재 재구성된 데이터 블록에서 적어도 제1 색 성분의 재구성된 샘플과 연관된 적어도 하나의 통계적 특성을 획득하는 단계;

상기 적어도 하나의 통계적 특성에 기초하여 복수의 샘플 오프셋 필터 중에서 타깃 샘플 오프셋 필터를 선택하는 단계 - 상기 타깃 샘플 오프셋 필터는 샘플 델타(delta) 측정과 샘플 오프셋 값 사이의 비선형 매핑을 포함함 -; 및

상기 현재 재구성된 데이터 블록의 제3 색 성분의 참조 샘플 및 상기 타깃 샘플 오프셋 필터를 사용하여 상기 현재 재구성된 데이터 블록의 제2 색 성분의 현재 샘플을 필터링하여, 상기 현재 샘플의 필터링된 재구성된 샘플을 생성하는 단계를 포함하는,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 2

제1항에 있어서,

상기 제1 색 성분은 상기 제3 색 성분과 동일한 색 성분인,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 3

제1항에 있어서,

제1 색 성분은 상기 제2 색 성분과 동일한 색 성분인,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 4

제1항에 있어서,

상기 제2 색 성분은 상기 제3 색 성분과 다른 색 성분인,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 5

제1항에 있어서,

상기 제2 색 성분은 상기 제3 색 성분과 동일한 색 성분인,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 6

제1항에 있어서,

적어도 상기 제1 색 성분은 1개, 2개 또는 3개의 색 성분을 포함하는,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 7

제1항에 있어서,

상기 적어도 하나의 통계적 특성은 상기 현재 재구성된 데이터 블록의 에지 정보를 포함하는,  
비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 8

제1항 내지 제7항 중 어느 한 항에 있어서,

상기 현재 재구성된 데이터 블록의 에지 정보는 제한된 방향 향상 필터링(CDEF: Constrained Directional Enhanced Filtering) 프로세스에서 도출된 에지 방향을 포함하고; 그리고

상기 복수의 샘플 오프셋 필터는 N개의 CDEF 에지 방향에 대응하는 N개의 샘플 오프셋 필터를 포함하며,

N은 1부터 8까지의 정수인,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 9

제1항 내지 제7항 중 어느 한 항에 있어서,

상기 복수의 샘플 오프셋 필터는 고레벨 선택스(HLS: high level syntax)의 프레임 레벨에서 시그널링되는,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 10

제1항 내지 제7항 중 어느 한 항에 있어서,

상기 적어도 하나의 통계적 특성은 상기 현재 재구성된 데이터 블록의 평활도 측정을 포함하는,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 11

제10항에 있어서,

상기 현재 재구성된 데이터 블록의 평활도 측정은 M-1개의 평활도 레벨 임계치를 특징으로 하는 M개의 미리 정의된 평활도 레벨 중 하나에 매핑되고;

상기 복수의 샘플 오프셋 필터는 상기 M개의 미리 정의된 평활도 레벨에 대응하는 M개의 샘플 오프셋 필터를 포함하고; 그리고

상기 타깃 샘플 오프셋 필터는 상기 평활도 측정에 매핑된 M개의 미리 정의된 평활도 레벨 중 하나에 따라 상기 M개의 샘플 오프셋 필터 중에서 선택되는,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 12

제1항 내지 제7항 중 어느 한 항에 있어서,

상기 적어도 하나의 통계적 특성은 상기 현재 재구성된 데이터 블록의 코딩 정보를 포함하는,

비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 13

제12항에 있어서,

코딩 정보는 상기 현재 재구성된 데이터 블록의 현재 예측 모드를 포함하고;

상기 복수의 샘플 오프셋 필터는 상이한 예측 모드에 대응하고; 그리고

상기 타깃 샘플 오프셋 필터는 상기 현재 재구성된 데이터 블록의 현재 예측 모드에 따라 상기 복수의 샘플 오



프셋 필터 중에서 선택되는,  
비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 14

제13항에 있어서,  
상기 상이한 예측 모드는 인트라(intra) DC 모드, 인트라 평면(Planar) 모드, 인트라 PAETH 모드, 인트라 SMOOTH 모드, 인트라 채귀 필터링 모드 및 인트라 SKIP 모드 중 적어도 하나를 포함하는,  
비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 15

제1항 내지 제7항 중 어느 한 항에 있어서,  
상기 복수의 샘플 오프셋 필터 각각은 한 세트의 필터 계수, 다수의 필터 탭(tap), 및 상기 다수의 필터 탭의 포지션과 연관되는,  
비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 16

제1항 내지 제7항 중 어느 한 항에 있어서,  
상기 현재 재구성된 데이터 블록의 제3 색 성분의 참조 샘플 및 상기 타깃 샘플 오프셋 필터를 사용하여 상기 현재 재구성된 데이터 블록의 제2 색 성분의 현재 샘플을 필터링하여, 상기 현재 샘플의 필터링된 재구성된 샘플을 생성하는 단계는:  
상기 제2 색 성분의 현재 샘플의 제1 위치 및 상기 타깃 샘플 오프셋 필터와 연관된 복수의 필터 탭의 제2 위치를 결정하는 단계;  
상기 제1 위치 및 상기 제2 위치에서 상기 제3 색 성분의 재구성된 샘플을 상기 참조 샘플로서 식별하는 단계;  
상기 현재 재구성된 데이터 블록의 제3 색 성분 모두에서 상기 제2 위치에 대응하는 참조 샘플과 상기 제1 위치에 대응하는 참조 샘플 사이의 델타 측정을 결정하는 단계;  
상기 델타 측정에 기초하여 상기 타깃 샘플 오프셋 필터로부터 샘플 오프셋 값을 추출하는 단계; 및  
상기 현재 샘플의 필터링된 재구성된 샘플을 생성하기 위해, 상기 샘플 오프셋 값을 사용하여 상기 제2 색 성분의 상기 현재 샘플을 필터링하는 단계를 포함하는,  
비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 17

제1항 내지 제7항 중 어느 한 항에 있어서,  
상기 복수의 샘플 오프셋 필터는 미리 결정되고;  
상기 복수의 샘플 오프셋 필터 및 상기 복수의 샘플 오프셋 필터 중에서 선택된 타깃 샘플 오프셋 필터의 인덱스는 시퀀스 레벨, 픽처 레벨 또는 코딩 트리 유닛(coding tree unit) 레벨에서 시그널링되는,  
비디오 스트림의 루프 내 필터링을 위한 방법.

#### 청구항 18

비디오 스트림의 루프 내 필터링을 위한 장치로서,  
처리 회로를 포함하며,  
상기 처리 회로는:  
상기 비디오 스트림의 현재 재구성된 데이터 블록에서 적어도 제1 색 성분의 재구성된 샘플과 연관된 적어도 하

나의 통계적 특성을 획득하고;

상기 적어도 하나의 통계적 특성에 기초하여 복수의 샘플 오프셋 필터 중에서 타깃 샘플 오프셋 필터를 선택하고 - 상기 타깃 샘플 오프셋 필터는 샘플 델타 측정과 샘플 오프셋 값 사이의 비선형 매핑을 포함함 -; 그리고

상기 현재 재구성된 데이터 블록의 제3 색 성분의 참조 샘플 및 상기 타깃 샘플 오프셋 필터를 사용하여 상기 현재 재구성된 데이터 블록의 제2 색 성분의 현재 샘플을 필터링하여, 상기 현재 샘플의 필터링된 재구성된 샘플을 생성하도록 구성되는,

비디오 스트림의 루프 내 필터링을 위한 장치.

## 청구항 19

제2항 내지 제7항 중 어느 한 항을 수행하도록 구성된 처리 회로를 포함하는,

비디오 스트림의 루프 내 필터링을 위한 장치.

## 청구항 20

컴퓨터 명령을 저장하기 위한 비-일시적 컴퓨터 판독 가능 매체로서,

상기 컴퓨터 명령은, 프로세서에 의해 실행될 때 상기 프로세서로 하여금, 제1항 내지 제7항 중 어느 한 항을 수행하게 하는,

비-일시적 컴퓨터 판독 가능 매체.

## 발명의 설명

### 기술 분야

[0001] 본 개시내용은 2022년 1월 4일자 출원된 미국 정규특허출원 제17/568,565호를 기반으로 하며 이에 대한 우선권의 이익을 주장하는데, 이 정규특허출원은 "ADAPTIVE NON-LINEAR MAPPING FOR SAMPLE OFFSET"이라는 명칭으로 2021년 3월 19일자 출원된 미국 가출원 제63/163,707호에 대해 주장한다. 두 출원 모두 그 전체가 인용에 의해 본 명세서에 포함된다.

[0002] 본 개시내용은 일반적으로 한 세트의 고급 비디오 코딩(advanced video coding) 기술을 설명하며, 구체적으로는 로컬 적응성을 갖는 샘플 오프셋 필터링에 관한 것이다.

### 배경 기술

[0003] 본 명세서에서 제공되는 이러한 배경 설명은 일반적으로 본 개시내용의 맥락을 제시하는 것을 목적으로 한다. 현재 지명된 발명자의 작업은 이 작업이 이 배경 섹션뿐만 아니라, 본 출원의 출원 시에 종래 기술로서 달리 한정할 수 없는 설명의 양상에서 설명되는 한도까지, 명확하게도 묵시적으로도 본 개시내용에 대한 종래 기술로서 인정되지 않는다.

[0004] 비디오 코딩 및 디코딩은 움직임 보상에 의한 인터 픽처(inter-picture) 예측을 사용하여 수행될 수 있다. 압축되지 않은 디지털 비디오는 일련의 픽처를 포함할 수 있으며, 각각의 픽처는 예를 들어, 1920×1080 휘도 샘플 및 연관된 전체 또는 서브샘플링된 색차 샘플의 공간 차원을 갖는다. 일련의 픽처는 예를 들어, 초당 60개의 픽처 또는 초당 60개의 프레임의 (대안으로 프레임 레이트로 지칭되는) 고정 또는 가변 픽처 레이트를 가질 수 있다. 압축되지 않은 비디오는 스트리밍 또는 데이터 처리를 위한 특정 비트레이트 요건을 갖는다. 예를 들어, 1920×1080의 픽셀 해상도, 60프레임/초의 프레임 레이트, 및 색 채널별 픽셀당 8비트에서 4:2:0의 크로마 서브샘플링(chroma subsampling)을 사용하는 비디오는 1.5Gbit/s에 가까운 대역폭을 필요로 한다. 이러한 비디오 1시간은 600기가바이트(Gbytes)를 넘는 저장 공간을 필요로 한다.

[0005] 비디오 코딩 및 디코딩의 한 가지 목적은 압축을 통한 비압축 입력 비디오 신호의 중복성 감소일 수 있다. 압축은 앞서 언급한 대역폭 및/또는 저장 공간 요건을 줄이는 데, 일부 경우에는 100배 이상 줄이는 데 도움이 될 수 있다. 무손실 압축 및 손실 압축뿐만 아니라 이들의 조합이 모두 이용될 수 있다. 무손실 압축은 디코딩 프로세스를 통해, 압축된 원본 신호로부터 원본 신호의 정확한 사본이 재구성될 수 있는 기법을 의미한다. 손실 압축은 원본 비디오 정보가 코딩 중에 완전히 유지되지 않고 디코딩 중에 완전히 복구 가능하지는 않는 코딩

/디코딩 프로세스를 의미한다. 손실 압축을 사용할 때, 재구성된 신호는 원본 신호와 동일하지 않을 수 있지만, 원본 신호와 재구성된 신호 사이의 왜곡은 약간의 정보 손실에도 불구하고, 재구성된 신호를 의도된 애플리케이션에 유용하게 하기에 충분히 작게 된다. 비디오의 경우, 손실 압축이 많은 애플리케이션에서 널리 이용된다. 허용 가능한 왜곡의 양은 애플리케이션에 좌우된다. 예를 들어, 특정 소비자 비디오 스트리밍 애플리케이션의 사용자는 영화나 텔레비전 방송 애플리케이션의 사용자보다 더 높은 왜곡을 허용할 수 있다. 특정 코딩 알고리즘에 의해 달성 가능한 압축비는 다양한 왜곡 허용도를 반영하도록 선택 또는 조정될 수 있는데: 더 높은 허용 가능한 왜곡은 일반적으로 더 높은 손실과 더 높은 압축비를 산출하는 코딩 알고리즘을 허용한다.

[0006] 비디오 인코더(encoder) 및 디코더(decoder)는 예를 들어 움직임 보상, 푸리에(Fourier) 변환, 양자화 및 엔트로피(entropy) 코딩을 포함하는 여러 광범위한 카테고리 및 단계로부터의 기법을 이용할 수 있다.

[0007] 비디오 코덱(codec) 기술은 인트라 코딩(intra coding)으로 알려진 기법을 포함할 수 있다. 인트라 코딩에서, 샘플 값은 이전에 재구성된 참조 픽처로부터의 샘플 또는 다른 데이터에 대한 참조 없이 표현된다. 일부 비디오 코덱에서, 픽처는 샘플의 블록으로 공간적으로 세분된다. 샘플의 모든 블록이 인트라 모드에서 코딩될 때, 그 픽처는 인트라 픽처로 지칭될 수 있다. 독립적 디코더 리프레시 픽처와 같은 인트라 픽처 및 이들의 파생물은 디코더 상태를 재설정하는 데 사용될 수 있고, 따라서 코딩된 비디오 비트스트림(coded video bitstream) 및 비디오 세션(session)의 첫 번째 픽처로서 또는 정지 이미지로서 사용될 수 있다. 이어서, 인트라 예측 후 블록의 샘플은 주파수 도메인으로 변환될 수 있고, 그렇게 생성된 변환 계수는 엔트로피 코딩 전에 양자화될 수 있다. 인트라 예측은 변환 전 도메인(pre-transform domain)에서 샘플 값을 최소화하는 기법을 나타낸다. 일부 경우에는, 변환 후 DC 값이 더 작을수록, 그리고 AC 계수가 더 작을수록, 엔트로피 코딩 후 블록을 나타내기 위해 주어진 양자화 단계 크기에 필요한 비트가 더 적다.

[0008] 예를 들어, MPEG-2 생성 코딩 기술로부터 알려진 것과 같은 종래의 인트라 코딩은 인트라 예측을 사용하지 않는다. 그러나 일부 더 새로운 비디오 압축 기술은 예를 들어, 공간적으로 이웃하는 인코딩 및/또는 디코딩 중에 획득되고 디코딩 순서가 선행하는 주변 샘플 데이터 및/또는 메타데이터를 기반으로 블록의 코딩/디코딩을 시도하는 기법을 포함한다. 이러한 기법은 이후 "인트라 예측" 기법이라 한다. 적어도 일부 경우에, 인트라 예측은 다른 참조 픽처가 아닌 재구성 중인 현재 픽처로부터의 참조 데이터만을 사용한다는 점에 주목한다.

[0009] 많은 다양한 형태의 인트라 예측이 있을 수 있다. 이러한 기법 중 하나보다 많은 기법이 주어진 비디오 코딩 기술에서 이용 가능한 경우, 사용 중인 기법은 인트라 예측 모드로 지칭될 수 있다. 특정 코덱에는 하나 이상의 인트라 예측 모드가 제공될 수 있다. 특정 경우에, 모드는 하위 모드를 가질 수 있고 그리고/또는 다양한 파라미터와 연관될 수 있으며, 비디오 블록에 대한 모드/하위 모드 정보 및 인트라 코딩 파라미터는 모드 코드워드에 개별적으로 또는 집합적으로 코딩되어 포함될 수 있다. 주어진 모드, 하위 모드 및/또는 파라미터 조합에 어떤 코드워드를 사용할지는 인트라 예측을 통한 코딩 효율 이득에 영향을 미칠 수 있으며, 코드워드를 비트스트림으로 변환하는 데 사용되는 엔트로피 코딩 기술도 그러할 수 있다.

[0010] 특정 모드의 인트라 예측은 H.264로 도입되었으며, H.265로 개정되었고, 공동 탐사 모델(JEM: joint exploration model), 다용도 비디오 코딩(VVC: versatile video coding) 및 벤치마크 세트(BMS: benchmark set)와 같은 더 최신 코딩 기술에서 더 개정되었다. 일반적으로 인트라 예측의 경우, 이용 가능하게 된 이웃 샘플 값을 사용하여 예측자 블록이 형성될 수 있다. 예를 들어, 특정 방향 및/또는 라인을 따라 특정 세트의 이웃 샘플의 이용 가능한 값이 예측자 블록에 복사될 수 있다. 사용 중인 방향에 대한 참조는 비트스트림으로 코딩될 수 있거나 자체적으로 예측될 수 있다.

[0011] 도 1a를 참조하면, 오른쪽 최하부에는 (H.265에 명시된 35개의 인트라 모드의 33개의 각도 모드에 대응하는) H.265의 33개의 가능한 인트라 예측자 방향에 명시된 9개의 예측자 방향의 서브세트가 도시된다. 화살표가 수렴하는 지점(101)은 예측되고 있는 샘플을 나타낸다. 화살표는 이웃 샘플이 101에서 샘플을 예측하는 데 사용되는 방향을 나타낸다. 예를 들어, 화살표(102)는 샘플(101)이 수평 방향에서 45도 각도로 오른쪽 상부에 있는 이웃 샘플 또는 샘플들로부터 예측된다는 것을 지시한다. 유사하게, 화살표(103)는 샘플(101)이 수평 방향에서 22.5도 각도로 샘플(101)의 왼쪽 아래에 있는 이웃 샘플 또는 샘플들로부터 예측된다는 것을 지시한다.

[0012] 도 1a를 참조하면, 왼쪽 최상부에 (볼트체 파선으로 지시된) 4×4 샘플의 정사각형 블록(104)이 도시되어 있다. 정사각형 블록(104)은 각각 "S", Y 차원에서의 포지션(예컨대, 행 인덱스) 및 X 차원에서의 포지션(예컨대, 열 인덱스)으로 표기된 16개의 샘플을 포함한다. 예를 들어, 샘플 S21은 Y 차원에서 (위에서) 두 번째 샘플이고 X 차원에서 (왼쪽에서) 첫 번째 샘플이다. 유사하게, 샘플 S44는 Y 차원 및 X 차원 모두에서 블록(104) 내의 네 번째 샘플이다. 블록은 크기가 4×4 샘플이므로, S44는 오른쪽 하단에 있다. 유사한 번호 매기기 체계를 따라

는 예시적인 참조 샘플이 추가로 도시된다. 참조 샘플은 블록(104)에 대해 R, Y 포지션(예컨대, 행 인덱스) 및 X 포지션(열 인덱스)으로 표기된다. H.264 및 H.265 모두에서, 재구성 중인 블록에 인접하게 이웃하는 예측 샘플이 사용된다.

- [0013] 블록(104)의 인트라 픽처 예측은 시그널링된 예측 방향에 따라 이웃 샘플로부터 참조 샘플 값을 복사함으로써 시작될 수 있다. 예를 들어, 코딩된 비디오 비트스트림이 이 블록(104)에 대해 화살표(102)의 예측 방향을 지시하는 시그널링을 포함한다고 가정하면—즉, 수평 방향에서 45도 각도로 샘플 또는 샘플들로부터 오른쪽 상부까지 샘플이 예측된다. 이러한 경우, 동일한 참조 샘플 R05로부터 샘플 S41, S32, S23 및 S14가 예측된다. 그런 다음, 참조 샘플 R08로부터 샘플 S44가 예측된다.
- [0014] 특정 경우에, 참조 샘플을 계산하기 위해; 특히 방향이 45도로 균등하게 나누어지지 않을 때, 예를 들어 보간을 통해 다수의 참조 샘플의 값이 조합될 수 있다.
- [0015] 비디오 코딩 기술이 계속 발달함에 따라 가능한 방향의 수가 증가했다. 예를 들어, H.264(2003년)에서는, 인트라 예측에 9개의 상이한 방향이 이용 가능하다. H.265(2013년)에서는 그것이 33으로 증가했고, 이 공개 당시 JEM/VVC/BMS는 최대 65개의 방향을 지원할 수 있다. 가장 적합한 인트라 예측 방향을 식별하는 데 도움이 되도록 실험적 연구가 수행되었으며, 방향에 대해 특정 비트 페널티를 수용하여, 소수의 비트에서 그러한 가장 적합한 방향을 인코딩하는 데 엔트로피 코딩의 특정 기법이 사용될 수 있다. 또한, 방향 자체는 간혹, 디코딩된 이웃 블록의 인트라 예측에 사용된 이웃 방향으로부터 예측될 수 있다.
- [0016] 도 1b는 시간 경과에 따라 개발된 다양한 인코딩 기술에서 증가하는 수의 예측 방향을 예시하기 위해 JEM에 따른 65개의 인트라 예측 방향을 도시하는 개략도(180)를 도시한다.
- [0017] 코딩된 비디오 비트스트림의 인트라 예측 유닛의 예측 매핑은 인트라 예측 방향을 나타내는 비트를 코딩된 비디오 비트스트림의 예측 방향에 매핑하기 위한 방식은 비디오 코딩 기술마다 달라질 수 있으며; 예를 들어, 예측 방향의 단순한 직접 매핑에서부터 인트라 예측 모드, 코드워드, 가장 가능성 있는 모드를 수반하는 복잡한 적응 방식 및 유사한 기술에 이르기까지 다양할 수 있다. 그러나 모든 경우에, 통계적으로 다른 특정 방향보다 비디오 콘텐츠에서 발생할 가능성이 적은, 인트라 예측에 대한 특정 방향이 있을 수 있다. 비디오 압축의 목표는 중복성의 감소이므로, 잘 설계된 비디오 코딩 기술에서 가능성이 더 적은 그러한 방향은 가능성이 더 높은 방향보다 더 많은 수의 비트로 표현될 수 있다.
- [0018] 인트라 픽처 예측 또는 인트라 예측은 움직임 보상에 기반할 수 있다. 움직임 보상에서, 움직임 벡터(이후 MV)로 지시되는 방향으로 공간적으로 시프트된 후, 이전에 재구성된 픽처 또는 그 일부(참조 픽처)로부터의 샘플 데이터는 새로 재구성된 픽처 또는 픽처 부분(예컨대, 블록)의 예측에 사용될 수 있다. 일부 경우에는, 참조 픽처가 현재 재구성 중인 픽처와 동일할 수 있다. MV는 X 차원과 Y 차원인 2차원 또는 3차원을 가질 수 있으며, 세 번째 차원은 (시간 차원과 유사한) 사용 중인 참조 픽처의 지시이다.
- [0019] 일부 비디오 압축 기법에서, 샘플 데이터의 특정 영역에 적용 가능한 현재 MV는 다른 MV로부터, 예를 들어 재구성 중인 영역에 공간적으로 인접하고 디코딩 순서에서 현재 MV에 선행하는 샘플 데이터의 다른 영역과 관련된 그러한 다른 MV로부터 예측될 수 있다. 그렇게 하면, 상관된 MV의 중복성 제거에 의존함으로써 MV를 코딩하는 데 필요한 데이터의 전체 양을 상당히 줄일 수 있으며, 이로써 압축 효율을 높일 수 있다. 예를 들어, MV 예측은 효과적으로 작동할 수 있는데, 이는 카메라로부터 도출된 (본래의 비디오로 알려진) 입력 비디오 신호를 코딩할 때 단일 MV가 적용 가능한 영역보다 더 큰 영역이 비디오 시퀀스에서 유사한 방향으로 이동하고 따라서 일부 경우에는, 인접 영역의 MV에서 도출된 유사한 움직임 벡터를 사용하여 예측될 수 있는 통계적 가능성이 있기 때문이다. 그 결과, 주어진 영역에 대한 실제 MV가 주변 MV로부터 예측된 MV와 유사하거나 동일하게 된다. 그러한 MV는 결국, 엔트로피 코딩 후에, MV가 이웃 MV(들)로부터 예측되기 보다는 직접 코딩되는 경우에 사용되는 것보다 더 적은 수의 비트로 표현될 수 있다. 일부 경우에, MV 예측은 원본 신호(즉: 샘플 스트림)에서 도출된 신호(즉: MV)의 무손실 압축의 일례일 수 있다. 다른 경우에, 예를 들어, 여러 주변 MV로부터 예측자를 계산할 때 반올림 오류로 인해 MV 예측 자체는 손실이 있을 수 있다.
- [0020] H.265/HEVC(ITU-T Rec. H.265, "High Efficiency Video Coding", 2016년 12월)에서 다양한 MV 예측 메커니즘이 설명된다. H.265가 특징하는 많은 MV 예측 메커니즘 중에서, 아래에는 이후 "공간 병합"으로 지칭되는 기법이 설명된다.
- [0021] 구체적으로, 도 2를 참조하면, 현재 블록(201)은 공간적으로 시프트된 동일한 크기의 이전 블록으로부터 예측 가능하도록 움직임 탐색 프로세스 동안 인코더에 의해 발견된 샘플을 포함한다. 해당 MV를 직접 코딩하는 대신

에, MV는 A0, A1 및 B0, B1, B2(각각 202 내지 206)로 표시된 5개의 주변 샘플 중 하나와 연관된 MV를 사용하여 하나 이상의 참조 픽처와 연관된 메타데이터로부터, 예를 들어 (디코딩 순서에서) 가장 최근의 참조 픽처로부터 도출될 수 있다. H.265에서, MV 예측은 이웃 블록이 사용하는 동일한 참조 픽처의 예측자를 사용할 수 있다.

## 발명의 내용

- [0022] 본 개시내용의 양상은 비디오 인코딩 및 디코딩에서 샘플 간 오프셋 필터링 및 로컬 샘플 오프셋 필터링을 위한 방법 및 장치를 제공한다.
- [0023] 일부 예시적인 구현에서, 비디오 스트림의 루프 내(in-loop) 필터링을 위한 방법이 개시된다. 이 방법은, 비디오 스트림의 현재 재구성된 데이터 블록에서 적어도 제1 색 성분의 재구성된 샘플과 연관된 적어도 하나의 통계적 특성을 획득하는 단계; 적어도 하나의 통계적 특성에 기초하여 복수의 샘플 오프셋 필터 중에서 타깃 샘플 오프셋 필터를 선택하는 단계 - 타깃 샘플 오프셋 필터는 샘플 델타(delta) 측정과 샘플 오프셋 값 사이의 비선형 매핑을 포함함 -; 및 현재 샘플의 필터링된 재구성된 샘플을 생성하기 위해, 현재 재구성된 데이터 블록의 제3 색 성분의 참조 샘플 및 타깃 샘플 오프셋 필터를 사용하여 현재 재구성된 데이터 블록의 제2 색 성분의 현재 샘플을 필터링하는 단계를 포함할 수 있다.
- [0024] 위의 구현에서, 제1 색 성분은 제3 색 성분과 동일한 색 성분일 수 있다. 제1 색 성분은 제2 색 성분과 동일한 색 성분일 수 있다. 제2 색 성분은 제3 색 성분과 다른 색 성분일 수 있다. 제2 색 성분은 제3 색 성분과 동일한 색 성분일 수 있다. 적어도 제1 색 성분은 1개, 2개 또는 3개의 색 성분을 포함할 수 있다.
- [0025] 위의 구현 중 임의의 구현에서, 적어도 하나의 통계적 특성은 현재 재구성된 데이터 블록의 에지 정보를 포함할 수 있다. 또한, 현재 재구성된 데이터 블록의 에지 정보는 제한된 방향 향상 필터링(CDEF: Constrained Directional Enhanced Filtering) 프로세스에서 도출된 에지 방향을 포함하고, 복수의 샘플 오프셋 필터는 N개의 CDEF 에지 방향에 대응하는 N개의 샘플 오프셋 필터를 포함하며, N은 1부터 8까지의 정수이다. 또한, 복수의 샘플 오프셋 필터는 고레벨 선택스(HLS: high level syntax)의 프레임 레벨에서 시그널링된다.
- [0026] 위의 구현 중 임의의 구현에서, 적어도 하나의 통계적 특성은 현재 재구성된 데이터 블록의 평활도 측정을 포함한다. 또한, 현재 재구성된 데이터 블록의 평활도 측정은 M-1개의 평활도 레벨 임계치를 특징으로 하는 M개의 미리 정의된 평활도 레벨 중 하나에 매핑되고; 복수의 샘플 오프셋 필터는 M개의 미리 정의된 평활도 레벨에 대응하는 M개의 샘플 오프셋 필터를 포함하고; 그리고 타깃 샘플 오프셋 필터는 평활도 측정에 매핑된 M개의 미리 정의된 평활도 레벨 중 하나에 따라 M개의 샘플 오프셋 필터 중에서 선택된다.
- [0027] 위의 구현 중 임의의 구현에서 적어도 하나의 통계적 특성은 현재 재구성된 데이터 블록의 코딩 정보를 포함한다. 또한, 코딩 정보는 현재 재구성된 데이터 블록의 현재 예측 모드를 포함하고; 복수의 샘플 오프셋 필터는 상이한 예측 모드에 대응하고; 그리고 타깃 샘플 오프셋 필터는 현재 재구성된 데이터 블록의 현재 예측 모드에 따라 복수의 샘플 오프셋 필터 중에서 선택된다. 또한, 상이한 예측 모드는 인트라 DC 모드, 인트라 평면(Planar) 모드, 인트라 PAETH 모드, 인트라 SMOOTH 모드, 인트라 채귀 필터링 모드 및 인트라 SKIP 모드 중 적어도 하나를 포함한다.
- [0028] 위의 구현 중 임의의 구현에서, 복수의 샘플 오프셋 필터 각각은 한 세트의 필터 계수, 다수의 필터 탭(tap), 및 다수의 필터 탭의 포지션과 연관된다.
- [0029] 위의 구현 중 임의의 구현에서, 현재 샘플의 필터링된 재구성된 샘플을 생성하기 위해, 현재 재구성된 데이터 블록의 제3 색 성분의 참조 샘플 및 타깃 샘플 오프셋 필터를 사용하여 현재 재구성된 데이터 블록의 제2 색 성분의 현재 샘플을 필터링하는 단계는, 제2 색 성분의 현재 샘플의 제1 위치 및 타깃 샘플 오프셋 필터와 연관된 복수의 필터 탭의 제2 위치를 결정하는 단계; 제1 위치 및 제2 위치에서 제3 색 성분의 재구성된 샘플을 참조 샘플로서 식별하는 단계; 현재 재구성된 데이터 블록의 제3 색 성분 모두에서 제2 위치에 대응하는 참조 샘플과 제1 위치에 대응하는 참조 샘플 사이의 델타 측정을 결정하는 단계; 델타 측정에 기초하여 타깃 샘플 오프셋 필터로부터 샘플 오프셋 값을 추출하는 단계; 및 현재 샘플의 필터링된 재구성된 샘플을 생성하기 위해, 샘플 오프셋 값을 사용하여 제2 색 성분의 현재 샘플을 필터링하는 단계를 포함할 수 있다.
- [0030] 위의 구현 중 임의의 구현에서, 복수의 샘플 오프셋 필터는 미리 결정될 수 있다. 복수의 샘플 오프셋 필터 및 복수의 샘플 오프셋 필터 중에서 선택된 타깃 샘플 오프셋 필터의 인덱스는 시퀀스 레벨, 픽처 레벨 또는 코딩 트리 유닛(coding tree unit) 레벨에서 시그널링될 수 있다.



[0031] 일부 구현에서, 비디오 인코딩 또는 디코딩 디바이스가 개시된다. 이 디바이스는 위의 방법 중 임의의 방법을 구현하도록 구성된 회로를 포함할 수 있다.

[0032] 본 개시내용의 양상은 또한, 비디오 디코딩 및/또는 인코딩을 위해 컴퓨터에 의해 실행될 때, 컴퓨터로 하여금 비디오 디코딩 및/또는 인코딩을 위한 방법을 수행하게 하는 명령을 저장하는 비-일시적 컴퓨터 판독 가능 매체를 제공한다.

### 도면의 간단한 설명

[0033] 개시된 청구 대상의 추가 특징, 성질 및 다양한 이점은 다음의 상세한 설명 및 첨부 도면으로부터 보다 명백해질 것이다.

도 1a는 인트라 예측 방향 모드의 예시적인 서브세트의 개략적인 예시를 도시한다.

도 1b는 예시적인 인트라 예측 방향의 예시를 도시한다.

도 2는 일 예에서 움직임 벡터 예측을 위한 현재 블록 및 그 주변 공간 병합 후보의 개략적인 예시를 도시한다.

도 3은 예시적인 실시예에 따른 통신 시스템(300)의 단순화된 블록도의 개략적인 예시를 도시한다.

도 4는 예시적인 실시예에 따른 통신 시스템(400)의 단순화된 블록도의 개략적인 예시를 도시한다.

도 5는 예시적인 실시예에 따른 비디오 디코더의 단순화된 블록도의 개략적인 예시를 도시한다.

도 6은 예시적인 실시예에 따른 비디오 인코더의 단순화된 블록도의 개략적인 예시를 도시한다.

도 7은 다른 예시적인 실시예에 따른 비디오 인코더의 블록도를 도시한다.

도 8은 다른 예시적인 실시예에 따른 비디오 디코더의 블록도를 도시한다.

도 9는 본 개시내용의 실시예에 따른 예시적인 적응 루프 필터를 도시한다.

도 10a - 도 10d는 본 개시내용의 실시예에 따라 수직 방향, 수평 방향 및 2개의 대각선 방향의 변화도 (gradient)를 각각 계산하기 위해 사용되는 서브샘플링된 포지션의 예를 도시한다.

도 10e는 적응 루프 필터(ALF: Adaptive Loop Filter)에 의해 사용하기 위한 다양한 변화도에 기초하여 블록 방향성을 결정하기 위한 예시적인 방식을 도시한다.

도 11a 및 도 11b는 본 개시내용의 예시적인 실시예에 따른 가상 경계에서의 수정된 블록 분류를 도시한다.

도 12a - 도 12f는 본 개시내용의 실시예에 따른 각각의 가상 경계에서 패딩(padding) 동작을 이용하는 예시적인 적응 루프 필터를 도시한다.

도 13은 본 개시내용의 일 실시예에 따른 최대 코딩 유닛 정렬 픽처 쿼드 트리(quadtree) 분할의 일례를 도시한다.

도 14는 본 개시내용의 예시적인 실시예에 따른, 도 13에 대응하는 쿼드 트리 분할 패턴을 도시한다.

도 15는 본 개시내용의 예시적인 실시예에 따라 크로마 성분을 생성하기 위해 사용되는 성분 간 필터를 도시한다.

도 16은 본 개시내용의 일 실시예에 따른 성분 간 ALF 필터의 일례를 도시한다.

도 17a - 도 17b는 본 개시내용의 실시예에 따른 루마(luma) 샘플에 대한 크로마 샘플의 예시적인 위치를 도시한다.

도 18은 본 개시내용의 일 실시예에 따른 블록에 대한 방향 탐색의 일례를 도시한다.

