



등록특허 10-2465720



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년11월10일
(11) 등록번호 10-2465720
(24) 등록일자 2022년11월07일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/117 (2014.01) *HO4N 19/10* (2014.01)
HO4N 19/105 (2014.01) *HO4N 19/159* (2014.01)
HO4N 19/174 (2014.01) *HO4N 19/176* (2014.01)
HO4N 19/187 (2014.01) *HO4N 19/196* (2014.01)
HO4N 19/463 (2014.01) *HO4N 19/61* (2014.01)
HO4N 19/80 (2014.01)
- (52) CPC특허분류
HO4N 19/117 (2015.01)
HO4N 19/10 (2015.01)
- (21) 출원번호 10-2018-7032182
- (22) 출원일자(국제) 2017년05월09일
심사청구일자 2020년04월23일
- (85) 번역문제출일자 2018년11월06일
- (65) 공개번호 10-2019-0006174
- (43) 공개일자 2019년01월17일
- (86) 국제출원번호 PCT/US2017/031767
- (87) 국제공개번호 WO 2017/196852
국제공개일자 2017년11월16일
- (30) 우선권주장
62/333,783 2016년05월09일 미국(US)
15/589,773 2017년05월08일 미국(US)
- (56) 선행기술조사문헌
WO2006108654 A2
US20140044161 A1

- (73) 특허권자
웰컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
카르체비츠 마르타
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
장 리
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
- 치엔 웨이-정
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
- (74) 대리인
특허법인코리아나

전체 청구항 수 : 총 28 항

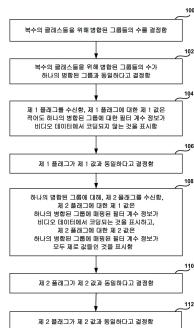
심사관 : 김영태

(54) 발명의 명칭 필터링 정보의 시그널링

(57) 요 약

비디오 디코더는 비디오 데이터의 비디오 블록들의 그룹에 대해, 복수의 클래스들을 위해 병합된 그룹들의 수가 하나의 병합된 그룹과 동일하다고 결정하고, 적어도 하나의 병합된 그룹에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하는 제 1 플래그를 수신하고, 하나의 병합된 그룹에 대해, 제 2 플래그를 수신하

(뒷면에 계속)

대 표 도 - 도 7

는 것으로서, 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 플래그를 수신하고; 제 2 플래그가 제 2 값과 동일하다고 결정하며; 그리고 모두 제로 값들을 사용하여 필터들의 세트로부터 하나 이상의 필터들을 결정하도록 구성된다.

(52) CPC특허분류

H04N 19/105 (2015.01)

H04N 19/159 (2015.01)

H04N 19/174 (2015.01)

H04N 19/176 (2015.01)

H04N 19/187 (2015.01)

H04N 19/196 (2015.01)

H04N 19/463 (2015.01)

H04N 19/61 (2015.01)

H04N 19/80 (2015.01)

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하기 위한 방법으로서,

상기 비디오 데이터의 비디오 블록들의 제 1 그룹에 대해, 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 단계로서, 상기 복수의 클래스들의 각각의 클래스는 활동 메트릭에 대한 값들의 범위 및 방향과 연관되고, 상기 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑되며, 동일한 필터 계수 정보에 매핑되는 하나 이상의 클래스들은 상기 병합된 그룹들의 동일한 병합된 그룹에 속하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 단계;

상기 복수의 클래스들을 위해 병합된 그룹들의 수가 1 과 동일하고, 그리고 상기 복수의 클래스들이 모두 상기 병합된 그룹들의 하나의 병합된 그룹에만 속한다고 결정하는 단계;

상기 비디오 데이터에서, 상기 비디오 블록들의 제 1 그룹에 대해, 제 1 플래그를 수신하는 단계로서, 상기 제 1 플래그에 대한 제 1 값은 상기 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 상기 비디오 데이터에서 코딩되지 않는 것을 표시하고, 상기 제 1 플래그에 대한 제 2 값은 상기 병합된 그룹들 모두에 대한 필터 계수 정보가 상기 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 플래그를 수신하는 단계;

상기 제 1 플래그가 상기 제 1 값과 동일한지 여부를 결정하는 단계;

상기 제 1 플래그가 상기 제 1 값과 동일하다고 결정하는 것에 응답하여, 상기 비디오 데이터에서, 상기 하나의 병합된 그룹에 대해, 제 2 플래그를 수신하는 단계로서, 상기 제 2 플래그에 대한 제 1 값은 상기 하나의 병합된 그룹에 매핑된 필터 계수 정보가 상기 비디오 데이터에서 코딩되는 것을 표시하고, 상기 제 2 플래그에 대한 제 2 값은 상기 하나의 병합된 그룹에 매핑된 상기 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 플래그를 수신하는 단계;

상기 제 2 플래그가 상기 제 1 값과 동일한지 또는 상기 제 2 값과 동일한지 여부를 결정하는 단계; 및

상기 모두 제로 값을 사용하여 상기 필터들의 세트로부터 하나 이상의 필터들을 결정하는 단계를 포함하고,

상기 모두 제로 값을 사용하여 상기 필터들의 세트로부터 하나 이상의 필터들을 결정하는 단계는:

상기 제 2 플래그가 상기 제 1 값과 동일하다고 결정하는 것에 응답하여, 차이값들을 하나 이상의 예측 필터들에 부가하는 단계로서, 상기 하나의 병합된 그룹에 매핑된 상기 필터 계수 정보는 상기 차이 값들을 포함하고, 상기 차이 값들은 모두 제로와 동일하고, 상기 필터들의 세트로부터의 상기 하나 이상의 필터들은 적어도 하나의 난제로 (non-zero) 계수를 갖는 적어도 하나의 필터를 포함하는, 상기 차이값들을 상기 하나 이상의 예측 필터들에 부가하는 단계; 및

상기 제 2 플래그가 상기 제 2 값과 동일하다고 결정하는 것에 응답하여, 상기 필터들의 세트로부터의 상기 하나 이상의 필터들은 모두 제로 필터 (all zero filter) 를 포함한다고 결정하는 단계로서, 상기 하나의 병합된 그룹에 매핑된 상기 필터 계수 정보는 필터 계수 값들을 포함하고, 상기 모두 제로 필터에 대한 모든 계수들은 제로와 동일한, 상기 필터들의 세트로부터의 상기 하나 이상의 필터들은 모두 제로 필터 (all zero filter) 를 포함한다고 결정하는 단계를 포함하는, 비디오 데이터를 디코딩하기 위한 방법.

청구항 2

삭제

청구항 3

삭제

청구항 4

제 1 항에 있어서,

상기 하나 이상의 예측 필터들은 하나 이상의 고정 필터들을 포함하는, 비디오 데이터를 디코딩하기 위한 방법.

청구항 5

제 1 항에 있어서,

상기 복수의 클래스들을 위해 병합된 그룹들의 수가 1 과 동일하다고 결정하는 단계는, 상기 비디오 데이터에서, 상기 필터들의 세트에서의 필터들의 총 수가 1 과 동일한 것을 표시하는 신택스를 수신하는 단계를 포함하는, 비디오 데이터를 디코딩하기 위한 방법.

청구항 6

제 1 항에 있어서,

상기 비디오 블록들의 제 1 그룹은 루마 비디오 블록들의 그룹을 포함하는, 비디오 데이터를 디코딩하기 위한 방법.

청구항 7

제 1 항에 있어서,

상기 비디오 블록들의 제 1 그룹은 슬라이스를 포함하는, 비디오 데이터를 디코딩하기 위한 방법.

청구항 8

삭제

청구항 9

제 1 항에 있어서,

비디오 블록들의 제 2 그룹에 대해 복수의 클래스들을 위해 병합된 그룹들의 수가 상기 비디오 블록들의 제 2 그룹에 대해 1 과 동일하다고 결정하는 단계;

상기 비디오 블록들의 제 2 그룹에 대해 상기 복수의 클래스들을 위해 병합된 그룹들의 수가, 상기 복수의 클래스들이 상기 비디오 블록들의 제 2 그룹에 대해 하나의 병합된 그룹을 포함하는 것을 표시하는 1 과 동일하다고 결정하는 것에 응답하여;

상기 비디오 데이터에서, 상기 비디오 블록들의 제 2 그룹에 대해, 상기 제 1 플래그의 제 2 인스턴스를 수신하는 단계;

상기 제 1 플래그의 상기 제 2 인스턴스가 상기 제 1 플래그에 대한 상기 제 2 값과 동일하다고 결정하는 단계;

상기 비디오 데이터에서 상기 제 2 플래그의 제 2 인스턴스를 수신하지 않으면서, 상기 제 2 플래그의 상기 제 2 인스턴스가 상기 제 2 플래그에 대한 상기 제 1 값과 동일하다고 추론하는 단계; 및

상기 비디오 데이터에서, 상기 비디오 블록들의 제 2 그룹에 대해 상기 하나의 병합된 그룹에 대한 필터 계수 정보를 수신하는 단계로서, 상기 비디오 블록들의 제 2 그룹에 대해 상기 하나의 병합된 그룹에 대한 상기 필터 계수 정보가 적어도 하나의 낸제로 계수를 포함하는, 상기 필터 계수 정보를 수신하는 단계를 더 포함하는, 비디오 데이터를 디코딩하기 위한 방법.

청구항 10

비디오 데이터를 인코딩하기 위한 방법으로서,

상기 비디오 데이터의 비디오 블록들의 제 1 그룹에 대해 필터들의 세트를 결정하는 단계;

상기 비디오 블록들의 제 1 그룹에 대해 필터 계수 정보의 세트들을 결정하는 단계;

복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 단계로서, 상기 복수의 클래스들의 각각의 클래스는 활동 메트릭에 대한 값들의 범위 및 방향과 연관되고, 상기 복수의 클래스들의 각각의 클래스는 상기 필터들의 세

트로부터의 필터에 매핑되며, 동일한 필터 계수 정보에 매핑되는 하나 이상의 클래스들은 상기 병합된 그룹들의 동일한 병합된 그룹에 속하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 단계;

상기 복수의 클래스들을 위해 병합된 그룹들의 수가 1 과 동일하고, 그리고 상기 복수의 클래스들 모두가 상기 병합된 그룹들의 하나의 병합된 그룹에만 속한다고 결정하는 단계;

상기 비디오 데이터에서의 포함을 위해, 제 1 값 또는 제 2 값으로 설정된 제 1 플래그를 생성하는 단계로서, 상기 제 1 플래그에 대한 상기 제 1 값은 상기 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 상기 비디오 데이터에서 코딩되지 않는 것을 표시하고, 상기 제 1 플래그에 대한 상기 제 2 값은 상기 병합된 그룹들 모두에 대한 필터 계수 정보가 상기 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 값 또는 상기 제 2 값으로 설정된 제 1 플래그를 생성하는 단계;

상기 제 1 값으로 설정된 상기 제 1 플래그를 생성하는 것에 응답하여, 상기 비디오 데이터에서의 포함을 위해, 제 1 값 또는 제 2 값으로 설정된 제 2 플래그를 생성하는 단계로서, 상기 제 2 플래그에 대한 상기 제 1 값은 상기 하나의 병합된 그룹에 매핑된 필터 계수 정보가 상기 비디오 데이터에서 코딩되는 것을 표시하고, 상기 제 2 플래그에 대한 상기 제 2 값은 상기 하나의 병합된 그룹에 매핑된 상기 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 값으로 설정된 제 2 플래그를 생성하는 단계를 포함하는, 비디오 데이터를 인코딩하기 위한 방법.

청구항 11

제 10 항에 있어서,

상기 필터 계수 정보의 세트들은 필터 계수 값을 포함하고, 상기 필터들의 세트는 모두 제로 필터를 포함하고, 상기 모두 제로 필터에 대한 모든 계수들은 제로와 동일한, 비디오 데이터를 인코딩하기 위한 방법.

청구항 12

제 10 항에 있어서,

상기 필터 계수 정보는 상기 필터들의 세트에서의 필터 계수들과 하나 이상의 예측 필터들의 필터 계수들 사이의 차이 값을 포함하는, 비디오 데이터를 인코딩하기 위한 방법.

청구항 13

제 12 항에 있어서,

상기 하나 이상의 예측 필터들은 하나 이상의 고정 필터들을 포함하는, 비디오 데이터를 인코딩하기 위한 방법.

청구항 14

제 10 항에 있어서,

상기 비디오 데이터에서의 포함을 위해, 상기 복수의 클래스들을 위해 병합된 그룹들의 수가 1 과 동일한 것을 표시하는, 선택지를 생성하는 단계를 더 포함하는, 비디오 데이터를 인코딩하기 위한 방법.

청구항 15

제 10 항에 있어서,

상기 비디오 블록들의 제 1 그룹은 루마 비디오 블록들의 그룹을 포함하는, 비디오 데이터를 인코딩하기 위한 방법.

청구항 16

제 10 항에 있어서,

상기 비디오 블록들의 제 1 그룹은 슬라이스를 포함하는, 비디오 데이터를 인코딩하기 위한 방법.

청구항 17

삭제

청구항 18

제 10 항에 있어서,

상기 비디오 블록들의 제 2 그룹에 대해 복수의 클래스들을 위해 병합된 그룹들의 수가 상기 비디오 블록들의 제 2 그룹에 대해 1 과 동일하다고 결정하는 단계;

상기 비디오 블록들의 제 2 그룹에 대해 상기 복수의 클래스들을 위해 병합된 그룹들의 수가, 상기 복수의 클래스들이 상기 비디오 블록들의 제 2 그룹에 대해 하나의 병합된 그룹을 포함하는 것을 표시하는, 1 과 동일하다고 결정하는 것에 응답하여, 상기 비디오 데이터에서의 포함을 위해, 상기 비디오 블록들의 제 2 그룹에 대해 상기 하나의 병합된 그룹에 대한 상기 비디오 데이터에 상기 제 2 플래그의 제 2 인스턴스를 포함하지 않으면서, 상기 비디오 블록들의 제 2 그룹에 대해, 상기 제 2 값으로 설정된 상기 제 1 플래그의 제 2 인스턴스를 생성하는 단계를 더 포함하는, 비디오 데이터를 인코딩하기 위한 방법.

청구항 19

비디오 데이터를 디코딩하기 위한 디바이스로서,

상기 비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하고;

상기 하나 이상의 프로세서들은,

상기 비디오 데이터의 비디오 블록들의 제 1 그룹에 대해, 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 것으로서, 상기 복수의 클래스들의 각각의 클래스는 활동 메트릭에 대한 값들의 범위 및 방향과 연관되고, 상기 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑되며, 동일한 필터 계수 정보에 매핑되는 하나 이상의 클래스들은 상기 병합된 그룹들의 동일한 병합된 그룹에 속하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하고;

상기 복수의 클래스들을 위해 병합된 그룹들의 수가 1 과 동일하고, 그리고 상기 복수의 클래스들이 모두 상기 병합된 그룹들의 하나의 병합된 그룹에만 속한다고 결정하고;

상기 비디오 데이터에서, 상기 비디오 블록들의 제 1 그룹에 대해, 제 1 플래그를 수신하는 것으로서, 상기 제 1 플래그에 대한 제 1 값은 상기 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 상기 비디오 데이터에서 코딩되지 않는 것을 표시하고, 상기 제 1 플래그에 대한 제 2 값은 상기 병합된 그룹들 모두에 대한 필터 계수 정보가 상기 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 플래그를 수신하고;

상기 제 1 플래그가 상기 제 1 값과 동일한지 여부를 결정하고;

상기 제 1 플래그가 상기 제 1 값과 동일하다고 결정하는 것에 응답하여, 상기 비디오 데이터에서, 상기 하나의 병합된 그룹에 대해, 제 2 플래그를 수신하는 것으로서, 상기 제 2 플래그에 대한 제 1 값은 상기 하나의 병합된 그룹에 매핑된 필터 계수 정보가 상기 비디오 데이터에서 코딩되는 것을 표시하고, 상기 제 2 플래그에 대한 제 2 값은 상기 하나의 병합된 그룹에 매핑된 상기 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 플래그를 수신하고;

상기 제 2 플래그가 상기 제 1 값과 동일한지 또는 상기 제 2 값과 동일한지 여부를 결정하고; 그리고

상기 모두 제로 값들을 사용하여 상기 필터들의 세트로부터 하나 이상의 필터들을 결정하도록 구성되고,

상기 모두 제로 값들을 사용하여 상기 필터들의 세트로부터 하나 이상의 필터들을 결정하기 위해, 상기 하나 이상의 프로세서들은:

상기 제 2 플래그가 상기 제 1 값과 동일하다고 결정하는 것에 응답하여, 차이값들을 하나 이상의 예측 필터들에 부가하는 것으로서, 상기 하나의 병합된 그룹에 매핑된 상기 필터 계수 정보는 상기 차이 값들을 포함하고, 상기 차이 값들은 모두 제로와 동일하고, 상기 필터들의 세트로부터의 상기 하나 이상의 필터들은 적어도 하나의 난제로 (non-zero) 계수를 갖는 적어도 하나의 필터를 포함하는, 상기 차이값들을 상기 하나 이상의 예측 필터들에 부가하고; 그리고

상기 제 2 플래그가 상기 제 2 값과 동일하다고 결정하는 것에 응답하여, 상기 필터들의 세트로부터의 상기 하

나 이상의 필터들은 모두 제로 필터 (all zero filter) 를 포함한다고 결정하는 것으로서, 상기 하나의 병합된 그룹에 매핑된 상기 필터 계수 정보는 필터 계수 값들을 포함하고, 상기 모두 제로 필터에 대한 모든 계수들은 제로와 동일한, 상기 필터들의 세트로부터의 상기 하나 이상의 필터들은 모두 제로 필터 (all zero filter) 를 포함한다고 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 20

삭제

청구항 21

삭제

청구항 22

제 19 항에 있어서,

상기 하나 이상의 예측 필터들은 하나 이상의 고정 필터들을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 23

제 19 항에 있어서,

상기 복수의 클래스들을 위해 병합된 그룹들의 수가 1 과 동일하다고 결정하기 위해, 상기 하나 이상의 프로세서들이 상기 비디오 데이터에서, 상기 필터들의 세트에서의 필터들의 총 수가 1 과 동일한 것을 표시하는 선택스를 수신하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 24

제 19 항에 있어서,

상기 비디오 블록들의 제 1 그룹은 루마 비디오 블록들의 그룹을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 25

제 19 항에 있어서,

상기 비디오 블록들의 제 1 그룹은 슬라이스를 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 26

삭제

청구항 27

제 19 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

비디오 블록들의 제 2 그룹에 대해 복수의 클래스들을 위해 병합된 그룹들의 수가 상기 비디오 블록들의 제 2 그룹에 대해 1 과 동일하다고 결정하고;

상기 비디오 블록들의 제 2 그룹에 대해 상기 복수의 클래스들을 위해 병합된 그룹들의 수가, 상기 복수의 클래스들이 상기 비디오 블록들의 제 2 그룹에 대해 하나의 병합된 그룹을 포함하는 것을 표시하는 1 과 동일하다고 결정하는 것에 응답하여, 상기 비디오 데이터에서, 상기 비디오 블록들의 제 2 그룹에 대해, 상기 제 1 플래그의 제 2 인스턴스를 수신하고;

상기 제 1 플래그의 상기 제 2 인스턴스가 상기 제 1 플래그에 대한 상기 제 2 값과 동일하다고 결정하고;

상기 비디오 데이터에서 상기 제 2 플래그의 제 2 인스턴스를 수신하지 않으면서, 상기 제 2 플래그의 상기 제 2 인스턴스가 상기 제 2 플래그에 대한 상기 제 1 값과 동일하다고 추론하며; 그리고

상기 비디오 데이터에서, 상기 비디오 블록들의 제 2 그룹에 대해 상기 하나의 병합된 그룹에 대한 필터 계수 정보를 수신하는 것으로서, 상기 비디오 블록들의 제 2 그룹에 대해 상기 하나의 병합된 그룹에 대한 상기 필터 계수 정보가 적어도 하나의 넌제로 계수를 포함하는, 상기 필터 계수 정보를 수신하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

청구항 28

비디오 데이터를 인코딩하기 위한 디바이스로서,

비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

상기 비디오 데이터의 비디오 블록들의 제 1 그룹에 대해 필터들의 세트를 결정하고;

상기 비디오 블록들의 제 1 그룹에 대해 필터 계수 정보의 세트들을 결정하고;

복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 것으로서, 상기 복수의 클래스들의 각각의 클래스는 활동 메트릭에 대한 값들의 범위 및 방향과 연관되고, 상기 복수의 클래스들의 각각의 클래스는 상기 필터들의 세트로부터의 필터에 매핑되며, 동일한 필터 계수 정보에 매핑되는 하나 이상의 클래스들은 상기 병합된 그룹들의 동일한 병합된 그룹에 속하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하고;

상기 복수의 클래스들을 위해 병합된 그룹들의 수가 1 과 동일하고, 그리고 상기 복수의 클래스들 모두가 상기 병합된 그룹들의 하나의 병합된 그룹에만 속한다고 결정하고;

상기 비디오 데이터에서의 포함을 위해, 제 1 값 또는 제 2 값으로 설정된 제 1 플래그를 생성하는 것으로서, 상기 제 1 플래그에 대한 상기 제 1 값은 상기 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 상기 비디오 데이터에서 코딩되지 않는 것을 표시하고, 상기 제 1 플래그에 대한 상기 제 2 값은 상기 병합된 그룹들 모두에 대한 필터 계수 정보가 상기 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 값 또는 제 2 값으로 설정된 제 1 플래그를 생성하고;

상기 제 1 값으로 설정된 상기 제 1 플래그를 생성하는 것에 응답하여, 상기 비디오 데이터에서의 포함을 위해, 제 1 값 또는 제 2 값으로 설정된 제 2 플래그를 생성하는 것으로서, 상기 제 2 플래그에 대한 상기 제 1 값은 상기 하나의 병합된 그룹에 매핑된 필터 계수 정보가 상기 비디오 데이터에서 코딩되는 것을 표시하고, 상기 제 2 플래그에 대한 상기 제 2 값은 상기 하나의 병합된 그룹에 매핑된 상기 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 값으로 설정된 제 2 플래그를 생성하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 29

제 28 항에 있어서,

상기 필터 계수 정보의 세트들은 필터 계수 값을 포함하고, 상기 필터들의 세트는 모두 제로 필터를 포함하고, 상기 모두 제로 필터에 대한 모든 계수들은 제로와 동일한, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 30

제 28 항에 있어서,

상기 필터 계수 정보는 상기 필터들의 세트에서의 필터들의 필터 계수들과 하나 이상의 예측 필터들의 필터 계수들 사이의 차이 값을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 31

제 30 항에 있어서,

상기 하나 이상의 예측 필터들은 하나 이상의 고정 필터들을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 32

제 28 항에 있어서,

상기 하나 이상의 프로세서들은 또한, 상기 비디오 데이터에서의 포함을 위해, 상기 복수의 클래스들을 위해 병합된 그룹들의 수가 1 과 동일한 것을 표시하는, 신택스를 생성하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 33

제 28 항에 있어서,

상기 비디오 블록들의 제 1 그룹은 루마 비디오 블록들의 그룹을 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 34

제 28 항에 있어서,

상기 비디오 블록들의 제 1 그룹은 슬라이스를 포함하는, 비디오 데이터를 인코딩하기 위한 디바이스.

