(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0148967
(43) 공개일자 2024년10월11일

(51) 국제특허분류(Int. Cl.)
H04N 19/597 (2014.01) *H04N 19/174* (2014.01)
H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/597 (2015.01)
H04N 19/174 (2015.01)
(21) 출원번호 10-2024-7033120(분할)
(22) 출원일자(국제) 2008년04월11일
심사청구일자 2024년10월04일
(62) 원출원 특허 10-2023-7008405
원출원일자(국제) 2008년04월11일
심사청구일자 2023년03월09일
(85) 번역문제출일자 2024년10월04일
(86) 국제출원번호 PCT/US2008/004747
(87) 국제공개번호 WO 2008/127676
국제공개일자 2008년10월23일
(30) 우선권주장
60/923,014 2007년04월12일 미국(US)
60/925,400 2007년04월20일 미국(US)

(71) 출원인
돌비 인터네셔널 에이비
아일랜드 디02 브이케이60 더블린 그랜드 커널 독 랜즈 블록 씨 서 존 로저슨스 키 77
(72) 발명자
팬디트, 푸어빈 비브하스
미국 뉴저지 08823, 프랭클린 파크, 피어트리 레인 23
인, 펭
미국 뉴저지 08550, 웨스트 윈저, 워위크 드라이브 65
티안, 동
미국 뉴저지 08550, 웨스트 윈저, 헤더 드라이브 20214
(74) 대리인
양영준, 백만기

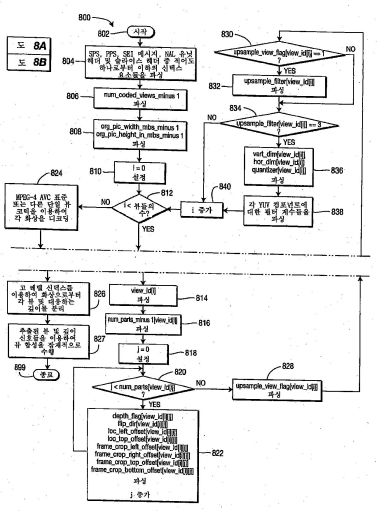
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 **비디오 인코딩 및 디코딩의 타일링**

(57) 요 약

예를 들어, 비디오 인코딩 및 디코딩의 뷰 타일링에 관한 구현들이 제공된다. 특정 방법은 단일 화상으로 조합된 다중 화상들을 포함하는 비디오 화상을 액세스하고(826), 액세스된 비디오 화상의 다중 화상들이 어떻게 조합되었는지를 나타내는 정보를 액세스하고(806,808,822), 다중 화상들 중 적어도 하나의 디코딩된 표현을 제공하도록 상기 비디오 화상을 디코딩하고(824,826), 액세스된 정보 및 디코딩된 비디오 화상을 출력으로서 제공하는(824,826) 것을 포함한다. 일부 다른 구현들은 단일 비디오 화상에 포함된 다중 화상들이 단일 비디오 화상으로 어떻게 조합되었는지를 나타내는 정보를 포멧팅 또는 처리하고, 조합된 다중 화상들의 인코딩된 표현을 포멧팅 또는 처리한다.

대 표 도



(52) CPC특허분류
H04N 19/70 (2015.01)

명 세 서

청구범위

청구항 1

컴퓨터 프로그램을 갖는 컴퓨터 판독가능 저장 매체로서, 상기 컴퓨터 프로그램은 명령어들을 갖고, 상기 명령어들은 컴퓨팅 디바이스에 의해 실행 될 때, 상기 컴퓨팅 디바이스로 하여금:

멀티-뷰 비디오의 비디오 화상들을 수신하고;

두 개의 비디오 화상들을 단일 비디오 화상으로 조합하고 - 제1 비디오 화상은 상기 멀티-뷰 비디오의 제1 뷰이고, 제2 비디오 화상은 상기 멀티-뷰 비디오의 제2 뷰임 -;

상기 두 개의 비디오 화상들이 어떻게 조합되었는지를 나타내는 시그널링 섹션을 생성하고 - 상기 시그널링 섹션 내의 신택스 요소들은 상기 두 개의 비디오 화상들 중 적어도 하나가 수평 방향 또는 수직 방향으로 개별적으로 플립핑(flipped)된 것을 나타내고, 상기 신택스 요소들은 슬라이스 헤더, 시퀀스 파라미터 세트(sequence parameter set), 화상 파라미터 세트(picture parameter set), 네트워크 추상 계층(network abstraction layer) 유닛 헤더, 또는 보조 향상 정보(supplemental enhancement information) 메시지 중 적어도 하나에 포함되고, 상기 제1 뷰의 상기 제1 비디오 화상은 플립핑되지 않고 상기 제2 뷰의 상기 제2 비디오 화상은 플립핑되고, 상기 제1 뷰의 상기 제1 비디오 화상 및 상기 제2 뷰의 상기 제2 비디오 화상은 서로 나란히 배열되거나 위 아래로 배열됨 -;

상기 조합된 두 개의 비디오 화상들의 인코딩된 표현을 제공하기 위해 상기 단일 비디오 화상을 인코딩하고;

출력으로서, 상기 시그널링 섹션 및 상기 인코딩된 표현을 조합하도록 야기하는, 컴퓨터 판독가능 저장 매체.

발명의 설명

기 술 분 야

[0001] 관련 출원의 상호참조

[0002] 본 출원은 (1) 미국 임시 출원 번호 제60/923,014호(2007년 4월 12일 출원) 및 명칭이 "Multiview information"(대리인 문서번호 PU070078) 및 (2) 미국 임시 출원 번호 제60/925,400호(2007년 4월 20일 출원) 및 명칭이"View Tiling in MVC Coding"(대리인 문서번호 PU070103)의 이득을 청구한다. 이 두 출원들 각각은 그 전체가 본원에 참조문헌으로 포함된다.

[0003] 본 원리들은 일반적으로 비디오 인코딩 및/또는 디코딩에 관한 것이다.

배 경 기 술

[0004] 비디오 디스플레이 제조자들은 단일 프레임상에 상이한 뷰들(views)을 배열 또는 타일링하는데 프레임워크를 이용할 수 있다. 그러면, 이 뷰들은 그들 각자의 위치들로부터 추출되어 랜더링될 수 있다.

발명의 내용

해결하려는 과제

[0005] 비디오 인코딩 및 디코딩의 타일링에 관한 기술을 제공한다.

과제의 해결 수단

[0006] 일반적인 측면에 따라, 단일 화상으로 조합된 다중 화상들을 포함하는 비디오 화상이 액세스된다. 액세스된 비디오 화상의 다중 화상들이 어떻게 조합되었는지를 나타내는 정보가 액세스된다. 비디오 화상은 조합된 다중 화상들의 디코딩된 표현을 제공하도록 디코딩된다. 액세스된 정보 및 디코딩된 비디오 화상은 출력으로서 제공된다.

[0007] 다른 일반적인 측면에 따라, 비디오 화상에 포함된 다중 화상들이 단일 화상으로 어떻게 조합되었는지는 나타내는 정보가 생성된다. 비디오 화상은 조합된 다중 화상들의 인코딩된 표현을 제공하도록 인코딩된다. 생성된 정보 및 인코딩된 비디오 화상은 출력으로서 제공된다.

[0008] 다른 일반적인 측면에 따라, 신호 또는 신호 구조는 단일의 비디오 화상에 포함된 다중 화상들이 단일 비디오 화상으로 어떻게 조합되었는지를 나타내는 정보를 포함한다. 또한 신호 또는 신호 구조는 조합된 다중 화상들의 인코딩된 표현을 포함한다.

[0009] 다른 일반적인 측면에 따라, 단일 화상으로 조합된 다중 화상들을 포함하는 비디오 화상이 액세스된다. 액세스된 비디오 화상의 다중 화상들이 어떻게 조합되었는지를 나타내는 정보가 액세스된다. 비디오 화상은 다중 화상들의 적어도 하나의 디코딩된 표현을 제공하도록 디코딩된다. 액세스된 정보 및 디코딩된 표현은 출력으로서 제공된다.

[0010] 다른 일반적인 측면에 따라, 단일 화상으로 조합된 다중 화상들을 포함하는 비디오 화상이 액세스된다. 액세스된 비디오 화상의 다중 화상들이 어떻게 조합되었는지를 나타내는 정보가 액세스된다. 비디오 화상은 조합된 다중 화상들의 디코딩된 표현을 제공하도록 디코딩된다. 디스플레이를 위해 다중 화상들의 적어도 하나를 선택하는 사용자 입력이 수신된다. 적어도 하나의 선택된 화상의 디코딩된 출력, 액세스된 정보에 기초하여 제공되는 디코딩된 출력, 디코딩된 표현 및 사용자 입력이 제공된다.

[0011] 하나 이상의 구현들의 상세들이 첨부 도면들 및 이하의 상세한 설명에서 설명된다. 하나의 특정 방식으로 기술된 경우라도 구현들은 다양한 방식들로 구성 또는 실시될 수 있다는 것은 알아야 한다. 예를 들어, 구현은 방법으로서 수행되거나 동작들의 세트를 수행하도록 구성된 장치로서 실시되거나 신호로서 실시될 수 있다. 다른 측면들 및 특징들은 첨부 도면들 및 청구범위와 함께 고려되는 이하의 상세한 설명으로부터 명백해질 것이다.

발명의 효과

[0012] 비디오 인코딩 및 디코딩의 타일링에 관한 기술을 제공한다.

도면의 간단한 설명

[0013] 도 1은 단일 프레임상에서 타일링된 4개의 뷰들의 예를 도시하는 도면.

도 2는 단일 프레임상에서 플립핑 및 타일링된 4개의 뷰들의 예를 도시하는 도면.

도 3은 본 원리들의 실시예에 따라, 본 원리들이 적용될 수 있는 비디오 인코더에 대한 블록도.

도 4는 본 원리들의 실시예에 따라, 본 원리들이 적용될 수 있는 비디오 디코더에 대한 블록도.

도 5a 및 도 5b는 본 원리들의 실시예에 따라, MPEG-4 AVC 표준을 사용하여 복수의 뷰들에 대한 화상들을 인코딩하는 방법의 흐름도.

도 6a 및 도 6b는 본 원리들의 실시예에 따라, MPEG-4 AVC 표준을 사용하여 복수의 뷰들에 대한 화상들을 디코딩하는 방법의 흐름도.

도 7a 및 도 7b는 본 원리들의 실시예에 따라, MPEC-4 AVC 표준을 사용하여 복수의 뷰들 및 깊이들에 대한 화상들을 인코딩하는 방법의 흐름도.

도 8a 및 도 8b는 본 원리들의 실시예에 따라, MPEC-4 AVC 표준을 사용하여 복수의 뷰들 및 깊이들에 대한 화상들을 디코딩하는 방법의 흐름도.

도 9는 본 원리들의 실시예에 따라, 깊이 신호의 예를 도시하는 도면.

도 10은 본 원리들의 실시예에 따라, 타일로서 부가된 깊이 신호의 예를 도시하는 도면.

도 11은 본 원리들의 실시예에 따라, 단일 프레임상에 타일링된 5 뷰들의 옐르 도시하는 도면.

도 12는 본 원리들의 실시예에 따라, 본 원리들이 적용될 수 있는 예시적인 다중-뷰 비디오 코딩(MVC) 인코더에 대한 블록도.

도 13은 본 원리들의 실시예에 따라, 본 원리들이 적용될 수 있는 예시적인 다중-뷰 비디오 코딩(MVC) 디코더에 대한 블록도.

도 14는 본 원리들의 실시예에 따라, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용한 화상들의 인코딩에 대비하여 복수의 뷰들에 대한 화상들을 처리하는 방법의 흐름도.

도 15a 및 도 15b는 본 원리들의 실시예에 따라, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용하여 복수의 뷰들에 대한 화상들을 인코딩하는 방법의 흐름도.

도 16은 본 원리들의 실시예에 따라, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용한 화상들의 디코딩에 대비하여 복수의 뷰들에 대한 화상들을 처리하는 방법의 흐름도.

도 17a 및 도 17b는 본 원리들의 실시예에 따라, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용하여 복수의 뷰들에 대한 화상들을 디코딩하는 방법의 흐름도.

도 18은 본 원리들의 실시예에 따라, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용한 화상들의 인코딩에 대비하여 복수의 뷰들 및 깊이들에 대한 화상들을 처리하는 방법의 흐름도.

도 19a 및 도 19b는 본 원리들의 실시예에 따라, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용하여 복수의 뷰들 및 깊이들에 대한 화상들을 인코딩하는 방법의 흐름도.

도 20은 본 원리들의 실시예에 따라, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용한 화상들의 디코딩에 대비하여 복수의 뷰들 및 깊이들에 대한 화상들을 처리하는 방법의 흐름도.

도 21a 및 도 21b는 본 원리들의 실시예에 따라, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용하여 복수의 뷰들 및 깊이들에 대한 화상들을 디코딩하는 방법의 흐름도.

도 22는 본 원리들의 실시예에 따라, 픽셀 레벨에서의 타일링 예들을 도시하는 도면.

도 23은 본 원리들의 실시예에 따라, 본 원리들이 적용될 수 있는 비디오 처리 디바이스에 대한 블록도.

발명을 실시하기 위한 구체적인 내용

[0014] 다양한 구현들은 비디오 인코딩 및 디코딩에서 뷰 타일링을 위한 방법 및 장치에 관한 것이다. 따라서, 당업자는, 여기에 명확히 기술되거나 도시되어 있지 않을지라도, 본 원리들을 실시하며 그 사상 및 범위내에 포함되는 다양한 장치들을 고안할 수 있을 것이라는 점을 이해할 수 있을 것이다.

[0015] 모든 예들 및 여기에 인용된 조건적인 언어는 독자가 발명자(들)에 의해 제공된 본 발명의 원리들 및 개념들을 당 기술 이상으로 이해하는 것을 돕기 위한 교육적인 목적을 위해 의도되었고, 이러한 구체적으로 인용된 예들 및 조건들에 대해 제한이 없는 것으로 해석된다.

[0016] 또한, 여기에서 원리들, 특징들, 본 원리들의 실시예들 및 그 특정 실시예들을 인용하는 모든 서술들은 구조적 및 기능적 등가물들을 포함하는 것으로 의도된다. 부가적으로, 이러한 등가물들은 앞으로 개발되는 등가물들, 즉 구조에 관계없이 동일한 기능을 수행하는 개발된 임의의 요소들 뿐만 아니나 현재 알려진 등가물들을 포함하는 것으로 의도된다.

[0017] 따라서, 예를 들어, 여기에 제시된 블록도들은 본 원리들을 실시하는 예시적인 회로의 개념적인 뷰들을 표현한다는 것을 당업자라면 이해할 것이다. 유사하게, 임의의 순서도들, 흐름도들, 상태 천이 다이어그램들, 의사 코드 등은 컴퓨터 판독가능 매체들에서 실제로 표현될 수 있고 이에 따라 이러한 컴퓨터가 명시적으로 도시되어 있는지 여부에 상관없이 컴퓨터 또는 프로세서에 의해 실행될 수 있는 다양한 처리들을 표현한다는 것을 이해할 것이다.

[0018] 도면들에서 도시된 다양한 요소들의 기능들은 적절한 소프트웨어와 연관된 소프트웨어 실행 가능한 하드웨어 및 전용 하드웨어의 사용을 통해 제공될 수 있다. 기능들은, 프로세서에 의해 제공될 때 단일 전용 프로세서, 단일 공유 프로세서 또는 일부가 공유될 수 있는 복수의 개별 프로세서들에 의해 제공될 수 있다. 또한, 용어 "프로세서" 또는 "제어기"의 명시적 사용은 소프트웨어 실행 가능한 하드웨어를 배타적으로 칭하는 것으로 해석되면 안 되고, 디지털 신호 프로세서("DSP") 하드웨어, 소프트웨어를 저장하기 위한 판독 전용 메모리("ROM"), 랜덤 액세스 메모리("RAM") 및 비-휘발성 저장매체를 제한 없이 암묵적으로 포함할 수 있다.

[0019] 종래의 및/또는 관습적인 다른 하드웨어가 또한 포함될 수도 있다. 유사하게, 도면들에서 도시된 임의의 스위치들은 단순히 개념적이다. 그들의 기능은 프로그램 로직의 동작을 통해, 전용 로직을 통해, 프로그램 제어와 전용 로직의 상호작용을 통해 또는 심지어 수동으로 수행될 수 있고, 특정 기술은 문맥으로부터 보다 구체적으

로 이해되는 바와 같이 구현자에 의해 선택 가능하다.

[0020] 청구항들에서, 특정 기능을 수행하는 수단으로서 표현된 임의의 요소는 예를 들어, 가) 상기 기능을 수행하는 회로 구성요소들의 조합 또는 나) 펌웨어, 마이크로코드 등을 포함하며, 기능을 수행하기 위해 해당 소프트웨어를 실행하는 적절한 회로와 조합된 임의의 유형의 소프트웨어를 포함하는 상기 기능을 수행하는 임의의 방식을 포함하도록 의도된다. 이러한 청구항들에 의해 규정된 본 원리들은, 다양한 인용된 수단에 의해 제공되는 기능들이 청구범위가 요구하는 방식으로 조합 및 접목된다는 사실이 존재한다. 따라서, 이러한 기능들을 제공할 수 있는 임의의 수단은 여기서 도시된 것과 동등하다고 간주된다.

[0021] 명세서에서 본 원리들의 "일 실시예"(또는 "일 구현") 또는 "실시예"(또는 "구현")에 대한 참조는 실시예와 관련하여 기술된 특별한 특징, 구조, 특성 등이 본 원리들의 적어도 하나의 실시예에 포함된다는 것을 의미한다. 따라서, 명세서 전체에 걸쳐서 다양한 위치들에 나오는 구(phrase) "일 실시예에서" 또는 "실시예에서"의 출현은 반드시 동일한 실시예 모두를 지칭하는 것은 아니다.

[0022] 예를 들어 "A 및/또는 B" 및 "A 및 B 중 적어도 하나"의 경우에서, 용어 "및/또는" 및 "중 적어도 하나"의 사용은 첫 번째 나열된 옵션(A)만 선택, 또는 두 번째 나열된 옵션(B)만 선택, 또는 옵션들 둘 다(A 및 B)의 선택을 포함하도록 의도된다. 추가적인 예로서, "A, B 및/또는 C" 및 "A, B 및 C 중 적어도 하나"의 경우에서, 이러한 구는 첫 번째 나열된 옵션(A)만 선택, 또는 두 번째 나열된 옵션(B)만 선택, 세 번째 나열된 옵션(C)만 선택 또는 첫 번째 및 두 번째 나열된 옵션들(A 및 B)만 선택, 첫 번째 및 세 번째 나열된 옵션들(A 및 C)만 선택, 두 번째 및 세 번째 나열된 옵션들(B 및 C)만 선택, 3개의 모든 옵션들(A 및 B 및 C)의 선택을 포함하도록 의도된다. 이는 다수의 나열된 항목들에 대해서도 당업자에 의해 쉽게 이해될 수 있기 때문에 확장될 수 있다.

[0023] 또한, 본 원리들의 하나 이상의 실시예들이 여기에서 MPEG-4 AVC 표준에 대해 기술되지만, 본 원리들은 이 표준에만 한정되는 것은 아니고, 따라서 본 원리들의 사상을 유지하면서, 다른 표준들, 권고안들 및 MPEG-4 AVC 표준의 확장들을 포함하는 확장들, 구체적으로는 비디오 코딩 표준들, 권고안들 및 그 확장들에 대해서 활용될 수 있다.

[0024] 또한, 본 원리들의 하나 이상의 실시예들은 여기에서 MPEG-4 AVC 표준의 멀티 뷰 비디오 코딩 확장에 대해 기술되지만, 본 원리들이 이 표준에만 한정되는 것은 아니고, 따라서 본 원리들의 사상을 유지하면서, 다른 비디오 코딩 표준들, 권고안들 및 멀티 뷰 코딩에 관련된 확장들에 대해서 활용될 수 있다. 멀티 뷰 비디오 코딩(MVC)은 멀티-뷰 시퀀스의 인코딩에 대한 압축 프레임워크(framework)이다. 멀티 뷰 비디오 코딩(MVC) 시퀀스는 서로 다른 뷰 포인트로부터 동일한 장면을 포착하는 2 이상의 비디오 시퀀스들의 세트이다.

[0025] 또한, 비디오 콘텐트에 대해 깊이 정보(depth information)를 이용하는 본 원리들의 하나 이상의 실시예들이 여기에서 기술되지만, 본 원리들은 이러한 실시예들로 한정되는 것은 아니고, 따라서 본 원리들의 사상을 유지하면서, 깊이 정보를 이용하지 않는 다른 실시예들이 구현될 수 있다는 것을 이해해야 한다.

[0026] 또한, 여기서 사용되는 것으로서, "고 레벨 신택스(high level syntax)"는 계층적으로 마이크로블록층 위에 있는 비트스트림에 존재하는 신택스를 지칭한다. 예를 들어, 여기서 사용된 고 레벨 신택스는 슬라이스 헤더 레벨, SEI(Supplemental Enhancement Information) 레벨, PPS(Picture Parameter Set) 레벨, SPS(Sequence Parameter Set) 레벨, VPS(View Parameter Set) 및 NAL(Network Abstraction Layer) 유닛 헤더 레벨의 신택스를 지칭할 수 있다(그러나 이것으로 한정되진 않음).

