



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2014년01월14일

(11) 등록번호 10-1350591

(24) 등록일자 2014년01월06일

(51) 국제특허분류(Int. Cl.)

H04N 19/46 (2014.01) H04N 13/00 (2006.01)

H04N 19/51 (2014.01)

(21) 출원번호 10-2009-7000135

(22) 출원일자(국제) 2007년07월05일

심사청구일자 2012년07월05일

(85) 번역문제출일자 2009년01월05일

(65) 공개번호 10-2009-0037423

(43) 공개일자 2009년04월15일

(86) 국제출원번호 PCT/US2007/015678

(87) 국제공개번호 WO 2008/005574

국제공개일자 2008년01월10일

(30) 우선권주장

60/807,706 2006년07월18일 미국(US)

60/818,874 2006년07월06일 미국(US)

(56) 선행기술조사문현

WO2006001653 A1

전체 청구항 수 : 총 46 항

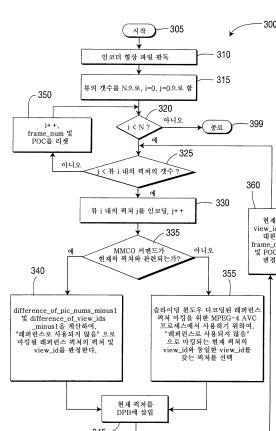
심사관 : 권오성

(54) 발명의 명칭 멀티뷰 비디오 인코딩 및 디코딩을 위해 프레임 넘버 및/또는 퍽쳐 오더 카운트를 분리시키는 방법 및 장치

(57) 요 약

멀티뷰 비디오 콘텐츠의 적어도 2개의 뷰 중 적어도 하나에 대응하는 적어도 하나의 퍽쳐를 디코딩하는 디코더(250) 및 디코딩 방법(410)으로서, 비트스트림에서, 적어도 하나의 퍽쳐에 대한 출력 오더 정보 및 코딩 오더 정보 중 적어도 하나는 적어도 하나의 퍽쳐가 대응하는 적어도 하나의 뷰로부터 분리되는 디코더 및 디코딩 방법이 개시되고 설명된다. 또한, 멀티뷰 비디오 콘텐츠의 2개의 뷰 중 적어도 하나에 대응하는 적어도 하나의 퍽쳐를 인코딩하여 최종 비트스트림을 형성하는 인코더(200) 및 인코딩 방법(360)으로서, 최종 비트스트림에서, 적어도 하나의 퍽쳐에 대한 출력 오더 정보 및 코딩 오더 정보 중 적어도 하나는 적어도 하나의 퍽쳐가 대응하는 적어도 하나의 뷰로부터 분리되는 인코더 및 인코딩 방법이 개시되고 설명된다.

대 표 도 - 도3



(72) 발명자
인, 펭
미국 08550 뉴저지주 웨스트 원저 워윅 로드 65

고밀라, 크리스티나
미국 08540 뉴저지주 프린스턴 채스트넛 코트 25씨

특허청구의 범위

청구항 1

멀티뷰 비디오를 위한 픽쳐 오더링 정보를 분리(decouple)하기 위한 장치로서,

멀티뷰 비디오 콘텐츠의 적어도 2개의 뷰(view) 중 적어도 하나에 대응하는 적어도 하나의 픽쳐를 인코딩하여 최종 비트스트림(resultant bitstream)을 형성하는 인코더(200) - 상기 최종 비트스트림에서, 적어도 하나의 픽쳐에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나가 상기 적어도 하나의 픽쳐가 대응하는 적어도 하나의 뷰로부터 분리됨 -

상기 인코더(200)는, 뷰 식별자를 사용하여 상기 적어도 하나의 픽쳐에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키고, 상기 뷰 식별자는 상기 최종 비트스트림 내의 매크로블록 레벨보다 높은 레벨에 존재하며,

상기 인코더(200)는 디코디드 레퍼런스 픽쳐 마킹 프로세스(decoded reference picture marking process)에서 사용하기 위하여 최종 비트스트림 내에 뷰 식별자를 포함시키는

픽쳐 오더링 정보 분리 장치.

청구항 2

제1항에 있어서,

상기 인코더(200)는 적어도 하나의 기준의 신팩스 구성요소를 사용하여 상기 적어도 하나의 픽쳐에 대한 상기 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 픽쳐 오더링 정보 분리 장치.

청구항 3

제1항에 있어서,

상기 뷰 식별자는 상기 최종 비트스트림 내의 슬라이스 레벨에 존재하는 픽쳐 오더링 정보 분리 장치.

청구항 4

제1항에 있어서,

상기 인코더(200)는 상기 디코디드 레퍼런스 픽쳐 마킹 프로세스에 의해 마킹되어야 하는 특정의 픽쳐가 상기 적어도 2개의 뷰 중 어디에 속하는지를 나타내도록 상기 최종 비트스트림 내에 뷰 식별자를 포함시키는 픽쳐 오더링 정보 분리 장치.

청구항 5

제1항에 있어서,

상기 인코더(200)는, 재정의된 디코디드 레퍼런스 픽쳐 마킹 프로세스에서 뷰 식별자의 사용을 지원하도록, ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항(International Organization for Standardization/International Electrotechnical Commission Moving Picture Experts Group-4 Part 10 Advanced Video Coding standard/International Telecommunication Union, Telecommunication Sector H.264 recommendation)에 대응하는 재정의된 디코디드 레퍼런스 픽쳐 마킹 프로세스에서의 사용을 위해 적어도 하나의 기준의 신팩스 구성요소의 시맨틱(semantics)이 재정의되도록 하여 적어도 하나의 기준의 신팩스 구성요소를 사용하는 픽쳐 오더링 정보 분리 장치.

청구항 6

제5항에 있어서,

상기 재정의된 디코디드 레퍼런스 픽쳐 마킹 프로세스에서, 현재 디코딩된 픽쳐와 동일한 뷰 식별자를 갖는 픽쳐들만이 마킹되는 픽쳐 오더링 정보 분리 장치.

청구항 7

제5항에 있어서,

슬라이딩 윈도우(sliding window) 디코디드 레퍼런스 꾹쳐 마킹 프로세스 및 적응성 메모리 컨트롤(adaptive memory control) 디코디드 레퍼런스 꾹쳐 마킹 프로세스 중 적어도 하나가 적용되는 꾹쳐 오더링 정보 분리 장치.

청구항 8

제5항에 있어서,

상기 재정의된 디코디드 레퍼런스 꾹쳐 마킹 프로세스에서, 상기 적어도 하나의 꾹쳐와는 상이한 뷰 식별자를 갖는 꾹쳐들은 이전에 사용되지 않은 신택스 구성요소를 사용하여 마킹되는 꾹쳐 오더링 정보 분리 장치.

청구항 9

제1항에 있어서,

상기 인코더(200)는, 디폴트 레퍼런스 꾹쳐 리스트 구성을 위해 상기 최종 비트스트림 내에 뷰 식별자를 포함시키는 꾹쳐 오더링 정보 분리 장치.

청구항 10

제9항에 있어서,

뷰 식별자로부터 추가의 지원을 받아 상기 레퍼런스 꾹쳐 리스트 구성을 위해 기존의 신택스 구성요소, 기존의 시맨틱 및 기존의 디코딩 프로세스 중 적어도 하나에 따라서, 상기 레퍼런스 꾹쳐 리스트 구성에 대응하는 디폴트 레퍼런스 꾹쳐 리스트 작성 프로세스를 위한 레퍼런스 리스트에 인터-뷰(inter-view) 레퍼런스 꾹쳐들이 추가되는 것이 금지되는 꾹쳐 오더링 정보 분리 장치.

청구항 11

제9항에 있어서,

뷰 식별자로부터 추가의 지원을 받아 상기 디폴트 레퍼런스 꾹쳐 리스트 구성을 위해 적어도 하나의 기존의 신택스 구성요소에 따라서 상기 레퍼런스 꾹쳐 리스트 구성에 대응하는 디폴트 레퍼런스 꾹쳐 리스트 작성 프로세스를 위한 레퍼런스 리스트에 인터-뷰 레퍼런스 꾹쳐들만이 추가되는 꾹쳐 오더링 정보 분리 장치.

청구항 12

제11항에 있어서,

상기 인터-뷰 레퍼런스 꾹쳐들은 시간적인 레퍼런스 이후에 추가되는 꾹쳐 오더링 정보 분리 장치.

청구항 13

제1항에 있어서,

상기 인코더(200)는, 재정의되는 레퍼런스 꾹쳐 리스트 리오더링(reordering) 프로세스에서 뷰 식별자의 사용을 지원하도록, 재정의된 레퍼런스 꾹쳐 리스트 리오더링 프로세스에서 사용하기 위하여 재정의되는 적어도 하나의 기존의 신택스 구성요소를 사용하는 꾹쳐 오더링 정보 분리 장치.

청구항 14

제13항에 있어서,

상기 재정의된 레퍼런스 꾹쳐 리스트 리오더링 프로세스에서, 현재 디코딩되는 꾹쳐와 동일한 뷰 식별자를 갖는 꾹쳐들만이 리오더링되는 꾹쳐 오더링 정보 분리 장치.

청구항 15

제14항에 있어서,

상기 뷰 식별자는, 적어도 2개의 뷰 중 어느 것이 대응하는 레퍼런스 꾹쳐 리스트 내에서 현재 인덱스에 이동되

어야 하는 특정 픽쳐에 대응하는지를 나타내는 픽쳐 오더링 정보 분리 장치.

청구항 16

제14항에 있어서,

상기 뷰 식별자는, 오더링되어야 하는 레퍼런스 픽쳐의 뷰 식별자가 상기 적어도 하나의 픽쳐의 뷰 식별자와 상이한 경우에만 필요한 픽쳐 오더링 정보 분리 장치.

청구항 17

제1항에 있어서,

시간적인 DIRECT 모드(temporal DIRECT mode)에서 뷰 식별자의 사용을 지원하도록, 상기 인코더(200)는, 시간적인 DIRECT 모드에 대하여 재정의된 기준의 선택스 구성요소를 사용하는 픽쳐 오더링 정보 분리 장치.

청구항 18

제17항에 있어서,

상기 시간적인 DIRECT 모드는, 픽쳐 오더 카운트 값(Picture Order Count value) 및 뷰 식별자 중 적어도 하나에 기초하여 유도되는 픽쳐 오더링 정보 분리 장치.

청구항 19

제1항에 있어서,

상기 인코더(200)는, 시간적인 DIRECT 모드에 대하여 기준의 선택스 구성요소, 기준의 시맨틱 및 기준의 디코딩 프로세스를 사용하는 픽쳐 오더링 정보 분리 장치.

청구항 20

제1항에 있어서,

내재 가중 예측(implicit weighted prediction)에서 상기 뷰 식별자의 사용을 지원하도록, 상기 인코더(200)는 내재 가중 예측에 대하여 재정의된 기준의 선택스 구성요소를 사용하는 픽쳐 오더링 정보 분리 장치.

청구항 21

제20항에 있어서,

상기 내재 가중 예측은, 픽쳐 오더 카운트 값 및 뷰 식별자 중 적어도 하나에 기초하여 유도되는 픽쳐 오더링 정보 분리 장치.

청구항 22

제1항에 있어서,

상기 인코더(200)는, 내재 가중 예측을 위해 기준의 선택스 구성요소, 기준의 시맨틱 및 기준의 디코딩 프로세스를 사용하는 픽쳐 오더링 정보 분리 장치.

청구항 23

제1항에 있어서,

상기 인코더(200)는, 상기 적어도 2개의 뷰 중 상이한 것들의 병렬 인코딩에 있어서 인터-뷰 의존성을 지정하기 위하여 특정의 픽쳐에 대응하는 적어도 2개의 뷰 중 특정의 하나를 사용하는 픽쳐 오더링 정보 분리 장치.

청구항 24

멀티뷰 비디오를 위한 픽쳐 오더링 정보를 분리하기 위한 방법으로서,

멀티뷰 비디오 콘텐츠의 적어도 2개의 뷰 중 적어도 하나에 대응하는 적어도 하나의 픽쳐를 인코딩하여 최종 비트스트림을 형성하는 인코딩 단계 - 상기 최종 비트스트림에서, 상기 적어도 하나의 픽쳐에 대한 코딩 오더 정

보 및 출력 오더 정보 중 적어도 하나는 상기 적어도 하나의 픽쳐가 대응하는 적어도 하나의 뷰로부터 분리됨 (360) -;

상기 인코딩 단계는, 뷰 식별자를 사용하여 상기 적어도 하나의 픽쳐에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키고(340, 355), 상기 뷰 식별자는 상기 최종 비트스트림 내의 매크로블록 레벨보다 높은 레벨에 존재하며,

상기 인코딩 단계는, 디코디드 레퍼런스 픽쳐 마킹 프로세스에서 사용하기 위해 상기 최종 비트스트림 내에 뷰 식별자를 포함하는(340, 355)

픽쳐 오더링 정보 분리 방법.

청구항 25

제24항에 있어서,

상기 인코딩 단계는, 적어도 하나의 기존의 신택스 구성요소를 사용하여 상기 적어도 하나의 픽쳐에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는(360) 픽쳐 오더링 정보 분리 방법.

청구항 26

제24항에 있어서,

상기 뷰 식별자는 상기 최종 비트스트림 내의 슬라이스 레벨에 존재하는 픽쳐 오더링 정보 분리 방법.

청구항 27

제24항에 있어서,

상기 인코딩 단계는, 상기 적어도 2개의 뷰 중 어느 것에 디코디드 레퍼런스 픽쳐 마킹 프로세스에 의해 마킹되어야 하는 특정한 픽쳐가 속하는지를 나타내기 위하여 상기 최종 비트스트림 내에 뷰 식별자를 포함하는(340, 355) 픽쳐 오더링 정보 분리 방법.

청구항 28

제24항에 있어서,

상기 인코딩 단계는, 재정의된 디코디드 레퍼런스 픽쳐 마킹 프로세스(1620)에서 뷰 식별자의 사용을 지원하도록, 재정의된 디코디드 레퍼런스 픽쳐 마킹 프로세스에서 사용하기 위하여 적어도 하나의 기존의 신택스 구성요소의 시맨틱이 재정의되도록 하여 적어도 하나의 기존의 신택스 구성요소를 사용하는 픽쳐 오더링 정보 분리 방법.

청구항 29

제28항에 있어서,

상기 재정의된 디코디드 레퍼런스 픽쳐 마킹 프로세스에서, 현재의 디코딩된 픽쳐와 동일한 뷰 식별자를 갖는 픽쳐들만이 마킹되는(1540, 1555) 픽쳐 오더링 정보 분리 방법.

청구항 30

제28항에 있어서,

슬라이딩 윈도우 디코디드 레퍼런스 픽쳐 마킹 프로세스 및 적응성 메모리 컨트롤 디코디드 레퍼런스 픽쳐 마킹 프로세스 중 적어도 하나가 적용되는 픽쳐 오더링 정보 분리 방법.

청구항 31

제28항에 있어서,

상기 재정의된 디코디드 레퍼런스 픽쳐 마킹 프로세스에서, 상기 적어도 하나의 픽쳐와 상이한 뷰 식별자를 갖는 픽쳐들은 이전에 사용되지 않은 신택스 구성요소를 사용하여 마킹되는 픽쳐 오더링 정보 분리 방법.

청구항 32

제24항에 있어서,

상기 인코딩 단계는, 디폴트 레퍼런스 꾹쳐 리스트 구성을 위해 상기 최종 비트스트림 내에 상기 뷰 식별자를 포함하는(530) 꾹쳐 오더링 정보 분리 방법.

청구항 33

제32항에 있어서,

상기 레퍼런스 꾹쳐 리스트 구성을 위한 적어도 하나의 기준의 선택스 구성요소에 따라서, 상기 레퍼런스 꾹쳐 리스트 구성에 대응하는 디폴트 레퍼런스 꾹쳐 리스트 작성 프로세스(530)를 위한 레퍼런스 리스트에 인터-뷰 레퍼런스 꾹쳐들이 추가되는 것이 금지되는 꾹쳐 오더링 정보 분리 방법.

청구항 34

제32항에 있어서,

상기 디폴트 레퍼런스 꾹쳐 리스트 구성을 위한 적어도 하나의 기준의 선택스 구성요소에 따라서, 상기 레퍼런스 꾹쳐 리스트 구성에 대응하는 디폴트 레퍼런스 꾹쳐 리스트 작성 프로세스(630)를 위한 레퍼런스 리스트에 인터-뷰 레퍼런스 꾹쳐들만이 추가되는 꾹쳐 오더링 정보 분리 방법.

청구항 35

제34항에 있어서,

상기 인터-뷰 레퍼런스 꾹쳐들은 시간적인 레퍼런스 후에 추가되는 꾹쳐 오더링 정보 분리 방법.

청구항 36

제24항에 있어서,

상기 인코딩 단계는, 재정의된 레퍼런스 꾹쳐 리스트 리오더링 프로세스에서 뷰 식별자의 사용을 지원하도록, 재정의된 레퍼런스 꾹쳐 리스트 리오더링 프로세스(2132)에서 사용하기 위하여 재정의되는 적어도 하나의 기준의 선택스 구성요소를 사용하는 꾹쳐 오더링 정보 분리 방법.

청구항 37

제36항에 있어서,

상기 재정의된 레퍼런스 꾹쳐 리스트 리오더링 프로세스에서, 현재의 디코딩된 꾹쳐와 동일한 뷰 식별자를 갖는 꾹쳐들만이 리오더링되는(2134) 꾹쳐 오더링 정보 분리 방법.

청구항 38

제37항에 있어서,

상기 뷰 식별자는, 대응하는 레퍼런스 꾹쳐 리스트 내에서 현재의 인덱스에 이동되어야 하는 특정의 꾹쳐가 적어도 2개의 뷰 중 어느 것에 대응하는지를 나타내는(2234) 꾹쳐 오더링 정보 분리 방법.

청구항 39

제37항에 있어서,

상기 뷰 식별자는, 오더링되어야 하는 레퍼런스 꾹쳐의 뷰 식별자가 상기 적어도 하나의 꾹쳐의 뷰 식별자와 상이한 경우에만 필요한 꾹쳐 오더링 정보 분리 방법.

청구항 40

제24항에 있어서,

상기 인코딩 단계는, 시간적인 DIRECT 모드(1130, 1230)에서 상기 뷰 식별자의 사용을 지원하도록, 상기 시간적

인 DIRECT 모드에 대하여 재정의되는 기존의 선택스 구성요소를 사용하는 꽉쳐 오더링 정보 분리 방법.

청구항 41

제40항에 있어서,

상기 시간적인 DIRECT 모드는 꽉쳐 오더 카운트 값 및 뷰 식별자 중 적어도 하나에 기초하여 유도되는(1230) 꽉쳐 오더링 정보 분리 방법.

청구항 42

제24항에 있어서,

상기 인코딩 단계는, 시간적인 DIRECT 모드에 대해 기존의 선택스 구성요소, 기존의 시맨틱 및 기존의 디코딩 프로세스를 사용하는 꽉쳐 오더링 정보 분리 방법.

청구항 43

제24항에 있어서,

상기 인코딩 단계는, 내재 가중 예측에서 상기 뷰 식별자의 사용을 지원하도록 내재 가중 예측(1132, 1232)을 위해 재정의되는 기존의 선택스 구성요소를 사용하는 꽉쳐 오더링 정보 분리 방법.

청구항 44

제43항에 있어서,

상기 내재 가중 예측은 꽉쳐 오더 카운트 값 및 뷰 식별자 중 적어도 하나에 기초하여 유도되는(1232) 꽉쳐 오더링 정보 분리 방법.

청구항 45

제24항에 있어서,

상기 인코딩 단계는, 내재 가중 예측을 위해 기존의 선택스 구성요소, 기존의 시맨틱 및 기존의 디코딩 프로세스를 사용하는 꽉쳐 오더링 정보 분리 방법.

청구항 46

제24항에 있어서,

상기 인코딩 단계는, 적어도 2개의 뷰 중 상이한 것들의 병렬 인코딩에 있어서의 인터-뷰 의존성(350)을 지정하기 위하여 특정의 꽉쳐에 대응하는 적어도 2개의 뷰 중 특정의 하나를 사용하는 꽉쳐 오더링 정보 분리 방법.

청구항 47

삭제

청구항 48

삭제

청구항 49

삭제

청구항 50

삭제

청구항 51

삭제

청구항 52

삭제

청구항 53

삭제

청구항 54

삭제

청구항 55

삭제

청구항 56

삭제

청구항 57

삭제

청구항 58

삭제

청구항 59

삭제

청구항 60

삭제

청구항 61

삭제

청구항 62

삭제

청구항 63

삭제

청구항 64

삭제

청구항 65

삭제

청구항 66

삭제

청구항 67

삭제

청구항 68

삭제

청구항 69

삭제

청구항 70

삭제

명세서

[0001] 관련 출원에 대한 상호참조

본 출원은 미국 가출원 제60/818,874호(2006년 7월 6일자) 및 미국 가출원 제60/807,706호(2006년 7월 18일자)의 우선권의 이익을 주장하며, 본 명세서에 전체가 참조로 포함되어 있다. 또한, 본 출원은 미국 출원 "Method and Apparatus for Decoupling Frame Number and/or Picture Order Count (POC) for Multi-view Video Encoding and Decoding"(Attorney Docket No. PU060220)에 관련되며, 상기 출원은 공동으로 양도되고, 본 명세서에 참조로서 포함되어 현재 함께 출원중에 있다.

기술 분야

[0003] 본 발명의 원리는 일반적으로 비디오 인코딩 및 디코딩에 관한 것으로서, 특히, 멀티뷰 비디오 인코딩 및 디코딩을 위해 프레임 넘버 및/또는 픽쳐 오더 카운트(POC: Picture Order Count)를 분리시키는 방법 및 장치에 관한 것이다.

배경 기술

[0004] ISO/IEC MPEG-4(International Organization for Standardization/ International Electrotechnical Commission Moving Picture Experts Group-4) Part 10의 AVC(Advanced Video Coding) 표준/ITU-T H.264 권장 사항(이하, MPEG-4 AVC 표준이라 함)에 있어서, 신택스(syntax) 요소 frame_num은 픽쳐에 대한 식별자로서 사용되며, MPEG-4 AVC 표준에 정의된 바와 같이 수개의 제한 사항을 갖는다. frame_num의 일차적인 목적은 픽쳐가 디코딩될 때마다 증분되는 카운터로서 역할하여, 데이터 손실이 있는 경우, 디코더가 일부 픽쳐들이 손실되었음을 검출하여, 문제를 해소할 수 있도록 하는 것이다. frame_num은 액세스 유닛의 디코딩 순서로 증가하며, 표시 순서를 지시할 필요는 없다. MMCO(Memory Management Control Operations)는 frame_num의 값을 사용하여, 장기 및 단기 기준으로서 픽쳐를 마킹하거나, 레퍼런스 픽쳐를 레퍼런스 픽쳐로서 사용하지 않는 것으로 마킹한다. frame_num은 또한 P 및 SP 슬라이스에 대한 디폴트 레퍼런스 리스트 오더링을 위해 사용된다.

