



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2023년10월10일

(11) 등록번호 10-2586032

(24) 등록일자 2023년09월27일

(51) 국제특허분류(Int. Cl.)
H04N 19/13 (2014.01) H04N 19/176 (2014.01)
H04N 19/70 (2014.01)

(52) CPC특허분류
H04N 19/13 (2015.01)
H04N 19/176 (2015.01)

(21) 출원번호 10-2021-7040643(분할)

(22) 출원일자(국제) 2011년10월06일

심사청구일자 2022년01월07일

(85) 번역문제출일자 2021년12월10일

(65) 공개번호 10-2021-0154878

(43) 공개일자 2021년12월21일

(62) 원출원 특허 10-2021-7011762
원출원일자(국제) 2011년10월06일

심사청구일자 2021년05월18일

(86) 국제출원번호 PCT/US2011/055047

(87) 국제공개번호 WO 2012/051033

국제공개일자 2012년04월19일

(30) 우선권주장
61/393,195 2010년10월14일 미국(US)

(56) 선행기술조사문헌
KR101874945 B1
Sugimoto K et al: "Proposal on improved entropy coding method for DCT coefficients", JCTVC-Meeting, no.JCTVC-B070, 17 July 2010.
KR1020100102701 A

(73) 특허권자
인터디지탈 브이씨 홀딩스 인코포레이티드

미국 19809 텔라웨어주 윌밍턴 스위트 300 벨뷰 파크웨이 200

(72) 발명자
후, 양
미국 12180 뉴욕주 트로이 에이퍼티. 에프-11 셋 테라스 285

솔, 조엘
미국 92037 캘리포니아주 라 줄라 빌라 라 줄라 드라이브 유니트 106 8722
(뒷면에 계속)

(74) 대리인
양영준, 백만기

전체 청구항 수 : 총 27 항

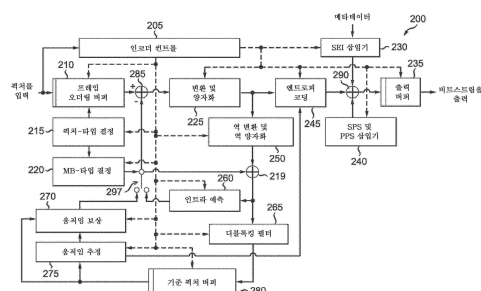
심사관 : 김영태

(54) 발명의 명칭 향상된 엔트로피 인코딩 및 디코딩을 위한 방법 및 장치

(57) 요약

향상된 엔트로피 인코딩 및 디코딩을 위한 방법들 및 장치들이 제공된다. 장치는 블록의 잔여물을 변환해서 변환 계수들을 획득하고, 변환 계수들을 양자화해서 양자화된 변환 계수들을 획득하며, 양자화된 변환 계수들을 엔트로피 코딩함으로써, 한 픽처에서 적어도 한 블록을 인코딩하기 위한 비디오 인코더(200)를 포함한다. 양자화
(뒷면에 계속)

대표도



된 변환 계수들은 처리되고 있는 양자화된 변환 계수들 중 현재 계수가 지정된 값보다 더 크거나 또는 동일한 값을 가진 블록의 최종 논제로 계수임을 나타내기 위해 플래그를 사용해서 인코딩된다.

(52) CPC특허분류

H04N 19/70 (2015.01)

(72) 발명자

루, 시아오안

미국 08540 뉴저지주 프린세톤 케네디 코트 30

인, 펑

미국 14850 뉴욕주 이스아카 존 스트리트 6

쟁, 윤페이

미국 95762 캘리포니아주 샌 디에고 에이퍼터.

5205 토리서클 4615

명세서

청구범위

청구항 1

비디오 인코딩을 위한 장치로서,

블록의 잔여물을 변환하여 변환 계수들을 획득하고, 상기 변환 계수들을 양자화하여 양자화된 변환 계수들을 획득하고, 상기 양자화된 변환 계수들을 엔트로피 코딩함으로써 하나의 픽처에서 적어도 하나의 블록을 인코딩하는 비디오 인코더를 포함하고, 상기 양자화된 변환 계수들은 처리되고 있는 상기 양자화된 변환 계수들 중 현재 변환 계수가 지정된 값인 2보다 더 크거나 또는 동일한 값을 가진 상기 블록에 대한 최종 논제로 계수임을 나타내기 위한 플래그를 사용하여 인코딩되고, 상기 지정된 값인 2는 복수의 값 중에서 선택되고, 상기 선택된 값은 상기 블록에 대해 고정 상태로 유지되고, 상기 플래그는 구문 요소인, 장치.

청구항 2

삭제

청구항 3

제1항에 있어서, 상기 지정된 값보다 더 작은 값을 가지는 상기 양자화된 변환 계수들 중의 후속의 논제로 계수들은 상기 지정된 값보다 더 작은 상기 값을 가지는 상기 후속의 논제로 계수들의 각각의 부호들만을 인코딩함으로써 인코딩되는, 장치.

청구항 4

제1항에 있어서, 상기 픽처는 비디오 시퀀스에 포함된 복수의 픽처 중 하나이고, 상기 지정된 값은 상기 픽처 내의 또는 상기 비디오 시퀀스 내의 상기 복수의 픽처 중 하나 이상의 다른 픽처 내의 이전에 처리된 블록들로부터의 도출된 통계에 응답하여 적응적으로 선택되는, 장치.

청구항 5

제1항에 있어서, 상기 지정된 값은 명시적으로 시그널링되는, 장치.

청구항 6

제1항에 있어서, 상기 지정된 값은 시퀀스 레벨, 프레임 레벨, 슬라이스 레벨, 및 블록 레벨 중 적어도 하나에서 명시적으로 시그널링되는, 장치.

청구항 7

제1항에 있어서, 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 레벨은 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 실제 값으로부터 상기 지정된 값을 감산하여 차이 값을 획득하고 상기 차이 값을 상기 레벨로서 인코딩함으로써 인코딩되어, 상기 차이 값을 상기 지정된 값에 가산함으로써 대응하는 디코더에서 상기 레벨을 재생하는, 장치.

청구항 8

제1항에 있어서, 적어도 sig_flag 구문 요소, 상기 플래그, last_flag 구문 요소, Bin_1 구문 요소, 레벨 구문 요소, 및 부호 구문 요소는 동일한 스캐닝 순서로 인코딩되고, 상기 sig_flag 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 논제로 값을 가지는 지를 나타내고, 상기 last_flag는 상기 논제로 값을 가진 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 주어진 스캐닝 순서로 상기 블록 내의 상기 논제로 값을 가진 최종 양자화된 변환 계수인지를 나타내고, 상기 Bin_1 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 절대값이 현재 알려지지 않은 논제로 값을 가짐을 나타내고, 상기 레벨 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 상기 지정된 값보다 더 큰 절대값을 가질 때 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 상기 절대값을 나타내며, 상기 부호 구문 요소는 상기 양자화된 변환 계수들

중 상기 현재 변환 계수의 대응하는 부호를 나타내는, 장치.

청구항 9

비디오 인코더에서의 방법으로서,

블록의 잔여물을 변환하여 변환 계수들을 획득하고, 상기 변환 계수들을 양자화하여 양자화된 변환 계수들을 획득하고, 상기 양자화된 변환 계수들을 엔트로피 코딩함으로써 하나의 픽처에서 적어도 하나의 블록을 인코딩하는 단계를 포함하고, 양자화된 변환 계수들은 처리되고 있는 상기 양자화된 변환 계수들 중 현재 변환 계수가 지정된 값인 2보다 더 크거나 또는 동일한 값을 가진 상기 블록에 대한 최종 논제로 계수임을 나타내기 위한 플래그를 사용하여 인코딩되고, 상기 지정된 값인 2는 복수의 값들 중에서 선택되고, 상기 선택된 값은 상기 블록에 대해 고정 상태로 유지되며, 상기 플래그는 구문 요소인, 방법.

청구항 10

삭제

청구항 11

제9항에 있어서, 상기 지정된 값보다 더 작은 값을 가지는 상기 양자화된 변환 계수들 중의 후속의 논제로 계수들은 상기 지정된 값보다 더 작은 상기 값을 가지는 상기 후속의 논제로 계수들의 각각의 부호들만을 인코딩함으로써 인코딩되는, 방법.

청구항 12

제9항에 있어서, 상기 픽처는 비디오 시퀀스에 포함된 복수의 픽처 중 하나이고, 상기 지정된 값은 상기 픽처 내의 또는 상기 비디오 시퀀스 내의 상기 복수의 픽처 중의 하나 이상의 다른 픽처 내의 이전에 처리된 블록들로부터 도출된 통계에 응답하여 적응적으로 선택되는, 방법.

청구항 13

제9항에 있어서, 상기 지정된 값은 명시적으로 시그널링되는, 방법.

청구항 14

제9항에 있어서, 상기 지정된 값은 시퀀스 레벨, 프레임 레벨, 슬라이스 레벨, 및 블록 레벨 중의 적어도 하나에서 명시적으로 시그널링되는, 방법.

청구항 15

제9항에 있어서, 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 레벨은 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 실제 값으로부터 상기 지정된 값을 감산하여 차이 값을 획득하고, 상기 차이 값을 상기 레벨로서 인코딩함으로써 인코딩되어, 상기 차이 값을 상기 지정된 값에 가산함으로써 대응하는 디코더에서 상기 레벨을 재생하는, 방법.

청구항 16

제9항에 있어서, 적어도 sig_flag 구문 요소, 상기 플래그, last_flag 구문 요소, Bin_1 구문 요소, 레벨 구문 요소, 및 부호 구문 요소는 동일한 스캐닝 순서로 인코딩되고, 상기 sig_flag 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 논제로 값을 가지는 지를 나타내고, 상기 last_flag는 상기 논제로 값을 가진 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 주어진 스캐닝 순서로 상기 블록 내의 상기 논제로 값을 가진 최종 양자화된 변환 계수인지를 나타내고, 상기 Bin_1 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 절대값이 현재 알려지지 않은 논제로 값을 가짐을 나타내고, 상기 레벨 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 상기 지정된 값보다 더 큰 절대값을 가질 때 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 상기 절대값을 나타내며, 상기 부호 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 대응하는 부호를 나타내는, 방법.

청구항 17

비디오 디코딩을 위한 장치로서,

양자화된 변환 계수들을 엔트로피 디코딩하고, 상기 양자화된 변환 계수들을 역양자화하여 변환 계수들을 획득하고, 상기 변환 계수들을 역변환하여 블록을 재구성하는 데 사용하기 위한 상기 블록의 재구성된 잔여물을 획득함으로써 하나의 픽처에서 적어도 하나의 블록을 디코딩하기 위한 비디오 디코더를 포함하고, 양자화된 변환 계수들은 디코딩되고, 플래그는 처리되고 있는 상기 양자화된 변환 계수들 중 현재 변환 계수가 지정된 값인 2보다 크거나 동일한 값을 갖는 상기 블록에 대한 최종 논제로 계수임을 나타내고, 상기 지정된 값인 2는 복수의 값 중에서 선택되고, 상기 선택된 값은 상기 블록에 대해 고정 상태로 유지되고, 상기 플래그는 구문 요소인, 장치.

청구항 18

삭제

청구항 19

제17항에 있어서, 상기 지정된 값보다 더 작은 값을 갖는 상기 양자화된 변환 계수들 중의 후속의 논제로 계수들은 상기 지정된 값보다 더 작은 상기 값을 갖는 상기 후속의 논제로 계수들의 각각의 부호들만을 디코딩함으로써 디코딩되는, 장치.

청구항 20

제17항에 있어서, 상기 지정된 값은 명시적으로 결정되는, 장치.

청구항 21

제17항에 있어서, 상기 지정된 값은 시퀀스 레벨, 프레임 레벨, 슬라이스 레벨, 및 블록 레벨 중의 적어도 하나로부터 명시적으로 결정되는, 장치.

청구항 22

제17항에 있어서, 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 레벨은 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 실제 값과 상기 지정된 값 사이에서 이전에 결정된 차이 값을 디코딩하고, 상기 레벨을 획득하기 위해 상기 차이 값을 상기 지정된 값에 가산함으로써 디코딩되는, 장치.

청구항 23

제17항에 있어서, 적어도 sig_flag 구문 요소, 상기 플래그, last_flag 구문 요소, Bin_1 구문 요소, 레벨 구문 요소, 및 부호 구문 요소는 동일한 스캐닝 순서로 인코딩되고, 상기 sig_flag 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 논제로 값을 가지는 지를 나타내고, 상기 last_flag는 상기 논제로 값을 가진 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 주어진 스캐닝 순서로 상기 블록 내의 상기 논제로 값을 가진 최종 양자화된 변환 계수인지를 나타내고, 상기 Bin_1 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 절대값이 현재 알려지지 않은 논제로 값을 가짐을 나타내고, 상기 레벨 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 상기 지정된 값보다 더 큰 절대값을 가질 때 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 상기 절대값을 나타내며, 상기 부호 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 대응하는 부호를 나타내는, 장치.

청구항 24

비디오 디코더에서의 방법으로서,

양자화된 변환 계수들을 엔트로피 디코딩하고, 상기 양자화된 변환 계수들을 역양자화하여 변환 계수들을 획득하고, 상기 변환 계수들을 역변환하여 블록을 재구성하는 데 사용하기 위한 상기 블록의 재구성된 잔여물을 획득함으로써 하나의 픽처에서 적어도 하나의 블록을 디코딩하는 단계를 포함하고, 양자화된 변환 계수들은 디코딩되고, 플래그는 처리되고 있는 상기 양자화된 변환 계수들 중 현재 변환 계수가 지정된 값인 2보다 크거나 동일한 값을 갖는 상기 블록에 대한 최종 논제로 계수임을 나타내고, 상기 지정된 값인 2는 복수의 값 중에서 선택되고, 상기 선택된 값은 상기 블록에 대해 고정 상태로 유지되고, 상기 플래그는 구문 요소인, 방법.

청구항 25

삭제

청구항 26

제24항에 있어서, 상기 지정된 값보다 더 작은 값을 갖는 상기 양자화된 변환 계수들 중의 후속의 논제로 계수들은 상기 지정된 값보다 더 작은 상기 값을 갖는 상기 후속의 논제로 계수들의 각각의 부호들만을 디코딩함으로써 디코딩되는, 방법.