청구항 35

삭제

청구항 36

제 28 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 비디오 블록들의 제 2 그룹에 대해 복수의 클래스들을 위해 병합된 그룹들의 수가 상기 비디오 블록들의 제 2 그룹에 대해 1 과 동일하다고 결정하고;

상기 비디오 블록들의 제 2 그룹에 대해 상기 복수의 클래스들을 위해 병합된 그룹들의 수가, 상기 복수의 클래스들이 상기 비디오 블록들의 제 2 그룹에 대해 하나의 병합된 그룹을 포함하는 것을 표시하는, 1 과 동일하다고 결정하는 것에 응답하여, 상기 비디오 데이터에서의 포함을 위해, 상기 비디오 블록들의 제 2 그룹에 대해 상기 하나의 병합된 그룹에 대한 상기 비디오 데이터에 상기 제 2 플래그의 제 2 인스턴스를 포함하지 않으면서, 상기 비디오 블록들의 제 2 그룹에 대해, 상기 제 2 값으로 설정된 상기 제 1 플래그의 제 2 인스턴스를 생성하도록 구성되는, 비디오 데이터를 인코딩하기 위한 디바이스.

발명의 설명**기술 분야**

[0001] 본 출원은 2016년 5월 9일 출원된 U.S. 가출원 제 62/333,783 호의 이익을 주장하며, 그 전체 내용은 참조로서 본 명세서에 통합된다.

[0002] 본 개시물은 비디오 코딩에 관한 것이다.

배경 기술

[0003] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 방송 시스템들, 무선 방송 시스템들, 개인용 디지털 보조기들 (personal digital assistant; PDA), 랩톱이나 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 전자책 리더들, 디지털 카메라들, 디지털 녹음 디바이스들, 디지털 미디어 재생기들, 비디오 게임 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화들, 이른바 "스마트 폰들", 비디오 원격화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 어드밴스트 비디오 코딩 (Advanced Video Coding; AVC), 최근에 완결된 고효율 비디오 코딩 (High Efficiency Video Coding; HEVC) 표준에 의해 정의된 표준들, 및 이러한 표준들의 확장들에서 설명된 바와 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한

비디오 압축 기법들을 구현함으로써, 디지털 비디오 정보를 보다 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0004] 비디오 압축 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 제거하기 위한 공간적 (인트라-픽처) 예측 및/또는 시간적 (인터-픽처) 예측을 수행한다. 블록-기반 비디오 코딩에 대해, 비디오 슬라이스 (예를 들어, 비디오 프레임 또는 비디오 프레임의 일부)는 비디오 블록들로 파티셔닝될 수도 있으며, 이는 또한 트리블록들, 코딩 유닛 (CU) 들, 및/또는 코딩 노드들로 지정될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 픽처에 있어서 이웃한 블록들에서의 참조 샘플들에 대해 공간 예측을 사용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 다른 픽처들에 있어서 참조 샘플들에 대해 시간 예측 또는 동일한 픽처에 있어서 이웃한 블록들에서의 참조 샘플들에 대해 공간 예측을 사용할 수도 있다. 픽처들은 프레임들로서 지정될 수도 있고, 참조 픽처들은 참조 프레임들로 지정될 수도 있다.

[0005] 공간 예측 또는 시간 예측은 코딩될 블록에 대한 예측 블록을 발생한다. 잔차 데이터는 코딩될 원래의 블록과 예측 블록 간의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터에 따라 인코딩되고, 잔차 데이터는 코딩된 블록과 예측 블록 사이의 차이를 표시한다. 인트라-코딩된 블록은 인트라 코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가의 압축을 위해, 잔차 데이터는 픽셀 도메인에서 변환 도메인으로 변환되어, 잔차 변환 계수들을 발생하고, 이 변환 계수들은, 그 후 양자화될 수도 있다. 초기에 2 차원 어레이로 배열된 양자화된 변환 계수들은, 변환 계수들의 1 차원 벡터를 생성하기 위해 스캐닝될 수도 있고, 엔트로피 코딩이 적용되어 훨씬 더 많은 압축을 달성할 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0006] 본 개시물은 비디오 데이터의 필터링에 관련된 기법들을 기재한다. 본 개시물의 기법들은, 예를 들어 적응적 루프 필터링 또는 다른 유형의 루프 필터링을 위해 사용될 수도 있다.

[0007] 일 예에 따라, 비디오 데이터를 디코딩하기 위한 방법은, 비디오 데이터의 비디오 블록들의 그룹에 대해, 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 단계로서, 복수의 클래스들의 각각의 클래스는 메트릭들에 대한 대응 값들을 갖고, 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑되며, 각각의 병합된 그룹은 동일한 필터 계수 정보에 매핑되는 하나 이상의 클래스들을 포함하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 단계; 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는, 1 과 동일하다고 결정하는 단계; 비디오 데이터에서, 비디오 블록들의 그룹에 대해, 제 1 플래그를 수신하는 단계로서, 제 1 플래그에 대한 제 1 값은 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하고, 제 1 플래그에 대한 제 2 값은 병합된 그룹들 모두에 대한 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 플래그를 수신하는 단계; 제 1 플래그가 제 1 값과 동일하다고 결정하는 단계; 비디오 데이터에서, 하나의 병합된 그룹에 대해, 제 2 플래그를 수신하는 단계로서, 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 플래그를 수신하는 단계; 제 2 플래그가 제 2 값과 동일하다고 결정하는 단계; 및 모두 제로 값들을 사용하여 필터들의 세트로부터 하나 이상의 필터들을 결정하는 단계를 포함한다.

[0008] 다른 예에 따라, 비디오 데이터를 인코딩하기 위한 방법은, 비디오 데이터의 비디오 블록들의 그룹에 대해 필터들의 세트를 결정하는 단계; 비디오 블록들의 그룹에 대해 필터 계수 정보의 세트를 결정하는 단계; 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 단계로서, 복수의 클래스들의 각각의 클래스는 메트릭들에 대한 대응 값들을 갖고, 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑되며, 각각의 병합된 그룹은 필터 계수 정보의 동일한 세트에 매핑되는 하나 이상의 클래스들을 포함하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 단계; 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는 1 과 동일하다고 결정하는 단계; 비디오 데이터에서의 포함을 위

해, 제 1 값으로 설정된 제 1 플래그를 생성하는 단계로서, 제 1 플래그에 대한 제 1 값은 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하고, 제 1 플래그에 대한 제 2 값은 병합된 그룹들 모두에 대한 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 값으로 설정된 제 1 플래그를 생성하는 단계; 비디오 데이터에서의 포함을 위해, 제 2 값으로 설정된 제 2 플래그를 생성하는 단계로서, 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 값으로 설정된 제 2 플래그를 생성하는 단계를 포함한다.

[0009] 발명의 다른 예에 따라, 비디오 데이터를 디코딩하기 위한 디바이스는, 비디오 데이터를 저장하도록 구성된 메모리 및 하나 이상의 프로세서들을 포함하고, 하나 이상의 프로세서들은, 비디오 데이터의 비디오 블록들의 그룹에 대해, 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 것으로서, 복수의 클래스들의 각각의 클래스는 메트릭들에 대한 대응 값들을 갖고, 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑되며, 각각의 병합된 그룹은 동일한 필터 계수 정보에 매핑되는 하나 이상의 클래스들을 포함하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하고; 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는, 1 과 동일하다고 결정하고; 비디오 데이터에서, 비디오 블록들의 그룹에 대해, 제 1 플래그를 수신하는 것으로서, 제 1 플래그에 대한 제 1 값은 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하고, 제 1 플래그에 대한 제 2 값은 병합된 그룹들 모두에 대한 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 플래그를 수신하고; 제 1 플래그가 제 1 값과 동일하다고 결정하고; 비디오 데이터에서, 하나의 병합된 그룹에 대해, 제 2 플래그를 수신하는 것으로서, 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 플래그를 수신하고; 제 2 플래그가 제 2 값과 동일하다고 결정하며; 그리고 모두 제로 값들을 사용하여 필터들의 세트로부터 하나 이상의 필터들을 결정하도록 구성된다.

[0010] 발명의 다른 예에 따라, 비디오 데이터를 인코딩하기 위한 디바이스는, 비디오 데이터를 저장하도록 구성된 메모리 및 하나 이상의 프로세서들을 포함하고, 하나 이상의 프로세서들은, 비디오 데이터의 비디오 블록들의 그룹에 대해 필터들의 세트를 결정하고; 비디오 블록들의 그룹에 대해 필터 계수 정보의 세트들을 결정하고; 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 것으로서, 복수의 클래스들의 각각의 클래스는 메트릭들에 대한 대응 값들을 갖고, 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑되며, 각각의 병합된 그룹은 필터 계수 정보의 동일한 세트에 매핑되는 하나 이상의 클래스들을 포함하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하고; 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는 1 과 동일하다고 결정하고; 비디오 데이터에서의 포함을 위해, 제 1 값으로 설정된 제 1 플래그를 생성하는 것으로서, 제 1 플래그에 대한 제 1 값은 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하고, 제 1 플래그에 대한 제 2 값은 병합된 그룹들 모두에 대한 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 값으로 설정된 제 1 플래그를 생성하고; 비디오 데이터에서의 포함을 위해, 제 2 값으로 설정된 제 2 플래그를 생성하는 것으로서, 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 값으로 설정된 제 2 플래그를 생성하도록 구성된다.

[0011] 다른 예에 따라, 비디오 데이터를 디코딩하기 위한 디바이스는, 비디오 데이터의 비디오 블록들의 그룹에 대해, 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 수단으로서, 복수의 클래스들의 각각의 클래스는 메트릭들에 대한 대응 값들을 갖고, 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑되며, 각각의 병합된 그룹은 동일한 필터 계수 정보에 매핑되는 하나 이상의 클래스들을 포함하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 수단; 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는, 1 과 동일하다고 결정하는 수단; 비디오 데이터에서, 비디오 블록들의 그룹에 대해, 제 1 플래그를 수신하는 수단으로서, 제 1 플래그에 대한 제 1 값은 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하고, 제 1 플래그에 대한 제 2 값은 병합된 그룹들 모두에 대한 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 플래그를 수신하는 수단; 제 1 플래그가 제 1 값과 동일하다고 결정하는 수단; 비디오 데이터에서, 하나의 병합된 그룹에 대해, 제 2 플래그를 수신하는 수단으로서, 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합

된 그룹에 매핑된 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 플래그를 수신하는 수단; 제 2 플래그가 제 2 값과 동일하다고 결정하는 수단; 및 모두 제로 값들을 사용하여 필터들의 세트로부터 하나 이상의 필터들을 결정하는 수단을 포함한다.

[0012] 다른 예에 따라, 비디오 데이터를 인코딩하기 위한 디바이스는, 비디오 데이터의 비디오 블록들의 그룹에 대해 필터들의 세트를 결정하는 수단; 비디오 블록들의 그룹에 대해 필터 계수 정보의 세트들을 결정하는 수단; 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 수단으로서, 복수의 클래스들의 각각의 클래스는 메트릭들에 대한 대응 값들을 갖고, 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑되며, 각각의 병합된 그룹은 필터 계수 정보의 동일한 세트에 매핑되는 하나 이상의 클래스들을 포함하는, 상기 복수의 클래스들을 위해 병합된 그룹들을 위해 병합된 그룹들의 수를 결정하는 수단; 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는 1 과 동일하다고 결정하는 수단; 비디오 데이터에서의 포함을 위해, 제 1 값으로 설정된 제 1 플래그를 생성하는 수단으로서, 제 1 플래그에 대한 제 1 값은 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하고, 제 1 플래그에 대한 제 2 값은 병합된 그룹들 모두에 대한 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하는 수단; 비디오 데이터에서의 포함을 위해, 제 2 값으로 설정된 제 2 플래그를 생성하는 수단으로서, 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 값으로 설정된 제 2 플래그를 생성하는 수단을 포함한다.

[0013] 다른 예에 따라, 컴퓨터 판독가능 저장 매체는 명령들을 저장하고, 명령들은 하나 이상의 프로세서들에 의해 실행될 때, 하나 이상의 프로세서들로 하여금, 비디오 데이터의 비디오 블록들의 그룹에 대해, 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하게 하는 것으로서, 복수의 클래스들의 각각의 클래스는 메트릭들에 대한 대응 값들을 갖고, 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑되며, 각각의 병합된 그룹은 동일한 필터 계수 정보에 매핑되는 하나 이상의 클래스들을 포함하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하게 하고; 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는, 1 과 동일하다고 결정하게 하고; 비디오 데이터에서, 비디오 블록들의 그룹에 대해, 제 1 플래그를 수신하게 하는 것으로서, 제 1 플래그에 대한 제 1 값은 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하고, 제 1 플래그에 대한 제 2 값은 병합된 그룹들 모두에 대한 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 플래그를 수신하게 하고; 제 1 플래그가 제 1 값과 동일하다고 결정하게 하고; 비디오 데이터에서, 하나의 병합된 그룹에 대해, 제 2 플래그를 수신하게 하는 것으로서, 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 플래그를 수신하게 하고; 제 2 플래그가 제 2 값과 동일하다고 결정하게 하며; 그리고 모두 제로 값들을 사용하여 필터들의 세트로부터 하나 이상의 필터들을 결정하게 한다.

[0014] 다른 예에 따라, 컴퓨터 판독가능 저장 매체는 명령들을 저장하고, 명령들은 하나 이상의 프로세서들에 의해 실행될 때, 하나 이상의 프로세서들로 하여금, 비디오 데이터의 비디오 블록들의 그룹에 대해 필터들의 세트를 결정하게 하고; 비디오 블록들의 그룹에 대해 필터 계수 정보의 세트들을 결정하게 하고; 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하게 하는 것으로서, 복수의 클래스들의 각각의 클래스는 메트릭들에 대한 대응 값들을 갖고, 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑되며, 각각의 병합된 그룹은 필터 계수 정보의 동일한 세트에 매핑되는 하나 이상의 클래스들을 포함하는, 상기 복수의 클래스들을 위해 병합된 그룹들의 수를 결정하게 하고; 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는 1 과 동일하다고 결정하게 하고; 비디오 데이터에서의 포함을 위해, 제 1 값으로 설정된 제 1 플래그를 생성하게 하는 것으로서, 제 1 플래그에 대한 제 1 값은 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하고, 제 1 플래그에 대한 제 2 값은 병합된 그룹들 모두에 대한 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하는, 상기 제 1 값으로 설정된 제 1 플래그를 생성하게 하고; 비디오 데이터에서의 포함을 위해, 제 2 값으로 설정된 제 2 플래그를 생성하게 하는 것으로서, 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 모두 제로 값들인 것을 표시하는, 상기 제 2 값으로 설정된 제 2 플래그를 생성하게 한다.

[0015] 개시물의 하나 이상의 양태들의 상세들은 첨부 도면들 및 하기의 설명에서 기술된다. 본 개시물에 기재된 기법들의 다른 피처들, 목적들, 및 이점들은 설명 및 도면들로부터, 그리고 청구항들로부터 명백해질 것이다.

도면의 간단한 설명

[0016] 도 1 은 본 개시물에 기재된 기법들을 활용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템을 도시하는 블록 다이어그램이다.

도 2 는 필터들에 대한 활동 메트릭 및 방향 메트릭에 대한 범위들의 맵핑을 도시하는 개념적 다이어그램이다.

도 3a 내지 도 3c 는 예시의 필터 형상들을 도시하는 개념적 다이어그램들이다.

도 4 는 필터 계수 시그널링을 위한 3 개의 카테고리들을 갖는 7x7 필터 형상의 일 예를 나타낸다.

도 5 는 본 개시물에 기재된 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 도시하는 블록 다이어그램이다.

도 6 은 본 개시물에 기재된 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 도시하는 블록 다이어그램이다.

도 7 은 본 개시물의 기법들에 따른 비디오 데이터를 디코딩하기 위한 기법을 구현하는 플로우 다이어그램이다.

도 8 은 본 개시물의 기법들에 따른 비디오 데이터를 인코딩하기 위한 기법들 도시하는 플로우 다이어그램이다.

도 9 는 본 개시물의 기법들에 따른 비디오 데이터를 디코딩하기 위한 기법을 도시하는 플로우 다이어그램이다.

발명을 실시하기 위한 구체적인 내용

[0017] 비디오 코딩은 통상적으로 동일한 픽처에서의 비디오 데이터의 이미 코딩된 블록으로부터 비디오 데이터의 블록을 예측하는 것 (즉, 인트라 예측) 또는 상이한 픽처에서 비디오 데이터의 이미 코딩된 블록으로부터 비디오 데이터의 블록을 예측하는 것을 수반한다. 일부 경우들에서, 비디오 인코더는 또한 예측 블록을 원래 블록과 비교함으로써 잔차 데이터를 계산한다. 따라서, 잔차 데이터는 예측 블록과 원래 블록 사이의 차이를 나타낸다. 비디오 인코더는 잔차 데이터를 변환하고 양자화하고 이 변환되고 양자화된 잔차 데이터를 인코딩된 비트스트림에서 시그널링한다. 비디오 디코더는 잔차 데이터를 예측 블록에 부가하여 예측 블록 단독 보다 더 근접하게 원래 비디오 블록과 매칭하는 재구성된 비디오 블록을 생성한다. 디코딩된 비디오의 품질을 더욱 개선하기 위해, 비디오 디코더는 재구성된 비디오 블록들 상에서 하나 이상의 필터링 동작들을 수행할 수 있다. 이러한 필터링 동작들의 예들은 디블록킹 필터링, 샘플 적응적 오프셋 (sample adaptive offset; SAO) 필터링, 및 적응적 루프 필터링 (adaptive loop filtering; ALF) 을 포함한다. 이를 필터링 동작들에 대한 파라미터들은 비디오 인코더에 의해 결정되고 인코딩된 비디오 비트스트림에서 명시적으로 시그널링될 수도 있거나, 파라미터들을 필요로 하지 않으면서 비디오 디코더에 의해 암시적으로 결정되어 인코딩된 비디오 비트스트림에서 명시적으로 시그널링될 수도 있다.