[0005] MPEG-4 AVC 표준에서 픽쳐 오더 카운트(POC)는 특정 픽쳐의 타이밍 또는 출력 순서(ordering)의 표시이다. 픽쳐 오더 카운트는, 디코딩 순서에 있어서의 이전의 IDR(Instantaneous Decoding Refresh) 픽쳐에 대하여, 또는 모든 레퍼런스 픽쳐들을 "레퍼런스로 사용되지 않음"으로 마킹하는 메모리 관리 제어 동작을 포함하고 있는 이전 픽쳐에 대하여, 출력 순서에 있어서 픽쳐 위치를 증가시킴에 따라서 감소하지 않는 값을 갖는 변수이다. 픽쳐 오더 카운트는 슬라이스 헤더 신택스 구성요소로부터 유도된다. 픽쳐 오더 카운트는 시간적인 DIRECT 모드에서 모션 벡터의 유도, 내재 가중 예측(implicit weighted prediction) 및 B 슬라이스에 대한 디폴트 초기 레퍼런스 픽쳐 리스트 오더링에 있어서 사용된다.

[0006] 특히, 시간적인 상관(correlation)을 사용한 DIRECT 모드 모션 파라미터들은, 통상적으로, 후속하는 레퍼런스 픽쳐 또는 더 정확하게는 제1 목록(List 1)의 같은 장소에 배치된(co-located) 위치 내의 모션 정보를 검토함으로써, 현재의 매크로블록/블록에 대하여 유도된다. 도 1을 참조하면, B 슬라이스 코딩에서의 시간적인 DIRECT 예측은 일반적으로 참조 번호 100으로 표시되어 있다. 객체가 일정한 속도로 움직인다는 가정을 따르면, 이 파라미터들은 관련 레퍼런스 픽쳐의 시간적인 거리(도 1 참조)에 따라서 스케일링된다. DIRECT 코딩된 블록에 대한 모션 벡터(\overrightarrow{MV}_{L0} 및 \overrightarrow{MV}_{L1})와 제1 리스트(List 1) 레퍼런스 픽쳐 내에서 그것의 같은 장소에 배치된 위치의 모션 벡터(\overrightarrow{MV})는 이하와 같이 계산된다.

$$X = (16384 + \text{abs}(TD_B / 2)) / TD_B \quad (1)$$

$$\text{ScaleFactor} = \text{clip}(-1024, 1023, (TD_B \times X + 32) >> 6) \quad (2)$$

$$\overrightarrow{MV}_{L0} = (\text{ScaleFactor} \times \overrightarrow{MV} + 128) >> 8 \quad (3)$$

$$\overrightarrow{MV}_{L1} = \overrightarrow{MV}_{L0} - \overrightarrow{MV} \quad (4)$$

[0007]

선행의 방정식에서, TD_B 및 TD_D 는 시간적인 거리, 또는 더 자세하게, 각각 현재의 픽쳐 및 List 1 픽쳐에 비교한 List 1 픽쳐 내의 같은 장소에 배치된 블록의 제0리스트(List 0) 모션 벡터에 의해 사용되는 레퍼런스 픽쳐의 픽쳐 오더 카운트(POC) 거리이다. List 1 레퍼런스 픽쳐 및 List 1 내의 같은 장소에 배치된 블록의 모션 벡터에 의해 참조되는 List 0 내의 레퍼런스 픽쳐가 DIRECT 모드의 2개의 기준으로서 사용된다. 레퍼런스 인덱스(refIdxL0)가 장기(long-term) 레퍼런스 픽쳐를 지칭하거나, DiffPicOrderCnt(pic1, pic0)가 0과 같다면, DIRECT 모드 파티션에 대한 모션 벡터(\overrightarrow{MV}_{L0} 및 \overrightarrow{MV}_{L1})는 이하의 식에 의해 유도된다.

$$\overrightarrow{MV}_{L0} = \text{병치}(\text{collocated}) \text{된 매크로블록의 모션 벡터}$$

$$\overrightarrow{MV}_{L1} = 0$$

내재(implicit) 가중 예측 툴(tool) 또한 가중치를 결정하기 위하여 픽쳐 오더 카운트를 사용한다. 가중 예측(WP) 내재 모드에 있어서, 가중 인자는 슬라이스 헤더에서 명시적으로 전송되지 않지만, 그 대신, 현재의 픽쳐와 레퍼런스 픽쳐 간의 상대 거리에 기초하여 유도된다. 내재 모드는 DIRECT 모드를 사용하는 것을 포함하여 양방향성 코딩 매크로블록 및 B 슬라이스 내의 매크로블록 파티션에 대해서만 사용된다. 내재 모드에 있어서, 오프셋 값(o_0 및 o_1)이 0과 같다는 것을 제외하고는, 수식 (1)의 공식이 사용되며, 수식 (6) 내지 수식 (10)의 아래 공식을 이용하여 가중 인자(W_0 및 W_1)들이 유도된다.

$$\text{predPartC}[x, y] = \text{Clip1C}((\text{predPartL0C}[x, y] * w_0 + \text{predPartL1C}[x, y] * w_1 + 2\log WD) >> (\log WD + 1)) + ((o_0 + o_1 + 1) >> 1) \quad (5)$$

$$X = (16384 + (TD_D >> 1)) / TD_D \quad (6)$$

$$Z = \text{clip3}(-1024, 1023, (TD_B \cdot X + 32) >> 6) \quad (7)$$

$$W_1 = Z >> 2 \quad (8)$$

$$W_0 = 64 - W_1 \quad (9)$$

[0012]

이는 이하의 나눗셈을 없앤 16 비트의 안전한 연산의 구현이다.

[0014]

$$W_1 = (64 \cdot TD_D) / TD_B \quad (10)$$

[0015]

$$\text{DiffPicOrderCnt(picA, picB)} = \text{PicOrderCnt(picA)} - \text{PicOrderCnt(picB)} \quad (11)$$

[0016]

여기서, TD_B 는 List 1 레퍼런스 픽쳐와 List 0 레퍼런스 픽쳐 간의 시간적인 차이로서, [-128, 127]의 범위에 고정되며, TD_B 는 현재 픽쳐와 List 0 레퍼런스 픽쳐의 차이로서, [-128, 127]의 범위에 고정된다. 멀티뷰 비디오 코딩에 있어서, TD_B 가 0이 될 수 있는 경우가 있을 수 있다 (이는 수식 (11)의 $\text{DiffPicOrderCnt(picA, picB)}$ 가 0인 경우 일어난다). 이 경우, 가중치 W_0 및 W_1 은 32로 설정된다.

[0017]

현재 MPEG-4 AVC에 준거하는 멀티뷰 비디오 코딩(MVC)의 구현예에 있어서, 레퍼런스 소프트웨어는 모든 비디오 시퀀스를 하나의 스트림으로 인터리빙함으로써 멀티뷰 예측(prediction)을 달성한다. 이와 같이, frame_run 및 Picture Order Count는 뷰(views) 사이에서 서로 결합된다. 이는 몇몇 단점을 갖는다. 하나의 단점으로서, 파셜 디코딩(partial decoding)에 있어서 frame_num의 값에 캡이 있을 수 있다. 이는 레퍼런스 픽쳐 리스트의 관리를 복잡하게 만들거나, frame_num 캡에 기초한 여러 손실 예측을 불가능하게 할 수 있다. 또 다른 단점으로

서, 꽂쳐 오더 카운트가 실제 물리적 의미를 가지고 있지 않으며, 이는 시간적인 DIRECT 모드 또는 내재 가중 예측 등의 꽂쳐 오더 카운트 정보에 의존하는 어떠한 코딩 틀이라도 파괴할 수 있다는 것이다. 또 다른 단점으로서, 결합(coupling)이 멀티뷰 시퀀스의 병렬 코딩을 더 어렵게 한다는 점이다.

발명의 상세한 설명

- [0018] 종래 기술의 이들 및 기타의 단점 및 불이익은, 멀티뷰 비디오 인코딩 및 디코딩을 위해 프레임 넘버 및 꽂쳐 오더 카운트(POC)를 분리시키는 방법 및 장치에 관한 본 발명에 의해 해소된다.
- [0019] 본 발명의 일 양태에 따르면, 장치가 제공된다. 본 장치는, 멀티뷰 비디오 콘텐츠의 2개의 뷰 중 적어도 하나에 대응하는 적어도 하나의 꽂쳐를 인코딩하여 최종 비트스트림을 형성하는 인코더를 포함한다. 최종 비트스트림에서, 적어도 하나의 꽂쳐에 대한 출력 오더 정보 및 코딩 오더 정보 중 적어도 하나는 적어도 하나의 꽂쳐가 대응하는 적어도 하나의 뷰로부터 분리된다.
- [0020] 본 발명의 또 다른 양태에 따르면, 방법이 제공된다. 본 방법은, 멀티뷰 비디오 콘텐츠의 적어도 2개의 뷰 중 적어도 하나에 대응하는 적어도 하나의 꽂쳐를 인코딩하여 최종 비트스트림을 형성하는 단계를 포함한다. 최종 비트스트림에서는, 적어도 하나의 꽂쳐에 대한 출력 오더 정보 및 코딩 오더 정보 중 적어도 하나가 적어도 하나의 꽂쳐가 대응하는 적어도 하나의 뷰로부터 분리된다.
- [0021] 본 발명의 또 다른 양태에 따르면, 장치가 제공된다. 본 장치는, 멀티뷰 비디오 콘텐츠에 대응하는 적어도 2개의 뷰 중 적어도 하나를 인코딩하는 인코더를 포함한다. 인코더는 ISO/IEC MPEG-4 (International Organization for Standardization/International Electrotechnical Commission Moving Picture Experts Group-4) Part 10 AVC (Advanced Video Coding) 표준/ITU-T H.264 (International Telecommunication Union, Telecommunication Sector H.264) recommendation에 대응하는 레퍼런스 꽂쳐 리스트 리오더링 및 디폴트 레퍼런스 꽂쳐 리스트 구성 프로세스에서 재정의된 변수들을 이용하여 적어도 2개의 뷰 중 적어도 하나를 인코딩한다.
- [0022] 본 발명의 또 다른 양태에 따르면, 장치가 제공된다. 본 장치는, 멀티뷰 비디오 콘텐츠에 대응하는 적어도 2개의 뷰 중 적어도 하나를 인코딩하는 인코더를 포함한다. 인코더는 ISO/IEC MPEG-4 Part 10 AVC standard/ITU-T H.264 권고사항의 디코디드 레퍼런스 꽂쳐 마킹 프로세스에서 재정의된 변수들을 이용하여 적어도 2개의 뷰 중 적어도 하나를 인코딩한다.
- [0023] 본 발명의 또 다른 양태에 따르면, 방법이 제공된다. 본 방법은, 멀티뷰 비디오 콘텐츠에 대응하는 적어도 2개의 뷰 중 적어도 하나를 인코딩하는 단계를 포함한다. 인코딩 단계는 ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항에 대응하는 레퍼런스 꽂쳐 리스트 리오더링 및 디폴트 레퍼런스 꽂쳐 리스트 구성 프로세스에서 재정의된 변수들을 이용하여 적어도 2개의 뷰 중 적어도 하나를 인코딩한다.
- [0024] 본 발명의 또 다른 양태에 따르면, 방법이 제공된다. 본 방법은, 멀티뷰 비디오 콘텐츠에 대응하는 적어도 2개의 뷰 중 적어도 하나를 인코딩하는 단계를 포함한다. 인코딩 단계는, ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항의 디코디드 레퍼런스 꽂쳐 마킹 프로세스에서 재정의된 변수들을 이용하여 적어도 2개의 뷰 중 적어도 하나를 인코딩한다.
- [0025] 본 발명의 이들 및 기타의 양태, 특징, 및 장점들은, 첨부 도면들과 연계하여 참조되어야 하는, 예시적인 실시 예들에 대한 이하의 상세한 설명으로부터 더 명확하게 될 것이다.

실시 예

- [0052] 본 발명은 멀티뷰 비디오 인코딩 및 디코딩을 위한 프레임 넘버 및 꽂쳐 오더 카운트(POC)를 분리(decoupling)시키는 방법 및 장치에 관한 것이다.
- [0053] 이하의 설명은 본 발명을 예시한다. 따라서, 본 명세서에서 명시적으로 기재 또는 도시되지 않아도, 당업자라면 본 발명을 구체화하며, 그 개념과 범주 내에 포함되는 다양한 배치를 구성할 수 있을 것이다.
- [0054] 본 명세서에 인용되는 모든 예들과 가정 어구는 발명자(들)에 의해 기술을 촉진하는데 기여되는 본 발명의 원리와 개념을 이해하는데 있어서 독자를 돋기위한 교시적 목적을 위한 것이며, 이러한 구체적으로 인용된 예들과 조건들로 제한되지 않는 것으로 상정되어야 한다.
- [0055] 또한, 본 명세서에서 구체적인 예뿐만 아니라 본 발명의 원리, 양태, 및 실시예들을 인용하는 모든 설명들은 그

구조적인 균등물 및 기능적인 균등물 모두를 포함하고자 한 것이다. 또한, 이러한 균등물은 현재 공지된 균등물뿐만 아니라 미래에 개발되는 균등물, 즉, 구조에 무관하게 동일한 기능을 수행하도록 개발되는 임의의 구성요소들 모두를 포함하고자 하는 것이다.

[0056] 따라서, 예를 들어, 당업자라면, 본 명세서에 개시된 블록도는 본 발명을 구체화하는 예시적인 회로의 개념도를 나타낸다는 것을 이해할 수 있을 것이다. 마찬가지로, 임의의 플로우차트, 흐름도, 상태 천이도, 의사코드, 등은 실질적으로 컴퓨터 관통가능 매체로 표현되어, 컴퓨터 또는 프로세서가 명시적으로 도시되었건 도시되지 않았건 간에, 컴퓨터 또는 프로세서에 의해 실행될 수 있는 다양한 프로세스들을 나타낸다는 것을 이해할 수 있을 것이다.

[0057] 도면에 나타낸 다양한 구성요소들의 기능들은 전용의 하드웨어뿐만 아니라 적절한 소프트웨어와 연계하여 소프트웨어를 실행할 수 있는 하드웨어를 사용하는 것을 통해 제공될 수 있다. 기능들은, 프로세서에 의해 구성되는 경우, 하나의 전용 프로세서에 의해, 하나의 공유 프로세서에 의해, 또는 그중 일부는 공유되는 복수의 개별 프로세서들에 의해 구성될 수 있다. 또한, "프로세서" 또는 "컨트롤러"라는 용어를 명시적으로 사용하는 것은, 소프트웨어를 실행할 수 있는 하드웨어를 배제하는 것을 말하는 것으로 이해되어서는 아니되며, 함축적으로는, 이에 한하지는 않지만, 디지털 신호 프로세서(DSP) 하드웨어, 소프트웨어를 저장하는 ROM, RAM, 및 불휘발성 저장장치를 포함할 수 있다.

[0058] 또한, 기타의 하드웨어, 종래의 및/또는 관용의 것들이 포함될 수 있다. 마찬가지로, 도면들에 도시된 임의의 스위치들은 개념적인 것일 뿐이다. 그 기능은 전용의 논리를 통해, 프로그램 컨트롤과 전용의 논리의 상호작용을 통해, 또는 심지어 수동으로 실행될 수 있으며, 그 특정 기법은 문맥상 더 구체적으로 알 수 있듯이 구현하는 자에 의해 선택가능하다.

[0059] 그 청구범위에 있어서, 구체적인 기능을 수행하는 수단으로서 표현되는 어떠한 구성요소라도, 예를 들어, a) 그 기능을 수행하는 회로 구성요소의 조합 또는 b) 그 기능을 수행하기 위해 소프트웨어를 실행하는 적절한 회로와 조합된 펌웨어, 마이크로코드 등을 포함하는 임의의 형태의 소프트웨어를 포함하는 기능을 수행하는 임의의 방법을 포함하고자 하는 것이다. 이러한 청구범위에 의 정의된 바와 같은 본 발명은, 인용된 다양한 수단들에 의해 제공되는 기능성들이 청구범위에서 요청하는 방식으로 함께 조합 및 결합된다는 사실에 귀속한다. 따라서, 그러한 기능성들을 제공할 수 있는 임의의 수단은 본 명세서에 기재된 바와 균등한 것으로 간주되어야 할 것이다.

[0060] 본 발명의 "일 실시예" 또는 "실시예"라고 명세서에 일컫는 것은, 실시예와 연계하여 명시되는 특정한 특징, 구조, 특성, 및 등이 본 발명의 적어도 하나의 실시예에 포함된다는 것을 의미한다. 따라서, 본 명세서의 각종 위치에서 "일 실시예" 또는 "실시예"라는 어구가 등장하면, 반드시 모두 동일한 실시예를 지칭하는 것은 아니다.

[0061] 본 명세서에서 사용되는 바로서, "하이 레벨 신팩스"라는 것은, 계층적으로 매크로블록 레이어 위에 존재하는 비트스트림에서의 신팩스를 일컫는다. 예컨대, 하이 레벨 신팩스는, 본 명세서에서 사용되는 바와 같이, 슬라이스 헤더 레벨, SEI(Supplemental Enhancement Information) 레벨, 픽쳐 파라미터 세트 레벨, 시퀀스 파라미터 세트 레벨 및 NAL 유닛 헤더 레벨(이에 한하지 않음)에서의 신팩스를 일컫을 수도 있다.

[0062] 또한, 본 명세서에서 사용되는 바로서, "이전에 사용되지 않은 신팩스"라는 것은, 임의의 현재 공지된 비디오 코딩 표준 및 권고사항(recommendations) 및, 이에 한하지 않지만, MPEG-4 AVC 표준을 포함하는 그 확장판에 아직 존재하지 않는 신팩스를 일컫는다.

[0063] 또한, 본 명세서에서 사용되는 바로서, "코딩 오더 정보(coding order information)"라는 것은, 비트스트림 내의 픽쳐들이 코딩 및/또는 디코딩되는 순서를 나타내는 비디오 비트스트림에 존재하는 정보를 일컫는다. 코딩 오더 정보는, 예를 들어, frame_num을 포함할 수 있다.

[0064] 또한, 본 명세서에서 사용되는 바로서, "출력 오더 정보"라는 것은, 비트스트림 내의 픽쳐들이 출력되는 순서를 나타내는 비디오 비트스트림에 존재하는 정보를 일컫는다. 출력 순서 정보는, 예를 들어, 픽쳐 오더 카운트(POC) 값을 포함할 수 있다.

[0065] 또한, 본 발명은 본 명세서에서 MPEG-4 AVC 표준에 대하여 설명되고 있지만, 이 표준에만 한하지 않고, 본 발명의 개념을 유지하면서, MPEG-4 AVC 표준의 확장판을 포함하는 다른 비디오 코딩 표준 및 그 확장판에 대하여도 활용될 수 있다는 것을 이해할 수 있을 것이다.

- [0066] 또한, 본 명세서에서 호환가능하게 사용되는 바로서, "크로스-뷰, 인터-뷰(cross-view 및 inter-view)"라는 것은, 모두 현재의 뷰와는 다른 뷰에 속한 핵심을 일컫는다.
- [0067] 도 2a를 참조하면, 멀티뷰 비디오 코딩(MVC) 인코더의 일례가 참조 번호 100에 개략적으로 나타내어져 있다. 인코더(100)는 변환기(110)의 입력에 신호 통신하도록 연결된 출력을 갖는 결합기(105)를 포함한다. 변환기(110)의 출력은 양자화기(115)의 입력에 신호 통신하도록 연결된다. 양자화기(115)의 출력은 엔트로피 코더(120)의 입력 및 역 양자화기(125)의 입력에 신호 통신하도록 연결된다. 역 양자화기(125)의 출력은 역 변환기(130)의 입력에 신호 통신하도록 연결된다. 역 변환기(130)의 출력은 결합기(135)의 제1 비반전 입력에 신호 통신하도록 연결된다. 결합기(135)의 출력은 인트라 예측기(145)의 입력 및 디블록킹 필터(150)의 입력에 신호 통신하도록 연결된다. 디블록킹 필터(150)의 출력은 레퍼런스 핵심 스토어(155)의 입력에 신호 통신하도록 연결된다(뷰 i에 대하여). 레퍼런스 핵심 스토어(155)의 출력은 모션 보상기(175)의 제1 입력 및 모션 예측기(180)의 제1 입력에 신호 통신하도록 연결된다. 모션 예측기(180)의 출력은 모션 보상기(175)의 제2 입력에 신호 통신하도록 연결된다.
- [0068] (다른 뷰에 대한) 레퍼런스 핵심 스토어(160)의 출력은 디스패리티 예측기(170)의 제1 입력 및 디스패리티 보상기(165)의 제1 입력에 신호 통신하도록 연결된다. 디스패리티 예측기(170)의 출력은 디스패리티 보상기(165)의 제2 입력에 신호 통신하도록 연결된다.
- [0069] 엔트로피 디코더(120)의 출력은 인코더(100)의 출력으로 활용가능하다. 결합기(105)의 비반전 입력은 인코더(100)의 입력으로서 활용가능하며, 디스패리티 예측기(170)의 제2 입력 및 모션 예측기(180)의 제2 입력에 신호 통신하도록 연결된다. 스위치(185)의 출력은 결합기(135)의 제2 비반전 입력 및 결합기(105)의 반전 입력에 신호 통신하도록 연결된다. 스위치(185)는 모션 보상기(175)의 출력과 신호 통신하도록 연결된 제1 입력, 디스패리티 보상기(165)의 출력에 신호 통신하도록 연결된 제2 입력, 및 인트라 예측기(145)의 출력에 신호 통신하도록 연결된 제3 입력을 포함한다.
- [0070] 도 2b를 참조하면, 멀티뷰 비디오 코딩(MVC) 디코더의 일례가 참조 번호 3200으로 개략적으로 나타내어져 있다. 디코더(3200)는 역 양자화기(3210)의 입력에 신호 통신하도록 연결된 출력을 갖는 엔트로피 디코더(3205)를 포함한다. 역 양자화기의 출력은 역 변환기(3215)의 입력에 신호 통신하도록 연결된다. 역 변환기(3215)의 출력은 결합기(3220)의 제1 비반전 입력에 신호 통신하도록 연결된다. 결합기(3220)의 출력은 디블록킹 필터(3225)의 입력 및 인트라 예측기(3230)의 입력에 신호 통신하도록 연결된다. 디블록킹 필터(3225)의 출력은 (뷰 i에 대한) 레퍼런스 핵심 스토어(3240)의 입력에 신호 통신하도록 연결된다. 레퍼런스 핵심 스토어(3240)의 출력은 모션 보상기(3235)의 제1 입력에 신호 통신하도록 접속된다.
- [0071] (다른 뷰에 대한) 레퍼런스 핵심 스토어(3245)의 출력은 디스패리티 보상기(3250)의 제1 입력에 신호 통신하도록 연결된다.
- [0072] 엔트로피 코더(3205)의 입력은 나머지 비트스트림을 수신하기 위하여 디코더(3200)의 입력으로서 활용가능하다. 또한, 스위치(3255)의 제어 입력은 스위치(3255)에 의해 어느 입력이 선택되는지를 제어하기 위하여 콘트롤 선택스를 수신하기 위한 디코더(3200)의 입력으로서 활용가능하다. 또한, 모션 보상기(3235)의 제2 입력은 모션 백터를 수신하기 위하여 디코더(3200)의 입력으로 활용가능하다. 또한, 디스패리티 보상기(3250)의 제2 입력은 디스패리티 백터를 수신하기 위하여 디코더(3200)의 입력으로서 활용가능하다.
- [0073] 스위치(3255)의 출력은 결합기(3220)의 제2 비반전 입력에 신호 통신하도록 연결된다. 스위치(3255)의 제1 입력은 디스패리티 보상기(3250)의 출력에 신호 통신하도록 연결된다. 스위치(3255)의 제2 입력은 모션 보상기(3235)의 출력에 신호 통신하도록 연결된다. 스위치(3255)의 제3 입력은 인트라 예측기(3230)의 출력에 신호 통신하도록 연결된다. 모드 모듈(3260)의 출력은 스위치(3255)에 의해 어느 입력이 선택되는지를 제어하기 위하여 스위치(3225)에 신호 통신하도록 연결된다. 디블록킹 필터(3225)의 출력은 디코더의 출력으로서 활용가능하다.
- [0074] 본 발명의 원리에 따르면, 멀티뷰 비디오 시퀀스의 효율적인 코딩을 위하여 MPEG-4 AVC 표준의 하이 레벨 선택스에 대하여 일부 변화가 제안된다. 일 실시예에 있어서, 멀티뷰 비디오 시퀀스를 코딩하는 경우 뷰(view) 사이에 프레임 넘버(frame_num) 및/또는 핵심 오더 카운트(POC) 값을 분리(decouple)하도록 제안된다. 하나의 가능한 적용예로서, MPEG-4 AVC에 준거한 디코딩 및 출력 프로세스를 각 뷰에 대하여 독립적으로 적용할 수 있다. 일 실시예에 있어서, 각각의 뷰에 대하여 뷰 id를 송신함으로써 뷰 사이의 프레임 넘버 및/또는 핵심 오더 카운트 값들이 분리된다. 이전에, 뷰 보간/합성, 뷰 랜덤 액세스, 병렬 처리, 등을 포함하는 몇몇 멀티뷰 비디오

코딩(MVC) 요구사항에 있어서 view_id 정보가 필요하므로, 하이 레벨 신팩스에 뷰 식별자(view_id)를 추가할 것이 다만 제안되었다. view_id 정보는 크로스-뷰(cross-view) 예측에 관한 특수한 코딩 모드에서만 유용할 수 있다. 본 발명의 원리에 따라서 멀티뷰 비디오 콘텐츠의 뷰들 사이에서 프레임 넘버와 꾹쳐 오더 카운트 값을 분리시키기 위하여 사용되는 것이 view_id이다. 또한, 일 실시예에 있어서, 멀티뷰 비디오 코딩에 대한 MPEG-4 AVC 표준에 있어서 코딩 툴을 고치기 위한 솔루션이 제안된다.