도 19는 본 개시내용의 일 실시예에 따른 부분 공간(subspace) 투영의 일례를 도시한다.

도 20은 본 개시내용의 일 실시예에 따른 성분 간 샘플 오프셋(CCSO: Cross-Component Sample Offset) 필터의 필터 지원 영역의 일례를 도시한다.

도 21a - 도 21c는 본 개시내용의 일 실시예에 따른 CCSO 필터에서 사용되는 예시적인 매핑을 도시한다.

도 22는 본 개시내용의 일 실시예에 따른 CCSO 필터의 예시적인 구현을 도시한다.

도 23은 본 개시내용의 일 실시예에 따른 예지 오프셋에서 픽셀 분류를 위한 4개의 예시적인 패턴을 도시한다.

도 24는 본 개시내용의 일 실시예에 따른 프로세스(2400)의 개요를 서술하는 흐름도를 도시한다.

도 25는 예시적인 실시예에 따른 컴퓨터 시스템의 개략적인 예시를 도시한다.

### 발명을 실시하기 위한 구체적인 내용

- [0034] 도 3은 본 개시내용의 일 실시예에 따른 통신 시스템(300)의 단순화된 블록도를 예시한다. 통신 시스템(300)은 예를 들어, 네트워크(350)를 통해 서로 통신할 수 있는 복수의 단말 디바이스를 포함한다. 예를 들어, 통신 시스템(300)은 네트워크(350)를 통해 상호 연결된 제1 쌍의 단말 디바이스(310, 320)를 포함한다. 도 3의 예에서, 제1 쌍의 단말 디바이스(310, 320)는 데이터의 단방향 전송을 수행할 수 있다. 예를 들어, 단말 디바이스(310)는 네트워크(350)를 통해 다른 단말 디바이스(320)로 전송하기 위한 (예컨대, 단말 디바이스(310)에 의해 캡처되는 비디오 픽처의 스트림의) 비디오 데이터를 코딩할 수 있다. 인코딩된 비디오 데이터는 하나 이상의 코딩된 비디오 비트스트림의 형태로 전송될 수 있다. 단말 디바이스(320)는 네트워크(350)로부터 코딩된 비디오 데이터를 수신하고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처를 복원하고, 복원된 비디오 데이터에 따라 비디오 픽처를 디스플레이할 수 있다. 단방향 데이터 전송은 미디어 서빙 애플리케이션 등에서 구현될 수 있다.
- [0035] 다른 예에서, 통신 시스템(300)은 예를 들어, 화상 회의 애플리케이션 중에 구현될 수 있는 코딩된 비디오 데이터의 양방향 전송을 수행하는 제2 쌍의 단말 디바이스(330, 340)를 포함한다. 데이터의 양방향 전송을 위해, 일례로, 단말 디바이스(330, 340)의 각각의 단말 디바이스는 네트워크(350)를 통해 단말 디바이스(330, 340) 중 다른 단말 디바이스로 전송하기 위한 (예컨대, 단말 디바이스에 의해 캡처되는 비디오 픽처의 스트림의) 비디오 데이터를 코딩할 수 있다. 단말 디바이스(330, 340)의 각각의 단말 디바이스는 또한, 단말 디바이스(330, 340) 중 다른 단말 디바이스에 의해 전송된 코딩된 비디오 데이터를 수신할 수 있고, 코딩된 비디오 데이터를 디코딩하여 비디오 픽처를 복원할 수 있으며, 복원된 비디오 데이터에 따라 액세스 가능한 디스플레이 디바이스에 비디오 픽처를 디스플레이할 수 있다.
- [0036] 도 3의 예에서, 단말 디바이스(310, 320, 330, 340)는 서버, 개인용 컴퓨터 및 스마트폰으로서 구현될 수 있지만, 본 개시내용의 기반 원리의 적용 가능성은 그렇게 제한되지는 않을 수 있다. 본 개시내용의 실시예는 데스크톱 컴퓨터, 랩톱 컴퓨터, 태블릿 컴퓨터, 미디어 플레이어, 웨어러블 컴퓨터, 전용 화상 회의 장비 등으로 구현될 수 있다. 네트워크(350)는 예를 들어, 와이어라인(유선) 및/또는 무선 통신 네트워크를 포함하여, 코딩된 비디오 데이터를 단말 디바이스(310, 320, 330, 340) 사이에 전달하는 임의의 수 또는 타입의 네트워크를 나타낸다. 통신 네트워크(350)는 회선 교환, 패킷 교환 및/또는 다른 타입의 채널에서 데이터를 교환할 수 있다. 대표적인 네트워크는 원격 통신 네트워크, 근거리 네트워크, 광역 네트워크 및/또는 인터넷을 포함한다. 본 논의의 목적으로, 네트워크(350)의 아키텍처(architecture) 및 토폴로지(topology)는 본 명세서에서 명시적으로 설명되지 않는 한 본 개시내용의 동작에 중요하지 않을 수 있다.
- [0037] 도 4는 개시된 청구 대상에 대한 애플리케이션에 대한 일례로, 비디오 스트리밍 환경에서의 비디오 인코더 및 비디오 디코더의 배치를 예시한다. 개시된 청구 대상은 예를 들어, 화상 회의, 디지털 TV 방송, 게임, 가상 현실, CD, DVD, 메모리 스틱 등을 포함하는 디지털 매체 상의 압축된 비디오의 저장 등을 포함하는 다른 비디오 가능 애플리케이션에 동일하게 적용 가능할 수 있다.
- [0038] 비디오 스트리밍 시스템은 압축되지 않은 비디오 픽처 또는 이미지의 스트림(402)을 생성하기 위한 비디오 소스(401), 예컨대 디지털 카메라를 포함할 수 있는 비디오 캡처 서브시스템(413)을 포함할 수 있다. 일례로, 비디오 픽처의 스트림(402)은 비디오 소스(401)의 디지털 카메라에 의해 기록되는 샘플을 포함한다. 인코딩된 비디오 데이터(404)(또는 코딩된 비디오 비트스트림)와 비교할 때 높은 데이터 볼륨을 강조하기 위해 굵은 선으로 표시된 비디오 픽처의 스트림(402)은 비디오 소스(401)에 결합된 비디오 인코더(403)를 포함하는 전자 디바이스(420)에 의해 처리될 수 있다. 비디오 인코더(403)는 아래에서 더 상세히 설명되는 바와 같이, 개시된 청구 대상의 양상을 가능하게 하거나 구현하기 위한 하드웨어, 소프트웨어, 또는 이들의 조합을 포함할 수 있다. 압축되지 않은 비디오 픽처의 스트림(402)과 비교할 때 더 낮은 데이터 볼륨을 강조하기 위해 가는 선으로 표시된 인코딩된 비디오 데이터(404)(또는 인코딩된 비디오 비트스트림(404))는 향후 사용을 위해 스트리밍 서버(405) 상에 또는 (도시되지 않은) 다운스트림 비디오 디바이스에 직접 저장될 수 있다. 도 4의 클라이언트 서브시스템(406, 408)과 같은 하나 이상의 스트리밍 클라이언트 서브시스템은 스트리밍 서버(405)에 액세스하여 인코딩된 비디오 데이터(404)의 사본(407, 409)을 리트리브(retrieve)할 수 있다. 클라이언트 서브시스템(406)은 예

를 들어, 전자 디바이스(430)에 비디오 디코더(410)를 포함할 수 있다. 비디오 디코더(410)는 인코딩된 비디오 데이터의 착신(incoming) 사본(407)을 디코딩하고, 압축되지 않은 그리고 디스플레이(412)(예컨대, 디스플레이 스크린) 또는 (도시되지 않은) 다른 렌더링 디바이스에 렌더링될 수 있는 발신 비디오 픽처의 스트림(411)을 생성할 수 있다. 비디오 디코더(410)는 본 개시내용에서 설명되는 다양한 기능 중 일부 또는 전부를 수행하도록 구성될 수 있다. 일부 스트리밍 시스템에서, 인코딩된 비디오 데이터(404, 407, 409)(예컨대, 비디오 비트스트림)은 특정 비디오 코딩/압축 표준에 따라 인코딩될 수 있다. 그러한 표준의 예는 ITU-T 권고 H.265를 포함한다. 일례로, 개발 중인 비디오 코딩 표준은 비공식적으로, 다용도 비디오 코딩(VVC)으로 알려져 있다. 개시되는 청구 대상은 VVC 및 다른 비디오 코딩 표준과 관련하여 사용될 수 있다.

[0039] 전자 디바이스(420, 430)는 (도시되지 않은) 다른 컴포넌트를 포함할 수 있다는 점이 주목된다. 예를 들어, 전자 디바이스(420)는 (도시되지 않은) 비디오 디코더를 포함할 수 있고, 전자 디바이스(430)는 (도시되지 않은) 비디오 인코더를 또한 포함할 수 있다.

[0040] 도 5는 아래 본 개시내용의 임의의 실시예에 따른 비디오 디코더(510)의 블록도를 도시한다. 비디오 디코더(510)는 전자 디바이스(530)에 포함될 수 있다. 전자 디바이스(530)는 수신기(531)(예컨대, 수신 회로)를 포함할 수 있다. 비디오 디코더(510)는 도 4의 예에서 비디오 디코더(410) 대신에 사용될 수 있다.

[0041] 수신기(531)는 비디오 디코더(510)에 의해 디코딩될 하나 이상의 코딩된 비디오 시퀀스를 수신할 수 있다. 동일한 또는 다른 실시예에서, 하나의 코딩된 비디오 시퀀스가 한 번에 디코딩될 수 있으며, 여기서 각각의 코딩된 비디오 시퀀스의 디코딩은 다른 코딩된 비디오 시퀀스와 독립적이다. 각각의 비디오 시퀀스는 다수의 비디오 프레임 또는 이미지와 연관될 수 있다. 코딩된 비디오 시퀀스는 하드웨어/소프트웨어 링크될 수 있는 채널(501)로부터, 인코딩된 비디오 데이터를 저장하는 저장 디바이스 또는 인코딩된 비디오 데이터를 전송하는 스트리밍 소스에 수신될 수 있다. 수신기(531)는 (도시되지 않은) 이들 각각의 처리 회로에 포워딩될 수 있는 다른 데이터, 이를테면 코딩된 오디오 데이터 및/또는 보조 데이터 스트림과 함께 인코딩된 비디오 데이터를 수신할 수 있다. 수신기(531)는 코딩된 비디오 시퀀스를 다른 데이터로부터 분리할 수 있다. 네트워크 지터(jitter)에 대처하기 위해, 버퍼 메모리(515)는 수신기(531)와 엔트로피 디코더/파서(parser)(520)(이하 "파서(520)") 사이에 배치될 수 있다. 특정 애플리케이션에서, 버퍼 메모리(515)는 비디오 디코더(510)의 일부로서 구현될 수 있다. 다른 애플리케이션에서, 버퍼 메모리(515)는 (도시되지 않은) 비디오 디코더(510) 외부에 있고 비디오 디코더(510)로부터 분리될 수 있다. 또 다른 애플리케이션에서, 예를 들어 네트워크 지터를 방지할 목적으로, 비디오 디코더(510) 외부에 있는 (도시되지 않은) 버퍼 메모리가 있을 수 있고, 예를 들어 재생 타이밍을 처리하기 위해 비디오 디코더(510) 내부에 다른 추가 버퍼 메모리(515)가 있을 수 있다. 수신기(531)가 충분한 대역폭 및 제어 가능성의 저장/전달 디바이스로부터, 또는 동시성(isosynchronous) 네트워크로부터 데이터를 수신하고 있을 때, 버퍼 메모리(515)는 필요하지 않을 수 있거나 작을 수 있다. 인터넷과 같은 최선형(best-effort) 패킷 네트워크 상에서의 사용을 위해, 충분한 크기의 버퍼 메모리(515)가 요구될 수 있으며, 그 크기는 비교적 클 수 있다. 이러한 버퍼 메모리는 적응적 크기로 구현될 수 있고, 비디오 디코더(510) 외부의 운영 체제 또는 유사한 엘리먼트(도시되지 않음)에서 적어도 부분적으로 구현될 수 있다.

[0042] 비디오 디코더(510)는 코딩된 비디오 시퀀스로부터 심벌(521)을 재구성하기 위한 파서(520)를 포함할 수 있다. 그러한 심벌의 카테고리는 비디오 디코더(510)의 동작을 관리하는 데 사용되는 정보, 및 잠재적으로는, 도 5에 도시된 바와 같이, 전자 디바이스(530)의 내장 부품일 수도 또는 아닐 수도 있지만 전자 디바이스(530)에 결합될 수 있는 디스플레이(512)(예컨대, 디스플레이 스크린)와 같은 렌더링 디바이스를 제어하기 위한 정보를 포함한다. 렌더링 디바이스(들)에 대한 제어 정보는 부가 확장 정보(SEI: Supplemental Enhancement Information) 메시지 또는 비디오 사용성 정보(VUI: Video Usability Information) 파라미터 세트 프래그먼트(도시되지 않음)의 형태일 수 있다. 파서(520)는 파서(520)에 의해 수신되는 코딩된 비디오 시퀀스를 파싱(parse)/엔트로피 디코딩할 수 있다. 코딩된 비디오 시퀀스의 엔트로피 코딩은 비디오 코딩 기술 또는 표준에 따를 수 있고, 가변 길이 코딩, 허프만(Huffman) 코딩, 상황 민감도를 갖거나 갖지 않는 산술 코딩 등을 포함하여, 다양한 원리를 따를 수 있다. 파서(520)는 하위 그룹에 대응하는 적어도 하나의 파라미터에 기초하여, 비디오 디코더 내의 픽셀의 하위 그룹 중 적어도 하나에 대한 한 세트의 하위 그룹 파라미터를 코딩된 비디오 시퀀스로부터 추출할 수 있다. 하위 그룹은 픽처의 그룹(GOP: Group of Pictures), 픽처, 타일, 슬라이스, 매크로 블록, 코딩 유닛(CU: Coding Unit), 블록, 변환 유닛(TU: Transform Unit), 예측 유닛(PU: Prediction Unit) 등을 포함할 수 있다. 파서(520)는 또한 변환 계수들(예컨대, 푸리에 변환 계수), 양자화기 파라미터 값, 움직임 벡터 등과 같은 정보를 코딩된 비디오 시퀀스로부터 추출할 수 있다.

[0043] 파서(520)는 심벌(521)을 생성하기 위해, 버퍼 메모리(515)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코



딩/파싱 동작을 수행할 수 있다.

- [0044] 심벌(521)의 재구성성은 코딩된 비디오 픽처 또는 그 부분의 타입(이를테면: 인터(inter) 및 인트라 픽처, 인터 및 인트라 블록) 및 다른 팩터에 따라 다수의 상이한 처리 또는 기능 유닛을 수반할 수 있다. 수반되는 유닛, 그리고 이들이 어떻게 수반되는지는 코딩된 비디오 시퀀스로부터 파서(520)에 의해 파싱된 하위 그룹 제어 정보에 의해 제어될 수 있다. 파서(520)와 아래의 다수의 처리 또는 기능 유닛 사이의 그러한 하위 그룹 제어 정보의 흐름은 단순성을 위해 도시되지 않는다.
- [0045] 이미 언급된 기능 블록 외에도, 비디오 디코더(510)는 아래에서 설명되는 바와 같이 다수의 기능 유닛으로 개념적으로 세분될 수 있다. 상업적 제약 하에서 동작하는 실제 구현에서, 이러한 기능 유닛 중 다수는 서로 밀접하게 상호 작용하고, 적어도 부분적으로 서로 통합될 수 있다. 그러나 개시된 청구 대상의 다양한 기능을 명확하게 설명할 목적으로, 기능 유닛으로의 개념적 세분화가 아래의 개시내용에서 채택된다.
- [0046] 제1 유닛은 스케일러/역변환 유닛(551)을 포함할 수 있다. 스케일러/역변환 유닛(551)은 파서(520)로부터의 심벌(들)(521)로서, 어떤 타입의 역변환을 사용할지, 블록 크기, 양자화 팩터/파라미터, 양자화 스케일링 행렬 등을 지시하는 정보를 포함하는 제어 정보뿐만 아니라 양자화된 변환 계수를 수신할 수 있다. 스케일러/역변환 유닛(551)은 애그리게이터(aggregator)(555)에 입력될 수 있는 샘플 값을 포함하는 블록을 출력할 수 있다.
- [0047] 일부 경우에서, 스케일러/역변환(551)의 출력 샘플은 인트라 코딩된 블록, 즉 이전에 재구성된 픽처로부터의 예측 정보를 사용하는 것이 아니라, 현재 픽처의 이전에 재구성된 부분으로부터의 예측 정보를 사용할 수 있는 블록에 속할 수 있다. 그러한 예측 정보는 인트라 픽처 예측 유닛(552)에 의해 제공될 수 있다. 일부 경우에서, 인트라 픽처 예측 유닛(552)은 이미 재구성된 그리고 현재 픽처 버퍼(558)에 저장된 주변의 블록 정보를 사용하여, 재구성 하의 블록과 동일한 크기 및 형상의 블록을 생성할 수 있다. 현재 픽처 버퍼(558)는 예를 들어, 부분적으로 재구성된 현재 픽처 및/또는 완전히 재구성된 현재 픽처를 버퍼링한다. 애그리게이터(555)는 일부 구현에서, 인트라 예측 유닛(552)이 생성한 예측 정보를 스케일러/역변환 유닛(551)에 의해 제공되는 출력 샘플 정보에 샘플 단위로 추가할 수 있다.
- [0048] 다른 경우에서, 스케일러/역변환 유닛(551)의 출력 샘플은, 인터 코딩되고 잠재적으로 움직임 보상된 블록에 속할 수 있다. 그러한 경우에, 움직임 보상 예측 유닛(553)은 인터 픽처 예측에 사용되는 샘플을 폐치하기 위해 참조 픽처 메모리(557)에 액세스할 수 있다. 블록에 속하는 심벌(521)에 따라 폐치된 샘플을 움직임 보상한 후에, 이러한 샘플은 애그리게이터(555)에 의해 스케일러/역변환 유닛(551)의 출력(유닛(551)의 출력은 잔차 샘플 또는 잔차 신호로 지칭될 수 있음)에 추가되어 출력 샘플 정보를 생성할 수 있다. 움직임 보상 예측 유닛(553)이 예측 샘플을 폐치하는 참조 픽처 메모리(557) 내의 어드레스는 예를 들어 X, Y 성분(시프트) 및 참조 픽처 성분(시간)을 가질 수 있는 심벌(521)의 형태로 움직임 보상 예측 유닛(553)에 이용 가능한 움직임 벡터에 의해 제어될 수 있다. 움직임 보상은 또한, 서브샘플 정밀 움직임 벡터가 사용 중일 때 참조 픽처 메모리(557)로부터 폐치된 샘플 값의 보간을 포함할 수 있고, 또한 움직임 벡터 예측 메커니즘 등과 연관될 수 있다.
- [0049] 애그리게이터(555)의 출력 샘플은 루프 필터 유닛(556)에서 다양한 루프 필터링 기법을 겪을 수 있다. 비디오 압축 기술은 (코딩된 비디오 비트스트림으로도 또한 지칭되는) 코딩된 비디오 시퀀스에 포함된 파라미터에 의해 제어되고 파서(520)로부터의 심벌(521)로서 루프 필터 유닛(556)에 이용 가능하게 되는 루프 내 필터 기술을 포함할 수 있지만, 코딩된 픽처 또는 코딩된 비디오 시퀀스의 (디코딩 순서로) 이전 부분의 디코딩 동안 획득된 메타 정보에 응답할 뿐만 아니라, 이전에 재구성되고 루프 필터링된 샘플 값에 또한 응답할 수 있다. 아래에서 보다 상세히 설명되는 바와 같이, 여러 타입의 루프 필터가 루프 필터 유닛(556)의 일부로서 다양한 순서로 포함될 수 있다.
- [0050] 루프 필터 유닛(556)의 출력은 렌더링 디바이스(512)에 출력될 수 있을 뿐만 아니라 향후 인터 픽처 예측에서의 사용을 위해 참조 픽처 메모리(557)에 저장될 수 있는 샘플 스트림일 수 있다.
- [0051] 특정 코딩된 픽처는 일단 완전히 재구성되면, 향후 인터 픽처 예측을 위한 참조 픽처로서 사용될 수 있다. 예를 들어, 현재 픽처에 대응하는 코딩된 픽처가 완전히 재구성되고 코딩된 픽처가 (예를 들어, 파서(520)에 의해) 참조 픽처로서 식별되면, 현재 픽처 버퍼(558)는 참조 픽처 메모리(557)의 일부가 될 수 있고, 다음의 코딩된 픽처의 재구성을 시작하기 전에 새로운 현재 픽처 버퍼가 재할당될 수 있다.
- [0052] 비디오 디코더(510)는 ITU-T Rec. H.265와 같은 표준에 채택된 미리 결정된 비디오 압축 기술에 따라 디코딩 동작을 수행할 수 있다. 코딩된 비디오 시퀀스는, 코딩된 비디오 시퀀스가 비디오 압축 기술 또는 표준에 문서화된 프로파일 및 비디오 압축 기술 또는 표준의 선택스 모두를 고수한다는 의미로, 사용되고 있는 비디오 압축

기술 또는 표준에 의해 특정된 선택스를 따를 수 있다. 구체적으로, 프로파일은 해당 프로파일 하에서의 사용에 이용 가능한 유일한 도구로서 비디오 압축 기술 또는 표준에서 이용 가능한 모든 도구로부터 특정 도구를 선택할 수 있다. 표준을 준수하자면, 코딩된 비디오 시퀀스의 복잡성은 비디오 압축 기술 또는 표준의 레벨에 의해 정의된 범위 내에 있을 수 있다. 일부 경우에서, 레벨은 최대 픽처 크기, 최대 프레임 레이트, (예를 들어, 초당 메가샘플로 측정된) 최대 재구성 샘플 레이트, 최대 참조 픽처 크기 등을 제한한다. 레벨에 의해 설정된 제한은 일부 경우에는, 코딩된 비디오 시퀀스로 시그널링되는 가상 참조 디코더(HRD: Hypothetical Reference Decoder) 버퍼 관리를 위한 메타데이터(metadata) 및 HRD 규격을 통해 추가로 제한될 수 있다.

[0053] 일부 예시적인 실시예에서, 수신기(531)는 인코딩된 비디오와 함께 추가 (중복) 데이터를 수신할 수 있다. 추가 데이터는 코딩된 비디오 시퀀스(들)의 일부로서 포함될 수 있다. 추가 데이터는 데이터를 적절히 디코딩하기 위해 그리고/또는 원본 비디오 데이터를 보다 정확하게 재구성하기 위해 비디오 디코더(510)에 의해 사용될 수 있다. 추가 데이터는 예를 들어, 시간적, 공간적 또는 신호 잡음비(SNR: signal noise ratio) 강화 계층, 중복 슬라이스, 중복 픽처, 순방향 에러 정정 코드 등의 형태일 수 있다.

[0054] 도 6은 본 개시내용의 예시적인 실시예에 따른 비디오 인코더(603)의 블록도를 도시한다. 비디오 인코더(603)는 전자 디바이스(620)에 포함될 수 있다. 전자 디바이스(620)는 전송기(640)(예컨대, 전송 회로)를 더 포함할 수 있다. 비디오 인코더(603)는 도 4의 예에서의 비디오 인코더(403) 대신에 사용될 수 있다.

[0055] 비디오 인코더(603)는 비디오 인코더(603)에 의해 코딩될 비디오 이미지(들)를 캡처할 수 있는 (도 6의 예에서 전자 디바이스(620)의 일부가 아닌) 비디오 소스(601)로부터 비디오 샘플을 수신할 수 있다. 다른 예에서, 비디오 소스(601)는 전자 디바이스(620)의 일부로서 구현될 수 있다.

[0056] 비디오 소스(601)는 임의의 적절한 비트 심도(예를 들어: 8비트, 10비트, 12비트, ...), 임의의 색 공간(예를 들어, BT.601 YCrCb, RGB, XYZ...), 및 임의의 적절한 샘플링 구조(예를 들어, YCrCb 4:2:0, YCrCb 4:4:4)일 수 있는 디지털 비디오 샘플 스트림의 형태로 비디오 인코더(603)에 의해 코딩될 소스 비디오 시퀀스를 제공할 수 있다. 미디어 서빙 시스템에서, 비디오 소스(601)는 이전에 준비된 비디오를 저장할 수 있는 저장 디바이스일 수 있다. 화상 회의 시스템에서, 비디오 소스(601)는 로컬 이미지 정보를 비디오 시퀀스로서 캡처하는 카메라일 수 있다. 비디오 데이터는 순차적으로 볼 때 움직임을 부여하는 복수의 개별 픽처 또는 이미지로서 제공될 수 있다. 픽처 자체는 픽셀의 공간 어레이로서 구성될 수 있으며, 여기서 각각의 픽셀은 사용 중인 샘플링 구조, 색 공간 등에 따라 하나 이상의 샘플을 포함할 수 있다. 당해 기술분야에서 통상의 지식을 가진 자는 픽셀과 샘플 간의 관계를 쉽게 이해할 수 있다. 아래의 설명은 샘플에 초점을 맞춘다.

[0057] 일부 예시적인 실시예에 따르면, 비디오 인코더(603)는 실시간으로 또는 애플리케이션에 의해 요구되는 임의의 다른 시간 제약 하에서 소스 비디오 시퀀스의 픽처를 코딩된 비디오 시퀀스(643)로 코딩 및 압축할 수 있다. 적절한 코딩 속도를 시행하는 것은 제어기(650)의 하나의 기능을 구성한다. 일부 실시예에서, 제어기(650)는 아래에서 설명되는 바와 같이 다른 기능 유닛에 기능적으로 결합되어 다른 기능 유닛을 제어할 수 있다. 단순성을 위해 결합은 도시되지 않는다. 제어기(650)에 의해 설정된 파라미터는 레이트 제어 관련 파라미터(픽처 스킵(skip), 양자화기, 레이트 왜곡 최적화 기법의 람다(lambda) 값, ...), 픽처 크기, 픽처의 그룹(GOP) 레이아웃, 최대 움직임 벡터 탐색 범위 등을 포함할 수 있다. 제어기(650)는 특정 시스템 설계에 최적화된 비디오 인코더(603)에 속하는 다른 적절한 기능을 갖도록 구성될 수 있다.

[0058] 일부 예시적인 실시예에서, 비디오 인코더(603)는 코딩 루프에서 동작하도록 구성될 수 있다. 지나치게 단순화된 설명으로서, 일례로, 코딩 루프는 (예컨대, 참조 픽처(들) 및 코딩될 입력 픽처에 기초하여 심벌 스트림과 같은 심벌의 생성을 담당하는) 소스 코더(630) 및 비디오 인코더(603)에 임베드(embed)된 (로컬) 디코더(633)를 포함할 수 있다. 디코더(633)는 (개시된 청구 대상에서 고려되는 비디오 압축 기술에서는 엔트로피 코딩에서 심벌과 코딩된 비디오 비트스트림 간의 임의의 압축은 무손실일 수 있기 때문에) 임베드된 디코더(633)가 엔트로피 코딩 없이 소스 코더(630)에 의해 코딩된 비디오 스트림을 처리하더라도 (원격) 디코더가 생성할 것과 유사한 방식으로 샘플 데이터를 생성하도록 심벌을 재구성한다. 재구성된 샘플 스트림(샘플 데이터)은 참조 픽처 메모리(634)에 입력된다. 심벌 스트림의 디코딩이 (로컬 또는 원격) 디코더 위치와 독립적인 비트 정확한 결과로 이어지므로, 참조 픽처 메모리(634) 내의 콘텐츠는 또한 로컬 인코더와 원격 인코더 사이에서 비트가 정확하다. 다시 말해서, 인코더의 예측 부분은, 디코딩 동안 예측을 사용할 때 디코더가 "보는" 것과 정확히 동일한 샘플 값을 참조 픽처 샘플로서 "본다". 참조 픽처 동기성의 이러한 기본 원리(및 예를 들어, 채널 에러로 인해 동기성이 유지될 수 없다면, 결과적인 드리프트(drift))는 코딩 품질을 개선하는 데 사용된다.

[0059] "로컬" 디코더(633)의 동작은 도 5와 함께 위에서 이미 상세히 설명된 비디오 디코더(510)와 같은 "원격" 디코

더의 동작과 동일할 수 있다. 그러나 도 5를 또한 간략하게 참조하면, 심벌이 이용 가능하고 엔트로피 코더(645) 및 파서(520)에 의한 코딩된 비디오 시퀀스로의 심벌의 인코딩/디코딩이 무손실일 수 있기 때문에, 버퍼 메모리(515) 및 파서(520)를 포함하는 비디오 디코더(510)의 엔트로피 디코딩 부분은 인코더 내의 로컬 디코더(633)에서 완전히 구현되지 않을 수 있다.

[0060] 이 시점에서 이루어질 수 있는 관찰은, 디코더에만 존재할 수 있는 파싱/엔트로피 디코딩을 제외한 임의의 디코더 기술이 실질적으로 동일한 기능적 형태로 대응하는 인코더에 반드시 존재할 필요가 있을 수 있다는 것이다. 이러한 이유로, 개시된 청구 대상은 때때로 인코더의 디코딩 부분과 연합하는 디코더 동작에 초점을 맞출 수 있다. 따라서 인코더 기술은 포괄적으로 설명된 디코더 기술의 역이기 때문에 인코더 기술의 설명은 축약될 수 있다. 특정 영역 또는 양상에서만 인코더의 보다 상세한 설명이 아래에 제공된다.

[0061] 동작 중에, 일부 예시적인 구현에서, 소스 코더(630)는 "참조 픽처"로서 지정된 비디오 시퀀스로부터의 하나 이상의 이전에 코딩된 픽처를 참조하여 입력 픽처를 예측 코딩하는 움직임 보상 예측 코딩을 수행할 수 있다. 이런 식으로, 코딩 엔진(632)은 입력 픽처의 픽셀 블록과 입력 픽처에 대한 예측 참조(들)로서 선택될 수 있는 참조 픽처(들)의 픽셀 블록 간의 색 채널에서의 차이(또는 잔차)를 코딩한다.

[0062] 로컬 비디오 디코더(633)는 소스 코더(630)에 의해 생성된 심벌에 기반하여, 참조 픽처로서 지정될 수 있는 픽처의 코딩된 비디오 데이터를 디코딩할 수 있다. 코딩 엔진(632)의 동작은 유리하게는 손실 프로세스일 수 있다. 코딩된 비디오 데이터가 (도 6에 도시되지 않은) 비디오 디코더에서 디코딩될 수 있을 때, 재구성된 비디오 시퀀스는 통상적으로 일부 에러를 갖는 소스 비디오 시퀀스의 복제물일 수 있다. 로컬 비디오 디코더(633)는 비디오 디코더에 의해 참조 픽처에 수행될 수 있는 디코딩 프로세스를 복제하고, 재구성된 참조 픽처가 참조 픽처 캐시(634)에 저장되게 할 수 있다. 이런 식으로, 비디오 인코더(603)는 (전송 에러 없이) 원단(far-end)(원격) 비디오 디코더에 의해 획득될 재구성된 참조 픽처로서 공통 콘텐츠를 갖는 재구성된 참조 픽처의 사본을 국소적으로 저장할 수 있다.

[0063] 예측기(635)는 코딩 엔진(632)에 대한 예측 탐색을 수행할 수 있다. 즉, 코딩될 새로운 픽처에 대해, 예측기(635)는 (후보 참조 픽셀 블록으로서) 샘플 데이터, 또는 새로운 픽처에 대한 적절한 예측 참조로서의 역할을 할 수 있는 참조 픽처 움직임 벡터, 블록 형상 등과 같은 특정 메타데이터에 대해 참조 픽처 메모리(634)를 탐색할 수 있다. 예측기(635)는 적절한 예측 참조를 발견하기 위해 샘플 블록  $\times$  픽셀 블록 단위로 동작할 수 있다. 일부 경우에는, 예측기(635)에 의해 획득된 탐색 결과에 의해 결정된 대로, 입력 픽처는 참조 픽처 메모리(634)에 저장된 다수의 참조 픽처로부터 인출된 예측 참조를 가질 수 있다.

[0064] 제어기(650)는 예를 들어, 비디오 데이터를 인코딩하기 위해 사용되는 파라미터 및 하위 그룹 파라미터의 설정을 포함하는, 소스 코더(630)의 코딩 동작을 관리할 수 있다.

[0065] 앞서 언급된 모든 기능 유닛의 출력은 엔트로피 코더(645)에서 엔트로피 코딩을 거칠 수 있다. 엔트로피 코더(645)는 허프만 코딩, 가변 길이 코딩, 산술 코딩 등은 기술에 따라 심벌의 무손실 압축에 의해, 다양한 기능 유닛에 의해 생성된 심벌을 코딩된 비디오 시퀀스로 변환한다.

[0066] 전송기(640)는 엔트로피 코더(645)에 의해 생성된 코딩된 비디오 시퀀스(들)를 버퍼링하여, 인코딩된 비디오 데이터를 저장할 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는 통신 채널(660)을 통한 전송을 준비할 수 있다. 전송기(640)는 비디오 코더(603)로부터의 코딩된 비디오 데이터를 전송될 다른 데이터, 예를 들어 코딩된 오디오 데이터 및/또는 보조 데이터 스트림(소스는 도시되지 않음)과 병합할 수 있다.

[0067] 제어기(650)는 비디오 인코더(603)의 동작을 관리할 수 있다. 코딩 동안, 제어기(650)는 각각의 코딩된 픽처에 특정 코딩된 픽처 타입을 할당할 수 있으며, 이는 개개의 픽처에 적용될 수 있는 코딩 기술에 영향을 미칠 수 있다. 예를 들어, 픽처는 종종 다음의 픽처 타입 중 하나로서 할당될 수 있다:

[0068] 인트라 픽처(I 픽처)는 예측의 소스로서 시퀀스 내의 임의의 다른 픽처를 사용하지 않고 코딩 및 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱은 예를 들어, 독립적 디코더 리프레시(IDR: Independent Decoder Refresh) 픽처를 포함하는 상이한 타입의 인트라 픽처를 허용한다. 당해 기술분야에서 통상의 지식을 가진 자는 I 픽처의 그러한 변형 그리고 이들 개개의 애플리케이션 및 특징을 알고 있다.

[0069] 예측 픽처(P 픽처)는 각각의 블록의 샘플 값을 예측하기 위해 최대 하나의 움직임 벡터 및 참조 인덱스를 사용하는 인트라 예측 또는 인터 예측을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다.

[0070] 양방향 예측 픽처(B 픽처)는 각각의 블록의 샘플 값을 예측하기 위해 최대 2개의 움직임 벡터 및 참조 인덱스를

사용하는 인트라 예측 또는 인터 예측을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다. 유사하게, 다수의 예측 픽처는 단일 블록의 재구성을 위해 2개보다 많은 참조 픽처 및 연관된 메타데이터를 사용할 수 있다.

