

- 전체 청구항 수 : 총 18 항

대표도 - 도2b



써, 상기 블록들의 서브세트들의 각각을 인코딩하는 단계(C3) - 상기 인코딩 단계는, 이미지의 첫 번째 블록에 대하여, 상기 엔트로피 인코딩 모듈의 상태 변수들을 초기화하는 서브-단계(C33)를 포함함 -, 및 상기 인코딩된 블록들의 서브세트들 중 적어도 하나를 나타내는 데이터의 적어도 하나의 서브-스트림을 생성하는 단계를 포함하는 인코딩 방법에 관한 것이다. 현재 블록이 검토중인 블록들의 서브세트의 인코딩되고 있는 첫 번째 블록인 경우, 상기 첫 번째 현재 블록에 대한 심볼 발생의 확률들은 적어도 하나의 다른 서브세트의 인코딩된 및 디코딩된 미리 결정된 블록에 대하여 결정되었던 것이다. 현재 유닛이 검토중인 서브세트의 마지막 인코딩된 유닛인 경우, 검토중인 상기 서브세트의 블록들의 인코딩 동안 상기 심볼들과 연관되었던 디지털 정보 모두는 검토중인 적어도 상기 서브세트를 나타내는 상기 서브-스트림에 기록되고(C45), 초기화 서브-단계가 실시된다(C46).

(52) CPC특허분류

H04N 19/124 (2015.01)

H04N 19/176 (2015.01)

H04N 19/197 (2015.01)

H04N 19/423 (2015.01)

H04N 19/436 (2015.01)

H04N 19/60 (2015.01)

명세서

청구범위

청구항 1

이미지를 코딩하기 위한 방법으로서,

상기 이미지를 복수의 블록들로 세그먼트화하는 단계(segmenting);

상기 블록들을 미리 결정된 수의 블록들의 행들로 그룹화하는 단계(grouping);

엔트로피 코딩 모듈을 사용하여, 상기 블록들의 행들의 현재 블록을 코딩하는 단계 - 상기 코딩하는 단계는,

상기 현재 블록이 행의 인코딩 순서에서 첫 번째 블록이고 상기 행이 상기 인코딩 순서에서 상기 이미지의 첫 번째 행이 아닐 때,

상기 현재 블록에 대한 심볼 발생의 확률들을 결정하는 것,

상기 결정된 확률들로 상기 엔트로피 코딩 모듈의 상태 변수들을 초기화하는 것, 및

상기 현재 블록을 코딩하는 것

을 포함하고,

상기 확률들은 적어도 하나의 다른 행의 미리 결정된 블록을 코딩하는 것에 의해 결정되었던 것이고, 상기 미리 결정된 블록은 상기 다른 행에서 상기 인코딩 순서로 두 번째 블록임 - ; 및

상기 이미지에 대한 적어도 하나의 데이터 서브-스트림을 생성하는 단계

를 포함하는 이미지를 코딩하기 위한 방법.

청구항 2

제1항에 있어서,

상기 현재 블록을 코딩하는 것은, 상기 현재 블록에서 예측된 블록을 감산함(subtracting)으로써 생성된 잔여 블록의 변환에 의해 획득된 변환 블록의 양자화를 포함하는 것인, 이미지를 코딩하기 위한 방법.

청구항 3

제2항에 있어서,

상기 예측된 블록은 상기 현재 블록의 근사치(approximation)인 것인, 이미지를 코딩하기 위한 방법.

청구항 4

제1항에 있어서,

상기 코딩하는 단계는,

상기 현재 블록이 상기 행 - 상기 행은 상기 인코딩 순서에서 상기 이미지의 첫 번째 행이 아님 - 의 상기 인코딩 순서에서 마지막 블록일 때,

상기 엔트로피 코딩 모듈에 대응하는 버퍼 메모리를 비우는 것(emptying) - 상기 비우는 것은, 상기 버퍼 메모리로부터 송신되지 않은 비트들을 추출하는 것 및 상기 송신되지 않은 비트들을 디코딩 모듈로 목적지가 예정된(destined) 데이터 스트림에 기록하는 것을 포함함 - ; 및

상기 엔트로피 코딩 모듈의 상태 변수들을 초기화하는 것 - 상기 초기화하는 것은 상한(high bound) 및 하한(low bound) 사이의 범위로 간격(interval)을 초기화하는 것을 포함함 -

을 포함하는 것인, 이미지를 코딩하기 위한 방법.

청구항 5

제1항에 있어서,

적어도 2개의 블록들의 행들이 적어도 하나의 다른 블록들의 행과 병렬로 코딩될 때, 상기 적어도 2개의 코딩된 블록들의 행들은 동일한 데이터 서브-스트림에 포함되는 것인, 이미지를 코딩하기 위한 방법.

청구항 6

제1항에 있어서,

상기 코딩된 블록들의 행들이 미리 결정된 순서로 병렬로 디코딩되도록 의도될 때, 상기 데이터 서브-스트림은 디코더로 송신되기 전에 상기 미리 결정된 순서에 따라 먼저 정렬되는(ordered) 것인, 이미지를 코딩하기 위한 방법.

청구항 7

인코더에 있어서,

하나 이상의 프로세서; 및

상기 하나 이상의 프로세서에 의해 실행될 때 상기 하나 이상의 프로세서로 하여금 동작들을 수행하게 하는 명령어들을 저장한 하나 이상의 저장 디바이스를 포함하고, 상기 동작들은,

이미지를 복수의 블록들로 세그먼트화하는 동작;

상기 블록들을 미리 결정된 수의 블록들의 행들로 그룹화하는 동작;

엔트로피 코딩 모듈을 사용하여, 상기 블록들의 행들의 현재 블록을 코딩하는 동작 - 상기 코딩하는 동작은,

상기 현재 블록이 행의 인코딩 순서에서 첫 번째 블록이고 상기 행이 상기 인코딩 순서에서 상기 이미지의 첫 번째 행이 아닐 때,

상기 현재 블록에 대한 심볼 발생의 확률들을 결정하는 것,

상기 결정된 확률들로 상기 엔트로피 코딩 모듈의 상태 변수들을 초기화하는 것, 및

상기 현재 블록을 코딩하는 것

을 포함하고,

상기 확률들은 적어도 하나의 다른 행의 미리 결정된 블록을 코딩하는 것에 의해 결정되었던 것이고, 상기 미리 결정된 블록은 상기 다른 행에서 상기 인코딩 순서로 두 번째 블록임 - ; 및

상기 이미지에 대한 적어도 하나의 데이터 서브-스트림을 생성하는 동작

을 포함하는 것인, 인코더.

청구항 8

제7항에 있어서,

상기 현재 블록을 코딩하는 것은, 상기 현재 블록에서 예측된 블록을 감산함으로써 생성된 잔여 블록의 변환에 의해 획득된 변환 블록의 양자화를 포함하는 것인, 인코더.

청구항 9

제8항에 있어서,

상기 예측된 블록은 상기 현재 블록의 근사치인 것인, 인코더.

청구항 10

제7항에 있어서,

상기 코딩하는 동작은,

상기 현재 블록이 상기 행 - 상기 행은 상기 인코딩 순서에서 상기 이미지의 첫 번째 행이 아님 - 의 상기 인코딩 순서에서 마지막 블록일 때,

상기 엔트로피 코딩 모듈에 대응하는 버퍼 메모리를 비우는 것 - 상기 비우는 것은, 상기 버퍼 메모리로부터 송신되지 않은 비트들을 추출하는 것 및 상기 송신되지 않은 비트들을 디코딩 모듈로 목적지가 예정된 데이터 스트림에 기록하는 것을 포함함 - ; 및

상기 엔트로피 코딩 모듈의 상태 변수들을 초기화하는 것 - 상기 초기화하는 것은 상한 및 하한 사이의 범위로 간격을 초기화하는 것을 포함함 -

을 포함하는 것인, 인코더.

청구항 11

제7항에 있어서,

적어도 2개의 블록들의 행들이 적어도 하나의 다른 블록들의 행과 병렬로 코딩될 때, 상기 적어도 2개의 코딩된 블록들의 행들은 동일한 데이터 서브-스트림에 포함되는 것인, 인코더.

청구항 12

제7항에 있어서,

상기 코딩된 블록들의 행들이 미리 결정된 순서로 병렬로 디코딩되도록 의도될 때, 상기 데이터 서브-스트림은 디코더로 송신되기 전에 상기 미리 결정된 순서에 따라 먼저 정렬되는 것인, 인코더.

청구항 13

하나 이상의 프로세서에 의해 실행될 때 상기 하나 이상의 프로세서로 하여금 동작들을 수행하게 하는 명령어들이 저장된 비일시적 컴퓨터 판독 가능한 매체에 있어서, 상기 동작들은,

이미지를 복수의 블록들로 세그먼트화하는 동작;

상기 블록들을 미리 결정된 수의 블록들의 행들로 그룹화하는 동작;

엔트로피 코딩 모듈을 사용하여, 상기 블록들의 행들의 현재 블록을 코딩하는 동작 - 상기 코딩하는 동작은,

상기 현재 블록이 행의 인코딩 순서에서 첫 번째 블록이고 상기 행이 상기 인코딩 순서에서 상기 이미지의 첫 번째 행이 아닐 때,

상기 현재 블록에 대한 심볼 발생의 확률들을 결정하는 것,

상기 결정된 확률들로 상기 엔트로피 코딩 모듈의 상태 변수들을 초기화하는 것, 및

상기 현재 블록을 코딩하는 것

을 포함하고,

상기 확률들은 적어도 하나의 다른 행의 미리 결정된 블록을 코딩하는 것에 의해 결정되었던 것이고, 상기 미리 결정된 블록은 상기 다른 행에서 상기 인코딩 순서로 두 번째 블록임 - ; 및

상기 이미지에 대한 적어도 하나의 데이터 서브-스트림을 생성하는 동작

을 포함하는 것인, 비일시적 컴퓨터 판독 가능한 매체.

청구항 14

제13항에 있어서,

상기 현재 블록을 코딩하는 것은, 상기 현재 블록에서 예측된 블록을 감산함으로써 생성된 잔여 블록의 변환에 의해 획득된 변환 블록의 양자화를 포함하는 것인, 비일시적 컴퓨터 판독 가능한 매체.

청구항 15

제14항에 있어서,

상기 예측된 블록은 상기 현재 블록의 근사치인 것인, 비밀시적 컴퓨터 판독 가능한 매체.

청구항 16

제13항에 있어서,

상기 코딩하는 동작은,

상기 현재 블록이 상기 행 - 상기 행은 상기 인코딩 순서에서 상기 이미지의 첫 번째 행이 아님 - 의 상기 인코딩 순서에서 마지막 블록일 때,

상기 엔트로피 코딩 모듈에 대응하는 버퍼 메모리를 비우는 것 - 상기 비우는 것은, 상기 버퍼 메모리로부터 송신되지 않은 비트들을 추출하는 것 및 상기 송신되지 않은 비트들을 디코딩 모듈로 목적지가 예정된 데이터 스트림에 기록하는 것을 포함함 - ; 및

상기 엔트로피 코딩 모듈의 상태 변수들을 초기화하는 것 - 상기 초기화하는 것은 상한 및 하한 사이의 범위로 간격을 초기화하는 것을 포함함 -

을 포함하는 것인, 비밀시적 컴퓨터 판독 가능한 매체.

청구항 17

제13항에 있어서,

적어도 2개의 블록들의 행들이 적어도 하나의 다른 블록들의 행과 병렬로 코딩될 때, 상기 적어도 2개의 코딩된 블록들의 행들은 동일한 데이터 서브-스트림에 포함되는 것인, 비밀시적 컴퓨터 판독 가능한 매체.

청구항 18

제13항에 있어서,

상기 코딩된 블록들의 행들이 미리 결정된 순서로 병렬로 디코딩되도록 의도될 때, 상기 데이터 서브-스트림은 디코더로 송신되기 전에 상기 미리 결정된 순서에 따라 먼저 정렬되는 것인, 비밀시적 컴퓨터 판독 가능한 매체.

발명의 설명

기술 분야

[0001] 본 발명은 일반적으로 이미지 프로세싱의 분야에 관한 것으로, 보다 정확하게는, 디지털 이미지들 및 디지털 이미지들의 시퀀스의 코딩 및 디코딩에 관한 것이다.

[0002] 본 발명은 따라서, 특히, 현재의 비디오 코더들(MPEG, H.264 등) 또는 근간의 비디오 코더들(ITU-T/VCEG(H.265) 또는 ISO/MPEG(HVC))에서 실시되는 비디오 코딩에 적용될 수 있다.

배경 기술

[0003] 현재의 비디오 코더들(MPEG, H264 등)은 비디오 시퀀스의 블록-방식(blockwise) 표시를 이용한다. 이미지들은 매크로-블록들로 세그먼트화되고(segmented), 각 매크로-블록은 그 자체를 블록들로 세그먼트화하며, 각 블록, 또는 매크로-블록은 인트라(intra)-이미지 또는 인터(inter)-이미지 예측에 의해 코딩된다. 따라서, 당업자에 의해 알려진 움직임 보상으로 하나 또는 둘 이상의 코딩-디코딩된 기준 이미지들에 대하여, 어떤 이미지들은 공간 예측(인트라 예측)에 의해 코딩되는 한편, 다른 이미지들은 시간 예측(인터 예측)에 의해 코딩된다. 게다가, 각 블록에 대하여, 원래 블록 마이너스 예측에 대응하는 잔차 블록(residual block)이 코딩될 수 있다. 이런 블록의 계수들은 가능하면 변환(transformation) 후에 양자화되고, 그 다음 엔트로피 코더에 의해 코딩된다.

