



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0132915
(43) 공개일자 2016년11월21일

(51) 국제특허분류(Int. Cl.)
H04N 19/86 (2014.01) *H04N 19/103* (2014.01)
H04N 19/117 (2014.01) *H04N 19/176* (2014.01)
H04N 19/182 (2014.01) *H04N 19/46* (2014.01)

(52) CPC특허분류
H04N 19/86 (2015.01)
H04N 19/103 (2015.01)

(21) 출원번호 10-2016-7027912

(22) 출원일자(국제) 2015년03월12일

심사청구일자 없음

(85) 번역문제출일자 2016년10월07일

(86) 국제출원번호 PCT/US2015/020160

(87) 국제공개번호 WO 2015/138714

국제공개일자 2015년09월17일

(30) 우선권주장

61/953,503 2014년03월14일 미국(US)

14/644,702 2015년03월11일 미국(US)

(71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(72) 발명자
김 우식
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
천 지안례
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)

(74) 대리인
특허법인코리아나

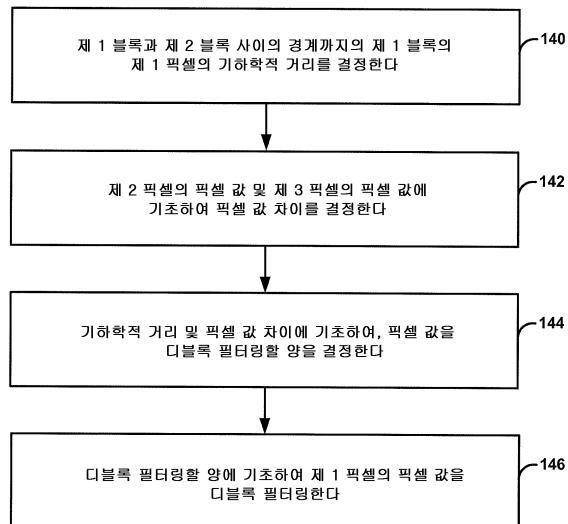
전체 청구항 수 : 총 29 항

(54) 발명의 명칭 **픽셀 거리를 이용한 디블록 필터링**

(57) 요 약

본 개시물은 디블록 필터링 기법들을 기술한다. 비디오 코더는 픽셀로부터 경계까지의 기하학적인 거리 및 적어도 2개의 픽셀들의 픽셀 값들 사이의 픽셀 값 차이에 기초하여 픽셀의 픽셀 값의 디블록 필터링 할 양을 결정할 수도 있다. 비디오 코더는 그후 결정된 양에 기초하여 픽셀을 디블록 필터링 할 수도 있다.

대 표 도 - 도7



(52) CPC특허분류

HO4N 19/117 (2015.01)

HO4N 19/176 (2015.01)

HO4N 19/182 (2015.01)

HO4N 19/46 (2015.01)

(72) 발명자

솔레 로할스 호엘

미국 92121-1714 캘리포니아주 샌디에고 모어하우

스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우

스 드라이브 5775

명세서

청구범위

청구항 1

비디오 데이터를 프로세싱하는 방법으로서,

제 1 블록의 제 1 픽셀로부터 상기 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정하는 단계;

제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정하는 단계;

상기 기하학적인 거리 및 상기 픽셀 값 차이에 기초하여 상기 제 1 픽셀의 픽셀 값을 디블록 필터링할 양을 결정하는 단계; 및

상기 디블록 필터링할 양에 기초하여 상기 제 1 픽셀의 픽셀 값을 디블록 필터링하는 단계를 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 픽셀의 상기 기하학적인 거리를 결정하는 단계는, 상기 제 1 픽셀과 상기 경계 사이의 픽셀들의 수에 기초하여 상기 기하학적인 거리를 결정하는 단계를 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 3

제 1 항에 있어서,

상기 픽셀 값 차이를 결정하는 단계는, 상기 제 2 픽셀의 픽셀 값과 상기 제 3 픽셀의 픽셀 값 사이의 차이를 결정하고, 상기 차이를 2로 나누는 단계를 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 4

제 1 항에 있어서,

상기 제 2 픽셀은 상기 경계에 가장 가까운 상기 제 1 블록의 픽셀을 포함하며, 상기 제 3 픽셀은 상기 경계에 가장 가까운 상기 제 2 블록의 픽셀을 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 5

제 1 항에 있어서,

상기 제 2 픽셀 및 상기 제 1 픽셀은 동일한 픽셀이며, 상기 제 3 픽셀은 상기 제 1 블록에서의 상기 제 1 픽셀과 상기 경계까지 등거리인 상기 제 2 블록에서의 픽셀을 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 6

제 1 항에 있어서,

상기 제 2 픽셀 및 상기 제 1 픽셀은 동일한 픽셀이며, 상기 제 3 픽셀은 상기 경계에 가장 가까운 상기 제 2 블록에서의 픽셀을 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 7

제 1 항에 있어서,

상기 제 2 픽셀은 상기 제 1 픽셀의 제 1 이웃하는 픽셀을 포함하며, 상기 제 3 픽셀은 상기 제 1 픽셀의 제 2 이웃하는 픽셀을 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 8

제 1 항에 있어서,

적어도 상기 픽셀 값 차이에 대한 가중치를 결정하는 단계를 더 포함하며,

상기 디블록 필터링할 양을 결정하는 단계는 상기 기하학적인 거리, 상기 픽셀 값 차이, 및 상기 픽셀 값 차이에 대한 상기 가중치에 기초하여 상기 디블록 필터링할 양을 결정하는 단계를 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 9

제 1 항에 있어서,

상기 디블록 필터링할 양이 제 1 임계값보다 크거나 또는 제 2 임계값 미만이라고 결정하는 단계; 및

상기 제 1 임계값보다 크거나 또는 상기 제 2 임계값 미만인 상기 디블록 필터링할 양에 기초하여, 디블록 필터링할 조정된 양을 결정하는 단계를 더 포함하며,

상기 픽셀 값을 디블록 필터링하는 단계는 상기 디블록 필터링할 조정된 양에 기초하여 상기 제 1 픽셀의 픽셀 값을 디블록 필터링하는 단계를 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 10

제 1 항에 있어서,

상기 디블록 필터링할 양을 결정하는 단계는, 상기 기하학적인 거리, 상기 픽셀 값 차이, 및 디블록 필터링될 픽셀들의 수에 기초하여, 상기 디블록 필터링할 양을 결정하는 단계를 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 11

제 1 항에 있어서,

상기 제 1 블록의 부분에 대한 기울기 값을 결정하는 단계를 더 포함하며, 상기 기울기 값은 상기 제 1 블록의 상기 부분의 평활도를 표시하며,

디블록 필터링하는 단계는 임계치 기울기 값 미만이거나 또는 동일한 상기 기울기 값에 기초하여 픽셀 값을 디블록 필터링하는 단계를 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 12

제 1 항에 있어서,

상기 제 1 블록을 발생시키는데 사용되는 제 1 변환 블록의 제 1 블록 사이즈 및 상기 제 2 블록을 발생시키는데 사용되는 제 2 변환 블록의 제 2 블록 사이즈를 결정하는 단계를 더 포함하며,

디블록 필터링하는 단계는 최대 허용가능 변환 블록 사이즈와 동일한 상기 제 1 블록 사이즈 및 상기 제 2 블록 사이즈 중 더 큰 사이즈, 또는 임계치 블록 사이즈보다 큰 상기 제 1 블록 사이즈 및 상기 제 2 블록 사이즈 중 더 작은 사이즈에 기초하여, 픽셀 값을 디블록 필터링하는 단계를 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 13

제 1 항에 있어서,

제 3 블록에서의 픽셀의 픽셀 값과 상기 디블록 필터링된 픽셀 값 사이의 차이를 결정하는 단계; 및

상기 제 3 블록에서의 픽셀의 픽셀 값과 상기 디블록 필터링된 픽셀 값 사이의 차이를 표시하는 정보를 시그널링하는 단계를 더 포함하며,

시그널링된 상기 정보는 상기 제 3 블록을 디코딩하는데 사용되는, 비디오 데이터를 프로세싱하는 방법.

청구항 14

제 1 항에 있어서,

제 3 블록에서의 픽셀의 픽셀 값과 상기 디블록 필터링된 픽셀 값 사이의 차이를 표시하는 정보를 수신하는 단계; 및

수신된 상기 정보에 기초하여 상기 제 3 블록을 인터-예측 또는 인트라-예측 디코딩하는 단계를 더 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 15

비디오 데이터를 프로세싱하는 디바이스로서,

픽셀 값을 저장하는 비디오 데이터 메모리; 및

비디오 코더를 포함하며,

상기 비디오 코더는,

제 1 블록의 제 1 픽셀로부터 상기 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정하는 것으로서, 상기 제 2 블록의 픽셀 값을 상기 비디오 데이터 메모리에 저장되는, 상기 기하학적인 거리를 결정하고;

제 2 픽셀의 픽셀 값과 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정하고;

상기 기하학적인 거리 및 상기 픽셀 값 차이에 기초하여 상기 제 1 픽셀의 픽셀 값을 디블록 필터링할 양을 결정하고; 그리고

상기 디블록 필터링할 양에 기초하여 상기 제 1 픽셀의 픽셀 값을 디블록 필터링하도록

구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 16

제 15 항에 있어서,

상기 제 1 픽셀의 상기 기하학적인 거리를 결정하기 위해, 상기 비디오 코더는 상기 제 1 픽셀과 상기 경계 사이의 픽셀들의 수에 기초하여 상기 기하학적인 거리를 결정하도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 17

제 15 항에 있어서,

상기 픽셀 값 차이를 결정하기 위해, 상기 비디오 코더는 상기 제 2 픽셀의 픽셀 값과 상기 제 3 픽셀의 픽셀 값 사이의 차이를 결정하고 상기 차이를 2로 나누도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 18

제 15 항에 있어서,

상기 제 2 픽셀은 상기 경계에 가장 가까운 상기 제 1 블록의 픽셀을 포함하며, 상기 제 3 픽셀은 상기 경계에 가장 가까운 상기 제 2 블록의 픽셀을 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 19

제 15 항에 있어서,

상기 제 2 픽셀 및 상기 제 1 픽셀은 동일한 픽셀이며, 상기 제 3 픽셀은 상기 제 1 블록에서의 상기 제 1 픽셀과 상기 경계까지 등거리인 상기 제 2 블록에서의 픽셀을 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 20

제 15 항에 있어서,

상기 제 2 픽셀 및 상기 제 1 픽셀은 동일한 픽셀이며, 상기 제 3 픽셀은 상기 경계에 가장 가까운 상기 제 2 블록에서의 픽셀을 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 21

제 15 항에 있어서,

상기 제 2 픽셀은 상기 제 1 픽셀의 제 1 이웃하는 픽셀을 포함하며, 상기 제 3 픽셀은 상기 제 1 픽셀의 제 2 이웃하는 픽셀을 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 22

제 15 항에 있어서,

상기 비디오 코더는, 적어도 상기 픽셀 값 차이에 대한 가중치를 결정하도록 구성되며,

상기 디블록 필터링할 양을 결정하기 위해, 상기 비디오 코더는 상기 기하학적인 거리, 상기 픽셀 값 차이, 및 상기 픽셀 값 차이에 대한 상기 가중치에 기초하여 상기 디블록 필터링할 양을 결정하도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 23

제 15 항에 있어서,

상기 비디오 코더는,

상기 디블록 필터링할 양이 제 1 임계값보다 크거나 또는 제 2 임계값 미만이라고 결정하고; 그리고

상기 제 1 임계값보다 크거나 또는 상기 제 2 임계값 미만인 상기 디블록 필터링할 양에 기초하여, 디블록 필터링할 조정된 양을 결정하도록 구성되며,

상기 픽셀 값을 디블록 필터링하기 위해, 상기 비디오 코더는 상기 디블록 필터링할 조정된 양에 기초하여 상기 제 1 픽셀의 픽셀 값을 디블록 필터링하도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 24

제 15 항에 있어서,

상기 디블록 필터링할 양을 결정하기 위해, 상기 비디오 코더는 상기 기하학적인 거리, 상기 픽셀 값 차이, 및 디블록 필터링될 픽셀들의 수에 기초하여 상기 디블록 필터링할 양을 결정하도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 25

제 15 항에 있어서,

상기 비디오 코더는, 상기 제 1 블록의 부분에 대한 기울기 값을 결정하도록 구성되며, 상기 기울기 값은 상기 제 1 블록의 상기 부분의 평활도를 표시하며,

디블록 필터링하기 위해, 상기 비디오 코더는 임계치 기울기 값 미만이거나 또는 동일한 상기 기울기 값에 기초하여 픽셀 값을 디블록 필터링하도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 26

제 15 항에 있어서,

상기 비디오 코더는, 상기 제 1 블록을 발생시키는데 사용되는 제 1 변환 블록의 제 1 블록 사이즈 및 상기 제 2 블록을 발생시키는데 사용되는 제 2 변환 블록의 제 2 블록 사이즈를 결정하도록 구성되며,

디블록 필터링하기 위해, 상기 비디오 코더는 최대 허용가능 변환 블록 사이즈와 동일한 상기 제 1 블록 사이즈 및 상기 제 2 블록 사이즈 중 더 큰 사이즈, 또는 임계치 블록 사이즈보다 큰 상기 제 1 블록 사이즈 및 상기

제 2 블록 사이즈 중 더 작은 사이즈에 기초하여 픽셀 값을 디블록 필터링하도록 구성되는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 27

제 15 항에 있어서,

상기 비디오 데이터를 프로세싱하는 디바이스는,

마이크로프로세서;

집적 회로 (IC); 또는

무선 핸드셋 디바이스

중 하나를 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 28

비디오 데이터를 프로세싱하는 디바이스로서,

제 1 블록의 제 1 픽셀로부터 상기 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정하는 수단;

제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정하는 수단;

상기 기하학적인 거리 및 상기 픽셀 값 차이에 기초하여 상기 제 1 픽셀의 픽셀 값을 디블록 필터링할 양을 결정하는 수단; 및

상기 디블록 필터링 할 양에 기초하여 상기 제 1 픽셀의 픽셀 값을 디블록 필터링하는 수단을 포함하는, 비디오 데이터를 프로세싱하는 디바이스.

청구항 29

명령들이 저장된 컴퓨터-판독가능 저장 매체로서,

상기 명령들은, 실행될 경우, 프로세싱하는 디바이스의 하나 이상의 프로세서들로 하여금,

제 1 블록의 제 1 픽셀로부터 상기 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정하게 하고;

제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정하게 하고;

상기 기하학적인 거리 및 상기 픽셀 값 차이에 기초하여 상기 제 1 픽셀의 픽셀 값을 디블록 필터링할 양을 결정하게 하고; 그리고

상기 디블록 필터링 할 양에 기초하여 상기 제 1 픽셀의 픽셀 값을 디블록 필터링하게 하는, 컴퓨터-판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 출원은 2014년 3월 14일에 출원된 미국 가출원 번호 제 61/953,503호의 이익을 주장하며, 이의 전체 내용이 본원에 전체적으로 참고로 포함된다.

기술 분야

[0003] 본 개시물은 비디오 코딩에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기들 (PDA들), 랩탑 또는 데스크탑 컴퓨터들, 디지털 카메라들, 디지털 리코딩 디바이스들,

디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 원격 화상회의 디바이스들 등을 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 디지털 비디오 정보를 좀 더 효율적으로 송수신하고 저장하기 위해, MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 퍼트 10, AVC (Advanced Video Coding) 에 의해 정의된 표준들, HEVC (High Efficiency Video Coding) 표준, 및 이런 표준들의 확장판들에 설명된 것들과 같은 비디오 압축 기법들을 구현한다.

[0005] 비디오 압축 기법들은 비디오 시퀀스들에 고유한 리턴던시를 감소시키거나 또는 제거하기 위해 공간 예측 및/또는 시간 예측을 포함한다. 블록-기반의 비디오 코딩에 있어, 비디오 프레임 또는 슬라이스는 블록들로 파티셔닝될 수도 있다. 각각의 블록은 더 파티셔닝될 수도 있다. 인트라-코딩된 (I) 프레임 또는 슬라이스에서의 블록들은 동일한 프레임 또는 슬라이스에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 인터-코딩된 (P 또는 B) 프레임 또는 슬라이스에서의 블록들은 동일한 프레임 또는 슬라이스에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 프레임들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 공간 또는 시간 예측은 코딩될 블록에 대한 예측 블록을 초래한다.

잔차 데이터는 코딩될 원래 블록과 예측 블록 사이의 픽셀 차이들을 나타낸다.

[0006] 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터, 및 코딩된 블록과 예측 블록 사이의 차이를 나타내는 잔차 데이터에 따라서 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라서 인코딩된다. 추가적인 압축을 위해, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어, 잔차 변환 계수들을 초래할 수도 있으며, 이 잔차 변환 계수는 그후 양자화될 수도 있다.

처음에 2차원 어레이로 배열된, 양자화된 변환 계수들은 엔트로피 코딩을 위한 변환 계수들의 1차원 벡터를 발생하기 위해 특정의 순서로 스캐닝될 수도 있다.

발명의 내용

과제의 해결 수단

[0007] 본 개시물은 픽처의 복원된 블록들을 디블록 필터링하는 예시적인 기법들을 설명한다. 비디오 코더 (예컨대, 비디오 인코더 또는 비디오 디코더) 는 블록의 경계까지의 픽셀의 거리에 기초하여, 그리고 픽셀들의 픽셀 값들 사이의 픽셀 값 차이에 기초하여, 블록의 픽셀의 픽셀 값을 디블록 필터링할 양을 결정한다. 일부 예들에서, 비디오 코더는 또한 픽셀의 픽셀 값을 디블록 필터링할 양을 결정하기 위해 디블록 필터링될 픽셀들의 개수를 이용할 수도 있다.

[0008] 일 예에서, 본 개시물은 비디오 데이터를 프로세싱하는 방법을 설명하며, 그 방법은 제 1 블록의 제 1 픽셀로부터 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정하는 단계; 제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정하는 단계; 기하학적인 거리 및 픽셀 값 차이에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링할 양을 결정하는 단계; 및 디블록 필터링할 양에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링하는 단계를 포함한다.

[0009] 일 예에서, 본 개시물은 비디오 데이터를 프로세싱하는 디바이스를 설명하며, 그 디바이스는 픽셀 값을 저장하는 비디오 데이터 메모리; 및 비디오 코더를 포함하며, 상기 비디오 코더는 제 1 블록의 제 1 픽셀로부터 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정하는 것으로서, 제 2 블록의 픽셀 값들은 비디오 데이터 메모리에 저장되는, 상기 기하학적인 거리를 결정하고, 제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정하고, 기하학적인 거리 및 픽셀 값 차이에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링할 양을 결정하고, 그리고 디블록 필터링할 양에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링하도록 구성된다.

[0010] 일 예에서, 본 개시물은 비디오 데이터를 프로세싱하는 디바이스를 설명하며, 그 디바이스는 제 1 블록의 제 1 픽셀로부터 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정하는 수단; 제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정하는 수단; 기하학적인 거리 및 픽셀 값 차이에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링할 양을 결정하는 수단; 및 디블록 필터링할 양에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링하는 수단을 포함한다.

[0011] 일 예에서, 본 개시물은 실행될 때, 하나 이상의 프로세서들로 하여금, 제 1 블록의 제 1 픽셀로부터 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정하게 하고, 제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정하게 하고, 기하학적인 거리 및 픽셀 값 차이에 기초하여 제 1 픽셀의 픽셀

값을 디블록 필터링할 양을 결정하게 하고, 그리고 디블록 필터링할 양에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링하게 하는 명령들을 저장하고 있는 컴퓨터-판독가능 저장 매체를 설명한다.

[0012] 하나 이상의 예들의 세부 사항들이 첨부도면 및 아래의 상세한 설명에서 개시된다. 다른 특성들, 목적들, 및 이점들은 설명 및 도면들로부터, 그리고 청구항들로부터 명백히 알 수 있을 것이다.

도면의 간단한 설명

[0013] 도 1 은 본 개시물에서 설명하는 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2 는 본 개시물에서 설명하는 기법들을 구현할 수도 있는 예시적인 비디오 인코더를 예시하는 블록도이다.

도 4 는 블록 경계를 가로지르는 픽셀 위치들에 대한 픽셀 값들의 일 예를 예시하는 개념도이다.

도 5 는 본 개시물에서 설명하는 기법들에 따라서 시그널링되는 디블록킹 필터 파라미터들에 기초하여 정의된 예시적인 디블록킹 필터의 구성요소들을 예시하는 블록도이다.

도 6 은 서브-블록들 사이의 비디오 블록의 예지에 가까운 픽셀 위치들을 예시하는 개념도이다.

도 7 은 비디오 데이터를 프로세싱하는 예시적인 기법을 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0014] 본 개시물은 비디오 코딩 및 압축의 분야에 관한 것이다. 일 예로서, 본 개시물은 HEVC (High Efficiency Video Coding) 표준 및 그의 범위 확장판 및 스크린 콘텐츠 코딩 확장판에 관련된다. 그러나, 본 기법들은 HEVC 표준 또는 그의 범위 확장판 및 스크린 콘텐츠 코딩 확장판에 한정되는 것으로 간주되어서는 안된다. 설명의 용이성을 위해, 기법들은 HEVC 표준에 대해 설명되지만, 기법들은 다른 비디오 코딩 표준들에 적용가능 할 수도 있으며, 일반적으로 비디오 코딩 기법들에 적용가능하다. 또한, 본 개시물에서 설명하는 기법들은 미래에 개발될 표준들의 일부가 될 수도 있다. 다시 말해서, 본 개시물에서 설명하는 기법들은 이전에 개발 된 비디오 코딩 표준들, 현재 개발 중인 비디오 코딩 표준들, 및 곧 있을 비디오 코딩 표준들에 적용가능할 수도 있다.

[0015] 비디오 코딩의 부분으로서, 비디오 인코더는 인코딩된 비디오 블록을 복원하고, 복원된 비디오 블록을 비디오 인코더의 디코딩된 픽처 버퍼 (DPB)에 저장하여, 복원된 비디오 블록이 후속 블록의 인터-예측 또는 인트라-예측에 사용될 수 있다. 비디오 디코더는 인코딩된 비디오 블록에 대한 비디오 데이터를 수신하고, 궁극적인 디스플레이를 위해 비디오 블록을 복원한다.

[0016] 일부의 경우, 블록의 경계를 따라서 블록킹 아티팩트들이 존재할 수도 있다. 이러한 블록킹 아티팩트들은, 특히 복원에 사용되는 블록의 블록 사이즈가 상대적으로 클 때 뷰어에게 현저할 수도 있다. 블록킹 아티팩트는, 일반적으로 뷰어가 하나의 블록으로부터 다른 블록으로의 전이를 인지할 수 없도록, 하나의 블록으로부터 다른 블록으로의 혼합된 전이보다는, 뷰어가 비디오 콘텐츠에서 두드러진 (distinct) 비디오 블록들을 인지하도록 하는 블록의 경계들에서의 파괴들 (disruptions)을 지칭한다. 더 일반적으로는, 블록킹 아티팩트들은 비디오 콘텐츠에서 하나의 블록으로부터 다른 블록까지 인지가능한 파괴를 초래하는 블록-기반 비디오 코딩의 결과일 수도 있다.

[0017] 본 개시물에서 설명하는 바와 같이, 블록킹 아티팩트를 평활화하기 위해, 비디오 코더 (예컨대, 비디오 인코더 또는 비디오 디코더)는 블록 경계에 가까운 복수의 픽셀들 (예컨대, 블록의 경계로부터 연장하는 N 개의 픽셀들) 상에 디블록킹 필터를 적용할 수도 있다. 블록킹 아티팩트의 평활화는, 블록의 하나의 측면으로부터의 픽셀 값들의 변화가 픽셀 값들 사이의 갑작스러운 급상승보다는, 블록 경계를 가로질러 매끄러운 전이가 되도록, 이웃하는 블록들 사이의 경계의 반대 측면들 상의 픽셀들 사이의 픽셀 값들에서의 갑작스러운 급상승을 평균화한다. 다시 말해서, 블록킹 아티팩트를 평활화하는 것은 픽셀 값들에서 매끄러운 전이가 존재하도록 경계에 가까운 픽셀 값들이 변하는 양에 있어서 감소를 초래한다.