청구항 27

제24항에 있어서, 상기 지정된 값은 명시적으로 결정되는, 방법.

청구항 28

제24항에 있어서, 상기 지정된 값은 시퀀스 레벨, 프레임 레벨, 슬라이스 레벨, 및 블록 레벨 중의 적어도 하나로부터 명시적으로 결정되는, 방법.

청구항 29

제24항에 있어서, 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 레벨은 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 실제 값과 상기 지정된 값 사이에서 이전에 결정된 차이 값을 디코딩하고, 상기 레벨을 획득하기 위해 상기 차이 값을 상기 지정된 값에 가산함으로써 디코딩되는, 방법.

청구항 30

제24항에 있어서, 적어도 sig_flag 구문 요소, 상기 플래그, last_flag 구문 요소, Bin_1 구문 요소, 레벨 구문 요소, 및 부호 구문 요소는 동일한 스캐닝 순서로 인코딩되고, 상기 sig_flag 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 논제로 값을 가지는 지를 나타내고, 상기 last_flag는 상기 논제로 값을 가진 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 주어진 스캐닝 순서로 상기 블록 내의 상기 논제로 값을 가진 최종 양자화된 변환 계수인지를 나타내고, 상기 Bin_1 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 절대값이 현재 알려지지 않은 논제로 값을 가짐을 나타내고, 상기 레벨 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수가 상기 지정된 값보다 더 큰 절대값을 가질 때 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 상기 절대값을 나타내며, 상기 부호 구문 요소는 상기 양자화된 변환 계수들 중 상기 현재 변환 계수의 대응하는 부호를 나타내는, 방법.

청구항 31

명령어들을 저장한 컴퓨터 판독가능 저장 매체로서, 상기 명령어들이 프로세서에 의해 실행될 때 제9항 및 제11항 내지 제16항 및 제24항 및 제26항 내지 제30항 중 어느 한 항의 방법이 구현되는, 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

[0001] <관련 출원 데이터>

[0002] 본 출원은, 그 전체가 본 명세서에 참조용으로 인용된, 2010년 10월 14일에 출원된, 미국 임시 출원 일련 번호 제61/393,195호의 이익을 주장한다.

[0003] 본 발명은 일반적으로 비디오 인코딩 및 디코딩에 관한 것으로, 특히, 향상된 엔트로피 인코딩 및 디코딩을 위한 방법들 및 장치들에 관한 것이다.

배경 기술

[0004] 비디오 코딩 표준들은 인트라/인터 프레임 상관성(intra/inter frame correlation)의 중복성(redundancy)을 이용하기 위해 예측 및 블록-기반 변환들을 사용한다. 더욱이, 엔트로피 코딩은 코딩된 비트스트림이 엔트로피

경계를 달성하게 하고, 코딩 효율을 더 향상시킨다.

- [0005] 비디오 코딩 시스템의 엔트로피 코딩의 중요한 사용은, 인트라/인터 예측, 블록 변환, 및 양자화 후의 잔류 데이터 블록인, 블록의 양자화된 변환 계수들의 코딩이다. 이러한 데이터의 경우, 엔트로피 코딩 툴들이 개발되어 왔으며, 허프만 코딩(Huffman coding) 등의 가변 길이 코딩 내지 산술 코딩(arithmetic coding)의 범위이다. 최신의 CABAC(context-adaptive binary arithmetic coding)는 높은 코딩 효율을 달성하지만, CABAC 코딩 프로시저의 체계적이지 않은 구현은 데이터 블록을 코딩하기 위해 2개의 스캔 패스들(two scanning passes)이 실행되게 야기한다.
- [0006] CABAC는 ISO/IEC(the International Organization for Standardization/International Electrotechnical Commission) MPEG-4(Moving Picture Experts Group-4) 파트 10 AVC(Advanced Video Coding) 표준/ITU-T(International Telecommunication Union, Telecommunication Sector) H.264 추천(이후부터는 "MPEG-4 AVC 표준")의 양자화된 변환 계수 블록에 대한 엔트로피 코딩 방법이다. CABAC는 2번의 메인 패스들로 블록을 코딩한다. 제1 패스에서, CABAC는 전방향 지그재그 스캔 순서에 따라 블록의 시그니피컨스 맵(significance map)을 코딩한다. 제2 패스에서, CABAC는 역 지그재그 스캔 순서로 논제로(non-zero) 값들을 코딩한다.
- [0007] 도 1을 참조하면, CABAC 코딩의 일례가 일반적으로 참조 부호(100)로 표시된다. 시그니피컨스 맵 코딩 패스, 즉 제1 패스에서, CABAC는 논제로 계수들의 위치들을 나타내기 위해 sig_flag 및 last_flag를 사용한다.
- [0008] 논제로 값들의 역 지그재그 코딩에서, 2개의 서브-코딩 프로세스들이 사용된다. 제1 서브-코딩 프로세스에서, Bin_1(즉, 제1 빈)이라고 하는 구문은 논제로 계수가 절대값 1을 갖는지의 여부를 나타내기 위해 사용된다. 논제로 계수가 절대값 1을 가지면, Bin_1 = 1 이고, 논제로 계수의 부호가 발송된다. 그렇지 않으면, Bin_1 = 0 이고, 인코딩은 제2 서브-코딩 프로세스로 이동한다. 제2 서브-코딩 프로세스에서, CABAC는, Bin_1 = 0에 대응하는, 1 보다 큰 절대값을 가진 계수들을 코딩하고, 후에 각각의 부호들을 송신한다.
- [0009] CABAC의 단점은, 대응 코딩이 2개의 스캔 패스들(즉, 시그니피컨스 맵을 코딩하기 위한 전방향 재그재그 스캔, 및 값들을 코딩하기 위한 역 지그재그 스캔)을 수반한다는 점이다. 또한, CABAC의 설계는 주로 더 작은 블록 크기들(예를 들어, 4×4 및 8×8)에 대한 것이다. CABAC는 더 큰 블록들(예를 들어, 16×16, 32×32, 및 64×64)에 대해서는 덜 효율적인 것으로 나타난다.
- [0010] 한 종래 기술의 방식은, 1 보다 큰 이산 코사인 변환(DCT) 계수의 최종 위치를 시그널링하기 위해 플래그를 추가할 것을 제안한다. 그러나, 종래 기술의 방식은 1 보다 큰 플래그로 제한되고, 여전히 2번의 스캔 패스들을 사용한다.

발명의 내용

해결하려는 과제

과제의 해결 수단

- [0011] 종래 기술의 여타 결점 및 단점들은 본 발명에 의해 처리되며, 본 발명은 향상된 엔트로피 인코딩 및 디코딩을 위한 방법들 및 장치들에 관한 것이다.
- [0012] 본 발명의 일 양상에 따라, 장치가 제공된다. 장치는 블록의 잔여물을 변환해서 변환 계수들을 획득하고, 변환 계수들을 양자화해서 양자화된 변환 계수들을 획득하며, 양자화된 변환 계수들을 엔트로피 코딩함으로써, 한 픽처에서 적어도 한 블록을 인코딩하기 위한 비디오 인코더를 포함한다. 양자화된 변환 계수들은 처리되고 있는 양자화된 변환 계수들 중 현재 계수가 지정된 값보다 더 크거나 또는 동일한 값을 가진 블록의 최종 논제로 계수임을 나타내기 위해 플래그를 사용해서 인코딩된다.
- [0013] 본 발명의 다른 양상에 따라, 비디오 인코더의 한 방법이 제공된다. 방법은 블록의 잔여물을 변환해서 변환 계수들을 획득하고, 변환 계수들을 양자화해서 양자화된 변환 계수들을 획득하며, 양자화된 변환 계수들을 엔트로피 코딩함으로써, 한 픽처에서 적어도 한 블록을 인코딩하는 단계를 포함한다. 양자화된 변환 계수들은 처리되고 있는 양자화된 변환 계수들 중 현재 계수가 지정된 값보다 더 크거나 또는 동일한 값을 가진 블록의 최종 논제로 계수임을 나타내기 위해 플래그를 사용해서 인코딩된다.
- [0014] 본 발명의 또 다른 양상에 따라, 장치가 제공된다. 장치는 양자화된 변환 계수들을 엔트로피 디코딩하고, 양자

화된 변환 계수들을 역양자화(de-quantizing)해서 변환 계수들을 획득하며, 변환 계수들을 역변환해서 블록을 재구성할 때 사용되는 블록의 재구성된 잔여물을 획득함으로써, 한 픽처에서 적어도 한 블록을 디코딩하기 위한 비디오 디코더를 포함한다. 양자화된 변환 계수들은 처리되고 있는 양자화된 변환 계수들 중 현재 계수가 지정된 값보다 더 크거나 또는 동일한 값을 가진 블록의 최종 논제로 계수임을 나타내기 위해 플래그를 사용해서 디코딩된다.

[0015] 본 발명의 또 다른 양상에 따라, 비디오 디코더의 한 방법이 제공된다. 방법은 양자화된 변환 계수들을 엔트로피 디코딩하고, 양자화된 변환 계수들을 역양자화해서 변환 계수들을 획득하며, 변환 계수들을 역변환해서 블록을 재구성할 때 사용되는 블록의 재구성된 잔여물을 획득함으로써, 한 픽처에서 적어도 한 블록을 디코딩하는 단계를 포함한다. 양자화된 변환 계수들은 처리되고 있는 양자화된 변환 계수들 중 현재 계수가 지정된 값보다 더 크거나 또는 동일한 값을 가진 블록의 최종 논제로 계수임을 나타내기 위해 플래그를 사용해서 디코딩된다.

[0016] 본 발명의 여타 양상들, 특징들 및 장점들은, 첨부 도면들과 관련해서 관독될, 일례의 실시예들의 이하의 상세한 설명으로부터 명백해질 것이다.

도면의 간단한 설명

[0017] 본 발명은 이하의 일례의 도면들에 따라 더 잘 이해될 수 있다.

도 1은 종래 기술에 따른, CABAC 코딩의 일례를 도시한 도면이다.

도 2는 본 발명의 일 실시예에 따라, 본 원리들이 적용될 수 있는 일례의 비디오 인코더를 도시한 블록도이다.

도 3은 본 발명의 일 실시예에 따라, 본 원리들이 적용될 수 있는 일례의 비디오 디코더를 도시한 블록도이다.

도 4는 본 발명의 일 실시예에 따라, 4×4 크기의 일례의 양자화된 변환 블록을 도시한 도면이다.

도 5는 본 발명의 일 실시예에 따른, 코딩 프로세스의 일례를 도시한 도면이다.

도 6은 본 발명의 일 실시예에 따라, Bin_1이 저장되지 않은 일례의 특별 경우들을 도시한 도면이다.

도 7a는 본 발명의 일 실시예에 따른, 엔트로피 인코딩의 일례의 방법을 도시한 흐름도이다.

도 7b는 본 발명의 일 실시예에 따른, last_ge2_flag를 인코딩하는 일례의 방법을 도시한 흐름도이다.

도 7c는 본 발명의 일 실시예에 따른, last_flag를 인코딩하는 일례의 방법을 도시한 흐름도이다.

도 7d는 본 발명의 일 실시예에 따른, Bin_1을 인코딩하는 일례의 방법을 도시한 흐름도이다.

도 7e는 본 발명의 일 실시예에 따른, level을 인코딩하는 일례의 방법을 도시한 흐름도이다.

도 8a는 본 발명의 일 실시예에 따른, 엔트로피 디코딩의 일례의 방법을 도시한 흐름도이다.

도 8b는 본 발명의 일 실시예에 따른, last_ge2_flag를 디코딩하는 일례의 방법을 도시한 흐름도이다.

도 8c는 본 발명의 일 실시예에 따른, last_flag를 디코딩하는 일례의 방법을 도시한 흐름도이다.

도 8d는 본 발명의 일 실시예에 따른, Bin_1을 디코딩하는 일례의 방법을 도시한 흐름도이다.

도 8e는 본 발명의 일 실시예에 따른, level을 디코딩하는 일례의 방법을 도시한 흐름도이다.

도 9는 본 발명의 일 실시예에 따른, 현재 변환 계수의 값을 선택 및 시그널링하는 방법을 도시한 흐름도이다.

도 10은 본 발명의 일 실시예에 따른, 현재 변환 계수의 값을 디코딩하는 일례의 방법을 도시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0018] 본 발명은 향상된 엔트로피 인코딩 및 디코딩을 위한 방법들 및 장치들에 관한 것이다.





[0019] 본 설명은 본 발명을 설명한다. 따라서, 본 명세서에 명확하게 기술 또는 도시되지 않더라도, 본 발명을 구현하며 본 원리 및 범위 내에 포함되는 각종 구성들을 당업자가 창안할 수 있음을 알 것이다.

[0020] 본 명세서에 인용된 모든 일례들 및 조건 언어는 발명인(들)이 부여한 본 원리들 및 개념들, 더 나아가 본 기술 분야를 독자가 이해하는 것을 교육학적으로 돕도록 의도된 것이며, 이러한 구체적으로 인용된 일례들 및 조건들로 제한되지 않는 것으로 해석될 것이다.