[0004] 인트라 예측 및 인터 예측은 이전에 코딩 및 디코딩되었던 어떤 블록들이 현재 블록을 예측하기 위해 디코더 또는 코더 중 어느 하나에서 사용될 수 있도록 이용가능한 것을 필요로 한다. 이와 같은 예측 코딩의 개략적 예가 도 1에 도시되는데, 여기서, 이미지 I_N 는 블록들로 분할되고, 이런 이미지의 현재 블록 MB_i 은 해칭된 화살표에

의해 표시된 바와 같은, 미리 결정된 수 3개의 이전에 코딩 및 디코딩된 블록들 MB_{r_1} , MB_{r_2} 및 MB_{r_3} 에 대하여 예측 코딩이 이루어진다. 상술한 3개의 블록들은, 특히, 현재 블록 MB_i 의 좌측에 바로 위치한 블록 MB_{r_1} , 및 현재 블록 MB_i 의 바로 위 및 우측 위에 각각 위치한 2개의 블록들 MB_{r_2} 및 MB_{r_3} 을 포함한다.

[0005] 엔트로피 코더가 여기서 더욱 특별한 관심대상이다. 엔트로피 코더는 도착의 순서로 정보를 인코딩한다. 이미지의 상단 좌측의 블록에서 시작하는 기준 PRS에 의해 도 1에 도시된 바와 같이, 전형적으로 블록들의 행 단위(row-by-row) 순회(traversal)는 "래스터-주사(raster-scan)" 형태로 수행된다. 각 블록에 대하여, 블록의 표현을 위해 필요한 정보의 다양한 항목들(블록의 유형, 예측의 모드, 잔차 계수(residual coefficient)들 등)은 엔트로피 코더로 순차적으로 보내진다.

[0006] AVC 압축 표준(또한 ISO-MPEG4 part 10 및 ITU-T H.264란 명칭으로 알려진)으로 도입된 합리적인 복잡성의 효율적인 산술 코더(arithmetical coder), 소위 "CABAC"("Context Adaptive Binary Arithmetic Coder")가 이미 알려져 있다.

[0007] 이런 엔트로피 코더는 다양한 개념들을 실시한다:

[0008] - 산술 코딩 : 문서 J. Rissanen and G. G. Langdon Jr, "Universal modeling and coding," IEEE Trans. Inform. Theory, vol. IT-27, pp. 12-23, Jan. 1981에서 처음으로 설명된 바와 같은 코더는 심볼을 코딩하기 위해 이런 심볼의 발생의 확률을 사용한다;

[0009] - 콘텍스트(context)에 적응 : 여기서, 이는 코딩될 심볼들의 발생의 확률의 적응을 의미한다. 한편, 학습이 작동 중(on the fly) 수행된다. 다른 한편, 이전 코딩된 정보의 상태에 따라 특정 콘텍스트가 코딩을 위해 사용된다. 각 콘텍스트에 대하여, 심볼의 발생의 고유(inherent) 확률이 대응되어 있다. 예를 들면, 콘텍스트는 주어진 구성 또는 이웃의 상태(예를 들면, 이웃으로부터 선택된 "인트라" 모드들의 수 등)에 따라 코딩된 심볼의 유형(잔차의 계수의 표시 또는 코딩 모드의 시그널링(signaling) 등)에 대응한다;

[0010] - 이진화(binanzation) : 코딩될 심볼들의 일련의 비트들의 정형(shaping)이 수행된다. 그 후에, 이들 다양한 비트들은 연속하여 이진 엔트로피 코더로 보내진다.

[0011] 따라서, 이들 엔트로피 코더는, 사용된 각 콘텍스트에 대하여, 고려중인(under consideration) 콘텍스트를 위해 이전에 코딩된 심볼들에 대하여 확률들을 작동중 학습하기 위한 시스템을 실시한다. 이런 학습은 이들 심볼들의 코딩의 순서에 기반한다. 전형적으로, 이미지는 앞에서 설명된 "래스터-주사" 유형의 순서에 따라 순회된다(traversed).

[0012] 0 또는 1과 같을 수 있는 주어진 심볼 b 의 코딩 동안, 이들 심볼의 발생의 확률 P_i 의 학습은 다음의 방식으로 현재 블록 MB_i 에 대하여 업데이트 된다 :

[0013]
$$p_i(b=0) = \alpha P_{i-1}(b=0) + \begin{cases} (1-\alpha) & \text{코딩된 비트가 0이면} \\ 0 & \text{그외} \end{cases}$$

[0014] 여기서, α 는 미리 결정된 값, 예를 들면, 0.95이고 P_{i-1} 은 이런 심볼의 마지막 발생 동안 연산된 심볼 발생 확률이다.

[0015] 이런 엔트로피 코딩의 개략적 예가 도 1에 도시되는데, 여기서 이미지 I_N 의 현재 블록 MB_i 는 엔트로피 코딩이 이루어진다. 블록 MB_i 의 엔트로피 코딩이 시작되면, 사용된 심볼 발생 확률들은 "래스터 주사" 유형의 블록들의 상술한 행 단위 순회에 따라 현재 블록 MB_i 에 바로 앞서는 것인, 이전에 코딩 및 디코딩된 블록의 코딩 이후에 이렇게 획득된 것들이다. 블록 간(block-to-block) 의존에 기반한 이런 학습은 어떤 블록들에 대하여 단지 도면의 명료성을 위해, 얇은 선 화살표들로 도 1에서 표시된다.

[0016] 이런 유형의 엔트로피 코딩의 문제점은 행의 시작에 위치한 심볼을 코딩하는 경우, 블록들의 "래스터 주사" 순회를 고려하면, 사용된 확률들이 주로 이전 행의 마지막에 위치한 심볼들에 대하여 관측된 것에 대응한다는 사실에 있다. 이제, 심볼 확률들의 가능한 공간 변화 때문에(예를 들면, 움직임 정보의 항목과 관련된 심볼에 대하여, 이미지의 우측 부분에 위치한 움직임은 좌측 부분에서 관찰된 것과 상이할 수 있고, 따라서 뒤따른 국부(local) 확률들에 대해서도 유사함), 확률들의 국부적 일치(conformity)의 부족이 관찰될 수 있고, 그에 의해

코딩 동안 효율성의 손실을 초래할 수 있다.

[0017] 이런 현상을 제한하기 위해, 코딩 및 디코딩이 순차적인 채로 남아있지만, 더 우수한 국부 정합성(consistency)의 보장을 목표로 블록들의 순회의 순서의 수정들에 대한 제안들이 이루어졌다.

[0018] 그 안에 이런 유형의 엔트로피 코더의 다른 문제점이 있다. 실제로, 심볼의 코딩 및 디코딩이 그것에 학습된 확률의 상태에 의존하면, 심볼들의 디코딩은 코딩 동안 사용된 것과 동일 순서로만 이루어질 수 있다. 전형적으로, 디코딩은 순차적이어야만 할 수 있고, 따라서, 몇몇 심볼들의 병렬 디코딩을 방지한다(예를 들면, 멀티-코어 아키텍처들로부터 이익을 얻기 위해).

[0019] 문헌 : Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, 및 Ajay Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 560-576, July 2003은 CABAC 엔트로피 코더가 비-정수의 비트들을 코딩될 현재 알파벳의 각 심볼에 할당하는 특별한 특징을 갖고, 이는 0.5보다 큰 심볼 발생 확률들에 대해 유리하다고 또한 언급한다. 특히, CABAC 코더는 그것이 몇몇 심볼들을 판독할 때까지 대기하고, 그 다음 판독된 이런 심볼들의 세트에, 코더가 디코더로 전송될 압축된 스트림에 기록하는 미리 결정된 수의 비트들을 할당한다. 이런 제공은 몇몇 심볼들에 대한 비트들을 "상호적으로 하는 것(mutualize)", 그리고 분수의 비트들에 관한 심볼을 코딩하는 것을 가능하게 하며, 이런 수는 심볼에 의해 실제로 이송된 정보와 근접한 정보를 반영한다. 판독된 심볼들과 관련된 다른 비트들은 압축된 스트림으로 전송되는 것이 아니라, 이런 다른 비트들을 다시 상호적으로 하는 것을 가능하게 하는 CABAC 코더에 의해 판독된 하나 또는 둘 이상의 새로운 심볼들에 할당되도록 대기하면서 대기중이다. 알려진 방식으로, 엔트로피 코더는 주어진 순간에 이들 전송하지 않은 비트들의 "비움(emptying)"을 착수한다. 달리 말하면, 상기 주어진 순간에, 코더는 아직 전송되지 않은 비트들을 추출하고 그것들을 디코더로 목적지가 예정된(destinated) 압축된 스트림에 기록한다. 이런 비움은 압축된 스트림이, 디코더가 알파벳의 심볼들을 모두 디코딩하게 할 비트들을 모두 포함함을 보장하도록, 예를 들면, 코딩될 마지막 심볼들이 판독되었던 순간에 발생한다. 더 일반적인 방식으로, 비움이 수행된 순간은 주어진 코더/디코더에 독특한 성능 및 기능성들의 함수로서 결정된다.

[0020] 2011년 4월 15일자의 인터넷 어드레스 http://research.microsoft.com/en-us/um/people/jinl/paper_2002/msri_jpeg.htm 에서 이용가능한 문헌은 JPEG 2000 압축 표준을 따르는 정지 이미지들을 코딩하기 위한 방법을 설명한다. 이런 방법에 따르면, 정지 이미지 데이터는 이산 웨이브렛 변환(discrete wavelet transform)이 이루어지고 뒤이어 양자화가 이루어지며, 그에 의해 양자화 지수들과 각각 관련된 양자화된 웨이브렛 계수들을 획득하는 것을 가능하게 한다. 획득된 양자화 지수들은 엔트로피 코더로 코딩된다. 양자화된 계수들은 소위 코딩-블록들, 전형적으로 크기가 64×64 또는 32×32 인 직사각형 블록들로 미리 분류된다. 각 코딩-블록은 그 후에 엔트로피 코딩에 의해 독립적으로 코딩된다. 따라서, 엔트로피 코더는, 그것이 현재 코딩-블록의 코딩을 착수하는 경우, 이전 코딩-블록들의 코딩 동안 연산된 심볼 발생 확률들을 이용하지 않는다. 엔트로피 코더는 따라서 코딩-블록의 코딩의 각 시작에서 초기화된 상태이다. 이런 방법은 이웃하는 코딩-블록들을 디코딩해야 할 필요 없이 코딩-블록의 데이터를 디코딩하는 장점을 나타낸다. 따라서, 예를 들면, 하나의 클라이언트 소프트웨어가 이미지의 식별된 서브-부분을 디코딩하기 위해 오직 클라이언트에 의해 서만 필요한 압축된 코딩-블록들을 제공하도록 하나의 서버 소프트웨어를 요청할 수 있다. 이런 방법은 또한 코딩-블록들의 병렬 인코딩 및/또는 디코딩을 허용하는 장점을 나타낸다. 따라서, 코딩-블록들의 크기가 작아질수록, 병렬 계산(parallelism)의 레벨은 높아진다. 예를 들면, 2로 고정된 병렬 계산의 레벨에 대하여, 2개의 코딩-블록들이 병렬로 코딩 및/또는 디코딩될 것이다. 이론적으로, 병렬 계산의 레벨의 값은 이미지의 코딩될 코드-블록들의 수와 동일하다. 그러나, 이런 방법으로 획득된 압축 성능은, 이런 코딩이 현재 코드-블록의 당면한 환경에 기인하는 확률들을 이용하지 않는다는 사실을 감안하면, 최적이지 않다.

발명의 내용

[0021] 본 발명의 목적들 중 하나는 상술한 종래 기술들의 문제점들을 해결하는 것이다.

[0022] 이를 위해, 본 발명의 주제는 다음 단계들을 포함하는 적어도 하나의 이미지를 코딩하기 위한 방법에 관한 것이다 :