[0075] 일 실시예에 따르면, 각각의 뷰는 상이한 view_id를 가지게 될 것이며, 따라서, 상이한 뷰들에 대하여 동일한 frame_num 및 POC가 재사용되도록 한다.

T0	T8	T4	T2	T6	T1	T3	T5	T7	(Time)	
S0	I0	I8	B4	B2	B6	B1	B3	B5	B7	(View 0 slice types)
S1	B0	B8	B4	B2	B6	B1	B3	B5	B7	(View 1 slice types)
S2	P0	P8	B4	B2	B6	B1	B3	B5	B7	(View 2 slice types)
0	1	2	3	4	5	6	7	8	(frame_num)	

[0076]

[0077] 별별 처리를 가능하도록 하기 위하여 상이한 뷰들이 코딩될 수 있는 방법이 수개 존재한다. 하나의 가능한 방법으로서, GOP에 대하여 하나의 뷰 내의 꾹쳐들이 먼저 코딩되고, 이어서, 동일한 GOP 사이즈에 대하여 또 다른 뷰로부터의 꾹쳐들이 코딩되는 식으로 그 GOP에 대하여 모든 뷰들이 인코딩될 때까지 인코딩하는 방법이 있다. 이 후, 다른 GOP들에 대하여 처리가 반복된다. 상기 예시에 있어서, 맨 먼저 뷰 S0 내의 꾹쳐들이 코딩되고, 이어서, 뷰 S2로부터의 꾹쳐들이 코딩되고, 뷰 S1의 꾹쳐들이 코딩된다.

[0078] 또 다른 가능한 방법으로서, 제일 먼저 코딩되어야 할 동일한 시간에 속하는 모든 뷰의 모든 꾹쳐들을 코딩하고, 이어서, 모든 뷰들의 또 다른 시간에 속하는 또 다른 꾹쳐들의 집합을 코딩하는 것이다. 이러한 프로세스는 모든 꾹쳐들이 코딩될 때까지 반복된다. 상기 예시에 있어서, 제일 먼저, 뷰 S0, S1, S2의 모든 꾹쳐들이 시간 T0에서 코딩되고, 이어서, T8, T2, 등에서 코딩된다. 본 발명은 꾹쳐들이 인코딩되는 순서에 대하여는 계의치 않는다.

[0079] 이하, 본 발명의 다양한 실시예에 따라서 MPEG-4 AVC 표준에 대한 변경에 대하여 말하고자 한다. 또한, 하나 이상의 변경이 멀티뷰 시퀀스의 별별 코딩을 가능하게 할 수 있는지를 보여줄 것이다. 그러나, 본 명세서에서 본 발명이 MPEG-4 AVC 표준에 대하여 기재되어 있지만, 본 명세서에 제공되는 본 발명의 교시에 관한 업계의 당업자라면 쉽게 이해할 수 있듯이, 본 발명의 범주를 유지하면서, MPEG-4 AVC 표준의 확장판 뿐만 아니라 다른 비디오 코딩 표준과 권고사항들 및 그 확장판에 대하여 구현될 수 있다는 것을 이해할 수 있을 것이다.

디코디드 레퍼런스 꾹쳐 마킹 프로세스

[0080] 현재의 MPEG-4 AVC 표준에 있어서, 디코딩된 꾹쳐 버퍼(DPB: decoded picture buffer) 내에 동일한 frame_num 을 갖는 다수의 꾹쳐들을 갖도록 허용되지 않는다. 그러나, 본 발명의 일 실시예에 따르면, frame_num 및/또는 꾹쳐 오더 카운트를 분리(decouple)시키므로, 멀티뷰 비디오 코딩(MVC)에 있어서 이러한 제한이 완화될 수 있으며, 즉, 각각의 뷰(view)가 자신만의 독립적인 frame_num 및/또는 꾹쳐 오더 카운트 값을 가질 것을 제안한다. 이를 위해서, 일 실시예에 있어서, view_id와 디코딩된 꾹쳐를 연관시킨다. 이는 각각의 꾹쳐에 대하여 또 다른 차원을 도입한다. 따라서, 일 실시예에 있어서, 디코디드 레퍼런스 꾹쳐 마킹 프로세스는 view_id를 포함하도록 재정의된다.

[0081] MPEG-4 AVC 표준이 디코딩된 레퍼런스 꾹쳐 마킹을 허용하는 방법에는 2개의 방법이 있다. MPEG-4 AVC 표준에서 디코딩된 레퍼런스 꾹쳐 마킹에 대한 첫번째 방법은 슬라이딩 윈도우(sliding window) 디코딩된 레퍼런스 꾹쳐 마킹을 포함한다. MPEG-4 AVC 표준에서 디코딩된 레퍼런스 꾹쳐 마킹에 대한 두번째 방법은 적응성 메모리 컨트롤(adaptive memory control) 디코딩된 레퍼런스 꾹쳐 마킹을 포함한다.

[0082] 본 발명의 다양한 실시예에 따르면, 이러한 방법들 중 하나 이상이 슬라이스 헤더에 존재하는 새로운 view_id를 고려하여 변경된다. 표 1은 본 발명의 일 실시예에 따른 슬라이스 헤더 신팩스를 나타낸다.

표 1

	C	Descriptor
<code>slice_header() {</code>		
<code>first_mb_in_slice</code>	2	<code>ue(v)</code>
<code>slice_type</code>	2	<code>ue(v)</code>
<code>pic_parameter_set_id</code>	2	<code>ue(v)</code>
<code>if(nal_unit_type == 22 nal_unit_type == 23) {</code>		
<code>view_parameter_set_id</code>	2	<code>ue(v)</code>
<code>view_id</code>	2	<code>ue(v)</code>
<code>}</code>		
<code>frame_num</code>	2	<code>u(v)</code>
<code>if(!frame_mbs_only_flag) {</code>		
<code>field_pic_flag</code>	2	<code>u(l)</code>
<code>if(field_pic_flag)</code>		
<code>bottom_field_flag</code>	2	<code>u(l)</code>
<code>}</code>		
<code>.....</code>		
<code>}</code>		

[0084]

MPEG-4 AVC 표준에서 디코딩된 레퍼런스 픽쳐 마킹의 첫번째 방법에 있어서, 동일한 frame_num/POC 값을 갖지만 상이한 view_id 값을 갖는 픽쳐들이 존재하는 경우 디폴트 거동이 지정되어야 한다. 본 발명에 따른 이러한 디폴트 거동의 일 실시예는 현재의 디코딩된 픽쳐와 동일한 view_id를 갖는 픽쳐들에 대하여 MMCO 커맨드를 적용하는 것뿐이다.

[0085]

MPEG-4 AVC 표준에서 디코딩된 레퍼런스 픽쳐 마킹의 두번째 방법에 있어서, 본 발명에 따른 다양한 실시예들이 제공되며, 여기서는, 새로운 메모리 관리 컨트롤 동작(MMCO: Memory Management Control Operation)을 도입하거나, 및/또는 마킹될 필요가 있는 픽쳐의 view_id를 고려하여 MPEG-4 AVC 표준에서 기존의 MMCO 커맨드를 변경시킨다. 기존의 MMCO(memory_management_control_operation이 1인 경우)를 재정의하는 일실시예에서는, 이하를 포함한다:

Let picNumX be specified by the following:

`picNumX = CurrPicNum - (difference_of_pic_nums_minus1 + 1).`

`viewIdX = CurrViewId - (difference_of_view_ids_minus1 + 1).`

[0086]

여기서, picNumX, CurrPicNum, difference_of_pic_nums_minus1 은 현재의 MPEG-4 AVC 표준에 정의된 바이며, viewIdX는 MMCO 커맨드를 사용하여 마킹되어야 하는 픽쳐의 viewId이며, CurrViewId는 현재의 디코딩되는 픽쳐의 viewId이며, difference_of_view_ids_minus1 은 현재의 viewId와 MMCO 커맨드를 사용하여 마킹되어야 하는 픽쳐의 viewId 사이의 차이이다.

[0087]

또한, 슬라이딩 윈도우 디코딩된 레퍼런스 픽쳐 마킹 프로세스의 디폴트 거동에 있어서, 현재의 픽쳐와 동일한 view_id를 갖는 픽쳐들만이 "레퍼런스로 사용되지 않음"으로 마킹되는 것으로 간주되어야 한다.

[0088]

도 3을 참조하면, 변형된 디코디드 레퍼런스 픽쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례가 뷰 퍼스트 코딩(view first coding)을 이용하는 참조 번호 300에 의해 개략적으로 나타내어져 있다.

[0089]

본 방법(300)은 시작 블록(305)에서 기능 블록 310을 진행한다. 기능 블록 310에서는, 인코더 형상 파일을 판독하여, 기능 블록 315에 진행한다. 기능 블록 315는 뷰의 갯수를 N으로 하고, 변수 i (뷰 넘버 인덱스) 및 j (픽쳐 넘버 인덱스) 모두를 0으로 설정하여, 결정 블록 320에 진행한다. 결정 블록 320에서는 i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 결정 블록 325에 진행한다. 그렇지 않다면, 종료 블록 399에 진행한다.

[0090]

결정 블록 325에서는 j가 뷰 i의 픽쳐 갯수보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 330에 진행한다. 그렇지 않다면, 기능 블록 350에 진행한다.

[0091]

기능 블록 330에서는, 뷰 i 내의 픽쳐 j를 인코딩하고, j를 증분하고, 결정 블록 335에 진행한다. 결정 블록 335에서는, MMCO 커맨드와 현재의 픽쳐가 연관되는지 여부를 판정한다. 만일 그렇다면, 기능 블록 340에 진행한다. 그렇지 않다면, 기능 블록 355에 진행한다.

[0092]

기능 블록 340에서는, difference_of_pic_nums_minus1 과 difference_of_view_ids_minus1 를 계산하여, "레퍼런스로 사용되지 않음"으로 마킹되어야할 레퍼런스 픽쳐의 픽쳐 및 view_id를 판정하고, 기능 블록 345에 진행

한다. 그렇지 않다면, 기능 블록 355에 진행한다.

한다. 기능 블록 345에서는, 현재의 픽쳐를 디코딩된 픽쳐 버퍼(DBP)에 삽입하고, 기능 블록 360에 진행한다. 기능 블록 360에서는, 현재의 view_id에 대한 frame_num 및 픽쳐 오더 카운트(POC)를 변경시키고, 기능 블록 325에 진행한다.

- [0095] 기능 블록 350에서는, i를 증분시키고, frame_num 및 픽쳐 오더 카운트(POC)를 리셋하여, 결정 블록 320에 복귀한다.
- [0096] 기능 블록 355에서는, 슬라이딩 윈도우 레퍼런스 픽쳐 마킹을 위해 MPEG-4 AVC 프로세스에 의해 사용하기 위하여, "레퍼런스로 사용하지 않음"으로 마킹되어야 하는 현재의 픽쳐의 view_id와 동일한 view_id를 갖는 픽쳐를 선택하여, 블록 355에 진행한다.
- [0097] 도 4를 참조하면, 변형된 디코디드 레퍼런스 픽쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례가 참조 번호 400으로 개략적으로 나타내어져 있다.
- [0098] 방법 400은 시작 블록 405에서 기능 블록 410으로 진행한다. 기능 블록 410에서는, 비트스트림, view_id, frame_num, 및 픽쳐 오더 카운트(POC)를 파싱하여, 기능 블록 415에 진행한다. 기능 블록 415에서는, 현재의 픽쳐를 디코딩하고, 결정 블록 420에 진행한다. 결정 블록 420에서는, MMCO 커맨드가 존재하는지 여부를 판정한다. 만일 그렇다면, 기능 블록 425에 진행한다. 그렇지 않다면, 기능 블록 440에 진행한다.
- [0099] 기능 블록 425에서는, difference_of_pic_nums_minus1 과 difference_of_view_ids_minus1 를 파싱하여, "레퍼런스로 사용하지 않음"으로 되는 레퍼런스 픽쳐의 픽쳐와 view_id를 판정하고, 기능 블록 430에 진행한다. 기능 블록 430에서는, 현재의 픽쳐를 디코더 픽쳐 버퍼(DBP)에 삽입하고, 결정 블록 435에 진행한다. 결정 블록 435에서는, 모든 픽쳐들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 종료 블록 499에 진행한다. 그렇지 않다면, 기능 블록 410에 진행한다.
- [0100] 기능 블록 440에서는, 슬라이딩 윈도우 디코딩된 레퍼런스 픽쳐 마킹을 위해 MPEG-4 AVC 프로세서에서 사용하기 위하여, "레퍼런스로 사용하지 않음"으로 마킹되는 현재의 픽쳐의 view_id와 동일한 view_id를 갖는 픽쳐를 선택하고, 기능 블록 430에 진행한다.
- [0101] 도 15를 참조하면, 변형된 디코디드 레퍼런스 픽쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례가 참조 번호 1500으로 개략적으로 나타내어져 있다.
- [0102] 본 방법(1500)은 시작 블록 1505에서 기능 블록 1510에 진행한다. 기능 블록 1510에서는, 인코더 형상 파일을 판독하고, 기능 블록 1515에 진행한다. 기능 블록 1515에서는, 뷰의 갯수를 N으로 하고, 변수 i (뷰 넘버 인덱스) 및 j (픽쳐 넘버 인덱스) 모두를 0으로 설정하여, 결정 블록 1520에 진행한다. 결정 블록 1520에서는, i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 결정 블록 1525에 진행한다. 그렇지 않다면, 종료 블록 1599에 진행한다.
- [0103] 결정 블록 1525에서는, j가 뷰 i에서 픽쳐들의 수보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 1530에 진행한다. 그렇지 않다면, 기능 블록 1550에 진행한다.
- [0104] 기능 블록 1530에서는, 뷰 i의 픽쳐 j를 인코딩하고, j를 증분시키고, 결정 블록 1535에 진행한다. 결정 블록 1535에서는, MMCO 커맨드가 현재의 픽쳐에 연관되는지 여부를 판정한다. 만일 그렇다면, 기능 블록 1540에 진행한다. 그렇지 않다면, 기능 블록 1555에 진행한다.
- [0105] 기능 블록 1540에서는, 현재 픽쳐의 view_id와 동일한 view_id를 갖는 픽쳐에 대해서만 관련된 MMCO 커맨드를 수행하고, 기능 블록 1545에 진행한다. 기능 블록 1545에서는, 현재의 픽쳐를 디코딩된 픽쳐 버퍼(DBP)에 삽입하고, 기능 블록 1560에 진행한다. 기능 블록 1560에서는, 현재 view_id의 frame_num 및 픽쳐 오더 카운트(POC)를 변경하고, 기능 블록 1525에 복귀한다.
- [0106] 기능 블록 1550에서는, i를 증분시키고, frame_num 및 픽쳐 오더 카운트(POC)를 리셋하여, 결정 블록 1520에 복귀한다.
- [0107] 기능 블록 1555에서는, 슬라이딩 윈도우 레퍼런스 픽쳐 마킹을 위해 MPEG-4 AVC 프로세스에서 사용하기 위하여 "레퍼런스로 사용되지 않음"으로 마킹되어야 하는 현재의 픽쳐의 view_id와 동일한 view_id를 갖는 픽쳐를 선택하고, 기능 블록 1555에 진행한다.
- [0108] 도 16을 참조하면, 변형된 디코디드 레퍼런스 픽쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례가 참조 번호 1600으로 개략적으로 나타내어져 있다.

- [0109] 본 방법(1600)은 시작 블록 1605에서 기능 블록 1610에 진행한다. 기능 블록 1610에서는, 비트스트림, view_id, frame_num 및 픽쳐 오더 카운트(POC)를 파싱하여, 기능 블록 1615에 진행한다. 기능 블록 1615에서는 현재의 픽쳐를 디코딩하여, 결정 블록 1620에 진행한다. 결정 블록 1620에서는, MMCO 커맨드가 존재하는지 여부를 판정한다. 만일 그렇다면, 기능 블록 1625에 진행한다. 그렇지 않다면, 기능 블록 1640에 진행한다.
- [0110] 기능 블록 1625에서는, MMCO 커맨드를 파싱하여, 현재의 픽쳐의 view_id와 동일한 view_id를 갖는 픽쳐에 대해서만 MMCO 커맨드를 수행하고, 기능 블록 1630에 진행한다. 기능 블록 1630에서는, 현재의 픽쳐를 디코더 픽쳐 버퍼(DPB)에 삽입하고, 결정 블록 1635에 진행한다. 결정 블록 1635에서는, 모든 픽쳐들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 종료 블록 1699에 진행한다. 그렇지 않다면, 기능 블록 1610에 복귀한다.
- [0111] 기능 블록 1640에서는, 슬라이딩 윈도우 디코딩된 레퍼런스 픽쳐 마킹을 위해 MPEG-4 AVC 프로세스에서 사용하기 위하여 "레퍼런스로 사용되지 않음"으로 마킹되어야 하는 현재의 픽쳐의 view_id와 동일한 view_id를 갖는 픽쳐를 선택하고, 기능 블록 1630에 진행한다.
- [0112] **레퍼런스 픽쳐 리스트 구성**
- [0113] 본 발명의 일 실시예에 따르면, view_id와 디코딩된 레퍼런스 픽쳐들을 연관시킨다. 따라서, 일 실시예에 있어서, view_id를 포함하도록 레퍼런스 픽쳐들에 대한 초기화 프로세스 및 레퍼런스 픽쳐 리스트에 대한 리오더링 프로세스를 재정의한다.
- [0114] MPEG-4 AVC 표준에서는 P 및 B 슬라이스에 대한 레퍼런스 리스트를 초기화하도록 디폴트 프로세스를 지정한다. 이러한 디폴트 프로세스는 그 후 비트스트림 내에 존재하는 특수한 RPLR(Reference Picture List Reordering) 커맨드들에 의해 변형될 수 있다.
- [0115] 레퍼런스 픽쳐들의 이러한 디폴트 오더링 및 리오더링은 frame_num 및 픽쳐 오더 카운트 값에 기초한다. 그러나, 동일한 frame_num/POC 값이 디코더 픽쳐 버퍼(DPB) 내에 존재하도록 하기 때문에, view_id를 이용하여 동일한 frame_num/POC 값들을 구분할 필요가 있다. 일 실시예에 있어서, 레퍼런스 픽쳐 리스트를 설정하기 위한 하나 이상의 프로세스들이 변경된다.
- [0116] P 및 B 슬라이스에 대한 레퍼런스 리스트들을 초기화하기 위한 디폴트 초기화 프로세스의 일 실시예는, 레퍼런스 리스트 내에서 시간적인 레퍼런스 픽쳐들만을 허용하고, 현재의 픽쳐의 view_id와는 다른 view_id를 갖는 모든 픽쳐들은 무시하는 것을 포함한다. 시간적인 레퍼런스 픽쳐들은 MPEG-4 AVC 표준에 명시된 동일한 디폴트 초기화 프로세스를 따르게 된다. 또 다른 실시예에서는, 가장 근접한 view_id의 것이 리스트 내에서 보다 먼저 위치되도록 리스트 내의 크로스-뷰 레퍼런스만을 위치시키는 것을 포함한다. 또 다른 실시예에서는, 시간적인 레퍼런스를 먼저 이용하여 레퍼런스 리스트들을 초기화한 후, 특정의 고정된 위치에, 예를 들어, 구성중인 레퍼런스 리스트들의 말단에 크로스-뷰 레퍼런스 프레임들을 위치시키는 것을 포함한다.
- [0117] 리스트를 리오더링하기 위한 레퍼런스 픽쳐 리스트 리오더링 커맨드들에 대하여, 일 실시예에 있어서, 새로운 커맨드들이 도입되거나, 및/또는 이동될 필요가 있는 픽쳐의 view_id를 고려하여 기존 커맨드들의 시맨틱(semantics)이 변형된다.
- [0118] 일 실시예에 있어서, MPEG-4 AVC 표준에 지정된 레퍼런스 픽쳐 리스트 리오더링 커맨드들이 변하지 않고 남도록, 이 프로세스에 사용되는 MPEG-4 AVC 표준 변수들을 이하와 같이 재정의한다.
- [0119] 레퍼런스 리스트들을 리오더링하는 것과 관련된 MPEG-4 AVC 표준의 변수들을 재정의하는 일 실시예를 이하에 나타낸다. 본 실시예에 있어서, 이하의 사항을 적용한다:
- FrameNum = frame_num * N + view_id; and
- MaxFrameNum = $2^{(\log_2 \text{max_frame_num_minus4} + 4)} * N$.
- [0120] 이하와 같이 변수 CurrPicNum가 유도된다: field_pic_flag가 0과 같다면, CurrPicNum = frame_num * N + view_id와 동일하게 설정된다; 그렇지 않고, field_pic_flag가 1과 같다면, CurrPicNum = $2 * (\text{frame_num} * N + \text{view_id}) + 1$ 과 동일하게 설정된다.

