



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0035681
(43) 공개일자 2019년04월03일

- | | |
|---|---|
| <p>(51) 국제특허분류(Int. Cl.)
 <i>H04N 19/13</i> (2014.01) <i>H04N 19/12</i> (2014.01)
 <i>H04N 19/132</i> (2014.01) <i>H04N 19/147</i> (2014.01)
 <i>H04N 19/176</i> (2014.01) <i>H04N 19/18</i> (2014.01)
 <i>H04N 19/192</i> (2014.01) <i>H04N 19/463</i> (2014.01)</p> <p>(52) CPC특허분류
 <i>H04N 19/13</i> (2015.01)
 <i>H04N 19/12</i> (2015.01)</p> <p>(21) 출원번호 10-2019-7000944
 (22) 출원일자(국제) 2017년07월12일
 심사청구일자 없음
 (85) 번역문제출일자 2018년01월10일
 (86) 국제출원번호 PCT/EP2017/067616
 (87) 국제공개번호 WO 2018/011303
 국제공개일자 2018년01월18일
 (30) 우선권주장
 16305919.9 2016년07월15일
 유럽특허청(EPO)(EP)</p> | <p>(71) 출원인
 인터디지탈 브이씨 홀딩스 인코포레이티드
 미국 19809 델라웨어주 월밍턴 스위트 300 벨뷰
 파크웨이 200</p> <p>(72) 발명자
 라세르, 세바스티앙
 프랑스 35576 쉼송 쉼비네 쉼에스 17616 아브뉴
 데 상 블랑 975 떼끄니폴로르 에르 에 데 프랑스
 내
 푸리, 사우라브
 프랑스 35576 쉼송 쉼비네 쉼에스 17616 아브뉴
 데 상 블랑 975 떼끄니폴로르 에르 에 데 프랑스
 내
 르 칼레, 패트릭
 프랑스 35576 쉼송 쉼비네 쉼에스 17616 아브뉴
 데 상 블랑 975 떼끄니폴로르 에르 에 데 프랑스
 내</p> <p>(74) 대리인
 양영준, 김연송, 백만기</p> |
|---|---|

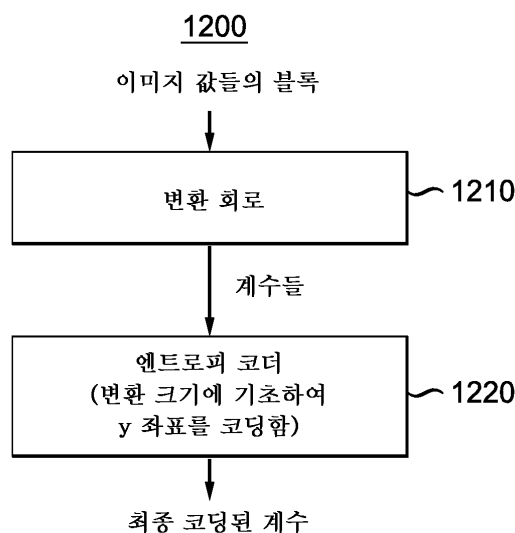
전체 청구항 수 : 총 15 항

(54) 발명의 명칭 적응형 변환 기반 비디오 압축을 위한 최종 계수 코딩

(57) 요약

최종 코딩된 계수 위치의 코딩은, 최종 코딩된 계수의 위치의 y 좌표의 코딩을 비디오 픽셀들의 블록으로부터 계수들의 블록을 생성하기 위해 사용되는 부분적 변환의 크기의 지식에 기초함으로써 수행된다. 이는 최종 코딩된 계수 파라미터의 컨텍스트 적응형 코딩을 훨씬 더 효율적으로 수행되게 할 수 있다.

대표도 - 도12



(52) CPC특허분류

H04N 19/132 (2015.01)

H04N 19/147 (2015.01)

H04N 19/176 (2015.01)

H04N 19/18 (2015.01)

H04N 19/192 (2015.01)

H04N 19/463 (2015.01)

명세서

청구범위

청구항 1

변환된 계수들의 블록을 코딩하기 위한 방법으로서,

변환된 계수들의 서브셋이 제로로 될 것이 알려지도록, 상기 변환된 계수들을 획득하기 위해 부분적 변환을 사용하여 이미지 값들의 블록을 변환하는 단계;

상기 변환된 계수들의 스캔에 의해 결정되는 최종 n -제로 변환된 계수의 상기 블록에서의 위치를 정의하는 2개의 좌표를 엔트로피 코딩하는 단계

를 포함하고,

상기 2개의 좌표 중 하나의 좌표(y)의 엔트로피 코딩은 상기 부분적 변환의 크기의 지식에 의존하는, 방법.

청구항 2

제1항에 있어서,

상기 부분적 변환의 크기의 지식은 비트스트림에서 인코딩되는, 방법.

청구항 3

제1항 또는 제2항에 있어서,

상기 2개의 좌표 각각을 프리픽스 및 서픽스로 분리하는 단계; 및

상기 y 좌표의 프리픽스의 값을 상기 부분적 변환의 크기의 지식으로부터 추론하는 단계를 더 포함하는, 방법.

청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 이미지 값들의 블록을 변환하기 위해, 오프라인으로 학습된 부분적 변환들의 적응형 세트가 사용될 수 있는, 방법.

청구항 5

변환된 계수들의 세트를 코딩하기 위한 장치로서,

변환된 계수들을 획득하기 위해 부분적 변환을 사용하여 이미지 값들의 블록에 대해 동작하는 변환 회로;

상기 변환된 계수들의 스캔에 의해 결정되는 상기 변환된 계수들로부터 최종 코딩된 계수의 위치를 표현하는 2개의 좌표를 생성하는 엔트로피 코더

를 포함하고,

상기 최종 코딩된 계수의 위치는 2개의 좌표(x, y)에 의해 제공되고, y 좌표의 상기 엔트로피 코딩은 상기 부분적 변환의 크기의 지식에 의존하는 것을 특징으로 하는, 장치.

청구항 6

제5항에 있어서,

상기 부분적 변환의 크기의 지식은 비트스트림에서 인코딩되는, 장치.

청구항 7

제5항 또는 제6항에 있어서,

상기 엔트로피 코더는 상기 2개의 좌표 각각을 프리픽스 및 서픽스로 분리하고; 상기 y 좌표의 프리픽스의 값을 상기 x 좌표로부터 추론하는, 장치.

청구항 8

변환된 계수들의 세트를 디코딩하기 위한 방법으로서,

최종 코딩된 계수의 위치를 생성하기 위해 코드를 엔트로피 디코딩하는 단계 - 상기 최종 코딩된 계수의 위치는 2개의 좌표(x,y)에 의해 제공되고, y 좌표의 상기 엔트로피 디코딩은 부분적 변환의 크기의 지식에 의존하는 것을 특징으로 함 -;

이미지 값들의 블록을 획득하기 위해 부분적 변환을 사용하여, 변환된 계수들을 역변환하는 단계를 포함하는, 방법.

청구항 9

제8항에 있어서,

상기 부분적 변환의 크기의 지식은 비트스트림으로부터 디코딩되는, 방법.

청구항 10

제8항 또는 제9항에 있어서,

상기 2개의 좌표 각각을 프리픽스 및 서픽스로 분리하는 단계; 및

상기 y 좌표의 프리픽스의 값을 상기 부분적 변환의 크기의 지식으로부터 추론하는 단계를 더 포함하는, 방법.

청구항 11

변환된 계수들의 세트를 디코딩하기 위한 장치로서,

상기 변환된 계수들로부터 최종 코딩된 계수의 위치를 표현하는 코드에 대해 동작하는 엔트로피 디코더 - 상기 최종 코딩된 계수의 위치는 2개의 좌표(x,y)에 의해 제공되고, y 좌표의 상기 엔트로피 디코딩은 부분적 변환의 크기의 지식에 의존하는 것을 특징으로 함 -;

이미지 값들의 블록을 획득하기 위해 부분적 변환을 사용하여, 변환된 계수들에 대해 동작하는 역변환 회로를 포함하는, 장치.

