



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2014년04월17일

(11) 등록번호 10-1385808

(24) 등록일자 2014년04월09일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/107 (2014.01) *HO4N 19/46* (2014.01)
HO4N 13/00 (2006.01)
- (21) 출원번호 10-2009-7007764
- (22) 출원일자(국제) 2007년10월16일
 심사청구일자 2012년10월16일
- (85) 번역문제출일자 2009년04월15일
- (65) 공개번호 10-2009-0077918
- (43) 공개일자 2009년07월16일
- (86) 국제출원번호 PCT/US2007/022082
- (87) 국제공개번호 WO 2008/048605
 국제공개일자 2008년04월24일
- (30) 우선권주장
 60/851,953 2006년10월16일 미국(US)

(56) 선행기술조사문현

KR1020040033275 A

KR1020060025554 A

US20100322311 A1

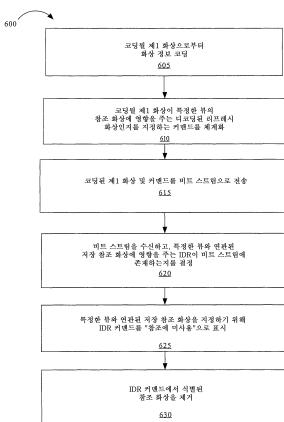
전체 청구항 수 : 총 10 항

심사관 : 권오성

(54) 발명의 명칭 비디오 동작 동안 NAL 유닛을 이용하여 동시 디코딩 리프레시를 시그널링하는 방법

(57) 요약

메모리 관리 방안은 멀티뷰 코딩된 비디오 화상 시스템과 연관된 참조 화상을 저장하는데 이용되는 메모리의 관리에 대해 규정한다. 동시 리프레시 디코드 화상의 코딩된 화상 정보와 함께 수신된 정보에 기반하여(620), 특정한 뷰와 연관된 참조 화상을 삭제하는 결정이 내려지고(625), 이러한 화상은 메모리로부터 삭제된다(630).

대 표 도 - 도6

특허청구의 범위

청구항 1

멀티뷰 동화상 시퀀스(multiview sequence of moving picture)에 대응하는 비디오 데이터를 코딩하는 방법으로서 - 인코딩될 화상은 시간과 뷰 상호관련성(a temporal and a view interrelationship) 모두를 갖고, 멀티뷰 시퀀스의 각각의 뷰는 각각의 오브젝트/장면을 나타냄 -,

비디오 화상에 대응하는 비디오 정보를 코딩하는 단계(605) - 상기 비디오 화상은 상기 멀티뷰 시퀀스의 적어도 제1 뷰에 대응하고 또한 상기 제1 뷰와는 다른, 상기 멀티뷰 시퀀스의 제2 뷰의 적어도 하나의 참조 화상(reference picture)에 의존하여 코딩될 수 있고, 상기 하나의 참조 화상은 다른 참조 화상들과 함께 디코딩된 화상 버퍼에 저장됨 -; 및

상기 멀티뷰 시퀀스의 상기 제2 뷰의 상기 적어도 하나의 화상을 포함하는 상기 제2 뷰의 모든 저장된 참조 화상들이, 상기 디코딩된 화상 버퍼에 저장된 모든 다른 참조 화상들을 삭제하지 않고 상기 디코딩된 화상 버퍼에서 추후에 삭제될 것인지 여부를 표시하는 상기 제1 뷰와 연관된 정보를 생성하고 전송하는 단계(610)

를 포함하는 비디오 데이터 코딩 방법.

청구항 2

제1항에 있어서,

상기 코딩된 비디오 정보를 전송하는 단계를 더 포함하고, 상기 코딩된 비디오 정보는 제1 NAL(network abstraction layer) 유닛으로 전송되고, 상기 제1 뷰와 연관된 상기 생성된 정보는 제2 NAL 유닛으로 전송되는 비디오 데이터 코딩 방법.

청구항 3

제2항에 있어서,

상기 제1 NAL 유닛은 연관된 NAL 유닛이고, 제2 NAL은 서픽스 NAL 유닛인 비디오 데이터 코딩 방법.

청구항 4

제2항에 있어서,

상기 생성하고 전송하는 단계는 상기 코딩된 비디오 정보를 전송 패킷의 페이로드로 전송하고, 상기 제1 뷰와 연관된 상기 생성된 정보는 제2 뷰의 상기 저장된 참조 화상들의 모두가 삭제될 것인지를 표시하는 비디오 데이터 코딩 방법.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 제1 뷰와 연관된 상기 생성된 정보는 제2 뷰의 저장된 참조 화상이 "참조를 위해 미사용(unused for reference)"으로서 마킹되고 삭제될 것인지 여부를 표시하는 비디오 데이터 코딩 방법.

청구항 6

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 제1 뷰 및 제2 뷰와는 상이한 제3 뷰의 적어도 하나의 저장된 참조 화상이 삭제되어야 하는지 여부를 표시하는 추가 정보가 생성 및 전송되고, 상기 제3 뷰의 참조 화상들만이 상기 추가의 정보와 함께 삭제되는 비디오 데이터 코딩 방법.

