



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년04월26일
(11) 등록번호 10-2391402
(24) 등록일자 2022년04월22일

- (51) 국제특허분류(Int. Cl.)
H04N 19/12 (2014.01) H04N 19/147 (2014.01)
H04N 19/176 (2014.01) H04N 19/88 (2014.01)
- (52) CPC특허분류
H04N 19/12 (2015.01)
H04N 19/147 (2015.01)
- (21) 출원번호 10-2021-7021840(분할)
- (22) 출원일자(국제) 2014년11월27일
심사청구일자 2021년07월12일
- (85) 번역문제출일자 2021년07월12일
- (65) 공개번호 10-2021-0090738
- (43) 공개일자 2021년07월20일
- (62) 원출원 특허 10-2016-7017077
원출원일자(국제) 2014년11월27일
심사청구일자 2019년06월10일
- (86) 국제출원번호 PCT/FR2014/053069
- (87) 국제공개번호 WO 2015/079179
국제공개일자 2015년06월04일
- (30) 우선권주장
1361822 2013년11월29일 프랑스(FR)
- (56) 선행기술조사문헌
JP2012105128 A
KR1020110014839 A

- (73) 특허권자
오렌지
프랑스 이시레몰리노 웨 뒤 프헤지덩 후저벨 111
(우: 92130)
- (72) 발명자
필립, 피에릭
프랑스 에프-35520 멜리세 튀 데스 오르메스 9
포르베르크, 헨드릭
독일 48153 윈스터 클레포른슈트라쎄 14
- (74) 대리인
특허법인 남앤남

전체 청구항 수 : 총 8 항

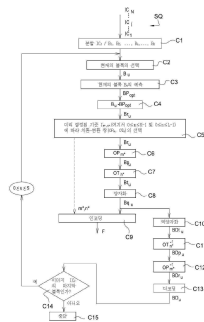
심사관 : 김영태

(54) 발명의 명칭 이미지를 인코딩하고 디코딩하기 위한 방법, 이미지를 인코딩하고 디코딩하기 위한 디바이스 및 상응하는 컴퓨터 프로그램

(57) 요약

본 발명은 블록들로 컷팅되는 적어도 하나의 이미지(IC_i)를 인코딩하는 방법에 관한 것이다. 본 발명에 따르면, 인코딩 방법은 인코딩될 현재의 블록(B_i)에 대해: 적어도 하나의 예측 변수 블록(BP_{opt})에 의해 현재의 블록을 예측하는 단계(C3); 예측 변수 블록과 현재의 블록 사이의 차이를 나타내는 잔여 데이터 블록을 결정하는 단계(뒷면에 계속)

대표도 - 도1



(C4); 각각 복수의 치환 연산($OP_0, OP_1, \dots, OP_m, \dots, OP_{K-1}$)(여기서 $0 \leq m \leq K-1$ 및 $0 \leq m^* \leq K-1$), 및 복수의 변환 연산($OT_0, OT_1, \dots, OT_n, \dots, OT_{L-1}$)(여기서 $0 \leq n \leq L-1$ 및 $0 \leq n^* \leq L-1$) 중에서의 치환 연산(OP_{m^*}) 및 변환 연산(OT_{n^*})을 포함하는 수학적 연산들의 쌍을 미리 결정된 기준에 따라 선택하는 단계(C5); 이러한 선택된 쌍의 치환 연산(OP_{m^*})을 결정된 잔여 블록의 데이터에 적용하는 단계(C6); 이러한 선택된 쌍의 변환 연산(OT_{n^*})을 치환된 데이터에 적용하는 단계(C7); 및 이러한 적용된 변환 연산으로부터 얻어지는 이러한 데이터를 인코딩하는 단계(C9)를 구현한다.

(52) CPC특허분류

H04N 19/176 (2015.01)

H04N 19/88 (2015.01)

명세서

청구범위

청구항 1

블록들로 컷팅되는 적어도 하나의 이미지(IC_j)를 나타내는 데이터 신호(F)를 디코딩하기 위한 방법으로서,

디코딩될 현재 블록(B₀)에 대해:

- 상기 데이터 신호에서, 상기 디코딩될 현재 블록과 연관된 잔여(residual) 데이터 블록을 결정하는 단계(D1),
- 변환 연산(OT_{n*}^{-1}) 및 치환 연산(OP_{m*}^{-1})을 포함하는 수학적 연산들의 쌍을 결정하는 단계(D6),
- 변환된 데이터를 생성하기 위해, 상기 결정된 쌍의 상기 변환 연산(OT_{n*}^{-1})을 상기 잔여 데이터 블록의 데이터에 적용하는 단계(D7),
- 치환된 데이터를 생성하기 위해, 상기 결정된 쌍의 상기 치환 연산(OP_{m*}^{-1})을 상기 변환된 데이터에 적용하는 단계(D8) - 상기 치환 연산은 상기 변환된 데이터를 이항(transpose)하는 연산임 -,
- 예측 디코딩에 의해 상기 현재 블록(B₀)을 재건(reconstruct)(D9)하기 위해 상기 치환된 데이터를 사용하는 단계를 포함하는 것을 특징으로 하는, 디코딩하기 위한 방법.

청구항 2

제1항에 있어서,

상기 수학적 연산들의 쌍을 결정하는 단계(D6)는:

- 상기 데이터 신호에서, 상기 쌍을 나타내는 정보를 판독하는 단계(D610),
- 상기 판독되는 정보와 연관된 쌍의 상기 치환 및 변환 연산들의 역 치환 및 변환 연산들을 결정하는 단계(D611)

를 구현하는, 디코딩하기 위한 방법.

청구항 3

제1항에 있어서,

상기 수학적 연산들의 쌍을 결정하는 단계(D6)는:

- 치환 연산이 뒤따르는 변환 연산을 포함하는 수학적 연산들의 복수의 쌍들을 상기 잔여 데이터 블록의 데이터에 적용하는 단계(D620) - 상기 변환 및 치환 연산들은 미리결정된 복수의 변환 연산들($OT_0^{-1}, OT_1^{-1}, \dots, OT_{n-1}^{-1}, \dots, OT_{L-1}^{-1}$)(여기서 $0 \leq n \leq L-1$) 및 미리결정된 복수의 치환 연산들($OP_0^{-1}, OP_1^{-1}, \dots, OP_m^{-1}, \dots, OP_{K-1}^{-1}$)(여기서 $0 \leq m \leq K-1$) 중으로부터 선택됨 -,
- 상기 적용되는 복수의 쌍들로부터 복수의 블록들을 재건하는 단계(D621),
- 각각의 재건된 블록의 픽셀들과 이웃 픽셀들 간의 연속성을 측정하는 것에 기초하여 상기 복수의 재건된 블록들에 적용되는 일관성(coherence)의 기준을 최대화함으로써, 상기 복수의 재건된 블록들 중으로부터의 재건된 블록을 선택하는 단계(D622)

로 구성되는, 디코딩하기 위한 방법.

청구항 4

제1항에 있어서,

상기 결정되는 변환 연산은 이산 사인 변환인, 디코딩하기 위한 방법.

청구항 5

제1항에 있어서,

상기 이항하는 연산은 상기 결정된 쌍의 상기 변환 연산을 적용하는 것의 결과로서 획득되는 데이터를 점진적으로 이동시키는 것으로 구성되는, 디코딩하기 위한 방법.

청구항 6

제1항에 있어서,

상기 치환 연산은 오직 1들 및 0들 만을 포함하는 행렬이고, 그리고 상기 변환 연산은 크기 M의 벡터들에 대해 연산하는 크기 MxM의 1차원 변환이고, M은 정수 변수인, 디코딩하기 위한 방법.

청구항 7

블록들로 컷팅되는 적어도 하나의 이미지(IC_i)를 나타내는 데이터 신호(F)를 디코딩하기 위한 디바이스(DO)로서,

디코딩될 현재 블록(B_i)에 대해:

- 상기 데이터 신호에서, 상기 디코딩될 현재 블록과 연관된 잔여 데이터 블록을 결정하기 위한 수단(MI_DO),
- 변환 연산(OP⁻¹_{ns}) 및 치환 연산(OP⁻¹_{ms})을 포함하는 수학적 연산들의 쌍을 결정하기 위한 수단(MDT_DO),
- 변환된 데이터를 생성하기 위해, 상기 결정된 쌍의 상기 변환 연산(OP⁻¹_{ns})을 상기 잔여 데이터 블록의 데이터에 적용하기 위한 변환 수단(MT⁻¹_DO),
- 치환된 데이터를 생성하기 위해, 상기 결정된 쌍의 상기 치환 연산(OP⁻¹_{ms})을 상기 변환된 데이터에 적용하기 위한 치환 수단(MPER⁻¹_DO) - 상기 치환 연산은 상기 변환된 데이터를 이항하는 연산임 -,
- 예측 디코딩에 의해 상기 현재 블록(B_i)을 재건하기 위해 상기 치환된 데이터를 사용하는 재건 수단(PRED⁻¹_DO)을 포함하는, 디코딩하기 위한 디바이스(DO).

청구항 8

컴퓨터 프로그램이 저장된 비-일시적 컴퓨터-판독가능 저장 매체로서,

상기 컴퓨터 프로그램은, 상기 컴퓨터 프로그램이 컴퓨터 상에서 실행될 때, 제1항 내지 제6항 중 어느 한 항에 따른 디코딩하는 방법을 구현하기 위한 명령어들을 포함하는, 비-일시적 컴퓨터-판독가능 저장 매체.

발명의 설명

기술 분야

- [0001] 본 발명은 일반적으로 이미지 처리의 분야에 관한 것으로, 보다 정확하게는 디지털 이미지 및 디지털 이미지 시퀀스의 인코딩 및 디코딩에 관한 것이다.
- [0002] 디지털 이미지의 인코딩/디코딩은 특히:
 - [0003] - 동일한 카메라로부터의 그리고 시간 맞춰 서로 계속되는 이미지들(2D 인코딩/디코딩),
 - [0004] - 상이한 시점을 따라 배향되는 상이한 카메라로부터의 이미지들(3D 인코딩/디코딩),

- [0005] - 텍스처 및 깊이의 상응하는 성분들(3D 인코딩/디코딩),
- [0006] 등을 포함하는 적어도 하나의 영상 시퀀스로부터의 이미지에 적용된다.
- [0007] 본 발명은 2D 또는 3D 이미지의 인코딩/디코딩과 유사한 방식으로 적용된다.
- [0008] 본 발명은 배타적으로는 아니지만, 특히 현재의 AVC 및 HEVC 영상 인코더 및 현재의 AVC 및 HEVC 영상 인코더의 확장(MVC, 3D-AVC, MV-HEVC, 3D-HEVC 등)으로 구현되는 영상 인코딩, 그리고 상응하는 디코딩에 적용될 수 있다.

배경 기술

- [0009] 현재의 영상 인코더(MPEG, H.264, HEVC 등)는 블록에 의한 영상 시퀀스의 표현을 사용한다. 이미지는 블록으로 컷팅되며, 블록은 재귀적으로 다시 작은 조각으로 컷팅될 수 있다. 그 다음 각각의 블록은 인트라 이미지 (intra-image) 또는 인터 이미지(inter-image) 예측에 의해 인코딩된다. 따라서, 일부 이미지는 공간적 예측(인트라 예측)에 의해 인코딩되며, 다른 이미지는 또한 당업자에게 알려진 모션 보정을 사용하여 하나 이상의 인코딩된-디코딩된 참조 이미지에 대하여 일시적 예측(인터 예측)에 의해 인코딩된다.
- [0010] 잔여 블록은 본래 블록 마이너스 예측에 상응하는 또한 예측 잔여물로 지칭되는 각각의 블록마다 인코딩된다. 잔여 블록은 변환의 수학적 연산을 사용하여 변환되며, 그 다음 양자화 예를 들어, 스칼라 타입의 수학적 연산을 사용하여 양자화된다. 계수는 양자화 단계의 끝부분에서 얻어진다. 그것은 그 다음 선택되었던 인코딩의 모드에 따른 관독 순서로 스케닝된다. HEVC 표준에서, 예를 들어, 관독 순서는 행해지는 예측에 의존하고 "수평", "수직" 또는 "대각선" 순서로 수행될 수 있다.
- [0011] 앞서 언급한 스캔의 끝부분에서, 계수의 1차원 리스트가 얻어진다. 이러한 리스트의 계수는 그 다음, 손실 없이 계수들을 인코딩하는 것이 목적인 엔트로피 인코딩에 의해 비트의 형태로 인코딩된다.
- [0012] 앞서 언급한 변환의 수학적 연산에 대하여, 블록 변환은 이하의 3개의 중요한 기준을 충족시키므로, 가장 많이 사용된다:
 - [0013] - 이들이 블록 변환이므로, 서로와 관계 없이 블록을 조작하는 것이 용이하며,
 - [0014] - 블록 변환은 주파수 영역에서 정보를 압축하는데 효과적이며, 속도 감소의 연산이 작동되며,
 - [0015] - 블록 변환은 제한된 알고리즘 복잡성을 보장하는 빠른 알고리즘을 사용하여 구현된다.
- [0016] 영상 인코딩의 분야에서, 이산 코사인 변환인, (영어 약어로) DCT, 또는 이산 사인 변환인, (영어 약어로) DST 가 그것들의 실행 속도에 바람직하다. 특히, 그것들은 소수의 연산으로 분리 가능한 방식으로 구현되고 고정 소수점 프로세서에서의 처리에 양호하게 적합한 구현을 갖는다.
- [0017] 엔트로피 인코딩 후에 얻어지는 비트는 디코더로 송신되는 것으로 의도되는 신호 또는 데이터 스트림에 포함된다.
- [0018] 그 자체로 알려진 방식으로, 그러한 신호는:
 - [0019] - 앞서 언급한 리스트에 포함되는 양자화된 계수,
 - [0020] - 사용되는 인코딩의 모드를 나타내는 정보로서, 특히:
 - [0021] 예측의 모드(인트라 예측, 인터 예측, 어떤 정보도 디코더로 송신되지 않는 예측을 행하는 디폴트 예측("스킵"));
 - [0022] 예측의 타입을 지정하는 정보(배향, 참조 이미지 등);
 - [0023] 블록 컷팅의 타입;
 - [0024] 변환의 타입 예를 들어, DCT 4x4, DCT 8x8 등
 - [0025] 필요하다면, 모션 정보;
 - [0026] 등인 정보를 포함한다.
- [0027] 스트림이 디코더에 의해 수신되었으면, 디코딩은 한 이미지씩, 그리고 각각의 이미지에 대해, 한 블록씩 행해진다. 각각의 블록의 경우, 스트림의 상응하는 요소들이 관독된다. 블록의 계수의 역양자화 및 역변환이 디코딩된

예측 잔여물을 생성하도록 수행된다. 그 다음, 블록의 예측이 계산되고 블록이 디코딩된 예측 잔여물에 예측을 추가함으로써 재건된다.