[0122] MPEG-4 AVC 표준에서 슬라이스에 대한 픽쳐 오더 카운트는 이하와 같이 정의된다:

```

if( picX is a frame or a complementary field pair )
    PicOrderCnt( picX ) = Min( TopFieldOrderCnt, BottomFieldOrderCnt ) of the
    frame or complementary field pair picX
else if( picX is a top field )
    PicOrderCnt( picX ) = TopFieldOrderCnt of field picX
else if( picX is a bottom field )
    PicOrderCnt( picX ) = BottomFieldOrderCnt of field picX

```

[0123]

[0124] 멀티뷰 비디오 코딩 슬라이스에 대하여, 레퍼런스 픽쳐 리스트 구성을 위한 디코딩 프로세스 및 디코디드 레퍼런스 픽쳐 마킹 프로세스에 있어서, 이하와 같이 픽쳐 오더 카운트가 유도된다:

[0125] $\text{PicOrderCnt(picX)} = \text{PicOrderCnt(picX)} * N + \text{view_id};$

[0126] 여기서, N은 뷰의 갯수를 지칭한다. 뷰의 갯수는 비트스트림 내에서 하이 레벨 십택스를 사용하여 나타내어지며, 인-밴드(in-band) 또는 아웃-오브-밴드(out-of-band)로 운반될 수 있다. 하나의 실시예로서, 이를 MPEG-4 AVC 표준의 파라미터 셋트에 포함시키는 것이다(예컨대, Sequence Parameter Set (SPS), Picture Parameter Set (PPS), 또는 View Parameter Set (VPS)).

[0127] 레퍼런스 리스트들을 리오더링하는 것에 관련한 MPEG-4 AVC 표준의 변수들을 재정의하는 또 다른 실시예를 이하에 나타낸다. 본 실시예에서는, 이하와 같이 적용한다:

[0128] $\text{FrameNum} = \text{GOP_length} * \text{view_id} + \text{frame_num}.$

[0129] 멀티뷰 비디오 코딩 슬라이스들에 대하여, 레퍼런스 픽쳐 리스트 구성을 위한 디코딩 프로세스 및 디코디드 레퍼런스 픽쳐 마킹 프로세스에 있어서 픽쳐 오더 카운트는 이하와 같이 유도된다.

[0130] $\text{PicOrderCnt(picX)} = \text{PicOrderCnt(picX)} + \text{GOP_length} * \text{view_id},$

[0131] 여기서, GOP_length는 각각의 뷰에 있어서 앵커(anchor) 픽쳐 및 시간적으로 앵커 픽쳐와 이전의 앵커 픽쳐 사이에 위치되는 모든 픽쳐들로서 정의된다.

[0132] 또 다른 실시예에 있어서, 현재의 뷰와 동일한 view_id를 갖는 픽쳐들만을 적용하도록 기존의 RPLR 커맨드들의 시맨틱을 변경시킨다.

[0133] 도 5를 참조하면, 변형된 레퍼런스 픽쳐 리스트 구성을 사용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례가 참조 번호 500으로 개략적으로 나타내어져 있다. 본 방법(500)은 시작 블록(505)에서 기능 블록(510)에 진행한다. 기능 블록 510에서는, 인코더 형상 파일을 판독하고, 기능 블록 515에 진행한다. 기능 블록 515에서는, 뷰의 갯수가 변수 N과 동일하게 하고, 변수 i (뷰 넘버 인덱스) 및 j (픽쳐 넘버 인덱스)를 모두 0으로 설정하고, 결정 블록 520에 진행한다. 결정 블록 520에서는 i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 525에 진행한다. 그렇지 않다면, 종료 블록 599에 진행한다.

[0134] 기능 블록 525에서는, j가 뷰 i 내의 픽쳐 갯수보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 530에 진행한다. 그렇지 않다면, 기능 블록 545에 진행한다.

[0135] 기능 블록 530에서는, 인터 픽쳐들에 대하여, 레퍼런스 리스트 초기화를 위해 MPEG-4 AVC 프로세서에 의해 사용하기 위하여 현재 픽쳐의 view_id와 동일한 view_id를 갖는 픽쳐들만을 포함시킨고, 기능 블록 532에 진행한다. 기능 블록 532에서는, 레퍼런스 리스트를 리오더링하고, 기능 블록 535에 진행한다. 기능 블록 535에서는, 뷰 i의 픽쳐 j를 인코딩하고, j를 중분시키고, 기능 블록 540에 진행한다. 기능 블록 540에서는, frame_num 및 픽쳐 오더 카운트(POC)를 중분시키고, 결정 블록 525에 진행한다.

[0136] 기능 블록 545에서는, i를 중분시키고, frame_num 및 픽쳐 오더 카운트(POC)를 리셋하고, 결정 블록 520에 진행한다.

[0137] 도 6을 참조하면, 변형된 레퍼런스 픽쳐 리스트 구성을 사용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 또 다른 일례가 참조 번호 600으로 개략적으로 나타내어져 있다.

[0138] 본 방법(600)에서는, 시작 블록 605에서 기능 블록 610에 진행한다. 기능 블록 610에서는, 인코더 형상 파일을

판독하고, 기능 블록 615에 진행한다. 기능 블록 615에서는, 뷔의 갯수가 변수 N과 동일하도록 하고, 변수 i (뷰 넘버 인덱스) 및 j (픽쳐 넘버 인덱스) 모두를 0으로 설정하고, 결정 블록 620에 진행한다. 결정 블록 620에서는, i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 625에 진행한다. 그렇지 않다면, 종료 블록 699에 진행한다.

[0139] 기능 블록 625에서는, j가 뷔 i의 픽쳐 갯수보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 630에 진행한다. 그렇지 않다면, 기능 블록 645에 진행한다.

[0140] 기능 블록 630에서는, 인터 픽쳐들에 대하여, 현재 픽쳐의 view_id와는 상이한 view_id를 갖는 픽쳐들만으로 레퍼런스 리스트들을 초기화하고, 현재 픽쳐와 동일한 시간에 샘플링하고, 가장 근접한 view_id의 것이 리스트 내에서 보다 먼저 위치되도록 오더링하여, 기능 블록 632에 진행한다. 기능 블록 632에서는, 레퍼런스 리스트를 리오더링하고, 기능 블록 635에 진행한다. 기능 블록 635에서는, 뷔 i의 픽쳐 j를 인코딩하고, j를 증분시키고, 기능 블록 640에 진행한다. 기능 블록 640에서는, frame_num 및 픽쳐 오더 카운트(POC)를 증분시키고, 결정 블록 625에 진행한다.

[0141] 기능 블록 645에서는, i를 증분시키고, frame_num 및 픽쳐 오더 카운트(POC)를 리셋시키고, 결정 블록 620에 진행한다.

[0142] 도 7을 참조하면, 변형된 레퍼런스 픽쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 또 다른 일례의 방법이 참조 번호 700으로 개략적으로 나타내어져 있다.

[0143] 본 방법(700)에서는, 시작 블록 705에서 기능 블록 710에 진행한다. 기능 블록 710에서는, 인코더 형상 파일을 판독하고, 기능 블록 715에 진행한다. 기능 블록 715에서는, 뷔의 갯수를 변수 N과 동일하게 하여, 변수 i (뷰 넘버 인덱스) 및 j (픽쳐 넘버 인덱스)를 모두 0으로 설정하고, 결정 블록 720에 진행한다. 결정 블록 720에서는, i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 725에 진행한다. 그렇지 않다면, 종료 블록 799에 진행한다.

[0144] 기능 블록 725에서는, j가 뷔 i의 픽쳐의 갯수보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 730에 진행한다. 그렇지 않다면, 기능 블록 745에 진행한다.

[0145] 기능 블록 730에서는, 레퍼런스 리스트 초기화를 위해 MPEG-4 AVC 프로세서에 의해 사용하기 위하여 현재 픽쳐의 view_id와 동일한 view_id를 갖는 픽쳐들만을 포함시키고, 기능 블록 732에 진행한다. 기능 블록 732에서는, 현재의 픽쳐와 동일한 시간적인 위치를 갖는 크로스-뷰 픽쳐들을 레퍼런스 리스트의 말단에 삽입하고, 기능 블록 735에 진행한다. 기능 블록 735에서는, 뷔 i의 픽쳐 j를 인코딩하고, j를 증분시키고, 기능 블록 740에 진행한다. 기능 블록 740에서는, frame_num 및 픽쳐 오더 카운트(POC)를 증분시키고, 결정 블록 725에 진행한다.

[0146] 기능 블록 745에서는, i를 증분시키고, frame_num 및 픽쳐 오더 카운트(POC)를 리셋하고, 기능 블록 720에 복귀한다.

[0147] 도 8을 참조하면, 변형된 레퍼런스 픽쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례가 참조 번호 800로 개략적으로 나타내어져 있다. 본 방법(800)에서는, 시작 블록 805에서 기능 블록 810에 진행한다. 기능 블록 810에서는, 비트스트림, view_id, frame_num, 및 픽쳐 오더 카운트(POC)를 파싱하여, 기능 블록 815에 진행한다. 기능 블록 815에서는, 레퍼런스 리스트 초기화를 위해 MPEG-4 AVC 프로세서에 의해 사용하기 위하여 현재 픽쳐의 view_id와 동일한 view_id를 갖는 픽쳐들만을 포함시키고, 기능 블록 820에 진행한다. 기능 블록 820에서는, 현재의 픽쳐를 디코딩하여, 기능 블록 825에 진행한다. 기능 블록 825에서는, 현재의 픽쳐를 디코딩된 픽쳐 버퍼에 삽입하고, 결정 블록 830에 진행한다. 결정 블록 830에서는, 모든 픽쳐들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 종료 블록 899에 진행한다. 그렇지 않다면, 기능 블록 810에 진행한다.

[0148] 도 9를 참조하면, 변형된 레퍼런스 픽쳐 리스트 구성을 사용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 또 다른 일례가 참조 번호 900으로 개략적으로 나타내어져 있다. 본 방법(900)에서는 시작 블록 905에서 기능 블록 910에 진행한다. 기능 블록 910에서는 비트 스트림, view_id, frame_num, 및 픽쳐 오더 카운트(POC)를 파싱하고, 기능 블록 915에 진행한다. 기능 블록 915에서는, 현재의 픽쳐의 view_id와는 상이한 view_id를 갖는 픽쳐들만으로 레퍼런스 리스트들을 초기화시키고, 현재의 픽쳐와 동일한 시간에 샘플링되고, 가장 근접한 view_id의 것이 리스트 내에서 가장 먼저 위치되도록 오더링되어, 기능 블록 920에 진행한다. 기능 블록 920에서는, 현재의 픽쳐를 디코딩하고, 기능 블록 925에 진행한다. 기능 블록 925에서는, 현재의 픽쳐를 디코딩된

픽쳐 버퍼(DPB)에 삽입하고, 결정 블록 930에 진행한다. 결정 블록 930에서는, 모든 픽쳐들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 종료 블록 999에 진행한다. 그렇지 않다면, 기능 블록 910에 복귀한다.

[0149] 도 10을 참조하면, 변형된 레퍼런스 픽쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 또 다른 일례가 참조 번호 1000으로 개략적으로 나타내어져 있다. 본 방법(1000)은 시작 블록 1005에서 기능 블록 1010에 진행한다. 기능 블록 1010에서는, 비트 스트림, view_id, frame_num, 및 픽쳐 오더 카운트(POC)를 파싱하고, 기능 블록 1015에 진행한다. 기능 블록 1015에서는, 레퍼런스 리스트 초기화를 위해 MPEG-4 AVC 프로세스에 의해 사용하기 위하여 현재의 view_id와 동일한 view_id를 갖는 픽쳐들만을 포함시키고, 기능 블록 1020에 진행한다. 기능 블록 1020에서는, 현재의 픽쳐와 동일한 시간적 위치를 갖는 크로스-뷰 픽쳐들을 레퍼런스 리스트의 말단에 삽입하고, 기능 블록 1025에 진행한다. 기능 블록 1025에서는, 현재의 픽쳐를 디코딩된 픽쳐 버퍼에 삽입하고, 결정 블록 1030에 진행한다. 결정 블록 1030에서는, 모든 픽쳐들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 종료 블록 1099에 진행한다. 그렇지 않다면, 기능 블록 1010에 복귀한다.

[0150] 도 17을 참조하면, 변형된 레퍼런스 픽쳐 리스트 구성 및 프레임 넘버 계산을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례가 참조 번호 1700에 개략적으로 나타내어져 있다.

[0151] 본 방법(1700)은 시작 블록(1705)에서 기능 블록 1710에 진행한다. 기능 블록 1710에서는, 인코더 형상 파일을 판독하여, 기능 블록 1715에 진행한다. 기능 블록 1715에서는, 뷰의 갯수가 변수 N과 동일하도록 하여, 변수 i (뷰 넘버 인덱스) 및 j (픽쳐 넘버 인덱스)를 모두 0으로 설정하고, 결정 블록 1720에 진행한다. 결정 블록 1720에서는, i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 1725에 진행한다. 그렇지 않다면, 종료 블록 1799에 진행한다.

[0152] 기능 블록 1725에서는, j가 뷰 i내의 픽쳐들의 수보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 1730에 진행한다. 그렇지 않다면, 기능 블록 1745에 진행한다.

[0153] 기능 블록 1730에서는, $frame_num = frame_num * N + view_id$ 로 설정하고, $PicOrderCnt(picX) = PicOrderCnt(picX) * N + view_id$ 로 설정하고, 기능 블록 1735에 진행한다. 기능 블록 1735에서는, 뷰 i 내의 픽쳐 j를 인코딩하고, j를 증분시키고, 기능 블록 1740에 진행한다. 기능 블록 1740에서는, frame_num 및 픽쳐 오더 카운트(POC)를 증분시키고, 결정 블록 1725에 복귀한다.

[0154] 기능 블록 1745에서는, i를 증분시키고, frame_num 및 픽쳐 오더 카운트(POC)를 리셋시키고, 결정 블록 1720에 복귀한다.

[0155] 도 18을 참조하면, 변형된 레퍼런스 픽쳐 리스트 구성 및 프레임 넘버 계산을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 또 다른 일례가 참조 번호 1800으로 개략적으로 나타내어져 있다.

[0156] 본 방법(1800)은 시작 블록 1805에서 기능 블록 1810에 진행한다. 기능 블록 1810에서는, 인코더 형상 파일을 판독하여, 기능 블록 1815에 진행한다. 기능 블록 1815에서는 뷰의 갯수를 변수 N과 동일하게 하여, 변수 i (뷰 넘버 인덱스) 및 j (픽쳐 넘버 인덱스) 모두를 0으로 설정하고, 결정 블록 1820에 진행한다. 결정 블록 1820에서는 i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 1825에 진행한다. 그렇지 않다면, 종료 블록 1899에 진행한다.

[0157] 기능 블록 1825에서는 j가 뷰 i 내의 픽쳐의 수보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 1830에 진행한다. 그렇지 않다면, 기능 블록 1845에 진행한다.

[0158] 기능 블록 1830에서는, $frame_num = GOP_length * view_id + frame_num$ 으로 설정하고, $PicOrderCnt(picX) = PicOrderCnt(picX) + GOP_length * view_id$ 로 설정하고, 기능 블록 1835에 진행한다. 기능 블록 1835에서는 뷰 i 내의 픽쳐 j를 인코딩하고, j를 증분시키고, 기능 블록 1840에 진행한다. 기능 블록 1840에서는 frame_num 및 픽쳐 오더 카운트(POC)를 증분시키고, 결정 블록 1825에 진행한다.

[0159] 기능 블록 1845에서는 i를 증분시키고, frame_num 및 픽쳐 오더 카운트(POC)를 리셋시키고, 결정 블록 1820에 복귀한다.

[0160] 도 19를 참조하면, 변형된 레퍼런스 픽쳐 리스트 구성 및 프레임 넘버 계산을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례가 참조 번호 1900에 개략적으로 나타내어져 있다. 본 방법(1900)은 시작 블록(1905)에서 기능 블록 1910에 진행한다. 기능 블록 1910에서는, 비트 스트림, view_id, frame_num, 및 픽쳐 오더 카운트(POC)를 파싱하고, 기능 블록 1915에 진행한다. 기능 블록 1915에서는, $frame_num = frame_num * N +$

view_id로 설정하고, $\text{PicOrderCnt(picX)} = \text{PicOrderCnt(picX)} * N + \text{view_id}$ 로 설정하고, 기능 블록 1920에 진행한다. 기능 블록 1920에서는, 현재의 픽쳐를 디코딩하고, 기능 블록 1925에 진행한다. 기능 블록 1925에서는, 현재의 픽쳐를 디코딩된 픽쳐 버퍼(DPB)에 삽입하고, 결정 블록 1930에 진행한다. 결정 블록 1930에서는, 모든 픽쳐들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 종료 블록 1999에 진행한다. 그렇지 않다면, 기능 블록 1910에 진행한다.

[0161] 도 20을 참조하면, 변형된 레퍼런스 픽쳐 리스트 구성 및 프레임 넘버 계산을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 또 다른 일례가 참조 번호 2000에 개략적으로 나타내어져 있다. 본 방법(2000)은 시작 블록 (2005)에서 기능 블록 2010에 진행한다. 기능 블록 2010에서는, 비트 스트림, view_id, frame_num, 및 픽쳐 오더 카운트(POC)를 파싱하고, 기능 블록 2015에 진행한다. 기능 블록 2015에서는, $\text{frame_num} = \text{GOP_length} * \text{view_id_frame_num}$ 로 설정하고, $\text{PicOrderCnt(picX)} = \text{PicOrderCnt(picX)} + \text{GOP_length} * \text{view_id}$ 로 설정하고, 기능 블록 2020에 진행한다. 기능 블록 2020에서는, 현재의 픽쳐를 디코딩하고, 기능 블록 2025에 진행한다. 기능 블록 2025에서는, 현재의 픽쳐를 디코딩된 픽쳐 버퍼(DPB)에 삽입하고, 결정 블록 2030에 진행한다. 결정 블록 2030에서는, 모든 픽쳐들이 디코딩되었는지를 판정한다. 만일 그렇다면, 블록 2099에 진행한다. 그렇지 않다면, 기능 블록 2010에 진행한다.

[0162] 도 21을 참조하면, 레퍼런스 픽쳐 리스트 리오더링(RPLR) 커맨드를 가지고 변형된 레퍼런스 픽쳐 리스트 초기화를 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례가 참조 번호 2100에 개략적으로 나타내어져 있다.

[0163] 본 방법(2100)은 시작 블록 2105에서 기능 블록 2110에 진행한다. 기능 블록 2110에서는, 인코더 형상 파일을 판독하여, 기능 블록 2115에 진행한다. 기능 블록 2115에서는, 뷰의 갯수가 변수 N과 동일하도록 하여, 변수 i (뷰 넘버 인덱스) 및 j (픽쳐 넘버 인덱스)를 모두 0으로 설정하고, 결정 블록 2120에 진행한다. 결정 블록 2120에서는, i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 2125에 진행한다. 그렇지 않다면, 종료 블록 2199에 진행한다.

[0164] 기능 블록 2125에서는, j가 뷰 i내의 픽쳐들의 수보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 2130에 진행한다. 그렇지 않다면, 기능 블록 2145에 진행한다.

[0165] 기능 블록 2130에서는, 인터 픽쳐들에 대하여, 디폴트 레퍼런스 리스트 초기화를 수행하고, 기능 블록 2132에 진행한다. 기능 블록 2132에서는, 인코더 형상 파일로부터 RPLR 커맨드를 판독하고, 기능 블록 2134에 진행한다. 기능 블록 2134에서는, 현재의 픽쳐의 view_id와 동일한 view_id를 갖는 픽쳐에 대해서만 RPLR 커맨드를 수행하고, 기능 블록 2135에 진행한다. 기능 블록 2135에서는, 뷰 i 내의 픽쳐 j를 인코딩하고, j를 증분시키고, 기능 블록 2140에 진행한다. 기능 블록 2140에서는, frame_num 및 픽쳐 오더 카운트(POC)를 증분시키고, 결정 블록 2130에 복귀한다.

[0166] 기능 블록 2145에서는, i를 증분시키고, frame_num 및 픽쳐 오더 카운트(POC)를 리셋시키고, 결정 블록 2120에 복귀한다.

[0167] 도 22를 참조하면, 레퍼런스 픽쳐 리스트 리오더링(RPLR) 커맨드를 가지고 변형된 레퍼런스 픽쳐 리스트 초기화를 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 또 다른 일례가 참조 번호 2200에 개략적으로 나타내어져 있다.

[0168] 본 방법(2200)은 시작 블록 2205에서 기능 블록 2210에 진행한다. 기능 블록 2210에서는, 인코더 형상 파일을 판독하여, 기능 블록 2215에 진행한다. 기능 블록 2215에서는 뷰의 갯수를 변수 N과 동일하게 하여, 변수 i (뷰 넘버 인덱스) 및 j (픽쳐 넘버 인덱스) 모두를 0으로 설정하고, 결정 블록 2220에 진행한다. 결정 블록 2220에서는 i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 2225에 진행한다. 그렇지 않다면, 종료 블록 2299에 진행한다.

[0169] 기능 블록 2225에서는 j가 뷰 i 내의 픽쳐의 수보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 2230에 진행한다. 그렇지 않다면, 기능 블록 2245에 진행한다.

[0170] 기능 블록 2230에서는, 인터 픽쳐들에 대하여, 디폴트 레퍼런스 리스트 초기화를 수행하고, 기능 블록 2232에 진행한다. 기능 블록 2232에서는, 인코더 형상 파일로부터 RPLR 커맨드를 판독하고, 기능 블록 2234에 진행한다. 기능 블록 2234에서는, RPLR 커맨드에 지시된 view_id에서 지정된 픽쳐에 대하여 RPLR 커맨드를 수행하고, 기능 블록 2235에 진행한다. 기능 블록 2235에서는, 뷰 i 내의 픽쳐 j를 인코딩하고, j를 증분시키고, 기능 블록 2240에 진행한다. 기능 블록 2240에서는, frame_num 및 픽쳐 오더 카운트(POC)를 증분시키고, 결정 블록

2230에 복귀한다.

[0171] 기능 블록 2245에서는, i를 증분시키고, frame_num 및 꾹쳐 오더 카운트(POC)를 리셋시키고, 결정 블록 2220에 복귀한다.

[0172] 도 23을 참조하면, 레퍼런스 꾹쳐 리스트 리오더링(RPLR) 커맨드를 가지고 변형된 레퍼런스 꾹쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례가 참조 번호 2300에 개략적으로 나타내어져 있다. 본 방법(2300)은 시작 블록(2305)에서 기능 블록 2310에 진행한다. 기능 블록 2310에서는, 비트 스트림, view_id, frame_num, 및 꾹쳐 오더 카운트(POC)를 파싱하고, 기능 블록 2315에 진행한다. 기능 블록 2315에서는, 레퍼런스 리스트 초기화를 위해 MPEG-4 AVC 프로세스에 의해 사용하기 위하여 현재 꾹쳐의 view_id와 동일한 view_id를 갖는 꾹쳐들만을 포함시키고, 기능 블록 2317에 진행한다. 기능 블록 2317에서는 RPLR 커맨드들을 판독하고, 기능 블록 2319에 진행한다. 기능 블록 2319에서는, 현재의 꾹쳐의 view_id와 동일한 view_id를 갖는 꾹쳐에 대해서만 RPLR 커맨드들을 수행하고, 기능 블록 2320에 진행한다. 기능 블록 2320에서는 현재의 꾹쳐를 디코딩하여, 기능 블록 2325에 진행한다. 기능 블록 2325에서는, 현재의 꾹쳐를 디코딩된 꾹쳐 버퍼에 삽입하고, 결정 블록 2330에 진행한다. 결정 블록 2330에서는 모든 꾹쳐들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 종료 블록 2399에 진행한다. 그렇지 않다면, 기능 블록 2310에 복귀한다.