청구항 12

제11항에 있어서,

상기 부분적 변환의 크기의 지식은 비트스트림으로부터 디코딩되는, 장치.

청구항 13

제11항 또는 제12항에 있어서,

상기 엔트로피 디코더는 상기 2개의 좌표 각각을 프리픽스 및 서픽스로 분리하고; 상기 y 좌표의 프리픽스의 값을 상기 부분적 변환의 크기의 지식으로부터 추론하는, 장치.

청구항 14

변환된 계수들의 세트를 제8항 내지 제10항 중 어느 한 항의 방법에 따라 디코딩하기 위한 또는 변환된 계수들의 블록을 제1항 내지 제4항 중 어느 한 항에 따라 코딩하기 위한 명령어들이 저장된 비일시적 컴퓨터 판독가능 저장 매체.

청구항 15

제8항 내지 제10항 중 어느 한 항의 방법에 따른 비트스트림이 저장된 비일시적 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 원리들은 일반적으로 비디오 압축 및 압축해제 시스템들에 관한 것으로, 보다 상세하게는 블록 기반 변환들에 관한 것이다.

배경 기술

[0002] 변환 T는 n개의 픽셀들의 블록을 n개의 변환된 계수들로 변환한다. 프로세스는 역변환(T^{-1})을 변환된 계수들에 적용하여 픽셀 값들을 다시 얻음으로써 가역적이다. 부분적 변환 P의 경우, n개의 픽셀들은 n 미만의 변환 계수들 m개로 변환되며, 여기서 $m < n$ 이다. 동등하게, 누락된 m-n개의 계수들은 제로로 설정되는 것으로 가정될 수 있다. 대략적 "역" 변환 P'는 초기 픽셀 값들의 근사화를 획득하기 위해 변환된 계수들에 적용된다. P'는 근사화이고, 부분적 변환이 비가역적이기 때문에 정확한 수학적 역이 아니다. 통상적으로, 계수들의 부분적으로 변환된 블록은 픽셀 블록의 저주파수 정보를 표현한다.

[0003] 일부 비디오 코딩 표준들에서, 변환된 계수들은 통상적으로 고주파수부터 저주파수로의 스캔 순서와 연관된 2차원(2D) 블록에 입력된다. 도 1에 도시된 바와 같이, 계수들의 양자화 이후, 고주파수로부터 저주파수로의 스캔 순서에 대한 제1 년-제로 양자화된 계수는 최종 코딩된 계수로 지칭된다.

[0004] 계수들과 연관된 2D 블록 토폴로지에서, 최종 코딩된 계수는 크기 $n = N \times N$ 의 블록에 대해 도 2에 도시된 바와 같이 2개의 좌표(x,y)를 갖는다.

[0005] 정의에 따라, $0 \leq x, y \leq N-1$ 이다. 이러한 2개의 좌표는 디코더 측에서 년-제로 코딩된 계수들을 결정하기 위한 정보로서 비트스트림으로 코딩된다. 적어도, 이들 좌표는, 좌표(x,y)의 계수가 제로가 아니며, 저주파수로부터 고주파수로의 스캔 순서에 대해 이러한 계수 이후에 오는 계수들은 모두 제로임을 표시한다.

[0006] 변환된 계수들의 중요성을 표시하는 추가적인 정보는 나머지 계수들(도 1에 물음표로 마킹됨)이 제로인지의 여부를 시그널링하기 위해 추가될 수 있다.

[0007] 종래에, 지난 수십 년 동안 개발된 많은 비디오 및 이미지 코덱들에서, 변환된 계수들을 획득하기 위해 예를 들어 이산 코사인 변환(Discrete Cosine Transform) 또는 이산 사인 변환(Discrete Sine Transform)과 같은 고정된 변환들이 각각의 블록의 픽셀들에 적용된다. 그 다음, 이들 계수들은 예를 들어, VLC, 산술 코더 또는 컨텍스트 적응형 이진 산술 코딩(CABAC)과 같은 엔트로피 코더에 의해 인코딩되는 양자화된 계수들을 획득하기 위해 양자화기 Q에 의해 양자화된다.

[0008] 최종 코딩된 계수 위치를 가능한 한 효율적으로 인코딩하는 것이 바람직하다.

[0009] 고정된 변환에 대한 대안으로, 콘텐츠-적응형 부분적 변환들이 사용될 수 있다. 이는 에너지의 더 양호한 압축을 유도하고 따라서 비트 레이트를 감소시킨다. 그러나, 적응형 변환들은 디코더에 전송되어, 리턴에서 비트 레이트를 증가시킬 것이다. 이러한 변환들을 전송할 때 주의를 기울이지 않으면, 글로벌 비트 레이트가 증가되어, 더 양호한 압축에도 불구하고, 적응형 변환들은 유리하지 않다.

[0010] HEVC/H.265 표준은 2개의 좌표(x,y)를 사용하는 최종 코딩된 계수 위치의 코딩을 소개하였다. 각각의 좌표는 절단된 단항 코드를 사용하여 이진화되고, 그 다음, HEVC 용어의 각각의 비트 또는 "빈(bin)"은 컨텍스트들에 기초한 채널 적응에 의한 CABAC를 사용하여 인코딩된다. HEVC에서 2개의 좌표 x 및 y는 별도로 인코딩 및 디코딩된다.

발명의 내용

[0011] 종래 기술의 이들 및 다른 결점들 및 단점들은, 부분적 변환의 크기의 지식을 사용하여 부분적 변환의 사용을 통해 획득되는 최종 코딩된 계수를 코딩하기 위한 방법 및 장치에 관한 본 원리들에 의해 처리된다.

[0012] 본 원리들의 일 양태에 따르면, 변환된 계수들의 세트를 코딩하기 위한 방법이 제공되며, 방법은, 변환된 계수들을 획득하기 위해 이미지 값들의 블록을 변환하는 단계 및 최종 코딩된 계수의 위치를 엔트로피 코딩하는 단계의 세트를 포함하고, 최종 코딩된 계수의 위치는 y 좌표의 엔트로피 코딩이 계수들을 획득하기 위해 사용되는 부분적 변환의 크기의 지식에 의존하도록 2개의 좌표에 의해 제공된다.

- [0013] 본 원리들의 다른 양태에 따르면, 변환된 계수들의 세트를 코딩하기 위한 장치는, 변환된 계수들을 획득하기 위해 이미지 값들의 블록에 대해 동작하는 변환 회로 및 엔트로피 코더를 포함하며, 변환된 계수들의 최종 코딩된 계수의 위치는 y 좌표의 엔트로피 코딩이 계수들을 획득하기 위해 사용되는 부분적 변환의 크기의 지식에 의존하도록 2개의 좌표에 의해 제공된다.
- [0014] 본 원리들의 다른 양태에 따르면, 변환된 계수들의 세트를 디코딩하기 위한 방법이 제공된다. 방법은, 최종 코딩된 계수의 위치를 생성하기 위해 변환된 계수들을 엔트로피 디코딩하는 단계 - 최종 코딩된 계수의 위치는 2개의 좌표(x, y)에 의해 제공되고, y 좌표의 상기 엔트로피 디코딩은 계수들을 획득하기 위해 사용되는 부분적 변환의 크기의 지식에 의존하는 것을 특징으로 함 -; 및 이미지 값들의 블록을 획득하기 위해 변환된 계수들을 역변환하는 단계를 포함한다.
- [0015] 본 원리들의 다른 양태에 따르면, 변환된 계수들의 세트를 디코딩하기 위한 장치가 제공된다. 장치는, 변환된 계수들로부터 최종 코딩된 계수의 위치를 표현하는 코드에 대해 동작하는 엔트로피 디코더를 포함하고, 최종 코딩된 계수의 위치는 2개의 좌표(x, y)에 의해 제공되고, y 좌표의 엔트로피 코딩은 계수들을 획득하기 위해 사용되는 부분적 변환의 크기의 지식에 의존하는 것을 특징으로 한다. 장치는 이미지 값들의 블록을 획득하기 위해 변환된 계수들에 대해 동작하는 역변환 회로를 더 포함한다.
- [0016] 본 원리들의 다른 양태에 따르면, 변환된 계수들의 세트를 디코딩하기 위한 명령어들이 저장된 비일시적 컴퓨터 판독가능 저장 매체가 제공되며, 최종 코딩된 계수의 위치는 2개의 좌표(x, y)에 의해 제공되고, y 좌표의 엔트로피 코딩은 계수들을 획득하기 위해 사용되는 부분적 변환의 크기의 지식에 의존하는 것을 특징으로 한다.
- [0017] 본 원리들의 다른 양태에 따르면, 변환된 계수들의 세트를 디코딩하기 위한 비트스트림이 저장된 비일시적 컴퓨터 판독가능 저장 매체가 제공되며, 최종 코딩된 계수의 위치는 2개의 좌표(x, y)에 의해 제공되고, y 좌표의 엔트로피 코딩은 계수들을 획득하기 위해 사용되는 부분적 변환의 크기의 지식에 의존하는 것을 특징으로 한다.
- [0018] 본 원리들의 이들 및 다른 양태들, 특징들 및 이점들은 첨부된 도면들과 관련하여 읽혀질 예시적인 실시예들에 대한 다음의 상세한 설명으로부터 명백해질 것이다.

