



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0045366
(43) 공개일자 2020년05월04일

(51) 국제특허분류(Int. Cl.)

H04N 19/61 (2014.01) H04N 19/119 (2014.01)
H04N 19/176 (2014.01) H04N 19/186 (2014.01)
H04N 19/70 (2014.01)

(52) CPC특허분류

H04N 19/61 (2015.01)
H04N 19/119 (2015.01)

(21) 출원번호 10-2018-0126329

(22) 출원일자 2018년10월22일
심사청구일자 없음

(71) 출원인

주식회사 윌리스표준기술연구소

경기도 성남시 분당구 황새울로 216, 5층(수내동)

(주)휴맥스

경기도 용인시 처인구 영문로 2 (유방동)

(72) 발명자

정재홍

서울시 성동구 돌레11길 1, 101동 1704호

손주형

경기도 의왕시 내손순환로 7, 308동 1402호

(뒷면에 계속)

(74) 대리인

임국일

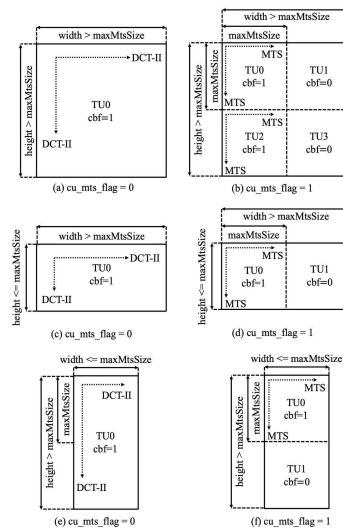
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 다중 변환 선택을 사용하는 비디오 신호 처리 방법 및 장치

(57) 요약

본 발명은 비디오 신호의 처리 방법 및 장치에 관한 것으로, 보다 상세하게는 비디오 신호를 인코딩하거나 디코딩하는 비디오 신호 처리 방법 및 장치에 관한 것이다.

대표도 - 도14



(52) CPC특허분류

H04N 19/176 (2015.01)

H04N 19/186 (2015.01)

H04N 19/625 (2015.01)

H04N 19/70 (2015.01)

(72) 발명자

김동철

경기도 수원시 장안구 하พล로46번길 17, 307동 50
6호

고건중

서울특별시 서초구 잠원로 157, 120동 1006호

박진삼

경기도 안양시 동안구 귀인로 213, 102동 1704호

명세서

청구범위

청구항 1

비디오 신호 처리 장치 및 방법.

발명의 설명

기술 분야

[0001] 본 발명은 비디오 신호의 처리 방법 및 장치에 관한 것으로, 보다 상세하게는 비디오 신호를 인코딩하거나 디코딩하는 비디오 신호 처리 방법 및 장치에 관한 것이다.

배경 기술

[0002] 압축 부호화란 디지털화된 정보를 통신 회선을 통해 전송하거나, 저장 매체에 적합한 형태로 저장하기 위한 일련의 신호 처리 기술을 의미한다. 압축 부호화의 대상에는 음성, 영상, 문자 등의 대상이 존재하며, 특히 영상을 대상으로 압축 부호화를 수행하는 기술을 비디오 영상 압축이라고 일컫는다. 비디오 신호에 대한 압축 부호화는 공간적인 상관관계, 시간적인 상관관계, 확률적인 상관관계 등을 고려하여 잉여 정보를 제거함으로써 이루어진다. 그러나 최근의 다양한 미디어 및 데이터 전송 매체의 발전으로 인해, 더욱 고효율의 비디오 신호 처리 방법 및 장치가 요구되고 있다.

발명의 내용

해결하려는 과제

[0003] 본 발명의 목적은 비디오 신호의 코딩 효율을 높이하고자 함에 있다.

과제의 해결 수단

[0004] 상기와 같은 과제를 해결하기 위해, 본 발명의 일 실시예를 따른 영상 복호화 방법은 변환 유닛을 분할하는 단계 및 변환에 적용되는 커널을 결정하는 단계를 포함한다.

발명의 효과

[0005] 본 발명의 실시예에 따르면, 비디오 신호의 코딩 효율이 높아질 수 있다.

도면의 간단한 설명

[0006] 도 1은 본 발명의 실시예에 따른 비디오 신호 인코더 장치의 개략적인 블록도.
 도 2는 본 발명의 실시예에 따른 비디오 신호 디코더 장치의 개략적인 블록도.
 도 3은 코딩 유닛을 분할하는 본 발명의 일 실시예를 나타내는 도면.
 도 4는 도 3의 분할 구조를 계층적으로 나타내는 방법의 일 실시예를 도시한 도면.
 도 5는 코딩 유닛을 분할하는 본 발명의 추가적인 실시예를 나타낸 도면.
 도 6은 화면내 예측을 위한 참조 픽셀 획득 방법을 나타낸 도면.
 도 7은 화면내 예측에 사용되는 예측 모드들의 일 실시예를 도시한 도면.
 도 8은 AMT (adaptive multiple core transform)에서 사용하는 변환 커널을 정의한 도면.
 도 9는 AMT에서 예측 모드에 따라 적용되는 변환 세트 및 변환 커널 후보를 도시한 도면.
 도 10은 도 8에서 정의한 DCT(discrete cosine transform)-II, DCT-V, DCT-VIII, DST(discrete sine

transform)-I, DST-VII 변환의 0 번째 (해당 변환 커널의 가장 저주파 성분) 기저 함수를 도시한 도면.

도 11은 본 발명의 일 실시예에 따른 다중 변환 선택 (MTS, multiple transform selection) 기술에서 사용하는 변환 커널 및 예측 모드에 따라 정의되는 변환 세트 및 변환 커널 후보를 도시한 도면.

도 12는 본 발명의 일 실시예에 따른 DST-IV, DCT-IV 기저 함수의 정의와 DCT-II, DCT-IV, DCT-VIII, DST-IV, DST-VII의 0 번째 (가장 낮은 주파수 성분) 기저 함수의 그래프를 도시한 도면.

도 13은 본 발명의 일 실시예에 따른 변환 유닛을 시그널링 없이 분할하는 방법을 나타낸 도면.

도 14는 본 발명의 일 실시예에 따른 MTS를 적용하는 방법을 나타낸 도면.

도 15는 휘도 성분과 색차 성분이 다른 부호화 구조로 부호화되는 dual tree 구조의 예를 나타낸 도면.

도 16은 MTS 관련 syntax를 부호화하기 위하여 컨텍스트 인덱스를 결정하는 다양한 방법을 나타낸 도면.

발명을 실시하기 위한 구체적인 내용

- [0007] 본 명세서에서 사용되는 용어는 본 발명에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어를 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도, 관례 또는 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한 특정 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 발명의 설명 부분에서 그 의미를 기재할 것이다. 따라서 본 명세서에서 사용되는 용어는, 단순한 용어의 명칭이 아닌 그 용어가 가진 실질적인 의미와 본 명세서의 전반에 걸친 내용을 토대로 해석되어야 함을 밝혀두고자 한다.
- [0008] 본 발명에서 다음 용어는 다음과 같은 기준으로 해석될 수 있고, 기재되지 않은 용어라도 하기 취지에 따라 해석될 수 있다. 코딩은 경우에 따라 인코딩 또는 디코딩으로 해석될 수 있고, 정보(information)는 값(values), 파라미터(parameter), 계수(coefficients), 성분(elements) 등을 모두 포함하는 용어로서, 경우에 따라 의미는 달리 해석될 수 있으므로 본 발명은 이에 한정되지 아니한다. '유닛'은 영상(픽처) 처리의 기본 단위 또는 픽처의 특정 위치를 지칭하는 의미로 사용되었으며, 경우에 따라서는 '블록', '파티션' 또는 '영역' 등의 용어와 서로 혼용하여 사용될 수 있다. 또한, 본 명세서에서 유닛은 코딩 유닛, 예측 유닛, 변환 유닛을 모두 포함하는 개념으로 사용될 수 있다.
- [0009] 도 1은 본 발명의 일 실시예에 따른 비디오 신호 인코딩 장치의 개략적인 블록도이다. 도 1을 참조하면, 본 발명의 인코딩 장치(100)는 크게 변환부(110), 양자화부(115), 역양자화부(120), 역변환부(125), 필터링부(130), 예측부(150) 및 엔트로피 코딩부(160)를 포함한다.
- [0010] 변환부(110)는 입력 받은 비디오 신호에 대한 화소값을 변환하여 변환 계수 값을 획득한다. 예를 들어, 이산 코사인 변환(Discrete Cosine Transform, DCT) 또는 웨이블릿 변환(Wavelet Transform) 등이 사용될 수 있다. 특히 이산 코사인 변환은 입력된 픽처 신호를 일정 크기의 블록 형태로 나누어 변환을 수행하게 된다. 변환에 있어서 변환 영역 내의 값들의 분포와 특성에 따라서 코딩 효율이 달라질 수 있다.
- [0011] 양자화부(115)는 변환부(110)에서 출력된 변환 계수 값을 양자화한다. 역양자화부(120)에서는 변환 계수 값을 역양자화하고, 역변환부(125)에서는 역양자화된 변환 계수값을 이용하여 원래의 화소값을 복원한다.
- [0012] 필터링부(130)는 복원된 픽처의 품질 개선을 위한 필터링 연산을 수행한다. 예를 들어, 디블록킹 필터 및 적응적 루프 필터 등이 포함될 수 있다. 필터링을 거친 픽처는 출력되거나 참조 픽처로 이용하기 위하여 복호 픽처 버퍼(Decoded Picture Buffer, 156)에 저장된다.
- [0013] 코딩 효율을 높이기 위하여 픽처 신호를 그대로 코딩하는 것이 아니라, 예측부(150)를 통해 이미 코딩된 영역을 이용하여 픽처를 예측하고, 예측된 픽처에 원 픽처와 예측 픽처 간의 레지듀얼 값을 더하여 복원 픽처를 획득하는 방법이 사용된다. 인트라 예측부(152)에서는 현재 픽처 내에서 화면내 예측을 수행하며, 인터 예측부(154)에서는 복호 픽처 버퍼(156)에 저장된 참조 픽처를 이용하여 현재 픽처를 예측한다. 인트라 예측부(152)는 현재 픽처 내의 복원된 영역들로부터 화면내 예측을 수행하여, 화면내 부호화 정보를 엔트로피 코딩부(160)에 전달한다. 인터 예측부(154)는 다시 모션 추정부(154a) 및 모션 보상부(154b)를 포함하여 구성될 수 있다. 모션 추정부(154a)에서는 복원된 특정 영역을 참조하여 현재 영역의 모션 벡터값을 획득한다. 모션 추정부(154a)에서는 참조 영역의 위치 정보(참조 프레임, 모션 벡터 등) 등을 엔트로피 코딩부(160)로 전달하여 비트스트림에 포함될 수 있도록 한다. 모션 추정부(154a)에서 전달된 모션 벡터값을 이용하여 모션 보상부(154b)에서는 화면간 모션 보상을 수행한다.

- [0014] 엔트로피 코딩부(160)는 양자화된 변환 계수, 화면간 부호화 정보, 화면내 부호화 정보 및 인터 예측부(154)로부터 입력된 참조 영역 정보 등을 엔트로피 코딩하여 비디오 신호 비트스트림을 생성한다. 여기서 엔트로피 코딩부(160)에서는 가변 길이 코딩(Variable Length Coding, VLC) 방식과 산술 코딩(arithmetic coding) 등이 사용될 수 있다. 가변 길이 코딩(VLC) 방식은 입력되는 심볼들을 연속적인 코드워드로 변환하는데, 코드워드의 길이는 가변적일 수 있다. 예를 들어, 자주 발생하는 심볼들을 짧은 코드워드로, 자주 발생하지 않은 심볼들은 긴 코드워드로 표현하는 것이다. 가변 길이 코딩 방식으로서 컨텍스트 기반 적응형 가변 길이 코딩(Context-based Adaptive Variable Length Coding, CAVLC) 방식이 사용될 수 있다. 산술 코딩은 연속적인 데이터 심볼들을 하나의 소수로 변환하는데, 산술 코딩은 각 심볼을 표현하기 위하여 필요한 최적의 소수 비트를 얻을 수 있다. 산술 코딩으로서 컨텍스트 기반 적응형 산술 부호화(Context-based Adaptive Binary Arithmetic Code, CABAC)가 이용될 수 있다.
- [0015] 상기 생성된 비트스트림은 NAL(Network Abstraction Layer) 유닛을 기본 단위로 캡슐화 되어 있다. NAL 유닛은 부호화된 슬라이스 세그먼트를 포함하는데, 상기 슬라이스 세그먼트는 정수 개의 코딩 트리 유닛(Coding Tree Unit)으로 이루어진다. 비디오 디코더에서 비트스트림을 디코딩하기 위해서는 먼저 비트스트림을 NAL 유닛 단위로 분리한 후, 분리된 각각의 NAL 유닛을 디코딩해야 한다.
- [0016] 도 2는 본 발명의 일 실시예에 따른 비디오 신호 디코딩 장치(200)의 개략적인 블록도이다. 도 2를 참조하면 본 발명의 디코딩 장치(200)는 크게 엔트로피 디코딩부(210), 역양자화부(220), 역변환부(225), 필터링부(230), 예측부(250)를 포함한다.
- [0017] 엔트로피 디코딩부(210)는 비디오 신호 비트스트림을 엔트로피 디코딩하여, 각 영역에 대한 변환 계수, 모션 정보 등을 추출한다. 역양자화부(220)는 엔트로피 디코딩된 변환 계수를 역양자화하고, 역변환부(225)는 역양자화된 변환 계수를 이용하여 원래의 화소값을 복원한다.
- [0018] 한편, 필터링부(230)는 픽처에 대한 필터링을 수행하여 화질을 향상시킨다. 여기에는 블록 왜곡 현상을 감소시키기 위한 디블록킹 필터 및/또는 픽처 전체의 왜곡 제거를 위한 적응적 루프 필터 등이 포함될 수 있다. 필터링을 거친 픽처는 출력되거나 다음 프레임에 대한 참조 픽처로 이용하기 위하여 복호 픽처 버퍼(Decoded Picture Buffer, 256)에 저장된다.
- [0019] 또한, 본 발명의 예측부(250)는 인트라 예측부(252) 및 인터 예측부(254)를 포함하고, 전술한 엔트로피 디코딩부(210)를 통해 디코딩된 부호화 타입, 각 영역에 대한 변환 계수, 모션 정보 등을 활용하여 예측 픽처를 복원하게 된다.
- [0020] 이와 관련하여, 상기 인트라 예측부(252)에서는 현재 픽처 내의 디코딩된 샘플로부터 화면내 예측을 수행하게 된다. 인터 예측부(254)는 복호 픽처 버퍼(256)에 저장된 참조 픽처 및 모션 정보를 이용하여 예측 픽처를 생성한다. 인터 예측부(254)는 다시 모션 추정부(254a) 및 모션 보상부(254b)를 포함하여 구성될 수 있다. 모션 추정부(254a)에서는 현재 블록과 코딩에 사용하는 참조 픽처의 참조 블록간의 위치 관계를 나타내는 모션 벡터를 획득하여 모션 보상부(254b)로 전달한다.
- [0021] 상기 인트라 예측부(252) 또는 인터 예측부(254)로부터 출력된 예측값, 및 역변환부(225)로부터 출력된 화소값이 더해져서 복원된 비디오 프레임이 생성된다.
- [0022] 이하에서는, 상기 인코딩 장치(100)와 디코딩 장치(200)의 동작에 있어서, 도 3 내지 도 5를 참조하여 코딩 유닛 및 예측 유닛 등을 분할하는 방법을 설명하기로 한다.
- [0023] 코딩 유닛이란 상기에서 설명한 비디오 신호의 처리 과정에서, 예를 들어 화면내(intra)/화면간(inter) 예측, 변환(transform), 양자화(quantization) 및/또는 엔트로피 코딩(entropy coding) 등의 과정에서 픽처를 처리하기 위한 기본 단위를 의미한다. 하나의 픽처를 코딩하는 데 있어서 사용되는 코딩 유닛의 크기는 일정하지 않을 수 있다. 코딩 유닛은 사각형 형태를 가질 수 있고, 하나의 코딩 유닛은 다시 여러 개의 코딩 유닛으로 분할 가능하다.
- [0024] 도 3은 코딩 유닛을 분할하는 본 발명의 일 실시예를 나타낸다. 예를 들어, $2N \times 2N$ 크기를 가지는 하나의 코딩 유닛은 다시 $N \times N$ 크기를 가지는 네 개의 코딩 유닛으로 분할될 수 있다. 이러한 코딩 유닛의 분할은 재귀적으로 이루어질 수 있으며, 모든 코딩 유닛들이 동일한 형태로 분할될 필요는 없다. 다만, 코딩 및 처리과정에서의 편의를 위하여 최대 코딩 유닛의 크기 및/또는 최소 코딩 유닛의 크기에 대한 제한이 있을 수 있다.
- [0025] 하나의 코딩 유닛에 대하여, 해당 코딩 유닛이 분할되는지 여부를 나타내는 정보를 저장할 수 있다. 도 4는 도

3에서 도시하는 코딩 유닛의 분할 구조를 플래그 값을 이용하여 계층적으로 나타내는 방법에 대한 일 실시예를 도시한 것이다. 코딩 유닛의 분할 여부를 나타내는 정보는 해당 유닛이 분할 된 경우 '1', 분할되지 않은 경우 '0'의 값으로 할당할 수 있다. 도 4에서 도시하듯이, 분할 여부를 나타내는 플래그 값이 1이면 해당 노드에 대응하는 코딩 유닛은 다시 4개의 코딩 유닛으로 나누어지고, 0이면 더 이상 나누어지지 않고 해당 코딩 유닛에 대한 처리 프로세스가 수행될 수 있다.

