



등록특허 10-2068431



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년01월20일
(11) 등록번호 10-2068431
(24) 등록일자 2020년01월14일

(51) 국제특허분류(Int. Cl.)
HO4N 19/117 (2014.01) *HO4N 19/132* (2014.01)
HO4N 19/172 (2014.01) *HO4N 19/186* (2014.01)

(52) CPC특허분류
HO4N 19/117 (2015.01)
HO4N 19/132 (2015.01)

(21) 출원번호 10-2018-7032998(분할)

(22) 출원일자(국제) 2012년11월07일
심사청구일자 2018년11월29일

(85) 번역문제출일자 2018년11월14일

(65) 공개번호 10-2018-0125051

(43) 공개일자 2018년11월21일

(62) 원출원 특허 10-2014-7019037
원출원일자(국제) 2012년11월07일
심사청구일자 2017년11월07일

(86) 국제출원번호 PCT/EP2012/072071

(87) 국제공개번호 WO 2013/068428
국제공개일자 2013년05월16일

(30) 우선권주장
1119206.9 2011년11월07일 영국(GB)
(뒷면에 계속)

(56) 선행기술조사문헌
KR1020110111852 A
EP02237557 A1
C-M Fu et al. "Sample Adaptive Offset for chroma", JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 6th Meeting: Torino, IT, 14-22 July, 2011, JCTVC-F057.

(73) 특허권자
캐논 가부시끼가이사
일본 도쿄도 오오따꾸 시모마루코 3조메 30방 2고

(72) 발명자
라로슈 기욤
프랑스 에프-35520 르라스 르 오 로내
지스께 크리스또프
프랑스 에프-35000 렌느 아브뉘 드 퀴셀레 18
(뒷면에 계속)

(74) 대리인
장수길, 이중희

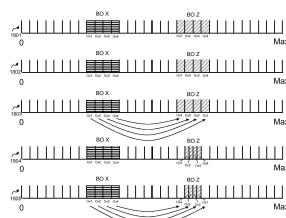
전체 청구항 수 : 총 12 항

심사관 : 김영태

(54) 발명의 명칭 영상의 재구성된 샘플 세트에 대한 보상 오프셋들의 인코딩/디코딩을 최적화하는 방법 및 장치

(57) 요약

본 발명은 복수의 샘플들을 포함하는 영상의 적어도 일부를 코딩 및 디코딩하는 방법 및 장치를 제공하고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함한다. 필터링은 적어도 하나의 필터링 파라미터를 구성된 샘플의 제1 성분 및 제2 성분 둘 다를 필터링하기 위한 공통 필터링
(뒷면에 계속)

대 표 도

파라미터로서 사용하는 것을 포함한다.

(52) CPC특허분류

HO4N 19/172 (2015.01)

HO4N 19/186 (2015.01)

(72) 발명자

프랑수와 에두아르

프랑스 에프-35890 부르그 데 꽁뜨 알레 뒤 로까르

18

온노 빠트리스

프랑스 에프-35890 렌느 알레 죠르쥬 빨령뜨 16

(30) 우선권주장

1203285.0 2012년02월24일 영국(GB)

1205304.7 2012년03월26일 영국(GB)

명세서

청구범위

청구항 1

인코딩된 영상의 복수의 샘플들을 디코딩하는 방법으로서, 각각의 샘플은 2개 이상의 성분을 포함하고,

상기 방법은,

상기 인코딩된 영상의 하나 이상의 샘플의 제1 크로마 성분과 제2 크로마 성분을 디코딩하여, 하나 이상의 재구성된 샘플의 제1 크로마 성분과 제2 크로마 성분을 제공하는, 디코딩 단계와,

하나 이상의 필터링 파라미터를 디코딩하는, 디코딩 단계와,

상기 하나 이상의 재구성된 샘플에 샘플 적응적 오프셋 필터링(sample adaptive offset filtering)을 실시하는, 실시 단계를 포함하고,

샘플 적응적 오프셋 필터링의 미사용, 대역 유형 및 에지 유형 중 하나인 샘플 적응적 오프셋 필터링의 유형, 및 에지 유형 필터링의 방향은, 상기 제1 크로마 성분 및 상기 제2 크로마 성분에 의해 공유되는 하나 이상의 디코딩된 필터링 파라미터에 의해 나타내지는,

인코딩된 영상의 복수의 샘플들을 디코딩하는 방법.

청구항 2

제1항에 있어서, 상기 제1 크로마 성분 및 상기 제2 크로마 성분은 각각 U 크로마 성분 및 V 크로마 성분인, 인코딩된 영상의 복수의 샘플들을 디코딩하는 방법.

청구항 3

제1항 또는 제2항의 방법을 실행하기 위한 프로그램을 저장하는 컴퓨터 판독 가능한 매체.

청구항 4

인코딩된 영상의 복수의 샘플들을 디코딩하는 영상 디코딩 장치로서, 각각의 샘플은 2개 이상의 성분을 포함하고,

상기 장치는,

상기 인코딩된 영상의 하나 이상의 샘플의 제1 크로마 성분과 제2 크로마 성분을 디코딩하여 하나 이상의 재구성된 샘플의 제1 크로마 성분과 제2 크로마 성분을 제공하고, 하나 이상의 필터링 파라미터를 디코딩하는, 디코딩 유닛과,

상기 하나 이상의 재구성된 샘플에 샘플 적응적 오프셋 필터링(sample adaptive offset filtering)을 실시하는, 실시 유닛을 포함하고,

샘플 적응적 오프셋 필터링의 미사용, 대역 유형 및 에지 유형 중 하나인 샘플 적응적 오프셋 필터링의 유형, 및 에지 유형 필터링의 방향은, 상기 제1 크로마 성분 및 상기 제2 크로마 성분에 의해 공유되는 하나 이상의 디코딩된 필터링 파라미터에 의해 나타내지는, 인코딩된 영상의 복수의 샘플들을 디코딩하는 영상 디코딩 장치.

청구항 5

제4항에 있어서, 상기 제1 크로마 성분 및 상기 제2 크로마 성분은 각각 U 크로마 성분 및 V 크로마 성분인, 인코딩된 영상의 복수의 샘플들을 디코딩하는 영상 디코딩 장치.

청구항 6

영상의 복수의 샘플들을 인코딩하는 방법으로서, 각각의 샘플은 2개 이상의 성분을 포함하고,

상기 방법은,

상기 영상의 하나 이상의 샘플의 제1 크로마 성분과 제2 크로마 성분을 처리하여, 하나 이상의 재구성된 샘플의 제1 크로마 성분과 제2 크로마 성분을 제공하는, 처리 단계와,

하나 이상의 필터링 파라미터를 인코딩하는, 인코딩 단계와,

상기 하나 이상의 재구성된 샘플에 샘플 적응적 오프셋 필터링(sample adaptive offset filtering)을 실시하는, 실시 단계를 포함하고,

샘플 적응적 오프셋 필터링의 미사용, 대역 유형 및 에지 유형 중 하나인 샘플 적응적 오프셋 필터링의 유형, 및 에지 유형 필터링의 방향은, 상기 제1 크로마 성분 및 상기 제2 크로마 성분에 의해 공유되는 하나 이상의 인코딩된 필터링 파라미터에 의해 나타내지는, 영상의 복수의 샘플들을 인코딩하는 방법.

청구항 7

제6항에 있어서, 상기 제1 크로마 성분 및 상기 제2 크로마 성분은 각각 U 크로마 성분 및 V 크로마 성분인, 영상의 복수의 샘플들을 인코딩하는 방법.

청구항 8

제6항에 있어서, 상기 처리 단계는,

상기 영상의 상기 하나 이상의 샘플의 상기 제1 크로마 성분과 상기 제2 크로마 성분을 인코딩하여 하나 이상의 인코딩된 샘플의 제1 크로마 성분과 제2 크로마 성분을 제공하는, 인코딩 단계를 포함하고,

상기 하나 이상의 인코딩된 샘플의 상기 제1 크로마 성분과 상기 제2 크로마 성분을 적어도 부분적으로 디코딩하여 상기 하나 이상의 재구성된 샘플의 상기 제1 크로마 성분과 상기 제2 크로마 성분을 제공하는, 디코딩 단계를 더 포함하는, 영상의 복수의 샘플들을 인코딩하는 방법.

청구항 9

제6항, 제7항, 또는 제8항의 방법을 실행하기 위한 프로그램을 저장하는 컴퓨터 판독 가능한 매체.

청구항 10

영상의 복수의 샘플들을 인코딩하는 영상 인코딩 장치로서, 각각의 샘플은 2개 이상의 성분을 포함하고,

상기 장치는,

상기 영상의 하나 이상의 샘플의 제1 크로마 성분과 제2 크로마 성분을 처리하여, 하나 이상의 재구성된 샘플의 제1 크로마 성분과 제2 크로마 성분을 제공하는, 처리 유닛과,

하나 이상의 필터링 파라미터를 인코딩하는, 인코딩 유닛을 포함하고,

샘플 적응적 오프셋 필터링(sample adaptive offset filtering)의 미사용, 대역 유형 및 에지 유형 중 하나인 샘플 적응적 오프셋 필터링의 유형, 및 에지 유형 필터링의 방향은, 상기 제1 크로마 성분 및 상기 제2 크로마 성분에 의해 공유되는 하나 이상의 인코딩된 필터링 파라미터에 의해 나타내지는, 영상의 복수의 샘플들을 인코딩하는 영상 인코딩 장치.

청구항 11

제10항에 있어서, 상기 제1 크로마 성분 및 상기 제2 크로마 성분은 각각 U 크로마 성분 및 V 크로마 성분인, 영상의 복수의 샘플들을 인코딩하는 영상 인코딩 장치.

청구항 12

제10항 또는 제11항에 있어서, 상기 처리 유닛은,

상기 영상의 상기 하나 이상의 샘플의 상기 제1 크로마 성분과 상기 제2 크로마 성분을 인코딩하여 하나 이상의 인코딩된 샘플의 제1 크로마 성분과 제2 크로마 성분을 제공하고, 상기 하나 이상의 인코딩된 샘플의 상기 제1 크로마 성분과 상기 제2 크로마 성분을 적어도 부분적으로 디코딩하여 상기 하나 이상의 재구성된 샘플의 상기 제1 크로마 성분과 상기 제2 크로마 성분을 제공하는, 영상의 복수의 샘플들을 인코딩하는 영상 인코딩 장치.

발명의 설명

기술 분야

[0001] 본 발명은 비디오 코덱의 코딩 효율 및/또는 디코딩 복잡도를 향상시키기 위해 영상의 재구성된 샘플 세트에 대한 보상 오프셋들의 인코딩 및/또는 디코딩을 최적화하는 방법 및 장치에 관한 것이다. 본 발명은 또한 디지털 영상 시퀀스를 인코딩 또는 디코딩하는 방법 및 장치에 관한 것이다.

배경 기술

[0002] 본 발명은 디지털 신호 처리의 분야에서 그리고 특히 비디오 스트림에서 공간 및 시간 중복성을 감소시키기 위해 움직임 보상을 사용하는 비디오 압축의 분야에서 적용될 수 있다.

[0003] 예를 들어, H.263, H.264, MPEG-1, MPEG-2, MPEG-4, SVC 등의 많은 비디오 압축 형식들은 공간 및 시간 중복성을 제거하기 위해 블록 기반 DCT(discrete cosine transform) 및 움직임 보상을 사용한다. 이들은 종종 예측 비디오 형식(predictive video format)이라고 한다. 비디오 신호의 각각의 프레임 또는 영상은 독립적으로 인코딩되고 디코딩될 수 있는 슬라이스들로 나누어진다. 슬라이스는 통상적으로 프레임의 직사각형 부분, 또는 보다 일반적으로, 프레임 또는 전체 프레임의 일부분이다. 게다가, 각각의 슬라이스는 매크로블록들(macroblock, MB)로 나누어질 수 있고, 각각의 매크로블록은 블록들(통상적으로, 64x64, 32x32, 16x16 또는 8x8 픽셀의 블록들)로 추가로 나누어진다.

[0004] HEVC(High Efficiency Video Coding)에서, 64x64 내지 4x4의 블록들이 사용될 수 있다. 이 분할은 LCU(largest coding unit,, 최대 코딩 단위)에 기초한 쿼드-트리 구조(quad-tree structure)에 따라 구성되어 있다. LCU는, 예를 들어, 64x64의 정사각형 블록에 대응한다. LCU가 나누어질 필요가 있는 경우, 분할 플래그(split flag)는 LCU가 4개의 32x32 블록들로 분할되어 있다는 것을 나타낸다. 동일한 방식으로, 이 4개의 블록들 중 임의의 것이 분할될 필요가 있는 경우, 분할 플래그는 참(true)으로 설정되고, 32x32 블록이 4개의 16x16 블록들로 나누어지며, 이하 마찬가지이다. 분할 플래그가 거짓(false)으로 설정되어 있을 때, 현재의 블록이 CU(coding unit, 코딩 단위)이다. CU는 64x64, 32x32, 16x16 또는 8x8 픽셀의 크기를 가진다.

[0005] 영상의 블록들을 코딩하는 2가지 계열의 코딩 모드 - 공간 예측에 기초한 코딩 모드[인트라 예측(INTRA prediction)이라고 함], 및 시간 예측에 기초한 코딩 모드[인터(INTER), 병합(Merge), 건너뛰(Skip)] - 가 있다. 공간 예측 모드 및 시간 예측 모드 둘 다에서, 잔차는 원래의 블록으로부터 예측을 차감함으로써 계산된다.

[0006] 인트라 블록(INTRA block)은 일반적으로 인트라 예측 프로세스에 의해 그의 인과 경계(causal boundary)에서의 인코딩된 픽셀로부터 예측된다. 인트라 예측에서, 예측 방향이 인코딩된다.

[0007] 시간 예측은, 비디오 시퀀스의 이전 또는 장래 프레임(previous or a future frame)인 참조 프레임(reference frame)에서, 인코딩될 블록에 가장 가까운 영상 부분(image portion) 또는 참조 영역(reference area)을 찾아내는 데 있다. 이 단계는 통상적으로 움직임 추정(motion estimation)이라고 한다. 그 다음에, 통상적으로 움직임 보상(motion compensation)이라고 하는 단계에서 참조 영역을 사용하여 인코딩될 블록이 예측된다 - 인코딩될 블록과 참조 부분 사이의 차가, 움직임 보상을 위해 사용할 참조 영역을 나타내는 움직임 벡터에 대한 움직임 정보의 항목과 함께, 인코딩된다 - 시간 예측에서, 적어도 하나의 움직임 벡터가 인코딩된다.

[0008] 움직임 정보를 인코딩하는 비용을 추가로 감소시키기 위해, 움직임 벡터를 직접 인코딩하기보다는, 움직임이 균질(homogeneous)하다고 가정하여, 통상적으로 인코딩될 블록을 둘러싸고 있는 블록들의 하나 이상의 움직임 벡터들로부터 계산되는 움직임 벡터와 움직임 벡터 예측자 간의 차로 움직임 벡터가 인코딩될 수 있다.

[0009] H.264에서, 예를 들어, 움직임 벡터들이 인코딩될 블록의 인과 이웃(causal neighbourhood)에 위치해 있는 움직임 벡터들로부터, 예를 들어, 인코딩될 블록의 위쪽에 그리고 좌측에 위치해 있는 3개의 블록들로부터, 계산되는 중간 예측자(median predictor)에 대해 인코딩된다. 인코딩 비용을 감소시키기 위해, 중간 예측자와 현재 블록 움직임 벡터 간의 차[잔차 움직임 벡터(residual motion vector)라고 함]만이 비트스트림에 인코딩된다.

[0010] 잔차 움직임 벡터를 사용한 인코딩은 얼마간의 비트레이트를 절감하지만, 디코딩될 블록의 움직임 벡터의 값을 디코딩하기 위해 디코더가 움직임 벡터 예측자의 동일한 계산을 수행하는 것을 필요로 한다.

[0011] 인코딩 프로세스 및 디코딩 프로세스 둘 다는 인코딩된 영상의 디코딩 프로세스를 포함할 수 있다. 이 프로세

스는 통상적으로 장래 움직임 추정을 위해 인코더측에서 수행되고, 이는 인코더 및 대응하는 디코더가 동일한 참조 프레임을 가질 수 있게 해준다.

[0012] 코딩된 프레임을 재구성하기 위해, 픽셀 영역에서 "디코딩된" 잔차를 제공하기 위해 잔차가 역양자화 및 역변환된다. 제1 재구성은 이어서 하나의 또는 몇개의 종류의 사후 필터링(post filtering) 프로세스에 의해 필터링된다. 동일한 참조 프레임이 인코더측 및 디코더측 둘 다에서 사용되도록, 이를 사후 필터(post filter)가 인코더측 및 디코더측에서 재구성된 프레임에 적용된다. 이 사후 필터링의 목표는 압축 아티팩트를 제거하고 영상 품질을 향상시키는 것이다. 예를 들어, H.264/AVC는 디블록킹 필터(deblocking filter)를 사용한다. 이 필터는 잔차의 DCT 양자화 및 블록 움직임 보상으로 인한 블록킹 아티팩트(blocking artifact)를 제거할 수 있다. 현재의 HEVC 표준에서, 3가지 유형의 루프 필터 - 디블록킹 필터, SAO(sample adaptive offset, 샘플 적응적 오프셋), 및 ALF(adaptive loop filter, 적응적 루프 필터) - 가 사용되고 있다.

[0013] 도 1은 공지된 HEVC 구현예의 루프 필터링 프로세스의 단계들을 나타낸 플로우차트이다. 초기 단계(101)에서, 인코더 또는 디코더는 전체 프레임의 재구성을 발생한다. 그 다음에, 단계(102)에서, 디블록킹된 재구성(103)을 발생시키기 위해 이 제1 재구성에 디블록킹 필터가 적용된다. 디블록킹 필터의 목표는 잔차 양자화 및 블록 움직임 보상 또는 블록 인트라 예측에 의해 발생된 블록 아티팩트를 제거하는 것이다. 이를 아티팩트는 낮은 비트레이트에서 시각적으로 중요하다. 디블록킹 필터는 2개의 이웃하는 블록들의 특성에 따라 블록 경계를 매끄럽게 하는 동작을 한다. 각각의 블록의 인코딩 모드, 잔차 코딩을 위해 사용되는 양자화 파라미터들, 및 경계에서의 이웃 픽셀 차가 고려된다. 모든 프레임들에 대해 동일한 기준/분류가 적용되고, 어떤 부가의 데이터도 전송되지 않는다. 디블록킹 필터는, 블록킹 아티팩트를 제거함으로써, 현재 프레임의 시각적 품질을 향상시키고, 또한 후속 프레임들에 대한 움직임 추정 및 움직임 보상을 향상시킨다. 실제로, 블록 아티팩트의 고주파 수가 제거되고, 따라서 후속 프레임들의 텍스처 잔차(texture residual)에 대해 이를 고주파수가 보상될 필요가 없다.

[0014] 디블록킹 필터 후에, 디블록킹된 재구성이 단계(104)에서 SAO(sample adaptive offset) 루프 필터에 의해 필터링된다. 얻어진 프레임(105)은 이어서 디스플레이되고 후속 인터 프레임들에 대한 참조 프레임으로서 사용될 재구성된 프레임(107)을 발생시키기 위해 단계(106)에서 ALF(adaptive loop filter)에 의해 필터링된다.

[0015] SAO 루프 필터 및 ALF의 목표는 어떤 정보도 전송되지 않는 디블록킹 필터와 달리 부가의 데이터를 송신함으로써 프레임 재구성을 향상시키는 것이다.

[0016] SAO 루프 필터의 원리는 각각의 픽셀을 클래스로 분류하고 클래스의 각각의 픽셀의 각자의 픽셀 값에 동일한 오프셋 값을 부가하는 것이다. 이와 같이, 각각의 클래스에 대해 하나의 오프셋이 전송된다. SAO 루프 필터링은 프레임 영역에 대한 2가지 종류의 분류 - 에지 오프셋(edge offset) 및 대역 오프셋(band offset) - 를 제공한다. 에지 오프셋 분류는 각각의 픽셀의 대응하는 픽셀 값을 2개의 이웃 픽셀의 픽셀 값과 비교함으로써 각각의 픽셀에 대한 클래스를 결정하는 것을 수반한다. 더욱이, 2개의 이웃 픽셀은 2개의 이웃 픽셀의 방향을 나타내는 파라미터에 의존한다. 이를 방향은 0도(수평 방향), 45도(대각 방향), 90도(수직 방향), 및 135도(제2 대각 방향)이다. 사용될 방향은 현재 HEVC 규격에서 sao_type_idx라고 하는 SAO 파라미터에 의해 주어진다. 그의 값은, 부록의 표 1에 나타낸 바와 같이, 통상적으로 0부터 5까지 변한다. sao_type_idx는 또한 SAO 필터링이 수행되어야 하는지 여부 및 SAO 필터링의 유형을 나타내는 데 사용된다.

[0017] 예시를 위해, 픽셀 값(또는 샘플) C에 가산될 오프셋은, 주어진 방향에 대해, 부록의 표 2에 나타낸 바와 같은 규칙들에 따라 결정될 수 있고, 여기서 C_{n1} 및 C_{n2} 는 (주어진 방향에 따라) 2개의 이웃하는 픽셀 또는 샘플의 값을 나타낸다. 그에 따라, 값 C가 2개의 값 C_{n1} 및 C_{n2} 미만일 때, C에 가산될 오프셋은 $+0_1$ 이고, 값 C가 C_{n1} 또는 C_{n2} 미만이고 다른 값(C_{n1} 또는 C_{n2})과 같을 때, 사용될 오프셋은 $+0_2$ 이며, 값 C가 C_{n1} 또는 C_{n2} 초과이고 다른 값(C_{n1} 또는 C_{n2})과 같을 때, 사용될 오프셋은 -0_3 이고, 값 C가 C_{n1} 및 C_{n2} 초과일 때, 사용될 오프셋은 -0_4 이다. 이를 조건 중 어느 것도 충족되지 않을 때, 어떤 오프셋 값도 현재 픽셀 값 C에 가산되지 않는다.

[0018] 유의할 점은, 에지 오프셋 모드에 따르면, 각각의 오프셋의 절대값만이 비트스트림에 인코딩되고, 적용될 부호가 현재 픽셀이 속하는 카테고리의 함수로서 결정된다는 것이다. 따라서, 부록의 표 2에 따르면, 플러스 오프셋은 카테고리 1 및 2와 연관되어 있는 반면, 마이너스 오프셋은 카테고리 3 및 4와 연관되어 있다.

[0019] 제2 유형의 분류는 픽셀 값에 의존하는 대역 오프셋 분류이다. SAO 대역 오프셋에서의 클래스는 픽셀 값 범위에 대응한다. 이와 같이, 동일한 오프셋이 주어진 픽셀 값 범위 내의 픽셀 값을 가지는 모든 픽셀들에 가산된

다.

[0020] 프레임 콘텐츠에 보다 적응적이기 위해, 현재 프레임을 몇개의 공간 영역들로 나누는 몇개의 프레임 영역들에 SAO 필터링이 적용된다. 현재, HEVC에서 프레임 영역들은 유한한 수의 최대 코딩 단위에 대응한다. 그 결과, 하나의 프레임 영역이 SAO에 의해 필터링될 수 있거나 그렇지 않을 수 있고, 따라서 어떤 프레임 영역들만이 필터링된다. 더욱이, SAO가 인에이블되어 있을 때, 단지 하나의 SAO 분류가 사용된다: 각각의 분류에 대해 전송되는 관련 파라미터들에 따라 에지 오프셋 또는 대역 오프셋. 마지막으로, 각각의 SAO 리프 노드에 대해, SAO 분류는 물론 그의 파라미터 및 모든 클래스의 오프셋이 전송된다.

[0021] 인코딩될 비디오 데이터의 영상이 샘플 값들의 2차원 어레이(색상 채널이라고도 함) 세트로서 제공될 수 있고, 그의 각각의 항목은 루미넌스 세기(luminance intensity) 및 중성 그레이스케일 색상(neutral grayscale colour)으로부터 청색 또는 적색 쪽으로의 크로미넌스 세기(chrominance intensity)의 척도(YUV) 또는 적색, 녹색 또는 청색 광 성분 세기의 척도(RGB) 등의 색상 성분의 세기(intensity)를 나타낸다. YUV 모델은 하나의 루마(Y) 성분 및 2개의 크로미넌스(chrominance)(UV) 성분으로 색 공간을 정의한다. 일반적으로, Y는 루미넌스 성분을 나타내고, U 및 V는 크로미넌스(색상) 또는 크로마 성분이다.

[0022] SAO 필터링이 통상적으로 루마 성분에 그리고 U 및 V 크로마 성분 둘 다에 독립적으로 적용된다.

[0023] SAO 대역 오프셋의 공지된 구현은 픽셀 값 범위를 도 2에 도시된 것과 동일한 크기의 사전 정의된 32개의 범위로 분할한다. 픽셀 값 범위의 최소값은 0이고, 최대값은 다음과 같은 관계식 $Max = 2^{\text{Bitdepth}} - 1$ 에 따라 픽셀 값들의 비트 심도(bit-depth)에 의존한다.

[0024] 전체 픽셀 값 범위를 32개의 범위로 분할하는 것은 각각의 픽셀을 분류하는 데 5 비트를 사용할 수 있게 하고, 고속 분류를 가능하게 한다. 그에 따라, 픽셀을 전체 범위의 32개의 클래스들 또는 범위들 중 하나로 분류하기 위해 단지 5 비트만이 검사된다. 이것은 일반적으로 8 비트로 인코딩된 값들의 최상위 5 비트를 검사하는 것에 의해 행해진다.

[0025] 예를 들어, 비트 심도가 8 비트일 때, 픽셀의 가능한 최대값은 255이다. 이와 같이, 픽셀 값 범위는 0과 255 사이이다. 이 8 비트의 비트 심도에 대해, 각각의 클래스는 8개 픽셀 값으로 된 범위를 포함한다.

[0026] SAO 대역 필터링의 목표는 제1 클래스 또는 대역에 의해 결정되는 4개의 연속적인 클래스들 또는 범위들로 된 그룹에 속하는 픽셀들을 필터링하는 것이다. 후자가 비트스트림에서 전송되고, 따라서 디코더가 필터링될 픽셀들의 4개의 연속적인 클래스들 또는 범위들을 결정할 수 있다. 이 위치를 나타내는 파라미터는 현재 HEVC 규격에서 sao_band_position로 지칭된다.

[0027] 예시를 위해, 필터링될 픽셀들의 4개의 연속적인 클래스들 또는 범위들(201 내지 204)로 된 그룹이 도 2에서 회색 영역으로 나타내어져 있다. 앞서 기술한 바와 같이, 이 그룹은 그의 위치(예컨대, sao_band_position)에 의해 식별될 수 있다. 주어진 예에 따르면, 클래스 또는 범위(201)는 64와 71 사이에 포함되는 값들을 가지는 픽셀들에 관한 것이다. 이와 유사하게, 클래스들 또는 범위들(202 내지 204)은, 각각, 72와 79 사이, 80과 87 사이, 그리고 88과 96 사이의 값들을 가지는 픽셀들에 관한 것이다.

[0028] 도 3은 인코더에서 현재 프레임 영역(303)(통상적으로 처리된 영상의 한 성분에 대응하는 LCU 블록)에 대한 오프셋들을 선택하는 방법의 단계들을 나타낸 플로우차트이다. 프레임 영역은 N개의 픽셀을 포함하고 있다. 초기 단계(301)에서, 변수 Sum_j 및 $SumNbPix_j$ 는 4개의 범위 각각에 대해 0의 값으로 설정되고, j는 현재 범위 또는 클래스 번호를 나타낸다. Sum_j 는 범위 j에서의 픽셀들의 값과 그의 대응하는 원래의 픽셀들의 값 사이의 차의 합이다. $SumNbPix_j$ 는 범위 j 내의 픽셀들의 수를 나타낸다.