[0071] 소스 픽처는 일반적으로, 공간적으로 복수의 샘플 코딩 블록(예를 들어, 각각  $4 \times 4$ ,  $8 \times 8$ ,  $4 \times 8$  또는  $16 \times 16$  샘플의 블록)으로 세분되고 코딩 블록 단위로 코딩될 수 있다. 블록은 블록의 개개의 픽처에 적용된 코딩 할당에 의해 결정된 다른 (이미 코딩된) 블록을 참조하여 예측 코딩될 수 있다. 예를 들어, I 픽처의 블록은 비-예측 코딩될 수 있거나, 이는 동일한 픽처의 이미 코딩된 블록을 참조하여 예측(공간 예측 또는 인트라 예측) 코딩될 수 있다. P 픽처의 픽셀 블록은 하나의 이전에 코딩된 참조 픽처를 참조하여 시간 예측을 통해 또는 공간 예측을 통해 예측 코딩될 수 있다. B 픽처의 블록은 하나 또는 2개의 이전에 코딩된 참조 픽처를 참조하여 시간 예측을 통해 또는 공간 예측을 통해 예측 코딩될 수 있다. 소스 픽처 또는 중간 처리된 픽처는 다른 목적을 위해 다른 타입의 블록으로 세분될 수 있다. 코딩 블록 및 다른 타입의 블록의 분할은 아래에서 더 상세히 설명되는 바와 같이 동일한 방식을 따를 수도 또는 따르지 않을 수도 있다.

[0072] 비디오 인코더(603)는 ITU-T Rec. H.265와 같은 미리 결정된 비디오 코딩 기술 또는 표준에 따라 코딩 동작을 수행할 수 있다. 비디오 인코더(603)의 동작에서, 비디오 인코더(603)는 입력 비디오 시퀀스에서 시간적 및 공간적 중복성을 활용하는 예측 코딩 동작을 포함하는 다양한 압축 동작을 수행할 수 있다. 코딩된 비디오 데이터는 그에 따라, 사용되는 비디오 코딩 기술 또는 표준에 의해 특정된 신택스를 따를 수 있다.

[0073] 일부 예시적인 실시예에서, 전송기(640)는 인코딩된 비디오와 함께 추가 데이터를 전송할 수 있다. 소스 코더(630)는 코딩된 비디오 시퀀스의 일부로서 이러한 데이터를 포함할 수 있다. 추가 데이터는 시간적/공간적/SNR 강화 계층, 다른 형태의 중복 데이터, 이를테면 중복 픽처 및 슬라이스, SEI 메시지, VUI 파라미터 세트 프레임트 등을 포함할 수 있다.

[0074] 비디오는 시간 순서로 복수의 소스 픽처(비디오 픽처)로서 캡처될 수 있다. (종종 인트라 예측으로 축약되는) 인트라 픽처 예측은 주어진 픽처에서 공간적 상관을 이용하고, 인터 픽처 예측은 픽처 간의 시간적 또는 다른 상관을 이용한다. 예를 들어, 현재 픽처로 지칭되는 인코딩/디코딩 하의 특정 픽처가 블록으로 파티셔닝될 수 있다. 현재 픽처 내의 블록은, 비디오에서 이전에 코딩되고 여전히 버퍼링된 참조 픽처 내의 참조 블록과 유사한 경우, 움직임 벡터로 지칭되는 벡터에 의해 코딩될 수 있다. 움직임 벡터는 참조 픽처 내의 참조 블록을 가리키며, 다수의 참조 픽처가 사용 중인 경우, 참조 픽처를 식별하는 3차원을 가질 수 있다.

[0075] 일부 예시적인 실시예에서, 양방향 예측 기법이 인터 픽처 예측에 사용될 수 있다. 이러한 양방향 예측 기법에 따르면, 비디오에서 디코딩 순서가 둘 다 현재 픽처를 선행하는(그러나 디스플레이 순서는 각각 과거 또는 미래일 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 움직임 벡터 및 제2 참조 픽처 내의 제2 참조 블록을 가리키는 제2 움직임 벡터에 의해 코딩될 수 있다. 블록은 제1 참조 블록과 제2 참조 블록의 조합에 의해 공동으로 예측될 수 있다.

[0076] 또한, 코딩 효율을 향상시키기 위해 인터 픽처 예측에 병합 모드 기법이 사용될 수 있다.

[0077] 본 개시내용의 일부 예시적인 실시예에 따르면, 인터 픽처 예측 및 인트라 픽처 예측과 같은 예측이 블록 단위로 수행된다. 예를 들어, 비디오 픽처 시퀀스의 픽처는 압축을 위해 코딩 트리 유닛(CTU: coding tree unit)으로 분할되며, 픽처의 CTU는  $64 \times 64$  픽셀,  $32 \times 32$  픽셀 또는  $16 \times 16$  픽셀과 같은 동일한 크기를 가질 수 있다. 일반적으로, CTU는 3개의 병렬 코딩 트리 블록(CTB: coding tree block): 1개의 루마 CTB 및 2개의 크로마 CTB를 포함할 수 있다. 각각의 CTU는 하나 또는 다수의 코딩 유닛(CU)으로 재귀적으로 쿼드 트리 분할될 수 있다. 예를 들어,  $64 \times 64$  픽셀의 CTU는  $64 \times 64$  픽셀의 하나의 CU, 또는  $32 \times 32$  픽셀의 4개의 CU로 분할될 수 있다.  $32 \times 32$  블록 중 하나 이상 각각은  $16 \times 16$  픽셀의 4개의 CU로 추가 분할될 수 있다. 일부 예시적인 실시예에서, 각각의 CU는 인코딩 동안 분석되어 인터 예측 타입 또는 인트라 예측 타입과 같은 다양한 예측 타입 중에서 CU에 대한 예측 타입을 결정할 수 있다. CU는 시간적 그리고/또는 공간적 예측 가능성에 따라 하나 이상의 예측 유닛(PU)으로 분할될 수 있다. 일반적으로, 각각의 PU는 루마 예측 블록(PB: prediction block)과 2개의 크로마 PB를 포함한다. 일 실시예에서, 코딩(인코딩/디코딩)에서의 예측 동작은 예측 블록 단위로 수행된다. CU를 PU(또는 상이한 색 채널의 PB)로 분할하는 것은 다양한 공간 패턴으로 수행될 수 있다. 예를 들어, 루마 또는 크로마 PB는  $8 \times 8$  픽셀,  $16 \times 16$  픽셀,  $8 \times 16$  픽셀,  $16 \times 8$  샘플 등과 같은 샘플에 대한 값(예를 들어, 루마 값)의 행렬을 포함할 수 있다.

[0078] 도 7은 본 개시내용의 다른 예시적인 실시예에 따른 비디오 인코더(703)의 도면을 도시한다. 비디오 인코더



(703)는 비디오 픽처의 시퀀스에서 현재 비디오 픽처 내의 샘플 값의 처리 블록(예컨대, 예측 블록)을 수신하고, 그 처리 블록을 코딩된 픽처의 일부인 코딩된 픽처로 인코딩하도록 구성된다. 예시적인 비디오 인코더(703)는 도 4의 예에서의 비디오 인코더(403) 대신에 사용될 수 있다.

[0079] 예를 들어, 비디오 인코더(703)는  $8 \times 8$  샘플의 예측 블록 등과 같은 처리 블록에 대한 샘플 값의 행렬을 수신한다. 그런 다음, 비디오 인코더(703)는 예를 들어, 레이트 왜곡 최적화(RDO: rate-distortion optimization)를 사용하여 처리 블록이 인트라 모드를 사용하여 가장 잘 코딩되는지, 인터 모드를 사용하여 가장 잘 코딩되는지 또는 양방향 예측 모드를 사용하여 가장 잘 코딩되는지 여부를 결정한다. 처리 블록이 인트라 모드에서 코딩되는 것으로 결정될 때, 비디오 인코더(703)는 처리 블록을 코딩된 픽처로 인코딩하기 위한 인트라 예측 기술을 사용할 수 있고; 처리 블록이 인터 모드 또는 양방향 예측 모드에서 코딩되는 것으로 결정될 때, 비디오 인코더(703)는 처리 블록을 코딩된 픽처로 인코딩하기 위해 인터 예측 또는 양방향 예측 기술을 각각 사용할 수 있다. 일부 예시적인 실시예에서, 병합 모드는 움직임 벡터가 예측자 외부의 코딩된 움직임 벡터 성분의 이점 없이 하나 이상의 움직임 벡터 예측자로부터 도출되는 인터 픽처 예측의 하위 모드로서 사용될 수 있다. 다른 일부 예시적인 실시예에서, 대상 블록에 적용 가능한 움직임 벡터 성분이 존재할 수 있다. 이에 따라, 비디오 인코더(703)는 처리 블록의 예측 모드를 결정하기 위한, 도 7에 명시적으로 도시되지 않은 컴포넌트, 이를테면 모드 결정 모듈을 포함할 수 있다.

[0080] 도 7의 예에서, 비디오 인코더(703)는 도 7의 예시적인 배열에 도시된 바와 같이 함께 결합되는 인터 인코더(730), 인트라 인코더(722), 잔차 계산기(723), 스위치(726), 잔차 인코더(724), 일반 제어기(721) 및 엔트로피 인코더(725)를 포함한다.

[0081] 인터 인코더(730)는 현재 블록(예컨대, 처리 블록)의 샘플을 수신하고, 블록을 참조 픽처의 하나 이상의 참조 블록(예컨대, 디스플레이 순서상 이전 픽처 및 이후 픽처의 블록)과 비교하고, 인터 예측 정보(예컨대, 인터 인코딩 기술에 따른 중복 정보의 설명, 움직임 벡터, 병합 모드 정보), 및 임의의 적절한 기법을 사용하여 인터 예측 정보를 기초로 인터 예측 결과(예컨대, 예측된 블록)를 계산하도록 구성된다. 일부 예에서, 참조 픽처는 (아래에서 보다 상세히 설명되는 바와 같이, 도 7의 잔차 디코더(728)로서 도시된) 도 6의 예시적인 인코더(620)에 임베딩된 디코딩 유닛(633)을 사용하여, 인코딩된 비디오 정보를 기초로 디코딩되는 디코딩된 참조 픽처이다.

[0082] 인트라 인코더(722)는 현재 블록(예컨대, 처리 블록)의 샘플을 수신하고, 블록을 동일한 픽처에 이미 코딩된 블록과 비교하고, 변환 이후 양자화된 계수를 생성하고, 그리고 또한 일부 경우에는 인트라 예측 정보(예컨대, 하나 이상의 인트라 인코딩 기법에 따른 인트라 예측 방향 정보)를 생성하도록 구성된다. 인트라 인코더(722)는 동일한 픽처의 인트라 예측 정보 및 참조 블록에 기초하여 인트라 예측 결과(예컨대, 예측된 블록)를 계산할 수 있다.

[0083] 일반 제어기(721)는, 일반 제어 데이터를 결정하고 일반 제어 데이터에 기초하여 비디오 인코더(703)의 다른 컴포넌트를 제어하도록 구성될 수 있다. 일례로, 일반 제어기(721)는 블록의 예측 모드를 결정하고, 그 예측 모드에 기초하여 스위치(726)에 제어 신호를 제공한다. 예를 들어, 예측 모드가 인트라 모드인 경우, 일반 제어기(721)는 잔차 계산기(723)에 의해 사용할 인트라 모드 결과를 선택하도록 스위치(726)를 제어하고, 인트라 예측 정보를 선택하고 비트스트림에 인트라 예측 정보를 포함시키도록 엔트로피 인코더(725)를 제어하고; 블록에 대한 예측 모드가 인터 모드인 경우, 일반 제어기(721)는 잔차 계산기(723)에 의해 사용할 인터 예측 결과를 선택하도록 스위치(726)를 제어하고, 인터 예측 정보를 선택하고 비트스트림에 인터 예측 정보를 포함시키도록 엔트로피 인코더(725)를 제어한다.

[0084] 잔차 계산기(723)는 수신된 블록과 인트라 인코더(722) 또는 인터 인코더(730)로부터 선택된 블록에 대한 예측 결과 사이의 차이(잔차 데이터)를 계산하도록 구성될 수 있다. 잔차 인코더(724)는 잔차 데이터를 인코딩하여 변환 계수를 생성하도록 구성될 수 있다. 예를 들어, 잔차 인코더(724)는 잔차 데이터를 공간 도메인에서 주파수 도메인으로 변환하도록 그리고 변환 계수를 생성하도록 구성될 수 있다. 그 다음, 변환 계수는 양자화 처리를 거쳐 양자화된 변환 계수를 얻는다. 다양한 예시적인 실시예에서, 비디오 인코더(703)는 또한 잔차 디코더(728)를 포함한다. 잔차 디코더(728)는 역변환을 수행하고 디코딩된 잔차 데이터를 생성하도록 구성된다. 디코딩된 잔차 데이터는 인트라 인코더(722) 및 인터 인코더(730)에 의해 적절하게 사용될 수 있다. 예를 들어, 인터 인코더(730)는 디코딩된 잔차 데이터 및 인터 예측 정보에 기초하여 디코딩된 블록을 생성할 수 있고, 인트라 인코더(722)는 디코딩된 잔차 데이터 및 인트라 예측 정보에 기초하여 디코딩된 블록을 생성할 수 있다. 디코딩된 블록은 디코딩된 픽처를 생성하도록 적절하게 처리되고, 디코딩된 픽처는 (도시되지 않은) 메모리 회

로에 버퍼링될 수 있고 참조 픽처로서 사용될 수 있다.

- [0085] 엔트로피 인코더(725)는 인코딩된 블록을 포함하도록 비트스트림을 포맷화하고 엔트로피 코딩을 수행하도록 구성될 수 있다. 엔트로피 인코더(725)는 비트스트림에 다양한 정보를 포함하도록 구성된다. 예를 들어, 엔트로피 인코더(725)는 비트스트림에 일반 제어 데이터, 선택된 예측 정보(예컨대, 인트라 예측 정보 또는 인터 예측 정보), 잔차 정보 및 다른 적절한 정보를 포함하도록 구성될 수 있다. 인터 모드 또는 양방향 예측 모드의 병합 하위 모드에서 블록을 코딩할 때는, 잔차 정보가 없을 수 있다.
- [0086] 도 8은 본 개시내용의 다른 실시예에 따른 예시적인 비디오 디코더(810)의 도면을 도시한다. 비디오 디코더(810)는 코딩된 비디오 시퀀스의 일부인 코딩된 픽처를 수신하고, 코딩된 픽처를 디코딩하여 재구성된 픽처를 생성하도록 구성된다. 일례로, 비디오 디코더(810)는 도 4의 예에서 비디오 디코더(410) 대신에 사용될 수 있다.
- [0087] 도 8의 예에서, 비디오 디코더(810)는 도 8의 예시적인 배열에 도시된 바와 같이 함께 결합되는 엔트로피 디코더(871), 인터 디코더(880), 잔차 디코더(873), 재구성 모듈(874) 및 인트라 디코더(872)를 포함한다.
- [0088] 엔트로피 디코더(871)는 코딩된 픽처가 구성되는 선택스 엘리먼트를 나타내는 특정 심벌을 코딩된 픽처로부터 재구성하도록 구성될 수 있다. 이러한 심벌은 예를 들어, 블록이 코딩되는 모드(예컨대, 인트라 모드, 인터 모드, 양방향 예측 모드, 병합 하위 모드 또는 다른 하위 모드), 인트라 디코더(872) 또는 인터 디코더(880)에 의해 예측에 사용되는 특정 샘플 또는 메타데이터를 식별할 수 있는 예측 정보(예컨대, 인트라 예측 정보 또는 인터 예측 정보), 예를 들어 양자화된 변환 계수 형태의 잔차 정보 등을 포함할 수 있다. 일례로, 예측 모드가 인터 또는 양방향 예측 모드인 경우, 인터 예측 정보가 인터 디코더(880)에 제공되고; 예측 타입이 인트라 예측 타입인 경우, 인트라 예측 정보가 인트라 디코더(872)에 제공된다. 잔차 정보는 역양자화를 거칠 수 있고 잔차 디코더(873)에 제공된다.
- [0089] 인터 디코더(880)는 인터 예측 정보를 수신하고, 인터 예측 정보에 기초하여 인터 예측 결과를 생성하도록 구성될 수 있다.
- [0090] 인트라 디코더(872)는 인트라 예측 정보를 수신하고, 인트라 예측 정보에 기초하여 예측 결과를 생성하도록 구성될 수 있다.
- [0091] 잔차 디코더(873)는 역양자화를 수행하여 역양자화된 변환 계수를 추출하고, 역양자화된 변환 계수를 처리하여 주파수 도메인에서 공간 도메인으로 잔차를 변환하도록 구성될 수 있다. 잔류 디코더(873)는 또한 (양자화기 파라미터(QP: Quantizer Parameter)를 포함하도록) 특정 제어 정보를 이용할 수 있으며, 이 정보는 엔트로피 디코더(871)에 의해 제공될 수 있다(데이터 경로는 단지 낮은 데이터 볼륨 제어 정보일 수 있으므로 이는 도시되지 않음).
- [0092] 재구성 모듈(874)은 공간 도메인에서 잔차 디코더(873)에 의해 출력된 잔차와 (일부 경우에 인터 또는 인트라 예측 모듈에 의해 출력된) 예측 결과를 조합하여, 재구성된 픽처의 일부를 재구성된 비디오의 일부로서 형성하는 재구성된 블록을 형성하도록 구성될 수 있다. 블록 분리(deblocking) 동작과 같은 다른 적절한 동작이 시각적 품질을 개선하기 위해 또한 수행될 수 있다는 점이 주목된다.
- [0093] 비디오 인코더(403, 603, 703) 및 비디오 디코더(410, 510, 810)는 임의의 적절한 기법을 사용하여 구현될 수 있다는 점이 주목된다. 일부 예시적인 실시예에서, 비디오 인코더(403, 603, 703) 및 비디오 디코더(410, 510, 810)는 하나 이상의 집적 회로를 사용하여 구현될 수 있다. 다른 실시예에서, 비디오 인코더(403, 603, 603) 및 비디오 디코더(410, 510, 810)는 소프트웨어 명령을 실행하는 하나 이상의 프로세서를 사용하여 구현될 수 있다.
- [0094] 일부 예시적인 구현에서, 인코딩 아티팩트(artifact)를 감소시키고 디코딩된 픽처의 품질을 개선하기 위해 루프 필터가 인코더 및 디코더에 포함될 수 있다. 예를 들어, 루프 필터(555)는 도 5의 디코더(530)의 일부로서 포함될 수 있다. 다른 예를 들어, 루프 필터는 도 6의 인코더(620)에서 임베드된 디코더 유닛(633)의 일부일 수 있다. 이러한 필터는 루프 필터로 지칭되는데, 이들은 디코더 또는 인코더에서 비디오 블록에 대한 디코딩 루프에 포함되기 때문이다. 각각의 루프 필터는 하나 이상의 필터링 파라미터와 연관될 수 있다. 이러한 필터링 파라미터는 미리 정의될 수 있거나 인코딩 프로세스 동안 인코더에 의해 도출될 수 있다. (인코더에 의해 도출된다면) 이러한 필터링 파라미터 또는 (미리 정의된다면) 이들의 인덱스는 인코딩된 형태로 최종 비트스트림에 포함될 수 있다. 그 다음, 디코더는 비트스트림으로부터 이러한 필터링 파라미터를 파싱하고 디코딩 동안 파싱

된 필터링 파라미터에 기초하여 루프 필터링을 수행할 수 있다.

[0095] 상이한 양상에서 아티팩트를 감소시키고 디코딩된 비디오 품질을 개선하기 위해 다양한 루프 필터가 사용될 수 있다. 이러한 루프 필터는 하나 이상의 블록 분리 필터, 적응 루프 필터(ALF), 성분 간 적응 루프 필터(CC-ALF: Cross-Component Adaptive Loop Filter), 제한된 방향 향상 필터(CDEF), 샘플 적응 오프셋(SAO: Sample Adaptive Offset) 필터, 성분 간 샘플 오프셋(CCSO) 필터 및 로컬 샘플 오프셋(LSO) 필터를 포함할 수 있지만 이에 제한되는 것은 아니다. 이러한 필터는 상호 의존적일 수도 또는 아닐 수도 있다. 이들은 디코더 또는 인코더의 디코딩 루프에 (존재한다면) 이들의 상호 의존성과 호환되는 임의의 적절한 순서로 배열될 수 있다. 이러한 다양한 루프 필터는 아래의 개시내용에서 더 상세히 설명된다.

[0096] 블록 기반 필터 적응을 이용한 적응 루프 필터(ALF)가 인코더/디코더에 의해 적용되어 아티팩트를 줄일 수 있다. ALF는 필터링 계수/파라미터 또는 이들의 인덱스가 비트스트림에서 시그널링되고 재구성된 픽처의 이미지 콘텐츠 및 왜곡을 기반으로 설계될 수 있다는 점에서 적응적이다. ALF는 인코딩 프로세스에 의해 유도되는 왜곡을 줄이고 복원된 이미지 품질을 개선하도록 적용될 수 있다.

[0097] 루마 성분에 대해, 예를 들어, 로컬 변화도의 방향 및 활동에 기초하여 루마 블록(예컨대, 4×4 루마 블록)에 대해 복수의 필터(예컨대, 25개의 필터) 중 하나가 선택될 수 있다. 이러한 필터의 필터 계수는 인코딩 프로세스 동안 인코더에 의해 도출될 수 있고 비트스트림으로 디코더에 시그널링될 수 있다.

[0098] ALF는 임의의 적절한 형상과 크기를 가질 수 있다. 도 9의 예를 참조하면, ALF(910-911)는 ALF(910)의 경우 5×5 다이아몬드 형상 그리고 ALF(911)의 경우 7×7 다이아몬드 형상과 같은 다이아몬드 형상을 가질 수 있다. ALF(910)에서 13개의 엘리먼트(920-932)가 필터링 프로세스에 사용되어 다이아몬드 모양을 형성할 수 있다. 7개의 값(예컨대, C0-C6)이 13개의 엘리먼트(920-932)에 대해 설명되는 예시적인 방식으로 사용되고 배열될 수 있다. ALF(911)에서, 25개의 엘리먼트(940-964)가 필터링 프로세스에 사용되어 다이아몬드 모양을 형성할 수 있다. 13개의 값(예컨대, C0-C12)이 설명되는 예시적인 방식으로 25개의 엘리먼트(940-964)에 대해 사용될 수 있다.

[0099] 도 9를 참조하면, 일부 예들에서, 2개의 다이아몬드 형상(910-911) 중 하나의 다이아몬드 형상의 ALF 필터가 루마 또는 크로마 블록을 처리하기 위해 선택될 수 있다. 예를 들어, 5×5 다이아몬드형 필터(910)가 크로마 성분(예컨대, 크로마 블록, 크로마 CB)에 적용될 수 있고, 7×7 다이아몬드형 필터(911)가 루마 성분(예컨대, 루마 블록, 루마 CB)에 적용될 수 있다. 다른 적절한 형상(들)과 크기(들)가 ALF에 사용될 수 있다. 예를 들어, 9×9 다이아몬드형 필터가 사용될 수 있다.

[0100] 값(예컨대, (910)에서 C0-C6 또는 (920)에서 C0-C12)에 의해 지시된 위치에서의 필터 계수는 0이 아닐 수 있다. 또한 ALF가 클리핑(clipping) 기능을 포함하는 경우, 해당 위치의 클리핑 값은 0이 아닐 수 있다. 클리핑 기능은 루마 또는 크로마 블록에서 필터 값의 상한을 제한하는 데 사용될 수 있다.

[0101] 일부 구현들에서, 루마 성분의 특정 블록에 적용될 특정 ALF는 루마 블록의 분류에 기초할 수 있다. 루마 성분의 블록 분류를 위해, 4×4 블록(또는 루마 블록, 루마 CB)이 예컨대, 25개의 상이한 ALF(예컨대, 상이한 필터 계수를 갖는 7×7 ALF 중 25개)에 대응하는 다수(예컨대, 25개)의 클래스 중 하나로 카테고리화 또는 분류될 수 있다. 식(1)을 사용하여 방향성 파라미터 D와 활동 값 A의 양자화된 값  $\hat{A}$ 를 기초로 분류 인덱스 C가 도출될 수 있다.

[0102] 
$$C = 5D + \hat{A} \quad \text{식(1)}$$

[0103] 방향성 파라미터 D와 양자화된 값  $\hat{A}$ 를 계산하기 위해, 다음과 같이 1-D 라플라시안(Laplacian)을 사용하여 수평 및 2개의 대각선 방향(예컨대, d1 및 d2)의 변화도  $g_v$ ,  $g_h$ ,  $g_{d1}$  및  $g_{d2}$ 가 각각 계산될 수 있다.

[0104] 
$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)| \quad \text{식(2)}$$

[0105] 
$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)| \quad \text{식(3)}$$

[0106] 
$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad \text{식(4)}$$

[0107] 
$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{j=j-2}^{j+3} D2_{k,l}, D2_{k,l} = |2R(k, l) - R(k-1, l+1) - R(k+1, l-1)| \quad \text{식(5)}$$

[0108] 여기서 인덱스(i, j)는 4×4 블록 내의 상부 왼쪽 샘플의 좌표를 의미하고, R(k, l)은 좌표 (k, l)에서 채구된 샘플을 지시한다. 방향(예컨대, d1 및 d2)은 2개의 대각선 방향을 나타낸다.

[0109] 위에서 설명된 블록 분류의 복잡성을 줄이기 위해, 서브샘플링된 1-D 라플라시안 계산이 적용될 수 있다. 도 10a - 도 10d는 수직(도 10a), 수평(도 10b) 및 2개의 대각선 방향(d1)(도 10c) 및 d2(도 10d)의 변화도  $g_v$ ,  $g_h$ ,  $g_{d1}$  및  $g_{d2}$ 를 각각 계산하는 데 사용되는 서브샘플링된 포지션의 예를 도시한다. 도 10a에서, 라벨 'V'는 수직 변화도  $g_v$ 를 계산하기 위한 서브샘플링된 포지션을 도시한다. 도 10b에서, 라벨 'H'는 수평 변화도  $g_h$ 를 계산하기 위한 서브샘플링된 포지션을 도시한다. 도 10c에서, 라벨 'D1'은 d1 대각선 변화도  $g_{d1}$ 을 계산하기 위한 서브샘플링된 포지션을 도시한다. 도 10d에서, 라벨 'D2'는 d2 대각선 변화도  $g_{d2}$ 를 계산하기 위한 서브샘플링된 포지션을 도시한다. 도 10a 및 도 10b는 동일한 서브샘플링된 포지션이 상이한 방향의 변화도 계산에 사용될 수 있음을 보여준다. 일부 다른 구현에서, 모든 방향에 대해 상이한 서브샘플링 방식이 사용될 수 있다. 또 다른 일부 구현에서, 상이한 방향에 대해 상이한 서브샘플링 방식이 사용될 수 있다.

[0110] 수평 및 수직 방향의 변화도  $g_v$  및  $g_h$ 의 최대값  $g_{h,v}^{max}$  및 최소값  $g_{h,v}^{min}$ 은 다음과 같이 설정될 수 있다:

[0111] 
$$g_{h,v}^{max} = \max(g_h, g_v), g_{h,v}^{min} = \min(g_h, g_v) \quad \text{식(6)}$$

[0112] 2개의 대각선 방향의 변화도  $g_{d1}$  및  $g_{d2}$ 의 최대값  $g_{d1,d2}^{max}$  및 최소값  $g_{d1,d2}^{min}$ 은 다음과 같이 설정될 수 있다:

[0113] 
$$g_{d1,d2}^{max} = \max(g_{d1}, g_{d2}), g_{d1,d2}^{min} = \min(g_{d1}, g_{d2}) \quad \text{식(7)}$$

[0114] 위의 값과 2개의 임계치  $t_1$  및  $t_2$ 를 기초로 방향성 파라미터 D가 아래와 같이 도출될 수 있다.

[0115] 단계 1. (1)  $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$  그리고 (2)  $g_{d1,d2}^{max} \leq t_1 \cdot g_{d1,d2}^{min}$ 이 참이라면, D는 0으로 설정된다.

[0116] 단계 2.  $g_{h,v}^{max} / g_{h,v}^{min} > g_{d1,d2}^{max} / g_{d1,d2}^{min}$ 이라면, 단계 3으로 계속되고; 그렇지 않으면 단계 4로 계속된다.

[0117] 단계 3.  $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ 이라면, D는 2로 설정되고; 그렇지 않으면 D는 1로 설정된다.

[0118] 단계 4.  $g_{d1,d2}^{max} > t_2 \cdot g_{d1,d2}^{min}$ 이라면, D는 4로 설정되고; 그렇지 않으면 D는 3으로 설정된다.

[0119] 즉, 방향성 파라미터 D는 도 10e에 예시된 바와 같이, 여러 이산 레벨로 표시되고 수평 방향과 수직 방향의 그리고 두 대각선 방향 사이의 루마 블록에 대한 변화도 값 확산을 기반으로 결정된다.

[0120] 활동 값 A는 다음과 같이 계산될 수 있다:

[0121] 
$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}) \quad \text{식(8)}$$

[0122] 수평 및 수직 1-D 라플라시안의 복합 측정치를 나타낸다. 루마 블록에 대한 활동 값 A는 예를 들어, 0부터 4까지의 범위로 추가 양자화될 수 있으며, 양자화된 값은  $\hat{A}$ 로 표기된다.

[0123] 루마 성분에 대해, 위에서 계산된 분류 인덱스 C는 다음에, 다이아몬드형 AFL 필터의 다수의 클래스(예컨대, 25개의 클래스) 중 하나를 선택하는 데 사용될 수 있다. 일부 구현에서, 픽처 내의 크로마 성분에 대해서는, 어떠한 블록 분류도 적용되지 않을 수 있으며, 따라서 각각의 크로마 성분에 대해 단일 세트의 ALF 계수가 적용될 수 있다. 이러한 구현에서, 크로마 성분에 대해 이용 가능한 다수의 ALF 계수 세트들이 있을 수 있지만, ALF 계수의 결정은 크로마 블록의 어떠한 분류에도 의존하지 않을 수 있다.

[0124] 기하학적 변환이 필터 계수 및 (클리핑 값으로도 또한 지칭되는) 대응하는 필터 클리핑 값에 적용될 수 있다. 블록(예컨대, 4x4 루마 블록)을 필터링하기 전에, 예를 들어, 블록에 대해 계산된 변화도 값(예컨대,  $g_v$ ,  $g_h$ ,  $g_{d1}$  및/또는  $g_{d2}$ )에 따라, 필터 계수 f(k, l) 및 대응하는 필터 클리핑 값 c(k, l)에 회전 또는 대각선 및 수직



플립과 같은 기하학적 변환이 적용될 수 있다. 필터 계수  $f(k, l)$  및 대응하는 필터 클리핑 값  $c(k, l)$ 에 적용된 기하학적 변환은 필터에 의해 지원되는 구역 내의 샘플에 기하학적 변환을 적용하는 것과 동일할 수 있다. 기하학적 변환은 각각의 방향성을 정렬함으로써 ALF가 적용되는 다른 블록을 더 유사하게 만들 수 있다.

대각선 플립, 수직 플립 및 회전을 포함하는 세 가지 기하학적 변환 옵션은 각각 식(9) - 식(11)에 의해 설명된 대로 수행될 수 있다.

$$f_D(k, l) = f(l, k), c_D(k, l) = c(l, k), \quad \text{식(9)}$$

$$f_V(k, l) = f(k, K - l - 1), c_V(k, l) = c(k, K - l - 1) \quad \text{식(10)}$$

$$f_R(k, l) = f(K - l - 1, k), c_R(k, l) = c(K - l - 1, k) \quad \text{식(11)}$$

여기서  $K$ 는 ALF 또는 필터의 크기를 나타내고  $0 \leq k, l \leq K - 1$ 은 계수의 좌표이다. 예를 들어, 위치(0, 0)는 왼쪽 상부 모서리에 있고, 위치( $K - 1, K - 1$ )는 필터  $f$  또는 클리핑 값 행렬(또는 클리핑 행렬)  $c$ 의 오른쪽 하단 모서리에 있다. 블록에 대해 계산된 변화도 값에 따라, 변환이 필터 계수 ( $f(k, l)$ ) 및 클리핑 값  $c(k, l)$ 에 적용될 수 있다. 변환과 4개의 변화도 사이의 관계의 일례가 표 1에 요약되어 있다.

표 1: 블록에 대해 계산된 변화도 및 변환의 매핑

표 1

변화도 값	변환
$g_{d2} < g_{d1}$ 및 $g_h < g_v$	변환 없음
$g_{d2} < g_{d1}$ 및 $g_v < g_h$	대각선 플립
$g_{d1} < g_{d2}$ 및 $g_h < g_v$	수직 플립
$g_{d1} < g_{d2}$ 및 $g_v < g_h$	회전

일부 실시예에서, 인코더에 의해 도출된 ALF 필터 파라미터는 픽처에 대한 적응 파라미터 세트(APS: Adaptation Parameter Set)에서 시그널링될 수 있다. APS에서, 루마 필터 계수 및 클리핑 값 인덱스의 하나 이상의 세트(예컨대, 최대 25개의 세트)가 시그널링될 수 있다. 이들은 APS에서 인덱싱될 수 있다. 일례로, 하나 이상의 세트 중 하나의 세트는 루마 필터 계수 및 하나 이상의 클리핑 값 인덱스를 포함할 수 있다. 크로마 필터 계수 및 클리핑 값 인덱스의 하나 이상의 세트(예컨대, 최대 8개의 세트)가 인코더에 의해 도출되고 시그널링될 수 있다. 시그널링 오버헤드를 줄이기 위해, 루마 성분에 대한(예컨대, 상이한 분류 인덱스를 갖는) 상이한 분류의 필터 계수가 병합될 수 있다. 슬라이스 헤더에는, 현재 슬라이스에 사용되는 APS의 인덱스가 시그널링될 수 있다.

다른 예에서, ALF의 시그널링은 CTU 기반일 수 있다.

