



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0025171
(43) 공개일자 2020년03월10일

<p>(51) 국제특허분류(Int. Cl.) <i>HO4N 19/12</i> (2014.01) <i>HO4N 19/184</i> (2014.01) <i>HO4N 19/61</i> (2014.01) <i>HO4N 19/625</i> (2014.01) <i>HO4N 19/70</i> (2014.01)</p> <p>(52) CPC특허분류 <i>HO4N 19/12</i> (2015.01) <i>HO4N 19/184</i> (2015.01)</p> <p>(21) 출원번호 10-2018-0102122 (22) 출원일자 2018년08월29일 심사청구일자 없음</p>	<p>(71) 출원인 주식회사 윌리스표준기술연구소 경기도 성남시 분당구 황새울로 216, 5층(수내동)</p> <p>(주)휴맥스 경기도 용인시 처인구 영문로 2 (유방동)</p> <p>(72) 발명자 정재홍 서울시 성동구 돌래11길 1, 101동 1704호</p> <p>손주형 경기도 의왕시 내손순환로 7, 308동 1402호 (뒷면에 계속)</p> <p>(74) 대리인 임국일</p>
---	---

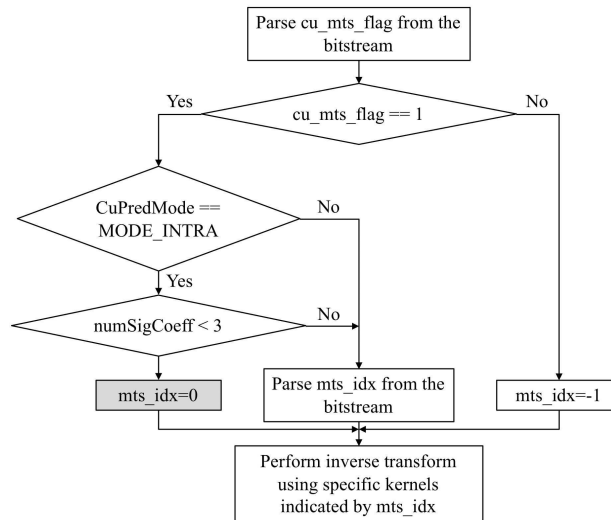
전체 청구항 수 : 총 1 항

(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 - 도18



(52) CPC특허분류

H04N 19/61 (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

곽진삼

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 높이고자 함에 있다.

과제의 해결 수단

[0004] 상기와 같은 과제를 해결하기 위해, 본 발명의 일 실시예를 따른 영상 복호화 방법은 수신한 비트스트림으로부터 변환에 적용되는 커널을 지시하는 선택스 요소를 파싱하는 단계와 변환에 적용되는 커널을 지시하는 선택스 요소가 시그널링되지 않았을 경우 이를 추론하는 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 MTS와 관련된 syntax를 도시한 도면.

도 14는 본 발명의 다른 실시예에 따른 MTS와 관련된 syntax를 도시한 도면.

도 15는 본 발명의 일 실시예에 따른 MTS와 관련된 semantics와 변환 커널을 지시하는 신택스 요소 mts_idx에 따른 가로와 세로 방향에 적용하는 변환 커널을 도시한 도면.

도 16은 본 발명의 일 실시예에 따른 복호화기에서 MTS와 관련된 처리 과정의 흐름도를 도시한 도면.

도 17은 본 발명의 제 1 방법에 따른 MTS 관련 semantics와 변환 커널을 지시하는 신택스 요소 mts_idx에 따른 가로와 세로 방향에 적용하는 변환 커널을 도시한 도면.

도 18은 본 발명의 제 1 방법에 따른 복호화기에서 MTS와 관련된 처리 과정의 흐름도를 도시한 도면.

도 19는 본 발명의 제 2 방법에 따른 MTS 관련 semantics와 변환 커널을 지시하는 신택스 요소 mts_idx에 따른 가로와 세로 방향에 적용하는 변환 커널을 도시한 도면.

도 20는 본 발명의 제 2 방법에 따른 복호화기에서 MTS와 관련된 처리 과정의 흐름도를 도시한 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 처리하

기 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, 분할될 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 정사각형의 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 블

록(prediction block)이라고 한다.

- [0034] 이하, 본 명세서에서 사용되는 유닛 이라는 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 디

코딩 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 적용적으로 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 i 는 주파수 도메인에서의 인덱스를 나타내며, 인덱스 j 는 기저 함수 내의 인덱스를 나타낸다. 즉, i 가 작아질수록 저주파 기저 함수를 나타내며, i 가 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^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 근사화 시키는 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 $T_i(j)$ 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.

[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 i 는 주파수 도메인에서의 인덱스를 나타내며, 인덱스 j 는 기저 함수 내의 인덱스를 나타낸다. 즉, i 가 작아질수록 저주파 기저 함수를 나타내며, i 가 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.
- [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 중요한 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.

[0076] 도 13(a), (b)는 본 발명의 일 실시예에 따른 MTS와 관련된 syntax를 도시한 도면이다. 변환 또는 역변환은 변환 유닛 단위에서 처리될 수 있으며, 변환 유닛의 사이즈는 부호화 유닛의 사이즈와 같으나 (부호화 유닛의 트리 구조와 동일하나), 부호화 유닛의 사이즈가 최대 변환 사이즈보다 클 경우에만 최대 변환 사이즈를 가지는 다수의 변환 유닛으로 분할될 수 있다. 예를 들어 부호화 유닛의 크기가 128×32 (가로 \times 세로)이고 최대 변환 사이즈가 64-length일 경우, 두 개의 64×32 변환 유닛으로 분할될 수 있으며, 각각의 변환 유닛에는 가로 방향으로 64-length, 세로 방향으로 32-length 변환/역변환이 적용될 수 있다.

[0077] 변환/역변환과 관련된 처리를 나타내는 transform_unit 에서는 픽처의 휘도 성분 좌상단 좌표를 기준으로 정의된 변환 블록의 휘도 성분 좌상단 좌표인 (x_0, y_0)와 변환 블록의 가로, 세로 방향으로의 크기를 나타내는 tbWidth, tbHeight, 트리 유형을 나타내는 treeType을 인자로 받는다. 부호화 유닛의 가로, 세로 크기가 모두 최대 변환 사이즈보다 작거나 같은 경우, 변환 유닛은 분할되지 않으므로 부호화 유닛의 좌상단 좌표와 해당 변환 유닛의 좌상단 좌표는 동일하다. 반면, 부호화 유닛의 가로 또는 세로 크기가 최대 변환 사이즈보다 클 경우, 변환 유닛은 부호화 유닛보다 작게 분할되므로 부호화 유닛의 좌상단 좌표와 해당 변환 유닛들의 좌상단 좌표는 다를 수 있다.

[0078] 휘도 성분에 대한 트리 구조와 색차 성분에 대한 트리 구조가 같거나 다르게 부호화될 수 있으므로 treeType 변수를 통해 트리 유형을 지시할 수 있다. treeType이 SINGLE_TREE인 경우, 휘도 성분과 색차 성분이 동일한 트리 구조로 부호화되었음을 의미하고, treeType이 DUAL_TREE_LUMA인 경우, 휘도 성분과 색차 성분의 트리가 다르게

부호화되었을 때, 현재 처리되는 트리가 휘도 성분임을 의미하며, treeType이 DUAL_TREE_CHROMA인 경우, 휘도 성분과 색차 성분의 트리가 다르게 부호화되었을 때, 현재 처리되는 트리가 색차 성분임을 의미한다.

[0079] 첫 번째 조건문에서는 현재 처리되는 트리가 휘도 성분 (Y)일 때, 좌상단 좌표가 (x0, y0)인 휘도 변환 블록이 0이 아닌 변환 계수 값을 하나 이상 포함하는지를 나타내는 syntax 요소인 tu_cbf_luma[x0][y0]를 시그널링/파싱한다. tu_cbf_luma[x0][y0]가 0인 경우, 해당 휘도 변환 블록 내의 모든 계수가 0임을 나타내므로 복호화기에서는 추가적인 파싱과 역양자화, 역변환을 거치지 않고 휘도 성분에 대한 해당 잔차 신호를 복원할 수 있다. tu_cbf_luma[x0][y0]가 1인 경우, 해당 휘도 변환 블록 내에 하나 이상의 0이 아닌 변환 계수가 존재함을 의미하므로, residual_coding 함수에서 변환 계수의 파싱과 역양자화 과정이 필요하며, 역양자화된 계수에 대하여 역변환을 적용함으로써 잔차 신호를 복원할 수 있다. tu_cbf_luma[x0][y0]는 context를 고려하여 CABAC (context adaptive binary arithmetic coding)의 regular coding engine으로 부호화/복호화될 수 있다.

[0080] 두 번째 조건문에서는 현재 처리되는 트리가 색차 성분 (Cb, Cr)일 때, 픽처의 휘도 성분 좌상단 좌표를 기준으로 블록의 휘도 성분 좌상단 좌표가 (x0, y0)인 색차 신호 Cb 변환 블록이 0이 아닌 변환 계수 값을 하나 이상 포함하는지를 나타내는 syntax 요소인 tu_cbf_cb[x0][y0]를 시그널링/파싱한다. tu_cbf_cb[x0][y0]가 0인 경우, 해당 Cb 변환 블록 내의 모든 계수가 0임을 나타내므로 복호화기에서는 추가적인 파싱과 역양자화, 역변환 과정 없이 Cb 신호에 대한 잔차 신호를 복원할 수 있다. tu_cbf_cb[x0][y0]가 1인 경우, 해당 Cb 변환 블록 내에 하나 이상의 0이 아닌 변환 계수가 존재함을 의미하므로, residual_coding 함수에서 변환 계수의 파싱과 역양자화 과정이 필요하며, 역양자화된 계수에 대하여 역변환을 적용함으로써 잔차 신호를 복원할 수 있다. tu_cbf_cb[x0][y0]는 context를 고려하여 CABAC (context adaptive binary arithmetic coding)의 regular coding engine으로 부호화/복호화될 수 있다.

[0081] Cb 신호와 마찬가지로, 픽처의 휘도 성분 좌상단 좌표를 기준으로 블록의 휘도 성분 좌상단 좌표가 (x0, y0)인 색차 신호 Cr 변환 블록이 0이 아닌 변환 계수 값을 하나 이상 포함하는지를 나타내는 syntax 요소인 tu_cbf_cr[x0][y0]를 시그널링/파싱한다. tu_cbf_cr[x0][y0]가 0인 경우, 해당 Cr 변환 블록 내의 모든 계수가 0임을 나타내므로 복호화기에서는 추가적인 파싱과 역양자화, 역변환 과정 없이 Cr 신호에 대한 잔차 신호를 복원할 수 있다. tu_cbf_cr[x0][y0]가 1인 경우, 해당 Cr 변환 블록 내에 하나 이상의 0이 아닌 변환 계수가 존재함을 의미하므로, residual_coding 함수에서 변환 계수의 파싱과 역양자화 과정이 필요하며, 역양자화된 계수에 대하여 역변환을 적용함으로써 잔차 신호를 복원할 수 있다. tu_cbf_cr[x0][y0]는 context를 고려하는 CABAC (context adaptive binary arithmetic coding)의 regular coding engine을 사용하여 부호화/복호화될 수 있다.