[0028] 이제 막 설명하였던 통상적 인코딩/디코딩 기법은 확실히 인코딩 성능의 개선을 가능하게 한다. 영상 맥락에 따르면, 이는 특히:

[0029] - 이미지를 송신하는데 사용되는 주어진 네트워크 속도(rate)에 대한 이미지의 품질의 개선,

[0030] - 이전에 설정된 이미지 품질 기준에 대한 이미지 송신 속도의 감소를 가능하게 한다.

[0031] 그러나, 그러한 인코딩 성능은 현재 최적화되어 있지 않고 특히 당업자에게 널리 알려진 기준들인 속도/왜곡(rate/distortion) 비용을 최소화하는 것 또는 최고 효과성/복잡성 중간물의 선택의 관점에서 볼 때 추가로 개선되어야 한다.

발명의 내용

해결하려는 과제

[0032] 본 발명의 목적들 중 하나는 앞서 언급한 종래 기술의 문제점들을 극복하는 것이다.

[0033] 이러한 목적으로, 본 발명의 하나의 목적은 블록들로 컷팅되는 적어도 하나의 이미지를 인코딩하는 방법에 관한 것이다.

과제의 해결 수단

[0034] 본 발명에 따른 인코딩하는 방법은 인코딩될 현재의 블록에 대해:

[0035] - 적어도 하나의 예측 변수 블록을 사용하여 현재의 블록을 예측하는 단계,

[0036] - 예측 변수 블록과 현재의 블록 사이의 차이를 나타내는 잔여 데이터 블록을 결정하는 단계,

[0037] - 복수의 치환 연산 및 복수의 변환 연산 중으로부터의 치환 연산 및 변환 연산을 포함하는 수학적 연산들의 쌍을 미리 결정된 기준에 따라 선택하는 단계,

[0038] - 선택된 쌍의 치환 연산을 결정된 잔여 블록의 데이터에 적용하는 단계,

[0039] - 선택된 쌍의 변환 연산을 치환된 데이터에 적용하는 단계,

[0040] - 적용된 변환 연산으로부터의 데이터를 인코딩하는 단계를 구현하는 것을 특징으로 한다.

[0041] 그러한 구성은 효과성/복잡성 중간물의 더 양호한 선택을 통해 디코더로 송신될 데이터 신호의 압축 성능을 상당히 개선하는 이점을 갖는다. 즉, 인코딩 성능은 인코딩 단계들의 구현의 단순함 및 디코딩의 제한된 복잡성 둘 다를 통해 종래 기술에서 얻어지는 성능보다 더 양호하다.

[0042] 하나의 특정 실시예에 따르면, 복수의 변환 연산은 이산 사인 변환을 포함한다.

[0043] DST 변환의 선택은 유리하게는 인코딩 성능을 최적화할 뿐만 아니라, 계산 리소스들을 감소시키는 것도 가능하게 한다.

[0044] 제1 변형예에 따르면, 이용 가능한 복수의 변환 연산 중으로부터 DST 변환의 선택은 인코딩에서 부과되며, 이는 변환을 선택하는 단계를 단순화한다.

[0045] 제2 변형예에 따르면, 복수의 변환 연산은 DST 변환인 단일 변환 연산을 포함한다. 이러한 제2 변형예는 물론, DST 변환과 연관되는 인덱스가 디코더에 이미 이용 가능한 경우에, 인코더가 DST 변환과 연관되는 인덱스를 디코더로 반드시 송신하도록 강제되지 않는 이점을 갖는다. 따라서, 인코딩 정보를 시그널링(signaling)하는 비용은 이로써 효과적으로 감소된다.

[0046] 다른 특정 실시예에 따르면, 미리 결정된 기준은 인코딩될 현재의 블록의 속도-왜곡 비용의 최소화이다.

[0047] 그러한 기준의 선택은 치환-변환 쌍의 선택을 최적화한다.

[0048] 다른 특정 실시예에 따르면, 선택되는 치환 연산은 이러한 잔여 블록의 데이터를 이항하는 연산에 따라 결정된 잔여 블록의 데이터를 점진적으로 이동시키는 것으로 구성된다.

- [0049] 이러한 타입의 치환은 계산하는 관점에 볼 때 특히 단순하다는 이점을 갖는다.
- [0050] 다양한 앞서 언급한 실시예 또는 특징은 앞서 정의된 바와 같은 인코딩하는 방법의 단계들에 독립적으로 또는 서로와의 조합으로 추가될 수 있다.
- [0051] 본 발명은 또한 블록들로 컷팅되는 적어도 하나의 이미지를 인코딩하는 디바이스에 관한 것이다.
- [0052] 그러한 인코딩 디바이스는 인코딩될 현재의 블록에 대해:
- [0053] - 적어도 하나의 예측 변수 블록을 사용하여 현재의 블록을 예측하는 모듈,
- [0054] - 예측 변수 블록과 현재의 블록 사이의 차이를 나타내는 잔여 데이터 블록을 결정하는 모듈,
- [0055] - 복수의 치환 연산 및 복수의 변환 연산 중으로부터의 치환 연산 및 변환 연산을 포함하는 수학적 연산들의 쌍을 미리 결정된 기준에 따라 선택하는 모듈,
- [0056] - 선택된 쌍의 치환 연산을 결정된 잔여 블록의 데이터에 적용하는 치환 모듈,
- [0057] - 선택된 쌍의 변환 연산을 치환된 데이터에 적용하는 변환 모듈,
- [0058] - 변환 모듈에 의해 전달되는 데이터를 인코딩하는 인코딩 모듈을 포함하는 것을 특징으로 한다.
- [0059] 그러한 인코딩 디바이스는 특히 앞서 언급한 인코딩하는 방법을 구현할 수 있다.
- [0060] 본 발명은 또한 블록들로 컷팅되는 적어도 하나의 이미지를 나타내는 데이터 신호를 디코딩하는 방법에 관한 것이다.
- [0061] 그러한 디코딩하는 방법은 디코딩될 현재의 블록에 대해:
- [0062] - 디코딩될 현재의 블록과 연관되는 현재의 잔여 블록을 나타내는 데이터를 데이터 신호에서 결정하는 단계,
- [0063] - 디코딩될 현재의 블록의 적어도 하나의 예측 변수 블록을 결정하는 단계,
- [0064] - 변환 연산 및 치환 연산을 포함하는 수학적 연산들의 쌍을 결정하는 단계,
- [0065] - 연산들의 결정된 쌍의 변환 연산을 현재의 잔여 블록을 나타내는 데이터에 적용하는 단계,
- [0066] - 결정된 쌍의 치환 연산을 변환 연산을 적용하는 단계의 결과로서 얻어지는 데이터에 적용하는 단계,
- [0067] - 결정된 예측 변수 블록 및 적용되는 치환 연산으로부터의 데이터를 사용하여 현재의 블록을 재건하는 단계를 포함하는 것을 특징으로 한다.
- [0068] 하나의 특정 실시예에 따르면, 수학적 연산들의 쌍을 결정하는 단계는:
- [0069] - 디코딩될 현재의 블록의 인코딩에서 선택되는 치환 및 변환 연산들의 쌍을 나타내는 정보를 데이터 신호에서 판독하는 단계,
- [0070] - 판독되는 정보와 연관되는 쌍의 치환 및 변환 연산들의 역치환 및 변환 연산들을 결정하는 단계를 구현한다.
- [0071] 그러한 일 실시예는 송신되는 치환 및 변환 연산들의 쌍을 시그널링하는 데이터의 양이 감소되며, 이는 인코딩에서 선택되었던 치환 및 변환 연산들의 쌍을 디코더 그 자체가 회수하는 것을 가능하게 하지 않는 결과를 가질 때, 구현된다.
- [0072] 다른 특정 실시예에 따르면, 수학적 연산들의 쌍을 결정하는 단계는:
- [0073] - 현재의 잔여 블록을 나타내는 데이터에 치환 연산이 뒤따르는 변환 연산을 포함하는 수학적 연산들의 복수의 쌍을 적용하는 단계로서, 변환 및 치환 연산들은 미리 결정된 복수의 변환 연산 및 미리 결정된 복수의 치환 연산 중으로부터 선택되는 단계,
- [0074] - 적용되는 복수의 쌍으로부터 복수의 블록을 재건하는 단계,
- [0075] - 복수의 재건된 블록에 적용되는 일관성의 기준을 최대화함으로써 복수의 재건된 블록 중으로부터의 재건된 블록을 선택하는 단계로 구성된다.
- [0076] 다른 그러한 실시예는 디코더가 그 자체로 인코딩에서 선택되었던 치환 및 변환 연산들의 쌍을 회수할 수 있을 때, 구현된다.

- [0077] 이러한 다른 실시예는 인코딩에서 선택되었던 치환 연산의 인덱스 및/또는 변환 연산의 인덱스가 디코더에서 수신되는 데이터 신호로 반드시 송신되는 것은 아니므로, 이전의 실시예에서 얻어지는 시그널링하는 비용보다 시그널링하는 비용의 더 큰 감소를 가능하게 한다.
- [0078] 다른 특정 실시예에 따르면, 결정되거나 선택된 변환 연산은 이산 사인 변환이다.
- [0079] 다른 특정 실시예에 따르면, 연산들의 결정된 쌍의 치환 연산은 이러한 데이터를 이항하는 연산에 따라 결정된 쌍의 변환 연산을 적용하는 단계의 결과로서 얻어지는 데이터를 점진적으로 이동시키는 것으로 구성된다.
- [0080] 다양한 앞서 언급한 실시예 또는 특징은 앞서 정의된 바와 같은 디코딩하는 방법의 단계들에 독립적으로 또는 서로와의 조합으로 추가될 수 있다.
- [0081] 상응하여, 본 발명은 또한 블록들로 컷팅되는 적어도 하나의 이미지를 나타내는 데이터 신호를 디코딩하는 디바이스에 관한 것이다.
- [0082] 그러한 디코딩 디바이스는 디코딩될 현재의 블록에 대해:
 - [0083] - 디코딩될 현재의 블록과 연관되는 현재의 잔여 블록을 나타내는 데이터를 데이터 신호에서 결정하는 모듈,
 - [0084] - 디코딩될 현재의 블록의 적어도 하나의 예측 변수 블록을 결정하는 모듈,
 - [0085] - 변환 연산 및 치환 연산을 포함하는 수학적 연산들의 쌍을 결정하는 모듈,
 - [0086] - 결정된 쌍의 변환 연산을 현재의 잔여 블록을 나타내는 데이터에 적용하는 변환 모듈,
 - [0087] - 결정된 쌍의 치환 연산을 변환된 데이터에 적용하는 치환 모듈,
 - [0088] - 결정된 예측 변수 블록 및 치환 모듈에 의해 전달되는 데이터를 사용하여 현재의 블록을 재건하는 예측 모듈을 포함하는 것을 특징으로 한다.
- [0089] 그러한 디코딩 디바이스는 특히 앞서 언급한 디코딩하는 방법을 구현할 수 있다.
- [0090] 본 발명은 또한 컴퓨터 프로그램이 컴퓨터 상에서 실행될 때, 본 발명에 따른 인코딩하고 디코딩하는 방법들 중 하나를 구현하는 명령어들을 포함하는 컴퓨터 프로그램에 관한 것이다.
- [0091] 이러한 프로그램은 임의의 프로그래밍 언어를 사용하고, 부분적으로 컴파일된 형태와 같이 소스 코드, 목적 코드, 또는 소스 코드와 목적 코드 사이의 중간 코드의 형태, 또는 임의의 다른 바람직한 형태일 수 있다.
- [0092] 본 발명은 또한 컴퓨터 프로그램이 기록되는 컴퓨터에 의해 관독 가능한 기록 매체를 목적으로 하며, 이러한 프로그램은 상술한 바와 같이 본 발명에 따른 인코딩하거나 디코딩하는 방법들 중 하나를 구현하는데 적절한 명령어들을 포함한다.
- [0093] 본 발명은 또한 컴퓨터 프로그램이 기록되는 컴퓨터에 의해 관독 가능한 기록 매체를 목적으로 하며, 이러한 프로그램은 상술한 바와 같이 본 발명에 따른 인코딩하거나 디코딩하는 방법을 구현하는데 적절한 명령어들을 포함한다.
- [0094] 기록 매체는 프로그램을 저장할 수 있는 임의의 엔티티 또는 디바이스일 수 있다. 예를 들어, 매체는 ROM 예를 들어, CD ROM 또는 마이크로 전자 회로 ROM, 또는 자기 기록 수단 예를 들어, USB 스틱 또는 하드 디스크와 같은 저장 수단을 포함할 수 있다.
- [0095] 두 번째로, 기록 매체는 무선으로 또는 다른 수단에 의해 전기 또는 광 케이블을 통하여 전달될 수 있는 전기 또는 광 신호와 같은 송신 가능 매체일 수 있다. 본 발명에 따른 프로그램은 특히, 인터넷 타입 네트워크 상에 다운로드될 수 있다.
- [0096] 대안적으로, 기록 매체는 프로그램을 포함하는 집적 회로일 수 있으며, 이러한 회로는 앞서 언급한 인코딩하거나 디코딩하는 방법을 실행시키거나 이것의 실행에 사용되는데 적절하다.
- [0097] 디코딩하는 방법, 인코딩 디바이스, 디코딩 디바이스, 컴퓨터 프로그램들 및 앞서 언급한 상응하는 기록 매체는 본 발명에 따른 인코딩하는 방법에 의해 부여되는 이점들과 적어도 동일한 이점들을 갖는다.