[0027] ISO/IEC(International Organization for Standardization/International Electrotechnical Commission) MPEG-4(Moving Picture Experts Group-4) 파트 10 AVC(Advanced Video Coding) 표준/ITU-T(International Telecommunication Union, Telecommunication Sector) H.264 권고안(이하 "MPEG-4 AVC 표준"이라 함)에 기초한 다중-비디오 코딩(MVC)의 구현에 있어서, 참조 소프트웨어는 하나의 인코더로 각 뷰를 인코딩하고, 뷰-교차(cross-view) 참조들을 고려함으로써 다중-뷰 예측을 달성한다. 각 뷰는 원래의 해상도로 인코더에 의해 별도의 비트스트림으로서 코딩되고, 모든 비트스트림들은 추후 디코딩되는 단일 비트스트림을 형성하도록 조합된다. 각 뷰는 별도의 YUV 디코딩된 출력을 생성한다.

[0028] 다중-뷰 예측을 위한 다른 방법은 일 세트의 뷰들을 의사 뷰들(pseudo-views)로 그룹핑하는 것을 수반한다. 이 방법의 일 예에서, 가능한 다운샘플링 또는 다른 동작들로 큰 프레임 또는 수퍼 프레임(super frame)상의 총 M 뷰들(동시에 샘플링됨) 중 모든 N 뷰들로부터 화상들을 타일링할 수 있다. 도 1을 참조하면, 단일 프레임상에서 타일링된 4개의 뷰들의 예가 개괄적으로 참조 번호(100)에 의해 표시된다. 모든 4개의 뷰들은 그들의 정규 방향(normal orientation)내에 있다.

[0029] 도 2을 참조하면, 단일 프레임상에서 플립핑(flipping) 및 타일링된 4개의 뷰들의 예가 개괄적으로 참조 번호(200)에 의해 표시된다. 상부-좌측 뷰는 그 정규 방향내에 있다. 상부-우측 뷰는 수평으로 플립핑된다. 하부-좌측 뷰는 수직으로 플립핑된다. 하부-우측 뷰는 수평으로 및 수직으로 플립핑된다. 따라서, 4개의 뷰들이 존재하는 경우, 각 뷰로부터의 화상은 타일과 같은 수퍼-프레임으로 배열된다. 이는 큰 해상도를 갖는 단일의 비-코딩된 입력 시퀀스를 발생한다.

[0030] 대안적으로, 우리는 이미지를 다운샘플링하여 보다 작은 해상도를 생성할 수 있다. 따라서, 우리는 함께 타일링되는 상이한 뷰들을 각각 포함하는 다수의 시퀀스들을 생성할 수 있다. 그 후, 각각의 이러한 시퀀스는 의사 뷰를 형성하며, 각각의 의사 뷰는 N개의 상이한 타일링된 뷰들을 포함한다. 도 1은 하나의 의사-뷰를 도시하고, 도 2는 또 다른 의사-뷰를 도시한다. 그 후, 이 의사 뷰들은 ISO/IEC MPEG-2 표준 및 MPEG-4 AVC 표준과 같은 기존의 비디오 코딩 표준들을 이용하여 인코딩될 수 있다.

[0031] 다중-뷰 예측을 위한 또 다른 방법은 단순히 새로운 표준을 이용하여 독립적으로 상이한 뷰들을 인코딩하고, 디코딩 후에 재생기에서 필요로 하는 뷰들을 타일링하는 것을 수반한다.

[0032] 또한, 다른 방법에서, 뷰들은 픽셀-단위(pixel-wise) 방식으로 또한 타일링될 수 있다. 예를 들어, 4개의 뷰들로 구성된 수퍼 뷰에서, 픽셀(x,y)은 뷰 0으로부터 올 수 있고, 픽셀(x+1,y)은 뷰 1로부터 올 수 있고, 픽셀(x,y+1)은 뷰 2로부터 올 수 있고, 픽셀(x+1,y+1)은 뷰 3으로부터 올 수 있다.

[0033] 많은 디스플레이 제조자들은 단일 프레임상에 상이한 뷰들을 배열 또는 타일링하고, 그 후 그 각각의 위치들로부터 뷰들을 추출하여 렌더링하는 이러한 프레임워크를 이용한다. 이 경우들에서, 비트스트림이 이러한 특성을 갖는지를 판단한 어떠한 표준 방식도 존재하지 않는다. 따라서, 시스템이 큰 프레임에서 상이한 뷰들의 화상들을 타일링하는 방법을 이용하는 경우, 상이한 뷰들을 추출하는 방법은 사유적(proprietary)이다.

[0034] 그러나, 비트스트림이 이러한 특성을 갖는지를 판단하는 어떠한 표준 방식도 존재하지 않는다. 우리는 디스플레이 또는 다른 후처리를 돕도록 렌더기 또는 재생기가 이러한 정보를 추출하는 것을 용이하게 하게 위해 고 레벨 신택스를 제안한다. 또한, 서브-화상들이 상이한 해상도를 갖는 것도 가능하고 일부 업샘플링은 궁극적으로 뷰를 렌더링하기 위해 필요할 수 있다. 사용자는 업샘플링 방법이 고 레벨 신택스에 또한 표시되기를 원할 수 있다. 부가적으로, 깊이 포커스를 변경하기 위해 파라미터들이 또한 전송될 수 있다.

[0035] 일 실시예에서, 우리는 각 화상이 상이한 뷰에 속하는 서브-화상들을 포함하는 MPEG-4 AVC 표준 호환 비트스트림에서 다중-뷰 정보를 시그널링하는 새로운 SEI(Supplemental Enhancement Information) 메시지를 제안한다. 실시예는 예를 들어, 이러한 프레임워크를 이용할 수 있는 3-차원(3D) 모니터들상에 다중-뷰 비디오 스트림들의 쉽고 편리한 디스플레이를 위해 의도된다. 그 개념은 고 레벨 신택스를 이용하여 이러한 정보를 시그널링하는 다른 비디오 코딩 표준들 및 권고안들로 확장될 수 있다.

[0036] 또한, 일 실시예에서, 우리는 다중-뷰 비디오 인코더 및/또는 디코더로 송신되기 전에 뷰들을 배열하는 시그널링 방법을 제안한다. 유리하게는, 이 실시예는 다중 뷰 코딩의 단순화된 구현을 유도할 수 있고, 코딩 효율성에 이득이 있을 수 있다. 특정한 뷰들은 함께 모아질 수 있고, 의사 뷰 또는 수퍼 뷰를 형성할 수 있고, 그 후 타일링된 수퍼 뷰는 예를 들어, 다중-뷰 비디오 코딩의 구현을 기반으로 한 현 MPEG-4 AVC 표준에 따라 공통의 다중-뷰 비디오 인코더 및/또는 디코더에 의해 보통 뷰로서 취급된다. 의사 뷰들의 기술의 사용을 시그널링하도록 다중-뷰 비디오 코딩의 시퀀스 파라미터 세트(SPS)에서의 새로운 플래그가 제안된다. 일 실시예는 이러한 프레임워크를 이용할 수 있는 3D 모니터들상에 다중-뷰 비디오 스트림의 쉽고 편리한 디스플레이를 위해 의도된다.

[0037] **단일-뷰 비디오 인코딩/디코딩 표준/권고안을 이용한 인코딩/디코딩**

[0038] ISO/IEC(International Organization for Standardization/International Electrotechnical Commission) MPEG-4(Moving Picture Experts Group-4) 파트 10 AVC(Advanced Video Coding) 표준/ITU-T(International Telecommunication Union, Telecommunication Sector) H.264 권고안(이하 "MPEG-4 AVC 표준"이라 함)에 기초한 다중-비디오 코딩(MVC)의 현재 구현에 있어서, 참조 소프트웨어는 하나의 인코더로 각 뷰를 인코딩하고, 뷰-교차 참조들을 고려함으로써 다중-뷰 예측을 달성한다. 각 뷰는 원래의 해상도에서 인코더에 의해 별도의 비트스트림으로서 코딩되고, 모든 비트스트림들은 추후 디코딩되는 단일 비트스트림을 형성하도록 조합된다. 각 뷰는 별도의 YUV 디코딩된 출력을 생성한다.

[0039] 다중-뷰 예측을 위한 다른 방법은 가능한 다운스트림 동작으로 큰 프레임 또는 수퍼 프레임상의 각 뷰(동시에

샘플링됨)로부터 화상들을 타일링하는 것을 수반한다. 도 1을 참조하면, 단일 프레임상에서 타일링된 4개의 뷰들의 예가 개괄적으로 참조 번호(100)에 의해 표시된다. 도 2를 참조하면, 단일 프레임상에서 플립핑 및 타일링된 4개의 뷰들의 예가 개괄적으로 참조 번호(200)에 의해 표시된다. 따라서, 4개의 뷰들이 존재하는 경우, 각 뷰로부터의 화상은 타일과 같은 수퍼-프레임으로 배열된다. 이는 큰 해상도를 갖는 단일의 비-코딩된 입력 시퀀스를 발생한다. 그 후, 이 신호는 ISO/IEC MPEG-2 표준 및 MPEG-4 AVC 표준과 같은 기존의 비디오 코딩 표준들을 이용하여 인코딩될 수 있다.

[0040] 다중-뷰 예측을 위한 또 다른 방법은 단순히 새로운 표준을 이용하여 독립적으로 상이한 뷰들을 인코딩하고, 디코딩 후에 재생기에서 필요로 하는 뷰들을 타일링하는 것을 수반한다.

[0041] 많은 디스플레이 제조자들은 단일 프레임상에 상이한 뷰들을 배열 또는 타일링하고, 그 후 그 각각의 위치들로부터 뷰들을 추출하여 렌더링하는 이러한 프레임워크를 이용한다. 이 경우들에서, 비트스트림이 이러한 특성을 갖는지를 판단한 어떠한 표준 방식도 존재하지 않는다. 따라서, 시스템이 큰 프레임에서 상이한 뷰들의 화상들을 타일링하는 방법을 이용하는 경우, 상이한 뷰들을 추출하는 방법은 사유적이다.

[0042] 도 3을 참조하면, MPEG-4 AVC 표준에 따라 비디오 인코딩을 수행할 수 있는 비디오 인코더가 개괄적으로 참조 번호(300)에 의해 표시된다.

[0043] 비디오 인코더(300)는 조합기(385)의 비-반전 입력과 신호를 통신하도록 연결된 출력을 갖는 프레임 오더링 버퍼(310)를 포함한다. 조합기(385)의 출력은 변환기 및 양자화기(325)의 제 1 입력과 신호를 통신하도록 연결된다. 변환기 및 양자화기(325)의 출력은 엔트로피 코더(345)의 제 1 입력 및 역 변환기 및 역 양자화기(350)의 제 1 입력과 신호를 통신하도록 연결된다. 엔트로피 코더(345)의 출력은 조합기(390)의 제 1 비-반전 입력과 신호를 통신하도록 연결된다. 조합기(390)의 출력은 출력 버퍼(335)의 제 1 입력과 신호를 통신하도록 연결된다.

[0044] 인코더 제어기(305)의 제 1 출력은 프레임 오더링 버퍼(310)의 제 2 입력, 역 변환기 및 역 양자화기(350)의 제 2 입력, 화상-유형 판단 모듈(315)의 입력, 매크로 블록-유형(MB-유형) 판단 블록(320), 인트라 예측(intra prediction) 모듈(360)의 제 2 입력, 디블록킹 필터(365)의 제 2 입력, 모션 보상기(370)의 제 1 입력, 모션 추정기(375)의 제 1 입력, 참조 화상 버퍼(380)의 제 2 입력과 신호를 통신하도록 연결된다.

[0045] 인코더 제어기(305)의 제 2 입력은 SEI(Supplemental Enhancement Information) 삽입기(330)의 제 1 입력, 변환기 및 양자화기(325)의 제 2 입력, 엔트로피 코더(345)의 제 2 입력, 출력 버퍼(335)의 제 2 입력, 시퀀스 파라미터 세트(SPS) 및 화상 파라미터 세트(PPS) 삽입기(340)의 입력과 신호를 통신하도록 연결된다.

[0046] 화상-유형 판단 모듈(315)의 제 1 입력은 프레임 오더링 버퍼(310)의 제 3 입력과 신호를 통신하도록 연결된다. 화상-유형 판단 모듈(315)의 제 2 출력은 매크로 블록-유형 판단 모듈(320)의 제 2 입력과 신호를 통신하도록 연결된다.

[0047] 시퀀스 파라미터 세트(SPS) 및 화상 파라미터 세트(PPS) 삽입기(340)의 출력은 조합기(390)의 제 3 비-반전 입력과 신호를 통신하도록 연결된다. SEI 삽입기(330)의 출력은 조합기(390)의 제 2 비-반전 입력과 신호를 통신하도록 연결된다.

[0048] 역 양자화기 및 역 변환기(350)의 출력은 조합기(319)의 제 1 비-반전 입력과 신호를 통신하도록 연결된다. 조합기(319)의 출력은 인트라 예측 모듈(360)의 제 1 입력 및 디블록킹 필터(365)의 제 1 입력과 신호를 통신하도록 연결된다. 디블록킹 필터(365)의 출력은 참조 화상 버퍼(380)의 제 1 입력과 신호를 통신하도록 연결된다. 참조 화상 버퍼(380)의 출력은 모션 추정기(375)의 제 2 입력 및 모션 보상기(370)의 제 1 입력과 신호를 통신하도록 연결된다. 모션 추정기(375)의 제 1 출력은 모션 보상기(370)의 제 2 입력과 신호를 통신하도록 연결된다. 모션 추정기(375)의 제 2 출력은 엔트로피 코더(345)의 제 3 입력과 신호를 통신하도록 연결된다.

[0049] 모션 보상기(370)의 출력은 스위치(397)의 제 1 입력과 신호를 통신하도록 연결된다. 인트라 예측 모듈(360)의 출력은 스위치(397)의 제 2 입력과 신호를 통신하도록 연결된다. 매크로 블록-유형 판단 모듈(320)의 출력은 제어 신호를 스위치(397)에 제공하기 위해 스위치(397)의 제 3 입력과 신호를 통신하도록 연결된다. 스위치(397)의 제 3 입력은 스위치의 "데이터" 입력(제어 입력 즉, 제 3 입력과 비교해서)이 모션 보상기(370) 또는 인트라 예측 모듈(360)에 의해 제공되는지를 결정한다. 스위치(397)의 출력은 조합기(319)의 제 2 비-반전 입력 및 조합기(385)의 반전 입력과 신호를 통신하도록 연결된다.

[0050] 프레임 오더링 버퍼(310) 및 인코더 제어기(105)의 입력들은 입력 화상(301)을 수신하는 인코더(300)의 입력으

로서 이용 가능하다. 또한, SEI(Supplemental Enhancement Information) 삽입기(330)의 입력은 메타데이터를 수신하는 인코더(300)의 입력으로서 이용 가능하다. 출력 버퍼(335)의 출력은 비트스트림을 출력하는 인코더(300)의 출력으로서 이용 가능하다.

[0051] 도 4를 참조하면, MPEG-4 AVC 표준에 따라 비디오 디코딩을 수행할 수 있는 비디오 디코더가 개괄적으로 참조 번호(400)에 의해 표시된다.

[0052] 비디오 디코더(400)는 엔트로피 디코더(445)의 제 1 입력과 신호를 통신하도록 연결된 출력을 갖는 입력 버퍼(410)를 포함한다. 엔트로피 디코더(445)의 제 1 출력은 역 변환기 및 역 양자화기(450)의 제 1 입력과 신호를 통신하도록 연결된다. 역 변환기 및 역 양자화기(450)의 출력은 조합기(425)의 제 2 비-반전 입력과 신호를 통신하도록 연결된다. 조합기(425)의 출력은 디블록킹 필터(465)의 제 2 입력 및 인트라 예측 모듈(460)의 제 1 입력과 신호를 통신하도록 연결된다. 디블록킹 필터(465)의 제 2 출력은 참조 화상 버퍼(480)의 제 1 입력과 신호를 통신하도록 연결된다. 참조 화상 버퍼(480)의 출력은 모션 보상기(470)의 제 2 입력과 신호를 통신하도록 연결된다.

[0053] 엔트로피 디코더(445)의 제 2 입력은 모션 보상기(470)의 제 3 입력 및 디블록킹 필터(465)의 제 1 입력과 신호를 통신하도록 연결된다. 엔트로피 디코더(445)의 제 3 출력은 디코더 제어기(405)의 입력과 신호를 통신하도록 연결된다. 디코더 제어기(405)의 제 1 출력은 엔트로피 디코더(445)의 제 2 입력과 신호를 통신하도록 연결된다. 디코더 제어기(405)의 제 2 출력은 역 변환기 및 역 양자화기(450)의 제 2 입력과 신호를 통신하도록 연결된다. 제어기(405)의 제 3 입력은 디블록킹 필터(465)의 제 3 입력과 신호를 통신하도록 연결된다. 디코더 제어기(405)의 제 4 출력은 인트라 예측 모듈(460)의 제 2 입력, 모션 보상기(470)의 제 1 입력 및 참조 화상 버퍼(480)의 제 2 입력과 신호를 통신하도록 연결된다.

[0054] 모션 보상기(470)의 출력은 스위치(497)의 제 1 입력과 신호를 통신하도록 연결된다. 인트라 예측 모듈(460)의 출력은 스위치(497)의 제 2 입력과 신호를 통신하도록 연결된다. 스위치(497)의 출력은 조합기(425)의 제 1 비-반전 입력과 신호를 통신하도록 연결된다.

[0055] 입력 버퍼(410)의 입력은 입력 스트림을 수신하는 디코더(400)의 입력으로서 이용 가능하다. 디블록킹 필터(465)의 제 1 출력은 출력 화상을 출력하는 디코더(400)의 출력으로서 이용 가능하다.

[0056] 도 5a 및 도 5b를 참조하면, MPEG-4 AVC 표준을 이용하여 복수의 뷰들에 대한 화상들을 인코딩하는 예시적인 방법이 개괄적으로 참조 번호(500)에 의해 표시된다.

[0057] 방법(500)은 제어를 기능 블록(504)에 넘겨주는 시작 블록(502)을 포함한다. 기능 블록(504)은 특정 시간 인스턴스의 각 뷰를 타일 포멧의 서브-화상으로서 배열하고, 제어를 기능 블록(506)에 넘겨준다. 기능 블록(506)은 신택스 요소(syntax element) num_coded_view_minus1를 설정하고, 제어를 기능 블록(508)에 넘겨준다. 기능 블록(508)은 신택스 요소들 org_pic_width_in_mbs_minus1 및 org_pic_height_in_mbs_minus1을 설정하고, 제어를 기능 블록(510)에 넘겨준다. 기능 블록(510)은 변수(i)를 0으로 설정하고, 제어를 판단 블록(512)에 넘겨준다. 판단 블록(512)은 변수(i)가 뷰들의 수보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(514)에 넘겨준다. 그렇지 않다면, 제어는 기능 블록(524)에 넘겨준다.

[0058] 기능 블록(514)은 신택스 요소 view_id[i]를 설정하고, 제어를 기능 블록(516)에 넘겨준다. 기능 블록(516)은 신택스 요소 num_parts[view_id[i]]를 설정하고, 제어를 기능 블록(518)에 넘겨준다. 기능 블록(518)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(520)에 넘겨준다. 판단 블록(520)은 변수(j)의 현재 값이 신택스 요소 num_parts[view_id[i]]의 현재 값보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(522)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(528)에 넘겨진다.

[0059] 기능 블록(522)은 이하의 신택스 요소들을 설정하고, 변수(j)를 증가시키고, 그 후 제어를 판단 블록(520)에 반환한다:

[0060] depth_flag[view_id[i]][j]; flip_dir[view_id[i]][j]; loc_left_offset[view_id[i]][j]; loc_top_offset[view_id[i]][j]; frame_crop_left_offset[view_id[i]][j]; frame_crop_right_offset[view_id[i]][j]; frame_crop_top_offset[view_id[i]][j]; and frame_crop_bottom_offset[view_id[i]][j].

[0061] 기능 블록(528)은 신택스 요소 upsample_view_flag[view_id[i]]를 설정하고, 제어를 판단 블록(530)으로 넘겨준다. 판단 블록(530)은 신택스 요소 upsample_view_flag[view_id[i]]의 값이 1인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(532)으로 넘겨진다. 그렇지 않다면, 제어는 판단 블록(534)으로 넘겨진다.

[0062] 기능 블록(532)은 신택스 요소 upsample_filter[view_id[i]]를 설정하고, 제어를 판단 블록(534)에 넘겨준다.

[0063] 판단 블록(534)은 신택스 요소 upsample_filter[view_id[i]]의 현재 값이 3인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(536)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(540)으로 넘겨진다.

[0064] 기능 블록(536)은 다음의 신택스 요소들 즉, vert_dim[view_id[i]]; hor_dim[view_id[i]]; 및 quantizer[view_id[i]]을 설정하고, 제어를 기능 블록(538)에 넘겨준다.

[0065] 기능 블록(538)은 각 YUV 컴포넌트에 대한 필터 계수들을 설정하고, 제어를 제어 블록(540)에 넘겨준다.

[0066] 기능 블록(540)은 변수(i)를 증가시키고, 제어를 판단 블록(512)에 반환한다.

[0067] 기능 블록(524)은 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), SEI(Supplemental Enhancement Information) 메시지, NAL(Network Abstraction Layer) 유닛 헤더, 및 슬라이스 헤더 중 적어도 하나에 이 신택스 요소들을 기록하고, 제어를 기능 블록(526)에 넘겨준다. 기능 블록(526)은 MPEG-4 AVC 표준 또는 다른 단일 뷰 코덱을 이용하여 각 화상을 인코딩하고, 제어를 종료 블록(599)에 넘겨준다.

[0068] 도 6a 및 도 6b를 참조하면, MPEG-4 AVC 표준을 이용하여 복수의 뷰들에 대한 화상을 디코딩하는 예시적인 방법이 개괄적으로 참조 번호(600)에 의해 표시된다.