[0082] 세 번째 조건문에서는 부호화 유닛 단위에서 MTS를 적용하는지를 나타내는 1-bit flag의 파싱 조건을 검사한다. MTS는 화면내 예측 또는 화면간 예측에 모두 적용가능한데, 이는 SPS (sequence parameter set)에 존재하는 1-bit flag인 sps_mts_intra_enabled_flag와 sps_mts_inter_enabled_flag로 지시할 수 있다. 즉, sps_mts_intra_enabled_flag가 0인 경우, 해당 영상의 모든 화면내 예측 모드로 예측된 블록에 대해서는 residual coding syntax 내에 cu_mts_flag가 존재하지 않음을 표현하므로 화면내 예측 모드로 예측된 블록에 대해서는 MTS를 적용하지 않고, MTS와 관련된 syntax들도 부호화/복호화하지 않는다. sps_mts_intra_enabled_flag가 1인 경우, 해당 영상의 모든 화면내 예측 모드로 예측된 블록에 대해서는 residual coding syntax 내에 cu_mts_flag가 존재함을 표현하므로 화면내 예측 모드로 예측된 블록에 대해서는 MTS를 적용할 수 있고, MTS와 관련된 syntax들을 부호화 유닛 단위에서 시그널링/파싱할 수 있다.

[0083] 마찬가지로, sps_mts_inter_enabled_flag가 0인 경우, 해당 영상의 모든 화면간 예측 모드로 예측된 블록에 대해서는 cu_mts_flag가 존재하지 않음을 표현하므로 화면간 예측 모드로 예측된 블록에 대해서는 MTS를 적용하지 않고, MTS와 관련된 syntax들도 부호화/복호화하지 않는다. sps_mts_inter_enabled_flag가 1인 경우, 해당 영상의 모든 화면간 예측 모드로 예측된 블록에 대해서는 cu_mts_flag가 존재함을 표현하므로 화면간 예측 모드로 예측된 블록에 대해서는 MTS를 적용할 수 있고, MTS와 관련된 syntax들을 부호화 유닛 단위에서 시그널링/파싱할 수 있다.

[0084] 상기 기술한 sps_mts_intra_enabled_flag와 sps_mts_inter_enabled_flag는 sps, pps (picture parameter set), slice header 등의 high level syntax 어느 하나에 포함될 수 있다.

[0085] CuPredMode[x0][y0]는 픽처의 휘도 성분 좌상단 좌표를 기준으로 (x0, y0) 위치에서의 블록이 화면내 예측 모드 또는 화면간 예측 모드로 예측되었는지를 나타내는 변수로 CuPredMode[x0][y0]가 MODE_INTRA일 경우, 해당 블록은 화면내 예측 모드로 부호화되었음을 나타내고, MODE_INTER일 경우, 해당 블록은 화면간 예측 모드

로 부호화되었음을 나타낸다. 즉, 세 번째 조건문의 첫 번째 줄은 (x0, y0) 위치에서의 블록이 화면내 예측으로 부호화되었을 경우 MTS를 적용할 수 있는지를 검사하며, 세 번째 조건문의 두 번째 줄은 (x0, y0) 위치에서의 블록이 화면간 예측으로 부호화되었을 경우 MTS를 적용할 수 있는지를 검사한다. 세 번째 조건문의 세 번째 줄은 첫 번째 조건문에 따라 파싱한 tu_cbf_luma[x0][y0]를 확인한다. 이는 해당 휘도 성분 블록의 모든 계수가 0일 경우, 역변환을 수행할 필요가 없기 때문이다. 그리고 MTS는 휘도 성분에만 적용하고 색차 성분에 대해서는 HEVC와 같이 DCT-II 변환을 적용하므로 treeType이 DUAL_TREE_CHROMA인 경우에는 색차 성분과 관련된 요소들만 포함하고 있으므로 MTS와 관련된 syntax 요소들을 파싱할 필요가 없다. MTS는 너비가 32 이하이고 세로가 32 이하인 휘도 성분 변환 블록에 적용 가능하므로, 세 번째 조건문의 네 번째 줄은 변환 블록의 사이즈를 검사한다.

[0086] 결과적으로 세 번째 조건문에서 검사하는 MTS를 적용할 수 있는 조건은 다음과 같다.

[0088] i) 화면내 예측 모드로 예측된 블록에 대하여 MTS를 적용할 수 있음을 나타내는 flag가 on이고 (sps_mts_intra_enabled_flag == 1) 해당 블록이 화면내 예측 모드로 예측된 블록이거나 (CuPredMode[x0][y0] == MODE_INTRA), 화면간 예측 모드로 예측된 블록에 대하여 MTS를 적용할 수 있음을 나타내는 flag가 on이고 (sps_mts_inter_enabled_flag == 1) 해당 블록이 화면간 예측 모드로 예측된 블록인 경우 (CuPredMode[x0][y0] == MODE_INTER)

[0089] ii) 휘도 성분의 변환 블록이 하나 이상의 0이 아닌 계수를 포함함을 나타내는 syntax 요소가 1인 경우 (tu_cbf_luma[x0][y0] == 1)

[0090] iii) 현재 처리되는 블록의 트리가 색차 성분만을 포함하는 트리가 아닌 경우 (treeType != DUAL_TREE_CHROMA)

[0091] iv) 변환 블록의 너비가 32 이하이고 (tbWidth <= 32), 높이가 32 이하인 경우 (tbHeight <= 32)

[0093] 상기 기술된 네 개의 조건이 모두 참일 경우, 해당 변환 블록에 MTS를 적용하는지를 나타내는 1-bit flag (cu_mts_flag[x0][y0])를 파싱할 수 있고, 이 flag는 context를 고려하는 CABAC (context adaptive binary arithmetic coding)의 regular coding engine을 사용하여 부호화/복호화될 수 있다. cu_mts_flag[x0][y0]가 1인 경우, 휘도 성분 변환 블록의 좌상단 좌표가 픽처의 좌상단 좌표를 기준으로 (x0, y0)인 휘도 성분 변환 블록에 MTS가 적용됨을 의미하고, cu_mts_flag[x0][y0]가 0인 경우, MTS가 해당 변환 블록에 적용되지 않음을 의미하며 DCT-II와 같은 기본 커널을 사용하여 역변환을 수행할 수 있다.

[0094] 네 번째 조건문에서는 휘도 성분 Y 변환 블록의 좌상단 좌표가 (x0, y0)인 변환 블록이 0이 아닌 계수를 하나 이상 포함하는 경우 (tu_cbf_luma[x0][y0] == 1), 휘도 성분 변환 블록 내의 양자화된 변환 계수를 파싱하고 복원하며 이는 residual_coding에서 처리될 수 있다. residual_coding의 입력 인자 중 첫 번째와 두 번째는 픽처의 좌상단 좌표를 기준으로 변환 블록의 좌상단 x 좌표, y 좌표이며, 세 번째와 네 번째는 변환 블록의 너비와 높이에 밀이 2인 로그를 취한 것이며, 다섯 번째는 휘도 성분과 색차 성분을 구분하는 인덱스로써, 0인 경우 휘도 성분 Y를 의미하며, 1인 경우 색차 성분 Cb를 의미하며, 2인 경우 색차 성분 Cr을 의미한다.

[0095] 다섯 번째 조건문에서는 색차 성분 Cb 변환 블록의 좌상단 좌표가 (x0, y0)인 변환 블록이 0이 아닌 계수를 하나 이상 포함하는지를 검사하며 참인 경우 (tu_cbf_cb[x0][y0] == 1), 색차 성분 Cb 변환 블록 내의 양자화된 변환 계수를 파싱하고 복원하며 이는 residual_coding에서 처리될 수 있다. 컬러 포맷 YCbCr 이 4:2:0인 경우, 색차 신호의 변환 블록의 사이즈는 휘도 성분 변환 블록 사이즈의 절반이므로 tbWidth/2, tbHeight/2에 각각 밀이 2인 로그를 취한 값이 인자로 전달된다. 컬러 포맷 YCbCr 이 4:4:4인 경우, 색차 신호의 변환 블록의 사이즈는 휘도 성분 변환 블록 사이와 같으므로 tbWidth, tbHeight에 각각 밀이 2인 로그를 취한 값이 인자로 전달된다.

[0096] 여섯 번째 조건문에서는 색차 성분 Cr 변환 블록의 좌상단 좌표가 (x0, y0)인 변환 블록이 0이 아닌 계수를 하나 이상 포함하는지를 검사하며 참인 경우 (tu_cbf_cr[x0][y0] == 1), 색차 성분 Cr 변환 블록 내의 양자화된 변환 계수를 파싱하고 복원하며 이는 residual_coding에서 처리될 수 있다. 컬러 포맷 YCbCr 이 4:2:0인 경우, 색차 신호의 변환 블록의 사이즈는 휘도 성분 변환 블록 사이즈의 절반이므로 tbWidth/2, tbHeight/2에 각각 밀이 2인 로그를 취한 값이 인자로 전달된다. 컬러 포맷 YCbCr 이 4:4:4인 경우, 색차 신호의 변환 블록의 사이즈는 휘도 성분 변환 블록 사이와 같으므로 tbWidth, tbHeight에 각각 밀이 2인 로그를 취한 값이 인자로

전달된다.

- [0098] 도 13(b)의 residual_coding은 픽처의 좌상단 좌표를 기준으로 변환 블록의 좌상단 x 좌표인 x0, y 좌표인 y0, 변환 블록의 너비에 밀이 2인 로그를 취한 값, 변환 블록의 높이에 밀이 2인 로그를 취한 값, 컬러 인덱스 (휘도 성분 Y의 cIdx는 0, 색차 성분 Cb의 cIdx는 1, 색차 성분 Cr의 cIdx는 2)를 인자로 받으며, 변환 블록 내에 존재하는 계수를 과싱하고 복원한다.
- [0099] 첫 번째 조건문에서는 변환 블록의 좌상단 좌표가 (x0, y0)이고 해당 블록의 컬러 인덱스가 cIdx로 지시되는 변환 블록에 대해 변환을 생략할지를 나타내는 1-bit flag인 transform_skip_flag[x0][y0][cIdx]를 과싱할 조건을 검사한다. 복호화기에서 역변환이 생략되는 경우는 변환 블록내의 모든 계수가 0인 경우 (해당 cbf가 0인 경우)와 transform_skip_flag로 지시되는 경우로 구분할 수 있다. 전자의 경우, 부호화기에서 변환과 양자화를 모두 수행한 것임을 나타내며, 후자의 경우, 부호화기에서 변환이 수행되지 않았음을 의미한다.
- [0100] 다음의 세 가지 조건이 모두 참일 때, context를 고려하는 CABAC (context adaptive binary arithmetic coding)의 regular coding engine을 사용하여 transform_skip_flag[x0][y0][cIdx]를 과싱할 수 있다.
- [0102] i) High level syntax에 포함된 1-bit flag인 transform_skip_enabled_flag가 1 (on)인 경우: transform_skip_enabled_flag는 SPS (sequence parameter set), PPS (picture parameter set), slice header 중 어느 하나에 포함될 수 있으며, transform_skip_enabled_flag가 1인 경우, residual coding syntax에서 transform_skip_flag가 존재함을 나타내며, 0인 경우에는 존재하지 않음을 나타낸다.
- [0103] ii) 휘도 성분 변환 블록이 아니거나 (cIdx != 0), MTS를 적용하지 않는 경우 (cu_mts_flag[x0][y0] == 0): 변환 생략은 휘도 성분과 색차 성분 모두에 적용이 가능하다. 휘도 성분 변환 블록에 대해 MTS를 적용하는 경우 (cIdx == 0 && cu_mts_flag[x0][y0] == 1), 변환을 수행하는 것이 명백하므로 transform_skip_flag[x0][y0][cIdx]를 부호화/복호화할 필요가 없다.
- [0104] iii) 변환 블록의 너비가 4 이하이고 (log2TbWidth <= 2), 높이가 4이하인 (log2TbHeight <= 2) 경우
- [0106] 상기 기술한 조건에 따라 transform_skip_flag[x0][y0][cIdx]의 과싱 여부가 결정될 수 있으며, transform_skip_flag[x0][y0][cIdx]가 1인 경우 변환 블록의 좌상단 좌표가 (x0, y0)이고 해당 블록의 컬러 인덱스가 cIdx로 지시되는 변환 블록에 대해 변환을 적용하지 않음을 지시하고 0인 경우 변환을 적용할지 말지의 여부는 다른 syntax 요소들에 의해 결정됨을 지시한다.
- [0107] 도 13(b)의 생략된 Residual Coding Syntax에서는 양자화된 변환 계수 레벨을 과싱하고 복원하는 과정을 수행한다.
- [0108] 두 번째 조건문에서는 가로 방향과 세로 방향에 적용되는 변환 커널을 지시하는 syntax 요소인 mts_idx[x0][y0]를 과싱할 조건을 검사한다. 다음의 세 가지 조건이 모두 참인 경우 context를 고려하는 CABAC (context adaptive binary arithmetic coding)의 regular coding engine을 사용하여 mts_idx[x0][y0]를 과싱할 수 있다.
- [0110] i) MTS를 적용함을 나타내는 1-bit flag가 on이고 (cu_mts_flag[x0][y0] == 1), 휘도 성분 변환 블록인 경우 (cIdx == 0): MTS는 휘도 성분 변환 블록에만 적용할 수 있으며, cu_mts_flag[x0][y0]가 1인 경우 좌상단 좌표가 (x0, y0)인 휘도 성분 변환 블록에 MTS가 적용되므로 mts_idx[x0][y0]를 과싱하여 가로, 세로 방향에 적용되는 변환 커널을 결정할 수 있다.
- [0111] ii) 해당 블록에 변환 생략이 적용되지 않는 경우 (transform_skip_flag[x0][y0] == 0): transform_skip_flag[x0][y0]가 1인 경우, 변환 생략이 적용됨을 지시하므로 어떤 변환 커널이 사용되는지를 결정할 필요가 없다.
- [0112] iii) 해당 블록이 화면내 예측 모드로 예측되었고 (CuPredMode[x0][y0] == MODE_INTRA) 해당 블록 내에 0이 아닌 계수가 3개 이상이거나 (numSigCoeff > 2), 해당 블록이 화면간 예측 모드로 예측된 경우