[0018] 본 개시물은 블록의 경계까지의 픽셀의 기하학적인 거리 및 픽셀 값들에서의 차이 (즉, 픽셀 값 차이)에 기초 하여, 블록에서의 픽셀들의 픽셀 값을 디블록 필터링할 양을 결정하는 예를 설명한다. 블록에서의 픽셀의 픽셀 값은 루마 블록에서의 픽셀의 루마 값, 또는 크로마 블록들 중 하나에서의 픽셀의 크로마 값들 중 하나일

수도 있다. 일부 예들에서, 비디오 코더는 또한 픽셀의 픽셀 값의 디블록 필터링할 양을 결정하기 위해 디블록 필터링될 픽셀들의 수를 이용할 수도 있다.

[0019] 일부의 경우, 블록 경계를 따른 블록킹 아티팩트는 상대적으로 큰 블록 사이즈들 (예컨대, 일 예로서 32x32)에 대해 현저할 수도 있다. 예를 들어, 비디오 코더는 블록을 코딩하는 것의 일부로서 변환 유닛을 결정한다.

변환 유닛의 블록 사이즈가 상대적으로 크면, 블록킹 아티팩트들이 현저할 수도 있다. HEVC 비디오 코딩 표준 및 다른 제안들에서의 기법들과 같은, 현재의 기법들은 블록킹 아티팩트들을 평활화하는 것이 존재한다; 그러나, 이러한 기법들은 큰 블록 사이즈들에 대한 블록킹 아티팩트들을 적절히 평활화할 수 없거나 또는 블록킹 아티팩트들의 적절한 평활화를 달성하기 위해 계산 및 구현 복잡성을 크게 증가시킬 수도 있다.

[0020] 본 개시물에서 설명하는 기법들은 기하학적인 거리 및 픽셀 값 차이에 기초하여 블록킹 아티팩트를 평활화함으로써 시각적 품질을 향상시킬 수도 있다. 예를 들어, 이 기법들은 픽셀 거리에 따라서 필터링하기 전 픽셀 값을 유지할 수도 있으며, 여기서, 픽셀 거리는 기하학적인 거리 및 픽셀 값 차이의 함수이다. 비디오 코더는 또한 프로세싱 및 구현을 촉진하는 일부 다른 기법들로서, 더 적은 개수의 픽셀들을 이용하여 디블록 필터링을 수행 가능할 수도 있다.

[0021] 도 1은 본 개시물의 예들에 따른 기법들을 이용하도록 구성될 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10)을 예시하는 블록도이다. 도 1에 나타낸 바와 같이, 시스템 (10)은 인코딩된 비디오를 통신 채널 (16)을 통해서 목적지 디바이스 (14)로 송신하는 소스 디바이스 (12)를 포함한다. 인코딩된 비디오 데이터는 또한 저장 매체 (34) 또는 파일 서버 (36) 상에 저장될 수도 있으며, 원하는 바에 따라 목적지 디바이스 (14)에 의해 액세스될 수도 있다. 저장 매체 또는 파일 서버에 저장될 때, 비디오 인코더 (20)는 코딩된 비디오 데이터를 저장 매체에 저장하기 위해, 코딩된 비디오 데이터를 네트워크 인터페이스, 컴팩트 디스크 (CD), 블루-레이 또는 디지털 비디오 디스크 (DVD) 버너 또는 스템핑 기능 디바이스, 또는 다른 디바이스들과 같은, 또 다른 디바이스에 제공할 수도 있다. 이와 유사하게, 네트워크 인터페이스, CD 또는 DVD 리더 등과 같은, 비디오 디코더 (30)와는 별개인 디바이스가 저장 매체로부터 코딩된 비디오 데이터를 취출하고, 그 취출된 데이터를 비디오 디코더 (30)에 제공할 수도 있다.

[0022] 소스 디바이스 (12) 및 목적지 디바이스 (14)는 데스크탑 컴퓨터들, 노트북 (즉, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 전화기 핸드셋들, 예컨대 소위 스마트폰들, 텔레비전, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 무선 핸드셋 디바이스 등을 포함한, 매우 다양한 디바이스들 중 임의의 디바이스를 포함할 수도 있다. 많은 경우에, 이런 디바이스들은 무선 통신용으로 탑재될 수도 있다. 그러므로, 통신 채널 (16)은 인코딩된 비디오 데이터의 송신에 적합한, 무선 채널, 유선 채널, 또는 무선 채널과 유선 채널의 조합을 포함할 수도 있다. 이와 유사하게, 파일 서버 (36)는 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해서 목적지 디바이스 (14)에 의해 액세스될 수도 있다. 이것은 파일 서버 상에 저장되는 인코딩된 비디오 데이터에 액세스하는데 적합한, 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양쪽의 조합을 포함할 수도 있다.

[0023] 본 개시물의 예들에 따른, 기법들은, 오버-디-에어 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상에의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들 중 임의의 애플리케이션의 지원 하에, 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10)은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화 통신과 같은, 지원 애플리케이션들로의 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0024] 일부 예들에서, 본 개시물에서 설명하는 기법들은 특히 HEVC (High Efficiency Video Coding) 표준에 따른 비디오 코딩에 유용할 수도 있다. 그러나, 본 개시물에서 설명되는 기법들은 임의의 특정의 비디오 코딩 표준에 한정되는 것으로 간주되어서는 안된다. 본 개시물에서 설명하는 기법들은 비-표준들 기반의 비디오 코딩에도 또한 적용 가능할 수도 있다.

[0025] 비디오 코딩 표준들은 ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 또는 ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual 및 그의 스케일러블 비디오 코딩 (SVC) 및 멀티뷰 비디오 코딩 (MVC) 확장판들을 포함한, (또한, ISO/IEC MPEG-4 AVC로서 알려진) ITU-T H.264를 포함한다. MVC의 최신 협동 초안은 2010년 3월, ITU-T 권고안 H.264, "Advanced Video Coding for generic audiovisual services"에 설명되어 있다.

- [0026] 게다가, 새로 개발된 비디오 코딩 표준, 즉, ITU-T 비디오 코딩 전문가 그룹 (VCEG) 와 ISO/IEC 동화상 전문가 그룹 (MPEG) 의 비디오 코딩에 관한 합동 연구팀 (JCT-VC) 에 의해 개발된, 고효율 비디오 코딩 (HEVC) 이 있다. "HEVC 작업 초안 10" 또는 "WD10" 로서 지정되는, ITU-T SG16 WP3 와 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 12차 회의: 2013년 1월 14-23 일, 스위스, 제네바, 문서 JCTVC-L1003v34, Bross 등이, "High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Last Call)" 에 설명되어 있는 HEVC 의 최신 안은 http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip로부터 입수 가능하다.
- [0027] HEVC 는 ITU-T WP3/16 및 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding) 에 의해 최근 개발된 비디오 코딩을 위한 국제 표준이다. HEVC 의 범위 확장판은 YCbCr 4:2:0, 예컨대 YCbCr 4:2:2, YCbCr 4:4:4, 및 RGB 이외의 칼라 공간들에 대한 비디오 코딩을 확장하는 것을 포함한다. 범위 확장판 초안은 2014년 1월, D. Flynn, M. Naccari, C. Rosewarne, J. Sole, G. Sullivan, 및 T. Suzuki, "High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 3", ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11, 문서 JCTVC-P1005 이다. 게다가, 새로운 표준화 활동이 스크린 콘텐츠 코딩에 대한 코딩 성능을 향상시키기 위해 이루어졌다. 새로운 표준화에 대한 활동은 2013년 1월, Joint Call for Proposals for Coding of Screen Content, ITU-T Q6/16 및 ISO/IEC JTC1/SC29/WG11, 문서 MPEG2014/N14175 이다.
- [0028] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 변조기/복조기 (모뎀) (22) 및 송신기 (24) 를 포함한다. 소스 디바이스 (12) 에서, 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하는 비디오 공급 인터페이스, 및/또는 컴퓨터 그래픽스 데이터를 소스 비디오로서 발생하는 컴퓨터 그래픽스 시스템과 같은 소스, 또는 이런 소스들의 조합을 포함할 수도 있다. 일 예로서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 예컨대, 스마트폰들 또는 태블릿 컴퓨터들 내에 제공될 수도 있는, 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 본 개시물에서 설명하는 기법들은 일반적으로는 비디오 코딩에 적용 가능할 수도 있으며, 무선 및/또는 유선 애플리케이션들, 또는 인코딩된 비디오 데이터가 로컬 디스크 상에 저장되는 애플리케이션에 적용될 수도 있다.
- [0029] 캡처되거나, 사전-캡처되거나, 또는 컴퓨터-발생된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 유선 또는 무선 통신 프로토콜과 같은 통신 표준에 따라서 모뎀 (22) 에 의해 변조되어, 송신기 (24) 를 통해서 목적지 디바이스 (14) 로 송신될 수도 있다. 모뎀 (22) 은 여러 믹서들, 필터들, 증폭기들 또는 신호 변조용으로 설계된 다른 구성요소들을 포함할 수도 있다. 송신기 (24) 는 증폭기들, 필터들, 및, 무선 통신의 경우, 하나 이상의 안테나들을 포함한, 데이터를 송신하기 위해 설계된 회로들을 포함할 수도 있다.
- [0030] 비디오 인코더 (20) 에 의해 인코딩된, 캡처된, 사전-캡처된, 또는 컴퓨터-발생된 비디오는 또한 추후 소비를 위해 저장 매체 (34) 또는 파일 서버 (36) 상에 저장될 수도 있다. 저장 매체 (34) 는 블루-레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 또는 인코딩된 비디오를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들을 포함할 수도 있다. 저장 매체 (34) 상에 저장된 인코딩된 비디오는 그후 디코딩 및 플레이백을 위해 목적지 디바이스 (14) 에 의해 액세스될 수도 있다. 도 1 에 나타내지는 않지만, 일부 예들에서, 저장 매체 (34) 및/또는 파일 서버 (36) 는 송신기 (24) 의 출력을 저장할 수도 있다.
- [0031] 또한, 저장 매체 (34) 및 파일 서버 (36) 가 비디오 인코더 (20) 로부터 비디오 데이터를 수신하는 것으로 예시되지만, 일부 예들에서, 저장 매체 (34) 및 파일 서버 (36) 는 비디오 인코더 (20) 이외의, 송신기 (24) 또는 일부 다른 유닛으로부터 비디오 데이터를 수신할 수도 있다. 이와 유사하게, 저장 매체 (34) 및 파일 서버 (36) 가 비디오 데이터를 비디오 디코더 (30) 로 출력하는 것으로 예시되지만, 일부 예들에서, 저장 매체 (34) 및 파일 서버 (36) 는 비디오 데이터를 비디오 디코더 (30) 이외의, 수신기 (26) 또는 일부 다른 유닛으로 출력할 수도 있다. 이를 예들에서, 비디오 디코더 (30) 는 수신기 (26) 또는 일부 다른 유닛으로부터 비디오 데이터를 수신할 수도 있다.
- [0032] 파일 서버 (36) 는 인코딩된 비디오를 저장하고 그 인코딩된 비디오를 목적지 디바이스 (14) 로 송신하는 것이 가능한 임의 종류의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버 (예컨대, 웹사이트용), FTP 서버, 네트워크 부착된 스토리지 (NAS) 디바이스들, 로컬 디스크 드라이브, 또는 인코딩된 비디오 데이터를 저장하고 이를 목적지 디바이스로 송신할 수 있는 임의의 다른 유형의 디바이스를 포함한다. 파일 서버 (36) 로부터

의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이 양쪽의 조합일 수도 있다. 파일 서버 (36)는 목적지 디바이스 (14)에 의해, 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해서 액세스될 수도 있다. 이것은 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀, 이더넷, USB 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 이들 양쪽의 결합을 포함할 수도 있다.

[0033] 목적지 디바이스 (14)는, 도 1의 예에서, 수신기 (26), 모뎀 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 목적지 디바이스 (14)의 수신기 (26)는 통신 채널 (16)을 통해서 정보를 수신하고, 모뎀 (28)은 그 정보를 복조하여, 비디오 디코더 (30)를 위한 복조된 비트스트림을 발생한다. 통신 채널 (16)을 통해서 통신된 정보는 비디오 데이터를 디코딩할 때에 비디오 디코더 (30)에 의한 사용을 위해서 비디오 인코더 (20)에 의해 발생되는 다양한 신택스 정보를 포함할 수도 있다. 이런 신택스는 또한 저장매체 (34) 또는 파일 서버 (36)상에 저장된 비디오 데이터와 함께 포함될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 비디오 데이터를 인코딩 또는 디코딩할 수 있는 각각의 인코더-디코더 (코덱)의 일부를 형성할 수도 있다.

[0034] 디스플레이 디바이스 (32)는 목적지 디바이스 (14)와 통합되거나 또는 그 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14)는 통합된 디스플레이 디바이스를 포함하며, 또한 외부 디스플레이 디바이스와 인터페이스하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14)는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32)는 그 디코딩된 비디오 데이터를 사용자에게 디스플레이하며, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 디바이스를 포함할 수도 있다.

[0035] 도 1의 예에서, 통신 채널 (16)은 임의의 무선 또는 유선 통신 매체, 예컨대 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들, 또는 무선 매체와 유선 매체들의 임의의 조합을 포함할 수도 있다. 통신 채널 (16)은 근거리 네트워크, 광역 네트워크, 또는 글로벌 네트워크, 예컨대 인터넷과 같은 페킷-기반 네트워크의 일부를 형성할 수도 있다. 통신 채널 (16)은 일반적으로 유선 또는 무선 매체들의 임의의 적합한 조합을 포함한, 비디오 데이터를 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 송신하기 위한, 임의의 적합한 통신 매체, 또는 상이한 통신 매체들의 컬렉션을 나타낸다. 통신 채널 (16)은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12)로부터 목적지 디바이스 (14)로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0036] 비디오 인코더 (20) 및 비디오 디코더 (30)는 비디오 압축 표준, 예컨대 HEVC 및/또는 HEVC 표준의 확장판에 따라서 동작할 수도 있다. HEVC 범위 확장판들은 현재 개발 중이다. 더욱이, HEVC 및 그의 확장판들의 설명이 예시의 목적을 위해 제공되며, 한정하는 것으로 간주되지 않아야 한다. 일반적으로, 비디오 인코더 (20) 및 비디오 디코더 (30)는 비디오 코딩, 좀더 구체적으로는, 좀 더 자세하게 설명되는 바와 같은, 디블록 필터링에 관련된, 본 개시물에서 설명되는 예시적인 기법들을 수행하도록 구성될 수도 있다.

[0037] 도 1에 나타내지는 않지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 오디오 인코더 및 디코더와 각각 통합될 수도 있으며, 오디오 및 비디오 양쪽의 인코딩을 공통 데이터 스트림 또는 별개의 데이터 스트림들로 처리하기에 적합한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용 가능한 경우, 일부 예들에서, MUX-DEMUX 유닛들은 멀티플렉서 프로토콜을 따를 수도 있다. 예들은 ITU H.223 멀티플렉서 프로토콜, 또는 다른 프로토콜들, 예컨대, 사용자 데이터그램 프로토콜 (UDP)을 포함한다.

[0038] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그래밍 가능 게이트 어레이들 (FPGA들), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 임의의 이들의 조합들과 같은, 다양한 적합한 인코더 회로 중 임의의 회로로 구현될 수도 있다. 이 기법들이 소프트웨어로 부분적으로 구현되는 경우, 디바이스는 본 개시물의 기법들을 수행하기 위해 소프트웨어용 명령들을 적합한 비일시성 컴퓨터-판독가능 매체에 저장하고, 그 명령들을 하드웨어에서 하나 이상의 프로세서들을 이용하여 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 쪽이든 개별 디바이스에서 결합된 인코더/디코더 (CODEC)의 부분으로서 통합될 수도 있다.

[0039] 비디오 디코더 (30)를 포함하는 디바이스의 예들은 무선 통신 디바이스, 집적 회로 (IC), 및 마이크로프로세서를 포함한다. 이와 유사하게, 비디오 인코더 (20)를 포함하는 디바이스의 예들은 무선 통신 디바이스, IC, 및 마이크로프로세서를 포함한다.

[0040]

일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 각각 본 개시물에서 설명하는 기법들을 구현하도록 구성된 하나 이상의 프로세서들을 포함할 수도 있다. 비디오 인코더 (20) 또는 비디오 디코더 (30)를 포함하는 디바이스는 또한 비디오 인코더 (20) 및 비디오 디코더 (30)가 이용하는 비디오 데이터를 저장하는 메모리를 포함할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)를 포함하는 디바이스가 IC 칩 또는 마이크로프로세서인 예들에서, 메모리는 IC 칩 또는 마이크로프로세서의 로컬 메모리일 수도 있거나, 또는 IC 칩 또는 마이크로프로세서를 포함하는 장치 (예컨대, 스마트-폰 또는 태블릿과 같은 무선 통신 디바이스)의 시스템 메모리, 또는 로컬 메모리와 시스템 메모리의 조합일 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)를 포함하는 디바이스가 무선 통신 디바이스 (또는, 다른 이러한 디바이스)인 예들에서, 메모리는 비디오 인코더 (20) 또는 비디오 디코더 (30)의 로컬 메모리, 비디오 인코더 (20) 또는 비디오 디코더 (30)를 포함하는 디바이스의 시스템 메모리, 또는 로컬 메모리와 시스템 메모리의 조합일 수도 있다.

[0041]

본 개시물에 따르면, 비디오 인코더 (20)는 디블록 필터링을 위한 본 개시물의 기법들 중 임의의 기법 또는 모두를 구현할 수도 있다. 이와 유사하게, 비디오 디코더 (30)는 디블록 필터링을 위한 이들 기법들 중 임의의 기법 또는 모두를 구현할 수도 있다. 비디오 코더는, 본 개시물에서 설명하는 바와 같이, 비디오 인코더 또는 비디오 디코더를 지칭할 수도 있다. 이와 유사하게, 비디오 코딩 유닛은 비디오 인코더 또는 비디오 디코더를 지칭할 수도 있다. 이와 유사하게, 비디오 코딩은 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30)가 본 개시물에서 설명되는 예시적인 기법들을 각각 수행할 수도 있기 때문에, 용어 "비디오 코더"는 비디오 인코더 또는 비디오 디코더를 포괄적으로 나타내기 위해 사용된다. 특정의 인코딩 기법들이 호출되는 본 개시물의 부분들에서, 비디오 코더는 비디오 인코더 (20)의 일 예를 나타내며, 그리고, 특정의 디코딩 기법들이 호출되는 본 개시물의 부분들에서, 비디오 코더는 비디오 디코더 (30)의 일 예를 나타낸다. 용어 "코드" 또는 "코딩"은 또한 인코딩한다/디코딩한다 또는 인코딩/디코딩 각각을 포괄적으로 지칭하기 위해 사용된다.

[0042]

디지털 비디오 디바이스들은 디지털 비디오 정보를 좀 더 효율적으로 인코딩하고 디코딩하는 비디오 압축 기법들을 구현한다. 비디오 압축은 비디오 시퀀스들에 고유한 리던던시를 감소시키거나 또는 제거하기 위해 공간 (인트라-프레임) 예측 및/또는 시간 (인터-프레임) 예측 기법들을 적용할 수도 있다.

[0043]

HEVC-규격 비디오 디코딩 디바이스는 이전 발생 디바이스들 (예컨대, ITU-T H.264/AVC 디바이스들)에 비해 여러 추가적인 능력들을 지원한다. 예를 들어, H.264는 9개의 인트라-예측 인코딩 모드들을 제공하는 반면, HEVC는 35개의 만큼이나 많은 인트라-예측 인코딩 모드들을 지원한다. 다음 섹션은 HEVC의 어떤 양태들을 좀 더 자세하게 설명한다.

[0044]

HEVC 표준에 따른 비디오 코딩에 있어, 비디오 프레임은 코딩 유닛들로 파티셔닝될 수도 있다. 코딩 유닛 (CU)은 일반적으로 여러 코딩 틀들이 비디오 압축을 위해 적용되는 기본 유닛으로서 역할을 하는 이미지 영역을 지칭한다. CU는 대개 Y로 표기되는 희도 성분, 및 U와 V로 표기되는 2개의 크로마 성분들을 갖는다. 비디오 샘플링 포맷에 따라서, U 및 V 성분들의 사이즈는, 샘플들의 개수의 관점에서, Y 성분의 사이즈와 동일하거나 또는 상이할 수도 있다.

[0045]

CU는 일반적으로 정사각형이며, 예컨대, ITU-T H.264와 같은, 다른 비디오 코딩 표준들 하에서 소위 매크로블록과 유사한 것으로 간주될 수도 있다. HEVC 표준의 현재 제안된 양태들의 일부에 따른 코딩이 본 출원 예시의 목적을 위해 설명될 것이다. 그러나, 본 개시물에서 설명하는 기법들은 HEVC, H.264 또는 다른 표준의 범위 확장판에 따라서 정의된 프로세스들, 또는 사유 비디오 코딩 프로세스들과 같은, 다른 비디오 코딩 프로세스들에 유용할 수도 있다.

[0046]

HEVC에 따르면, CU는 하나 이상의 예측 유닛들 (PU들) 및/또는 하나 이상의 변환 유닛들 (TU들)을 포함할 수도 있다. 비트스트림 내 신팩스 데이터는 픽셀들의 개수의 관점에서 최대 CU인 최대 코딩 유닛 (LCU)을 정의한다. 일반적으로, CU가 사이즈 구별을 갖지 않는다는 점을 제외하고는, CU는 H.264의 매크로블록과 유사한 목적을 갖는다. 따라서, CU는 서브-CU들로 분할될 수도 있다. 일반적으로, 본 개시물에서 CU에 대한 언급들은 픽처의 최대 코딩 유닛 또는 LCU의 서브-CU를 지칭할 수도 있다. LCU는 서브-CU들로 분할될 수도 있으며, 각각의 서브-CU는 서브-CU들로 추가로 분할될 수도 있다. 비트스트림에 대한 신팩스 데이터는 LCU가 분할되는 최대 횟수를 정의할 수도 있으며, 이는 CU 깊이로 지칭된다. 따라서, 비트스트림은 또한 최소 코딩 유닛 (SCU)을 정의할 수도 있다. 본 개시물은 또한 CU, PU, 또는 TU 중 임의의 것을 지칭하기 위해 용어 "블록" 또는 "부분"을 사용한다. 일반적으로, "부분"은 비디오 프레임의 임의의 서브-세트를 지칭할 수도 있다.

[0047]

LCU 는 큐드트리 데이터 구조와 연관될 수도 있다. 일반적으로, 큐드트리 데이터 구조는 CU 당 하나의 노드를 포함하며, 여기서, 루트 노드는 LCU 에 대응한다. CU 가 4개의 서브-CU들로 분할되면, CU 에 대응하는 노드는 4개의 리프 노드들을 포함하며, 그 리프 노드 각각은 서브-CU들 중 하나에 대응한다. 큐드트리 데이터 구조의 각각의 노드는 대응하는 CU 에 대한 선택스 데이터를 제공할 수도 있다. 예를 들어, 큐드트리에 서의 노드는 그 노드에 대응하는 CU 가 서브-CU들로 분할되는지의 여부를 나타내는 분할 플래그를 포함할 수도 있다. CU 에 대한 선택스 엘리먼트들은 회귀적으로 정의될 수도 있으며, CU 가 서브-CU들로 분할되는지의 여부에 의존할 수도 있다. CU 가 추가로 분할되지 않으면, 리프-CU 로서 지칭된다. 본 개시물에서, 리프-CU 의 4 개의 서브-CU들은 또한 어떤 원래 리프-CU 의 명시적인 분할이 존재하지 않더라도, 리프-CU들로서 지칭될 것이다. 예를 들어, 16x16 사이즈에서의 CU 가 추가로 분할되지 않으면, 4개의 8x8 서브-CU들은 또한 16x16 CU 가 전혀 분할되지 않았더라도 리프-CU들로서 지칭될 것이다.