[0069] 방법(600)은 기능 블록(604)에 제어를 넘겨주는 시작 블록(602)을 포함한다. 기능 블록(604)은 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), SEI(Supplemental Enhancement Information) 메시지, NAL(Network Abstraction Layer) 유닛 헤더 및 슬라이스 헤더 중 적어도 하나로부터 이하의 신택스 요소들을 파싱(parsing)하고, 제어를 기능 블록(606)에 넘겨준다. 기능 블록(606)은 신택스 요소 num_coded_views_minus1을 파싱하고, 제어를 기능 블록(608)에 넘겨준다. 기능 블록(608)은 신택스 요소들 org_pic_width_in_mbs_minus1 및 org_pic_height_in_mbs_minus1을 파싱하고, 제어를 기능 블록(610)에 넘겨준다. 기능 블록(610)은 변수(i)를 0으로 설정하고, 제어를 판단 블록(612)에 넘겨준다. 판단 블록(612)은 변수(i)가 뷰들의 수보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(614)에 넘겨진다. 그렇지 않다면, 제어는 기능 블록(624)에 넘겨진다.

[0070] 기능 블록(614)은 신택스 요소 view_id[i]를 파싱하고, 제어를 기능 블록(616)에 넘겨준다. 기능 블록(616)은 신택스 요소 num_parts_minus1[view_id[i]]를 파싱하고, 제어를 기능 블록(618)에 넘겨준다. 기능 블록(618)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(620)에 넘겨준다. 판단 블록(620)은 변수(j)의 현재 값이 신택스 요소 num_parts[view_id[i]]의 값보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(622)에 넘겨진다. 그렇지 않다면, 제어는 기능 블록(628)에 넘겨진다.

[0071] 기능 블록(622)은 이하의 신택스 요소들을 파싱하고, 변수(j)를 증가시키고, 그 후 제어를 판단 블록(620)에 반환한다:

[0072] depth_flag[view_id[i][j]; flip_dir[view_id[i][j]; loc_left_offset[view_id[i][j];
loc_top_offset[view_id[i][j]; frame_crop_left_offset[view_id[i][j];
frame_crop_right_offset[view_id[i][j]; frame_crop_top_offset[view_id[i][j]; and
frame_crop_bottom_offset[view_id[i][j].

[0073] 기능 블록(628)은 신택스 요소 upsample_view_flag[view_id[i]]를 파싱하고, 제어를 판단 블록(630)에 넘겨준다. 판단 블록(630)은 신택스 요소 upsample_view_flag[view_id[i]]의 현재 값이 1인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(632)에 넘겨진다. 그렇지 않다면 제어는 판단 블록(634)에 넘겨진다.

[0074] 기능 블록(632)은 신택스 요소 upsample_filter[view_id[i]]를 파싱하고, 제어를 판단 블록(634)에 넘겨준다.

[0075] 판단 블록(634)은 신택스 요소 upsample_filter[view_id[i]]의 현재 값이 3인지를 결정한다. 만약 그렇다면 제어는 기능 블록(636)에 넘겨진다. 그렇지 않다면 제어는 기능 블록(640)에 넘겨진다.

[0076] 기능 블록(636)은 이하의 신택스 요소들 즉, vert_dim[view_id[i]], hor_dim[view_id[i]] 및 quantizer[view_id[i]]를 파싱하고, 제어를 기능 블록(638)에 넘겨준다.

[0077] 기능 블록(638)은 각 YUV 컴포넌트를 위한 필터 계수들을 파싱하고, 제어를 기능 블록(640)에 넘겨준다.

[0078] 기능 블록(640)은 변수(i)를 증가시키고, 제어를 판단 블록(612)에 반환한다.

[0079] 기능 블록(624)은 MPEG-4 AVC 표준 또는 다른 단일 뷰 코덱을 이용하여 각 화상을 디코딩하고, 제어를 기능 블록(626)에 넘겨준다. 기능 블록(626)은 고 레벨 신택스를 이용하여 화상으로부터 각 뷰를 분리하고, 제어를 종료 블록(699)으로 넘겨준다.

[0080] 도 7a 및 도 7b를 참조하면, MPEG-4 AVC 표준을 이용하여 복수의 뷰들 및 깊이들에 대한 화상을 인코딩하는 예시적인 방법이 개괄적으로 참조 번호(700)에 의해 표시된다.

[0081] 방법(700)은 기능 블록(704)에 제어를 넘겨주는 시작 블록(702)을 포함한다. 기능 블록(704)은 파일 포멧의 서브 화상으로서 특정 시간 인스턴스의 각 뷰 및 대응하는 깊이를 배열하고, 제어를 기능 블록(706)에 넘겨준다. 기능 블록(706)은 신택스 요소 num_coded_view_minus1를 설정하고, 제어를 기능 블록(708)에 넘겨준다. 기능 블록(708)은 신택스 요소들 org_pic_width_in_mbs_minus1 및 org_pic_height_in_mbs_minus1을 설정하고, 제어를 기능 블록(710)에 넘겨준다. 기능 블록(710)은 변수(i)를 0으로 설정하고, 제어를 판단 블록(712)에 넘겨준다. 판단 블록(712)은 변수(i)가 뷰들의 수보다 작은지를 결정한다. 만약 그렇다면, 제어를 기능 블록(714)에 넘겨준다. 그렇지 않다면, 제어를 기능 블록(724)에 넘겨준다.

[0082] 기능 블록(714)은 신택스 요소 view_id[i]를 설정하고, 제어를 기능 블록(716)에 넘겨준다. 기능 블록(716)은 신택스 요소 num_parts[view_id[i]]를 설정하고, 제어를 기능 블록(718)에 넘겨준다. 기능 블록(718)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(720)에 넘겨준다. 판단 블록(720)은 변수(j)의 현재 값이 신택스 요소 num_parts[view_id[i]]의 현재 값보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(722)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(728)에 넘겨진다.

[0083] 기능 블록(722)은 이하의 신택스 요소들을 설정하고, 변수(j)를 증가시키고, 그 후 제어를 판단 블록(720)에 반환한다:

[0084] depth_flag[view_id[i]][j]; flip_dir[view_id[i]][j]; loc_left_offset[view_id[i]][j]; loc_top_offset[view_id[i]][j]; frame_crop_left_offset[view_id[i]][j]; frame_crop_right_offset[view_id[i]][j]; frame_crop_top_offset[view_id[i]][j]; and frame_crop_bottom_offset[view_id[i]][j].

[0085] 기능 블록(728)은 신택스 요소 upsample_view_flag[view_id[i]]를 설정하고, 제어를 판단 블록(730)으로 넘겨준다. 판단 블록(730)은 신택스 요소 upsample_view_flag[view_id[i]]의 값이 1인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(732)으로 넘겨진다. 그렇지 않다면, 제어는 판단 블록(734)으로 넘겨진다.

[0086] 기능 블록(732)은 신택스 요소 upsample_filter[view_id[i]]를 설정하고, 제어를 판단 블록(734)에 넘겨준다.

[0087] 판단 블록(734)은 신택스 요소 upsample_filter[view_id[i]]의 현재 값이 3인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(736)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(740)으로 넘겨진다.

[0088] 기능 블록(736)은 다음의 신택스 요소들 즉, vert_dim[view_id[i]]; hor_dim[view_id[i]]; 및 quantizer[view_id[i]]을 설정하고, 제어를 기능 블록(738)에 넘겨준다.

[0089] 기능 블록(738)은 각 YUV 컴포넌트에 대한 필터 계수들을 설정하고, 제어를 제어 블록(740)에 넘겨준다.

[0090] 기능 블록(740)은 변수(i)를 증가시키고, 제어를 판단 블록(712)에 반환한다.

[0091] 기능 블록(724)은 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), SEI(Supplemental Enhancement Information) 메시지, NAL(Network Abstraction Layer) 유닛 헤더, 및 슬라이스 헤더 중 적어도 하나에 이 신택스 요소들을 기록하고, 제어를 기능 블록(726)에 넘겨준다. 기능 블록(726)은 MPEG-4 AVC 표준 또는 다른 단일 뷰 코덱을 이용하여 각 화상을 인코딩하고, 제어를 종료 블록(799)에 넘겨준다.

[0092] 도 8a 및 도 8b를 참조하면, MPEG-4 AVC 표준을 이용하여 복수의 뷰들 및 깊이들에 대한 화상을 디코딩하는 예시적인 방법이 개괄적으로 참조 번호(800)에 의해 표시된다.

[0093] 방법(800)은 기능 블록(804)에 제어를 넘겨주는 시작 블록(802)을 포함한다. 기능 블록(804)은 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), SEI(Supplemental Enhancement Information) 메시지, NAL(Network Abstraction Layer) 유닛 헤더 및 슬라이스 헤더 중 적어도 하나로부터 이하의 신택스 요소들을 파싱하고, 제어를 기능 블록(806)에 넘겨준다. 기능 블록(806)은 신택스 요소 num_coded_views_minus1을 파싱하고, 제어를 기능 블록(808)에 넘겨준다. 기능 블록(808)은 신택스 요소들 org_pic_width_in_mbs_minus1 및 org_pic_height_in_mbs_minus1을 파싱하고, 제어를 기능 블록(810)에 넘겨준다. 기능 블록(810)은 변수(i)를

0으로 설정하고, 제어를 판단 블록(812)에 넘겨준다. 판단 블록(812)은 변수(i)가 뷰들의 수보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(814)에 넘겨진다. 그렇지 않다면, 제어는 기능 블록(824)에 넘겨진다.

[0094] 기능 블록(814)은 신택스 요소 view_id[i]를 파싱하고, 제어를 기능 블록(816)에 넘겨준다. 기능 블록(816)은 신택스 요소 num_parts_minus1[view_id[i]]를 파싱하고, 제어를 기능 블록(818)에 넘겨준다. 기능 블록(818)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(820)에 넘겨준다. 판단 블록(820)은 변수(j)의 현재 값이 신택스 요소 num_parts[view_id[i]]의 값보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(822)에 넘겨진다. 그렇지 않다면, 제어는 기능 블록(828)에 넘겨진다.

[0095] 기능 블록(822)은 이하의 신택스 요소들을 파싱하고, 변수(j)를 증가시키고, 그 후 제어를 판단 블록(820)에 반환한다:

[0096] depth_flag[view_id[i]][j]; flip_dir[view_id[i]][j]; loc_left_offset[view_id[i]][j]; loc_top_offset[view_id[i]][j]; frame_crop_left_offset[view_id[i]][j]; frame_crop_right_offset[view_id[i]][j]; frame_crop_top_offset[view_id[i]][j]; and frame_crop_bottom_offset[view_id[i]][j].

[0097] 기능 블록(828)은 신택스 요소 upsample_view_flag[view_id[i]]를 파싱하고, 제어를 판단 블록(830)에 넘겨준다. 판단 블록(830)은 신택스 요소 upsample_view_flag[view_id[i]]의 현재 값이 1인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(832)에 넘겨진다. 그렇지 않다면 제어는 판단 블록(834)에 넘겨진다.

[0098] 기능 블록(832)은 신택스 요소 upsample_filter[view_id[i]]를 파싱하고, 제어를 판단 블록(834)에 넘겨준다.

[0099] 판단 블록(834)은 신택스 요소 upsample_filter[view_id[i]]의 현재 값이 3인지를 결정한다. 만약 그렇다면 제어는 기능 블록(836)에 넘겨진다. 그렇지 않다면 제어는 기능 블록(840)에 넘겨진다.

[0100] 기능 블록(836)은 이하의 신택스 요소들 즉, vert_dim[view_id[i]], hor_dim[view_id[i]] 및 quantizer[view_id[i]]를 파싱하고, 제어를 기능 블록(838)에 넘겨준다.

[0101] 기능 블록(838)은 YUV 컴포넌트를 위한 필터 계수들을 파싱하고, 제어를 기능 블록(840)에 넘겨준다.

[0102] 기능 블록(840)은 변수(i)를 증가시키고, 제어를 판단 블록(812)에 반환한다.

[0103] 기능 블록(824)은 MPEG-4 AVC 표준 또는 다른 단일 뷰 코덱을 이용하여 각 화상을 디코딩하고, 제어를 기능 블록(826)에 넘겨준다. 기능 블록(826)은 고 레벨 신택스를 이용하여 화상으로부터 각 뷰 및 대응하는 깊이를 분리하고, 제어를 기능 블록(827)에 넘겨준다. 기능 블록(827)은 추출된 뷰 및 깊이 신호들을 이용하여 뷰 파싱을 잠재적으로 수행하고, 종료 블록(899)으로 넘겨준다.

[0104] 도 7 및 도 8에서 사용된 깊이에 대해, 도 9은 깊이가 각 대응하는 이미지의 위치(도시되지 않음)에 대한 픽셀 값으로서 제공되는 깊이 신호(900)의 예를 도시한다. 또한, 도 10은 타일(1000)에 포함된 2개의 깊이 신호들의 예를 도시한다. 타일(1000)의 상부-우측 부분은 타일(1000)의 상부-좌측상의 이미지에 대응하는 깊이 값들을 갖는 깊이 신호이다. 타일(1000)의 하부-우측 부분은 타일(1000)의 하부-좌측상의 이미지에 대응하는 깊이 값들을 갖는 깊이 신호이다.

[0105] 도 11을 참조하면, 단일 프레임상에 타일링된 5개의 뷰들의 예가 개괄적으로 참조 번호(1100)에 의해 표시된다. 상부 4개의 뷰들은 정규 방향에 있다. 5번째 뷰 또한 정규 방향에 있지만, 타일(1100)의 하부를 따라 2개의 부분들로 나뉜다. 5번째 뷰의 좌측-부분은 5번째 뷰의 "상부"를 도시하고, 5번째 뷰의 우측-부분은 5번째 뷰의 "하부"를 도시한다.

[0106] **다중-뷰 비디오 인코딩/디코딩 표준/권고안을 이용한 인코딩/디코딩**

[0107] 도 12를 참조하면, 예시적인 다중-뷰 비디오 코딩(MVC) 인코더가 개괄적으로 참조 번호(1200)에 의해 표시된다. 인코더(1200)는 변환기(1210)의 입력과 신호를 통신하도록 연결된 출력을 갖는 조합기(1205)를 포함한다. 변환기(1210)의 출력은 양자화기(1215)의 입력과 신호를 통신하도록 연결된다. 양자화기(1215)의 출력은 엔트로피 코더(1220)의 입력 및 역 양자화기(1225)의 입력과 신호를 통신하도록 연결된다. 역 양자화기(1225)의 출력은 역 변환기(1230)의 입력과 신호를 통신하도록 연결된다. 역 변환기(1230)의 출력은 조합기(1235)의 제 1 비-반전 입력과 신호를 통신하도록 연결된다. 조합기(1235)의 출력은 인트라 예측기(1245)의 입력 및 디블록킹 필터(1250)의 입력과 신호를 통신하도록 연결된다. 디블록킹 필터(1250)의 출력은 참조 화상 저장소(1255)(뷰 i에

대한)의 입력과 신호를 통신하도록 연결된다. 참조 화상 저장소(1255)의 출력은 모션 보상기(1275)의 제 1 입력 및 모션 추정기(1280)의 제 1 입력과 신호를 통신하도록 연결된다. 모션 추정기(1280)의 출력은 모션 보상기(1275)의 제 2 입력과 신호를 통신하도록 연결된다.

[0108] 참조 화상 저장소(1260)(다른 뷰들에 대한)의 출력은 디스패리티 추정기(disparity estimator;1270)의 제 1 입력 및 디스패리티 보상기(1265)의 제 1 입력과 신호를 통신하도록 연결된다. 디스패리티 추정기(1270)의 출력은 디스패리티 보상기(1265)의 제 2 입력과 신호를 통신하도록 연결된다.

[0109] 엔트로피 디코더(1220)의 출력은 인코더(1200)의 출력으로서 이용 가능하다. 조합기(1205)의 비-반전 입력은 인코더(1200)의 입력으로서 이용 가능하고, 디스패리티 추정기(1270)의 제 2 입력 및 모션 추정기(1280)의 제 2 입력과 신호를 통신하도록 연결된다. 스위치(1285)의 출력은 조합기(1235)의 제 2 비-반전 입력 및 조합기(1205)의 반전 입력과 신호를 통신하도록 연결된다. 스위치(1285)는 모션 보상기(1275)의 출력과 신호를 통신하도록 연결된 제 1 입력, 디스패리티 보상기(1265)의 출력과 신호를 통신하도록 연결된 제 2 입력 및 인트라 예측기(1245)의 출력과 신호를 통신하도록 연결된 제 3 입력을 포함한다.

[0110] 모드 판단 모듈(1240)은 어느 입력이 스위치(1285)에 의해 선택되는지를 제어하도록 스위치(1285)에 연결된 출력을 갖는다.

[0111] 도 13을 참조하면, 예시적인 다중-뷰 비디오 코딩(MVC) 디코더가 개괄적으로 참조 번호(1300)에 의해 표시된다. 디코더(1300)는 역 양자화기(1310)의 입력과 신호를 통신하도록 연결된 출력을 갖는 엔트로피 디코더(1305)를 포함한다. 역 양자화기의 출력은 역 변환기(1315)의 입력과 신호를 통신하도록 연결된다. 역 변환기(1315)의 출력은 조합기(1320)의 제 1 비-반전 입력과 신호를 통신하도록 연결된다. 조합기(1320)의 출력은 디블록킹 필터(1325)의 입력 및 인트라 예측기(1330)의 입력과 신호를 통신하도록 연결된다. 디블록킹 필터(1325)의 출력은 참조 화상 저장소(1340)(뷰 i에 대한)의 입력과 신호를 통신하도록 연결된다. 참조 화상 저장소(1340)의 출력은 모션 보상기(1335)의 제 1 입력과 신호를 통신하도록 연결된다.

[0112] 참조 화상 저장소(1345)(다른 뷰들에 대한)의 출력은 디스패리티 보상기(1350)의 제 1 입력과 신호를 통신하도록 연결된다.

[0113] 엔트로피 코더(1305)의 입력은 잔여 비트스트림을 수신하도록 디코더(1300)에 대한 입력으로서 이용 가능하다. 또한, 모드 모듈(1360)의 입력은 어느 입력이 스위치(1355)에 의해 선택되었는지를 제어하도록 제어 신택스를 수신하기 위해 디코더(1300)에 대한 입력으로서 또한 이용 가능하다. 또한, 모션 보상기(1335)의 제 2 입력은 모션 벡터들을 수신하도록 디코더(1300)의 입력으로서 이용 가능하다. 또한, 디스패리티 보상기(1350)의 제 2 입력은 디스패리티 벡터들을 수신하도록 디코더(1300)에 대한 입력으로서 이용 가능하다.

[0114] 스위치(1355)의 출력은 조합기(1320)의 제 2 비-반전 입력과 신호를 통신하도록 연결된다. 스위치(1355)의 제 1 입력은 디스패리티 보상기(1350)의 출력과 신호를 통신하도록 연결된다. 스위치(1355)의 제 2 입력은 모션 보상기(1335)의 출력과 신호를 통신하도록 연결된다. 스위치(1355)의 제 3 입력은 인트라 예측기(1330)의 출력과 신호를 통신하도록 연결된다. 모드 모듈(1360)의 출력은 어느 입력이 스위치(1355)에 의해 선택되는지를 제어하도록 스위치(1355)와 신호를 통신하도록 연결된다. 디블록킹 필터(1325)의 출력은 디코더(1300)의 출력으로서 이용 가능하다.

[0115] 도 14를 참조하면, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용한 화상들의 인코딩에 대비하여 복수의 뷰들에 대한 화상들을 처리하는 예시적인 방법이 개괄적으로 참조 번호(1400)에 의해 개괄적으로 표시된다.

[0116] 방법(1400)은 제어를 기능 블록(1410)으로 넘겨주는 시작 블록(1405)을 포함한다. 기능 블록(1410)은 타일 포멧의 수퍼-화상으로서 특정 시간 인스턴스에서 총 M 뷰들 사이에서 모든 N 뷰들을 배열하고, 제어를 기능 블록(1415)에 넘겨준다. 기능 블록(1415)은 신택스 요소 num_coded_views_minus1을 설정하고, 제어를 기능 블록(1420)에 넘겨준다. 기능 블록(1420)은 모든(num_coded_views_minus1+1) 뷰들에 대해 신택스 요소 view_id[i]를 설정하고, 제어를 기능 블록(1425)에 넘겨준다. 기능 블록(1425)은 앵커 화상들(anchor pictures)들에 대해 인터-뷰(inter-view) 참조 종속 정보를 설정하고, 제어를 기능 블록(1430)에 넘겨준다. 기능 블록(1430)은 비-앵커 화상들에 대해 인터-뷰 참조 종속 정보를 설정하고, 제어를 기능 블록(1435)에 넘겨준다. 기능 블록(1435)은 신택스 요소 pseudo_view_present_flag를 설정하고, 제어를 판단 블록(1440)에 넘겨준다. 판단 블록(1440)은 신택스 요소 pseudo_view_present_flag의 현재 값이 참(true)인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1445)에 넘겨진다. 그렇지 않다면, 제어는 종료 블록(1499)에 넘겨진다.

[0117] 기능 블록(1445)은 이하의 신택스 요소들 즉, tiling_mode; org_pic_width_in_mbs_minus1; 및 org_pic_height_in_mbs_minus1을 설정하고, 제어를 기능 블록(1450)에 넘겨준다. 기능 블록(1450)은 각 코딩된 뷰에 대해 신택스 요소 pseudo_view_info(view_id)를 호출하고, 제어를 종료 블록(1499)에 넘겨준다.

[0118] 도 15a 및 도 15b를 참조하면, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용하여 복수의 뷰들에 대한 화상들을 인코딩하는 예시적인 방법이 개괄적으로 참조 번호(1500)에 의해 표시된다.