[0026] 상기에서 설명한 코딩 유닛의 구조는 재귀적인 트리 구조를 이용하여 나타낼 수 있다. 즉, 하나의 픽처 또는 최대 크기 코딩 유닛을 루트(root)로 하여, 다른 코딩 유닛으로 분할되는 코딩 유닛은 분할된 코딩 유닛의 개수만큼의 자식(child) 노드를 가지게 된다. 따라서, 더 이상 분할되지 않는 코딩 유닛이 리프(leaf) 노드가 된다. 하나의 코딩 유닛에 대하여 정방형 분할만이 가능하다고 가정할 때, 하나의 코딩 유닛은 최대 4개의 다른 코딩 유닛으로 분할될 수 있으므로 코딩 유닛을 나타내는 트리는 쿼드 트리(Quad tree) 형태가 될 수 있다.

[0027] 인코더에서는 비디오 픽처의 특성(예를 들어, 해상도)에 따라서 혹은 코딩의 효율을 고려하여 최적의 코딩 유닛의 크기가 선택되고 이에 대한 정보 또는 이를 유도할 수 있는 정보가 비트스트림에 포함될 수 있다. 예를 들면, 최대 코딩 유닛의 크기 및 트리의 최대 깊이가 정의될 수 있다. 정방형 분할을 할 경우, 코딩 유닛의 높이 및 너비는 부모 노드의 코딩 유닛의 높이 및 너비의 반이 되므로, 상기와 같은 정보를 이용하면 최소 코딩 유닛 크기를 구할 수 있다. 혹은 역으로, 최소 코딩 유닛 크기 및 트리의 최대 깊이를 미리 정의하여 이용하고, 이를 이용하여 최대 코딩 유닛의 크기를 유도하여 이용할 수 있다. 정방형 분할에서 유닛의 크기는 2의 배수 형태로 변화하기 때문에, 실제 코딩 유닛의 크기는 2를 밑으로 하는 로그값으로 나타내어 전송 효율을 높일 수 있다.

[0028] 디코더에서는 현재 코딩 유닛이 분할되었는지 여부를 나타내는 정보를 획득할 수 있다. 이러한 정보는 특정 조건 하에만 획득하게(전송되게) 하면 효율을 높일 수 있다. 예를 들어 현재 코딩 유닛이 분할 가능한 조건은 현재 위치에서 현재 코딩 유닛 크기를 더한 것이 픽처의 크기보다 작고, 현재 유닛 크기가 기 설정된 최소 코딩 유닛 크기보다 큰 경우이므로, 이러한 경우에만 현재 코딩 유닛이 분할되었는지를 나타내는 정보를 획득할 수 있다.

[0029] 만약 상기 정보가 코딩 유닛이 분할되었음을 나타내는 경우, 분할된 코딩 유닛의 크기는 현재 코딩 유닛의 반이 되고, 현재 처리 위치를 기준으로 하여 4개의 정방형 코딩 유닛들로 분할된다. 각 분할된 코딩 유닛들에 대해서 상기와 같은 처리를 반복할 수 있다.

[0030] 도 5는 코딩 유닛을 분할하는 본 발명의 추가적인 실시예를 나타낸다. 본 발명의 추가적인 실시예에 따르면, 전술한 쿼드 트리 형태의 코딩 유닛은 수평 분할 또는 수직 분할의 바이너리 트리(binary tree) 구조로 더욱 분할될 수 있다. 즉, 루트 코딩 유닛에 대하여 직사각형의 쿼드 트리 분할이 먼저 적용되며, 쿼드 트리의 리프 노드에서 직사각형의 바이너리 트리 분할이 추가적으로 적용될 수 있다. 일 실시예에 따르면, 바이너리 트리 분할은 대칭적인 수평 분할 또는 대칭적인 수직 분할일 수 있으나, 본 발명은 이에 한정되지 않는다.

[0031] 바이너리 트리의 각 분할 노드에서, 분할 형태(즉, 수평 분할 또는 수직 분할)를 지시하는 플래그가 추가적으로 시그널링 될 수 있다. 일 실시예에 따르면, 상기 플래그의 값이 '0'인 경우 수평 분할이 지시되고, 상기 플래그의 값이 '1'인 경우 수직 분할이 지시될 수 있다.

[0032] 다만, 본 발명의 실시예에서 코딩 유닛의 분할 방법은 전술한 방법들로 한정되지 않으며, 비대칭적인 수평/수직 분할, 3개의 직사각형 코딩 유닛으로 분할되는 트리플 트리(triple tree) 등이 적용될 수도 있다.

[0033] 코딩을 위한 픽처 예측(모션 보상)은 더 이상 나누어지지 않는 코딩 유닛(즉 코딩 유닛 트리의 리프 노드)을 대상으로 이루어진다. 이러한 예측을 수행하는 기본 단위를 이하에서는 예측 유닛(prediction unit) 또는 예측 블록(prediction block)이라고 한다.

[0034] 이하, 본 명세서에서 사용되는 유닛이라는 용어는 예측을 수행하는 기본 단위인 상기 예측 유닛을 대체하는 용어로 사용될 수 있다. 다만, 본 발명이 이에 한정되는 것은 아니며, 더욱 광의적으로는 상기 코딩 유닛을 포함하는 개념으로 이해될 수 있다.

[0035] 디코딩이 수행되는 현재 유닛을 복원하기 위해서 현재 유닛이 포함된 현재 픽처 또는 다른 픽처들의 디코딩된 부분이 이용될 수 있다. 복원에 현재 픽처만을 이용하는, 즉 화면내 예측만을 수행하는 픽처(슬라이스)를 인트라 픽처 또는 I 픽처(슬라이스), 화면내 예측과 화면간 예측을 모두 수행할 수 있는 픽처(슬라이스)를 인터 픽처(슬라이스)라고 한다. 인터 픽처(슬라이스) 중 각 유닛을 예측하기 위하여 최대 하나의 모션 벡터 및 레퍼런스 인덱스를 이용하는 픽처(슬라이스)를 예측 픽처(predictive picture) 또는 P 픽처(슬라이스)라고 하며, 최대

두 개의 모션 벡터 및 레퍼런스 인덱스를 이용하는 픽처(슬라이스)를 쌍예측 픽처(Bi-predictive picture) 또는 B 픽처(슬라이스) 라고 한다.

- [0036] 인트라 예측부에서는 현재 픽처 내의 복원된 영역들로부터 대상 유닛의 픽셀값을 예측하는 화면내 예측(Intra prediction)을 수행한다. 예를 들어, 현재 유닛을 중심으로, 좌측 및/또는 상단에 위치한 유닛들의 복원된 픽셀로부터 현재 유닛의 픽셀값을 예측할 수 있다. 이때, 현재 유닛의 좌측에 위치한 유닛들은 현재 유닛에 인접한 좌측 유닛, 좌측 상단 유닛 및 좌측 하단 유닛을 포함할 수 있다. 또한, 현재 유닛의 상단에 위치한 유닛들은 현재 유닛에 인접한 상단 유닛, 좌측 상단 유닛 및 우측 상단 유닛을 포함할 수 있다.
- [0037] 한편, 인터 예측부에서는 현재 픽처가 아닌 복원된 다른 픽처들의 정보를 이용하여 대상 유닛의 픽셀값을 예측하는 화면간 예측(Inter prediction)을 수행한다. 이때, 예측에 이용되는 픽처를 참조 픽처(reference picture)라고 한다. 화면간 예측 과정에서 현재 유닛을 예측하는데 어떤 참조 영역을 이용하는지는, 해당 참조 영역이 포함된 참조 픽처를 나타내는 인덱스 및 모션 벡터(motion vector) 정보 등을 이용하여 나타낼 수 있다.
- [0038] 화면간 예측에는 L0 예측, L1 예측 및 쌍예측(Bi-prediction)이 있을 수 있다. L0 예측은 L0 픽처 리스트에 포함된 1개의 참조 픽처를 이용한 예측이고, L1 예측은 L1 픽처 리스트에 포함된 1개의 참조 픽처를 이용한 예측을 의미한다. 이를 위해서는 1세트의 모션 정보(예를 들어, 모션 벡터 및 참조 픽처 인덱스)가 필요할 수 있다. 쌍예측 방식에서는 최대 2개의 참조 영역을 이용할 수 있는데, 이 2개의 참조 영역은 동일한 참조 픽처에 존재할 수도 있고, 서로 다른 픽처에 각각 존재할 수도 있다. 즉, 쌍예측 방식에서는 최대 2세트의 모션 정보(예를 들어, 모션 벡터 및 참조 픽처 인덱스)가 이용될 수 있는데, 2개의 모션 벡터가 동일한 참조 픽처 인덱스에 대응될 수도 있고 서로 다른 참조 픽처 인덱스에 대응될 수도 있다. 이때, 참조 픽처들은 시간적으로 현재 픽처 이전이나 이후 모두에 표시(또는 출력)될 수 있다.
- [0039] 모션 벡터 및 참조 픽처 인덱스를 이용하여 현재 유닛의 참조 유닛을 획득할 수 있다. 상기 참조 유닛은 상기 참조 픽처 인덱스를 가진 참조 픽처 내에 존재한다. 또한, 상기 모션 벡터에 의해서 특정된 유닛의 픽셀값 또는 보간(interpolation)된 값이 상기 현재 유닛의 예측값(predictor)으로 이용될 수 있다. 서브펠(sub-pel) 단위의 픽셀 정확도를 갖는 모션 예측을 위하여 이를 테면, 휘도 신호에 대하여 8-탭 보간 필터가, 색차 신호에 대하여 4-탭 보간 필터가 사용될 수 있다. 다만, 서브펠 단위의 모션 예측을 위한 보간 필터는 이에 한정되지 않는다. 이와 같이 모션 정보를 이용하여, 이전에 디코딩된 픽처로부터 현재 유닛의 텍스처를 예측하는 모션 보상(motion compensation)이 수행된다.
- [0040] 이하, 도 6 및 도 7을 참고로 본 발명의 실시예에 따른 화면내 예측 방법을 더욱 구체적으로 설명하도록 한다. 전술한 바와 같이, 인트라 예측부에서는 현재 유닛의 좌측 및/또는 상단에 위치한 인접 픽셀들을 참조 픽셀로 이용하여 현재 유닛의 픽셀값을 예측한다.
- [0041] 도 6에 도시된 바와 같이, 현재 유닛의 크기가 $N \times N$ 일 경우, 현재 유닛의 좌측 및/또는 상단에 위치한 최대 $4N+1$ 개의 인접 픽셀들을 사용하여 참조 픽셀들이 설정될 수 있다. 참조 픽셀로 사용될 적어도 일부의 인접 픽셀이 아직 복원되지 않은 경우, 인트라 예측부는 기 설정된 규칙에 따른 참조 샘플 패딩 과정을 수행하여 참조 픽셀을 획득할 수 있다. 또한, 인트라 예측부는 화면내 예측의 오차를 줄이기 위해 참조 샘플 필터링 과정을 수행할 수 있다. 즉, 인접 픽셀들 및/또는 참조 샘플 패딩 과정에 의해 획득된 픽셀들에 필터링을 수행하여 참조 픽셀들이 획득할 수 있다. 인트라 예측부는 이와 같이 획득된 참조 픽셀들을 이용하여 현재 유닛의 픽셀들을 예측한다.
- [0042] 도 7은 화면내 예측에 사용되는 예측 모드들의 일 실시예를 도시한다. 화면내 예측을 위해, 화면내 예측 방향을 지시하는 화면내 예측 모드 정보가 시그널링될 수 있다. 현재 유닛이 화면내 예측 유닛일 경우, 비디오 신호 디코딩 장치는 비트스트림으로부터 현재 유닛의 화면내 예측 모드 정보를 추출한다. 비디오 신호 디코딩 장치의 인트라 예측부는 추출된 화면내 예측 모드 정보에 기초하여 현재 유닛에 대한 화면내 예측을 수행한다.
- [0043] 본 발명의 일 실시예에 따르면, 화면내 예측 모드는 총 67개의 모드를 포함할 수 있다. 각각의 화면내 예측 모드는 기 설정된 인덱스(즉, 인트라 모드 인덱스)를 통해 지시될 수 있다. 예를 들어, 도 7에 도시된 바와 같이 인트라 모드 인덱스 0은 평면(planar) 모드를 지시하고, 인트라 모드 인덱스 1은 DC 모드를 지시하며, 인트라 모드 인덱스 2-66은 서로 다른 방향 모드들(즉, 각도 모드들)을 각각 지시할 수 있다. 화면내 예측부는 현재 유닛의 화면내 예측 모드 정보에 기초하여, 현재 유닛의 화면내 예측에 사용될 참조 픽셀들 및/또는 보간된 참조 픽셀들을 결정한다. 인트라 모드 인덱스가 특정 방향 모드를 지시할 경우, 현재 유닛의 현재 픽셀로부터 상기 특정 방향에 대응하는 참조 픽셀 또는 보간된 참조 픽셀이 현재 픽셀의 예측에 사용된다. 따라서, 화면내 예측

모드에 따라 서로 다른 세트의 참조 픽셀들 및/또는 보간된 참조 픽셀들이 화면내 예측에 사용될 수 있다.

- [0044] 참조 픽셀들 및 화면내 예측 모드 정보를 이용하여 현재 유닛의 화면내 예측이 수행되고 나면, 비디오 신호 디코딩 장치는 역변환부로부터 획득된 현재 유닛의 잔차 신호를 현재 유닛의 화면내 예측값과 더하여 현재 유닛의 픽셀 값들을 복원한다.
- [0046] 원본 신호와 화면간 예측 또는 화면내 예측을 통해 생성한 예측 신호의 차이 신호인 잔차 신호는 픽셀 도메인 전 영역에 에너지가 분산되어 있으므로, 잔차 신호의 픽셀 값 자체를 부호화할 경우, 압축 효율이 떨어지는 문제가 발생한다. 따라서 픽셀 도메인의 잔차 신호를 변환 부호화를 통해 주파수 도메인의 저주파 영역으로 에너지를 집중시키는 과정이 필요하다.
- [0047] HEVC (high efficiency video coding) 표준에서는 신호가 픽셀 도메인에서 고르게 분포하는 경우(이웃하는 픽셀 값이 유사한 경우)에 효율적인 DCT-II (discrete cosine transform type-II)를 대부분 사용하고, 화면내 예측된 4x4 블록에만 DST-VII (discrete sine transform type-VII)을 한정적으로 사용하여 픽셀 도메인의 잔차 신호를 주파수 영역으로 변환하였다. DCT-II 변환의 경우, 화면간 예측을 통해 생성한 잔차 신호 (픽셀 도메인에서 에너지가 고르게 분포하는 경우)에 적합할 수 있으나, 화면내 예측을 통해 생성한 잔차 신호의 경우, 현재 부호화 유닛 주변의 복원된 참조 샘플을 사용하여 예측하는 화면내 예측의 특성 상, 참조 샘플과 멀어질수록 잔차 신호의 에너지가 증가하는 경향을 보일 수 있으므로 DCT-II 변환만을 사용하는 경우, 높은 부호화 효율을 달성할 수 없다.
- [0048] AMT (adaptive multiple core transform)은 예측 방법에 따라 여러 개의 기 설정된 변환 커널 중 적응적으로 변환 커널을 선택하는 변환 기법으로, 어떤 예측 방법을 사용하였는지에 따라 잔차 신호의 픽셀 도메인에서의 패턴 (수평 방향으로 신호의 특성, 수직 방향으로 신호의 특성)이 달라지기 때문에, 단순히 DCT-II만을 사용했을 때 보다 높은 부호화 효율을 기대할 수 있다.
- [0049] 도 8은 AMT에서 사용하는 변환 커널의 정의를 도시한 도면으로, AMT에 적용되는 DCT-II, DCT-V (discrete cosine transform type-V), DCT-VIII (discrete cosine transform type-VIII), DST-I (discrete sine transform type-I), DST-VII 커널의 수식을 나타낸 것이다.
- [0050] DCT와 DST는 각각 cosine, sine의 함수로 표현이 가능하며, 샘플 수 N에 대한 변환 커널의 기저 함수를 $T_i(j)$ 로 표현할 경우, 인덱스 i는 주파수 도메인에서의 인덱스를 나타내며, 인덱스 j는 기저 함수 내의 인덱스를 나타낸다. 즉, i가 작아질수록 저주파 기저 함수를 나타내며, i가 커질수록 고주파 기저 함수를 나타낸다. 기저 함수 $T_i(j)$ 는 2차원 행렬로 표현했을 때, i 번째 행의 j 번째 요소를 나타낼 수 있으며, 도 8(a)에 도시한 변환 커널은 모두 분리 가능한 특성을 가지고 있으므로, 잔차 신호 X에 대하여 가로 방향과 세로 방향에 각각 변환을 수행할 수 있다. 즉, 잔차 신호 블록을 X라 하고, 변환 커널 행렬을 T라 했을 때, 잔차 신호 X에 대한 변환은 TXT^T 으로 나타낼 수 있다. 이때, T'는 변환 커널 행렬 T의 전치행렬 (transpose)를 의미한다.
- [0051] DCT와 DST는 정수가 아닌 소수 형태이므로, 이를 그대로 하드웨어 부호화기, 복호화기에 구현하기에는 부담이 따른다. 따라서 소수 형태의 변환 커널에 scaling과 rounding을 통해 정수 형태의 변환 커널로 근사화시켜야 한다. 변환 커널의 정수 정밀도는 8-bit 또는 10-bit로 결정될 수 있으나, 정밀도가 떨어질 경우, 부호화 효율이 감소할 수 있다. 근사화에 따라 DCT와 DST의 정규 직교 (orthonormal) 성질은 유지되지 않을 수 있으나, 이에 따른 부호화 효율 손실이 크지 않으므로, 변환 커널을 정수 형태로 근사화시키는 것이 하드웨어 부호화기, 복호화기 구현 측면에서 유리하다.
- [0052] 도 9(a), (b)는 AMT에서 화면내 예측 모드에 따른 변환 세트 및 변환 세트에 따라 정의되는 변환 커널 후보를 나타낸 도면이다. 화면내 예측은 부호화 유닛 주변의 복원된 참조 샘플을 활용하여 무방향성 예측 { INTRA_PLANAR (0번 모드), INTRA_DC (1번 모드) }, 방향성 예측 { INTRA_ANGULAR2, INTRA_ANGULAR3, ... }으로 구성될 수 있으며, 추가적으로 직사각형 블록에 적용되는 wide angle 예측 모드 및 색차 성분 신호를 복원된 휘도 성분 신호로부터 예측하는 CCLM (cross-component linear model)을 포함할 수 있다. 이러한 화면내 예측의 방법에 따라 잔차 신호 블록의 수평 방향과 수직 방향의 패턴이 다를 수 있으므로, 예측 모드에 따라 사용할 수 있는 변환 커널 세트를 정의하고, 세트 내에서 사용하는 변환 후보 인덱스를 가로 방향과 세로 방향에 대하여 각각 1-bit으로 시그널링하여 복호화기에서는 부호화기에서 찾은 최적의 변환 커널을 적용하여 역변환을 수행할 수 있다. 도 9(a)는 67개의 화면내 예측 모드를 사용하는 경우, 예측 모드에 따라 정의되는 변환 세트 인덱스를 도시한 것이며, V (vertical)은 수직 방향에 적용하는 변환 세트를 나타내며, H (horizontal)은 수평 방향으로

