



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2013년08월05일  
(11) 등록번호 10-1293425  
(24) 등록일자 2013년07월30일

(51) 국제특허분류(Int. Cl.)  
    H04N 7/24 (2011.01) H04N 7/26 (2006.01)  
(21) 출원번호 10-2012-7006005  
(22) 출원일자(국제) 2010년08월06일  
    심사청구일자 2012년03월06일  
(85) 번역문제출일자 2012년03월06일  
(65) 공개번호 10-2012-0054052  
(43) 공개일자 2012년05월29일  
(86) 국제출원번호 PCT/US2010/044780  
(87) 국제공개번호 WO 2011/017661  
    국제공개일자 2011년02월10일  
(30) 우선권주장  
    12/757,231 2010년04월09일 미국(US)  
    (뒷면에 계속)  
(56) 선행기술조사문헌  
    US20100376122 A1  
    W02009048502 A2  
전체 청구항 수 : 총 43 항

(73) 특허권자  
    **퀄컴 인코포레이티드**  
    미국 92121-1714 캘리포니아주 샌 디에고 모어하  
    우스 드라이브 5775  
(72) 발명자  
    **천 잉**  
    미국 92121-1714 캘리포니아주 샌디에고 모어하우  
    스 드라이브 5775  
    **천페이송**  
    미국 92121-1714 캘리포니아주 샌디에고 모어하우  
    스 드라이브 5775  
    **카르체비츠 마르타**  
    미국 92121-1714 캘리포니아주 샌디에고 모어하우  
    스 드라이브 5775  
(74) 대리인  
    **특허법인코리아나**

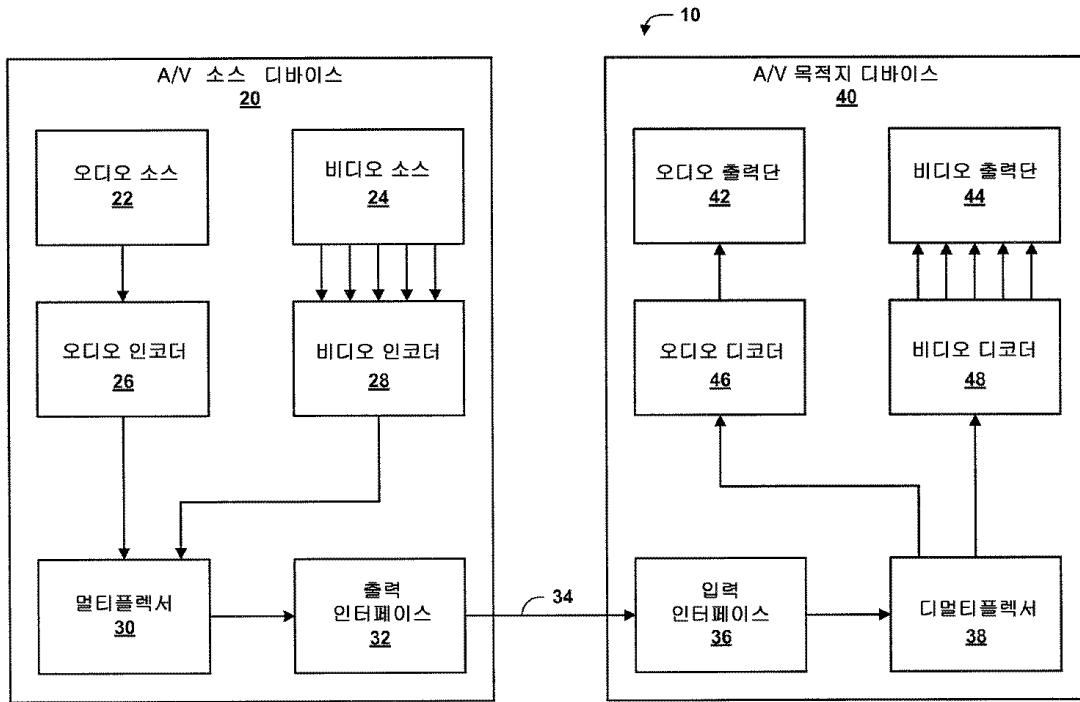
심사관 : 조우연

(54) 발명의 명칭 **MVC 동작점 특성들의 시그널링**

**(57) 요약**

소스 및 목적지 비디오 디바이스들이 MPEG-2 (Motion Picture Experts Group) 시스템 비트스트림에 대한 동작점의 세부사항들을 시그널링하는 데이터 구조들을 사용할 수 있다. 하나의 예에서, 장치는 MPEG-2 (Motion Picture Experts Group) 시스템 표준 비트스트림의 멀티뷰 비디오 코딩 (MVC) 동작점에 상응하는 데이터 구조를 구축하는 멀티플렉서로서, 데이터 구조는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 랜더링 능력을 기술하는 랜더링 능력 값, MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하고, 데이터 구조를 비트스트림의 일부로서 포함하는, 멀티플렉서, 및 이 데이터 구조를 포함하는 비트스트림을 출력하는 출력 인터페이스를 구비한다.

대표도



(30) 우선권주장

61/232,272 2009년08월07일 미국(US)

61/248,738 2009년10월05일 미국(US)

61/266,861 2009년12월04일 미국(US)

## 특허청구의 범위

### 청구항 1

소스 디바이스 (source device) 로, MPEG-2 (Motion Picture Experts Group) 시스템 표준 비트스트림의 멀티뷰 비디오 코딩 (MVC) 동작점에 각각 상응하는 복수의 MVC 기술자 (descriptor) 들을 구축하는 단계로서, 각 MVC 기술자는 상기 상응하는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 랜더링 능력을 기술하는 랜더링 능력 값, 및 상기 상응하는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값을 시그널링하고, 각 MVC 기술자는 상기 비트스트림의 일부로서 포함되는, 상기 복수의 MVC 기술자들을 구축하는 단계; 및

상기 복수의 MVC 기술자들을 포함하는 상기 비트스트림을 출력하는 단계를 포함하는, 멀티미디어 데이터를 전송하는 방법.

### 청구항 2

제 1 항에 있어서,

각 MVC 기술자를 구축하는 단계는 하나 이상의 2차원 디스플레이 디바이스들 및 3차원 디스플레이 디바이스들로 하여금 상기 비트스트림을 상기 하나 이상의 2차원 디스플레이 디바이스들 및 3차원 디스플레이 디바이스들에 적응하게 하고 상기 하나 이상의 2차원 디스플레이 디바이스들 및 3차원 디스플레이 디바이스들로의 다양한 대역폭들의 전송 미디어를 수용하게 하도록 상기 MVC 기술자를 구축하는 단계를 포함하는, 멀티미디어 데이터를 전송하는 방법.

### 청구항 3

제 1 항에 있어서,

상기 랜더링 능력 값은 적어도, 상기 상응하는 MVC 동작점에 대한 랜더링의 대상이 되는 뷰들의 수, 상기 상응하는 MVC 동작점의 비디오 데이터에 대한 프레임 레이트, 및 상기 상응하는 MVC 동작점에 대한 시간적 식별자 값을 기술하는, 멀티미디어 데이터를 전송하는 방법.

### 청구항 4

제 1 항에 있어서,

상기 디코딩 능력 값은 적어도, 상기 상응하는 MVC 동작점에 대한 디코딩될 뷰들의 수, 상기 MVC 동작점에 상응하는 레벨 값, 및 상기 MVC 동작점에 상응하는 프로파일 값을 기술하는, 멀티미디어 데이터를 전송하는 방법.

### 청구항 5

제 1 항에 있어서,

상기 복수의 MVC 기술자들을 구축하는 단계는 상기 상응하는 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하기 위해 상기 복수의 MVC 기술자들을 구축하는 단계를 포함하고,

상기 비트레이트 값은 상기 상응하는 MVC 동작점에 대한 평균 비트레이트 및 상기 상응하는 MVC 동작점에 대한 최대 비트레이트 중 하나를 기술하는, 멀티미디어 데이터를 전송하는 방법.

### 청구항 6

제 1 항에 있어서,

각 MVC 기술자를 구축하는 단계는, 프로그램 맵 테이블 데이터 구조 또는 프로그램 스트림 맵 데이터 구조에서 동작점 기술자를 포함하고, 상기 비트스트림은 MPEG-2 프로그램 스트림을 포함하는, 멀티미디어 데이터를 전송하는 방법.

### 청구항 7

삭제

**청구항 8**

제 1 항에 있어서,

각 MVC 기술자는 동작점 기술자를 포함하고,

각 MVC 기술자를 구축하는 단계는,

상기 MVC 동작점에 대한 뷰들의 수에 포함된 비디오 데이터에 대한 최대 프레임 레이트를 기술하는 프레임 레이트 값을 동작점 기술자에서 포함하는 단계;

상기 MVC 동작점의 렌더링의 대상이 되는 뷰들의 수에 대한 뷰-식별자 값들을 상기 동작점 기술자에서 포함하는 단계로서, 상기 뷰-식별자 값들 각각은 렌더링의 대상이 되는 뷰들 중의 하나에 상응하는, 상기 렌더링의 대상이 되는 뷰들에 대한 뷰-식별자 값들을 포함하는 단계;

상기 MVC 동작점에 대한 디코딩될 뷰들의 수에 대한 뷰-식별자 값들을 상기 동작점 기술자에서 포함하는 단계로서, 상기 뷰-식별자 값들 각각은 디코딩될 뷰들 중의 하나에 상응하는, 상기 디코딩될 뷰들에 대한 뷰-식별자 값들을 포함하는 단계; 및

상기 MVC 동작점에 대한 뷰들의 비디오 데이터로부터 어셈블링된 비디오 스트림에 대한 프레임 레이트에 상응하는 시간적-식별자 값을 동작점 기술자에서 포함하는 단계를 포함하는, 멀티미디어 데이터를 전송하는 방법.

**청구항 9**

삭제

**청구항 10**

MPEG-2 (Motion Picture Experts Group) 시스템 표준 비트스트림의 멀티뷰 비디오 코딩 (MVC) 동작점에 각각 상응하는 복수의 MVC 기술자들을 구축하고, 상기 비트스트림의 일부로서 각 MVC 기술자를 포함하는 멀티플렉서로서, 각 MVC 기술자는 상기 상응하는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, 및 상기 상응하는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값을 시그널링하는, 상기 멀티플렉서; 및

상기 복수의 MVC 기술자들을 포함하는 상기 비트스트림을 출력하는 출력 인터페이스를 포함하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 11**

제 10 항에 있어서,

상기 렌더링 능력 값은 적어도, 상기 상응하는 MVC 동작점에 대한 렌더링의 대상이 되는 뷰들의 수, 상기 상응하는 MVC 동작점의 비디오 데이터에 대한 프레임 레이트, 및 상기 상응하는 MVC 동작점에 대한 시간적 식별자 값을 기술하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 12**

제 10 항에 있어서,

상기 디코딩 능력 값은 적어도, 상기 상응하는 MVC 동작점에 대한 디코딩될 뷰들의 수, 상기 MVC 동작점에 상응하는 레벨 값, 및 상기 MVC 동작점에 상응하는 프로파일 값을 기술하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 13**

제 10 항에 있어서,

상기 멀티플렉서는 상기 상응하는 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하기 위해 상기 복수의 MVC 기술자들을 구축하고,

상기 비트레이트 값은 상기 상응하는 MVC 동작점에 대한 평균 비트레이트 및 상기 상응하는 MVC 동작점에 대한

최대 비트레이트 중 하나를 기술하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 14**

제 10 항에 있어서,

상기 멀티플렉서는 프로그램 맵 테이블 데이터 구조 또는 프로그램 스트림 맵 데이터 구조에서 동작점 기술자로서 각 MVC 기술자를 구축하고, 상기 비트스트림은 MPEG-2 프로그램 스트림을 포함하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 15**

삭제

**청구항 16**

제 10 항에 있어서,

상기 멀티플렉서는 각 MVC 기술자를 동작점 기술자로서 구축하며, 각 MVC 동작점은 상기 비트스트림의 뷰들의 수의 서브세트에 상응하고, 각 MVC 기술자를 구축하기 위해, 상기 멀티플렉서는 상기 상응하는 MVC 동작점에 대한 뷰들에 포함된 비디오 데이터에 대한 최대 프레임 레이트를 기술하는 프레임 레이트 값을 상기 동작점 기술자에서 포함하며, 상기 상응하는 MVC 동작점의 렌더링의 대상이 되는 뷰들에 대한 뷰 식별자 값들을 상기 동작점 기술자에서 포함하고, 상기 뷰-식별자 값들 각각은 렌더링의 대상이 되는 뷰들 중의 하나에 상응하고, 상기 멀티플렉서는 상기 상응하는 MVC 동작점에 대한 디코딩될 뷰들에 대한 뷰-식별자 값들을 상기 동작점 기술자에서 포함하고, 상기 뷰-식별자 값들 각각은 디코딩될 뷰들 중의 하나에 상응하며, 상기 멀티플렉서는 상기 상응하는 MVC 동작점에 대한 상기 뷰들의 상기 비디오 데이터로부터 어셈블링된 비디오 스트림에 대한 프레임 레이트에 상응하는 시간적-식별자 값을 상기 동작점 기술자에서 포함하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 17**

삭제

**청구항 18**

제 10 항에 있어서,

상기 장치는,

집적 회로;

마이크로프로세서; 및

멀티플렉서를 구비한 무선 통신 디바이스 중 적어도 하나를 포함하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 19**

MPEG-2 (Motion Picture Experts Group) 시스템 표준 비트스트림의 멀티뷰 비디오 코딩 (MVC) 동작점에 각각 상응하는 복수의 MVC 기술자들을 구축하는 수단으로서, 각 MVC 기술자는 상기 상응하는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, 및 상기 상응하는 MVC 동작점을 사용하기 위해 상기 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값을 시그널링하고, 각 MVC 기술자는 비트스트림의 일부로서 포함되는, 상기 복수의 MVC 기술자들을 구축하는 수단; 및

상기 복수의 MVC 기술자들을 포함하는 상기 비트스트림을 출력하는 수단을 포함하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 20**

제 19 항에 있어서,

상기 렌더링 능력 값은 적어도, 상기 상응하는 MVC 동작점에 대한 렌더링의 대상이 되는 뷰들의 수, 상기 상응하는 MVC 동작점의 비디오 데이터에 대한 프레임 레이트, 및 상기 상응하는 MVC 동작점에 대한 시간적 식별자

값을 기술하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 21**

제 19 항에 있어서,

상기 디코딩 능력 값은 적어도, 상기 상응하는 MVC 동작점에 대한 디코딩될 뷰들의 수, 상기 MVC 동작점에 상응하는 레벨 값, 및 상기 MVC 동작점에 상응하는 프로파일 값을 기술하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 22**

제 19 항에 있어서,

상기 복수의 MVC 기술자들을 구축하는 수단은 상기 상응하는 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하기 위해 상기 복수의 MVC 기술자들을 구축하고,

상기 비트레이트 값은 상기 상응하는 MVC 동작점에 대한 평균 비트레이트 및 상기 상응하는 MVC 동작점에 대한 최대 비트레이트 중 하나를 기술하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 23**

제 19 항에 있어서,

각 MVC 기술자는 동작점 기술자를 포함하고,

각 MVC 기술자를 구축하는 수단은,

상기 MVC 동작점에 대한 뷰들의 수에 포함된 비디오 데이터에 대한 최대 프레임 레이트를 기술하는 프레임 레이트 값을 동작점 기술자에서 포함하는 수단;

상기 MVC 동작점의 랜더링의 대상이 되는 뷰들의 수에 대한 뷰-식별자 값들을 상기 동작점 기술자에서 포함하는 수단으로서, 상기 뷰-식별자 값들 각각은 랜더링의 대상이 되는 뷰들 중의 하나에 상응하는, 상기 랜더링의 대상이 되는 뷰들에 대한 뷰-식별자 값들을 포함하는 수단;

상기 MVC 동작점에 대한 디코딩될 뷰들의 수에 대한 뷰-식별자 값들을 상기 동작점 기술자에서 포함하는 수단으로서, 상기 뷰-식별자 값들 각각은 디코딩될 뷰들 중의 하나에 상응하는, 상기 디코딩될 뷰들에 대한 뷰-식별자 값들을 포함하는 수단; 및

상기 MVC 동작점에 대한 뷰들의 비디오 데이터로부터 어셈블링된 비디오 스트림에 대한 프레임 레이트에 상응하는 시간적-식별자 값을 동작점 기술자에서 포함하는 수단을 포함하는, 멀티미디어 데이터를 전송하는 장치.

**청구항 24**

실행될 때, 소스 디바이스의 프로세서로 하여금,

MPEG-2 (Motion Picture Experts Group) 시스템 표준 비트스트림의 멀티뷰 비디오 코딩 (MVC) 동작점에 각각 상응하는 복수의 MVC 기술자들을 구축하게 하는 명령어로서, 각 MVC 기술자는 상기 상응하는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 랜더링 능력을 기술하는 랜더링 능력 값, 및 상기 상응하는 MVC 동작점을 사용하기 위해 상기 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값을 시그널링하고, 각 MVC 기술자는 비트스트림의 일부로서 포함되는, 상기 복수의 MVC 기술자들을 구축하게 하는 명령어; 및

출력 인터페이스로 하여금 상기 복수의 MVC 기술자들을 포함하는 상기 비트스트림을 출력하게 하는 명령어를 포함하는, 멀티미디어 데이터를 전송하기 위한 명령어들을 포함하는 컴퓨터 판독가능 저장 매체.

**청구항 25**

제 24 항에 있어서,

상기 랜더링 능력 값은 적어도, 상기 상응하는 MVC 동작점에 대한 랜더링의 대상이 되는 뷰들의 수, 상기 상응하는 MVC 동작점의 비디오 데이터에 대한 프레임 레이트, 및 상기 상응하는 MVC 동작점에 대한 시간적 식별자 값을 기술하는, 멀티미디어 데이터를 전송하기 위한 명령어들을 포함하는 컴퓨터 판독가능 저장 매체.

**청구항 26**

제 24 항에 있어서,

상기 디코딩 능력 값은 적어도, 상기 상응하는 MVC 동작점에 대한 디코딩될 뷰들의 수, 상기 MVC 동작점에 상응하는 레벨 값, 및 상기 MVC 동작점에 상응하는 프로파일 값을 기술하는, 멀티미디어 데이터를 전송하기 위한 명령어들을 포함하는 컴퓨터 판독가능 저장 매체.

**청구항 27**

제 24 항에 있어서,

상기 프로세서로 하여금, 상기 복수의 MVC 기술자들을 구축하게 하는 명령어는 상기 프로세서로 하여금, 상기 상응하는 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하기 위해 상기 복수의 MVC 기술자들을 구축하게 하고,

상기 비트레이트 값은 상기 상응하는 MVC 동작점에 대한 평균 비트레이트 및 상기 상응하는 MVC 동작점에 대한 최대 비트레이트 중 하나를 기술하는, 멀티미디어 데이터를 전송하기 위한 명령어들을 포함하는 컴퓨터 판독가능 저장 매체.

**청구항 28**

제 24 항에 있어서,

상기 MVC 기술자는 동작점 기술자를 포함하고, 상기 프로세서로 하여금 각 MVC 기술자를 구축하게 하는 명령어는, 상기 프로세서로 하여금,

상기 MVC 동작점에 대한 뷰들의 수에 포함된 비디오 데이터에 대한 최대 프레임 레이트를 기술하는 프레임 레이트 값을 동작점 기술자에서 포함하게 하는 명령어;

상기 MVC 동작점의 랜더링의 대상이 되는 뷰들의 수에 대한 뷰-식별자 값들을 상기 동작점 기술자에서 포함하게 하는 명령어로서, 상기 뷰-식별자 값들 각각은 랜더링의 대상이 되는 뷰들 중의 하나에 상응하는, 상기 랜더링의 대상이 되는 뷰들에 대한 뷰-식별자 값들을 포함하게 하는 명령어;

상기 MVC 동작점에 대한 디코딩될 뷰들의 수에 대한 뷰-식별자 값들을 상기 동작점 기술자에서 포함하게 하는 명령어로서, 상기 뷰-식별자 값들 각각은 디코딩될 뷰들 중의 하나에 상응하는, 상기 디코딩될 뷰들에 대한 뷰-식별자 값들을 포함하게 하는 명령어; 및

상기 MVC 동작점에 대한 뷰들의 비디오 데이터로부터 어셈블링된 비디오 스트림에 대한 프레임 레이트에 상응하는 시간적-식별자 값을 동작점 기술자에서 포함하게 하는 명령어를 포함하는, 멀티미디어 데이터를 전송하기 위한 명령어들을 포함하는 컴퓨터 판독가능 저장 매체.