청구항 7

뷰들의 멀티뷰 시퀀스의 코딩된 비디오 화상들을 나타내는 수신된 비트 스트림을 디코딩하는 방법으로서 - 디코딩될 화상들은 시간과 뷰 상호관련성 모두를 갖고, 멀티뷰 시퀀스의 각각의 뷰는 각각의 오브젝트/장면을 나타냄 -,

상기 멀티뷰 시퀀스의 제1 뷰와 연관되고 또한 상기 제1 뷰와는 다른, 상기 멀티뷰 시퀀스의 제2 뷰의 적어도 하나의 참조 화상에 의존하여 코딩될 수 있는 코딩된 비디오 정보를 디코딩하기 위해 상기 비트 스트림을 프로세싱하는 단계;

다른 참조 화상들과 함께 상기 제2 뷰의 상기 참조 화상을 디코딩된 화상 버퍼에 저장하는 단계;

상기 멀티뷰 시퀀스의 상기 제2 뷰의 상기 적어도 하나의 참조 화상을 포함하는 상기 제2 뷰의 모든 저장된 참조 화상들의 삭제를 필요로 하는 정보가 상기 비트 스트림에 존재하는지 여부를 판정하는 단계(625);

상기 제2 뷰의 상기 저장된 참조 화상의 모두를 참조를 위해 미사용으로서 마킹하는 단계; 및

상기 디코딩된 화상 버퍼에서 모든 다른 참조 화상들을 삭제하지 않고 상기 제2 뷰의 상기 저장된 참조 화상의 모두를 추후에 삭제하는 단계

를 포함하는 비트스트림 디코딩 방법.

청구항 8

제7항에 있어서,

메모리로부터 제3 뷰와 연관된 적어도 하나의 참조 화상을 상기 디코딩된 화상 버퍼에 보유하는 단계를 더 포함하며,

상기 적어도 제1 뷰, 제2 뷰, 및 제3 뷰는 상이한 뷰를 나타내는 비트스트림 디코딩 방법.

청구항 9

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 정보는 코딩된 화상이 동시 리프레시 디코드 화상(instantaneous refresh decode picture)이고 상기 코딩된 화상이 동시 리프레시 디코드 화상인 것을 표시하는 비디오 데이터 코딩 방법.

청구항 10

제7항 또는 제8항에 있어서,

상기 정보는 상기 코딩된 화상이 동시 리프레시 디코드 화상이고 상기 코딩된 화상이 동시 리프레시 화상인 것을 표시하는 비트스트림 디코딩 방법.

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

작제

명세서

기술분야

[0001] 관련 출원에 대한 상호참조

본 출원은 2006년 10월 16일에 출원된 미국 가출원 제60/851,953호의 우선권을 주장하며, 상기 출원은 본 명세서에 참조 인용된다.

본 발명은 동화상(moving picture)에 관한 것으로, 특히 동화상을 코딩하는데 이용되는 참조 화상의 저장에 관한 것이다.

배경기술

많은 인터프레임 인코딩 시스템들은 참조 화상을 이용하고 있으며, 이러한 참조 화상의 이용은 인코딩된 비트 스트림의 길이를 줄이는데 도움이 된다. 이러한 유형은 인트라프레임 인코딩 방식을 이용하는 것보다 인코딩 효율성이 어느정도 낫다. 따라서, 많은 인코딩 표준들은 인트라프레임 인코딩 방식과 인터프레임 인코딩 방식을 통합하여 일련의 동영상으로부터 비트 스트림을 인코딩한다. 본 기술분야에서 알려져 있는 바와 같이, 화상 자체내의 요소들(인트라프레임)만을 이용하여 인코딩되는 "I" 화상, 화상 자체내의 요소들 및/또는 2개의 사전에 코딩된 참조 화상의 요소들(인터프레임)을 이용하여 인코딩되는 "B" 화상, 및 화상 자체내의 요소들 및/또는 1개의 사전에 코딩된 참조 화상의 요소들(인터프레임)을 이용하여 인코딩되는 "P" 화상 등의 상이한 유형의 참조 화상들이 인코딩 표준들에 이용되고 있다. "B" 화상과 "P" 화상 모두가 복수의 참조 화상을 이용할 수 있지만, "B" 화상을 블록당 많아야 2개의 모션-보상된 예측 신호가 화면간 예측(inter prediction)에 이용되게 하고, "P" 화상은 예측 블록당 하나의 예측자만이 이용되게 한다는 점에서 양자 간에 차이가 있다.

"B" 또는 "P" 화상들이 인코딩 및/또는 디코딩될 때, 이러한 화상들은 적절히 인코딩되거나 디코딩 동작 동안 적절히 구성될 수 있도록 다른 참조 프레임들을에 의존하게 된다. 인코딩/디코딩 시스템은 참조 화상이 저장될 수 있는 일정한 유형의 메모리 위치를 제공하여 다른 화상들이 이러한 참조 화상들을 고려하여 인코딩 또는 디코딩되게 한다. 확실히 어느 정도 후에는, 코딩될 더 이상의 화상들이 추후의 코딩 동작 동안 참조 화상을 이용하지 않을 것이기 때문에 참조 화상이 코딩 동작에 이용될 수 없다.

참조 화상들 모두를 저장 장치에 영구히 저장할 수도 있지만, 이러한 방안은 메모리 자원을 비효율적으로 이용하는 것이다. 따라서, 본 기술분야에서 알려져 있는 바와 같이, FIFO(First in First Out) 또는 LIFO(Last in First Out) 메모리 동작을 이용하는 등의 메모리 방안이 (불필요한 참조 화상들을 버림으로써) 참조 화상들을 필요한 공간을 줄이는 메모리 장치의 참조 화상 저장 동작에 이용될 수 있다. 하지만, 이러한 메모리 동작들은 인코딩 및/또는 디코딩되는 화상들이 시간과 뷰 상호관련성(temporal and view inter-relationship) 모두를 갖는 멀티뷰 코딩 시스템의 이용을 고려할 때 바람직하지 않은 결과를 발생시킬 수도 있다. 즉, 멀티뷰 코딩 시스템은 각각의 오브젝트/장면의 상이한 뷰를 각기 나타내는 복수 뷰의 동화상을 갖는 양상을 띤다. 참조 화상은 2개의 상이한 뷰와 연관된 화상들의 인코딩 또는 디코딩에 이용될 수도 있다.