[0048]

리프-CU 는 하나 이상의 PU들을 포함할 수도 있다. 일반적으로, PU 는 대응하는 CU 의 모두 또는 부분을 나타내며, PU 에 대한 참조 샘플을 취출하기 위한 데이터를 포함할 수도 있다. 예를 들어, PU 가 인터-모드 인코딩될 때, PU 는 그 PU 에 대한 모션 벡터를 정의하는 데이터를 포함할 수도 있다. 모션 벡터를 정의하는 데이터는 예를 들어, 모션 벡터의 수평 성분, 모션 벡터의 수직 성분, 모션 벡터에 대한 해상도 (예컨대, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도), 모션 벡터가 가리키는 참조 프레임, 및/또는 모션 벡터에 대한 참조 리스트 (예컨대, 리스트 0 또는 리스트 1) 을 기술할 수도 있다. PU(들) 을 정의하는 리프-CU 에 대한 데이터는 또한 예를 들어, 하나 이상의 PU들로의 CU 의 파티셔닝을 기술할 수도 있다. 파티셔닝 모드들은 CU 가 인코딩되지 않거나, 인트라-예측 모드 인코딩되거나, 또는 인터-예측 모드 인코딩되는지의 여부에 따라서 상이 할 수도 있다. 인트라 코딩에 있어, PU 는 아래에서 설명되는 리프 변환 유닛과 동일하게 처리될 수도 있다.

[0049]

HEVC 표준은 변환 유닛들 (TU들) 에 따른 변환들을 가능하게 하며, 이 TU들은 상이한 CU들에 대해 상이할 수도 있다. TU들은 일반적으로 파티셔닝된 LCU 에 대해 정의되는 주어진 CU 내 PU들의 사이즈에 기초하여 사이징 되지만, 이것이 항상 그런 것은 아니다. TU들은 일반적으로 PU들과 동일한 사이즈이거나 또는 그보다 작다. 일부 예들에서, CU 에 대응하는 잔차 샘플들은 "잔차 큐드 트리" (RQT) 로서 알려진 큐드트리 구조를 이용하여 더 작은 유닛들로 세분될 수도 있다. RQT 의 리프 노드들은 변환 유닛들 (TU들) 로서 지칭될 수도 있다. TU들과 연관되는 픽셀 차이 값들은 변환 계수들을 발생하기 위해 변환될 수도 있으며, 그 변환 계수들은 양자화될 수도 있다. TU 는 루마 변환 블록 및 2개의 크로마 변환 블록들을 포함한다. 이와 같이, TU 에 적용되는, 아래에 설명되는 임의의 코딩 프로세스는 실제로, 루마 및 크로마 변환 블록들에 적용될 수도 있다.

[0050]

일반적으로, PU 는 예측 프로세스에 관련된 데이터를 지칭한다. 예를 들어, PU 가 인트라-모드 인코딩될 때, PU 는 그 PU 에 대한 인트라-예측 모드를 기술하는 데이터를 포함할 수도 있다. 또 다른 예로서, PU 가 인터-모드 인코딩될 때, PU 는 그 PU 에 대한 모션 벡터를 정의하는 데이터를 포함할 수도 있다.

[0051]

일반적으로, TU 는 변환 및 양자화 프로세스들에 사용된다. 하나 이상의 PU들을 갖는 주어진 CU 는 또한 하나 이상의 TU들을 포함할 수도 있다. 예측 이후, 비디오 인코더 (20) 는 PU 에 따라서 코딩 노드에 의해 식별되는 비디오 블록으로부터 잔차 값들을 계산할 수도 있다. 코딩 노드는 그후 원래 비디오 블록 대신, 잔차 값들을 참조하기 위해 업데이트된다. 잔차 값들은 엔트로피 코딩을 위한 직렬화된 변환 계수들을 발생하기 위해 TU들에 규정된 변환들 및 다른 변환 정보를 이용하여, 변환 계수들로 변환되고, 양자화되고, 그리고 스캐닝될 수도 있는 픽셀 차이 값들을 포함한다. 코딩 노드는 또 다시 이를 직렬화된 변환 계수들을 참조하기 위해 업데이트될 수도 있다. 본 개시물은 일반적으로 CU 의 코딩 노드를 지칭하기 위해 용어 "비디오 블록" 을 사용한다. 일부 특정의 경우들에서, 본 개시물은 또한 트리블록, 즉, LCU, 또는 코딩 노드 및 PU들 및 TU들을 포함하는 CU 를 지칭하기 위해 용어 "비디오 블록" 을 이용할 수도 있다.

[0052]

비디오 시퀀스는 일반적으로 비디오 프레임들 또는 픽처들의 시리즈를 포함한다. 픽처들의 그룹 (GOP) 은 일반적으로 비디오 픽처들의 하나 이상의 시리즈를 포함한다. GOP 는 GOP 의 헤더에, 픽처들의 하나 이상의 헤더에, 또는 다른 곳에, GOP 에 포함된 다수의 픽처들을 기술하는 선택스 데이터를 포함할 수도 있다. 픽처의 각각의 슬라이스는 개별 슬라이스에 대한 인코딩 모드를 기술하는 슬라이스 선택스 데이터를 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로 비디오 데이터를 인코딩하기 위해 개개의 비디오 슬라이스들 내 비디오 블록들에 대해 동작한다. 비디오 블록은 CU 내 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정 또는 가변 사이즈들을 가질 수도 있으며, 규정된 코딩 표준에 따라서 사이즈가 상이할 수도 있다.

[0053]

블록 (예컨대, 비디오 데이터의 예측 유닛) 을 코딩하기 위해, 그 블록에 대한 예측자가 먼저 유도된다. 예

즉 블록으로 또한 지칭되는 예측자는 인트라 (I) 예측 (즉, 공간 예측) 또는 인터 (P 또는 B) 예측 (즉, 시간 예측)을 통해서 유도될 수 있다. 그러므로, 일부 예측 유닛들은 동일한 프레임 (또는, 슬라이스)에서 이 웃하는 참조 블록들에서의 참조 샘플들에 대해 공간 예측을 이용하여 인트라-코딩될 수도 있으며 (I), 다른 예측 유닛들은 다른 이전에-코딩된 프레임들 (또는, 슬라이스들)에서의 참조 샘플들의 블록들에 대해 단방향 인터-코딩되거나 (P) 또는 양방향 인터-코딩될 수도 있다 (B). 각 경우에, 참조 샘플들은 코딩될 블록에 대한 예측 블록을 형성하는데 사용될 수도 있다.

[0054] 예측 블록의 식별 시, 원래 비디오 데이터 블록에서의 픽셀들과 그의 예측 블록에서의 픽셀들 사이의 차이가 결정된다. 이 차이는 예측 잔차 데이터로서 지정될 수도 있으며, 코딩되는 블록에서의 픽셀 값들과, 코딩된 블록을 나타내기 위해 선택된 예측 블록에서의 픽셀 값들 사이의 픽셀 차이들을 나타낸다. 더 나은 압축을 달성하기 위해, 예측 잔차 데이터는 변환 계수들을 발생하기 위해 예컨대, 이산 코사인 변환 (DCT), 정수 변환, Karhunen-Loeve (K-L) 변환, 또는 또 다른 변환을 이용하여 변환될 수도 있다.

[0055] TU 와 같은 변환 블록에서의 잔차 데이터는 공간, 픽셀 도메인에 상주하는 픽셀 차이 값들의 2차원 (2D) 어레이로 배열될 수도 있다. 변환은 잔차 픽셀 값들을 주파수 도메인과 같은 변환 도메인에서의 변환 계수들의 2차원 어레이로 변환한다.

[0056] 추가적인 압축을 위해, 변환 계수들은 엔트로피 코딩 이전에 양자화될 수도 있다. 엔트로피 코더는 그후 그 양자화된 변환 계수들에, 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 2진 산술 코딩 (CABAC), 확률 간격 파티셔닝 엔트로피 코딩 (PIPE) 등과 같은 엔트로피 코딩을 적용한다. 일부 예들에서, 비디오 인코더 (20)는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 발생하기 위해, 미리 정의된 스캐닝 순서를 이용하여, 양자화된 변환 계수들을 스캐닝할 수도 있다. 다른 예들에서, 비디오 인코더 (20)는 적응적 스캐닝을 수행할 수도 있다. 양자화된 변환 계수들을 스캐닝하여 1차원 벡터를 형성한 후, 비디오 인코더 (20)는 1차원 벡터를 엔트로피 인코딩할 수도 있고, 비디오 인코더 (20)는 또한 비디오 데이터를 디코딩할 때에 비디오 디코더 (30)에 의해 사용하기 위한 인코딩된 비디오 데이터와 연관되는 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0057] 산술 코딩은 심볼들을 비-정수 길이 코드워드들에 맵핑하는 것이 가능하기 때문에 높은 코딩 효율을 가진 많은 압축 알고리즘들에서 사용되는 엔트로피 코딩의 형태이다. CABAC 코딩은 일부 정보에 대해 적용되고 다른 정보에 대해서는 바이패스될 수도 있다.

[0058] 비디오 디코더 (30)는 비디오 인코더 (20)의 프로세스의 역을 수행하여 블록을 디코딩한다. 예를 들어, 비디오 디코더 (30)는 비트스트림으로부터 수신된 계수 데이터로부터 TU 를 결정하고, 비트스트림으로부터의 모션 벡터 정보 또는 인트라-코딩 모드 정보로부터 예측자를 식별한다. 비디오 디코더 (30)는 그후 예측자를 TU 에 가산하여 블록을 발생시킨다.

[0059] 위에서 설명된 바와 같이, 비디오 인코더 (20) 및 비디오 디코더 (30)는 본 개시물에서 설명되는 예시적인 디블록 필터링 기법들을 구현하도록 구성될 수도 있다. 예를 들어, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30))는 디블록 필터링을 수행하여, 블록 경계들을 따라서 블록킹 아티팩트를 제거할 수도 있다. 블록킹 아티팩트들은 뷰어가 비디오 블록들의 경계들을 볼 수 있는 비디오 콘텐츠 결함들을 지칭한다. 디블록 필터링은 뷰어가 비디오 블록들을 인지하지 않도록 (즉, 하나의 블록으로부터 다른 블록으로의 비디오 콘텐츠에서의 어떤 갑작스러운 변화도 존재하지 않도록) 경계에 가까운 픽셀들의 픽셀 값들을 평활화한다.

[0060] 그러나, 디블록 필터링은 비디오 콘텐츠를 또한 과-평활화하지 않아야 하는데, 그런 과-평활화가 시작적 품질을 열화시킬 수 있기 때문이다. 더욱이, 어떤 경우들에서, 비디오 콘텐츠 자체는 갑작스러운 변화들을 포함할 수도 있으며, 디블록 필터링은 이러한 변화들을 과도하게 평활화하지 않아야 한다. 예를 들어, 실제 이미지 에지들 (예컨대, 실례로서, 백색 백그라운드 상에 블랙 포어그라운드 오브젝트로 형성된 에지)에 있어서, 디블록 필터링은 이러한 실제 이미지 에지들에 대해 평활화하지 않아야 한다.

[0061] 현재, HEVC 표준은 큰 블록 사이즈 변환 (예컨대, 상대적으로 큰 TU 사이즈들)을 이용하는데, 이는 블록 경계들을 따라서 현저한 블록킹 아티팩트를 초래할 수 있다. HEVC 는 강한 필터링 방법을 가지는 인-루프 디블록 필터링 방식을 포함한다. 그러나, 블록 사이즈가 클 때 심한 블록킹 아티팩트를 처리하는데 비효율적이라는 점이 주목되어 왔다.

[0062] 이 이슈를 해결하기 위해, 많은 제안들이 HEVC 에 대해 이루어졌다. 2011년 11월, Z. Shi, X. Sun, 및 J.

Xu, "CE12, Subset 1: Report of deblocking for large size blocks", ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11, 문서 JCTVC-G409에서, 더 긴 필터가 더 큰 변환 블록의 경계를 평활화하기 위해 제안되었다.

2012년 10월, G. Van der Auwera, R. Joshi, 및 M. Karczewicz, "Non-CE1: Deblocking of Large Block Artifacts", ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11, 문서 JCTVC-K138; 2012년 10월, S. Lu, O. Nakagami, 및 T. Suzuki, "CE1: On deblocking filter", ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11, 문서 JCTVC-K149; 2012년 10월, S. Lu, O. Nakagami, 및 T. Suzuki, "NonCE1: Simple improvement of Deblocking filter", ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11, 문서 JCTVC-K150; 및 2012년 10월, D.-K. Kwon 및 M. Budagavi, "Non-CE1: Suppression of blocking artifacts at large TU boundaries", ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11, 문서 JCTVC-K269에서, 디블록킹 필터링 파라미터들이 큰 변환 블록 (예컨대, TU) 사이즈에 대해 조정된다. 이들 기법들은 더 큰 변환 블록들의 경계들을 따라서 블록킹 아티팩트를 감소시키는 것을 도울 수 있다. 그러나, HEVC에서 어떤 변화도 이루어지지 않았다.

[0063] 2013년 2월, 구글 VP9, IETF, 네트워크 작업 그룹, A. Grange 및 H. Alvestrand, "A VP9 Bitstream Overview"에서, 긴 필터가 블록킹 아티팩트를 감소시키기 위해 사용되는데, 이는 "CE12, Subset 1: Report of deblocking for larger size blocks" 와 유사하다. "CE12, Subset 1: Report of deblocking for larger size blocks" 및 "Non-CE1: Suppression of blocking artifacts at large TU boundaries" 양쪽이 큰 변환 블록들의 경계들을 따라서 블록킹 아티팩트를 감소시킬 수 있다. 그러나, 이들 기법들은 블록 경계들을 따라서 많은 픽셀들에 액세스하는 것을 필요로 하는데, 이는 계산 및 구현 복잡성을 증가시킨다. 또한, 그들 중에서 필터들은 아주 강한 저역 통과 필터들이며, 이들은 시각적 품질 열화를 초래하는 과-평활화를 수행할 수 있다. 한편, "Non-CE1: Deblocking of Large Block Artifacts", "CE1: On deblocking filter", "NonCE1: Simple improvement of Deblocking filter", 및 "Non-CE1: Suppression of blocking artifacts at large TU boundaries"에서 제안된 바와 같은 디블록킹 필터 파라미터들을 수정하는 것에 의한 성능이, 경계들을 따라서 수정된 픽셀들의 개수가 충분하지 않기 때문에 제한될 수 있으며, 일부 경우들에서, 이들 방법들은 시각적 품질을 열화시키는 실제 이미지 에지들의 평활화를 초래할 수 있다.

[0064] 본 개시물에서 설명되는 예시적인 기법들은 디블록 필터링에 대해 위에서 설명된 이슈들 중 하나 이상을 극복할 수도 있다. 예를 들어, 본 개시물은 블록킹 아티팩트를 감소시키는 새로운 필터 설계를 설명한다. 본 개시물에서 설명하는 기법들은 픽셀 거리 (예컨대, 정의된 픽셀 거리)에 따라서 필터링하기 이전 픽셀 값을 유지함으로써 시각적 품질을 향상시킨다. 또한, 프로세싱 및 구현을 촉진시키는, "CE12, Subset 1: Report of deblocking for large size blocks" 및 "Non-CE1: Suppression of blocking artifacts at large TU boundaries"에서 설명된 것들과 같은, 위에서 설명된 다른 기법들 중 적어도 일부, 아니면, 모두에 비해, 필터링에 더 적은 픽셀들의 개수를 필요로 한다.

[0065] 일부 예들에서, 비디오 인코더 (20)는 인코딩 중인 블록의 복원의 일부로서 디블록 필터링을 수행할 수도 있다. 예를 들어, 위에서 설명한 바와 같이, 비디오 인코더 (20)는 잔차 데이터를 포함하는 TU를 결정한다. TU는 그후 변환되고 양자화된다. 게다가, 비디오 인코더 (20)는 블록을 역양자화하고 역변환하여, TU를 복원한다. 비디오 인코더 (20)는 예측자를 복원된 TU에 가산하여, 복원된 블록을 발생시킨다. 본 개시물에서 설명하는 기법들에서, 비디오 인코더 (20)는 본 개시물에서 설명하는 기법들을 이용하여, 복원된 블록을 디블록 필터링하여, 비디오 인코더 (20)가 그 블록이 후속 블록의 인터- 또는 인트라-예측 인코딩에 사용될 수 있도록 디코딩된 픽처 버퍼 (DPB)에 저장하는 필터링된 복원된 블록을 발생시킬 수도 있다.

[0066] 비디오 디코더 (30)는 블록을 발생시키는 부분으로서 디블록 필터링을 수행할 수도 있다. 예를 들어, 비디오 디코더 (30)는 블록의 계수들을 수신하고, 블록을 역양자화하고 역변환하여 TU를 복원한다. 비디오 디코더 (30)는 예측자를 복원된 TU에 가산하여, 복원된 블록을 발생시킨다. 비디오 디코더 (30)는 본 개시물에서 설명하는 기법들을 이용하여, 복원된 블록을 디블록 필터링하여, 비디오 디코더 (30)가 블록이 후속 블록의 인터- 또는 인트라-예측 디코딩에, 그리고 디스플레이를 위해 디코딩된 비디오의 출력에, 사용될 수 있도록 그의 DPB에 저장하는 필터링된 복원된 블록을 발생시킨다.

[0067] 본 개시물에서 설명하는 기법들에서, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30))는 제 1 블록의 제 1 픽셀로부터 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정할 수도 있다. 비디오 코더는 제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정할 수도 있다. 비디오 코더는 기하학적인 거리 및 픽셀 값 차이에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링할 양을 결정하고, 디블록 필터링할 양에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링할 수도 있다. 디블록 필터

링 기법들의 예들이 도 4에 예시된 예와 함께 제공된다.

[0068] 도 4는 블록 경계를 가로지르는 픽셀 위치들에 대한 픽셀 값들의 일 예를 예시하는 개념도이다. 예를 들어, 도 4는 블록 (84A) 및 블록 (84B)을 예시한다. 도 4는 일 예를 예시하며, 여기서 "|"가 블록 (84A)과 블록 (84B) 사이의 블록 경계 (86)이다. 블록 경계 (86)는 2개의 이웃하는 TU들 사이의 경계일 수도 있지만, 일부 예들에서, 2개의 이웃하는 PU들 사이의 경계일 수도 있다. 본 예에서, 경계 (86)가 2개의 TU들 사이의 경계인 경우, 블록 (84A) 및 블록 (84B)은 양쪽 다 TU들일 수도 있다. 본 예에서, 경계 (86)가 2개의 PU들 사이의 경계인 경우, 블록 (84A) 및 블록 (84B)은 양쪽 다 PU들일 수도 있다.

[0069] 경계 (86)는 수평 블록 경계 또는 수직 블록 경계일 수 있다. 예의 목적들을 위해, 경계 (86)는 수직 블록 경계로서 예시된다. 그러나, 예시적인 기법들은, 경계가 수평인 경우들에 동일하게 적용가능하다.

[0070] 블록 (84B)은 코딩되고 있는 (예컨대, 인코딩되거나 또는 디코딩되고 있는) 현재의 블록일 수도 있으며, 블록 (84A)은 이전에 코딩된 블록일 수도 있다. 예를 들어, 블록 (84A)은 비디오 코더의 DPB에 저장될 수도 있다. 이 예에서, 블록 (84B)은 디블록 필터링 전의 복원된 블록 (예컨대, 복원된 TU 플러스 예측자)일 수도 있다.

[0071] 또한, 이 예에서, 블록 (84A)의 일부는 이미 디블록 필터링되었지만, 블록 (84A)의 우측-끝 경계는 아직 디블록 필터링되지 않았을 수도 있다 (예컨대, 픽셀들 p(0) 내지 p(N-1)). 예를 들어, 비디오 코더는 좌측면 상에서 블록 (84A)에 이웃하는 블록의 우측-끝 경계에 가까운 픽셀들과 함께, 블록 (84A)의 좌측-끝 경계에 가까운 픽셀들을 디블록 필터링하였을 수도 있다. 비디오 코더는 그 결과적인 블록을 블록 (84A)으로서 비디오 코더의 DPB에 저장하고, 이제 블록 (84A)을 취출하여 블록 (84B)의 좌측-끝 경계에 가까운 필터 픽셀들 (예컨대, 픽셀들 q(0) 내지 q(N-1))을 디블록 필터링하였을 수도 있다. 이와 유사하게, 비디오 코더가 블록 (84B)의 우측에 있는 블록을 코딩할 때, 비디오 코더는 블록 (84B)의 우측-끝 경계에 가까운 픽셀들을 디블록 필터링할 것이다. 비디오 코더는 (예컨대, 경계가 수평 경계인 경우에 대해) 블록 (84B) 위에 그리고 아래에 있는 블록들에 기초하여 유사한 디블록 필터링을 수행할 수도 있다. 일부 예들에서, 디블록킹의 일부분으로서, 비디오 코더는 블록 (84A)의 일부 부분이 이미 디블록 필터링된 상황에서 필터 블록 (84A)의 픽셀들 p(0) 내지 p(N-1)을 디블록 필터링하고, 필터 블록 (84B)의 픽셀들 q(0) 내지 q(N-1)을 디블록 필터링 할 수도 있으며, 여기서, 블록 (84B)은 인코딩되거나 또는 디코딩되고 있는 현재의 블록이다.

[0072] 도 4에서, 블록 (84A)의 픽셀들은 픽셀 값들 p(0) 내지 p(N-1)을 각각 가지는 픽셀들 0 내지 N-1이며, 블록 (84B)의 픽셀들은 픽셀 값들 q(0) 내지 q(N-1)을 각각 가지는 픽셀들 0 내지 N-1이다. 블록에서의 픽셀의 픽셀 값은 루마 블록에서의 픽셀의 루마 값, 또는 크로마 블록들 중 하나에서의 픽셀의 크로마 값들 중 하나일 수도 있다. 예를 들어, p(0) 내지 p(N-1)은 블록 (84A)이 루마 블록인 예들에서는 픽셀의 루마 값들에, 또는 블록 (84A)이 크로마 블록들 중 하나인 예들에서는 픽셀의 크로마 값들 중 하나에 대응할 수도 있다. 이와 유사하게, q(0) 내지 q(N-1)은 블록 (84B)이 루마 블록인 예들에서는 픽셀의 루마 값들에, 또는 블록 (84B)이 크로마 블록들 중 하나인 예들에서는 픽셀의 크로마 값들 중 하나에 대응할 수도 있다.

[0073] 픽셀 값들 p(0) 내지 p(N-1)을 가지는 픽셀들은 동일한 로우에 있을 수도 있으며, 픽셀 값들 q(0) 내지 q(N-1)을 가지는 픽셀들은 픽셀 값들 p(0) 내지 p(N-1)을 가지는 픽셀들과 동일한 로우에 있을 수도 있다. 변수 "N"은 디블록 필터링될 픽셀들의 개수 (즉, 블록 (84A)에서 N-1개의 픽셀들 및 블록 (84B)에서 N-1개의 픽셀들)를 나타낸다.

[0074] 도 4에 예시된 바와 같이, 픽셀 값들 p(0) 내지 p(N-1)을 가지는 픽셀들은 모두 경계 (86)에 가까우며, 계속해서 경계 (86)로부터 이동한다. 예를 들어, 블록 (84A)의 픽셀 값 p(0)을 가지는 픽셀은 경계 (86)에 가장 가까우며, 블록 (84A)의 픽셀 값 p(1)을 가지는 픽셀은 다음으로 가장 가까우며, 및 블록 (84A)의 픽셀 값 p(N-1)을 가지는 픽셀은 경계 (86)로부터 가장 멀리 떨어져 있다. 이와 유사하게, 픽셀 값들 q(0) 내지 q(N-1)을 가지는 픽셀들은 모두 경계 (86)에 가까우며, 계속해서 경계 (86)로부터 멀리 이동한다. 예를 들어, 블록 (84B)의 픽셀 값 q(0)을 가지는 픽셀은 경계 (86)에 가장 가까우며, 블록 (84B)의 픽셀 값 q(1)을 가지는 픽셀은 다음으로 가장 가까우며, 블록 (84B)의 픽셀 값 q(N-1)을 가지는 픽셀은 경계 (86)로부터 가장 멀리 떨어져 있다. 또한, 픽셀 값 p(0)을 가지는 픽셀 및 픽셀 값 q(0)을 가지는 픽셀은 경계 (86)까지 등거리일 수도 있으며, 픽셀 값 p(1)을 가지는 픽셀 및 픽셀 값 q(1)을 가지는 픽셀은 경계 (86)까지 등거리일 수도 있으며, 기타 등등으로 등거리일 수도 있다.