도면의 간단한 설명

- [0019] 도 1은 변환 유닛의 스캔 순서를 도시한다.
- 도 2는 최종 코딩된 계수에 대한 좌표의 예를 도시한다.
- 도 3은 12개의 계수들을 갖는 4×4 변환 유닛에서 코딩의 예를 도시한다.
- 도 4는 좌표 값에 대한 서픽스(suffix) 및 프리픽스(prefix) 결정을 도시한다.
- 도 5는 온-더-플라이(on-the-fly) 학습 방식을 도시한다.
- 도 6은 다양한 장면들에 대한 BD-레이트 이득 대 변환 벡터들의 수를 도시한다.
- 도 7은 부분적 변환에 의한 8×8 블록에 대한 코딩을 도시한다.
- 도 8은 콘텍스트 값의 구조를 도시한다.
- 도 9는 콘텍스트 값의 진화를 도시한다.
- 도 10은 이웃 채널들에 따른 콘텍스트 선택의 예를 도시한다.
- 도 11은 본 원리들을 사용하여 변환된 계수들의 세트를 코딩하기 위한 방법의 일 실시예를 도시한다.
- 도 12는 본 원리들을 사용하여 변환된 계수들의 세트를 코딩하기 위한 장치의 일 실시예를 도시한다.
- 도 13은 본 원리들을 사용하여 변환된 계수들의 세트를 디코딩하기 위한 방법의 일 실시예를 도시한다.
- 도 14는 본 원리들을 사용하여 변환된 계수들의 세트를 디코딩하기 위한 장치의 일 실시예를 도시한다.

발명을 실시하기 위한 구체적인 내용

- [0020] 다음의 실시예들에 의해 해결되는 기술적인 문제는 2D 부분적 변환이 적용된 픽셀들의 변환된 블록에서의 최종 코딩된 계수의 위치를 코딩하는 비용을 감소시키는 것이다.

- [0021] 이러한 실시예들은 비디오 코딩 표준들에서 통상적으로 사용되는 엔트로피 코딩 방식의 개선이다. 하나의 이러한 비디오 코딩 표준은 HEVC/H.265 표준이지만, 실시예들은 그 표준에 제한되지 않는다.
- [0022] 본 원리들의 주요 아이디어는, 부분적 변환이 픽셀 값들의 블록에 대해 사용되면, 일부 변환된 계수들은 필수적으로 제로이며, 따라서 최종 코딩된 계수 및 최종 코딩된 계수의 위치를 표현하는 좌표(x,y)에 대한 일부 제약들을 부과하는 것으로 공지되어 있다는 점이다. 바람직한 실시예들에서, 초점은, 픽셀들의 블록을 변환 계수들로 변환하기 위해 부분적 변환이 사용되는 경우, 2개의 좌표(x,y)를 효율적으로 인코딩하기 위해 이러한 제약들을 사용하는 것이다.
- [0023] 본 아이디어들은, (x,y)의 코딩이 더 효율적이도록 $m < n$ 인 경우 모든 좌표 쌍들(x,y)이 허용가능하지는 않다는 사실을 사용하는 것이다. 이는 개선된 압축 성능을 유도할 수 있다.
- [0024] 특히, 부분적 변환들을 사용하는 경우, 모든 좌표 쌍들(x,y)이 허용가능하지는 않다는 사실을 명시적으로 고려하여, (x,y)의 코딩이 더 효율적이다. 부분적 변환들에 의한 이러한 특정 실시예에서, 부분적 변환들을 사용하여 수정된 HEVC 표준에 대해 테스트들이 수행되었고, (x,y)의 개선된 코딩은 약 -0.5%의 압축 이득을 유도하는 것으로 나타났다.
- [0025] 본 명세서에 제시된 아이디어들은 좌표 쌍(x,y)을 코딩하는 HEVC 알고리즘의 주요 부분을 유지하지만, 특징, 즉, x가 공지되면 y를 코딩하는 가능한 범위의 감소를 추가한다.
- [0026] 예를 들어, 도 3에서와 같이, $m = 12$ 개의 변환된 계수들을 유도하는 부분적 변환 P를 갖는 4×4 블록을 고려한다. 결과적으로, 계수들 0 내지 11은 널-제로일 수 있지만 계수들 12 내지 15는 반드시 제로임을 안다.
- [0027] 최종 코딩된 계수에 대한 좌표 x는 범위 0(위치들 0, 1, 3, 6에 대한 것) 내지 3(오직 위치 9에 대한 것) 내에 있다. x가 공지되면, y의 범위는 실제로 x에 의존한다. 상기 예에서,
- [0028] $x=0$ 또는 $x=1$ 은 범위 [0,3] 내에 있는 y를 유도한다.
- [0029] $x=2$ 는 범위 [0,2] 내에 있는 y를 유도한다.
- [0030] $x=3$ 은 $y=0$ 을 유도하고; 이러한 경우, y는 심지어 코딩될 필요가 없다.
- [0031] 이러한 코딩 프로세스는 x를 디코딩할 때 디코딩가능하며, 그 다음, m이 공지되고 어디에선가 인코딩된다고 가정하면 y에 대한 범위를 연역하고, 그 다음 y를 디코딩하는 것을 쉽게 이해한다.
- [0032] HEVC에서, 좌표 x 또는 y는 도 4에 도시된 바와 같이 프리픽스 및 서픽스로 분리된다.
- [0033] 예를 들어, 좌표 값이 14인 경우 프리픽스는 7이고 서픽스는 2 비트로 코딩된다. 서픽스는 제1 값이 좌표 값으로부터 감산된 후 나머지, 즉 $14-12 = 2$ 이다. 좌표 값이 3이면 프리픽스는 3이지만 서픽스는 없다.
- [0034] 프리픽스는 절단된 단항 코딩을 사용하여 이진화된다. 절단은 프리픽스 값에 대한 상한을 제공하는 블록 크기에 대한 지식에 기초하여 수행된다. 이진화된 프리픽스의 각각의 비트는 CABAC 및 전용 컨텍스트를 사용하여 인코딩된다.
- [0035] 예를 들어, 4×4 블록에서 가능한 프리픽스는, 1, 01, 001 및 000(절단됨)으로 각각 이진화되는 0, 1, 2 또는 3이다. 다른 예에서, 8×8 블록에서 가능한 프리픽스는 0 내지 5이고, 1, 01, 001, 0001, 00001 및 00000(절단됨)으로 이진화된다.
- [0036] 서픽스는 고정 길이 코딩을 사용하여 이진화되며, 고정 길이 서픽스는 컨텍스트들을 사용하지 않고 우회 모드에서 CABAC를 사용하여 그대로 인코딩된다. 예를 들어, 16×16 블록에서 좌표 값이 14인 경우, 프리픽스는 7(0000000으로 이진화됨)이고 서픽스는 2(10으로 이진화되고 2비트로 코딩됨)이다.
- [0037] 프리픽스 및 서픽스로의 이진화 프로세스는 오직 프리픽스에 대해서만 컨텍스트들을 사용하여 HEVC에서와 같이 유지된다. 본 원리들의 주요 특징은 y의 컨텍스트들이 x의 값에 따라 범위에서 감소될 수 있다는 것이다.
- [0038] 통상적으로, 변환 T는 n개의 픽셀들의 블록을 $m=n$ 개의 변환된 계수들로 변환한다. 프로세스는 역변환(T^{-1})을 변환된 계수들에 적용하여 픽셀 값들을 다시 얻음으로써 가역적이다. 부분적 변환 P의 경우, n개의 픽셀들은 $m < n$ 개의 변환된 계수들로 변환된다. 동등하게, 부족한 $m-n$ 개의 계수들은 제로로 설정되는 것으로 가정될 수 있다. 초기 픽셀 값들의 근사치를 획득하기 위해 "역"변환 P'(물론 부분적 변환이 비가역적이기 때문에 수학적 역은 아님)가 변환된 계수에 적용된다. 통상적으로, 부분적으로 변환된 계수들은 픽셀 블록의 저주파수 정보를 표현