도면의 간단한 설명

- [0098] 다른 특징들 및 이점들이 도면들을 참조하여 설명하는 바람직한 실시예를 읽을 때에 나타날 것이다:

- 도 1은 본 발명에 따른 인코딩하는 방법의 단계들을 나타낸다.
- 도 2는 본 발명에 따른 인코딩 디바이스의 일 실시예를 나타낸다.
- 도 3a는 4x4 블록의 경우에 사용되는 다양한 치환-변환 쌍을 포함하는 표를 나타낸다.
- 도 3b 및 도 3c는 8x8 블록의 경우에 사용되는 다양한 치환-변환 쌍을 포함하는 표를 나타낸다.
- 도 4는 본 발명에 따른 디코딩 디바이스의 일 실시예를 나타낸다.
- 도 5는 본 발명에 따른 디코딩하는 방법의 주요 단계들을 나타낸다.
- 도 6a는 도 5의 디코딩하는 방법에서 구현되는 치환 및 변환 연산들의 쌍을 결정하는 단계의 제1 실시예를 나타낸다.
- 도 6b는 도 5의 디코딩하는 방법에서 구현되는 치환 및 변환 연산들의 쌍을 결정하는 단계의 제2 실시예를 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0099] 본 발명에 따른 인코딩하는 방법이 예를 들어, HEVC 표준에 순응하는 인코딩에 의해 얻어지는 2진 스트림과 유사한 2진 스트림의 이미지 또는 이미지들의 시퀀스를 인코딩하는데 사용되는 본 발명의 일 실시예를 이제 설명할 것이다.
- [0100] 이러한 실시예에서, 본 발명에 따른 인코딩하는 방법은 예를 들어, 초기에 HEVC 표준에 순응하는 인코더로의 변경에 의한 소프트웨어 또는 하드웨어로 구현된다. 본 발명에 따른 인코딩하는 방법은 도 1에 나타내어지는 바와 같이 단계들(C1 내지 C15)을 포함하는 알고리즘의 형태로 나타내어진다.
- [0101] 본 발명의 실시예에 따르면, 본 발명에 따른 인코딩하는 방법은 도 2에 나타내어지는 인코딩 디바이스(CO)로 구현된다.
- [0102] 도 2에 도시된 바와 같이, 그러한 인코딩 디바이스는 버퍼 메모리(MT_CO)를 포함하는 메모리(MEM_CO), 예를 들어, 마이크로프로세서(μ P)가 구비되고 본 발명에 따른 인코딩하는 방법을 구현하는 컴퓨터 프로그램(PG_CO)에 의해 제어되는 처리부(UT_CO)를 포함한다. 초기화에서, 컴퓨터 프로그램(PG_CO)의 코드 명령어들은 예를 들어, 처리부(UT_CO)의 프로세서에 의해 실행되기 전에, RAM 메모리(나타내어지지 않음)로 로딩된다.
- [0103] 도 1에 나타내어지는 인코딩하는 방법은 인코딩될 이미지들의 시퀀스(SQ)의 임의의 현재의 이미지에 적용된다.
- [0104] 도 1에 나타내어지는 단계(C1) 중에서, 그 자체로 알려진 방식으로, 이미지들(IC₁, ..., IC_j, ..., IC_N) (1 ≤ j ≤ N)의 시퀀스(SQ)에 속하는 현재의 이미지(IC_j)는 64x64, 32x32 또는 16x16 픽셀들인 크기의 전형적인 값들인, 주어진 크기의 복수의 블록(B₁, B₂, ..., B_u, ..., B_S) (1 ≤ u ≤ S)으로 분할된다. 그러한 분할 단계는 모듈이 처리부(UT_CO)의 마이크로프로세서(μ P)에 의해 제어되는 도 2에 나타내어지는 분할 소프트웨어 모듈(MP_CO)에 의해 구현된다.
- [0105] 본 발명의 의미 내에서, "블록"이란 용어가 (영어 용어에서) "코딩 단위"를 의미한다는 점이 주목되어야 한다. 후자 전문 용어는 예를 들어, 문서 "ITU-T 권장 H.265 / 표준 ISO/IEC 23008-2(ITU-T Recommendation H.265 / Standard ISO/IEC 23008-2)"에서의 HEVC 표준에 특히 사용된다.
- [0106] 특히, 그러한 코딩 단위는 또한 블록들, 매크로블록들 또는 다른 기하학적 형상들을 나타내는 픽셀들의 세트들로 불리는 직사각형 또는 정사각형 형상의 픽셀들의 세트들을 함께 그룹화한다.
- [0107] 바람직하게는 본 발명에서, 인코딩될 블록들은 특히 2의 배수 거듭제곱 예를 들어, 타입 4x4 및 8x8의 크기이다. 그러나, 본 발명은 또한 크기 MxM(M=16, 32 또는 64)의 블록들에 보다 일반적으로 적용된다. PxM 및 MxP 블록들이 또한 사용 가능하며, P 및 M이 설정된 2, 4, 8, 16, 32 및 64에서 P 및 M의 값을 취한다.
- [0108] 이러한 블록들(B₁, B₂, ..., B_u, ..., B_S)은 예를 들어, 래스터(raster) 스캔 타입인 미리 결정된 스캐닝 순서에 따라 인코딩되는 것으로 의도된다. 이는 블록들이 좌측에서 우측으로, 그 다음 하향으로 차례로 인코딩되는 것을 의미한다.
- [0109] 다른 타입들의 스캐닝이 물론 가능하다. 따라서, 슬라이스들로 불리는 다수의 하위 이미지로 이미지(IC_j)를 컷팅

하고 이러한 타입의 컷팅을 각각의 하위 이미지에 독립적으로 적용하는 것이 가능하다. 또한, 상술한 바와 같이, 행들의 연속이 아닌 열들의 연속을 인코딩하는 것이 가능하다. 또한, 어느 방향으로든 행들 또는 열들을 스캐닝하는 것이 가능하다.

- [0110] 각각의 블록은 하위 블록들 자체가 재분할할 수 있는 하위 블록들로 그 자체가 추가로 분할될 수 있다.
- [0111] 도 1에 나타내어지는 단계(C2) 중에서, 인코더(CO)는 예를 들어, 제1 블록(B_1)과 같은 이미지(IC_j)의 인코딩될 제1 블록(B_i)을 현재의 블록으로 선택한다.
- [0112] 도 1에 나타내어지는 단계(C3) 중에서, 현재의 블록(B_i)의 예측은 인트라 및/또는 인터 예측의 알려진 기법들에 의해 수행된다. 이러한 목적으로, 블록(B_i)은 복수의 미리 결정된 예측 모드 중으로부터 선택되는 예측 모드에 따라 적어도 하나의 예측 변수 블록에 대하여 예측된다.
- [0113] 그러한 예측 변수 블록은 예를 들어, 이미 인코딩되었거나, 인코딩된 다음 디코딩되었거나, 디코딩되지 않았던 픽셀들의 블록이다. 그러한 예측 변수 블록은 도 2에 나타내어지는 바와 같은 인코더(CO)의 버퍼 메모리(MT_CO)에 앞서 저장된다.
- [0114] 예측 단계(C3)의 끝부분에서, 최적의 예측 변수 블록(BP_{opt})은 이러한 미리 결정된 예측 모드들이 예를 들어, 당 업자에게 널리 알려진 속도-왜곡 기준을 최소화함으로써 경합하여 배치된 후에 얻어진다. 블록(BP_{opt})은 현재의 블록(B_i)의 근사치로서 고려된다. 이러한 예측과 관련한 정보 아이템들은 디코더로 송신될 신호 또는 데이터 스트림에 포함되는 것으로 의도된다. 그러한 정보 아이템들은 특히 예측의 타입(인트라 또는 인트라), 그리고 적용 가능한 경우, 선택되는 예측 모드, 현재의 블록이 재분할되었으면 현재의 블록의 분할의 타입, 참조 이미지 인덱스 및 인터 예측 모드가 선택되었던 경우에 사용되는 모션 벡터를 포함한다. 이러한 정보 아이템들은 인코더(CO)에 의해 압축된다.
- [0115] 도 1에 나타내어지는 단계(C4) 중에서, 현재의 블록(B_i)과 관련한 데이터는 예측 변수 블록(BP_{opt})의 데이터와 비교된다. 보다 정확하게는 이러한 단계 중에서, 얻어지는 예측 변수 블록(BP_{opt})과 현재의 블록(B_i) 사이의 차이가 통상적으로 계산된다.
- [0116] 잔여 블록(Br_i)으로 불리는 잔여 데이터의 세트가 그 다음 단계(C4)의 끝부분에서 얻어진다.
- [0117] 단계들(C3 및 C4)은 모듈이 처리부(UT_CO)의 마이크로프로세서(μP)에 의해 제어되는 도 2에 나타내어지는 예측 인코딩 소프트웨어 모듈(PRED_CO)에 의해 구현된다.
- [0118] 도 1에 나타내어지는 단계(C5) 중에서, 치환 연산(OP_{n^*}) 및 변환 연산(OT_{n^*})을 포함하는 수학적 연산들의 쌍이 복수의 치환 연산($OP_0, OP_1, \dots, OP_m, \dots, OP_{K-1}$)(여기서 $0 \leq m \leq K-1$ 및 $0 \leq m^* \leq K-1$), 및 복수의 변환 연산($OT_0, OT_1, \dots, OT_n, \dots, OT_{L-1}$)(여기서 $0 \leq n \leq L-1$ 및 $0 \leq n^* \leq L-1$) 중으로부터 미리 결정된 기준에 따라 선택된다. 복수의 치환 및 변환 연산은 도 2의 인코더(CO)의 전용 메모리(MD_CO)에 저장된다.
- [0119] 그러한 단계(C5)는 모듈이 처리부(UT_CO)의 마이크로프로세서(μP)에 의해 제어되는 도 2에 나타내어지는 계산 소프트웨어 모듈(CAL_CO)에 의해 구현된다.
- [0120] 복수의 변환 연산($OT_0, OT_1, \dots, OT_n, \dots, OT_{L-1}$)은 하나 이상의 1D 변환을 포함한다.
- [0121] 1D 변환은 크기 M의 벡터들을 연산하는 크기 $M \times M$ 의 이산 변환을 의미한다. 이러한 목적으로:
- [0122] - 제1 변환 연산은 크기 $M \times M$ 의 블록 상에 한 벡터씩 적용되며,
- [0123] - 이항 연산은 변환된 블록 상에 적용되며,
- [0124] - 제2 변환 연산은 또한 벡터 마다 이항 후에 얻어지는 블록 상에 적용된다.
- [0125] 이러한 1D 변환은 이산 사인(DST) 또는 이산 코사인(DCT) 변환들이라는 점에서 3각법 형태로 표현되는 기반들에 기초한다. 바람직하게는, 복수의 변환($OT_0, OT_1, \dots, OT_n, \dots, OT_{L-1}$)에 속하는 변환들은 문서 "DCT-II 변환과 DCT-VI 변환과 DST-VII 변환 사이의 관계(Relationship between DCT-II, DCT-VI, and DST-VII transforms)",

Reznik ICASSP 2013에 설명된 DCT II, III 및 IV 변환들 및 DST IV, VI 및 VII 변환들이다.

[0126] 이들 중에서, DCT II, III, 및 DST VII 및 IV 변환들은 주파수 영역에서 잔여 신호 압축을 행하는 그것들의 능력으로 인해 특히 유용하다.

[0127] 그러한 변환들은 예를 들어, 크기 4x4, 8x8 등이다.

[0128] 변형예로서, 사용되는 변환들은 교호일 수 있으며 즉, 제1 타입의 변환 및 제2 타입의 변환이 데이터가 본 설명에 후술할 것인 바와 같이 이전에 치환되었던 현재의 잔여 블록(Br_u)의 행들 및 열들 상에 각각 적용된다.

[0129] 이러한 변형예에 따르면, 앞서 언급한 변환들은 직사각형일 수도 있다. 직사각형 형상의 현재의 잔여 블록(Br_u)의 경우, 행 변환은 열 변환과 상이한 크기를 가질 것이다.

[0130] 행 변환 및 열 변환의 조합이 현재의 잔여 블록(Br_u)에 직접 적용될 수 있다는 점이 주목되어야 한다.

[0131] 영상 인코딩/디코딩의 분야에서, 고정 소수점 프로세서들 상에서 가능한 구현을 행하는 것의 목적으로, 그러한 변환들은 근사화되며 즉, 실제 값들이 유리수들에 의해 근사화된다. 그러므로, 변환들은 엄밀한 의미에서 정확하게 회수되지 않고 오히려 근사화된 형태로 회수된다.

[0132] 따라서 예를 들어, 크기 4의 DST VII 변환의 경우에, 크기 4의 DST VII 변환은 이하의 식에 따라 표현될 수 있다:

[0133]
$$S_{i,k} = \frac{2}{3} \sin \left[\pi \cdot \frac{(2k+1)(i+1)}{9} \right]$$
 여기서 $0 \leq i < 4$ 및 $0 \leq k < 4$

[0134] 이하의 계수들의 4x4 행렬이 그 다음 얻어지며, i는 열들을 따라 전개되고 k는 행들을 따라 전개된다:

[0135] 0.2280 0.4285 0.5774 0.6565

[0136] 0.5774 0.5774 0.0000 0.5774

[0137] 0.6565 0.2280 0.5774 0.4285

[0138] 0.4285 0.6565 0.5774 0.2280

[0139] 위의 이러한 계수값들은 가능한 유효 고정 소수점 구현을 만드는 이하의 방식으로 분수들의 형태로 나타내어질 수도 있다:

[0140] 29/128 55/128 74/128 84/128

[0141] 74/128 74/128 0 -74/128

[0142] 84/128 -29/128 -74/128 55/128

[0143] 55/128 -84/128 74/128 -29/128

[0144] 복수의 치환 연산(OP₀, OP₁, ..., OP_m, ..., OP_{k-1})은 하나 이상의 치환 연산을 포함한다. 본 발명의 의미 내에서, 일 예가 이하에 나타내어지는 치환 연산은 1들 및 0들 만을 포함하는 행렬의 형태로 정의된다. 이하에 나타내어진 예에서, 그러한 치환 행렬은 각각의 인덱스들 I₀, I₁, I₂, I₃, I₄, I₅, I₆, I₇, I₈, I₉, I₁₀, I₁₁, I₁₂, I₁₃, I₁₄, I₁₅의 연속적인 계수들 예를 들어, 크기 4x4의 블록에 대한 16개의 세트(E_{EN})에 적용된다.

E _{SO}	치환 행렬	E _{EN}
I ₃	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	I ₀
I ₂	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	I ₁
I ₁	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	I ₂
I ₀	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	I ₃
I ₇	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	I ₄
I ₆	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0	I ₅
I ₅	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	I ₆
I ₄	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	I ₇
I ₁₁	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0	I ₈
I ₁₀	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	I ₉
I ₉	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	I ₁₀
I ₈	0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	I ₁₁
I ₁₅	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	I ₁₂
I ₁₄	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	I ₁₃
I ₁₃	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	I ₁₄
I ₁₂	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	I ₁₅

[0145]

[0146]

위의 치환 연산의 끝부분에서, 치환된 인덱스들 I₃, I₂, I₁, I₀, I₇, I₆, I₅, I₄, I₁₁, I₁₀, I₉, I₈, I₁₅, I₁₄, I₁₃, I₁₂의 세트(E_{SO})가 얻어진다. 그러한 치환 연산은 인덱스들의 레디렉션(redirection)만을 포함하므로, 도 2의 인코더(CO)의 계산 복잡성을 만들어내지 않는 이점을 갖는다.

[0147]

앞서 언급한 단계(C5) 중에서 보다 정확하게는, 선택은 인코딩될 현재의 블록(B_i)의 인코딩 성능을 최적화하는 사용될 치환-변환 쌍에 행해진다. 그러한 선택은 수개의 방식으로 행해질 수 있다.

[0148]

일 실시예에 따르면, 단계(C5)에 사용되는 선택의 미리 결정된 기준은 양자화된 샘플들의 송신 비용으로 양자화 후에 왜곡의 가중치를 측정하는 것으로 구성되는 당업자에게 널리 알려진 속도-왜곡 기준이다. 전형적으로, 고려되는 치환(OP_m) - 변환(OT_n) 쌍의 경우, 기준(J_{m,n})은 이하의 방식으로 표현된다:

[0149]

$$J_{m,n}(\lambda) = \|x - OP_m^t OT_n^{-1} [X_{m,n}]\|^2 + \lambda \cdot R([X_{m,n}])$$

, 여기서:

[0150]

- λ는 송신 속도 제약을 제어하는 라그랑주(Lagrangian) 가중치 연산자이며,

[0151]

- ()^t는 이항 행렬 연산자이며,

[0152]

- OT_n⁻¹은 역변환을 나타내며,

[0153]

- [X_{m,n}]은 양자화 연산자에 따라 수행되는 X_{m,n}의 근사치를 나타내며, X_{m,n}은 고려되는 연산들 OT_n, OP_m의 쌍에 의해 인코딩될 현재의 블록(B_i)의 변환의 결과이며,

[0154]

- R()은 양자화된 값들을 송신하는데 필요한 속도를 계산하는 함수이다.