적용하는 변환 세트를 의미한다. 화면내 예측 모드에 따라 다른 변환 세트를 사용할 수 있으며, 특정 예측 모드에서 수평과 수직 방향에 적용되는 변환 세트는 다를 수 있다. 도 9(b)는 화면내 예측에서 사용하는 변환 세트 및 변환 세트에 따라 사용할 수 있는 변환 커널 후보를 도시한다. Transform Set 0은 {DST-VII, DCT-VIII}로 구성되며, Transform Set 1은 {DST-VII, DST-I}으로 구성되며, Transform Set 2는 {DST-VII, DCT-V}로 구성된다. 화면내 예측의 경우, 현재 블록 주변의 복원된 참조 샘플을 활용하여 예측하는 화면내 예측의 특성 상, 참조 샘플과 멀어질수록, 즉, 잔차 신호 블록의 좌상단 좌표를 기준으로 수평 방향과 수직 방향으로 멀어질수록 잔차 신호의 에너지가 증가하는 경향이 있으므로 이를 잘 표현하는 DST-VII이 효과적이다. 따라서 모든 Transform Set에 DST-VII이 포함되어 있음을 확인할 수 있다.

[0053] 도 9(c)는 화면간 예측에서 사용하는 변환 세트 및 변환 세트에 따라 사용할 수 있는 변환 커널 후보를 도시한다. 화면간 예측의 경우, 사용할 수 있는 변환 세트는 하나이며, Transform Set 0 {DCT-VIII, DST-VII}으로 구성된다.

[0054] AMT는 휘도 성분에만 적용 가능하며, 색차 성분에 대해서는 HEVC와 같이 DCT-II 변환을 사용할 수 있다. 부호화 유닛 단위에서 AMT를 컨트롤할 수 있도록 1-bit flag로 on/off를 지시할 수 있으며, 이 flag가 off를 지시할 경우 색차 성분과 같이 기본 커널인 DCT-II를 사용할 수 있다. 반면 이 flag가 on을 지시하는 경우, 예측 모드에 따라 기 설정된 변환 세트 내에서 사용하는 변환 후보 인덱스를 시그널링하여 복호화기에서 인덱스에 해당하는 변환 커널을 적용하여 역변환을 수행할 수 있다. 수평과 수직 방향에 각기 다른 변환을 적용할 수 있으므로 각각 1-bit씩 총 2-bit로 사용하는 변환 인덱스를 지시할 수 있다. 화면내 예측의 경우, 해당 블록에 AMT를 적용함을 나타내는 flag가 on이라도 non-zero 계수의 개수에 따라 변환 후보 인덱스를 시그널링하지 않을 수도 있다. 예를 들어, non-zero 계수의 개수가 하나 또는 둘인 경우 변환 후보 인덱스를 시그널링하지 않으며 이때에는 수평, 수직 방향 모두 DST-VII을 사용하여 부호화/복호화한다.

[0055] 도 10은 도 8에서 정의한 DCT-II, DCT-V, DCT-VIII, DST-I, DST-VII 변환의 0 번째 (해당 변환 커널의 가장 저주파 성분) 기저 함수를 도시한 도면이다. 도 8(a)에서 정의한 DCT/DST의 변환 기저 함수인 $Ti(j)$ 에 대해 N 이 8이고, i 가 0일 때의 그래프로써, 가로축은 변환 기저 함수 내의 인덱스 j ($j=0, 1, \dots$)을 나타내며, 세로축은 신호의 크기 값을 나타낸다.

[0056] 앞서 기술하였듯이 DST-VII은 인덱스 j 가 증가할수록 신호가 증가하는 경향을 보이므로 화면내 예측과 같이 잔차 신호 블록 내에서 블록의 좌상단 좌표를 기준으로 수평, 수직 방향으로의 거리가 멀어질수록 잔차 신호의 에너지가 증가하는 잔차 신호의 패턴에 효율적일 수 있다.

[0057] 반면 DST-VII과의 duality 특성을 만족하는 DCT-VIII의 경우, 인덱스 j 가 증가할수록 신호의 크기가 감소하는 패턴을 보이므로 잔차 신호 블록 내에서 블록의 좌상단 좌표를 기준으로 수평, 수직 방향으로의 거리가 멀어질수록 잔차 신호의 에너지가 감소하는 잔차 신호의 패턴에 효율적일 수 있다. 즉, 화면간 예측과 같이 블록의 한쪽 경계에서 큰 잔차 값을 가지고 블록의 다른 한쪽 경계로 갈수록 잔차 신호의 절대값이 감소하는 패턴에 효율적일 수 있다.

[0058] DST-I의 경우, 기저 함수 내의 인덱스 j 가 증가할수록 신호가 증가하다가 어느 점을 기점으로 감소하는 형태의 위로 볼록한 신호 형태를 가지므로 잔차 블록내의 중앙으로 이동할수록 잔차 신호의 에너지가 증가하는 잔차 신호의 패턴에 효율적일 수 있다.

[0059] 기존의 영상 압축 표준에서 사용한 DCT-II의 경우, 0 번째 기저 함수는 DC를 나타내며, 화면간 예측과 같이 잔차 블록 내의 화소 값 분포가 균일한 잔차 신호의 패턴에 효율적일 수 있다.

[0060] DCT-V의 경우, DCT-II와 유사하지만 j 가 0일 때의 값이 j 가 0이 아닐 때의 값보다 작은 값을 가지므로 j 가 1일 때 직선이 꺾이는 형태의 신호 모형을 가진다.

[0061] DCT-II만을 주요하게 사용한 기존의 비디오 코덱의 경우, 예측 모드와 원본 신호의 특성에 따라 달라지는 잔차 신호의 패턴에 적응적으로 변환을 수행할 수 없으므로 최적의 부호화 효율을 달성할 수 없으나, 다양한 변환 커널을 예측 모드에 따라 달리 사용하여 잔차 신호의 패턴에 최적화된 변환 커널을 선택하여 변환 부호화를 수행하는 AMT의 경우 높은 압축 효율을 기대할 수 있다.

[0063] AMT와 마찬가지로, 다중 변환 선택 (MTS, multiple transform selection) 기술은 예측 모드에 따라 적응적으로 변환 커널을 선택하여 부호화 효율을 향상시킬 수 있는 변환 부호화 방법이다.

- [0064] 도 11은 본 발명의 일 실시예에 따른 MTS 기술에서 사용하는 변환 커널 및 예측 모드에 따라 정의되는 변환 세트 및 변환 커널 후보를 도시한 도면이다.
- [0065] 도 11(a)는 MTS에 적용되는 DCT-II, DCT-VIII, DST-VII 커널의 수식을 나타낸 것이다. DCT와 DST는 각각 cosine, sine의 함수로 표현이 가능하며, 샘플 수 N에 대한 변환 커널의 기저 함수를 $T_i(j)$ 로 표현할 경우, 인덱스 i는 주파수 도메인에서의 인덱스를 나타내며, 인덱스 j는 기저 함수 내의 인덱스를 나타낸다. 즉, i가 작아질수록 저주파 기저 함수를 나타내며, i가 커질수록 고주파 기저 함수를 나타낸다. 기저 함수 $T_i(j)$ 는 2차원 행렬로 표현했을 때, i 번째 행의 j 번째 요소를 나타낼 수 있으며, 도 11(a)에 도시한 변환 커널은 모두 분리 가능한 특성을 가지고 있으므로, 잔차 신호 X에 대하여 가로 방향과 세로 방향에 각각 변환을 수행할 수 있다. 즉, 잔차 신호 블록을 X라 하고, 변환 커널 행렬을 T라 했을 때, 잔차 신호 X에 대한 변환은 TXT' 으로 나타낼 수 있다. 이때, T'는 변환 커널 행렬 T의 전치행렬 (transpose)를 의미한다.
- [0066] DCT와 DST는 정수가 아닌 소수 형태이므로, 이를 그대로 하드웨어 부호화기, 복호화기에 구현하기에는 부담이 따른다. 따라서 소수 형태의 변환 커널에 scaling과 rounding을 통해 정수 형태의 변환 커널로 근사화시켜야 한다. 변환 커널의 정수 정밀도는 8-bit 또는 10-bit로 결정될 수 있으나, 정밀도가 떨어질 경우, 부호화 효율이 감소할 수 있다. 근사화에 따라 DCT와 DST의 정규 직교 (orthonormal) 성질은 유지되지 않을 수 있으나, 이에 따른 부호화 효율 손실이 크지 않으므로, 변환 커널을 정수 형태로 근사화 시키는 것이 하드웨어 부호화기, 복호화기 구현 측면에서 유리하다.
- [0067] 도 11(b), (c)는 화면내 예측 모드에 따른 변환 세트 및 변환 세트에 따라 정의되는 변환 커널 후보를 나타낸 도면이다. 화면내 예측은 부호화 유닛 주변의 복원된 참조 샘플을 활용하여 무방향성 예측 { INTRA_PLANAR (0번 모드), INTRA_DC (1번 모드) }, 방향성 예측 { INTRA_ANGULAR2, INTRA_ANGULAR3, ... }으로 구성될 수 있으며, 추가적으로 직사각형 블록에 적용되는 wide angle 예측 모드 및 색차 성분 신호를 복원된 휘도 신호로부터 예측하는 CCLM (cross-component linear model)을 포함할 수 있다. 이러한 화면내 예측의 방법에 따라 잔차 신호 블록의 수평 방향과 수직 방향의 패턴이 다를 수 있으므로, 예측 모드에 따라 사용할 수 있는 변환 커널 세트를 정의하고, 세트 내에서 사용하는 변환 후보 인덱스를 가로와 세로 방향에 대하여 각각 1-bit로 시그널링하여 복호화기에서는 부호화기에서 찾은 최적의 변환 커널을 적용하여 역변환을 수행할 수 있다. 도 11(b)는 67개의 화면내 예측 모드를 사용하는 경우, 예측 모드에 따라 정의되는 변환 세트 인덱스를 도시한 것이며, V (vertical)은 수직 방향에 적용하는 변환 세트를 나타내며, H (horizontal)은 수평 방향으로 적용하는 변환 세트를 의미한다. 화면내 예측 모드에 따라 다른 변환 세트를 사용할 수 있으며, 특정 예측 모드에서 수평과 수직 방향에 적용되는 변환 세트는 다를 수 있다. 도 11(c)는 화면내 예측에서 사용하는 변환 세트 및 변환 세트에 따라 사용할 수 있는 변환 커널 후보를 도시한다. Transform Set 0, 1, 2 모두 {DST-VII, DCT-VIII}로 구성된다. 즉, 화면내 예측 모드에 상관없이 하나의 변환 세트를 사용 (모든 화면내 예측 모드에 동일한 변환 커널 후보 사용)하는 것으로 해석할 수 있으나, AMT와 같이 변환 세트 각각이 모두 다른 변환 커널 후보로 구성될 수도 있다. 변환 커널은 정수로 근사화되고, 이는 8-bit 또는 10-bit 정밀도로 표현되는데, 모든 커널은 부호화기, 복호화기에 저장되어야하므로 변환 커널의 종류가 늘어날수록 부호화기, 복호화기의 메모리 부담이 커진다. 따라서 부호화 효율 성능에 가장 큰 영향을 미치는 DCT-II, DCT-VIII, DST-VII 만을 사용하는 것이 효율적이라고 볼 수 있다.
- [0068] 도 11(d)는 화면간 예측에서 사용하는 변환 세트 및 변환 세트에 따라 사용할 수 있는 변환 커널 후보를 도시한다. 화면간 예측의 경우, 사용할 수 있는 변환 세트는 하나이며, Transform Set 0 {DCT-VIII, DST-VII}으로 구성된다. 또는 화면내 예측과 화면간 예측의 통일성을 위해 화면간 예측에서 사용하는 변환 커널 세트를 {DST-VII, DCT-VIII}로 구성할 수도 있다.
- [0069] MTS는 휘도 성분에만 적용 가능하며, 색차 성분에 대해서는 HEVC와 같이 DCT-II 변환을 사용할 수 있다. 부호화 유닛 단위에서 MTS를 컨트롤할 수 있도록 1-bit flag로 on/off를 지시할 수 있으며, 이 flag가 off를 지시할 경우 색차 성분과 같이 기본 커널인 DCT-II를 사용할 수 있다. 반면 이 flag가 on을 지시하는 경우, 예측 모드에 따라 기 설정된 변환 세트 내에서 사용하는 변환 후보 인덱스를 시그널링하여 복호화기에서 인덱스에 해당하는 변환 커널을 적용하여 역변환을 수행할 수 있다. 수평과 수직 방향에 각기 다른 변환을 적용할 수 있으므로 각각 1-bit씩 총 2-bit로 사용하는 변환 인덱스를 지시할 수 있다. 화면내 예측의 경우, 해당 블록에 MTS를 적용함을 나타내는 flag가 on이라도 non-zero 계수의 개수에 따라 변환 후보 인덱스를 시그널링하지 않을 수도 있다. Non-zero 계수의 개수가 하나 또는 둘인 경우 변환 후보 인덱스를 시그널링하지 않으며 이때에는 수평, 수직 방향 모두 DST-VII를 사용하여 부호화/복호화한다.

