



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0110516
(43) 공개일자 2015년10월02일

- | | |
|--|---|
| (51) 국제특허분류(Int. Cl.)
<i>HO4N 19/80</i> (2014.01) <i>HO4N 19/423</i> (2014.01)
<i>HO4N 19/82</i> (2014.01) | (71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775 |
| (52) CPC특허분류
<i>HO4N 19/80</i> (2015.01)
<i>HO4N 19/423</i> (2015.01) | (72) 발명자
쿠마르 산지브
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775 |
| (21) 출원번호 10-2015-7018813 | 판 더 아우베라 게르트
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775 |
| (22) 출원일자(국제) 2013년12월17일
심사청구일자 없음 | (뒷면에 계속) |
| (85) 번역문제출일자 2015년07월13일 | (74) 대리인
특허법인코리아나 |
| (86) 국제출원번호 PCT/US2013/075824 | |
| (87) 국제공개번호 WO 2014/100015
국제공개일자 2014년06월26일 | |
| (30) 우선권주장
13/720,499 2012년12월19일 미국(US) | |

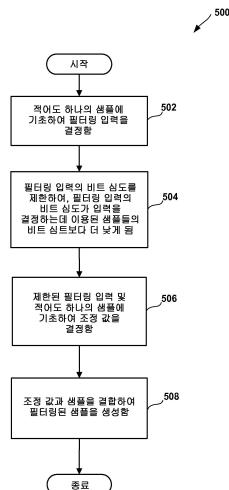
전체 청구항 수 : 총 48 항

(54) 발명의 명칭 라인 베퍼가 감소된 디블로킹 필터

(57) 요 약

특정 양태들에 따른 비디오 정보를 필터링하도록 구성되는 장치는 메모리 유닛, 및 메모리 유닛과 통신하는 프로세서를 포함한다. 메모리 유닛은 적어도 2 개의 인접하는 비디오 블록들을 포함하는 비디오 정보를 저장하고, 각각의 비디오 블록은 복수의 비디오 샘플들을 포함하고, 각각의 비디오 샘플은 비트 심도를 갖는다. 프로세서는 비디오 샘플 및 조정 값에 적어도 부분적으로 기초하여 필터링된 비디오 샘플들을 결정한다. 프로세서는 제한된 비트 심도를 갖는 입력으로부터 적어도 부분적으로 조정 값을 결정한다. 입력은 하나 이상의 비디오 샘플들의 세트로부터 결정되고, 그 비트 심도는 이 비트 심도가 하나 이상의 비디오 샘플들의 비트 심도보다 더 작도록 제한된다.

대 표 도 - 도5



(52) CPC특허분류
HO4N 19/82 (2015.01)

(72) 발명자

왕 카이

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

샤오 슈

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명세서

청구범위

청구항 1

비디오 정보를 필터링하도록 구성되는 장치로서,

비디오 정보를 저장하도록 구성되는 메모리 유닛으로서, 상기 비디오 정보는 적어도 2 개의 공간적으로 인접하는 비디오 블록들을 포함하고, 각각의 비디오 블록은 복수의 비디오 샘플들을 포함하고, 각각의 비디오 샘플은 비트 심도를 갖는, 상기 메모리 유닛; 및

상기 메모리 유닛과 통신하는 프로세서를 포함하며,

상기 프로세서는,

입력 값 및 조정 값을 적어도 부분적으로 기초하여 필터링된 비디오 샘플들을 결정하도록 구성되고,

상기 입력 값은 상기 비디오 샘플들 중 하나 이상의 비디오 샘플들의 제 1 세트로부터 결정되고 상기 제 1 세트의 총 비트 심도보다 적은 비트 심도를 갖고,

상기 조정 값을 상기 입력 값과, 상기 비디오 샘플들 중 하나 이상의 비디오 샘플들의 제 2 세트의 결합을 포함하는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 2

제 1 항에 있어서,

상기 제 1 세트에서의 상기 비디오 샘플들 및 상기 제 2 세트에서의 상기 비디오 샘플들은 상기 인접하는 비디오 블록들 중 2 개 사이에서 연장되는, 공간적으로 인접하는 비디오 샘플들의 연속 라인에 위치되는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 3

제 1 항에 있어서,

상기 입력 값은 상한 값 및 하한 값을 갖는 미리 정해진 범위 내에서 제약되고,

상기 프로세서는 또한, 상기 입력 값이 상기 하한 값 미만이면, 상기 입력 값을 상기 하한 값으로 설정하고, 상기 입력 값이 상기 상한 값을 초과하면, 상기 입력 값을 상기 상한 값으로 설정함으로써 상기 입력 값을 제약하도록 구성되는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 4

제 1 항에 있어서,

상기 조정 값을 상한 값 및 하한 값을 갖는 미리 정해진 범위 내에서 제약되고,

상기 프로세서는 또한, 상기 조정 값이 상기 하한 값 미만이면, 상기 조정 값을 상기 하한 값으로 설정하고, 상기 조정 값이 상기 상한 값을 초과하면, 상기 조정 값을 상기 상한 값으로 설정함으로써 상기 조정 값을 제약하도록 구성되는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 5

제 1 항에 있어서,

상기 프로세서는 또한, 상기 입력 값을 양자화함으로써 상기 입력 값의 상기 비트 심도를 감소시키도록 구성되는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 6

제 1 항에 있어서,

상기 프로세서는 또한, 라인 버퍼에 상기 입력 값을 저장하도록 구성되는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 7

제 1 항에 있어서,

상기 프로세서는 또한, 상기 필터링된 비디오 샘플을 이용하여 상기 비디오 정보를 인코딩하도록 구성되는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 8

제 1 항에 있어서,

상기 프로세서는 또한, 상기 필터링된 비디오 샘플을 이용하여 상기 비디오 정보를 디코딩하도록 구성되는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 9

제 1 항에 있어서,

상기 프로세서는 또한, 필터링 강도에 적어도 부분적으로 기초하여 상기 필터링된 비디오 샘플을 결정하도록 구성되는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 10

제 9 항에 있어서,

상기 필터링 강도는 상기 비디오 샘플들 중 하나 이상의 비디오 샘플의 제 3 세트에 기초하여 결정되는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 11

제 10 항에 있어서,

하나 이상의 비디오 샘플들의 상기 제 1 세트는 상기 제 2 세트에도 상기 제 3 세트에도 포함되지 않은 적어도 하나의 비디오 샘플을 포함하는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 12

제 10 항에 있어서,

하나 이상의 비디오 샘플들의 상기 제 2 세트에서의 적어도 하나의 비디오 샘플은 하나 이상의 비디오 샘플들의 상기 제 3 세트에 있는, 비디오 정보를 필터링하도록 구성되는 장치.

청구항 13

비디오 정보를 필터링하는 방법으로서,

비디오 정보를 저장하는 단계로서, 상기 비디오 정보는 적어도 2 개의 공간적으로 인접하는 비디오 블록들을 포함하고, 각각의 비디오 블록은 복수의 비디오 샘플들을 포함하고, 각각의 비디오 샘플은 비트 심도를 갖는, 상기 비디오 정보를 저장하는 단계; 및

입력 값 및 조정 값에 적어도 부분적으로 기초하여 필터링된 비디오 샘플들을 결정하는 단계를 포함하며,

상기 입력 값은 상기 비디오 샘플들 중 하나 이상의 비디오 샘플들의 제 1 세트로부터 결정되고 상기 제 1 세트의 총 비트 심도보다 적은 비트 심도를 갖고,

상기 조정 값은 상기 입력 값과, 상기 비디오 샘플들 중 하나 이상의 비디오 샘플들의 제 2 세트의 결합을 포함하는, 비디오 정보를 필터링하는 방법.

청구항 14

제 13 항에 있어서,

상기 제 1 세트에서의 상기 비디오 샘플들 및 상기 제 2 세트에서의 상기 비디오 샘플들은 상기 인접하는 비디오 블록들 중 2 개 사이에서 연장되는, 공간적으로 인접하는 비디오 샘플들의 연속 라인에 위치되는, 비디오 정보를 필터링하는 방법.

청구항 15

제 13 항에 있어서,

상기 입력 값은 상한 값 및 하한 값을 갖는 미리 정해진 범위 내에서 제약되고,

상기 입력 값이 상기 하한 값 미만이면, 상기 입력 값을 상기 하한 값으로 설정하고, 상기 입력 값이 상기 상한 값을 초과하면, 상기 입력 값을 상기 상한 값으로 설정함으로써 상기 입력 값이 제약되는, 비디오 정보를 필터링하는 방법.

청구항 16

제 13 항에 있어서,

상기 조정 값은 상한 값 및 하한 값을 갖는 미리 정해진 범위 내에서 제약되고,

상기 조정 값이 상기 하한 값 미만이면, 상기 조정 값을 상기 하한 값으로 설정하고, 상기 조정 값이 상기 상한 값을 초과하면, 상기 조정 값을 상기 상한 값으로 설정함으로써 상기 조정 값이 제약되는, 비디오 정보를 필터링하는 방법.

청구항 17

제 13 항에 있어서,

상기 입력 값의 상기 비트 십도는 상기 입력 값을 양자화함으로써 제한되는, 비디오 정보를 필터링하는 방법.

청구항 18

제 13 항에 있어서,

상기 입력 값을 라인 베퍼에 저장하는 단계를 더 포함하는, 비디오 정보를 필터링하는 방법.

청구항 19

제 13 항에 있어서,

상기 필터링된 비디오 샘플을 이용하여 상기 비디오 정보를 인코딩하는 단계를 더 포함하는, 비디오 정보를 필터링하는 방법.

청구항 20

제 13 항에 있어서,

상기 필터링된 비디오 샘플을 이용하여 상기 비디오 정보를 디코딩하는 단계를 더 포함하는, 비디오 정보를 필터링하는 방법.

청구항 21

제 13 항에 있어서,

상기 필터링된 비디오 샘플은 필터링 강도에 적어도 부분적으로 기초하여 결정되는, 비디오 정보를 필터링하는 방법.

청구항 22

제 21 항에 있어서,

상기 필터링 강도는 상기 비디오 샘플들 중 하나 이상의 비디오 샘플의 제 3 세트에 기초하여 결정되는, 비디오 정보를 필터링하는 방법.

청구항 23

제 22 항에 있어서,

하나 이상의 비디오 샘플들의 상기 제 1 세트는 상기 제 2 세트에도 상기 제 3 세트에도 포함되지 않은 적어도 하나의 비디오 샘플을 포함하는, 비디오 정보를 필터링하는 방법.

청구항 24

제 22 항에 있어서,

하나 이상의 비디오 샘플들의 상기 제 2 세트에서의 적어도 하나의 비디오 샘플은 하나 이상의 비디오 샘플들의 상기 제 3 세트에 있는, 비디오 정보를 필터링하는 방법.

청구항 25

명령들이 저장된 컴퓨터 판독가능 저장 매체로서,

상기 명령들은 실행될 때 장치로 하여금,

비디오 정보를 저장하게 하는 것으로서, 상기 비디오 정보는 적어도 2 개의 공간적으로 인접하는 비디오 블록들을 포함하고, 각각의 비디오 블록은 복수의 비디오 샘플들을 포함하고, 각각의 비디오 샘플은 비트 심도를 갖는, 상기 비디오 정보를 저장하게 하고; 그리고

입력 값 및 조정 값을 적어도 부분적으로 기초하여 필터링된 비디오 샘플들을 결정하게 하며,

상기 입력 값은 상기 비디오 샘플들 중 하나 이상의 비디오 샘플들의 제 1 세트로부터 결정되고 상기 제 1 세트의 총 비트 심도보다 적은 비트 심도를 갖고,

상기 조정 값은 상기 입력 값과, 상기 비디오 샘플들 중 하나 이상의 비디오 샘플들의 제 2 세트의 결합을 포함하는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 26

제 25 항에 있어서,

상기 제 1 세트에서의 상기 비디오 샘플들 및 상기 제 2 세트에서의 상기 비디오 샘플들은 상기 인접하는 비디오 블록들 중 2 개 사이에서 연장되는, 공간적으로 인접하는 비디오 샘플들의 연속 라인에 위치되는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 27

제 25 항에 있어서,

상기 입력 값은 상한 값 및 하한 값을 갖는 미리 정해진 범위 내에서 제약되고,

상기 입력 값이 상기 하한 값 미만이면, 상기 입력 값을 상기 하한 값으로 설정하고, 상기 입력 값이 상기 상한 값을 초과하면, 상기 입력 값을 상기 상한 값으로 설정함으로써 상기 입력 값을 제약하는 명령들을 더 포함하는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 28

제 25 항에 있어서,

상기 조정 값은 상한 값 및 하한 값을 갖는 미리 정해진 범위 내에서 제약되고,

상기 조정 값이 상기 하한 값 미만이면, 상기 조정 값을 상기 하한 값으로 설정하고, 상기 조정 값이 상기 상한 값을 초과하면, 상기 조정 값을 상기 상한 값으로 설정함으로써 상기 조정 값을 제약하는 명령들을 더

포함하는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 29

제 25 항에 있어서,

상기 입력 값을 양자화함으로써 상기 입력 값의 상기 비트 십도를 제한하는 명령들을 더 포함하는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 30

제 25 항에 있어서,

상기 입력 값을 라인 버퍼에 저장하는 명령들을 더 포함하는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 31

제 25 항에 있어서,

상기 필터링된 비디오 샘플을 이용하여 상기 비디오 정보를 인코딩하는 명령들을 더 포함하는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 32

제 25 항에 있어서,

상기 필터링된 비디오 샘플을 이용하여 상기 비디오 정보를 디코딩하는 명령들을 더 포함하는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 33

제 25 항에 있어서,

상기 필터링된 비디오 샘플은 필터링 강도에 적어도 부분적으로 기초하여 결정되는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 34

제 33 항에 있어서,

상기 필터링 강도는 상기 비디오 샘플들 중 하나 이상의 비디오 샘플의 제 3 세트에 기초하여 결정되는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 35

제 34 항에 있어서,

하나 이상의 비디오 샘플들의 상기 제 1 세트는 상기 제 2 세트에도 상기 제 3 세트에도 포함되지 않은 적어도 하나의 비디오 샘플을 포함하는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 36

제 34 항에 있어서,

하나 이상의 비디오 샘플들의 상기 제 2 세트에서의 적어도 하나의 비디오 샘플은 하나 이상의 비디오 샘플들의 상기 제 3 세트에 있는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 37

비디오 정보를 필터링하는 장치로서,

비디오 정보를 저장하는 수단으로서, 상기 비디오 정보는 적어도 2 개의 공간적으로 인접하는 비디오 블록들을 포함하고, 각각의 비디오 블록은 복수의 비디오 샘플들을 포함하고, 각각의 비디오 샘플은 비트 십도를 갖는, 상기 비디오 정보를 저장하는 수단; 및

입력 값 및 조정 값에 적어도 부분적으로 기초하여 필터링된 비디오 샘플들을 결정하는 수단을 포함하며,
상기 입력 값은 상기 비디오 샘플들 중 하나 이상의 비디오 샘플들의 제 1 세트로부터 결정되고 상기 제 1 세트
의 총 비트 심도보다 적은 비트 심도를 갖고,
상기 조정 값은 상기 입력 값과, 상기 비디오 샘플들 중 하나 이상의 비디오 샘플들의 제 2 세트의 결합을 포함
하는, 비디오 정보를 필터링하는 장치.

청구항 38

제 37 항에 있어서,

상기 제 1 세트에서의 상기 비디오 샘플들 및 상기 제 2 세트에서의 상기 비디오 샘플들은 상기 인접하는 비디
오 블록들 중 2 개 사이에서 연장되는, 공간적으로 인접하는 비디오 샘플들의 연속 라인에 위치되는, 비디오 정
보를 필터링하는 장치.

청구항 39

제 37 항에 있어서,

상기 입력 값은 상한 값 및 하한 값을 갖는 미리 정해진 범위 내에서 제약되고,

상기 필터링된 비디오 샘플들을 결정하는 수단은 또한, 상기 입력 값이 상기 하한 값 미만이면, 상기 입력 값을
상기 하한 값으로 설정하고, 상기 입력 값이 상기 상한 값을 초과하면, 상기 입력 값을 상기 상한 값으로 설정
함으로써 상기 입력 값을 제약하도록 구성되는, 비디오 정보를 필터링하는 장치.

청구항 40

제 37 항에 있어서,

상기 조정 값은 상한 값 및 하한 값을 갖는 미리 정해진 범위 내에서 제약되고,

상기 필터링된 비디오 샘플들을 결정하는 수단은 또한, 상기 조정 값이 상기 하한 값 미만이면, 상기 조정 값을
상기 하한 값으로 설정하고, 상기 조정 값이 상기 상한 값을 초과하면, 상기 조정 값을 상기 상한 값으로 설정
함으로써 상기 조정 값을 제약하도록 구성되는, 비디오 정보를 필터링하는 장치.

청구항 41

제 37 항에 있어서,

상기 필터링된 비디오 샘플들을 결정하는 수단은 또한, 상기 입력 값을 양자화함으로써 상기 입력 값의 상기 비
트 심도를 감소시키도록 구성되는, 비디오 정보를 필터링하는 장치.

청구항 42

제 37 항에 있어서,

상기 필터링된 비디오 샘플들을 결정하는 수단은 또한, 라인 버퍼에 상기 입력 값을 저장하거나 또는 상기 라인
버퍼로부터 상기 입력 값을 획득하도록 구성되는, 비디오 정보를 필터링하는 장치.

청구항 43

제 37 항에 있어서,

상기 장치는 상기 필터링된 비디오 샘플을 이용하여 상기 비디오 정보를 인코딩하는 수단을 더 포함하는, 비디
오 정보를 필터링하는 장치.

청구항 44

제 37 항에 있어서,

상기 장치는 상기 필터링된 비디오 샘플을 이용하여 상기 비디오 정보를 디코딩하는 수단을 더 포함하는, 비디
오 정보를 필터링하는 장치.

청구항 45

제 37 항에 있어서,

상기 필터링된 비디오 샘플들을 결정하는 수단은 또한, 필터링 강도에 적어도 부분적으로 기초하여 상기 필터링된 비디오 샘플을 결정하도록 구성되는, 비디오 정보를 필터링하는 장치.