[0155]

함수(R)은 정보 아이템들의 완전한 인코딩에 의해 즉, 양자화되고 인코딩된 값들을 송신하는데 필요한 정보 아이템들을 카운팅하는 것을 통하여, 또는 예를 들어, 양자화된 값들의 절댓값들의 합의 계산 또는 제로로 양자화 되는 값들의 카운트에 기반할 수 있는 속도의 추정에 의해 얻어질 수 있다.

[0156]

J_{m,n}에 대해 얻어지는 다양한 값 사이의 비교를 따라서, 도 2에서의 계산 모듈(CAL_CO)에 의해 계산되는 최저값이 보유된다. 그러한 값은 J_{m*,n*}으로 표시되고 고려되는 치환(OP_{m*}) - 변환(OT_{n*}) 연산들의 쌍과 연관된다.

[0157]

본 발명의 단순화된 실시예에서, 복수의 치환 연산은 예를 들어, 2개의 치환(OP₀ 및 OP₁)을 포함하는 반면에, 복수의 변환 연산은 단일 변환(OT₀) 예를 들어, DST VII를 포함한다. 이러한 목적으로 앞서 언급한 단계(C5) 중에서, 2개만의 속도-왜곡 기준 즉:

[0158]

- 치환(OP₀) 및 변환(OT₀)에 의존하는 기준(J_{0,0}),

- [0159] - 및 치환(OP₁) 및 변환(OT₀)에 의존하는 기준(J_{1,0})이 계산된다.
- [0160] 도 2의 계산 모듈(CAL_CO)은 2개의 기준(J_{0,0} 및 J_{1,0})을 비교하고, 최저값을 갖는 것을 보유하고, 인덱스(m*,n*)을 이러한 값과 연관시킨다. 따라서, J_{0,0}<J_{1,0}이면, 그 때 m*,n*=0이며, 그렇지 않으면 m*,n*=1이다.
- [0161] 앞서 언급한 선택 단계(C5)의 실시예의 하나의 변형예에 따르면, 앞서 언급한 선택 단계(C5)는 앞서 언급한 예측 단계(C3) 동안의 최적의 예측 변수 블록(B_{opt})에 대한 탐색과 함께 수행될 수 있다. 보다 정확하게는 고려되는 각각의 예측 모드의 경우, 하나의 치환-변환 쌍은 앞서 언급한 속도-왜곡 기준(J_{m,n})을 최소화하는 것으로 결정된다. 이러한 방식으로, 기준(J_{m,n})의 추정 동안의 속도 정보 아이템들은 예측 모드의 송신을 고려한다. 따라서, 모든 예측/치환-변환 조합들은 속도-왜곡 기준을 최소화하는 것을 보유하기 위해 값이 구해진다.
- [0162] 그러한 변형예는 현재의 블록의 다양한 가능한 크기를 고려함으로써 더 확장될 수 있다. 현재의 블록의 하위 블록들로의 분할의 유용성은 이용 가능 치환-변환 쌍들을 고려함으로써 속도-왜곡 기준을 고려하여 결정될 수 있다.
- [0163] 변환-치환 쌍들의 예들은 도 3a, 도 3b 및 도 3c를 참조하여 나타내어진다.
- [0164] 도 3a의 예에서, 표(T1)은 크기 4x4의 현재의 잔여 블록(Br_u) 및 연속적인 픽셀 인덱스들(I₀ 내지 I₁₅)에 적용되는 것으로 의도되는 변환-치환 쌍들을 포함한다.
- [0165] 도 3b 및 도 3c의 예에서, 표(T2)는 크기 8x8의 현재의 잔여 블록(Br_u) 및 연속적인 픽셀 인덱스들(I₀ 내지 I₆₃)에 적용되는 것으로 의도되는 변환-치환 쌍들을 포함한다.
- [0166] 도 3a를 참조하면, 표(T1)은 예를 들어, 28번의 치환(P₀ 내지 P₂₇)을 포함한다. "픽셀 인덱스"로 칭해지는 표(T1)의 제1 열은 현재의 잔여 블록(Br_u)에서 (0과 15 사이의) 고려되는 픽셀 인덱스 번호를 나타낸다. 그 다음의 열들은 각각 특정 치환(P₀ 내지 P₂₇)과 연관되며, 치환(P₀)은 픽셀 인덱스들의 순서에 어떤 영향도 갖지 않으므로, "항등식" 치환으로 칭해진다. 주어진 치환의 경우, "픽셀 인덱스" 열에 의해 나타내어지는 각각의 픽셀 인덱스는 선택된 열의 상응하는 값에 의해 나타내어지는 픽셀 인덱스 번호로 대체된다.
- [0167] 각각의 열(P₀ 내지 P₂₇)은 고려되는 치환에 따른 특정 DST 또는 DCT 변환과 연관되며, 이러한 특정 변환은 표(T1)의 마지막 행에 나타내어진다.
- [0168] 도 3b 및 도 3c를 참조하면, 표(T2)는 예를 들어, 29번의 치환(P₀ 내지 P₂₈)을 포함한다. "픽셀 인덱스"로 칭해지는 표(T2)의 제1 열은 현재의 잔여 블록(Br_u)에서 (0과 63 사이의) 고려되는 픽셀 인덱스 번호를 나타낸다. 그 다음의 열들은 각각 특정 치환(P₀ 내지 P₂₈)과 연관되며, 치환(P₀)은 픽셀 인덱스들의 순서에 어떤 영향도 갖지 않으므로, "항등식" 치환으로 칭해진다. 주어진 치환의 경우, "픽셀 인덱스" 열에 의해 나타내어지는 각각의 픽셀 인덱스는 선택된 열의 상응하는 값에 의해 나타내어지는 픽셀 인덱스 번호로 대체된다.
- [0169] 도 3b 및 도 3c의 예에서, 각각의 치환은 동일한 DCT 변환과 연관된다.
- [0170] 표들(T1 및 T2)에 나타내어지는 치환들은 학습에 의해 수립된다. 이러한 목적으로, 치환들은 단순한 기하학적 변환들: 열들/행들의 순서 도치, 블록들의 이동, 행들 또는 열들의 이동으로부터 유도된다.
- [0171] 일부 치환은 인터-픽셀 관계가 변경되지 않을 때, 고려되는 변환들이 약간 영향을 받는다는 의미로 구현의 일부 치환의 단순함을 위해 결정되며 즉, 치환 후에, 각각의 픽셀은 치환 전의 각각의 픽셀의 이웃하는 것들에 근접하게 유지된다. 바람직한 치환은 예를 들어, 픽셀 블록의 데이터를 미리 결정된 방향으로 점진적으로 이동시키는 이항 연산자이다. 다른 치환들이 예상된다.
- [0172] 예를 들어, 도 3a에 나타내어지는 표(T1)에서:
- [0173] - 치환(P₁)은 픽셀 블록의 미러 연산자이며, 픽셀 인덱스들은 수직 대칭에 의해 서로에 대하여 도치되며,
- [0174] - 치환(P₄)은 잔여 블록(Br_u)을 분할하는 45° 각도에 대하여 이항 즉, 대각선 대칭에 의해 픽셀 인덱스들을 도

치하는 것을 가능하게 한다.

[0175] 다양한 치환이 잔여 블록들의 고유 특성들의 연구를 통해 학습되었다. 픽셀 당 평균 잔여물의 발광량에 따르면, 픽셀들은 특정 변환과 호환되도록 재순서화된다. DCT가 일정한 휘도의 프로파일을 갖는 인코딩 잔여물들에 적절한 것 즉, 잔여 블록이 픽셀 당 비교적 균일한 휘도를 갖는 것이 종래 기술에 알려져 있다. DST는 블록의 상단(예를 들어, 도 3a에서의 인덱스들(I₀ 내지 I₃))에 그리고 좌측(예를 들어, 도 3a에서의 인덱스들(I₀ 내지 I₁₂)) 상에 위치되는 픽셀들에서 더 적은 휘도를 갖는 잔여 블록들에 적절한 프로파일을 갖는다.

[0176] 따라서, 특히 도 3a의 표(T1)에서, 치환들은 마지막 행에 나타내어지는 변환에 따라 수립된다.

[0177] "항등식" 치환(P₀)은 잔여 블록(Br_u)이 연관된 변환과 호환되는 형상을 가질 때, 보유되는 치환들 중 일부를 형성한다. 다른 한편으로는, 잔여 블록이 연관된 변환과 호환되지 않을 때, 그 때 앞서 언급한 속도-왜곡 기준(J_{m,n})은 치환(P₀) 이외의 치환으로 훨씬 더 작은 값을 가질 것이다.

[0178] 치환-변환 쌍들의 다른 구성들이 물론 가능하고 일부를 본 설명에서 나중에 언급할 것이다.

[0179] 도 1에 나타내어지는 단계(C6) 중에서, 선택된 연산들의 이러한 쌍의 치환 연산(OP_{m*})이 치환된 블록(Bp_u)을 생성하기 위해 현재의 잔여 블록(Br_u)의 데이터에 적용된다.

[0180] 단계(C6)는 모듈이 처리부(UT_CO)의 마이크로프로세서(μP)에 의해 제어되는 도 2에 나타내어지는 바와 같은 치환 소프트웨어 모듈(MPER_CO)에 의해 구현된다.

[0181] 도 1에 나타내어지는 단계(C7) 중에서, 선택된 연산들의 이러한 쌍의 변환 연산(OT_{m*})이 변환된 블록(Bt_u)을 생성하기 위해 치환된 블록(Bp_u)의 데이터에 적용된다.

[0182] 단계(C7)는 모듈이 처리부(UT_CO)의 마이크로프로세서(μP)에 의해 제어되는 도 2에 나타내어지는 바와 같은 변환 소프트웨어 모듈(MT_CO)에 의해 구현된다.

[0183] 변환 연산(OT_{m*})은 이하의 형태로 공간적 영역에서 주파수 영역으로 치환된 블록(Bp_u)을 변환하는 것으로 구성된다:

[0184]
$$X_{Bt_u} = B \cdot (Ax_{Bp_u})^t$$

[0185] 여기서:

[0186] - x_{Bp_u} 는 공간적 영역에서 현재의 치환된 블록(Bp_u)을 행렬 형태로 나타내며,

[0187] - X_{Bt_u} 는 변환된 영역에서 얻어지는 블록을 행렬 형태로 나타내며,

[0188] - B 및 A는 앞서 제공된 균들로부터의 1D 변환들을 나타낸다.

[0189] 변환의 결과가 이하의 수학적으로 동등한 형태로 기록될 수도 있다는 점을 주목해야 한다:

[0190]
$$X=Sx$$

[0191] 여기서:

[0192] - X 및 x는 그 때 벡터들의 형태로 나타나며,

[0193] - S는 x에 포함되는 요소의 수의 제곱을 포함하는 변환 행렬을 나타낸다. 행렬(S)는 1D 행렬들(B 및 A)의 일반화된 곱으로서 기록된다. 즉, 행렬(S)는 블록들에 의한 행렬(A)의 반복 및 행렬(B)의 항에 의한 행렬들(A) 각각의 곱셈에 의해 구성되는 MxM 행렬(M은 1D 변환들의 크기임)을 구성함으로써 형성된다. 이하에 예시되는 바와 같이, 경우 M=4에 대해, 타입 16x16의 행렬(S)이 그 때 얻어진다:

$B_{0,0} * A$	$B_{0,1} * A$	$B_{0,2} * A$	$B_{0,3} * A$
$B_{1,0} * A$	$B_{1,1} * A$	$B_{1,2} * A$	$B_{1,3} * A$
$B_{2,0} * A$	$B_{2,1} * A$	$B_{2,2} * A$	$B_{2,3} * A$
$B_{3,0} * A$	$B_{3,1} * A$	$B_{3,2} * A$	$B_{3,3} * A$

[0194]
 [0195] 바람직한 실시예에서, 이러한 행렬(S)은 DST를 나타내는 행렬이다.

[0196] 치환 연산(OP_m)은 이하에 나타내어지는 방식으로 픽셀들에 적용되는 현재의 잔여 블록(Br_u)을 나타내는 행렬 (X_{Br_u})의 엔트리들 상에 치환을 적용하는 것으로 구성된다:

<u>Bp_u</u>	치환 행렬 P _m	<u>Br_u</u>
l ₃	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	l ₀
l ₂	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	l ₁
l ₁	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	l ₂
l ₀	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	l ₃
l ₇	0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	l ₄
l ₆	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	l ₅
l ₅	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	l ₆
l ₄	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	l ₇
l ₁₁	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	l ₈
l ₁₀	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	l ₉
l ₉	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	l ₁₀
l ₈	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	l ₁₁
l ₁₅	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	l ₁₂
l ₁₄	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	l ₁₃
l ₁₃	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	l ₁₄
l ₁₂	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	l ₁₅

[0197]
 [0198] 관계 $X_{Btu} = SOP_{m} * x_{Br_u}$ 가 그 때 얻어지며,
 [0199] 여기서, P_m은 픽셀 위치들의 도치의 효과를 갖는 x의 인덱스들의 치환 행렬을 나타낸다.

[0200] 도 1에 나타내어지는 단계(C8) 중에서, 변환된 블록(Bt_u)은 예를 들어, 스칼라 또는 벡터 양자화와 같은 통상적 양자화 연산에 따라 양자화된다. 양자화된 계수들의 블록(Bq_u)이 그 다음 얻어진다. 그러한 단계는 모듈이 처리부(UT_CO)의 마이크로프로세서(μP)에 의해 제어되는 도 2에 나타내어지는 바와 같은 양자화 소프트웨어 모듈(MQ_CO)에 의해 수행된다.

[0201] 도 1에 나타내어지는 단계(C9) 중에서, 양자화된 계수들의 블록(Bq_u)의 데이터가 인코딩된다. 바람직한 실시예에서, 수행되는 인코딩은 이와 같이 알려져 있는 연산 또는 허프먼(Huffman) 엔트로피 인코딩이다. 단계(C9)는 그 때:

- [0202] - 이러한 현재의 블록과 연관되는 심볼 또는 심볼들을 판독하는 단계,
- [0203] - 비트들과 같은 디지털 정보를 판독되는 심볼(들)과 연관시키는 단계로 구성된다.

[0204] 그러한 엔트로피 인코딩 단계는 모듈이 처리부(UT_CO)의 마이크로프로세서(μP)에 의해 제어되는 도 2에 나타내

어지는 엔트로피 인코딩 소프트웨어 모듈(MCE)에 의해 구현된다. 엔트로피 인코딩 모듈(MCE)은 예를 들어, CABAC(영어로 "컨텍스트 적응형 2진 연산 코더") 타입이다. 그것은 이와 같이 알려져 있는 허프먼 인코더일 수도 있다.

[0205] 이러한 단계(C9) 중에서, 앞서 언급한 단계(C5)에서 보유되는 기준(J_{m^*,n^*})의 인덱스(m^*,n^*)이 선택적으로 인코딩된다. 그러한 단계는 디코더가 이러한 정보를 회수할 수 없을 때, 필요하다.