[0018] 본 개시물은 비디오 인코딩 및/또는 비디오 디코딩 프로세스들에서 재구성된 비디오 데이터를 필터링하는 것과 연관된 기법들을 기재하며, 특히 본 개시물은 ALF 와 관련된 기법들을 기재한다. 하지만, 기재된 기법들은 필터 파라미터들의 명시적 시그널링을 필요로 하는, 루프 필터링의 다른 유형들과 같은, 다른 필터링 스킴들에 또한 적용될 수도 있다. 본 개시물에 따라, 필터링은 인코더에 적용되고, 필터 정보는 비트스트림에서 인코딩되어 디코더가 인코더에서 적용되었던 필터링을 식별하는 것을 가능하게 한다. 비디오 인코더는 몇몇 상이한 필터링 시나리오들을 테스팅하고, 예를 들어 레이트 왜곡 분석에 기초하여, 재구성된 비디오 품질과 압축 품질 사이에서 원하는 트레이드오프를 생성하는 필터 또는 필터들의 세트를 선택할 수도 있다. 비디오 디코더는 필터 정보를 포함하거나 암시적으로 필터 정보를 도출하는 인코딩된 비디오 데이터를 수신하고, 그 비디오 데이터를 디코딩하며, 필터링 정보에 기초하여 필터링을 적용한다. 이러한 방식으로, 비디오 디코더는 비디오 인코더에서 적용되었던 동일한 필터링을 적용한다.

[0019] 본 개시물은 ALF 에 관련된, 특히 사이드 정보, 예를 들어 필터들을 송신하기 위한, 필터 파라미터들을 코딩하기 위한 기법들에 관련된다. 본 명세서에 기재된 기법들은 어드밴스드 비디오 코덱들, 예컨대 HEVC 의 확장들 또는 차세대 비디오 코딩 표준들과 함께 사용될 수도 있다.

[0020] 본 개시물에서 사용된 바와 같이, 용어 비디오 코딩은 일반적으로 비디오 인코딩 또는 비디오 디코딩 중 어느 하나를 지칭한다. 유사하게, 용어 비디오 코더는 비디오 인코더 또는 비디오 디코더를 지칭할 수도 있다.

또한, 비디오 디코딩에 관하여 본 개시물에 기재된 소정의 기법들은 또한 비디오 인코딩에 적용될 수도 있고, 그 역 또한 마찬가지이다. 예를 들어, 종종 비디오 인코더들 및 비디오 디코더들은 동일한 프로세스 또는 상반의 프로세스들을 수행하도록 구성된다. 또한, 비디오 인코더는 통상적으로 비디오 데이터를 인코딩하는 방법을 결정하는 프로세스들의 부분으로서 비디오 디코딩을 수행한다.

[0021] 도 1은 본 개시물에 기재된 기법들을 활용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템 (10)을 도시하는 블록 다이어그램이다. 도 1에 나타낸 바와 같이, 시스템 (10)은 목적지 디바이스 (14)에 의해 나중에 디코딩될 인코딩된 비디오 데이터를 생성하는 소스 디바이스 (12)를 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (14)는, 데스크톱 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 이른바 "스마트" 폰들과 같은 전화 핸드셋들, 이른바 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함하는 다양한 범위의 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14)는 무선 통신을 위해 구비될 수도 있다.

[0022] 목적지 디바이스 (14)는 링크 (16)를 통해 디코딩될 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16)는 인코딩된 비디오 데이터를 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 이동시킬 수 있는 임의의 유형의 매체 또는 디바이스를 포함할 수도 있다. 일 예에서, 링크 (16)는 소스 디바이스 (12)로 하여금 실시간으로 비디오 데이터를 목적지 디바이스 (14)로 직접적으로 송신하는 것을 가능하게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조되고, 목적지 디바이스 (14)로 송신될 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 임의의 무선 또는 유선 통신 매체, 예컨대 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 패킷 기반 네트워크, 예컨대 로컬 영역 네트워크, 광역 네트워크, 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.

[0023] 대안으로, 인코딩된 데이터는 출력 인터페이스 (22)로부터 저장 디바이스 (26)로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스 (26)로부터 액세스될 수도 있다. 저장 디바이스 (26)는 임의의 다양한 분산된 또는 국부적으로 액세스된 데이터 저장 매체들, 예컨대 하드 드라이브, 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적절한 디지털 저장 매체들을 포함할 수도 있다. 추가의 예에서, 저장 디바이스 (26)는 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오를 유지할 수도 있는 다른 중간 저장 디바이스 또는 파일 서버에 대응할 수도 있다. 목적지 디바이스 (14)는 스트리밍 또는 다운로드를 통해 저장 디바이스 (26)로부터의 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신할 수 있는 임의의 유형의 서버일 수도 있다. 예시의 파일 서버들은 (예를 들어 웹사이트용의) 웹서버, FTP 서버, NAS (network attached storage) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14)는 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 무선 채널 (예를 들어, 와이파이 접속), 유선 접속 (예를 들어, DSL, 케이블 모뎀 등), 또는 이들 양자의 조합을 포함할 수도 있다. 저장 디바이스 (26)로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양자의 조합일 수도 있다.

[0024] 본 개시물의 기법들은 반드시 무선 어플리케이션들 또는 설정들에 제한되지 않는다. 그 기법들은 다양한 멀티미디어 어플리케이션들, 예컨대 지상파 (over-the-air) 텔레비전 방송들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예를 들어 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 어플리케이션들 중 임의의 것의 지원에서의 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10)은, 비디오 스트리밍, 비디오 플레이백, 비디오 방송, 및/또는 비디오 텔레포니와 같은 어플리케이션들을 지원하기 위해 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0025] 도 1의 예에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20) 및 출력 인터페이스 (22)를 포함한다. 일부 경우들에서, 출력 인터페이스 (22)는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 소스 디바이스 (12)에서, 비디오 소스 (18)는 비디오 캡처 디바이스, 예를 들어 비디오 카메라, 이

전에 캡처된 비디오를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오피드 인터페이스, 및/또는 비디오 데이터로서 컴퓨터 그래픽스 데이터를 생성하기 위한 컴퓨터 그래픽스 시스템, 또는 그러한 소스들의 조합과 같은 소스를 포함할 수도 있다. 일 예로서, 비디오 소스 (18) 가 비디오 카메라인 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 이른바 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 하지만, 본 개시물에 기재된 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 어플리케이션들에 적용될 수도 있다.

[0026] 캡처된, 사전-캡처된, 또는 컴퓨터 생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (12) 의 출력 인터페이스 (22) 를 통해 목적지 디바이스 (14) 에 직접 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는 대안으로) 디코딩 및/또는 플레이백을 위해, 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 이후의 액세스를 위해 저장 디바이스 (26) 상에 저장될 수도 있다.

[0027] 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부 경우들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 링크 (16) 를 통해 인코딩된 비디오 데이터를 수신한다. 링크 (16) 를 통해 통신된 또는 저장 디바이스 상에 제공된 인코딩된 비디오 데이터는, 비디오 데이터를 디코딩하는데 있어서, 비디오 디코더 (30) 와 같은, 비디오 디코더에 의한 사용을 위해 비디오 인코더 (20) 에 의해 생성된 다양한 선택스 엘리먼트들을 포함할 수도 있다. 이러한 선택스 엘리먼트들은 통신 매체 상에서 송신된, 저장 매체 상에 저장된, 또는 필터 서버에 저장된 인코딩된 비디오 데이터로 포함될 수도 있다.

[0028] 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나 그 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함하고 또한 외부 디스플레이 디바이스와 인터페이스하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다.

일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 액정 디스플레이 (LCD), 페라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다.

[0029] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 최근에 완결된 고효율 비디오 코딩 (HEVC) 표준과 같은 비디오 압축 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HM) 을 따를 수도 있다. 대안으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 독점적인 또는 산업 표준들, 예컨대 대안으로 ISO/IEC MPEG-4, 파트 10, 어드밴스드 비디오 코딩 (AVC) 으로 지칭되는 ITU-T H.264 표준, 또는 그러한 표준들의 확장들, 예컨대 스케일러블 비디오 코딩 (Scalable Video Coding; SVC) 및 멀티 뷰 비디오 코딩 (Multi-view Video Coding; MVC) 확장들에 따라 동작할 수도 있다. 하지만, 본 개시물의 기법들은 임의의 특정 코딩 표준에 제한되지 않는다. 비디오 압축 표준들의 다른 예들은 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263 및 ISO/IEC MPEG-4 비주얼을 포함한다.

[0030] 본 개시물의 기법들은 설명을 용이하게 하기 위해 HEVC 용어를 활용할 수도 있다. 하지만, 본 개시물의 기법들은 HEVC 에 제한되지 않는 것이 가정되어야 하며, 실제로, 본 개시물의 기법들은 HEVC 및 그 확장들에 대한 후속 표준들에서 구현될 수도 있음이 명백히 고려된다.

[0031] 도 1 에 나타내지는 않았지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있으며, 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함하여, 공통 데이터 스트림 또는 별도의 데이터 스트림들에서 오디오 및 비디오 양자 모두의 인코딩을 핸들링할 수도 있다. 적용가능하다면, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP) 과 같은 다른 프로토콜들을 따를 수도 있다.

[0032] 비디오 인코더 (22) 및 비디오 디코더 (30) 각각은 임의의 다양한 적합한 인코더 회로, 예컨대 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 집적회로 (ASIC) 들, 필드 프로그램가능 게이트 어레이 (FPGA) 들, 이산 로직, 소프트웨어, 하드웨어, 펌웨어, 또는 이들의 임의의 조합들로서 구현될 수도 있다.

이 기법들이 부분적으로 소프트웨어로 구현되는 경우, 디바이스는 그 소프트웨어에 대한 명령들을 적합한 비일시적 컴퓨터 판독가능 매체에 저장할 수도 있고, 본 개시물의 기법들을 수행하기 위해 하나 이상의 프로세서들을 사용하는 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있고, 이를 중 어느 것은 결합된 인코더/디코더 (CODEC) 의 일부로서 각각의 디바이스에 통합될 수도 있다.

- [0033] 위에 도입된 바와 같이, JCT-VC 는 HEVC 표준의 개발을 완결하였다. HEVC 표준 노력들은 HEVC 테스트 모델 (HM) 로서 지칭되는 비디오 코딩 디바이스의 진화 모델에 기초하였다. HM 은 예를 들어, ITU-T H.264/AVC 에 따라, 기존의 디바이스들에 대한 비디오 코딩 디바이스들의 몇 가지 부가 능력들을 가정한다. 예를 들어, H.264 가 9 개의 인트라 예측 인코딩 모드를 제공하는 반면, HM 은 35 개의 인트라 예측 인코딩 모드를 제공할 수도 있다.
- [0034] ITU-T VCEG (Q6/16) 및 ISO/IEC MPEG (JTC 1/SC 29/WG 11) 는 현재 HEVC 표준의 압축 능력을 잠재적으로 초과하는 압축 능력을 갖는 장래의 비디오 코딩 기술의 표준화에 대한 잠재적인 필요성을 연구하고 있다 (현재의 확장들 및 화면 콘텐츠 코딩 및 높은 동적 범위 코딩을 위한 단기 확장들을 포함). 이 그룹들은 이 분야의 전문가들에 의해 제안한 압축 기술 설계들을 평가하기 위해 공동 비디오 탐구 팀 (Joint Video Exploration Team; JVET) 로서 알려진 공동 콜라보레이션 노력에서 이러한 탐구 활동에 대해 함께 작업하고 있다. JVET 는 처음에 2015 년 10 월 19 일 ~ 21 일 동안 만났다. 참조 소프트웨어의 버전인 공동 탐구 모델 2 (Joint Exploration Model 2; JEM 2) 는 다음으로부터 다운로드될 수 있다. https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-2.0/. JEM2 에 대한 알고리즘은 J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, J. Boyce, "Algorithm description of Joint Exploration Test Model 2" , JVET-B1001, San Diego, Mar. 2016 에 기재되어 있으며, 이러한 기재는 참조로서 본 명세서에 통합된다.
- [0035] HEVC 및 다른 비디오 코딩 사양들에서, 비디오 시퀀스는 통상적으로 일련의 픽처들을 포함한다. 픽처들은 또한, "프레임들" 로서 지칭될 수도 있다. 일 예의 접근법에 있어서, 픽처는 S_L , S_{Cb} , 및 S_{Cr} 로 표기된 3 개의 샘플 어레이들을 포함할 수도 있다. 이러한 예의 접근법에서, S_L 은 루마 샘플들의 2-차원 어레이 (즉, 블록) 이다. S_{Cb} 는 Cb 크로미넌스 샘플들의 2-차원 어레이이다. S_{Cr} 은 Cr 크로미넌스 샘플들의 2-차원 어레이이다. 크로미넌스 샘플들은 또한, 본원에서 "크로마" 샘플들로서 지칭될 수도 있다. 다른 경우들에서, 픽처는 모노크롬일 수도 있고 단지 루마 샘플들의 어레이만을 포함할 수도 있다.
- [0036] 픽처의 인코딩된 표현을 생성하기 위해, 비디오 인코더 (20) 는 코딩 트리 유닛 (CTU) 들의 세트를 생성할 수도 있다. CTU들 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2 개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용된 신팩스 구조들을 포함할 수도 있다. 모노크롬 픽처들 또는 3 개의 별도의 컬러 평면들을 갖는 픽처들에서, CTU 는 코딩 트리 블록의 샘플들을 코딩하는데 사용된 신팩스 구조들 및 단일 코딩 트리 블록을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 또한, "트리 블록" 또는 "최대 코딩 유닛" (LCU) 으로서 지칭될 수도 있다. HEVC 의 CTU들은 H.264/AVC 와 같은 다른 표준들의 매크로블록들과 광범위하게 유사할 수도 있다. 하지만, CTU 는 반드시 특정 사이즈에 제한되지 않고, 하나 이상의 코딩 유닛 (CU) 들을 포함할 수도 있다. 슬라이스는 레스터 스캔 오더로 연속적으로 오더링된 정수의 CTU들을 포함할 수도 있다.
- [0037] 코딩된 CTU 를 생성하기 위해, 비디오 인코더 (20) 는 CTU 의 코딩 트리 블록 (CTB) 들 상에서 큐드-트리 파티셔닝을 재귀적으로 수행하여 코딩 트리 블록들을 코딩 블록들, 이로써 명칭 "코딩 트리 유닛들" 로 분할할 수도 있다. 코딩 블록은 샘플들의 NxN 블록일 수도 있다. CU 는 루마 샘플 어레이, Cb 샘플 어레이, 및 Cr 샘플 어레이를 갖는 픽처의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 2 개의 대응하는 코딩 블록들, 및 코딩 블록들의 샘플들을 코딩하는데 사용된 신팩스 구조들을 포함할 수도 있다. 모노크롬 픽처들 또는 3 개의 별도의 컬러 평면들을 갖는 픽처들에서, CU 는 코딩 블록의 샘플들을 코딩하는데 사용된 신팩스 구조들 및 단일 코딩 블록을 포함할 수도 있다.
- [0038] 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은, 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비-정사각형) 블록이다. CU 의 예측 유닛 (PU) 은 루마 샘플들의 예측 블록, 크로마 샘플들의 2 개의 대응 예측 블록들, 및 예측 블록들을 예측하는데 사용된 신팩스 구조들을 포함할 수도 있다. 모노크롬 픽처들 또는 3 개의 별도의 컬러 평면들을 갖는 픽처들에서, PU 는 단일 예측 블록 및 그 예측 블록을 예측하는데 사용된 신팩스 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 의 루마, Cb, 및 Cr 예측 블록들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다.
- [0039] 비디오 인코더 (20) 는 인트라 예측 또는 인터 예측을 사용하여 PU 에 대한 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 인트라 예측을 사용하여 PU 의 예측 블록을 생성하는 경우, 비디오 인코더 (20) 는

PU 와 연관된 픽처의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위해 인터 예측을 사용하는 경우, 비디오 인코더 (20) 는 PU 와 연관된 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.

[0040] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 대한 예측 루마, Cb 및 Cr 예측 블록들을 생성한 후에, 비디오 인코더 (20) 는 CU 에 대한 루마 잔차 블록을 생성할 수도 있다. CU 의 루마 잔차 블록에서 각각의 샘플은, CU 의 예측 루마 블록들 중 하나에서의 루마 샘플과 CU 의 원래의 루마 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낸다. 또한, 비디오 인코더 (20) 는 CU 에 대한 Cb 잔차 블록을 생성할 수도 있다. CU 의 Cb 잔차 블록에서 각각의 샘플은, CU 의 예측 Cb 블록들 중 하나에서의 Cb 샘플과 CU 의 원래의 Cb 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낸다. 비디오 인코더 (20) 는 또한, CU 에 대한 Cr 잔차 블록을 생성할 수도 있다. CU 의 Cr 잔차 블록에서 각각의 샘플은, CU 의 예측 Cr 블록들 중 하나에서의 Cr 샘플과 CU 의 원래의 Cr 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낸다.

[0041] 또한, 비디오 인코더 (20) 는 퀘드-트리 파티셔닝을 사용하여, CU 의 루마, Cb, 및 Cr 잔차 블록들을 하나 이상의 루마, Cb, 및 Cr 변환 블록들로 분해할 수도 있다. 변환 블록은, 동일한 변환이 적용되는 샘플들의 직사각형 (예를 들어, 정사각형 또는 비정사각형) 블록일 수도 있다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2 개의 대응하는 변환 블록들, 및 변환 블록들을 변환하는데 사용된 신택스 구조들을 포함할 수도 있다. 따라서, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관된 루마 변환 블록은 CU 의 루마 잔차 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다. 모노크롬 픽처들 또는 3 개의 별도의 컬러 평면들을 갖는 픽처들에서, TU 는 변환 블록의 샘플들을 변환하는데 사용된 신택스 구조들 및 단일 변환 블록을 포함할 수도 있다.

[0042] 비디오 인코더 (20) 는 TU 의 루마 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 루마 계수 블록을 생성할 수도 있다. 계수 블록은 변환 계수들의 2-차원 어레이일 수도 있다. 변환 계수는 스칼라 양일 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cb 계수 블록을 생성하기 위해 TU 의 Cb 변환 블록에 하나 이상의 변환들을 적용할 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cr 계수 블록을 생성하기 위해 TU 의 Cr 변환 블록에 하나 이상의 변환들을 적용할 수도 있다.

[0043] 계수 블록 (예를 들어, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록) 을 생성한 후에, 비디오 인코더 (20) 는 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로, 계수들을 표현하기 위해 사용된 데이터의 양을 가능한 감소시키기 위해 변환 계수들이 양자화되어, 추가의 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후에, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 신택스 엘리먼트들 상에 콘텍스트-적응적 이진 산술 코딩 (Context-Adaptive Binary Coding; CABAC) 을 수행할 수도 있다.

[0044] 비디오 인코더 (20) 는 코딩된 픽처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 네트워크 추상 계층 (Network Abstraction Layer; NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛은, NAL 유닛에서의 데이터 유형의 표시를 포함하는 신택스 구조 및 에뮬레이션 방지 비트로 필요에 따라 배치된 로우 바이트 시퀀스 페이로드 (raw byte sequence payload; RBSP) 의 형태로 그 데이터를 포함하는 바이트들이다. NAL 유닛들의 각각은 NAL 유닛 헤더를 포함하고, RBSP 를 캡슐화한다. NAL 유닛 헤더는 NAL 유닛 유형 코드를 나타내는 신택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 지정된 NAL 유닛 유형 코드는 NAL 유닛의 유형을 나타낸다. RBSP 는 NAL 유닛 내에 캡슐화되는 정수의 바이트들을 포함하는 신택스 구조일 수도 있다. 일부 경우들에서, RBSP 는 제로 비트들을 포함한다.

[0045] NAL 유닛들의 상이한 유형들은 RBSP들의 상이한 유형들을 캡슐화할 수도 있다. 예를 들어, NAL 유닛의 제 1 유형은 PPS 에 대한 RBSP 를 캡슐화할 수도 있고, NAL 유닛의 제 2 유형은 코딩된 슬라이스에 대한 RBSP 를 캡슐화할 수도 있으며, NAL 유닛의 제 3 유형은 SEI 메시지에 대한 RBSP 를 캡슐화할 수도 있는 등이다. (SEI 메시지들 및 파라미터 세트들에 대한 RBSP들과는 반대인) 비디오 데이터에 대한 RBSP들을 캡슐화하는 NAL 유닛들은 VCL NAL 유닛들로서 지칭될 수도 있다.