[0075] 더 일반적으로는, 픽셀 값 p(i)를 가지는 픽셀과 경계 (86) 사이의 거리는 픽셀 값 q(i)를 가지는 픽셀과 경

계 (86) 사이의 거리와 동일하다. 다시 말해서, $p(i)$ 및 $q(i)$ 가 블록 경계 (86)에 대한 각각의 측면에서 위치 i 에서의 픽셀 값들을 표시한다고 하고, 여기서, i 는 경계 (86)로부터 가장 가까운 픽셀에 대해 0이며 위치가 경계 (86)로부터 멀리 이동함에 따라서 증가한다.

[0076] 도 4에 예시된 바와 같이, 블록 (84A) 및 블록 (84B)에서 각각 가장 가까운 픽셀들에 대한 픽셀 값인, 픽셀 값 $p(0)$ 과 $q(0)$ 사이에 갑작스러운 변화가 있다. 픽셀 값들 $p(0)$ 과 $q(0)$ 사이의 갑작스러운 변화는 블록 (84A)과 블록 (84B) 사이에 블록 아티팩트들을 초래할 수도 있다. 이러한 블록 아티팩트들은 비디오 콘텐츠의 시각적 품질에서의 감소를 초래한다. 본 개시물에서 설명되는 디블록 필터링 기법들은 시각적 품질을 열화시키는 과-평활화를 수행함이 없이 계산 효율적인 방법으로 이러한 블록 아티팩트들을 감소시킬 수도 있다.

[0077] 일반적으로, 비디오 코더는 적어도 2개의 픽셀들 사이의 픽셀 값 차이 및 기하학적인 거리에 기초하여 픽셀의 픽셀 값들을 디블록 필터링할 양을 결정한다. 본 개시물에서, 용어 "픽셀 거리"는 픽셀의 픽셀 값들을 디블록 필터링할 양으로서 정의되며, 픽셀 값 차이 및 기하학적인 거리의 함수이다.

[0078] 예를 들어, 픽셀 거리 l (예컨대, 디블록 필터링할 양을 결정하는데 사용되는 값)는 $l(i) = f(n(i), d(i))$ 로서 정의되며, 여기서 n 은 수정될 (예컨대, 디블록 필터링될) 픽셀과 블록 경계 (예컨대, 경계 (86))의 로케이션 사이의 기하학적인 거리를 나타내며, d 는 블록 경계 (86)에 대한 픽셀 값들 사이의 픽셀 값 거리 (예컨대, 차이)를 나타낸다. 일부 예들에서, $l(i)$ 의 값은 픽셀의 픽셀 값의 디블록 필터링할 양과 동일하다. 일부 예들에서, $l(i)$ 의 값은 픽셀의 픽셀 값의 디블록 필터링할 양을 결정하기 위해 추가로 수정될 수도 있다.

[0079] 일부 예들에서, 비디오 코더는 픽셀과 경계 (86) 사이의 픽셀들의 수에 기초하여 픽셀의 기하학적인 거리를 결정할 수도 있다. 일 예로서, 기하학적인 거리, n 은, 수정될 (예컨대, 디블록 필터링될) 픽셀과 블록 경계 (86) 사이의 픽셀들의 개수에 따라서 설정될 수 있다. 예를 들어, n 은 $p(0)$ 및 $q(0)$ 에 대해 0이며, n 은 $p(1)$ 및 $q(1)$ 에 대해 1이며, 기타 등등이다. 이것은 방정식: $n(i) = i$ 를 암시하며, 여기서, $i = 0 \dots N-1$ 이고, i 는 픽셀 인덱스이다.

[0080] 상기 예에서, $p(0)$ 과 $q(0)$ 사이의 거리는 무시된다 (즉, 0으로 설정된다). 기하학적인 거리, n 은, $p(0)$ 과 $q(0)$ 사이의 거리를 고려하여 (즉, 픽셀 값 $p(0)$ 을 가지는 픽셀 및 픽셀 값 $q(0)$ 을 가지는 픽셀과 경계 (86) 사이의 거리가 제로가 아닌 예들에서) 설정될 수 있다. 예를 들어, n 은 $p(0)$ 및 $q(0)$ 에 대해 0.5로 설정된다. 그 후, n 은 $p(1)$ 및 $q(1)$ 에 대해 1.5이고, $p(2)$ 및 $q(2)$ 에 대해 2.5이고, 기타 등등이다. 따라서, 이 예에서도, 픽셀의 기하학적인 거리는 픽셀과 경계 (86) 사이의 픽셀들의 수에 기초한다. 이것은 방정식: $n(i) = (2x i + 1)/2$ 을 암시한다.

[0081] 상기는 경계 (86) 까지의 디블록 필터링될 픽셀의 기하학적인 거리를 결정하는 예시적인 방법들을 설명한다. 경계 (86) 까지의 픽셀의 기하학적인 거리를 결정하는 다른 방법들이 있을 수도 있다. 예를 들어, 픽셀 값들 $p(0)$ 및 $q(0)$ 을 가지는 픽셀들이 경계 (86)로부터 0 유닛들 또는 0.5 유닛들인 것이 예시의 목적을 위해 제공된다. 픽셀 값들 $p(0)$ 및 $q(0)$ 을 가지는 픽셀들과 경계 (86) 사이의 거리는 0 또는 0.5와는 상이할 수도 있다.

[0082] 다음은 픽셀 값 차이를 결정하는 예시적인 방법들을 설명한다. 일부 예들에서, 픽셀 값 차이는 디블록 필터링될 픽셀과 다른 픽셀 사이에 존재할 수도 있다. 그러나, 본 기법들은 이에 제한되지 않는다. 일부 예들에서, 픽셀 값 차이는 2개의 픽셀들 사이에 존재할 수도 있으며, 그 2개의 픽셀들 중 어느 하나도 디블록 필터링될 픽셀이 아니다.

[0083] 비디오 코더는 $q(j) - p(i)$ 와 같이, 다른 픽셀 값으로부터 하나의 픽셀 값을 감산함으로써, 2개의 픽셀 값들, 즉, $p(i)$ 와 $q(j)$ 사이의 픽셀 값 차이를 계산한다. 픽셀 값 거리, d 는, 픽셀들, $p(N-1) p(N-2) \dots p(1)$ $p(0)$ 및 $q(0) q(1) \dots q(N-2) q(N-1)$ 의 모두 또는 일부 사이의 픽셀 값 차이를 고려함으로써 계산될 수 있다.

[0084] 일 예로서, 위치 i 에서의 d , 즉, $d(i)$ 는 $d(i) = q(0) - p(0)$, 또는 $d(i) = (q(0) - p(0))/2$ 로서 설정될 수 있다. 이 경우, $d(i)$ 는 모든 픽셀 위치들에 대해 동일하다. 예를 들어, 비디오 코더는 경계 (86)에 가장 가까운 블록 (84A)의 픽셀의 픽셀 값 (예컨대, $p(0)$), 및 경계 (86)에 가장 가까운 블록 (84B)의 픽셀의 픽셀 값 (예컨대, $q(0)$)에 기초하여 픽셀 값 차이를 결정할 수도 있다.

[0085] 이 예에서, 비디오 코더는 픽셀 값 차이를 결정하지만, 픽셀 값 차이는 디블록 필터링될 픽셀에 기초하지 않을 수도 있다. 예를 들어, 픽셀 값 $p(1)$ 을 가지는 픽셀 (즉, 경계 (86)에 가장 가까운 픽셀 다음의 픽셀)이 디블록 필터링된다고 가정한다. 이 예에서, 픽셀 값 $p(1)$ 을 가지는 픽셀에 대한 픽셀 값 차이는 $q(0)$ 과 $p(0)$ 의 픽셀 값 차이에 기초한다. 이 예에서, 비디오 코더는, 픽셀 값들 중 어느 것도 반드시 필터링중인

픽셀에 대한 것이 아닌 상황에서 2개의 픽셀들 사이의 픽셀 값 차이들을 결정하고, 그 결과를 2로 나누어, 픽셀 값을 디블록 필터링할 양을 결정하는데 사용되는 픽셀 값 차이를 결정한다.

[0086] 또한, 이 예에서, 픽셀 값 차이에 대해, 픽셀들 중 하나는 블록 (84A)로부터 유래하며 (픽셀 값 $p(0)$ 을 가지는 픽셀), 다른 픽셀은 블록 (84B)로부터 유래한다 (픽셀 값 $q(0)$ 을 가지는 픽셀). 더욱이, 이 예에서, 비디오 코더는 픽셀 값 차이를 결정하기 위해 필터링될 픽셀을 이용하지 않을 수도 있지만, 비디오 코더는 (예컨대, 위에서 설명된 $n(i)$ 에 대한 방정식들 중 하나를 이용하여) 기하학적인 거리를 결정하기 위해 필터링될 픽셀을 이용할 수도 있다.

[0087] 다른 예로서, d 는 픽셀 로케이션에 따라서 변경될 수 있다. 예를 들어, $d(i)$ 는 $d(i) = q(i) - p(i)$ 또는 $d(i) = ((q(i) - p(i))/2$ 로서 위치 i 에서의 각각의 픽셀에 대해 설정될 수 있다. 예를 들어, 비디오 코더는 블록 (84A)에서의 픽셀의 픽셀 값 및 블록 (84B)에서의 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정할 수도 있으며, 여기서, 개개의 블록들 (84A 및 84B)에서의 픽셀들은 경계 (86)까지 동일한 거리이다. 다시 말해서, 블록 (84A)에서의 픽셀 값 $p(i)$ 를 가지는 픽셀은 블록 (84B)에서의 픽셀 값 $q(i)$ 를 가지는 픽셀과, 경계 (86)까지 등거리이다.

[0088] 이 예에서, 비디오 코더는 픽셀 값 차이를 결정하고, 픽셀 값 차이는 디블록 필터링될 픽셀에 기초한다. 예를 들어, 픽셀 값 $p(1)$ 을 가지는 픽셀 (즉, 경계 (86)에 가장 가까운 픽셀 다음의 픽셀)이 디블록 필터링되는 것으로 가정한다. 이 예에서, 픽셀 값 $p(1)$ 을 가지는 픽셀에 대한 픽셀 값 차이는 $q(1)$ 과 $p(1)$ 의 픽셀 값 차이에 기초한다. 이 예에서, 비디오 코더는 픽셀 값들 중 하나가 필터링중인 픽셀에 대한 것인 상황에서 2개의 픽셀들 사이의 픽셀 값 차이들을 결정하고, 그 결과를 2로 나누어, 픽셀 값을 디블록 필터링할 양을 결정하는데 사용되는 픽셀 값 차이를 결정한다.

[0089] 위와 같이, 이 예에서, 픽셀 값 차이에 대해, 픽셀들 중 하나는 블록 (84A)으로부터 유래하며 (픽셀 값 $p(i)$ 를 가지는 픽셀), 다른 픽셀은 블록 (84B)로부터 유래하며 (픽셀 값 $q(i)$ 를 가지는 픽셀), 그러나 픽셀 값 차이를 결정하는데 사용되는 픽셀들 중 적어도 하나는 디블록 필터링될 픽셀이다 (예컨대, 픽셀 값 $p(i)$ 를 가지는 픽셀 및 픽셀 값 $q(i)$ 를 가지는 픽셀이 디블록 필터링될 것이다). 또한, 기하학적인 거리는 경계 (86)까지의 필터링될 픽셀의 거리에 기초하여 결정된다.

[0090] 다른 예에서, $d(i)$ 는 $d(i) = q(0) - p(i)$, $d(i) = q(i) - p(0)$, $(q(0) - p(i))/2$, 또는 $d(i) = (q(i) - p(0))/2$ 로서 설정될 수 있다. 이 예에서, 비디오 코더는 필터링될 픽셀의 픽셀 값 (예컨대, $p(i)$ 또는 $q(i)$), 및 경계 (86)에 가장 가까운 다른 블록에서의 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정한다. 예를 들어, 블록 (84A)에서의 i 번째 픽셀에 대한 픽셀 값 차이는 $p(i)$ 및 $q(0)$ 에 기초하며, 여기서, $q(0)$ 는 블록 (84B)에서 그리고 경계 (86)에서 가장 가까운 픽셀에 대한 픽셀 값이다.

[0091] 다른 예로서, 위치 i 에서의 d , 즉 $d(i)$ 는, $d_p(i) = p(i+1) - p(i-1)$, 또는 $(p(i+1) - p(i-1))/2$, 및 $d_q(i) = q(i+1) - q(i-1)$, 또는 $(q(i+1) - q(i-1))/2$ 로서 이웃하는 픽셀들을 이용하여 계산될 수 있다. 이 경우, 각각, $p(-1)$ 는 $q(0)$ 로서 설정될 수 있으며 $q(-1)$ 은 $p(0)$ 로서 설정될 수 있다. 이 예에서, 비디오 코더는 디블록 필터링될 픽셀에 이웃하는 픽셀들의 픽셀 값들에 기초하여 픽셀 값 차이를 결정한다. 다시 말해서, 비디오 코더는 디블록 필터링될 픽셀의 제 1 이웃하는 픽셀, 및 디블록 필터링될 픽셀의 제 2 이웃하는 픽셀에 기초하여, 픽셀 값 차이를 결정한다. 2개의 이웃하는 픽셀들은 동일한 로우 또는 칼럼에 있을 수도 있다. 비디오 코더는 디블록 필터링될 실제 픽셀의 픽셀 값을 이용하지 않을 수도 있지만, 디블록 필터링될 실제 픽셀의 픽셀 값을 이용하는 것이 가능할 수도 있다.

[0092] 또한, 이웃하는 픽셀들은 경계 (86)에 가장 가깝게 로케이트된 픽셀들을 제외한, 디블록 필터링될 픽셀과 동일한 블록에 존재할 수도 있다. 예를 들어, 픽셀 값 $p(0)$ 을 가지는 픽셀이 디블록 필터링되는 것으로 가정한다. 픽셀 값 $p(0)$ 을 가지는 픽셀에 이웃하는 픽셀들은 $p(1)$ 및 $q(0)$ 이다. 픽셀 값 $p(1)$ 을 가지는 픽셀은 픽셀 값 $p(0)$ 을 가지는 픽셀과 동일한 블록 (예컨대, 블록 (84A))에 존재하지만, 픽셀 값 $q(0)$ 을 가지는 픽셀은 상이한 블록 (예컨대, 블록 (84B))에 존재한다.

[0093] 일부 예들에서, d 는 픽셀 값 차이들의 배수를 가중 총합으로서 결합함으로써 계산될 수 있다. 예를 들어, $d(i) = \sum_j w(j) \times d(j)$ 이며, 여기서 $\sum_j w(j) = 1$, $j = 0 \dots N-1$ 이다. 이 예에서, 픽셀 값 차이는 다수의 픽셀들의 픽셀 값들의 픽셀 값 차이들의 가중치에 기초한다. 일부 예들에서, 픽셀 값 차이는 예컨대, $q(j) - p(j)$ 가 $d(j)$ 와 동일한 경우에 대해, 디블록 필터링될 픽셀들의 모두에 대한 픽셀 값들의 픽셀 값 차이

들의 가중치에 기초한다. 일반적으로, $d(j)$ 의 방정식은 $d(i)$ 의 상기 예시적인 방정식들 중 임의의 방정식 일 수도 있다.

[0094] 이 예에서, 비디오 코더는 적어도 픽셀 값 차이에 대한 가중치 (예컨대, $d(j)$ 의 적어도 하나의 인스턴스에 대한 가중치, 여기서, $d(j)$ 의 방정식은 $d(i)$ 에 대한 상기 방정식들 중 임의의 방정식과 동일하다) 를 결정할 수도 있다. 일부 예들에서, 비디오 코더는 픽셀 값 차이들의 각각에 대한 가중치 (예컨대, $d(j)$ 의 모든 인스턴스에 대한 가중치) 를 결정할 수도 있다. $\sum_j w(j) = 1$, $j = 0 \dots N-1$ 의 가중치 방정식은 가중치 방정식의 일 예이며, 비디오 코더는 다른 기법들을 이용하여, 픽셀 값 차이들 (즉, $d(j)$) 에 대한 가중치를 결정할 수도 있다.

[0095] 픽셀 값 차이 (예컨대, $d(i)$) 및 경계 (86) 까지의 필터링될 픽셀의 기하학적인 거리 (예컨대, $n(i)$) 에 의해, 비디오 코더는 픽셀을 디블록 필터링할 양을 결정할 수도 있다 (즉, 픽셀 거리는 $d(i)$ 과 $n(i)$ 를 결합하여 형성될 수도 있다). 예를 들어, $l(i)$ 는 양쪽의 항들을 곱하여 계산될 수 있다. 비디오 코더가 디블록 필터링할 양을 결정할 수도 있는 일 예시적인 방법은 $l(i) = d(i) \times (N - n(i))/N$ 이다. 다시 말해서, 비디오 코더는 픽셀 값 차이, 경계로부터의 픽셀의 기하학적인 거리, 그리고, 일부 예들에서는, 디블록 필터링될 픽셀들의 개수 (예컨대, N) 에 기초하여, 픽셀의 픽셀 값을 디블록 필터링할 양을 결정할 수도 있다.

[0096] 비디오 코더는 다음 방정식들: $p'(i) = g1(l, p(i))$, 및 $q'(i) = g2(l, q(i))$ 에 기초하여, 픽셀 값을 필터링 할 수도 있다 (즉, 그후, 각각의 픽셀 값이 수정된다). 함수들 $g1$ 및 $g2$ 의 예들로서, 비디오 코더는 다음 방정식들: $p'(i) = p(i) + l(i)$, 및 $q'(i) = q(i) - l(i)$ 을 수행할 수도 있다. 이를 방정식들에서, $p'(i)$ 는 블록 (84A) 에서의 i 번째 픽셀에 대한 디블록 필터링된 픽셀 값을 나타내며, $q'(i)$ 는 블록 (84B) 에서 i 번째 픽셀에 대한 디블록 필터링된 값을 나타낸다.

[0097] 그러나, 일부의 경우, $l(i)$ 가 양의 값이면, $p'(i)$ 가 최대 허용가능 픽셀 값보다 더 크게하거나, 또는 $q'(i)$ 가 최소 허용가능 픽셀 값보다 작게 하는 것이 가능할 수도 있다. $l(i)$ 가 음의 값이면, $p'(i)$ 가 최소 허용가능 픽셀 값보다 작게 하고 $q'(i)$ 가 최대 허용가능 픽셀 값보다 크게 하는 것이 가능할 수도 있다. 예를 들어, 최소 픽셀 값은 0 일 수도 있으며, 최대 픽셀 값은 255 일 수도 있다.

[0098] 일부 예들에서, 디블록 필터링된 픽셀의 픽셀 값이 최대 허용가능 픽셀 값 미만이거나 또는 동일하고 최소 허용 가능 픽셀 값보다 크거나 또는 동일함을 보장하기 위해, 비디오 코더는 $g1$ 및 $g2$ 함수들의 추가적인 예들인, 다음 방정식들: $p'(i) = clip(Min, Max, p(i) + l(i))$, 및 $q'(i) = clip(Min, Max, q(i) - l(i))$ 을 수행할 수도 있으며, 여기서, Min 및 Max 는 최소 및 최대 픽셀 샘플 값을 나타낸다. 이러한 방법으로, $p(i) + l(i)$ 가 최대 또는 최소 값보다 크거나 또는 미만이면, 비디오 코더는 디블록 필터링된 픽셀의 픽셀 값을 Max 또는 Min 으로 각각 클리핑한다. 이와 유사하게, $q(i) - l(i)$ 가 최대 또는 최소 값보다 크거나 또는 미만이면, 비디오 코더는 디블록 필터링된 픽셀의 픽셀 값을 Max 또는 Min 으로 각각 클리핑한다.

[0099] 더욱이, 일부 예들에서, 픽셀의 픽셀 값이 디블록 필터링되는 양을 제한하는 것이 가능할 수도 있다. 예를 들어, HEVC 에서의 디블록 필터링 프로세스에서, 수정할 수 있는 픽셀 값 (예컨대, 픽셀의 픽셀 값이 필터링되는 양) 이 제한되며 그 한계를 설정하는 임계값이 시그널링될 수 있다. 동일한 방법으로, $l(i)$ 는 임계값을 이용하여 클리핑될 수 있다. 예를 들어, 비디오 인코더 (20) 는 최대 허용가능 조정 임계값을 시그널링할 수도 있으며 (즉, $l(i)$ 의 절대값은 최대 허용가능 조정 임계값 미만이거나 또는 동일하다), 비디오 디코더 (30) 는 이러한 임계값을 수신할 수도 있다. 일부 예들에서, HEVC 디블록 필터에 대한 임계값은 상수 값을 이용하여 그것을 스케일링함으로써 사용될 수 있거나, 또는 별개의 값이 최대 허용가능 조정 임계값을 결정하기 위해 비트스트림으로 시그널링될 수 있다. 픽셀 값이 수정될 (예컨대, 필터링됨) 수 있는 양을 제한하는 것이 모든 예에서 필요하지 않는 것으로 이해되어야 한다.

[0100] 이와 같이, 비디오 코더는 경계까지의 픽셀들의 픽셀 거리들에 기초하여 (즉, 픽셀 거리들이 픽셀 값 차이들 및 블록의 경계까지의 기하학적인 거리에 기초하는 경우) 픽셀들의 픽셀 값을 디블록 필터링할 양을 결정할 수도 있다. 예를 들어, $l(i)$ 는 경계까지의 픽셀들의 거리들의 함수인 값이다. 일부 예들에서, $l(i)$ 는 오프셋 값으로서 간주될 수도 있다. 일부 예들에서, 픽셀들 (n) 의 개수는 적어도 8개와 동일하다.

[0101] 일부 예들에서, $l(i)$ 는 경계에 더 가까운 픽셀들에 대해 더 크고, 경계로부터 더 멀리 있는 픽셀들에 대해 더 작을 수도 있다. 예를 들어, $l(i)$ 에 대한 방정식의 일 예는 $d(i) \times (N - n(i))/N$ 이다. $n(i)$ 의 값은 경계 (86) 로부터 더 멀리 떨어져 있는 픽셀들에 대해 더 크고, 경계 (86) 에 더 가까운 픽셀들에 대해 더 작을 수도 있다. 따라서, $N - n(i)$ 의 값은 경계 (86) 로부터 더 멀리 떨어져 있는 픽셀들에 대해 더 작고, 경계

(86) 에 더 가까운 픽셀에 대해 더 크다. 따라서, 1(i) 는 경계 (86) 에 더 가까운 픽셀들에 대해 더 크고, 경계 (86) 로부터 더 멀리 떨어져 있는 픽셀들에 대해 더 작다.

[0102] 상기 설명은 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 가 종종 긴 필터링 프로세스로서 또한 지칭되는, 디블록 필터링을 수행하는 방법들에 대한 예들을 제공한다. 그러나, 비디오 코더는 디블록 필터링을 모든 경우들에서 수행하지 않을 수도 있다. 다음은 긴 필터링 프로세스가 호출되는 시점에 대한 조건들 (예컨대, 비디오 코더가 본 개시물에서 설명되는 디블록킹 필터링 기법들을 수행할 때의 예들) 을 설명한다.