((CuPredMode[x0][y0] == MODE_INTER): numSigCoeff 변수는 변환 블록 내에 존재하는 0이 아닌 계수의 개수를 나타내며, 변환 블록 내에 존재하는 0이 아닌 계수의 개수가 2개 이하인 경우 (1 또는 2), mts_idx[x0][y0]의 시그널링/파싱 없이 기 설정된 커널을 사용하여 가로 방향과 세로 방향에 변환을 수행할 수 있다. 기 설정된 변환 커널은 DST-VII일 수 있으며, DST-VII 대신 DST-IV를 사용할 수도 있다.

[0114] 도 14는 본 발명의 다른 실시예에 따른 MTS와 관련된 syntax를 도시한 도면이다. 도 13에서 도시한 syntax와 다른 점은 변환 블록의 가로, 세로 방향에 적용되는 커널을 시시하는 syntax 요소인 mts_idx를 파싱하는 조건 (두 번째 if 조건문)이다.

[0115] 도 13 (b)에서 기술한 mts_idx를 파싱하기 위한 3개의 조건 중 첫 번째 조건 (cu_mts_flag[x0][y0] && (cIdx == 0))이 참일 경우, 다음의 논리식 1이 거짓이 됨을 자명하게 알 수 있다.

[0117] [논리식 1]

[0118] $cIdx \neq 0 \ || \ cu_mts_flag[x0][y0] == 0$

[0120] 한편, 논리식 1은 도 13 (b)의 첫 번째 조건문에서 검사되는 transform_skip_flag를 파싱할 지를 결정하는 조건식의 일부이므로, 논리식 1이 거짓일 경우, transform_skip_flag는 파싱되지 않고, 기 정의된 semantics에 의해 0으로 추론될 수 있다. 따라서, mts_idx를 파싱하기 위한 3개의 조건 중 첫 번째 조건 (cu_mts_flag[x0][y0] && (cIdx == 0))이 참인 것은 다음의 논리식 2가 항상 참임을 보장할 수 있다.

[0122] [논리식 2]

[0123] $transform_skip_flag[x0][y0][cIdx] == 0$

[0124] $!transform_skip_flag[x0][y0][cIdx]$

[0126] 상기 기술한 내용에 따라, mts_idx를 파싱하기 위한 조건 중, 첫 번째 조건을 만족할 경우, 두 번째 조건을 항상 만족시킬 수 있으므로 두 번째 조건을 추가적으로 검사하는 것은 불필요한 요소로 볼 수 있다.

[0127] 다음의 두 가지 조건이 모두 참인 경우 context를 고려하는 CABAC (context adaptive binary arithmetic coding)의 regular coding engine을 사용하여 mts_idx[x0][y0]를 파싱할 수 있다.

[0129] i) MTS를 적용함을 나타내는 1-bit flag가 on이고 (cu_mts_flag[x0][y0] == 1), 휘도 성분 변환 블록인 경우 (cIdx == 0): MTS는 휘도 성분 변환 블록에만 적용할 수 있으며, cu_mts_flag[x0][y0]가 1인 경우 좌상단 좌표가 (x0, y0)인 휘도 성분 변환 블록에 MTS가 적용되므로 mts_idx[x0][y0]를 파싱하여 가로, 세로 방향에 적용되는 변환 커널을 결정할 수 있다. 현재 휘도 성분 블록에 MTS를 적용한다는 것은 변환을 생략하지 않는다는 것을 의미하므로 (transform_skip_flag[x0][y0]), transform_skip_flag 값을 확인하는 과정은 생략될 수 있다.

[0130] ii) 해당 블록이 화면내 예측 모드로 예측되었고 (CuPredMode[x0][y0] == MODE_INTRA) 해당 블록 내에 0이 아닌 계수가 3개 이상이거나 (numSigCoeff > 2), 해당 블록이 화면간 예측 모드로 예측된 경우 ((CuPredMode[x0][y0] == MODE_INTER): numSigCoeff 변수는 변환 블록 내에 존재하는 0이 아닌 계수의 개수를 나타내며, 변환 블록 내에 존재하는 0이 아닌 계수의 개수가 2개 이하인 경우 (1 또는 2), mts_idx[x0][y0]의 시그널링/파싱 없이 기 설정된 커널을 사용하여 가로 방향과 세로 방향에 변환을 수행할 수 있다. 기 설정된 변환 커널은 DST-VII일 수 있으며, DST-VII 대신 DST-IV를 사용할 수도 있다.

[0132] 도 15는 본 발명의 일 실시예에 따른 MTS와 관련된 semantics와 mts_idx에 따른 가로와 세로 방향에 적용하는

변환 커널을 도시한 도면이다.

- [0133] mts_idx[x0][y0]는 픽처의 휘도 성분 좌상단 좌표를 기준으로 휘도 성분 좌상단 좌표가 (x0, y0)인 변환 블록에 대하여 가로, 세로 방향에 적용하는 변환 커널을 지시한다. 이때, mts_idx[x0][y0]가 존재하지 않을 때 (시그널링/파싱되지 않았을 때)에는 -1로 추론할 수 있다.
- [0134] mts_idx의 MSB (most significant bit)는 예측 모드에 따라 기 설정된 변환 커널 세트에서 세로 방향에 적용되는 변환 커널 후보 인덱스를 지시할 수 있으며, LSB (least significant bit)는 예측 모드에 따라 기 설정된 변환 커널 세트에서 가로 방향에 적용되는 변환 커널 후보 인덱스를 지시할 수 있다. 즉, 0인 경우 변환 커널 세트 내의 첫 번째 변환 커널을 나타낼 수 있으며, 1인 경우 변환 커널 세트 내의 두 번째 변환 커널을 나타낼 수 있다.
- [0135] mts_idx에 따른 수평, 수직 방향에 적용되는 변환 커널은 다음과 같이 해석할 수 있다.
 - [0137] i) mts_idx가 -1인 경우: 수평 - DCT-II, 수직 - DCT-II
 - [0138] ii) mts_idx가 0인 경우: 수평 - 변환 세트 내 첫 번째 후보, 수직 - 변환 세트 내 첫 번째 후보
 - [0139] iii) mts_idx가 1인 경우: 수평 - 변환 세트 내 두 번째 후보, 수직 - 변환 세트 내 첫 번째 후보
 - [0140] iv) mts_idx가 2인 경우: 수평 - 변환 세트 내 첫 번째 후보, 수직 - 변환 세트 내 두 번째 후보
 - [0141] v) mts_idx가 3인 경우: 수평 - 변환 세트 내 두 번째 후보, 수직 - 변환 세트 내 두 번째 후보
- [0143] 상기 기술한 mts_idx에 따른 수평, 수직 방향에 사용되는 변환 커널은 mts_idx의 시그널링 방법에 의존적이지 않다. 도 15의 실시예에서는 mts_idx를 고정 길이의 2-bit로 시그널링하는 예를 기술하였으나, mts_idx를 truncated binary와 같은 가변 길이로 이진화할 때에도 mts_idx 값에 따른 수평, 수직 방향에 사용되는 변환 커널은 i) 내지 v)에서 기술한 규칙에 따라 동일하게 사용할 수 있다.
- [0144] 또한 상기 기술한 mts_idx에 따른 수평, 수직 방향에 사용되는 변환 커널은 변환 세트를 구성하는 방법에 의존적이지 않다. 도 11의 MTS 실시예에서는 화면내 예측에 대해 변환 세트 구성을 {DST-VII, DCT-VIII}, 화면간 예측에 대해 {DCT-VIII, DST-VII}으로 구성하는 예를 도시하였으나, 변환 세트를 구성하는 변환 커널 후보가 달라지거나, 사용할 수 있는 변환 세트 수가 달라져도, i) 내지 v)에서 기술한 규칙에 따라 동일하게 사용할 수 있다. 예를 들어, 화면내 예측에 대해 {DST-IV, DCT-IV}, 화면간 예측에 대해 {DCT-IV, DST-IV}로 변환 세트를 구성하는 경우에도 동일하게 적용할 수 있다.
- [0145] trTypeHor, trTypeVer는 각각 가로, 세로 방향에 적용하는 변환 커널을 나타내는 변수로, trTypeHor 또는 trTypeVer가 0일 때에는 DCT-II가 해당 방향의 변환에 사용될 수 있으며, 1일 때에는 DST-VII이 해당 방향의 변환에 사용될 수 있고, 2일 때에는 DCT-VIII이 해당 방향의 변환에 사용될 수 있다. 도 12에서 기술하였듯이 구현의 복잡도를 고려하여 DST-VII은 DST-IV로 대체될 수도 있으며, DCT-VIII은 DCT-IV로 대체될 수도 있다.
- [0146] 도 16는 본 발명의 일 실시예에 따른 복호화기에서 MTS와 관련된 처리 과정의 흐름도를 도시한 도면이다.
- [0147] 수신한 비트스트림으로부터 현재 블록에 MTS를 적용할지를 나타내는 1-bit flag인 cu_mts_flag를 파싱할 조건이 참인 경우 cu_mts_flag를 파싱한다. cu_mts_flag가 0인 경우, 현재 블록에 MTS를 적용하지 않음을 나타내므로 가로, 세로 방향에 적용하는 변환 커널을 지시하는 syntax 요소인 mts_idx를 파싱하지 않고 -1로 추론한다. mts_idx가 -1인 경우, 기본 커널인 DCT-II를 가로, 세로 방향에 모두 적용할 수 있다. cu_mts_flag가 1인 경우 해당 휘도 성분 변환 블록에 MTS를 적용함을 지시하고, mts_idx를 파싱할 조건이 참인 경우 mts_idx를 파싱한다. 현재 블록의 예측 방법이 화면내 예측인지 화면간 예측인지를 나타내는 요소가 MODE_INTRA인 경우 (해당 블록이 화면내 예측 모드로 예측되었을 경우), 해당 휘도 성분 변환 블록 내의 0이 아닌 계수의 개수가 2 이하인 경우 (numSigCoeff<3), mts_idx는 -1로 추론한다. 반면 해당 블록이 화면내 예측 모드로 예측되었고, 해당 휘도 성분 블록 내의 0이 아닌 계수의 개수가 3 이상인 경우 mts_idx를 비트스트림으로부터 파싱한다. 현재 블록이 화면간 예측 모드를 사용하여 예측되었을 경우에는 추가적인 조건 없이 mts_idx를 비트스트림으로부터 파싱한다.