- [0021] 게다가, 본 명세서에서 본 발명의 원리들, 양상들, 및 실시예들 뿐만 아니라 그 특정 일례들을 인용하는 모든 진술들은, 구조적 및 기능적 동등물들을 둘 다 포함하는 것으로 의도된다. 또한, 이러한 동등물들은 현재 공지된 동등물들뿐만 아니라 차후에 개발되는 동등물들, 즉, 구조와 무관하게, 동일한 기능을 실행하는 임의의 개발된 요소들을 포함하는 것으로 의도된다.
- [0022] 따라서, 예를 들어, 본 명세서에서 제시된 블록도들은 본 발명을 구현하는 일례의 회로의 개념적인 뷰들을 나타냄을 당업자는 알 것이다. 유사하게, 컴퓨터 또는 프로세서가 명확하게 도시되든 아니든 간에, 임의의 플로차트들, 흐름도들, 상태 전이도들, 의사 코드 등이 컴퓨터 판독 가능 매체에서 대체로 표현될 수 있어서 컴퓨터 또는 프로세서에 의해 실행될 수 있는 각종 프로세스들을 나타냄을 알 것이다.
- [0023] 도면들에 도시된 각종 요소들의 기능들은 적합한 소프트웨어와 관련하여 소프트웨어를 실행할 수 있는 하드웨어 뿐만 아니라 전용 하드웨어를 사용해서 제공될 수 있다. 프로세서에 의해 제공될 때, 기능들은 단일 전용 프로세서에 의해, 단일 공유 프로세서에 의해, 또는 복수의 개별 프로세서들 - 그 중 일부는 공유될 수 있음 - 에 의해 제공될 수 있다. 더욱이, 용어 "프로세서" 또는 "컨트롤러"의 명확한 사용은 소프트웨어를 실행할 수 있는 하드웨어를 배타적으로 나타내는 것으로 해석되어서는 안 되고, 제한 없이, 디지털 신호 프로세서("DSP") 하드웨어, 소프트웨어를 저장하기 위한 판독 전용 메모리("ROM"), 랜덤 액세스 메모리("RAM"), 및 비휘발성 스토리지를 암암리에 포함할 수 있다.
- [0024] 종래의 및/또는 주문형, 다른 하드웨어가 또한 포함될 수 있다. 유사하게, 도면들에 도시된 임의의 스위치들은 단지 개념적인 것이다. 그 기능은 프로그램 로직의 동작을 통해, 전용 로직을 통해, 프로그램 제어 및 전용 로직의 상호 동작을 통해, 또는 심지어는 수동으로 실행될 수 있으며, 문맥으로부터 더 구체적으로 이해되는 바와 같이 특정 기술이 구현자에 의해 선택될 수 있다.
- [0025] 그 청구항들에서, 지정된 기능을 실행하기 위한 수단으로서 표현된 임의의 요소는, 예를 들어, a) 기능을 실행하는 회로 요소들의 조합, 또는 b) 기능을 실행하기 위해 소프트웨어를 실행하기 위한 적합한 회로와 결합된, 펌웨어, 마이크로코드 등을 포함하는 임의의 형태의 소프트웨어를 포함하는 기능을 실행하는 임의의 방법을 아우르도록 의도된 것이다. 이러한 청구항들에 의해 정의된 본 원리들은, 각종 인용된 수단에 의해 제공된 기능들이 청구항들이 요구하는 방식으로 함께 조합 및 제공된다는 사실에 있다. 따라서, 이러한 기능들을 제공할 수 있는 임의의 수단은 본 명세서에 도시된 바와 동등하다고 간주된다.
- [0026] 본 발명의 "one embodiment(일 실시예)" 또는 "an embodiment(일 실시예)" 뿐만 아니라 그 다른 변형들에 대한 언급은, 본 실시예와 관련해서 기술된 특정 특징, 구조, 특성 등이 본 발명의 적어도 하나의 실시예에 포함됨을 의미한다. 따라서, 본 명세서에 걸쳐 각종 장소들에 나타나는 구절 "in one embodiment(일 실시예에서)" 또는 "in an embodiment(일 실시예에서)" 뿐만 아니라 임의의 다른 변형들의 출현들은 반드시 모두 동일한 실시예를 나타내는 것은 아니다.
- [0027] 예를 들어, "A/B", "A and/or B(A 및/또는 B)" 및 "at least one of A and B(A 및 B 중 적어도 하나)"의 경우들에서, 이하의 "/", "and/or(및/또는)", 및 "at least one of(~ 중 적어도 하나)" 중 임의의 것의 사용은, 처음 열거된 옵션 (A) 만을 선택함, 또는 2번째로 열거된 옵션 (B) 만을 선택함, 또는 옵션들 둘 다(A 및 B)를 선택함을 망라하려는 의도임을 알 것이다. 다른 일례로서, "A, B, and/or C(A, B, 및/또는 C)" 및 "at least one of A, B, and C(A, B, 및 C 중 적어도 하나)"의 경우들에서, 이러한 구절은 처음 열거된 옵션 (A) 만을 선택함, 또는 2번째로 열거된 옵션 (B) 만을 선택함, 또는 3번째로 열거된 옵션 (C) 만을 선택함, 또는 처음과 2번째로 열거된 옵션들(A 및 B) 만을 선택함, 또는 처음과 3번째로 열거된 옵션들(A 및 C) 만을 선택함, 또는 2번째와 3번째로 열거된 옵션들(B 및 C) 만을 선택함, 또는 모든 3개의 옵션들(A 및 B 및 C)을 선택함을 망라하려는 의도이다. 이는, 열거된 다수의 아이템들에 대해, 당업자에 의해 쉽게 명백해지는 바와 같이, 확장될 수 있다.
- [0028] 또한, 본 명세서에서 사용되는 단어들 "픽처(picture)" 및 "이미지(image)"는 상호 교환되어 사용되며, 비디오 시퀀스로부터의 정지 화상 또는 화면을 나타낸다. 공지된 바와 같이, 픽처는 프레임 또는 필드일 수 있다.
- [0029] 도 2를 참조하면, 본 원리들이 적용될 수 있는 일례의 비디오 인코더가 일반적으로 참조 부호(200)로 표시된다. 비디오 인코더(200)는 결합기(285)의 비반전 입력과 신호 통신하는 출력을 가진 프레임 오더링 버퍼(frame ordering buffer)(210)를 포함한다. 결합기(285)의 출력은 변환기 및 양자화기(225)의 제1 입력과 신호 통신으로 연결된다. 변환기 및 양자화기(225)의 출력은 엔트로피 코더(245)의 제1 입력 및 역 변환기 및 역 양자화기(250)의 제1 입력과 신호 통신으로 연결된다. 엔트로피 코더(245)의 출력은 결합기(290)의 제1 비반전 입력과

신호 통신으로 연결된다. 결합기(290)의 출력은 출력 버퍼(235)의 제1 입력과 신호 통신으로 연결된다.

- [0030] 인코더 컨트롤러(205)의 제1 출력은 프레임 오더링 버퍼(210)의 제2 입력, 역 변환기 및 역 양자화기(250)의 제2 입력, 픽처-타입 결정 모듈(215)의 입력, 매크로블록-타입(MB-타입) 결정 모듈(220)의 제1 입력, 인트라 예측 모듈(260)의 제2 입력, 디블록킹 필터(265)의 제2 입력, 움직임 보상기(270)의 제1 입력, 움직임 추정기(275)의 제1 입력, 및 기준 픽처 버퍼(280)의 제2 입력과 신호 통신으로 연결된다.
- [0031] 인코더 컨트롤러(205)의 제2 출력은 보충 강화 정보(SEI) 삽입기(230)의 제1 입력, 변환기 및 양자화기(225)의 제2 입력, 엔트로피 코더(245)의 제2 입력, 출력 버퍼(235)의 제2 입력, 및 시퀀스 파라미터 집합(PPS) 및 픽처 파라미터 집합(PPS) 삽입기(240)의 입력과 신호 통신으로 연결된다.
- [0032] SEI 삽입기(230)의 출력은 결합기(290)의 제2 비반전 입력과 신호 통신으로 연결된다.
- [0033] 픽처-타입 결정 모듈(215)의 제1 출력은 프레임 오더링 버퍼(210)의 제3 입력과 신호 통신으로 연결된다. 픽처-타입 결정 모듈(215)의 제2 출력은 매크로블록-타입 결정 모듈(220)의 제2 입력과 신호 통신으로 연결된다.
- [0034] 시퀀스 파라미터 집합(PPS) 및 픽처 파라미터 집합(PPS) 삽입기(240)의 출력은 결합기(290)의 제3 비반전 입력과 신호 통신으로 연결된다.
- [0035] 역 변환기 및 역 양자화기(250)의 출력은 결합기(219)의 제1 비반전 입력과 신호 통신으로 연결된다. 결합기(219)의 출력은 인트라 예측 모듈(260)의 제1 입력 및 디블록킹 필터(265)의 제1 입력과 신호 통신으로 연결된다. 디블록킹 필터(265)의 출력은 기준 픽처 버퍼(280)의 제1 입력과 신호 통신으로 연결된다. 기준 픽처 버퍼(180)의 출력은 움직임 추정기(275)의 제2 입력 및 움직임 보상기(270)의 제3 입력과 신호 통신으로 연결된다. 움직임 추정기(275)의 제1 출력은 움직임 보상기(270)의 제2 입력과 신호 통신으로 연결된다. 움직임 추정기(275)의 제2 출력은 엔트로피 코더(245)의 제3 입력과 신호 통신으로 연결된다.
- [0036] 움직임 보상기(270)의 출력은 스위치(297)의 제1 입력과 신호 통신으로 연결된다. 인트라 예측 모듈(260)의 출력은 스위치(297)의 제2 입력과 신호 통신으로 연결된다. 매크로블록-타입 결정 모듈(220)의 출력은 스위치(297)의 제3 입력과 신호 통신으로 연결된다. 스위치(297)의 제3 입력은, (제어 입력, 즉, 제3 입력에 비해) 스위치의 "데이터" 입력이 움직임 보상기(270) 또는 인트라 예측 모듈(260)에 의해 제공되는지의 여부를 결정한다. 스위치(297)의 출력은 결합기(219)의 제2 비반전 입력 및 결합기(285)의 반전 입력과 신호 통신으로 연결된다.
- [0037] 프레임 오더링 버퍼(210)의 제1 입력 및 인코더 컨트롤러(205)의 입력은, 입력 픽처를 수신하기 위해, 인코더(200)의 입력들로서 사용될 수 있다. 더욱이, 보충 강화 정보(SEI) 삽입기(230)의 제2 입력은, 메타데이터를 수신하기 위해, 인코더(200)의 입력들로서 사용될 수 있다. 출력 버퍼(235)의 출력은, 비트스트림을 출력하기 위해, 인코더(200)의 출력들로서 사용될 수 있다.
- [0038] 도 3을 참조하면, 본 원리들이 적용될 수 있는 일례의 비디오 디코더가 참조 부호(300)로 일반적으로 표시된다. 비디오 디코더(300)는 엔트로피 디코더(345)의 제1 입력과 신호 통신으로 연결된 출력을 가진 입력 버퍼(310)를 포함한다. 엔트로피 디코더(345)의 제1 출력은 역 변환기 및 역 양자화기(350)의 제1 입력과 신호 통신으로 연결된다. 역 변환기 및 역 양자화기(350)의 출력은 결합기(325)의 제2 비반전 입력과 신호 통신으로 연결된다. 결합기(325)의 출력은 디블록킹 필터(365)의 제2 입력 및 인트라 예측 모듈(360)의 제1 입력과 신호 통신으로 연결된다. 디블록킹 필터(365)의 제2 출력은 기준 픽처 버퍼(380)의 제1 입력과 신호 통신으로 연결된다. 기준 픽처 버퍼(380)의 출력은 움직임 보상기(370)의 제2 입력과 신호 통신으로 연결된다.
- [0039] 엔트로피 디코더(345)의 제2 출력은 움직임 보상기(370)의 제3 입력, 디블록킹 필터(365)의 제1 입력, 및 인트라 예측 모듈(360)의 제3 입력과 신호 통신으로 연결된다. 엔트로피 디코더(345)의 제3 출력은 디코더 컨트롤러(305)의 입력과 신호 통신으로 연결된다. 디코더 컨트롤러(305)의 제1 출력은 엔트로피 디코더(345)의 제2 입력과 신호 통신으로 연결된다. 디코더 컨트롤러(305)의 제2 출력은 역 변환기 및 역 양자화기(350)의 제2 입력과 신호 통신으로 연결된다. 디코더 컨트롤러(305)의 제3 출력은 디블록킹 필터(365)의 제3 입력과 신호 통신으로 연결된다. 디코더 컨트롤러(305)의 제4 출력은 인트라 예측 모듈(360)의 제2 입력, 움직임 보상기(370)의 제1 입력, 및 기준 픽처 버퍼(380)의 제2 입력과 신호 통신으로 연결된다.
- [0040] 움직임 보상기(370)의 출력은 스위치(397)의 제1 입력과 신호 통신으로 연결된다. 인트라 예측 모듈(360)의 출력은 스위치(397)의 제2 입력과 신호 통신으로 연결된다. 스위치(397)의 출력은 결합기(325)의 제1 비반전 입력과 신호 통신으로 연결된다.

