



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0125959
(43) 공개일자 2024년08월20일

(51) 국제특허분류(Int. Cl.)
H04N 19/625 (2014.01) H04N 19/122 (2014.01)
H04N 19/124 (2014.01) H04N 19/129 (2014.01)
H04N 19/13 (2014.01) H04N 19/176 (2014.01)
H04N 19/18 (2014.01) H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/625 (2015.01)
H04N 19/122 (2015.01)
(21) 출원번호 10-2024-7023641
(22) 출원일자(국제) 2022년12월16일
심사청구일자 없음
(85) 번역문제출일자 2024년07월15일
(86) 국제출원번호 PCT/US2022/053150
(87) 국제공개번호 WO 2023/114468
국제공개일자 2023년06월22일
(30) 우선권주장
63/290,307 2021년12월16일 미국(US)
(뒷면에 계속)

(71) 출원인
베이징 다지아 인터넷 인포메이션 테크놀로지 컴퍼니 리미티드
중국 100085 베이징 하이덴 디스트릭트 시얼치 웨스트 로드 넘버 16 빌딩 12 플로어 8 룸 101
(72) 발명자
시우, 샤오위
미국 92122 캘리포니아 샌 디에고 400호 유니버시티 센터 레인 8910
옌, 닝
미국 92122 캘리포니아 샌 디에고 스위트 400 유니버시티 센터 레인 8910
(뒷면에 계속)
(74) 대리인
특허법인비엘티

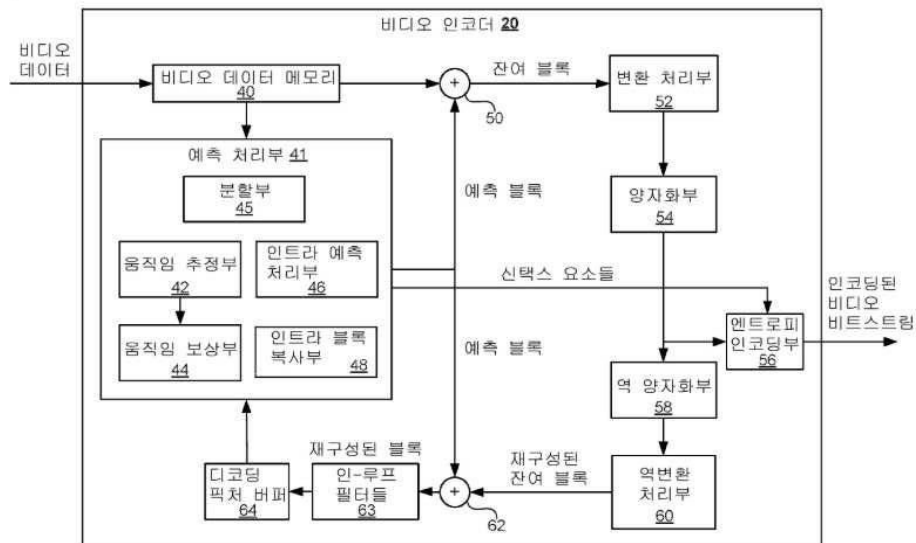
전체 청구항 수 : 총 41 항

(54) 발명의 명칭 블록 기반 비디오 코딩을 위한 부호 예측

(57) 요약

본 개시내용의 구현예들은 비디오 디코더 측에서의 변환 계수 부호 예측을 위한 비디오 디코딩 장치 및 방법을 제공한다. 상기 방법은, 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 포함하는 비트스트림을 수신하는 단계를 포함할 수 있다. 상기 방법은, 디코더의 프로세서에 의해, 비디오로부터 비디오 프레임의 변
(뒷면에 계속)

대표도



환 블록과 연관된 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트를 생성하는 단계를 더 포함할 수 있다. 상기 방법은, 프로세서에 의해, 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하는데 사용되는 하나 또는 그 이상의 컨텍스트에 기초하여 부호 시그널링 비트들의 시퀀스를 디코딩하여 각각의 후보 변환 계수들의 예측된 부호들의 정확성의 지시를 획득하는 단계를 더 포함할 수 있다. 상기 방법은, 프로세서에 의해, 예측된 부호들의 세트 및 디코딩된 부호 시그널링 비트들의 시퀀스에 기초하여 후보 변환 계수들의 세트에 대한 원래 부호들을 추정하는 단계를 더 포함할 수 있다.

(52) CPC특허분류

H04N 19/124 (2015.01)
H04N 19/129 (2015.01)
H04N 19/13 (2015.01)
H04N 19/176 (2015.01)
H04N 19/18 (2015.01)
H04N 19/70 (2015.01)

(72) 발명자

첸, 이-웬

미국 92122 캘리포니아 샌 디에고 400호 유니버시티 센터 레인 8910

쿠오, 체-웨이

미국 92122 캘리포니아 샌 디에고 스위트 400 유니버시티 센터 레인 8910

첸, 웨이

미국 92122 캘리포니아 샌 디에고 400호 유니버시티 센터 레인 8910

주, 홍-젠

미국 92122 캘리포니아 샌 디에고 스위트 400 유니버시티 센터 레인 8910

가오, 한

92122 캘리포니아 샌 디에고 400호 유니버시티 센터 레인 8910

왕, 상린

미국 92122 캘리포니아 샌 디에고 400호 유니버시티 센터 레인 8910

유, 빙

중국 100085 베이징 하이덴 디스트릭트 칭허 스트리트 시얼치 미들 로드 29번

(30) 우선권주장

PCT/US2022/040442	2022년08월16일	미국(US)
PCT/US2022/043607	2022년09월15일	미국(US)
PCT/US2022/049264	2022년11월08일	미국(US)

명세서

청구범위

청구항 1

변환 계수들의 부호 예측을 위한 비디오 디코딩 방법에 있어서,

후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 포함하는 비트스트림을 수신하는 단계;

디코더의 프로세서에 의해, 비디오로부터 비디오 프레임의 변환 블록과 연관된 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트를 생성하는 단계;

상기 프로세서에 의해, 상기 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하는데 사용되는 하나 또는 그 이상의 컨텍스트에 기초하여 상기 부호 시그널링 비트들의 시퀀스를 디코딩하여, 각각의 후보 변환 계수들의 예측된 부호들의 정확성(correctness)의 지시(indication)를 획득하는 단계;및

상기 프로세서에 의해, 상기 예측된 부호들의 세트 및 디코딩된 상기 부호 시그널링 비트들의 시퀀스에 기초하여 상기 후보 변환 계수들의 세트에 대한 원래 부호들을 추정하는 단계를 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 2

제 1 항에 있어서,

상기 예측된 부호들의 세트 및 상기 부호 시그널링 비트들의 시퀀스에 기초하여 후보 변환 계수들의 세트에 대한 원래 부호들을 추정하는 단계는,

각각의 부호 시그널링 비트들에 의해 가리키는 정확성에 따라 후보 변환 계수들의 예측된 부호들을 업데이트하여 후보 변환 계수들의 원래 부호들을 결정하는 단계를 더 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 3

제 1 항에 있어서,

각각의 부호 시그널링 비트를 디코딩하는 단계는,

상기 부호 시그널링 비트에 대응하는 후보 변환 계수의 크기(magnitude)에 기초하여 상기 부호 시그널링 비트를 엔트로피-인코딩하는데 사용되는 컨텍스트를 결정하는 단계를 더 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 4

제 3 항에 있어서,

상기 후보 변환 계수들의 크기(magnitude)는 복수의 크기 세그먼트들 내에 속하고, 복수의 컨텍스트들은 각각의 크기 세그먼트에 속하는 변환 계수들에 대한 부호 시그널링 비트를 엔트로피-인코딩하기 위하여 할당되며,

상기 부호 시그널링 비트를 엔트로피-인코딩하는데 사용되는 컨텍스트들을 결정하는 단계는,

상기 복수의 크기 세그먼트들 중에서 상기 부호 시그널링 비트에 대응하는 후보 변환 계수의 크기가 속하는 크기 세그먼트를 결정하는 단계; 및

상기 크기 세그먼트에 할당된 컨텍스트를 상기 부호 시그널링 비트를 엔트로피-인코딩하는데 사용되는 컨텍스트로 결정하는 단계를 더 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 5

제 4 항에 있어서,

상기 복수의 크기 세그먼트들 중 연속하는 2개의 크기 세그먼트 마다 임계값에 의해 분리되는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 6

제 5 항에 있어서,

상기 임계값은 상기 디코더에서 미리 프로그래밍된 고정값인 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 7

제 5 항에 있어서,

상기 임계값은 상기 인코더에 의해 결정되고, 상기 비트스트림에서 상기 디코더로 시그널링되는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 8

제 3 항에 있어서,

상기 비트스트림은 상기 후보 변환 계수들의 세트의 양자화된 레벨들의 인코딩된 데이터를 더 포함하며,

상기 비디오 디코딩 방법은,

역양자화 없이 상기 비트스트림으로부터 후보 변환 계수의 양자화된 레벨을 직접 파싱함으로써 후보 변환 계수의 크기를 결정하는 단계를 더 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 9

제3항에 있어서,

상기 비트스트림은 상기 후보 변환 계수들의 세트의 양자화된 레벨들의 인코딩된 데이터를 더 포함하며,

상기 비디오 디코딩 방법은,

상기 비트스트림으로부터 후보 변환 계수의 양자화된 레벨을 파싱하는 단계;

후보 변환 계수의 양자화된 레벨 및 후보 변환 계수의 전이 상태에 기초하여 후보 변환 계수의 양자화 인덱스를 결정하는 단계; 및

후보 변환 계수의 양자화 인덱스를 기반으로 후보 변환 계수의 크기를 결정하는 단계;를 더 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 10

제 1 항에 있어서,

상기 후보 변환 계수들의 세트는 크기에 따라 재정렬되는 변환 블록의 변환 계수들로부터 선택되며,

상기 비디오 디코딩 방법은,

상기 비트스트림으로부터 파싱된 상기 후보 변환 계수들의 세트를 재정렬하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 11

제 10 항에 있어서,

부호 시그널링 비트들을 엔트로피-인코딩하는데 사용되는 컨텍스트는 재정렬된 변환 계수들의 크기에 기초하여 결정되는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 12

제 10 항에 있어서,

부호 시그널링 비트들을 엔트로피-인코딩하는데 사용되는 컨텍스트는 재정렬하기 전의 변환 계수들의 크기에 기초하여 결정되는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 13

제 1 항에 있어서,

각각의 부호 시그널링 비트를 디코딩하는 단계는,

상기 부호 시그널링 비트에 대응하는 후보 변환 계수의 스캔 위치에 기초하여 상기 부호 시그널링 비트를 엔트로피-인코딩하는데 사용되는 컨텍스트를 결정하는 단계를 더 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 14

제 13항에 있어서,

상기 변환 계수들의 스캔 위치들은 복수의 그룹들로 분류되고, 복수의 컨텍스트들은 각각의 그룹들에 속하는 변환 계수들에 대한 부호 시그널링 비트를 엔트로피-인코딩하기 위하여 할당되며,

상기 부호 시그널링 비트를 엔트로피-인코딩하는데 사용되는 컨텍스트를 결정하는 단계는,

상기 복수의 그룹 중에서 상기 부호 시그널링 비트에 대응하는 후보 변환 계수의 스캔 위치가 속하는 그룹을 결정하는 단계; 및

상기 그룹에 할당된 컨텍스트를 상기 부호 시그널링 비트를 엔트로피-인코딩하는데 사용되는 컨텍스트로 결정하는 단계;를 더 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 15

제 1항에 있어서,

각각의 부호 시그널링 비트를 디코딩하는 단계는,

코딩 모드, 블록 크기 또는 상기 부호 시그널링 비트에 대응하는 후보 변환 계수의 컴포넌트 채널 정보에 기초하여 상기 부호 시그널링 비트를 엔트로피-인코딩하는데 사용되는 컨텍스트를 결정하는 단계를 더 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 16

제 1항에 있어서,

상기 비트스트림에서의 상기 변환 계수들의 세트는 복수의 서로 다른 변환 코어들을 적용하여 생성되며,

각각의 부호 시그널링 비트를 디코딩하는 단계는,

상기 부호 시그널링 비트에 대응하는 후보 변환 계수를 생성하기 위해 적용된 변환 코어에 기초하여 상기 부호 시그널링 비트를 엔트로피-인코딩하는데 사용되는 컨텍스트를 결정하는 단계를 더 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 17

제 16항에 있어서,

상기 변환 코어들은 다중 변환 선택(Multiple Transform Selection: MTS) 및 저주파 분리 불가능 변환(low-frequency non-separable transform: LFNST)을 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 18

제 1항에 있어서,

상기 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트를 생성하는 단계는,

상기 후보 변환 계수들의 세트에 대한 복수의 후보 가설들을 생성하는 단계; 및

상기 복수의 후보 가설에서 상기 후보 변환 계수들의 세트에 대한 예측 부호 세트로서 가설을 선택하는 단계;를 더 포함하는 것을 특징으로 하는,

비디오 디코딩 방법.

청구항 19

후보 변환 계수들의 세트 및 상기 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 포함하는 비트스트림을 저장하도록 구성된 메모리; 및

상기 메모리에 결합되고 제1항 내지 제18항 중 어느 한 항에 따른 변환 계수들의 부호 예측을 위한 비디오 디코딩 방법을 수행하도록 구성된 프로세서;를 포함하는 것을 특징으로 하는,

비디오 디코딩 장치.

청구항 20

프로세서에 의해 실행될 때, 프로세서로 하여금, 제1항 내지 제18항 중 어느 한 항에 따른 변환 계수의 부호 예측을 위한 비디오 디코딩 방법을 수행하도록 하는 명령어들을 그 안에 저장한 비-일시적 컴퓨터 판독가능 저장 매체로서,

상기 비트스트림은 상기 컴퓨터 판독 가능 저장 매체에 저장되는 것을 특징으로 하는,

비-일시적 컴퓨터 판독 가능 저장 매체.

청구항 21

후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 포함하는 비트스트림을 저장하며, 상기 비트스트림은 제1항 내지 제18항 중 어느 한 항에 따른 변환 계수들의 부호 예측을 위한 비디오 디코딩 방법에 의해 디코딩 가능한 것을 특징으로 하는,

비-일시적 컴퓨터 판독 가능 저장 매체.

청구항 22

변환 계수의 부호 예측을 위한 비디오 인코딩 방법에 있어서,

인코더의 프로세서에 의해, 비디오로부터 비디오 프레임의 변환 블록과 연관된 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트를 획득하는 단계;

상기 프로세서에 의해, 상기 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트를 결정하는 단계;

상기 프로세서에 의해, 상기 하나 또는 그 이상의 컨텍스트에 따라 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하는 단계; 및

후보 변환 계수의 양자화된 레벨들의 인코딩된 데이터 및 후보 변환 계수에 대한 인코딩된 부호 시그널링 비트들의 시퀀스를 비트스트림에 저장하는 단계;를 포함하되,

상기 부호 시그널링 비트들은 후보 변환 계수들의 세트에 대한 원래 부호들과 비교하여 각각의 후보 변환 계수들의 예측된 부호들의 정확성을 가리키는데 사용되는 것을 특징으로 하는,

비디오 인코딩 방법.

청구항 23

제 22항에 있어서,

상기 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트는 상기 부호 시그널링 비트에 대응하는 후보 변환 계수들의 세트의 크기들에 기초하여 결정되는 것을 특징으로 하는,

비디오 인코딩 방법.

청구항 24

제 23항에 있어서,

상기 후보 변환 계수들의 세트에 대한 상기 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 이상의 컨텍스트들을 결정하는 단계는,

상기 변환 계수들의 크기 범위를 복수의 크기 세그먼트로 분할하는 단계; 및

각각의 크기 세그먼트에 컨텍스트를 할당하여, 크기 세그먼트에 속하는 변환 계수들에 대한 부호 시그널링 비트들을 엔트로피-인코딩하는 단계;를 더 포함하는,

비디오 인코딩 방법.

청구항 25

제 24항에 있어서,

상기 복수의 크기 세그먼트들 중 연속하는 2개의 크기 세그먼트들 마다, 임계값에 의해 분리되는 것을 특징으로 하는,

비디오 인코딩 방법.

청구항 26

제 25항에 있어서,

상기 임계값은 상기 인코더에서 미리 프로그램된 고정값인 것을 특징으로 하는,

비디오 인코딩 방법.

청구항 27

제 25항에 있어서,

상기 임계값을 결정하는 단계; 및

상기 비트스트림에서 디코더에 상기 임계값을 시그널링하는 단계;를 더 포함하는 것을 특징으로 하는,

비디오 인코딩 방법.

청구항 28

제 23항에 있어서,
상기 변환 블록의 변환 계수를 각각의 크기에 따라 재정렬하는 단계; 및
재정렬된 변환 계수에서 상기 후보 변환 계수들의 세트를 선택하는 단계;를 더 포함하는 것을 특징으로 하는,
비디오 인코딩 방법.

청구항 29

제 28항에 있어서,
상기 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트는 상기 재정렬된 변환 계수들의 크기에 기초하여 결정되는 것을 특징으로 하는,
비디오 디코딩 방법.

청구항 30

제 28항에 있어서,
상기 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트는 재정렬하기 전의 변환 계수들의 크기에 기초하여 결정되는 것을 특징으로 하는,
비디오 디코딩 방법.

청구항 31

제 23항에 있어서,
상기 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트는 후보 변환 계수들의 양자화된 레벨들을 사용하여 결정되는 것을 특징으로 하는,
비디오 인코딩 방법.

청구항 32

제 23항에 있어서,
상기 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트는 후보 변환 계수들의 양자화 인덱스들을 사용하여 결정되며, 각각의 양자화 인덱스는 상기 후보 변환 계수의 양자화된 레벨 및 전이 상태에 기초하여 결정되는 것을 특징으로 하는,
비디오 인코딩 방법.

청구항 33

제 22항에 있어서,
상기 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트는 상기 후보 변환 계수들의 스캔 위치들에 기초하여 결정되는 것을 특징으로 하는,
비디오 인코딩 방법.

청구항 34

제 33항에 있어서,
상기 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트를 결정하는 단계는,
상기 후보 변환 계수들의 스캔 위치들을 복수의 그룹들로 분류하는 단계; 및
그룹에 속하는 변환 계수에 대한 부호 시그널링 비트를 엔트로피-인코딩하기 위해 각 그룹에 컨텍스트를 할당하는 단계;를 더 포함하는 것을 특징으로 하는,

비디오 인코딩 방법.

청구항 35

제 22항에 있어서,

상기 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트는 코딩 모드, 블록 크기 또는 각각의 후보 변환 계수의 컴포넌트 채널 정보에 기초하여 결정되는 것을 특징으로 하는,

비디오 인코딩 방법.

청구항 36

제 22항에 있어서,

상기 변환 계수들의 세트의 양자화된 레벨들은 복수의 서로 다른 변환 코어들을 적용함으로써 생성되며,

상기 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트는 후보 변환 계수들을 생성하기 위해 적용된 변환 코어들에 기초하여 결정되는 것을 특징으로 하는,

비디오 인코딩 방법.

청구항 37

제 36항에 있어서,

상기 변환 코어들은 다중 변환 선택(Multiple Transform Selection: MTS) 및 저주파 분리 불가능 변환(low-frequency non-separable transform: LFNST)을 포함하는 것을 특징으로 하는,

비디오 인코딩 방법.

청구항 38

제 22항에 있어서,

상기 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트를 생성하는 단계는,

상기 후보 변환 계수들의 세트에 대한 복수의 후보 가설들을 생성하는 단계; 및

상기 복수의 후보 가설에서 상기 후보 변환 계수 세트에 대한 예측 부호 세트로서 가설을 선택하는 단계;를 더 포함하는 것을 특징으로 하는,

비디오 인코딩 방법.

청구항 39

제22항 내지 제38항 중 어느 한 항에 따른 변환 계수들의 부호 예측을 위한 비디오 디코딩 방법을 수행하도록 구성된 프로세서; 및

상기 프로세서에 결합되고 상기 비트스트림을 저장하도록 구성된 메모리;를 포함하는 것을 특징으로 하는

비디오 인코딩 장치.

청구항 40

프로세서에 의해 실행될 때, 프로세서로 하여금, 제22항 내지 제38항 중 어느 한 항에 따른 변환 계수의 부호 예측을 위한 비디오 인코딩 방법을 수행하게 하는 명령어들을 그 안에 저장한 비-일시적 컴퓨터 판독가능 저장 매체로서,

상기 비트스트림은 상기 컴퓨터 판독 가능 저장 매체에 저장되는 것을 특징으로 하는,

비-일시적 컴퓨터 판독 가능 저장 매체.

청구항 41

제22항 내지 제38항 중 어느 한 항에 따른 변환 계수들의 부호 예측을 위한 비디오 인코딩 방법에 의해 인코딩된 비트스트림을 그 안에 저장하는,

비-일시적 컴퓨터 판독 가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 발명은 비디오 코딩 및 압축에 관한 것이다. 보다 구체적으로, 본 발명은 블록 기반 비디오 코딩에서 부호 예측을 위한 비디오 프로세싱 시스템 및 방법에 관한 것이다.

배경 기술

[0002] 디지털 비디오는 디지털 TV, 랩톱 또는 데스크톱 컴퓨터, 태블릿 컴퓨터, 디지털 카메라, 디지털 기록 장치, 디지털 미디어 플레이어, 비디오 게임 콘솔, 스마트 폰, 비디오 화상 회의 장치, 비디오 스트리밍 장치와 같은 다양한 전자 장치에 의해 지지될 수 있다. 전자 장치들은 통신 네트워크를 통해 디지털 비디오 데이터를 송수신하거나 다른 방법으로 통신하고, 및/또는 디지털 비디오 데이터를 저장 장치에 저장한다. 통신 네트워크의 제한된 대역폭 용량 및 저장 장치의 제한된 메모리 자원으로 인해, 비디오 코딩은 비디오 데이터가 통신되거나 저장되기 전에 하나 또는 그 이상의 비디오 코딩 표준에 따라 비디오 데이터를 압축하는 데 사용될 수 있다. 예를 들어, 비디오 코딩 표준은 다용도 비디오 코딩(Versatile Video Coding: VVC), 공동 탐사 테스트 모델 (Joint Exploration Test Model: JEM), 고효율 비디오 코딩(High-Efficiency Video Coding: HEVC/H.265), 고급 비디오 코딩(Advanced Video Coding: AVC/H.264), 동영상 전문가 그룹(Moving Picture Expert Group: MPEG) 코딩 등을 포함할 수 있다. 비디오 코딩은 일반적으로 비디오 데이터에 내재된 중복성을 이용하는 예측 방법(예를 들어, 인터-예측, 인트라-예측 등)을 이용한다. 비디오 코딩은 비디오 데이터를 더 낮은 비트 레이트를 사용하는 형태로 압축하는 것을 목표로 하며, 비디오 품질의 저하를 피하거나 최소화한다.

발명의 내용

해결하려는 과제

[0003] 본 개시의 구현 방식들은 사이드 윈도우를 이용한 양방향 필터링을 위한 비디오 프로세싱 방법을 제공한다.

과제의 해결 수단

[0004] 상기 비디오 디코딩 방법은 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 포함하는 비트스트림을 수신하는 단계를 포함할 수 있다. 상기 비디오 디코딩 방법은, 디코더의 프로세서에 의해, 비디오로부터 비디오 프레임의 변환 블록과 연관된 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트를 생성하는 단계를 더 포함할 수 있다. 상기 비디오 디코딩 방법은, 프로세서에 의해, 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하는데 사용되는 하나 또는 그 이상의 컨텍스트에 기초하여 부호 시그널링 비트들의 시퀀스를 디코딩하여 각각의 후보 변환 계수들의 예측된 부호들의 정확성의 지시를 획득하는 단계를 더 포함할 수 있다. 상기 비디오 디코딩 방법은, 프로세서에 의해, 예측된 부호들의 세트 및 디코딩된 부호 시그널링 비트들의 시퀀스에 기초하여 후보 변환 계수들의 세트에 대한 원래 부호들을 추정하는 단계를 추가로 포함할 수 있다.

[0005] 본 개시내용의 구현예들은 또한 비디오 디코딩 장치를 제공한다. 비디오 디코딩 장치는 위에서 요약된 바와 같은 변환 계수들의 부호 예측을 위한 비디오 디코딩 방법을 수행하도록 구성된 메모리를 포함할 수 있다.

[0006] 본 개시내용의 구현예들은 또한 프로세서에 의해 실행될 때, 프로세서가 위에서 요약된 바와 같은 변환 계수들의 부호 예측을 위한 비디오 디코딩 방법을 수행하도록 하는 명령어들을 저장한 비-일시적 컴퓨터 판독가능 저장 매체를 제공한다. 본 개시내용의 구현예들은 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 포함하는 비트스트림을 그 안에 저장한 비-일시적 컴퓨터 판독가능 저장 매체를 추가로 제공한다. 비트스트림은 위에서 요약된 바와 같이, 변환 계수들의 부호 예측을 위한 비디오 디코딩 방법에 의해 디코딩 가능하다.

[0007] 본 개시내용의 구현예들은 또한 변환 계수들의 부호 예측을 위한 비디오 인코딩 방법을 제공한다. 상기 비디오 인코딩 방법은, 인코더의 프로세서에 의해, 비디오로부터 비디오 프레임의 변환 블록과 연관된 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트를 생성하는 단계를 포함할 수 있다. 상기 비디오 인코딩 방법은, 프로

세서에 의해, 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트를 결정하는 단계를 더 포함할 수 있다. 상기 비디오 인코딩 방법은, 프로세서에 의해, 하나 또는 그 이상의 컨텍스트에 따른 부호 시그널링 비트들의 시퀀스를 결정하는 단계를 더 포함할 수 있다. 부호 시그널링 비트들은 후보 변환 계수들의 세트에 대한 원래 부호들과 비교하여 각각의 후보 변환 계수들의 예측된 부호들의 정확성을 가리키기 위해 하나 또는 그 이상의 컨텍스트를 사용하여 엔트로피-인코딩된다. 상기 비디오 인코딩 방법은, 후보 변환 계수들의 양자화된 레벨들의 인코딩된 데이터 및 후보 변환 계수들에 대한 부호 시그널링 비트들의 인코딩된 시퀀스를 비트스트림에 저장하는 단계를 추가로 포함할 수 있다.

[0008] 본 개시내용의 구현예들은 비디오 인코딩 장치를 제공한다. 비디오 디코딩 장치는 위에서 요약된 바와 같은 변환 계수들의 부호 예측을 위한 비디오 인코딩 방법을 수행하도록 구성된 메모리를 포함할 수 있다.

[0009] 본 개시내용의 구현예들은 프로세서에 의해 실행될 때, 프로세서가 위에서 요약된 바와 같은 변환 계수들의 부호 예측을 위한 비디오 인코딩 방법을 수행하도록 하는 명령어들을 저장한 비-일시적 컴퓨터 판독가능 저장 매체를 제공한다. 본 개시내용의 구현예들은 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 포함하는 비트스트림을 그 안에 저장한 비-일시적 컴퓨터 판독가능 저장 매체를 추가로 제공한다. 비트스트림은 위에서 요약한 바와 같이 변환 계수의 부호 예측을 위한 비디오 인코딩 방법에 의해 디코딩 가능하다.

[0010] 전술한 일반적인 설명과 이하의 상세한 설명은 모두 예시일 뿐이며, 본 개시에 제한되지 않음을 이해해야 한다.

도면의 간단한 설명

[0011] 본 명세서에 포함되고 본 명세서의 일부를 구성하는 첨부된 도면은 본 명세서와 일치하는 예들을 예시하며, 설명과 함께, 본 명세서의 원리를 설명하는 역할을 한다.

도 1은 본 개시내용의 일부 구현예들에 따른 비디오 블록들을 인코딩 및 디코딩하기 위한 예시적인 시스템을 나타내는 블록도이다.

도 2는 본 개시내용의 일부 구현예들에 따른 예시적인 비디오 인코더를 나타내는 블록도이다.

도 3은 본 개시내용의 일부 구현예들에 따른 예시적인 비디오 디코더를 나타내는 블록도이다.

도 4a 내지 도 4e는 본 개시내용의 일부 구현예들에 따른 프레임이 서로 다른 크기와 형상의 복수의 비디오 블록들로 재귀적으로 분할되는 방법을 예시하는 그래픽 표현이다.

도 5는 일부 실시예들에 따른 계수 그룹에서 변환 계수들의 좌측 상단 스캔 순서를 나타내는 그래픽 표현이다.

도 6은 일부 실시예들에 따른 저주파 분리 불가능 변환(LFNST: low-frequency non-separable transform) 프로세스를 나타내는 그래픽 표현이다.

도 7은 일부 실시예들에 따른 순방향 LFNST로 입력되는 주 변환 계수(primary transform coefficient)들의 좌측 상단 영역을 나타내는 그래픽 표현이다.

도 8은 일부 실시예들에 따른 인트라 템플릿 매칭의 검색 영역을 나타내는 그래픽 표현이다.

도 9는 일부 실시예들에 따른 부호 예측의 예시적인 프로세스를 나타내는 그래픽 표현이다.

도 10은 일부 실시예들에 따른 부호 예측을 위한 비용 함수의 계산을 나타내는 그래픽 표현이다.

도 11은 일부 실시예들에 따른 종속 스칼라 양자화에 사용되는 2개의 예시적인 스칼라 양자화기를 나타내는 그래픽 표현이다.

도 12a는 일부 실시예들에 따른 종속 스칼라 양자화에 사용되는 4개의 상태들을 갖는 상태 머신을 사용하여 상태 전이를 나타내는 그래픽 표현이다.

도 12b는 일부 실시예들에 따른 도 12a의 상태 전이에 따른 예시적인 양자화기 선택을 나타내는 표이다.

도 13은 본 개시내용의 일부 구현예들에 따른 블록 기반 비디오 코딩에서의 예시적인 부호 예측 프로세스를 나타내는 블록도이다.

도 14는 본 개시내용의 일부 구현예들에 따른 템플릿들의 선형 조합에 기초한 예시적인 가설 생성을 나타내는 그래픽 표현이다.

도 15a는 일부 실시예들에 따른 기존 부호 예측 방식의 예시적인 구현을 나타내는 그래픽 표현이다.

도 15b는 본 개시내용의 일부 구현예들에 따른 벡터-기반 부호 예측 방식의 예시적인 구현을 나타내는 그래픽 표현이다.

도 16a는 본 개시내용의 일부 구현예들에 따른 좌측 대각선 방향으로의 좌측 대각선 비용 함수의 예시적인 계산을 나타내는 그래픽 표현이다.

도 16b는 본 개시내용의 일부 구현예들에 따른 우측 대각선 방향으로의 우측 대각선 비용 함수의 예시적인 계산을 나타내는 그래픽 표현이다.

도 17은 본 개시내용의 일부 구현예들에 따른 현재 블록의 이웃하는 재구성된 샘플들 내에서 도미넌트 그래디언트 방향을 캡처하기 위한 방법의 흐름도이다.

도 18a는 본 개시내용의 일부 구현예들에 따른 비용 함수에 대한 샘플 추정 방향(sample extrapolation direction)의 그래디언트 기반 선택에서 예시적인 템플릿 샘플들 및 그래디언트 필터 윈도우들을 나타내는 그래픽 표현이다.

도 18b는 본 개시내용의 일부 구현예들에 따른 비용 함수에 대한 샘플 추정 방향의 그래디언트 기반 선택에서 예시적인 그래디언트 히스토그램(histogram of gradient: HoG)을 나타내는 그래픽 표현이다.

도 19는 일부 구현예들에 따른 변환 계수들의 부호를 예측하기 위한 부호 예측 영역을 나타내는 그래픽 표현이다.

도 20a는 본 개시내용의 일부 구현예들에 따른 변환 계수 부호 예측을 위한 예시적인 컨텍스트 모델링을 나타내는 그래픽 표현이다.

도 20b는 본 개시내용의 일부 구현예들에 따른 변환 계수 부호 예측을 위한 다른 예시적인 컨텍스트 모델링을 나타내는 그래픽 표현이다.

도 21은 본 개시내용의 일부 구현예들에 따른 블록 기반 비디오 코딩에서의 부호 예측을 위한 예시적인 방법의 흐름도이다.

도 22는 본 개시내용의 일부 구현예들에 따른 비디오 인코더에 의해 수행되는 변환 계수 부호 예측을 위한 예시적인 비디오 인코딩 방법의 흐름도이다.

도 23은 본 개시내용의 일부 구현예들에 따른 비디오 디코더에 의해 수행되는 변환 계수 부호 예측을 위한 예시적인 비디오 디코딩 방법의 흐름도이다.

도 24는 본 개시내용의 일부 구현예들에 따른 사용자 인터페이스와 결합된 컴퓨팅 환경을 나타내는 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0012] 이하 그 예가 첨부 도면에 도시되어 있는 특정 구현 방식들을 상세하게 설명한다. 다음의 상세한 설명에서, 본 명세서에 기재된 내용의 이해를 돕기 위하여 복수의 비-제한적인 특정 세부 사항이 설명된다. 그러나, 청구항의 범위를 벗어나지 않고 다양한 변경을 진행할 수 있고 이러한 특정 세부사항 없이 주제가 실시될 수 있다는 것은 당업자에게 자명한 것이다. 예를 들어, 본 명세서에 기재한 기술적 방안은 디지털 비디오 기능을 구비한 다양한 유형의 전자 장치에서 구현될 수 있다는 것은 당업자에게 자명한 것이다.

[0013] 본 개시내용의 설명, 청구항 및 첨부된 도면에 사용된 "제1", "제2" 등의 용어는 객체들을 구별하기 위해 사용되고, 임의의 특정 순서 또는 차례를 기술하기 위해 사용되지 않는다는 것이 예시되어야 한다. 이러한 방식으로 사용되는 데이터는 적절한 조건 하에서 상호 교환될 수 있음을 이해해야 하며, 이에 본 명세서에 기술된 본 개시내용의 실시예들은 첨부된 도면들에 나타나 있거나 본 개시에 기술된 것 이외의 순서로 구현될 수 있다.

[0014] 도 1은 본 개시내용의 일부 구현예들에 따른 비디오 블록들을 병렬로 인코딩 및 디코딩하기 위한 예시적인 시스템(10)을 도시하는 블록도이다. 도 1에 도시된 바와 같이, 시스템(10)은 목적지 장치(14)에 의해 나중에 디코딩될 비디오 데이터를 생성하고 인코딩하는 소스 장치(12)를 포함한다. 소스 장치(12) 및 목적지 장치(14)는 데스크톱 또는 랩탑 컴퓨터, 태블릿 컴퓨터, 스마트폰, 셋톱 박스, 디지털 텔레비전, 카메라, 디스플레이 장치, 디지털 미디어 플레이어, 비디오 게임 콘솔, 비디오 스트리밍 장치와 같은 다양한 전자 장치 중의 어느 한 항을 포함할 수 있다. 일부 구현예들에서, 소스 장치(12) 및 목적지 장치(14)는 무선 통신 기능을 구비할 수 있다.