[0029] 단계(302)에서, 카운터 변수 i는 값 0으로 설정된다. 그 다음에, 단계(304)에서 프레임 영역(303)의 제1 픽셀 P_i 가 추출되고, 단계(305)에서 현재 픽셀 P_i 에 대응하는 클래스 번호 J가 획득된다. 그 다음에, 단계(306)에서 부록의 표 2를 참조하여 앞서 기술한 바와 같이 현재 픽셀 P_i 의 클래스 번호 J가 값 "해당 없음"에 대응하는지를 판정하기 위해 테스트가 수행된다. 현재 픽셀 P_i 의 클래스 번호 J가 값 "해당 없음"에 대응하는 경우, 프레임 영역(303)의 후속 픽셀들을 분류하기 위해 카운터 변수 i의 값이 1만큼 증가된다. 그렇지 않고, 현재 픽셀 P_i 의 클래스 번호 J가 값 "해당 없음"에 대응하지 않는 경우, 현재 픽셀 P_i 에 대응하는 $SumNbPix_j$ 변수가 1만큼

증가되고, 단계(307)에서 P_i 와 그의 원래의 값 P_i^{org} 사이의 차가 Sum_j 에 가산된다.

[0030] 그 다음 단계에서, 프레임 영역(303)의 다른 픽셀들에 분류를 적용하기 위해 카운터 변수 i 가 1만큼 증가된다. 단계(309)에서, 프레임 영역(303)의 N 개의 픽셀들 모두가 분류되었는지(즉, $i \geq N$ 인지) 여부가 판정되고, 예인 경우, 오프셋 선택 알고리즘의 최종 결과로서 각각의 클래스 j 에 대한 오프셋을 제공하는 오프셋 테이블(311)을 생성하기 위해 단계(310)에서 각각의 클래스에 대한 $Offset_j$ 가 계산된다. 이 오프셋은 클래스 j 의 픽셀의 픽셀 값과 그 각자의 원래의 픽셀 값 사이의 차의 평균으로서 계산된다. 클래스 j 에 대한 $Offset_j$ 는 이하의 식에 의해 주어진다:

$$Offset_j = \frac{Sum_j}{SumNbPix_j}$$

[0031] [0032] 계산된 오프셋 $Offset_j$ 는 왜곡의 면에서 최적의 오프셋인 것으로 간주될 수 있다. 최적의 오프셋은 이하에서 $Oopt_j$ 라고 한다. 이 오프셋으로부터, 레이트 왜곡 기준에 따라 개선된 오프셋 값 O_{RDj} 를 결정하는 것이 가능하다.

[0033] [0034] 도 4는 레이트 왜곡 기준에 따라 개선된 오프셋을 결정하는 방법의 단계들을 나타낸 플로우차트이다. 초기 단계(401)에서, 현재 범위 또는 클래스 번호 j 의 레이트 왜곡 값 J_j 가 사전 결정된 최대 가능 값(maximum possible value)(MAX_VALUE)으로 초기화된다.

[0035] 그 다음에, 단계(402)에서 오프셋 O_j 를 $Oopt_j$ 로부터 0으로 변화시키기 위해 루프가 시작된다. 값 $Oopt_j$ 가 마이너스인 경우, 변수 O_j 는 0에 도달할 때까지 1만큼 증가되고, 값 $Oopt_j$ 가 플러스인 경우, 변수 O_j 는 0에 도달할 때까지 1만큼 감소된다.

[0036] [0037] 단계(403)에서, 변수 O_j 에 관련된 레이트 왜곡 비용[$J(O_j)$ 로 표시됨]이, 예를 들어, 이하의 식에 따라 계산된다:

$$J(O_j) = SumNbPix_j \times O_j \times O_j - Sum_j \times O_j \times 2 + \lambda R(O_j)$$

[0038] λ 는 Lagrange 파라미터이고, $R(O_j)$ 는 O_j 와 연관되어 있는 코드워드에 대해 필요한 비트 수를 제공하는 함수이다. $SumNbPix_j \times O_j \times O_j - Sum_j \times O_j \times 2$ 에 대응하는 식의 일부는 오프셋 O_j 에 의해 주어지는 왜곡의 면에서의 향상에 관한 것이다.

[0039] 단계(404)에서, 값들 $J(O_j)$ 및 J_j 가 서로 비교된다. 값 $J(O_j)$ 이 값 J_j 미만인 경우, J_j 는 $J(O_j)$ 의 값으로 설정되고 O_{RDj} 는 O_j 의 값으로 설정된다.

[0040] 단계(405)에서, 오프셋 O_j 의 가능한 값들 모두가 처리되었는지(즉, $O_j = 0$ 인지) 여부가 판정된다. 오프셋 O_j 가 0인 경우, 루프가 종료되고, 클래스 j 에 대한 개선된 오프셋 값(O_{RDj})이 식별된다. 그렇지 않은 경우, 루프는 그 다음 O_j 값으로 계속된다.

[0041] 유의할 점은, 도 3을 참조하여 기술된 알고리즘이 대역 오프셋 분류 유형에 따라 제1 클래스 또는 범위의 위치(sao_band_position)를 결정하는 데 사용될 수 있다는 것이다. 그를 위해, 인덱스 j 는 ([1, 4] 대신에) 구간 [0, 32]의 값을 나타낸다. 환언하면, 값 4는 도 3의 모듈들(301, 310, 및 311)에서 값 32로 대체된다.

[0042] 보다 구체적으로는, t 도 2에 나타낸 32개의 클래스 각각에 대해, 즉 각각의 범위 j (j 는 구간 [0, 32]에 속함)에 대해, 현재 픽셀의 값과 그의 원래의 값 $Porg_j$ 사이의 차 Sum_j 가 계산될 수 있다.

[0043] 그 다음에, 도 4를 참조하여 기술된 것과 유사한 알고리즘에 따라 32개의 클래스에 대해 레이트 왜곡의 면에서 개선된 오프셋 O_{RDj} 가 계산된다.

[0044] 그 다음에, 제1 클래스 또는 범위의 위치가 결정될 수 있다.

[0045] 도 5는 HEVC의 SAO 대역 오프셋에 대한 SAO 대역 위치를 결정하는 방법의 단계들을 나타낸 플로우차트이다. 이들 단계가 도 4를 참조하여 기술된 단계들 이후에 수행되기 때문에, J_j 로 표시된 레이트 왜곡 값이 각각의 클래스 j 에 대해 이미 계산되었다.

[0045] 초기 단계(501)에서, 레이트 왜곡 값 J 가 사전 결정된 최대 가능 값(MAX_VALUE)으로 초기화된다. 그 다음에, 인덱스 i 를 4개의 연속적인 클래스들로 된 그룹의 제1 클래스의 28개의 가능한 위치들에 대응하는 0부터 27까지 변화시키기 위해 단계(502)에서 루프가 시작된다.

[0046] 단계(503)에서, 현재 대역, 즉 인덱스 i 를 가지는 클래스로부터 4개의 연속적인 클래스를 포함하는 대역의 레이트 왜곡 값에 대응하는 변수 $J'j$ 가 0으로 초기화된다. 그 다음에, 인덱스 j 를 현재 대역의 4개의 클래스에 대응하는 i 부터 $i+3$ 까지 변화시키기 위해 단계(504)에서 루프가 시작된다.

[0047] 그 다음에, 단계(505)에서, 변수 J 의 값이 인덱스 j 를 가지는 클래스의 레이트 왜곡 값(J_j)의 값만큼 증가된다. 현재 대역의 4개의 클래스에 대해, 즉 인덱스 j 가 $i+3$ 에 도달할 때까지, 이 단계가 반복된다[단계(506)]. 단계(507)에서, 현재 대역의 레이트 왜곡 값 $J'j$ 가 레이트 왜곡 값 J 미만인지 여부를 판정하기 위해 테스트가 수행된다. 현재 대역의 레이트 왜곡 값 $J'j$ 가 레이트 왜곡 값 J 미만인 경우, 레이트 왜곡 값 J 가 현재 대역의 레이트 왜곡 값 $J'j$ 로 설정되고, sao_band_position로 표시되는 대역 위치 값이 인덱스 i 의 값으로 설정된다.

[0048] 사용될 대역 위치를 결정하기 위해 4개의 연속적인 클래스로 된 그룹의 제1 클래스의 28개의 가능한 위치들에 대해 이들 단계가 반복된다[단계(508)].

[0049] 도 6은 SAO 루프 필터에 따라, 프레임 영역(통상적으로 처리된 영상의 하나의 성분에 대응하는 LCU 블록)을 필터링하는 방법의 단계들을 나타낸 플로우차트이다.

[0050] 이러한 알고리즘은 일반적으로 디코더에서 프레임들을 디코딩하기 위해 그리고 인코더에서 후속 프레임들의 움직임 추정 및 보상을 위해 사용되는 참조 프레임들을 발생하기 위해 구현된다.

[0051] 초기 단계(601)에서, SAO 필터링 파라미터들이, 예를 들어, 수신된 비트스트림으로부터 획득된다. 주어진 프레임 영역에 대해, 이들 파라미터는 통상적으로 테이블(603)에 저장될 수 있는 4개의 오프셋 및 sao_type_idx의 파라미터를 포함한다. 후자에 따라, 이들 파라미터는 SAO 대역 위치를 추가로 포함할 수 있다. 예시를 위해, 획득된 sao_type_idx 유형 파라미터가 5인 경우, 대응하는 sao_band_position 유형 파라미터가 이어서 획득된다(602). 주어진 프레임 영역의 SAO 파라미터의 값을 이전 프레임 영역의 대응하는 파라미터의 값의 함수로서 결정할 수 있게 하는 파라미터들과 같은 기타 파라미터들이 획득될 수 있다. 유의할 점은, sao_type_idx 파라미터에 대한 값 0 등의 주어진 SAO 파라미터의 주어진 값이 SAO 필터링이 적용되어서는 안된다는 것을 나타낼 수 있다는 것이다.

[0052] 단계(604)에서, 카운터 변수 i 는 값 0으로 설정된다. 그 다음에, N 개의 픽셀을 포함하는 프레임 영역(605)의 제1 픽셀 P_i 가 단계(606)에서 추출되고, 단계(607)에서 예지 오프셋 분류 또는 대역 오프셋 분류에 따라 분류되고, 이에 대해서는 앞서 기술하였다.

[0053] 단계(608)에서, 픽셀 P_i 가 필터링될 픽셀 클래스에 속하는지 여부를 판정하기 위해 테스트가 수행된다. 픽셀 P_i 가 필터링될 픽셀 클래스에 속하는 경우, 그의 관련 클래스 번호 j 가 식별되고, 단계(610)에서 그의 관련 오프셋 값 $Offset_j$ 가 오프셋 테이블(603)로부터 추출된다. 그 다음에, 단계(611)에서, P'_i 라고 하는 새로운 픽셀 값을 생성하기 위해 $Offset_j$ 가 픽셀 P_i 의 값에 가산된다(612). 단계(613)에서, 픽셀 P'_i 가 처리된 프레임 영역(616) 내의 픽셀 P_i 를 대체한다. 그렇지 않고, 픽셀 P_i 가 필터링될 픽셀 클래스에 속하지 않는 경우, 단계(613)에서 픽셀 P_i (609)가 프레임 영역에 그대로 있다.

[0054] 그 다음에, 픽셀 P_i 를 처리한 후에, 현재 프레임 영역(605)의 그 다음 픽셀에 대해 유사하게 필터를 적용하기 위해 단계(614)에서 카운터 변수 i 가 1만큼 증가된다.

[0055] 단계(615)에서, 현재 프레임 영역(605)의 N 개의 픽셀들 모두가 처리되었는지(즉, $i \geq N$ 인지) 여부가 판정된다. 예인 경우, 도 1을 참조하여 앞서 기술한 바와 같이[참조 번호(105)], 처리된 프레임 영역(616)이 재구성되고 SAO 재구성된 프레임에 가산될 수 있다.

[0056] 보상의 선택 및 전송을 위한 공지된 프로세스의 단점은 많은 수의 파라미터들이 인코더로부터 디코더로 전송되어 SAO 버퍼에 저장되어야만 한다는 것이다. 더욱이, 이들 파라미터를 디코딩하고 처리하기 위해 디코더에서 수행되는 프로세스가 복잡하다.

발명의 내용

[0057] 이상의 문제점들 중 하나 이상을 해결하기 위해 본 발명이 안출되었다.

[0058] 본 발명의 제1 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 인코딩하는 방법이 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 방법은,

[0059] 적어도 하나의 인코딩된 샘플을 제공하기 위해 영상의 적어도 하나의 샘플의 제1 및 제2 성분을 인코딩하는 단계;

[0060] 적어도 하나의 인코딩된 샘플을 디코딩하는 단계;

[0061] 적어도 하나의 필터링 파라미터를 영상의 적어도 하나의 샘플의 적어도 하나의 성분과 적어도 하나의 대응하는 디코딩된 샘플의 적어도 하나의 대응하는 성분 간의 차의 함수로서 계산하는 단계 - 적어도 하나의 필터링 파라미터는 루프 필터에서 디코딩된 샘플의 제1 및 제2 성분 둘 다를 필터링하기 위한 공통 필터링 파라미터로서 사용되기 위해 계산됨 - 를 포함한다.

[0062] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0063] 본 발명의 제2 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 디코딩하는 방법이 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 방법은,

[0064] 적어도 하나의 재구성된 샘플의 제1 및 제2 성분을 제공하기 위해 적어도 하나의 인코딩된 샘플의 제1 및 제2 성분을 디코딩하는 단계;

[0065] 적어도 하나의 필터링 파라미터를 디코딩하는 단계;

[0066] 루프 필터에서 적어도 하나의 재구성된 샘플을 필터링하는 단계 - 필터링은 적어도 하나의 디코딩된 필터링 파라미터를 적어도 하나의 재구성된 샘플의 제1 성분 및 제2 성분 둘 다를 필터링하기 위한 공통 필터링 파라미터로서 사용하는 것을 포함함 - 를 포함한다.

[0067] 본 발명의 제3 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 인코딩하는 인코딩 장치가 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 인코딩 장치는,

[0068] 적어도 하나의 인코딩된 샘플을 제공하기 위해 영상의 적어도 하나의 샘플의 제1 및 제2 성분을 인코딩하는 수단;

[0069] 적어도 하나의 인코딩된 샘플을 디코딩하는 수단;

[0070] 적어도 하나의 필터링 파라미터를 영상의 적어도 하나의 샘플의 적어도 하나의 성분과 적어도 하나의 대응하는 디코딩된 샘플의 적어도 하나의 대응하는 성분 간의 차의 함수로서 계산하는 수단 - 적어도 하나의 필터링 파라미터는 루프 필터에서 디코딩된 샘플의 제1 및 제2 성분 둘 다를 필터링하기 위한 공통 필터링 파라미터로서 사용되기 위해 계산됨 - 을 포함한다.

[0071] 본 발명의 제4 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 디코딩하는 디코딩 장치가 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 디코딩 장치는,

[0072] 적어도 하나의 재구성된 샘플의 제1 및 제2 성분을 제공하기 위해 적어도 하나의 인코딩된 샘플의 제1 및 제2 성분을 디코딩하는 수단;

[0073] 적어도 하나의 필터링 파라미터를 디코딩하는 수단;

[0074] 루프 필터에서 적어도 하나의 재구성된 샘플을 필터링하는 수단 - 필터링은 적어도 하나의 디코딩된 필터링 파라미터를 적어도 하나의 재구성된 샘플의 제1 성분 및 제2 성분 둘 다를 필터링하기 위한 공통 필터링 파라미터로서 사용하는 것을 포함함 - 을 포함한다.

[0075] 본 발명에 따른 방법들 중 적어도 일부는 컴퓨터 구현될 수 있다. 그에 따라, 본 발명은 하드웨어로만 된 실시예, 소프트웨어로만 된 실시예(펌웨어, 상주 소프트웨어, 마이크로 코드 등을 포함함), 또는 소프트웨어와 하드

웨어 측면을 겸비한 실시예의 형태를 취할 수 있고, 이들 모두가 일반적으로 본 명세서에서 "회로", "모듈", 또는 "시스템"으로 지칭되어 있을 수 있다. 게다가, 본 발명은 컴퓨터 사용가능 프로그램 코드가 구현되어 있는 임의의 유형의(tangible) 표현 매체로 구현되어 있는 컴퓨터 프로그램 제품의 형태를 취할 수 있다.

[0076] 본 발명이 소프트웨어로 구현될 수 있기 때문에, 본 발명은 프로그램가능 장치 또는 임의의 적당한 전달 매체(carrier medium)에 프로비저닝하기 위한 컴퓨터 판독가능 코드로 구현될 수 있다. 유형의 전달 매체는 플로피 디스크, CD-ROM, 하드 디스크 드라이브, 자기 테이프 장치 또는 고상 메모리 장치 등과 같은 저장 매체를 포함할 수 있다. 일시적 전달 매체(transient carrier medium)는 전기 신호, 전자 신호, 광학 신호, 음향 신호, 자기 신호, 또는 전자기 신호(예컨대, 마이크로파 또는 RF 신호) 등의 신호를 포함할 수 있다.

[0077] 이와 같이, 본 발명의 제5 측면에 따르면, 프로그램가능 장치에 대한 컴퓨터 프로그램 제품이 제공되고, 컴퓨터 프로그램 제품은 프로그램가능 장치에 로드되어 실행될 때 본 발명의 전술한 제1 또는 제2 측면을 실시하는 방법의 각각의 단계를 구현하는 명령어 시퀀스를 포함한다.

[0078] 이와 유사하게, 본 발명의 제6 측면에 따르면, 본 발명의 전술한 제1 또는 제2 측면을 실시하는 방법의 각각의 단계를 구현하는 컴퓨터 프로그램의 명령어들을 저장하는 컴퓨터 판독가능 저장 매체가 제공된다.

도면의 간단한 설명

[0079] 이하의 도면들을 참조하여, 단지 예로서, 본 발명의 실시예들이 이제부터 기술될 것이다.

도 1은 종래 기술의 루프 필터링 프로세스의 단계들을 나타낸 플로우차트.

도 2는 종래 기술의 HEVC 프로세스의 샘플 적응적 대역 오프셋 분류를 그래픽으로 나타낸 도면.

도 3은 HEVC의 SAO 대역 오프셋에 대한 보상 오프셋을 결정하는 프로세스의 단계들을 나타낸 플로우차트.

도 4는 레이트 왜곡 관점으로부터 SAO 오프셋을 선택하는 프로세스를 나타낸 플로우차트.

도 5는 HEVC의 SAO 대역 오프셋에 대한 SAO 대역 위치를 결정하는 방법의 단계들을 나타낸 플로우차트.

도 6은 SAO 루프 필터에 따라 프레임 영역을 필터링하는 방법의 단계들을 나타낸 플로우차트.

도 7은 본 발명의 하나 이상의 실시예들이 구현될 수 있는 데이터 통신 시스템을 개략적으로 나타낸 블록도.

도 8은 본 발명의 하나 이상의 실시예들이 구현될 수 있는 처리 장치의 구성요소들을 나타낸 블록도.

도 9는 본 발명의 실시예들에 따른 인코딩 방법의 단계들을 나타낸 플로우차트.

도 10은 본 발명의 하나 이상의 실시예들에 따른 루프 필터링 프로세스의 단계들을 나타낸 플로우차트.

도 11은 본 발명의 실시예들에 따른 디코딩 방법의 단계들을 나타낸 플로우차트.

도 12는 적응 파라미터 세트(Adaptive Parameter Set) 모드에 따라 비트스트림으로부터 SAO 파라미터들을 획득하는 방법의 단계들을 나타낸 플로우차트.

도 13은 LCU 모드에 따라 비트스트림으로부터 SAO 파라미터들을 획득하는 방법의 단계들을 나타낸 플로우차트.

도 14는 비트스트림에서 SAO 파라미터들을 판독하는 방법의 단계들을 나타낸 플로우차트.

도 15 및 도 16은 현재 프레임 영역을 처리하기 위해 루프 필터에서 사용될 파라미터들을 결정하는 본 발명의 제1 실시예에 따른 방법의 단계들을 나타낸 플로우차트.

도 17은 한 성분과 연관되어 있는 오프셋들의 값들이 제2 성분과 연관되어 있는 오프셋들의 미러링된 값들에 대응하는 특정의 실시예를 나타낸 도면.

도 18은, 대역 오프셋 분류를 사용하여 SAO 루프 필터를 적용할 때, 제2 성분과 연관되어 있는 오프셋들의 값들이 제1 성분과 연관되어 있는 오프셋들의 값들로부터 어떻게 획득될 수 있는지를 나타낸 도면.

도 19는, 대역 오프셋 분류를 사용하여 SAO 루프 필터를 적용할 때, 제2 성분 Z와 연관되어 있는 sao_band_position 파라미터의 값이 제1 성분 X와 연관되어 있는 sao_band_position 파라미터의 값으로부터 어떻게 획득될 수 있는지를 나타낸 도면.

도 20은 레이트 왜곡 기준에 따라 2개 또는 3개의 성분에 의해 사용될 최적의 오프셋을 결정하는 방법의 단계들

을 나타낸 플로우차트.

도 21은 2개의 성분들 간에 공유되는 SAO 대역 위치를 결정하는 방법의 단계들을 나타낸 플로우차트.

도 22는 본 발명의 특정의 실시예에 따른, 비트스트림을 디코딩하는 방법의 단계들을 나타낸 플로우차트.

도 23은 하나의 성분의 SAO 분류를 다른 성분 Z를 유사하게 필터링하기 위해 사용하는 것을 나타낸 도면.

발명을 실시하기 위한 구체적인 내용

[0080]

도 7은 본 발명의 하나 이상의 실시예들이 구현될 수 있는 데이터 통신 시스템을 나타낸 것이다. 데이터 통신 시스템은 데이터 통신 네트워크(700)를 통해 데이터 스트림의 데이터 패킷들을 수신 장치[이 경우에, 클라이언트 단말(702)]로 전송하는 동작을 하는 전송 장치[이 경우에, 서버(701)]를 포함하고 있다. 데이터 통신 네트워크(700)는 WAN(Wide Area Network) 또는 LAN(Wide Area Network)일 수 있다. 이러한 네트워크는, 예를 들어, 무선 네트워크(WiFi / 802.11a 또는 b 또는 g), 이더넷 네트워크, 인터넷 네트워크, 또는 몇개의 상이한 네트워크들로 이루어진 혼합 네트워크일 수 있다. 본 발명의 특정의 실시예에서, 데이터 통신 시스템은 서버(701)가 동일한 데이터 콘텐츠를 다수의 클라이언트들로 송신하는 디지털 텔레비전 방송 시스템일 수 있다.

[0081]

서버(701)에 의해 제공되는 데이터 스트림(704)은 비디오 및 오디오 데이터를 나타내는 멀티미디어 데이터로 이루어져 있을 수 있다. 본 발명의 어떤 실시예들에서, 오디오 및 비디오 데이터 스트림은, 각각, 마이크 및 카메라를 사용하여 서버(701)에 의해 포착될 수 있다. 어떤 실시예들에서, 데이터 스트림은 다른 데이터 제공자로부터 서버(701)에 저장되거나 서버(701)에 의해 수신될 수 있거나, 또는 서버(701)에서 발생될 수 있다. 서버(701)는 특히 인코더에 입력으로서 제시되는 데이터의 보다 콤팩트한 표현인 전송을 위한 압축된 비트스트림을 제공하기 위해 비디오 및 오디오 스트림을 인코딩하는 인코더를 구비하고 있다.

[0082]

전송된 데이터의 품질 대 전송된 데이터의 양에 대한 보다 나은 비를 획득하기 위해, 비디오 데이터의 압축은, 예를 들어, HEVC 형식 또는 H.264/AVC 형식을 따를 수 있다.

[0083]

클라이언트(702)는 전송된 비트스트림을 수신하고, 디스플레이 장치 상에서 비디오 영상을 그리고 스피커에 의해 오디오 데이터를 재생하기 위해 재구성된 비트스트림을 디코딩한다.

[0084]

도 7의 예에서 스트리밍 시나리오가 고려되고 있지만, 본 발명의 어떤 실시예들에서, 인코더와 디코더 사이의 데이터 통신이, 예를 들어, 광 디스크 등의 미디어 저장 장치를 사용하여 수행될 수 있다는 것을 잘 알 것이다.

[0085]

본 발명의 하나 이상의 실시예들에서, 비디오 영상이 최종 영상에 필터링된 픽셀들을 제공하기 위해 영상의 재구성된 픽셀들에 적용하기 위한 보상 오프셋들을 나타내는 데이터와 함께 전송된다.

[0086]

도 8은 본 발명의 적어도 하나의 실시예를 구현하도록 구성되어 있는 처리 장치(800)를 개략적으로 나타낸 것이다. 처리 장치(800)는 마이크로컴퓨터, 워크스테이션, 또는 경량 휴대용 장치 등의 장치일 수 있다. 장치(800)는 다음과 같은 것들에 연결되어 있는 통신 버스(813)를 포함하고 있다:

- CPU로 표시된 마이크로프로세서 등의 중앙 처리 유닛(811);

- 본 발명을 구현하는 컴퓨터 프로그램들을 저장하는, ROM으로 표시된 판독 전용 메모리(807);

- 본 발명의 실시예들의 방법의 실행가능 코드를 저장하는 RAM으로 표시된 랜덤 액세스 메모리(812)는 물론, 본 발명의 실시예들에 따른 디지털 영상 시퀀스를 인코딩하는 방법 및/또는 비트스트림을 디코딩하는 방법을 구현하는 데 필요한 변수들 및 파라미터들을 기록하도록 구성되어 있는 레지스터들; 및

- 통신 네트워크(803)(처리될 디지털 데이터가 이를 통해 전송되거나 수신됨)에 연결되어 있는 통신 인터페이스(802).

선택적으로, 장치(800)는 또한 다음과 같은 구성요소들을 포함할 수 있다:

- 본 발명의 하나 이상의 실시예들의 방법들을 구현하는 컴퓨터 프로그램들 및 본 발명의 하나 이상의 실시예들의 구현 동안 사용되거나 생성되는 데이터를 저장하는, 하드 디스크 등의 데이터 저장 수단(804);

- 디스크(806)에 대한 디스크 드라이브(805)[디스크 드라이브는 디스크(806)로부터 데이터를 판독하거나 상기 디스크에 데이터를 기입하도록 구성되어 있음];

- 데이터를 디스플레이하는 및/또는 키보드(810) 또는 임의의 다른 포인팅 수단에 의해 사용자와의 그래픽 인터

페이지로서 역할하는 화면(809).

[0095] 장치(800)는, 예를 들어, 디지털 카메라(820) 또는 마이크(808) 등의 다양한 주변 장치들에 연결될 수 있고, 각각은 멀티미디어 데이터를 장치(800)에 제공하기 위해 입출력 카드(도시 생략)에 연결되어 있다.

[0096] 통신 버스는 장치(800)에 포함되어 있거나 그에 연결되어 있는 다양한 요소들 간의 통신 및 상호운용성을 제공한다. 버스의 표현은 제한하는 것이 아니며, 상세하게는, 중앙 처리 유닛은 명령어들을 직접 또는 장치(800)의 다른 요소에 의해 장치(800)의 임의의 요소로 전달하는 동작을 한다.

[0097] 디스크(806)는, 예를 들어, 콤팩트 디스크(CD-ROM)(재기입가능하거나 그렇지 않음), ZIP 디스크 또는 메모리 카드 등의 임의의 정보 매체로, 그리고, 일반적인 표현으로, 마이크로컴퓨터에 의해 또는 마이크로프로세서에 의해 관리될 수 있고, 장치 내에 통합되어 있거나 그렇지 않으며, 어쩌면 이동식이고 하나 이상의 프로그램들 - 그의 실행은 구현될 본 발명에 따른 디지털 영상 시퀀스를 인코딩하는 방법 및/또는 비트스트림을 디코딩하는 방법을 가능하게 함 - 을 저장하도록 구성되어 있는 정보 저장 수단으로 대체될 수 있다.