청구항 46

제 45 항에 있어서,

상기 필터링 강도는 상기 비디오 샘플들 중 하나 이상의 비디오 샘플의 제 3 세트에 기초하여 결정되는, 비디오 정보를 필터링하는 장치.

청구항 47

제 46 항에 있어서,

하나 이상의 비디오 샘플들의 상기 제 1 세트는 상기 제 2 세트에도 상기 제 3 세트에도 포함되지 않은 적어도 하나의 비디오 샘플을 포함하는, 비디오 정보를 필터링하는 장치.

청구항 48

제 46 항에 있어서,

하나 이상의 비디오 샘플들의 상기 제 2 세트에서의 적어도 하나의 비디오 샘플은 하나 이상의 비디오 샘플들의 상기 제 3 세트에 있는, 비디오 정보를 필터링하는 장치.

발명의 설명**기술 분야**

[0001] 이 개시물은 비디오 코딩에 관한 것이다.

배경 기술

[0002] 디지털 비디오 성능들은, 디지털 텔레비전, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, PDA들 (personal digital assistants), 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, 이북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 디바이스들, 비디오 게임 콘솔, 셀룰러 또는 위성 라디오 텔레폰들, 소위 "스마트폰들", 화상 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, AVC (Advanced Video Coding), 현재 개발 하에 있는 HEVC (High Efficiency Video Coding) 에 의해 정의된 표준들, 및 이러한 표준들의 확장들에서 설명된 바와 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 보다 효율적으로 송신하고, 수신하고, 인코딩하고, 디코딩하고, 및/또는 저장할 수도 있다.

[0003] 비디오 코딩 기법들은 비디오 시퀀스들에 내재하는 중복을 감소시키거나 제거하기 위한 공간적 (인트라-픽쳐) 예측 및/또는 시간적 (인터-픽쳐) 예측을 포함한다. 블록 기반 비디오 코딩에서, 비디오 슬라이스 (예를 들어, 비디오 프레임 또는 비디오 프레임의 일부분) 는 비디오 블록들로 파티셔닝될 수도 있고, 이 비디오 블록들은 복수의 비디오 샘플들 (또한 픽셀들로 알려짐) 을 포함한다. 비디오 블록들은 또한 트리블록들, 코딩 유닛들 (CUs), 및/또는 코딩 노드들로 지칭될 수도 있다. 픽쳐의 인트라-코딩된 (I) 슬라이스 내의 비디오 블록들은 동일한 픽쳐 내의 이웃하는 블록들 내의 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 픽쳐의 인터-코딩된 (P 또는 B) 슬라이스 내의 비디오 블록들은 동일한 픽쳐 내의 이웃하는 블록들 내의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 픽쳐들 내의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 픽쳐들은 프레임들이라고 지칭될 수도 있고, 참조 픽쳐들은 참조 프레임들로 지칭될 수도 있다.

[0004] 공간적 또는 시간적 예측은 코딩될 블록에 대해 예측 블록을 초래한다. 잔차 데이터는 코딩되는 원래 블록과 예측 블록 사이의 샘플 차이들을 나타낸다. 인터 코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가르키는 모션 벡터, 및 코딩된 블록과 예측 블록 사이의 차이를 나타내는 잔차 데이터에 따라

인코딩된다. 인트라 코딩된 블록은 인트라 코딩 모드와 잔차 데이터에 따라 인코딩된다. 더 많은 압축을 위해, 잔차 데이터는 샘플 도메인으로부터 변환 도메인으로 변환될 수도 있어, 잔차 변환 계수들을 초래할 수도 있고, 그 후 이들은 양자화될 수도 있다. 초기에 2차원 어레이로 배열된 양자화된 변환 계수들은 변환 계수들의 1차원 벡터를 생성하기 위해 스캔될 수도 있고, 더욱 더 많은 압축을 달성하기 위해 엔트로피 코딩이 적용될 수도 있다.

발명의 내용

과제의 해결 수단

[0005] 일반적으로, 이 개시물은 블록 기반 비디오 데이터의 필터링에 관련된 기술을 설명한다. 인코딩 동작들이 비디오 블록들 상에서 수행되고 결과 데이터가 압축될 때, 블록들 사이의 에지들 (또는 바운더리들)에서의 샘플들 사이에서 급격한 트랜지션들이 발생할 수도 있다. 이를 급격한 변화들은 종종 블로킹 아티팩트들로서 지칭된다. 일부 예들에서, 디블로킹 필터는 이러한 아티팩트들을 감소시키거나 또는 이를 비쥬얼 효과들을 완화시키는데 이용될 수도 있다. 디블로킹 필터는 하나 이상의 입력들에 의존할 수도 있고, 이를 입력들의 비트 심도는 이 개시물의 기법들에 따라 감소될 수도 있다.

[0006] 디블로킹 필터들이 하드웨어로 구현될 때, 라인 버퍼들이, 필터링 동안에 이용되는 입력들의 값을 저장하는데 이용된다. 이러한 입력들은 블록 에지들 근처에 위치되는 비디오 샘플들을 포함할 수도 있다. 본 개시물의 특정 실시형태들은 라인 버퍼들에 의해 이용된 메모리의 양을 감소시키는데 이용될 수도 있다. 예를 들어, 블록 에지의 각각의 사이드 상에서의 4 개의 비디오 샘플들을 저장하는 대신에, 디블로킹 필터는, 각각의 사이드로부터의 3 개의 샘플들, 및 샘플의 것보다 낮은 비트 심도를 갖는 하나의 입력 또는 계산 값을 저장하도록 구성될 수도 있다. 이러한 입력의 더 낮은 비트 심도는 이것이 샘플보다는 라인 버퍼들에서 보다 적은 공간을 점유하는 것을 허용한다. 그 결과, 라인 버퍼들이 더 소형으로, 더 비용 효과적으로, 그리고 효율적으로 될 수도 있다.

[0007] 이 개시물의 특정 양태들에 따른 비디오 정보를 필터링하도록 구성되는 장치는 메모리 유닛, 및 메모리 유닛과 통신하는 프로세서를 포함한다. 메모리 유닛은 적어도 2 개의 인접하는 비디오 블록들을 포함하는 비디오 정보를 저장한다. 각각의 비디오 블록은 복수의 비디오 샘플들을 포함하고, 각각의 비디오 샘플은 비트 심도를 갖는다. 프로세서는 비디오 샘플 및 조정 값에 적어도 부분적으로 기초하여 필터링된 비디오 샘플들을 결정한다. 프로세서는 제한된 비트 심도를 갖는 입력으로부터 적어도 부분적으로 조정 값을 결정한다. 입력은 하나 이상의 비디오 샘플들의 세트로부터 결정되고, 그 비트 심도는 이 비트 심도가 하나 이상의 비디오 샘플들의 비트 심도보다 더 작도록 제한된다.

[0008] 하나 이상의 예들의 세부사항들이 첨부 도면들 및 하기의 설명에서 제시된다. 다른 특징들, 목적들 및 이점들은 하기의 설명 및 도면들, 및 하기의 특허청구범위로부터 명확해질 것이다.

도면의 간단한 설명

[0009] 도 1 은 이 개시물의 양태들에 따른 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 나타내는 블록도이다.

도 2 는 이 개시물의 양태들에 따른 기법들을 구현할 수도 있는 비디오 인코더의 실시예를 나타내는 블록도이다.

도 3 은 이 개시물의 양태들에 따른 기법들을 구현할 수도 있는 비디오 디코더의 일 예를 나타내는 블록도이다.

도 4a 는 이 개시물의 양태들에 따라 필터링될 수도 있는 다수의 비디오 블록들을 포함하는 비디오 슬라이스를 나타내는 개념도이다.

도 4b 는 도 4a 에서의 특정 비디오 블록들을 보다 자세하게 나타내는 개념도이다.

도 5 는 이 개시물의 양태들에 따라, 비트 심도가 감소된 입력들을 갖는 디블로킹 필터에 대한 예시적인 프로세스를 나타내는 플로우차트이다.

도 6 은 HEVC Working Draft 6 에 기초하여 디블로킹 필터에 대하여 자세히 다루어진, 도 5 로부터의 예시적인 프로세스의 버전을 나타내는 플로우차트이다.

도 7a 는 디블로킹 필터에 의해 이용된 입력들이 라인 버터들에 저장될 수도 있는 방법의 예를 나타내는 개념도이다.

도 7b 는 이 개시물의 양태들에 따라, 디블로킹된 필터에 의해 이용된, 비트 심도가 감소된 입력들이, 사이즈가 감소된 라인 버퍼들에 저장될 수도 있는 방법의 예를 나타내는 개념도이다.

발명을 실시하기 위한 구체적인 내용

[0010] 이 개시물의 실시형태들은 인코딩 및 디코딩 양쪽 모두를 포함할 수도 있는 블록 기반 비디오 코딩에 관련된다. 예를 들어, 실시형태들은 HEVC (High Efficiency Video Coding) 구현과 함께 또는 구현 내에 관련되거나 또는 이용될 수도 있다. 이전에 언급된 바와 같이, 블록 기반 비디오 인코더는 비디오 슬라이스들 (예를 들어, 비디오 프레임들) 을 블록들로 분할하며, 이 블록들 각각은 복수의 샘플들을 포함한다. 각각의 샘플의 값은 비디오 특정 부분의 하나 이상의 양태들을 나타낸다. 예를 들어, 비디오 블록의 하단 우측 코너에 서의 샘플의 값은 비디오 꾹쳐의 대응하는 부분에 대한 루미넌스 ("루마") 또는 크로미넌스 ("크로마") 레벨을 나타낼 수도 있다.

[0011] 단지 예시만의 목적을 위하여, 이 개시물의 몇몇 기법들이 예들로 설명된다. 예들의 일부는 4 개의 로우들 및 4 개의 컬럼들로 정렬된 16 개의 샘플들을 포함하는 블록들을 포함한다. 기법들은 다양한 개수들의 로우들 및 컬럼들에서 보다 적은 또는 보다 많은 샘플들을 포함하는 블록들을 가진 예들로 확장될 수 있음을 이해하여야 한다.

[0012] 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 ITU-T H.264 를 포함한다. 추가로, 새로운 비디오 코딩 표준, 즉 고효율 비디오 코딩 (High-Efficiency Video Coding; HEVC) 이 ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group; VCEG) 및 ISO/IEC 동화상 전문가 그룹 (MPEG) 의 JCT-VC (Joint Collaboration Team on Video Coding) 에 의해 개발되고 있다. "HEVC Working Draft 6" 로서 지칭되는 HEVC 의 드래프트은 2012년 10월 24일자의 http://phenix.int-evry.fr/jct/doc_end_user/documents/8_San%20Jose/wg11/JCTVC-H1003-v6.zip 으로부터 다운로드 가능하다. HEVC Working Draft 6 에 대한 전체 인용은 JCTVC-H1003 [Bross 등의, "High Efficiency Video Coding (HEVC) Text Specification Draft 6"; ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 7차 미팅: 스위스 제네바, 2011년 11월 21일 내지 2011년 11월 30일] 이다. 이들 참고문헌들 각각은 여기서는 그 전체 내용을 참조로서 포함한다.

[0013] 이전에 언급된 바와 같이, 블록 기반 비디오 인코더는 예측 (예를 들어, 인터 프레임 또는 인트라 프레임) 에 의해 비디오 데이터를 압축할 수도 있다. 인코더는 또한, 특정 블록에서의 예측 샘플과 실제 샘플 사이의 차이들을 나타내는 잔차 데이터를 생성할 수도 있다. 잔차 데이터는 샘플 도메인으로부터 변환 도메인으로 변환될 수도 있고, 결과적인 변환 계수들이 양자화될 수도 있다. 양자화된 변환 계수들은 인코딩되었던 비디오 데이터의 근사값을 나타낼 수도 있고, 근사값은 어느 정도의 에러를 나타낼 수도 있다. 양자화된 변환 계수들이 각각의 블록에 대하여 별개로 계산되기 때문에, 에러의 분포는 블록들 간의 바운더리들에서 급격하게 변화할 수도 있다.

[0014] 디코딩이 수행될 때, 양자화 및 변환 프로세스들은 역으로 되어, 각각의 블록에 대한 샘플들의 세트를 생성한다. 이들 샘플들은 위에 설명된 양자화 프로세스에 의해 도입된 에러 또는 아티팩트 때문에, 인코딩 되었던 오리지널 샘플들과 정확하게 동일하지 않을 수도 있다. 그 결과, 인코딩된 데이터로부터 렌더링되는 꾹쳐들은 블로킹 아티팩트들을 포함할 수도 있고, 이를 테면, 꾹쳐가 디스플레이될 때 시각적으로 왜곡 불연속 성들로서 나타날 수도 있는 블록 에지들을 가로지르는 샘플 값들에서의 급격한 트랜지션들이 존재한다.

[0015] 이 개시물에서 설명된 기법들은 이러한 블로킹 아티팩트들의 효과들을 감소시키는데 이용되는 디블로킹 필터들에 관련된 이슈들을 해결할 수도 있다. 디블로킹 필터는 인접하는 블록들에서의 샘플들을 조정하여 블록 에지를 가로지르는 트랜지션들을 평활화시킬 수도 있다. 디블로킹 필터에 의해 수행되는 조정들은 블록들 내의 샘플들을 입력들로서 이용하는 결정들 및 계산들을 수반할 수도 있다. 이 개시물의 기법들은 하나 이상의 대체 입력들을 선택하는데 이용될 수도 있고, 이는 하나 이상의 샘플들 신에 결정들 및 계산들에 이용될 수 있다. 대체 입력이 샘플들로부터 자체적으로 유도될 수도 있지만, 그 비트 심도는 이것이 대체한 샘플들의 비트 심도보다 더 낮아지도록 제한될 수도 있다.

[0016] 개시된 기법들은 요구되는 라인 버퍼 저장의 양을 감소시키도록 디블로킹 필터들의 하드웨어 구현들에 이용될 수도 있다. 라인 버퍼들은 메모리로부터의 입력들을 재페치하는 것 (re-fetching) 을 회피하기 위하여 필터

링 프로세스에 대해 요구되는 입력들을 저장하는데 이용된다. 디블로킹 필터를 구현하는데 요구되는 라인 버퍼들의 수는 필터링 프로세스 내의 결정들 및 계산들에 이용되는 입력들의 수에 의해 결정된다. 라인 버퍼들의 비트 심도는 거기에 저장된 입력들의 비트 심도에 의해 결정된다. 위에 나타낸 바와 같이, 이 개시물의 기법들은 적어도 하나의 입력들의 비트 심도를 감소시키는데 이용될 수도 있다. 개시된 기법들은 또한 필터에 의해 요구되는 입력들의 수를 감소시키는데 이용될 수도 있다. 기법들이 고해상도 비디오 슬라이스들에 이용될 때, 디블로킹에 요구되는 총 라인 버퍼 저장이 충분히 감소될 수 있다.

[0017]

신규의 시스템들, 장치들, 및 방법들의 여러 양태들이 첨부된 도면들을 참조로 아래에 보다 완전하게 설명된다. 그러나, 이 개시물은 많은 상이한 형태들로 구현될 수도 있고, 이 개시물 전반에서 제시된 임의의 특정 구조 또는 기능으로 제한되는 것으로 간주되어서는 안된다. 오히려, 이러한 양태들은 이 개시물이 철저하고 완벽하게 되도록 제공되며, 당업자들에게 이 개시물의 범위를 충분히 전달할 것이다. 여기에서의 교시들에 기초하여, 당해 기술 분야의 당업자는 본 개시물의 범위가, 본 발명의 임의의 다른 양태와 독립적으로 구현되든 또는 결합하여 구현되든 간에, 여기에 개시된 신규의 시스템들, 장치들 및 방법들의 임의의 양태들을 커버하도록 의도됨을 이해하여야 한다. 예를 들어, 여기에 제시된 임의의 수의 양태들을 이용하여 장치가 구현될 수도 있거나 또는 방법이 실시될 수도 있다. 또한, 본 발명의 범위는 여기에 제시된 본 발명의 다양한 양태들에 더하여 또는 그 외에 다른 구조, 기능성, 또는 구조와 기능성을 이용하여 실시되는 그러한 장치 또는 방법을 커버하도록 의도된다. 여기에 개시된 임의의 양태는 청구항의 하나 이상의 요소들에 의해 구체화될 수도 있음이 이해되어야 한다.

[0018]

비록 특정 양태들이 여기에 설명되지만, 이러한 양태들의 많은 변형예들 및 치환예들이 본 개시물의 범위 내에 속한다. 비록 바람직한 양태들의 일부 이득들 및 이점들이 언급되었지만, 본 개시물의 범위는 특정 이득들, 이용들, 또는 목적들로 제한되도록 의도되지 않는다. 오히려, 본 개시물들의 양태들은 상이한 무선 기술들, 시스템 구성들, 네트워크들, 및 송신 프로토콜들에 이용될 수도 있는 상이한 비디오 필터들 및 비디오 코더들에 널리 적용가능하게 의도되며, 이를 양태들 중 일부는 도면들에서 그리고 특정 양태들의 다음 설명에서 예로서 예시된다. 상세한 설명 및 도면들은 제한하는 것이기 보다는 단지 본 개시물의 예시일 뿐이며, 본 개시물의 범위는 첨부된 청구항들 및 그의 등가물들에 의해 정의된다.

[0019]

도 1은 이 개시물의 양태들에 따른 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 나타내는 블록도이다. 도 1에 도시된 바와 같이, 시스템 (10)은 목적지 디바이스 (14)에 의해 나중의 시간에 디코딩될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12)를 포함한다. 특히, 소스 디바이스 (12)는 비디오 데이터를 판독가능 매체 (16)를 통하여 목적지 디바이스 (14)에 제공한다. 소스 디바이스 (12) 및 목적지 디바이스 (14)는, 데스크탑 컴퓨터들, 노트북 (예를 들어, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 세탁박스들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게임용 콘솔들, 비디오 스트리밍 디바이스 등을 포함하는 다양한 범위의 디바이스들 중 임의의 것을 포함할 수도 있다. 몇몇 경우들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14)에는 무선 통신이 탑재될 수도 있다.

