(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년01월17일
(11) 등록번호 10-1351935
(24) 등록일자 2014년01월09일

(51) 국제특허분류(Int. Cl.)
H04N 19/423 (2014.01) *H04N 19/105* (2014.01)
H04N 13/00 (2006.01)
(21) 출원번호 10-2009-7007078
(22) 출원일자(국제) 2007년10월12일
심사청구일자 2012년10월10일
(85) 번역문제출일자 2009년04월06일
(65) 공개번호 10-2009-0077906
(43) 공개일자 2009년07월16일
(86) 국제출원번호 PCT/US2007/021902
(87) 국제공개번호 WO 2008/048515
국제공개일자 2008년04월24일
(30) 우선권주장
60/851,522 2006년10월13일 미국(US)
60/851,589 2006년10월13일 미국(US)
(56) 선행기술조사문헌
WO2007114612 A1
WO2006001653 A1
JP2008182669 A
WO2007081178 A1
전체 청구항 수 : 총 4 항

(73) 특허권자
톰슨 라이센싱
프랑스 92130 이씨레뮬리노 루 잔다르크 1-5
(72) 발명자
팬디트, 퍼빈, 비브하스
미국, 뉴저지주 08823, 프랜크린 파크, 피어 트리 래인 23
수, 예핑
미국, 워싱턴주 98682, 밴쿠버, 아파트먼트 비8, 노쓰이스트 원헌드레드나인쓰 에비뉴 3508
인, 펭
미국, 뉴저지주 08550, 웨스트 윈저, 워위크 로드 65
(74) 대리인
김학수, 문경진

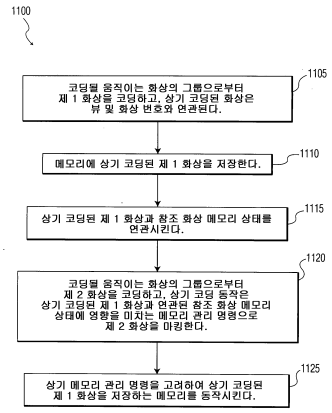
심사관 : 권오성

(54) 발명의 명칭 **멀티뷰 비디오 코딩을 수반하는 참조 화상 관리를 위한 방법**

(57) 요 약

일련의 메모리 관리 동작 명령이 메모리 내에 저장되는 디코딩된 참조 화상들의 메모리 관리(1110)와 멀티뷰 비디오 코딩 동작을 위해 설명된다. 비디오 코딩 동작은 저장된 참조 화상과 연관된 뷰에 비해 화상이 코딩될 뷰를 고려하고(1120), 이 경우 저장된 참조 화상들의 메모리 상태에 영향을 미치는 메모리 관리 동작 명령이 인에이블되며, 그러한 영향은 단기 참조 화상, 장기 참조 화상이거나 필요로 하지 않는 참조 화상을 지정하는 참조 화상의 지정일 수 있다(1125).

대 표 도 - 도11



특허청구의 범위

청구항 1

멀티뷰(multiview) 비디오 코딩을 사용한 제 1 참조 화상의 메모리 관리 방법으로서,

제 1 참조 화상을 메모리에 저장하는 단계(1110)로서, 상기 제 1 참조 화상은 메모리 상태 및 뷰와 연관되고, 상기 메모리 상태는 장기 참조 프레임, 단기 참조 프레임, 및 참조를 위해 사용되지 않는 것으로부터 선택되는, 제 1 참조 화상을 메모리에 저장하는 단계(1110)와,

상기 저장된 제 1 참조 화상의 메모리 상태에 영향을 미치는 정보로 비디오 화상을 코딩하는 단계(1120)로서, 상기 제 1 참조 화상과 연관된 뷰가 상기 코딩된 비디오 화상과 연관된 뷰와 상이한, 비디오 화상을 코딩하는 단계(1120)와,

메모리 상태 변경 명령을 통해, 상기 제 1 참조 화상의 메모리 상태를, 장기 참조 프레임, 단기 참조 프레임, 및 상기 저장된 제 1 참조 프레임이 참조 프레임 상태를 위해 사용되지 않은 것에 할당될 때 상기 저장된 제 1 참조 프레임이 상기 메모리로부터 삭제되는 참조 프레임을 위해 사용되지 않은 것으로부터 선택된 상기 제 1 참조 화상에 대한 상태로 변경하기 위한 상태를 할당하는 단계를 포함하고,

비디오 화상의 제 1 참조 및 뷰의 메모리 상태와 상이한 제 2 뷰와 연관된 제 2의 저장된 참조 화상의 메모리 상태는, 상기 제 2 참조 화상이 상기 메모리 상태 변경 명령에 의해 영향을 받지 않도록 코딩되는, 멀티뷰 비디오 코딩을 사용한 참조 화상의 메모리 관리 방법.

청구항 2

제 1항에 있어서,

상기 참조 화상은 처음에 H.264 기반의 코딩 동작을 사용하여 코딩되고, 상기 메모리 상태 변경은 멀티뷰 코딩 동작 동안 수행되는, 멀티뷰 비디오 코딩을 사용한 참조 화상의 메모리 관리 방법.

청구항 3

제 1항에 있어서,

상기 참조 화상은 코딩되고, 상기 메모리 상태에서의 상기 변경은 시간상(temporal) 코딩과 인터-뷰(inter-view) 코딩 모두를 행하는 비디오 코딩 동작에서 수행되는, 멀티뷰 비디오 코딩을 사용한 참조 화상의 메모리 관리 방법.

청구항 4

제 1항 내지 제 3항 중 어느 한 항에 있어서,

현재 코딩되는 상기 화상의 참조 마킹 모드들 중에서 선택하기 위해, 마킹 모드 구문 요소 플래그(marking mode syntax element flag)가 호출되는, 멀티뷰 비디오 코딩을 사용한 참조 화상의 메모리 관리 방법.

청구항 5

삭제

청구항 6

삭제

청구항 7

삭제

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

명 세 서

기 술 분 야

[0001] 관련 출원에 대한 상호-참조

[0002] 본 출원은, 2006년 10월 13일 출원된 미국 가출원 일련 번호 60/851,522호와 2006년 10월 13일 출원된 미국 가출원 일련 번호 60/851,589호의 이익을 주장하고, 이들 모두 그 전문이 본 명세서에 참조로 통합되어 있다.

[0003] 본 발명은 동화상 분야에 관한 것으로, 특히 멀티-뷰 비디오 코딩과 연관된 동화상의 메모리 유지 이슈에 관한 것이다.

배 경 기 술

[0004] 많은 인터프레임 인코딩 시스템들은 참조 화상들을 사용하고, 그러한 참조 화상들의 사용은 인코딩된 비트 스트림의 크기를 감소시키는 것에 도움이 된다. 이러한 타입의 결과는 인코딩 효율이 그 자체로 단지 인트라 프레임 인코딩 기술들을 사용하는 것보다 더 양호하다는 것이다. 그러므로, 많은 인코딩 표준들은 일련의 동영상으로부터 비트스트림을 인코딩하기 위해, 인트라 프레임과 인터 프레임 인코딩 기술 모두를 통합한다. 관련 분야에 공지된 것처럼, 오직 화상 자체 내로부터의 요소들을 사용하여 인코딩되는 "I" 화상(인트라 프레임), 화상 자체 내로부터의 요소 및/또는 2개의 이전 참조 화상들로부터의 요소들을 사용하여 인코딩되는 "B" 화상(인터프레임), 및 화상 자체내로부터의 요소들 및/또는 하나의 이전 참조 화상으로부터의 요소들(인터프레임)을 사용하여 인코딩되는 "P" 화상과 같은 상이한 타입의 참조 화상들이 인코딩 표준들을 위해 사용된다. "B"와 "P" 화상들은 다수의 참조 화상들을 사용할 수 있지만, 이들 타입의 화상들 모두 사이의 차이점은, "P"가 예측된 블록마다 오직 하나의 예측기를 사용하는 것을 허용하는데 반해, "B"는 블록마다 많아야 2개의 움직임 보상된 예측 신호들을 지닌 사이 예측(inter prediction)의 사용을 허용한다는 점이다.

[0005] "B" 또는 "P" 화상들이 인코딩되고/인코딩되거나 디코딩될 때, 그러한 화상들은 다른 참조 프레임들에 의존하게 되어, 그러한 화상들은 디코딩 동작 동안 적당하게 인코딩되거나 구성될 수 있다. 인코딩/디코딩 시스템은 참조 화상이 저장될 수 있도록 일부 타입의 메모리 위치를 제공해야 하지만, 다른 화상들은 그러한 참조 화상들을 고려하여 인코딩되거나 디코딩된다. 분명히, 잠시 후, 참조 화상은 코딩 동작을 위해 사용될 수 없는데, 이는 추후의 코딩 동작 동안 참조 화상을 사용할 코딩될 어떠한 화상도 존재하지 않기 때문이다.

[0006] 비록, 저장 디바이스에 모든 참조 화상을 영구히 저장할 수 있지만, 그러한 해결책은 메모리 자원을 비효율적으로 사용할 뿐이다. 그러므로, 관련 분야에 공지된 것과 같은 FIFO(First in First Out)나 LIFO(Last in First Out) 메모리 동작을 사용하는 것과 같음 메모리 기술들은, 그러한 참조 화상들을 위해 요구된 공간을 감소시키는 것(불필요한 참조 화상들을 버림으로써)을 돕기 위해 참조 화상들의 저장소를 지닌 메모리 디바이스를 동작시키는 경우 사용될 수 있다. 하지만, 그러한 메모리 동작은 인코딩되고/인코딩되거나 디코딩되는 화상들이 시간적인 및 뷰 사이 관계(view inter-relationship) 모두를 가지는 멀티뷰 코딩 시스템의 사용을 고려할 때 바람직하지 않은 결과들을 만들어낼 수 있다. 즉, 멀티뷰 코딩 시스템은 각 뷰가 각각의 물체/장면의 상이한 뷰를 나타내는 동화상들의 다수의 뷰를 가지는 양상을 도입한다. 이제, 하나의 참조 화상이 2개의 상이한 뷰와 연관된 화상들의 인코딩이나 디코딩에서 사용될 수 있다. 그러므로, 간단한 메모리 기술들은 그러한 환경에서 사용될 수 없다.

발명의 상세한 설명

[0007] 종래 기술의 이들 및 다른 결점 및 단점은, 본 발명의 원리에 의해 다루어지고, 이러한 본 발명의 원리는 비디오 인코딩을 위한 움직임 추정 예측기와 같은 이용 가능한 움직임 정보를 재사용하기 위한 방법 및 장치에 관한 것이다.

[0008] 본 발명의 원리들의 일 양상에 따르면, 디코더에 의해 디코딩되는 화상으로부터의 정보를 고려하여 메모리 디바이스에 저장된 참조 화상에 대해 메모리 관리 동작을 수행하는 코더(coder)가 제공되고, 이러한 경우 그러한 정보는 참조 화상과 연관된 뷰 정보와 관련된다.

[0009] 본 발명의 원리들의 이들 및 다른 양상, 특징 및 장점은 첨부 도면과 연계하여 읽어질 예시적인 실시예들의 후속하는 상세한 설명으로부터 분명해진다.

[0010] 본 발명은 후속하는 예시적인 도면에 따라 더 잘 이해될 수 있다.

실 시 예

[0022] 본 발명의 원리는 임의의 인트라-프레임과 인터-프레임 기반의 인코딩 표준에 적용될 수 있다. 본 명세서 전반에 걸쳐 사용되는 "화상"이라는 용어는 "화상"이라는 용어 자체뿐만 아니라, "프레임", "필드", 및 "슬라이스"로서 관련 분야에서 알려질 수 있는 비디오 이미지 정보의 다양한 형태들을 설명하기 위한 포괄적인 용어로서 사용된다.

[0023] 또, 본 발명의 설명부에서는, C 언어 타입의 포맷팅을 사용하는 다양한 명령어들(구문 요소들)이 그러한 명령어들에서 설명자들을 위한 후속하는 명명법을 사용하는 도면에서 열거되고 있다.

[0024] u(n): n개의 비트를 사용하는 부호 없는 정수. n이 구문 표에서 "v"이라면, 비트들의 개수는 다른 구문 요소들의 값에 의존하는 방식으로 변한다.