- [0041] 입력 버퍼(310)의 입력은 입력 비트스트림을 수신하기 위해, 디코더(300)의 입력으로서 사용될 수 있다. 디블록킹 필터(365)의 제1 출력은 출력 픽처를 출력하기 위해, 디코더(300)의 출력으로서 사용될 수 있다.
- [0042] 상술된 바와 같이, 본 발명은 향상된 엔트로피 인코딩 및 디코딩을 위한 방법들 및 장치들에 관한 것이다. 유익하게, 본 발명은 CABAC의 체계적이지 않은 약점을 극복한다. 본 발명은 바이너리 산술 코딩 엔진을 또한 사용하지만, 바이너리 빈들(binary bins)을 저장하는 체계적인 코딩 프로시저로, 인코더 및 디코더 둘 다에서 바이너리 산술 코딩 엔진의 사용을 감소시킨다. 이러한 체계적인 코딩 방법 및 감소된 바이너리 빈들로, 종래 기술의 CABAC 시스템들 보다 더 간단한 코딩 시스템 및 더 높은 압축 효율이 달성된다.
- [0043] 따라서, 본 발명은 더 적은 구문 빈들로 계수 블록들을 체계적으로 엔트로피 코딩하는 것에 관한 것이다. 체계적으로, 계수 값이 주어진 스캔 순서로 발견(또는 처리)될 때 계수 값을 코딩한다. 구체적으로 말해서, 제로 및 논제로 계수들을 나타내기 위해 "시그니피컨스 플래그(significance flag)"(sig_flag)를 사용한다. 논제로의 경우, 각각, 위치들의 왼쪽 집합이 "greater or equal to 2(2 보다 더 크거나 또는 동일한)"(ge2) 다수의 논제로 계수들을 포함하는 지 및 시그니피컨스 계수들을 포함하는 지를 나타내기 위해 "2 보다 더 크거나 또는 동일한 최종 계수"(last_ge2_flag) 및 "최종 플래그"(last_flag)를 사용한다. 시그니피컨트 계수가 발견되거나 또는 sig_flag(sig_flag = 1)에 의해 표시될 때마다, 그 값은 즉시 코딩된다. 본 발명의 방식에 적어도 3가지 이점들이 있다:
- [0044] (1) 전체 블록이 한 번의 스캔으로 코딩되어, 시스템이 더 간단하게 된다.
- [0045] (2) 바이너리 빈들이 감소해서, 일부 바이너리 산술 코딩 동작들이 인코더 및 디코더에 저장된다.
- [0046] (3) 이웃 코딩 계수들의 레벨 정보가 sig_flag, last_flag, last_ge2_flag, 및 level 빈들 등의 구문의 문맥 모델들을 설계하는데 사용될 수 있다.
- [0047] 따라서, 기술된 엔트로피 코딩 시스템은 종래 기술의 시스템들보다 더 간단하고 더 효율적인 엔트로피 코딩 시스템이다.
- [0048] 비디오 코딩 시스템에서, 생 데이터는 인트라 또는 인터 예측으로 처리되어서, 인트라 또는 인터 프레임 상관성을 제거하고, 그 후, 4×4 , 8×8 , 16×16 , 32×32 , 및 64×64 DCT(또는 일부 다른 변환) 등의 블록-기반 변환으로 처리되어서, 상관성을 더 제거한다. 그 후, 양자화가 변환 블록들의 계수들에 적용된다. 일 실시예에서, 엔트로피 코딩이 마지막으로 실행되어, 각각의 변환된 블록의 양자화된 계수들을 코딩하여, 출력 비트스트림에 대해 동일한 바를 제공한다.
- [0049] 도 4를 참조하면, 4×4 크기의 일례의 양자화된 변환 블록이 참조 부호(400)로 일반적으로 표시된다. 예측, 변환, 및 양자화 후에, 블록의 대부분의 에너지는 저 빈도 위치들(변환 블록의 상부 좌측 코너에 있음)에 집중되고, 고 빈도 계수들(블록의 하부 우측 코너에 있음)의 대부분은 제로들이다. 이러한 데이터 블록을 엔트로피 코딩하기 위해, 효율적으로 바이너리 빈들로 블록 내부의 계수 값들 및 그 위치들을 포함하는 블록 정보를 표현할 필요가 있다. 그 후, 바이너리 빈들은 바이너리 산술 코딩 엔진으로 코딩된다.
- [0050] 바이너리 빈들로 블록 정보를 표현하기 위해, 이하의 구문들을 사용한다. 설명의 편의성을 위해, 구문 요소들의 일부는 CABAC 등의 기존 방법들로부터 차용된 것이다. 또한, 새로운 구문 요소들은 다음과 같이 본 발명을 가능케 하기 위해 소개된다:
- [0051]  **Sig_flag:** CABAC와 동일하게 정의된다. Sig_flag = 1은 대응 계수가 논제로(시그니피컨트)임을 의미한다. Sig_flag = 0은 대응 계수가 제로임을 의미한다.
- [0052]  **Last_ge2_flag:** 현재 논제로 계수가 주어진 스캔 순서에 의해 현재 블록에서 1 보다 더 큰 절대값을 가진 최종 계수인 지의 여부를 나타내기 위해 본 발명에 따라 소개된 새로운 구문 요소이다. 표현 "ge2"는 "Greater or Equal to 2(2 보다 더 크거나 또는 동일함)"에서 비롯된다. Last_ge2_flag = 1은 현재 논제로 계수가 최종 계수임을 의미한다. Last_ge2_flag = 0은 현재 논제로 계수가 최종 계수가 아님을 의미한다.
- [0053]  **Last_flag:** CABAC와 동일하게 정의된다. Last_flag는, 현재 논제로 계수가 주어진 스캔 순서에서 현재 블록의 최종 계수인 지의 여부를 의미한다. Last_flag = 1은 현재 논제로 계수가 최종 계수임을 의미한다. Last_flag = 0은 현재 논제로 계수가 최종 계수가 아님을 의미한다.
- [0054]  **Bin_1:** 계수가 논제로라고 알려지지만, 절대값 면에서는 1이거나 또는 1 보다 더 크다(ge2)고 인식될 때,

bin₁이 명확하게 하기 위해 송신된다. Bin₁ = 1은 논제로 계수가 절대값 1을 가짐을 의미한다. Bin₁ = 0은 논제로 계수가 1 보다 더 큰 절대값(ge2)을 가짐을 의미한다. 2, 3 또는 더 큰 수 등의 다른 계수 값들로서 이 정보를 코딩하는 대신, 계수가 1 또는 1 보다 더 큰 절대값을 가짐을 나타내기 위해 Bin₁을 사용함을 주지하라. 이는 전형적인 데이터 블록에서, 논제로들의 대략 절반은 절대값 1을 가져서, 구체적으로 처리하기에 더 효율적이기 때문이다.




- [0055]  **Level:** 계수가 1 보다 더 큰 절대값(ge2)을 가진다고 인식될 때, 절대값인 level을 송신한다. 원래, 이 level은 바이너리가 아니어서, 예를 들어, 바이너리화하기 위해 CABAC에서 사용되는 UEGO 방법 - 이로만 제한되지 않음 - 등의 일부 바이너리화 방법을 사용하고, 그 후 바이너리 산술 코딩으로 바이너리 빈들을 코딩한다.
- [0056]  **Sign:** 모든 논제로 계수에 대해, sign은 는, 각각, "+" 및 "-"에 대해 0 또는 1로서 송신된다.
- [0057] 도 5를 참조하면, 코딩 프로세스의 일례가 참조 부호(500)로 일반적으로 표시된다. 코딩 프로세스(500)는 도 4의 일례의 블록(400)의 코딩에 대하여 기술된다. 주어진 스캔 순서로, 예를 들어, CABAC의 전방향 재그재그 스캔으로 데이터를 스캔한다. 재구성된 계수들은 도 5의 제1 행으로 제공된다.
- [0058]  제1 계수 "10"의 경우, 논제로이고(sig_flag = 1), 최종 ge2가 아니다(last_ge2_flag = 0). 값을 코딩하기 위해, 먼저 Bin₁ = 0을 송신해서, 절대값이 1 보다 더 큼을 나타낸다. 그 후, level을 가진 절대값을 코딩한다. 여기서, 오직 10-2 = 8을 코딩하고, 디코더는, 절대값이 8+2 = 10 이어야만 한다고 인식한다. 마지막으로, 부호 "+"를 0과 함께 발송한다.
- [0059]  제2 계수 "0"의 경우, sig_flag = 0 이다. 그 후, 이 계수에 대한 모든 정보가 발송되었고, 인코더는 다음 계수를 처리하도록 이동한다.
- [0060]  다음 계수 "-1"의 경우, 논제로이고(sig_flag = 1), 최종 ge2가 아니다(last_ge2_flag = 0). 값을 코딩하기 위해, 먼저 Bin₁ = 0을 송신해서, 절대값이 1임을 나타내고, 그 후, level을 처리할 필요가 없다. 마지막으로, 부호 "-"를 1과 함께 발송한다.
- [0061]  다음 계수 "2"의 경우, 논제로이고(sig_flag = 1), 최종 ge2이다(last_ge2_flag = 1). last_ge2_flag = 1 후에, last_flag를 발송해서 현재 계수가 최종 논제로 계수인 지의 여부를 나타낼 필요가 있다. 여기서, 최종 논제로 계수가 아니어서, last_flag = 0 이다. last_ge2_flag = 1은, 이 계수가 1 보다 더 큰 절대값을 가져야만 함(즉, ge2 이어야만 함)을 암암리에 나타내서, Bin₁ = 0 이 저장됨을 주지하라. 2-2=0을 코딩함으로써 level을 가진 절대값을 발송하고, 그 후, 디코더는, 절대값이 0+2=2 임을 인식한다. 마지막으로, 부호 "+"를 0과 함께 발송한다.
- [0062]  다음 계수 "0"의 경우, sig_flag = 0 이다.
- [0063]  다음 계수 "1"의 경우, 논제로이고(sig_flag = 1), 최종 논제로 계수가 아니다(last_flag = 0). last_ge2_flag = 1 후에, 모든 시그니피칸트 계수들은 절대값 1을 가져야만 해서, 더 이상 Bin₁ 또는 level을 가진 절대값을 코딩할 필요가 없다. 부호 "+"를 0과 함께 발송하기만 하면 된다.
- [0064]  다음 계수 "0"의 경우, sig_flag = 0 이다.
- [0065]  다음 계수 "-1"의 경우, 논제로이고(sig_flag = 1), 최종 논제로 계수이다(last_flag = 1). 부호 "-"를 1과 함께 발송한 후에, 이 블록의 코딩은 완료된다.
- [0066] 상술된 코딩 일례로부터, 기술된 실시예의 적어도 하나의 신규 양상은 last_ge2_flag의 사용임을 알 수 있다. 적어도 이하를 포함해서, last_ge2_flag의 수개의 장점들이 있다:
- [0067] 1. Last_ge2_flag는 last_flag를 저장한다. last_ge2_flag = 0이면, 이하의 스캔 위치들에서 논제로(구체적으로 1 보다 큼) 계수들이어야만 한다. 그 후, last_flag는 0이어야만 해서, last_ge2_flag = 1 일 때까지 last_flag들을 저장한다.
- [0068] CABAC에 의한 및 제안된 방법에 의한 동일한 일례의 블록의 코딩을 비교하라. 도 1에서, CABAC에 의한 블록의 코딩은, 00001인, 5개의 last_flag들을 필요로 하고, 도 5에서, 새로운 방법에 의한 동일한 블록의 코딩은 3개의 last_ge2_flag들(001) 및 3개의 last_flag들(001)을 필요로 한다. 블록에 N개의 논제로 계수들이 있다고





추정하면, CABAC는 N개의 last_flag들을 필요로 하고, 새로운 방법은 총 N+1개의 last_ge2_flag들 및 last_flag들을 필요로 한다. last_ge2_flag에 의한 다음 2개의 절약들에 비해, 이 하나의 여분의 플래그가 꽤 가치가 있다.

- [0069] 2. 스캔 경로의 끝에서, 연속의 소위 트레일링(trailing) 계수들, 즉, 1과 동일한 절대값을 가진 변환 계수 레벨들의 발생을 목격할 개연성이 높다. 예를 들어, 도 5에서, 5개의 시그니피컨트 계수들: 10, -1, 2, 1, -1이 있으며, 끝에 있는 "1" 및 "-1"은 트레일링 계수들이다. Last_ge2_flag는 CABAC의 트레일링 계수들을 위해 Bin_1들을 저장한다. last_ge2_flag = 1 후에, 일부 계수들이 논제로들로 표시되면, 절대값 1을 가져야만 한다. CABAC 코딩에는 실제로 트레일링 계수들이 있다. CABAC에서, 각각의 트레일링 계수는 (ge2 대신) 1임을 나타내기 위해 하나의 Bin_1을 필요로 한다. last_ge2_flag = 1로 암암리에 나타내기 때문에, 트레일링 계수들에 대한 Bin_1들이 저장된다. 큰 변환 블록들에는, 비교적 큰 수의 트레일링 계수들이 있어서, 저장에 시그니피컨트하다.
- [0070] 3. Last_ge2_flag는 논-트레일링(non-trailing) 계수들을 위해 다른 Bin_1을 저장한다. last_ge2_flag가 일부 계수에서 0으로부터 1로 변할 때, 계수는 ge2이어야만 해서(즉, 계수가 1 보다 더 큰 절대값을 가져야만 함), 0이어야만 하는, Bin_1을 송신할 필요가 없다. 일례는 도 1의 코딩 일례의 계수 2이다.
- [0071] 이러한 오직 Bin_1만의 저장은, last_ge2_flag가 0으로부터 1로 변하는 블록에만 존재한다. 즉, 블록에 대해 1 보다 더 많은 last_ge2_flag들이 송신된다. 블록에 대해 오직 하나의 last_ge2_flag가 송신된다고 가정하면, 1 이어야만 하고, 대응 계수는 1 또는 ge2의 절대값을 가질 수 있다. 일부 일례의 경우들이 도 6에 제공되며, 여기서, 계수들은 일부 주어진 스캔 순서로 배열된다.
- [0072] 도 6을 참조하면, Bin_1이 저장되지 않은 일례의 특별 경우들이 참조 부호(600)로 일반적으로 표시된다. 또한, 4개의 특별 경우들이 있으며, 각각은 경우 1, 경우 2, 경우 3, 및 경우 4로 각각 표시된다. 경우 1 및 2에서, 모든 시그니피컨트 계수들의 절대값들은 2 보다 더 작다. 경우 3 및 4에서, 제1 시그니피컨트 계수만이 1 보다 더 큰 절대값을 가진다. 모든 경우들에서, last_ge2_flag는 제1 시그니피컨트 계수에서 1로 설정되어서, last_ge2_flag에 대해 "0으로부터 1로의 변화"가 없으며, 오직 하나의 빈이 last_ge2_flag에 대해 사용된다. 따라서, last_ge2_flag = 1인 계수는 (경우 1 및 경우 2에서와 같이) 1이거나 또는 (경우 3 및 경우 4에서와 같이) 1 보다 더 클 수 있으며, 이는 Bin_1에 의해 표시되어야만 한다. 경우 1 및 경우 2에서, 제1 시그니피컨트 계수 "1"은 last_ge2_flag = 1 및 Bin_1 = 1 을 가진다. 경우 3 및 경우 4에서, 제1 시그니피컨트 계수 2(또는 2 보다 일반적으로 더 크거나 또는 2와 동일함)는 last_ge2_flag = 1 및 Bin_1 = 0을 가진다. 즉, 블록에 대한 last_ge2_flag가 1이어야만 하는 오직 하나의 빈을 포함할 때, 대응 Bin_1은 인코딩될 필요가 있다.
- [0073] 도 7a를 참조하면, 엔트로피 인코딩의 일례의 방법이 참조 부호(700)로 일반적으로 표시된다. 방법은 결정 블록(701)으로 제어를 넘기는 개시 블록(712)을 포함한다. 결정 블록(701)은, 블록에 시그니피컨트 계수들이 있는 지를 결정한다. 있으면, 제어는 기능 블록(702)으로 넘겨진다. 없으면, 제어는 종료 블록(799)으로 넘겨진다. 기능 블록(702)은 last_ge2_flag = 0 및 last_flag = 0 으로 설정하고, 제어를 기능 블록(703)으로 넘긴다. 기능 블록(703)은, last_flag = 0이면, 1 내지 계수들의 수(#)의 범위를 가진 변수 j를 사용해서 루프를 시작하고, 제어를 기능 블록(704)으로 넘긴다. 기능 블록(704)은 sig_flag를 인코딩하고, 제어를 결정 블록(705)으로 넘긴다. 결정 블록(705)은 sig_flag = 1인 지를 결정한다. 그렇다면, 제어는 기능 블록(706)으로 넘겨진다. 그렇지 않으면, 제어는 루프 제한 블록(711)으로 넘겨진다. 기능 블록(706)은 필요한 경우 last_ge2_flag를 인코딩하고, 제어를 기능 블록(707)으로 넘긴다. 기능 블록(707)은 필요한 경우 last_flag를 인코딩하고, 제어를 기능 블록(708)으로 넘긴다. 기능 블록(708)은 필요한 경우 Bin_1을 인코딩하고, 제어를 기능 블록(709)으로 넘긴다. 기능 블록(709)은 필요한 경우 level을 인코딩하고, 제어를 기능 블록(710)으로 넘긴다. 기능 블록(710)은 sign을 인코딩하고, 제어를 루프 제한 블록(711)으로 넘긴다. 루프 제한 블록(711)은 루프를 종료하고, 제어를 종료 블록(799)으로 넘긴다.
- [0074] 기능 블록(703)에 관하여, 몇몇 스캔 순서로 블록의 계수들에 대해 루프한다. last_flag = 1인 계수 후에, 계수들에 대한 루프가 필요 없다. 결정 블록(705)에 관하여, sig_flag = 1(시그니피컨트)이면, 블록들(706-710)에 의해 계수를 더 코딩한다. 그렇지 않으면, 다음 계수에 대해 루프한다. 기능 블록(706)에 관하여, last_ge2_flag를 동일하게 처리하고, last_ge2_flag의 프로세싱은 도 7b와 관련하여 더 기술됨을 주지하라. 기능 블록(707)에 관하여, last_flag를 동일하게 처리하고, last_flag의 프로세싱은 도 7c와 관련하여 더 기술됨을 주지하라. 기능 블록(708)에 관하여, Bin_1을 동일하게 처리하고, Bin_1의 프로세싱은 도 7d와 관련하여 더 기술됨을 주지하라. 기능 블록(709)에 관하여, level을 동일하게 처리하고, level의 프로세싱은 도 7e와 관련