한다.

- [0039] 부분적 변환이 사용되는 경우, 일부 $m \times n$ 개의 변환된 계수들은 반드시 제로이므로, 최종 코딩된 계수의 위치 및 좌표(x,y)에 일부 제약들을 부과하는 것으로 공지되어 있다. 이러한 경우, 이러한 제약들을 명시적으로 사용함으로써, 실시예는 TU 픽셀들을 변환된 계수들로 변환하기 위해 부분적 변환이 사용되는 경우, 2개의 좌표(x,y)의 효율적인 인코딩을 자동으로 처리하는 것으로 나타났다.
- [0040] DCT 또는 DST와 같은 시스터매틱 변환들 대신, 상이한 분류 및 변환 최적화 방식들을 사용하는 대규모 트레이닝 세트에 대해 오프라인으로 학습되는 직교 변환들의 적응형 세트가 대신 사용될 수 있다. 변환들의 이러한 세트는 코덱에 공급되고 세트 중 최상의 변환은 레이트 왜곡 최적화(RDO) 루프에서 선택된다. 보다 적응적인 접근법은 시퀀스의 특정 인트라 프레임에 대한 직교 변환들의 세트를 학습하는 것이다. 이는 본 명세서의 나머지 부분에서 온-더-플라이 블록 기반 변환 학습 방식으로 지칭된다. 이러한 방식은 도 5에 도시되어 있다.
- [0041] 도 5의 블록도는, 알고리즘이 2개의 부분들, 즉, 잔차 블록들의 비디오/이미지 코덱으로의 분류 및 변환들의 새로운 세트의 생성으로 분리되는 통상적인 온-더-플라이 방식을 도시한다. 제1 단계는 잔차 블록들을 K개의 상이한 클래스들($S_1, \dots, S_K$)로 분류한다. 제2 단계에서, 그 특정 클래스에 대한 재구성 에러의 최소화를 사용하여 각각의 클래스에 대한 새로운 변환이 획득된다. 일반적으로, 변환들의 직교 세트를 생성하기 위해 SVD(Singular Value Decomposition) 및 KLT(Karhunen-Loeve Transform)가 사용된다. 이러한 2개의 단계들은 솔루션의 수렴 또는 중지 기준에 도달될 때까지 반복된다. 블록도에서 볼 수 있는 바와 같이, 시스템에 대한 입력은 분리가능하지 않은 직교 변환들($T_1 \dots T_K$)의 일부 초기 세트와 함께 인트라 프레임 또는 이미지이다. 시스템은 디코더에 전송된 비트스트림으로 인코딩될 필요가 있는 선택스 정보와 함께 학습된 변환들($T'_1 \dots T'_K$)의 세트를 출력한다. 일반적으로, 이러한 변환 기본 벡터들을 인코딩하기 위해 요구되는 오버헤드 비트들은 프레임 인코딩하기 위해 요구되는 비트들에 비해 상당히 크다.
- [0042] SVD의 에너지 압축 특성으로 인해, 오직 처음 'm'개의 벡터들만이 디코더로 송신되고 나머지 (n-m)개의 변환 벡터들은 그램-슈미트(Gram-Schmidt) 방법과 유사한 완료 알고리즘을 사용하여 생성되거나 제로로 강제되어 부분적 변환을 유도하는 학습된 변환의 불완전한 표현을 연역함으로써 오버헤드 비용이 상당히 감소될 수 있음이 관측되었다.
- [0043] BD-레이트(Bjontegaard Distortion rate)에 대한 변환의 최종 몇몇 벡터들을 드롭하는 효과를 예시하기 위해, 4K 시퀀스들 'PeopleOnStreet' 및 'Traffic'에 대해 크기 64×64 의 4개의 분리가능하지 않은 최적화된 변환들이 학습되었다. 이러한 시퀀스들에 대해 처음 'm'개의 벡터들이 유지되고 그 다음 기본 벡터들 중 나머지가 완료 알고리즘을 사용하여 완료되는 인코딩 테스트들이 수행된다. 도 6은 코딩된 기본 벡터들의 수에 대한 성능 이득의 변화를 도시한다. 수직 축은 앵커(HEVC 테스트 소프트웨어 HM15.0)에 대한 백분율 이득이며 변환 비용을 고려하지 않는다.
- [0044] 도 2에서 변환 벡터들의 단지 처음 절반, 즉 $m = 32$ 만 유지함으로써 BD-레이트 측면에서 무시할 수 있는 성능 드롭이 존재함이 관찰되었다. $m = 16$ 의 경우, 즉 처음 16 개의 기본 벡터들만이 코딩되는 경우, 모든 'n'개의 기본 벡터들을 코딩하여 획득된 전체 비트 레이트에 비해 성능이 1% 드롭되지만 변환 벡터들의 오버헤드 비용은 전체 오버헤드의 1/4까지 감소한다. $m = 8$ 의 경우, BD-레이트 성능 측면에서 상당한 성능 손실이 있지만 오버헤드가 더 감소한다. 일반적으로, 이것은 성능 손실과 오버헤드 비용 사이의 절충점이 존재함을 나타낸다.
- [0045] 오직 처음 'm'개의 변환 벡터들만을 인코딩할 때 BD-레이트의 측면에서 성능 드롭은 비디오 콘텐츠에 의존하기 때문에, 'm'의 최적 값을 추정하기 위해서 콘텐츠-적응형 방법이 필요하다. 직관적으로, 낮은 비트-레이트들에서, 계수들의 대부분은 제로로 양자화되고 높은 비트-레이트에서, 계수의 에너지는 고주파수들에서도 중요하다. 따라서, 'm'의 값은 양자화 파라미터 QP 뿐만 아니라 콘텐츠에 의존한다.
- [0046] 평균적으로 잔차 신호 에너지는 처음 몇몇 계수들에 더 많이 집중되며, 여기서 DC 계수는 평균적으로 최대 에너지를 갖고, 더 높은 주파수 계수들로 이동함에 따라 감소된다. 따라서, 고주파 계수들의 대부분은 제로로 양자화된다. 오버헤드로서 프레임과 함께 코딩되도록 요구되는 'm'의 최상의 값을 컴퓨팅하기 위해 간단한 임계치 기반 방법이 적용될 수 있다.
- [0047] E를 DCT 계수들의 에너지의 합이라고 하자. 임계치 t는 파라미터 p와 E의 곱셈에 의해 정의되어, 다음을 얻는다.

$$t = p \cdot E$$

'm'의 값은 이러한 임계치보다 큰 평균 에너지를 갖는 계수들의 수로부터 간단히 컴퓨팅될 수 있다. 'p'의 값은 실험적으로 발견될 수 있다. 표 1은 인코딩된 벡터들의 총 수의 백분율의 측면에서 'p'의 선택된 값에 대해 이러한 임계치보다 큰 벡터들의 수 'm'의 변화를 보여준다. 높은 QP에서 요구되는 벡터들의 평균 수는 낮은 QP에서 요구되는 수에 비해 훨씬 적다는 것이 표 1에서 관찰된다. 아울러, 벡터들의 수 'm'은 또한 콘텐츠에 따라 변한다.