- [0070] 변환 부호화의 경우, 분리 가능한 변환 커널에 대하여 2차원 행렬로 표현하여 가로 방향과 세로 방향에 각각 변환을 수행하는 것이므로 2차원 행렬곱 연산을 두 차례 수행하는 것으로 볼 수 있다. 이는 많은 연산량을 수반하므로 구현 관점에서 문제가 될 수 있다. 따라서 구현 관점에서 DCT-II와 같이 butterfly structure를 사용하여 연산량을 줄일 수 있는지 또는 해당 변환 커널을 구현 복잡도가 낮은 변환 커널들로 분해가 가능한지가 중요할 이슈가 될 수 있다. 이러한 관점에서 보았을 때, DST-VII와 DCT-VIII의 구현 복잡도는 높은 편이므로 DST-VII, DCT-VIII과 유사한 특성을 보이면서 구현 복잡도가 낮은 변환은 DST-VII와 DCT-VIII을 대체할 수 있다.
- [0071] DST-IV (discrete sine transform type-IV)와 DCT-IV (discrete cosine transform type-IV)는 각각 DST-VII, DCT-VIII을 대체할 수 있는 후보로 볼 수 있다. 샘플 수 $2N$ 에 대한 DCT-II partial butterfly structure는 샘플 수 N 에 대한 DCT-IV 커널을 포함하고 있고, 샘플 수 N 에 대한 DST-IV 커널은 샘플 수 N 에 대한 DCT-IV 커널로부터 간단한 연산인 부호 반전과 해당 기저 함수를 역순으로 정렬함으로써 구현할 수 있으므로, 샘플수 $2N$ 에 대한 DCT-II로부터 간단하게 샘플수 N 에 대한 DST-IV와 DCT-IV를 구현할 수 있다.
- [0072] 도 12는 본 발명의 일 실시예에 따른 DST-IV, DCT-IV 기저 함수의 정의와 DCT-II, DCT-IV, DCT-VIII, DST-IV, DST-VII의 0 번째 (가장 낮은 주파수 성분) 기저 함수의 그래프를 도시한 도면이다. 도 11(a), 도 12(a)에서 정의한 DCT/DST의 변환 기저 함수인 $T_i(j)$ 에 대해 N 이 8이고, i 가 0일 때의 그래프로써, 가로축은 변환 기저 함수 내의 인덱스 j ($j=0, 1, \dots$)을 나타내며, 세로축은 신호의 크기 값을 나타낸다.
- [0073] DST-IV와 DST-VII은 유사한 신호 모형으로, 인덱스 j 가 증가할수록 신호가 증가하는 경향을 보이므로 화면내 예측과 같이 잔차 신호 블록 내에서 블록의 좌상단 좌표를 기준으로 수평, 수직 방향으로의 거리가 멀어질수록 잔차 신호의 에너지가 증가하는 잔차 신호의 패턴에 효율적일 수 있다.
- [0074] DCT-IV와 DCT-VIII은 유사한 신호 모형으로, 인덱스 j 가 증가할수록 신호의 크기가 감소하는 패턴을 보이므로 잔차 신호 블록 내에서 블록의 좌상단 좌표를 기준으로 수평, 수직 방향으로의 거리가 멀어질수록 잔차 신호의 에너지가 감소하는 잔차 신호의 패턴에 효율적일 수 있다. 즉, 화면간 예측과 같이 블록의 한쪽 경계에서 큰 잔차 값을 가지고 블록의 다른 한쪽 경계로 갈수록 잔차 신호의 절대값이 감소하는 패턴에 효율적일 수 있다.
- [0075] 잔차 신호의 패턴은 예측 방법뿐만 아니라, 블록의 크기에도 영향을 받으므로 이를 고려하여 변환 커널 세트를 구성할 수 있다. 작은 길이에 대해서는 DST-VII, DCT-VIII보다 DST-IV, DCT-IV가 부호화 이득 관점에서 더 효율적이므로, 가로 또는 세로의 길이가 임계치보다 작거나 같은 경우, 해당 방향에 DST-VII 대신 DST-IV, DCT-VIII 대신 DCT-IV를 사용하고, 가로 또는 세로의 길이가 임계치보다 큰 경우, 해당 방향에 DST-VII, DCT-VIII을 사용할 수 있다. 예를 들어, 임계치가 8인 경우, 가로 길이가 8이고, 세로 길이가 16인 8×16 블록에 대하여 MTS가 적용되는 경우, 가로 방향에 대해서는 {DST-IV, DCT-IV} 중의 하나를 사용하고, 세로 방향에 대해서는 {DST-VII, DCT-VIII} 중의 하나를 사용할 수 있다. 이때, 각 방향에 사용되는 커널은 각각 1-bit로 지시될 수 있다.
- [0077] 도 3 내지 도 5에서 기술한 것처럼 코딩 유닛 (또는 부호화 유닛)은 조건에 따라 다양한 모양으로 분할될 수 있으며, 블록의 분할 정보가 지시되어 부호화기에서 찾은 최적의 부호화 구조를 복호화기에서 복원할 수 있다. 반면, 변환을 수행하는 기본 단위인 변환 유닛 (TU, Transform Unit)은 분할되지 않을 수 있다. 변환 유닛을 분할하는 경우, 부호화 유닛을 루트 노드로 하여 RQT (Residual Quad-Tree) 형태, 즉, 부호화 유닛과 동일한 사이드로부터 쿼드트리 형태로 변환 유닛을 분할할 수 있으나 부호화기의 복잡도 증가 대비 얻을 수 있는 부호화 이득이 크지 않을 수 있다. 변환 유닛을 분할하지 않는 경우, 대부분의 경우 변환 유닛은 분할되지 않으나, 변환 유닛의 가로 또는 세로의 길이가 변환 블록의 최대 크기보다 크고 transform tree syntax structure가 존재함을 나타내는 1-bit flag가 1인 경우, 변환 유닛은 시그널링 없이 분할될 수 있다.
- [0078] 도 13은 변환 유닛을 시그널링 없이 분할하는 본 발명의 일 실시예를 나타낸 것으로, $MaxTbSizeY$ 는 휘도 성분 변환 블록의 최대 크기를 나타내는 변수로 변환 커널의 최대 사이즈에 따라 결정될 수 있으며, $tbWidth$ 는 변환 블록의 가로 길이, $tbHeight$ 는 변환 블록의 높이를 나타내는 변수이다. 코딩 트리가 2의 지수승 크기의 부호화 유닛으로 분할되고, $MaxTbSizeY$ 또한 2의 지수승인 경우에 대하여, 변환 블록의 가로 또는 세로가 $MaxTbSizeY$ 보다 큰 경우 가로 또는 세로가 $MaxTbSizeY$ 인 다수의 변환 유닛으로 분할될 수 있다.
- [0079] 도 13 (a)는 $tbWidth$ 와 $tbHeight$ 가 모두 $MaxTbSizeY$ 이하인 경우로, 이 경우 변환 유닛은 분할되지 않는다. 즉, 변환 유닛의 사이즈는 부호화 유닛의 사이즈와 동일하다고 볼 수 있다. 가로 방향에 대해서는 N-point ($N=tbWidth$) 변환 커널이 적용될 수 있으며, 세로 방향에 대해서는 M-point ($M=tbHeight$) 변환 커널이 적용될 수 있다.

- [0080] 도 13 (b)는 $tbWidth=2*MaxTbSizeY$, $tbHeight=2*MaxTbSizeY$ 인 경우로, 이 경우 가로와 세로의 길이가 모두 $MaxTbSizeY$ 인 $MaxTbSizeY \times MaxTbSizeY$ 크기의 4개의 변환 유닛들로 분할될 수 있다. 분할된 변환 유닛은 z-order index 기준으로 처리될 수 있으며 각각의 변환 유닛을 TU0, TU1, TU2, TU3로 표현할 수 있다. 각각의 변환 유닛에 대하여 가로 방향에 대해서는 N-point ($N=MaxTbSizeY$) 변환 커널이 적용될 수 있으며, 세로 방향에 대해서는 M-point ($M=MaxTbSizeY$) 변환 커널이 적용될 수 있다.
- [0081] 도 13 (c)는 $tbWidth=2*MaxTbSizeY$, $tbHeight \leq MaxTbSizeY$ 인 경우로, 이 경우 가로 길이가 $MaxTbSizeY$ 이고 세로 길이가 $tbHeight$ 인 $MaxTbSizeY \times tbHeight$ 크기의 2개의 변환 유닛들로 분할될 수 있다. 분할된 변환 유닛은 z-order index 기준으로 처리될 수 있으며 각각의 변환 유닛을 TU0, TU1로 표현할 수 있다. 각각의 변환 유닛에 대하여 가로 방향에 대해서는 N-point ($N=MaxTbSizeY$) 변환 커널이 적용될 수 있으며, 세로 방향에 대해서는 M-point ($M=tbHeight$) 변환 커널이 적용될 수 있다.
- [0082] 도 13 (d)는 $tbWidth \leq MaxTbSizeY$, $tbHeight=2*MaxTbSizeY$ 인 경우로, 이 경우 가로 길이가 $tbWidth$ 이고 세로 길이가 $MaxTbSizeY$ 인 $tbWidth \times MaxTbSizeY$ 크기의 2개의 변환 유닛들로 분할될 수 있다. 분할된 변환 유닛은 z-order index 기준으로 처리될 수 있으며 각각의 변환 유닛을 TU0, TU1로 표현할 수 있다. 각각의 변환 유닛에 대하여 가로 방향에 대해서는 N-point ($N=tbWidth$) 변환 커널이 적용될 수 있으며, 세로 방향에 대해서는 M-point ($M=MaxTbSizeY$) 변환 커널이 적용될 수 있다.
- [0083] 휘도 성분 Y와 색차 성분 Cb, Cr이 동일한 부호화 구조를 가지는 single tree structure의 경우, 상기 기술된 블록의 사이즈 조건에 따라 휘도 성분 변환 블록이 분할되는 경우, 동시에 색차 성분 변환 블록도 분할될 수 있다.
- [0084] 반면, 휘도 성분과 색차 성분이 서로 다른 부호화 구조를 가지는 dual tree structure의 경우, 상기 기술된 블록의 사이즈 조건에 따라 휘도 성분 변환 블록이 분할되더라도 색차 성분은 다른 트리 구조를 가지므로 색차 성분 변환 블록은 분할되지 않을 수 있다. 색차 성분 변환 블록의 분할 여부는 휘도 성분 변환 블록의 분할 여부에 상관 없이 독립적으로 상기 기술된 블록의 사이즈 조건에 따라 결정될 수 있다. 즉, 색차 성분 변환 블록의 가로와 세로 길이가 변환 블록의 최대 사이즈보다 크기에 따라 결정될 수 있다.
- [0086] 변환 유닛 분할의 다른 실시예로, single tree structure에서 휘도 성분 변환 블록이 분할되더라도 색차 성분 변환 블록은 분할되지 않을 수 있다. 변환 블록의 최대 사이즈는 변환 부호화에서 사용하는 변환 커널의 최대 사이즈에 따라 결정될 수 있다. 예를 들어, 4, 8, 32, 64-point 변환 커널을 사용하는 경우, 변환 블록의 최대 사이즈는 64로 결정될 수 있다. 일반적으로, 변환 커널의 사이즈가 커질수록 부호화 효율이 증가하는 경향을 보이므로, 부호화 효율 측면에서는 가능한 큰 크기의 변환 커널을 사용하는 것이 유리하나, 복잡도를 고려하여 변환 커널의 최대 사이즈를 결정해야 한다. 이때, 휘도 성분의 변환 블록이 변환 블록의 최대 사이즈보다 크더라도, 컬러 포맷에 따라 색차 성분 변환 블록은 변환 블록의 최대 사이즈보다 크지 않을 수 있다. 따라서, 색차 성분 변환 블록은 분할하지 않고, 큰 사이즈의 변환 커널을 적용하여 부호화 효율을 높일 수 있다. 예를 들어, 4:2:0 컬러 포맷 영상의 128×64 부호화 유닛은 128×64 Y 부호화 블록과 64×32 Cb, Cr 부호화 블록으로 구성될 수 있는데, 변환 유닛은 64×64 크기의 두 개의 Y 변환 블록과 64×32 Cb, Cr 변환 블록으로 구성될 수 있다. 이 경우, 색차 성분 변환 블록은 가로와 세로 모두 64보다 크지 않으므로 분할되지 않으며, 두 개의 64×64 휘도 성분 변환 블록이 처리된 후, 64×32 색차 성분 변환 블록이 처리될 수 있다. 화면내 예측의 경우, 가까운 참조 샘플을 사용하는 것이 잔차 신호를 줄일 수 있으므로, 휘도 성분 변환 블록이 분할되는 경우 색차 성분 변환 블록도 분할되는 것이 오히려 유리할 수 있다. 따라서 화면간 예측일 경우에만 single tree에서 휘도 성분 변환 블록과 색차 성분 변환 블록의 분할을 다르게 할 수도 있다.
- [0088] MTS는 부호화 유닛의 너비와 높이가 모두 $maxMtsSize$ 이하인 휘도 성분 블록에 적용할 수 있으며, $maxMtsSize$ 는 MTS에서 사용하는 변환 커널의 최대 사이즈를 나타낸다. 예를 들어, MTS에서 4, 8, 16, 32-point DST-VII, DCT-VIII을 사용하는 경우, $maxMtsSize=32$ 가 될 수 있다. $maxMtsSize$ 는 SPS (Sequence Parameter Set) 등의 high level syntax에서 지시될 수 있으며, 시그널링 없이 복호화기에서 고정된 값을 사용할 수도 있다. 본 발명은 $maxMtsSize$ 의 값에 상관 없이 적용될 수 있다. Transform tree syntax가 존재하는 지를 나타내는 1-bit flag (cu_cbf)가 1이고, 부호화 유닛의 너비와 높이가 모두 $maxMtsSize$ 이하인 경우, 1-bit flag로 해당 블록에 MTS가 적용되는 지를 부호화 유닛 레벨에서 지시할 수 있으며, 1-bit flag가 1일 경우, 해당 부호화 유닛의

회도 성분 블록에 DCT-II가 아닌 다른 커널이 적용될 수 있다. 이하에서는 상기 부호화 유닛 레벨 1-bit flag를 `cu_mts_flag`로 명칭한다. `cu_mts_flag`가 1일 때, 가로, 세로 방향에 적용되는 변환 커널을 각각 1-bit flag로 변환 유닛 레벨에서 지시할 수 있다. 이하에서는 가로, 세로 방향에 적용되는 변환 커널을 지시하는 변환 유닛 레벨 1-bit flag를 각각 `tu_mts_hor_flag`, `tu_mts_ver_flag`로 명칭한다. `tu_mts_hor_flag` 또는 `tu_mts_ver_flag`가 0일 경우, 변환 커널 세트 내의 첫 번째 변환 커널을 지시할 수 있으며, 1일 경우, 변환 커널 세트 내의 두 번째 변환 커널을 지시할 수 있다. 예를 들어 변환 커널 세트를 {DST-VII, DCT-VIII}로 구성한 경우, `tu_mts_hor_flag`가 0일 때는 가로 방향에 DST-VII이 적용됨을 나타내고 1일 때는 가로 방향에 DCT-VIII이 적용됨을 나타낸다. `tu_mts_ver_flag`도 동일한 방식으로 해석할 수 있다.

[0089] 부호화 유닛의 너비 또는 높이가 `maxMtsSize`보다 큰 경우 MTS에서 사용하는 변환 커널을 사용할 수 없으므로 해당 블록에는 MTS를 적용할 수 없고 관련된 신택스 (`cu_mts_flag`, `tu_mts_hor_flag`, `tu_mts_ver_flag` 등)은 시그널링되지 않고, 복호화기에서 기설정된 규칙에 따라 추론될 수 있다. 예를 들어, `cu_mts_flag`는 시그널링 없이 0으로 추론되고, 해당 블록의 가로, 세로 방향에 모두 DCT-II가 적용될 수 있다.

[0091] 부호화 유닛의 너비 또는 높이가 `maxMtsSize`보다 큰 경우에도 MTS를 적용하기 위하여 TU를 분할할 수 있다.

[0092] 도 14는 본 발명의 일 실시예에 따른 MTS 적용 방법을 나타낸 도면이다. `width`와 `height`는 각각 블록의 너비와 높이를 나타내며, `cbf`는 해당 변환 블록에 0이 아닌 변환 계수가 존재하는 지를 나타낸다. `cbf`가 1일 경우, 해당 변환 블록에 0이 아닌 변환 계수가 하나 이상 존재함을 의미하고, 0일 경우 해당 변환 블록에 0이 아닌 변환 계수가 존재하지 않음을 나타낸다. `width`와 `height`가 모두 `maxMtsSize` 이하인 블록에 대해서는 기존과 같이 MTS 관련 syntax를 해석하여 MTS를 적용할 수 있고, 그 이외의 경우 (`width` 또는 `height`가 `maxMtsSize`보다 큰 경우)에는 `cu_mts_flag`의 해석을 달리하여 MTS를 적용할 수 있다. 즉, `width` 또는 `height`가 `maxMtsSize`보다 큰 경우에도 `cu_mts_flag`를 시그널링하여, `cu_mts_flag`가 1일 때는 TU를 분할하여 MTS를 적용하고, `cu_mts_flag`가 0일 때는 가로, 세로 방향에 모두 DCT-II를 적용할 수 있다. `cu_mts_flag`가 0일 때의 TU 분할은 도 13에서 기술한 방법을 따를 수 있다.

[0094] 도 14 (a)는 `width`와 `height`가 모두 `maxMtsSize`보다 크고 `cu_mts_flag`가 0인 경우를 나타낸다. 회도 성분 `cbf`가 1인 경우 해당 회도 성분 변환 블록에 0이 아닌 변환 계수가 하나 이상 존재함을 의미하므로 회도 성분 블록의 가로와 세로 방향에 DCT-II를 적용하여 잔차 신호를 복원할 수 있다. 회도 성분 `cbf`가 0인 경우, 모든 변환 계수가 0임을 의미하므로 역변환 과정 없이 잔차 신호를 복원할 수 있다. 색차 성분도 마찬가지로, 색차 성분의 `cbf`에 따라 가로와 세로 방향에 모두 DCT-II를 적용하거나 역변환 과정 없이 잔차 신호를 복원할 수 있다. 도 14 (a)에서는 TU가 분할되지 않는 경우를 도시하였으나, 도 13에서 기술한 것처럼 `width` 또는 `height`가 `MaxTbSizeY`보다 큰 경우에는 TU가 분할될 수 있다.

[0095] 도 14 (b)는 `width`와 `height`가 모두 `maxMtsSize`보다 크고 (`width=2*maxMtsSize`, `height=2*maxMtsSize`) `cu_mts_flag`가 1인 경우를 나타낸다. 가로와 세로가 모두 `maxMtsSize`인 `maxMtsSize` x `maxMtsSize` 크기의 다수의 변환 유닛으로 분할될 수 있으며, `cbf`가 1인 회도 성분 변환 블록에 대하여 `tu_mts_hor_flag`, `tu_mts_ver_flag`가 지시하는 변환 커널을 각각 가로, 세로 방향에 적용할 수 있다. `cbf`가 0인 회도 성분 변환 블록의 경우 역변환 과정 없이 잔차 신호를 복원할 수 있다. 색차 성분 변환 블록의 경우, 색차 성분의 `cbf`에 따라 가로와 세로 방향에 모두 DCT-II를 적용하거나 역변환 과정 없이 잔차 신호를 복원할 수 있다.

[0097] 도 14 (c)는 `width`가 `maxMtsSize`보다 크고 `height`가 `maxMtsSize` 이하일 때, `cu_mts_flag`가 0인 경우를 나타낸다. 회도 성분 변환 블록의 `cbf`가 1인 경우, 해당 회도 성분 변환 블록에 0이 아닌 변환 계수가 하나 이상 존재함을 의미하므로, 가로와 세로 방향에 모두 DCT-II를 적용하여 잔차 신호를 복원할 수 있고, `cbf`가 0인 경우 역변환 과정 없이 잔차 신호를 복원할 수 있다. 색차 성분 변환 블록의 경우, `cbf`에 따라 가로와 세로 방향에 모두 DCT-II를 적용하거나 역변환 과정 없이 잔차 신호를 복원할 수 있다. 도 14 (c)에서는 TU가 분할되지 않는 경우를 도시하였으나, 도 13에서 기술한 것처럼 `width`가 `MaxTbSizeY`보다 큰 경우에는 TU가 분할될 수 있다.