예컨대, 도 1은 멀티뷰 비디오 코딩 시스템에 이용되는 참조 화상 구조의 일 실시예를 나타내고 있다. 구체적으로, 제시된 구조는 A. Vetro, Y. Su, H. Kimata, A. Smolic의 "Joint Multiview Video Model(JMVM) 1.0", JVT-T208.doc, Klagenfurt, Austria, July, 2006에서 제안된 멀티뷰 인코딩(MVC) 방안에 따라 시간(T0-T100) 동안 8개의 다른 뷰(S0-S7)를 이용하는 것에 관한 것이다. 이 멀티뷰 인코딩 표준은 AVC(Advanced Video Coding) 표준에서의 코딩에 기반한다(G. Sullivan, T. Wiegand, A. Luthra의 "Draft of Version 4 of H.264/AVC (ITU-T Recommendation H.264 및 ISO/IEC 14496-10 (MPEG-4 part 10) Advanced Video Coding)", Palma de Mallorca, ES 18-22, October 2004를 참조). 양 표준 간의 큰 차이는, AVC가 코딩 멀티뷰 화상들을 처리하지 못하지만 MVC는 처리할 수 있다는 점이다.

다시 도 1을 참조하면, 예를 들어 T1에서 뷰 S1과 연관된 화상을 코딩할 때, 코딩될 화상이 동일한 뷰(T0에서의 S1과 T2에서의 S1)로부터의 화상(참조 화상)과 관련되고, 코딩될 화상이 상이한 뷰(T1에서의 S0과 T1에서의 S2)로부터의 화상과 관련되는 것을 볼 수 있다. 따라서, S1, T1과 연관된 화상을 코딩할 때에는, 버퍼, 레지스터, RAM 등의 메모리 장치에 참조 화상(T0에서의 S1, T2에서의 S1, T1에서의 S0 및 T1에서의 S2)을 보유하는 것이 적절할 것이고, 이러한 디코딩된 화상은 DPB(decoded picture buffer)로 불리는 장치에 저장될 것이다.

[0009] 참조 화상을 DPB에서 관리하는 한가지 방식은 외부에서 생성되고 코더에 통신될 수 있는 신택스 요소(커맨드)를 이용해 DPB 일부를 비우는 것이다. AVC 상세에서는, 커맨드가 삽입되는 NAL(network abstract layer)을 이용하여 DPB에 저장된 참조 화상 모두가 "참조에 미사용(unused for reference)"이라는 것을 나타내는데 이용되는 IDR(instantaneous decoding refresh)을 표시할 수 있다. 이는, DPB에서의 참조 화상 모두가 IDR이 수신된 후에는 결국 제거되어야 한다는 것을 의미한다. IDR 화상은 이들이 (인터프레임 코딩이 아니라) 인트라프레임 코딩에 의존하는 "I" 또는 "SI" 화상(슬라이스)과 연관되기 때문에 이를 행할 수 있다. 그러므로, 코딩된 화상 시퀀스에서의 제1 화상은 일반적으로 IDR 화상이다.

[0010] 그러나, 복수의 뷰들이 코딩될 필요가 있을 수 있는 MVC 코딩 상황에서의 문제들을 해결하고자 할 때에는 현재의 IDR 구현이 비효율적이다. 예컨대, 뷰 S0이 AVC 호환 뷰라고 하자. AVC 호환 ID 화상이 시간 T16에서 뷰 S0에 존재한다면, 뷰 S0에서의 참조 화상만이 "참조에 미사용"으로 마킹되어야 하는지는 명확하지 않다. 즉, AVC 및 MVC에 있어 IDR 화상과 관련된 현 원리하에서는, DPB에서 임의 뷰의 모든 저장된 참조 화상이 "참조에 미사용"으로 마킹되고, DPB로부터 제거될 수 있는데, 이는 바람직한 결과가 아닐 수도 있다.

개요

[0012] 종래 기술의 이러한 문제점들은 비디오 인코딩을 위해 모션 추정 예측자로서 이용 가능한 모션 정보를 재이용하는 방법 및 장치에 관한 본 발명의 원리에 의해 해결된다.

[0013] 본 발명의 원리의 일 측면에 따르면, 디코딩된 화상 버퍼상에서 메모리 관리 동작을 수행하는 멀티뷰 비디오 코딩 환경에 이용되는 코더가 제공되며, 메모리 관리 동작은 제어 정보에 기반하여 특정한 뷰와 연관된 참조 화상을 제거할 것이다.

[0014] 첨부된 도면들과 함께 실시예들에 대한 다음의 상세한 설명을 통해 본 발명의 원리의 여러 측면, 특징 및 이점들을 알 수 있을 것이다.

발명의 상세한 설명

[0022] 본 발명의 원리는 인코딩 표준에 기반한 임의의 인트라프레임과 인터프레임에 적용될 수 있다. 본 명세서에서 이용되는 "화상(picture)"이란 용어는 "화상" 그 자체뿐만 아니라 여러 형태의 비디오 이미지 정보를 "프레임", "필드" 및 "슬라이스"로 기술하는 포괄적인 용어로서 이용되고 있다. 따라서, 화상이란 용어가 여러 요소의 비디오 정보를 나타내는데 이용되고 있지만, AVC는 슬라이스들의 이용을 지칭할 수도 있으며, 화상이 세부 분할되는 방법에 상관없이 참조 화상들이 "참조 화상"과 동일한 화상으로부터 슬라이스를 이용할 수 있다는 점에 유의하여야 한다.