- [0149] 도 13 내지 16에서 기술한 MTS 기법에서는 mts_idx 가 존재하지 않을 때 (시그널링/과실되지 않았을 때), mts_idx 를 추론하는 방식에 문제점이 존재한다. cu_mts_flag 가 1이고 화면내 예측 모드로 예측된 블록이 두 개 이하의 0이 아닌 계수를 가질 때, mts_idx 는 도 14에서 도시한 것처럼 정의된 semantics에 의해 cu_mts_flag 가 0일 때와 동일하게 -1로 추론된다. 따라서 MTS를 적용하지 않을 때와 같이 가로, 세로 방향 모두 DCT-II가 적용된다. 이는 부호화 효율의 저하를 야기할 수 있으므로, cu_mts_flag 가 1이고 화면내 예측 모드로 예측된 블록이 두 개 이하의 0이 아닌 계수를 가질 때, mts_idx 는 cu_mts_flag 가 0일 때 추론되는 mts_idx 와 다른 값으로 추론되어야 하고, 기본 커널인 DCT-II가 아닌 DST-VII를 사용하여 수평, 수직 방향으로 역변환을 수행해야 한다. DST-VII이 DST-IV로 대체되는 경우, DST-IV를 사용할 수도 있다.
- [0151] 도 17은 본 발명의 제 1 방법에 따른 MTS 관련 semantics와 mts_idx 에 따른 가로와 세로 방향에 적용하는 변환 커널을 도시한 도면이다.
- [0152] 제 1 방법은 mts_idx 의 값을 추가하지 않고 시그널링되는 첫 번째 mts_idx 를 활용하여 cu_mts_flag 가 0일 때와 다른 mts_idx 값으로 설정하는 것이다. $mts_idx[x0][y0]$ 는 픽처의 휘도 성분 좌상단 좌표를 기준으로 휘도 성분 좌상단 좌표가 $(x0, y0)$ 인 변환 블록에 대하여 가로, 세로 방향에 적용하는 변환 커널을 지시한다. 이때, $mts_idx[x0][y0]$ 가 존재하지 않을 때 (시그널링/과실되지 않았을 때)에는 다음의 조건에 따라 추론할 수 있다.
- [0154] i) 픽처의 휘도 성분 좌상단 좌표를 기준으로 휘도 성분 좌상단 좌표가 $(x0, y0)$ 인 휘도 성분 변환 블록에 MTS를 적용함을 나타내는 1-bit flag가 0인 경우 ($cu_mts_flag[x0][y0] == 0$ 인 경우), -1로 추론할 수 있다.
- [0155] ii) 그렇지 않은 경우 ($cu_mts_flag[x0][y0] == 1$ 인 경우), 0으로 추론할 수 있다.
- [0156] $cu_mts_flag[x0][y0]$ 가 1일 때에 $mts_idx[x0][y0]$ 가 존재하지 않는 경우는 해당 블록이 화면내 예측 모드로 예측되었고 블록 내의 0이 아닌 계수의 개수가 2인 경우만 존재하므로, $cu_mts_flag[x0][y0]$ 가 1인 지를 나타내는 조건이 해당 블록이 화면내 예측으로 부호화되었고 블록 내의 0이 아닌 계수의 개수가 2보다 작음을 표현할 수 있다. 도 15의 표에서 시그널링되는 첫 번째 mts_idx 는 0이므로, $mts_idx[x0][y0]$ 는 0으로 추론된다.
- [0158] mts_idx 의 MSB (most significant bit)는 예측 모드에 따라 기 설정된 변환 커널 세트에서 세로 방향에 적용되는 변환 커널 후보 인덱스를 지시할 수 있으며, LSB (least significant bit)는 예측 모드에 따라 기 설정된 변환 커널 세트에서 가로 방향에 적용되는 변환 커널 후보 인덱스를 지시할 수 있다. 즉, 0인 경우 변환 커널 세트 내의 첫 번째 변환 커널을 나타낼 수 있으며, 1인 경우 변환 커널 세트 내의 두 번째 변환 커널을 나타낼 수 있다.
- [0159] mts_idx 에 따른 수평, 수직 방향에 적용되는 변환 커널은 다음과 같이 해석할 수 있다.
- [0161] i) mts_idx 가 -1인 경우: 수평 - DCT-II, 수직 - DCT-II
- [0162] ii) mts_idx 가 0인 경우: 수평 - 변환 세트 내 첫 번째 후보, 수직 - 변환 세트 내 첫 번째 후보
- [0163] iii) mts_idx 가 1인 경우: 수평 - 변환 세트 내 두 번째 후보, 수직 - 변환 세트 내 첫 번째 후보
- [0164] iv) mts_idx 가 2인 경우: 수평 - 변환 세트 내 첫 번째 후보, 수직 - 변환 세트 내 두 번째 후보
- [0165] v) mts_idx 가 3인 경우: 수평 - 변환 세트 내 두 번째 후보, 수직 - 변환 세트 내 두 번째 후보
- [0167] 상기 기술한 mts_idx 에 따른 수평, 수직 방향에 사용되는 변환 커널은 mts_idx 의 시그널링 방법에 의존적이지 않다. 도 17의 실시예에서는 mts_idx 를 고정 길이의 2-bit로 시그널링하는 예를 기술하였으나, mts_idx 를 truncated binary와 같은 가변 길이로 이진화할 때에도 mts_idx 값에 따른 수평, 수직 방향에 사용되는 변환 커널은 i) 내지 v)에서 기술한 규칙에 따라 동일하게 사용할 수 있다.

- [0168] 또한 상기 기술한 mts_idx에 따른 수평, 수직 방향에 사용되는 변환 커널은 변환 세트를 구성하는 방법에 의존적이지 않다. 도 11의 MTS 실시예에서는 화면내 예측에 대해 변환 세트 구성을 {DST-VII, DCT-VIII}, 화면간 예측에 대해 {DCT-VIII, DST-VII}으로 구성하는 예를 도시하였으나, 변환 세트를 구성하는 변환 커널 후보가 달라지거나, 사용할 수 있는 변환 세트 수가 달라져도, i) 내지 v)에서 기술한 규칙에 따라 동일하게 사용할 수 있다. 예를 들어, 화면내 예측에 대해 {DST-IV, DCT-IV}, 화면간 예측에 대해 {DCT-IV, DST-IV}로 변환 세트를 구성하는 경우에도 동일하게 적용할 수 있다.
- [0169] trTypeHor, trTypeVer는 각각 가로, 세로 방향에 적용하는 변환 커널을 나타내는 변수로, trTypeHor 또는 trTypeVer가 0일 때에는 기본 커널인 DCT-II가 해당 방향의 변환에 사용될 수 있으며, 1일 때에는 DST-VII이 해당 방향의 변환에 사용될 수 있고, 2일 때에는 DCT-VIII이 해당 방향의 변환에 사용될 수 있다. 도 12에서 기술하였듯이 구현의 복잡도를 고려하여 DST-VII은 DST-IV로 대체될 수도 있으며, DCT-VIII은 DCT-IV로 대체될 수도 있다.
- [0170] 현재 블록이 화면내 예측 모드로 예측되었고 변환 블록 내의 0이 아닌 계수의 개수가 2 이하인 경우, mts_idx는 0으로 추론되고 수평, 수직 방향으로 모두 DST-VII을 적용할 수 있게 된다. DST-VII이 DST-IV로 대체되는 경우, DST-VII 대신 DST-IV를 적용할 수도 있다.
- [0171] 본 발명은 도 17의 표에서 정의된 mts_idx 값에 국한되지 않고 적용될 수 있다. 도 17의 실시예에서는 시그널링되는 첫 번째 mts_idx가 0인 예를 도시하였으나, 0이 아닌 다른 값이더라도 적용할 수 있다. 예를 들어 cu_mts_flag가 0일 때 추론되는 mts_idx가 3이고, 시그널링되는 mts_idx가 4, 5, 6, 7인 경우, cu_mts_flag가 1일 때 추론되는 mts_idx는 4가 되며 mts_idx가 4일 때의 변환 커널을 적용할 수 있다.
- [0172] 또한 도 17의 실시예에서는 mts_idx가 2-bit의 고정 비트 길이로 이진화되어 시그널링하는 예를 도시하였으나, 심볼을 가변 길이로 나타내는 truncated unary, truncated binary 등에도 동일한 규칙으로 적용할 수 있다.
- [0173] 도 18은 본 발명의 제 1 방법에 따른 복호화기에서 MTS와 관련된 처리 과정의 흐름도를 도시한 도면이다.
- [0174] 수신한 비트스트림으로부터 현재 블록에 MTS를 적용할지를 나타내는 1-bit flag인 cu_mts_flag를 파싱할 조건이 참인 경우 cu_mts_flag를 파싱한다. cu_mts_flag가 0인 경우, 현재 블록에 MTS를 적용하지 않음을 나타내므로 가로, 세로 방향에 적용하는 변환 커널을 지시하는 syntax 요소인 mts_idx를 파싱하지 않고 -1로 추론한다. mts_idx가 -1인 경우, 기본 커널인 DCT-II를 가로, 세로 방향에 모두 적용할 수 있다. cu_mts_flag가 1인 경우 해당 휘도 성분 변환 블록에 MTS를 적용함을 지시하고, mts_idx를 파싱할 조건이 참인 경우 mts_idx를 파싱한다. 현재 블록의 예측 방법이 화면내 예측인지 화면간 예측인지를 나타내는 요소가 MODE_INTRA인 경우 (해당 블록이 화면내 예측 모드로 예측되었을 경우), 해당 휘도 성분 변환 블록 내의 0이 아닌 계수의 개수가 2 이하인 경우 (numSigCoeff<3), mts_idx는 0으로 추론된다. 반면 해당 블록이 화면내 예측 모드로 예측되었고, 해당 휘도 성분 블록 내의 0이 아닌 계수의 개수가 3 이상인 경우 mts_idx를 비트스트림으로부터 파싱한다. 현재 블록이 화면간 예측 모드를 사용하여 예측되었을 경우에는 추가적인 조건 없이 mts_idx를 비트스트림으로부터 파싱한다.
- [0176] 도 19는 본 발명의 제 2 방법에 따른 MTS 관련 semantics와 mts_idx에 따른 가로와 세로 방향에 적용하는 변환 커널을 도시한 도면이다.
- [0177] 제 2 방법은 기 설정된 mts_idx와 중복되지 않는 새로운 값을 추가하고 이때 사용하는 변환 커널을 시그널링되는 첫 번째 mts_idx에서 사용하는 변환 커널로 설정하는 것이다. 화면간 예측된 블록에 대해서는 cu_mts_flag가 1일 때, mts_idx가 추론되는 경우가 없으므로 이에 대한 커널은 정의하지 않을 수 있다.
- [0178] mts_idx[x0][y0]는 픽처의 휘도 성분 좌상단 좌표를 기준으로 휘도 성분 좌상단 좌표가 (x0, y0)인 변환 블록에 대하여 가로, 세로 방향에 적용하는 변환 커널을 지시한다. 이때, mts_idx[x0][y0]가 존재하지 않을 때 (시그널링/파싱되지 않았을 때)에는 다음의 조건에 따라 추론할 수 있다.
- [0180] i) 픽처의 좌상단 좌표를 기준으로 좌상단 좌표가 (x0, y0)인 휘도 성분 변환 블록에 MTS를 적용함을 나타내는 1-bit flag가 0인 경우 (cu_mts_flag[x0][y0] == 0인 경우), -1로 추론할 수 있다.
- [0181] ii) 그렇지 않은 경우 (cu_mts_flag[x0][y0] == 1인 경우), -2로 추론할 수 있다.