[0098] 도 14 (d)는 `width`가 `maxMtsSize`보다 크고 (`width=2*maxMtsSize`) `height`가 `maxMtsSize` 이하일 때, `cu_mts_flag`가 1인 경우를 나타낸다. 가로가 `maxMtsSize`이고 세로가 `height`인 `maxMtsSize` x `height` 크기의 다수의 변환 유닛으로 분할될 수 있으며, `cbf`가 1인 회도 성분 변환 블록에 대하여 `tu_mts_hor_flag`, `tu_mts_ver_flag`가 지시

하는 변환 커널을 각각 가로, 세로 방향에 적용할 수 있다. cbf가 0인 휘도 성분 변환 블록의 경우 역변환 과정 없이 잔차 신호를 복원할 수 있다. 색차 성분 변환 블록의 경우, 색차 성분의 cbf에 따라 가로와 세로 방향에 모두 DCT-II를 적용하거나 역변환 과정 없이 잔차 신호를 복원할 수 있다.

[0100] 도 14 (e)는 width가 maxMtsSize 이하이고 height가 maxMtsSize보다 클 때, cu_mts_flag가 0인 경우를 나타낸다. 휘도 성분 변환 블록의 cbf가 1인 경우, 해당 휘도 성분 변환 블록에 0이 아닌 변환 계수가 하나 이상 존재함을 의미하므로, 가로와 세로 방향에 모두 DCT-II를 적용하여 잔차 신호를 복원할 수 있고, cbf가 0인 경우 역변환 과정 없이 잔차 신호를 복원할 수 있다. 색차 성분 변환 블록의 경우, cbf에 따라 가로와 세로 방향에 모두 DCT-II를 적용하거나 역변환 과정 없이 잔차 신호를 복원할 수 있다. 도 14 (e)에서는 TU가 분할되지 않는 경우를 도시하였으나, 도 13에서 기술한 것처럼 height가 MaxTbSizeY보다 큰 경우에는 TU가 분할될 수 있다.

[0101] 도 14 (f)는 width가 maxMtsSize 이하이고 height가 maxMtsSize보다 클 때 (height=2*maxMtsSize), cu_mts_flag가 1인 경우를 나타낸다. 가로가 width이고 세로가 maxMtsSize인 width x maxMtsSize 크기의 다수의 변환 유닛으로 분할될 수 있으며, cbf가 1인 휘도 성분 변환 블록에 대하여 tu_mts_hor_flag, tu_mts_ver_flag가 지시하는 변환 커널을 각각 가로, 세로 방향에 적용할 수 있다. cbf가 0인 휘도 성분 변환 블록의 경우 역변환 과정 없이 잔차 신호를 복원할 수 있다. 색차 성분 변환 블록의 경우, 색차 성분의 cbf에 따라 가로와 세로 방향에 모두 DCT-II를 적용하거나 역변환 과정 없이 잔차 신호를 복원할 수 있다.

[0103] 도 14 (b), (d), (f)에서 MTS를 적용하는 다른 실시예로, 복잡도를 고려하여 cbf가 0이 아닌 휘도 성분 변환 블록은 변환 커널을 지시하는 syntax 요소를 공유할 수 있다. 즉, cbf가 0이 아닌 휘도 성분 변환 블록에 대하여 z-order index 처리 순서 상, 첫 번째 휘도 성분 변환 블록만 tu_mts_hor_flag, tu_mts_ver_flag를 파싱하고, 나머지 휘도 성분 변환 블록은 cbf가 0이 아닌 첫 번째 휘도 성분 변환 블록에서 사용하는 변환 커널과 동일한 변환 커널을 적용할 수 있다.

[0105] 도 14 (b), (d), (f)에서 MTS를 적용하는 다른 실시예로, 복잡도를 고려하여 cbf가 0이 아닌 휘도 성분 변환 블록은 변환 커널을 지시하는 syntax 요소를 파싱하지 않고, 변환 커널 세트 내의 첫 번째 변환 커널을 가로와 세로 방향에 모두 적용할 수 있다. 상기 기술한 방법은 화면내 예측에만 적용될 수도 있다.

[0107] 도 14 (b), (d), (f)에서 single tree일 때 휘도 성분 블록의 사이즈 조건에 따라 다수의 변환 유닛으로 분할되는 경우, 색차 성분 변환 블록도 동시에 분할될 수 있으나 cbf가 0이 아닌 색차 성분 변환 블록에는 DCT-II가 적용되므로 휘도 성분 변환 블록이 분할되더라도 색차 성분 변환 블록은 분할되지 않을 수도 있다. 이 경우, 색차 성분 블록에 대해서는 분할되었을 때보다 더 큰 변환 커널을 사용할 수 있으므로 색차 성분에 대한 부호화 효율을 향상시킬 수 있다.

[0109] 도 14의 실시예는 cu_mts_flag를 파싱하여 1일 경우, 블록의 너비 또는 높이가 maxMtsSize보다 큰 경우 변환 유닛을 너비가 min(width, maxMtsSize)이고, 높이가 min(height, maxMtsSize)인 다수의 변환 유닛으로 분할하여 MTS를 적용할 수 있고 (이때, min(x, y)함수는 x, y 중에서 작은 값을 반환하는 함수로, x<y를 만족하는 경우 x를 반환하고, 그렇지 않은 경우 y를 반환한다.), cu_mts_flag가 0인 경우에 도 13에서 기술한 TU분할 방법을 따를 수 있다. 즉, cu_mts_flag가 0이고 블록의 너비 또는 높이가 MaxTbSizeY보다 큰 경우에 변환 유닛은 분할될 수 있다.

[0111] MTS를 적용하는 다른 실시예로, 부호화 유닛의 사이즈에 따라 cu_mts_flag를 시그널링할 수 있다. 도 14의 실시예와는 달리, 블록의 너비와 높이가 모두 MaxTbSizeY 이하인 경우에만 cu_mts_flag를 시그널링하고, 그 이외의 경우에는 MTS관련 syntax를 시그널링하지 않을 수 있다. cu_mts_flag가 1이고 블록의 너비 또는 높이가 maxMtsSize보다 큰 경우 도 14에서 기술한 방법에 따라 변환 유닛을 분할하고 MTS를 적용할 수 있다. 블록의 너비 또는 높이가 MaxTbSizeY보다 큰 경우, 도 13에서 기술한 바와 같이, 변환 유닛을 너비가 min(width,

MaxTbSizeY)이고 높이가 $\min(\text{height}, \text{MaxTbSizeY})$ 인 다수의 변환 유닛으로 분할하고, 각각의 변환 유닛의 가로와 세로 방향에는 변환 커널을 지시하는 요소를 파싱하지 않고, DCT-II가 적용될 수 있다.

[0113] 다른 실시예로, 블록의 너비와 높이가 모두 MaxTbSizeY 이하인 경우를 포함하여, 블록의 너비 또는 높이 둘 중의 하나만 MaxTbSizeY 이하이고 나머지 하나는 MaxTbSizeY보다 큰 경우에도 cu_mts_flag를 시그널링할 수도 있다.

[0114] 블록의 너비와 높이가 모두 MaxTbSizeY 이하이고, 도 14에서 기술한 바와 같이 MTS를 적용할 수 있다. 블록의 너비가 MaxTbSizeY 이하이고 높이가 MaxTbSizeY보다 클 때, cu_mts_flag가 1인 경우, 너비가 $\min(\text{width}, \text{maxMtsSize})$ 이고 높이가 MaxTbSizeY인 다수의 변환 유닛으로 분할할 수 있고, 각각의 휘도 성분 변환 블록에 대하여 cbf가 0이 아닐 경우 가로 방향은 변환 커널을 지시하는 tu_mts_hor_flag를 파싱하여 역변환을 수행하고, 세로 방향은 변환 커널을 지시하는 syntax 요소 파싱 없이 DCT-II를 적용할 수 있다. cu_mts_flag가 0일 때에는 너비가 width이고 높이가 MaxTbSizeY인 width x MaxTbSizeY 사이즈의 다수의 변환 유닛으로 분할하고 cbf가 0이 아닐 경우 가로와 세로 방향 모두에 변환 커널을 지시하는 syntax 요소 파싱 없이 DCT-II를 적용할 수 있다. 블록의 너비가 MaxTbSizeY보다 크고 높이가 MaxTbSizeY 이하일 때, cu_mts_flag가 1인 경우, 너비가 MaxTbSizeY이고 높이가 $\min(\text{height}, \text{maxMtsSize})$ 인 다수의 변환 유닛으로 분할할 수 있고, 각각의 휘도 성분 변환 블록에 대하여 cbf가 0이 아닐 경우 가로 방향은 변환 커널을 지시하는 syntax 요소 파싱 없이 DCT-II를 적용하고 세로 방향은 변환 커널을 지시하는 tu_mts_ver_flag를 파싱하여 역변환을 수행할 수 있다. cu_mts_flag가 0일 때에는 너비가 MaxTbSizeY이고 높이가 height인 다수의 변환 유닛으로 분할하고 cbf가 0이 아닐 경우 가로와 세로 방향 모두에 변환 커널을 지시하는 syntax 요소 파싱 없이 DCT-II를 적용할 수 있다. 블록의 너비와 높이가 모두 MaxTbSizeY보다 큰 경우에는 도 13에서 기술한 변환 유닛 분할 방법을 따를 수 있다.

[0116] 색차 성분 변환 블록의 경우, HEVC 표준과 같이 가로와 세로 방향에 모두 DCT-II를 적용할 수 있으나, DCT-II 이외의 변환 커널을 적용할 수 있도록 MTS를 확장할 수 있다.

[0118] 색차 성분에 MTS를 적용하는 첫 번째 실시예로, 색차 성분에 대응되는 해당 휘도 변환 블록에서 사용하는 변환 커널을 색차 성분 변환 블록에 적용할 수 있다. 휘도 성분과 색차 성분이 동일한 부호화 구조로 부호화되는 single tree 구조일 경우, 변환 유닛 레벨에서 시그널링되는 휘도 성분 변환 블록에 적용되는 변환 커널을 지시하는 syntax 요소를 색차 성분 변환 블록에도 적용할 수 있다.

[0119] 도 15는 휘도 성분과 색차 성분이 다른 부호화 구조로 부호화되는 dual tree 구조의 예를 도시한 것이다. Dual tree일 경우, 색차 성분 블록에 상응하는 휘도 성분 블록이 하나가 아닌 여러개일 수 있으므로 TL (top-left), TR (top-right), CR (center), BL (bottom-left), BR (bottom-right) 중, CR 위치의 휘도 성분 블록에 적용된 변환 커널을 해당 색차 성분 변환 블록에 적용할 수 있다. Dual tree 구조의 경우, CTU (Coding Tree Unit) 내에 존재하는 모든 휘도 성분 부호화 블록을 디코딩한 후, 색차 성분 부호화 블록이 처리되므로 색차 성분 블록에 대하여 CR 위치의 휘도 성분 블록에서 사용한 변환 커널을 참조하기 위해서는 휘도 성분 블록에서 어떤 변환 커널을 사용했는지를 저장하고 있어야 한다.

[0121] 색차 성분에 MTS를 적용하는 두 번째 실시예로, 색차 성분에 대응되는 해당 휘도 부호화 블록의 cu_mts_flag가 1일 경우, 색차 성분 변환 블록의 가로와 세로 방향에 모두 변환 커널 세트의 첫 번째 변환 커널을 적용할 수 있다. Single tree 구조일 경우, 부호화 유닛 레벨에서 시그널링되는 cu_mts_flag가 1일 경우 해당 색차 성분 블록에 변환 커널 세트의 첫 번째 변환 커널을 적용하고, cu_mts_flag가 0인 경우 가로와 세로 방향에 모두 DCT-II를 적용할 수 있다.

[0122] Dual tree일 경우, CR 위치의 cu_mts_flag가 1일 때에는 해당 색차 성분 변환 블록의 가로와 세로에 모두 변환 커널 세트의 첫 번째 변환 커널을 적용할 수 있고, CR 위치의 cu_mts_flag가 0일 때에는 가로와 세로에 모두 DCT-II를 적용할 수 있다. 색차 성분 변환 블록에서 휘도 성분 블록의 cu_mts_flag를 참조하기 위해서는 휘도 성분 블록의 처리가 끝난 이후에도 cu_mts_flag를 저장하고 있어야 한다. 또는 CR 위치의 cu_mts_flag를 참조하지 않고, dual tree의 색차 성분 부호화 블록에도 cu_mts_flag를 명시적으로 시그널링할 수 있다. 색차 성분 블

록의 `cu_mts_flag`가 0인 경우 가로와 세로 방향에 모두 DCT-II가 적용되고, `cu_mts_flag`가 1일 경우 가로와 세로 방향에 모두 변환 커널 세트의 첫 번째 변환 커널이 적용될 수 있다.

- [0124] 두 가지 실시예에서 색차 성분 변환 블록에 MTS가 적용되기 위해서는 색차 성분 변환 블록의 너비가 4 이상이고 `maxMtsSize` 이하이거나 높이가 4 이상이고 `maxMtsSize` 이하일 수 있다. 상기 조건에서 한 방향의 길이는 4 이상 `maxMtsSize` 이하이고, 나머지 다른 한 방향의 길이가 4 미만이거나 `maxMtsSize`보다 큰 경우, 한쪽 방향에만 MTS를 적용하고 나머지 방향에는 DCT-II를 적용할 수 있다.
- [0125] 휘도 성분에서 사용한 예측 모드와 색차 성분에서 사용한 예측 모드가 서로 다를 경우, 잔차 신호의 패턴이 서로 다를 수 있으므로 화면 내 예측 모드 중 DM (Direct Mode)일 때만 해당 색차 성분 변환 블록에 MTS를 적용할 수도 있다. MDMS (Multiple Direct Mode Signalling)이 사용되는 경우, CR 위치의 휘도 성분 블록에서 사용한 화면내 예측 모드를 해당 색차 성분 블록에 사용하지 않고, TL, TR, CR, BL, BR 위치 중의 하나를 시그널링하여 그 위치의 휘도 성분 블록에서 사용한 화면내 예측 모드를 해당 색차 성분 블록에 적용할 수 있다. 두 번째 실시예에서 MDMS를 사용하고 DM 모드에서만 색차 성분 변환 블록에 MTS를 적용하는 경우, CR 위치에서 사용한 변환 커널을 참조하지 않고, DM 위치를 지시하는 곳에서 사용한 변환 커널을 색차 성분 변환 블록에 적용할 수 있다.
- [0127] 부호화 유닛에 MTS가 적용되는 지를 지시하는 부호화 유닛 레벨 1-bit flag인 `cu_mts_flag`는 컨텍스트를 고려하는 CABAC (Context Adaptive Binary Arithmetic Coding)을 통해 부호화될 수 있으며, CABAC으로 부호화되는 syntax 요소에서 사용되는 컨텍스트는 컨텍스트 인덱스인 `ctxIdx`로 지시될 수 있다.
- [0129] 도 16은 `cu_mts_flag`의 컨텍스트 인덱스를 결정하는 다양한 방법을 나타낸 도면이다.
- [0130] Method I에서 `ctxIdx`는 quad-tree dpeth를 나타내는 `qtDepth`에 따라 결정될 수 있다. 블록의 사이즈에 따라 MTS가 적용되는 빈도가 변하는 경향이 있으므로 블록의 사이즈를 나타내는 인자 중의 하나인 quad-tree depth를 이용하여 컨텍스트 인덱스를 결정할 수 있다.
- [0132] Method II에서 `ctxIdx`는 quad-tree (QT) depth를 나타내는 `qtDepth`와 multi type tree (MTT)의 depth를 나타내는 `mttDepth`의 합으로 결정할 수 있다. MTT는 가로 방향과 세로 방향으로 binary split과 ternary split으로 분할하는 것을 허용하는 부호화 구조로, QT와 MTT를 모두 허용하는 블록 구조로 부호화되는 경우에 컨텍스트 인덱스를 결정할 때, 블록의 사이즈를 고려할 경우 `qdDepth`와 더불어 `mttDepth`를 추가적으로 고려할 수 있다.
- [0134] Method III에서 `ctxIdx`는 해당 부호화 유닛이 화면내 예측인지, 아니면 화면간 예측인지에 따라 결정될 수 있다. `CuPredMode`는 해당 부호화 유닛이 화면내 예측인지 화면간 예측인지를 나타내는 변수로, `CuPredMode`가 `MODE_INTRA`인 경우 해당 부호화 유닛이 화면내 예측으로 부호화된 것을 나타낼 수 있다. $x ? y : z$ 형태의 3항 연산자는 x 가 참이거나 0이 아닐 경우 y 를 반환하고, 그렇지 않을 경우 z 를 반환한다. 화면내 예측, 화면간 예측인지에 따라 MTS가 적용되는 빈도가 변할 수 있으므로, 이를 고려하여 컨텍스트 인덱스를 결정할 수 있다.
- [0136] Method IV는 Method I과 Method III을 결합한 형태로, 블록의 사이즈를 나타내는 지표 중의 하나인 QT depth와 예측 방법에 따라서 컨텍스트 인덱스를 결정할 수 있다. 이때, `NUM_MTS_CU_FLAG_CTX_INTRA`는 화면내 예측 모드에서 `cu_mts_flag`를 부호화할 때 사용하는 컨텍스트의 수를 나타내는 변수이다.
- [0138] Method V는 Method II와 Method III을 결합한 형태로, 블록의 사이즈를 나타내는 QT depth와 MTT depth의 합과 예측 방법에 따라서 컨텍스트 인덱스를 결정할 수 있다.