하여 더 기술됨을 주지하라.

- [0075] 도 7b를 참조하면, last_ge2_flag를 인코딩하기 위한 일례의 방법이 참조 부호(720)로 일반적으로 표시된다. 방법(720)은 결정 블록(721)으로 제어를 넘기는 개시 블록(719)을 포함한다. 결정 블록(721)은 last_ge2_flag = 0인 지를 결정한다. 그렇다면, 제어는 결정 블록(722)으로 넘겨진다. 그렇지 않으면, 제어는 종료 블록(798)으로 넘겨진다. 결정 블록(722)은 현재 계수 후에 1 보다 더 큰 계수들이 있는 지를 결정한다. 그렇다면, 제어는 기능 블록(723)으로 넘겨진다. 그렇지 않으면, 제어는 기능 블록(724)으로 넘겨진다. 기능 블록(723)은 last_ge2_flag = 0으로 설정하고, 제어를 결정 블록(725)으로 넘긴다. 결정 블록(724)은 last_ge2_flag = 1로 설정하고, 제어를 결정 블록(725)으로 넘긴다. 결정 블록(725)은, 현재 스캔 위치가 최종 스캔 위치인 지를 결정한다. 그렇다면, 제어는 종료 블록(798)으로 넘겨진다. 그렇지 않으면, 제어는 기능 블록(726)으로 넘겨진다. 기능 블록(726)은 last_ge2_flag를 인코딩하고, 제어를 종료 블록(798)으로 넘긴다.
- [0076] 도 7c를 참조하면, last_flag를 인코딩하기 위한 일례의 방법이 참조 부호(730)로 일반적으로 표시된다. 방법은 결정 블록(731)으로 제어를 넘기는 개시 블록(729)을 포함한다. 결정 블록(731)은 last_ge2_flag = 1인 지를 결정한다. 그렇다면, 제어는 결정 블록(732)으로 넘겨진다. 그렇지 않으면, 제어는 종료 블록(797)으로 넘겨진다. 결정 블록(732)은 현재 계수 후에 시그니피컨트 계수들이 있는 지를 결정한다. 그렇다면, 제어는 기능 블록(733)으로 넘겨진다. 그렇지 않으면, 제어는 기능 블록(734)으로 넘겨진다. 기능 블록(733)은 last_flag = 0으로 설정하고, 제어를 결정 블록(735)으로 넘긴다. 기능 블록(734)은 last_flag = 1로 설정하고, 제어를 결정 블록(735)으로 넘긴다. 결정 블록(735)은, 현재 스캔 위치가 최종 스캔 위치인 지를 결정한다. 그렇다면, 제어는 종료 블록(797)으로 넘겨진다. 그렇지 않으면, 제어는 기능 블록(736)으로 넘겨진다. 기능 블록(736)은 last_flag를 인코딩하고, 제어를 종료 블록(797)으로 넘긴다.
- [0077] 도 7d를 참조하면, Bin_1을 인코딩하기 위한 일례의 방법이 참조 부호(740)로 일반적으로 표시된다. 방법(740)은 결정 블록(741)으로 제어를 넘기는 개시 블록(739)을 포함한다. 결정 블록(741)은 (last_ge2_flag = 0)인 지를 또는 (블록의 제1 시그니피컨트 계수인 현재 계수에 대해 last_ge2_flag = 1)인 지를 결정한다. 그렇다면, 제어는 결정 블록(742)으로 넘겨진다. 그렇지 않으면, 제어는 종료 블록(796)으로 넘겨진다. 결정 블록(742)은 변환 계수의 절대값이 1(Abs(currCoeff) = 1)인 지를 결정한다. 그렇다면, 제어는 기능 블록(743)으로 넘겨진다. 그렇지 않으면, 제어는 기능 블록(744)으로 넘겨진다. 기능 블록(743)은 Bin_1 = 1로 설정하고, 제어를 기능 블록(745)으로 넘긴다. 기능 블록(744)은 Bin_1 = 0으로 설정하고, 제어를 기능 블록(745)으로 넘긴다. 기능 블록(745)은 Bin_1을 인코딩하고, 제어를 종료 블록(796)으로 넘긴다.
- [0078] 도 7e를 참조하면, level을 인코딩하기 위한 일례의 방법이 참조 부호(750)로 일반적으로 표시된다. 방법(750)은 결정 블록(751)으로 제어를 넘기는 개시 블록(749)을 포함한다. 결정 블록(751)은 변환 계수의 절대값이 2 보다 크거나 동일한(Abs(currCoeff) >= 2) 지를 결정한다. 그렇다면, 제어는 기능 블록(752)으로 넘겨진다. 그렇지 않으면, 제어는 종료 블록(795)으로 넘겨진다. 기능 블록(752)은 level을 인코딩하고, 여기서, level = abs(currCoeff) - 2 이며, 제어를 종료 블록(795)으로 넘긴다.
- [0079] 방법(700)의 코딩 순서는 도 5의 일례의 코딩 순서와 동일하다. 그러나, 코딩 순서는 이하의 규칙들을 만족시키는 한 블록들(706-710)에 대해 유연할 수 있다:
- [0080] (기능 블록(706)에 대한) Last_ge2_flag는 (기능 블록(708)에 대한) Bin_1 전에 처리되고, (기능 블록(708)에 대한) Bin_1은 (기능 블록(709)에 대한) level 전에 처리된다. 즉, 706 → 708 → 709.
- [0081] (기능 블록(707)에 대한) last_flag의 프로세싱은 이하의 기능 블록들(706, 708, 또는 709)에 이어질 수 있다.
- [0082] (기능 블록(710)에 대한) sign의 프로세싱은 4개의 기능 블록들(706, 707, 708, 709) 중 임의의 기능 블록의 전에 또는 후에 실행될 수 있다.
- [0083] 디코더의 프로세싱 순서는 인코더의 프로세싱 순서와 일치해야만 한다.
- [0084] 블록의 최종 스캔 위치에서 계수를 인코딩할 때 특별 처리가 있음을 주지하라.
- [0085] 방법(720)에서, 최종 계수 전에 last_ge2_flag가 여전히 0이면, last_ge2_flag는 최종 계수에 대해 1이어야만 해서, last_ge2_flag는 인코딩될 필요가 없다.

- [0086]  방법(730)에서, 유사하게, 최종 계수 전에 last_flag가 여전히 0이면, last_flag는 최종 계수에 대해 1이어야만 해서, last_flag는 인코딩될 필요가 없다.
- [0087]  Bin_1(740): 기능 블록(741)에 관하여, (기능 블록(726)에 의해 인코딩 및 송신될 필요가 없더라도) 최종 계수에 대한 기능 블록(724)에 의해 last_ge2_flag가 1로 설정되고, 블록의 제1 시그니피컨트 계수이면, Bin_1은 블록들(742-745)에 대해 테스트 및 인코딩되어야만 한다.
- [0088]  (기능 블록(709)에 대한) level 정보는 필요한 경우 인코딩되어야만 한다.
- [0089] 도 8a를 참조하면, 엔트로피 디코딩의 일례의 방법이 참조 부호(800)로 일반적으로 표시된다. 방법(800)은 결정 블록(801)으로 제어를 넘기는 개시 블록(819)을 포함한다. 결정 블록(801)은, 블록에 시그니피컨트 계수들이 있는 지를 결정한다. 있으면, 제어는 기능 블록(802)으로 넘겨진다. 없으면, 제어는 종료 블록(899)으로 넘겨진다. 기능 블록(802)은 last_ge2_flag = 0 및 last_flag = 0 으로 설정하고, 제어를 기능 블록(803)으로 넘긴다. 기능 블록(803)은, last_flag = 0이면, 1 내지 계수들의 수(#)의 범위를 가진 변수 j를 사용해서 루프를 시작하고, 제어를 기능 블록(804)으로 넘긴다. 기능 블록(804)은 sig_flag를 디코딩하고, 제어를 결정 블록(805)으로 넘긴다. 결정 블록(805)은 sig_flag = 1인 지를 결정한다. 그렇다면, 제어는 기능 블록(806)으로 넘겨진다. 그렇지 않으면, 제어는 루프 제한 블록(811)으로 넘겨진다. 기능 블록(806)은 필요한 경우 last_ge2_flag를 디코딩하고, 제어를 기능 블록(807)으로 넘긴다. 기능 블록(807)은 필요한 경우 last_flag를 디코딩하고, 제어를 기능 블록(808)으로 넘긴다. 기능 블록(808)은 필요한 경우 Bin_1을 디코딩하고, 제어를 기능 블록(809)으로 넘긴다. 기능 블록(809)은 필요한 경우 level을 디코딩하고, 제어를 기능 블록(810)으로 넘긴다. 기능 블록(810)은 sign을 디코딩하고, 제어를 루프 제한 블록(811)으로 넘긴다. 기능 블록(811)은 루프를 종료하고, 제어를 종료 블록(899)으로 넘긴다.
- [0090] 기능 블록(803)에 관하여, 인코더와 동일한 스캔 순서로 블록의 계수들에 대해 루프한다. last_flag = 1인 계수 후에 계수들에 대한 루프가 필요 없다. 기능 블록(806)에 관하여, last_ge2_flag를 동일하게 처리하고, last_ge2_flag의 프로세싱은 도 8b와 관련하여 더 기술됨을 주지하라. 기능 블록(807)에 관하여, last_flag를 동일하게 처리하고, last_flag의 프로세싱은 도 8c와 관련하여 더 기술됨을 주지하라. 기능 블록(808)에 관하여, Bin_1을 동일하게 처리하고, Bin_1의 프로세싱은 도 8d와 관련하여 더 기술됨을 주지하라. 기능 블록(809)에 관하여, level을 동일하게 처리하고, level의 프로세싱은 도 8e와 관련하여 더 기술됨을 주지하라.
- [0091] 일 실시예에서, 방법(800)의 디코딩 순서는 방법(700)의 인코딩 순서와 일치함을 알 것이다. 그러나, 디코딩 순서는 그에 대한 인코딩 순서와 일치하는 한 기능 블록들(806-810)에 대해 유연할 수 있다.
- [0092] 도 8b를 참조하면, last_ge2_flag를 디코딩하기 위한 일례의 방법이 참조 부호(820)로 일반적으로 표시된다. 방법(820)은 결정 블록(821)으로 제어를 넘기는 개시 블록(812)을 포함한다. 결정 블록(821)은 last_ge2_flag = 0인 지를 결정한다. 그렇다면, 제어는 결정 블록(822)으로 넘겨진다. 그렇지 않으면, 제어는 종료 블록(898)으로 넘겨진다. 결정 블록(822)은 현재 스캔 위치가 최종 스캔 위치인 지를 결정한다. 그렇다면, 제어는 블록(823)으로 넘겨진다. 그렇지 않으면, 제어는 기능 블록(824)으로 넘겨진다. 기능 블록(823)은 last_ge2_flag = 1로 설정하고, 제어를 종료 블록(898)으로 넘긴다. 결정 블록(824)은 last_ge2_flag를 디코딩하고, 제어를 종료 블록(898)으로 넘긴다.
- [0093] 도 8c를 참조하면, last_flag를 디코딩하기 위한 일례의 방법이 참조 부호(830)로 일반적으로 표시된다. 방법(830)은 결정 블록(831)으로 제어를 넘기는 개시 블록(825)을 포함한다. 결정 블록(831)은 last_ge2_flag = 1인 지를 결정한다. 그렇다면, 제어는 결정 블록(832)으로 넘겨진다. 그렇지 않으면, 제어는 종료 블록(897)으로 넘겨진다. 결정 블록(832)은 현재 스캔 위치가 최종 스캔 위치인 지를 결정한다. 그렇다면, 제어는 종료 블록(897)으로 넘겨진다. 그렇지 않으면, 제어는 기능 블록(833)으로 넘겨진다. 기능 블록(833)은 last_flag를 디코딩하고, 제어를 종료 블록(897)으로 넘긴다.
- [0094] 도 8d를 참조하면, Bin_1을 디코딩하기 위한 일례의 방법이 참조 부호(840)로 일반적으로 표시된다. 방법(840)은 기능 블록(841)으로 제어를 넘기는 개시 블록(834)을 포함한다. 기능 블록(841)은 Bin_1 = 1을 설정하고, 제어를 결정 블록(842)으로 넘긴다. 결정 블록(842)은 (last_ge2_flag = 0)인 지를 또는 (블록의 제1 시그니피컨트 계수인 현재 계수에 대해 last_ge2_flag = 1)인 지를 결정한다. 그렇다면, 제어는 기능 블록(843)으로 넘겨진다. 그렇지 않으면, 제어는 종료 블록(896)으로 넘겨진다. 기능 블록(843)은 Bin_1을 디코딩하고, 제어를 종료 블록(896)으로 넘긴다.