[0020]

목적지 디바이스 (14)는 컴퓨터 판독가능 매체 (16)를 통하여 디코딩된 인코딩된 비디오 데이터를 수신할 수도 있다. 컴퓨터 판독가능 매체 (16)는 인코딩된 비디오 데이터를 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 일 실시예에서, 컴퓨터 판독가능 매체 (16)는 소스 디바이스 (12)로 하여금 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 실시간으로 직접적으로 송신하게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조될 수도 있고, 목적지 디바이스 (14)로 전송될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 예컨대 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 패킷 기반의 네트워크, 예컨대 근거리 통신망 (local area network), 광역 통신망 (wide-area network), 또는 인터넷과 같은 글로벌 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12)로부터 목적지 디바이스 (14)로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0021]

일부 예들에서, 인코딩된 데이터는 출력 인터페이스 (22)로부터 저장 디바이스로 출력될 수도 있다. 이와 유사하게, 인코딩된 데이터는 입력 인터페이스 (28)에 의해 저장 디바이스로부터 액세스될 수도 있다. 저장 디바이스는 하드 드라이브들, 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 불휘발성 메모리와 같은 임의의 다양한 분산된 또는 로컬하게 액세스된 데이터 저장 매체, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적절한 디지털 저장 매체를 포함할 수도 있다. 다른 예에서, 저장 디바이스

는 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 다른 중간 저장 디바이스 또는 파일 서버에 대응할 수도 있다. 목적지 디바이스 (14)는 스트리밍 또는 다운로드를 통해 저장 디바이스로부터의 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장할 수 있고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신할 수 있는 임의의 타입의 서버일 수도 있다. 예시적인 파일 서버들은 (예를 들어 웹사이트 용의) 웹서버, FTP 서버, NAS (network attached storage) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14)는 인터넷을 접속을 포함하는 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버에 저장된 인코딩된 비디오 데이터를 액세스하는데 적합한 무선 채널(예를 들어, 와이파이 접속), 유선 접속(예를 들어, DSL, 케이블 모뎀 등), 또는 이들의 조합을 포함할 수도 있다. 저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이들의 조합일 수도 있다.

[0022] 이 개시물의 기법들은 무선 애플리케이션들 또는 설정들에 반드시 제한되는 것은 아니다. 본 기법은 임의의 다양한 멀티미디어 애플리케이션들, 예컨대 지상파 (over-the-air) 텔레비전 방송들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 이를 테면, DASH (dynamic adaptive streaming over HTTP), 데이터 저장 매체 상에 인코딩된 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩 또는 다른 애플리케이션들을 지원하여 비디오 코딩에 적용될 수도 있다. 몇몇 예들에서, 시스템 (10)은, 비디오 스트리밍, 비디오 플레이백, 비디오 방송, 및/또는 영상 전화와 같은 애플리케이션들을 지원하기 위해 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0023] 도 1의 실시예에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22)를 포함한다. 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 이 개시물에 따르면, 소스 디바이스 (12)의 비디오 인코더 (20)는 다수의 표준들 또는 표준 확장안에 따라 비디오 데이터를 포함하는 비트스트림을 코딩하기 위한 기법들을 적용하도록 구성될 수도 있다. 다른 예들에서, 소스 디바이스와 목적지 디바이스는 다른 컴포넌트들 또는 장치들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12)는 외부 비디오 소스 (18), 이를 테면 외부 카메라로부터 비디오 데이터를 수신할 수도 있다. 이와 달리, 목적지 디바이스 (14)는, 통합된 디스플레이 디바이스를 포함하기보다는, 외부 디스플레이 디바이스와 인터페이스할 수도 있다.

[0024] 도 1의 예시된 시스템 (10)은 단지 일 예에 불과하다. 이 개시물의 기법들은 임의의 디지털 비디오 인코더 및/또는 디코더에 의해 수행될 수도 있다. 또한, 이 개시물의 기법들들은 또한 비디오 인코딩 또는 디코딩 이전에, 또는 인코딩 또는 디코딩에 후속하여 예를 들어, 디코딩 후에 적용되는 필터에 의해 수행될 수도 있다. 일부 예들에서, 디바이스들 (12, 14)은, 디바이스들 (12, 14) 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하는 것과 실질적으로 대칭적인 방식으로 동작할 수도 있다. 따라서, 시스템 (10)은, 예를 들어, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 화상 전화를 위해, 비디오 디바이스들 (12, 14) 사이의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0025] 소스 디바이스 (12)의 비디오 소스 (18)는 비디오 캡쳐 디바이스, 이를 테면, 비디오 카메라, 이전에 캡쳐된 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 컨텐츠 제공자로부터 비디오를 수신하기 위한 비디오 퍼드 인터페이스를 포함할 수도 있다. 다른 대안으로, 비디오 소스 (18)는 소스 비디오로서 컴퓨터 그래픽 기반 데이터, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성 비디오의 조합을 생성할 수도 있다. 일부 경우들에서, 비디오 소스 (18)가 비디오 카메라이면, 소스 디바이스 (12)와 목적지 디바이스 (14)는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 위에서 언급된 바와 같이, 본 개시에서 설명된 기술들은 일반적으로 비디오 코딩에 적용될 수 있으며, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에서, 캡쳐된, 프리캡쳐된, 또는 컴퓨터로 생성된 비디오는 비디오 인코더 (20)에 의해 인코딩될 수도 있다. 그 후, 인코딩된 비디오 정보는 컴퓨터 판독가능 매체 (16) 상에 출력 인터페이스 (22)에 의해 출력될 수도 있다.

[0026] 컴퓨터 판독가능 매체 (16)는 일시적 매체, 이를 테면, 무선 브로드캐스트 또는 유선 네트워크 송신, 또는 저장 매체 (즉, 비일시적 저장 매체), 이를 테면, 하드 디스크, 플래시 드라이브, 콤팩트 디스크, 디지털 비디오 디스크, 또는 다른 컴퓨터 판독가능 매체를 포함할 수도 있다. 일부 예들에서, 네트워크 서버 (도시 생략)는 예를 들어, 네트워크 송신, 직접 유선 통신 등을 통하여, 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신하고 인코딩된 비디오 데이터를 목적지 디바이스 (14)에 제공할 수도 있다. 이와 유사하게, 매체 제조 설비, 이를 테면, 디스크 스탬핑 설비의 컴퓨팅 디바이스가 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신하고 인코딩된 비디오 데이터를 포함하는 디스크를 제조할 수도 있다. 따라서, 컴퓨터 판독가

능 매체 (16) 는 여러 예들에서 여러 형태들의 하나 이상의 컴퓨터 판독가능 매체를 포함하도록 이해되어질 수도 있다.

[0027] 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 컴퓨터 판독가능 매체 (16) 로부터 정보를 수신한다. 컴퓨터 판독가능 매체 (16) 의 정보는 비디오 인코더 (20) 에 의해 정의된 신택스 정보를 포함할 수도 있으며, 이 신택스 정보는 블록들 및 다른 코딩된 유닛들, 예를 들어, GOP들의 특성들 및/또는 프로세싱을 기술하는 신택스 엘리먼트들을 포함하고 비디오 디코더 (30) 에 의해 또한 이용된다. 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 임의의 여러 디스플레이 디바이스들, 이를 테면, 음극선관 (cathode ray tube; CRT), 액정 디스플레이 (liquid crystal display; LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (organic light emitting diode; OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스를 포함할 수도 있다.

[0028] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 코딩 표준, 이를 테면, 현재 개발 중에 있는 고효율 비디오 코딩 (HEVC) 표준에 따라 동작할 수도 있고, HEVC 테스트 모델(HEVC Test Model; HM) 을 따를 수도 있다. 대안으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 다르게는 MPEG-4, 파트 10, 고급 비디오 코딩 (AVC) 으로도 지칭되는 ITU-T H.264 표준과 같은 다른 사설 표준 혹은 산업 표준, 또는 이러한 표준들의 확장안들에 따라 동작할 수도 있다. 이 개시물의 기법들은 위에 나열된 표준들 중 어느 것을 제한없이 포함하는, 임의의 특정 코딩 표준으로 제한되지 않는다. 비디오 코딩 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263을 포함한다. 도 1 에 도시되지 않았지만, 몇몇 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 오디오 인코더 및 디코더와 통합될 수도 있고, 공통 데이터 스트림 또는 개별적인 데이터 스트림들에서 오디오 및 비디오 양쪽 모두의 인코딩을 핸들링하기 위해 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능하다면, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (user datagram protocol; UDP) 과 같은 다른 프로토콜에 따를 수도 있다.

[0029] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 임의의 다양한 적절한 인코더 회로부, 이를 테면, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들(DSPs), 주문형 반도체들 (application specific integrated circuits; ASICs), 필드 프로그램가능 게이트 어레이들 (FPGAs), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들로서 구현될 수도 있다. 상기 기법들이 부분적으로 소프트웨어로 구현되는 경우, 디바이스는 그 소프트웨어에 대한 명령들을 적절한 비일시적 컴퓨터 판독가능 매체에 저장할 수도 있고, 본 개시의 기술들을 수행하기 위해 하나 이상의 프로세서들을 사용하는 하드웨어에서 그 명령들을 실행할 수도 있다.

비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있고, 이를 중 어느 것도 결합된 인코더/디코더 (CODEC) 의 일부로서 각각의 디바이스에 통합될 수도 있다.

비디오 인코더 (20) 및/또는 비디오 디코더 (30) 를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 이를 테면 셀룰러 전화기를 포함할 수도 있다.

[0030] JCT-VC는 HEVC 표준의 개발에 노력하고 있다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HEVC Test Model; HM) 로 지칭되는 비디오 코딩 디바이스의 진보한 모델에 기초한다. HM 은, 예를 들어, ITU-T H.264/AVC 에 따른 기존 디바이스들에 의해 비디오 코딩 디바이스들의 여러가지 추가적인 능력들을 가정한다. 예를 들어, H.264 가 9개의 인트라 예측 인코딩 모드들을 제공하지만, HM은 33개만큼 많은 인트라 예측 인코딩 모드들을 제공할 수도 있다.

[0031] 일반적으로, HM 의 작업 모델은, 비디오 프레임 또는 픽쳐가 루마 (luma) 및 크로마 (chroma) 샘플들을 포함하는 최대 코딩 유닛들 (largest coding units; LCU들) 또는 트리블록들의 시퀀스로 분할될 수도 있다. 비트 스트림 내의 신택스 데이터는 샘플들의 수의 관점에서 최대 코딩 유닛인 LCU 에 대한 사이즈를 정의할 수도 있다. 슬라이스는 코딩 순서에서 다수의 연속적인 트리블록들을 포함한다. 비디오 프레임 또는 픽쳐는 하나 이상의 슬라이스들로 파티션될 수도 있다. 각각의 트리블록은 큐드트리에 따라 코딩 유닛들 (coding units; CU들) 로 스플릿될 수도 있다. 일반적으로, 큐드트리 데이터 구조는 CU당 하나의 노드를 포함하는데, 여기서 루트 노드는 트리블록에 대응한다. CU가 4개의 서브-CU들로 스플릿되면, 그 CU에 대응하는 노드는 4개의 리프 노드들을 포함하고, 그 각각은 서브-CU들 중 하나에 대응한다.

[0032] 큐드트리 데이터 구조의 각각의 노드는 대응하는 CU에 대해 신택스 데이터를 제공할 수도 있다. 예를 들어, 큐드트리 내의 노드는, 그 노드에 대응하는 CU 가 서브-CU들로 스플릿되는지의 여부를 나타내는 스플릿 플래그를 포함할 수도 있다. CU에 대한 신택스 엘리먼트들은 재귀적으로 정의될 수도 있고, CU가 서브-CU들로 스플릿되는지의 여부에 의존할 수도 있다. CU가 더이상 스플릿되지 않으면, 그것은 리프-CU로서 지칭된다.

이 개시물에서, 리프-CU 의 4개의 서브-CU들은 또한 오리지널 리프-CU 의 명백한 스플릿이 없는 경우에도 리프-CU들로서 또한 지칭될 것이다. 예를 들어, 16x16 사이즈에서 CU가 더 이상 스플릿되지 않으면, 4개의 8x8 서브-CU들은, 16x16 CU가 스플릿되지 않았지만, 리프-CU들로서 또한 지칭될 것이다.

[0033]

CU 가 사이즈 구별을 갖지 않는다는 점을 제외하면, CU는 H.264 표준의 매크로 블록과 유사한 목적을 갖는다. 예를 들어, 트리 블록은 4 개의 자식 노드들 (또한, 서브-CU들로서 지칭됨) 로 스플릿될 수도 있으며, 각각의 차일드 노드는 이어서 부모 노드로 되어, 다른 4 개의 차일드 노드들로 스플릿될 수도 있다. 큐드트리의 리프 노드로서 지칭되는 최종적인, 스플릿되지 않은 차일드 노드는 리프-CU 로 또한 지칭되는 코딩 노드를 포함한다. 코딩된 비트스트림과 관련된 신택스 데이터는 트리블록이 스플릿될 수도 있는 최대 횟수 (최대 CU 심도로서 지칭됨) 를 정의할 수도 있고, 코딩 노드들의 최소 사이즈를 또한 정의할 수도 있다. 따라서, 비트스트림은 최소 코딩 유닛 (smallest coding unit; SCU) 을 또한 정의할 수도 있다. 이 개시물은 HEVC 의 환경에서 CU, PU, 또는 TU 중 어느 것으로 지칭되는 용어 "블록", 또는 다른 표준들에서의 환경에서 유사한 데이터 구조들 (예를 들어, H.264/AVC 에서 매크로블록들 및 이들의 서브블록들) 을 이용한다.

[0034]

CU 는 코딩 노드 및 코딩 노드와 연관된 변환 유닛들 (transform units; TU들) 및 예측 유닛들 (prediction units; PU들) 을 포함한다. CU 의 사이즈는 코딩 노드의 사이즈에 대응하고 정사각형 형상임에 틀림없다.

CU의 사이즈는 8x8 샘플들에서부터 최대 64x64 샘플들 이상을 갖는 트리블록의 사이즈까지의 범위에 있을 수 있다. 각각의 CU는 하나 이상의 PU들 및 하나 이상의 TU들을 포함할 수도 있다. CU와 관련된 신택스 데이터는, 예를 들어, CU를 하나 이상의 PU들로 파티션하는 것을 기술할 수도 있다. 파티션 모드들은, CU 가 스kip 또는 다이렉트 모드 인코딩되는지, 인트라 예측 모드 인코딩되는지, 또는 인터 예측 모드 인코딩되는지의 사이에서 상이할 수도 있다. PU들은 비정사각형의 형상으로 파티션될 수도 있다. CU와 연관된 신택스 데이터는, 예를 들어, CU를 큐드트리에 따라 하나 이상의 TU들로 파티션하는 것을 또한 기술할 수도 있다.

TU 는 정사각형 형상 또는 비정사각형 형상 (예를 들어, 직사각형) 일 수 있다.

[0035]

HEVC 표준은 TU들에 따른 변환들을 허용하는데, 이것은 상이한 CU들에 대해 상이할 수도 있다. TU들은 파티션된 LCU에 대해 정의된 주어진 CU 내에서의 PU들의 사이즈에 기초하여 통상 사이징되지만, 이것이 항상 그런 것은 아닐 수도 있다. TU들은 통상적으로, PU들과 동일한 사이즈이거나 또는 더 작다. 일부 예들에서, CU 에 대응하는 잔차 샘플들은, "잔차 큐드 트리 (residual quad tree; RQT)" 로서 알려진 큐드트리 구조를 사용하여 더 작은 유닛들로 미세분할될 수도 있다. RQT 의 리프 노드들은 변환 유닛들 (TUs) 로 지칭될 수도 있다. TU들과 관련된 샘플 차이값들은 변환되어 변환 계수들을 생성할 수도 있고, 변환 계수는 양자화될 수도 있다.

[0036]

리프-CU는 하나 이상의 예측 유닛들 (PUs) 을 포함할 수도 있다. 일반적으로, PU는 대응하는 CU 의 전체 또는 일부에 대응하는 공간 영역을 나타내며, PU에 대한 참조 샘플을 취출하는 데이터를 포함할 수도 있다. 또한, PU 는 예측에 관련된 데이터를 포함한다. 예를 들어, PU 가 인트라 모드 인코딩될 때, PU 에 대한 데이터는 잔차 큐드트리 (RQT) 내에 포함될 수도 있고, 이 잔차 큐드트리는 PU 에 대응하는 TU 에 대한 인트라 예측 모드를 기술하는 데이터를 포함할 수도 있다. 다른 예로서, PU 가 인터 모드 인코딩될 때, PU 는 PU 에 대한 하나 이상의 모션 벡터들을 정의하는 데이터를 포함할 수도 있다. PU 에 대한 모션 벡터를 정의하는 데이터는, 예를 들어, 모션 벡터의 수평 성분, 모션 벡터의 수직 성분, 모션 벡터에 대한 해상도 (예를 들어, 1/4 샘플 정밀도 또는 1/8 샘플 정밀도), 모션 벡터가 가리키는 참조 픽쳐, 및/또는 모션 벡터에 대한 참조 픽쳐 리스트 (예를 들어, List 0, List 1, 또는 List C) 를 기술할 수도 있다.

[0037]

하나 이상의 PU들을 갖는 리프-CU는 하나 이상의 변환 유닛들 (TUs) 을 또한 포함할 수도 있다. 변환 유닛들은, 위에서 논의된 바와 같이, RTU (또한 TU 큐드트리 구조라 지칭됨) 를 사용하여 특정될 수도 있다. 예를 들어, 리프-CU가 4개의 변환 유닛들로 스플릿되는지의 여부를 스플릿 플래그가 나타낼 수도 있다. 그 후, 각각의 변환 유닛은 4개의 추가의 서브 TU들로 더 스플릿될 수도 있다. TU가 더 이상 스플릿되지 않으면, 그것은 리프-TU로 지칭될 수도 있다. 일반적으로, 인트라 코딩에 대해, 리프-CU에 속하는 모든 리프-TU 들은 동일한 인트라 예측 모드를 공유한다. 즉, 리프-CU의 모든 TU들에 대한 예측된 값들을 계산하기 위해 동일한 인트라 예측 모드가 일반적으로 적용된다. 인트라 코딩에 대해, 비디오 인코더는, TU에 대응하는 CU 의 부분과 오리지널 블록 사이의 차이로서, 인트라 예측 모드를 사용하여 각각의 리프-TU에 대한 잔차 값을 계산할 수도 있다. TU는 PU의 사이즈에 반드시 제한되는 것은 아니다. 따라서, TU들은 PU 보다 더 클 수도 또는 더 작을 수도 있다. 인트라 코딩에 대해, PU 는 동일한 CU 에 대하여 대응하는 리프-TU와 함께 공동 위치될 수도 있다. 일부 예들에서, 리프-TU의 최대 사이즈는 대응하는 리프-CU의 사이즈에 대응할 수도

있다.