[0046] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 또한, 비디오 디코더 (30) 는 비트스트림을 파싱 (parse) 하여 비트스트림으로부터 신택스 엘리먼트들을 획득할 수도 있다.

비디오 디코더 (30)는 비트스트림으로부터 획득된 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 재구성할 수도 있다. 비디오 데이터를 재구성하기 위한 프로세스는 일반적으로, 비디오 인코더 (20)에 의해 수행된 프로세스와 상반될 수도 있다. 또한, 비디오 디코더 (30)는 현재 CU의 TU들과 연관된 계수 블록들을 역 양자화할 수도 있다. 비디오 디코더 (30)는 변환 계수 블록들 상에서 역 변환 역 변환하여, 현재 CU의 TU들과 연관된 변환 블록들을 재구성할 수도 있다. 비디오 디코더 (30)는 현재 CU의 TU들의 변환 블록들의 대응 샘플들에 현재 CU의 PU들에 대한 예측 블록들의 샘플들을 가산하는 것에 의해 현재 CU의 코딩 블록들을 재구성할 수도 있다. 픽처의 각각의 CU에 대해 코딩 블록들을 재구성하는 것에 의해, 비디오 디코더 (30)는 픽처를 재구성할 수도 있다.

[0047] 비디오 코딩 분야에서, 디코딩된 비디오 신호의 품질을 강화하기 위해 필터링을 적용하는 것이 일반적이다. 필터는, 필터링된 프레임이 장래 프레임의 예측에 사용되지 않는 포스트-필터로서 적용되거나, 필터링된 프레임이 장래 프레임들을 예측하는데 사용되는 인-루프 필터로서 적용될 수 있다. 필터는 예를 들어, 원래 신호와 디코딩된 필터링된 신호 사이의 에러를 최소화함으로써 설계될 수 있다. 유사하게, 계수들을 변환하기 위해, 필터 $h(k, l)$ 의 계수들 ($k = -K, \dots, K, l = -K, \dots, K$)은 다음의 공식에 따라, 양자화되고,

$$f(k, l) = \text{round}(\text{normFactor} \cdot h(k, l))$$

[0048] 코딩되며, 디코더에 전송될 수도 있다. normFactor 는 예를 들어, 2^n 과 동일하게 설정될 수도 있다. 통상적으로 normFactor 의 값이 클수록 더 정확한 양자화를 유도하고, 양자화된 필터 계수들 $f(k, l)$ 은 통상적으로 더 우수한 성능을 제공한다. 다른 한편으로, normFactor 의 값이 클수록 또한 통상적으로 더 많은 비트를 송신하는 것을 필요로 하는 계수 $f(k, l)$ 를 산출한다.

[0049] 비디오 디코더 (30)에서, 디코더 필터 계수들 $f(k, l)$ 은 다음과 같이 재구성된 이미지 $R(i, j)$ 에 적용된다.

$$\tilde{R}(i, j) = \sum_{k=-K}^K \sum_{l=-K}^K f(k, l) R(i + k, j + l) / \sum_{k=-K}^K \sum_{l=-K}^K f(k, l), \quad (1)$$

[0050] 여기서, i 및 j 는 프레임 내의 픽셀들의 좌표이다.

[0051] JEM에 채용된 인-루프 적응적 루프 필터는 원래 J. Chen, Y. Chen, M. Karczewicz, X. Li, H. Liu, L. Zhang, X. Zhao, "Coding tools investigation for next generation video coding", SG16-Geneva-C806, Jan. 2015에서 제안되었으며, 그 기재는 본 명세서에 참조로서 통합된다. ALF는 HEVC에서 제안되었고, 다양한 작업 드래프트들 및 테스트 모델 소프트웨어, 즉 HEVC 테스트 모델(또는 "HM")에 포함되었지만, ALF는 HEVC의 최종 버전에 포함되지 않았다. 관련된 기술들 중에서, HEVC 테스트 모델 버전 HM-3.0에서의 ALF 설계가 가장 효율적인 설계로 주장되었다(T. Wiegand, B. Bross, W.J. Han, J.R. Ohm 및 G. J. Sullivan, "Working Draft 3 of High-Efficiency Video Coding" ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11의 JCT-VC (Joint Collaborative Team on Video Coding), JCTVC-E603, 5th Meeting: Geneva, CH, 16-23 March, 2011, 이하 "Working Draft 3"를 참조하며, 그 전체 내용은 참조로서 본 명세서에 통합된다). 따라서, HM-3.0으로부터의 ALF 설계가 본 명세서에서 도입된다.

[0052] HM-3.0에 포함된 ALF의 버전은 픽처 레벨 최적화에 기초한다. 즉, ALF 계수는 전체 프레임이 코딩된 후 도출된다. 블록 기반 적응(block based adaptation; BA) 및 영역 기반 적응(region based adaptation; RA)으로 지칭되는 루마 컴포넌트에 대한 2개의 모드가 있었다. 이 2개의 모드는 동일한 필터 형상들, 필터링 동작들 뿐만 아니라 동일한 신택스 엘리먼트들을 공유한다. BA와 RA간 하나의 차이는 분류 방법이며, 여기서 분류는 일반적으로 픽셀 또는 픽셀들의 블록에 적용하기 위해 필터들의 세트로부터 어느 것을 필터링할지를 결정하도록 픽셀 또는 픽셀들의 블록을 분류하는 것을 지칭한다.

[0053] 일 예의 접근법에서, BA의 분류는 블록 레벨이다. 루마 컴포넌트에 대해, 전체 픽처에서의 4x4 블록들은 1차원 (1D) 라플라시안(Laplacian) 방향(예를 들어, 3 방향들까지) 및 2 차원 (2D) 라플라시안 활동(예를 들어, 최대 5 활동 값들까지)에 기초하여 분류된다. 일 예의 접근법에서, 픽처에서의 각각의 4x4 블록은 1차원 (1D) 라플라시안 방향 및 2 차원 (2D) 라플라시안 활동에 기초한 그룹 인덱스를 할당받는다. 방향

Dir_b 및 비양자화 활동 Act_b의 일 예의 계산은 하기의 식 (2)-(5)에 나타나 있으며, 여기서 $\hat{I}_{i,j}$ 는 4x4 블록

의 상부 좌측 픽셀 위치에 대한 상대적 좌표 (i, j) 를 갖는 재구성된 픽셀을 표시하고, V_{ij} 및 H_{ij} 는 (i, j) 에 위치된 픽셀의 수직 및 수평 구배의 절대값들이다. 이와 같이, 방향 Dir_b 는 4×4 블록에서의 수직 구배 및 수평 구배의 절대값들을 비교함으로써 생성되고, Act_b 는 4×4 블록에서의 양 방향들에서의 구배의 합이다.

Act_b 는 위에 논의된 "WD3: Working Draft 3 of High-Efficiency Video Coding" 문헌에 기재된 바와 같이, 0 내지 4 에 포함된 범위로 추가로 양자화된다.

$$V_{i,j} = |\hat{l}_{i,j} \times 2 - \hat{l}_{i,j-1} - \hat{l}_{i,j+1}| \quad (2)$$

$$H_{i,j} = |\hat{l}_{i,j} \times 2 - \hat{l}_{i-1,j} - \hat{l}_{i+1,j}| \quad (3)$$

$$\text{Dir}_b = \begin{cases} 1, & \text{if } (\sum_{i=0}^3 \sum_{j=0}^3 H_{i,j} > 2 \times \sum_{i=0}^3 \sum_{j=0}^3 V_{i,j}) \\ 2, & \text{if } (\sum_{i=0}^3 \sum_{j=0}^3 V_{i,j} > 2 \times \sum_{i=0}^3 \sum_{j=0}^3 H_{i,j}) \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$[0056] \quad \text{Act}_b = \sum_{i=0}^3 \sum_{j=0}^3 (\sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} (V_{m,n} + H_{m,n})) \quad (5)$$

[0057] 따라서, 일 예의 접근법에서, 각각의 블록은 15 (5×3) 그룹들 (즉, 다음과 같은 클래스들)로부터 하나로 카테고리화될 수 있다. 인덱스는 블록의 Dir_b 및 Act_b 의 값에 따라 각각 4×4 블록을 할당받는다. 그룹 인덱스는 C 로 지칭하고 C 는 $5\text{Dir}_b + \hat{A}$ 와 동일하게 설정하며, 여기서 \hat{A} 는 Act_b 의 양자화된 값이다. 따라서, ALF 의 15 개의 세트들까지, 파라미터들이 픽처의 루마 컴포넌트에 대해 시그널링될 수 있다. 시그널링 비용을 절약하기 위해, 그룹들은 그룹 인덱스 값과 함께 병합될 수도 있다. 각각의 병합된 그룹에 대해, ALF 계수들의 세트가 시그널링된다.

[0058] 도 2 는 BA 분류를 위해 사용된 이러한 15 개의 그룹들 (또한 클래스들로서 지칭됨) 을 도시하는 개념적 다이어그램이다. 도 2 의 예에서, 필터들은 방향 메트릭 및 활동 메트릭에 대한 값들의 범위 (즉, 범위 0 부터 범위 4) 에 매핑된다. 도 2 에서의 방향 메트릭은 위의 식 4로부터 0, 1 및 2 의 값들에 대응할 수도 있는, 방향 없음, 수평, 및 수직의 값들을 갖는 것으로 나타나 있다. 도 2 의 특정 예는 15 개의 카테고리들에 매핑되는 것과 같은 6 개의 상이한 필터들 (즉, 필터 1, 필터 2, ..., 필터 6) 을 나타내지만, 더 많거나 더 적은 필터들이 유사하게 사용될 수도 있다. 도 2 는 그룹들 (222 내지 235) 로서 식별되는, 15 개의 그룹들로 나타나 있지만, 더 많거나 더 적은 그룹들이 또한 사용될 수도 있다. 예를 들어, 활동 메트릭에 대한 5 개의 범위들 대신, 더 많거나 더 적은 범위들이 사용되어 더 많은 그룹들을 야기할 수도 있다. 부가적으로, 단지 2 개의 방향들 대신, 부가 또는 대안의 방향들 (예를 들어, 45 도 방향 및 135 도 방향) 이 또한 사용될 수도 있다.

[0059] 하기에서 더 상세하게 설명될 바와 같이, 블록들의 각각의 그룹과 연관된 필터들은 하나 이상의 병합 플래그들을 사용하여 시그널링될 수도 있다. 1 차원 그룹 병합을 위해, 그룹이 이전 그룹과 동일한 필터에 매핑되는지를 표시하기 위해 단일 플래그가 전송될 수도 있다. 2 차원 병합을 위해, 그룹이 제 1 이웃한 블록 (예를 들어, 수평 또는 수직 이웃 중 하나) 와 동일한 필터에 매핑되는지를 표시하기 위해 제 1 플래그가 전송될 수도 있고, 그 플래그가 거짓이면, 그룹이 제 2 이웃한 블록 (예를 들어, 수평 이웃 또는 수직 이웃 중 다른 하나) 에 매핑되는 것을 표시하기 위해 제 2 플래그가 전송될 수도 있다.

[0060] 클래스들은 병합된 그룹들로 불리는 것으로 그룹화될 수도 있으며, 여기서 병합된 그룹에서의 각각의 클래스는 동일한 필터에 매핑된다. 일 예로서 도 2 를 참조하면, 그룹들 (221, 222, 및 223) 은 제 1 병합된 그룹으로 그룹화될 수도 있고; 그룹들 (224 및 225) 은 제 2 병합된 그룹으로 그룹화될 수도 있는 등이다. 통상적으로, 소정의 필터에 매핑된 모든 클래스들이 동일한 병합된 그룹에 있어야 하는 것은 아니지만, 병합된 그룹에서의 모든 클래스들은 동일한 필터에 매핑되어야 한다. 환언하면, 2 개의 병합된 그룹들은 동일한 필터에 매핑될 수도 있다.

[0061] 필터 계수들은 블록화 현상 (blockness) 을 감소시킬수 있고 및/또는 그렇지 않으면 다른 방식으로 비디오 품질을 개선할 수 있는 비디오 블록 필터링의 원하는 레벨들을 측정하기 위해 정의되거나 선택될 수도 있다. 필터 계수들의 세트는, 예를 들어 비디오 블록들의 예지들 또는 비디오 블록들 내의 다른 위치들을 따라 필터링이

어떻게 적용되는지를 정의할 수도 있다. 상이한 필터 계수들은 비디오 블록들의 상이한 픽셀들에 대해 필터링의 상이한 레벨들을 야기할 수도 있다. 필터링은 예를 들어, 원치 않는 아티팩트들을 소거하는 것을 돋기 위해 인접 픽셀 값들의 강도에서 차이를 평활화하거나 선명하게 할 수도 있다.

[0062] 본 개시물에서, 용어 "필터"는 일반적으로 필터 계수들의 세트를 지칭한다. 예를 들어, 3x3 필터는 9 개의 필터 계수들의 세트에 의해 정의될 수도 있고, 5x5 필터는 25 개의 필터 계수들의 세트에 의해 정의될 수도 있으며, 9x5 필터는 45 개의 필터 계수들의 세트에 의해 정의될 수도 있는 등이다. 용어 "필터들의 세트"는 일반적으로 하나보다 많은 필터의 그룹을 지칭한다. 예를 들어, 2 개의 3x3 필터들의 세트는 9 개의 필터 계수들의 제 1 세트 및 9 개의 필터 계수들의 제 2 세트를 포함할 수 있다. 일반적으로 때때로 "필터 서포트 (filter support)"로 불리는 용어 "형상"은 특정 필터에 대한 필터 계수들의 로우들의 수 및 필터 계수들의 컬럼들의 수를 지칭한다. 예를 들어, 9x9는 제 1 형상의 일 예이고, 7x7는 제 2 형상의 일 예이며, 5x5는 제 3 형상의 일 예이다. 일부 경우들에서, 필터들은 다이아몬드 형상, 다이아몬드 같은 형상, 원형 형상, 원형 같은 형상, 육각형 형상, 팔각형 형상, 십자 형상, X 형상, T 형상, 다른 기하학적 형상, 또는 다수의 다른 형상 또는 구성을 취할 수도 있다.

[0063] 일 예의 접근법에서, 3 개의 원형 대칭 필터 형상들까지가 서포트된다. 하나의 이러한 예시의 접근법에서, 3 개의 필터 형상들은 도 3a 내지 도 3c 에 나타낸 것들이다. 나타낸 예들에서, 도 3a 는 5x5 다이아몬드를 도시하고; 도 3b 는 7x7 다이아몬드를 도시하며; 도 3c 는 절단된 (truncated) 9x9 다이아몬드를 도시한다. 도 3a 내지 도 3c 의 예들은 다이아몬드 형상들이지만 다른 형상들이 사용될 수도 있다. 가장 일반적인 경우들에서, 필터의 형상에 관계 없이, 필터 마스크에서의 중심 픽셀은 필터링되고 있는 픽셀이다. 다른 예들에서, 필터링된 픽셀은 필터 마스크의 중심으로부터 오프셋될 수도 있다.

[0064] 일 예의 접근법에서, ALF 계수들의 단일 세트가 픽처에서의 크로마 컴포넌트들의 각각에 적용된다. 하나의 이러한 접근법에서, 5x5 다이아몬드 형상 필터가 항상 사용될 수도 있다.

[0065] 이러한 예에서, 비디오 디코더 (30)는 하기 식 (6)에 나타낸 바와 같은 계산들에 기초하여 $I'_{i,j}$ 가 되도록 각각의 픽셀 샘플 $\hat{I}_{i,j}$ 을 필터링할 수도 있으며, 여기서 L 은 필터 길이를 지칭하고, $f_{m,n}$ 는 필터 계수를 나타내며, o 는 필터 오프셋 또는 DC 계수를 표시한다.

$$[0066] I'_{i,j} = \sum_{m=-L}^L \sum_{n=-L}^L f_{m,n} \times \hat{I}_{i+m, j+n} + o \quad (6)$$

[0067] 일 예의 접근법에서, 단 하나의 필터만이 2 개의 크로마 컴포넌트들에 대해 서포트된다.

[0068] 일 예의 접근법에서, 필터들의 총 수 (또는 병합된 그룹들의 총 수)는, ALF 가 슬라이스에 대해 인에이블될 때, 비디오 인코더 (20)로부터 비디오 디코더 (30)로 먼저 시그널링된다. 일부 구현들에서, 그러한 시그널링은 루마 컴포넌트에 대해서만 필요할 수도 있는데, 이는 크로마 컴포넌트만이 하나의 연관된 필터를 가지기 때문이다. 다른 구현들에서, 이러한 시그널링은 루마 및 크로마 컴포넌트들 양자 모두에 대해 포함될 수도 있다. 서포트된 필터 형상들의 수 (예를 들어, 3)의 필터 서포트 인덱스가 또한 시그널링되어 선택된 필터 형상을 표시할 수도 있다. 일부 예들에서, 필터들의 세트에서의 모든 필터들은 동일한 형상을 갖는다.

[0069] 비디오 디코더 (30)는 또한 클래스들에 필터 인덱스들을 할당하기 위한 정보를 수신할 수도 있다. C 의 비연속적인 값들을 갖는 클래스들이 병합될 수도 있으며, 이는 그러한 클래스들이 동일한 필터에 매핑되는 것을 의미한다. 일 예로서, 도 2 를 참조하면, 사용된 스캔 순서에 의존하여, 각각의 클래스는 병합 좌 플래그, 병합 우 플래그, 병합 위 플래그, 또는 병합 아래 플래그 중 하나 이상을 수신할 수도 있다. 각각의 클래스에 대해 하나의 플래그를 코딩함으로써, 클래스가 병합되거나 병합되지 않는 경우, 필터 인덱스는 비디오 디코더 (30)에 의해 도출될 수도 있다. 예를 들어, 각각의 클래스에 대해, 비디오 디코더 (30)는 플래그를 수신할 수도 있다. 플래그가 제 1 값이면, 비디오 디코더 (30)는 이전 클래스에 대해 사용된 동일한 필터 인덱스에 클래스를 매핑할 수도 있다. 플래그가 제 2 값이면, 비디오 디코더 (30)는 클래스가 상이한 필터 인덱스에 매핑하고 그 필터 인덱스를 도출하는 것을 결정할 수 있다. 비디오 디코더 (30)는 예를 들어, 새로운 필터 인덱스가 이전 클래스의 필터 인덱스보다 하나 크거나 하나 작다.

[0070] 일 예에서, 비디오 디코더 (30)는 필터들 중 적어도 하나는 코딩되지 않아야 하는지 여부를 표시하기 위해 `froceCoeff0` 플래그를 수신할 수도 있으며, 이는 필터 계수들에 대한 값들이 비트스트림에서 명시적으로 코딩

되지 않는 것을 의미한다. *froceCoeff0* 플래그가 0 과 동일할 때, 필터들 모두가 코딩되어야 하며, 이는 비디오 디코더 (30) 가, 예를 들어 실제 값들로서 또는 실제 값들과 예측 값들 사이의 차이로서, 비트스트림에서 계수들을 수신하는 것을 의미한다. *froceCoeff0* 플래그가 1 과 동일할 때, 필터가 시그널링되어야 하는지 여부를 표시하기 위해 부가 플래그 (*CodedVarBin* 로 나타냄) 가 각각의 병합된 그룹에 대해 코딩될 수도 있다. 필터가 시그널링되지 않을 때, 그러면 비디오 디코더 (30) 는 필터와 연관된 필터 계수들 모두를 0 과 동일하게 설정한다. 통상적으로, 병합 후 단 하나의 필터만이 있을 때, 또는 환언하면, 단 하나의 병합된 그룹만이 있을 때, *froceCoeff0* 플래그는 시그널링되지 않는다.

[0071] 다수의 병합된 그룹들에 대해 다수의 필터들이 시그널링되어야 할 때, 상이한 방법들이 사용될 수도 있다. 제 1 예에서, 모든 필터들은 필터 정보로 직접 코딩될 수도 있다. 이러한 예에서, 필터 계수들의 값들은 임의의 예측 인코딩 기법들을 사용하지 않으면서 비트스트림으로 인코딩될 수도 있다. 다른 예에서, 제 1 필터의 필터 계수들은 직접 코딩될 수도 있는 한편, 나머지 필터들은 필터 정보로 예측 코딩된다. 이 경우, 필터 계수들의 값들은 이전에 코딩된 필터와 연관된 필터 계수들에 대해 잔차 값들 또는 차이들로 정의될 수도 있다. 이전에 코딩된 필터는 가장 최근에 코딩된 필터일 수도 있어서, 예측자에 의해 사용되고 있는 필터 및 현재 필터의 필터 인덱스들은 연속적이다. 일 예에서, 위의 2 가지 방법들 중 하나의 사용을 표시하기 위해, 하나의 플래그는 병합된 그룹들의 수가 1 보다 크고 *froceCoeff0* 가 0 과 동일할 때 코딩될 수도 있다.