[0173] 도 24를 참조하면, 레퍼런스 꾹쳐 리스트 리오더링(RPLR) 커맨드를 가지고 변형된 레퍼런스 꾹쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 또 다른 일례가 참조 번호 2400에 개략적으로 나타내어져 있다. 본 방법(2400)은 시작 블록(2405)에서 기능 블록 2410에 진행한다. 기능 블록 2410에서는, 비트 스트림, view_id, frame_num, 및 꾹쳐 오더 카운트(POC)를 파싱하고, 기능 블록 2415에 진행한다. 기능 블록 2415에서는, 레퍼런스 리스트 초기화를 위해 MPEG-4 AVC 프로세스에 의해 사용하기 위하여 현재 꾹쳐의 view_id와 동일한 view_id를 갖는 꾹쳐들만을 포함시키고, 기능 블록 2417에 진행한다. 기능 블록 2417에서는 RPLR 커맨드들을 판독하고, 기능 블록 2419에 진행한다. 기능 블록 2419에서는, RPLR 커맨드에 표시된 view_id에서 지정된 꾹쳐에 대하여 RPLR 커맨드들을 수행하고, 기능 블록 2420에 진행한다. 기능 블록 2420에서는 현재의 꾹쳐를 디코딩하여, 기능 블록 2425에 진행한다. 기능 블록 2425에서는, 현재의 꾹쳐를 디코딩된 꾹쳐 버퍼에 삽입하고, 결정 블록 2430에 진행한다. 결정 블록 2430에서는 모든 꾹쳐들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 종료 블록 2499에 진행한다. 그렇지 않다면, 기능 블록 2410에 복귀한다.

시간적인 DIRECT 모드

[0175] 전술한 바와 같이, 시간적인 DIRECT 모드에서는 주어진 매크로블록에 대한 모션 벡터를 유도하기 위하여 꾹쳐 오더 카운트 정보를 사용한다. frame_num 및/또는 꾹쳐 오더 카운트 값을 분리하고, 멀티뷰 비디오 콘텐츠의 각각의 뷰에 대하여 view_id를 도입하고, 디코더 꾹쳐 버퍼와 레퍼런스 리스트들 내에 크로스-뷰 꾹쳐들을 위치시키도록 하기 때문에, 일 실시예에 있어서, 크로스-뷰 꾹쳐가 현재의 뷰와는 상이한 뷰로부터의 꾹쳐를 지칭하도록 올바르게 유도식을 다루기 위해서 이 모드를 또한 상술한다.

[0176] 시간적인 DIRECT 모드에 있어서, 이하의 예시적인 경우가 있다:

[0177] (1) 레퍼런스 리스트 1 내의 꾹쳐 및 레퍼런스 리스트 0 내의 꾹쳐가 상이한 POC 및 동일한 view_id를 갖는다;

[0178] (2) 레퍼런스 리스트 1 내의 꾹쳐 및 레퍼런스 리스트 0 내의 꾹쳐가 상이한 POC 및 상이한 view_id를 갖는다;

[0179] (3) 레퍼런스 리스트 1 내의 꾹쳐 및 레퍼런스 리스트 0 내의 꾹쳐가 동일한 POC 및 상이한 view_id를 갖는다;

[0180] (4) 레퍼런스 리스트 1 내의 꾹쳐 및 레퍼런스 리스트 0 내의 꾹쳐가 동일한 POC 및 동일한 view_id를 갖는다.

[0181] 시간적인 DIRECT 모드에서 모션 벡터를 얻는 일 실시예에서는, 비트스트림 내에 존재하는 view_id 정보를 단지 무시하는 기존의 MPEG-4 AVC 방법을 사용하는 것이다. 또 다른 실시예에 있어서, 꾹쳐 오더 카운트 정보와 함께 view_id 정보를 고려하여 시간적인 DIRECT 모드를 재정의한다.

내재 가중 예측(Implicit Weighted Prediction)

[0183] 시간적인 DIRECT 모드와 마찬가지로, 내재 가중치 예측(전술한 바와 같음)에서는 꾹쳐 오더 카운트 값을 사용하여 레퍼런스 꾹쳐들에 적용될 가중치들을 판정한다. 그 결과, 일 실시예에 있어서, 시간적인 DIRECT 모드에 적용하는 모든 변화들은 간접적으로 내재 가중 예측 모드를 확립한다. 또 다른 실시예에 있어서, 내재 가중치 예측 모드에서 가중치를 얻는 방법은 꾹쳐 오더 카운트 정보와 함께 view_id 정보를 고려하여 재정의될 수 있다. 예를 들어, 전술한 바와 같이 view_id 정보 및 뷰의 갯수를 고려하여 꾹쳐 오더 카운트를 계산할 수 있으며, 그

후, 내재 가중 예측을 수행하기 위하여 요구된 값들을 얻기 위하여 꽂쳐 오더 카운트들 간의 차이를 구할 수 있다.

[0184] 도 11을 참조하면, 시간적인 DIRECT 모드 및 내재 가중 예측을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례가 참조 번호 1100에 개략적으로 나타내어져 있다.

[0185] 본 방법(1100)에서는 시작 블록 1105에서 기능 블록 1110에 진행한다. 기능 블록 1110에서는 인코더 형상 파일을 판독하고, 기능 블록 1115에 진행한다. 기능 블록 1115에서는, 뷰의 갯수가 변수 N과 동일하도록 하여, 변수 i (뷰 넘버 인덱스) 및 j (꽉쳐 넘버 인덱스)를 모두 0으로 설정하고, 결정 블록 1120에 진행한다. 결정 블록 1120에서는, i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 1125에 진행한다. 그렇지 않다면, 종료 블록 1199에 진행한다.

[0186] 기능 블록 1125에서는, j가 뷰 i내의 꽂쳐들의 수보다 작은지 여부를 판정한다. 만일 그렇다면, 결정 블록 1132에 진행한다. 그렇지 않다면, 기능 블록 1145에 진행한다.

[0187] 기능 블록 1132에서는, 현재 슬라이스에 대하여 가중 예측이 되도록 되는지 여부를 판정한다. 만일 그렇다면, 기능 블록 1134에 진행한다. 그렇지 않다면, 기능 블록 1136에 진행한다.

[0188] 기능 블록 1134에서는, 가중 예측을 위해 view_id 정보를 무시하고, 기능 블록 1136에 진행한다.

[0189] 기능 블록 1136에서는, 현재의 매크로블록을 인코딩하기 시작하고, 결정 블록 1138에 진행한다. 결정 블록 1138에서는, 매크로블록에 대하여 DIRECT 모드를 선택할지 여부를 판정한다. 만일 그렇다면, 기능 블록 1142에 진행한다. 그렇지 않다면, 기능 블록 1152에 진행한다.

[0190] 기능 블록 1142에서는, DIRECT 모드에 대하여 view_id를 무시하고 기능 블록 1152에 진행한다.

[0191] 기능 블록 1152에서는, 현재의 매크로블록을 인코딩하고, 결정 블록 1154에 진행한다. 결정 블록 1154에서는 모든 매크로블록이 인코딩되었는지 여부를 판정한다. 만일 그렇다면, 기능 블록 1156에 진행한다. 그렇지 않다면, 기능 블록 1136에 진행한다.

[0192] 기능 블록 1156에서는 변수 j를 증분시키고, 기능 블록 1140에 진행한다. 기능 블록 1140에서는, fram_num 및 꽂쳐 오더 카운트를 증분시키고, 결정 블록 1125에 복귀한다.

[0193] 기능 블록 1145에서는 i를 증분시키고, fram_num 및 꽂쳐 오더 카운트를 리셋하고, 결정 블록 1120에 복귀한다.

[0194] 도 12를 참조하면, 시간적인 DIRECT 모드 및 내재 가중 예측을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 또 다른 일례가 참조 번호 1200에 개략적으로 나타내어져 있다.

[0195] 본 방법(1200)에서는, 시작 블록 1205에서 기능 블록 1210에 진행한다. 기능 블록 1210에서는, 인코더 형상 파일을 판독하고, 기능 블록 1215에 진행한다. 기능 블록 1215에서는, 뷰의 갯수가 변수 N과 동일하도록 하여, 변수 i (뷰 넘버 인덱스) 및 j (꽉쳐 넘버 인덱스)를 모두 0으로 설정하고, 결정 블록 1220에 진행한다. 결정 블록 1220에서는, i가 N보다 작은지 여부를 판정한다. 만일 그렇다면, 기능 블록 1225에 진행한다. 그렇지 않다면, 종료 블록 1299에 진행한다.

[0196] 기능 블록 1225에서는, j가 뷰 i내의 꽂쳐들의 수보다 작은지 여부를 판정한다. 만일 그렇다면, 결정 블록 1232에 진행한다. 그렇지 않다면, 기능 블록 1245에 진행한다.

[0197] 결정 블록 1232에서는, 현재 슬라이스에 대하여 가중 예측이 되도록 되는지 여부를 판정한다. 만일 그렇다면, 기능 블록 1234에 진행한다. 그렇지 않다면, 기능 블록 1236에 진행한다.

[0198] 기능 블록 1234에서는, 가중 예측을 위해 view_id 정보를 무시하고, 기능 블록 1236에 진행한다.

[0199] 기능 블록 1236에서는, 현재의 매크로블록을 인코딩하기 시작하고, 결정 블록 1238에 진행한다. 결정 블록 1238에서는, 매크로블록에 대하여 DIRECT 모드를 선택할지 여부를 판정한다. 만일 그렇다면, 기능 블록 1242에 진행한다. 그렇지 않다면, 기능 블록 1252에 진행한다.

[0200] 기능 블록 1242에서는, DIRECT 모드에 대하여 view_id를 고려하고, 기능 블록 1252에 진행한다.

[0201] 기능 블록 1252에서는, 현재의 매크로블록을 인코딩하고, 결정 블록 1254에 진행한다. 결정 블록 1254에서는 모든 매크로블록이 인코딩되었는지 여부를 판정한다. 만일 그렇다면, 기능 블록 1256에 진행한다. 그렇지 않다면, 기능 블록 1236에 진행한다.

- [0202] 기능 블록 1256에서는 변수 *j*를 증분시키고, 기능 블록 1240에 진행한다. 기능 블록 1240에서는, *fram_num* 및 꽂쳐 오더 카운트를 증분시키고, 결정 블록 1225에 복귀한다.
- [0203] 기능 블록 1245에서는 *i*를 증분시키고, *fram_num* 및 꽂쳐 오더 카운트를 리셋하고, 결정 블록 1220에 복귀한다.
- [0204] 도 13을 참조하면, 변형된 디코디드 레퍼런스 꽂쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례가 참조 번호 1300에 개략적으로 나타내어져 있다.
- [0205] 본 방법(1300)에서는, 개시 블록 1305에서 기능 블록 1310에 진행한다. 기능 블록 1310에서는, 비트스트림, *view_id*, *frame_num*, 및 꽂쳐 오더 카운트(POC)를 파싱하고, 기능 블록 1315에 진행한다. 기능 블록 1315에서는, 매크로블록 모드, 모션 벡터, *ref_idx*를 파싱하여, 결정 블록 1320에 진행한다. 결정 블록 1320에서는, 꽂쳐에 대하여 가중 예측이 되도록 되는지 여부를 판정한다. 만일 그렇다면, 기능 블록 1325에 진행한다. 그렇지 않다면, 결정 블록 1330에 진행한다.
- [0206] 기능 블록 1325에서는, 가중 예측을 위해 *view_id* 정보를 무시하고, 결정 블록 1330에 진행한다.
- [0207] 결정 블록 1330에서는, 매크로블록이 DIRECT 모드 매크로블록인지 여부를 판정한다. 만일 그렇다면, 기능 블록 1355에 진행한다. 그렇지 않다면, 기능 블록 1335에 진행한다.
- [0208] 기능 블록 1355에서는, DIRECT 모드에 있어 *view_id* 정보를 무시하고, 기능 블록 1335에 진행한다.
- [0209] 기능 블록 1335에서는, 현재의 매크로블록을 디코딩하고, 결정 블록 1340에 진행한다. 결정 블록 1340에서는, 모든 매크로블록들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 기능 블록 1345에 진행한다. 그렇지 않다면, 기능 블록 1315에 복귀한다.
- [0210] 기능 블록 1345에서는 현재의 꽂쳐를 디코딩된 꽂쳐 버퍼에 삽입하고, 결정 블록 1350에 진행한다. 결정 블록 1350에서는, 모든 꽂쳐들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 종료 블록 1399에 진행한다. 그렇지 않다면, 기능 블록 1310에 진행한다.
- [0211] 도 14를 참조하면, 변형된 디코디드 레퍼런스 꽂쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 또 다른 일례가 참조 번호 1400에 개략적으로 나타내어져 있다.
- [0212] 본 방법(1400)에서는, 시작 블록 1405에서 기능 블록 1410에 진행한다. 기능 블록 1410에서는, 비트스트림, *view_id*, *frame_num*, 및 꽂쳐 오더 카운트(POC)를 파싱하고, 기능 블록 1415에 진행한다. 기능 블록 1415에서는, 매크로블록 모드, 모션 벡터, *ref_idx*를 파싱하고, 결정 블록 1420에 진행한다. 결정 블록 1420에서는, 꽂쳐에 대하여 가중 예측이 되도록 되는지 여부를 판정한다. 만일 그렇다면, 기능 블록 1425에 진행한다. 그렇지 않다면, 결정 블록 1430에 진행한다.
- [0213] 기능 블록 1425에서는, 가중 예측을 위해 *view_id* 정보를 무시하고, 결정 블록 1430에 진행한다.
- [0214] 결정 블록 1430에서는, 매크로블록이 DIRECT 모드 매크로블록인지 여부를 판정한다. 만일 그렇다면, 기능 블록 1455에 진행한다. 그렇지 않다면, 기능 블록 1435에 진행한다.
- [0215] 기능 블록 1455에서는, DIRECT 모드를 위해 *view_id* 정보를 고려하고, 기능 블록 1435에 진행한다.
- [0216] 기능 블록 1435에서는, 현재의 매크로블록을 디코딩하고, 결정 블록 1440에 진행한다. 결정 블록 1440에서는, 모든 매크로블록들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 기능 블록 1445에 진행한다. 그렇지 않다면, 기능 블록 1415에 복귀한다.
- [0217] 기능 블록 1445에서는, 현재의 꽂쳐를 디코딩된 꽂쳐 버퍼에 삽입하고, 결정 블록 1450에 진행한다. 결정 블록 1450에서는, 모든 꽂쳐들이 디코딩되었는지 여부를 판정한다. 만일 그렇다면, 종료 블록 1499에 진행한다. 그렇지 않다면, 기능 블록 1410에 복귀한다.
- [0218] **MVC의 병렬 코딩(Parallel Coding of MVC)**
- [0219] 멀티뷰 비디오 콘텐츠 시퀀스들의 처리에 관여되는 데이터 량으로 인하여, 많은 응용 예, 즉히, 실시간의 제약이 있는 경우에, 멀티뷰 비디오 코딩에 있어서의 병렬 인코딩/디코딩의 지원이 매우 중요하다. 현재의 MPEG-4 AVC에 준거한 멀티뷰 비디오 코딩의 구현 예에 있어서, 크로스-뷰 예측이 가능하지만, 크로스-뷰 레퍼런스와 시간적인 레퍼런스를 구분할 설비는 존재하지 않는다. *view_id* 서포트를 멀티뷰 비디오 코딩 인코더 및/또는 디코더에 추가하고, 본 명세서에서 제안된 바와 같은 디코딩된 레퍼런스 꽂쳐 관리의 구성 및 레퍼런스 리스트 구성에

view_id들을 포함시킴으로써, 병렬 처리 엔진들 간의 데이터 의존성이 명확하게 정의되며, 이는 MVC 코덱의 병렬 구현을 용이하게 한다.

[0220] 이하, 본 발명의 많은 부수적인 장점/특징들 중 일부에 대하여 설명하며, 그중 일부는 상기 언급한 것들이다. 예를 들어, 하나의 장점/특징은 멀티뷰 비디오 콘텐츠의 적어도 2개의 뷰 중 적어도 하나에 대응하는 적어도 하나의 핵심을 인코딩하여 그 최종 비트스트림을 형성하는 인코더를 포함하는 장치이다. 그 최종 비트스트림에 있어서, 상기 적어도 하나의 핵심에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나는 적어도 하나의 핵심이 대응하는 적어도 하나의 뷰로부터 분리된다.

[0221] 또 다른 장점/특징은, 전술한 바와 같이 인코더를 갖는 장치이며, 여기서, 인코더는 ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항(International Organization for Standardization/International Electrotechnical Commission Moving Picture Experts Group-4 Part 10 Advanced Video Coding standard/International Telecommunication Union, Telecommunication Sector H.264 recommendation)에 대응하는 적어도 하나의 기준 신택스 구성요소(frame_num 및 pic_order_cnt_lsb)를 사용하여 상기 적어도 하나의 핵심에 대한 코딩 오더 정보와 출력 오더 정보 중 적어도 하나를 분리시킨다.

[0222] 또 다른 장점/특징은, 전술한 바와 같은 인코더를 갖는 장치이며, 여기서, 인코더는 뷰 식별자를 이용하여 적어도 하나의 핵심에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시킨다.

[0223] 또 다른 장점/특징은, 전술한 바와 같이 뷰 식별자를 이용하여 적어도 하나의 핵심에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 인코더를 갖는 장치이며, 여기서, 뷰 식별자는 최종 비트스트림 내에서 슬라이스 레벨에 존재한다.

[0224] 또 다른 장점/특징은, 전술한 바와 같이 뷰 식별자를 이용하여 상기 적어도 하나의 핵심에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 인코더를 갖는 장치이며, 여기서, 뷰 식별자는 최종 비트스트림 내에서 매크로블록 레벨보다 높은 레벨에 존재한다.

[0225] 또한, 또 다른 장점/특징은, 전술한 바와 같이 더 높은 레벨에 존재하는 뷰 식별자를 이용하여 상기 적어도 하나의 핵심에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 인코더를 갖는 장치이며, 여기서, 인코더는 디코디드 레퍼런스 핵심 마킹 프로세스에서 사용하기 위하여 최종 비트스트림 내에 뷰 식별자를 포함시킨다.

[0226] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 디코디드 레퍼런스 핵심 마킹 프로세스에 사용하기 위하여 최종 비트스트림 내에 뷰 식별자를 포함시키는 인코더를 갖는 장치이며, 여기서, 인코더는 적어도 2개의 뷰 중에서 어디에 디코디드 레퍼런스 핵심 마킹 프로세스에 의해 마킹되어야 하는 특정 핵심이 속하는지를 나타내기 위하여 최종 비트스트림 내에 뷰 식별자를 포함시킨다.

[0227] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자를 이용하여 적어도 하나의 핵심에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 인코더를 갖는 장치이며, 여기서, 인코더는 재정의된 디코디드 레퍼런스 핵심 마킹 프로세스에서 뷰 식별자의 사용을 지원하도록, ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항(International Organization for Standardization/International Electrotechnical Commission Moving Picture Experts Group-4 Part 10 Advanced Video Coding standard/International Telecommunication Union, Telecommunication Sector H.264 recommendation)에 대응하는 재정의된 디코디드 레퍼런스 핵심 마킹 프로세스에서 사용하기 위하여 재정의된 적어도 하나의 기준 신택스 구성요소의 시멘틱을 갖는, 기준 신택스 구성요소(no_output_of_prior_pics_flag, long_term_reference_flag, adaptive_ref_pic_marking_mode_flag, memory_management_control_operation, difference_of_pic_nums_minus1, long_term_pic_num, long_term_frame_idx, max_long_term_frame_idx_plus1) 중 적어도 하나를 사용한다.

[0228] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자 및 적어도 하나의 기준 신택스 구성요소를 사용하는 인코더를 갖는 장치이며, 여기서, 재정의된 디코디드 레퍼런스 핵심 마킹 프로세스에서는, 현재 디코딩되는 핵심과 동일한 뷰 식별자를 갖는 핵심들만이 마킹된다.

[0229] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자와 적어도 하나의 기준 신택스 구성요소를 사용하는 인코더를 갖는 장치이며, 여기서, 슬라이딩 윈도우 디코딩된 레퍼런스 핵심 마킹 프로세스 및 적응성 메모리 컨트롤 디코딩된 레퍼런스 핵심 마킹 프로세스 중 적어도 하나가 적용된다.

- [0230] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자 및 적어도 하나의 기존 신팩스 구성요소를 사용하는 인코더를 갖는 장치이며, 여기서, 재정의된 디코디드 레퍼런스 꾹쳐 마킹 프로세스에 있어서, 이전에 사용되지 않은 신팩스 구성요소(difference_of_view_ids_minus1)를 사용하여 상기 적어도 하나의 꾹쳐와는 상이한 뷰 식별자를 갖는 꾹쳐들이 마킹된다.
- [0231] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자를 이용하여 상기 적어도 하나의 꾹쳐에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 인코더를 갖는 장치이며, 여기서, 인코더는 디폴트 레퍼런스 꾹쳐 리스트 구성에 대한 최종 비트스트림 내에 뷰 식별자를 포함시킨다.
- [0232] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 디폴트 레퍼런스 꾹쳐 리스트 구성에 대한 최종 비트스트림 내에 뷰 식별자를 포함시키는 인코더를 갖는 장치이며, 여기서, 인터-뷰 레퍼런스 꾹쳐들은, 적어도 하나의 기존 신팩스 구성요소(frame_num 및 pic_order_cnt_lsb), 기존의 신팩스, 및 뷰 식별자로부터 추가의 지원을 갖는 레퍼런스 꾹쳐 리스트 구성에 대한 기존의 디코딩 프로세스에 따라서, 레퍼런스 꾹쳐 리스트 구성에 대응하는 디폴트 레퍼런스 꾹쳐 리스트 작성 프로세스를 위한 레퍼런스 리스트에 추가되지 않도록 금지되며, 기존의 신팩스, 기존의 시맨틱, 및 기존의 디코딩 프로세스는 ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항에 대응한다.
- [0233] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 디폴트 레퍼런스 꾹쳐 리스트 구성을 위한 최종 비트스트림 내에서 뷰 식별자를 포함하는 인코더를 갖는 장치이며, 뷰 식별자로부터 추가의 지원을 갖는 디폴트 레퍼런스 꾹쳐 리스트 구성에 대한 적어도 하나의 기존 신팩스 구성요소 (frame_num 및 pic_order_cnt_lsb)에 따라서, 인터-뷰 레퍼런스 꾹쳐들만이 레퍼런스 꾹쳐 리스트 구성에 대응하는 디폴트 레퍼런스 꾹쳐 리스트 작성 프로세스를 위한 레퍼런스 리스트에 추가된다.
- [0234] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 디폴트 레퍼런스 꾹쳐 리스트 작성 프로세스를 위한 레퍼런스 리스트에 인터-뷰 레퍼런스 꾹쳐들만이 추가되는 인코더를 갖는 장치이며, 여기서, 인터-뷰 레퍼런스 꾹쳐들은 시간적인 레퍼런스 이후에 추가된다.
- [0235] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자를 사용하여 상기 적어도 하나의 꾹쳐에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 인코더를 갖는 장치이며, 여기서, 인코더는 재정의된 레퍼런스 꾹쳐 리스트 리오더링 프로세스에서 뷰 식별자의 사용을 지원하기 위하여, ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항에 대응하는 재정의된 레퍼런스 꾹쳐 리스트 리오더링 프로세스에서 사용하기 위하여 재정의되는, 기존의 신팩스 구성요소(ref_pic_list_reordering_flag_I0, reordering_of_pic_nums_idc, abs_diff_pic_num_minus1, long_term_pic_num, ref_pic_list_reordering_flag_I1, reordering_of_pic_nums_idc, abs_diff_pic_num_minus1, long_term_pic_num) 중 적어도 하나를 사용한다.
- [0236] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자 및 적어도 하나의 기존 신팩스 구성요소를 사용하는 인코더를 갖는 장치이며, 여기서, 재정의된 레퍼런스 꾹쳐 리스트 리오더링 프로세스에 있어서, 현재 디코딩되는 꾹쳐와 동일한 뷰 식별자를 갖는 꾹쳐들만이 리오더링된다.
- [0237] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 현재 디코딩되는 꾹쳐와 동일한 뷰 식별자를 갖는 꾹쳐들만이 리오더링되는 인코더를 갖는 장치이며, 여기서, 뷰 식별자는 적어도 2개의 뷰들 중에서 어느 것이 대응하는 레퍼런스 꾹쳐 리스트 내에서 현재의 인덱스로 이동되는 특정의 꾹쳐인지를 나타낸다.
- [0238] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 현재 디코딩되는 꾹쳐와 동일한 뷰 식별자를 갖는 꾹쳐들만이 리오더링되는 인코더를 갖는 장치이며, 여기서, 뷰 식별자는 리오더링되는 레퍼런스 꾹쳐의 뷰 식별자가 적어도 하나의 꾹쳐와는 상이한 경우에만 요구된다.
- [0239] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자를 사용하여 적어도 하나의 꾹쳐에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 인코더를 갖는 장치이며, 여기서, 인코더는 시간적인 DIRECT 모드에 대하여 재정의되는 기존의 신팩스 구성요소(pic_order_cnt_lsb)을 사용하며, 기존의 신팩스는 시간적인 DIRECT 모드에서 뷰 식별자의 사용을 지원하기 위하여 ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항에 대응한다.
- [0240] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자 및 기존의 신팩스 구성요소를 사용한 인코더를 갖는 장치이며, 여기서, 시간적인 DIRECT 모드는 꾹쳐 오더 카운트 값과 뷰 식별자 중 적어도 하나에 기초하여 유도된다.