[0023] 다음의 본 발명의 원리는 AVC에서 규정되는 바와 같이 NAL로 알려져 있는 요소들과 결합하여 일반적으로 설명되어 진다. 또한, 본 발명의 원리는 데이터 패킷 등의 데이터를 전송하는데 이용되고 데이터와 제어 패킷 모두에 삽입되는 헤더, 페이로드, 비트 스트림 등을 포함하는 복수의 포맷에도 적용될 수 있다는 점에 유의하여야 한다.

[0024] 본 명세서에서는, 화상을 코딩하는데 이용되는 코딩된 비디오 화상 정보로 참조 화상이 규정된다. 또한, 많은 비디오 코딩 시스템의 동작에서, 참조 화상이 DPB 등의 메모리에 저장된다. 어떤 참조 화상이 유지 혹은 삭제되어야 하는지를 관리하기 위해, DPB는 (일반적으로 코더에 의해) 저장된 참조 화상에 메모리 상태를 할애하는데 이용되고 메모리 관리 커맨드 동작(MMCO)으로 알려져 있는 커맨드를 이용한다. 예컨대, AVC/MVC 코더에 이용되는 메모리 상태는, 단기 참조 화상, 장기 참조 화상, 또는 참조 화상으로 이용되지 않은 것으로 마킹된 화상(이 경우, 메모리가 필요하면 DPB로부터 참조 화상이 버려질 것이다)을 포함한다. 저장된 참조 화상의 상태는 더 많은 화상이 코딩됨에 따라 변할 수 있는데, 예를 들어 하나의 화상이 코딩될 때 단기로 지정된 참조 화상이 제2 화상이 코딩될 때에는 장기 참조 화상으로 식별될 수 있다.

[0025] 또한, 본 명세서에서, C 언어 타입의 포맷팅을 이용하는 여러 커맨드(신택스 요소)는 이러한 커맨드에서 디스크립터로 다음의 목록을 사용하는 도면들에 상세히 기재되어 있다.

[0026] u(n):n 비트를 이용하는 미사인 정수. n이 신택스 테이블에서 "v"일 때, 비트수는 다른 신택스 요소들의 값에 따라 변한다. 이 디스크립터의 분석 프로세스는 msbf(msb first)를 갖는 미사인 정수의 이진 표현으로 해석되는 함수 read_bits(n)의 리턴 값에 의해 지정된다.

[0027] 또한, 다음과 같이 정의될 수 있다.