[0119] 방법(1500)은 입력 파라미터 pseudo_view_id를 가지며 제어를 기능 블록(1504)에 넘겨주는 시작 블록(1502)을 포함한다. 기능 블록(1504)은 신택스 요소 num_sub_views_minus1을 설정하고, 제어를 기능 블록(1506)에 넘겨준다. 기능 블록(1506)은 변수(i)를 0으로 설정하고 제어를 판단 블록(1508)으로 넘겨준다. 판단 블록(1508)은 변수(i)가 sub_views의 수보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1510)으로 넘겨진다. 그렇지 않다면, 제어를 기능 블록(1520)으로 넘겨준다.

[0120] 기능 블록(1510)은 신택스 요소 sub_view_id[i]을 설정하고, 제어를 기능 블록(1512)으로 넘겨준다. 기능 블록(1512)은 신택스 요소 num_parts_minus1[sub_view_id[i]]를 설정하고, 제어를 기능 블록(1514)으로 넘겨준다. 기능 블록(1514)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(1516)으로 넘겨준다. 판단 블록(1516)은 변수(j)가 신택스 요소 num_parts_minus1[sub_view_id[i]]보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1518)으로 넘겨진다. 그렇지 않다면, 제어는 판단 블록(1522)으로 넘겨진다.

[0121] 기능 블록(1518)은 이하의 신택스 요소들을 설정하고, 변수(j)를 증가시키고, 제어를 판단 블록(1516)으로 반환한다:

[0122] loc_left_offset[sub_view_id[i]][j]; loc_top_offset[sub_view_id[i]][j];
frame_crop_left_offset[sub_view_id[i]][j]; frame_crop_right_offset[sub_view_id[i]][j];
frame_crop_top_offset[sub_view_id[i]][j]; and
frame_crop_bottom_offset[sub_view_id[i][j].

[0123] 기능 블록(1520)은 다중-뷰 비디오 코딩(MVC)을 이용하여 현재 뷰에 대한 현재 화상을 인코딩하고, 제어를 종료 블록(1599)에 넘겨준다.

[0124] 판단 블록(1522)은 신택스 요소 tiling_mode가 0인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1524)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(1538)으로 넘겨진다.

[0125] 기능 블록(1524)은 신택스 요소 flip_dir[sub_view_id[i]] 및 신택스 요소 upsample_view_flag[sub_view_id[i]]를 설정하고, 제어를 판단 블록(1526)으로 넘겨준다. 판단 블록(1526)은 신택스 요소 upsample_view_flag[sub_view_id[i]]가 0인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1528)으로 넘겨진다. 그렇지 않다면, 제어는 판단 블록(1530)으로 넘겨진다.

[0126] 기능 블록(1528)은 신택스 요소 upsample_filter[sub_view_id[i]]를 설정하고, 제어를 판단 블록(1530)으로 넘겨진다. 판단 블록(1530)은 신택스 요소 upsample_filter[sub_view_id[i]]가 3인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1532)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(1536)으로 넘겨진다.

[0127] 기능 블록(1532)은 이하의 신택스 요소들 즉, vert_dim[sub_view_id[i]]; hor_dim[sub_view_id[i]]; 및 quantizer[sub_view_id[i]]를 설정하고, 제어를 기능 블록(1534)으로 넘겨준다. 기능 블록(1534)은 각 YUV 컴포넌트에 대한 필터 계수들을 설정하고, 제어를 기능 블록(1536)으로 넘겨준다.

[0128] 기능 블록(1536)은 변수(i)를 증가시키고, 제어를 판단 블록(1508)으로 반환한다.

[0129] 기능 블록(1538)은 신택스 요소 pixel_dist_x[sub_view_id[i]] 및 신택스 요소 flip_dist_y[sub_view_id[i]]를 설정하고, 제어를 기능 블록(1540)으로 넘겨준다. 기능 블록(1540)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(1542)으로 넘겨준다. 판단 블록(1542)은 변수(j)의 현재 값이 신택스 요소 num_parts[sub_view_id[i]]의 현재 값보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1544)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(1536)으로 넘겨진다.

[0130] 기능 블록(1544)은 신택스 요소 num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]를 설정하고, 제어를 기능 블록(1546)으로 넘겨준다. 기능 블록(1546)은 모든 픽셀 타일링 필터들에 대한 계수들을 설정하고, 제어를 기능 블록(1536)으로 넘겨준다.

[0131] 도 16을 참조하면, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용한 화상들의 디코딩에 대비하여 복

수의 뷰들에 대한 화상들을 처리하는 예시적인 방법이 개괄적으로 도면 번호(1600)에 의해 표시된다.

[0132] 방법(1600)은 기능 블록(1615)에 제어를 넘겨주는 시작 블록(1605)을 포함한다. 기능 블록(1615)은 신택스 요소 num_coded_views_minus1를 파싱하고, 제어를 기능 블록(1620)으로 넘겨준다. 기능 블록(1620)은 모든 (num_coded_views_minus1 +1) 뷰들에 대한 신택스 요소 view_id[i]를 파싱하고, 제어를 기능 블록(1625)으로 넘겨준다. 기능 블록(1625)은 앵커 화상들에 대한 인터-뷰 참조 종속 정보를 파싱하고, 제어를 기능 블록(1630)으로 넘겨준다. 기능 블록(1630)은 비-앵커 화상들에 대한 인터-뷰 참조 종속 정보를 파싱하고 제어를 기능 블록(1635)에 넘겨준다. 기능 블록(1635)은 신택스 요소 pseudo_view_present_flag를 파싱하고, 제어를 판단 블록(1640)으로 넘겨준다. 판단 블록(1640)은 신택스 요소pseudo_view_present_flag의 현재 값이 참인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1645)으로 넘겨진다. 그렇지 않다면, 제어는 종료 블록(1699)으로 넘겨진다.

[0133] 기능 블록(1645)은 이하의 신택스 요소들, 즉 tiling_mode; org_pic_width_in_mbs_minus1; 및 org_pic_height_in_mbs_minus1를 파싱하고, 제어를 기능 블록(1650)으로 넘겨준다. 기능 블록(1650)은 각 코딩된 뷰에 대해 신택스 요소 pseudo_view_info(view_id)를 호출하고, 제어를 종료 블록(1699)으로 넘겨준다.

[0134] 도 17a 및 도 17b를 참조하면, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용하여 복수의 뷰들에 대한 화상들을 디코딩하는 예시적인 방법이 개괄적으로 참조 번호(1700)에 의해 표시된다.

[0135] 방법(1700)은 입력 파라미터 pseudo_view_id를 갖고 시작하며 제어를 기능 블록(1704)에 넘겨주는 시작 블록(1702)을 포함한다. 기능 블록(1704)은 신택스 요소 num_sub_views_minus1을 파싱하고, 제어를 기능 블록(1706)에 넘겨준다. 기능 블록(1706)은 변수(i)를 0으로 설정하고, 제어를 판단 블록(1708)으로 넘겨준다. 판단 블록(1708)은 변수(i)가 sub_views의 수보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1710)으로 넘겨진다. 그렇지 않다면, 제어를 기능 블록(1720)으로 넘겨준다.

[0136] 기능 블록(1710)은 신택스 요소 sub_view_id[i]을 파싱하고, 제어를 기능 블록(1712)으로 넘겨준다. 기능 블록(1712)은 신택스 요소 num_parts_minus1[sub_view_id[i]]를 파싱하고, 제어를 기능 블록(1714)으로 넘겨준다. 기능 블록(1714)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(1716)으로 넘겨준다. 판단 블록(1716)은 변수(j)가 신택스 요소 num_parts_minus1[sub_view_id[i]]보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1718)으로 넘겨진다. 그렇지 않다면, 제어는 판단 블록(1722)으로 넘겨진다.

[0137] 기능 블록(1718)은 이하의 신택스 요소들을 설정하고, 변수(j)를 증가시키고, 제어를 판단 블록(1716)으로 반환한다:

loc_left_offset[sub_view_id[i]][j]; loc_top_offset[sub_view_id[i]][j];
frame_crop_left_offset[sub_view_id[i]][j]; frame_crop_right_offset[sub_view_id[i]][j];
frame_crop_top_offset[sub_view_id[i]][j]; and
[0138] frame_crop_bottom_offset[sub_view_id[i][j].

[0139] 기능 블록(1720)은 다중-뷰 비디오 코딩(MVC)을 이용하여 현재 뷰에 대한 현재 화상을 디코딩하고, 제어를 기능 블록(1721)으로 넘겨준다. 기능 블록(1721)은 고 레벨 신택스를 이용하여 화상으로부터 각 뷰를 분리하고, 제어를 종료 블록(1799)에 넘겨준다.

[0140] 디코딩된 화상으로부터 각 뷰의 분리는 비트스트림에서 표시된 고 레벨 신택스를 이용하여 이루어진다. 이 고 레벨 신택스는 정확한 위치 및 화상에 존재하는 뷰들의 가능한 방향(및 가능한 대응하는 깊이)을 표시할 수 있다.

[0141] 판단 블록(1722)은 신택스 요소 tiling_mode가 0인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1724)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(1738)으로 넘겨진다.

[0142] 기능 블록(1724)은 신택스 요소 flip_dir[sub_view_id[i]] 및 신택스 요소 upsample_view_flag[sub_view_id[i]]를 파싱하고, 제어를 판단 블록(1726)으로 넘겨준다. 판단 블록(1726)은 신택스 요소 upsample_view_flag[sub_view_id[i]]가 0인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1728)으로 넘겨진다. 그렇지 않다면, 제어는 판단 블록(1730)으로 넘겨진다.

[0143] 기능 블록(1728)은 신택스 요소 upsample_filter[sub_view_id[i]]를 파싱하고, 제어를 판단 블록(1730)으로 넘겨진다. 판단 블록(1730)은 신택스 요소 upsample_filter[sub_view_id[i]]가 3인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1732)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(1736)으로 넘겨진다.

[0144] 기능 블록(1732)은 이하의 신택스 요소들 즉, vert_dim[sub_view_id[i]]; hor_dim[sub_view_id[i]]; 및 quantizer[sub_view_id[i]]를 파싱하고, 제어를 기능 블록(1734)으로 넘겨준다. 기능 블록(1734)은 각 YUV 컴포넌트에 대한 필터 계수들을 파싱하고, 제어를 기능 블록(1736)으로 넘겨준다.

[0145] 기능 블록(1736)은 변수(i)를 증가시키고, 제어를 판단 블록(1708)으로 반환한다.

[0146] 기능 블록(1738)은 신택스 요소 pixel_dist_x[sub_view_id[i]] 및 신택스 요소 flip_dist_y[sub_view_id[i]]를 파싱하고, 제어를 기능 블록(1740)을 넘겨준다. 기능 블록(1740)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(1742)으로 넘겨준다. 판단 블록(1742)은 변수(j)의 현재 값이 신택스 요소 num_parts[sub_view_id[i]]의 현재 값보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1744)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(1736)으로 넘겨진다.

[0147] 기능 블록(1744)은 신택스 요소 num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]를 파싱하고, 제어를 기능 블록(1746)으로 넘겨준다. 기능 블록(1746)은 모든 픽셀 타일링 필터들에 대한 계수들을 파싱하고, 제어를 기능 블록(1736)으로 넘겨준다.

[0148] 도 18을 참조하면, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용한 화상들의 인코딩에 대비하여 복수의 뷰들 및 깊이들에 대한 화상들을 처리하는 예시적인 방법이 개괄적으로 도면 번호(1800)에 의해 표시된다.

[0149] 방법(1800)은 제어를 기능 블록(1810)에 넘겨주는 시작 블록(1805)을 포함한다. 기능 블록(1810)은 타일 포멧의 수퍼-화상으로서 특정 시간 인스턴스에서 총 M 뷰들 및 깊이 맵들 중에 모든 N 뷰들 및 깊이 맵들을 배열하고, 제어를 기능 블록(1815)으로 넘겨준다. 기능 블록(1815)은 신택스 요소 num_coded_views_minus1을 설정하고, 제어를 기능 블록(1820)에 넘겨준다. 기능 블록(1820)은 view_id[i]에 대응하는 모든 (num_coded_views_minus1+1) 깊이들에 대해 신택스 요소 view_id[i]를 설정하고, 제어를 기능 블록(1825)에 넘겨준다. 기능 블록(1825)은 앵커 깊이 화상들에 대한 인터-뷰 참조 종속 정보를 설정하고, 제어를 기능 블록(1830)에 넘겨준다. 기능 블록(1830)은 비-앵커 깊이 화상들에 대한 인터-뷰 참조 종속 정보를 설정하고, 제어를 기능 블록(1835)에 넘겨준다. 기능 블록(1835)은 신택스 요소 pseudo_view_present_flag를 설정하고, 제어를 판단 블록(1840)으로 넘겨준다. 판단 블록(1840)은 신택스 요소 pseudo_view_present_flag의 현재값이 참인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1845)으로 넘겨진다. 그렇지 않다면, 제어는 종료 블록(1899)으로 넘겨진다.

[0150] 기능 블록(1845)은 이하의 신택스 요소들, 즉 tiling_mode; org_pic_width_in_mbs_minus1; 및 org_pic_height_in_mbs_minus1를 설정하고 제어를 기능 블록(1850)으로 넘겨준다. 기능 블록(1850)은 각 코딩된 뷰에 대해 pseudo_view_info(view_id)를 호출하고, 제어를 종료 블록(1899)으로 넘겨준다.

[0151] 도 19a 및 도 19b를 참조하면, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용하여 복수의 뷰들 및 깊이들에 대한 화상들을 인코딩하는 예시적인 방법이 개괄적으로 참조 번호(1900)에 의해 표시된다.

[0152] 방법(1900)은 제어를 기능 블록(1904)에 넘겨주는 시작 블록(1902)을 포함한다. 기능 블록(1904)은 신택스 요소 num_sub_views_minus1을 설정하고, 제어를 기능 블록(1906)에 넘겨준다. 기능 블록(1906)은 변수(i)를 0으로 설정하고, 제어를 판단 블록(1908)으로 넘겨준다. 판단 블록(1908)은 변수(i)가 sub_views의 수보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1910)으로 넘겨진다. 그렇지 않다면, 제어를 기능 블록(1920)으로 넘겨준다.

[0153] 기능 블록(1910)은 신택스 요소 sub_view_id[i]을 설정하고, 제어를 기능 블록(1912)으로 넘겨준다. 기능 블록(1912)은 신택스 요소 num_parts_minus1[sub_view_id[i]]를 설정하고, 제어를 기능 블록(1914)으로 넘겨준다. 기능 블록(1914)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(1916)으로 넘겨준다. 판단 블록(1916)은 변수(j)가 신택스 요소 num_parts_minus1[sub_view_id[i]]보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1918)으로 넘겨진다. 그렇지 않다면, 제어는 판단 블록(1922)으로 넘겨진다.

[0154] 기능 블록(1918)은 이하의 신택스 요소들을 설정하고, 변수(j)를 증가시키고, 제어를 판단 블록(1916)으로 반환한다:

[0155] loc_left_offset[sub_view_id[i]][j]; loc_top_offset[sub_view_id[i]][j];
frame_crop_left_offset[sub_view_id[i]][j]; frame_crop_right_offset[sub_view_id[i]][j];
frame_crop_top_offset[sub_view_id[i]][j]; and
frame_crop_bottom_offset[sub_view_id[i][j].

[0156] 기능 블록(1920)은 다중-뷰 비디오 코딩(MVC)을 이용하여 현재 뷰에 대한 현재 깊이를 인코딩하고, 제어를 종료

블록(1999)으로 넘겨준다. 깊이 신호는 그 대응하는 비디오 신호가 인코딩된 방식과 유사하게 인코딩될 수 있다. 예를 들어, 뷰에 대한 깊이 신호는 다른 깊이 신호들만을, 또는 비디오 신호들만을 또는 깊이 및 비디오 신호들 둘 다를 포함하는 타일상에 포함될 수 있다. 타일(의사-뷰)은 그 후 MVC에 대한 단일 뷰로서 취급되고, 또한, 아마도 MVC에 대한 다른 뷰들로서 취급되는 다른 타일들 또한 존재한다.

[0157] 판단 블록(1922)은 신택스 요소 tiling_mode가 0인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1924)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(1938)으로 넘겨진다.

[0158] 기능 블록(1924)은 신택스 요소 flip_dir[sub_view_id[i]] 및 신택스 요소 upsample_view_flag[sub_view_id[i]]를 설정하고, 제어를 판단 블록(1926)으로 넘겨준다. 판단 블록(1926)은 신택스 요소 upsample_view_flag[sub_view_id[i]]의 현재값이 1인지를 결정한다. 만약 그렇다면 제어는 기능 블록(1928)으로 넘겨진다. 그렇지 않다면 제어는 판단 블록(1930)으로 넘겨진다.

[0159] 기능 블록(1928)은 신택스 요소 upsample_filter[sub_view_id[i]]를 설정하고, 제어를 판단 블록(1930)으로 넘겨준다. 판단 블록(1930)은 신택스 요소 upsample_filter[sub_view_id[i]]의 값이 3인지를 결정한다. 만약 그렇다면 제어는 기능 블록(1932)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(1936)으로 넘겨진다.

[0160] 기능 블록(1932)은 이하의 신택스 요소들 즉, vert_dim[sub_view_id[i]]; hor_dim[sub_view_id[i]]; 및 quantizer[sub_view_id[i]]를 설정하고, 제어를 기능 블록(1934)으로 넘겨준다. 기능 블록(1934)은 각 YUV 컴포넌트에 대한 필터 계수들을 설정하고, 제어를 기능 블록(1936)으로 넘겨준다.

[0161] 기능 블록(1936)은 변수(i)를 증가시키고, 제어를 판단 블록(1908)으로 반환한다.

[0162] 기능 블록(1938)은 신택스 요소 pixel_dist_x[sub_view_id[i]] 및 신택스 요소 flip_dist_y[sub_view_id[i]]를 설정하고, 제어를 기능 블록(1940)으로 넘겨준다. 기능 블록(1940)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(1942)으로 넘겨준다. 판단 블록(1942)은 변수(j)의 현재 값이 신택스 요소 num_parts[sub_view_id[i]]의 현재 값보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(1944)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(1936)으로 넘겨진다.

[0163] 기능 블록(1944)은 신택스 요소 num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]를 설정하고, 제어를 기능 블록(1946)으로 넘겨준다. 기능 블록(1946)은 모든 픽셀 타일링 필터들에 대한 계수들을 설정하고, 제어를 기능 블록(1936)으로 넘겨준다.

[0164] 도 20을 참조하면, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용한 화상들의 디코딩에 대비하여 복수의 뷰들 및 깊이들에 대한 화상들을 처리하는 예시적인 방법이 개괄적으로 참조 번호(2000)에 의해 표시된다.

[0165] 방법(2000)은 제어를 기능 블록(2015)에 넘겨주는 시작 블록(2005)을 포함한다. 기능 블록(2015)은 신택스 요소 num_coded_views_minus1을 파싱하고, 제어를 기능 블록(2020)에 넘겨준다. 기능 블록(2020)은 view_id[i]에 대응하는 모든(num_coded_views_minus1+1) 깊이들에 대해 신택스 요소 view_id[i]를 파싱하고, 제어를 기능 블록(2025)에 넘겨준다. 기능 블록(2025)은 앵커 깊이 화상들에 대한 인터-뷰 참조 종속 정보를 파싱하고, 제어를 기능 블록(2030)에 넘겨준다. 기능 블록(2030)은 비-앵커 깊이 화상들에 대한 인터-뷰 참조 종속 정보를 파싱하고, 제어를 기능 블록(2035)에 넘겨준다. 기능 블록(2035)은 신택스 요소 pseudo_view_present_flag를 파싱하고, 제어를 판단 블록(2040)으로 넘겨준다. 판단 블록(2040)은 신택스 요소 pseudo_view_present_flag의 현재 값이 참인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(2045)으로 넘겨진다. 그렇지 않다면, 제어는 종료 블록(2099)으로 넘겨진다.

[0166] 기능 블록(2045)은 이하의 신택스 요소들, 즉 tiling_mode; org_pic_width_in_mbs_minus1; 및 org_pic_height_in_mbs_minus1를 파싱하고, 제어를 기능 블록(2050)으로 넘겨준다. 기능 블록(2050)은 각 코딩된 뷰에 대해 pseudo_view_info(view_id)를 호출하고, 제어를 종료 블록(2099)으로 넘겨준다.

[0167] 도 21a 및 도 21b를 참조하면, MPEG-4 AVC 표준의 다중 뷰 비디오 코딩(MVC) 확장을 이용하여 복수의 뷰들 및 깊이들에 대한 화상들을 디코딩하는 예시적인 방법이 개괄적으로 참조 번호(2100)에 의해 표시된다.

[0168] 방법(2100)은 입력 파라미터 pseudo_view_id로 시작하며 제어를 기능 블록(2104)에 넘겨주는 시작 블록(2102)을 포함한다. 기능 블록(2104)은 신택스 요소 num_sub_views_minus1을 파싱하고, 제어를 기능 블록(2106)에 넘겨준다. 기능 블록(2106)은 변수(i)를 0으로 설정하고, 제어를 판단 블록(2108)으로 넘겨준다. 판단 블록(2108)은 변수(i)가 sub_views의 수보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(2110)으로

넘겨진다. 그렇지 않다면, 제어를 기능 블록(2120)으로 넘겨준다.

[0169] 기능 블록(2110)은 신택스 요소 sub_view_id[i]을 파싱하고, 제어를 기능 블록(2112)으로 넘겨준다. 기능 블록(2112)은 신택스 요소 num_parts_minus1[sub_view_id[i]]를 파싱하고, 제어를 기능 블록(2114)으로 넘겨준다. 기능 블록(2114)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(2116)으로 넘겨준다. 판단 블록(2116)은 변수(j)가 신택스 요소 num_parts_minus1[sub_view_id[i]]보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(2118)으로 넘겨진다. 그렇지 않다면, 제어는 판단 블록(2122)으로 넘겨진다.