- [0095] 도 8e를 참조하면, level을 디코딩하기 위한 일례의 방법이 참조 부호(850)로 일반적으로 표시된다. 방법(850)은 결정 블록(851)으로 제어를 넘기는 개시 블록(844)을 포함한다. 결정 블록(851)은 ($\text{Bin}_1 = 0$)인 지를 또는 (블록의 제1 시그니피컨트 계수인 현재 계수에 대해 $\text{last_ge2_flag} = 1$)인 지를 결정한다. 그렇다면, 제어는 기능 블록(852)으로 넘겨진다. 그렇지 않으면, 제어는 기능 블록(853)으로 넘겨진다. 기능 블록(852)은 level을 디코딩하고, 현재 계수의 절대값을 $\text{level} + 2(\text{Abs}(\text{currCoeff}) = \text{level} + 2)$ 로 설정하고, 제어를 종료 블록(895)으로 넘긴다. 기능 블록(853)은 현재 계수의 절대값을 $1(\text{abs}(\text{currCoeff}) = 1)$ 로 설정하고, 제어를 종료 블록(895)으로 넘긴다.
- [0096] 블록의 최종 스캔 위치에서 계수를 디코딩할 때 특별 처리가 있음을 주지하라.
- [0097]  방법(820)에서, 최종 계수 전에 last_ge2_flag 가 여전히 0이면, last_ge2_flag 는 최종 계수에 대해 1이어야만 해서, last_ge2_flag 는 디코딩될 필요가 없다. 대신, 기능 블록(823)에 대해 단지 1로 설정된다.
- [0098]  방법(830)에서, 유사하게, 최종 계수 전에 last_flag 가 여전히 0이면, last_flag 는 최종 계수에 대해 1이어야만 해서, last_flag 는 디코딩될 필요가 없다.
- [0099]  (방법(840)에 대한) Bin_1 : 기능 블록(842)에 관하여, (기능 블록(824)에 의해 디코딩될 필요가 없더라도) 823에서 최종 계수에 대해 last_ge2_flag 가 1로 설정되고, 블록의 제1 시그니피컨트 계수이면, Bin_1 은 기능 블록(843)에 의해 디코딩되어야만 한다.
- [0100]  (방법(850)에 대한) level: 851에서, $\text{Bin}_1 = 0$ 이면, 또는 (기능 블록(824)에 의해 디코딩될 필요가 없더라도) 기능 블록(823)에 의해 최종 계수에 대해 last_ge2_flag 가 1로 설정되고, 최종 계수가 제1 시그니피컨트 계수가 아니면, level은 기능 블록(852)에 의해 디코딩되어야만 한다. 그렇지 않으면, 기능 블록(853)에 의해 이 최종 계수의 절대값은 1로 설정하라.
- [0101] 제안된 방법의 다른 장점은, sig_flag , last_flag 등의 다른 구문들의 동일한 스캔 패스의 계수 레벨의 코딩이다. CABAC에서, 계수 레벨 정보는 역 지그재그 스캔 순서로 코딩되며, 이는 블록을 코딩하기 위한 제2 패스로서 알 수 있다. 역 지그재그 순서 코딩에서, 레벨 정보는 코딩된 계수들의 역 지그재그 위치로 설계된 문맥 모델들로 코딩될 수 있다. 구체적으로 말하면, 제1 코딩 계수(역 지그재그 순서)는 문맥 모델 0으로 코딩되고, 제2 코딩 계수는 문맥 모델 1로 코딩되는 등이다. 이 문맥 모델 설계는 동일한 확률(0.5/0.5) 모델들(즉, 빈당 1 비트로 직접 bin들을 출력함)에 의한 레벨 bin들의 코딩에 비해 약간의 이득을 보여준다.
- [0102] 이 작은 페널티는 1 패스 코딩 방법의 적절히 설계된 문맥 모델들에 의해 쉽게 보상될 수 있다. 코딩 계수들의 계수 레벨 정보가 주어지면, sig_flag , last_ge2_flag , last_flag , Bin_1 및 level bin들의 문맥 모델들은 문맥 모델들의 성능을 더 향상시키고 더 높은 코딩 효율을 달성하기 위해 코딩된 이웃들로부터 인식된 레벨 정보에 기초하여 설계될 수 있다.
- [0103] 도 9를 참조하면, 현재 변환 계수의 값을 선택 및 시그널링하는 방법이 참조 부호(900)로 일반적으로 표시된다. 방법(900)은 기능 블록(910)으로 제어를 넘기는 개시 블록(905)을 포함한다. 기능 블록(910)은 픽처들 및 값들의 집합을 입력하고, 제어를 기능 블록(920)으로 넘긴다. 기능 블록(920)은 미리 처리된 블록들 또는 픽처들의 통계들에 기초하여 값을 적응적으로 선택하고, 제어를 기능 블록(930)으로 넘긴다. 기능 블록(930)은 명확하게 시퀀스 레벨, 프레임 레벨, 슬라이스 레벨, 또는 블록 레벨에서 선택된 값을 시그널링하고, 제어를 종료 블록(999)으로 넘긴다.
- [0104] 도 10을 참조하면, 현재 변환 계수의 값을 디코딩하는 일례의 방법이 참조 부호(1000)로 일반적으로 표시된다. 방법(1000)은 기능 블록(1010)으로 제어를 넘기는 개시 블록(1005)을 포함한다. 기능 블록(1010)은 시퀀스 레벨, 프레임 레벨, 슬라이스 레벨, 또는 블록 레벨에서 값을 디코딩하고, 제어를 종료 블록(1099)으로 넘긴다.
- [0105] 따라서, 본 발명은 양자화된 변환 블록을 체계적으로 코딩하는 향상된 엔트로피 인코딩 및 디코딩을 위한 방법들 및 장치들을 유익하게 제공한다. 이 방식과 연관된 적어도 2개의 신규성들이 있다. 첫째, 소개된 last_ge2_flag 는 인코더 및 디코더에서 바이너리 산술 코딩될 바이너리 bin들을 감소시키며, 따라서, 더 간단한 시스템이 달성된다. 둘째, 다른 구문들과 동일한 스캔 순서로의 계수 값 정보의 코딩은 블록의 엔트로피 코딩이 하나의 스캔 패스로 완성되게 한다. 값 정보는 구문들의 문맥 모델들을 향상시키는데 사용되어서, 더 높은 코딩 효율을 더 달성할 수 있다.
- [0106] last_ge2_flag 가 단지 일 실시예임을 알 것이다. 당업자에 의해 쉽게 명백해지는 바와 같이, 플래그는 대안으

로 `last_geX_flag`라 할 수 있으며, X 는 임의의 수이다.

- [0107] 본 발명의 수반되는 다수의 장점들/특징들 중 일부가 이제 설명될 것이며, 그 중 일부는 상술되었다. 예를 들어, 한 장점/특징은, 블록의 잔여물을 변환해서 변환 계수들을 획득하고, 변환 계수들을 양자화해서 양자화된 변환 계수들을 획득하며, 양자화된 변환 계수들을 엔트로피 코딩함으로써, 한 픽처에서 적어도 한 블록을 인코딩하기 위한 비디오 인코더를 가진 장치이다. 양자화된 변환 계수들은 처리되고 있는 양자화된 변환 계수들 중 현재 계수가 지정된 값보다 더 크거나 또는 동일한 값을 가진 블록의 최종 논제로 계수임을 나타내기 위해 플래그를 사용해서 단일 패스로 인코딩된다.
- [0108] 다른 장점/특징은 상술된 비디오 인코더를 가진 장치이며, 지정된 값은 2이다.
- [0109] 또 다른 장점/특징은 상술된 비디오 인코더를 가진 장치이며, 지정된 값보다 더 작은 값을 가진 양자화된 변환 계수들 중에서 후속의 논제로 계수들이 지정된 값보다 더 작은 값을 가진 후속의 논제로 계수들의 각각의 부호들만을 인코딩함으로써 인코딩된다.
- [0110] 또 다른 장점/특징은 상술된 비디오 인코더를 가진 장치이며, 지정된 값은 복수의 값들로부터 선택된다.
- [0111] 또한, 다른 장점/특징은 상술된 비디오 인코더를 가진 장치이며, 지정된 값은 상술된 바와 같이 복수의 값들로부터 선택되고, 픽처는 비디오 시퀀스에 포함된 복수의 픽처 중 하나이며, 지정된 값은 상기 픽처에서 또는 비디오 시퀀스의 복수의 픽처 중에서의 하나의 또는 그 이상의 다른 픽처들에서 미리 처리된 블록들로부터 유도된 통계들에 대응해서 적응적으로 선택된다.
- [0112] 또한, 다른 장점/특징은 상술된 비디오 인코더를 가진 장치이며, 지정된 값은 명확하게 시그널링된다.
- [0113] 또한, 다른 장점/특징은 상술된 비디오 인코더를 가진 장치이며, 지정된 값은 시퀀스 레벨, 프레임 레벨, 슬라이스 레벨, 및 블록 레벨 중 적어도 하나에서 명확하게 시그널링된다.
- [0114] 또한, 다른 장점/특징은 상술된 비디오 인코더를 가진 장치이며, 양자화된 변환 계수들 중 현재 계수의 레벨은, 차이 값을 지정된 값에 추가함으로써 대응 디코더에서 레벨을 재생하기 위해, 양자화된 변환 계수들 중 현재 계수의 실제 값으로부터 지정된 값을 추출하여 차이 값을 획득하고 차이 값을 레벨로서 인코딩함으로써 인코딩된다.
- [0115] 또한, 다른 장점/특징은 상술된 비디오 인코더를 가진 장치이며, 적어도 `sig_flag` 구문 요소, 플래그, `last_flag` 구문 요소, `Bin_1` 구문 요소, `level` 구문 요소, 및 `sign` 구문 요소가 동일한 스캔 순서로 인코딩되고, `sig_flag` 구문 요소는 양자화된 변환 계수들 중 현재 계수가 논제로 값을 가지는 지를 나타내며, `last_flag`는 논제로 값을 가진 양자화된 변환 계수들 중 현재 계수가 주어진 스캔 순서로 블록의 논제로 값을 가진 최종 양자화된 변환 계수인 지를 나타내고, `Bin_1` 구문 요소는 양자화된 변환 계수들 중 현재 계수의 절대값이 현재 알려지지 않은 논제로 값을 가짐을 나타내며, `level` 구문 요소는 양자화된 변환 계수들 중 현재 계수가 지정된 값보다 더 큰 절대값을 가질 때의 양자화된 변환 계수들 중 현재 계수의 절대값을 나타내고, `sign` 구문 요소는 양자화된 변환 계수들 중 현재 계수의 대응 부호를 나타낸다.
- [0116] 본 발명의 여타 특징들 및 장점들은 본 명세서의 교시에 기초하여 당업자에 의해 쉽게 확인될 수 있다. 본 발명의 교시는 하드웨어, 소프트웨어, 펌웨어, 특별 목적 프로세서들, 또는 그 조합들의 각종 형태들로 구현될 수 있다.
- [0117] 더 양호하게, 본 발명의 교시는 하드웨어 및 소프트웨어의 조합으로서 구현된다. 더욱이, 소프트웨어는 프로그램 스토리지 유닛에 유형으로 구현된 애플리케이션 프로그램으로서 구현될 수 있다. 애플리케이션 프로그램은 임의의 적합한 아키텍처를 포함하는 기계에 업로딩되어, 기계에 의해 실행될 수 있다. 양호하게, 기계는 하나의 또는 그 이상의 중앙 처리 장치들("CPU"), 랜덤 액세스 메모리("RAM"), 및 입력/출력("I/O") 인터페이스들 등의 하드웨어를 가진 컴퓨터 플랫폼에서 구현된다. 컴퓨터 플랫폼은 또한 운영 체제 및 마이크로명령 코드들을 포함할 수 있다. 본 명세서에 기술된 각종 프로세스들 및 기능들은 CPU에 의해 실행될 수 있는, 마이크로명령 코드의 파트 또는 애플리케이션 프로그램의 파트이거나, 또는 임의의 그 조합일 수 있다. 또한, 각종 다른 주변 유닛들은 추가 데이터 스토리지 유닛 및 인쇄 유닛 등의 컴퓨터 플랫폼에 연결될 수 있다.
- [0118] 첨부 도면들에 도시된 구성 시스템 컴포넌트들 및 방법들의 일부가 양호하게 소프트웨어로 구현되기 때문에, 시스템 컴포넌트들 또는 프로세스 기능 블록들 간의 실제 커넥션들은 본 원리들이 프로그래밍될 수 있는 방식에 따라 상이할 수 있음을 또한 알 것이다. 본 명세서의 교시가 주어지면, 당업자는 본 발명의 여타 유사한 구현

들 또는 구성들을 숙고할 수 있을 것이다.