표 1

표 1: 인코딩할 벡터들의 수 대 상이한 시퀀스들에 대한 임계치

QP	'p'	인코딩할 벡터들의 총 수의 백분율 (%단위)			
		PeopleOnStreet	Traffic	Nebuta	SteamLocomotive
22	0.01	0.23	0.24	0.30	0.32
	0.005	0.27	0.29	0.32	0.35
	0.001	0.45	0.48	0.46	0.46
27	0.01	0.20	0.22	0.28	0.31
	0.005	0.25	0.27	0.31	0.34
	0.001	0.38	0.39	0.38	0.43
32	0.01	0.18	0.19	0.26	0.28
	0.005	0.21	0.22	0.29	0.30
	0.001	0.29	0.30	0.39	0.36
37	0.01	0.14	0.15	0.23	0.23
	0.005	0.18	0.18	0.26	0.25
	0.001	0.23	0.23	0.30	0.30

본 원리들이 부분적 변환들을 사용하여 적용되는 경우, y의 범위는 x에 의존한다.

마지막 16-12 = 4개가 항상 제로가 되도록 m = 12를 갖는 부분적 변환을 사용하여 도 3의 예를 다시 고려한다. 계수 위치 12와 연관된 좌표(2,2)를 어떻게 가장 효율적으로 코딩해야 하는가? 하나의 가능성은 먼저 x=2를 코딩하는 것이다. 이의 이진화는 HEVC에서와 같이 001이다. 그 다음, y에 대한 가능한 범위는 0 내지 2이고, 따라서 HEVC에서와 같이 3까지의 절단된 단항 코딩 대신에, 2까지의 절단된 단항 코딩을 사용할 수 있음을 안다. 이는 y에 대해 001 대신 00의 이진화를 유도한다. 그리고, 이진화 프로세스에서 하나의 비트를 얻고, 따라서 더 양호한 압축 성능을 유도한다.

도 3의 예에서 이러한 실시예를 사용한 이진화의 일부 추가적인 예들:

(000,0) 대신에 (000, 없음)으로 이진화된 (3,0)

HEVC에서와 같이 (0,0)으로 이진화된 (0,0)

(3,3)은 불가능한 좌표이다.

부분적 변환들에 의한 실시예는 예를 들어, m = 57인 부분적 변환을 사용하는 8 × 8 TU의 예에서 도 7에서와 같이 서픽스를 사용하는 좌표 y로 쉽게 확장된다. 처음 57개의 변환된 계수들은 제로가 아닐 수 있지만, 최종 64-57 = 7개의 계수들은 반드시 제로이다. 최종 코딩된 계수 위치가 55라고 가정하자. 이는 좌표(x,y) = (5, 6)을 제공한다. y에 대한 프리픽스는 5이고 이의 서픽스는 0이다. HEVC에서, 서픽스는 1 비트로 코딩되어야 하지만, 이러한 경우 서픽스는 1일 수 없음을 아는데, 이는, 계수 55 바로 아래에 있는 계수 58이 이러한 경우 필수적으로 제로로 될 것이 알려져 있어서, 서픽스는 코딩되지 않고 코딩의 1 비트가 얻어지기 때문이다.

HEVC/H.265 표준에서, 이진 데이터를 코딩하기 위한 새로운 톨이 산술 코더, 즉, 콘텍스트 적응형 이진 산술 코딩(즉 CABAC)에서 제안되었다. 값 0 또는 1을 취하는 이진 심볼 s는 확률 p에 따라 1로 그리고 1-p는 0으로 코딩된다. 이러한 확률은 콘텍스트로부터 연역되고 각각의 심볼 코딩 이후 적응된다.

콘텍스트 값은 8 비트 값이다(도 8 참조). 선행 비트는 가장 가능성있는 심볼(즉, MPS)을 표현하고, 다음 7개의 비트들은 확률 p가 연역되는 확률 p'(또는 상태)를 표현한다.

콘텍스트 값의 업데이트는, 코딩된 심볼이 MPS와 동일할지 여부에 따라 도 9에 설명된 프로세스에 따라 수행된다.

2개의 표들 즉, 코딩된 심볼이 MPS이면 transIdxMPS, 및 코딩된 심볼이 MPS가 아니면, 즉, 최소 가능성 심볼

(LPS)이면, transIdxLPS를 통해 진화가 이루어진다. 이러한 표들은 이름이 pStateIdx인 엔트리 p'에 대해 표 2에 제공된다.

표 2

표 2: 콘텍스트 상태의 진화에 대한 표
표 9-41 - 상태 전환 표

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

MPS가 될 심볼 s의 확률 PMPS는 0부터 127까지 8 비트로 선형적으로 양자화된다. 이는,

$$p_{MPS} = (p' + 64) / 127 = (pStateIdx + 64) / 127$$

에 의해 콘텍스트 값으로부터 연역되고, 1이 될 심볼 s의 확률 p는 MPS의 값에 따라 PMPS로부터 명백하게 연역된다.

$p = p_{MPS} \text{MPS} = 1$ 인 경우

$p = 1 - p_{MPS} \text{MPS} = 0$ 인 경우

콘텍스트 적응형 코딩은, 코딩이 심볼이 속한 채널의 통계를 동적으로 따라가도록 허용하는 강력한 툴이다. 또한, 통계를 혼합하여 프로세스의 이점을 상실하지 않기 위해 각각의 채널은 자기 자신의 콘텍스트를 가져야 한다. 이는 많은 채널들을 모델링하기 위해 수백 개의 콘텍스트들을 사용하는 HEVC/H.265에서 많은 콘텍스트들의 광범위한 사용을 유도하였다. 예를 들어, 콘텍스트들을 사용하는 모든 채널들 사이에 다음이 존재한다.

- 모션 벡터 잔차들

- TU 코딩 플래그들,

- 최종 중요 계수 위치,

- 코딩 그룹 코딩 플래그들,

- 변환된 계수의 중요한 플래그들,

- 변환된 계수 크기(1 초과 및 2 초과) 플래그들,

- SAO 데이터,

- 다른 데이터

이러한 콘텍스트들/채널들 모두는 또한 대체로, 컬러 채널, 즉, 채널이 루마인지 또는 크로마인지 여부, 변환 유닛 크기, 변환된 계수의 위치, 인접 심볼 값들 및 다른 팩터들에 의존한다.

일례로, 도 10에 도시된 바와 같이, 코딩 그룹(CG) 코딩 플래그는 현재 CG의 아래 및 우측의 CG 코딩 플래그들이 1인지 여부에 따라 선택된다.

대체로, 콘텍스트 선택은 많은 것들, 및 그에 따른 많은 수의 콘텍스트들에 의존한다. 물론, 디코더는 인코더와의 동기화 및 스트림의 파싱을 보장하기 위해 인코더 측에서 수행된 것과 일치하도록 콘텍스트 값들을 업데이트

트해야 한다.