[0170] 기능 블록(2118)은 이하의 신택스 요소들을 설정하고, 변수(j)를 증가시키고, 제어를 판단 블록(2116)으로 반환한다:

[0171] loc_left_offset[sub_view_id[i]][j]; loc_top_offset[sub_view_id[i]][j];
frame_crop_left_offset[sub_view_id[i]][j]; frame_crop_right_offset[sub_view_id[i]][j];
frame_crop_top_offset[sub_view_id[i]][j]; and
frame_crop_bottom_offset[sub_view_id[i][j].

[0172] 기능 블록(2120)은 다중-뷰 비디오 코딩(MVC)을 이용하여 현재 화상을 디코딩하고, 제어를 기능 블록(2121)으로 넘겨준다. 기능 블록(2121)은 고 레벨 신택스를 이용하여 화상으로부터 각 뷰를 분리하고, 제어를 종료 블록(2199)에 넘겨준다. 고 레벨 신택스를 이용한 화상으로부터 각 뷰의 분리는 앞서 기술한 바와 같다.

[0173] 판단 블록(2122)은 신택스 요소 tiling_mode가 0인지를 결정한다. 만약 그렇다면, 제어는 기능 블록(2124)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(2138)으로 넘겨진다.

[0174] 기능 블록(2124)은 신택스 요소 flip_dir[sub_view_id[i]] 및 신택스 요소 upsample_view_flag[sub_view_id[i]]를 파싱하고, 제어를 판단 블록(2126)으로 넘겨준다. 판단 블록(2126)은 신택스 요소 upsample_view_flag[sub_view_id[i]]의 현재값이 1인지를 결정한다. 만약 그렇다면 제어는 기능 블록(2128)으로 넘겨진다. 그렇지 않다면 제어는 판단 블록(2130)으로 넘겨진다.

[0175] 기능 블록(2128)은 신택스 요소 upsample_filter[sub_view_id[i]]를 파싱하고, 제어를 판단 블록(2130)으로 넘겨준다. 판단 블록(2130)은 신택스 요소 upsample_filter[sub_view_id[i]]의 값이 3인지를 결정한다. 만약 그렇다면 제어는 기능 블록(2132)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(2136)으로 넘겨진다.

[0176] 기능 블록(2132)은 이하의 신택스 요소들 즉, vert_dim[sub_view_id[i]]; hor_dim[sub_view_id[i]]; 및 quantizer[sub_view_id[i]]를 파싱하고, 제어를 기능 블록(2134)으로 넘겨준다. 기능 블록(2134)은 각 YUV 컴포넌트에 대한 필터 계수들을 파싱하고, 제어를 기능 블록(2136)으로 넘겨준다.

[0177] 기능 블록(2136)은 변수(i)를 증가시키고, 제어를 판단 블록(2108)으로 반환한다.

[0178] 기능 블록(2138)은 신택스 요소 pixel_dist_x[sub_view_id[i]] 및 신택스 요소 flip_dist_y[sub_view_id[i]]를 파싱하고, 제어를 기능 블록(2140)으로 넘겨준다. 기능 블록(2140)은 변수(j)를 0으로 설정하고, 제어를 판단 블록(2142)으로 넘겨준다. 판단 블록(2142)은 변수(j)의 현재 값이 신택스 요소 num_parts[sub_view_id[i]]의 현재 값보다 작은지를 결정한다. 만약 그렇다면, 제어는 기능 블록(2144)으로 넘겨진다. 그렇지 않다면, 제어는 기능 블록(2136)으로 넘겨진다.

[0179] 기능 블록(2144)은 신택스 요소 num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]를 파싱하고, 제어를 기능 블록(2146)으로 넘겨준다. 기능 블록(2146)은 모든 픽셀 타일링 필터들에 대한 계수들을 파싱하고, 제어를 기능 블록(2136)으로 넘겨준다.

[0180] 도 22를 참조하면, 픽셀 레벨에서의 타일링 예들이 개괄적으로 참조 번호(2200)에 의해 표시된다. 도 22는 아래에서 추가로 설명된다.

[0181] **MPEG-4 AVC 또는 MVC를 이용한 뷰 타일링**

[0182] 다중-뷰 비디오 코딩의 응용은 FTV(free view point TV)이다. 이 응용은 사용자가 2개 이상의 뷰들 사이를 자유롭게 이동할 수 있을 것을 필요로 한다. 이를 달성하기 위해, 2개의 뷰들 사이의 "가상(virtual)"의 뷰들이 보간(interpolation) 또는 합성될 필요가 있다. 뷰 보간을 수행하기 위한 몇몇 방법들이 존재한다. 이 방법들 중 하나는 뷰 보간/합성을 위해 깊이를 이용한다.

[0183] 각 뷰는 연관된 깊이 신호를 가질 수 있다. 따라서, 깊이는 비디오 신호의 또 다른 유형로 고려될 수 있다. 도 9는 깊이 신호(900)의 예를 도시한다. FTV와 같은 응용들을 가능하게 하기 위해, 깊이 신호는 비디오 신호

와 함께 전송된다. 제안된 타일링 프레임워크에서, 깊이 신호는 타일들의 하나로서 또한 부가될 수 있다. 도 10은 타일로서 부가된 깊이 신호의 예를 도시한다. 깊이 신호들/타일들은 도 10의 우측상에 도시된다.

[0184] 깊이가 전체 프레임의 타일로서 인코딩되면, 고 레벨 신택스는 렌더기가 깊이 신호를 적절히 사용할 수 있도록 어느 타일이 깊이 신호인지를 표시하여야 한다.

[0185] 입력 시퀀스(예를 들어, 도 1에 도시됨)가 MPEC-4 AVC 표준 인코더(또는 다른 비디오 코딩 표준 및/또는 권고안에 대응하는 인코더)를 이용하여 인코딩된 경우, 제안된 고 레벨 신택스는 예를 들어, SPS(Sequence Parameter Set), PPS(Picture Parameter Set), 슬라이스 헤더, 및 또는 SEI(Supplemental Enhancement Information) 메시지에 존재할 수 있다. 제안된 방법의 실시예는 신택스가 SEI 메시지에 존재하는 표 1에서 도시된다.

[0186] 의사 뷰들의 입력 시퀀스(예를 들어, 도 1에 도시됨)가 MPEC-4 AVC 표준 인코더(또는 다른 비디오 코딩 표준 및/또는 권고안에 대해 다중-뷰 비디오 코딩 표준에 대응하는 인코더)의 다중-뷰 비디오 코딩(MVC) 확장을 이용하여 인코딩된 경우에서, 제안된 고 레벨 신택스는 SPS, PPS, 슬라이스 헤더, SEI 메시지 또는 특정 프로파일에 존재할 수 있다. 제안된 방법의 실시예가 표 1에 도시된다. 표 1은 본 원리들의 실시예에 따라 제안된 신택스 요소들을 포함하는 SPS 구조에 존재하는 신택스 요소들을 도시한다.

[0187]

seq_parameter_set_mvc_extension( ) {	C	Descriptor
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
pseudo_view_present_flag		u(1)
if (pseudo_view_present_flag) {		
tiling_mode		
org_pic_width_in_mbs_minus1		
org_pic_height_in_mbs_minus1		
for(i = 0; i < num_views_minus_1; i++)		
pseudo_view_info(i);		
}		
}		

[0188] 표 1

[0189] 표 2는 본 원리들의 실시예에 따라, 표 1의 pseudo_view_info 신택스 요소에 대한 신택스 요소들을 도시한다.

[0190]

pseudo_view_info (pseudo_view_id) {	C	Descriptor
num_sub_views_minus_1[pseudo_view_id]	5	ue(v)
if (num_sub_views_minus_1 != 0) {		
for (i = 0; i < num_sub_views_minus_1[pseudo_view_id]; i++) {		
sub_view_id[i]	5	ue(v)
num_parts_minus1[sub_view_id[i]]	5	ue(v)
for(j = 0; j <= num_parts_minus1[sub_view_id[i]]; j++) {		
loc_left_offset[sub_view_id[i]] [j]	5	ue(v)
loc_top_offset[sub_view_id[i]] [j]	5	ue(v)
frame_crop_left_offset[sub_view_id[i]] [j]	5	ue(v)
frame_crop_right_offset[sub_view_id[i]] [j]	5	ue(v)
frame_crop_top_offset[sub_view_id[i]] [j]	5	ue(v)
frame_crop_bottom_offset[sub_view_id[i]] [j]	5	ue(v)
}		
if (tiling_mode == 0) {		
flip_dir[sub_view_id[i][j]	5	u(2)
upsample_view_flag[sub_view_id[i]]	5	u(1)
if(upsample_view_flag[sub_view_id[i]])		
upsample_filter[sub_view_id[i]]	5	u(2)
if(upsample_filter[sub_view_id[i]] == 3) {		
vert_dim[sub_view_id[i]]	5	ue(v)
hor_dim[sub_view_id[i]]	5	ue(v)
quantizer[sub_view_id[i]]	5	ue(v)
for (yuv= 0; yuv< 3; yuv++) {		
for (y = 0; y < vert_dim[sub_view_id[i]] - 1; y ++) {		
for (x = 0; x < hor_dim[sub_view_id[i]] - 1; x ++)		
filter_coeffs[sub_view_id[i]] [yuv][y][x]	5	se(v)
}		
}		
}		
} // if(tiling_mode == 0)		
else if (tiling_mode == 1) {		
pixel_dist_x[sub_view_id[i]]		
pixel_dist_y[sub_view_id[i]]		
for(j = 0; j <= num_parts[sub_view_id[i]]; j++) {		
num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]][j]		
for (coeff_idx = 0; coeff_idx <= num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]][j]; j++)		
pixel_tiling_filter_coeffs[sub_view_id[i]][j]		
} // for (j = 0; j <= num_parts[sub_view_id[i]]; j++)		
} // else if (tiling_mode == 1)		
} // for (i = 0; i < num_sub_views_minus_1; i++)		
} // if (num_sub_views_minus_1 != 0)		
}		

[0191] 표 2

[0192] **표 1 및 표 2에 제공된 신택스 요소들의 의미론(semantics)**

[0193] pseudo_view_present_flag가 참이면 일부 뷰가 다수의 서브-뷰들의 수퍼 뷰임을 나타낸다..

[0194] tiling_mode가 0이면 서브-뷰들이 화상 레벨에서 타일링되었음을 나타낸다. 값이 1이면 타일링이 픽셀 레벨에서 이루어졌음을 나타낸다.

[0195] 새로운 SEI 메시지는 MPEC-4 AVC 표준 또는 MPEC-4 AVC 표준의 확장에서 사용되지 않은 SEI 페이로드 유형에 대한 값을 사용할 수 있다. 새로운 SEI 메시지는 이하의 의미론을 갖는 몇몇 신택스 요소들을 포함한다.

[0196] num_coded_views_minus1 + 1 은 비트스트림에 의해 지원되는 코딩된 뷰들의 수를 나타낸다. num_coded_views_minus1의 값은 0 내지 1023의 범위에 있다.

[0197] org_pic_width_in_mbs_minus1 + 1은 매크로 블록들의 유닛들로 각 뷰에서 화상의 폭을 지정한다.

[0198] 매크로 블록들의 유닛들로 화상 폭에 대한 변수는 다음과 같이 유도된다:

[0199] PicWidthInMbs = org_pic_width_in_mbs_minus1 + 1

[0200] 루마(luma) 컴포넌트에 대한 화상 폭의 변수는 다음과 같이 유도된다:

[0201] PicWidthInSamplesL = PicWidthInMbs * 16

[0202] 크로마(chroma) 컴포넌트들에 대한 화상 폭의 변수는 다음과 같이 유도된다:

[0203] PicWidthInSamplesC = PicWidthInMbs * MbWidthC

[0204] org_pic_height_in_mbs_minus1 + 1 은 매클로블록들의 유닛들로 각 뷰에서 화상의 높이를 지정한다.

[0205] 매크로 블록들의 유닛들로 화상 높이에 대한 변수는 다음과 같이 유도된다:

[0206] PicHeightInMbs = org_pic_height_in_mbs_minus1 + 1

[0207] 루마(luma) 컴포넌트에 대한 화상 높이의 변수는 다음과 같이 유도된다:

[0208] PicHeightInSamplesL = PicHeightInMbs * 16

[0209] 크로마(chroma) 컴포넌트들에 대한 화상 높이의 변수는 다음과 같이 유도된다:

[0210] PicHeightInSamplesC = PicHeightInMbs * MbHeightC

[0211] num_sub_views_minus1 + 1은 현재 뷰에 포함된 코딩된 서브-뷰들의 수를 나타낸다. num_coded_views_minus1의 값은 0 내지 1023의 범위에 있다.

[0212] sub_view_id[i]는 i로 표시된 디코딩 순서로 서브-뷰의 sub_view_id를 지정한다.

[0213] num_parts[sub_view_id[i]]는 sub_view_id[i]의 화상이 분할되는 부분들의 수를 지정한다.

[0214] loc_left_offset[sub_view_id[i]][j] 및 loc_top_offset[sub_view_id[i]][j]는 왼쪽 및 상부 픽셀들의 오프셋의 위치를 각각 지정하며, 여기서 현재 부분(j)은 sub_view_id[i]에 동일한 sub_view_id를 갖는 뷰의 마지막 재구성 화상에 위치된다.

[0215] view_id[i]는 i로 표시되는 코딩 순서를 갖는 뷰의 view_id이다.

[0216] frame_crop_left_offset[view_id[i]][j], frame_crop_right_offset[view_id[i]][j], frame_crop_top_offset[view_id[i]][j], 및 frame_crop_bottom_offset[view_id[i]][j]는 출력용 프레임 좌표들에서 지정된 사각형 영역으로 num_part(j) 및 view_id(i)의 부분인 코딩된 비디오 시퀀스에서의 화상들의 샘플들을 지정한다.

[0217] 변수들 CropUnitX 및 CropUnitY은 다음과 같이 유도된다:

[0218] - 만약 chroma_format_idc가 0이면, CropUnitX 및 CropUnitY는 다음과 같이 유도된다:

[0219] CropUnitX = 1

[0220] CropUnitY = 2 - frame_mbs_only_flag

[0221] - 그렇지 않다면(chroma_format_idc가 1,2 또는 3), CropUnitX 및 CropUnitY는 다음과 같이 유도된다:

[0222] CropUnitX = SubWidthC

[0223] CropUnitY = SubHeightC * (2 - frame_mbs_only_flag)

[0224] 프레임 크로핑 직사각형은 CropUnitX * frame_crop_left_offset 내지 PicWidthInSamplesL - (CropUnitX * frame_crop_right_offset + 1)으로부터 수평 프레임 좌표들 및 CropUnitY * frame_crop_top_offset 내지 (16 * FrameHeightInMbs) - (CropUnitY * frame_crop_bottom_offset + 1)로부터의 수직 프레임 좌표들을 갖는 루마 샘플들을 포함한다. frame_crop_left_offset의 값은 0 내지 (PicWidthInSamplesL / CropUnitX) - (frame_crop_right_offset + 1)의 범위에 있고, frame_crop_top_offset의 값은 0 내지 (16 * FrameHeightInMbs / CropUnitY) - (frame_crop_bottom_offset + 1)의 범위에 있다.

[0225] chroma_format_idc가 0이 아니면, 2개의 크로마 어레이들의 대응하는 지정된 샘플들은 프레임 좌표들(x / SubWidthC, y / SubHeightC)을 갖는 샘플이며, 여기서 (x, y)는 지정된 루마 샘플들의 프레임 좌표들이다.

[0226] 디코딩된 필드들에 대해, 디코딩된 필드의 지정된 샘플들은 프레임 좌표들에 지정된 직사각형내에 있는 샘플들이다.

[0227] num_parts[view_id[i]]는 view_id[i]의 화상이 분할되는 부분들의 수를 지정한다.

[0228] depth_flag[view_id[i]]는 현재 부분이 깊이 신호인지를 지정한다. depth_flag가 0이면, 현재 부분은 깊이 신호가 아니다. depth_flag가 1이면 현재 부분은 view_id[i]에 의해 식별된 뷰와 연관된 깊이 신호이다.

[0229] flip_dir[sub_view_id[i]][j]는 현재 부분에 대한 플립핑 방향(flipping direction)을 지정한다. flip_dir이 0이면 플립핑이 없음을 나타내고, flip_dir이 1이면 수평 방향의 플립핑을 나타내고, flip_dir이 2이면 수직 방향의 플립핑을 나타내고, flip_dir이 3이면 수평 및 수직 방향의 플립핑을 나타낸다.

[0230] flip_dir[view_id[i]][j]는 현재 부분에 대한 플립핑 방향(flipping direction)을 지정한다. flip_dir이 0이면 플립핑이 없음을 나타내고, flip_dir이 1이면 수평 방향의 플립핑을 나타내고, flip_dir이 2이면 수직 방향의 플립핑을 나타내고, flip_dir이 3이면 수평 및 수직 방향의 플립핑을 나타낸다.

[0231] loc_left_offset[view_id[i]][j], loc_top_offset[view_id[i]][j]는 픽셀들 오프셋들의 위치를 지정하며, 여기서 현재 부분(j)은 view_id[i]와 같은 view_id를 갖는 뷰의 마지막 재구성 화상에 위치된다.

[0232] upsample_view_flag[view_id[i]]는 view_id[i]에 의해 지정된 뷰에 속하는 화상이 업샘플링될 필요가 있는지를 나타낸다. upsample_view_flag[view_id[i]]가 0이면 view_id[i]와 같은 view_id를 갖는 화상이 업샘플링되지 않을 것임을 지정한다. upsample_view_flag[view_id[i]]가 1이면 view_id[i]와 같은 view_id를 갖는 화상이 업샘플링될 것임을 지정한다.

[0233] upsample_filter[view_id[i]]는 업샘플링에 사용되는 필터의 유형을 나자트낸다. upsample_filter[view_id[i]]이 0이면 6-탭 AVC 필터가 사용되어야 함을 나타내고, upsample_filter[view_id[i]]이 1이면 4-탭 SVC 필터가 사용되어야 함을 나타내고, upsample_filter[view_id[i]]이 2이면 이중선형 필터가 사용되어야 함을 나타내고, upsample_filter[view_id[i]]이 3이면 주문형(custom) 필터 계수들이 전송됨을 나타낸다. upsample_filter[view_id[i]]가 존재하지 않으면, 0으로 설정된다. 이 실시예에서, 우리는 2D 주문형 필터를 사용한다. 이는 1D 필터들, 임의의 다른 비선형 필터로 쉽게 확장될 수 있다.

[0234] vert_dim[view_id[i]]는 주문형 2D 필터의 수직 치수를 지정한다.

[0235] hor_dim[view_id[i]]는 주문형 2D 필터의 수평 치수를 지정한다.

[0236] quantizer[view_id[i]]는 각 필터 계수에 대한 양자와 인수(factor)를 지정한다.

[0237] filter_coeffs[view_id[i]][yuv][y][x]는 양자화된 필터 계수들을 지정한다. yuv는 필터 계수들이 적용될 컴포넌트를 시그널링한다. yuv가 0이면 Y 컴포넌트가 지정되고, yuv가 1이면 U 컴포넌트가 지정되고, yuv가 2이면 V 컴포넌트가 지정된다.

[0238] pixel_dist_x[sub_view_id[i]] 및 pixel_dist_y[sub_view_id[i]]는 각각 sub_view_id[i]와 같은 sub_view_id

를 갖는 뷰의 인접 픽셀들사이에서 마지막 재구성 의사 뷰의 수평 방향 및 수직 방향간의 거리를 지정한다.

[0239] num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i][j]] + 1은 타일링 보드가 1로 설정된 경우 필터 계수들의 수를 나타낸다.

[0240] pixel_tiling_filter_coeffs[sub_view_id[i][j]]는 타일링된 화상을 필터링하는데 사용될 수 있는 필터를 나타내는데 필요로 되는 필터 계수들을 시그널링한다.

[0241] **픽셀 레벨에서의 타일링 예들**

[0242] 도 22를 참조하면, 4개의 뷰들로부터 픽셀들을 타일링함으로써 의사 뷰의 합성을 도시하는 2개의 예들이 개괄적으로 참조 번호들(2210 및 2220)에 의해 각각 표시된다. 4개의 뷰들은 집합적으로 참조 번호(2250)에 의해 표시된다. 도 22의 제 1 예에 대한 신택스 값들은 이하의 표 3에 제공된다.

[0243]

pseudo_view_info (pseudo_view_id) {	Value
num_sub_views_minus_1[pseudo_view_id]	3
sub_view_id[0]	0
num_parts_minus1[0]	0
loc_left_offset[0][0]	0
loc_top_offset[0][0]	0
pixel_dist_x[0][0]	0
pixel_dist_y[0][0]	0
sub_view_id[1]	0
num_parts_minus1[1]	0
loc_left_offset[1][0]	1
loc_top_offset[1][0]	0
pixel_dist_x[1][0]	0
pixel_dist_y[1][0]	0
sub_view_id[2]	0
num_parts_minus1[2]	0
loc_left_offset[2][0]	0
loc_top_offset[2][0]	1
pixel_dist_x[2][0]	0
pixel_dist_y[2][0]	0
sub_view_id[3]	0
num_parts_minus1[3]	0
loc_left_offset[3][0]	1
loc_top_offset[3][0]	1
pixel_dist_x[3][0]	0
pixel_dist_y[3][0]	0

[0244] 표 3

[0245] 도 22의 제 2 예에 대한 신택스 값들은 2개의 신택스 요소들 즉, loc_left_offset[3][0]=5 및 loc_top_offset[3][0]=3을 제외하면 모두 동일하다.