[0098] 실행가능 코드는 관리 전용 메모리(807)에, 하드 디스크(804)에, 또는 이동식 디지털 매체[예를 들어, 앞서 기술한 바와 같은 디스크(806) 등]에 저장될 수 있다. 한 변형례에 따르면, 프로그램들의 실행가능 코드는, 실행되기 전에 하드 디스크(804) 등의 장치(800)의 저장 수단들 중 하나에 저장되기 위해, 인터페이스(802)를 통해 통신 네트워크(803)에 의해 수신될 수 있다.

[0099] 중앙 처리 유닛(811)은 본 발명에 따른 프로그램 또는 프로그램들의 소프트웨어 코드의 명령어들 또는 일부분들, 상기한 저장 수단들 중 하나에 저장되어 있는 명령어들의 실행을 제어 및 지시하도록 구성되어 있다. 전원을 결 시에, 비휘발성 메모리에[예를 들어, 하드 디스크(804) 상에 또는 관리 전용 메모리(807)에] 저장되어 있는 프로그램 또는 프로그램들은 랜덤 액세스 메모리(812) 내로 전송되는 것(그러면, 랜덤 액세스 메모리는 프로그램 또는 프로그램들의 실행가능 코드를 포함하고 있음)은 물론, 본 발명을 구현하는 데 필요한 변수들 및 파라미터들을 저장하는 레지스터들로도 전송된다.

[0100] 이 실시예에서, 장치는 본 발명을 구현하기 위해 소프트웨어를 사용하는 프로그램가능 장치이다. 그렇지만, 다른 대안으로서, 본 발명은 하드웨어로[예를 들어, ASIC(Application Specific Integrated Circuit)의 형태로] 구현될 수 있다.

[0101] 도 9는 본 발명의 적어도 하나의 실시예에 따른 인코더(900)의 블록도를 나타낸 것이다. 인코더가 연결된 모듈들로 나타내어져 있고, 각각의 모듈은, 예를 들어, 장치(800)의 CPU(811)에 의해 실행될 프로그래밍 명령어들의 형태로, 본 발명의 하나 이상의 실시예들에 따른, 영상 시퀀스의 영상을 인코딩하는 적어도 하나의 실시예를 구현하는 방법의 적어도 하나의 대응하는 단계를 구현하도록 구성되어 있다.

[0102] 원래의 디지털 영상 시퀀스 i0 내지 in(901)이 인코더(900)에 의해 입력으로서 수신된다. 각각의 디지털 영상은 픽셀이라고 하는 샘플 세트로 표현된다.

[0103] 인코딩 프로세스의 구현 후에 비트스트림(910)이 인코더(900)에 의해 출력된다. 비트스트림(910)은 복수의 인코딩 단위들 또는 슬라이스들을 포함하고, 각각의 슬라이스는 슬라이스를 인코딩하는 데 사용되는 인코딩 파라미터들의 인코딩 값들을 전송하는 슬라이스 헤더(slice header), 및 인코딩된 비디오 데이터를 포함하는 슬라이스 보디(slice body)를 포함하고 있다.

[0104] 입력 디지털 영상들 i0 내지 in(901)은 모듈(902)에 의해 픽셀 블록들로 나누어진다. 블록들은 영상 일부분들에 대응하고, 가변 크기(예컨대, 4x4, 8x8, 16x16, 32x32, 64x64 픽셀)를 가질 수 있다. 각각의 입력 블록에 대해 코딩 모드가 선택된다. 2가지 계열의 코딩 모드 - 공간 예측 코딩에 기초한 코딩 모드(인트라 예측), 및 시간 예측에 기초한 코딩 모드[인터 코딩, 병합, 건너뜀] - 가 제공된다. 가능한 코딩 모드들이 테스트된다.

[0105] 모듈(903)은 인트라 예측 프로세스를 구현하고, 여기서 인코딩될 주어진 블록이 인코딩될 상기 블록의 이웃의 픽셀들로부터 계산되는 예측자에 의해 예측된다. 인트라 코딩이 선택되는 경우 잔차를 제공하기 위해, 선택된 인트라 예측자의 표시 및 주어진 블록과 그의 예측자 간의 차가 인코딩된다.

[0106] 시간 예측은 움직임 추정 모듈(904) 및 움직임 보상 모듈(905)에 의해 구현된다. 먼저, 참조 영상 세트(916) 중에서 참조 영상이 선택되고, 인코딩될 주어진 블록에 가장 가까운 영역인 참조 영상의 일부분(참조 영역 또는 영상 부분이라고도 불리움)이 움직임 추정 모듈(904)에 의해 선택된다. 움직임 보상 모듈(905)은 이어서 선택된 영역을 사용하여 인코딩될 블록을 예측한다. 선택된 참조 영역과 주어진 블록 간의 차(잔차 블록이라고도

함)가 움직임 보상 모듈(905)에 의해 계산된다. 선택된 참조 영역은 움직임 벡터로 표시된다.

[0107] 이와 같이, 이들 경우(공간 예측 및 시간 예측) 둘 다에서, 잔차는 원래의 블록으로부터 예측을 차감함으로써 계산된다.

[0108] 모듈(903)에 의해 구현되는 인트라 예측에서, 예측 방향이 인코딩된다. 시간 예측에서, 적어도 하나의 움직임 벡터가 인코딩된다.

[0109] 인터 예측이 선택되는 경우, 움직임 벡터에 대한 정보 및 잔차 블록이 인코딩된다. 비트레이트를 추가로 감소시키기 위해, 움직임이 균질하다고 가정하여, 움직임 벡터가 움직임 벡터 예측자에 대한 차에 의해 인코딩된다. 움직임 정보 예측자 세트의 움직임 벡터 예측자들이 움직임 벡터 예측 및 코딩 모듈(917)에 의해 움직임 벡터 필드(918)로부터 획득된다.

[0110] 인코더(900)는 레이트 왜곡 기준(rate-distortion criterion) 등의 인코딩 비용 기준을 적용함으로써 코딩 모드를 선택하는 선택 모듈(906)을 추가로 포함하고 있다. 중복성을 추가로 감소시키기 위해, 변환 모듈(907)에 의해 변환(DCT 등)이 잔차 블록에 적용되고, 획득되는 변환된 데이터는 이어서 양자화 모듈(908)에 의해 양자화되고, 엔트로피 인코딩 모듈(909)에 의해 엔트로피 인코딩된다. 마지막으로, 인코딩되고 있는 현재 블록의 인코딩된 잔차 블록이 비트스트림(910)에 삽입된다.

[0111] 인코더(900)는 또한, 후속 영상의 움직임 추정을 위한 참조 영상을 생성하기 위해, 인코딩된 영상의 디코딩을 수행한다. 이것은 비트스트림을 수신하는 인코더 및 디코더가 동일한 참조 프레임을 가질 수 있게 해준다. 역 양자화 모듈(911)은 양자화된 데이터의 역양자화를 수행하고, 이어서 역변환 모듈(912)에 의한 역변환이 있게 된다. 역 인트라 예측(reverse intra prediction) 모듈(913)은 주어진 블록에 대해 어느 예측자를 사용할지를 결정하기 위해 예측 정보를 사용하고, 역 움직임 보상(reverse motion compensation) 모듈(914)은 실제로 모듈(912)에 의해 획득된 잔차를 참조 영상 세트(916)로부터 획득된 참조 영역에 가산한다.

[0112] 재구성된 픽셀 프레임을 필터링하기 위해 사후 필터링이 이어서 모듈(915)에 의해 적용된다. 본 발명의 실시예들에서, 보상 오프셋이 재구성된 영상의 재구성된 픽셀의 픽셀 값에 가산되는 SAO 루프 필터가 사용된다.

[0113] 도 10은 본 발명의 적어도 하나의 실시예에 따른 루프 필터링 프로세스의 단계들을 나타낸 플로우차트이다. 초기 단계(1001)에서, 인코더는 전체 프레임의 재구성을 발생한다. 그 다음에, 단계(1002)에서, 디블록킹된 재구성(1003)을 발생시키기 위해 이 제1 재구성에 디블록킹 필터가 적용된다. 디블록킹 필터의 목표는 잔차 양자화 및 블록 움직임 보상 또는 블록 인트라 예측에 의해 발생된 블록 아티팩트를 제거하는 것이다. 이들 아티팩트는 낮은 비트레이트에서 시각적으로 중요하다. 디블록킹 필터는 2개의 이웃하는 블록들의 특성에 따라 블록 경계를 매끄럽게 하는 동작을 한다. 각각의 블록의 인코딩 모드, 잔차 코딩을 위해 사용되는 양자화 파라미터들, 및 경계에서의 이웃 픽셀 차가 고려된다. 모든 프레임들에 대해 동일한 기준/분류가 적용되고, 어떤 부가의 데이터도 전송되지 않는다. 디블록킹 필터는, 블록킹 아티팩트를 제거함으로써, 현재 프레임의 시각적 품질을 향상시키고, 또한 후속 프레임들에 대한 움직임 추정 및 움직임 보상을 향상시킨다. 실제로, 블록 아티팩트의 고주파수가 제거되고, 따라서 후속 프레임들의 텍스처 잔차(texture residual)에 대해 이를 고주파수가 보상될 필요가 없다.

[0114] 디블록킹 필터 이후에, 디블록킹된 재구성이 단계(1004)에서 픽셀들의 분류에 기초하여 SAO(sample adaptive offset) 루프 필터에 의해 필터링된다. 얻어진 프레임(1005)은 이어서 디스플레이되고 후속 인터 프레임들에 대한 참조 프레임으로서 사용될 재구성된 프레임(1007)을 발생시키기 위해 단계(1006)에서 ALF(adaptive loop filter)에 의해 필터링될 수 있다.

[0115] 단계(1004)에서, 프레임 영역의 각각의 픽셀이 그의 픽셀 값에 따라 결정된 분류의 클래스로 분류된다. 클래스는 결정된 픽셀 값 범위에 대응한다. 동일한 보상 오프셋 값이 주어진 픽셀 값 범위 내의 픽셀 값을 가지는 모든 픽셀들의 픽셀 값에 가산된다.

[0116] 샘플 적응적 오프셋 필터링을 위한 픽셀들을 디코딩하기 위해 서버와 클라이언트 사이에서 교환되고 및/또는 로컬적으로 또는 원격적으로 저장될 SAO 파라미터들을 코딩하는 것 및/또는 디코딩하는 것이 도 12 내지 도 23 중 임의의 도면을 참조하여 이후에서 보다 상세히 설명될 것이다.

[0117] 도 11은 본 발명의 일 실시예에 따른, 인코더로부터 데이터를 수신하기 위해 사용될 수 있는 디코더(1100)의 블록도를 나타낸 것이다. 디코더는 연결된 모듈들로 나타내어져 있고, 각각의 모듈은, 예를 들어, 장치(800)의 CPU(811)에 의해 실행될 프로그래밍 명령어들의 형태로, 디코더(1100)에 의해 구현되는 방법의 대응하는 단계를

구현하도록 구성되어 있다.

[0118] 디코더(1100)는 인코딩 단위들을 포함하는 비트스트림(1101)을 수신하고, 각각의 인코딩 단위는 인코딩 파라미터들에 관한 정보를 포함하는 헤더 및 인코딩된 비디오 데이터를 포함하는 보디로 이루어져 있다. 도 9와 관련하여 설명한 바와 같이, 인코딩된 비디오 데이터는 엔트로피 인코딩되어 있고, 움직임 벡터 예측자의 인덱스가, 주어진 블록에 대해, 사전 결정된 수의 비트에 인코딩되어 있다. 수신된 인코딩된 비디오 데이터는 모듈(1102)에 의해 엔트로피 디코딩된다. 잔차 데이터가 이어서 모듈(1103)에 의해 역양자화되고, 픽셀 값을 획득하기 위해 모듈(1104)에 의해 역변환이 적용된다.

[0119] 코딩 모드를 나타내는 모드 데이터가 또한 엔트로피 디코딩되고, 모드에 기초하여, 인트라 유형 디코딩 또는 인터 유형 디코딩이 인코딩된 영상 데이터 블록에 대해 수행된다.

[0120] 인트라 모드의 경우에, 비트스트림에 명시된 인트라 예측 모드에 기초하여 역 인트라 예측 모듈(1105)에 의해 인트라 예측자가 결정된다.

[0121] 모드가 인터인 경우, 인코더에 의해 사용되는 참조 영역을 찾아내기 위해 움직임 예측 정보가 비트스트림으로부터 추출된다. 움직임 예측 정보는 참조 프레임 인덱스 및 움직임 벡터 잔차로 이루어져 있다. 움직임 벡터 디코딩 모듈(1110)에 의해 움직임 벡터를 획득하기 위해 움직임 벡터 예측자가 움직임 벡터 잔차에 가산된다.

[0122] 움직임 벡터 디코딩 모듈(1110)은 움직임 예측에 의해 인코딩된 각각의 현재 블록에 대해 움직임 벡터 디코딩을 적용한다. 현재 블록에 대한 움직임 벡터 예측자의 인덱스가 획득되면, 현재 블록과 연관되어 있는 움직임 벡터의 실제 값이 디코딩되고 모듈(1106)에 의해 역 움직임 보상을 적용하는 데 사용될 수 있다. 역 움직임 보상(1106)을 적용하기 위해, 디코딩된 움직임 벡터에 의해 표시되는 참조 영상 부분이 참조 영상(1108)으로부터 추출된다. 후속하는 디코딩된 움직임 벡터의 역 예측에 사용되기 위해, 움직임 벡터 필드 데이터(1111)가 디코딩된 움직임 벡터로 업데이트된다.

[0123] 마지막으로, 디코딩된 블록이 획득된다. 도 9를 참조하여 기술된 바와 같이 인코더에서 적용되는 사후 필터링 모듈(915)과 유사하게, 사후 필터링 모듈(1107)에 의해 사후 필터링이 적용된다. 디코딩된 비디오 신호(1109)가 최종적으로 디코더(1000)에 의해 제공된다.

[0124] 도 12는 적응적 파라미터 세트(Adaptive Parameter Set, APS) 모드에 따라 비트스트림으로부터 SAO 파라미터들을 획득하는 방법의 단계들을 나타낸 플로우차트이다. 그에 따라, 도 12는 프레임 영역마다 단일의 SAO 파라미터 세트를 연관시키기 위해 적응적 파라미터 세트 모드가 사용될 때 SAO 파라미터들이 비트스트림에 어떻게 저장되는지를 나타내고 있다. 이 모드에 따르면, SAO 파라미터 세트는 LCU(Largest Coding Unit)의 크기의 배수인 크기를 가지는 직사각형 프레임 영역에 대해 유효하다. 이 특정의 모드에 대해 프레임의 SAO 파라미터 모두를 저장하기 위한 비트의 양을 감소시키기 위해, 런 길이 코딩(run-length coding) 기법은 물론, 상기 프레임 영역으로부터의 예측 플래그도 사용된다.

[0125] HEVC의 현재 규격에서 구현되는 APS 모드에 따르면, SAO 루프 필터링이 주어진 픽처의 성분들에 대해 인에이블되어 있는지 여부를 나타내는 플래그들이 있다. 이것은, 예를 들어, YUV 방식에 따라 인코딩된 픽처를 생각하고 있을 때, U 또는 V 성분에 대한 SAO 파라미터들이 인에이블되거나 디스에이블될 수 있다는 것을 의미한다.

[0126] 이러한 플래그들은 초기 단계(1201)에서 판독된다. 그 다음에, 단계(1202)에서, SAO 파라미터 세트가 적용되어야 하는 프레임 영역의 크기가 획득된다. 앞서 기술한 바와 같이, APS 모드를 사용할 때 SAO 루프 필터링이 적용될 필요가 있는 프레임 영역은 LCU 크기와 같거나 LCU 크기의 배수 크기에 대응하는 영역에 대응한다.

[0127] 그 다음에, X로 표시된 시퀀스의 각각의 색상 성분을 연속적으로 처리하기 위해 단계(1203)에서 루프가 시작된다. 이 루프는 단계(1204)에서 수행되는 각각의 성분에 대한 "단일성(unicity)" 플래그를 판독하는 것을 가능하게 한다. 그 다음에, 단계(1205)에서 판독된 "단일성" 플래그가 참인지 여부를 판정하기 위해 테스트가 수행된다. 판독된 "단일성" 플래그가 참인 경우, 전체 영상에 대한 현재 처리된 성분 X에 관한 단일의 SAO 파라미터 세트만이 존재한다. 이러한 경우에, 단계(1206)에서 비트스트림에서 SAO 파라미터들이 판독된다. 이러한 단계는 도 14를 참조하여 상세히 기술된다. 그렇지 않고 판독된 "단일성" 플래그가 참이 아닌 경우, 현재 처리된 성분 X에 관한 전체 영상에 대한 2개 이상의 SAO 파라미터가 있다. 이러한 경우에, SAO 파라미터들이 단계들(1208 내지 1218)을 참조하여 기술되는 바와 같이 후속 단계들에서 판독된다.

[0128] 그 다음에, 단계(1207)에서 현재 처리된 성분 X가 처리할 마지막 성분인지 여부를 판정하기 위해 테스트가 수행된다. 현재 처리된 성분 X가 처리할 마지막 성분인 경우, 알고리즘은 단계(1203)로 분기되고, 남아 있는 성분

(들)에 대해 이전의 단계들이 반복된다.

[0129] 유의할 점은, "단일성" 플래그가 하나의 성분에 대해 참인 경우, 단계들(1208 내지 1218)이 그 성분에 대해 수행되지 않는다.

[0130] 단계(1208)로부터, 성분마다 2개 이상의 SAO 파라미터 세트가 존재할 때 각각의 프레임 영역에 대해 그리고 각각의 성분에 대해 SAO 파라미터들이 결정된다.

[0131] 단계(1202)에서 결정된 프레임 영역 크기에 따라 처리될 주어진 꾹처의 각각의 프레임 영역을 연속적으로 처리하기 위해 단계(1208)에서 루프가 시작된다. 도 12의 주어진 예에서, 프레임 영역들이 래스터 스캔 순서 (raster scan order)로 처리되고, 그로써 프로세스는 영상의 프레임 영역들의 첫번째 행에서 시작하고 영상의 마지막 행까지 계속된다. 영상의 좌측으로부터 우측으로 프레임 영역들이 처리된다.

[0132] 그 다음에, X로 표시된 각각의 성분을 연속적으로 처리하기 위해 단계(1209)에서 루프가 시작된다. 유의할 점은, 단계(1201)에서 판독된, 현재 성분의 "인에이블" 플래그가 거짓인 경우, 단계(1209)에서 관련 성분에 대한 그 다음 단계들이 무시되고, 그 다음 성분이 처리되기 위해 선택된다. 단계(1210)에서, "행 반복 플래그 (repeat row flag)"가 판독된다. 그 다음에, 단계(1211)에서, 그 플래그가 참인지 여부를 판정하기 위해 테스트가 수행된다. 참인 경우, 프레임 영역의 이전 행의 SAO 파라미터들이 프레임 영역의 현재 행에 대해 사용된다. 이것은 현재 행이 이전 행과 동일한 파라미터들을 사용한다는 것만을 신호하는 것에 의해 SAO 파라미터들을 비트스트림에 표현하는 데 많은 비트를 절감하는 것을 가능하게 만들어 준다.

[0133] 그렇지 않고, 판독된 "행 반복 플래그"가 거짓인 경우, 현재 프레임 영역이 비트스트림에 코딩되고, 알고리즘이 단계(1213)로 분기하여, 현재 행에 이전 프레임 영역과 연관되어 있는 SAO 파라미터들을 인코딩하고 표현하기 위해 (런 길이 코딩 또는 상기 프레임 영역의 SAO 파라미터들의 값을 사용하는) 예측 기법이 사용되는지 여부를 판정하기 위해 테스트가 수행된다. 그 경우에, 현재 행에서 단계(1214)에서 이전 프레임 영역으로부터 현재 프레임 영역의 SAO 파라미터들이 추론된다. 보다 구체적으로는, 2개의 구문 요소 - SAO 파라미터들이 런 길이 방법에 따라 코딩되는지 또는 상기 프레임 영역의 SAO 파라미터들로부터 추론되는지를 각각 나타내는 sao_run_diff 및 sao_merge_up_flag - 가 사용된다. 그렇지 않고, 단계(1213)에서 대답이 아니오인 경우, 이 특정의 프레임 영역에 대한 SAO 파라미터들이 비트스트림에 명시적으로 코딩된다. 그에 따라, 단계(1215)에서 비트스트림으로부터 이들 SAO 파라미터가 판독된다. 이 단계는 1206과 유사하고, 도 14를 참조하여 추가로 기술된다.

[0134] 단계(1212) 이후, 전체 행에 대해 SAO 파라미터들이 결정될 때, 단계(1216)에서 현재 성분이 처리될 마지막 성분인지 여부를 판정하기 위해 테스트가 수행된다. 처리될 마지막 성분이 아닌 경우, 알고리즘은 단계(1209)로 분기되어 그 다음 성분(들)을 처리한다. 이와 유사하게, 단계(1214) 또는 단계(1215)를 수행하는 것에 의해 현재 프레임 영역에 대해 SAO 파라미터들이 결정될 때, 알고리즘은 단계(1216)로 분기하여 현재 성분이 처리될 마지막 성분인지 여부를 판정한다. 다시 말하지만, 처리될 마지막 성분이 아닌 경우, 알고리즘은 단계(1209)로 분기되어 그 다음 성분(들)을 처리한다.

[0135] 그렇지 않고, 모든 성분들이 처리된 경우, 단계(1217)에서 마지막 프레임 영역이 처리되었는지 여부를 판정하기 위해 테스트가 수행된다. 마지막 프레임 영역이 처리되지 않은 경우, 프로세스는 단계(1208)로 분기되고, 이전의 단계들이 반복된다. 그렇지 않은 경우, 프로세스가 종료한다.

[0136] 이 스테이지에서, 도 14를 참조하여 제시되는 상이한 파라미터들을 갖는 SAO 파라미터 세트가 각각의 성분에 대해 각각의 프레임 영역과 연관되어 있다.

[0137] 도 13은 LCU 모드에 따라 비트스트림으로부터 SAO 파라미터들을 획득하는 방법의 단계들을 나타낸 플로우차트이다. 따라서, 도 13은 SAO 파라미터들을 APS 모드로 인코딩하는 대안을 나타낸 것이다. 보다 구체적으로는, 도 13은 LCU(Largest Coding Unit)마다 단일의 SAO 파라미터 세트를 연관시키기 위해 LCU 모드 표현이 사용될 때 SAO 파라미터들이 비트스트림에 어떻게 저장될 수 있는지를 나타내고 있다. LCU마다 하나의 SAO 파라미터 세트를 인코딩하는 것(이는 비용이 많이 소요됨)을 피하기 위해 예측 방식이 사용될 수 있다. 주어진 예에서, 이 예측 모드는 현재 LCU의 좌측에 배치된 LCU가 동일한 SAO 파라미터들을 사용하는지를 검사하는 것이다. 아니오인 경우, 현재 LCU 위쪽에 배치된 LCU에 대해 제2 검사가 수행된다. 이 예측 기법은 LCU 모드에 대한 SAO 파라미터들을 표현하는 데이터의 양의 감소를 가능하게 한다. 이는 병합(Merge) 플래그라고 하는 전용 플래그의 사용에 기초하고 있다.

[0138] 단계(1301)에서, X로 표시된 각각의 성분을 연속적으로 처리하기 위해 루프가 시작된다. 예시를 위해, 영상들

이 여전히 YUV 방식에 따라 인코딩된다. 그 다음에, 단계(1303)에서, sao_merge_left_flag_X(X는 Y, U, 또는 V를 나타냄)로 표시된 플래그가 처리될 비트스트림(1302)으로부터 판독되고 디코딩된다. 그의 값이 참인 경우, 단계(1304)에서, 현재 LCU의 좌측에 배치된 LCU의 SAO 파라미터들이 현재 LCU를 처리하는 데 사용되기 위해 선택된다. 보다 구체적으로는, 이것은 단계(1308)에서 현재 LCU에 적용될 SAO 루프 필터의 유형을 결정하는 것을 가능하게 한다.

[0139] 그렇지 않고, sao_merge_left_flag_X로 표시된 판독된 플래그가 거짓인 경우, sao_merge_up_flag_X(X는 Y, U, 또는 V를 나타냄)로 표시된 플래그가 비트스트림(1302)으로부터 판독되고 디코딩된다. 그의 값이 참인 경우, 단계(1306)에서, 현재 LCU의 위쪽에 배치된 LCU의 SAO 파라미터들이 현재 LCU를 처리하는 데 사용되기 위해 선택된다. 다시 말하지만, 이것은 단계(1308)에서 현재 LCU에 적용될 SAO 루프 필터의 유형을 결정하는 것을 가능하게 한다.

[0140] 그렇지 않고, sao_merge_up_flag_X로 표시된 판독된 플래그가 거짓인 경우, 단계(1307)에서 현재 LCU를 처리하는 데 사용될 SAO 파라미터들이 비트스트림(1302)으로부터 판독되고 디코딩되고, 후자는 도 14를 참조하여 기술된다.

[0141] 이 스테이지에서, 단계(1308)에서 현재 LCU를 처리하기 위한 SAO 파라미터들이 획득되고 현재 LCU에 적용될 SAO 루프 필터의 유형이 결정된다.

[0142] 그 다음에, 단계(1309)에서, 현재 LCU에 대해 모든 성분들 Y, U 및 V가 처리되었는지 여부를 판정하기 위해 테스트가 수행된다. 모든 성분들이 처리되지는 않은 경우, 그 다음 성분이 선택되고, 알고리즘은 단계(1301)로 분기되며, 이전의 단계들이 반복된다.

[0143] 그 다음에, 그 다음 LCU가 이와 유사하게 처리될 수 있다.

[0144] 도 14는 비트스트림에서 SAO 파라미터들을 판독하는 방법의 단계들을 나타낸 플로우차트이다. 그에 따라, 도 14는 SAO 파라미터들이 인코딩된 비트스트림에 저장되는 방식을 나타내고 있다.

[0145] 초기 단계(1402)에서, sao_type_idx_X(X는 Y, U, 또는 V를 나타냄)로 표시된 플래그가 비트스트림(1401)으로부터 판독되고 디코딩된다. 이 구문 요소를 나타내는 코드워드는 고정 길이 코드 또는 임의의 산술 코딩 방법을 사용할 수 있다.

[0146] 부록의 표 1을 참조하여 앞서 기술한 바와 같이, 이 구문 요소는 성분 X를 처리하기 위해 프레임 영역에 적용될 SAO 루프 필터의 유형을 결정하는 것을 가능하게 만들어 준다. 주어진 예에 따르면, 플래그 sao_type_idx_X(또는 총칭하여 sao_type_idx)는 적용될 SAO 루프 필터에 의존하는 5개의 값을 중 하나를 취할 수 있다. 부록의 표 1에 나타낸 바와 같이, 0은 SAO 루프 필터 없음에 대응하고, 1 내지 4는 부록의 표 2를 참조하여 기술된 바와 같은 예지 오프셋 SAO 루프 필터의 4가지 카테고리에 대응하며, 값 5는 대역 오프셋 SAO 루프 필터에 대응한다.

[0147] flag sao_type_idx_X를 판독하고 디코딩한 후에, 단계(1402)에서 이 인덱스의 값이 0 초과(strictly positive)인지 여부를 판정하기 위해 테스트가 수행된다. 이 인덱스의 값이 0인 경우 - 이는 현재 프레임 영역에 대해 SAO 루프 필터 없음(no SAO loop filter)이 적용되어야 한다는 것을 의미함 -, 단계(1407)에서 SAO 파라미터들의 결정이 종료한다. 그렇지 않고, 플래그 sao_type_idx_X의 값이 0 초과인 경우, 현재 프레임 영역에 대한 SAO 파라미터들이 비트스트림(1401)에 존재한다.