일 실시예에서, (클리핑 인덱스로도 또한 지칭되는) 클리핑 값 인덱스는 APS로부터 디코딩될 수 있다. 클리핑 값 인덱스는 예를 들어, 클리핑 값 인덱스와 대응하는 클리핑 값 사이의 관계에 기초하여 대응하는 클리핑 값을 결정하는 데 사용될 수 있다. 관계는 미리 정의되어 디코더에 저장될 수 있다. 일례로, 관계는 루마 성분에 대한 클리핑 값 인덱스 및 대응하는 클리핑 값의(예컨대, 루마 CB에 사용되는) 표, 및 크로마 성분에 대한 클리핑 값 인덱스 및 대응하는 클리핑 값의(예컨대, 크로마 CB에 사용되는) 표와 같은 하나 이상의 표에 의해 설명된다. 클리핑 값은 비트 심도  $B$ 에 종속될 수 있다. 비트 심도  $B$ 는 내부 비트 심도, 필터링될 CB에서 재구성된 샘플의 비트 심도 등을 의미할 수 있다. 일부 예에서, 식(12)을 사용하여(예컨대, 루마에 대한 그리고/또는 크로마에 대한) 클리핑 값들의 표가 획득될 수 있다.  $\text{AlfClip} = \{\text{round}(2^{B-a*n}) \text{ for } n \in [0 \dots N - 1]\}$ , 식(12)

여기서  $\text{AlfClip}$ 은 클리핑 값이고,  $B$ 는 비트 심도(예컨대,  $\text{bitDepth}$ )이고,  $N$ (예컨대,  $N = 4$ )은 허용된 클리핑 값의 수이고,  $a$ 는 미리 정의된 상수 값이다. 일례로,  $a$ 는 2.35와 같다.  $n$ (클리핑 인덱스 또는  $\text{clipIdx}$ 로도 또한 지칭되는) 클리핑 값 인덱스이다. 표 2는  $N = 4$ 인 식(12)을 사용하여 얻어진 표의 일례를 도시한다. 클리핑 인덱스  $n$ 은 표 2에서 0, 1, 2 및 3( $N-1$ 까지)이 될 수 있다. 표 2는 루마 블록 또는 크로마 블록에 사용될

수 있다.

표 2 - AlfClip은 비트 심도  $B$  및 clipIdx에 좌우될 수 있다.

표 2

bitDepth	clipIdx			
	0	1	2	3
8	255	64	16	4
9	511	108	23	5
10	1023	181	32	6
11	2047	304	45	7
12	4095	512	64	8
13	8191	861	91	10
14	16383	1448	128	11
15	32767	2435	181	13
16	65535	4096	256	16

현재 슬라이스에 대한 슬라이스 헤더에서, 현재 슬라이스에 사용될 수 있는 루마 필터 세트를 지정하기 위해 하나 이상의 APS 인덱스(예컨대, 최대 7개의 APS 인덱스)가 시그널링될 수 있다. 필터링 프로세스는 픽처 레벨, 슬라이스 레벨, CTB 레벨 등과 같은 하나 이상의 적절한 레벨로 제어될 수 있다. 예시적인 실시예에서, 필터링 프로세스는 CTB 레벨로 제어될 수 있다. ALF가 루마 CTB에 적용되는지 여부를 지시하기 위해 플래그가 시그널링될 수 있다. 루마 CTB는 APS에서 시그널링되는 복수의 고정 필터 세트(예컨대, 16개의 고정 필터 세트) 및 필터 세트(들)(예컨대, 위에서 설명된 바와 같이 인코더에 의해 도출되고 시그널링된 필터 세트(들)로도 또한 지칭되는 최대 25개의 필터) 중에서 필터 세트를 선택할 수 있다. 적용될 필터 세트(예컨대, 복수의 고정 필터 세트 및 시그널링된 필터 세트(들) 중 필터 세트)를 지시하기 위해 루마 CTB에 대해 필터 세트 인덱스가 시그널링될 수 있다. 복수의 고정 필터 세트는 인코더 및 디코더에서 미리 정의되고 하드 코딩될 수 있으며, 미리 정의된 필터 세트로 지칭될 수 있다. 따라서 미리 정의된 필터 계수는 시그널링될 필요가 없다.

크로마 성분에 대해, 현재 슬라이스에 사용될 크로마 필터 세트를 지시하기 위해 슬라이스 헤더에서 APS 인덱스가 시그널링될 수 있다. CTB 레벨에서, APS에 하나보다 많은 크로마 필터 세트가 있다면 각각의 크로마 CTB에 대해 필터 세트 인덱스가 시그널링될 수 있다.

필터 계수는 128과 동일한 노름(norm)으로 양자화될 수 있다. 곱셈 복잡도를 줄이기 위해, 비중앙 포지션의 계수 값이 -27부터 27-1까지의 범위에 있을 수 있도록 비트스트림 적합성이 적용될 수 있다. 일례로, 중앙 포지션 계수는 비트스트림에서 시그널링되지 않으며 128과 같은 것으로 간주될 수 있다.

일부 실시예에서, 클리핑 인덱스 및 클리핑 값의 선택스 및 시맨틱(semantics)은 다음과 같이 정의된다:  $\text{alf\_luma\_clip\_idx}[\text{sfIdx}][j]$ 는 sfIdx로 지시되는 시그널링된 루마 필터의  $j$ 번째 계수를 곱하기 전에 사용할 클리핑 값의 클리핑 인덱스를 지정하는 데 사용될 수 있다. 비트스트림 적합성의 요건은, sfIdx = 0 내지  $\text{alf\_luma\_num\_filters\_signalled\_minus1}$  그리고  $j = 0$  내지 11인  $\text{alf\_luma\_clip\_idx}[\text{sfIdx}][j]$ 의 값이 예를 들어, 0부터 3까지의 범위에 있을 것이라는 것을 포함할 수 있다.

$\text{filtIdx} = 0$  내지  $\text{NumAlfFilters} - 1$  그리고  $j = 0$  내지 11인 엘리먼트  $\text{AlfClipL}[\text{adaptation\_parameter\_set\_id}][\text{filtIdx}][j]$ 를 갖는 루마 필터 클리핑 값  $\text{AlfClipL}[\text{adaptation\_parameter\_set\_id}]$ 는 BitDepthY와 같게 설정된 bitDepth 및  $\text{alf\_luma\_clip\_idx}[\text{alf\_luma\_coeff\_delta\_idx}[\text{filtIdx}][j]]$ 와 같게 설정된 clipIdx에 따라 표 2에 지정된 바와 같이 도출될 수 있다.

$\text{Alf\_chroma\_clip\_idx}[\text{altIdx}][j]$ 는 인덱스 altIdx와 대체 크로마 필터의  $j$ 번째 계수를 곱하기 전에 사용할 클리핑 값의 클리핑 인덱스를 지정하는 데 사용될 수 있다. 비트스트림 적합성의 요건은 altIdx = 0 내지  $\text{alf\_chroma\_num\_alt\_filters\_minus1}$ ,  $j = 0$  내지 5인  $\text{alf\_chroma\_clip\_idx}[\text{altIdx}][j]$ 의 값이 0부터 3까지의 범위에 있을 것이라는 것을 포함할 수 있다.

$\text{altIdx} = 0$  내지  $\text{alf\_chroma\_num\_alt\_filters\_minus1}$ ,  $j = 0$  내지 5인 엘리먼트  $\text{AlfClipC}[\text{adaptation\_parameter\_set\_id}][\text{altIdx}][j]$ 를 갖는 크로마 필터 클리핑 값  $\text{AlfClipC}[\text{adaptation\_parameter\_set\_id}][\text{altIdx}][j]$ 는 BitDepthC와 같게 설정된 bitDepth 및  $\text{alf\_chroma\_clip\_idx}[\text{alf\_chroma\_coeff\_delta\_idx}[\text{altIdx}][j]]$ 와 같게 설정된 clipIdx에 따라 표 2에 지정된 바와 같이 도출될 수 있다.

adaptation\_parameter\_set\_id ][ altIdx ][ j ]를 갖는 크로마 필터 클리핑 값 AlfClipC[ adaptation\_parameter\_set\_id ][ altIdx ]는 BitDepthC와 함께 설정된 bitDepth 및 alf\_chroma\_clip\_idx[ altIdx ][ j ]와 함께 설정된 clipIdx에 따라 표 2에 지정된 바와 같이 도출될 수 있다.

[0145] 일 실시예에서, 필터링 프로세스는 아래와 같이 설명될 수 있다. 디코더 측에서 ALF가 CTB에 대해 인에이블(enable)되면, CTB의 CU(또는 CB) 내의 샘플  $R(i, j)$ 이 필터링될 수 있어, 식(13)을 사용하여 아래에 도시된 것과 같이 필터링된 샘플 값  $R'(i, j)$ 가 될 수 있다. 일례로, CU의 각각의 샘플이 필터링된다.

$$R'(i, j) = R(i, j) + \left( \left( \sum_{k \neq 0} \sum_{l \neq 0} f(k, l) \times K(R(i+k, j+l) - R(i, j), c(k, l)) + 64 \right) \gg 7 \right) \quad \text{식 (13)}$$

[0147] 여기서  $f(k, l)$ 은 디코딩된 필터 계수를 나타내고,  $K(x, y)$ 는 클리핑 함수이고,  $c(k, l)$ 은 디코딩된 클리핑 파라미터(또는 클리핑 값)를 나타낸다. 변수  $k$ 와  $l$ 은  $-L/2$ 와  $L/2$  사이에서 변할 수 있으며 여기서  $L$ 은 필터 길이를 나타낸다(예컨대, 각각 루마 성분 및 크로마 성분에 대한 도 9의 예시적인 다이아몬드 필터(910, 911)의 경우,  $L=5$  및  $7$ ). 클리핑 함수  $K(x, y) = \min(y, \max(-y, x))$ 는 클리핑 함수 Clip3( $-y, y, x$ )에 대응한다. 클리핑 함수  $K(x, y)$ 를 통합함으로써, 루프 필터링 방법(예컨대, ALF)이 비선형 프로세스가 되며 비선형 ALF로 지칭될 수 있다.

[0148] 선택된 클리핑 값은 다음과 같이 "alf\_data" 선택스 엘리먼트로 코딩될 수 있는데: 적절한 인코딩 방식(예컨대, Golomb 인코딩 방식)이 표 2에 도시된 것과 같이, 선택된 클리핑 값에 대응하는 클리핑 인덱스를 인코딩하는 데 사용될 수 있다. 인코딩 방식은 필터 세트 인덱스를 인코딩하는 데 사용되는 동일한 인코딩 방식일 수 있다.

[0149] 일 실시예에서, 가상 경계 필터링 프로세스가 ALF의 라인 버퍼 요건을 줄이는 데 사용될 수 있다. 이에 따라, 수정된 블록 분류 및 필터링이 CTU 경계(예컨대, 수평 CTU 경계) 근처의 샘플에 대해 이용될 수 있다. 가상 경계(1130)는 수평 CTU 경계(1120)를 도 11a에 도시된 바와 같이 " $N_{\text{samples}}$ "개의 샘플만큼 시프트함으로써 라인으로 정의될 수 있으며, 여기서  $N_{\text{samples}}$ 은 양의 정수일 수 있다. 일례로,  $N_{\text{samples}}$ 은 루마 성분에 대해 4와 같고,  $N_{\text{samples}}$ 은 크로마 성분에 대해 2와 같다.

[0150] 도 11a를 참조하면, 루마 성분에 대해 수정된 블록 분류가 적용될 수 있다. 일례로, 가상 경계(1130) 위의  $4 \times 4$  블록(1110)의 1-D 라플라시안 변화도 계산을 위해, 가상 경계(1130) 위의 샘플만이 사용된다. 유사하게, 도 11b를 참조하면, CTU 경계(1121)로부터 시프트되는 가상 경계(1131) 아래의  $4 \times 4$  블록(1111)의 1-D 라플라시안 변화도 계산을 위해, 가상 경계(1131) 아래의 샘플만이 사용된다. 활동 값 A의 양자화는 1D 라플라시안 변화도 계산에 사용된 감소된 수의 샘플을 고려함으로써 그에 따라 스케일링될 수 있다.

[0151] 필터링 처리를 위해, 가상 경계에서의 대칭 패딩 연산이 루마 성분과 크로마 성분 모두에 대해 사용될 수 있다. 도 12a - 도 12f는 가상 경계에서 루마 성분에 대한 이러한 수정된 ALF 필터링의 예를 예시한다. 필터링되는 샘플이 가상 경계 아래에 위치되는 경우, 가상 경계 위에 위치되는 인접 샘플이 패딩될 수 있다. 필터링되는 샘플이 가상 경계 위에 위치되는 경우, 가상 경계 아래에 위치되는 인접 샘플이 패딩될 수 있다. 도 12a를 참조하면, 가상 경계(1210) 아래에 위치되는 샘플 C2에 이웃 샘플 C0이 패딩될 수 있다. 도 12b를 참조하면, 가상 경계(1220) 위에 위치되는 샘플 C2에 이웃 샘플 C0이 패딩될 수 있다. 도 12c를 참조하면, 가상 경계(1230) 아래에 위치되는 샘플 C5-C7에 각각 이웃 샘플 C1-C3이 패딩될 수 있다. 샘플 C0은 샘플 C6으로 패딩될 수 있다. 도 12d를 참조하면, 가상 경계(1240) 위에 위치되는 샘플 C5-C7에 각각 이웃 샘플 C1-C3이 패딩될 수 있다. 샘플 C0은 샘플 C6으로 패딩될 수 있다. 도 12e를 참조하면, 가상 경계(1250) 아래에 위치되는 샘플 C10, C11, C12, C11 및 C10에 각각 이웃 샘플 C4-C8이 패딩될 수 있다. 샘플 C1-C3은 샘플 C11, C12 및 C11로 패딩될 수 있다. 샘플 C0은 샘플 C12로 패딩될 수 있다. 도 12f를 참조하면, 가상 경계(1260) 위에 위치되는 샘플 C10, C11, C12, C11 및 C10에 각각 이웃 샘플 C4-C8이 패딩될 수 있다. 샘플 C1-C3은 샘플 C11, C12 및 C11로 패딩될 수 있다. 샘플 C0은 샘플 C12로 패딩될 수 있다.

[0152] 일부 예에서, 위의 설명은 샘플(들) 및 이웃 샘플(들)이 가상 경계의 왼쪽(또는 오른쪽) 및 오른쪽(또는 왼쪽)에 위치될 때 적절하게 적용될 수 있다.

[0153] 최대 코딩 유닛(LCU: largest coding unit) 정렬 픽처 쿼드 트리 분할이 사용될 수 있다. 코딩 효율을 향상시키기 위해, 코딩 유닛 동기 픽처 쿼드 트리 기반 적응 루프 필터가 비디오 코딩에 사용될 수 있다. 일례로, 루마 픽처는 다수의 다중 레벨 쿼드 트리 파티션으로 분할될 수 있고, 각각의 파티션 경계는 최대 코딩 유닛(LCU)의 경계에 맞춰 정렬된다. 각각의 파티션은 필터링 프로세스를 가질 수 있으며, 따라서 필터 유닛 또는 필터

링 유닛(FU)으로 지칭될 수 있다.

- [0154] 예시적인 2-패스(pass) 인코딩 흐름이 다음과 같이 설명된다. 첫 번째 패스에서, 쿼드 트리 분할 패턴과 각각의 FU의 최상의 필터(또는 최적의 필터)가 결정될 수 있다. 필터링 왜곡은 결정 프로세스 동안 고속 필터링 왜곡 추정(FFDE: fast filtering distortion estimation) 기법에 의해 추정될 수 있다. 결정된 트리 분할 패턴과 FU(예컨대, 모든 FU)의 선택된 필터에 따라, 재구성된 픽처가 필터링될 수 있다. 두 번째 패스에서, CU 동기 ALF 온/오프 제어가 수행될 수 있다. ALF 온/오프 결과에 따르면, 첫 번째 필터링된 픽처는 재구성된 픽처에 의해 부분적으로 복원된다.
- [0155] 레이트 왜곡 기준을 사용함으로써 픽처를 다중 레벨 쿼드 트리 파티션으로 분할하기 위해 하향식 분할 전략이 채택될 수 있다. 각각의 파티션은 FU로 지칭될 수 있다. 분할 프로세스는 도 13에 도시된 바와 같이, 쿼드 트리 파티션을 LCU 경계와 정렬할 수 있다. 도 13은 본 개시내용의 일 실시예에 따른 LCU 정렬 픽처 쿼드 트리 분할의 일례를 도시한다. 일례로, FU의 인코딩 순서는 z-스캔 순서를 따른다. 예를 들어, 도 13을 참조하면, 픽처는 10개의 FU로 분할되고(예컨대,  $FU_0$ - $FU_9$ , 분할 깊이가 2이면,  $FU_0$ ,  $FU_1$ ,  $FU_9$ 가 첫 번째 레벨의 FU이고,  $FU_8$ ,  $FU_5$ ,  $FU_7$ ,  $FU_6$ 가 두 번째 깊이 레벨의 FU이며,  $FU_3$ - $FU_4$ 이 세 번째 깊이 레벨의 FU임), 인코딩 순서는  $FU_0$ 부터  $FU_9$ , 예컨대  $FU_0$ ,  $FU_1$ ,  $FU_2$ ,  $FU_3$ ,  $FU_4$ ,  $FU_5$ ,  $FU_6$ ,  $FU_7$ ,  $FU_8$ ,  $FU_9$ 이다.
- [0156] 픽처 쿼드 트리 분할 패턴을 지시하기 위해, 분할 플래그(쿼드 트리 분할을 나타내는 "1" 및 쿼드 트리 분할이 없음을 나타내는 "0")가 인코딩되어 z-스캔 순서로 전송될 수 있다. 도 14는 본 개시내용의 일 실시예에 따른, 도 13에 대응하는 쿼드 트리 분할 패턴을 도시한다. 도 14의 예에 도시된 바와 같이, 쿼드 트리 분할 플래그는 z 스캔 순서로 인코딩된다.
- [0157] 각각의 FU의 필터는 레이트 왜곡 기준에 기초하여 2개의 필터 세트들로부터 선택될 수 있다. 첫 번째 세트는 현재 FU에 대해 새로 도출된 1/2 대칭 정사각형 및 마름모 형상 필터를 가질 수 있다. 두 번째 세트는 시간 지연 필터 버퍼에서 가져올 수 있다. 시간 지연 필터 버퍼는 이전 픽처의 FU에 대해 이전에 도출된 필터를 저장할 수 있다. 두 필터 세트의 최소 비율 왜곡 비용을 갖는 필터가 현재 FU에 대해 선택될 수 있다. 유사하게, 현재 FU가 가장 작은 FU가 아니고 4개의 자식 FU로 더 분할될 수 있다면, 4개의 자식 FU의 비율 왜곡 비용이 계산될 수 있다. 분할 경우 및 비분할 경우의 비율 왜곡 비용을 재귀적으로 비교함으로써, 픽처 쿼드 트리 분할 패턴(즉, 현재 FU의 쿼드 트리 분할이 중지되어야 하는지 여부)이 결정될 수 있다.
- [0158] 일부 예들에서, 최대 쿼드 트리 분할 레벨 또는 깊이는 미리 정의된 수로 제한될 수 있다. 예를 들어, 최대 쿼드 트리 분할 레벨 또는 깊이는 2일 수 있고, 따라서 FU의 최대 수는 16(또는 4 내지 최대 깊이 수의 제곱)일 수 있다. 쿼드 트리 분할 결정 중에, 최하위 쿼드 트리 레벨(가장 작은 FU)에서 16개의 FU의 위너(Wiener) 계수를 도출하기 위한 상관 값이 재사용될 수 있다. 나머지 FU는 최하위 쿼드 트리 레벨에서 16개의 FU의 상관으로부터 나머지 FU의 위너 필터를 도출할 수 있다. 따라서 일례로, 모든 FU의 필터 계수를 도출하기 위해 단 하나의 프레임 버퍼 액세스만이 존재한다.
- [0159] 쿼드 트리 분할 패턴이 결정된 후, 필터링 왜곡을 더 줄이기 위해 CU 동기 ALF 온/오프 제어가 수행될 수 있다. 필터링 왜곡과 비필터링 왜곡을 비교함으로써, 리프 CU가 대응하는 로컬 구역에서 ALF를 명시적으로 온/오프 전환할 수 있다. 코딩 효율은 ALF 온/오프 결과에 따라 필터 계수를 재설계함으로써 더 향상될 수 있다. 일례로, 재설계 프로세스는 추가 프레임 버퍼 액세스를 필요로 한다. 따라서 코딩 유닛 동기 픽처 쿼드 트리 기반 적응 루프 필터(CS-PQALF: coding unit synchronous picture quadtree-based adaptive loop filter) 인코더 설계와 같은 일부 예에서, 프레임 버퍼 액세스의 수를 최소화하기 위해 CU 동기 ALF 온/오프 결정 후에는 재설계 프로세스가 필요하지 않다.
- [0160] 성분 간 필터링 프로세스는 성분 간 적응 루프 필터(CC-ALF)와 같은 성분 간 필터를 적용할 수 있다. 성분 간 필터는 크로마 성분(예컨대, 루마 CB에 대응하는 크로마 CB)을 미세화하기 위해 루마 성분(예컨대, 루마 CB)의 루마 샘플 값을 사용할 수 있다. 일례로, 루마 CB 및 크로마 CB는 CU에 포함된다.
- [0161] 도 15는 본 개시내용의 예시적인 실시예에 따라 크로마 성분을 생성하기 위해 사용되는 성분 간 필터(예컨대, CC-ALF)를 도시한다. 예를 들어, 도 15는 첫 번째 크로마 성분(예컨대, 첫 번째 크로마 CB), 두 번째 크로마 성분(예컨대, 두 번째 크로마 CB) 및 루마 성분(예컨대, 루마 CB)에 대한 필터링 프로세스를 도시한다. 루마 성분은 샘플 적응 오프셋(SAO) 필터(1510)에 의해 필터링되어 SAO 필터링된 루마 성분(1541)을 생성할 수 있다. SAO 필터링된 루마 성분(1541)은 ALF 루마 필터(1516)에 의해 추가로 필터링되어 필터링된 루마 CB(1561)(예컨



대, 'Y')가 될 수 있다.

[0162] 첫 번째 크로마 성분은 SAO 필터(1512) 및 ALF 크로마 필터(1518)에 의해 필터링되어 첫 번째 중간 성분(1552)을 생성할 수 있다. 또한, SAO 필터링된 루마 성분(1541)은 첫 번째 크로마 성분에 대한 성분 간 필터(예컨대, CC-ALF)(1521)에 의해 필터링되어 두 번째 중간 성분(1542)을 생성할 수 있다. 이어서, 필터링된 첫 번째 크로마 성분(1562)(예컨대, 'Cb')이 두 번째 중간 성분(1542) 및 첫 번째 중간 성분(1552) 중 적어도 하나에 기초하여 생성될 수 있다. 일례로, 필터링된 첫 번째 크로마 성분(1562)(예컨대, 'Cb')은 두 번째 중간 성분(1542)과 첫 번째 중간 성분(1552)을 가산기(1522)로 결합함으로써 생성될 수 있다. 따라서 첫 번째 크로마 성분에 대한 예시적인 성분 간 적응 루프 필터링 프로세스는 CC-ALF(1521)에 의해 수행되는 단계 및 예를 들어, 가산기(1522)에 의해 수행되는 단계를 포함할 수 있다.

[0163] 위의 설명은 두 번째 크로마 성분에 적용될 수 있다. 두 번째 크로마 성분은 SAO 필터(1514) 및 ALF 크로마 필터(1518)에 의해 필터링되어 세 번째 중간 성분(1553)을 생성할 수 있다. 또한, SAO 필터링된 루마 성분(1541)은 두 번째 크로마 성분에 대한 성분 간 필터(예컨대, CC-ALF)(1531)에 의해 필터링되어 네 번째 중간 성분(1543)을 생성할 수 있다. 이어서, 필터링된 두 번째 크로마 성분(1563)(예컨대, 'Cr')이 네 번째 중간 성분(1543) 및 세 번째 중간 성분(1553) 중 적어도 하나에 기초하여 생성될 수 있다. 일례로, 네 번째 중간 성분(1543)과 세 번째 중간 성분(1553)을 가산기(1532)로 결합함으로써 필터링된 두 번째 크로마 성분(1563)(예컨대, 'Cr')이 생성될 수 있다. 일례로, 두 번째 크로마 성분에 대한 성분 간 적응 루프 필터링 프로세스는 이에 따라 CC-ALF(1531)에 의해 수행되는 단계 및 예를 들어, 가산기(1532)에 의해 수행되는 단계를 포함할 수 있다.

[0164] 성분 간 필터(예컨대, CC-ALF(1521), CC-ALF(1531))는 루마 성분(또는 루마 채널)에 임의의 적절한 필터 형상을 갖는 선형 필터를 적용하여 각각의 크로마 성분(예컨대, 첫 번째 크로마 성분, 두 번째 크로마 성분)을 미세화함으로써 동작할 수 있다. CC-ALF는 다른 색 성분으로부터의 샘플을 기반으로 한 하나의 색 성분의 코딩 왜곡을 줄이기 위해 색 성분 간의 상관을 이용한다.

[0165] 도 16은 본 개시내용의 일 실시예에 따른 CC-ALF 필터(1600)의 일례를 도시한다. 필터(1600)는 0이 아닌 필터 계수 및 0 필터 계수를 포함할 수 있다. 필터(1600)는 (검정색 채움을 갖는 원으로 지시된) 필터 계수(1610)에 의해 형성된 다이아몬드 형상(1620)을 갖는다. 일례로, 필터(1600)의 0이 아닌 필터 계수는 필터 계수(1610)에 포함되고, 필터 계수(1610)에 포함되지 않은 필터 계수는 0이다. 따라서 필터(1600)의 0이 아닌 필터 계수는 다이아몬드 형상(1620)에 포함되고, 다이아몬드 형상(1620)에 포함되지 않은 필터 계수는 0이다. 일례로, 필터(1600)의 필터 계수의 수는 필터 계수(1610)의 수와 동일하며, 이는 도 14에 도시된 예에서 18이다.

[0166] CC-ALF는 (CC-ALF 필터 계수로도 또한 지칭되는) 임의의 적절한 필터 계수를 포함할 수 있다. 다시 도 15를 참조하면, CC-ALF(1521) 및 CC-ALF(1531)는 도 16에 도시된 다이아몬드 형상(1620)과 같은 동일한 필터 형상 및 동일한 수의 필터 계수를 가질 수 있다. 일례로, CC-ALF(1521)의 필터 계수의 값은 CC-ALF(1531)의 필터 계수의 값과 다르다.

[0167] 일반적으로, CC-ALF의 필터 계수(예컨대, 인코더에 의해 도출된 0이 아닌 필터 계수)는 예를 들어 APS에서 전송될 수 있다. 일례로, 필터 계수는 인수(예컨대,  $2^{10}$ )에 의해 스케일링될 수 있고 고정 소수점 표현에 대해 반올림될 수 있다. CC-ALF의 적용은 가변 블록 크기에 대해 제어될 수 있고 샘플의 각각의 블록에 대해 수신된 컨텍스트 코딩된 플래그(예컨대, CC-ALF 인에이블 플래그)에 의해 시그널링될 수 있다. CC-ALF 인에이블 플래그와 같은 컨텍스트 코딩된 플래그는 블록 레벨과 같은 임의의 적절한 레벨에서 시그널링될 수 있다. CC-ALF 인에이블 플래그와 함께 블록 크기는 각각의 크로마 성분에 대한 슬라이스 레벨에서 수신될 수 있다. 일부 예에서, (크로마 샘플에서의) 블록 크기 16x16, 32x32 및 64x64가 지원될 수 있다.

[0168] 일례로, CC-ALF의 선택스 변경은 아래 표 3에서 설명된다.

[0169] 표 3: CC-ALF의 선택스 변경

if( slice_cross_component_alf_cb_enabled_flag )	
<b>alf_ctb_cross_component_cb_idc</b> [ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]	ae(v)
if( slice_cross_component_alf_cb_enabled_flag == 0    alf_ctb_cross_component_cb_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ] == 0 )	
if( slice_alf_chroma_idc == 1    slice_alf_chroma_idc == 3 ) {	
<b>alf_ctb_flag</b> [ 1 ][ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]	ae(v)
if( alf_ctb_flag[ 1 ][ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ] && aps_alf_chroma_num_alt_filters_minus1 > 0 )	
<b>alf_ctb_filter_alt_idx</b> [ 0 ][ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]	ae(v)
}	
if( slice_cross_component_alf_cr_enabled_flag )	
<b>alf_ctb_cross_component_cr_idc</b> [ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]	ae(v)
if( slice_cross_component_alf_cr_enabled_flag == 0    alf_ctb_cross_component_cr_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ] == 0 )	
if( slice_alf_chroma_idc == 2    slice_alf_chroma_idc == 3 ) {	
<b>alf_ctb_flag</b> [ 2 ][ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]	ae(v)
if( alf_ctb_flag[ 2 ][ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ] && aps_alf_chroma_num_alt_filters_minus1 > 0 )	
<b>alf_ctb_filter_alt_idx</b> [ 1 ][ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]	ae(v)
}	

[0170]

[0171] 위의 예시적인 CC-ALF 관련 선택스의 시맨틱은 아래에 설명될 수 있다: 0과 같은 **alf\_ctb\_cross\_component\_cb\_idc**[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]는 성분 간 Cb 필터가 루마 위치( xCtb, yCtb )에서 Cb 색 성분 샘플의 블록에 적용되지 않음을 지시할 수 있다.

[0172] 0과 같지 않은 **alf\_ctb\_cross\_component\_cb\_idc**[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]는 **alf\_ctb\_cross\_component\_cb\_idc**[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]번째 성분 간 Cb 필터가 루마 위치( xCtb, yCtb )에서 Cb 색 성분 샘플의 블록에 적용됨을 지시할 수 있다.

[0173] 0과 같은 **alf\_ctb\_cross\_component\_cr\_idc**[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]는 성분 간 Cr 필터가 루마 위치( xCtb, yCtb )에서의 Cr 색 성분 샘플의 블록에 적용되지 않음을 지시할 수 있다.

[0174] 0과 같지 않은 **alf\_ctb\_cross\_component\_cr\_idc**[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]는 **alf\_ctb\_cross\_component\_cr\_idc**[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]번째 성분 간 Cr 필터가 루마 위치(xCtb, yCtb)에서 Cr 색 성분 샘플의 블록에 적용됨을 지시할 수 있다.

[0175] 크로마 샘플링 포맷의 예가 아래에 설명된다. 일반적으로, 루마 블록은 2개의 크로마 블록과 같은 하나 이상의 크로마 블록에 대응할 수 있다. 크로마 블록(들) 각각에서의 샘플의 수는 루마 블록에서의 샘플의 수보다 적을 수 있다. (예컨대, chroma\_format\_idc에 의해 지정된 크로마 서브샘플링 포맷으로도 또한 지칭되는) 크로마 서브샘플링 포맷은 크로마 블록(들) 각각과 대응하는 루마 블록 사이의 크로마 수평 서브샘플링 인수(예컨대, SubWidthC) 및 크로마 수직 서브샘플링 인수(예컨대, SubHeightC)를 지시할 수 있다. 크로마 서브샘플링 방식은 공칭 4(수평)×4(수직) 블록에 대해 4:x:y 포맷으로 지정될 수 있으며, x는 수평 크로마 서브샘플링 인수(블록의 첫 번째 행에 유지되는 크로마 샘플 수)이고, y는 블록의 두 번째 행에 얼마나 많은 크로마 샘플이 유지되는지이다. 일례로, 크로마 서브샘플링 포맷은 4:2:0일 수 있는데, 이는 크로마 수평 서브샘플링 인수(예컨대, SubWidthC) 및 크로마 수직 서브샘플링 인수(예컨대, SubHeightC)가 둘 다 도 17a - 도 17b에 도시된 바와 같이 2임을 지시한다. 다른 예에서, 크로마 서브샘플링 포맷은 4:2:2일 수 있는데, 이는 크로마 수평 서브샘플링

인수(예컨대, SubWidthC)는 2이고, 크로마 수직 서브샘플링 인수(예컨대, SubHeightC)는 1임을 지시한다. 또 다른 예에서, 크로마 서브샘플링 포맷은 4:4:4일 수 있는데, 이는 크로마 수평 서브샘플링 인수(예컨대, SubWidthC) 및 크로마 수직 서브샘플링 인수(예컨대, SubHeightC)가 1임을 지시한다. 이에 따라, (크로마 샘플 포지션으로도 또한 지칭되는) 크로마 샘플 포맷 또는 타입은 루마 블록 내의 적어도 하나의 대응하는 루마 샘플에 대한 크로마 블록 내의 크로마 샘플의 상대 포지션을 지시할 수 있다.

[0176] 도 17a - 도 17b는 본 개시내용의 실시예에 따른 루마 샘플에 대한 크로마 샘플의 예시적인 위치를 도시한다. 도 17a를 참조하면, 루마 샘플(1701)은 행(1711-1718)에 위치된다. 도 17a에 도시된 루마 샘플(1701)은 픽처의 일부를 나타낼 수 있다. 일례로, 루마 블록(예컨대, 루마 CB)은 루마 샘플(1701)을 포함한다. 루마 블록은 4:2:0의 크로마 서브샘플링 포맷을 갖는 2개의 크로마 블록에 대응할 수 있다. 일례로, 각각의 크로마 블록은 크로마 샘플(1703)을 포함한다. 각각의 크로마 샘플(예컨대, 크로마 샘플(1703(1)))은 4개의 루마 샘플(예컨대, 루마 샘플(1701(1)-1701(4)))에 대응한다. 일례로, 4개의 루마 샘플은 왼쪽 최상부 샘플(1701(1)), 오른쪽 최상부 샘플(1701(2)), 왼쪽 최하부 샘플(1701(3)), 오른쪽 최하부 샘플(1701(4))이다. 크로마 샘플(예컨대, (1703(1)))은 왼쪽 최상부 샘플(1701(1))과 왼쪽 최하부 샘플(1701(3)) 사이의 왼쪽 중앙 포지션에 위치될 수 있으며, 크로마 샘플(1703)을 갖는 크로마 블록의 크로마 샘플 타입은 크로마 샘플 타입 0으로 지칭될 수 있다. 크로마 샘플 타입 0은 왼쪽 최상부 샘플(1701(1)) 및 왼쪽 최하부 샘플(1701(3))의 중간에서 왼쪽 중앙 포지션에 대응하는 상대 포지션 0을 지시한다. 4개의 루마 샘플(예컨대, (1701(1)-1701(4)))은 크로마 샘플(1703)(1)의 이웃 루마 샘플로 지칭될 수 있다.

[0177] 일례로, 각각의 크로마 블록은 크로마 샘플(1704)을 포함할 수 있다. 크로마 샘플(1703)을 참조로 한 위의 설명은 크로마 샘플(1704)에 적용될 수 있으며, 따라서 간결함을 위해 상세한 설명은 생략될 수 있다. 크로마 샘플(1704) 각각은 4개의 대응하는 루마 샘플의 중앙 포지션에 위치될 수 있으며, 크로마 샘플(1704)을 갖는 크로마 블록의 크로마 샘플 타입은 크로마 샘플 타입 1로 지칭될 수 있다. 크로마 샘플 타입 1은 4개의 루마 샘플(예컨대, (1701(1)-1701(4)))의 중앙 포지션에 대응하는 상대 포지션 1을 지시한다. 예를 들어, 크로마 샘플(1704) 중 하나는 루마 샘플(1701(1)-1701(4))의 중앙 부분에 위치될 수 있다.