[0038] 또한, 리프-CU들의 TU들은 잔차 큐드트리들 (RQTs)로 지칭되는 각각의 큐드트리 데이터 구조들과 또한 연관될 수도 있다. 즉, 리프-CU는 리프-CU가 TU들로 파티션되는 방법을 나타내는 큐드트리를 포함할 수도 있다.

TU 큐드트리의 루트 노드는 일반적으로 리프-CU에 대응하지만, CU 큐드트리의 루트 노드는 일반적으로 트리 블록 (또는 LCU)에 대응한다. 스플릿되지 않은 RQT의 TU들은 리프 TU들로서 지칭된다. 일반적으로, 다르게 지시되지 않는 한, 이 개시물은 리프-CU 및 리프-TU를 각각 지칭하기 위해 용어 CU 및 TU를 사용한다.

[0039] 비디오 시퀀스는 일련의 비디오 프레임들 또는 팩쳐들을 통상 포함한다. 팩쳐들의 그룹 (GOP)은 일련의 하나 이상의 비디오 팩쳐들을 일반적으로 포함한다. GOP는 GOP의 헤더, GOP의 하나 이상의 팩쳐들의 헤더, 또는 그 외의 곳에, GOP에 포함된 팩쳐들의 수를 기술하는 선택스 데이터를 포함할 수도 있다. 팩쳐의 각각의 슬라이스는 각각의 슬라이스에 대한 인코딩 모드를 기술하는 슬라이스 선택스 데이터를 포함할 수도 있다. 비디오 인코더 (20)는 비디오 데이터를 인코딩하기 위해 개개의 비디오 팩쳐들 내의 비디오 블록들에 대해 통상 동작한다. 비디오 블록은 CU 내의 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정된 또는 가변적인 사이즈들을 가질 수도 있고, 특정 코딩 표준에 따라 사이즈가 상이할 수도 있다.

[0040] 일 예로서, HM은 다양한 PU 사이즈들에서의 예측을 지원한다. 특정 CU의 사이즈가 $2Nx2N$ 이라고 가정하면, HM은 $2Nx2N$ 또는 NxN 의 PU 사이즈들에서의 인트라 예측, 및 $2Nx2N$, $2NxN$, $Nx2N$, 또는 NxN 의 대칭적 PU 사이즈들에서의 인터 예측을 지원한다. HM은 $2NxN$, $2NxN$, $nLx2N$, 및 $nRx2N$ 의 PU 사이즈들에서의 인터 예측에 대한 비대칭적 파티셔닝을 또한 지원한다. 비대칭적 파티셔닝에서, CU의 한 방향은 파티션되지 않지만, 나머지 방향은 25% 및 75%로 파티션된다. 25% 파티션에 대응하는 CU의 부분은 "Up", "Down", "Left", 또는 "Right"의 표시가 뒤따르는 "n"에 의해 나타내어진다. 따라서, 예를 들어, " $2NxN$ "는 상부의 $2Nx0.5N$ PU 및 하부의 $2Nx1.5N$ PU로 수평으로 구획되는 $2Nx2N$ CU를 의미한다.

[0041] 이 개시물에서, " NxN " 및 "N 바이 N", 이를 테면, 16x16 샘플들 또는 16 바이 16 샘플들은 수직 차원 및 수평 차원의 면에서 비디오 블록의 샘플 차원들을 언급하기 위해 상호 교환적으로 이용될 수도 있다. 일반적으로, $16x16$ 블록은 수직 방향으로 16 샘플들 ($Y=16$) 및 수평 방향으로 16 샘플들 ($x=16$)을 가질 수도 있다. 마찬가지로, NN 블록은 일반적으로 수직 방향으로 N 개의 샘플들 및 수평 방향으로 N 개의 샘플들을 구비하는데, 여기서 N 은 음이 아닌 정수 값을 나타낸다. 블록에서의 샘플들은 로우들 및 컬럼들로 정렬될 수도 있다.

또한, 블록들은 수평 방향에서의 샘플들의 수가 수직 방향에서의 것과 반드시 동일할 필요는 없다. 예를 들면, 블록들은 NxM 샘플들을 포함할 수도 있으며, 여기서 M 은 N 과 반드시 동일한 것은 아니다.

[0042] CU의 PU들을 사용하는 인트라 예측 또는 인터 예측 코딩에 후속하여, 비디오 인코더 (20)는 CU의 TU들에 대한 잔차 데이터를 계산할 수도 있다. PU들은 공간 도메인 (샘플 도메인으로도 또한 지칭됨)에서의 예측적 샘플 데이터를 생성하기 위한 모드 또는 방법을 기술하는 선택스 데이터를 포함할 수도 있고, TU들은, 잔차 비디오 데이터에 대한, 예를 들면, 이산 코사인 변환(discrete cosine transform; DCT), 정수 변환, 웨이브릿 변환, 또는 개념적으로 유사한 변환과 같은 변환의 적용에 후속하는 변환 도메인에서의 계수들을 포함할 수도 있다. 잔차 데이터는 PU들에 대응하는 예측 값들과 인코딩되지 않은 팩쳐의 샘플들 사이의 샘플 차이들에 대응할 수도 있다. 비디오 인코더 (20)는 CU에 대한 잔차 데이터를 포함하는 TU를 형성하고, 그 다음 TU들을 변환하여 CU에 대한 변환 계수들을 생성할 수도 있다.

[0043] 변환 계수들을 생성하기 위한 임의의 변환들에 후속하여, 비디오 인코더 (20)는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로, 변환 계수 블록의 계수들을 표현하기 위해 사용되는 데이터의 양을 최대한 줄이기 위해 변환 계수들을 양자화하여 추가적인 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 계수들의 일부 또는 전부와 관련된 비트 심도를 감소시킬 수도 있다. 예를 들면, n -비트 값은 양자화 동안 m -비트 값으로 내림 (round down) 될 수도 있는데, 여기서 n 은 m 보다 더 크다.

[0044] 양자화 다음에, 비디오 인코더 (20)는 변환 계수들을 스캐닝하여, 양자화된 변환 계수들을 포함하는 2 차원 매트릭스로부터 1 차원 벡터를 생성할 수도 있다. 스캔은 어레이의 전방에 보다 높은 에너지 (이에 따라 보다 낮은 주파수) 계수들을 두고 어레이의 후방에 보다 낮은 에너지 (이에 따라 보다 높은 주파수)를 두도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성하기 위해 양자화된 변환 계수들을 스캔하기 위한 미리 정의된 스캔 순서를 활용할 수도 있다. 다른 예들에서, 비디오 인코더 (20)는 적응 스캔 (adaptive scan)을 활용할 수도 있다. 양자화된 변환 계수들을 스캐닝하여 1차원 벡터를 형성한 이후, 비디오 인코더 (20)는, 예를 들면, 콘텍스트 적응 가변 길이 코딩 (context adaptive variable length coding; CAVLC), 콘텍스트 적응 이진 산술 코딩 (context adaptive binary

arithmetic coding; CABAC), 신팩스 기반 콘텍스트 적응 이진 산술 코딩(syntax-based context-adaptive binary arithmetic coding; SBAC), 확률 인터벌 파티션 엔트로피(probability interval partitioning entropy; PIPE) 코딩, 또는 다른 엔트로피 인코딩 방법론에 따라, 1차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20)는 비디오 데이터를 디코딩시 비디오 디코더 (30)에 의해 사용하기 위한 인코딩된 비디오와 관련된 신팩스 엘리먼트들을 또한 엔트로피 인코딩할 수도 있다.

[0045] CABAC를 수행하기 위해, 비디오 인코더 (20)는 콘텍스트 모델 내의 콘텍스트를 송신될 심볼에 할당할 수도 있다. 콘텍스트는, 예를 들면, 심볼의 이웃하는 값들이 넌제로 (non-zero)인지 또는 넌제로가 아닌지의 여부에 관련될 수도 있다. CAVLC를 수행하기 위해, 비디오 인코더 (20)는 송신될 심볼에 대한 가변 길이 코드를 선택할 수도 있다. VLC에서의 코드워드들은, 상대적으로 더 짧은 코드들이 고확률 (more probable) 심볼들에 대응하고, 상대적으로 더 긴 코드들이 저확률 (less probable) 심볼들에 대응하게 하도록 구성될 수도 있다. 이렇게 하여, VLC의 사용은, 예를 들면, 송신될 각각의 심볼에 대해 동일한 길이의 코드워드들을 사용하는 것을 통해 비트 절감을 달성할 수도 있다. 확률 결정은 심볼들에 할당된 콘텍스트에 기초할 수도 있다.

[0046] 비디오 인코더 (20)는 신팩스 데이터, 예컨대 블록 기반 신팩스 데이터, 프레임 기반 신팩스 데이터, 및 GOP 기반 신팩스 데이터를, 예를 들면, 프레임 헤더, 블록 헤더, 슬라이스 헤더, 또는 GOP 헤더에서, 비디오 디코더 (30)로 또한 전송할 수도 있다. GOP 신팩스 데이터는 각각의 GOP에서의 다수의 프레임들을 설명할 수도 있고, 프레임 신팩스 데이터는 대응하는 프레임을 인코딩하기 위해 사용된 인코딩/예측 모드를 나타낼 수도 있다.

[0047] 도 2 는 이 개시물의 양태들에 따른 기법들을 구현할 수도 있는 비디오 인코더의 실시예를 나타내는 블록도이다. 일부 예들에서, 이 개시물에 설명된 기법들은 비디오 인코더 (20)의 여러 컴포넌트들 간에 공유될 수도 있다. 일부 예들에서, 프로세서 (도시 생략)는 이 개시물에 설명된 기법들 어느 것 또는 모두를 수행하도록 구성될 수도 있다.

[0048] 비디오 인코더 (20)는 비디오 슬라이스들 내에서 비디오 블록들의 인트라 및 인터 코딩을 수행할 수도 있다. 인트라 코딩은 소정의 비디오 프레임 또는 픽쳐 내에서 비디오에서의 공간적 용장성을 감소시키거나 제거하기 위해 공간적 예측에 의존한다. 인터 코딩은 비디오 시퀀스의 인접한 프레임들 또는 픽쳐들 내의 비디오에서의 시간적 용장성을 감소시키거나 제거하기 위해 시간적 예측에 의존한다. 인트라 모드 (I 모드)는 수개의 공간 기반의 코딩 모드들 중 임의의 것을 지칭할 수도 있다. 단방향 예측(P 모드) 또는 양방향 예측(B 모드)과 같은 인터 모드들은 수개의 시간 기반의 코딩 모드들 중 임의의 것을 지칭할 수도 있다.

[0049] 도 2 의 예에서, 비디오 인코더 (20)는 모드 선택 유닛 (40), 참조 프레임 메모 (66), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56)을 포함한다. 모드 선택 유닛 (40)은 이어서 모션 보상 유닛 (44), 모션 추정 유닛 (42), 인트라 예측 유닛 (46) 및 파티션 유닛 (48)을 포함한다.

비디오 블록 재구성을 위해, 비디오 인코더 (20)는 또한 역양자화 유닛 (58), 역변환 모듈 (60), 합산기 (62) 및 디블로킹 필터 (64)를 포함한다.

[0050] 인코딩 프로세스 동안, 비디오 인코더 (20)는 코딩될 비디오 프레임 또는 슬라이스를 수신한다. 프레임 또는 슬라이스는 복수의 비디오 블록들로 분할될 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 하나 이상의 참조 프레임들에서의 하나 이상의 블록들에 대한 수신된 비디오 블록의 인터 예측 코딩을 수행하여 시간적 예측을 제공한다. 인트라 예측 유닛 (46)은 코딩될 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 인접 블록들에 대한 수신된 비디오 블록의 인트라 예측 코딩을 대안적으로 수행하여 공간적 압축을 제공할 수도 있다. 비디오 인코더 (20)는 다수의 코딩 과정들을 수행하여, 예를 들면, 비디오 데이터의 각각의 블록에 대한 적절한 코딩 모드를 선택할 수도 있다.

[0051] 또한, 파티션 유닛 (48)은, 이전의 코딩 과정들에서의 이전의 파티셔닝 스킁들의 평가에 기초하여, 비디오 데이터의 블록들을 서브블록들로 파티션할 수도 있다. 예를 들면, 파티션 유닛 (48)은, 초기에, 프레임 또는 슬라이스를 LCU들로 파티션하고, 레이트 왜곡 분석 (예를 들면, 레이트 왜곡 최적화)에 기초하여 LCU들의 각각을 서브-CU들로 파티션할 수도 있다. 모드 선택 유닛 (40)은 LCU의 서브-CU들로의 파티셔닝을 나타내는 큐드트리 데이터 구조를 더 생성할 수도 있다. 큐드트리의 리프 노드 CU들은 하나 이상의 PU들 및 하나 이상의 TU들을 포함할 수도 있다.

[0052] 모드 선택 유닛 (40)은, 예를 들면, 여러 결과들에 기초하여 코딩 모드들 중 하나, 인트라 또는 인터를 선택할 수도 있고, 결과적인 인트라 코딩된 블록 또는 인터 코딩된 블록을 합산기 (50)에 제공하여 잔차 블록 데이터

를 생성하고 합산기 (62)에 제공하여 참조 프레임으로서 사용하기 위한 인코딩된 블록을 재구성한다. 모드 선택 유닛 (40)은 또한 십екс 엘리먼트들, 이를 테면, 모션 벡터들, 인트라 모드 표시자들, 파티션 정보, 및 다른 이러한 십екс 정보를 엔트로피 인코딩 유닛 (56)에 제공한다.

[0053] 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 고도로 통합될 수도 있지만, 개념적 목적을 위하여 개별적으로 설명된다. 모션 추정 유닛 (42)에 의해 수행된 모션 추정은 모션 벡터들을 생성하는 프로세스이며, 모션 벡터들은 비디오 블록들에 대한 모션을 예측한다. 예를 들어, 모션 벡터는 현재 프레임 내에서 코딩되고 있는 현재 블록(또는 다른 코딩된 유닛)에 대한, 참조 프레임 내의 예측 블록(또는 다른 코딩된 유닛)에 대한 현재 비디오 프레임 또는 픽쳐 내의 비디오 블록의 PU의 변위를 표시할 수도 있다. 예측 블록은 샘플 차이의 관점에서, 코딩될 비디오 블록과 밀접하게 일치하는 것으로 발견된 블록인데, 샘플 차이는 절대 차의 합(sum of absolute difference; SAD), 제곱 차의 합(sum of square difference; SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 참조 프레임 메모리 (66)에 저장된 참조 픽쳐들의 서브-정수 샘플 포지션들에 대한 값들을 계산할 수도 있다. 예를 들면, 비디오 인코더 (20)는 참조 픽쳐의 1/4 샘플 포지션들, 1/8 샘플 포지션들, 또는 다른 분수의 샘플 포지션들의 값들을 보간할 수도 있다. 따라서, 모션 추정 유닛 (42)은 완전 샘플 포지션들 및 분수적 샘플 포지션들에 대한 모션 검색을 수행하고 분수적 샘플 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0054] 모션 추정 유닛 (42)은 PU의 포지션을 참조 픽쳐의 예측 블록의 포지션과 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 픽쳐는 제 1 참조 픽쳐 리스트(List 0) 또는 제 2 참조 픽쳐 리스트(List 1)로부터 선택될 수도 있는데, 이를 각각은 참조 프레임 메모리 (66)에 저장된 하나 이상의 참조 픽쳐들을 식별한다. 모션 추정 유닛 (42)은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56)과 모션 보상 유닛 (44)에 전송한다.

[0055] 모션 보상 유닛 (44)은 PU의 포지션을 참조 픽쳐의 예측 블록의 포지션과 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 픽쳐는 제 1 참조 픽쳐 리스트(List 0) 또는 제 2 참조 픽쳐 리스트(List 1)로부터 선택될 수도 있는데, 이를 각각은 참조 프레임 메모리 (66)에 저장된 하나 이상의 참조 픽쳐들을 식별한다. 모션 추정 유닛 (42)은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56)과 모션 보상 유닛 (44)에 전송한다.

모션 보상 유닛 (44)은 PU의 포지션을 참조 픽쳐의 예측 블록의 포지션과 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 픽쳐는 제 1 참조 픽쳐 리스트(List 0) 또는 제 2 참조 픽쳐 리스트(List 1)로부터 선택될 수도 있는데, 이를 각각은 참조 프레임 메모리 (66)에 저장된 하나 이상의 참조 픽쳐들을 식별한다. 모션 추정 유닛 (42)은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56)과 모션 보상 유닛 (44)에 전송한다.

모션 추정 유닛 (42)은 PU의 포지션을 참조 픽쳐의 예측 블록의 포지션과 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 픽쳐는 제 1 참조 픽쳐 리스트(List 0) 또는 제 2 참조 픽쳐 리스트(List 1)로부터 선택될 수도 있는데, 이를 각각은 참조 프레임 메모리 (66)에 저장된 하나 이상의 참조 픽쳐들을 식별한다. 모션 추정 유닛 (42)은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56)과 모션 보상 유닛 (44)에 전송한다.

모션 추정 유닛 (42)은 PU의 포지션을 참조 픽쳐의 예측 블록의 포지션과 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 픽쳐는 제 1 참조 픽쳐 리스트(List 0) 또는 제 2 참조 픽쳐 리스트(List 1)로부터 선택될 수도 있는데, 이를 각각은 참조 프레임 메모리 (66)에 저장된 하나 이상의 참조 픽쳐들을 식별한다. 모션 추정 유닛 (42)은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56)과 모션 보상 유닛 (44)에 전송한다.

모션 추정 유닛 (42)은 PU의 포지션을 참조 픽쳐의 예측 블록의 포지션과 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 픽쳐는 제 1 참조 픽쳐 리스트(List 0) 또는 제 2 참조 픽쳐 리스트(List 1)로부터 선택될 수도 있는데, 이를 각각은 참조 프레임 메모리 (66)에 저장된 하나 이상의 참조 픽쳐들을 식별한다. 모션 추정 유닛 (42)은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56)과 모션 보상 유닛 (44)에 전송한다.