**청구항 29**

목적지 디바이스로, MPEG-2 (Motion Picture Experts Group) 시스템 표준 비트스트림의 멀티뷰 비디오 코딩 (MVC) 동작점에 상응하는 복수의 MVC 기술자들을 수신하는 단계로서, 각 MVC 기술자는 상기 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 랜더링 능력을 기술하는 랜더링 능력 값, 및 상기 MVC 동작점을 사용하기 위해 상기 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값을 시그널링하는, 상기 복수의 MVC 기술자들을 수신하는 단계;

상기 MVC 기술자에 의해 시그널링된 디코딩 능력에 기초하여 상기 목적지 디바이스의 비디오 디코더가 상기 MVC 동작점에 상응하는 뷰들의 수를 디코딩할 수 있는지를 각 MVC 기술자마다 결정하는 단계;

상기 MVC 기술자에 의해 시그널링된 랜더링 능력에 기초하여 상기 목적지 디바이스가 상기 MVC 동작점에 상응하는 상기 뷰들을 랜더링할 수 있는지를 각 MVC 기술자마다 결정하는 단계;

상기 상응하는 MVC 기술자에 기초하여 동작점을 선택하는 단계로서, 상기 선택하는 단계는 상기 비디오 디코더가 상기 선택된 동작점에 상응하는 뷰들을 디코딩 및 랜더링할 수 있다는 것을 결정하는 단계를 포함하는, 상기 동작점을 선택하는 단계; 및

상기 선택된 MVC 동작점에 상응하는 상기 뷰들을 상기 목적지 디바이스의 상기 비디오 디코더에 송신하는 단계

를 포함하는, 멀티미디어 데이터를 수신하는 방법.

**청구항 30**

제 29 항에 있어서,

각 MVC 기술자는 동작점 기술자를 포함하고, 상기 동작점 기술자는 상기 MVC 동작점에 대한 상기 뷰들에 포함된 비디오 데이터에 대한 최대 프레임 레이트를 기술하는 프레임 레이트 값, 상기 MVC 동작점의 렌더링의 대상이 되는 뷰들의 수에 대한 뷰-식별자 값들을 포함하고, 상기 뷰-식별자 값들 각각은 렌더링의 대상이 되는 상기 뷰들 중의 하나에 상응하고, 상기 동작점 기술자는 상기 MVC 동작점에 대한 디코딩될 뷰들의 수에 대한 뷰-식별자 값들을 포함하고, 상기 뷰-식별자 값들 각각은 디코딩될 상기 뷰들 중의 하나에 상응하며, 상기 동작점 기술자는 상기 MVC 동작점에 대한 상기 뷰들의 상기 비디오 데이터로부터 어셈블링된 비디오 스트림에 대한 프레임 레이트에 상응하는 시간적-식별자 값을 포함하는, 멀티미디어 데이터를 수신하는 방법.

**청구항 31**

제 30 항에 있어서,

상기 비디오 디코더가 뷰들을 디코딩할 수 있는지를 결정하는 단계는, 상기 비디오 디코더가 상기 프레임 레이트 값에 의해 지시되는 프레임 레이트에서 디코딩될 뷰들의 수에 동등한 뷰들의 수를 디코딩할 수 있는지를 결정하는 단계를 포함하는, 멀티미디어 데이터를 수신하는 방법.

**청구항 32**

삭제

**청구항 33**

제 29 항에 있어서,

상기 목적지 디바이스는 상기 목적지 디바이스에 의해 렌더링될 수 있는 지원된 뷰들의 수를 기술하는 지원된 뷰들의 수 및 상기 목적지 디바이스에 의해 디스플레이될 수 있는 비디오 데이터의 프레임 레이트를 기술하는 프레임 레이트 값으로 구성되며,

상기 목적지 디바이스가 상기 MVC 동작점에 상응하는 상기 뷰들을 렌더링할 수 있는지를 결정하는 단계는,

상기 MVC 동작점에 상응하는 뷰들의 수를 상기 지원된 뷰들의 수와 비교하는 단계; 및

상기 MVC 동작점에 상응하는 상기 뷰들의 프레임 레이트를 상기 프레임 레이트 값과 비교하는 단계를 포함하고,

상기 MVC 동작점에 상응하는 상기 뷰들을 상기 비디오 디코더에 송신하는 단계는, 상기 MVC 동작점에 상응하는 상기 뷰들의 수가 상기 지원된 뷰들의 수 이하인 경우 및 상기 MVC 동작점에 상응하는 상기 뷰들의 상기 프레임 레이트가 상기 프레임 레이트 값 이하인 경우, 상기 MVC 동작점에 상응하는 뷰들을 상기 비디오 디코더에 송신하는 단계를 포함하는, 멀티미디어 데이터를 수신하는 방법.

**청구항 34**

제 33 항에 있어서,

상기 지원된 뷰들의 수는 상기 프레임 레이트 값에 역비례하는, 멀티미디어 데이터를 수신하는 방법.

**청구항 35**

MPEG-2 (Motion Picture Experts Group) 시스템 표준 비트스트림의 멀티뷰 비디오 코딩 (MVC) 동작점에 상응하는 복수의 MVC 기술자들을 수신하도록 구성된 입력 인터페이스로서, 각 MVC 기술자는 상기 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, 및 상기 MVC 동작점을 사용하기 위해 상기 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값을 시그널링하는, 상기 입력 인터페이스;

비디오 데이터를 디코딩하도록 구성된 비디오 디코더; 및

상기 MVC 기술자에 의해 시그널링된 디코딩 능력에 기초하여 상기 비디오 디코더가 상기 MVC 동작점에 상응하는



뷰들의 수를 디코딩할 수 있는지를 각 MVC 기술자마다 결정하며, 상기 MVC 기술자에 의해 시그널링된 렌더링 능력에 기초하여 장치가 상기 MVC 동작점에 상응하는 뷰들을 렌더링할 수 있는지를 각 MVC 기술자마다 결정하며, 상기 비디오 디코더가 선택된 동작점에 상응하는 뷰들을 디코딩 및 렌더링할 수 있다는 것을 결정함으로써 상기 상응하는 MVC 기술자에 기초하여 동작점을 선택하며, 상기 선택된 MVC 동작점에 상응하는 상기 뷰들을 상기 비디오 디코더에 송신하도록 구성된 디멀티플렉서를 포함하는, 멀티미디어 데이터를 수신하는 장치.

**청구항 36**

제 35 항에 있어서,

각 MVC 기술자는 동작점 기술자를 포함하고, 상기 동작점 기술자는 상기 MVC 동작점에 대한 뷰들에 포함된 비디오 데이터에 대한 최대 프레임 레이트를 기술하는 프레임 레이트 값, 상기 MVC 동작점의 렌더링의 대상이 되는 뷰들의 수에 대한 뷰-식별자 값들을 포함하고, 상기 뷰-식별자 값들 각각은 렌더링의 대상이 되는 상기 뷰들 중의 하나에 상응하고, 상기 동작점 기술자는 상기 MVC 동작점에 대한 디코딩될 뷰들의 수에 대한 뷰-식별자 값들을 포함하고, 상기 뷰-식별자 값들 각각은 디코딩될 상기 뷰들 중의 하나에 상응하며, 상기 동작점 기술자는 상기 MVC 동작점에 대한 상기 뷰들의 상기 비디오 데이터로부터 어셈블링된 비디오 스트림에 대한 프레임 레이트에 상응하는 시간적-식별자 값을 포함하는, 멀티미디어 데이터를 수신하는 장치.

**청구항 37**

제 36 항에 있어서,

상기 비디오 디코더가 상기 뷰들을 디코딩할 수 있는지를 결정하기 위해, 상기 디멀티플렉서는 프레임 레이트 값에 의해 지시되는 프레임 레이트에서 디코딩될 상기 뷰들의 수에 동등한 뷰들의 수를 상기 비디오 디코더가 디코딩할 수 있는지를 결정하도록 구성되는, 멀티미디어 데이터를 수신하는 장치.

**청구항 38**

삭제

**청구항 39**

제 35 항에 있어서,

목적지 디바이스에 의해 렌더링될 수 있는 지원된 뷰들의 수를 기술하는 지원된 뷰들의 수, 및 상기 목적지 디바이스에 의해 디스플레이될 수 있는 비디오 데이터의 프레임 레이트를 기술하는 프레임 레이트 값을 저장하도록 구성된 컴퓨터 판독가능 저장 매체를 더 포함하며,

상기 장치가 상기 MVC 동작점에 상응하는 상기 뷰들을 렌더링할 수 있는지를 결정하기 위해, 상기 디멀티플렉서는 상기 MVC 동작점에 상응하는 뷰들의 수를 상기 지원된 뷰들의 수와 비교하고, 상기 MVC 동작점에 상응하는 상기 뷰들의 상기 프레임 레이트를 상기 프레임 레이트 값과 비교하도록 구성되며,

상기 디멀티플렉서는, 상기 MVC 동작점에 상응하는 상기 뷰들의 수가 상기 지원된 뷰들의 수 이하인 경우 및 상기 MVC 동작점에 상응하는 상기 뷰들의 상기 프레임 레이트가 상기 프레임 레이트 값 이하인 경우, 상기 MVC 동작점에 상응하는 뷰들을 상기 비디오 디코더에 송신하도록 구성되는, 멀티미디어 데이터를 수신하는 장치.

**청구항 40**

제 39 항에 있어서,

상기 지원된 뷰들의 수는 상기 프레임 레이트 값에 역비례하는, 멀티미디어 데이터를 수신하는 장치.

**청구항 41**

제 35 항에 있어서,

상기 장치는,

집적 회로;

마이크로프로세서; 및

디멀티플렉서를 구비한 무선 통신 디바이스 중 적어도 하나를 포함하는, 멀티미디어 데이터를 수신하는 장치.

**청구항 42**

복수의 멀티뷰 비디오 코딩 (MVC) 기술자들을 수신하는 수단으로서, 각 MVC 기술자는 MPEG-2 (Motion Picture Experts Group) 시스템 표준 비트스트림의 MVC 동작점에 상응하고, 각 MVC 기술자는 상기 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, 및 상기 MVC 동작점을 사용하기 위해 상기 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값을 시그널링하는, 상기 복수의 MVC 기술자들을 수신하는 수단;

상기 MVC 기술자에 의해 시그널링된 디코딩 능력에 기초하여 장치의 비디오 디코더가 상기 MVC 동작점에 상응하는 뷰들의 수를 디코딩할 수 있는지를 각 MVC 기술자마다 결정하는 수단;

상기 MVC 기술자에 의해 시그널링된 렌더링 능력에 기초하여 상기 장치가 상기 MVC 동작점에 상응하는 상기 뷰들을 렌더링할 수 있는지를 각 MVC 기술자마다 결정하는 수단;

상기 상응하는 MVC 기술자에 기초하여 동작점을 선택하는 수단으로서, 상기 선택하는 것은 상기 비디오 디코더가 상기 선택된 동작점에 상응하는 뷰들을 디코딩 및 렌더링할 수 있다는 것을 결정하는 것을 포함하는, 상기 동작점을 선택하는 수단; 및

상기 선택된 MVC 동작점에 상응하는 상기 뷰들을 상기 장치의 상기 비디오 디코더에 송신하는 수단을 포함하는, 멀티미디어 데이터를 수신하는 장치.

**청구항 43**

제 42 항에 있어서,

각 MVC 기술자는 동작점 기술자를 포함하고, 상기 동작점 기술자는 상기 MVC 동작점에 대한 상기 뷰들에 포함된 비디오 데이터에 대한 최대 프레임 레이트를 기술하는 프레임 레이트 값, 상기 MVC 동작점의 렌더링의 대상이 되는 뷰들의 수에 대한 뷰-식별자 값들을 포함하고, 상기 뷰-식별자 값들 각각은 렌더링의 대상이 되는 상기 뷰들 중의 하나에 상응하고, 상기 동작점 기술자는 상기 MVC 동작점에 대한 디코딩될 뷰들의 수에 대한 뷰-식별자 값들을 포함하고, 상기 뷰-식별자 값들 각각은 디코딩될 상기 뷰들 중의 하나에 상응하며, 상기 동작점 기술자는 상기 MVC 동작점에 대한 상기 뷰들의 상기 비디오 데이터로부터 어셈블링된 비디오 스트림에 대한 프레임 레이트에 상응하는 시간적-식별자 값을 포함하는, 멀티미디어 데이터를 수신하는 장치.

**청구항 44**

제 43 항에 있어서,

상기 비디오 디코더가 뷰들을 디코딩할 수 있는지를 결정하는 수단은, 상기 비디오 디코더가 상기 프레임 레이트 값에 의해 지시되는 프레임 레이트에서 상기 뷰들의 수에 동등한 뷰들의 수를 디코딩할 수 있는지를 결정하는 수단을 포함하는, 멀티미디어 데이터를 수신하는 장치.

**청구항 45**

삭제

**청구항 46**

제 42 항에 있어서,

목적지 디바이스에 의해 렌더링될 수 있는 지원된 뷰들의 수를 기술하는 지원된 뷰들의 수 및 상기 목적지 디바이스에 의해 디스플레이될 수 있는 비디오 데이터의 프레임 레이트를 기술하는 프레임 레이트 값을 저장하는 수단을 더 포함하고,

상기 목적지 디바이스가 상기 MVC 동작점에 상응하는 상기 뷰들을 렌더링할 수 있는지를 결정하는 수단은,

상기 MVC 동작점에 상응하는 뷰들의 수를 상기 지원된 뷰들의 수와 비교하는 수단; 및

상기 MVC 동작점에 상응하는 상기 뷰들의 프레임 레이트를 상기 프레임 레이트 값과 비교하는 수단을 포함하고,

상기 MVC 동작점에 상응하는 상기 뷰들을 상기 비디오 디코더에 송신하는 수단은, 상기 MVC 동작점에 상응하는 상기 뷰들의 수가 상기 지원된 뷰들의 수 이하인 경우 및 상기 MVC 동작점에 상응하는 상기 뷰들의 상기 프레임 레이트가 상기 프레임 레이트 값 이하인 경우, 상기 MVC 동작점에 상응하는 뷰들을 상기 비디오 디코더에 송신하는 수단을 포함하는, 멀티미디어 데이터를 수신하는 장치.

**청구항 47**

실행될 때, 목적지 디바이스의 프로세서로 하여금,

MPEG-2 (Motion Picture Experts Group) 시스템 표준 비트스트림의 멀티뷰 비디오 코딩 (MVC) 동작점에 상응하는 복수의 MVC 기술자들을 수신하게 하는 명령어로서, 각 MVC 기술자는 상기 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, 및 상기 MVC 동작점을 사용하기 위해 상기 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값을 시그널링하는, 상기 복수의 MVC 기술자들을 수신하게 하는 명령어;

상기 MVC 기술자에 의해 시그널링된 디코딩 능력에 기초하여 상기 목적지 디바이스의 비디오 디코더가 상기 MVC 동작점에 상응하는 뷰들의 수를 디코딩할 수 있는지를 각 MVC 기술자마다 결정하게 하는 명령어;

상기 MVC 기술자에 의해 시그널링된 렌더링 능력에 기초하여 상기 목적지 디바이스가 상기 MVC 동작점에 상응하는 상기 뷰들을 렌더링할 수 있는지를 각 MVC 기술자마다 결정하게 하는 명령어;

상기 상응하는 MVC 기술자에 기초하여 동작점을 선택하게 하는 명령어로서, 상기 선택하는 것은 상기 비디오 디코더가 상기 선택된 동작점에 상응하는 뷰들을 디코딩 및 렌더링할 수 있다는 것을 결정하는 것을 포함하는, 상기 동작점을 선택하게 하는 명령어; 및

상기 선택된 MVC 동작점에 상응하는 상기 뷰들을 상기 목적지 디바이스의 상기 비디오 디코더에 송신하게 하는 명령어를 포함하는, 멀티미디어 데이터를 수신하기 위한 명령어들을 포함하는 컴퓨터 판독가능 저장 매체.

**청구항 48**

제 47 항에 있어서,

각 MVC 기술자는 동작점 기술자를 포함하고, 상기 동작점 기술자는 상기 MVC 동작점에 대한 상기 뷰들에 포함된 비디오 데이터에 대한 최대 프레임 레이트를 기술하는 프레임 레이트 값, 상기 MVC 동작점의 렌더링의 대상이 되는 뷰들의 수에 대한 뷰-식별자 값들을 포함하고, 상기 뷰-식별자 값들 각각은 렌더링의 대상이 되는 상기 뷰들 중의 하나에 상응하고, 상기 동작점 기술자는 상기 MVC 동작점에 대한 디코딩될 뷰들의 수에 대한 뷰-식별자 값들을 포함하고, 상기 뷰-식별자 값들 각각은 디코딩될 상기 뷰들 중의 하나에 상응하며, 상기 동작점 기술자는 상기 MVC 동작점에 대한 상기 뷰들의 상기 비디오 데이터로부터 어셈블링된 비디오 스트림에 대한 프레임 레이트에 상응하는 시간적-식별자 값을 포함하는, 멀티미디어 데이터를 수신하기 위한 명령어들을 포함하는 컴퓨터 판독가능 저장 매체.

**청구항 49**

제 48 항에 있어서,

상기 프로세서로 하여금 상기 비디오 디코더가 뷰들을 디코딩할 수 있는지를 결정하게 하는 명령어는, 상기 프로세서로 하여금 상기 비디오 디코더가 상기 프레임 레이트 값에 의해 지시되는 프레임 레이트에서 상기 뷰들의 수에 동등한 뷰들의 수를 디코딩할 수 있는지를 결정하게 하는 명령어를 포함하는, 멀티미디어 데이터를 수신하기 위한 명령어들을 포함하는 컴퓨터 판독가능 저장 매체.

**청구항 50**

삭제

**청구항 51**

제 47 항에 있어서,

상기 목적지 디바이스는 상기 목적지 디바이스에 의해 렌더링될 수 있는 지원된 뷰들의 수를 기술하는 지원된 뷰들의 수 및 상기 목적지 디바이스에 의해 디스플레이될 수 있는 비디오 데이터의 프레임 레이트를 기술하는

프레임 레이트 값으로 구성되며,

상기 프로세서로 하여금, 상기 목적지 디바이스가 상기 MVC 동작점에 상응하는 상기 뷰들을 렌더링할 수 있는지를 결정하게 하는 명령어는, 상기 프로세서로 하여금,

상기 MVC 동작점에 상응하는 뷰들의 수를 상기 지원된 뷰들의 수와 비교하게 하는 명령어; 및

상기 MVC 동작점에 상응하는 상기 뷰들의 프레임 레이트를 상기 프레임 레이트 값과 비교하게 하는 명령어를 포함하고,

상기 프로세서로 하여금, 상기 MVC 동작점에 상응하는 상기 뷰들을 상기 비디오 디코더에 송신하게 하는 명령어는, 상기 프로세서로 하여금, 상기 MVC 동작점에 상응하는 상기 뷰들의 수가 상기 지원된 뷰들의 수 이하인 경우 및 상기 MVC 동작점에 상응하는 상기 뷰들의 상기 프레임 레이트가 상기 프레임 레이트 값 이하인 경우, 상기 MVC 동작점에 상응하는 뷰들을 상기 비디오 디코더에 송신하게 하는 명령어를 포함하는, 멀티미디어 데이터를 수신하기 위한 명령어들을 포함하는 컴퓨터 판독가능 저장 매체.

## 명세서

### 기술분야

[0001] 본 개시물은 인코딩된 비디오 데이터의 전송에 관한 것이다.

### 배경기술

[0002] 본 출원은 미국가출원들인 2009년 8월 7일자로 출원된 제61/232,272호, 2009년 10월 5일자로 출원된 제61/248,738호, 및 2009년 12월 4일자로 출원된 제61/266,861호를 우선권 주장하며, 그것들 각각의 전체 내용들은 그것들 개개의 전부가 참고문헌으로 여기에 통합된다.