- [0241] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자를 사용하여 적어도 하나의 픽쳐에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 인코더를 갖는 장치이며, 여기서, 인코더는 기존의 신택스 구성요소(pic_order_cnt_lsb), 기존의 시맨틱, 및 시간적인 DIRECT 모드에 대한 기존의 디코딩 프로세스를 사용하며, 기존의 신택스, 기존의 시맨틱, 및 기존의 디코딩 프로세스는 ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항에 대응한다.
- [0242] 또한, 또 다른 장점/특징은, 전술한 바와 같이 뷰 식별자를 이용하여 적어도 하나의 픽쳐에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 인코더를 갖는 장치이며, 여기서, 인코더는 내재 가중 예측에 대하여 재정의되는 기존의 신택스 구성요소(pic_order_cnt_lsb)을 사용하며, 기존의 신택스는 내재 가중 예측에서 뷰 식별자의 사용을 지원하기 위하여 ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항에 대응한다.
- [0243] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자 및 적어도 하나의 기존 신택스 구성요소를 사용하는 인코더를 갖는 장치이며, 여기서, 내재 가중 예측은 픽쳐 오더 카운트 및 뷰 식별자 중 적어도 하나에 기초하여 유도된다.
- [0244] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 뷰 식별자를 사용하여 적어도 하나의 픽쳐에 대한 코딩 오더 정보 및 출력 오더 정보 중 적어도 하나를 분리시키는 인코더를 갖는 장치이며, 여기서, 인코더는 기존의 신택스 구성요소(pic_order_cnt_lsb), 기존의 시맨틱, 및 내재 가중 예측에 대한 기존의 디코딩 프로세스를 사용하며, 기존의 신택스, 기존의 시맨틱, 및 기존의 디코딩 프로세스는 ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항에 대응한다.
- [0245] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 인코더를 갖는 장치로서, 여기서, 인코더는 적어도 2개의 뷰 중 상이한 것들의 병렬 인코딩에 있어서 인터-뷰 의존성을 명시하기 위하여 특정 픽쳐에 대응하는 적어도 2개의 뷰들 중 특정의 하나를 사용한다.
- [0246] 또한, 또 다른 장점/특징은, 멀티뷰 비디오 콘텐츠에 대응하는 적어도 2개의 뷰 중 적어도 하나를 인코딩하기 위한 인코더를 포함하는 장치이다. 인코더는 ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항에 대응하는 디폴트 레퍼런스 픽쳐 리스트 구성 프로세스 및 레퍼런스 픽쳐 리스트 리오더링에 있어서 재정의된 변수들을 사용하여 적어도 2개의 뷰들 중 적어도 하나를 인코딩한다.
- [0247] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 인코더를 포함하는 장치로서, 여기서, 뷰의 갯수 및 뷰 식별자 정보 중 적어도 하나가 변수를 재정의하기 위하여 사용된다.
- [0248] 또한, 또 다른 장점/특징은, 전술한 바와 같이, 인코더를 포함하는 장치로서, 여기서, GOP 길이 및 뷰 식별 정보 중 적어도 하나가 변수를 재정의하기 위하여 사용된다.
- [0249] 또 다른 장점/특징은 멀티뷰 비디오 콘텐츠에 대응하는 적어도 2개의 뷰들 중 적어도 하나를 인코딩하기 위한 인코더를 포함하는 장치이다. 인코더는 ISO/IEC MPEG-4 Part 10 AVC 표준/ITU-T H.264 권고사항의 디코디드 레퍼런스 픽쳐 마킹 프로세스에 있어서 재정의된 변수들을 이용하여 적어도 2개의 뷰 중 적어도 하나를 인코딩한다.
- [0250] 또한, 또 다른 장점/특징은 전술한 바와 같은 인코더를 포함하는 장치이며, 여기서, 뷰의 갯수와 뷰의 식별자 정보 중 적어도 하나가 변수를 재정의하기 위하여 사용된다.
- [0251] 또한, 또 다른 장점/특징은 전술한 바와 같은 인코더를 포함하는 장치이며, 여기서, GOP 길이 및 뷰 식별자 정보 중 적어도 하나가 변수를 재정의하기 위하여 사용된다.
- [0252] 다양한 본 발명의 양태들에 있어서, 특정의 신택스 명칭, 특히, 이전에 사용되지 않은 신택스 명칭을 선택하는 것은, 예시와 명확화를 위한 것이며, 따라서, 본 명세서에 주어진 본 발명의 교시에 있어서, 본 발명의 개념을 유지하면서, 본 명세서에 제공된 신택스 명칭을 대신하여 및/또는 이에 추가하여 다른 명칭 및/또는 특징 등이 사용될 수 있다는 것을 이해할 수 있을 것이다.
- [0253] 이러한 및 기타의 본 발명의 특징 및 장점들은 본 명세서의 교시에 기초하여 당업자라면 용이하게 확인될 수 있을 것이다. 본 발명의 교시는 하드웨어, 소프트웨어, 펌웨어, 특수 목적 프로세서, 또는 그 조합의 다양한 형태로 구현될 수 있다는 것을 이해할 수 있을 것이다.
- [0254] 보다 바람직하게는, 본 발명의 교시는 하드웨어와 소프트웨어의 조합으로 구현된다. 또한, 소프트웨어는 프로

그램 저장부 상에 실체적으로 구현되는 어플리케이션 프로그램으로서 구현될 수 있다. 어플리케이션 프로그램은 임의의 적합한 아키텍처를 구비한 머신에 탑재되어 이에 의해 실행될 수 있다. 바람직하게는, 머신은 하나 이상의 CPU, RAM 및 I/O 인터페이스 등의 하드웨어를 갖는 컴퓨터 플랫폼 상에 구현된다. 컴퓨터 플랫폼은 또한 운영 체제 및 마이크로인스트럭션 코드를 포함할 수 있다. 본 명세서의 기재된 다양한 프로세스들과 함수들은, CPU에 의해 실행될 수 있는 마이크로인스트럭션 코드의 일부 또는 어플리케이션 프로그램의 일부, 또는 그 조합일 수 있다. 또한, 추가의 데이터 저장 장치 및 인쇄 장치 등의 각종 다른 주변 장치들이 컴퓨터 플랫폼에 연결될 수 있다.

[0255] 구성 시스템 성분들 및 첨부 도면에 도시된 방법들 중 일부는 소프트웨어에 구현되는 것이 바람직하기 때문에, 실제의 시스템 성분들 또는 프로세스 기능 블록들 사이의 연결은 본 발명이 프로그래밍되는 방식에 따라서 상이 할 수 있다. 본 명세서의 교시에 따라서, 당업자라면, 이러한 본 발명의 구현에 또는 구성예들 및 그와 유사한 것들을 상정할 수 있을 것이다.

[0256] 본 명세서에서는 첨부 도면을 참조하여 예시적인 실시예들을 설명하였지만, 본 발명은 그러한 구체적인 실시예들에 한하지 않고, 본 발명의 범주 또는 개념을 일탈하지 않고서, 당업자에 의해 다양한 변경예 및 변형예들이 달성될 수 있을 것이다. 이러한 모든 변경예 및 변경예들은 첨부된 청구항들에 명기된 바와 같이 본 발명의 범주 내에 포함되는 것으로 간주되어야 한다.

도면의 간단한 설명

[0026] 본 발명의 원리는 이하의 예시적인 도면에 따라서 더 잘 이해될 수 있다.

도 1은 B 슬라이스 코딩에 있어서 시간적인 DIRECT 예측을 나타낸 도면이다.

도 2a는 본 발명의 일 실시예에 따라서 본 발명의 원리가 적용될 수 있는 MVC(Multi-view Video Coding) 인코더의 일례를 나타낸 블록도이다.

도 2b는 일 실시예에 따라서 본 발명의 원리가 적용될 수 있는 MVC(Multi-view Video Coding) 디코더의 일례를 나타낸 블록도이다.

도 3은 본 발명의 일 실시예에 따라서 변형된 디코디드 레퍼런스 꾹쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례를 나타낸 흐름도이다.

도 4는 본 발명의 일 실시예에 따라서 변형된 디코디드 레퍼런스 꾹쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례를 나타낸 흐름도이다.

도 5는 본 발명의 일 실시예에 따라서 변형된 레퍼런스 꾹쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례를 나타낸 흐름도이다.

도 6은 본 발명의 일 실시예에 따라서 변형된 레퍼런스 꾹쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 또 다른 예를 나타낸 흐름도이다.

도 7은 본 발명의 일 실시예에 따라서 변형된 레퍼런스 꾹쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 또 다른 예를 나타낸 흐름도이다.

도 8은 본 발명의 일 실시예에 따라서 변형된 레퍼런스 꾹쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례를 나타낸 흐름도이다.

도 9는 본 발명의 일 실시예에 따라서 변형된 레퍼런스 꾹쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 또 다른 예를 나타낸 흐름도이다.

도 10은 본 발명의 일 실시예에 따라서 변형된 레퍼런스 꾹쳐 리스트 구성을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 또 다른 예를 나타낸 흐름도이다.

도 11은 본 발명의 일 실시예에 따라서 시간적인 DIRECT 모드 및 내재 가중 예측을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례를 나타낸 흐름도이다.

도 12는 본 발명의 일 실시예에 따라서 시간적인 DIRECT 모드 및 내재 가중 예측을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 또 다른 일례를 나타낸 흐름도이다.

도 13은 본 발명의 일 실시예에 따라서 변형된 디코디드 레퍼런스 꾹쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠

를 디코딩하는 방법의 일례를 나타낸 흐름도이다.

[0041] 도 14는 본 발명의 일 실시예에 따라서 변형된 디코디드 레퍼런스 픽쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 또 다른 일례를 나타낸 흐름도이다.

[0042] 도 15는 본 발명의 일 실시예에 따라서 변형된 디코디드 레퍼런스 픽쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례를 나타낸 흐름도이다.

[0043] 도 16은 본 발명의 일 실시예에 따라서 변형된 디코디드 레퍼런스 픽쳐 마킹을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례를 나타낸 흐름도이다.

[0044] 도 17은 본 발명의 일 실시예에 따라서 변형된 레퍼런스 픽쳐 리스트 구성과 프레임 넘버 계산을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례를 나타낸 흐름도이다.

[0045] 도 18은 본 발명의 일 실시예에 따라서 변형된 레퍼런스 픽쳐 리스트 구성과 프레임 넘버 계산을 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 또 다른 일례를 나타낸 흐름도이다.

[0046] 도 19는 본 발명의 일 실시예에 따라서 변형된 레퍼런스 픽쳐 리스트 구성과 프레임 넘버 계산을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례를 나타낸 흐름도이다.

[0047] 도 20은 본 발명의 일 실시예에 따라서 변형된 레퍼런스 픽쳐 리스트 구성과 프레임 넘버 계산을 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 또 다른 일례를 나타낸 흐름도이다.

[0048] 도 21은 본 발명의 일 실시예에 따라서 RPLR(Reference Picture List Reordering) 커맨드를 이용한 변형된 레퍼런스 픽쳐 리스트 초기화를 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 일례를 나타낸 흐름도이다.

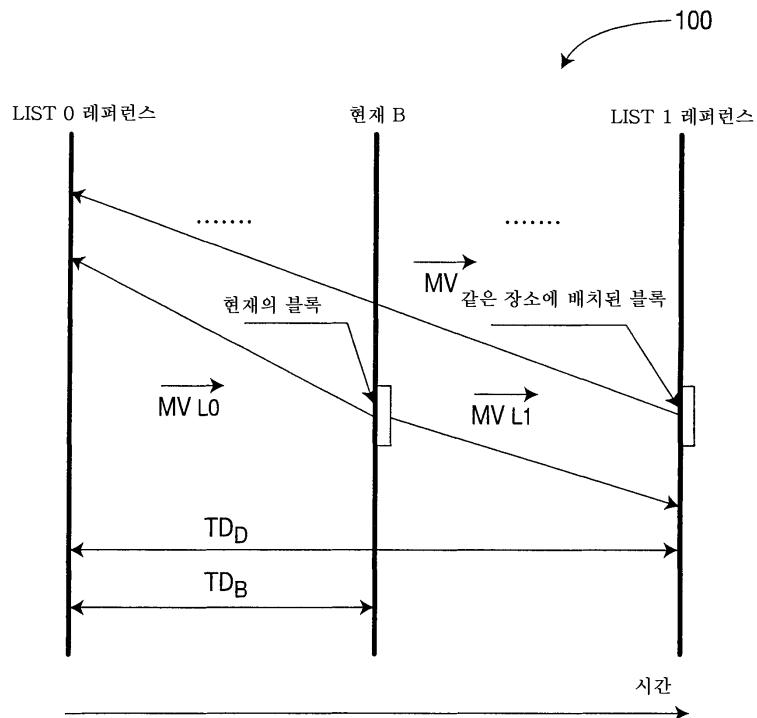
[0049] 도 22는 본 발명의 일 실시예에 따라서 RPLR 커맨드를 이용한 변형된 레퍼런스 픽쳐 리스트 초기화를 이용하여 멀티뷰 비디오 콘텐츠를 인코딩하는 방법의 또 다른 일례를 나타낸 흐름도이다.

[0050] 도 23은 본 발명의 일 실시예에 따라서 RPLR 커맨드를 이용한 변형된 레퍼런스 픽쳐 리스트 초기화를 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 일례를 나타낸 흐름도이다.

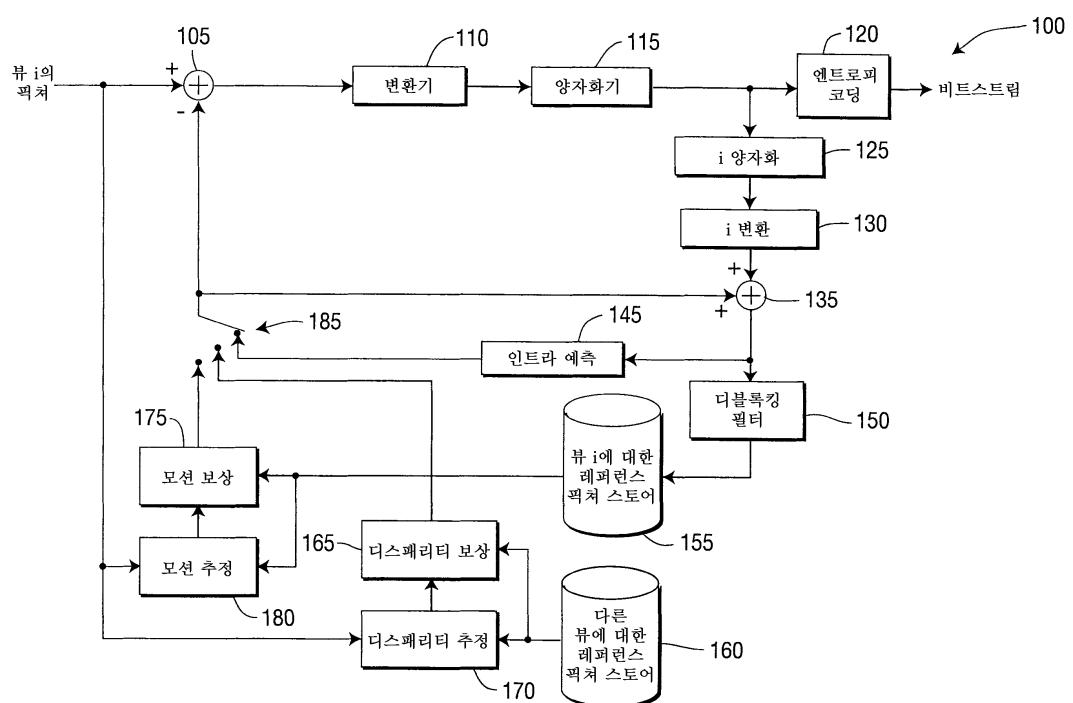
[0051] 도 24는 본 발명의 일 실시예에 따라서 RPLR 커맨드를 이용한 변형된 레퍼런스 픽쳐 리스트 초기화를 이용하여 멀티뷰 비디오 콘텐츠를 디코딩하는 방법의 또 다른 일례를 나타낸 흐름도이다.