[0025] 이 설명자를 위한 구문 분석 과정은 최상위 비트가 처음 기입되는 방식으로 부호가 없는 정수의 2진 표현으로서 설명된 함수인 read_bits(n)의 반환값에 의해 특정된다.

[0026] ue(v): 좌측 비트가 처음에 오는 부호가 없는 정수인 Exp_Golomb-coded 구문 요소.

[0027] se(v): 좌측 비트가 처음에 오는 부호가 있는 정수인 Exp_Golomb-coded 구문 요소.

[0028] C: 구문 요소가 적용될 카테고리, 즉 무슨 레벨에 특별한 필드가 적용되어야 하는지를 나타낸다.

[0029] 본 설명은 본 발명의 원리를 예시한다. 그러므로, 비록 본 명세서에서 명백히 설명되거나 도시되지 않을지라도, 당업자라면 본 발명의 원리를 구현하고 본 발명의 취지와 범주 내에 포함되는 다양한 장치를 고안할 수 있다는 것을 알게 된다.

[0030] 본 명세서에서 인용된 모든 예와 조건부 언어는 읽는 사람으로 하여금 본 발명자(들)가 관련 분야를 개발시키기 위해 기여하게 될 본 발명의 원리와 개념을 이해하는데 도움을 주기 위한 교육학적인 목적을 위해 의도되며 그러한 특별히 인용된 예와 조건들에 제한되는 것으로 해석되지 않는다.

[0031] 더욱이, 본 명세서에서 본 발명의 원리, 양상, 및 실시예를 인용하는 모든 문장들은, 그것들의 특정 예와 더불어, 그것들의 구조상 및 기능상 등가물 모두를 포함하는 것으로 의도된다. 게다가, 그러한 등가물은 앞으로 개발될 등가물, 즉 구조에 관계없이 동일한 기능을 수행하는 개발될 임의의 요소뿐만 아니라 현재 알려진 등가물 모두를 포함하는 것으로 의도된다.

[0032] 그러므로, 예컨대 당업자라면 본 명세서에 나타난 블록도가 본 발명의 원리를 구현하는 예시적인 회로의 개념도를 나타냄을 알게 된다. 유사하게, 임의의 흐름 차트, 흐름도, 상태 전이도, 의사 코드 등이 실질적으로 컴퓨터 판독 가능한 매체에서 나타날 수 있고, 컴퓨터나 프로세서에 의해 그렇게 실행될 수 있는 다양한 프로세스들을, 그러한 컴퓨터나 프로세서가 명백히 도시되는지에 관계없이, 나타냄을 알게 된다.

[0033] 도면에 도시된 다양한 요소의 기능들은 적절한 소프트웨어와 관련하여 소프트웨어를 실행할 수 있는 하드웨어와 함께 전용 하드웨어의 사용을 통해 제공될 수 있다. 프로세서에 의해 제공될 때, 그 기능들은 단일 전용 프로세서, 단일 공유된 프로세서, 또는 일부가 공유될 수 있는 복수의 개별 프로세서에 의해 제공될 수 있다. 더욱이, "프로세서"나 "제어기"와 같은 용어의 명백한 사용은 소프트웨어를 실행할 수 있는 하드웨어만을 가리키는 것으로 여겨져서는 안 되고, 제한 없이 디지털 신호 처리기(DSP) 하드웨어, 소프트웨어를 저장하기 위한 읽기-전용

메모리(ROM), 랜덤 액세스 메모리(RAM), 및 비휘발성 저장기를 은연중에 포함할 수 있다.

[0034] 종래의 및/또는 주문형 다른 하드웨어가 또한 포함될 수 있다. 유사하게, 도면에 도시된 임의의 스위치들은 단지 개념적인 것이다. 그것들의 기능은 전용 로직, 프로그램 제어 및 전용 로직의 상호작용을 통한 프로그램 로직의 동작을 통해, 또는 심지어 수동으로 실행될 수 있고, 상황으로부터 더 특별하게 이해되는 바와 같이 구현자에 의해 특별한 기술이 선택 가능하다.

[0035] 청구항에서, 특정된 기능을 수행하기 위한 수단으로서 표현된 임의의 요소는, 예컨대 a) 그러한 기능을 수행하는 회로 요소들의 결합 또는 b) 따라서 그러한 기능을 수행하기 위한 소프트웨어를 실행하기 위한 적절한 회로와 결합된 펌웨어, 마이크로코드(microcode) 등을 포함하는 임의의 형태의 소프트웨어를 포함하여 그러한 기능을 수행하는 임의의 방식을 포함하는 것으로 의도된다. 그러한 청구항에 의해 한정되는 본 발명의 원리는, 다양한 인용된 수단에 의해 제공된 기능성들이 결합되고 청구항이 주장하는 방식으로 모아진다는 사실에 그 성질이 있다. 그러므로 이들 기능성을 제공할 수 있는 임의의 수단은 본 명세서에 도시된 것과 등가인 것으로 간주된다.

[0036] 본 명세서에서 본 발명의 원리의 "일 실시예" 또는 "실시예"라는 것은 본 발명의 원리의 적어도 하나의 실시예에 포함되는 실시예와 함께 설명된 특별한 특성, 구조, 특징 등을 의미하는 것이다. 그러므로, 본 명세서 전반에 걸쳐 다양한 곳에 등장하는 "일 실시예에서" 또는 "실시예에서"라는 어구의 등장은 반드시 모두 동일한 실시예를 가리키는 것은 아니다.

[0037] 도 1은 멀티뷰 코딩 시스템에서 사용된 참조 화상 구조의 예시적인 실시예를 나타낸다. 특히, 제시된 구조는 2006년 7월 오스트리아 클라겐푸르트(Klagenfurt)에서의 JVT-T208.doc에 A.Vetro, Y. Su, H.Kimata, A. Smolic에 의해 "Joint Multiview Video Model(JMVM) 1.0"에서 제안된 멀티뷰 인코딩(MVC) 구조에 따라 시간(T0 내지 T100) 동안 8개의 상이한 뷰(S0 내지 S7)의 사용에 관한 것이다. 이러한 멀티뷰 인코딩 표준은 AVC(Advanced Video Coding) 표준(2004년 10월 18일부터 22일까지 스페인 Palma de Mallorca에서 G.Sullivan, T.Wiegand, A.Luthra에 의한, "Draft of Version 4 of H.264/AVC{ITU-T 권고안 H.264와 ISO/IEC 14496-10(MPEG-4 part 10) Advanced Video coding}에서의 코딩"에 기초한다. 양 기준의 큰 차이점은 AVC가 멀티뷰 화상의 코딩을 다루지 않는데 비해 MVC는 멀티뷰 화상의 코딩을 다룬다는 것이다.

[0038] 다시 도 1을 참조하면, 예컨대 T1에서 뷰(S1)와 연관된 화상을 코딩할 때 코딩될 화상이 동일한 뷰(T0에서의 S1과 T2에서의 S1)로부터의 화상들(참조 화상들)과 관련되고, 코딩될 화상이 상이한 뷰(T1에서의 S0과 T1에서의 S2)로부터의 화상들로부터의 화상들에 관련된다는 점을 볼 수 있다. 따라서, S1, T1과 연관된 화상을 코딩할 때, 하드웨어, 소프트웨어, 또는 그것들이 결합으로 구현될 수 있는 버퍼, 레지스터, RAM 등과 같은 메모리 디바이스에 참조 화상들(T0에서 S1, T2에서 S1, T1에서 S0, T1에서 S2)을 유지하는 것이 이치에 맞다. 하지만 그러한 참조 화상들은 T1에서의 화상(S1)을 위해 사용된 참조 화상과는 상이한 참조 화상들의 사용에 의존하는 T98에서의 화상(S7)의 코딩을 고려할 때 유용하지 않을 수 있다.

[0039] 코딩 동작을 위한 버퍼의 효과적인 메모리 관리에 관한 한 가지 해결책이, AVC 비디오 표준과 연관되는 디코딩된 화상 버퍼(DPB)의 사용시 개시된다. 도 2에서의 블록도(200)의 간략화된 버전에서, 코더(205), 코딩 버퍼(210), 및 디코딩된 화상 버퍼(215) 사이의 동작이 도시되어 있다. 코딩 동작(인코딩이나 디코딩) 동안, 코더(205)에 의해 현재 코딩되는 화상이 코딩 버퍼(210)에 존재하는데 반해, 이전에 코딩된 참조 화상들은 디코딩된 화상 버퍼(215)에 저장되어 있다. AVC는 코더(205)가 어떻게 디코딩된 화상 버퍼(215)에서의 참조 화상들이 유지되어야 하는지를 특정하는 것을 허용하는 메모리 관리 제어 동작들(MMCO)로서 알려진 사용 명령들을 개시한다. 즉, 한 화상이 인코딩될 때 그러한 MMCO들이 그러한 화상 전에 오는 참조 화상들로 무엇이 행해져야 할지를 특정하는 것에 관해 현재 인코딩되는 화상의 헤더에 입력된다. 이러한 동작을 "마킹(marking)"이라고 한다. 이들 명령은 이후 디코딩된 화상 버퍼(215)에 존재하는 참조 화상으로 무엇이 행해져야 하는지를 결정하는 것에 관해 미래에 코더(205)에 의해 사용될 수 있다. 비록 화상이라는 용어가 다양한 요소들의 비디오 정보를 나타내기 위해 사용되더라도, AVC는 그러한 참조 화상들이 "참조 화상"과 동일한 화상으로부터의 슬라이스들을 사용할 수 있는 슬라이스들의 사용을 가리키고, 어떻게 한 화상이 하위 분할될 수 있는지에 관계없이, 본 발명의 원리가 적용된다는 점이 주목되어야 한다.

[0040] 도 3은 MMCO 명령들을 구현하기 위해 사용되는 AVC에서의 명령(**dec_ref_pic_marking**)을 나타낸다. 특히, 한 참조 화상이 단기(short term) 참조 화상, 장기(long term) 참조 화상으로 마킹되거나, MMCO 명령을 이용할 때 그 화상이 참조 화상이 아닌 것으로 마킹된다(그러한 경우 참조 화상을 메모리가 필요로 할 때 버려지게 된다). 참조 화상의 상태는, 한 화상이 코딩된 화상일 때 단기라고 지정되는 참조 화상이, 제 2 화상이 코딩될 때 장기

참조 화상이라고 식별될 수 있는 것과 같이, 더 많은 화상이 코딩될 때 변경될 수 있다.

[0041] 도 3은 또한 화상 헤더들(슬라이스 헤더들)의 마킹을 수행하기 위한 2가지 상이한 모드 사이에 사용되는 **adaptive_ref_pic_marking_mode_flag**라고 알려진 명령 플래그를 나타낸다. 그 플래그가 "0"으로 설정될 때에는, 단기 참조 화상들을 위한 FIFO 메커니즘을 제공하는 슬라이딩 윈도우(sliding window) 참조 마킹 모드가 활성화된다. 플래그가 "1"로 설정될 때에는, 참조 화상들을 "참조를 위해 사용되지 않는(unused for reference)"이라고 마킹하고 참조 화상들에 장기 프레임 인덱스들을 할당하는 등의 구문 요소들을 제공하는 적응성 참조 화상 마킹 모드가 활성화된다. AVC에서 사용된 MMCO 명령들을 통한 참조 화상들에 관한 다양한 할당이 아래 표 1에 도시되어 있다.

표 1

[0042]

memory_management_control_operation	메모리 관리 제어 동작
0	종료 memory_management_control_operation 구문 요소 루프
1	단기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹함
2	장기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹함
3	단기 참조 화상을 "장기 참조를 위해 사용됨"이라고 마킹하고, 그것에 장기 프레임 인덱스를 할당함
4	최대 장기 프레임 인덱스를 특정하고, 최대값 보다 큰 장기 프레임 인덱스들을 가지는 모든 장기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹함
5	모든 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹하고, MaxLongTermFrameIdx 변수를 "장기 프레임 인덱스들이 없음"이라고 설정함
6	현재 화상을 "장기 참조를 위해 사용됨"이라고 마킹하고, 그것에 장기 프레임 인덱스를 할당함

[0043] AVC의 설계가 지닌 한 가지 문제점은, 한 화상의 실제 코딩 순서(화상들의 시퀀스에서의)를 나타내는 그것들 각각의 프레임 번호(frame_num)와, 한 화상이 디스플레이될 순서인 화상 각각의 화상 순서 카운트(POC)에 의해 화상들이 식별될 수 있다는 점이다. 하지만 MVC는 다수의 뷰가 MVC에서 고려되어야 하기 때문에 AVC보다 더 복잡한데 비해, AVC는 단지 하나의 뷰에 관계된다. 따라서, MVC에서 추가적인 값의 뷰 아이디(view_id)가 하나의 특별한 화상을 하나의 특별한 뷰에 연관시키기 위해 사용된다.