- [0015] 일부 구현예들에서, 목적지 장치(14)는 링크(16)를 통하여 디코딩된 인코딩된 비디오 데이터를 수신할 수 있다. 링크(16)는 인코딩된 비디오 데이터를 소스 장치(12)로부터 목적지 장치(14)로 포워딩할 수 있는 임의의 유형의 통신 매체 또는 장치를 포함할 수 있다. 일 예에서, 링크(16)는 인코딩된 비디오 데이터를 소스 장치(12)가 목적지 장치(14)에 실시간으로 직접 전송할 수 있도록 하는 통신 매체일 수 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조되어 목적지 장치(14)로 전송될 수 있다. 통신 매체는 무선 주파수(RF) 스펙트럼 또는 하나 또는 그 이상의 물리적 전송 라인과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수 있다. 통신 매체는 근거리 네트워크(local area network), 광역 네트워크(wide-area network)와 같은 패킷-기반 네트워크(packet-based network) 또는 인터넷(Internet)과 같은 글로벌 네트워크(global network)의 일부분을 형성할 수 있다. 통신 매체는 라우터, 스위치, 기지국, 또는 소스 장치(12)로부터 목적지 장치(14)로의 통신을 용이하게 하는데 사용할 수 있는 임의의 다른 장비를 포함할 수 있다.
- [0016] 일부 다른 구현 방식들에서, 인코딩된 비디오 데이터는 출력 인터페이스(22)로부터 저장 장치(32)로 전송될 수 있다. 다음, 저장 장치(32) 내의 인코딩된 비디오 데이터는 목적지 장치(14)가 입력 인터페이스(28)를 통하여 액세스할 수 있다. 저장 장치(32)는 다양한 분산 또는 로컬 액세스 데이터 저장 매체(예하면, 하드 드라이브, Blu-ray 디스크, DVD(Digital Versatile Disks), CD-ROM(Compact Disc Read-Only Memories), 플래시 메모리, 휘발성 또는 비 휘발성 메모리), 또는 인코딩된 비디오 데이터를 저장하기에 적절한 기타 디지털 저장 매체 중의 임의의 하나를 포함할 수 있다. 추가 예에서, 저장 장치(32)는 소스 장치(12)에 의해 생성된 인코딩된 비디오 데이터를 저장할 수 있는 파일 서버(file server) 또는 다른 중간 저장 장치에 대응할 수 있다. 목적지 장치(14)는 스트리밍 또는 다운로드를 통해 저장 장치(32)로부터 저장된 비디오 데이터에 액세스할 수 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 인코딩된 비디오 데이터를 목적지 장치(14)로 전송할 수 있는 임의의 유형의 컴퓨터일 수 있다. 예시적인 파일 서버는 웹 서버(예 : 웹 사이트 용), 파일 전송 프로토콜(File Transfer Protocol:FTP) 서버, 네트워크 연결 스토리지(Network Attached Storage: NAS) 장치 또는 로컬 디스크 드라이브(local disk driver)를 포함할 수 있다. 목적지 장치(14)는 파일 서버에 저장된 인코딩된 비디오 데이터의 액세스에 적합한 무선 채널(예하면, Wi-Fi(Wireless Fidelity) 연결), 유선 연결(예하면, DSL(Digital Subscriber Line), 케이블 모뎀 등) 또는 임의의 조합을 포함하는 임의의 표준 데이터 연결을 통해 인코딩된 비디오 데이터에 액세스할 수 있다. 저장 장치(32)로부터의 인코딩된 비디오 데이터의 전송은 스트리밍 전송, 다운로드 전송 또는 그들의 조합일 수 있다.
- [0017] 도 1에 도시된 바와 같이, 소스 장치(12)는 비디오 소스(18), 비디오 인코더(20) 및 출력 인터페이스(22)를 포함한다. 비디오 소스(18)는 비디오 캡처 장치(예를 들어, 비디오 카메라), 이전에 캡처된 비디오를 포함하는 비디오 아카이브(video archive), 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스 및/또는 컴퓨터 그래픽 데이터를 소스 비디오로 생성하기 위한 컴퓨터 그래픽 시스템, 또는 이들의 조합을 포함할 수 있다. 일 예로, 비디오 소스(18)가 보안 감시 시스템의 비디오 카메라인 경우, 소스 장치(12) 및 목적지 장치(14)는 카메라 폰 또는 비디오 폰을 형성할 수 있다. 그러나, 본 출원에서 설명하는 구현 방식들은 일반적으로 비디오 코딩에 적용될 수 있는 것이고, 무선 및/또는 유선 애플리케이션에 적용될 수 있다.
- [0018] 캡처된, 사전에 캡처된 또는 컴퓨터에 의하여 생성된 비디오는 비디오 인코더(20)에 의해 인코딩될 수 있다. 인코딩된 비디오 데이터는 소스 장치(12)의 출력 인터페이스(22)를 통해 목적지 장치(14)로 직접 전송될 수 있다. 또한, 인코딩된 비디오 데이터는 (또는 대안 적으로) 저장 장치(32)에 저장되어 추후에 목적지 장치(14) 또는 다른 장치에 의하여 액세스되어 디코딩 및/또는 재생되도록 할 수 있다. 출력 인터페이스(22)는 모뎀 및/또는 송신기를 더 포함할 수 있다.
- [0019] 목적지 장치(14)는 입력 인터페이스(28), 비디오 디코더(30) 및 디스플레이 장치(34)를 포함한다. 입력 인터페이스(28)는 수신기 및/또는 모뎀을 포함 할 수 있으며, 링크(16)를 통해 인코딩된 비디오 데이터를 수신할 수 있다. 링크(16)를 통해 전송되거나 저장 장치(32) 상에 제공되는 인코딩된 비디오 데이터는 비디오 인코더(20)에 의해 생성되어 비디오 디코더(30)가 비디오 데이터를 디코딩하기 위하여 사용되는 다양한 신택스 요소들(syntax elements)을 포함할 수 있다. 이러한 신택스 요소들은 통신 매체를 통해 전송되거나, 저장 매체에 저장되거나 파일 서버에 저장되는 인코딩된 비디오 데이터에 포함될 수 있다.
- [0020] 일부 구현예들에서, 목적지 장치(14)는 통합 디스플레이 장치 및 목적지 장치(14)와 통신하도록 구성된 외부 디스플레이 장치일 수 있는 디스플레이 장치(34)를 포함 할 수 있다. 디스플레이 장치(34)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하며, 액정 디스플레이(LCD), 플라스마 디스플레이(plasma display), 유기 발광 다이오드(OLED) 디스플레이 또는 다른 유형의 디스플레이 장치와 같은 다양한 디스플레이 장치 중의 임의의 하

나를 포함할 수 있다.

[0021] 비디오 인코더(20) 및 비디오 디코더(30)는 VVC, HEVC, MPEG-4, Part 10, AVC(Advanced Video Coding) 또는 이러한 표준들의 확장과 같은 독점적 또는 산업 표준들에 따라 동작할 수 있다. 본 출원은 특정된 비디오 인코딩/디코딩 표준에 제한되지 않고, 다른 비디오 인코딩/디코딩 표준에 적용될 수 있음을 이해하여야 한다. 일반적으로, 소스 장치(12)의 비디오 인코더(20)는 이러한 현재 또는 미래의 표준 중 임의의 하나에 따라 비디오 데이터를 인코딩하도록 구성될 수 있음을 고려되어야 한다. 유사하게, 목적지 장치(14)의 비디오 디코더(30)는 이러한 현재 또는 미래의 표준 중 임의의 하나에 따라 비디오 데이터를 디코딩하도록 구성될 수 있음을 일반적으로 고려되어야 한다.

[0022] 비디오 인코더(20) 및 비디오 디코더(30) 각각은 하나 또는 그 이상의 마이크로 프로세서, 디지털 신호 프로세서(digital signal processor, DSP), 주문형 집적 회로(application specific integrated circuit: ASIC), 현장 프로그래머블 게이트 어레이(Field Programmable Gate Array: FPGA), 이산 로직과 같은 다양한 적합한 인코더 회로, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합과 같은 다양한 적합한 인코더 및/또는 디코더 회로 중 어느 하나로서 구현될 수 있다. 부분적으로 소프트웨어로 구현되는 경우, 전자 장치는 소프트웨어에 대한 명령어를 적절한 비-일시적 컴퓨터 판독 가능 매체에 저장하고 하나 또는 그 이상의 프로세서를 사용하여 하드웨어에서 명령어를 실행하여 본 개시에서 개시한 비디오 인코딩/디코딩 동작을 수행할 수 있다. 비디오 인코더(20) 및 비디오 디코더(30) 각각은 하나 또는 그 이상의 인코더 또는 디코더에 포함될 수 있으며, 이들은 결합된 인코더/디코더(CODEC)의 일부분으로 각각의 장치에 직접될 수 있다.

[0023] 도 2는 본 출원에서 설명한 일부 구현예들에 따른 예시적인 비디오 인코더(20)를 도시하는 블록도이다. 비디오 인코더(20)는 비디오 프레임 내의 비디오 블록들의 인트라 및 인터 예측 코딩을 수행할 수 있다. 인트라 예측 코딩은 공간적 예측에 의존하여 주어진 비디오 프레임 또는 픽처 내의 비디오 데이터의 공간 중복성을 줄이거나 제거할 수 있다. 인터 예측 코딩은 시간적 예측에 의존하여 비디오 시퀀스의 인접한 비디오 프레임 또는 픽처 내의 비디오 데이터의 시간 중복성을 줄이거나 제거할 수 있다. "프레임"이라는 용어는 비디오 코딩 분야에서 "이미지" 또는 "픽처"라는 용어에 대한 동의어로서 사용될 수 있음에 유의해야 한다.

[0024] 도 2에 도시된 바와 같이, 비디오 인코더(20)는 비디오 데이터 메모리(40), 예측 처리부(41), 디코딩 픽처 버퍼(DPB, 64), 가산기(50), 변환 처리부(52), 양자화부(54) 및 엔트로피 인코딩부(56)를 포함한다. 예측 처리부(41)는 움직임 추정부(42), 움직임 보상부(44), 분할부(45), 인트라 예측 처리부(46) 및 인트라 블록 복사(BC)부(48)를 포함한다. 일부 구현예들에서, 비디오 인코더(20)는 또한 비디오 블록 재구성을 위한 역 양자화부(58), 역변환 처리부(60) 및 가산기(62)를 포함한다. 디-블록킹 필터와 같은 인-루프 필터(63)는 가산기(62)와 DPB(64) 사이에 위치하여 블록 경계를 필터링하여 재구성된 비디오로부터 블록 아티팩트(block artifacts)를 제거할 수 있다. 디-블록킹 필터 이외에 SAO 필터 및/또는 적응적 인-루프 필터(ALF)와 같은 다른 인-루프 필터가 또한 사용되어 가산기(62)의 출력을 필터링할 수 있다. 일부 예들에서, 인-루프 필터들은 생략될 수 있고, 디코딩된 비디오 블록은 가산기(62)에 의해 DPB(64)에 직접 제공될 수 있다. 비디오 인코더(20)는 고정된 또는 프로그래밍 가능한 하드웨어 유닛의 형태로 구성되거나 하나 또는 그 이상의 도시된 고정된 또는 프로그래밍 가능한 하드웨어 장치에 분할될 수 있다.

[0025] 비디오 데이터 메모리(40)는 비디오 인코더(20)의 구성 요소들에 의해 인코딩된 비디오 데이터를 저장할 수 있다. 비디오 데이터 메모리(40) 중의 비디오 데이터는 예를 들어 도 1에 나타내는 바와 같이 비디오 소스(18)로부터 획득될 수 있다. DPB(64)는 비디오 인코더(20)가 비디오 데이터를(예를 들어, 인트라 또는 인터 예측 코딩 모드에서) 인코딩하기 위하여 사용되는 기준 비디오 데이터(예를 들어, 기준 프레임 또는 픽처)를 저장하는 버퍼이다. 비디오 데이터 메모리(40) 및 DPB(64)는 다양한 메모리 장치들 중의 어느 하나에 의해 형성될 수 있다. 다양한 예에서, 비디오 데이터 메모리(40)는 비디오 인코더(20)의 다른 구성 요소들과 함께 온칩(on-chip)되거나, 이러한 구성 요소들에 대하여 오프 칩(off-chip)될 수 있다.

[0026] 도 2에 도시된 바와 같이, 비디오 데이터를 수신 후, 예측 처리부(41) 내의 분할부(45)는 비디오 데이터를 비디오 블록들로 분할(partitioning)한다. 한편, 이러한 분할은 비디오 데이터와 연관된 쿼드-트리(QT) 구조와 같은 미리 설정된 분할 구조에 따라 비디오 프레임을 슬라이스(slice), 타일(tile)(예를 들어, 비디오 블록들의 세트들) 또는 다른 더 큰 코딩 단위(coding unit: CU)로 분할하는 것을 포함할 수 있다. 비디오 프레임은 샘플 값들을 갖는 샘플들의 2차원 어레이 또는 매트릭스이거나, 간주될 수 있다. 어레이 내의 샘플은 픽셀 또는 펠(pel)이라고도 할 수 있다. 어레이 또는 픽처의 수평 및 수직 방향(또는 축들)의 복수의 샘플은 비디오 프레임의 크기 및/또는 해상도를 정의한다. 비디오 프레임은, 예를 들어, QT 파티셔닝을 사용함으로써 복수의 비디오 블록

들로 분할될 수 있다. 비디오 블록은 비디오 프레임보다 작은 차원이지만 샘플 값을 갖는 샘플의 2차원 어레이 또는 매트릭스이거나, 간주될 수 있다. 비디오 블록의 수평 및 수직 방향(또는 축들)의 복수의 샘플은 비디오 블록의 크기를 정의한다. 비디오 블록은, 예를 들어, QT 파티셔닝, 이진 트리(Binary-Tree:BT) 파티셔닝, 트리 플 트리(Triple-Tree:TT) 파티셔닝 또는 이들의 임의의 조합을 반복적으로 사용함으로써 하나 또는 그 이상의 블록 파티셔닝 또는 서브-블록(다시 블록을 형성할 수 있음)으로 더 파티셔닝될 수 있다. 본 명세서에서 사용되는 "블록" 또는 "비디오 블록"이라는 용어는 프레임 또는 픽처의 일부, 특히 직사각형(정사각형 또는 비-정사각형) 부분일 수 있음을 유의해야 한다. 예를 들어, HEVC 및 VVC를 참조하면, 블록 또는 비디오 블록은 CTU, CU, 예측 유닛(Prediction Unit:PU) 또는 변환 유닛(Transform Unit:TU)이거나 이에 대응할 수 있고, 및/또는 대응하는 블록, 예를 들어, 코딩 트리 블록(Coding Tree Block:CTB), 코딩 블록(Coding Block:CB), 예측 블록(Prediction Block:PB) 또는 변환 블록(Transform Block:TB)일 수 있다. 대안적으로 또는 추가적으로, 블록 또는 비디오 블록은 CTB, CB, PB, TB 등의 서브-블록이거나 이에 대응할 수 있다.

[0027] 예측 처리부(41)는 오차 결과(예를 들어, 코딩 레이트 및 왜곡 레벨)에 기초하여 현재 비디오 블록에 대한 복수의 인트라 예측 코딩 모드들 중 하나 또는 복수의 인터 예측 코딩 모드들 중 하나와 같은 복수의 가능한 예측 코딩 모드들 중 하나를 선택할 수 있다. 예측 처리부(41)는 처리 결과인 인트라 또는 인터 예측 코딩된 블록(예를 들어, 예측 블록)을 가산기(50)에 제공하여 잔여 블록을 생성하고, 가산기(62)에 제공하여 인코딩된 블록을 재구성하여 추후에 기준 프레임의 일부로서 사용할 수 있다. 한편, 예측 처리부(41)는 움직임 벡터, 인트라-모드 지시자(indicator), 분할 정보(partition information) 및 기타 이러한 신택스 정보와 같은 신택스 요소들을 엔트로피 인코딩부(56)에 제공할 수 있다.

[0028] 현재 비디오 블록에 대한 적절한 인트라 예측 코딩 모드를 선택하기 위하여, 예측 처리부(41) 내의 인트라 예측 처리부(46)는 코딩된 현재 블록과 동일한 프레임 내의 하나 또는 그 이상의 인접한 블록에 대하여 현재 비디오 블록의 인트라 예측 코딩을 수행하여 공간 예측을 제공할 수 있다. 예측 처리부(41) 내의 움직임 추정부(42) 및 움직임 보상부(44)는 하나 또는 그 이상의 기준 프레임 중의 하나 또는 그 이상의 예측 블록에 대한 현재 비디오 블록의 인터 예측 코딩을 수행하여 시간적 예측을 제공할 수 있다. 비디오 인코더(20)는 복수의 코딩 패스(coding passes)를 수행하여, 예를 들어, 비디오 데이터의 각 블록에 대한 적절한 코딩 모드를 선택할 수 있다.

[0029] 일부 구현예들에서, 움직임 추정부(42)는 비디오 프레임 시퀀스 내의 미리 결정된 패턴에 따라 움직임 벡터를 생성하는 것을 통하여 현재 비디오 프레임에 대한 인터 예측 모드를 확정하며, 상기 움직임 벡터는 기준 프레임 내의 예측 블록에 대한 현재 비디오 프레임 내의 비디오 블록의 변위를 가리킬 수 있다. 움직임 추정부(42)에 의해 수행되는 움직임 추정은 비디오 블록들의 움직임을 추정할 수 있는 움직임 벡터들을 생성하는 과정이다. 예를 들어, 움직임 벡터는, 기준 프레임 내의 예측 블록에 대한 현재 비디오 프레임 또는 픽처 내의 비디오 블록의 변위를 가리킬 수 있다. 미리 정해진 패턴은 시퀀스의 비디오 프레임을 P 프레임 아니면 B 프레임으로 지정할 수 있다. 인트라 BC 부(48)는 인터 예측을 위한 움직임 추정부(42)의 움직임 벡터들의 결과와 유사한 방식으로 인트라 BC 코딩을 위한 벡터들(예를 들어 블록 벡터들)을 확정할 수 있거나, 움직임 추정부(42)를 이용하여 블록 벡터를 확정할 수 있다.

[0030] 픽셀 차이의 측면에서, 비디오 블록에 대한 예측 블록은 코딩된 비디오 블록과 밀접하게 일치하는 것으로 간주되는 기준 프레임의 블록 또는 기준 블록일 수도 있고, 이에 상응할 수도 있으며, 픽셀 차이는 절대차의 합(sum of absolute difference: SAD), 제곱차의 합(sum of square difference: SSD), 또는 기타 차이 메트릭에 의해 결정될 수 있다. 일부 구현예들에서, 비디오 인코더(20)는 DPB(64)에 저장된 기준 프레임들의 서브-정수 픽셀 위치들에 대한 값들을 계산할 수 있다. 예를 들어, 비디오 인코더(20)는 기준 프레임의 1/4 픽셀 위치, 1/8 픽셀 위치 또는 기타 분수 픽셀 위치의 값들을 보간할 수 있다. 따라서, 움직임 추정부(42)는 전체 픽셀 위치 및 분수 픽셀 위치에 대한 움직임 검색을 수행하고 분수 픽셀 정밀도로 움직임 벡터를 출력할 수 있다.

[0031] 움직임 추정부(42)는 기준 프레임의 예측 블록의 위치를 비교하여 인터 예측 코딩된 프레임 내의 비디오 블록의 움직임 벡터를 계산하되, 상기 기준 프레임은 DPB(64)에 저장된 하나 또는 그 이상의 기준 프레임을 각각 식별하는 제 1 기준 프레임 리스트(List 0) 또는 제 2 기준 프레임 리스트(List 1)로부터 선택될 수 있다. 움직임 추정부(42)는 계산된 움직임 벡터를 움직임 보상부(44)로 전송하고, 다음 엔트로피 인코딩부(56)로 전송한다.

[0032] 움직임 보상부(44)에 의해 수행되는 움직임 보상은, 움직임 추정부(42)가 결정한 움직임 벡터에 기초하여 예측 블록을 페치(fetch)하거나 생성하는 것을 포함할 수 있다. 현재 비디오 블록의 움직임 벡터를 수신하면, 움직임 보상부(44)는 기준 프레임 리스트들 중의 하나에서 움직임 벡터가 가리키는 예측 블록을 찾고, DPB(64)로부터 예측 블록을 검색하고, 예측 블록을 가산기(50)로 전달할 수 있다. 다음, 가산기(50)는 코딩되고 있는 현재 비

디오 블록의 픽셀 값으로부터 움직임 보상부(44)가 제공한 예측 블록의 픽셀 값을 감산하는 것을 통하여 잔여 블록의 픽셀 차이값을 형성할 수 있다. 잔여 비디오 블록을 형성하는 픽셀 차이값은 루마 또는 크로마 차이 성분 또는 양자를 포함할 수 있다. 또한, 움직임 보상부(44)는 비디오 디코더(30)가 비디오 프레임의 비디오 블록을 디코딩할 경우에 사용하는 비디오 프레임의 비디오 블록과 연관된 신텍스 요소들을 생성할 수 있다. 신텍스 요소들은, 예를 들어, 예측 블록을 식별하기 위하여 사용되는 움직임 벡터들을 정의하는 신텍스 요소들, 예측 모드를 가리키는 임의의 플래그, 또는 본 명세서에서 설명된 임의의 다른 신텍스 정보를 포함할 수 있다. 움직임 추정부(42) 및 움직임 보상부(44)는 함께 통합될 수 있음을 유의하며, 이는 도 2에서 개념적 목적을 위해 별도로 예시된다.

[0033] 일부 구현예들에서, 인트라 BC부(48)는 움직임 추정부(42) 및 움직임 보상부(44)와 관련하여 위에서 설명한 것과 유사한 방식으로 벡터들을 생성하고 예측 블록들을 페치(fetch)할 수 있지만, 이러한 예측 블록들은 코딩되고 있는 현재 블록과 같은 프레임 내에 있으며, 벡터들은 움직임 벡터가 아닌 블록 벡터라 한다. 구체적으로, 인트라 BC부(48)는 현재 블록을 인코딩하기 위하여 사용하는 인트라 예측 모드를 결정할 수 있다. 일부 예들에서, 인트라 BC부(48)는 예를 들어 각각의 인코딩 패스 동안 다양한 인트라 예측 모드들을 사용하여 현재 블록을 인코딩할 수 있고, 레이트-왜곡 분석을 통해 그들의 성능을 테스트할 수 있다. 다음으로, 인트라 BC부(48)는 다양한 테스트된 인트라 예측 모드들 중에서 적절한 인트라 예측 모드를 선택하여 사용하고 그에 따른 인트라 모드 식별자를 생성할 수 있다. 예를 들어, 인트라 BC부(48)는 테스트된 다양한 인트라 예측 모드들에 대한 레이트-왜곡 분석들(rate-distortion analysis)을 이용하여 레이트-왜곡 값들(rate-distortion value)을 계산할 수 있고, 테스트된 모드들 중 최적의 레이트-왜곡 특성을 갖는 인트라-예측 모드를 적절한 인트라 예측 모드로 선택하여 사용할 수 있다. 일반적으로, 레이트 왜곡 분석은 인코딩된 블록과 인코딩된 블록을 생성하기 위하여 인코딩되는 원래의 인코딩되지 않은 블록 사이의 왜곡(또는 오차)의 양 및 인코딩된 블록을 생성하기 위한 비트율(즉, 비트 수)을 결정한다. 인트라 BC부(48)는 다양한 인코딩된 블록들에 대한 왜곡(distortion) 및 레이트로부터 비율을 계산하여 어느 인트라 예측 모드가 블록에 대하여 최적의 레이트-왜곡 값을 가리키는지를 결정할 수 있다.

[0034] 다른 예들에서, 인트라 BC부(48)는 전체적으로 또는 부분적으로 움직임 추정부(42) 및 움직임 보상부(44)를 사용하여 본 명세서에서 설명된 구현 방식들에 따른 인트라 BC 예측을 위한 기능들을 수행할 수 있다. 두 경우 모두, 인트라 블록 복사(block copy)에 대하여, 예측 블록은 절대 차이의 합(SAD), 제곱 차이(SSD)의 합 또는 다른 차이 메트릭에 의해 결정되는 픽셀 차이 면에서 코딩될 블록과 밀접하게 매칭되는 것으로 간주되는 블록일 수 있으며, 예측 블록의 식별은 서브-정수 픽셀 위치에 대한 값의 계산을 포함할 수 있다.

[0035] 예측 블록이 인트라 예측에 따라 동일한 프레임 또는 인터 예측에 따라 다른 프레임으로부터 생성되는 것과 관련 없이, 비디오 인코더(20)는 코딩되는 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이값들을 형성하는 것을 통하여, 잔여 블록을 형성할 수 있다. 잔여 블록을 형성하는 픽셀 차이값들은 루마(luma) 및 크로마(chroma) 성분 차이를 모두 포함할 수 있다.

[0036] 위에서 설명한 움직임 추정부(42) 및 움직임 보상부(44)에 의해 수행되는 인터 예측 또는 인트라 BC부(48)에 의해 수행되는 인트라 블록 복사 예측에 대한 대안으로서, 인트라 예측 처리부(46)는 현재 비디오 블록을 인트라 예측할 수 있다. 구체적으로, 인트라 예측 처리부(46)는 현재 블록을 인코딩하기 위해 사용할 인트라 예측 모드를 결정할 수 있다. 예를 들어, 인트라 예측 처리부(46)는 예를 들어 각각의 인코딩 패스 동안 다양한 인트라 예측 모드들을 사용하여 현재 블록을 인코딩할 수 있고, 인트라 예측 처리부(46)(또는 일부 예들에서 모드 선택부)는 테스트된 인트라 예측 모드들로부터 적절한 인트라 예측 모드를 선택하여 사용할 수 있다. 인트라 예측 처리부(46)는 블록에 대해 선택한 인트라 예측 모드를 가리키는 정보를 엔트로피 인코딩부(56)에 제공할 수 있다. 엔트로피 인코딩부(56)는 선택된 인트라-예측 모드를 가리키는 정보를 비트스트림에 인코딩할 수 있다.

[0037] 예측 처리부(41)가 인터 예측 또는 인트라 예측을 통해 현재 비디오 블록에 대한 예측 블록을 결정한 후, 가산기(50)는 현재 비디오 블록으로부터 예측 블록을 감산하여 잔여 블록(residual block)을 형성한다. 잔여 블록 내의 잔여 비디오 데이터는 하나 또는 그 이상의 변환 단위(TU)에 포함될 수 있으며, 변환 처리부(52)에 제공된다. 변환 처리부(52)는 이산 코사인 변환(Discrete Cosine Transform:DCT) 또는 개념적으로 유사한 변환과 같은 변환을 사용하여 잔여 비디오 데이터를 변환 계수로 변환한다.

[0038] 변환 처리부(52)는 얻은 변환 계수들을 양자화부(54)에 전송할 수 있다. 양자화부(54)는 변환 계수들을 양자화하여 비트 레이트를 더 감소할 수 있다. 또한, 양자화 프로세스는 계수들의 일부 또는 전부와 관련된 비트 깊이(bit depth)를 감소할 수 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 수정될 수 있다. 일부 예들에

서, 양자화부(54)는 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수 있다. 또는, 엔트로피 인코딩부(56)가 상기 스캔을 수행할 수 있다.

[0039] 양자화한 다음, 엔트로피 인코딩부(56)는 엔트로피 인코딩 기술을 사용, 예를 들어, 컨텍스트 적응형 가변 길이 코딩(context adaptive variable length coding: CAVLC), 컨텍스트 적응형 이진 산술 코딩(context adaptive binary arithmetic coding: CABAC), 선택스 기반 컨텍스트 적응형 이진 산술 코딩(syntax-based context-adaptive binary arithmetic coding: SBAC), 확률 간격 분할 엔트로피(probability interval partitioning entropy: PIPE) 코딩 또는 다른 엔트로피 인코딩 방법 또는 기술을 사용하여 양자화된 변환 계수들을 비디오 비트스트림으로 인코딩할 수 있다. 그 후, 인코딩된 비트스트림은 도 1에 도시된 바와 같이 비디오 디코더(30)로 전송되거나, 나중에 비디오 디코더(30)로 전송되거나 비디오 디코더(30)에 의해 검색될 수 있도록 도 1에 도시된 바와 같이 저장 장치(32)에 보관될 수 있다. 엔트로피 인코딩부(56)는 또한 엔트로피 인코딩 기술을 사용하여 코딩되고 있는 현재 비디오 프레임에 사용되는 움직임 벡터 및 다른 선택스 요소를 인코딩할 수 있다.

[0040] 역 양자화부(58) 및 역변환 처리부(60)는 다른 비디오 블록들의 예측을 위한 기준 블록을 생성하기 위하여, 각각 역 양자화 및 역변환을 적용하여 픽셀 도메인에서 잔여 블록을 재구성할 수 있다. 재구성된 잔여 블록(residual block)이 생성될 수 있다. 위에서 설명한 바와 같이, 움직임 보상부(44)는 DPB(64)에 저장된 프레임들의 하나 또는 그 이상의 기준 블록으로부터 움직임 보상된 예측 블록을 생성할 수 있다. 움직임 보상부(44)는 또한 하나 또는 그 이상의 보간 필터를 예측 블록에 적용하여 움직임 추정에 사용하기 위한 서브-정수 픽셀 값들을 산출할 수 있다.

[0041] 가산기(62)는 재구성된 잔여 블록과 움직임 보상부(44)에 의해 생성된 움직임 보상된 예측 블록을 가산하여 DPB(64)에 저장하기 위한 기준 블록을 생성할 수 있다. 다음, 기준 블록은 인트라 BC부(48), 움직임 추정부(42), 움직임 보상부(44)에 의하여 예측 블록으로 사용되어 후속 비디오 프레임 내의 다른 비디오 블록을 인트라 예측할 수 있다.

[0042] 도 3은 본 출원의 일부 구현예들에 따른 예시적인 비디오 디코더(30)를 도시하는 블록도이다. 비디오 디코더(30)는 비디오 데이터 메모리(79), 엔트로피 디코딩부(80), 예측 처리부(81), 역 양자화부(86), 역변환 처리부(88), 가산기(90) 및 DPB(92)를 포함한다. 예측 처리부(81)는 움직임 보상부(82), 인트라 예측부(84) 및 인트라 BC부(85)를 더 포함한다. 비디오 디코더(30)는 일반적으로 도 2와 관련하여 비디오 인코더(20)에 대하여 위에서 설명한 인코딩 프로세스와 반대인 디코딩 프로세스를 수행할 수 있다. 예를 들어, 움직임 보상부(82)는 엔트로피 디코딩부(80)로부터 수신된 움직임 벡터들에 기초하여 예측 데이터를 생성할 수 있고, 인트라 예측부(84)는 엔트로피 디코딩부(80)로부터 수신된 인트라 예측 모드 지시자들에 기초하여 예측 데이터를 생성할 수 있다.

[0043] 일부 예들에서, 비디오 디코더(30)의 구성들은 본 출원의 구현 방식들을 구현하도록 기능들을 수행할 수 있다. 또한, 일부 예들에서, 본 개시내용의 구현 방식들을 구현하도록 비디오 디코더(30)의 구성들을 하나 또는 그 이상으로 분할되어 구현될 수 있다. 예를 들어, 인트라 BC부(85)는 본 출원의 구현 방식들을 단독으로 수행하거나, 움직임 보상부(82), 인트라 예측부(84) 및 엔트로피 디코딩부(80)와 같은 비디오 디코더(30)의 다른 부분들과 조합하여 수행할 수 있다. 일부 예들에서, 비디오 디코더(30)는 인트라 BC부(85)를 포함하지 않을 수 있고, 인트라 BC부(85)의 기능은 움직임 보상부(82)와 같은 예측 처리부(81)의 다른 부분들에 의하여 수행할 수 있다.

[0044] 비디오 데이터 메모리(79)는 인코딩된 비디오 비트스트림과 같은 비디오 디코더(30)의 다른 부분들에 의해 디코딩된 비디오 데이터를 저장할 수 있다. 비디오 데이터 메모리(79)에 저장된 비디오 데이터는 예를 들어 저장 장치(32)로부터 획득하거나, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통하여 획득하거나, 또는 물리적 데이터 저장 매체(예를 들어: 플래시 드라이브 또는 하드 디스크)에 액세스하는 것을 통하여 카메라와 같은 로컬 비디오 소스로부터 획득할 수 있다. 비디오 데이터 메모리(79)는 인코딩된 비디오 비트스트림의 인코딩된 비디오 데이터를 저장하는 코딩 픽처 버퍼(Coded Picture Buffer: CPB)를 포함할 수 있다. 비디오 디코더(30)의 디코딩 픽처 버퍼(Decoded picture buffer: DPB)(92)는 (예를 들어, 인트라 또는 인트라 예측 코딩 모드에서) 비디오 디코더(30)가 비디오 데이터를 디코딩하는데 사용하기 위한 기준 비디오 데이터를 저장할 수 있다. 비디오 데이터 메모리(79) 및 DPB(92)는 SDRAM, MRAM, RRAM을 포함하는 DRAM과 같은 다양한 메모리 장치 또는 다른 유형의 메모리 장치 중의 어느 하나로 형성될 수 있다. 설명을 위해, 비디오 데이터 메모리(79) 및 DPB(92)는 도 3에서 비디오 디코더(30)의 두 개의 별개의 부분으로 도시되었다. 그러나, 비디오 데이터 메모리(79) 및 DPB(92)는 동일한 메모리 장치 또는 별도의 메모리 장치에 의해 제공될 수 있다는 것은 당업자에게 자명한 것이다. 일부 예들에서, 비디오 데이터 메모리(79)는 비디오 디코더(30)의 다른 부분들과 함께 온-칩(on-chip)일 수 있

거나, 이러한 부분들과 오프-칩(off-chip)일 수 있다.

- [0045] 디코딩 프로세스 동안, 비디오 디코더(30)는 인코딩된 비디오 프레임의 비디오 블록들 및 연관된 선택스 요소들을 가리키는 인코딩된 비디오 비트스트림을 수신한다. 비디오 디코더(30)는 비디오 프레임 레벨 및/또는 비디오 블록 레벨에서 선택스 요소들을 수신할 수 있다. 비디오 디코더(30)의 엔트로피 디코딩부(80)는 엔트로피 디코딩 기술을 이용하여 비트스트림을 디코딩함으로써 양자화된 계수들, 움직임 벡터들 또는 인트라-예측 모드 지시자들 및 기타 선택스 요소들을 획득할 수 있다. 엔트로피 디코딩부(80)는 움직임 벡터 또는 인트라-예측 모드 지시자들 및 다른 선택스 요소들을 예측 처리부(81)로 전달한다.
- [0046] 비디오 프레임이 인트라 예측 코딩(예를 들어, I) 프레임으로 코딩되거나 다른 유형의 프레임에서 인트라 코딩된 예측 블록에 대해 코딩 될 경우, 예측 처리부(81)의 인트라 예측부(84)는 시그널링된 인트라 예측 모드 및 현재 프레임의 이전에 디코딩된 블록들로부터의 기준 데이터에 기초하여 현재 비디오 프레임의 비디오 블록에 대한 예측 데이터를 생성할 수 있다.
- [0047] 비디오 프레임이 인터 예측 코딩(즉, B 또는 P)된 프레임으로 코딩 될 경우, 예측 처리부(81)의 움직임 보상부(82)는 엔트로피 디코딩부(80)로부터 수신된 움직임 벡터들 및 다른 선택스 요소들에 기초하여 현재 비디오 프레임의 비디오 블록에 대한 하나 또는 그 이상의 예측 블록을 생성한다. 각각의 예측 블록은 기준 프레임 리스트들 중 하나의 기준 프레임 리스트 내의 기준 프레임으로부터 생성될 수 있다. 비디오 디코더(30)는 DPB(92)에 저장된 기준 프레임들에 기초하여 디폴트 구성 기술들을 사용하여 기준 프레임 리스트들(예를 들어, List0 및 List1)을 생성할 수 있다.
- [0048] 일부 예들에서, 비디오 블록이 본 명세서에서 설명한 인트라 BC 모드에 따라 코딩 될 경우, 예측 처리부(81)의 인트라 BC부(85)는 엔트로피 디코딩부(80)로부터 수신된 블록 벡터들 및 다른 선택스 요소들에 기초하여 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 비디오 인코더(20)에 의해 처리된 현재 비디오 블록과 동일한 픽처의 재구성된 영역 내에 있을 수 있다.
- [0049] 움직임 보상부(82) 및/또는 인트라 BC부(85)는 움직임 벡터들 및 다른 선택스 요소들을 파싱하여 현재 비디오 프레임의 비디오 블록에 대한 예측 정보를 확정한 다음, 예측 정보를 사용하여 디코딩되는 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 움직임 보상부(82)는 수신된 선택스 요소들 중 일부를 사용하여 비디오 프레임의 비디오 블록들의 코딩에 사용되는 예측 모드(예를 들어, 인트라 또는 인터 예측), 인터 예측 프레임 유형(예를 들어, B 또는 P), 프레임에 대한 기준 프레임 리스트들 중 하나 또는 그 이상에 대한 구성 정보, 프레임의 각 인터 예측 코딩된 비디오 블록에 대한 움직임 벡터들, 프레임의 각 인터 예측 코딩된 비디오 블록에 대한 인터 예측 상태, 및 현재 비디오 프레임의 비디오 블록들을 디코딩하기 위한 기타 정보를 확정한 수 있다.
- [0050] 유사하게, 인트라 BC부(85)는 수신된 선택스 요소들 중 일부(예를 들어 플래그)를 사용하여, 현재 비디오 블록이 인트라 BC 모드를 사용하여 예측되었으므로, 프레임의 어느 비디오 블록들이 재구성된 영역 내에 있으며 DPB(92)에 저장되어야 하는 구성 정보(construction information), 프레임의 각 인트라 BC 예측된 비디오 블록에 대한 블록 벡터들, 프레임의 각 인트라 BC 예측된 비디오 블록에 대한 인트라 BC 예측 상태 및 현재 비디오 프레임 내의 비디오 블록들을 디코딩하기 위한 기타 정보를 확정한 수 있다.
- [0051] 움직임 보상부(82)는 또한 기준 블록들의 서브-정수 픽셀에 대한 보간 값들을 계산하기 위해 비디오 블록들의 인코딩 동안 비디오 인코더(20)가 사용하는 보간 필터들을 사용하여 보간을 수행할 수 있다. 이 경우, 움직임 보상부(82)는 수신된 선택스 요소들로부터 비디오 인코더(20)가 사용하는 보간 필터들을 확정하고, 확정한 보간 필터들을 사용하여 예측 블록들을 생성할 수 있다.
- [0052] 역 양자화부(86)는 비디오 프레임 내의 각 비디오 블록에 대해 비디오 인코더(20)가 계산한 동일한 양자화 파라미터를 사용하여, 비트스트림 내에 제공되며 엔트로피 디코딩부(80)에 의하여 디코딩된 변환 계수들을 역 양자화하여 양자화 정도를 확정한다. 역 변환 처리부(88)는 픽셀 도메인에서 잔여 블록을 재구성하기 위하여, 역 변환(예를 들어, 역 DCT, 역 정수 변환 또는 개념적으로 유사한 역 변환 프로세스)을 변환 계수에 적용한다.
- [0053] 움직임 보상부(82) 또는 인트라 BC부(85)에서 벡터들 및 기타 선택스 요소들에 기초하여 현재 비디오 블록에 대한 예측 블록을 생성한 다음, 가산기(90)는 역 변환 처리부(88)에서 제공한 잔여 블록 및 움직임 보상부(82) 및 인트라 BC부(85)에 의해 생성된 대응하는 예측 블록을 가산하는 것을 통하여 현재 비디오 블록에 대한 디코딩된 비디오 블록을 재구성한다. 디코딩된 비디오 블록은 현재 비디오 블록에 대한 재구성된 블록(reconstructed block)으로도 지칭될 수 있다. 디-블로킹 필터, SAO 필터, 및/또는 ALF와 같은 인-루프 필터(91)가 디코딩된 비

디오 블록을 추가로 처리하기 위해 가산기(90)와 DPB(92) 사이에 위치할 수 있다. 일부 예들에서, 인-루프 필터(91)는 생략될 수 있고, 디코딩된 비디오 블록은 가산기(90)에 의해 DPB(92)에 직접 제공될 수 있다. 주어진 프레임 내의 디코딩된 비디오 블록들은 다음 비디오 블록들의 후속 움직임 보상에 사용되는 기준 프레임들을 저장한 DPB(92)에 저장된다. DPB(92) 또는 DPB(92)와 분리된 메모리 장치는 또한 디코딩된 비디오를 저장하여 도 1의 디스플레이 장치(34)와 같은 디스플레이 장치 상에 나중에 표시할 수 있다.