[0148] 그 다음에, 단계(1403)에서 적용될 SAO 루프 필터의 유형이 대역 오프셋 유형인지 여부(sao_type_idx_X == 5)를 판정하기 위해 테스트가 수행된다. 적용될 SAO 루프 필터가 대역 오프셋 유형인 경우, 단계(1404)에서 비트스트림(1401)에서 SAO 대역의 위치가 판독되고 디코딩된다.

[0149] 단계(1404)에서 SAO 대역의 위치를 판독한 후에, 단계(1406)에서 비트스트림(1401)으로부터 4개의 $Offset_j$ (j 는 1부터 4까지 변함)를 판독하고 디코딩하기 위해 단계(1405)에서 루프가 시작된다. 이들 4개의 오프셋은 도 2를 참조하여 기술된 SAO 대역 오프셋의 4개의 범위에 관련된 4개의 오프셋에 대응한다.

[0150] 이와 유사하게, 플래그 sao_type_idx_X의 값이 플러스이고 5와 상이한 경우, 비트스트림(1401)으로부터 4개의 $Offset_j$ (j 는 1부터 4까지 변함)를 판독하고 디코딩하기 위해 단계(1405)에서 루프가 시작된다. 이들 4개의 오프셋은 부록의 표 2를 참조하여 기술된 SAO 대역 오프셋의 4개의 유형의 4개의 오프셋에 대응한다.

[0151] 단계(1406)에서 4개의 오프셋이 판독되고 디코딩되었을 때, 단계(1407)에서 SAO 파라미터들의 판독이 종료한다.

[0152] 제1 주된 실시예에 따르면, SAO 파라미터들 중 일부 또는 전부가 성분들 간에 공유된다. 그를 위해, 하나의 성분으로부터 다른 성분으로 SAO 파라미터들을 도출하는 프로세스들이 정의된다. 2가지 종류의 SAO 분류(에지 오프셋 및 대역 오프셋)에 대해 성분들 간의 이러한 의존 관계가 정의될 수 있다.

[0153] 도 13을 참조하여 앞서 기술한 바와 같이, SAO 파라미터들을 이전에 처리된 SAO 파라미터들의 함수로서 획득하기 위해, LCU 모드에 대한 HEVC의 현재 구현에서 sao_merge_left_flag_X 및 sao_merge_up_flag_X로 표시된 2개의 플래그가 사용된다. sao_merge_left_flag_X로 표시된 플래그는 현재 LCU의 성분 X를 처리하는 데 사용될 SAO 파라미터들이 현재 LCU의 좌측에 배치된 LCU의 동일한 성분을 처리하는 데 사용되는 것이라고 명시하는 것을 목적으로 한다. 이와 유사하게, sao_merge_up_flag_X로 표시된 플래그는 현재 LCU의 성분 X를 처리하는 데 사용될 SAO 파라미터들이 현재 LCU의 위쪽에 배치된 LCU의 동일한 성분을 처리하는 데 사용되는 것이라고 명시하는 것을 목적으로 한다.

[0154] 특정의 실시예에 따르면, 이들 플래그는 현재 LCU의 2개 또는 3개의 성분에 대해 공통이다. 환언하면, sao_merge_left_flag 및 sao_merge_up_flag 유형의 플래그들이 현재 LCU의 성분을 처리하는 데 사용될 SAO 파라미터들이 다른 LCU의 동일한 성분을 처리하는 데 사용되는 것이라고 명시하기 위해 더 이상 사용되지 않고, 현재 LCU의 2개 또는 3개의 성분을 처리하는 데 사용될 SAO 파라미터들이 다른 LCU의 동일한 2개 또는 3개의 성분을 처리하는 데 사용되는 것이라고 명시하기 위해 사용된다. 예시를 위해, sao_merge_left_flag_UV 및 sao_merge_up_flag_UV로 표시된 플래그들은 현재 LCU의 U 및 V 성분을 처리하는 데 사용될 SAO 파라미터들이, 각각, 현재 LCU의 좌측에 또는 위쪽에 배치된 LCU의 U 및 V 성분을 처리하는 데 사용된 것이라고 명시하기 위해 고려될 수 있다.

[0155] 다른 특정의 실시예에 따르면, 플래그들이 현재 LCU의 2개 또는 3개의 성분을 처리하는 데 사용될 SAO 파라미터들이 다른 LCU의 성분을 처리하는 데 사용되는 것이라고 명시하기 위해 사용된다. 그 결과, 현재 LCU의 성분과 연관되어 있는 플래그가 좌측 또는 위쪽 LCU 중 어느 것이 참조로서 사용되어야 하는지, 그리고 그 LCU에서, SAO 파라미터들을 획득하기 위해 성분들 중 어느 것이 사용되어야 하는지를 나타낼 수 있다. 따라서, 플래그들이 다음과 같은 것들일 수 있다: sao_merge_left_flag_Y, sao_merge_left_flag_U, sao_merge_left_flag_V, sao_merge_up_flag_Y, sao_merge_up_flag_U, 및 sao_merge_up_flag_V. 예시를 위해, 플래그 sao_merge_left_flag_V를 참조하는 현재 LCU의 성분의 SAO 파라미터들은 현재 LCU의 좌측에 배치된 LCU의 성분 V와 연관되어 있는 SAO 파라미터들이다.

[0156] 이 실시예에서, LCU의 각각의 성분과 연관되어 있는 SAO 파라미터들이, 다른 LCU의 SAO 파라미터들을 참조할 때, (이 LCU의 각각의 성분에 대해) 동일한 것이 바람직하다. 예시를 위해, 성분이 플래그 sao_merge_left_flag_V를 참조하는 현재 LCU의 모든 성분들의 SAO 파라미터들은 현재 LCU의 좌측에 배치된 LCU의 성분 V와 연관되어 있는 SAO 파라미터들이다. 그에 따라, 좌측 루마(Y) 성분의 SAO 파라미터들이 현재 LCU의 크로마(U) 성분을 필터링하는 데 사용될 수 있다.

[0157] 도 15는 현재 프레임 영역을 처리하기 위해 루프 필터에서 사용될 파라미터들을 결정하는 본 발명의 제1 실시예에 따른 방법의 단계들을 나타낸 플로우차트이다. 도 15에 주어진 예는 앞서 기술한 2개의 특정의 실시예의 결합으로 볼 수 있다.

[0158] 단계(1502)에서, sao_merge_left_flag_Y로 표시된 플래그가 비트스트림(1501)으로부터 획득되고, 이 플래그의 값이 1인지 여부(즉, 참인지)를 판정하기 위해 테스트가 수행된다. sao_merge_left_flag_Y로 표시된 플래그가 참인 경우, 단계(1503)에서, 현재 LCU의 성분들 Y, U 및 V를 필터링하는 데 사용될 SAO 파라미터들이 현재 LCU의 좌측에 배치된 LCU에서의 성분 Y를 필터링하기 위해 이전에 결정된 SAO 파라미터들과 같은 것으로 판정된다.

[0159] 그에 따라, 현재 LCU의 성분 U 및 V에 적용될 루프 필터의 유형을 정의하는, 현재 LCU에 대한 플래그 sao_type_idx_U 및 sao_type_idx_V의 값이 현재 LCU의 좌측에 배치된 LCU의 플래그 sao_type_idx_Y의 값과 같은 것으로 간주된다. 마찬가지로, 현재 LCU의 성분들 U 및 V를 필터링하는 데 사용되는 다른 SAO 파라미터들(예를 들어, 오프셋 값들) 및, 필요한 경우, 대역 위치들(sao_band_position_U 및 sao_band_position_V)이 현재 LCU의 좌측에 배치된 LCU의 성분 Y의 SAO 파라미터들로부터 획득된다.

[0160] 그렇지 않고, sao_merge_left_flag_Y로 표시된 플래그가 거짓인 경우, sao_merge_up_flag_Y로 표시된 플래그가 비트스트림(1501)으로부터 획득되고, 단계(1504)에서 이 플래그의 값이 1인지 여부(즉, 참인지)를 판정하기 위해 테스트가 수행된다. sao_merge_up_flag_Y로 표시된 플래그가 참인 경우, 단계(1505)에서, 현재 LCU의 성분들 Y, U 및 V를 필터링하는 데 사용될 SAO 파라미터들이 현재 LCU의 위쪽에 배치된 LCU에서의 성분 Y를 필터링

하기 위해 이전에 결정된 SAO 파라미터들과 같은 것으로 판정된다.

[0161] 그에 따라, 현재 LCU의 성분들 U 및 V에 적용될 루프 필터의 유형을 정의하는, 현재 LCU에 대한 플래그 sao_type_idx_U 및 sao_type_idx_V의 값이 현재 LCU의 위쪽에 배치된 LCU의 플래그 sao_type_idx_Y의 값과 같은 것으로 간주된다. 마찬가지로, 현재 LCU의 성분들 U 및 V를 필터링하는 데 사용되는 다른 SAO 파라미터들(예를 들어, 오프셋 값들) 및, 필요한 경우, 대역 위치들(sao_band_position_U 및 sao_band_position_V)이 현재 LCU의 위쪽에 배치된 LCU의 성분 Y의 SAO 파라미터들로부터 획득된다.

[0162] 그렇지 않고, sao_merge_up_flag_Y로 표시된 플래그가 거짓인 경우, 도 14를 참조하여 앞서 기술한 바와 같이, 단계(1511)에서 현재 LCU의 성분 Y에 대한 SAO 파라미터들이 비트스트림(1501)으로부터 추출된다. 이러한 경우에, 플래그들 sao_type_idx_Y, sao_band_position_Y(필요한 경우), 및 4개의 offsetj_Y의 값이 비트스트림(1501)으로부터 추출된다.

[0163] 그 다음에, sao_merge_up_flag_Y로 표시된 플래그가 거짓인 경우, 단계(1506)에서, sao_merge_left_flag_UV로 표시된 플래그가 비트스트림(1501)으로부터 획득되고, 이 플래그의 값이 1인지 여부(즉, 참인지)를 판정하기 위해 테스트가 수행된다. sao_merge_left_flag_UV로 표시된 플래그가 참인 경우, 단계(1507)에서, 현재 LCU의 성분들 Y, U 및 V를 필터링하는 데 사용될 SAO 파라미터들이 현재 LCU의 좌측에 배치된 LCU에서의 성분들 U 및 V를, 각각, 필터링하기 위해 이전에 결정된 SAO 파라미터들과 같은 것으로 판정된다.

[0164] 그렇지 않고, sao_merge_left_flag_UV로 표시된 플래그가 거짓인 경우, sao_merge_up_flag_UV로 표시된 플래그가 비트스트림(1501)으로부터 획득되고, 단계(1508)에서 이 플래그의 값이 1인지 여부(즉, 참인지)를 판정하기 위해 테스트가 수행된다. sao_merge_up_flag_UV로 표시된 플래그가 참인 경우, 단계(1509)에서, 현재 LCU의 성분들 U 및 V를 필터링하는 데 사용될 SAO 파라미터들이 현재 LCU의 위쪽에 배치된 LCU에서의 성분들 U 및 V를, 각각, 필터링하기 위해 이전에 결정된 SAO 파라미터들과 같은 것으로 판정된다.

[0165] 그렇지 않고, sao_merge_up_flag_UV로 표시된 플래그가 거짓인 경우, 도 14를 참조하여 앞서 기술한 바와 같이, 단계(1510)에서 현재 LCU의 성분들 U 및 V에 대한 SAO 파라미터들이 비트스트림(1501)으로부터 추출된다. 이러한 경우에, 플래그들 sao_type_idx_U, sao_type_idx_V, sao_band_position_U(필요한 경우), sao_band_position_V(필요한 경우), 4개의 offsetj_U, 및 4개의 offsetj_V의 값이 비트스트림(1501)으로부터 추출된다. 이 스테이지[참조 번호(1512)]에서, 현재 LCU의 모든 성분들에 대한 모든 SAO 파라미터들이 이용가능하다.

[0166] 그 다음에, 단계(1513)에서, SAO 루프 필터가 현재 LCU의 모든 성분들에 적용되고, 디코딩 프로세스는 그 다음 디코딩 단계(1314)로 간다.

[0167] 여전히 특정의 실시예에 따르면, 현재 LCU의 모든 성분들에 공통인 플래그들(예를 들어, sao_merge_left_flag_YUV 및 sao_merge_up_flag_YUV 플래그들)이 사용될 수 있다. 이러한 경우에, 이들 플래그 둘 다가 거짓인 경우, 각각의 성분에 관련된 플래그들[총칭하여, sao_merge_left_flag_X 및 sao_merge_up_flag_X(X는 Y, U, 또는 V를 나타냄)라고 지칭됨]이 사용될 수 있다.

[0168] 도 16은 현재 프레임 영역을 처리하기 위해 루프 필터에서 사용될 파라미터들을 결정하는 본 발명의 제1 실시예에 따른 방법의 단계들을 나타낸 플로우차트이다. 도 16에 주어진 예는 보다 구체적으로는 앞서 기술된 제3 특정의 실시예에 관한 것이다.

[0169] 단계(1602)에서, sao_merge_left_flag_YUV로 표시된 플래그가 비트스트림(1601)으로부터 획득되고, 이 플래그의 값이 1인지 여부(즉, 참인지)를 판정하기 위해 테스트가 수행된다. sao_merge_left_flag_YUV로 표시된 플래그가 참인 경우, 단계(1603)에서, 현재 LCU의 성분들 Y, U 및 V를 필터링하는 데 사용될 SAO 파라미터들이 현재 LCU의 좌측에 배치된 LCU에서의 성분들 Y, U 및 V를, 각각, 필터링하기 위해 이전에 결정된 SAO 파라미터들과 같은 것으로 판정된다.

[0170] 그에 따라, 현재 LCU의 성분들 Y, U 및 V에 적용될 루프 필터의 유형을 정의하는, 현재 LCU에 대한 플래그 sao_type_idx_Y, sao_type_idx_U 및 sao_type_idx_V의 값이 현재 LCU의 좌측에 배치된 LCU의 대응하는 플래그들의 값과 같은 것으로 간주된다. 마찬가지로, 현재 LCU의 성분들 Y, U 및 V를 필터링하는 데 사용되는 다른 SAO 파라미터들이 현재 LCU의 좌측에 배치된 LCU의 대응하는 성분의 SAO 파라미터들로부터 획득된다.

[0171] 그렇지 않고, sao_merge_left_flag_YUV로 표시된 플래그가 거짓인 경우, sao_merge_up_flag_YUV로 표시된 플래그가 비트스트림(1601)으로부터 획득되고, 단계(1604)에서 이 플래그의 값이 1인지 여부(즉, 참인지)를 판정하

기 위해 테스트가 수행된다. `sao_merge_up_flag_YUV`로 표시된 플래그가 참인 경우, 단계(1605)에서, 현재 LCU의 성분들 Y, U 및 V를 필터링하는 데 사용될 SAO 파라미터들이 현재 LCU의 위쪽에 배치된 LCU에서의 성분들 Y, U 및 V를, 각각, 필터링하기 위해 이전에 결정된 SAO 파라미터들과 같은 것으로 판정된다.

[0172] 그렇지 않고, `sao_merge_up_flag_YUV`로 표시된 플래그가 거짓인 경우, SAO 파라미터들이 이전의 처리된 SAO 파라미터들의 합수로서 획득되어야 하는지 여부를 판정하기 위해, 현재 LCU의 각각의 성분과 연관되어 있는 플래그들이 디코딩된다.

[0173] 그를 위해, 현재 LCU의 각각의 성분을 순차적으로 처리하기 위해 단계(1606)에서 루프가 시작된다. 각각의 성분 X(X는 Y, U 또는 V임)에 대해, `sao_merge_left_flag_X`로 표시된 플래그가 비트스트림(1601)으로부터 추출되고, 단계(1607)에서 이 플래그의 값이 1인지 여부(즉, 참인지)를 판정하기 위해 테스트가 수행된다. `sao_merge_left_flag_X`로 표시된 플래그가 참인 경우, 단계(1608)에서, 현재 LCU의 성분 X를 필터링하는 데 사용될 SAO 파라미터들이 현재 LCU의 좌측에 배치된 LCU에서의 동일한 성분을 필터링하기 위해 이전에 결정된 SAO 파라미터들과 같은 것으로 판정된다.

[0174] 그렇지 않고, `sao_merge_left_flag_X`로 표시된 플래그가 거짓인 경우, `sao_merge_up_flag_X`로 표시된 플래그가 비트스트림(1601)으로부터 획득되고, 단계(1609)에서 이 플래그의 값이 1인지 여부(즉, 참인지)를 판정하기 위해 테스트가 수행된다. `sao_merge_up_flag_X`로 표시된 플래그가 참인 경우, 단계(1610)에서, 현재 LCU의 성분을 필터링하는 데 사용될 SAO 파라미터들이 현재 LCU의 위쪽에 배치된 LCU에서의 동일한 성분을 필터링하기 위해 이전에 결정된 SAO 파라미터들과 같은 것으로 판정된다.

[0175] 그렇지 않고, `sao_merge_up_flag_X`로 표시된 플래그가 거짓인 경우, 단계(1611)에서 현재 LCU에 대한 성분 X에 대한 SAO 파라미터들이 비트스트림(1601)으로부터 추출된다.

[0176] 그 다음에, 현재 LCU의 성분 X에 대한 SAO 파라미터들을 획득한 후에, 단계(1612)에서, 성분 X가 성분 V인지 여부를 판정하기 위해 테스트가 수행된다. 성분 X가 성분 V가 아닌 경우, 프로세스는 단계(1606)로 분기되어 그 다음 성분을 처리한다. 그렇지 않고, 성분 X가 성분 V인 경우, 단계(1613)에서 SAO 파라미터들이 디코딩되고 현재 LCU의 모든 성분들에 대해 이용가능하다.

[0177] 그 다음에, 단계(1614)에서, 현재 LCU의 성분들이 획득된 SAO 파라미터들을 사용하여 SAO 루프 필터에서 필터링되고, 디코더는 단계(1615)에서의 그 다음 디코딩 단계로 간다.

[0178] 유의할 점은, 도 15 및 도 16을 참조하여 기술된 실시예들이 도 13을 참조하여 기술된 LCU 구문에 기초한다는 것이다. 그렇지만, 이를 실시예는 또한 도 12를 참조하여 기술된 APS에 대한 SAO 구문에 기초할 수 있다. 그 특정의 경우에, 성분들 간에 공유되는 파라미터들은 SAO 파라미터들은 물론 unicity 및 repeat row로 표시된 플래그들 그리고 `sao_run_diff` 및 `sao_merge_up_flag`로 표시된 구문 요소들이다.

[0179] 상세하게는 부록의 표 1을 참조하여 앞서 기술한 바와 같이, `sao_type_idx`로 표시된 구문 요소 또는 플래그는 그의 값이 0일 때 현재 LCU의 현재 성분이 처리되지 않는다는 것을 명시한다. 그렇지 않은 경우, 이는 SAO 유형을 명시한다. 플래그 `sao_type_idx`의 값이 1, 2, 3, 또는 4인 경우, 현재 LCU를 필터링하기 위해 에지 오프셋 분류가 사용되고, 플래그 `sao_type_idx`의 값이 5인 경우, 현재 LCU를 필터링하기 위해 SAO 대역 오프셋이 사용된다.

[0180] 앞서 기술한 바와 같은 본 발명의 실시예들에 따르면, `sao_type_idx`로 표시된 이러한 플래그들이 2개 또는 3개의 성분들 간에 공유될 수 있다.

[0181] 다른 특정의 실시예에서, 성분들은 `sao_type_idx`로 표시된 플래그의 어떤 값들만을 공유할 수 있다. 환연하면, 제2 성분은 제1 성분의 `sao_type_idx`로 표시된 플래그의 값이 사전 결정된 값 세트에 속하는 경우에만, 이 값을 공유한다. 따라서, 이 실시예에 따르면, `sao_type_idx`로 표시된 플래그의 값이 조건부로 공유된다.

[0182] 예시를 위해, 성분 U를 갖는 주어진 LCU에 대해, 성분 Y와 연관되어 있는 `sao_type_idx`로 표시된 플래그의 값이 공유되는 것으로 가정될 수 있다. 그렇지만, 이 값이 에지 오프셋 유형의 SAO 분류를 나타내는 경우(즉, `sao_type_idx` = 1, 2, 3, 또는 4인 경우), 에지 오프셋의 방향이 그럼에도 불구하고, 성분 Y에 의해 주어지는 것과 같은 에지 오프셋 분류의 방향을 사용하는 대신에, 성분 U와 연관되어 있는 비트스트림 일부로부터 획득될 수 있는 것이 가능하게 될 수 있다. 성분 V와 연관되어 있는 `sao_type_idx`로 표시된 플래그의 값이 에지 오프셋이 아닌 경우(`sao_type_idx` = 0 또는 5인 경우), 어떤 방향 정보도 비트스트림으로부터 추출되지 않는다. 환연하면, 성분 Y에 대한 SAO 유형이 0 또는 5인 경우, (유형에 관한) 이 SAO 파라미터가 성분들 U 및 V에 의해

사용될 것이다. 그렇지만, SAO 유형이 성분 Y에 대한 예지 오프셋 유형(즉, 1부터 4까지의 값)인 경우, 성분들 U 및 V에 대해 전용 SAO 유형들이 사용될 것이다.

[0183] 앞서 기술한 바와 같이, (예지 오프셋 유형의 픽셀 분류를 사용하는 SAO 루프 필터와 연관되어 있는) SAO 파라미터들은 5개의 가능한 클래스들 중 4개의 클래스에 관련되어 있는 4개의 오프셋을 포함한다. 제1 클래스(클래스 "0")는 SAO 필터링 없음에 대응한다. 여기서 상기할 점은, 예지 오프셋 분류의 방향이 SAO 유형에서 신호된다는 것이다.

[0184] 본 발명의 실시예에 따르면, 예지 오프셋 분류의 방향과 무관하게, 현재 LCU에 대해 예지 오프셋 필터가 선택될 때, 이들 4개의 오프셋이 2개 또는 3개의 성분들 간에 공유될 수 있다.

[0185] 제1 특정의 실시예에 따르면, 2개 또는 3개의 성분들에 대해 동일한 오프셋들이 사용된다. 그 결과, 3개의 성분을 갖는 픽셀을 필터링하기 위해 2개 또는 3개의 성분이 동일한 4개의 오프셋을 공유하는 경우, 각각, 8개 또는 12개 대신에, 단지 4개의 오프셋이 비트스트림으로부터 추출되면 된다.

[0186] 자연히, 2개 또는 3개의 성분을 동일한 오프셋들로 처리하는 것을 가능하게 하기 위해 이러한 공통 오프셋들이 인코더측에서 계산될 필요가 있다. 이하에서 기술되는 바와 같이, 이러한 인코딩 선택은 대역 오프셋 필터링을 위해 사용되는 것과 동일하다.

[0187] 다른 특정의 실시예에서, 성분 Z와 연관되어 있는 오프셋들 $0z_j$ 는 다음과 같이 가중 인자 β_j 에 따라 다른 성분 X의 오프셋들 $0x_j$ 로부터 도출된다:

$$0z_j = \beta_j 0x_j$$

[0188] 가중 인자 β_j 는 j 의 값이 ($\beta = \beta_j$)인지 여부에 관계없이 동일할 수 있다. β 의 값 또는 β_j 의 값들이 사전 결정될 수 있고 일정할 수 있거나, 시퀀스 헤더 또는 슬라이스 헤더와 같은 특정의 헤더들에서 전송될 수 있거나, 심지어 LCU별로(또는 보다 일반적으로 프레임 영역별로) 전송될 수 있다.

[0189] 값들이 LCU마다 변할 때, 값들이 인코더측 및 디코더측에서 동적으로 계산될 수 있다.

[0190] 가중 인자들의 값이 이하의 식에 따라 계산될 수 있고:

$$\beta_j = \frac{\sum_{i=0}^N \max(|c_i^Z - c_{in1}^Z|, |c_i^Z - c_{in2}^Z|)}{\sum_{i=0}^N \max(|c_i^X - c_{in1}^X|, |c_i^X - c_{in2}^X|)}$$

[0191] 여기서 c_i^Z 는 클래스 j의 픽셀 i의 성분 Z의 값이고, 여기서 c_{in1}^Z 및 c_{in2}^Z 는, 특히 부록의 표 2를 참조하여 앞서 기술한 바와 같이, 클래스 j의 픽셀 i의 이웃들의 성분 Z의 값들이다.

[0192] 그에 따라, 가중 인자 β_j 의 값은 분류된 픽셀들 및 그의 이웃 픽셀들의 성분 Z의 평균(최대) 차와 분류된 픽셀들 및 그의 이웃 픽셀들의 성분 X의 평균(최대) 차 사이의 비이다. 가중 인자 β_j 의(또는 β 의) 값들을 계산하기 위해 다른 관계식 또는 식이 사용될 수 있다.

[0193] 여전히 다른 특정의 실시예에서, 비트스트림에 인코딩되어 있는 오프셋들의 값이 가중 인자 β_j (또는 β)의 값에 의존할 수 있지만, 이들이 결정되어 있다. 이러한 실시예에서, 가중 인자 β 또는 β_j 의 값이 1 미만인 경우, 오프셋 $0x_j$ 의 값 대신에, 오프셋 $0z_j$ 의 값이 비트스트림에 인코딩된다.

[0194] 실제로, 식 $0z_j = \beta_j 0x_j$ 에서, $\beta_j < 1$ 인 경우, $0z_j < 0x_j$ 이다. 그 결과, 오프셋 값 $0z_j$ 는 $0x_j$ 의 값보다 비트스트림에 인코딩되기 위해 적은 수의 비트를 필요로 한다. 디코더가 가중 인자 β_j 의 값을 알고 있거나 결정할 수 있기 때문에, 디코더는 비트스트림으로부터 $0z_j$ 의 값을 디코딩할 수 있고 이하의 식에 따라 $0x_j$ 의 값을 결정할 수 있다:

$$0z_j = (1 / \beta_j) \times 0x_j$$

[0195] 여전히 특정의 실시예에서, 제2 성분 Z와 연관되어 있는 오프셋들 $0z_j$ 의 절대값은 제1 성분 X와 연관되어 있는 오프셋들 $0x(5-j)$ 의 절대값과 같다. 여기서 유의할 점은, 이 특정의 실시예에 따르면, 예지 오프셋 클래스의 오프셋의 부호가 아티팩트를 감소시키기 위해 사전 결정되어 있기 때문에 절대값만이 사용된다는 것이다.

[0196] 도 17은 한 성분과 연관되어 있는 오프셋들의 값들이 제2 성분과 연관되어 있는 오프셋들의 미러링된 값들에 대

응하는 이러한 특정의 실시예를 나타낸 것이다. 예시된 바와 같이, 오프셋들의 역순을 고려함으로써, 성분 Z의 4개의 클래스에 대한 오프셋이 성분 X로부터의 오프셋들로부터 도출된다. 여기서 상기할 점은, 오프셋들에 적용될 부호가 분류로부터 추론된다는 것이다.