- [0182] cu_mts_flag[x0][y0]가 1일 때에 mts_idx[x0][y0]가 존재하지 않는 경우는 해당 블록이 화면내 예측 모드로 예측되었고 블록 내의 0이 아닌 계수의 개수가 2인 경우만 존재하므로, cu_mts_flag[x0][y0]가 1인 지를 나타내는 조건이 해당 블록이 화면내 예측으로 부호화되었고 블록 내의 0이 아닌 계수의 개수가 2보다 작음을 표현할 수 있다. 도 16에서 기 설정된 mts_idx는 -1, 0, 1, 2, 3이므로 이들과 중복되지 않는 정수 값을 사용하여 추론되는 mts_idx (-2)와 이때 적용되는 커널을 mts_idx가 0일 때의 커널로 설정할 수 있다.
- [0184] mts_idx의 MSB (most significant bit)는 예측 모드에 따라 기 설정된 변환 커널 세트에서 세로 방향에 적용되는 변환 커널 후보 인덱스를 지시할 수 있으며, LSB (least significant bit)는 예측 모드에 따라 기 설정된 변환 커널 세트에서 가로 방향에 적용되는 변환 커널 후보 인덱스를 지시할 수 있다. 즉, 0인 경우 변환 커널 세트 내의 첫 번째 변환 커널을 나타낼 수 있으며, 1인 경우 변환 커널 세트 내의 두 번째 변환 커널을 나타낼 수 있다.
- [0185] mts_idx에 따른 수평, 수직 방향에 적용되는 변환 커널은 다음과 같이 해석할 수 있다.
- [0187] i) mts_idx가 -2인 경우: 화면내 예측된 블록의 수평 - 변환 세트 내 첫 번째 후보, 화면내 예측된 블록의 수직 - 변환 세트 내 첫 번째 후보, 화면간 예측된 블록에 대해서는 정의되지 않음
- [0188] ii) mts_idx가 -1인 경우: 수평 - DCT-II, 수직 - DCT-II
- [0189] iii) mts_idx가 0인 경우: 수평 - 변환 세트 내 첫 번째 후보, 수직 - 변환 세트 내 첫 번째 후보
- [0190] iv) mts_idx가 1인 경우: 수평 - 변환 세트 내 두 번째 후보, 수직 - 변환 세트 내 첫 번째 후보
- [0191] v) mts_idx가 2인 경우: 수평 - 변환 세트 내 첫 번째 후보, 수직 - 변환 세트 내 두 번째 후보
- [0192] vi) mts_idx가 3인 경우: 수평 - 변환 세트 내 두 번째 후보, 수직 - 변환 세트 내 두 번째 후보
- [0194] 상기 기술한 mts_idx에 따른 수평, 수직 방향에 사용되는 변환 커널은 mts_idx의 시그널링 방법에 의존적이지 않다. 도 19의 실시예에서는 mts_idx를 고정 길이의 2-bit로 시그널링하는 예를 기술하였으나, mts_idx를 truncated binary와 같은 가변 길이로 이진화할 때에도 mts_idx 값에 따른 수평, 수직 방향에 사용되는 변환 커널은 i) 내지 vi)에서 기술한 규칙에 따라 동일하게 사용할 수 있다.
- [0195] 또한 상기 기술한 mts_idx에 따른 수평, 수직 방향에 사용되는 변환 커널은 변환 세트를 구성하는 방법에 의존적이지 않다. 도 11의 MTS 실시예에서는 화면내 예측에 대해 변환 세트 구성을 {DST-VII, DCT-VIII}, 화면간 예측에 대해 {DCT-VIII, DST-VII}으로 구성하는 예를 도시하였으나, 변환 세트를 구성하는 변환 커널 후보가 달라지거나, 사용할 수 있는 변환 세트 수가 달라져도, i) 내지 vi)에서 기술한 규칙에 따라 동일하게 사용할 수 있다. 예를 들어, 화면내 예측에 대해 {DST-IV, DCT-IV}, 화면간 예측에 대해 {DCT-IV, DST-IV}로 변환 세트를 구성하는 경우에도 동일하게 적용할 수 있다.
- [0196] trTypeHor, trTypeVer는 각각 가로, 세로 방향에 적용하는 변환 커널을 나타내는 변수로, trTypeHor 또는 trTypeVer가 0일 때에는 기본 커널인 DCT-II가 해당 방향의 변환에 사용될 수 있으며, 1일 때에는 DST-VII이 해당 방향의 변환에 사용될 수 있고, 2일 때에는 DCT-VIII이 해당 방향의 변환에 사용될 수 있다. 도 12에서 기술하였듯이 구현의 복잡도를 고려하여 DST-VII은 DST-IV로 대체될 수도 있으며, DCT-VIII은 DCT-IV로 대체될 수도 있다.
- [0197] 현재 블록이 화면내 예측 모드로 예측되었고 변환 블록 내의 0이 아닌 계수의 개수가 2 이하인 경우, -2로 추론되고 수평, 수직 방향으로 모두 DST-VII을 적용할 수 있게 된다. DST-VII이 DST-IV로 대체되는 경우, DST-VII 대신 DST-IV를 적용할 수도 있다.
- [0198] 본 발명은 도 19의 표에서 정의된 mts_idx 값에 국한되지 않고 적용될 수 있다. 도 19의 실시예에서는 기 설정된 mts_idx가 -1, 0, 1, 2, 3인 예를 도시하였으나, 다른 값들에도 적용할 수 있다. 예를 들어 cu_mts_flag가 0일 때 추론되는 mts_idx가 3이고, 시그널링되는 mts_idx가 4, 5, 6, 7인 경우, cu_mts_flag가 1일 때 추론되는 mts_idx는 3, 4, 5, 6, 7이 아닌 정수 값으로 설정하고 (2), 이때 화면내 예측된 블록에 사용하는 커널은 mts_idx가 4일 때 화면내 예측에서 사용하는 커널로 정의할 수 있다. 화면간 예측된 블록에 대해서는

cu_mts_flag가 1일 때 mts_idx가 추론되는 경우가 없으므로 이에 대하여 사용하는 커널을 정의하지 않을 수 있다.

[0199] 또한 도 19의 실시예에서는 mts_idx가 2-bit의 고정 비트 길이로 이진화되어 시그널링하는 예를 도시하였으나, 심볼을 가변 길이로 나타내는 truncated unary, truncated binary 등에도 동일한 규칙으로 적용할 수 있다.

[0200] 도 20은 본 발명의 제 2 방법에 따른 복호화기에서 MTS와 관련된 처리 과정의 흐름도를 도시한 도면이다.

[0201] 수신한 비트스트림으로부터 현재 블록에 MTS를 적용할지를 나타내는 1-bit flag인 cu_mts_flag를 파싱할 조건이 참인 경우 cu_mts_flag를 파싱한다. cu_mts_flag가 0인 경우, 현재 블록에 MTS를 적용하지 않음을 나타내므로 가로, 세로 방향에 적용하는 변환 커널을 지시하는 syntax 요소인 mts_idx를 파싱하지 않고 -1로 추론한다. mts_idx가 -1인 경우, 기본 커널인 DCT-II를 가로, 세로 방향에 모두 적용할 수 있다. cu_mts_flag가 1인 경우 해당 휘도 성분 변환 블록에 MTS를 적용함을 지시하고, mts_idx를 파싱할 조건이 참인 경우 mts_idx를 파싱한다. 현재 블록의 예측 방법이 화면내 예측인지 화면간 예측인지를 나타내는 요소(CuPredMode)가 MODE_INTRA인 경우 (해당 블록이 화면내 예측 모드로 예측되었을 경우), 해당 휘도 성분 변환 블록 내의 0이 아닌 계수의 개수가 2 이하인 경우 (numSigCoeff<3), mts_idx는 -2로 추론된다. 반면 해당 블록이 화면내 예측 모드로 예측되었고, 해당 휘도 성분 블록 내의 0이 아닌 계수의 개수가 3 이상인 경우 mts_idx를 비트스트림으로부터 파싱한다. 현재 블록이 화면간 예측 모드를 사용하여 예측되었을 경우에는 추가적인 조건 없이 mts_idx를 비트스트림으로부터 파싱한다.

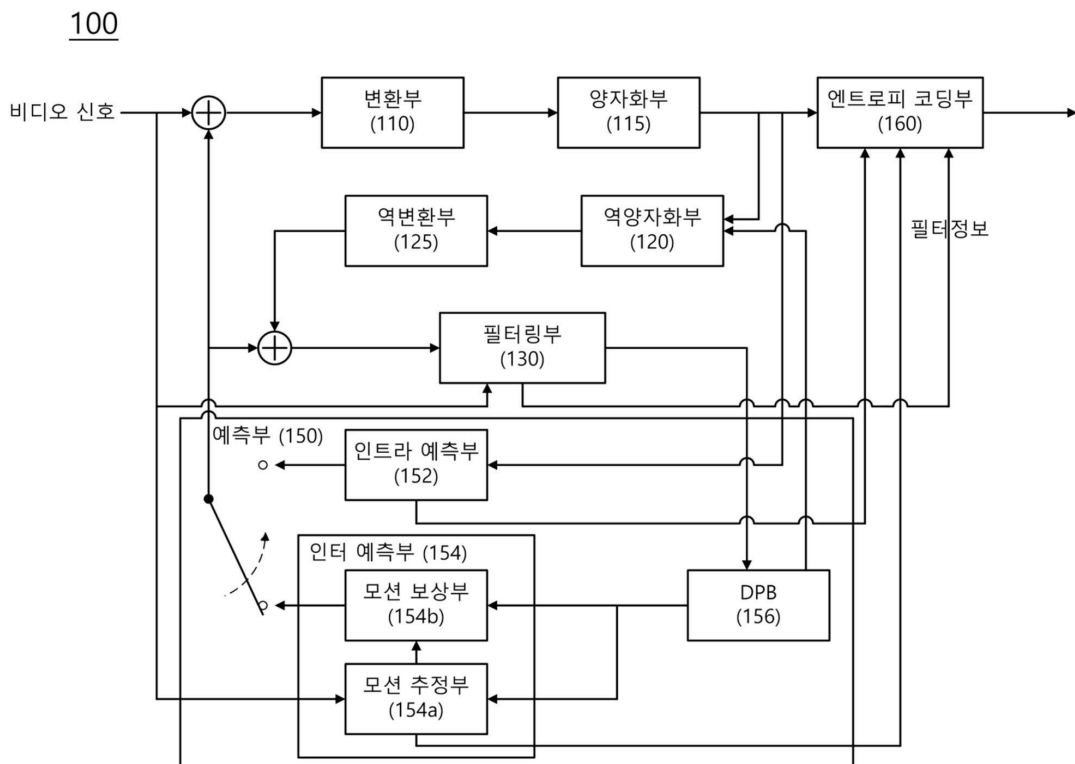
[0203] 이상에서는 본 발명을 구체적인 실시예를 통하여 설명하였으나, 당업자라면 본 발명의 취지 및 범위를 벗어나지 않고 수정, 변경을 할 수 있다. 따라서 본 발명의 상세한 설명 및 실시예로부터 본 발명이 속하는 기술분야에 속한 사람이 용이하게 유추할 수 있는 것은 본 발명의 권리범위에 속하는 것으로 해석된다.