[0044] 그러므로, AVC의 MMCO를 MVC로부터의 view_id들의 사용과 결합할 때, 종래 기술의 MMCO들의 현재 사용은 오직 사용자가 동일한 view_id의 화상들에 따라 MMCO들을 공급하는 것을 허용한다. 즉, 코딩되는 화상은 오직 동일한 view_id 타입의 다른 화상들을 가리킬 수 있다(뷰 1 타입 화상들은 오직 다른 뷰 1 타입 화상들을 위한 MMCO 명령을 공급할 수 있다). MMCO 명령들의 현재 사용을 통해 이들 상이한 뷰들 모두를 놓치지 않고 따라가야 하는 것은, DPB를 동작시키기 위한 메모리 관리의 비효율적인 사용을 방지한다.

[0045] 특히, 도 4는 추가 구문이 SPS(크로스 뷰 참조들을 시그널링하기 위해 사용되는)에서 추가된 현재의 MVC를 나타낸다. 추가된 구문은 아래에 설명되는 방식으로 앵커(anchor)(즉, I 화상들)와 비-앵커(non-anchor) 화상들을 위해 사용될 크로스-뷰(cross-view) 참조들을 표시하기 위해 사용된다.

[0046] 앵커 화상은 통상적으로 모든 슬라이스 오직 동일한 화상 순서 카운트를 갖는 슬라이스를 참조하는, 즉 현재 뷰에서는 슬라이스가 아닌 다른 뷰들에서의 슬라이스만 참조하는 코딩된 화상을 나타낸다는 점을 주목하라. 그러한 화상은 anchor_pic_flag를 1로 설정함으로써 시그널링된다. 앵커 화상을 디코딩한 후, 디스플레이 순서에서의 후속하는 코딩된 화상들 전부는 앵커 화상 전에 임의의 다른 화상 디코더로부터의 사이 예측을 사용하는 것 내에서 디코딩될 수 있게 된다. 하나의 뷰 내의 한 화상이 앵커 화상이라면, 다른 뷰들에서의 동일한 시간상 인덱스(temporal index) 내의 모든 화상들 또한 앵커 화상들이라고 알려지게 된다.

[0047] 참조 예측 목록들 내로 현재 뷰와는 상이한 뷰로부터의 참조 화상들을 놓기 위해 후속 절차가 행해진다.

[0048] - 0으로부터 num_multiview_ref_for_listX-1까지의 "I"의 각각의 값에 관해,

[0049] - 현재 화상과 시간상 정렬되는 view_reference_view_for_list_X[i]로부터의 재구성된 화상이 얻어지고 디코딩

된 화상 버퍼(DPB) 내로 삽입된다.

[0050] - 그 화상으로의 인덱스가 RefPicListX에서의 다음 빈 슬롯으로 삽입된다.

[0051] 이러한 특정된 구현예에서, MMCO 명령은 오직 개별 뷰들하고만 직접적으로 연관되고, 다른 뷰들에서는 화상들을 마킹할 수 없다. 직접적인 결과로서, 크로스-뷰 참조 화상들은 필수적인 것(위에서 표시된 것처럼)보다 길게 DPB에서 머무를 수 있는데, 이는 그러한 화상이 오직 비트스트림에서 나중에 그것 자체의 뷰에서 한 화상에 의해 "참조를 위해 사용되지 않음"이라고 마킹될 수 있기 때문이다. 예컨대, 도 1을 참조하면, 뷰(S0)에서 T0 내지 T11에 관한 화상들은 오직 뷰들(S1,S2)을 위해 필요하고, 이후 참조를 위해 사용됨이라고 마킹된다. 따라서, 그러한 화상들을 저장하는 DPB는 큰 저장 영역을 요구하게 된다. 따라서, 한 화상이 어떤 뷰와 연관되는지를 고려하지 않고 DPB를 클리어하는 유일한 방식은, 새로운 GOP(group of picture)의 시작과 연관된 화상이나 IDR(instantaneous decoding refresh) 화상이 참조 화상들의 DPB를 완전히 클리어하는 것을 표시하는 것이다.

[0052] 그러므로 본 발명은 크로스-뷰들이 적용될 수 있는 MMCO를 제공함으로써 DPB 문제에 대한 해결책을 제안하고, 이는 하나의 화상이 코딩될 때 그러한 화상이 어떻게 뷰들(현재 코딩되는 화상과 동일하지 않은 뷰들)에 걸쳐 참조 화상들을 고려할지에 대한 정보를 포함하게 된다는 것을 의미한다.

[0053] 본 발명의 몇 가지 실시예들이 AVC 표준의 관점에서 제시되고, 이 경우 비록 그 원리들이 또한 다수의 뷰 화상들을 사용하는 다른 코딩 표준들에 적용된다는 점이 이해되어야 하더라도, 새로운 높은 레벨의 구문 요소들이 한정되고 논의된다.

[0054] 도 5에서 제시된 일 실시예에서는, 뷰들에 걸쳐 화상들을 마킹하기 위해 사용되는 새로운 구문 요소인 **dec_ref_pic_marking_mvc_extension**()이 있다. 이 함수는 코딩되는 화상의 화상 헤더 구문을 나타내는 도 6에서 나타난 대응하는 slice_header_mvc_extension() 함수로부터 호출된다(특별히 이러한 명령은 AVC에 도시된 슬라이스 헤더로부터 적응된다).

[0055] 이러한 새로운 구문이 오직 현재 뷰 외의 뷰에 있는 화상을 마킹하기 위해 사용되므로, 또한 시스템이 동일한 뷰 내의 화상들을 마킹하는 것을 허용하는 옵션을 제공하는 것이 고려되어야 한다. 동일한 뷰 내의 화상들의 마킹은, 새로운 마킹 프로세스 후 AVC 호환(compatible) 함수인 **dec_ref_pic_marking**(도 5)를 호출함으로써 인에이블된다. 그러한 함수는 MVC 기반의 마킹의 활성화 전 또는 후에 호출될 수 있다는 점이 주목된다.

[0056] AVC 구문이 오직 단일 뷰를 취하는, MVC에 관한 AVC 기반의 **dec_ref_pic_marking()**에는 추가적인 제약을 받는데, 이는 멀티뷰 시스템이 AVC 표준에서 초기에 다루어지지 않기 때문이다. 따라서, AVC 구문은 코딩되는 현재 화상이 속하는 뷰에만 적용되어야 한다.

[0057] 도 5를 다시 참조하면, 아래에 한정되는 몇 가지 추가 구문이 추가된다.

[0058] 현재 코딩되는 화상의 참조 마킹 모드 중에서 선택하기 위해 사용되는 **mvc_adaptive_ref_pic_marking_mode_flag**. "0"에서의 플래그는 단기 참조 화상들에 DPB에서의 FIFO 기초들이 할당되는 슬라이딩 윈도우 참조 화상 마킹 모드를 나타낸다. "1"에서의 플래그는 현재 코딩되는 화상과 연관된 뷰 외의 뷰에서의 참조 화상들을 마킹하기 위해 요소들이 제공될 수 있는 적응성 참조 화상 마킹 모드를 나타낸다. 다른 뷰들에서의 참조 화상들에 관한 상태들은 "참조를 위해 사용되지 않음"과 "장기 프레임 인덱스들"을 포함한다.

[0059] "장기 참조를 위해 사용됨"이라고 현재 마킹되는 화상들{프레임들, 보충(complementary) 필드 쌍들, 및 비-필드(non-field) 필드들}의 개수가 Max(Num_ref_frames,1)와 같을 때, 플래그는 1과 같게 된다.

[0060] **memory_management_control_operation**은 코더에 의한 참조 화상 마킹 동작에 영향을 미치기 위해 적용될 제어 동작(MMCO)을 특정한다. memory_management_control_operation 구문 요소 다음에는, 제어 동작의 값에 의해 특정된 동작을 위해 필수적인 데이터가 온다. 멀티뷰를 위해 MMCO들과 연관된 값들과 제어 동작들은 아래 표 2에 특정된다. memory_management_control_operation 구문 요소들은, 그러한 명령들이 화상 헤더(예컨대, 슬라이스 헤더)에서 등장하는 순서대로 코딩 처리하여 처리되고, 각 MMCO에 관해 표현된 의미론상 제약을 받는 사항(semantic constraint)은 개별 MMCO가 처리되는 특정 위치에서 적용된다.

표 2

[0061]

memory_management_control_operation	메모리 관리 제어 동작
0	memory_management_control_operation 구문 요소 루프를 종료함

1	그 자체가 "참조를 위해 사용되지 않음"이라고 마킹되지 않는 경우, 단기 참조 화상을 마킹함
2	그 자체가 "참조를 위해 사용되지 않음"이라고 마킹되지 않는 경우, 장기 참조 화상을 마킹함
3	그 자체가 "장기 참조를 위해 사용됨"이라고 마킹되지 않는 경우, 단기 참조 화상을 마킹하고, 장기 프레임 인덱스를 그것에 할당함
4	최대 장기 프레임 인덱스를 특정하고, 최대값보다 큰 장기 프레임 인덱스들을 가지는 모든 장기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹함
5	하나의 뷰 내의 모든 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹함
6	그 자체가 "참조를 위해 사용되지 않음"으로 마킹되지 않는 경우, 모든 뷰에서의 모든 참조 화상을 마킹하고, MaxLongTermFrameIdx 변수를 "장기 프레임 인덱스들이 없음"으로 설정함
7	그 자체가 "단기 참조를 위해 사용됨"이라고 마킹된 경우 이 외에 뷰에서의 장기 참조 화상을 마킹함

[0062] memory_management_control_operation은, 그것이 코딩 프로세스에 의해 처리될 때 특정된 참조 화상이 "단기 참조를 위해 사용됨"이라고 마킹되지 않는 한, 화상 헤더(예컨대, 슬라이스 헤더)에서 1과 같지 않게 된다.

[0063] memory_management_control_operation은, 그것이 디코딩 프로세스에 의해 처리될 때 특정된 장기 화상 번호가 "장기 참조를 위해 사용됨"이라고 마킹되는 참조 화상을 가리키지 않는 한, 슬라이스 헤더에서 2와 같지 않게 된다.

[0064] memory_management_control_operation은, 그것이 디코딩 프로세스에 의해 처리될 때 특정된 참조 화상이 "단기 참조를 위해 사용됨"이라고 마킹되지 않는 한, 슬라이스 헤더에서 3과 같지 않게 된다.

[0065] memory_management_control_operation은, 그것이 디코딩 프로세스에 의해 처리될 때, 변수인 MaxLongTermFrameIdx의 값이 "어떠한 장기 프레임 인덱스들도 없음"과 같게 된다면, 3,5 또는 6과 같지 않게 된다.

[0066] 4와 같은 1개 이하의 memory_management_control_operation이 화상 헤더에 존재하게 된다.

[0067] 5와 같은 1개 이하의 memory_management_control_operation이 화상 헤더에 존재하게 된다.

[0068] 6과 같은 1개 이하의 memory_management_control_operation이 화상 헤더에 존재하게 된다.

[0069] 단기 참조 프레임의 부분이거나 단기 보충 참조 필드 쌍의 부분인 필드에 장기 프레임 인덱스를 할당하는, 3과 같은 필드 및 memory_management_control_operation 명령을 디코딩할 때, 동일한 장기 프레임의 인덱스를 동일한 프레임의 나머지 필드나 보충 참조 필드 쌍에 할당하기 위한 또 다른 memory_management_control_operation이 구문 구조를 마킹하는 동일한 디코딩된 참조 화상에 존재하게 된다.

[0070] 위의 요구 조건은, 심지어 MMCO에 의해 참조된 필드가 3과 같고, 계속해서 예컨대 MMCO가 필드로 하여금 "참조를 위해 사용되지 않음"이라고 마킹되게 하는 화상 헤더에서 2와 같을 때와 같이, "참조를 위해 사용되지 않음"이라고 마킹될 때에도 충족되어야 한다는 점을 주목하라.