도면

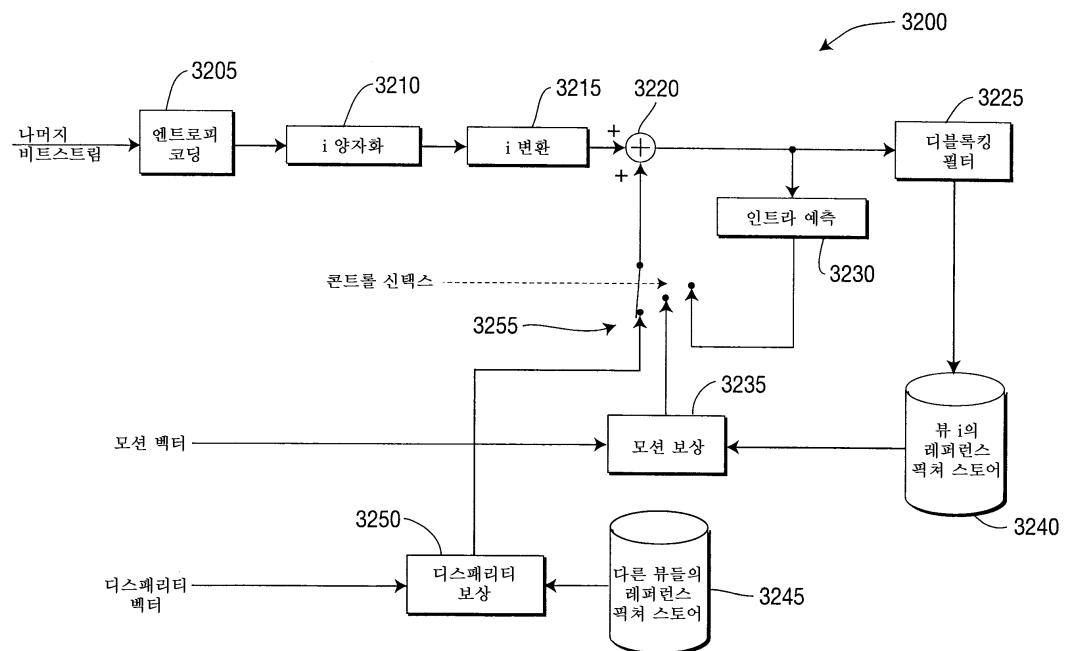
도면1



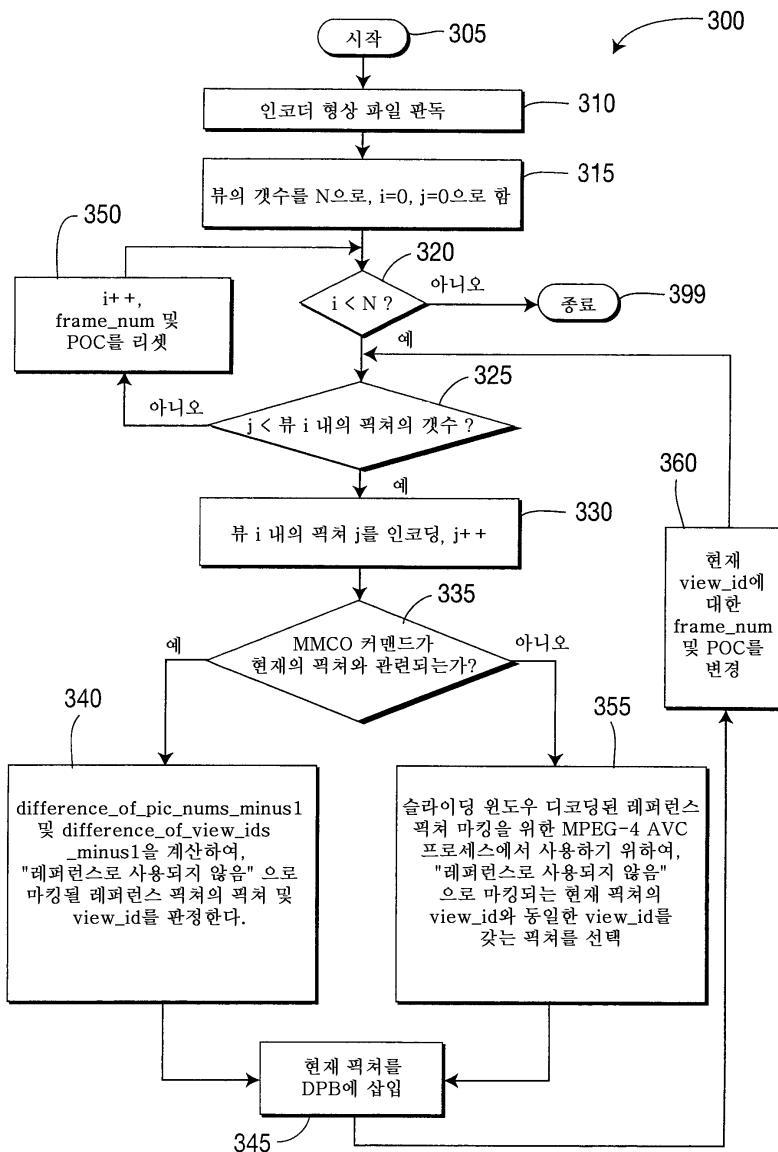
도면2a



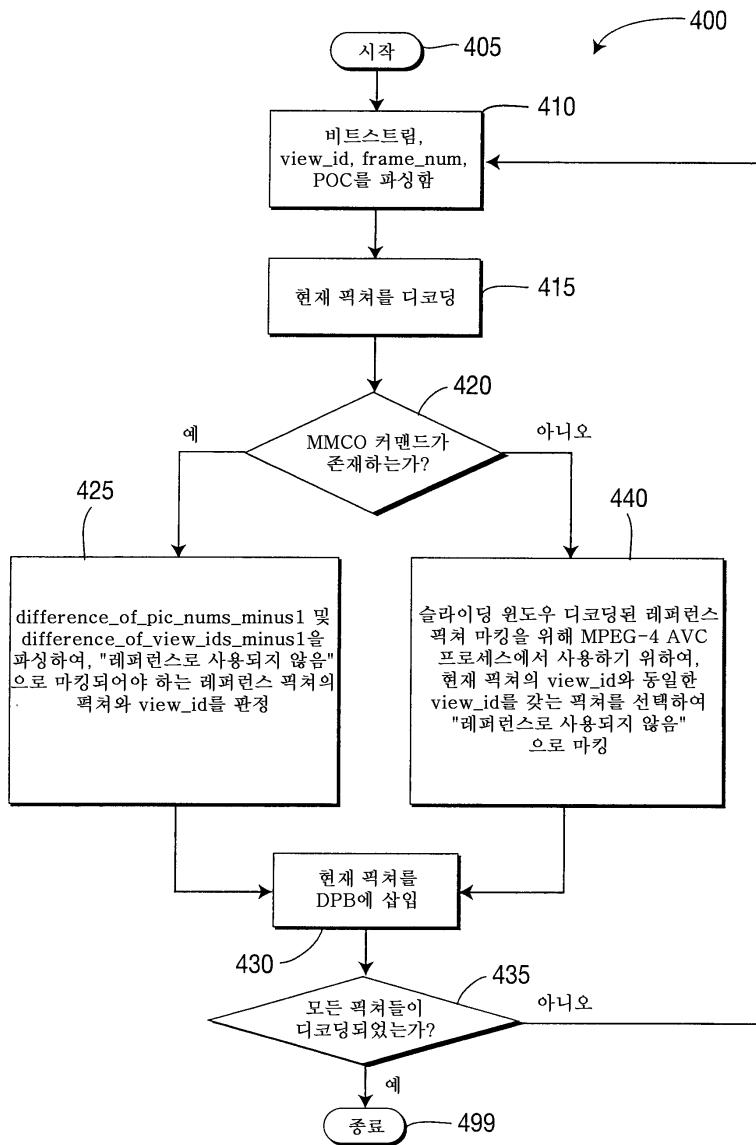
도면2b



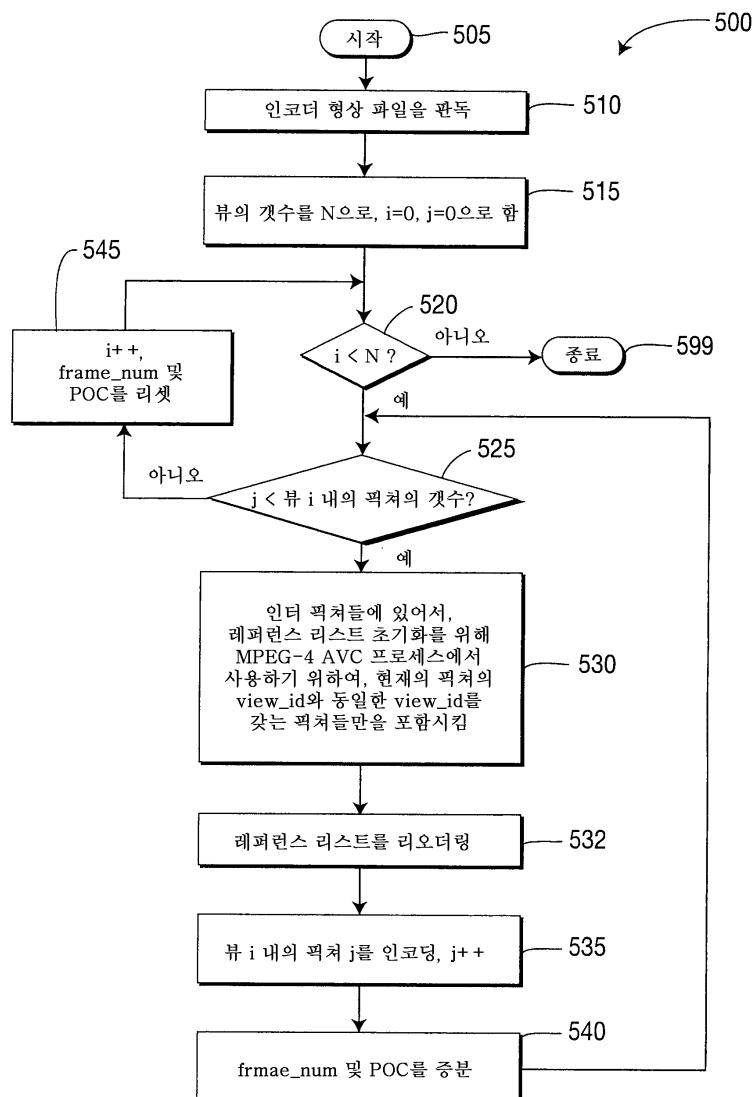
도면3



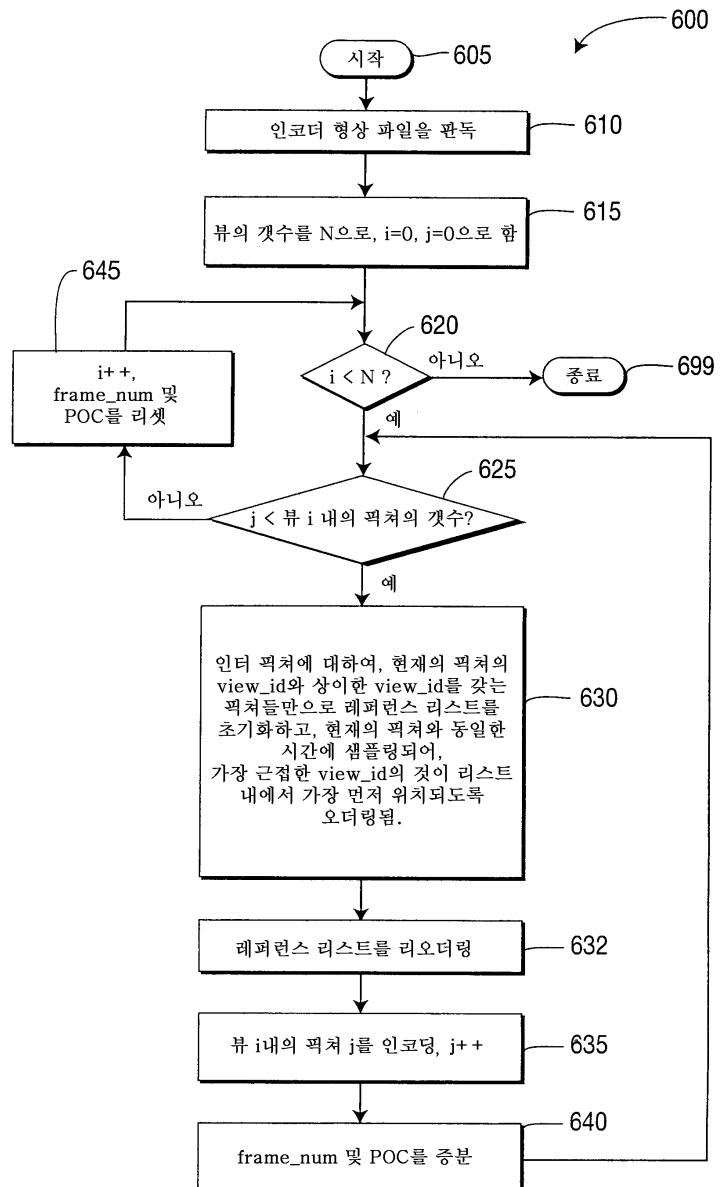
도면4



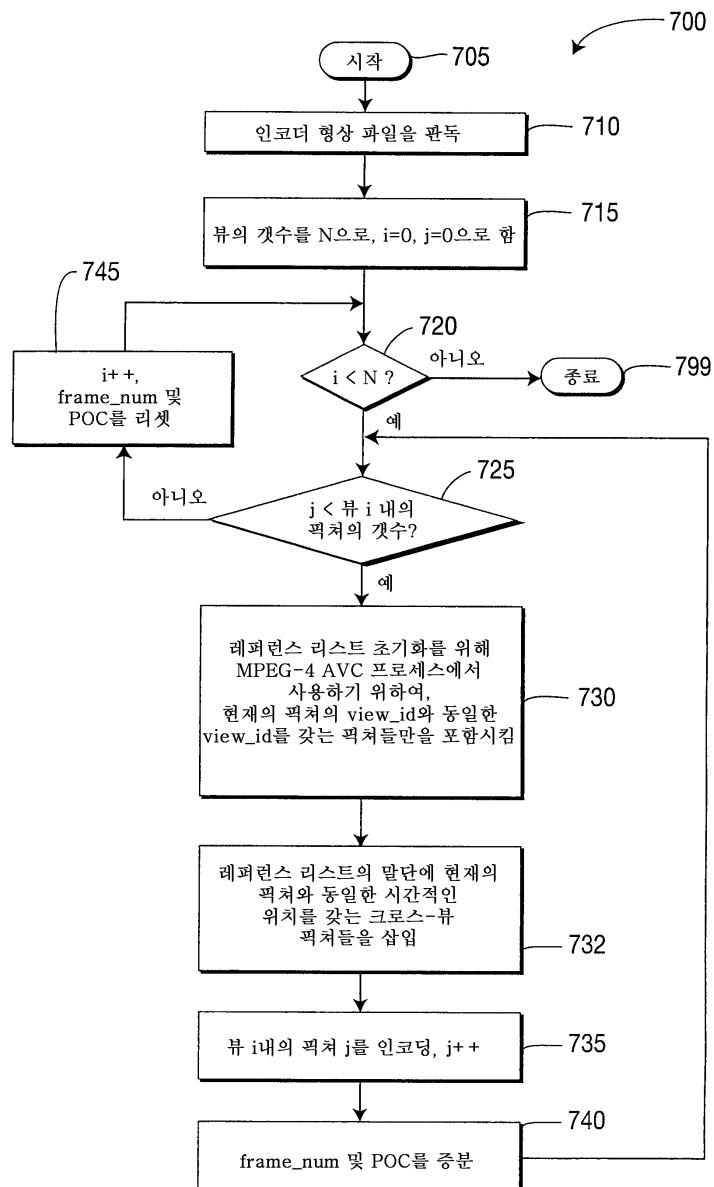
도면5



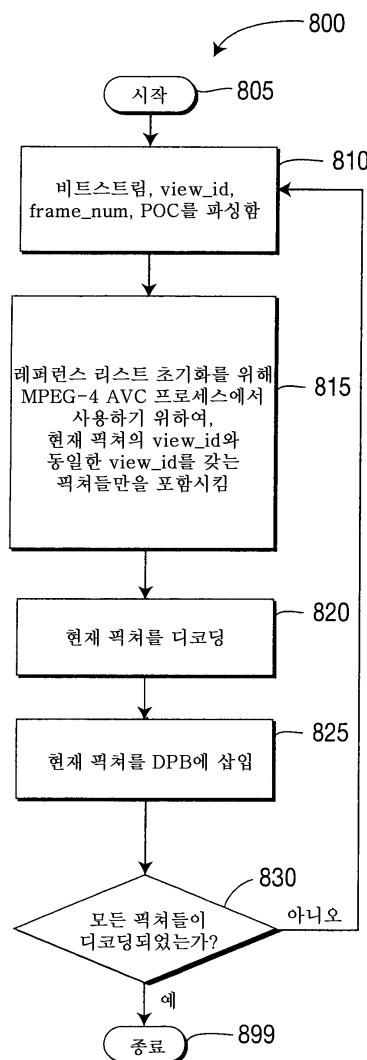
도면6



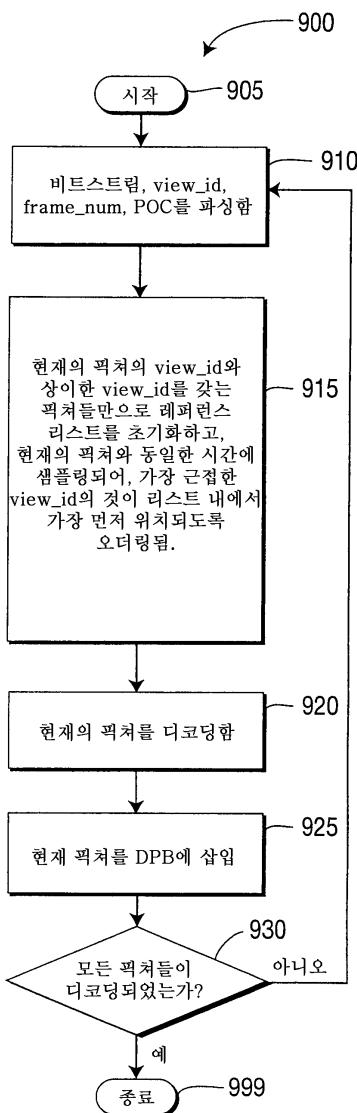
도면7



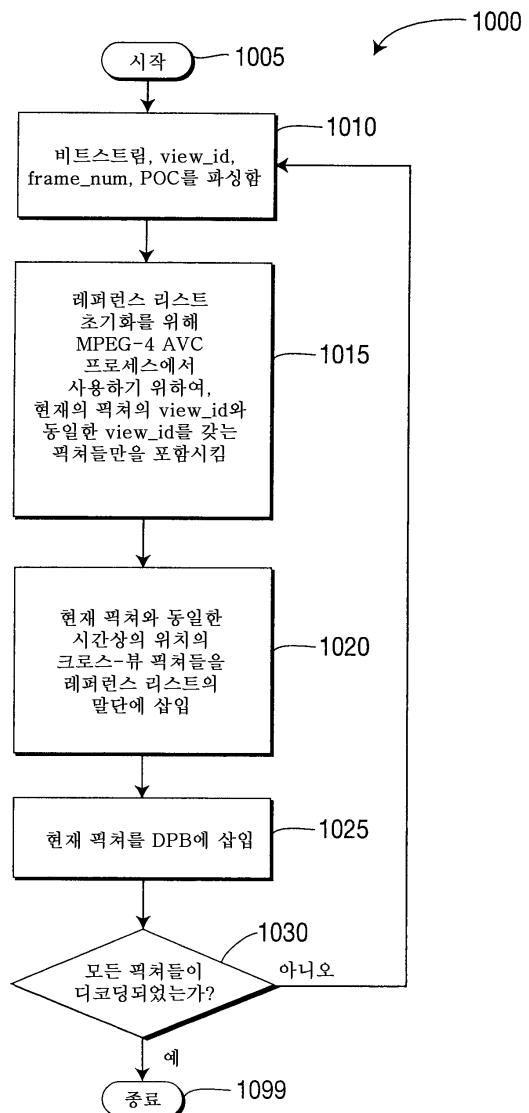
도면8



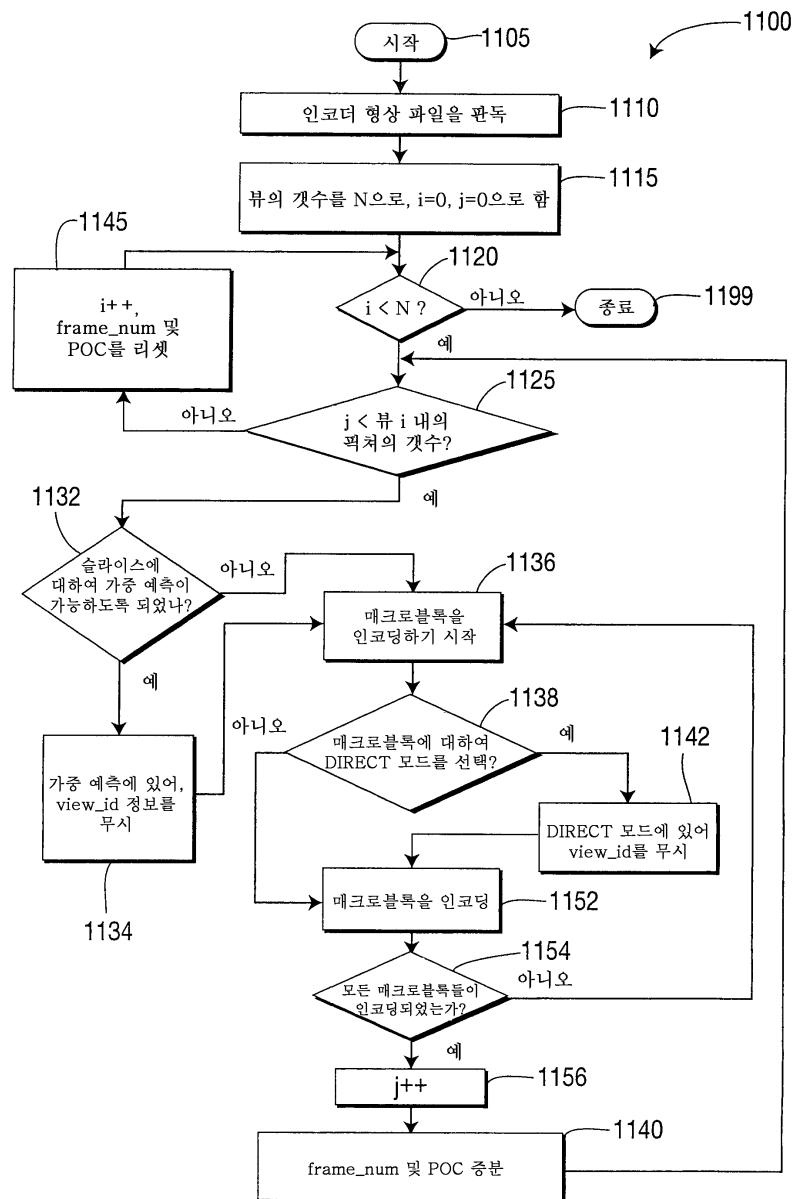
도면9



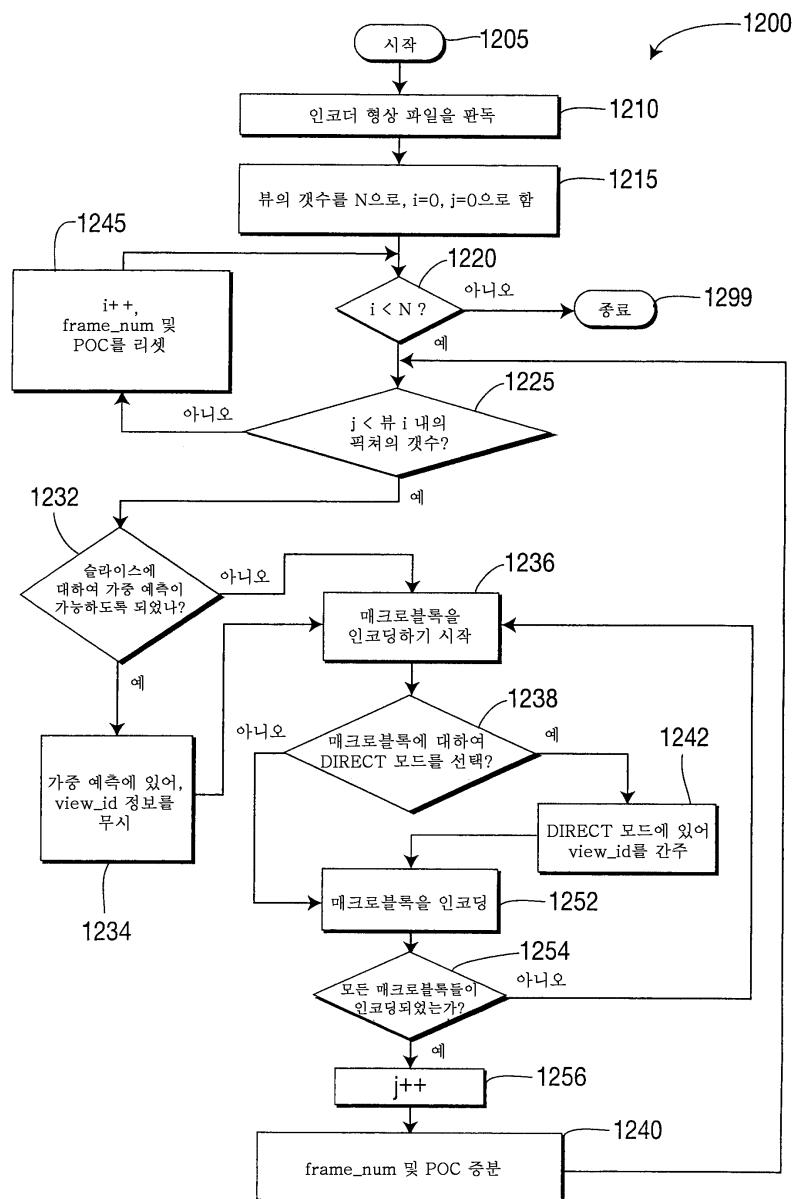
도면10