[0103] 예를 들어, 비디오 코더는 날카로운 에지들 또는 텍스쳐들을 가지는 영역이 디블록킹 필터에 의해 과-평활화되지 않도록, 단지 긴 필터 (long filter) 를 평활화 영역에 적용할 수도 있다. 위에서 설명된 바와 같이, 디블록킹 필터링 기법들에 의해 수행되는 평활화는 픽셀 값들이 경계의 반대 측면들 상에서 변하는 양에서의 감소를 초래한다. 그러나, 그 비디오 콘텐츠는 비디오 콘텐츠에서 에지들이 존재하도록 할 것이다. 이들 에지들을 평활화 (예컨대, 과-평활화) 하는 것은 시각적 품질에 악영향을 미칠 수도 있다.

[0104] 예를 들어, 비디오 블록에서, 에지는 2개의 이웃하는 픽셀들 사이의 픽셀 값들에서 갑작스러운 변화가 있을 때 발생할 수도 있으며 (예컨대, 일 예로서, 블록 내 백색 부분에 이웃하는 블랙 부분, 그러나 덜 엄격한 에지들의 예들도 또한 있다), 이러한 부분들을 평활화하는 것은 시각적 품질을 열화시킬 수도 있다. 이 예에서, 디블록 필터링이 백색 부분으로의 블록 부분 전이들이 있었던 부분에 적용되었으면, 그 결과는 에지에서 흐려져서 (예컨대, 과 평활화되어) 비디오 품질에서 열화를 초래할 것이다. 따라서, 급격한 자연스러운 에지들 (drastic natural edges) 이 존재하는 부분들 (예컨대, 비디오 콘텐츠 자체가 픽셀 값들에서의 갑작스러운 변화들을 포함하는 부분들) 에는 디블록 필터링 기법들을 적용하지 않고, 그리고 급격한 자연스러운 에지들이 존재하지 않는 부분들 (예컨대, 비디오 콘텐츠 자체가 아닌 비디오 코딩의 부분으로서 추가된 블록킹 아티팩트들이 제거되도록 하여 상대적으로 매끄러운 비디오 콘텐츠) 에는 디블록 필터링 기법들을 적용하는 것이 유익할 수도 있다.

[0105] 일부 예들에서, 비디오 코더는 기울기 (gradient) 에 의해 로컬 영역의 평활도를 결정할 수도 있다. 기울기로부터의 결정된 평활도의 결과에 기초하여, 비디오 코더는 디블록킹 필터링을 수행할 수도 있다.

[0106] 예를 들어, X 개의 픽셀들을 가지는 디블록킹 경계에 대해, 긴 필터가 적용될 때 경계의 각각의 측면에서의 XxM 픽셀들이 프로세싱된다고 가정한다. 예를 들어, 도 4 에 예시된 바와 같이, 경계 (86) 에 대해, X 는 경계 (86) 를 따른 수직 픽셀들의 개수를 지칭하며, M 은 디블록 필터링될 수평 픽셀들의 개수를 지칭한다. 수평 경계에 대해, X 는 수평 경계를 따른 수평 픽셀들의 개수를 지칭할 것이며, M 은 디블록 필터링되는 수직 픽셀들의 개수를 지칭한다. 기울기 계산을 위한 다음 설명에서, 변수 "M" 은 상기 변수 "N" 과 동일한 것으로 간주될 수도 있으며, 여기서, N 은 블록에서 디블록 필터링될 픽셀들의 개수를 나타낸다.

[0107] 일부 예들에서, 비디오 코더는 긴 필터를 적용할지 여부를 결정하기 위해, 디블록킹 경계의 각각의 측면 (예컨대, 경계 (86) 의 각각의 측면) 에서 XxM' (M' 은 M 보다 크거나 또는 동일하다) 픽셀 영역의 기울기들을 결정할 수도 있다. 디블록킹 경계의 양면의 기울기들이 임계치 기울기 값 미만이라는 결정에 기초하여, 비디오 코더는 긴 필터를 적용할 수도 있다 (즉, 본 개시물에서 설명되는 디블록 필터링 기법들의 하나 이상의 예들).

그렇지 않으면, 비디오 코더는 긴 필터를 적용하지 않을 수도 있다. 이와 같이, 긴 필터링 프로세스 (예컨대, 본 개시물에서 설명되는 디블록 필터링 프로세스) 가 제 1 유형의 필터링으로서 간주될 수도 있으며, 위에서 인용한 문서들 또는 HEVC 표준에서 설명된 것들과 같은 다른 디블록 필터링 기법들 중 임의의 하나가 제 2 유형의 필터링으로 간주될 수도 있다. 일부 예들에서, 비디오 코더는 이웃하는 픽셀들에 기초하여 기울기를 결정하고, 그 기울기를 임계치 기울기 값과 비교할 수도 있다. 이들 예들에서, 비디오 코더는 그 비교에 기초하여 제 1 유형의 필터링 또는 제 2 유형의 필터링을 수행할 수도 있다.

[0108] 계산 복잡성을 감소시키기 위해, 비디오 코더는 단지 부분 픽셀들만을 선택하여, XxM' 픽셀들의 기울기를 계산할 수도 있다. 예를 들어, 기울기는 2개의 연속된 픽셀들 사이의 차이에 기초하여 계산될 수도 있다. 그러나, 비디오 코더는 다수의 픽셀들 (예컨대, 또는 블록에서의 픽셀들 모두) 을 이용하여 기울기를 결정할 수도 있다. 예를 들어, 평면이 모델링될 수도 있으며, 그의 기울기는 블록에서의 픽셀들을 이용하여 추정될 수 있다.

[0109] 일 예로서, 인터리브된 픽셀들이 다음과 같이 기울기를 계산하는데 사용된다: $P(i, j)$ 가 하나의 XxM' 블록에서의 픽셀들을 나타낸다고 하고 여기서 $i = 0, 1, 2 \dots X-1$, $j = 0, 1, 2 \dots M'-1$ 이며, 그리고 블록의 기울기는 다음과

같이 계산된다: $i = 0 \dots X-1, j = 0 \dots M'-1$ 인 경우 $\text{gradientH} = \sum_{i,j} \text{abs}(P(i, j) - P(i, \min(M'-1, j+1)))$; $i = 0 \dots X-1, j = 0 \dots M'-1$ 인 경우 $\text{gradientV} = \sum_{i,j} \text{abs}(P(i, j) - P(\min(i+1, X-1), j))$; 및 기울기 ($\text{gradient} = \text{gradientH} + \text{gradientV}$). 이 예에서, 비디오 코더가 기울기가 임계치 미만이라고 결정하는 것에 기초하여, 비디오 코더는 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할 수도 있다. 또한, 비디오 코더가 기울기가 임계치보다 크다고 결정하는 것에 기초하여, 비디오 코더는 HEVC 표준에서의 필터링 기법들과 같은 필터링 기법들을 수행할 수도 있다.

[0110] 상기 예에서, 비디오 코더는 수평 또는 수직 디블록킹 경계에 대해 고려된 수평 기울기 및 수직 기울기 양쪽 (예컨대, 수평 기울기 및 수직 기울기 양쪽)을 이용하였다. 일부 예들에서, 비디오 코더는 어떤 경계에 대해, 하나의 방향의 기울기를 결정하고 그리고 다른 어떤 방향의 기울기도 결정하지 않고, 그 기울기를 고려할 수도 있다 (예컨대, 수직 기울기는 수평 경계가 필터링될 때 사용되며 수평 기울기는 수직 경계가 필터링될 때 사용된다).

[0111] 기울기를 계산하는데 계산의 수를 감소시키기 위해, 비디오 코더는 XxM' 영역을 서브-샘플링하여, 기울기를 계산할 수도 있다. 예를 들어, 수평 및 수직 기울기들은 다음과 같이 계산될 수 있다: ($i = 0, 2, 4 \dots X-2, j = 0, 2, 4 \dots M'-2$) 및 ($i = 1, 3, 5 \dots X-1, j = 1, 3, 5 \dots M'-1$) 인 경우 $\text{gradientH} = \sum_{i,j} \text{abs}(P(i, j) - P(i, \min(M'-1, j+1)))$, 및 ($i = 0, 2, 4 \dots N-2, j = 0, 2, 4 \dots M'-2$) 및 ($i = 1, 3, 5 \dots X-1, j = 1, 3, 5 \dots M'-1$) 인 경우 $\text{gradientV} = \sum_{i,j} \text{abs}(P(i, j) - P(\min(i+1, X-1), j))$.

[0112] 상기 예에서, 비디오 코더는 필터링될 블록의 픽셀들의 모두를 포함하는 XxM' 영역에 대한 기울기 값을 결정한다 (즉, 상기 예에서, 각각의 XxM' 영역에 대한 하나의 기울기 값이 계산된다는 점에 유의한다). 일부 예들에서, 각각의 영역은 파티셔닝될 수 있으며, 비디오 코더는 각각의 파티션의 필터링을 제어하기 위해 각각의 영역에서 다수의 기울기들을 결정할 (예컨대, 계산할) 수도 있다. 일 예로서, 파티션은 XxM' 의 영역에서의 라인일 수도 있다. 아래 예에서, 영역은 각각의 라인으로 세그먼트화되며, 여기서, 비디오 코더는 경계가 수직이면 위치 i 에서의 수평 기울기를 $\text{gradientH}(i) = \sum_j \text{abs}(P(i, j) - P(i, \min(M'-1, j+1)))$ 로서, 그리고 경계가 수평이면 위치 j 에서의 수직 기울기를 $\text{gradientV}(j) = \sum_i \text{abs}(P(i, j) - P(\min(i+1, X-1), j))$ 로서 결정한다. 그 후, 비디오 코더는 $\text{gradientH}(i)$ 에 의해 수직 위치 i 에서의 픽셀들의 필터링을 제어할 수도 있으며, 수평 위치 j 에서의 픽셀들의 필터링은 $\text{gradientV}(j)$ 에 의해 제어될 수 있다.

[0113] 일부 예들에서, 또한 비디오 코더가 각각의 픽셀 위치에서의 기울기의 사용에 의해 각각의 픽셀의 필터링을 제어하는 것이 가능하다. 예를 들어, 어떤 픽셀 위치에서의 기울기가 임계치보다 더 높다고 비디오 코더가 결정하면, 이 위치는 실제 (true) 이미지 에지로서 간주될 수 있으며, 비디오 코더는 이러한 픽셀의 필터링을 회피할 것이다 (또는, 적어도 본 개시물에서 설명하는 기법들을 이용하여 픽셀을 필터링하지 않을 것이다).

[0114] 더욱이, 일부 예들에서, 비디오 코더는 블록 (예컨대, XxM'), 세그먼트 또는 라인 파티션, 또는 픽셀에서의 기울기 결정들을 결합하여, 픽셀 또는 픽셀들을 디블록 필터링할지 여부를 결정할 수도 있다. 다시 말해서, 상기 예들에서 - 블록 레벨, 세그먼트 또는 라인 레벨, 및 픽셀 레벨 제어는 - 필터링의 결정을 행하기 위해 함께 결합될 수 있다.

[0115] 일부 예들에서, 비디오 코더는 기울기를 결정하고 그 기울기를 임계치 기울기와 비교한다. 이 비교에 기초하여, 비디오 코더는 본 개시물에서 설명하는 기법들을 이용하여 픽셀의 픽셀 값을 디블록 필터링하거나, 다른 기법들을 이용하여 픽셀의 픽셀 값을 디블록 필터링하거나, 또는 픽셀 값을 디블록 필터링하지 않는다. 예를 들어, 기울기가 임계치 기울기 미만이면, 비디오 코더는 본 개시물에서 설명하는 기법들을 이용하여 픽셀들의 픽셀 값을 디블록 필터링할 수도 있으며; 아니면, 비디오 코더는 다른 기법들을 이용하여 디블록 필터링하거나 또는 디블록 필터링하지 않을 수도 있다.

[0116] 비디오 인코더 (20)는 기울기 값 (예컨대, 임계치 기울기)에 대한 임계값을 SPS, PPS 또는 슬라이스 헤더로 시그널링할 수도 있으며, 비디오 디코더 (30)는 임계치 기울기를 SPS, PPS, 또는 슬라이스 헤더로 수신할 수도 있다. 일부 예들에서, 임계값은 인트라 슬라이스 및 비-인트라 슬라이스에 대해 별개로 시그널링될 수 있다. 시그널링된 임계값은 "CE12, Subset 1: Report of deblocking for large size blocks"에서 처럼, 기준 방법들에서 언급된 분산을 포함한, 다른 평활도 메트릭들에 대해 적용될 수 있다. 일부 예들에서, 임계값은 시그널링이 회피될 수 있도록, 인코더 측면 (비디오 인코더 (20)) 및 디코더 측면 (비디오 디코더 (30))에서 동일한 상수 값을 수 있다.

- [0117] 비디오 코더는 각각의 수평 및 수직 필터링 방향에 대해 동일한 임계치 기울기 값을 적용할 수도 있거나, 또는 상이한 임계값이 각각의 필터링 방향에 대해 사용될 수 있다. 임계치 기울기 값들이 상이한 방향들에 대해 상이하면, 각각의 방향에 대해, 비디오 인코더 (20) 는 비디오 디코더 (30) 가 수신하는 별개의 값들을 시그널링할 수도 있다. 다른 예로서, 비디오 인코더 (20) 는 하나의 값을 시그널링하고 그 값을 상이하게 스케일링하여 각각의 방향에 대해 기울기 임계값들을 발생시킬 수도 있다.
- [0118] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 변환 블록 사이즈 (예컨대, 블록의 인코딩 또는 디코딩에 사용되는 변환 블록의 사이즈), 코딩 모드 (예컨대, 인트라- 또는 인터-예측 코딩), 모션 벡터, 복원된 픽셀 값들, 등과 같은, 비디오 인코더 (20) 및 비디오 디코더 (30) 양쪽으로 입수가능한 정보에 의존하여 기울기 임계치를 조정할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 비디오 인코더 (20) 및 비디오 디코더 (30) 양쪽에 의해 입수가능한 정보에 의해 선택될 수 있는 경우들의 세트에 대해 상이한 임계치 기울기 값을 시그널링할 수도 있다.
- [0119] 전술한 메트릭들에 기초하여, 비디오 코더는 디블록 필터링을 제어할 수도 있다. 일 예로서, 비디오 코더는 기울기가 임계치 기울기보다 더 클 때 디블록 필터링을 턴오프시킬 수도 있다. 일부 예들에서, 기울기 값의 범위에 따라서, 상이한 강도에 의한 필터링이 적용될 수 있다. 예를 들어, 비디오 코더는 기울기를 이용하여 1(i) 의 값을 조정할 (예컨대, 결정된 기울기에 의해 1(i) 의 값을 스케일링할) 수도 있다. 다시 말해서, 변경의 양이 큰 기울기를 가지는 영역에 대해 작은 1(i) 로 감소되도록 기울기에 따라서 픽셀 거리 1(i) 를 감소시키는 것이 가능하거나, 또는 비디오 코더는 기울기가 충분히 작지 않을 때 더 약한 필터 (예컨대, HEVC 에서 정의된 필터들) 를 적용할 수도 있다.
- [0120] 일반적으로, 일부 예들에서, 비디오 코더는 픽셀 값들이 디블록 필터링될 하나 이상의 픽셀들을 포함하는 블록의 부분의 기울기 값을 결정할 수도 있으며, 여기서, 기울기 값은 블록의 부분의 평활도를 표시한다 (즉, 블록의 부분에 이미지 에지들이 존재하는지 여부를 표시한다). 블록의 부분은 디블록 필터링될 블록의 픽셀들의 모두를 포함하는 영역 (예컨대, XxM' 영역) 일 수도 있다. 기울기 값을 결정할 때에, 비디오 코더는 기울기 값을 결정하기 위해 영역에서의 픽셀들의 서브-샘플을 이용할 수도 있다 (즉, 기울기 값 결정을 위해 영역에서의 픽셀들의 모두를 반드시 이용할 필요가 없다).
- [0121] 블록의 부분은 디블록 필터링될 픽셀들을 포함하는 블록의 1차원 라인일 수도 있다. 이 예에서, 비디오 코더는 라인 단위로 기울기 값을 결정하고 (예컨대, 각각의 라인에 대한 기울기 값), 기울기 값 결정을 위해 라인에서의 픽셀들의 모두를 반드시 이용할 필요가 없을 수도 있다. 블록의 부분은 잠재적으로 필터링될 각각의 픽셀 (예컨대, 픽셀 단위로 각각의 픽셀에 대한 기울기 값) 일 수도 있다.
- [0122] 비디오 코더는 기울기 값을 임계치 기울기 값과 비교하고, 그 비교에 기초하여 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할 수도 있다. 예를 들어, 기울기 값이 임계치 기울기 값 미만이면, 비디오 코더는 본 개시물에서 설명된 바와 같이 디블록 필터링 기법들을 수행할 수도 있으며, 기울기 값이 임계치 기울기 값보다 크면, 비디오 코더는 일부 다른 유형의 디블록 필터링을 수행할 수도 있거나 또는 디블록 필터링을 수행하지 않을 수도 있다.
- [0123] 더욱이, 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 양쪽이 기울기 값을 결정할 필요가 없을 수도 있다. 예를 들어, 비디오 인코더 (20) 는 기울기 값을 결정하고, 본 개시물에서 설명되는 바와 같이 디블록 필터링을 수행할지 여부를 결정할 수도 있다. 비디오 인코더 (20) 는 블록이 본 개시물에서 설명하는 기법들을 이용하여 디블록 필터링될지 여부를 표시하는 선택스 엘리먼트 (예컨대, 플래그) 를 시그널링할 수도 있다. 이러한 경우, 비디오 디코더 (30) 는 기울기 값을 결정하거나 또는 기울기 값에 기초하여 본 개시물에서 설명된 바와 같은 디블록 필터링을 수행할지 여부를 결정할 필요가 없을 수도 있다. 이러한 방법으로, 비디오 디코더 (30) 는 계산적으로 복잡한 프로세스일 수도 있는, 기울기 값을 결정할 필요가 없을 수도 있다.
- [0124] 그러나, 대역폭 효율을 위해, 일부 예들에서, 비디오 디코더 (30) 는 기울기 값 계산을 수행하고 결정된 기울기 값에 기초하여 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할지 여부를 결정할 수도 있다. 이러한 방법으로, 비디오 디코더 (30) 가 이러한 결정을 행할 것이기 때문에, 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할지 여부를 나타내는 어떤 추가적인 선택스 엘리먼트도 요구되지 않을 것이다.
- [0125] 상기는 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할지 여부를 결정하는데 사용될 수도 있는 일 예를 설명한다. 그러나, 기울기 값에 기초하여, 본 개시물에서 설명되는 디블록 필터링을 수행할지 여부를 결정하는 것은 일 예이다. 다음은 기존 HEVC 디블록킹 필터들과 조합하여 필터가 적용되는 방법의 추가적인 예

들을 설명한다.

[0126] 일부 예들에서, 비디오 코더는 아래에서 설명되는 예시적인 조건들 중 하나 이상이 만족될 때 본 개시물에서 설명되는 디블록 필터링 기법들을 적용할 수도 있다. 그렇지 않으면, 다른 제안들 (예컨대, HEVC 사양)에서 정의된 디블록 필터링 프로세스가 적용될 수도 있다.

[0127] 또한, 다음 예들에서, 비디오 코더는 또한 아래의 조건들에 더해서, 기울기 값 결정을 이용하여, 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할지 여부를 결정할 수도 있다. 예를 들어, 다음 조건들 중 어느 것도 참이 아니더라도 그리고 TU 경계의 2개의 8x8 블록들로부터의 기울기가 어떤 임계치보다 더 작더라도, 비디오 코더는 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할 수도 있다.

[0128] 일부 예들에서, 비디오 코더는 2개의 인접한 (예컨대, 이웃하는) TU들에 대한 사이즈들을 결정할 수도 있다. 또, TU 는 비디오 인코더 (20)에 의해 변환되어 양자화되고 비디오 디코더 (30)에 의해 역양자화되어 역변환되는 블록이다. 일부 경우들에서 블록의 변환 및/또는 양자화가 스kip될 수도 있음에 유의해야 한다. TU에 있어서, 루마 성분에 대한 변환 블록, 제 1 크로마 성분에 대한 변환 블록, 및 제 2 크로마 성분에 대한 변환 블록이 존재한다. 본 개시물에서, TU의 사이즈는 루마 성분 또는 크로마 성분들 중 어느 하나에 대한 변환 블록일 수도 있는 변환 블록의 사이즈를 지칭한다. 예를 들어, 비디오 코더는 루마 성분 및 2개의 크로마 성분들의 블록들에 대해 디블록 필터링 기법들을 수행할 수도 있다. 그 성분들 중 하나 이상에 대해, 비디오 코더는 2개의 인접한 변환 블록들의 사이즈들을 결정할 수도 있다.

[0129] 2개의 인접한 TU들 사이의 더 큰 변환 사이즈 (예컨대, 그 성분들 중 하나의 변환 블록) 가 최대 허용가능 변환 블록 사이즈 (예컨대, MaxTrafoSize) 와 동일하다는 결정에 기초하여, 비디오 코더는 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할 수도 있다. 일부 예들에서, 최대 허용가능 변환 블록 사이즈는 32x32 이다.

[0130] 일부 예들에서, 블록들 중 하나가 최대 허용가능 변환 블록 사이즈와 동일한 것에 더해서, 비디오 코더는 관련된 CU 가 인트라-코딩되거나 (인트라-예측 코딩되거나) 또는 인터-코딩되는지 (인터-예측 코딩되는지) 여부를 결정할 수도 있다. 코딩되는 블록인 CU, 및 TU 는 블록의 부분이다.

[0131] 관련된 CU 가 인트라-코딩된다고 비디오 코더가 결정하면, 인접한 TU들 중 하나는 최대 허용가능 변환 블록 사이즈이다. 비디오 코더는 본 개시물에서 설명하는 기법들을 이용하여 픽셀 값들을 디블록 필터링한다. 관련된 CU 가 인트라-코딩되지 않지만, CU의 파티션 사이즈가 2Nx2N 이고 TU의 cbf 가 1 과 동일하다 (즉, TU의 변환이 스kip된다) 고 비디오 코더가 결정하면, 비디오 코더는 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할 수도 있다.

[0132] 다른 예에서, 아래 조건들 중 어느 하나 또는 양자가 상기 2개의 조건들의 상부에 추가될 수 있다. 예를 들어, 최대 변환 유닛 사이즈 (MaxTrafoSize) 가 16x16 보다 크거나 또는 동일하면, 비디오 코더는 본 개시물에서 설명되는 디블록 필터링 기법들을 수행하도록 구성될 수도 있다.

[0133] 또한, 상기 조건들 중 하나는 인접한 TU들 중 하나의 사이즈가 최대 허용가능 TU 사이즈와 동일한지 여부에 기초하였다. 다음은 인접한 TU들 중 더 작은 TU의 사이즈가 특정의 사이즈보다 크거나 또는 동일한지 여부에 기초하는 조건들의 일 예이다. 예를 들어, 2개의 TU들 사이의 더 작은 변환 사이즈가 16x16 보다 크거나 또는 동일하다고 비디오 코더가 결정하면, 비디오 코더는 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할 수도 있다. 예를 들어, 관련된 CU 가 인트라-코딩되고 2개의 TU들 사이의 더 작은 변환 사이즈가 16x16 보다 크거나 또는 동일하면, 비디오 코더는 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할 수도 있다.

다른 예로서, 관련된 CU 가 인트라-코딩되지 않고 CU의 파티션 사이즈가 2Nx2N 이고, 그리고 2개의 TU들 사이의 더 작은 변환 사이즈가 16x16 보다 크거나 또는 동일하면, 비디오 코더는 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할 수도 있다.

[0134] 디블록 필터링 기법들이 적용될 때의 상기 예시적인 조건들은 단지 예시의 목적을 위해 제공되는 것으로 이해되어야 한다. 비디오 코더가 본 개시물에서 설명되는 디블록 필터링 기법들을 수행할지 여부를 결정하기 위해 평가할 수도 있는 다른 조건들이 있을 수도 있다.