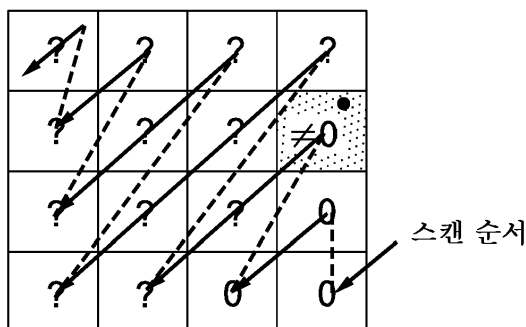
- [0080] 본 명세서에서 설명된 실시예들은 이미지 블록 값들로부터 계수들을 획득하기 위해 사용되는 부분적 변환의 크기를 고려함으로써 잠재적 y 좌표 후보들의 범위를 감소시키는 것을 제안한다.
- [0081] 설명된 원리들을 구현하기 위해 필요한 연관된 선택스는 부분적 변환의 크기 m 의 값의 인코딩이어야 한다. 이러한 값은, 개선된 이진화 및 절단된 단항 코딩이 또한 디코더에서 결정가능하도록 프레임 또는 슬라이스 레벨로 코딩될 수 있다.
- [0082] 디코딩 프로세스에 관하여, 본 명세서에서 설명된 원리들은 그에 따라 최종 코딩된 계수 위치를 디코딩할 수 있도록 디코더에 영향을 미쳐야 한다.
- [0083] 부분적 변환들을 사용하는 이러한 실시예들에서, y 서픽스의 개선된 이진화 및 절단된 단항 코딩이 또한 영향을 받는다.
- [0084] 본 원리들의 상기 특징들에 추가로, 설명된 실시예들은 m 의 코딩 또는 계수들의 서브세트가 결정될 수 있는 정보가 비트스트림에 존재함을 제공한다. 또한, 최종 코딩된 계수는 2개의 좌표(x, y)에 의해 제공되고, x 의 서브세트 및 값은 y 에 대한 범위를 결정하기 위해 사용된다. 또한, 절단된 코딩은 결정된 범위에 기초하여 y 프리픽스 및 서픽스에 대해 사용된다.
- [0085] 제안된 아이디어는 비디오 스트림의 선택스에 존재하고 적용될 디코딩 방법을 의미한다는 점에서 규범적이다. 이와 같이, 본 아이디어들은 HEVC의 계승자와 같은 비디오 표준으로 구현될 수 있다.
- [0086] 변환된 계수들의 세트를 코딩하기 위한 방법(1100)의 일 실시예가 도 11에 도시되어 있다. 방법은 시작 블록(1101)에서 시작하고 제어는 변환된 계수 값들을 생성하기 위해 이미지 값들의 블록을 변환하기 위한 블록(1110)으로 진행한다. 제어는 블록(1110)으로부터, y 좌표를 코딩하기 위해 변환의 크기의 지식을 사용하여 최종 코딩된 계수를 엔트로피 코딩하기 위한 블록(1120)으로 진행한다.
- [0087] 변환된 계수들의 세트를 코딩하기 위한 장치(1200)의 일 실시예가 도 12에 도시되어 있다. 장치는 그 입력 포트 상의 이미지 값들의 블록을 수신하고 그 출력 포트 상에 변환된 계수들을 생성하는 변환 회로(1210)를 포함한다. 변환 회로(1210)의 이러한 출력 포트는 엔트로피 코더(1220)로의 입력과 신호 연결성을 갖는다. 엔트로피 코더(1220)는 최종 코딩된 계수 값을 생성하도록 y 좌표를 코딩하기 위해 변환의 크기의 지식을 사용하여 최종 코딩된 계수 위치의 y 좌표를 코딩한다.
- [0088] 변환된 계수들의 세트를 디코딩하기 위한 방법(1300)의 실시예가 도 13에 도시되어 있다. 방법은 시작 블록(1301)에서 시작하고, 제어는 y 좌표를 디코딩하기 위한 변환의 크기의 지식을 사용하여 최종 코딩된 계수를 엔트로피 디코딩하기 위한 블록(1310)으로 진행한다. 제어는 블록(1310)으로부터, 이미지 값들을 생성하기 위해 계수들을 역변환하기 위한 블록(1320)으로 진행한다.
- [0089] 변환된 계수들의 세트를 디코딩하기 위한 장치(1400)의 실시예가 도 14에 도시되어 있다. 장치는 최종 코딩된 계수에 대한 코드를 포함하는 변환 계수들을 수신하고 y 좌표를 디코딩하기 위한 변환의 크기의 지식을 사용하여 최종 코딩된 계수의 y 좌표를 디코딩하는 엔트로피 디코더(1410)를 포함한다. 엔트로피 디코더(1410)의 출력은 역변환 회로(1420)의 입력과 신호 연결성을 갖는다. 역변환 회로(1420)는 그 입력 포트에서 변환된 계수들을 수신하고 이들을 역변환하여 그 출력 포트 상에 이미지 값들의 블록을 생성한다.
- [0090] 전술한 실시예들은 셋톱 박스(STB)들, 모뎀들, 게이트웨이들 또는 비디오 인코딩 또는 디코딩을 수행하는 다른 디바이스들에서 구현될 수 있다.
- [0091] 도면들에 도시된 다양한 엘리먼트들의 기능들은 적절한 소프트웨어와 관련하여 소프트웨어를 실행할 수 있는 하드웨어 뿐만 아니라 전용 하드웨어의 사용을 통해 제공될 수 있다. 프로세서에 의해 제공되는 경우, 기능들은 단일 전용 프로세서, 단일 공유 프로세서, 또는 복수의 개별적인 프로세서들에 의해 제공될 수 있고, 이들 중 일부는 공유될 수 있다. 또한, 용어 "프로세서" 또는 "제어기"의 명시적 사용은 소프트웨어를 실행할 수 있는 하드웨어만을 배타적으로 지칭하는 것으로 해석되어서는 안되며, 제한없이, 디지털 신호 프로세서("DSP") 하드웨어, 소프트웨어를 저장하기 위한 판독 전용 메모리("ROM") 소프트웨어, 랜덤 액세스 메모리("RAM") 및 비휘발성 스토리지를 묵시적으로 포함할 수 있다.
- [0092] 다른 하드웨어(종래 및/또는 주문형)가 또한 포함될 수 있다. 유사하게, 도면들에 도시된 임의의 스위치들은 단지 개념적이다. 이들의 기능은 프로그램 로직의 동작을 통해, 전용 로직을 통해, 프로그램 제어 및 전용 로

직의 상호작용을 통해, 또는 심지어는 수동으로 수행될 수 있고, 특정 기술은 문맥으로부터 더 구체적으로 이해되는 바와 같이 구현자에 의해 선택가능하다.

