



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0002789
(43) 공개일자 2025년01월07일

- (51) 국제특허분류(Int. Cl.)
H04N 19/61 (2014.01) H04N 19/132 (2014.01)
H04N 19/186 (2014.01) H04N 19/70 (2014.01)
H04N 19/82 (2014.01)
- (52) CPC특허분류
H04N 19/61 (2015.01)
H04N 19/132 (2015.01)
- (21) 출원번호 10-2024-7040565 (분할)
(22) 출원일자(국제) 2020년12월29일
심사청구일자 2024년12월06일
- (62) 원출원 특허 10-2022-7023785
원출원일자(국제) 2020년12월29일
심사청구일자 2023년06월12일
- (85) 번역문제출일자 2024년12월06일
(86) 국제출원번호 PCT/CN2020/141080
(87) 국제공개번호 WO 2021/139572
국제공개일자 2021년07월15일
- (30) 우선권주장
62/958,582 2020년01월08일 미국(US)
- (71) 출원인
광둥 오포 모바일 텔레커뮤니케이션즈 코퍼레이션
리미티드
중국, 광둥 523860, 동관, 창안, 우샤, 하이빈 로
드, 넘버 18
- (72) 발명자
휘, 권옌
중국 523860 광둥 등관 창안 우샤 하이빈 로드 넘
버 18
완, 쇠이
중국 523860 광둥 등관 창안 우샤 하이빈 로드 넘
버 18
마, 옌취
중국 523860 광둥 등관 창안 우샤 하이빈 로드 넘
버 18
- (74) 대리인
특허법인씨엔에스

전체 청구항 수 : 총 16 항

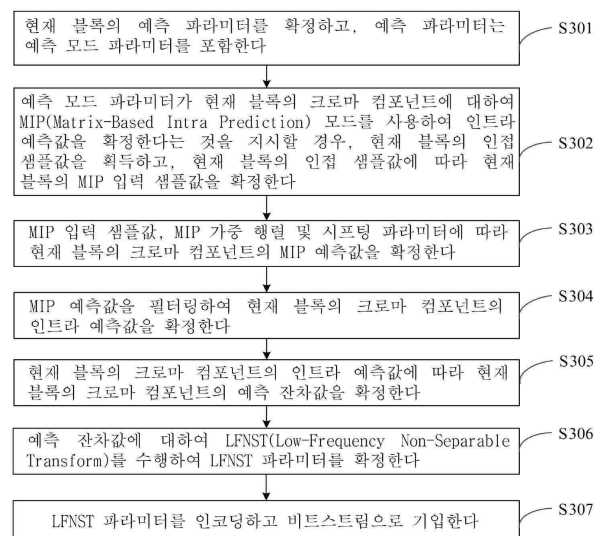
(54) 발명의 명칭 인코딩 방법, 디코딩 방법, 인코더, 디코더 및 저장 매체

(57) 요약

본 출원의 실시예는 인코딩 방법, 디코딩 방법, 인코더, 디코더 및 저장 매체를 개시한다. 상기 방법은, 현재 블
록의 예측 파라미터를 확정하는 것 - 예측 파라미터는 예측 모드 파라미터를 포함함 - 과, 예측 모드 파라미터가
현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을

(뒷면에 계속)

대표도 - 도3



확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하는 것과, MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하는 것과, MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하는 것과, 현재 블록의 크로마 컴포넌트의 인트라 예측값에 따라 현재 블록의 크로마 컴포넌트의 예측 잔차값을 확정하는 것과, 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정하는 것과, LFNST 파라미터를 인코딩하고 비트스트림으로 기입하는 것을 포함한다.

(52) CPC특허분류

H04N 19/186 (2015.01)

H04N 19/70 (2015.01)

H04N 19/82 (2015.01)

명세서

청구범위

청구항 1

인코더에 적용되는 인코딩 방법으로서,

현재 블록의 예측 파라미터를 확정하는 것 - 상기 예측 파라미터는 예측 모드 파라미터를 포함함 - 과,

상기 예측 모드 파라미터가 상기 현재 블록의 색상 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction)를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 상기 현재 블록의 인접 샘플값을 확정하고, 상기 현재 블록의 인접 샘플값에 따라 상기 현재 블록의 MIP 입력 샘플값을 확정하는 것과,

상기 MIP 입력 샘플값, MIP 가중 행렬, 시프팅 오프셋 파라미터 및 시프팅 넘버 파라미터에 따라 상기 현재 블록의 색상 컴포넌트의 MIP 예측값을 확정하는 것 - 상기 MIP 예측값은 상기 현재 블록의 색상 컴포넌트의 일부 샘플의 예측값이고, 상기 시프팅 오프셋 파라미터는 제1 고정값이고, 상기 시프팅 넘버 파라미터는 제2 고정값임 - 과,

상기 MIP 예측값을 필터링하여 상기 현재 블록의 색상 컴포넌트의 인트라 예측값을 확정하는 것과,

상기 현재 블록의 색상 컴포넌트의 인트라 예측값에 따라 상기 현재 블록의 색상 컴포넌트의 예측 잔차값을 확정하는 것과,

상기 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정하는 것과,

상기 LFNST 파라미터를 인코딩하고 비트스트림으로 기입하는 것을 포함하는 것을 특징으로 하는 인코딩 방법.

청구항 2

제 1 항에 있어서,

상기 MIP 입력 샘플값, MIP 가중 행렬, 시프팅 오프셋 파라미터 및 시프팅 넘버 파라미터에 따라 상기 현재 블록의 색상 컴포넌트의 MIP 예측값을 확정하는 것은,

상기 MIP 입력 샘플값의 합과 상기 시프팅 오프셋 파라미터의 값의 곱을 확정하는 것과,

제 1 오프셋의 값을 제 1 상수값과 상기 곱의 차이로 설정하는 것과,

상기 예측 파라미터에 따라 상기 현재 블록의 MIP 가중 행렬을 확정하는 것과,

상기 MIP 가중 행렬과 상기 MIP 입력 샘플값의 제 1 가중 합을 계산하는 것과,

상기 제 1 가중 합과 상기 제 1 오프셋의 제 1 합을 계산하는 것과,

상기 제 1 합에 대하여 이진 오른쪽 시프팅(binary right-shifting)을 수행하여 제 1 오른쪽 시프트 값(first right-shift value)을 획득하는 것 - 상기 오른쪽으로 시프트된 비트의 수는 상기 시프팅 넘버 파라미터의 값과 같음 - 과,

상기 현재 블록의 색상 컴포넌트의 MIP 예측값을 상기 제 1 오른쪽 시프트 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 합과 동일하게 설정하는 것 - 상기 제 1 임시 참조값은 상기 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 획득함 - 을 포함하는 것을 특징으로 하는 인코딩 방법.

청구항 3

제 1 항에 있어서,

상기 예측 파라미터는 상기 현재 블록의 사이즈 파라미터를 더 포함하고,

상기 현재 블록의 인접 샘플값에 따라 상기 현재 블록의 MIP 입력 샘플값을 확정하는 것은,

상기 현재 블록의 사이즈 파라미터에 따라 상기 현재 블록의 블록 사이즈 인덱스 값을 확정하는 것과,

상기 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 제 1 임시 참조값을 획득하는 것과,

상기 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우, 상기 MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 제 2 상수값과 상기 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하고, 상기 MIP 입력 샘플값 중 인덱스 번호 i 에 대응하는 값을 상기 제 1 임시 참조값 중 인덱스 번호 i 에 대응하는 값과 상기 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하는 것 - i 는 0보다 큰 정수임 - 과,

상기 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위를 벗어나는 경우, 상기 MIP 입력 샘플값 중 인덱스 번호 j 에 대응하는 값을 상기 제 1 임시 참조값 중 인덱스 번호 $(j+1)$ 에 대응하는 값과 상기 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하는 것 - j 는 0 이상의 정수임 - 을 포함하는 것을 특징으로 하는 인코딩 방법.

청구항 4

제 3 항에 있어서,

상기 제 2 상수값은 $1 \ll (\text{BitDepth}-1)$ 이고, BitDepth는 상기 현재 블록의 상기 색상 컴포넌트의 비트 깊이를 나타내는 것을 특징으로 하는 인코딩 방법.

청구항 5

제 2 항에 있어서,

상기 시프팅 넘버 파라미터의 값은 6이고,

상기 제 1 상수값은 32이며,

상기 시프팅 오프셋 파라미터의 값은 32인 것을 특징으로 하는 인코딩 방법.

청구항 6

제 1 항에 있어서,

상기 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정하는 것은,

상기 현재 블록의 상기 예측 잔차값에 대하여 제 1차 변환을 수행하여 제 1 변환 계수를 획득하는 것과,

상기 제 1 변환 계수 중 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정하는 것과,

적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정한 경우, 상기 제 1 변환 계수 중 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 계수를 확정하는 것과,

상기 LFNST 계수를 양자화하여 양자화 계수를 획득하는 것과,

상기 양자화 계수를 인코딩하고 인코딩된 양자화 계수를 비트스트림으로 기입하는 것을 포함하는 것을 특징으로 하는 인코딩 방법.

청구항 7

제 6 항에 있어서,

상기 LFNST 파라미터는 LFNST 인덱스 번호를 포함하고, 상기 LFNST 파라미터를 확정하는 것은,

상기 제 1 변환 계수 중 적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정한 경우, 상기 LFNST 인덱스 번호의 값을 0보다 크게 설정하는 것과,

상기 제 1 변환 계수 중 적어도 일부 변환 계수에 대하여 LFNST를 수행하지 않기로 확정한 경우, 상기 LFNST 인덱스 번호의 값을 0으로 설정하는 것과,

상기 LFNST 인덱스 번호를 인코딩하고 비트스트림으로 기입하는 것을 더 포함하는 것을 특징으로 하는 인코딩

방법.

청구항 8

디코더에 적용되는 디코딩 방법으로서,

비트스트림을 파싱하여 현재 블록의 예측 파라미터 및 LFNST(Low-Frequency Non-Separable Transform) 파라미터를 획득하는 것 - 상기 예측 파라미터는 예측 모드 파라미터를 포함함 - 과,

상기 예측 모드 파라미터가 상기 현재 블록의 색상 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction)를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 상기 현재 블록의 인접 샘플값을 확정하고, 상기 현재 블록의 인접 샘플값에 따라 상기 현재 블록의 MIP 입력 샘플값을 확정하는 것과,

상기 MIP 입력 샘플값, MIP 가중 행렬, 시프팅 오프셋 파라미터 및 시프팅 넘버 파라미터에 따라 상기 현재 블록의 색상 컴포넌트의 MIP 예측값을 확정하는 것 - 상기 MIP 예측값은 상기 현재 블록의 색상 컴포넌트의 일부 샘플의 예측값이고, 상기 시프팅 오프셋 파라미터는 제1 고정값이고, 상기 시프팅 넘버 파라미터는 제2 고정값임 - 과,

상기 MIP 예측값을 필터링하여 상기 현재 블록의 색상 컴포넌트의 인트라 예측값을 확정하는 것과,

상기 LFNST 파라미터가 상기 현재 블록에 대하여 LFNST를 수행함을 지시할 경우, 상기 현재 블록의 재구성된 변환 계수를 확정하고, 상기 재구성된 변환 계수 중 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수를 획득하는 것과,

상기 제 2 변환 계수에 대하여 제 1 차 변환을 수행함으로써 상기 현재 블록의 색상 컴포넌트의 재구성된 잔차값을 획득하는 것과,

상기 현재 블록의 색상 컴포넌트의 인트라 예측값 및 상기 재구성된 잔차값에 따라 상기 현재 블록의 색상 컴포넌트의 재구성된 값을 확정하는 것을 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 9

제 8 항에 있어서,

상기 MIP 입력 샘플값, MIP 가중 행렬, 시프팅 오프셋 파라미터 및 시프팅 넘버 파라미터에 따라 상기 현재 블록의 색상 컴포넌트의 MIP 예측값을 확정하는 것은,

상기 MIP 입력 샘플값의 합과 상기 시프팅 오프셋 파라미터의 값의 곱을 확정하는 것과,

제 1 오프셋의 값을 제 1 상수값과 상기 곱의 차이로 설정하는 것과,

상기 예측 파라미터에 따라 상기 현재 블록의 MIP 가중 행렬을 확정하는 것과,

상기 MIP 가중 행렬과 상기 MIP 입력 샘플값의 제 1 가중 합을 계산하는 것과,

상기 제 1 가중 합과 상기 제 1 오프셋의 제 1 합을 계산하는 것과,

상기 제 1 합에 대하여 이진 오른쪽 시프팅(binary right-shifting)을 수행하여 제 1 오른쪽 시프트 값(first right-shift value)을 획득하는 것 - 상기 오른쪽으로 시프트된 비트의 수는 상기 시프팅 넘버 파라미터의 값과 같음 - 과,

상기 현재 블록의 색상 컴포넌트의 MIP 예측값을 상기 제 1 오른쪽 시프트 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 합과 동일하게 설정하는 것 - 상기 제 1 임시 참조값은 상기 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 획득함 - 을 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 10

제 8 항에 있어서,

상기 예측 파라미터는 상기 현재 블록의 사이즈 파라미터를 더 포함하고,

상기 현재 블록의 인접 샘플값에 따라 상기 현재 블록의 MIP 입력 샘플값을 확정하는 것은,

상기 현재 블록의 사이즈 파라미터에 따라 상기 현재 블록의 블록 사이즈 인덱스 값을 확정하는 것과,

상기 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 제 1 임시 참조값을 획득하는 것과,

상기 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우, 상기 MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 제 2 상수값과 상기 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하고, 상기 MIP 입력 샘플값 중 인덱스 번호 i 에 대응하는 값을 상기 제 1 임시 참조값 중 인덱스 번호 i 에 대응하는 값과 상기 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하는 것 - i 는 0보다 큰 정수임 - 과,

상기 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위를 벗어나는 경우, 상기 MIP 입력 샘플값 중 인덱스 번호 j 에 대응하는 값을 상기 제 1 임시 참조값 중 인덱스 번호 $(j+1)$ 에 대응하는 값과 상기 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하는 것 - j 는 0 이상의 정수임 - 을 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 11

제 10 항에 있어서,

상기 제 2 상수값은 $1 \ll (\text{BitDepth}-1)$ 이고, BitDepth는 상기 현재 블록의 상기 색상 컴포넌트의 비트 깊이를 나타내는 것을 특징으로 하는 디코딩 방법.

청구항 12

제 9 항에 있어서,

상기 시프팅 넘버 파라미터의 값은 6이고,

상기 제 1 상수값은 32이며,

상기 시프팅 오프셋 파라미터의 값은 32인 것을 특징으로 하는 디코딩 방법.

청구항 13

제 8 항에 있어서,

비트스트림을 파싱하여 LFNST 파라미터를 획득하는 것은,

상기 비트스트림을 파싱하여 LFNST 인덱스 번호를 획득하는 것과,

상기 LFNST 인덱스 번호의 값이 0보다 큰 경우, 상기 현재 블록에 대하여 LFNST를 수행하기로 확정하는 것과,

상기 LFNST 인덱스 번호의 값이 0인 경우, 상기 현재 블록에 대하여 LFNST를 수행하지 않기로 확정하는 것을 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 14

제 8 항에 있어서,

상기 현재 블록의 재구성된 변환 계수를 확정하는 것은,

상기 비트스트림을 파싱하여 상기 현재 블록의 양자화된 계수를 획득하는 것과,

상기 양자화된 계수에 대하여 스케일링하여 상기 현재 블록의 재구성된 변환 계수를 획득하는 것을 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 15

인코더로서,

적어도 하나의 프로세서와 메모리를 포함하고,

상기 메모리는 상기 적어도 하나의 프로세서에 연결되어 적어도 하나의 실행 가능한 컴퓨터 프로그램을 저장하도록 구성되고,

상기 적어도 하나의 프로세서는 상기 적어도 하나의 실행 가능한 컴퓨터 프로그램을 실행할 때에 제 1 항 내지

제 7 항 중 어느 한 항에 기재된 방법을 실행하도록 구성되는 것을 특징으로 하는 인코더.

청구항 16

디코더로서,

적어도 하나의 프로세서와 메모리를 포함하고,

상기 메모리는 상기 적어도 하나의 프로세서에 연결되어 적어도 하나의 실행 가능한 컴퓨터 프로그램을 저장하도록 구성되고,

상기 적어도 하나의 프로세서는 상기 적어도 하나의 실행 가능한 컴퓨터 프로그램을 실행할 때에 제 8 항 내지 제 14 항 중 어느 한 항에 기재된 방법을 실행하도록 구성되는 것을 특징으로 하는 디코더.

발명의 설명

기술 분야

[0001] (관련 출원)

[0002] 본 출원은 Junyan Huo, Shuai Wan 및 Yanzhuo Ma의 이름으로 2020년 1월 8일에 출원된 미국 선출원의 우선권을 주장하며, 출원 번호는 62/958,582이고, 명칭이 "VIDEO ENCODING AND DECODING METHOD, APPARATUS AND COMMUNICATION SYSTEM"이며, 그 전문이 본원에 참조로 포함된다.

[0003] 본 출원의 실시예는 비디오 코딩의 기술 분야에 관한 것으로, 특히 인코딩 방법, 디코딩 방법, 인코더, 디코더 및 저장 매체에 관한 것이다.

배경 기술

[0004] 사람들이 비디오 디스플레이 품질에 대한 요구가 높아짐에 따라, 고화질 및 초고화질 비디오와 같은 새로운 비디오 응용 형태가 등장하고 있다. H.265/고효율 비디오 코딩(high efficiency video coding, HEVC)은 이미 신속하게 발전하는 비디오 애플리케이션의 요구를 만족할 수 없다. 공동 비디오 탐색 팀(Joint video exploration team, JVET)은 차세대 비디오 코딩 표준 H.266/다용도 비디오 코딩(versatile video coding, VVC)을 제안한다.

[0005] H.266/VVC에서 MIP(Matrix-Based Intra Prediction)는 인트라 예측 모드이며, 현재 블록의 인트라 예측 블록을 획득하는 데에 사용된다. LFNST(Low-Frequency Non-Separable Transform) 기술을 사용하여 현재 블록의 예측 잔차를 확정한다. 그러나 LFNST 기술이 MIP 모드 예측에 적용될 경우에 VVC의 코딩 효율에 부정적인 영향을 미친다.

발명의 내용

[0006] 본 출원의 실시예는 코딩 효율을 향상시킬 수 있는 인코딩 방법, 디코딩 방법, 인코더, 디코더 및 저장 매체를 제공한다.

[0007] 본 출원의 실시예의 기술 방안은 다음과 같이 달성할 수 있다.

[0008] 제 1 양태에서, 본 출원의 실시예는 인코더에 적용되는 인코딩 방법을 제공한다. 상기 방법은,

[0009] 현재 블록의 예측 파라미터를 확정하는 것 - 예측 파라미터는 예측 모드 파라미터를 포함함 - 과,

[0010] 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하는 것과,

[0011] MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하는 것 - MIP 예측값은 현재 블록의 크로마 컴포넌트 중 일부 샘플의 예측값임 - 과,

[0012] MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하는 것과,

[0013] 현재 블록의 크로마 컴포넌트의 인트라 예측값에 따라 현재 블록의 크로마 컴포넌트의 예측 잔차값을 확정하는 것과,

- [0014] 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정하는 것과,
- [0015] LFNST 파라미터를 인코딩하고 비트스트림으로 기입하는 것을 포함한다.
- [0016] 제 2 양태에서, 본 출원의 실시예는 디코더에 적용되는 디코딩 방법을 제공한다. 상기 방법은,
- [0017] 비트스트림을 파싱하여 현재 블록의 예측 파라미터 및 LFNST(Low-Frequency Non-Separable Transform) 파라미터를 획득하는 것 - 예측 파라미터는 예측 모드 파라미터를 포함함 - 과,
- [0018] 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하는 것과,
- [0019] MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하는 것 - MIP 예측값은 현재 블록의 크로마 컴포넌트 중 일부 샘플의 예측값임 - 과,
- [0020] MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하는 것과,
- [0021] LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행함을 지시할 경우, 현재 블록의 재구성된 변환 계수 블록을 확정하고, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득하는 것과,
- [0022] 제 2 변환 계수 블록에 대하여 제 1차 변환을 수행함으로써 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 획득하는 것과,
- [0023] 현재 블록의 크로마 컴포넌트의 인트라 예측값 및 재구성된 잔차 블록에 따라 현재 블록의 크로마 컴포넌트의 재구성된 블록을 확정하는 것을 포함한다.
- [0024] 제 3 양태에서, 본 출원의 실시예는 인코더를 제공한다. 인코더는 제 1 확정 유닛, 제 1 예측 유닛, 제 1 변환 유닛 및 인코딩 유닛을 포함한다.
- [0025] 제 1 확정 유닛은 현재 블록의 예측 파라미터를 확정하도록 구성되고, 예측 파라미터는 예측 모드 파라미터를 포함한다.
- [0026] 제 1 확정 유닛은 또한 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하도록 구성된다.
- [0027] 제 1 예측 유닛은 MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하도록 구성되며, 여기서 MIP 예측값은 현재 블록의 크로마 컴포넌트 중 부분 샘플의 예측값이다. 제 1 예측 유닛은 또한 MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하도록 구성된다.
- [0028] 제 1 변환 유닛은 현재 블록의 크로마 컴포넌트의 인트라 예측값에 따라 현재 블록의 크로마 컴포넌트의 예측 잔차값을 확정하고, 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정하도록 구성된다.
- [0029] 인코딩 유닛은 LFNST 파라미터를 인코딩하고 비트스트림으로 기입하도록 구성된다.
- [0030] 제 4 양태에서, 본 출원의 실시예는 인코더를 제공한다. 인코더는 제 1 메모리 및 제 1 프로세서를 포함한다. 제 1 메모리는 제 1 프로세서에서 실행 가능한 컴퓨터 프로그램을 저장하도록 구성된다. 제 1 프로세서는 컴퓨터 프로그램을 실행할 때에 제 1 양태의 방법을 실행하도록 구성된다.
- [0031] 제 5 양태에서, 본 출원의 실시예는 디코더를 제공한다. 디코더는 파싱 유닛, 제 2 확정 유닛, 제 2 예측 유닛 및 제 2 변환 유닛을 포함한다.
- [0032] 파싱 유닛은 비트스트림을 파싱하여 현재 블록의 예측 파라미터 및 LFNST(Low-Frequency Non-Separable Transform) 파라미터를 획득하도록 구성되고, 예측 파라미터는 예측 모드 파라미터를 포함한다.
- [0033] 제 2 확정 유닛은 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득

하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하도록 구성된다.

- [0034] 제 2 예측 유닛은 MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하고, 및 MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하도록 구성되며, MIP 예측값은 현재 블록의 크로마 컴포넌트 중 일부 샘플의 예측값이다.
- [0035] 제 2 변환 유닛은 LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행함을 지시할 경우, 현재 블록의 재구성된 변환 계수 블록을 확정하고, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득하고, 제 2 변환 계수 블록에 대하여 제 1차 변환을 수행함으로써 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 획득하도록 구성된다.
- [0036] 제 2 확정 유닛은 또한 현재 블록의 크로마 컴포넌트의 인트라 예측값 및 재구성된 잔차 블록에 따라 현재 블록의 크로마 컴포넌트의 재구성된 블록을 확정하도록 구성된다.
- [0037] 제 6 양태에서, 본 출원의 실시예는 디코더를 제공한다. 디코더는 제 2 메모리 및 제 2 프로세서를 포함한다. 제 2 메모리는 제 2 프로세서에서 실행 가능한 컴퓨터 프로그램을 저장하도록 구성된다. 제 2 프로세서는 컴퓨터 프로그램을 실행할 때 제 2 양태의 방법을 실행하도록 구성된다.
- [0038] 제 7 양태에서, 본 출원의 실시예는 컴퓨터 저장 매체를 제공한다. 컴퓨터 저장 매체는 컴퓨터 프로그램을 저장한다. 컴퓨터 프로그램이 제 1 프로세서에 의해 실행될 경우에 제 1 양태의 방법을 구현하거나, 또는 컴퓨터 프로그램이 제 2 프로세서에 의해 실행될 경우에 제 2 양태의 방법을 구현한다.
- [0039] 본 출원의 실시예는 인코딩 방법, 디코딩 방법, 인코더, 디코더 및 저장 매체를 제공한다. 인코더 측에서, 현재 블록의 예측 파라미터를 확정한 다음에, 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하며, MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하고, MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하며, 현재 블록의 크로마 컴포넌트의 인트라 예측값에 따라 현재 블록의 크로마 컴포넌트의 예측 잔차값을 확정하고, 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정하고, LFNST 파라미터를 인코딩하고 비트스트림으로 기입한다. 디코더 측에서, 비트스트림을 파싱하여 현재 블록의 예측 파라미터 및 LFNST 파라미터를 획득한 다음에, 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하고, MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하고, MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하고, LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행함을 지시할 경우, 현재 블록의 재구성된 변환 계수 블록을 확정하고, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득하고, 제 2 변환 계수 블록에 대하여 첫번째 변환을 수행함으로써 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 획득하고, 현재 블록의 크로마 컴포넌트의 인트라 예측값 및 재구성된 잔차 블록에 따라 현재 블록의 크로마 컴포넌트의 재구성된 블록을 확정한다. 이와 같이, MIP 모드에 대하여, 인코딩 및 디코딩 성능을 보장할 뿐만 아니라 복잡성을 줄일 수 있고, 인코딩 및 디코딩 프로세스에 필요한 저장 공간을 줄일 수 있으며, 인코딩 및 디코딩 효율을 효과적으로 향상시킬 수 있다. 또한, LFNST 기술이 MIP 모드 예측에 적용될 경우, MIP 파라미터의 도입으로 인해 LFNST 변환이 더욱 유연해질 수 있고, 인코딩 및 디코딩 효율이 더욱 향상될 수 있다.

도면의 간단한 설명

- [0040] 도 1은 본 출원의 실시예에서 제공되는 인코더의 구조를 나타내는 개략도이다.
- 도 2는 본 출원의 실시예에서 제공되는 디코더의 구조를 나타내는 개략도이다.
- 도 3은 본 출원의 실시예에서 제공되는 인코딩 방법의 흐름도이다.
- 도 4는 본 출원의 실시예에서 제공되는 다른 인코딩 방법의 흐름도이다.
- 도 5는 본 출원의 실시예에서 제공되는 매트릭스 MIP(Matrix-Based Intra Prediction) 예측 프로세스의 흐름도이다.

도 6은 본 출원의 실시예에서 제공되는 MIP 모드를 사용하여 인트라 예측 블록을 획득하는 흐름도이다.

도 7은 본 출원의 실시예에서 제공되는 LFNST(Low Frequency Non-separable Transform)를 수행하는 절차를 도시하는 도면이다.

도 8은 본 출원의 실시예에서 제공되는 LFNST 파라미터를 인코딩하는 흐름도이다.

도 9는 본 출원의 실시예에서 제공되는 디코딩 방법의 흐름도이다.

도 10은 본 출원의 실시예에서 제공되는 LFNST 파라미터를 파싱하는 흐름도이다.

도 11은 본 출원의 실시예에서 제공되는 LFNST를 수행하는 절차를 도시하는 또 다른 도면이다.

도 12는 본 출원의 실시예에서 제공되는 다른 인코더의 구조를 나타내는 개략도이다.

도 13은 본 출원의 실시예에서 제공되는 인코더의 구체적인 하드웨어 구조를 나타내는 개략도이다.

도 14는 본 출원의 실시예에서 제공되는 다른 디코더의 구조를 나타내는 개략도이다.

도 15는 본 출원의 실시예에서 제공되는 디코더의 구체적인 하드웨어 구조를 나타내는 개략도이다.

도 16은 본 출원의 실시예에서 제공되는 송신 장치의 구조를 나타내는 개략도이다.

도 17은 본 출원의 실시예에서 제공되는 타겟 디바이스의 구조를 나타내는 개략도이다.

도 18은 본 출원의 실시예에서 제공되는 통신 시스템의 구조를 나타내는 개략도이다.

발명을 실시하기 위한 구체적인 내용

[0041] 본 출원의 실시예의 특징 및 기술적 내용을 보다 구체적으로 이해하기 위하여, 이하, 첨부된 도면을 참조하여 본 출원의 실시예를 상세하게 설명한다. 첨부된 도면은 단지 참조 및 설명하는 데에 사용될 뿐이지, 본 출원의 실시예를 한정하고자 하는 것은 아니다.

[0042] 본 명세서에서 사용되는 모든 기술 및 과학 용어는 달리 정의되지 않는 한, 본 출원의 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 갖는다. 본 명세서에서 사용되는 용어는 단지 본 출원의 실시예를 설명하기 위한 것으로, 본 출원을 한정하고자 하는 것은 아니다.

[0043] 다음 설명에서, 언급된 "일부 실시예"는 모든 가능한 실시예의 서브 세트를 설명하지만, "일부 실시예"는 모든 가능한 실시예의 동일한 서브 세트 또는 다른 서브 세트일 수 있으며, 충돌이 없는 한 서로 결합될 수 있음을 이해할 수 있다. 본 출원의 실시예에서 언급된 "제 1/제 2/제 3"이라는 용어는 단지 유사한 대상을 구별하는 데에 사용되며, 대상의 특정 순서를 나타내지 않는다는 점에 유념해야 한다. "제 1/제 2/제 3"과 관련하여, 허용되는 경우에 특정 순서 또는 선후 순서를 서로 교환할 수 있으며, 여기서 설명되는 본 출원의 실시예가 여기서 도시되거나 설명된 것과 다른 순서로 실현될 수 있도록 함을 이해할 수 있다.

[0044] 비디오 압축 기술이 발전됨에 따라, ITU-T(International Telecommunication Union-Transmission)와 ISO(International Organization for Standardization)/IEC(International Electrotechnical Commission)는 차세대 비디오 인코딩 표준을 개발하기 위해 VVC(Versatile Video Coding)라는 표준화 프로젝트를 시작하였다. 그 목적은 높은 해상도, 높은 프레임 속도, 높은 비트 심도, 높은 동적 범위, 넓은 색 영역 및 전방향 시야각 중 하나 이상의 특징을 갖는 고화질 비디오를 인코딩할 때, H.265/HEVC 표준과 비교하면, VVC 성능이 약 50% 향상될 수 있도록 하는 것이다. ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11의 공동 비디오 전문가 팀(joint video experts team, JVET)이 상기 표준화 프로젝트를 담당한다. 다양한 인트라 예측 모드와 인터 예측 모드는 고화질 비디오를 부호화할 때 높은 압축 효율을 실현할 수 있음이 검증되어, VVC WD(Working Draft)에 채택되었다.

[0045] 여기서, MIP(Matrix-Based Intra Prediction) 모드는 인트라 예측 모드이다. VVC에서 인코더 또는 디코더는 MIP 모드를 호출하여 현재 블록의 인트라 예측 블록을 획득할 수 있다. 그 다음에, 현재 블록의 예측 잔차에 대하여 부차적인 변환(또는 2차 변환이라고 함)인 LFNST(Low Frequency Non-separable Transform)를 적용할 수 있다.

[0046] 관련 기술에서, LFNST 변환 커널 후보 집합과 LFNST 전치 지시 파라미터를 갖는 고정된 구성의 MIP 모드 예측에 LFNST를 적용할 수 있지만, 이것은 VVC의 코딩 효율에 부정적인 영향을 미친다.

- [0047] 따라서, 본 출원의 실시예는 인코딩 방법을 제공하며, 인코딩 방법의 기본 사상은 다음과 같다. 현재 블록의 예측 파라미터를 확정하며, 예측 파라미터는 예측 모드 파라미터를 포함한다. 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정한다. MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하며, MIP 예측값은 현재 블록의 크로마 컴포넌트 중 일부 샘플의 예측값이다. MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정한다. 현재 블록의 크로마 컴포넌트의 인트라 예측값에 따라 현재 블록의 크로마 컴포넌트의 예측 잔차값을 확정한다. 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정한다. LFNST 파라미터를 인코딩하고 비트스트림으로 기입한다. 본 출원의 다른 실시예는 디코딩 방법을 제공하며, 디코딩 방법의 기본 사상은 다음과 같다. 비트스트림을 파싱하여 현재 블록의 예측 파라미터 및 LFNST(Low-Frequency Non-Separable Transform) 파라미터를 획득하며, 예측 파라미터는 예측 모드 파라미터를 포함한다. 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정한다. MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하며, MIP 예측값은 현재 블록의 크로마 컴포넌트 중 일부 샘플의 예측값이다. MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정한다. LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행함을 지시할 경우, 현재 블록의 재구성된 변환 계수 블록을 확정하고, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득한다. 제 2 변환 계수 블록에 대하여 첫번째 변환을 수행함으로써 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 획득한다. 현재 블록의 크로마 컴포넌트의 인트라 예측값 및 재구성된 잔차 블록에 따라 현재 블록의 크로마 컴포넌트의 재구성된 블록을 확정한다. 이와 같이, MIP 모드에 대하여, 인코딩 및 디코딩 성능을 보장할 뿐만 아니라 복잡성을 줄일 수 있고, 인코딩 및 디코딩 프로세스에 필요한 저장 공간을 줄일 수 있으며, 인코딩 및 디코딩 효율을 효과적으로 향상시킬 수 있다. 또한, LFNST 기술이 MIP 모드 예측에 적용될 경우, MIP 파라미터의 도입으로 인해 LFNST 변환이 더욱 유연해질 수 있고, 인코딩 및 디코딩 효율이 더욱 향상될 수 있다.
- [0048] 이하, 첨부된 도면을 참조하여 본 출원의 실시예를 상세하게 설명한다.
- [0049] 도 1을 참조하면, 도 1은 본 출원의 실시예에서 제공되는 인코더의 시스템 구조를 나타내는 블록도이다. 인코더(100)는 분할 유닛(101), 예측 유닛(102), 제 1 가산기(107), 변환 유닛(108), 양자화 유닛(109), 역양자화 유닛(110), 역변환 유닛(111), 제 2 가산기(112), 필터링 유닛(113), 디코딩된 픽처 버퍼(Decoded Picture Buffer, DPB) 유닛(114) 및 엔트로피 코딩 유닛(115)을 포함한다. 여기서, 인코더(100)의 입력은 일련의 픽처 또는 하나의 정지 픽처로 구성된 비디오일 수 있고, 인코더(100)의 출력은 입력 비디오의 압축된 버전의 비트스트림("코드스트림"이라고도 함)을 나타내는 데에 사용된다.
- [0050] 분할 유닛(101)은 입력 비디오의 픽처를 하나 이상의 CTU(Coding Tree Unit)로 분할한다. 분할 유닛(101)은 픽처를 복수의 타일(tile)로 분할하고, 하나의 타일을 하나 이상의 브릭(brick)으로 더 분할할 수 있다. 여기서, 하나의 타일 또는 하나의 브릭은 하나 이상의 완전한 및/또는 부분 CTU를 포함할 수 있다. 또한, 분할 유닛(101)은 하나 이상의 슬라이스(slice)를 형성할 수 있으며, 하나의 슬라이스는 픽처에서 래스터 순서로 배열된 하나 이상의 타일을 포함하거나, 픽처 중 직사각형 영역을 덮는 하나 이상의 타일을 포함할 수 있다. 분할 유닛(101)은 또한 하나 이상의 서브-픽처를 형성할 수 있으며, 하나의 서브-픽처는 하나 이상의 슬라이스, 타일 또는 브릭을 포함할 수 있다.
- [0051] 인코더(100)의 인코딩 프로세스에서 분할 유닛(101)은 예측 유닛(102)에 CTU를 전달한다. 일반적으로, 예측 유닛(102)은 블록 분할 유닛(103), 모션 추정(motion estimation, ME) 유닛(104), 모션 보상(motion compensation, MC) 유닛(105), 인트라 예측 유닛(106)으로 구성될 수 있다. 구체적으로, 블록 분할 유닛(103)은 쿼드트리 분할(quadtrees split), 바이너리 분할(binary split) 및 트리너리 분할(ternary split)을 반복적으로 사용하여 입력 CTU를 더 작은 코딩 유닛(coding unit, CU)으로 추가 분할한다. 예측 유닛(102)은 ME 유닛(104) 및 MC 유닛(105)을 사용하여 CU의 인터 예측 블록을 획득할 수 있다. 인트라 예측 유닛(106)은 MIP 모드를 포함하는 다양한 인트라 예측 모드를 사용하여 CU의 인트라 예측 블록을 획득할 수 있다. 예시에서, 레이트 왜곡 최적화된 모션 추정 방식은 인터 예측 블록을 획득하기 위해 ME 유닛(104) 및 MC 유닛(105)에 의해 호출될 수 있고, 레이트 왜곡 최적화 모드 확정 방식은 인트라 예측 블록을 획득하기 위해 인트라 예측 유닛(106)에 의해 호출될 수 있다.

- [0052] 예측 유닛(102)은 CU의 예측 블록을 출력하고, 제 1 가산기(107)는 분할 유닛(101)의 출력 중의 CU와 CU의 예측 블록 간의 차이값, 즉 잔차 CU를 계산한다. 변환 유닛(108)은 잔차 CU를 판독하고, 잔차 CU에 대하여 하나 이상의 변환 조작을 수행하여 계수를 획득한다. 양자화 유닛(109)은 계수를 양자화하여 양자화된 계수(즉, 레벨(levels))를 출력한다. 역양자화 유닛(110)은 양자화된 계수를 스케일링(scaling)하여 재구성된 계수를 출력한다. 역변환 유닛(111)은 변환 유닛(108)의 변환에 대응하는 하나 이상의 역변환을 수행하고 재구성된 잔차를 출력한다. 제 2 가산기(112)는 재구성된 잔차와 예측 유닛(102)으로부터의 CU의 예측 블록을 가산함으로써 재구성된 CU를 계산한다. 제 2 가산기(112)는 또한 그 출력을 예측 유닛(102)으로 송신하여 인트라 예측 참조로서 사용한다. 픽처 또는 서브-픽처 내의 모든 CU가 재구성된 후에, 필터링 유닛(113)은 재구성된 픽처 또는 서브-픽처에 대하여 인-루프 필터링을 수행한다. 필터링 유닛(113)은 디블록킹 필터, SAO(Sample Adaptive Offset) 필터, ALF(Adaptive Loop Filter), LMCS(luma mapping with chroma scaling) 필터 및 신경망의 필터 등과 같은 하나 이상의 필터를 포함한다. 또는, 필터링 유닛(113)이 CU가 다른 CU를 인코딩하기 위한 기준으로 사용되지 않는다고 확정할 때, 필터링 유닛(113)은 CU 내의 하나 이상의 타겟 픽셀에 대하여 인루프 필터링을 수행한다.
- [0053] 필터링 유닛(113)의 출력은 디코딩된 픽처 또는 서브 픽처이며, 이러한 디코딩된 픽처 또는 서브 픽처는 DPB 유닛(114)으로 버퍼링된다. DPB 유닛(114)은 타이밍 및 제어 정보에 따라 디코딩된 픽처 또는 서브 픽처를 출력한다. 여기서, DPB 유닛(114)에 저장된 픽처는 예측 유닛(102)에서 인터 예측 또는 인트라 예측을 수행하기 위한 참조로 사용될 수도 있다. 마지막으로, 엔트로피 코딩 유닛(115)은 인코더(100)로부터의 픽처를 디코딩하는 데에 필요한 파라미터(예를 들면, 제어 파라미터 및 추가 정보 등)를 바이너리 형태로 변환하고, 각 데이터 유닛의 선택스 구조에 따라 이러한 바이너리 형태를 비트스트림에 기록한다. 즉, 인코더(100)는 최종적으로 비트스트림을 출력한다.
- [0054] 또한, 인코더(100)는 제 1 프로세서 및 컴퓨터 프로그램을 기록하는 제 1 메모리를 가질 수 있다. 제 1 프로세서가 컴퓨터 프로그램을 판독하고 실행할 때, 인코더(100)는 입력 비디오를 판독하고 대응하는 비트스트림을 생성한다. 또한, 인코더(100)는 또한 하나 이상의 칩을 갖는 컴퓨팅 장치일 수 있다. 칩에서 집적회로로 구현된 이러한 유닛은 도 1의 대응하는 유닛과 유사한 연결 및 데이터 교환 기능을 갖는다.
- [0055] 도 2를 참조하면, 본 출원의 실시예에서 제공되는 디코더의 시스템 구조의 블록도의 예시를 도시한다. 도 2에 도시된 바와 같이, 디코더(200)는 파싱 유닛(201), 예측 유닛(202), 스케일링 유닛(205), 변환 유닛(206), 가산기(207), 필터링 유닛(208) 및 디코딩된 픽처 버퍼 유닛(209)을 포함할 수 있다. 디코더(200)의 입력은 비디오 또는 정적 픽처의 압축된 버전을 나타내기 위한 비트스트림이고, 디코더(200)의 출력은 일련의 픽처로 구성된 디코딩된 비디오 또는 한장의 디코딩된 정적 픽처일 수 있다.
- [0056] 디코더(200)의 입력 비트스트림은 인코더(100)에 의해 생성된 비트스트림일 수 있다. 파싱 유닛(201)은 입력된 비트스트림을 파싱하고, 입력된 비트스트림으로부터 선택스 요소의 값을 획득한다. 파싱 유닛(201)은 선택스 요소의 이진 표현을 수치 값으로 변환하고, 수치 값을 디코더(200)의 유닛에 전송하여 하나 이상의 디코딩된 픽처를 획득한다. 파싱 유닛(201)은 또한 디코딩된 픽처를 표시하기 위해 입력 비트스트림으로부터 하나 이상의 선택스 요소를 파싱할 수 있다.
- [0057] 디코더(200)의 디코딩 과정에서, 파싱 유닛(201)은 선택스 요소의 값 및 선택스 요소의 값에 따라 설정되거나 확정된, 하나 이상의 디코딩된 픽처를 획득하기 위한 하나 이상의 변수를 디코더(200)의 유닛에 송신한다.
- [0058] 예측 유닛(202)은 현재 디코딩 블록(예를 들어, CU)의 예측 블록을 확정한다. 예측 유닛(202)은 모션 보상 유닛(203) 및 인트라 예측 유닛(204)을 포함할 수 있다. 구체적으로, 인터 디코딩 모드는 현재 디코딩 블록을 디코딩하는 데에 사용된다고 지시되는 경우, 예측 유닛(202)은 파싱 유닛(201)으로부터의 관련 파라미터를 모션 보상 유닛(203)으로 전달하여 인터 예측 블록을 획득한다. 인트라 예측 모드(MIP 모드 인덱스 값을 기반으로 지시된 MIP 모드를 포함함)는 현재 디코딩 블록을 디코딩하는 데에 사용된다고 지시되는 경우, 예측 유닛(202)은 파싱 유닛(201)으로부터의 관련 파라미터를 인트라 예측 유닛(204)으로 전달하여 인트라 예측 블록을 획득한다.
- [0059] 스케일링 유닛(205)은 인코더(100)의 역양자화 유닛(110)과 동일한 기능을 갖는다. 스케일링 유닛(205)은 파싱 유닛(201)으로부터의 양자화된 계수(즉, levels)에 대하여 스케일링 조작을 수행하여 재구성된 계수를 획득한다.
- [0060] 변환 유닛(206)은 인코더(100)의 역변환 유닛(111)과 동일한 기능을 갖는다. 변환 유닛(206)은 하나 이상의 변환 조작(즉, 인코더(100)의 역변환 유닛(111)에 의해 수행되는 하나 이상의 변환 조작의 역조작)을 수행하여 재구성된 잔차를 획득한다.