### 배경

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인휴대 정보단말기들 (PDA들), 랩톱 또는 데스크톱 컴퓨터들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 라디오 전화기들, 비디오 원격회의 디바이스들 등을 포함하는 넓은 범위의 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263 또는 ITU-T H.264/MPEG-4, 파트 10, 고급 비디오 코딩 (AVC) 에 의해 규정된 표준들 및 이러한 표준들의 확장물들에 기재된 것들과 같은 비디오 압축 기법들을 구현하여, 디지털 비디오 정보를 더 효율적으로 전송하고 수신한다.

[0005] 비디오 압축 기법들은 공간적 예측 및/또는 시간적 예측을 수행하여 비디오 시퀀스들에 내재하는 리던던시를 감소하거나 제거한다. 블록 기반 비디오 코딩의 경우, 비디오 프레임 또는 슬라이스는 매크로블록들로 구획될 수 있다. 각각의 매크로블록은 더 구획될 수도 있다. 인트라 코딩된 (I) 프레임 또는 슬라이스 내의 매크로블록들은 이웃 매크로블록들에 관한 공간적 예측을 이용하여 인코딩된다. 인터 코딩된 (P 또는 B) 프레임 또는 슬라이스 내의 매크로블록들은 동일한 프레임 또는 슬라이스 내의 이웃 매크로블록들에 관한 공간적 예측 또는 다른 참조 프레임들에 관한 시간적 예측을 이용할 수 있다.

[0006] 비디오 데이터가 인코딩된 후, 비디오 데이터는 전송 또는 저장을 위해 멀티플렉서에 의해 패키징될 수 있다. MPEG-2는 많은 비디오 인코딩 표준들에 대한 전송 레벨을 규정하는 "시스템들(Systems)" 섹션을 포함한다.

MPEG-2 전송 레벨 시스템들은 MPEG-2 비디오 인코더들, 또는 다른 비디오 인코딩 표준들을 따른 기타 비디오 인코더들에 의해 사용될 수 있다. 예를 들어, MPEG-4는 MPEG-2와는 다른 인코딩 및 디코딩 방법론들을 규정하지만, MPEG-4 표준의 기법들을 구현하는 비디오 인코더들은 여전히 MPEG-2 전송 레벨 방법론들을 활용할 수 있다.

[0007] 일반적으로, 이 개시물에서의 "MPEG-2 시스템들"에 대한 언급은 MPEG-2에 의해 규정된 비디오 데이터의 전송 레벨을 말하는 것이다. MPEG-2에 의해 규정된 전송 레벨 역시 이 개시물에서는 "MPEG-2 전송 스트림" 또는 간단히 "전송 스트림"이라고 한다. 유사하게, MPEG-2 시스템들의 전송 레벨은 또한 프로그램 스트림들을 포함한다. 전송 스트림 및 프로그램 스트림들은 대체로 유사한 데이터를 전달하는 상이한 포맷들을 포함하는 데, 전송 스트림은 오디오 및 비디오 데이터 양쪽 모두를 구비하는 하나 이상의 "프로그램들"을 포함하는 반면, 프로그램 스트림들은 오디오 및 비디오 데이터 양쪽 모두를 구비하는 하나의 프로그램을 포함한다.

[0008] H.264/AVC에 기초하여 새로운 비디오 코딩 표준들을 개발하려는 노력들이 있어 왔다. 하나의 이러한 표준은 스케일러블 비디오 코딩 (SVC) 표준인데, 이는 H.264/AVC의 스케일러블 확장물이다. 다른 표준은 멀티-뷰 비디오 코딩 (MVC) 인데, 이는 H.264/AVC의 멀티뷰 확장물이다. MPEG-2 시스템들의 사양서는 압축된 멀티미디어 (비디오 및 오디오) 데이터 스트림들이 디지털 전송 또는 저장에 적합한 단일 데이터 스트림을 형성하도록 다른 데이터와 함께 다중화될 수 있는 방법을 기술한다. MPEG-2 시스템들의 최근의 사양은 "Information Technology-Generic Coding of Moving Pictures and Associated Audio: Systems, Recommendation H.222.0; International Organisation for Standardisation, ISO/IEC JTC1/SC29/WG11; Coding of Moving Pictures and Associated Audio" 2006년 5월에 명시되어 있다. MPEG에서 최근에 디자인한 MPEG-2 시스템들에 대한 MVC의 전송 표준 및 이 사양서의 최근 버전은 "Study of ISO/IEC 13818-1:2007/FPDAM4 Transport of MVC", *MPEG doc. N10572, MPEG of ISO/IEC JTC1/SC29/WG11*, 미국, 하와이주, 마우이, 2009년 4월이다.

**발명의 내용**

**과제의 해결 수단**

[0009] 대체로, 이 개시물은 MPEG-2 (Motion Picture Experts Group) 시스템들의 멀티뷰 비디오 코딩을 개선하기 위한 기법들을 설명한다. 특히, 이 개시물의 기법들은 MPEG-2 시스템 비트스트림의 동작점에 대한 데이터 구조에 관한 것이고, 이 데이터 구조는 수신 디바이스에 대한 렌더링 능력, 수신 디바이스에 대한 디코딩 능력, 및 일부 예들에서는 동작점에 대한 비트레이트를 시그널링한다. 이 데이터 구조는 MPEG-2 시스템 비트스트림에 포함되는 동작점 기술자(operation point descriptor)에 해당할 수 있다.

[0010] 동작점의 비디오 데이터를 적절히 디코딩하고 디스플레이하기 위해, 수신 디바이스는 데이터 구조에서 시그널링되는 렌더링 능력 및 디코딩 능력에 의해 기술되는 특성들을 충족해야만 한다. MPEG-2 시스템 비트스트림들은 프로그램의 다양한 뷰들에 상응하는 복수의 동작점들을 구비할 수 있다. 프로그램을 위해 상이한 동작점들을 사용하는 것은 다양한 클라이언트 디바이스들이 적응을 수행하는 것을 가능하게 한다. 다시 말하면, 상이한 렌더링 및 디코딩 능력들을 갖는 클라이언트 디바이스들은 2차원 또는 3차원 비디오 데이터를 디스플레이하기 위해 동일한 프로그램으로부터 뷰들을 추출할 수 있다. 클라이언트 디바이스들은 다양한 대역폭 능력들의 전송 미디어에 적응하는 가변 비트레이트들의 데이터를 검색하도록 서버 디바이스와 협상할 수도 있다.

[0011] 하나의 예에서, 방법은 소스 디바이스로, MPEG-2 시스템 표준 비트스트림의 멀티뷰 비디오 코딩 (MVC) 동작점에 상응하는 데이터 구조를 구축하는 단계로서, 데이터 구조는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하고, 데이터 구조는 비트스트림의 일부로서 포함되는, 상기 데이터 구조를 구축하는 단계, 및 이 데이터 구조를 포함하는 비트스트림을 출력하는 단계를 구비한다.

[0012] 다른 예에서, 장치는 MPEG-2 시스템 표준 비트스트림의 MVC 동작점에 상응하는 데이터 구조를 구축하는 멀티플렉서로서, 데이터 구조는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하고, 데이터 구조를 비트스트림의 일부로서 포함하는, 상기 멀티플렉서, 및 이 데이터 구조를 포함하는 비트스트림을 출력하는 출력 인터페이스를 구비한다.

[0013] 다른 예에서, 장치는 MPEG-2 시스템 표준 비트스트림의 MVC 동작점에 상응하는 데이터 구조를 구축하기 위한 수단으로서, 데이터 구조는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하고, 데이터 구조는 비트스트림의 일부로서 포함되는, 상기 데이터 구조를 구축하기 위한 수단, 및 이 데이터 구조를 포함하는 비트스트림을 출력하기 위한 수단을 구비한다.

[0014] 다른 예에서, 컴퓨터 판독가능 저장 매체는 소스 디바이스의 프로세서로 하여금, MPEG-2 시스템 표준 비트스트림의 MVC 동작점에 상응하는 데이터 구조를 구축하게 하는 명령어로서, 이 데이터 구조는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하고, 이 데이터 구조는 비트스트림의 일부로서 포함되는, 상기 데이터

구조를 구축하게 하는 명령어, 및 출력 인터페이스로 하여금 이 데이터 구조를 포함하는 비트스트림을 출력하게 하는 명령어를 포함한다.

[0015] 다른 예에서, 방법은 목적지 디바이스로, MPEG-2 (Motion Picture Experts Group) 시스템 표준 비트스트림의 MVC 동작점에 상응하는 데이터 구조를 수신하는 단계로서, 데이터 구조는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하는, 상기 데이터 구조를 수신하는 단계; 데이터 구조에 의해 시그널링된 디코딩 능력에 기초하여 목적지 디바이스의 비디오 디코더가 MVC 동작점에 상응하는 뷰들을 디코딩할 수 있는지를 결정하는 단계; 데이터 구조에 의해 시그널링된 렌더링 능력에 기초하여 목적지 디바이스가 MVC 동작점에 상응하는 뷰들을 렌더링할 수 있는지를 결정하는 단계; 및 목적지 디바이스의 비디오 디코더가 MVC 동작점에 상응하는 뷰들을 디코딩하고 렌더링할 수 있다고 결정되는 경우 MVC 동작점에 상응하는 뷰들을 목적지 디바이스의 비디오 디코더에 송신하는 단계를 포함한다.

[0016] 다른 예에서, 장치는 MPEG-2 시스템 표준 비트스트림의 멀티뷰 비디오 코딩 (MVC) 동작점에 상응하는 데이터 구조를 수신하도록 구성된 입력 인터페이스로서, 데이터 구조는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하는, 상기 입력 인터페이스; 비디오 데이터를 디코딩하도록 구성된 비디오 디코더; 및 데이터 구조에 의해 시그널링된 디코딩 능력에 기초하여 비디오 디코더가 MVC 동작점에 상응하는 뷰들을 디코딩할 수 있는지를 결정하고, 데이터 구조에 의해 시그널링된 렌더링 능력에 기초하여 장치가 MVC 동작점에 상응하는 뷰들을 렌더링할 수 있는지를 결정하고, 비디오 디코더가 MVC 동작점에 상응하는 뷰들을 디코딩하고 렌더링할 수 있다고 결정되는 경우 MVC 동작점에 상응하는 뷰들을 비디오 디코더에 송신하도록 구성된 디멀티플렉서를 포함한다.

[0017] 다른 예에서, 장치는 MPEG-2 시스템 표준 비트스트림의 MVC 동작점에 상응하는 데이터 구조를 수신하기 위한 수단으로서, 데이터 구조는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하는, 상기 데이터 구조를 수신하기 위한 수단; 데이터 구조에 의해 시그널링된 디코딩 능력에 기초하여 장치의 비디오 디코더가 MVC 동작점에 상응하는 뷰들을 디코딩할 수 있는지를 결정하기 위한 수단; 데이터 구조에 의해 시그널링된 렌더링 능력에 기초하여 장치가 MVC 동작점에 상응하는 뷰들을 렌더링할 수 있는지를 결정하기 위한 수단; 및 장치의 비디오 디코더가 MVC 동작점에 상응하는 뷰들을 디코딩하고 렌더링할 수 있다고 결정되는 경우 MVC 동작점에 상응하는 뷰들을 장치의 비디오 디코더에 송신하기 위한 수단을 포함한다.

[0018] 다른 예에서, 컴퓨터-판독가능 저장 매체는, 목적지 디바이스의 프로세서로 하여금, MPEG-2 시스템 표준 비트스트림의 MVC 동작점에 상응하는 데이터 구조를 수신하게 하는 명령어로서 이 데이터 구조는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 렌더링 능력을 기술하는 렌더링 능력 값, MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링하는, 상기 데이터 구조를 수신하게 하는 명령어, 데이터 구조에 의해 시그널링된 디코딩 능력에 기초하여 목적지 디바이스의 비디오 디코더가 MVC 동작점에 상응하는 뷰들을 디코딩할 수 있는지를 결정하게 하는 명령어, 데이터 구조에 의해 시그널링된 렌더링 능력에 기초하여 목적지 디바이스가 MVC 동작점에 상응하는 뷰들을 렌더링할 수 있는지를 결정하게 하는 명령어; 및 목적지 디바이스의 비디오 디코더가 MVC 동작점에 상응하는 뷰들을 디코딩하고 렌더링할 수 있다고 결정되는 경우 MVC 동작점에 상응하는 뷰들을 목적지 디바이스의 비디오 디코더에 송신하게 하는 명령어를 포함한다.

[0019] 하나 이상의 예들의 세부사항들은 첨부 도면들 및 다음의 설명에서 언급된다. 다른 특징들, 목적들, 및 이점들은 상세한 설명 및 도면들로부터, 그리고 청구항들로부터 명확하게 될 것이다.

**도면의 간단한 설명**

[0020] 도 1은 오디오/비디오 (A/V) 소스 디바이스가 오디오 및 비디오 데이터를 A/V 목적지 디바이스에 전송하는 예시적인 시스템을 도시하는 블록도이다.

도 2는 본 개시물에 따른 멀티플렉서의 구성요소들의 예시적인 배치구성을 도시하는 블록도이다.



도 3은 본 개시물에 따른 프로그램 지정 정보 테이블들의 예시적인 세트를 도시하는 블록도이다.

도 4 내지 도 6은 동작점 기술자에 포함될 수 있는 데이터의 세트들의 다양한 예를 도시하는 개념도들이다.

도 7은 예시적인 MVC 예측 패턴을 도시하는 개념도이다.

도 8은 동작점의 특성들을 시그널링하는 데이터 구조를 이용하기 위한 예시적인 방법을 도시하는 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

- [0021] 이 개시물의 기법들은 대체로 MPEG-2 (Motion Picture Experts Group) 시스템들, 다시 말하면, 전송 레벨 세부 사항들에 관하여 MPEG-2를 따르는 시스템들에서의 멀티뷰 비디오 코딩 (MVC) 의 개선을 위한 것이다. MPEG-4는, 예를 들어, 비디오 인코딩을 위한 표준들을 제공하지만, 일반적으로 MPEG-4 표준을 따르는 비디오 인코더들이 MPEG-2 전송 레벨 시스템들을 활용할 것이라고 가정한다. 따라서, 이 개시물의 기법들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 또는 MPEG-2 전송 스트림들 및/또는 프로그램 스트림들 ("programme streams"라고 기재하기도 함) 을 활용하는 다른 어떤 비디오 인코딩 표준이라도 따르는 비디오 인코더들에 적용가능하다.
- [0022] 특히, 이 개시물의 기법들은 MPEG-2 전송 스트림들 및 프로그램 스트림들에 대한 전송 레벨에서 구문(syntax) 요소들을 수정할 수 있다. 예를 들어, 이 개시물의 기법들은 동작점의 특징들을 기술하기 위해 전송 스트림으로 전송되는 기술자(descriptor)를 구비한다. 서버 디바이스는, 예를 들어, MPEG-2 전송 계층 비트스트림에서 다양한 동작점들을 제공할 수 있는데, 동작점들 각각은 멀티뷰 비디오 코딩 비디오 데이터의 특정 뷰들의 개별 서브세트에 해당한다. 다시 말하면, 동작점은 일반적으로 비트스트림의 뷰들의 서브세트에 상응한다. 일부 예들에서, 동작점 각각의 뷰는 동일한 프레임 레이트의 비디오 데이터를 포함한다.
- [0023] 목적지 디바이스는 디코딩할 그리고 궁극적으로는 사용자에게 나타낼 (예컨대, 디스플레이할) 동작점들 중의 하나를 비트스트림에 포함된 동작점 기술자들을 사용하여 선택할 수 있다. 모든 뷰들에 대한 데이터를 수신시에 비디오 디코더에 전해주는 대신, 목적지 디바이스는 선택된 동작점의 뷰들만을 비디오 디코더에 송신할 수 있다. 이런 방식으로, 목적지 디바이스는 디코딩되지 않을 뷰들에 대한 데이터를 폐기할 수 있다. 목적지 디바이스는 비트스트림에 대한 동작점들 중 최고 품질을 지원하는 것에 기초하여 동작점을 선택할 수 있다.
- [0024] 서버 디바이스는 단일 전송 스트림 또는 프로그램 스트림에서 복수의 서브-비트스트림들 (이것들 각각은 동작점에 상응할 수 있다) 을 송신할 수 있다. 여러 섹션들에서 이 개시물이 "전송 스트림" 또는 "프로그램 스트림"을 각각 따로 언급할 수는 있지만, 이 개시물의 기법들은 대체로 MPEG-2 전송 스트림들 및 프로그램 스트림들의 어느 하나 또는 양쪽 모두에 적용될 수 있다는 것이 이해되어야 한다. 대체로, 이 개시물은 이 개시물의 기법들을 수행하기 위한 예시적인 데이터 구조로서 기술자들의 사용을 설명한다. 기술자들은 스트림의 기능을 확장하는데 사용된다. 이 개시물의 기술자들은 이 개시물의 기법들을 구현하기 위해 전송 스트림들 및 프로그램 스트림들 양자에 의해 사용될 수 있다. 이 개시물은 동작점에 대한 렌더링 능력 값, 동작점에 대한 디코딩 능력 값, 및 동작점에 대한 비트레이트 값을 시그널링하는데 사용될 수 있는 예시적인 데이터 구조로서 기술자들에 주로 집중하고 있지만, 다른 데이터 구조들이 이들 기법들을 수행하기 위해 사용될 수도 있다는 것은 이해되어야 한다.
- [0025] 이 개시물의 기법들에 따라서, 소스 디바이스 (20) 는 동작점의 특성들을 기술하는 동작점 기술자를 구축할 수 있다. 이 특성들은 예를 들어, 어떤 뷰들이 동작점에 포함되는지와 이 동작점의 뷰들에 대한 프레임 레이트들을 포함할 수 있다. 동작점 기술자는 동작점을 수신하고 디코딩하기 위해 비디오 디코더에 의해 지원되어야 하는 렌더링 능력, 동작점을 수신하고 디코딩하기 위하여 비디오 디코더에 의해 지원되어야 하는 디코딩 능력, 및 동작점에 대한 비트레이트를 명시할 수 있다.
- [0026] 이 개시물의 기법들은 일반적으로 마치 동작점이 전송 스트림의 프로그램 맵 테이블 또는 프로그램 스트림의 프로그램 스트림 맵에 의해 시그널링되는 자신 소유의 프로그램인 것처럼 각각의 동작점을 나타낼 수 있다. 다르게는, 프로그램이 다수의 동작점들을 포함하는 경우, 이 개시물의 기법들은 동작점들이 동작점 기술자들에서 리어셈블링 (reassemble) 된 방법에 관한 정보를 제공한다. 동작점 기술자들은 동작점들의 의존도들을 추가로 시그널링할 수 있고, 이는 비트들을 절약할 수 있다.
- [0027] 도 1은 오디오/비디오 (A/V) 소스 디바이스 (20) 가 오디오 및 비디오 데이터를 A/V 목적지 디바이스 (40) 에 전송하는 예시적인 시스템 (10) 을 도시하는 블록도이다. 도 1의 시스템 (10) 은 비디오 원격회의 시스템, 서버/클라이언트 시스템, 브로드캐스트/수신기 시스템, 또는 비디오 데이터가 소스 디바이스, 이를테면 A/V 소

스 디바이스 (20) 로부터 목적지 디바이스, 이를테면 A/V 목적지 디바이스 (40) 로 송신되는 임의의 다른 시스템에 해당할 수 있다. 일부 예들에서, A/V 소스 디바이스 (20) 및 A/V 목적지 디바이스 (40) 는 양방향 정보 교환을 수행할 수 있다. 다시 말하면, A/V 소스 디바이스 (20) 및 A/V 목적지 디바이스 (40) 는 오디오 및 비디오 데이터를 코딩하고 디코딩하는 (그리고 전송하고 수신하는) 양 쪽 모두를 할 수 있다. 일부 예들에서, 오디오 인코더 (26) 는 보코더라고도 하는 음성 인코더를 포함할 수 있다.