- [0140] Method VI에서 $ctxIdx$ 는 QP (Quantization Parameter)를 고려하여 결정할 수 있다. $\text{floor}(x)$ 는 x 의 소수점을 버린 값을 반환하는 함수이며, $\text{clip3}(a, b, x)$ 는 x 를 a 와 b 사이의 값으로 클리핑하는 함수이다. $ctxIdx_{max}$ 는 $ctxIdx$ 의 최대값을 나타내며, N 은 QP 구간을 양자화하는 파라미터로 양의 정수이다. QP가 커질수록 양자화 스텝 사이즈가 커지고, 이에 따라 어떤 변환을 수행하든 변환된 잔차 신호가 많이 소실될 수 있다. 따라서 QP가 커질수록 MTS가 적용되는 빈도가 줄어드는 경향을 보일 수 있으므로, QP를 고려하여 컨텍스트 인덱스를 결정할 수 있다.
- [0142] Method VII에서 $ctxIdx$ 는 부호화 유닛 모양, 즉, 블록의 긴변 대 짧은 변의 비율에 따라 결정할 수 있다. $cbWidth$ 와 $cbHeight$ 는 각각 부호화 유닛의 너비와 높이를 나타내며 $\log_2(x)$ 는 밑이 2인 로그함수를 나타낸다. 블록의 모양에 따라 MTS가 적용되는 빈도가 변할 수 있으므로 블록의 긴변 대 짧은 변의 비율에 따라 컨텍스트 인덱스를 결정할 수 있다.
- [0144] Method VIII에서 $ctxIdx$ 는 부호화 유닛의 모양, 즉, 블록이 정사각형 블록인지 직사각형 블록인지에 따라 결정할 수 있다. 블록이 정사각형 블록인 경우 $ctxIdx$ 는 0으로 결정될 수 있으며, 직사각형 블록인 경우 $ctxIdx$ 는 1로 결정될 수 있다. 블록의 모양에 따라 MTS가 적용되는 빈도가 변할 수 있으므로 블록이 정사각형인지 직사각형인지에 따라 컨텍스트 인덱스를 결정할 수 있다.
- [0146] Method IX에서 $ctxIdx$ 는 현재 블록 주변의 블록에서 MTS를 사용했는 지에 따라 결정할 수 있다. 픽처의 좌상단 휘도 성분을 기준으로 블록의 휘도 성분 좌상단 좌표가 (x_0, y_0) 인 부호화 유닛에 대하여 주변 위치는 L (left)와 A (above)를 사용할 수 있으며, L의 좌표는 (x_0-1, y_0) , A 좌표는 (x_0, y_0-1) 로 정의될 수 있다. $condL$ 은 L 위치의 cu_mts_flag 값을 나타내며, $availableL$ 은 L 위치의 블록이 참조 가능한지를 나타낸다. $condA$ 는 A 위치의 cu_mts_flag 값을 나타내며, $availableA$ 는 A 위치의 블록이 참조 가능한지를 나타낸다. 즉, L 위치에서 블록이 존재하고 참조 가능한 경우에 cu_mts_flag 가 1인 경우 $ctxIdx$ 는 1 증가하며, A 위치에서 블록이 존재하고 참조 가능한 경우에 cu_mts_flag 가 1인 경우 $ctxIdx$ 는 추가로 1 만큼 더 증가할 수 있다. 현재 블록의 주변에서 MTS가 많이 사용되는 경우, 현재 블록에도 MTS가 사용될 확률이 높으므로, 이를 반영하기 위하여 주변의 MTS 적용 정보를 활용하여 컨텍스트 인덱스를 결정할 수 있다. 도 16에서는 두 개의 주변 블록을 참조하는 실시예를 도시하였으나, 본 발명은 참조하는 주변 블록의 개수에 제한되지 않는다. 즉, M ($M>0$)개의 주변 블록을 참조하는 경우, M개의 위치에서 블록이 존재하고 참조가능함과 cu_mts_flag 의 값에 따라 $ctxIdx$ 를 결정할 수 있으며 $ctxIdx$ 는 M만큼 증가할 수 있다.
- [0148] 이상에서는 본 발명을 구체적인 실시예를 통하여 설명하였으나, 당업자라면 본 발명의 취지 및 범위를 벗어나지 않고 수정, 변경을 할 수 있다. 따라서 본 발명의 상세한 설명 및 실시예로부터 본 발명이 속하는 기술분야에 속한 사람이 용이하게 유추할 수 있는 것은 본 발명의 권리범위에 속하는 것으로 해석된다.