[0072] 도 4 는 필터 계수들의 골롬 (Golomb) 인코딩을 도시한다. 도 4 에 나타낸 예에서, 하나의 필터 서포트는 다수의 카테고리들로 스플릿된다. 나타낸 예에서, 7x7 다이아몬드 필터 형상은 3 개의 카테고리들을 갖는다. 각각의 정사각형은 하나의 필터 계수를 나타낸다. 동일한 수를 갖는 정사각형들은 동일한 골롬 파라미터로 코딩된다.

[0073] 하나의 그러한 예시의 접근법에서, 필터 계수의 시그널링은 골롬 파라미터들 및 계수들의 2 부분을 포함한다. 골롬 파라미터들에 대해, 일 예의 접근법에서, 골롬 파라미터 (*kMin* 으로 나타냄) 의 하나의 값이 먼저 시그널링된 다음, 각각의 카테고리에 대해 1 비트 플래그가 뒤따른다. 1 비트 플래그는 카테고리 *i* (*i* 는 7x7 다이아몬드 대칭 필터 서포트에 대해, 1 부터 3 까지 포함) 의 파라미터가, *i* 가 1 보다 클 때 이전 카테고리 (*i-1*) 에 대해 사용된 골롬 파라미터와 비교하여; 또는 *i* 가 1 과 동일할 때 *kMin* 과 비교하여, 동일하거나 1 만큼 증가되는지 여부를 표시한다. 선택된 골롬 파라미터들에 기초하여, 그 후 계수들의 절대값들이 코딩된 후 부호 플래그가 뒤따른다.

[0074] 시간 예측이 또한 필터 계수들을 예측하는데 활용될 수도 있다. 일 예의 접근법에서, 참조 픽처들의 ALF 계수들은 저장되고 현재 픽처의 ALF 계수들로서 재사용되도록 허용된다. 현재 픽처에 대해, 비디오 코더는 참조 픽처들에 대해 저장된 ALF 계수들을 사용하도록 선택하고 새로운 ALF 계수들의 시그널링을 바이패스할 수도 있다. 이 경우, 참조 픽처들 중 하나에 대한 인덱스만이 시그널링되고, 표시된 참조 픽처의 저장된 ALF 계수들이 현재 픽처에 대해 간단히 승계된다. 하나의 그러한 예시의 접근법에서, 시간 예측의 사용을 표시하기 위해, 인덱스를 전송하기 전에 먼저 하나의 플래그가 코딩된다. 일부 예들에서, 현재 픽처의 참조 픽처가 아닌 이전에 코딩된 픽처와 연관된 ALF 계수들은 또한 ALF 시간 예측에 활용될 수도 있다.

[0075] 다음으로 기하학적 정보 기반 적응적 루프 필터링 (Geometric transformation-based adaptive loop filtering; GALF) 이 논의될 것이다. GALF 에서, 분류는 고려된 대각선 구배들로 수정되고 기하학적 변환들이 필터 계수들에 적용될 수도 있다. GALF 는 M. Karczewicz, L. Zhang, W.-J. Chien, X. Li, "Improvements on adaptive loop filter", ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 Exploration Team (JVET), Doc. JVET-B0060_r1, 2nd Meeting: San Diego, USA, 20-26 February 2016 에 기재되어 있으며, 그 기재는 본 명세서에 참조로서 통합된다.

[0076] 일 예의 접근법에서, 각각의 2x2 블록은 다음과 같이, 그 방향성 및 활동의 양자화된 값에 기초하여 25 개의 클래스들 (C) 중 하나로 카테고리화된다.

$$C=5D+A$$

(7)

[0078] 수평, 수직 및 2 개의 대각선 구배들의 값들은 1-D 라플라시안을 사용하여 계산된다:

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|, \quad (8)$$

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|, \quad (9)$$

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, \quad D1_{k,l} \\ = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad (10)$$

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, \quad D2_{k,l} \\ = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \quad (11)$$

[0080] $= |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|$

[0081] 인덱스들 i 및 j 는 2×2 블록에서의 상부 좌측 픽셀의 좌표들을 지칭한다. 방향성 D 를 할당하기 위해, 수평 및 수직 구배들의 최대 및 최소의 비율,

$$g_{h,v}^{\max} = \max(g_h, g_v), \quad g_{h,v}^{\min} = \min(g_h, g_v), \quad (12)$$

[0082] 및

[0083] 2 개의 대각선 구배들의 최대 및 최소의 비율,

$$g_{d0,d1}^{\max} = \max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{\min} = \min(g_{d0}, g_{d1}), \quad (13)$$

[0084] [0085] i] 서로에 대해 그리고 임계들 t_1 및 t_2 의 세트와 비교된다.

[0086] 단계 1. $g_{h,v}^{\max} \leq t_1 \cdot g_{h,v}^{\min}$ 이고 $g_{d0,d1}^{\max} \leq t_1 \cdot g_{d0,d1}^{\min}$ 인 경우, D 는 0 으로 설정됨.

[0087] 단계 2. $g_{h,v}^{\max}/g_{h,v}^{\min} > g_{d0,d1}^{\max}/g_{d0,d1}^{\min}$ 가 단계 3 으로부터 계속하고, 그렇지 않으면 단계 4 로부터 계속하는 경우.

[0088] 단계 3. $g_{h,v}^{\max} > t_2 \cdot g_{h,v}^{\min}$ 인 경우, D 는 2 로 설정되고, 그렇지 않으면 D 는 1 로 설정됨.

[0089] 단계 4. $g_{d0,d1}^{\max} > t_2 \cdot g_{d0,d1}^{\min}$ 인 경우, D 는 4 로 설정되고, 그렇지 않으면 D 는 3 으로 설정됨.

[0090] 활동 값 A 는 다음과 같이 계산된다.

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}). \quad (14)$$

[0092] A 는 추가로 0 부터 4 를 포함하는 범위로 양자화되고, 양자화된 값은 \hat{A} 로 나타낸다.

[0093] 비디오 코더는 필터 계수들의 기하학적 변환들을 수행할 수도 있다. 대각선, 수직 플립 및 회전을 포함한, 3 개의 기하학적 변환들이 하기에 기재된다.

[0094] 대각선: $f_D(k,l) = f(l,k)$,

[0095] 수직 플립: $f_V(k,l) = f(k, K-l-1), \quad (15)$

[0096] 회전: $f_R(k, l) = f(K - l - 1, k)$.

[0097] 여기서 K 는 필터의 사이즈이고 $0 \leq k, l \leq K - 1$ 는 계수 좌표들이어서, 위치 $(0,0)$ 은 상부 좌측 코너에 있고, 위치 $(K - 1, K - 1)$ 는 하부 우측 코너에 있다. 변환들은 그 블록에 대해 계산된 구배 값들에 의존하여 필터 계수들 $f(k, l)$ 에 적용된다. 식 8 내지 11 을 사용하여 계산된 4 개의 구배들 및 변환 사이의 관계가 표 1 에 기재된다.

[0098] [표 1] 변환들 및 하나의 블록에 대해 계산된 구배의 매핑

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

[0099]

[0100] 시간 예측이 이용가능하지 않을 때 (예를 들어, 인트라 프레임들) 코딩 효율을 개선하기 위해, 고정 필터들의 세트 (예를 들어, 16 고정 필터들) 가 각각의 클래스에 할당될 수도 있다. 이러한 맥락에서, 고정 필터들은 부가 시그널링 없이 비디오 인코더 및 디코더 양자 모두에 의해 알려진 미리정의된 필터들을 지칭한다. 고정 필터의 사용을 표시하기 위해, 각각의 클래스에 대한 플래그가 시그널링되고, 필요하다면, 선택된 고정 필터의 인덱스가 시그널링된다. 고정 필터가 주어진 클래스에 대해 선택될 때에도, 적응적 필터의 계수들 $f(k, l)$ 은 여전히 이 클래스에 대해 전송될 수도 있으며, 이 경우 재구성된 이미지에 적용될 필터의 계수들은 계수들의 양자 모두의 세트들의 합이다. 클래스들의 수는 상이한 고정 필터들이 그들에 대해 선택되었더라도 비트스트림에서 시그널링된 동일한 계수들 $f(k, l)$ 을 공유할 수 있다.

[0101]

3 가지 경우들이 GALF 에 대해 정의된다. 제 1 경우에, 25 개의 클래스들의 필터들 중 어느 것도 고정 필터로부터 예측되지 않는다. 제 2 경우에, 클래스들의 모든 필터들이 고정 필터로부터 예측된다. 제 3 경우에, 일부 클래스들과 연관된 필터들은 고정 필터들로부터 예측되고 클래스들의 나머지와 연관된 필터들은 고정 필터들로부터 예측되지 않는다. 인덱스는 먼저 3 가지 경우들 중 하나를 표시하기 위해 코딩된다. 또한, 다음이 적용된다:

[0102]

제 1 경우에 대해, 고정 필터의 인덱스를 추가로 시그널링할 필요가 없다;

[0103] 그렇지 않으면, 제 2 경우에 대해, 각각의 클래스에 대해 선택된 고정 필터의 인덱스가 시그널링된다;

[0104]

그렇지 않으면 (제 3 경우에 대해), 각각의 클래스에 대해 1 비트가 먼저 시그널링되고, 고정 필터가 사용되면, 인덱스가 추가로 시그널링된다.

[0105]

일 예의 접근법에서, 필터 계수들을 나타내는데 필요한 비트들의 수를 감소시키기 위해, 상이한 클래스들이 병합될 수 있다. 하지만, 위의 Wiegand 등에서 기재된 접근법에서와는 달리, 이웃하지 않는 클래스들 또는 C 의 비연속 값들을 갖는 클래스들 조차도, 클래스들의 임의의 세트가 병합될 수 있다. 병합되는 클래스들을 상세하는 정보는 25 개의 클래스들의 각각에 대해 인덱스 ic 를 전송함으로써 제공되며, 여기서 ic 는 클래스에 할당된 필터 인덱스를 표시한다. 동일한 인덱스 ic 를 갖는 클래스들은 동일한 필터, 예를 들어 동일한 필터 계수들, 동일한 필터 형상 등을 공유한다. 하나의 그러한 접근법에서, 인덱스 ic 는 절단된 고정 길이 방법으로 코딩된다. GALF 의 일부 예시의 구현들에서, 코딩되는 필터 계수들은 최적의 도출된 필터 계수들과 예측된 필터 계수들 (예를 들어, 고정 필터들로부터 선택된 필터) 사이의 차이들일 수도 있다.

[0106]

유사하게, 일부 예시의 접근법들에서, `froceCoeff0` 플래그가 또한 사용된다. `froceCoeff0` 플래그가 1 과 동일할 때, 병합된 그룹들 (코딩될 모든 필터들) 의 각각에 대해 `CodedVarBin` 로 나타낸, 1 비트 플래그가 시그널링된 필터 계수들이 모두 제로인지 여부를 표시하기 위해 추가로 시그널링된다. 또한, 이러한 플래그

가 1 과 동일할 때, 예측적 코딩, 즉 현재 필터와 이전에 코딩된 필터 사이의 차이를 코딩하는 것이 디스에이블 된다. GALF에서, 시그널링된 필터 계수들은 실제 필터 계수들 또는 필터 계수 차이들일 수도 있다.

[0107] 고정 필터들로부터의 예측이 허용될 때, 위에 언급된 시그널링/코딩될 필터들은 재구성된 이미지에 적용된 필터와 선택된 고정 필터 사이의 차이들일 수도 있음을 유의해야 한다. 이 경우, 동일한 *ic*를 갖는 상이한 클래스들은 동일한 필터 계수 차이들을 가질 수도 있다. 하지만, 재구성된 이미지에 적용된 필터는 상이한 선택된 고정 필터들로 인해 상이할 수도 있다. 본 개시물은 일반적으로 필터들을 시그널링하기 위한 기법들을 기재할 수도 있다. 하지만, 달리 언급되지 않으면, 필터(들)은 일반적으로 실제 필터들과 예측된 필터들 사이의 차이 값들 또는 실제 필터 계수값들일 수도 있는 필터 계수 정보를 지칭한다.

[0108] 계수들과 같은 다른 정보가 예를 들어, JEM2.0에서 동일한 방식으로 코딩될 수도 있다. ALF/GALF 설계들에 대한 기준 제안들은 일부 잠재적인 문제들을 가질 수도 있다. 제 1 예로서, ALF 파라미터 시그널링에 대한 현재 설계들로, 다수의 클래스들이 상이한 필터 인덱스들에 할당될 수 있는 한편, 코딩될 필터들의 계수들은 모두 제로이다. 그러한 경우에서, 불필요한 비트는 필터들/병합된 그룹들의 총 수를 시그널링하는데 그리고 그러한 클래스들에 대한 필터 인덱스들을 시그널링하는데 낭비될 수도 있다. 제 2 예로서, *froceCoeff0* 가 1 과 동일할 때, 필터 계수들의 값들, 즉 가장 최근에 이전에 코딩된 필터들과 연관된 필터 계수들에 대한 차이들 또는 잔차 값들의 예측 코딩이 항상 디스에이블될 수도 있으며, 이는 효율을 감소시킬 수도 있다. 제 3 예로서, 현재 접근법들에서, 하나의 카테고리의 골룸 파라미터는, 계수들의 값들이 더 큰 인덱스들을 갖는 카테고리들에 대해 올라가고 있다는 가정에 기초하여, 하나의 이전 카테고리의 값과 비교하여 항상 동일하거나 1 만큼 증가된다. 하지만, 이전 필터 또는 고정 필터로부터의 예측이 인에이블될 때, 그렇지는 않다.

[0109] 위에 언급된 문제들을 해결하기 위해, 다음의 기법들이 개시된다. 다음의 기법들은 개별적으로, 또는 대안으로, 임의의 조합으로 적용될 수도 있다. 다음의 기법들은 비디오 인코더 (20) 또는 비디오 디코더 (30) 중 어느 하나에 의해 적용될 수도 있다. 일부 경우들에서, 비디오 디코더 (30)는 비디오 인코더 (20)에 의해 수행된 인코딩 프로세싱에 상반하는 디코딩 프로세스를 수행할 수도 있다. 일 예로서, 비디오 인코더 (20)가 소정의 정보를 송신하면, 비디오 디코더 (30)는 그러한 정보를 수신할 수도 있다.

[0110] 다음의 논의에서, (고정 필터들 또는 이전에 코딩된 필터로부터의 잠재적 예측 후) 코딩될 하나의 필터의 필터 계수들은 모두 제로일 때 (DC 계수가 있거나 없음), 그러한 필터는 제로 필터로 명명된다.

[0111] 본 개시물은 그룹화 후에 2 이상의 필터들이 있을 때 코딩 시나리오들을 위한 기법들을 도입한다. 비디오 디코더는, 예를 들어 각각의 클래스에 대해 필터 인덱스들을 수신함으로써 그룹화를 수행하며, 클래스들은 동일한 그룹에 속하는 동일한 필터 인덱스를 갖는다. 그러한 시나리오에 대한 일 예의 기법에서, 다수의 클래스들과 연관된 필터들이 제로 필터 (Zero Filter) 들일 때, 이들 클래스들에 대한 필터 인덱스들 (즉, 위의 *ic*)은 동일한 값들을 할당받는다. 즉, 이들 클래스들은 함께 그룹화될 수도 있다.

[0112] 하나님의 그러한 예에서, 이들 클래스들에 대해, 필터 인덱스는 하나의 주어진 수, 예를 들어 최대 필터 인덱스 (필터들의 총 수 또는 필터들의 총 수 마이너스 1) 또는 최소 필터 인덱스로 설정될 수도 있다. 게다가, 이들 필터들의 시그널링 (예를 들어, 인코딩된 비트스트림에서의 송신 및 수신)은 스킵될 수도 있다. 최종적으로, 필터들의 총 수는 제로 필터들이 아닌 필터들의 총 수 플러스 1 (제로 필터가 있는 경우) 또는 플러스 0 (제로 필터가 없는 경우)으로 정의될 수도 있다.

[0113] 그러한 시나리오에 대한 또 다른 예의 기법에 따라, *froceCoeff0* 가 1 과 동일할 때 (즉, 일부 병합된 그룹에 대해 적어도 하나의 제로 필터), 필터 계수들의 값들, 즉 이전에 코딩된 필터와 연관된 필터 계수들에 대한 차이들 및 잔차 값들이 여전히 적용될 수도 있다. 일 예의 접근법에 있어서, 제로 필터가 아닌 필터는 이전에 코딩된 필터들 중 하나 이상으로부터 예측될 수도 있다. 다른 예의 접근법에 있어서, 제로 필터가 아닌 필터는 제로 필터들이 아닌 이전에 코딩된 필터들 중 하나 이상으로부터만 예측될 수도 있다. 또 다른 예의 접근법에 있어서, 재구성된 이미지에 적용될 필터의 계수들은 시그널링된 계수들, 고정 필터 계수들, 및/또는 그것의 이전에 코딩된 필터들의 합수일 수도 있다. 그러한 예시의 접근법에서 합수는 합산으로서 정의된다.

[0114] 본 개시물은 또한 그룹화 후 단지 하나의 필터만 있을 때 코딩 시나리오들을 위한 기법들을 도입한다. 그러한 시나리오에서, 필터가 제로 필터일 때, 필터 서포트 정보는 코딩되지 않을 수도 있다. 또한, 고정 필터들로부터의 예측이 허용될 때, 필터 서포트는 고정 필터들의 필터 서포트로서 정의될 수도 있다.

- [0115] 그러한 시나리오에 대한 다른 예에서, 하나의 병합된 그룹만이 있을 때, 필터 중 적어도 하나가 제로 필터인지 여부를 표시하는 `froceCoeff0` 플래그가 여전히 시그널링될 수도 있다. 즉, 하나의 병합된 그룹 및 단 하나의 제로 필터의 경우가 허용된다. 예를 들어, 비디오 블록들의 그룹에 대해, 비디오 디코더 (30)는 복수의 클래스들을 위해 병합된 그룹들의 수가 1과 동일하다고 결정하고, 블록들의 그룹에 대한 필터들의 세트로부터 적어도 하나의 필터가 비디오 데이터에서 코딩되지 않는 것을 표시하도록 설정된 `froceCoeff0` 플래그를 수신하며, 그리고 비디오 데이터에서, 하나의 병합된 그룹에 대해, 하나의 병합된 그룹에 매핑된 필터가 모두 제로 필터인 것을 표시하는 `CodedVarBin` 을 수신할 수도 있다.
- [0116] 다른 예에서, 단 하나의 병합된 그룹만 있고 필터가 고정 필터들 중 하나로부터 예측될 때, 일 예의 접근법에서, `froceCoeff0` 플래그가 시그널링된다. 그렇지 않으면, 단 하나의 병합된 그룹만 있고 필터가 고정 필터들 중 하나로부터 예측되지 않을 때, `froceCoeff0` 플래그의 시그널링은 스킵될 수도 있고 필터 계수들은 코딩될 수도 있다.
- [0117] 일 예의 접근법에서, 그룹화 후의 필터들의 총 수에 관계없이, 코딩될 하나의 필터의 모든 계수들이 제로인 것을 표시하는 플래그 (`CodedVarBin`)의 시그널링이 제거될 수도 있고, 이를 플래그들의 값들은 시그널링된 필터 인덱스들로부터 도출될 수도 있다. 일 예에서, 최대 또는 최소 필터 인덱스가 하나의 클래스에 대해 사용될 때, 코딩될 대응 필터의 필터 계수들은 모두 제로일 수도 있다.
- [0118] 일 예의 접근법에 있어서, 그룹화 후의 필터들의 총 수에 관계없이, 각각의 클래스의 필터들의 인덱스를 코딩하는 대신, 각각의 필터와 연관된 클래스 인덱스들을 코딩한다. 하나의 그러한 예시의 접근법에서, 각각의 필터에 대해, 이러한 필터를 사용하는 최소 클래스 인덱스는 직접 코딩된다. 나머지 클래스들의 각각에 대해, 클래스가 동일한 필터를 사용하는 경우, 그것의 이전 클래스 인덱스와 비교하여 클래스 인덱스의 차이가 코딩된다.
- [0119] 다른 그러한 예시의 접근법에서, 동일한 필터를 갖는 클래스들 모두가 코딩되었다는 것을 표시하기 위해 특수 값이 코딩된다. 대안으로, 동일한 필터를 사용하는 클래스들의 총 수가 코딩된다.
- [0120] 또 다른 예시의 접근법에서, 비디오 코더는 필요할 때 각각의 클래스에 대한 필터들의 코딩 인덱스들과 각각의 필터에 대한 코딩 클래스 인덱스들 사이에서 스위칭한다. 하나의 그러한 예시의 접근법에서, 플래그는 2 가지 방법들 중 선택된 하나를 표시하기 위해 시그널링된다. 또 다른 그러한 접근법에서, 플래그는 제로 필터들을 배제한 시그널링될 필터들의 총 수에 의존할 수도 있다.
- [0121] 다음으로 골룸 파라미터들의 시그널링이 논의될 것이다. 일 예의 접근법에서, 필터 계수들을 코딩할 때, 제1 카테고리에 대한 1 비트 플래그 및 제1 골룸 파라미터를 코딩하는 대신, 제1 카테고리의 골룸 파라미터가 직접 코딩된다. 일 예의 접근법에서, 카테고리 i 와 (i-1)에 대한 골룸 파라미터 사이의 차이가 추가로 코딩된다.
- [0122] 일 예의 접근법에서, 연속적인 카테고리들에 대한 2 개의 골룸 파라미터들의 차이는 1 보다 크거나 0 보다 작을 수도 있다.
- [0123] 일 예의 접근법에서, 필터 계수들을 코딩할 때, 골룸 파라미터들을 항상 오름 차순으로 유지하는 대신, 골룸 파라미터들이 이전 카테고리에 대해 사용된 골룸 파라미터와 비교하여 증가되거나 감소될 수도 있다. 하나의 그러한 예의 접근법에 있어서, 골룸 파라미터 증분 또는 감분의 범위는 상이한 카테고리들에 대해 상이할 수도 있다. 즉, 이전 것과 비교한 차이를 표시하는 동일한 코딩/디코딩된 값은 상이한 카테고리들에 대한 차이 값들에 매핑될 수도 있다. 또 다른 그러한 접근법에서, 디폴트G로 나타낸 하나의 골룸 파라미터가 먼저 시그널링된 다음, 디폴트G 와 하나의 카테고리에 대한 골룸 파라미터 사이의 차이들이 후속한다.
- [0124] 다음으로 크로마 컴포넌트들에 대한 예측 코딩 및 필터들이 논의될 것이다. 일 예의 접근법에서, (고정 필터들로부터 가능한 예측 후) 시그널링될 필터들에 대해, 이들의 각 하나는 예측 코딩을 사용할지 여부를 선택할 수도 있다. 하나의 이러한 예시의 접근법에서, 필터들의 적어도 하나가 예측 코딩을 사용할지 여부를 표시하기 위해 1 비트 플래그가 먼저 시그널링될 수도 있다. 또한, 코딩될 제1 필터를 제외한 각각의 필터에 대해, 예측 코딩의 사용을 표시하기 위해 하나의 플래그가 시그널링될 수도 있다. 대안으로, 각각의 필터와 연관된 플래그는 예를 들어, 필터가 고정 필터로부터 예측되지 않은 하나의 클래스를 연관시키는지 여부와

같은, 소정의 조건들에 기초하여 스kip될 수도 있다.