[0071] 보충 참조 필드 쌍의 제 1 필드(디코딩 순서에서의)가 1과 같은 long_term_reference_flag 또는 6과 같은 memory_management_control_operation 명령을 포함할 때, 보충 참조 필드 쌍의 나머지 필드에 관한 디코딩된 참조 화상 마킹 구문 구조는 나머지 필드에 동일한 장기 프레임 인덱스를 할당하는 6과 같은 memory_management_control_operation 명령을 포함하게 된다.

[0072] 위의 요구 조건은, 심지어 보충 참조 필드 쌍이, 계속해서 예컨대 MMCO가 필드로 하여금 "참조를 위해 사용되지 않음"이라고 마킹되게 하는 제 2 필드의 화상 헤더에서 2와 같을 때와 같이, "참조를 위해 사용되지 않음"이라고 마킹될 때에도 충족되어야 한다는 점을 주목하라.

[0073] **difference_of_view_id**는 현재의 memory_management_control_operation이 적용 가능한 view_id를 유도하기 위해 사용된다.

[0074] **difference_of_pic_nums**는 그 자체가 장기 프레임 인덱스에 이전에 할당되지 않은 경우 뷰에서 단기 참조 화상

에 장기 프레임 인덱스를 할당하고, 자체가 "참조를 위해 사용되지 않음"으로 마킹되는 경우 이외에 뷰에서 단기 참조 화상을 마킹하기 위해 사용된다. 연관된 memory_management_control_operation이 디코딩 프로세스에 의해 처리되면, difference_of_pic_nums로부터 유도된 그 결과 화상 번호는 "참조를 위해 사용됨"으로서 마킹된 참조 화상 중 하나에 할당된 화상 번호가 된다.

[0075] 그 결과 화상 번호는 다음과 같이 제약을 받는다.

[0076] - field_pic_flag가 0과 같다면, 그 결과 화상 번호는 참조 프레임들이나 보충 참조 필드 쌍들에 할당된 화상 번호들의 세트 중 하나가 된다.

[0077] - field_pic_flag가 0과 같다면, 그 결과 화상 번호는 양 필드가 "참조를 위해 사용됨"이라고 마킹되는 보충 참조 필드 쌍이나 양 필드가 "참조를 위해 사용됨"이라고 마킹되는 프레임에 할당된 화상 번호이어야 한다. 특히, field_pic_flag가 0과 같다면, 단일 필드가 "참조를 위해 사용됨"이라고 마킹되는 쌍으로 되지 않은 필드나 프레임의 마킹은 1과 같은 memory_management_control_operation에 의해 영향을 받을 수 없다.

[0078] - 그렇지 않으면(field_pic_flag가 1과 같다면,), 그 결과 화상 번호는 참조 필드들에 할당된 화상 번호들의 세트 중 하나가 된다.

[0079] **long_term_frame_idx**는 현재 화상의 view_id와는 상이한 view_id를 지닌 화상에 장기 프레임 인덱스를 할당하기 위해 사용된다(이 경우 memory_management_control_operation은 2와 같음). 연관된 memory_management_control_operation이 디코딩 프로세스에 의해 처리될 때, long_term_frame_idx의 값은 0부터 MaxLongTermFrameIdx까지 포함하는 범위에 있게 된다.

[0080] **difference_of_pic_nums**의 구문은 현재 화상의 picNum보다 큰 picNum을 지닌 화상들을 선택하는 것을 허용한다. 이는 마킹을 더 효율적이게 한다.

[0081] 표 2에 도시된 상이한 함수들의 적용이 아래에 도시된다.

[0082] MMCO가 1과 같을 때, 이는 단기 참조 화상이 "참조를 위해 사용되지 않음"이라고 한정됨을 나타낸다. 그러므로, picNumX를

[0083] picNumX = CurrpicNum - (difference_of_pic_nums)로 특정한다.

[0084] viewId를

[0085] viewIdX = CurrViewId - (difference_of_view_id)로 특정한다.

[0086] field_pic_flag에 따라, 단기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹하기 위해, picNumX의 값이 다음과 같이 사용된다.

[0087] - field_pic_flag가 0과 같다면, 단기 참조 프레임이나 viewIdX에 의해 특정된 뷰에서 picNumX에 의해 특정된 단기 보충 참조 필드 쌍, 및 이들 필드 모두는 "참조를 위해 사용되지 않음"이라고 마킹된다.

[0088] - 그렇지 않으면(field_pic_flag가 1과 같다면), viewIdX에 의해 특정된 뷰에서 picNumX에 의해 특정된 단기 참조 필드는 "참조를 위해 사용되지 않음"이라고 마킹된다. 참조 필드가 참조 프레임이나 보충 참조 필드 쌍의 부분이라면, 프레임이나 보충 필드 쌍은 또한 "참조를 위해 사용되지 않음"이라고 마킹되지만, 나머지 필드의 마킹은 변경되지 않는다.

[0089] MMCO가 2와 같을 때에는, 이러한 상황이 장기 참조 화상이 "참조를 위해 사용되지 않음"이라고 변경됨을 나타낸다. field_pic_flag에 따라, LongTermPicNum의 값이 다음과 같이 "참조를 위해 사용되지 않음"이라고 장기 참조 화상을 마킹하기 위해 사용된다.

[0090] - field_pic_flag가 0과 같다면, 장기 참조 프레임이나 long_term_pic_num과 같은 LongTermPicNum을 가지는 장기 보충 참조 필드 쌍, 및 이들 필드 모두는 "참조를 위해 사용되지 않음"이라고 마킹된다.

[0091] - 그렇지 않으면(field_pic_flag가 1과 같다면), long_term_pic_num과 같은 LongTermPicNum에 의해 특정된 장기 참조 필드는 "참조를 위해 사용되지 않음"이라고 마킹된다. 참조 필드가 참조 프레임이나 보충 참조 필드 쌍의 부분일 때에는, 그 프레임이나 보충 필드 쌍이 또한 "참조를 위해 사용되지 않음"이라고 마킹되지만, 나머지 필드의 마킹은 변경되지 않는다.

[0092] MMCO가 3과 같을 때, 이러한 상황은 LongTermFrameIdx를 단기 참조 화상으로 할당하는 프로세스를 나타낸다(단

기 참조 화상을 장기 참조 화상으로 만듦).

[0093] 구문 요소 difference_of_pic_nums와 difference_of_view_id가 주어지면, 변수 picNumX와 viewIdX가 위에서 특정된 것처럼 얻어진다. picNumX는 "단기 참조를 위해 사용됨"이라고 마킹되고, viewIdX에 의해 특정된 뷰에서는 "존재하지 않음(non-existing)"이라고 마킹되지 않는 프레임이나 보충 참조 필드 쌍 또는 쌍으로 되지 않은 참조 필드를 가리킨다.

[0094] long_term_frame_idx와 같은 LongTermFrameIdx가 장기 참조 프레임이나 장기 보충 참조 필드 쌍에 이미 할당된 경우에는, 그러한 프레임 또는 보충 필드 쌍, 및 그러한 필드 모두는 "참조를 위해 사용되지 않음"이라고 마킹된다.

[0095] LongTermFrameIdx가 이미 쌍으로 되지 않은 참조 필드에 할당되고, 그 필드가 picNumX에 의해 특정된 화상의 보충 필드가 아닌 경우에는, 그 필드가 "참조를 위해 사용되지 않음"이라고 마킹된다.

[0096] field_pic_flag에 따라, LongTermFrameIdx의 값은, 다음과 같이 화상을 "단기 참조를 위해 사용됨"으로부터 "장기 참조를 위해 사용됨"으로 마킹하기 위해 사용된다.

[0097] - field_pic_flag가 0과 같다면, 단기 참조 프레임이나 viewIdX에 의해 특정된 뷰에서 picNumX에 의해 특정된 단기 보충 참조 필드 쌍, 및 그것의 필드 모두는 "단기 참조를 위해 사용됨"으로부터 "장기 참조를 위해 사용됨"으로 변경되고, long_term_frame_idx와 같은 LongTermFrameIdx가 할당된다.

[0098] - 그렇지 않으면(field_pic_flag가 1과 같다면), viewIdX에 의해 특정된 뷰에서 picNumX에 의해 특정된 단기 참조 필드의 마킹은 "단기 참조를 위해 사용됨"으로부터 "장기 참조를 위해 사용됨"으로 변경되고, long_term_frame_idx와 같은 LongTermFrameIdx가 할당된다. 그 필드가 참조 프레임이나 보충 참조 필드 쌍의 부분이고, 동일한 참조 프레임이나 보충 참조 필드 쌍의 나머지 필드 또한 "장기 참조를 위해 사용됨"이라고 마킹되면, 참조 프레임이나 보충 참조 필드 쌍 또한 "장기 참조를 위해 사용됨"이라고 마킹되고, long_term_frame_idx와 같은 LongTermFrameIdx가 할당된다.

[0099] MMCO가 4와 같을 때에는, LongTermFrameIdx 값이 max_long_term_frame_idx_plus1-1과 연관된 값보다 클 경우 "장기 참조를 위해 사용됨'으로부터 "참조를 위해 사용되지 않음"으로 참조 화상의 상태를 변경하기 위한 상황이 호출된다.

[0100] 변수인 **MaxLongTermFrameIdx**는 다음과 같이 결정된다.

[0101] max_long_term_frame_idx_plus1이 0과 같다면, MaxLongTermFrameIdx는 "어떠한 장기 프레임 인덱스들도 없음"과 같게 설정된다.

[0102] 그렇지 않으면(max_long_term_frame_idx_plus1이 0보다 크다면), MaxLongTermFrameIdx는 max_long_term_frame_idx_plus1-1과 같게 설정된다.

[0103] 4와 같은 memory_management_control_operation 명령이 장기 참조 화상들을 "참조를 위해 사용되지 않음"으로 마킹하기 위해 사용될 수 있음을 주목하라. max_long_term_frame_idx_plus1의 송신 주파수는 본 발명에서는 특정되지 않지만, 코더의 설계자에 의해 선택될 수 있다. 하지만, 인코더는 인트라 리프레시(intra refresh) 요청 메시지와 같은 에러 메시지를 수신하게 되면, 4와 같은 memory_management_control_operation 명령을 보내야 한다.

[0104] 5와 같은 MMCO는 viewIdX(위에서 유도된 것과 같은)에 의해 특정되는 뷰에서의 참조 화상들 모두가 "참조를 위해 사용되지 않음"이라고 마킹되는 상황을 나타낸다. 즉, 이러한 MMCO는 특별한 뷰(각 참조 화상을 명확하게 식별할 필요없이)에 관한 화상들 모두를 변경하는 기능을 코더에 제공한다. 이러한 타입의 기능은 현재 코딩되는 화상과 동일한 뷰인 모든 참조 화상들의 상태를 변경하기 위해 호출될 수 있다. 유사하게, 이러한 명령은 현재 코딩되는 화상과 연관된 뷰와 동일하지 않은 특별한 뷰의 참조 화상들의 상태를 변경하기 위해 호출될 수 있다.

[0105] 6과 같은 MMCO를 가진다는 것은 상태들이 "참조를 위해 사용되지 않음"으로 변경되고, MaxLongTermFrameIdx 변수가 "어떠한 장기 프레임 인덱스들도 없음"이라고 설정되는 현재 뷰와 연관된 뷰 외의 모든 뷰에서 모든 참조 화상을 가지는 상황을 나타낸다. 그 명령은 현재 코딩되는 화상에 관한 뷰와 연관되지 않은 뷰들에 관한 것인 모든 참조 화상의 종국적인 제거를 위해 DPB를 제어한다. 위에서 주목된 것처럼, 5와 같은 MMCO는 특별한 뷰와 연관된 참조 화상들의 상태를 변경하기 위한 것인데 반해, 현재의 MMCO(6과 같은)는 코딩되는 화상의 뷰와 동일하지 않은 뷰들과 연관되는 참조 화상 모두에 영향을 미친다.

[0106] MMCO가 7과 같다는 것은, 참조 화상의 상태가 "장기 참조 화상'으로부터 "단기 참조를 위해 사용됨"으로 변경되는 상황을 나타낸다. 그러한 참조 화상은 현재 코딩되는 화상과 연관된 뷰와는 상이한 뷰와 연관된다.