[0028] A/V 소스 디바이스 (20) 는, 도 1의 예에서, 오디오 소스 (22) 및 비디오 소스 (24) 를 포함한다. 오디오 소스 (22) 는 예를 들어 오디오 인코더 (26) 에 의해 인코딩될 포착된 오디오 데이터를 나타내는 전기적 신호들을 생성하는 마이크론을 포함할 수 있다. 다르게는, 오디오 소스 (22) 는 이전에 기록된 오디오 데이터를 저장하는 저장 매체, 컴퓨터화 신시사이저와 같은 오디오 데이터 발생기, 또는 임의의 다른 오디오 데이터 소스를 포함할 수 있다. 비디오 소스 (24) 는 비디오 인코더 (28) 에 의해 인코딩될 비디오 데이터를 생성하는 비디오 카메라, 이전에 기록된 비디오 데이터로 인코딩된 저장 매체, 비디오 데이터 발생 유닛, 또는 임의의 다른 비디오 데이터 소스를 포함할 수 있다.

[0029] 가공되지 않은(raw) 오디오 및 비디오 데이터는 아날로그 또는 디지털 데이터를 포함할 수 있다. 아날로그 데이터는 오디오 인코더 (26) 및/또는 비디오 인코더 (28) 에 의해 인코딩되기 전에 디지털화될 수 있다. 오디오 소스 (22) 는 송화(speaking) 참가자가 말을 하는 동안 오디오 데이터를 송화 참가자로부터 얻을 수 있고, 비디오 소스 (24) 는 동시에 송화 참가자의 비디오 데이터를 얻을 수 있다. 다른 예들에서, 오디오 소스 (22) 는 저장된 오디오 데이터를 포함하는 컴퓨터-판독가능 저장 매체를 포함할 수 있고, 비디오 소스 (24) 는 저장된 비디오 데이터를 포함하는 컴퓨터-판독가능 저장 매체를 포함할 수 있다. 이런 방식으로, 이 개시물에서 설명된 기법들은 라이브, 스트리밍, 실시간 오디오 및 비디오 데이터 또는 보관된 미리 기록된 오디오 및 비디오 데이터에 적용될 수 있다.

[0030] 비디오 프레임들에 상응하는 오디오 프레임들은 일반적으로 비디오 프레임들 내에 들어 있는 비디오 소스 (24) 에 의해 포착된 비디오 데이터와 동시에 오디오 소스 (22) 에 의해 포착된 오디오 데이터를 담고 있는 오디오 프레임들이다. 예를 들어, 송화 참가자가 일반적으로 송화에 의해 오디오 데이터를 생성하는 동안, 오디오 소스 (22) 는 오디오 데이터를 포착하고, 비디오 소스 (24) 는 동시에 다시 말하면, 오디오 소스 (22) 가 오디오 데이터를 포착하는 동안 동일한 시간에 송화 참가자의 비디오 데이터를 포착한다. 그래서, 오디오 프레임이 하나 이상의 특정 비디오 프레임들에 일시적으로 상응할 수 있다. 따라서, 오디오 프레임이 비디오 프레임에 상응한다는 것은 일반적으로 오디오 데이터 및 비디오 데이터가 동일한 시간에 포착되었고 오디오 프레임 및 비디오 프레임이 동일한 시간에 포착된 오디오 데이터 및 비디오 데이터를 각각 포함하는 상황에 해당한다.

[0031] 일부 예들에서, 오디오 인코더 (26) 는 인코딩된 오디오 프레임에 대한 오디오 데이터가 기록되었던 시간을 나타내는 타임스탬프를 각각의 인코딩된 오디오 프레임 내에 인코딩할 수 있고, 마찬가지로, 비디오 인코더 (28) 는 인코딩된 비디오 프레임에 대한 비디오 데이터가 기록되었던 시간을 나타내는 각각의 인코딩된 비디오 프레임에서의 타임스탬프를 인코딩할 수 있다. 이런 예들에서, 오디오 프레임이 비디오 프레임에 상응한다는 것은 오디오 프레임이 타임스탬프를 포함한다는 것과 비디오 프레임이 동일한 타임스탬프를 포함한다는 것일 수 있다. A/V 소스 디바이스 (20) 는 오디오 인코더 (26) 및/또는 비디오 인코더 (28) 가 타임스탬프들을 발생할 수 있게 하거나 또는 오디오 소스 (22) 및 비디오 소스 (24) 가 오디오 및 비디오 데이터를 각각 타임스탬프와 관련시키기 위해 사용할 수 있는 내부 클럭을 구비할 수 있다.

[0032] 일부 예들에서, 오디오 소스 (22) 는 오디오 데이터가 기록되었던 시간에 상응하는 데이터를 오디오 인코더 (26) 에 송신할 수 있고, 비디오 소스 (24) 는 비디오 데이터가 기록되었던 시간에 상응하는 데이터를 비디오 인코더 (28) 에 송신할 수 있다. 일부 예들에서, 오디오 인코더 (26) 는 인코딩되는 오디오 데이터 내에 인코딩된 오디오 데이터의 상대 시간적 순서를 나타내지만 오디오 데이터가 기록되었던 절대 시간을 반드시 나타낼 필요는 없는 시퀀스 식별자를 인코딩할 수 있고, 마찬가지로, 비디오 인코더 (28) 는 인코딩된 비디오 데이터의 상대 시간적 순서를 나타내는데 시퀀스 식별자들을 사용할 수도 있다. 마찬가지로, 일부 예들에서, 시퀀스 식별자가 타임스탬프와 매핑될 수 있거나 또는 그렇지 않으면 상관될 수 있다.

[0033] 이 개시물의 기법들은 대체로 인코딩된 멀티미디어 (예컨대, 오디오 및 비디오) 데이터의 전송과, 전송되는 멀티미디어 데이터의 수신 및 그 다음의 해석 및 디코딩을 위한 것이다. 이 개시물의 기법들은 특히 멀티뷰 비디오 코딩 (MVC) 데이터, 다시 말하면, 복수의 뷰들을 포함하는 비디오 데이터의 전송에 적용가능하다. 도 1의 예에서 보인 바와 같이, 비디오 소스 (24) 는 한 장면의 복수의 뷰들을 비디오 인코더 (28) 에 제공할



수 있다. MVC는 3차원 디스플레이, 이를테면 입체 또는 자동입체 (무안경식) 3차원 디스플레이에 의해 사용될 3차원 비디오 데이터를 발생하는데 유용할 수 있다.

[0034] A/V 소스 디바이스 (20) 는 "서비스"를 A/V 목적지 디바이스 (40) 에 제공할 수 있다. 서비스는 일반적으로 MVC 데이터의 이용가능한 뷰들의 서브세트에 해당한다. 예를 들어, MVC 데이터는 0 내지 7의 순서의 8개의 뷰들에 이용가능할 수 있다. 하나의 서비스는 2개의 뷰들을 갖는 스테레오 비디오에 상응할 수 있으며, 다른 서비스는 4개의 뷰들에 상응할 수 있고, 또 다른 서비스는 모두 8개의 뷰들에 상응할 수 있다. 일반적으로, 서비스는 이용가능한 뷰들의 임의의 조합 (다시 말하면, 임의의 서브세트) 에도 상응한다. 서비스는 이용가능한 뷰들뿐 아니라 오디오 데이터의 조합에 상응할 수도 있다. 동작점은 서비스에 상응할 수 있어서, A/V 소스 디바이스 (20) 는 A/V 소스 디바이스 (20) 에 의해 제공되는 각각의 서비스에 동작점 기술자를 추가로 제공할 수 있다.

[0035] A/V 소스 디바이스 (20) 는 이 개시물의 기법들을 따라서 뷰들의 서브세트에 상응하는 서비스들을 제공할 수 있다. 일반적으로, 뷰는 "view\_id"라고도 하는 뷰 식별자에 의해 나타내어진다. 뷰 식별자들은 일반적으로 뷰를 식별하는데 사용될 수 있는 구문 (syntax) 요소들을 포함한다. MVC 인코더는 뷰가 인코딩될 때 뷰의 view\_id를 제공한다. view\_id는 뷰 간(inter-view) 예측을 위해 MVC 디코더에 의해 또는 다른 목적을 위해, 예컨대, 렌더링을 위해 다른 유닛들에 의해 사용될 수 있다.

[0036] 뷰 간 예측은 공통 시간적 위치에 있는 하나 이상의 프레임들을 참조하여 프레임의 MVC 비디오 데이터를 상이한 뷰들의 인코딩된 프레임으로서 인코딩하는 기법이다. 도 7은 아래에서 매우 상세히 논의될 것으로, 뷰 간 예측을 위한 예시적인 코딩 체계(scheme)를 제공한다. 일반적으로, MVC 비디오 데이터의 인코딩되는 프레임은 공간적으로, 시간적으로, 및/또는 공통 시간적 위치에서의 다른 뷰들의 프레임들을 참조하여 예측 인코딩될 수 있다. 따라서, 다른 뷰들의 예측에 기준이 되는 참조 뷰들은, 대개 참조 뷰들이 기준으로서 역할을 하기 위해 다른 뷰들에 앞서 디코딩되어서, 이들 디코딩된 뷰들이 관련있는 뷰들을 디코딩할 때 기준으로서 사용될 수 있다. 디코딩 순서는 view\_ids의 순서에 상응할 필요는 없다. 그러므로, 뷰들의 디코딩 순서 뷰 순서를 이용하여 기술된다. 뷰 순서 색인들은 액세스 단위에서 상응하는 뷰 성분들의 디코딩 순서를 나타내는 색인들이다.

[0037] 데이터 (오디오든 비디오든) 각각의 개별 스트림은 기본 스트림(elementary stream)이라고 한다. 기본 스트림은 프로그램의 단일의 디지털 코딩된 (어쩌면 압축된) 성분이다. 예를 들어, 프로그램의 코딩된 비디오 또는 오디오 부분은 기본 스트림일 수 있다. 기본 스트림은 프로그램 스트림 또는 전송 스트림으로 다중화되기 전에 패킷화된 기본 스트림 (PES) 으로 변환될 것이다. 동일한 프로그램 내에서, 스트림 ID는 하나의 기본 스트림에 속한 PES-패킷들을 다른 것으로부터 구별하는데 사용된다. 기본 스트림의 기본 데이터 단위는 패킷화된 기본 스트림 (PES) 패킷이다. 그래서, MVC 비디오 데이터의 각 뷰는 개별 기본 스트림들에 상응한다. 마찬가지로, 오디오 데이터는 하나 이상의 개별 기본 스트림들에 상응한다.

[0038] MVC 코딩된 비디오 시퀀스는 수 개의 서브-비트스트림들로 분할될 수 있는데, 서브-비트스트림들 각각은 기본 스트림이다. 각각의 서브-비트스트림은 MVC view\_id 서브세트를 이용하여 식별될 수 있다. 각각의 MVC view\_id 서브세트의 개념에 기초하여, MVC 비디오 서브-비트스트림이 정의된다. MVC 비디오 서브-비트스트림은 MVC view\_id 서브세트에 열거된 뷰들의 NAL 단위들을 담고 있다. 프로그램 스트림은 일반적으로 기본 스트림들로부터 유래되는 NAL 단위들만을 담고 있다. 또한 어떤 두 개의 기본 스트림들이라도 동일한 뷰를 담을 수 없도록 디자인된다.

[0039] 도 1의 예에서, 멀티플렉서 (30) 는 비디오 데이터를 포함하는 기본 스트림들을 비디오 인코더 (28) 로부터 그리고 오디오 데이터를 포함하는 기본 스트림들을 오디오 인코더 (26) 로부터 수신한다. 일부 예들에서, 비디오 인코더 (28) 및 오디오 인코더 (26) 는 각각이 인코딩된 데이터로부터 PES 패킷들을 형성하는 패킷화기들을 구비할 수 있다. 다른 예들에서, 비디오 인코더 (28) 및 오디오 인코더 (26) 는 각각이 인코딩된 데이터로부터 PES 패킷들을 형성하는 개별 패킷화기들과 인터페이싱할 수 있다. 또 다른 예들에서, 멀티플렉서 (30) 는 인코딩된 오디오 및 비디오 데이터로부터 PES 패킷들을 형성하는 패킷화기들을 구비할 수 있다.

[0040] "프로그램"은 이 개시물에서 사용된 바대로, A/V 소스 디바이스 (20) 의 서비스에 의해 전달되는 오디오 데이터 및 비디오 데이터, 예컨대 오디오 기본 스트림 및 이용가능한 뷰들의 서브세트의 조합을 포함할 수 있다. 각각의 PES 패킷은 PES 패킷이 속한 기본 스트림을 식별하는 stream\_id 를 구비한다. 멀티플렉서 (30) 는 기본 스트림들을, 구성하는 프로그램 스트림들 또는 전송 스트림들로 어셈블링하는 것을 담당한다. 프로그램 스트림 및 전송 스트림은 상이한 응용들을 겨냥하는 2개의 대체가능한 다중화물들이다.

- [0041] 대체로, 프로그램 스트림은 하나의 프로그램에 대한 데이터를 구비할 수 있는 반면, 전송 스트림은 하나 이상의 프로그램들에 대한 데이터를 구비할 수 있다. 멀티플렉서 (30)는 프로그램 스트림 및 전송 스트림의 어느 한쪽 또는 양쪽 모두를, 제공되는 서비스, 스트림이 통과할 매체, 송신될 프로그램들의 수, 또는 기타 고려사항들에 기초하여 인코딩할 수 있다. 예를 들어, 비디오 데이터가 저장 매체에 인코딩될 경우, 멀티플렉서 (30)는 프로그램 스트림을 형성할 가능성이 더 많을 수 있는 반면, 비디오 데이터가 네트워크를 통해 스트리밍되거나, 브로드캐스트되거나, 또는 화상 통화물의 부분으로서 송신될 경우, 멀티플렉서 (30)는 전송 스트림을 사용할 가능성이 더 많을 수 있다.
- [0042] 멀티플렉서 (30)는 디지털 저장 서비스로부터의 단일 프로그램의 저장 및 디스플레이를 위해 프로그램 스트림을 사용하는 것을 지지하는 쪽으로 치우쳐 있을 수 있다. 프로그램 스트림은 에러 없는 환경들 또는 직면하는 에러들에 덜 민감한 환경들에서 사용하는 것을 의도하는데, 프로그램 스트림들이 에러들에 더 민감할 수 있기 때문이다. 프로그램 스트림은 단순히 자신에 속한 기본 스트림들을 포함하고 보통은 가변 길이들의 패킷들을 담고 있다. 프로그램 스트림에서, 기여하는 기본 스트림들로부터 유래되는 PES-패킷들은 "팩들(packs)"로 편성된다. 팩은 팩-헤더, 옵션적 시스템-헤더, 및 기여하는 기본 스트림들 중의 어느 것으로부터 취해진 임의의 수의 PES-패킷들을 임의의 순서로 포함한다. 시스템 헤더는 프로그램 스트림의 최대 데이터 레이트, 기여하는 비디오 및 오디오 기본 스트림들의 수, 추가 타이밍 정보, 또는 기타의 정보와 같은 프로그램 스트림의 특성들의 요약물 담고 있다. 디코더는 시스템 헤더에 들어 있는 정보를 이용하여 디코더가 프로그램 스트림을 디코딩할 있는지를 결정할 수 있다.
- [0043] 멀티플렉서 (30)는 잠재적으로 에러 발생이 쉬운 채널들을 통한 복수의 프로그램들의 동시 전달을 위해 전송 스트림을 사용할 수 있다. 전송 스트림은 브로드캐스팅과 같은 멀티-프로그램 애플리케이션들을 위해 고안된 다중화물이라서, 단일 전송 스트림은 많은 독립 프로그램들을 수용할 수 있다. 전송 스트림은 전송 패킷들 각각이 188-바이트 길이인 연속하는 전송 패킷들을 포함할 수 있다. 짧은 고정 길이 패킷들의 사용은 전송 스트림이 프로그램 스트림보다는 에러들에 덜 민감하게 한다. 게다가, 각각의 188-바이트 길이 전송 패킷에는 표준 에러 방지 프로세스, 이를테면 리드-솔로몬 인코딩을 통해 패킷들을 처리함으로써 부가적인 에러 방지가 주어질 수 있다. 전송 스트림의 개선된 에러 내성은 에러발생이 쉬운 채널들을 존속시킬 양호한 기회가 예를 들어 브로드캐스트 환경에서 발견될 수 있다는 것을 의미한다.
- [0044] 전송 스트림은 그것의 증대된 에러 내성 및 많은 동시 프로그램들(simultaneous programs)을 운반하는 능력 때문에 프로그램 스트림보다 양호하다고 보여진다. 그러나, 전송 스트림은 프로그램 스트림보다 더 정교한 다중화물이고 결과적으로 프로그램 스트림보다는 생성하기가 더 어렵고 역다중화하기는 더 복잡하다. 전송 패킷의 제 1 바이트는 0x47 (16진수 47, 이진수 '01000111', 십진수 71)의 값을 갖는 동기화 바이트일 수 있다. 단일 전송 스트림은 각각의 프로그램이 많은 패킷화된 기본 스트림들을 포함하는 많은 상이한 프로그램들을 운반할 수 있다. 멀티플렉서 (30)는 13비트 패킷 식별자 (PID) 필드를 이용하여 하나의 기본 스트림의 데이터를 담고 있는 전송 패킷들을 다른 기본 스트림들의 데이터를 운반하는 전송 패킷들로부터 구별할 수 있다. 멀티플렉서의 의무는 각각의 기본 스트림에 고유 PID 값이 주어지는 것을 보장하는 것이다. 전송 패킷의 마지막 바이트는 연속성 카운트 필드일 수 있다. 멀티플렉서 (30)는 동일한 기본 스트림들에 속한 연속하는 전송 패킷들 사이에서 연속성 카운트 필드의 값을 증분 만큼 증가시킨다. 이는 목적지 디바이스, 이를테면 A/V 목적지 디바이스 (40)의 디코더 또는 다른 유닛이 전송 패킷의 손실 또는 이득을 검출하고 및 바라건대 그렇지 않으면 그러한 이벤트로 생길 에러들을 숨기는 것을 가능하게 한다.
- [0045] 멀티플렉서 (30)는 프로그램의 기본 스트림들에 대한 PES 패킷들을 오디오 인코더 (26) 및 비디오 인코더 (28)로부터 수신하고 이 PES 패킷들로부터 상응하는 네트워크 추상화 계층 (NAL) 단위들을 형성한다. H.264/AVC (Advanced Video Coding)의 예에서, 코딩된 비디오 세그먼트들은 NAL 단위들에 편제되는데, 이는 화상 통화, 스토리지, 브로드캐스트, 또는 스트리밍과 같은 "네트워크 친화적 (network-friendly)" 비디오 표현 어드레싱 응용들을 제공한다. NAL 단위들은 비디오 코딩 계층 (VCL) NAL 단위들 및 비-VCL NAL 단위들로 분류될 수 있다. VCL NAL 단위들은 코어 압축 엔진을 담고 있고 블록, 매크로블록, 및/또는 슬라이스 레벨들을 포함할 수 있다. 기타 NAL 단위들은 비-VCL NAL 단위들이다.
- [0046] 멀티플렉서 (30)는 NAL이 속한 프로그램을 식별하는 헤더, 뿐만 아니라 페이로드, 예컨대, 오디오 데이터, 비디오 데이터, 또는 NAL 단위에 상응하는 전송 또는 프로그램 스트림을 기술하는 데이터를 포함하는 NAL 단위들을 형성할 수 있다. 예를 들어, H.264/AVC에서, NAL 단위는 1바이트 헤더 및 가변 사이즈의 페이로드를 구비한다. 하나의 예에서, NAL 단위 헤더는 priority\_id 요소, temporal\_id 요소, anchor\_pic\_flag 요소, view\_id 요소, non\_idr\_flag 요소, 및 inter\_view\_flag 요소를 포함한다. 기존의 MVC에서, H.264에 의해

정의되는 NAL 단위는 4-바이트 MVC NAL 단위 헤더 및 NAL 단위 페이로드를 구비하는 점두 NAL 단위들 및 MVC 코딩된 슬라이스 NAL 단위들을 제외하면, 존속된다.

[0047] NAL 헤더의 priority\_id 요소는 간단한 일-경로(one-path) 비트스트림 적응 프로세스를 위해 사용될 수 있다. temporal\_id 요소는 상응하는 NAL 단위의 시간적 레벨을 지정하기 위해 사용될 수 있는데, 상이한 시간적 레벨들이 상이한 프레임 레이트들에 상응한다.