[0054] 전형적인 비디오 코딩 프로세스(예를 들어, 비디오 인코딩 프로세스 및 비디오 디코딩 프로세스를 포함함)에서, 비디오 시퀀스는 일반적으로 프레임들 또는 픽처들의 소정 순서의 세트를 포함한다. 각 프레임은 SL, SCb 및 SCr로 표시되는 3 개의 샘플 행렬들을 포함할 수 있다. SL은 루마 샘플들의 2 차원 행렬이다. SCb는 Cb 크로마 샘플들의 2 차원 행렬이다. SCr은 Cr 크로마 샘플들의 2 차원 행렬이다. 다른 경우에, 프레임은 단색일 수 있으므로, 루마 샘플들의 2 차원 행렬 하나만 포함할 수 있다.

[0055] 도 4a에 도시된 바와 같이, 비디오 인코더(20)(또는 보다 구체적으로, 분할 유닛(45))는 먼저 프레임들CTU의 세트로 분할함으로써 프레임의 인코딩된 표현(representation)을 생성한다. 비디오 프레임은 좌측에서 우측 및 상측에서 하측으로 래스터 스캔 순서(raster scan order)로 순차적으로 배열된 정수 수량의 CTU들을 포함할 수 있다. 각 CTU는 가장 큰 논리 코딩 단위이고, CTU의 너비와 높이는비디오 인코더(20)에 의해 시퀀스 파라미터 세트(sequence parameter set)로서시그널링되어, 비디오 시퀀스의 모든 CTU들은 128 X 128, 64 X 64, 32 X 32 및 16 X 16 중 하나인 동일한 크기를 갖는다. 그러나, 본 발명에서의 CTU는 반드시 특정 크기로 제한되는 것은 아니라는 점에 유의해야 한다. 도 4b에 도시된 바와 같이, 각 CTU는 루마 샘플들의 하나의CTB, 크로마 샘플들의 두 개의 대응하는 코딩 트리 블록들 및 코딩 트리 블록들의 샘플들을 코딩하기 위하여 사용되는 선택 요소들을 포함할 수 있다. 선택 요소들은 인터 또는 인트라 예측, 인트라 예측 모드, 움직임 벡터, 및 기타 파라미터들을 포함하여 코딩된 픽셀 블록의 상이한 유형의 단위의 특성 및 비디오 시퀀스가 비디오 디코더(30)에서 재구성될 수 있는 방법을 설명한다. 모노크롬 픽처(monochrome picture)들 또는 3개의 각각의 컬러 평면을 갖는 픽처들에서, CTU는 코딩 트리 블록의 샘플들을 코딩하기 위하여 사용되는 단일 코딩 트리 블록 및 선택 요소들을 포함할 수 있다. 코딩 트리 블록은 샘플들의 N×N 블록일 수 있다.

[0056] 더 나은 성능을 달성하기 위해, 비디오 인코더 (20) 는 CTU의 코딩 트리 블록들 상에서 이진-트리 분할(binary-tree partitioning), 삼진- 트리 분할(ternary-tree partitioning), 쿼드-트리 분할(quad-tree partitioning) 또는 이들의 조합과 같은 트리 분할을 재귀적으로 수행하고 CTU를 더 작은 코딩 단위(CU)로 분할할 수 있다. 도 4c에 도시된 바와 같이, 64×64 CTU(400)는 먼저 각각 32×32의 블록 크기를 갖는 4개의 더 작은 CU들로 분할된다. 4개의 더 작은 CU들 중, CU(410)와 CU(420)는 블록 크기에 따라 각각 16×16의 4개의 CU로 분할된다. 2개의 16×16 CU들(430 및 440)은 블록 크기에 따라 각각 8×8의 4개의 CU들로 더 분할된다. 도 4d는 도 4c에 도시된 바와 같은 CTU(400)의 분할 프로세스의 최종 결과를 나타내는 쿼드-트리 데이터 구조를 도시하며, 쿼드 트리의 각각의 리프 노드는 32×32부터 8×8까지 범위의 대응하는 크기의 하나의 CU에 대응한다. 도 4b에 도시된 CTU와 유사하게, 각각의 CU는 루마 샘플들의 코딩 블록(CB) 및 동일한 크기의 프레임의 크로마 샘플들의 2개의 대응하는 코딩 블록들 및 코딩 블록의 샘플을 코딩하기 위하여 사용되는 선택 요소들을 포함할 수 있다. 모노크롬 픽처들 또는 3개의 각각의 컬러 평면을 갖는 픽처들에서, CU는 단일 코딩 블록 및 코딩 블록의 샘플들을 코딩하는 데 사용되는 선택 구조들을 포함할 수 있다. 도 4c 및 4d에 도시된 쿼드-트리 분할은 단지 예시를 위한 것이며, 하나의 CTU는 쿼드/삼진/이진-트리 분할들을 기반으로 하는 다양한 로컬 특성에 적응하기 위해 CU들로 분할될 수 있다는 점에 유의해야 한다. 멀티-타입 트리 구조에서 하나의 CTU는 쿼드-트리 구조로 분할되고, 각각의 쿼드-트리 리프 CU는 이진 및 삼진-트리 구조로 더 분할될 수 있다. 도 4e에 도시된 바와 같이, 폭이 W이고 높이가 H인 코딩 블록에는 여러 가지 가능한 분할 타입, 즉, 쿼터너리(ternary) 분할, 수직 이진 분할, 수평 이진 분할, 수직 삼진 분할, 수직 확장 삼진 분할, 수평 삼진 분할 및 수평 확장 삼진 분할 등이 있다.

[0057] 일부 구현들에서, 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 혹은 그 이상의 M×N 예측 블록들(PB)로 더 분할할 수 있다. PB는 동일한 예측(인터 또는 인트라 예측)이 적용되는 샘플들의 직사각형(정사각형 또는 비-정사각형) 블록을 포함할 수 있다. CU의 PU는 루마 샘플들의PB, 크로마 샘플들의 2개의 대응하는 PB, 및PB를 예측하기 위하여 사용되는 선택 요소들을 포함할 수 있다. 모노크롬 픽처들 또는 3개의 각각의 컬러 평면을 갖는 픽처들에서, PU는 단일 PB 및 PB를 예측하기 위하여 사용되는 선택 구조들을 포함할 수 있다. 비디오 인코더 (20)는 CU의 각 PU의 루마 PB, Cb PB 및 Cr PB들에 대한 예측 루마 블록, 예측 Cb 블록 및 예측 Cr 블록들을 생성할 수 있다.

[0058] 비디오 인코더(20)는 PU에 대한 예측 블록들을 생성하기 위해 인트라 예측 또는 인터 예측을 사용할 수 있다. 비디오 인코더(20)가 PU의 예측 블록들을 생성하기 위해 인트라 예측을 사용하는 경우, 비디오 인코더(20)는 PU

와 연관된 프레임의 디코딩된 샘플들에 기반하여 PU의 예측 블록들을 생성할 수 있다. 비디오 인코더(20)가 PU의 예측 블록들을 생성하기 위해 인터 예측을 사용하는 경우, 비디오 인코더(20)는 PU와 연관된 프레임이 아닌 하나 혹은 그 이상의 프레임들의 디코딩된 샘플들에 기반하여 PU의 예측 블록들을 생성할 수 있다.

[0059] 비디오 인코더(20)가 CU의 하나 혹은 그 이상의 PU들의 예측 루마 블록, Cb 블록, 및 Cr 블록들을 생성한 후, 비디오 인코더(20)는, CU의 오리지널 루마 코딩 블록으로부터 CU의 예측 루마 블록들을 감산함으로써 CU에 대한 루마 잔여 블록을 생성하여 CU의 루마 잔여 블록의 각 샘플이 CU의 예측 루마 블록들 중 하나의 루마 샘플과 CU의 오리지널 루마 코딩 블록의 대응하는 샘플 사이의 차이를 나타내도록 할 수 있다. 이와 유사하게, 비디오 인코더(20)는, CU에 대한 Cb 잔여 블록 및 Cr 잔여 블록을 각각 생성하여 CU의 Cb 잔여 블록의 각 샘플이 CU의 예측 Cb 블록들 중 하나의 Cb 샘플과 CU의 오리지널 Cb 코딩 블록의 대응하는 샘플 사이의 차이를 나타내고 CU의 Cr 잔여 블록의 각 샘플이 CU의 예측 Cr 블록들 중 하나의 Cr 샘플과 CU의 오리지널 Cr 코딩 블록의 대응하는 샘플 사이의 차이를 나타내도록 할 수 있다.

[0060] 추가로, 도 4c에 도시된 바와 같이, 비디오 인코더(20)는 CU의 루마, Cb 및 Cr 잔여 블록들을 각각 하나 혹은 그 이상의 루마, Cb 및 Cr 변환 블록들로 분해하기 위해 쿼드-트리 분할을 사용할 수 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 직사각형(정사각형 또는 비 정사각형) 블록을 포함할 수 있다. CU의 변환 단위(TU)는 루마 샘플들의 변환 블록, 크로마 샘플들의 2개의 대응하는 변환 블록들 및 변환 블록 샘플들을 변환하기 위하여 사용되는 선택스 요소들을 포함할 수 있다. 따라서, CU의 각 TU는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수 있다. 일부 예들에서, TU와 연관된 루마 변환 블록은 CU의 루마 잔여 블록의 서브-블록일 수 있다. Cb 변환 블록은 CU의 Cb 잔여 블록의 서브-블록일 수 있다. Cr 변환 블록은 CU의 Cr 잔여 블록의 서브-블록일 수 있다. 모노크롬 픽처들 또는 3개의 각각의 컬러 평면을 갖는 픽처들에서, TU는 단일 변환 블록 및 변환 블록의 샘플들을 예측하기 위하여 사용하는 선택스 구조들을 포함할 수 있다.

[0061] 비디오 인코더(20)는 TU에 대한 루마 계수 블록을 생성하기 위해 TU의 루마 변환 블록에 하나 혹은 그 이상의 변환들을 적용할 수 있다. 계수 블록은 변환 계수들의 2차원 어레이일 수 있다. 변환 계수는 스칼라 양(scalar quantity)일 수 있다. 비디오 인코더(20)는 TU에 대한 Cb 계수 블록을 생성하기 위해 TU의 Cb 변환 블록에 하나 혹은 그 이상의 변환들을 적용할 수 있다. 비디오 인코더(20)는 TU에 대한 Cr 계수 블록을 생성하기 위해 TU의 Cr 변환 블록에 하나 혹은 그 이상의 변환들을 적용할 수 있다.

[0062] 계수 블록(예를 들어, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록)을 생성한 후, 비디오 인코더(20)는 계수 블록을 양자화할 수 있다. 양자화는 일반적으로 변환 계수를 양자화하여 변환 계수를 나타내기 위하여 사용되는 데이터의 양을 가능한 감소시켜 추가적인 압축을 제공하는 프로세스를 나타낸다. 비디오 인코더(20)가 계수 블록을 양자화한 후, 비디오 인코더(20)는 엔트로피 인코딩 기술을 적용하여 양자화된 변환 계수들을 나타내는 선택스 요소들을 인코딩할 수 있다. 예를 들어, 비디오 인코더(20)는 양자화된 변환 계수들을 나타내는 선택스 요소들에 대해 컨텍스트-적응 이진 산술 코딩(Context-Adaptive Binary Arithmetic Coding: CABAC)을 수행할 수 있다. 마지막으로, 비디오 인코더(20)는 저장 장치(32)에 저장되거나 목적지 장치(14)로 전송되는, 코딩된 프레임들 및 관련 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수 있다.

[0063] 비디오 인코더(20)에 의해 생성된 비트스트림을 수신한 후, 비디오 디코더(30)는 비트스트림을 파싱(parsing)하여 비트스트림으로부터 선택스 요소들을 획득할 수 있다. 비디오 디코더(30)는 비트스트림으로부터 획득된 선택스 요소들에 적어도 부분적으로 기반하여 비디오 데이터의 프레임들을 재구성할 수 있다. 비디오 데이터를 재구성하는 프로세스는 일반적으로 비디오 인코더(20)에 의해 수행되는 인코딩 프로세스와 반대된다. 예를 들어, 비디오 디코더(30)는 현재 CU의 TU들과 연관된 계수 블록들에 대해 역변환을 수행하여 현재 CU의 TU들과 연관된 잔여 블록을 재구성할 수 있다. 비디오 디코더(30)는 또한 현재 CU의 PU들에 대한 예측 블록들의 샘플들을 현재 CU의 TU들의 변환 블록들의 대응하는 샘플들에 가산함으로써 현재 CU의 코딩 블록들을 재구성한다. 프레임의 각각의 CU에 대한 코딩 블록들을 재구성한 후, 비디오 디코더(30)는 프레임을 재구성할 수 있다.

[0064] 전술한 바와 같이, 비디오 코딩은 주로 2개의 모드, 즉, 인트라-프레임 예측(또는 인트라-예측) 및 인터-프레임 예측(또는 인터-예측)을 사용하여 비디오 압축을 달성한다. 인트라 블록 카피(intra block copy: IBC)는 인트라-프레임 예측 또는 제3 모드로 간주될 수 있다는 점에 유의해야 한다. 두 모드 사이에서, 인터-프레임 예측은 참조 비디오 블록으로부터 현재 비디오 블록을 예측하기 위한 움직임 벡터들의 사용함으로 하여 인트라-프레임 예측보다 코딩 효율에 더 기여한다.

[0065] 그러나, 비디오 데이터 캡처링(capturing) 기술의 지속적인 향상과 비디오 데이터의 세부 사항을 보존하기 위한 더욱 미세한 비디오 블록 크기로 인해, 현재 프레임에 대한 움직임 벡터들을 나타내기 위하여 필요한 데이터의

양도 크게 증가하였다. 이 문제를 극복하는 한 가지 방법은 공간적 및 시간적 도메인들 모두에서 인접한 CU 그룹이 예측 목적으로 유사한 비디오 데이터를 가질 뿐만 아니라 이러한 인접한 CU들 사이의 움직임 벡터들도 유사하다는 사실로부터 이익을 얻는 것이다. 따라서, 공간적으로 인접한 CU들 및/또는 시간적으로 동일 위치(co-located) CU들의 움직임 정보는 현재 CU의 "움직임 벡터 예측자(Motion Vector Predictor: MVP)라고도 하는 공간적 및 시간적 상관 관계를 탐색하여 현재 CU의 움직임 정보(예를 들어, 움직임 벡터)의 근사치로 사용할 수 있다.

[0066] 현재 CU의 실제 움직임 벡터를 비디오 비트스트림에 인코딩하는 대신(예를 들어, 도 2와 관련하여 기술한 바와 같이 움직임 추정부(42)에 의해 결정되는 실제 움직임 벡터), 현재 CU의 실제 움직임 벡터로부터 현재 CU의 움직임 벡터 예측자(motion vector predictor)를 감산하여 현재 CU에 대한 움직임 벡터 차이(Motion Vector Difference: MVD)를 생성한다. 이렇게 함으로써, 프레임의 각 CU에 대해 움직임 추정부(42)에 의해 결정된 움직임 벡터를 비디오 비트스트림으로 인코딩할 필요가 없고, 비디오 비트스트림에서 움직임 정보를 나타내기 위하여 사용되는 데이터의 양이 상당히 감소될 수 있다.

[0067] 코드 블록의 인터-프레임 예측 동안 기준 프레임에서 예측 블록을 선택하는 프로세스와 유사하게, 비디오 인코더(20)와 비디오 디코더(30) 모두는 현재 CU의 공간적으로 인접하는 CU들 및/또는 시간적으로 동일 위치 CU들과 연관된 잠재적 후보 움직임 벡터를 사용하는 현재 CU에 대한 움직임 벡터 후보 리스트("머지 리스트"라고도 함)를 구성하고, 다음으로 현재 CU에 대한 움직임 벡터 예측자로서 움직임 벡터 후보 리스트로부터 하나의 멤버를 선택하기 위해, 비디오 인코더(20)와 비디오 디코더(30) 모두에 의해 일련의 룰이 채택될 필요가 있다. 이렇게 함으로써, 비디오 인코더(20)로부터 움직임 벡터 후보 리스트 자체를 비디오 디코더(30)로 전송할 필요가 없으며, 움직임 벡터 후보 리스트 내에서 선택된 움직임 벡터 예측자의 인덱스는 비디오 인코더(20)와 비디오 디코더(30)가 현재 CU를 인코딩 및 디코딩하기 위해 움직임 벡터 후보 리스트 내에서 동일한 움직임 벡터 예측자를 사용하기에 충분하다. 따라서, 선택된 움직임 벡터 예측자의 인덱스만 비디오 인코더(20)로부터 비디오 디코더(30)로 전송될 필요가 있다.

[0068] 블록 기반 비디오 코딩 프로세스(예를 들어, 강화된 압축 모델(Enhanced Compression Model; ECM))에서의 변환 계수 코딩에 관한 간략한 논의가 본 명세서에서 제공된다. 구체적으로, 각 변환 블록은 먼저 복수의 계수 그룹(CG)들로 분할되며, 각 계수 그룹은 루마 컴포넌트에 대한 4×4 서브블록 및 크로마 컴포넌트에 대한 2×2 서브블록의 변환 계수를 포함한다. 변환 블록 내의 변환 계수들의 코딩은 계수 그룹들의 단위로 수행된다. 예를 들어, 변환 블록 내의 계수 그룹들은 제1 미리 결정된 스캔 순서에 기초하여 스캔되고 코딩된다. 각각의 계수 그룹을 코딩할 때, 계수 그룹의 변환 계수들은 각각의 서브블록 내의 제2 미리 결정된 스캔 순서에 기초하여 스캔된다. ECM에서, 동일한 좌측 상단 스캔 순서를 적용하여 변환 블록 내의 계수 그룹들 및 각각의 계수 그룹 내의 상이한 변환 계수들을 스캔한다(예를 들어, 제1 및 제2 미리 결정된 스캔 순서들 모두가 좌측 상단 스캔 순서이다). 도 5는 일부 실시예들에 따른 계수 그룹에서 변환 계수들의 좌측 상단 스캔 순서를 나타내는 그래픽 표현이다. 도 5의 숫자 0 내지 15는 계수 그룹의 각 변환 계수의 대응하는 스캔 순서를 나타낸다.

[0069] ECM에서의 변환 계수 코딩 방식에 따라, 먼저, 변환 블록이 임의의 0이 아닌 변환 계수들을 포함하는지를 가리키기 위해 각각의 변환 블록에 대해 플래그가 시그널링된다. 변환 블록에 적어도 하나의 0이 아닌 변환 계수가 존재하는 경우, 좌측 상단 스캔 순서에 따라 스캔된 마지막 하나의 0이 아닌 변환 계수의 위치는 비디오 인코더(20)로부터 비디오 디코더(30)로 명시적으로 시그널링된다. 마지막 하나의 0이 아닌 변환 계수의 위치가 시그널링됨에 따라, 마지막 계수 그룹(즉, 마지막 하나의 0이 아닌 계수를 포함하는 계수 그룹) 이전에 코딩된 모든 계수 그룹에 대해 플래그가 추가로 시그널링된다. 이에 대응하여, 플래그의 수는 해당 계수 그룹에 0이 아닌 변환 계수가 포함되는지 여부를 가리킨다. 계수 그룹의 플래그가 0과 같으면(계수 그룹의 모든 변환 계수가 0임을 가리킴), 계수 그룹에 대한 추가 정보를 보낼 필요가 없다. 그렇지 않으면(예를 들어, 계수 그룹의 플래그는 1과 같음), 계수 그룹 내의 각 변환 계수의 절대값 및 부호는 스캔 순서에 따라 비트스트림에서 시그널링된다. 그러나, 기존 설계에서는 변환 계수의 부호가 바이패스 코딩되어(예를 들어, 컨텍스트 모델이 적용되지 않음), 현재 설계에서는 비효율적인 변환 코딩을 초래한다. 본 개시내용과 일치하게, 변환 계수의 부호 예측을 갖는 개선된 LFNST 프로세스를 이하에서 보다 상세히 설명하여, 변환 코딩 효율이 향상될 수 있도록 한다.

[0070] 도 6은 일부 실시예들에 따른 LFNST 프로세스를 나타내는 그래픽 표현이다. VVC에서, 주 변환 후에 2차 변환 툴(예를 들어, LFNST)을 적용하여 인트라 코딩된 블록들의 변환 계수들의 에너지를 압축한다. 도 6에 나타난 바와 같이, 비디오 인코더(20)에서 순방향 주 변환(603)과 양자화(605) 사이에는 순방향 LFNST(604)가 적용되고, 비디오 디코더(30)에서 역양자화(607)와 역방향 주 변환(609) 사이에는 역방향 LFNST(608)가 적용된다. 예를 들어, LFNST 프로세스는 순방향 LFNST(604) 및 역방향 LFNST(608)를 모두 포함할 수 있다. 일부 예로서, 4×4

순방향 LFNST(604)의 경우, 16개의 입력 계수가 있을 수 있고; 8×8 순방향 LFNST(604)의 경우, 64개의 입력 계수가 있을 수 있고; 4×4 역방향 LFNST(608)의 경우, 8개의 입력 계수가 있을 수 있고; 8×8 역방향 LFNST(608)의 경우, 16개의 입력 계수가 있을 수 있다.

[0071] 순방향 LFNST(604)에서는, 다양한 변환 크기를 갖는 분리 불가능 변환들이 코딩 블록의 크기에 기초하여 적용되고, 이는 행렬 곱셈 프로세스를 이용하여 나타낼 수 있다. 예를 들어, 순방향 LFNST(604)가 4×4 블록에 적용된다고 가정한다. 4×4 블록의 샘플은 다음 식 (1)과 같은 행렬 X 를 이용하여 표현할 수 있다:

$$X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix} \quad (1).$$

[0073] 행렬 X 는 다음 식 (2)와 같이 벡터 $\vec{X}$ 로 직렬화될 수 있다:

$$\vec{X} = [X_{00} \ X_{01} \ X_{02} \ X_{03} \ X_{10} \ X_{11} \ X_{12} \ X_{13} \ \dots \ X_{30} \ X_{31} \ X_{32} \ X_{33}]^T \quad (2).$$

[0075] 상기 식 (1) 또는 (2)에서, X 는 순방향 주 변환(603)을 통해 얻어진 계수 행렬을 나타내고, $X_{i,j}$ 는 행렬 X 에서의 주 변환 계수를 나타낸다. 그러면, 식 (3)에 따라 순방향 LFNST(604)가 다음과 같이 적용된다:

$$\vec{F} = T \cdot \vec{X} \quad (3).$$

[0077] 상기 식 (3)에서 $\vec{F}$ 는 순방향 LFNST(604) 이후의 변환 계수들을 나타내고, T 는 변환 커널(예를 들어, 변환 행렬)을 나타낸다. 상기 예에서 X 는 4×4 행렬이므로, T 는 16×16 행렬이다. 이어서 16×1 벡터 $\vec{F}$ 는 소정의 스캔 순서에 따라 4×4 블록(예를 들어, 4×4 크기의 계수 행렬)으로 재정렬된다. 벡터 $\vec{F}$ 에서 제2 변환 계수 이전에 위치하는 제1 변환 계수는 4×4 블록에서의 제2 변환 계수보다 더 작은 스캐닝 인덱스와 연관될 수 있다.

[0078] 일부 구현예들에서, 축소된 분리 불가능 변환 커널이 LFNST 프로세스에 적용될 수 있다. 예를 들어, 상기 식 (3)에 기초하여, 순방향 LFNST (604)는 변환 계수를 저장하기 위해 계산 동작 및 메모리 소스 측면에서 비용이 많이 드는 직접 행렬 곱셈(direct matrix multiplication)에 기초한다. 따라서 LFNST 설계에서 축소된 분리 불가능 변환 커널을 사용하여 N 차원 벡터를 다른 공간의 R 차원 벡터에 매핑함으로써, LFNST 프로세스의 구현 비용을 줄일 수 있다(여기서, $R < N$). 예를 들어, 변환 커널에 $N \times N$ 행렬을 사용하는 대신, 식 (4)에 표시된 대로 순방향 LFNST(604)에서 $R \times N$ 행렬이 변환 커널로 사용된다:

$$T_{R \times N} = \begin{bmatrix} t_{0,0} & t_{0,1} & \dots & t_{0,N-1} \\ t_{1,0} & t_{1,1} & \dots & t_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ t_{R-1,0} & t_{R-1,1} & \dots & t_{R-1,N-1} \end{bmatrix} \quad (4).$$

[0080] 상기 식 (4)에서, $T_{R \times N}$ 에서의 R 개의 기저(base) 벡터는 원래의 N 차원 변환 커널(즉, $N \times N$)의 첫 R 개의 기저 벡터를 선택하여 생성된다. 또한, 역방향 LFNST(608)에 대한 역변환 행렬은, $T_{R \times N}^T$ 이 직교하고 있는 경우에, 순방향 변환 행렬 $T_{R \times N}$ 의 전치이다.

[0081] 8×8 LFNST의 경우, 인자 $N/R = 4$ 가 적용되면, 64×64 변환 행렬은 순방향 LFNST(604)에 대해 16×48 변환 행렬로 감소되고, 64×64 역변환 행렬은 역방향 LFNST(608)에 대해 48×16 역변환 행렬로 축소된다. 이는 주 변환 계수들의 좌측 상단 영역의 8×8 서브블록에 LFNST 프로세스를 적용함으로써 달성된다. 구체적으로, 16×48 순방향 LFNST가 적용되면, 좌측 상단 8×8 서브블록(그 우측 하단 4×4 서브블록 제외) 내의 3개의 4×4 서브블록으로부터 48개의 변환 계수들을 입력으로 사용한다. 일부 예들에서, LFNST 프로세스는 좌측 상단 4×4 서브블록 이외의 모든 변환 계수가 0인 경우에만 적용 가능하도록 제한되는데, 이는 LFNST가 적용될 때 모든 오직 주 변환 계수들이 0이어야 한다는 것을 가리킨다. 또한, 최악의 경우의 복잡성을 제어하기 위해(픽셀당 곱셈들의 측면에서), 4×4 및 8×8 코딩 블록들에 대한 LFNST 행렬은 각각 8×16 및 8×48 변환으로 강제된다. 4×M 및 M×4 코딩 블록($M \geq 4$)의 경우, LFNST의 분리 불가능 변환 행렬은 16×16이다.

[0082] LFNST 변환 시그널링에서는 총 4개의 변환 세트가 존재하며, LFNST 설계에서는 변환 세트당 2개의 분리 불가능

변환 커널이 인에이블된다. 인트라 블록의 인트라 예측 모드에 따라 4개의 변환 세트 중에서 변환 세트를 선택한다. 인트라 예측 모드에서 변환 세트로의 맵핑은 다음 표 1과 같이 미리 결정된다. 3개의 크로스-컴포넌트 선형 모델(Cross-Component Linear Model: CCLM)모드들(예를 들어, INTRA_LT_CCLM, INTRA_T_CCLM, 또는 INTRA_L_CCLM) 중 하나가 현재 블록($81 \leq \text{predModeIntra} \leq 83$)에 사용되면, 현재 크로마 블록에 대해 변환 세트 "0"이 선택된다. 각 변환 세트에 대해, 비트스트림에서 LFNST 인덱스를 시그널링함으로써 선택된 분리 불가능 2차 변환 후보를 가리킨다.

표 1

IntraPredMode	세트
$\text{IntraPredMode} < 0$	1
$0 \leq \text{IntraPredMode} \leq 1$	0
$2 \leq \text{IntraPredMode} \leq 12$	1
$13 \leq \text{IntraPredMode} \leq 23$	2
$24 \leq \text{IntraPredMode} \leq 44$	3
$45 \leq \text{IntraPredMode} \leq 55$	2
$56 \leq \text{IntraPredMode} \leq 80$	1
$81 \leq \text{IntraPredMode} \leq 83$	0

일부 예들에서, LFNST는 최초 16×16 서브블록이외의 모든 변환 계수들이 0인 경우 인트라 블록에 적용되도록 제한되기 때문에, LFNST 인덱스 시그널링은 마지막 유효한(즉, 0이 아닌) 변환 계수의 위치에 의존한다. 예를 들어, 4×4 및 8×8 코딩 블록의 경우, LFNST 인덱스는 마지막 유효 변환 계수의 위치가 8보다 작은 경우에만 시그널링된다. 다른 코딩 블록 크기의 경우, LFNST 인덱스는 마지막 유효 변환 계수의 위치가 16보다 작은 경우에만 시그널링된다. 그렇지 않으면(즉, LFNST 인덱스가 시그널링되지 않음), LFNST 인덱스는 0인 것으로 추정되며, 즉, LFNST는 디세이블된다. 또한, 변환 계수를 캐싱하기 위한 버퍼의 크기를 줄이기 위해, LFNST는 시퀀스 파라미터 세트(SPS)에서 시그널링된 바와 같이 현재 코딩 블록의 폭 또는 높이 중 어느 하나가 최대 변환 크기(즉, 64)보다 클 경우 허용되지 않는다. 한편, LFNST는 주 변환이 DCT2인 경우에만 적용된다. 또한, LFNST는 인트라 및 인터 슬라이스의 인트라 코딩 블록과 루마 및 크로마 컴포넌트 모두에 적용된다. 듀얼 트리 또는 로컬 트리가 인에이블되는 경우(루마 및 크로마 컴포넌트의 파티션이 잘못 정렬된 경우), LFNST 인덱스는 루마 및 크로마 컴포넌트에 대해 별도로 시그널링된다(즉, 루마 및 크로마 컴포넌트는 상이한 LFNST 변환을 적용할 수 있음). 그렇지 않으면, 단일 트리가 적용될 때(루마 및 크로마 컴포넌트의 파티션이 정렬된 경우), LFNST는 단일 LFNST 인덱스가 시그널링되는 루마 컴포넌트에만 적용된다.

ECM에서의 LFNST 설계는 큰 블록 크기의 잔여 샘플들의 더 나은 에너지 압축을 제공하기 위해 추가적인 LFNST 커널들이 도입되는 것을 제외하고는 VVC에서의 것과 동일하다. 구체적으로, 변환 블록의 폭 또는 높이가 16보다 작지 않은 경우, 주 변환으로부터 생성된 저주파 변환 계수의 좌측 상단 영역에 새로운 LFNST 변환이 도입된다. 현재 ECM에서, 도 7a에 도시된 바와 같이, 저주파 영역은 주 변환 계수의 좌측 상단 코너에 6개의 4×4 서브블록(예를 들어, 도 7에서 회색으로 표시된 6개의 4×4 서브블록)을 포함한다. 이 경우, 순방향 LFNST(604)에 대한 계수 입력의 수는 96개이다. 또한, 최악의 계산 복잡도를 제어하기 위해 순방향 LFNST(604)의 계수 출력의 수는 32개로 설정된다.

구체적으로, $W \geq 16$ 및 $H \geq 16$ 인 $W \times H$ 변환 블록의 경우, 32×96 순방향 LFNST가 적용되며, 이는 좌측 상단 영역에 있는 6개의 4×4 서브블록으로부터 96개의 변환 계수를 입력으로 받아 32개의 변환 계수를 출력한다. 다른 측면에서, ECM 내의 8×8 LFNST는 4개의 4×4 서브블록 모두에서 변환 계수들을 입력으로서 이용하고, 32개의 변환 계수들(즉, 순방향 LFNST(604)에 대한 32×64 행렬 및 역방향 LFNST(608)에 대한 64×32 행렬)을 출력한다. 이는 8×8 LFNST가 좌측 상단 영역에 있는 3개의 4×4 서브블록에만 적용되고 16개의 변환 계수(즉, 순방향 LFNST(604)에 대한 16×48 행렬 및 역방향 LFNST(608)에 대한 48×16 행렬)만을 생성하는 VVC와 상이하다. 또한, LFNST 세트들의 총 개수는 VVC의 4개로부터 ECM의 35개로 증가된다. VVC와 유사하게, LFNST 세트의 선택은 현재 코딩 유닛의 인트라 예측 모드에 의존하고, 각각의 LFNST 세트는 3개의 상이한 변환 커널들을 포함한다.

일부 예들에서, HEVC에서 사용되는 DCT2 변환 이외에 인터 및 인트라 코딩된 블록들 양자 모두의 잔여들을 변환하기 위해 다중 변환 선택(Multiple Transform Selection: MTS) 방식이 적용된다. MTS 방식은 DCT8 및 DST7 변환에서 선택된 복수의 변환들을 사용한다.

[0088] 예를 들어, 시퀀스 레벨에서 2개의 제어 플래그가 지정되어 인트라 모드와 인터 모드에 대한 MTS 방식이 각각 인에이블된다. 시퀀스 레벨에서 MTS 방식이 인에이블되는 경우, 다른 CU 레벨 플래그가 추가로 시그널링되어 MTS 방식의 적용 여부를 가리킨다. 일부 구현예들에서, MTS 방식은 루마 컴포넌트에만 적용된다. 게다가, MTS 방식은 다음의 조건들이 만족될 때에만 시그널링된다: (a) 폭과 높이가 모두 32보다 작거나 같다; (b) 코딩된 블록 플래그(coded block flag: CBF)가 1과 같다. MTS CU 플래그(MTS_CU_flag)가 0과 같으면, DCT2는 수평 방향과 수직 방향 모두에 적용된다. MTS CU 플래그가 1인 경우에는 다른 2개의 플래그를 추가적으로 시그널링하여 수평 방향과 수직 방향에 대한 변환 유형을 각각 가리킨다. MTS 수평 및 수직 제어 플래그와 적용되는 변환 간의 매핑은 다음 표 2와 같다.

표 2

[0089]

MTS_CU_flag	MTS_Hor_flag	MTS_Ver_flag	인트라/인터	
			수평	수직
0			DCT2	
1	0	0	DST7	DST7
1	0	1	DCT8	DST7
1	1	0	DST7	DCT8
1	1	1	DCT8	DCT8

[0090] 변환 매트릭스의 정밀도와 관련하여 모든 MTS 변환 계수는 6비트 정밀도로 DCT2 코어 변환과 동일하다. VVC가 HEVC에 사용되는 모든 변환 크기를 지원한다는 점을 감안할 때 4 포인트, 8 포인트, 16 포인트, 32 포인트 DCT-2 변환과 4 포인트 DST-7 변환을 포함하여 HEVC에 사용되는 모든 변환 코어는 VVC와 동일하게 유지된다. 한편, VVC 변환 설계에서는 64 포인트 DCT-2, 4 포인트 DCT-8, 8 포인트, 16 포인트, 32 포인트 DST-7 및 DCT-8을 포함한 다른 변환 코어를 추가로 지원한다. 또한, 큰 사이즈 DST-7 및 DCT-8의 복잡도를 줄이기 위해, DST-7 및 DCT-8 변환 블록의 폭 또는 높이 중 어느 하나가 32인 경우, 16×16 저주파 영역 밖에 위치하는 고주파 변환 계수를 0(Zero-out(제로 아웃)이라고도 함)으로 설정한다.

[0091] VVC에서, DCT2 이외에, DST7 및 DCT8 변환 커널들만이 인트라 및 인터 코딩을 위해 활용된다. 인트라 코딩의 경우, 일반적으로 잔여 시그널의 통계적 특성들은 인트라 예측 모드에 의존한다. 추가적인 주 변환들은 잔여 특성들의 다양성을 처리하는 데 도움이 될 수 있다.

[0092] DCT5, DST4, DST1 및 아이덴티티 변환(identity transform: IDT)을 포함하는 추가적인 주 변환들이 ECM에 사용된다. 또한, MTS 세트는 TU 크기 및 인트라 모드 정보에 의존하여 만들어진다. 16개의 서로 다른 TU 크기를 고려할 수 있으며, 인트라 모드 정보에 따라 각 TU 크기에 대해 5개의 서로 다른 클래스를 고려할 수 있다. 각각의 클래스에 대해, 4개의 서로 다른 변환 쌍(VVC의 변환과 동일함)이 고려된다. 총 80개의 서로 다른 클래스들을 고려할 수 있지만, 그러한 서로 다른 클래스들 중 일부는 동일한 변환 세트를 공유하는 경우가 많다. 따라서, 얻어진 룩업 테이블(LUT)에는 58개(80개 미만)의 고유 엔트리들이 존재한다.

[0093] 각도 모드의 경우, TU 형상 및 인트라 예측에 대한 조인트 대칭이 고려된다. 따라서, TU 형상 A×B를 갖는 모드 i(i>34)는 TU 형상 B×A를 갖는 모드 j=(68 - i)에 대응되는 동일한 클래스에 매핑될 수 있다. 그러나, 각 변환 쌍에 대해, 수평 및 수직 변환 커널의 순서가 스왑된다. 예를 들어, 모드 18(수평 예측)을 갖는 16×4 블록과 모드 50(수직 예측)을 갖는 4×16 블록은 동일한 클래스에 매핑되고, 수직 및 수평 변환 커널이 스왑된다. 광각 모드들에 대해, 가장 가까운 종래의 각도 모드가 변환 세트 결정에 사용된다. 예를 들어, 모드 2는 -2와 -14 사이의 모든 모드에 사용된다. 마찬가지로, 모드 66은 모드 67 내지 모드 80에 사용된다.