부호의 설명

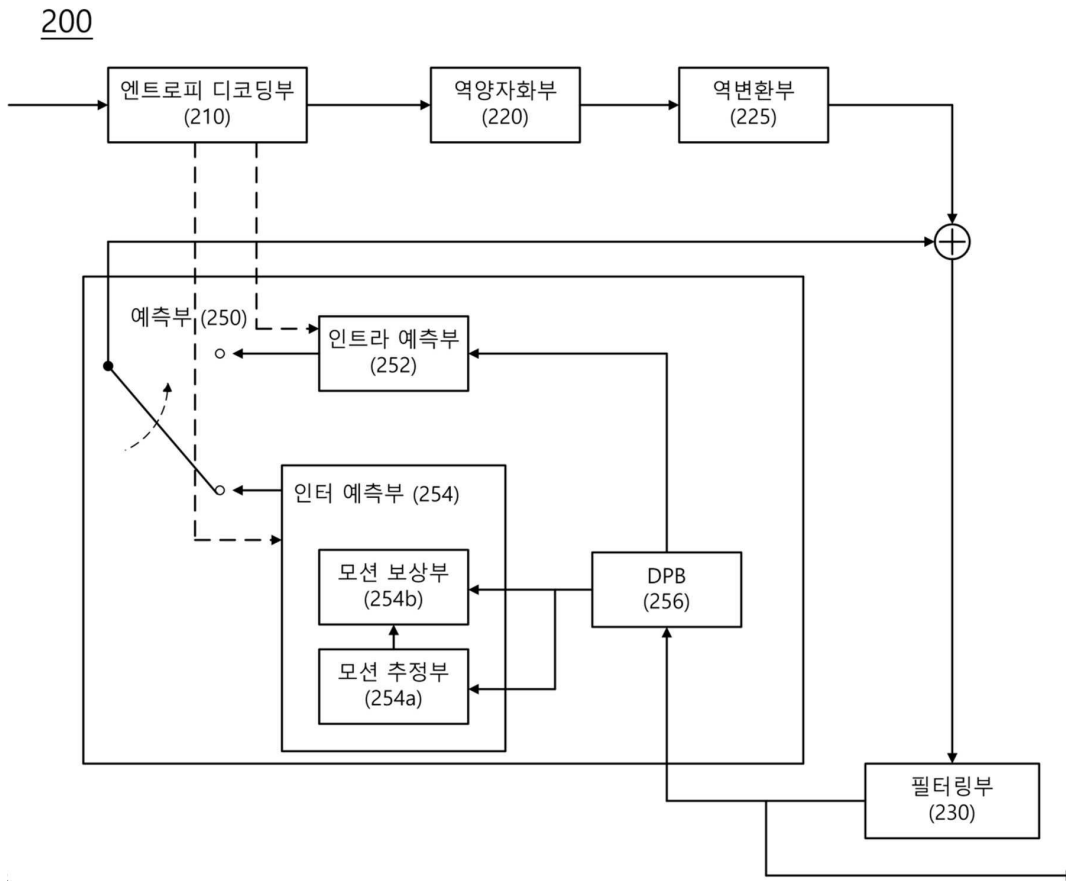
[0204] 100 : 인코딩 장치 200 : 디코딩 장치

도면

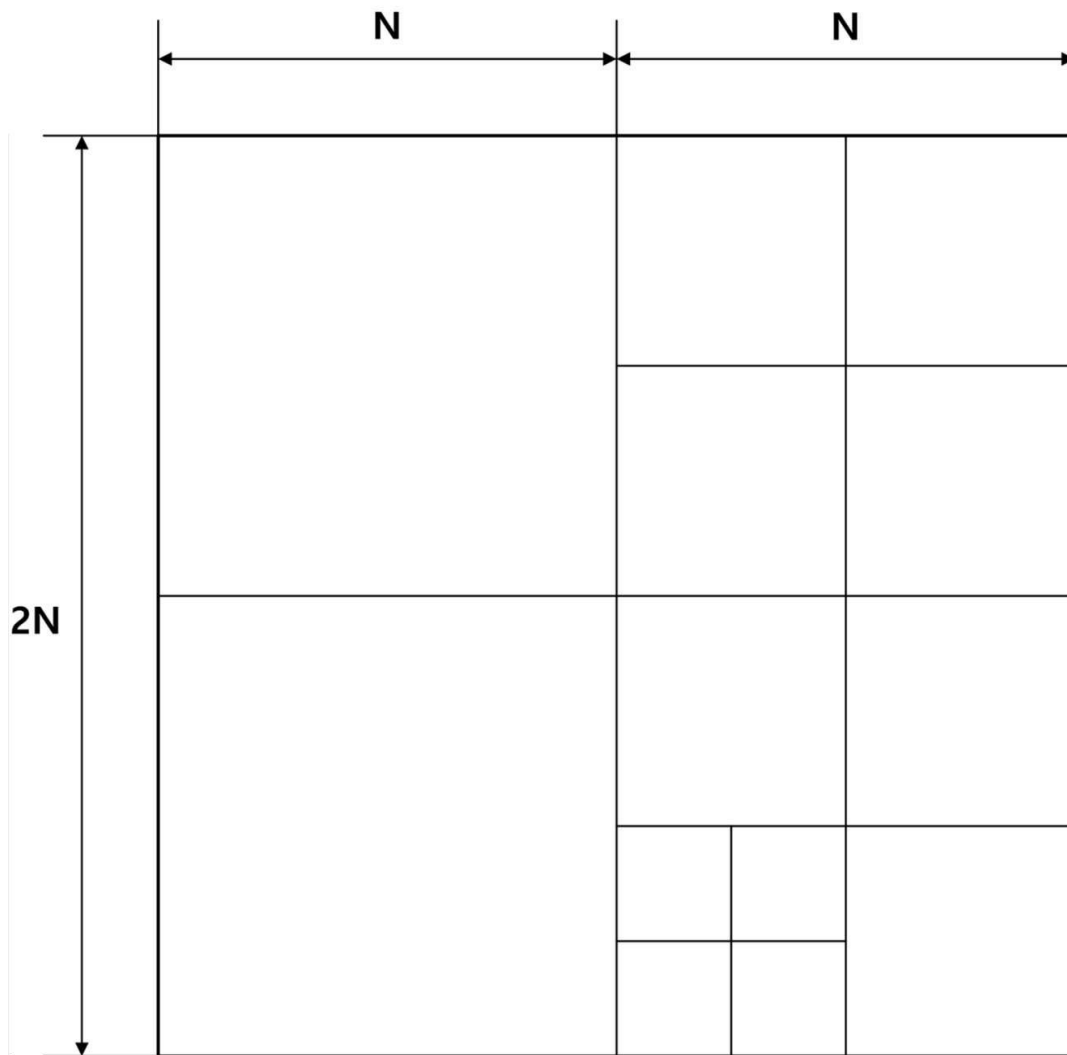
도면1



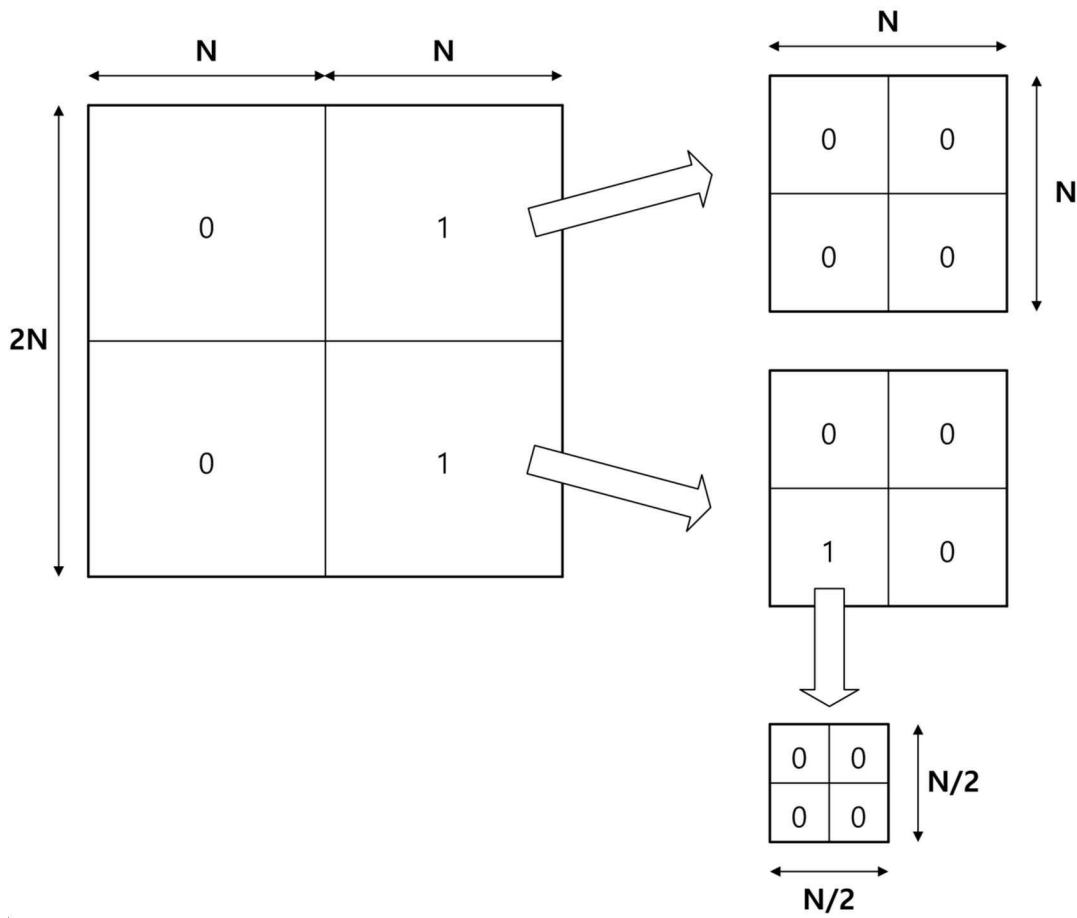
도면2



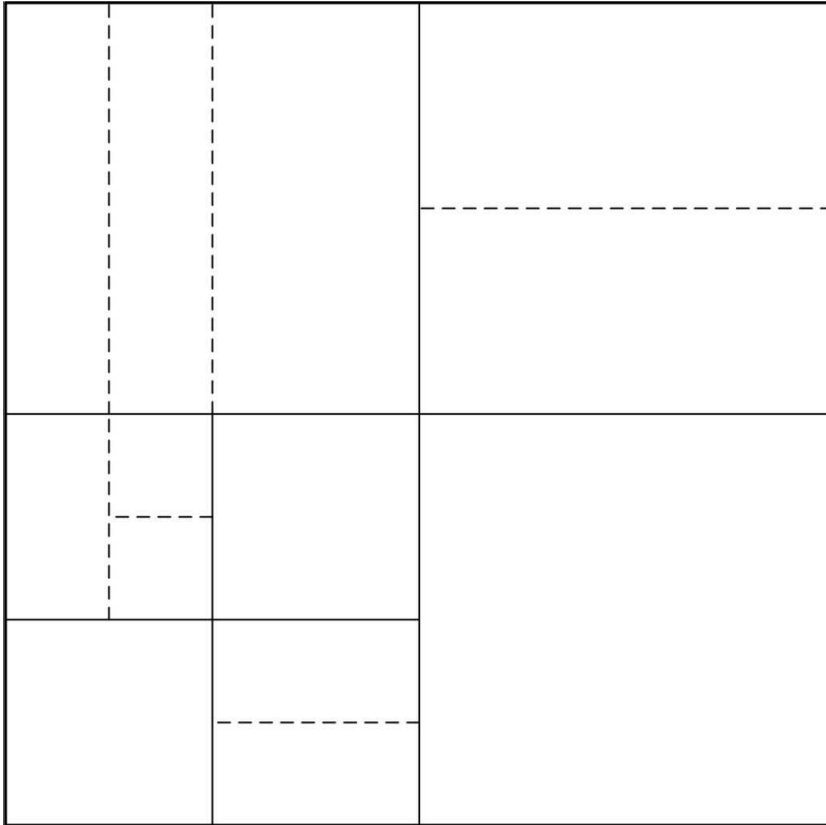
도면3



도면4



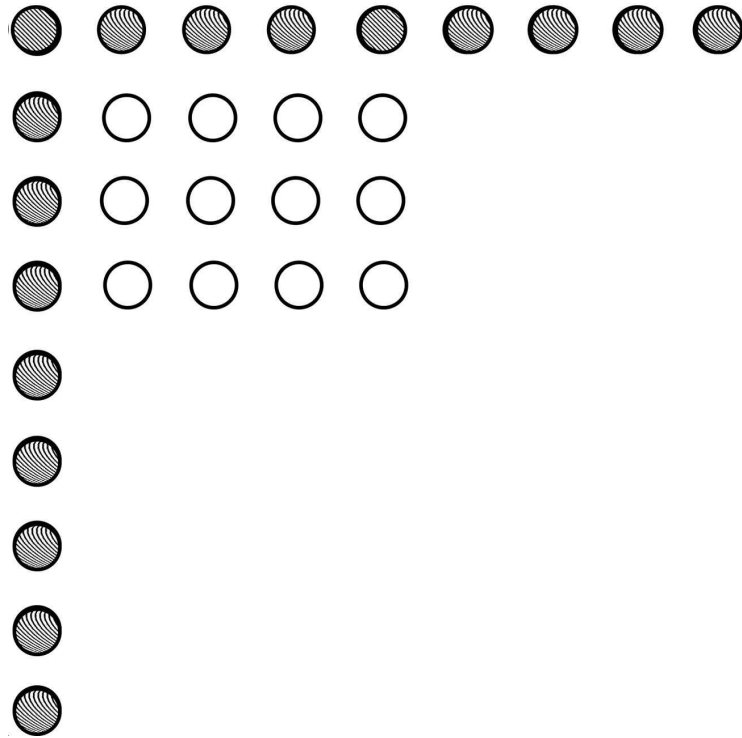
도면5



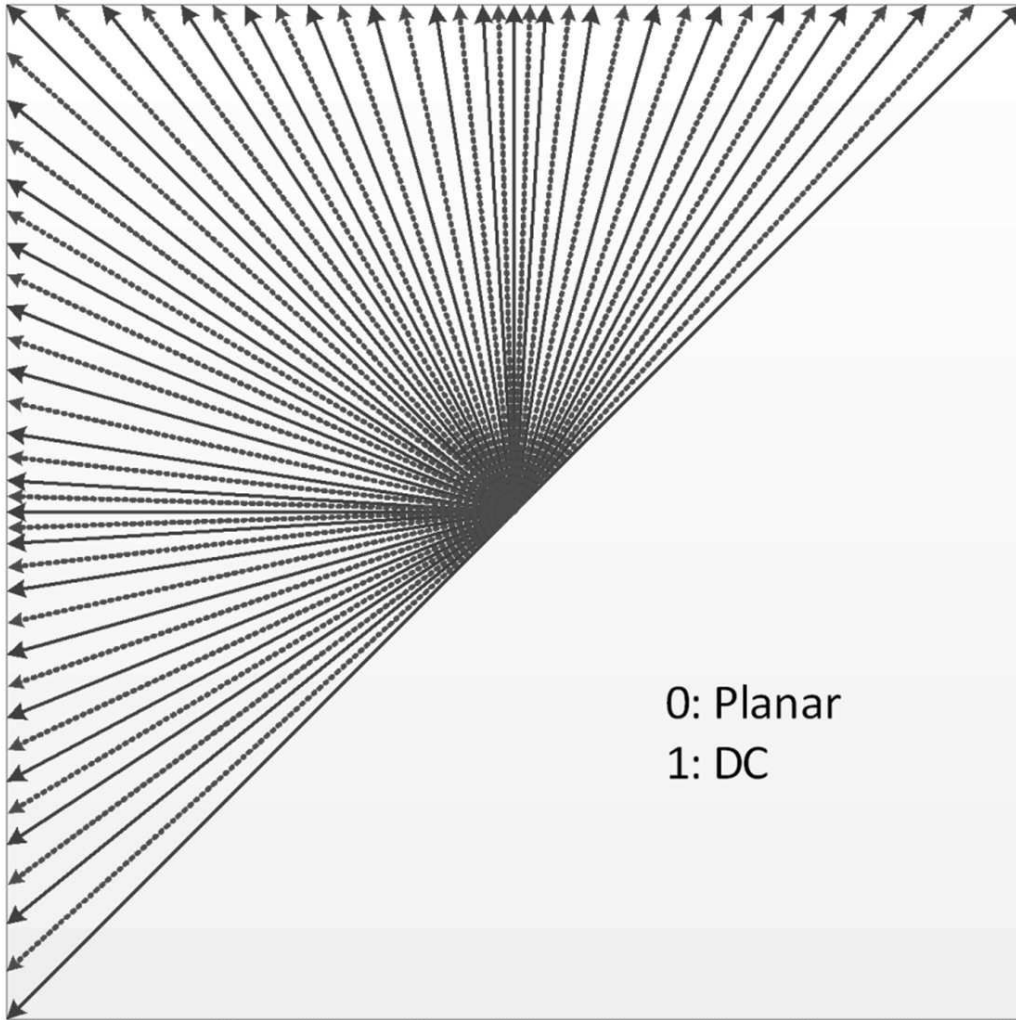
도면6

● 참조 픽셀

○ 현재 유닛의 픽셀



도면7



도면8

Transform Type	Basis function $T_i(j)$, $i, j=0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ <p style="text-align: center;">where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$</p>