[0246] 오프셋은 뷰에 대응하는 픽셀들이 특정한 오프셋 위치에서 시작해야 한다는 것을 나타낸다. 이는 도 22(2220)에서 도시된다. 이는 예를 들어, 공통 객체가 하나의 뷰로부터 다른것으로 이동되게 보이는 영상들을 2개의 뷰들이 생성할 때 행해질 수 있다. 예를 들어, 제 1 및 제 2 카메라들(제 1 및 제 2 뷰들을 나타냄)이 객체의 화상을 촬영하면, 객체는 제 1 뷰에 비해 제 2 뷰에서 오른쪽으로 5픽셀만큼 이동되게 보일 수 있다. 이는 제 1 뷰의 픽셀(i-5,j)이 제 2 뷰의 픽셀(i,j)에 대응한다는 것을 의미한다. 2개의 뷰들의 픽셀들이 단순히 픽셀단위로 타일링되는 경우, 타일의 인접 픽셀들간의 많은 상관이 존재하지 않을 수 있고, 공간 코딩 이득들이 작을 수 있다. 역으로, 하나의 뷰로부터의 픽셀(i-5,j)이 두 번째 뷰로부터의 픽셀(i,j) 옆에 위치하도록 타일링을 이동시킴으로써, 공간 상관이 증가할 수 있고, 공간 코딩 이득 또한 증가할 수 있다. 그 이유는 예를 들어, 제 1 및 제 2 뷰들의 객체에 대한 대응하는 픽셀들이 서로 옆에 타일링되기 때문이다.

[0247] 따라서, loc_left_offset 및 loc_top_offset의 존재는 코딩 효율에 유리할 수 있다. 오프셋 정보는 외부 수단에 의해 얻어질 수 있다. 예를 들어, 카메라들의 위치 정보 또는 뷰들간의 전역 디스패리티 벡터들은 이러한 오프셋 정보를 결정하는데 이용될 수 있다.

[0248] 오프셋의 결과로서, 의사 뷰의 일부 화소들은 임의의 뷰로부터 픽셀 값들이 할당되지 않는다. 상기 예를 지속해서, i=0...4의 값들에 대해, 하나의 뷰로부터의 타일링 픽셀(i-5,j)이 두 번째 뷰로부터의 픽셀(i,j)과 나란히 있으면, 타일링할 하나의 뷰로부터 픽셀(i-5,j)이 존재하지 않고, 따라서, 이러한 픽셀들은 타일에서 엠티(empty)이다. 임의의 뷰로부터 픽셀 값들이 할당되지 않은 의사-뷰(타일)의 이러한 픽셀들에 대해서, 적어도 하나의 구현은 AVC의 모션 보상의 서브-픽셀 보간 절차와 유사한 보간 절차를 이용한다. 즉, 엠티 타일 픽셀은 인접 픽셀들로부터 보간될 수 있다. 이러한 보간은 타일에서 보다 큰 공간 상관 및 타일에 대한 보다 큰 코딩 이득을 발생시킬 수 있다.

[0249] 비디오 코딩에서, 우리는 I, P, 및 B 화상들과 같은 각 화상에 대해 상이한 코딩 유형을 선택한다. 다중-뷰 비디오 코딩에서, 또한, 우리는 앵커 및 비-앵커 화상들을 규정한다. 일 실시예에서, 우리는 그룹핑(grouping)의

결정이 화상 유형에 기초하여 이루어진다는 것을 제안한다. 이 그룹핑 정보는 고 레벨 신택스에서 시그널링된다.

[0250] 도 11을 참조하면, 단일 프레임상에 타일링된 5 뷰들의 예가 개괄적으로 참조 번호(1100)에 의해 표시된다. 특히, 볼룸(ballroom) 시퀀스는 단일 프레임상에 타일링된 5 뷰들로 도시된다. 부가적으로, 제 5 뷰가 2개의 부분들로 분할되어, 4각형 프레임상에 배열될 수 있다는 것을 알 수 있다. 여기서, 각 뷰는 크기가 QVGA이고, 따라서 총 프레임 치수는 640*600이다. 600이 16의 배수가 아니기 때문에, 608로 확장되어야 한다.

[0251] 이 예에 있어서, 가능한 SEI 메시지는 표 4에 도시된 바와 같이 될 수 있다.

[0252]

multiview_display_info(payloadSize) {	Value
num_coded_views_minus1	5
org_pic_width_in_mbs_minus1	40
org_pic_height_in_mbs_minus1	30
view_id[0]	0
num_parts[view_id[0]]	1
depth_flag[view_id[0]][0]	0
flip_dir[view_id[0]][0]	0
loc_left_offset[view_id[0]] [0]	0
loc_top_offset[view_id[0]] [0]	0
frame_crop_left_offset[view_id[0]] [0]	0
frame_crop_right_offset[view_id[0]] [0]	320
frame_crop_top_offset[view_id[0]] [0]	0
frame_crop_bottom_offset[view_id[0]] [0]	240
upsample_view_flag[view_id[0]]	1
if(upsample_view_flag[view_id[0]]) {	
vert_dim[view_id[0]]	6
hor_dim[view_id[0]]	6
quantizer[view_id[0]]	32
for (yuv= 0; yuv< 3; yuv++) {	
for (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {	
for (x = 0; x < hor_dim[view_id[i]] - 1; x ++)	
filter_coeffs[view_id[i]] [yuv][y][x]	XX
view_id[1]	1
num_parts[view_id[1]]	1
depth_flag[view_id[0]][0]	0
flip_dir[view_id[1]][0]	0
loc_left_offset[view_id[1]] [0]	0
loc_top_offset[view_id[1]] [0]	0
frame_crop_left_offset[view_id[1]] [0]	320
frame_crop_right_offset[view_id[1]] [0]	640
frame_crop_top_offset[view_id[1]] [0]	0
frame_crop_bottom_offset[view_id[1]] [0]	320

[0253]

upsample_view_flag[view_id[1]]	1
if(upsample_view_flag[view_id[1]]) {	
vert_dim[view_id[1]]	6
hor_dim[view_id[1]]	6
quantizer[view_id[1]]	32
for (yuv= 0; yuv< 3; yuv++) {	
for (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {	
for (x = 0; x < hor_dim[view_id[i]] - 1; x ++)	
filter_coeffs[view_id[i]] [yuv][y][x]	XX
.....(similarly for view 2,3)	
view_id[4]	4
num_parts[view_id[4]]	2
depth_flag[view_id[0]][0]	0
flip_dir[view_id[4]][0]	0
loc_left_offset[view_id[4]] [0]	0
loc_top_offset[view_id[4]] [0]	0
frame_crop_left_offset[view_id[4]] [0]	0
frame_crop_right_offset[view_id[4]] [0]	320
frame_crop_top_offset[view_id[4]] [0]	480
frame_crop_bottom_offset[view_id[4]] [0]	600
flip_dir[view_id[4]][1]	0
loc_left_offset[view_id[4]] [1]	0
loc_top_offset[view_id[4]] [1]	120
frame_crop_left_offset[view_id[4]] [1]	320
frame_crop_right_offset[view_id[4]] [1]	640
frame_crop_top_offset[view_id[4]] [1]	480
frame_crop_bottom_offset[view_id[4]] [1]	600
upsample_view_flag[view_id[4]]	1
if(upsample_view_flag[view_id[4]]) {	
vert_dim[view_id[4]]	6
hor_dim[view_id[4]]	6
quantizer[view_id[4]]	32
for (yuv= 0; yuv< 3; yuv++) {	
for (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {	
for (x = 0; x < hor_dim[view_id[i]] - 1; x ++)	
filter_coeffs[view_id[i]] [yuv][y][x]	XX

[0254] 표 4

[0255] 표 5는 표 4에서 도시된 예에 대해 다중-뷰 정보를 전송하는 일반적인 신택스 구조를 도시한다.

[0256]

multiview_display_info(payloadSize) {	C	Descriptor
num_coded_views_minus1	5	ue(v)
org_pic_width_in_mbs_minus1	5	ue(v)
org_pic_height_in_mbs_minus1	5	ue(v)
for(i = 0; i <= num_coded_views_minus1; i++) {		
view_id[i]	5	ue(v)
num_parts[view_id[i]]	5	ue(v)
for(j = 0; j <= num_parts[i]; j++) {		
depth_flag[view_id[i]][j]		
flip_dir[view_id[i]][j]	5	u(2)
loc_left_offset[view_id[i]][j]	5	ue(v)
loc_top_offset[view_id[i]][j]	5	ue(v)
frame_crop_left_offset[view_id[i]][j]	5	ue(v)
frame_crop_right_offset[view_id[i]][j]	5	ue(v)
frame_crop_top_offset[view_id[i]][j]	5	ue(v)
frame_crop_bottom_offset[view_id[i]][j]	5	ue(v)
}		
upsample_view_flag[view_id[i]]	5	u(1)
if(upsample_view_flag[view_id[i]])		
upsample_filter[view_id[i]]	5	u(2)
if(upsample_filter[view_id[i]] == 3) {		
vert_dim[view_id[i]]	5	ue(v)
hor_dim[view_id[i]]	5	ue(v)
quantizer[view_id[i]]	5	ue(v)
for (yuv= 0; yuv< 3; yuv++) {		
for (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {		
for (x = 0; x < hor_dim[view_id[i]] - 1; x ++)		
filter_coeffs[view_id[i]] [yuv][y][x]	5	se(v)
}		
}		
}		
}		
}		

[0257] 표 5

[0258] 도 23을 참조하면, 비디오 처리 디바이스(2300)가 도시된다. 비디오 처리 디바이스(2300)는 예를 들어, 인코딩된 비디오를 수신하고, 예를 들어, 저장을 위해 또는 디스플레이를 위해 디코딩된 비디오를 사용자에게 제공하는 셋톱 박스 또는 다른 디바이스일 수 있다. 따라서, 디바이스(2300)는 그 출력을 텔레비전, 컴퓨터 모니터 또는 컴퓨터 또는 다른 처리 디바이스에 제공할 수 있다.

[0259] 디바이스(2300)는 데이터 신호(2320)를 수신하는 디코더(2310)를 포함한다. 데이터 신호(2320)는 예를 들어, AVC 또는 MVC 호환 스트림을 포함할 수 있다. 디코더(2310)는 수신된 신호(2320)의 일부 또는 모두를 디코딩하고, 출력으로서 디코딩된 비디오 신호(2330) 및 타일링 정보(2340)를 제공한다. 디코딩된 비디오(2330) 및 타일링 정보(2340)는 선택기(2350)에 제공된다. 디바이스(2300)는 상용자 입력(2370)을 수신하는 사용자 인터페이스(2360)를 또한 포함한다. 사용자 인터페이스(2360)는 사용자 입력(2370)에 기초하여 화상 선택 신호(2380)를 선택기(2350)에 제공한다. 화상 선택 신호(2380) 및 사용자 입력(2370)은 다중 화상들 중 어느 것이 사용자가 디스플레이하기를 원하는지 것인지를 나타낸다. 선택기(2350)는 출력(2390)으로서 선택된 화상(들)을 제공한다. 선택기(2350)는 출력(2390)으로서 제공하기 위해 디코딩된 비디오(2330)의 화상들 중 어느것을 선택하도록 화상 선택 정보(2380)를 사용한다. 선택기(2350)는 디코딩된 비디오(2330)의 선택된 화상(들)을 발견하기 위해 타일링 정보(2340)를 이용한다.

[0260] 다양한 구현들에서, 선택기(2350)는 사용자 인터페이스(2360)를 포함하고, 다른 구현들에서, 어떠한 사용자 인터페이스(2360)도 필요 없는데, 그 이유는 선택기(2350)는 별도의 인터페이스 기능이 수행되지 않고 직접 사용자 입력(2370)을 수신하기 때문이다. 선택기(2350)는 소프트웨어로 또는 예를 들어, 집적 회로로서 구현될 수 있다. 선택기(2350)는 디코더(2310)를 또한 포함할 수 있다.

[0261] 보다 일반적으로, 이 응용에서 기술된 다양한 구현들의 디코더들은 전체 타일을 포함하는 디코딩된 출력을 제공할 수 있다. 부가적으로 또는 대안적으로, 디코더들은 타일로부터 하나 이상의 선택된 화상들(예를 들어, 이미지들 또는 깊이 신호들)만을 포함하는 디코딩된 출력을 제공할 수 있다.

[0262] 앞서 언급한 바와 같이 고 레벨 신택스는 본 발명의 원리들의 하나 이상의 실시예들에 따라 시그널링을 수행하는데 이용될 수 있다. 고 레벨 신택스는 예를 들어, 큰 프레임에 존재하는 코딩된 뷰들의 수; 모든 뷰들의 원래의 폭 및 높이; 각 코딩된 뷰에 대해, 뷰에 대응하는 뷰 식별자; 각 코딩된 뷰에 대해, 뷰의 프레임이 분할된 부분들의 수; 뷰의 각 부분에 대해, 플립핑 방향(예를 들어, 플립핑 안함, 수평 플립핑만, 수직 플립핑만 또는 수평 및 수직 플립핑일 수 있음); 뷰의 각 부분에 대해, 뷰에 대한 마지막 프레임에서 현재 부분이 속한 다수의 매크로 블록 또는 픽셀들의 좌측 위치; 뷰의 각 부분에 대해, 뷰에 대한 마지막 프레임에서 현재 부분이 속한 다수의 매크로 블록 또는 픽셀들의 부분의 상부 위치; 다수의 매크로 블록들 또는 픽셀들의 크로핑 윈도우의, 현재 큰 디코딩된/인코딩된 프레임에서의 좌측 위치; 뷰의 각 부분에 대해, 다수의 매크로 블록들 또는 픽셀들

의 크로핑 윈도우의, 현재 큰 디코딩된/인코딩된 프레임에서의 우측 위치; 뷰의 각 부분에 대해, 다수의 매크로블록들 또는 픽셀들의 크로핑 윈도우의, 현재 큰 디코딩된/인코딩된 프레임에서의 상부 위치; 및 다수의 매크로블록들 또는 픽셀들의 크로핑 윈도우의, 현재 큰 디코딩된/인코딩된 프레임에서의 하부 위치; 각 코딩된 뷰에 대해, 뷰가 출력 전에 업샘플링이 될 필요가 있는지 여부(만약, 업샘플링이 수행될 필요가 있으면, 고 레벨 신택스는 업샘플링 방법(AVC 6-탭 필터, SVC 4-탭 필터, 이중선형 필터 또는 커스텀 1D,2D 선형 또는 비-선형 필터를 포함하는 것(이것으로 한정되지 않음))을 나타내는데 사용될 수 있음) 중 임의의 것을 시그널링하는데(이에 제한되지 않음) 이용될 수 있다.

[0263] 용어들 "인코더" 및 "디코더"는 일반적인 구조들을 암시하며 임의의 특정한 기능들 또는 특징들로 제한되지 않는다는 점에 주의한다. 예를 들어, 디코더는 인코딩된 비트스트림을 포함하는 변조된 캐리어를 수신하고, 인코딩된 비트스트림을 복조하고, 비트스트림을 디코딩할 수 있다.

[0264] 다양한 방법들이 기술되었다. 대부분의 이 방법들은 충분한 개시를 제공하기 위해 상세히 기술되었다. 그러나, 이 방법들을 위해 기술된 특정 특징들 대부분 또는 특징들 중 하나를 변환할 수 있는 변환물들이 숙고될 수 있다는 점에 주의한다. 또한, 인용된 특징들 대부분이 당 분야에서 잘 알려져 있으며, 따라서, 매우 상세히는 기술하지 않았다.

[0265] 또한, 몇몇 구현들에서 특정 정보를 송신하기 위해 고 레벨 신택스를 이용하도록 참조가 이루어졌다. 그러나, 다른 구현들이 동일한 정보(또는 이 정보의 변환물)를 제공하기 위해 저 레벨 신택스 또는 다른 매커니즘들(예를 들어, 인코딩된 데이터의 부분으로서 정보는 송신하는 등)을 이용할 수 있다는 점에 주의한다.

[0266] 다양한 구현들은 다중 뷰들이 단일 화상으로 타일링되고, 단일 화상으로서 인코딩되고, 단일 화상으로서 송신되는 것을 허용하도록 타일링 및 적절한 시그널링을 제공한다. 시그널링 정보는 후-처리기가 뷰들/화상들을 분리하게 할 수 있다. 또한, 타일링된 다중 화상들은 뷰일 수 있고, 화상들 중 적어도 하나는 깊이 정보일 수 있다. 이 구현들은 하나 이상의 이점을 제공할 수 있다. 예를 들어, 사용자들은 타일링된 방식으로 다중 뷰들을 디스플레이하고자 할 수 있고, 이들 다양한 구현들은 타일링된 방식으로 인코딩 및 전송/저장하기 이전에 타일링함으로써 이 뷰들을 인코딩 및 전송 또는 저장하는 효율적인 방식을 제공한다.

[0267] AVC 및/또는 MVC의 문맥으로 다중 뷰들을 타일링하는 구현들은 또한 부가적인 이점들을 제공한다. AVC는 표면상 단일 뷰만을 위해 사용되고 어떠한 부가적인 뷰들도 기대되지 않는다. 그러나, 예를 들어, 타일링된 화상들이 다른 뷰들(예를 들어, 의사-뷰의 상부 좌측 화상이 뷰 1, 상부 우측 화상이 뷰 2 등)에 속한다는 것을 인지하도록 타일링된 뷰들을 배열하기 때문에, 이러한 AVC-기반 구현들은 AVC 환경에서 다중 뷰들을 제공할 수 있다.

[0268] 부가적으로, MVC는 다중 뷰들을 이미 포함하고, 따라서, 다중 뷰들은 단일 의사-뷰에 포함되는 것으로 기대되지 않는다. 또한, MVC는 지원될 수 있는 뷰들의 수에 대한 한정을 갖고 있고, 이러한 MVC 기반 구현들을 부가적인 뷰들이 타일링되는 것을 허용함으로써(AVC-기반 구현들에서와 같이) 지원될 수 있는 뷰들의 수를 효과적으로 증가시킨다. 예를 들어, 각 의사-뷰가 MVC의 지원되는 뷰들 중 하나에 대응할 수 있고, 디코더는 각 "지원되는 뷰"가 실제로 사전-배열된 타일링 순서에서 4개의 뷰들을 포함한다는 것을 인지할 수 있다. 따라서, 이러한 구현에서, 가능한 뷰들의 수는 "지원되는 뷰들"의 수의 4배이다.

[0269] 여기서 기술된 구현들은 예를 들어, 방법 또는 처리, 장치, 또는 소프트웨어 프로그램으로 구현될 수 있다. 구현의 하나의 유형(예를 들어, 방법으로서만 논의됨)의 문맥으로 논의될지라도, 논의된 특징들의 구현은 다른 유형들(예를 들어, 장치 또는 프로그램)으로 또한 구현될 수 있다. 장치는 예를 들어, 적절한 하드웨어, 소프트웨어, 펌웨어로 구현될 수 있다. 방법들은 일반적으로 예를 들어, 컴퓨터, 마이크로프로세서, 집적 회로 또는 프로그램 가능한 로직 디바이스를 포함하는 처리 디바이스들을 지칭하는 예컨대 프로세서와 같은 장치로 구현될 수 있다. 프처리 디바이스들은 예를 들어, 컴퓨터, 셀 전화기들, 휴대용/개인 휴대 정보 단말기들("PDA들"), 및 종단 사용자들간에 정보의 통신을 용이하게 하는 다른 디바이스들과 같은 통신 디바이스들을 또한 포함한다.

[0270] 여기서 기술된 다양한 프로세스들 및 특징들의 구현들은 구체적으로 예를 들어, 데이터 인코딩 및 디코딩과 연관된 장비 또는 애플리케이션들과 같이 다양한 서로 다른 장비 또는 애플리케이션들로 실시될 수 있다. 장비들의 예들은 비디오 코더들, 비디오 디코더들, 비디오 코덱들, 웹 서버들, 셋-톱 박스들, 랩톱들, 개인용 컴퓨터들, 셀 전화들, PDA들 및 다른 통신 디바이스들을 포함한다. 명확하게 하기 위해서, 장비는 이동될 수 있고, 또한 이동 차량에 설치될 수 있다.

[0271] 부가적으로, 방법들은 프로세서에 의해 수행되는 명령들에 의해 구현될 수 있고, 이러한 명령들은 예를 들어,

집적 회로, 소프트웨어 캐리어 또는 예컨대 하드 디스크, 콤팩트 디스크, 랜덤 액세스 메모리("RAM"), 또는 판독 전용 메모리("ROM")와 같은 다른 저장 디바이스와 같이 프로세서-판독가능 매체상에 저장될 수 있다. 명령들은 프로세서-판독가능 매체상에 실체적으로 구현되는 애플리케이션 프로그램을 형성할 수 있다. 명확하게 하기 위해, 프로세서는 예를 들어, 프로세스를 실행하는 명령들을 갖는 프로세서-판독가능 매체를 포함할 수 있다. 이러한 애플리케이션 프로그램들은 임의의 적절한 아키텍처를 포함하는 기계로 업로드 되거나 상기 기계에 의해 실행될 수 있다. 바람직하게는, 기계는 중앙 처리 장치("CPU"), 랜덤 액세스 메모리("RAM") 및 입력/출력("I/O") 인터페이스들과 같은 하드웨어를 구비한 컴퓨터 플랫폼상에 구현된다. 컴퓨터 플랫폼은 운영 체제 및 마이크로명령 코드를 또한 포함할 수 있다. 여기서 기술된 다양한 프로세스들 기능들은 CPU에 의해 실행될 수 있는 마이크로명령 코드의 부분 또는 애플리케이션 프로그램의 부분 또는 이들의 임의의 조합이 될 수 있다. 또한, 다양한 다른 주변 유닛들은 부가적인 데이터 저장 유닛 및 프린팅 유닛과 같은 컴퓨터 플랫폼에 연결될 수 있다.