[0023] - 미리 결정된 심볼들의 세트에 속하는 심볼들을 포함할 수 있도록 상기 이미지를 복수의 블록들로 세그먼트화하는 단계,

- [0024] - 블록들을 미리 결정된 수의 블록들의 서브세트들로 분류하는 단계,
- [0025] - 엔트로피 코딩 모듈에 의해, 고려중인 서브세트의 각 블록의 심볼들과 디지털 정보를 연관시킴으로써, 상기 블록들의 서브세트들의 각각을 코딩하는 단계 - 상기 코딩 단계는 상기 이미지의 첫 번째 블록에 대하여, 상기 엔트로피 코딩 모듈의 상태 변수들을 초기화하는 서브-단계를 포함함 -,
- [0026] - 상기 코딩된 블록들의 서브세트들 중 적어도 하나를 나타내는 적어도 하나의 데이터 서브-스트림을 생성하는 단계.
- [0027] 본 발명에 따른 상기 방법은 다음에 주목할 만하다 :
- [0028] - 현재 블록이 고려중인 서브세트의 코딩될 상기 첫 번째 블록인 경우, 상기 첫 번째 현재 블록에 대한 심볼 발생의 확률들의 결정이 착수되고, 상기 확률들은 적어도 하나의 다른 서브세트의 코딩된 및 디코딩된 미리 결정된 블록에 대하여 결정되었던 것이며,
- [0029] - 상기 현재 블록이 고려중인 상기 서브세트의 마지막 코딩된 블록인 경우, 다음이 착수된다 :
- [0030] ● 고려중인 상기 서브세트를 나타내는 상기 서브-스트림에, 고려중인 상기 서브세트의 블록들의 코딩 동안 상기 심볼들과 연관되었던 상기 디지털 정보의 전체의 기록,
- [0031] ● 상기 초기화 서브-단계의 실시.
- [0032] 앞서 설명된 상기 기록 단계는, 설명부에서 위에 설명되었던 바와 같이, 서브 세트의 블록들 중 마지막 블록이 코딩되자마자, 아직 전송되지 않은 디지털 정보(비트들)의 비율을 수행하는 단계에 해당한다.
- [0033] 상술한 기록 단계 및 엔트로피 코딩 모듈을 재초기화하는 단계의 결합은 적어도 하나의 코딩된 블록들의 서브세트에 각각 대응하는 다양한 데이터 서브-스트림들을 포함하는 코딩된 데이터 스트림을 생성하는 것을 가능하게 하고, 상기 스트림은 병렬 계산의 다양한 레벨들에 따라 병렬로 디코딩되도록 적응되며, 이는 블록들의 서브세트들에 적용되었던, 순차적인 또는 병렬의 코딩의 유형에 관계없다. 따라서, 큰 자유도는, 기대된 코딩/디코딩 성능의 함수로서, 병렬 계산의 레벨의 선택에 대하여 디코딩시 획득될 수 있다. 디코딩시 병렬 계산의 레벨은 가변적이고, 블록들의 서브세트의 디코딩을 시작하는 경우, 디코더는 항상 초기화된 상태이기 때문에, 코딩시 병렬 계산의 레벨과 상이할 수도 있다.
- [0034] 제 1 예에 따르면, 엔트로피 코딩 모듈의 상태 변수들은 미리 결정된 심볼들의 세트의 심볼들 중에서 심볼의 발생의 확률을 나타내는 간격의 2개의 한계이다.
- [0035] 제 2 예에 따르면, 엔트로피 코딩 모듈의 상태 변수들은 당업자에게 잘 알려진 LZW(Lempel-Ziv-Welch) 엔트로피 코더의 변환 테이블에 포함된 심볼들의 스트링들이고, 2011년 6월 21일자의 다음의 인터넷 어드레스에 설명된다 :
- [0036] <http://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Welch>.
- [0037] 고려된 블록들의 서브세트의 첫 번째 현재 블록의 엔트로피 코딩 동안 상기 다른 서브세트의 첫 번째 블록에 대하여 결정된 심볼 발생 확률들을 이용하는 주요한 장점은 상기 다른 서브세트의 다른 연속적인 블록들에 의해 학습된 심볼 발생 확률들을 고려하지 않고, 상기 심볼 발생 확률들의 업데이트 만을 버퍼 메모리에 저장함으로써 코더의 버퍼 메모리를 절약하는 것이다.
- [0038] 고려된 블록들의 서브세트의 첫 번째 현재 블록의 엔트로피 코딩 동안, 상기 다른 서브세트의 첫 번째 블록 이외의 블록, 예를 들면, 두 번째 블록에 대하여 결정된 심볼 발생 확률들을 이용하는 주요한 장점은 더 정확한 획득 및 따라서 심볼들의 발생의 확률들의 더 우수한 학습이며, 그에 의해 우수한 비디오 압축 성능을 초래한다.
- [0039] 특정 실시예에서, 상기 블록들의 서브세트들은 순차적으로 아니면 병렬로 코딩된다.
- [0040] 블록들의 서브세트들이 순차적으로 코딩된다는 사실은 H.264/MPEG-4 AVC 표준에 따르는 본 발명에 따른 코딩 방법을 제공하는 장점을 갖는다.
- [0041] 블록들의 서브세트들이 병렬로 코딩된다는 사실은 코더 프로세싱 시간을 가속화하고, 이미지의 코딩을 위한 다중 플랫폼 아키텍처로부터 이익을 얻는 장점을 갖는다.
- [0042] 다른 특정 실시예에서, 적어도 두 개의 블록들의 서브세트들이 적어도 하나의 다른 블록들의 서브세트와 병렬로

코딩되는 경우, 상기 적어도 두 개의 코딩된 블록들의 서브세트들은 동일한 데이터 서브-스트림에 포함된다.

- [0043] 이런 제공은 특히 데이터 서브-스트림들의 시그널링을 절약하는 것을 가능하게 한다. 실제로, 디코딩 유닛이 서브-스트림을 가능한 일찍 디코딩할 수 있도록, 압축된 파일에서 검토중인 서브-스트림이 시작하는 위치를 나타내는 것이 필요하다. 몇몇 블록들의 서브세트들은 동일한 데이터 서브-스트림에 포함되고, 단일 표시자(indicator)가 필요하며, 그에 의해 압축된 파일의 크기가 감소시킨다.
- [0044] 또 다른 특정 실시예에서, 상기 코딩된 블록들의 서브세트들이 미리 결정된 순서로 병렬로 디코딩되도록 의도될 때, 상기 블록들의 서브세트들의 각각을 각각 코딩한 후 전달되는 상기 데이터 서브-스트림들은 그들의 디코딩을 목적으로 전송되기 전에 상기 미리 결정된 순서에 따라 먼저 정렬된다(orderd).
- [0045] 이런 제공은 이미지를 디코딩하고 그 다음 재-인코딩해야 할 필요 없이, 코딩된 데이터 스트림을 특정한 유형의 디코딩에 적응시키는 것을 가능하게 한다.
- [0046] 상관적으로, 본 발명은 추가로 다음을 포함하는 적어도 하나의 이미지를 코딩하기 위한 디바이스에 관한 것이다 :
- [0047] - 미리 결정된 심볼들의 세트에 속하는 심볼들을 포함할 수 있도록 상기 이미지를 복수의 블록들로 세그먼트화하기 위한 수단,
- [0048] - 상기 블록들을 미리 결정된 수의 블록들의 서브세트들로 분류하기 위한 수단,
- [0049] - 상기 블록들의 서브세트들의 각각을 코딩하기 위한 수단 - 상기 코딩 수단은 고려중인 서브세트의 각 블록의 심볼들과 디지털 정보를 연관시킬 수 있도록 엔트로피 코딩 모듈을 포함하고, 상기 코딩 수단은 상기 이미지의 첫 번째 블록에 대하여, 상기 엔트로피 코딩 모듈의 상태 변수들을 초기화하기 위한 서브-수단을 포함함 -,
- [0050] - 상기 코딩된 블록들의 서브세트들 중 적어도 하나를 나타내는 적어도 하나의 데이터 서브-스트림을 생성하기 위한 수단.
- [0051] 이런 코딩 디바이스는 그것이 다음을 포함한다는 것에 주목할 만하다 :
- [0052] - 현재 블록이 고려중인 서브세트의 코딩될 상기 첫 번째 블록인 경우, 적어도 하나의 다른 서브세트의 코딩된 및 디코딩된 미리 결정된 블록에 대하여 결정되었던 것과 같이, 상기 첫 번째 블록에 대한 심볼 발생의 확률들을 결정하는, 상기 현재 블록에 대한 심볼 발생의 확률들을 결정하기 위한 수단,
- [0053] - 상기 현재 블록이 고려중인 상기 서브세트의 마지막 코딩된 블록인 경우, 고려중인 상기 서브세트를 나타내는 상기 서브-스트림에, 고려중인 상기 서브세트의 블록들의 코딩 동안 상기 심볼들과 연관되었던 상기 디지털 정보의 전체를 기록하도록 활성화되는 기록 수단을 포함하고,
- [0054] 상기 초기화 서브-수단은 또한 상기 엔트로피 코딩 모듈의 상기 상태 변수들을 재초기화하도록 활성화된다.
- [0055] 대응하는 방식으로, 본 발명은 또한 적어도 하나의 코딩된 이미지를 나타내는 스트림을 디코딩하기 위한 방법에 관한 것이고, 이는 다음을 포함한다 :
- [0056] - 디코딩될 적어도 하나의 블록들의 서브세트에 각각 대응하는 미리 결정된 수의 데이터 서브-스트림들을 상기 스트림에서 식별하는 단계 - 상기 블록들은 미리 결정된 심볼들의 세트에 속하는 심볼들을 포함할 수 있음 -,
- [0057] - 엔트로피 디코딩 모듈에 의해, 상기 식별된 서브-스트림들 중 적어도 하나에서, 상기 적어도 하나의 식별된 서브-스트림에 대응하는 서브세트의 각 블록의 심볼들과 연관된 디지털 정보를 관독함으로써, 상기 식별된 블록들의 서브세트들을 디코딩하는 단계 - 상기 디코딩 단계는 상기 이미지의 디코딩될 첫 번째 블록에 대하여, 상기 엔트로피 디코딩 모듈의 상태 변수들을 초기화하는 서브-단계를 포함함 -.
- [0058] 이런 디코딩 방법은 다음에 주목할 만하다 :
- [0059] - 현재 블록이 고려중인 서브세트의 디코딩될 상기 첫 번째 블록인 경우, 고려중인 상기 서브세트의 상기 첫 번째 블록에 대한 심볼 발생의 확률들의 결정이 착수되고, 상기 확률들은 적어도 하나의 다른 서브세트의 디코딩된 미리 결정된 블록에 대하여 결정되었던 것들이고,
- [0060] - 상기 현재 블록이 고려중인 서브세트의 마지막 디코딩된 블록인 경우, 상기 초기화 서브-단계의 실시가 착수된다.
- [0061] 특정 실시예에서, 상기 블록들의 서브세트들은 순차적으로 아니면 병렬로 디코딩된다.

- [0062] 다른 특정 실시예에서, 적어도 두 개의 블록들의 서브세트들이 적어도 하나의 다른 블록들의 서브세트와 병렬로 디코딩되는 경우, 상기 식별된 데이터 서브-스트림들 중 하나는 상기 적어도 두 개의 블록들의 서브세트들을 나타낸다.
- [0063] 또 다른 특정 실시예에서, 상기 코딩된 블록들의 서브세트들이 미리 결정된 순서로 병렬로 디코딩되도록 의도된 경우, 상기 코딩된 블록들의 서브세트들에 각각 대응하는 상기 데이터 서브-스트림들은 디코딩될 상기 스트림에서 상기 미리 결정된 순서로 미리 정렬된다.
- [0064] 상관적으로, 본 발명은 추가로 적어도 하나의 코딩된 이미지를 나타내는 스트림을 디코딩하기 위한 디바이스에 관한 것이며, 이는 다음을 포함한다 :
- [0065] - 디코딩될 적어도 하나의 블록들의 서브세트에 각각 대응하는 미리 결정된 수의 데이터 서브-스트림들을 상기 스트림에서 식별하기 위한 수단 - 상기 블록들은 미리 결정된 심볼들의 세트에 속하는 심볼들을 포함할 수 있음 - ,
- [0066] - 상기 식별된 블록들의 서브세트들을 디코딩하기 위한 수단 - 상기 디코딩 수단은 상기 식별된 서브-스트림들 중 적어도 하나에서, 상기 적어도 하나의 식별된 서브-스트림에 대응하는 서브세트의 각 블록의 심볼들과 연관된 디지털 정보를 판독할 수 있도록 엔트로피 디코딩 모듈을 포함하고, 상기 디코딩 수단은 상기 이미지의 디코딩될 첫 번째 블록에 대하여, 상기 엔트로피 디코딩 모듈의 상태 변수들을 초기화하기 위한 서브-수단을 포함함 - .
- [0067] 이런 디코딩 디바이스는, 그것이, 현재 블록이 고려중인 서브세트의 디코딩될 상기 첫 번째 블록인 경우, 적어도 하나의 다른 서브세트의 디코딩된 미리 결정된 블록에 대하여 결정되었던 것과 같이, 상기 첫 번째 블록에 대한 심볼 발생의 확률들을 결정하는, 현재 블록에 대한 심볼 발생의 확률들의 결정하기 위한 수단을 포함하고,
- [0068] 상기 현재 블록이 고려중인 서브세트의 마지막 디코딩된 블록인 경우, 상기 초기화 서브-수단은 상기 엔트로피 디코딩 모듈의 상기 상태 변수들을 재초기화하도록 활성화되는 점에 주목할 만하다.
- [0069] 본 발명은 또한 프로그램이 컴퓨터에 의해 실행되는 경우, 앞서 설명한 코딩 또는 디코딩 방법의 단계들의 실행을 위한 명령들을 포함하는 컴퓨터 프로그램을 목적으로 한다.
- [0070] 이런 프로그램은 임의의 프로그래밍 언어를 사용할 수 있고, 소스 코드, 목적 코드(object code), 또는 부분적으로 컴파일된 형태와 같은, 소스 코드와 목적 코드 사이의 중간 코드의 형태 또는 임의의 다른 바람직한 형태일 수 있다.
- [0071] 본 발명의 또 다른 주제는 또한 컴퓨터에 의해 판독 가능하고 상술한 바와 같은 컴퓨터 프로그램 명령들을 포함하는 기록 매체를 목적으로 한다.
- [0072] 기록 매체는 프로그램을 저장할 수 있는 임의의 엔티티 또는 디바이스일 수 있다. 예를 들면, 이런 매체는 ROM과 같은 저장 수단, 예를 들면, CD ROM 또는 마이크로 전자 회로 ROM, 또는 그 외의 자기 기록 수단, 예를 들면, 디스켓(플로피 디스크) 또는 하드 디스크를 포함할 수 있다.
- [0073] 게다가, 이런 기록 매체는 전기 또는 광학 케이블을 통하여, 무선에 의해, 또는 다른 수단에 의해 전달될 수 있는 전기 또는 광학 신호와 같은 전달 가능한 매체일 수 있다. 본 발명에 따른 프로그램은 특히 인터넷 유형의 네트워크로부터 다운로드될 수 있다.
- [0074] 대안적으로, 이런 기록 매체는 프로그램이 그 안에 통합되는 집적회로일 수 있고, 회로는 검토중인 방법을 실행하기 위해 또는 프로그램의 실행에 사용되도록 적용될 수 있다.
- [0075] 상술한 코딩 디바이스, 디코딩 방법, 디코딩 디바이스 및 컴퓨터 프로그램들은 본 발명에 따른 코딩 방법에 의해 부여되는 것들과 적어도 동일한 장점을 나타낸다.
- [0076] 다른 특징들 및 장점들은 도면들을 참조하여 설명된 2개의 바람직한 실시예들을 읽을 때 명백하게 될 것이다.