[0206] m^*,n^* 이 디코더로 송신되어야 한다면, m^*,n^* 은 그 때 예를 들어, 고정 길이 2진 코드로 표현된다.

[0207] 예를 들어, 치환-변환 쌍들의 4개의 값: 0, 1, 2, 3이 시그널링되어야 한다면,

[0208] - 값 0은 예를 들어, 코드 00과 연관되며,

[0209] - 값 1은 예를 들어, 코드 01과 연관되며,

[0210] - 값 2는 예를 들어, 코드 10과 연관되며,

[0211] - 값 3은 예를 들어, 코드 11과 연관된다.

[0212] 변형예로서, 인덱스(m^*,n^*)은 가변 길이 코드를 사용하여 표현될 수 있다.

[0213] 예를 들어, 시그널링될 치환-변환 쌍들의 4개의 값: 0, 1, 2, 3의 경우에:

[0214] - 값 0은 예를 들어, 코드 0과 연관되며,

[0215] - 값 1은 예를 들어, 코드 10과 연관되며,

[0216] - 값 2는 예를 들어, 코드 110과 연관되며,

[0217] - 값 3은 예를 들어, 코드 111과 연관된다.

[0218] 바람직한 실시예에 따르면, 인덱스(m^*,n^*)의 송신은 2개의 표시기:

[0219] - 위의 코드들의 제1 비트를 시그널링하는 플래그,

[0220] - 및 2진 형태로 될 개연적인 치환-변환 쌍들을 나타내는 상호 보완적 신택스(syntax) 요소로 분리된다.

[0221] 플래그는 유리하게는 보유되는 쌍 번호의 인코딩으로부터 분리되는 연산 인코더를 사용하여 인코딩된다. CABAC 인코더는 바람직하게는 현재의 블록의 재분할의 크기에 따라 특정 맥락으로 사용된다. 따라서, 이러한 플래그가 주어진 상태에서 매우 흔하게 채용될 때 즉, 이러한 플래그의 '0' 또는 '1'일 확률이 매우 높을 때, 이러한 플래그의 비용의 상당한 감소를 얻는 것이 가능하다.

[0222] 따라서, 이러한 바람직한 모드에 따르면:

[0223] - 값 0은 예를 들어, 0으로 설정되는 플래그와 연관되며,

[0224] - 값 1은 예를 들어, 1로 설정되는 플래그, 및 보수 0와 연관되며,

[0225] - 값 2는 예를 들어, 1로 설정되는 플래그, 및 보수 10과 연관되며,

[0226] - 값 3은 예를 들어, 1로 설정되는 플래그, 및 보수 11과 연관된다.

[0227] 다른 바람직한 실시예에 따르면, 인덱스(m^*,n^*)은 플래그에 의해서만 송신된다. 이러한 바람직한 모드는 단일 변환 예를 들어, DST가 적용되고 2개의 치환 연산(OP₀ 및 OP₁)이 DST와의 조합으로 적용될 수 있는 경우에 적용 가능하다. 앞서 언급한 단계(C5)에서 선택되는 치환-변환 쌍(SOP₀ 또는 SOP₁)은 그 때 1 비트로 시그널링될 것이다. 예로서 주어진 관계에 따르면:

[0228] - 쌍(SOP₀)의 선택은 '1'로 설정되는 1 비트로 나타내어지며,

[0229] - 쌍(SOP₁)의 선택은 '0'으로 설정되는 1 비트로 나타내어진다.

[0230] 이러한 바람직한 실시예에서, CABAC 인코더는 바람직하게는 현재의 블록의 재분할의 크기에 따라 특정 맥락으로 사용된다. 따라서, 이러한 플래그가 주어진 상태에서 매우 흔하게 채용될 때 즉, 이러한 플래그의 '0' 또는 '1'일 확률이 매우 높을 때, 이러한 플래그의 비용의 상당한 감소를 얻는 것이 가능하다.

1'일 확률이 매우 높을 때, 이러한 플래그의 비용의 상당한 감소를 얻는 것이 가능하다.

- [0231] 2개보다 더 많은 치환-변환 쌍이 선택될 수 있는 경우에, 고정 또는 가변 길이 코드가 사용될 것이다.
- [0232] 다른 가능한 실시예에 따르면, 단계(C9) 중에서, 인덱스(m^*, n^*)의 값의 부분적 인코딩이 수행된다. 본 설명에서 나중에 설명할 것인 바와 같이, 그러한 일 실시예는 디코더가 그 자체로 인덱스(m^*, n^*) 중 모두 또는 일부를 회수할 수 있을 때, 가능하다.
- [0233] 변형예로서, 이러한 다른 실시예는 상술한 바람직한 실시예에서 구현되는 시그널링과 결합될 수도 있다.
- [0234] 예로서 주어진 특정 구성에 따르면:
- [0235] - 단일 변환(S) 예를 들어, DST가 적용되고 치환 연산(OP₀)이 DST와의 조합으로 적용되는 경우에, 앞서 언급한 단계(C5)에서 선택되는 치환-변환 쌍(SOP₀)은 그 때 '0'으로 설정되는 1 비트로 시그널링될 것이며,
- [0236] - 단일 변환(S) 예를 들어, DST가 적용되고 다수의 치환 연산이 DST와의 조합으로 적용될 수 있는 경우에, 앞서 언급한 단계(C5)에서 선택되는 치환-변환 쌍은 그 때 '1'로 설정되는 1 비트로 시그널링될 것이다.
- [0237] 인코딩 단계(C9)의 끝부분에서, 양자화된 계수들의 블록(B_{q_u})의 인코딩된 데이터의 세트 및 선택적으로 인코딩되는 인덱스(m^*, n^*)의 전체 또는 하나의 부분을 포함하는 신호 또는 데이터 스트림(F)이 그 때 전달된다. 그러한 스트림은 그 때 통신 네트워크(나타내어지지 않음)를 통하여 원격 단말기로 송신된다. 이는 도 4에 나타내어지는 디코더(D0)를 포함한다. 그 자체로 알려진 방식으로, 스트림(F)는 예측의 타입(인터 또는 인트라), 그리고 적용 가능한 경우, 선택되는 예측 모드, 블록이 분할되었으면 블록의 분할의 타입, 참조 이미지 인덱스 및 인터 예측 모드에 사용되는 모션 벡터와 같은 인코더(C0)에 의해 인코딩되는 일부 정보를 더 포함한다.
- [0238] 도 1에 나타내어지는 단계(C10) 중에서, 블록(B_{q_u})은 통상적 역양자화 연산에 따라 역양자화되며, 통상적 역양자화 연산은 단계(C8)에서 수행되는 양자화의 반대 연산이다. 역양자화된 계수들의 블록(BD_{t_u})이 그 다음 얻어진다.
- [0239] 그러한 역양자화 단계는 모듈이 처리부(UT_CO)의 마이크로프로세서(μP)에 의해 제어되는 도 2에 나타내어지는 바와 같은 역양자화 소프트웨어 모듈(MQ⁻¹_CO)에 의해 수행된다.
- [0240] 도 1에 나타내어지는 단계(C11) 중에서, 변환 연산(OT_{n*})의 반대인 변환 연산(OT⁻¹_{n*})은 디코딩된 치환된 블록(BD_{p_u})을 생성하기 위해 역양자화된 계수들의 블록(BD_{t_u})의 데이터에 적용된다.
- [0241] 단계(C11)는 모듈이 처리부(UT_CO)의 마이크로프로세서(μP)에 의해 제어되는 도 2에 나타내어지는 바와 같은 변환 소프트웨어 모듈(MT⁻¹_CO)에 의해 구현된다.
- [0242] 도 1에 나타내어지는 단계(C12) 중에서, 치환 연산(OP_{m*})의 반대인 치환 연산(OP⁻¹_{m*})이 디코딩된 잔여 블록(BD_{r_u})을 생성하기 위해 디코딩된 치환된 블록(BD_{p_u})의 데이터에 적용된다.
- [0243] 단계(C12)는 모듈이 처리부(UT_CO)의 마이크로프로세서(μP)에 의해 제어되는 도 2에 나타내어지는 바와 같은 치환 소프트웨어 모듈(MPER⁻¹_CO)에 의해 구현된다.
- [0244] 도 1에 나타내어지는 단계(C13) 중에서, 디코딩된 블록(BD_u)은 디코딩된 잔여 블록(BD_{r_u})을 앞서 언급한 단계(C3)에서 얻어지는 예측 변수 블록(BP_{opt})에 추가함으로써 구성된다. 이러한 마지막 블록이 본 설명에서 나중에 설명할 이미지(IC_i)를 디코딩하는 방법의 끝부분에서 얻어지는 디코딩된 블록과 동일하다는 점이 주목될 것이다. 디코딩된 블록(BD_u)은 그 다음 인코딩될 그 다음의 블록에 대한 예측 변수 블록으로서 인코더(C0)에 의해 사용되기 위해 도 2에서의 버퍼 메모리(MT_CO)에 저장된다.
- [0245] 그러한 단계는 모듈이 처리부(UT_CO)의 마이크로프로세서(μP)에 의해 제어되는 도 2에 나타내어지는 역예측 인코딩 소프트웨어 모듈(PRED⁻¹_CO)에 의해 구현된다.

- [0246] 도 1에 나타내어지는 단계(C14) 중에서, 인코더(CO)는 이제 막 인코딩되었던 현재의 블록(B_i)이 이미지(IC_j)의 마지막 블록인지 여부를 검사한다.
- [0247] 현재의 블록이 이미지(IC_j)의 마지막 블록이면, 도 1에 나타내어지는 그 다음의 단계(C15) 중에서, 인코딩하는 방법이 종료된다.
- [0248] 그러한 경우가 아니라면, 인코딩될 그 다음의 블록을 선택하는 단계(C2)가 앞서 언급한 래스터 스캔 스캐닝 순서에 따라 반복되며, 그 다음 단계들(C3 내지 C14)이 이러한 선택되는 그 다음의 블록에 대해 반복된다.
- [0249] 이제 막 상술하였던 인코딩 단계들이 고려되는 현재의 이미지(IC_j)의 인코딩될 모든 블록들($B_1, B_2, \dots, B_u, \dots, B_s$)에 대해 구현된다.
- [0250] 치환-변환 쌍들의 특정 구성들
- [0251] 치환-변환 쌍들의 다른 가능한 구성들이 부가 예들로서 이하에 이제 언급될 것이다.
- [0252] 일 실시예에 따르면, 특히 도 3a 내지 도 3c와 관련하여 앞서 제공된 치환들 및 변환들의 세트를 사용하는 것보다 오히려, 이러한 세트가 제한되는 것이 제공된다.
- [0253] 따라서 크기 4×4 의 블록들의 경우, 예를 들어, 치환-변환 쌍의 수에 따른 다양한 중간물이 이하에 제공된다.
- [0254] 이러한 목적으로, 사용될 다수의 상이한 치환-변환 쌍을 한정하는 $1+Y$ 모드가 결정된다.
- [0255] $Y=0$ 의 경우, $1+0$ 모드는 하나의 변환 및 하나의 "항등식" 치환(P_0)을 포함하는 쌍을 한정한다.
- [0256] $Y>0$ 의 경우, $1+Y$ 모드는 복수의 변환 및 복수의 치환의 조합으로부터 형성되는 다수의 쌍을 한정한다.
- [0257] 예로서:
- [0258] - $Y=1$ 의 경우, $1+1$ 모드는 2개의 가능한 변환 및 치환 쌍을 한정하는 모드이며, 2개의 상이한 변환: 하나의 "항등식" 치환(P_0) 및 하나의 DST 변환으로 형성되는 하나의 쌍 및 하나의 "항등식" 치환(P_0) 및 하나의 DCT 변환으로 형성되는 다른 쌍이 제공되며,
- [0259] - $Y=2$ 의 경우, $1+2$ 모드는 3개의 가능한 변환 및 치환 쌍을 한정하는 모드이며, 2개의 상이한 변환 및 3개의 상이한 치환(P_0, P_1, P_2)이 이하의 3개의 치환-변환 쌍: P_0 -DST, P_1 -DST, P_2 -DCT을 구성하도록 제공된다.
- [0260] 앞서 언급한 $1+0, 1+1$ 및 $1+2$ 모드들은 사용될 변환의 수를 감소시키기 위해 복잡한 다양한 중간물을 제공하여, 최고 치환-변환 쌍을 선택하기 위한 인코딩 동안의 검사될 쌍의 수를 감소시킨다.
- [0261] 따라서 $1+1$ 모드에서, 도 1에 나타내어지는 선택 단계(C5) 중에서, DCT 및 DST 변환들만이 단일의 가능한 치환만이 제공되므로, 속도-왜곡 기준($J_{m,n}$)에 따라 비교될 필요가 있다.
- [0262] $1+2$ 모드에서, 앞서 언급한 선택 단계(C5) 중에서, 3개의 속도-왜곡 기준만이 3개의 치환-변환 쌍만이 그러한 모드에 있으므로, 비교될 필요가 있다.
- [0263] 치환들의 저장은 치환-변환 쌍의 수가 제한되는 경우에 감소될 수도 있다.
- [0264] 인코더(CO)에서 이용 가능하게 되는 치환-변환 쌍의 수 및 블록들의 크기에 따른 중간물들의 세트가 이하에 나타내어진다.
- [0265] P_0+dst 모드 즉, 순전히 dst 접근법이 예를 들어, DCT와 같은 다른 변환들에 기반하여 치환-변환 쌍들의 세트와 결합하여 배치된다.

모드	1+1		1+2		1+3		1+4		1+8		1+16	
쌍	치환	변환	치환	변환	치환	변환	치환	변환	치환	변환	치환	변환
0	P ₀	dst	P ₀	dst	P ₁	dst	P ₀	dst	P ₀	dst	P ₀	dst
1	P ₀	dct	P ₁	dst	P ₂₇	dst	P ₂	dst	P ₅	dst	P ₁₁	dst
2			P ₂	dct	P ₆	dst	P ₁	dst	P ₆	dst	P ₁₂	dst
3							P ₃	dct	P ₇	dst	P ₁₃	dst
4							P ₄	dct	P ₁	dst	P ₁₄	dst
5									P ₈	dct	P ₁₅	dst
6									P ₉	dct	P ₁₆	dst
7									P ₃	dct	P ₁₇	dst
8									P ₁₀	dct	P ₁₈	dst
9											P ₁₉	dct
10											P ₂₀	dct
11											P ₂₁	dct
12											P ₂₂	dct
13											P ₂₃	dct
14											P ₂₄	dct
15											P ₂₅	dct
16											P ₂₆	dct

크기 4x4의 블록들에 대한 치환-변환 쌍의 수

[0266]

모드	1+2		1+3		1+4		1+8		1+16	
쌍	치환	변환	치환	변환	치환	변환	치환	변환	치환	변환
0	P ₀	dct	P ₀	dct	P ₀	dct	P ₀	dct	P ₀	dct
1	P ₁	dct	P ₃	dct	P ₅	dct	P ₅	dct	P ₉	dct
2	P ₂	dct	P ₄	dct	P ₆	dct	P ₈	dct	P ₁₀	dct
3			P ₅	dct	P ₇	dct	P ₉	dct	P ₁₁	dct
4					P ₈	dct	P ₁₆	dct	P ₁₂	dct
5							P ₁₇	dct	P ₁₃	dct
6							P ₁₈	dct	P ₁₄	dct
7							P ₂₂	dct	P ₁₅	dct
8							P ₂₃	dct	P ₁₉	dct
9									P ₂₀	dct
10									P ₂₁	dct
11									P ₂₂	dct
12									P ₂₄	dct
13									P ₂₅	dct
14									P ₂₆	dct
15									P ₂₇	dct
16									P ₂₈	dct

크기 8x8의 블록들에 대한 치환-변환 쌍의 수

[0267]

[0268]

따라서, 현재의 블록의 인코딩은 인코더(CO)에 복수의 상이한 치환-변환 쌍을 제공함으로써 효과적으로 개선된다.