[0125] 일 예의 접근법에서, 필터를 예측하기 위해 항상 최근 이전에 코딩된 필터를 사용하는 대신, 그 하나의 필터는 연속적으로 코딩되지 않은 하나 이상의 다른 코딩된 필터들로부터 예측될 수도 있다.

[0126] 일 예의 접근법에서, 필터는 다수의 이전에 코딩된 필터들로부터 예측될 수도 있다. 예측자는 2 이상이 이전에 코딩된 필터들의 함수에 의해 생성될 수도 있다.

[0127] 일 예의 접근법에서, 이전에 코딩된 필터로부터의 예측이 인에이블될 때, 이전에 코딩된 필터들 중 어느 것이 추가로 시그널링되는지를 표시하기 위해 인덱스가 사용된다. 대안으로, 일 예의 접근법에서, 현재 필터와 그 예측 필터의 필터 인덱스들의 차이가 시그널링될 수도 있다.

[0128] 일 예의 접근법에서, 상이한 필터가 각각의 크로마 컴포넌트에 대해 사용될 수도 있다. 하나의 그러한 예의 접근법에서, 2 개의 필터들이 시그널링되고, 각각의 필터는 하나의 크로마 컴포넌트에 적용된다. 다른 그러한 예의 접근법에서, 제 2 크로마 컴포넌트의 필터는 예측적으로 코딩된다. 하나의 그러한 예의 접근법에서, 2 개의 필터들 사이의 필터 계수들의 차이가 코딩된다. 다른 그러한 예의 접근법에서, 상이한 컬러 컴포넌트들이 동일하거나 상이한 필터들을 사용해야 하는지 여부를 표시하기 위해 하나의 플래그가 먼저 시그널링된다.

[0129] 일 예의 접근법에서, 크로마 컴포넌트들에 대한 필터가 루마 컴포넌트들에 대한 필터들 중 하나로부터 승계된다. 예를 들어, 필터는 슬라이스 당 또는 블록 당 승계된다. 하나의 그러한 예의 접근법에서, 크로마 컴포넌트들에 대한 필터 계수들의 시그널링은 스kip될 수도 있다. 하나의 그러한 예의 접근법에서, 루마 컴포넌트에 대한 클래스 0 의 필터는 크로마 컴포넌트들 중 하나 이상에 의해 승계될 수도 있다. 다른 그러한 예의 접근법에서, 루마 컴포넌트에 대해 사용된 최소 인덱스를 갖는 필터가 크로마 컴포넌트들 중 하나 이상에 의해 승계된다.

[0130] 도 5 는 본 개시물에 기재된 기법들을 구현할 수도 있는 일 예의 비디오 인코더 (20) 를 도시하는 블록 다이어 그램이다. 비디오 인코더 (20) 는 비디오 슬라이스를 내에서 비디오 블록들의 인트라 및 인터 코딩을 수행 할 수도 있다. 인트라 코딩은 주어진 비디오 프레임 또는 꾹쳐 내의 비디오에서 공간 리던던시를 감소 또는 제거하기 위해 공간 예측에 의존한다. 인터 코딩은 비디오의 인접 프레임들 또는 꾹쳐들 내의 비디오에서 시간 리던던시를 감소 또는 제거하기 위해 시간 예측에 의존한다. 인트라 모드 (I 모드) 는 몇몇 공간 기반 압축 모드들 중 임의의 것을 지칭할 수도 있다. 일 방향 예측 (P 모드) 또는 양 방향 예측 (B 모드) 과 같은 인터 모드들은, 몇몇 시간 기반 압축 모드들 중 임의의 것을 지칭할 수도 있다.

[0131] 도 5 의 예에서, 비디오 인코더 (20) 는 비디오 데이터 메모리 (33), 파티셔닝 유닛 (35), 예측 프로세싱 유닛 (41), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 엔트로피 인코딩 유닛 (56) 을 포함한다. 예측 프로세싱 유닛 (41) 은 모션 추정 유닛 (MEU)(42), 모션 보상 유닛 (MCU)(44), 및 인트라 예측 유닛 (46) 을 포함한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 역 양자화 유닛 (58), 역 변환 프로세싱 유닛 (60), 합산기 (62), ALF 유닛 (64), 및 디코딩된 꾹쳐 베퍼 (DPB)(66) 를 포함한다.

[0132] 도 5 에 나타낸 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 수신하고 수신된 비디오 데이터를 비디오 데이터 메모리 (33) 에 저장한다. 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (33) 에 저장된 비디오 데이터는, 예를 들어 비디오 소스 (18) 로부터 획득될 수도 있다. DPB (66) 는 예를 들어, 인트라 또는 인터 코딩 모드들에서, 비디오 인코더 (20) 에 의해 인코딩 비디오 데이터에서의 사용을 위해 참조 비디오 데이터를 저장하는 참조 꾹쳐 메모리일 수도 있다. 비디오 데이터 메모리 (33) 및 DPB (66) 는 동기식 동적 랜덤 액세스 메모리 (SDRAM), 자기저항성 RAM (MRAM), 저항성 RAM (RRAM) 또는 다른 유형의 메모리 디바이스들을 포함한, DRAM 과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (33) 및 DPB (66) 는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 다른 컴포넌트들과 온-칩, 또는 그러한 컴포넌트들에 대해 오프-칩일 수도 있다.

[0133] 파티셔닝 유닛 (35) 은 비디오 데이터 메모리 (33) 로부터 비디오 데이터를 취출하고 비디오 데이터를 비디오 블록들로 파티셔닝한다. 이러한 파티셔닝은 또한, 슬라이스들, 타일들, 또는 다른 더 큰 유닛들로의 파티셔닝 뿐만 아니라, 예를 들어 LCU들 및 CU들의 쿼드크리 구조에 따른, 비디오 블록 파티셔닝을 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로 인코딩될 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들

을 도시한다. 슬라이스는 다수의 비디오 블록들로 (그리고 가능하게는 타일들로서 지칭되는 비디오 블록들 의 세트로) 분할될 수도 있다. 예측 프로세싱 유닛 (41)은 여러 결과들 (예를 들어, 코딩 레이트 및 왜곡 레벨)에 기초한 현재 비디오 블록에 대해, 복수의 인터 코딩 모드들 중 하나 또는 복수의 인트라 코딩 모드들 중 하나와 같은, 복수의 가능한 코딩 모드들 중 하나를 선택할 수도 있다. 예측 프로세싱 유닛 (41)은 결과의 인트라 또는 인터 코딩된 블록을 합산기 (50)에 제공하여 잔차 블록 데이터를 생성하고, 합산기 (62)에 제공하여 참조 픽처로서 사용하기 위한 인코딩된 비디오 블록을 재구성할 수도 있다.

[0134] 예측 프로세싱 유닛 (41) 내의 인트라 예측 유닛 (46)은 공간 압축을 제공하기 위해 코딩될 현재 블록과 동일한 프레임 또는 슬라이스에서 하나 이상의 이웃한 블록들에 대해 현재 비디오 블록의 인트라 예측 코딩을 수행할 수도 있다. 예측 프로세싱 유닛 (41) 내의 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 시간 압축을 제공하기 위해 하나 이상의 참조 픽처들에서 하나 이상의 예측 블록들에 대해 현재 비디오 블록의 인터 예측 코딩을 수행한다.

[0135] 모션 추정 유닛 (42)은 비디오 시퀀스에 대해 미리 결정된 패턴에 따라 비디오 슬라이스에 대한 인터 예측 모드를 결정하도록 구성될 수도 있다. 미리 결정된 패턴은 P 슬라이스들 또는 B 슬라이스들로서 시퀀스에서 비디오 슬라이스들을 지정할 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 고도로 통합될 수도 있지만, 개념적 목적을 위해 별도로 도시된다. 모션 추정 유닛 (42)에 의해 수행된 모션 추정은 비디오 블록들에 대한 모션을 추정하는, 모션 벡터들을 생성하는 프로세스이다. 모션 벡터는, 예를 들어 참조 픽처 내의 예측 블록에 대한 현재 비디오 프레임 또는 픽처 내에서 비디오 블록의 PU의 배치를 표시할 수도 있다.

[0136] 예측 블록은 절대 차의 합 (SAD), 제곱 차의 합 (SSD) 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 픽셀 차이에 관하여 코딩될 비디오 블록의 PU와 근접하게 매칭하는 것으로 발견되는 블록이다. 일부 예들에서, 비디오 인코더 (20)는 DPB (66)에 저장된 참조 픽처들의 서브-정수 픽셀 위치들에 대한 값을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20)는 참조 픽처의 1/4 픽셀 위치, 1/8 픽셀 위치, 또는 다른 분수 픽셀 위치의 값을 보간할 수도 있다. 따라서, 모션 추정 유닛 (42)은 전체 픽셀 위치 및 분수 픽셀 위치에 대한 탐색을 수행하고 분수 픽셀 정밀도로 모션 벡터를 출력할 수도 있다.

[0137] 모션 추정 유닛 (42)은 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를, PU의 위치를 참조 픽처의 예측 블록의 위치와 비교함으로써 계산한다. 참조 픽처는 각각이 DPB (66)에 저장된 하나 이상의 참조 픽처들을 각각 식별하는 제 1 참조 픽처 리스트 (List 0) 또는 제 2 참조 픽처 리스트 (List 1)로부터 선택될 수도 있다. 모션 추정 유닛 (42)은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44)에 전송한다.

[0138] 모션 보상 유닛 (44)에 의해 수행된 모션 보상은, 모션 추정에 의해 결정된 모션 벡터에 기초하여 예측 블록을 페칭 (fetching) 또는 생성하는 것, 가능하게는 서브 픽셀 정밀도에 대한 보간을 수행하는 것을 수반할 수도 있다. 현재 비디오 블록의 PU에 대한 모션 벡터의 수신 시, 모션 보상 유닛 (44)은 참조 픽처 리스트들 중 하나에서 포인트하는 예측 블록을 로케이팅할 수도 있다. 비디오 인코더 (20)는 코딩되고 있는 현재 비디오 블록의 픽셀 값을로부터 예측 블록의 픽셀 값을 감산하는 것에 의해 잔차 비디오 블록을 형성한다. 픽셀 차이 값들은 블록에 대한 잔차 데이터를 형성하고, 루마 및 크로마 차이 컴포넌트들 양자 모두를 포함할 수도 있다. 합산기 (50)는 이러한 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낼 수도 있다. 모션 보상 유닛 (44)은 또한 비디오 슬라이스의 비디오 블록들을 디코딩하는데 있어서 비디오 디코더 (30)에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관된 선택스 엘리먼트들을 생성할 수도 있다.

[0139] 예측 프로세싱 유닛 (41)이 현재 비디오 블록에 대해 예측 블록을 생성한 후, 인트라 예측 또는 인터 예측 중 어느 하나를 통해, 비디오 인코더 (20)는 현재 비디오 블록으로부터 예측 블록을 감산하는 것에 의해 잔차 비디오 블록을 형성한다. 잔차 블록에서의 잔차 비디오 데이터는 하나 이상의 TU들에 포함되고 변환 프로세싱 유닛 (52)에 적용될 수도 있다. 변환 프로세싱 유닛 (52)은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은, 변환을 사용하여 잔차 비디오 데이터를 잔차 변환 계수들로 변환한다. 변환 프로세싱 유닛 (52)은 픽셀 도메인으로부터의 잔차 비디오 데이터를 주파수 도메인과 같은, 변환 도메인으로 컨버팅할 수도 있다.

[0140] 변환 프로세싱 유닛 (52)은 결과의 변환 계수들을 양자화 유닛 (54)에 전송할 수도 있다. 양자화 유닛 (54)은 변환 계수들을 양자화하여 비트 레이트를 더 감소시킨다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트를 감소시킬 수도 있다. 양자화도는 양자화 파라미터를 조정함으로써 수정될 수도 있다. 일부 예들에서, 양자화 유닛 (54)은 그 후 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수

도 있다. 대안으로, 엔트로피 인코딩 유닛 (56) 은 스캔을 수행할 수도 있다.

[0141] 양자화 다음에, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 콘텍스트-적응적 가변 길이 코딩 (context adaptive variable length coding; CAVLC), 콘텍스트-적응적 이진 산술 코딩 (CABAC), 신택스 기반 콘텍스트 적응적 이진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding; SBAC), 확률 인터벌 파티셔닝 엔트로피 (probability interval partitioning entropy; PIPE) 코딩 또는 다른 엔트로피 코딩 방법 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 인코딩 다음에, 인코딩된 비트스트림은 비디오 디코더 (30) 에 송신되거나, 비디오 디코더 (30) 에 의한 이후의 송신 또는 취출을 위해 아카이브될 수도 있다. 엔트로피 인코딩 유닛 (56) 은 또한, 코딩되고 있는 현재 비디오 슬라이스에 대해 모션 벡터들 및 다른 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0142] 역 양자화 유닛 (58) 및 역 변환 프로세싱 유닛 (60) 은 참조 픽처의 참조 블록으로서 이후에 사용하기 위해 픽셀 도메인에서 잔차 블록을 재구성하는데 역 양자화 및 역 변환을 각각 적용한다. 모션 보상 유닛 (58) 은 참조 픽처 리스트들 중 하나의 참조 픽처 리스트 내에서 참조 픽처들 중 하나의 참조 블록의 예측 블록에 잔차 블록을 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44) 은 또한, 모션 추정에서의 사용을 위한 서브 정수 픽셀 값들을 계산하기 위해 재구성된 잔차 블록에 하나 이상의 보간 필터들을 적용할 수도 있다. 합산기 (62) 는 재구성된 블록을 생성하기 위해 모션 보상 유닛 (44) 에 의해 생성된 모션 보상 예측 블록에 재구성된 잔차 블록을 가산한다.

[0143] ALF 유닛 (64) 은 재구성된 블록 (예를 들어, 합산기 (62) 의 출력) 을 필터링하고 필터링된 재구성된 블록을 참조 블록으로서 사용하기 위해 DPB (66) 에 저장한다. 참조 블록은 후속 비디오 프레임 또는 픽처에서 블록을 인터 예측하기 위해 참조 블록으로서 모션 보상 유닛 (44) 및 모션 추정 유닛 (42) 에 사용될 수도 있다. 도 5 에 명시적으로 나타내지는 않았지만, 비디오 인코더 (20) 는 부가 필터들, 예컨대 디블록 필터, 샘플 적응적 오프셋 (SAO) 필터, 또는 다른 유형의 루프 필터들을 포함할 수도 있다. 디블록 필터는, 예를 들어 재구성된 비디오로부터 블록화 현상 아티팩트들을 제거하기 위해 블록 경계들을 필터링하는데 디블록킹 필터를 적용할 수도 있다. SAO 필터는 전체 코딩 품질을 개선하기 위해 재구성된 픽셀 값들에 오프셋들을 적용할 수도 있다. 부가 루프 필터들 (인 루프 또는 포스트 루프) 이 또한 사용될 수도 있다.

[0144] 다른 컴포넌트들과 협력하는 ALF 유닛 (64) 은 본 개시물에 기재된 다양한 기법들을 수행하도록 구성될 수도 있다. ALF 유닛 (64) 은 예를 들어, 위에 논의된 신택스 엘리먼트들과 같은 신택스 엘리먼트들에 대한 값들을 결정할 수도 있고, 엔트로피 인코딩 유닛 (56) 은 예를 들어, 인코딩된 비디오 데이터의 비트스트림에서 그러한 신택스 엘리먼트들을 인코딩할 수도 있다.

[0145] ALF 유닛 (64) 은 또한, 비디오 블록의 개별 픽셀에 대해, 또는 서브 블록 (예를 들어, 2x2, 4x4, 또는 일부 다른 사이즈 서브 블록) 에 대해, 하나 이상의 메트릭들에 대한 값을 결정함으로써 비디오 블록들에 필터링을 적용하고, 하나 이상의 메트릭들에 기초하여, 픽셀 또는 서브 블록에 대한 클래스를 결정할 수도 있다. ALF 유닛 (64) 은 예를 들어, 상술한 식들 (2-5 또는 7-14) 을 사용하여 클래스 및 메트릭들에 대한 값을 결정할 수도 있다. ALF 유닛 (64) 은 그 후 결정된 클래스에 매핑된 필터들의 세트로부터, 필터를 사용하여 픽셀 또는 서브 블록을 필터링할 수도 있다.

[0146] 도 6 은 본 개시물에 기재된 기법들을 구현할 수도 있는 일 예의 비디오 디코더 (30) 를 도시하는 블록 다이어 그램이다. 도 6 의 비디오 디코더 (30) 는, 예를 들어 도 5 의 비디오 인코더 (20) 에 대해 상술한 시그널링을 수신하도록 구성될 수도 있다. 도 6 의 예에서, 비디오 디코더 (30) 는 비디오 데이터 메모리 (78), 엔트로피 디코딩 유닛 (80), 예측 프로세싱 유닛 (81), 역 양자화 유닛 (86), 역 변환 프로세싱 유닛 (88), 합산기 (90), 및 DPB (94) 를 포함한다. 예측 프로세싱 유닛 (81) 은 모션 추정 유닛 (82) 및 인트라 예측 유닛 (84) 을 포함한다. 비디오 디코더 (30) 는, 일부 예들에서, 일반적으로 도 5 로부터 비디오 인코더 (20) 에 대해 기재된 인코딩 패스에 상반하는 디코딩 패스를 수행할 수도 있다.