도면의 간단한 설명

- [0077] - 도 1은 종래 기술의 이미지 코딩 다이어그램을 도시하고,
- 도 2a는 본 발명에 따른 코딩 방법의 주요 단계들을 도시하며,

- 도 2b는 도 2a의 코딩 방법에서 실시되는 코딩을 상세하게 도시하고,
- 도 3a는 본 발명에 따른 코딩 디바이스의 제 1 실시예를 도시하며,
- 도 3b는 도 3a의 코딩 디바이스의 코딩 유닛을 도시하고,
- 도 3c는 본 발명에 따른 코딩 디바이스의 제 2 실시예를 도시하며,
- 도 4a는 제 1 바람직한 실시예에 따른 이미지 코딩/디코딩 다이어그램을 도시하고,
- 도 4b는 제 2 바람직한 실시예에 따른 이미지 코딩/디코딩 다이어그램을 도시하며,
- 도 5a는 본 발명에 따른 디코딩 방법의 주요 단계들을 도시하고,
- 도 5b는 도 5a의 디코딩 방법에서 실시되는 디코딩을 상세하게 도시하며,
- 도 6a는 본 발명에 따른 디코딩 디바이스의 실시예를 도시하고,
- 도 6b는 도 6a의 디코딩 디바이스의 디코딩 유닛을 도시하며,
- 도 7a는 순차적 유형의 코딩 및 병렬 유형의 디코딩을 실시하는 이미지 코딩/디코딩 다이어그램을 도시하고,
- 도 7b는 각각 상이한 병렬 계산의 레벨들로 병렬 유형의 코딩/디코딩을 실시하는 이미지 코딩/디코딩 다이어그램을 도시한다.

발명을 실시하기 위한 구체적인 내용

- [0078] 코딩 부분의 제 1 실시예의 상세한 설명
- [0079] 본 발명의 실시예가 이제 설명될 것이고, 여기서, 본 발명에 따른 코딩 방법은 H.264/MPEG-4 AVC 표준을 따르는 코딩에 의해 획득되는 것과 많이 유사한 이진 스트림에 따른 이미지들의 시퀀스를 코딩하도록 사용된다. 이런 실시예에서, 본 발명에 따른 코딩 방법은, 예를 들면, 초기에 H.264/MPEG-4 AVC 표준을 따르는 코더의 수정들에 의해 소프트웨어 또는 하드웨어 방식으로 실시된다. 본 발명에 따른 코딩 방법은 도 2a에 도시된 단계들(C1 내지 C5)을 포함하는 알고리즘의 형태로 도시된다.
- [0080] 본 발명의 실시예에 따르면, 본 발명에 따른 코딩 방법은 코딩 디바이스(CO)에서 실시되는데, 이들 중 2개의 실시예들이 도 3a 및 도 3c에 각각 도시된다.
- [0081] 도 2a를 참조하여, 제 1 코딩 단계(C1)는, 도 4a 또는 도 4b에 도시된 바와 같이, 코딩될 이미지들의 시퀀스의 이미지(IE)를 복수의 블록들 또는 매크로-블록들(MB)로 세그먼트화하는 단계이다. 상기 매크로-블록들은 하나 또는 둘 이상의 심볼들을 포함할 수 있고, 상기 심볼들은 미리 결정된 심볼들의 세트의 일부를 형성한다. 도시된 예들에서, 상기 블록들(MB)은 정사각형 형상을 갖고 모두가 동일한 크기를 갖는다. 반드시 블록들의 크기의 배수일 필요는 없는 이미지의 크기의 함수로서, 좌측 상의 마지막 블록들 및 하부에서 마지막 블록들은 정사각형이 아닐 수 있다. 대안적인 실시예에서, 블록들은 예를 들면, 직사각형 크기일 수 있고, 및/또는 서로 정렬되지 않을 수 있다.
- [0082] 각 블록 또는 매크로블록은 게다가 그 자체로 그것들 자체가 세분될 수 있는(subdividable) 서브-블록들로 분할될 수 있다.
- [0083] 이런 세그먼트화는, 예를 들면, 그런 잘 알려진 파티션 알고리즘을 사용하는 도 3a에 도시된 파티션(partition) 모듈(PCO)에 의해 수행된다.
- [0084] 도 2a를 참조하여, 제 2 코딩 단계(C2)는 상술한 블록들을 순차적으로 또는 병렬로 코딩되도록 의도된, 미리 결정된 수(P)의 연속하는 블록들의 서브세트들(SE1, SE2, ..., SEk, ..., SEP)로 분류하는 단계이다. 도 4a 및 도 4b에 도시된 예들에서, P=6이지만, 도면들의 명확성을 위해 단지 4개의 서브세트들(SE1, SE2, SE3, SE4)만 도시된다. 이들 4개의 블록들의 서브세트들은 각각 대시(dash) 도시되고, 이미지(IE)의 첫 번째 4개의 블록들의 열들로 각각 구성된다.
- [0085] 이런 분류는 그 자체로 잘 알려진 알고리즘으로 도 3a에 도시된 연산 모듈(GRCO)에 의해 수행된다.
- [0086] 도 2a를 참조하여, 제 3 코딩 단계(C3)는 상기 블록들의 서브세트들(SE 1 내지 SE 6)의 각각을 코딩하는 단계에 상응하고, 고려중인 서브세트의 블록들은 순회(PS)의 미리 결정된 순서에 따라 코딩되는데, 이는, 예를 들면,

순차적 유형이다. 도 4a 및 도 4b에 도시된 예들에서, 현재 서브세트($SE_k(1 \leq k \leq P)$)의 블록들은 화살표 PS에 의해 지시된 바와 같이, 좌측에서 우측으로 차례로 코딩된다.

[0087] 제 1 변형예에 따르면, 이런 코딩은 순차적인 유형이고, 도 3a에 도시된 바와 같은 단일 코딩 유닛(UC)에 의해 실시된다. 그 자체로 알려진 방식으로, 코더(CO)는 현재 블록의 코딩과 협력하여 계속적으로 재갱신되는 바와 같은 심볼 발생 확률들을 포함하도록 적응되는 버퍼 메모리(MT)를 포함한다.

[0088] 도 3b에 더 상세하게 도시된 바와 같이, 코딩 유닛(UC)은 다음을 포함한다 :

[0089] ● 적어도 하나의 이전 코딩 및 디코딩된 블록에 대하여 현재 블록을 예측 코딩하기 위한 모듈, MCP로 지시됨;

[0090] ● 상기 이전 코딩 및 디코딩된 블록들에 대하여 연산된 적어도 하나의 심볼 발생의 확률을 이용함으로써 상기 현재 블록을 엔트로피 코딩하기 위한 모듈, MCE로 지시됨.

[0091] 예측 코딩 모듈(MCP)은, 예를 들면, 인트라 및/또는 인터 모드에서와 같은, 종래의 예측 기법들에 따라 현재 블록의 예측 코딩을 수행할 수 있는 소프트웨어 모듈이다.

[0092] 엔트로피 코딩 모듈(MCE)은 CABAC 유형의 그 일부이지만, 설명부에서 더 설명될 것과 같이, 본 발명에 따라 수정된다.

[0093] 변형예로서, 엔트로피 코딩 모듈(MCE)은 그 자체로 알려진 허프만(Huffman) 코더일 수 있다.

[0094] 도 4a 및 도 4b에 도시된 예들에서, 유닛(UC)은 첫 번째 열(SE1)의 블록들을 좌측에서 우측으로 코딩한다. 그것이 첫 번째 열(SE1)의 마지막 블록에 도달하면, 그것은 두 번째 열(SE2)의 첫 번째 블록으로 이동한다. 그것이 두 번째 열(SE2)의 마지막 블록에 도달하면, 그것은 세 번째 열(SE3)의 첫 번째 블록으로 이동한다. 그것이 세 번째 열(SE3)의 마지막 블록에 도달하면, 그것은 네 번째 열(SE4)의 첫 번째 블록으로 이동하고, 이미지(IE)의 마지막 블록이 코딩될 때까지, 이런 식이다.

[0095] 앞서 방금 설명되었던 것과 다른 유형들의 순회도 물론 가능하다. 따라서, 이미지(IE)를 몇 개의 서브-이미지들로 세그먼트화하고 이런 유형의 세그먼트화를 각각의 서브-이미지에 독립적으로 적용하는 것이 가능하다. 각 코딩 유닛이, 앞서 설명된 바와 같이, 연속한 열들이 아니라 연속한 행들을 처리하는 것이 또한 가능하다. 열들 또는 행들 중 어느 방향으로든 순회하는 것이 또한 가능하다.

[0096] 제 2 변형예에 따르면, 이런 코딩은 병렬 유형이고, 단지 그것은 도 3c에 도시된 예에서 $R=2$ 인, 미리 결정된 수(R)의 코딩 유닛들($UC_k(1 \leq k \leq R)$)에 의해 실시된다는 사실에 의해서 순차적인 코딩의 제 1 변형예와 구분된다. 이런 병렬 코딩은 코딩 방법의 상당한 가속화를 야기한다고 알려진다.

[0097] 코딩 유닛들(UC_k) 각각은 도 3b에서 도시된 코딩 유닛(UC)과 동일하다. 대응하는 방식으로, 코딩 유닛(UC_k)은 예측 코딩 모듈(MCP_k) 및 엔트로피 코딩 모듈(MCE_k)을 포함한다.

[0098] 도 4a 및 도 4b를 다시 참조하여, 제 1 유닛(UC_1)은, 예를 들면, 홀수 번째의 열들의 블록들을 코딩하는 한편, 제 2 유닛(UC_2)은, 예를 들면, 짝수 번째의 열들의 블록들을 코딩한다. 더 정확하게, 제 1 유닛(UC_1)은 첫 번째 열(SE1)의 블록들을 좌측에서 우측으로 코딩한다. 그것이 첫 번째 열(SE1)의 마지막 블록에 도달하면, 그것은 $(2n+1)$ 번째 열, 즉, 세 번째 열(SE3)의 첫 번째 블록으로 이동하는 식이다. 제 1 유닛(UC_1)에 의해 수행되는 프로세싱에 병렬로, 제 2 유닛(UC_2)은 두 번째 열(SE2)의 블록들을 좌측에서 우측으로 코딩한다. 그것이 두 번째 열(SE2)의 마지막 블록에 도달하면, 그것은 $(2n)$ 번째 열, 여기서는 네 번째 열(SE4)의 첫 번째 블록으로 이동하는 식이다. 상술한 두 개의 순회들은 이미지(IE)의 마지막 블록이 코딩될 때까지, 반복된다.

[0099] 도 2a를 참조하여, 제 4 코딩 단계(C_4)는 상술한 코딩 유닛(UC) 또는 상술한 코딩 유닛들(UC_k)의 각각에 의해 압축된 처리된 블록들을 나타내는 L 개의 서브-스트림들($F_1, F_2, \dots, F_m, \dots, F_L(1 \leq m \leq L \leq P)$)의 비트들뿐만 아니라, 각 서브세트(SE_k)의 처리된 블록들의 디코딩된 버전의 생성이다. $SED_1, SED_2, \dots, SED_k, \dots, SED_P$ 로 표시된 고려중인 서브세트의 디코딩된 처리된 블록들은 설명부에서 더 상세하게 될 동기화 메커니즘에 따라 도 3a에 도시된 코딩 유닛(UC) 또는 도 3c에 도시된 코딩 유닛들(UC_k)의 각각에 의해 재사용될 수 있다.

[0100] 도 3b를 참조하여, L 개의 서브-스트림들을 생성하는 단계는 예를 들면, 비트들과 같은 데이터 스트림들을 생성하도록 적응되는 스트림 발생 소프트웨어 모듈(MGSF 또는 $MGSF_k$)에 의해 실시된다.

[0101] 도 2a를 참조하여, 제 5 코딩 단계(C_5)는 상술한 L 개의 서브-스트림들($F_1, F_2, \dots, F_m, \dots, F_L$)에 기반하여 전체(global) 스트림(F)을 구성하는 단계에 상응한다. 일 실시예에 따르면, 서브-스트림들($F_1, F_2, \dots, F_m, \dots,$

FL)은 전체 스트림(F)에서 각각의 서브-스트림(F_m)의 위치를 디코더로 나타내도록 의도된 정보의 추가 항목과 단지 병치된다(juxtapose). 전체 스트림은 그 후에 통신 네트워크(미도시)에 의해 원격 단말로 전송된다. 원격 단말은 도 5a에 도시된 디코더(D0)를 포함한다. 그것이 디코딩 및 그 다음 이미지의 재-인코딩을 필요로 하지 않기 때문에 특별히 유리한 다른 실시예에 따르면, 코더(CO)는, 스트림(F)을 디코더(D0)로 전송하기 이전에, 디코더(D0)가 서브-스트림들을 디코딩할 수 있는 순서에 대응하는 미리 결정된 순서로 L개의 서브-스트림들(F₁, F₂, ..., F_m, ..., FL)을 미리 정렬한다.