DCT-V	$T_i(j) = \omega_0 \cdot \omega_1 \cdot \sqrt{\frac{2}{2N-1}} \cdot \cos\left(\frac{2\pi \cdot i \cdot j}{2N-1}\right),$ <p style="text-align: center;">where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}, \omega_1 = \begin{cases} \sqrt{\frac{2}{N}} & j = 0 \\ 1 & j \neq 0 \end{cases}$</p>
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				

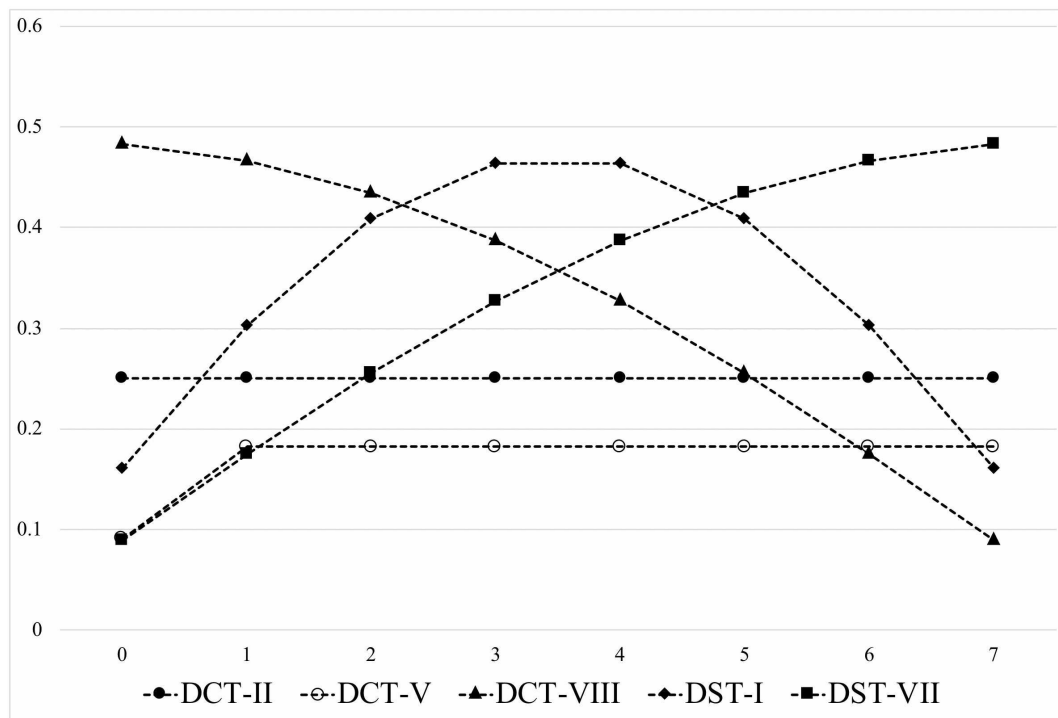
(b)

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	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ <p style="text-align: center;">where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$</p>
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

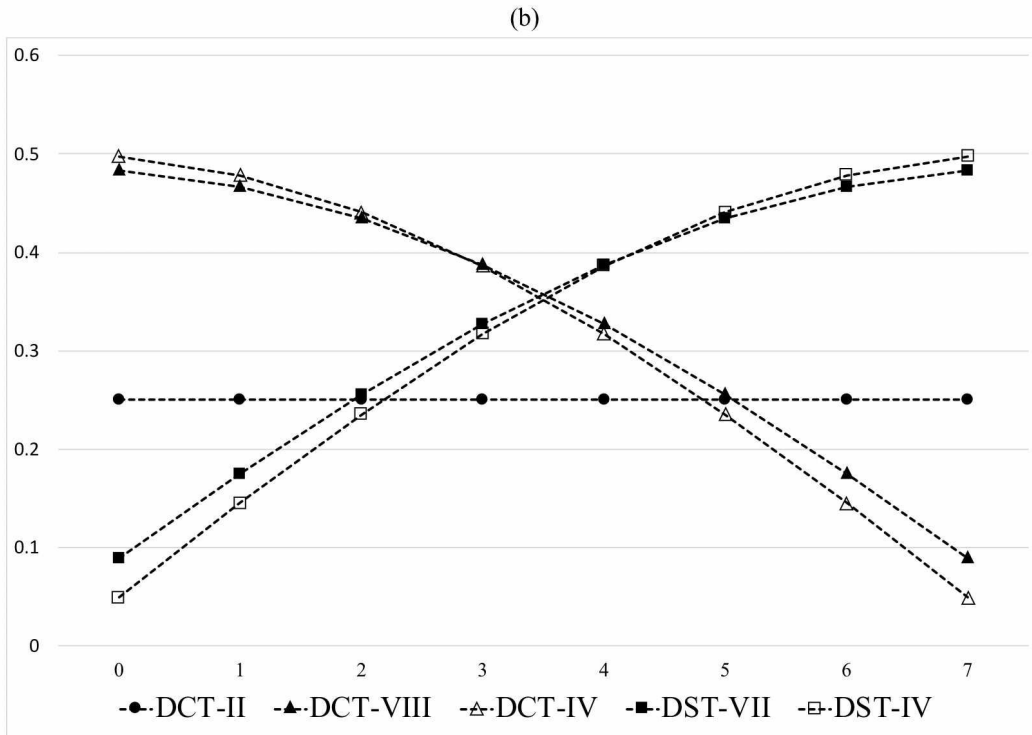
(d)