[0178] 일례로, 각각의 크로마 블록은 크로마 샘플(1705)을 포함한다. 크로마 샘플(1705) 각각은 4개의 대응하는 루마 샘플(1701)의 왼쪽 최상부 샘플과 같이 위치되는 왼쪽 최상부 포지션에 위치될 수 있으며, 크로마 샘플(1705)을 갖는 크로마 블록의 크로마 샘플 타입은 크로마 샘플 타입 2로 지칭될 수 있다. 이에 따라, 크로마 샘플(1705) 각각은 각각의 크로마 샘플에 대응하는 4개의 루마 샘플(1701)의 왼쪽 최상부 샘플과 같이 위치된다. 크로마 샘플 타입 2는 4개의 루마 샘플(1701)의 최상부 왼쪽 포지션에 대응하는 상대 포지션 2를 지시한다. 예를 들어, 크로마 샘플(1705) 중 하나는 루마 샘플(1701(1)-1701(4))의 최상부 왼쪽 포지션에 위치될 수 있다.

[0179] 일례로, 각각의 크로마 블록은 크로마 샘플(1706)을 포함한다. 크로마 샘플(1706) 각각은 대응하는 최상부 왼쪽 샘플과 대응하는 최상부 오른쪽 샘플 사이의 최상부 중앙 포지션에 위치될 수 있으며, 크로마 샘플(1706)을 갖는 크로마 블록의 크로마 샘플 타입은 크로마 샘플 타입 3으로 지칭될 수 있다. 크로마 샘플 타입 3은 왼쪽 최상부 샘플과 오른쪽 최상부 샘플 사이의 최상부 중앙 포지션에 대응하는 상대 포지션 3을 지시한다. 예를 들어, 크로마 샘플(1706) 중 하나는 루마 샘플(1701(1)-1701(4))의 최상부 중앙 포지션에 위치될 수 있다.

[0180] 일례로, 각각의 크로마 블록은 크로마 샘플(1707)을 포함한다. 크로마 샘플(1707) 각각은 4개의 대응하는 루마 샘플(1701)의 왼쪽 최하부 샘플과 같이 위치되는 왼쪽 최하부 포지션에 위치될 수 있으며, 크로마 샘플(1707)을 갖는 크로마 블록의 크로마 샘플 타입은 크로마 샘플 타입 4로 지칭될 수 있다. 이에 따라, 크로마 샘플(1707) 각각은 각각의 크로마 샘플에 대응하는 4개의 루마 샘플(1701)의 왼쪽 최하부 샘플과 같이 위치된다. 크로마 샘플 타입 4는 4개의 루마 샘플(1701)의 최하부 왼쪽 포지션에 대응하는 상대 포지션 4를 지시한다. 예를 들어, 크로마 샘플(1707) 중 하나는 루마 샘플(1701(1)-1701(4))의 최하부 왼쪽 포지션에 위치될 수 있다.

[0181] 일례로, 각각의 크로마 블록은 크로마 샘플(1708)을 포함한다. 크로마 샘플(1708) 각각은 최하부 왼쪽 샘플과 최하부 오른쪽 샘플 사이의 최하부 중앙 포지션에 위치되며, 크로마 샘플(1708)을 갖는 크로마 블록의 크로마 샘플 타입은 크로마 샘플 타입 5로 지칭될 수 있다. 크로마 샘플 타입 5는 4개의 루마 샘플(1701)의 최하부 왼쪽 샘플과 최하부 오른쪽 샘플 사이의 최하부 중앙 포지션에 대응하는 상대 포지션 5를 지시한다. 예를 들어, 크로마 샘플(1708) 중 하나는 루마 샘플(1701(1)-1701(4))의 최하부 왼쪽 샘플과 최하부 오른쪽 샘플 사이에 위치될 수 있다.

[0182] 일반적으로, 임의의 적절한 크로마 샘플 타입이 크로마 서브샘플링 포맷에 사용될 수 있다. 크로마 샘플 타입 0-5는 서브샘플링 포맷 4:2:0으로 설명된 예시적인 크로마 샘플 타입을 제공한다. 크로마 서브샘플링 포맷

4:2:0에 대해 추가 크로마 샘플 타입이 사용될 수 있다. 또한, 다른 크로마 샘플 타입 및/또는 크로마 샘플 타입 0-5의 변형이 4:2:2, 4:4:4 등과 같은 다른 크로마 서브샘플링 포맷에 사용될 수 있다. 일례로, 크로마 샘플(1705, 1707)을 조합하는 크로마 샘플 타입이 크로마 서브샘플링 포맷 4:2:2에 사용될 수 있다.

[0183] 다른 예에서, 루마 블록은 4개의 루마 샘플(예컨대, (1701(1)-1701(4)))의 최상부 2개의 샘플(예컨대, (1701(1)-1701(2))) 및 4개의 루마 샘플(예컨대, (1701(1)-1701(4)))의 최하부 2개의 샘플(예컨대, (1701(3)-1701(4)))을 각각 포함하는 행(1711-1712)과 같은 교대 행을 갖는 것으로 간주된다. 이에 따라, 행(1711, 1713, 1715, 1717)은 (최상부 필드로도 또한 지칭되는) 현재 행으로 지칭될 수 있고, 행(1712, 1714, 1716, 1718)은 (최하부 필드로도 또한 지칭되는) 다음 행으로 지칭될 수 있다. 4개의 루마 샘플(예컨대, (1701(1)-1701(4)))이 현재 행(예컨대, (1711))과 다음 행(예컨대, (1712))에 위치된다. 위의 상대 크로마 포지션 2-3은 현재 행에 위치되며, 위의 상대 크로마 포지션 0-1은 각각의 현재 행과 각각의 다음 행 사이에 위치되고, 위의 상대 크로마 포지션 4-5는 다음 행에 위치된다.

[0184] 크로마 샘플(1703, 1704, 1705, 1706, 1707 또는 1708)은 각각의 크로마 블록의 행(1751-1754)에 위치된다. 행(1751-1754)의 특정 위치는 크로마 샘플의 크로마 샘플 타입에 좌우될 수 있다. 예를 들어, 각각의 크로마 샘플 타입 0-1을 갖는 크로마 샘플(1703-1704)에 대해, 행(1751)은 행(1711-1712) 사이에 위치된다. 각각의 크로마 샘플 타입 2-3을 갖는 크로마 샘플(1705-1706)에 대해, 행(1751)은 현재 행(1711)과 같이 위치된다. 각각의 크로마 샘플 타입 4-5를 갖는 크로마 샘플(1707-1708)에 대해, 행(1751)은 다음 행(1712)과 같이 위치된다. 위의 설명은 행(1752-1754)에 적합하게 적응될 수 있으며, 간결함을 위해 상세한 설명은 생략된다.

[0185] 임의의 적절한 스캐닝 방법이 도 17a에서 앞서 설명한 루마 블록 및 대응하는 크로마 블록(들)을 디스플레이, 저장 및/또는 전송하기 위해 사용될 수 있다. 일부 예시적인 구현에서, 순차 스캐닝이 사용될 수 있다.

[0186] 대안으로 도 17b에 도시된 바와 같이, 인터레이스 스캔(interlaced scan)이 사용될 수 있다. 위에서 설명한 바와 같이, 크로마 서브샘플링 포맷은 4:2:0일 수 있다(예컨대, chroma\_format\_idc는 1과 같다). 일례로, 가변 크로마 위치 타입(예컨대, ChromaLocType)은 현재 행(예컨대, ChromaLocType이 chroma\_sample\_loc\_type\_top\_field임) 또는 다음 행(예컨대, ChromaLocType이 chroma\_sample\_loc\_type\_bottom\_field임)을 지시할 수 있다. 현재 행(1711, 1713, 1715, 1717)과 다음 행(1712, 1714, 1716, 1718)이 개별적으로 스캔될 수 있다. 예를 들어, 현재 행(1711, 1713, 1715, 1717)이 먼저 스캔되고, 이어서 다음 행(1712, 1714, 1716, 1718)이 스캔될 수 있다. 현재 행은 루마 샘플(1701)을 포함할 수 있는 한편, 다음 행은 루마 샘플(1702)을 포함할 수 있다.

[0187] 유사하게, 대응하는 크로마 블록은 인터레이스 방식으로 스캔될 수 있다. 채움이 없는 크로마 샘플(1703, 1704, 1705, 1706, 1707 또는 1708)을 포함하는 행(1751, 1753)은 현재 행(또는 현재 크로마 행)으로 지칭될 수 있고, 회색 채움을 갖는 크로마 샘플(1703, 1704, 1705, 1706, 1707 또는 1708)을 포함하는 행(1752, 1754)은 다음 행(또는 다음 크로마 행)으로 지칭될 수 있다. 일례로, 인터레이스 스캔 동안, 행(1751, 1753)이 먼저 스캔된 후, 행(1752, 1754)의 스캔이 이어질 수 있다.

[0188] 위에서 설명된 ALF 외에, 제한된 방향 향상 필터(CDEF)가 또한 비디오 코딩에서 루프 필터링을 위해 사용될 수 있다. 루프 내 CDEF는 이미지의 세부사항을 유지하면서 양자화 링잉(ringing) 아티팩트와 같은 코딩 아티팩트를 필터링하는 데 사용될 수 있다. 일부 코딩 기술에서 샘플 적응 오프셋(SAO) 알고리즘이 픽셀의 서로 다른 클래스에 대한 신호 오프셋을 정의함으로써 유사한 목표를 달성하는 데 사용될 수 있다. SAO와 달리, CDEF는 비선형 공간 필터이다. 일부 예에서 CDEF 필터의 설계는 쉽게 벡터화 가능(예컨대, 단일 명령, 다중 데이터(SIMD: single instruction, multiple data) 동작으로 구현 가능)한 것으로 제약되는데, 이는 중앙값 필터 및 양방향 필터와 같은 다른 비선형 필터에 대한 경우가 아니었다.

[0189] CDEF 설계는 다음 관찰에서 비롯된다. 일부 상황에서, 코딩된 이미지의 링잉 아티팩트의 양은 양자화 단계 크기에 대략 비례할 수 있다. 양자화된 이미지에 유지되는 가장 작은 디테일도 또한 양자화 단계 크기에 비례한다. 이에 따라, 이미지 세부사항을 유지하는 것은 더 작은 양자화 단계 크기를 요구할 것이며, 더 높은 바람직하지 않은 양자화 링잉 아티팩트를 산출할 것이다. 다행히도, 주어진 양자화 단계 크기에 대해, 링잉 아티팩트의 진폭은 세부사항의 진폭 미만일 수 있으며, 이로써 충분한 세부사항을 유지하면서 링잉 아티팩트를 필터링하기 위해 균형을 유지하도록 CDEF를 설계할 기회를 제공한다.

[0190] CDEF는 먼저 각각의 블록의 방향을 식별할 수 있다. 그 다음, CDEF는 식별된 방향을 따라 적응적으로 필터링하고 식별된 방향에서 45° 회전된 방향을 따라 더 적은 정도로 필터링할 수 있다. 필터 강도는 명시적으로 시그



널링될 수 있으며, 이는 세부사항의 블러링(blurring)에 대한 고도의 제어를 허용한다. 필터 강도에 대해 효율적인 인코더 탐색이 설계될 수 있다. CDEF는 2개의 루프 내 필터를 기반으로 할 수 있으며 조합된 필터는 비디오 코딩에 사용할 수 있다. 일부 예시적인 구현에서, CDEF 필터(들)는 루프 내 필터링을 위해 블록 분리 필터(들)를 따를 수 있다.

[0191] 도 18에 예시된 바와 같이, 예를 들어 블록 분리 필터 이후에 재구성된 픽셀(또는 샘플)에 대해 방향 탐색이 동작할 수 있다. 재구성된 픽셀은 디코더에 이용 가능하기 때문에, 방향은 시그널링을 필요로 하지 않을 수 있다. 방향 탐색은 직선이 아닌 에지를 적절하게 처리하기에 충분히 작고(에지가 필터링 블록에서 충분히 직선으로 나타나도록) 양자화된 이미지에 적용될 때 방향을 안정적으로 추정하기에 충분히 큰 적절한 크기를 갖는 블록(예컨대,  $8 \times 8$  블록)에서 작동할 수 있다.  $8 \times 8$  구역에 걸쳐 일정한 방향을 가지면 필터의 벡터화가 더 쉬워질 수 있다. 각각의 블록에 대해, 양자화된 블록과 완전한 방향성 블록 간의 차이 측정치, 이를테면 제곱 차의 합(SSD: sum of squared differences), RMS 오류 등을 최소화함으로써 블록의 패턴과 가장 잘 일치하는 방향이 결정될 수 있다. 일례로, 완전한 방향성 블록(예컨대, 도 18의 (1820) 중 하나)은 한 방향으로 라인을 따라 모든 픽셀이 동일한 값을 갖는 블록을 의미한다. 도 18은 본 개시내용의 예시적인 실시예에 따른  $8 \times 8$  블록(1810)에 대한 방향 탐색의 일례를 도시한다. 도 18에 도시된 예에서, 45도 방향(1823)이 오차(1840)를 최소화할 수 있기 때문에 한 세트의 방향(1820) 중 45도 방향(1823)이 선택된다. 예를 들어, 45도 방향에 대한 오차는 12이며 행(1840)으로 지시되는 12 내지 87 범위의 오차 중 가장 작다.

[0192] 예시적인 비선형 저역 통과 방향성 필터가 아래에서 보다 상세히 설명된다. 방향을 식별하는 것은 식별된 방향을 따라 필터 탭을 정렬하여 방향성 에지 또는 패턴을 유지하면서 링잉 아티팩트를 줄이는 데 도움이 될 수 있다. 그러나 일부 예에서, 방향성 필터링만으로는 충분히 링잉 아티팩트를 줄일 수 없다. 기본 방향(예컨대, 식별된 방향)을 따라 놓이지 않은 픽셀에 대한 추가 필터 탭을 사용하는 것이 바람직하다. 블러링 위험을 줄이기 위해, 추가 필터 탭은 보다 보수적으로 처리될 수 있다. 이에 따라, CDEF가 기본 탭과 보조 탭을 정의할 수 있다. 일부 예시적인 구현에서, 완전한 2차원(2-D: two-dimensional) CDEF 필터는 다음과 같이 표현될 수 있다:

$$y(i, j) = x(i, j) + \text{round} \left( \sum_{m, n} w_{d, m, n}^{(p)} f(x(m, n) - x(i, j), S^{(p)}, D) + \sum_{m, n} w_{d, m, n}^{(s)} f(x(m, n) - x(i, j), S^{(s)}, D) \right), \quad \text{식 (14)}$$

[0194] 식 (14)에서, D는 감쇠 파라미터를 나타내고,  $S^{(p)}$  및  $S^{(s)}$ 는 각각 1차 탭 및 2차 탭의 강도를 나타내며, 함수  $\text{round}(\cdot)$ 는 동점(ties)을 0에서 멀리 반올림하는 연산을 나타낼 수 있고,  $w_{d, m, n}^{(p)}$  및  $w_{d, m, n}^{(s)}$ 는 필터 가중치를 나타내고,  $f(d, S, D)$ 는 필터링된 픽셀(예컨대,  $x(i, j)$ )과 이웃 픽셀(예컨대,  $x(m, n)$ ) 각각 간의 차이  $d$ (예컨대,  $d = x(m, n) - x(i, j)$ )에 대해 작동하는 제약 함수를 나타낸다. 그 차이가 작을 때,  $f(d, S, D)$ 는 차이  $d$ 와 같을 수 있고(예컨대,  $f(d, S, D) = d$ ), 따라서 필터는 선형 필터처럼 동작할 수 있다. 그 차이가 클 때,  $f(d, S, D)$ 는 0과 같을 수 있으며(예컨대,  $f(d, S, D) = 0$ ), 이는 필터 탭을 효과적으로 무시한다.

[0195] 다른 루프 내 처리 컴포넌트로서, 한 세트의 루프 내 복원 방식이 블록 분리 연산을 넘어, 일반적으로 잡음을 제거하고 에지의 품질을 향상시키기 위해 블록 분리 후 비디오 코딩에서 사용될 수 있다. 한 세트의 루프 내 복원 방식은 적절한 크기의 타일마다 프레임(또는 픽처) 내에서 전환 가능할 수 있다. 루프 내 복원 방식의 일부 예는 분리 가능한 대칭 위너 필터, 및 부분 공간 투영이 있는 이중 자체 가이드 필터를 기반으로 아래에서 설명된다. 콘텐츠 통계는 프레임 내에서 상당히 다를 수 있기 때문에, 필터는 프레임의 상이한 구역에서 상이한 필터가 트리거될 수 있는 전환 가능한 프레임워크 내에 통합될 수 있다.

[0196] 예시적인 분리 가능한 대칭 위너 필터가 아래에서 설명된다. 위너 필터가 전환 가능한 필터 중 하나로서 사용될 수 있다. 저하된 프레임의 모든 각각의 픽셀(또는 샘플)은 픽셀 주변의  $w \times w$  윈도우 내의 픽셀의 비-인과(non-causal) 필터링된 버전으로서 재구성될 수 있으며, 여기서  $w = 2r + 1$ 이며 정수  $r$ 에 대해 홀수이다. 2-D 필터 탭은  $w^2 \times 1$ 개의 엘리먼트를 갖는 열 벡터화 형태의 벡터  $F$ 로 표기될 수 있으며, 간단한 선형 최소 평균 제곱 오차(LMMSE: linear minimum mean square error) 최적화는  $F = H^{-1} M$ 으로 주어진 필터 파라미터로 이어질 수

있으며, 여기서  $H$ 는  $E[XX^T]$ 과 같고 픽셀 주변의  $w \times w$  윈도우에서  $w^2$ 개의 샘플의 열 벡터화된 버전인  $x$ 의 자기공분산(auto-covariance)이며, 여기서  $M$ 은  $E[YY^T]$ 은 추정될 스칼라 소스 샘플  $y$ 와  $x$ 의 상호 상관을 나타낸다. 인코더는 블록 분리된 프레임 및 소스에서의 실현으로부터  $H$  및  $M$ 을 추정하고 결과적인 필터  $F$ 를 디코더에 송신하도록 구성될 수 있다. 그러나 일부 예시적인 구현에서,  $w^2$ 개의 탭을 전송하는데 상당한 비트레이트 비용이 발생할 수 있다. 또한, 분리 불가능한 필터링은 디코딩을 엄청나게 복잡하게 만들 수 있다. 따라서  $F$ 의 성질에 대해 복수의 추가 제약이 부가될 수 있다. 예를 들어, 필터링이 분리 가능한 수평 및 수직  $w$ -탭 컨볼루션으로서 구현될 수 있게  $F$ 가 분리 가능하도록 제한될 수 있다. 일례로, 수평 및 수직 필터 각각이 대칭으로 제약된다. 또한, 일부 예시적인 구현에서, 수평 및 수직 필터 계수의 합은 1로 합산되는 것으로 가정될 수 있다.

[0197] 부분 공간 투영을 이용한 이중 자체 가이드 필터링은 또한 루프 내 복원을 위한 전환 가능한 필터 중 하나로서 사용될 수 있으며 아래에서 설명된다. 일부 예시적인 구현에서, 필터링되지 않은 샘플  $x$ 로부터 필터링된 출력  $y$ 를 계산하기 위해 로컬 선형 모델이 사용되는 이미지 필터링에서 가이드 필터링이 사용될 수 있다. 로컬 선형 모델은 다음과 같이 작성될 수 있으며:

[0198]  $y = Fx + G$  식(15)

[0199] 여기서  $F$  및  $G$ 는 필터링된 픽셀의 이웃에 있는 (가이드 이미지도도 또한 지칭되는) 열화 이미지 및 가이드 이미지의 통계를 기반으로 결정될 수 있다. 가이드 이미지가 열화된 이미지와 동일하다면, 결과적인 자체 가이드 필터링은 에지 보존 평활화 효과를 가질 수 있다. 본 개시내용의 일부 양상에 따르면, 특정 형태의 자체 안내 필터링은 2개의 파라미터: 반경  $r$  및 잡음 파라미터  $e$ 에 좌우될 수 있으며, 다음과 같이 열거된다:

[0200] 1. 모든 각각의 픽셀 주위의  $(2r + 1) \times (2r + 1)$  윈도우에서 픽셀의 평균( $\mu$ ) 및 분산( $\sigma^2$ )을 구한다. 예를 들어, 픽셀의 평균( $\mu$ ) 및 분산( $\sigma^2$ )을 구하는 것은 통합 이미지를 기반으로 하는 박스 필터링으로 효율적으로 구현될 수 있다.

[0201] 2. 식(16)에 기초하여 모든 각각의 픽셀에 대한 파라미터  $f$  및  $g$ 를 계산한다:

[0202]  $f = \sigma^2 / (\sigma^2 + e); g = (1 - f)\mu$  식(16)

[0203] 3. 사용할 픽셀 주변의  $3 \times 3$  윈도우에서 파라미터  $f$  및  $g$ 의 값의 평균으로서 모든 각각의 픽셀에 대한  $F$  및  $G$ 를 계산한다.

[0204] 이중 자체 가이드 필터링은 반경  $r$  및 잡음 파라미터  $e$ 에 의해 제어될 수 있으며, 여기서 더 큰 반경  $r$ 은 더 높은 공간 분산을 의미할 수 있고 더 높은 잡음 파라미터  $e$ 는 더 높은 범위 분산을 의미할 수 있다.

[0205] 도 19는 본 개시내용의 예시적인 실시예에 따른 부분 공간 투영의 일례를 도시한다. 도 19에 도시된 예에서, 부분 공간 투영은 비용이 적게 드는 복원  $X_1$  및  $X_2$ 를 사용하여 소스  $Y$ 에 더 가까운 최종 복원  $X_t$ 를 생성할 수 있다. 비용이 적게 드는 복원  $X_1$  및  $X_2$ 가 소스  $Y$ 에 가깝지 않더라도, 적절한 승수  $\{\alpha, \beta\}$ 는 비용이 적게 드는 복원  $X_1$  및  $X_2$ 가 올바른 방향으로 움직인다면 비용이 적게 드는 복원  $X_1$  및  $X_2$ 를 소스  $Y$ 에 훨씬 더 가깝게 가져올 수 있다. 예를 들어, 최종 복원  $X_t$ 는 아래 식(17)에 기초하여 획득될 수 있다.

[0206]  $X_t = Y + \alpha(X_1 - Y) + \beta(X_2 - Y)$  식(17)

[0207] 위에서 설명된 블록 분리 필터, ALF, CDEF 및 루프 복원 외에도, 성분 간 샘플 오프셋(CCSO) 필터 또는 CCSO로 지칭되는 루프 필터링 방법이 또한 (재구성 샘플로도 또한 지칭되는) 재구성된 샘플의 왜곡을 줄이도록 루프 필터링 프로세스에서 구현될 수 있다. CCSO 필터는 루프 필터링 스테이지 내 어느 곳에나 배치될 수 있다. CCSO 필터링 프로세스에서, 제1 색 성분의 처리된 입력 재구성 샘플에 기초하여 출력 오프셋을 결정하기 위해 비선형 매핑이 사용될 수 있다. 출력 오프셋은 CCSO의 필터링 프로세스에서 제2 색 성분의 재구성된 샘플에 더해질 수 있다.

[0208] 입력 재구성 샘플은 도 20에 도시된 바와 같이, 필터 지원 영역에 위치한 제1 색 성분에서 나올 수 있다. 구체적으로, 도 20은 본 개시내용의 일 실시예에 따른 CCSO 필터의 필터 지원 영역의 일례를 도시한다. 필터 지원 영역은 4개의 재구성된 샘플:  $p_0, p_1, p_2$  및  $p_3$ 를 포함할 수 있다. 도 20의 예에서 4개의 입력 재구성 샘플은 수직 방향 및 수평 방향으로 십자형을 따른다. 일례로, 제1 색 성분의 ( $c$ 로 표기된) 중앙 샘플 및 제2 색 성분

에서 필터링될 (f로 표기된) 샘플이 함께 위치된다. 입력 재구성 샘플을 처리할 때, 다음 단계가 적용될 수 있다:

[0209] 단계 1: 4개의 재구성된 샘플:  $p_0$ ,  $p_1$ ,  $p_2$ ,  $p_3$ 과 중앙 샘플  $c$  사이의 델타 값(예컨대, 차이)이 계산되고, 각각  $m_0$ ,  $m_1$ ,  $m_2$  및  $m_3$ 으로 표기된다. 예를 들어,  $p_0$ 와  $c$  사이의 델타 값은  $m_0$ 이다.

[0210] 단계 2:  $m_0$  내지  $m_3$ 의 델타 값은 다수(예컨대, 4개)의 이산 값으로 더 양자화될 수 있다. 양자화된 값은 예를 들어  $m_0$ ,  $m_1$ ,  $m_2$  및  $m_3$ 에 대해 각각  $d_0$ ,  $d_1$ ,  $d_2$  및  $d_3$ 으로 표기될 수 있다. 일례로,  $d_0$ ,  $d_1$ ,  $d_2$  및  $d_3$  각각에 대한 양자화된 값은 다음 양자화 프로세스에 기초하여 -1, 0 또는 1일 수 있으며:

[0211]  $m_i < -N$ 이라면,  $d_i = -1$ ; 식(18)

[0212]  $-N \leq m_i \leq N$ 이라면,  $d_i = 0$ ; 식(19)

[0213]  $m_i > N$ 이라면,  $d_i = 1$ . 식(20)

[0214] 여기서  $N$ 은 양자화 단계 크기이고,  $N$ 의 예시적인 값은 4, 8, 12, 16 등이며,  $d_i$  및  $m_i$ 는 각각의 양자화된 값 및 델타 값을 의미하고, 여기서  $i$ 는 0, 1, 2 또는 3이다.

[0215] 양자화된 값  $d_0$  내지  $d_3$ 은 비선형 매핑의 조합을 식별하는 데 사용될 수 있다. 도 20에 도시된 예에서, CCSO 필터는 4개의 필터 입력  $d_0$  내지  $d_3$ 을 가지며, 각각의 필터 입력은 3개의 양자화된 값(예컨대, -1, 0, 1) 중 하나를 가질 수 있고, 따라서 총 조합 수는  $81$ (예컨대,  $3^4$ )이다. 도 27a - 도 27c는 본 개시내용의 일 실시예에 따른 81개의 조합의 일례를 도시한다. 마지막 열은 각각의 조합에 대한 출력 오프셋 값을 나타낼 수 있다. 출력 오프셋 값은 0, 1, -1, 3, -3, 5, -5, -7 등과 같은 정수일 수 있다. 첫 번째 열은 양자화된  $d_0$ ,  $d_1$ ,  $d_2$  및  $d_3$ 의 이러한 조합에 할당된 인덱스를 나타낸다. 중간 열은 양자화된  $d_0$ ,  $d_1$ ,  $d_2$  및  $d_3$ 의 가능한 모든 조합을 나타낸다.

[0216] CCSO 필터의 최종 필터링 프로세스는 다음과 같이 적용될 수 있으며:

[0217]  $f' = clip(f + s)$ , 식(21)

[0218] 여기서  $f$ 는 필터링될 재구성된 샘플이고,  $s$ 는 예를 들어, 도 21a - 도 21c로부터 검색된 출력 오프셋 값이다. 식(21)에 도시된 예에서, 필터링될 재구성된 샘플  $f$ 의 필터링된 샘플 값  $f'$ 는 비트 심도와 연관된 범위로 더 클리핑될 수 있다.

[0219] 로컬 샘플 오프셋(LSO) 방법 또는 LSO 필터링 프로세스가 비디오 코딩에 사용될 수 있다. LSO에서는, CCSO에서 사용되는 것과 유사한 필터링 접근 방식이 적용될 수 있다. 그러나 필터링 프로세스에서 사용되는 입력 재구성 샘플이 속하는 색 성분과 동일한 색 성분인 색 성분에 대해 출력 오프셋 값이 적용될 수 있다. 이에 따라, LSO에서는, 필터링 프로세스에 사용된 입력 재구성 샘플(예컨대,  $p_0$ - $p_3$ ,  $c$ ) 및 필터링될 재구성된 샘플(예컨대,  $f$ )은 루마 성분, 크로마 성분, 또는 임의의 적합한 성분과 같은 동일한 성분에 있다. LSO는 CCSO와 유사하거나 동일한 (도 20에 도시된 것과 같은) 필터 형상일 수 있다.

[0220] 도 20에 도시된 바와 같이, 제1 색상의  $p_0$ ,  $p_1$ ,  $p_2$  및  $p_3$ 을 갖는 제1 색상의 샘플  $c$ 에 대응하는 필터링될 제2 색상의 재구성된 샘플  $f$ 의 예시적인 CCSO 필터링은 5-탭 CCSO 필터 설계로 지칭될 수 있다. 대안으로, 상이한 수의 필터 탭을 갖는 다른 CCSO 설계가 사용될 수 있다. 예를 들어, 복잡성이 더 낮은 3탭 CCSO 설계가 비디오 코딩에 사용될 수 있다. 도 22는 본 개시내용의 실시예에 따른 CCSO의 예시적인 구현을 도시한다. 3-탭 CCSO 구현을 위해 8개의 서로 다른 예시적인 필터 형상 중 임의의 형상이 정의될 수 있다. 필터 형상 각각은 (제1 색 성분으로도 또한 지칭되는) 제1 성분에서 (3개의 탭으로도 또한 지칭되는) 3개의 재구성된 샘플의 포지션을 정의할 수 있다. 3개의 재구성된 샘플은 도 22에서 동일한 번호(1-8 중 하나)로 표기된 바와 같이, ( $c$ 로 표기된) 중앙 샘플 및 2개의 대칭적으로 위치한 샘플을 포함할 수 있다. 일례로, 필터링될 제2 색 성분의 재구성된 샘플은 중앙 샘플  $c$ 와 같이 위치된다. 명확하게 하기 위해, 필터링될 제2 색 성분의 재구성된 샘플은 도 22에 도시되지 않는다.

[0221] 샘플 적응 오프셋(SAO) 필터는 비디오 코딩에서 사용될 수 있다. 일부 예시적인 구현에서, SAO 필터 또는 SAO 필터링 프로세스는 예를 들어, 슬라이스 헤더에서 오프셋 값을 사용함으로써 블록 분리 필터 이후 재구성 신호에 적용될 수 있다. 루마 샘플의 경우, 인코더는 SAO 필터가 현재 슬라이스에 적용되는지 여부를 결정할 수 있다. SAO 필터가 인에이블된다면, 표 4에 도시된 바와 같이, 현재 픽처가 4개의 하위 구역으로 재귀적으로 분할될 수 있으며, 각각의 하위 구역에 대해 6개의 SAO 타입(예컨대, SAO 타입 1-6) 중 하나가 선택될 수 있다.

SAO 필터는 재구성된 픽셀을 복수의 카테고리로 분류할 수 있으며, 현재 하위 구역 내 각각의 카테고리의 픽셀에 오프셋을 추가함으로써 왜곡을 줄일 수 있다. 예지 속성은 SAO 타입 1-4의 픽셀 분류에 사용될 수 있고, 픽셀 강도는 SAO 타입 5-6의 픽셀 분류에 사용될 수 있다.

[0222] 표 4 SAO 타입의 규칙

표 4

SAO 타입	사용될 SAO 타입	카테고리 수
0	없음	0
1	1-D 0도 패턴 에지 오프셋	4
2	1-D 90도 패턴 에지 오프셋	4
3	1-D 135도 패턴 에지 오프셋	4
4	1-D 45도 패턴 에지 오프셋	4
5	중앙 대역의 대역 오프셋	16
6	중앙 대역의 대역 오프셋	16

[0224] 대역 오프셋(BO: band offset)은 하위 구역의 픽셀(예컨대, 모든 픽셀)을 다수의 대역으로 분류하는데 사용될 수 있으며, 여기서 각각의 대역은 동일한 강도 간격의 픽셀을 포함할 수 있다. 강도 범위는 최소 강도 값(예컨대, 0)에서 최대 강도 값(예컨대, 8비트 픽셀의 경우 255)까지 복수의 간격(예컨대, 32개의 간격)으로 균등하게 분할될 수 있으며, 각각의 간격은 오프셋을 가질 수 있다. 후속하여, 복수의 구간 또는 대역(예컨대, 32개의 대역)이 2개의 그룹으로 나뉠 수 있다. 한 그룹은 16개의 중앙 대역을 포함할 수 있고, 다른 그룹은 16개의 나머지 대역을 포함할 수 있다. 일례로, 하나의 그룹에서의 오프셋만이 전송된다. BO의 픽셀 분류 연산과 관련하여, 각각의 픽셀의 5개의 최상위 비트가 대역 인덱스로서 직접 사용될 수 있다.

[0225] 에지 오프셋(EO: edge offset)은 도 23에 도시된 바와 같이, 에지 방향 정보의 에지 방향 정보를 고려하여 픽셀 분류를 위해 4개의 1-D 3-픽셀 패턴을 사용할 수 있다. 도 23은 EO에서의 픽셀 분류를 위한 4개의 1-D 3-픽셀 패턴의 예를 도시한다. 왼쪽에서 오른쪽으로, 4개의 1-D 3-픽셀 패턴은 1D 0도 패턴(2310), 1D 90도 패턴(2320), 1D 135도 패턴(2330) 및 1D 45도 패턴(2340)에 각각 대응한다. 픽처(예컨대, 현재 픽처)의 각각의 하위 구역에 대해, 각각의 픽셀을 픽셀의 2개의 이웃 픽셀과 비교함으로써 픽셀을 다수의 카테고리로 분류하기 위해 4개의 패턴 중 하나가 선택될 수 있다. 선택은 비트스트림에서 부가 정보로서 송신될 수 있다. 표 5는 EO에 대한 픽셀 분류 규칙을 보여준다.

[0226] 표 5 EO에 대한 픽셀 분류 규칙

표 5

카테고리	조건
1	$c < 2 \text{ neighbors}$
2	$c < 1 \text{ neighbor} \ \&\& \ c == 1 \text{ neighbor}$
3	$c > 1 \text{ neighbor} \ \&\& \ c == 1 \text{ neighbor}$
4	$c > 2 \text{ neighbors}$
0	위의 항목 없음

[0228] 일례로, 디코더 측의 SAO는 라인 버퍼가 저장될 수 있도록 LCU 독립적으로 동작되는 것이 바람직하다. SAO를 LCU 독립적으로 동작시키기 위해, 일례로, 각각의 LCU의 최상부 및 최하부 행의 픽셀은 90도, 135도 및 45도 분류 패턴이 선택될 때 SAO 처리되지 않으며, 0도, 135도 및 45도 패턴이 선택되면 각각의 LCU의 최외측 및 최우측 열의 픽셀은 SAO 처리되지 않는다.