- [0102] 따라서, 설명부에서 더 상세하게 설명될 것과 같이, 본 발명에 따른 디코더는 전체 스트림(F) 내의 서브-스트림들(F₁, F₂, ..., F_m, ..., FL)을 분리시키고 그리고 디코더를 구성되는 하나 또는 둘 이상의 디코딩 유닛들로 그것들을 할당할 수 있다. 전체 스트림에서의 서브-스트림들의 이런 분해는 병렬로 동작하는 단일 코딩 유닛 또는 그 밖의 몇몇 코딩 유닛들의 사용의 선택과 무관하고, 이런 접근법에 의해 병렬로 동작하는 유닛들을 포함하는 코더만 또는 디코더만 갖는 것이 가능함에 주목될 것이다.
- [0103] 전체 스트림(F)의 이런 구성은 도 3a 및 도 3c에 도시된 바와 같은 스트림 구성 모듈(CF)에서 실시된다.
- [0104] 상술한 코딩 단계(C3) 동안, 코딩 유닛(UC 또는 UCK)에서 실시되는 바와 같은 본 발명의 다양한 특정 서브-단계들은 도 2b를 참조하여 이제 설명될 것이다.
- [0105] 단계 C31의 과정에서, 코딩 유닛(UC 또는 UCK)은, 예를 들면, 첫 번째 열(SE1)과 같은, 도 4a 또는 도 4b에 도시된 현재 열(SEK)의 코딩될 첫 번째 블록을 현재 블록으로서 선택한다.
- [0106] 단계 C32의 과정에서, 유닛(UC 또는 UCK)은 현재 블록이 상술한 단계 C1에서 블록들로 세그먼트화되었던 이미지(IE)의 첫 번째 블록(상부에서 그리고 좌측에 위치된)인지의 여부를 검사한다.
- [0107] 그런 경우이면, 단계 C33의 과정에서, 엔트로피 코딩 모듈(MCE 또는 MCEK)은 그것의 상태 변수들의 초기화를 착수한다. 이전에 설명된 산술 코딩을 이용하는 도식된 예에 따르면, 이는 미리 결정된 심볼들의 세트에 포함된 심볼의 발생의 확률을 나타내는 간격의 초기화를 수반한다. 그 자체로 알려진 방식으로, 이런 간격은 각각 아래 및 위의 2개의 한계들(L 및 H)로 초기화된다. 하한(L)의 값은 0으로 고정되는 반면, 상한의 값은 1로 고정되며, 그에 의해 미리 결정된 심볼들의 세트의 모든 심볼들 중에서 첫 번째 심볼의 발생의 확률에 대응한다. 이런 간격의 크기(R)는 따라서 $R = H - L = 1$ 에 의해 이런 시점에서 정의된다. 초기화된 간격은 관례적으로 미리 결정된 심볼들의 세트의 심볼들의 발생의 확률들을 각각 나타내는 복수의 미리 결정된 서브-간격들로 더 파티션된다(partitioned).
- [0108] 변형예로서, 사용된 엔트로피 코딩이 LZW 코딩인 경우, 심볼들의 스트림들의 변환 테이블은 초기화되고, 그래서 그것은 한번 및 단 한번만 가능한 심볼들 모두를 포함한다.
- [0109] 상술한 단계 C32 다음에, 현재 블록이 이미지(IE)의 첫 번째 블록이 아닌 경우, 다음의 설명부에서 나중에 설명될 단계 C40의 과정에서, 필요한 이전에 코딩 및 디코딩된 블록들의 이용가능성의 결정이 착수된다.
- [0110] 단계 C34의 과정에서, 도 4a 또는 도 4b에 도시된 첫 번째 열(SE1)의 첫 번째 현재 블록 MB1의 코딩이 착수된다. 이런 단계 C34는 이하에 설명될 복수의 서브-단계들 C341 내지 C348을 포함한다.
- [0111] 도 2b에서 도시된 첫 번째 서브-단계 C341의 과정에서, 블록 MB1이 적어도 하나의 이전에 코딩 및 디코딩된 블록들에 대하여 예측되는 과정에서, 현재 블록 MB1의 예측 코딩이 인트라 및/또는 인터 예측의 알려진 기법들에 의해 착수된다.
- [0112] H.246 표준에서 제안된 바와 같은 인트라 예측의 다른 모드들이 가능함은 말할 필요도 없다.
- [0113] 현재 블록 MB1은, 현재 블록이 이전에 코딩 및 디코딩된 이미지에 기인한 블록에 대하여 예측되는 과정에서, 또한 인터 모드로 예측 코딩이 또한 이루어질 수 있다. 다른 유형들의 예측을 물론 생각할 수 있다. 현재 블록에 대하여 가능한 예측들 중에서, 최적 예측이 당업자에게 잘 알려진 비트 전송률 왜곡(bitrates distortion) 기준에 따라 선택된다.
- [0114] 상기 상술한 예측 코딩 단계는 현재 블록 MB₁의 근사값인 예측된 블록 MB_{p1}을 구성하는 것을 가능하게 한다. 이런 예측 코딩과 관련된 정보는 다음에 디코더(D0)로 전송되는 스트림(F)에 기록될 것이다. 이런 정보는 특히 예측의 유형(인트라 또는 인트라), 및 적합하면, 인트라 예측의 모드, 블록 또는 매크로블록이 세분되었던 경우, 블록 또는 매크로블록의 파티션의 유형, 기준 이미지 인덱스 및 인터 예측 모드에서 사용된 변위 벡터를 포함한다. 이런 정보는 코더(CO)에 의해 압축된다.

- [0115] 다음의 서브-단계 C342의 과정에서, 잔차 블록 MB_{r1} 을 생성하기 위해 현재 블록 MB_1 에서 예측된 블록 MB_{p1} 의 값이 착수된다.
- [0116] 다음의 서브-단계 C343의 과정에서, 변환된 블록 MB_{t1} 을 생성하기 위해, 예를 들면, DCT 유형의 이산 코사인 변환과 같은 종래의 직접 변환 동작에 따라 잔차 블록 MB_{r1} 의 변환이 착수된다.
- [0117] 다음의 서브-단계 C344의 과정에서, 예를 들면, 스칼라 양자화와 같은 종래의 양자화 동작에 따라 변환된 블록 MB_{t1} 의 양자화가 착수된다. 양자화된 계수들의 블록 MB_{q1} 이 그 다음 획득된다.
- [0118] 다음의 서브-단계 C345의 과정에서, 양자화된 계수들의 블록 MB_{q1} 의 엔트로피 코딩이 착수된다. 바람직한 실시예에서, 이는 CABAC 엔트로피 코딩을 수반한다. 이런 단계는,
- [0119] a) 상기 현재 블록과 연관되는 미리 결정된 심볼들의 세트의 심볼 또는 심볼들을 판독하고,
- [0120] b) 비트들과 같은 디지털 정보를 판독된 심볼(들)과 연관하는 것에 상응한다.
- [0121] 사용된 코딩이 LZW 코딩인 상술한 변형예에서, 현재 변환 테이블에서 심볼의 코드에 대응하는 정보의 디지털 항목이 코딩될 심볼들과 연관되고, 변환 테이블의 갱신은 그 자체로 알려진 절차를 따라 수행된다.
- [0122] 다음의 서브-단계 C346의 과정에서, 블록 MB_{q1} 의 역양자화(dequantization)가 종래의 역양자화 동작에 따라 착수되는데, 이는 단계 C344에서 수행된 양자화의 반대 동작이다. 역양자화된 계수들의 블록 MB_{dq1} 이 그 다음 획득된다.
- [0123] 다음의 서브-단계 C347의 과정에서, 위의 단계 C343에서 수행된 직접 변환의 반대의 동작인 역양자화된 계수들의 블록 MB_{dq1} 의 역변환이 착수된다. 디코딩된 잔차 블록 MB_{Dr1} 이 그 다음 획득된다.
- [0124] 다음의 서브-단계 C348의 과정에서, 예측된 블록 MB_{p1} 를 디코딩된 잔차 블록 MB_{Dr1} 에 가산함으로써 디코딩된 블록 MB_{D1} 의 구성이 착수된다. 디코딩된 잔차 블록은 설명부에서 더 상세하게 설명될 이미지(IE)를 디코딩하는 방법의 종료시 획득된 디코딩된 블록과 동일한 것임에 유의해야 한다. 디코딩된 블록 MB_{D1} 은 따라서 코딩 유닛(Uck) 또는 미리 결정된 수(R)의 코딩 유닛들 중 일부를 형성하는 임의의 다른 코딩 유닛에 의해 사용되도록 이용가능하게 제공된다.
- [0125] 상술한 코딩 단계 C34의 종료시, 도 3b에 도시된 바와 같은 엔트로피 코딩 모듈(MCE 또는 MCE_k)은 첫 번째 블록의 코딩과 협력하여 계속하여 재갱신된 바와 같은 확률들 모두를 포함한다. 이들 확률들은 가능한 구문들(syntaxes)의 다양한 성분들 및 다양한 관련된 코딩 컨텍스트들에 대응한다.
- [0126] 상술한 코딩 단계 C34 다음에, 단계 C35의 과정에서, 현재 블록이 이런 동일 열의 j 번째 블록인지의 여부를 결정하기 위한 검사가 수행되는데, 여기서 j는 적어도 1과 같은 코더(CO)에 알려진 미리 결정된 값이다.
- [0127] 그런 경우이면, 도 2b에 도시된 단계 C36의 과정에서, j 번째 블록에 대하여 연산된 확률들의 세트가 도 3a 또는 도 3b 및 도 4a 및 도 4b에 도시된 바와 같은 코더(CO)의 버퍼 메모리(MT)에 저장되고, 상기 메모리의 크기는 연산된 수의 확률들을 저장하도록 적응된다.
- [0128] 도 2b에 도시된 단계 C37의 과정에서, 코딩 유닛(UC 또는 UC_k)은 방금 코딩되었던 열(SE_k)의 현재 블록이 이미지(IE)의 마지막 블록인지의 여부를 검사한다. 이런 단계는 또한 단계 C35의 과정에서 현재 블록이 열(SE_1)의 j 번째 블록이 아닌 경우 실시된다.
- [0129] 현재 블록이 이미지(IE)의 마지막 블록인 경우, 단계 C38의 과정에서, 코딩 방법은 종료된다.
- [0130] 이런 경우가 아니면, 단계 C39의 과정에서, 도 4a 또는 도 4b에서의 화살표 PS에 의해 도시된 순회의 순서에 따라 코딩될 다음 블록 MB_i 의 선택이 착수된다.
- [0131] 도 2b에 도시된 단계 C40의 과정에서, 현재 블록 MB_i 을 코딩하기 위해 필요한 이전에 코딩 및 디코딩된 블록들의 이용가능성의 결정이 착수된다.
- [0132] 이것이 첫 번째 열(SE_1)인 경우, 이런 단계는 코딩될 현재 블록 MB_i 의 좌측에 위치된 적어도 하나의 블록의 이용가능성을 확인하는 단계에 상응한다. 그러나, 도 4a 또는 도 4b에 도시된 실시예에서 선택된 순회 PS의 순서

를 고려하면, 블록들은 고려중인 열(SE_k)에서 차례로 코딩된다. 그 결과, 좌측의 코딩 및 디코딩된 블록은 항상 이용가능하다(열의 첫 번째 블록을 제외하고). 도 4a 또는 도 4b에 도시된 예에서, 이는 코딩될 현재 블록의 바로 좌측에 위치한 블록이다.

- [0133] 이것이 첫 번째 열과 상이한 열(SE_k)인 경우, 상기 결정 단계는 더욱이, 이전 열(SE_{k-1})에 위치한 미리 결정된 수 (N')의 블록들, 예를 들면, 현재 블록의 위 및 우측 위에 각각 위치한 2개의 블록들이 현재 블록의 코딩을 위해 이용가능한지의 여부, 즉, 그것들이 코딩 유닛(UC 또는 UC_{k-1})에 의해 이미 코딩되었고 그 다음 디코딩되었는지의 여부를 확인하는 단계에 상응한다.
- [0134] 이런 검사 단계가 코딩 방법을 느리게 하는 경향이 있기 때문에, 본 발명에 따른 대안적인 방식으로, 열들의 코딩이 병렬 유형인 경우, 도 3c에 도시된 클럭(CLK)이, 현재 블록의 위 및 우측 위 각각에 위치한 2개의 블록들의 이용가능성을 확인할 필요없이, 이들 2개의 블록들의 이용가능성을 보장하도록 블록들의 코딩의 진행을 동기화하도록 적용된다. 따라서, 코딩 유닛(UCK)은, 현재 블록의 코딩을 위해 사용되는 이전 열(SE_{k-1})의 미리 결정된 수 N'(예를 들면, N'=2)개의 코딩 및 디코딩된 블록들의 시프트로 항상 첫 번째 블록을 코딩하도록 개시한다. 소프트웨어 관점에서, 이런 클럭의 실시는 코더(CO)에서 이미지(IE)의 블록들을 처리하기 위한 시간을 현저하게 가속하는 것을 가능하게 한다.
- [0135] 도 2b에 도시된 단계 C41의 과정에서, 현재 블록이 고려중인 열(SE_k)의 첫 번째 블록인지의 여부를 결정하기 위한 검사가 수행된다.
- [0136] 그런 경우이면, 단계 C42의 과정에서, 이전 열(SE_{k-1})의 j 번째 블록의 코딩 동안 연산된 심볼 발생 확률들만의 버퍼 메모리(MT)에서의 관독이 착수된다.
- [0137] 도 4a에 도시된 제 1 변형예에 따르면, j 번째 블록은 이전 열(SE_{k-1})의 첫 번째 블록(j=1)이다. 이런 관독은 CABAC 코더의 확률들을 버퍼 메모리(MT)에 존재하는 것으로 대체하는 것에 상응한다. 메모리가 두 번째, 세 번째 및 네 번째 열들(SE2, SE3 및 SE4)의 첫 번째 각 블록들을 수반하기 때문에, 이런 관독 단계는 얇은 선들에 의해 표시된 화살표들에 의해 도 4a에 도시된다.
- [0138] 도 4b에 도시되는 상술한 단계 C43의 제 2 변형예에 따르면, j 번째 블록은 이전 열(SE_{k-1})의 두 번째 블록(j=2)이다. 이런 관독은 CABAC 코더의 확률들을 버퍼 메모리(MT)에 존재하는 것으로 대체하는 것에 상응한다. 메모리가 두 번째, 세 번째 및 네 번째 열들(SE2, SE3 및 SE4)의 첫 번째 각 블록들을 수반하기 때문에, 이런 관독 단계는 얇은 대시 선들에 의해 표시된 화살표들에 의해 도 4b에 도시된다.
- [0139] 단계 C42의 다음에, 현재 블록은 위에서 설명된 단계들 C34 내지 C38의 반복에 의해 코딩되고 그 다음 디코딩된다.
- [0140] 상술한 단계 C41의 다음에, 현재 블록이 고려중인 열(SE_k)의 첫 번째 블록이 아니면, 동일 열(SE_k)에 위치되는 이전에 코딩 및 디코딩된 블록, 즉, 도시된 예에서 현재 블록의 바로 좌측에 위치한 코딩 및 디코딩된 블록에 기인하는 확률들의 관독은 유리하게 착수되지 않는다. 실제로, 도 4a 또는 도 4b에 도시된 바와 같이, 동일 열에 위치한 블록들의 관독을 위한 순차적 순회 PS를 고려하면, 현재의 블록의 코딩이 시작하는 순간에 CABAC 코더에 존재하는 심볼 발생 확률들은 틀림없이 이런 동일 열의 이전 블록의 코딩/디코딩 이후에 존재하는 것들이다.
- [0141] 그 결과, 도 2b에 도시된 단계 C43의 과정에서, 상기 현재 블록의 엔트로피 코딩에 대한 심볼 발생의 확률들의 학습이 착수되는데, 이런 확률들은 도 4a 또는 도 4b의 이중 실선 화살표들에 의해 도시된 바와 같이, 동일 열의 상기 이전 블록에 대하여 연산되었던 것에만 대응한다.
- [0142] 단계 C43의 다음에, 현재 블록은 앞서 설명된 단계들 C34 내지 C38의 반복에 의해 코딩되고 그 다음 디코딩된다.
- [0143] 단계 C44의 과정에서, 현재 블록이 고려중인 열(SE_k)의 마지막 블록인지의 여부를 결정하기 위한 검사가 그 다음에 수행된다.
- [0144] 이것이 그 경우가 아니면, 단계 C44 다음에, 코딩될 다음 블록 MB_i를 선택하는 단계 C39가 다시 실시된다.
- [0145] 현재 블록이 고려중인 열(SE_k)의 마지막 블록인 경우, 단계 C45의 과정에서, 도 3a 또는 도 3c의 코딩 디바이스(CO)는 설명부에서 상술한 바와 같이 비움을 수행한다. 이런 목적을 위해, 모듈(MGSFk)이 고려중인 상기 열(SE_k)의 코딩된 블록들을 나타내는 이전 트레인을 포함하는 데이터 서브-스트림(F_m)에, 고려중인 상기 열(SE_k)