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

도면12

(a)

Transform Type	Basis function $T_i(j), i, j=0, 1, \dots, N-1$
DCT-IV	$T_i(j) = \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N}\right)$
DST-IV	$T_i(j) = \sqrt{\frac{2}{N}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N}\right)$



도면13

(a)

	Descriptor
transform_unit(x0, y0, tbWidth, tbHeight, treeType) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA)	
tu_cbf_luma [x0][y0]	ae(v)
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) {	
tu_cbf_cb [x0][y0]	ae(v)
tu_cbf_cr [x0][y0]	ae(v)
}	
if(((CuPredMode[x0][y0] == MODE_INTRA) && sps_mts_intra_enabled_flag) ((CuPredMode[x0][y0] == MODE_INTER) && sps_mts_inter_enabled_flag)) && tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32))	
cu_mts_flag [x0][y0]	ae(v)
if(tu_cbf_luma[x0][y0])	
residual_coding(x0, y0, log2(tbWidth), log2(tbHeight), 0)	
if(tu_cbf_cb[x0][y0])	
residual_coding(x0, y0, log2(tbWidth / 2), log2(tbHeight / 2), 1)	
if(tu_cbf_cr[x0][y0])	
residual_coding(x0, y0, log2(tbWidth / 2), log2(tbHeight / 2), 2)	
}	

(b)

	Descriptor
residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	
if(transform_skip_enabled_flag && (cIdx != 0 cu_mts_flag[x0][y0] == 0) && (log2TbWidth <= 2) && (log2TbHeight <= 2))	
transform_skip_flag [x0][y0][cIdx]	ae(v)
: (Residual Coding Syntax)	
if(cu_mts_flag[x0][y0] && (cIdx == 0) && !transform_skip_flag[x0][y0][cIdx] && ((CuPredMode[x0][y0] == MODE_INTRA && numSigCoeff > 2) (CuPredMode[x0][y0] == MODE_INTER)))	
mts_idx [x0][y0]	ae(v)
}	

도면14

	Descriptor
residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	
if(transform_skip_enabled_flag && (cIdx != 0 cu_mts_flag[x0][y0] == 0) && (log2TbWidth <= 2) && (log2TbHeight <= 2))	
transform_skip_flag [x0][y0][cIdx]	ae(v)
: (Residual Coding Syntax)	
if(cu_mts_flag[x0][y0] && (cIdx == 0) && ((CuPredMode[x0][y0] == MODE_INTRA && numSigCoeff > 2) (CuPredMode[x0][y0] == MODE_INTER)))	
mts_idx [x0][y0]	ae(v)
}	

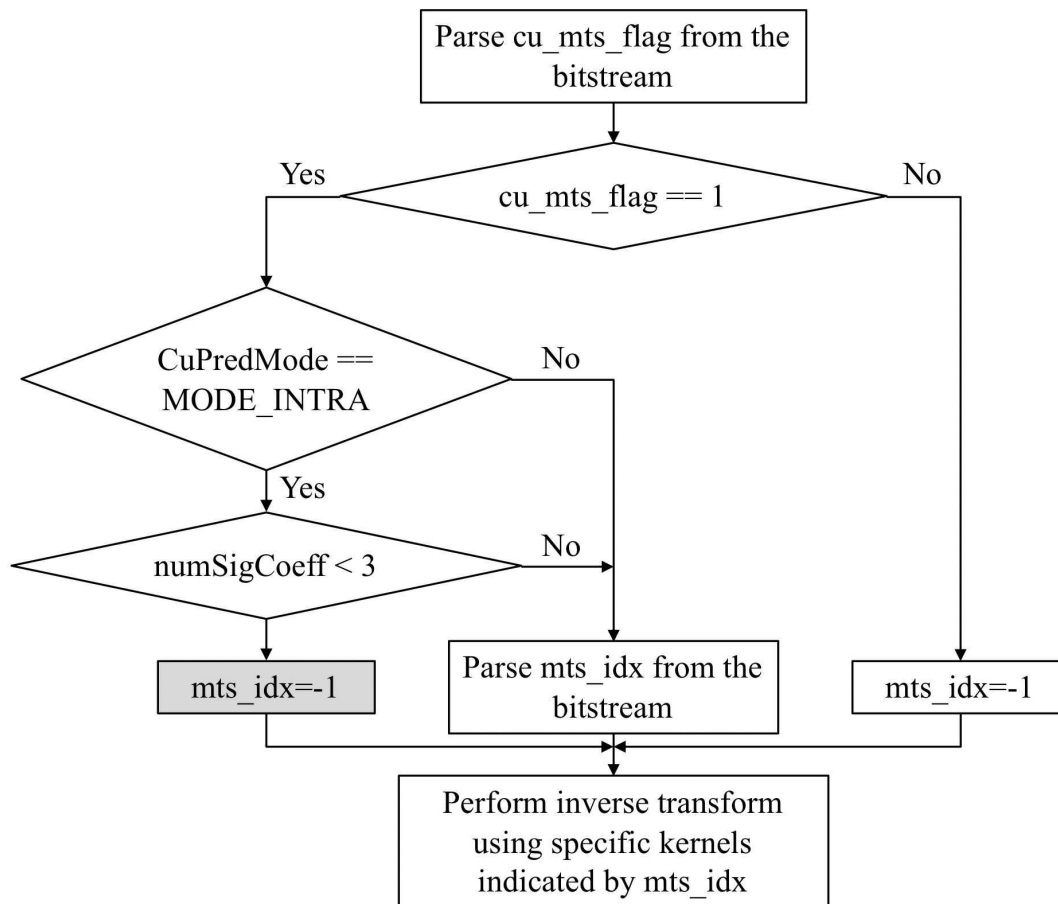
도면15

mts_idx[x0][y0] specifies which transform kernels are applied to the luma residual samples along the horizontal and vertical direction of the current transform block. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture.

When **mts_idx**[x0][y0] is not present, it is inferred to be equal to -1.

mts_idx[x][y]	CuPredMode[x][y] == MODE_INTRA		CuPredMode[x][y] == MODE_INTER	
	trTypeHor	trTypeVer	trTypeHor	trTypeVer
-1 (inferred)	0	0	0	0
0 (00)	1	1	2	2
1 (01)	2	1	1	2
2 (10)	1	2	2	1
3 (11)	2	2	1	1

도면16



도면17

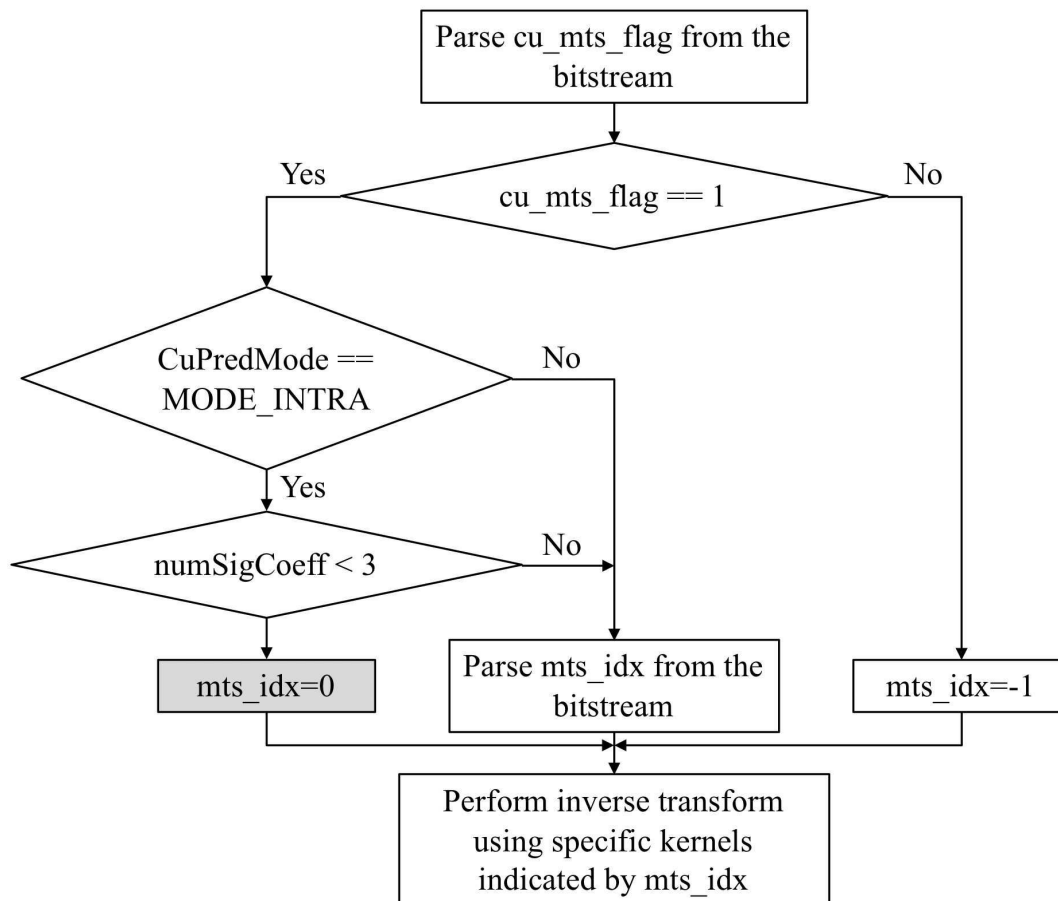
mts_idx[x0][y0] specifies which transform kernels are applied to the luma residual samples along the horizontal and vertical direction of the current transform block. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture.

When **mts_idx**[x0][y0] is not present, it is inferred as follows:

- If **cu_mts_flag**[x0][y0] is equal to 0, **mts_idx**[x0][y0] is inferred to be equal to -1.
- Otherwise (**cu_mts_flag**[x0][y0] is equal to 1), **mts_idx**[x0][y0] is inferred to be equal to 0.

mts_idx[x][y]	CuPredMode[x][y] == MODE_INTRA		CuPredMode[x][y] == MODE_INTER	
	trTypeHor	trTypeVer	trTypeHor	trTypeVer
-1 (inferred)	0	0	0	0
0 (00 / inferred)	1	1	2	2
1 (01)	2	1	1	2
2 (10)	1	2	2	1
3 (11)	2	2	1	1

도면18



도면19

mts_idx[x0][y0] specifies which transform kernels are applied to the luma residual samples along the horizontal and vertical direction of the current transform block. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture.

When **mts_idx**[x0][y0] is not present, it is inferred as follows:

- If **cu_mts_flag**[x0][y0] is equal to 0, **mts_idx**[x0][y0] is inferred to be equal to -1.
- Otherwise (**cu_mts_flag**[x0][y0] is equal to 1), **mts_idx**[x0][y0] is inferred to be equal to -2.

mts_idx[x][y]	CuPredMode[x][y] == MODE_INTRA		CuPredMode[x][y] == MODE_INTER	
	trTypeHor	trTypeVer	trTypeHor	trTypeVer
-2 (inferred)	1	1	N/A	N/A
-1 (inferred)	0	0	0	0
0 (00)	1	1	2	2
1 (01)	2	1	1	2
2 (10)	1	2	2	1
3 (11)	2	2	1	1

도면20