[0147] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 비디오 인코더 (20) 로부터 연관된 신택스 엘리먼트들 및 인코딩된 비디오 슬라이스의 비디오 블록들을 나타내는 인코딩된 비디오 비트스트림을 수신한다. 비디오 디코더 (30) 는 비디오 데이터 메모리 (78) 에 수신된 인코딩된 비디오 비트스트림을 저장한다. 비디오 데이터 메모리 (78) 는 비디오 디코더 (30) 의 컴포넌트들에 의해 디코딩될, 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (78) 에 저장된 비디오 데이터는, 예를 들어 링크 (16) 를 통해, 저장 디바이스 (26) 로부터, 또는 카메라와 같은 로컬 비디오 소스로부터, 또는 물리적 데이터 저장

매체들에 액세스하는 것에 의해 획득될 수도 있다. 비디오 데이터 메모리 (78)는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (CPB)를 형성할 수도 있다. DPB (94)는 예를 들어, 인트라 또는 인터 코딩 모드들에서, 비디오 디코더 (30)에 의해 디코딩 비디오 데이터에서의 사용을 위한 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 메모리 (78) 및 DPB (94)는 DRAM, SDRAM, MRAM, RRAM 또는 다른 유형의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (78) 및 DPB (94)는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (78)는 비디오 디코더 (30)의 다른 컴포넌트들과 온-칩, 또는 그러한 컴포넌트들에 대해 오프-칩 일 수도 있다.

[0148] 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (80)은 비디오 데이터 메모리 (78)에 저장된 비디오 데이터를 엔트로피 디코딩하여 양자화된 계수들, 모션 벡터들, 및 다른 신팩스 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛 (80)은 모션 벡터들 및 다른 신팩스 엘리먼트들을 예측 프로세싱 유닛 (81)에 포워딩한다. 비디오 디코더 (30)는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신팩스 엘리먼트들을 수신할 수도 있다.

[0149] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (81)의 인트라 예측 유닛 (84)은 현재 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터 및 시그널링된 인트라 예측 모드에 기초한 현재 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 슬라이스 (예를 들어, B 슬라이스 또는 P 슬라이스)로서 코딩될 때, 예측 프로세싱 유닛 (81)의 모션 보상 유닛 (82)은 엔트로피 인코딩 유닛 (80)으로부터 수신된 다른 신팩스 엘리먼트들 및 모션 벡터들에 기초한 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 참조 픽처 리스트들 중 하나의 참조 픽처 리스트 내에서 참조 픽처들 중 하나의 참조 픽처로부터 생성될 수도 있다. 비디오 디코더 (30)는 DPB (94)에 저장된 참조 픽처들에 기초한 디폴트 구성 기법들을 사용하여, 참조 프레임 리스트들, List 0 및 List 1을 구성할 수도 있다.

[0150] 모션 보상 유닛 (82)은 모션 벡터들 및 다른 신팩스 엘리먼트들을 과정함으로써 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 예측 정보를 사용하여 디코딩되고 있는 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (82)은 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용된 예측 모드 (예를 들어, 인트라 또는 인터 예측), 인터 예측 슬라이스 유형 (예를 들어, B 슬라이스 또는 P 슬라이스), 슬라이스에 대한 참조 픽처 리스트들의 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 스테이터스, 및 현재 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 결정하기 위해 수신된 신팩스 엘리먼트의 일부를 사용한다.

[0151] 모션 보상 유닛 (82)은 또한 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (82)은 참조 블록들의 서브 정수 픽셀들에 대해 보간된 값들을 계산하기 위해 비디오 블록들의 인코딩 동안 비디오 인코더 (20)에 의해 사용된 보간 필터들을 사용할 수도 있다. 이 경우, 모션 보상 유닛 (82)은 수신된 신팩스 엘리먼트들로부터 비디오 인코더 (20)에 의해 사용된 보간 필터들을 결정하고 그 보간 필터들을 사용하여 예측 블록들을 생성할 수도 있다.

[0152] 역 양자화 유닛 (86)은 엔트로피 디코딩 유닛 (80)에 의해 디코딩되고 비트스트림에서 제공된 양자화된 변환 계수들을 역 양자화, 즉 양자화 해제한다. 역 양자화 프로세스는 양자화도, 및 마찬가지로 적용되어야 하는 역 양자화도를 결정하기 위해 비디오 슬라이스에서 각각의 비디오 블록에 대해 비디오 인코더 (20)에 의해 계산된 양자화 파라미터의 사용을 포함할 수도 있다. 역 변환 프로세싱 유닛 (88)은 픽셀 도메인에서 잔차 블록들을 생성하기 위해, 역 변환, 즉 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스를 변환 계수들에 적용한다.

[0153] 예측 프로세싱 유닛이 예를 들어, 인트라 또는 인터 예측을 사용하여 현재 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30)는 모션 보상 유닛 (82)에 의해 생성된 대응 예측 블록들과, 역 변환 프로세싱 유닛 (88)으로부터의 잔차 블록들을 합산하는 것에 의해 재구성된 비디오 블록을 생성한다. 합산기 (90)는 이러한 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. ALF 유닛 (92)은 예를 들어, 본 개시물에 기재된 ALF 기법들 중 하나 이상을 사용하여, 재구성된 비디오 블록을 필터링한다.

[0154] ALF 유닛 (92)은 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (80)과 같은 다른 컴포넌트들과 협력하여 본 개시물이 기재된 다양한 기법들을 수행하도록 구성될 수도 있다. ALF 유닛 (92)은 예를 들어, 위에 논의된

신택스 엘리먼트들과 같은 선택스 엘리먼트들을 엔트로피 디코딩 유닛 (80) 또는 다른 곳으로부터 수신하고 그려한 선택스 엘리먼트들에 기초하여, 하나 이상의 필터들의 세트를 결정할 수도 있다.

[0155] ALF 유닛 (92) 은 예를 들어, 비디오 블록의 개별 픽셀에 대해, 또는 서브 블록 (예를 들어, 2x2, 4x4, 또는 일부 다른 사이즈 서브 블록)에 대해, 하나 이상의 메트릭들에 대한 값들을 결정함으로써 비디오 블록을 필터링하고, 하나 이상의 메트릭들에 기초하여, 픽셀 또는 서브 블록에 대한 클래스를 결정할 수도 있다. ALF 유닛 (92) 은 예를 들어, 상술한 식들 (2-5 또는 7-14) 를 사용하여 클래스 및 메트릭들에 대한 값들을 결정할 수도 있다. ALF 유닛 (92) 은 그 후 결정된 클래스에 매핑된 필터들의 세트로부터, 필터를 사용하여 픽셀 또는 서브 블록을 필터링할 수도 있다.

[0156] 도 6 에 명시적으로 나타내지는 않았지만, 비디오 디코더 (30) 는 또한 디블록킹 필터, SAO 필터, 또는 다른 유형의 필터들 중 하나 이상을 포함할 수도 있다. 다른 루프 필터들이 (코딩 루프에서 또는 코딩 루프 후) 또한 픽셀 천이들을 평활화하거나 그렇지 않으면 비디오 품질을 개선하기 위해 사용될 수도 있다. 주어진 프레임 또는 픽처에서의 디코딩된 비디오 블록들은 그 후 DPB (94) 에 저장되며, 이는 후속 모션 보상을 위해 사용된 참조 픽처들을 저장한다. DPB (94) 는 도 1 의 디스플레이 디바이스 (32) 와 같은, 디스플레이 디바이스 상의 이후의 프레젠테이션을 위해 디코딩된 비디오를 저장하는 부가 메모리의 부분이거나 이 부가 메모리와 별도일 수도 있다.

[0157] 도 7 은 본 개시물의 기법들에 따른 비디오 데이터를 디코딩하기 위한 기법을 도시하는 플로우 다이어그램이다. 도 7 의 기법들은 예를 들어, 비디오 디코더 (30) 에 의해 수행될 수도 있지만, 도 7 의 기법들은 임의의 특정 유형의 비디오 디코더에 제한되지 않는다. 도 7 의 예에서, 비디오 디코더 (30) 는 비디오 데이터의 비디오 블록들의 그룹에 대해, 복수의 클래스들을 위해 병합된 그룹들의 수를 결정한다 (100). 비디오 블록들의 그룹은 예를 들어, 비디오 데이터의 슬라이스일 수도 있고, 또는 SPS 또는 PPS 와 같은 소정의 파라미터 세트가 적용되는 모든 블록들일 수도 있다. 비디오 블록들의 그룹은 또한 LCU 의 CU들과 같은 더 큰 블록의 서브 블록들일 수도 있다. 복수의 클래스들의 각각의 클래스는 도 2 를 참조하여 위에 논의된 메트릭들과 같은, 메트릭들에 대한 대응 값들을 가지며, 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑된다. 각각의 병합된 그룹은 동일한 필터 계수 정보에 매핑되는 하나 이상의 클래스들을 포함한다.

[0158] 비디오 디코더 (30) 는 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는, 1 과 동일하다고 결정한다 (102). 복수의 클래스들을 위해 병합된 그룹들의 수가 1 과 동일하다고 결정하기 위해, 비디오 디코더 (30) 는 예를 들어, 필터들의 세트에서의 필터들의 총 수가 1 과 동일한 것을 표시하는 선택스를, 비디오 데이터에서 수신할 수도 있다.

[0159] 비디오 디코더는 비디오 데이터에서, 비디오 블록들의 그룹들에 대해, 제 1 플래그 (예를 들어, 상술한 **froceCoeff0** 플래그) 를 수신한다 (104). 제 1 플래그에 대한 제 1 값은 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하고, 제 1 플래그에 대한 제 2 값은 병합된 그룹들 모두에 대한 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시한다. 비디오 디코더 (30) 는 제 1 플래그가 제 1 값과 동일하다고 결정한다 (106). 비디오 디코더 (30) 는, 비디오 데이터에서, 하나의 병합된 그룹에 대해, 제 2 플래그 (예를 들어, 상술한 **CodedVarBin**) 를 수신한다 (108). 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 모든 제로 값들인 것을 표시한다. 비디오 디코더 (30) 는 제 2 플래그가 제 2 값과 동일하다고 결정한다 (110).

[0160] 비디오 디코더 (30) 는 모든 제로 값들을 사용하여 필터들의 세트로부터 하나 이상의 필터들을 결정한다 (112). 일부 예들에서, 하나의 병합된 그룹에 매핑된 필터 계수 정보는 필터 계수 값들일 수도 있다. 그 경우, 필터들의 세트로부터의 하나 이상의 필터들은 적어도 하나의 모두 제로 필터를 포함하고, 여기서 모두 제로 필터에 대한 모든 계수들은 제로와 동일하다. 다른 예들에서, 하나의 병합된 그룹에 매핑된 필터 계수 정보는 차이 값들일 수도 있다. 그 경우, 차이 값들은 모두 제로와 동일하고, 모두 제로 값들을 사용하여 필터들의 세트로부터의 하나 이상의 필터들을 결정하기 위해, 비디오 디코더 (30) 는 하나 이상의 예측 필터들에 차이 값들을 부가할 수도 있다. 그러한 예에서, 필터들의 세트는 하나 이상의 예측 필터들과 동일하거나 이 필터들을 포함할 수도 있다. 하나 이상의 예측 필터들은 예를 들어, 하나 이상의 고정 필터들일 수도 있다.

[0161] 비디오 블록들의 제 1 그룹에 대해, 비디오 디코더 (30) 는 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는 1 과 동일하다고 결정할 수도 있다. 비디오

블록들의 제 2 그룹에 대해 복수의 클래스들을 위해 병합된 그룹들의 수가 비디오 블록들의 제 2 그룹에 대한 하나의 병합된 그룹과 동일하다고 결정하는 것에 응답하여, 비디오 디코더 (30)는 비디오 데이터에서, 비디오 블록들의 제 2 그룹에 대해, 제 1 플래그의 제 2 인스턴스를 수신할 수도 있다. 비디오 디코더 (30)는 제 1 플래그의 제 1 인스턴스가 제 2 값과 동일하다고 결정할 수도 있고, 비디오 디코더 (30)는 비디오 데이터에서, 비디오 블록들의 제 2 그룹에 대한 하나의 병합된 그룹에 대해, 제 2 플래그의 제 2 인스턴스를 수신하는 것을 스kip할 수도 있다. 제 2 플래그의 제 2 인스턴스를 수신하는 것을 스kip하기 위해, 비디오 디코더 (30)는 제 2 플래그의 제 2 인스턴스가 비디오 블록들의 제 2 그룹에 대한 하나의 병합된 그룹에 대해 비디오 데이터에서 제 1 값들을 수신하지 않으면서 제 1 값과 동일한 것을 추론할 수도 있다. 비디오 디코더 (30)는, 비디오 데이터에서, 비디오 블록들의 제 2 그룹에 대해 하나의 병합된 정보에 대한 필터 계수 정보를 수신할 수도 있으며, 비디오 블록들의 제 2 그룹에 대해 하나의 병합된 그룹에 대한 계수 필터 정보는 적어도 하나의 년 제로 계수를 포함한다.

[0162] 도 8은 본 개시물의 기법들에 따른 비디오 데이터를 인코딩하기 위한 기법을 도시하는 플로우 다이어그램이다.

도 8의 기법들은 예를 들어, 비디오 인코더 (20)에 의해 수행될 수도 있지만, 도 8의 기법들이 비디오 인코더의 임의의 특정 유형에 제한되지 않는다. 도 8의 예에서, 비디오 인코더 (20)는 비디오 데이터의 비디오 블록들의 그룹에 대해 필터들의 세트를 결정한다 (120). 비디오 블록들의 그룹은 예를 들어, 비디오 데이터의 슬라이스일 수도 있고, 또는 SPS 또는 PPS와 같은 소정의 파라미터 세트가 적용되는 블록들 모두일 수도 있다. 비디오 블록들의 그룹은 또한 LCU의 CU들과 같은, 더 큰 블록의 서브 블록들일 수도 있다.

[0163] 비디오 인코더 (20)는 비디오 블록들에 대해 필터 계수 정보의 세트를 결정한다 (122). 일부 예들에서, 필터 계수 정보의 세트들은 필터 계수 값들일 수도 있는 한편, 다른 예들에서, 필터 계수 정보의 세트들은 하나 이상의 예측 필터들의 필터 계수들과 필터 계수들의 세트에서의 필터들의 필터 계수들 사이의 차이 값들을 포함할 수도 있다. 하나 이상의 예측 필터들은 예를 들어, 하나 이상의 고정 필터들일 수도 있다.

[0164] 비디오 인코더 (20)는 복수의 클래스들을 위해 병합된 그룹들의 수를 결정한다 (124). 복수의 클래스들의 각각의 클래스는 메트릭들에 대한 대응 값들을 가지며, 복수의 클래스들의 각각의 클래스는 필터들의 세트로부터의 필터에 매핑된다. 각각의 병합된 그룹은 필터 계수 정보의 동일한 세트에 매핑되는 하나 이상의 클래스들을 포함한다. 비디오 인코더 (20)는 복수의 클래스들을 위해 병합된 그룹들의 수가, 복수의 클래스들이 하나의 병합된 그룹을 포함하는 것을 표시하는, 1과 동일하다고 결정한다 (126). 비디오 인코더 (20)

는 비디오 데이터에서의 포함을 위해, 제 1 값으로 설정된 제 1 플래그 (예를 들어, 상술한 *froceCoeff0* 플래그)를 생성한다 (128). 제 1 플래그에 대한 제 1 값은 병합된 그룹들 중 적어도 하나에 대한 필터 계수 정보가 비디오 데이터에서 코딩되지 않는 것을 표시하고, 제 1 플래그에 대한 제 2 값은 병합된 그룹들 모두에 대한 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시한다. 비디오 인코더 (20)는 비디오 데이터에서의 포함을 위해, 제 2 값으로 설정된 제 2 플래그 (상술한 *CodedVarBin* 플래그)를 생성한다 (130). 제 2 플래그에 대한 제 1 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 비디오 데이터에서 코딩되는 것을 표시하고, 제 2 플래그에 대한 제 2 값은 하나의 병합된 그룹에 매핑된 필터 계수 정보가 모두 제로 값들인 것을 표시한다.

[0165] 비디오 블록의 제 2 그룹에 대해, 비디오 인코더 (20)는 비디오 블록의 제 2 그룹에 대한 복수의 클래스들을 위해 병합된 그룹들의 수가 하나의 병합된 그룹과 동일하다고 결정하고, 비디오 블록들의 제 2 그룹에 대한 복수의 클래스들을 위해 병합된 그룹들의 수가 비디오 블록들의 제 2 그룹에 대한 하나의 병합된 그룹과 동일하다고 결정하는 것에 응답하여, 비디오 데이터에서의 포함을 위해, 비디오 블록들의 제 2 그룹에 대해, 비디오 블록들의 제 2 그룹에 대한 하나의 병합된 그룹에 대해 비디오 데이터에서 제 2 플래그의 제 2 인스턴스를 포함하지 않으면서, 제 2 값으로 설정된 제 1 플래그의 제 2 인스턴스를 생성할 수도 있다.

[0166] 도 9는 본 개시물의 기법들에 따른 비디오 데이터를 디코딩하기 위한 기법을 도시하는 플로우 다이어그램이다.

도 9의 기법들은 예를 들어, 비디오 디코더 (30)에 의해 수행될 수도 있지만, 도 9의 기법들은 비디오 디코더의 임의의 특정 유형에 제한되지 않는다. 도 9의 예에서, 비디오 디코더 (30)는 비디오 데이터의 루마 블록들의 그룹에 대해, 루마 블록들의 그룹에 대한 2 이상의 필터들의 세트를 결정한다 (140). 루마 블록들의 그룹으로부터의 루마 블록에 대해, 비디오 디코더 (30)는 루마 블록에 대해 하나 이상의 메트릭들을 결정한다 (142). 하나 이상의 메트릭들에 기초하여, 비디오 디코더 (30)는 필터들의 세트로부터 필터를 선택한다 (144). 비디오 디코더 (30)는 선택된 필터를 사용하여 루마 블록을 필터링한다 (146).

[0167] 비디오 디코더 (30) 는 루마 블록들의 그룹에 대해 필터들의 세트로부터 크로마 필터를 선택한다 (148). 비디오 디코더 (30) 는 크로마 필터를 사용하여 크로마 블록을 필터링한다 (150). 따라서, 루마 블록들에 대해 결정된 필터들의 세트로부터의 필터를 재사용하는 것에 의해, 비디오 디코더 (30) 는 크로마 블록들에 대한 필터 도출 프로세스를 잠재적으로 간략화한다. 일부 예들에서, 비디오 디코더 (30) 는 블록 단위로 크로마 필터링을 위해 사용될 루마 필터를 선택할 수도 있는 한편, 다른 예들에서, 비디오 디코더 (30) 는 슬라이스 단위, 또는 일부 다른 기반으로 크로마 필터링을 위해 루마 필터를 선택할 수도 있다. 일부 예들에서, 선택된 루마 필터는 크로마 블록을 필터링하는데 이용가능한 유일한 필터 (또는 유일한 년제로 필터) 일 수도 있다.

[0168] 일부 예들에서, 2 이상의 필터들의 세트에서의 각각의 필터는 하나 이상의 클래스들과 연관되고, 루마 블록들의 그룹에 대해 필터들의 세트로부터 크로마 필터를 선택하기 위해, 비디오 디코더 (30) 는 특정 클래스와 연관된 필터를 결정하고 크로마 필터로서 특정 클래스와 연관된 필터를 선택한다. 다른 예에서, 2 이상의 필터들의 세트에서의 각각의 필터는 필터 인덱스와 연관되고, 각각의 필터 인덱스는 하나 이상의 클래스들에 매핑한다. 루마 블록들의 그룹에 대해 필터들의 세트로부터 크로마 필터를 선택하기 위해, 비디오 디코더 (30) 는 크로마 필터로서 최소 필터 인덱스와 연관된 필터를 선택한다.

[0169] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 이 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 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독 가능 매체들, 또는 예를 들어 통신 프로토콜에 따라, 일 장소에서 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독 가능 매체는 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독 가능 저장 매체 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에 설명된 기법들의 구현을 위한 명령들, 코드, 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독 가능 매체를 포함할 수도 있다.

[0170] 한정이 아닌 예시로서, 이러한 컴퓨터 판독 가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장 디바이스, 자기 디스크 저장 디바이스 또는 다른 자기 저장 디바이스, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독 가능 매체라고 적절히 칭해진다. 예를 들어, 소프트웨어가 동축 케이블, 광섬유 케이블, 연선, 디지털가입자 회선 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 사용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신되면, 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 매체는 접속들, 반송파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않고, 대신에 비일시적인, 유형의 저장 매체들과 관련됨을 이해해야 한다. 본원에서 사용된 디스크 (disk) 와 디스크 (disc) 는, 컴팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크, 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 들은 통상 자기적으로 데이터를 재생하는 반면, 디스크 (disc) 들은 레이저들을 이용하여 광학적으로 데이터를 재생한다. 상기의 조합들이 또한, 컴퓨터 판독 가능 매체들의 범위 내에 포함되어야 한다.