[0135] 도 2 는 본 개시물에서 설명되는 바와 같은 기법들을 이용할 수도 있는 비디오 인코더 (20)의 일 예를 예시하는 블록도이다. 비디오 인코더 (20)는 디블록 필터링을 이용하는 다른 코딩 표준들 또는 방법들에 대해, 예시의 목적을 위해, 그러나 본 개시물의 한정 없이, HEVC 코딩의 상황에서 설명될 것이다. 비디오 인코더 (20)는 비디오 프레임들 내 CU들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라 코딩은 주어진 비디오 프레임 내 비디오 데이터에서 공간 리던던시를 감소시키거나 또는 제거하기 위해 공간 예측에 의존한다.

인터-코딩은 비디오 시퀀스의 현재의 프레임과 이전에 코딩된 프레임들 사이에 시간 리턴던시를 감소시키거나 또는 제거하기 위해 시간 예측에 의존한다. 인트라-모드 (I-모드)는 여러 공간-기반의 비디오 압축 모드들 중 임의의 모드를 지칭할 수도 있다. 단방향 예측 (P-모드) 또는 양방향 예측 (B-모드)와 같은 인터-모드들은 여러 시간-기반의 비디오 압축 모드들 중 임의의 모드를 지칭할 수도 있다.

[0136] 도 2 에 나타낸 바와 같이, 비디오 데이터 메모리 (39)는 비디오 프레임 내 현재의 비디오 블록을 인코딩하는데 사용되는 비디오 데이터를 수신한다. 비디오 데이터 메모리 (39)는 (예컨대, 비디오 데이터를 저장하도록 구성된) 비디오 인코더 (20)의 구성요소들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (39)에 저장된 비디오 데이터는 예를 들어, 비디오 소스 (18)로부터 획득될 수도 있다. 디코딩된 픽처 버퍼 (DPB) (64) (또한, 참조 픽처 버퍼 또는 참조 프레임 버퍼로서 지칭됨)는 비디오 인코더 (20)에 의해 비디오 데이터를 인코딩할 때 (예컨대, 인트라 또는 인터-코딩 모드들에서는, 또한 인트라- 또는 인터-예측 코딩 모드들로서 지칭됨)에 사용하기 위해 참조 비디오 데이터를 저장하는 버퍼의 일 예이다. 비디오 데이터 메모리 (39) 및 DPB (64)는 동기식 DRAM (SDRAM)을 포함한 동적 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은, 다양한 메모리 디바이스들 중 임의의 메모리 디바이스에 의해 형성될 수도 있다. 비디오 데이터 메모리 (39) 및 DPB (64)는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 여러 예들에서, 비디오 데이터 메모리 (39)는 비디오 인코더 (20)의 다른 구성요소들과의 온칩, 또는 그들 구성요소들에 대한 오프-칩일 수도 있다.

[0137] 도 2 의 예에서, 비디오 인코더 (20)는 모션 보상 유닛 (44), 모션 추정 유닛 (42), 인트라-예측 프로세싱 유닛 (46), 디코딩된 픽처 버퍼 (DPB) (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56)을 포함한다. 도 2 에 예시된 변환 프로세싱 유닛 (52)은 잔차 데이터의 블록에 실제 변환 또는 변환의 결합들을 적용하는 유닛이며, CU 의 변환 유닛 (TU)으로서 또한 지칭될 수도 있는 변환 계수들의 블록과 혼동되지 않아야 한다. 비디오 블록 복원을 위해, 비디오 인코더 (20)는 또한 역양자화 유닛 (58), 역변환 프로세싱 유닛 (60), 및 합산기 (62)를 포함한다.

[0138] 디블록킹 필터 (63)는 블록 경계들을 디블록 필터링하여 복원된 비디오로부터 블록킹 현상 (blockiness) 아티팩트들을 제거할 수도 있다. 일부 예들에서, 디블록킹 필터 (63)는 본 개시물에서 설명되는 예시적인 기법들을 구현하고 비디오 인코더 (20)의 다른 유닛들과 조합하여 그렇게 동작하도록 구성될 수도 있다. 예를 들어, 디블록킹 필터 (63)는, 단독으로 또는 비디오 인코더 (20)의 다른 유닛들과 조합하여, 제 1 블록의 제 1 픽셀로부터 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정하고, 제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정하고, 기하학적인 거리 및 픽셀 값 차이에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링할 양을 결정하고, 그리고 디블록 필터링할 양에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링할 수도 있다.

[0139] 도 2 에 예시되지는 않지만, 비디오 인코더 (20)는 샘플 적응 오프셋 (SAO) 필터 및/또는 적응 루프 필터 (ALF)를 포함할 수도 있다. SAO 필터 및/또는 ALF 필터는 디블록킹 필터 (63) 전에 또는 디블록킹 필터 (63) 이후 로케이트될 수도 있다. 디블록킹 필터 (63)가 이웃하는 블록들 사이의 블록킹 현상을 저감하도록 구성되는 경우, SAO 필터 및/또는 ALF 필터는, 비디오 콘텐츠가 원래 비디오 콘텐츠와 동일하게 나타나도록, 또는 비디오 코딩 효율을 증가시키도록, 다른 유형들의 필터링을 수행하도록 구성될 수도 있다. 디블록킹 필터 (63)는 복원된 블록이 원래 블록과 더욱 유사하도록 픽셀 값들을 수정하기 위해, 필터링을 적용하는 것이 가능하더라도, 반드시 필터링을 적용하지는 않을 수도 있다. 대신, 디블록킹 필터 (63)는 블록킹 현상을 제거한다. 또한, SAO 및 ALF 필터들은 단지 동일한 블록의 비디오 데이터에 기초할 수도 있으며, 여기서, 디블록킹 필터 (63)는 상이한 블록들로부터의 비디오 데이터를 이용한다.

[0140] 인코딩 프로세스 동안, 비디오 인코더 (20)는 코딩될 비디오 프레임 또는 슬라이스를 수신한다. 프레임 또는 슬라이스는 최대 코딩 유닛들 (LCU들)과 같은 다수의 비디오 블록들로 분할될 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 시간 압축을 제공하기 위해 하나 이상의 참조 프레임들에서 하나 이상의 블록들에 대해 그 수신된 비디오 블록의 인터-예측 코딩을 수행한다. 인트라-예측 프로세싱 유닛 (46)은 공간 압축을 제공하기 위해 코딩될 블록과 동일한 프레임 또는 슬라이스에서 하나 이상의 이웃하는 블록들에 대해서 그 수신된 비디오 블록의 인트라-예측 코딩을 수행할 수도 있다.

[0141] 모드 선택 유닛 (40)은 코딩 모드들, 즉, 인트라 또는 인터 중 하나를, 예컨대, 각각의 모드에 대한 에러 (즉, 왜곡) 결과들에 기초하여 선택하고, 결과적인 인트라- 또는 인터-예측된 블록 (예컨대, PU)을 합산기 (50)에

제공하여 잔차 블록 데이터를 발생하고, 그리고 합산기 (62)에 제공하여 참조 프레임에서의 사용을 위해, 인코딩된 블록을 복원할 수도 있다. 합산기 (62)는 아래에서 더 자세히 설명하는 바와 같이, 그 예측된 블록을 그 블록에 대한 역변환 프로세싱 유닛 (60)으로부터의 역 양자화된, 역변환된 데이터와 결합하여, 인코딩된 블록을 복원한다. 일부 비디오 프레임들은 I-프레임들로서 지정될 수도 있으며, 여기서, I-프레임에서의 모든 블록들은 인트라-예측 모드에서 인코딩된다. 일부의 경우, 인트라-예측 프로세싱 유닛 (46)은 예컨대, 모션 추정 유닛 (42)에 의해 수행되는 모션 탐색이 그 블록의 충분한 예측을 초래하지 않을 때, P- 또는 B-프레임에서 블록의 인트라-예측 인코딩을 수행할 수도 있다.

[0142] 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 고도로 통합될 수도 있지만, 개념적인 목적들을 위해 별개로 예시된다. 모션 추정(또는, 모션 탐색)은 모션 벡터들을 발생하는 프로세스이며, 이 프로세스는 비디오 블록들에 대한 모션을 추정한다. 모션 벡터는, 예를 들어, 참조 프레임의 참조 샘플에 대한, 현재의 프레임에서의 예측 유닛의 변위를 나타낼 수도 있다. 모션 추정 유닛 (42)은 예측 유닛을 DPB (64)에 저장된 참조 프레임의 참조 샘플들과 비교함으로써 인터-코딩된 프레임의 예측 유닛에 대한 모션 벡터를 계산한다. 참조 샘플은 코딩중인 PU를 포함하는 CU의 부분에 픽셀 차이의 관점에서 가깝게 매칭하는 것으로 발견되는 블록일 수도 있으며, 이것은 SAD (sum of absolute difference), SSD (sum of squared difference), 또는 다른 차이 매트릭스들에 의해 결정될 수도 있다. 참조 샘플은 참조 프레임 또는 참조 슬라이스 내 어디서든지 발생할 수도 있으며, 반드시 참조 프레임 또는 슬라이스의 블록(예컨대, 코딩 유닛) 경계에서 발생하지 않을 수도 있다. 일부 예들에서, 참조 샘플은 분수 픽셀 위치에서 발생할 수도 있다.

[0143] 모션 추정 유닛 (42)이 참조 블록을 로케이트할 수도 있는 탐색 공간을 확장하기 위해, 모션 추정 유닛 (42)은 참조 픽처에 대해 내삽 필터링을 수행할 수도 있다. 참조 픽처에 포함되는 픽셀들은 정수 픽셀들로서 지칭되며 정수 픽셀 위치들에 로케이트된다. 내삽 필터링을 수행하는 것의 일부로서, 모션 추정 유닛 (42)은 정수 픽셀들 사이에 픽셀 값들을 내삽할(즉, 새로운 픽셀 로케이션들에 대해 새로운 픽셀 값들을 계산할) 수도 있다. 이들 내삽된 픽셀들은 서브-픽셀들 또는 분수 픽셀들로서 지칭된다. 참조 블록을 식별하기 위해, 모션 추정 유닛 (42)은 정수 픽셀들 및 분수 픽셀들 양쪽을 탐색하여 바람직한 참조 블록을 로케이트할 수도 있다.

[0144] 모션 추정 유닛 (42)은 그 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44)으로 전송한다. 모션 벡터에 의해 식별된 참조 프레임의 부분은 참조 샘플로서 지정될 수도 있다. 모션 보상 유닛 (44)은 현재의 CU의 예측 유닛에 대한 예측 값을, 예컨대, PU에 대한 모션 벡터에 의해 식별되는 참조 샘플을 취출함으로써, 계산할 수도 있다.

[0145] 인트라-예측 프로세싱 유닛 (46)은 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 수행되는 인터-예측에 대한 대안으로서, 수신된 블록을 인트라-예측할 수도 있다. 인트라-예측 프로세싱 유닛 (46)은 블록들에 대한 좌우, 상하 인코딩 순서를 가정할 때, 이웃하는, 이전에 코딩된 블록들, 예컨대, 현재의 블록의 상부, 상부 및 우측, 상부 및 좌측, 또는 좌측 블록들에 대해 그 수신된 블록을 예측할 수도 있다. 인트라-예측 프로세싱 유닛 (46)은 다양한 상이한 인트라-예측 모드들로 구성될 수도 있다. 예를 들어, 인트라-예측 프로세싱 유닛 (46)은 인코딩되는 CU의 사이즈에 기초하여, 어떤 수의 방향 예측 모드들, 예컨대, 33 개의 방향 예측 모드들로 구성될 수도 있다.

[0146] 인트라-예측 프로세싱 유닛 (46)은 인트라-예측 모드를, 예를 들어, 여러 인트라-예측 모드들에 대한 에러 값을 계산하여 최저 에러 값을 산출하는 모드를 선택함으로써, 선택할 수도 있다. 방향 예측 모드들은 공간적으로 이웃하는 픽셀들의 값을 결합하여 그 결합된 값을 PU 내 하나 이상의 픽셀 위치들에 적용하는 기능들을 포함할 수도 있다. 일단 PU 내 모든 픽셀 위치들에 대한 값들이 계산되었으면, 인트라-예측 프로세싱 유닛 (46)은 PU와 인코딩될 수신된 블록 사이의 픽셀 차이들에 기초하여, 예측 모드에 대한 에러 값을 계산할 수도 있다. 인트라-예측 프로세싱 유닛 (46)은 허용 가능한 에러 값을 산출하는 인트라-예측 모드가 발견될 때까지 인트라-예측 모드들을 테스트하는 것을 계속할 수도 있다. 인트라-예측 프로세싱 유닛 (46)은 그 후 PU를 합산기 (50)로 전송할 수도 있다.

[0147] 비디오 인코더 (20)는 코딩중인 원래 비디오 블록으로부터 모션 보상 유닛 (44) 또는 인트라-예측 프로세싱 유닛 (46)에 의해 계산된 예측 데이터를 감산함으로써 잔차 블록을 형성한다. 합산기 (50)는 이 감산 연산을 수행하는 구성요소 또는 구성요소들을 나타낸다. 잔차 블록은 픽셀 차이 값들의 2차원 매트릭스에 대응할 수도 있으며, 여기서, 잔차 블록에서의 값들의 수는 잔차 블록에 대응하는 PU 내 픽셀들의 개수와 동일하다. 잔차 블록에서의 값들은 PU에, 그리고 코딩될 원래 블록에서 병치된 픽셀들의 값들 사이의 차이들, 즉, 예

러에 대응할 수도 있다. 그 차이들은 코딩되는 블록의 유형에 의존하여 크로마 또는 루마 차이들일 수도 있다.

[0148] 변환 프로세싱 유닛 (52)은 잔차 블록으로부터 하나 이상의 TU들을 포함할 수도 있다. 변환 프로세싱 유닛 (52)은 복수의 변환들 중에서 변환을 선택한다. 이 변환은 블록 사이즈, 코딩 모드 등과 같은 하나 이상의 코딩 특성들에 기초하여 선택될 수도 있다. 변환 프로세싱 유닛 (52)은 그후 그 선택된 변환을 TU에 적용하여, 변환 계수들의 2차원 어레이를 포함하는 비디오 블록을 발생한다.

[0149] 변환 프로세싱 유닛 (52)은 결과적인 변환 계수들을 양자화 유닛 (54)으로 전송할 수도 있다. 양자화 유닛 (54)은 그후 변환 계수들을 양자화할 수도 있다. 엔트로피 인코딩 유닛 (56)은 그후 매트릭스에서의 양자화된 변환 계수들의 스캐닝을 스캐닝 모드에 따라서 수행할 수도 있다. 본 개시물은 엔트로피 인코딩 유닛 (56)을 스캐닝을 수행하는 것으로서 기술한다. 그러나, 다른 예들에서, 양자화 유닛 (54)과 같은, 다른 프로세싱 유닛들이 스캐닝을 수행할 수 있음이 이해되어야 한다.

[0150] 일단 변환 계수들이 1차원 어레이로 스캐닝되면, 엔트로피 인코딩 유닛 (56)은 CABAC, 신택스-기반 컨텍스트-적용 2진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE), 또는 또 다른 엔트로피 코딩 방법론과 같은 엔트로피 코딩을 계수들에 적용할 수도 있다. 엔트로피 인코딩 유닛 (56)은 변환 계수들을 코딩하도록 구성될 수도 있다. CABAC를 수행하기 위해, 엔트로피 인코딩 유닛 (56)은 어떤 컨텍스트에 적용할 컨텍스트 모델을 선택하여, 송신될 심볼들을 인코딩할 수도 있다. 엔트로피 인코딩 유닛 (56)에 의한 엔트로피 코딩 이후, 결과적인 인코딩된 비디오는 비디오 디코더 (30)와 같은 또 다른 디바이스로 송신되거나, 또는 추후 송신 또는 취출을 위해 아카이브될 수도 있다.

[0151] 역양자화 유닛 (58) 및 역변환 프로세싱 유닛 (60)은 역양자화 및 역변환을 각각 적용하여, 예컨대, 참조 블록으로서의 추후 사용을 위해, 픽셀 도메인에서 잔차 블록을 복원한다. 모션 보상 유닛 (44)은 잔차 블록을 DPB (64)의 프레임들 중 하나의 예측 블록에 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44)은 또한 하나 이상의 내삽 필터들을 그 복원된 잔차 블록에 적용하여, 모션 추정에 사용하기 위한 서브-정수 픽셀 값들을 계산할 수도 있다.

[0152] 합산기 (62)는 복원된 잔차 블록을 모션 보상 유닛 (44)에 의해 (어쩌면 내삽 필터링으로) 로케이트된 모션 보상된 예측 블록에 가산하여, DPB (64)에의 저장을 위한 복원된 비디오 블록을 발생한다. 복원된 비디오 블록은 후속 비디오 프레임에서 블록을 인터-코딩하기 위해 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 참조 블록으로서 사용될 수도 있다.

[0153] 이와 같이, 비디오 인코더 (20)는 경계까지의 픽셀들의 픽셀 거리들에 기초하여 픽셀들의 픽셀 값들을 디블록 필터링할 양을 결정하고 그 결정된 양에 기초하여 디블록 필터링하도록 구성되는 하나 이상의 프로세서들을 포함하는 비디오 인코더의 일 예일 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 적어도 8개의 픽셀들의 픽셀 값들을 디블록 필터링할 양을 결정할 수도 있다. 일부 예에서, 비디오 인코더 (20)는 경계에 더 가까운 픽셀들의 픽셀 값들이 경계로부터 더 멀리 떨어져 있는 픽셀들의 픽셀 값들보다 크게 수정된다고 결정할 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 경계까지의 픽셀들의 픽셀 거리들의 합수인 오프셋 값을 결정하여, 디블록 필터링할 양을 결정할 수도 있다.

[0154] 일부 예들에서, 비디오 인코더 (20)는 제 1 유형의 필터링 (예컨대, 본 개시물에서 설명된 바와 같은 긴 디블록 필터링)을 수행할지 여부를 결정할 수도 있다. 비디오 인코더 (20)는 제 1 유형의 필터링을 수행하지 않기로 결정되면, (예컨대, 베이스 HEVC 표준에서 정의된 바와 같은) 제 2 유형의 디블록 필터링을 수행할 수도 있다. 일부 예에서, 비디오 인코더 (20)는 이웃하는 블록들에 기초하여 기울기를 결정할 수도 있으며, 그 기울기들을 임계값과 비교할 수도 있다. 이러한 예들에서, 비디오 인코더 (20)는 그 비교에 기초하여 본 개시물에서 설명되는 디블록 필터링 기법들을 수행하거나, 또는 그 비교에 기초하여 디블록 필터링 기법들, 예컨대 HEVC 표준에서의 디블록 필터링 기법 또는 일부 다른 디블록 필터링 기법을 수행할 수도 있다.

[0155] 도 3은 인코딩된 비디오 시퀀스를 디코딩하는 비디오 디코더 (30)의 일 예를 예시하는 블록도이다. 비디오 디코더 (30)는 디블록 필터링을 위한 방법들을 수행하도록 구성된 디바이스의 일 예를 나타낸다. 도 3의 예에서, 비디오 데이터 메모리 (69)는 인코딩된 비디오를 수신한다. 비디오 데이터 메모리 (69)는 비디오 디코더 (30)의 구성요소들에 의해 디코딩될, 인코딩된 비디오 비트스트림과 같은, 비디오 데이터를 저장할 (예컨대, 비디오 데이터를 저장하도록 구성될) 수도 있다. 비디오 데이터 메모리 (69)에 저장된 비디오 데이터는 카메라와 같은 로컬 비디오 소스로부터, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해서, 또

는 물리적인 데이터 저장 매체들에 액세스함으로써 획득될 수도 있다. 비디오 데이터 메모리 (69)는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (CPB)를 형성할 수도 있다.

[0156] 디코딩된 픽처 버퍼 (DPB) (82)는 비디오 디코더 (30)에 의해 (예컨대, 인트라- 또는 인터-코딩 모드들에서) 비디오 데이터를 디코딩할 때에 사용하기 위해 참조 비디오 데이터를 저장한다. 비디오 데이터 메모리 (69) 및 DPB (82)는 SDRAM을 포함한 DRAM, MRAM, RRAM, 또는 다른 유형들의 메모리 디바이스들과 같은, 다양한 메모리 디바이스들 중 임의의 메모리 디바이스에 의해 형성될 수도 있다. 비디오 데이터 메모리 (69) 및 DPB (82)는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 여러 예들에서, 비디오 데이터 메모리 (69)는 비디오 디코더 (30)의 다른 구성요소들과의 온칩, 또는 그들 구성요소들에 대한 오프-칩일 수도 있다.

[0157] 비디오 디코더 (30)는 엔트로피 디코딩 유닛 (70), 모션 보상 유닛 (72), 인트라-예측 프로세싱 유닛 (74), 역양자화 유닛 (76), 역변환 프로세싱 유닛 (78), 디코딩된 픽처 버퍼 (DPB) (82) 및 합산기 (80)를 포함한다. 도 3의 비디오 디코더 (30)는 일부 예들에서, 비디오 인코더 (20) (도 2 참조)에 대해 설명되는 인코딩 과정과는 일반적으로 반대인 디코딩 과정을 수행할 수도 있다.

[0158] 엔트로피 디코딩 유닛 (70)은 인코딩된 비트스트림에 대해 엔트로피 디코딩 프로세스를 수행하여, 변환 계수들의 1차원 어레이를 취출한다. 사용되는 엔트로피 디코딩 프로세스는 비디오 인코더 (20)에 의해 사용되는 엔트로피 코딩 (예컨대, CABAC)에 의존한다. 인코더에 의해 사용되는 엔트로피 코딩 프로세스는 인코딩된 비트스트림으로 시그널링될 수도 있거나 또는 미리 결정된 프로세스일 수도 있다. 엔트로피 디코딩 유닛 (70)은 변환 계수들을 코딩하도록 구성될 수도 있다.

[0159] 일부 예들에서, 엔트로피 디코딩 유닛 (70) (또는, 역양자화 유닛 (76))은 비디오 인코더 (20)의 엔트로피 인코딩 유닛 (56) (또는, 양자화 유닛 (54))에 의해 사용되는 스캐닝 모드를 미러링하는 스캐닝을 이용하여 수신된 값들을 스캐닝할 수도 있다. 계수들의 스캐닝이 역양자화 유닛 (76)에서 수행될 수도 있지만, 스캐닝은 예시의 목적을 위해 엔트로피 디코딩 유닛 (70)에 의해 수행되는 것으로 설명될 것이다. 게다가, 예시의 용이를 위해 별개의 기능적 유닛들로서 나타내지만, 엔트로피 디코딩 유닛 (70), 역양자화 유닛 (76), 및 비디오 디코더 (30)의 다른 유닛들의 구조 및 기능은 서로 고도로 통합될 수도 있다.

[0160] 역양자화 유닛 (76)은 비트스트림으로 제공되어 엔트로피 디코딩 유닛 (70)에 의해 디코딩되는 양자화된 변환 계수들을 역양자화한다, 즉, 양자화 해제한다. 역양자화 프로세스는 종래의 프로세스, 예컨대, HEVC에 대해 제안되거나 또는 H.264 디코딩 표준에 의해 정의된 프로세스들과 유사한, 프로세스를 포함할 수도 있다. 역양자화 프로세스는 양자화의 정도를 결정하기 위해, 그리고, 이와 유사하게, 적용되어야 하는 역양자화의 정도를 결정하기 위해, CU에 대해 비디오 인코더 (20)에 의해 계산된 양자화 파라미터 QP의 사용을 포함할 수도 있다. 역양자화 유닛 (76)은 계수들이 1차원 어레이로부터 2차원 어레이로 변환되기 전후에 변환 계수들을 역양자화할 수도 있다.