[0107] 도 7은 구문 요소인 difference_of_pics_nums_minus1이 제시된(구문 요소인difference_of_pic_nums를 사용하는 대신) 본 발명의 원리들의 대안적인 실시예를 나타낸다. 그러한 변화가 암시하는 것은 현재 화상의 picNumX보다 큰 picNumX를 지닌 화상을 선택할 수 없는 상황에 관한 것이다. 이러한 실시예와 연관된 MMCO들은 위에서 식별된 것과 동일하게 동작한다(표 2에서).

[0108] 도 8은 화상이 AVC 동작에서 코딩되는 시간 동안 구문 요소 명령인 slice_header_mvc_extension()을 호출하도록 화상 헤더(슬라이스 헤더와 같은) 명령들이 수정되는 본 발명의 원리들의 대안적인 실시예를 나타낸다. 즉, 이전에 제시되었던 것이 고려중인 화상과는 상이한 뷰의 참조 화상들이었던 것 대신, 멀티뷰들을 위한 MMCO 명령들이 AVC 인코딩(모든 참조 화상들의 상이한 뷰들이 고려될 수 있는) 동안 이 실시예에서 생길 수 있다.

[0109] 도 9는 구문 요소 명령인 **dec_ref_pic_marking_mvc_extension()**의 구성을 개시한다. 이러한 새로운 구문(도 8에 도시된 것과 같은 화상 헤더/슬라이스 헤더에서 부르는 것과 같은)은 적절한 difference_of_view_id 구문을 설정함으로써 현재 뷰가 아닌 뷰에 있는 화상들을 마킹하기 위해 사용된다. 코딩되는 화상과 연관된 뷰와 연관된 참조 화상들이 그것들의 메모리 상태가 변하게 허용하기 위해, difference_of_view_id 구문은 0으로 설정된다. 이러한 제안된 구문 요소는 기존의 AVC 기능을 명령인 def_ref_pic_marking을 사용하는 DPB 관리로 대체한다. dec_ref_pic_marking_mvc_extension()과 연관된 다양한 구문 요소들이 아래에 설명된다.

[0110] **mvc_adaptive_ref_pic_marking_mode_flag**는 현재 코딩되는 화상의 참조 마킹 모드 중에서 선택하기 위해 사용된다. "0"에서의 플래그는 단기 참조 화상에 DPB에서 FIFO 기초가 할당되는 슬라이딩 윈도우 참조 화상 마킹 모드를 나타낸다. "1"에서의 플래그는 참조 화상들을 "참조를 위해 사용되지 않음"으로서 마킹하고, "장기 프레임 인덱스들"을 할당하기 위해 요소들이 제공될 수 있는 적응성 참조 화상 마킹 모드를 나타낸다.

[0111] mvc_adaptive_ref_pic_marking_mode_flag는, 현재 "장기 참조를 위해 사용됨"으로서 마킹되는 프레임들, 보충 필드 쌍들, 및 쌍이 아닌 필드들의 개수가 Max와 같을 때(num_ref_frames, 1), 1과 같게 된다.

[0112] memory_management_control_operation(MMCO)은 참조 화상 마킹에 영향을 미치기 위해 적용될 제어 동작을 특정한다. MMCO 구문 요소 다음에는 MMCO의 값에 의해 특정된 동작에 필수적인 데이터가 온다. 호출된 MMCO와 연관된 값들과 제어 동작들이 표 3(아래)에 도시되어 있다. 이 실시예에서의 MMCO 구문 요소들은 그것들이 슬라이스 헤더에서 등장하는 순서대로 디코딩 프로세스에 의해 처리되고, 각 memory_management_control_operation에 관해 표현된 의미론상 제약을 받는 사항은 개별 MMCO가 처리되는 순서대로 특정 위치에서 적용된다.

[0113] memory_management_control_operation의 해석을 위해, 참조 화상이라는 용어는 다음과 같이 설명된다.

[0114] - 현재 화상이 프레임이라면, 참조 화상이라는 용어는 참조 프레임이나 보충 참조 필드 쌍을 가리킨다.

[0115] - 그렇지 않으면(현재 화상이 필드임), 참조 화상이라는 용어는 참조 필드나 참조 프레임의 필드를 가리킨다.

[0116] memory_management_control_operation은, 그것이 디코딩 프로세스에 의해 처리될 때, 특정된 참조 화상이 "단기 참조를 위해 사용됨"으로서 마킹되지 않는 한, 슬라이스 헤더에서 1과 같지 않게 된다.

[0117] memory_management_control_operation은, 그것이 디코딩 프로세스에 의해 처리될 때, 특정된 장기 화상 번호가 "장기 참조를 위해 사용됨"으로서 마킹되는 참조 화상을 가리키지 않는 한, 슬라이스 헤더에서 2와 같지 않게 된다.

[0118] memory_management_control_operation은, 그것이 디코딩 프로세스에 의해 처리될 때, 특정된 참조 화상이 "단기 참조를 위해 사용됨"으로서 마킹되지 않는 한, 슬라이스 헤더에서 3과 같지 않게 된다.

[0119] memory_management_control_operation은, 그것이 디코딩 프로세스에 의해 처리될 때, 변수인 MaxLongTermFrameIdx의 값이 "어떠한 장기 프레임 인덱스들도 없음"과 같다면, 3,5 또는 6과 같지 않게 된다.

[0120] 단기 참조 프레임의 부분이거나 단기 보충 참조 필드 쌍의 부분인 필드에 장기 프레임 인덱스를 할당하는, 3과 같은 필드 및 memory_management_control_operation 명령을 디코딩하는 것이 존재할 때, 동일한 장기 프레임의 인덱스를 동일한 프레임의 나머지 필드나 보충 참조 필드 쌍에 할당하기 위한 또 다른 memory_management_control_operation 명령이 구문 구조를 마킹하는 동일한 디코딩된 참조 화상에 존재하게 된다.

[0121] 위의 요구 조건은, 심지어 3과 같은 memory_management_control_operation에 의해 참조된 필드가 계속해서 "참조를 위해 사용되지 않음"이라고 마킹될 때에도 충족되어야 한다는 점을 주목하라(예컨대, 2와 같은 memory_management_control_operation이 그 필드로 하여금 "참조를 위해 사용되지 않음"이라고 마킹되게 하는 동일한 슬라이스 헤더에 존재할 때).

[0122] 보충 참조 필드 쌍의 제 1 필드(디코딩 순서에서의)가 1과 같은 long_term_reference_flag 또는 6과 같은 memory_management_control_operation을 포함할 때, 보충 참조 필드 쌍의 나머지 필드에 관한 디코딩된 참조 화상 마킹 구문 구조는 나머지 필드에 동일한 장기 프레임 인덱스를 할당하는 6과 같은 memory_management_control_operation 명령을 포함하게 된다.

[0123] 위의 요구 조건은, 심지어 보충 참조 필드 쌍의 제 1 필드가, 계속해서 "참조를 위해 사용되지 않음"이라고 마킹될 때에도 충족되어야 한다는 점을 주목하라(예컨대, 2와 같은 memory_management_control_operation이 제 1 필드로 하여금 "참조를 위해 사용되지 않음"이라고 마킹되게 하는 제 2 필드의 슬라이스 헤더에 존재할 때).

[0124] **difference_of_view_id**는 현재의 memory_management_control_operation이 적용 가능한 view_id를 유도하기 위해 사용된다.

[0125] **difference_of_pic_nums**는 장기 프레임 인덱스를 단기 참조 화상에 할당하거나 단기 참조 화상을 "참조를 위해 사용되지 않음"으로 마킹하기 위해 사용된다. 연관된 memory_management_control_operation이 디코딩 프로세스에 의해 처리되면, difference_of_pic_nums로부터 유도된 그 결과 화상 번호는 장기 프레임 인덱스에 이전에 할당되지 않고 "참조를 위해 사용됨"으로서 마킹된 참조 화상 중 하나에 할당된 화상 번호가 된다. 그 결과 화상 번호는 다음과 같이 제약을 받는다.

[0126] - field_pic_flag가 0과 같다면, 그 결과 화상 번호는 참조 프레임들이나 보충 참조 필드 쌍들에 할당된 화상 번호들의 세트 중 하나가 된다.

[0127] - field_pic_flag가 0과 같다면, 그 결과 화상 번호는 양 필드가 "참조를 위해 사용됨"이라고 마킹되는 보충 참조 필드 쌍이나, 양 필드가 "참조를 위해 사용됨"이라고 마킹되는 프레임에 할당된 화상 번호이어야 한다. 특히, field_pic_flag가 0과 같다면, 단일 필드가 "참조를 위해 사용됨"이라고 마킹되는 쌍으로 되지 않은 필드나 프레임의 마킹은 1과 같은 memory_management_control_operation에 의해 영향을 받을 수 없다.

[0128] - 그렇지 않으면(field_pic_flag가 1과 같다면,), 그 결과 화상 번호는 참조 필드들에 할당된 화상 번호들의 세트 중 하나가 된다.

[0129] **long_term_pic_num**은 장기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹하기 위해 사용된다(2와 같은 memory_management_control_operation으로). 연관된 memory_management_control_operation이 디코딩 프로세스에 의해 처리되면, long_term_pic_num이 현재 "장기 참조를 위해 사용됨"이라고 마킹되는 참조 화상들 중 하나에 할당된 장기 화상 번호와 같게 된다. 그 결과 장기 화상 번호는 다음과 같이 제약을 받는다.

[0130] - field_pic_flag가 0과 같다면, 그 결과 장기 화상 번호는 참조 프레임들이나 보충 참조 필드 쌍들에 할당된 장기 화상 번호들의 세트 중 하나가 된다.

[0131] - 그렇지 않으면(field_pic_flag가 1과 같다면,), 그 결과 화상 번호는 참조 필드들에 할당된 화상 번호들의 세트 중 하나가 된다.

[0132] field_pic_flag가 0과 같다면, 그 결과 장기 화상 번호는 양 필드가 "참조를 위해 사용됨"이라고 마킹되는 보충 참조 필드 쌍이나, 양 필드가 "참조를 위해 사용됨"이라고 마킹되는 프레임에 할당된 장기 화상 번호가 되어야 한다는 점을 주목하라. 특히, field_pic_flag가 0과 같을 때, 단일 필드가 "참조를 위해 사용됨"이라고 마킹되는 쌍이 아닌 필드나 프레임의 마킹은 2와 같은 memory_management_control_operation에 의해 영향을 받을 수 없다.

[0133] - 그렇지 않으면(field_pic_flag가 1과 같다면,), 그 결과 장기 화상 번호는 참조 필드들에 할당된 장기 화상 번호들의 세트 중 하나가 된다.

[0134] **long_term_frame_idx**는 장기 프레임 인덱스를 한 화상에 할당하기 위해 사용된다(3 또는 6과 같은 memory_management_control_operation으로). 연관된 memory_management_control_operation이 디코딩 프로세스에 의해 처리될 때, long_term_frame_idx의 값은 0부터 MaxLongTermFrameIdx를 포함하는 범위에 있게 된다.

[0135] 구문인 difference_of_pic_nums는, 현재 화상의 picNum보다 큰 picNum을 지닌 화상들을 선택하는 것을 허용한

다. 이는 마킹을 더 효율적이 되게 한다.

[0136] 디코딩된 참조 화상 마킹 프로세스는 상이한 MMCO 명령에 관해 아래에 설명된다.

[0137] viewId가

[0138] viewIdX = CurrViewId - (difference_of_view_id)에 의해 특정되게 한다.

[0139] 모든 MMCO 명령(아래 표 3에 도시된 것과 같은)은 viewIdX와 같이 위에서 유도된 viewId에 적용된다.

표 3

[0140]

memory_management_control_operation	메모리 관리 제어 동작
0	memory_management_control_operation 구문 요소 루프를 종료함
1	단기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹함
2	장기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹함
3	단기 참조 화상을 "장기 참조를 위해 사용됨"이라고 마킹하고, 장기 프레임 인덱스를 그것에 할당함
4	최대 장기 프레임 인덱스를 특정하고, 최대값보다 큰 장기 프레임 인덱스들을 가지는 모든 장기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹함
5	모든 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹하고, 변수인 MaxLongTermFrameIdx를 "어떠한 장기 프레임 인덱스도 없음"이라고 설정함
6	현재 화상을 "장기 참조를 위해 사용됨"이라고 마킹하고, 그것에 장기 프레임 인덱스를 할당함
7	그 자체가 "단기 참조를 위해 사용됨"이라고 마킹된 경우 이 외에 뷰에서의 장기 참조 화상을 마킹함

[0141] MMCO가 0과 같을 때, 화상 헤더들(예컨대, 슬라이스 헤더들)의 마킹이 종료된다.