- [0028] ue(v):unsigned integer Exp-Golomb-coded syntax element with the left bit first;
- [0029] se(v):signed integer Exp-Golomb-coded syntax element with the left bit first.
- [0030] C:선택스 요소가 적용되는 카테고리를 나타낸다.
- [0031] 이제, 본 발명의 원리에 대해 설명한다. 당업자들이라면 본 명세서에서 명확히 설명되지 않을지라도 본 발명의 사상과 범주내에 포함되고 본 발명의 원리를 구현하는 여러 배치를 도출해 낼 수 있을 것이다.
- [0032] 본 명세서에서 인용된 모든 예들과 조건어들은 본 발명의 원리 및 개념들의 이해를 돋기 위한 것으로, 이러한 구체적인 예들과 조건들에 국한하려는 것이 아니다.
- [0033] 또한, 특정한 예들뿐만 아니라 본 발명의 원리, 측면 및 실시예들에 대한 언급은 본 발명의 구조 및 기능적인 균등물들을 포괄하고 있다. 더욱이, 이러한 균등물들은 현재 알려져 있는 균등물뿐만 아니라 장래에 개발될 균등물, 즉 구조에 상관없이 동일한 기능을 수행하도록 개발된 임의의 요소들을 포함한다.
- [0034] 도 2는 AVC에서 이용되는 바와 같이 NAL에 이용되는 선택스를 나타내고 있으며, AVC 호환 비트 스트림은 도시된 바와 같이 1 또는 5의 NAL 유닛 타입을 이용하는 코딩된 화상을 포함한다. MVC 코딩된 화상은 코딩된 화상에 대해 NAL 유닛 타입 20 및 21을 이용한다. NAL 유닛 타입 1과 20 모두는 각각의 비디오 코딩 표준에 대한 비 IDR 화상(슬라이스)을 나타내지만, NAL 유닛 타입 5 및 21은 IDR 화상을 나타낸다. 코더는, (예를 들어 비트 스트림에서) 5 또는 21 중 어느 하나의 NAL 유닛 타입을 수신할 때, DPB에 저장된 참조 화상의 상태가 "참조에 미사용"으로 변경되게 할 것이다.
- [0035] 본 발명의 일 실시예에서는, NAL 유닛을 NAL에 이용되는 서픽스 NAL 유닛으로 부른다. 서픽스 NAL 유닛은 디코딩 순서에서 다른 NAL 유닛에 후속하는 NAL 유닛으로 규정되고, 연관된 NAL 유닛으로 지칭되는 선행 NAL 유닛의 기술 정보를 포함한다. 바람직하게는, 서픽스 NAL의 후속이 연관된 NAL 유닛 바로 다음에 이어지는 것이다.
- [0036] 규정된 바와 같이, 서픽스 NAL 유닛은 20 또는 21의 nal_ref_idc를 가질 것이다. svc_mvc_flag가 0일 때, 서픽스 NAL 유닛은 0의 dependency_id와 quality_level을 가지고, 코딩된 슬라이스를 포함하지 않을 것이다. svc_mvc_flag가 1일 때, 서픽스 NAL은 0의 view_level을 가지고, 코딩된 화상 정보(슬라이스)를 포함하지 않을 것이지만, 제어 정보는 포함될 수도 있다. 서픽스 NAL 유닛은 연관된 NAL 유닛과 동일한 코딩된 화상에 속한다.
- [0037] slice_layer_in_svc_mvc_extension_rbsp() function의 구조를 규정하는 서픽스 NAL 유닛의 선택스가 도 3에 도시되어 있다. 이 서픽스 NAL 유닛은 연관된 NAL 유닛에 관한 정보를 얻어 적절한 동작을 취하기 위해 NAL 유닛에 존재하는 정보를 추출하는 MVC 호환 코더에 의해 이용될 수 있다.
- [0038] 따라서, 서픽스 NAL 유닛에서 어떤 뷰가 IDR 콜에 의해 영향받아야 하는지를 표시하는 정보가 제공되는 새로운 선택스가 제안된다. 즉, 새로운 선택스는 연관된 뷰에 대해서는 (DPB에서) 저장된 참조 화상이 "참조에 미사용"으로 마킹되게 하지만, 다른 뷰의 저장된 참조 화상은 그 메모리 상태를 유지하게 할 것이다.
- [0039] 본 발명의 일 실시예에서 IDR 화상을 DPB상에 갖는 동작을 지정하는 선택스 요소 mark_view_only가 도 4에 도시되어 있다. 서픽스 NAL 유닛에서 mark_view_only가 1일 때, 동일한 서픽스 NAL 유닛에 존재하는 view_id의 연관 뷰와 연관되는 DPB의 참조 화상 모두가 "참조에 미사용"으로 마킹된다. mark_view_only가 0일 때에는, DPB에 존재하는 참조 화상 모두가 "참조에 미사용"으로 마킹된다.
- [0040] 본 발명의 다른 실시예에서는, IDR 화상이 MVC NAL 유닛(타입 21)에 존재할 때, IDR 화상이 그 자신의 뷰에서의 화상만을 참조에 미사용으로 마킹하는 제약을 부과한다.
- [0041] 또 다른 실시예에서는, 연관된 NAL 유닛 이전에 전송되는 프리픽스 NAL 유닛이 전개될 수 있다. 또 다른 실시 예에서, IDR과 연관된 특정한 뷰를 선택하기 위한 커맨드 타입은 본 발명의 원리에 따라 커맨드를 부가하기 위해 사용자 데이터가 규정될 수 있는 NAL 유닛 어딘가에 캡슐화될 수 있다.
- [0042] 본 발명의 또 다른 실시예에서는, 어떤 참조 화상이 "참조에 미사용"으로 마킹되어야 하는지를 나타내는데 이용되는 제어 패킷이 비트 스트림내에 전개될 수도 있다. 구체적으로, 제어 패킷은 remove_reference_view(또는 이 커맨드와 유사한 커맨드)와 같은 선택스를 포함하고, 이 커맨드와 연관된 값은 어떤 저장된 참조 화상이 (연관된 뷰 또는 뷰들을 통해) DPB로부터 제거되어야 하는지를 나타낸다.
- [0043] 이러한 선택스는 어떤 뷰 또는 뷰들이 DPB로부터 동시에 제거되어야 하는지를 나타내는 제어 워드를 제공하기

위해 전개될 수 있다. 예컨대, 비디오 시퀀스가 그와 연관된 (뷰 0으로 시작하는) 8개의 뷰를 갖는다면, (뷰 0으로 시작하는) 뷰들 1, 4 및 5와 연관된 참조 화상을 제거하는데 이용되는 값은 (11001101)과 같은 8개의 비트 값에 따라 규정될 것이다. 유도된 이러한 값은 좌에서 우로 시작하고, 뷰 0에 값 "1"이 주어지면, 뷰 0과 연관된 참조 화상이 유지될 것이다. 우측으로 이동하여 뷰 1에 대해 값 "0"이 주어진다. 따라서, 본 발명의 일 실시예에서, DPB는 뷰 1과 연관된 참조 화상 모두를 제거할 것이다. 당업자들이라면 본 실시예의 원리에 따라 다른 커맨드 및 값들을 구현할 수 있음을 알 것이다.

[0044] 도 5는 본 발명의 원리에 따라 이용되는 코딩 시스템의 일 실시예를 나타내고 있다. 도 5의 블록도(500)에는, 코더(505), 코딩 버퍼(510), 디코딩된 화상 버퍼(515)와 데이터 포맷터(520) 간의 동작이 도시되어 있다. 코딩 동작(인코딩 혹은 디코딩) 동안에, 코더(505)에 의해 현재 코딩 중인 화상은 코딩 버퍼(510)에 존재하지만, 이전에 코딩된 참조 화상들은 디코딩된 화상 버퍼(515)에 저장된다. 전술한 바와 같이, AVC는 디코딩된 화상 버퍼(515)에서의 참조 화상이 어떻게 유지되어야 하는지를 코더(505)가 지정하게 하는 메모리 관리 제어 동작(MMC 0)과 같은 커맨드의 이용을 개시하고 있다. 즉, 화상이 인코딩 중일 때, MMC0는 현재 인코딩 중인 화상의 헤더에 입력되어 이 화상 전에 왔던 참조 화상에 대해 무엇을 해야하는지를 지정한다. 이 동작이 "마킹"으로 알려져 있다. 코더(505)는 이후 이러한 커맨드를 이용하여 디코딩된 화상 버퍼(515)에 존재하는 참조 화상들에 대해 무엇을 해야하는지를 결정한다. 여러 요소의 비디오 정보를 나타내는데 화상이란 용어를 사용하였지만, AVC는 슬라이스들의 이용을 지칭할 수도 있으며, 화상이 세부 분할되는 방법에 상관없이 참조 화상들이 "참조 화상"과 동일한 화상으로부터 슬라이스를 이용할 수 있다는 점에 유의하여야 한다.