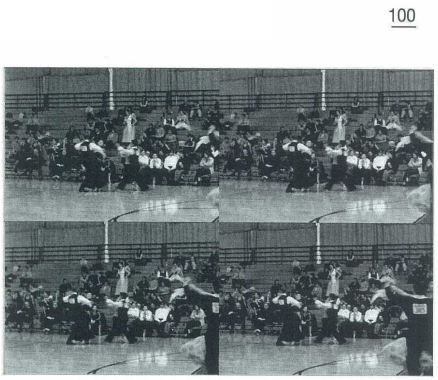
[0272] 당업자에게 명백한 바와 같이, 구현들은 예를 들어, 저장 또는 전송될 수 있는 정보를 전달하도록 포멧팅된 신호를 또한 생성할 수 있다. 정보는 예를 들어, 기술된 구현들 중 하나에 의해 생성되는 데이터 또는 방법을 수행하기 위한 명령들을 포함할 수 있다. 이러한 신호는 예를 들어, 전자기파(예를 들어, 스펙트럼의 라디오 주파수 부분을 이용하여)로서 또는 기저대역 신호로서 포멧팅될 수 있다. 포멧팅은 예를 들어, 데이터 스트림을 인코딩, 신택스를 생성 및 인코딩된 데이터 스트림 및 신택스로 캐리어를 변조하는 것을 포함할 수 있다. 신호가 전달하는 정보는 예를 들어, 아날로그 또는 디지털 정보일 수 있다. 신호는 알려진 바와 같이 다양한 서로 다른 유선 또는 무선 링크들을 통해 전송될 수 있다.

[0273] 첨부된 도면들에서 도시된 구성 시스템 컴포넌트들 및 방법들의 일부는 바람직하게는 소프트웨어로 구현될 수 있기 때문에, 시스템 컴포넌트들간의 실제 연결 또는 프로세스 기능 블록들은 본 원리들이 프로그래밍되는 방식에 따라 상이할 수 있다는 점을 또한 이해해야 한다. 여기에 제공된 교시는 당업자가 본 원리들의 상기 및 유사한 구현들 또는 구성들을 숙고할 수 있을 것이다.

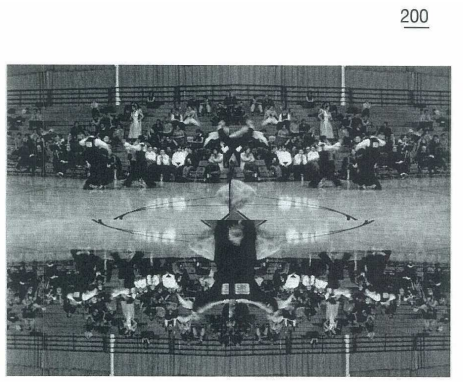
[0274] 다수의 구현들이 기술되었다. 그럼에도 불구하고, 다양한 변환물들이 만들어질 수 있다는 것을 이해할 것이다. 예를 들어, 상이한 구현들의 요소들은 다른 구현들을 생성하도록 조합, 보충, 변환 또는 제거될 수 있다. 부가적으로, 당업자는 다른 구조들 또는 프로세스들이 개시된 것을 대체할 수 있고, 결과적인 구현들이 개시된 구현들과 적어도 실질적으로 동일한 결과(들)를 달성하기 위해 적어도 실질적으로 동일한 방식(들)으로 적어도 실질적으로 동일한 기능(들)을 수행할 것이라는 점을 이해할 것이다. 특히, 예시적인 구현들이 첨부된 도면을 참조하여 여기서 기술되었지만, 본 원리들은 이러한 정확한 실시예들로 제한되지 않고, 다양한 변경 들 및 변환물들이 본 원리들의 범위 또는 사상에서 벗어남 없이 당업자에 의해 유효하게 될 수 있다는 점을 이해할 수 있다. 따라서, 상기 및 다른 구현들은 본 명세서에 의해 숙고되고 다음의 청구범위의 범위 내에 있다.

도면

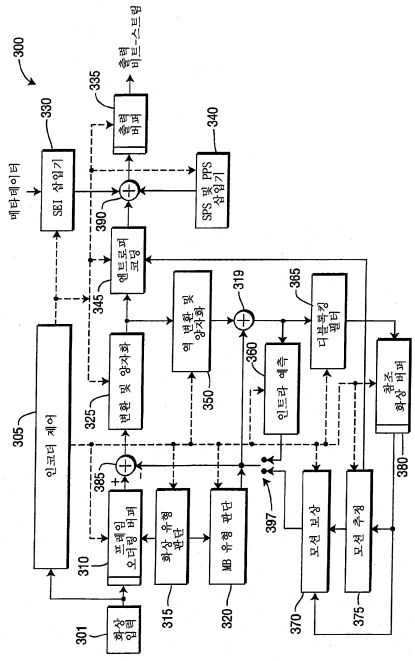
도면1



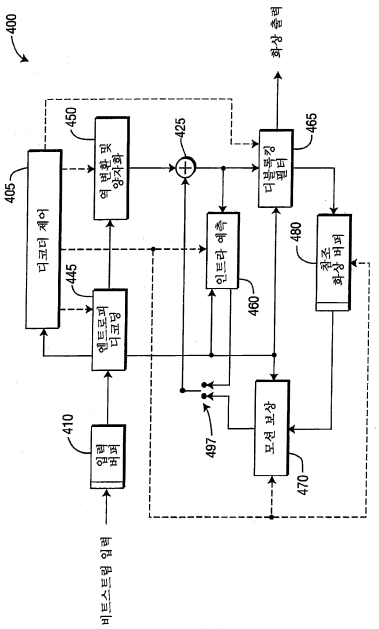
도면2


200

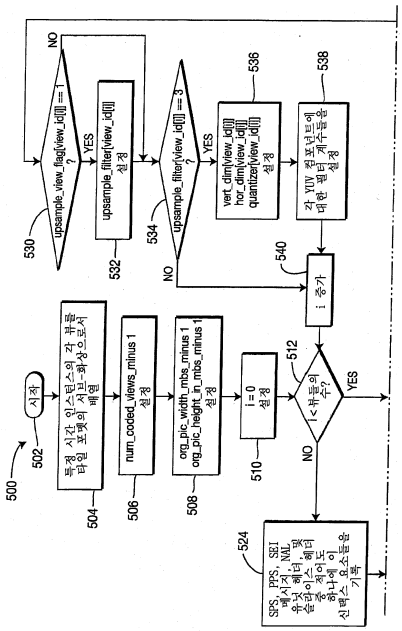
도면3


300
출력 비트스트림
335
출력 버퍼
330
SEI 삽입기
340
SPS 및 PPS 삽입기
390
엔트로피 코딩
345
역 변환 및 양자화
319
365
디블록킹 필터
350
360
인트라 예측
환경 화상 버퍼
380
305
인코더 제어
325
변환 및 양자화
385
397
310
프레임 버퍼
화상 유형 판단
MB 유형 판단
모션 보상
모션 추정
370
375
315
320
301
화상 입력

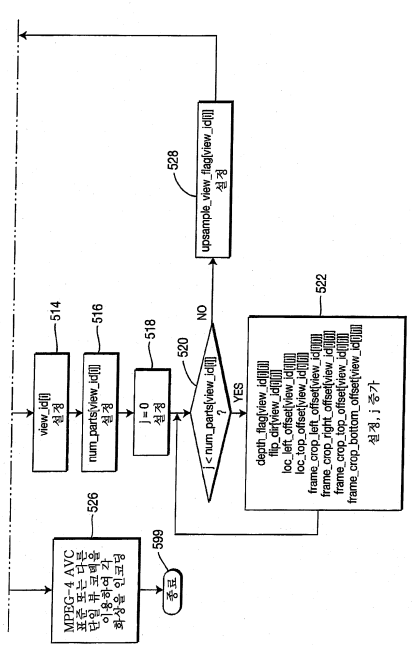
도면4


400
405
디코더 제어
450
역 변환 및 양자화
425
465
디블록킹 필터
화상 출력
445
엔트로피 디코딩
460
인트라 예측
480
참조 화상 버퍼
410
엔탤 버퍼
497
모션 보상
470
비트스트림 입력

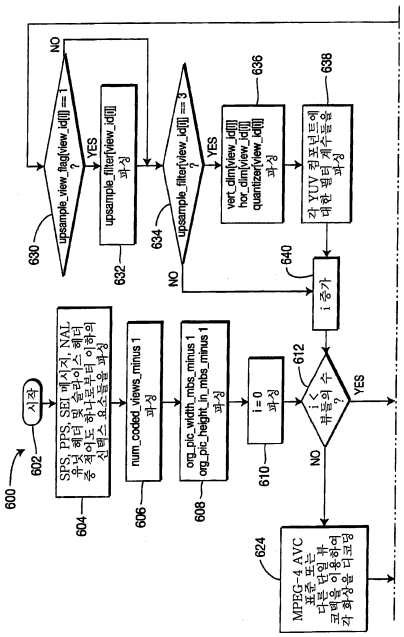
도면5a


500
시작
502
특정 시간 인스턴스의 각 뷰들 담을 포맷의 서브-화상으로서 베열
504
num_coded_views_minus 1 설정
506
org_pic_width_mbs_minus 1
org_pic_height_in_mbs_minus 1 설정
508
i = 0 설정
510
i < 뷰들의 수?
512
YES
NO
i 증가
540
upsample_view_flag[view_id[i]] == 1 ?
530
YES
upsample_filter[view_id[i]] 설정
532
NO
upsample_filter[view_id[i]] == 3 ?
534
YES
vert_dim[view_id[i]]
hor_dim[view_id[i]]
quantizer[view_id[i]] 설정
536
각 VUI 컬러 넌트에 대한 필터 계수들을 설정
538
SPS, PPS, SEI 메시지 헤더 및 슬라이스 헤더 등의 하나에 이 신택스 요소들을 기록
524

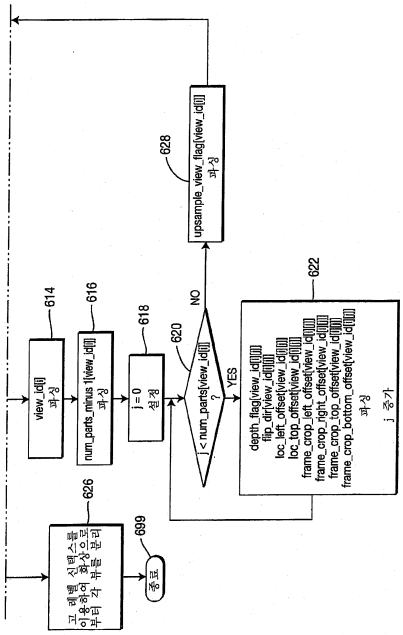
도면5b


514
view_id[i] 설정
516
num_parts[view_id[i]] 설정
518
j = 0 설정
520
j < num_parts[view_id[i]] ?
YES
NO
522
depth_flag[view_id[i]]
flip_dir[view_id[i]]
loc_left_offset[view_id[i]]
loc_top_offset[view_id[i]]
frame_crop_left_offset[view_id[i]]
frame_crop_right_offset[view_id[i]]
frame_crop_top_offset[view_id[i]]
frame_crop_bottom_offset[view_id[i]]
설정, j 증가
526
MPEG-4 AVC 표준 또는 단일 뷰 코덱을 이용하여 각 화상을 인코딩
528
upsample_view_flag[view_id[i]] 설정
599
종료

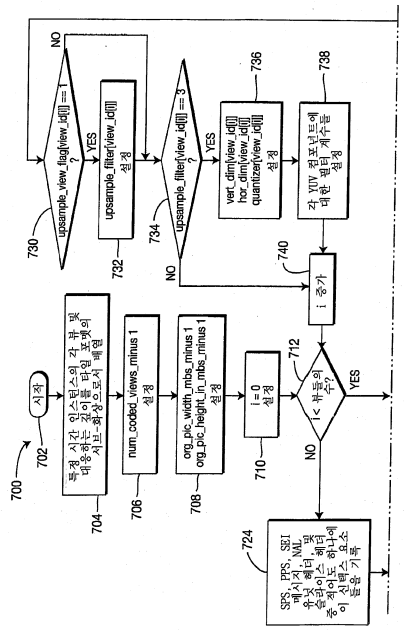
도면6a


600
602
시작
604
SPS, PPS, SEI 메시지, NAL 유닛 헤더 및 슬라이스 헤더 이하의 중첩 신택스 요소들을 파싱
606
num_coded_views_minus 1 파싱
608
org_pic_width_mbs_minus 1
org_pic_height_in_mbs_minus 1 파싱
610
i = 0 설정
612
i < 뷰들의 수 ?
YES
NO
624
MPEG-4 AVC 표준 또는 단일 뷰 코덱을 이용하여 각 화상을 디코딩
630
upsample_view_flag[view_id[i]] == 1 ?
YES
NO
632
upsample_filter[view_id[i]] 파싱
634
upsample_filter[view_id[i]] == 3 ?
YES
NO
636
vert_dim[view_id[i]]
hor_dim[view_id[i]]
quantizer[view_id[i]] 파싱
638
각 YUV 컴포넌트에 대한 멀티 계수들을 파싱
640
i 증가

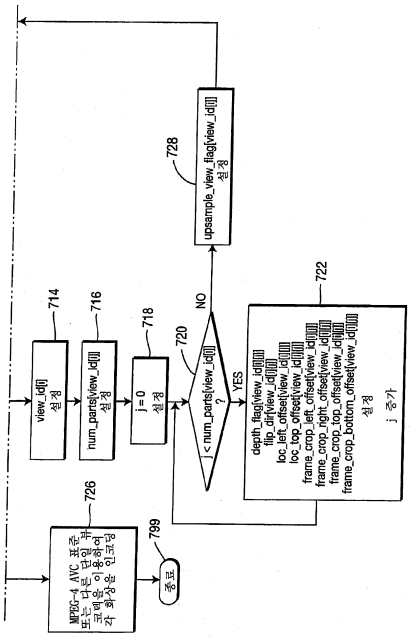
도면6b


614
view_id[i] 파싱
616
num_parts_minus 1[view_id[i]] 파싱
618
j = 0 설정
620
NO
j < num_parts[view_id[i]] ?
YES
622
depth_flag[view_id[i]][j]
dlp_dir[view_id[i]][j]
loc_left_offset[view_id[i]][j]
loc_top_offset[view_id[i]][j]
frame_crop_left_offset[view_id[i]][j]
frame_crop_right_offset[view_id[i]][j]
frame_crop_top_offset[view_id[i]][j]
frame_crop_bottom_offset[view_id[i]][j]
파싱
j 증가
626
고 레벨 신택스를 이용하여 획상으로 뷰들 분리
628
upsample_view_flag[view_id[i]] 파싱
699
종료

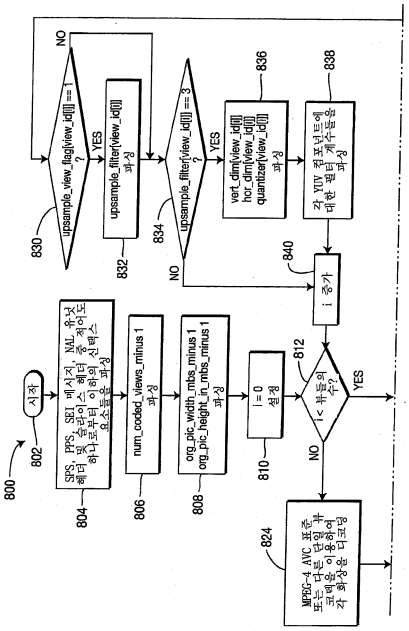
도면7a


700
702
시작
704
특정 시간 인스턴스의 각 픽쳐 및 해당하는 깊이맵 타입 포맷의 시브-획상으로서 배열
706
num_coded_views_minus 1 설정
708
org_pic_width_mbs_minus 1
org_pic_height_in_mbs_minus 1
설정
710
i = 0 설정
712
i < 뷰들의 수 ?
NO
YES
724
SPS, PPS, SEI 메시지 헤더 슬라이스 헤더 중 적어도 하나에 신택스 요소 삽입 기록
730
upsample_view_flag[view_id[i]] == 1 ?
YES
NO
732
upsample_filter[view_id[i]] 설정
734
upsample_filter[view_id[i]] == 3 ?
YES
NO
736
vert_dim[view_id[i]]
hor_dim[view_id[i]]
quantizer[view_id[i]]
설정
738
각 VOV 컴포넌트에 대한 필터 계수들 설정
740
i 증가

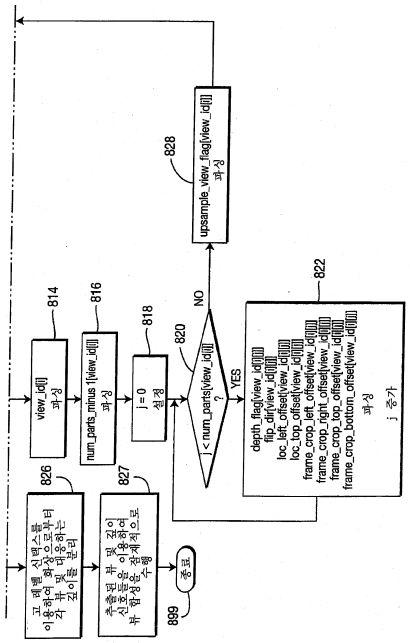
도면7b


714
716
718
720
722
726
728
799
view_id[i] 설정
num_parts[view_id[i]] 설정
j = 0 설정
j < num_parts[view_id[i]] ?
YES
NO
depth_flag[view_id[i]][j]
flip_dir[view_id[i]][j]
loc_left_offset[view_id[i]][j]
loc_top_offset[view_id[i]][j]
frame_crop_left_offset[view_id[i]][j]
frame_crop_right_offset[view_id[i]][j]
frame_crop_top_offset[view_id[i]][j]
frame_crop_bottom_offset[view_id[i]][j]
설정
j 증가
upsample_view_flag[view_id[i]] 설정
MPEG-4 AVC 표준 또는 다른 단일 뷰 코덱을 이용하여 각 화상을 인코딩
종료

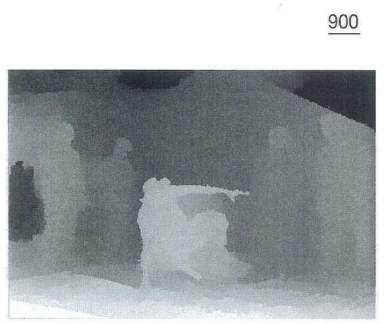
도면8a


800
802
804
806
808
810
812
824
830
832
834
836
838
840
시작
SPS, PPS, SEI 메시지, NAL 유닛 헤더 및 슬라이스 헤더 또는 픽처 이하의 신택스 요소들을 파싱
num_coded_views_minus 1 파싱
org_pic_width_mbs_minus 1
org_pic_height_in_mbs_minus 1 파싱
i = 0 설정
i < 뷰들의 수?
YES
NO
MPEG-4 AVC 표준 또는 다른 단일 뷰 코덱을 이용하여 각 화상을 디코딩
upsample_view_flag[view_id[i]] == 1 ?
YES
NO
upsample_filter[view_id[i]] 파싱
upsample_filter[view_id[i]] == 3 ?
YES
NO
vert_dim[view_id[i]]
hor_dim[view_id[i]]
quantizer[view_id[i]]
파싱
각 YUV 컴포넌트에 대한 필터 계수들을 파싱
i 증가

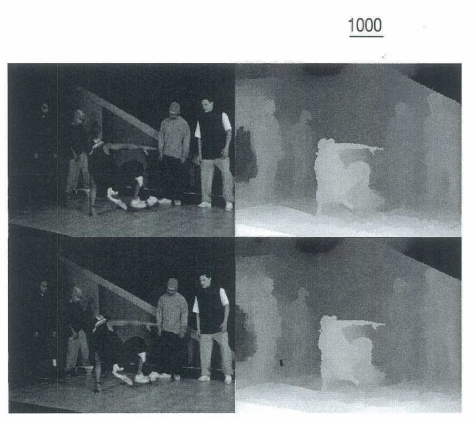
도면8b



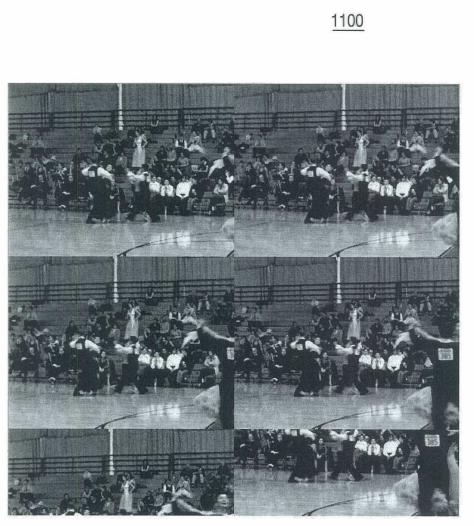
도면9



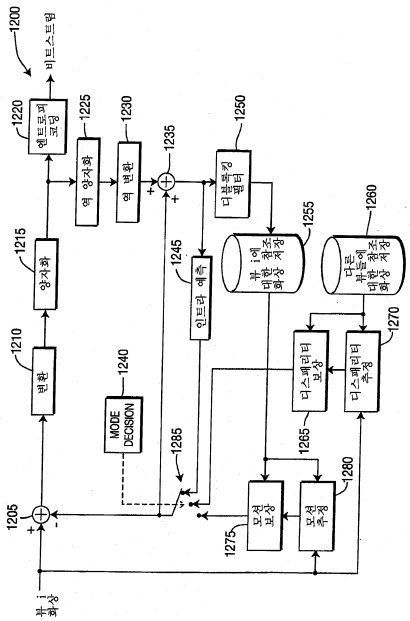
도면10



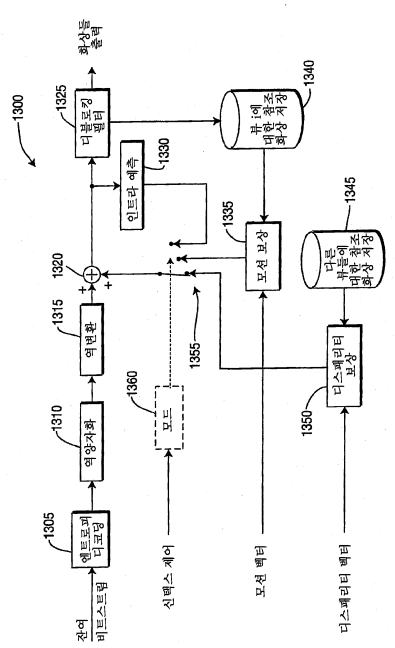
도면11


1100

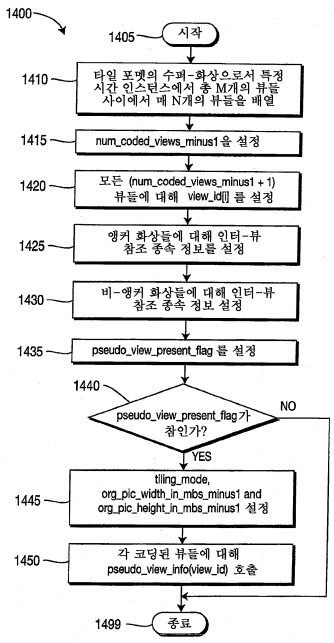
도면12


1200
비트스트림
1220
엔트로피 코딩
1225
역 양자화
1230
역 변환
1235
1250
다블록킹 필터
1255
1260
1215
양자화
1245
인트라 예측
1210
변환
1240
MODE DECISION
1270
디스페리티 보상
디스페리티 추정
1285
1265
1280
1205
1275
모션 보상
모션 추정
복원