[0056] 인트라 예측 유닛 (46)은, 위에 설명된 바와 같이, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 수행된 인터 예측에 대한 대안으로서 현재 블록을 인트라 예측할 수도 있다. 특히, 인트라 예측 유닛 (46)은 현재 블록을 인코딩하는 데 이용하기 위한 인트라 예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 프로세싱 유닛 (46)은 예를 들어, 개별 인코딩 과정들 동안에 다양한 인트라 예측 모드들을 이용하여 현재 블록을 인코딩할 수도 있고, 인트라 예측 모듈 (46) (또는 일부 예들에서, 모드 선택 유닛 (40))은 테스트된 모드들로부터 이용하기에 적합한 인트라 예측 모드를 선택할 수도 있다.

[0057] 예를 들어, 인트라 예측 유닛 (46)은 다양한 테스트된 인트라 예측 모드들에 대한 레이트 왜곡 분석을 이용하여 레이트 왜곡 값들을 계산하고, 테스트된 모드들 중에서 최선의 레이트 왜곡 특성을 갖는 인트라 예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은 일반적으로, 인코딩된 블록 및 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 인코딩되지 않은 블록 사이의 왜곡(또는 에러)의 양, 뿐만 아니라 인코딩된 블록을 생성하는 데 사용되는 비트 레이트(즉, 비트들의 수)를 결정한다. 인트라 예측 유닛 (46)은 어떤 인트라 예측 모드가 그 블록에 대한 최선의 레이트 왜곡 값을 나타내는지를 결정하기 위해 여러 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들을 결정할 수도 있다.

[0058] 블록에 대한 인트라 예측 모드를 선택한 후, 인트라 예측 모듈 (46)은 그 블록에 대한 선택된 인트라 예측 모드를 나타내는 정보를 엔트로피 인코딩 유닛 (56)에 제공할 수도 있다. 엔트로피 인코딩 유닛 (56)은 선택된 인트라 예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20)는 인코딩된 비트 스트림에 구성 데이터를 포함할 수도 있다. 구성 데이터는 복수의 인트라 예측 모드 인덱스 테이블들, 및 복수

의 변경된 인트라 예측 모드 인덱스 테이블들 (또한, 코드워드 매핑 테이블들로서 지칭됨) 을 포함할 수도 있다. 추가로 구성 데이터는 인코딩 콘텍스트들 각각에 대한 이용을 위하여, 최고화를 인트라 예측 모드, 인트라 예측 모드 인덱스 테이블, 및 변경된 인트라 예측 모드 인덱스의 표시들과 함께 여러 블록들에 대한 콘텍스트들을 인코딩하는 정의들을 포함할 수도 있다.

[0059] 비디오 인코더 (20) 는 코딩되고 있는 오리지널 비디오 블록으로부터 모션 선택 유닛 (40) 으로부터의 예측 데이터를 감산함으로써 잔차 비디오 블록을 형성한다. 합산기 (50) 는 이 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 변환 프로세싱 유닛 (52) 은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 잔차 블록에 적용하여, 잔차 변환 계수 값들을 포함하는 비디오 블록을 생성한다. 변환 처리 유닛 (52) 은 DCT 와 개념적으로 유사한 다른 변환들을 수행할 수도 있다. 웨이블릿 변환들, 정수 변환들, 서브 대역 변환들, 또는 다른 유형의 변환들이 또한 이용될 수 있다. 어떠한 경우에도, 변환 프로세싱 유닛 (52) 은 잔차 블록에 변환을 적용하여, 잔차 변환 계수들의 블록을 생성한다. 변환은 잔차 정보를 샘플 도메인으로부터 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다. 변환 프로세싱 유닛 (52) 은 결과적으로 생성된 변환 계수들을 양자화 유닛 (54) 에 전송할 수도 있다. 양자화 유닛 (54) 은 변환 계수들을 양자화하여 비트 레이트를 더 감소시킬 수도 있다.

[0060] 양자화 유닛 (54) 에 의해 수행된 양자화는 가능한 입력 값들의 세트를 가능한 출력 값들의 보다 적은 세트에 매핑하는 것을 포함한다. 양자화는 예를 들어, 입력 값들을 어림처리 함으로써 보다 낮은 정밀도를 갖는 출력 값들을 생성함으로써 구현될 수도 있다. 대안으로서, 양자화는 특정 정밀도를 초과하는 숫자들을 간단히 무시함으로써 실현될 수도 있다. 어떠한 경우에도, 양자화를 수반하는 정밀도의 손실은 종종 입력 값들과 출력값들 사이의 불일치들을 야기한다. 이를 불일치들은 양자화를 수행하는 비디오 코더들, 이를 테면, 비디오 인코더 (20) 에서의 에러 소스이다.

[0061] 비디오 인코더 (20) 에서, 변환 프로세싱 유닛 (52) 으로부터의 변환 계수들이 양자화 유닛 (54) 에 의해 양자화된 후, 이들은 보다 낮은 비트 심도를 가질 수도 있다. 즉, 이들은 표현하는데 있어 보다 적은 바이너리 숫자들을 필요로 할 수도 있고, 이에 의해 인코딩된 비디오의 사이즈를 감소시킨다. 양자화 정도와, 비트 깊이에서의 관련 감소는 양자화 파라미터를 조정함으로써 변경될 수도 있다. 일부 예들에서, 그 후, 양자화 유닛 (54) 은 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수도 있다. 대안으로서, 엔트로피 인코딩 유닛 (56) 이 스캔을 수행할 수도 있다.

[0062] 양자화 이후, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 코딩한다. 예를 들면, 엔트로피 인코딩 유닛 (56) 은 콘텍스트 적응 가변 길이 코딩 (CAVLC), 콘텍스트 적응 이진 산술 코딩 (CABAC), 신택스 기반 콘텍스트 적응 이진 산술 코딩 (SBAC), 확률 인터벌 구획 엔트로피 (PIPE) 코딩, 또는 다른 엔트로피 인코딩 기법들을 수행할 수도 있다. 콘텍스트 기반 엔트로피 코딩의 경우에, 콘텍스트는 이웃하는 블록들에 기초할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 코딩에 후속하여, 인코딩된 비트스트림은 다른 디바이스 (예를 들어, 비디오 디코더 (30)) 에 전달되거나 또는 후속 전송 또는 취출을 위하여 아카이브될 수도 있다.

[0063] 역양자화 유닛 (58) 및 역변환 유닛 (60) 은 각각, 역양자화 및 역변환을 적용하여, 예를 들어, 참조 블록의 일부로서 나중의 사용을 위하여 샘플 도메인의 잔차 블록을 재구성한다. 재구성된 잔차 블록은 위에 설명된 바와 같이, 양자화 유닛 (54) 에 의해 도입된 에러의 결과로서, 오리지널 잔차 블록과는 상이할 수도 있다.

[0064] 모션 보상 유닛 (44) 은 참조 프레임 메모리 (66) 의 프레임들 중 하나의 예측 블록에 잔차 블록을 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44) 은 모션 추정에서 사용하기 위한 서브-정수 샘플 값들을 계산하기 위해, 재구성된 잔차 블록에 하나 이상의 보간 필터들을 또한 적용할 수도 있다. 합산기 (62) 는 디블로킹 필터 (64) 에 전송되는 재구성된 잔차 블록을 생성하기 위하여 모션 보상 유닛 (44) 에 의해 생성된 모션 보상된 예측 블록 재구성된 잔차 블록을 추가한다.

[0065] 디블로킹 필터 (64) 는 인접하는 비디오 블록들 간의 에지를 가로질러 필터링을 수행한다. 비디오 인코더 (20) 의 일부 구현들에 있어서, 디블로킹 필터 (64) 는 비디오의 상이한 부분들을 필터링할 때 상이한 필터링 강도들을 적용할 수도 있으며, 비디오의 부분들이 전혀 필터링되지 않을 수도 있다. 따라서, 디블로킹 필터 (64) 는 에지를 필터링할지의 여부와, 어떤 필터링 강도를 이용할지에 대한 결정을 행하도록 구성될 수도 있다. 디블로킹 필터 (64) 가 에지를 필터링하도록 결정하면, 이는 샘플들의 값들을 조정함으로써 에지의 각각의 사이드에서의 샘플들 간의 트랜지션들을 평활화할 수도 있다.

[0066] 필터링 강도를 결정하기 위하여, 디블로킹 필터 (64)는 입력으로서 인접하는 블록들로부터 하나 이상의 샘플들을 이용할 수도 있다. 블록들로부터의 샘플들은 또한, 계산 조정들의 목적으로 입력들로서 이용될 수도 있다. 예를 들어, 디블로킹 필터 (64)는 조정 값을 계산하고, 그 조정 값을 특정 샘플에 추가함으로써 특정 샘플을 조정할 수도 있다. 조정 값은 샘플이 속하는 블록 에지에 대하여 결정되는 필터링 강도 뿐만 아니라 블록들로부터의 다른 샘플들에 의존할 수도 있다.

[0067] 이 개시물의 양태들에 따르면, 디블로킹 필터 (64)는 제한된 비트 심도를 갖는 입력들에 의존함으로써 샘플들을 조정하는 최적화된 모드를 구현할 수도 있다. 추가로, 디블로킹 필터 (64)는 또한, 필터링 강도, 및 우선적으로 필터링 할지의 여부를 결정하기 위하여 제한된 비트 심도를 갖는 입력들을 이용할 수도 있다. 필요 한 입력들의 비트 심도를 감소시키기 위하여 디블로킹 필터 (64)에 이용될 수도 있는 기법들은 추가적인 도면들 및 예들에 대하여 아래 자세하게 설명된다.

[0068] 디블로킹 필터 (64)는 후속하는 인코딩된 예측들이 기초로 하는 참조 데이터에 그 출력이 통합되기 때문에 인루프 필터이다. 특히, 디블로킹 필터 (64)에 의해 필터링되었던 재구성된 비디오 블록들은 참조 프레임 메모리 (66)에 저장될 수도 있다. 그 후, 참조 프레임 메모리 (66)로부터의 재구성된 그리고 필터링된 비디오 블록들은 비디오에서의 다른 블록들을 코딩하기 위하여 참조 블록들로서 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 이용될 수도 있다.

[0069] 디블로킹 필터 (64)가 인루프 필터로서 묘사되어 있지만, 본 발명의 기법들은 포스트 루프 필터들과 같은 다른 유형의 필터들에도 또한 적용될 수 있다. 또한, 도 2의 콘텍스트에서, 추가적인 필터들 (인루프, 포스트 루프 등)이 또한 디블로킹 필터 (64)에 더하여 이용될 수도 있다.

[0070] 도 3은 이 개시물의 양태들에 따른 기법들을 구현할 수도 있는 비디오 디코더의 일 예를 나타내는 블록도이다. 비디오 디코더 (30)는 이 개시물의 기법들의 모두 또는 어느 것을 수행하도록 구성될 수도 있다. 일 예로서, 디블로킹 필터 (82)는 이 개시물에 서령된 기법들 모두 또는 어느 것을 수행하도록 구성될 수도 있다. 그러나, 이 개시물의 양태들은 이렇게 제한되는 것은 아니다. 일부 예들에서, 이 개시물에 설명된 기법들은 비디오 디코더 (30)의 여러 컴포넌트들 간에 공유될 수도 있다. 일부 예들에서, 프로세서 (도시 생략)는 이 개시물에 설명된 기법들 어느 것 또는 모두를 수행하도록 구성될 수도 있다.

[0071] 도 3의 예에서, 비디오 디코더 (30)는 엔트로피 디코딩 유닛 (70), 모션 보상 유닛 (72), 인트라 예측 유닛 (74), 역양자화 유닛 (76), 역변환 유닛 (78), 참조 프레임 메모리 (84), 합산기(80) 및 디블로킹 필터 (82)를 포함한다. 비디오 디코더 (30)는 몇몇 예들에서, 비디오 인코더 (20)(도 2)에 대해 설명된 인코딩 과정에 일반적으로 역순인 디코딩 과정을 수행할 수도 있다. 모션 보상 유닛 (72)은 엔트로피 디코딩 유닛 (70)으로부터 수신된 모션 벡터들에 기초하여 예측 데이터를 생성할 수도 있고, 한편, 인트라 예측 유닛 (74)은 엔트로피 디코딩 유닛 (70)으로부터 수신된 인트라 예측 모드 표시자들에 기초하여 예측 데이터를 생성할 수도 있다.

[0072] 디코딩 프로세스 중에, 비디오 디코더 (30)는 비디오 인코더 (20)로부터 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신택스들을 표현하는 인코딩된 비디오 비트스트림을 수신한다. 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 양자화된 계수들, 모션 벡터들, 또는 인트라 예측 모드 표시자들, 및 다른 신택스 엘리먼트들을 생성하기 위해 비트스트림을 엔트로피 디코딩한다. 엔트로피 디코딩 유닛 (70)은 모션 벡터들 및 다른 신택스 엘리먼트들을 모션 보상 유닛 (72)에 전달한다. 비디오 디코더 (30)는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신택스 엘리먼트들을 수신할 수도 있다.

[0073] 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩되면, 인트라 예측 유닛 (74)은 현재 프레임 또는 꾹쳐의 이전에 디코딩된 블록들로부터의 데이터와 시그널링된 인트라 예측 모드에 기초하여 현재 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터 코딩된 (즉, B, P 또는 GPB) 슬라이스로서 코딩되면, 모션 보상 유닛 (72)은 엔트로피 디코딩 유닛 (70)으로부터 수신된 다른 신택스 엘리먼트들과 모션 벡터들에 기초하여 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 참조 꾹쳐 리스트들 중 하나 내의 참조 꾹쳐들 중 하나로부터 생성될 수도 있다. 비디오 디코더 (30)는 참조 프레임 메모리 (84)에 저장된 참조 꾹쳐들에 기초한 디폴트 구성 기법들을 이용하여, 참조 프레임 리스트들, List 0 및 List 1을 구성할 수도 있다. 모션 보상 유닛 (72)은 모션 벡터들 및 다른 신택스 엘리먼트들을 파싱하는 것에 의해 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 그 예측 정보를 사용하여, 디코딩되고 있는 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들면, 모션 보상 유닛 (72)은, 비디오 슬라이스의 비디오 블록들을 코딩하기 위해 사용되는 예측 모드 (예를 들면,

인트라 예측 또는 인터 예측), 인터 예측 슬라이스 타입 (예를 들면, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스에 대한 하나 이상의 참조 픽쳐 리스트들에 대한 구성 정보, 슬라이스의 각각의 인터 인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터 코딩된 비디오 블록에 대한 인터 예측, 및 현재의 비디오 슬라이스의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정하기 위해, 수신된 신팩스 엘리먼트들의 몇몇을 사용한다.

[0074] 모션 보상 유닛 (72)은 보간 필터들에 기초한 보간을 또한 수행할 수도 있다. 모션 보상 유닛 (72)은 비디오 블록들의 인코딩 동안 비디오 인코더 (20)에 의해 임용되는 것과 같이 보간 필터들을 사용하여 참조 블록들의 서브-정수 샘플들에 대한 보간된 값들을 계산할 수도 있다. 이 경우, 모션 보상 유닛 (72)은 수신된 신팩스 엘리먼트들로부터 비디오 인코더 (20)에 의해 사용되는 보간 필터들을 결정하고 보간 필터들을 사용하여 예측 블록을 생성할 수도 있다.

[0075] 역양자화 유닛 (76)은 비트스트림으로 제공되고 엔트로피 디코딩 유닛 (70)에 의해 디코딩된 양자화된 변환 계수들을 역양자화, 즉, 양자화 해제한다. 역양자화 프로세스는 양자화의 정도, 및, 마찬가지로, 적용되어야 하는 역양자화의 정도를 결정하기 위해, 비디오 슬라이스의 각각의 비디오 블록에 대해 비디오 디코더 (30)에 의해 계산된 양자화 파라미터 (QP_Y)의 사용을 포함할 수도 있다.

[0076] 역변환 유닛 (78)은 샘플 도메인에서의 잔차 블록들을 생성하기 위해 변환 계수들에 대해 역변환, 예를 들면, 역 DCT, 역정수 변환, 또는 개념적으로 유사한 역변환 프로세스를 적용한다.

[0077] 모션 보상 유닛 (72)이 모션 벡터들 및 다른 신팩스 엘리먼트에 기초하여 현재의 비디오 블록에 대한 예측 블록을 생성한 후에, 비디오 디코더 (30)는 역 변환 모듈 (78)로부터의 잔차 블록들을 모션 보상 유닛 (82)에 의해 발생된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (80)는 이 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다.

[0078] 합산기 (80)에 의해 생성된 디코딩된 비디오 블록들에서의 샘플들은 이전에 설명된 바와 같이, 인코딩 프로세스 동안에 이용된 양자화 또는 유사한 압축 기법들로부터 야기되는 에러를 포함할 수도 있다. 이 에러는 블로킹 아티팩트들을 야기할 수도 있다.

[0079] 디블로킹 필터 (82)는 블로킹 아티팩트들을 제거하기 위하여 합산기 (80)로부터 디코딩된 블록들을 필터링한다. 디블로킹 필터 (82)는 필터링할지의 여부를 결정하는 것, 필터링 강도를 결정하는 것, 및 블로킹 에지를 가로질러 트랜지션들을 평활화하기 위해 디코딩된 블록들에서의 샘플들을 조정하는 것을 포함하는 필터링 프로세스를 구현할 수도 있다. 필터링 프로세스의 일부 또는 어느 것의 부분은 하나 이상의 샘플들로부터 유도되는 입력들에 의존할 수도 있다. 이 개시물에 따르면, 입력들은 이들이 유도되는 샘플들의 비트 심도보다 더 낮은 비트 심도를 가질 수도 있다. 입력들의 비트 심도를 감소시키는데 이용될 수도 있는 기법들은 추가적인 도면들 및 예들에 대하여 아래 보다 자세하게 설명된다.

[0080] 디블로킹 필터 (82) 외의 필터들이 또한, 비디오 품질을 향상시키기 위하여 (코딩 루프 내에서 또는 코딩 루프 후에) 이용될 수도 있다. 디블로킹 필터 (82)로부터의 디코딩된 비디오 블록들은 참조 프레임 메모리 (84)에 저장되고, 이 메모리는 후속하는 모션 보상에 대한 참조 픽쳐들을 제공한다. 참조 프레임 메모리 (84)는 또한 비디오 디코더 (30)로부터의 출력을 위한 디코딩된 비디오를 저장한다. 비디오 출력은 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상에 이후에 제시될 수도 있다.