[0045] 화상들이 인코딩되면, 이들은 비트 스트림의 일부로 전송될 수 있으며, 이러한 데이터는 데이터 네트워크상의 전송을 위해 데이터 포맷터(520)를 이용하여 비트 스트림에 포맷팅된다. 전송 스트림(IP 패킷 또는 MPEG-2 전송 스트림 등)으로 또한 전송되는 NAL 유닛 품으로 데이터가 전송되는 것이 바람직하며, 데이터 포맷터(520)는 NAL 유닛을 전송 패킷으로 전송한다. 따라서, 데이터 포맷터(520)는 NAL 유닛으로 어드레싱된 커맨드와 코딩된 화상 정보 모두를 전송할 수 있고, 이러한 NAL 유닛은 프리픽스 및/또는 서픽스 NAL 유닛일 수 있다. 또한, 데이터 포맷터(520)는 IDR 정보 커맨드를 NAL 유닛의 임의의 사용자 규정부내에 추가할 수도 있다. 데이터 포맷터(520)는 또한 어드레싱된 데이터 커맨드를 데이터 패킷의 헤더, 데이터 패킷의 페이로드 또는 전송 패킷의 이들 조합에 입력할 수도 있다는 점을 알아야 한다.

[0046] 본 발명의 일 실시예에서, 데이터 포맷터(520)는 전송 패킷의 코딩된 비트 스트림을 수신하고, (동화상 시퀀스를 구성하기 위해) 코더(505)에 의해 디코딩된 비디오 화상 데이터의 품으로 디코딩될 수 있는 NAL 유닛으로 이러한 수신된 데이터를 포맷팅할 수 있다. 즉, 데이터 포맷터(520)는 NAL 유닛을 판독하여 어떤 화상이 IDR 화상을 나타내는지, 코더(505)가 특정한 뷰와 연관된 참조 화상을 "참조에 미사용"으로 마킹하기 위해 NAL 데이터를 판독하는데 이용되는 유닛인지를 결정할 수 있다. 코더(505)는 그후 본 실시예에서와 같이 동작하여 수신된 비트 스트림을 디코딩하는데 이용되고, 코딩 버퍼(510)와 디코딩된 화상 버퍼(515)는 AVC 및 MVC 비디오 코딩 표준에 규정된 방식에 따라 이용될 것이다.

[0047] 도 6은 본 발명의 일 실시예에서 IDR 화상을 이용하는 방법의 순서도(600)를 나타내고 있다. 단계 605에서, 인코딩될 화상의 화상 데이터가 코더(505)에 의해 프로세싱된다. 화상 데이터가 인코딩될 때, 코더(505)는 단계 610에서 코딩 중인 화상이 동시 디코딩 리프레시 화상을 나타낼 것인지를 지정하는 커맨드를 추가한다. 이 커맨드의 일부는 (IDR이면) 화상이 저장된(또는 DPB(515)에 저장될) 참조 화상 모두에 영향을 주거나, 또는 특정한 뷰와 연관된 저장 참조 화상이 "참조에 미사용"으로 마킹될지 여부를 나타낼 것이다.

[0048] 단계 615에서, 데이터 포맷터(520)는 단계 610에서 코더에 의해 전개된 커맨드를 이용하고, 이러한 IDR 커맨드를 NAL(바람직하게는 서픽스 NAL, 전술한 바와 같이 본 발명의 원리에 따라 다른 전송 포맷도 이용될 수 있음)로 전송한다.

[0049] 단계 620에서, 데이터 포맷터(520)는 코딩된 데이터 스트림을 수신하고, NAL을 판독하여 수신된 NAL이 IDR을 나타내는지, 저장된 참조 화상(뷰에 의해 식별됨)이 IDR 동작에 의해 영향을 받을 것인지를 결정한다. 단계 625에서, 코더(505)는, (바람직한 실시예에서) 수신된 연관 NAL로부터 코딩된 화상 정보를 디코딩하는 것과 같이, IDR 커맨드가 뷰에 의해 서픽스 NAL에서 식별되는 저장된 참조 화상을 "참조에 미사용"으로 마킹하게 한다. 단계 630에서, DPB(515)는 이러한 커맨드를 실행하고, IDR 커맨드에서 선택되는 저장된 참조 화상을 "참조에 미사용"으로 마킹하며, DPB(515)는 이러한 참조 화상을 결국 제거할 것이다.

[0050] 따라서, 당업자들이라면 본 명세서에 제시된 블록도들이 본 발명의 원리를 구현하는 예시적인 회로의 개념도를 나타내고 있음을 알 것이다. 마찬가지로, 임의의 순서도, 상태 천이도, 의사코드 등은, 컴퓨터 또는 프로세서

등이 명확히 도시되어 있는지 여부에 상관없이 컴퓨터 또는 프로세서 등에 의해 실행되고, 컴퓨터 관독가능 매체에 실질적으로 표시할 수 있는 여러 프로세스를 나타내고 있음을 알 것이다.