[0229] 아래 표 6은, 파라미터가 이웃 CTU로부터 병합되지 않는다면 CTU에 대해 시그널링될 수 있는 신택스를 설명한다.

[0230] 표 6 - SAO 신택스의 일례

표 6

sao_offset_vlc( rx, ry, cIdx ) {	<b>Descriptor</b>
sao_type_idx[ cIdx ][ rx ][ ry ]	ue(v)
if( sao_type_idx[ cIdx ][ rx ][ ry ] == 5 ) {	
sao_band_position[ cIdx ][ rx ][ ry ]	u(5)
for( i = 0; i < 4; i++ )	
sao_offset[ cIdx ][ rx ][ ry ][ i ]	se(v)
} else if( sao_type_idx[ cIdx ][ rx ][ ry ] != 0 )	
for( i = 0; i < 4; i++ )	
sao_offset[ cIdx ][ rx ][ ry ][ i ]	ue(v)
}	

[0231]

[0232]

위에서 설명된 CCSO 루프 필터링 프로세스에 대한 일부 구현에서, 도 21a - 도 21c의 5-탭 CCSO 필터링에 대해 예시적으로 설명되는 바와 같이, 성분 간 샘플 오프셋 값과 양자화된 델타 값(또는 이들의 인덱스)의 조합 간의 대응은 CCSO 룩업 테이블(LUT: lookup table)로 지칭될 수 있다. LUT 중 하나 이상은 비디오 인코딩 또는 디코딩 동안 CCSO 필터링 프로세스에서 잠재적으로 사용될 수 있다. CCSO 필터링을 위해 여러 LUT가 제공되는 경우, 다양한 레벨(예컨대, 픽처 레벨, 슬라이스 레벨, CTB 레벨, CB 레벨, FU 레벨 등)에서 루프 필터링 프로세스 동안 인코더 또는 디코더에 의해 다수의 LUT로부터의 선택이 동적으로 그리고 적응적으로 이루어질 수 있다.

[0233]

이러한 LUT 각각은 임의의 적절한 수의 탭(예컨대, 5개의 탭 또는 3개의 탭, 또는 임의의 다른 수의 탭) 및 델타 양자화 레벨(예컨대, 위에서 설명된 3-레벨 델타 양자화, 또는 임의의 다른 수의 델타 양자화 레벨)에 기초할 수 있다. 이에 따라, LUT는 상이한 크기일 수 있다(예컨대, 위에서 설명된 바와 같이, 5-탭 및 3 양자화 레벨 CCSO에 대한  $(5-1)^3=81$ 개의 양자화된 델타 조합 및 오프셋 대응; 또는 3-탭 및 3 양자화 레벨 CCSO에 대한  $(3-1)^3=8$ 개의 양자화된 델타 조합 및 오프셋 대응).

[0234]

위에서 설명된 바와 같이, 이러한 LUT 중 일부는 미리 정의될 수 있다. 예를 들어, 이러한 미리 정의된 CCSO LUT는 CCSO 필터링 프로세스에 의한 일반적인 사용을 위해 트레이닝(training) 이미지 데이터를 사용하여 오프라인으로 사전 트레이닝될 수 있다. 이러한 미리 정의된 LUT는 고정 상수(다양한 미리 정의된 양자화된 델타 값 조합에 대한 고정 상수 오프셋)일 수 있으며, 따라서 이러한 미리 정의된 LUT의 내용은 비디오 비트스트림에서 인코더에서 디코더로 시그널링될 필요가 없을 수 있다. 대신에, 이러한 LUT는 사전 저장될 수 있거나, 비디오 인코더 또는 비디오 디코더에서 CCSO 필터링 프로세스에 의한 사용을 위해 하드웨어링(hardware)되거나 하드 코딩될 수 있다.

[0235]

위에서 추가로 설명된 바와 같이, CCSO 필터링 프로세스 동안 사용되는 미리 정의된/사전 트레이닝된 LUT 외에 CCSO LUT 중 일부는 오프라인 트레이닝되는 것이 아니라 인코딩 프로세스 동안 인코더에 의해 도출될 수 있다. 이러한 CCSO LUT는 미리 정의되지 않으며, 따라서 이들의 내용은 비트스트림에서 명시적으로 시그널링될 필요는 없을 것이다. 이러한 인코더 도출 LUT의 시그널링은 큰 LUT에 대해 특히, 프레임마다 상당한 오버헤드를 수반하기 때문에 이는 대개 비용이 많이 들며, 이로써 잠재적으로 상당한 그리고 바람직하지 않은 전체 비트레이트 손실을 야기한다. 이에 따라, 비트스트림에서 이러한 LUT를 조직하고, 인코딩하고, 시그널링하기 위한 효율적인 방식을 고안하는 것이 바람직할 수 있다.

[0236]

일부 예시적인 구현에서, 비디오를 인코딩 또는 디코딩할 때 미리 정의된 LUT만이 CCSO 필터링 프로세스에서 사용될 수 있다. 다른 일부 예시적인 구현에서는, 비디오를 인코딩 또는 디코딩할 때 인코더 도출 LUT만이 CCSO 필터링 프로세스에서 사용될 수 있다. 또 다른 일부 예시적인 구현에서, 비디오를 인코딩 또는 디코딩할 때 미리 정의된 LUT와 인코더 도출 LUT 모두 CCSO 필터링 프로세스에서 사용될 수 있고, 특정 FU의 CCSO 필터링은 미리 정의된 그리고 인코더 도출 LUT로부터 선택된 임의의 LUT를 사용할 수 있다.

[0237]

위에서 상세히 설명된 바와 같이, CCSO 프로세스는 입력으로서 제1 색 성분(예컨대, Y 또는 Cb 또는 Cr, 즉 루마 성분을 포함하고 단지 크로마 성분으로 제한되지 않음)의 재구성된 샘플을 사용하는 필터링 프로세스를 의미



하고, 출력은 특정 CCSO LUT에 따른 제1 색 성분의 다른 색 성분인 제2 색 성분에 적용된다. CCSO 필터의 예시적인 5-탭 필터 형상이 도 20에 도시되고 대응하는 예시적인 LUT가 도 21a - 도 21c에 도시된다.

[0238] 제1 색 성분과 제2 색 성분이 서로 다른 필터링 프로세스는 CCSO로 지칭되지만, 위에서 설명된 바와 같이 로컬 샘플 오프셋(LSO)으로 지칭되는 인트라 색 오프셋 프로세스도 또한 구현될 수 있다. LSO 필터 프로세스는 입력으로서 제1 색 성분의 재구성된 샘플(예컨대, Y 또는 Cb 또는 Cr)을 사용할 수 있고, 출력은 특정 LSO LUT에 따라 동일한 제1 색 성분에 적용된다. 특정 LSO LUT는 LSO에 대한 하나 이상의 LUT로부터 선택될 수 있고, CCSO 프로세스에서의 성분 간 샘플 오프셋의 결정과 유사하게 로컬 샘플 오프셋을 결정하는 데 사용될 수 있다. CCSO LUT와 마찬가지로, 이러한 LSO LUT는 예를 들어, 고정 상수 LUT로서 미리 정의(오프라인 트레이닝)될 수 있거나, 인코딩 프로세스 동안 인코더에 의해 도출될 수 있다. 인코더 도출된 LSO LUT는 비트스트림에서 시그널링될 필요가 있을 것이며, 반면에 미리 정의된/고정된/일정한/오프라인 트레이닝된 LSO LUT는 위에서 설명된 미리 정의된 CCSO LUT와 유사하게, 인코더 또는 디코더에서 미리 저장되거나 하드와이어링 또는 하드 코딩될 수 있으며, 시그널링될 필요가 없을 수 있다.

[0239] 다음 개시내용은 디코더 및 인코더의 코딩 블록에서 재구성된 샘플의 루프 내 필터링 동안 샘플 오프셋 LUT의 동적 및 적응적 선택의 다양한 예시적인 구현을 추가로 설명한다. 이러한 구현 및 이들의 기본 원칙은 CCSO 필터링에 적용 가능할 뿐만 아니라 LSO 필터링에도 적용된다. 설명의 편의상, 일부 상황에서는 CCSO가 구체적으로 언급될 수 있다. 그러나 아래의 개시내용은 CCSO에 국한되지 않고 LSO에도 적용 가능하다.

[0240] 일부 예시적인 구현에서, CCSO 및/또는 LSO에 대해 하나 이상의 미리 정의된 룩업 테이블(LUT)이 정의될 수 있고, 이러한 룩업 테이블은 예를 들어, 식(21)에 따라 CCSO 또는 LSO 필터링된 샘플 값을 계산하기 위해 특정 색 성분의 재구성된 샘플 값에 적용되는 오프셋 값을 도출하는 데 사용될 수 있다. 이러한 미리 정의된 CCSO 및/또는 LSO LUT는 사전에 인코더와 디코더 간에 공유되고 알려지게 된다. 따라서 이러한 LUT는 인코더 또는 디코더 디바이스 중 임의의 디바이스에 저장되거나, 하드와이어링되거나, 하드 코딩될 수 있다.

[0241] 각각의 LUT는 특정 델타 값과 샘플 오프셋 값 사이의 특정 비선형 매핑을 나타낸다. 다수의 LUT가 사용에 이용 가능한 경우, 각각의 색 성분에 대한 다양한 필터 레벨(FU 또는 임의의 다른 레벨)에서의 LUT의 선택이 인코더에서 이루어져 디코더에 시그널링될 수 있다. 아래에서 설명되는 일부 구현에서, LUT의 선택(또는 다양한 델타 값과 샘플 오프셋 값 사이의 다양한 비선형 매핑)은 다양한 로컬 적응성에 따라 구현될 수 있다.

[0242] 일부 구현에서, CCSO 또는 LSO 필터링이 적용되는 블록 영역의 코딩된 정보에 기초하여 도출된 특정 통계에 기초할 수 있다. 필터링 블록 영역의 이러한 통계 특징은 샘플 오프셋 필터 선택(예컨대, CCSO 또는 LSO 룩업 테이블의 선택)에 의해 적응적으로 처리될 수 있는 잠재적 왜곡과 상관되는 재구성된 샘플의 이미지 특징을 나타낼 수 있다. 이러한 통계는 다음 중 하나 이상을 포함할 수 있지만 이에 제한되는 것은 아니다.

[0243] I. 위에서 설명된 CDEF에서 도출된 에지 방향 또는 캐니 에지(Canny Edge) 검출기 또는 쇼벨 에지(Sobel Edge) 검출기와 같은 다른 에지 검출 방법을 사용하여 도출된 에지 방향.

[0244] II. CDEF가 적용되는 다양한 필터링 레벨에서 샘플 영역 내에서의 재구성된 샘플의 평활도 측정.

[0245] --일 실시예에서, 평활도는 샘플 영역 내의 샘플 값의 범위에 의해 계산될 수 있다. 샘플 값의 범위는 최대 샘플 값과 최소 샘플 값의 절대 차로서 정의될 수 있다.

[0246] --다른 실시예에서, 평활도는 샘플 영역 내에 위치한 픽셀의 샘플 값의 사분위수 범위에 의해 계산될 수 있다. 샘플 값의 사분위수 범위는 A번째 백분위수에서 샘플 값의 B번째 백분위수를 뺀 값으로서 정의된다. A의 예시적인 값은 75%일 수 있고, B의 예시적인 값은 25%일 수 있다.

[0247] --또 다른 실시예에서, 평활도는 샘플 영역 내의 샘플 값의 분산에 의해 계산될 수 있다. 분산은 다음과 같이 계산될 수 있으며:

$$S^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}, \quad (21)$$

[0249] 여기서  $S^2$ 는 분산이고,  $x_i$ 는 샘플 1 내지 샘플 n을 커버하는 현재 샘플 영역의 i번째 샘플을 의미하며,  $\bar{x}$ 는 샘플 평균값을 나타내고, n은 CCSO 및/또는 LSO가 적용되는 샘플 영역 내의 총 샘플 수를 나타낸다.

[0250] --또 다른 실시예에서, 평활도는 샘플 영역 내의 샘플 값의 표준 편차에 의해 계산될 수 있다. 표준 편차는 다

음과 같이 계산될 수 있다:

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}, \quad (22)$$

[0251]

[0252]

--또 다른 실시예에서, 평활도는 샘플 영역 내의 변화도 값의 범위에 의해 계산될 수 있다. 변화도 값의 범위는 최대 변화도 값과 최소 변화도 값의 절대 차로써 정의될 수 있다. 각각의 샘플 위치에 대한 변화도 값은 현재 포지션에 위치한 샘플 값과 그 주변 포지션에 위치한 샘플 값 간의 차이로써 계산될 수 있다.

[0253]

III. CCSO 또는 LSO가 적용되는 블록에 대해 시그널링되는 예측 모드(예컨대, 인트라 DC 모드가 적용되는지, 인트라 평면 모드가 적용되는지, 인트라 PAETH 모드가 적용되는지, 인트라 SMOOTH 모드가 적용되는지, 인트라 채귀 필터링 모드가 적용되는지, 인트라 SKIP 모드가 적용되는지)를 포함하지만 이에 제한되지 않는 코딩된/코딩 정보, (코딩된 블록 플래그와 같은) 계수 코딩과 관련된 정보, 블록 크기, 양자화 파라미터, 및 현재 블록과 연관된 움직임 벡터(크기).

[0254]

IV. CCSO 또는 LSO를 적용할 때 인코더와 디코더 모두에서 이용 가능한 정보, 이를테면 (ALF와 같은) 루프 필터링 프로세스에서 도출된 샘플 분류 정보, 블록 분리 프로세스에서 도출된 필터링 강도, CDEF 및 루프 복원.

[0255]

블록에 대해 CCSO 또는 LSO 필터링(또는 CCSO 또는 LSO LUT)에서 사용되는 비선형 매핑은 블록의 통계에 따라 선택될 수 있다. 통계는 위에서 설명된 I, II, III 및 IV를 포함할 수 있지만 이에 제한되는 것은 아니다.

[0256]

일 실시예에서, CCSO의 입력인 색 성분은 통계를 도출하는 데 사용될 수 있다.

[0257]

다른 실시예에서, CCSO의 출력에 의해 필터링되는 색 성분은 통계를 도출하는 데 사용될 수 있다.

[0258]

다른 실시예에서, 상기 I에서 정의된 통계(예지 방향 통계)가 사용된다면, CCSO에서 현재 블록에 사용되는 LUT는 현재 블록의 예지 정보에 기초하여 선택될 수 있다.

[0259]

CDEF에서 도출된 예지 방향이 통계로서 사용되는 일례에서, CCSO 및/또는 LSO에 대해 통틀어 미리 정의된 수(예컨대, 8개)의 LUT가 있을 수 있다. LUT 각각은 예시적인 CDEF 예지 도출 프로세스의 출력인, 예컨대 8개의 예지 방향 중 하나에 대응할 수 있다. LUT 중 하나는 예지 방향에 기초한 현재 블록에 대한 이러한 대응에 따라 선택될 수 있다. 예시적인 8개의 LUT는 (미리 정의되지 않는다면) 다양한 레벨, 예컨대 HLS(APS, 슬라이스 헤더, 프레임 헤더, PPS)의 프레임 레벨에서 시그널링될 수 있다.

[0260]

CDEF에서 도출된 예지 방향이 통계로서 사용되는 다른 보다 일반적인 예에서, 총 N개의 LUT가 존재할 수 있는데, LUT 각각은 예시적인 CDEF 예지 도출 프로세스의 출력인 8개의 예지 방향 중 하나보다 많은 예지 방향에 대응한다. LUT 중 하나는 예지 방향에 기초한 현재 블록에 대한 이러한 대응에 따라 선택될 수 있다. N의 예시적인 값은 정수이고,  $1 < N < 8$ 이다. N개의 LUT는 (미리 정의되지 않는다면) HLS(APS, 슬라이스 헤더, 프레임 헤더, PPS)의 프레임 레벨에서 시그널링될 수 있다.

[0261]

일부 예시적인 구현에서, 위의 II에 정의된 통계(평활도 통계)가 사용된다면, CCSO 및/또는 LSO의 현재 블록에 대한 LUT의 선택은 측정 평활도 값이 하나 이상의 주어진 임계치보다 더 작은지(또는 더 큰지) 여부에 기초할 수 있다.

[0262]

일례로, 현재 블록의 표준 편차(S)는 LUT 선택을 위한 미리 정의된 수의 임계값과 비교될 수 있다. 예를 들어, 선택할 다수의 LUT, 예컨대 선택할 4개의 LUT가 존재할 수 있다. 대응하게, 3개의 임계값(S1, S2, S3)이 있을 수 있다. 이에 따라, 4개의 LUT로부터의 LUT 선택은 다음 기준에 기초할 수 있다:  $S < S1$ 이라면 LUT1이 선택될 수 있고,  $S1 \leq S < S2$ 라면 LUT2가 선택될 수 있고,  $S2 \leq S < S3$ 이라면 LUT3이 선택될 수 있고,  $S \geq S4$ 라면 LUT4가 선택될 수 있다.

[0263]

일 실시예에서, 위의 III에서 정의된 통계(코딩/코딩된 정보 통계)가 사용된다면, CCSO 및/또는 LSO에서 현재 블록에 사용되는 LUT의 선택은 코딩/코딩된 정보에 기초할 수 있다.

[0264]

일례로, 현재 블록에 대한 LUT는 현재 블록의 예측 모드에 기초하여 선택될 수 있다. 다양한 예측 모드는 다양한 LUT에 대응할 수 있다. 현재 블록에서 사용되는 예측 모드의 경우, 대응하는 LUT가 선택될 수 있다. 예시적인 예측 모드는 인트라 DC 모드, 인트라 평면 모드, 인트라 PAETH 모드, 인트라 SMOOTH 모드, 인트라 채귀 필터링 모드, 인트라 SKIP 모드 등을 포함하지만 이에 제한되는 것은 아니다.

[0265]

일부 예시적인 구현에서, 필터 계수, 필터 탭 포지션 및 필터 탭의 수를 포함하는, 그러나 이에 제한되는 것은

아닌 위의 다양한 통계 정보가 CCSO에서 사용되는 필터를 결정하는 데 사용될 수 있다. 이러한 결정은 이러한 필터링 파라미터가 상이한 특업 테이블로 구현될 때 위에서 설명된 바와 같은 LUT의 선택과 동일할 수 있다. 대안으로, 이러한 복합 LUT가 특정 복합 LUT에서 이러한 파라미터의 변형을 포함하도록 구성된다면, 선택은 LUT의 선택보다는 LUT에서 엔트리를 선택하는 것을 수반할 수 있다.

[0266] 위의 구현 중 임의의 구현에서, LUT를 선택하기 위한 통계는 임의의 색 성분의 통계에 기초할 수 있다. 이는 단일 색 성분 또는 여러 색 성분의 조합을 기반으로 할 수 있다. 이는 필터링될 색 성분과 동일한 색 성분을 기반으로 하거나 필터링될 색 성분과 상이한 다른 색 성분을 기반으로 할 수 있다. 통계가 수행되는 색 성분(들)은 "적어도 제1 색 성분"으로 지칭될 수 있다. 선택된 샘플 오프셋 필터에 입력되고 필터링될 색 성분은 "제2 색 성분"으로 지칭될 수 있다. 선택된 샘플 오프셋 필터에서 샘플 오프셋 값을 찾는 데 사용되는 델타 값은 색 성분 중 "제3 색 성분"으로 지칭되는 임의의 색 성분에서 필터링되는 샘플과 같이 위치되는 샘플로부터 취해질 수 있다. 제1, 제2 및 제3 색 성분은 동일하거나 상이할 수 있다. 예를 들어, 제2 및 제3 색 성분은 상이하거나(따라서 "성분 간 샘플 오프셋" 필터링) 동일할 수 있다(따라서 "로컬 샘플 오프셋" 필터링).

[0267] 일부 예시적인 구현에서, CCSO 또는 LSO가 적용되는 색 성분에 대해 다수(예컨대, M개)의 LUT가 시그널링되고, 현재 블록에 적용하기 위해 선택된 LUT의 인덱스도 또한 시그널링된다. M의 예시적인 값은 1 내지 1024 내의 임의의 정수이다.

[0268] 일부 예시적인 구현에서, 다수(예컨대, M개)의 LUT 및 (미리 정의되지 않는다면) 실제 LUT는 HLS(APS, 슬라이스 헤더, 프레임 헤더, PPS, SPS, VPS)에서의 CTU/SB 레벨 또는 시퀀스 레벨 또는 픽처 레벨을 포함하지만 이에 제한되는 것은 아닌 다양한 레벨에서 시그널링될 수 있다. 현재 블록에 대해 선택된 LUT의 인덱스는 HLS(APS, 슬라이스 헤더, 프레임 헤더, PPS, SPS, VPS)의 CTU/SB 레벨 또는 코딩된 블록 레벨 또는 필터 단위 레벨을 포함하지만 이에 제한되지 않는 다양한 레벨에서 시그널링될 수 있다.

[0269] CCSO 또는 LSO 필터에 관한 "루프 필터링 파라미터"라는 용어는 본 개시내용에서 사용될 때, (예컨대, 미리 정의된 또는 인코더 도출된) LUT 타입, LUT에 대한 인덱스, LUT의 델타 레벨 수, 델타에 대한 양자화 단계, 각각의 필터에 대한 탭 수, 각각의 필터에 대한 필터링된 샘플 포지션에 대한 탭 포지션 등을 지시하는 플래그를 포함하지만 이에 제한되지 않는, CCSO 또는 LSO 필터의 다양한 특징을 지시하는 비트스트림에서 시그널링되는 파라미터를 의미한다.

[0270] 도 24는 루프 내 샘플 간 오프셋 필터링 또는 로컬 샘플 오프셋 필터링에 대한 위의 구현의 기초가 되는 원리를 따르는 예시적인 방법의 흐름도(2400)를 도시한다. 2401에서 예시적인 방법 흐름이 시작된다. S2410에서, 비디오 스트림의 현재 재구성된 데이터 블록에서 적어도 제1 색 성분의 재구성된 샘플과 연관된 적어도 하나의 통계적 특성이 획득된다. S2420에서, 적어도 하나의 통계적 특성에 기초하여 복수의 샘플 오프셋 필터 중에서 타깃 샘플 오프셋 필터가 선택되며, 타깃 샘플 오프셋 필터는 샘플 델타 측정과 샘플 오프셋 값 사이의 비선형 매핑을 포함한다. S2430에서, 현재 샘플의 필터링된 재구성된 샘플을 생성하기 위해, 현재 재구성된 데이터 블록의 제3 색 성분의 참조 샘플 및 타깃 샘플 오프셋 필터를 사용하여 현재 재구성된 데이터 블록의 제2 색 성분의 현재 샘플이 필터링된다. S2499에서 예시적인 방법 흐름이 끝난다.

[0271] 본 개시내용의 실시예는 개별적으로 사용될 수 있거나 임의의 순서로 조합될 수 있다. 추가로, 방법(또는 실시예), 인코더 및 디코더 각각은 처리 회로(예컨대, 하나 이상의 프로세서 또는 하나 이상의 집적 회로)에 의해 구현될 수 있다. 일례로, 하나 이상의 프로세서는 비-일시적 컴퓨터 판독 가능 매체에 저장된 프로그램을 실행한다. 본 개시내용의 실시예는 루마 블록 또는 크로마 블록에 적용될 수 있다.

[0272] 위에서 설명된 기법들은 컴퓨터 판독 가능 명령을 사용하여 컴퓨터 소프트웨어로서 구현될 수 있고 하나 이상의 컴퓨터 판독 가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 25는 개시된 청구 대상의 특정 실시예를 구현하기에 적합한 컴퓨터 시스템(2500)을 도시한다.

[0273] 컴퓨터 소프트웨어는 하나 이상의 컴퓨터 중앙 처리 유닛(CPU: central processing unit), 그래픽 처리 유닛(GPU: Graphics Processing Unit) 등에 의해, 직접적으로 또는 해석, 마이크로코드 실행 등을 통해 실행될 수 있는 명령을 포함하는 코드를 생성하기 위해 어셈블리, 컴파일, 링크 또는 유사한 메커니즘을 거칠 수 있는 임의의 적절한 기계 코드 또는 컴퓨터 언어를 사용하여 코딩될 수 있다.

[0274] 명령은 예를 들어 개인용 컴퓨터, 태블릿 컴퓨터, 서버, 스마트폰, 게임 디바이스, 사물 인터넷 디바이스 등을 포함하는 다양한 타입의 컴퓨터 또는 이들의 컴포넌트 상에서 실행될 수 있다.

[0275] 컴퓨터 시스템(2500)에 대해 도 25에 도시된 컴포넌트는 본질적으로 예시이며, 본 개시내용의 실시예를 구현하



는 컴퓨터 소프트웨어의 사용 또는 기능의 범위에 대해 어떠한 제한도 제안하도록 의도되지 않는다. 컴포넌트의 구성이 컴퓨터 시스템(2500)의 예시적인 실시예에 예시된 컴포넌트 중 임의의 컴포넌트 또는 그러한 컴포넌트의 조합과 관련된 어떠한 의존성이나 요건도 갖는 것으로 해석되지 않아야 한다.

[0276] 컴퓨터 시스템(2500)은 특정 인간 인터페이스 입력 디바이스를 포함할 수 있다. 그러한 인간 인터페이스 입력 디바이스는 예를 들어, 촉각 입력(이를테면: 키 스트로크(keystroke), 스와이프(swipe), 데이터 글러브(glove) 움직임), 오디오 입력(이를테면: 음성, 박수), 시각적 입력(이를테면: 제스처), 후각적 입력(도시되지 않음)을 통해 하나 이상의 인간 사용자에게 의한 입력에 응답할 수 있다. 인간 인터페이스 디바이스는 또한, 오디오(이를테면: 스피치, 음악, 주변 사운드), 이미지(이를테면: 스캔된 이미지, 정지 이미지 카메라로부터 획득된 사진 이미지), 비디오(이를테면, 2차원 비디오, 입체 비디오를 포함하는 3차원 비디오)와 같은, 인간에 의한 의식적 입력과 반드시 직접적으로 관련되는 것은 아닌 특정 미디어를 캡처하는 데 사용될 수 있다.

[0277] 입력 인간 인터페이스 디바이스는: 키보드(2501), 마우스(2502), 트랙 패드(2503), 터치 스크린(2510), (도시되지 않은) 데이터 글러브, 조이스틱(2505), 마이크로폰(2506), 스캐너(2507), 카메라(2508) 중 하나 이상(각각 도시된 것 중 단 하나)을 포함할 수 있다.

[0278] 컴퓨터 시스템(2500)은 또한 특정 인간 인터페이스 출력 디바이스를 포함할 수 있다. 그러한 인간 인터페이스 출력 디바이스는 예를 들어, 촉각 출력, 사운드, 광 및 후각/미각을 통해 하나 이상의 인간 사용자의 감각을 자극할 수 있다. 그러한 인간 인터페이스 출력 디바이스는 촉각 출력 디바이스(예를 들어, 터치 스크린(2510), (도시되지 않은) 데이터 글러브 또는 조이스틱(2505)에 의한 촉각 피드백, 그러나 입력 디바이스로서 기능하지 않는 촉각 피드백 디바이스가 또한 존재할 수 있음), 오디오 출력 디바이스(이를테면: 스피커(2509), (도시되지 않은) 헤드폰들), 시각적 출력 디바이스(이를테면, 각각 터치 스크린 입력 능력을 갖거나 갖지 않고, 각각 촉각적 피드백 능력을 갖거나 갖지 않는 CRT 스크린, LCD 스크린, 플라즈마 스크린, OLED 스크린을 포함하는 스크린(2510) — 이들 중 일부는 스테레오그래픽 출력과 같은 수단을 통해 2차원 시각적 출력 또는 3차원 이상의 출력을 출력하는 것이 가능할 수 있음 —; (도시되지 않은) 가상 현실 안경, (도시되지 않은) 홀로그래픽 디스플레이 및 연기 탱크), 및 (도시되지 않은) 프린터를 포함할 수 있다.

[0279] 컴퓨터 시스템(2500)은 또한, 인간 액세스 가능 저장 디바이스 및 이들의 연관된 매체, 이를테면 CD/DVD 또는 유사한 매체(2521)를 갖는 CD/DVD ROM/RW(2520), 썸 드라이브(thumb-drive)(2522), 착탈식 하드 드라이브 또는 솔리드 스테이트 드라이브(2523), 레저시 자기 매체, 이를테면 (도시되지 않은) 테이프 및 플로피 디스크, 특수 ROM/ASIC/PLD 기반 디바이스, 이를테면 (도시되지 않은) 보안 동글(security dongle) 등을 포함하는 광학 매체를 포함할 수 있다.

[0280] 당해 기술분야에서 통상의 지식을 가진 자는 또한, 현재 개시된 청구 대상과 관련하여 사용되는 바와 같은 "컴퓨터 판독 가능 매체"라는 용어가 전송 매체, 반송파, 또는 다른 일시적인 신호를 포괄하지 않는다고 이해해야 한다.

[0281] 컴퓨터 시스템(2500)은 또한 하나 이상의 통신 네트워크(2555)에 대한 인터페이스(2554)를 포함할 수 있다. 네트워크는 예를 들어 무선, 유선, 광일 수 있다. 네트워크는 추가로, 근거리, 광역, 대도시, 차량 및 산업, 실시간, 지연 용인(delay-tolerant) 등일 수 있다. 네트워크의 예는 근거리 네트워크, 이를테면 이더넷, 무선 LAN, GSM, 3G, 4G, 5G, LTE 등을 포함하는 셀룰러 네트워크, 케이블 TV, 위성 TV 및 지상 브로드캐스트 TV를 포함하는 TV 유선 또는 무선 광역 디지털 네트워크, CANBus를 포함하는 차량 및 산업용 등을 포함한다. 특정 네트워크는 일반적으로, (예를 들어, 컴퓨터 시스템(2500)의 USB 포트와 같은) 특정 범용 데이터 포트 또는 주변 버스(2549)에 부착된 외부 네트워크 인터페이스 어댑터를 필요로 하며; 다른 것은 일반적으로, 아래에서 설명되는 바와 같은 시스템 버스에 대한 부착(예를 들어, PC 컴퓨터 시스템으로의 이더넷 인터페이스 또는 스마트폰 컴퓨터 시스템으로의 셀룰러 네트워크 인터페이스)에 의해 컴퓨터 시스템(2500)의 코어에 통합된다. 이러한 네트워크 중 임의의 네트워크를 사용하여, 컴퓨터 시스템(2500)은 다른 엔티티와 통신할 수 있다. 그러한 통신은 예를 들어, 근거리 또는 광역 디지털 네트워크를 사용하는 다른 컴퓨터 시스템에 대해 단방향, 수신 전용(예를 들어, 브로드캐스트 TV), 단방향 송신 전용(예를 들어, 특정 CANbus 디바이스에 대한 CANbus), 또는 양방향일 수 있다. 위에서 설명된 바와 같이, 특정 프로토콜 및 프로토콜 스택이 그러한 네트워크 및 네트워크 인터페이스 각각에 사용될 수 있다.

[0282] 앞서 언급한 인간 인터페이스 디바이스, 인간 액세스 가능 저장 디바이스 및 네트워크 인터페이스는 컴퓨터 시스템(2500)의 코어(2540)에 부착될 수 있다.