- [0061] 가산기(207)는 자신의 입력(예측 유닛(202)으로부터의 예측 블록과 변환 유닛(206)으로부터의 재구성된 잔차)에 대하여 가산 조작을 수행하여 현재 디코딩 블록의 재구성된 블록을 획득한다. 재구성된 블록은 또한 예측 유닛(202)으로 송신됨으로써, 인트라 예측 모드에서 인코딩된 다른 블록에 대한 참조로서 사용되도록 한다.
- [0062] 픽처 또는 서브 픽처 내의 모든 CU가 재구성된 후에, 필터링 유닛(208)은 재구성된 픽처 또는 서브 픽처에 대하여 인루프 필터링을 수행한다. 필터링 유닛(208)은 디블록킹 필터, 샘플 적응 보상 필터, 적응 루프 필터(Adaptive Loop Filter), 루마 매핑 크로마 스케일링 필터(luma mapping with chroma scaling filter) 등과 같은 하나 이상의 필터를 포함한다. 또는, 필터링 유닛(208)이 재구성된 블록이 다른 블록을 디코딩하기 위한 참조로서 사용되지 않는다고 확정할 때, 필터링 유닛(208)은 재구성된 블록 중의 하나 이상의 타겟 픽셀에 대하여 인루프 필터링을 수행한다. 여기서, 필터링 유닛(208)의 출력은 디코딩된 픽처 또는 서브 픽처이고, 디코딩된 픽처 또는 서브 픽처는 DPB 유닛(209)에 버퍼링된다. DPB 유닛(209)은 타이밍 및 제어 정보에 따라 디코딩된 픽처 또는 서브 픽처를 출력한다. DPB부(209)에 저장된 픽처는 또한 예측 유닛(202)에 의한 인터 예측 또는 인트라 예측을 수행하기 위한 참조로서 사용될 수 있다.
- [0063] 또한, 디코더(200)는 제 2 프로세서 및 컴퓨터 프로그램을 기록하는 제 2 메모리를 가질 수 있다. 제 1 프로세서가 컴퓨터 프로그램을 판독하고 실행할 때, 디코더(200)는 입력된 비트스트림을 판독하고 대응하는 디코딩된 비디오를 생성한다. 또한, 디코더(200)는 또한 하나 이상의 칩을 갖는 컴퓨팅 장치일 수 있다. 칩에서 집적회로로 구현된 이러한 유닛은 도 2의 대응하는 유닛과 유사한 연결 및 데이터 교환 기능을 갖는다.
- [0064] 본 출원의 실시예에서 제공되는 인코딩 방법은 주로 인코더(100)의 인트라 예측 유닛(106) 및 변환 유닛(108)에 적용되며, 본 출원의 실시예에서 제공되는 디코딩 방법은 주로 디코더(200)의 인트라 예측 유닛(204) 및 변환 유닛(206)에 적용된다. 즉, 본 출원의 실시예는 인코더, 디코더, 또는 인코더와 디코더 모두에 동시에 적용될 수 있으며, 이것에 대하여 한정하지 않는다.
- [0065] 또한, 본 출원의 실시예가 인코더(100)에 적용되는 경우, "현재 블록"은 구체적으로 인트라 예측에서 현재 인코딩될 픽처 블록("코딩 블록"으로도 지칭될 수 있음)을 가리키며, 본 출원의 실시예가 디코더(200)에 적용되는 경우, "현재 블록"은 구체적으로 인트라 예측에서 현재 디코딩될 픽처 블록("디코딩 블록"으로도 지칭될 수 있음)을 가리킨다.
- [0066] 인코더(100)가 본 출원의 실시예에서 제공하는 인코딩 방법을 이용하여 더 나은 예측 효과를 얻을 수 있는 경우, 부호화 성능을 향상시키는 데에 사용되며, 대응하게 디코더(200)는 본 출원의 실시예에서 제공하는 디코딩 방법을 이용하여 비디오 디코딩 품질을 향상시킬 수 있으며, 따라서 디코딩 성능을 향상시킨다.
- [0067] 본 출원의 하나의 실시예에 있어서, 도 3은 본 출원의 실시예에서 제공되는 인코딩 방법의 흐름도이다. 도 3에 도시된 바와 같이, 상기 방법은 아래 내용을 포함한다.
- [0068] S301, 현재 블록의 예측 파라미터를 확정하고, 예측 파라미터는 예측 모드 파라미터를 포함한다.
- [0069] 인코딩될 픽처는 복수개의 픽처 블록으로 분할될 수 있으며, 각 인코딩될 픽처 블록은 코딩 블록이라고 할 수 있다. 여기서, 각 코딩 블록은 제 1 화상 컴포넌트, 제 2 화상 컴포넌트 및 제 3 화상 컴포넌트를 포함할 수 있다. 현재 블록은 비디오의 픽처에 있어서의 현재 제 1 화상 컴포넌트, 제 2 화상 컴포넌트, 또는 제 3 화상 컴포넌트의 예측이 수행될 코딩 블록이다.
- [0070] 현재 블록에 대하여 제 1 화상 컴포넌트의 예측이 수행되고, 제 1 화상 컴포넌트는 휘도 컴포넌트이고, 즉 예측할 화상 컴포넌트가 휘도 컴포넌트이라고 가정하면, 현재 블록은 휘도 블록이라고도 할 수 있으며, 현재 블록의 휘도 컴포넌트의 예측 값을 획득한다. 또는, 현재 블록에 대하여 제 2 화상 컴포넌트의 예측이 수행되고, 제 2 화상 컴포넌트가 크로마 컴포넌트이고, 즉 예측할 화상 컴포넌트가 크로마 컴포넌트라고 가정하면, 현재 블록을 크로마 블록이라고도 할 수 있으며, 현재 블록의 휘도 컴포넌트의 예측값을 획득한다.
- [0071] 또한, 예측 파라미터는 예측 모드 파라미터를 포함할 수 있으며, 예측 모드 파라미터는 현재 블록이 사용하는 예측 모드를 지시하는 데에 사용되고, 상이한 예측 모드는 상이한 예측 모드 파라미터에 대응된다는 점에 유념해야 한다. 예측 모드 파라미터를 확정하는 것에 관하여, 간단한 의사 결정 전략을 사용할 수 있으며, 예를 들어, 왜곡 값의 크기에 따라 확정하거나, 복잡한 의사 결정 전략을 사용할 수도 있으며, 예를 들어, RDO(Rate Distortion Optimization)의 결과에 따라 확정하며, 본 출원의 실시예는 이것에 대하여 한정하지 않는다. 일반적으로 RDO 방식을 사용하여 현재 블록의 예측 모드 파라미터를 확정할 수 있으며, 즉 현재 블록의 예측 파라미터를 확정한다.

- [0072] 구체적으로, 일부 실시예에서, 현재 블록의 예측 파라미터를 확정하는 것은 아래 내용을 포함할 수 있다.
- [0073] 다양한 예측 모드를 이용하여 현재 블록에 대하여 프리코딩 처리를 수행하여 다양한 예측 모드에 대응하는 비율 왜곡 대가 값을 획득한다.
- [0074] 획득한 복수의 비율 왜곡 대가 값에서 최적의 비율 왜곡 대가 값을 선택하고, 최적의 비율 왜곡 대가 값에 대응하는 예측 모드에서의 예측 파라미터를 현재 블록의 예측 파라미터로 확정한다.
- [0075] 즉, 인코더 측에서, 현재 블록에 대하여, 다양한 예측 모드를 이용하여 현재 블록에 대하여 프리코딩 처리를 수행할 수 있다. 다양한 예측 모드는 일반적으로 인터 예측 모드, 전통적인 인트라 예측 모드 및 비전통적인 인트라 예측 모드를 포함한다. 전통적인 인트라 예측 모드는 직류(direct current, DC) 모드, 평면(PLANAR) 모드, 각도 인트라 예측 모드 등을 포함할 수 있다. 비전통적인 인트라 예측 모드는 MIP 모드, CCLM(Cross-Component Linear Model Prediction) 모드, IBC(Intra Block Copy) 모드 및 PLT(Palette) 모드 등을 포함할 수 있다. 인터 예측 모드는 전통적인 인터 예측 모드 및 블록 간 기하학적 분할 예측(Geometrical partitioning for inter blocks, GEO) 모드 등을 포함할 수 있다.
- [0076] 이와 같이, 다양한 예측 모드를 이용하여 현재 블록에 대하여 프리코딩 처리를 수행한 다음에, 다양한 예측 모드에 대응하는 비율 왜곡 대가 값을 획득할 수 있다. 그 다음에, 획득한 복수의 비율 왜곡 대가 값에서 최적의 비율 왜곡 대가 값을 선택하고(일반적으로 최소 비율 왜곡 대가 값이 바로 최적의 비율 왜곡 대가 값이다), 최적의 비율 왜곡 대가 값에 대응하는 예측 모드에서의 예측 파라미터를 현재 블록의 예측 파라미터로 확정한다.
- [0077] 또한, 다양한 예측 모드를 이용하여 별도로 현재 블록에 대하여 프리코딩 처리를 수행하여 다양한 예측 모드에 대응하는 왜곡 값을 획득할 수 있다. 획득한 복수의 왜곡 값에서 최적의 왜곡 값을 선택하고, 최적의 왜곡 값에 대응하는 예측 모드에서의 예측 파라미터를 현재 블록의 예측 파라미터로 확정한다. 이와 같이, 인코더는 최종적으로 확정된 예측 파라미터 중 예측 모드 파라미터가 지시하는 예측 모드를 사용하여 현재 블록에 대하여 예측 코딩을 수행하며, 예측 잔차가 더 작아질 수 있고, 코딩 효율이 향상될 수 있다.
- [0078] S302, 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정한다.
- [0079] 현재 블록에 대하여, 본 출원의 실시예는 MIP 모드를 이용하여 현재 블록의 크로마 컴포넌트에 대하여 인트라 예측을 수행한다는 점에 유념해야 한다. 이 과정에서 먼저 현재 블록의 인접 샘플값을 획득할 필요가 있고, 그 다음에 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정한다.
- [0080] MIP 모드에 대하여, 먼저 MIP 코어 파라미터를 설정해야 함을 이해할 수 있다. MIP 코어 파라미터는 현재 블록의 유형(mipSizeId로 나타냄), 각 변의 참조 샘플의 개수(boundySize로 나타냄), MIP 입력 샘플의 개수(inSize로 나타냄) 및 행렬 곱셈에서 출력되는 MIP 예측 블록 크기(predSize×predSize로 배열됨)를 포함할 수 있다. MIP 모드에서는 현재 블록의 너비와 높이에 따라 현재 블록을 세가지 유형으로 분류할 수 있으며, mipSizeId는 0, 1 또는 2와 같을 수 있고, mipSizeId는 현재 블록의 유형을 나타내며, 즉 본 출원의 실시예에서 언급된 "현재 블록의 블록 크기 인덱스 값"이다. 서로 다른 mipSizeId에 대하여, 참조 샘플의 개수(각 변은 boundySize개의 참조 샘플이 필요됨), MIP 입력 샘플의 개수(inSize), 행렬 곱셈에서 출력되는 MIP 예측 블록의 크기(predSize×predSize로 배열됨)도 다르다.
- [0081] 또한, 예측 파라미터는 예측 모드 파라미터 외에 현재 블록의 크기 파라미터를 더 포함할 수 있음에 유념해야 한다. 현재 블록의 크기 파라미터는 현재 블록의 너비(nTbW로 나타냄) 및 높이(nTbH로 나타냄)를 포함할 수 있다. 또한, 현재 블록의 크기 파라미터에 따라 현재 블록의 블록 크기 인덱스 값(즉, mipSizeId)을 확정할 수 있다.
- [0082] 한가지 가능한 실시예에서, 현재 블록의 크기 파라미터에 따라 현재 블록의 블록 크기 인덱스 값을 확정하는 것은 아래 내용을 포함한다.
- [0083] 현재 블록의 너비와 높이가 모두 4인 경우, 현재 블록의 블록 크기 인덱스 값은 0으로 설정될 수 있다.
- [0084] 그렇지 않고, 현재 블록의 너비와 높이가 모두 8이거나, 또는 현재 블록의 너비 및 높이 중 하나가 4인 경우, 현재 블록의 블록 크기 인덱스 값은 1로 설정될 수 있다.
- [0085] 그렇지 않고, 현재 블록이 다른 크기의 블록인 경우, 현재 블록의 블록 크기 인덱스 값은 2로 설정될 수

있다.

[0086] 다른 한가지 가능한 실시예에서, 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정하는 것은 아래 내용을 포함한다.

[0087] 현재 블록의 너비와 높이가 모두 4인 경우, 현재 블록의 블록 사이즈 인덱스 값은 0으로 설정될 수 있다.

[0088] 그렇지 않고, 현재 블록의 너비와 높이 중 하나가 4인 경우, 현재 블록의 블록 사이즈 인덱스 값은 1로 설정될 수 있다.

[0089] 그렇지 않고, 현재 블록이 다른 크기의 블록인 경우, 현재 블록의 블록 사이즈 인덱스 값은 2로 설정될 수 있다.

[0090] 이와 같이, 현재 블록의 블록 사이즈 인덱스 값에 따라, 표 1에 표시된 룩업 테이블(look-up table, LUT)을 통해 인접 경계 참조 샘플의 개수(변수는 boundarySize임)와 MIP 예측 블록의 크기(변수는 predSize이고, MIP 예측 블록의 크기는predSize×predSize임)를 확정할 수 있으며, MIP 행렬 곱셈 연산 과정에 사용되는 MIP 입력 샘플값의 개수(inSize로 나타냄)를 계산한다. 계산식은 다음과 같다.

[0091]
$$inSize=(2 \times boundarySize)-(mipSizeId==2)?1:0 \quad (1)$$

[0092] 식 (1)에서 연산자의 연산 규칙은 ITU-T H.265 표준에서 정의된 연산자의 연산 규칙과 동일하며, 예를 들어 "=="는 논리적 "같음" 연산자이다.

[0093] [표 1]

mipSizeId	boundarySize	predSize
0	2	4
1	4	4
2	4	8

[0094]

[0095] 이와 같이, 표 1에 따르면, mipSizeId의 값이 0일 때, boundarySize의 값은 2이고, predSize의 값은 4이다. 즉, 이 때 참조 픽셀은 각 예지별로 2개의 픽셀이 선택되고, 행렬 곱셈의 출력은 4x4의 MIP 예측 블록이다.

[0096] mipSizeId의 값이 1인 경우, boundarySize의 값은 4이고, predSize의 값은 4이다. 즉, 이 때 참조 픽셀은 각 예지별로 4개의 픽셀이 선택되고, 행렬 곱셈의 출력은 4x4의 MIP 예측 블록이다.

[0097] mipSizeId의 값이 2인 경우, boundarySize의 값은 4이고, predSize의 값은 8이다. 즉, 이 때 참조 픽셀은 각 예지별로 4개의 픽셀이 선택되고, 행렬 곱셈의 출력은 8x8의 MIP 예측 블록이다.

[0098] 또한, 현재 블록의 블록 사이즈 인덱스 값에 따라 표 2에 표시된 룩업 테이블을 통해 boundarySize, inSize, predSize의 값을 동시에 확정할 수 있다.

[0099] [표 2]

mipSizeId	boundarySize	inSize	predSize
0	2	4	4
1	4	8	4
2	4	7	8

[0100]

[0101] 이와 같이, 표 2에 따르면, mipSizeId의 값이 0일 때, boundarySize의 값은 2이고, inSize의 값은 4이며, predSize의 값은 4이다. 즉, 이 때 참조 샘플은 각 예지별로 2개의 픽셀이 선택되고, 행렬 곱셈 입력 샘플의 수량은 4이고, 행렬 곱셈의 출력은 4×4의 MIP 예측 블록이다.

[0102] mipSizeId의 값이 1일 때, boundarySize의 값은 4이고, inSize의 값은 8이며, predSize의 값은 4이다. 즉, 이 때 참조 샘플은 각 예지별로 4개의 픽셀이 선택되고, 행렬 곱셈 입력 샘플의 수량은 8이고, 행렬 곱셈의 출력은

4×4의 MIP 예측 블록이다.

- [0103] mipSizeId의 값이 2일 때, boundarySize의 값은 4이고, inSize의 값은 7이며, predSize의 값은 8이다. 즉, 이때 참조 샘플은 각 에지별로 4개의 픽셀이 선택되고, 행렬 곱셈 입력 샘플의 수량은 7이고, 행렬 곱셈의 출력은 8×8의 MIP 예측 블록이다.
- [0104] 또한, MIP 코어 파라미터를 설정한 후에 참조 픽셀을 획득해야 한다. 그 다음에 참조 픽셀 및 MIP 코어 파라미터에 따라 MIP 입력 샘플값이 구성된다. 여기서, 참조 픽셀은 현재 블록의 왼쪽 인접 샘플값 및 현재 블록의 위쪽 인접 샘플값을 포함하는 현재 블록의 인접 샘플값을 의미한다. 즉, 현재 블록의 왼쪽 인접 샘플값과 위쪽 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값이 확정될 수 있다.
- [0105] 일부 실시예에서, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하는 것은 아래 내용을 포함할 수 있다.
- [0106] 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정한다.
- [0107] 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 제 1 임시 참조값을 획득한다.
- [0108] 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우, 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정한다. MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 제 2 상수값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정한다. MIP 입력 샘플값 중 인덱스 번호 i에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 i에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하며, 여기서 i는 0보다 큰 정수이다.
- [0109] 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위를 벗어나는 경우, MIP 입력 샘플값 중 인덱스 번호 j에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 (j+1)에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하며, j는 0 이상의 정수이다.
- [0110] 또한, 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우, 상기 방법은 다음 내용을 더 포함할 수 있다.
- [0111] 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정한다. MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값과 제 2 상수값의 차이로 설정한다. MIP 입력 샘플값 중 인덱스 번호 i에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 i에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하며, 여기서 i는 0보다 큰 정수이다.
- [0112] 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링 처리를 수행한 다음에 제 1 임시 참조값을 획득할 수 있다. 제 1 임시 참조값에 대하여, 구체적으로 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링 처리를 수행한 다음에 필터링된 인접 샘플값을 버퍼(pTemp로 나타냄)에 버퍼링할 수 있다. 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값은 pTemp[0]이고, 제 1 임시 참조값 중 인덱스 번호 i에 대응하는 값은 pTemp[i]이다.
- [0113] 또한, 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄)이 미리 설정된 범위 내에 있는지 여부에 따라 현재 블록의 사이즈 파라미터의 값이 미리 설정된 범위 내에 있는지 여부를 확인할 수 있다. 구체적으로, mipSizeId=0 또는 1인 경우, 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있음을 나타내고, 즉 현재 블록의 사이즈 파라미터의 값이 미리 설정된 범위 내에 있음을 나타낸다. mipSizeId=2인 경우, 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위를 벗어났음을 나타내고, 즉 현재 블록의 사이즈 파라미터의 값이 미리 설정된 범위 내에 있지 않음을 나타낸다.
- [0114] 즉, MIP 입력 샘플값은 버퍼(pTemp로 나타냄), 현재 블록의 블록 사이즈 인덱스 값(MipSizeId로 나타냄), 현재 블록의 인접 샘플값의 비트 깊이(BitDepth로 나타냄)에 따라 결정된다. MIP 입력 샘플값에 포함된 입력 샘플 개수는 단지 현재 블록의 블록 사이즈 인덱스 값에만 관련되고, 마지막으로 MIP 입력 샘플값 중 인덱스 번호 x에 대응하는 값(p[x]로 나타냄)을 획득할 수 있다.
- [0115] 또한, 일부 실시예에 있어서, 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정하는 것은 아래 내용을 포함할 수 있다.
- [0116] 제 2 상수값을 2의 정수 거듭제곱과 동일하게 설정하며, 거듭제곱의 지수는 현재 블록의 인접 샘플값의 비트 깊이에서 1을 뺀 값과 같다.

- [0117] 또한, 일부 실시예에 있어서, 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정하는 것은 아래 내용을 포함할 수 있다.
- [0118] "1"에 대하여 이진 왼쪽 시프팅을 수행하여 제 2 상수값을 획득하고, 왼쪽으로 시프트된 비트의 수는 현재 블록의 인접 샘플값의 비트 깊이에서 1을 뺀 것과 같다.
- [0119] 즉, 현재 블록의 인접 샘플값의 비트 깊이(BitDepth로 나타냄)를 획득한 다음에, 제 2 상수값은 $1 \ll (\text{BitDepth} - 1)$ 또는 $2^{(\text{BitDepth} - 1)}$ 로 표시될 수 있다. 이와 같이, 현재 블록의 사이즈 파라미터의 값이 미리 설정된 범위 내에 있는 경우, 제 2 상수값을 결합하여 현재 블록의 MIP 입력 샘플값을 확정할 수 있다.
- [0120] 또한, MIP 입력 샘플은 행렬 곱셈 연산에 사용되는 행렬 벡터임을 유념해야 한다. 현재 관련 기술 방안은, 버퍼(pTemp로 나타냄), 현재 블록의 유형(즉, 현재 블록의 블록 사이즈 인덱스 값, mipSizeId로 나타냄), 현재 블록의 인접 샘플값의 비트 깊이(BitDepth로 나타냄), MIP 입력 샘플의 개수에 따라 확정되고, 마지막으로 MIP 입력 샘플값 중 인덱스 번호 x에 대응하는 값(p[x]로 나타냄)을 획득한다.
- [0121] 구체적으로, x번째 입력 샘플값 p[x]의 구성 과정은 다음과 같다.
- [0122] 가능한 실시예에서, 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우, 상기 구성 과정은 다음 내용을 포함할 수 있다.
- [0123] 제 2 상수값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이를 계산하여 MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 획득한다.
- [0124] 제 1 임시 참조값 중 인덱스 번호 i에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값에 대하여 감법 연산을 수행하여 MIP 입력 샘플값 중 인덱스 번호 i에 대응하는 값을 획득한다. i는 0보다 크고 N보다 작은 양의 정수이며, N은 입력 샘플 행렬에 포함된 요소의 수를 나타낸다.
- [0125] 본 출원의 실시예에서, 차이를 계산함에 있어서 피감수를 제 2 상수값과 동일하게 설정하고, 차이를 계산함에 있어서 감수를 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값과 동일하게 설정한다.
- [0126] 즉, mipSizeId=0 또는 1인 경우, $1 \ll (\text{BitDepth} - 1)$ 빼기 pTemp[0]을 p[0]으로 하고; x가 0과 같지 않으면, pTemp[x] 빼기 pTemp[0]을 p[x]로 한다. 구체적으로 다음과 같다.
- [0127]
$$\begin{cases} p[0] = (1 \ll (\text{BitDepth} - 1)) - p\text{Temp}[0] \\ p[x] = p\text{Temp}[x] - p\text{Temp}[0] \quad \text{for } x=1, \dots, \text{inSize}-1 \end{cases} \quad (2)$$
- [0128] 다른 가능한 실시예에서, 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우, 구성 과정은 다음 내용을 포함할 수 있다.
- [0129] 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값과 제 2 상수값의 차이를 계산하여, MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 획득한다.
- [0130] 제 1 임시 참조값 중 인덱스 번호 i에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값에 대하여 감법 연산을 수행하여 MIP 입력 샘플 값 중 인덱스 번호 i에 대응하는 값을 획득한다. i는 0보다 크고 N보다 작은 양의 정수이고, N은 입력 샘플 행렬에 포함된 요소의 수를 나타낸다.
- [0131] 본 출원의 실시예에서, 차이를 계산함에 있어서 피감수를 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값과 동일하게 설정하고, 차이를 계산함에 있어서 감수를 제 2 상수값과 동일하게 설정한다.
- [0132] 즉, mipSizeId=0 또는 1인 경우, x가 0인 경우, MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값(p[0]으로 나타냄)은 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값(pTemp[0]으로 나타냄)에서 제 2 상수값(즉, $1 \ll (\text{BitDepth} - 1)$)을 빼서 얻을 수 있다. x가 0이 아닌 경우, MIP 입력 샘플값 중 인덱스 번호 x에 대응하는 값(p[x]로 나타냄)은 제 1 임시 참조값 중 인덱스 번호 x에 대응하는 값(pTemp[x]로 나타냄)에서 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값(pTemp[0]으로 나타냄)을 빼서 얻을 수 있다. 구체적으로 다음과 같다.
- [0133]
$$\begin{cases} p[0] = p\text{Temp}[0] - (1 \ll (\text{BitDepth} - 1)) \\ p[x] = p\text{Temp}[x] - p\text{Temp}[0] \quad \text{for } x=1, \dots, \text{inSize}-1 \end{cases} \quad (3)$$
- [0134] 또 다른 가능한 실시예에서, 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위를 벗어나는 경우, 구성 과

정은 다음 내용을 포함할 수 있다.

- [0135] 제 1 임시 참조값 중 인덱스 번호 (i+1)에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값에 대하여 감법 연산을 수행하여 MIP 입력 샘플값 중 인덱스 번호 i에 대응하는 값을 획득한다. i는 0보다 크거나 같고 N보다 작은 양의 정수이고, N은 입력 샘플 행렬에 포함된 요소의 수량을 나타낸다.
- [0136] 즉, mipSizeId=2인 경우, 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값, 즉 pTemp[0]을 무시할 수 있다. MIP 입력 샘플값 중 인덱스 번호 x에 대응하는 값(p[x]로 나타냄)은 제 1 임시 참조값 중 인덱스 번호 (x+1)에 대응하는 값(즉, pTemp[x+1])에서 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값(즉, pTemp[0])을 빼서 얻을 수 있다. x는 0 이상의 양의 정수이다. 구체적으로 다음과 같다.
- [0137]
$$p[x]=pTemp[x+1]-pTemp[0] \quad \text{for } x=0,...,inSize-1 \quad (4)$$
- [0138] 여전히 4×4의 현재 블록을 예로 들어, 버퍼 pTemp에 4개의 값이 저장되어 있다. 즉, 제 1 임시 참조값은 인덱스 번호 0에 대응하는 값(즉, pTemp[0]), 인덱스 번호 1에 대응하는 값(즉, pTemp[1]), 인덱스 번호 2에 대응하는 값(즉, pTemp[2]) 및 인덱스 번호 3에 대응하는 값(즉, pTemp[3])을 포함한다. 이 때, 식 (2) 또는 식 (3) 또는 식 (4)에 따라 4개의 MIP 입력 샘플값을 확인할 수 있고, p[x]로 표시하며, x=0, 1, 2, 3. 4개의 MIP 입력 샘플값은 1×4의 MIP 입력 샘플 행렬을 구성할 수도 있다.
- [0139] 또한, 본 출원의 실시예에서, p[x]의 값은 통일된 계산 방법으로 획득할 수 있으며, mipSizeId의 값을 판단할 필요가 없다. 일부 실시예에서, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값은 확정하는 것은 다음 내용을 포함한다.
- [0140] 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 제 1 임시 참조값을 획득한다.
- [0141] 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정한다. 제 2 상수값을 제 1 임시 참조값 후의 데이터 유닛에 버퍼링하여 제 2 임시 참조값을 획득한다.
- [0142] MIP 입력 샘플값 중 인덱스 번호 j에 대응하는 값을 제 2 임시 참조값 중 인덱스 번호 (j+1)에 대응하는 값과 제 2 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하며, j는 0 이상의 정수이다.
- [0143] 즉, 본 출원의 실시예에서, 버퍼 pTemp의 끝에 제 2 상수값(즉, $(1 \ll (\text{BitDepth}-1))$)을 추가하여 추가 요소로 하며, 이런 경우에 현재 블록의 사이즈 파라미터를 고려할 필요가 없으며(즉, mipSizeId의 값을 고려할 필요가 없음), p[x]를 직접 $p[x]=pTemp[x+1]-pTemp[0]$ 로 설정할 수 있으며, x=0, ..., inSize - 1이다.
- [0144] 이와 같이, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정된 다음에, 진일보로 현재 블록의 MIP 예측값을 확정할 수 있다.
- [0145] S303, MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정한다.
- [0146] MIP 예측값은 현재 블록의 크로마 컴포넌트에서 부분 샘플의 예측값이다.
- [0147] 시프팅 파라미터(shifting parameter)는 시프팅 오프셋 파라미터(shifting offset parameter)와 시프팅 넘버 파라미터(shifting number parameter)를 포함한다. 이런 경우, 도 4에 도시된 바와 같이, S303은 다음 단계를 포함할 수 있다.
- [0148] S303-1, 시프팅 오프셋 파라미터의 값에 따라 MIP 입력 샘플값의 합과 시프팅 오프셋 파라미터의 곱을 확정하고, 시프팅 오프셋 파라미터의 값은 고정 상수이다.
- [0149] 시프팅 오프셋 파라미터는 시프팅 보상 파라미터 또는 오프셋 인자라고도 하며, f0로 표시할 수 있다. 본 출원의 실시예에서, 시프팅 오프셋 파라미터의 값은 32, 46, 56, 66 등과 같은 고정 상수로 설정될 수 있다. 시프팅 오프셋 파라미터의 값은 시프팅 오프셋 파라미터 테이블과 관련되도록 설정될 수 있으며, 시프팅 오프셋 파라미터의 값은 룩업 테이블을 통해 확정될 수 있으며, 이것에 대하여 한정하지 않는다.
- [0150] S303-2, 시프팅 넘버 파라미터의 값에 따라 제 1 상수값을 확정한다.
- [0151] S303-3, 제 1 오프셋의 값은 제 1 상수값과 곱의 차이로 설정된다.
- [0152] 시프팅 넘버 파라미터는 시프팅 인자, 시프팅 비트 수, 가중치 시프팅 값 등으로 불리기도 하며, sW, shift, weight shift 등으로 나타낼 수 있다. 본 출원의 실시예에서, 시프팅 넘버 파라미터는 sW로 나타낸다. 또한, 제

1 오프셋은 oW로 나타낼 수 있으며, 제 1 오프셋은 시프팅 넘버 파라미터와 시프팅 오프셋 파라미터 모두에 관련된다.

[0153] 본 출원의 실시예에서, 시프팅 넘버 파라미터의 값은 고정 상수, 예를 들어, 5, 6 또는 7 등으로 설정될 수 있다. 시프팅 넘버 파라미터의 값은 또한 시프팅 넘버 파라미터 테이블과 관련되도록 설정될 수 있으며, 시프팅 넘버 파라미터의 값은 룩업 테이블을 통해 확정될 수 있으며, 이것에 대하여 한정하지 않는다.

[0154] S303-4, 예측 파라미터에 따라 현재 블록의 MIP 가중 행렬을 확정한다.

[0155] 예측 파라미터는 예측 모드 파라미터를 포함할 수 있고, 현재 블록의 사이즈 파라미터도 포함할 수 있음에 유념해야 한다. 예측 파라미터에서, 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트는 MIP 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시하는 경우, 가중치 행렬 테이블을 미리 설정하여 메모리 또는 저장 유닛에 저장할 수 있다. 메모리 또는 저장 유닛은 인코더에 통합될 수 있거나, 별도로 설치될 수도 있다. 이와 같이, 현재 블록의 블록 사이즈 인덱스 값(mipSizeId)과 MIP 모드 인덱스 값(modeId)에 따라, 룩업 테이블을 통해 현재 블록이 사용해야 하는 MIP 가중 행렬(또는 MIP 가중치 행렬 또는 MIP 행렬이라고도 함)을 확정할 수 있으며, mWeight[x][y]로 나타낸다. 현재 블록의 블록 사이즈 인덱스 값(mipSizeId)은 현재 블록의 사이즈 파라미터에 의해 확정되며, MIP 가중 행렬 mWeight[x][y]의 크기는 단지 현재 블록의 블록 사이즈 인덱스 값에만 관련되며, 표 3에 나타난 바와 같다.

[0156] 표 3의 MIP 가중 행렬에서 열의 개수는 행렬 곱셈 입력 샘플의 개수 inSize이고, 행의 개수는 행렬 곱셈 출력 샘플의 개수 predSize×predSize이며, 따라서 현재 블록의 MIP 가중 행렬을 확정할 수 있다.

[0157] [표 3]

mipSizeId	열의 개수	행의 개수
0	4	16
1	8	16
2	7	64

[0158]

[0159] 인코더에 있어서, 시프팅 넘버 파라미터 테이블은 미리 설정되어 메모리나 저장 유닛에 저장될 수 있음을 이해할 수 있다. 여기서 메모리 또는 저장 유닛은 인코더에 통합될 수 있거나, 별도로 설치될 수도 있다. 본 출원의 실시예에서, 시프팅 넘버 파라미터(sW)는 아래 몇가지 방식으로 확정될 수 있다.

[0160] 가능한 실시예에서, 상이한 블록 사이즈 인덱스 값 및 상이한 MIP 모드 인덱스 값에 대하여, 시프팅 넘버 파라미터는 상이할 수 있다. 일부 실시예에서, 상기 방법은 또한 다음 내용을 포함할 수 있다.

[0161] 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정하는 경우, 현재 블록의 MIP 모드 인덱스 값을 확정한다.

[0162] MIP 모드 인덱스 값에 따라 제 1 사전 설정 룩업 테이블에서 MIP 모드 인덱스 값에 대응하는 값을 조회하며, 제 1 사전 설정 룩업 테이블은 MIP 모드 인덱스 값과 시프팅 넘버 파라미터의 값 사이의 대응관계를 기록하는 데에 사용된다.

[0163] 조회된 값을 시프팅 넘버 파라미터의 값으로 확정한다.

[0164] 시프팅 넘버 파라미터의 값은 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄)과 MIP 모드 인덱스 값(modeId)에 따라 조회될 수 있다. 표 4의 제 1 사전 설정 룩업 테이블에서 서로 다른 mipSizeId 및 modeId에 대하여, 행렬 곱셈에 필요한 시프팅 넘버 파라미터는 룩업 테이블을 통해 확정할 수 있다.

[0165] [표 4]

sW	modeId																	
mipSizeId	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	6	6	6	5	6	5	5	6	5	6	6	6	6	6	5	6	5	5
1	6	6	6	6	6	6	6	6	6	7								
2	7	5	6	6	6	6												

[0166]

[0167]

그러나 인코더 측에서는 표 4를 룩업 테이블 형태로 메모리나 저장 유닛에 저장해야 한다. 그러나 저장에는 대가가 필요하며 조회 과정도 마찬가지이다. 표 4의 시프팅 넘버 파라미터의 값은 현재 블록의 블록 사이즈 인덱스 값 및 MIP 모드 인덱스 값과 모두 관련이 있기 때문에, 메모리 점유가 증가되고 계산 복잡도도 증가된다.

[0168]

메모리 점유를 줄이고 계산 복잡도를 줄이기 위해, 본 출원의 실시예에서는 시프팅 넘버 파라미터를 확정하는 방법을 간소화할 수 있다.

[0169]

다른 가능한 실시예에서, 시프팅 넘버 파라미터의 값은 고정 상수로 설정될 수 있으며, 블록 사이즈 인덱스 값 및 MIP 모드 인덱스 값과 관련이 없는 고정 상수이다. 예를 들어, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 넘버 파라미터의 값은 5로 설정될 수 있다. 또는, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 넘버 파라미터의 값은 6으로 설정될 수 있다. 또는, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 넘버 파라미터의 값은 7로 설정될 수 있다. 본 출원의 실시예에서, 바람직하게는, 시프팅 넘버 파라미터의 값은 6과 동일하게 설정되며, 이것에 대하여 한정하지 않는다.

[0170]

또 다른 가능한 실시예에서, 시프팅 넘버 파라미터의 값에 대하여, 상기 방법은 아래 내용을 더 포함할 수 있다.

[0171]

현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정한다.

[0172]

현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정한다.

[0173]

현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정할 수 있다. 일부 실시예에서, 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 크기 인덱스 값을 확정하는 것은 다음 내용을 포함할 수 있다.

[0174]

현재 블록의 너비와 높이가 모두 4인 경우, 현재 블록의 블록 사이즈 인덱스 값은 0으로 설정된다.

[0175]

현재 블록의 너비와 높이가 모두 8이거나, 또는 현재 블록의 너비와 높이 중 하나가 4인 경우, 현재 블록의 블록 사이즈 인덱스 값은 1로 설정된다.

[0176]

현재 블록의 너비와 높이가 위의 조건을 만족하지 않는 경우, 현재 블록의 블록 사이즈 인덱스 값은 2로 설정된다.

[0177]

이와 같이, 현재 블록의 블록 사이즈 인덱스 값을 확정된 다음에, 진일보로 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정할 수 있다.

[0178]

선택적으로, 일부 실시예에서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정하는 것은 아래 내용을 포함할 수 있다.

[0179]

블록 사이즈 인덱스 값이 각각 0, 1 및 2인 경우, 현재 블록의 블록 사이즈 인덱스 값에 대응하는 시프팅 넘버 파라미터의 값은 각각 5, 6 및 5이다.

[0180]

선택적으로, 일부 실시예에 있어서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정하는 것은 아래 내용을 포함할 수 있다.

[0181]

시프팅 넘버 파라미터의 값을 현재 블록의 너비 또는 높이와 현재 블록의 블록 사이즈 인덱스 값에 대응하는 제 1 사전 설정값의 비율과 동일하게 설정한다.