[0094] 인트라 템플릿 매칭 예측은 현재 프레임의 재구성된 부분으로부터 예측 블록을 복사하는 인트라 예측 모드의 일 예로서, 재구성된 부분의 L 자형 템플릿은 현재 템플릿과 매칭된다. 미리 결정된 검색 범위에 대해, 비디오 인코더(20)는 현재 프레임의 재구성된 부분에서 현재 템플릿과 가장 유사한 템플릿(예를 들어, SAD 비용에 기초함)을 검색하고, 해당 블록을 예측 블록으로 사용한다. 그러면, 비디오 인코더(20)는 이 모드의 사용을 시그널링하고, 디코더 측에서 동일한 예측 동작을 수행한다. 도 8에 도시된 바와 같이, 현재 블록의 L자형 인과적 이웃을 미리 결정된 검색 영역에서의 다른 블록과 매칭시킴으로써, 예측 신호를 생성하고, 미리 결정된 검색 영역은 (a) 현재 CTU(R1); (b) 좌측 상단 CTU(R2); (c) 상측 CTU(R3); 및 (d) 좌측 CTU(R4)를 포함한다. 인트라 템플릿 매칭은 폭 및 높이에서 64 미만의 크기를 갖는 CU들에 대해 인에이블된다. 한편, 인트라 템플릿 매칭 예측 모드는 CU 레벨에서 플래그를 시그널링함으로써 가리킨다. 인트라 템플릿 매칭이 폭 또는 높이가 4 내지

16(4 및 16 포함)인 코딩 블록에 적용되는 경우, 해당 차원에 적용되는 주 변환은 DST-VII로 설정된다. 그렇지 않으면(즉, 폭 또는 높이가 4보다 작거나 16보다 큰 경우), DCT-II가 해당 차원에 적용된다.

[0095] 도 9는 일부 예들에 따른 부호 예측의 예시적인 프로세스를 나타내는 그래픽 표현이다. 일부 구현예들에서, 변환 블록의 이웃 블록의 샘플들로부터 변환 블록 내의 변환 계수들의 부호를 추정하도록 부호 예측을 수행할 수 있다. 추정된 부호들의 정확성은 부호 예측이 정확한지 여부를 가리키기 위해 컨텍스트 모델에 따라 코딩될 수 있다. 예를 들어, 하나의 컨텍스트 모델에서, 각 추정된 부호 간의 차이는 코딩될 수 있으며, 대응하는 참 부호는 "0"(또는 "1")을 가져 추정된 부호가 참 부호와 동일함(또는 동일하지 않음)을 가리킬 수 있다. 부호가 높은 비율로 정확하게 추정될 수 있는 경우(예를 들어, 부호의 90% 또는 95%가 정확하게 추정됨), 추정된 부호와 실제 부호 간의 차이는 0인 경향이 있으며, 이는 VVC에서 변환 계수에 대한 바이패스 코딩된 부호와 비교할 때 CABAC에 의해 효율적으로 엔트로피 코딩될 수 있다. 본 개시내용과 일치하게, 다른 컨텍스트 모델을 사용하여 부호 예측의 정확성을 엔트로피-코딩함으로써 코딩 효율성을 더욱 향상시킬 수 있다. 이들 다른 예시적인 컨텍스트 모델들은 이하에서 더 상세히 설명될 것이다.

[0096] 일반적으로, 현재 블록과 이웃 블록 사이의 경계에 있는 샘플 사이에는 높은 상관관계가 존재하며, 이는 부호 예측 방식에 의해 현재 블록의 변환 계수의 부호를 예측할 수 있다. 도 9에 도시된 바와 같이, 현재 블록에 M 개의 0이 아닌 변환 계수가 있다고 가정한다(M 개의 부호는 각각 + 또는 - 인). 그러면, 부호들의 가능한 총 조합의 수는 2^M 이 된다. 부호 예측 방식은 부호들의 각 조합을 이용하여 대응하는 가설(예를 들어, 현재 블록의 상단 및 좌측 경계에서 재구성된 샘플)을 생성하고, 대응하는 가설에서 재구성된 샘플들을 이웃 블록의 추정된 샘플들과 비교하여 재구성된 샘플들과 추정된 샘플들 간의 샘플 차이(예를 들어, SSD 또는 SAD)를 구한다. 샘플 차이를 최소화하는 부호들의 조합(2^M 개의 가능한 부호들의 조합 중)이 현재 블록에서 예측된 부호들로 선택된다.

[0097] 도 9에 나타난 바와 같은 일부 구현예들에서, M 개의 부호들의 각각의 조합에 대한 대응하는 가설을 생성하기 위해, M 개의 대응하는 변환 계수들은 역양자화 연산 및 역변환에 의해 처리되어 잔여 샘플들을 얻을 수 있다. 잔여 샘플을 예측 샘플과 함께 추가하여 재구성된 샘플을 얻을 수 있으며, 여기에는 현재 블록의 상단 및 좌측 경계에 있는 재구성된 샘플이 포함된다 (L자형 회색 영역(902)에 표시됨).

[0098] 일부 구현예들에서, 현재 블록과 이웃 블록의 경계에서 샘플 사이의 공간적 불연속성을 측정하는 비용 함수가 부호 조합의 선택에 사용된다. 비용 함수는 L2 노름(norm)(SSD)을 사용하는 대신 아래 식 (5)와 같이 L1 노름(SAD)에 기초할 수 있다:

$$\text{costHV} = \sum_{n=0}^{N-1} |2B_{-1,n} - B_{-2,n} - P_{0,n}| + \sum_{m=0}^{M-1} |2C_{m,-1} - C_{m,-2} - P_{m,0}| \quad (5).$$

[0100] 상기 식 (5)에서, $B_{i,n}$ ($i = -2, -1$ 인 경우)는 현재 블록의 상단 이웃 블록으로부터의 현재 블록의 이웃 샘플들을 나타낸다. $C_{m,i}$ ($j = -2, -1$ 인 경우)는 현재 블록의 좌측 이웃 블록으로부터의 현재 블록의 이웃 샘플들을 나타낸다. $P_{0,n}$ 및 $P_{m,0}$ 은 각각 현재 블록의 상단 및 좌측 경계 상의 대응하는 재구성된 샘플들을 나타낸다. N 과 M 은 각각 현재 블록의 폭과 높이를 나타낸다. 도 10은 부호 예측을 위한 비용 함수의 계산에 사용되는 현재 블록의 대응 샘플 $P_{0,n}$, $P_{m,0}$ 과 이웃 블록의 대응 샘플 $B_{i,n}$ 과 $C_{m,i}$ 를 나타낸다.

[0101] 일부 구현예들에서, 템플릿 기반 가설 재구성 방법은 복수의 역변환을 수행하는 복잡성을 피하기 위해 부호 예측 방식에 적용될 수 있다. 각각의 템플릿은 현재 블록의 상단 및 좌측 경계에 있는 재구성된 샘플들의 세트일 수 있고, 다른 모든 계수가 0인 상태에서 특정 계수가 1로 설정되는 계수 행렬에 역변환을 적용함으로써 획득될 수 있다. 역변환들(예를 들어, DCT, DST)이 선형인 것을 감안하면, 대응하는 가설은 미리 계산된 템플릿들의 세트의 선형 조합에 의해 생성될 수 있다.

[0102] 일부 구현예들에서, 예측된 부호들은 2개의 세트들로 분류되고 각각의 세트는 단일 CABAC 컨텍스트에 의해 코딩된다. 예를 들어, 제1 세트는 변환 블록의 좌측 상단 코너에서의 변환 계수들의 예측 부호들을 포함하고, 제2 세트는 변환 블록의 모든 다른 위치들에서의 변환 계수들의 예측 부호들을 포함한다.

[0103] HEVC와 마찬가지로, 스칼라 양자화가 VVC에서 사용된다. 일부 구현예들에서, VVC에서의 스칼라 양자화는 종속 스칼라 양자화로서 구현될 수 있다. 종속 스칼라 양자화는 변환 계수에 대한 허용 가능한 재구성 값들의 세트가 재구성 순서에서 현재 변환 계수 레벨에 선행하는 변환 계수 레벨들의 값들에 의존하는 방법을 지칭한다. 이 방법의 도미넌트 효과는 HEVC에서 사용되는 종래의 독립적인 스칼라 양자화와 비교하여, 허용 가능한 재구성 벡터

들이 N-차원 벡터 공간 내에 더 조밀하게 패킹된다는 것이다(N은 변환 블록 내의 변환 계수들의 수를 나타낸다). 즉, N-차원 단위 부피당 허용 가능한 재구성 벡터들의 주어진 평균 수에 대해, 입력 벡터와 그것의 가장 가까운 재구성 벡터 사이의 평균 왜곡이 감소된다는 것을 의미한다.

[0104] 종속 스칼라 양자화는 (a) 재구성 레벨이 다른 두 스칼라 양자화기를 정의하고, (b) 두 스칼라 양자화기 간에 전환하는 프로세스를 정의함으로써 구현될 수 있다. 도 11은 본 개시내용의 일부 구현예에 따른 VVC의 종속 스칼라 양자화에 사용되는 2개의 예시적인 스칼라 양자화기를 예시한다. 도 11에 나타난 바와 같이, Q0 및 Q1로 표시되는 2개의 스칼라 양자화기는 VVC의 양자화 설계에 적용된다. 이용 가능한 재구성 레벨들의 위치는 양자화 스텝 크기 Δ 에 의해 고유하게 특정된다. 이 구현예에서, 2개의 스칼라 양자화기(Q0 및 Q1) 사이의 선택은 비트 스트림에서 명시적으로 시그널링되지 않는다. 대신에, 현재 변환 계수에 사용되는 양자화기는 비디오 인코더(20)에 의한 코딩 순서 또는 비디오 디코더(30)에 의한 재구성 순서로 현재 변환 계수에 선행하는 변환 계수 레벨의 패리티에 의해 결정된다.

[0105] 일부 구현예들에서, 2개의 스칼라 양자화기들 사이의 스위칭은 상태 머신을 통해 이루어진다. 예를 들어, 도 12a는 일부 예들에 따른 종속 스칼라 양자화에 사용되는 4개의 상태들을 갖는 상태 머신을 사용하여 상태 전이를 나타내는 그래픽 표현이다. 도 12에 예시된 바와 같이, 상태들은 0, 1, 2, 3의 4개의 서로 다른 값들을 취할 수 있다. 이는 코딩/재구성 순서에서 현재 변환 계수 이전의 변환 계수 레벨들의 패리티들에 의해 고유하게 결정된다.

[0106] 일부 구현예들에서, 변환 블록에 대한 역양자화가 시작될 때, 상태가 0으로 설정된다. 변환 계수는 스캐닝 순서(즉, 그들이 엔트로피 디코딩되는 동일한 순서)로 재구성된다.

[0107] 현재 변환 계수가 재구성되면, 상태 머신에 따라 상태가 업데이트된다. 예를 들어, 도 12a에서, k는 변환 계수 레벨의 값을 나타낸다. 각 상태에서, 변환 계수 레벨 k의 패리티, 즉 $(k \& 1)$ 에 기초하여 다음 상태가 결정된다. $(k \& 1) == 1$ 인 다음 상태는 $(k \& 1) == 0$ 인 경우의 다음 상태와 다르다. 도 12a에 도시된 바와 같이, 상태 머신은 4개의 상태들 각각으로부터 2개의 서로 다른 상태들을 가리키는 2개의 화살표들을 포함한다. 도 12b는 일부 실시예들에 따른 도 12a의 상태 전이에 따른 예시적인 양자화기 선택을 나타내는 표이다. 예를 들어, 도 12a 및 도 12b에 따르면, 상태 1에서 다음 상태는 $(k \& 1) == 0$ 이면 2이고, $(k \& 1) == 1$ 이면 0일 것이다.

[0108] 이에 상응하여, 디코더에서, 식 (6)에 따라 하나의 변환 계수의 재구성된 양자화 인덱스가 계산될 수 있다:

[0109]
$$quantIdx = (abs(k) \ll 1) - (state \& 1) \quad (6).$$

[0110] 여기서 $abs()$ 는 입력의 절대값을 계산하는 함수이고 $state$ 는 현재 변환 계수의 레벨을 과싱할 때 상태 전환 머신의 현재 상태이다. 또한 디코더 측에서는 식 (7)에 따라 역양자화 후 재구성된 변환 계수를 구할 수 있다:

[0111]
$$transCoeff = quantIdx \cdot \Delta \quad (7).$$

[0112] 부호 예측 방식의 현재 설계에 존재하는 몇 가지 예시적인 결함이 본 명세서에서 확인된다. 제1 예에서, 현재 ECM에서의 부호 예측은 주 변환(예를 들어, DCT 및 DST 변환)만이 적용되는 변환 블록에서의 변환 계수에 대한 부호 예측에만 적용 가능하다. 진술한 바와 같이, LFNST는 주 변환으로부터의 변환 계수에 적용되어 인트라-코딩된 블록들의 잔여 샘플의 더 나은 에너지 압축을 제공할 수 있다. 그러나 부호 예측은 현재 ECM 설계에서 LFNST가 적용되는 변환 블록들에 대해 바이패스된다.

[0113] 제2예에서, 부호 예측의 복잡도를 제어하기 위해 변환 블록에 대해 미리 결정된 예측 부호의 최대 개수(L_{max} 로 표시됨)가 결정된다. 현재 ECM에서 비디오 인코더는 복잡도와 코딩 효율 사이의 트레이드오프에 기초하여 최대 개수의 값(예를 들어, $L_{max} = 8$)을 결정하고 그 값을 비디오 디코더로 전송한다. 더욱이, 각각의 변환 블록에 대해, 비디오 인코더 또는 디코더는 모든 변환 계수들을 래스터-스캔 순서로 스캔할 수 있고 첫 L_{max} 개의 0이 아닌 변환 계수들이 부호 예측을 위한 후보 변환 계수들로서 선택된다. 변환 블록에서 서로 다른 변환 계수를 동일하게 처리하는 것은 부호 예측의 정확도 측면에서 최적이지 아닐 수 있다. 예를 들어, 크기(magnitude)가 상대적으로 큰 변환 계수의 경우, 부호를 예측하면 올바른 예측을 얻을 가능성이 더 높을 수 있다. 이는 이러한 변환 계수에 잘못된 부호를 사용하면 상대적으로 작은 크기의 변환 계수를 사용하여 생성된 것보다 블록 경계의 재구성된 샘플에 더 큰 영향을 미치는 경향이 있기 때문이다.

[0114] 제3 예에서, 비디오 인코더 또는 디코더는 명시적인 부호 값들을 직접 코딩하는 대신에, 예측된 부호들의 정확성을 코딩할 수 있다. 예를 들어, 양수 부호를 갖는 변환 계수의 경우, 그것의 예측된 부호가 또한 양수이면,

비디오 인코더로부터 비디오 디코더로의 비트스트림에 빈 "0"만 가리키면 된다. 이 경우, 예측된 부호는 변환 계수의 실제 부호(또는 원래 부호)와 동일하고, 이 변환 계수에 대한 부호 예측이 정확하다는 것을 가리킨다. 그렇지 않으면(예를 들어, 예측된 부호가 음수인 반면 실제 부호는 양수인 경우), 빈 "1"이 비디오 인코더로부터 비디오 디코더로의 비트스트림에 포함될 수 있다. 모든 부호가 정확하게 예측되면, 비트스트림에서 가리킨 해당 빈은 0이고, 이는 CABAC에 의해 효율적으로 엔트로피 코딩될 수 있다. 부호들 중 일부가 잘못 예측되는 경우, 비트스트림에서 가리킨 해당 빈이 1이다. 산술 코딩 및 적절한 컨텍스트 모델을 사용하면 해당 확률에 따라 빈을 코딩하는 데 효율적일 수 있지만, 부호 값을 가리키기 위해 비트스트림에 생성되는 무시할 수 없는 비트들이 여전히 존재한다.

[0115] 제4 예에서, 현재 블록과 이웃 블록 사이의 경계들에서 샘플들 사이의 공간적 불연속성은 ECM에서 부호 예측의 현재 설계에서 최상의 부호 예측 조합을 선택하는 데 사용된다. 수직 방향과 수평 방향에 따른 경사 차이의 L1 노름(norm)은 공간적 불연속성을 캡처하는 데 사용된다. 그러나 이미지 신호의 분포는 일반적으로 균일하지 않기 때문에 수직 방향과 수평 방향만 사용하면 공간적 불연속성을 정확하게 캡처하지 못할 수 있다.

[0116] 본 개시내용과 일치하게, 블록 기반 비디오 코딩에서의 부호 예측을 위한 비디오 처리 방법 및 시스템은 상기 예시적인 결합들 중 하나 또는 그 이상을 해결하기 위해 본 명세서에 제공된다. 본 명세서에 개시된 방법 및 시스템은 하드웨어 코덱 구현들의 진화도를 고려하면서 부호 예측의 코딩 효율을 향상시킬 수 있다. 본 명세서에 개시된 방법 및 시스템은 블록들의 변환 계수들에 부호 예측 기술을 적용하는 변환 블록들의 코딩 효율을 향상시킬 수 있다.

[0117] 예를 들어, 전술한 바와 같이, 부호 예측은 변환 블록과 그 공간적 이웃 블록의 경계에 위치하거나 경계에 가까운 경계 샘플(boundary sample)들(변두리 샘플(border sample)들이라고도 함) 간의 상관관계에 기초하여 변환 블록에서 변환 계수들의 부호를 예측할 수 있다. 상관관계의 존재가 어느 특정 변환이 적용되는 것과 무관하다는 점을 감안할 때, 두 코딩 툴(즉, LFNST와 부호 예측)은 서로 간섭하지 않으며, 공동으로 적용될 수 있다. 또한, LFNST는 주 변환의 변환 계수 에너지를 더 압축하기 때문에, LFNST의 변환 계수에 대한 부호 예측은 주 변환의 부호 예측보다 더 정확할 수 있다. LFNST로부터의 변환 계수에 대한 잘못된 부호 예측은 경계 샘플의 평활도에 더 많은 불일치를 산출할 수 있기 때문이다. 따라서, 본 개시내용과 일치하게, 변환 계수 코딩의 코딩 효율을 향상시키기 위해 LFNST 및 부호 예측의 조합을 인에이블하는 조합 방식이 본 명세서에 개시된다. 또한, 본 명세서에서는 역변환의 수를 감소시키기 위해 예측된 부호들의 상이한 조합들에 대한 경계 샘플들을 재구성하기 위한 템플릿 기반 가설 생성 방식도 개시된다.

[0118] 다른 예에서, 상술한 바와 같이 부호 예측을 위한 후보 변환 계수들의 선택을 위한 변환 블록 내의 상이한 변환 계수들에 동등한 처리를 제공하는 대신, 그러한 변환 계수들의 부호들이 더 쉽게 예측된다는 점을 고려할 때, 이웃 블록들의 경계 샘플들 간의 불일치를 초래할 수 있는 변환 계수들에 더 높은 가중치들이 주어질 수 있다. 본 개시내용과 일치하게, 본 명세서에 개시된 방법 및 시스템은 부호 예측의 정확도를 향상시키기 위해 하나 또는 그 이상의 선택 기준에 기초하여 부호 예측을 위한 후보 변환 계수(예를 들어, 변환 블록에 대해 부호가 예측되어야 하는 변환 계수)를 선택할 수 있다. 예를 들어, 재구성된 경계 샘플에 더 많은 영향을 미치는 변환 계수(재구성된 경계 샘플에 영향을 덜 미치는 변환 계수가 아님)를 부호 예측의 후보 변환 계수로 선택하여 부호 예측의 정확도를 향상시킬 수 있다.

[0119] 또 다른 예에서, 변환 블록 내의 변환 계수들의 부호들이 높은 정확도 비율로 예측되는 경우(예를 들어, 예측된 부호들의 정확도가 80% 또는 90%와 같은 임계값보다 높은 경우), 변환 블록의 경계 샘플들과 그 이웃 블록들 사이에 강한 상관관계가 존재한다. 이 경우, 대부분의 시나리오에 대해 정확하게 예측될 수 있는 연속적인 변환 계수들(예를 들어, 변환 블록의 초기에 0이 아닌 변환 계수들 중 몇 개에 대해)이 있을 수 있다는 것이 일반적으로 발생한다. 이러한 시나리오에서는 복수의 빈 대신 단일 빈을 사용하여 모든 연속 변환 계수의 부호가 올바르게 예측되는지 여부를 가리켜 부호 예측의 시그널링 오버헤드를 절감할 수 있다. 본 개시내용과 일치하게, 부호 예측의 시그널링 오버헤드를 감소시키기 위해 벡터 기반 부호 예측 방식이 본 명세서에 개시된다. 각각의 0이 아닌 변환 계수의 부호를 개별적으로 예측하는 기존 부호 예측과 달리, 개시된 벡터 기반 부호 예측 방식은 연속적인 0이 아닌 후보 변환 계수의 세트를 그룹화하고, 예측된 부호의 정확성을 가리키는데 사용되는 평균 빈(또는 비트) 수가 효율적으로 감소될 수 있도록 그 대응하는 부호를 함께 예측한다.

[0120] 또 다른 예에서, 수직 및 수평 방향만을 사용하는 것은 현재 블록과 그 이웃 블록 사이의 경계에서 샘플 사이의 공간적 불연속성을 정확하게 캡처하지 못할 수 있다. 따라서, 공간적 불연속성을 더 정확하게 캡처하기 위해 더 많은 방향이 도입될 수 있다. 본 개시내용과 일치하게, 공간적 불연속성을 더 정확하게 캡처하기 위해 수직 및

수평 방향의 그래디언트 및 대각선 방향의 그래디언트를 모두 고려하는 개선된 비용 함수가 본 명세서에 개시된다.

[0121] 도 13은 본 개시내용의 일부 구현예들에 따른 블록 기반 비디오 코딩에서의 예시적인 부호 예측 프로세스(1100)를 나타내는 블록도이다. 일부 구현예들에서, 부호 예측 프로세스(1300)는 변환 프로세싱 유닛(52)에 의해 수행될 수 있다. 일부 구현예들에서, 부호 예측 프로세스(1300)는 비디오 인코더(20) 또는 비디오 디코더(30)의 하나 또는 그 이상의 프로세서(예를 들어, 하나 또는 그 이상의 비디오 프로세서)에 의해 수행될 수 있다. 본 개시내용 전체에 걸쳐, LFNST는 일반성의 손실 없이 2차 변환의 예로서 사용된다. 2차 변환의 다른 예들이 또한 본 명세서에 적용될 수 있는 것으로 고려된다.

[0122] ECM의 기존 설계에서는 LFNST가 적용되는 변환 블록에 대해 부호 예측이 디세이블되어 있다. 그러나 부호 예측의 원리는 변환 블록과 그 공간적 이웃 블록의 경계 샘플들 사이의 상관관계에 기초하여 변환 계수의 부호를 예측하는 것이며, 이는 변환 블록에 적용된 특정 변환 유형(예를 들어, 주 변환인지 2차 변환인지) 또는 변환 코어(예를 들어, DCT인지 DST인지)에 의존하지 않는다. 따라서, 부호 예측과 LFNST가 함께 적용되어 본 명세서에서 변환 코딩의 효율성을 더욱 향상시킬 수 있다. 본 개시내용과 일치하게, 부호 예측 프로세스(1300)는 주 변환과 2차 변환이 공동으로 적용되는 변환 블록에서 변환 계수의 부호를 예측하기 위해 적용될 수 있다.

[0123] 부호 예측 프로세스(1300)의 예시적인 개요가 본 명세서에 제공된다. 초기에, 부호 예측 프로세스(1300)는 비디오 오로부터의 비디오 프레임의 변환 블록에 주 변환 및 2차 변환을 적용하여 변환 블록에 대한 변환 계수를 생성함으로써 계수 생성 동작(1302)을 수행할 수 있다. 다음으로, 부호 예측 프로세스(1300)는 변환 계수들로부터 부호 예측을 위한 후보 변환 계수들의 세트를 선택함으로써 계수 선택 동작(1304)을 수행할 수 있다. 이어서, 부호 예측 프로세스(1300)는 템플릿 기반 가설 생성 방식을 적용하여 후보 변환 계수들의 세트에 대한 복수의 후보 가설들로부터 가설을 선택함으로써 가설 생성 동작(1306)을 수행할 수 있다. 추가적으로, 부호 예측 프로세스(1300)는 선택된 가설과 연관된 부호 후보들의 조합을 후보 변환 계수들의 세트에 대한 예측 부호들의 세트로 결정함으로써 부호 생성 동작(1108)을 수행할 수 있다. 동작(1302), 동작(1304), 동작(1306), 및 동작(1308)은 각각 이하에서 보다 상세히 설명된다.

[0124] 예를 들어, 비디오 인코더(20)의 변환 처리부(52)는 주 변환과 2차 변환을 공동으로 적용함으로써(예를 들어, 도 6에 도시된 바와 같이, 순방향 주 변환(603)과 순방향 LFNST(604)가 함께 적용됨) 잔여 비디오 데이터를 변환 블록의 변환 계수들로 변환할 수 있다. 하기 기재된 바와 같은 하나 또는 그 이상의 선택 기준들에 기초하여 변환 블록의 변환 계수들로부터 미리 결정된 수(예를 들어, L)의 0이 아닌 변환 계수들을 후보 변환 계수들로서 선택하되, $1 \leq L \leq$ 예측될 수 있는 부호들의 최대 수이다. 다음으로, 템플릿 기반 가설 생성 방식을 적용함으로써, L개의 후보 변환 계수들에 대해 각각 서로 다른 부호 후보들의 조합들을 사용하여 복수의 후보 가설들을 생성할 수 있고, 이로 인해 총 2^L 개의 후보 가설들을 생성할 수 있다. 각각의 후보 가설은 변환 블록의 상단 및 좌측 경계들에 재구성된 샘플들을 포함할 수 있다. 그런 다음, 수평, 수직 및 대각선 방향을 따라 결합된 그래디언트를 통합하는 비용 함수를 사용하여 각각의 후보 가설 재구성에 대한 비용을 계산할 수 있다. 복수의 후보 가설들로부터의 최소 비용과 연관된 후보 가설은 L개의 후보 변환 계수들의 부호를 예측하기 위한 가설로 결정될 수 있다. 예를 들어, 최소 비용과 연관된 후보 가설을 생성하는 데 사용되는 부호 후보들의 조합은 L개의 후보 변환 계수들에 대한 예측 부호로 사용된다.

[0125] 우선, 부호 예측 프로세스(1300)는 계수 생성 동작(1302)을 수행할 수 있으며, 여기서 주 변환(예를 들어, DCT, DST 등) 및 2차 변환(예를 들어, LFNST)이 변환 블록에 공동으로 적용되어 변환 블록에 대한 변환 계수를 생성할 수 있다. 예를 들어, 주 변환은 변환 블록에 적용되어 변환 블록에 대한 주 변환 계수를 생성할 수 있다. 그러면, LFNST를 변환 블록에 적용하여 주 변환 계수에 기초하여 LFNST 변환 계수를 생성할 수 있다.

[0126] 부호 예측 프로세스(1300)는 계수 선택 동작(1304)을 계속 수행할 수 있으며, 여기서 부호 예측을 위한 후보 변환 계수들의 세트는 하나 또는 그 이상의 선택 기준에 기초하여 변환 블록의 변환 계수들로부터 선택될 수 있다. 후보 변환 계수들의 선택을 통해, 정확하게 예측될 수 있는 후보 변환 계수들의 개수를 최대화할 수 있고, 부호 예측의 정확도를 향상시킬 수 있다.

[0127] 일부 구현예들에서, 후보 변환 계수들의 세트는 변환 계수들의 크기에 기초하여 변환 블록의 변환 계수들로부터 선택될 수 있다. 예를 들어, 후보 변환 계수들의 세트는 변환 블록 내의 나머지 변환 계수들보다 더 큰 크기를 갖는 하나 또는 그 이상의 변환 계수들을 포함할 수 있다.

[0128] 일반적으로, 더 큰 크기를 갖는 변환 계수들에 대해, 이러한 변환 계수의 예측된 부호가 정확할 가능성이 더 높

다. 이는 크기가 더 큰 이러한 변환 계수가 재구성된 샘플의 품질에 더 많은 영향을 미치는 경향이 있고, 이러한 변환 계수에 잘못된 부호를 사용하면 변환 블록과 공간 이웃 블록의 경계 샘플들 간에 불연속성이 생성될 가능성이 더 높기 때문이다. 이러한 이론적 근거에 기초하여, 부호 예측을 위한 후보 변환 계수들의 세트는 변환 블록 내의 0이 아닌 변환 계수들의 크기에 기초하여 변환 블록의 변환 계수들로부터 선택될 수 있다.

[0129] 부호 예측을 위해 변환 계수 크기 기반 재정렬 방식을 구현하는 여러 가지 방법이 있을 수 있다. 제1 구현예에서, 이 방식은 역양자화 후 재구성된 변환 계수(즉, 역양자화 변환 계수)를 정렬에 직접 사용하여 더 큰 크기를 갖는 역양자화된 변환 계수가 부호 예측을 위해 더 작은 크기를 갖는 변환 계수보다 앞에 배치되도록 할 수 있다. 예를 들어, 변환 블록 내의 모든 0이 아닌 변환 계수들을 스캔하고 정렬하여 크기의 내림차순에 따라 계수 리스트를 형성할 수 있다. 크기가 가장 큰 변환 계수는 계수 리스트로부터 선택되어, 후보 변환 계수 세트에서 제1 후보 변환 계수로 배치될 수 있고, 두 번째로 큰 크기를 갖는 변환 계수는 계수 리스트로부터 선택되어, 후보 변환 계수 세트에서 제2 후보 변환 계수로 배치될 수 있고, 선택한 후보 변환 계수의 수가 미리 결정된 수 L에 도달할 때까지 계속된다.

[0130] 제2 구현예에서, 역양자화된 변환 계수를 직접 사용하는 대신, 각 변환 계수의 양자화 인덱스(예를 들어, 식 (6)에 따라 얻은 바와 같은 *quantIdx*)를 사용하여 변환 계수의 크기를 나타낼 수 있고, 이러한 정렬을 위해 사용될 수 있다. 식 (7)에 나타낸 바와 같이, 하나의 역양자화된 변환 계수의 값은 양자화 인의 곱과 동일하고, 하나의 변환 블록 내의 모든 변환 계수의 역양자화에 대해 스텝 크기가 동일하기 때문에, 두 구현예는 실제로 수학적으로 동일하다. 그러나, 양자화 인덱스 *quantIdx*가 파싱 단계(역양자화된 변환 계수의 획득보다 더 빠름)에서 획득될 수 있음을 감안할 때, 제2 구현예는 일부 특정 하드웨어에 의해 구현될 때 특정 이점을 제공할 수 있다.

[0131] 일부 구현예들에서, 후보 변환 계수들의 세트는 비디오 코딩에 적용된 엔트로피 코딩의 계수 스캔 순서에 기초하여 변환 블록의 변환 계수들로부터 선택될 수 있다. 자연스러운 비디오 콘텐츠는 풍부한 저주파 정보를 가질 수 있기 때문에, 비디오 콘텐츠의 처리로부터 획득된 0이 아닌 변환 계수들의 크기는 저주파 위치에서 더 크고 고주파 위치로 갈수록 더 작은 경향이 있다. 따라서, 계수 스캔 순서(예를 들어, 지그 재그 스캔, 좌측 상단 스캔, 수평 스캔, 또는 수직 스캔 등)를 최신 비디오 코덱에서 사용하여 엔트로피 코딩을 위해 변환 블록의 변환 계수들을 스캔할 수 있다. 이 계수 스캔 순서를 사용함으로써, 더 큰 크기(일반적으로 더 낮은 주파수에 해당)를 갖는 변환 계수들이 더 작은 크기(통상적으로 더 높은 주파수에 해당)를 갖는 변환 계수들보다 먼저 스캔된다. 이 근거에 기초하여, 본 명세서에 개시된 부호 예측을 위한 후보 변환 계수들의 세트는 엔트로피 코딩을 위한 계수 스캔 순서에 기초하여 변환 블록의 변환 계수들로부터 선택될 수 있다. 예를 들어, 계수 스캔 순서를 사용하여 변환 블록 내의 모든 변환 계수들을 스캔함으로써 계수 리스트가 획득될 수 있다. 그러면, 계수 리스트 내의 첫 L개의 0이 아닌 변환 계수들이 부호 예측을 위한 후보 변환 계수들의 세트로서 자동으로 선택될 수 있다.

[0132] 일부 구현예들에서, 인트라 코딩된 블록에 대해, 부호 예측을 위한 후보 변환 계수들의 세트는 블록의 인트라 예측 방향에 기초하여 블록의 변환 계수들로부터 선택될 수 있다. 예를 들어, 인트라 예측 방향들(예를 들어, VVC 및 ECM에서의 67개의 인트라 예측 방향들)과 일관된 복수의 스캔 순서들은 비디오 인코더(20) 및 비디오 디코더(30) 둘 모두에서 룩-업 테이블로서 결정되고 저장될 수 있다. 인트라 블록의 변환 계수들을 코딩할 때, 비디오 인코더(20) 또는 비디오 디코더(30)는 인트라 블록의 인트라 예측에 가장 가까운 스캔 순서들로부터 스캔 순서를 식별할 수 있다. 비디오 인코더(20) 또는 비디오 디코더(30)는 식별된 스캔 순서를 사용하여 인트라 블록의 모든 0이 아닌 변환 계수들을 스캔하여 계수 리스트를 획득하고, 계수 리스트로부터 첫 L개의 0이 아닌 변환 계수들을 후보 변환 계수들의 세트로서 선택할 수 있다.

[0133] 일부 구현예들에서, 비디오 인코더(20)는 변환 블록의 변환 계수들에 대한 스캔 순서를 결정하고, 결정된 스캔 순서를 비디오 디코더(30)에 시그널링할 수 있다. 결정된 스캔 순서를 가리키는 하나 또는 그 이상의 새로운 신택스 엘리먼트는 비트스트림을 통해 시그널링될 수 있다. 예를 들어, 복수의 고정 스캔 순서(예를 들어, 상이한 변환 블록 크기 및 코딩 모드에 대해)는 비디오 인코더(20)에 의해 미리 결정될 수 있고, 비디오 디코더(30)와 미리 공유될 수 있다. 그러면, 고정된 스캔 순서들로부터 스캔 순서를 선택한 후, 비디오 인코더(20)는 선택된 스캔 순서를 비디오 디코더(30)에 가리키기 위해 단일 인덱스만을 시그널링하면 된다. 다른 예에서, 하나 또는 그 이상의 새로운 신택스 엘리먼트들은 변환 계수들의 임의의 선택된 스캔 순서의 시그널링을 인에이블하기 위해 사용될 수 있다. 일부 구현예들에서, 하나 또는 그 이상의 신택스 엘리먼트들은 다양한 코딩 레벨들, 예를 들어, 시퀀스 파라미터 세트(SPS), 픽처 파라미터 세트(PPS), 픽처(또는 슬라이스) 레벨, CTU(또는 CU) 레벨 등에서 시그널링될 수 있다.