[0051] 도면에 도시되어 있는 여러 구성요소들의 기능은 전용 하드웨어뿐만 아니라 적합한 소프트웨어와 연관되어 소프트웨어를 실행시킬 수 있는 하드웨어의 이용을 통해 제공될 수 있다. 이러한 기능들은, 프로세서에 의해 제공될 때, 단일의 전용 프로세서, 단일의 공유 프로세서 혹은 일부가 공유될 수 있는 복수의 개별 프로세서에 의해 제공될 수 있다. 또한, "프로세서" 혹은 "컨트롤러"와 같은 용어들은 소프트웨어를 실행시킬 수 있는 하드웨어만을 지칭하는 것은 아니며, DSP 하드웨어, 소프트웨어 저장용 ROM, RAM 및 비휘발성 저장매체(이러한 예에 국한되지는 않음)를 또한 포함할 수 있다.

[0052] 또한, 종래의 다른 하드웨어들도 포함될 수 있다. 마찬가지로, 도면들에 도시되어 있는 임의의 스위치들은 단지 개념적인 것으로, 이들의 기능들은 프로그램 로직, 전용 로직, 프로그램 제어와 전용 로직 간의 상호작용을 통하여거나 수동으로 수행될 수 있으며, 구현자는 환경에 따라 특정한 방안을 구체적으로 선택할 수 있다.

[0053] 첨부된 청구범위에서, 지정된 기능을 수행하는 수단으로 표현된 임의의 구성요소들은 예를 들어 a)그 기능을 수행하는 회로 요소들의 조합 혹은 b)적합한 회로와 결합되어 펌웨어, 마이크로코드 등의 기능을 실행시키는 임의 형태의 소프트웨어를 포함하는 기능들을 어떤 식으로든 수행하는 것을 포괄한다. 이러한 청구범위에 의해 규정되는 본 발명의 원리는 여러 인용된 수단들에 의해 제공되는 기능들이 청구범위에서 요구되는 방식과 결합되고 통합되어 진다. 따라서, 이러한 기능들을 제공할 수 있는 임의의 수단들은 본 명세서에서 설명되어진 수단들과 균등한 것으로 간주되어야 한다.

[0054] 또한, 본 명세서에서 본 발명의 원리의 "일 실시예"에 대한 참조는, 이러한 실시예와 관련하여 설명되어지는 특징한 특성, 구조, 특징 등이 본 발명의 원리의 적어도 하나의 실시예에 포함되고 있다는 것을 의미하는 것으로 이해하여야 한다. 따라서, 본 명세서에서 "일 실시예" 혹은 "실시예"와 같은 용어가 반드시 동일한 실시예를 지칭하는 것으로 이해하여서는 안된다.

[0055] 당업자들이라면 본 명세서의 개시 내용을 통해 본 발명의 원리의 여러 특징 및 이점들을 쉽게 알 수 있을 것이다. 또한, 본 발명의 원리들의 개시 내용은 하드웨어, 소프트웨어, 펌웨어, 특수 프로세서 혹은 이들의 조합 등 여러 형태로 구현될 수 있다는 것을 알 것이다.

[0056] 바람직하게는, 본 발명의 원리들의 개시 내용이 하드웨어와 소프트웨어의 조합으로 구현되는 것이다. 또한, 소프트웨어는 프로그램 저장 매체상에 실재 구현된 애플리케이션 프로그램으로 실행될 수도 있다. 애플리케이션 프로그램은 임의의 적합한 아키텍처를 포함하는 기계에 업로드되거나 기계에 의해 실행될 수 있다. 이러한 기계는 하나 이상의 CPU, RAM, I/O 인터페이스와 같은 하드웨어를 갖는 컴퓨터 플랫폼에서 구현되는 것이 바람직하다. 또한, 컴퓨터 플랫폼은 운영 시스템과 마이크로 명령 코드를 포함할 수 있다. 본 명세서에서 설명되는 여러 프로세스와 기능들은 CPU에 의해 실행될 수 있는 마이크로 명령 코드 혹은 애플리케이션 프로그램의 일부이거나 이들의 조합일 수도 있다. 더욱이, 여러 다른 주변 유닛들은 추가의 데이터 저장 유닛 및 인쇄 유닛 등의 컴퓨터 플랫폼에 접속될 수 있다.

[0057] 또한, 첨부된 도면에 도시되어 있는 구성 시스템 컴포넌트와 방법들 중 일부가 소프트웨어로 구현되는 것이 바람직하므로, 시스템 컴포넌트들 간의 실제 접속 혹은 프로세스 기능 블록들은 본 발명의 원리가 프로그램되는 방식에 따라 상이할 수도 있다. 당업자들이라면 본 명세서의 개시 내용을 통해 본 발명의 원리의 여러 유사한 구현 혹은 구성을 도출해낼 수 있을 것이다.

[0058] 본 명세서에서는 첨부된 도면을 참조하여 본 발명의 실시예들에 대해 설명하였지만, 본 발명의 원리는 이러한 실시예들에 국한되지 않으며, 당업자들이라면 본 발명의 원리의 사상과 범주를 벗어나지 않고서 여러 변경 및 수정을 행할 수 있을 것이다. 따라서, 이러한 변경 및 수정들은 다음의 청구범위에 개시되어 있는 본 발명의 원리의 범주내에 포함되는 것으로 의도되어 진다.

도면의 간단한 설명

[0015] 다음의 도면을 참조하면 본 발명의 원리를 보다 잘 이해할 수 있을 것이다.