의 각 블록의 코딩 동안 판독된 심볼(들)과 연관되었던 비트들의 전체를 기록하는 방식으로, 상기 코딩 유닛(Uck)은 대응하는 서브-스트림 생성 모듈(MGSFk)로, 비트들의 전체를 전송한다. 이런 비율은 도 4a 및 도 4b에서 각 열(SEk)의 끝에 삼각형으로 기호화된다.

[0146] 도 2b에 도시된 단계 C46의 과정에서, 코딩 유닛(UC 또는 Uck)은 상술한 단계 C33과 동일한 단계를 수행하고, 즉, 미리 결정된 심볼들의 세트에 포함된 심볼의 발생의 확률을 나타내는 간격을 다시 초기화한다. 이런 재초기화는 도 4a 및 도 4b에서 각 열(SEk)의 시작에서 검정 점으로 도시된다.

[0147] 단계들 C45 및 C46을 코딩의 이런 레벨로 수행하는 이점은 코딩 유닛(UC) 또는 코딩 유닛(Uck)에 의해 처리되는 다음 블록의 코딩 동안, 코더(CO)가 초기화된 상태라는 것이다. 따라서, 설명부에서 더 설명될 것과 같이, 그것이 초기화된 상태에 있게 되기에 충분하기 때문에, 병렬로 작동하는 디코딩 유닛이 이런 점으로부터 압축된 스트림(F)을 직접 디코딩하는 것을 가능하게 한다.

[0148] 디코딩 부분의 실시예의 상세한 설명

[0149] 본 발명에 따른 디코딩 방법의 실시예가 이제 설명될 것이고, 여기서, 디코딩 방법은 초기에 H.264/MPEG-4 AVC 표준을 따르는 디코더의 수정예들에 의해 소프트웨어 또는 하드웨어 방식으로 실시된다.

[0150] 본 발명에 따른 디코딩 방법은 도 5a에 도시된 단계들(D1 내지 D4)을 포함하는 알고리즘의 형태로 도시된다.

[0151] 본 발명의 실시예들에 따르면, 본 발명에 따른 디코딩 방법은 도 6a에 도시된 디코딩 디바이스(DO)에서 실시된다.

[0152] 도 5a를 참조하여, 제 1 디코딩 단계(D1)는, 도 4a 또는 도 4b에 도시된 바와 같이, 이전에 코딩된 블록들 또는 매크로블록 MB의 P개의 서브세트들(SE1, SE2, ..., SEk, ..., SEP) 각각을 포함하는 L개의 서브-스트림들(F1, F2, ..., Fm, ..., FL)의 상기 스트림(F)에서의 식별이다. 이런 목적을 위해, 스트림(F)의 각 서브-스트림(Fm)은 디코더(DO)가 스트림(F)의 각각의 서브-스트림(Fm)의 위치를 결정하게 하도록 의도된 표시자와 연관된다. 변형예로서, 상술한 코딩 단계 C3의 종료시, 코더(CO)는 스트림(F)에서 서브-스트림들(F1, F2, ..., Fm, ..., FL)을 디코더(DO)에 의해 기대된 순서로 정렬하고, 그에 의해 서브-스트림 표시자들의 스트림(F) 내로의 삽입을 방지한다. 이런 제공은 따라서 데이터 스트림(F)의 비트 전송률의 면에서 비용을 감소하는 것을 가능하게 한다.

[0153] 도 4a 또는 도 4b에 도시된 예에서, 상기 블록들 MB는 정사각형 형상을 갖고, 모두가 동일 크기를 갖는다. 반드시 블록들의 크기의 배수일 필요가 없는 이미지의 크기에 따라, 좌측 상의 마지막 블록들 및 하부에서의 마지막 블록들은 정사각형이 아닐 수 있다. 대안적인 실시예에서, 블록들은 예를 들면, 직사각형 크기일 수 있고 및/또는 서로 정렬되지 않을 수 있다.

[0154] 각 블록 또는 매크로블록은 게다가 그 자체로 그것들 자체가 세분될 수 있는 서브-블록들로 분할될 수 있다.

[0155] 이런 식별은 도 6a에 도시된 바와 같은 스트림 추출 모듈(EXDO)에 의해 수행된다.

[0156] 도 4a 또는 도 4b에 도시된 예에서, 미리 결정된 수 P는 6과 같지만, 도면들의 명료성을 위해, 단지 4개의 서브세트들(SE1, SE2, SE3, SE4)만이 대시로 도시된다.

[0157] 도 5a를 참조하여, 제 2 디코딩 단계(D2)는 블록들의 상기 서브세트들(SE1, SE2, SE3 및 SE4)의 각각을 디코딩하는 단계이고, 고려중인 서브세트의 블록들은 순회 PS의 미리 결정된 순차적 순서에 따라 코딩된다. 도 4a 또는 도 4b에 도시된 예들에서, 현재 서브세트(SEk(1 ≤ k ≤ P))의 블록들은, 화살표 PS에 의해 지시된 바와 같이, 좌측에서 우측으로 차례로 디코딩된다. 단계 D2의 종료시, 디코딩된 블록들의 서브세트들(SED1, SED2, SED3, ..., SEDk, ..., SEDP)이 획득된다.

[0158] 이런 디코딩은 순차적 유형일 수 있고, 그 결과, 단일 디코딩 유닛으로 수행될 수 있다.

[0159] 그러나, 다중플랫폼 디코딩 아키텍처로부터 이익을 얻을 수 있도록, 블록들의 서브세트들의 디코딩은 병렬 유형이고, 예를 들면, 도 6a에 도시된 바와 같이, R=4인 다수(R)의 디코딩 유닛들(UDk(1 ≤ k ≤ R))에 의해 실시된다. 이런 제공은 따라서 디코딩 방법의 상당한 가속을 가능하게 한다. 그 자체로 알려진 방식으로, 디코더(DO)는 현재 블록의 디코딩과 협력하여 계속적으로 재갱신되는 바와 같은 심볼 발생 확률들을 포함하도록 적응되는 버퍼 메모리(MT)를 포함한다.

[0160] 도 6b에 더 상세하게 도시된 바와 같이, 디코딩 유닛들(UDk)의 각각은 다음을 포함한다 :

[0161] ● 적어도 하나의 이전 디코딩된 블록에 대하여 연산된 적어도 하나의 심볼 발생의 확률을 학습함으로써 상기

현재 블록을 엔트로피 디코딩하기 위한 모듈, MDEk로 지시됨,

- [0162] ● 상기 이전 디코딩된 블록들에 대하여 현재 블록의 예측 디코딩을 하기 위한 모듈, MDPk로 지시됨.
- [0163] 예측 디코딩 모듈(SUDPk)은, 예를 들면, 인트라 및/또는 인터 모드와 같은 종래의 예측 기법들에 따라, 현재 블록의 예측 디코딩을 수행할 수 있다.
- [0164] 엔트로피 디코딩 모듈(MDEk)은 CABAC 유형의 그 일부이지만, 설명부에서 더 설명될 것과 같이, 본 발명에 따라 수정된다.
- [0165] 변형예로서, 엔트로피 디코더 모듈(MDEk)은 그 자체로 알려진 허프만 디코더일 수 있다.
- [0166] 도 4a 또는 도 4b에 도시된 예들에서, 제 1 유닛(UD1)은 첫 번째 열(SE1)의 블록들을 좌측에서 우측으로 디코딩한다. 그것이 첫 번째 열(SE1)의 마지막 블록에 도달하면, 그것은 (N+1) 번째 열, 여기서 5 번째 열의 첫 번째 블록으로 이동하는 식이다. 제 2 유닛(UC2)은 두 번째 열(SE2)의 블록들을 좌측에서 우측으로 디코딩한다. 그것이 두 번째 열(SE2)의 마지막 블록에 도달하면, 그것은 (N+2) 번째 열, 여기서 6 번째 열의 첫 번째 블록으로 이동하는 식이다. 이런 순회는 네 번째 열(SE4)의 블록들을 좌측에서 우측으로 디코딩하는 유닛(UD4)까지 반복된다. 그것이 첫 번째 열의 마지막 블록에 도달하면, 그것은 (N+4) 번째 열, 여기서 8 번째 열의 첫 번째 블록으로 이동하며, 마지막 식별된 서브-스트림의 마지막 블록이 디코딩될 때까지, 이런 식이다.
- [0167] 앞서 방금 설명되었던 것과 다른 유형들의 순회도 물론 가능하다. 예를 들면, 각 디코딩 유닛이, 앞서 설명된 바와 같이, 네스티드(nested) 열들이 아니라 네스티드 행들을 처리할 수 있다. 열들 또는 행들 중 어느 하나의 방향으로 순회하는 것이 또한 가능하다.
- [0168] 도 5a를 참조하여, 제 3 디코딩 단계(D3)는 디코딩 단계 D2에서 획득된 각 디코딩된 서브세트(SED1, SED2, ..., SEDk, ..., SEDP)에 기반하여 디코딩된 이미지(ID)의 재구성이다. 보다 정확하게는, 각 디코딩된 서브세트(SED1, SED2, ..., SEDk, ..., SEDP)의 디코딩된 블록들은 도 6a에 도시된 바와 같은 이미지 재구성 유닛(URI)으로 전송된다. 이런 단계 D3의 과정에서, 유닛(URI)은 디코딩된 블록들을, 이런 블록들이 이용 가능하게 되면, 디코딩된 이미지에 기록한다.
- [0169] 도 5a에 도시된 제 4 디코딩 단계(D4)의 과정에서, 완전히 디코딩된 이미지(ID)는 도 6a에 도시된 유닛(URI)에 의해 전달된다.
- [0170] 상술한 병렬 디코딩의 단계 D2 동안, 디코딩 유닛(UDk)에서 실시되는 바와 같은 본 발명의 다양한 특정 서브-단계들은 도 5b를 참조하여 이제 설명될 것이다.
- [0171] 단계 D21의 과정에서, 디코딩 유닛(UDk)은 도 4a 또는 도 4b에 도시된 현재 열(SEk)의 디코딩될 첫 번째 블록을 현재 블록으로서 선택한다.
- [0172] 단계 D22의 과정에서, 디코딩 유닛(UDk)은 현재 블록이 디코딩된 이미지의 첫 번째 블록, 본 예에서 서브-스트림(F1)의 첫 번째 블록인지의 여부를 검사한다.
- [0173] 그런 경우이면, 단계 D23의 과정에서, 엔트로피 디코딩 모듈(MDE 또는 MDEk)은 그의 상태 변수들의 초기화를 착수한다. 도시된 예에 따르면, 이는 미리 결정된 심볼들의 세트에 포함된 심볼의 발생의 확률을 나타내는 간격의 초기화를 수반한다.
- [0174] 변형예로서, 사용된 엔트로피 디코딩이 LZW 디코딩이면, 심볼들의 스트림들의 변환 테이블은 초기화되고, 그래서 그것은 한번 및 한번만 가능한 심볼들 모두를 포함한다. 단계 D23이 상술한 코딩 단계 C33과 동일하기 때문에, 그것은 다음에 설명되지 않을 것이다.
- [0175] 상술한 D22 다음에, 현재 블록이 디코딩된 이미지(ID)의 첫 번째 블록이 아니면, 다음의 설명부에서 나중에 설명될 단계 D30의 과정에서, 필요한 이전에 디코딩된 블록들의 이용가능성의 결정이 착수된다.
- [0176] 단계 D24의 과정에서, 도 4a 또는 도 4b에 도시된 첫 번째 열(SE1)의 첫 번째 현재 블록 MB1의 디코딩이 착수된다. 이런 단계 D24는 이하에 설명될 복수의 서브-단계들 D241 내지 D246을 포함한다.
- [0177] 첫 번째 서브-단계 D241의 과정에서, 현재 블록과 관련된 구문 성분들의 엔트로피 디코딩이 착수된다. 이런 단계는 주로 다음에 상응한다 :
- [0178] a) 상기 첫 번째 열(SE1)과 연관된 서브-스트림에 포함된 비트들을 판독,