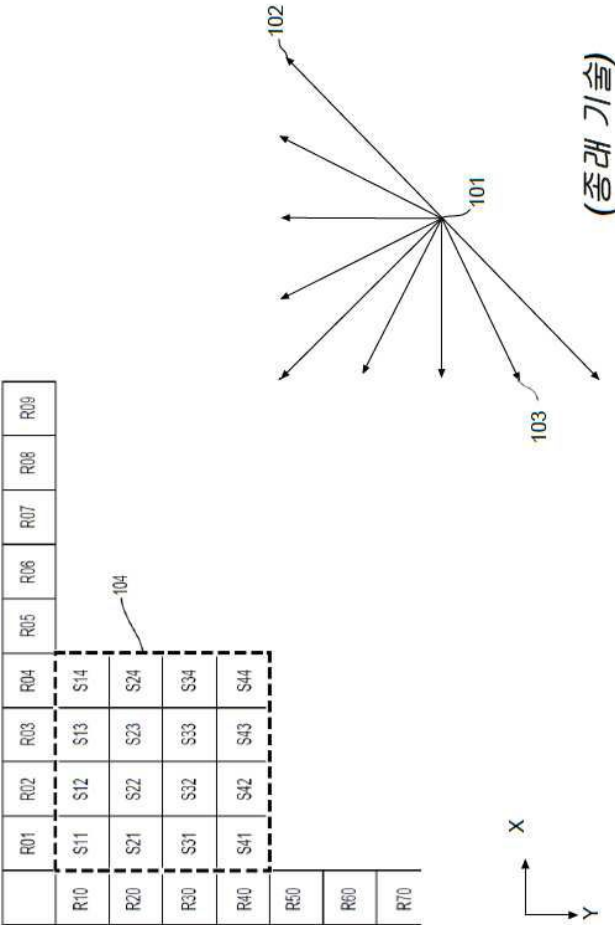
- [0283] 코어(2540)는 하나 이상의 중앙 처리 유닛(CPU)(2541), 그래픽 처리 유닛(GPU)(2542), 필드 프로그래밍 가능 게이트 영역(FPGA: Field Programmable Gate Area)(2543) 형태의 특수 프로그래밍 가능 처리 유닛, 특정 작업을 위한 하드웨어 가속기(2544), 그래픽 어댑터(2550) 등을 포함할 수 있다. 이러한 디바이스는 판독 전용 메모리(ROM: Read-only memory)(2545), 랜덤 액세스 메모리(2546), 내부 비-사용자 액세스 가능 하드 드라이브, SSD 등과 같은 내부 대용량 저장소(2547)와 함께, 시스템 버스(2548)를 통해 접속될 수 있다. 일부 컴퓨터 시스템에서, 시스템 버스(2548)는 추가 CPU, GPU 등에 의한 확장을 가능하게 하도록 하나 이상의 물리적 플러그의 형태로 액세스 가능할 수 있다. 주변 디바이스는 코어의 시스템 버스(2548)에 직접적으로 또는 주변 버스(2549)를 통해 부착될 수 있다. 일례로, 스크린(2510)은 그래픽 어댑터(2550)에 접속될 수 있다. 주변 버스를 위한 아키텍처는 PCI, USB 등을 포함한다.
- [0284] CPU(2541), GPU(2542), FPGA(2543) 및 가속기(2544)는 조합하여, 앞서 언급된 컴퓨터 코드를 구성할 수 있는 특정 명령을 실행할 수 있다. 그 컴퓨터 코드는 ROM(2545) 또는 RAM(2546)에 저장될 수 있다. 전이(transitional) 데이터가 또한 RAM(2546)에 저장될 수 있는 반면, 영구 데이터는 예를 들어, 내부 대용량 저장소(2547)에 저장될 수 있다. 메모리 디바이스 중 임의의 디바이스에 대한 고속 저장 및 리트리브는 하나 이상의 CPU(2541), GPU(2542), 대용량 저장소(2547), ROM(2545) 및 RAM(2546) 등과 밀접하게 연관될 수 있는 캐시 메모리의 사용을 통해 가능해질 수 있다.
- [0285] 컴퓨터 판독 가능 매체는 다양한 컴퓨터 구현 동작을 수행하기 위한 컴퓨터 코드를 가질 수 있다. 매체 및 컴퓨터 코드는 본 개시내용의 목적으로 특별히 설계 및 구성된 것일 수 있거나, 이들은 컴퓨터 소프트웨어 분야에서 통상의 지식을 가진 자에게 잘 알려져 있고 이용 가능한 종류일 수 있다.
- [0286] 비-제한적인 예로서, 아키텍처(2500)를 갖는 컴퓨터 시스템, 그리고 구체적으로 코어(2540)는 (CPU, GPU, FPGA, 가속기 등을 포함하는) 프로세서(들)가 하나 이상의 유형(tangible) 컴퓨터 판독 가능 매체에 구현된 소프트웨어를 실행하는 결과로서 기능을 제공할 수 있다. 그러한 컴퓨터 판독 가능 매체는 위에서 소개된 바와 같은 사용자 액세스 가능 대용량 저장소뿐만 아니라, 비-일시적 성질의 코어(2540)의 특정 저장소, 이를테면 코어 내부 대용량 저장소(2547) 또는 ROM(2545)과 연관된 매체일 수 있다. 본 개시내용의 다양한 실시예를 구현하는 소프트웨어는 그러한 디바이스에 저장되고 코어(2540)에 의해 실행될 수 있다. 컴퓨터 판독 가능 매체는 특정 요구에 따라 하나 이상의 메모리 디바이스 또는 칩을 포함할 수 있다. 소프트웨어는 코어(2540) 및 구체적으로는 코어(3040) 내의 (CPU, GPU, FPGA 등을 포함하는) 프로세서로 하여금, RAM(2546)에 저장된 데이터 구조를 정의하는 것과 소프트웨어에 의해 정의된 프로세스에 따라 그러한 데이터 구조를 수정하는 것을 포함하여, 본 명세서에서 설명된 특정 프로세스 또는 특정 프로세스의 특정 부분을 실행하게 할 수 있다. 추가로 또는 대안으로서, 컴퓨터 시스템은 본 명세서에서 설명되는 특정 프로세스 또는 특정 프로세스의 특정 부분을 실행하도록 소프트웨어 대신 또는 소프트웨어와 함께 동작할 수 있는 회로(예를 들어: 가속기(2544))에 하드와이어링되거나 달리 구현된 로직의 결과로서 기능을 제공할 수 있다. 적절한 경우, 소프트웨어에 대한 참조는 로직을 포함할 수 있고, 그 반대로 가능하다. 컴퓨터 판독 가능 매체에 대한 참조는 적절한 경우, 실행을 위한 소프트웨어를 저장하는 회로(이를테면, 집적 회로(IC: integrated circuit)), 실행을 위한 로직을 구현하는 회로, 또는 이 둘 모두를 포괄할 수 있다. 본 개시내용은 하드웨어와 소프트웨어의 임의의 적절한 조합을 포괄한다.
- [0287] 본 개시내용은 여러 예시적인 실시예를 설명했지만, 본 개시내용의 범위 내에 속하는 변경, 치환 및 다양한 대체 등가물이 있다. 위에서 설명된 구현 및 실시예에서, 프로세스의 임의의 동작이 원하는 대로, 임의의 양 또는 순서로 조합 또는 정렬될 수 있다. 추가로, 위에서 설명된 프로세스의 동작 중 2개 이상은 병렬로 수행될 수 있다. 따라서 당해 기술분야에서 통상의 지식을 가진 자는, 본 명세서에서 명시적으로 도시되거나 설명되지는 않지만, 본 개시내용의 원리를 구현하고 따라서 본 개시내용의 사상 및 범위 내에 있는 다수의 시스템 및 방법을 안출할 수 있을 것이라고 인식될 것이다.
- [0288] 부록 A: 약어
- [0289] JEM: 공동 탐사 모델(joint exploration model)
- [0290] VVC: 다용도 비디오 코딩(versatile video coding)
- [0291] BMS: 벤치마크 세트(benchmark set)
- [0292] MV: 움직임 벡터(Motion Vector)
- [0293] HEVC: 고효율 비디오 코딩(High Efficiency Video Coding)

[0294]	SEI: 부가 확장 정보(Supplementary Enhancement Information)
[0295]	VUI: 비디오 사용성 정보(Video Usability Information)
[0296]	GOP: 픽처의 그룹(Group of Pictures)
[0297]	TU: 변환 유닛(Transform Unit)
[0298]	PU: 예측 유닛(Prediction Unit)
[0299]	CTU: 코딩 트리 유닛(Coding Tree Unit)
[0300]	CTB: 코딩 트리 블록(Coding Tree Block)
[0301]	PB: 예측 블록(Prediction Block)
[0302]	HRD: 가상 참조 디코더(Hypothetical Reference Decoder)
[0303]	SNR: 신호 잡음비(Signal Noise Ratio)
[0304]	CPU: 중앙 처리 유닛(Central Processing Unit)
[0305]	GPU: 그래픽 처리 유닛(Graphics Processing Unit)
[0306]	CRT: 음극선관(Cathode Ray Tube)
[0307]	LCD: 액정 디스플레이(Liquid-Crystal Display)
[0308]	OLED: 유기 발광 다이오드(Organic Light-Emitting Diode)
[0309]	CD: 콤팩트 디스크(Compact Disc)
[0310]	DVD: 디지털 비디오 디스크(Digital Video Disc)
[0311]	ROM: 판독 전용 메모리(Read-Only Memory)
[0312]	RAM: 랜덤 액세스 메모리(Random Access Memory)
[0313]	ASIC: 주문형 집적 회로(Application-Specific Integrated Circuit)
[0314]	PLD: 프로그래밍 가능한 로직 디바이스(Programmable Logic Device)
[0315]	LAN: 근거리 네트워크(Local Area Network)
[0316]	GSM: 글로벌 모바일 통신 시스템(Global System for Mobile communications)
[0317]	LTE: 롱 텀 에볼루션(Long-Term Evolution)
[0318]	CANBus: 제어기 영역 네트워크 버스(Controller Area Network Bus)
[0319]	USB: 범용 직렬 버스(Universal Serial Bus)
[0320]	PCI: 주변 컴포넌트 상호 접속(Peripheral Component Interconnect)
[0321]	FPGA: 필드 프로그래밍 가능 게이트 영역(Field Programmable Gate Areas)
[0322]	SSD: 솔리드 스테이트 드라이브(solid-state drive)
[0323]	IC: 집적 회로(Integrated Circuit)
[0324]	HDR: 높은 동적 범위(standard dynamic range)
[0325]	SDR: 표준 동적 범위(standard dynamic range)
[0326]	JVET: 공동 비디오 탐사 팀(Joint Video Exploration Team)
[0327]	MPM: 가장 가능성 있는 모드(most probable mode)
[0328]	WAIP: 광각 인트라 예측(Wide-Angle Intra Prediction)
[0329]	CU: 코딩 유닛(Coding Unit)

- [0330] PU: 예측 유닛(Prediction Unit)
- [0331] TU: 변환 유닛(Transform Unit)
- [0332] CTU: 코딩 트리 유닛(Coding Tree Unit)
- [0333] PDP: 포지션 의존 예측 조합(Position Dependent Prediction Combination)
- [0334] ISP: 인트라 서브 파티션(Intra Sub-Partitions)
- [0335] SPS: 시퀀스 파라미터 설정(Sequence Parameter Setting)
- [0336] PPS: 픽처 파라미터 세트(Picture Parameter Set)
- [0337] APS: 적응 파라미터 세트(Adaptation Parameter Set)
- [0338] VPS: 비디오 파라미터 세트(Video Parameter Set)
- [0339] DPS: 디코딩 파라미터 세트(Decoding Parameter Set)
- [0340] ALF: 적응 루프 필터(Adaptive Loop Filter)
- [0341] SAO: 샘플 적응 오프셋(Sample Adaptive Offset)
- [0342] CC-ALF: 성분 간 적응 루프 필터(Cross-Component Adaptive Loop Filter)
- [0343] CDEF: 제한된 방향 향상 필터(Constrained Directional Enhancement Filter)
- [0344] CCSO: 성분 간 샘플 오프셋(Cross-Component Sample Offset)
- [0345] LSO: 로컬 샘플 오프셋(Local Sample Offset)
- [0346] LR: 루프 복원 필터(Loop Restoration Filter)
- [0347] AV1: AOMedia 비디오 1(AOMedia Video 1)
- [0348] AV2: AOMedia 비디오 2(AOMedia Video 2)

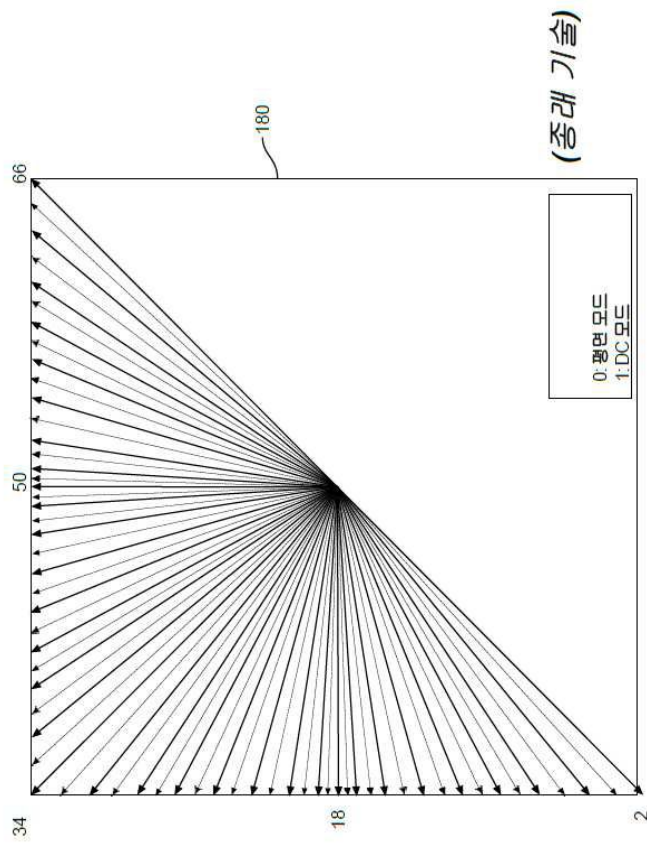
도면

도면1a

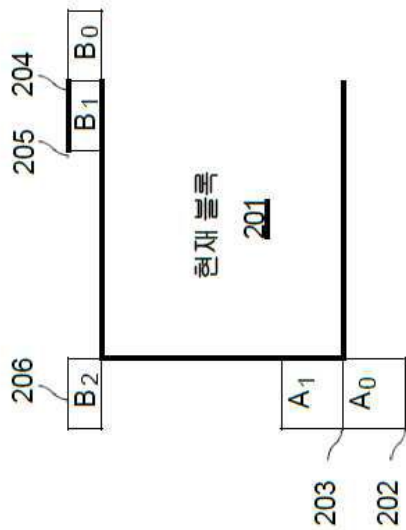




도면1b

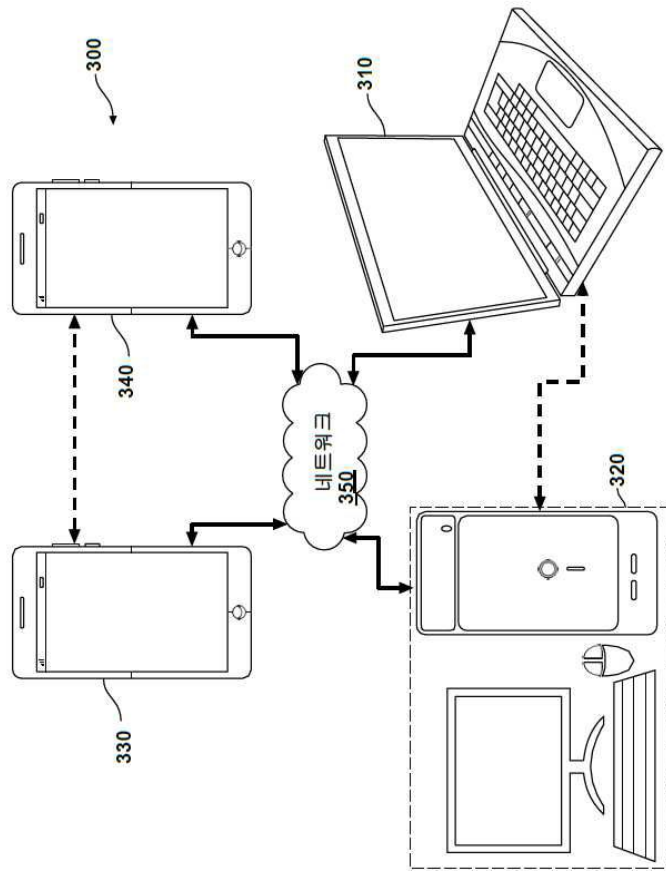


도면2

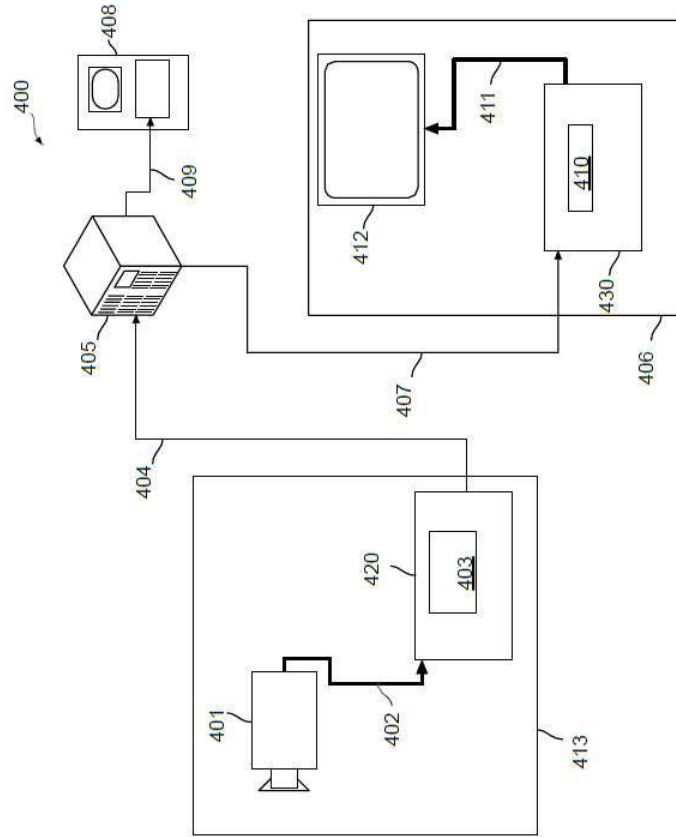


(종래 기술)

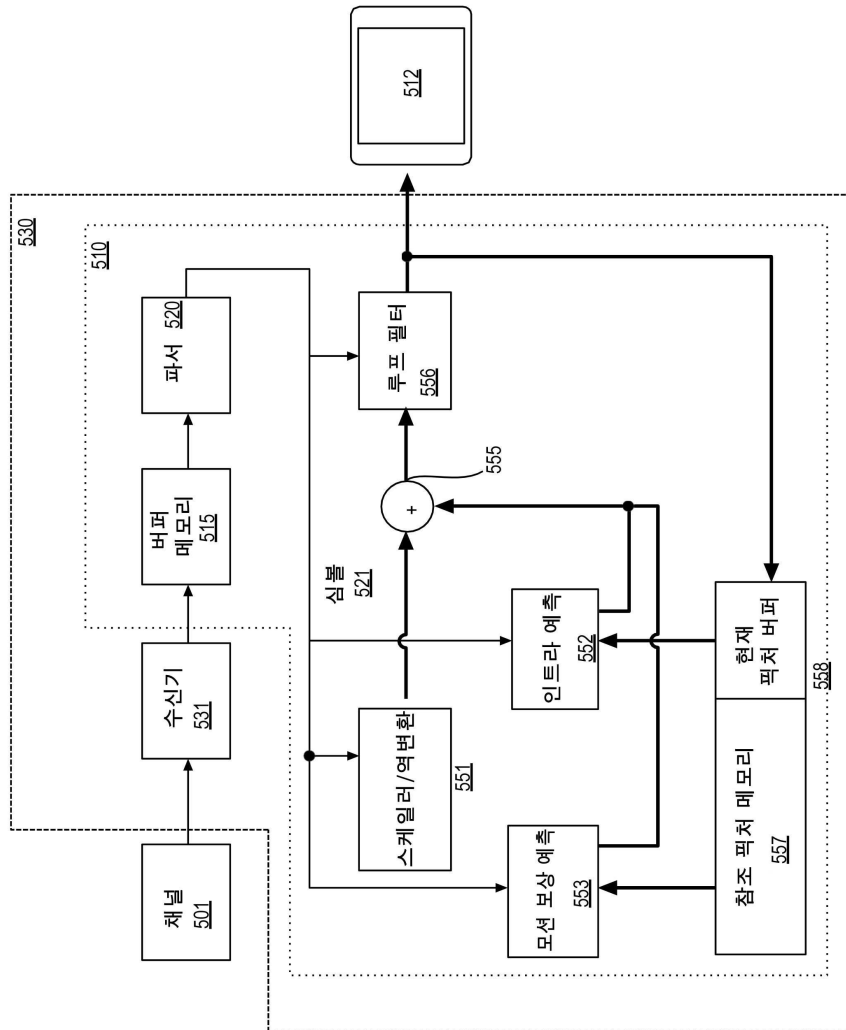
도면3



도면4

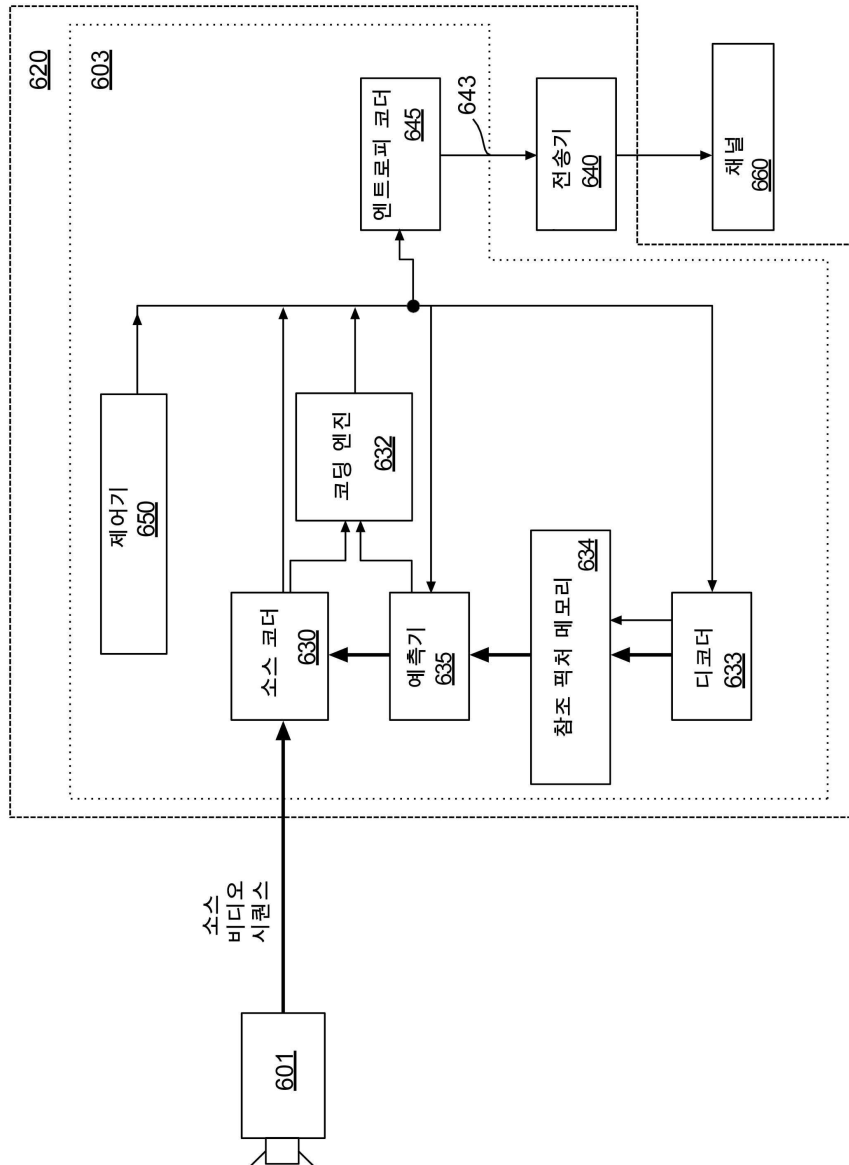


도면5

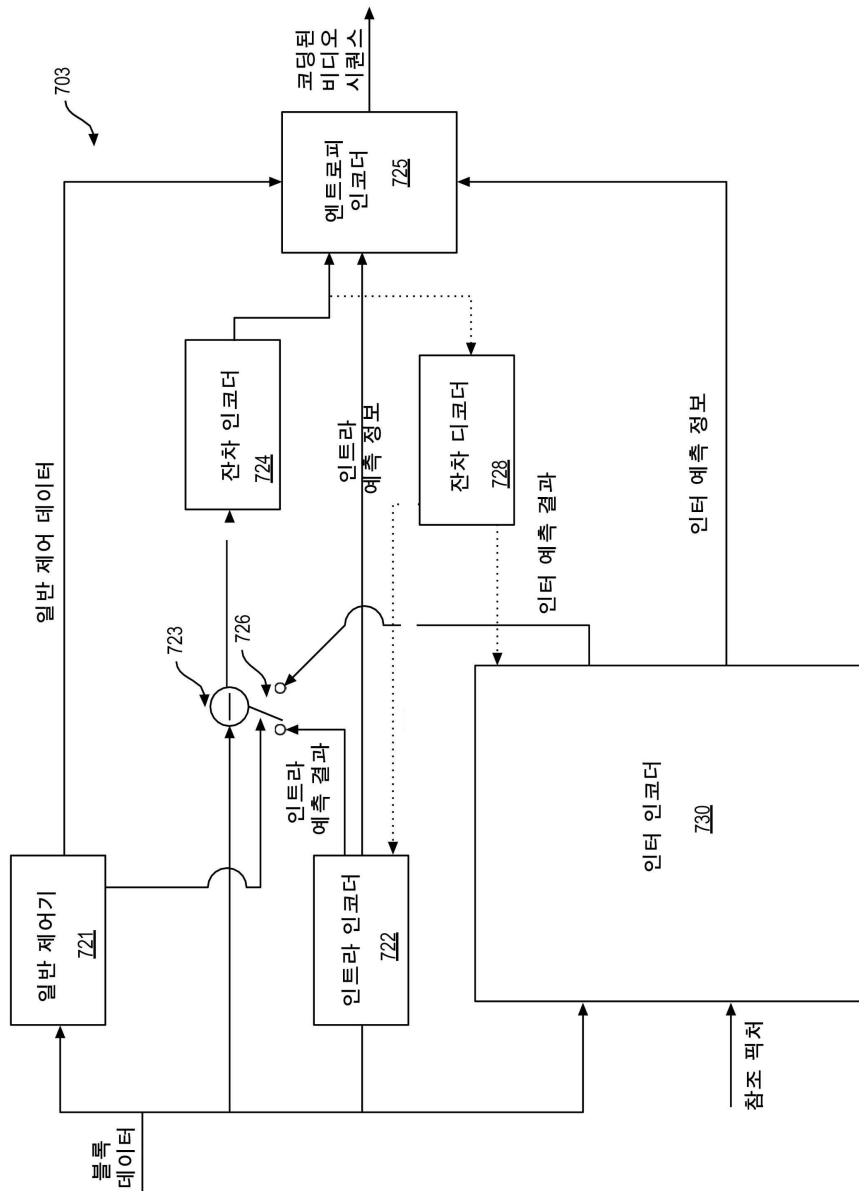




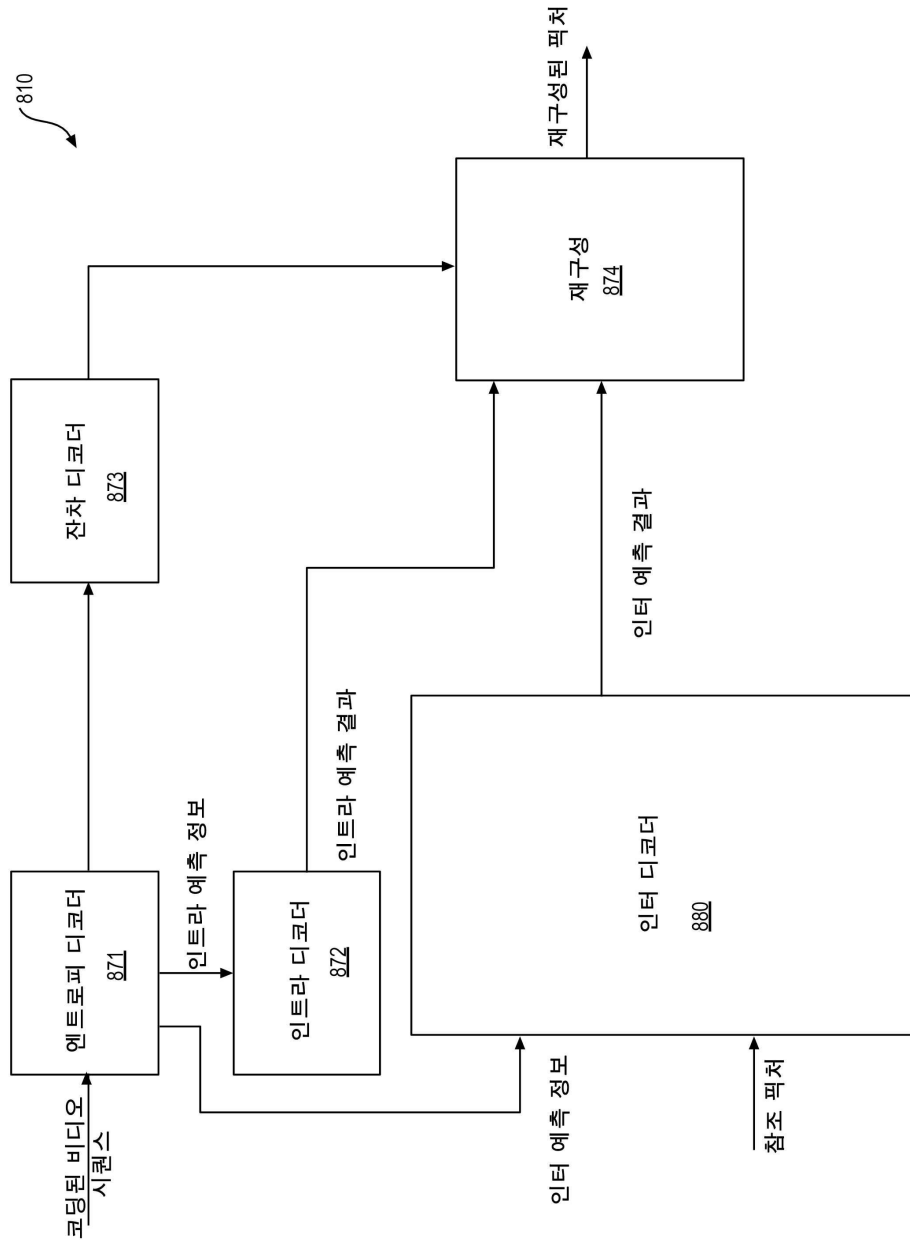
도면6



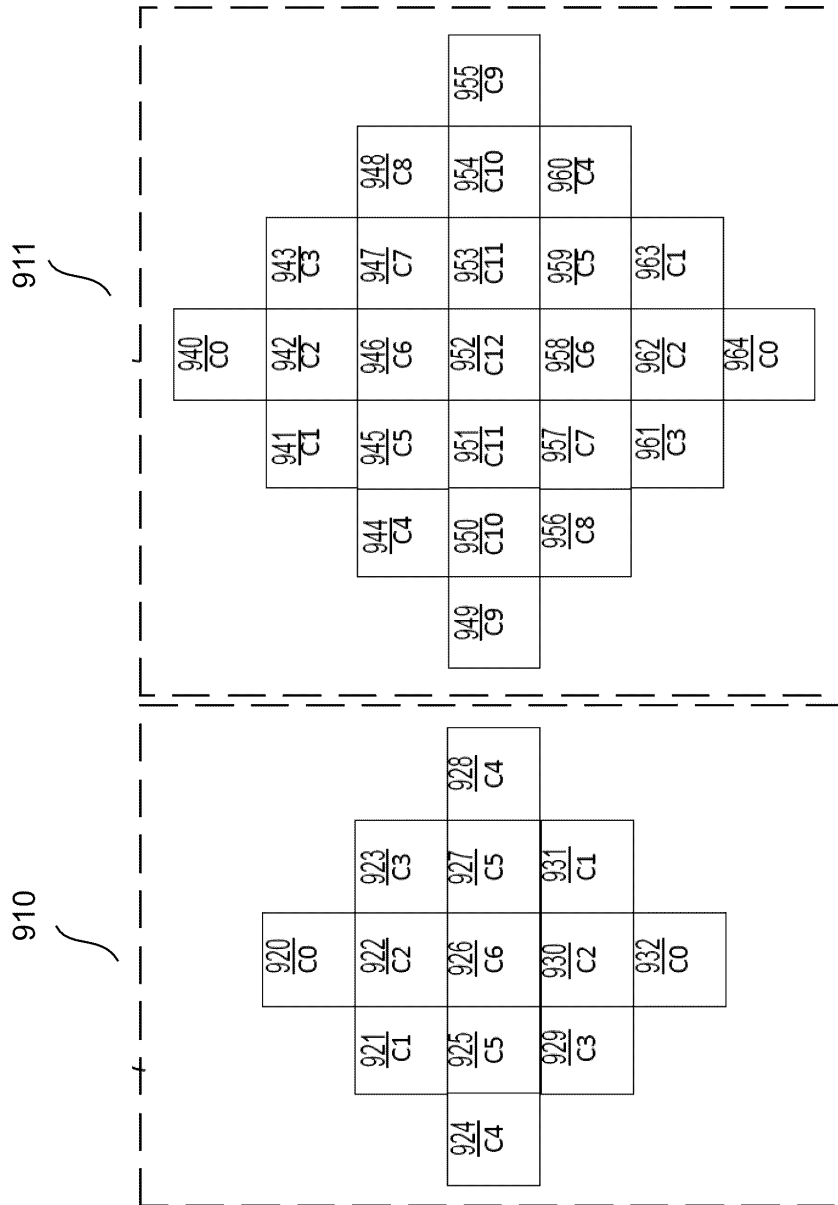
도면7



도면8



도면9



도면10a

	✓		✓		✓		✓
✓		✓		✓		✓	
	✓		✓		✓		✓
✓						✓	
	✓		✓		✓		✓
✓						✓	
	✓		✓		✓		✓
✓		✓		✓		✓	