[0048] anchor\_pic\_flag 요소는 화상이 앵커 화상인지 비-앵커 화상인지를 나타낼 수 있다. 앵커 화상들 및 출력 순서 (다시 말하면, 디스플레이 순서) 에서 그것에 계속되는 모든 화상들은 디코딩 순서 (다시 말하면, 비트스트림 순서) 상의 이전 화상들의 디코딩 없이도 제대로 디코딩될 수 있고, 그래서, 임의 액세스 지점들로서 사용될 수 있다. 앵커 화상들 및 비-앵커 화상들은 상이한 의존도들을 가질 수 있고 이것들 양쪽 모두는 시퀀스 매개변수 집합으로 시그널링된다. 기타 플래그들이 이 장의 다음 섹션들에서 논의되고 사용된다. 이러한 앵커 화상은 개방형 GOP(Group Of Pictures) 액세스 지점이라고도 할 수 있는 반면, 폐쇄형 GOP 액세스 지점은 non\_idr\_flag 요소가 영(0)이 될 때 지원된다. non\_idr\_flag 요소는 화상이 순간 디코더 리프레시(IDR) 또는 뷰 IDR (V-IDR) 화상인지를 나타낸다. 대체로, IDR 화상, 및 출력 순서 또는 비트스트림 순서상 이것에 연속되는 모든 화상들은, 디코딩 순서 또는 디스플레이 순서 중 어느 것에서나 이전 화상들의 디코딩 없이 제대로 디코딩될 수 있다.

[0049] view\_id 요소는 뷰를 식별하는데 사용될 수 있는 구문 정보를 포함할 수 있는데, 이는 MVC 디코더 내부, 예컨대, 뷰 간 예측, 및 디코더 외부, 예컨대, 렌더링을 위해 데이터 상호작용성을 위해 사용될 수 있다. inter\_view\_flag 요소는 상응하는 NAL 단위가 뷰 간 예측을 위해 다른 뷰들에 의해 사용되는지를 지정할 수 있다. AVC를 준수할 수 있는 기본 뷰에 대한 4-바이트 NAL 단위 헤더 정보를 운반하기 위해, 점두 NAL 단위가 MVC에서 정의된다. MVC의 측면에서, 기본 뷰 액세스 단위는 뷰의 현재 시간 인스턴스의 VCL NAL 단위들뿐만 아니라, NAL 단위 헤드만을 담고 있는 자신의 선두 NAL 단위를 구비한다. H.264/AVC 디코더는 점두 NAL 유닛을 무시할 수 있다.

[0050] 비디오 데이터를 자신의 페이로드에 구비하는 NAL 단위는 비디오 데이터의 다양한 입도(granularity) 레벨들을 포함할 수 있다. 예를 들어, NAL 단위는 비디오 데이터의 블록, 매크로블록, 복수의 매크로블록들, 비디오 데이터의 슬라이스, 또는 비디오 데이터의 전체 프레임을 포함할 수 있다. 멀티플렉서 (30) 는 인코딩된 비디오 데이터를 비디오 인코더 (28) 로부터 기본 스트림들의 PES 패킷들의 형태로 수신할 수 있다. 멀티플렉서 (30) 는 stream\_id들을 상응하는 프로그램들에, 예컨대, 데이터베이스 또는 다른 데이터 구조, 이를테면 프로그램 맵 테이블 (PMT) 또는 프로그램 스트림 맵 (PSM) 에서 매핑함으로써 각각의 기본 스트림을 상응하는 프로그램에 관련시킬 수 있다.

[0051] 멀티플렉서 (30) 는 액세스 단위들을 복수의 NAL 단위들로부터 어셈블링할 수도 있다. 대체로, 액세스 단위는 한 프레임의 비디오 데이터를 표현하기 위한 하나 이상의 NAL 단위들, 뿐만 아니라 그 프레임에 상응하는 오디오 데이터를 그러한 오디오 데이터가 이용가능할 때 포함할 수 있다. 액세스 단위는 일반적으로 하나의 출력 타임 인스턴스에 대한 모든 NAL 단위들, 예컨대, 하나의 타임 인스턴스에 대한 모든 오디오 및 비디오 데이터를 구비한다. 예를 들어, 각각의 뷰가 초당 20개 프레임들의 프레임 레이트 (fps) 를 가진다면, 각각의 타임 인스턴스는 0.05 초의 시간 간격에 상응할 수 있다. 이 시간 간격 동안, 동일한 액세스 단위 (동일한 타임 인스턴스) 의 모든 뷰들에 대한 지정 프레임들은 동시에 렌더링될 수 있다. H.264/AVC에 해당하는 예에서, 액세스 단위는 코딩된 화상을 하나의 타임 인스턴스에 포함할 수 있는데, 이는 기본의 (primary) 코딩된 화상으로서 표현될 수 있다. 따라서, 액세스 단위는 공통의 시간적 인스턴스의 모든 오디오 및 비디오 프레임들, 예컨대, 시간 X에 상응하는 모든 뷰들을 포함할 수 있다. 이 개시물은 또한 특정 뷰의 인코딩된 화상을 "뷰 성분(view component)"이라고 한다. 다시 말하면, 뷰 성분은 특정 시간에서의 특정 뷰에 대한 인코딩된 화상 (또는 프레임) 을 포함할 수 있다. 따라서, 액세스 단위는 공통의 시간적 인스턴스의 모든 뷰 성분들을 포함하는 것으로 정의될 수 있다. 액세스 단위들의 디코딩 순서는 출력 또는 디스플레이 순서와 반드시 동일할 필요는 없다.

[0052] 멀티플렉서 (30) 는 프로그램에 관련한 데이터를 NAL 단위 내에 포함시킬 수도 있다. 예를 들어, 멀티플렉서 (30) 는 프로그램 맵 테이블 (PMT) 또는 프로그램 스트림 맵 (PSM) 을 포함하는 NAL 단위를 만들 수 있다. 대체로, PMT가 전송 스트림을 기술하는데 사용되는 반면, PSM은 프로그램 스트림을 기술하는데 사용된다. 아래에서 도 2의 예에 관하여 더 상세히 설명되는 바와 같이, 멀티플렉서 (30) 는 오디오 인코더 (26) 및 비디오 인코더 (28) 로부터 수신된 기본 스트림들을 프로그램들과 관련시키고 그에 따라 개별 전송 스트림들 및/

또는 프로그램 스트림들과 관련시키는 데이터 저장 유닛을 포함하거나 그것과 상호작용할 수 있다.

[0053] 대부분의 비디오 표준들과 마찬가지로, H.264/AVC는 예러 없는 비트스트림들에 대한 구문, 의미론, 및 디코딩 프로세스를 이것들 중의 어떤 것이라도 일정한 프로파일 또는 레벨을 따르게끔 정의한다. H.264/AVC는 인코더를 지정하지는 않지만, 인코더는 발생된 비트스트림들이 디코더에 대한 표준을 준수하는 것을 보장하는 임무를 맡는다. 비디오 코딩 표준의 측면에서, "프로파일"은 알고리즘들, 특징들(features), 또는 도구들 및 그것들에 적용되는 제약들의 서브세트에 해당한다. H.264 표준에 의해 정의된 바와 같이, 예를 들어, "프로파일"은 H.264 표준에 의해 지정되는 전체 비트스트림 구문의 서브세트이다. "레벨"은 예를 들어, 화상들의 해상도, 비트 레이트, 및 매크로블록 (MB) 처리 레이트에 관련되는 디코더 메모리 및 계산과 같은 디코더 자원 소비의 한계에 상응한다.

[0054] H.264 표준은, 예를 들어, 주어진 프로파일의 구문에 의해 부과되는 경계들 내에서, 디코딩된 화상들의 지정된 사이즈와 같은 비트스트림 내의 구문 요소들에 의해 취해지는 값들에 의존하여 인코더들 및 디코더들의 성능에서의 큰 변화를 요구하는 것이 여전히 가능하다는 것을 인정한다. H.264 표준은 추가로 특정 프로파일 내에서 구문의 모든 가상적 사용들을 다룰 수 있는 디코더를 구현하는 것이 실제적이지도 않고 경제적이지도 않다는 것을 인정한다. 따라서, H.264 표준은 "레벨"을 비트스트림에서 구문(syntax) 요소들의 값들에 부과되는 지정된 제약들의 집합으로서 정의한다. 이들 제약들은 값들에 대한 간단한 제한들일 수 있다. 다르게는, 이들 제약들은 값들의 산술적 조합들 (예컨대, 화상 폭 곱하기 화상 높이 곱하기 초당 디코딩되는 화상들의 수)에 대한 제약들의 형태를 취할 수 있다. H.264 표준은 추가로 개개의 구현물들이 각각의 지원되는 프로파일들에 대해 상이한 레벨을 지원할 수 있게 한다.

[0055] 프로파일을 준수하는 디코더는 보통은 프로파일에서 정의되는 모든 특징들을 지지한다. 예를 들어, 코딩 특징으로서, B-화상 코딩은 H.264/AVC의 기준 프로파일에 의해 지원되지 않지만 H.264/AVC의 다른 프로파일들에서 지원된다. 레벨을 준수하는 디코더는 레벨에서 정의된 한계를 넘어서는 자원들을 요구하지 않는 어떤 비트스트림이라도 디코딩할 수 있어야 한다. 프로파일들 및 레벨들의 정의들은 해석능력에 도움이 될 수 있다. 예를 들어, 비디오 전송 동안, 한 쌍의 프로파일 및 레벨 정의들은 전체 전송 세션을 위해 협상되고 합의될 수 있다. 더 구체적으로는, H.264/AVC에서, 레벨은 예를 들어, 처리될 것이 필요한 매크로블록들의 수, 디코딩된 화상 버퍼 (DPB) 사이즈, 코딩된 화상 버퍼 (CPB) 사이즈, 세로방향 움직임 벡터 범위, 연속하는 2개의 MB들의 최대 움직임 벡터 수, 및 B-블록이 8x8화소 미만의 서브-매크로블록 파티션들을 가지는지의 여부에 관한 한계를 정의할 수 있다. 이런 방식으로, 디코더는 디코더가 비트스트림을 제대로 디코딩할 수 있는지를 결정할 수 있다.

[0056] 매개변수 집합들은 일반적으로 시퀀스-계층 헤더 정보를 시퀀스 매개변수 집합들 (SPS) 내에 그리고 드물게 변경되는 화상-계층 헤더 정보를 화상 매개변수 집합들 (PPS) 내에 담고 있다. 매개변수 집합들을 이용하여, 이 드물게 변경되는 정보는 각각의 시퀀스 또는 화상에 대해 반복될 필요는 없고; 그래서, 코딩 효율이 개선될 수 있다. 더욱이, 매개변수 집합들의 이용은 헤더 정보의 대역 외 전송을 가능할 수 있게 함으로써, 불필요한 전송이 필요 없게 하여 예러 내성을 달성한다. 대역 외 전송에서, 매개변수 집합 NAL 단위들은 다른 NAL 단위들과는 상이한 채널상으로 전송된다.

[0057] MPEG-2 시스템들의 표준은 시스템의 확장들을 "기술자들 (descriptors)"에 의해 허용한다. PMT들 및 PSM들은 하나 이상의 기술자들이 삽입될 수 있는 기술자 루프들을 구비한다. 대체로, 기술자가 프로그램들 및/또는 프로그램 요소들의 정의를 확장하는데 사용될 수 있는 데이터 구조를 포함할 수 있다. 이 개시물은 이 개시물의 기법들을 수행하기 위한 동작점 기술자들을 설명한다. 대체로, 이 개시물의 동작점 기술자는 랜더링 능력, 디코딩 능력, 및 동작점에 대한 비트레이트를 기술함으로써 기존의 MVC 확장 기술자를 개선시킨다. 목적지 디바이스, 이를테면 A/V 목적지 디바이스 (40)는 각각의 동작점에 대한 동작점 기술자들을 사용하여 비트스트림의 동작점들 중 디코딩될 하나를 선택할 수 있다.

[0058] 각각의 PMT 또는 PSM은 동작점의 특성들을 기술하는 동작점 기술자를 구비할 수 있다. 예를 들어, 소스 디바이스 (20)는 클라이언트 디바이스 (40)에 대한 랜더링 능력을 기술하는 랜더링 능력 값을 제공하는 동작점 기술자를 제공할 수 있다. 클라이언트 디바이스 (40)가 동작점의 비디오 데이터를 적절히 랜더링 (예컨대, 디스플레이) 하도록 하기 위하여, 클라이언트 디바이스 (40)는 랜더링 능력 값에 의해 시그널링되는 랜더링 능력들을 충족해야 한다. 랜더링 능력 값은 예를 들어, 디스플레이할 뷰들의 수 (예컨대, 랜더링의 대상이 되는 뷰들의 수) 및/또는 이 뷰들에 대한 비디오 데이터의 프레임 레이트를 기술할 수 있다. 그래서, 클라이언트 디바이스 (40)는 클라이언트 디바이스 (40)의 비디오 출력단 (44)이 동작점 기술자에 의해 지정되는 프



레이프 레이트로 동작점의 뷰들의 수를 디스플레이할 수 있을 때 렌더링 능력들이 충족된다고 결정할 수 있다.

[0059] 소스 디바이스 (20) 가 MVC 비트스트림을 멀티캐스트 또는 브로드캐스트 프로토콜을 이용하여 전송하는 예들에서, 소스 디바이스 (20) 는 전체 MVC 비트스트림을 전송 스트림들로 패킷화할 수 있는데, 이 전송 스트림들은 다양한 렌더링 능력들을 갖는 클라이언트 디바이스들에 의해 수신될 수 있다. 예를 들어, 일부 3차원 프로그램들은 상이한 수들의 뷰들 (예컨대, 2개의 뷰들, 4개의 뷰들, 6개의 뷰들, 또는 8개의 뷰들) 을 가질 수 있고, 다양한 디바이스들은 한 쌍 및 4쌍들의 뷰들 중 어느 것이든 이용할 수 있다. 그래서, 각각의 클라이언트 디바이스는 클라이언트 디바이스에 의해 디스플레이될 수 있는 뷰들의 지원되는 수에 기초하여 어떤 동작점이 사용될지를 결정할 수 있다. 예를 들어, 클라이언트 디바이스 (40) 는 비디오 출력단 (44) 에 의해 디스플레이될 수 있는 뷰들의 수 및 비디오 출력단 (44) 이 비디오 데이터를 디스플레이할 수 있는 프레임 레이트를 결정하고 비디오 출력단 (44) 의 렌더링 능력들에 기초하여 사용되어야 할 동작점들을 결정함으로써, 어떤 동작점들을 사용할지를 결정할 수 있다.

[0060] 소스 디바이스가 MVC 비트스트림을 유니캐스트 프로토콜을 이용하여 전송하는 예에서, 클라이언트 디바이스 (40) 는 상응하는 동작점 기술자들에서 지정되는 렌더링 능력을 점검함으로써 허용가능한 수의 뷰들로 프로그램에 상응하는 세션을 확립할 수 있다. 마찬가지로, MVC 비트스트림이 로컬 재생을 위해 컴퓨터-관독가능 저장 매체에 인코딩되는 예들에서, 클라이언트 디바이스 (40) 는 PMT들 또는 PSM들의 동작점 기술자들에서 지정된 렌더링 능력을 점검함으로써 적당한 프로그램을 선택할 수 있다.

[0061] 소스 디바이스 (20) 는 디코딩 능력들 값을 동작점 기술자로 제공할 수도 있다. 디코딩될 뷰들의 수는 디스플레이할 뷰들의 수와 동일할 필요는 없다. 그러므로, 동작점 기술자는 동작점에 대한 디코딩할 뷰들의 수 및 디스플레이할 뷰들의 수를 별도로 시그널링할 수 있다. 더군다나, 동작점 기술자는 동작점에 상응하는 뷰들을 명확하게 식별할 수 있다. 일정한 클라이언트 디바이스들은 예컨대, 시야각 (viewing angle) 에 기초하여 다양한 목적을 위한 특정 뷰들을 선호할 수 있다. 따라서, 클라이언트 디바이스 (40) 는 어떤 뷰들이 동작점에서 이용가능한지에 기초하여 동작점을 선택하도록 구성될 수 있다.

[0062] 일부 예들에서, 동작점에서 시그널링되는 디코딩 능력들은 동작점에 상응하는 레벨과 프로파일을 부가적으로 또는 대신 지정할 수 있다. 소스 디바이스 (20) 가 비트스트림을 멀티캐스트 또는 브로드캐스트 프로토콜을 이용하여 전송하는 예들에서, 상이한 디코딩 능력들을 가지는 다양한 클라이언트 디바이스들이 비트스트림을 수신할 수 있다. 예를 들어, 일부 디코더들은 30 fps를 갖는 2개의 뷰들만을 디코딩할 수 있는 반면, 일부 디코더들은 60 fps를 갖는 4개의 뷰들을 디코딩할 수 있다. 소스 디바이스 (20) 가 비트스트림을 유니캐스트 프로토콜을 이용하여 전송하는 예들에서, 클라이언트 디바이스 (40) 는 PMT들의 기술자들에서 지정된 디코딩 능력을 점검한 후 적당한 세션을 (특정한 3차원 프로그램을 위해) 확립할 수 있다. 마찬가지로, 로컬 재생을 위해, 클라이언트 디바이스 (40) 는 PMT들 또는 PSM들의 동작점 기술자들로 지정되는 디코딩 능력을 점검함으로써 적당한 프로그램을 선택할 수 있다.

[0063] 소스 디바이스 (20) 는 부가적으로 비트레이트 정보를 동작점 기술자로 시그널링할 수 있다. 비트레이트 정보는 동작점에 대한 평균 비트레이트 및 최대 비트레이트 중 어느 한쪽 또는 양쪽 다를 기술할 수 있다. 예를 들어, 소스 디바이스 (20) 가 비트스트림을 유니캐스트 프로토콜을 이용하여 전송할 때, 비트스트림을 전송하는데 사용되는 채널은 대역폭 측면에서 제한될 수 있다. 따라서, 클라이언트 디바이스 (40) 는 통신 채널에 대한 허용가능(tolerable) 최대 또는 평균 비트레이트를 갖는 동작점을 선택할 수 있다.

[0064] 일부 예들에서, 소스 디바이스 (20) 는 동작점의 프레임 레이트를 동작점 기술자에서 추가로 지정할 수 있다. 동작점의 일정한 뷰들은 동작점의 프레임 레이트에 매칭되지 않는 프레임 레이트들을 가질 수 있다. 그래서, 클라이언트 디바이스 (40) 는 비디오 데이터를 디스플레이할 목적으로 디코딩된 비디오 데이터를 리어셈블링하는 프로세스를 용이하게 하도록 동작점의 프레임 레이트 및 그러한 뷰의 프레임 레이트를 결정할 수 있다. 다양한 예들에서, 2개의 동작점들의 프레임 레이트들이 매칭되지 않는 경우, 클라이언트 디바이스 (40) 는 더 높은 프레임 레이트를 갖는 동작점의 뷰들로부터 프레임들을 버릴 수 있거나 또는 낮은 프레임 레이트를 갖는 동작점의 뷰들로부터의 프레임들을 보간할 수 있다.

[0065] 통상, 기본 스트림은 기본 스트림이 SEI 메시지들 및 점두 NAL 단위들을 구별하는지를 각각 기술하는 플래그들인 "no\_sei\_nal\_unit\_present" 및 "no\_prefix\_nal\_unit\_present"를 구비한다. 이 개시물은 클라이언트 디바이스들, 이를테면 클라이언트 디바이스 (40) 가, SEI 메시지들 및/또는 점두 NAL 단위들이 동작점 내에 존재하는지를 추정하지만, 이들 값들을 동작점을 위해 명백하게 시그널링하지는 않는 것을 제안한다. SEI 메시지들이 동작점에 존재하는지를 결정하기 위하여, 클라이언트 디바이스 (40) 는 동작점에 대한 기본 스트림들의

no\_sei\_nal\_unit\_present 값들 중의 최대 값이 일 (1) 과 동일한지를 결정할 수 있다. 마찬가지로, 점두 NAL 단위들이 동작점에 존재하는지를 결정하기 위하여, 클라이언트 디바이스 (40) 는 동작점에 대한 기본 스트림들의 no\_prefix\_nal\_unit\_present 값들 중의 최대 값이 일 (1) 과 동일한지를 결정할 수 있다.