[0081] 도 4a는 다수의 비디오 블록들을 포함하는 비디오 슬라이스를 설명하는 개념도이다. 도시된 바와 같이, 비디오 슬라이스 (402)는 16 개의 비디오 블록들을 포함한다. 비디오 슬라이스 (402)는 도 3의 비디오 디코더 (30)에 의해 또는 도 2의 비디오 인코더 (20)에 의해 디코딩 또는 인코딩될 수도 있다. 비디오 슬라이스 (402)는 또한 예를 들어, 도 2의 디블로킹 필터 (64) 또는 도 3의 디블로킹 필터 (82)에 의해 필터링될 수도 있다. 비디오 슬라이스 (402)의 필터링은 제한된 비트 심도를 가진 입력들의 이용과 관련된 이 개시물의 양태들에 따라 구성될 수도 있다.

[0082] 비디오 슬라이스 (402)는 각각, 높이에 있어서 4 개의 샘플과 폭에 있어서 4 개의 샘플을 갖는 인접하는 블록들 (404, 406)을 포함한다. 인접하는 블록들 (404, 406)은 에지 (408)를 공유한다. 블록들 (404, 406) 및 에지 (408)는 도 4b에 보다 자세하게 도시되어 있다.

[0083] 도 4b의 개념도에서, 블록들 (404 및 406)을 포함하는 개별적인 샘플들, 예를 들어, 샘플 (410)이 가시화된다. 에지 (408)의 한 사이드 상의 샘플들은 문자 "p"로 지정되는 한편, 에지의 다른 사이드 상의 샘플

들은 문자 "q"로 지정된다. 예지 (408)의 각각의 사이트 상에서, 예지에 최근접한 샘플들은 아래 첨자의 숫자 0으로 지정되는 한편, 예지로부터 더 멀리 있는 샘플들은 점점 더 큰 정수들로 지정되고 3으로 끝난다.

동일한 문자들과 수들은 일부 실시형태들에서, 로우들 각각이 개별적으로 필터링되기 때문에 로우들 (412, 414, 416, 418) 각각에서 반복된다. 따라서, 필터링 프로세스는 로우 (412)에서 시작할 수도 있고 로우 (414), 로우 (416) 및 로우 (418)로 순차적으로 이동할 수도 있다.

[0084] 예지 (408)가 도 4b에 수직방향으로 도시되어 있지만, 유사한 문자화 및 넘버링 방식 수평 방향 예지들을 필터링하는 목적을 위하여 또한 채택될 수도 있다. 수평방향 예지의 경우에, p_3 내지 p_0 및 q_0 내지 q_3 로 지정된 샘플들은 수평방향 로우들 보다는 수직방향 로우들로 배열되며, 샘플들은 로우가 아닌 컬럼으로 순차적으로 필터링될 수도 있다.

[0085] 도 5는 제한된 비트 심도를 갖는 입력들을 이용하는 디블로킹 필터에 대한 예시적인 프로세스를 나타내는 플로우차트이다. 프로세스 (500)의 엘리먼트들은 비디오 인코더 (20)의 디블로킹 필터 (64)에 대하여 부분적으로 설명되지만, 프로세스는 다른 컴포넌트들에 의해, 이를 테면, 디코더 (30)의 디블로킹 필터 (82)에 의해 또는 비디오 필터들의 다른 유형들에 의해 수행될 수도 있다. 프로세스 (500)의 엘리먼트들은 또한 예지 (408)뿐만 아니라 블록 (404, 406)에 대하여 로우 (412), 및 그 내부에 위치된 샘플들에 대하여 설명된다.

그러나, 프로세스 (500)는 블록들, 예지들, 및 샘플들의 여러 다른 조합들 및 구성을 상에서 수행될 수도 있다. 도 5에 대하여 설명된 모든 실시형태들은 서로 조합하여 또는 개별적으로 구현될 수도 있다.

[0086] 프로세스 (500)는 예지 (408)에서 블로킹 아티팩트를 없애기 위하여 블록들 (404, 406)에서의 샘플들을 필터링할 때 디블로킹 필터 (64)에 의해 적용될 수도 있다. 프로세스 (500)의 엘리먼트들, 이를 테면, 엘리먼트들 (502, 506, 508)은 하나 이상의 샘플들을 수반하는 계산들을 포함한다. 프로세스 (500)는 계산들이 오직 하나의 로우로부터의 샘플들에만 수반되도록, 한번에 하나의 로우, 예를 들어, 로우 (412)에 적용된다. 대안으로서, 프로세스 (500)는 계산들이 하나 보다 많은 로우로부터의 샘플들에 수반되도록 적용될 수 있다.

[0087] 프로세스 (500)는 엘리먼트 (502)로 시작하며, 여기에서 디블로킹 필터 (64)가 적어도 하나의 샘플에 기초하여 필터링 입력을 결정한다. 적어도 하나의 샘플은 로우 (412)로부터 유래할 수도 있거나 또는 로우들의 임의의 조합으로부터 유래할 수도 있다. 디블로킹 필터 (64)는 하나 이상의 산술 연산들, 하나 이상의 논리 연산들, 하나 이상의 비트와이즈 연산들, 또는 이들의 조합에 따라 샘플들로부터 필터링 입력을 결정할 수도 있다. 필터링 입력은 샘플들에 더하여, 여러 파라미터들 또는 상수들로부터 결정될 수도 있다.

[0088] 엘리먼트 (504)에서, 디블로킹 필터 (64)는 엘리먼트 (502)에서 결정되었던 필터링 입력의 비트 심도를 제한한다. 필터링 입력의 제한된 비트 심도는 입력을 결정하는데 이용되었던 샘플들의 비트 심도보다 작다. 예를 들어, 필터링 입력은 8비트들의 비트 심도를 각각 갖는, 로우 (412)에서의 샘플들로부터 결정될 수도 있다. 3개의 샘플들의 총 비트 심도는 24비트들이고, 이에 따라 필터링 입력은 24비트들보다 더 낮은 임의의 비트 심도로 제한될 수도 있다. 일부 구현예들에서, 디블로킹 필터 (64)는 하나 보다 많은 샘플로부터 필터링 입력을 결정할 수도 있고, 입력의 비트 심도가 단일의 샘플의 비트 심도 보다 더 낮도록 입력의 비트 심도를 제한할 수도 있다. 예를 들어, 통합된 샘플들의 총 비트 심도는 24비트들일 수도 있지만, 필터링 입력의 제한된 비트 심도는 예를 들어, 6비트들일 수도 있다. 어느 경우에도, 디블로킹 필터 (64)가 필터링 입력에 부여하는 제한은 입력의 비트 심도를 감소시켜, 입력이 공간 또는 메모리를 덜 차지하게 한다. 엘리먼트 (504)에서 달성되는 공간 또는 메모리 절감들은 디블로킹 필터 (64)가 하드웨어로 구현될 때 유리할 수도 있다.

[0089] 필터링 입력의 비트 심도는 여러 기법들에 의해 엘리먼트 (504)에서 제한될 수도 있다. 예를 들어, 비트 심도는 입력으로부터 최상위 비트들 중 적어도 하나를 드롭시키거나 또는 소정의 범위 내로 입력을 제약함으로써 제한될 수도 있다. 일부 실시형태들에서, 비트 심도는 양자화에 의해 제한될 수도 있다. 비트 심도는 또한 입력 사이즈 및 대응하는 저장 요건들을 감소시킬 수 있는 하나 이상의 산술, 논리 또는 비트와이즈 연산들을 적용함으로써 제한될 수도 있다.

[0090] 엘리먼트 (504)에서 필터링 입력의 비트 심도를 제한하는 것은 특정 환경들에서 디블로킹 필터 (64)의 유효성에 악영향을 줄 수도 있다. 예를 들어, 비트 심도가 입력의 크기를 제약함으로써 제한되면, 디블로킹 필터 (64)의 필터링 강도가 결과적으로 제한될 수도 있다. 다른 예에서, 비트 심도가 입력의 정밀도를 감소시킴으로써 제한되면, 필터의 적응능력이 결과적으로 제한될 수도 있다. 입력의 비트 심도를 제한하는 것으로부터 야기되는 임의의 악영향은 제한의 심각성에 비례할 수도 있다. 따라서, 제한 정도는 공간 효율과 필터링

유효성 간에 밸런스를 가하도록 선택될 수도 있다. 그러나, 이후에 자세히 설명될 바와 같이, 디블로킹 필터 (64)의 유효성은 특정 환경들에서 특정 필터링 구현들에 대하여 전혀 영향을 받지 않을 수도 있다.

[0091] 디블로킹 필터 (64)는 엘리먼트 (506)에서 조정 값을 결정하기 위해 엘리먼트 (504)에서 생성되었던 제한된 비트 심도를 갖는 입력을 이용할 것이다. 조정 값은 디블로킹 필터 (64)가 필터링되고 있는 샘플에 적용되는 조정의 양에 대응한다. 입력은 다수의 샘플들에 대한 조정 값을 계산하거나 또는 단일의 샘플에 대한 조정 값을 계산하기 위해 이용될 수도 있다. 입력에 더하여, 조정 값이 하나 이상의 샘플들에 의존할 수도 있다. 조정 값은 하나 이상의 산술 연산들, 하나 이상의 논리 연산들, 하나 이상의 비트와이즈 연산들, 또는 이들의 조합들에 따라 샘플들로부터 결정될 수도 있다. 조정 값은 입력 및 조정 값에 더하여 상술들 또는 여러 파라미터들로부터 결정될 수도 있다.

[0092] 일부 실시형태들에서, 디블로킹 필터 (64)는 미리 결정된 범위 내에서 엘리먼트 (506)에서 결정된 조정 값을 제약할 것이다. 일 실시형태에서, 조정 값을 제약하는 것은 과도한 필터링을 방지한다.

[0093] 조정 값이 엘리먼트 (506)에서 결정된 후, 이는 엘리먼트 (508)에서 샘플과 결합하여 필터링된 샘플을 생성한다. 디블로킹 필터 (64)는 통상적으로 샘플과 조정 값을 결합하기 위한 합산에 의존하지만, 다른 산술 연산들, 논리 연산들, 비트와이즈 연산들 또는 이들의 조합이 또한 이용될 수도 있다. 블록 (508)에서 생성되는 필터링된 샘플은 프로세스 (500)의 출력이다.

[0094] 도 6은 디블로킹 필터링 프로세스 (600)의 다른 실시형태를 나타내는 플로우차트이다. 프로세스 (600)는 도 4의 블록들 (404, 406), 에지 (408), 로우 (412) 및 연관된 샘플들 상에 동작하는 디블로킹 필터에 대하여 아래 설명된다. 프로세스 (600)는 블록들, 에지들, 로우들, 컬럼들, 및 샘플들의 상이한 구성들에도 또한 적용될 수도 있다. 또한, 프로세스 (600)는 라인 버퍼들에서의 입력들의 저장을 고려하며, 이는 디블로킹 필터들의 하드웨어 구현들에서 통상적으로 발생한다. 그러나, 프로세스 (600)는 단지 일 예에 불과하며, 본 개시물의 기법들은 다른 예들에서 소프트웨어, 또는 하드웨어와 소프트웨어의 조합에 의해 구현될 수도 있다.

[0095] 에지와 연관된 블로킹 아티팩트의 효과들을 완화시키기 위하여, 디블로킹 필터는 한번에 하나의 로우 또는 컬럼에서의 샘플들을 조정할 수도 있다. 예를 들어, 에지 (408)와 연관된 블로킹 아티팩트를 필터링하기 위하여, 디블로킹 필터는 무엇보다도 로우 (412)에서의 샘플들을 조정할 수도 있다. 예를 들어, 필터는 샘플들에 대한 강 또는 약 필터링을 적용할 수도 있다. 강 필터링과 약 필터링 사이의 선택은 다수의 로우들 또는 컬럼들에 대하여 한번 행해질 수도 있고, 결정을 행할 때 로우들 또는 컬럼들의 일부가 고려되지 않을 수도 있다. 로우 (412)가 고려되는 것으로 가정하면, 강/약 필터링 결정은 로우 (412)에서의 에지 (408)의 각각의 사이드 상에 위치된 4개의 샘플들 (즉, p₃, p₂, p₁, p₀ 및 q₀, q₁, q₂, q₃)에 의존할 수도 있다.

[0096] 강/약 결정에 의존하여, 디블로킹 필터는 강 필터링 모드 또는 약 필터링 모드에 따라 조정 값을 계산한다. 강/약 필터링 모드의 일 유형은 드래프트에서 제공된다. 드래프트의 강 필터링 모드는 에지 (408)의 각각의 사이드 상의 모든 4개의 샘플들, 즉, p₃, p₂, p₁, p₀ 및 q₀, q₁, q₂, q₃을 이용한다. 드래프트의 약 필터링 모드는 각각의 사이드 상에서 에지 (408)에 최근접하여 위치된 오직 2개 또는 3개의 샘플들만을 이용한다. 따라서, 결국, 디블로킹 필터는 드래프트에 따라 강/약 필터링을 결정하고 강 또는 약 필터링을 적용하기 위하여 에지 (408)의 각각의 사이드 상에서 최대 4개의 샘플들을 필요로 한다. 필터링 프로세스 동안에 이들 샘플들의 이용을 용이하게 하기 위하여, 디블로킹 필터는 아래 설명된 바와 같이 라인 버퍼들에 샘플들 중 일부를 저장한다.

[0097] 디블로킹 필터는 하드웨어로 구현될 수도 있다. 예를 들어, 디블로킹 필터는 비디오 인코더 또는 디코더의 부분으로서 칩 상에서 구현될 수도 있다. 라인 버퍼는 외부 메모리 액세스에 관련된 문제들을 회피하기 위하여 온-칩 (내부) 메모리로서 구현될 수도 있다. 디블로킹 필터는 필터링 프로세스 동안에 외부 메모리로부터 샘플을 재폐지할 필요가 없도록 필터링 프로세스에 이용된 샘플을 라인 버퍼에 저장할 수도 있다.

[0098] 온-칩 메모리는 통상적으로 매우 제한된 사이즈를 가지며, 이에 따라 온-칩 메모리는 데이터량이 가능한 가장 낮게 내부에 일시적으로 저장된 상태로 되는 것이 유리하다. 디블로킹 필터에 의해 요구되는 라인 버퍼들의 사이즈는 라인 버퍼들에 저장되는 필터링 입력들의 총 비트 심도에 의존한다. 프로세스 (600)의 이점은 감소된 비트 심도를 갖는 입력들을 이용한다는 것이며, 이 감소된 비트 심도는 드래프트에 의해 특정된 입력들보다 적은 라인 버퍼 공간을 요구한다. 칩 제조 동안에, 추가적인 라인 버퍼 용량은 추가적인 비용들을 내포하기 때문에 라인 버퍼들의 사이즈를 감소시키는 것이 유리하다.

[0099] 디블로킹 필터에 의해 이용된 모든 샘플들이 라인 버퍼들에 반드시 저장될 필요가 있는 것은 아니다. 프로세스 (600) 의 예에서, 블록 에지의 p 사이드 상의 샘플들은 라인 버퍼들에 저장되지만, q 사이드 상의 샘플들은 저장되지 않는다. 라인 버퍼들에 저장된 샘플들은 메모리의 별도의 영역, 이를 테면 프레임 버퍼에 저장될 수도 있다. 대안으로서, 프로세스 (600) 에서와 같이, 샘플들은 디블로킹 필터에서의 이들의 이용 직전에 사전 코딩 스테이지에서 계산될 수도 있고 어떠한 메모리로부터 판독되지 않을 수도 있다. 어떠한 경우에도, 라인 버퍼에 달리 저장된 적어도 하나의 입력이 더 낮은 비트 심도를 가진 입력으로 대체되는 한, 라인 버퍼 용량은 절감될 것이다.

[0100] 프로세스 (600) 는 강 필터링 및 약 필터링 뿐만 아니라 강/약 결정을 포함한다. 그러나, 프로세스 (600) 가 강/약 결정 및 강 필터링에 대해 이용하는 입력들의 총 비트 심도는 드래프트에 의해 특정되는 입력들의 비트 심도보다 더 낮다. 구체적으로, 일 실시형태에서, 프로세스 (600) 가 실행될 때, 샘플 (p_3) 은 강/약 결정의 목적으로 샘플 (p_2) 로 대체되고, 이는 강 필터링의 목적으로 대안의 입력 (p_{short}) 으로 대체된다. 에지 (408) 의 양쪽 사이드들 상에서의 필터링 프로세스 대칭성을 유지하기 위하여, 동등한 대체들이 q_3 에 대해서도 또한 행해진다. 대안의 입력 (p_{short}) 은 도 5 에서의 프로세스 (500) 의 엘리먼트 (504) 에 대하여 설명되었던 제한된 비트 심도를 갖는 필터링 입력의 일 실시형태이다. p_{short} 의 제한된 비트 심도는 이것이 p_3 보다 라인 버퍼들에서의 공간을 덜 점유하는 것을 허용하며, 이에 의해 디블로킹 필터에 의해 이용된 라인 버퍼 용량을 감소시킨다.