- [0182] 여기서, 제 1 사전 설정값은 현재 블록의 경계로부터 획득된 MIP 입력 샘플값의 수량을 나타낸다. 이런 경우에 상기 방법은 다음 내용을 더 포함할 수 있다.
- [0183] 현재 블록의 블록 사이즈 인덱스 값이 각각 0, 1 및 2인 경우, 현재 블록의 블록 크기 인덱스 값에 대응하는 제 1 사전 설정값은 각각 2, 4 및 4이다.
- [0184] 즉, 제 1 사전 설정값이 현재 블록의 경계로부터 획득된 MIP 입력 샘플값의 수량을 나타낼 때, 현재 블록의 블록 사이즈 인덱스 값이 0인 경우, 대응하는 제 1 사전 설정값은 2와 같고, 현재 블록의 블록 사이즈 인덱스 값이 1인 경우, 대응하는 제 1 사전 설정값은 4와 같고, 현재 블록의 블록 사이즈 인덱스 값이 2인 경우, 대응하는 제 1 사전 설정값은 4와 같으며, 이와 같이 현재 블록의 너비 또는 높이와 대응하는 제 1 사전 설정값의 비율에 따라 시프팅 넘버 파라미터의 값을 확정할 수 있다.
- [0185] 선택적으로, 일부 실시예에서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0186] 시프팅 넘버 파라미터의 값을 현재 블록의 너비 또는 높이와 현재 블록의 블록 사이즈 인덱스 값에 대응하는 제 2 사전 설정값의 비율과 동일하게 설정한다.
- [0187] 여기서, 제 2 사전 설정값은 MIP 가중 행렬을 직접 이용하여 계산하여 획득한 현재 블록의 MIP 예측 블록의 크기를 나타낸다. 이런 경우에 상기 방법은 또한 다음 내용을 포함할 수 있다.
- [0188] 현재 블록의 블록 사이즈 인덱스 값이 각각 0, 1, 2인 경우, 현재 블록의 블록 사이즈 인덱스 값에 대응하는 제 2 사전 설정값은 각각 4, 4, 8이다.
- [0189] 즉, 제 2 사전 설정값이 MIP 가중 행렬을 직접 이용하여 계산하여 획득한 현재 블록의 MIP 예측 블록의 크기를 나타낼 때, 현재 블록의 블록 사이즈 인덱스 값이 0인 경우, 대응하는 제 2 사전 설정값은 4와 같고, 현재 블록의 블록 사이즈 인덱스 값이 1인 경우, 대응하는 제 2 사전 설정값은 4와 같고, 현재 블록의 블록 사이즈 인덱스 값이 2인 경우, 대응하는 제 2 사전 설정값은 8과 같으며, 이와 같이 현재 블록의 너비 또는 높이와 대응하는 제 2 사전 설정값의 비율에 따라 시프팅 넘버 파라미터의 값을 확정할 수 있다.
- [0190] 또 다른 가능한 실시예에서, 시프팅 넘버 파라미터 테이블은 최소화될 수 있고, 여전히 룩업 테이블에 의해 시프팅 넘버 파라미터의 값을 확정한다. 선택적으로, 일부 실시예에서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0191] 블록 사이즈 인덱스 값에 따라 제 2 사전 설정 룩업 테이블로부터 블록 사이즈 인덱스 값에 대응하는 값을 조회한다. 여기서, 제 2 사전 설정 룩업 테이블은 블록 사이즈 인덱스 값과 시프팅 넘버 파라미터의 값 사이의 대응 관계를 기록하는 데에 사용된다.
- [0192] 조회된 값은 시프팅 넘버 파라미터의 값으로 확정한다.
- [0193] 설명하여야만 하는 것은, 단지 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄)에 따라 시프팅 넘버 파라미터의 값을 조회할 수 있다. 표 5에 예시된 제 2 사전 설정 룩업 테이블에서, 각 블록 사이즈 인덱스 값은 고정된 값에 대응할 수 있다. 즉, 각 블록의 사이즈 또는 각 블록의 사이즈 집합은 표 5와 같이 고정된 시프팅 넘버 파라미터의 값을 가질 수 있다.

[표 5]

mipSizeId	sW
0	5
1	6
2	5

- [0195]
- [0196] 표 5에 따르면, 현재 블록의 블록 사이즈 인덱스 값이 각각 0, 1, 2일 때, 블록 사이즈 인덱스 값에 대응하는 시프팅 넘버 파라미터의 값은 각각 5, 6, 5이다.
- [0197] 상술한 실시예에서, 시프팅 넘버 파라미터의 확정 방법을 간소화함으로써, 특히 시프팅 넘버 파라미터 테이블을

최소화하거나, 또는 시프팅 넘버 파라미터 값을 고정함으로써, 최소화 룩업 테이블의 저장을 실현할 수 있다. 따라서, 계산 복잡도를 증가시키지 않고 MIP 모드에서 시프팅 넘버 파라미터 테이블을 저장하기 위해 점유되는 메모리를 감소시킬 수 있다.

[0198] 인코더에서 시프팅 오프셋 파라미터 테이블을 미리 설정할 수 있으며, 시프팅 오프셋 파라미터 테이블은 메모리 또는 저장 유닛에 저장될 수 있다. 여기서 메모리 또는 저장 유닛은 인코더에 통합될 수 있거나, 별도로 설치될 수도 있다. 본 출원의 실시예에서, 시프팅 오프셋 파라미터(f0)는 다음과 같은 방식으로 확정될 수 있다.

[0199] 가능한 실시예에서, 상이한 블록 사이즈 인덱스 값 및 상이한 MIP 모드 인덱스 값에 따라, 시프팅 오프셋 파라미터도 상이할 수 있다. 일부 실시예에서, 상기 방법은 또한 다음 내용을 포함할 수 있다.

[0200] 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정하는 경우, 현재 블록의 MIP 모드 인덱스 값이 확정된다.

[0201] MIP 모드 인덱스 값에 따라, 제 3 사전 설정 룩업 테이블로부터 MIP 모드 인덱스 값에 대응하는 값을 조회하고, 제 3 사전 설정 룩업 테이블은 MIP 모드 인덱스 값과 시프팅 오프셋 파라미터의 값 사이의 대응관계를 기록하는 데에 사용된다.

[0202] 조회된 값은 시프팅 오프셋 파라미터의 값으로 확정된다.

[0203] 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄) 및 MIP 모드 인덱스 값(modeId)에 따라 시프팅 오프셋 파라미터의 값을 조회할 수 있음에 유의해야 한다. 표 6에 예시된 제 3 사전 설정 룩업 테이블과 같이, 상이한 mipSizeId 및 modeId에 대하여, 행렬 곱셈에 필요한 시프팅 오프셋 파라미터는 룩업 테이블을 통해 확정할 수 있다.

[0204] [표 6]

f0	modeId																	
mipSizeId	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	34	19	7	32	27	24	21	13	24	15	27	20	16	7	20	23	21	24
1	17	20	11	21	17	11	23	10	21	11								
2	8	46	16	10	13	11												

[0205] 그러나 인코더 측에서는 표 6을 룩업 테이블 형태로 메모리나 저장 유닛에 저장해야 한다. 그러나 저장에는 대가가 필요하며 조회 과정도 마찬가지로 대가가 필요하다. 표 6의 시프팅 오프셋 파라미터의 값은 현재 블록의 블록 사이즈 및 MIP 모드 인덱스 값과 모두 관련이 있기 때문에, 메모리 점유가 증가되고 계산 복잡도도 증가된다.

[0207] 메모리 점유를 줄이고 계산 복잡도를 줄이기 위해, 본 출원의 실시예에서는 시프팅 오프셋 파라미터를 확정하는 방법을 간소화할 수 있다.

[0208] 다른 가능한 실시예에서, 시프팅 오프셋 파라미터의 값은 고정 상수로 설정될 수 있으며, 블록 사이즈 인덱스 값 및 MIP 모드 인덱스 값과 관련이 없는 고정 상수이다. 일반적으로, 고정 상수의 범위는 0~100이다. 예를 들어, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 오프셋 파라미터의 값은 32로 설정될 수 있다. 또는, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 오프셋 파라미터의 값은 46으로 설정될 수 있다. 또는, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 오프셋 파라미터의 값은 56으로 설정될 수 있다. 또는, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 오프셋 파라미터의 값은 66으로 설정될 수 있다. 본 출원의 실시예에서, 바람직하게는, 시프팅 오프셋 파라미터의 값은 32와 동일하게 설정되며, 이것에 대하여 한정하지 않는다.

[0209] 또 다른 가능한 실시예에서, 시프팅 오프셋 파라미터의 값에 대하여, 상기 방법은 아래 내용을 더 포함할 수 있다.

- [0210] 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정한다.
- [0211] 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 오프셋 파라미터의 값을 확정한다.
- [0212] 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정할 수 있다. 그 다음에 현재 블록의 블록 사이즈 인덱스 값에 따라 진일보로 시프팅 오프셋 파라미터의 값을 확정할 수 있다.
- [0213] 선택적으로, 일부 실시예에서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 오프셋 파라미터의 값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0214] 블록 사이즈 인덱스 값이 각각 0, 1 및 2인 경우, 현재 블록의 블록 사이즈 인덱스 값에 대응하는 시프팅 오프셋 파라미터의 값은 각각 34, 23 및 46이다.
- [0215] 선택적으로, 일부 실시예에 있어서, 시프팅 오프셋 파라미터 테이블은 최소화될 수 있고, 여전히 룩업 테이블에 의해 시프팅 인자를 확정한다. 선택적으로, 일부 실시예에서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 오프셋 파라미터의 값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0216] 블록 사이즈 인덱스 값에 따라 제 4 사전 설정 룩업 테이블로부터 블록 사이즈 인덱스 값에 대응하는 값을 조회한다. 여기서, 제 4 사전 설정 룩업 테이블은 블록 사이즈 인덱스 값과 시프팅 오프셋 파라미터의 값 사이의 대응 관계를 기록하는 데에 사용된다.
- [0217] 조회된 값은 시프팅 오프셋 파라미터의 값으로 확정한다.
- [0218] 설명하여야만 하는 것은, 단지 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄)에 따라 시프팅 오프셋 파라미터의 값을 조회할 수 있다. 표 7에 예시된 제 4 사전 설정 룩업 테이블과 같이, 각 블록 사이즈 인덱스 값은 고정된 값에 대응할 수 있다. 즉, 각 블록의 사이즈 또는 각 블록의 사이즈 집합은 표 7에 예시된 것과 같이 고정된 시프팅 오프셋 파라미터의 값을 가질 수 있다.
- [0219] [표 7]

mipSizeId	f0
0	34
1	23
2	46

- [0220]
- [0221] 표 7에 따르면, 현재 블록의 블록 사이즈 인덱스 값이 각각 0, 1, 2일 때, 블록 사이즈 인덱스 값에 대응하는 시프팅 오프셋 파라미터의 값은 각각 34, 23, 46이다.
- [0222] 상술한 실시예에서, 시프팅 오프셋 파라미터의 확정 방법을 간소화함으로써, 특히 시프팅 오프셋 파라미터 테이블을 최소화하거나, 또는 시프팅 오프셋 파라미터 값을 고정함으로써, 최소화 룩업 테이블의 저장을 실현할 수 있다. 따라서, 계산 복잡도를 증가시키지 않고 MIP 모드에서 시프팅 오프셋 파라미터 테이블을 저장하기 위해 점유되는 메모리를 감소시킬 수 있다.
- [0223] 이와 같이, 시프팅 오프셋 파라미터(f0) 및 시프팅 넘버 파라미터(sW)를 확정한 다음에, 제 1 상수값에 대하여, 선택적으로, 일부 실시예에서, 시프팅 넘버 파라미터의 값에 따라 제 1 상수값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0224] 제 1 상수값을 2의 정수 거듭제곱과 동일하게 설정하며, 거듭제곱의 지수는 시프팅 넘버 파라미터의 값에서 1을 뺀 값과 같다.
- [0225] 선택적으로, 일부 실시예에서, 시프팅 넘버 파라미터의 값에 따라 제 1 상수값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0226] "1"에 대하여 이진 왼쪽 시프팅을 수행하여 제 1 상수값을 획득하고, 왼쪽으로 시프트된 비트의 수는 시프팅 넘버 파라미터의 값에서 1을 뺀 것과 같다.
- [0227] 즉, 시프팅 넘버 파라미터(sW)를 획득한 다음에, 제 1 상수값은 $1 \ll (sW-1)$ 또는 $2^{(sW-1)}$ 로 표시될 수 있다. 시

프팅 넘버 파라미터의 값이 6으로 설정될 경우, 제 1 상수값은 32와 같다.

[0228] 여기서, 제 1 오프셋을 oW 로 나타내면, 시프팅 오프셋 파라미터(fO)에 따라 MIP 입력 샘플값의 합과 시프팅 오프셋 파라미터의 곱을 계산하여 획득할 수 있으며, 즉, $fO \times (\sum_{i=0}^{inSize-1} p[i])$ 이다. oW 의 값은 $(1 \ll (sW - 1)) - fO \times (\sum_{i=0}^{inSize-1} p[i])$ 로 설정될 수 있다.

[0229] 이와 같이, MIP 모드에서 MIP 가중 행렬, MIP 입력 샘플값, 시프팅 넘버 파라미터, 제 1 오프셋을 획득할 수 있으며, 이후에 현재 블록의 MIP 예측값을 확정하는 것이 편리해지도록 한다.

[0230] S303-5, MIP 가중 행렬, MIP 입력 샘플값, 시프팅 넘버 파라미터 및 제 1 오프셋에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정한다.

[0231] 여기서, MIP 예측값은 현재 블록의 크로마 컴포넌트에서 부분 샘플의 예측값을 나타낸다.

[0232] MIP 가중 행렬, MIP 입력 샘플값, 시프팅 넘버 파라미터 및 제 1 오프셋에 따라 현재 블록의 MIP 예측값을 확정하는 것은 다음 내용을 포함할 수 있다.

[0233] MIP 가중 행렬과 MIP 입력 샘플값의 제 1 가중치 합을 계산한다.

[0234] 제 1 가중치 합과 제 1 오프셋의 제 1 합을 계산한다.

[0235] 제 1 합에 대하여 이진 오른쪽 시프팅(binary right-shifting)을 수행하여 제 1 오른쪽 시프트 값(first right-shift value)을 획득하며, 오른쪽으로 시프트된 비트의 수는 시프팅 넘버 파라미터의 값과 같다.

[0236] 현재 블록의 MIP 예측값을 제 1 오른쪽 시프트 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 합과 동일하게 설정하며, 제 1 임시 참조값은 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 획득한다.

[0237] 구체적으로, MIP 모드에서, 현재 블록의 블록 사이즈 인덱스 값($mipSizeId$ 로 나타냄)와 MIP 모드 인덱스 값($modeId$ 로 나타냄)에 따라 MIP 가중 행렬($mWeight$ 로 나타냄), 시프팅 넘버 파라미터(sW 로 나타냄) 및 시프팅 오프셋 파라미터(fO 로 나타냄)를 확인할 수 있다. 그 다음에 MIP 입력 샘플값($p[x]$ 로 나타냄), $mWeight$, sW 및 fO 를 행렬 곱셈에 입력하여 행렬 곱셈에서 출력되는 MIP 예측값($predMip[x][y]$ 로 나타낼 수 있음)을 획득하며, $predMip[x][y]$ 의 샘플을 $predSize \times predSize$ 에 따라 행렬/어레이 형태로 배열한다. $predMip[x][y]$ 의 계산식은 다음과 같다.

$$\begin{cases} predMip[x][y] = (((\sum_{i=0}^{inSize-1} mWeight[i][y \times predSize + x] \times p[i]) + \\ oW) \gg sW) + pTemp[0] \\ oW = (1 \ll (sW - 1)) - fO \times (\sum_{i=0}^{inSize-1} p[i]) \end{cases} \quad (5)$$

[0238] $[x][y]$ 는 픽셀의 위치 좌표를 나타내고, x 는 가로 방향을 나타내고, y 는 세로 방향을 나타내며, $inSize$ 는 입력 샘플의 개수를 나타내고, $p[i]$ 는 MIP 입력 샘플값 중 인덱스 번호 i 에 대응하는 값을 나타낸다. $pTemp[0]$ 은 제 1 임시 참조값에서 인덱스 번호 0에 대응하는 값을 나타낸다. " $\gg$ "는 이진 오른쪽 시프팅 연산자(binary right-shifting operator)를 나타내고, " $\ll$ "는 이진 왼쪽 시프팅 연산자(binary left-shifting operator)를 나타낸다. $x=0, \dots, predSize-1, y=0, \dots, predSize-1$. 이와 같이, 위의 계산식(5)에 따라 $predMip[x][y]$ 를 계산하여 MIP 예측 블록을 획득할 수 있다.

[0240] 일부 실시예에서, sW 의 고정값이 6이고 fO 의 고정값이 32인 경우, 계산식은 다음과 같다.

$$\begin{cases} predMip[x][y] = (((\sum_{i=0}^{inSize-1} mWeight[i][y \times predSize + x] \times p[i]) + \\ oW) \gg 6) + pTemp[0] \\ oW = 32 - 32 \times (\sum_{i=0}^{inSize-1} p[i]) \end{cases} \quad (6)$$

[0241] 또한, 현재 블록의 MIP 예측 블록을 얻기 위하여 MIP 예측 블록 중 샘플의 예측값에 대하여 클리핑(clipping)을 수행할 필요가 있다. 그 다음에 MIP 예측 블록에 대하여 전치할지 여부를 판단한다. 판단 결과가 "예"이면, MIP

예측 블록 중의 예측 샘플값을 전치할 필요가 있으며, 전치된 MIP 예측 블록을 현재 블록의 MIP 예측 블록으로 확정한다. 판단 결과가 "아니오"인 경우, MIP 예측 블록 중의 예측 샘플값을 전치할 필요가 없으며, 직접 이 MIP 예측 블록을 현재 블록의 MIP 예측 블록으로 확정하여 현재 블록의 MIP 예측값을 획득할 수 있다. MIP 예측값은 현재 블록 중 일부 샘플의 예측값이다.

[0243] S304, MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정한다.

[0244] MIP 예측값을 필터링하여 현재 블록의 인트라 예측값을 확정하는 것은, 다음 내용을 포함할 수 있다.

[0245] MIP 예측 블록의 사이즈 파라미터와 현재 블록의 사이즈 파라미터가 동일한지 여부를 판단한다.

[0246] MIP 예측 블록의 사이즈 파라미터와 현재 블록의 사이즈 파라미터가 동일한 경우, 현재 블록의 인트라 예측 블록을 MIP 예측 블록과 동일하게 설정하며, MIP 예측 블록은 현재 블록 중 모든 픽셀 위치의 예측 샘플값을 포함한다.

[0247] MIP 예측 블록의 사이즈 파라미터와 현재 블록의 사이즈 파라미터가 다른 경우, MIP 예측 블록을 필터링하여 필터링된 예측 블록을 획득하고, 필터링된 예측 블록을 현재 블록의 인트라 예측 블록으로 설정한다.

[0248] 여기서 필터링은 업 샘플링 필터링 또는 저역통과 필터링을 포함할 수 있다.

[0249] MIP 예측 블록은 MIP 예측값으로 구성된다는 점에 유념해야 한다. MIP 예측 블록을 획득한 다음에, MIP 예측 블록의 사이즈 파라미터는 4x4 MIP 예측 블록과 8x8 MIP 예측 블록의 두가지 유형만 포함하기 때문에, 현재 블록의 사이즈 파라미터와 MIP 예측 블록의 사이즈 파라미터는 동일하거나 동일하지 않을 수 있다. 즉, MIP 예측 블록에 대응하는 샘플값은 현재 블록을 완전히 채우지 못할 수 있으므로, 최종 예측값을 생성하기 위해서는 MIP 예측 블록에 대한 업 샘플링이 필요하다. 즉, MIP 예측 블록의 사이즈 파라미터와 현재 블록의 사이즈 파라미터가 동일한지 여부를 판단하여 MIP 예측 블록을 업 샘플링할지 여부를 확정한다.

[0250] 또한, MIP 예측 블록의 사이즈 파라미터와 현재 블록의 사이즈 파라미터가 동일한 경우, 즉 MIP 예측 블록의 너비 및 높이가 모두 현재 블록과 동일한 경우, MIP 예측 블록에 대하여 업 샘플링을 수행할 필요가 없음을 나타낸다. 이런 경우, 직접 MIP 예측 블록을 현재 블록에 채울 수 있다. 즉 채운 후의 현재 블록에 빈 픽셀이 없다. 이런 경우, 현재 블록의 각 픽셀의 인트라 예측값은 MIP 예측 블록의 각 픽셀의 예측값으로 직접 대체될 수 있다. 다음과 같다.

$$\text{predSamples}[x][y] = \text{predMip}[x][y] \quad (7)$$

[0251] [x][y]는 픽셀의 위치 좌표를 나타내고, x는 가로 방향을 나타내고, y는 세로 방향을 나타낸다. predSamples[x][y]는 현재 블록 중 위치 좌표 [x][y]의 픽셀에 대응하는 인트라 예측값을 나타내고, predMip[x][y]는 MIP 예측 블록 중 위치 좌표 [x][y]의 픽셀에 대응하는 예측값을 나타낸다. 이와 같이, 식 (7)에 따르면, MIP 예측 블록 predMip[x][y]를 직접 현재 블록의 인트라 예측 블록 predSamples[x][y]로 할 수 있으며, 즉 직접 현재 블록 내의 적어도 하나의 픽셀의 인트라 예측값으로 할 수 있다.

[0253] 또한, MIP 예측 블록의 사이즈 파라미터와 현재 블록의 사이즈 파라미터가 다른 경우, 즉 MIP 예측 블록의 너비 및 높이 중 적어도 하나가 현재 블록과 다른 경우, MIP 예측 블록은 현재 블록을 완전히 채울 수 없다. 즉, 채운 후의 현재 블록에 빈 픽셀이 있으며, 이런 경우에 MIP 예측 블록에 대하여 필터링을 수행할 필요가 있음을 나타낸다. 즉, 수평 방향과 수직 방향 모두에서 업 샘플링이 필요한 경우, MIP 예측 블록에 대하여 수평 방향 업 샘플링을 수행한 다음에 수직 방향 업 샘플링을 수행하여 predSamples[x][y]로 나타낼 수 있는 제 1 업 샘플링 블록을 얻을 수 있다. 그 다음에 MIP 예측 블록에 대하여 수직 방향 업 샘플링을 수행한 다음에 수평 방향 업 샘플링을 수행하여 predSamplesTemp[x][y]로 나타낼 수 있는 제 2 업 샘플링 블록을 얻는다. 마지막으로 predSamples[x][y]와 predSamplesTemp[x][y]에 대하여 가중 평균 계산을 수행하여, 최종적으로 현재 블록의 크로마 컴포넌트의 인트라 예측값, 즉 현재 블록의 인트라 예측 블록을 획득한다.

[0254] 예시적으로, 현재 블록의 변의 길이 nTbS(여기서, S는 각각 W 및 H로 대체될 수 있음)와 predMip의 변의 길이 predSize(여기서, predSize는 현재 블록의 blockSizeIdx에만 관련됨)가 동일한 경우, 직접 MIP 예측 블록을 현재 블록의 인트라 예측 블록으로 설정할 수 있다. 그렇지 않으면 현재 블록의 인트라 예측 블록을 얻기 위해 MIP 예측 블록에 대하여 필터링을 수행해야 한다. 여기서, 4x4 현재 블록을 예로 들면, 현재 블록의 사이즈 파라미터와 MIP 예측 블록의 사이즈 파라미터가 동일한 경우, MIP 예측 블록에 대하여 필터링을 수행할 필요가 없다. 따라서 직접 MIP 예측 블록을 현재 블록의 인트라 예측 블록으로 설정할 수 있으며, 현재 블록의 적어도 하

나의 픽셀의 인트라 예측값을 획득할 수 있다.

- [0255] 간단히 말하면, MIP 모드의 예측 과정에 대하여, MIP 모드의 입력 데이터는 현재 블록의 위치(xTbCmp, yTbCmp), 현재 블록에 적용된 MIP 모드 인덱스 값(modeId로 나타낼 수 있음), 현재 블록의 높이(nTbH로 나타낼 수 있음), 현재 블록의 너비(nTbW로 나타낼 수 있음), 전치가 필요한지 여부를 나타내는 전치 지시 식별자(즉, MIP 전치 지시 파라미터, isTransposed로 나타낼 수 있음) 등을 포함할 수 있다. MIP 모드의 출력 데이터는 현재 블록의 인트라 예측 블록을 포함할 수 있다. 인트라 예측 블록의 픽셀 좌표 [x][y]에 대응하는 예측값은 predSamples[x][y]이고, 여기서 x=0, 1, ..., nTbW-1이고; y=0, 1, ..., nTbH-1이다.
- [0256] 도 5에 도시된 바와 같이, MIP 예측 과정은 구체적으로 다음 단계를 포함한다.
- [0257] S501, 핵심 파라미터를 구성한다.
- [0258] S501에 대하여, MIP 코어 파라미터는 현재 블록의 유형(mipSizeId로 나타냄), 각 변의 참조 샘플 개수(boundySize로 나타냄), MIP 입력 샘플 개수(inSize로 나타냄) 및 행렬 곱셈에서 출력되는 MIP 예측 블록의 크기(predSize × predSize로 배열됨)를 포함할 수 있다. 현재 블록의 크기에 따라 현재 블록을 세가지 유형으로 분류할 수 있으며, 여기서 mipSizeId로 현재 블록의 유형을 나타낸다. 여기서, mipSizeId는 0, 1 또는 2와 같을 수 있다. 또한, 상이한 유형의 현재 블록에 대하여, 참조 샘플의 개수 및 행렬 곱셈에서 출력되는 MIP 예측 블록의 크기는 다르다.
- [0259] S502, 참조 샘플을 획득한다.
- [0260] S502에 대하여, 현재 블록을 예측할 때, 현재 블록의 위쪽 인접 블록 및 왼쪽 인접 블록은 모두 이미 인코딩된 블록이고, MIP 모드의 참조 픽셀은 현재 블록과 인접된 위쪽 인접 행의 픽셀과 왼쪽 인접 열의 픽셀의 재구성된 값이다. 다시 말하면, 현재 블록의 상측변의 인접 참조 픽셀(refT로 나타냄)과 좌측변의 인접 참조 픽셀(refL로 나타냄)을 획득하는 과정이 바로 참조 픽셀을 획득하는 과정이다.
- [0261] S503, 입력 샘플을 구성한다.
- [0262] S503에 대하여, 이 단계는 행렬 곱셈의 입력에 사용되며, 구체적으로 다음 내용을 포함할 수 있다.
- [0263] S5031, 참조 샘플을 획득한다.
- [0264] S5032, 참조 샘플 버퍼를 구성한다.
- [0265] S5033, 행렬 곱셈 입력 샘플을 유도한다.
- [0266] S5031에 대하여, 참조 샘플을 획득하는 과정은 참조 픽셀에 대하여 다운 샘플링을 수행하는 과정이다. S5032에 대하여, 참조 샘플 버퍼를 구성하는 데에 두가지 채우기 방법이 있으며, 구체적으로 다음 내용을 포함할 수 있다.
- [0267] S5032-1, 전치가 필요하지 않을 때에 버퍼를 채우는 방법.
- [0268] S5032-2, 전치가 필요할 때에 버퍼를 채우는 방법.
- [0269] S504, 인트라 예측값을 생성한다.
- [0270] S504에 대하여, 이 단계는 현재 블록의 MIP 예측값을 획득하는 데에 사용되며, 구체적으로 다음 내용을 포함할 수 있다.
- [0271] S5041, 행렬 곱셈 출력 샘플의 MIP 예측 블록을 구성한다.
- [0272] S5042, 행렬 곱셈 출력 샘플의 MIP 예측 블록에 대하여 클리핑(clipping)을 수행한다.
- [0273] S5043, 행렬 곱셈의 출력 샘플의 MIP 예측 블록에 대하여 전치 처리(transposition)를 수행한다.
- [0274] S5044, MIP 예측값을 생성한다.
- [0275] S5041에 대하여, MIP 예측 블록을 구성하는 과정에서 구체적으로 다음 내용을 포함할 수 있다.
- [0276] S5041-1, 가중 행렬을 획득한다.
- [0277] S5041-2, 시프팅 파라미터를 획득한다.

- [0278] S5041-3, 행렬 곱셈 연산을 수행한다.
- [0279] 즉, MIP 예측 블록을 구성하는 과정에서 가중 행렬뿐만 아니라 시프팅 파라미터(시프팅 오프셋 파라미터 및 시프팅 넘버 파라미터를 포함함)도 획득해야 한다. 그 다음에 행렬 곱셈 출력 샘플의 MIP 예측 블록을 얻기 위해 행렬 곱셈 연산을 수행한다. S5044에 대하여, 최종 MIP 예측값을 생성하는 것은 두가지 경우를 포함하며, 구체적으로 다음 내용을 포함할 수 있다.
- [0280] S5044-1, 업 샘플링이 필요하지 않는 예측값을 생성한다.
- [0281] S5044-2, 업 샘플링이 필요한 예측값을 생성한다.
- [0282] 이와 같이, 본 출원의 실시예에서는 상술한 S501~S504의 네개의 단계를 거친 후에 현재 블록의 크로마 컴포넌트 중 적어도 하나의 픽셀의 인트라 예측값을 얻을 수 있다.
- [0283] S305, 현재 블록의 크로마 컴포넌트의 인트라 예측값에 따라 현재 블록의 크로마 컴포넌트의 예측 잔차값을 확정한다.
- [0284] 현재 블록의 크로마 컴포넌트의 인트라 예측값을 획득한 다음에, 현재 블록의 크로마 컴포넌트의 실제값과 현재 블록의 크로마 컴포넌트의 인트라 예측값의 차이를 계산할 수 있으며, 따라서 현재 블록의 크로마 컴포넌트의 예측 잔차값, 즉 현재 블록의 잔차 블록을 획득할 수 있다.
- [0285] S306, 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정한다.
- [0286] 도 1에 도시된 변환 유닛(108)에 대하여, 변환 유닛(108)은 현재 블록의 잔차 블록에 대하여 제 1차 변환, 예를 들어, DCT(discrete cosine transform)초기 설계를 기반으로 하는 정수 변환을 수행할 수 있다. 변환 유닛(108)은 또한 현재 블록의 잔차 블록에 대하여 LFNST와 같은 제 2차 변환을 수행할 수 있다. 구체적으로, 일부 실시예에서, 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정하는 것은 다음 내용을 포함할 수 있다.
- [0287] 예측 잔차값에 따라 현재 블록의 잔차 블록을 확정하고, 잔차 블록에 대하여 제 1차 변환을 수행하여 제 1 변환 계수 블록을 획득한다.
- [0288] 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정한다.
- [0289] 확정 결과가 예인 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 파라미터를 확정한다.
- [0290] 본 출원의 실시예에서, 인코더는 잔차 블록의 예측 잔차값에 대하여 제 1차 변환("코어 변환" 또는 "1차 변환" 또는 "메인 변환"이라고도 함)을 수행하여 제 1차 변환 후의 제 1 변환 계수 블록을 획득한다. 그 다음에 제 1 변환 계수 블록 중의 부분 또는 모든 변환 계수에 대하여 LFNST 변환("2차 변환" 또는 "제 2 변환"이라고도 함)을 수행하여 LFNST 파라미터를 확정한다. 여기서 제 1차 변환은 LFNST와 다른 변환이라는 점에 유념해야 한다.
- [0291] 또한, 임의의 현재 블록이 모두 LFNST를 수행할 수 있는 것이 아니라는 점에 유념해야 한다. 제 1 변환 계수 블록 중 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정하기 위하여 제 1차 변환 후에 일부 조건(현재 블록의 사이즈 파라미터 중의 최소값, 현재 블록의 블록 사이즈 인덱스 값 등)을 확인해야 한다. 선택적으로, 일부 실시예에서, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정하는 것은 다음 내용을 포함할 수 있다.
- [0292] 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정하는 경우, 현재 블록의 사이즈 파라미터 중의 최소값을 확정한다.
- [0293] 최소값에 따라 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정한다.
- [0294] 또한, 최소값에 따라 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정하는 것은 다음 내용을 포함할 수 있다.
- [0295] 최소값이 제 1 사전 설정 임계값 이상인 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정한다.

- [0296] 여기서, 제 1 사전 설정 임계값은 LFNST 수행 여부를 평가하기 위한 미리 설정된 임계값을 나타내는 데에 사용될 수 있다. 본 출원의 실시예에서, 제 1 사전 설정된 임계값은 8로 설정될 수 있으며, 이것에 대하여 구체적으로 제한하지 않는다.
- [0297] 즉, 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측을 수행하는 경우, 현재 블록의 사이즈 파라미터 중의 최소값을 확인할 수 있다. 예를 들어, 너비 및 높이의 최소값이 8 이상인 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정한다.
- [0298] 선택적으로, 일부 실시예에서, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 결정하는 것은 다음 내용을 포함할 수 있다.
- [0299] 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 결정하는 경우, 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정한다.
- [0300] 블록 사이즈 인덱스 값에 따라 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정한다.
- [0301] 또한, 블록 사이즈 인덱스 값에 따라 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 결정하는 것은 다음 내용을 포함할 수 있다.
- [0302] 블록 사이즈 인덱스 값이 제 2 사전 설정 임계값과 동일한 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정한다.
- [0303] 여기서, 제 2 사전 설정 임계값도 LFNST 수행 여부를 평가하기 위한 미리 설정된 임계값을 나타내는 데에 사용될 수 있다. 본 출원의 실시예에서, 제 2 사전 설정 임계값은 사전 설정값(예를 들어, 2) 또는 다수의 사전 설정값(예를 들어, 1, 2) 중 하나로 설정될 수 있으며, 구체적으로 한정하지 않는다.
- [0304] 즉, 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측을 수행하는 경우, 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값(mipSizeId)을 확인할 수 있다. 예를 들어, mipSizeId의 값이 2와 같을 때, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정할 수 있다.
- [0305] 선택적으로, 일부 실시예에서, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 결정하는 것은 다음 내용을 포함할 수 있다.
- [0306] 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하지 않는 경우에 제 1 대가 결과를 계산한다.
- [0307] 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하는 경우에 제 2 대가 결과를 계산한다.
- [0308] 제 1 대가 결과 및 제 2 대가 결과에 따라, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정한다.
- [0309] 제 1 대가 결과 및 제 2 대가 결과에 따라, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 결정하는 것은 다음 내용을 포함할 수 있다.
- [0310] 제 1 대가 결과가 제 2 대가 결과보다 큰 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정한다.
- [0311] 본 출원의 실시예에서 전통적인 레이트 왜곡 최적화 방법이 채택될 수 있음을 유념해야 한다. 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하지 않는 경우에 제 1 대가 결과를 계산한다. 그 다음, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하는 경우에 제 2 대가 결과를 계산한다. 제 1 대가 결과와 제 2 대가 결과를 비교하여, 제 1 대가 결과가 제 2 대가 결과보다 큰 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정한다.
- [0312] 이와 같이 확정 결과가 예인 경우, 즉 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할 필요가 있다고 확정한 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 파라미터를 확정할 수 있다.

- [0313] S307, LFNST 파라미터를 인코딩하고 비트스트림으로 기입한다.
- [0314] LFNST 파라미터는 `lfnst_index`로 나타낼 수 있는 LFNST 인덱스 번호를 포함한다. 일부 실시예에서, LFNST 파라미터를 확정하는 것은 다음 내용을 포함할 수 있다.
- [0315] 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정된 경우, LFNST 인덱스 번호의 값은 0보다 크게 설정된다.
- [0316] 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하지 않기로 확정된 경우, LFNST 인덱스 번호의 값은 0으로 설정된다.
- [0317] 또한, S307에 대하여, LFNST 파라미터를 인코딩하고 비트스트림으로 기입하는 것은 아래 내용을 포함할 수 있다.
- [0318] LFNST 인덱스 번호를 인코딩하고 비트스트림으로 기입한다.
- [0319] 즉, LFNST 파라미터, 즉 `lfnst_index`를 확정된 다음에, `lfnst_index`가 0이면 LFNST를 실행하지 않음을 의미한다. `lfnst_index`가 0보다 크면 LFNST를 실행할 필요가 있음을 의미한다. 인코딩과정에서, 서술자가 "ae(v)"인 CABAC(context-based adaptive binary arithmetic coding)을 사용하여 `lfnst_index`를 인코딩할 수 있다.
- [0320] 예시적으로, 도 8을 참조하면, 종래 기술에서 제공되는 LFNST 파라미터의 선택스 구조에 대한 설명의 일례를 나타낸다. 이런 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정하는 판단 조건은 다음 중 적어도 하나를 포함할 수 있다: `treeType == DUAL_TREE_CHROMA`, 또는 `!IntraMipFlag[ x0 ][ y0 ]`, 또는 `Min( lfnstWidth, lfnstHeight ) >= 16`.

[표 8]

<pre> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_ flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && lfnstNotTs Flag == 1 && (treeType== DUAL_TREE_CHROMA !IntraMipFlag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>	
<pre> if((IntraSubPartitionsSplitType!= ISP_NO_SPLIT LfnstDcO nly == 0) && LfnstZeroOutSigCoeffFlag == 1) </pre>	
<code>lfnst_idx</code>	<code>ae(v)</code>
<pre> } </pre>	

- [0322]
- [0323] 표 9A 및 표 9B를 참조하면, 본 출원의 실시예에서 제공되는 LFNST 파라미터의 선택스 구조에 대한 설명의 일례를 나타낸다. 이런 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정하는 판단 조건은 다음 중 적어도 하나를 포함할 수 있다: `intra_mip_flag[x0][y0] == 1 && Min(lfnstWidth, lfnstHeight) >= 8`, 또는 `intra_mip_flag[x0][y0] == 1 && mipSizeId == 2`.
- [0324] 구체적으로, 표 9A의 선택스 구조에 나타난 바와 같이, 현재 블록의 사이즈 파라미터의 최소값이 8이상(즉, 너비와 높이가 모두 8이상)인 경우, MIP 모드를 사용하여 현재 블록의 `lfnst_index`에 대하여 인코딩할 수 있다(즉, `intra_mip_flag[x0][y0]==1`인 경우). 또는 상기 판단 조건은 너비와 높이가 모두 제 1값(예를 들면, 8)이상이고, 또한 너비와 높이의 합이 제 2값(예를 들면, 16)이상인 것으로 설정할 수도 있다. 표 9B의 선택스 구조에 나타난 바와 같이, 현재 블록의 블록 사이즈 인덱스 값(`mipSizeId`)이 2인 경우, MIP 모드를 사용하여 현재 블록의 `lfnst_index`를 인코딩할 수 있다(즉, `intra_mip_flag[x0][y0]==1`인 경우). 또는, 상기 판단 조건은 `mipSizeId`가 복수의 사전 설정값(예를 들면, 1 및 2) 중 하나와 같도록 설정할 수도 있다.

[0325] [표 9A]

<code>coding_unit(x0,y0,cbWidth,cbHeight,cqtDepth,treeType, modeType){</code>	Descriptor
.....	
<code>if(...&&(intra_mip_flag[x0][y0]==1&&Min(lfstWidth, lfstHeight)>=8)&&...){</code>	
.....	
<code>lfst_idx</code>	<code>ae(v)</code>
<code>}</code>	
.....	
<code>}</code>	

[0326]

[0327] [표 9B]

<code>coding_unit(x0,y0,cbWidth,cbHeight,cqtDepth,treeType, modeType){</code>	Descriptor
.....	
<code>if(...&&(intra_mip_flag[x0][y0]==1&&mipSizeId==2)&&...){</code>	
.....	
<code>lfst_idx</code>	<code>ae(v)</code>
<code>}</code>	
.....	
<code>}</code>	

[0328]

[0329] 또한, 현재 블록에 대하여 LFNST가 수행될 수 있다고 판단되면, 현재 블록에 사용되는 LFNST 변환 커널(kernel로 나타냄)을 확정해야 한다는 점에 유념해야 한다. 일부 실시예에서, 예측 잔차값에 대하여 LFNST를 수행하여 LFNST 파라미터를 확정하는 것은, 현재 블록에 사용되는 LFNST 변환 커널을 확정하는 것과, LFNST 변환 커널을 이용하여 예측 잔차값에 대하여 LFNST를 수행하여 LFNST 파라미터를 확정하는 것을 포함한다.