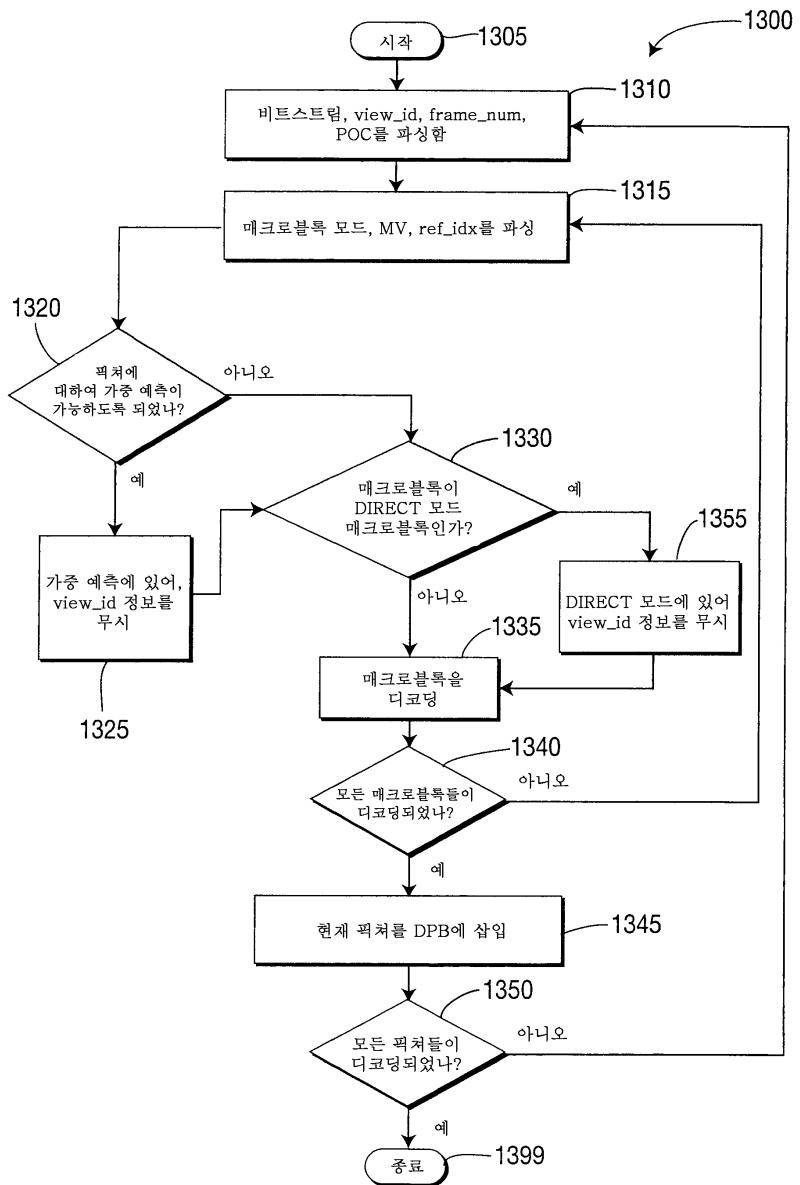
도면11



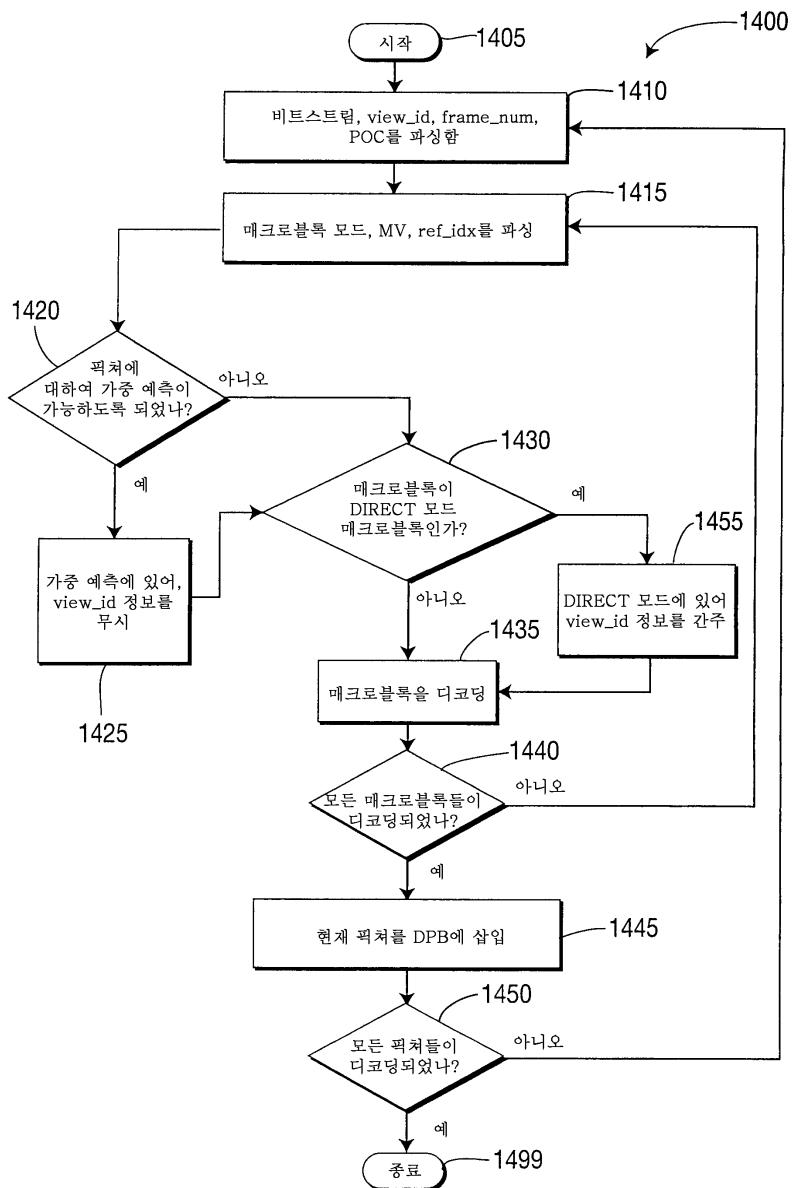
도면12



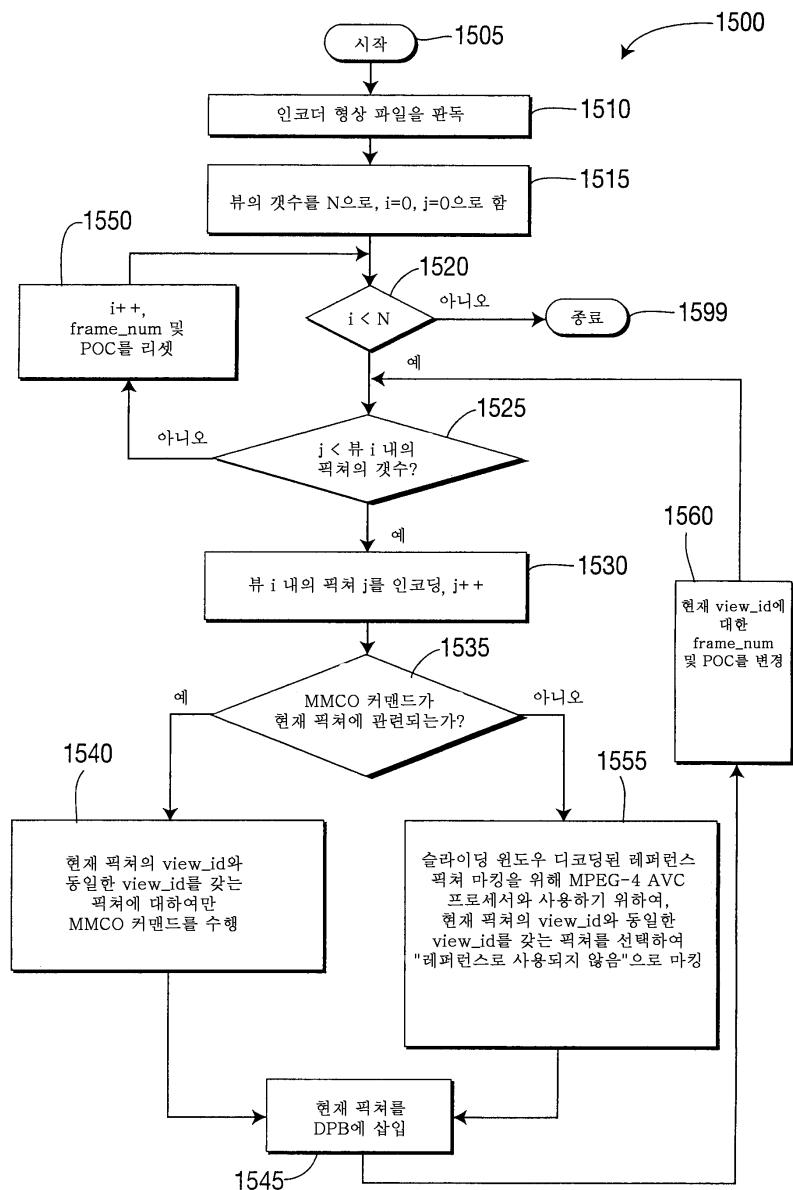
도면13



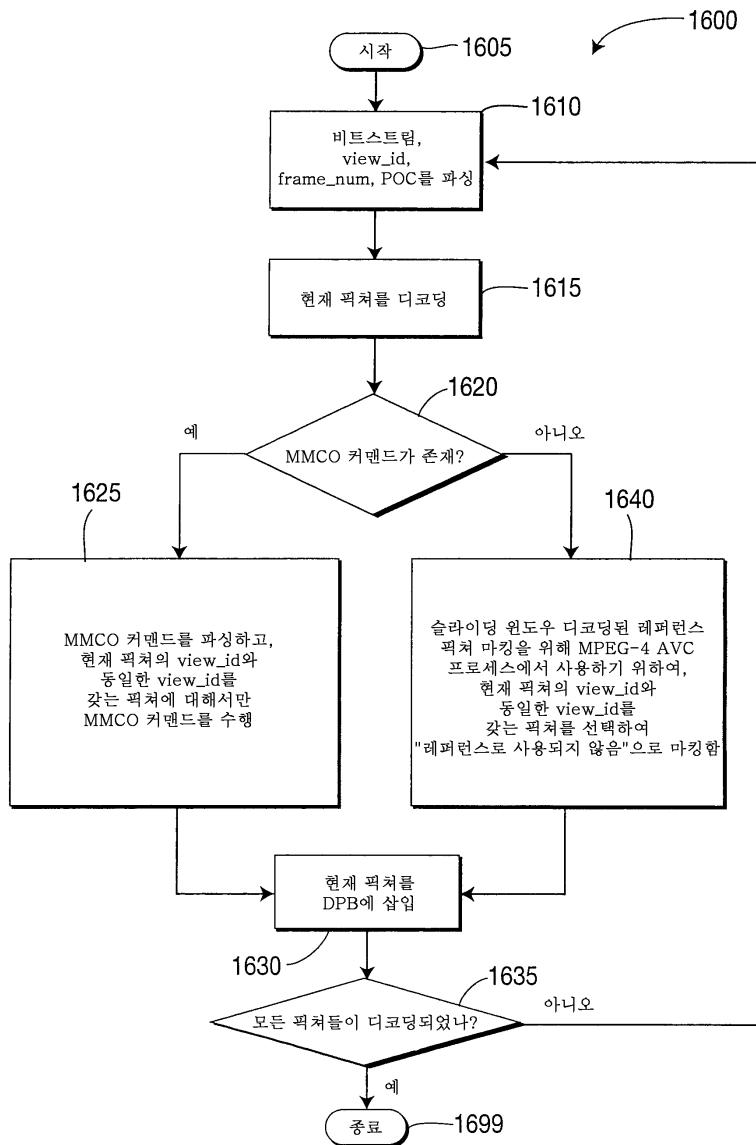
도면14



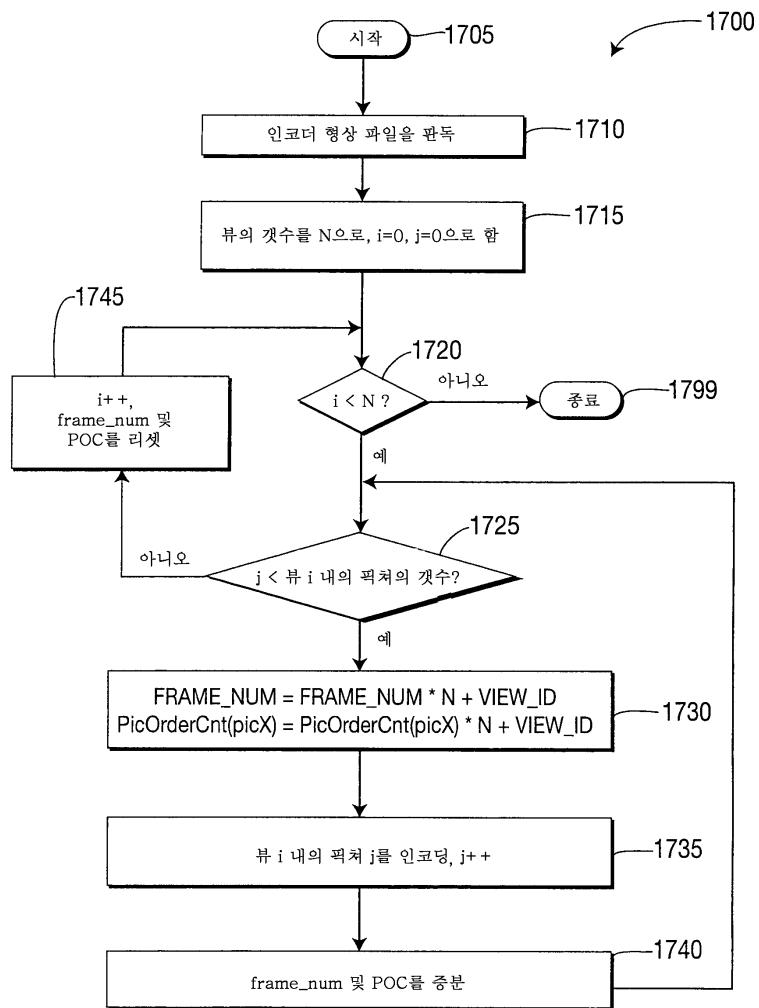
도면15



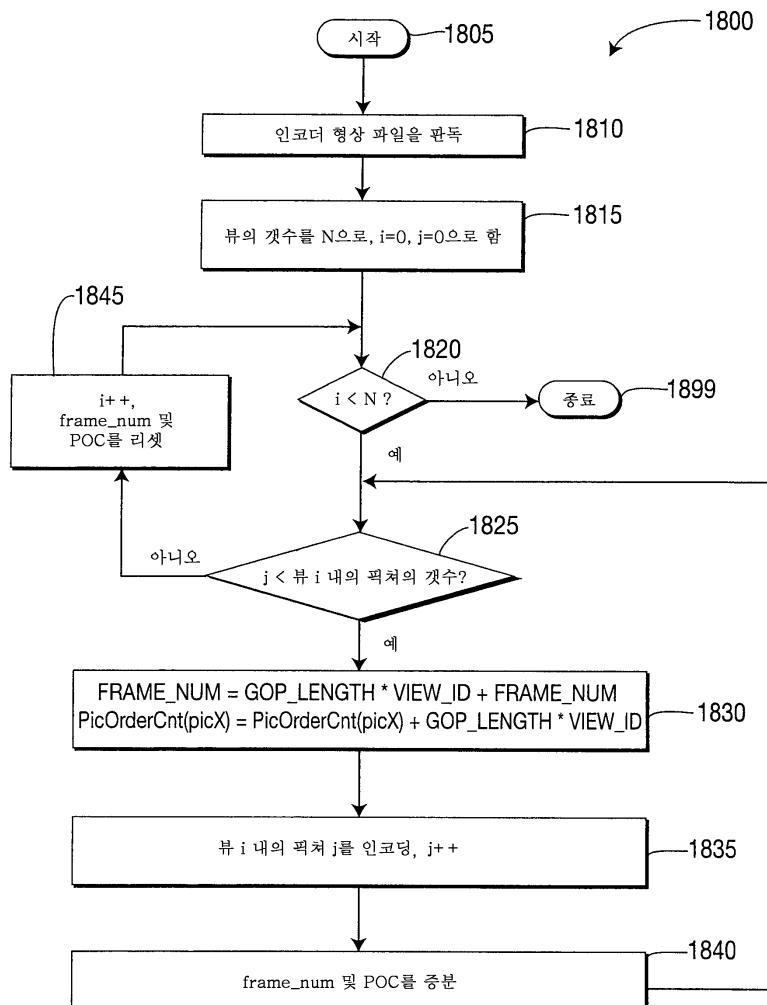
도면16



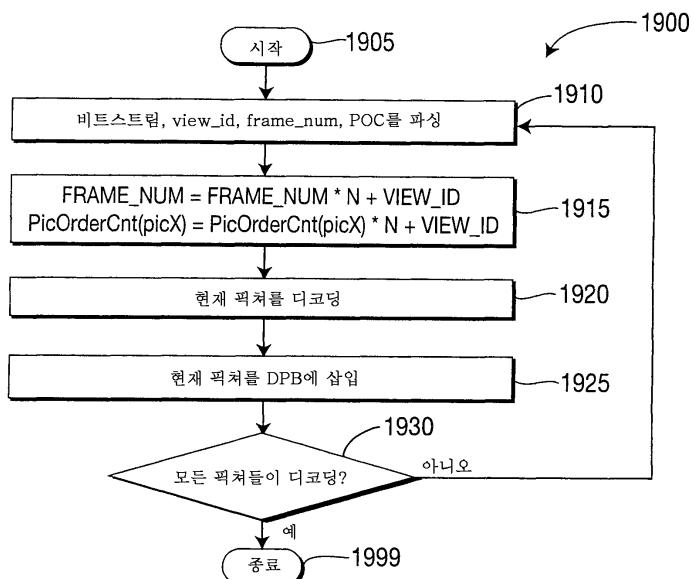
도면17



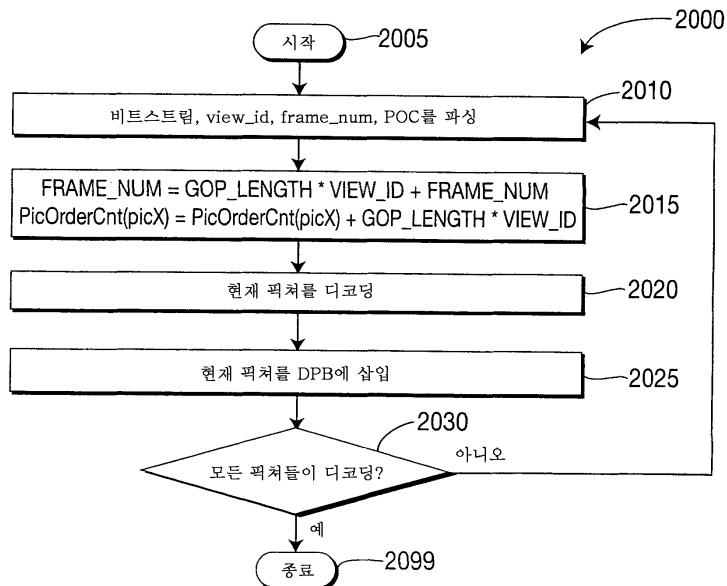
도면18



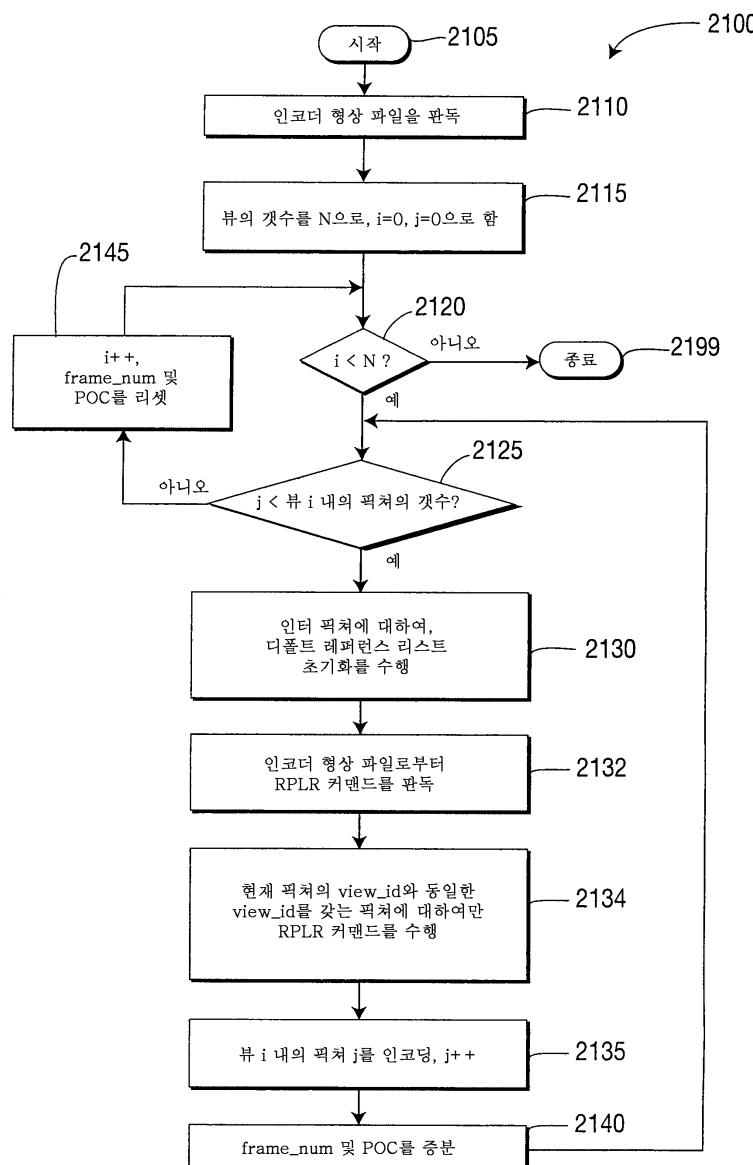
도면19



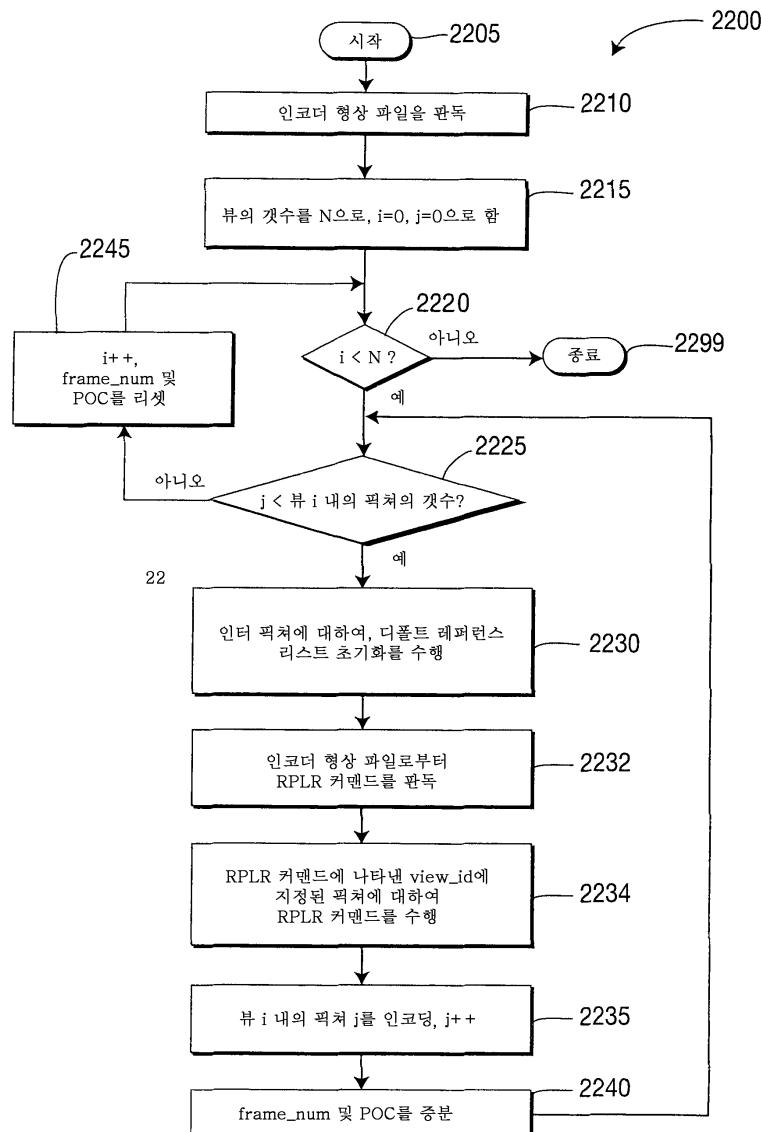
도면20



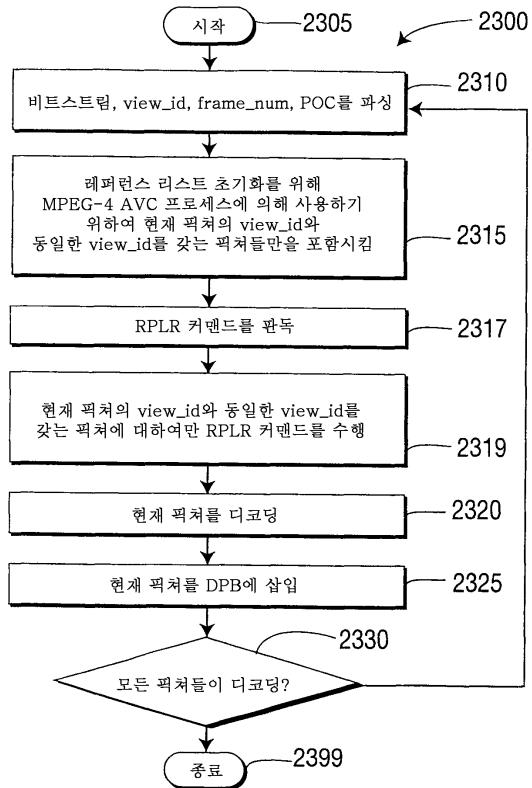
도면21



도면22



도면23



도면24