[0200] 여전히 다른 특정의 실시예에서, 제2 성분과 연관되어 있는 오프셋들로서 사용되는 제1 성분과 연관되어 있는 오프셋들의 미러링된 값들이 가중 인자 β_j 로 가중될 수 있다. 이 특정의 실시예에 따르면, 제2 성분 Z와 연관되어 있는 오프셋들 $0z_j$ 의 절대값은 이하의 식에 따라 제1 성분 X와 연관되어 있는 오프셋들 $0x(5-j)$ 의 가중 절대값과 같다.

$$0z_j = \beta_j \ 0x(5-j)$$

[0202] 앞서 기술한 바와 같이, SAO 대역 오프셋 필터는 2가지 유형의 파라미터들을 필요로 한다. 제1 유형은 위치(sao_band_position)에 관한 것이고, 제2 유형은 주어진 위치로부터 정의되는 대역의 4개의 클래스에 관련된 4개의 오프셋에 관한 것이다. 본 발명의 실시예에 따르면, 프레임 영역의 성분들과 연관되어 있는 2가지 유형의 파라미터가 이미 처리된 다른 프레임 영역의 성분들과 연관되어 있는 대응하는 파라미터들로부터 개별적으로 획득될 수 있다. 그렇지만, 위치(sao_band_position)에 관련된 실시예들 및 오프셋에 관련된 실시예들이 쉽게 결합될 수 있다. 더욱이, 예지 오프셋 실시예들을 참조하여 앞서 기술한 바와 같이, 대역 오프셋 필터링을 위한 위치 및 오프셋과 관련되어 있는 이들 실시예들 또는 실시예들의 결합이, 앞서 기술한 바와 같이, 병합 플래그들의 값들 또는 APS에 대한 SAO 구문의 값들로 적용된다.

[0203] 유의할 점은, 한 성분과 연관되어 있는 SAO 파라미터들이 2개 또는 3개의 성분을 처리하는 데 사용될 수 있기 때문에, 이하의 설명이, 예시를 위해, 2개의 성분을 처리하는 데 하나의 성분과 연관되어 있는 SAO 파라미터들을 사용하는 것에 집중되어 있다는 것이다.

[0204] 도 18은, 대역 오프셋 분류를 사용하여 SAO 루프 필터를 적용할 때, 제2 성분 Z와 연관되어 있는 오프셋들의 값들이 제1 성분 X와 연관되어 있는 오프셋들의 값들로부터 어떻게 획득될 수 있는지를 나타낸 것이다.

[0205] 참조 번호(1801)는 현재 HEVC 규격에 따라 대역 오프셋 분류를 사용하여 SAO 루프 필터를 적용하기 위해 SAO 파라미터들(상세하게는, 성분들과 연관되어 있는 오프셋들)을 사용하는 것을 나타낸 것이다. 각각의 성분과 연관되어 있는 오프셋들 모두는 인코딩된 영상을 전송하는 데 사용되는 비트스트림에 인코딩되어 있다. 예시를 위해, 비트스트림은 제1 성분 X와 연관되어 있는 오프셋들(0x1 내지 0x4)(총칭하여, $0x_j$ 라고 지칭됨), 및 제2 성분 Z와 연관되어 있는 오프셋들(0z1 내지 0z4)(총칭하여, $0z_j$ 라고 지칭됨)을 포함한다. 참조 번호(1801)로 나타낸 바와 같이, 오프셋들 $0x_j$ 는 B0 X로 표시된 4개의 연속적인 클래스들을 포함하는 제1 대역에 대응하고, 오프셋들 $0z_j$ 는 B0 Z로 표시된 4개의 연속적인 클래스들을 역시 포함하는 제2 대역에 대응한다.

[0206] 참조 번호(1802)는 오프셋들 $0z_j$ 가 오프셋들 $0x_j$ 와 같은 제1 특정의 실시예를 나타낸다. 따라서, 오프셋들 $0z_j$ 는 비트스트림에 인코딩되지 않는다. 이들은 예지 오프셋 필터링을 참조하여 앞서 기술한 바와 같은 대응하는 병합 플래그들을 사용함으로써 복원될 수 있다. 이와 유사하게, 오프셋들 $0z_j$ 가 오프셋들 $0x_j$ 의 함수($0z_j = \beta_j \ 0x_j$)로서 결정될 수 있도록 가중 인자 β 또는 β_j 가 사용될 수 있다. 다시 말하지만, 가중 인자 β 또는 β_j 의 값 또는 값들이 비트스트림에서의 상이한 레벨들에서(시퀀스 레벨, 프레임 레벨, 슬라이스 레벨, 또는 LCU 레벨에서) 사전 결정되거나 표현될 수 있다. 이와 관련하여, 가중 인자 β 또는 β_j 의 값 또는 값들이 마이너스일 수 있다(예지 오프셋의 경우, 오프셋들의 부호가 클래스의 함수로서 결정됨). 상세하게는, 가중 인자가 마이너스 1일 수 있다($\beta = -1$).

[0207] 여전히 다른 특정의 실시예에서, 가중 인자 β 또는 β_j 의 값 또는 값들이 동적으로 결정될 수 있다. 예시를 위해, 가중 인자 β 또는 β_j 의 값 또는 값들이 대역의 위치(sao_band_position)의 함수로서 결정될 수 있다. 예를 들어, 대역 Z의 클래스 j 및 대역 X의 클래스 j가 구간(도 18에서 굵은 선으로 나타냄)의 중앙을 고려해서 동일한 하위 구간에 있지 않은 경우, $\beta = -1$ 이고, 그렇지 않은 경우 $\beta = 1$ 이다. 여기서, 전체 구간이 2개의 하위 구간으로 분할되어 있는 것이 생각되고 있으며, 여기서 (픽셀 값에 대해 8 비트 심도가 사용될 때) 제1 구간은 0부터 127까지의 픽셀 값들로 된 구간이고, 제2 구간은 128부터 256까지의 픽셀 값들로 된 구간이다.

[0208] 다른 실시예에 따르면, 가중 인자 β_j 의 값들은 성분 Z의 현재 클래스(또는 가중 인자 β 의 값에 대한 클래스 세트)와 구간의 중앙 간의 거리와 성분 X의 현재 클래스(또는 가중 인자 β 의 값에 대한 클래스 세트)와 구간의 중앙 간의 거리의 비의 함수로서 결정되고, 이는 다음과 같이 표현될 수 있고:

$$\beta_j = \frac{sao_band_position_Z_j - 16}{sao_band_position_X_j - 16}$$

[0209]

[0210] 여기서 $sao_band_position_Z_j - 16$ 은 현재 성분 Z와 구간의 중앙 간의 거리이고 $sao_band_position_X_j - 16$ 은 현재 성분 X와 구간의 중앙 간의 거리이다. 다른 실시예에서, 역비(inverse ratio)가 생각될 수 있다. 예지 오프셋에 대해서와 같이, 가중 인자 β 또는 β_j 의 값 또는 값들을 결정하기 위해 기타 관계식 및 식이 사용될 수 있다. 더욱이, 앞서 기술한 바와 같이, 오프셋들 $0x_j$ 또는 오프셋들 $0z_j$ 를 인코딩하는 데 필요한 비트 수를 감소시키기 위해, 가중 인자 β 또는 β_j 의 값 또는 값들(그렇지만, 이들이 결정되어 있음)에 의존하여 오프셋들 $0x_j$ 또는 오프셋들 $0z_j$ 가 비트스트림에 삽입될 수 있다.

[0211]

참조 번호(1803)는 제1 성분과 연관되어 있는 오프셋들의 미러링된 값들이 대역 오프셋 필터링을 위해 제2 성분과 연관되어 있는 오프셋들의 값들로서 어떻게 사용될 수 있는지를 나타내고 있다. 예지 오프셋 필터링을 참조하여 앞서 기술한 바와 같이, 오프셋들 $0z_j$ 와 오프셋들 $0x_j$ 사이의 관계식이 이하의 식으로 표현될 수 있다:

[0212]

$$0z_j = 0x(5-j)$$

[0213]

다시 말하지만, 이 특정의 실시예는 식 $0z_j = \beta_j 0x(5-j)$ 에 따라 가중 인자의 사용과 결합될 수 있다. 특정의 실시예에서, β_j 는 -1이다.

[0214]

대역 오프셋 필터링을 위한 오프셋들의 구체적인 도출은 픽셀 대역 오프셋 분류의 수정을 가능하게 하는 클래스의 크기의 증가 또는 감소이다. 참조 번호(1804)는 이러한 분류 적응(classification adaptation)의 한 예를 나타낸 것이다. 예시된 예에 따르면, 성분 X에 대한 대역 분류와 비교하여 성분 Z에 대해 클래당 픽셀의 수가 2로 나누어진다. 따라서, 대역 오프셋 필터링에 따른 성분 Z에 대한 픽셀 분류는 6 비트를 검사하는 것을 필요로 한다. 환연하면, 픽셀을 분류하기 위해 최상위 6 비트가 검사되어야만 한다. 따라서, 성분 Z에 대해 전체 픽셀 값 구간이 64개의 범위들로 나누어지는 것이 생각될 수 있다.

[0215]

이 관계가 구현예에 대해 고정되어 있을 수 있거나, 시퀀스, 프레임, 슬라이스 또는 LCU 레벨에서 신호될 수 있다. 이는 LCU 레벨(또는 시퀀스, 프레임 또는 슬라이스)에서 성분 X에 대한 픽셀 범위 크기와 성분 Z에 대한 픽셀 범위 크기의 비를 고려함으로써 동적으로 결정될 수 있다. 그에 따라, 성분 Z에 대한 클래스들의 크기가 다음과 같이 결정될 수 있고:

[0216]

모든 $i = 0$ 내지 N에 대해,

$$size(class_Z) = \frac{\max(P_{zi}) - \min(P_{zi})}{\max(P_{xi}) - \min(P_{xi})} size(class_X)$$

[0217]

[0218] 여기서 P_{zi} 는 성분 Z의 픽셀들이고 P_{xi} 는 성분 X의 픽셀들이다. 대역 분류를 간략화하기 위해, 클래스당 픽셀 값들의 수가 (최상위 비트들의 비트 검사를 가능하게 하는) 2의 몇수이다.

[0219]

이 특정의 실시예는 참조 번호(1805)로 예시되어 있는 바와 같이 대역 오프셋의 오프셋들의 도출에 대해 기술되어 있는 모든 다른 특정의 실시예들과 결합될 수 있고, 여기서 이 실시예는 성분 X와 연관되어 있는 미러링된 값 오프셋들의 사용과 결합된다.

[0220]

SAO 예지 및 대역 필터링 둘 다에 적용가능한 다른 특정의 실시예에서, 하나의 성분으로부터의 오프셋들이 2개 또는 3개의 성분으로부터 도출될 수 있다.

[0221]

1개 또는 2개의 다른 성분에 대한 오프셋들을 결정(또는 도출)하기 위해 하나의 성분으로부터의 오프셋들을 사용하는 것에 관련되어 있는 SAO 예지 및 대역 필터링 둘 다에 적용가능한 모든 기술된 실시예들은, 제2 성분 및 /또는 제3 성분의 오프셋들이 저장될 필요가 없기 때문에, 이러한 도출이 (모든 LCU들에 대해) 시스템적으로 사용될 때, 특히 요청된 메모리 버퍼의 크기에 관한 장점들을 제공한다.

[0222]

(SAO 예지 및 대역 필터링 둘 다에 대한) 다른 특정의 실시예에서, 제1 성분으로부터의 오프셋들에 따라 제2 성분으로부터의 오프셋들을 예측하기 위해 성분들 간의 오프셋 도출이 사용될 수 있다. (설명은 성분들 간에 오프셋들을 공유하는 것에 중점을 두고 있다.) 이 실시예에서, 성분 Z의 오프셋 값을 직접 인코딩하는 대신에, 성

분 Z의 오프셋들의 잔차들이 비트스트림에 인코딩된다. 디코더측에서, 성분 X의 오프셋들에 잔차 오프셋들을 가산함으로써 성분 Z의 오프셋들이 획득된다.

[0223] SAO 에지 및 대역 오프셋 필터링 둘 다에 적용가능한 다른 특정의 실시예에서, 제1 성분과 연관되어 있는 오프셋들에 따라 제2 성분과 연관되어 있는 오프셋들의 컨텍스트 산술 코딩(context arithmetic coding)에 대한 컨텍스트로서 성분들 간의 오프셋 도출이 사용될 수 있다.

[0224] 여전히 SAO 에지 및 대역 필터링 둘 다에 적용가능한 다른 특정의 실시예에서, 2개의 성분들 간의 오프셋 도출이 비트스트림에서 신호될 수 있고, 모든 성분들에 대해 오프셋들이 비트스트림에 인코딩되어 있는 종래의 모드와 경쟁할 수 있다.

[0225] 앞서 기술한 바와 같이, 대역 위치(sao_band_position)는 전체 픽셀 값 구간에서의 대역의 제1 클래스의 위치이다. 표준의 5 비트 검사를 고려함으로써, sao_band_position 파라미터의 값은 범위 [0; 32]에 속한다. 본 발명의 실시예에 따르면, 이 파라미터가 2개 또는 3개의 성분들 간에 공유될 수 있다. 예시를 위해, 설명은 성분 X와 연관되어 있는 그의 값으로부터 성분 Z에 관한 이 파라미터를 도출하는 것에 중점을 두고 있다.

[0226] 도 19는, 대역 오프셋 분류를 사용하여 SAO 루프 필터를 적용할 때, 제2 성분 Z와 연관되어 있는 sao_band_position 파라미터의 값이 제1 성분 X와 연관되어 있는 sao_band_position 파라미터의 값으로부터 어떻게 획득될 수 있는지를 나타낸 것이다.

[0227] 참조 번호(1901)는 현재 HEVC 규격에 따라 비트스트림에서 전송되는 2개의 sao_band_position 값의 한 예를 나타낸 것이다.

[0228] 참조 번호(1902)에 의해 예시된 특정의 실시예에서, 성분 Z 및 성분 X 둘 다에 대해 공통 sao_band_position 값이 사용된다.

[0229] 특정의 실시예들에서, 대역의 중앙 위치가 전체 픽셀 구간의 중앙과 비교된다. 얻어진 값은 sao_band_position - 14[14는 구간의 중앙(16) - 대역의 중앙(2)에 대응함]이다.

[0230] 그에 따라, 참조 번호(1903)에 의해 예시된 특정의 실시예에서, 성분 Z와 연관되어 있는 sao_band_position_Z는 전체 픽셀 구간의 중앙에 대해 성분 X와 연관되어 있는 sao_band_position_X와 대칭인 것으로 정의된다. 따라서, (sao_band_position_Z - 14)는 -(sao_band_position_X - 14)와 같은 것으로 간주된다. 이 예에서, sao_band_position_X가 6이기 때문에, sao_band_position_Z는 따라서 22이다[-(6-14)+14= 22].

[0231] 여전히 참조 번호(1904)에 의해 예시된 특정의 실시예에서, 성분 Z와 연관되어 있는 sao_band_position_Z 값을 성분 X와 연관되어 있는 sao_band_position_X 값 및 전체 픽셀 구간의 중앙의 위치의 함수로서 결정하기 위해 가중 인자 β 가 사용된다. 이것은, 이전의 실시예에서와 같이, 전체 픽셀 구간의 중앙과 각각의 성분 X 및 Z의 sao_band_position 값 간의 상대 위치에 기초한다. 이 예에 따르면, 사용될 관계식은 다음과 같을 수 있다:

$$\text{sao_band_position}_Z - 14 = \beta (\text{sao_band_position}_X - 14)$$

[0233] 즉,

$$\text{sao_band_position}_Z = \beta (\text{sao_band_position}_X - 14) + 14$$

[0235] 도 19에 주어진 예에서, $\beta = -1/2$. 그 경우에, sao_band_position_X는 6이고, 그 결과 sao_band_position_Z는 18이다[-1/2(6-14)+14=18].

[0236] 가중 인자 β 의 값이 구현예에 대해 사전 결정되거나 임의의 레벨(시퀀스, 프레임, 슬라이스 또는 LCU)에서 전송될 수 있다.

[0237] 여전히 SAO 에지 및 대역 필터링 둘 다에 적용가능한 다른 특정의 실시예에서, 2개의 성분들 간의 sao_band_position 파라미터의 도출이 비트스트림에서 신호될 수 있고, 모든 성분들에 대해 sao_band_position 파라미터 값이 비트스트림에 인코딩되어 있는 종래의 모드와 경쟁할 수 있다.

[0238] SAO 파라미터들이 2개 또는 3개의 색상 성분들 간에 공유될 때의 SAO 파라미터들의 인코딩 선택은 바람직하게는 코딩 효율에 관련되어 있다. 환연하면, 공유되는 SAO 파라미터들을 선택하는 것은 유리하게도 이들 파라미터 및 연관된 값들의 코딩 효율을 고려한다. 여기서 상기할 점은, 레이트 왜곡 비용의 면에서의 최상의 오프셋의 선택이 에지 및 대역 오프셋 필터링 둘 다에 대해 동일하다는 것이다.

[0239] 어느 SAO 유형(에지 또는 대역)이 사용되는 간에, 이들 오프셋이 2개의 성분 Z와 X 간에 공유될 때의 오프셋들

의 인코딩 선택은 대역 중앙 선택 이전에 이하에서 본 명세서에 기술되어 있다.

[0240] 하나의 성분에 대한 왜곡의 면에서 최적의 오프셋을 결정하는 인코딩 프로세스를 나타낸 도 3을 참조하여 기술한 바와 같이, 성분 X에 관련되어 있는 $SumX_j$ 및 $SumNbPixX_j$ 값 및 성분 Z에 관련되어 있는 $SumZ_j$ 및 $SumNbPixZ_j$ 값이 모든 클래스 j에 대해 알려져 있다. $SumX_j$ 및 $SumZ_j$ 는, 각각, 성분 X 및 Z에 대해, 클래스 j에서의 픽셀 값들과 그의 원래의 값들 간의 차의 합이다. $SumNbPixX_j$ 및 $SumNbPixZ_j$ 는, 각각, X 및 Z 성분에 대해, 클래스 j에서의 픽셀들의 수이다. 이들 4개의 파라미터의 값들이 모든 클래스들에 대해 이미 계산되어 있는 것으로 가정된다.

[0241] 목적이 $Oz_j = Ox_j$ 를 얻는 것인 특정의 실시예에서, 최적의 오프셋 $Oopt_j$ 가 이하의 식에 의해 결정될 수 있다:

$$Oopt_j = Oz_j = Ox_j = (SumX_j + SumZ_j) / (SumNbPixX_j + SumNbPixZ_j)$$

[0243] 그 다음에, 앞서 기술한 바와 같이, 각각의 성분을 독립적으로 처리하기 위해, 인코더는 레이트 왜곡 기준의 면에서 최상의 오프셋 Ox_j 를 결정한다.

[0244] $Oz_j = Ox_j$ 인 특정의 실시예에서, 이것은 다음과 같이 레이트 왜곡 비용을 최소화하는 것이고:

$$J(Ox_j, Oz_j) = SumNbPixX_j \times (Ox_j \times Ox_j) - SumX_j \times Ox_j \times 2 + \\ SumNbPixZ(j) \times (Ox_j \times Ox_j) - SumZ(j) \times Ox_j \times 2 + \lambda R(Ox_j)$$

[0246] 이는 이하의 표현식으로 간략화될 수 있다:

$$J(Ox_j, Oz_j) = (SumNbPixX_j + SumNbPixZ_j) \times (Ox_j \times Ox_j) - (SumX_j + \\ SumZ_j) \times Ox_j \times 2 + \lambda R(Ox_j)$$

[0248] 미러링된 값들이 예지 오프셋 경우에 대해 오프셋들 [$Oz_j = -Ox(5-j)$]을 결정하는 것을 목표로 하는 특정의 실시예에서, 최적의 오프셋이 이하의 식에 의해 정의될 수 있다:

$$Oopt_j = Oz(5-j) = Ox_j = (SumX_j - SumZ(5-j)) / (SumNbPixX_j + \\ SumNbPixZ(5-j))$$

[0250] 그 다음에, 인코더는 레이트 왜곡 기준의 면에서 최상의 오프셋 Ox_j 를 결정한다. $Oz(5-j) = -Ox_j$ 인 특정의 실시예에 따르면, 이것은 다음과 같이 레이트 왜곡 비용을 최소화하는 것이다:

$$J(Ox_j, Oz_j) = SumNbPixX_j \times (Ox_j \times Ox_j) - SumX_j \times Ox_j \times 2 + \\ SumNbPixZ(5-j) \times (Oz(5-j) \times Oz(5-j)) - SumZ(5-j) \times Oz(5-j) \times 2 + \lambda R(Ox_j)$$

[0252] 또는

$$J(Ox_j, Oz_j) = SumNbPixX_j \times (Ox_j \times Ox_j) - SumX_j \times Ox_j \times 2 + \\ SumNbPixZ(5-j) \times (Ox_j \times Ox_j) + SumZ(5-j) \times Ox_j \times 2 + \lambda R(Ox_j)$$

[0254] $Oz_j = -Ox_j$ 인 특정의 실시예에서, 최적의 오프셋 $Oopt_j$ 는 이하의 식에 의해 결정된다:

$$Oopt_j = (SumX_j - SumZ_j) / (SumNbPixX_j + SumNbPixZ_j)$$

[0256] 그 다음에, 인코더는 레이트 왜곡 기준의 면에서 최상의 오프셋 Ox_j 를 결정한다. $Oz_j = -Ox_j$ 인 특정의 실시예에 따르면, 이것은 다음과 같이 레이트 왜곡 비용을 최소화하는 것이고:

$$J(Ox_j, Oz_j) = SumNbPixX_j \times (Ox_j \times Ox_j) - SumX_j \times Ox_j \times 2 + \\ SumNbPixZ(j) \times (Ox(j) \times Ox(j)) + SumZ(j) \times Ox(j) \times 2 + \lambda R(Ox_j),$$

[0258] 이는 이하의 표현식으로 간략화될 수 있다:

$$J(Oxj, Ozj) = (SumNbPixXj + SumNbPixZj) \times (Oxj \times Ozj) - (SumXj - SumZj) \times Ozj \times 2 + \lambda R(Oxj)$$

[0259] $Ozj = \beta_j Oxj$ 인 특정의 실시예에서, 인코더는 성분 X에 대한 최적의 $OoptXj$ 값 및 성분 Z에 대한 최적의 $OoptZj$ 값을 독립적으로 결정한다. 그 다음에, 인코더는 레이트 왜곡 기준의 면에서 최상의 오프셋 Oxj 를 결정한다. $Ozj = \beta_j Oxj$ 인 특정의 실시예에 따르면, 이것은 도 4를 참조하여 기술된 바와 같이, 레이트 왜곡 비용을 최소화하는 것이다.

[0260] 도 20은 레이트 왜곡 기준에 따라 2개 또는 3개의 성분에 의해 사용될 최적의 오프셋을 결정하는 방법의 단계들을 나타낸 플로우차트이다. 도 20에 예시되어 있는 프로세스는 도 4를 참조하여 기술된 것에 기초하고 있다. 단계들(2001, 2004, 2005, 및 2006)은, 각각, 단계들(401, 404, 405, 및 406)과 유사하다.

[0261] 단계(2002)는 단계(2002)에서 시작되는 루프가 오프셋 Oxj 를 $sign(OoptXj) \times \max(f \times OoptXj, OoptZj)$ (f 는 $\beta_j < 1$ 인 경우 $1/\beta_j$ 이고, 그렇지 않은 경우 β_j 임)로부터 0으로 변하게 한다는 점에서 단계(402)와 상이하다.

[0262] 이는 또한 다음과 같이 쓰여질 수 있다:

$$J(Oxj, Ozj) = SumNbPixXj \times (Oxj \times Ozj) - SumXj \times Ozj \times 2 +$$

$$SumNbPixZ(j) \times (\beta_j \times Ozj \times \beta_j \times Ozj) - SumZ(j) \times \beta_j \times Ozj \times 2 + \lambda R(Oxj)$$

[0263] 이는 또한 다음과 같이 쓰여질 수 있다:

$$J(Oxj, Ozj) = (SumNbPixXj + SumNbPixZ(j) \times (\beta_j) \times (\beta_j)) \times (Oxj \times Ozj)$$

$$- (SumXj + SumZ(j) \times \beta_j) \times Ozj \times 2 + \lambda R(Oxj)$$

[0264] 단계(2003)에서, 예시를 위해, $\beta_j = -1$ 인 특정의 경우가 생각되고 있다.

[0265] 2개의 성분 X와 Z 간에 공유되는 sao_band_position 값의 선택이 이하의 관계식에 따라 결정될 수 있다:

$$sao_band_position_Z = \beta (sao_band_position_X - 14) + 14$$

[0266] 도 21은 2개의 성분 간에 공유되는 SAO 대역 위치를 결정하는 방법의 단계들을 나타낸 플로우차트이다. 이는 도 5를 참조하여 기술된 프로세스에 기초하고 있다. 도 21에 예시되어 있는 프로세스는 성분 X와 연관되어 있는 sao_band_position_X의 값으로부터 성분 Z와 연관되어 있는 sao_band_position_Z의 값을 결정하는 것을 목표로 한다.

[0267] 도 21에 예시되어 있는 프로세스와 도 5에 예시되어 있는 프로세스 간의 주된 차이점은 성분 Z와 연관되어 있는 클래스들의 레이트 왜곡 값들 및 성분 X와 연관되어 있는 클래스들의 레이트 왜곡 값들 둘 다가 이하의 식에 따라 서로 가산되는 모듈(2105)에 관련되어 있고:

$$J'i = J'i + JZj + JXjx$$

[0268] 여기서 JZj 는 성분 Z의 클래스 j에 대한 레이트 왜곡 값이고, $JXjx$ 는 성분 X의 클래스 j에 대한 레이트 왜곡 값이다. 더욱이, sao_band_position_Z와 sao_band_position_X를 결합하여 선택하기 위해, 인덱스 jx 가 이하의 식에 따라 도출된다:

$$sao_band_position_Z = \beta (sao_band_position_X - 14) + 14$$

[0269] 그에 따라, jx 는 $\beta (j - 14) + 14$ 이다.

[0270] 디코딩 복잡도를 감소시키기 위해 성분 분류를 회피하는 제2 주된 실시예에 따르면, SAO 클래스에서의 픽셀들이 단일 성분의 함수로서 결정된다. 픽셀 위치들이 이어서 필터링될 필요가 있는 성분 또는 성분들을 결정하는 데 사용된다. 그에 따라, 픽셀 성분이 그 성분에 따라 필터링되어야 하는 경우, 그 픽셀의 다른 성분 또는 2개의 다른 성분이 필터링된다.

[0271] 도 22는 본 발명의 특정의 실시예에 따른, 비트스트림을 디코딩하는 방법의 단계들을 나타낸 플로우차트이다.

이 도면에 예시되어 있는 프로세스는 도 6에 예시되어 있는 것에 기초하고 있다. 이는 성분 Z를 필터링하기 위해 성분 X의 분류를 사용하는 것을 목표로 한다.

[0278] 초기 단계(2201)는 도 12, 도 13 및 도 14를 참조하여 기술된 프로세스들에 따라 SAO 파라미터들을 결정하는 것이다. 그렇지만, 도 22에 예시된 알고리즘에 따르면, sao_type_idx 파라미터의 값, 그리고, 이 값이 5인 경우, 성분 X에 대한 sao_band_position 파라미터의 값만이 단계(2202)에서 비트스트림(도시 생략)으로부터 결정된다. 실제로, 이들 파라미터가 분류에 관련되어 있기 때문에, 이들이 성분 X의 분류에 따라 필터링되는 성분 Z를 분류할 필요는 없다.

[0279] 그 다음에, 단계(2203)에서, 성분 X와 연관되어 있는 4개의 오프셋이 비트스트림으로부터 획득된다. 이와 유사하게, 단계(2204)에서 성분 Z와 연관되어 있는 4개의 오프셋이 비트스트림으로부터 획득된다. 유의할 점은, 앞서 기술한 바와 같이, 이들 오프셋이 다른 성분들 및/또는 다른 LCU와 연관되어 있는 다른 오프셋들의 함수로서 결정될 수 있다는 것이다.