[0330] 본 출원의 실시예에서, LFNST에는 4개의 변환 커널 후보 세트가 있고, 4개의 변환 커널 후보 세트는 set0, set1, set2 및 set3을 포함할 수 있다. 여기서, 본 출원의 실시예에서, MIP 파라미터를 이용하여 LFNST 변환 커널 후보 세트를 확정할 수 있고, 그 다음에 LFNST 변환 커널 후보 세트로부터 현재 블록에 사용되는 LFNST 변환 커널을 선택한다. 따라서, 일부 실시예에서, 상기 방법은 다음 내용을 더 포함할 수 있다.

[0331] 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정하는 경우, 현재 블록의 예측 파라미터로부터 MIP 파라미터를 확정한다.

[0332] MIP 파라미터에 따라 LFNST 변환 커널 후보 세트를 확정한다.

[0333] LFNST 변환 커널 후보 세트에서 LFNST 인덱스 번호가 지시하는 변환 커널을 현재 블록에 사용되는 LFNST 변환 커널로 설정한다.

[0334] LFNST 인덱스 번호의 값이 현재 블록에 대하여 LFNST를 수행하고 또한 LFNST 변환 커널 후보 세트에 있어서의 LFNST 변환 커널의 인덱스 번호를 지시하도록 LFNST 인덱스 번호의 값을 설정한다.

[0335] LFNST 변환 커널 후보 세트는 미리 설정된 두개 이상의 LFNST 변환 커널을 포함한다.

[0336] MIP 파라미터는 MIP 모드 인덱스 값(modeId로 나타냄) 및 MIP 전치 지시 파라미터(isTransposed로 나타냄) 중 적어도 하나를 포함할 수 있음을 유념해야 한다. 현재 블록에 사용되는 LFNST 변환 커널에 대하여, MIP 전치 지시 파라미터의 값이 MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치가 수행됨을 나타낼 때, 현재 블록에 사용되는 LFNST 변환 커널을 획득하기 위해 선택한 변환 커널에 대하여 행렬 전치를 수행할 필요가 있다.

[0337] 또한, 일부 실시예에서, MIP 파라미터에 따라 LFNST 변환 커널 후보 세트를 확정하는 것은 다음 내용을 포함할

수 있다.

[0338] MIP 모드 인덱스 값에 따라 LFNST 인트라 예측 모드 인덱스 번호의 값을 확정한다.

[0339] LFNST 인트라 예측 모드 인덱스 번호의 값을 기반으로, 제 5 사전 설정 룩업 테이블을 통해 LFNST 변환 커널 후보 세트의 인덱스 값을 확정한다.

[0340] LFNST 변환 커널 후보 세트의 인덱스 값에 따라 복수의 후보 LFNST 변환 커널 후보 세트로부터 상기 LFNST 변환 커널 후보 세트를 선택한다.

[0341] 여기서, LFNST 인트라 예측 모드 인덱스 번호는 predModeIntra로 나타낼 수 있고, LFNST 변환 커널 후보 세트의 인덱스 값은 lfnstTrSetIdx로 나타낼 수 있다.

[0342] 제 5 사전 설정 룩업 테이블은 표 10에 나타낸 바와 같다. MIP 모드 인덱스 값(즉, modeId의 값)에 따라 predModeIntra의 값을 확정할 수 있다. 그 다음에 predModeIntra의 값에 따라 표 10을 통해 lfnstTrSetIdx의 값을 직접 확정할 수 있다. 즉, 현재 블록에 대해 선택한 LFNST 변환 커널 후보 세트를 확정한다. 여기서, lfnstTrSetIdx의 값은 LFNST에서 사용되는 변환 커널 후보 세트를 나타낸다. modeId의 값은 0, 1, 2, 3, 4 및 5를 포함할 수 있으므로, predModeIntra의 값도 0, 1, 2, 3, 4 또는 5이며, lfnstTrSetIdx와의 대응 관계는 다음과 같다.

[0343] [표 10]

predModeIntra	lfnstTrSetIdx
	x
0 ≤ predModeIntra ≤ 2	0
predModeIntra == 3	2
predModeIntra == 4	1
predModeIntra == 5	3

[0344]

[0345] 구체적인 실시예에서, MIP 모드 인덱스 값에 따라 LFNST 인트라 예측 모드 인덱스 번호의 값을 확정하는 것은, LFNST 인트라 예측 모드 인덱스 번호의 값을 MIP 모드 인덱스 값과 동일하게 설정하는 것을 포함할 수 있다.

[0346] 즉, 본 출원의 실시예는 predModeIntra의 값을 modeId의 값과 동일하게 설정할 수 있으며, 그 다음에 predModeIntra의 값에 따라 표 10을 통해 lfnstTrSetIdx의 값을 직접 확정한다. 즉 현재 블록에 대해 선택한 LFNST 변환 커널 후보 세트를 확정한다. 또한, 본 출원의 다른 특정 실시예에서, 본 출원의 실시예는 modeId의 값을 PLANAR 모드의 값으로 직접 매핑한 다음에, PLANAR 모드에 대응하는 predModeIntra의 값을 사용하여 lfnstTrSetIdx의 값을 확정할 수 있다는 점에 유념해야 한다. 즉 현재 블록에 대해 선택한 LFNST 변환 커널 후보 세트를 확정한다.

[0347] 일부 실시예에서, LFNST 파라미터는 LFNST 계수를 더 포함할 수 있다. LFNST 파라미터를 확정하는 것은 다음 내용을 포함할 수 있다.

[0348] 제 1 변환 계수 블록 중 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 계수를 획득한다.

[0349] 대응하게, LFNST 파라미터를 인코딩하고 비트스트림으로 기입하는 것은 다음 내용을 포함한다.

[0350] LFNST 계수를 양자화하여 양자화 계수를 획득한다.

[0351] 양자화 계수를 인코딩하고 비트스트림으로 기입한다.

[0352] lfnst_index가 0이 아닌 경우, 즉 현재 블록에 대하여 LFNST를 수행하기로 확정된 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 계수를 획득할 수 있음을 유념해야 한다. 그 다음에 LFNST 계수에 대하여 양자화 및 인코딩하고 비트스트림으로 기입한다.

[0353] d[x][y]는 제 1 변환 계수 블록이라고 가정하며, 여기서 x=0, ..., nTbW-1, y=0, ..., nTbH-1이며, 여기서 nTbW와 nTbH는 각각 현재 블록의 너비와 높이를 나타낸다. v[x]는 LFNST 계수를 포함하는 블록을 나타낸다고 가정하며, 여기서 x = 0, ..., nLfnstOutSize-1이며, nLfnstOutSize=(nTbW ≥ 8 && nTbH ≥ 8)? 48:16이다.

[0354] 가능한 실시예에서, 직접 수평 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를

수행하여 LFNST 계수를 획득할 수 있다. 구체적으로 다음과 같다.

$$d[x][y] = (y < 4) ? v[x + (y << \text{Log2LfnstSize})] : ((x < 4) ? v[32 + x + ((y - 4) << 2)] : d[x][y]) \quad (8)$$

log2LfnstSize는 다음과 같이 확정된다.

$$\text{Log2LfnstSize} = (\text{nTbW} \geq 8 \&\& \text{nTbH} \geq 8) ? 3 : 2 \quad (9)$$

다른 가능한 실시예에서, 스캔 순서는 수평 스캔 순서 및 수직 스캔 순서를 포함할 수 있다. 여기서, 스캔 순서는 MIP 전치 지시 파라미터(isTransposed로 나타냄)의 값과 관련된다.

선택적으로, 일부 실시예에서, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 계수를 획득하는 것은 다음 내용을 포함할 수 있다.

MIP 전치 지시 파라미터의 값이 MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하는 것을 지시하는 경우, 수직 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 계수를 획득한다.

MIP 전치 지시 파라미터의 값이 MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하지 않는 것을 지시하는 경우, 수평 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 계수를 획득한다.

구체적으로, isTransposed의 값이 0과 같으면, MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하지 않고, 수평 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할 수 있으며, 구체적으로 다음과 같다.

$$d[x][y] = (y < 4) ? v[x + (y << \text{Log2LfnstSize})] : ((x < 4) ? v[32 + x + ((y - 4) << 2)] : d[x][y]) \quad (10)$$

isTransposed의 값이 1과 같으면, MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하고, 수직 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할 수 있으며, 구체적으로 다음과 같다.

$$d[x][y] = (x < 4) ? v[y + (x << \text{Log2LfnstSize})] : ((y < 4) ? v[32 + y + ((x - 4) << 2)] : d[x][y]) \quad (11)$$

상기 수학식 (9)에 따라 log2LfnstSize를 확정한다.

선택적으로, 일부 실시예에서, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 계수를 획득하는 것은 다음 내용을 포함할 수 있다.

MIP 전치 지시 파라미터의 값이 MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행함을 지시하는 경우, 수평 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 계수를 획득한다.

MIP 전치 지시 파라미터의 값이 MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하지 않음을 지시하는 경우, 수직 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 계수를 획득한다.

구체적으로, isTransposed의 값이 1과 같은 경우, MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하고, 수평 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할 수 있으며, 구체적으로 상기 식(10)에 나타난 바와 같다.

isTransposed의 값이 0과 같은 경우, MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하지 않고, 수직 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할 수 있으며, 구체적으로 상기 식(11)에 나타난 바와 같다.

- [0372] 위의 수학적식 (9)에 따라 $\log_2 \text{LfnstSize}$ 를 확정한다.
- [0373] 또한, 본 출원의 실시예에서, LFNST는 MIP 모드 외에 기존의 인트라 예측 모드(예를 들어, 각도 인트라 예측 모드, DC 모드, PLANAR 모드 등)에도 적용될 수 있다. 구체적인 실현 방식은 MIP 모드의 실현 방식과 유사하다.
- [0374] 일부 실시예에서, MIP 파라미터에 따라 LFNST 변환 커널 후보 세트를 확정하는 것은 다음 내용을 포함할 수 있다.
- [0375] 기존의 인트라 예측 모드의 인덱스 값에 따라 LFNST 인트라 예측 모드 인덱스 번호의 값을 확정한다.
- [0376] LFNST 인트라 예측 모드 인덱스 번호의 값에 따라 제 6 사전 설정 룩업 테이블을 통해 LFNST 변환 커널 후보 세트의 인덱스 값을 확정한다.
- [0377] LFNST 변환 커널 후보 세트의 인덱스 값에 따라, 복수의 후보 LFNST 변환 커널 후보 세트로부터 상기 LFNST 변환 커널 후보 세트를 선택한다.
- [0378] 제 6 사전 설정 룩업 테이블은 표 11과 같다. 기존의 인트라 예측 모드의 인덱스 값에 따라 predModeIntra 의 값을 확정할 수 있다. 그 다음에 predModeIntra 의 값에 따라 표 11을 통해 lfnstTrSetIdx 의 값을 직접 확정할 수 있다. 즉, 현재 블록에 대해 선택한 LFNST 변환 커널 후보 세트를 확정한다. 여기서, 기존 인트라 예측 모드의 인덱스 값은 predModeIntra 에 할당되어 lfnstTrSetIdx 의 값을 확정할 수 있다.
- [0379] [표 11]

predModeIntra	lfnstTrSetIdx
$\text{predModeIntra} < 0$	1
$0 \leq \text{predModeIntra} \leq 1$	0
$2 \leq \text{predModeIntra} \leq 12$	1
$13 \leq \text{predModeIntra} \leq 23$	2
$24 \leq \text{predModeIntra} \leq 44$	3
$45 \leq \text{predModeIntra} \leq 55$	2
$56 \leq \text{predModeIntra} \leq 80$	1

- [0380]
- [0381] 이하, 도 1에 도시된 인코더(100)를 참조하여 본 출원의 실시예의 인코딩 방법을 구체적으로 설명한다.
- [0382] 본 출원의 실시예 중의 현재 블록(또는 "코딩 블록"이라고 함)은 CU 또는 CU의 파티션(예를 들면, 변환 블록)일 수 있음을 이해해야 한다. 구체적으로, 인트라 예측 유닛(106)이 MIP 모드가 현재 블록에 대하여 인코딩을 수행하는 데에 사용된다고 확정할 때, 또는 인트라 예측 유닛(106)이 레이트 왜곡 최적화 방법을 사용하여 현재 블록을 인코딩하는 데에 사용되는 MIP 모드를 평가할 때, 인트라 예측 유닛(106)은 현재 블록의 인트라 예측 블록을 획득하며, 현재 블록의 인트라 예측 블록은 현재 블록 내의 적어도 하나의 픽셀(샘플)의 인트라 예측값을 포함한다.
- [0383] 구체적으로, 인트라 예측 유닛(106)에 대하여, MIP 모드를 이용하여 인트라 예측 블록을 획득하는 단계는 다음과 같다.
- [0384] 먼저, 인트라 예측 유닛(106)은 현재 블록의 인접 픽셀로부터 하나 이상의 참조 픽셀을 획득한다. 예를 들면, 인접 픽셀에 대하여 다운 샘플링을 수행하여 획득하거나, 또는 직접 인접 픽셀로부터 추출한다.
- [0385] 다음, 인트라 예측 유닛(106)은 획득한 참조 픽셀, MIP 행렬 및 시프팅 파라미터를 이용하여 현재 블록의 픽셀 위치에 대응하는 하나 이상의 부분 예측 픽셀을 확정한다. 여기서, 픽셀 위치는 현재 블록 중의 미리 설정된 픽셀 위치일 수 있다. 예를 들어, 픽셀 위치는 균일한 수평 및 수직 좌표값을 갖는다. 시프팅 파라미터는 시프팅 넘버 파라미터 및 시프팅 오프셋 파라미터를 포함하고, 시프팅 파라미터는 부분 픽셀의 인트라 예측값을 획득하는 과정에서 시프팅 동작에 사용된다.
- [0386] 마지막으로, 현재 블록의 일부 픽셀에 대응하는 예측 픽셀을 획득하면, 인트라 예측 유닛(106)은 현재 블록 중 일부 픽셀을 제외한 나머지 픽셀에 대응하는 예측 픽셀을 획득해야 한다. 예를 들어, 인트라 예측 유닛(106)은 보간 필터를 사용하여 나머지 픽셀에 대응하는 예측 픽셀을 획득하고, 보간 필터의 입력은 부분 픽셀 및 인접 픽셀이다.

- [0387] 도 6에 도시된 바와 같이, 본 출원의 실시예에서 제공되는 MIP 모드를 이용하여 인트라 예측 블록을 획득하는 흐름도이다. 이 절차는 인코더(100)에서 구현될 수 있으며, 여기서 "현재 블록"은 "코딩 블록"을 지칭한다.
- [0388] 단계 601, 인트라 예측 유닛(106)은 현재 블록의 인접 픽셀을 획득하며, 예를 들면, 인접 픽셀은 도 6의 단계 601의 현재 블록에 인접한 회색 정방형으로 표기된다. 인트라 예측 유닛(106)은 인접 픽셀로부터 하나 이상의 참조 픽셀을 획득한다. 도 6에 도시된 단계 601의 예시에서, 선택적으로, 인트라 예측 유닛(106)은 두개의 이웃 픽셀의 평균치를 계산하고 평균치를 참조 픽셀로 사용할 수 있다. 선택적으로, 인트라 예측 유닛(106)은 2개의 이웃 픽셀마다 하나를 참조 픽셀로 선택한다. 예를 들어, 도 6에 도시된 단계 601의 예시에서, 인트라 예측 유닛(106)은 현재 블록의 8개의 위쪽 인접 픽셀로부터 4개의 참조 픽셀을 선택하고, 현재 블록의 8개의 좌측 인접 픽셀로부터 다른 4개의 참조 픽셀을 선택한다.
- [0389] 단계 601의 구체적인 과정은 다음과 같다.
- [0390] 인트라 예측 유닛(106)은 블록 분할 유닛(103)으로부터 현재 블록의 너비와 높이를 획득하고, 너비와 높이는 변수 cbWidth 및 cbHeight로 나타낸다. 예시로서, 인트라 예측 유닛(206)은 인트라 예측 모드를 확정하기 위해 레이트 왜곡 최적화 모드 결정 방법을 호출하며, 현재 블록은 하나 이상의 변환 블록으로 분할된다. 변수 nTbW 및 nTbH를 각각 변환 블록의 너비와 높이로 한다. 현재 블록의 인트라 예측값을 획득하기 위해 MIP 모드를 인트라 예측 모드로 사용하는 경우, 인트라 예측 유닛(106)은 MIP의 블록 사이즈 인덱스 값, 즉 mipSizeId로 나타내는 변수를 확정한다.
- [0391] 선택적으로, 인트라 예측 유닛(106)은 mipSizeId의 값을 다음과 같이 확정한다.
- [0392] - nTbW 및 nTbH가 모두 4인 경우, mipSizeId는 0으로 설정된다.
- [0393] - 그렇지 않고, cbWidth 또는 cbHeight가 4인 경우, mipSizeId는 1로 설정된다.
- [0394] - 그렇지 않으면 mipSizeId는 2로 설정된다.
- [0395] 구체적으로, 현재 블록의 사이즈 파라미터가 8x8일 때(즉, cbWidth와 cbHeight가 모두 8인 경우), mipSizeId는 2로 설정된다.
- [0396] 선택적으로, 인트라 예측 유닛(106)은 mipSizeId의 값을 다음과 같이 확정한다.
- [0397] - nTbW와 nTbH가 모두 4인 경우, mipSizeId는 0으로 설정된다.
- [0398] - 그렇지 않고, cbWidth 또는 cbHeight가 4인 경우, 또는 cbWidth와 cbHeight가 모두 8인 경우, mipSizeId는 1로 설정된다.
- [0399] - 그렇지 않으면 mipSizeId는 2로 설정된다.
- [0400] 구체적으로, 현재 블록의 사이즈 파라미터가 8x8인 경우, mipSizeId는 1로 설정된다.
- [0401] 또한, 인트라 예측 유닛(106)은 mipSizeId에 따라 boundarySize 및 predSize의 값을 획득한다. 구체적으로 다음과 같다.
- [0402] - mipSizeId가 0인 경우, boundarySize는 2로 설정되고, predSize는 4로 설정된다.
- [0403] - 그렇지 않고, mipSizeId가 1인 경우, boundarySize는 4로 설정되고, predSize는 4로 설정된다.
- [0404] - 그렇지 않으면(mipSizeId가 2인 경우), boundarySize는 4로 설정되고, predSize는 8로 설정된다.
- [0405] boundarySize는 현재 블록의 상측변 인접 참조 픽셀과 좌측변 인접 참조 픽셀 각각에서 획득된 참조 픽셀의 수량을 나타낸다.
- [0406] 인트라 예측 유닛(106)은 또한 버퍼 pTemp에 저장된 참조 픽셀의 순서를 지시하기 위해 변수 isTransposed를 획득할 수 있다. 예를 들어, isTransposed는 0과 같다는 것은, 인트라 예측 유닛(106)은 먼저 현재 블록의 상측변 인접 참조 픽셀로부터 참조 픽셀을 획득한 다음에 현재 블록의 좌측변 인접 참조 픽셀로부터 참조 픽셀을 획득하는 것을 나타내고, isTransposed는 1과 같다는 것은, 인트라 예측 유닛(106)은 먼저 현재 블록의 좌측변 인접 참조 픽셀로부터 참조 픽셀을 획득한 다음에 현재 블록의 상측변 인접 참조 픽셀로부터 참조 픽셀을 획득하는 것을 나타낸다. isTransposed의 값은 엔트로피 코딩 유닛(115)으로 전송되어 인코딩되고 비트스트림에 기입될 MIP 모드의 파라미터 중 하나로 한다.

- [0407] 위의 수식식 1과 같이, 인트라 예측 유닛(106)은 MIP 모드가 사용될 때 사용되는 참조 픽셀의 개수를 나타내기 위해 변수 $inSize$ 의 값을 획득한다.
- [0408] 인트라 예측 유닛(106)은 현재 블록의 인접 참조 픽셀을 사용하여 참조 픽셀 그룹(어레이 $p[x]$ 로 저장되고, x 는 $0, \dots, inSize - 1$ 임)을 획득하기 위해 다음 프로세스를 호출한다.
- [0409] 인트라 예측 유닛(106)은 현재 블록의 상측변 인접 참조 픽셀(예를 들면, 어레이 $refT$ 에 저장됨)로부터 $nTbW$ 참조 픽셀을 획득하고, 현재 블록의 좌측변 인접 참조 픽셀(예를 들면, 어레이 $refL$ 에 저장됨)로부터 $nTbH$ 참조 픽셀을 획득한다. 인트라 예측 유닛(106)은 $refT$ 에 대하여 다운 샘플링을 호출하여 $boundarySize$ 참조 픽셀을 획득하고, $boundarySize$ 참조 픽셀을 $refT$ 에 저장한다. 인트라 예측 유닛(106)은 $refL$ 에 대하여 다운 샘플링을 호출하여 $boundarySize$ 참조 픽셀을 획득하고, $boundarySize$ 참조 픽셀을 $refL$ 에 저장한다.
- [0410] 인트라 예측 유닛(106)은 또한 버퍼($pTemp$)에 저장된 참조 픽셀의 순서를 지시하기 위해 변수 $isTransposed$ 를 획득할 수 있다. 예를 들어, $isTransposed$ 는 0(또는 FALSE)과 같다는 것은, 인트라 예측 유닛(106)은 먼저 현재 블록의 상측변 인접 참조 픽셀로부터 참조 픽셀을 획득한 다음에 현재 블록의 좌측변 인접 참조 픽셀로부터 참조 픽셀을 획득함을 나타낸다. 그렇지 않고, $isTransposed$ 는 1(또는 TRUE)과 같다는 것은, 인트라 예측 유닛(106)은 먼저 현재 블록의 좌측변 인접 참조 픽셀로부터 참조 픽셀을 획득한 다음에 현재 블록의 상측변 인접 참조 픽셀로부터 참조 픽셀을 획득함을 나타낸다. 인트라 예측 유닛(106)은 레이트 왜곡 최적화 방법을 사용하여, 또는 인접 참조 픽셀과 현재 블록 사이의 상관성 비교를 기반으로 $isTransposed$ 의 값을 확정할 수 있다. $isTransposed$ 의 값은 엔트로피 코딩 유닛(115)으로 전송되어 인코딩되고 비트스트림에 기입될 MIP 모드의 파라미터 중 하나로 한다.
- [0411] 인트라 예측 유닛(106)은 $isTransposed$ 가 지시하는 순서에 따라 $refT$ 및 $refL$ 중의 요소를 버퍼 $pTemp$ 에 넣는다.
- [0412] 선택적으로, 인트라 예측 유닛(106)은 $p[x]$ 를 획득하며, $x=0, \dots, inSize-1$ 이다. 다음과 같다.
- [0413] - $mipSizeId$ 가 2인 경우, $p[x]=pTemp[x+1]-pTemp[0]$ 이다.
- [0414] 그렇지 않으면($mipSizeId$ 가 2보다 작은 경우), $p[0]=pTemp[0]-(1 \ll (BitDepth-1))$, $p[x]=pTemp[x]-pTemp[0]$ 이다.
- [0415] $BitDepth$ 는 현재 블록 중 픽셀의 색상 컴포넌트의 비트 깊이이다. 여기서, 색상 컴포넌트는 RGB 컴포넌트 중 하나, YUV 컴포넌트 중 하나 또는 YCbCr 컴포넌트 중 하나이며, 예를 들어, Y 컴포넌트 등이다.
- [0416] 선택적으로, 인트라 예측 유닛(106)은 $p[x]$ 를 획득할 수 있으며, $x=0, \dots, inSize-1$ 이다. 다음과 같다.
- [0417] - $mipSizeId$ 가 2인 경우, $p[x]=pTemp[x+1]-pTemp[0]$ 이다.
- [0418] - 그렇지 않으면($mipSizeId$ 가 2보다 작음), $p[0]=(1 \ll (BitDepth-1))-pTemp[0]$, $p[x]=pTemp[x]-pTemp[0]$ 이다.
- [0419] 선택적으로, 인트라 예측 유닛(106)은 통일적인 계산 방법을 사용하여 $p[x]$ 의 값을 획득할 수 있으며, $mipSizeId$ 의 값을 판단할 필요가 없다. 예를 들어, 버퍼 $pTemp$ 에 $(1 \ll (BitDepth-1))$ 을 부가 요소로 추가하며, 이 때 인트라 예측 유닛(106)은 $p[x]$ 를 $pTemp[x+1]-pTemp[0]$ 으로 계산한다.
- [0420] 단계 602에서, 인트라 예측 유닛(106)은 참조 픽셀 그룹과 MIP 가중 행렬을 이용하여 현재 블록의 MIP 예측값을 획득한다. MIP 가중 행렬은 MIP 모드에 대응하는 MIP 모드 인덱스 값($ModeId$ 로 나타냄) 및 MIP의 블록 사이즈 인덱스 값($mipSizeId$ 로 나타냄)에 따라 미리 정의된 MIP 가중 행렬 그룹에서 선택된다.
- [0421] 인트라 예측 유닛(106)은 현재 블록 내의 하나 이상의 픽셀 위치에 대응하는 부분 예측 픽셀의 MIP 예측값($predMip[x][y]$ 로 나타냄)을 획득한다. 도 6에 도시된 단계 602의 예시에서, 부분 예측 픽셀은 현재 블록에서 회색 정방향으로 표기된 픽셀이다. 예측 모듈(601)의 입력은 단계 601에서 획득된 참조 픽셀 $p[x]$ 이다. 예측 모듈(601)은 MIP 가중 행렬 및 시프팅 파라미터를 사용하여 부분 예측 픽셀을 계산한다. 여기서, 시프팅 파라미터는 시프팅 넘버 파라미터와 시프팅 오프셋 파라미터를 포함한다.
- [0422] 가능한 실시예에서, 예측 모듈(601)이 좌표 (x, y) 로 픽셀을 나타내는 경우, 획득한 예측 픽셀은 $predMip[x][y]$ 로 나타내며, $predMip[x][y]$ 의 계산 공식은 식 (5)와 식 (6)에 나타낸 바와 같다.
- [0423] 식 (5) 또는 식 (6)에서, $mWeight[i][j]$ 는 MIP 가중 행렬이다. 여기서 행렬 요소는 미리 설정된 상수이거나, 또는 훈련 방법을 사용하여 적응적으로 업데이트될 수 있으며, 훈련 방법의 입력은 하나 이상의 코딩된 픽처 또는

블록이고, 또는 외부 장치에 의해 인코더(100)에 제공되는 다른 비트스트림 중의 픽처이다. f_0 는 oW 를 확정하는 데에 사용되는 시프팅 오프셋 파라미터이다. sW 는 시프팅 넘버 파라미터이다. $p[i]$ 는 참조 픽셀을 사용하여 계산된 MIP 입력 샘플값이다. $pTemp[0]$ 은 제 1 임시 참조값에서 인덱스 번호 0에 대응하는 값을 나타낸다. ">>"는 VVC 중 정의된 이진 오른쪽 시프팅 연산자이다. 현재 블록에 대하여 인트라 예측을 수행하기 위해 MIP 모드가 최종적으로 확정될 때, 인트라 예측 유닛(106)은 또한 $mWeight[i][j]$ 를 엔트로피 코딩 유닛(115)으로 송신할 수 있다. 엔트로피 코딩 유닛(115)은 $mWeight[i][j]$ 를 비트스트림의 하나 이상의 데이터 유닛에 기입할 수 있다.

- [0424] 예측 모듈(601)은 현재 블록의 사이즈 및 현재 블록에 사용되는 MIP 모드에 따라 sW 및 f_0 의 값을 확정할 수 있다. 하나의 예시에서, 예측 모듈(601)은 룩업 테이블을 사용하여 sW 및 f_0 의 값을 획득한다.
- [0425] 선택적으로, 예측 모듈(601)은 위의 표 4를 사용하여 현재 블록의 사이즈 파라미터 및 MIP 모드에 따라 sW 를 확정할 수 있다.
- [0426] 선택적으로, 예측 모듈(601)은 또한 위의 표 5를 사용하여 현재 블록의 사이즈 파라미터에 따라 sW 를 확정할 수 있다.
- [0427] 선택적으로, 예측 모듈(601)은 sW 를 직접 상수값으로 설정할 수 있다. 예를 들어, 다양한 사이즈 파라미터를 갖는 블록 및 다른 MIP 모드에 대하여, 예측 모듈(601)은 sW 를 5로 설정할 수 있다. 또는, 다양한 사이즈 파라미터를 갖는 블록 및 다른 MIP 모드에 대하여, 예측 모듈(601)은 sW 를 6으로 설정할 수 있다. 또는 다양한 사이즈 파라미터를 갖는 블록 및 다른 MIP 모드에 대하여, 예측 모듈(601)은 sW 를 7로 설정할 수 있다.
- [0428] 선택적으로, 예측 모듈(601)은 위의 표 6을 사용하여 현재 블록의 사이즈 파라미터 및 MIP 모드에 따라 f_0 를 확정할 수 있다.
- [0429] 선택적으로, 예측 모듈(601)은 위의 표 7을 사용하여 현재 블록의 사이즈 파라미터에 따라 f_0 를 확정할 수 있다.
- [0430] 선택적으로, 예측 모듈(601)은 f_0 를 직접 상수값(예를 들어, 0 내지 100)으로 설정할 수 있다. 예를 들어, 다양한 사이즈 파라미터를 갖는 블록 및 다른 MIP 모드에 대하여, 예측 모듈(601)은 f_0 를 32로 설정한다. 또는 예측 모듈(601)은 f_0 를 46으로 설정한다. 또는 예측 모듈(601)은 f_0 를 56으로 설정한다. 또는 예측 모듈(601)은 f_0 를 66으로 설정한다.
- [0431] 인트라 예측 유닛(106)은 $predMip$ 내의 MIP 예측값에 대하여 클리핑 조작을 수행할 수 있다. $isTransposed$ 가 1 (또는 TRUE)일 때, $predSize \times predSize$ 어레이 $predMip[x][y]$ ($x=0, \dots, predSize-1, y=0, \dots, predSize-1$)는 $predTemp[y][x]=predMip[x][y]$ 로 변환되고, 그리고 $predMip=predTemp$ 이다.
- [0432] 예시적으로, 선택적으로, 현재 블록의 사이즈 파라미터가 8x8인 경우(즉, $cbWidth$ 및 $cbHeight$ 가 모두 8인 경우), 인트라 예측 유닛(106)은 $mipSizeId$ 를 2로 확정하고, 8x8 $predMip$ 를 획득한다.
- [0433] 선택적으로, 현재 블록의 사이즈 파라미터가 8x8인 경우(즉, $cbWidth$ 및 $cbHeight$ 가 모두 8인 경우), 인트라 예측 유닛(106)은 $mipSizeId$ 를 1로 확정하고, 4x4 $predMip$ 를 획득한다.
- [0434] 단계 603, 인트라 예측 유닛(106)은 현재 블록의 인트라 예측값(어레이에 $predSamples[x][y]$ 로 저장되며 $x=0, \dots, nTbW-1, y=0, \dots, nTbH-1$)을 획득한다. 다음과 같다.
- [0435] - 인트라 예측 유닛(106)이 $nTbW$ 가 $predSize$ 보다 크거나 또는 $nTbH$ 가 $predSize$ 보다 크다고 확정하면, 인트라 예측 유닛(106)은 $predMip$ 을 사용하여 $predSamples$ 를 획득하기 위해 업 샘플링 프로세스를 호출한다. 인트라 예측 유닛(106)은 현재 블록 중의 일부 픽셀을 제외한 나머지 픽셀에 대응하는 예측 픽셀을 획득한다. 도 6에 도시된 바와 같이, 인트라 예측 유닛(106)은 필터링 모듈(602)을 사용하여 현재 블록 중의 부분 픽셀 이외의 나머지 픽셀에 대응하는 예측 픽셀을 획득할 수 있다. 필터링 모듈(602)의 입력은 단계 602에서 회색 정방향으로 표기된 픽셀이다. 필터링 모듈(602)은 하나 이상의 보간 필터를 사용할 수 있고, 상기 입력을 이용하여 현재 블록 중의 부분 픽셀 이외의 나머지 픽셀에 대응하는 예측 픽셀을 획득한다. 예를 들어, 입력은 참조 픽셀 및 현재 블록 중 하나 이상의 픽셀 위치의 부분 예측 픽셀을 포함한다. 또는, 입력은 이웃 픽셀 및 현재 블록 중 하나 이상의 픽셀 위치의 부분 예측 픽셀을 포함한다. 또는, 입력은 참조 픽셀, 이웃 픽셀 및 현재 블록 중 하나 이상의 픽셀 위치의 부분 예측 픽셀을 포함한다. 예를 들어, 선택적으로, 현재 블록의 사이즈가 8x8일 때(즉, $cbWidth$ 및 $cbHeight$ 가 모두 8임), 인트라 예측 유닛(106)은 $mipSizeId$ 를 1로 확정하고, 인트라 예측 유닛(106)은 현재 블록의 8x8 인트라 예측 블록을 획득하기 위해 업 샘플링을 4x4 $predMip$ 에 응용한다.

- [0436] - 그렇지 않으면, 인트라 예측 유닛(106)은 현재 블록의 인트라 예측 블록을 현재 블록의 MIP 예측 블록과 동일하게 설정한다. 즉, $\text{predSamples}[x][y](x=0, \dots, \text{nTbW}-1, y=0, \dots, \text{nTbH}-1)$ 를 $\text{predMip}[x][y]$ 와 같게 설정한다. 예를 들어, 구체적으로, 현재 블록의 사이즈 매개변수가 8x8일 때(즉, cbWidth 및 cbHeight가 모두 8인 경우), 인트라 예측 유닛(106)은 mipSizeId를 2로 확정하고, 인트라 예측 유닛(106)은 현재 블록의 predSamples를 획득하며, 그 사이즈는 8x8(즉, cbWidth와 cbHeight가 모두 8임)이다.
- [0437] 이와 같이, 도 6에 도시된 절차의 블록도에 따르면, 단계 603 이후에, 인트라 예측 유닛(106)은 현재 블록(즉, CU)의 인트라 예측 블록을 획득할 수 있고, 즉 현재 블록의 적어도 하나의 픽셀의 인트라 예측값을 확정할 수 있다.
- [0438] 예측 유닛(102)은 현재 블록의 인트라 예측 블록을 출력한다. 제 1 가산기(107)는 분할 유닛(101)의 출력 중의 현재 블록과 현재 블록의 인트라 예측 블록 사이의 차이값, 즉 잔차 블록(즉, 잔차 CU)을 계산한다. 변환 유닛(108)은 잔차 블록을 판독하고, 잔차 블록에 대하여 하나 이상의 변환 조작을 수행하여 계수를 얻는다. 양자화 유닛(109)은 계수를 양자화하여 양자화된 계수(즉, 레벨)를 출력한다.
- [0439] 변환 유닛(108)은 잔차 블록에 대하여 최초 DCT 기반으로 설계된 정수 변환과 같은 제 1차 변환을 수행한다. 변환 유닛(108)은 2차 변환이 상기 블록에 적용될 수 있는지 여부를 확정한다. 적용이 허용될 때, 변환 유닛(108)은 제 1차 변환을 수행한 후에 획득된 계수에 2차 변환을 적용할지 여부를 추가로 확정한다. LFNST는 2차 변환의 예시이다.
- [0440] 도 7을 참조하면, 본 출원의 실시예에서 제공되는 LFNST를 수행하는 절차를 도시하는 도면이다. 도 7에서 블록(701)은 잔차 블록을 나타낸다.
- [0441] 단계 701, 변환 유닛(108)은 블록(701)에 대하여 제 1차 변환(즉, 코어 변환), 예를 들어, 원래 DCT를 기반으로 설계된 정수 변환을 수행하여 제 1차 변환에 대응하는 변환 계수("제 1 변환 계수"라고 함)를 포함하는 블록(702)을 획득한다.
- [0442] 단계 702, 변환 유닛(108)은 블록(702) 중의 전부 또는 일부 제 1 변환 계수에 대하여 LFNST와 같은 2차 변환을 수행하는 것이 허용되는지 여부를 확정한다.
- [0443] 예시적으로, 선택적으로, LFNST를 포함하는 변환 블록의 인트라 예측 모드가 MIP 모드이고, 변환 블록의 너비 및 높이의 최소값이 8이상인 경우, 변환 유닛(108)이 변환 블록에 대하여 LFNST를 수행하는 것을 허용한다.
- [0444] 선택적으로, LFNST를 포함하는 변환 블록의 인트라 예측 모드가 MIP 모드이고, mipSizeId의 값이 사전 설정값(예를 들면, 2) 또는 복수의 사전 설정값(예를 들면, 1, 2) 중 하나인 경우, 변환 유닛(108)이 변환 블록에 대하여 LFNST를 수행하는 것을 허용한다.
- [0445] 변환 유닛(108)이 변환 블록에 대하여 LFNST를 수행하는 것을 허용할 때, 변환 유닛(108)은 LFNST 파라미터, 예를 들어, lfnst_index를 확정한다. lfnst_index가 0일 때, 변환 유닛(108)이 변환 블록에 대하여 LFNST를 수행하지 않는 것을 나타낸다. lfnst_index가 0보다 큰 경우, LFNST 변환 커널 후보 세트 중의 변환 커널을 이용하여 변환 블록에 대하여 LFNST를 수행하는 것을 나타낸다.
- [0446] 변환 유닛(108)은 전통적인 레이트 왜곡 최적화 방법을 사용하여 lfnst_index의 값을 확정할 수 있다. 우선, 변환 유닛(108)은 변환 블록에 대하여 LFNST를 수행하지 않고 제 1 대가 결과를 획득한다.
- [0447] 변환 유닛(108)은 MIP 파라미터에 따라 LFNST 변환 커널 후보 세트를 선택한다. MIP 파라미터는 MIP 모드 인덱스 값(즉, modeId) 및 MIP 전치 지시 파라미터(즉, isTransposed) 중 적어도 하나를 포함할 수 있다.
- [0448] 변환 유닛(108)은 변수 predModeIntra의 값을 modeId의 값과 동일하게 설정한다. 변환 유닛(108)은 블록(702)에서 제 1 변환 계수의 일부, 예를 들어, 서브 블록(7001, 7002, 7003)의 계수를 선택하고, 서브 블록의 계수에 대하여 LFNST를 수행하기 위한 변환 커널을 확정한다. 서브 블록의 사이즈는 사전 설정값, 예를 들어, 8x8일 수 있다. 변환 유닛(108)은 predModeIntra의 값에 따라 LFNST 변환 커널 후보 세트의 인덱스 값(즉, lfnstTrSetIdx)을 확정한다. MIP 모드에서의 블록 코딩에 대하여, 변환 유닛(108)은 표 10의 룩업 테이블을 사용하여 lfnstTrSetIdx의 값을 획득할 수 있다.
- [0449] 변환 유닛(108)은 전통적인 레이트 왜곡 최적화 방법을 사용하여 lfnst_index의 값을 확정할 수 있다(즉, 이 값이 0보다 크다는 것은 LFNST 변환 커널 후보 세트에서 사용되는 변환 커널은 lfnstTrSetIdx에 의해 표시됨을 나타냄). 변환 유닛(108)은 lfnst_index의 값을 LFNST 변환 커널 후보 세트에 있어서의 변환 커널의 인덱스 번호

로 설정하고, 이렇게 하면 대가 함수가 최소값을 갖도록 하고, 또한 제 2 대가 결과를 획득한다.