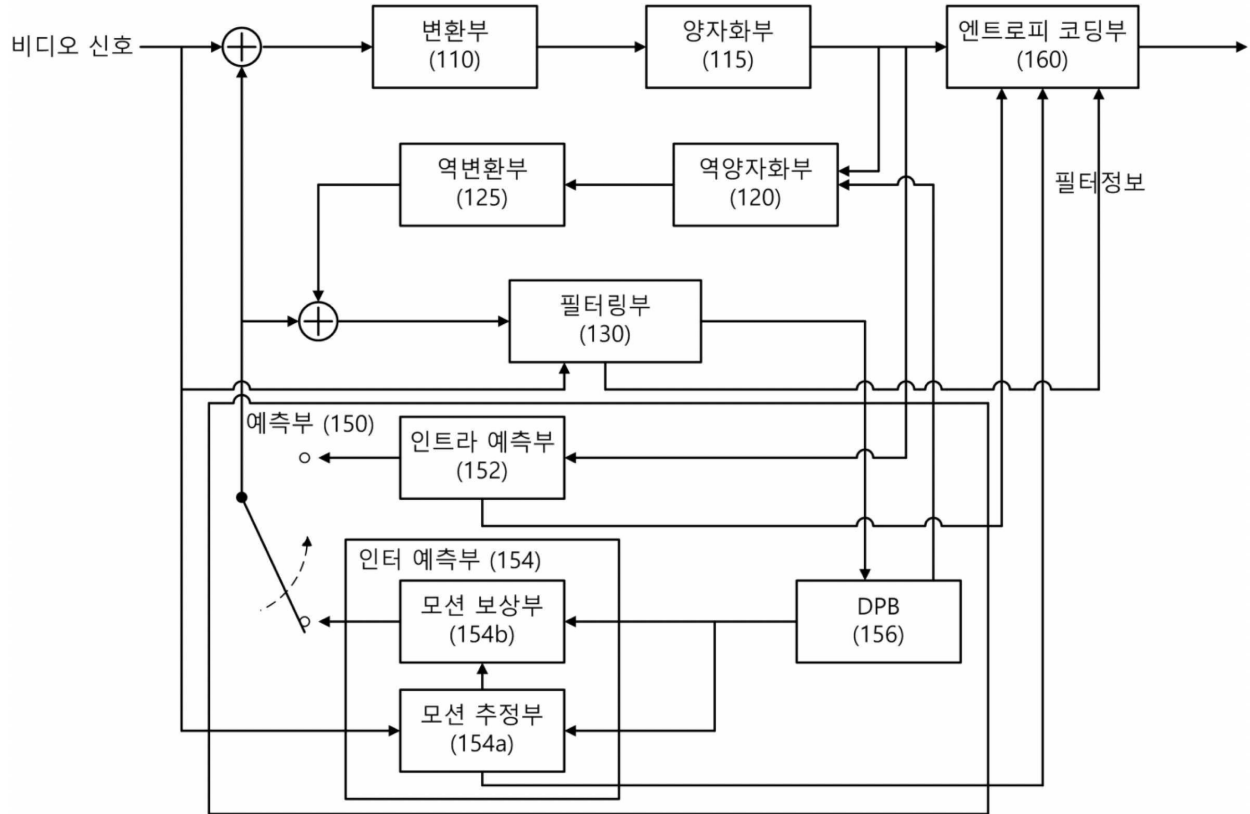
부호의 설명

- [0149] 100 : 인코딩 장치 200 : 디코딩 장치

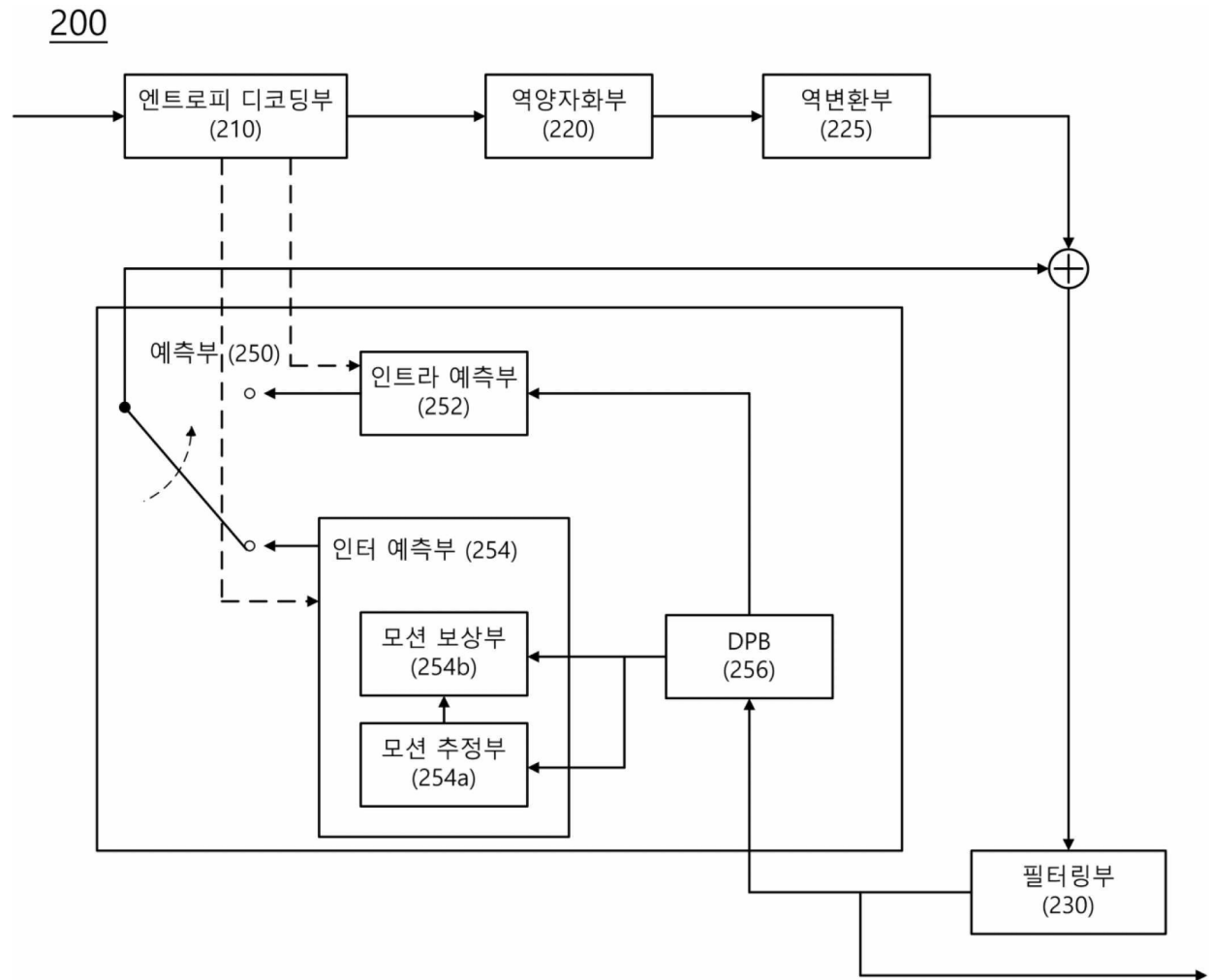
도면

도면1

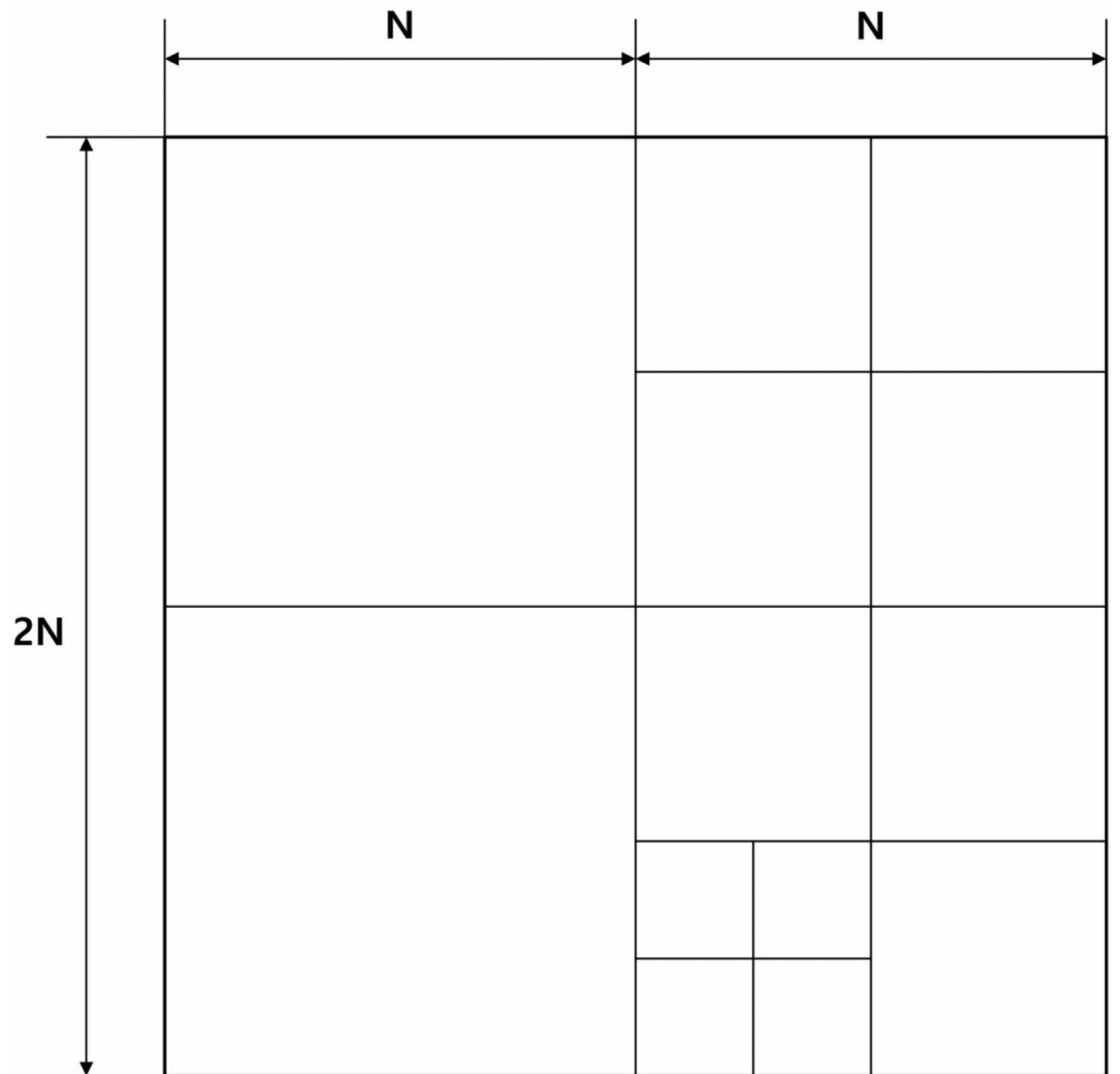
100



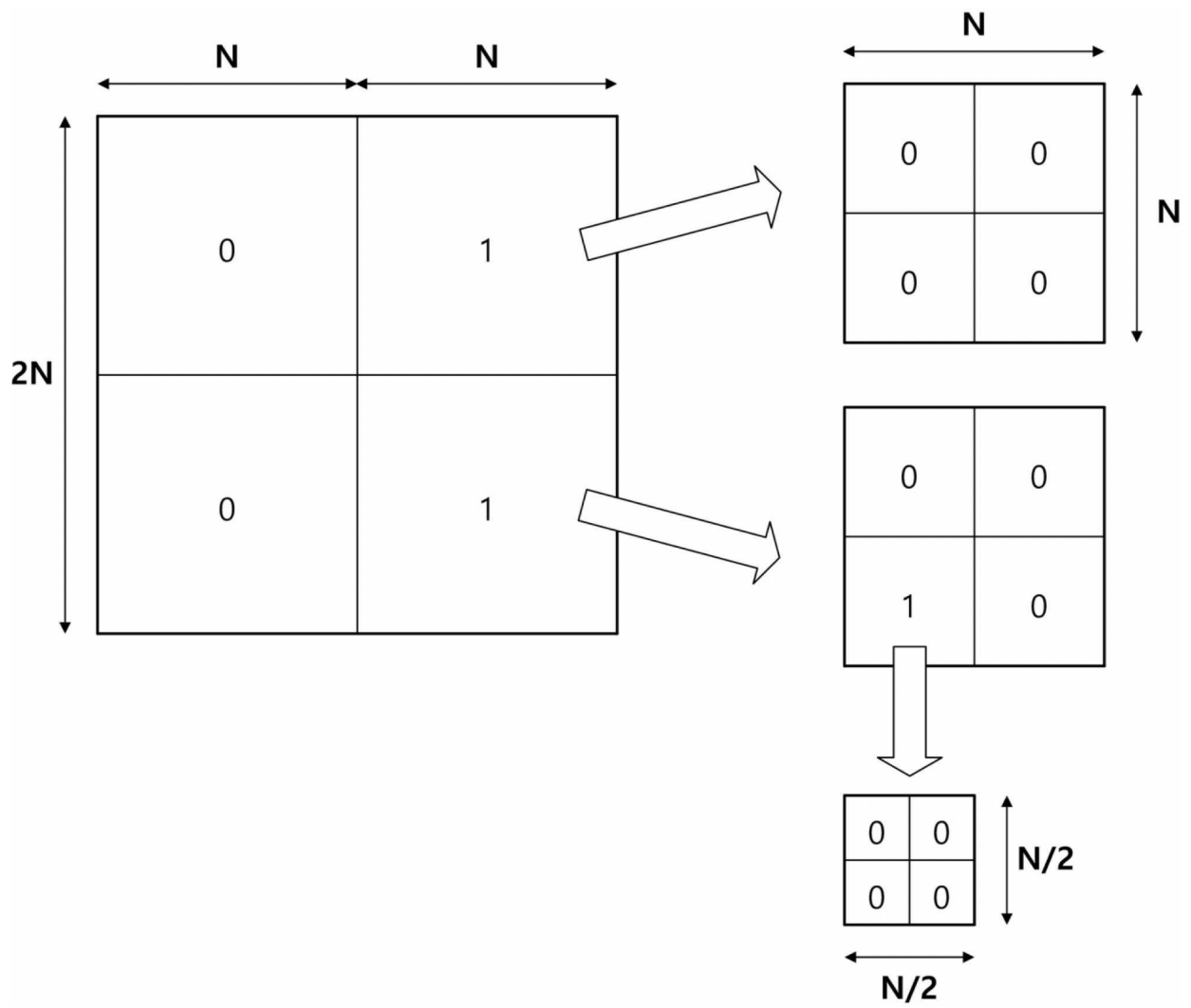
도면2



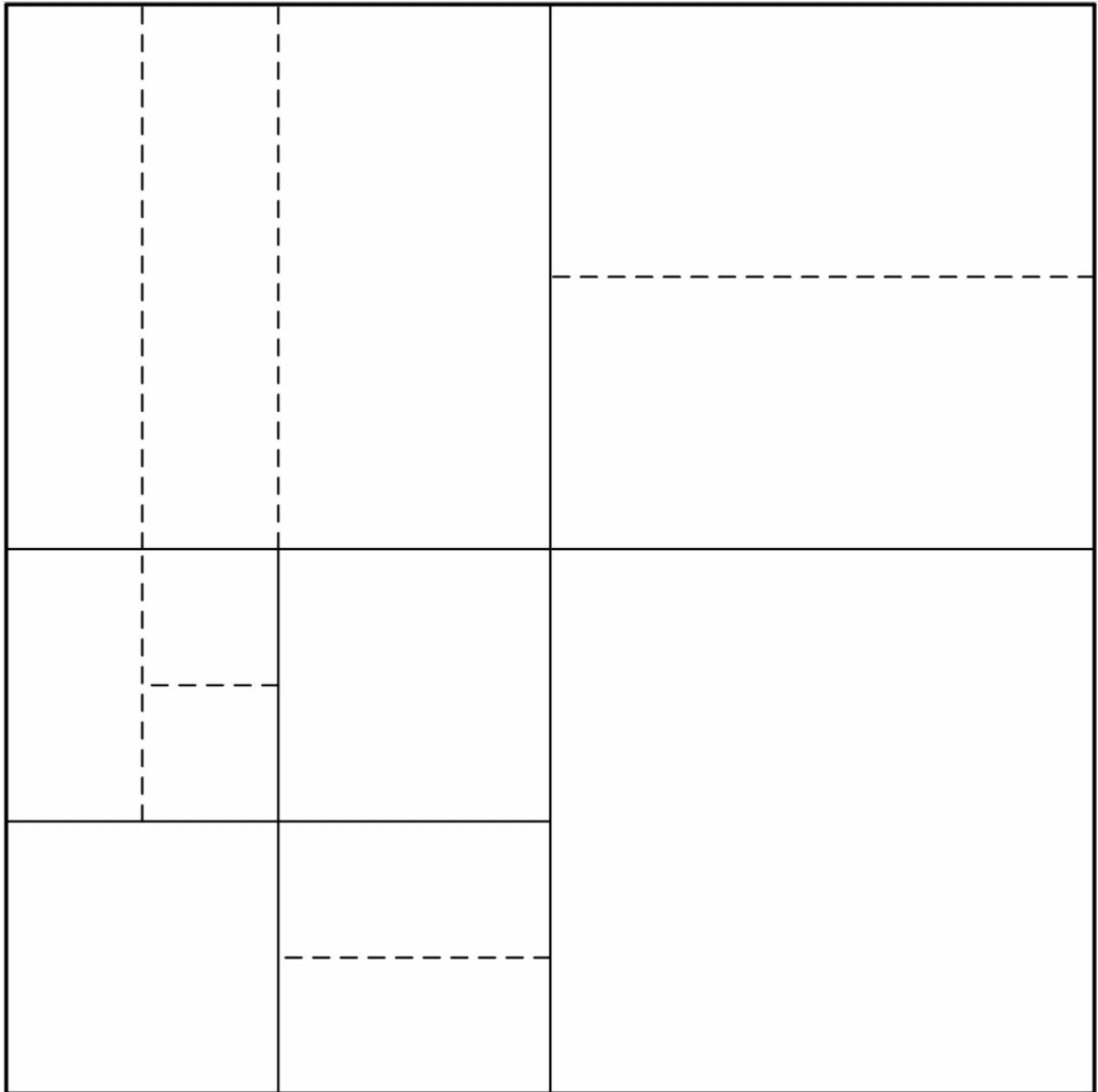
도면3



도면4



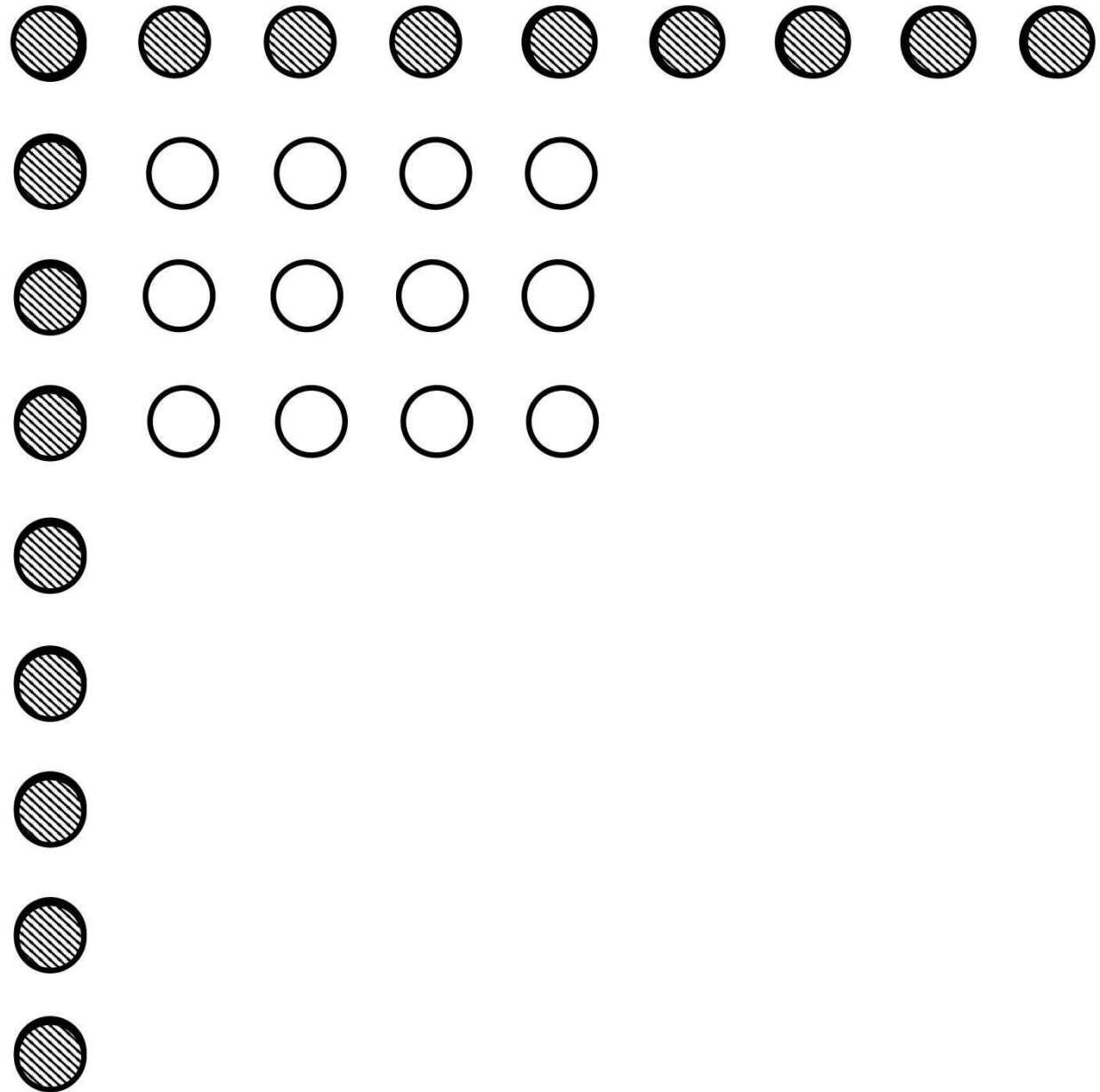
도면5



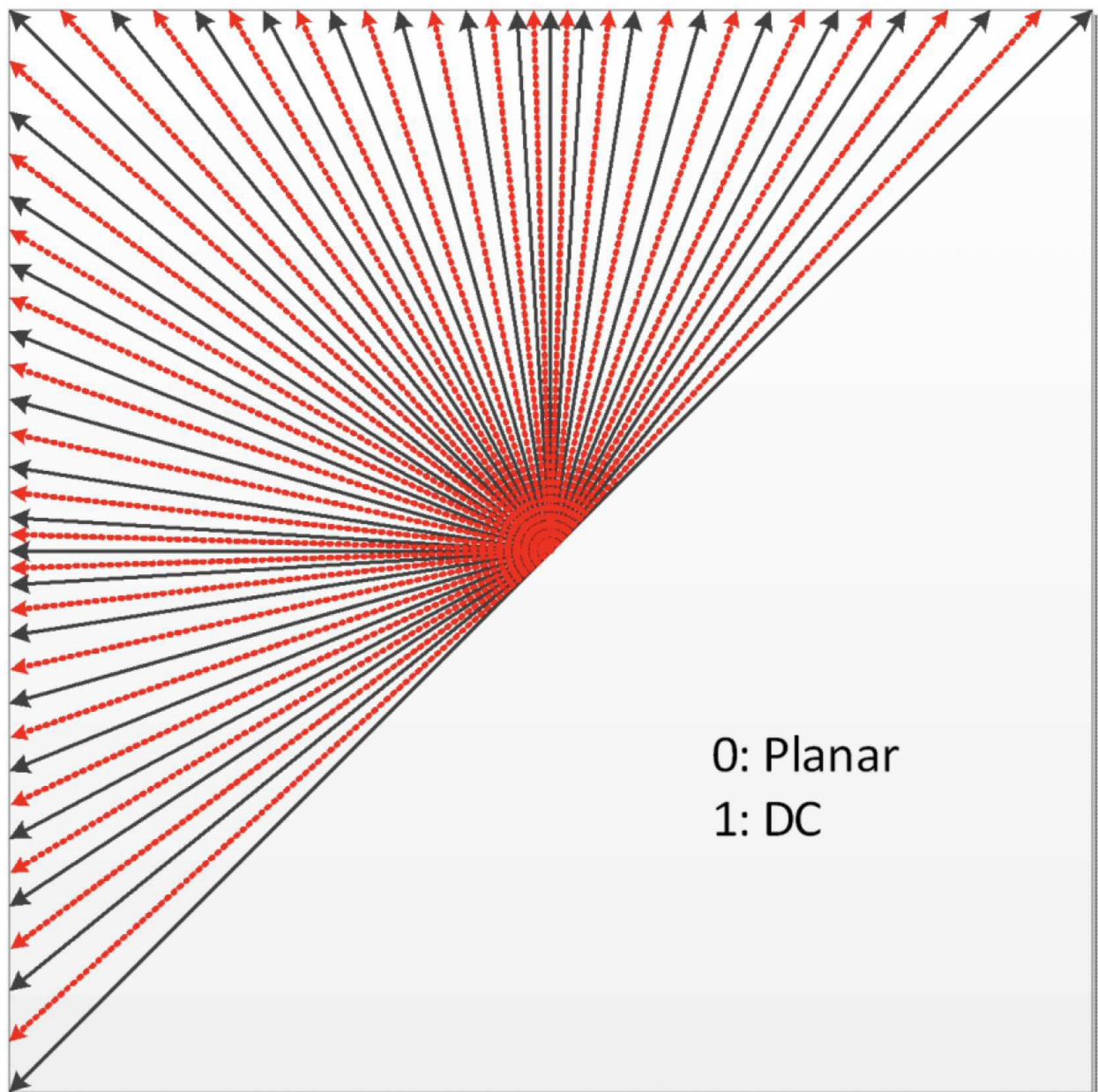
도면6

 참조 픽셀

 현재 유닛의 픽셀



도면7



도면8

Transform Type	Basis function $T_i(j)$, $i, j=0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ $\text{where } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-V	$T_i(j) = \omega_0 \cdot \omega_1 \cdot \sqrt{\frac{2}{2N-1}} \cdot \cos\left(\frac{2\pi \cdot i \cdot j}{2N-1}\right),$ $\text{where } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}, \omega_1 = \begin{cases} \sqrt{\frac{2}{N}} & j = 0 \\ 1 & j \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-I	$T_i(j) = \sqrt{\frac{2}{N+1}} \cdot \sin\left(\frac{\pi \cdot (i+1) \cdot (j+1)}{N+1}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

도면9

(a)

Intra Mode	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
V	2	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0
H	2	1	0	1	0	1	0	1	0	1	0	1	0	1	2	2	2	2
Intra Mode	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
V	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	
H	2	2	2	2	2	1	0	1	0	1	0	1	0	1	0	1	0	
Intra Mode	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
V	1	0	1	0	1	0	1	0	1	0	1	2	2	2	2	2	2	2
H	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0
Intra Mode	53	54	55	56	57	58	59	60	61	62	63	64	65	66				
V	2	2	1	0	1	0	1	0	1	0	1	0	1	0				
H	0	0	1	0	1	0	1	0	1	0	1	0	1	0				

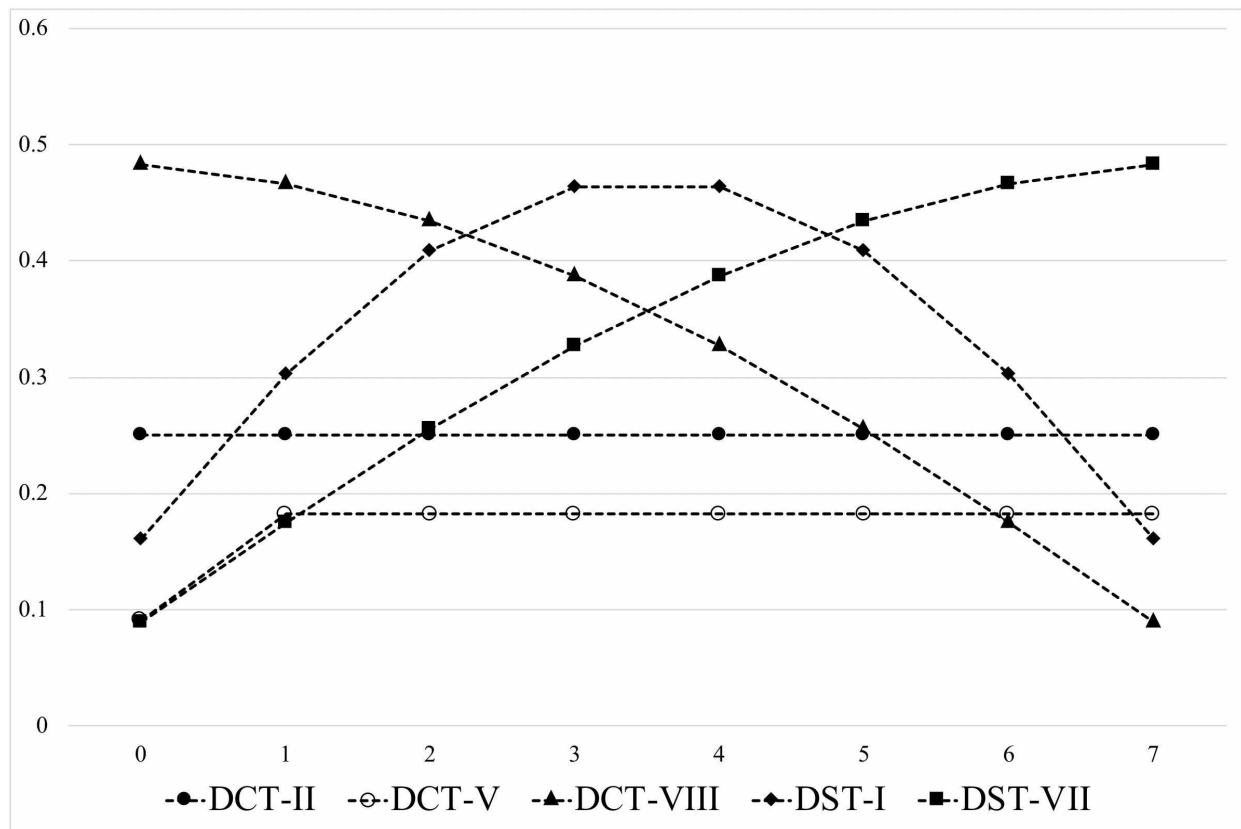
(b)