[0161] 역변환 프로세싱 유닛 (78)은 역양자화된 변환 계수들에 역변환을 적용한다. 일부 예들에서, 역변환 프로세싱 유닛 (78)은 비디오 인코더 (20)로부터의 시그널링에 기초하여, 또는 블록 사이즈, 코딩 모드 등과 같은 하나 이상의 코딩 특성들로부터 변환을 추론함으로써, 역변환을 결정할 수도 있다. 일부 예들에서, 역변환 프로세싱 유닛 (78)은 현재의 블록을 포함하는 LCU에 대한 퀘드트리의 루트 노드에서의 시그널링된 변환에 기초하여 현재의 블록에 적용할 변환을 결정할 수도 있다. 이의 대안으로, 변환은 LCU 퀘드트리에서 리프-노드 CU에 대한 TU 퀘드트리의 루트에서 시그널링될 수도 있다. 일부 예들에서, 역변환 프로세싱 유닛 (78)은 캐스케이드된 역변환을 적용할 수도 있으며, 여기서, 역변환 프로세싱 유닛 (78)은 디코딩되는 현재의 블록의 변환 계수들에 2개 이상의 역변환들을 적용한다. 인트라-예측 프로세싱 유닛 (74)은 현재의 블록에 대한 디코딩된 블록들로부터의, 시그널링된 인트라-예측 모드 및 데이터에 기초하여, 현재의 블록에 대한 예측 데이터를 발생할 수도 있다.

[0162] 모션 보상 유닛 (72)은 인코딩된 비트스트림으로부터 모션 벡터, 모션 예측 방향 및 참조 인덱스를 취출할 수도 있다. 참조 예측 방향은 인터-예측 모드가 단방향 (예컨대, P 프레임) 또는 양방향 (B 프레임)인지 여부를 나타낸다. 참조 인덱스는 어느 참조 프레임에 후보 모션 벡터가 기초하는지를 나타낸다.

[0163] 취출된 모션 예측 방향, 참조 프레임 인덱스, 및 모션 벡터에 기초하여, 모션 보상 유닛은 현재의 부분에 대한 모션 보상된 블록을 발생한다. 이들 모션 보상된 블록들은 본질적으로 잔차 데이터를 발생하는데 사용되는

예측 블록을 재생성한다.

[0164]

모션 보상 유닛 (72)은 모션 보상된 블록들을, 어쩌면 모션 보상 유닛 (44)에 의해 수행된 동일한 내삽 필터링을 수행함으로써, 발생할 수도 있다. 서브-픽셀 정밀도를 갖는 모션 추정에 이용되는 내삽 필터들에 대한 식별자들이 신택스 엘리먼트들에 포함될 수도 있다. 모션 보상 유닛 (72)은 비디오 블록의 인코딩 동안 비디오 인코더 (20)에 의해 사용되는 것과 같은 내삽 필터들을 이용하여, 참조 블록의 서브-정수 픽셀들에 대한 내삽된 값들을 계산할 수도 있다. 모션 보상 유닛 (72)은 수신된 신택스 정보에 따라서 비디오 인코더 (20)에 의해 사용되는 내삽 필터들을 결정하고 그 내삽 필터들을 이용하여 예측 블록들을 발생할 수도 있다.

[0165]

게다가, HEVC 예에서, 모션 보상 유닛 (72) 및 인트라-예측 프로세싱 유닛 (74)은 (예컨대, 큐드트리에 의해 제공되는) 신택스 정보의 일부를 이용하여, 인코딩된 비디오 시퀀스의 프레임(들)을 인코딩하는데 사용되는 LCU들의 사이즈들을 결정할 수도 있다. 모션 보상 유닛 (72) 및 인트라-예측 프로세싱 유닛 (74)은 또한 신택스 정보를 이용하여, 인코딩된 비디오 시퀀스의 프레임의 각각의 CU가 어떻게 분할되는지 (그리고, 유사하게, 서브-CU들이 어떻게 분할되는지)를 기술하는 분할 정보를 결정할 수도 있다. 신택스 정보는 또한 인코딩된 비디오 시퀀스를 디코딩하기 위해, 어떻게 각각의 분할이 인코딩되는지를 나타내는 모드들 (예컨대, 인트라- 또는 인터-예측, 그리고 인트라-예측에 있어, 인트라-예측 인코딩 모드), 각각의 인터-인코딩된 PU에 대한 하나 이상의 참조 프레임들 (및/또는 참조 프레임들에 대한 식별자들을 포함하는 참조 리스트들), 및 다른 정보를 포함할 수도 있다.

[0166]

합산기 (80)는 잔차 블록들을 모션 보상 유닛 (72) 또는 인트라-예측 프로세싱 유닛 (74)에 의해 발생된 대응하는 예측 블록들과 결합하여, 디코딩된 블록들을 형성한다. 블록킹 현상 아티팩트들을 제거하기 위해 디블록킹 필터 (81)가 또한 그 디코딩된 블록들을 필터링하는데 적용될 수도 있다. 디블록킹 필터 (81)는 본 개시물에서 설명되는 예시적인 디블록 필터링 기법들을 구현하도록 구성될 수도 있다. 예를 들어, 디블록킹 필터 (81)는, 단독으로 또는 다른 구성요소들과 조합하여, 제 1 블록의 제 1 픽셀로부터 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정하고, 제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정하고, 기하학적인 거리 및 픽셀 값 차이에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링 할 양을 결정하고, 그리고 디블록 필터링 할 양에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링할 수도 있다.

디코딩된 비디오 블록들은 그후 DPB (82)에 저장될 수도 있으며, 그 DPB (82)는 후속 모션 보상을 위해 참조 블록들을 제공하며, 또한 (도 1의 디스플레이 디바이스 (32)와 같은) 디스플레이 디바이스 상에의 제시를 위해 디코딩된 비디오를 발생한다.

[0167]

도 2 와 유사하게, 도 3에 예시되지는 않지만, 비디오 디코더 (30)는 샘플 적응 오프셋 (SAO) 필터 및/또는 적응 루프 필터 (ALF)를 포함할 수도 있다. SAO 필터 및/또는 ALF 필터는 디블록킹 필터 (81) 이전에 또는 디블록킹 필터 (81) 이후에 로케이트될 수도 있다. 디블록킹 필터 (81)가 이웃하는 블록들 사이의 블록킹 현상을 저감하도록 구성되는 경우, SAO 필터 및/또는 ALF 필터는, 비디오 콘텐츠가 원래 비디오 콘텐츠와 동일하게 나타나도록, 또는 비디오 코딩 효율을 증가시키도록, 다른 유형들의 필터링을 수행하도록 구성될 수도 있다. 디블록킹 필터 (81)는 복원된 블록이 원래 블록과 더욱 유사하도록 픽셀 값들을 수정하기 위해, 필터링을 적용하는 것이 가능하더라도, 반드시 필터링을 적용하지 않을 수도 있다. 대신, 디블록킹 필터 (81)는 블록킹 현상을 제거한다. 또한, SAO 및 ALF 필터들은 단지 동일한 블록의 비디오 데이터에 기초할 수도 있으며, 여기서, 디블록킹 필터 (81)는 상이한 블록들로부터의 비디오 데이터를 이용한다.

[0168]

이와 같이, 비디오 디코더 (30)는 경계까지의 픽셀들의 픽셀 거리들에 기초하여 픽셀들의 픽셀 값들을 디블록 필터링 할 양을 결정하고 그 결정된 양에 기초하여 디블록 필터링하도록 구성된 하나 이상의 프로세서들을 포함하는 비디오 디코더의 일 예일 수도 있다. 일부 예들에서, 비디오 디코더 (30)는 적어도 8개의 픽셀들의 픽셀 값들을 디블록 필터링 할 양을 결정할 수도 있다. 일부 예에서, 비디오 디코더 (30)는 경계에 더 가까운 픽셀들의 픽셀 값들이 경계로부터 더 멀리 떨어져 있는 픽셀들의 픽셀 값들보다 크게 수정된다고 결정할 수도 있다. 일부 예들에서, 비디오 디코더 (30)는 경계까지의 픽셀들의 픽셀 거리들의 함수인 오프셋 값을 결정하여, 디블록 필터링 할 양을 결정할 수도 있다.

[0169]

일부 예들에서, 비디오 디코더 (30)는 제 1 유형의 필터링 (예컨대, 본 개시물에서 설명된 바와 같은 긴 디블록 필터링)을 수행할지 여부를 결정할 수도 있다. 비디오 디코더 (30)는 제 1 유형의 필터링을 수행하지 않기로 결정되면 (예컨대, 베이스 HEVC 표준에서 정의된 바와 같은) 제 2 유형의 디블록 필터링을 수행할 수도 있다. 일부 예에서, 비디오 디코더 (30)는 이웃하는 블록들에 기초하여 기울기를 결정할 수도 있으며, 그 기울기들을 임계값과 비교할 수도 있다. 이러한 예들에서, 비디오 디코더 (30)는 그 비교에 기초하여 본

개시물에서 설명되는 디블록 필터링 기법들을 수행하거나, 또는 그 비교에 기초하여 HEVC 표준에서의 디블록 필터링 기법들과 같은 디블록 필터링 기법들 또는 일부 다른 디블록 필터링 기법을 수행할 수도 있다.

[0170] 도 5 는 본 개시물에서 설명되는 기법들에 기초하여 정의된 예시적인 디블록킹 필터 (100) 의 구성요소들을 예시하는 블록도이다. 일반적으로, 도 2 의 디블록킹 필터 (63) 및 도 3 의 디블록킹 필터 (81) 중 어느 하나 또는 양자는 디블록킹 필터 (100) 의 구성요소들과 실질적으로 유사한 구성요소들을 포함할 수도 있다. 비디오 인코더들, 비디오 디코더들, 비디오 인코더/디코더들 (코덱들), 및 기타 등등과 같은, 다른 비디오 코딩 디바이스들은 또한 디블록킹 필터 (100) 와 실질적으로 유사한 구성요소들을 포함할 수도 있다. 디블록킹 필터 (100) 는 하드웨어, 소프트웨어, 또는 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어 또는 펌웨어로 구현될 때, (하나 이상의 프로세서들 또는 프로세싱 유닛들 및 소프트웨어 또는 펌웨어에 대한 명령들을 저장하기 위한 메모리와 같은) 대응하는 하드웨어가 또한 제공될 수도 있다.

[0171] 도 5 의 예에서, 디블록킹 필터 (100) 는 디블록킹 결정 유닛 (104), 메모리에 저장된 서포트 정의들 (102), 디블록킹 필터링 유닛 (106), 메모리에 저장된 디블록킹 필터 정의들 (108), 에지 로케이팅 유닛 (103), 및 에지 로케이션들 데이터 구조 (105) 를 포함한다. 디블록킹 필터 (100) 의 구성요소들 중 임의의 구성요소 또는 그 모두가 기능적으로 통합될 수도 있다. 디블록킹 필터 (100) 의 구성요소들은 단지 예시의 목적을 위해 별개로 예시된다. 일반적으로, 디블록킹 필터 (100) 는 예컨대, 예측 데이터를 블록들에 대한 잔차 데이터와 결합하는 합산 구성요소로부터, 디코딩된 블록들 (또는, 복원된 블록들) 에 대한 데이터를 수신한다. 데이터는 블록들이 예측된 방법의 표시를 더 포함할 수도 있다. 아래에서 설명하는 예에서, 디블록킹 필터 (100) 는 LCU 및 그 LCU 에 대한 CU 쿼드트리와 연관된 디코딩된 비디오 블록을 포함하는 데이터를 수신하도록 구성되며, 여기서, CU 쿼드트리는 LCU 가 CU들로 파티셔닝되는 방법 및 리프-노드 CU들의 PU들 및 TU들에 대한 예측 모드들을 기술한다.

[0172] 디블록킹 필터 (100) 는 에지 로케이션들 데이터 구조 (105) 를 디블록킹 필터 (100) 의 메모리에, 또는 대응하는 비디오 코딩 디바이스에 의해 제공되는 외부 메모리에 유지할 수도 있다. 일부 예들에서, 에지 로케이팅 유닛 (103) 은 LCU 가 CU들로 파티셔닝되는 방법을 표시하는, LCU 에 대응하는 CU 쿼드트리를 수신할 수도 있다. 에지 로케이팅 유닛 (103) 은 그후 CU 쿼드트리를 분석하여, 디블록킹을 위한 후보들인 LCU 에서 CU 들의 TU들 및 PU들과 연관된 디코딩된 비디오 블록들 사이의 에지들 (예컨대, 경계들) 을 결정할 수도 있다. 예를 들어, 에지 로케이팅 유닛 (103) 은 경계 (86) 를 로케이트할 수도 있다.

[0173] 에지 로케이션들 데이터 구조 (105) 는 수평 치수, 수직 치수, 및 수평 에지들 및 수직 에지들을 표시하는 치수를 가지는 어레이를 포함할 수도 있다. 일반적으로, 비디오 블록들 사이의 에지들 (예컨대, 경계들) 은 LCU 의 가장 작은-사이즈의 CU들, 또는 CU들의 TU들 및 PU들과 연관된 2개의 비디오 블록들 사이에 발생할 수도 있다. LCU 가 NxN 의 사이즈를 갖는다고 가정하면, 그리고 LCU 의 가장 작은-사이즈의 CU 가 사이즈 MxM 을 갖는다고 가정하면, 어레이에는 [N/M]x[N/M]x2 의 사이즈를 포함할 수도 있으며, 여기서, "2" 는 CU들 사이의 에지들의 2개의 가능한 방향들 (수평 및 수직) 을 나타낸다. 예를 들어, LCU 가 64x64 픽셀들 및 8x8 가장 작은-사이즈의 CU 를 갖는다고 가정하면, 어레이에는 [8]x[8]x[2] 엔트리들을 포함할 수도 있다.

[0174] 각각의 엔트리는 일반적으로 2개의 비디오 블록들 사이의 가능한 에지에 대응할 수도 있다. 에지들은 에지 로케이션들 데이터 구조 (105) 의 엔트리들의 각각에 대응하는 LCU 내 위치들의 각각에 실제로 존재하지 않을 수도 있다. 따라서, 데이터 구조의 값들은 거짓 (false) 으로 초기화될 수도 있다. 일반적으로, 에지 로케이팅 유닛 (103) 은 CU 쿼드트리를 분석하여, LCU 의 CU들의 TU들 및 PU들과 연관된 2개의 비디오 블록들 사이의 에지들의 로케이션들을 결정하고 에지 로케이션들 데이터 구조 (105) 에서의 대응하는 값들을 참으로 설정할 수도 있다.

[0175] 일반적으로, 어레이의 엔트리들은 대응하는 에지가 LCU 에 디블록킹을 위한 후보로서 존재하는지 여부를 기술한다. 즉, LCU 의 CU들의 TU들 및 PU들과 연관되는 2개의 이웃하는 비디오 블록들 사이의 에지가 존재한다고 에지 로케이팅 유닛 (103) 이 결정할 때, 에지 로케이팅 유닛 (103) 은 에지 로케이션들 데이터 구조 (105) 에서의 대응하는 엔트리의 값을, 에지가 존재한다는 것을 표시하도록 (예컨대, "참" 의 값으로) 설정할 수도 있다.

[0176] 디블록킹 결정 유닛 (104) 은 일반적으로 2개의 이웃하는 블록들에 대해, 2개의 블록들 사이의 에지가 디블록킹되어야 하는지 여부를 결정한다. 디블록킹 결정 유닛 (104) 은 에지 로케이션들 데이터 구조 (105) 를 이용하여 에지들의 로케이션들을 결정할 수도 있다. 에지 로케이션들 데이터 구조 (105) 의 값이 불린 (Boolean) 값을 가질 때, 디블록킹 결정 유닛 (104) 은 일부 예들에서 "참" 값은 에지의 존재를 표시하고

"거짓" 값은 어떤 에지도 존재하지 않는다는 것을 표시한다고 결정할 수도 있다. 예를 들어, 디블록킹 결정 유닛 (104)은 기울기를 결정하고 본 개시물에서 설명하는 기법들을 이용하여 디블록 필터링할지 여부를 결정할 수도 있다. 다른 예로서, 디블록킹 결정 유닛 (104)은 변환 사이즈, 및 CU가 인트라-코딩되거나 또는 인트라-코딩되지 않는지 여부를 결정하고, 이러한 정보를 이용하여 본 개시물에서 설명하는 기법들을 디블록 필터링할지 여부를 결정할 수도 있다.

[0177] 일반적으로, 디블록킹 결정 유닛 (104)은 하나 이상의 디블록킹 결정 기능부들로 구성된다. 기능부들은 블록들 사이의 에지를 가로지르는 픽셀들의 라인들에 적용되는 복수의 계수들을 포함할 수도 있다. 예를 들어, 그 기능부들은 에지에 수직한 8개의 픽셀들의 라인에 적용될 수도 있으며, 여기서, 픽셀들 중 4개는 2개의 블록들 중 하나에 존재하며 다른 4개의 픽셀들은 2개의 블록들 중 다른 하나에 존재한다. 서포트 정의들 (102)은 기능부들에 대한 서포트를 정의한다. 일반적으로, "서포트"는 기능부들이 적용되는 픽셀들에 대응한다. 서포트의 세트들의 여러 예들이 도 6에 대해 아래에서 더 자세히 설명된다.

[0178] 디블록킹 결정 유닛 (104)은 하나 이상의 디블록킹 결정 기능부들을 서포트 정의들 (102)에 의해 정의된 바와 같은, 하나 이상의 서포트의 세트들에 적용하여, 비디오 데이터의 2개의 블록들 사이의 특정의 에지가 블록킹되어야 하는지 여부를 결정하도록 구성될 수도 있다. 디블록킹 결정 유닛 (104)로부터 유래하는 파선은 필터링됨이 없이 출력되고 있는 블록들에 대한 데이터를 나타낸다. 2개의 블록들 사이의 에지가 필터링되지 않아야 한다고 디블록킹 결정 유닛 (104)이 결정하는 경우들에서, 디블록킹 필터 (100)는 데이터를 변경함이 없이 블록들에 대한 데이터를 출력할 수도 있다. 즉, 데이터는 디블록킹 필터링 유닛 (106)을 우회할 수도 있다. 한편, 에지가 디블록킹되어야 한다고 디블록킹 결정 유닛 (104)이 결정할 때, 디블록킹 결정 유닛 (104)은 디블록킹 필터링 유닛 (106)으로 하여금 에지를 디블록킹하기 위해 에지에 가까운 픽셀들에 대한 값을 필터링하게 할 수도 있다.

[0179] 디블록킹 필터링 유닛 (106)은 디블록킹 결정 유닛 (104)에 의해 표시되는 바와 같은, 디블록킹될 에지들에 대한 디블록킹 필터 파라미터들 (108)로부터 디블록킹 필터들의 정의들을 취출한다. 일반적으로, 에지의 필터링은 디블록킹될 현재의 에지의 이웃으로부터의 픽셀들의 값들을 이용한다. 따라서, 양쪽의 디블록킹 결정 기능부들 및 디블록킹 필터들은 에지의 양면 상에 어떤 서포트 영역을 가질 수도 있다. 디블록킹 필터를 에지 (예컨대, 경계)의 이웃에 있는 픽셀들에 적용함으로써, 디블록킹 필터링 유닛 (106)은 에지에 가까운 높은 주파수 전이들이 약화되도록 픽셀들의 값들을 평활화할 수도 있다. 이와 같이, 에지 (예컨대, 경계)에 가까운 픽셀들에의 디블록킹 필터들의 적용은 에지 가까이에서 블록킹 현상 아티팩트들을 감소시킬 수도 있다.

[0180] 도 6은 서브-블록 (130)과 서브-블록 (132) 사이의 비디오 블록의 에지 (즉, 경계) (134)에 가까운 픽셀 위치들을 예시하는 개념도이다. 일 예로서, 경계 (134)는 CU에 정의된 2개의 TU들 사이의 TU 에지, 또는 CU에 정의된 2개의 PU들 사이의 정의된 PU 에지와 같은, 내측 CU 에지를 포함할 수도 있다. 픽셀 위치들의 각각은 포맷 $[p|q]I_J$ 을 이용하여 지정되며, 여기서, p는 서브-블록 (130)에 대응하고 q는 서브-블록 (132)에 대응하며, I는 에지 (134)로부터의 거리에 대응하며, J는 서브-블록들 (130 및 132)의 상부로부터 저부까지의 로우 표시자 (row indicator)에 대응한다. 일부 예들에서, 디블록킹 결정 기능부들 및 디블록킹 필터들에 사용되는 서포트는 8개의 픽셀들의 라인을 갖는다. 이러한 예들에서, $0 \leq X \leq 7$ 인 주어진 라인 X에 대해, 픽셀들 p_{3X} 내지 q_{3X} 의 각각이 서포트로서 사용될 수도 있다.

[0181] 도 7은 비디오 데이터를 프로세싱하는 예시적인 기법을 예시하는 플로우차트이다. 예시의 목적을 위해, 도 7의 예는 도 6에 관하여 자세히 설명된다. 예를 들어, 설명의 용이성을 위해, 도 7은 픽셀 p_{4_3} 이 디블록 필터링될 픽셀인 것으로 설명된다. 또한, 도 7의 예는 비디오 코더가 기법들을 수행하는 것으로 설명된다. 비디오 코더는 비디오 인코더 (20) 또는 비디오 디코더 (30)를 포함적으로 기술하기 위해 사용된다.

[0182] 비디오 코더는 제 1 블록의 제 1 픽셀로부터 제 1 블록과 제 2 블록 사이의 경계까지의 기하학적인 거리를 결정할 수도 있다 (140). 예를 들어, 비디오 코더는 경계 (134)까지의 블록 (130)의 픽셀 p_{4_3} 의 기하학적인 거리를 결정할 수도 있으며, 여기서, 경계 (134)는 블록 (130)과 블록 (132) 사이의 경계이다. 일 예로서, 비디오 코더는 제 1 픽셀과 경계 사이의 픽셀들의 수에 기초하여 기하학적인 거리를 결정할 수도 있다. 예를 들어, 픽셀 p_{4_3} 의 기하학적인 거리는 $n(i) = i$ 인 방정식에 기초하여 4이거나, 또는 $n(i) = (2i + 1)/2$ 인 방정식에 기초하여 9/2이다.

[0183] 비디오 코더는 제 2 픽셀의 픽셀 값 및 제 3 픽셀의 픽셀 값에 기초하여 픽셀 값 차이를 결정할 수도 있다

(142). 예를 들어, 비디오 코더는 제 2 픽셀의 픽셀 값과 제 3 픽셀의 제 1 픽셀 값 사이의 차이를 결정하고, 그 차이의 결과를 2로 나눌 수도 있다.

[0184] 일부 예들에서, 제 2 픽셀은 경계에 가장 가까운 제 1 블록의 픽셀이며 (예컨대, 블록 (130)의 p_{0_3} 이 경계 (134)에 가장 가깝다), 제 3 픽셀은 경계에 가장 가까운 제 2 블록의 픽셀이다 (예컨대, 블록 (132)의 q_{0_3} 이 경계 (134)에 가장 가깝다). 일부 예들에서, 제 2 픽셀 및 제 1 픽셀은 동일하며, 제 3 픽셀은 제 1 블록에서의 제 1 픽셀과, 경계까지 둘거리인 제 2 블록에서의 픽셀이다 (예컨대, q_{4_3} 은 경계 (134)까지의 p_{4_3} 과, 경계 (134)까지 둘거리이다). 다른 예로서, 제 2 픽셀 및 제 1 픽셀은 동일하며, 제 3 픽셀은 경계에 가장 가까운 픽셀이다 (예컨대, q_{0_3} 이 경계 (134)에 가장 가깝다). 일부 예들에서, 제 2 및 제 3 픽셀들은 이웃하는 픽셀들 (예컨대, p_{5_3} 및 p_{3_3})이다.

[0185] 비디오 코더는 기하학적인 거리 및 픽셀 값 차이에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링할 양을 결정할 수도 있다 (144). 비디오 코더는 위에서 설명된 예시적인 기법들 중 하나 이상을 이용하여 픽셀 값 차이를 결정하고, 적어도 픽셀 값 차이에 대한 가중치를 결정할 수도 있다. 이들 예들에서, 비디오 코더는 기하학적인 거리, 픽셀 값 차이, 및 픽셀 값 차이에 대한 가중치에 기초하여, 디블록 필터링할 양을 결정할 수도 있다. 일부 예들에서, 비디오 코더는 기하학적인 거리, 픽셀 값 차이, 및 디블록 필터링될 픽셀들의 수 (픽셀들의 개수는 비트스트림으로 시그널링되거나 또는 사전-저장될 수도 있다)에 기초하여 디블록 필터링할 양을 결정할 수도 있다.