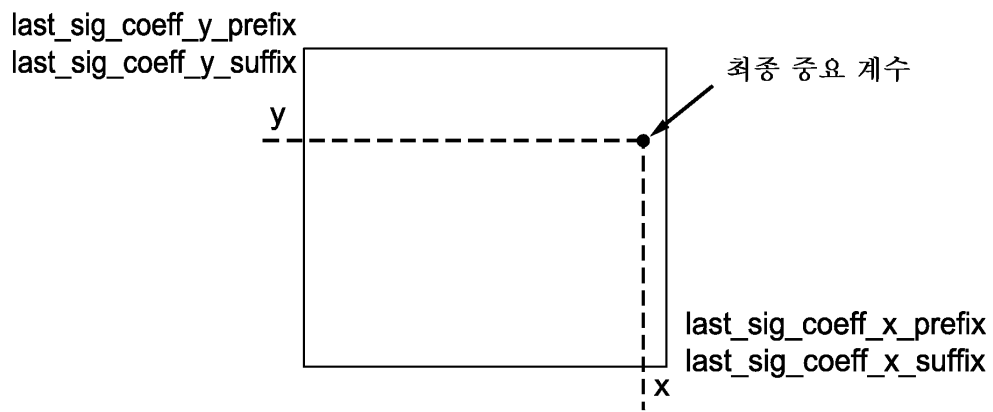
- [0093] 본 설명은 본 원리들을 예시한다. 따라서, 본 기술분야의 통상의 기술자들은 본원에 명시적으로 설명되거나 도시되지는 않았지만, 본 원리들을 구현하고 그 사상 및 범위 내에 포함되는 다양한 배열들을 고안할 수 있을 것임을 인식할 것이다.
- [0094] 본원에 인용된 모든 예들 및 조건부 언어는, 독자가 본 원리들 및 발명자(들)에 의해 기술 발전에 기여된 개념들을 이해하는 것을 돕는 교육적 목적을 위한 것이고, 이러한 특별히 인용된 예들 및 조건들을 제한하지 않는 것으로 해석되어야 한다.
- [0095] 또한, 원리들, 양태들 및 본 원리들의 실시예들 뿐만 아니라 이의 특정 예들을 기재한 본 명세서의 모든 설명들은 이의 구조적 및 기능적 등가물들 둘 모두를 포함하도록 의도된다.
- [0096] 추가적으로, 이러한 균등물들은 현재 공지된 균등물들 뿐만 아니라 장래에 개발되는 균등물들, 즉 구조와 무관하게, 동일한 기능을 수행하는 임의의 개발된 엘리먼트들 모두를 포함하는 것으로 의도된다.
- [0097] 따라서, 예를 들어, 본 명세서에 제시된 블록도들은 본 원리들을 구현하는 예시적인 회로의 개념도들을 표현함을 본 기술분야의 통상의 기술자들은 인식할 것이다. 유사하게, 임의의 플로우차트들, 흐름도들, 상태 전이도들, 의사 코드 등은 컴퓨터 판독가능 매체에서 실질적으로 표현될 수 있고, 따라서 컴퓨터 또는 프로세서가 명시적으로 도시되든 도시되지 않든 이러한 컴퓨터 또는 프로세서에 의해 실행될 수 있는 다양한 프로세스들을 표현함을 인식할 것이다.
- [0098] 본 명세서의 청구항들에서, 특정 기능을 수행하기 위한 수단으로 표현된 임의의 엘리먼트는 예를 들어 a) 그 기능을 수행하는 회로 엘리먼트들의 조합 또는 b) 임의의 형태의 소프트웨어를 포함하는 그 기능을 수행하는 임의의 방법을 포함하도록 의도되고, 따라서, 기능을 수행하기 위해 그 소프트웨어를 실행하기 위한 적절한 회로와 결합된 펌웨어, 마이크로코드 등을 포함한다. 이러한 청구항들에 의해 정의된 바와 같은 본 원리들은, 다양한 인용된 수단들에 의해 제공되는 기능들이, 청구항들이 요구하는 방식으로 조합되고 결합된다는 사실에 있다. 따라서, 이러한 기능들을 제공할 수 있는 임의의 수단은 본 명세서에 제시된 것들과 동등하다고 간주된다.
- [0099] 본 원리들의 "일 실시예" 또는 "실시예" 뿐만 아니라 이들의 다른 변화예들에 대한 본 명세서에서의 참조는, 실시예와 관련하여 설명되는 특정한 특징, 구조, 특성 등이 본 원리들의 적어도 하나의 실시예에 포함되는 것을 의미한다. 따라서, 본 명세서 전반에 걸친 다양한 위치들에서 등장하는 구 "일 실시예에서" 또는 "실시예에서" 뿐만 아니라 다른 변화예들은, 반드시 동일한 실시예 전부를 참조할 필요는 없다.