[0142] MMCO가 1과 같을 때, 특별한 참조 프레임은 "단기 참조 화상"으로부터 "참조를 위해 사용되지 않음"으로 바뀌는 것과 연관된 상태를 가질 것이다.

[0143] picNumX를

[0144] picNumX = CurrPicNum - (difference_of_pic_nums)라고 특정한다.

[0145] 또한, field_pic_flag에 따라 picNumX의 값이 다음과 같이, 단기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹하기 위해 사용된다.

[0146] field_pic_flag가 0과 같으면, viewIdX에 의해 특정된 뷰에서 picNumX에 의해 특정된 단기 참조 프레임이나 단기 보충 참조 필드 쌍, 및 그것의 필드 모두는 "참조를 위해 사용되지 않음"이라고 마킹된다.

[0147] 그렇지 않으면(field_pic_flag가 1과 같다면), viewIdX에 의해 특정된 뷰에서 picNumX에 의해 특정된 단기 참조 필드는 "참조를 위해 사용되지 않음"이라고 마킹된다. 참조 필드가 참조 프레임이나 보충 참조 필드 쌍의 부분일 때, 그 프레임이나 보충 필드 쌍은 또한 "참조를 위해 사용되지 않음"이라고 마킹되지만, 나머지 필드의 마킹은 변경되지 않는다.

[0148] MMCO가 2와 같을 때, 특별한 참조 화상은 "장기 참조 화상"으로부터 "참조를 위해 사용되지 않음"으로 바뀌는 것과 연관된 상태를 가질 것이다. field_pic_flag에 따라, LongTermPicNum의 값이 다음과 같이 장기 참조 화상을 "참조를 위해 사용되지 않음"으로 마킹하기 위해 사용된다.

[0149] field_pic_flag가 0과 같다면, long_term_pic_num과 같은 LongTermPicNum을 가지는 장기 참조 프레임이나 장기 보충 참조 필드 쌍, 및 그것의 필드 모두는 "참조를 위해 사용되지 않음"이라고 마킹된다.

[0150] - 그렇지 않으면(field_pic_flag가 1과 같다면), long_term_pic_num과 같은LongTermPicNum에 의해 특정된 장기 참조 필드가 "참조를 위해 사용되지 않음"이라고 마킹된다. 그 참조 필드가 참조 프레임이나 보충 참조 필드 쌍의 부분일 때에는, 그 프레임 또는 보충 필드 쌍 또한 "참조를 위해 사용되지 않음"이라고 마킹되지만, 나머지

필드의 마킹은 변경되지 않는다.

[0151] MMCO가 3과 같을 때에는, 단기 참조 화상을 장기 참조 화상으로 할당하는 LongTermFrameIdx에 특별한 참조 프레임이 할당된다. 구문 요소인 difference_of_pic_nums와 difference_of_view_id가 주어지면, 변수인 picNumX와 viewIdX가 위에서 특정된 것처럼 얻어진다. picNumX는 viewIdX에 의해 특정된 뷰에서 "존재하지 않는"이라고 마킹되지 않고, "단기 참조를 위해 사용됨"이라고 마킹된 프레임 또는 보충 참조 필드 쌍 또는 쌍이 아닌 참조 필드를 가리키게 된다.

[0152] long_term_frame_idx와 같은 LongTermFrameIdx가 장기 참조 프레임이나 장기 보충 참조 필드 쌍에 이미 할당되는 경우에는, 그 프레임 또는 보충 필드 쌍, 및 그것의 필드 모두는 "참조를 위해 사용되지 않음"이라고 마킹된다. LongTermFrameIdx가 쌍이 아닌 참조 필드에 이미 할당되고, 그 필드가 picNumX에 의해 특정된 화상의 보충 필드가 아닌 경우에는, 그 필드가 "참조를 위해 사용되지 않음"이라고 마킹된다. field_pic_flag에 따라, LongTermFrameIdx의 값이 다음과 같이 "단기 참조를 위해 사용됨"으로부터 "장기 참조를 위해 사용됨"이라고 변경되는 화상을 마킹하기 위해 사용된다.

[0153] - field_pic_flag가 0과 같다면, viewIdX에 의해 특정된 뷰에서 picNumX에 의해 특정된 단기 참조 프레임이나 단기 보충 참조 필드 쌍의 마킹은 "단기 참조를 위해 사용됨"으로부터 "장기 참조를 위해 사용됨"이라고 변경되고, long_term_frame_idx와 같은 LongTermFrameIdx가 할당된다.

[0154] - 그렇지 않으면(field_pic_flag가 1과 같다면), viewIdX에 의해 특정된 뷰에서 picNumX에 의해 특정된 단기 참조 필드의 마킹은 "단기 참조를 위해 사용됨"으로부터 "장기 참조를 위해 사용됨"이라고 변경되고, long_term_frame_idx와 같은 LongTermFrameIdx가 할당된다. 그 필드가 참조 프레임이나 보충 참조 필드 쌍의 부분이고, 동일한 참조 프레임이나 보충 참조 필드 쌍의 나머지 필드가 또한 "장기 참조를 위해 사용됨"이라고 마킹되면, 그 참조 프레임이나 보충 참조 필드 쌍 또한 "장기 참조를 위해 사용됨"이라고 마킹되고, long_term_frame_idx와 같은 LongTermFrameIdx가 할당된다.

[0155] MMCO가 4와 같을 때에는, 최대 장기 프레임 인덱스 값이 특정되고, 이를 통해 장기 참조 프레임이라고 식별되고 최대값보다 큰 프레임 인덱스들을 가지는 모든 참조 프레임은 "참조를 위해 사용되지 않음"이라고 분류된다. 특히(함수 호출의 명명법 내에서의), LongTermFrameIdx가 max_long_term_frame_idx_plus1-1보다 크고, "장기 참조를 위해 사용됨"이라고 마킹되는 모든 화상들은 "참조를 위해 사용되지 않음"이라고 마킹된다.

[0156] 변수인 **MaxLongTermFrameIdx**는 다음과 같이 유도된다.

[0157] - max_long_term_frame_idx_plus1이 0과 같다면, MaxLongTermFrameIdx는 "어떠한 장기 프레임 인덱스도 없음"과 같게 설정된다.

[0158] - 그렇지 않으면(max_long_term_idx_plus1이 0보다 크다면), MaxLongTermFrameIdx는 max_long_term_frame_idx_plus1-1과 같게 설정된다. 이러한 MMCO는 장기 참조 화상을 "참조를 위해 사용되지 않음"이라고 마킹하기 위해 사용될 수 있다는 점이 주목된다. max_long_term_frame_idx_plus1을 송신하는 주파수는 본 발명을 따르는 코더의 설계자에 의해 결정되어야 한다. 하지만, 그 코더는 인트라 리프레시 요청 메시지와 같은 에러 메시지를 수신하게 되면, 4와 같은 memory_management_control_operation 명령을 보내야 한다.

[0159] MMCO가 5와 같을 때에는, 코더가 viewIdX(특정 뷰)에 의해 특정되는 뷰에서의 참조 화상들 모두를 "참조를 위해 사용되지 않음"이라고 마킹하고, 변수인 MaxLongTermFrameIdx를 "어떠한 장기 프레임 인덱스들도 없음"과 같게 설정한다. 이는 특별한 뷰로 식별된 참조 화상들이 "참조를 위해 사용되지 않음"으로 설정됨을 의미하고, 이 경우 그러한 참조 화상은 그 전에는 "장기(long-term)"라고 마킹되었다.

[0160] MMCO가 6과 같을 때에는, 코딩되는 현재 화상이 "장기간 사용됨"이라고 마킹되고, 장기 프레임 인덱스가 그 화상에 할당된다. 특히, long_term_frame_idx와 같은 변수 LongTermFrameIdx가 이미 장기 참조 프레임이나 장기 보충 참조 필드 쌍에 할당될 때, 그 프레임이나 보충 필드 쌍, 및 그것의 필드 모두는 "참조를 위해 사용되지 않음"이라고 마킹된다. LongTermFrameIdx가 쌍이 아닌 참조 필드에 할당되고, 그 필드가 현재 화상의 보충 필드가 아닐 때에는, 그 필드가 "참조를 위해 사용되지 않음"이라고 마킹된다.

[0161] 현재 화상은 "장기 참조를 위해 사용됨"이라고 마킹되고, long_term_frame_idx와 같은 LongTermFrameIdx가 할당된다.

[0162] field_pic_flag가 0과 같을 때에는, 그것의 필드 모두가 또한 "장기 참조를 위해 사용됨"이라고 마킹되고,

long_term_frame_idx와 같은 LongTermFrameIdx가 할당된다.

[0163] field_pic_flag가 1과 같고, 현재 화상이 보충 참조 필드 쌍의 제 2 필드(디코딩 순서로)이며, 보충 참조 필드 쌍의 제 1 필드가 또한 현재 "장기 참조를 위해 사용됨"이라고 마킹된다면, 보충 참조 필드 쌍 역시 "장기 참조를 위해 사용됨"이라고 마킹되고, long_term_frame_idx와 같은 LongTermFrameIdx가 할당된다.

[0164] 현재 디코딩된 참조 화상을 마킹한 후, "참조를 위해 사용됨"이라고 마킹된 적어도 하나의 필드를 지닌 프레임들의 총수, "참조를 위해 사용됨"이라고 마킹된 적어도 하나의 필드를 지닌 보충 필드 쌍들의 개수, "참조를 위해 사용됨"이라고 마킹된 쌍이 아닌 필드들의 개수를 더한 것은 Max(num_ref_frames,1)보다 크지 않게 된다. 일부 상황 하에서는, 위 진술이 6과 같은 memory_management_control_operation 구문 요소가 1,2 또는 4와 같은 memory_management_control_operation 구문 요소에 관하여 디코딩된 참조 화상 마킹 구문에 등장할 수 있는 순서에 제약을 둘 수 있다는 점을 주목하라.

[0165] MMCO가 7과 같을 때에는, viewIdX에 의해 식별된 뷰에서 long_term_pic_num에 의해 식별된 장기 참조 화상은 "단기 참조를 위해 사용됨"이라고 마킹된다. 이는 특별한 프레임(pic number에 의해 식별된)과 특정된 뷰가 "장기 참조 화상"으로부터 "단기 참조 화상"으로 변경되는 상태를 가짐을 의미한다.

[0166] 본 발명의 대안적인 실시예가 도 10에 도시된 구문 요소에 따라 개시되어 있고, 이 경우 **difference_of_pic_nums** 또는 **difference_of_pic_nums_minus1**이 **difference_of_view_id**의 값에 기초하여 송신된다. 이러한 해결책은 도 8에 관해 설명된 difference_of_pic_nums 구문 요소를 사용하는 대신, 시간상 경우에 관해서만 MMCO 명령들을 사용할 때 제안된다.

[0167] 도 11은 본 발명의 원리들에 따른 일반화된 참조 화상 마킹 방법의 블록도(1100)를 개시한다.

[0168] 단계(1105)는 화상을 코딩하는 일반화된 개념을 나타낸다(코딩 동작은 통상적으로 움직이는 화상들에 기초한 비디오의 그룹으로부터의 화상을 인코딩한다). 이러한 동작은 또한 코딩된 화상의 디코딩을 나타낼 수 있다.

[0169] 하지만, 현재 단계에서는 코딩되는 화상이 멀티-뷰 비디오 코딩 시스템에서 사용되는 복수의 뷰로부터의 특별한 뷰와 연관된다. 바람직한 실시예는 AVC 코더의 상황 내에서의 MVC 비디오 표준에 개시된 원리들을 사용하지만, 다른 멀티-뷰 비디오 표준들이 사용될 수 있다는 것을 알아야 한다. 중요한 것은, 코딩된 화상이 화상 ID 번호와 뷰 ID 번호와 연관된다는 점이다. 화상 ID는 코딩된 화상들의 시퀀스 내에서의 코딩된 화상들 번호를 나타낸다. 코딩된 화상은 또한 코딩된 화상이 연관되는 뷰에 대응하는 뷰 ID 번호(1과 n 사이에 있는 것으로, n=뷰들의 총 개수)를 가지게 된다. 예컨대, 뷰"2"와 연관되는 코딩된 화상은 "2"의 뷰 ID 번호를 가지는 것으로 알려진다.