[0171] 명령들은, 하나 이상의 디지털 신호 프로세서 (DSP) 들, 범용 마이크로프로세서들, 주문형 접적 회로 (ASIC) 들, 필드 프로그래머블 로직 어레이 (FPGA) 들, 또는 다른 등가의 접적 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 용어 "프로세서" 는 상기의 구조 또는 본원에 설명된 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 또한, 일부 양태들에서, 본원에 설명된 기능성은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있고, 또는 결합형 코덱에 통합될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0172] 본 개시물의 기법들은 무선 핸드셋, 접적 회로 (IC), 또는 IC 들의 세트 (예를 들어, 칩 세트) 를 포함하는 광범위한 디바이스들 또는 장치들로 구현될 수도 있다. 개시된 기술들을 수행하도록 구성된 디바이스들의 기능적 양태를 강조하기 위해 다양한 소자들, 모듈들, 또는 유닛들이 본 개시물에서 설명되었지만, 반드시 상이한 하드웨어 유닛들에 의해 실현될 필요는 없다. 오히려, 전술한 바와 같이 다양한 유닛들은 적합한 소프트웨어 및/또는 펌웨어와 관련되어, 전술된 하나 이상의 프로세서들을 포함하여 상호 동작적인 하드웨어 유닛들의

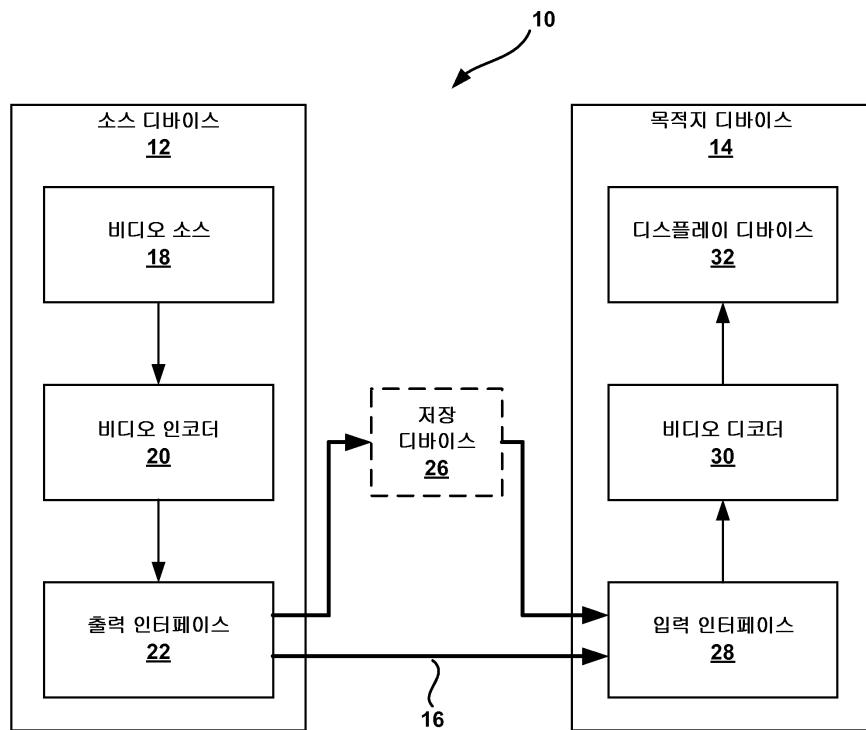
집합에 의해 제공되고 또는 코덱 하드웨어 유닛에 결합될 수도 있다.

[0173]

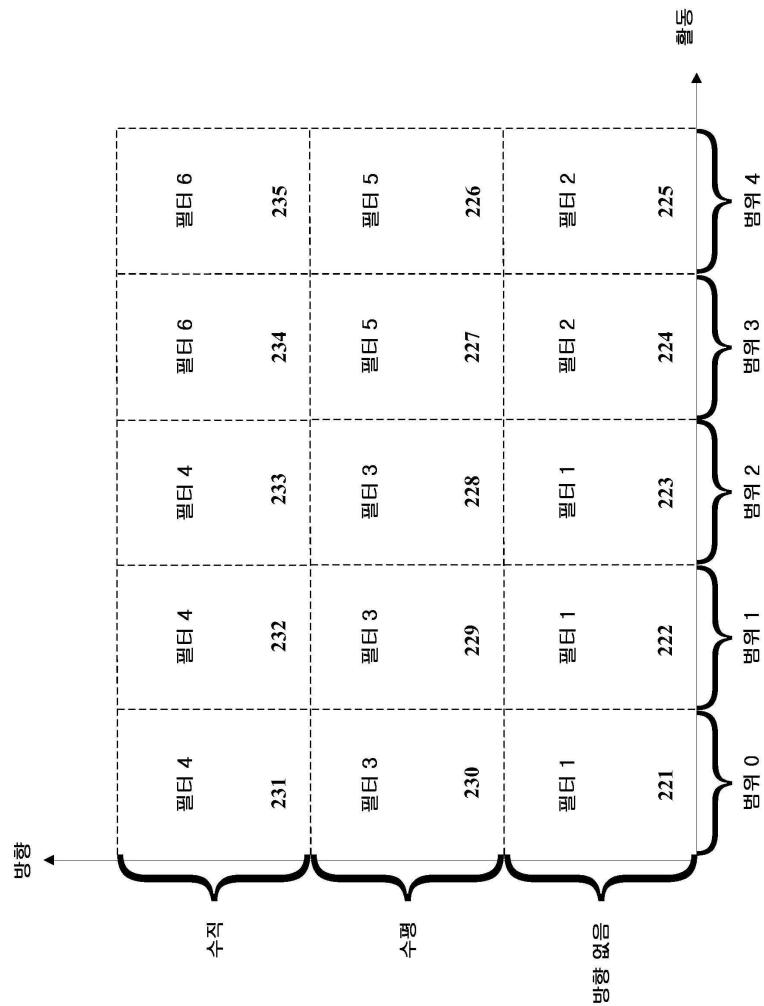
다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

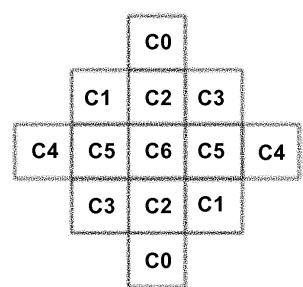
도면1



도면2



도면3a



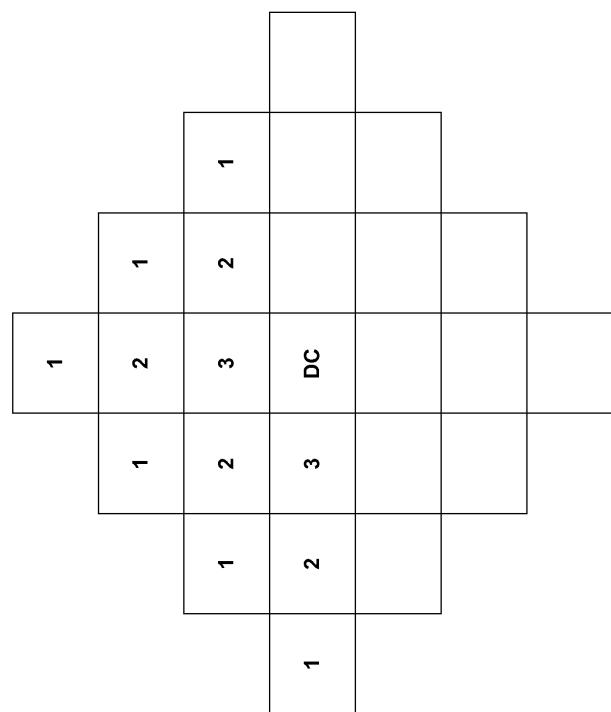
도면3b

		C0							
	C1	C2	C3						
	C4	C5	C6	C7	C8				
C9	C10	C11	C12	C11	C10	C9			
	C8	C7	C6	C5	C4				
	C3	C2	C1						
						C0			

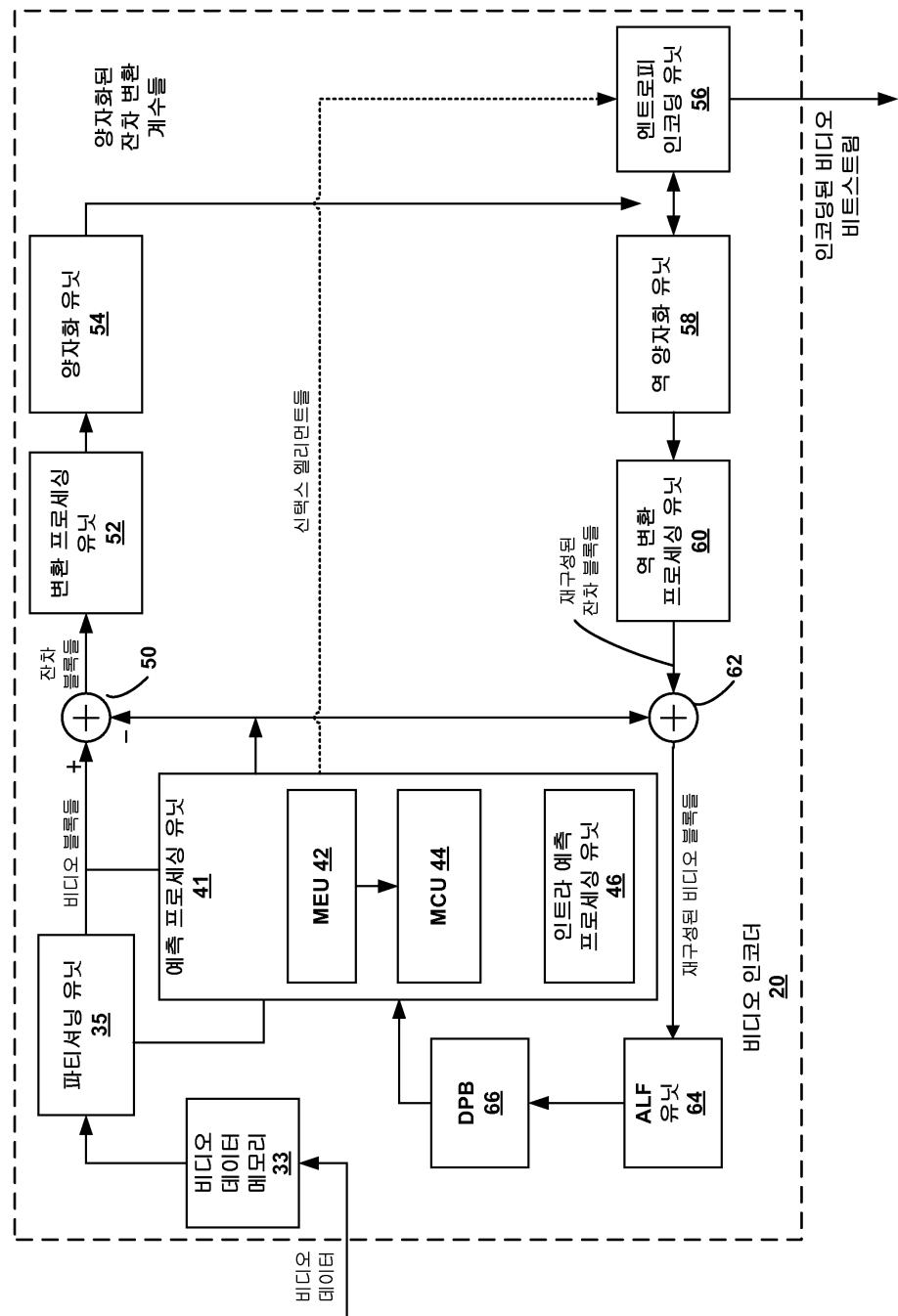
도면3c

	C0	C1	C2						
	C3	C4	C5	C6	C7				
	C8	C9	C10	C11	C12	C13	C14		
C15	C16	C17	C18	C19	C18	C17	C16	C15	
	C14	C13	C12	C11	C10	C9	C8		
	C7	C6	C5	C4	C3				
	C2	C1	C0						

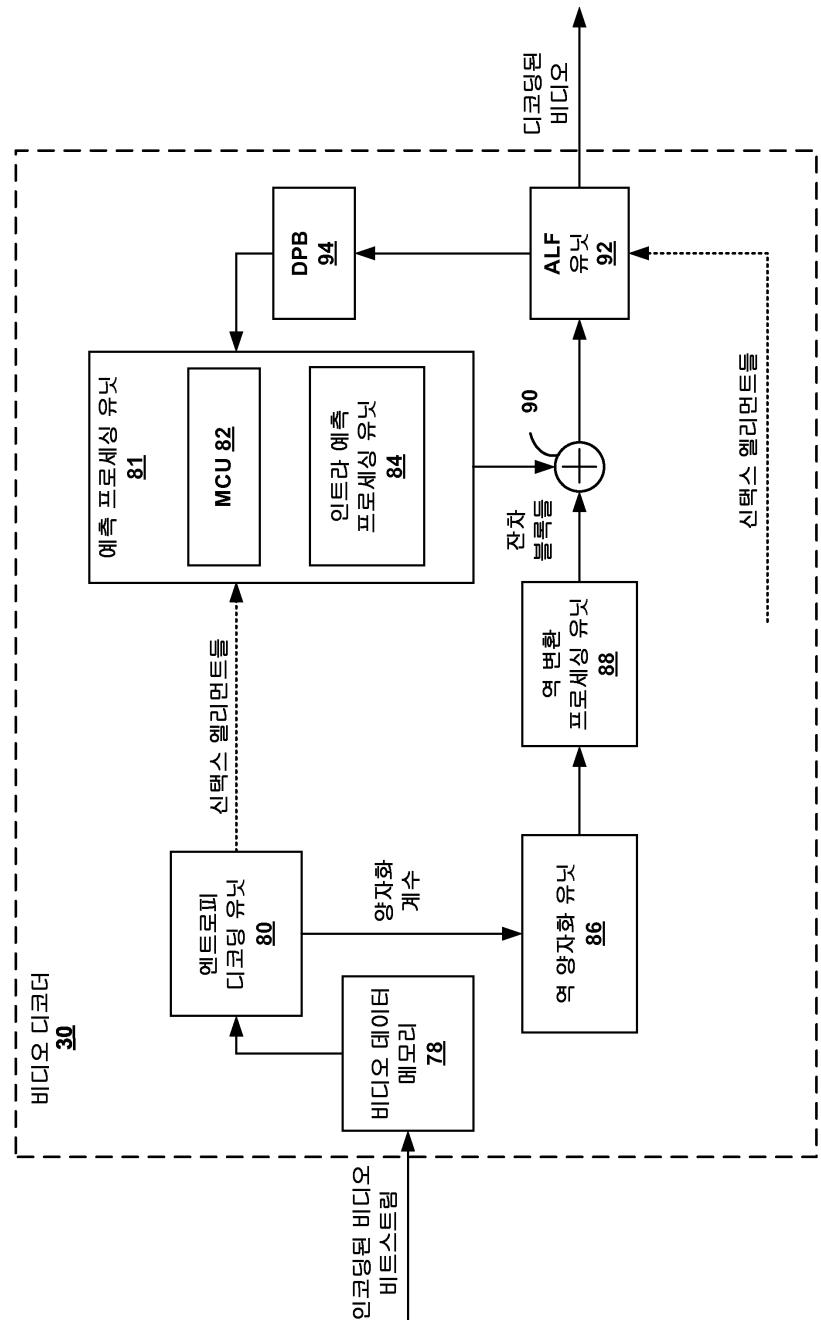
도면4



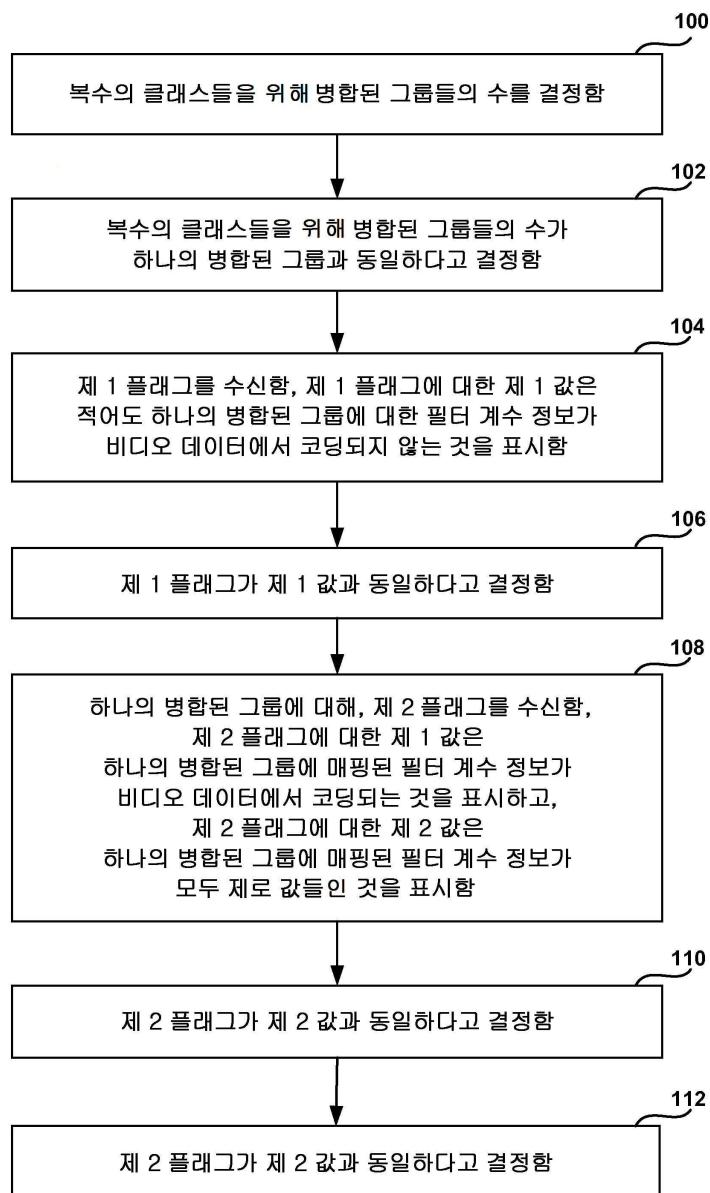
도면5



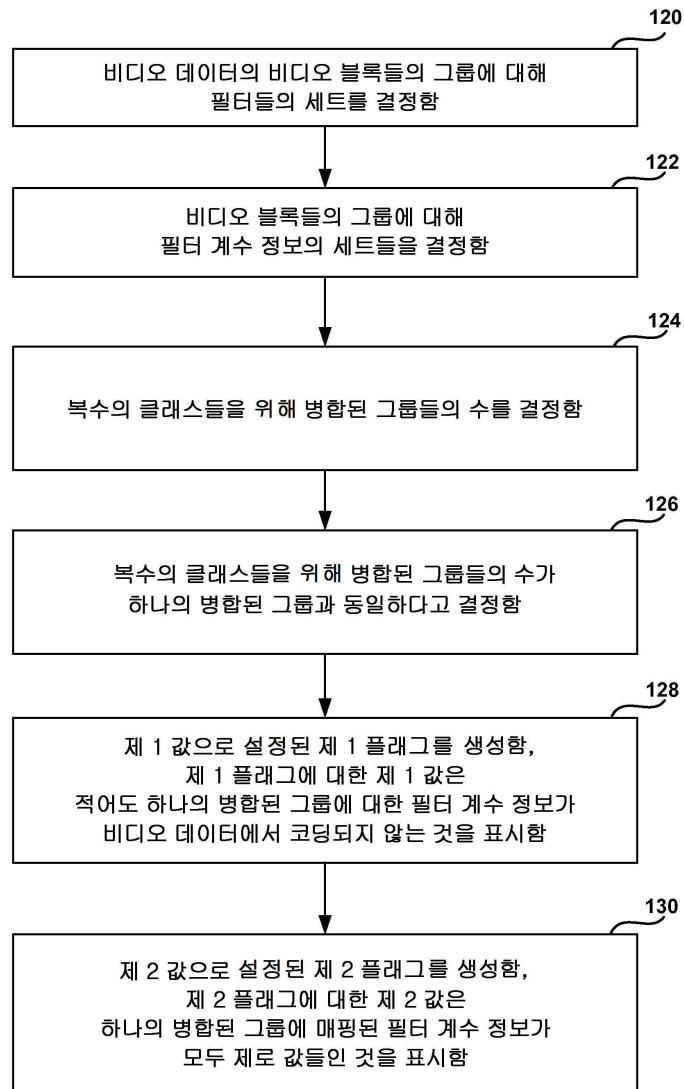
도면6



도면7



도면8



도면9