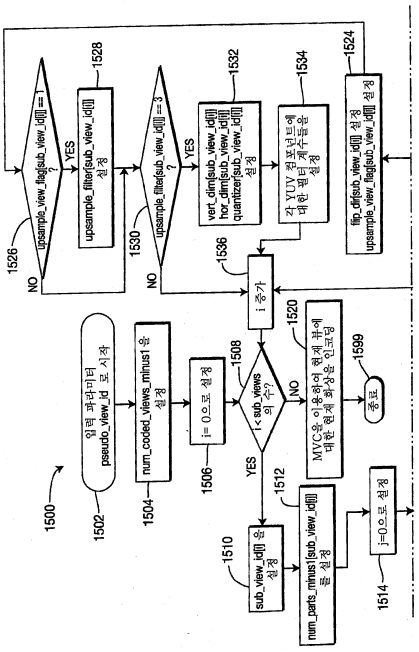
도면13


1300
화상들 출력
1325
디블로킹 필터
1340
다른 뷰에 대한 참조 화상
1330
인트라 예측
1335
모션 보상
1345
다른 뷰들에 대한 참조 화상
1320
1315
역변환
1355
1360
모드
1310
역양자화
1350
디스패리티 보상
1305
엔트로피 디코딩
수신 비트스트림
신택스 제어
모션 벡터
디스패리티 벡터

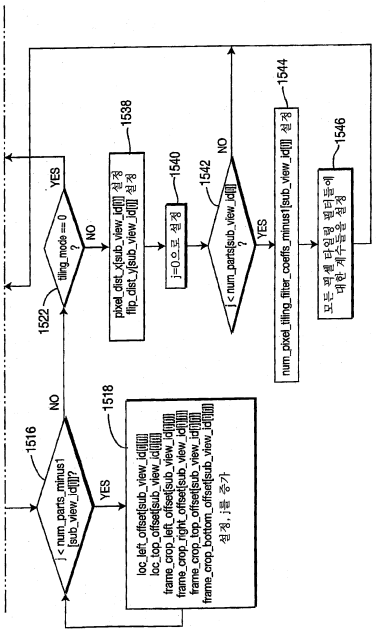
도면14


1400
1405
시작
1410
타일 포맷의 수퍼-화상으로서 특정 시간 인스턴스에서 총 M개의 뷰들 사이에서 매 N개의 뷰들을 배열
1415
num_coded_views_minus1을 설정
1420
모든 (num_coded_views_minus1 + 1) 뷰들에 대해 view_id[] 를 설정
1425
앵커 화상들에 대해 인터-뷰 참조 종속 정보를 설정
1430
비-앵커 화상들에 대해 인터-뷰 참조 종속 정보 설정
1435
pseudo_view_present_flag 를 설정
1440
pseudo_view_present_flag가 참인가?
NO
YES
1445
tiling_mode, org_pic_width_in_mbs_minus1 and org_pic_height_in_mbs_minus1 설정
1450
각 코딩된 뷰들에 대해 pseudo_view_info(view_id) 호출
1499
종료

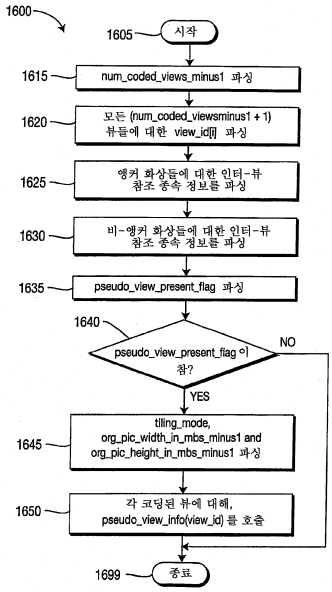
도면15a


1500
1502
입력 파라미터 pseudo_view_id 로 시작
1504
num_coded_views_minus1 을 설정
1506
j= 0으로 설정
1508
i < sub_views 의 수?
YES
NO
1510
sub_view_id[i] 을 설정
1512
num_parts_minus1[sub_view_id[i]] 를 설정
1514
j=0으로 설정
1520
MVC을 이용하여 현재 뷰에 대한 현재 화상을 인코딩
1599
종료
1522
i 증가
1524
flip_dir[sub_view_id[i]] 설정 upsample_view_flag[sub_view_id[i]]
1526
upsample_view_flag[sub_view_id[i]] = 1 ?
1528
upsample_filter[sub_view_id[i]] 설정
1530
upsample_filter[sub_view_id[i]] = 3 ?
1532
vert_dim[sub_view_id[i]] hor_dim[sub_view_id[i]] quantizer[sub_view_id[i]] 설정
1534
각 YUV 컴포넌트에 대한 필터 계수들을 설정

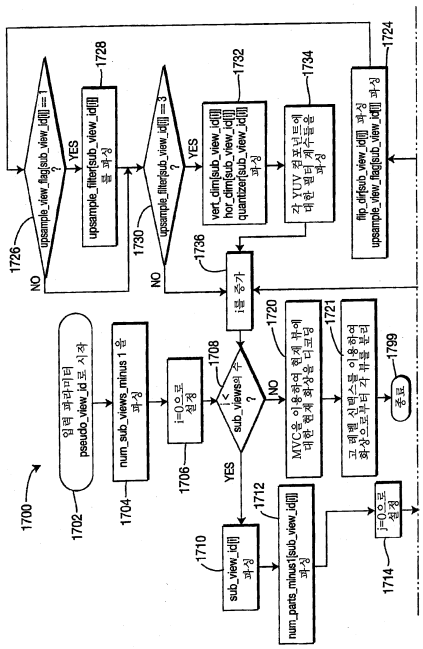
도면15b


1516
j < num_parts_minus1 [sub_view_id[i]]?
NO
YES
1518
loc_left_offset[sub_view_id[i]] loc_top_offset[sub_view_id[i]] frame_crop_left_offset[sub_view_id[i]] frame_crop_right_offset[sub_view_id[i]] frame_crop_top_offset[sub_view_id[i]] frame_crop_bottom_offset[sub_view_id[i]] 설정, j를 증가
1522
tiling_mode == 0 ?
YES
NO
1538
pixel_dist_x[sub_view_id[i]] 설정 flip_dist_y[sub_view_id[i]] 설정
1540
j=0으로 설정
1542
j < num_parts[sub_view_id[i]] ?
NO
YES
1544
num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]] 설정
1546
모든 필터 타일링 필터들에 대한 계수들을 설정

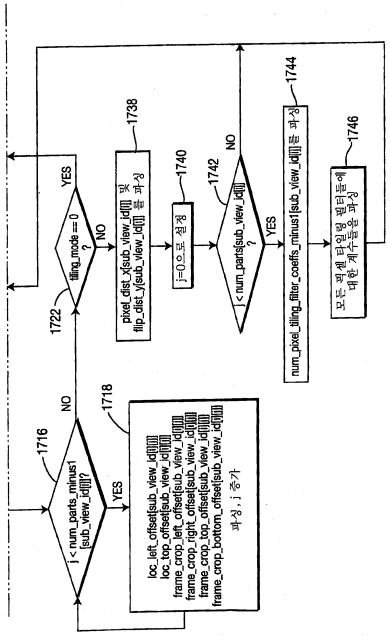
도면16


1600
1605
시작
1615
num_coded_views_minus1 파싱
1620
모든 (num_coded_viewsminus1 + 1) 뷰들에 대한 view_id[i] 파싱
1625
앵커 화상들에 대한 인터-뷰 참조 종속 정보를 파싱
1630
비-앵커 화상들에 대한 인터-뷰 참조 종속 정보를 파싱
1635
pseudo_view_present_flag 파싱
1640
pseudo_view_present_flag 이 참?
NO
YES
1645
tiling_mode, org_pic_width_in_mbs_minus1 and org_pic_height_in_mbs_minus1 파싱
1650
각 코딩된 뷰에 대해, pseudo_view_info(view_id) 를 호출
1699
종료

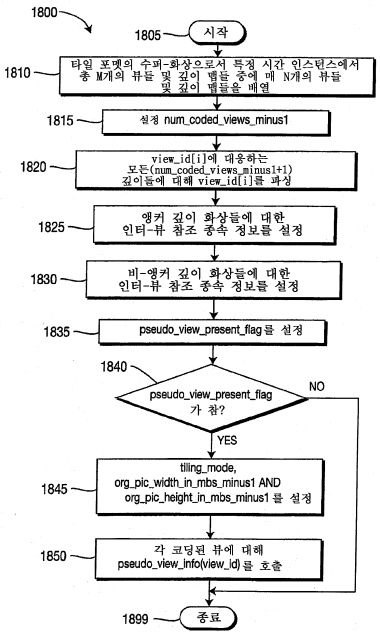
도면17a


1700
1702
임의 파라미터 pseudo_view_id로 시작
1704
num_sub_views_minus 1 을 파싱
1706
i=0으로 설정
1708
i < sub_views의 수 ?
YES
NO
1710
sub_view_id[i] 파싱
1712
num_parts_minus1[sub_view_id[i]] 파싱
1714
j=0으로 설정
1720
MVC용 이용하여 현재 뷰에 대한 현재 화상을 디코딩
1721
고레벨 신택스를 이용하여 참조으로부터 각 부분을 분리
1724
flip_dir[sub_view_id[i]] 파싱
upsample_view_flag[sub_view_id[i]] 파싱
1726
upsample_view_flag[sub_view_id[i]] == 1 ?
YES
NO
1728
upsample_filter[sub_view_id[i]] 를 파싱
1730
upsample_filter[sub_view_id[i]] == 3 ?
YES
NO
1732
vert_dim[sub_view_id[i]] hor_dim[sub_view_id[i]] quantizer[sub_view_id[i]] 파싱
1734
각 YUV 컴포넌트에 대한 필터 계수들을 획득
1736
i 를 증가
1799
종료

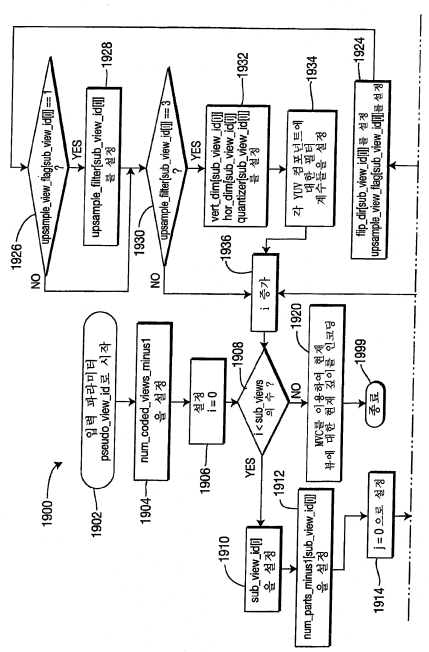
도면17b


tiling_mode = 0 ?
YES
NO
1722
pixel_dist_x[sub_view_id[i]] 및 tile_dist_y[sub_view_id[i]] 를 파싱
1738
j=0으로 설정
1740
j < num_parts[sub_view_id[i]] ?
1742
num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]를 파싱
1744
모든 픽셀 필터링 필터들에 대한 계수들을 파싱
1746
i < num_parts_minus1 [sub_view_id[i]] ?
1716
loc_left_offset[sub_view_id[i]]
loc_top_offset[sub_view_id[i]]
frame_crop_left_offset[sub_view_id[i]]
frame_crop_right_offset[sub_view_id[i]]
frame_crop_top_offset[sub_view_id[i]]
frame_crop_bottom_offset[sub_view_id[i]]
과정 j 증가
1718

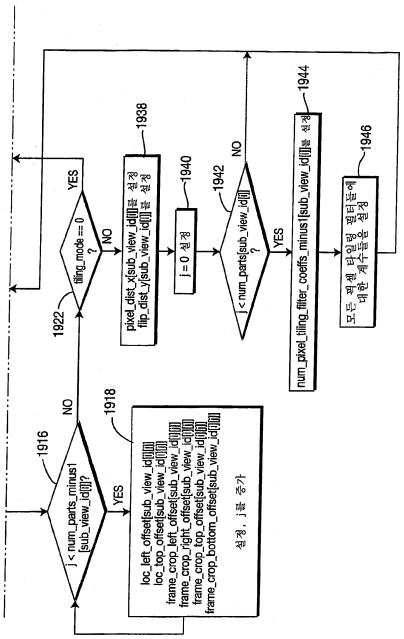
도면18


1800
1805
시작
1810
타일 포멧의 수퍼-화상으로서 특정 시간 인스턴스에서 총 M개의 뷰들 및 깊이 맵들 중에 N개의 뷰들 및 깊이 맵들을 배열
1815
설정 num_coded_views_minus1
1820
view_id[i]에 대응하는 모든(num_coded_views_minus1+1) 깊이들에 대해 view_id[i]를 파싱
1825
앵커 깊이 화상들에 대한 인터-뷰 참조 종속 정보를 설정
1830
비-앵커 깊이 화상들에 대한 인터-뷰 참조 종속 정보를 설정
1835
pseudo_view_present_flag를 설정
1840
pseudo_view_present_flag 가 참?
NO
YES
1845
tiling_mode, org_pic_width_in_mbs_minus1 AND org_pic_height_in_mbs_minus1 를 설정
1850
각 코딩된 뷰에 대해 pseudo_view_info(view_id)를 호출
1899
종료

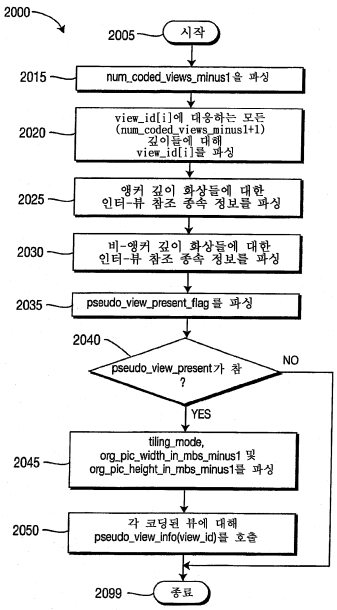
도면19a


1900
1902
입력 파라미터 pseudo_view_id 로 시작
1904
num_coded_views_minus1 을 설정
1906
i 를 0 설정
1908
i < sub_views 의 수 ?
YES
NO
1910
sub_view_id[i] 을 설정
1912
num_parts_minus1[sub_view_id[i]] 을 설정
1914
i 를 0 으로 설정
1916
i < num_parts_minus1[sub_view_id[i]] ?
YES
NO
1918
loc_left_offset[sub_view_id[i]][j]
loc_top_offset[sub_view_id[i]][j]
frame_crop_left_offset[sub_view_id[i]][j]
frame_crop_right_offset[sub_view_id[i]][j]
frame_crop_top_offset[sub_view_id[i]][j]
frame_crop_bottom_offset[sub_view_id[i]][j]
설정, j 를 증가
1920
MVC를 이용하여 현재 뷰에 대한 현재 깊이를 인코딩
1999
종료
1922
tiling_mode == 0 ?
YES
NO
1924
flip_dif[sub_view_id[i]] 을 설정
upsample_view_flag[sub_view_id[i]] 을 설정
1926
upsample_view_flag[sub_view_id[i]] == 1 ?
YES
NO
1928
upsample_filter[sub_view_id[i]] 을 설정
1930
upsample_filter[sub_view_id[i]] == 3 ?
YES
NO
1932
vert_dim[sub_view_id[i]]
hor_dim[sub_view_id[i]]
quantizer[sub_view_id[i]] 을 설정
1934
각 1D/ 컴 포넌트에 대한 계수들을 설정
1936
i 를 증가
1938
pixel_dist_x[sub_view_id[i]] 을 설정
flip_dist_y[sub_view_id[i]] 을 설정
1940
j = 0 설정
1942
j < num_parts[sub_view_id[i]] ?
YES
NO
1944
num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]] 을 설정
1946
모든 픽셀 타일링 필터들에 대한 계수들을 설정

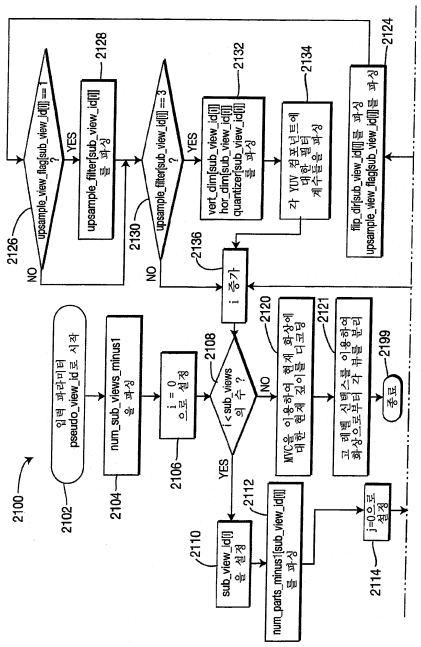
도면19b



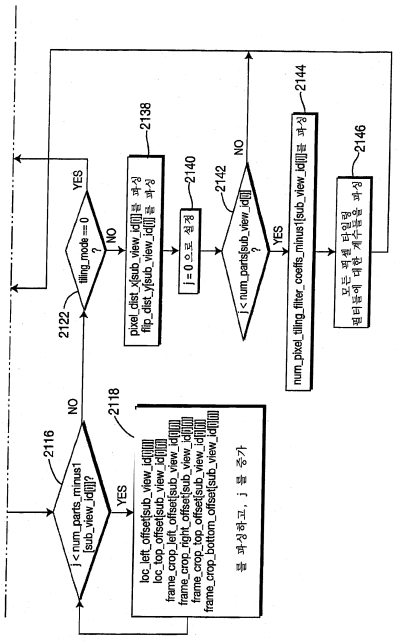
도면20


2000
2005
시작
2015
num_coded_views_minus1을 파싱
2020
view_id[i]에 대응하는 모든 (num_coded_views_minus1+1) 깊이들에 대해 view_id[i]를 파싱
2025
앵커 깊이 화상들에 대한 인터-뷰 참조 종속 정보를 파싱
2030
비-앵커 깊이 화상들에 대한 인터-뷰 참조 종속 정보를 파싱
2035
pseudo_view_present_flag를 파싱
2040
pseudo_view_present가 참 ?
NO
YES
2045
tiling_mode, org_pic_width_in_mbs_minus1 및 org_pic_height_in_mbs_minus1를 파싱
2050
각 코딩된 뷰에 대해 pseudo_view_info(view_id)를 호출
2099
종료

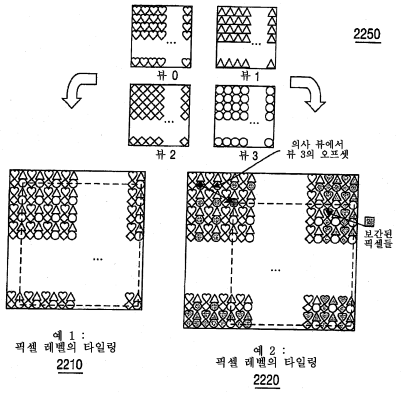
도면21a


2100
2102
입력 파라미터 pseudo_view_id로 시작
2104
num_sub_views_minus1 을 파싱
2106
i = 0 으로 설정
2108
i < sub_views 의 수 ?
YES
NO
2110
sub_view_id[i] 을 설정
2112
num_parts_minus1[sub_view_id[i]] 를 파싱
2114
i=0으로 설정
2120
MVC을 이용하여 현재 화상에 대한 현재 깊이를 디코딩
2121
고 레벨 신택스를 이용하여 화상으로부터 뷰를 분리
2199
종료
2124
flip_dir[sub_view_id[i]]를 파싱
upsample_view_flag[sub_view_id[i]]를 파싱
2126
upsample_view_flag[sub_view_id[i]] = 1 ?
YES
NO
2128
upsample_filter[sub_view_id[i]] 를 파싱
2130
upsample_filter[sub_view_id[i]] = 3 ?
YES
NO
2132
vert_dim[sub_view_id[i]] hor_dim[sub_view_id[i]] quantizer[sub_view_id[i]] 를 파싱
2134
각 YUV 컴포넌트에 대한 필터 계수를 파싱
2136
i 증가

도면21b



도면22



도면23