[0066] 위에서 논의된 예들은 MVC 비트스트림 각각의 동작점을 위해 포함되는 동작점 기술자들에 집중하고 있다. 대체예로서, 소스 디바이스 (20) 는 유사한 데이터를 시그널링하는 MVC 확장 기술자들을 제공할 수 있다. 예를 들어, 소스 디바이스 (20) 는 하나를 넘는 MVC 확장 기술자를 기본 스트림에 상응하는 MVC 비디오 서브-비트스트림과 관련시킬 수 있다. 소스 디바이스 (20) 는 서브-비트스트림에 대한 MVC 확장 기술자에서, 프레임 레이트, 디스플레이될 뷰들의 view\_id 서브세트, 및 디코딩될 뷰들의 수를 지정할 수 있다. 소스 디바이스 (20) 는 추가로 MVC 확장 기술자들 및 상응하는 동작점 사이의 매핑을 시그널링할 수 있다.

[0067] ITU-T H.261, H.262, H.263, MPEG-1, MPEG-2 및 H.264/MPEG-4 파트 10과 같은 비디오 압축 표준들은 시간적 리던던시를 줄이기 위해 움직임 보상 시간적 예측을 이용한다. 인코더는 움직임 벡터들에 따라 현재 인코딩된 화상들을 예측하기 위해 일부의 이전에 인코딩된 화상들 (여기서는 프레임들이라고도 한다) 로부터의 움직임 보상 예측을 이용한다. 전형적인 비디오 코딩에는 3가지 주요한 화상 유형들이 존재한다. 그것들은 인트라 코딩된 화상 ("I-화상들" 또는 "I-프레임들"), 예측된 화상들 ("P-화상들" 또는 "P-프레임들") 및 양방향 예측된 화상들 ("B-화상들" 또는 "B-프레임들") 이다. P-화상들은 시간적 순서에서 현재 화상 앞의 참조 화상만을 사용한다. B-화상에서, B-화상 각각의 블록은 하나 이상의 참조 화상들로부터 예측될 수 있다. 이들 참조 화상들은 시간적 순서에서 현재 화상 앞 또는 뒤에 위치될 수 있다.

[0068] H.264 코딩 표준을 따르면, 일 예로서, B-화상들은 이전에 코딩된 참조 화상들의 2개의 목록들인, 목록 0 및 목록 1을 이용한다. 이들 2개의 목록들은 각각이 시간적 순서에서 과거 및/또는 미래의 코딩된 화상들을 담고 있다. B-화상의 블록들은 목록 0의 참조 화상으로부터의 움직임 보상 예측, 목록 1의 참조 화상으로부터의 움직임 보상 예측, 또는 목록 0 및 목록 1 양쪽 다의 참조 화상들의 조합으로부터의 움직임 보상 예측과 같은 수 개의 방법들 중의 하나로 예측될 수 있다. 목록 0 및 목록 1 양쪽 다의 참조 화상들의 조합을 얻기 위하여, 2개의 움직임 보상된 참조 영역들이 각각 목록 0 및 목록 1 참조 화상으로부터 얻어진다. 그것들의 조합은 현재 블록을 예측하는데 이용될 수 있다.

[0069] ITU-T H.264 표준은 인트라 예측을 다양한 블록 사이즈들, 이를테면 루마(luma) 성분들에 대한 16 바이 16, 8 바이 8, 또는 4 바이 4, 및 크로마(chroma) 성분들에 대한 8×8에서, 뿐만 아니라 인터 예측을 다양한 블록 사이즈들, 이를테면 루마 성분들에 대한 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 및 4×4 및 크로마 성분들에 대한 상응하는 스케일의 사이즈들에서 지원한다. 이 개시물에서, "×" 및 "바이 (by)"는 세로방향 및 가로방향 치수들의 측면에서 블록의 화소 치수들을, 예컨대, 16×16개의 화소들 또는 16 바이 16개의 화소들을 상호 교환적으로 말하는데 사용될 수 있다. 대체로, 16×16 블록은 세로 방향의 16개의 화소들 (y = 16) 및 가로 방향의 16개의 화소들 (x = 16) 을 가질 수 있다. 비슷하게, NxN 블록은 일반적으로 세로 방향의 N개의 화소들 및 가로 방향의 N개의 화소들을 가지며, 여기서 N은 음이 아닌 정수 값을 나타낸다. 블록의 화소들은 행들 및 열들로 배열될 수 있다.

[0070] 16 바이 16보다 작은 블록 사이즈들은 16 바이 16 매크로블록의 파티션들이라고 할 수 있다. 비디오 블록들은 화소 도메인에서는 화소 데이터의 블록들을 포함하거나, 또는 변환 도메인에서는, 예컨대, 다음의 변환 응용들 이를테면 이산 코사인 변환 (DCT), 정수 변환, 웨이브릿 변환, 또는 코딩된 비디오 블록들 및 예측 비디오 블록들 사이의 화소 차이들을 나타내는 잔여 비디오 블록 데이터에 대한 개념적으로 유사한 변환에서는 변환 계수들의 블록을 포함할 수 있다. 일부 경우들에서, 비디오 블록은 변환 도메인의 양자화된 변환 계수들의 블록들을 포함할 수 있다.

[0071] 작은 비디오 블록들일수록 양호한 해상도를 제공할 수 있고, 높은 세부사항 레벨들을 포함하는 비디오 프레임의 위치들을 위해 사용될 수 있다. 대체로, 매크로블록들 및 때때로 서브-블록들이라고 하는 다양한 파티션들은 비디오 블록들이라고 간주한다. 부가하여, 슬라이스는 복수의 비디오 블록들, 이를테면 매크로블록들 및 /또는 서브-블록들이라고 간주될 수 있다. 각각의 슬라이스는 비디오 프레임의 독자적으로 디코딩 가능한 단위일 수 있다. 다르게는, 프레임들 자체가 디코딩가능 단위들일 수 있거나, 또는 프레임의 기타 부분들이 디코딩가능 단위들로서 정의될 수 있다. "코딩된 단위(coded unit)" 또는 "코딩 단위(coding unit)"라는 용어는 전체 프레임, 프레임의 슬라이스, 시퀀스라고도 하는 화상들의 그룹 (GOP) 과 같은 비디오 프레임의 임의의 독립적 디코딩가능 단위, 또는 적용가능 코딩 기법들에 따라 정의되는 다른 독립적 디코딩가능 단위를 말한다.

- [0072] 매크로블록이란 용어는 16×16개의 화소들을 포함하는 2차원 화소 어레이에 따른 화상들 및/또는 비디오 데이터를 인코딩하기 위한 데이터 구조를 말한다. 각각의 화소는 색차 성분 및 휘도 성분을 포함한다. 따라서, 매크로블록은 각각이 8×8개 화소들의 2차원 어레이를 포함하는 4개의 휘도 블록들, 각각이 16×16개 화소들의 2차원 어레이를 포함하는 2개의 색차 블록들, 및 구문 정보, 이를테면 코딩된 블록 패턴 (CBP), 인코딩 모드 (예컨대, 인트라- (I), 또는 인터- (P 또는 B) 인코딩 모드들), 인트라 인코딩된 블록의 파티션들에 대한 파티션 사이즈 (예컨대, 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 또는 4×4), 또는 인트라 인코딩된 매크로블록에 대한 하나 이상의 움직임 벡터들을 포함하는 헤더를 정의할 수 있다.
- [0073] 비디오 인코더 (28), 비디오 디코더 (48), 오디오 인코더 (26), 오디오 디코더 (46), 멀티플렉서 (30), 및 디멀티플렉서 (38) 각각은 적용가능한 바대로 다양한 적당한 인코더 또는 디코더 회로 중의 어느 것으로서, 이를테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 개별 로직 회로, 소프트웨어, 하드웨어, 펌웨어 또는 이것들의 어떤 조합으로도 구현될 수 있다. 비디오 인코더 (28) 및 비디오 디코더 (48) 각각은 하나 이상의 인코더들 또는 디코더들 내에 구비될 수 있고, 그것들 각각은 결합형 비디오 인코더/디코더 (CODEC) 의 부분으로서 통합될 수 있다. 비슷하게, 오디오 인코더 (26) 및 오디오 디코더 (46) 각각은 하나 이상의 인코더들 또는 디코더들 내에 구비될 수 있고, 그것들 각각은 결합형 CODEC의 부분으로서 통합될 수 있다. 비디오 인코더 (28), 비디오 디코더 (48), 오디오 인코더 (26), 오디오 디코더 (46), 멀티플렉서 (30), 및/또는 디멀티플렉서 (38) 를 구비한 장치는 집적회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 이를테면 셀룰러 전화기를 포함할 수 있다.
- [0074] 이 개시물의 기법들은 동작점들의 특성들의 시그널링을 제공하지 않는 MVC 서브-비트스트림들에 대한 기존의 기법들을 넘어서는 어떤 이점들을 제공할 수 있다. 각각의 서브-비트스트림은 상응하는 비트스트림의 하나 이상의 뷰들을 구비할 수 있다. 일부 경우들에서, 동작점은 상이한 비트스트림들의 뷰들에 상응할 수 있다. 이 개시물의 기법들은 상응하는 동작점의 뷰들을 식별하는 동작점 기술자를 제공한다.
- [0075] 멀티플렉서 (30) 가 NAL 단위 및/또는 액세스 단위를 수신된 데이터로부터 어셈블링한 후, 멀티플렉서 (30) 는 그 단위를 출력을 위해 출력 인터페이스 (32) 에 전해준다. 출력 인터페이스 (32) 는 예를 들어, 전송기, 송수신기, 데이터를 예를 들어, 광학적 드라이브, 자기 매체 드라이브 (예, 플로피 드라이브) 와 같은 컴퓨터 판독가능 매체에 기록하기 위한 디바이스, 범용 직렬 버스 (USB) 포트, 네트워크 인터페이스, 또는 기타 출력 인터페이스를 포함할 수 있다. 출력 인터페이스 (32) 는 NAL 단위 또는 액세스 단위를 예를 들어, 전송 신호, 자기 매체, 광학적 매체, 메모리, 플래시 드라이브, 또는 기타 컴퓨터 판독가능 매체와 같은 컴퓨터 판독가능 매체 (34) 에 출력한다.
- [0076] 궁극적으로, 입력 인터페이스 (36) 는 컴퓨터 판독가능 매체 (34) 로부터 데이터를 검색한다. 입력 인터페이스 (36) 는 예를 들어, 광학적 드라이브, 자기 매체 드라이브, USB 포트, 수신기, 송수신기, 또는 기타의 컴퓨터 판독가능 매체 인터페이스를 포함할 수 있다. 입력 인터페이스 (36) 는 NAL 단위 또는 액세스 단위를 디멀티플렉서 (38) 에 제공할 수 있다. 디멀티플렉서 (38) 는 전송 스트림 또는 프로그램 스트림을 구성요소인 PES 스트림들로 역다중화하며, PES 스트림들을 패킷화하여 인코딩된 데이터를 회수하고, 인코딩된 데이터를 오디오 디코더 (46) 또는 비디오 디코더 (48) 에, 예컨대, 스트림의 PES 패킷 헤더들에 의해 지시된 바와 같이, 인코딩된 데이터가 오디오 또는 비디오 스트림의 부분인지에 따라 송신한다. 오디오 디코더 (46) 는 인코딩된 오디오 데이터를 디코딩하고 디코딩된 오디오 데이터를 오디오 출력단 (42) 에 출력하는 반면, 비디오 디코더 (48) 는 인코딩된 비디오 데이터를 디코딩하고 스트림의 복수의 뷰들을 구비할 수 있는 디코딩된 비디오 데이터를 비디오 출력단 (44) 에 송신한다. 비디오 출력단 (44) 은 한 장면의 복수의 뷰들을 이용하는 디스플레이, 예컨대, 장면 각각의 뷰를 동시에 표현하는 입체 또는 자동입체 디스플레이를 포함할 수 있다.
- [0077] 특히, 디멀티플렉서 (38) 는 수신된 비트스트림의 동작점을 선택할 수 있다. 예를 들어, 디멀티플렉서 (38) 는 비트스트림의 동작점들의 특성들을 비교하여 A/V 목적지 디바이스 (40) 에 의해 사용될 적절한 동작점을 선택할 수 있다. 대체로, 디멀티플렉서 (38) 는 비디오 디코더 (48) 에 의해 디코딩될 사용자에게 대한 최고 품질의 시청 경험을 제공할 동작점들 중의 하나를 선택하기 위해 시도할 수 있다. 예를 들어, 디멀티플렉서 (38) 는 비디오 디코더 (48) 의 랜더링 능력들 및 디코딩 능력들을 비트스트림의 동작점 기술자들에 의해 시그널링되는 제안된 랜더링 및 디코딩 능력들과 비교할 수 있다. 디멀티플렉서 (38) 가 결정하여 비디오 디코더 (48) 에 의해 적절히 디코딩되는 동작점들 중에서, 디멀티플렉서 (38) 는 최고 품질 비디오 데이터, 예컨대, 최고 프레임 레이트 및/또는 비트레이트를 제공하는 동작점을 선택할 수 있다. 다른 예들에서, 디멀티플렉서 (38) 는 지원되는 동작점들 중 하나를 예를 들어, 소비 전력과 같은 다른 고려사항들에 기초하여 선택할 수

있다.

- [0078] 도 2는 멀티플렉서 (30)(도 1)의 구성요소들의 예시적인 배치구성을 도시하는 블록도이다. 도 2의 예에서, 멀티플렉서 (30)는 스트림 관리 유닛 (60), 비디오 입력 인터페이스 (80), 오디오 입력 인터페이스 (82), 다중화된 스트림 출력 인터페이스 (84), 및 프로그램 지정 정보 테이블들 (88)을 구비한다. 스트림 관리 유닛 (60)은 NAL 단위 구성부 (62), PMT 구성부 (64), 스트림 식별자 (스트림 ID) 록업 유닛 (66), 및 프로그램 식별자 (PID) 지정 유닛 (68)을 구비한다.
- [0079] 도 2의 예에서, 비디오 입력 인터페이스 (80) 및 오디오 입력 인터페이스 (82)는 인코딩된 비디오 데이터 및 인코딩된 오디오 데이터로부터 PES 단위들을 형성하기 위한 개별 패킷화기들을 구비한다. 다른 예들에서, 비디오 및/또는 오디오 패킷화기들은 멀티플렉서 (30) 외부에 있는 유닛 또는 모듈에 구비될 수 있다. 도 2의 예에 관하여, 비디오 입력 인터페이스 (80)는 비디오 인코더 (28)로부터 수신되는 인코딩된 비디오 데이터로부터 PES 패킷들을 형성할 수 있고 오디오 입력 인터페이스 (82)는 오디오 인코더 (26)로부터 수신되는 인코딩된 오디오 데이터로부터 PES 패킷들을 형성할 수 있다.
- [0080] 스트림 관리 유닛 (60)은 PES 패킷들을 비디오 입력 인터페이스 (80) 및 오디오 입력 인터페이스 (82)로부터 수신한다. 각각의 PES 패킷은 PES 패킷이 속한 기본 스트림을 식별하는 스트림 ID를 구비한다. 스트림 ID 록업 유닛 (66)은 PES 패킷이 상응하게 되는 프로그램을 프로그램 지정 정보 테이블들 (88)에 질의함으로써 결정할 수 있다. 다시 말하면, 스트림 ID 록업 유닛 (66)은 수신된 PES 패킷이 어떤 프로그램에 상응하는지를 결정할 수 있다. 각각의 프로그램은 복수의 기본 스트림들을 포함할 수 있는 반면, 일반적으로, 하나의 기본 스트림은 하나의 프로그램에만 상응한다. 그러나, 일부 예들에서, 기본 스트림은 복수의 프로그램들에 포함될 수 있다. 각각의 PES 패킷은 멀티플렉서 (30)로부터 출력되는 복수의 스트림들에 포함될 수 있는데, 다양한 서비스들이 각각 이용가능한 오디오 및 비디오 스트림들의 여러 서브세트들을 구비할 수 있기 때문이다. 따라서, 스트림 ID 록업 유닛 (66)은 PES 패킷이 하나 이상의 출력 스트림들 (예컨대, 하나 이상의 전송 또는 프로그램 스트림들)에 그리고 특히 PES 패킷을 포함하는 출력 스트림들에 포함되어야 할지를 결정할 수 있다.
- [0081] 하나의 예에서, 각각의 기본 스트림은 프로그램에 상응한다. 멀티플렉서 (30)는 각각의 기본 스트림이 특정 프로그램과, 따라서, 프로그램 ID (PID)와 관계되는 것을 보장하는 것을 담당할 수 있다. 멀티플렉서 (30)에 의해 인식되지 않는 스트림 ID (예컨대, 스트림 ID는 프로그램 지정 정보 테이블들 (88)에 저장되지 않음)를 구비하는 PES 패킷이 수신되는 경우, PID 지정 유닛 (68)은 하나 이상의 새로운 엔트리들을 프로그램 지정 정보 테이블들 (88)에 생성하여 새로운 스트림 ID를 비사용 PID와 관계를 가지도록 한다.
- [0082] PES 패킷에 상응하는 프로그램을 결정한 후, NAL 단위 구축부 (62)는 예컨대, PES 패킷의 스트림 ID에 상응하는 프로그램의 PID를 포함하여, NAL 단위 헤더를 가지도록 PES 패킷을 캡슐화함으로써, PES 패킷을 포함하는 NAL 단위를 형성한다. 일부 예들에서, NAL 단위 구축부 (62), 또는 스트림 관리 유닛 (60)의 또 다른 서브-유닛은, 복수의 NAL 단위들을 포함하는 액세스 단위를 형성할 수 있다.
- [0083] PMT 구축부 (64)는 프로그램 맵 테이블들 (PMT들)을 멀티플렉서 (30)의 상응하는 출력 스트림을 위해, 프로그램 지정 정보 테이블들 (88)로부터의 정보를 이용하여 생성한다. 또 다른 예에서, 스트림 관리 유닛 (60)은 멀티플렉서 (30)에 의해 출력되는 프로그램 스트림에 대한 프로그램 스트림 맵들을 생성하기 위해 PSM 구축부를 포함할 수 있다. 일부 예들에서, 멀티플렉서 (30)는 PMT 구축부 (64) 및 PSM 구축부 양쪽 모두를 포함할 수 있고 전송 스트림 및 프로그램 스트림 중 어느 한쪽 또는 양쪽 모두를 출력할 수 있다. 도 2의 예에서, PMT 구축부 (64)는 이 개시물에 의해 설명되는 새로운 기술자들, 예컨대, 동작점 기술자, 뿐만 아니라 임의의 다른 필요한 기술자들 및 PMT에 대한 PMT 데이터를 포함하는 PMT를 구축할 수 있다. PMT 구축부 (64)는 주기적으로, 예컨대, 일정한 기간 후 또는 일정한 양의 데이터가 전송된 후, 전송 스트림을 위해 후속 PMT를 송신할 수 있다. PMT 구축부 (64)는 생성된 PMT들을, 예컨대 PMT를 상응하는 NAL 단위 헤더와 함께 상응하는 PID를 포함하여 캡슐화함으로써, PMT를 포함하는 NAL 단위를 형성하는 NAL 단위 구축부 (62)에 전해 줄 수 있다.
- [0084] PMT 구축부 (64)는 프로그램 각각의 동작점에 대한 데이터 구조, 이를테면 동작점 기술자를 생성할 수 있다. PMT 구축부 (64)에 의해 생성된 데이터 구조는 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 랜더링 능력을 기술하는 랜더링 능력 값, 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링할 수 있다.