- [0134] 일부 구현예들에서, 후보 변환 계수들의 세트는 변환 블록의 재구성된 경계 샘플들에 대한 변환 계수들의 영향 점수들에 기초하여 변환 블록의 변환 계수들로부터 선택될 수 있다. 구체적으로, 상기 식 (5)에 나타난 바와 같이, 부호들의 조합(즉, 예측된 부호들 또는 부호 예측인자)의 선택은 현재 변환 블록과 그 공간적 이웃 블록들 사이의 샘플들의 그래디언트의 불연속성을 최소화하기 위한 비용 함수에 기초한다. 따라서, 현재 변환 블록의 상단 및 좌측 경계에 있는 재구성된 샘플에 상대적으로 큰 영향을 미치는 변환 계수의 부호는 (5)에서 계산된 바와 같이 경계 샘플 간의 평활도에 큰 변화를 초래할 수 있기 때문에, 정확하게 예측될 가능성이 더 높다. 정확한 부호 예측의 백분율을 최대화하기 위해, 이러한 변환 계수(즉, 재구성된 경계 샘플에 더 많은 영향을 미치는 변환 계수)의 부호는 다른 변환 계수(즉, 재구성된 경계 샘플에 더 적은 영향을 미치는 변환 계수)보다 먼저 예측될 수 있다. 이러한 이론적 근거에 기초하여, 본 명세서에 개시된 부호 예측을 위한 후보 변환 계수들의 세트는 현재 변환 블록의 상단 및 좌측 경계들 상의 재구성된 샘플들에 대한 그들의 영향 점수들에 기초하여 선택될 수 있다.
- [0135] 예를 들어, 비디오 인코더(20) 또는 디코더(30)는 변환 블록의 재구성된 경계 샘플들에 대한 그들의 대응하는 영향 점수의 측정에 기초하여 모든 변환 계수들을 정렬할 수 있다. 변환 계수가 재구성된 경계 샘플에 더 큰 영향 점수를 갖는 경우, 정확하게 예측되는 것이 더 쉽기 때문에 부호 예측 후보 리스트에서 더 작은 인덱스로 할당될 수 있다. 본 명세서에 개시된 후보 변환 계수들의 세트는 부호 예측 후보 리스트에서 L개의 가장 작은 인덱스들을 갖는 L개의 변환 계수들일 수 있다.
- [0136] 일부 구현예들에서, 상이한 기준들을 적용하여 재구성된 경계 샘플들에 대한 변환 계수의 영향 점수를 정량화할 수 있다. 예를 들어, 변환 계수에 의해 발생하는 재구성된 경계 샘플의 변동 에너지를 측정하는 값을 영향 점수로 사용할 수 있으며, 이는 (L1 노름에서) 다음과 같이 얻을 수 있다:
- [0137]
$$\mathbf{v} = |c_{i,j}| \cdot (\sum_{n=0}^{N-1} |T_{i,j}(\mathbf{0}, n)| + \sum_{m=0}^{M-1} |T_{i,j}(m, \mathbf{0})|) \quad (8).$$
- [0138] 상기 식 (8)에서 $c_{i,j}$ 는 변환 블록 내 위치 (i, j) 에서의 변환 계수를 나타낸다. $T_{i,j}(l, k)$ 는 변환 계수 $c_{i,j}$ 와 연관된 템플릿의 위치 (l, k) 에서의 대응하는 경계 샘플을 나타낸다. N 및 M 은 각각 변환 블록의 폭 및 높이를 나타낸다. $\mathbf{v}$ 는 위치 (i, j) 에서의 변환 계수의 영향 점수를 나타낸다.
- [0139] 또 다른 예에서, 상기 식 (8)의 L1 노름은 L2 노름으로 대체될 수 있으므로, 영향 점수(예를 들어, 변환 계수에 의해 야기된 재구성된 경계 샘플의 변동의 에너지에 대한 측정)는 다음과 같이 L2 노름을 사용하여 계산될 수 있다:
- [0140]
$$\mathbf{v} = |c_{i,j}| \cdot (\sum_{n=0}^{N-1} |T_{i,j}(\mathbf{0}, n)|^2 + \sum_{m=0}^{M-1} |T_{i,j}(m, \mathbf{0})|^2) \quad (9).$$
- [0141] 본 개시내용과 일치하게, 상기 식 (8) 및 (9)에서 계산을 위해 상단 및 좌측 경계 샘플(예를 들어, $T_{i,j}(\mathbf{0}, n)$ 및 $T_{i,j}(m, \mathbf{0})$)로 가리킨 바와 같이)만이 사용되지만, 본 명세서에 개시된 변환 계수 선택 방식은 대응하는 비용 함수에서 사용되는 현재 변환 블록의 재구성된 샘플을 변경함으로써 임의의 부호 예측 방식에도 적용될 수 있다.
- [0142] 변환 계수 크기 기저 방식과 유사하게, 영향 점수 기반 방식을 구현하는 다른 방법이 있을 수 있다. 제1 구현예에서, 방식은 식 (8) 및 (9)에 표시된 바와 같이, 역양자화된 변환 계수 $c_{i,j}$ 의 크기를 직접 사용할 수 있다. 제2 구현예에서, 양자화 인덱스 값 $quantIdx$ 를 대신 적용하여 역양자화된 변환 계수들 $c_{i,j}$ 를 대체함으로써 대응하는 비용 측정값을 계산할 수 있다. 구체적으로, 양자화 인덱스 값 $quantIdx$ 를 적용하면, 식 (8) 및 (9)는 아래의 식 (10) 및 (11)이 된다:
- [0143]
$$\mathbf{M} = |quantIdx_{i,j}| \cdot (\sum_{n=0}^{N-1} |T_{i,j}(\mathbf{0}, n)| + \sum_{m=0}^{M-1} |T_{i,j}(m, \mathbf{0})|) \quad (10).$$
- [0144]
$$\mathbf{M} = |quantIdx_{i,j}| \cdot (\sum_{n=0}^{N-1} |T_{i,j}(\mathbf{0}, n)|^2 + \sum_{m=0}^{M-1} |T_{i,j}(m, \mathbf{0})|^2) \quad (11).$$
- [0145] 여기서, $quantIdx_{i,j}$ 는 변환 블록의 위치 (i, j) 에서의 변환 계수의 양자화 인덱스를 나타낸다.
- [0146] 하나의 역양자화된 변환 계수의 값은 양자화 인덱스 $quantIdx$ 및 대응하는 스텝 크기 $\mathbf{v}$ 의 곱과 동일하고, 하나의 변환 블록 내의 모든 변환 계수의 역양자화에 대해 스텝 크기가 동일하기 때문에, 두 방법은 실제로 수학적으로 동일하다.

- [0147] 부호 예측 프로세스(1300)는 가설 생성 동작(1106)을 계속 수행할 수 있으며, 여기서 템플릿 기반 가설 생성 방식이 적용되어 복수의 후보 가설로부터 후보 변환 계수들의 세트에 대한 가설을 선택할 수 있다. 초기에, 후보 변환 계수들의 세트에 포함되는 계수들의 총 개수에 기초하여, 후보 변환 계수들의 세트에 대해 복수의 후보 후보들의 조합들이 결정될 수 있다. 예를 들어, 총 L개의 후보 변환 계수들이 존재하는 경우, 후보 변환 계수들의 세트에 대한 복수의 후보 후보들의 조합들은 2^L 개의 후보 후보들의 조합이 될 수 있다. 각 후보 후보는 음수 부호(-) 또는 양수 부호(+)일 수 있다. 후보 후보들의 각각의 조합은 총 L개의 음수 또는 양수 부호를 포함할 수 있다. 예를 들어, L=2인 경우, 복수의 후보 후보 조합은 각각 (+, +), (+, -), (-, -) 및 (-, -)인 $2^2=4$ 개의 후보 후보 조합을 포함할 수 있다.
- [0148] 다음으로, 템플릿 기반 가설 생성 방식을 적용하여 복수의 후보 후보 조합에 대한 복수의 후보 가설을 각각 생성할 수 있다. 수행될 필요가 있는 역방향 주 변환 및 역방향 2차 변환의 복잡성을 감소시키기 위해, 본 명세서에 개시된 템플릿 기반 가설 생성 방식은 변환 블록의 재구성된 경계 샘플의 생성을 최적화하는 데 사용될 수 있다. 템플릿 기반 가설 생성 방식을 구현하기 위한 두 가지 예시적인 방법이 본 명세서에 개시된다. 템플릿 기반 가설 생성 방식을 구현하기 위한 다른 예시적인 방법들이 가능할 것으로 고려되며, 이는 본 명세서에서 제한되지 않는다.
- [0149] 제1 예시적인 방법에서, 템플릿들의 선형 조합에 기초하여 각 후보 후보들의 조합에 대한 대응하는 후보 가설이 생성될 수 있고, 이에 따라 복수의 후보 후보들의 조합에 대해 복수의 후보 가설들이 각각 생성될 수 있다. 각각의 템플릿은 후보 변환 계수들의 세트로부터의 후보 변환 계수에 대응할 수 있다. 각각의 템플릿은 변환 블록의 상단 및 좌측 경계 상의 재구성된 샘플들의 그룹을 나타낼 수 있다. 각 템플릿은 변환 블록에 역방향 2차 변환 및 역방향 주 변환을 적용함으로써 생성될 수 있으며, 여기서 1로 설정된 템플릿에 대응하는 후보 변환 계수를 제외하고, 후보 변환 계수의 세트의 각각의 후보 변환 계수는 0으로 설정된다(예를 들어, 템플릿에 대응하는 후보 변환 계수는 1로 설정되는 반면, 나머지 후보 변환 계수들 각각은 0으로 설정된다).
- [0150] 예를 들어, 각 후보 후보들의 조합에 대한 대응하는 후보 가설은 템플릿들의 선형 조합으로 설정될 수 있다. 해당 후보 변환 계수에 대응하는 템플릿의 경우, 템플릿에 대한 해당 가중치는 해당 후보 변환 계수에 대응하는 역양자화 변환 계수의 크기로 설정될 수 있다. 템플릿들의 선형 조합에 기초한 가설 생성의 예는 도 14에 예시되어 있으며, 이는 아래에서 보다 상세히 설명된다.
- [0151] 후보 변환 계수들의 부호를 예측하기 위해, 비디오 인코더(20) 또는 디코더(30)는 비용 함수로부터 계산된 비용 값을 최소화할 수 있는 후보 후보들의 조합과 연관된 가설을 식별하기 전에 모든 후보 가설들을 살펴볼 수 있다. 상술한 제1 예시적인 방법에서, 각각의 후보 가설은 복수의 템플릿들의 조합에 기초하여 생성될 수 있으며, 이는 이러한 조합에 관련된 샘플당 계산들(예를 들어, 덧셈, 곱셈 및 시프트)을 고려할 때 상대적으로 복잡하다. 비용 함수로부터 계산된 비용 값을 최소화하는 가설의 식별과 관련된 계산 복잡도를 감소시키기 위해, 제2 예시적인 방법이 본 명세서에 도입된다.
- [0152] 제2 예시적인 접근법에서, 복수의 후보 가설과 연관된 복수의 후보 후보의 조합은 복수의 후보 가설에 대한 복수의 가설 인덱스로서 각각 취급될 수 있다. 예를 들어, 숫자 0과 1은 각각 양수 부호(+)와 음수 부호(-)를 나타내도록 구성할 수 있다. 후보 가설에 대응하는 후보 후보의 조합은 후보 가설에 대한 고유한 표현(즉, 가설 인덱스)으로 사용될 수 있다. 예를 들어, 예측되는 후보가 세 개(예를 들어, L=3)라고 가정한다. 가설 인덱스(000)는 3개의 후보 후보를 모두 양수로 설정함으로써 생성되는 후보 가설을 나타낼 수 있다(예를 들어, 3개의 후보 후보가 (+, +, +)임). 마찬가지로, 가설 인덱스(010)는 제2 후보 후보를 음수로 설정하면서 제1 후보 후보 및 제3 후보 후보를 양수로 설정(예를 들어, 3개의 후보 후보가 (+, -, +)임)하여 생성된 후보 가설을 나타낼 수 있다.
- [0153] 그러면, 복수의 후보 가설들은 복수의 가설 인덱스들의 게리 코드(Gary code) 순서에 기초하여 생성될 수 있으며, 이에 따라 이전 가설 인덱스를 갖는 이전 후보 가설의 재구성된 샘플들은 현재 가설 인덱스를 갖는 현재 후보 가설을 생성하는데 사용될 수 있다. 현재 후보 가설의 현재 가설 인덱스는 복수의 가설 인덱스의 게리 코드 순서에서 이전 후보 가설의 이전 가설 인덱스 바로 뒤에 있을 수 있다. 현재 가설 인덱스는 이전 가설 인덱스의 부호 후보를 양(또는 음)에서 음(또는 양)으로 변경하여 생성할 수 있다. 예를 들어, 현재 가설 인덱스는 이전 가설 인덱스의 단일 "0"(또는 "1")을 "1"(또는 "0")로 변경하여 얻을 수 있다.
- [0154] 예를 들어, 복수의 가설 인덱스들은 복수의 가설 인덱스들의 게리 코드 순서에 기초하여 재정렬되어, 재정렬된 가설 인덱스들의 시퀀스를 생성할 수 있다. 가설 인덱스들의 재정렬된 시퀀스에서 제1 가설 인덱스에 대해, 각

후보 변환 계수들의 세트가 1로 설정되는 변환 블록에 역방향 2차 변환 및 역방향 주 변환을 적용함으로써 제1 가설 인덱스에 대응하는 제1 후보 가설이 생성될 수 있다. 제1 가설 인덱스 바로 뒤에 있는 가설 인덱스들의 재정렬된 순서의 제2 가설 인덱스에 대해, 제2 가설 인덱스에 대응하는 제2 후보 가설은 (a) 제1 가설 인덱스에 대응하는 제1 후보 가설과 (b) 제2 후보 가설에 대한 조정 항에 기초하여 생성될 수 있다. 다음 표 3은 후보 변환 계수의 개수가 3(예를 들어, L=3)인 경우, LFNST에 대한 모든 후보 가설을 생성하는 예시적인 프로세스를 나타낸다.

표 3

부호 후보	가설 인덱스	후보 가설	가설 생성
(+, +, +)	000	H000	모든 부호 후보가 양수로 설정된 계수 행렬에 역변환을 적용함.
(+, +, -)	001	H001	$H000 - C_2 * T001$
(+, -, -)	011	H011	$H001 - C_1 * T010$
(+, -, +)	010	H010	$H011 + C_2 * T001$
(-, -, +)	110	H110	$H010 - C_0 * T100$
(-, -, -)	111	H111	$H110 - C_2 * T001$
(-, +, -)	101	H101	$H111 + C_1 * T010$
(-, +, +)	100	H100	$H101 - C_2 * T001$

상기 표 3에서, 제1 열은 각각 $2^3=8$ 개의 부호 후보 조합을 예시한다. 제2 열은 각각 양수 부호(+)와 음수 부호(-)를 나타내기 위해 숫자 0과 1을 사용하여 부호 후보의 조합에 해당하는 가설 인덱스를 나타낸다. 제2 열에 있는 가설 인덱스들은 계리 코드 순서(예를 들어, 000, 001, 011, 010, 110, 111, 101, 100)에 따라 정렬된다. 제3 열은 부호 후보들의 조합과 가설 인덱스들에 각각 대응하는 후보 가설들을 예시한다. 제4 열은 각각 후보 가설의 계산을 보여준다. 표 3에서, 제 4 열의 TXYZ는 대응하는 템플릿(즉, 변환 블록의 상단 및 좌측 경계의 재구성된 샘플)을 나타내며, 이는 특정 변환 계수가 1로 설정되는 반면, 다른 모든 변환 계수는 0과 같은 변환 블록의 계수 행렬에 역변환을 적용함으로써 생성될 수 있다. 예를 들어, T100은 계수 행렬에 제1 부호 후보에 대응하는 변환 계수만 1로 설정되고 계수 행렬의 모든 변환 계수는 0으로 설정된 계수 행렬에 역변환을 적용하여 생성된 해당 템플릿을 나타낸다. C_0 , C_1 및 C_2 는 각각 제1, 제2 및 제3 부호 후보와 연관된 역양자화 변환 계수의 절대값을 나타낸다.

표 3을 참조하면, 제1 가설 인덱스(000)에 대하여, 각 후보 변환 계수가 1로 설정된 변환 블록과 연관된 계수 행렬에 역방향 2차 변환 및 역방향 주 변환을 적용하여 제1 후보 가설(H000)을 생성할 수 있다. 제1 가설 인덱스(000)의 바로 뒤에 있는 제2 가설 인덱스(001)에 대하여, (a) 제1 후보 가설(H000) 및 (b) 제2 후보 가설을 위한 조정 항(예를 들어, $-C_2 * T001$)에 기초하여 제2 후보 가설(H001)을 생성할 수 있다. 마찬가지로, 제2 가설 인덱스(001) 바로 뒤에 있는 제3 가설 인덱스(011)에 대하여, 제3 후보 가설(H011)은 (a) 제2 후보 가설(H001) 및 (b) 제3 후보 가설을 위한 조정 항(예를 들어, $-C_1 * T010$)에 기초하여 생성할 수 있다. 제3 가설 인덱스(011) 바로 뒤에 있는 제4 가설 인덱스(010)에 대하여, (a) 제3 후보 가설(H011) 및 (b) 제4 후보 가설(예를 들어, $C_2 * T001$)을 위한 조정 항에 기초하여 제4 후보 가설(H010)이 생성할 수 있다.

이어서, 복수의 후보 가설들로부터 수평, 수직, 대각선 방향을 따라 결합된 그래디언트를 포함하는 비용 함수에 기초하여 최소 비용과 관련된 가설을 결정할 수 있다. 상술한 바와 같이, 비용 함수가 수평 방향 및 수직 방향의 그래디언트(예를 들어, 상기 식 (5)에 나타난 바와 같이)만을 이용하는 경우, 고도로 불균일한 이미지 신호에 대해 잘 동작하지 않을 수 있다. 본 개시내용과 일치하게, 비용 함수의 정확성을 향상시키기 위해 하나 또는 그 이상의 대각선 방향을 따른 그래디언트가 이용된다. 예를 들어, 좌측 대각선 방향과 우측 대각선 방향을 포함하는 두 대각선 방향도 비용 함수에 통합될 수 있다. 예를 들어, 두 대각선 방향에 대한 비용 함수는 다음 식 (12) 및 (13)에 따라 설명할 수 있다:

$$\text{costD1} = \sum_{n=0}^{N-1} |2B_{-1,n-1} - B_{-2,n-2} - P_{0,n}| + \sum_{m=0}^{M-1} |2C_{m-1,-1} - C_{m-2,-2} - P_{m,0}| \quad (12).$$

$$\text{costD2} = \sum_{n=0}^{N-1} |2B_{-1,n+1} - B_{-2,n+2} - P_{0,n}| + \sum_{m=0}^{M-1} |2C_{m+1,-1} - C_{m+2,-2} - P_{m,0}| \quad (13).$$

상기 식 (8) 또는 (9)에서, $B_{-1,n-1}$, $B_{-2,n-2}$, $B_{-1,n+1}$ 및 $B_{-2,n+2}$ 는 변환 블록의 상단 이웃 블록으로부터의 이웃 샘플을 나타낸다. $C_{m-1,-1}$, $C_{m-2,-2}$, $C_{m+1,-1}$ 및 $C_{m+2,-2}$ 는 변환 블록의 좌측 이웃 블록으로부터의 이웃 샘플을 나타낸다. $P_{0,n}$ 및 $P_{m,0}$ 은 각각 변환 블록의 상단 및 좌측 경계에서의 재구성된 샘플을 나타낸다. N 및 M 은 각각 변환 블록의 폭 및 높이를 나타낸다. costD1 및 costD1 은 각각 좌측 대각선 방향 및 우측 대각선 방향에 대한 좌측 대각선 비용 함수 및 우측 대각선 비용 함수를 나타낸다.

대각선 방향의 두 비용 함수(예를 들어, costD1 , costD2)는 수평-수직 비용 함수(예를 들어, 상기 식 (5)에 나타난 비용 costHV)와 함께 사용할 수 있다. 그러면 부호 예측을 위한 비용 함수는 수평 및 수직 방향의 그래디언트를 포함하는 수평-수직 비용 함수, 좌측 대각선 방향의 그래디언트를 포함하는 좌각선 비용 함수, 우측 대각선 방향의 그래디언트를 포함하는 우각선 비용 함수에 기초하여 결정될 수 있다. 예를 들어, 비용 함수는 식 (14)에 나타난 바와 같이 수평-수직 비용 함수, 좌측 대각선 비용 함수, 우측 대각선 비용 함수의 가중합일 수 있다:

$$\text{cost} = \text{costHV} + \omega(\text{costD1} + \text{costD2}) \quad (14).$$

상기 식 (14)에서 ω 은 좌측 대각선 비용 함수와 우측 대각선 비용 함수에 대한 가중치를 나타낸다.

다른 예에서, 비용 함수는 식 (15)와 같이 수평-수직 비용 함수, 좌측 대각선 비용 함수 및 우측 대각선 비용 함수 중 최소값일 수 있다:

$$\text{cost} = \min\{\text{costHV}, \text{costD1}, \text{costD2}\} \quad (15).$$

상술한 식 (5)와 비교하여, 본 명세서에 개시된 식 (14) 또는 (15)에서의 비용 함수는 대각선 방향을 따라 비용 함수 costD1 , costD2 를 지원하기 위해 더 많은 이웃 픽셀을 필요로 할 수 있으며, 이는 도 16a 내지 도 16b를 참조하여 이하에서 보다 상세히 설명된다.

일부 구현들에서, 각각의 후보 가설에 대한 대응하는 비용은 상기 식 (14) 또는 (15)를 사용하여 결정될 수 있다. 그 다음, 복수의 후보 가설들에 대해 복수의 비용이 각각 계산될 수 있다. 상기 복수의 비용들 중 최소 비용이 결정될 수 있다. 최소 비용과 연관된 후보 가설은 복수의 후보 가설들로부터 결정되고 부호 예측을 위한 가설이 되도록 선택될 수 있다.

부호 예측 프로세스(1300)는, 부호 생성 동작(1108)을 계속 수행할 수 있는데, 여기서 선택된 가설과 연관된 부호 후보들의 조합은 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트로 결정된다. 예를 들어, 선택된 가설을 생성하기 위해 사용되는 부호 후보들의 조합(예를 들어, L 개의 부호 후보들)이 L 개의 후보 변환 계수들에 대한 예측 부호들로서 사용될 수 있다.

일부 구현예들에서, 부호 생성 동작(1308)은 또한 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 생성하기 위해 예측된 부호들의 세트에 벡터 기반 부호 예측 방식을 적용하는 것을 포함할 수 있다. 부호 시그널링 비트들의 시퀀스를 포함하는 비트스트림은 비디오 인코더(20)에 의해 생성되고 도 1의 저장 디바이스(32)에 저장될 수 있다. 대안적으로 또는 추가적으로, 비트스트림은 도 1의 링크(16)를 통해 비디오 디코더(30)로 전송될 수 있다.

상술한 바와 같이, 변환 블록 내의 변환 계수들의 부호들이 잘 예측되는 경우, 복수의 연속된 변환 계수들의 부호들이 정확하게 예측될 가능성이 매우 높다. 이 경우, 기존 부호 예측 설계들에서의 시그널링 방식은 각 변환 계수들의 대응하는 부호가 정확하게 예측될 수 있음을 개별적으로 가리키기 위해 복수의 빈들 "0"을 시그널링할 필요가 있기 때문에, 변환 블록의 부호 값들을 시그널링하기 위한 오버헤드 측면에서 분명히 비효율적이다. 기존 부호 예측 방식의 예시적인 구현에는 도 15a를 참조하여 이하에서 보다 상세히 설명된다.

본 개시내용과 일치하게, 본 명세서에 개시된 벡터 기반 부호 예측 방식을 적용함으로써 부호 시그널링의 효율을 향상시킬 수 있다. 구체적으로, 변환 블록에 대한 후보 변환 계수는 복수의 그룹으로 나눌 수 있으며, 각 그룹 내의 후보 변환 계수의 부호를 함께 예측할 수 있다. 이 경우, 그룹에 있는 후보 변환 계수의 원래 부호(또는 실제 부호)가 각각 예측된 부호와 동일한 경우, 그룹 내의 모든 부호들이 정확하게 예측됨을 가리키기 위해 값이 "0"인 빈(bin)만 비트스트림으로 전송해야 한다. 그렇지 않으면(즉, 그룹 내에 원래 부호가 예측된 부호와

상이한 후보 변환 계수가 적어도 존재함), 그룹 내의 후보 변환 계수들에 대한 모든 부호들이 정확하게 예측되지 않음을 가리키기 위해 값 "1"인 빈이 먼저 비트스트림에서 시그널링될 수 있다. 그런 다음, 그룹 내의 각각의 예측된 부호의 대응하는 정확성을 개별적으로 알리기 위해 비디오 인코더(20)로부터 비디오 디코더(30)로의 비트스트림에서 추가 빈들이 또한 시그널링될 수 있다. 본 명세서에 개시된 벡터 기반 부호 예측 방식의 예시적인 구현에는 이하에서 도 15b를 참조하여 보다 상세히 설명된다.

[0173] 일부 구현예들에서, 후보 변환 계수들의 세트는 복수의 후보 변환 계수들의 그룹들로 분할될 수 있다. 후보 변환 계수들의 각각의 그룹의 경우, 후보 변환 계수들의 그룹에 대해 하나 또는 그 이상의 부호 시그널링 비트들을 생성하여 예측된 부호들의 정확성을 가리킬 수 있다.

[0174] 일 예에서, 부호 시그널링 비트들은 후보 변환 계수들의 그룹의 원래 부호들이 후보 변환 계수들의 그룹의 예측된 부호들과 동일하지 여부에 기초하여 생성될 수 있다. 후보 변환 계수들의 그룹의 원래 부호들이 후보 변환 계수들의 그룹의 예측 부호들과 동일한 것에 응답하여, 값이 0("0")인 빈이 생성되어 부호 시그널링 비트로서 비트스트림에 추가될 수 있다. 예를 들어, 비트스트림은 후보 변환 계수들의 그룹의 예측된 부호들이 정확하게 예측됨을 가리키기 위해 "0"을 포함할 수 있다.

[0175] 한편, 후보 변환 계수들의 그룹의 원래 부호들이 후보 변환 계수들의 그룹의 예측 부호들과 동일하지 않은 것에 응답하여, 1("1")의 값을 갖는 빈이 생성될 수 있다. 후보 변환 계수들의 그룹의 예측된 부호들의 대응하는 정확성을 알리기 위한 추가 빈들의 세트도 생성될 수 있다. 그런 다음, 값이 1인 빈과 추가 빈들의 세트를 비트스트림에 부호 시그널링 비트로서 추가할 수 있다. 예를 들어, 추가 빈들의 세트는 원래 부호들 및 후보 변환 계수들의 그룹의 예측된 부호들의 XOR 결과일 수 있다. 값이 "0"인 추가 빈은 추가 빈에 대응하는 후보 변환 계수의 예측 부호가 정확하게 예측됨을 가리킬 수 있는 반면, 값이 "1"인 추가 빈은 추가 빈에 대응하는 후보 변환 계수의 예측 부호가 부정확하게 예측됨을 가리킬 수 있다. 비트스트림은, (a) 후보 변환 계수들의 그룹의 예측된 부호들이 정확하게 예측되지 않음을 가리키기 위한 "1"; 및 (b) 어떤 예측된 부호들이 정확하게 예측되고 어떤 예측된 부호들이 부정확하게 예측되는지를 가리키기 위한 추가 빈들의 세트;를 포함할 수 있다.

[0176] 부호 시그널링 비트들은 다른 컨텍스트 모델들을 이용하여 유사하게 생성될 수 있다. 예를 들어, 부호 시그널링 비트들은 후보 변환 계수들의 그룹의 예측된 부호들이 정확하게 예측되었음을 가리키는 "0"을 포함할 수 있고, 예측된 부호들이 정확하지 않음을 가리키는 1("1")의 값을 갖는 빈을 포함할 수 있다. 부정확한 부호 예측을 갖는 후보 변환 계수 그룹의 예측된 부호들의 대응하는 정확성을 알리기 위한 추가적인 빈들의 세트도 생성될 수 있다.

[0177] 일부 구현예들에서, 각 후보 변환 계수의 그룹의 크기는 하나 또는 그 이상의 미리 결정된 기준에 기초하여 적응적으로 변경될 수 있다. 하나 또는 그 이상의 미리 결정된 기준은 변환 블록의 폭 또는 높이, 변환 블록의 코딩 모드(예를 들어, 인트라 또는 인터 코딩), 및 변환 블록 내의 0이 아닌 변환 계수의 개수 등을 포함할 수 있다. 일부 구현예들에서, 각 후보 변환 계수의 그룹의 크기는 SPS, PPS, 슬라이스 또는 픽처 레벨, CTU 또는 CU 레벨, 또는 변환 블록 레벨과 같은 다양한 코딩 레벨에서 비트스트림에서 시그널링될 수 있다.

[0178] 일부 구현예들에서, 하나 또는 그 이상의 제약들이 본 명세서에 개시된 벡터-기반 부호 예측 방식의 적용 시나리오들을 제한하기 위해 적용될 수 있다. 예를 들어, 본 명세서에 개시된 벡터 기반 부호 예측 방식은 변환 블록 내의 변환 계수들 중 제1 부분 변환 계수에 대한 부호들을 처리하는 데 적용될 수 있는 반면, 변환 블록 내의 변환 계수들 중 제2 부분 변환 계수에 대한 부호들은 기존의 부호 예측 방식을 사용하여 처리될 수 있다. 추가의 예에서, 본 명세서에 개시된 벡터 기반 부호 예측 방식은 변환 블록으로부터 첫 N(예를 들어, N=2, 3, 4, 5, 6, 7, 또는 8 등)개의 0이 아닌 후보 변환 계수에 적용 가능할 수 있는 반면, 변환 블록을 형성하는 다른 후보 변환 계수의 부호는 도 15a에 도시된 기존의 부호 예측 방식을 이용하여 처리될 수 있으며, 이는 본 개시내용에서 나중에 설명될 것이다.

[0179] 본 개시내용과 일치하게, 본 명세서에 개시된 부호 예측 프로세스(1100)는 일부 시나리오들에서 디세이블될 수 있다. 예를 들어, LFNST가 인트라 템플릿 매칭 모드에 의해 코딩되는 코딩 블록들에 적용되는 경우, 주 변환은 DST-VII가 될 수 있다. ECM에서의 LFNST 코어 변환들은 주로 주 변환이 DCT-II일 때 트레이닝된다는 점을 감안하면, 인트라 템플릿 매칭 블록들의 대응하는 LFNST 변환 계수들은 다른 LFNST 블록들의 변환 계수들과 비교할 때 상이한 특성들을 나타낼 수 있다. 이러한 이론적 근거에 기초하여, 부호 예측 프로세스(1300)는 현재 코딩 블록이 인트라 템플릿 매칭 블록이고 LFNST를 사용하여 코딩될 때 디세이블될 수 있다.

[0180] 본 개시내용과 일치하게, LFNST 블록들에 대한 예측된 부호들의 최대 개수 및 비-LFNST 블록들에 대한 예측된

부호들의 최대 개수는 부호 예측의 계산 복잡도를 제어하기 위해 상이할 수 있다. 예를 들어, LFNST 블록들에 대한 예측된 부호들의 최대 수는 6(또는 4)으로 설정될 수 있는 반면, 비-LFNST 블록들에 대한 예측된 부호들의 최대 수는 6(또는 4)과 상이한 값을 가질 수 있다. 또한, LFNST를 적용하고 LFNST를 적용하지 않는 비디오 블록들에 대해 예측된 부호들의 최대 개수의 상이한 값들이 적용될 수 있다. 일부 구현예들에서, 비디오 인코더(20)는 인코더의 대응하는 복잡도 또는 성능 선호도에 기초하여 LFNST 블록들에 대한 예측된 부호들의 최대 수를 결정할 수 있고, 최대 개수를 비디오 디코더(30)에 시그널링할 수 있다. LFNST 블록들에 대한 예측된 부호들의 최대 개수가 비디오 디코더(30)에 시그널링될 때, 이는 다양한 코딩 레벨들, 예를 들어, 시퀀스 파라미터 세트(SPS), 픽처 파라미터 세트(PPS), 픽처 또는 슬라이스 레벨, 또는 CTU 또는 CU 레벨에서 시그널링될 수 있다. 일부 구현예들에서, 비디오 인코더(20)는 LFNST를 적용하고 LFNST를 적용하지 않는 비디오 블록들에 대한 예측된 부호들의 최대 개수의 상이한 값들을 결정하고 비디오 인코더(20)로부터 비디오 디코더(30)로 최대 개수의 값들을 시그널링할 수 있다.

[0181] 본 개시내용과 일치하게, 주 변환 및 2차 변환 모두의 변환 계수가 고정되어 있다는 점을 가정하면, 비디오 인코더(20) 또는 비디오 디코더(30)는 상이한 변환 블록 크기 및 주 변환 및 2차 변환 조합의 상이한 조합에 대한 템플릿(예를 들어, 템플릿의 샘플)을 미리 계산할 수 있다. 비디오 인코더(20) 또는 비디오 디코더(30)는 최적화된 구현을 위해 템플릿들의 샘플들을 즉석에서 생성하는 복잡성을 피하기 위해 템플릿들(예를 들어, 템플릿들의 샘플들)을 내부 또는 외부 메모리에 저장할 수 있다. 템플릿들의 샘플들은 저장 크기와 샘플 정밀도 사이의 상이한 트레이드오프들(tradeoffs)을 달성하기 위해 상이한 분수 정밀도들로 저장될 수 있다. 예를 들어, 비디오 인코더(20) 또는 비디오 디코더(30)는 고정 팩터(예를 들어, 64, 128, 256)로 템플릿들의 플로팅 샘플들을 스케일링하고 스케일링된 샘플들을 가장 가까운 정수로 반올림할 수 있다. 반올림된 샘플들은 메모리에 저장될 수 있다. 그런 다음, 템플릿을 사용하여 후보 가설을 재구성하는 경우, 후보 가설에서 생성된 샘플들이 정확한 동적 범위에 있음을 보장하기 위하여 해당 샘플의 크기를 먼저 원래 정밀도로 축소할 수 있다.

[0182] 도 14는 본 개시내용의 일부 구현예들에 따른 템플릿들의 선형 조합에 기초한 예시적인 가설 생성을 나타내는 그래픽 표현이다. 도 14에서, 4개의 패터닝된 블록 0-3은 그 부호가 예측될 후보 변환 계수들을 나타낸다. 팩터 C_0 , C_1 , C_2 , 및 C_3 는 4개의 후보 변환 계수들의 역양자화된 변환 계수들의 대응하는 값을 나타낸다. 템플릿 0-3은 4개의 후보 변환 계수들 0-3에 각각 대응할 수 있다. 예를 들어, 후보 변환 계수 0에 대응하는 템플릿 0은 변환 블록에 역방향 2차 변환 및 역방향 주 변환을 적용하여 생성될 수 있으며, 여기서 후보 변환 계수 0은 1로 설정되고, 변환 블록 내의 나머지 후보 변환 계수들은 0으로 설정된다. 마찬가지로, 템플릿 1-3은 각각 생성될 수 있다. 후보 가설은 템플릿 0-1을 각각 가중치 C_0 - C_3 와 가산함으로써 생성될 수 있다.

[0183] 도 15a는 일부 예들에 따른 기존의 부호 예측 방식의 예시적인 구현을 도시한 그래픽 표현이다. 도 15b는 본 개시내용의 일부 구현예들에 따른 벡터 기반 부호 예측 방식의 예시적인 구현을 도시한 그래픽 표현이다. 기존의 부호 예측 방식과 본 명세서에 개시된 벡터 기반 부호 예측 방식의 예시적인 비교가 도 15a 내지 도 15b를 참조하여 본 명세서에 예시된다.

[0184] 도 15a 내지 도 15b에서, 부호 예측을 위한 후보 변환 계수들로서 선택되는 변환 블록 내의 6개의 0이 아닌 변환 계수들이 존재한다. 후보 변환 계수들은 래스터 스캔 순서를 사용하여 변환 블록의 계수 행렬로부터 스캔된다. 후보 변환 계수들의 원래 부호들 및 예측된 부호들도 도 15a 내지 도 15b에 도시된다. 예를 들어, 값이 "-2"인 제1 후보 변환 계수의 원래 부호 및 예측된 부호는 모두 "-"(도 15a 내지 도 15b에서 "1"로 표시됨)이다. 값이 "3"인 제2 후보 변환 계수의 원래 부호 및 예측된 부호는 모두 "+"(도 13a 내지 도 13b에서 "0"으로 표시됨)이다. 값이 "1"인 제3 후보 변환 계수의 원래 부호 및 예측된 부호는 각각 "+" 및 "-"(도 13a 내지 도 13b에서 각각 "0" 및 "1"로 표시됨)이다. 제3 후보 변환 계수의 원래 부호는 잘못 예측된다. 도 15a 내지 도 15b에 도시된 바와 같이, 제3 변환 계수를 제외한 모든 다른 후보 변환 계수들의 원래 부호들은 대응하는 예측 부호들(즉, 정확하게 예측됨)과 동일하다.

[0185] 도 15a를 참조하면, 총 6개의 빈(즉, 0, 0, 1, 0, 0, 0)이 생성되며, 각 빈은 후보 변환 계수에 대응한다. 6개의 빈은 원래 부호와 6개의 후보 변환 계수의 예측된 부호 사이에서 XOR 연산을 수행하여 생성할 수 있다. 6개의 빈은 6개의 예측된 부호의 대응하는 정확성을 가리키는데 사용될 수 있다. 예를 들어, 제1 빈 및 제2 빈의 값이 각각 "0"이면 제1 및 제2 후보 변환 계수에 대한 예측된 부호가 정확함을 가리킨다. 값이 "1"인 제3 빈은 제3 변환 계수에 대한 예측된 부호가 정확하지 않음을 가리킨다. 6개의 빈은 엔트로피 코딩을 위해 CABAC로 전송될 수 있다.

[0186] 도 15b를 참조하면, 본 명세서에 개시된 벡터 기반 부호 예측 방식은 6개의 후보 변환 계수를 3개의 그룹으로

나누고, 각 그룹은 2개의 연속된 후보 변환 계수를 포함한다. 그룹 #0 및 #2의 후보 변환 계수의 부호는 정확하게 예측될 수 있기 때문에, 2개의 그룹에 대해 각각 값이 "0"인 2개의 빈만 생성된다. 그룹 #1의 경우, 부호를 정확하게 예측할 수 없는 제3 후보 변환 계수를 포함하기 때문에, 그룹이 원래의 예측 부호와 상이한 후보 변환 계수를 적어도 포함함을 가리키기 위해 비트스트림에 값이 "1"인 빈(도 15b에서 밑줄 친 부분)을 생성하고 시그널링한다. 이어서, 그룹 #1의 제3 및 제4 계수에 대해 값이 "1" 및 "0"인 2개의 추가 빈을 생성하여 이들의 부호가 정확하게 예측될 수 있는지 여부를 가리킨다. 이에 상응하여, 본 명세서에 개시된 벡터 기반 부호 예측 방식이 적용될 때 CABAC를 위해 생성되는 빈은 총 5개이며, 이는 도 15a에 도시된 기존 부호 예측 방식에 의해 생성되는 빈보다 적은 비트를 갖는다. 따라서, 본 명세서에 개시된 벡터 기반 부호 예측 방식이 적용됨으로써, 시그널링 오버헤드가 감소될 수 있고, 변환 블록의 코딩 효율을 향상시킬 수 있다.