[0280] 단계(2205)에서, 카운터 변수 i는 값 0으로 설정된다. 그 다음에, N개의 픽셀을 포함하는 성분 X의 프레임 영역[참조 번호(2206)]의 제1 픽셀 P_{X_i} 가 단계(2207)에서 추출된다. 이와 유사하게, 성분 Z의 프레임 영역[참조 번호(2208)]의 제1 픽셀 P_{Z_i} 가 단계(2209)에서 추출된다. 예시를 위해, 여기서 성분 Z의 프레임 영역이 또한 N개의 픽셀을 포함하는(성분 X 및 Z가 프레임 영역당 동일한 양의 픽셀을 가지는) 것으로 한다.

[0281] 그 다음에, 단계(2210)에서, 제1 픽셀 P_{X_i} 의 성분 X가, 부록의 표 2 및 도 2를, 각각, 참조하여 앞서 기술된 바와 같이, sao_type_idx_X 파라미터의 값(그리고, 필요한 경우, sao_band_position_X 파라미터의 값)에 따라 예지 오프셋 분류 또는 대역 오프셋 분류에 따라 분류된다.

[0282] 단계(2211)에서, 픽셀 P_{X_i} 의 성분 X가 필터링되어야 하는지 여부(즉, P_{X_i} 가 SAO 클래스에 속하는지)를 판정하기 위해 테스트가 수행된다. P_{X_i} 가 SAO 클래스에 속하는 경우, 관련 클래스 번호 j가 식별되고, 단계(2212)에서 대응하는 오프셋 값이 성분 X와 연관되어 있는 오프셋 테이블로부터 획득된다. 그 다음에, 단계(2213)에서, 픽셀 P_{X_i} 의 성분 X가 획득된 오프셋을 그의 값에 가산함으로써 필터링된다. 픽셀 P_{X_i} 의 필터링된 성분 X는 P'_{X_i} [참조 번호(2214)]라고 한다. 동시에 또는 순차적으로, 성분 Z와 연관되어 있고 클래스 번호 j에 대응하는 오프셋 값이 단계(2215)에서 성분 Z와 연관되어 있는 오프셋 테이블로부터 획득된다. 그 다음에, 단계(2216)에서, 픽셀 P_{Z_i} 의 성분 Z가 획득된 오프셋을 그의 값에 가산함으로써 필터링된다. 픽셀 P_{Z_i} 의 필터링된 성분 Z는 P'_{Z_i} [참조 번호(2217)]라고 한다.

[0283] 그 다음에, 단계(2218)에서, 필터링된 성분 P'_{X_i} 및 P'_{Z_j} 가 필터링된 프레임 영역(각각, 2219 및 2220이라고 함)의 성분 X 및 성분 Z에 넣어진다. 단계(2218)에서, 필터링된 픽셀 P'_{X_i} 및 P'_{Z_j} 는, 각각, 2219 및 2220이라고 하는 처리된 프레임들의 픽셀 P_{X_i} 및 P_{Z_j} 를 대체한다.

[0284] P_{X_i} 가 SAO 클래스에 속하지 않는 경우, 필터링된 프레임 영역의 성분 X 및 성분 Z에서의 성분들 P_{X_i} 및 P_{Z_j} 가 단계들(2221, 2222, 및 2218)에서 그대로 있다.

[0285] 그 다음에, 단계(2223)에서, 현재 프레임 영역의 그 다음 픽셀들에 필터 프로세스를 마찬가지로 적용하기 위해 카운터 변수 i가 1만큼 증가된다.

[0286] 단계(2224)에서, 현재 프레임 영역의 N개의 픽셀들 모두가 처리되었는지($i \geq N$ 인지) 여부가 판정된다. 예인 경우, 프레임 영역의 필터링된 성분들 X 및 Z가 재구성되고, SAO 재구성된 프레임에 가산될 수 있다.

[0287] 도 22는 양 성분이 현재 프레임 영역에 대해 동일한 수의 픽셀들을 가질 때 다른 성분 Z를 유사하게 필터링하기 위해 하나의 성분의 SAO 분류를 사용하는 것을 나타내고 있다.

[0288] 그렇지만, 프레임 영역의 성분들이 항상 동일한 수의 픽셀들을 가지는 것은 아니다. N_z 로 표시된 성분 Z에 대한 픽셀 수에 관해 N_x 로 표시된 성분 X에 대한 상대 픽셀 수에 따라, 2개의 경우가 생각될 수 있다. 프레임 영역 내의 2개의 상이한 성분들의 픽셀 수들 간의 차가 통상적으로 2의 배수이기 때문에, 픽셀 성분 수들 간의 비정수 관계를 고려하는 것이 일반적으로 필요하지 않다.

[0289] 성분 Z의 픽셀들의 수가 성분 X의 픽셀들의 수보다 큰 경우($N_x < N_z$)를 고려하여, P_{Z_i} 가 N_z/N_x 개의 픽셀들로 된 그룹인 것으로 생각함으로써 도 22를 참조하여 기술된 알고리즘이 쉽게 적용될 수 있다. 그에 따라, 오프셋 $Offset_{Z_i}$ 가 픽셀 그룹 P_{Z_i} 에 가산될 때, 이 오프셋은 픽셀 그룹 P_{Z_i} 의 모든 픽셀들에 적용된다. 더욱이, 특정의 실시예에서, 오프셋 $Offset_{Z_i}$ 가 픽셀 그룹 P_{Z_i} 와 동일한 크기의 오프셋 그룹인 것이 생각될 수 있다.

[0290] 성분 X의 픽셀들의 수가 성분 Z의 픽셀들의 수보다 큰 경우($N_x > N_z$)를 고려하여, 몇개의 실시예들이 생각될 수 있다. 이러한 경우에, 도 22를 참조하여 기술된 단계들(2216 및 2218)은 성분 X의 픽셀들의 수와 성분 Z의 픽셀들의 수 사이의 관계를 고려할 필요가 있다. 도 22에서, P_{zi} 가 $P_z(i/N_x)$ 에 의해 변경된다.

[0291] 제1 특정의 실시예에서, P_{xi} 가 적어도 한번 필터링된 경우에만 오프셋 $Offset_{zi}$ 가 그룹 $P_z(i/N_x)$ 에 가산된다.

[0292] 다른 특정의 실시예에 따르면, 도 22의 단계(2218)는 P_{xi} 가 필터링된 횟수를 고려한다. 횟수가 $2xN_x/N_z$ 초과인 경우, $Offset_{zi}$ 가 $P_z(i/N_x)$ 에 가산된다.

[0293] 여전히 특정의 실시예에 따르면, 단계(2218)는 $P_z(i/N_x)$ 또는 $P'z(i/N_x)$ 를 N_x/N_z 번 수신하고, 성분 Z에 대한 필터링된 프레임 영역에 배치되는 픽셀 값을 생성하기 위해 평균을 계산한다(2220).

[0294] 도 23은 하나의 성분 X의 SAO 분류를 다른 성분 Z를 유사하게 필터링하기 위해 사용하는 것을 나타낸 것이다.

[0295] 예시를 위해, 블록(2301)의 회색 픽셀들은 현재 구현 HEVC에 따라 성분 X에 대한 현재 클래스 j의 유형 i의 SAO 루프 필터에 대해 필터링된 픽셀들을 나타낸다. 이와 유사하게, 블록(2302)의 흑색 픽셀들은 현재 HEVC 규격에 따라 성분 Z에 대한 현재 클래스 j의 유형 i의 SAO 루프 필터에 대해 필터링된 픽셀들을 나타낸다.

[0296] 따라서, 도 23에 예시된 바와 같이, 현재 HEVC 규격을 고려하면 성분 X에서의 회색 픽셀들 및 성분 Z에서의 흑색 픽셀들이 항상 동일한 위치에 있는 것은 아니다.

[0297] 여전히 예시를 위해, 블록(2303)의 회색 픽셀들은 현재 구현 HEVC에 따라 성분 X에 대한 현재 클래스 j의 유형 i의 SAO 루프 필터에 대해 필터링된 픽셀들을 나타낸다. 그렇지만, 블록(2304)의 흑색 픽셀들은 성분 X의 분류에 따라 성분 Z에 대해 SAO 루프 필터에 대해 필터링된 픽셀들을 나타낸다[블록(2303)]. 그에 따라, 회색 픽셀들 및 흑색 픽셀들이 동일한 위치에 있다. 이 결과 양 성분에 대해 단지 하나의 분류를 사용한다.

[0298] 유의할 점은, 이들 주요 실시예 둘 다가 코딩 효율 및 디코딩 복잡도를 최적화하기 위해 결합될 수 있다는 것이다.

[0299] 본 발명이 특정의 실시예들을 참조하여 이상에서 기술되어 있지만, 본 발명이 특정의 실시예들로 제한되지 않고, 본 발명의 범위 내에 속하는 수정들이 기술 분야의 당업자에게 명백할 것이다. 상세하게는, 이전의 설명이, 예시를 위해, YUV 성분들을 참조하고 있지만, RGB(적색, 녹색 및 청색 성분)와 같은 기타 방식들이 사용될 수 있다.

[0300] 예를 들어, 이전의 실시예들이 영상의 픽셀들 및 그의 대응하는 픽셀 값을 관계하여 기술되어 있지만, 본 발명과 관련하여, 픽셀 그룹이 대응하는 그룹 픽셀 값을 함께 고려될 수 있다는 것을 잘 알 것이다. 샘플은 따라서 영상의 하나 이상의 픽셀에 대응할 수 있다.

[0301] 본 발명의 추가의 측면들이 이하에 기재되어 있다.

[0302] 본 발명의 제1 추가의 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 인코딩하는 방법이 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 방법은,

[0303] 적어도 하나의 인코딩된 샘플을 제공하기 위해 영상의 적어도 하나의 샘플의 제1 및 제2 성분을 인코딩하는 단계;

[0304] 적어도 하나의 인코딩된 샘플을 디코딩하는 단계;

[0305] 적어도 하나의 필터링 파라미터를 영상의 적어도 하나의 샘플의 적어도 하나의 성분과 적어도 하나의 대응하는 디코딩된 샘플의 적어도 하나의 대응하는 성분 간의 차의 함수로서 계산하는 단계 - 적어도 하나의 필터링 파라미터는 루프 필터에서 디코딩된 샘플의 적어도 제1 및 제2 성분을 독립적으로 필터링하는 데 사용되기 위해 계산됨 - 를 포함한다.

[0306] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0307] 일 실시예에서, 적어도 하나의 필터링 파라미터를 코딩하는 효율이 최적이도록 적어도 하나의 필터링 파라미터가 복수의 추정된 필터링 파라미터들의 함수로서 결정된다.

[0308] 일 실시예에서, 적어도 하나의 필터링 파라미터가 영상의 일부의 샘플들의 적어도 하나의 성분과 대응하는 디코

딩된 샘플들의 적어도 하나의 대응하는 성분 간의 차, 영상의 일부에서의 적어도 하나의 성분의 샘플들의 수, 및 영상의 일부의 샘플들의 2개의 성분의 값의 함수로서 결정된다.

[0309] 일 실시예에서, 적어도 하나의 필터링 파라미터가 영상의 적어도 하나의 샘플의 제1 성분과 적어도 하나의 대응하는 디코딩된 샘플의 대응하는 제1 성분 간의 차의 함수로서 계산된 제1 필터링 파라미터 및 영상의 적어도 하나의 샘플의 제2 성분과 적어도 하나의 대응하는 디코딩된 샘플의 대응하는 제2 성분 간의 차의 함수로서 계산된 제2 필터링 파라미터의 함수로서 결정된다.

[0310] 일 실시예에서, 제1 및 제2 필터링 파라미터는 영상의 일부의 복수의 샘플들의 함수로서 계산된다.

[0311] 일 실시예에서, 적어도 하나의 필터링 파라미터는 필터링될 샘플 성분에 가산될 적어도 하나의 오프셋을 포함한다.

[0312] 일 실시예에서, 적어도 하나의 필터링 파라미터는 적어도 하나의 오프셋을 가중하기 위한 적어도 하나의 가중인자를 추가로 포함한다.

[0313] 일 실시예에서, 적어도 하나의 필터링 파라미터는 필터링될 적어도 하나의 샘플 성분 클래스를 정의하는 대역의 적어도 하나의 위치를 추가로 포함한다.

[0314] 일 실시예에서, 이 방법은 적어도 하나의 위치를 상이한 성분들에 대해 추정된 상이한 레이트 왜곡 값들의 함수로서 결정하는 단계를 추가로 포함한다.

[0315] 여전히 일 실시예에서, 적어도 하나의 필터링 파라미터는 제1 샘플의 적어도 제1 성분과 연관되어 있는 적어도 하나의 플래그를 추가로 포함하고, 적어도 하나의 플래그는 적어도 제1 성분을 필터링하기 위한, 제2 샘플의 적어도 제2 성분과 연관되어 있는 적어도 하나의 필터링 파라미터에 대한 참조를 나타낸다.

[0316] 일 실시예에서, 적어도 하나의 필터링 파라미터는 제1 샘플의 적어도 제1 성분과 연관되어 있는 적어도 하나의 플래그를 추가로 포함하고, 적어도 하나의 플래그는 제1 샘플의 복수의 성분들을 필터링하기 위한, 제2 샘플의 적어도 제2 성분과 연관되어 있는 적어도 하나의 필터링 파라미터에 대한 참조를 나타낸다.

[0317] 본 발명의 제2 추가의 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 디코딩하는 방법이 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 방법은,

[0318] 적어도 하나의 재구성된 샘플의 제1 및 제2 성분을 제공하기 위해 적어도 하나의 인코딩된 샘플의 제1 및 제2 성분을 디코딩하는 단계;

[0319] 적어도 하나의 필터링 파라미터를 디코딩하는 단계; 및

[0320] 루프 필터에서 적어도 하나의 재구성된 샘플을 필터링하는 단계 - 적어도 하나의 재구성된 샘플의 제1 성분 및 제2 성분의 필터링은 디코딩된 적어도 하나의 필터링 파라미터의 적어도 하나의 공통 필터링 파라미터에 기초함 - 를 포함한다.

[0321] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0322] 일 실시예에서, 이 방법은 적어도 하나의 필터링 파라미터에 대한 참조를 나타내는 적어도 하나의 플래그를 획득하는 단계를 추가로 포함한다.

[0323] 일 실시예에서, 이 방법은 획득된 적어도 하나의 플래그를 분석하고, 이 분석에 응답하여, 적어도 하나 공통 필터링 파라미터를 획득하는 단계를 추가로 포함한다.

[0324] 여전히 일 실시예에서, 이 방법은 적어도 하나의 필터링 파라미터에 대한 참조를 나타내는 적어도 다른 플래그를 획득하는 단계 및 획득된 적어도 다른 플래그를 분석하는 단계를 포함하고, 적어도 하나의 플래그를 획득하는 단계는 획득된 적어도 다른 플래그를 분석하는 단계에 응답하여 수행된다.

[0325] 일 실시예에서, 이 방법은 적어도 하나의 재구성된 샘플의 적어도 제1 성분을 필터링하는 데 사용될 필터링 파라미터를 디코딩된 적어도 하나의 필터링 파라미터의 함수로서 계산하는 단계를 추가로 포함한다.

[0326] 일 실시예에서, 이 방법은 적어도 하나의 재구성된 샘플의 제1 및 제2 성분을 필터링하는 데 사용될 필터링 파라미터를 디코딩된 적어도 하나의 필터링 파라미터의 함수로서 계산하는 단계를 추가로 포함한다.

[0327] 여전히 일 실시예에서, 이 방법은 적어도 하나의 가중 인자를 획득하는 단계를 추가로 포함하고, 계산된 필터링 파라미터는 획득된 적어도 하나의 가중 인자에 기초한다.

[0328] 일 실시예에서, 이 방법은 적어도 하나의 가중 인자를 디코딩하는 단계를 추가로 포함하고, 적어도 하나의 가중 인자는 디코딩될 영상의 일부와 함께 수신된다.

[0329] 일 실시예에서, 공통 필터링 파라미터는 적어도 하나의 재구성된 샘플의 적어도 제1 성분에 가산될 적어도 하나의 오프셋 값을 포함한다.

[0330] 일 실시예에서, 공통 필터링 파라미터는 필터링될 적어도 하나의 샘플 성분 클래스를 정의하는 대역의 적어도 하나의 위치를 포함한다.

[0331] 여전히 일 실시예에서, 이 방법은 적어도 하나의 재구성된 샘플의 적어도 제1 성분을 필터링하기 위해 대역의 적어도 하나의 위치에 적용될 적어도 하나의 가중 인자를 획득하는 단계를 추가로 포함한다.

[0332] 본 발명의 제3 추가의 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 인코딩하는 방법이 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 방법은,

[0333] 적어도 하나의 인코딩된 샘플을 제공하기 위해 영상의 적어도 하나의 샘플의 제1 및 제2 성분을 인코딩하는 단계;

[0334] 적어도 하나의 인코딩된 샘플을 디코딩하는 단계; 및

[0335] 적어도 하나의 필터링 파라미터를 영상의 적어도 하나의 샘플의 성분과 적어도 하나의 디코딩된 샘플의 대응하는 성분 간의 차의 합수로서 계산하는 단계 - 적어도 하나의 필터링 파라미터는 루프 필터에서 디코딩된 샘플의 적어도 제1 및 제2 성분을 독립적으로 필터링하는 데 사용되기 위해 계산됨 - 를 포함한다.

[0336] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0337] 본 발명의 제4 추가의 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 디코딩하는 방법이 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 방법은,

[0338] 제1 재구성된 샘플의 제1 성분 및 제2 재구성된 샘플의 제2 성분을 제공하기 위해 제1 인코딩된 샘플의 제1 성분 및 제2 인코딩된 샘플의 제2 성분을 디코딩하는 단계;

[0339] 적어도 하나의 필터링 파라미터를 디코딩하는 단계; 및

[0340] 루프 필터에서 재구성된 샘플들을 필터링하는 단계 - 제1 재구성된 샘플의 제1 성분 및 제2 재구성된 샘플의 제2 성분의 필터링은 디코딩된 적어도 하나의 필터링 파라미터의 적어도 하나의 공통 필터링 파라미터에 기초함 - 를 포함한다.

[0341] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0342] 일 실시예에서, 이 방법은 적어도 하나의 재구성된 샘플의 제1 성분을 필터링하는 데 사용될 필터링 파라미터를 적어도 하나의 재구성된 샘플의 제2 성분을 필터링하는 데 사용될 필터링 파라미터의 함수로서 계산하는 단계를 추가로 포함한다.

[0343] 일 실시예에서, 제1 복수의 필터링 파라미터들은 제1 재구성된 샘플의 제1 성분과 연관되어 있고, 제2 복수의 필터링 파라미터들은 제2 재구성된 샘플의 제2 성분과 연관되어 있으며, 제1 복수의 필터링 파라미터들의 필터링 파라미터 값들은 제2 복수의 필터링 파라미터들의 필터링 파라미터 값들과 같고, 제1 복수의 필터링 파라미터들의 필터링 파라미터 값들은 제2 복수의 필터링 파라미터들의 필터링 파라미터 값들의 순서와 역순으로 되어 있다.

[0344] 다른 실시예에서, 제1 복수의 필터링 파라미터들은 제1 재구성된 샘플의 제1 성분과 연관되어 있고, 제2 복수의 필터링 파라미터들은 제2 재구성된 샘플의 제2 성분과 연관되어 있으며, 제1 복수의 필터링 파라미터들의 필터링 파라미터 값들은 제2 복수의 필터링 파라미터들의 필터링 파라미터 값들과 사전 결정된 가중 인자를 곱한 것

과 같다.

[0345] 본 발명의 제5 추가의 측면에 따르면, 프로그램가능 장치에 대한 컴퓨터 프로그램 제품이 제공되고, 컴퓨터 프로그램 제품은 프로그램가능 장치에 로드되어 실행될 때 앞서 기술한 제1 내지 제4 측면 중 하나를 실시하는 방법의 각각의 단계를 구현하는 명령어 시퀀스를 포함한다.

[0346] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0347] 본 발명의 제6 추가의 측면에 따르면, 앞서 기술한 제1 내지 제4 측면 중 하나를 실시하는 방법의 각각의 단계를 구현하는 컴퓨터 프로그램의 명령어들을 저장하는 컴퓨터 판독가능 저장 매체가 제공된다.

[0348] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0349] 본 발명의 제7 추가의 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 인코딩하는 인코딩 장치가 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 인코딩 장치는,

[0350] 적어도 하나의 인코딩된 샘플을 제공하기 위해 영상의 적어도 하나의 샘플의 제1 및 제2 성분을 인코딩하는 수단;

[0351] 적어도 하나의 인코딩된 샘플을 디코딩하는 수단; 및

[0352] 적어도 하나의 필터링 파라미터를 영상의 적어도 하나의 샘플의 적어도 하나의 성분과 적어도 하나의 대응하는 디코딩된 샘플의 적어도 하나의 대응하는 성분 간의 차의 함수로서 계산하는 수단 - 적어도 하나의 필터링 파라미터는 루프 필터에서 디코딩된 샘플의 적어도 제1 및 제2 성분을 독립적으로 필터링하는 데 사용되기 위해 계산됨 - 을 포함한다.

[0353] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0354] 일 실시예에서, 이 장치는 플래그들을 인코딩하는 수단을 추가로 포함하고, 적어도 하나의 필터링 파라미터는 제1 샘플의 적어도 제1 성분과 연관되어 있는 적어도 하나의 플래그를 추가로 포함하며, 적어도 하나의 플래그는 적어도 제1 성분을 필터링하기 위한, 제2 샘플의 적어도 제2 성분과 연관되어 있는 적어도 하나의 필터링 파라미터에 대한 참조를 나타낸다.

[0355] 다른 실시예에서, 이 장치는 플래그들을 인코딩하는 수단을 추가로 포함하고, 적어도 하나의 필터링 파라미터는 제1 샘플의 적어도 제1 성분과 연관되어 있는 적어도 하나의 플래그를 추가로 포함하며, 적어도 하나의 플래그는 제1 샘플의 복수의 성분들을 필터링하기 위한, 제2 샘플의 적어도 제2 성분과 연관되어 있는 적어도 하나의 필터링 파라미터에 대한 참조를 나타낸다.

[0356] 본 발명의 제8 추가의 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 디코딩하는 디코딩 장치가 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 장치는,

[0357] 적어도 하나의 재구성된 샘플의 제1 및 제2 성분을 제공하기 위해 적어도 하나의 인코딩된 샘플의 제1 및 제2 성분을 디코딩하는 수단;

[0358] 적어도 하나의 필터링 파라미터를 디코딩하는 수단; 및

[0359] 루프 필터에서 적어도 하나의 재구성된 샘플을 필터링하는 수단 - 적어도 하나의 재구성된 샘플의 제1 성분 및 제2 성분의 필터링은 디코딩된 적어도 하나의 필터링 파라미터의 적어도 하나의 공통 필터링 파라미터에 기초함 - 을 포함한다.

[0360] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0361] 일 실시예에서, 이 장치는 적어도 하나의 필터링 파라미터에 대한 참조를 나타내는 적어도 하나의 플래그를 획득하는 수단, 적어도 하나의 필터링 파라미터에 대한 참조를 나타내는 적어도 다른 플래그를 획득하는 수단, 및 획득된 적어도 다른 플래그를 분석하는 수단을 포함하고, 적어도 하나의 플래그를 획득하는 수단은 획득된 적어

도 다른 플래그를 분석하는 수단에 응답한다.

[0362] 일 실시예에서, 이 장치는 적어도 하나의 재구성된 샘플의 적어도 제1 성분을 필터링하는 데 사용될 필터링 파라미터를 디코딩된 적어도 하나의 필터링 파라미터의 함수로서 계산하는 수단을 추가로 포함한다.

[0363] 여전히 일 실시예에서, 이 장치는 적어도 하나의 재구성된 샘플의 제1 및 제2 성분을 필터링하는 데 사용될 필터링 파라미터를 디코딩된 적어도 하나의 필터링 파라미터의 함수로서 계산하는 수단을 추가로 포함한다.

[0364] 본 발명의 제9 추가의 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 인코딩하는 인코딩 장치가 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 장치는,

[0365] 적어도 하나의 인코딩된 샘플을 제공하기 위해 영상의 적어도 하나의 샘플의 제1 및 제2 성분을 인코딩하는 수단;

[0366] 적어도 하나의 인코딩된 샘플을 디코딩하는 수단; 및

[0367] 적어도 하나의 필터링 파라미터를 영상의 적어도 하나의 샘플의 성분과 적어도 하나의 디코딩된 샘플의 대응하는 성분 간의 차의 함수로서 계산하는 수단 - 적어도 하나의 필터링 파라미터는 루프 필터에서 디코딩된 샘플의 적어도 제1 및 제2 성분을 독립적으로 필터링하는 데 사용되기 위해 계산됨 - 을 포함한다.

[0368] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0369] 본 발명의 제10 추가의 측면에 따르면, 복수의 샘플들을 포함하는 영상의 적어도 일부를 디코딩하는 디코딩 장치가 제공되고, 각각의 샘플은 적어도 2개의 성분을 포함하며, 인코딩된 영상은 적어도 하나의 필터링 파라미터를 포함하고, 이 장치는,

[0370] 제1 재구성된 샘플의 제1 성분 및 제2 재구성된 샘플의 제2 성분을 제공하기 위해 제1 인코딩된 샘플의 제1 성분 및 제2 인코딩된 샘플의 제2 성분을 디코딩하는 수단;

[0371] 적어도 하나의 필터링 파라미터를 디코딩하는 수단; 및

[0372] 루프 필터에서 재구성된 샘플들을 필터링하는 수단 - 제1 재구성된 샘플의 제1 성분 및 제2 재구성된 샘플의 제2 성분의 필터링은 디코딩된 적어도 하나의 필터링 파라미터의 적어도 하나의 공통 필터링 파라미터에 기초함 - 을 포함한다.

[0373] 성분들 간의 상호 의존성을 고려함으로써, 본 발명은 SAO 파라미터들을 저장하는 데 필요한 메모리 버퍼를 감소시키기 위해 그리고 분류의 복잡도를 감소시키기 위해 SAO의 코딩 효율을 개선하는 것을 가능하게 만들어준다.

[0374] 일 실시예에서, 이 장치는 필터링 파라미터들을 계산하는 수단을 추가로 포함하고, 제1 복수의 필터링 파라미터들은 제1 재구성된 샘플의 제1 성분과 연관되어 있고, 제2 복수의 필터링 파라미터들은 제2 재구성된 샘플의 제2 성분과 연관되어 있으며, 제1 복수의 필터링 파라미터들의 필터링 파라미터 값들은 제2 복수의 필터링 파라미터들의 필터링 파라미터 값들과 같고, 제1 복수의 필터링 파라미터들의 필터링 파라미터 값들은 제2 복수의 필터링 파라미터들의 필터링 파라미터 값들의 순서와 역순으로 되어 있다.

[0375] 다른 실시예에서, 이 장치는 필터링 파라미터들을 계산하는 수단을 추가로 포함하고, 제1 복수의 필터링 파라미터들은 제1 재구성된 샘플의 제1 성분과 연관되어 있고, 제2 복수의 필터링 파라미터들은 제2 재구성된 샘플의 제2 성분과 연관되어 있으며, 제1 복수의 필터링 파라미터들의 필터링 파라미터 값들은 제2 복수의 필터링 파라미터들의 필터링 파라미터 값들과 사전 결정된 가중 인자를 곱한 것과 같다.

[0376] 단지 예로서 제공되어 있고 첨부된 특허청구범위에 의해서만 결정되는 본 발명의 범위를 제한하기 위한 것이 아닌 이상의 예시적인 실시예들을 참조하면 많은 추가의 수정들 및 변형들이 기술 분야의 당업자에게 안출될 것이다. 상세하게는, 상이한 실시예들로부터의 상이한 특징들이, 적절한 경우, 상호 교환될 수 있다.

[0377] 특허청구범위에서, 용어 "포함하는"은 다른 구성요소들 또는 단계들을 배제하지 않으며, 단수 표현 "한" 또는 "어떤"은 복수를 배제하지 않는다. 상이한 특징들이 서로 다른 종속 청구항에 언급되어 있다는 단순한 사실이 이들 특징의 조합이 유리하게 사용될 수 없다는 것을 말하는 것은 아니다.