- [0085] 예를 들어, PMT 구축부 (64) 는 동작점을 위해 디스플레이될 다수의 뷰들 및 동작점들의 뷰들에 대한 프레임 레이트를, 프로그램 지정 정보 테이블들 (88) 에 의해 저장된 정보 또는 비디오 인코더 (28) 로부터 비디오 입력 인터페이스 (80) 를 경유하여 수신된 정보에 기초하여 결정할 수 있다. PMT 구축부 (64) 는 동작점들의 뷰들의 수 및 뷰들에 대한 프레임 레이트 중 어느 한쪽 또는 양쪽 모두를 데이터 구조의 렌더링 능력 값을 이용하여 시그널링할 수 있다.
- [0086] PMT 구축부 (64) 는 동작점을 위해 디코딩될 뷰들의 수 및 동작점들의 뷰들에 상응하는 프로파일에 대한 레벨 값을 결정할 수도 있다. 예를 들어, PMT 구축부 (64) 는 처리될 것이 필요한 매크로블록들의 수, 디코딩된 화상 버퍼 사이즈, 코딩된 화상 버퍼 사이즈, 세로방향 움직임 벡터 범위, 2개의 연속하는 매크로블록들 당 최대 움직임 벡터 수, 및/또는 B-블록이 8×8개 화소들 미만의 서브-매크로블록 파티션들을 가질 수 있는지를 결정할 수 있고, 이들 결정들을 이용하여 동작점에 대한 레벨을 결정할 수 있다. PMT 구축부 (64) 는 이 정보를 비디오 인코더 (28) 로부터 비디오 입력 인터페이스 (80) 를 경유하여 수신할 수 있다. 그 다음 PMT 구축부 (64) 는 디코딩할 뷰들의 수 및/또는 프로파일의 레벨 값을 동작점에 대한 디코딩 능력 값을 이용하여 나타낼 수 있다.
- [0087] PMT 구축부 (64) 는 추가로 동작점에 대한 비트레이트 값을 결정하고 비트레이트 값을 데이터 구조 내에 인코딩할 수 있다. 비트레이트 값은 동작점에 대한 평균 비트레이트 또는 최대 비트레이트에 상응할 수 있다. PMT 구축부 (64) 는 동작점에 대한 비트레이트를 계산할 수 있거나 또는 비트레이트의 지시 (indication) 를 비디오 인코더 (28) 로부터 수신할 수 있다.
- [0088] 다중화된 스트림 출력 인터페이스 (84) 는 스트림 관리 유닛 (60) 으로부터 하나 이상의 NAL 단위들 및/또는 액세스 단위들, 예컨대, PES 패킷들 (예컨대, 오디오 또는 비디오 데이터) 을 포함하는 NAL 단위들 및/또는 PMT를 포함하는 NAL 단위들을 수신할 수 있다. 일부 예들에서, 다중화된 스트림 출력 인터페이스 (84) 는 NAL 단위들이 스트림 관리 유닛 (60) 으로부터 수신된 후, 하나 이상의 NAL 단위들로부터 공통 시간적 위치에 상응하는 액세스 단위들을 형성할 수 있다. 다중화된 스트림 출력 인터페이스 (84) 는 NAL 단위들 또는 액세스 단위들을 상응하는 전송 스트림 또는 프로그램 스트림의 출력으로서 전송한다. 다중화된 스트림 출력 인터페이스 (84) 는 PMT 구축부 (64) 로부터 데이터 구조를 수신할 수 있고 이 데이터 구조를 비트스트림의 부분으로서 구비할 수 있다.
- [0089] 도 3은 프로그램 지정 정보 테이블들 (88) 의 예시적인 집합을 도시하는 블록도이다. 전송 패킷이 속하는 기본 스트림은 전송 패킷의 PID 값에 기초하여 결정될 수 있다. 디코더가 수신된 데이터를 적절히 디코딩하기 위하여, 디코더에는 기본 스트림들이 각각의 프로그램에 속하는지를 결정할 수 있는 것이 필요하다. 프로그램 지정 정보는 프로그램 지정 정보 테이블 (88) 에 포함된 것으로, 프로그램들 및 구성요소인 기본 스트림들 사이의 관계들을 명확히 지정할 수 있다. 도 3의 예에서, 프로그램 지정 정보 테이블들 (88) 은 네트워크 정보 테이블 (100), 조건부 액세스 테이블 (102), 프로그램 액세스 테이블 (104), 및 프로그램 맵 테이블 (106) 을 구비한다. 도 3의 예의 경우, 출력 스트림이 MPEG-2 전송 스트림을 포함하는 것으로 가정된다. 대체 예에서, 출력 스트림은 프로그램 스트림을 포함할 수 있는데, 이 경우 프로그램 맵 테이블 (106) 은 프로그램 스트림 맵에 의해 대체될 수 있다.
- [0090] MPEG-2 시스템들의 사양서는 전송 스트림으로 운반되는 모든 프로그램이 그것에 관련된 프로그램 맵 테이블, 이를테면 프로그램 맵 테이블 (106) 을 가진다고 명시한다. 프로그램 맵 테이블 (106) 은 프로그램 및 프로그램이 포함하는 기본 스트림들에 관한 세부사항들을 구비할 수 있다. 하나의 예로서, 프로그램 번호 3으로서 식별되는 프로그램은, PID 33을 갖는 비디오 기본 스트림, PID 57을 갖는 영어 오디오 스트림, 및 PID 60을 갖는 중국어 오디오 스트림을 담을 수 있다. 하나를 넘는 프로그램을 포함하는 것이 PMT에 허용된다.
- [0091] MPEG-2 시스템들의 사양서에 의해 명시되는 기본 프로그램 맵 테이블은 MPEG-2 시스템들의 사양서 내에서 명시되는 많은 기술자들, 예컨대, 기술자들 (108) 로 장식된다. 기술자들 (108) 은 MPEG-2 시스템들의 사양서의 지정된 기술자들 중의 어느 것 또는 전부를 구비할 수 있다. 대체로, 기술자들, 이를테면 기술자들 (108) 은 프로그램 또는 그것의 구성요소인 기본 스트림들 또는 서브-비트스트림들에 관한 추가 정보를 수송한다. 기술자들은 비디오 인코딩 매개변수들, 오디오 인코딩 매개변수들, 언어 ID, 팬-및-스캔(pan-and-scan) 정보, 조건부 액세스 세부사항들, 저작권 정보, 또는 다른 그러한 정보를 구비할 수 있다. 브로드캐스터 또는 다른 사용자는 부가적인 사실 기술자들을 정의할 수 있다.
- [0092] 이 개시물은 MPEG-2 시스템들을 준수하는 비트스트림에서 동작점의 특성들을 기술하는 동작점 기술자들을 제공한다. 기술자들 (108) 은 상응하는 비트스트림 각각의 동작점에 대한 동작점 기술자들을 구비할 수 있다.

도 3에 보인 바와 같이, 기술자들 (108) 은 MVC 확장 기술자들 (110), 계층구조(hierarchy) 기술자 (112), 및 동작점 기술자들 (114) 을 구비한다. 동작점 기술자들 (114) 각각은 비트스트림의 특정 동작점에 상응할 수 있고, 동작점에 대해, 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 랜더링 능력을 기술하는 랜더링 능력 값, 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 동작점의 비트레이트를 기술하는 비트레이트 값을 시그널링할 수 있다. 비디오에 관련된 성분 기본 스트림들에는, 계층적으로 코딩된 비디오, 오디오, 및 사실 스트림들의 성분들을 담고 있는 프로그램 요소들을 식별하는 정보를 제공하는 계층구조 기술자도 있다.

[0093] 아래의 표 1은 MVC 확장 기술자들 (110) 에 포함된 데이터의 하나의 예를 제공한다. 표 1에 보인 다양한 필드들 및 이 필드들의 비트 값이들은 단지 하나의 예이다. 하나의 예에서, 각각의 MVC 비디오 서브-비트스트림은 MVC 확장 기술자들 (110) 중 상응하는 MVC 비디오 서브-비트스트림의 특성들을 명시하는 상응하는 하나의 MVC 확장 기술자에 관련된다. MVC 비디오 서브-비트스트림은 다른 MVC 비디오 서브-비트스트림들과 어셈블링되는 것이 필요할 수 있다. 다시 말하면, 특정 서브-비트스트림을 디코딩하고 표현하기 위하여, 클라이언트 디바이스는 2개의 서브-비트스트림들을 구비하는 공통의 비트스트림의 다른 서브-비트스트림으로부터 비디오 데이터를 추출하고 디코딩할 필요가 있을 수 있다.

표 1

표 1 - MVC 확장 기술자

[0094]

구문	비트 수	연상기호
MVC_extension_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
average_bit_rate	16	uimsbf
maximum_bitrate	16	uimsbf
예약됨(reserved)	4	bslbf
view_order_index_min	10	bslbf
view_order_index_max	10	bslbf
temporal_id_start	3	bslbf
temporal_id_end	3	bslbf
no_sei_nal_unit_present	1	bslbf
예약됨	1	bslbf
}		

[0095] 표 1의 예에서, 기술자 태그 필드는 MPEG-2 시스템들의 표준에 의해 언급된 바와 같이, 모든 기술자 내에 특이 그 기술자를 식별하기 위해 포함된 8비트 기술자 태그 필드에 상응할 수 있다. MPEG-2 시스템들의 표준은 일정한 기술자 태그들을 정의하고 다른 기술자 태그 값들, 예컨대, 36 내지 63의 값들을 "예약됨 (reserved)"이라고 표시한다. 그러나 MPEG-2 시스템들 표준에 대한 개정안 4는 MVC 확장 기술자를 "49"로 설정하는 것을 제안하는데, 이것은 MPEG-2 시스템들의 사양서에서 명시된 바와 같이, 예약된 기술자 태그들 중의 하나에 상응한다. 이 개시물은, 그래서, MVC 확장 기술자들 (110) 의 descriptor\_tag의 값을 "49"의 값으로 설정하는 것을 제안한다.

[0096] 다시, 기술자 길이 필드는 MPEG-2 시스템들의 표준에 의해 언급된 바와 같이, 모든 기술자에도 포함된 8비트 기술자 길이 필드에 상응할 수 있다. 멀티플렉서 (30) 는 기술자 길이 필드의 값을 MVC 확장 기술자들 (110) 중 기술자 길이 필드 바로 뒤의 상응하는 하나의 바이트들의 수와 동일하도록 설정할 수 있다. MVC 확장 기술자의 길이가 바뀌지 않기 때문에, 멀티플렉서 (30) 는 MVC 확장 기술자들 (110) 각각에 대한 기술자 길이 필드의 값을 8의 값으로 설정하여, 기술자 길이 필드 뒤에 8개 비트들의 정보가 존재함을 나타낸다.

[0097] 평균 비트 레이트 필드는 리어셈블링된 AVC 비디오 스트림의 평균 비트 레이트를 초당 킬로비트 단위로 나타내는 16비트 필드를 포함할 수 있다. 다시 말하면, 평균 비트 레이트 필드는 MVC 확장 기술자들 (110) 중 하나에 상응하는 전송 스트림 또는 프로그램 스트림의 구성하는 부분들로부터 비디오 스트림이 어셈블링될 때의 비디오 스트림에 대한 평균 비트 레이트를 기술한다. 일부 예들에서, 멀티플렉서 (30) 는 평균 비트 레이트 필드의 값을 평균 비트 레이트가 MVC 확장 기술자들 (110) 중의 하나에 의해 지시되지 않음을 나타내는 영(0)으로 설정할 수 있다.

- [0098] 최대 비트 레이트 필드는 리어셈블링된 AVC 비디오 스트림의 최대 비트 레이트를 초당 킬로비트 단위로 나타내는 16비트 필드를 포함할 수 있다. 다시 말하면, 최대 비트 레이트 필드는 MVC 확장 기술자들 (110) 중 하나에 상응하는 전송 스트림 또는 프로그램 스트림의 구성하는 부분들로부터 비디오 스트림이 어셈블링될 때의 비디오 스트림에 대한 최대 비트 레이트를 기술한다. 일부 예들에서, 멀티플렉서 (30) 는 최대 비트 레이트 필드의 값을 최대 비트 레이트가 MVC 확장 기술자들 (110) 중의 하나에 의해 지시되지 않음을 나타내는 영(0)으로 설정할 수 있다.
- [0099] 뷰 순서 색인 최소 필드는 관련된 MVC 비디오 서브-비트스트림에 들어 있는 모든 NAL 단위들의 뷰 순서 색인의 최소 값을 지시하는 10 비트 필드를 포함할 수 있다. 마찬가지로, 뷰 순서 색인 최대 필드는 관련된 MVC 비디오 서브-비트스트림에 들어 있는 모든 NAL 단위들의 뷰 순서 색인의 최대 값을 지시하는 10 비트 필드이다.
- [0100] 시간적 ID 시작 필드는 관련 MVC 비디오 서브-비트스트림에 들어 있는 모든 NAL 단위들의 NAL 단위 헤더 구문 요소의 temporal\_id의 최소 값을 나타내는 3 비트 필드를 포함할 수 있다. 다시 말하면, 시간적 ID 값은 각각의 NAL 단위에 대한 헤더에 구비된다. 대체로, 시간적 ID 값은 비교적 큰 시간적 ID 값들이 더 높은 프레임레이트들에 상응하는 경우의 특정 프레임레이트에 상응한다. 예를 들어, 시간적 ID에 대한 '0'이란 값은 초당 15개 프레임들의 프레임레이트 (fps) 에 상응할 수 있고, 시간적 ID에 대한 '1'이란 값은 30 fps의 프레임레이트에 상응할 수 있다. 이런 방식으로, 0이란 시간적 ID를 갖는 모든 화상들을 이 예에서 한 집합으로 모으는 것은 15 fps의 프레임레이트를 갖는 비디오 세그먼트를 형성하는데 사용될 수 있는 반면, 0이란 시간적 ID를 갖는 모든 화상들 및 1이란 시간적 ID를 갖는 모든 화상들을 다른 집합으로 모으는 것은 30 fps의 프레임레이트를 갖는 상이한 비디오 세그먼트를 형성하는데 사용될 수 있다. 멀티플렉서 (30) 는 MVC 비디오 서브-비트스트림의 NAL 단위들의 모두의 최소 시간적 ID를 결정하고 시간적 ID 시작 필드의 값을 이 결정된 최저 시간적 ID 값과 동일하도록 설정한다.
- [0101] 시간적 ID 끝 필드는 관련 MVC 비디오 서브-비트스트림에 들어 있는 모든 NAL 단위들의 NAL 단위 헤더 구문 요소의 시간적 ID의 최대 값을 나타내는 3 비트 필드를 포함할 수 있다. 따라서, 멀티플렉서 (30) 는 MVC 비디오 서브-비트스트림의 NAL 단위들의 모두의 최대 시간적 ID를 결정하고 시간적 ID 시작 필드의 값을 이 결정된 최대 시간적 ID 값과 동일하도록 설정한다.
- [0102] no SEI NAL 단위 존재(no\_sei\_nal\_unit\_present) 필드는, '1'로 설정될 때 추가 향상 정보(supplemental enhancement information) NAL 단위들이 관련 MVC 비디오 서브-비트스트림에 존재하지 않음을 나타내는 1비트 플래그를 포함할 수 있다. 멀티플렉서 (30) 는 하나 이상의 추가 향상 정보 NAL 단위들이 비트스트림 내에 위치되어 있는지를 결정할 수 있고 no SEI NAL 단위들이 비트스트림 내에 존재하는 경우 no SEI NAL 단위 존재 필드의 값을 '1'의 값으로 설정하지만, 적어도 하나의 SEI NAL 단위가 비트스트림에 존재하는 경우 no SEI NAL 단위 존재 필드의 값을 '0'의 값으로 설정할 수 있다.
- [0103] 아래의 표 2는 MVC 계층구조 기술자들 (112) 에 포함된 데이터의 하나의 예를 제공한다. MPEG-2 시스템들에서, 계층구조 기술자는 내장형(embedded) 비디오 프로그램 스트림을 담고 있는 비디오의 프로그램 스트림을 위해 정의될 수 있다. 표 2에 보인 다양한 필드들 및 이 필드들의 비트 깊이들은 하나의 예로서 제공된다.  
hierarchy\_layer\_index 값은 현재의 프로그램 스트림의 계층 인덱스를 식별하고, hierarchy\_embedded\_layer\_index 값은 종속 계층을 식별한다. MVC 디자인에서, 프로그램 스트림은 다른 프로그램 스트림에 계층구조 기술자를 이용하여 종속될 수 있다. 다시 말하면, 프로그램 스트림들 간의 의존도(종속도)들은 계층 기술자에 포함된 데이터에 기초하여 결정될 수 있다.

**표 2**

**표 2 - 계층구조 기술자**

[0104]

구문	비트 수	연상기호
hierarchy_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
예약됨	1	bslbf
temporal_scalability_flag	1	bslbf
spatial_scalability_flag	1	bslbf
quality_scalability_flag	1	bslbf

hierarchy_type	4	Uimsbf
예약됨	2	bslbf
hierarchy_layer_index	6	uimsbf
Tref_present_flag	1	bslbf
예약됨	1	bslbf
hierarchy_embedded_layer_index	6	uimsbf
예약됨	2	bslbf
hierarchy_channel	6	uimsbf
}		

- [0105] 위에서 언급된 바와 같이, MPEG-2 시스템들의 사양서는 각각의 기술자가 기술자 태그 필드 및 기술자 길이 필드를 구비함을 명시한다. 따라서, 계층구조 기술자 (112) 는 기술자 태그 필드 및 기술자 길이 필드를 구비한다. MPEG-2 시스템들의 사양서를 따라서, 멀티플렉서 (30) 는 기술자 태그 필드의 값을 계층구조 기술자 (112) 를 위해 "4"의 값으로 설정할 수 있다.
- [0106] 계층구조 기술자 (112) 의 길이는 연역적으로 결정될 수 있는데, 계층구조 기술자 (112) 각각의 인스턴스가 동일한 양의 데이터를 구비하여야 하기 때문이다. 하나의 예에서, 멀티플렉서 (30) 는 기술자 길이 필드의 값을, 기술자 길이 필드의 끝을 뒤따르는 계층구조 기술자 (112) 의 인스턴스에서 4 바이트를 나타내는 4의 값으로 설정할 수 있다.
- [0107] 계층 유형 필드는 관련 계층구조의 계층(hierarchy layer) 및 그것의 계층구조 내장(embedded) 계층 사이의 계층구조적 관계를 기술한다. 하나의 예에서, 멀티플렉서 (30) 는 계층구조 유형 필드의 값을 예컨대 아래의 표 3에 의해 설명된 바와 같은 계층구조적 관계에 기초하여 설정한다. 하나의 예로서, 확장성이 하나를 넘는 차원에 적용될 때, 멀티플렉서 (30) 는 계층 유형 필드를 "8"의 값 ("표 3에 보인 바와 같은 "결합 확장성 (Combined Scalability)") 으로 설정할 수 있고, 멀티플렉서 (30) 는 시간적 확장성 플래그 필드, 공간적 확장성 플래그 필드, 및 품질 확장성 플래그 필드의 값들을, 개별 스트림들의 PES 패킷들 및 PES 패킷 헤더들로부터 검색된 데이터에 따라 설정한다. 대체로, 멀티플렉서 (30) 는 다양한 뷰들 및/또는 오디오 데이터 스트림들에 상응하는 상이한 스트림들 사이의 의존도 (중속도) 들을 결정할 수 있다. 멀티플렉서 (30) 는 항상 계층을 포함하는 중속 스트림이 공간적 계층, 신호 대 잡음 (SNR) 항상 계층, 품질 항상 계층, 또는 다른 유형의 항상 계층인지를 결정할 수도 있다.
- [0108] 또 다른 예로서, MVC 비디오 서브-비트스트림들의 경우, 멀티플렉서 (30) 는 계층구조 유형 필드를 '9'의 값 (표 3에 보인 바와 같은 "MVC") 으로 설정할 수 있고 확장성 플래그 필드, 공간적 확장성 플래그 필드, 및 품질 확장성 플래그 필드 각각의 값들을 '1'로 설정할 수 있다. 또 다른 예로서, MVC 기반 뷰 서브-비트스트림들의 경우, 멀티플렉서 (30) 는 계층구조 유형 필드의 값을 '15'의 값으로 설정할 수 있고 확장성 플래그 필드, 공간적 확장성 플래그 필드, 및 품질 확장성 플래그 필드 각각의 값들을 '1'로 설정할 수 있다. 또 다른 예로서, 접두 MVC 서브-비트스트림의 경우, 멀티플렉서 (30) 는 계층구조 유형 필드를 '14'의 값으로 설정할 수 있고 확장성 플래그 필드, 공간적 확장성 플래그 필드, 및 품질 확장성 플래그 필드를 '1'로 설정할 수 있다.
- [0109] 계층구조 계층 색인 필드는 관련 프로그램 요소의 고유 색인을 정의하는 6비트 필드를 코딩 계층 계층구조들의 테이블에 포함할 수 있다. 색인들은 단일 프로그램 정의 내에서 고유할 수 있다. 비디오 스트림들이 ITU-T Rec. H.264 | ISO/IEC 14496-10의 부록 G에 정의된 하나 이상의 프로파일들을 준수하는 AVC의 비디오 서브-비트스트림들의 경우, 동일한 액세스 단위의 비디오 서브-비트스트림들의 관련 SVC 의존도 표현들이 hierarchy\_layer\_index가 증가하는 순서로 리어샘플링된다면 비트스트림 순서가 올바르게 될 것인 방식으로 프로그램 요소 색인은 지정된다. ITU-T Rec. H.264 | ISO/IEC 14496-10의 부록 H에 정의된 하나 이상의 프로파일들을 준수하는 AVC 비디오 스트림들의 MVC 비디오 서브-비트스트림들의 경우, 접두 MVC 서브-비트스트림에 대한 계층구조 기술자에 명시된 hierarchy\_layer\_index 값보다 그 값들이 크게 되는 방식으로 프로그램 요소 색인은 지정된다.
- [0110] 계층구조 내장 계층 색인 필드는 계층구조 기술자 (112) 의 상응하는 인스턴스에 관련된 기본 스트림을 디코딩하기 전에 액세스되는 것이 필요한 프로그램 요소의 계층구조 테이블 색인을 정의하는 6비트 필드를 포함할 수 있다. 이 개시물은 계층구조 유형 필드가 15의 값 (다시 말하면, 기본 계층에 상응하는 값) 일 때를 위해 계층구조 내장 계층 색인 필드에 대한 값을 정의하지 않고 남겨둔다.
- [0111] 계층구조 채널 필드는 관련 프로그램 요소를 위해 의도된 채널 번호를 나타내는 6비트 필드를 전송 채널들의 순



서집합(ordered set) 내에 포함할 수 있다. 가장 강건한(robust) 전송 채널은 전체 전송 계층구조 정의에 관하여 계층구조 채널 필드의 최저 값에 의해 정의된다. 주어진 계층구조 채널이 동일한 시간에 수 개의 프로그램 요소들에 지정될 수 있음에 주의한다.