[0187] 본 개시내용과 일치하게, 비록 래스터 스캔 순서가 도 15b에 도시된 바와 같이 변환 블록의 계수 행렬로부터 후보 변환 계수들을 얻기 위해 사용되지만, 임의의 다른 스캔 순서가 또한 부호 예측을 위한 후보 변환 계수들을 선택하는데 사용될 수 있다. 예를 들어, 후보 변환 계수들은 상술한 하나 또는 그 이상의 선택 기준에 기초하여 선택될 수 있다. 유사한 설명은 본 명세서에서 반복되지 않는다.

[0188] 도 16a는 본 개시내용의 일부 구현예들에 따른 좌측 대각선 방향으로의 좌측 대각선 비용 함수의 예시적인 계산을 나타내는 그래픽 표현이다. 도 16b는 본 개시내용의 일부 구현예들에 따른 우측 대각선 방향으로의 우측 대각선 비용 함수의 예시적인 계산을 나타내는 그래픽 표현이다. costHV의 계산을 위한 상기 식 (8)과 비교하여, 상기 식 (14) 또는 (15)에 도시된 좌측 대각선 방향으로의 비용 함수 costD1 또는 우측 대각선 방향으로의 비용 함수 costD2는 대각선 방향을 따른 비용 함수 costD1, costD2의 계산을 지원하기 위해 더 많은 이웃 픽셀을 필요로 할 수 있다(도 16a 내지 도 16b의 영역(1602, 1604 및 1606) 내에 표시된 픽셀로서 도시됨). 영역(1602, 1604, 1606) 내의 이러한 픽셀들이 이용가능하지 않는 경우, 가장 가까운 패딩 방법을 채택하여 이러한 이용 불가능한 위치를 채울 수 있다. 예를 들어, 영역(1606) 내의 $B_{-1,4}$ 를 이용 불가능한 경우, $B_{-1,4}$ 에 가장 가까운 이용 가능한 픽셀인 $B_{-1,3}$ 은 $B_{-1,4}$ 의 위치를 채우는 데 사용된다(예를 들어, $B_{-1,4} = B_{-1,3}$). 영역(1602) 내의 $B_{-1,-1}$ ($C_{-1,-1}$ 로도 표시됨), $B_{-1,-2}$ ($C_{-1,-1}$), $B_{-2,-1}$ ($C_{-1,-1}$) 및 $B_{-2,-2}$ ($C_{-1,-1}$)가 이용 불가능한 경우, 이용 불가능한 위치들을 채우기 위한 몇 가지 예시적인 방법들이 본 명세서에 개시된다.

[0189] 제1 예시적인 방법에서, 각각의 이용 불가능한 위치는 아래의 식 (16) 내지 식 (19)에 나타난 바와 같이 가장 가까운 이용 가능한 픽셀들에 가중치를 부여함으로써 채워질 수 있다:

[0190]
$$B_{-2,-2} = (B_{-2,0} + C_{0,-2}) * 0.5 \quad (16)$$

[0191]
$$B_{-2,-1} = (2 * B_{-2,0} + C_{0,-1})/3 \quad (17)$$

[0192]
$$B_{-1,-2} = (B_{-1,0} + 2 * C_{0,-2})/3 \quad (18)$$

[0193]
$$B_{-1,-1} = (B_{-1,0} + C_{0,-1}) * 0.5 \quad (19).$$

[0194] 제2 예시적인 방법에서, 이용 불가능한 위치들 중 일부는 그들의 가장 가까운 이용 가능한 픽셀들로 각각 채워질 수 있다. 예를 들어, 영역(1602)에서의 $B_{-1,-2}$ 가 이용 불가능한 경우, $C_{0,-2}$ 로 채워진다. $B_{-2,-1}$ 이 이용 불가능한 경우, $B_{-2,0}$ 로 채워진다. 그러나, $B_{-2,-2}$ 및 $B_{-1,-1}$ 이 이용 불가능한 경우, 상기 식 (16) 및 식 (17)에 따라 계산된 가장 가까운 2개의 이웃 픽셀들의 평균으로 채워질 수 있다.

[0195] 제3 예시적인 방법에서, 이용 가능한 이웃 재구성된 샘플들만이 비용 함수를 계산하는데 사용된다. 현재 블록의 상단/좌측 경계를 따른 하나의 경계 샘플의 비용 계산이 임의의 이용 불가능한 재구성된 샘플을 포함하는 경우, 이들은 해당 방향의 비용 계산에 사용되지 않을 것이다. 예를 들어, 도 16b에서는 경계 샘플 $P_{0,0}$, $P_{0,1}$ 및 $P_{1,0}$ 만이 우측 상단 방향의 비용 함수 값을 계산하는 데 사용되는 반면, $P_{0,2}$, $P_{0,3}$, $P_{2,0}$ 및 $P_{3,0}$ 은 비용 계산이 이용 가능하지 않은 적어도 하나의 기준 샘플을 참조한다는 점을 고려할 때 사용되지 않을 것이다.

[0196] 본 개시내용과 일치하게, 좌측 대각선 및 우측 대각선(즉, 도 16a 내지 도 16b에 도시된 135° 및 45°)이 상기 식 (14) 또는 식 (15)에 도시된 비용 함수의 계산에서 예시 목적으로 사용되지만, 임의의 다른 측정 컴포넌트(예를 들어, 하나 또는 그 이상의 임의의 방향을 따른 연속성 측정)가 부호 예측을 위한 비용 함수의 계산에 통합될 수 있음이 고려된다.

- [0197] 제4 구현예에서, 하나의 현재 블록의 이웃 샘플들 중 텍스처 정보의 그래디언트 분석(gradient analysis)에 기초하는 샘플 추정법(sample extrapolation method)이 부호 예측을 위한 비용 함수의 정밀도를 향상시키기 위해 구현될 수 있다. 항상 고정된 추정 방향(예를 들어, 상기 이웃 샘플에 대한 수직 추정 및 수평 이웃 샘플에 대한 수평 추정)을 사용하는 대신, 현재 블록의 상단 및 좌측 재구성된 샘플의 텍스처 분석이 인코더 및 디코더 모두에서 수행될 수 있으며, 이웃 샘플의 그래디언트의 가장 주요한 방향이 현재 블록의 상단 및 좌측 경계를 따라 경계 샘플을 추정하도록 선택될 수 있다.
- [0198] 예를 들어, 도 17은 본 개시내용의 일부 구현예들에 따른 현재 블록의 이웃하는 재구성된 샘플들 내에서 도미넌트 그래디언트 방향을 캡처하기 위한 방법(1700)의 흐름도이다. 방법(1700)은 비디오 인코더(20) 또는 비디오 디코더(30)와 연관된 비디오 프로세서에 의해 구현될 수 있고, 후술되는 바와 같은 단계(1702) 내지 단계(1712)를 포함할 수 있다. 단계들 중 일부는 본 명세서에 제공된 개시를 수행하기 위해 선택적일 수 있다. 또한, 단계들 중 일부는 동시에 수행될 수 있고, 도 17에 도시된 것과 다른 순서로 수행될 수도 있다.
- [0199] 단계(1702)에서, 그래디언트 도출을 위하여 현재 블록에 대해 기준 샘플들을 선택한다. 일부 구현예들에서, 기준 샘플들은 템플릿을 형성한다. 예를 들어, 도 18a는 본 개시내용의 일부 구현예들에 따른 비용 함수에 대한 샘플 추정 방향의 그래디언트 기반 선택에서 예시적인 템플릿 샘플들 및 그래디언트 필터 윈도우들을 나타내는 그래픽 표현이다. 도 18a에 도시된 바와 같이, 현재 블록에 상단 및 좌측의 이웃하는 재구성된 샘플들의 N개의 행들 및 열들의 템플릿이 그래디언트 도출을 위한 기준 샘플들로서 사용된다. 도 18a의 예에서, 템플릿 크기는 3과 동일하다.
- [0200] 단계(1704)에서, 그래디언트 히스토그램(Histogram of Oriented Gradients: HoG)을 초기화한다. 예를 들어, HoG는 복수의 엔트리들로 생성될 수 있으며, 여기서, 그래디언트 히스토그램의 각각의 엔트리는 미리 결정된 각도 방향에서의 그래디언트들의 누적 크기에 대응한다. 각각의 엔트리는 0으로서 초기화될 수 있다. 예를 들어, 도 18b는 본 개시내용의 일부 구현예들에 따른 비용 함수에 대한 샘플 추정 방향의 그래디언트 기반 선택에서 예시적인 그래디언트 히스토그램(HoG)을 나타내는 그래픽 표현이다. 실제로, 상이한 미리 결정된 방향들이 개시된 그래디언트 분석 방식에서 사용될 수 있다. 일 실시예에서, VVC/ECM에서 규칙적인 인트라 예측의 65 개 각도 방향들에 대해 정의된 것과 동일한 방향들이 사용된다.
- [0201] 단계(1706)에서, 그래디언트 필터 윈도우를 기준 샘플들에 적용하여 해당 그래디언트들을 계산한다. 도 18a에 도시된 바와 같이, 하나의 NXN 그래디언트 필터 윈도우가 템플릿의 중간 행/열에 위치한 각각의 템플릿 샘플에 적용되어(즉, 필터 윈도우가 샘플 위치의 중심에 있음), 그 대응하는 수평 그래디언트 G_h 및 수직 그래디언트 G_v 를 각각 계산한다.
- [0202] 단계(1708)에서, 각 기준 샘플의 그래디언트의 각도(Angle) 및 크기(Amp)를 계산한다. 예를 들어, 샘플의 그래디언트는 식 (20) 및 식 (21)에 따라 계산될 수 있다:
- [0203]
$$\text{Angle} = \tan^{-1}\left(\frac{G_h}{G_v}\right) \quad (20)$$
- [0204]
$$\text{Amp} = |G_h| + |G_v| \quad (21)$$
- [0205] 단계(1710)에서, 그래디언트의 각도를 미리 정의된 방향들 중 하나로 변환할 수 있고, HoG 내의 대응하는 엔트리를 업데이트할 수 있다. 예를 들어, 도 18에 도시된 바와 같이, 각 각도에 대한 HoG의 진폭은 그 각도의 샘플 그래디언트의 크기(Amp)를 추가하여 업데이트된다. 결과적으로 진폭은 누적된 크기이다.
- [0206] 단계(1712)에서, 가장 큰 누적 크기를 가지는 HoG의 엔트리가 현재 블록의 비용 함수를 위한 이웃 샘플들을 추정하는데 사용되는 방향으로 선택될 것이다. 예를 들어, 도 18에 도시된 바와 같이, 가장 큰 엔트리가 동그라미로 표시된다.
- [0207] 상기 방법에서, 가장 큰 크기를 가지는 방향을 샘플 추정 방향으로 선택한다. 이러한 방법은 일부 노이즈(예를 들어, 다른 코딩 모듈에 의해 야기되는 양자화 에러 및/또는 코딩 노이즈)가 있을 때 반드시 신뢰할 수 있는 것은 아닐 수 있다. 이러한 문제를 해결하기 위해, 도미넌트 그래디언트 방향이 추정 방향으로 채택되기 전에 특정 조건이 적용될 수 있다.
- [0208] 예를 들어, 일 구현예에서, 선택된 도미넌트 그래디언트 방향은 선택된 그래디언트 방향에 속하는 템플릿 샘플이 충분히 있는 경우(예를 들어, 선택된 방향에 속하는 샘플의 백분율이 미리 결정된 임계값을 초과하는 것과 같이 충분히 큰 경우)에만 비용 함수 계산의 샘플 추정을 위해 인에이블될 수 있다. 그렇지 않으면(예를 들어,

선택된 방향에 속하는 템플릿 샘플의 수가 충분히 크지 않은 경우), 디폴트 추정(예를 들어, 상기 이웃에 대한 수직 추정 및 좌측 이웃에 대한 수평 추정)이 여전히 적용된다.

[0209] 다른 구현예에서, 선택된 도미넌트 그래디언트는 도미넌트 그래디언트 방향과 연관된 그래디언트 크기가 충분히 큰 경우(예를 들어, 모든 그래디언트 방향들의 크기의 합에 대한 선택된 그래디언트 방향의 크기의 비율이 다른 미리 결정된 임계값보다 큰 경우)에만 비용 함수 계산의 샘플 추정을 위해 인에이블될 수 있다. 그렇지 않으면(예를 들어, 선택된 방향의 그래디언트 크기가 유의하지 않음), 디폴트 추정이 여전히 적용된다.

[0210] 또 다른 구현예에서, 상기 제약들은 함께 적용된다. 즉, 선택된 방향은 선택된 방향과 연관된 템플릿 샘플들의 개수가 충분히 많고 그래디언트 크기가 충분히 큰 경우에만 현재 블록의 샘플 추정에 대해 인에이블된다. 그렇지 않으면, 디폴트 추정은 여전히 적용된다.

[0211] 또 다른 구현예에서, 선택된 도미넌트 그래디언트 방향이 인에이블될 때, 선택된 방향에 직교하는 방향은 하나의 블록 내의 부호를 예측할 때 비용 함수의 계산에서 샘플 추정에 사용된다. 예를 들어, 선택된 방향이 45도인 경우, 현재 블록의 상단 및 좌측 경계를 따라 샘플을 추정하는 방향으로 135도가 사용될 것이다. 마찬가지로, 선택된 방향이 135도인 경우, 45도는 현재 블록의 상단 및 좌측 경계를 따라 샘플들을 추정하는 방향으로 사용될 것이다.

[0212] 상기 개시된 구현예들에서, 부호 예측은 하나의 변환 블록의 좌측 상단 4×4 서브-블록들(예를 들어, 도 7에 나타난 바와 같이)에 위치되는 계수들의 부호들을 예측하는 데에만 적용될 수 있다. 일반적으로, 이러한 방법은 공통 변환 블록들의 에너지의 대부분이 보통 저주파에서 소수의 변환 계수들에 집중되기 때문에 합리적이다. 이에 대응하여, 좌측 상단 코너들에서의 변환 계수들의 부호들은 통계적으로 다른 위치들에서의 변환 계수들보다 더 용이하다. 그러나, 이러한 가정이 항상 정확한 것은 아닐 수 있다. 예를 들어, 복잡한 모션 필드를 갖는 인터 블록들(예를 들어, 하나의 인터 블록이 복수의 서브 블록들로 분할되고 각각의 서브 블록이 자체의 MV를 가질 수 있는 서브 블록 인터 모드)의 경우, 그들의 예측 신호들에서 상이한 모션들 사이의 경계를 따라 많은 예지들이 생성될 수 있다. 이 경우, DCT/DST 변환을 적용한 후, 무시할 수 없는 고주파 변환 계수들이 생성되고 해당 계수들이 좌측 상단 4×4 코너들로부터 두드러질 수 있다. 이러한 고주파 계수들의 부호들은 상기 개시된 부호 예측 설계들에 따라 예측될 수 없다.

[0213] 일부 대안적 구현예들에서, 순방향 LFNST에 입력되어 나중에 부호 예측에 사용되는 변환 계수들의 영역은 코딩 성능을 추가로 향상시키기 위해 소수의 좌측 상단 4×4 서브-블록들(예를 들어, 도 7에 도시된 바와 같이)로부터 확장될 수 있다. 예를 들어, 도 19는 일부 예들에 따른 변환 계수들의 부호들을 예측하기 위한 부호 예측 영역을 나타내는 그래픽 표현이다.

[0214] 구체적으로, 현재 변환 블록의 좌측 상단 $A \times B$ 영역의 부호들을 선택하여 예측할 수 있으며, 여기서 M 및 N 은 식 (22)에 따라 계산된다:

[0215]
$$A = \min(\text{width}, TH)$$

[0216]
$$B = \min(\text{height}, TH) \quad (22).$$

[0217] 여기서 width 와 height 는 변환 블록의 폭과 높이이고 TH 는 부호 예측을 위한 최대 영역 크기(영역 크기 임계값이라고도 함)이다. TH 의 값을 조작하면 부호 예측 기술에 대해 상이한 성능 및 복잡성 트레이드 오프(trade-off)를 제공할 수 있다. 부호 예측의 영역을 확대하면 부호를 예측할 변환 계수의 수가 증가할 수 있지만, 인코더와 디코더 모두에서 테스트해야 하는 부호 조합의 수가 증가하기 때문에 계산 복잡성이 증가할 수 있다.

[0218] 일부 구현예들에서, TH 의 값을 결정하기 위해 상이한 방법들이 사용될 수 있다. 하나의 구현예에서, 고정된 값(예를 들어, 8, 16, 32 등)이 모든 시퀀스 및 코딩 시나리오들에서 TH 에 대해 사용될 수 있다. 영역 크기 임계값 TH 가 고정되어 있는 경우, 인코더 및 디코더는 부호 예측을 위한 변환 계수들을 위치시킬 때 동일한 값(즉, 시그널링되지 않음)을 사용하여 해당 영역을 결정한다. 다른 구현예에서, 인코더는 비디오 시퀀스의 특정 특성뿐만 아니라 그 바람직한 성능/복잡성 트레이드 오프에 따라 최적의 영역 크기 임계값을 결정하는 유연성이 주어질 수 있고, 인코더로부터 디코더로 대응하는 값을 시그널링할 수 있다.

[0219] 일 예시적인 방법에서, 영역 크기 임계값은 어떤 크기가 선택되는지를 가리키기 위해 비트스트림에서 고정된 수의 비트들만이 디코더로 시그널링되지만 하면 되도록 고정된 수의 세트로부터 선택될 수 있다. 예를 들어, 허용된 영역 크기 임계값 {4, 8, 16, 32}이 4개 있다고 가정하면, 그룹으로부터 선택된 특정 값을 가리키기 위해 단

2개의 비트만이 시그널링되면 된다. 다른 예시적인 방법에서, 영역 크기 임계값은 인코더 측의 변환 블록에 대해 적응적으로 결정될 수 있다. 따라서, 영역 크기 임계값의 값은 임의의 숫자일 수 있다. 이 경우, 그 알 수 없는 값으로 인해, 고정-길이 코드워드를 사용할 수 없고, 대신에, 일부 가변-길이 코드워드(예를 들어, 지수 곱셈, 단항 코드 등)가 비트스트림에서 결정된 영역 크기 임계값의 값을 가리키기 위해 적용될 수 있다.

[0220] 한편, TH 의 값이 비트스트림으로 전송될 때, 상이한 레벨들, 예를 들어, 비디오 파라미터 세트(VPS), 시퀀스 파라미터 세트(SPS), 픽처 파라미터 세트(PPS), 슬라이스 헤더 및 코딩 블록 레벨에서 시그널링될 수 있다. 시퀀스 레벨(예를 들어, VPS 또는 SPS)에서 시그널링될 때, 비디오 시퀀스 내의 모든 변환 블록들에 대해 하나의 동일한 영역 크기 임계값이 공유됨을 의미한다. 픽처/슬라이스 레벨(즉, PPS 또는 슬라이스 헤더)에서 시그널링될 때, 영역 크기 임계값이 픽처에서 픽처로 또는 슬라이스에서 슬라이스로 각각 달라질 수 있다. 유사하게, 값이 코딩 블록들에서 시그널링될 때, 이는 부호 예측 영역들의 최대 적응 능력을 제공하지만, 값이 각각의 개별 코딩 블록에 대해 개별적으로 시그널링되어야 하는 것을 감안할 때 더 많은 코딩 비트들을 소비한다.

[0221] 디코더 측에서, 디코더는 변환 블록의 크기 및 영역 크기 임계값 TH 에 기초하여 부호 예측 영역을 결정할 수 있다. TH 가 고정된 값인 경우, 비트스트림에서 시그널링되지 않고 디코더에서 미리 프로그래밍될 수 있다. 디코더는 모든 시퀀스 및 시나리오에 대해 이 미리 프로그래밍된 고정된 값을 사용할 수 있다. 그렇지 않으면, TH 가 고정된 값이 아니라 인코더에 의해 결정된 값인 경우, 비트스트림에서 디코더로 시그널링될 것이다. 따라서, 디코더는 변환 블록의 크기 및 시그널링된 영역 크기 임계값 TH 에 기초하여 부호 예측 영역을 결정한다.

[0222] 일부 구현예들에서, 디코더는 먼저 어느 레벨에서 영역 크기 임계값이 시그널링되는지를 결정할 수 있다. 상술한 바와 같이, TH 는 VPS, SPS, PPS, 슬라이스 헤더 및 코딩 블록 레벨 등과 같은 상이한 레벨로 시그널링될 수 있다. 시그널링되는 레벨에 기초하여, 디코더는 시그널링되는 TH 가 적용되어야 하는 정도를 결정할 수 있다. 예를 들어, TH 가 VPS 또는 SPS 레벨에서 시그널링될 때, 디코더는 비디오 시퀀스의 모든 변환 블록에 TH 를 적용할 것이다. PPS 또는 슬라이스 헤더 레벨에서 TH 가 시그널링될 때, 디코더는 현재 픽처/슬라이스 내의 모든 변환 블록들에 TH 를 적용하고, 각각의 상이한 픽처/슬라이스에 대해 상이한 TH 를 판독할 것이다. 코딩 블록 레벨에서 TH 가 시그널링될 때, 디코더는 비트스트림으로부터 판독된 상이한 TH 를 각각의 상이한 코딩 블록에 적용할 것이다.

[0223] 영역 크기 임계값이 변환 블록에 대해 적응적으로 결정된 값인 경우, 디코더는 비트스트림에서 시그널링된 코드워드에 기초하여 TH 의 값을 결정할 수 있다. 상술한 바와 같이, TH 는 미리 결정된 값들의 그룹으로부터 선택되는 경우 고정-길이 코드워드를 이용하여 시그널링될 수 있거나, 변환 블록에 대해 적응적으로 결정되는 임의의 수인 경우 가변-길이 코드워드를 이용하여 시그널링될 수 있다.

[0224] 일부 구현예들에서, 디코더는 현재 변환 블록에 적용될 TH 의 값을 결정한 후, 변환 블록의 크기 및 영역 크기 임계값 TH (예를 들어, 식 (22)에 따라)에 기초하여 부호 예측 영역을 결정할 수 있다. 예를 들어, 도 19에 도시된 바와 같이, 부호 예측 영역의 폭(A)은 변환 블록의 폭(width) 및 영역 크기 임계값 TH 중 작은 것으로 결정되고, 부호 예측 영역의 높이(B)는 변환 블록의 높이(height) 및 영역 크기 임계값 TH 중 작은 것으로 결정된다.

[0225] 확장된 부호 예측 영역 방식은 상술한 부호 예측 기술들 중 어느 것도 간섭하지 않는 것으로 고려되며, 이는 용이하게 적용될 수 있고 확장된 부호 예측 영역과 결합되어 향상된 코딩 성능을 제공할 수 있다. 하나의 구체적인 예로서, 확장된 부호 예측 영역은 부호 예측 재정렬 및 LFNST 부호 예측과 결합될 수 있다.

[0226] 전술한 설명들에서, 하나의 컨텍스트 모델은 부호 시그널링 비트들에서 예측된 부호들의 정확성을 코딩하기 위한 예로서 기술된다. 특히, 각 예측된 부호 간의 차이는 코딩될 수 있으며, 대응하는 참 부호는 "0"(또는 "1")을 가져 추정된 부호가 참 부호와 동일함(또는 동일하지 않음)을 가리킬 수 있다. 본 개시내용의 일부 구현예들과 일치하게, 코딩 효율성을 더욱 향상시키기 위해, 다른 컨텍스트 모델을 사용하여 부호 예측의 정확성을 엔트로피-코딩할 수 있다.

[0227] 제1 예시적인 방법에서, 변환 계수들의 레벨 값(즉, 크기(magnitude))의 컨텍스트는 부호 예측의 정확성을 엔트로피-코딩하는데 사용될 수 있다. 크기가 클수록 정확한 예측을 할 확률이 높아지는 경향이 있기 때문이다. 예를 들어, 이러한 방법이 적용될 때, 가능한 크기 값들의 범위는 여러 세그먼트들로 분할될 수 있고, 상이한 컨텍스트들이 서로 다른 세그먼트들에서의 계수들의 부호 예측 엔트로피-코딩에 할당될 수 있다.

[0228] 일부 구현예들에서, 변환 계수들의 크기(magnitude)의 범위는 복수의 크기 세그먼트들로 분할되고, 각 크기 세그먼트에 컨텍스트를 할당하여 크기 세그먼트에 속하는 변환 계수들에 대한 부호 시그널링 비트들을 엔트로피-

인코딩할 수 있다. 복수의 크기 세그먼트들 중에서, 연속하는 2개의 크기 세그먼트들마다, 상기 크기 세그먼트들이 중첩되지 않도록 임계치에 의해 분리될 수 있다.

[0229] 하나의 특정한 예에서, 임계값 $thres$ 는 변환 계수 크기의 범위를 2개의 세그먼트들, 즉 (1, $thres$) 및 [$thres$, $+\infty$)로 분할하는데 사용될 수 있다. 이러한 분할을 통해 하나의 변환 계수가 제1 세그먼트에 속하는 경우, 하나의 제1 컨텍스트가 변환 계수의 부호를 코딩하도록 할당된다. 그렇지 않으면, 변환 계수가 제2 세그먼트에 속하는 경우, 하나의 제2 컨텍스트가 변환 계수의 부호를 코딩하도록 할당된다.

[0230] 다양한 구현예들에서, 임계값의 값(또는 임계값들의 값들)은 상이한 방식으로 결정되고 특정될 수 있다. 하나의 예시적인 방법에서, 값은 고정될 수 있고(예를 들어, 2, 3, ... 등), 인코더 및 디코더 모두에서 미리 프로그래밍될 수 있다. 따라서, 고정된 값은 시그널링할 필요 없이 인코더 및 디코더 측 모두에서 사용된다. 다른 예시적인 방법에서, 인코더는 최적의 임계값을 선택하는 자유가 주어지고, 대응하는 값을 디코더에 시그널링할 수 있다.

[0231] 상술한 바와 같이, 각 변환 계수의 크기는 인코더 측에서 변환 계수의 부호 값을 코딩하기 위한 컨텍스트를 결정하는데 사용된다. 따라서, 디코더 측에서는 변환 계수의 부호 값을 디코딩하기 위해, 그에 따라 컨텍스트를 결정하는데 각 변환 계수의 크기를 적절하게 사용한다. 변환 계수가 디코더 측에서 역양자화가 수행된 후에만 획득될 수 있음을 가정할 때, 이러한 방식은 파싱 단계에서 역양자화가 수행될 필요가 있다는 사실로 인해 비트스트림 파싱의 복잡성을 크게 증가시킬 수 있다. 이러한 복잡성 문제를 해결하기 위해, 하나의 구현예에서, 디코더는 변환 계수의 레벨 k (즉, 역양자화가 적용되지 않고 비트스트림으로부터 직접 파싱된 변환 계수의 양자화된 값)를 직접 사용하여 하나의 변환 계수의 부호를 코딩하는 컨텍스트를 결정할 수 있다.

[0232] 다른 구현예에서, 비트스트림으로부터 파싱되는 레벨 k 를 직접 사용하는 대신에, 디코더는 양자화 인덱스(즉, $quantIdx$)를 사용하여 하나의 변환 계수의 부호를 인코딩하는 컨텍스트를 결정할 수 있다. 설명된 바와 같이, VVC/ECM에서, 양자화/역양자화를 위해 2개의 스칼라 양자화기가 적용될 수 있다. 하나의 변환 계수에 사용되는 실제 양자화기는 현재 변환 계수에 선행하는 변환 계수의 패리티에 의해 결정된다. 따라서, 변환 계수들에 대한 양자화 인덱스들은 부호들을 인코딩하기 위한 컨텍스트를 결정하는데 사용될 수 있다. 앞서 설명한 바와 같이, 양자화 인덱스는, 예를 들어, 식 (6)에 따라, 현재 변환 계수의 전이 상태(즉, $state$)에 기초하여 레벨 k 의 값으로부터 계산될 수 있다.

[0233] 다양한 구현예들에서, 상술한 모든 크기/레벨(magnitudes/levels) 기반 컨텍스트 모델링 방법들은 도 13과 관련하여 설명된 바와 같이 개시된 변환 계수 선택 방식들과 결합될 수 있다. 예를 들어, 이러한 조합은 2개의 상이한 방법으로 구현될 수 있다. 일 방법에서는, 변환 계수들이 재정렬될 수 있고, 재정렬된 변환 계수(즉, 변환 계수들이 재정렬된 후의 변환 계수)들의 크기/레벨들을 사용하여 하나의 변환 블록에서 변환 계수들의 부호들을 코딩하는데 적용되는 해당 컨텍스트를 결정할 수 있다. 다른 방법에서는, 변환 계수들이 재정렬되지 않고, 변환 계수들의 크기/레벨들을 재정렬하지 않고 사용(즉, 변환 계수들이 재정렬되기 전의 변환 계수들)하여 변환 블록에서 변환 계수들의 부호를 코딩하는데 적용되는 컨텍스트들을 결정한다.

[0234] 도 20a 및 도 20b는 각각 상술한 두 가지 방식을 구현하는 변환 계수 부호 예측을 위한 컨텍스트 모델링의 두 가지 예를 나타낸다. 도 20a에 도시된 바와 같이, 제1 방식이 적용될 때, 변환 계수들은 그들의 크기/레벨에 따라, 예를 들어, 6, 3, 2, 2, 1, 1, 1, 및 1의 순서로 재정렬된다. 재정렬된 변환 계수들의 크기/레벨들(magnitudes/levels)은 비트스트림에 포함되는 부호 시그널링 비트들의 컨텍스트들을 선택하는데 사용된다. 이러한 제1 방식의 이점은 컨텍스트 모델링에 사용되는 해당 크기/레벨들이 부호가 코딩되는 변환 계수와 일치한다는 것이다. 이러한 제1 방식이 적용될 때, 인코더/디코더는 부호들이 코딩/디코딩되기 전에 변환 계수들을 재정렬할 필요가 있는데, 이는 비트스트림 파싱의 복잡성을 증가시켜 파싱 처리량(parsing throughput)을 감소시킬 수 있다. 도 20b는 이러한 파싱 문제를 회피할 수 있는 제2 방식의 구현을 나타낸다. 이러한 제2 방식에서, 재정렬되지 않은 변환 계수의 크기/레벨들은 부호들을 인코딩하기 위한 컨텍스트 결정에 사용된다. 따라서, 재정렬은 파싱 단계에서 적용될 필요가 없다. 즉, 컨텍스트 모델링에 사용되는 크기/레벨들은 부호들과 오정렬(misalign)될 수 있다.

[0235] 제2 방법에서, 변환 계수들의 스캔 위치들의 컨텍스트는 부호 예측을 엔트로피-코딩하는데 사용될 수 있다. 예를 들어, 하나의 변환 블록의 앞 L 개의 위치들에 위치한 변환 계수들의 코딩에 하나의 단일 컨텍스트가 적용될 수 있고, 다른 컨텍스트는 다른 위치들의 계수들의 코딩에 적용될 수 있다. 하나의 구체적인 예에서, L 의 값은 1과 동일하게 설정될 수 있다. 따라서, 하나의 컨텍스트는 하나의 변환 계수의 제1 변환 계수(즉, DC 계수)의 부호를 코딩하도록 할당되고, 다른 컨텍스트는 모든 다른 변환 계수들(즉, AC 계수)의 부호를 코딩하도록 할당

된다. 대안적으로, 하나의 변환 블록 내의 모든 부호 예측 위치들은 그들 각각의 중요도(또는 정확한 예측의 확률)에 따라 서로 다른 그룹들로 분류될 수 있고, 상이한 컨텍스트들이 각각의 그룹에 개별적으로 할당될 수 있다.

- [0236] 제3 방법에서, 부호 예측 코딩을 위해, 코딩 모드의 컨텍스트, 블록 크기 및 컴포넌트 채널 정보와 같은 코딩 관련 정보가 사용될 수 있다. 예를 들어, 상이한 컨텍스트가 인터 및 인트라 모드에 사용될 수 있다. 다른 예에서, 상이한 컨텍스트 모델이 상이한 블록에 사용될 수 있다. 또 다른 예에서, 상이한 컨텍스트가 루마 및 크로마 컴포넌트에 사용될 수 있다.
- [0237] 제4 방법에서, 변환 계수들을 결정하기 위한 변환 타입들은 부호 예측을 위한 컨텍스트로서 사용될 수 있다. 위에서 설명된 바와 같이, MTS 및 LFNST와 같이 하나의 변환 블록에서의 계수들의 에너지를 압축하기 위해 적용되는 복수의 변환 코어들이 존재한다. 서로 다른 변환 코어에서 생성된 변환 계수가 서로 다른 통계 특성을 나타낼 수 있음을 가정할 때, 서로 다른 변환 계수들의 부호를 엔트로피-인코딩하기 위해 서로 다른 컨텍스트가 사용될 수 있다.
- [0238] 상술한 컨텍스트 모델링 방식들이 개별적으로 설명되지만, 이들 중 임의의 것이 함께 사용될 수 있음이 고려된다. 실제로 각각의 방법은 상이한 컨텍스트 디자인을 달성하기 위해 다른 것들과 자유롭게 결합될 수 있다.
- [0239] 도 21은 본 개시내용의 일부 구현예들에 따른 블록 기반 비디오 코딩에서의 부호 예측을 위한 예시적인 방법(2100)의 흐름도이다. 방법(2100)은 비디오 인코더(20) 또는 비디오 디코더(30)와 연관된 비디오 프로세서에 의해 구현될 수 있으며, 이하에서 설명한 바와 같은 단계(2102) 내지 단계(2108)를 포함할 수 있다. 단계들 중 일부는 본 명세서에 제공된 개시내용을 수행하기 위해 선택적일 수 있다. 나아가, 일부 단계들은 동시에 수행될 수도 있고, 도 21에 도시된 것과 다른 순서로 수행될 수도 있다.
- [0240] 단계(2102)에서, 비디오 프로세서는 비디오로부터의 비디오 프레임의 변환 블록에 주 변환 및 2차 변환을 적용하여 변환 블록에 대한 변환 계수를 생성할 수 있다.
- [0241] 단계(2104)에서, 비디오 프로세서는 변환 계수들로부터 부호 예측을 위한 후보 변환 계수들의 세트를 선택할 수 있다.
- [0242] 단계(2106)에서, 비디오 프로세서는 템플릿 기반 가설 생성 방식을 적용하여, 후보 변환 계수들의 세트에 대한 복수의 후보 가설들 중에서 가설을 선택할 수 있다.
- [0243] 단계(2108)에서, 비디오 프로세서는 선택된 가설과 연관된 부호 후보들의 조합을 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트로 결정할 수 있다.
- [0244] 본 개시내용과 일치하게, 방법(2100) 및 도 21은 비디오 인코더 측 또는 비디오 디코더 측에서 수행될 수 있다. 방법(2100)이 비디오 인코더 측에서 수행되는 경우, 비디오 인코더 측에서의 변환 계수 부호 예측을 위한 인코딩 방법으로서 고려될 수 있다. 이어서, 부호 예측의 정확성을 가리키는 부호 시그널링 비트가 생성되고, 디코더로 전송되는 비트스트림에 포함되도록 인코딩된다. 비디오 디코더 측에서 방법(2100)이 수행되는 경우, 비디오 디코더 측에서의 변환 계수 부호 예측을 위한 디코딩 방법으로서 고려될 수 있다. 부호 시그널링 비트들은 변환 계수들의 양자화된 레벨들의 인코딩된 데이터를 더 포함하는 비트스트림에서 디코더로 시그널링된다. 디코더는 부호 시그널링 비트들을 사용하여 그의 부호 예측을 정정하여 수신된 변환 계수들의 원래 부호를 추정할 수 있다. 예시적인 인코딩 방법 및 비디오 디코더 측에서의 변환 계수 부호 예측을 위한 예시적인 디코딩 방법이 각각 도 22 및 도 23을 참조하여 이하에 제공된다.
- [0245] 도 22은 본 개시내용의 일부 구현예들에 따른 비디오 인코더에 의해 수행되는 변환 계수 부호 예측을 위한 예시적인 비디오 인코딩 방법(2200)의 흐름도이다. 방법(2200)은 비디오 인코더(20)와 연관된 비디오 프로세서에 의해 구현될 수 있으며, 이하에서 설명한 바와 같은 단계(2202) 내지 단계(2122)를 포함할 수 있다. 단계들 중 일부는 본 명세서에 제공된 개시내용을 수행하기 위해 선택적일 수 있다. 나아가, 일부 단계들은 동시에 수행될 수도 있고, 도 22에 도시된 것과 다른 순서로 수행될 수도 있다.
- [0246] 단계(2202)에서, 비디오 프로세서는 비디오로부터의 비디오 프레임의 변환 블록에 주 변환을 적용하여, 변환 블록에 대한 변환 계수를 생성할 수 있다.
- [0247] 단계(2104)에서, 비디오 프로세서는 변환 블록의 서브 블록들에 2차 변환을 적용하여 변환 블록에 대한 변환 계수들을 생성할 수 있다. 일부 구현예들에서, 2차 변환은 순방향 LFNST일 수 있다. 일부 구현예들에서, 비디오 프

로세서는 변환 계수 부호 예측을 수행하기 위한 부호 예측 영역을 결정할 수 있다.