- [0179] b) 판독된 비트들에 기반하여 심볼들을 재구성.
- [0180] 사용된 디코딩이 LZW 디코딩인 상술한 변형예에서, 현재 변환 테이블에서 심볼의 코드에 대응하는 정보의 디지털 항목이 판독되고, 그 자체로 알려진 절차에 따라, 심볼은 판독된 코드에 기반하여 재구성되며, 변환 테이블의 갱신이 수행된다.
- [0181] 보다 정확하게는, 현재 블록과 관련된 구문 성분들은 도 6b에 도시된 바와 같은 CABAC 엔트로피 디코딩 모듈(MDE1)에 의해 디코딩된다. 이 엔트로피 디코딩 모듈은 구문 성분들을 생성하기 위해 압축된 파일의 비트들의 서브-스트림(F1)을 디코딩하고, 동시에, 이런 모듈이 심볼을 디코딩할 때, 이런 심볼의 발생의 확률들이, 상술한 엔트로피 코딩 단계 C345 동안, 이런 동일한 심볼의 코딩 동안 획득된 것과 동일하게 되도록 하는 방법으로 그것의 확률들을 재갱신한다.
- [0182] 다음의 서브-단계 D242의 과정에서, 블록 MB1이 적어도 하나의 이전에 디코딩된 블록에 대하여 예측되는 과정에서, 현재 블록 MB1의 예측 디코딩이 인트라 및/또는 인터 예측의 알려진 기법들에 의해 착수된다.
- [0183] H.264 표준에서 제안된 바와 같은 인트라 예측의 다른 모드들이 가능함은 말할 필요도 없다.
- [0184] 이런 단계의 과정에서, 예측 디코딩은 특히 예측의 유형(인트라 또는 인터), 및 적합하면, 인트라 예측의 모드, 블록 또는 매크로블록이 세분되었던 경우, 블록 또는 매크로블록의 파티션의 유형, 기준 이미지 인덱스 및 인터 예측 모드에서 사용된 변위 벡터를 포함하며, 이전 단계에서 디코딩된 구문 성분들로 수행된다.
- [0185] 상기 상술한 예측 디코딩 단계는 예측된 블록 MB_{p1}을 구성하는 것을 가능하게 한다.
- [0186] 다음의 서브-단계 D243의 과정에서, 이전에 디코딩된 구문 성분들로 양자화된 잔차 블록 MB_{q1}의 구성이 착수된다.
- [0187] 다음의 서브-단계 D244의 과정에서, 디코딩된 역양자화된 블록 MDB_{t1}을 생성하기 위해, 상술한 단계 C344에서 수행된 양자화의 반대 동작인 종래의 역양자화 동작에 따라 양자화된 잔차 블록 MB_{q1}의 역양자화가 착수된다.
- [0188] 다음의 서브-단계 D245의 과정에서, 앞서 단계 C343에서 수행된 직접 변환의 반대의 동작인 역양자화된 블록 MDB_{t1}의 역변환이 착수된다. 디코딩된 잔차 블록 MBDr₁이 그 다음 획득된다.
- [0189] 다음의 서브-단계 D246의 과정에서, 예측된 블록 MB_{p1}에 디코딩된 잔차 블록 MBDr₁을 가산함으로써 디코딩된 블록 MBD₁의 구성이 착수된다. 디코딩된 블록 MBD₁은 따라서 디코딩 유닛(UD1) 또는 미리 결정된 수(N)의 디코딩 유닛들 중 일부를 형성하는 임의의 다른 디코딩 유닛에 의해 사용되도록 이용가능하게 제공된다.
- [0190] 상술한 디코딩 단계 D246의 종료시, 도 6b에 도시된 바와 같은 엔트로피 디코딩 모듈(MDE1)은 첫 번째 블록의 디코딩과 협력하여 계속하여 재갱신된 바와 같은 확률들 모두를 포함한다. 이들 확률들은 가능한 구문의 다양한 성분들 및 다양한 관련된 디코딩 콘텍스트들에 대응한다.
- [0191] 상술한 디코딩 단계 D24 다음에, 단계 D25의 과정에서, 현재 블록이 이런 동일 열의 j 번째 블록인지의 여부를 결정하기 위한 검사가 수행되는데, 여기서 j는 적어도 1과 같은 디코더(DO)에 알려진 미리 결정된 값이다.
- [0192] 그런 경우이면, 단계 D26의 과정에서, j 번째 블록에 대하여 연산된 확률들의 세트는 도 6a 및 도 4a 또는 도 4b에 도시된 바와 같은 디코더(DO)의 버퍼 메모리(MT)에 저장되고, 상기 메모리의 크기는 연산된 수의 확률들을 저장하도록 적응된다.
- [0193] 단계 D27의 과정에서, 유닛(UDk)은 방금 디코딩되었던 현재 블록이 마지막 서브-스트림의 마지막 블록인지의 여부를 검사한다.
- [0194] 그런 경우이면, 단계 D28의 과정에서, 디코딩 방법은 종료된다.
- [0195] 그런 경우가 아니면, 단계 D29의 과정에서, 도 4a 또는 도 4b의 화살표 PS에 의해 도시된 순회의 순서에 따라 디코딩될 다음 블록 MB_i의 선택이 착수된다.
- [0196] 상술한 단계 D25의 과정에서, 현재 블록이 고려중인 열(SEDk)의 j 번째 블록이 아니면, 위의 단계 D27이 착수된다.
- [0197] 상술한 단계 D29에 뒤이은 단계 D30의 과정에서, 현재 블록 MB_i를 디코딩하기 위해 필요한 이전에 디코딩된 블

록들의 이용가능성의 결정이 착수된다. 이는 상이한 디코딩 유닛들(UDk)에 의한 블록들의 병렬 디코딩을 수반한다는 사실을 고려하면, 이들 블록들은 이들 블록들의 디코딩에 할당된 디코딩 유닛에 의해 디코딩되지 않았고 그것들이 따라서 아직 이용가능하지 않는 상황이 발생할 수 있다. 상기 결정 단계는 이전 열(SEk-1)에 위치한 미리 결정된 수 N'개의 블록들, 예를 들면, 현재 블록의 위 및 우측 위에 각각 위치한 2개의 블록들이 현재 블록의 디코딩을 위해 이용가능한지의 여부, 즉, 그것들이 그것들의 디코딩에 할당된 디코딩 유닛(UDk-1)에 의해 이미 디코딩되었는지의 여부를 확인하는 단계에 상응한다. 상기 결정 단계는 또한 디코딩될 현재 블록 MB_i의 좌측에 위치한 적어도 하나의 블록의 이용가능성을 확인하는 단계에 상응한다. 그러나, 도 4a 또는 도 4b에 도시된 실시예에서 선택된 순회 PS의 순서를 고려하면, 블록들은 고려중인 열(SEk)에서 차례로 디코딩된다. 그 결과, 좌측의 디코딩된 블록은 항상 이용가능하다(열의 첫 번째 블록을 제외하고). 도 4a 또는 도 4b에 도시된 예에서, 이는 디코딩될 현재 블록의 바로 좌측에 위치한 블록을 수반한다. 이런 목적을 위해, 현재 블록의 위 및 우측 위에 각각 위치한 2개의 블록들의 이용가능성만이 검사된다.

[0198] 이런 검사 단계가 디코딩 방법을 느리게 하는 경향이 있기 때문에, 본 발명에 따른 대안적인 방식으로, 도 6a에 도시된 클럭(CLK)은, 현재 블록의 위 및 우측 위 각각에 위치한 2개의 블록들의 이용가능성을 확인할 필요없이, 이들 2개의 블록들의 이용가능성을 보장하도록 블록들의 디코딩의 진행을 동기화하도록 적용된다. 따라서, 도 4a 또는 4b에 도시된 바와 같이, 디코딩 유닛(UDk)은 항상 첫 번째 블록을 현재 블록의 디코딩을 위해 사용되는 이전 열(SEk-1)의 미리 결정된 수(N')(여기서 N'=2)의 디코딩된 블록들의 시프트를 갖고 디코딩하기 시작한다. 소프트웨어 관점에서, 이런 클럭의 실시는 디코더(D0)에서 각 서브세트(SEk)의 블록들을 처리하는 시간을 현재 하게 가속하는 것을 가능하게 한다.

[0199] 단계 D31의 과정에서, 현재 블록이 고려중인 열(SEk)의 첫 번째 블록인지의 여부를 결정하기 위한 검사가 수행된다.

[0200] 그런 경우이면, 단계 D32의 과정에서, 이전 열(SEk-1)의 j 번째 블록의 디코딩 동안 연산된 심볼 발생 확률들만의 버퍼 메모리(MT)에서의 판독이 착수된다.

[0201] 도 4a에 도시된 제 1 변형예에 따르면, j 번째 블록은 이전 열(SEk-1)의 첫 번째 블록(j=1)이다. 이런 판독은 CABAC 디코더의 확률들을 버퍼 메모리(MT)에 존재하는 것들로 대체하는 것에 상응한다. 이것이 두 번째, 세 번째 및 네 번째 열들(SE2, SE3 및 SE4)의 각각의 첫 번째 블록들을 수반하기 때문에, 이런 판독 단계는 얇은 선들에 의해 도시된 화살표들에 의해 도 4a에 도시된다.

[0202] 도 4b에 도시되는 상술한 단계 D32의 제 2 변형예에 따르면, j 번째 블록은 이전 열(SEk-1)의 두 번째 블록(j=2)이다. 이런 판독은 CABAC 디코더의 확률들을 버퍼 메모리(MT)에 존재하는 것들로 대체하는 것에 상응한다. 이것이 두 번째, 세 번째 및 네 번째 열들(SE2, SE3 및 SE4)의 각각의 첫 번째 블록들을 수반하기 때문에, 이런 판독 단계는 얇은 대시 선들에 의해 도시된 화살표들에 의해 도 4b에 도시된다.

[0203] 단계 D32의 다음에, 현재 블록이 위에서 설명된 단계들 D24 내지 D28의 반복에 의해 디코딩된다.

[0204] 상술한 단계 D31의 다음에, 현재 블록이 고려중인 열(SEk)의 첫 번째 블록이 아닌 경우, 동일 열(SEk)에 위치되는 이전에 디코딩된 블록, 즉, 도시된 예에서 현재 블록의 바로 좌측에 위치한 디코딩된 블록에 기인하는 확률들의 판독은 유리하게는, 착수되지 않는다. 실제로, 도 4a 또는 도 4b에 도시된 바와 같이, 동일 열에 위치한 블록들의 판독을 위한 순차적 순회 PS를 고려하면, CABAC 디코더에 존재하는 심볼 발생 확률들은 현재의 블록의 코딩이 시작하는 순간에, 틀림없이 이런 동일 열의 이전 블록의 디코딩 이후에 존재하는 것이다.

[0205] 그 결과, 단계 D33의 과정에서, 상기 현재 블록의 엔트로피 디코딩에 대한 심볼 발생의 확률들의 학습이 착수되는데, 상기 확률들은 도 4a 또는 도 4b의 이중 실선 화살표들에 의해 도시된 바와 같이, 동일 열에서의 상기 이전 블록에 대하여 연산되었던 것에만 대응한다.

[0206] 단계 D33 다음에, 현재 블록이 위에 설명된 단계 D24 내지 D28의 반복에 의해 디코딩된다.

[0207] 단계 D34의 과정에서, 현재 블록이 고려중인 열(SEk)의 마지막 블록인지의 여부를 결정하기 위한 검사가 그 후에 수행된다.

[0208] 그런 경우가 아니면, 단계 D34의 다음에, 코딩될 다음 블록 MB_i를 선택하는 단계 D29가 다시 실시된다.

[0209] 현재 블록이 고려중인 열(SEk)의 마지막 블록인 경우, 단계 D35의 과정에서, 디코딩 유닛(UDk)은 상술한 단계 D23과 동일한 단계를 수행하는데, 즉, 미리 결정된 심볼들의 세트에 포함된 심볼의 발생의 확률을 나타내는 간

격을 다시 초기화한다. 이런 재초기화는 각 열(SE_k)의 시작에서 검정 점에 의해 도 4a 및 도 4b에 도시된다.

[0210] 따라서, 디코더(DO)는 열의 각 시작에서 초기화된 상태이고, 그에 의해 디코딩의 병렬 계산의 레벨을 선택하는 관점에서 뛰어난 유연성 및 디코딩시 프로세싱 시간의 최적화를 허용한다.

[0211] 도 7a에 도시된 예시적인 코딩/디코딩 다이어그램에서, 도 3a에 도시된 바와 같이, 코더(CO)는 단일 코딩 유닛(UC)을 포함하는 한편, 디코더(DO)는 6개의 디코딩 유닛들을 포함한다.