[0269]

또한, 치환-변환 쌍들의 크기들을 조합하는 것이 가능하다. 예를 들어, 복잡성 및 품질의 매우 양호한 중간물이 4x4 블록들에 대해 1+2 모드 및 8x8 블록들에 대해 1+8 모드를 보유함으로써 달성될 수 있다. 인코더(CO)는 그 다음 속도-왜곡 기준을 최소화함으로써 크기 8x8에 대해 동일한 제공되는 것들 중으로부터 최고 4x4 변환을 선택한다. 조합된 선택은 현재의 블록의 분할을 선택할 때 행해질 수 있다. 예로서, 이하의 조합들은 주어진 이미지의 품질에 대한 송신 속도를 감소시키는 면에서 조합들의 연관된 성능들과 함께 이하에 제공된다:

조합	4x4 모드	8x8 모드	성능
0	1+2	$P_0 + dct$	-0.76%
1	1+4	$P_0 + dct$	-0.78%
2	1+8	$P_0 + dct$	-0.86%
3	1+16	$P_0 + dct$	-0.80%
4	1+2	1+2	-0.72%
5	1+2	1+4	-0.78%
6	1+2	1+8	-1.09%
7	1+2	1+16	-0.92%
8	$P_0 + dst$	1+2	-0.10%
9	$P_0 + dst$	1+4	-0.24%
10	$P_0 + dst$	1+8	-0.37%
11	$P_0 + dst$	1+16	-0.41%

[0270]

[0271] 특히, 1.09%의 속도의 감소가 4x4 블록들에 대한 1+2 모드 및 8x8 블록들에 대한 1+8 모드를 사용하는 조합 6을 사용하여 얻어진다. 이러한 속도의 감소는 영상 인코딩 특히, 낮은 복잡성의 기법에 대해 매우 상당하다.

[0272] 도 2에서의 인코더(CO)가 기준(J_{m^*,n^*})의 인덱스(m^*, n^*)의 전체를 송신하는데 필요하지 않은 경우, 증가된 인코딩 성능들은 이하에 나타내어진 바와 같이 얻어진다:

4x4 모드	성능
1+2	-4.36%
1+4	-6.13%
1+8	-8.28%
1+16	-10.35%

[0273]

[0274] 디코딩부의 상세한 설명

[0275] 디코딩하는 방법이 HEVC 표준에 초기에 순응하는 디코더에 대한 변경들에 의해 소프트웨어 또는 하드웨어로 구현되는 본 발명에 따른 디코딩하는 방법의 일 실시예를 이제 설명할 것이다. 본 발명에 따른 디코딩하는 방법은 도 5에 나타내어지는 바와 같이 단계들(D1 내지 D12)을 포함하는 알고리즘의 형태로 나타내어진다.

[0276] 도 4에 도시된 바와 같이, 본 발명의 이러한 실시예에 따른 디코더(DO)는 버퍼 메모리(MT_DO)를 포함하는 메모리(MEM_DO), 예를 들어, 마이크로프로세서(μP)가 구비되고 본 발명에 따른 디코딩하는 방법을 구현하는 컴퓨터 프로그램(PG_DO)에 의해 제어되는 처리부(UT_DO)를 포함한다. 초기화 시에, 컴퓨터 프로그램(PG_DO)의 코드 명령어들은 예를 들어, 처리부(UT_DO)의 프로세서에 의해 실행되기 전에, RAM 메모리로 로딩된다.

[0277] 도 5에 나타내어지는 디코딩하는 방법은 디코딩될 이미지들의 시퀀스(SQ)의 임의의 현재의 이미지에 적용된다.

[0278] 이러한 목적으로, 디코딩될 현재의 이미지(IC_j)를 나타내는 정보 아이템들은 도 1의 인코딩하는 방법의 결과로서 전달되는 디코더에서 수신되는 신호 또는 데이터 스트림(F)에서 식별된다.

[0279] 도 5를 참조하면, 단계(D1) 중에서, 양자화된 잔여물들($B_{q1}, B_{q2}, \dots, B_{qu}, \dots, B_{qs}$) ($1 \leq u \leq S$)는 도 1의 인코딩하는 방법에 따른 앞서 언급한 래스터 스캔 스캐닝에 따라 이전에 인코딩된 블록들($B_1, B_2, \dots, B_u, \dots, B_s$)과 연관되는 이러한 스트림(F)에서 식별된다.

[0280] 그러한 신원 확인 단계는 도 4에 나타내어지는 바와 같은 스트림 분석 신원 확인 모듈(MI_DO)에 의해 구현되며, 이러한 모듈은 처리부(UT_DO)의 마이크로프로세서(μP)에 의해 제어된다.

[0281] 이러한 양자화된 잔여물들($B_{q1}, B_{q2}, \dots, B_{qu}, \dots, B_{qs}$)은 미리 결정된 스캐닝 순서에 따라 디코딩되는 것으로

의도되며, 미리 결정된 스캐닝 순서는 예를 들어, 순차적이며 즉, 이러한 잔여물들은 이러한 잔여물들이 인코딩되었던 래스터 스캔 순서에 따라 차례로 디코딩되는 것으로 의도된다.

- [0282] 이제 막 상술했던 스캐닝의 타입 이외의 스캐닝의 타입들이 물론 가능하고 예들이 앞서 언급되었던 인코딩에서 선택되는 스캐닝 순서에 의존한다.
- [0283] 바람직하게는 본 발명에서, 디코딩될 블록들은 특히 2의 배수 거듭제곱 예를 들어, 4x4 및 8x8인 크기이다. 그러나, 본 발명은 또한 크기 $M \times M$ ($M=16, 32$ 또는 64)의 블록들에 보다 일반적으로 적용된다. $P \times M$ 및 $M \times P$ 블록들이 또한 사용 가능하며, P 및 M 이 설정된 2, 4, 8, 16, 32 및 64에서 P 및 M 의 값을 취한다.
- [0284] 각각의 블록은 하위 블록들 자체가 재분할할 수 있는 하위 블록들로 그 자체가 추가로 분할될 수 있다.
- [0285] 도 5에 나타내어지는 단계(D2) 중에서, 도 4의 디코더(D0)는 제1 양자화된 잔여 블록(B_{q_1})을 현재의 블록으로 선택한다.
- [0286] 도 5에 나타내어지는 단계(D3) 중에서, 디코딩될 현재의 블록(B_c)의 예측은 인트라 및/또는 인터 예측의 알려진 기법들에 의해 수행된다. 이러한 목적으로, 디코딩될 블록(B_c)은 복수의 미리 결정된 예측 모드 중으로부터 선택되는 예측 모드에 따라 적어도 하나의 예측 변수 블록에 대하여 예측된다.
- [0287] 특히, 선택되는 예측 모드는 도 2의 인코더(C0)가 도 4의 디코더(D0)로 이러한 모드를 시그널링했던 경우에 스트림(F)에서 직접 관독된다.
- [0288] 그러한 예측 변수 블록은 예를 들어, 이미 디코딩되거나 디코딩되지 않았던 픽셀들의 블록이다. 그러한 예측 변수 블록은 도 4에 나타내어지는 바와 같이 디코더(D0)의 버퍼 메모리(MT_DO)에 앞서 저장된다.
- [0289] 예측 단계(D3)의 끝부분에서, 선택되는 예측 모드에 순응하는 최적 예측 변수 블록(BP_{opt})이 얻어진다. 블록(BP_{opt})은 디코딩될 현재의 블록(B_c)의 근사치이다. 이러한 예측과 관련하는 정보 아이템들은 데이터 스트림(F)에서 관독된다. 그러한 정보 아이템들은 특히 예측의 타입(인트라 또는 인트라), 현재의 블록이 재분할되었으면 현재의 블록의 분할의 타입, 참조 이미지 인덱스 및 인터 예측 모드가 선택되었던 경우에 사용되는 모션 벡터를 포함한다. 이러한 정보 아이템들은 통상적 방식으로 나중에 디코딩될 것이다.
- [0290] 단계(D3)는 모듈이 처리부(UT_DO)의 마이크로프로세서(μP)에 의해 제어되는 도 4에 나타내어지는 예측 디코딩 소프트웨어 모듈(PRED_DO)에 의해 구현된다.
- [0291] 도 5에 나타내어지는 단계(D4) 중에서, 양자화된 잔여 블록(B_{q_1})의 데이터는 디코딩된 양자화된 잔여 블록(BD_{q_1})을 생성하기 위해 디코딩된다. 바람직한 실시예에서, 수행되는 디코딩은 이와 같이 알려져 있는 연산 또는 허프먼 엔트로피 디코딩이다. 단계(D4)는 그 때:
 - [0292] - 디코딩될 이러한 현재의 블록과 연관되는 심볼 또는 심볼들을 관독하는 단계,
 - [0293] - 비트들과 같은 디지털 정보를 관독되는 심볼(들)과 연관시키는 단계로 구성된다.
- [0294] 그러한 엔트로피 디코딩 단계는 모듈이 처리부(UT_DO)의 마이크로프로세서(μP)에 의해 제어되는 도 4에 나타내어지는 엔트로피 디코딩 소프트웨어 모듈(MDE)에 의해 구현된다. 엔트로피 디코딩 모듈(MDE)은 예를 들어, CABAC 타입이다. 그것은 이와 같이 알려져 있는 허프먼 디코더일 수도 있다.
- [0295] 도 5에 나타내어지는 단계(D5) 중에서, 양자화된 잔여 블록(B_{q_1})은 통상적 역양자화 연산에 따라 역양자화되며, 통상적 역양자화 연산은 도 1의 단계(C7)에서 수행되는 양자화 연산의 반대 연산이다. 디코딩된 역양자화된 블록(BD_{t_1})이 그 다음 얻어진다.
- [0296] 그러한 역양자화 단계는 처리부(UT_DO)의 마이크로프로세서(μP)에 의해 제어되는 도 4에 나타내어지는 바와 같은 역양자화 소프트웨어 모듈(MQ^{-1}_{DO})에 의해 수행된다.
- [0297] 도 5에 나타내어지는 단계(D6) 중에서, 변환(OT_{n^*}) 및 치환(OP_{m^*}) 연산들 각각의 반대 연산들인, 변환 연산($OT_{n^*}^{-1}$) 및 치환 연산($OP_{m^*}^{-1}$)을 포함하는 수학적 연산들의 쌍이 결정되며, 이러한 쌍은 도 1의 앞서 언급한 선택 단

계(C5)에서 보유되는 미리 결정된 기준(J_{m^*,n^*})에 순응한다.

- [0298] 그러한 단계(D6)는 모듈이 처리부(UT_DO)의 마이크로프로세서(μP)에 의해 제어되는 도 4에 나타내어지는 결정 소프트웨어 모듈(MDT_DO)에 의해 구현된다.
- [0299] 도 6a에 나타내어지는 바와 같이, 앞서 언급한 단계(C5)에서 보유되는 기준(J_{m^*,n^*})의 인덱스(m^*, n^*)가 도 1의 인코딩 단계(C9)의 결과로서 인덱스(m^*, n^*)의 전체로 스트림(F)으로 송신되었던 경우에, 결정 단계(D6)는:
- [0300] - 도 6a에 나타내어지는 하위 단계(D610) 중에서, 판독되는 인덱스와 연관되는 치환(OP_{m^*}) 및 변환(OT_{n^*}) 연산들의 쌍을 회수하기 위해 데이터 신호(F)에서 이러한 인덱스를 판독하는 단계,
- [0301] - 도 6a에 나타내어지는 하위 단계(D611) 중에서, 판독되는 인덱스와 연관되는 쌍의 치환(OP_{m^*}) 및 변환(OT_{n^*}) 연산들의 역치환($OP_{m^*}^{-1}$) 및 변환($OT_{n^*}^{-1}$) 연산들을 결정하는 단계로 구성된다.
- [0302] 그러한 단계들은 도 4의 디코더(D0)가 그 자체로 인덱스(m^*, n^*)의 전체를 회수할 수 없을 때, 필요하다.
- [0303] 하위 단계들(D610 및 D611)은 도 4에 나타내어지는 소프트웨어 하위 모듈(SMDT1_DO)에 의해 구현된다.
- [0304] 이러한 단계(D6) 중에서, 인덱스(m^*, n^*)는 디코딩된다.
- [0305] 보다 정확하게는, 디코더(D0)는 기준(J_{m^*,n^*})과 연관되는 치환-변환 쌍의 값을 회수하고 상응하는 역치환-역변환을 치환-변환 쌍의 값과 연관시키기 위해 예를 들어, 고정 길이 2진 코드를 판독한다.
- [0306] 예를 들어, 앞서 언급한 단계(C9)와 관련하여 설명했던 바와 같이, 치환-변환 쌍들의 4개의 값: 0, 1, 2, 3이 스트림(F)으로 시그널링될 것 같은 경우에,
- [0307] - 디코더(D0)가 코드 00을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*,n^*})과 연관되었던 치환-변환 쌍에 값 0을 할당하며,
- [0308] - 디코더(D0)가 코드 01을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*,n^*})과 연관되었던 치환-변환 쌍에 값 1을 할당하며,
- [0309] - 디코더(D0)가 코드 10을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*,n^*})과 연관되었던 치환-변환 쌍에 값 2를 할당하며,
- [0310] - 디코더(D0)가 코드 11을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*,n^*})과 연관되었던 치환-변환 쌍에 값 3을 할당한다.
- [0311] 앞서 언급한 단계(C9)와 관련하여 설명했던 변형예에서, 디코더(D0)에 의해 판독되도록 의도되는 인덱스(m^*, n^*)는 가변 길이 코드를 사용하여 표현된다.
- [0312] 예를 들어, 앞서 언급한 단계(C9)와 관련하여 설명했던 바와 같이, 치환-변환 쌍들의 4개의 값: 0, 1, 2, 3이 스트림(F)으로 시그널링될 것 같은 경우에,
- [0313] - 디코더(D0)가 코드 0을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*,n^*})과 연관되었던 치환-변환 쌍에 값 0을 할당하며,
- [0314] - 디코더(D0)가 코드 10을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*,n^*})과 연관되었던 치환-변환 쌍에 값 1을 할당하며,
- [0315] - 디코더(D0)가 코드 110을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*,n^*})과 연관되었던 치환-변환 쌍에 값 2를 할당하며,
- [0316] - 디코더(D0)가 코드 111을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*,n^*})과 연관되었던 치환-변환 쌍에 값 3을 할당한다.