[0170] 단계(1110)에서는, 코딩된 화상이 메모리 디바이스(DPB와 같은)에 저장되고, 코딩된 화상에는 메모리 상태가 할당된다. 코딩된 화상은 그 화상이 참조 화상으로서 사용될 수 있도록 저장된다. 전술한 바와 같이, 코딩된 화상은 그것과 관련된 적어도 3개의 상이한 메모리 상태를 가질 수 있다.

[0171] "장기 참조 화상"은 코딩된 화상이 참조 화상으로서 저장될 때를 나타낸다. 장기 참조 화상으로서 지정되는 코딩된 화상에는 인덱스 번호가 할당된다(장기 화상 인덱스에서). 이 화상은 그러한 화상이 미래의 화상들을 코딩할 때 참조 화상으로서 사용될 수 있도록, 당분간 유지된다.

[0172] "단기 참조 화상"은 코딩된 화상이 참조 화상으로서 짧은 시간 동안 유지될 것임을 나타낸다. 이러한 경우, 참조 화상은 공간이 요구될 때 메모리 디바이스(DPB) 밖으로 이동하게 된다.

[0173] "참조로서 사용되지 않음"은 코딩된 화상이 참조 화상으로서 사용되지 않을 것임을 나타낸다. 이 경우, DPB는 공간이 요구될 때 (LIFO/FIFO 이용) 또는 DPB로부터 직접 지워질 때, 참조 화상을 제거할 수 있다.

[0174] 참조 화상으로서 사용되지 않는 것으로 지정된 화상이 직접 삭제될 수 있고 단계(1110)와 단계(1115) 동안 DPB에 결코 저장되지 않는 것이 가능하다.

[0175] 단계(1120)에서는, 제 2 비디오 화상이 코딩된다. 이때, 비디오 화상은 메모리 관리 명령 동작(다른 실시예들과 설명된 것과 같은)으로 마킹되고, 이 경우 MMCO는 DPB와 같은 참조 화상들을 저장하기 위해 사용된 메모리 디바이스에서 저장된 참조 화상에 영향을 미친다.

[0176] 전술한 바와 같이, 참조 화상과 연관된 메모리 상태를 어떻게 변경할지를 결정하기 위해 상이한 MMCO가 이용될 수 있다. 그러한 변경은 현재 코딩되는 화상이 저장된 참조 화상과 같은/상이한 뷰를 가지는지에 기초하여 이루어질 수 있다(예컨대, 동일한 뷰 ID를 갖는 모든 화상은 동시에 변경되는 데, 이는 포괄적 변경을 나타낸다).

화상과 연관된 메모리 상태의 변경은 직접 행해질 수 있고, 이 경우 특별히 저장된 참조 화상이 식별되고, MMCO는 그 화상의 메모리 상태가 변경될 것임을 특정한다(예컨대, "장기 참조", "단기 참조"와 "참조로서 사용되지 않음"의 메모리 상태 사이의 변경으로서, 이는 국부적인 변경을 나타낸다).

[0177] 또한, 전술한 바와 같이 본 발명의 다양한 구현예는 현재 코딩되는 화상의 뷰 ID가, 화상들이 상이한 뷰 ID(상이한 뷰들에 걸쳐)를 지닌 참조 화상 또는 코딩되는 화상과 동일한 뷰 ID(특정 뷰에 관한)를 가지는 참조 화상에 관해서만 변경될 수 있는지에 영향을 미치는 상황을 허용한다.

[0178] 단계(1125)는 메모리 저장 디바이스가 참조 화상과 연관된 메모리 상태를 구현하는 참조 화상을 저장하는 MMCO 명령을 구현하는 것이다. 이들 타입의 동작 또한 위에서 설명된다.

[0179] 본 발명의 원리의 이들 및 다른 특징과 장점은, 당업자가 본 발명의 가르침에 기초하여 쉽게 확인할 수 있다. 본 발명의 원리의 가르침은 다양한 형태의 하드웨어, 소프트웨어, 펌웨어, 특별한 목적의 프로세서, 또는 이들의 결합으로 구현될 수 있음이 이해되어야 한다.

[0180] 가장 바람직하게는, 본 발명의 원리의 가르침은 하드웨어와 소프트웨어의 결합으로서 구현된다. 게다가, 그러한 소프트웨어는 프로그램 저장 유닛 상에 확실하게 구현된 응용 프로그램으로서 구현될 수 있다. 이 응용 프로그램은 임의의 적합한 아키텍처를 포함하는 기계에 업로드되거나 그러한 기계에 의해 실행될 수 있다. 바람직하게, 그러한 기계는 하나 이상의 중앙 처리 장치(CPU), 랜덤 액세스 메모리(RAM), 입/출력(I/O) 인터페이스들과 같은 하드웨어를 가지는 컴퓨터 플랫폼에서 구현된다. 이 컴퓨터 플랫폼은 또한 운영 체제와 마이크로인스트럭션 코드를 포함할 수 있다. 본 명세서에서 설명된 다양한 프로세스들과 기능들은 마이크로인스트럭션 코드의 부분이거나 응용 프로그램의 부분, 또는 이들의 결합일 수 있고, 이들은 CPU에 의해 실행될 수 있다. 또한, 추가 데이터 저장 유닛이나 프린팅 유닛과 같은 다양한 다른 주변 유닛이 컴퓨터 플랫폼에 연결될 수 있다.

[0181] 첨부 도면에 도시된 구성 시스템 성분과 방법의 일부가 바람직하게는 소프트웨어로 구현되기 때문에, 시스템 성분들 또는 프로세스 기능 블록들 사이의 실제 연결은 본 발명의 원리가 프로그래밍되는 방식에 따라 상이할 수 있다는 점이 또한 이해되어야 한다. 본 명세서에서의 가르침으로, 당업자라면 본 발명의 원리의 이들 및 유사한 구현예 또는 구성예를 예측할 수 있다.

[0182] 비록, 예시적인 실시예가 첨부 도면을 참조하여 본 명세서에서 설명되었지만, 본 발명의 원리는 이들 정밀한 실시예에 제한되지 않고, 다양한 변경예와 수정예가 본 발명의 원리의 범주와 취지로부터 벗어나지 않으면서 당업자에 의해 실행될 수 있음이 이해되어야 한다. 모든 그러한 변경예와 수정예는 첨부된 청구항에서 전개된 본 발명의 원리들의 범주 내에 포함되는 것으로 의도된다.

산업상 이용 가능성

[0183] 전술한 바와 같이, 본 발명은 동화상 분야, 특히 멀티-뷰 비디오 코딩과 연관된 동화상의 메모리 유지를 필요로 하는 분야에 이용 가능하다.

도면의 간단한 설명

[0011] 도 1은 비디오 화상들이 도면에 나타난 방식대로 참조 화상들을 사용하여 코딩되는, 상이한 시각에서의 비디오 화상 뷰들을 멀티뷰 코딩하는 예시적인 실시예를 도시하는 도면.

[0012] 도 2는 본 발명의 원리들에 따른 비디오 코더의 예시적인 실시예를 도시하는 도면.

[0013] 도 3은 본 발명의 원리들에 따라 사용된 구문 요소인 dec_ref_pic_marking()을 위한 의사 코드의 일 실시예를 도시하는 도면.

[0014] 도 4는 본 발명의 원리들에 따라 사용된 구문 요소인 seq_parameter_set_mvc_extension()을 위한 의사 코드를 도시하는 도면.

[0015] 도 5는 본 발명의 원리들에 따라 사용된 구문 요소인 dec_ref_pic_marking_mvc_extension()을 위한 의사 코드의 일 실시예를 도시하는 도면.

[0016] 도 6은 본 발명의 원리들에 따라 사용된 샘플 화상 헤더의 일 실시예를 도시하는 도면.

[0017] 도 7은 본 발명의 원리들에 따라 사용된 구문 요소인 dec_ref_pic_marking_mvc_extension()을 위한 의사 코드의 일 실시예를 도시하는 도면.

[0018] 도 8은 본 발명의 원리들에 따라 사용된 샘플 화상 헤더의 일 실시예를 도시하는 도면.

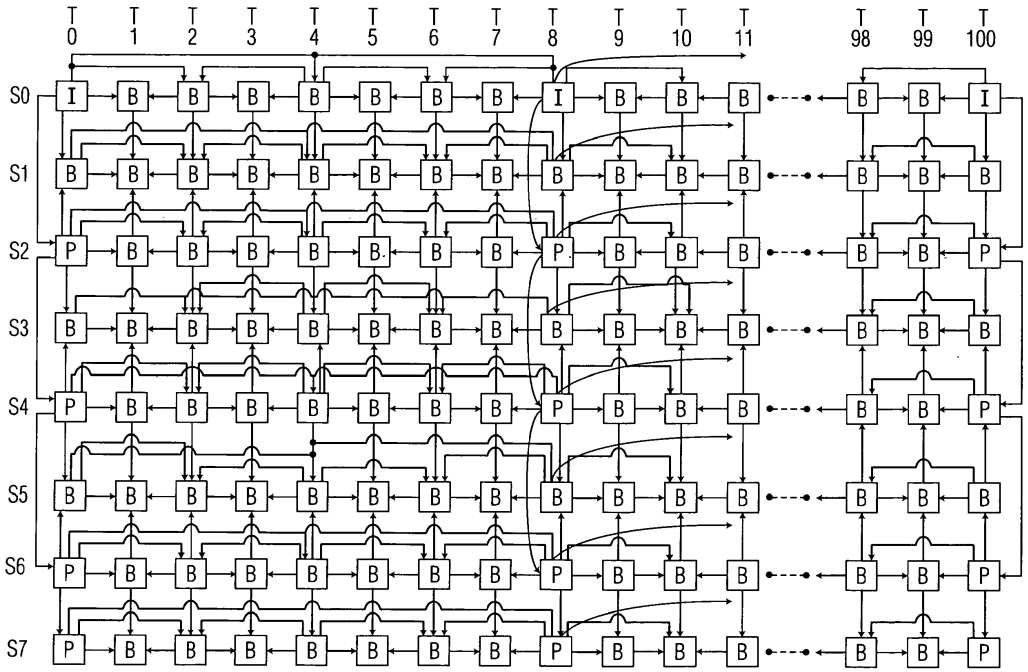
[0019] 도 9는 본 발명의 원리들에 따라 사용된 구문 요소인 dec_ref_pic_marking_mvc_extension()을 위한 의사 코드의 일 실시예를 도시하는 도면.

[0020] 도 10은 본 발명의 원리들에 따라 사용된 구문 요소인 dec_ref_pic_marking_mvc_extension()을 위한 의사 코드의 일 실시예를 도시하는 도면.

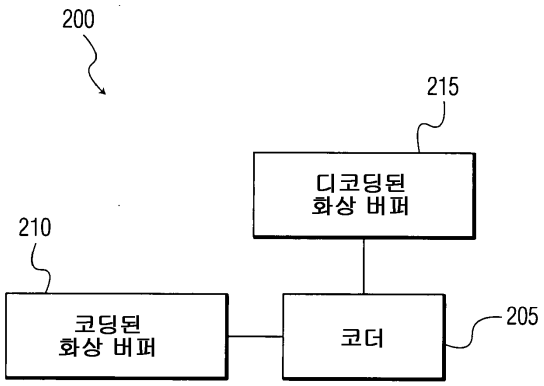
[0021] 도 11은 본 발명의 원리들에 따를 화상 마킹 방법의 일 실시예의 블록도.