- [0450] 제 1 대가 결과가 제 2 대가 결과보다 크면, 변환 유닛(108)은 변환 블록에 대하여 LFNST를 수행하기로 확정하고, 변환 커널의 인덱스 번호를 나타내는 `lfnst_index`를 엔트로피 코딩 유닛(115)으로 전송한다. 그렇지 않으면, 변환 유닛(108)은 변환 블록에 대하여 LFNST를 수행하지 않기로 확정하고, `lfnst_value`를 0으로 설정하고, LFNST를 수행하지 않음을 나타내는 `lfnst_index`를 엔트로피 코딩 유닛(115)으로 전송한다.
- [0451] `lfnst_index`가 0이 아닌 경우, 변환 유닛(108)은 서브 블록(7001, 7002, 7003)으로부터의 계수를 사용하여 변환 블록에 대하여 LFNST를 수행하고, LFNST 계수를 획득하고, LFNST 계수를 블록(703)의 서브 블록(7011, 7012, 7013)에 배치한다. `d[x][y]`는 제 1 변환 계수의 블록이라고 가정하고, $x=0, \dots, nTbW-1, y=0, \dots, nTbH-1$ 이며, $nTbW$ 및 $nTbH$ 는 각각 블록(703)의 너비와 높이이다. `v[x]`는 LFNST 계수의 블록이라고 가정하고, $x=0, \dots, nLfnstOutSize-1, nLfnstOutSize=(nTbW \geq 8 \ \&\& \ nTbH \geq 8)? \ 48:16$ 이다.
- [0452] 선택적으로, 변환 유닛(108)은 다음 명령을 사용하여 변환 계수의 블록(703)을 획득하고, `d[x][y]` 중의 LFNST 계수의 블록(703)을 양자화 유닛(109)에 전달하며, 구체적으로 식(8)에 나타난 바와 같다. $\log 2LfnstSize$ 는 위의 식(9)에 나타난 바와 같다.
- [0453] 선택적으로, `isTransposed`가 0과 같을 때, 변환 유닛(108)은 다음 명령을 사용하여 LFNST 계수의 블록(703)을 획득하고, `d[x][y]` 중의 LFNST 계수의 블록(703)을 양자화 유닛(109)에 전달하며, 구체적으로 식(10)에 나타난 바와 같다.
- [0454] `isTransposed`가 1과 같을 때, 변환 유닛(108)은 다음 명령을 사용하여 LFNST 계수의 블록(703)을 획득하고, `d[x][y]` 중의 LFNST 계수의 블록(703)을 양자화 유닛(109)에 전달하며, 구체적으로 식(11)에 나타난 바와 같다.
- [0455] 선택적으로, `isTransposed`가 1과 같을 때, 변환 유닛(108)은 다음 명령을 사용하여 LFNST 계수의 블록(703)을 획득하고, `d[x][y]` 중의 LFNST 계수의 블록(703)을 양자화 유닛(109)에 전달하며, 구체적으로 식(10)에 나타난 바와 같다.
- [0456] `isTransposed`가 0과 같을 때, 변환 유닛(108)은 다음 명령을 사용하여 LFNST 계수의 블록(703)을 획득하고, `d[x][y]` 중의 LFNST 계수의 블록(703)을 양자화 유닛(109)에 전달하며, 구체적으로 식(11)에 나타난 바와 같다.
- [0457] $\log 2LfnstSize$ 는 위의 식(9)에 따라 확정된다.
- [0458] 또한, 변환 유닛(108)은 또한 전통적인 인트라 예측 모드(즉, 각도 인트라 예측 모드, DC 모드 및 PLANAR 모드 등)로 인코딩된 변환 블록에 대하여 `lfnst_index`의 값을 확정할 수 있다. MIP 모드에서 변환 블록의 `lfnst_index`의 값을 확정하는 레이트 왜곡 최적화 방법과 유사하며, 이 때 변환 블록에 대하여 LFNST를 수행하지 않고 제 1 대가 결과를 확정한다.
- [0459] 변환 유닛(108)은 룩업 테이블을 사용하여 기존의 인트라 예측 모드의 모드 인덱스 값(`predModeIntra`에 할당)에 따라 `lfnstTrSetIdx`의 값을 확정한다. 즉 현재 블록에 대해 선택된 LFNST 변환 커널 후보 세트를 확정하고, 구체적으로 표 11에 나타난 바와 같다.
- [0460] 그 다음, 변환 유닛(108)은 전통적인 레이트 왜곡 최적화 방법을 사용하여 `lfnst_index`의 값을 확정할 수 있다 (즉, 이 값이 0보다 크다는 것은 LFNST 변환 커널 후보 세트에서 사용되는 변환 커널이 `lfnstTrSetIdx`에 의해 표시됨을 나타냄). 변환 유닛(108)은 `lfnst_index`의 값을 LFNST 변환 커널 후보 세트에 있어서의 변환 커널의 인덱스 번호로 설정하고, 이렇게 하면 대가 함수가 최소값을 갖도록 하고, 또한 제 2 대가 결과를 획득한다.
- [0461] 제 1 대가 결과가 제 2 대가 결과보다 크면, 변환 유닛(108)은 변환 블록에 대하여 LFNST를 수행하기로 확정하고, 변환 커널의 인덱스 번호를 나타내는 `lfnst_index`를 엔트로피 코딩 유닛(115)으로 전송한다. 그렇지 않으면, 변환 유닛(108)은 변환 블록에 대하여 LFNST를 수행하지 않기로 확정하고, `lfnst_value`를 0으로 설정하고, LFNST를 수행하지 않음을 나타내는 `lfnst_index`를 엔트로피 코딩 유닛(115)으로 전송한다.
- [0462] `lfnst_index`가 0이 아닌 경우, 변환 유닛(108)은 서브 블록(7001, 7002, 7003)으로부터의 계수를 사용하여 변환 블록에 대하여 LFNST를 수행하고, LFNST 계수를 획득하고, LFNST 계수를 블록(703)의 서브 블록(7011, 7012, 7013)에 배치한다. `d[x][y]`는 제 1 변환 계수의 블록이라고 가정하고, $x=0, \dots, nTbW-1, y=0, \dots, nTbH-1$ 이며, $nTbW$ 및 $nTbH$ 는 각각 블록(703)의 너비와 높이이다. `v[x]`는 LFNST 계수의 블록이라고 가정하고, $x=0, \dots, nLfnstOutSize-1, nLfnstOutSize=(nTbW \geq 8 \ \&\& \ nTbH \geq 8)? \ 48:16$ 이다.
- [0463] 선택적으로, `predModeIntra`가 34보다 작거나 같을 때, 변환 유닛(108)은 다음 명령을 사용하여 LFNST 계수의 블

록(703)을 획득하고, $d[x][y]$ 중의 LFNST 계수의 블록(703)을 양자화 유닛(109)에 전달하며, 구체적으로 식(10)에 나타낸 바와 같다.

- [0464] predModeIntra가 34보다 클 때, 변환 유닛(108)은 다음 명령을 사용하여 LFNST 계수의 블록(703)을 획득하고, $d[x][y]$ 중의 LFNST 계수의 블록(703)을 양자화 유닛(109)에 전달하며, 구체적으로 식(11)에 나타낸 바와 같다.
- [0465] 선택적으로, predModeIntra가 34보다 클 때, 변환 유닛(108)은 다음 명령을 사용하여 LFNST 계수의 블록(703)을 획득하고, $d[x][y]$ 중의 LFNST 계수의 블록(703)을 양자화 유닛(109)에 전달하며, 구체적으로 식(10)에 나타낸 바와 같다.
- [0466] predModeIntra가 34보다 작거나 같을 때, 변환 유닛(108)은 다음 명령을 사용하여 LFNST 계수의 블록(703)을 획득하고, $d[x][y]$ 중의 LFNST 계수의 블록(703)을 양자화 유닛(109)에 전달하며, 구체적으로 식(11)에 나타낸 바와 같다.
- [0467] $\log_2 \text{LfnstSize}$ 는 위의 식(9)에 따라 확정된다.
- [0468] 또한, 역양자화 유닛(110)은 양자화된 계수를 스케일링(scaling)하여 재구성된 계수를 출력한다. 역변환 유닛(111)은 변환 유닛(108)의 변환에 대응하는 하나 이상의 역변환을 수행하고 재구성된 잔차를 출력한다. 제 2 가산기(112)는 재구성된 잔차와 예측 유닛(102)으로부터의 현재 블록의 인트라 예측 블록을 가산함으로써 재구성된 CU를 계산한다. 제 2 가산기(112)는 또한 그 출력을 예측 유닛(102)으로 송신하여 인트라 예측 참조로서 사용한다.
- [0469] 필터링 유닛(113)의 출력은 디코딩된 픽처 또는 서브 픽처이며, 이러한 디코딩된 픽처 또는 서브 픽처는 DPB 유닛(114)으로 송신된다. DPB 유닛(114)은 타이밍 및 제어 정보에 따라 디코딩된 픽처를 출력한다. 여기서, DPB 유닛(114)에 저장된 픽처는 예측 유닛(102)에서 인트라 예측 또는 인트라 예측을 수행하기 위한 참조로 사용될 수도 있다.
- [0470] 엔트로피 코딩 유닛(115)은 인코더(100)로부터의 유닛의 파라미터(디코딩된 픽처를 획득하는데 필요함) 및 제어 파라미터와 추가 정보를 이진 표현으로 변환하고, 각 데이터 유닛의 선택스 구조에 따라 이러한 이진 표현을 생성된 비트스트림(또는 "비디오 비트스트림"이라고 함)에 기입하며, 구체적으로 표 9A 및 표 9B에 나타낸 바와 같다.
- [0471] 표 9A의 선택스 구조를 이용하여, 코딩 유닛에서 변환 블록의 너비와 높이의 최소값이 8이상인 경우(즉, 너비와 높이가 모두 8이상인 경우), 엔트로피 코딩 유닛(115)은 MIP 모드를 사용하여 코딩 유닛의 lfnst_index 를 인코딩한다. 또 다른 선택 가능한 조건은 너비와 높이가 모두 제 1 값(예를 들면, 8)보다 크거나 같고, 너비와 높이의 합이 제 2 값(예를 들면, 16)보다 크거나 같은 것으로 설정될 수 있다.
- [0472] 표 9B의 선택스 구조를 사용하여, 엔트로피 코딩 유닛(115)은 코딩 유닛의 mipSizeId 가 2와 같을 때에 MIP 모드를 사용하여 코딩 유닛의 lfnst_index 를 인코딩한다. 다른 선택 가능한 조건은 다음과 같이 설정될 수 있다. 코딩 유닛의 mipSizeId 는 여러 값(예를 들면, 1 및 2) 중 하나와 같다.
- [0473] 도 8은 본 출원의 실시예에서 제공되는 LFNST 파라미터를 인코딩하는 흐름도이다. 도 8에 도시된 바와 같이, 상기 프로세스는 다음 내용을 포함할 수 있다.
- [0474] S801, 엔트로피 코딩 유닛(115)은 LFNST 파라미터를 인코딩할지 여부를 확정하기 위해 위의 조건을 검사한다.
- [0475] S802, 엔트로피 코딩 유닛(115)은 컨텍스트 기반 적응 이진 산술 코딩(context-based adaptive binary arithmetic coding, CABAC)을 사용하여 LFNST 파라미터를 인코딩한다.
- [0476] S801에서 엔트로피 코딩 유닛(115)은 검사하여 LFNST 파라미터, 예를 들어, lfnst_index 의 값을 확정하고, 다음에 S802에서 표 9A 및 표 9B의 서술자가 "ae(v)"인 CABAC를 사용하여 lfnst_index 의 값을 인코딩할 수 있다.
- [0477] 본 출원의 실시예에서, 인코더(100)는 프로세서 및 인코딩 프로그램을 기록하는 저장 매체를 갖는 컴퓨팅 장치일 수 있다. 프로세서가 인코딩 프로그램을 판독하고 실행하면, 인코더(100)는 입력 영상을 읽고 대응하는 비트스트림을 생성한다. 또한, 인코더(100)는 하나 이상의 칩을 갖는 컴퓨팅 장치일 수 있다. 칩 상의 집적 회로로 구현된 유닛은 도 1의 대응 유닛과 유사한 연결 및 데이터 교환을 통해 유사한 기능을 갖는다.
- [0478] 본 출원의 실시예는 인코딩 방법을 제공한다. 현재 블록의 예측 파라미터를 확정하고, 예측 파라미터는 예측 모드 파라미터를 포함하며, 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra

Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하며, MIP 입력 샘플값, MIP 가중행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하고, MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하며, 현재 블록의 크로마 컴포넌트의 인트라 예측값에 따라 현재 블록의 크로마 컴포넌트의 예측 잔차값을 확정하고, 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정하고, LFNST 파라미터를 인코딩하고 비트스트림으로 기입한다. 이와 같이, MIP 모드에 대하여, 인코딩 성능을 보장할 뿐만 아니라 복잡성을 줄이고, 인코딩 과정에 필요한 저장 공간을 줄이며, 인코딩 효율을 효과적으로 향상시킬 수 있다. 또한, LFNST 기술을 MIP 모드 예측에 적용하면, MIP 파라미터의 도입으로 인해 LFNST 변환이 보다 유연해지고 인코딩 효율이 더욱 향상될 수 있다.

- [0479] 본 출원의 다른 실시예에서, 도 9를 참조하면, 본 출원의 실시예에서 제공되는 디코딩 방법의 흐름도이다. 도 9에 도시된 바와 같이, 상기 방법은 다음 내용을 포함할 수 있다.
- [0480] S901, 비트스트림을 파싱하여 현재 블록의 예측 파라미터 및 LFNST(Low-Frequency Non-Separable Transform) 파라미터를 획득하고, 예측 파라미터는 예측 모드 파라미터를 포함한다.
- [0481] 디코딩될 픽처는 복수의 픽처 블록으로 분할될 수 있으며, 각 디코딩될 픽처 블록을 디코딩 블록이라고 할 수 있다. 여기서, 각 디코딩 블록은 제 1 화상 컴포넌트, 제 2 화상 컴포넌트 및 제 3 화상 컴포넌트를 포함할 수 있다. 현재 블록은 비디오 픽처에 있어서의 현재 제 1 화상 컴포넌트, 제 2 화상 컴포넌트, 또는 제 3 화상 컴포넌트의 예측이 수행될 디코딩 블록이다.
- [0482] 또한, 예측 파라미터는 예측 모드 파라미터를 포함할 수 있으며, 예측 모드 파라미터는 현재 블록이 사용하는 예측 모드를 지시하는 데에 사용되고, 상이한 예측 모드는 상이한 예측 모드 파라미터에 대응된다는 점에 유념해야 한다. 예측 모드는 일반적으로 인터 예측 모드, 전통적인 인트라 예측 모드 및 비전통적인 인트라 예측 모드 등을 포함한다. 전통적인 인트라 예측 모드는 DC 모드, PLANAR 모드, 각도 인트라 예측 모드 등을 포함할 수 있다. 비전통적 인트라 예측 모드는 MIP 모드, CCLM 모드, IBC 모드 및 PLT 모드 등을 포함할 수 있다. 즉, 인코더는 최적의 예측 모드를 선택하여 현재 블록을 인코딩한다. 이 과정에서 현재 블록의 예측 모드를 확정하고, 따라서 대응하는 예측 모드 파라미터를 획득하며, 그 다음에 예측 모드 파라미터를 포함하는 예측 파라미터는 비트스트림에 기입되어 인코더에서 디코더로 전송된다.
- [0483] 이와 같이, 디코더에서 비트스트림을 파싱하여 현재 블록의 예측 파라미터를 획득할 수 있고, 파싱을 통해 획득된 예측 파라미터에 포함된 예측 모드 파라미터를 이용하여 현재 블록이 MIP 모드를 사용하는지 여부를 확정할 수 있다.
- [0484] S902, 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정한다.
- [0485] 현재 블록에 대하여, 본 출원의 실시예는 MIP 모드를 이용하여 현재 블록의 크로마 컴포넌트에 대하여 인트라 예측을 수행한다는 점에 유념해야 한다. 이 과정에서 먼저 현재 블록의 인접 샘플값을 획득할 필요가 있고, 그 다음에 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정한다.
- [0486] 현재 블록의 인접 샘플값은 현재 블록의 왼쪽 인접 샘플값 및 현재 블록의 위쪽 인접 샘플값을 포함할 수 있다. 즉, 현재 블록의 왼쪽 인접 샘플값과 위쪽 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값이 확정될 수 있다.
- [0487] 일부 실시예에서, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하는 것은 아래 내용을 포함할 수 있다.
- [0488] 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정한다.
- [0489] 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 제 1 임시 참조값을 획득한다.
- [0490] 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우, 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정한다. MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 제 2 상수값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정한다. MIP 입력 샘플값 중 인덱스 번호 i 에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 i 에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의

차이로 설정하며, 여기서 i 는 0보다 큰 정수이다.

- [0491] 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위를 벗어나는 경우, MIP 입력 샘플값 중 인덱스 번호 j 에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 $(j+1)$ 에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하며, j 는 0 이상의 정수이다.
- [0492] 또한, 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우, 상기 방법은 다음 내용을 더 포함할 수 있다.
- [0493] 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정한다. MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값과 제 2 상수값의 차이로 설정한다. MIP 입력 샘플값 중 인덱스 번호 i 에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 i 에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하며, 여기서 i 는 0보다 큰 정수이다.
- [0494] 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링 처리를 수행한 다음에 제 1 임시 참조값을 획득할 수 있다. 제 1 임시 참조값에 대하여, 구체적으로 필터링된 인접 샘플값을 버퍼(pTemp로 나타냄)에 버퍼링할 수 있다. 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값은 pTemp[0]이고, 제 1 임시 참조값 중 인덱스 번호 i 에 대응하는 값은 pTemp[i]이다.
- [0495] 본 출원의 실시예에서, 예측 파라미터는 예측 모드 파라미터 외에 현재 블록의 사이즈 파라미터를 더 포함할 수 있음에 유념해야 한다. 현재 블록의 사이즈 파라미터는 현재 블록의 너비(nTbW로 나타냄) 및 높이(nTbH로 나타냄)를 포함할 수 있다. 또한, 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값(즉, mipSizeId)을 확정할 수 있다.
- [0496] 또한, 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄)이 미리 설정된 범위 내에 있는지 여부에 따라 현재 블록의 사이즈 파라미터의 값이 미리 설정된 범위 내에 있는지 여부를 확인할 수 있다. 구체적으로, mipSizeId=0 또는 1인 경우, 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있음을 나타내고, 즉 현재 블록의 사이즈 파라미터의 값이 미리 설정된 범위 내에 있음을 나타낸다. mipSizeId=2인 경우, 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위를 벗어났음을 나타내고, 즉 현재 블록의 사이즈 파라미터의 값이 미리 설정된 범위 내에 있지 않음을 나타낸다.
- [0497] 즉, MIP 입력 샘플값은 버퍼(pTemp로 나타냄), 현재 블록의 블록 사이즈 인덱스 값(MipSizeId로 나타냄), 현재 블록의 인접 샘플값의 비트 깊이(BitDepth로 나타냄)에 따라 결정된다. MIP 입력 샘플값에 포함된 입력 샘플 개수는 단지 현재 블록의 블록 사이즈 인덱스 값에만 관련되고, 마지막으로 MIP 입력 샘플값 중 인덱스 번호 x 에 대응하는 값(p[x]로 나타냄)을 획득할 수 있다.
- [0498] 또한, 일부 실시예에 있어서, 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정하는 것은 아래 내용을 포함할 수 있다.
- [0499] 제 2 상수값을 2의 정수 거듭제곱과 동일하게 설정하며, 거듭제곱의 지수는 현재 블록의 인접 샘플값의 비트 깊이에서 1을 뺀 값과 같다.
- [0500] 또한, 일부 실시예에 있어서, 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정하는 것은 아래 내용을 포함할 수 있다.
- [0501] "1"에 대하여 이진 왼쪽 시프팅을 수행하여 제 2 상수값을 획득하고, 왼쪽으로 시프트된 비트의 수는 현재 블록의 인접 샘플값의 비트 깊이에서 1을 뺀 것과 같다.
- [0502] 즉, 현재 블록의 인접 샘플값의 비트 깊이(BitDepth로 나타냄)를 획득한 다음에, 제 2 상수값은 $1 \ll (\text{BitDepth} - 1)$ 또는 $2^{(\text{BitDepth} - 1)}$ 로 표시될 수 있다. 이와 같이, 현재 블록의 사이즈 파라미터의 값이 미리 설정된 범위 내에 있는 경우, 제 2 상수값을 결합하여 현재 블록의 MIP 입력 샘플값을 확정할 수 있다.
- [0503] 또한, MIP 입력 샘플은 행렬 곱셈 연산에 사용되는 행렬 벡터임을 유념해야 한다. 현재 관련 기술 방안은, 버퍼(pTemp로 나타냄), 현재 블록의 유형(즉, 현재 블록의 블록 사이즈 인덱스 값, mipSizeId로 나타냄), 현재 블록의 인접 샘플값의 비트 깊이(BitDepth로 나타냄), MIP 입력 샘플의 개수에 따라 확정되고, 마지막으로 MIP 입력 샘플값 중 인덱스 번호 x 에 대응하는 값(p[x]로 나타냄)을 획득한다.
- [0504] 하나의 가능한 실시예에 있어서, mipSizeId=0 또는 1인 경우, $1 \ll (\text{BitDepth} - 1)$ 빼기 pTemp[0]을 p[0]으로 하고; x 가 0과 같지 않으면, pTemp[x] 빼기 pTemp[0]을 p[x]로 한다. 구체적으로 식(2)에 나타난 바와 같다.

- [0505] 다른 가능한 실시예에 있어서, mipSizeId=0 또는 1인 경우, pTemp[0] 빼기 $1 \ll (\text{BitDepth}-1)$ 을 p[0]으로 하고; x가 0과 같지 않으면, pTemp[x] 빼기 pTemp[0]을 p[x]로 한다. 구체적으로 식(3)에 나타난 바와 같다.
- [0506] 또 다른 가능한 실시예에 있어서, mipSizeId=2인 경우, pTemp[x+1] 빼기 pTemp[0]을 p[x]로 한다. 구체적으로 식(4)에 나타난 바와 같다.
- [0507] 또한, 본 출원의 실시예에서, p[x]의 값은 통일된 계산 방법으로 획득할 수 있으며, mipSizeId의 값을 판단할 필요가 없다. 이런 경우, 본 출원의 실시예에서, $1 \ll (\text{BitDepth}-1)$ 은 추가 요소로 버퍼 pTemp의 끝에 추가되며, p[x]를 pTemp[x+1]-pTemp[0]와 같게 설정하며, x=0, ..., inSize-1이다.
- [0508] 이와 같이, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확인한 다음에, 진일보로 현재 블록의 MIP 예측값을 확인할 수 있다.
- [0509] S903, MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확인한다.
- [0510] MIP 예측값은 현재 블록의 크로마 컴포넌트에서 부분 샘플의 예측값이다.
- [0511] 시프팅 파라미터(shifting parameter)는 시프팅 오프셋 파라미터(shifting offset parameter)와 시프팅 넘버 파라미터(shifting number parameter)를 포함할 수 있음을 유념해야 한다. 이런 경우, S903에 대하여, MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확인하는 것은 다음 내용을 포함할 수 있다.
- [0512] 시프팅 오프셋 파라미터의 값에 따라 MIP 입력 샘플값의 합과 시프팅 오프셋 파라미터의 곱을 확인한다. 시프팅 오프셋 파라미터의 값은 고정 상수이다.
- [0513] 시프팅 넘버 파라미터의 값에 따라 제 1 상수값을 확인한다. 시프팅 넘버 파라미터의 값은 고정 상수이다.
- [0514] 제 1 오프셋의 값은 제 1 상수값과 상기 곱의 차이로 설정된다.
- [0515] 예측 파라미터에 따라 현재 블록의 MIP 가중 행렬을 확인한다.
- [0516] MIP 가중 행렬, MIP 입력 샘플값, 시프팅 넘버 파라미터 및 제 1 오프셋에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확인한다.
- [0517] 시프팅 오프셋 파라미터는 시프팅 보상 파라미터 또는 오프셋 인자라고도 하며, f0로 표시할 수 있다. 본 출원의 실시예에서, 시프팅 오프셋 파라미터의 값은 32, 46, 56, 66 등과 같은 고정 상수로 설정될 수 있다. 시프팅 오프셋 파라미터의 값은 시프팅 오프셋 파라미터 테이블과 관련되도록 설정될 수 있으며, 시프팅 오프셋 파라미터의 값은 룩업 테이블을 통해 확정될 수 있으며, 이것에 대하여 한정하지 않는다.
- [0518] 시프팅 넘버 파라미터는 시프팅 인자, 시프팅 비트 수, 가중치 시프팅 값 등으로 불리기도 하며, sW, shift, weight shift 등으로 나타낼 수 있다. 본 출원의 실시예에서, 시프팅 넘버 파라미터는 sW로 나타낸다. 본 출원의 실시예에서, 시프팅 넘버 파라미터의 값은 고정 상수, 예를 들어, 5, 6 또는 7 등으로 설정될 수 있다. 시프팅 넘버 파라미터의 값은 또한 시프팅 넘버 파라미터 테이블과 관련되도록 설정될 수 있으며, 시프팅 넘버 파라미터의 값은 룩업 테이블을 통해 확정될 수 있으며, 이것에 대하여 한정하지 않는다.
- [0519] 또한, 제 1 오프셋은 oW로 나타낼 수 있으며, 제 1 오프셋은 시프팅 넘버 파라미터와 시프팅 오프셋 파라미터 모두에 관련된다.
- [0520] 예측 파라미터는 예측 모드 파라미터 및 현재 블록의 사이즈 파라미터를 포함할 수 있음에 유념해야 한다. 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트는 MIP 모드를 사용하여 인트라 예측값을 확인한다는 것을 지시하는 경우, 가중치 행렬 테이블을 미리 설정하여 메모리 또는 저장 유닛에 저장할 수 있다. 메모리 또는 저장 유닛은 디코더에 통합될 수 있거나, 별도로 설치될 수도 있다. 이와 같이, 현재 블록의 블록 사이즈 인덱스 값(mipSizeId)과 MIP 모드 인덱스 값(modeId)에 따라, 룩업 테이블을 통해 현재 블록이 사용해야 하는 MIP 가중 행렬(또는 MIP 가중치 행렬 또는 MIP 행렬이라고도 함)을 확정할 수 있으며, mWeight[x][y]로 나타낸다.
- [0521] 디코더에 있어서, 시프팅 넘버 파라미터 테이블은 미리 설정되어 메모리나 저장 유닛에 저장될 수 있음을 유념해야 한다. 여기서 메모리 또는 저장 유닛은 디코더에 통합될 수 있거나, 별도로 설치될 수도 있다. 본 출원의 실시예에서, 시프팅 넘버 파라미터(sW)는 아래 몇가지 방식으로 확정될 수 있다.

- [0522] 가능한 실시예에서, 상이한 블록 사이즈 인덱스 값 및 상이한 MIP 모드 인덱스 값에 대하여, 시프팅 넘버 파라미터는 상이할 수 있다. 일부 실시예에서, 상기 방법은 또한 다음 내용을 포함할 수 있다.
- [0523] 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정하는 경우, 비트스트림을 파싱하여 현재 블록의 MIP 모드 인덱스 값을 획득한다.
- [0524] MIP 모드 인덱스 값에 따라 제 1 사전 설정 룩업 테이블에서 MIP 모드 인덱스 값에 대응하는 값을 조회하며, 제 1 사전 설정 룩업 테이블은 MIP 모드 인덱스 값과 시프팅 넘버 파라미터의 값 사이의 대응관계를 기록하는 데에 사용된다.
- [0525] 조회된 값을 시프팅 넘버 파라미터의 값으로 확정한다.
- [0526] 시프팅 넘버 파라미터의 값은 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄)과 MIP 모드 인덱스 값(modeId)에 따라 조회될 수 있다. 표 4의 제 1 사전 설정 룩업 테이블에서 서로 다른 mipSizeId 및 modeId에 대하여, 행렬 곱셈에 필요한 시프팅 넘버 파라미터는 룩업 테이블을 통해 확정할 수 있다.
- [0527] 그러나 디코더 측에서는 표 4를 룩업 테이블 형태로 메모리나 저장 유닛에 저장해야 한다. 그러나 저장에는 대가가 필요하며 조회 과정도 마찬가지이다. 표 4의 시프팅 인자는 현재 블록의 블록 사이즈 및 MIP 모드 인덱스 값과 모두 관련이 있기 때문에, 메모리 점유가 증가되고 계산 복잡도도 증가된다.
- [0528] 메모리 점유를 줄이고 계산 복잡도를 줄이기 위해, 본 출원의 실시예에서는 시프팅 인자를 확정하는 방법을 간소화할 수 있다.
- [0529] 다른 가능한 실시예에서, 시프팅 넘버 파라미터의 값은 고정 상수로 설정될 수 있으며, 블록 사이즈 인덱스 값 및 MIP 모드 인덱스 값과 관련이 없는 고정 상수이다. 예를 들어, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 넘버 파라미터의 값은 5로 설정될 수 있다. 또는, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 넘버 파라미터의 값은 6으로 설정될 수 있다. 또는, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 넘버 파라미터의 값은 7로 설정될 수 있다. 본 출원의 실시예에서, 바람직하게는, 시프팅 넘버 파라미터의 값은 6과 동일하게 설정되며, 이것에 대하여 한정하지 않는다.
- [0530] 또 다른 가능한 실시예에서, 시프팅 넘버 파라미터의 값에 대하여, 상기 방법은 아래 내용을 더 포함할 수 있다.
- [0531] 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정한다.
- [0532] 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정한다.
- [0533] 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정할 수 있다. 일부 실시예에서, 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 크기 인덱스 값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0534] 현재 블록의 너비와 높이가 모두 4인 경우, 현재 블록의 블록 사이즈 인덱스 값은 0으로 설정된다.
- [0535] 현재 블록의 너비와 높이가 모두 8이거나, 또는 현재 블록의 너비와 높이 중 하나가 4인 경우, 현재 블록의 블록 사이즈 인덱스 값은 1로 설정된다.
- [0536] 현재 블록의 너비와 높이가 위의 조건을 만족하지 않는 경우, 현재 블록의 블록 사이즈 인덱스 값은 2로 설정된다.
- [0537] 이와 같이, 현재 블록의 블록 사이즈 인덱스 값을 확정된 다음에, 진일보로 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정할 수 있다.
- [0538] 선택적으로, 일부 실시예에서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정하는 것은 아래 내용을 포함할 수 있다.
- [0539] 블록 사이즈 인덱스 값이 각각 0, 1 및 2인 경우, 현재 블록의 블록 사이즈 인덱스 값에 대응하는 시프팅 넘버 파라미터의 값은 각각 5, 6 및 5이다.
- [0540] 선택적으로, 일부 실시예에 있어서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정하는 것은 아래 내용을 포함할 수 있다.

- [0541] 시프팅 넘버 파라미터의 값을 현재 블록의 너비 또는 높이와 현재 블록의 블록 사이즈 인덱스 값에 대응하는 제 1 사전 설정값의 비율과 동일하게 설정한다.
- [0542] 여기서, 제 1 사전 설정값은 현재 블록의 경계로부터 획득된 MIP 입력 샘플값의 수량을 나타낸다. 이런 경우에 상기 방법은 다음 내용을 더 포함할 수 있다.
- [0543] 현재 블록의 블록 사이즈 인덱스 값이 각각 0, 1 및 2인 경우, 현재 블록의 블록 크기 인덱스 값에 대응하는 제 1 사전 설정값은 각각 2, 4 및 4이다.
- [0544] 즉, 제 1 사전 설정값이 현재 블록의 경계로부터 획득된 MIP 입력 샘플값의 수량을 나타낼 때, 현재 블록의 블록 사이즈 인덱스 값이 0인 경우, 대응하는 제 1 사전 설정값은 2와 같고, 현재 블록의 블록 사이즈 인덱스 값이 1인 경우, 대응하는 제 1 사전 설정값은 4와 같고, 현재 블록의 블록 사이즈 인덱스 값이 2인 경우, 대응하는 제 1 사전 설정값은 4와 같으며, 이와 같이 현재 블록의 너비 또는 높이와 대응하는 제 1 사전 설정값의 비율에 따라 시프팅 넘버 파라미터의 값을 확정할 수 있다.
- [0545] 선택적으로, 일부 실시예에서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0546] 시프팅 넘버 파라미터의 값을 현재 블록의 너비 또는 높이와 현재 블록의 블록 사이즈 인덱스 값에 대응하는 제 2 사전 설정값의 비율과 동일하게 설정한다.
- [0547] 여기서, 제 2 사전 설정값은 MIP 가중 행렬을 직접 이용하여 계산하여 획득한 현재 블록의 MIP 예측 블록의 크기를 나타낸다. 이런 경우에 상기 방법은 또한 다음 내용을 포함할 수 있다.
- [0548] 현재 블록의 블록 사이즈 인덱스 값이 각각 0, 1, 2인 경우, 현재 블록의 블록 사이즈 인덱스 값에 대응하는 제 2 사전 설정값은 각각 4, 4, 8이다.
- [0549] 즉, 제 2 사전 설정값이 MIP 가중 행렬을 직접 이용하여 계산하여 획득한 현재 블록의 MIP 예측 블록의 크기를 나타낼 때, 현재 블록의 블록 사이즈 인덱스 값이 0인 경우, 대응하는 제 2 사전 설정값은 4와 같고, 현재 블록의 블록 사이즈 인덱스 값이 1인 경우, 대응하는 제 2 사전 설정값은 4와 같고, 현재 블록의 블록 사이즈 인덱스 값이 2인 경우, 대응하는 제 2 사전 설정값은 8과 같으며, 이와 같이 현재 블록의 너비 또는 높이와 대응하는 제 2 사전 설정값의 비율에 따라 시프팅 넘버 파라미터의 값을 확정할 수 있다.
- [0550] 또 다른 가능한 실시예에서, 시프팅 넘버 파라미터 테이블은 최소화될 수 있고, 여전히 룩업 테이블에 의해 시프팅 넘버 파라미터의 값을 확정한다. 선택적으로, 일부 실시예에서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 넘버 파라미터의 값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0551] 블록 사이즈 인덱스 값에 따라 제 2 사전 설정 룩업 테이블로부터 블록 사이즈 인덱스 값에 대응하는 값을 조회한다. 여기서, 제 2 사전 설정 룩업 테이블은 블록 사이즈 인덱스 값과 시프팅 넘버 파라미터의 값 사이의 대응 관계를 기록하는 데에 사용된다.
- [0552] 조회된 값은 시프팅 넘버 파라미터의 값으로 확정한다.
- [0553] 설명하여야만 하는 것은, 단지 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄)에 따라 시프팅 넘버 파라미터의 값을 조회할 수 있다. 표 5에 예시된 제 2 사전 설정 룩업 테이블에서, 각 블록 사이즈 인덱스 값은 고정된 값에 대응할 수 있다. 즉, 각 블록의 사이즈 또는 각 블록의 사이즈 집합은 표 5와 같이 고정된 시프팅 넘버 파라미터의 값을 가질 수 있다.
- [0554] 표 5에 따르면, 현재 블록의 블록 사이즈 인덱스 값이 각각 0, 1, 2일 때, 블록 사이즈 인덱스 값에 대응하는 시프팅 넘버 파라미터의 값은 각각 5, 6, 5이다.
- [0555] 이렇게 시프팅 넘버 파라미터의 확정 방법을 간소화함으로써, 특히 시프팅 넘버 파라미터 테이블을 최소화하거나, 또는 시프팅 넘버 파라미터 값을 고정함으로써, 최소화 룩업 테이블의 저장을 실현할 수 있다. 따라서, 계산 복잡도를 증가시키지 않고 MIP 모드에서 시프팅 인자 테이블을 저장하기 위해 점유되는 메모리를 감소시킬 수 있다.
- [0556] 디코더에서 시프팅 오프셋 파라미터 테이블을 미리 설정할 수 있으며, 시프팅 오프셋 파라미터 테이블은 메모리 또는 저장 유닛에 저장될 수 있다. 여기서 메모리 또는 저장 유닛은 디코더에 통합될 수 있거나, 별도로 설치될 수도 있다. 본 출원의 실시예에서, 시프팅 오프셋 파라미터(f0)는 다음과 같은 방식으로 확정될 수 있다.

- [0557] 가능한 실시예에서, 상이한 블록 사이즈 인덱스 값 및 상이한 MIP 모드 인덱스 값에 따라, 시프팅 오프셋 파라미터도 상이할 수 있다. 일부 실시예에서, 상기 방법은 또한 다음 내용을 포함할 수 있다.
- [0558] 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정하는 경우, 비트스트림을 파싱하여 현재 블록의 MIP 모드 인덱스 값을 확정한다.
- [0559] MIP 모드 인덱스 값에 따라, 제 3 사전 설정 룩업 테이블로부터 MIP 모드 인덱스 값에 대응하는 값을 조회하고, 제 3 사전 설정 룩업 테이블은 MIP 모드 인덱스 값과 시프팅 오프셋 파라미터의 값 사이의 대응관계를 기록하는 데에 사용된다.
- [0560] 조회된 값은 시프팅 오프셋 파라미터의 값으로 확정된다.
- [0561] 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄) 및 MIP 모드 인덱스 값(modeId)에 따라 시프팅 오프셋 파라미터의 값을 조회할 수 있음에 유의해야 한다. 표 6에 예시된 제 3 사전 설정 룩업 테이블과 같이, 상이한 mipSizeId 및 modeId에 대하여, 행렬 곱셈에 필요한 시프팅 오프셋 파라미터는 룩업 테이블을 통해 확정할 수 있다.
- [0562] 그러나 디코더 측에서는 표 6을 룩업 테이블 형태로 메모리나 저장 유닛에 저장해야 한다. 그러나 저장에는 대가가 필요하며 조회 과정도 마찬가지로 대가가 필요하다. 표 6의 시프팅 오프셋 파라미터의 값은 현재 블록의 블록 사이즈 인덱스 값 및 MIP 모드 인덱스 값과 모두 관련이 있기 때문에, 메모리 점유가 증가되고 계산 복잡도도 증가된다.
- [0563] 메모리 점유를 줄이고 계산 복잡도를 줄이기 위해, 본 출원의 실시예에서는 시프팅 오프셋 파라미터를 확정하는 방법을 간소화할 수 있다.
- [0564] 다른 가능한 실시예에서, 시프팅 오프셋 파라미터의 값은 고정 상수로 설정될 수 있으며, 블록 사이즈 인덱스 값 및 MIP 모드 인덱스 값과 관련이 없는 고정 상수이다. 일반적으로, 고정 상수의 범위는 0~100이다. 예를 들어, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 오프셋 파라미터의 값은 32로 설정될 수 있다. 또는, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 오프셋 파라미터의 값은 46으로 설정될 수 있다. 또는, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 오프셋 파라미터의 값은 56으로 설정될 수 있다. 또는, 서로 다른 블록 사이즈 인덱스 값과 서로 다른 MIP 모드 인덱스 값에 대하여, 시프팅 오프셋 파라미터의 값은 66으로 설정될 수 있다. 본 출원의 실시예에서, 바람직하게는, 시프팅 오프셋 파라미터의 값은 32와 동일하게 설정되며, 이것에 대하여 한정하지 않는다.
- [0565] 또 다른 가능한 실시예에서, 시프팅 오프셋 파라미터의 값에 대하여, 상기 방법은 아래 내용을 더 포함할 수 있다.
- [0566] 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정한다.
- [0567] 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 오프셋 파라미터의 값을 확정한다.
- [0568] 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정할 수 있다. 그 다음에 현재 블록의 블록 사이즈 인덱스 값에 따라 진일보로 시프팅 오프셋 파라미터의 값을 확정할 수 있다.
- [0569] 선택적으로, 일부 실시예에서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 오프셋 파라미터의 값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0570] 블록 사이즈 인덱스 값이 각각 0, 1 및 2인 경우, 현재 블록의 블록 사이즈 인덱스 값에 대응하는 시프팅 오프셋 파라미터의 값은 각각 34, 23 및 46이다.
- [0571] 선택적으로, 일부 실시예에 있어서, 시프팅 오프셋 파라미터 테이블은 최소화될 수 있고, 여전히 룩업 테이블에 의해 시프팅 인자를 확정한다. 선택적으로, 일부 실시예에서, 현재 블록의 블록 사이즈 인덱스 값에 따라 시프팅 오프셋 파라미터의 값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0572] 블록 사이즈 인덱스 값에 따라 제 4 사전 설정 룩업 테이블로부터 블록 사이즈 인덱스 값에 대응하는 값을 조회한다. 여기서, 제 4 사전 설정 룩업 테이블은 블록 사이즈 인덱스 값과 시프팅 오프셋 파라미터의 값 사이의 대응 관계를 기록하는 데에 사용된다.
- [0573] 조회된 값은 시프팅 오프셋 파라미터의 값으로 확정한다.