[0119] 일례의 실시예들이 첨부 도면들과 관련하여 본 명세서에 기술되었지만, 본 발명은 이러한 정밀한 실시예들로 제한되지 않으며, 각종 변경들 및 수정들이 본 발명의 범위 또는 원리로부터 벗어나지 않은 채로 당업자에 의해 달성될 수 있음을 알 것이다. 이러한 모든 변경들 및 수정들은 첨부된 청구항들에 기재된 본 발명의 범위 내에 포함되는 것으로 의도된다.

도면

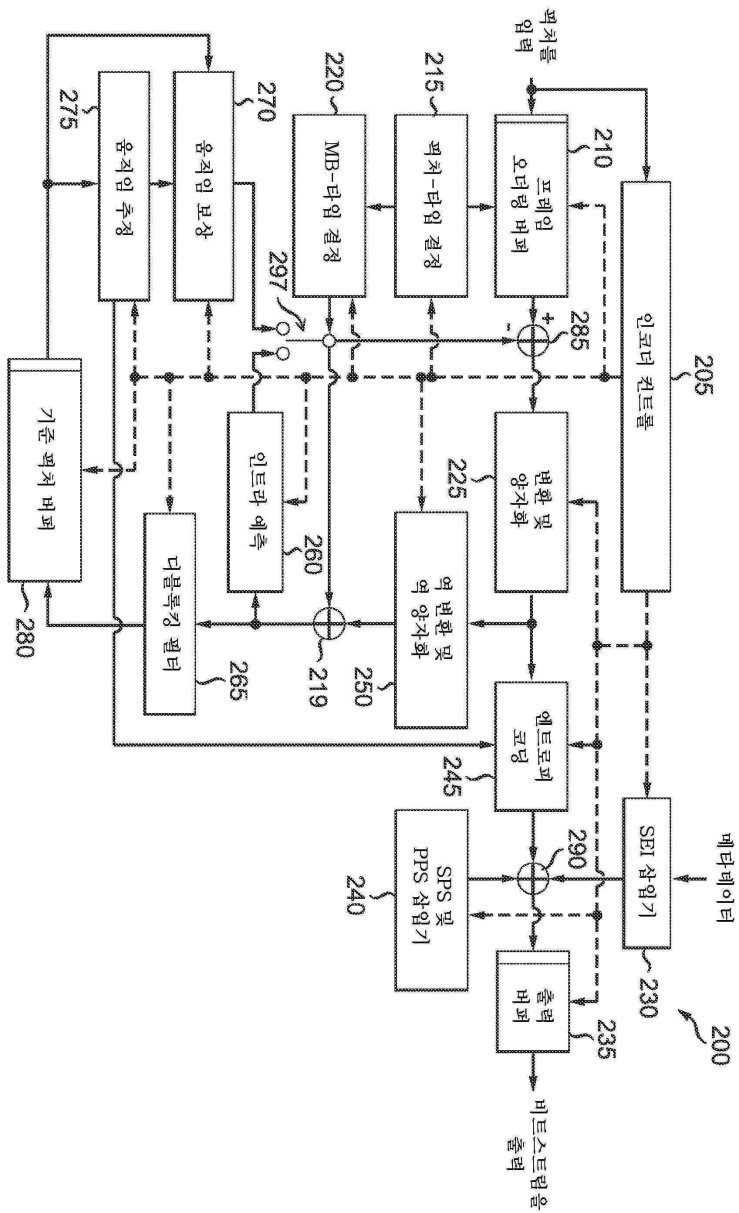
도면1

100
↓

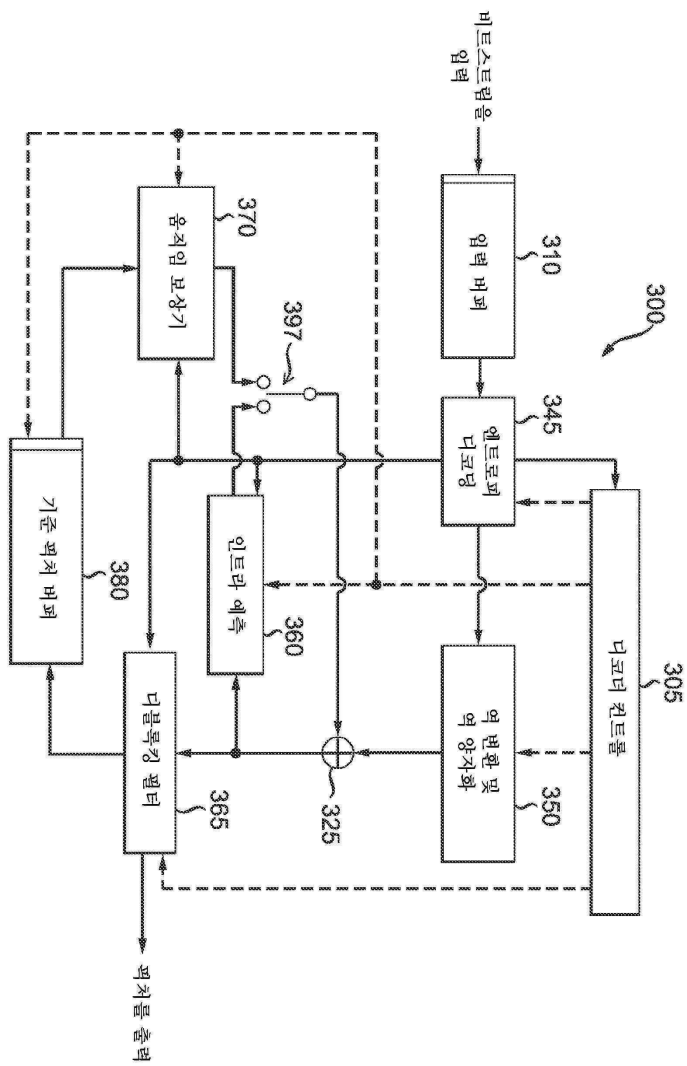
계수들	10	0	-1	2	0	1	0	-1	0...
Sig_flag	1	0	1	1	0	1	0	1	
Last_flag	0		0	0		0		1	
Bin_1	0		1	0		1		1	
Level	10-2=8			2-2=0					
Sign	0		1	0		0		1	

종래 기술

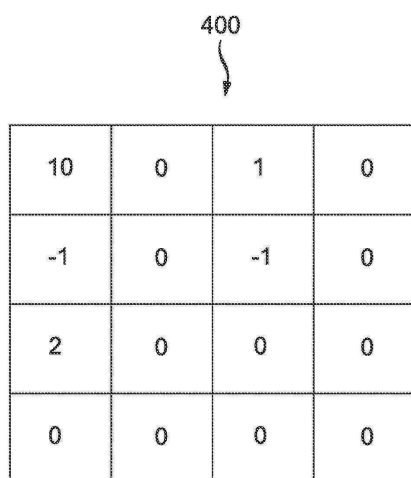
도면2



도면3



도면4

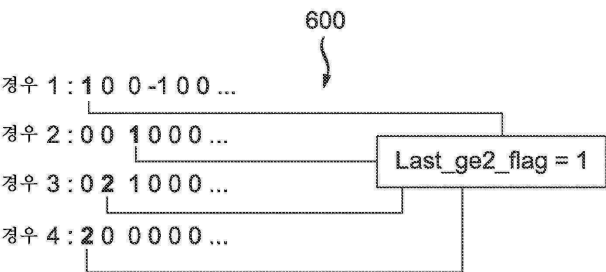


도면5

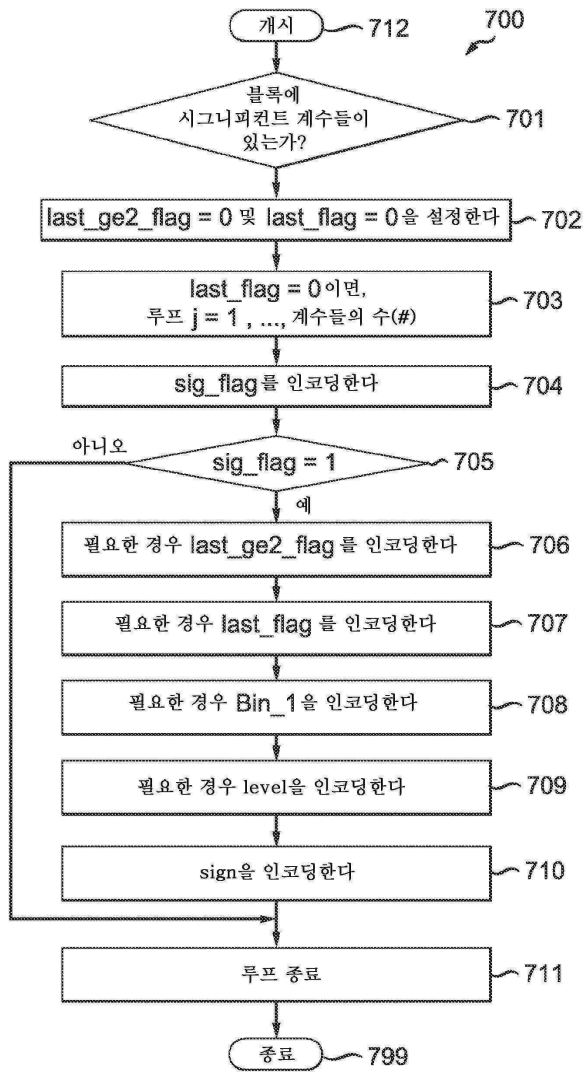
500

계수들	10	0	-1	2	0	1	0	-1	0...
Sig_flag	1	0	1	1	0	1	0	1	
Last_ge2_flag	0		0	1					
Last_flag				0		0		1	
Bin_1	0		1						
Level	10-2=8			2-2=0					
Sign	0		1	0		0		1	