Transform Set	Transform Candidates
0	DST-VII, DCT-VIII
1	DST-VII, DST-I
2	DST-VII, DCT-V

(c)

Transform Set	Transform Candidates
0	DCT-VIII, DST-VII

도면10



도면11

(a)

Transform Type	Basis function $T_i(j)$, $i, j=0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ $\text{where } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

(b)

Intra Mode	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
V	2	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0
H	2	1	0	1	0	1	0	1	0	1	0	1	0	1	2	2	2	2
Intra Mode	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
V	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	
H	2	2	2	2	2	1	0	1	0	1	0	1	0	1	0	1	0	
Intra Mode	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
V	1	0	1	0	1	0	1	0	1	0	1	2	2	2	2	2	2	2
H	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0
Intra Mode	53	54	55	56	57	58	59	60	61	62	63	64	65	66				
V	2	2	1	0	1	0	1	0	1	0	1	0	1	0				
H	0	0	1	0	1	0	1	0	1	0	1	0	1	0				

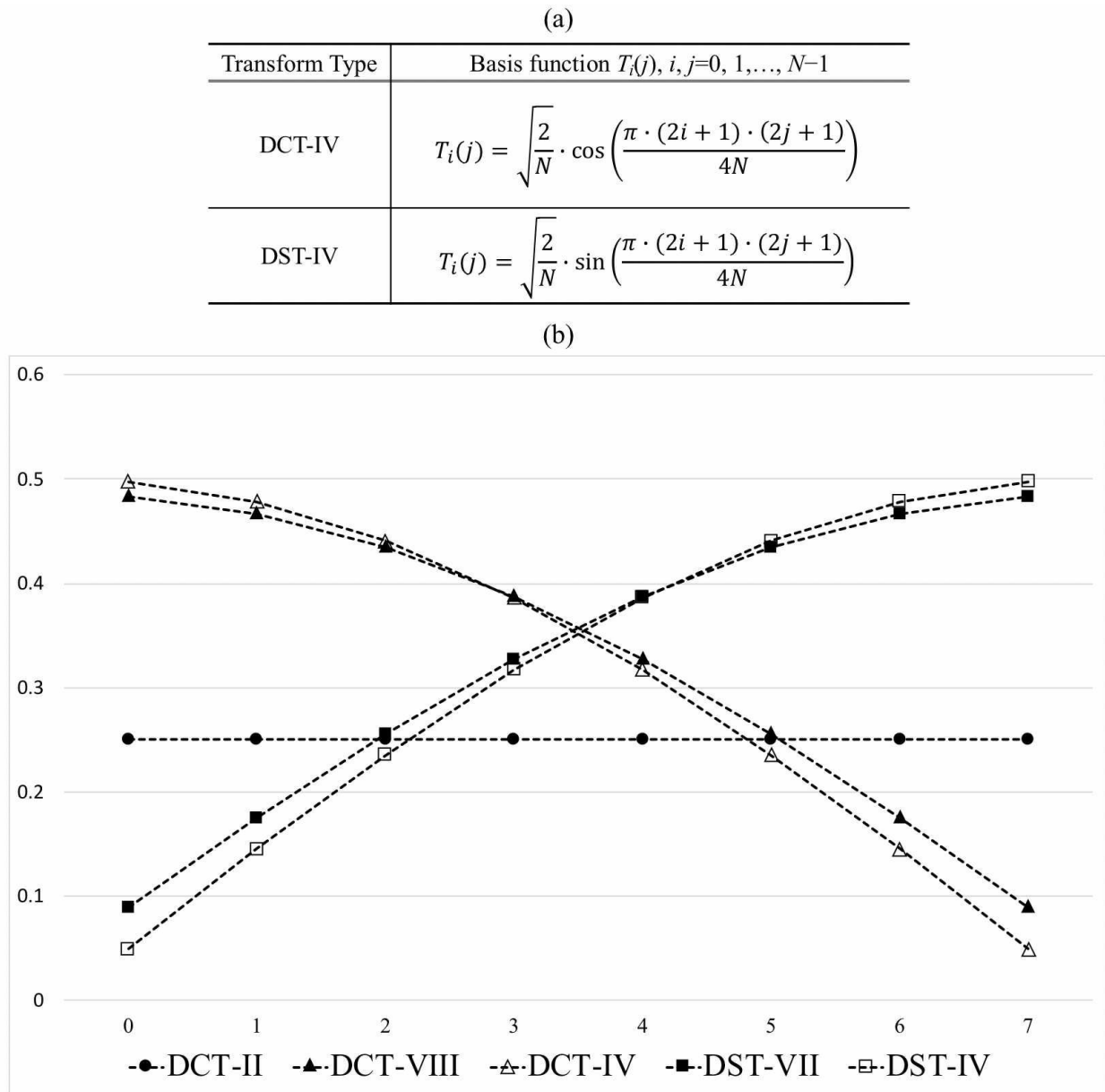
(c)

Transform Set	Transform Candidates
0	DST-VII, DCT-VIII
1	DST-VII, DCT-VIII
2	DST-VII, DCT-VIII

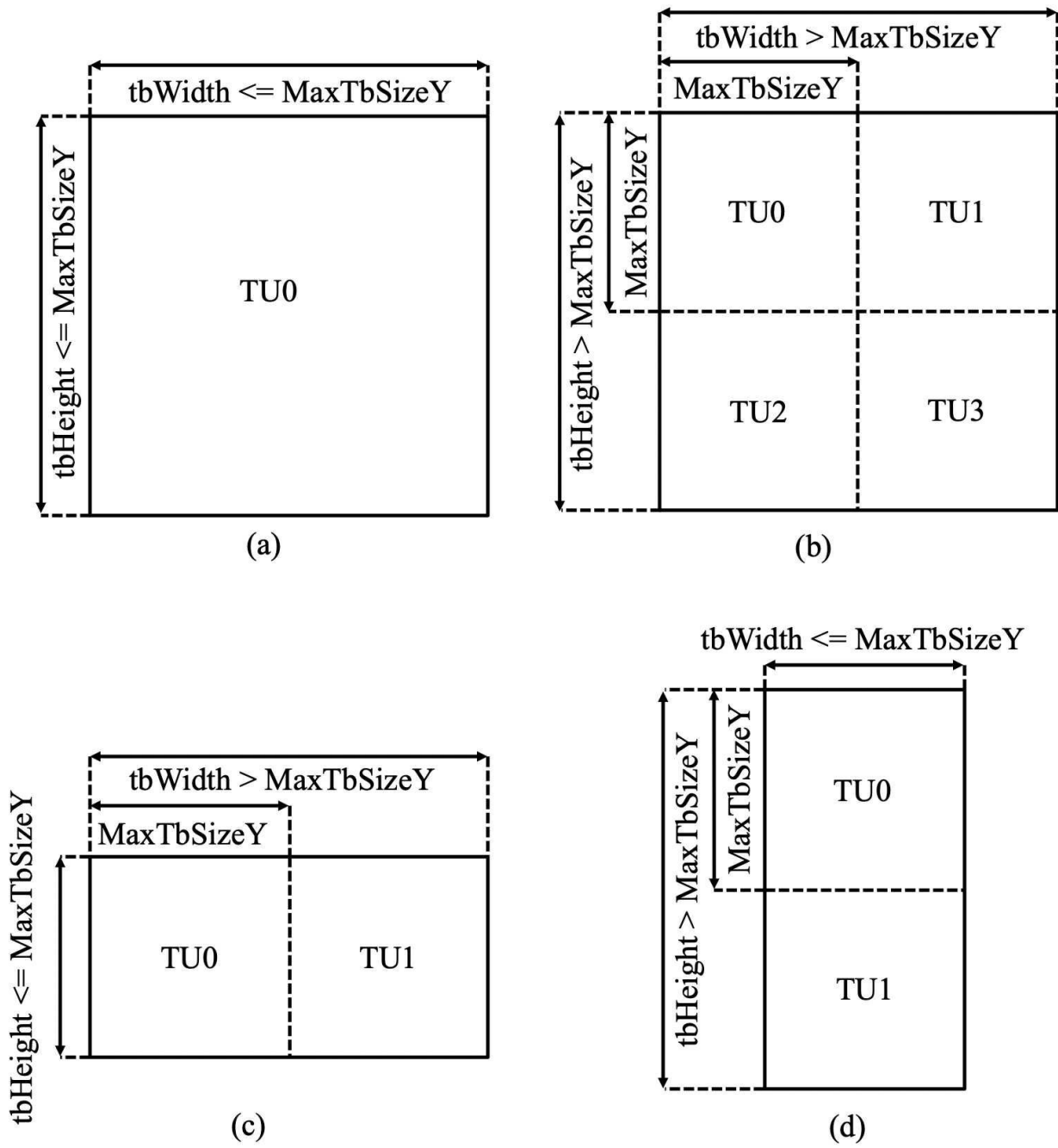
(d)

Transform Set	Transform Candidates
0	DCT-VIII, DST-VII

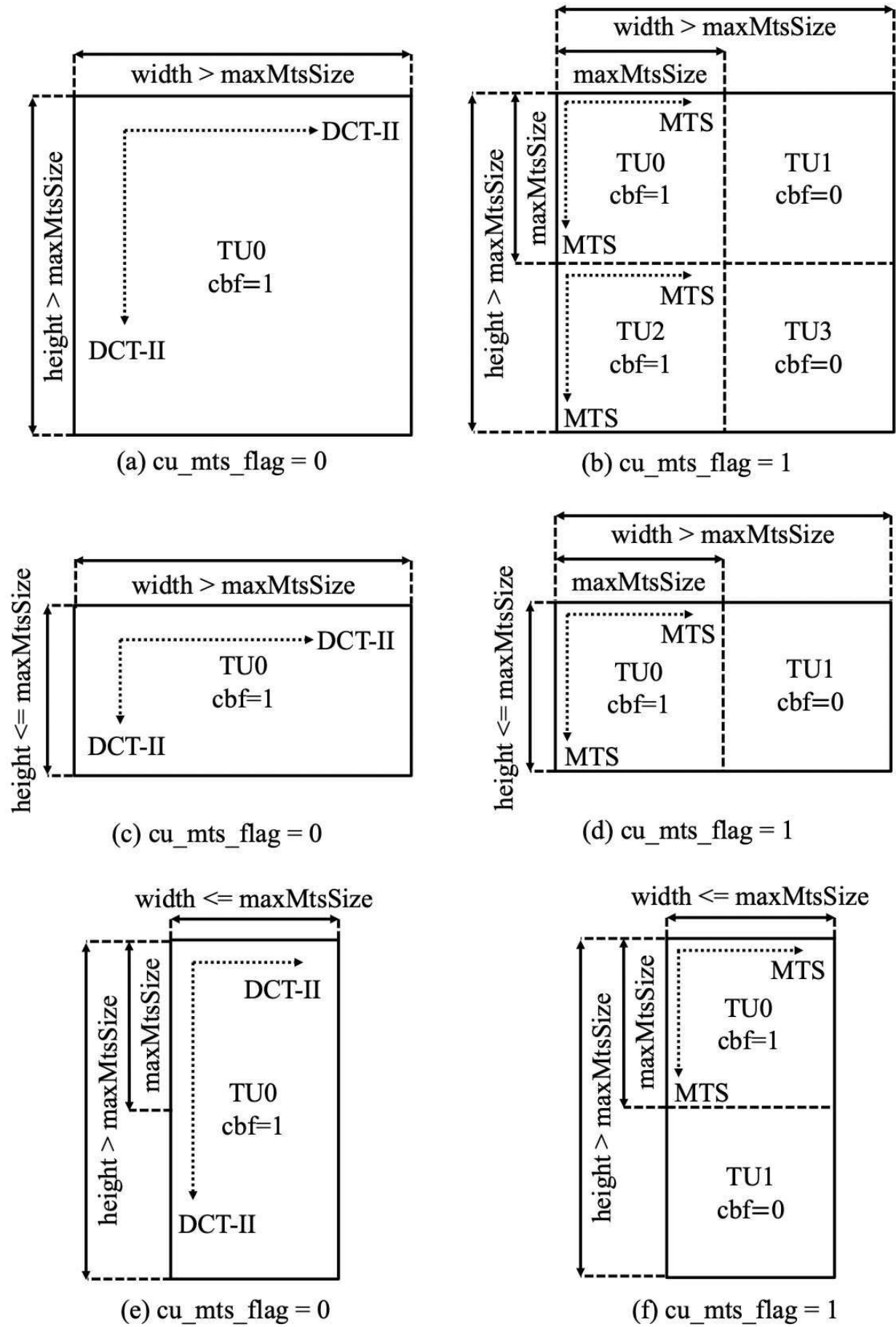
도면12



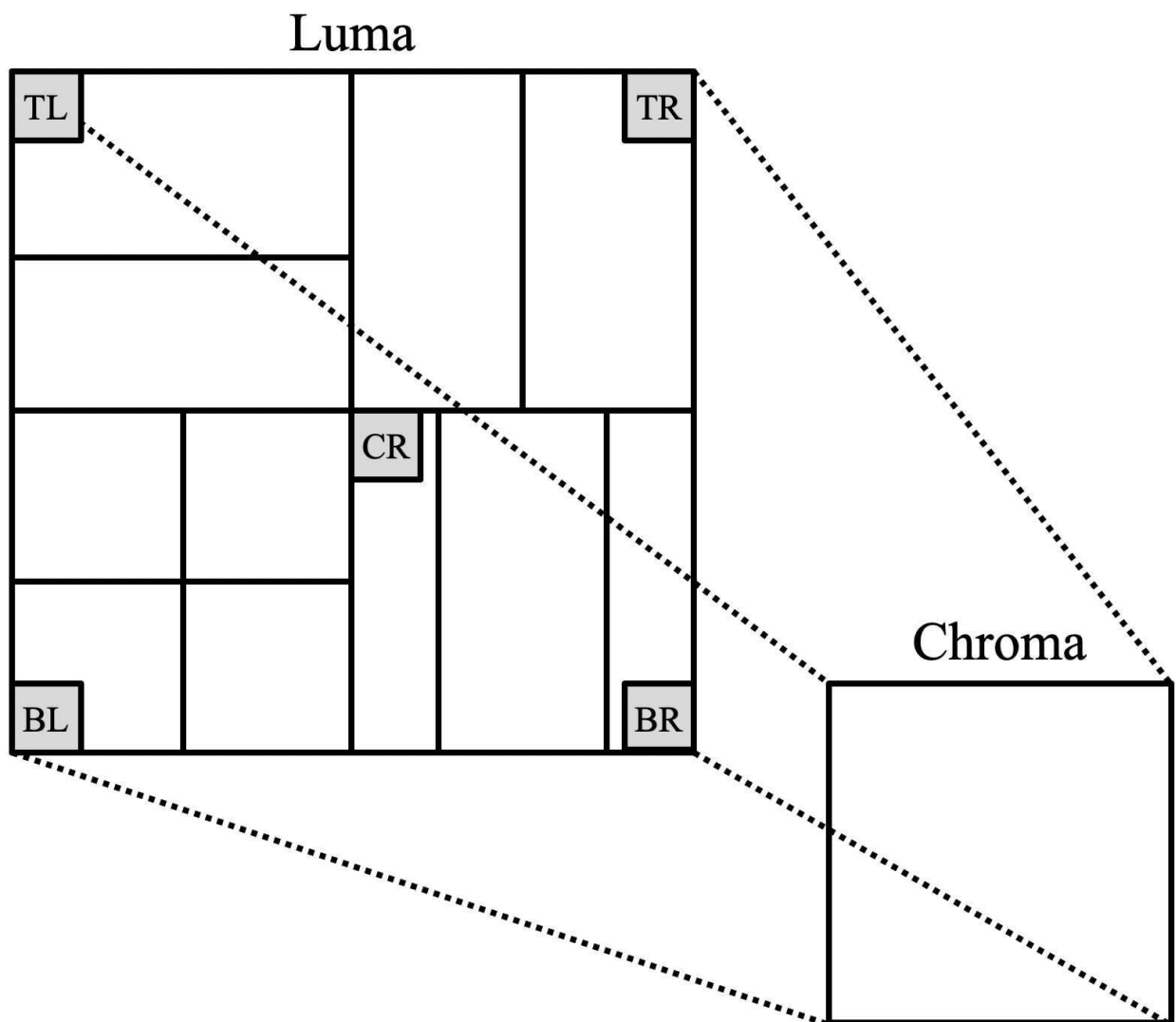
도면13



도면14



도면15



도면16

[Method I]

ctxIdx = qtDepth

[Method II]

ctxIdx = qtDepth + mttDepth

[Method III]

ctxIdx = CuPredMode == MODE_INTRA ? 0 : 1

[Method IV]

ctxIdx = CuPredMode == MODE_INTRA ? qtDepth : (NUM_MTS_CU_FLAG_CTX_INTRA + qtDepth)

[Method V]

depth = qdDepth + mttDepth

ctxIdx = CuPredMode == MODE_INTRA ? depth : (NUM_MTS_CU_FLAG_CTX_INTRA + depth)

[Method VI]

ctxIdx = clip₃(0, ctxIdx_{max}, floor(QP/N))

Where clip₃(a, b, x) = $\begin{cases} a ; x < a \\ b ; x > b \\ x ; \text{otherwise} \end{cases}$

ctxIdx_{max} represents the maximum value of ctxIdx and N is a positive integer.

[Method VII]

ctxIdx = cbWidth > cbHeight ? log2(floor(cbWidth/cbHeight)) : log2(floor(cbHeight/cbWidth))

[Method VIII]

ctxIdx = cbWidth == cbHeight ? 0 : 1

[Method IX]

ctxIdx = (condL && availableL) + (condA && availableA)