- [0248] 단계(2206)에서, 비디오 프로세서는 변환 계수들로부터 부호 예측을 위한 후보 변환 계수들의 세트를 선택할 수 있다. 일부 구현예들에서, 후보 변환 계수들은 변환 블록의 그들의 크기들(magnitudes)에 기초하여 재정렬된 변환 계수들로부터 선택될 수 있다. 예를 들어, 변환 계수들은 가장 높은 크기(magnitudes)로부터 가장 낮은 크기(magnitudes)로 재정렬될 수 있으며, 크기가 가장 높은 계수를 선택하여 부호 예측을 수행한다.
- [0249] 단계(2208)에서, 비디오 프로세서는 후보 변환 계수들의 세트에서의 후보 변환 계수들의 총 개수에 기초하여, 후보 변환 계수들의 세트에 대한 복수의 부호 후보들의 조합을 결정할 수 있다.
- [0250] 단계(2210)에서, 비디오 프로세서는 템플릿 기반의 가설 생성 방식을 적용하여, 복수의 부호 후보들의 조합에 대해 복수의 후보 가설들을 각각 생성할 수 있다.
- [0251] 단계(2212)에서, 비디오 프로세서는 비용 함수에 기초하여 복수의 후보 가설들로부터 최소 비용과 연관된 가설을 선택할 수 있다. 일부 구현예들에서, 비용 함수는 도미넌트 그래디언트 방향에 기초하여 이웃 샘플들을 추정함으로써 계산될 수 있다. 일부 구현예들에서, 상술된 바와 같이, 도미넌트 그래디언트 방향은 비용 함수 계산을 목적으로 추정 방향으로 채택되기 전에 특정 조건들을 만족해야 한다. 비디오 프로세서는 이러한 조건들이 충족되는지 여부를 결정할 수 있다. 이 경우, 이웃 샘플 추정은 비용 함수를 계산함에 있어서 도미넌트 그래디언트 방향을 따라 수행된다. 그렇지 않으면, 디폴트 추정 방향들(예를 들어, 상기 이웃에 대한 수직 추정 및 좌측 이웃에 대한 수평 추정)이 사용될 수 있다.
- [0252] 단계(2214)에서, 비디오 프로세서는 선택된 가설과 관련된 부호 후보들의 조합을 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트로 결정할 수 있다.
- [0253] 단계(2216)에서, 비디오 프로세서는 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하기 위한 하나 또는 그 이상의 컨텍스트를 결정할 수 있다. 일부 구현예들에서, 변환 블록에 대하여 하나의 컨텍스트 모델을 결정할 수 있다. 일부 다른 구현예들에서, 변환 블록의 상이한 변환 계수들에 대하여 상이한 컨텍스트 모델들을 결정할 수 있다. 상이한 컨텍스트는 변환 계수의 상이한 특징에 기초하여 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩할 수 있다. 제1 예에서, 컨텍스트들은 부호 시그널링 비트에 대응하는 후보 변환 계수들의 세트의 크기들(magnitudes)에 기초하여 결정된다. 제2 예에서, 컨텍스트들은 후보 변환 계수들의 스캔 위치들에 기초하여 결정된다. 제3 예에서, 컨텍스트들은 코딩 모드, 블록 크기 또는 각 후보 변환 계수의 컴포넌트 채널 정보에 기초하여 결정된다. 제4 예에서, 컨텍스트들은 후보 변환 계수들을 생성하기 위해 적용된 변환 코어들에 기초하여 결정된다.
- [0254] 단계(2218)에서, 비디오 프로세서는 하나 또는 그 이상의 컨텍스트에 따라 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스를 생성할 수 있다. 상술한 바와 같이, 다양한 컨텍스트 모델들 및 이들의 조합들이 예측된 부호들의 정확성을 코딩하기 위해 사용될 수 있다. 일부 구현예들에서, 벡터-기반 부호 예측 방식을 적용함으로써 부호 시그널링 비트들의 시퀀스를 생성할 수 있다.
- [0255] 단계(2220)에서, 비디오 프로세서는 후보 변환 계수들을 양자화하여 후보 변환 계수들의 양자화된 레벨들을 획득할 수 있다.
- [0256] 단계(2222)에서, 비디오 프로세서는 부호 시그널링 비트들의 시퀀스 및 후보 변환 계수들의 양자화된 레벨들의 인코딩된 데이터를 포함하는 비트스트림을 생성할 수 있다.
- [0257] 도 23은 본 개시내용의 일부 구현예들에 따른 비디오 디코더에 의해 수행되는 변환 계수 부호 예측을 위한 예시적인 비디오 디코딩 방법(2300)의 흐름도이다. 방법(2300)은 비디오 디코더(30)와 연관된 비디오 프로세서에 의해 구현될 수 있으며, 이하에서 설명한 바와 같은 단계(2302) 내지 단계(2318)를 포함할 수 있다. 단계들 중 일부는 본 명세서에 제공된 개시내용을 수행하기 위해 선택적일 수 있다. 나아가, 일부 단계들은 동시에 수행될 수도 있고, 도 23에 도시된 것과 다른 순서로 수행될 수도 있다.
- [0258] 단계(2302)에서, 비디오 프로세서는 후보 변환 계수들의 세트를 역양자화할 수 있다. 일부 구현예들에서, 비디오 프로세서는 부호 시그널링 비트들의 시퀀스 및 변환 블록과 연관된 양자화된 변환 계수들을 포함하는 비트스트림을 수신할 수 있다. 비디오 프로세서는 도 3의 역 양자화부(86)를 통해 양자화된 변환 계수들로부터 역양자화된 변환 계수들을 생성할 수 있다.
- [0259] 후보 변환 계수들의 세트는 변환 계수 부호 예측을 위해 사용된다. 역양자화된 변환 계수들은 변환 블록과 연관된다. 비디오 디코더(30)에서의 변환 블록의 역양자화된 변환 계수들은 비디오 인코더(20)에서의 변환 블록의

변환 계수들과 동일할 수 있다. 일부 구현예들에서, 후보 변환 계수들의 세트가 변환 블록의 인코더 측에서 크기에 따라 재정렬되는 변환 계수들로부터 선택되는 경우, 디코더의 비디오 프로세서는 비트스트림으로부터 파싱된 후보 변환 계수들의 세트를 재정렬할 수 있다.

- [0260] 단계(2304)에서, 비디오 프로세서는 후보 변환 계수들의 세트 중의 후보 변환 계수들의 총 개수에 기초하여, 후보 변환 계수들의 세트에 대한 복수의 부호 후보들의 조합을 결정할 수 있다.
- [0261] 단계(2306)에서, 비디오 프로세서는 템플릿 기반 가설 생성 방식을 적용하여, 복수의 부호 후보들의 조합에 대해 복수의 후보 가설들을 각각 생성할 수 있다.
- [0262] 단계(2308)에서, 비디오 프로세서는 비용 함수에 기초하여 복수의 후보 가설들로부터 최소 비용과 연관된 가설을 선택할 수 있다. 일부 구현예들에서, 비용 함수는 상술한 바와 같이, 도미넌트 그래디언트 방향에 기초하여 이웃 샘플들을 추정함으로써 계산될 수 있다.
- [0263] 단계(2310)에서, 비디오 프로세서는 선택된 가설과 연관된 부호 후보들의 조합을 후보 변환 계수들의 세트에 대한 예측된 부호들의 세트로 결정할 수 있다.
- [0264] 단계(2312)에서, 비디오 프로세서는 인코더로부터 수신된 부호 시그널링 비트들의 시퀀스를 엔트로피-인코딩하는데 사용되는 하나 또는 그 이상의 컨텍스트를 결정할 수 있다. 부호 시그널링 비트들의 시퀀스는 비트스트림에서 디코더로 전송된다. 전술한 바와 같이, 인코더는 단일 컨텍스트 모델 또는 복수의 컨텍스트 모델들을 사용하여 부호 시그널링 비트를 엔트로피-인코딩할 수 있다. 디코더는 이러한 컨텍스트(들)을 결정하여 부호 시그널링 비트들을 디코딩하여 후보 변환 계수들에 대한 부호 예측의 정확성의 지시를 획득할 수 있다.
- [0265] 제1 예에서, 컨텍스트들은 부호 시그널링 비트에 대응하는 후보 변환 계수들의 세트의 크기들(magnitudes)에 기초하여 결정된다. 예를 들어, 변환 계수들의 크기들은 복수의 크기 세그먼트들 내에 속하며, 여기서 컨텍스트들은 각각의 크기 세그먼트들에 속하는 변환 계수들에 대한 부호 시그널링 비트들을 엔트로피-인코딩하기 위하여 할당된다. 비디오 프로세서는 각각의 후보 변환 계수들의 크기가 속하는 크기 세그먼트를 결정하고, 부호 시그널링 비트를 엔트로피-인코딩하는데 사용되는 컨텍스트로서 상기 크기 세그먼트에 할당된 컨텍스트를 결정할 수 있다. 복수의 크기 세그먼트들 중에서, 연속하는 2개의 크기 세그먼트들 마다 임계값에 의해 분리되고, 상기 임계값은 디코더에서 미리 프로그래밍된 고정 값 또는 인코더에 의해 결정되고 디코더에 시그널링될 수 있는 값이다. 일부 구현예들에서, 후보 변환 계수들의 크기들은 역양자화 없이 비트스트림으로부터 직접 파싱된 후보 변환 계수들의 양자화된 레벨들이다. 일부 구현예들에서, 후보 변환 계수들의 크기들은 후보 변환 계수들의 양자화 인덱스들이다.
- [0266] 제2 예에서, 컨텍스트들은 후보 변환 계수들의 스캔 위치에 기초하여 결정된다. 예를 들어, 변환 계수들의 스캔 위치들은 복수의 그룹들로 분류되고, 컨텍스트들은 각각의 그룹들에 속하는 변환 계수들에 대한 부호 시그널링 비트들을 엔트로피-인코딩하기 위하여 할당된다. 비디오 프로세서는 복수의 그룹 중 부호 시그널링 비트에 대응하는 후보 변환 계수의 스캔 위치가 속하는 그룹을 결정하고, 상기 그룹에 할당된 컨텍스트를 부호 시그널링 비트를 엔트로피-인코딩하는데 사용되는 컨텍스트로 결정할 수 있다.
- [0267] 제3 예에서, 컨텍스트들은 코딩 모드, 블록 크기 또는 각각의 후보 변환 계수들의 컴포넌트 채널 정보에 기초하여 결정된다. 제4 예에서, 컨텍스트들은 후보 변환 계수들을 생성하기 위해 적용된 변환 코어들에 기초하여 결정된다.
- [0268] 단계(2314)에서, 비디오 프로세서는 하나 또는 그 이상의 컨텍스트에 기초하여 부호 시그널링 비트들의 시퀀스를 디코딩할 수 있다.
- [0269] 단계(2316)에서, 비디오 프로세서는 예측된 부호들의 세트 및 비디오 인코더로부터 수신된 부호 시그널링 비트들의 시퀀스에 기초하여 후보 변환 계수들의 세트에 대한 원래 부호들을 추정할 수 있다.
- [0270] 예를 들어, 도 15b를 참조하면, 예측된 부호들의 세트는 값 (1, 0)을 갖는 그룹 #0, 값 (1, 0)을 갖는 그룹 #2 및 값 (1, 0)을 갖는 그룹 #3을 포함할 수 있으며, 여기서 1은 음수 부호를 가리키고 0은 양수 부호를 가리킨다. 부호 시그널링 비트들의 시퀀스는 그룹 #0에 대한 비트 "0", 그룹 #2에 대한 비트 "1, 1, 0" 및 그룹 #3에 대한 비트 "0"을 포함할 수 있다. 그룹 #0에 대한 비트는 "0"의 값을 가지므로, (1, 0)의 값을 갖는 이 그룹에 대한 예측된 부호가 정확함(예를 들어, 예측된 부호들이 원래 부호들과 동일함)을 가리키기 때문에 그룹 #0에 대한 추정된 원래 부호는 (1, 0)으로 결정된다. 그룹 #1에 대한 비트의 제1 비트는 "1"의 값을 가지므로, 이 그룹 (1, 0)에 대한 예측된 부호가 부정확함(예를 들어, 예측된 부호들이 원래 부호들과 동일하지 않음)을

가리키기 때문에 그룹 #1에 대한 추정된 원래 부호는 이 그룹 (1, 0)에 대한 예측된 부호 및 그룹 #1에 대한 제 2 비트 및 제3 비트 "1, 0"의 XOR 결과인 것으로 결정된다(예를 들어, 추정된 원래 부호 = $XOR((1, 0), (1, 0)) = (0, 0)$). 그룹 #2에 대한 비트는 "0"의 값을 가지므로, (1, 0)의 값을 갖는 이 그룹에 대한 예측된 부호가 정확함(예를 들어, 예측된 부호들이 원래 부호들과 동일함)을 가리키기 때문에 그룹 #2에 대한 추정된 원래 부호는 (1, 0)으로 결정된다. 그런 다음, (1, 0, 0, 0, 0, 0, 1, 0)을 포함하는 그룹 #0, #1 및 #2의 추정된 원래 부호들을 각각 연결함으로써, 후보 변환 계수들의 세트에 대한 추정된 원본 부호들을 형성한다. 상술한 바와 같이, 비디오 인코더는 예측된 부호들의 정확성을 코딩하기 위해 상이한 컨텍스트 모델들을 사용할 수 있으며, 예측된 부호들의 정확성은 이러한 컨텍스트 모델들에 따라 조정된다.

[0271] 단계(2318)에서, 비디오 프로세서는 후보 변환 계수들의 세트에 대한 추정된 원래 부호들에 기초하여 역양자화된 변환 계수들을 업데이트할 수 있다. 예를 들어, 비디오 프로세서는 추정된 원래 부호들을 후보 변환 계수들의 세트에 대응하는 변환 블록에서 역양자화된 변환 계수들에 대한 참 부호들로서 사용할 수 있다.

[0272] 일부 구현예들에서, 역양자화된 변환 계수들이 업데이트된 후, 비디오 프로세서는 역양자화된 변환 계수들에 역방향 주 변환 및 역방향 2차 변환을 더 적용하여, 변환 블록에 대응하는 잔여 블록에 잔여 샘플들을 생성할 수 있다. 역방향 2차 변환은 LFNST를 포함하는 2차 변환에 대응한다. 역방향 주 변환은 DCT-II, DCT-V, DCT-VIII, DST-I, DST-IV, DST-VII, 또는 아이덴티티 변환(identity transform)들을 포함하는 주 변환에 대응한다.

[0273] 일부 구현예들에서, 후보 변환 계수들의 세트에 대한 부호 시그널링 비트들의 시퀀스가 비디오 인코더에 의해 생성되는 것은, 비디오 인코더에 의해, 벡터 기반 부호 예측 방식을 비디오 인코더 측에서 선택된 다른 후보 변환 계수들의 세트에 대한 다른 예측된 부호들의 세트에 적용하는 단계; 부호 시그널링 비트들의 시퀀스를 포함하는 비트스트림을 생성하는 단계;를 포함한다. 다른 후보 변환 계수들의 세트는 비디오 디코더 측의 후보 변환 계수들의 세트에 대응하는 비디오 인코더 측의 변환 계수들이다.

[0274] 일부 구현예들에서, 벡터 기반 부호 예측 방식을 다른 후보 변환 계수들의 세트에 대한 다른 예측 부호들 세트에 적용하는 것은, 다른 후보 변환 계수들의 세트를 복수의 후보 변환 계수들의 그룹들로 분할하는 단계; 및 각 후보 변환 계수들의 그룹에 대해, 후보 변환 계수들의 그룹의 원래 부호들이 후보 변환 계수들의 그룹의 예측된 부호들과 동일한지 여부에 기초하여, 후보 변환 계수들의 그룹에 대한 하나 또는 그 이상의 부호 시그널링 비트들을 생성하는 단계;를 더 포함한다.

[0275] 일부 구현예들에서, 후보 변환 계수들의 그룹에 대한 하나 또는 그 이상의 부호 시그널링 비트들을 생성하는 것은, 후보 변환 계수들의 그룹의 원래 부호들이 후보 변환 계수들의 그룹의 예측된 부호들과 동일한 것에 응답하여, 0의 값을 갖는 빈(bin)을 생성하는 단계; 및 상기 빈을 부호 시그널링 비트로서 비트스트림에 추가하는 단계;를 포함한다. 일부 구현예들에서, 후보 변환 계수들의 그룹에 대한 하나 또는 그 이상의 부호 시그널링 비트들을 생성하는 것은, 후보 변환 계수들의 그룹의 원래 부호들이 후보 변환 계수들의 그룹의 예측된 부호들과 동일하지 않은 것에 응답하여, 1의 값을 갖는 빈을 생성하는 단계; 후보 변환 계수들의 그룹의 예측된 부호들의 대응하는 정확성을 알리기 위한 추가 빈들의 세트를 생성하는 단계; 및 비트스트림 빈 및 추가 빈들의 세트를 부호 시그널링 비트로서 비트스트림에 추가하는 단계;를 포함한다.

[0276] 도 24는 본 개시내용의 일부 구현예들에 따른 사용자 인터페이스(2450)와 결합된 컴퓨팅 환경(2410)을 나타낸다. 컴퓨팅 환경(2410)은 데이터 프로세싱 서버의 일부일 수 있다. 상술한 비디오 인코더(20) 또는 비디오 디코더(30) 내의 비디오 프로세서는 컴퓨팅 환경(2410)을 사용하여 구현될 수 있다. 컴퓨팅 환경(2410)은 프로세서(2420), 메모리(2430) 및 입출력(I/O) 인터페이스(2440)를 포함한다.

[0277] 프로세서(2420)는 일반적으로 디스플레이, 데이터 획득, 데이터 통신 및 이미지 프로세싱과 관련된 동작들과 같은 컴퓨팅 환경(2410)의 전반적인 동작들을 제어한다. 프로세서(2420)는 상술한 방법들의 전부 또는 일부의 단계들을 수행하기 위한 명령들을 실행하기 위한 하나 또는 그 이상의 프로세서들을 포함할 수 있다. 더욱이, 프로세서(2420)는 프로세서(2420)와 다른 컴포넌트들 사이의 상호 작용을 용이하게 하는 하나 또는 그 이상의 모듈들을 포함할 수 있다. 프로세서는 CPU(Central Processing Unit), 마이크로-프로세서, 단일 칩 머신, GPU(Graphical Processing Unit) 등일 수 있다.

[0278] 메모리(2430)는 컴퓨팅 환경(2410)의 동작을 지원하기 위하여 다양한 유형의 데이터를 저장하도록 구성된다. 메모리(2430)는 미리 확정된 소프트웨어(2432)를 포함할 수 있다. 이러한 데이터의 예로는 컴퓨팅 환경(2410)에서 작동하는 모든 애플리케이션 또는 방법의 명령어, 비디오 데이터 세트, 이미지 데이터들을 포함한다. 메모리(2430)는 SRAM(Static Random Access Memory), EEPROM(Electrically Erasable Programmable Read-Only

Memory), EPROM(Erasable Programmable Read-Only Memory), PROM(Programmable Read-Only Memory), PROM(Programmable Read-Only Memory) ROM(Read-Only Memory), 자기 메모리, 플래시 메모리, 자기 디스크 또는 광 디스크 등 임의의 유형의 휘발성 또는 비휘발성 메모리 장치 또는 그 조합을 사용하는 것을 통하여 구현할 수 있다.

[0279] I/O 인터페이스(2440)는 프로세서(2420)와 키보드, 클릭 휠, 버튼 등과 같은 주변 인터페이스 모듈 사이의 인터페이스를 제공한다. 버튼은 홈 버튼, 스캔 시작 버튼 및 스캔 중지 버튼을 포함할 수 있으나, 이에 한정되는 것은 아니다. I/O 인터페이스(2440)는 인코더 및 디코더와 결합될 수 있다.

[0280] 일부 구현예들에서, 컴퓨팅 환경(2410)에서의 프로세서(2420)에 의하여 실행 가능한, 예를 들어 메모리(2430)에 있는 상술한 방법들을 수행하기 위한 복수의 프로그램들을 포함하는 비-일시적 컴퓨터 판독 가능 저장 매체가 더 제공된다. 일 예에서, 복수의 프로그램들은 컴퓨팅 환경(2410)에서 프로세서(2420)에 의해 실행되어, 인코딩된 비디오 정보(예를 들어, 인코딩된 비디오 프레임들을 나타내는 비디오 블록들, 및/또는 연관된 하나 또는 그 이상의 선택스 엘리먼트 등)를 포함하는 비트스트림 또는 데이터 스트림을 수신(예를 들어, 도 2의 비디오 인코더(20)로부터)할 수 있고, 또한 컴퓨팅 환경(2410)에서 프로세서(2420)에 의해 실행되어, 수신된 비트스트림 또는 데이터 스트림에 따라 상술한 디코딩 방법을 수행할 수 있다. 다른 예에서, 복수의 프로그램들은 컴퓨팅 환경(2410)에서 프로세서(2420)에 의해 실행되어, 비디오 정보(예를 들어, 비디오 프레임을 나타내는 비디오 블록들, 및/또는 연관된 하나 이상의 선택스 엘리먼트 등)를 비트스트림 또는 데이터 스트림으로 인코딩하기 위해 전술한 인코딩 방법을 수행할 수 있으며, 또한 컴퓨팅 환경(2410)에서 프로세서(2420)에 의해 실행되어 비트스트림 또는 데이터 스트림을 전송할 수도 있다(예를 들어, 도 3의 비디오 디코더(30)로 전송함). 대안적으로, 비-일시적 컴퓨터 판독 가능 저장 매체는, 예를 들어, 비디오 데이터의 디코딩에 있어서 디코더(예를 들어, 비디오 디코더(예를 들어, 도 3의 비디오 디코더(30))에 의한 사용을 위해 상술된 인코딩 방법을 사용하여, 인코더(예를 들어, 비디오 인코더(20))에 의해 생성된 인코딩된 비디오 정보(예를 들어, 인코딩된 비디오 프레임들을 나타내는 비디오 블록들, 및/또는 연관된 하나 또는 그 이상의 선택스 엘리먼트들 등)를 포함하는 비트스트림 또는 데이터 스트림을 그 안에 저장할 수 있다. 비-일시적 컴퓨터 판독 가능 저장 매체는, 예를 들어, ROM, RAM(Random Access Memory), CD-ROM, 자기 테이프, 플로피 디스크, 광 데이터 저장 장치 등일 수 있다.

[0281] 일부 구현예들에서, 하나 또는 그 이상의 프로세서(예를 들어, 프로세서(2420))들 및 하나 또는 그 이상의 프로세서들에 의하여 실행 가능한 복수의 프로그램을 저장하는 비-일시적 컴퓨터 판독 가능 저장 매체 또는 메모리(2430)를 포함하는 컴퓨팅 장치가 더 제공되며, 상기 하나 또는 그 이상의 프로세서들은 복수의 프로그램들을 실행할 때 상술한 방법들을 수행하도록 구성된다.

[0282] 일부 구현예들에서, 컴퓨팅 환경(2410)에서의 프로세서(2420)에 의하여 실행 가능한, 예를 들어 메모리(2430)에 있는 상술한 방법들을 수행하기 위한 복수의 프로그램들을 포함하는 컴퓨터 프로그램 제품이 더 제공된다. 예를 들어, 컴퓨터 프로그램 제품은 비-일시적 컴퓨터 판독 가능 저장 매체를 포함할 수 있다.

[0283] 일부 구현예들에서, 컴퓨팅 환경(2410)은 상술한 방법들을 수행하기 위하여 하나 또는 그 이상의 ASIC, DSP, 디지털 신호 프로세싱 장치(Digital Signal Processing Device: DSPD), 프로그래머블 로직 장치(Programmable Logic Device: PLD), FPGA, GPU, 컨트롤러, 마이크로-컨트롤러, 마이크로-프로세서 또는 기타 전자 컴포넌트들로 구현될 수 있다.

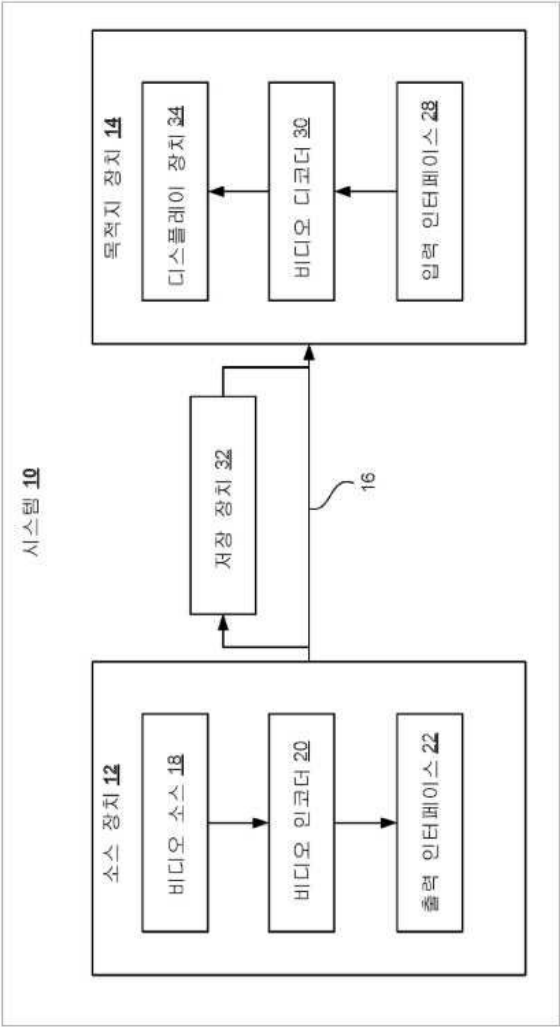
[0284] 본 개시내용에 대한 설명은 예시의 목적으로 제시된 것이며, 본 개시내용을 총망라하거나 이에 한정하기 위한 것이 아니다. 많은 보정, 변형, 그리고 대안적인 구현예들은 상술한 설명 및 관련 도면에 제시된 시사점으로부터 이익을 얻은 당업자들에게 자명할 것이다.

[0285] 특별히 반대되는 기재가 없는 한, 본 개시내용에 따른 방법의 단계들의 순서는 예시하기 위한 것일 뿐이며, 본 개시내용에 따른 방법의 단계들은 구체적으로 상술한 순서에 한정되는 것이 아니라, 현실적인 여건에 따라 변경될 수 있다. 또한, 본 개시내용에 따른 방법의 단계 중 적어도 하나는 실제 요구에 따라 조정, 조합 또는 삭제될 수 있다.

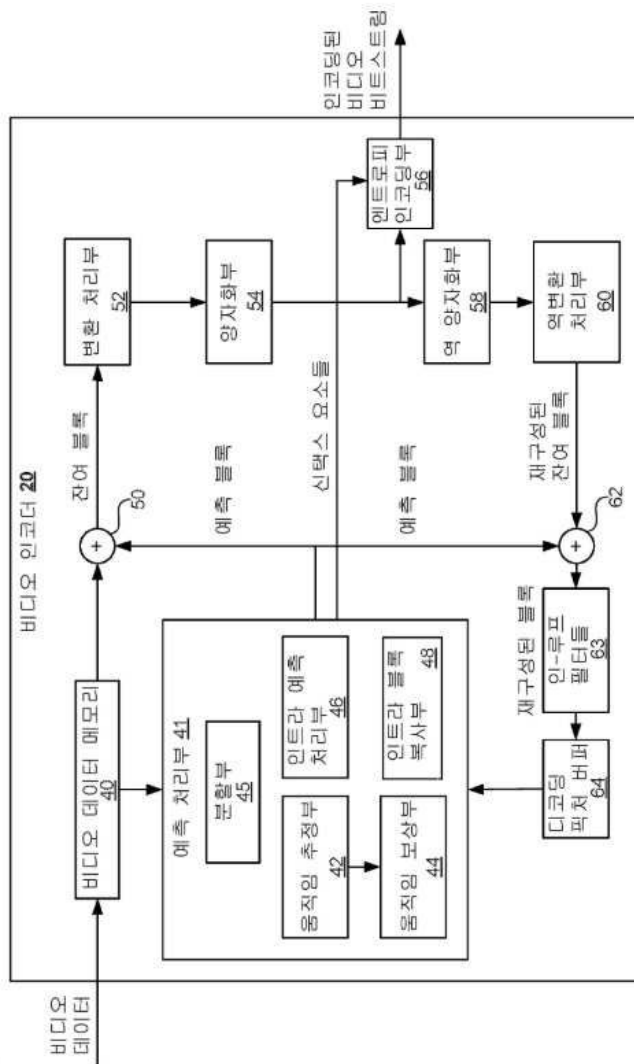
[0286] 실시예들은 개시의 원리들을 설명하고 당업자가 다양한 구현 방식들에 대한 개시를 이해할 수 있도록 하고, 고려되는 특정 용도에 적합한 다양한 변형들을 갖는 기본 원리들 및 다양한 구현 방식들을 가장 잘 활용하기 위해 선택되고 설명되었다. 따라서, 개시내용의 범위는 개시된 구현 방식의 구체적인 예에 한정되지 않으며, 보정 및 기타 구현 방식은 본 개시내용의 범위 내에 포함되도록 의도된 것으로 이해되어야 한다.