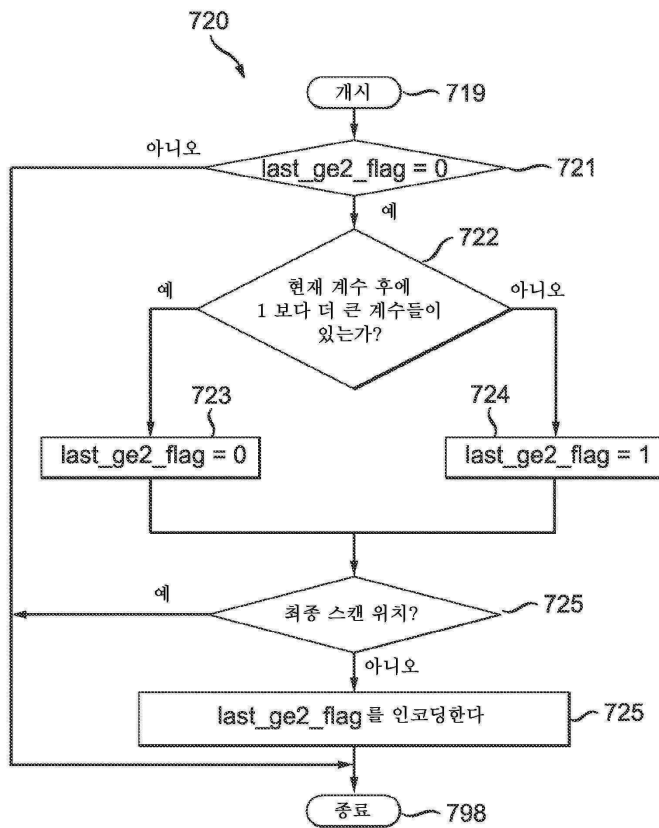
도면6



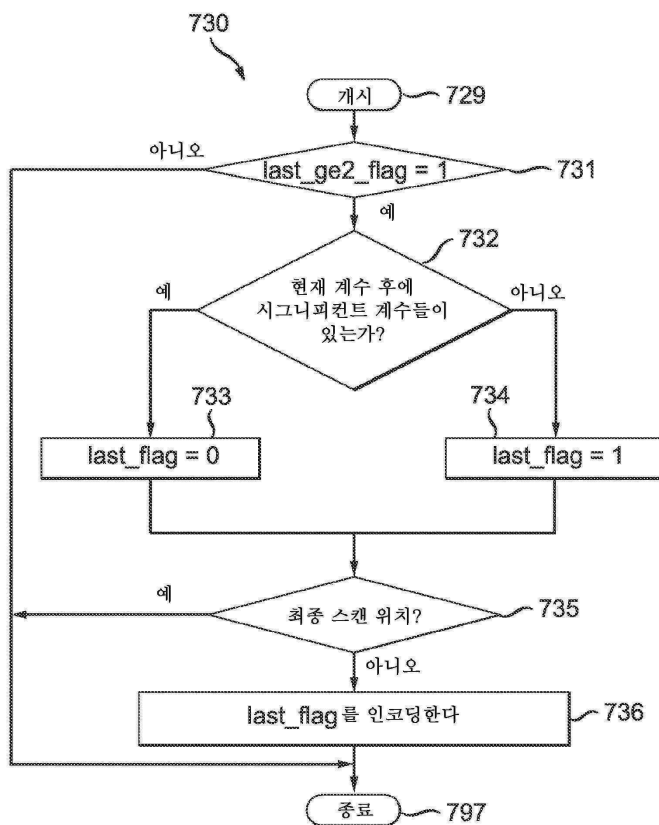
도면7a



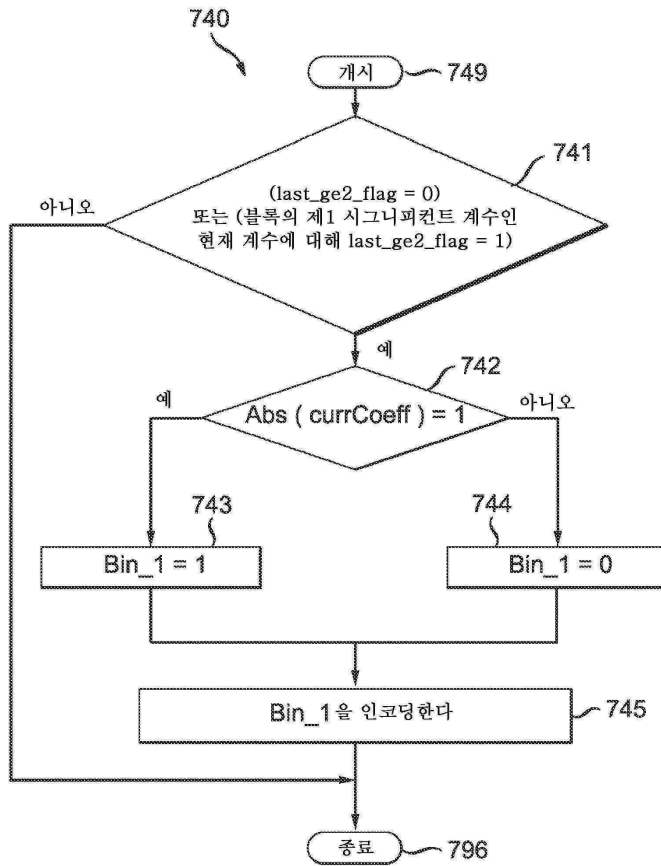
도면7b



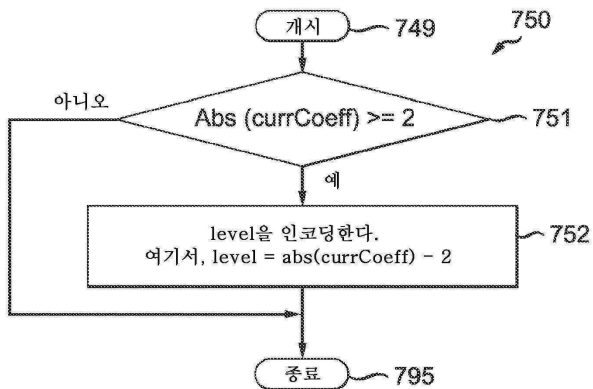
도면7c



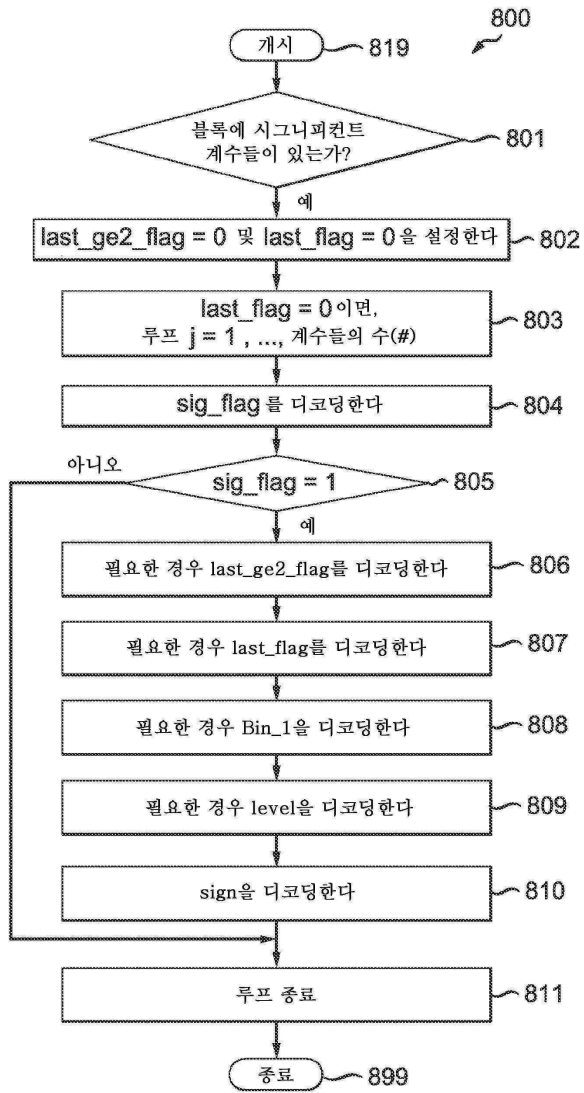
도면7d



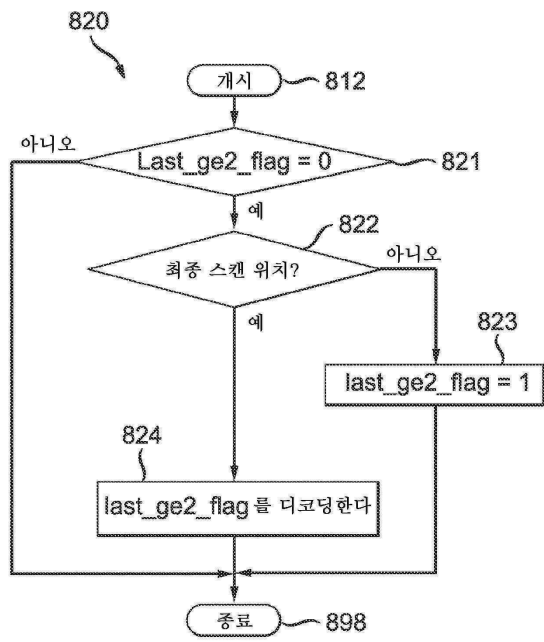
도면7e



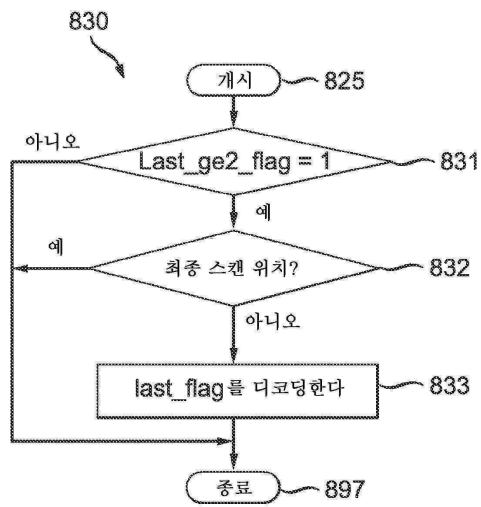
도면 8a



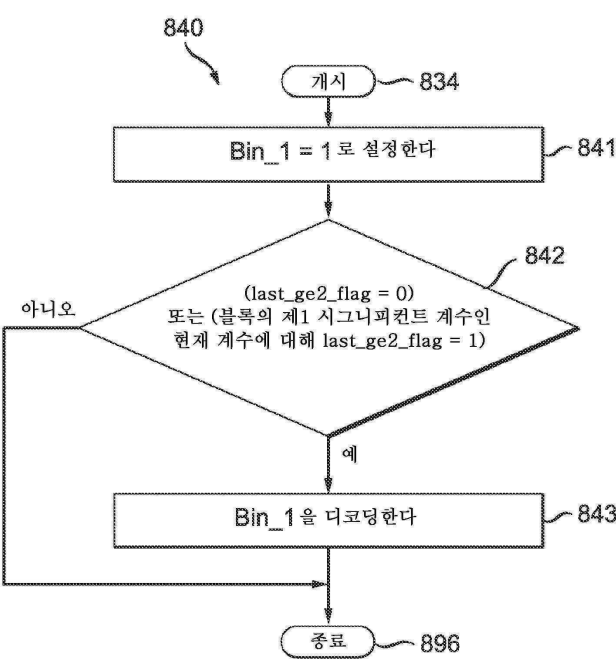
도면8b



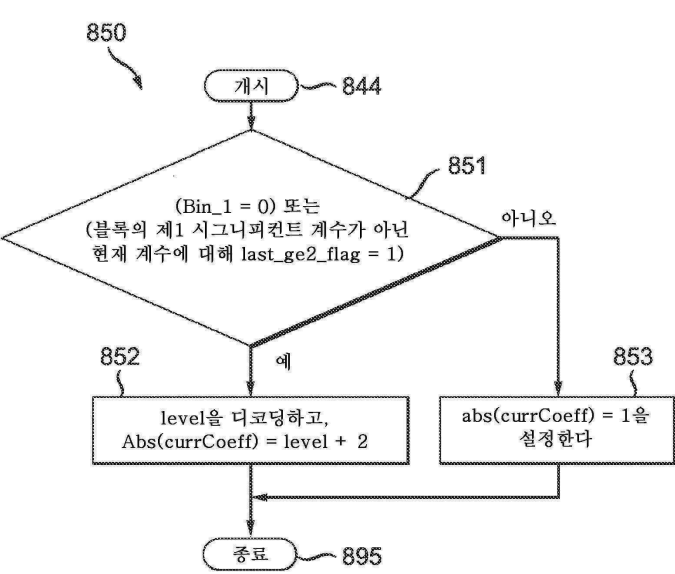
도면8c



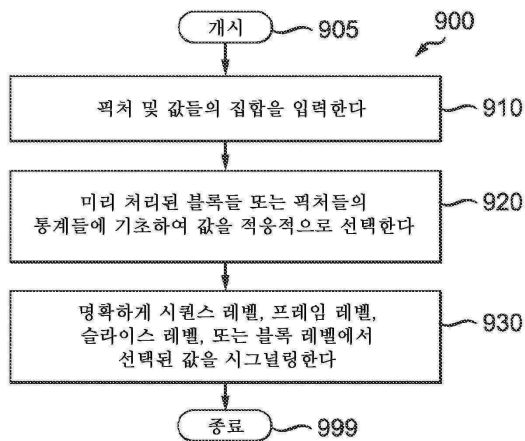
도면8d



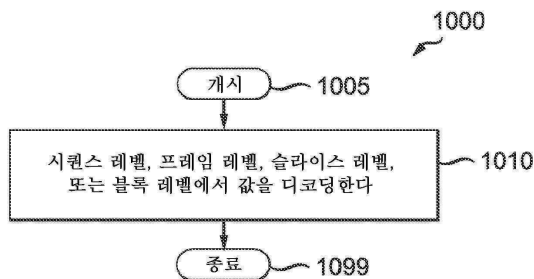
도면8e



도면9



도면10



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 17

【변경전】

비디오 디코딩을 위한 장치로서,

양자화된 변환 계수들을 엔트로피 디코딩하고, 상기 양자화된 변환 계수들을 역양자화하여 변환 계수들을 획득하고, 상기 변환 계수들을 역변환하여 상기 블록을 재구성하는 데 사용하기 위한 상기 블록의 재구성된 잔여물을 획득함으로써 하나의 픽처에서 적어도 하나의 블록을 디코딩하기 위한 비디오 디코더를 포함하고, 양자화된 변환 계수들은 디코딩되고, 플래그는 처리되고 있는 상기 양자화된 변환 계수들 중 현재 변환 계수가 지정된 값인 2보다 크거나 동일한 값을 갖는 상기 블록에 대한 최종 논제로 계수임을 나타내고, 상기 지정된 값인 2는 복수의 값 중에서 선택되고, 상기 선택된 값은 상기 블록에 대해 고정 상태로 유지되고, 상기 플래그는 구문 요소인, 장치.

【변경후】

비디오 디코딩을 위한 장치로서,

양자화된 변환 계수들을 엔트로피 디코딩하고, 상기 양자화된 변환 계수들을 역양자화하여 변환 계수들을 획득하고, 상기 변환 계수들을 역변환하여 블록을 재구성하는 데 사용하기 위한 상기 블록의 재구성된 잔여물을 획득함으로써 하나의 픽처에서 적어도 하나의 블록을 디코딩하기 위한 비디오 디코더를 포함하고, 양자화된 변환 계수들은 디코딩되고, 플래그는 처리되고 있는 상기 양자화된 변환 계수들 중 현재 변환 계수가 지정된 값인 2보다 크거나 동일한 값을 갖는 상기 블록에 대한 최종 논제로 계수임을 나타내고, 상기 지정된 값인 2는 복수의 값 중에서 선택되고, 상기 선택된 값은 상기 블록에 대해 고정 상태로 유지되고, 상기 플래그는 구문 요소인, 장치.

【직권보정 2】

【보정항목】 청구범위

【보정세부항목】 청구항 24

【변경전】

비디오 디코더에서의 방법으로서,

양자화된 변환 계수들을 엔트로피 디코딩하고, 상기 양자화된 변환 계수들을 역양자화하여 변환 계수들을 획득하고, 상기 변환 계수들을 역변환하여 상기 블록을 재구성하는 데 사용하기 위한 상기 블록의 재구성된 잔여물을 획득함으로써 하나의 픽처에서 적어도 하나의 블록을 디코딩하는 단계를 포함하고, 양자화된 변환 계수들은 디코딩되고, 플래그는 처리되고 있는 상기 양자화된 변환 계수들 중 현재 변환 계수가 지정된 값인 2보다 크거나 동일한 값을 갖는 상기 블록에 대한 최종 논제로 계수임을 나타내고, 상기 지정된 값인 2는 복수의 값 중에서 선택되고, 상기 선택된 값은 상기 블록에 대해 고정 상태로 유지되고, 상기 플래그는 구문 요소인, 방법.

【변경후】

비디오 디코더에서의 방법으로서,

양자화된 변환 계수들을 엔트로피 디코딩하고, 상기 양자화된 변환 계수들을 역양자화하여 변환 계수들을 획득하고, 상기 변환 계수들을 역변환하여 블록을 재구성하는 데 사용하기 위한 상기 블록의 재구성된 잔여물을 획득함으로써 하나의 픽처에서 적어도 하나의 블록을 디코딩하는 단계를 포함하고, 양자화된 변환 계수들은 디코딩되고, 플래그는 처리되고 있는 상기 양자화된 변환 계수들 중 현재 변환 계수가 지정된 값인 2보다 크거나 동일한 값을 갖는 상기 블록에 대한 최종 논제로 계수임을 나타내고, 상기 지정된 값인 2는 복수의 값 중에서 선택되고, 상기 선택된 값은 상기 블록에 대해 고정 상태로 유지되고, 상기 플래그는 구문 요소인, 방법.

【직권보정 3】

【보정항목】 청구범위

【보정세부항목】 청구항 31

【변경전】

명령어들을 저장한 컴퓨터 판독가능 저장 매체로서, 상기 명령어들이 프로세서에 의해 실행될 때 제9항 내지 제16항 및 제24항 내지 제30항 중 어느 한 항의 방법이 구현되는, 컴퓨터 판독가능 저장 매체.

【변경후】

명령어들을 저장한 컴퓨터 판독가능 저장 매체로서, 상기 명령어들이 프로세서에 의해 실행될 때 제9항 및 제11항 내지 제16항 및 제24항 및 제26항 내지 제30항 중 어느 한 항의 방법이 구현되는, 컴퓨터 판독가능 저장 매체.