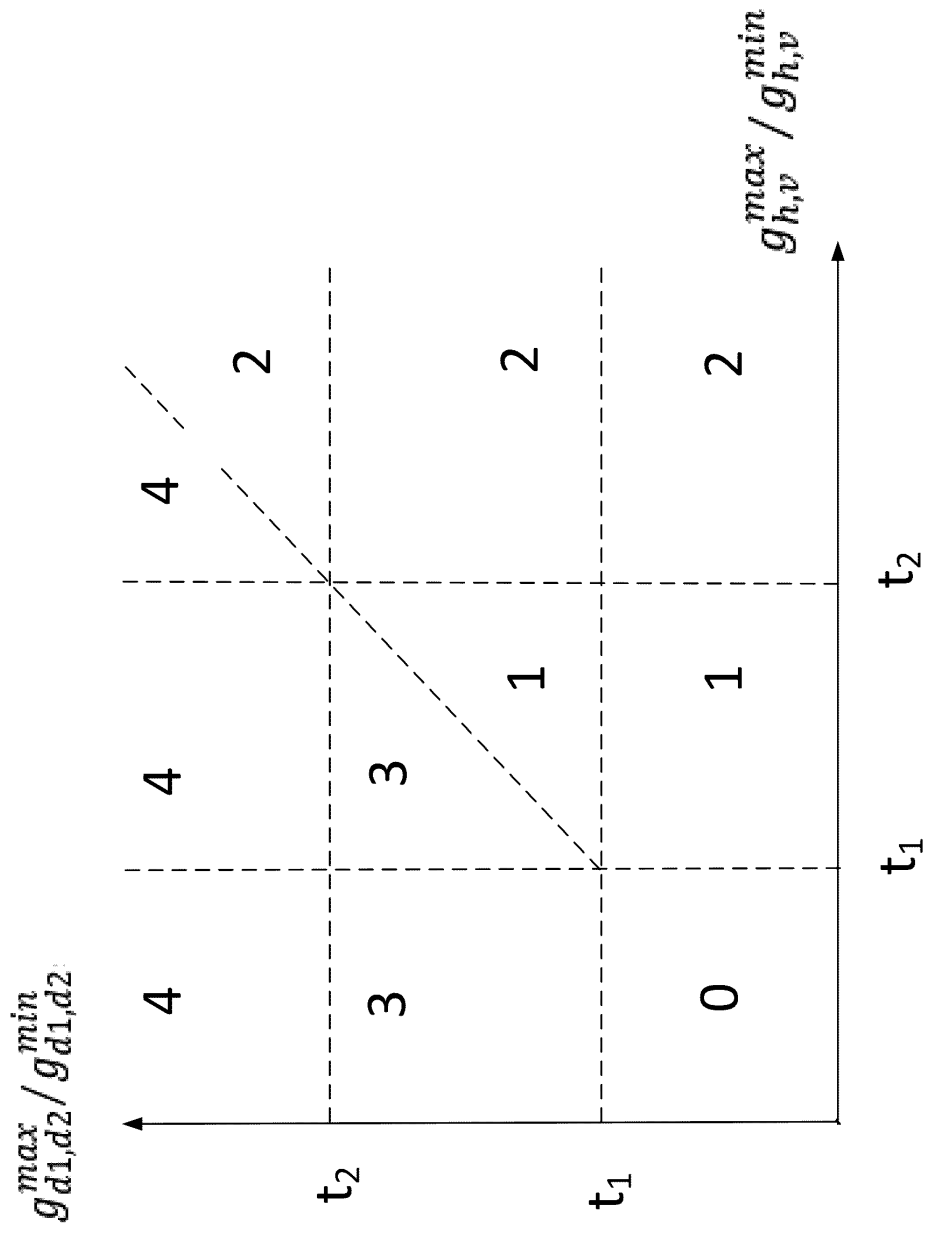
도면10b

	H		H		H		H
H		H		H		H	
	H		H		H		H
H						H	
	H		H		H		H
H						H	
	H		H		H		H
H		H		H		H	

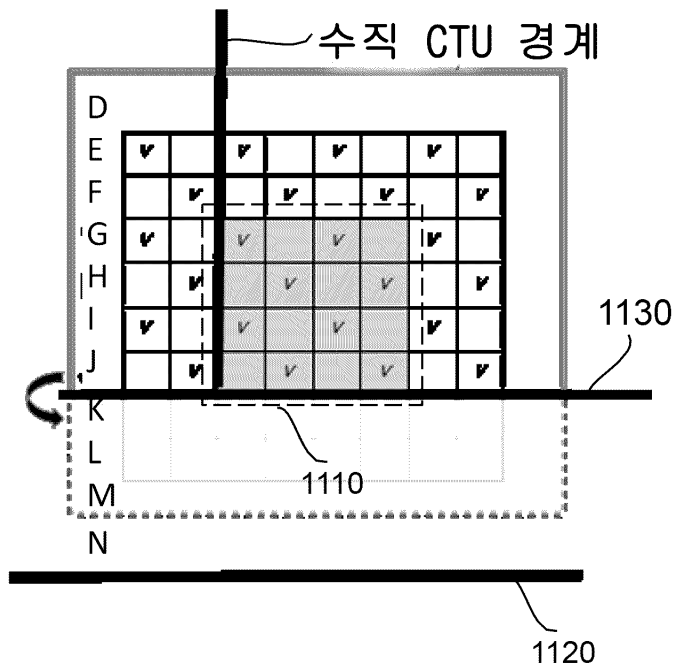




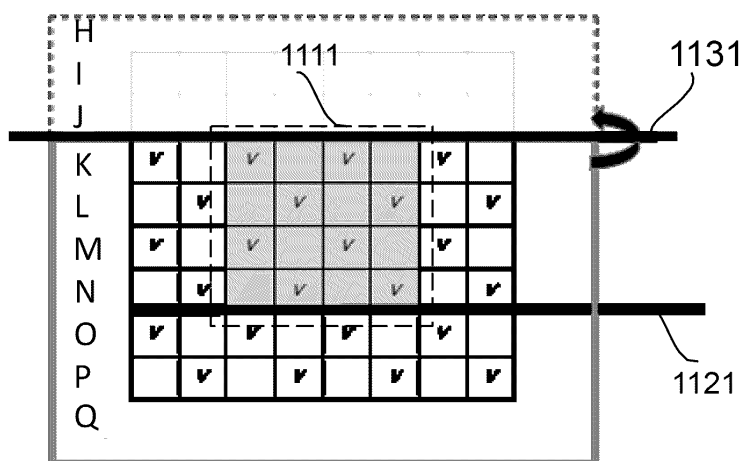
도면10e



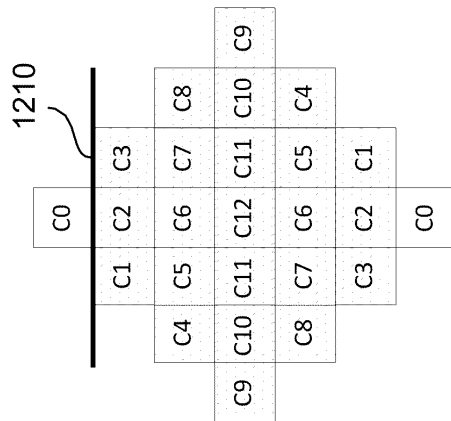
도면11a



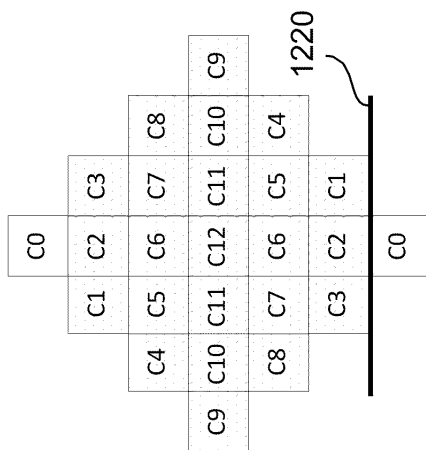
도면11b



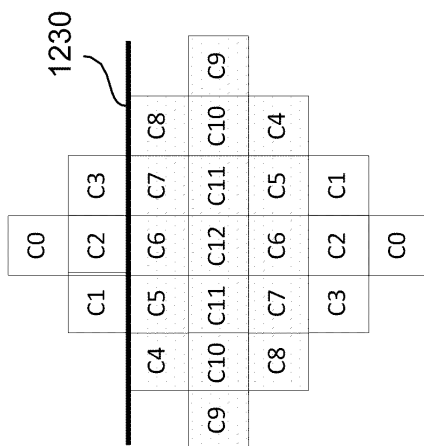
도면12a



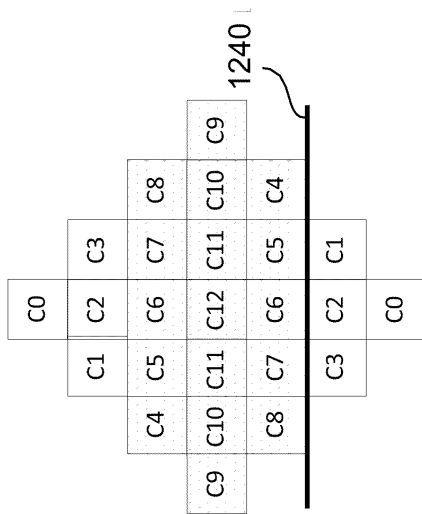
도면12b



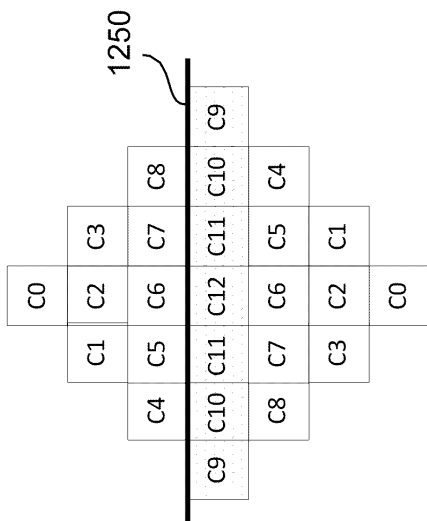
도면12c



도면12d

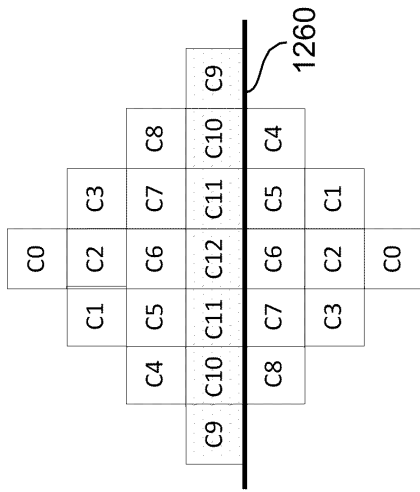


도면12e

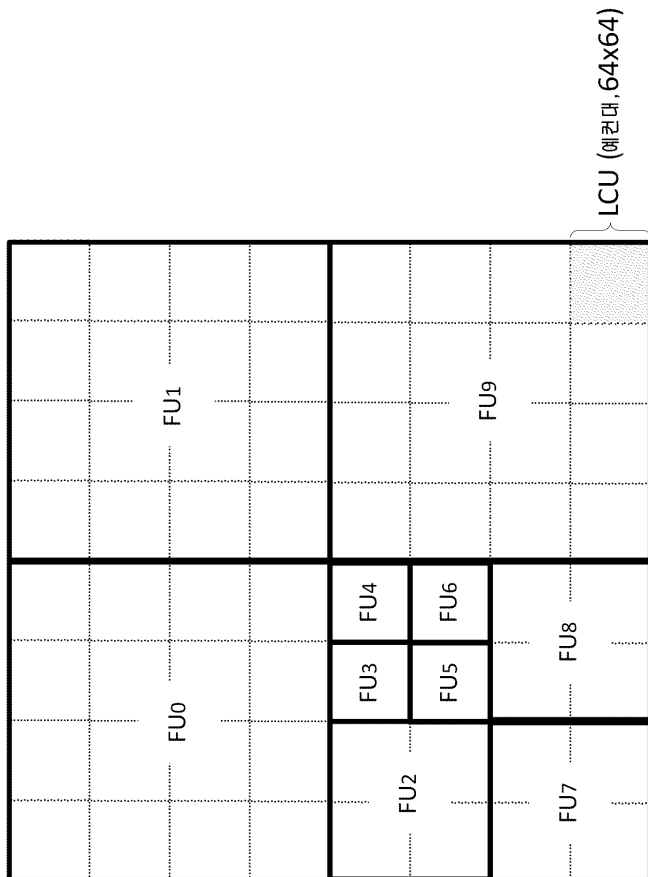




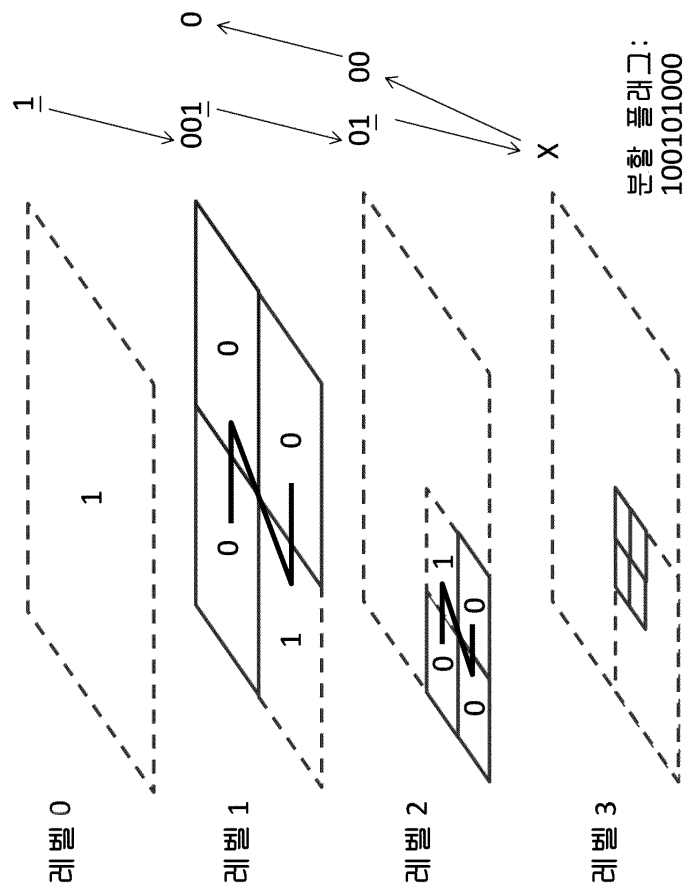
도면12f



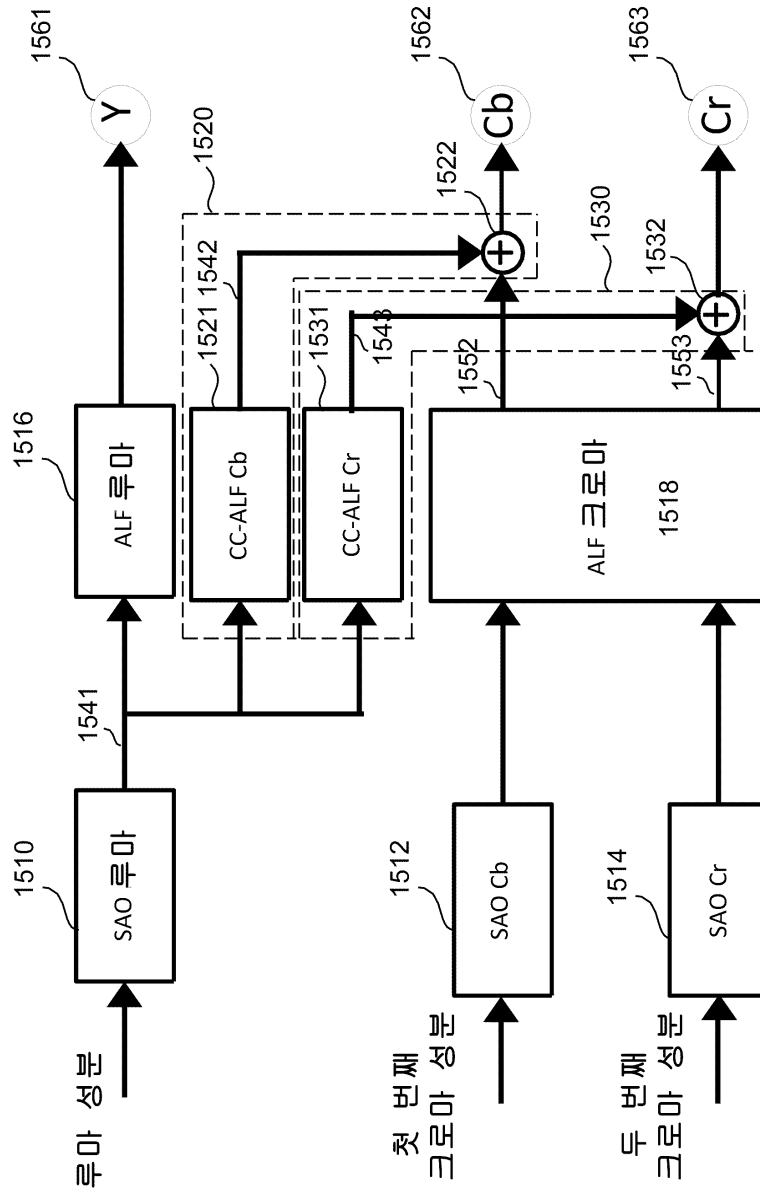
도면13



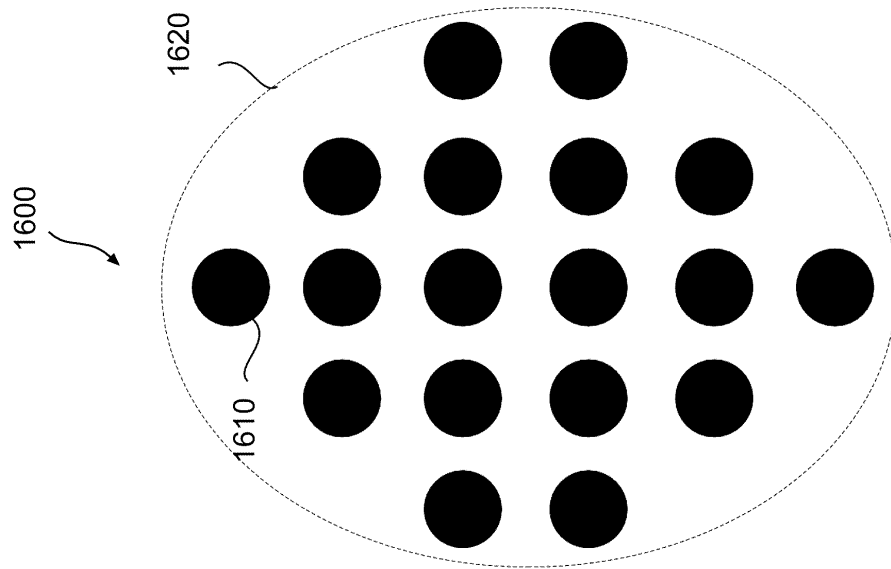
도면14



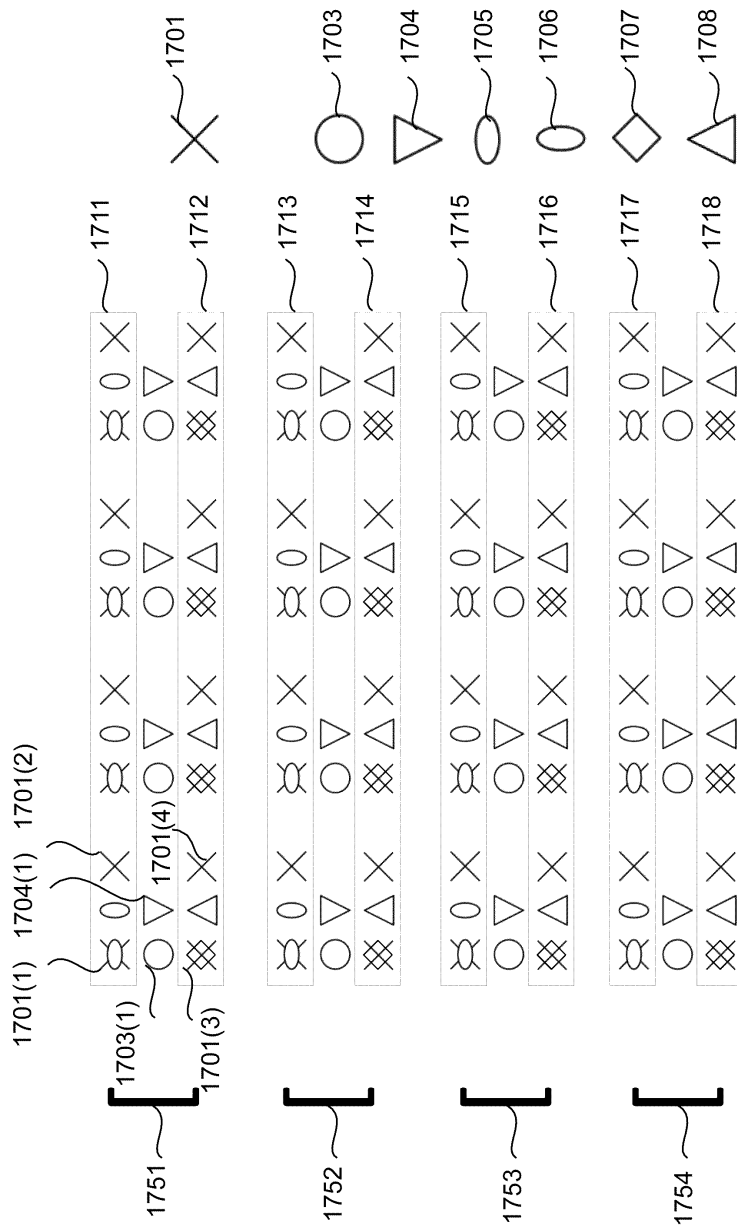
도면15



도면16

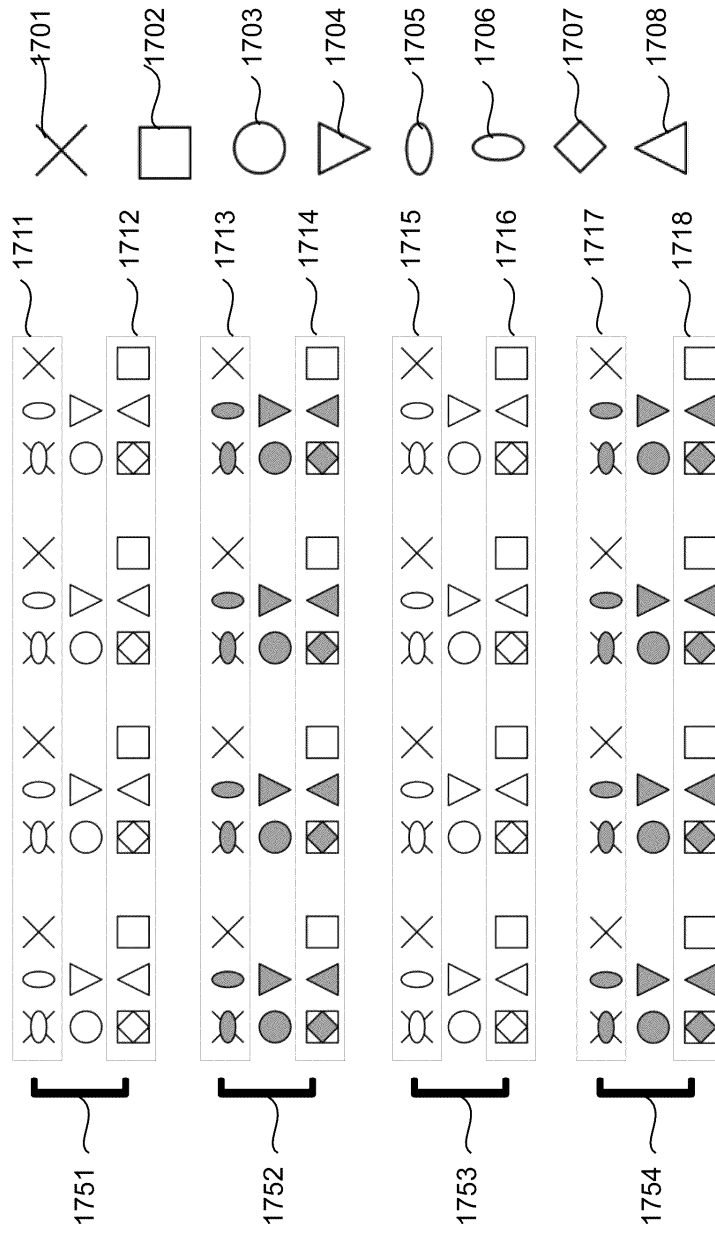


도면17a

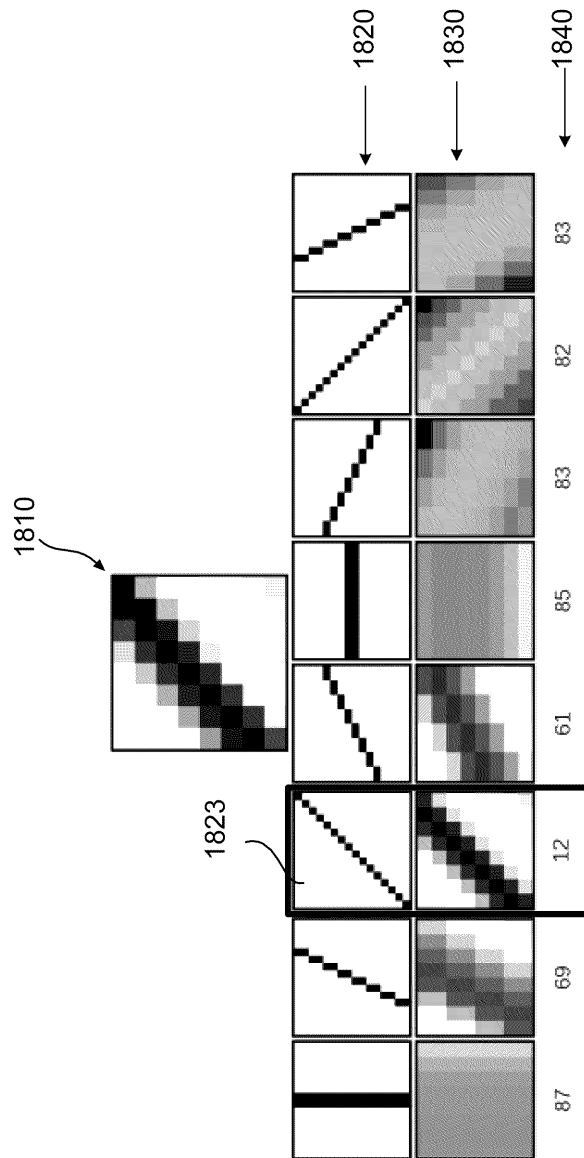




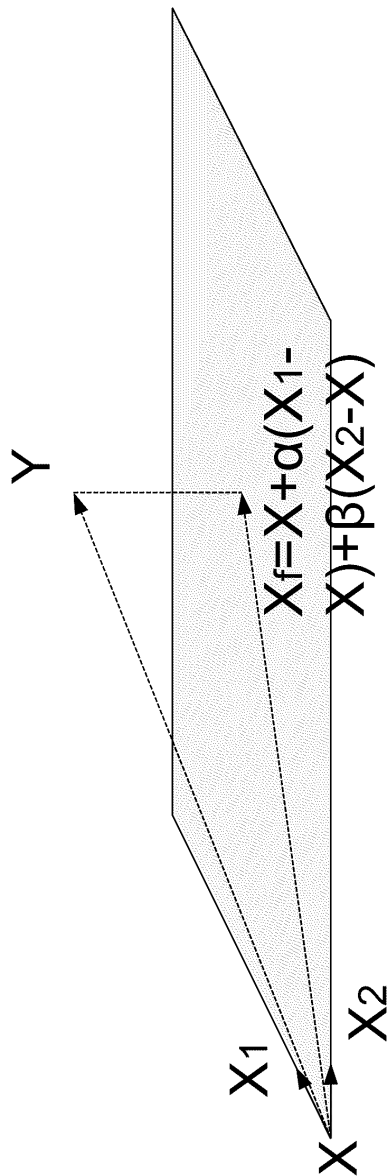
도면17b



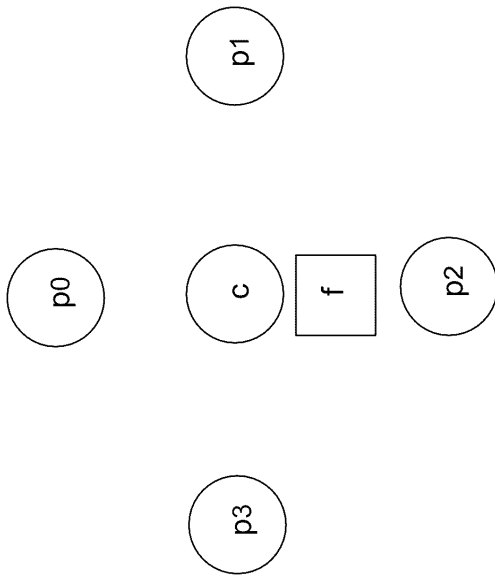
도면18



도면19



도면20



도면21a

조합	d0	d1	d2	d3	오프셋
0	-1	-1	-1	-1	s0
1	-1	-1	-1	0	s1
2	-1	-1	-1	1	s2
3	-1	-1	0	-1	s3
4	-1	-1	0	0	s4
5	-1	-1	0	1	s5
6	-1	-1	1	-1	s6
7	-1	-1	1	0	s7
8	-1	-1	1	1	s8
9	-1	0	-1	-1	s9
10	-1	0	-1	0	s10
11	-1	0	-1	1	s11
12	-1	0	0	-1	s12
13	-1	0	0	0	s13
14	-1	0	0	1	s14
15	-1	0	1	-1	s15
16	-1	0	1	0	s16
17	-1	0	1	1	s17
18	-1	1	-1	-1	s18
19	-1	1	-1	0	s19
20	-1	1	-1	1	s20
21	-1	1	0	-1	s21
22	-1	1	0	0	s22
23	-1	1	0	1	s23
24	-1	1	1	-1	s24
25	-1	1	1	0	s25
26	-1	1	1	1	s26



도면21b

도 21a에서 계속

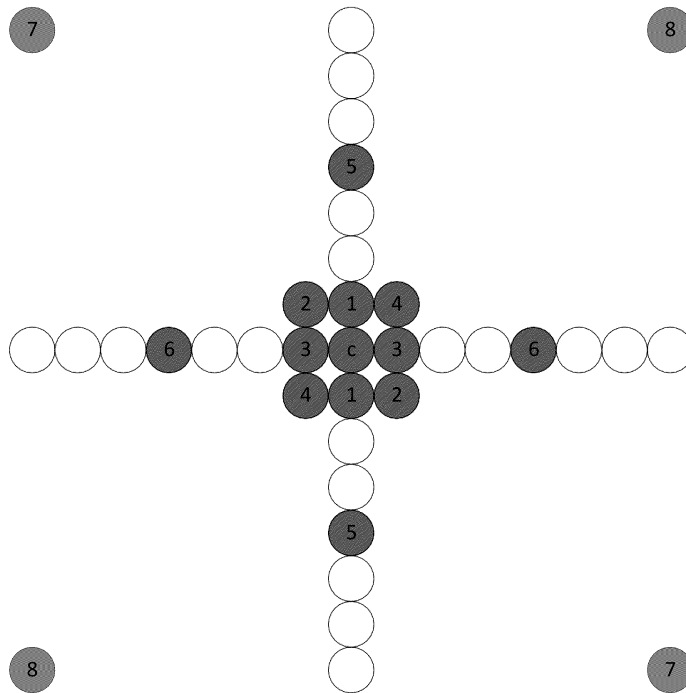
조합	d0	d1	d2	d3	오프셋
27	0	-1	-1	-1	s27
28	0	-1	-1	0	s28
29	0	-1	-1	1	s29
30	0	-1	0	-1	s30
31	0	-1	0	0	s31
32	0	-1	0	1	s32
33	0	-1	1	-1	s33
34	0	-1	1	0	s34
35	0	-1	1	1	s35
36	0	0	-1	-1	s36
37	0	0	-1	0	s37
38	0	0	-1	1	s38
39	0	0	0	-1	s39
40	0	0	0	0	s40
41	0	0	0	1	s41
42	0	0	1	-1	s42
43	0	0	1	0	s43
44	0	0	1	1	s44
45	0	1	-1	-1	s45
46	0	1	-1	0	s46
47	0	1	-1	1	s47
48	0	1	0	-1	s48
49	0	1	0	0	s49
50	0	1	0	1	s50
51	0	1	1	-1	s51
52	0	1	1	0	s52
53	0	1	1	1	s53

도면21c

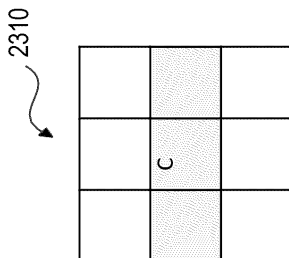
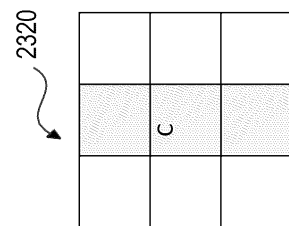
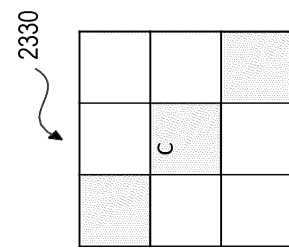
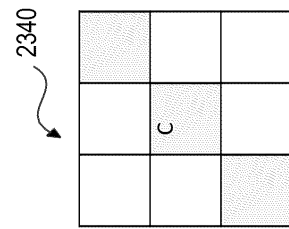
도 21b에서 계속

조합	d0	d1	d2	d3	오프셋
54	1	-1	-1	-1	s54
55	1	-1	-1	0	s55
56	1	-1	-1	1	s56
57	1	-1	0	-1	s57
58	1	-1	0	0	s58
59	1	-1	0	1	s59
60	1	-1	1	-1	s60
61	1	-1	1	0	s61
62	1	-1	1	1	s62
63	1	0	-1	-1	s63
64	1	0	-1	0	s64
65	1	0	-1	1	s65
66	1	0	0	-1	s66
67	1	0	0	0	s67
68	1	0	0	1	s68
69	1	0	1	-1	s69
70	1	0	1	0	s70
71	1	0	1	1	s71
72	1	1	-1	-1	s72
73	1	1	-1	0	s73
74	1	1	-1	1	s74
75	1	1	0	-1	s75
76	1	1	0	0	s76
77	1	1	0	1	s77
78	1	1	1	-1	s78
79	1	1	1	0	s79
80	1	1	1	1	s80

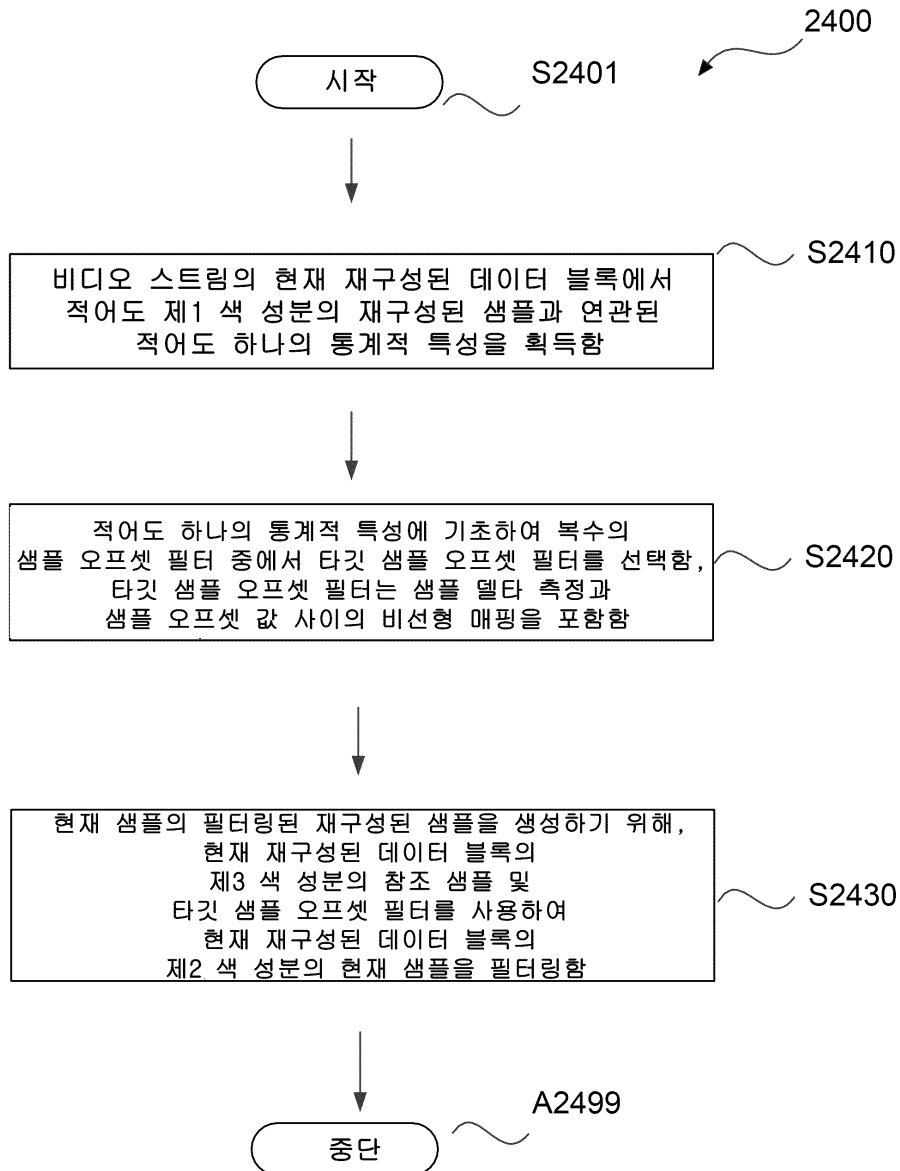
도면22



도면23



도면24



도면25