- [0574] 설명하여야만 하는 것은, 단지 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄)에 따라 시프팅 오프셋 파라미터의 값을 조회할 수 있다. 표 7에 예시된 제 4 사전 설정 룩업 테이블과 같이, 각 블록 사이즈 인덱스 값은 고정된 값에 대응할 수 있다. 즉, 각 블록의 사이즈 또는 각 블록의 사이즈 집합은 표 7에 예시된 것과 같이 고정된 시프팅 오프셋 파라미터의 값을 가질 수 있다.
- [0575] 표 7에 따르면, 현재 블록의 블록 사이즈 인덱스 값이 각각 0, 1, 2일 때, 블록 사이즈 인덱스 값에 대응하는 시프팅 오프셋 파라미터의 값은 각각 34, 23, 46이다.
- [0576] 이렇게 시프팅 오프셋 파라미터의 확정 방법을 간소화함으로써, 특히 시프팅 오프셋 파라미터 테이블을 최소화 하거나, 또는 시프팅 오프셋 파라미터 값을 고정함으로써, 최소화 룩업 테이블의 저장을 실현할 수 있다. 따라서, 계산 복잡도를 증가시키지 않고 MIP 모드에서 시프팅 오프셋 파라미터 테이블을 저장하기 위해 점유되는 메모리를 감소시킬 수 있다.
- [0577] 이와 같이, 시프팅 오프셋 파라미터(f0) 및 시프팅 넘버 파라미터(sW)를 확정한 다음에, 제 1 상수값에 대하여, 선택적으로, 일부 실시예에서, 시프팅 넘버 파라미터의 값에 따라 제 1 상수값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0578] 제 1 상수값을 2의 정수 거듭제곱과 동일하게 설정하며, 거듭제곱의 지수는 시프팅 넘버 파라미터의 값에서 1을 뺀 값과 같다.
- [0579] 선택적으로, 일부 실시예에서, 시프팅 넘버 파라미터의 값에 따라 제 1 상수값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0580] "1"에 대하여 이진 왼쪽 시프팅을 수행하여 제 1 상수값을 획득하고, 왼쪽으로 시프트된 비트의 수는 시프팅 넘버 파라미터의 값에서 1을 뺀 것과 같다.
- [0581] 즉, 시프팅 넘버 파라미터(sW)를 획득한 다음에, 제 1 상수값은 $1 \ll (sW-1)$ 또는 $2^{(sW-1)}$ 로 표시될 수 있다. 시프팅 넘버 파라미터의 값이 6으로 설정될 경우, 제 1 상수값은 32와 같다.
- [0582] 여기서, 제 1 오프셋을 oW로 나타내면, 시프팅 오프셋 파라미터(f0)에 따라 MIP 입력 샘플값의 합과 시프팅 오프셋 파라미터의 곱을 계산하여 획득할 수 있으며, 즉, $f0 \times (\sum_{i=0}^{inSize-1} p[i])$ 이다. oW의 값은 $(1 \ll (sW-1)) - f0 \times (\sum_{i=0}^{inSize-1} p[i])$ 로 설정될 수 있다.
- [0583] 이와 같이, MIP 모드에서 MIP 가중 행렬, MIP 입력 샘플값, 시프팅 넘버 파라미터, 제 1 오프셋을 획득할 수 있으며, 이후에 현재 블록의 MIP 예측값을 확정하는 것이 편리해지도록 한다. 구체적으로, 일부 실시예에서, MIP 가중 행렬, MIP 입력 샘플값, 시프팅 넘버 파라미터 및 제 1 오프셋에 따라 현재 블록의 MIP 예측값을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0584] MIP 가중 행렬과 MIP 입력 샘플값의 제 1 가중치 합을 계산한다.
- [0585] 제 1 가중치 합과 제 1 오프셋의 제 1 합을 계산한다.
- [0586] 제 1 합에 대하여 이진 오른쪽 시프팅(binary right-shifting)을 수행하여 제 1 오른쪽 시프트 값(first right-shift value)을 획득하며, 오른쪽으로 시프트된 비트의 수는 시프팅 넘버 파라미터의 값과 같다.
- [0587] 현재 블록의 MIP 예측값을 제 1 오른쪽 시프트 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 합과 동일하게 설정하며, 제 1 임시 참조값은 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 획득한다.
- [0588] MIP 모드에서, 현재 블록의 블록 사이즈 인덱스 값(mipSizeId로 나타냄)과 MIP 모드 인덱스 값(modeId로 나타냄)에 따라 MIP 가중 행렬(mWeight로 나타냄), 시프팅 넘버 파라미터(sW로 나타냄) 및 시프팅 오프셋 파라미터(f0로 나타냄)를 확정할 수 있다. 그 다음에 MIP 입력 샘플값(p[x]로 나타냄), mWeight, sW 및 f0를 행렬 곱셈에 입력하여 행렬 곱셈에서 출력되는 MIP 예측값(predMip[x][y]로 나타낼 수 있음)을 획득하며, predMip[x][y]의 샘플을 predSize×predSize에 따라 행렬/어레이 형태로 배열한다. predMip[x][y]의 계산식은 식(5) 또는 식(6)에 예시된 바와 같다.
- [0589] S904, MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정한다.

- [0590] MIP 예측 블록은 MIP 예측값으로 구성된다는 점에 유념해야 한다. MIP 예측 블록을 획득한 다음에, MIP 예측 블록의 사이즈 파라미터와 현재 블록의 사이즈 파라미터가 동일한지 여부를 판단함으로써, 현재 블록의 인트라 예측값을 확정할 수 있다. 구체적으로, 판단 결과에 따라 MIP 예측 블록의 사이즈 파라미터와 현재 블록의 사이즈 파라미터가 동일한 경우, 현재 블록의 인트라 예측 블록을 MIP 예측 블록과 동일하게 설정하며, MIP 예측 블록은 현재 블록 중 모든 픽셀 위치의 예측값을 포함한다. MIP 예측 블록의 사이즈 파라미터와 현재 블록의 사이즈 파라미터가 다른 경우, MIP 예측 블록을 필터링하여 필터링된 예측 블록을 획득하고, 필터링된 예측 블록을 현재 블록의 인트라 예측 블록으로 설정한다. 필터링은 업 샘플링 필터링 또는 저역통과 필터링을 포함할 수 있다.
- [0591] S905, LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행함을 지시할 경우, 현재 블록의 재구성된 변환 계수 블록을 확정하고, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득한다.
- [0592] 또한, 임의의 현재 블록이 모두 LFNST를 수행할 수 있는 것이 아니라는 점에 유념해야 하며, 비트스트림에 LFNST 파라미터가 반드시 존재하는 것이 아니라는 것을 의미한다. 이런 경우에 비트스트림에 LFNST 파라미터가 존재하는지 여부를 확정하기 위하여 일부 조건(현재 블록의 사이즈 파라미터 중의 최소값, 현재 블록의 블록 사이즈 인덱스 값 등)을 확인해야 한다. 따라서, 일부 실시예에서, 비트스트림을 파싱하여 LFNST 파라미터를 획득하는 것은 다음 내용을 포함할 수 있다.
- [0593] 비트스트림에 LFNST 파라미터가 존재하는지 여부를 확정한다.
- [0594] 비트스트림에 LFNST 파라미터가 존재한다는 것을 확인한 경우에 비트스트림을 파싱하여 LFNST 파라미터를 획득한다.
- [0595] 즉, 먼저 비트스트림에 LFNST 파라미터가 존재하는지 여부를 확정해야 하며, 확정 결과가 '예'인 경우, 즉 비트스트림에 LFNST 파라미터가 존재하는 경우에만 비트스트림을 파싱하여 LFNST 파라미터를 획득하는 단계를 수행할 수 있다.
- [0596] 선택적으로, 일부 실시예에서, 비트스트림에 LFNST 파라미터가 존재하는지 여부를 확정하는 것은 다음 내용을 포함할 수 있다.
- [0597] 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정하는 경우, 현재 블록의 사이즈 파라미터 중의 최소값을 확정한다.
- [0598] 최소값에 따라 비트스트림에 LFNST 파라미터가 존재하는지 여부를 확정한다.
- [0599] 또한, 최소값에 따라 비트스트림에 LFNST 파라미터가 존재하는지 여부를 확정하는 것은 다음 내용을 포함할 수 있다.
- [0600] 최소값이 제 1 사전 설정 임계값 이상인 경우, 비트스트림에 LFNST 파라미터가 존재한다고 확정한다.
- [0601] 여기서, 제 1 사전 설정 임계값은 LFNST 수행 여부를 평가하기 위한 미리 설정된 임계값을 나타내는 데에 사용될 수 있다. 본 출원의 실시예에서, 제 1 사전 설정된 임계값은 8로 설정될 수 있으며, 이것에 대하여 구체적으로 제한하지 않는다.
- [0602] 즉, 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측을 수행하는 경우, 현재 블록의 사이즈 파라미터 중의 최소값을 확정할 수 있다. 예를 들어, 너비 및 높이의 최소값이 8 이상인 경우, 비트스트림에 LFNST 파라미터가 존재한다고 확정하고, 비트스트림을 파싱하여 `lnst_index`의 값과 같은 LFNST 파라미터를 획득할 수 있다.
- [0603] 선택적으로, 일부 실시예에서, 비트스트림에 LFNST 파라미터가 존재하는지 여부를 확정하는 것은 다음 내용을 포함할 수 있다.
- [0604] 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정하는 경우, 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정한다.
- [0605] 블록 사이즈 인덱스 값에 따라 비트스트림에 LFNST 파라미터가 존재하는지 여부를 확정한다.
- [0606] 또한, 블록 사이즈 인덱스 값에 따라 비트스트림에 LFNST 파라미터가 존재하는지 여부를 확정하는 것은 다음 내용을 포함할 수 있다.

- [0607] 블록 사이즈 인덱스 값이 제 2 사전 설정 임계값과 동일한 경우, 비트스트림에 LFNST 파라미터가 존재한다고 확정한다.
- [0608] 여기서, 제 2 사전 설정 임계값도 LFNST 수행 여부를 평가하기 위한 미리 설정된 임계값을 나타내는 데에 사용될 수 있다. 본 출원의 실시예에서, 제 2 사전 설정 임계값은 사전 설정값(예를 들어, 2) 또는 다수의 사전 설정값(예를 들어, 1, 2) 중 하나로 설정될 수 있으며, 구체적으로 한정하지 않는다.
- [0609] 즉, 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측을 수행하는 경우, 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값(mipSizeId)을 확정할 수 있다. 예를 들어, mipSizeId의 값이 2와 같을 때, 비트스트림에 LFNST 파라미터가 존재한다고 확정하고, 비트스트림을 파싱하여 lfnst_index의 값과 같은 LFNST 파라미터를 획득할 수 있다.
- [0610] 일부 실시예에 있어서, 비트스트림을 파싱하여 LFNST 파라미터를 획득하는 것은 다음 내용을 포함할 수 있다.
- [0611] 비트스트림을 파싱하여 LFNST 인덱스 번호의 값을 획득한다.
- [0612] LFNST 인덱스 번호는 lfnst_index로 나타낼 수 있으며, LFNST 인덱스 번호의 값은 현재 블록에 대하여 LFNST를 수행할지 여부를 확정하는 데에 사용된다는 점에 유념해야 한다. 구체적으로, 일부 실시예에서, 상기 방법은 또한 다음 내용을 포함할 수 있다.
- [0613] LFNST 인덱스 번호의 값은 0보다 큰 경우, 현재 블록에 대하여 LFNST를 수행하기로 확정한다.
- [0614] LFNST 인덱스 번호의 값은 0인 경우, 현재 블록에 대하여 LFNST를 수행하지 않기로 확정한다.
- [0615] lfnst_index의 값을 획득한 다음에, lfnst_index의 값이 0보다 큰 경우, 현재 블록에 대하여 LFNST를 수행하기로 확정하고, lfnst_index의 값이 0인 경우, 현재 블록에 대하여 LFNST를 수행하지 않기로 확정한다.
- [0616] lfnst_index의 값과 같은 LFNST 파라미터를 확인한 다음에, lfnst_index의 값이 0보다 큰 경우, LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행하는 것을 지시함을 나타낸다. 이런 경우에 일부 실시예에서, LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행하는 것을 지시하는 경우, 현재 블록의 재구성된 변환 계수 블록을 확정하는 것은 다음 내용을 포함할 수 있다.
- [0617] 비트스트림을 파싱하여 현재 블록의 양자화된 계수를 획득한다.
- [0618] 양자화된 계수에 대하여 역양자화를 수행하여 현재 블록의 재구성된 변환 계수 블록을 획득한다.
- [0619] 구체적으로 역양자화(inverse quantization)는 바로 스케일링(scaling)이다. 구체적인 예시에서, 양자화된 계수에 대하여 역양자화를 수행하여 현재 블록의 재구성된 변환 계수 블록을 획득하는 것은, 양자화된 계수에 대하여 스케일링하여 현재 블록의 재구성된 변환 계수 블록을 획득한다.
- [0620] 이와 같이, LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행하는 것을 지시하는 경우, 재구성된 변환 계수 블록 중의 전부 또는 일부 재구성된 변환 계수에 대하여 LFNST를 수행하여 제 2 변환 계수 블록을 획득할 수 있다.
- [0621] 또한, 현재 블록에 대하여 LFNST가 수행될 수 있다고 판단되면, 현재 블록에 사용되는 LFNST 변환 커널(kernel로 나타냄)을 확정해야 한다는 점에 유념해야 한다. 일부 실시예에서, 재구성된 변환 계수 블록 중의 적어도 일부 재구성된 변환 계수에 대하여 LFNST를 수행하여 제 2 변환 계수 블록을 획득하는 것은, LFNST 변환 커널을 이용하여 재구성된 변환 계수 블록 중의 적어도 일부 재구성된 변환 계수에 대하여 LFNST를 수행하여 제 2 변환 계수 블록을 획득하는 것을 포함한다.
- [0622] 본 출원의 실시예에서, LFNST에는 4개의 변환 커널 후보 세트가 있고, 4개의 변환 커널 후보 세트는 set0, set1, set2 및 set3을 포함할 수 있다. 여기서, 본 출원의 실시예에서, MIP 파라미터를 이용하여 LFNST 변환 커널 후보 세트를 확정할 수 있고, 그 다음에 LFNST 변환 커널 후보 세트로부터 현재 블록에 사용되는 LFNST 변환 커널을 선택한다. 따라서, 일부 실시예에서, 상기 방법은 다음 내용을 더 포함할 수 있다.
- [0623] 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정하는 경우, 현재 블록의 예측 모드 파라미터로부터 MIP 파라미터를 확정한다.
- [0624] MIP 파라미터에 따라 LFNST 변환 커널 후보 세트를 확정한다.
- [0625] LFNST 변환 커널 후보 세트에서 LFNST 인덱스 번호가 지시하는 변환 커널을 현재 블록에 사용되는 LFNST 변환

커널로 설정한다. LFNST 변환 커널 후보 세트는 미리 설정된 두개 이상의 LFNST 변환 커널을 포함한다.

- [0626] MIP 파라미터는 MIP 모드 인덱스 값(modeId로 나타냄) 및 MIP 전치 지시 파라미터(isTransposed로 나타냄) 중 적어도 하나를 포함할 수 있음을 유념해야 한다. 현재 블록에 사용되는 LFNST 변환 커널에 대하여, MIP 전치 지시 파라미터의 값이 MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치가 수행됨을 나타낼 때, 현재 블록에 사용되는 LFNST 변환 커널을 획득하기 위해 선택한 변환 커널에 대하여 행렬 전치를 수행할 필요가 있다.
- [0627] 또한, 일부 실시예에서, MIP 파라미터에 따라 LFNST 변환 커널 후보 세트를 확정하는 것은 다음 내용을 포함할 수 있다.
- [0628] MIP 모드 인덱스 값에 따라 LFNST 인트라 예측 모드 인덱스 번호의 값을 확정한다.
- [0629] LFNST 인트라 예측 모드 인덱스 번호의 값을 기반으로, 제 5 사전 설정 룩업 테이블을 통해 LFNST 변환 커널 후보 세트의 인덱스 값을 확정한다.
- [0630] LFNST 변환 커널 후보 세트의 인덱스 값에 따라 복수의 후보 LFNST 변환 커널 후보 세트로부터 상기 LFNST 변환 커널 후보 세트를 선택한다.
- [0631] 여기서, LFNST 인트라 예측 모드 인덱스 번호는 predModeIntra로 나타낼 수 있고, LFNST 변환 커널 후보 세트의 인덱스 값은 lfnstTrSetIdx로 나타낼 수 있다.
- [0632] 제 5 사전 설정 룩업 테이블은 표 10에 나타낸 바와 같다. MIP 모드 인덱스 값(즉, modeId의 값)에 따라 predModeIntra의 값을 확정할 수 있다. 그 다음에 predModeIntra의 값에 따라 표 10을 통해 lfnstTrSetIdx의 값을 직접 확정할 수 있다. 즉, 현재 블록에 대해 선택한 LFNST 변환 커널 후보 세트를 확정한다. 여기서, lfnstTrSetIdx의 값은 LFNST에서 사용되는 변환 커널 후보 세트를 나타낸다. modeId의 값은 0, 1, 2, 3, 4 및 5를 포함할 수 있으므로, predModeIntra의 값도 0, 1, 2, 3, 4 또는 5이며, lfnstTrSetIdx와의 대응 관계는 구체적으로 표 10에 나타낸 바와 같다.
- [0633] 구체적인 실시예에서, MIP 모드 인덱스 값에 따라 LFNST 인트라 예측 모드 인덱스 번호의 값을 확정하는 것은, LFNST 인트라 예측 모드 인덱스 번호의 값을 MIP 모드 인덱스 값과 동일하게 설정하는 것을 포함할 수 있다.
- [0634] 즉, 본 출원의 실시예는 predModeIntra의 값을 modeId의 값과 동일하게 설정할 수 있으며, 그 다음에 predModeIntra의 값에 따라 표 10을 통해 lfnstTrSetIdx의 값을 직접 확정한다. 즉 현재 블록에 대해 선택한 LFNST 변환 커널 후보 세트를 확정한다. 또한, 본 출원의 다른 특정 실시예에서, 본 출원의 실시예는 MIP 모드 인덱스 값을 PLANAR 모드의 값으로 직접 매핑한 다음에, PLANAR 모드에 대응하는 predModeIntra의 값을 사용하여 lfnstTrSetIdx의 값을 확정할 수 있다는 점에 유념해야 한다. 즉 현재 블록에 대해 선택한 LFNST 변환 커널 후보 세트를 확정한다.
- [0635] 또한, LFNST를 수행하는 과정에서 스캔 순서는 수평 스캔 순서 및 수직 스캔 순서를 포함할 수 있다. 여기서, 스캔 순서는 MIP 전치 지시 파라미터(isTransposed로 나타냄)의 값과 관련된다.
- [0636] 선택적으로, 일부 실시예에 있어서, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득하는 것은 다음 내용을 포함할 수 있다.
- [0637] MIP 전치 지시 파라미터의 값이 MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하는 것을 지시하는 경우, 수직 스캔 순서에 따라 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득한다.
- [0638] MIP 전치 지시 파라미터의 값이 MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하지 않는 것을 지시하는 경우, 수평 스캔 순서로 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득한다.
- [0639] 구체적으로, isTransposed의 값이 0과 같으면, MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하고, 수평 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할 수 있으며, 구체적으로 식 (10)에 나타낸 바와 같다.
- [0640] isTransposed의 값이 1과 같으면, MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하지 않고, 수직 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할 수 있으며, 구체적으로 식 (11)에 나타낸 바와 같다.

- [0641] 상기 수학식 (9)에 따라 $\log_2 \text{LfnstSize}$ 를 확정한다.
- [0642] 선택적으로, 일부 실시예에서, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득하는 것은 다음 내용을 포함할 수 있다.
- [0643] MIP 전치 지시 파라미터의 값이 MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행함을 지시하는 경우, 수평 스캔 순서로 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득한다.
- [0644] MIP 전치 지시 파라미터의 값이 MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하지 않음을 지시하는 경우, 수직 스캔 순서로 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득한다.
- [0645] 구체적으로, isTransposed 의 값이 1과 같은 경우, MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하고, 수평 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할 수 있으며, 구체적으로 상기 식(10)에 나타난 바와 같다.
- [0646] isTransposed 의 값이 0과 같은 경우, MIP 모드에서 사용되는 샘플 입력 벡터에 대하여 전치 처리를 수행하지 않고, 수직 스캔 순서로 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할 수 있으며, 구체적으로 상기 식(11)에 나타난 바와 같다.
- [0647] 위의 수학식 (9)에 따라 $\log_2 \text{LfnstSize}$ 를 확정한다.
- [0648] S906, 제 2 변환 계수 블록에 대하여 제 1차 변환을 수행함으로써 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 획득한다.
- [0649] S907, 현재 블록의 크로마 컴포넌트의 인트라 예측값 및 재구성된 잔차 블록에 따라 현재 블록의 크로마 컴포넌트의 재구성된 블록을 확정한다.
- [0650] 디코더에서 비트스트림을 파싱함으로써 LFNST 계수의 양자화된 값을 획득할 수 있음에 유념해야 한다. 역양자화 유닛에 의해 양자화된 값에 대하여 역양자화("스케일링(scaling)"이라고 부를 수 있음) 처리를 수행함으로써 재구성된 변환 계수 블록을 획득한다. 재구성된 변환 계수 블록 중의 적어도 일부 재구성된 변환 계수에 대하여 역 LFNST 변환을 수행하여 제 2 변환 계수 블록을 획득할 수 있다. 그리고 역 제 1차 변환을 통해 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 회복할 수 있다. 여기서, 역 제 1차 변환은 인코더에서의 제 1차 변환(또는 코어 변환(core transform))에 대응하는 역변환이다. 표준에서는 디코더의 "역변환"만 정의하므로, 표준에서는 "역 LFNST 변환"을 "LFNST 변환"라고도 할 수 있다.
- [0651] 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 회복한 다음에, 현재 블록의 크로마 컴포넌트의 인트라 예측 값과 재구성된 잔차 블록에 대하여 가산 처리를 수행하여 현재 블록의 크로마 컴포넌트의 재구성된 블록을 확정할 수 있다.
- [0652] 이하, 도 2에 도시된 디코더(200)를 결합하여 본 출원의 실시예의 디코딩 방법에 대하여 구체적으로 설명한다.
- [0653] 디코더(200)의 입력 비트스트림은 인코더(100)에 의해 생성된 비트스트림일 수 있다. 파싱 유닛(201)은 입력된 비트스트림을 파싱하고, 입력된 비트스트림으로부터 선택스 요소의 값을 획득한다. 파싱 유닛(201)은 선택스 요소의 이진 표현을 수치 값으로 변환하고, 수치 값을 디코더(200)의 유닛으로 송신하여 하나 이상의 디코딩된 픽처를 획득한다. 파싱 유닛(201)은 또한 입력 비트스트림으로부터 하나 이상의 선택스 요소를 파싱하여 디코딩된 픽처를 렌더링할 수 있다.
- [0654] 도 10을 참조하면, 본 출원의 실시예에서 제공되는 LFNST 파라미터를 파싱하는 흐름도이다. 도 10에 도시된 바와 같이, 상기 프로세스는 다음 내용을 포함할 수 있다.
- [0655] S1001, 파싱 유닛(201)은 입력 비트스트림에 LFNST 파라미터가 존재하는지 여부를 확인한다.
- [0656] LFNST 파라미터는 lfnst_index 의 값일 수 있음을 유념해야 한다. 여기서, LFNST 파라미터를 나타내는 선택스 구조의 예시는 위의 표 8에 예시된 선택스 구조의 예시와 동일할 수 있고, 위의 표 9A 및 표 9B에 예시된 선택스 구조의 예시와도 동일할 수 있다.
- [0657] 구체적으로, 현재 블록의 인트라 예측 모드가 MIP 모드인 경우, 파싱 유닛(201)은 다음과 같은 방법으로 입력 비트스트림으로부터 lfnst_index 의 코딩 비트를 얻는다. 선택적으로, 표 9A의 선택스 구조를 사용하여, 현재 블

록 중 변환 블록의 너비 및 높이의 최소값이 8보다 크거나 같을 때(즉, 너비 및 높이가 모두 8보다 크거나 같을 때), 파싱 유닛(201)은 MIP 모드를 사용하여 `lfnst_index` 비트를 판독한다. 높이는 8)보다 크거나 같습니다. 또 다른 선택적인 조건은 너비와 높이가 제 1값(예를 들면, 8)보다 크거나 같고, 너비와 높이의 합이 제 2 (예를 들면, 16)보다 크거나 같은 것으로 설정할 수 있다.

- [0658] 선택적으로, 표 9B의 선택스 구조를 사용하여, 현재 블록의 `mipSizeId`가 2일 때, 파싱 유닛(201)은 MIP 모드를 사용하여 `lfnst_index` 비트를 판독한다. 다른 선택적인 조건은 다음과 같이 설정될 수 있다. 현재 블록의 `mipSizeId`는 여러 값(예를 들면, 1 및 2) 중 하나와 같다.
- [0659] S1002, 파싱 유닛(201)은 LFNST 파라미터에 대하여 적응적 이진 산술 디코딩(adaptive binary arithmetic decoding)을 수행하여 LFNST 파라미터를 획득한다.
- [0660] 여기서, 인코더(100) 측에서, 엔트로피 코딩 유닛(115)은 표 9A 및 표 9B의 서술자가 "ae(v)"인 CABAC를 사용하여 LFNST 파라미터(예를 들어, `lfnst_index`의 값)를 인코딩한다. 그 다음에 디코더(200) 측에서 파싱 유닛(201)을 통해 적응적 이진 산술 디코딩을 수행하여 `lfnst_index` 비트로부터 `lfnst_index`의 값을 획득할 수 있다.
- [0661] 파싱 유닛(201)은 선택스 요소의 값 및 선택스 요소의 값에 따라 설정되거나 확정된, 하나 이상의 디코딩된 픽처를 획득하기 위한 하나 이상의 변수를 디코더(200)의 유닛에 송신한다. 예측 유닛(202)은 현재 디코딩 블록(예를 들어, CU)의 인트라 예측 블록을 확정한다. 인트라 디코딩 모드는 현재 디코딩 블록을 디코딩하는 데에 사용된다고 지시되는 경우, 예측 유닛(202)은 파싱 유닛(201)으로부터의 관련 파라미터를 모션 보상 유닛(203)으로 전달하여 인트라 예측 블록을 획득한다. 인트라 예측 모드(MIP 모드 인덱스 값을 기반으로 지시된 MIP 모드를 포함함)는 현재 디코딩 블록을 디코딩하는 데에 사용된다고 지시되는 경우, 예측 유닛(202)은 파싱 유닛(201)으로부터의 관련 파라미터를 인트라 예측 유닛(204)으로 전달하여 인트라 예측 블록을 획득한다.
- [0662] 본 출원의 실시예 중의 현재 블록(또는 "디코딩 블록"이라고 함)은 CU 또는 CU의 파티션(예를 들면, 변환 블록)일 수 있음을 이해해야 한다. 구체적으로, MIP 모드가 현재 블록을 디코딩하는 데에 사용된다고 지시되는 경우, 인트라 예측 유닛(204)은 현재 블록의 인트라 예측 블록을 획득하며, 현재 블록의 인트라 예측 블록은 현재 블록 내의 적어도 하나의 픽셀(샘플)의 인트라 예측값을 포함한다.
- [0663] 구체적으로, 인트라 예측 유닛(204)에 대하여, MIP 모드를 이용하여 인트라 예측 블록을 획득하는 단계는 다음과 같다.
- [0664] 먼저, 인트라 예측 유닛(204)은 현재 블록의 인접 픽셀로부터 하나 이상의 참조 픽셀을 획득한다. 예를 들면, 인접 픽셀에 대하여 다운 샘플링을 수행하여 획득하거나, 또는 직접 인접 픽셀로부터 추출한다.
- [0665] 다음, 인트라 예측 유닛(204)은 획득한 참조 픽셀, MIP 행렬 및 시프팅 파라미터를 이용하여 현재 블록의 픽셀 위치에 대응하는 하나 이상의 부분 예측 픽셀을 확정한다. 여기서, 픽셀 위치는 현재 블록 중의 미리 설정된 픽셀 위치일 수 있다. 예를 들어, 픽셀 위치는 균일한 수평 및 수직 좌표값을 갖는다. 시프팅 파라미터는 시프팅 넘버 파라미터 및 시프팅 오프셋 파라미터를 포함하고, 시프팅 파라미터는 부분 픽셀의 인트라 예측값을 획득하는 과정에서 시프팅 동작에 사용된다.
- [0666] 마지막으로, 현재 블록의 일부 픽셀에 대응하는 예측 픽셀을 획득하면, 인트라 예측 유닛(204)은 현재 블록 중 일부 픽셀을 제외한 나머지 픽셀에 대응하는 예측 픽셀을 획득해야 한다. 예를 들어, 인트라 예측 유닛(204)은 보간 필터를 사용하여 나머지 픽셀에 대응하는 예측 픽셀을 획득하고, 보간 필터의 입력은 부분 픽셀 및 인접 픽셀이다.
- [0667] 도 6에 도시된 바와 같이, 본 출원의 실시예에서 제공되는 MIP 모드를 이용하여 인트라 예측 블록을 획득하는 흐름도이다. 이 절차는 디코더(200)에서 구현될 수 있으며, 여기서 "현재 블록"은 "디코딩 블록"을 지칭한다.
- [0668] 단계 601, 인트라 예측 유닛(204)은 현재 블록의 인접 픽셀을 획득하며, 예를 들면, 인접 픽셀은 도 6의 단계 601의 현재 블록에 인접한 회색 정방향으로 표기된다. 인트라 예측 유닛(204)은 인접 픽셀로부터 하나 이상의 참조 픽셀을 획득한다. 도 6에 도시된 단계 601의 예시에서, 선택적으로, 인트라 예측 유닛(204)은 두개의 이웃 픽셀의 평균치를 계산하고 평균치를 참조 픽셀로 사용할 수 있다. 선택적으로, 인트라 예측 유닛(204)은 2개의 이웃 픽셀마다 하나를 참조 픽셀로 선택한다. 예를 들어, 도 6에 도시된 단계 601의 예시에서, 인트라 예측 유닛(204)은 현재 블록의 8개의 위쪽 인접 픽셀로부터 4개의 참조 픽셀을 선택하고, 현재 블록의 8개의 좌측 인접 픽셀로부터 다른 4개의 참조 픽셀을 선택한다.

- [0669] 단계 601의 구체적인 과정은 다음과 같다.
- [0670] 인트라 예측 유닛(204)은 파싱 유닛(201)으로부터 현재 블록의 너비와 높이를 획득하고, 너비와 높이는 변수 cbWidth 및 cbHeight로 나타낸다. 파싱 유닛(201)으로부터의 파라미터에 따라 현재 블록은 하나 이상의 변환 블록으로 분할된다. 변수 nTbW 및 nTbH를 각각 변환 블록의 너비와 높이로 한다. 현재 블록의 인트라 예측값을 획득하기 위해 MIP 모드를 인트라 예측 모드로 사용하는 경우, 인트라 예측 유닛(204)은 MIP의 블록 사이즈 인덱스 값, 즉 mipSizeId로 나타내는 변수를 확정한다.
- [0671] 선택적으로, 인트라 예측 유닛(204)은 mipSizeId의 값을 다음과 같이 확정한다.
- [0672] - nTbW 및 nTbH가 모두 4인 경우, mipSizeId는 0으로 설정된다.
- [0673] - 그렇지 않고, cbWidth 또는 cbHeight가 4인 경우, mipSizeId는 1로 설정된다.
- [0674] - 그렇지 않으면 mipSizeId는 2로 설정된다.
- [0675] 구체적으로, 현재 블록의 사이즈 파라미터가 8x8일 때(즉, cbWidth와 cbHeight가 모두 8인 경우), mipSizeId는 2로 설정된다.
- [0676] 선택적으로, 인트라 예측 유닛(204)은 mipSizeId의 값을 다음과 같이 확정한다.
- [0677] - nTbW와 nTbH가 모두 4인 경우, mipSizeId는 0으로 설정된다.
- [0678] - 그렇지 않고, cbWidth 또는 cbHeight가 4인 경우, 또는 cbWidth와 cbHeight가 모두 8인 경우, mipSizeId는 1로 설정된다.
- [0679] - 그렇지 않으면 mipSizeId는 2로 설정된다.
- [0680] 구체적으로, 현재 블록의 사이즈 파라미터가 8x8인 경우, mipSizeId는 1로 설정된다.
- [0681] 또한, 인트라 예측 유닛(204)은 mipSizeId에 따라 boundarySize 및 predSize의 값을 획득한다. 구체적으로 다음과 같다.
- [0682] - mipSizeId가 0인 경우, boundarySize는 2로 설정되고, predSize는 4로 설정된다.
- [0683] - 그렇지 않고, mipSizeId가 1인 경우, boundarySize는 4로 설정되고, predSize는 4로 설정된다.
- [0684] - 그렇지 않으면(mipSizeId가 2인 경우), boundarySize는 4로 설정되고, predSize는 8로 설정된다.
- [0685] boundarySize는 현재 블록의 상측변 인접 참조 픽셀과 좌측변 인접 참조 픽셀 각각에서 획득된 참조 픽셀의 수량을 나타낸다.
- [0686] 인트라 예측 유닛(204)은 또한 파싱 유닛(701)으로부터의 버퍼 pTemp에 저장된 참조 픽셀의 순서를 지시하는 파라미터에 따라 MIP 예측값을 계산하는 데에 사용되는 참조 픽셀을 획득할 수 있다. 인트라 예측 유닛(204)은 변수 isTransposed를 버퍼 pTemp에 저장된 참조 픽셀의 순서를 나타내는 파싱 유닛(701)으로부터의 파라미터와 동일하게 설정한다. 예를 들어, isTransposed는 0(또는 FALSE)과 같다는 것은, 인트라 예측 유닛(204)은 먼저 현재 블록의 상측변 인접 참조 픽셀로부터 참조 픽셀을 획득한 다음에 현재 블록의 좌측변 인접 참조 픽셀로부터 참조 픽셀을 획득하는 것을 나타내고, isTransposed는 1(또는 TRUE)과 같다는 것은, 인트라 예측 유닛(204)은 먼저 현재 블록의 좌측변 인접 참조 픽셀로부터 참조 픽셀을 획득한 다음에 현재 블록의 상측변 인접 참조 픽셀로부터 참조 픽셀을 획득하는 것을 나타낸다.
- [0687] 위의 수학적 식 (1)과 같이, 인트라 예측 유닛(204)은 MIP 모드가 사용될 때 사용되는 참조 픽셀의 개수를 나타내기 위해 변수 inSize의 값을 획득한다.
- [0688] 인트라 예측 유닛(204)은 현재 블록의 인접 참조 픽셀을 사용하여 참조 픽셀 그룹(어레이 p[x]로 저장되고, x는 0, ..., inSize-1임)을 획득하기 위해 다음 프로세스를 호출한다.
- [0689] 인트라 예측 유닛(204)은 현재 블록의 상측변 인접 참조 픽셀(예를 들면, 어레이 refT에 저장됨)로부터 nTbW 참조 픽셀을 획득하고, 현재 블록의 좌측변 인접 참조 픽셀(예를 들면, 어레이 refL에 저장됨)로부터 nTbH 참조 픽셀을 획득한다. 인트라 예측 유닛(204)은 refT에 대하여 다운 샘플링을 호출하여 boundarySize 참조 픽셀을 획득하고, boundarySize 참조 픽셀을 refT에 저장한다. 인트라 예측 유닛(204)은 refL에 대하여 다운 샘플링을 호출하여 boundarySize 참조 픽셀을 획득하고, boundarySize 참조 픽셀을 refL에 저장한다.

- [0690] 인트라 예측 유닛(204)은 refT 및 refL 중의 요소를 버퍼 pTemp에 넣기 위해 isTransposed가 지시하는 순서를 획득할 수 있다.
- [0691] 선택적으로, 인트라 예측 유닛(204)은 $p[x]$ 를 획득하며, $x=0, \dots, \text{inSize}-1$ 이다. 다음과 같다.
- [0692] - mipSizeId가 2인 경우, $p[x]=pTemp[x+1]-pTemp[0]$ 이다.
- [0693] 그렇지 않으면(mipSizeId가 2보다 작은 경우), $p[0]=pTemp[0]-(1<<(\text{BitDepth}-1))$, $p[x]=pTemp[x]-pTemp[0]$ 이다.
- [0694] BitDepth는 현재 블록 중 픽셀의 색상 컴포넌트의 비트 깊이이다. 여기서, 색상 컴포넌트는 RGB 컴포넌트 중 하나, YUV 컴포넌트 중 하나 또는 YCbCr 컴포넌트 중 하나이며, 예를 들어, Y 컴포넌트 등이다.
- [0695] 선택적으로, 인트라 예측 유닛(204)은 $p[x]$ 를 획득할 수 있으며, $x=0, \dots, \text{inSize}-1$ 이다. 다음과 같다.
- [0696] - mipSizeId가 2인 경우, $p[x]=pTemp[x+1]-pTemp[0]$ 이다.
- [0697] - 그렇지 않으면(mipSizeId가 2보다 작음), $p[0]=(1<<(\text{BitDepth}-1))-pTemp[0]$, $p[x]=pTemp[x]-pTemp[0]$ 이다.
- [0698] 선택적으로, 인트라 예측 유닛(204)은 통일적인 계산 방법을 사용하여 $p[x]$ 의 값을 획득할 수 있으며, mipSizeId의 값을 판단할 필요가 없다. 예를 들어, 버퍼 pTemp에 $(1<<(\text{BitDepth}-1))$ 을 부가 요소로 추가하며, 이 때 인트라 예측 유닛(204)은 $p[x]$ 를 $pTemp[x+1]-pTemp[0]$ 으로 계산한다.
- [0699] 단계 602에서, 인트라 예측 유닛(204)은 참조 픽셀 그룹과 MIP 가중 행렬을 이용하여 현재 블록의 MIP 예측값을 획득한다. MIP 가중 행렬은 MIP 모드에 대응하는 MIP 모드 인덱스 값(ModeId로 나타냄) 및 MIP의 블록 사이즈 인덱스 값(mipSizeId로 나타냄)에 따라 미리 정의된 MIP 가중 행렬 그룹에서 선택된다. 인트라 예측 유닛(204)은 파싱 유닛(201)으로부터 ModeId의 값을 획득한다.
- [0700] 인트라 예측 유닛(204)은 현재 블록 내의 하나 이상의 픽셀 위치에 대응하는 부분 예측 픽셀의 MIP 예측값(predMip[x][y]로 나타냄)을 획득한다. 도 6에 도시된 단계 602의 예시에서, 부분 예측 픽셀은 현재 블록에서 회색 정방향으로 표기된 픽셀이다. 예측 모듈(601)의 입력은 단계 601에서 획득된 참조 픽셀 $p[x]$ 이다. 예측 모듈(601)은 MIP 가중 행렬 및 시프팅 파라미터를 사용하여 부분 예측 픽셀을 계산한다. 여기서, 시프팅 파라미터는 시프팅 넘버 파라미터와 시프팅 오프셋 파라미터를 포함한다.
- [0701] 가능한 실시예에서, 예측 모듈(601)이 좌표 (x, y)로 픽셀을 나타내는 경우, 예측 픽셀은 predMip[x][y]로 나타내며, predMip[x][y]의 계산 공식은 식 (5)와 식 (6)에 나타낸 바와 같다.
- [0702] 식 (5) 또는 식 (6)에서, mWeight[i][j]는 MIP 가중 행렬이다. 여기서 행렬 요소는 미리 설정된 상수이거나, 또는 훈련 방법을 사용하여 적응적으로 업데이트될 수 있으며, 훈련 방법의 입력은 하나 이상의 코딩된 픽처 또는 블록이고, 또는 외부 장치에 의해 디코더(200)에 제공되는 다른 비트스트림 중의 픽처이고, 또는 파싱 유닛(201)이 MIP 가중 행렬을 포함하는 입력 코드 스트림의 특수 데이터 유닛을 파싱함으로써 획득한 픽처이다. mWeight[i][j]는 파싱 유닛(201)으로부터의 대응하는 하나 이상의 파라미터에 의해 지시되는 MIP 모드에 따라 확정될 수 있다. f0는 oW를 확정하는 데에 사용되는 시프팅 오프셋 파라미터이다. sW는 시프팅 넘버 파라미터이다. p[i]는 참조 픽셀을 사용하여 계산된 것이며, 즉 본 출원의 실시예 중의 MIP 입력 샘플값이다. pTemp[0]은 참조 픽셀 중의 첫번째 픽셀이고, 즉 본 출원의 실시예 중의 제 1 임시 참조값에서 인덱스 번호 0에 대응하는 값을 나타낸다. ">>"는 VVC 중 정의된 이진 오른쪽 시프팅 연산자이다.
- [0703] 예측 모듈(601)은 현재 블록의 사이즈 및 현재 블록에 사용되는 MIP 모드에 따라 sW 및 f0의 값을 확정할 수 있다. 하나의 예시에서, 예측 모듈(601)은 룩업 테이블을 사용하여 sW 및 f0의 값을 획득한다.
- [0704] 선택적으로, 예측 모듈(601)은 위의 표 4를 사용하여 현재 블록의 사이즈 파라미터 및 MIP 모드에 따라 sW를 확정할 수 있다.
- [0705] 선택적으로, 예측 모듈(601)은 또한 위의 표 5를 사용하여 현재 블록의 사이즈 파라미터에 따라 sW를 확정할 수 있다.
- [0706] 선택적으로, 예측 모듈(601)은 sW를 직접 상수값으로 설정할 수 있다. 예를 들어, 다양한 사이즈 파라미터를 갖는 블록 및 다른 MIP 모드에 대하여, 예측 모듈(601)은 sW를 5로 설정할 수 있다. 또는, 다양한 사이즈 파라미터를 갖는 블록 및 다른 MIP 모드에 대하여, 예측 모듈(601)은 sW를 6으로 설정할 수 있다. 또는 다양한 사이즈 파라미터를 갖는 블록 및 다른 MIP 모드에 대하여, 예측 모듈(601)은 sW를 7로 설정할 수 있다.