도면

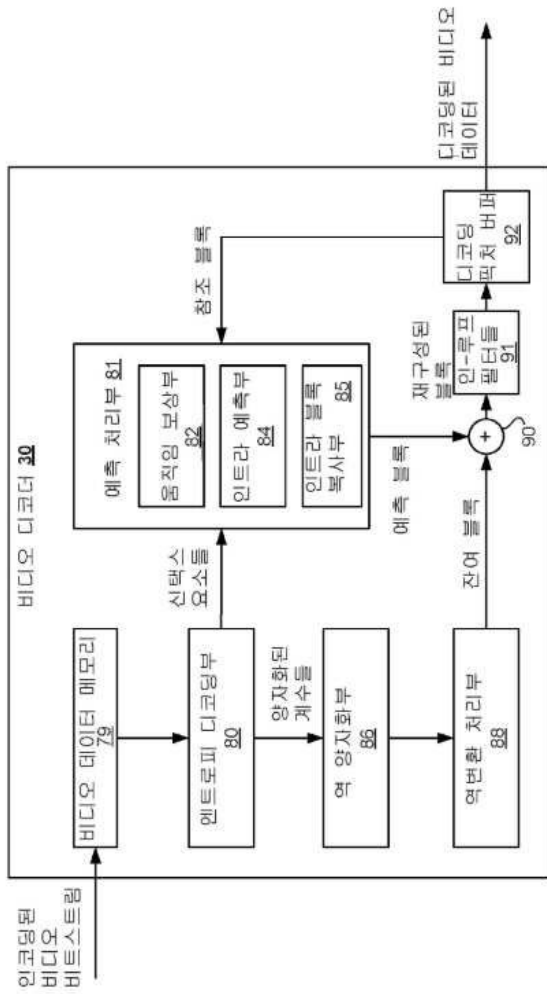
도면1



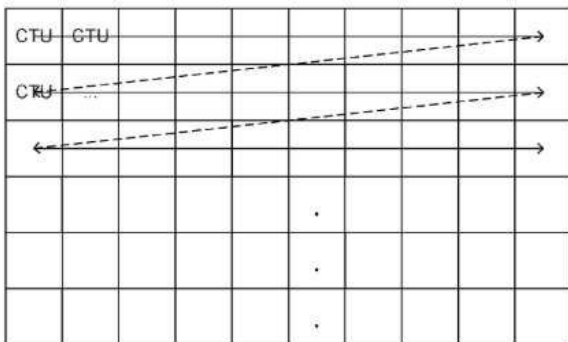
도면2



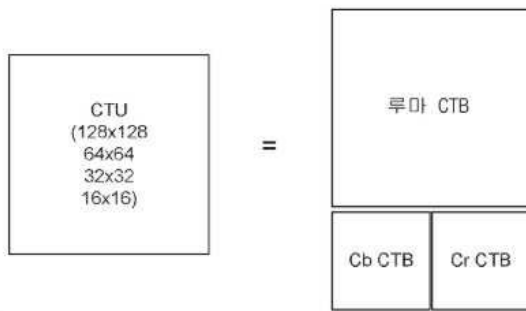
도면3



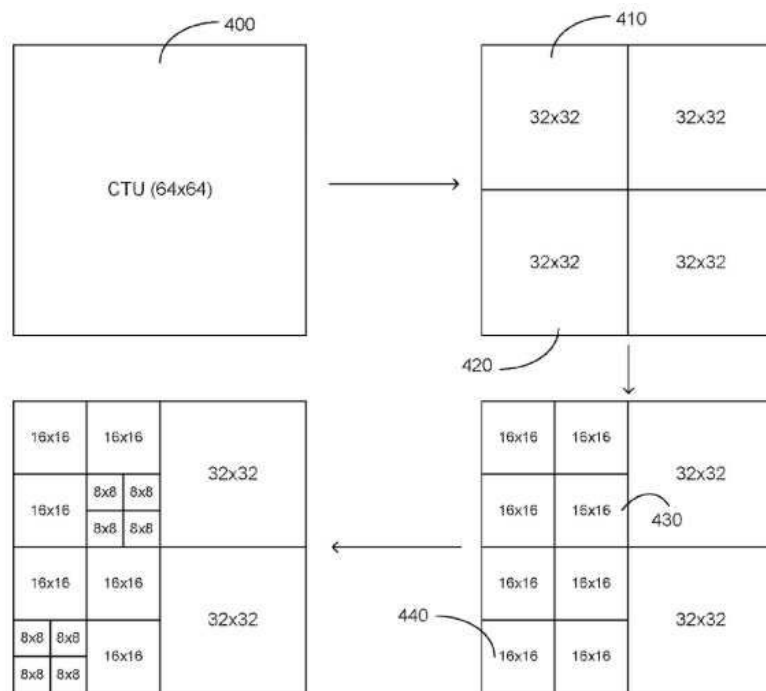
도면4a



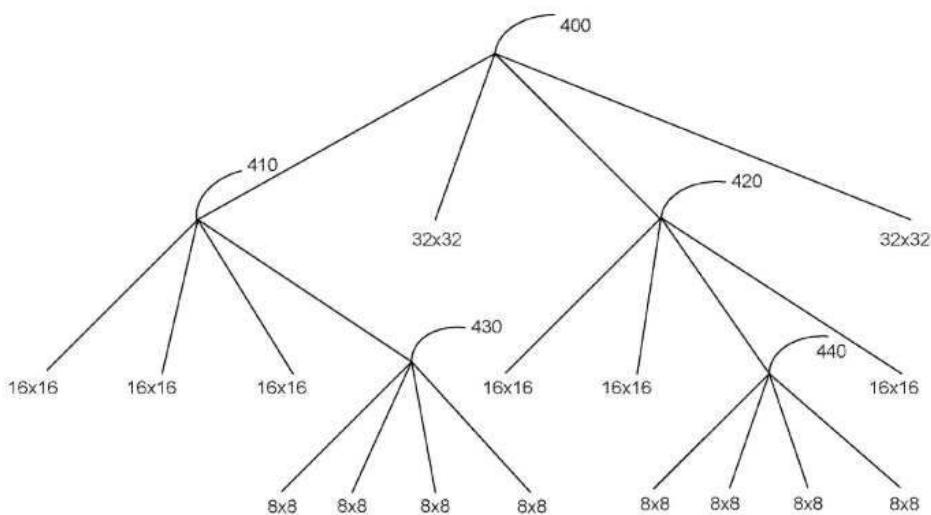
도면4b



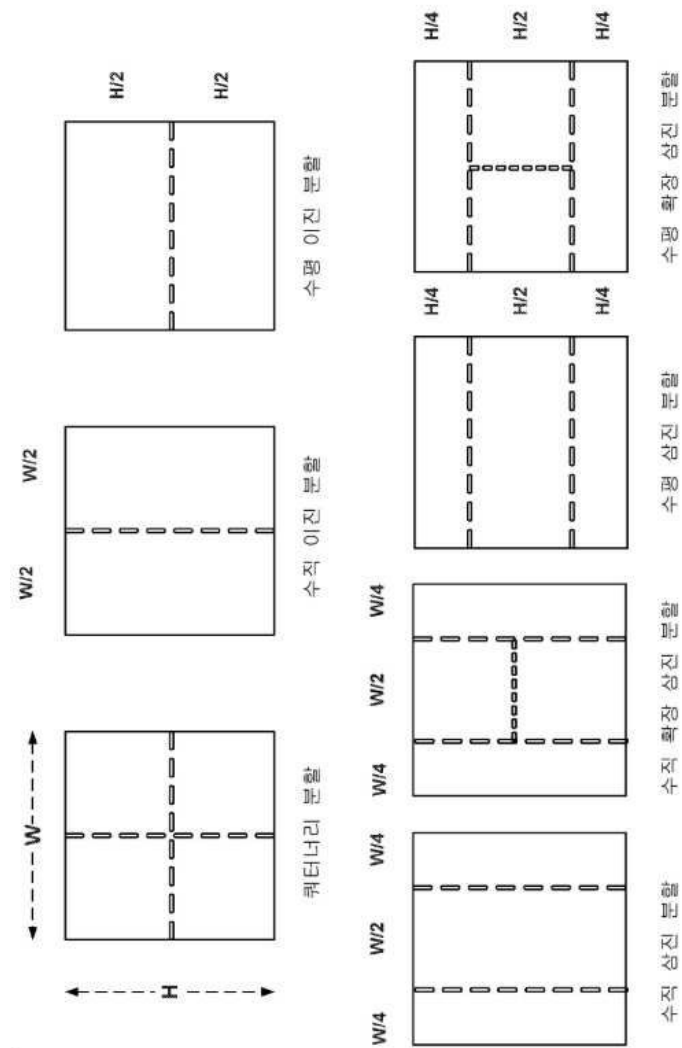
도면4c



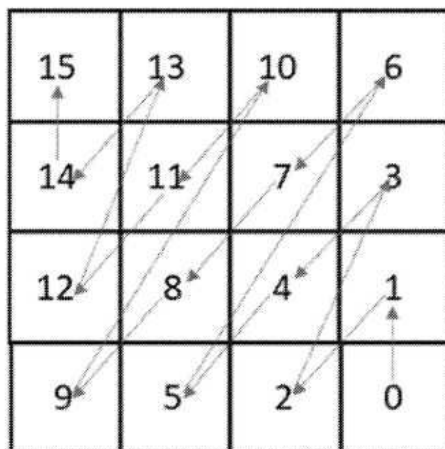
도면4d



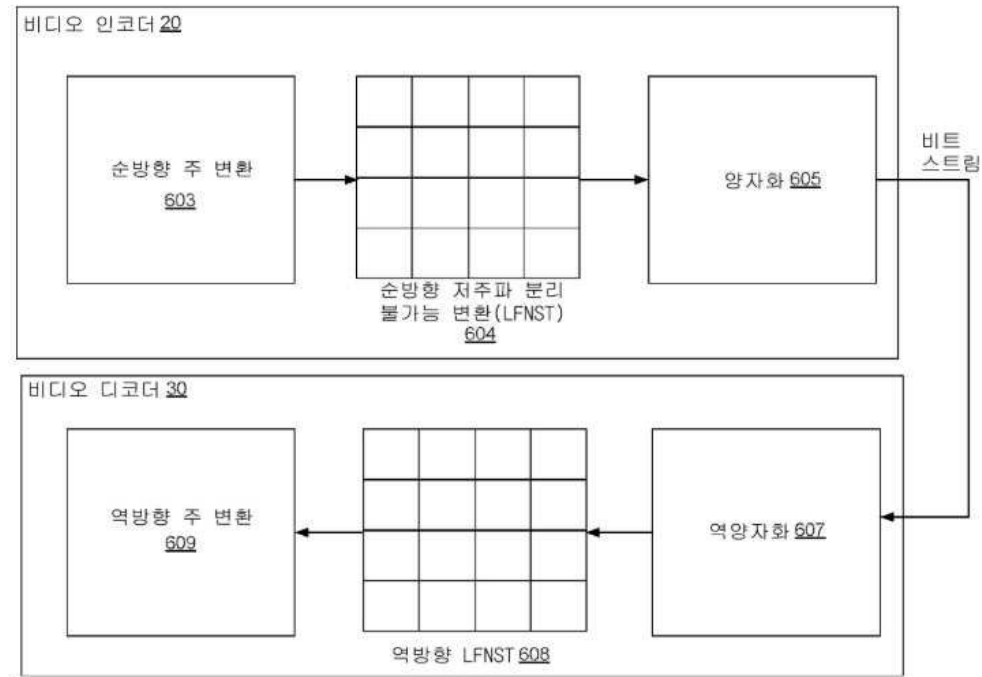
도면4e



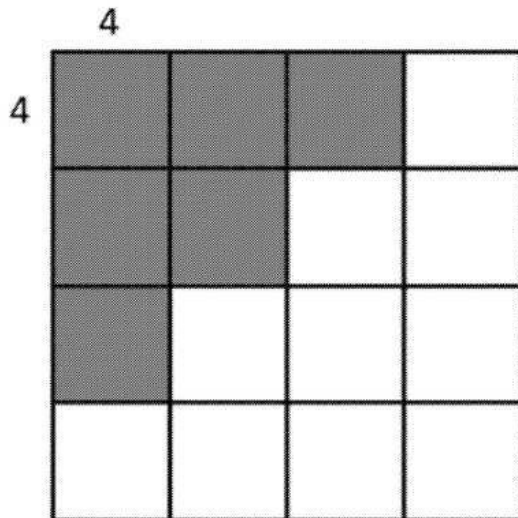
도면5



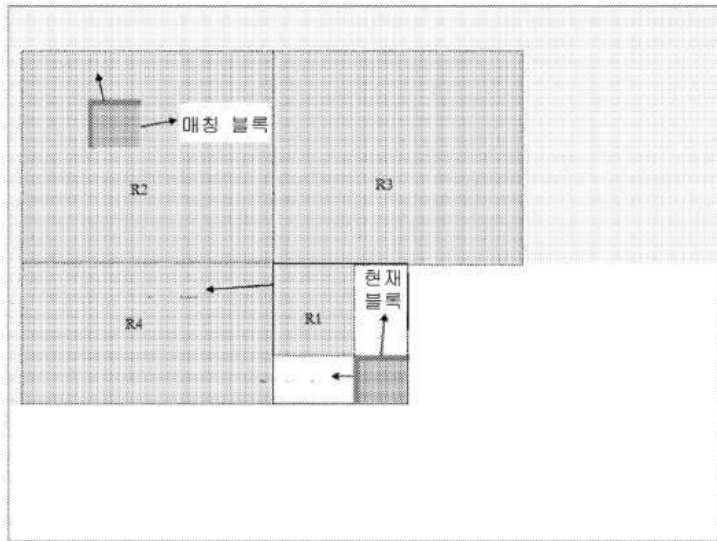
도면6



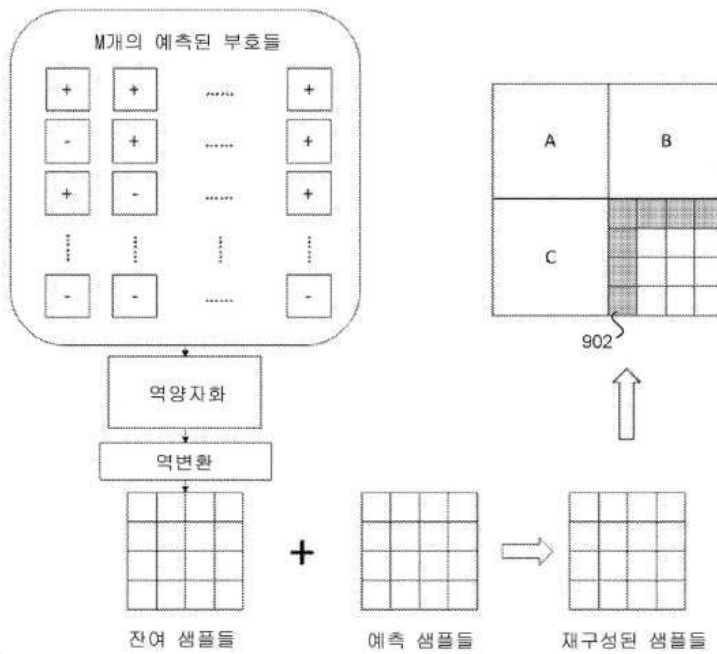
도면7



도면8



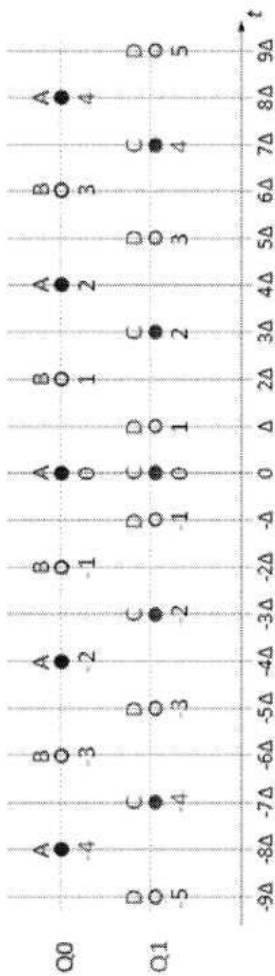
도면9



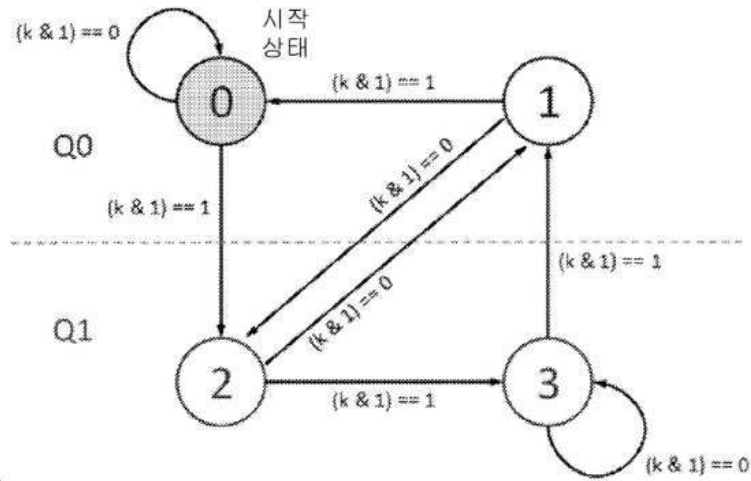
도면10

				$B_{2,0}$	$B_{2,1}$	$B_{2,2}$	$B_{2,3}$
				$B_{1,0}$	$B_{1,1}$	$B_{1,2}$	$B_{1,4}$
		$C_{0,2}$	$C_{0,1}$	$P_{0,0}$	$P_{0,1}$	$P_{0,2}$	$P_{0,3}$
		$C_{1,2}$	$C_{1,1}$	$P_{1,0}$			
		$C_{2,2}$	$C_{2,1}$	$P_{2,0}$			
		$C_{3,2}$	$C_{3,1}$	$P_{3,0}$			

도면11



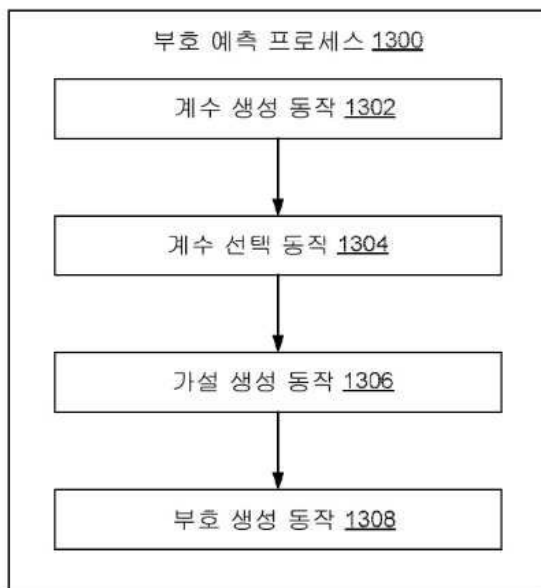
도면12a



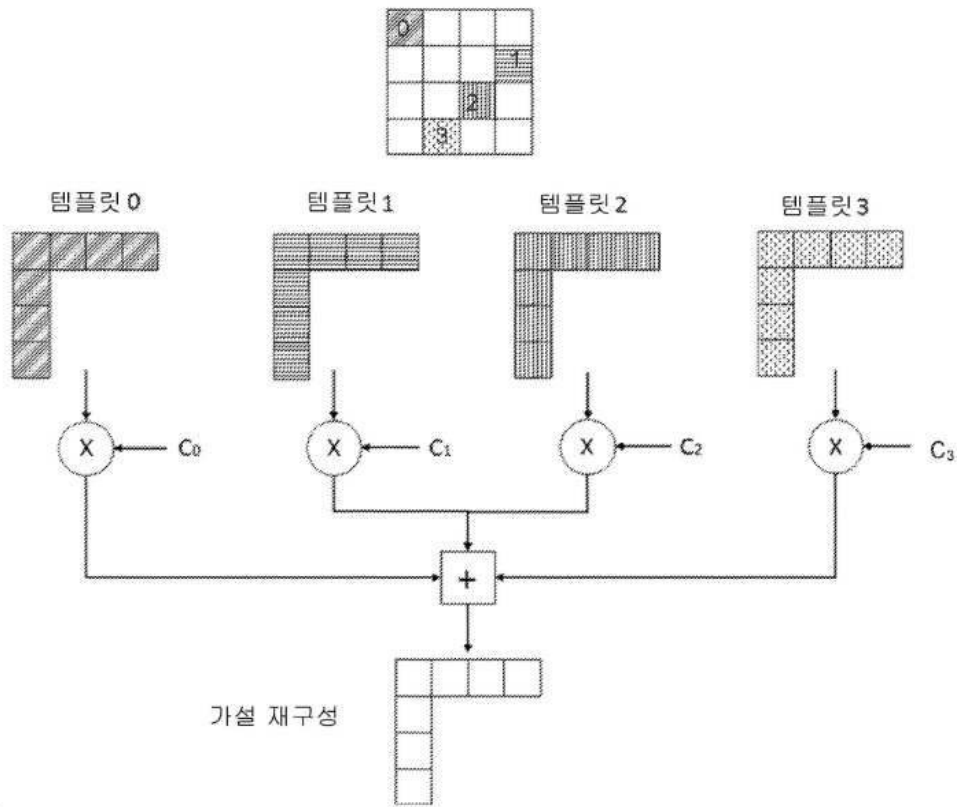
도면12b

현재 상태	다음 상태 ...	
	$(k \& 1) == 0$	$(k \& 1) == 1$
0	0	2
1	2	0
2	1	3
3	3	1

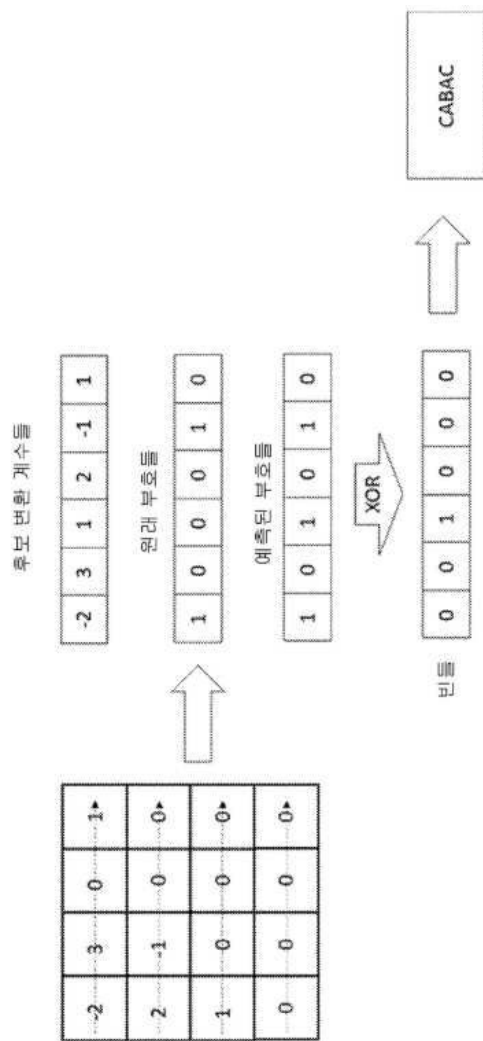
도면13



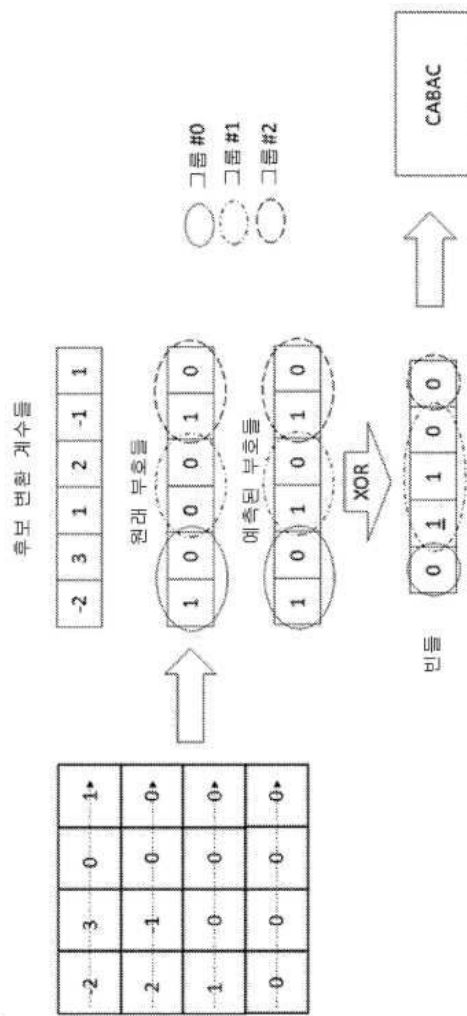
도면14



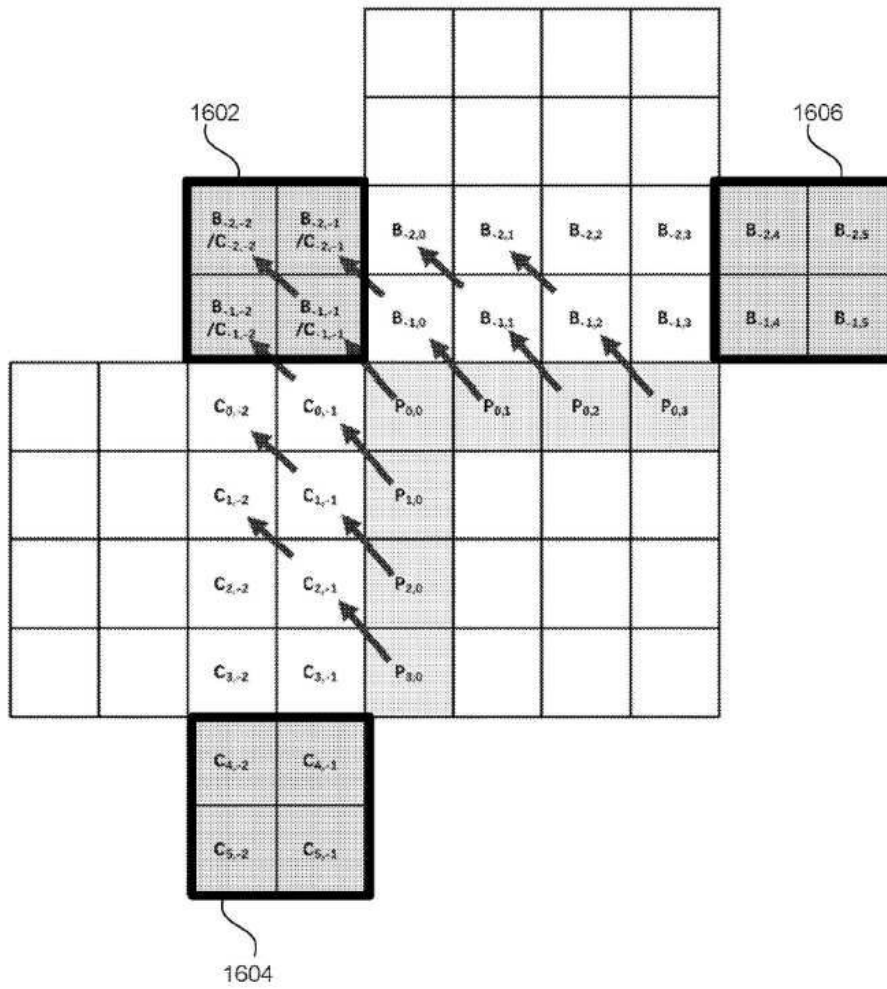
도면15a



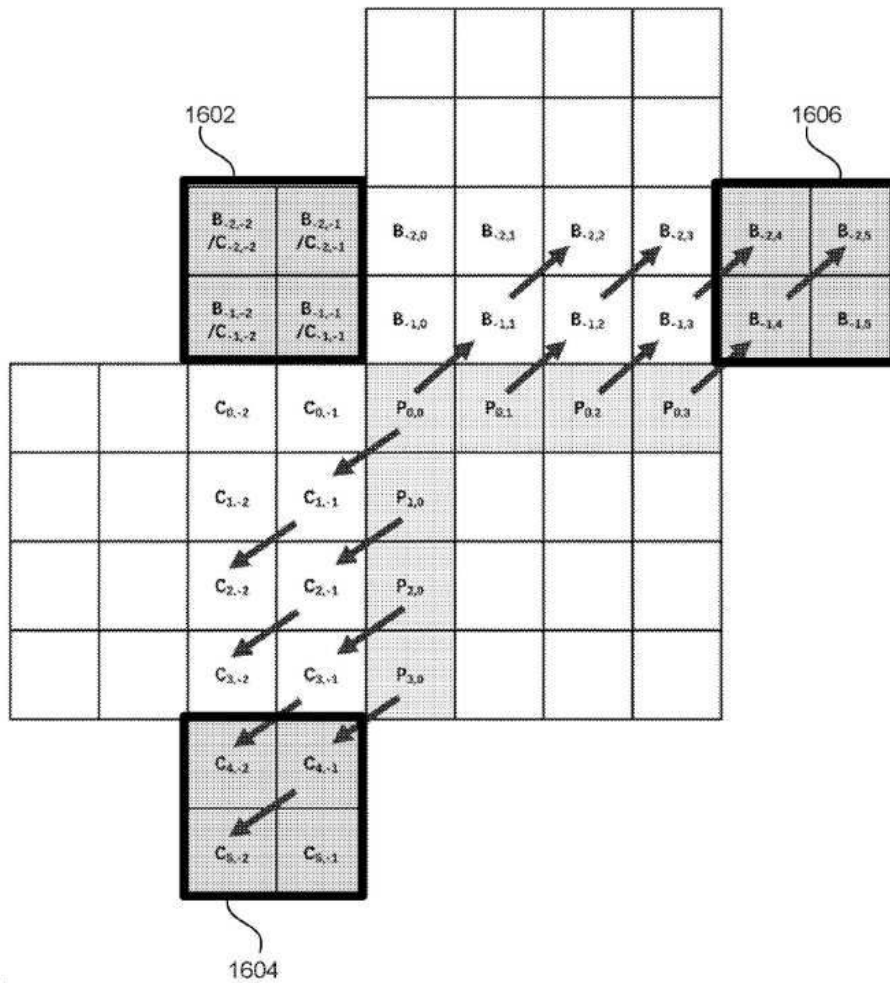
도면15b



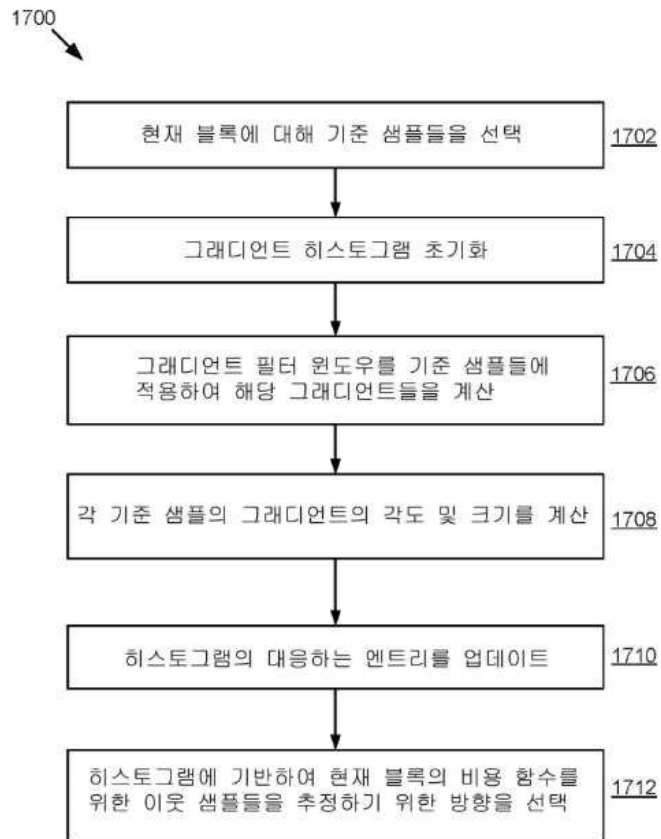
도면 16a



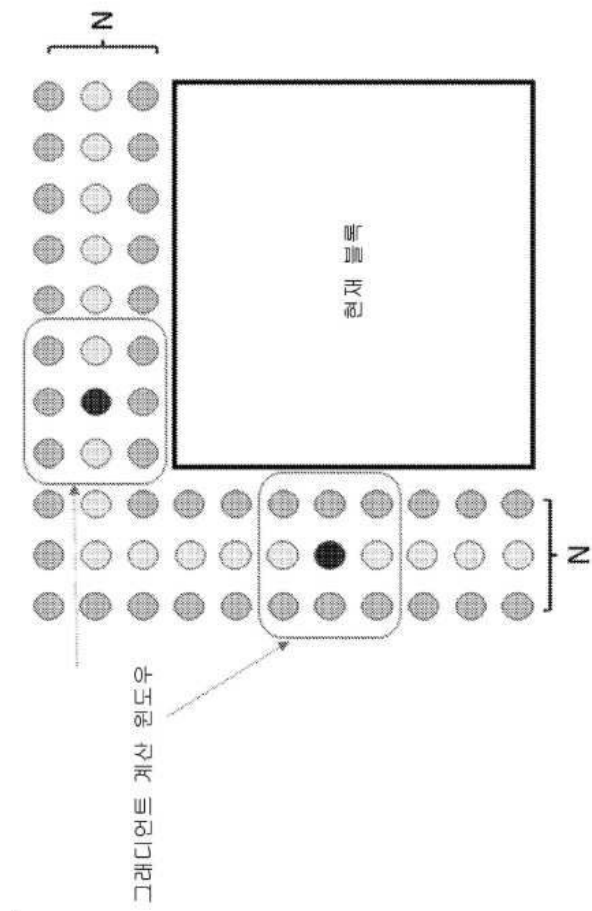
도면 16b



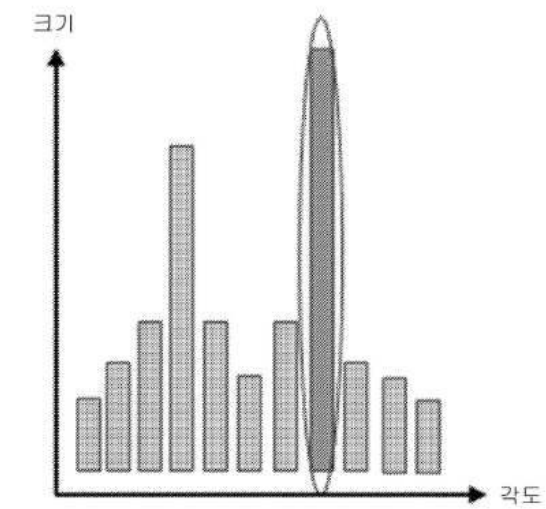
도면17



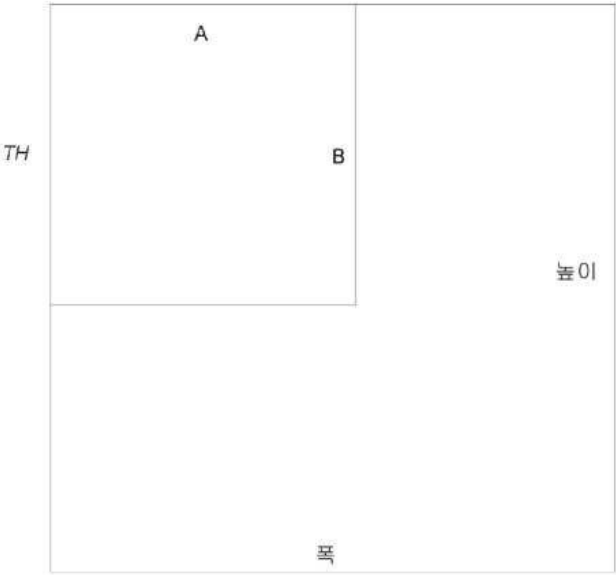
도면18a



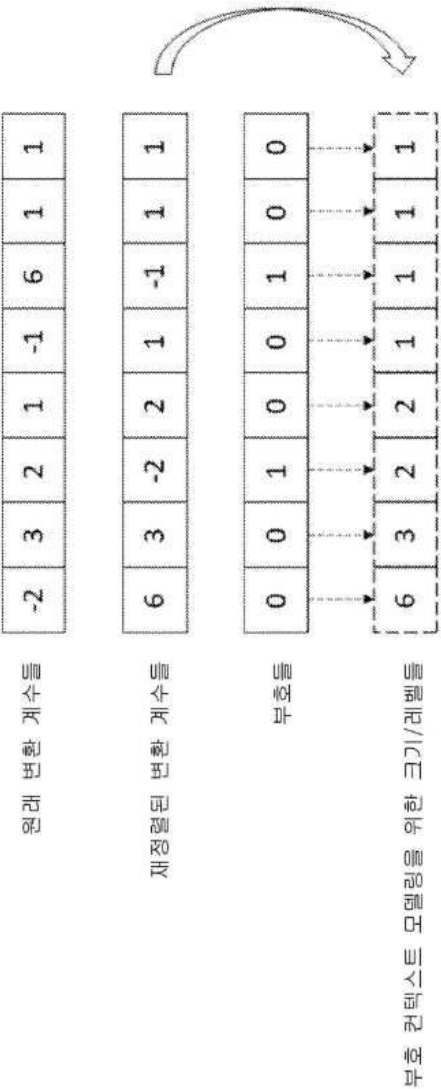
도면18b



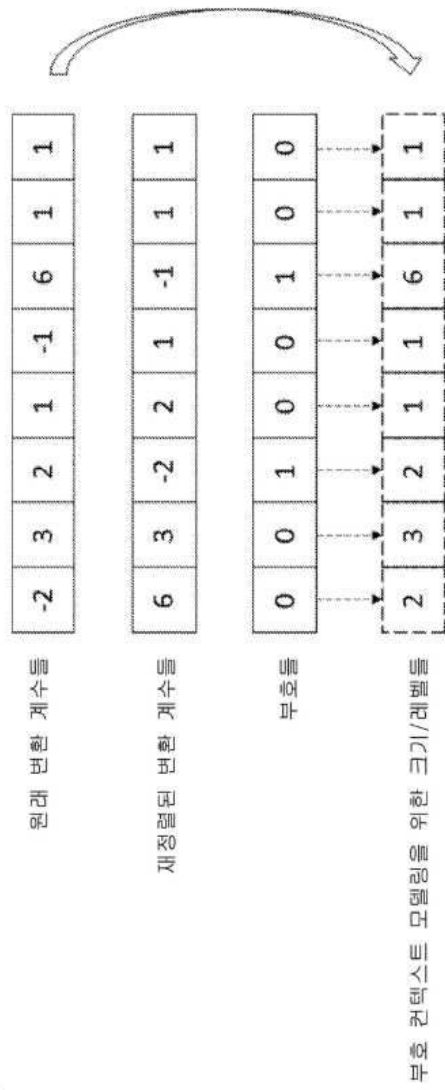
도면19



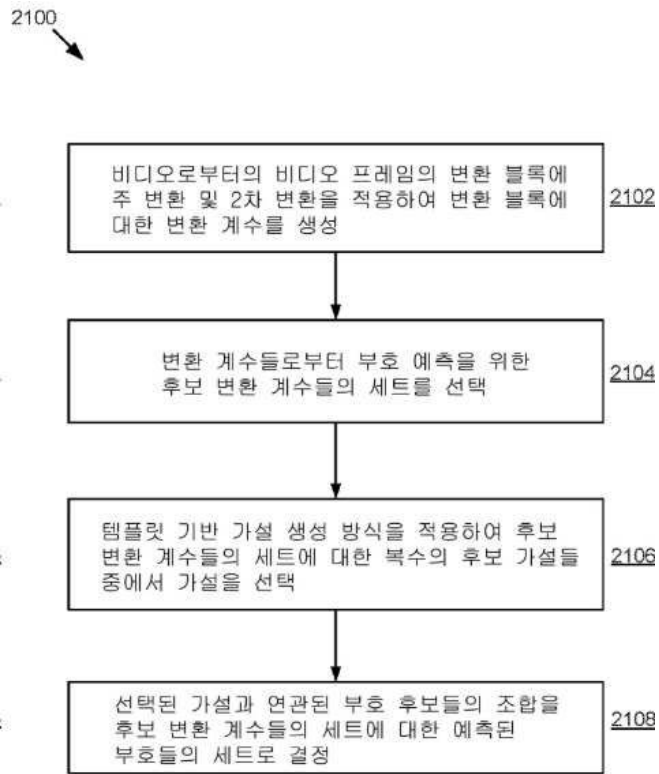
도면20a



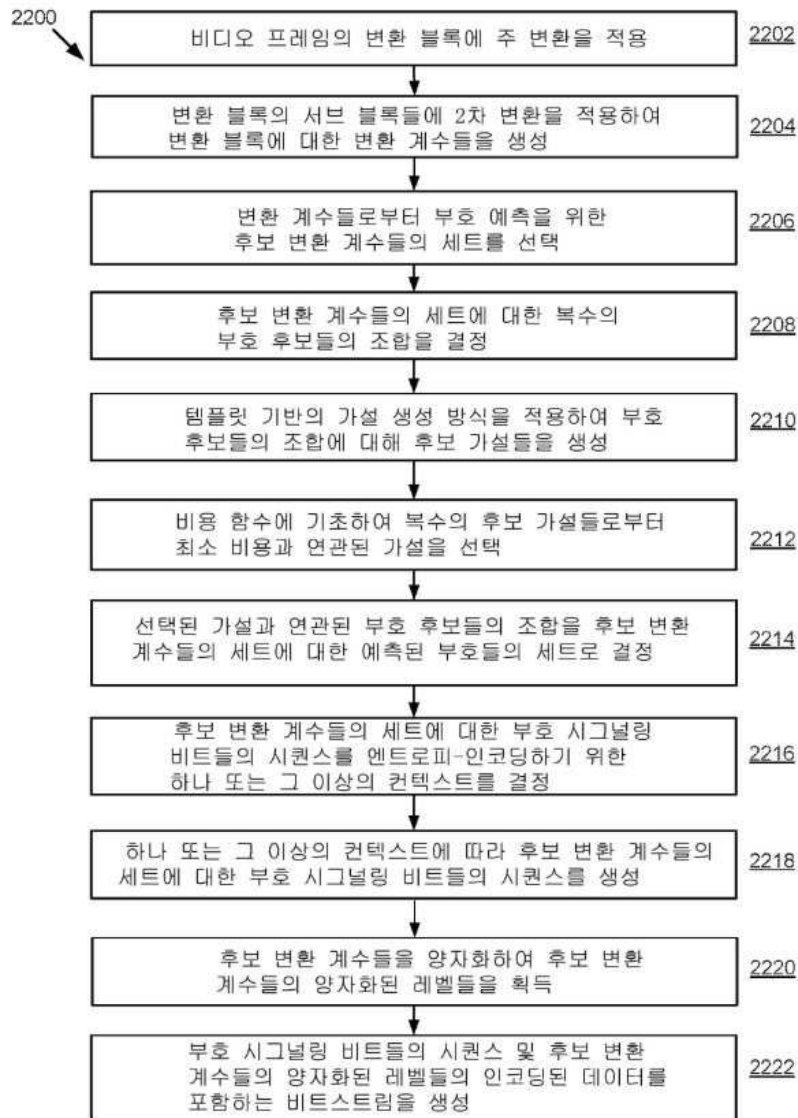
도면20b



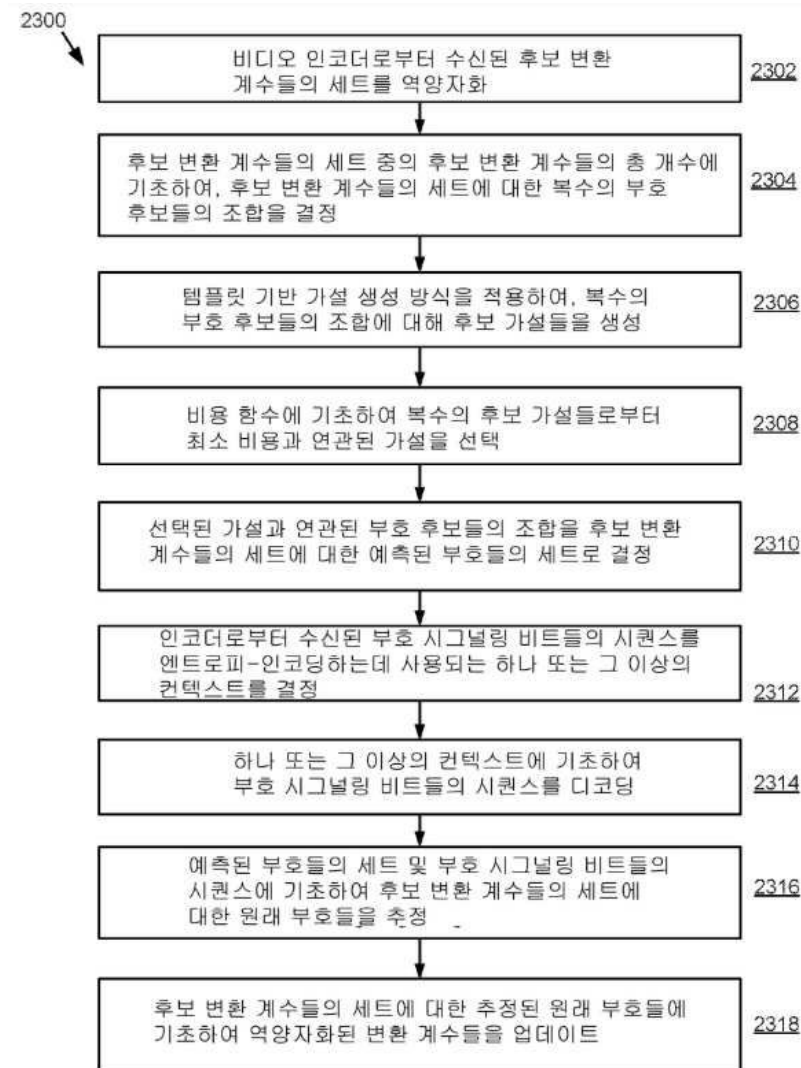
도면21



도면22



도면23



도면24