[0212] 코딩 유닛(UC)은 열들(SE1, SE2, SE3, SE4, SE5, 및 SE6)을 순차적으로 코딩한다. 도시된 예에서, 열들 SE1 내지 SE4은 완전히 코딩되고, 열 SE5은 코딩되는 과정이며, 열 SE6은 아직 코딩되지 않았다. 코딩의 순차적 발생(sequentiality)을 고려하면, 코딩 유닛(UC)은 열들 SE1, SE2, SE3 및 SE4의 코딩의 순서로 차례로 정렬된 서브-스트림들(F1, F2, F3, F4)을 포함하는 스트림(F)을 전달하도록 적응된다. 이런 목적을 위해, 서브-스트림들(F1, F2, F3 및 F4)은 코딩된 열들(SE1, SE2, SE3, SE4)을 각각 기호화하는 것과 동일한 해칭으로 기호화된다. 상기 코딩된 열들의 코딩의 종료시 비율 단계들 및, 코딩될/디코딩될 다음 열의 코딩 또는 디코딩의 시작시 확률들의 간격의 재초기화에 의해, 디코더(DO)는 그것이 서브-스트림들을 디코딩하도록 서브-스트림을 관독할 때마다, 초기화된 상태이고, 따라서, 최적 방식으로, 예를 들면, 4개의 상이한 플랫폼들 상에 설치될 수 있는 디코딩 유닛들(UD1, UD2, UD3 및 UD4)로 4개의 서브-스트림들(F1, F2, F3, F4)을 병렬로 디코딩할 수 있다.

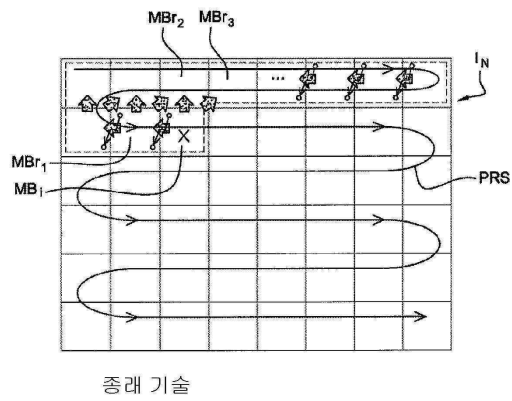
[0213] 도 7b에 도시된 예시적인 코딩/디코딩 다이어그램에서, 도 3c에 도시된 바와 같이, 코더(CO)는 2개의 코딩 유닛들(UC1 및 UC2)을 포함하는 한편, 디코더(DO)는 6개의 디코딩 유닛들을 포함한다.

[0214] 코딩 유닛 UC1은 홀수 번째의 열들(SE1, SE3 및 SE5)을 순차적으로 코딩하는 한편, 코딩 유닛 UC2는 짝수 번째의 열들(SE2, SE4, 및 SE6)을 순차적으로 코딩한다. 이런 목적을 위해, 열들 SE1, SE3 및 SE5는 화이트 백그라운드(white background)를 나타내는 한편, 열들 SE2, SE4 및 SE6은 도티드 백그라운드(dotted background)를 나타낸다. 도시된 예에서, 열들 SE1 내지 SE4은 완전히 코딩되고, 열 SE5는 코딩되고 있는 과정이며, 열 SE6은 아직 코딩되지 않았다. 수행된 코딩이 레벨 2의 병렬 유형인 사실을 고려하면, 코딩 유닛 UC1은 열들 SE1 및 SE3의 각각의 코딩 다음에 획득된 2개의 파트들(F1 및 F3)로 분해된 서브-스트림(F_{2n+1})을 전달하도록 적응되는 한편, 코딩 유닛 UC2는 열들 SE2 및 SE4의 각각의 코딩 다음에 획득된 2개의 파트들(F2 및 F4)로 분해된 서브-스트림(F_{2n})을 전달하도록 적응된다. 코더(CO)는 따라서 2개의 서브-스트림들(F_{2n+1} 및 F_{2n})의 병치(juxtaposition) 및 따라서 도 7a에 도시된 것과 상이한 서브-스트림들(F1, F2, F3, F4)의 순서를 포함하는 스트림(F)을 디코더(DO)에 전송하도록 적응된다. 이런 목적을 위해, 서브-스트림들(F1, F2, F3 및 F4)은 코딩된 열들 SE1, SE2, SE3, SE4을 각각 기호화하는 것과 동일한 해칭으로 기호화되고, 서브-스트림들 F1 및 F3은 화이트 백그라운드(홀수 번째의 열들의 코딩)을 나타내며, 서브-스트림들 F2 및 F4는 도티드 백그라운드(짝수 번째의 열들의 코딩)을 나타낸다.

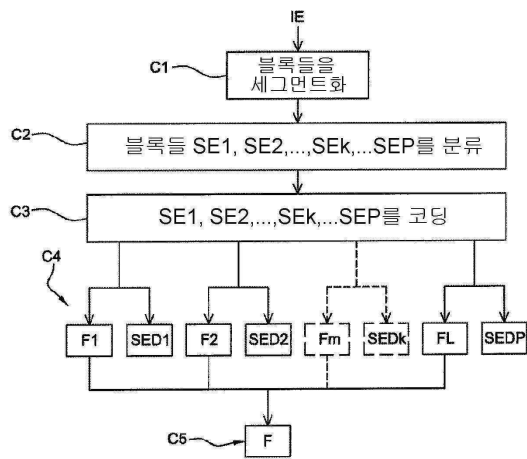
[0215] 도 7a와 함께 언급된 이점들에 대하여, 이런 코딩/디코딩 다이어그램은 그의 디코딩의 병렬 계산의 레벨이 코딩의 병렬 계산의 레벨과 완전히 무관한 디코더를 이용할 수 있고, 그에 의해 코더/디코더의 동작을 한 층 더 최적화하는 것을 가능하게 하는 장점이 더 존재한다.

도면

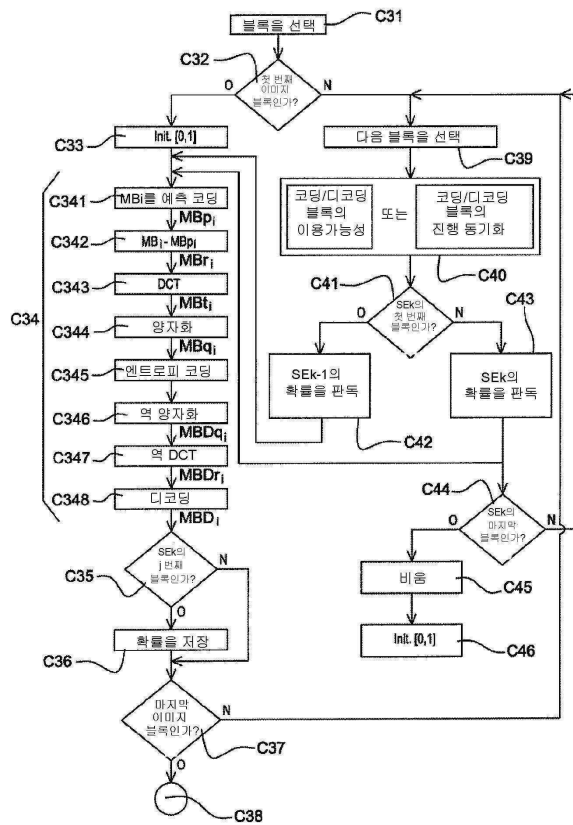
도면1



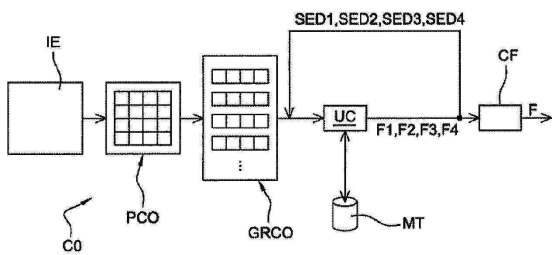
도면2a



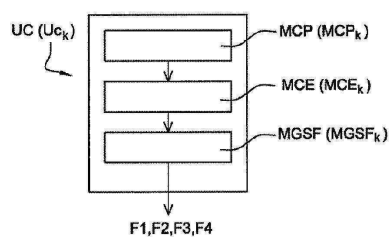
도면2b



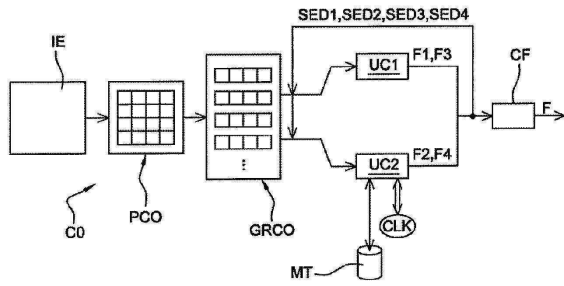
도면3a



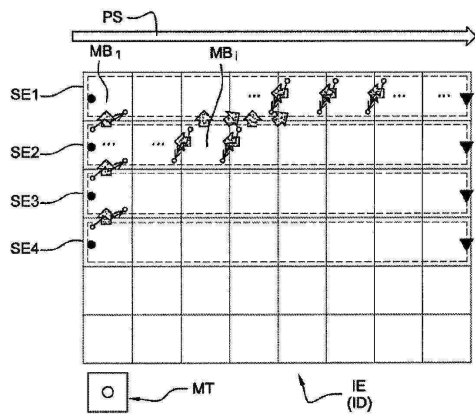
도면3b



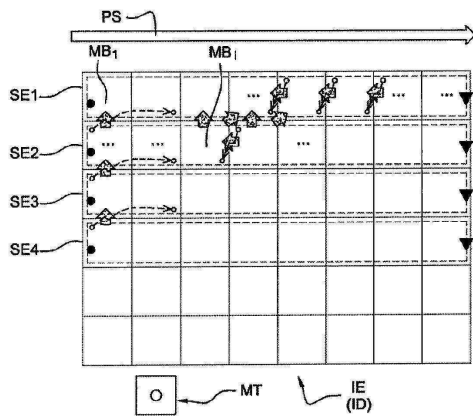
도면3c



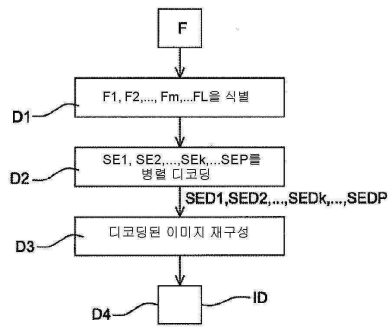
도면4a



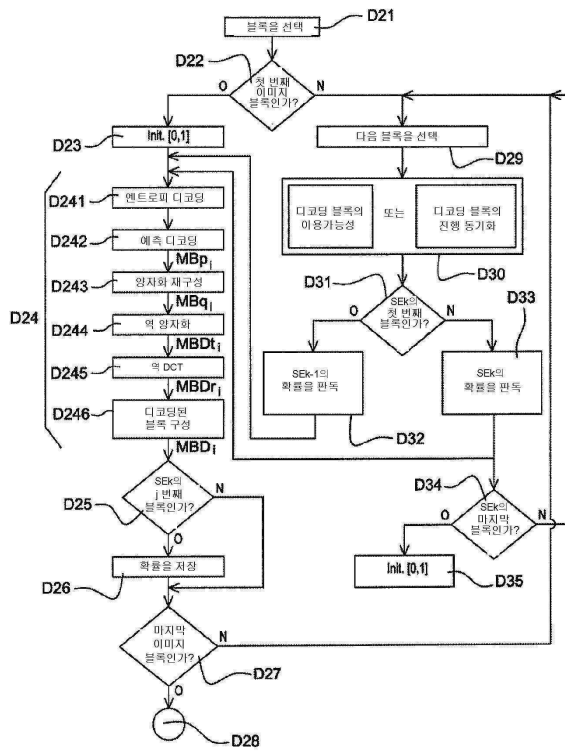
도면4b



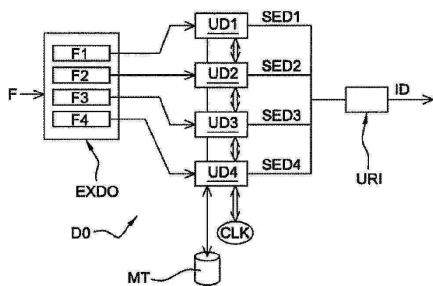
도면5a



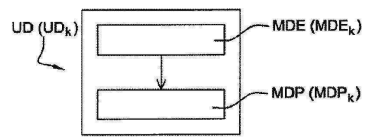
도면5b



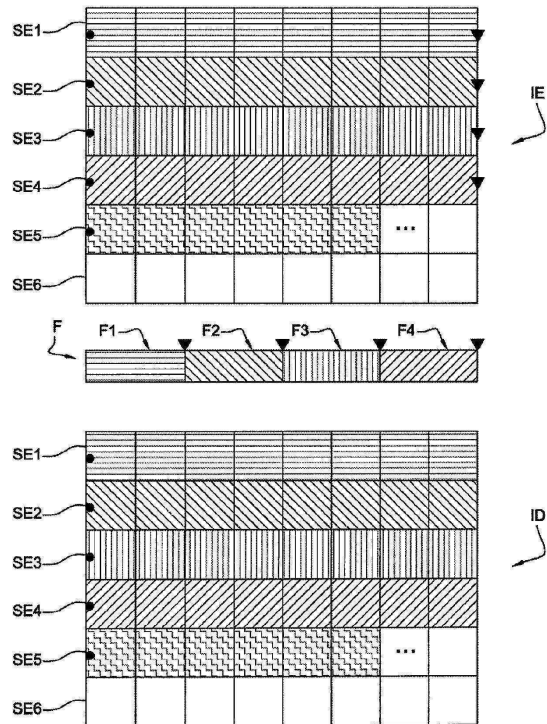
도면6a



도면6b



도면7a



도면7b