- [0317] 앞서 언급한 단계(C9)와 관련하여 설명하는 바람직한 실시예에 따르면, 디코더(D0)에 의해 판독되는 것으로 도되는 인덱스(m^*, n^*)는 2개의 표시기:
- [0318] - 인코딩에서 선택되는 치환-변환 쌍(OP_{m^*}, OT_{n^*})과 연관되는 코드의 제1 비트를 시그널링하는 플래그, 및
- [0319] - 2진 형태로 덜 개연적인 치환-변환 쌍들을 나타내는 상호 보완적 신텍스 요소를 포함한다.
- [0320] 예를 들어, 앞서 언급한 단계(C9)와 관련하여 설명했던 바와 같이, 치환-변환 쌍들의 4개의 값: 0, 1, 2, 3이 스트림(F)으로 시그널링될 것 같은 경우에,
- [0321] - 디코더(D0)가 플래그 '0'을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*, n^*})과 연관되었던 치환-변환 쌍에 값 0을 할당하며,
- [0322] - 디코더(D0)가 플래그 '1' 및 2진 보수 '0'을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*, n^*})과 연관되었던 치환-변환 쌍에 값 1을 할당하며,
- [0323] - 디코더(D0)가 플래그 '1' 및 2진 보수 '10'을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*, n^*})과 연관되었던 치환-변환 쌍에 값 2를 할당하며,
- [0324] - 디코더(D0)가 플래그 '1' 및 2진 보수 '11'을 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*, n^*})과 연관되었던 치환-변환 쌍에 값 3을 할당한다.
- [0325] 앞서 언급한 단계(C9)와 관련하여 설명하는 다른 바람직한 실시예에 따르면, 디코더(D0)에 의해 판독되는 것으로 의도되는 인덱스(m^*, n^*)는 하나의 플래그만으로 구성된다. 이러한 다른 바람직한 모드는 단일 변환 예를 들어, DST가 코딩에서 적용되고 2개의 치환 연산(OP_0 및 OP_1)이 DST와의 조합으로 적용될 것 같은 경우에 적용 가능하다.
- [0326] 이러한 목적으로,
- [0327] - 디코더(D0)가 '1'로 설정되는 비트를 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*, n^*})과 연관되었던 치환-변환 쌍(SOP_0)에 이러한 비트의 값을 할당하며,
- [0328] - 디코더(D0)가 '0'으로 설정되는 비트를 판독하면, 디코더(D0)는 인코딩에서, 기준(J_{m^*, n^*})과 연관되었던 치환-변환 쌍(SOP_1)에 이러한 비트의 값을 할당한다.
- [0329] 2개보다 더 많은 치환-변환 쌍이 스트림(F)으로 시그널링될 것 같은 경우에, 디코더(D0)는 고정 또는 가변 길이 코드를 판독하는데 적절하다.
- [0330] 앞서 언급한 단계(C9)와 관련하여 설명하는 다른 가능한 실시예에 따르면, 인덱스(m^*, n^*)는 스트림(F)으로 시그널링되지 않거나 인덱스(m^*, n^*)의 일부만이 스트림(F)으로 시그널링된다.
- [0331] 도 6b에 나타내어지는 바와 같이, 앞서 언급한 단계(C5)에서 보유되는 기준(J_{m^*, n^*})의 인덱스(m^*, n^*)가 도 1의 인코딩 단계(C9)의 결과로서 스트림(F)으로 송신되지 않았던 경우에, 결정 단계(D6)는:
- [0332] - 도 6b에 나타내어지는 하위 단계(D620) 중에서, 디코딩된 역양자화된 블록(BD_{t_u})을 나타내는 데이터에 치환 연산이 뒤따르는 변환 연산을 포함하는 수학적 연산들의 복수의 쌍을 적용하는 단계로서, 이러한 변환 및 치환 연산들은 도 4의 디코더(D0)의 메모리(MT_D0)에서 이용 가능한 미리 결정된 복수의 변환 연산($OT_0^{-1}, OT_1^{-1}, \dots, OT_n^{-1}, \dots, OT_{L-1}^{-1}$)(여기서 $0 \leq n \leq L-1$) 및 미리 결정된 복수의 치환 연산($OP_0^{-1}, OP_1^{-1}, \dots, OP_m^{-1}, \dots, OP_{K-1}^{-1}$)(여기서 $0 \leq m \leq K-1$) 중으로부터 각각 선택되는 단계,
- [0333] - 도 6b에 나타내어지는 하위 단계(D621) 중에서, 적용되는 이러한 복수의 쌍으로부터 복수의 블록을 재건하는 단계,
- [0334] - 도 6b에 나타내어지는 하위 단계(D622) 중에서, 복수의 재건된 블록에 적용되는 일관성의 기준을 최대화함으

로써 복수의 재건된 블록 중으로부터 재건된 블록을 선택하는 단계로 구성된다.

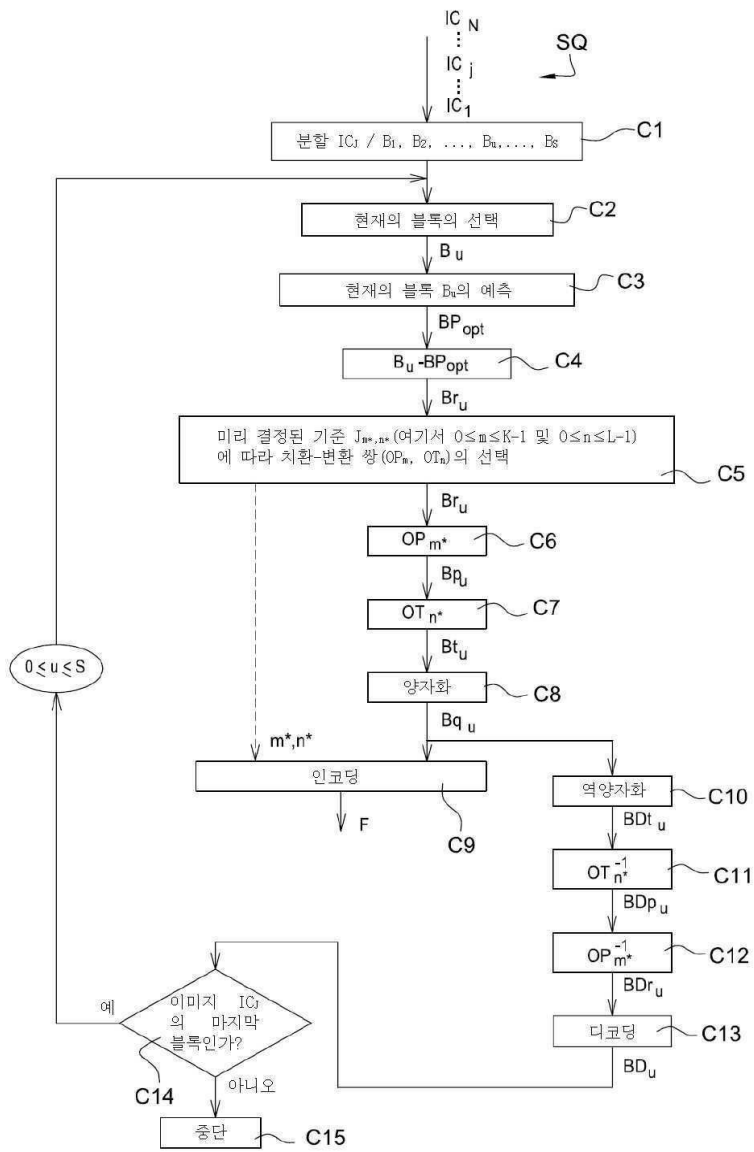
- [0335] 하위 단계(D622)의 끝부분에서, 선택되는 재건된 블록과 연관되는 변환 및 치환 연산들의 쌍이 그 다음 식별된다.
- [0336] 하위 단계들(D620 내지 D622)은 도 4의 디코더(DO)가 그 자체로 인덱스(m^*, n^*)의 전체를 회수할 수 있을 때, 구현된다.
- [0337] 그러한 하위 단계들은 도 4에 나타내어지는 소프트웨어 하위 모듈(SMDT2_DO)에 의해 구현된다.
- [0338] 앞서 언급한 일관성의 기준은 재건된 블록의 인근에 대하여 고려되는 재건된 블록의 픽셀들의 일관성을 측정하는 것으로 구성된다. 일관성 측정은 차이들의 절댓값들의 합을 계산함으로써 재건된 블록의 픽셀들과 재건된 블록의 이웃하는 것 사이의 연속성의 측정일 수 있다.
- [0339] 예로서 주어지고 앞서 언급한 단계(C9)와 관련하여 설명했던 다른 특정 구성에 따르면:
- [0340] - 단일 변환 예를 들어, DST가 인코딩 또는 디코딩에 적용될 것 같고 치환 연산(OP₀)이 DST와의 조합으로 적용되는 경우에, 디코더(DO)가 '0'으로 설정되는 비트를 판독하면, 디코더(DO)는 인코딩에서, 기준(J_{m*,n*})과 연관되었던 치환-변환 쌍(SOP₀)에 이러한 비트의 값을 할당하며,
- [0341] - 단일 변환 예를 들어, DST가 적용될 것 같고 다수의 치환 연산이 DST와의 조합으로 적용될 수 있는 경우에, 디코더(DO)가 '1'로 설정되는 비트를 판독하면, 디코더(DO)는 디코딩에 적용될 것 같은 복수의 가능한 치환-변환 쌍에 이러한 비트의 값을 할당하고, 예를 들어, 일관성 측정들을 사용하여 어느 것이 실제로 시그널링되었던 쌍인지를 그 자체로 추론한다.
- [0342] 복수의 역치환 연산(OP⁻¹₀, OP⁻¹₁, ..., OP⁻¹_m, ..., OP⁻¹_{k-1})은 예들이 인코딩하는 방법의 위의 설명에 인용되었던 복수의 치환 연산(OP₀, OP₁, ..., OP_m, ..., OP_{k-1})의 반대에 상응한다.
- [0343] 복수의 변환 연산(OT⁻¹₀, OT⁻¹₁, ..., OT⁻¹_n, ..., OT⁻¹_{L-1})은 예들이 인코딩하는 방법의 위의 설명에 인용되었던 복수의 변환 연산(OT₀, OT₁, ..., OT_n, ..., OT_{L-1})의 반대에 상응한다. 도 5에 나타내어지는 단계(D7) 중에서, 변환 연산(OT_{n*})의 반대인 변환 연산(OT⁻¹_{n*})은 디코딩된 치환된 블록(BD_{p_u})을 생성하기 위해 역양자화된 계수들(BD_{t_u})의 블록의 데이터에 적용된다.
- [0344] 단계(D7)는 모듈이 처리부(UT_DO)의 마이크로프로세서(μP)에 의해 제어되는 도 4에 나타내어지는 바와 같은 변환 소프트웨어 모듈(MT⁻¹_DO)에 의해 구현된다.
- [0345] 도 5에 나타내어지는 단계(D8) 중에서, 치환 연산(OP_{m*})의 반대인 치환 연산(OP⁻¹_{m*})이 디코딩된 잔여 블록(BD_{r_u})을 생성하기 위해 디코딩된 치환된 블록(BD_{p_u})의 데이터에 적용된다.
- [0346] 단계(D8)는 모듈이 처리부(UT_DO)의 마이크로프로세서(μP)에 의해 제어되는 도 4에 나타내어지는 바와 같은 치환 소프트웨어 모듈(MPER⁻¹_DO)에 의해 구현된다.
- [0347] 도 5에 나타내어지는 단계(D9) 중에서, 현재의 블록(B_i)은 단계(D3)에서 얻어지는 예측 변수 블록(BP_{opt})을 디코딩된 현재의 잔여 블록(BD_{r_u})에 추가함으로써 재건된다.
- [0348] 이러한 단계(D9)는 모듈이 처리부(UT_DO)의 마이크로프로세서(μP)에 의해 제어되는 도 4에 나타내어지는 역예측 소프트웨어 모듈(PRED⁻¹_DO)에 의해 구현된다.
- [0349] 디코딩된 블록(BD_u)은 그 다음 얻어지고 디코딩될 그 다음의 블록에 대한 예측 변수 블록으로서 디코더(DO)에 의해 사용되기 위해 도 4에서의 버퍼 메모리(MT_DO)에 저장된다.
- [0350] 도 5에 나타내어지는 단계(D10) 중에서, 이러한 디코딩된 블록(BD_u)은 현재의 디코딩된 이미지(IDC_j)에 기록된다. 그러한 단계는 도 4에 나타내어지는 바와 같은 이미지 재건 소프트웨어 모듈(URI)에 의해 구현되며,

이러한 모듈은 처리부(UT_DO)의 마이크로프로세서(μ P)에 의해 제어된다.

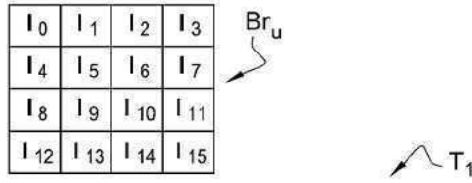
- [0351] 도 5에 나타내어지는 단계(D11) 중에서, 디코더(D0)는 이제 막 디코딩되었던 현재의 블록(BD_u)이 스트림(F)에 포함되는 마지막 블록인지 여부를 검사한다.
- [0352] 그러한 경우라면, 도 5에 나타내어지는 단계(D12) 중에서, 디코딩하는 방법이 종료된다.
- [0353] 그러한 경우가 아니라면, 단계(D2) 중에서, 앞서 언급한 래스터 스캔 스캐닝 순서에 따라 디코딩될 그 다음의 잔여 블록의 선택이 수행된다.
- [0354] 이제 막 상술하였던 디코딩 단계들이 고려되는 현재의 이미지(IC_j)의 디코딩될 모든 블록들($B_1, B_2, \dots, B_u, \dots, B_s$)에 대해 구현된다.
- [0355] 상술하였던 실시예들이 단순히 지시적인 것으로 주어졌고 결코 제한적이지 않고, 많은 변경들이 그러나, 본 발명의 범위로부터 벗어나지 않는 범위 내에서 당업자에 의해 용이하게 행해질 수 있다는 것은 말할 필요도 없다.
- [0356] 따라서, 크기 4x4 또는 8x8의 현재의 블록들에 적용되는 것 대신에, 본 발명에 따른 인코딩하고/하거나 디코딩하는 방법은 예를 들어, 16x16 및 32x32와 같은 상이한 가능한 크기들을 갖는 현재의 블록들에 적용될 수도 있다.