[0101] 일 실시형태에서, 디블로킹 필터가 강 필터링을 적용할 때, 3 개의 샘플들, 예를 들어, p_2 , p_1 , p_0 및 q_0 , q_1 , q_2 가 에지 (408) 의 각각의 사이드 상에서 조정된다. 결과적으로 조정된 샘플들은 p'_2 , p'_1 , p'_0 및 q'_0 , q'_1 , q'_2 로 지정된다. 강 필터링 계산들은 드래프트에서 다음과 같이 특정된다:

$$p'_0 = p_0 + \text{Clip}_{(-2*tc)-(2*tc)}((p_2 + 2*p_1 - 6*p_0 + 2*q_0 + q_1 + 4) >> 3)$$

$$p'_1 = p_1 + \text{Clip}_{(-2*tc)-(2*tc)}((p_2 - 3*p_1 + p_0 + q_0 + 2) >> 2)$$

$$p'_2 = p_2 + \text{Clip}_{(-2*tc)-(2*tc)}((2*p_3 - 5*p_2 + p_1 + p_0 + q_0 + 4) >> 3)$$

$$q'_0 = q_0 + \text{Clip}_{(-2*tc)-(2*tc)}((p_1 + 2*p_0 - 6*q_0 + 2*q_1 + q_2 + 4) >> 3)$$

$$q'_1 = q_1 + \text{Clip}_{(-2*tc)-(2*tc)}((p_0 + q_0 - 3*q_1 + q_2 + 2) >> 2)$$

$$q'_2 = q_2 + \text{Clip}_{(-2*tc)-(2*tc)}((p_0 + q_0 + q_1 - 5*q_2 + 2*q_3 + 4) >> 3)$$

[0102] [0103] 위의 식의 각각에서, 필터링된 샘플 (p'_i 또는 q'_i) 은 오리지널 샘플 (p_i 또는 q_i), 및 $\text{Clip}_{(-2*tc)-(2*tc)}(\dots)$ 에 의해 특정된 조정 값에 의해 결정된다. Clip 함수는 필터가 과도하게 큰 조정들을 행하는 것을 방지한다. 일 실시형태에서, Clip 함수는 다음과 같이 정의된다:

$$\text{Clip}_{a-b}(x) = \{a, \text{if } x < a; b, \text{if } x > b; x, \text{otherwise}\}$$

[0104] [0105] 따라서, $\text{Clip}_{(-2*tc)-(2*tc)}(\dots)$ 의 출력은 $-2*tc$ 와 $2*tc$ 사이에 제약되며, tc 는 임계값이다. Clip 함수의 적용은 제약들의 특정 값들과 무관하게 "클리핑 (clipping)" 으로서 지칭될 수도 있다.

[0106] 위에 도시된 강 필터링 계산은 이전에 언급된 바와 같이 에지의 각각의 사이드 상에서의 4 개의 샘플들을 이용한다. 최외각 샘플들 (p_3 및 q_3) 은 다른 샘플들에 대한 강 필터링 계산들에 이용되지만 자체 필터링되지 않는다. 계산들에 이용된 샘플들 중, p_3 및 q_3 은 이들이 에지 (408)로부터 가장 멀리 위치되기 때문에 가장 적은 식들에 나타난다. 따라서, 이들은 필터링 품질의 손실 없이도 비트 심도 감소를 위한 최상의 후보들을 제공한다.

[0107] 프로세스 (600) 는 엘리먼트 (602) 에서 시작하며, 여기에서 디블로킹 필터는 단축된 입력 (p_{short}) 을 계산한다.

예를 들어, 일부 실시형태들에서, 여러 기법들을 이용하여 단축된 입력 값을 계산할 수도 있다:

[0108] $p_{short} = Clip_{a-b}(p_3 - 2*p_2 + p_1)$

[0109] 또는,

[0110] $p_{short} = Clip_{a-b}(2*p_3 - 5*p_2 + p_1 + 2*p_0).$

[0111] 이들 예들의 공식 양쪽 모두에서, p_3 을 포함하는 샘플들은 산술 연산에 의해 결합되고, 결과의 비트 심도는 Clip 함수의 적용에 의해 제한된다. 산술 연산에 의한 샘플들의 결합 및 Clip 함수의 적용은 도 5로부터의 프로세스 (500) 의 엘리먼트들 (502 및 504) 에 각각 대응하는 특정 실시형태들이다. p_3 이 p_{short} 를 계산하는데 이용되고 있지만, p_{short} 가 라인 버퍼의 자리에 저장되기 때문에 p_3 은 라인 버퍼의 그 자리에 저장되지 않는다.

[0112] 엘리먼트 (604) 에서, p_{short} 는 p_2 , p_1 , 및 p_0 와 함께 라인 버퍼들에 저장된다. 이 제한된 비트 심도에 의해, p_{short} 는 p_3 이 점유하는 것보다 라인 버퍼들에서의 더 적은 공간을 점유한다. 따라서, 디블로킹 필터에 의해 요구되는 총 라인 버퍼 용량은 감소되고, 이는 예를 들어, 이 개시물의 기법들에 따라 비디오 인코더들 또는 디코더들을 구현하는 칩들을 제조하는 비용을 감소시킬 수도 있다.

[0113] 엘리먼트 (606) 에서, 디블로킹 필터는 강 및 약 필터링 간의 결정을 행하기 위하여 라인 버퍼들로부터 샘플들을 판독한다. 강/약 결정은 드래프트에서 다음과 같이 정의된다:

[0114] 4개의 샘플 에지 세그먼트 ($i=0\dots3$) 에 대해,

[0115] $sw_i = 2(|p_{2,i} - 2p_{1,i} + p_{0,i}| + |q_{0,i} - 2q_{1,i} + q_{2,i}|) < (\beta / 4)$ (조건 1)

[0116] and $(|p_{3,i} - p_{0,i}| + |q_{0,i} - q_{3,i}|) < (\beta / 8)$ (조건 2)

[0117] and $|p_{0,i} - q_{0,i}| < ((5t_c + 1) / 2)$ (조건 3)

[0118] IF ($sw_0 \text{ AND } sw_3$) THEN “Strong filter” ELSE “Weak filter”

[0119] 상술한 조건들은 4개의 샘플 에지 세그먼트에 대한 강/약 결정을 정의한다. 따라서, 단일의 강/약 결정은 에지를 따르는 4 개의 로우들 또는 4 개의 컬럼들의 세트에 적용된다. 이전에 나타낸 바와 같이, 4 개의 로우들 또는 4 개의 컬럼들 중 첫번째 및 마지막 로우 또는 컬럼만이 결정에 포함된다. 포함된 로우들 및 컬럼들에 대하여, 드래프트는 p_3 및 q_3 이 조건 2 에서 고려될 것을 요구한다. 그러나, 프로세스 (600) 에서, p_3 은 라인 버퍼들에 존재하지 않는다. 따라서, p_3 은 조건 2 에서 p_2 로 대체된다. 추가로, 대칭성을 유지하기 위하여 q_3 는 q_2 로 대체된다. 따라서, 조건 2 는 다음과 같이 된다:

...

and $(|p_{2,i} - p_{0,i}| + |q_{0,i} - q_{2,i}|) < (\beta / 8)$

...

[0120] 이 변화에서, 강/약 결정은 더 이상 p_3 또는 q_3 에 의존하지 않고 따라서 디블로킹 필터는 엘리먼트 (606) 에서 p_0 , p_2 및 q_0 , q_2 만을 판독한다. 이전에 설명된 바와 같이, p 사이드 상의 샘플들은 라인 버퍼로부터 판독되고 q 사이드 상의 샘플들은 이전 코딩 스테이지로부터 직접 획득된다. 엘리먼트 (608) 에서, 디블로킹 필터는 위에서 변경된 바와 같이, 드래프트에서 특정된 조건들에 따라 강 또는 약 필터링을 선택한다.

[0122] 강 필터링이 선택되면, 디블로킹 필터는 엘리먼트 (610) 에서, 라인 버퍼로부터 필수적인 p 사이드 입력들을 판독하고 이전의 코딩 스테이지로부터 필수적인 q 사이드 입력들을 획득함으로써, 필터링된 샘플 값을 p_0' , p_1' , p_2' 및 q_0' , q_1' , q_2' 을 계산하도록 준비한다. 전체적으로, 필수적인 입력들은 단축된 입력들 (p_{short} 및

q_{short})에 더하여, p_0 , p_1 , p_2 및 q_0 , q_1 , q_2 를 포함한다. 입력 (p_{short})은 엘리먼트 (602)에서 이전에 계산되었고, 라인 버퍼들로부터 판독될 수 있다. 그러나, 입력 (q_{short})은 그 때 계산되지 않았는데 이는 q 사이드 샘플들이 이 예에서는 라인 버퍼들에 저장되지 않기 때문이다. 따라서, 엘리먼트 (612)에서, 디블로킹 필터는 이전 코딩 스테이지로부터 획득된 q 사이드 샘플들의 일부 또는 전부를 이용하여 q_{short} 를 계산한다. q_{short} 를 계산하는데 이용되는 공식은, p 사이드 샘플들이 q 사이드 샘플들로 대체되는 것을 제외하고는, p_{short} 를 계산하는데 이용된 공식과 동일하다. 따라서, p_{short} 에 대하여 위에 제공된 예시적인 공식들에 기초하여, q_{short} 공식의 예들은 다음을 포함한다:

$$[0123] q_{short} = Clip_{a-b}(q_3 - 2*q_2 + q_1)$$

[0124] 또는

$$[0125] q_{short} = Clip_{a-b}(2*q_3 - 5*q_2 + q_1 + 2*q_0).$$

[0126] 이전에 도시된 바와 같이, p_2' 및 q_2' 에 대한 강 필터링 계산들은 샘플들 (p_3 및 q_3)에 의존한다. 그러나, 프로세스 (600)에서, p_3 및 q_3 는 p_{short} 및 q_{short} 로 대체된다. 따라서, 일 실시형태에서, 다음의 공식이 p_2' 및 q_2' 에 이용된다.

$$[0127] p_2' = p_2 + Clip_{(-2*tc)-(2*tc)}((2*p_{short} - p_2 - p_1 + q_0 + 4) >> 3),$$

[0128] 이는 다음과 같이 확장될 수도 있다.

$$[0129] p_2' = p_2 + Clip_{(-2*tc)-(2*tc)}((2 * Clip_{a-b}(p_3 - 2*p_2 + p_1) - p_2 - p_1 + q_0 + 4) >> 3),$$

[0130] 그리고

$$[0131] q_2' = q_2 + Clip_{(-2*tc)-(2*tc)}((2 * q_{short} - q_2 - q_1 + p_0 + 4) >> 3),$$

[0132] 이는 다음과 같이 확장될 수도 있다,

$$[0133] q_2' = q_2 + Clip_{(-2*tc)-(2*tc)}((2 * Clip_{a-b}(q_3 - 2*q_2 + q_1) - q_2 - q_1 + p_0 + 4) >> 3).$$

[0134] 도 5에 대하여 이전에 설명된 바와 같이, 제한된 비트 심도를 가진 필터링 입력들, 이를 테면, p_{short} 및 q_{short} 의 이용은 디블로킹 필터의 성능에 악영향을 줄 수 있다. 특히, p_{short} 및 q_{short} 를 계산하는데 이용되는 Clip 함수는 디블로킹 필터의 필터링 강도를 제한할 수도 있다. 그러나, Clip 함수는 이것이 p_{short} 및 q_{short} 의 비트 심도를 감소시키기 때문에 유용하다. 실제로, 디블로킹 필터의 강도에 대한 제한은 Clip 함수가 제공하는 비트 심도 감소에 비례한다. 따라서, 원하는 범위 (a-b)가 공간 효율성 및 필터링 강도 간에 밸런싱을 가하도록 선택될 수도 있기 때문에, 어떠한 특정 제약 범위도 p_{short} 및 q_{short} 를 계산하는데 이용되는 Clip 함수에 대하여 기술되지 않는다.

[0135] 그러나, 드래프트에 달리 따르지 않는 디블로킹 필터에서는, p_{short} 및 q_{short} 에 대하여 위에 제공된 공식에 따른 적절한 클리핑이 필터링 강도에 악영향을 주지 않을 것이다. 이 유리한 결과는 드래프트가 강 필터링을 요구하는 조건들로부터 따른다. 구체적으로, 디블로킹 필터는,

[0136] $d < \beta / 4$ 이면 강 필터링만을 이용하며, 이는,

[0137] $|p_2 - 2*p_1 + p_0| < \beta / 8$ 을 의미하며, 이는,

[0138] $|p_2 - 2*p_1 + p_0| < 8$ 을 의미한다.

[0139] 용어 $|p_2 - 2*p_1 + p_0|$ 는 p_1 에서 센터링된 라플라시안에 대응한다. 라플라시안은 샘플들 간의 트랜지션들의 급격화도를 측정한다. 따라서, 강 필터링은 p_1 의 근방에서의 샘플들 사이의 트랜지션들이 매우 급격하지

않는 경우에만 이용된다. p_1 주변의 급격한 트랜지션들의 부재시, p_2 에 적용될 필터링 양은 일반적으로 클립프된 입력들 (p_{short} 및 q_{short})의 이용에 의해 부여되는 제한에도 불구하고 이것이 수용될 수 있기에 충분히 낮아야 한다.

[0140] p_{short} 및 q_{short} 를 포함한 강 필터링 공식은 엘리먼트 (614 및 616)에서 계산된다. 먼저, 엘리먼트 (614)에서, 디블로킹 필터는 조정 값들을 계산한다. 본 에에서, 조정 값들은 강 필터링 공식에서 $Clip_{(-2*tc)-(2*tc)}(\dots)$ 로 표현된다. 엘리먼트 (616)에서, 조정 값들은 샘플들 (p_0 , p_1 , p_2 및 q_0 , q_1 , q_2)에 추가되어 필터링된 샘플들 (p'_0 , p'_1 , p'_2 및 q'_0 , q'_1 , q'_2)을 생성한다. 필터링된 샘플들은 프로세스 (600)의 최종 출력이다.

[0141] 엘리먼트 (608)에서 강 필터링 대신에 약 필터링이 선택되면, 디블로킹 필터는, 드래프트에서 정의된 약 필터링 공식에 따라 조정 값들 및 필터링된 샘플들을 계산한다. 엘리먼트 (620)에서, 디블로킹 필터는 라인 베퍼들 및 이전 코딩 스테이지로부터의 샘플들을 판독함으로써 약 필터 공식을 계산하도록 준비될 것이다. 약 필터링에서 입력들로서 변경 및 이용되는 샘플들의 수는 변할 수도 있지만 드래프트의 약 필터링 공식은 p_3 또는 q_3 에 결코 의존하지 않는다. 약 필터링 공식은 많아야 샘플들 (p_0 , p_1 , p_2 및 q_0 , q_1 , q_2)에 의존할 것이다. 이에 따라, 공식은 프로세스 (600)에서 변경없이 이용될 수 있다.

[0142] 약 필터링에 필수적인 입력 샘플들이 엘리먼트 (620)에서 획득된 후, 디블로킹 필터는 엘리먼트 (622)에서, 최대 4 개의 샘플들, 즉, p_0 , p_1 및 q_0 , q_1 에 대하여 조정 값들을 계산한다. 그후, 엘리먼트 (624)에서, 디블로킹 필터는 오리지널 샘플들에 조정 값들을 추가하여 최대 4 개의 필터링된 샘플들, 즉 p'_0 , p'_1 및 q'_0 , q'_1 을 생성한다. 필터링된 샘플들은 프로세스 (600)의 최종 출력이다.

[0143] 도 6은 일련의 별도의 단계들에서의 필터링 프로세스를 나타내며, 이를 단계들 중 일부 또는 전부는 일부 실시 형태들에서 동시에 발생하거나 결합될 수도 있다.

[0144] 도 7a는 인접하는 비디오 블록들에서의 샘플들, 및 샘플들의 일부를 저장하는데 이용될 수도 있는 라인 베퍼들을 나타내는 개념도이다. 도 7a는 디블로킹 필터에 대한 가능한 저장 요건들의 일 예를 제공한다. 에지 (706)를 가로지르는 필터링에 대한 준비시, 디블로킹 필터는 블록 (702)으로부터의 샘플들을 라인 베퍼들에 저장한다. 블록 (704)으로부터의 샘플들은 도 6의 프로세스 (600)에 대하여 이전에 설명된 바와 같이, 필터링 프로세스 동안에 다른 소스들로부터 획득될 수 있기 때문에, 이 예에서는 블록 (704)으로부터의 샘플들이 라인 베퍼들에 저장되지 않는다.

[0145] 위에 설명된 바와 같이, 강/약 필터링 결정 및 강 필터링 계산의 일부 실시형태들은 필터링될 각각의 로우에 있어서, 에지의 각각의 사이드 상의 4 개의 샘플들을 이용한다. 블록 (702)에서, 4 개의 로우들이 존재하며, 로우들 각각은 4 개의 샘플들을 갖는다. 따라서, 디블로킹 필터는 라인 베퍼들 (710)에 블록 (702)으로부터의 샘플들 모두를 저장한다. 이에 따라, 라인 베퍼들 (710)은 32 개의 샘플들을 저장할 수 있다. 각각의 샘플이 예를 들어, 8 비트들의 비트 심도를 갖는다면, 라인 베퍼들의 총 용량은 적어도 256 비트들일 필요가 있다.

[0146] 도 7b는 인접하는 비디오 블록들에서의 샘플들, 및 비트 심도가 감소된 라인 베퍼들을 나타내는 개념도이다. 도 7b는 공간 절약 변경들과 함께 디블로킹 필터에 대한 가능한 저장 요건들의 일 예를 제공한다. 변경들은 도 6의 프로세스 (600)에 대하여 위에 설명된 본 개시물의 양태들에 따라 설명된다. 변경들의 결과로서, 디블로킹 필터는 각각의 로우의 단부에서 샘플들 (p_3 및 q_3)을 이용함이 없이 동작하도록 구성된다. 그 대신에, 필터는 예를 들어, p_3 , p_2 , p_1 및 q_3 , q_2 , q_1 의 클리프된 결합들로부터 각각 결정되는 p_{short} 및 q_{short} 에 의존한다. 도 7b에 나타낸 바와 같이, 입력 (p_{short})은 모듈 (720)에 의해 각각의 로우에 대하여 계산되고 샘플들 (p_2 , p_1 , p_0)과 함께 라인 베퍼들 (724)에 저장된다. 입력 (p_{short})은 샘플들 각각보다 더 낮은 비트 심도를 갖는다. 따라서, p_{short} 를 저장하는데 이용되는 공간 (726)(또는 메모리 사이즈)은 샘플을 저장하는데 이용되는 공간 (728)(또는 메모리 사이즈) 보다 더 작다. 그 결과, 라인 베퍼들 (724)의 총 사이즈는 감소될 수도 있고, 이에 의해 디블로킹 필터로 비용을 감소시키고 코딩 디바이스의 성능 및/또는 효율을 증가시킨다.

[0147]

예에 따라, 본원에 설명된 기법들의 소정의 작용들 또는 이벤트들은 상이한 시퀀스로 수행될 수 있으며, 부가, 병합, 또는 모두 배제될 수도 있다 (예를 들어, 반드시 모든 설명된 작용들 또는 이벤트들이 기법들의 실시를 위해 필요한 것은 아니다). 또한, 소정의 예들에서, 작용들 및 이벤트들은, 순차적으로 수행되는 대신에, 예를 들어, 멀티 스레드 프로세싱, 인터럽트 프로세싱, 또는 멀티 프로세서들을 통해 동시에 수행될 수도 있다.