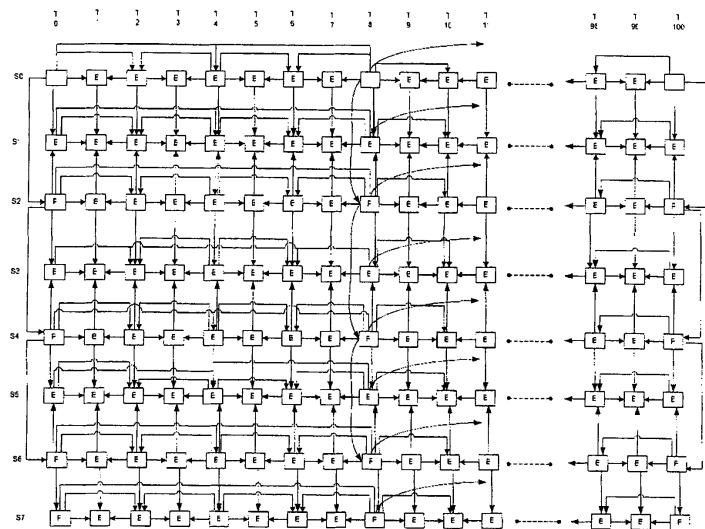
[0016] 도 1은 상이한 시간에서 비디오 화상 뷰의 멀티뷰 코딩의 일 실시예를 나타내는 도면으로, 비디오 화상들이 참조 화상들을 이용해 코딩되는 것을 나타내는 도면.

[0017] 도 2는 본 발명의 원리에 따라 NAL 유닛을 지정하는데 이용되는 코드들의 일 실시예를 나타내는 도면.

- [0018] 도 3은 본 발명의 원리에 따라 이용되는 신택스 요소 ref_pic_list_reordering()에 대한 의사 코드의 일 실시예를 나타내는 도면.
- [0019] 도 4는 본 발명의 원리에 따라 이용되는 신택스 요소 mark_view_only()에 대한 의사 코드의 일 실시예를 나타내는 도면.
- [0020] 도 5는 본 발명의 원리에 따라 이용될 코딩 시스템의 일 실시예를 나타내는 도면.
- [0021] 도 6은 본 발명의 원리에 따라 IDR 화상을 이용하는 코딩의 일 실시예를 나타내는 도면.

도면

도면1



도면2

nal_unit_type	NAL 유닛 및 RBSP 신박스 구조의 콘텐트	C
0	미지정	
1	비IDR 화상의 코딩된 슬라이스 slice_layer_without_partitioning_rbsp()	2, 3, 4
2	코딩된 슬라이스 테이터 파티션 A slice_data_partition_a_layer_rbsp()	2
3	코딩된 슬라이스 테이터 파티션 B slice_data_partition_b_layer_rbsp()	3
4	코딩된 슬라이스 테이터 파티션 C slice_data_partition_c_layer_rbsp()	4
5	IDR 화상의 코딩된 슬라이스 slice_layer_without_partitioning_rbsp()	2, 3
6	추가 개선 정보(SEI) sei_rbsp()	5
7	시퀀스 파라미터 세트 seq_parameter_set_rbsp()	0
8	화상 파라미터 세트 pic_parameter_set_rbsp()	1
9	액세스 유닛 구획 문자 access_unit_delimiter_rbsp()	6
10	시퀀스 끝 end_of_seq_rbsp()	7
11	스트림 끝 end_of_stream_rbsp()	8
12	필러 테이터 filler_data_rbsp()	9
13	시퀀스 파라미터 세트 확장 seq_parameter_set_extension_rbsp()	10
14..18	유보	
19	파티션 없는 보조 코딩 화상의 코딩된 슬라이스 slice_layer_without_partitioning_rbsp()	2, 3, 4
20	스케일 가능한 확장에서 비IDR 화상의 코딩된 슬라이스 slice_layer_in_svc_mvc_extension_rbsp()	2, 3, 4
21	스케일 가능한 확장에서 IDR 화상의 코딩된 슬라이스 slice_layer_in_svc_mvc_extension_rbsp()	2, 3
22..23	유보	
24..31	미지정	

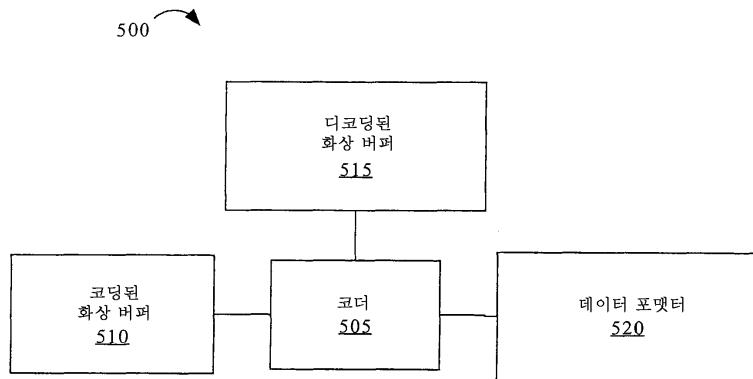
도면3

slice_layer_in_svc_mvc_extension_rbsp() {	C	디스크립터
if(svc_mvc_flag == 1) {		
if(view_level == 0) {		
view_id	2	u(10)
anchor_pic_flag	2	u(1)
} else {		
slice_header()		
}		
} else {		
.....		
}		
}		

도면4

	C	디스크립터
slice_layer_in_svc_mvc_extension_rbsp() {		
if(svc_mvc_flag == 1) {		
if(view_level == 0) {		
view_id	2	u(10)
anchor_pic_flag	2	u(1)
if(nal_unit_type == 21)		
mark_view_only	2	u(1)
} else {		
slice_header()		
}		
} else {		
.....		
}		
}		

도면5



도면6