도면

도면1



도면2



도면3

dec_ref_pic_marking( ) {	C	설명자
if(nal_unit_type = = 5 | | nal_unit_type = = 21) { /* nal_unit_type 21 is specified in Annex Error! Reference source not found. */		
no_output_of_prior_pics_flag	215	u(1)
long_term_reference_flag	215	u(1)
} else {		
adaptive_ref_pic_marking_mode_flag	215	u(1)
if(adaptive_ref_pic_marking_mode_flag)		
do {		
memory_management_control_operation	215	ue(v)
if(memory_management_control_operation = = 1 | | memory_management_control_operation = = 3)		
difference_of_pic_nums_minus]	215	ue(v)
if(memory_management_control_operation = = 2)		
long_term_pic_num	215	ue(v)
if(memory_management_control_operation = = 3 | | memory_management_control_operation = = 6)		
long_term_frame_idx	215	ue(v)
if(memory_management_control_operation = = 4)		
max_long_term_frame_idx_plus1	215	ue(v)
} while(memory_management_control_operation != 0)		
}		
}		

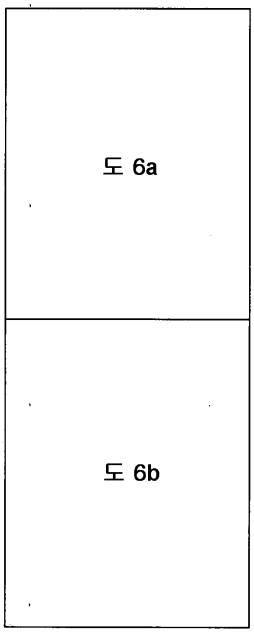
도면4

seq_parameter_set_mvc_extension( ) {	C	설명자
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1 ; i++) {		
num_multiview_refs_for_list0[i]		ue(v)
num_multiview_refs_for_list1[i]		ue(v)
for(j = 0; j < num_multiview_refs_for_list0; j++) {		
anchor_reference_view_for_list_0[i][j]		u(10)
non_anchor_reference_view_for_list_0[i][j]		u(10)
}		
for(j = 0; j < num_multiview_refs_for_list1; j++) {		
anchor_reference_view_for_list_1[i][j]		u(10)
non_anchor_reference_view_for_list_1[i][j]		u(10)
}		
}		

도면5

dec_ref_pic_marking_mvc_extension( ) {	C	설명자
mvc_adaptive_ref_pic_marking_mode_flag	215	u(1)
If(mvc_adaptive_ref_pic_marking_mode_flag)		
do {		
memory_management_control_operation	215	ue(v)
if(memory_management_control_operation = = 1 || memory_management_control_operation = = 3 || memory_management_control_operation = = 7) {		
difference_of_view_id	215	se(v)
difference_of_pic_nums	215	se(v)
}		
if(memory_management_control_operation = = 2 || memory_management_control_operation = = 7)		
long_term_pic_num	215	ue(v)
if(memory_management_control_operation = = 4)		
max_long_term_frame_idx_plus1	215	ue(v)
if(memory_management_control_operation = = 5)		
difference_of_view_id_minus1	215	ue(v)
if(memory_management_control_operation = = 3)		
long_term_frame_idx	215	ue(v)
} while(memory_management_control_operation != 0)		
dec_ref_pic_marking( )		
}		

도면6



도면6a

slice_header_mvc_extension( ) {	C	설명자
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		
if(nal_unit_type = = 21)		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type = = 0) {		
pic_order_cnt_lsb	2	u(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt_bottom	2	se(v)
}		
if(pic_order_cnt_type = = 1 && !delta_pic_order_always_zero_flag) {		
delta_pic_order_cnt [0]	2	se(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt [1]	2	se(v)
}		
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	2	ue(v)
if(slice_type = = EB)		
direct_spatial_mv_pred_flag	2	u(1)
if(slice_type = = EB | slice_type = = EB) {		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag) {		
num_ref_idx_10_active_minus1	2	ue(v)
if(slice_type = = EB)		
num_ref_idx_l1_active_minus1	2	ue(v)
}		
}		

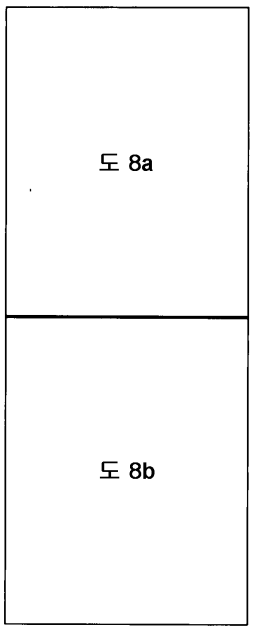
도면6b

ref_pic_list_reordering( )	2	
if((weighted_pred_flag && (slice_type = = EP)) | | (weighted_bipred_idc = = 1 && slice_type = = EB))		
pred_weight_table( )	2	
if(nal_ref_idc != 0)		
dec_ref_pic_marking_mvc_extension ( )	*2*	
if(entropy_coding_mode_flag && slice_type != I && slice_type != SI)		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)
if(slice_type = = SP | | slice_type = = SI) {		
if(slice_type = = SP)		
sp_for_switch_flag	2	u(1)
slice_qs_delta	2	se(v)
}		
if(deblocking_filter_control_present_flag) {		
disable_deblocking_filter_idc	2	ue(v)
if(disable_deblocking_filter_idc != 1) {		
slice_alpha_c0_offset_div2	2	se(v)
slice_beta_offset_div2	2	se(v)
}		
}		
if(num_slice_groups_minus1 > 0 && slice_group_map_type > = 3 && slice_group_map_type < = 5)		
slice_group_change_cycle	2	u(v)
}		

도면7

dec_ref_pic_marking_mvc_extension( ) {	C	설명자
mvc_adaptive_ref_pic_marking_mode_flag	215	u(1)
if(mvc_adaptive_ref_pic_marking_mode_flag)		
do {		
memory_management_control_operation	215	ue(v)
if(memory_management_control_operation = = 1 | | memory_management_control_operation = = 3) {		
difference_of_view_id	215	se(v)
difference_of_pic_nums_minus1	215	se(v)
}		
if(memory_management_control_operation = = 2)		
long_term_pic_num	215	ue(v)
if(memory_management_control_operation = = 4)		
max_long_term_frame_idx_plus1	215	ue(v)
if(memory_management_control_operation = = 5)		
difference_of_view_id	215	se(v)
if(memory_management_control_operation = = 3)		
long_term_frame_idx	215	ue(v)
} while(memory_management_control_operation != 0)		
dec_ref_pic_marking( )		
}		

도면8



도면8a

slice_header_mvc_extension( ) {	C	설명자
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		
if(nal_unit_type = = 21)		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type = = 0) {		
pic_order_cnt_lsb	2	u(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt_bottom	2	se(v)
}		
if(pic_order_cnt_type = = 1 && !delta_pic_order_always_zero_flag) {		
delta_pic_order_cnt [0]	2	se(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt [1]	2	se(v)
}		
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	2	ue(v)
if(slice_type = = EB)		
direct_spatial_mv_pred_flag	2	u(1)
if(slice_type = = EP | | slice_type = = EB) {		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag) {		
num_ref_idx_10_active_minus1	2	ue(v)
if(slice_type = = EB)		
num_ref_idx_l1_active_minus1	2	ue(v)
}		
}		

도면8b

ref_pic_list_reordering( )	2	
if((weighted_pred_flag && (slice_type = = EP)) | | (weighted_bipred_idc = = 1 && slice_type = = EB))		
pred_weight_table( )	2	
if(nal_ref_idc != 0)		
dec_ref_pic_marking_mvc_extension ( )	*2*	
if(entropy_coding_mode_flag && slice_type != I && slice_type != SI)		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)
if(slice_type = = SP | | slice_type = = SI) {		
if(slice_type = = SP)		
sp_for_switch_flag	2	u(1)
slice_qs_delta	2	se(v)
}		
if(deblocking_filter_control_present_flag) {		
disable_deblocking_filter_idc	2	ue(v)
if(disable_deblocking_filter_idc != 1) {		
slice_alpha_c0_offset_div2	2	se(v)
slice_beta_offset_div2	2	se(v)
}		
}		
if(num_slice_groups_minus1 > 0 && slice_group_map_type > = 3 && slice_group_map_type < = 5)		
slice_group_change_cycle	2	u(v)
}		

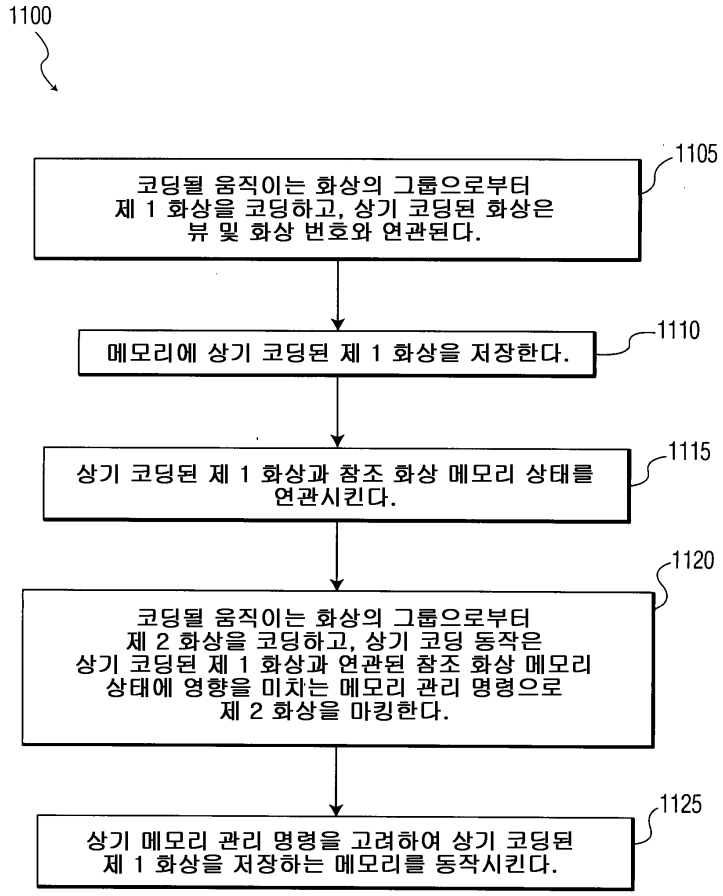
도면9

dec_ref_pic_marking_mvc_extension( ) {	C	설명자
if(nal_unit_type = = 5 | | nal_unit_type = = 21) {		
no_output_of_prior_pics_flag	215	u(1)
long_term_reference_flag	215	u(1)
} else {		
mvc_adaptive_ref_pic_marking_mode_flag	215	u(1)
if(mvc_adaptive_ref_pic_marking_mode_flag)		
do {		
if(memory_management_control_operation != 0 | | memory_management_control_operation ! = 6)		
difference_of_view_id	215	se(v)
memory_management_control_operation	215	ue(v)
if(memory_management_control_operation = = 1 | | memory_management_control_operation = = 3) {		
difference_of_pic_nums	215	se(v)
}		
if(memory_management_control_operation = = 2 | | memory_management_control_operation = = 7)		
long_term_pic_num	215	ue(v)
if(memory_management_control_operation = = 3 | | memory_management_control_operation = = 6) {		
long_term_frame_idx	215	ue(v)
if(memory_management_control_operation = = 4)		
max_long_term_frame_idx_plus1	215	ue(v)
} while(memory_managment_control_operation != 0)		
}		
}		

도면10

dec_ref_pic_marking_mvc_extension( ) {	C	설명자
mvc_adaptive_ref_pic_marking_mode_flag	215	u(1)
if(mvc_adaptive_ref_pic_marking_mode_flag)		
do {		
memory_management_control_operation	215	ue(v)
if(memory_management_control_operation = = 1 | | memory_management_control_operation = = 3) {		
difference_of_view_id	215	se(v)
difference_of_pic_nums_minus1	215	se(v)
}		
if(memory_management_control_operation = = 2)		
long_term_pic_num	215	ue(v)
if(memory_management_control_operation = = 4)		
max_long_term_frame_idx_plus1	215	ue(v)
if(memory_management_control_operation = = 5)		
difference_of_view_id	215	se(v)
if(memory_management_control_operation = = 3)		
long_term_frame_idx	215	ue(v)
} while(memory_management_control_operation != 0)		
dec_ref_pic_marking( )		
}		

도면11


1100
코딩될 움직이는 화상의 그룹으로부터 제 1 화상을 코딩하고, 상기 코딩된 화상은 뷰 및 화상 번호와 연관된다.
1105
메모리에 상기 코딩된 제 1 화상을 저장한다.
1110
상기 코딩된 제 1 화상과 참조 화상 메모리 상태를 연관시킨다.
1115
코딩될 움직이는 화상의 그룹으로부터 제 2 화상을 코딩하고, 상기 코딩 동작은 상기 코딩된 제 1 화상과 연관된 참조 화상 메모리 상태에 영향을 미치는 메모리 관리 명령으로 제 2 화상을 마킹한다.
1120
상기 메모리 관리 명령을 고려하여 상기 코딩된 제 1 화상을 저장하는 메모리를 동작시킨다.
1125