[0148]

하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독 가능 매체 상에 저장되거나 송신될 수도 있고, 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독 가능 매체들은, 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터 판독 가능 저장 매체, 또는 예컨대 통신 프로토콜에 따라 한 장소에서 다른 장소로 컴퓨터 프로그램의 전달을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독 가능 매체들은 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독 가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독 가능 매체를 포함할 수도 있다.

[0149]

비제한적인 예로서, 이러한 컴퓨터 판독 가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 이송 또는 저장하기 위해 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속은 컴퓨터 판독 가능한 매체라고 적절히 지칭된다. 예를 들어, 소프트웨어가 동축 케이블, 광섬유 케이블, 연선, 디지털 가입자 회선 (digital subscriber line; DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신되는 경우, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 그러나, 컴퓨터 판독 가능 저장 매체 및 데이터 저장 매체가 접속들, 반송파들, 신호들, 또는 다른 송신 매체를 포함하지 않지만, 대신 비일시적인 유형의 저장 매체에 관련됨이 이해되어야만 한다. 본원에서 사용된 디스크(disk)와 디스크(disc)는, 컴팩트 디스크(CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크(DVD), 플로피디스크 및 블루레이 디스크를 포함하는데, 여기서 디스크(disk)는 통상 자기적으로 데이터를 재생하고, 디스크(disc)는 레이저를 이용하여 광학적으로 데이터를 재생한다. 상기의 조합들도 컴퓨터 판독 가능한 매체의 범위 내에 또한 포함되어야만 한다.

[0150]

명령들은, 하나 이상의 디지털 신호 프로세서 (digital signal processor; DSP) 들, 범용 마이크로프로세서들, 주문형 집적 회로 (application specific integrated circuit; ASIC) 들, 필드 프로그래머블 로직 어레이 (field programmable logic array; FPGA) 들, 또는 다른 등가의 집적 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에서 사용된 용어 "프로세서"는 임의의 앞서 설명된 구조 또는 본원에서 설명된 기술들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 또한, 몇몇 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈 내에 제공되거나, 또는 통합 코덱에 통합될 수도 있다. 또한, 상기 기술들은 하나 이상의 회로들 또는 로직 소자들에서 완전히 구현될 수 있다.

[0151]

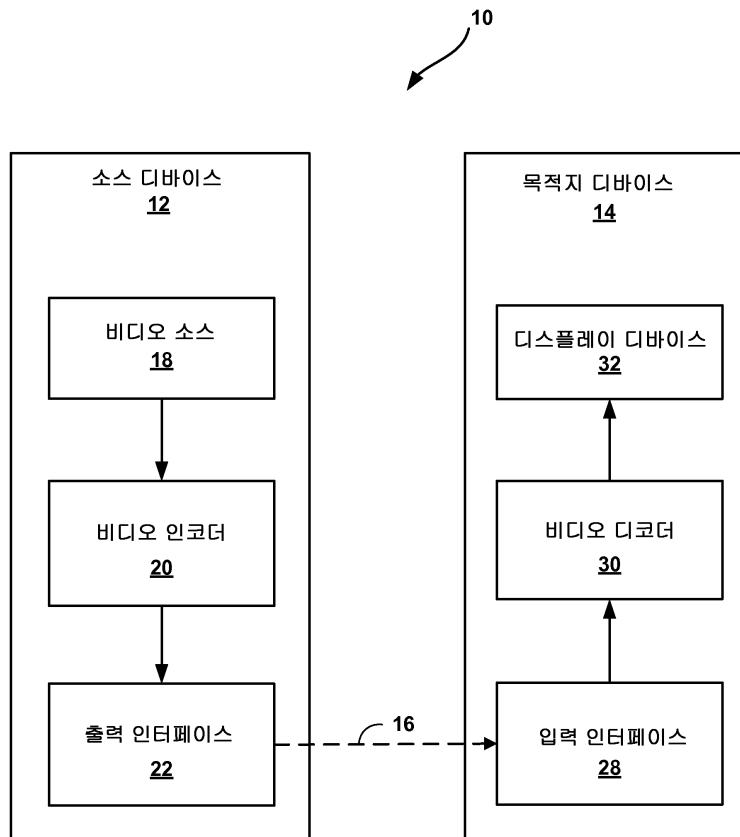
이 개시물의 기법들은 무선 핸드셋, 집적 회로 (integrated circuit; IC), 또는 IC 들의 세트 (예를 들어, 칩셋)를 포함하여, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양상들을 강조하기 위해 다양한 컴포넌트들, 모듈들, 또는 유닛들이 이 개시물에서 설명되나, 반드시 상이한 하드웨어 유닛들로 실현될 것을 요구하지는 않는다. 오히려, 상술된 바와 같이, 적합한 소프트웨어 및/또는 펌웨어와 결합하여, 상술된 하나 이상의 프로세서들을 포함해, 다양한 유닛들이 코덱 하드웨어 유닛으로 결합되거나 상호동작가능한 하드웨어 유닛들의 집합으로 제공될 수도 있다.

[0152]

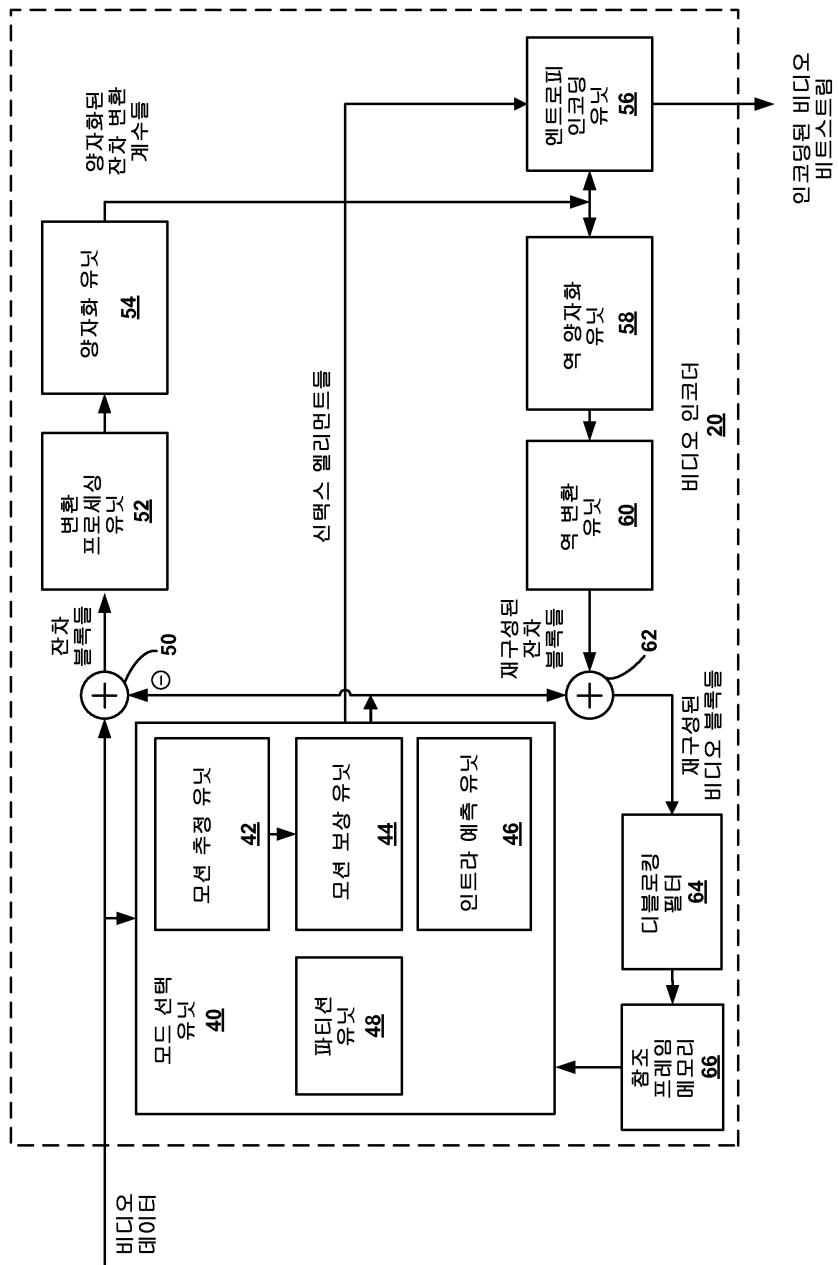
다양한 예들이 설명되었다. 이러한 예들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

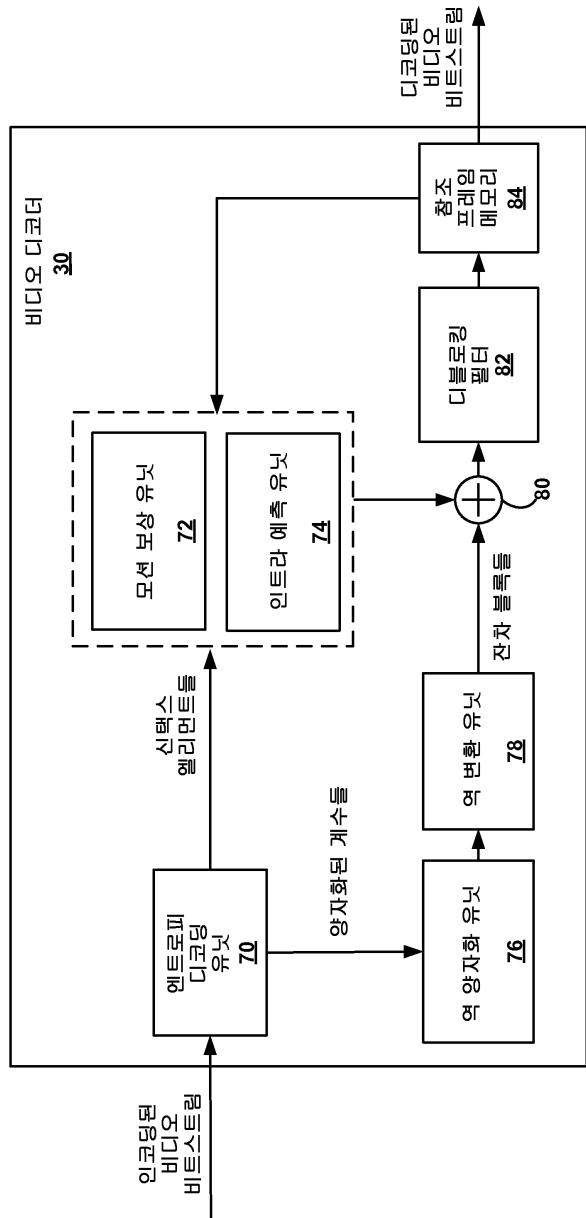
도면1



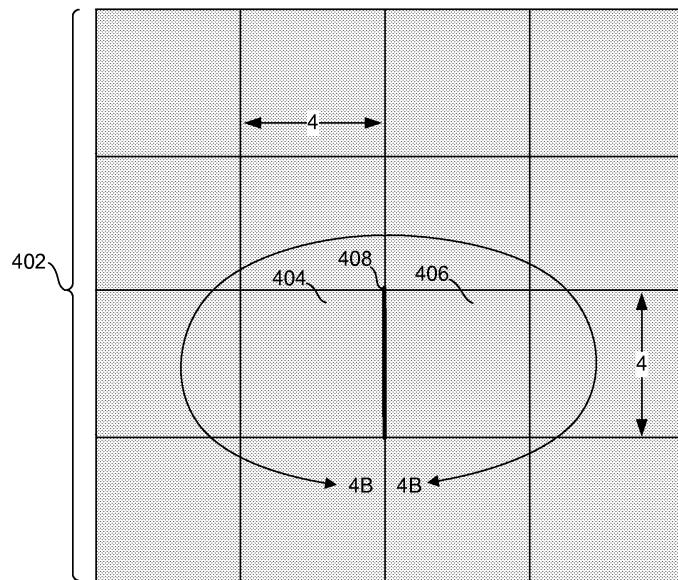
도면2



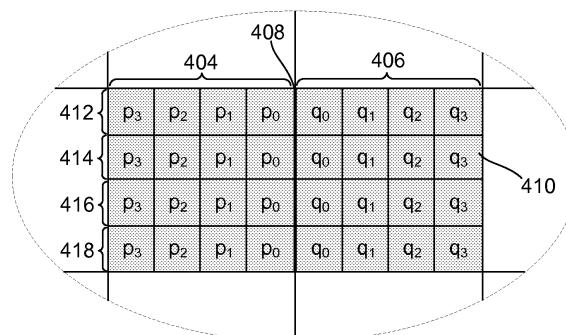
도면3



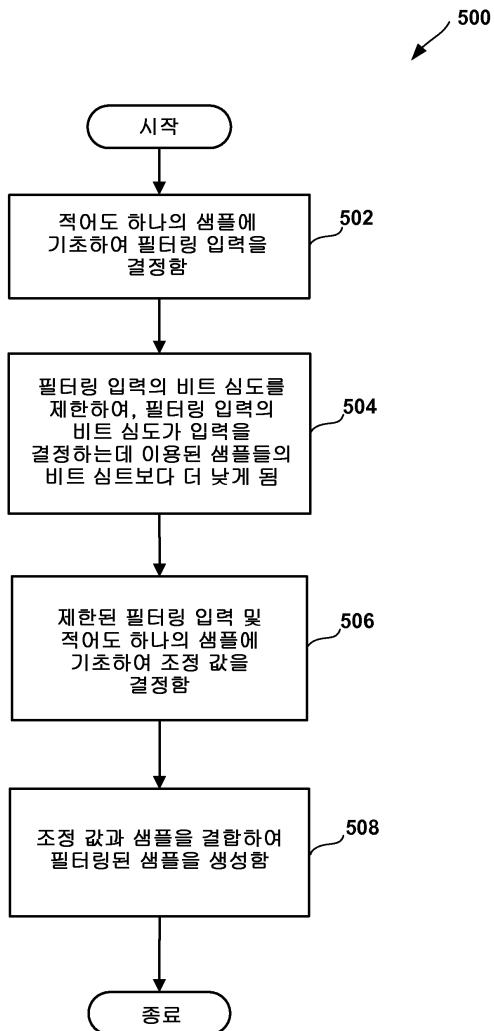
도면4a



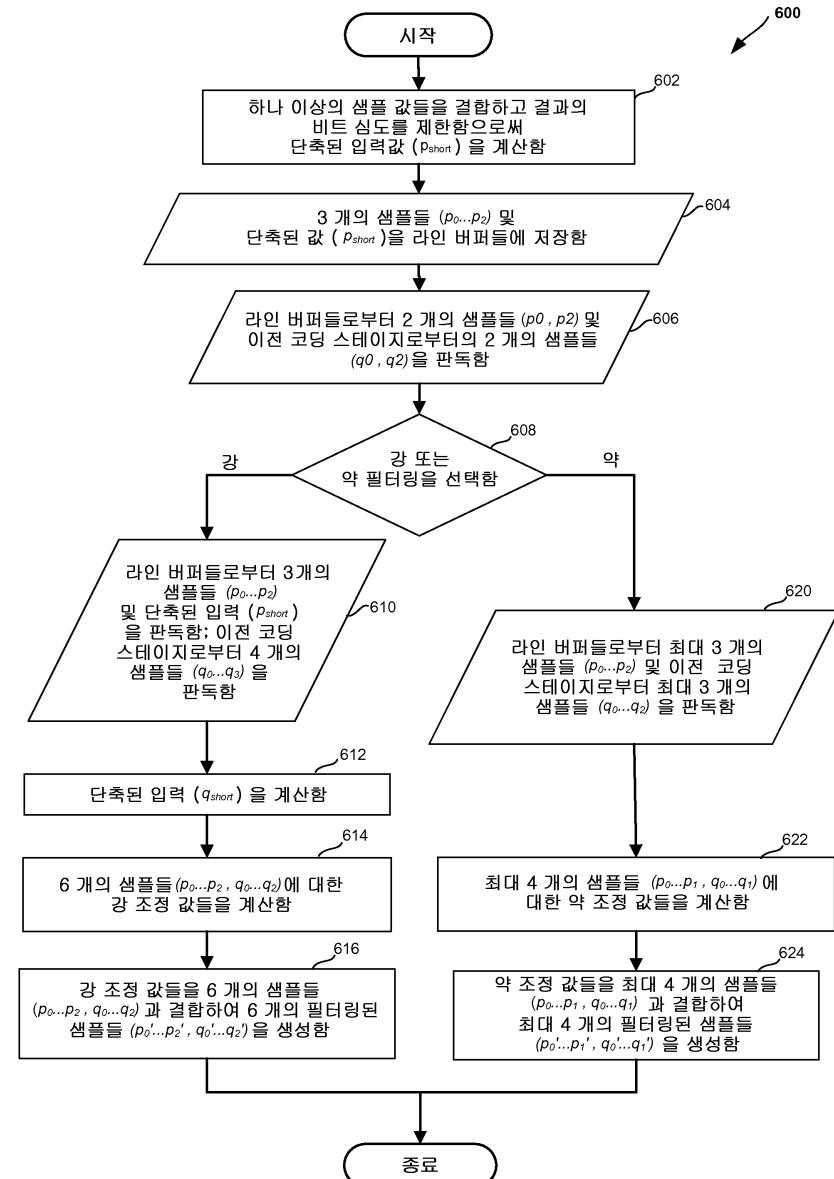
도면4b



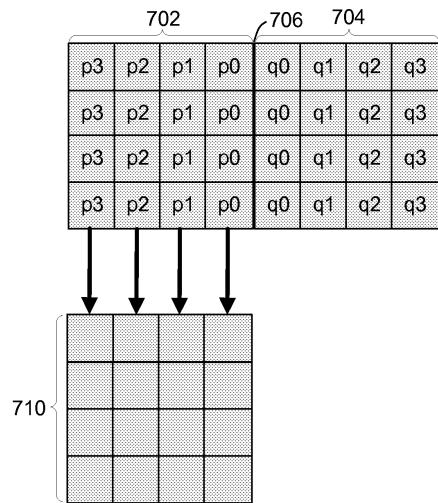
도면5



도면6



도면7a



도면7b