[0186] 비디오 코더는 디블록 필터링할 양에 기초하여 제 1 픽셀의 픽셀 값을 디블록 필터링할 수도 있다 (146). 일부의 경우, 비디오 코더는 디블록 필터링할 양이 제 1 임계값보다 크거나 또는 제 2 임계값 미만이라고 결정하고, 제 1 임계값보다 크거나 또는 제 2 임계값 미만인 디블록 필터링할 양에 기초하여, 디블록 필터링할 조정된 양을 결정할 수도 있다. 비디오 코더는 그 디블록 필터링할 조정된 양에 기초하여, 픽셀 값을 디블록 필터링할 수도 있다.

[0187] 더욱이, 일부 예들에서, 비디오 코더는 본 개시물에서 설명하는 기법들을 이용하여, 디블록 필터링할지 여부를 결정할 수도 있다. 예를 들어, 비디오 코더는 제 1 블록의 부분에 대한 기울기 값을 결정할 수도 있으며, 여기서, 기울기 값은 제 1 블록의 부분의 평활도를 나타낸다. 그 부분의 예들은 디블록 필터링될 블록의 영역들보다 추가적인 픽셀들을 포함하는 블록의 영역, 블록의 라인, 또는 심지어 블록의 픽셀을 포함한다. 비디오 코더는 임계치 기울기 값 미만이거나 또는 동일한 기울기 값에 기초하여, 픽셀 값을 디블록 필터링할 수도 있다.

[0188] 다른 예로서, 비디오 코더는 제 1 블록을 발생시키는데 사용되는 제 1 변환 블록의 제 1 블록 사이즈 및 제 2 블록을 발생시키는데 사용되는 제 2 변환 블록의 제 2 블록 사이즈를 결정할 수도 있다. 비디오 코더는 최대 허용가능 변환 블록 사이즈와 동일한 제 1 블록 사이즈 및 제 2 블록 사이즈 중 더 큰 사이즈, 또는 임계치 블록 사이즈보다 큰 제 1 블록 사이즈 및 제 2 블록 사이즈 중 더 작은 사이즈에 기초하여, 픽셀 값을 디블록 필터링할 수도 있다.

[0189] 일부 예들에서, 비디오 코더는 디블록 필터링된 픽셀 값을 후속 블록 (예컨대, 제 3 블록)의 인터-예측 또는 인트라-예측의 부분으로서 이용할 수도 있다. 비디오 코더가 비디오 인코더 (20)인 예들에서, 비디오 인코더 (20)는 제 3 블록에서의 픽셀의 픽셀 값과 디블록 필터링된 픽셀 값 사이의 차이를 (예컨대, 제 3 블록을 인터-예측 또는 인트라-예측 인코딩하는 것의 일부로서) 결정할 수도 있다. 비디오 인코더 (20)는 제 3 블록에서의 픽셀의 픽셀 값과 디블록 필터링된 픽셀 값 사이의 차이를 표시하는 정보를 시그널링할 수도 있으며, 여기서, 시그널링된 정보는 제 3 블록을 디코딩하는데 사용된다.

[0190] 비디오 코더가 비디오 디코더 (30)이면, 비디오 디코더 (30)는 제 3 블록에서의 픽셀의 픽셀 값과 디블록 필터링된 픽셀 값 사이의 차이를 표시하는 정보를 수신할 수도 있다. 비디오 디코더 (30)는 수신된 정보에 기초하여 제 3 블록을 인터-예측 또는 인트라-예측 디코딩할 수도 있다.

[0191] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터-판독가능 매체 상에 저장되거나 또는 컴퓨터-판독가능 매체를 통해서 송신될 수도 있으며, 하드웨어-기반의 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체들, 또는 예를 들어, 통신 프로토콜에 따라 일 장소로부터 다른 장소로의 컴퓨터

프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이런 방법으로, 컴퓨터-판독가능 매체들은 일반적으로 (1) 비일시성 유형의 컴퓨터-판독가능 저장 매체, 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에서 설명하는 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터-판독가능 매체를 포함할 수도 있다.

[0192] 일 예로서, 이에 한정하지 않고, 이런 컴퓨터-판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광디스크 스토리지, 자기디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터-판독가능 매체로 적절히 지칭된다. 예를 들어, 동축 케이블, 광섬유 케이블, 연선, 디지털가입자 회선 (DSL), 또는 무선 기술들, 예컨대 적외선, 라디오, 및 마이크로파를 이용하여 명령들이 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되는 경우, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 무선 기술들 예컨대 적외선, 라디오, 및 마이크로파가 그 매체의 정의에 포함된다. 그러나, 컴퓨터-판독가능 저장 매체 및 데이터 저장 매체는 접속부들, 반송파들, 신호들, 또는 다른 일시성 매체를 포함하지 않고, 그 대신, 비-일시성 유형의 저장 매체로 송신되는 것으로 해석되어야 한다.

디스크 (disk) 및 디스크 (disc) 는, 본원에서 사용할 때, 컴팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 Blu-ray 디스크를 포함하며, 디스크들 (disks) 은 데이터를 자기적으로 보통 재생하지만, 디스크들 (disks) 은 레이저로 데이터를 광학적으로 재생한다. 앞에서 언급한 것들의 결합들이 또한 컴퓨터-판독가능 매체들의 범위 내에 포함되어야 한다.

[0193] 명령들은 하나 이상의 DSP들, 범용 마이크로프로세서들, ASIC들, FPGA들, 또는 다른 등가의 접속 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 용어 "프로세서"는, 본원에서 사용될 때 전술한 구조 중 임의의 구조 또는 본원에서 설명하는 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 게다가, 일부 양태들에서, 본원에서 설명하는 기능은 인코딩 및 디코딩을 위해 구성되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱에 포함될 수도 있다. 또한, 이 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들로 전적으로 구현될 수 있다.

[0194] 이 예에 따라서, 본원에서 설명하는 방법들 중 임의의 방법의 어떤 행위들 또는 이벤트들은 상이한 시퀀스로 수행될 수 있으며, 모두 함께 추가하거나, 병합하거나, 또는 삭제할 수도 있는 (예컨대, 모든 설명된 행위들 또는 이벤트들이 방법의 실시에 필요한 것은 아닌) 것으로 인식되어야 한다. 더욱이, 어떤 실시형태들에서, 행위들 또는 이벤트들은 순차적으로 보다는, 동시에, 예컨대, 멀티-쓰레드된 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통해서 수행될 수도 있다.

[0195] 당업자들은 본원에서 개시된 방법들, 시스템들, 및 장치들과 관련하여 설명되는 여러가지 예시적인 로직 블록들, 모듈들, 회로들 및 알고리즘 단계들이 전자적 하드웨어, 프로세서에 의해 실행되는 컴퓨터 소프트웨어, 또는 양쪽의 조합들로서 구현될 수도 있음을 알 수 있을 것이다. 이러한 하드웨어와 소프트웨어의 상호 교환가능성을 명확히 예시하기 위하여, 이상에서는, 여러 예시적인 구성요소들, 블록들, 모듈들, 회로들 및 단계들을 그들의 기능의 관점에서 일반적으로 설명되었다. 이런 기능이 하드웨어 또는 소프트웨어로 구현되는지 여부는 특정의 애플리케이션 및 전체 시스템에 부과되는 설계 제한 사항들에 의존한다. 숙련자들은 각각의 특정의 애플리케이션 마다 설명한 기능을 여러가지 방법으로 구현할 수도 있으며, 그러나 이런 구현 결정들이 본 발명의 범위로부터 일탈을 초래하는 것으로 해석되어서는 안된다.

[0196] 더욱이, 본원에서 개시되는 예들은 범용 프로세서, DSP, ASIC, FPGA 또는 다른 프로그래밍가능 로직 디바이스, 이산 게이트 또는 트랜지스터 로직, 이산 하드웨어 구성요소들, 또는 본원에서 설명되는 기능들을 수행하도록 설계된 이들의 임의의 조합과 같은, 전자 디바이스 또는 회로로 구현되거나 또는 수행될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있으며, 그러나 대안적으로는, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로제어기 또는 상태 머신일 수도 있다. 프로세서는 또한 컴퓨팅 디바이스들의 조합, 예컨대, DSP 와 마이크로프로세서의 조합, 복수의 마이크로프로세서들, DSP 코어와 결합된 하나 이상의 마이크로프로세서들, 또는 임의의 다른 이러한 구성으로서 구현될 수도 있다.

[0197] 본원에서 개시된 실시형태들과 관련하여 설명되는 방법 또는 알고리즘의 단계들은 하드웨어로 직접, 프로세서에 의해 실행되는 소프트웨어 모듈로, 또는 이 둘의 조합으로 구현될 수도 있다. 소프트웨어 모듈은 RAM 메모리, 플래시 메모리, ROM 메모리, EPROM 메모리, EEPROM 메모리, 레지스터들, 하드 디스크, 착탈식 디스크, CD-

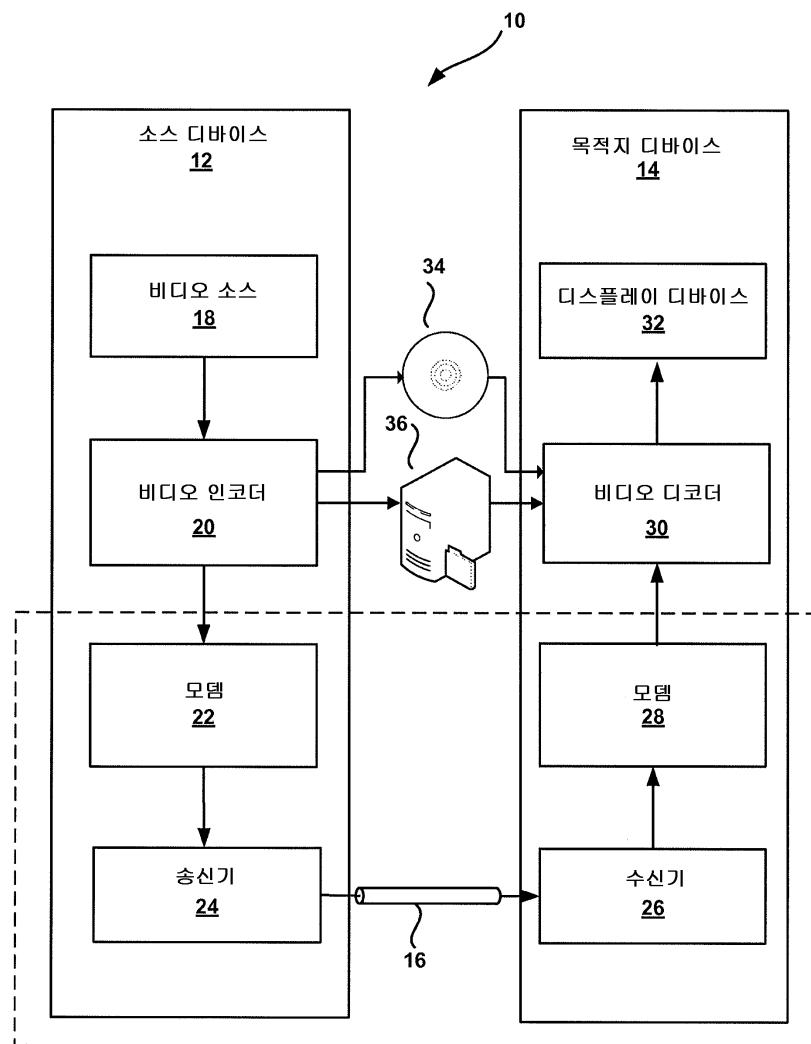
ROM, 또는 당업계에 알려져 있는 임의의 다른 유형의 저장 매체에 상주할 수도 있다. 예시적인 저장 매체는 프로세서가 저장 매체로부터 정보를 판독하고 그에 정보를 기록할 수 있도록 프로세서에 커플링된다. 대안적으로는, 저장 매체는 프로세서에 통합될 수도 있다. 프로세서 및 저장 매체는 ASIC에 상주할 수도 있다. ASIC은 사용자 단말에 상주할 수도 있다. 대안적으로는, 프로세서 및 저장 매체는 사용자 단말에 별개의 구성요소들로서 상주할 수도 있다.

[0198]

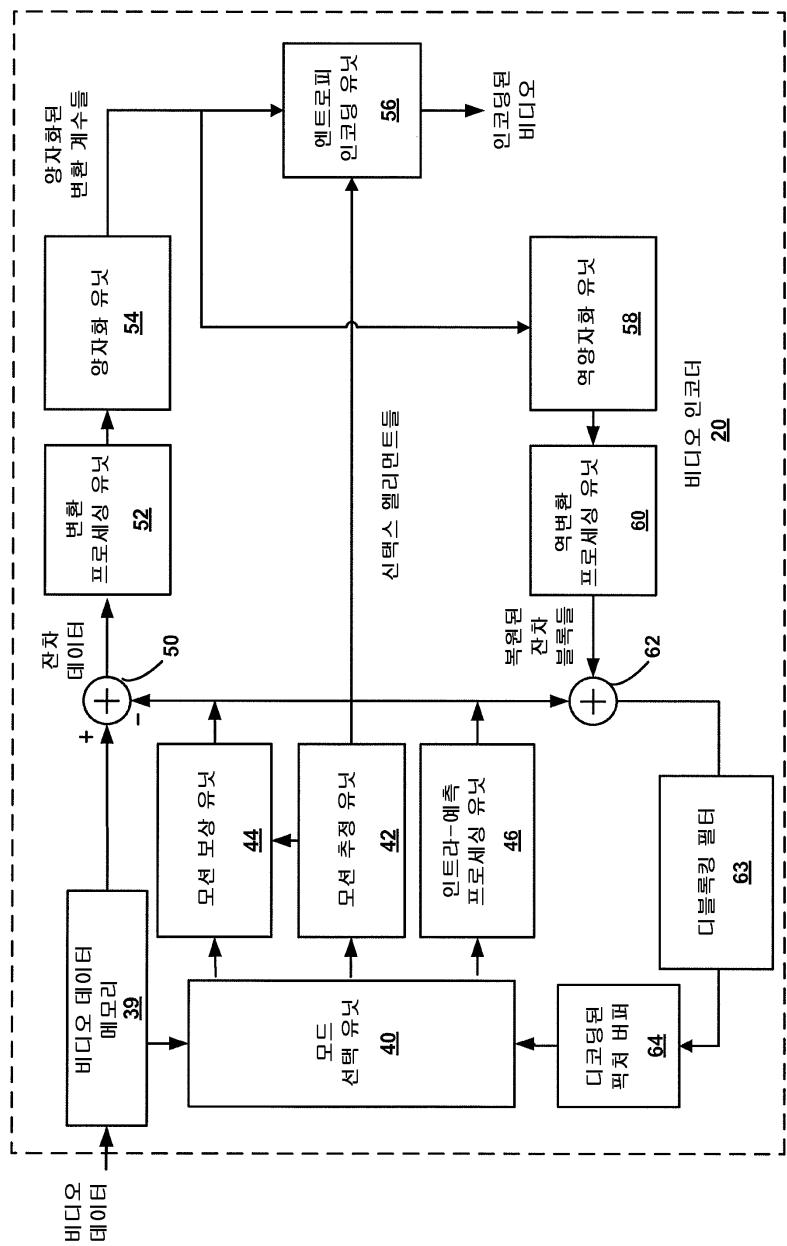
본 개시물의 기법들은 무선 헨드셋, 접적 회로 (IC) 또는 IC들의 세트 (예컨대, 칩 세트)를 포함한, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 개시한 기법들을 수행하도록 구성되는 디바이스들의 기능적 양태들을 강조하기 위해서 여러 구성요소들, 모듈들, 또는 유닛들이 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하지는 않는다. 대신, 위에서 설명한 바와 같이, 여러 유닛들이 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명한 바와 같은 하나 이상의 프로세서들을 포함한, 상호작용하는 하드웨어 유닛들의 컬렉션으로 제공될 수도 있다.

도면

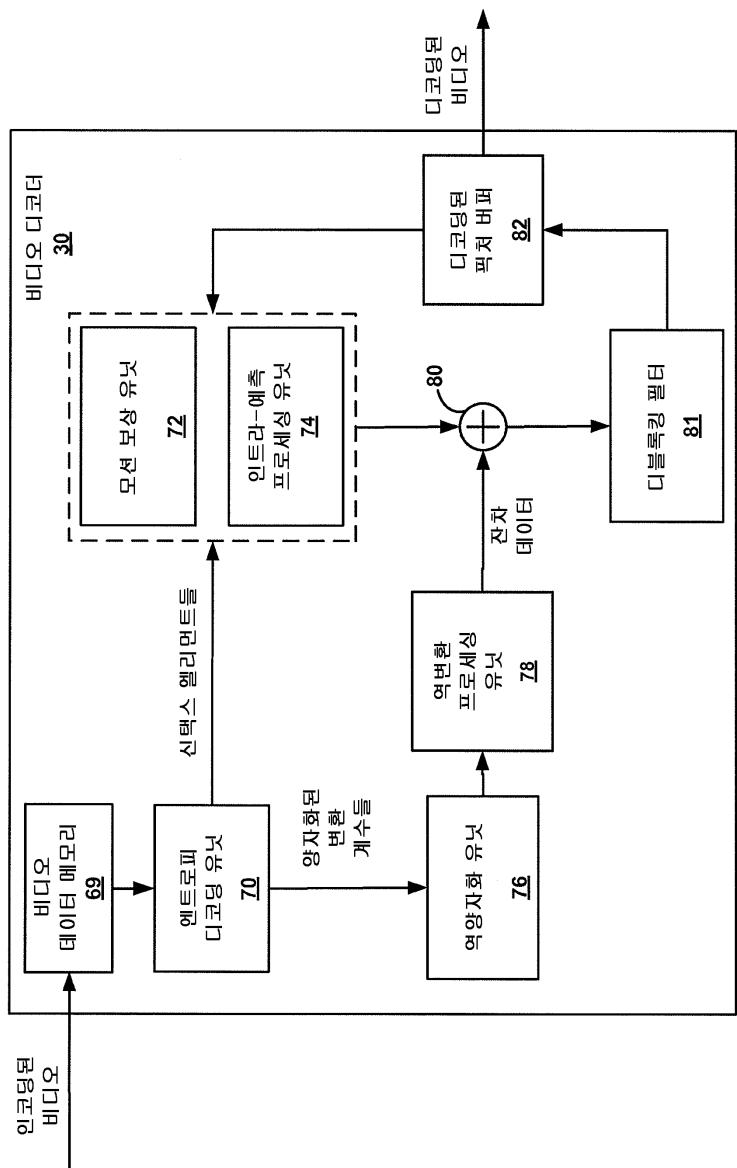
도면1



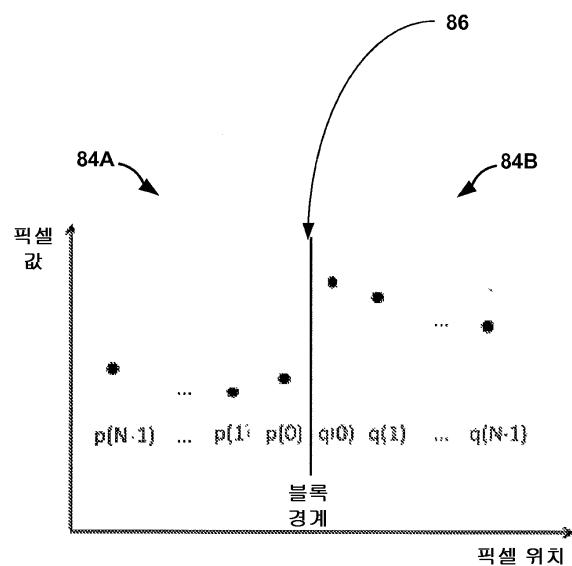
도면2



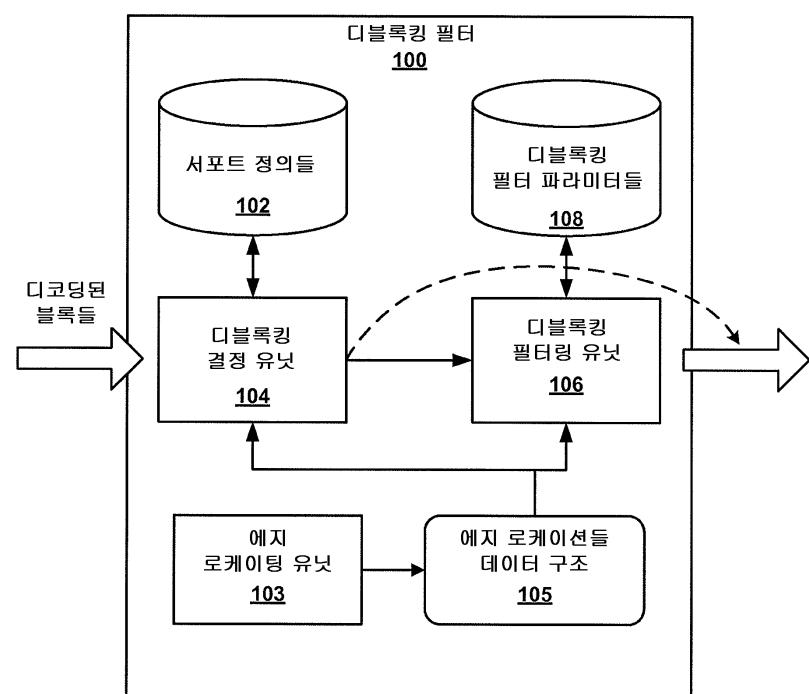
도면3



도면4



도면5



도면6

p7 ₀	p6 ₀	p5 ₀	p4 ₀	p3 ₀	p2 ₀	p1 ₀	p0 ₀	q0 ₀	q1 ₀	q2 ₀	q3 ₀	q4 ₀	q5 ₀	q6 ₀	q7 ₀
p7 ₁	p6 ₁	p5 ₁	p4 ₁	p3 ₁	p2 ₁	p1 ₁	p0 ₁	q0 ₁	q1 ₁	q2 ₁	q3 ₁	q4 ₁	q5 ₁	q6 ₁	q7 ₁
p7 ₂	p6 ₂	p5 ₂	p4 ₂	p3 ₂	p2 ₂	p1 ₂	p0 ₂	q0 ₂	q1 ₂	q2 ₂	q3 ₂	q4 ₂	q5 ₂	q6 ₂	q7 ₂
p7 ₃	p6 ₃	p5 ₃	p4 ₃	p3 ₃	p2 ₃	p1 ₃	p0 ₃	q0 ₃	q1 ₃	q2 ₃	q3 ₃	q4 ₃	q5 ₃	q6 ₃	q7 ₃
p7 ₄	p6 ₄	p5 ₄	p4 ₄	p3 ₄	p2 ₄	p1 ₄	p0 ₄	q0 ₄	q1 ₄	q2 ₄	q3 ₄	q4 ₄	q5 ₄	q6 ₄	q7 ₄
p7 ₅	p6 ₅	p5 ₅	p4 ₅	p3 ₅	p2 ₅	p1 ₅	p0 ₅	q0 ₅	q1 ₅	q2 ₅	q3 ₅	q4 ₅	q5 ₅	q6 ₅	q7 ₅
p7 ₆	p6 ₆	p5 ₆	p4 ₆	p3 ₆	p2 ₆	p1 ₆	p0 ₆	q0 ₆	q1 ₆	q2 ₆	q3 ₆	q4 ₆	q5 ₆	q6 ₆	q7 ₆
p7 ₇	p6 ₇	p5 ₇	p4 ₇	p3 ₇	p2 ₇	p1 ₇	p0 ₇	q0 ₇	q1 ₇	q2 ₇	q3 ₇	q4 ₇	q5 ₇	q6 ₇	q7 ₇

도면7