도면

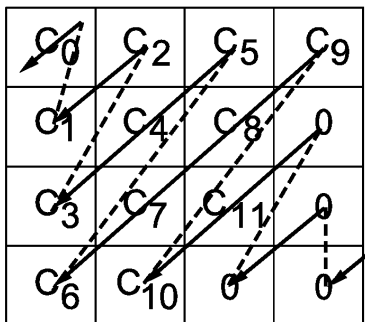
도면1



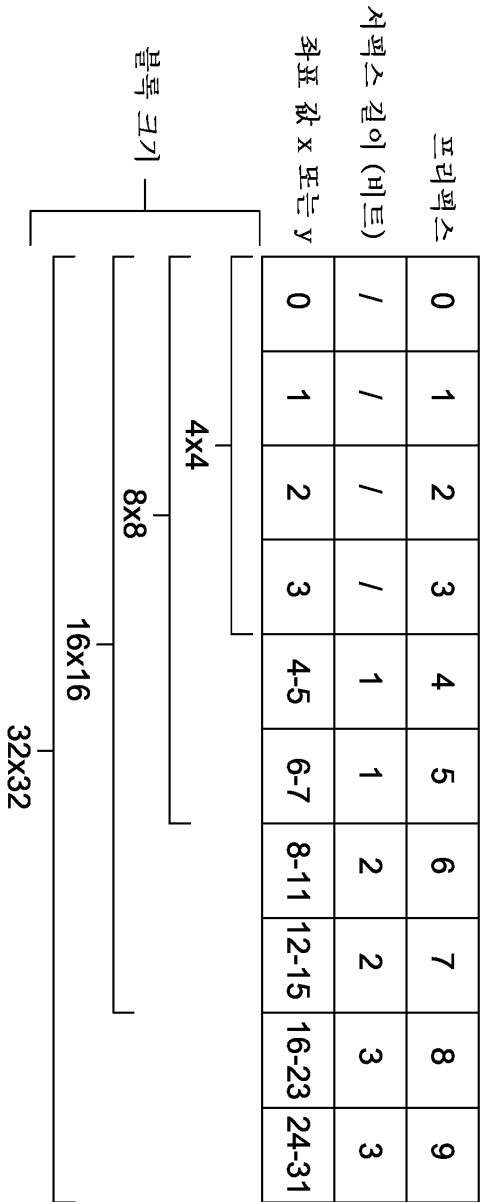
도면2



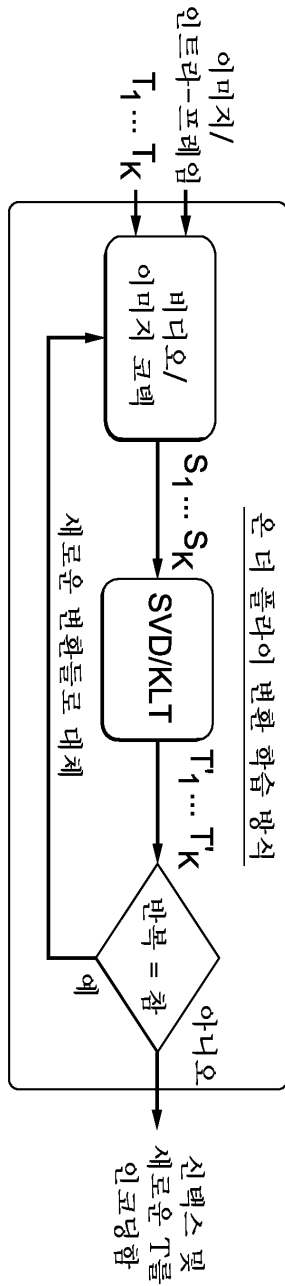
도면3



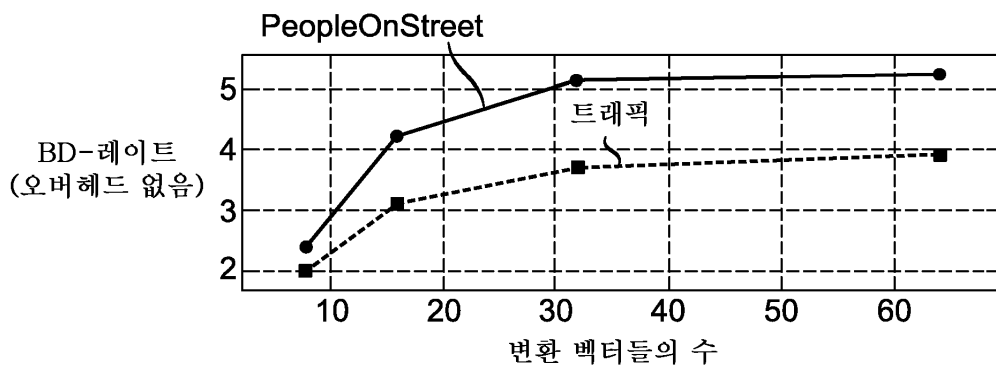
도면4



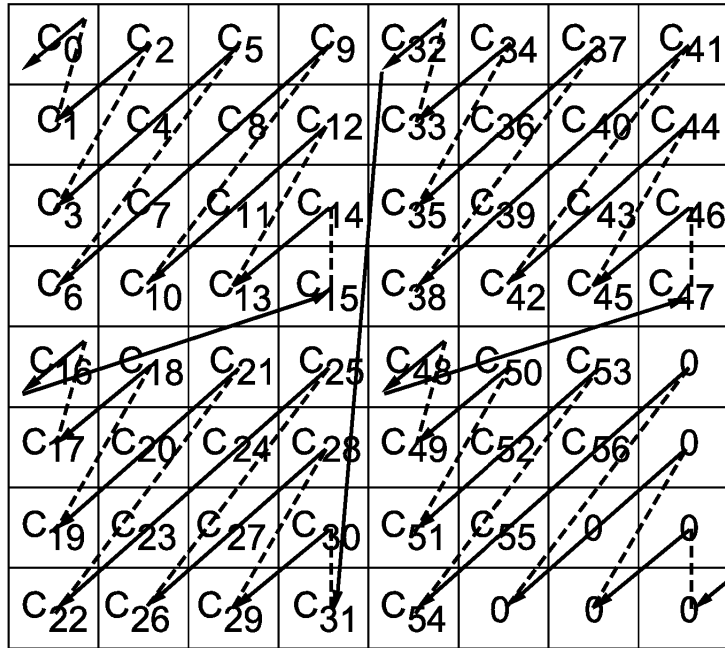
도면5



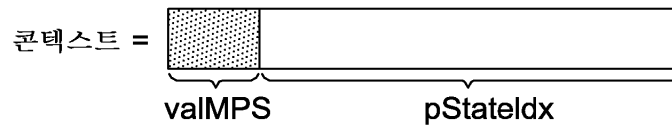
도면6



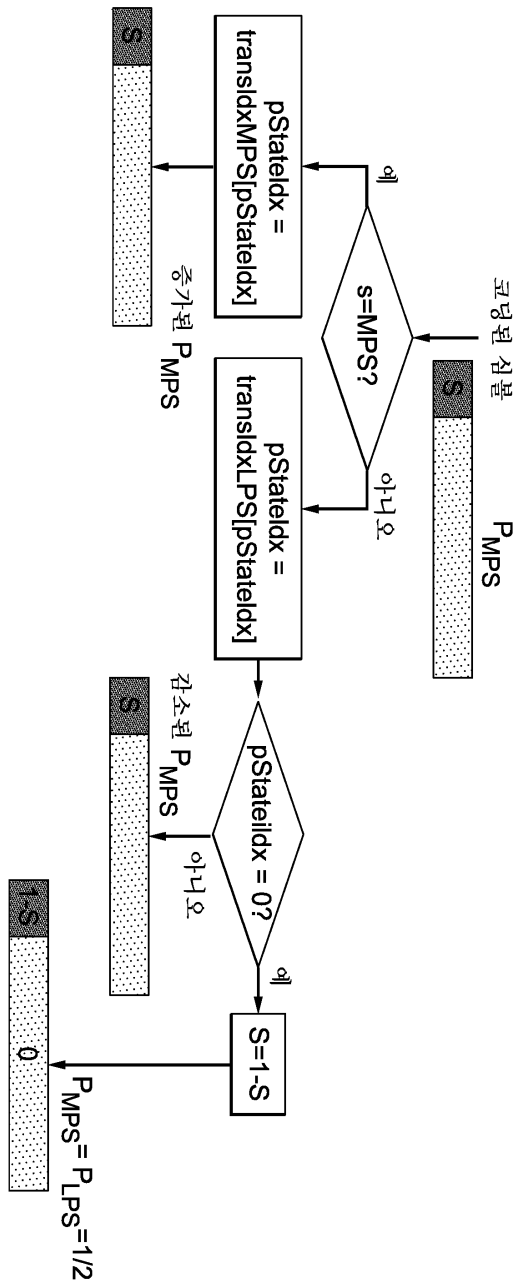
도면7



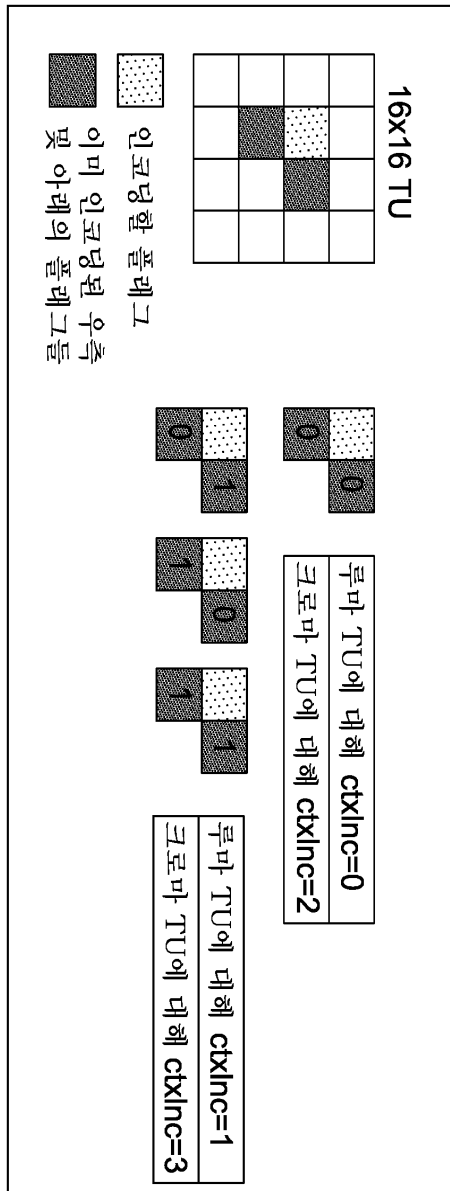
도면8



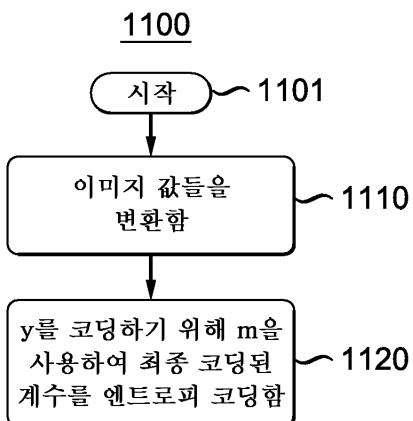
도면9



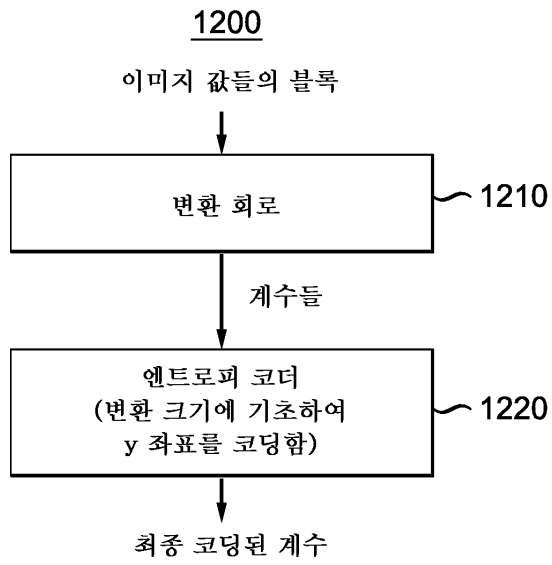
도면10



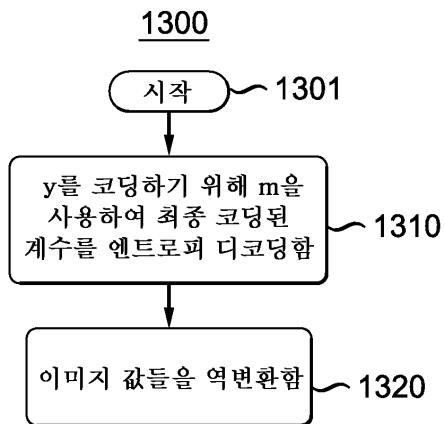
도면11



도면12



도면13



도면14