[0378] 부록

표 1

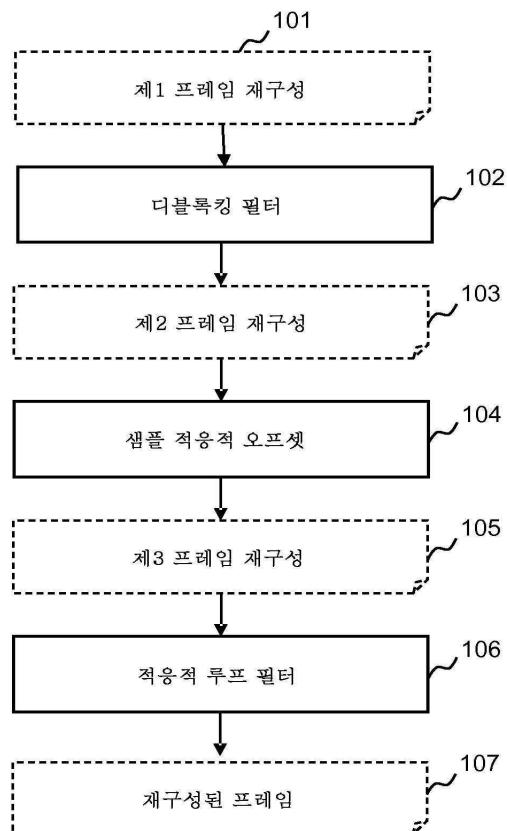
sao_type_idx	SAO 유형	SAO 유형 의미
0	없음	SAO 필터링 없음이 프레임 영역에 적용됨
1	에지	1D 0도
2	에지	1D 90도
3	에지	1D 135도
4	에지	1D 45도
5	대역	대역 위치로 대역 오프셋

표 2

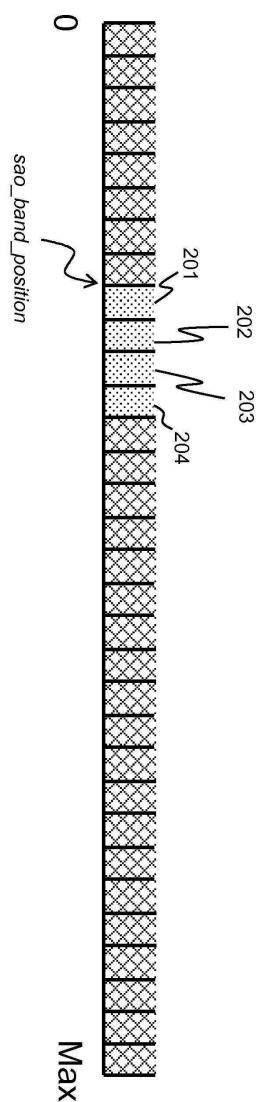
카테고리(J)	조건	오프셋
1	$C < Cn_1$ and $C < Cn_2$	$+O_1$
2	$(C < Cn_1 \text{ and } C == Cn_2) \text{ or } (C < Cn_2 \text{ and } C == Cn_1)$	$+O_2$
3	$(C > Cn_1 \text{ and } C == Cn_2) \text{ or } (C > Cn_2 \text{ and } C == Cn_1)$	$-O_3$
4	$C > Cn_1$ and $C > Cn_2$	$-O_4$
해당 없음	상기한 것들 중 어느 것도 아님	해당 없음