도면

도면1



도면3a



픽셀 인덱스	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	
0	0	3	14	8	0	0	15	0	12	0	4	11	3	3	3	12	14	3	14	13	4	3	3	4	14	12	12	12	
1	1	2	13	9	4	4	14	1	8	3	5	10	2	7	0	13	15	2	12	14	1	2	1	5	10	13	15	13	
2	2	1	12	10	8	6	13	2	4	2	6	9	1	11	1	14	12	1	13	15	2	1	2	6	6	14	14	14	
3	3	0	15	11	12	3	12	3	0	1	3	8	0	15	2	15	13	0	15	12	0	12	7	7	2	15	13	15	
4	4	7	10	4	1	1	11	4	13	4	8	15	7	5	7	8	10	4	10	9	8	15	0	8	15	8	8	8	
5	5	6	9	5	5	2	10	5	9	7	9	14	11	9	4	4	11	7	8	10	5	14	5	9	11	1	11	9	
6	6	5	8	6	9	9	9	6	5	6	10	13	15	10	5	0	8	6	9	11	6	13	6	10	7	2	10	10	
7	7	4	11	7	13	7	8	7	1	5	7	12	4	14	6	11	9	5	11	8	3	8	11	11	3	11	9	11	
8	8	11	6	0	2	5	7	8	14	8	12	3	6	4	11	9	6	9	6	4	12	11	4	12	13	4	4	4	
9	9	10	5	1	6	6	6	9	10	11	13	2	10	8	8	5	7	8	4	5	9	10	9	13	9	9	7	5	
10	10	9	4	2	10	10	5	10	6	10	14	1	14	6	9	1	4	11	5	6	10	9	10	14	5	10	6	6	
11	11	8	7	3	14	11	4	11	2	9	11	0	8	13	10	7	5	10	7	7	7	4	15	15	1	7	5	7	
12	12	15	2	12	3	12	3	12	15	12	0	7	5	0	12	10	2	14	2	0	15	7	8	0	12	0	0	0	
13	13	14	1	13	7	13	2	13	11	15	1	7	9	1	13	6	3	13	0	1	13	6	13	1	8	5	3	1	
14	14	13	0	14	11	14	1	14	7	14	2	5	13	2	14	2	0	12	1	2	14	5	14	2	4	6	2	2	
15	15	12	3	15	15	15	0	15	3	13	15	4	12	12	15	3	1	15	3	3	11	0	12	3	0	3	1	3	
타입	dct	dst	dct	dct	dct	dst	dst	dst	dct	dct	dct	dst	dst	dst	dst	dst	dst	dst	dct	dct	dct	dct	dct	dct	dct	dct	cct	dct	dst

도면3b

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

Br_u

T₂

반영 인덱스	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28
0	0	7	0	7	0	0	0	3	7	0	1	1	6	7	8	24	48	48	59	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	1	9	1	4	0	57	2	2	7	6	9	25	49	49	58	1	1	1	1	1	1	1	1	3	9
2	2	1	2	1	2	18	2	5	1	50	3	3	2	5	10	26	50	50	57	2	2	2	2	2	4	5	7	2	10
3	3	2	3	2	3	27	3	6	2	43	4	4	3	3	11	27	51	51	56	3	3	3	3	3	5	6	3	1	11
4	4	3	4	3	4	36	4	7	3	36	5	5	4	4	12	28	52	52	0	4	4	4	4	4	6	7	4	4	12
5	5	4	5	4	5	45	5	8	4	29	6	6	5	13	13	29	53	53	1	5	5	5	5	5	7	2	5	5	13
6	6	5	6	5	6	54	6	9	5	22	7	7	0	14	14	30	54	54	2	6	6	6	6	6	2	3	6	6	14
7	7	6	7	6	7	63	7	10	6	15	0	0	1	15	15	31	55	55	3	7	7	7	7	7	3	4	2	7	15
8	8	15	40	15	24	8	24	10	15	8	9	9	14	0	16	6	40	56	50	8	8	9	15	40	8	8	8	8	8
9	9	8	41	8	25	17	25	11	8	1	10	10	15	1	17	9	41	57	49	9	9	10	8	41	9	9	9	11	17
10	10	9	42	9	26	26	26	12	9	58	11	11	10	2	18	10	42	58	48	10	10	11	9	42	12	13	15	10	18
11	11	10	43	10	27	35	27	13	10	51	12	12	11	11	19	11	43	59	8	11	11	12	10	43	13	14	11	9	19
12	12	11	44	11	28	44	28	14	11	44	13	13	12	12	20	12	44	60	9	12	12	13	11	44	14	15	12	12	20
13	13	12	45	12	29	53	29	15	12	37	14	14	13	21	21	13	45	61	10	13	13	14	12	45	15	10	13	13	21
14	14	13	46	13	30	62	30	16	13	30	15	15	8	22	22	14	46	62	11	14	14	15	13	46	10	11	14	14	22
15	15	14	47	14	31	7	31	17	14	23	16	16	9	23	23	15	47	63	12	15	15	8	14	47	11	12	10	15	23
16	16	23	48	23	32	16	32	17	15	16	17	17	22	8	24	16	0	40	41	18	16	18	22	32	16	16	16	16	16
17	17	16	49	16	33	25	33	18	16	9	18	18	23	9	25	17	1	41	40	19	17	19	23	33	17	17	17	19	25
18	18	17	50	17	34	34	34	19	17	2	19	19	18	10	26	18	2	42	16	20	18	20	16	34	20	21	23	18	26
19	19	18	51	18	35	43	35	20	18	59	20	20	19	19	27	19	3	43	17	21	19	21	17	35	21	22	19	17	27
20	20	19	52	19	36	52	36	21	19	52	21	21	20	20	28	20	4	44	18	22	20	22	16	36	22	23	20	20	28
21	21	20	53	20	37	61	37	22	20	45	22	22	21	29	29	21	5	45	19	23	21	23	19	37	23	18	21	21	29
22	22	21	54	21	38	6	38	23	21	38	23	23	16	30	30	22	6	46	20	16	22	16	20	38	16	19	22	22	30
23	23	22	55	22	39	15	39	24	22	31	24	24	17	31	31	23	7	47	21	17	23	17	21	39	19	20	18	23	31
24	24	24	31	8	31	40	24	40	25	23	24	25	25	30	16	32	0	8	0	32	26	48	27	26	8	24	24	24	24
25	25	24	9	24	41	33	41	26	24	17	26	26	31	17	33	1	9	1	24	27	49	26	30	9	25	25	25	27	33
26	26	25	10	25	42	42	42	27	25	10	27	27	26	18	34	2	10	2	25	28	50	29	31	10	26	29	31	26	34
27	27	26	11	26	43	51	43	28	26	3	28	28	27	27	35	3	11	3	26	29	51	30	24	11	29	30	27	25	35
28	28	27	12	27	44	60	44	29	27	60	29	29	28	28	36	4	12	4	27	30	52	31	25	12	30	31	28	28	36
29	29	28	13	28	45	5	45	30	28	53	30	30	29	37	37	5	13	5	28	31	53	24	26	13	31	26	29	29	37
30	30	29	14	29	46	14	46	31	29	46	31	31	24	38	38	6	14	6	29	24	54	25	27	14	26	27	30	30	38
31	31	30	15	30	47	23	47	39	30	39	24	24	25	39	39	7	15	7	30	25	55	26	28	15	27	28	26	31	39
32	32	39	16	39	48	32	48	24	39	32	33	33	38	24	0	32	16	8	33	34	56	36	36	16	32	32	32	32	40
33	33	32	17	32	49	41	49	32	32	25	34	34	39	25	1	33	17	9	34	35	57	37	37	17	33	33	33	33	41
34	34	33	18	33	50	50	50	33	33	28	35	35	34	26	2	34	18	10	35	36	58	38	38	18	36	37	39	34	42
35	35	34	19	34	51	59	51	34	34	11	36	36	35	35	3	35	19	11	36	37	59	39	39	19	37	38	35	35	43
36	36	35	20	35	52	4	52	35	35	4	37	37	36	36	4	36	20	12	37	38	60	32	32	20	38	39	36	36	44
37	37	36	21	36	53	13	53	36	36	61	38	38	37	45	5	37	21	13	38	39	61	33	33	21	39	34	37	37	45
38	38	37	22	37	54	22	54	37	37	54	39	39	32	46	6	38	22	14	39	32	62	34	34	22	34	35	38	38	46
39	39	38	23	38	55	31	55	38	38	47	32	32	33	47	7	39	23	15	31	33	63	35	35	23	35	36	34	39	47

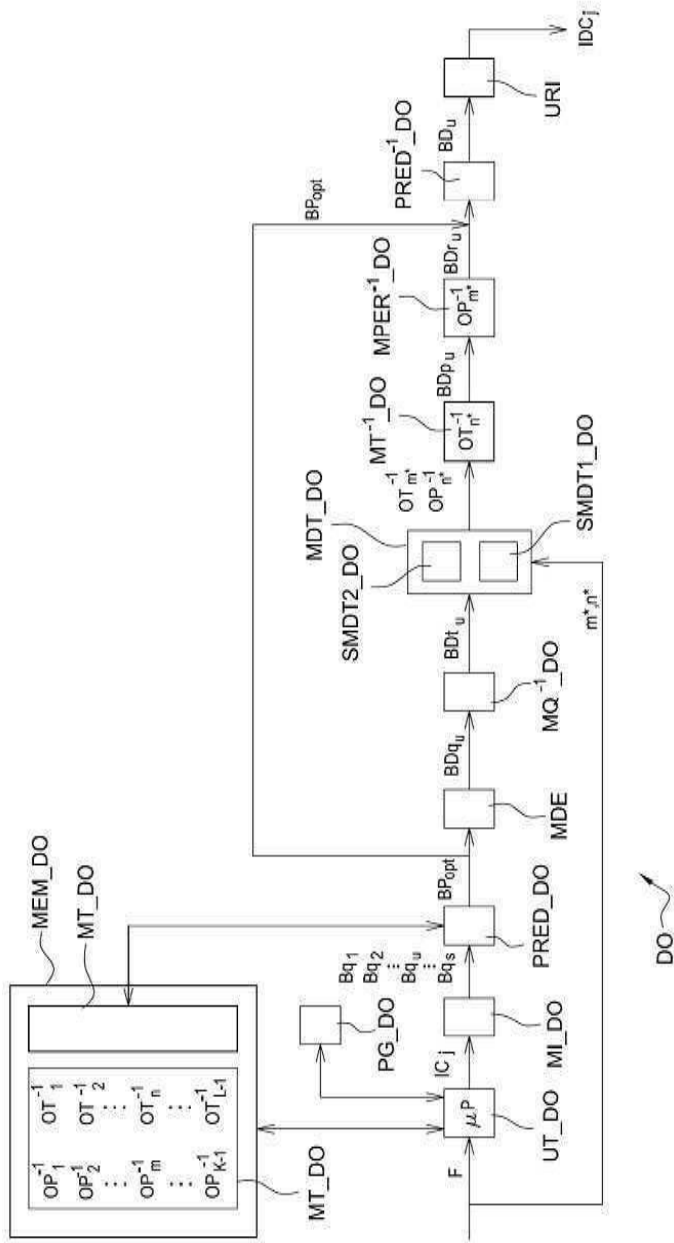
:

도면3c

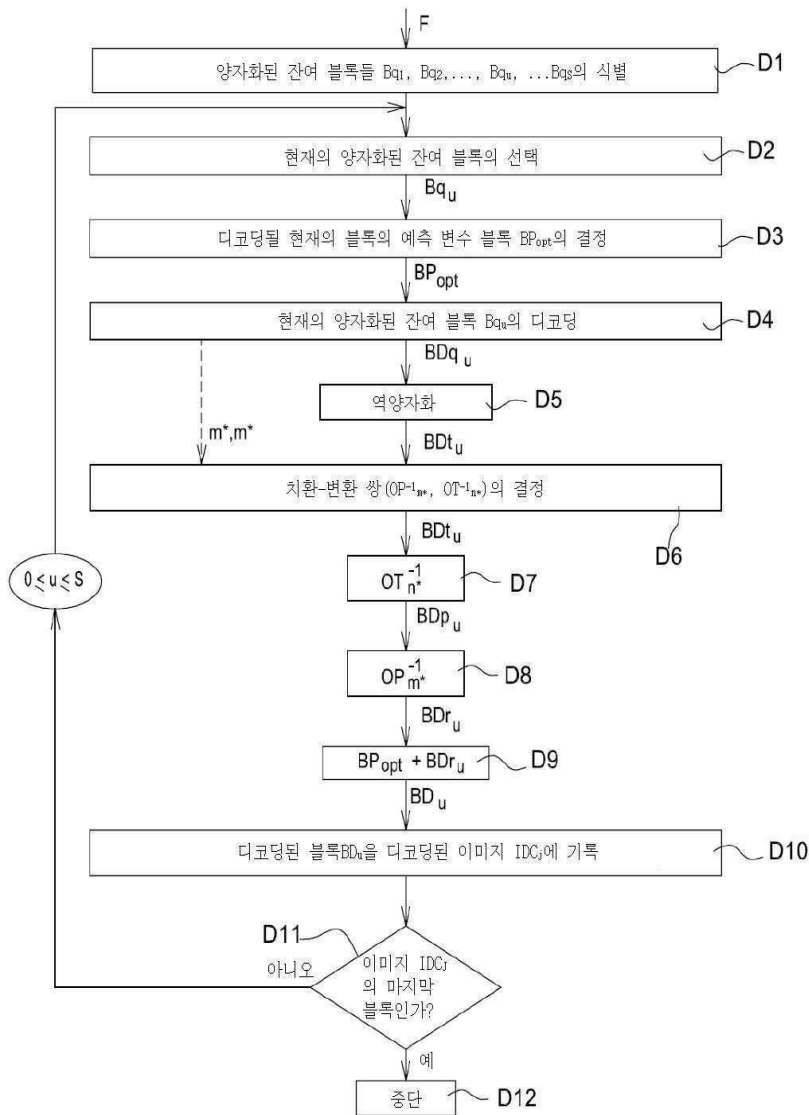
⋮  T₂

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

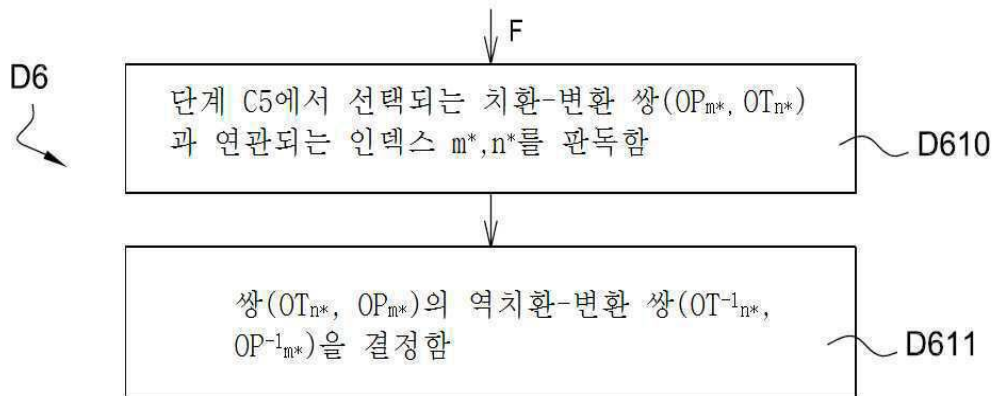
도면4



도면5



도면6a



도면6b