- [0707] 선택적으로, 예측 모듈(601)은 위의 표 6을 사용하여 현재 블록의 사이즈 파라미터 및 MIP 모드에 따라 f_0 를 확정할 수 있다.
- [0708] 선택적으로, 예측 모듈(601)은 위의 표 7을 사용하여 현재 블록의 사이즈 파라미터에 따라 f_0 를 확정할 수 있다.
- [0709] 선택적으로, 예측 모듈(601)은 f_0 를 직접 상수값(예를 들어, 0 내지 100)으로 설정할 수 있다. 예를 들어, 다양한 사이즈 파라미터를 갖는 블록 및 다른 MIP 모드에 대하여, 예측 모듈(601)은 f_0 를 32로 설정한다. 또는 예측 모듈(601)은 f_0 를 46으로 설정한다. 또는 예측 모듈(601)은 f_0 를 56으로 설정한다. 또는 예측 모듈(601)은 f_0 를 66으로 설정한다.
- [0710] 인트라 예측 유닛(204)은 predMip 내의 MIP 예측값에 대하여 클리핑 조작을 수행할 수 있다. isTransposed가 1 (또는 TRUE)일 때, $\text{predSize} \times \text{predSize}$ 어레이 $\text{predMip}[x][y](x=0, \dots, \text{predSize}-1, y=0, \dots, \text{predSize}-1)$ 는 $\text{predTemp}[y][x]=\text{predMip}[x][y]$ 로 변환되고, 그리고 $\text{predMip}=\text{predTemp}$ 이다.
- [0711] 예시적으로, 선택적으로, 현재 블록의 사이즈 파라미터가 8x8인 경우(즉, cbWidth 및 cbHeight가 모두 8인 경우), 인트라 예측 유닛(204)은 mipSizeId를 2로 확정하고, 8x8 predMip를 획득한다.
- [0712] 선택적으로, 현재 블록의 사이즈 파라미터가 8x8인 경우(즉, cbWidth 및 cbHeight가 모두 8인 경우), 인트라 예측 유닛(204)은 mipSizeId를 1로 확정하고, 4x4 predMip를 획득한다.
- [0713] 단계 603, 인트라 예측 유닛(204)은 현재 블록의 인트라 예측값(어레이에 $\text{predSamples}[x][y]$ 로 저장되며 $x=0, \dots, \text{nTbW}-1, y=0, \dots, \text{nTbH}-1$)을 획득한다. 다음과 같다.
- [0714] - 인트라 예측 유닛(204)이 nTbW 가 predSize 보다 크거나 또는 nTbH 가 predSize 보다 크다고 확정하면, 인트라 예측 유닛(204)은 predMip를 사용하여 predSamples를 획득하기 위해 업 샘플링 프로세스를 호출한다. 인트라 예측 유닛(204)은 현재 블록 중의 일부 픽셀을 제외한 나머지 픽셀에 대응하는 예측 픽셀을 획득한다. 도 6에 도시된 바와 같이, 인트라 예측 유닛(204)은 필터링 모듈(602)을 사용하여 현재 블록 중의 부분 픽셀 이외의 나머지 픽셀에 대응하는 예측 픽셀을 획득할 수 있다. 필터링 모듈(602)의 입력은 단계 602에서 회색 정방향으로 표기된 픽셀이다. 필터링 모듈(602)은 하나 이상의 보간 필터를 사용할 수 있고, 상기 입력을 이용하여 현재 블록 중의 부분 픽셀 이외의 나머지 픽셀에 대응하는 예측 픽셀을 획득한다. 예를 들어, 입력은 참조 픽셀 및 현재 블록 중 하나 이상의 픽셀 위치의 부분 예측 픽셀을 포함한다. 또는, 입력은 이웃 픽셀 및 현재 블록 중 하나 이상의 픽셀 위치의 부분 예측 픽셀을 포함한다. 또는, 입력은 참조 픽셀, 이웃 픽셀 및 현재 블록 중 하나 이상의 픽셀 위치의 부분 예측 픽셀을 포함한다. 예를 들어, 선택적으로, 현재 블록의 사이즈가 8x8일 때(즉, cbWidth 및 cbHeight가 모두 8임), 인트라 예측 유닛(204)은 mipSizeId를 1로 확정하고, 인트라 예측 유닛(204)은 현재 블록의 8x8 인트라 예측 블록을 획득하기 위해 업 샘플링을 4x4 predMip에 응용한다.
- [0715] - 그렇지 않으면, 인트라 예측 유닛(204)은 현재 블록의 인트라 예측 블록을 현재 블록의 MIP 예측 블록과 동일하게 설정한다. 즉, $\text{predSamples}[x][y](x=0, \dots, \text{nTbW}-1, y=0, \dots, \text{nTbH}-1)$ 를 $\text{predMip}[x][y]$ 와 같게 설정한다. 예를 들어, 현재 블록의 사이즈 매개변수가 8x8일 때(즉, cbWidth 및 cbHeight가 모두 8인 경우), 인트라 예측 유닛(204)은 현재 블록의 predSamples를 획득하며, 그 사이즈는 8x8(즉, cbWidth와 cbHeight가 모두 8임)이다.
- [0716] 이와 같이, 도 6에 도시된 절차의 블록도에 따르면, 단계 603 이후에, 인트라 예측 유닛(204)은 현재 블록(즉, CU)의 인트라 예측 블록을 획득할 수 있고, 즉 현재 블록의 적어도 하나의 픽셀의 인트라 예측값을 확정할 수 있다.
- [0717] 스케일링 유닛(205)은 인코더(100)의 역양자화 유닛(110)과 동일한 기능을 갖는다. 스케일링 유닛(205)은 파싱 유닛(201)으로부터의 양자화된 계수(즉, levels)에 대하여 스케일링 조작을 수행하여 재구성된 계수를 획득한다.
- [0718] 변환 유닛(206)은 인코더(100)의 역변환 유닛(111)과 동일한 기능을 갖는다. 변환 유닛(206)은 하나 이상의 변환 조작(즉, 인코더(100)의 역변환 유닛(111)에 의해 수행되는 하나 이상의 변환 조작의 역조작)을 수행하여 재구성된 잔차를 획득한다.
- [0719] lfst_index가 0이 아닌 경우, 변환 유닛(206)은 재구성된 계수에 대하여 2차 변환을 수행하여 2차 변환 계수 블록을 획득한다. 그 다음에 2차 변환 계수 블록에 대하여 최초 DCT 기반으로 설계된 정수 변환과 같은 제 1차 변환을 수행하여 재구성된 잔차 블록을 획득한다. LFNST는 2차 변환의 예시이다.

- [0720] 도 11을 참조하면, 본 출원의 실시예에서 제공되는 LFNST를 수행하는 절차를 도시하는 또 다른 도면이다. 도 11에서, 블록(1101)은 재구성된 변환 계수를 포함하는 재구성된 변환 계수 블록이다.
- [0721] 단계 1101, 변환 유닛(206)은 블록(1101) 중의 재구성된 변환 계수의 전부 또는 일부에 대하여 2차 변환 LFNST를 수행한다.
- [0722] 현재 블록의 인트라 예측 모드가 MIP 모드인 경우, 변환 유닛(206)은 MIP 파라미터에 따라 LFNST 변환 커널 후보 세트를 선택한다. MIP 파라미터는 MIP 모드 인덱스 값(즉, modeId) 및 MIP 전치 지시 파라미터(즉, isTransposed) 중 적어도 하나를 포함할 수 있다.
- [0723] 변환 유닛(206)은 변수 predModeIntra의 값을 modeId의 값과 동일하게 설정한다. 변환 유닛(206)은 서브 블록(11001, 11002, 11003)의 계수에 대하여 LFNST를 수행하기 위한 변환 커널을 확정한다. 서브 블록의 사이즈는 사전 설정값, 예를 들어, 8x8일 수 있다. 변환 유닛(206)은 predModeIntra의 값에 따라 LFNST 변환 커널 후보 세트의 인덱스 값(즉, lfnstTrSetIdx)을 확정한다. MIP 모드에서의 블록 디코딩에 대하여, 변환 유닛(206)은 표 10의 룩업 테이블을 사용하여 lfnstTrSetIdx의 값을 획득할 수 있다. 변환 유닛(206)은 LFNST를 수행하기 위한 변환 커널을 lfnstTrSetIdx가 지시하는 LFNST 변환 커널 후보 세트 중의 lfnst_index가 지시하는 변환 커널로 확정한다.
- [0724] lfnst_index가 0이 아닌 경우, 변환 유닛(206)은 서브 블록(11001, 11002, 11003)으로부터의 계수를 사용하여 변환 블록에 대하여 LFNST를 수행하고, LFNST 계수를 획득하고, LFNST 계수를 블록(1102)의 서브 블록(11011, 11012, 11013)에 배치한다. d[x][y]는 제 1 변환 계수의 블록이라고 가정하고, x=0, ..., nTbW-1, y=0, ..., nTbH-1이며, nTbW 및 nTbH는 각각 블록(1102)의 너비와 높이이다. v[x]는 LFNST 계수의 블록이라고 가정하고, x=0, ..., nLfnstOutSize-1, nLfnstOutSize=(nTbW>=8 && nTbH>= 8)? 48:16이다.
- [0725] 선택적으로, 변환 유닛(206)은 다음 명령을 사용하여 변환 계수의 블록(1102)을 획득하고, 구체적으로 식(8)에 나타낸 바와 같다. log2LfnstSize는 위의 식(9)에 나타낸 바와 같다.
- [0726] 선택적으로, isTransposed가 0과 같을 때, 변환 유닛(206)은 다음 명령을 사용하여 LFNST 계수의 블록(1102)을 획득하고, 구체적으로 식(10)에 나타낸 바와 같다.
- [0727] isTransposed가 1과 같을 때, 변환 유닛(206)은 다음 명령을 사용하여 LFNST 계수의 블록(1102)을 획득하고, 구체적으로 식(11)에 나타낸 바와 같다.
- [0728] 선택적으로, isTransposed가 1과 같을 때, 변환 유닛(206)은 다음 명령을 사용하여 LFNST 계수의 블록(1102)을 획득하고, 구체적으로 식(10)에 나타낸 바와 같다.
- [0729] isTransposed가 0과 같을 때, 변환 유닛(206)은 다음 명령을 사용하여 LFNST 계수의 블록(1102)을 획득하고, 구체적으로 식(11)에 나타낸 바와 같다.
- [0730] log2LfnstSize는 위의 식(9)에 따라 확정된다.
- [0731] 또한, 현재 블록의 인트라 예측 모드가 전통적인 인트라 예측 모드(즉, 각도 인트라 예측, DC 모드 및 PLANAR 모드)인 경우, 변환 유닛(206)은 표 11과 같은 룩업 테이블을 사용하여 전통적인 인트라 예측 모드의 모드 인덱스 값(predModeIntra에 할당됨)에 따라 lfnstTrSetIdx의 값을 확정하고, 즉 현재 블록에 대해 선택된 LFNST 변환 커널 후보 세트를 확정한다. 변환 유닛(206)은 LFNST를 수행하기 위한 변환 커널을 lfnstTrSetIdx가 지시하는 LFNST 변환 커널 후보 세트 중의 lfnst_index가 지시하는 변환 커널로 확정한다.
- [0732] lfnst_index가 0이 아닌 경우, 변환 유닛(206)은 서브 블록(11001, 11002, 11003)으로부터의 계수를 사용하여 변환 블록에 대하여 LFNST를 수행하고, LFNST 계수를 획득하고, LFNST 계수를 블록(1102)의 서브 블록(11011, 11012, 11013)에 배치한다. d[x][y]는 제 1 변환 계수의 블록이라고 가정하고, x=0, ..., nTbW-1, y=0, ..., nTbH-1이며, nTbW 및 nTbH는 각각 블록(1102)의 너비와 높이이다. v[x]는 LFNST 계수의 블록이라고 가정하고, x=0, ..., nLfnstOutSize-1, nLfnstOutSize=(nTbW>=8 && nTbH>= 8)? 48:16이다.
- [0733] 선택적으로, predModeIntra가 34보다 작거나 같을 때, 변환 유닛(206)은 다음 명령을 사용하여 LFNST 계수의 블록(1102)을 획득하고, 구체적으로 식(10)에 나타낸 바와 같다.
- [0734] predModeIntra가 34보다 클 때, 변환 유닛(206)은 다음 명령을 사용하여 LFNST 계수의 블록(1102)을 획득하고, 구체적으로 식(11)에 나타낸 바와 같다.

- [0735] 선택적으로, predModeIntra가 34보다 클 때, 변환 유닛(206)은 다음 명령을 사용하여 LFNST 계수의 블록(1102)을 획득하고, 구체적으로 식(10)에 나타난 바와 같다.
- [0736] predModeIntra가 34보다 작거나 같을 때, 변환 유닛(206)은 다음 명령을 사용하여 LFNST 계수의 블록(1102)을 획득하고, 구체적으로 식(11)에 나타난 바와 같다.
- [0737] log2LfnstSize는 위의 식(9)에 따라 확정된다.
- [0738] 단계 1102, 변환 유닛(206)은 블록(1102) 중의 변환 계수에 대하여 제 1차 변환(즉, 코어 변환), 예를 들어, 원래 DCT를 기반으로 설계된 정수 변환을 수행하여 현재 블록의 재구성된 잔차 블록(1103)을 획득한다.
- [0739] 또한, 가산기(207)는 자신의 입력(예측 유닛(202)으로부터의 인트라 예측 블록과 변환 유닛(206)으로부터의 재구성된 잔차 블록)에 대하여 가산 조작을 수행하여 현재 블록의 재구성된 블록을 획득한다. 재구성된 블록은 또한 예측 유닛(202)으로 송신됨으로써, 인트라 예측 모드에서 인코딩된 다른 블록에 대한 참조로서 사용되도록 한다.
- [0740] 픽처 또는 서브 픽처 내의 모든 CU가 재구성된 후에, 필터링 유닛(208)은 재구성된 픽처 또는 서브 픽처에 대하여 인루프 필터링을 수행한다. 필터링 유닛(208)은 더블록킹 필터, 샘플 적응 오프셋 필터, 적응 루프 필터(Adaptive Loop Filter), 루마 매핑 및 크로마 스케일링 필터(luma mapping with chroma scaling filter) 및 신경망 기반 필터 등과 같은 하나 이상의 필터를 포함한다. 또는, 필터링 유닛(208)이 재구성된 블록이 다른 블록을 디코딩하기 위한 참조로서 사용되지 않는다고 확정할 때, 필터링 유닛(208)은 재구성된 블록 중의 하나 이상의 타겟 픽셀에 대하여 인루프 필터링을 수행한다.
- [0741] 필터링 유닛(208)의 출력은 디코딩된 픽처 또는 서브 픽처이고, 디코딩된 픽처 또는 서브 픽처는 DPB 유닛(209)에 전송된다. DPB 유닛(209)은 타이밍 및 제어 정보에 따라 디코딩된 픽처를 출력한다. DPB부(209)에 저장된 픽처는 또한 예측 유닛(202)에 의한 인트라 예측 또는 인트라 예측을 수행하기 위한 참조로서 사용될 수 있다.
- [0742] 본 출원의 실시예에서, 디코더(200)는 프로세서 및 디코딩 프로그램을 기록하는 저장 매체를 갖는 컴퓨팅 장치일 수 있다. 프로세서가 디코딩 프로그램을 판독하고 실행하면, 디코더(200)는 입력 비트스트림을 읽고 대응하는 디코딩된 비디오를 생성한다. 또한, 디코더(100)는 하나 이상의 칩을 갖는 컴퓨팅 장치일 수 있다. 칩 상의 집적 회로로 구현된 유닛은 도 2의 대응 유닛과 유사한 연결 및 데이터 교환을 통해 유사한 기능을 갖는다.
- [0743] 본 출원의 실시예는 디코딩 방법을 제공한다. 비트스트림을 파싱하여 현재 블록의 예측 파라미터 및 LFNST 파라미터를 획득한다. 예측 파라미터는 예측 모드 파라미터를 포함한다. 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정한다. MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정한다. MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정한다. LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행함을 지시할 경우, 현재 블록의 재구성된 변환 계수 블록을 확정하고, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득한다. 제 2 변환 계수 블록에 대하여 제 1차 변환을 수행함으로써 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 획득한다. 현재 블록의 크로마 컴포넌트의 인트라 예측값 및 재구성된 잔차 블록에 따라 현재 블록의 크로마 컴포넌트의 재구성된 블록을 확정한다. 이와 같이, MIP 모드에 대하여, 디코딩 성능을 보장할 뿐만 아니라 복잡성을 줄이고, 디코딩 과정에 필요한 저장 공간을 줄이며, 디코딩 효율을 효과적으로 향상시킬 수 있다. 또한, LFNST 기술을 MIP 모드 예측에 적용하면, MIP 파라미터의 도입으로 인해 LFNST 변환이 보다 유연해지고 디코딩 효율이 더욱 향상될 수 있다.
- [0744] 본 출원의 또 다른 실시예에서, 상술한 실시예와 동일한 발명의 구상을 기반으로, 도 12를 참조하면, 본 출원의 실시예에서 제공된 인코더(120)의 개략적인 구조도를 예시한다. 도 12에 도시된 바와 같이, 인코더(120)는 제 1 확정 유닛(1201), 제 1 예측 유닛(1202), 제 1 변환 유닛(1203) 및 인코딩 유닛(1204)을 포함한다.
- [0745] 제 1 확정 유닛(1201)은 현재 블록의 예측 파라미터를 확정하도록 구성되고, 예측 파라미터는 예측 모드 파라미터를 포함한다.
- [0746] 제 1 확정 유닛(1201)은 또한 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하도록 구성된다.

- [0747] 제 1 예측 유닛(1202)은 MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하도록 구성되며, 여기서 MIP 예측값은 현재 블록의 크로마 컴포넌트 중 부분 샘플의 예측값이다. 제 1 예측 유닛은 또한 MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하도록 구성된다.
- [0748] 제 1 변환 유닛(1203)은 현재 블록의 크로마 컴포넌트의 인트라 예측값에 따라 현재 블록의 크로마 컴포넌트의 예측 잔차값을 확정하고, 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정하도록 구성된다.
- [0749] 인코딩 유닛(1204)은 LFNST 파라미터를 인코딩하고 비트스트림으로 기입하도록 구성된다.
- [0750] 일부 실시예에 있어서, 현재 블록의 인접 샘플값은 현재 블록의 왼쪽 인접 샘플값 및 현재 블록의 위쪽 인접 샘플값을 포함할 수 있다.
- [0751] 일부 실시예에 있어서, 시프팅 파라미터는 시프팅 오프셋 파라미터와 시프팅 넘버 파라미터를 포함한다.
- [0752] 제 1 확정 유닛(1201)은 구체적으로 시프팅 오프셋 파라미터의 값에 따라 MIP 입력 샘플값의 합과 시프팅 오프셋 파라미터의 곱을 확정하고, 여기서 시프팅 오프셋 파라미터의 값은 고정 상수이며; 시프팅 넘버 파라미터의 값에 따라 제 1 상수값을 확정하고, 여기서 시프팅 넘버 파라미터의 값은 고정 상수이며; 제 1 오프셋의 값을 제 1 상수값과 곱의 차이로 설정하며; 예측 파라미터에 따라 현재 블록의 MIP 가중 행렬을 확정하며; MIP 가중 행렬, MIP 입력 샘플값, 시프팅 넘버 파라미터 및 제 1 오프셋에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하도록 구성된다.
- [0753] 일부 실시예에서, 예측 파라미터는 현재 블록의 사이즈 파라미터를 더 포함한다.
- [0754] 일부 실시예에서, 제 1 확정 유닛(1201)은 또한 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정하고; 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 제 1 임시 참조값을 획득하며; 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우, 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정하며; MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 제 2 상수값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하고; MIP 입력 샘플값 중 인덱스 번호 i 에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 i 에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하며, 여기서 i 는 0보다 큰 정수이고; 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위를 벗어나는 경우, MIP 입력 샘플값 중 인덱스 번호 j 에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 $(j+1)$ 에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하도록 구성되며, j 는 0 이상의 정수이다.
- [0755] 일부 실시예에서, 제 1 확정 유닛(1201)은 또한 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우에 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정하고; MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값과 제 2 상수값의 차이로 설정하며; MIP 입력 샘플값 중 인덱스 번호 i 에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 i 에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하도록 구성되며, 여기서 i 는 0보다 큰 정수이다.
- [0756] 일부 실시예에서, 제 1 확정 유닛(1201)은 또한 현재 블록의 너비와 높이가 모두 4인 경우에 현재 블록의 블록 사이즈 인덱스 값을 0으로 설정하고; 현재 블록의 너비와 높이가 모두 8이거나, 또는 현재 블록의 너비와 높이 중 하나가 4인 경우에 현재 블록의 블록 사이즈 인덱스 값을 1로 설정하며; 현재 블록의 너비와 높이가 위의 조건을 만족하지 않는 경우에 현재 블록의 블록 사이즈 인덱스 값을 2로 설정하도록 구성된다.
- [0757] 일부 실시예에서, 제 1 확정 유닛(1201)은 또한 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 제 1 임시 참조값을 획득하고; 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정하며, 제 2 상수값을 제 1 임시 참조값 후의 데이터 유닛에 버퍼링하여 제 2 임시 참조값을 획득하며; MIP 입력 샘플값 중 인덱스 번호 j 에 대응하는 값을 제 2 임시 참조값 중 인덱스 번호 $(j+1)$ 에 대응하는 값과 제 2 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하도록 구성되며, j 는 0 이상의 정수이다.
- [0758] 일부 실시예에서, 제 1 확정 유닛(1201)은 또한 제 2 상수값을 2의 정수 거듭제곱과 동일하게 설정하도록 구성되며, 거듭제곱의 지수는 현재 블록의 인접 샘플값의 비트 깊이에서 1을 뺀 값과 같다.
- [0759] 일부 실시예에서, 제 1 확정 유닛(1201)은 또한 "1"에 대하여 이진 왼쪽 시프팅을 수행하여 제 2 상수값을 획득

하도록 구성되고, 왼쪽으로 시프트된 비트의 수는 현재 블록의 인접 샘플값의 비트 깊이에서 1을 뺀 것과 같다.

- [0760] 일부 실시예에서, 시프팅 넘버 파라미터의 값은 블록 사이즈 인덱스 값 및 MIP 모드 인덱스 값과 관련이 없는 고정 상수이다.
- [0761] 일부 실시예에서, 시프팅 넘버 파라미터의 값은 6으로 설정된다.
- [0762] 일부 실시예에서, 시프팅 오프셋 파라미터의 값은 블록 사이즈 인덱스 값 및 MIP 모드 인덱스 값과 관련이 없는 고정 상수이다.
- [0763] 일부 실시예에서, 시프팅 오프셋 파라미터의 값은 32로 설정된다.
- [0764] 일부 실시예에서, 제 1 확정 유닛(1201)은 또한 제 1 상수값을 2의 정수 거듭제곱과 동일하게 설정하도록 구성되며, 거듭제곱의 지수는 시프팅 넘버 파라미터의 값에서 1을 뺀 값과 같다.
- [0765] 일부 실시예에서, 제 1 확정 유닛(1201)은 또한 "1"에 대하여 이진 왼쪽 시프팅을 수행하여 제 1 상수값을 획득하도록 구성되고, 왼쪽으로 시프트된 비트의 수는 시프팅 넘버 파라미터의 값에서 1을 뺀 것과 같다.
- [0766] 일부 실시예에서, 도 12를 참조하면, 인코더(120)는 제 1 계산 유닛(1205)을 더 포함할 수 있으며, MIP 가중 행렬과 MIP 입력 샘플값의 제 1 가중 합을 계산하고, 제 1 가중 합과 제 1 오프셋의 제 1 합을 계산하며, 제 1 합에 대하여 이진 오른쪽 시프팅(binary right-shifting)을 수행하여 제 1 오른쪽 시프트 값(first right-shift value)을 획득하도록 구성되고, 오른쪽으로 시프트된 비트의 수는 시프팅 넘버 파라미터의 값과 같다.
- [0767] 제 1 확정 유닛(1201)은 또한 현재 블록의 크로마 컴포넌트의 MIP 예측값을 제 1 오른쪽 시프트 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 합과 동일하게 설정하도록 구성되며, 제 1 임시 참조값은 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 획득한다.
- [0768] 일부 실시예에서, 제 1 변환 유닛(1203)은 예측 잔차값에 따라 현재 블록의 잔차 블록을 확정하고, 잔차 블록에 대하여 제 1차 변환을 수행하여 제 1 변환 계수 블록을 획득하도록 구성된다.
- [0769] 제 1 확정 유닛(1201)은 또한 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행할지 여부를 확정하도록 구성된다.
- [0770] 제 1 변환 유닛(1203)은 또한 확정 결과가 예인 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 파라미터를 확정하도록 구성된다.
- [0771] 일부 실시예에서, 제 1 확정 유닛(1201)은 또한 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정하는 경우, 현재 블록의 사이즈 파라미터 중의 최소값을 확정하고, 최소값이 제 1 사전 설정 임계값 이상인 경우, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정하도록 구성된다.
- [0772] 일부 실시예에서, LFNST 파라미터는 LFNST 인덱스 번호를 포함하고, 제 1 확정 유닛(1201)은 또한 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하기로 확정한 경우, LFNST 인덱스 번호의 값은 0보다 크게 설정되고, 제 1 변환 계수 블록 중의 적어도 일부 변환 계수에 대하여 LFNST를 수행하지 않기로 확정한 경우, LFNST 인덱스 번호의 값은 0으로 설정된다.
- [0773] 인코딩 유닛(1204)은 LFNST 인덱스 번호를 인코딩하고 비트스트림으로 기입하도록 구성된다.
- [0774] 일부 실시예에 있어서, LFNST 파라미터는 LFNST 계수를 더 포함한다. 도 12에 도시된 바와 같이, 인코더(120)는 양자화 유닛(1206)을 더 포함한다.
- [0775] 제 1 변환 유닛(1203)은 제 1 변환 계수 블록 중 적어도 일부 변환 계수에 대하여 LFNST를 수행하여 LFNST 계수를 획득하도록 구성된다.
- [0776] 양자화 유닛(1206)은 LFNST 계수를 양자화하여 양자화 계수를 획득하도록 구성된다.
- [0777] 인코딩 유닛(1204)은 양자화 계수를 인코딩하고 비트스트림으로 기입하도록 구성된다.
- [0778] 본 출원의 실시예에서, "유닛"은 회로의 일부, 프로세서의 일부, 프로그램의 일부 또는 소프트웨어 등일 수 있으며, 물론 모듈일 수도 있고, 비모듈일 수도 있음을 이해할 수 있다. 또한, 본 출원의 실시예의 각 컴포넌트는 하나의 처리 유닛에 통합될 수 있고, 또는 각 유닛이 단독으로 물리적으로 존재할 수도 있으며, 또는 2개 이상의 유닛이 하나의 유닛에 통합될 수도 있다. 상기 통합 유닛은 하드웨어 또는 소프트웨어 기능 모듈의 형태로

실현될 수 있다.

- [0779] 통합된 유닛이 소프트웨어 기능 유닛 형태로 구현되어 별도의 제품으로 판매되거나 사용되는 경우, 컴퓨터 관독 가능 저장 매체에 저장될 수 있다. 이러한 이해를 기초로 하여 본 발명의 기술방안의 본질, 혹은 기술 분야에 기여하는 부분 또는 기술적 방안의 전부 또는 일부를 소프트웨어 제품의 형태로 구현될 수 있다. 해당 컴퓨터 소프트웨어 제품은 하나의 저장 매체에 저장되며, 컴퓨터 장치(개인용 컴퓨터, 서버 또는 네트워크 장치일 수 있음) 또는 프로세서가 본 발명의 각 실시예에 따른 방법의 전부 또는 일부를 실행할 수 있도록 하는 다수의 명령어를 포함한다. 전술한 저장 매체는 USB, 외장 하드, 읽기 전용 기억 장치(Read-Only Memory, ROM), 랜덤 액세스 메모리(Random Access Memory, RAM), 디스크 또는 광 디스크 등과 같은 프로그램 코드를 저장할 수 있는 매체를 포함한다.
- [0780] 본 출원의 실시예는 인코더(120)에 적용되는 컴퓨터 저장 매체를 제공한다. 컴퓨터 저장 매체에 컴퓨터 프로그램이 저장되어 있다. 컴퓨터 프로그램이 제 1 프로세서에 의해 실행될 때에 상술한 실시예 중의 임의의 방법을 실현한다.
- [0781] 인코더(120)의 구성 및 컴퓨터 저장 매체에 따라, 도 13을 참조하면, 도 13은 본 출원의 실시예에 따른 인코더(120)의 구체적인 하드웨어 구조를 나타내는 도면이다. 도 13에 도시된 바와 같이, 인코더(120)는 제 1 통신 인터페이스(1301), 제 1 메모리(1302) 및 제 1 프로세서(1303)를 포함할 수 있다. 각 컴포넌트는 제 1 버스 시스템(1304)을 통해 함께 결합된다. 제 1 버스 시스템(1304)은 이러한 컴포넌트 간의 연결과 통신을 실현하는 데에 사용된다. 제 1 버스 시스템(1304)은 데이터 버스 이외에 또한 전력 버스, 제어 버스 및 상태 신호 버스를 포함한다. 명확하게 설명하기 위하여, 도 13에 도시된 바와 같이, 다양한 버스는 제 1 버스 시스템(1304)으로 표기되어 있다.
- [0782] 제 1 통신 인터페이스(1301)는 다른 외부 네트워크 요소와 정보를 송수신하는 과정에서 신호를 송수신하는 데에 사용된다.
- [0783] 제 1 메모리(1302)는 제 1 프로세서(1303)에 의해 실행 가능한 컴퓨터 프로그램을 저장하는 데에 사용된다.
- [0784] 제 1 프로세서(1303)는 컴퓨터 프로그램을 실행할 때 다음을 실행하도록 구성된다.
- [0785] 현재 블록의 예측 파라미터를 확정하며, 예측 파라미터는 예측 모드 파라미터를 포함한다. 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정한다. MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하며, MIP 예측값은 현재 블록의 크로마 컴포넌트 중 일부 샘플의 예측값이다. MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정한다. 현재 블록의 크로마 컴포넌트의 인트라 예측값에 따라 현재 블록의 크로마 컴포넌트의 예측 잔차값을 확정한다. 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정한다. LFNST 파라미터를 인코딩하고 비트스트림으로 기입한다.
- [0786] 본 출원의 실시예의 제 1 메모리(1302)는 휘발성 메모리 또는 비 휘발성 메모리일 수 있으며, 또는 휘발성 메모리 및 비 휘발성 메모리 양자를 포함할 수 있다. 비 휘발성 메모리는 읽기 전용 메모리(Read-Only Memory, ROM), 프로그램 가능 읽기 전용 메모리(Programmable Read-Only Memory, PROM), 삭제 가능 프로그램 가능 읽기 전용 메모리(Erasable Programmable Read-Only Memory, EPROM), 전기적 삭제 가능 프로그램 가능 읽기 전용 메모리(Electrically Erasable Programmable Read-Only Memory, EEPROM) 또는 플래시 메모리(Flash Memory)일 수 있다. 휘발성 메모리는 외부 고속 캐시로 사용되는 랜덤 액세스 메모리(Random Access Memory, RAM)일 수 있다. 예시적이지만 한정적이지 않은 예를 들어, 다양한 형태의 RAM을 사용할 수 있으며, 예를 들어, 정적 랜덤 액세스 메모리(Static RAM, SRAM), 동적 랜덤 액세스 메모리(Dynamic Random Access Memory, DRAM), 동기 동적 랜덤 액세스 메모리(Synchronous Dynamic Random Access Memory, SDRAM), 더블 데이터 레이트 동기 동적 랜덤 액세스 메모리(Double Data Rate Synchronous Dynamic Random Access Memory, DDR SDRAM), 강화 동기 동적 랜덤 액세스 메모리(Enhanced Synchronous Dynamic Random Access Memory, ESDRAM), 동기 링크 동적 랜덤 액세스 메모리(Synch-link Dynamic Random Access Memory, SLDRAM) 및 직접 램버스 랜덤 액세스 메모리(Direct Rambus Random Access Memory, DRRAM)이다. 본 명세서에 기재된 시스템 및 방법의 제 1 메모리(1302)는 이들과 다른 임의의 적절한 유형의 메모리를 포함하지만, 이것에 한정되지 않는다.
- [0787] 제 1 프로세서(1303)는 신호 처리 능력을 갖는 집적 회로 칩일 수 있다. 실시 과정에서 상술한 방법 실시예의

각 단계는 제 1 프로세서(1303)의 하드웨어 형태의 집적 논리 회로(integrated logic circuit) 또는 소프트웨어 형태의 명령에 의해 완성될 수 있다. 제 1 프로세서(1303)는 범용 프로세서, 디지털 신호 프로세서(Digital Signal Processor, DSP), 주문형 집적 회로(Application Specific Integrated Circuit, ASIC), 필드 프로그래머블 게이트 어레이(Field Programmable Gate Array, FPGA) 또는 다른 프로그래머블 로직 디바이스, 이산 게이트 또는 트랜지스터 논리 장치, 개별 하드웨어 컴포넌트일 수 있다. 프로세서는 본 발명의 실시예에 개시된 방법, 단계 및 논리 블록도를 실현 또는 실행할 수 있다. 범용 프로세서는 마이크로 프로세서 또는 임의의 통상적인 프로세서 등일 수 있다. 본 발명의 실시예에서 개시된 방법의 단계는 직접 하드웨어 디코딩 프로세서에 의해 실행 및 완성될 수 있거나, 또는 디코딩 프로세서의 하드웨어 및 소프트웨어 모듈의 조합에 의해 실행 및 완성될 수 있다. 소프트웨어 모듈은 랜덤 액세스 메모리(random access memory, RAM), 플래시 메모리, 읽기 전용 메모리(read only memory, ROM), 프로그래머블 읽기 전용 메모리(programmable ROM, PROM) 또는 전기적으로 지울 수 있는 프로그래머블 메모리, 레지스터 등 본 기술 분야의 성숙한 저장 매체에 있을 수 있다. 저장 매체는 제 1 메모리(1302)에 있다. 제 1 프로세서(1303)는 제 1 메모리(1302)의 정보를 읽고 프로세서의 하드웨어를 사용하여 상술한 방법의 단계를 완성한다.

[0788] 본 명세서에 기재된 실시예는 하드웨어, 소프트웨어, 펌웨어, 미들웨어, 마이크로 코드 또는 그 조합에 의해 실현될 수 있음을 이해할 수 있다. 하드웨어에 의해 실현되는 경우, 처리 유닛은 하나 이상의 주문형 집적 회로(Application Specific Integrated Circuits, ASIC), 디지털 신호 처리(Digital Signal Processing, DSP), DSPD(DSP Device), 프로그래머블 로직 장치(Programmable Logic Device, PLD), 필드 프로그래머블 게이트 어레이(Field-Programmable Gate Array, FPGA), 범용 프로세서, 컨트롤러, 마이크로 컨트롤러, 마이크로 프로세서, 본 출원의 상기 기능을 실행하는 다른 전자 유닛 또는 그 조합으로 구현될 수 있다. 소프트웨어에 의해 실현되는 경우, 본 명세서에 기재된 기술은 본 명세서에 기재된 기능을 수행하는 모듈(예를 들어, 절차, 함수(functions) 등)에 의해 실현될 수 있다. 소프트웨어 코드는 메모리에 저장되고, 또한 프로세서에 의해 실행된다. 메모리는 프로세서 내부 또는 프로세서 외부에서 구현될 수 있다.

[0789] 선택적으로, 다른 실시예로서, 제 1 프로세서(1303)는 컴퓨터 프로그램을 실행하면 상기 실시예에 기재된 임의의 하나의 방법을 실행하는 데에 사용된다.

[0790] 본 출원의 실시예는 인코더를 제공한다. 인코더는 제 1 확정 유닛, 제 1 예측 유닛, 제 1 변환 유닛 및 인코딩 유닛을 포함할 수 있다. 이와 같이, MIP 모드에 대하여, 인코딩 성능을 보장할 뿐만 아니라 복잡성을 줄일 수 있고, 인코딩 프로세스에 필요한 저장 공간을 줄일 수 있으며, 인코딩 효율을 효과적으로 향상시킬 수 있다. 또한, LFNST 기술이 MIP 모드 예측에 적용될 경우, MIP 파라미터의 도입으로 인해 LFNST 변환이 더욱 유연해질 수 있고, 인코딩 효율이 더욱 향상될 수 있다.

[0791] 본 출원의 또 다른 실시예에서, 상술한 실시예와 동일한 발명의 구상을 기반으로, 도 14를 참조하면, 본 출원의 실시예에서 제공된 디코더(140)의 개략적인 구조도를 예시한다. 도 14에 도시된 바와 같이, 디코더(140)는 파싱 유닛(1401), 제 2 확정 유닛(1402), 제 2 예측 유닛(1403), 제 2 변환 유닛(1404)을 포함한다.

[0792] 파싱 유닛(1401)은 비트스트림을 파싱하여 현재 블록의 예측 파라미터 및 LFNST 파라미터를 획득하도록 구성되고, 예측 파라미터는 예측 모드 파라미터를 포함한다.

[0793] 제 2 확정 유닛(1402)은 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하도록 구성된다.

[0794] 제 2 예측 유닛(1403)은 MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하고, 및 MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하도록 구성되며, MIP 예측값은 현재 블록의 크로마 컴포넌트 중 일부 샘플의 예측값이다.

[0795] 제 2 변환 유닛(1404)은 LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행함을 지시할 경우, 현재 블록의 재구성된 변환 계수 블록을 확정하고, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득하고, 제 2 변환 계수 블록에 대하여 제 1차 변환을 수행함으로써 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 획득하도록 구성된다.

[0796] 제 2 확정 유닛(1402)은 또한 현재 블록의 크로마 컴포넌트의 인트라 예측값 및 재구성된 잔차 블록에 따라 현재 블록의 크로마 컴포넌트의 재구성된 블록을 확정하도록 구성된다.

[0797] 일부 실시예에 있어서, 현재 블록의 인접 샘플값은 현재 블록의 왼쪽 인접 샘플값 및 현재 블록의 위쪽 인접 샘플값

플값을 포함한다.