도면

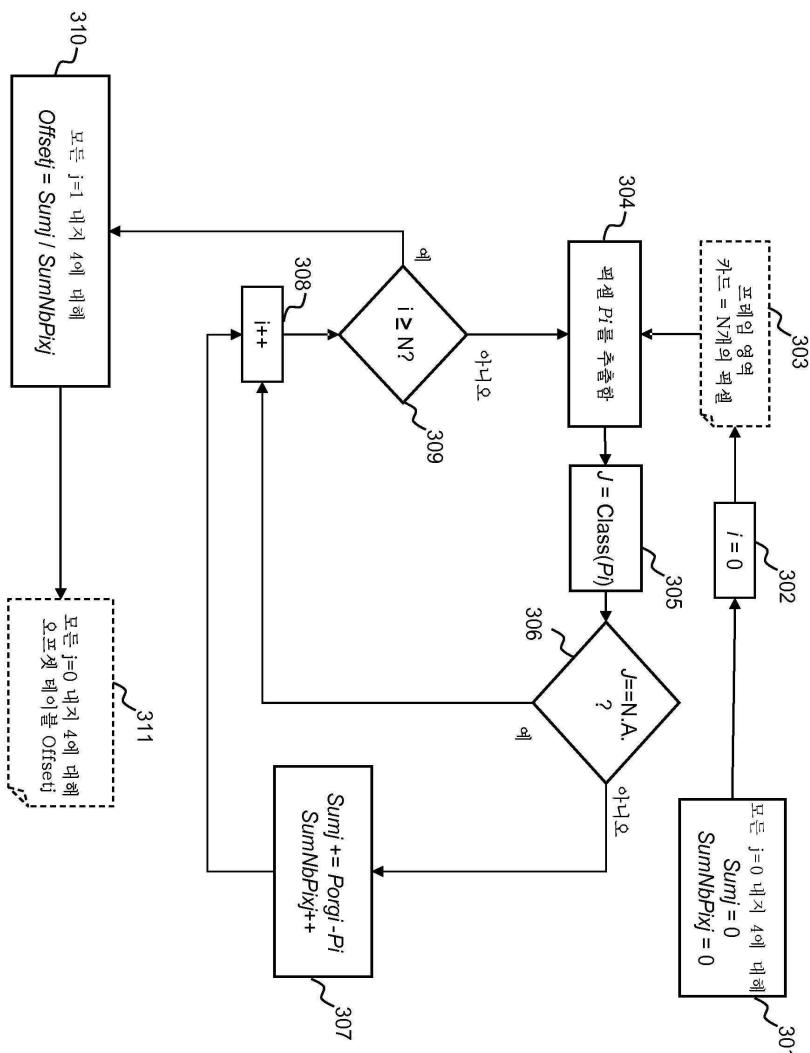
도면1



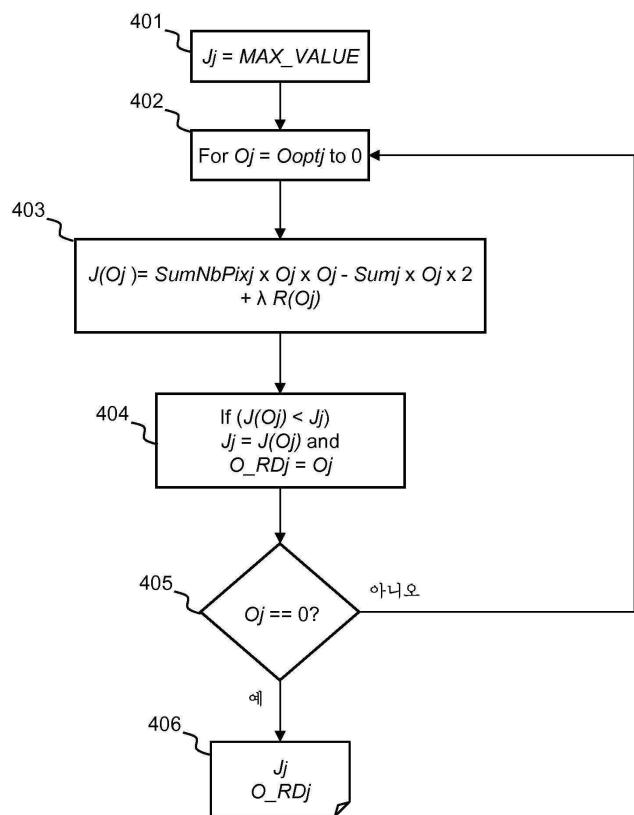
도면2



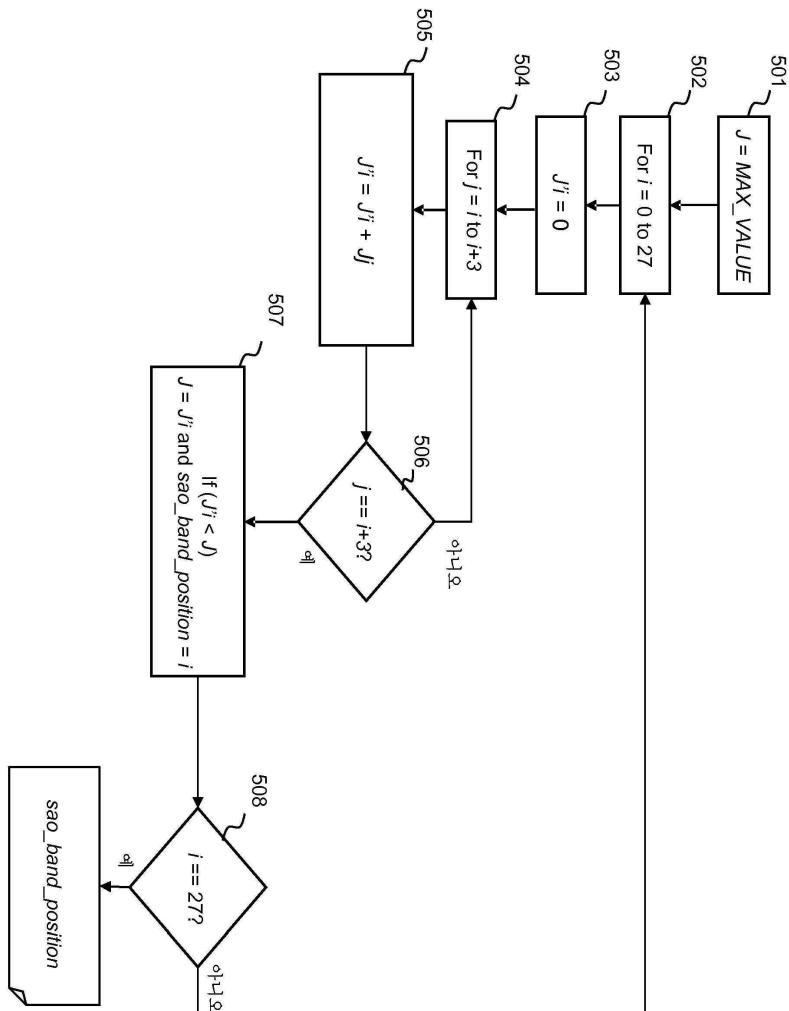
도면3



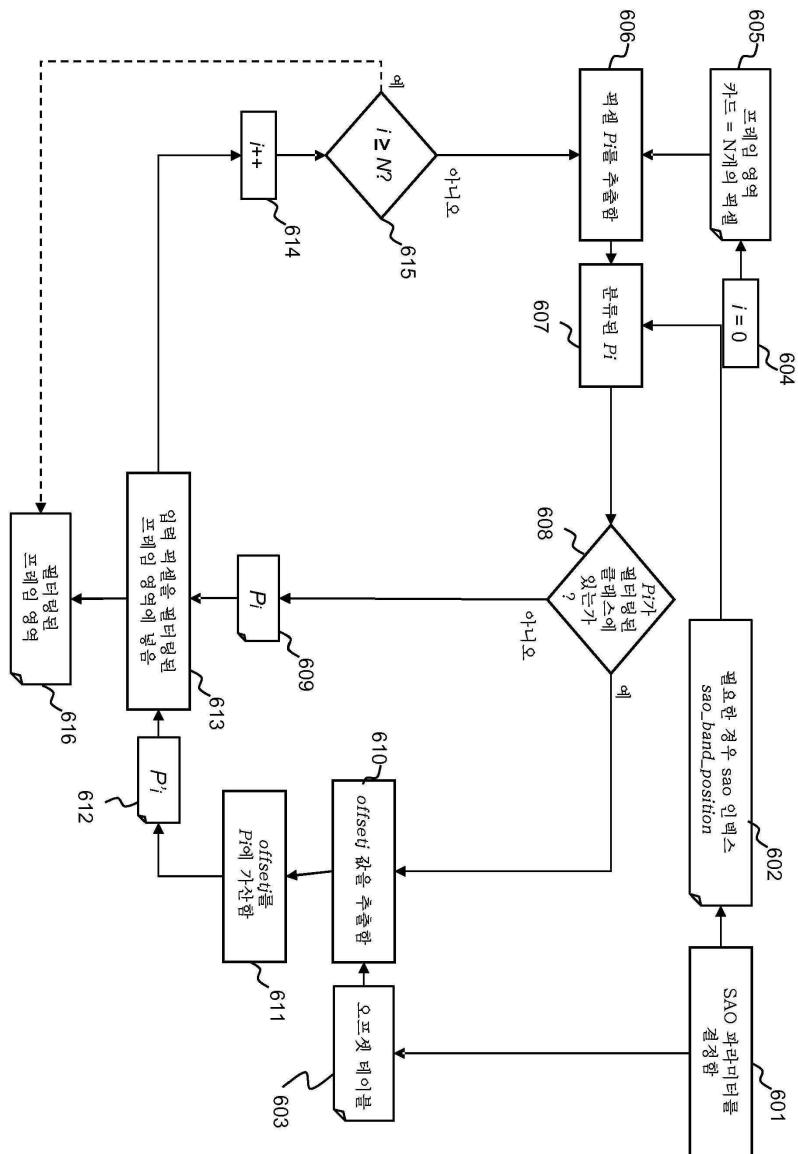
도면4



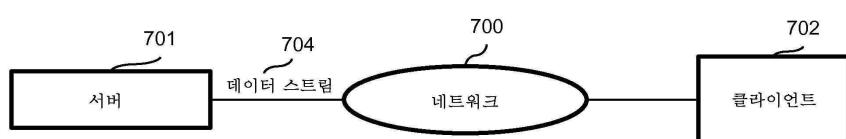
દોષ 5



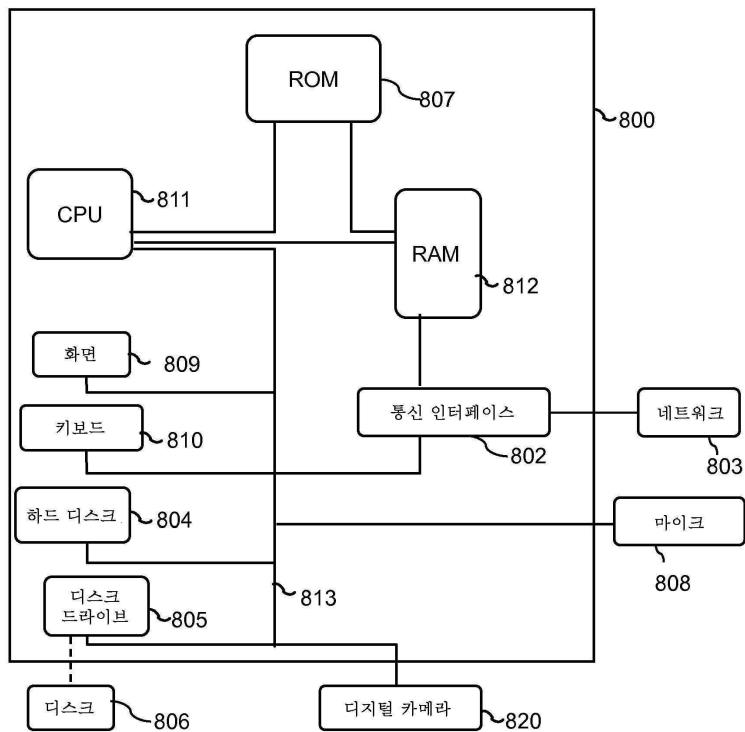
도면6



도면7

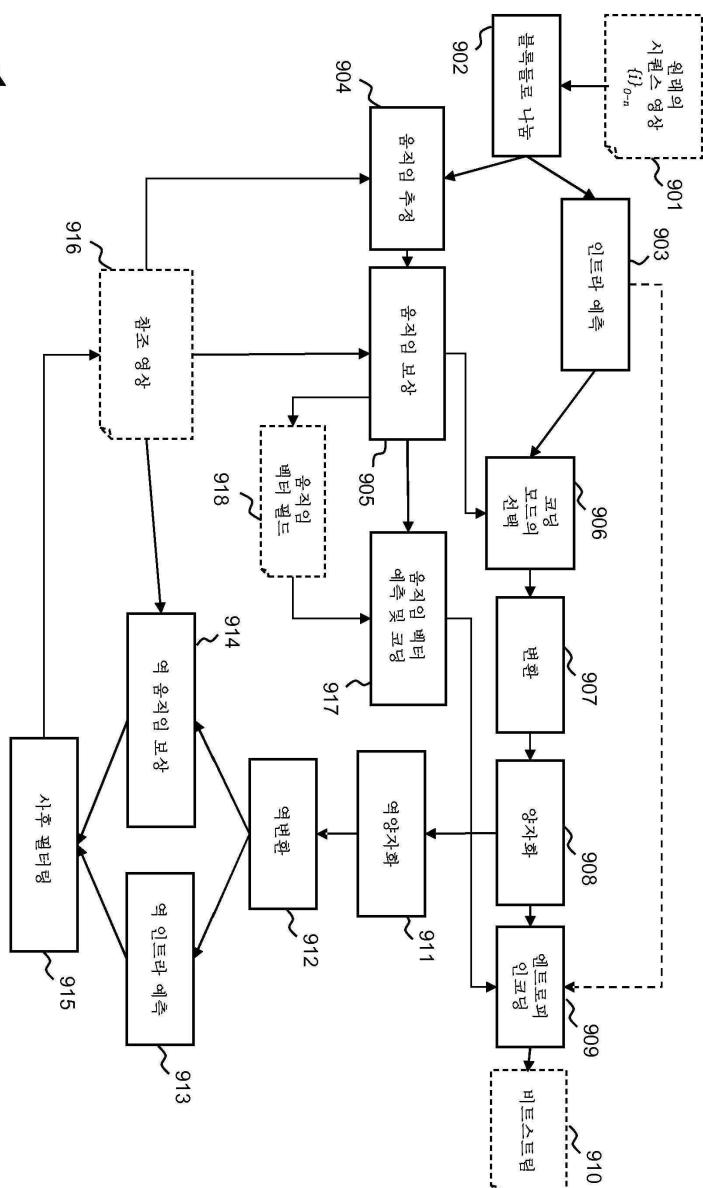


도면8

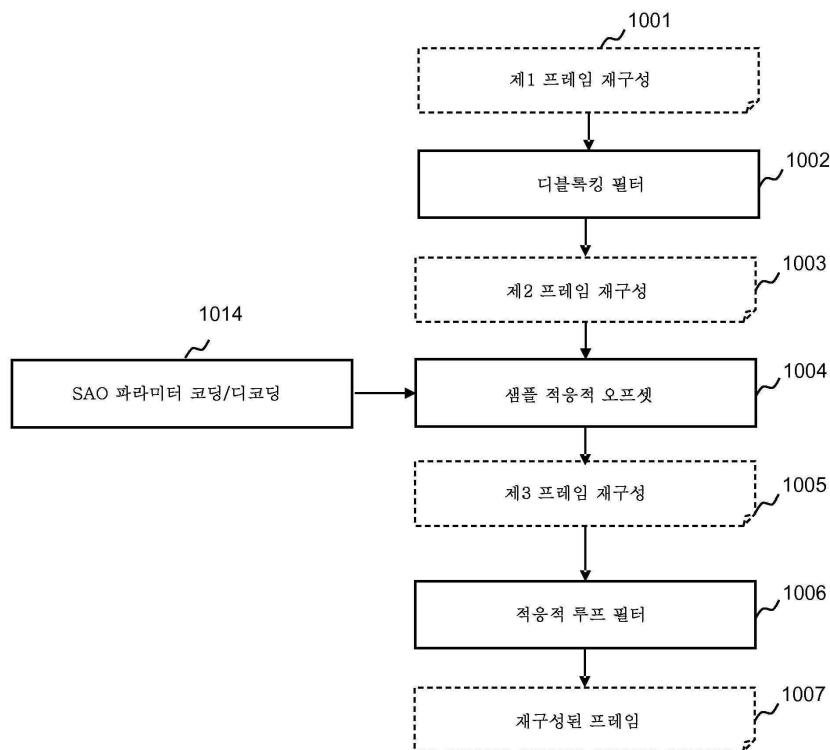


6면

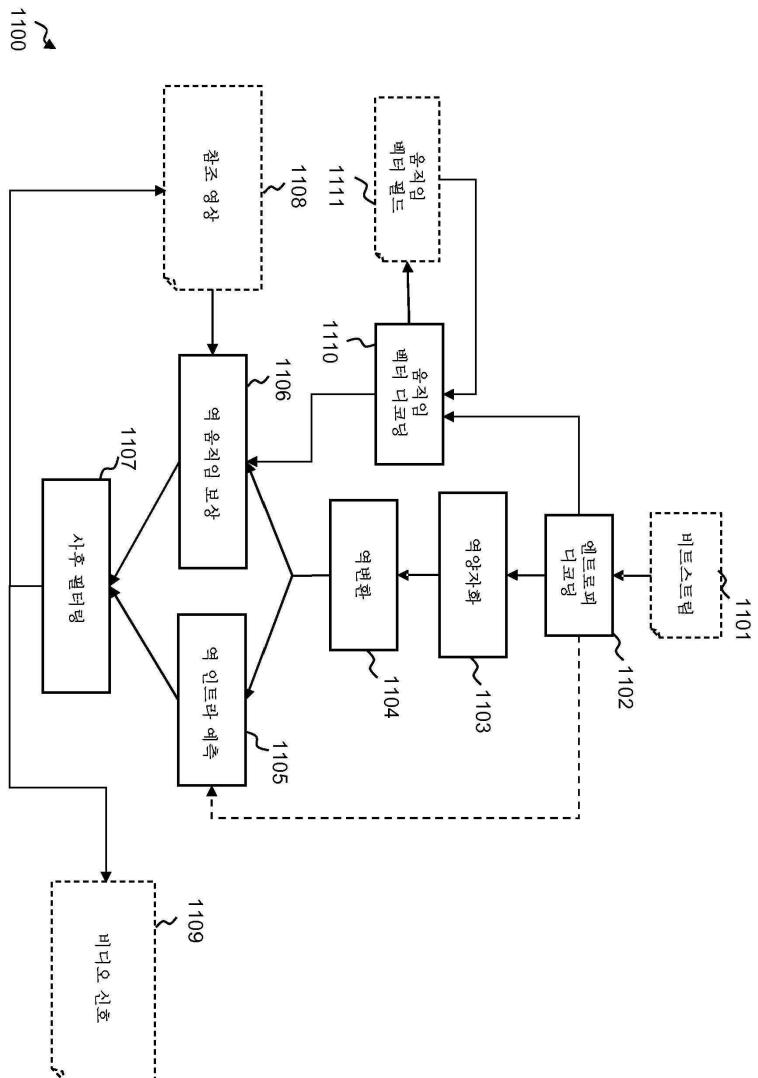
900



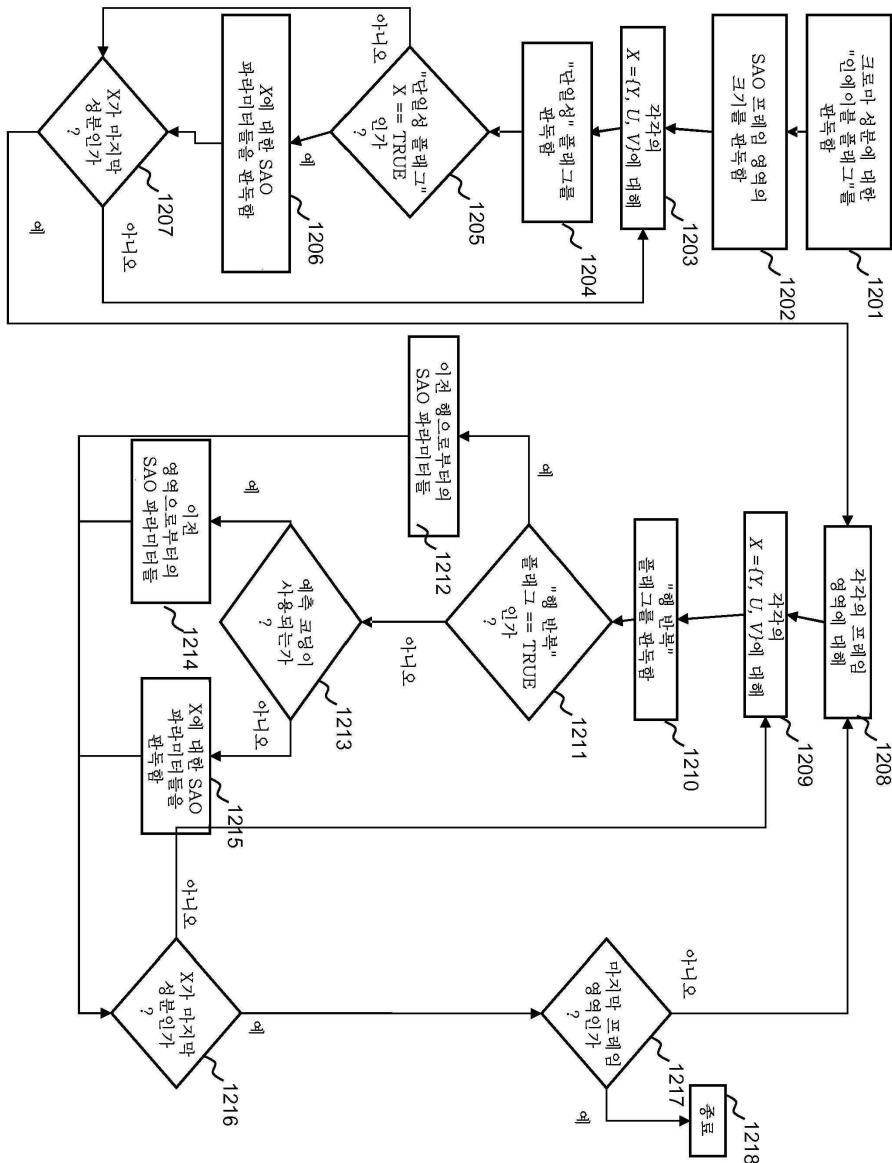
도면10



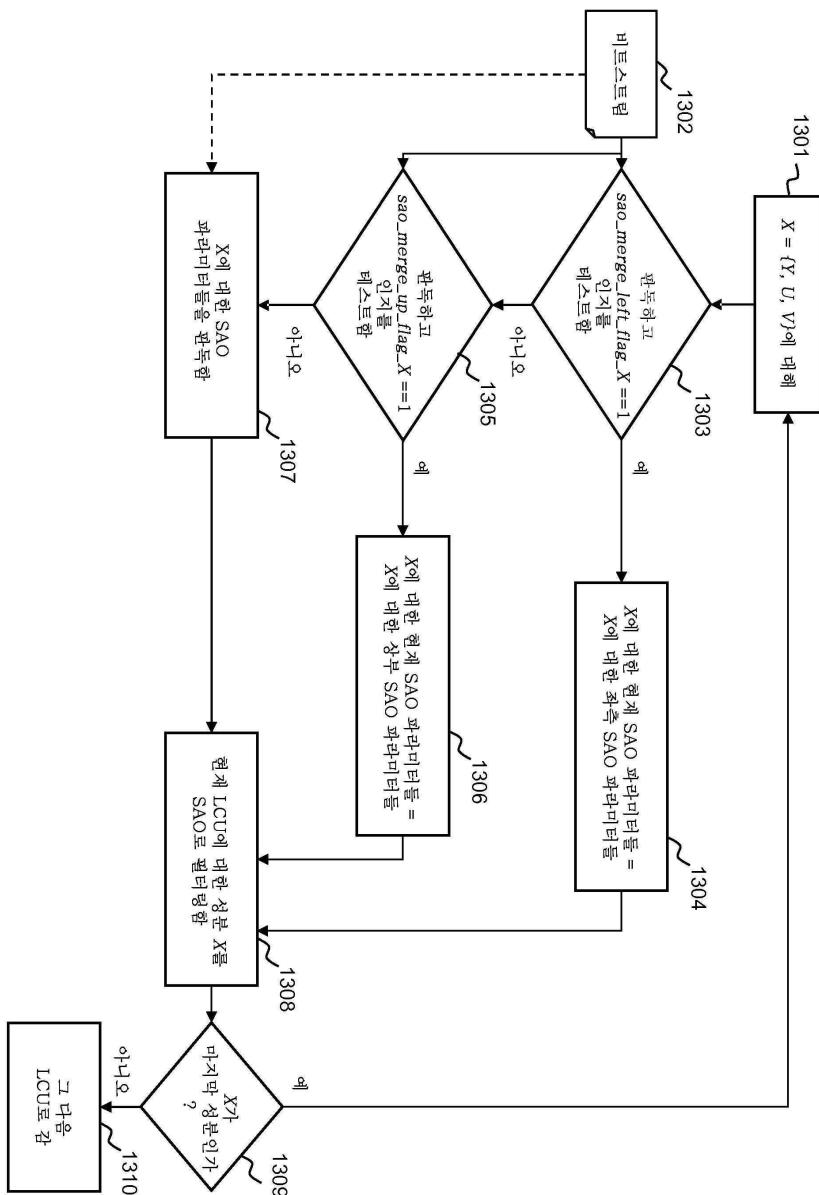
도면11



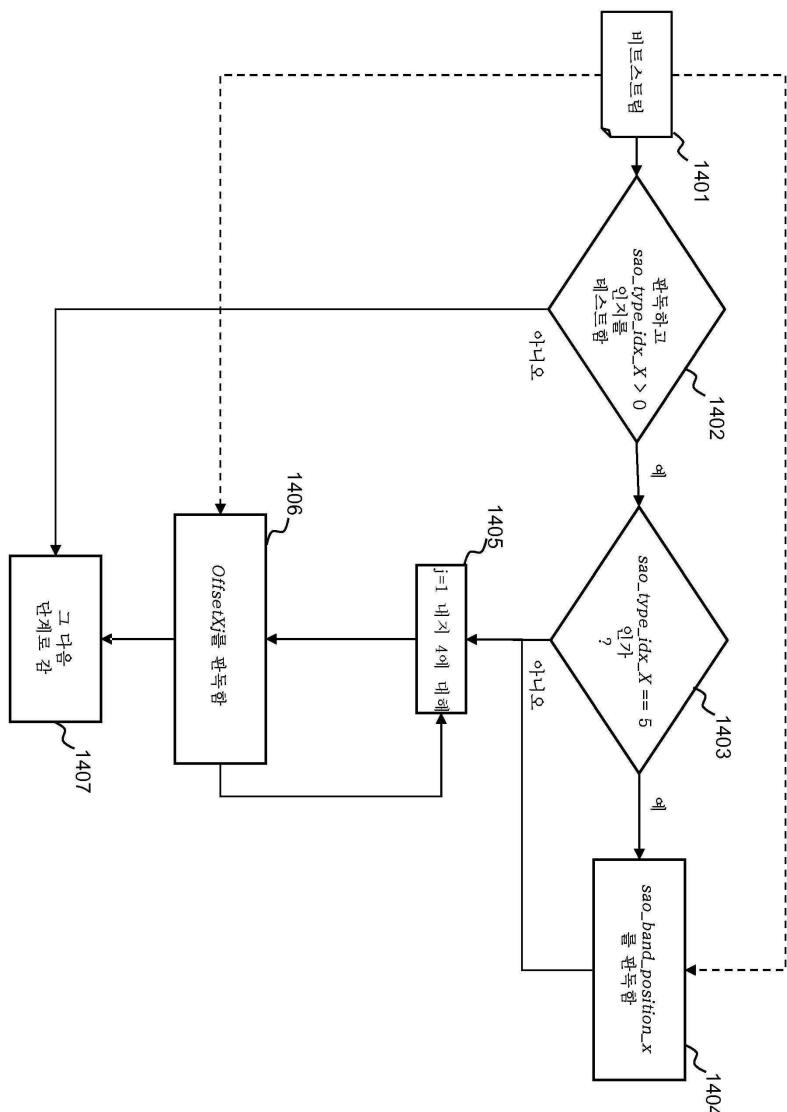
도면12



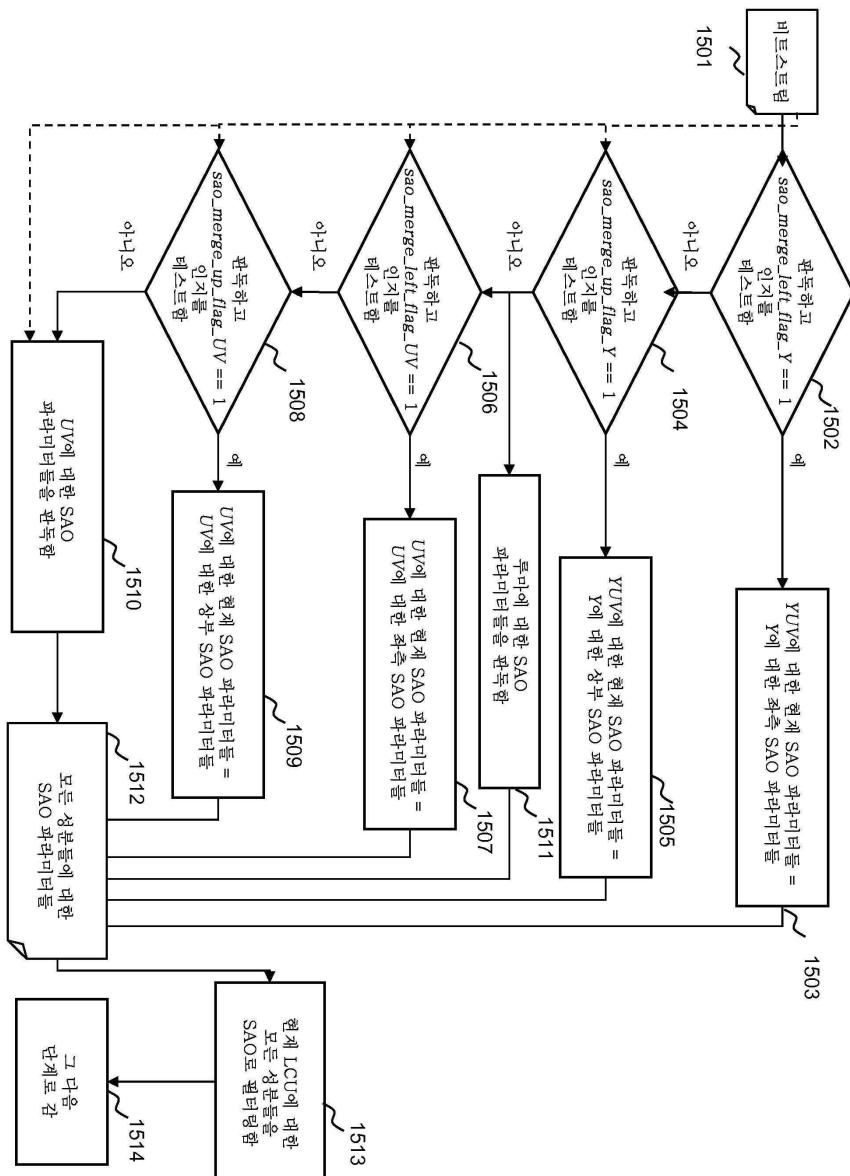
도면13



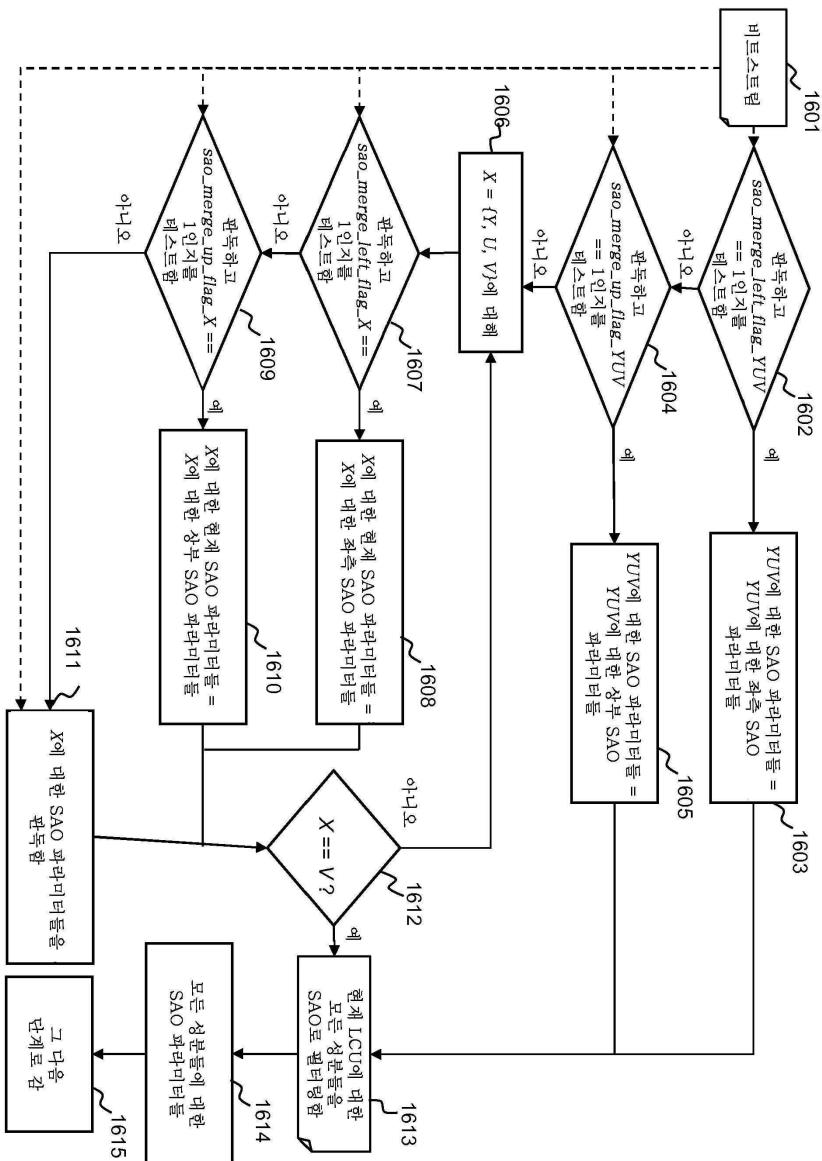
도면14



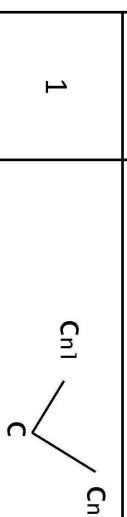
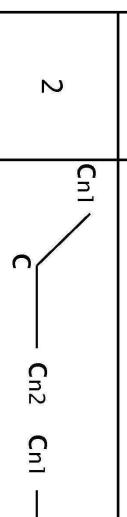
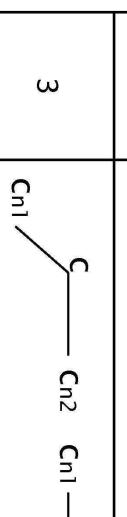
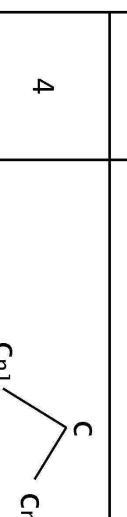
도면15



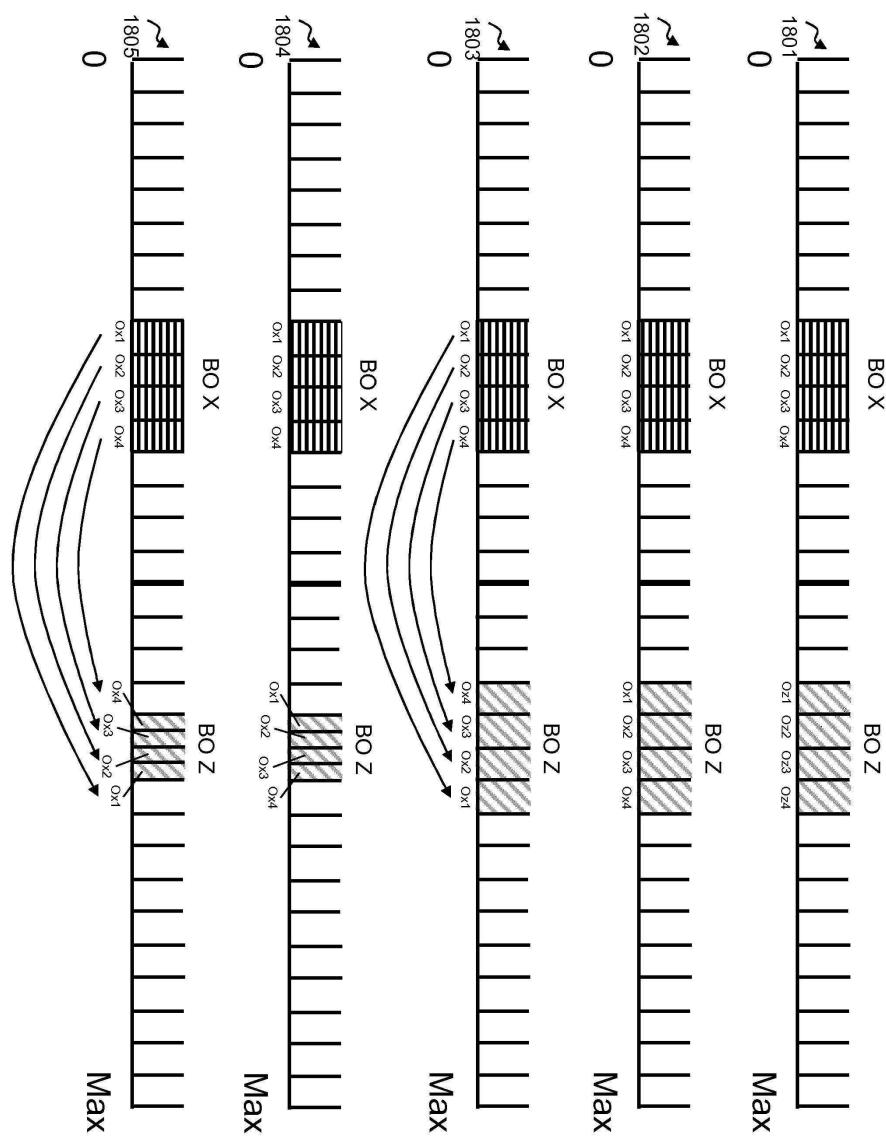
도면16



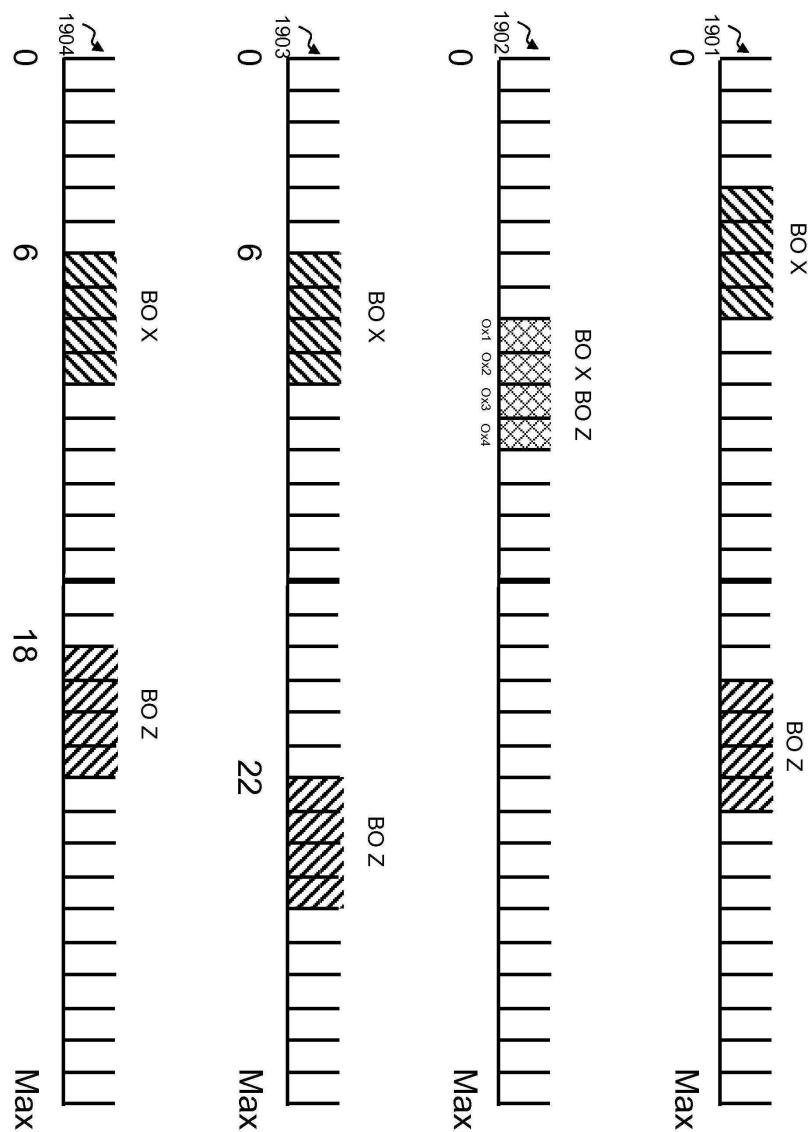
도면17

카테고리	신호 유형	오프셋 X	오프셋 Z
1		$+O_{x1}$	$\pm O_{x4}$
2		$+O_{x2}$	$\pm O_{x3}$
3		$-O_{x3}$	$\pm O_{x2}$
4		$-O_{x4}$	O_{x1}

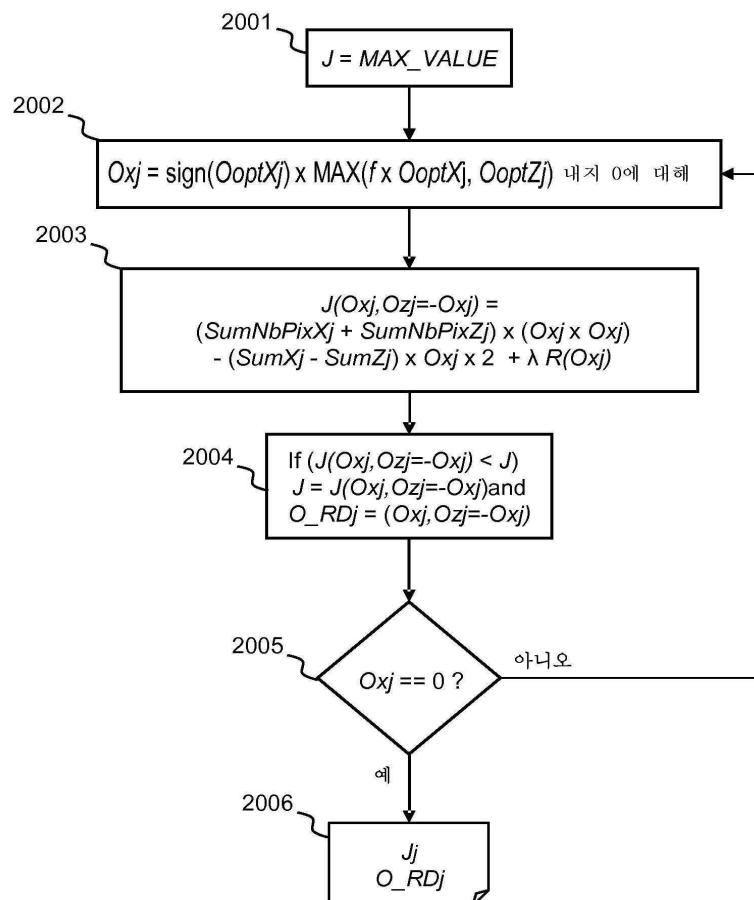
도면18



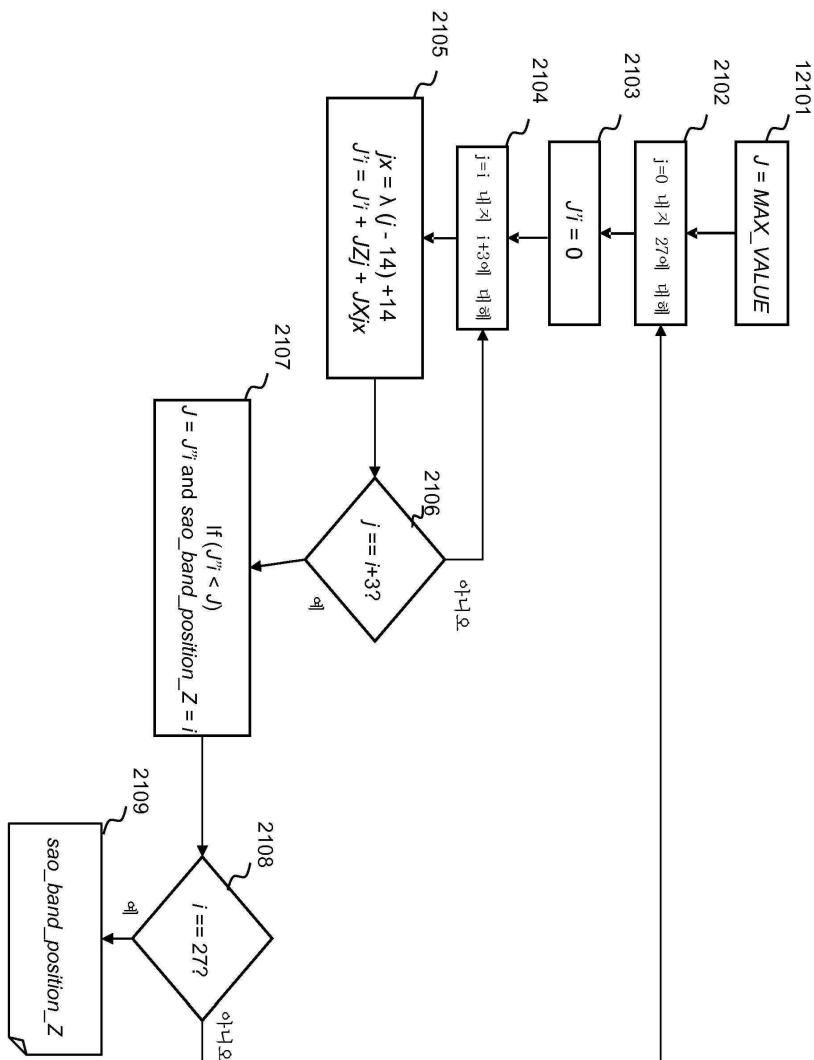
도면19



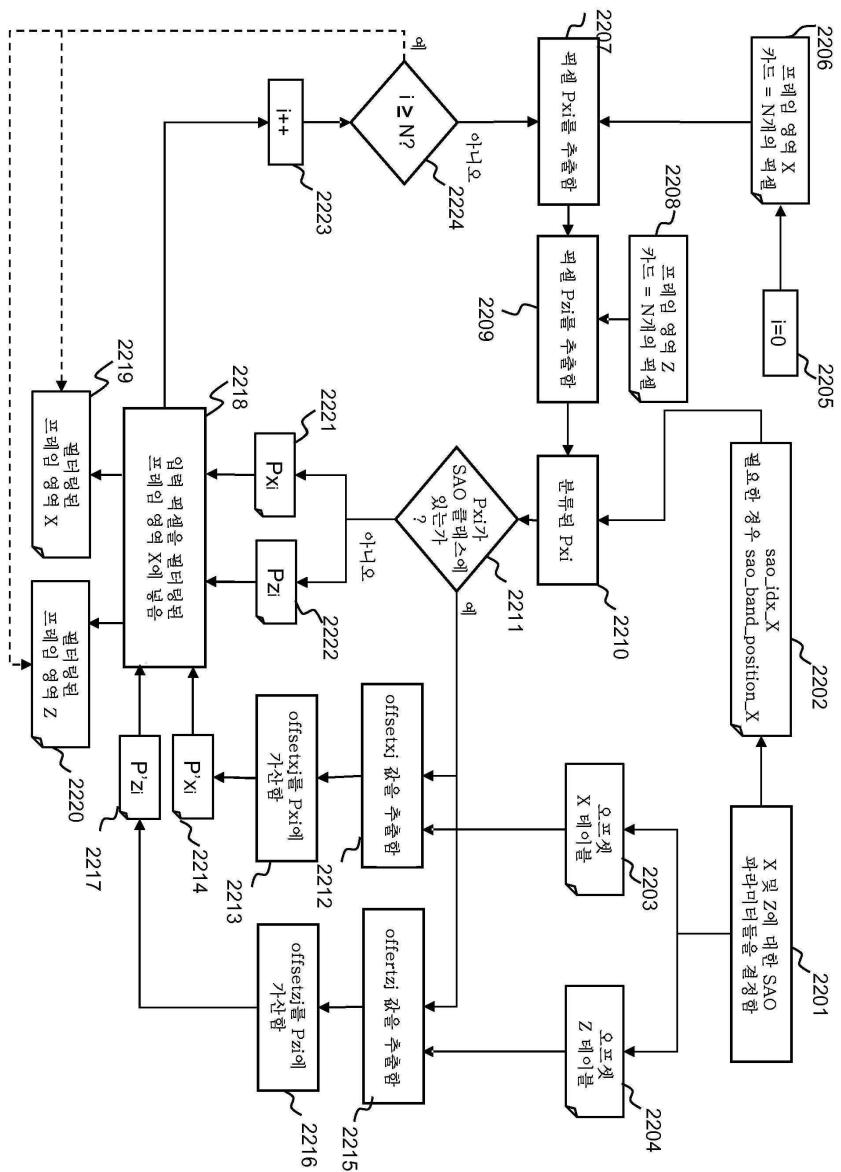
도면20



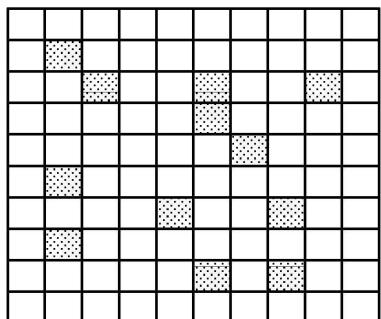
도면21



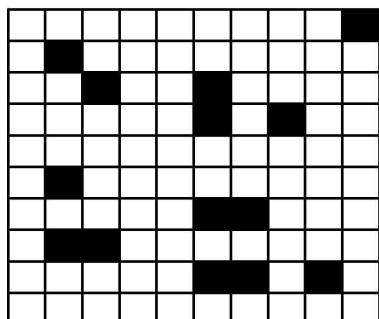
도면22



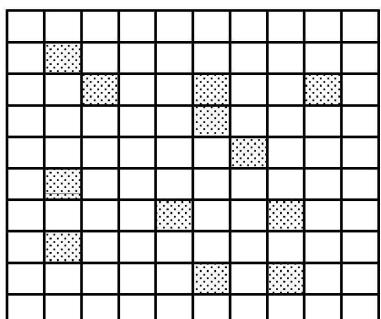
도면23



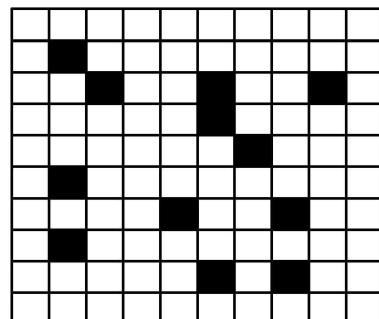
2301



2302



2303



2304