- [0798] 일부 실시예에 있어서, 시프팅 파라미터는 시프팅 오프셋 파라미터와 시프팅 넘버 파라미터를 포함한다.
- [0799] 제 2 확정 유닛(1402)은 구체적으로 시프팅 오프셋 파라미터의 값에 따라 MIP 입력 샘플값의 합과 시프팅 오프셋 파라미터의 곱을 확정하고, 여기서 시프팅 오프셋 파라미터의 값은 고정 상수이며; 시프팅 넘버 파라미터의 값에 따라 제 1 상수값을 확정하고, 여기서 시프팅 넘버 파라미터의 값은 고정 상수이며; 제 1 오프셋의 값을 제 1 상수값과 곱의 차이로 설정하며; 예측 파라미터에 따라 현재 블록의 MIP 가중 행렬을 확정하며; MIP 가중 행렬, MIP 입력 샘플값, 시프팅 넘버 파라미터 및 제 1 오프셋에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측 값을 확정하도록 구성된다.
- [0800] 일부 실시예에서, 예측 파라미터는 현재 블록의 사이즈 파라미터를 더 포함한다.
- [0801] 일부 실시예에서, 제 2 확정 유닛(1402)은 또한 현재 블록의 사이즈 파라미터에 따라 현재 블록의 블록 사이즈 인덱스 값을 확정하고; 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 제 1 임시 참조값을 획득하며; 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우, 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정하며; MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 제 2 상수값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하고; MIP 입력 샘플값 중 인덱스 번호 i 에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 i 에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하며, 여기서 i 는 0보다 큰 정수이고; 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위를 벗어나는 경우, MIP 입력 샘플값 중 인덱스 번호 j 에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 $(j+1)$ 에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하도록 구성되며, j 는 0 이상의 정수이다.
- [0802] 일부 실시예에서, 제 2 확정 유닛(1402)은 또한 현재 블록의 블록 사이즈 인덱스 값이 미리 설정된 범위 내에 있는 경우에 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정하고; MIP 입력 샘플값 중 인덱스 번호 0에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값과 제 2 상수값의 차이로 설정하며; MIP 입력 샘플값 중 인덱스 번호 i 에 대응하는 값을 제 1 임시 참조값 중 인덱스 번호 i 에 대응하는 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하도록 구성되며, 여기서 i 는 0보다 큰 정수이다.
- [0803] 일부 실시예에서, 제 2 확정 유닛(1402)은 또한 현재 블록의 너비와 높이가 모두 4인 경우에 현재 블록의 블록 사이즈 인덱스 값을 0으로 설정하고; 현재 블록의 너비와 높이가 모두 8이거나, 또는 현재 블록의 너비와 높이 중 하나가 4인 경우에 현재 블록의 블록 사이즈 인덱스 값을 1로 설정하며; 현재 블록의 너비와 높이가 위의 조건을 만족하지 않는 경우에 현재 블록의 블록 사이즈 인덱스 값은 2로 설정하도록 구성된다.
- [0804] 일부 실시예에서, 제 2 확정 유닛(1402)은 또한 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 제 1 임시 참조값을 획득하고; 현재 블록의 인접 샘플값의 비트 깊이에 따라 제 2 상수값을 확정하며, 제 2 상수값을 제 1 임시 참조값 후의 데이터 유닛에 버퍼링하여 제 2 임시 참조값을 획득하며; MIP 입력 샘플값 중 인덱스 번호 j 에 대응하는 값을 제 2 임시 참조값 중 인덱스 번호 $(j+1)$ 에 대응하는 값과 제 2 임시 참조값 중 인덱스 번호 0에 대응하는 값의 차이로 설정하도록 구성되며, j 는 0 이상의 정수이다.
- [0805] 일부 실시예에서, 제 2 확정 유닛(1402)은 또한 제 2 상수값을 2의 정수 거듭제곱과 동일하게 설정하도록 구성되며, 거듭제곱의 지수는 현재 블록의 인접 샘플값의 비트 깊이에서 1을 뺀 값과 같다.
- [0806] 일부 실시예에서, 제 2 확정 유닛(1402)은 또한 "1"에 대하여 이진 왼쪽 시프팅을 수행하여 제 2 상수값을 획득하도록 구성되고, 왼쪽으로 시프트된 비트의 수는 현재 블록의 인접 샘플값의 비트 깊이에서 1을 뺀 것과 같다.
- [0807] 일부 실시예에서, 시프팅 넘버 파라미터의 값은 블록 사이즈 인덱스 값 및 MIP 모드 인덱스 값과 관련이 없는 고정 상수이다.
- [0808] 일부 실시예에서, 시프팅 넘버 파라미터의 값은 6으로 설정된다.
- [0809] 일부 실시예에서, 시프팅 오프셋 파라미터의 값은 블록 사이즈 인덱스 값 및 MIP 모드 인덱스 값과 관련이 없는 고정 상수이다.
- [0810] 일부 실시예에서, 시프팅 오프셋 파라미터의 값은 32로 설정된다.
- [0811] 일부 실시예에서, 제 2 확정 유닛(1402)은 또한 제 1 상수값을 2의 정수 거듭제곱과 동일하게 설정하도록 구성되며, 거듭제곱의 지수는 시프팅 넘버 파라미터의 값에서 1을 뺀 값과 같다.

- [0812] 일부 실시예에서, 제 2 확정 유닛(1402)은 또한 "1"에 대하여 이진 왼쪽 시프팅을 수행하여 제 1 상수값을 획득하도록 구성되고, 왼쪽으로 시프트된 비트의 수는 시프팅 넘버 파라미터의 값에서 1을 뺀 것과 같다.
- [0813] 일부 실시예에 있어서, 도 14를 참조하면, 디코더(140)는 제 2 계산 유닛(1405)을 더 포함할 수 있으며, MIP 가장 행렬과 MIP 입력 샘플값의 제 1 가장 합을 계산하고, 제 1 가장 합과 제 1 오프셋의 제 1 합을 계산하며, 제 1 합에 대하여 이진 오른쪽 시프팅(binary right-shifting)을 수행하여 제 1 오른쪽 시프트 값(first right-shift value)을 획득하도록 구성되고, 오른쪽으로 시프트된 비트의 수는 시프팅 넘버 파라미터의 값과 같다.
- [0814] 제 2 확정 유닛(1402)은 또한 현재 블록의 크로마 컴포넌트의 MIP 예측값을 제 1 오른쪽 시프트 값과 제 1 임시 참조값 중 인덱스 번호 0에 대응하는 값의 합과 동일하게 설정하도록 구성되며, 제 1 임시 참조값은 현재 블록의 인접 샘플값에 대하여 다운 샘플링 필터링을 수행하여 획득한다.
- [0815] 일부 실시예에 있어서, 제 2 확정 유닛(1402)은 또한 비트스트림에 LFNST 파라미터가 존재하는지 여부를 결정하도록 구성된다.
- [0816] 파싱 유닛(1401)은 또한 비트스트림에 LFNST 파라미터가 존재한다는 것을 확인한 경우에 비트스트림을 파싱하여 LFNST 파라미터를 획득하도록 구성된다.
- [0817] 일부 실시예에 있어서, 제 2 확정 유닛(1402)은 또한 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 결정하는 경우, 현재 블록의 사이즈 파라미터 중의 최소값을 확인하고, 최소값이 제 1 사전 설정 임계값 이상인 경우, 비트스트림에 LFNST 파라미터가 존재한다고 결정하도록 구성된다.
- [0818] 일부 실시예에 있어서, 도 14를 참조하면, 디코더(140)는 역양자화 유닛(1406)을 더 포함한다.
- [0819] 파싱 유닛(1401)은 또한 LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행하는 것을 지시하는 경우, 비트스트림을 파싱하여 현재 블록의 양자화된 계수를 획득하도록 구성된다.
- [0820] 역양자화 유닛(1406)은 양자화된 계수에 대하여 역양자화를 수행하여 현재 블록의 재구성된 변환 계수 블록을 획득하도록 구성된다.
- [0821] 일부 실시예에 있어서, 역양자화 유닛(1406)은 구체적으로 양자화된 계수에 대하여 스케일링하여 현재 블록의 재구성된 변환 계수 블록을 획득하도록 구성된다.
- [0822] 일부 실시예에 있어서, 파싱 유닛(1401)은 구체적으로 비트스트림을 파싱하여 LFNST 인덱스 번호의 값을 획득하고, LFNST 인덱스 번호의 값이 0보다 큰 경우, 현재 블록에 대하여 LFNST를 수행하기로 결정하며, LFNST 인덱스 번호의 값이 0인 경우, 현재 블록에 대하여 LFNST를 수행하지 않기로 결정하도록 구성된다.
- [0823] 본 출원의 실시예에서, "유닛"은 회로의 일부, 프로세서의 일부, 프로그램의 일부 또는 소프트웨어 등일 수 있으며, 물론 모듈일 수도 있고, 비모듈일 수도 있음을 이해할 수 있다. 또한, 본 출원의 실시예의 각 컴포넌트는 하나의 처리 유닛에 통합될 수 있고, 또는 각 유닛이 단독으로 물리적으로 존재할 수도 있으며, 또는 2개 이상의 유닛이 하나의 유닛에 통합될 수도 있다. 상기 통합 유닛은 하드웨어 또는 소프트웨어 기능 모듈의 형태로 실현될 수 있다.
- [0824] 통합된 유닛이 소프트웨어 기능 유닛 형태로 구현되어 별도의 제품으로 판매되거나 사용되는 경우, 컴퓨터 판독 가능 저장 매체에 저장될 수 있다. 이러한 이해를 기초로 하여 본 실시예는 디코더(140)에 적용되는 컴퓨터 저장 매체를 제공한다. 컴퓨터 저장 매체에 컴퓨터 프로그램이 저장되어 있다. 컴퓨터 프로그램이 제 2 프로세서에 의해 실행될 때에 상술한 실시예 중의 임의의 방법을 실현한다.
- [0825] 디코더(140)의 구성 및 컴퓨터 저장 매체에 따라, 도 15를 참조하면, 도 15는 본 출원의 실시예에 따른 디코더(140)의 구체적인 하드웨어 구조를 나타내는 도면이다. 도 15에 도시된 바와 같이, 디코더(140)는 제 2 통신 인터페이스(1501), 제 2 메모리(1502) 및 제 2 프로세서(1503)를 포함할 수 있다. 각 컴포넌트는 제 2 버스 시스템(1504)을 통해 함께 결합된다. 제 2 버스 시스템(1504)은 이러한 컴포넌트 간의 연결과 통신을 실현하는 데에 사용된다. 제 2 버스 시스템(1504)은 데이터 버스 이외에 또한 전력 버스, 제어 버스 및 상태 신호 버스를 포함한다. 명확하게 설명하기 위하여, 도 15에 도시된 바와 같이, 다양한 버스는 제 2 버스 시스템(1504)으로 표기되어 있다.
- [0826] 제 2 통신 인터페이스(1401)는 다른 외부 네트워크 요소와 정보를 송수신하는 과정에서 신호를 송수신하는 데에 사용된다.

- [0827] 제 2 메모리(1502)는 제 2 프로세서(1503)에 의해 실행 가능한 컴퓨터 프로그램을 저장하는 데에 사용된다.
- [0828] 제 2 프로세서(1503)는 컴퓨터 프로그램을 실행할 때 다음을 실행하도록 구성된다.
- [0829] 비트스트림을 파싱하여 현재 블록의 예측 파라미터 및 LFNST 파라미터를 획득하며, 예측 파라미터는 예측 모드 파라미터를 포함한다. 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정한다. MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하며, MIP 예측값은 현재 블록의 크로마 컴포넌트 중 일부 샘플의 예측값이다. MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정한다. LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행함을 지시할 경우, 현재 블록의 재구성된 변환 계수 블록을 확정하고, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득한다. 제 2 변환 계수 블록에 대하여 제 1차 변환을 수행함으로써 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 획득한다. 현재 블록의 크로마 컴포넌트의 인트라 예측값 및 재구성된 잔차 블록에 따라 현재 블록의 크로마 컴포넌트의 재구성된 블록을 확정한다.
- [0830] 선택적으로, 다른 실시예로서, 제 2 프로세서(1503)는 또한 컴퓨터 프로그램을 실행할 때에 상술한 실시예 중 임의의 방법을 실행하도록 구성된다.
- [0831] 제 2 메모리(1502)와 제 1 메모리(1302)의 하드웨어 기능은 유사하고, 제 2 프로세서(1503)와 제 1 프로세서(1303)의 하드웨어 기능은 유사하므로, 여기서 상세하게 설명하지 않는다.
- [0832] 본 출원의 실시예는 디코더를 제공한다. 디코더는 파싱 유닛, 제 2 확정 유닛, 제 2 예측 유닛 및 제 2 변환 유닛을 포함할 수 있다. 이와 같이, MIP 모드에 대하여, 디코딩 성능을 보장할 뿐만 아니라 복잡성을 줄일 수 있고, 디코딩 프로세스에 필요한 저장 공간을 줄일 수 있으며, 디코딩 효율을 효과적으로 향상시킬 수 있다. 또한, LFNST 기술이 MIP 모드 예측에 적용될 경우, MIP 파라미터의 도입으로 인해 LFNST 변환이 더욱 유연해질 수 있고, 디코딩 효율이 더욱 향상될 수 있다.
- [0833] 본 출원의 또 다른 실시예에 있어서, 도 16을 참조하면, 도 16은 본 출원의 실시예에서 제공되는 송신 장치의 구조를 나타내는 개략도이다. 도 16에 도시된 바와 같이, 예시적인 송신 장치(1600)가 도시되어 있다. 획득 유닛(1601)은 비디오 신호를 획득하고 비디오 신호를 인코더(1602)로 전송한다. 획득 유닛(1601)은 하나 이상의 카메라(깊이 카메라 포함)를 포함하는 장치일 수 있다. 획득 유닛(1601)은 비디오를 획득하기 위해 비트스트림을 부분적으로 또는 완전히 디코딩하는 장치일 수 있다. 획득 유닛(1601)은 또한 오디오 신호를 캡처하기 위한 하나 이상의 요소를 포함할 수 있다. 인코더(1602)의 특정 구현 방식은 상술한 실시예에서 설명된 인코더(100) 또는 인코더(120)이며, 획득 유닛(1601)으로부터의, 입력 비디오로서의 비디오 신호를 인코딩하여 비디오 비트스트림을 생성한다. 인코더(1602)는 또한 오디오 신호를 인코딩하여 오디오 코드스트림을 생성하도록 하나 이상의 오디오 인코더를 포함할 수 있다. 저장/송신 유닛(1603)은 인코더(1602)로부터 비디오 비트스트림을 수신한다. 저장/송신 유닛(1603)은 또한 인코더(1602)로부터 오디오 비트스트림을 수신하고, 비디오 비트스트림과 오디오 비트스트림을 함께 압축하여 미디어 파일(예를 들어, ISO 기반 미디어 파일 포맷) 또는 전송 스트림을 형성할 수 있다.
- [0834] 선택적으로, 저장/송신 유닛(1603)은 미디어 파일 또는 전송 스트림을 하드 디스크, DVD 디스크, 클라우드, 휴대용 저장 장치와 같은 저장 유닛에 기입한다.
- [0835] 선택적으로, 저장/송신 유닛(1603)은 비트스트림을 인터넷, 유선 네트워크, 셀룰러 네트워크, 무선 근거리 통신망 등과 같은 전송 네트워크로 전송한다.
- [0836] 본 출원의 또 다른 실시예에서, 도 17을 참조하면, 도 17은 본 출원의 실시예에서 제공되는 타겟 디바이스의 구조를 나타내는 개략도이다. 도 17에 도시된 바와 같이, 예시적인 타겟 디바이스(1700)가 도시되어 있다. 수신 유닛(1701)은 네트워크로부터 미디어 파일 또는 전송 스트림을 수신하거나, 또는 저장 장치로부터 미디어 파일 또는 전송 스트림을 판독한다. 수신 유닛(1701)은 미디어 파일 또는 전송 스트림으로부터의 비디오 비트스트림 및 오디오 비트스트림을 분리한다. 수신 유닛(1701)은 또한 비디오 비트스트림을 추출함으로써 새로운 비디오 비트스트림을 생성할 수 있다. 수신 유닛(1701)은 또한 오디오 비트스트림을 추출함으로써 새로운 오디오 비트스트림을 생성할 수 있다. 디코더(1702)는 하나 이상의 비디오 디코더를 포함하며, 예를 들어 그 특정 구현은 상술한 실시예에서 설명된 디코더(200) 또는 디코더(140)이다.

- [0837] 디코더(1702)는 또한 하나 이상의 오디오 디코더를 포함할 수 있다. 디코더(1702)는 수신 유닛(1701)으로부터의 비디오 비트스트림 및 오디오 비트스트림을 디코딩하여 디코딩된 비디오 및 하나 이상의 채널에 대응하는 하나 이상의 디코딩된 오디오를 획득한다. 렌더링 유닛(1703)은 재구성된 비디오에 대해 작업을 수행하여 디스플레이에 적합하도록 한다. 이러한 작업은 감지 품질을 개선하기 위한 다음 작업 중 하나 이상을 포함할 수 있다: 노이즈 감소, 합성, 색 공간 변환, 업 샘플링, 다운 샘플링 등. 렌더링 유닛(1703)은 또한 디스플레이된 오디오 신호의 감지 품질을 개선하기 위해 디코딩된 오디오에 대하여 동작을 수행할 수 있다.
- [0838] 본 출원의 또 다른 실시예에서, 도 18을 참조하면, 도 18은 본 출원의 실시예에서 제공되는 통신 시스템의 구조를 나타내는 개략도이다. 도 18에 도시된 바와 같이, 예시적인 통신 시스템(1800)이 도시되어 있다. 소스 디바이스(1801)는 도 16에 도시된 송신 장치(1600)일 수 있다. 저장/송신 유닛(1603)의 출력은 비트스트림을 저장하거나 전송하기 위해 저장 매체/전송 네트워크(1802)에 의해 처리된다. 타겟 디바이스(1803)는 도 17에 도시된 타겟 디바이스(1700)일 수 있다. 여기서, 수신 유닛(1701)은 저장 매체/전송 네트워크(1802)로부터 비트스트림을 획득할 수 있다. 수신 유닛(1701)은 미디어 파일 또는 전송 스트림으로부터 새로운 비디오 비트스트림을 추출할 수 있다. 수신 유닛(1701)은 또한 미디어 파일 또는 전송 스트림으로부터 새로운 오디오 비트스트림을 추출할 수 있다.
- [0839] 본 출원에 있어서, ‘포함한다’ 또는 그 변형과 같은 용어는 비 배타적인 포함을 커버하기 위한 것으로서 일련의 요소를 포함하는 프로세스, 방법, 개체 또는 장치는 나열된 요소뿐만 아니라 나열되지 않은 다른 요소를 포함하고, 또는 프로세스, 방법, 개체 또는 장치에 고유한 다른 요소를 포함할 수 있다. 더 많은 제한이 없는 상황에서 ‘...을 포함한다’라는 문구에 의해 제한된 요소를 포함하는 프로세스, 방법, 개체 또는 장치에 다른 동일한 요소가 존재하는 것을 배제하지 않는다.
- [0840] 상술한 실시예의 시퀀스 번호는 실시예의 우열을 나타내는 것이 아니라, 단지 설명하는 데에 사용된다.
- [0841] 본 출원의 여러 방법 실시예에 기재된 방법은 충돌이 없는 한 임의로 조합하여 새로운 방법 실시예를 획득할 수 있다.
- [0842] 본 출원의 여러 제품 실시예에 개시된 특징은 충돌이 없는 한 임의로 조합하여 새로운 제품 실시예를 획득할 수 있다.
- [0843] 본 출원의 여러 방법 또는 장치 실시예에 개시된 특징은 충돌이 없는 한 임의로 조합하여 새로운 방법 실시예 또는 장치 실시예를 얻을 수 있다.
- [0844] 상술한 것은 단지 본 발명의 구체적인 실시예이며, 본 발명의 보호 범위는 이것에 한정되는 것은 아니다. 당업자라면 본 발명에 개시된 기술 범위 내에서 변경 또는 교체를 쉽게 도출할 수 있으며, 이러한 변경 또는 교체는 모두 본 발명의 범위 내에 포함되어야 한다. 따라서 본 출원의 보호 범위는 특허 청구 범위에 의해 결정된다.

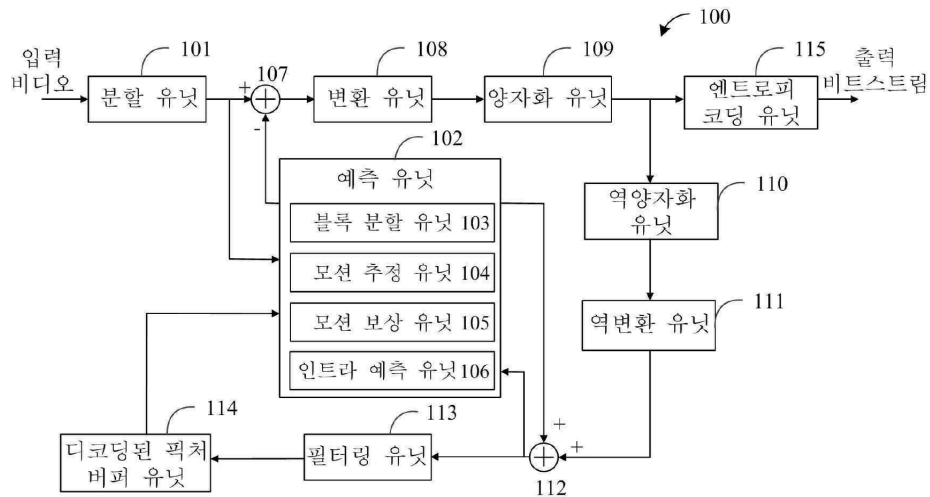
산업상 이용가능성

- [0845] 본 출원의 실시예에 있어서, 인코더 측에서, 현재 블록의 예측 파라미터를 확정한 다음에, 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP(Matrix-Based Intra Prediction) 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하며, MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하고, MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하며, 현재 블록의 크로마 컴포넌트의 인트라 예측값에 따라 현재 블록의 크로마 컴포넌트의 예측 잔차값을 확정하고, 예측 잔차값에 대하여 LFNST(Low-Frequency Non-Separable Transform)를 수행하여 LFNST 파라미터를 확정하고, LFNST 파라미터를 인코딩하고 비트스트림으로 기입한다. 디코더 측에서, 비트스트림을 파싱하여 현재 블록의 예측 파라미터 및 LFNST 파라미터를 획득한 다음에, 예측 모드 파라미터가 현재 블록의 크로마 컴포넌트에 대하여 MIP 모드를 사용하여 인트라 예측값을 확정한다는 것을 지시할 경우, 현재 블록의 인접 샘플값을 획득하고, 현재 블록의 인접 샘플값에 따라 현재 블록의 MIP 입력 샘플값을 확정하고, MIP 입력 샘플값, MIP 가중 행렬 및 시프팅 파라미터에 따라 현재 블록의 크로마 컴포넌트의 MIP 예측값을 확정하고, MIP 예측값을 필터링하여 현재 블록의 크로마 컴포넌트의 인트라 예측값을 확정하고, LFNST 파라미터가 현재 블록에 대하여 LFNST를 수행함을 지시할 경우, 현재 블록의 재구성된 변환 계수 블록을 확정하고, 재구성된 변환 계수 블록 중의 적어도 일부분 재구성된 변환 계수에 대하여 LFNST를 수행함으로써 제 2 변환 계수 블록을 획득하고, 제 2 변환 계수 블록에 대하여 제 1 차 변환을 수행함으로써 현재 블록의 크로마 컴포넌트의 재구성된 잔차 블록을 획득하고, 현재 블록의 크로마 컴포넌트의 인트라 예측값 및 재구성된 잔차 블록에 따라 현재 블록의 크로

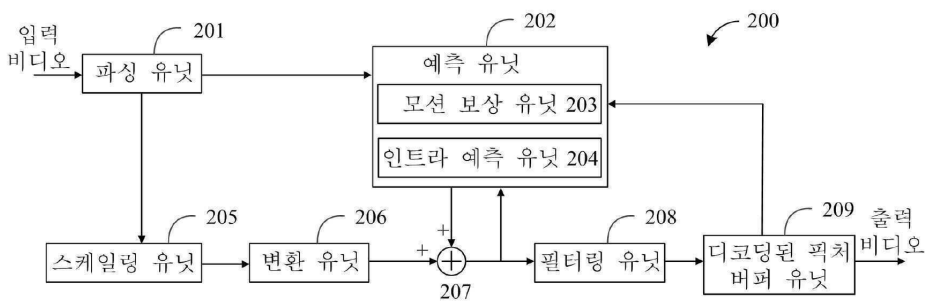
마 컴포넌트의 재구성된 블록을 확정한다. 이와 같이, MIP 모드에 대하여, 인코딩 및 디코딩 성능을 보장할 뿐만 아니라 복잡성을 줄일 수 있고, 인코딩 및 디코딩 프로세스에 필요한 저장 공간을 줄일 수 있으며, 인코딩 및 디코딩 효율을 효과적으로 향상시킬 수 있다. 또한, LFNST 기술이 MIP 모드 예측에 적용될 경우, MIP 파라미터의 도입으로 인해 LFNST 변환이 더욱 유연해질 수 있고, 인코딩 및 디코딩 효율이 더욱 향상될 수 있다.

도면

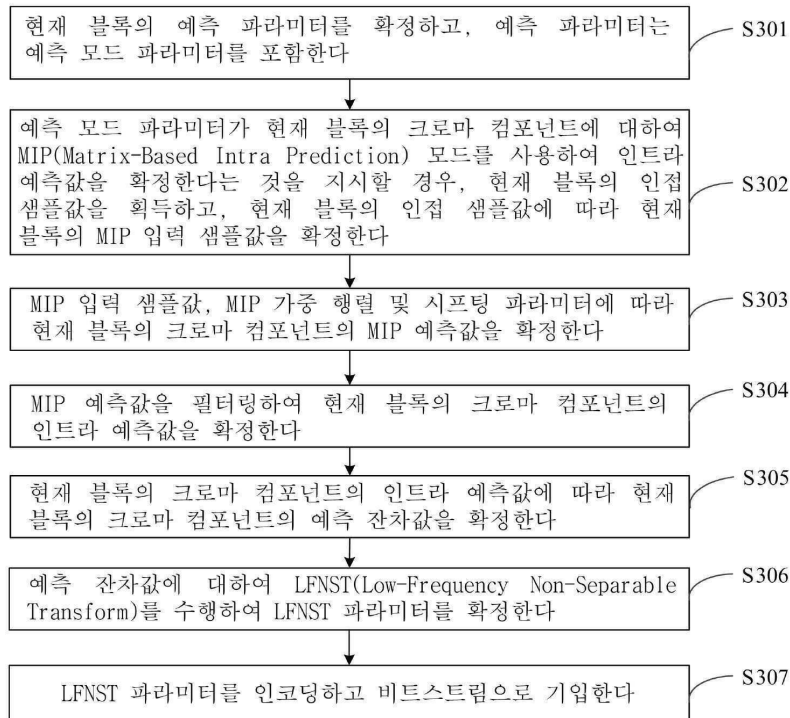
도면1



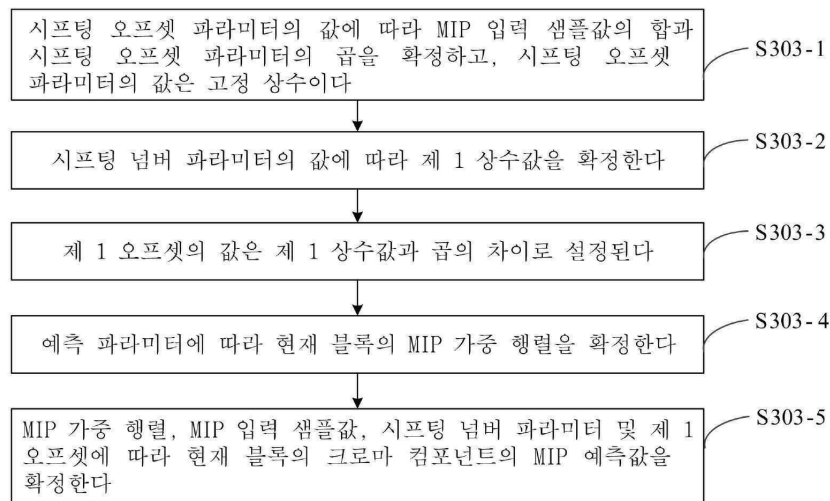
도면2



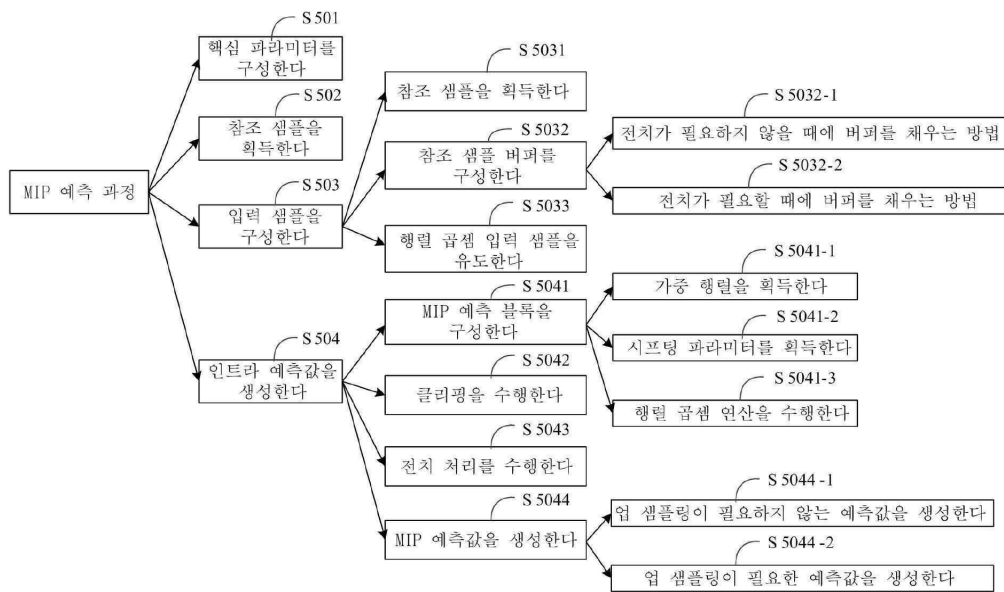
도면3



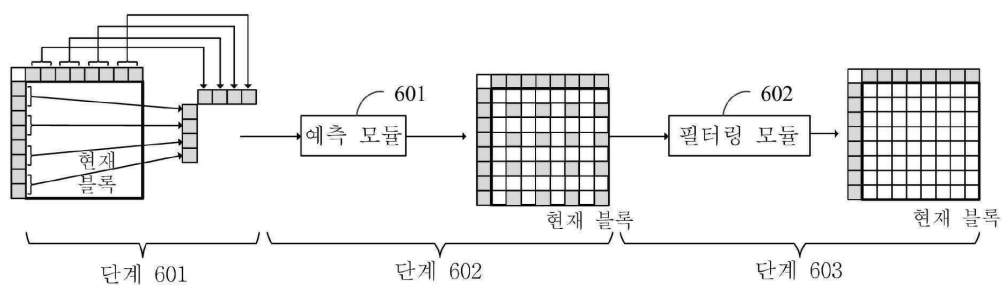
도면4



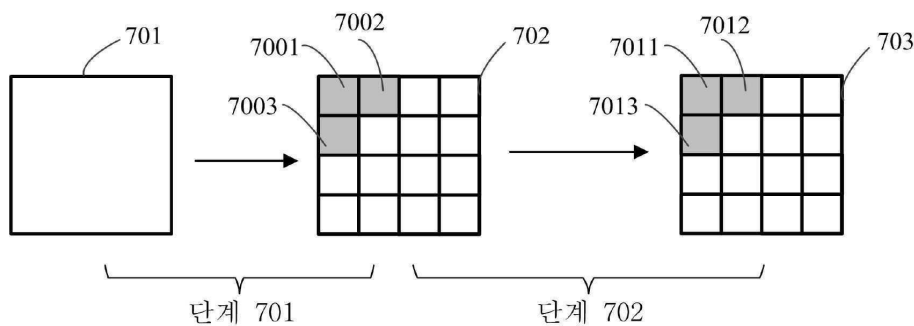
도면5



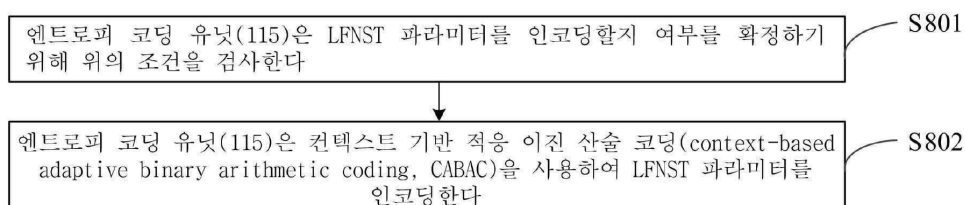
도면6



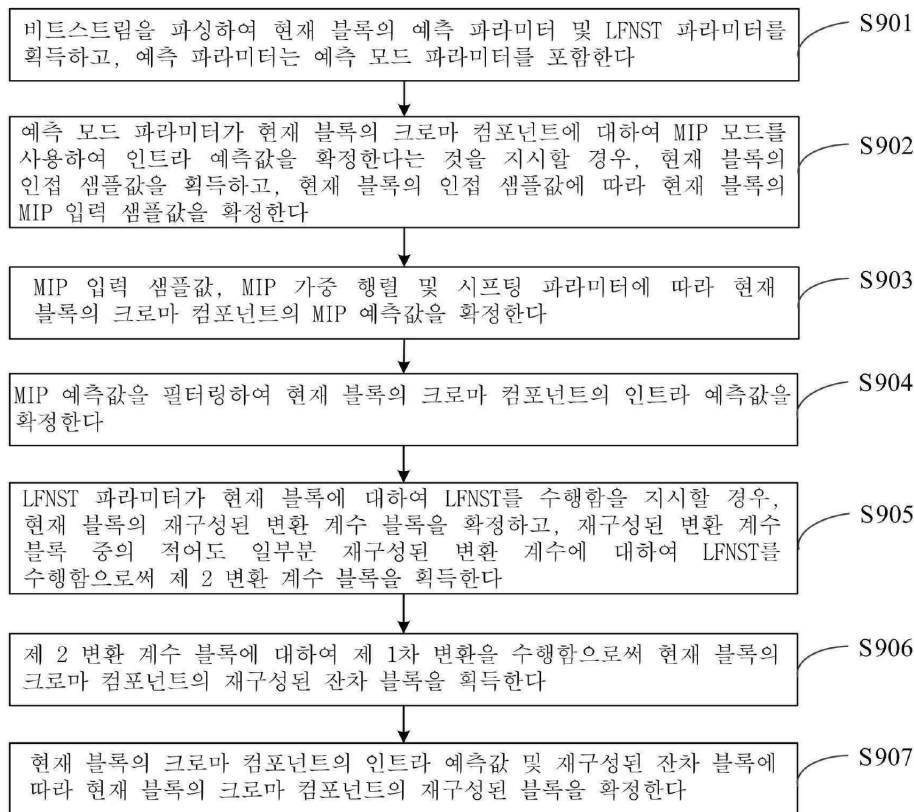
도면7



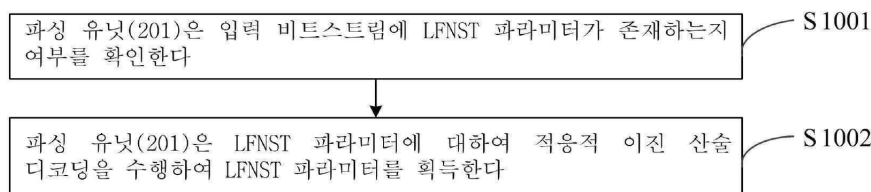
도면8



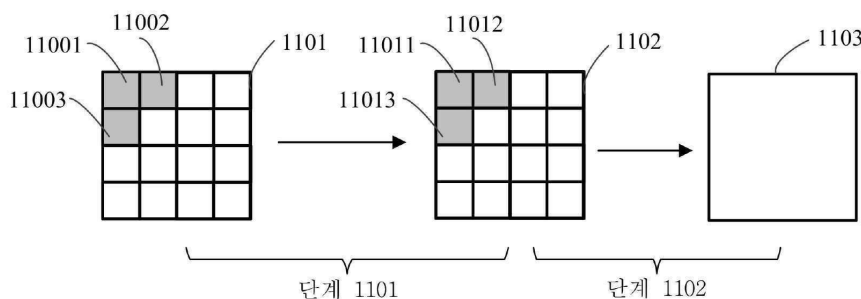
도면9



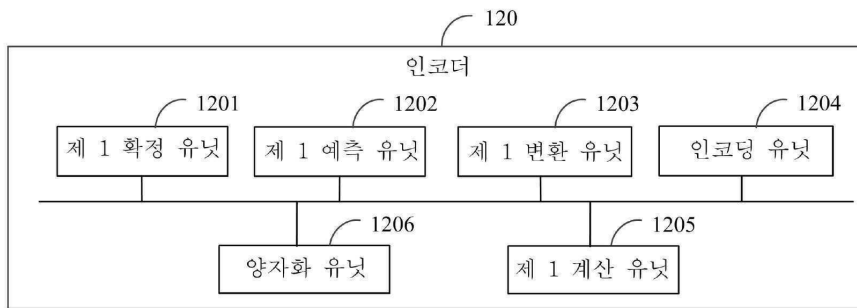
도면10



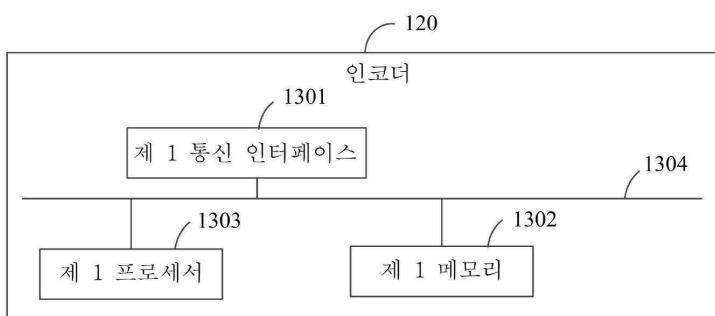
도면11



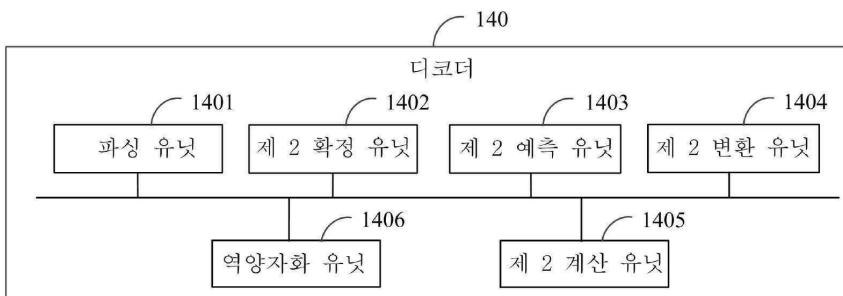
도면12



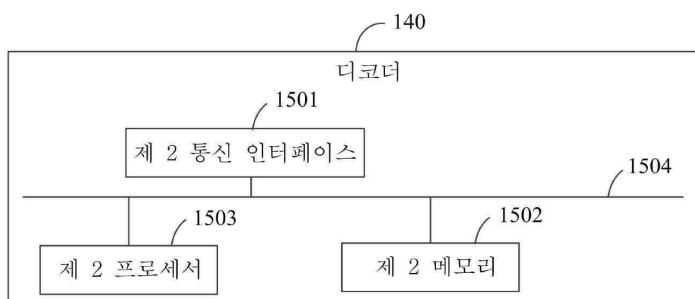
도면13



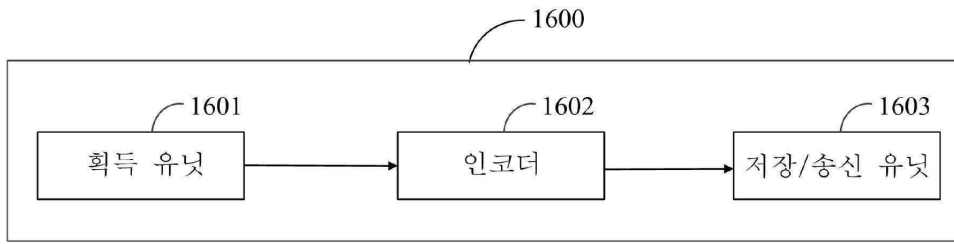
도면14



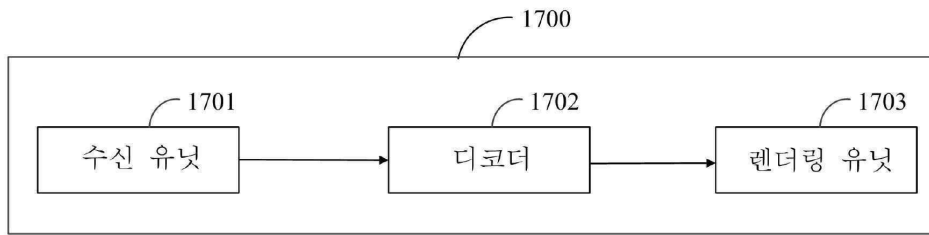
도면15



도면16



도면17



도면18