[0112] 표 1 및 2의 예약됨(reserved) 필드들은 장래의 표준들의 개발에 의한 장래의 사용을 위해 예약된다. 이 개시물의 기법들은 현 시점에서는 예약된 필드들의 값들에 의미론적 의미를 지정하는 것을 제안하지 않는다.

[0113] 아래의 표 3은 위에서 설명된 계층구조 유형 필드에 대한 잠정적인 값들을 도시한다:

**표 3**

**표 3 - 계층구조 유형 필드 값들**

[0114]

값	설명
0	예약됨
1	공간적 확장성
2	SNR 확장성
3	시간적 확장성
4	데이터 파티셔닝
5	확장 비트-스트림
6	사실 스트림
7	멀티-뷰 프로파일
8	결합 확장성(Combined Scalability)
9	MVC 비디오 서브-비트스트림
10-3	예약됨
14	접두 MVC 서브-비트스트림
15	기본 계층, 또는 MVC 기본 뷰 서브-비트스트림, 또는 MVC의 AVC 비디오 서브-비트스트림

[0115] 일부 예들에서, 계층구조 기술자 (112)는 중분형 서브-비트스트림 및 내장형 서브-비트스트림들에 의해 시그널링되는 MVC 서브-비트스트림을 시그널링하는데 사용될 수 있다. 내장형 서브-비트스트림들은 hierarchy\_embedded\_layer\_index에 상응하는 직접 종속 서브-비트스트림 및 이 직접 종속 서브-비트스트림의 모든 내장형 서브-비트스트림들을 구비한다. 이 개시물에서, 명시적으로 담겨 있는 뷰들은 향상 뷰들(enhance views)이라 불리는 반면, 내장되는 뷰들은 종속 뷰들(dependent views)이라고 불린다.

[0116] 멀티플렉서 (30)의 출력이 프로그램 스트림을 포함하는 일 예에서, 프로그램 지정 정보 테이블들 (88)은 프로그램 스트림 맵 (PSM)을 구비할 수 있다. PSM은 상응하는 프로그램 스트림 내의 기본 스트림들의 설명 및 기본 스트림들의 서로에 대한 관계들을 제공한다. 일부 예들에서, 프로그램 스트림 맵은 전송 스트림에 상응할 수도 있다. 상응하는 전송 스트림으로 운반될 때, PSM 구조는 수정되지 않아야 한다. 멀티플렉서 (30)는 PES 패킷의 스트림 id 값을 0xBC, 다시 말하면, 이진수 값 10111100, 또는 십진수 값 188에 해당하는 16진수 값 BC로 설정함으로써 PSM이 PES 패킷 내에 존재함을 나타낼 수 있다.

[0117] 멀티플렉서 (30)는 전송 스트림에서 이용가능한 모든 프로그램들의 완전한 목록을 프로그램 관련 테이블 (104)에 유지한다. 멀티플렉서 (30)는 또한 프로그램 관련 테이블들을 NAL 단위들에 내장시킬 수 있다. 멀티플렉서 (30)는 NAL 단위에 PID 값 0을 지정함으로써 NAL 단위가 프로그램 관련 테이블을 구비함을 나타낼 수 있다. 멀티플렉서 (30)는 각각의 프로그램을 상응하는 프로그램 맵 테이블을 담고 있는 전송 패킷들의 PID 값과 함께, 프로그램 관련 테이블 (104) 내에 열거할 수 있다. 위에서 언급된 동일한 예를 이용하여, 프로그램 번호 3의 기본 스트림들을 지정하는 예의 프로그램 맵 테이블은 1001의 PID를 가지고, 다른 PMT는 1002의 다른 PID를 가진다. 이것 또는 유사한 집합들의 정보는 프로그램 관련 테이블 (104)에 구비될 수 있다.

[0118] 프로그램 지정 정보 테이블들 (88)은 또한 네트워크 정보 테이블 (NIT)(100) 및 조건부 액세스 테이블 (CAT)(102)을 구비한다. PAT에 지정된 바와 같이, 프로그램 번호 영(0)은 특수한 의미를 가진다. 특히, 프로그램 번호 영은 네트워크 정보 테이블 (100)에 대한 길을 알려주는데 사용될 수 있다. 이 테이블은 옵션적이고, 제시될 때, 이 테이블은 전송 스트림을 운반하는 물리적 네트워크에 관한 정보 이블테면 채널 주파수들, 위성 중계기 세부사항들, 변조 특성들, 서비스 발신자, 서비스 이름, 및 이용가능한 대체 네트워크들의 세부사항들을 제공할 수 있다.

- [0119] 전송 스트림 내의 어떤 기본 스트림들이 스크램블된다면, 조건부 액세스 테이블 (102) 이 제시되어야 한다. 조건부 액세스 테이블 (102) 은 사용중인 스크램블링 시스템 (들) 의 세부사항들을 제공하고 조건부 액세스 관리 및 자격 정보를 담고 있는 전송 패킷들의 PID 값들을 제공한다. 이 정보의 형식은 MPEG-2 시스템들의 표준 내에 명시되지 않는다.
- [0120] 도 4는 동작점 기술자들 (114)(도 3) 중의 하나에 포함될 수 있는 데이터의 예시적인 집합을 도시하는 블록도이다. 도 4의 예에서, 동작점 기술자 (118) 는 기술자 태그 필드 (120), 기술자 길이 필드 (122), 프레임 레이트 필드 (124), 디스플레이 뷰들의 수 필드 (126), 디코딩 뷰들의 수 필드 (128), 뷰 식별자 필드들 (130), 평균 비트레이트 필드 (132), 최대 비트레이트 필드 (134), 시간적 식별자 필드 (136), 및 예약된 트레일링 비트 필드들 (138) 을 포함한다.
- [0121] 도 4의 예에서, 프레임 레이트 필드 (124) 및 디스플레이 뷰들의 수 필드 (126) 는 예시적인 렌더링 능력 값에 해당하며, 디코딩 뷰들의 수 필드 (128) 는 예시적인 디코딩 능력 값에 해당하고, 평균 비트레이트 필드 (132) 및 최대 비트레이트 필드 (134) 는 예시적인 비트레이트 값에 해당한다. 동작점 기술자 (118) 는 동작점의 특성들, 이를테면 렌더링 능력, 디코딩 능력, 및 비트레이트를 시그널링하는데 사용될 수 있는 데이터 구조의 하나의 예일 뿐이다. 아래의 도 5 및 6은 이들 특성들을 시그널링하는 동작점 기술자들의 대체 예들을 제공한다.
- [0122] 위에서 설명된 바와 같이, MPEG-2 시스템들의 사양서는 각 필드가 8비트인 기술자 태그 필드 및 기술자 길이 필드를 각각의 기술자가 가짐을 명시한다. 그래서, 멀티플렉서 (30)(도 1) 는 MVC 동작점 기술자를 나타내는 값을 기술자 태그 필드 (120) 에 지정할 수 있다. 멀티플렉서 (30) 는 동작점에 대한 뷰들의 수 및 동작점 기술자에 대한 예약된 비트들의 수를 결정한 다음, 기술자 길이 필드 (122) 를 뒤따르는 동작점 기술자 (118) 의 바이트 길이를 계산한다. 멀티플렉서 (30) 는 동작점 기술자 (118) 를 인스턴스화할 때, 이 계산된 길이 값을 기술자 길이 필드 (122) 에 지정할 수 있다.
- [0123] 프레임 레이트 필드 (124) 는 리어셈블링된 AVC 비디오 스트림의 최대 프레임 레이트를 256 초당 프레임으로 나타내는 16비트 필드를 포함할 수 있다. 다시 말하면, 멀티플렉서 (30) 는 256초 기간의 최대 프레임 레이트를 계산하여 프레임 레이트 필드 (124) 의 값을 설정할 수 있다. 일부 예들에서, 256으로 나누는 것은 부동소수점 값을 정수 값으로 변환하는 결과를 가져올 수 있다. 다른 예들에서, 256 초 외의 기간들이 사용될 수 있다. 프레임 레이트 필드 (124) 에 관하여 설명된 256 초 기간은 동작점의 최대 프레임 레이트가 계산될 수 있는 하나의 잠정적인 예일 뿐이다.
- [0124] 디스플레이 뷰들의 수 필드 (126) 는 리어셈블링된 AVC 비디오 스트림의 출력 대상이 되는 뷰들의 수의 값을 나타내는 10비트 필드를 포함할 수 있다. 대체로, 디스플레이 뷰들의 수 필드 (126) 는 디스플레이될 뷰들의 수를 상응하는 동작점을 위해 나타낸다. 상이한 디스플레이들이 상이한 수의 뷰들을 디스플레이할 수 있기 때문에, 클라이언트 디바이스는 디스플레이 뷰들의 수 필드 (126) 의 값을 이용하여 클라이언트 디바이스에 대한 디스플레이상에 가능한 한 많이 디스플레이될 뷰들을 가지는 동작점을 선택할 수 있다. 예를 들어, 클라이언트 디바이스가 4개의 뷰들을 디스플레이할 능력이 있으면, 클라이언트 디바이스는 4개의 뷰들이 상응하는 동작점에 대해 디스플레이될 것임을 나타내는 값을 갖는 디스플레이 뷰들의 수 필드으로써 동작점을 선택할 수 있다. 따라서, 디스플레이 뷰들의 수 필드 (126) 는 렌더링 능력 값의 부분으로서 포함될 수 있다. 비슷하게, 멀티플렉서 (30) 는 디스플레이 뷰들의 수 필드 (126) 의 값을 동작점을 위해 디스플레이될 뷰들의 수에 따라 설정할 수 있다.
- [0125] 디코딩 뷰들의 수 필드 (128) 는 리어셈블링된 AVC 비디오 스트림을 디코딩하기 위해 요구되는 뷰들의 수의 값을 나타내는 10비트 필드를 포함할 수 있다. 이 값은 디스플레이 뷰들의 수 필드 (126) 에 의해 나타내어지는 디스플레이될 뷰들의 수와는 상이할 수 있다. 이는 뷰 종속성들 때문에 디코딩을 위해 요구되지만 실제로는 디스플레이되지 않는 뷰들로 인해 생긴다.
- [0126] 일 예로서 간략히 도 7을 참조하면, 뷰들 (S0 및 S1) 은 동작점을 위해 디스플레이될 뷰들일 수 있다. 뷰 S0는 임의의 다른 뷰들의 디코딩 없이 직접 디코딩될 수 있다. 그러나, 뷰 S1을 디코딩하기 위해서는, 뷰 S2 또한 디코딩되어 하는데, 뷰 S1이 뷰 S2를 참조하는 예측 데이터를 구비하기 때문이다. 그러므로, 이 예에서, 디스플레이 뷰들의 수 필드 (126) 는 2의 값을 가질 것이지만, 디코딩 뷰들의 수 필드 (128) 는 3의 값을 가질 것이다. 일부 예들에서, 디스플레이될 뷰는 하나 이상의 다른 뷰들로부터 보간될 수 있어서, 디스플레이될 뷰들의 수는 디코딩될 뷰들의 수보다 클 수 있다. 다시 말하면, 기본 뷰 및 깊이 정보를 이용하여, 비디오 디코더 (48)(도 1) 는 제 2의 뷰를 보간할 수 있다. 비디오 디코더 (48) 는 깊이 정보를 계산하여 새

로운 뷰를 보간하기 위해 둘 이상의 뷰들을 이용할 수 있거나, 또는 비디오 디코더 (48) 는 뷰에 대한 깊이 정보를 소스 디바이스 (20) 로부터 수신할 수 있다.

[0127] 디코딩 뷰들의 수 필드 (128) 는 클라이언트 디바이스의 디코더 (이를테면 목적지 디바이스 (40) 의 비디오 디코더 (48)) 가 디코딩 뷰들의 수 필드 (128) 의 값과 동일한 수의 뷰들을 디코딩할 수 있어야한다는 점에서 디코딩 능력 값에 상응할 수 있다. 따라서, 클라이언트 디바이스는 클라이언트 디바이스의 비디오 디코더가 디코딩할 수 있는 뷰들의 수를 나타내는 디코딩 뷰들의 수 필드를 갖는 동작점을 선택할 수 있다.

[0128] 도 4의 동작점 기술자 (118) 역시 뷰 식별자 필드들 (130) 을 구비한다. 뷰 식별자 필드들 (130) 각각은 리어셈블링된 AVC 비디오 비트스트림에 들어 있는 NAL 단위들의 view\_id의 값을 나타내는 10비트 필드를 포함할 수 있다. 그래서, 동작점에 대한 각각의 디스플레이되는 뷰들의 뷰 식별자들은 뷰 식별자 필드들 (130) 을 이용하여 시그널링된다. 다시 말하면, 뷰 식별자 필드들 (130) 의 뷰 식별자들은 디스플레이되는 뷰들에 상응한다. 그래서, 디코딩되었지만 디스플레이되지 않는 뷰들은 도 4의 예에서 뷰 식별자 필드들 (130) 에 의해 시그널링되지 않는다.

[0129] 평균 비트레이트 필드 (132) 는 리어셈블링된 AVC 비디오 스트림의 평균 비트 레이트를 초당 킬로비트로 나타내는 16비트 필드를 포함할 수 있다. 0으로 설정될 때, 평균 비트 레이트는 명시되지 않는다. 다시 말하면, 평균 비트레이트 필드 (132) 에 대한 영(0)의 값은 평균 비트레이트 필드 (132) 가 리어셈블링된 AVC 비디오 스트림의 평균 비트레이트를 결정하는데 사용되지 않아야 한다는 것을 의미한다.

[0130] 최대 비트레이트 필드 (134) 는 리어셈블링된 AVC 비디오 스트림의 최대 비트 레이트를 초당 킬로비트로 나타내는 16비트 필드를 포함할 수 있다. 0으로 설정될 때, 최대 비트 레이트는 명시되지 않는다. 다시 말하면, 최대 비트레이트 필드 (134) 에 대한 값이 영으로 설정될 때, 최대 비트레이트 필드 (134) 는 리어셈블링된 AVC 비디오 스트림의 최대 비트레이트를 결정하는데 사용되지 않아야 한다.

[0131] 시간적 식별자 필드 (136) 는 리어셈블링된 AVC 비디오 스트림의 프레임 레이트에 상응하는, temporal\_id의 값을 나타내는 3비트 필드를 포함할 수 있다. 다시 말하면, temporal\_id는 위에서 논의된 바와 같이 리어셈블링된 AVC 비디오 스트림의 프레임 레이트를 결정하는데 사용될 수 있다.

[0132] 예의 동작점 기술자 (118) 는 또한 예약된 트레일링 비트 필드들 (138) 을 구비한다. 하나의 예에서, 예컨대, 아래의 테이블 4에 보인 바와 같이, 예약된 트레일링 비트들의 수는 부가적인 시그널링 및 동작점 기술자 (118) 의 패딩(padding) 양쪽 모두를 위해 사용될 수 있어서, 동작점 기술자 (118) 는 바이트 경계에서 끝난다. 예를 들어, 위에서 논의된 바와 같이, 동작점 기술자 (118) 는 각각의 디스플레이되는 뷰의 뷰 식별자를 나타내는데 10비트를 사용할 수 있다. 뷰 식별자들 및 예약된 트레일링 비트들의 위해 사용되는 비트들 외의 비트들의 고정된 수는 이 예에서는 87이다. 그래서, 동작점 기술자 (118) 가 바이트 경계에서 끝나는 (다시 말하면, 8로써 균등하게 나눌 수 있는 비트 수를 가지는) 것을 보장하기 위하여, 멀티플렉서 (30) 는 다음 수학적식에 따라 트레일링 비트들의 수를 추가할 수 있다:

[0133] 트레일링 비트 수 = (1 + 6\*num\_display\_views) % 8

[0134] 여기서 '%'는 수학적 모듈로 연산자를 나타낸다. 다시 말하면, A % B는 B로 나눈 A의 나머지가 되어서, 이 나머지는 0 및 B-1 사이의 정수의 범위 내에 있게 된다.

[0135] 표 4는 동작점 기술자들 (118)(도 4) 중의 하나에 포함될 수 있는 데이터의 예시적인 집합을 요약하고 있다.

**표 4**

**표 4 - MVC 동작점 기술자**

[0136]

구문	비트 수	연상기호
MVC_op_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
frame_rate	16	uimsbf
num_display_views	10	uimsbf
num_decode_views	10	uimsbf
for (i = 0 ; i< num_display_views; i++){		
view_id	10	uimsbf

}		
average_bit_rate	16	uimsbf
maximum_bitrate	16	uimsbf
temporal_id	3	uimsbf
for (i = 0; i < (1 + 6 * num_display_views) % 8; i++) {		
reserved_bit	1	bslbf
}		
}		

[0137] 도 5는 동작점 기술자들 (114)(도 3) 중의 하나에 포함될 수 있는 데이터의 대체 예의 집합을 도시하는 블록도이다. 대체로, 동작점 기술자들 (114) 각각은 공통의 형식을 가져야 하며, 그래서 클라이언트 디바이스는 단일 형식의 동작점 기술자들을 수신하도록 구성될 수 있다. 그래서, 동작점 기술자들 (114) 각각은 도 4, 도 5, 또는 도 6의 동작점 기술자와 유사한 형식, 또는 유사한 시그널링 데이터를 구비하는 다른 공통의 형식을 가질 수 있다.

[0138] 도 5의 예에서, 동작점 기술자 (140) 는 기술자 태그 필드 (142), 기술자 길이 필드 (144), profile\_IDC 필드 (146), level\_IDC 필드 (148), 프레임 레이트 필드 (149), 디스플레이 뷰들의 수 필드 (150), 디코딩 뷰들의 수 필드 (152), 평균 비트레이트 필드 (154), 최대 비트레이트 필드 (156), 시간적 식별자 필드 (158), 예약된 비트 필드 (160), 뷰 순서 색인 필드들 (162), 뷰 식별자 필드들 (164), 및 예약된 트레일링 비트 필드들 (166) 을 구비한다. IDC는 "지시자(indicator)"를 의미한다. 아래에서 설명되는 바와 같이, 예의 동작점 기술자 (140) 는 동작점에 대한 profile\_idc 및 level\_idc 값들, 뿐만 아니라 동작점이 어셈블링되는 방법에 관한 정보를 명시적으로 시그널링한다.

[0139] 디스플레이 뷰들의 수 필드 (150) 및 프레임 레이트 필드 (149) 는 동작점 기술자 (140) 에 의해 시그널링되는 랜더링 능력 값에 상응한다. 프로파일\_IDC 필드 (146), level\_IDC 필드 (148), 및 디코딩 뷰들의 수 필드 (152) 는, 도 5의 예에서, 동작점 기술자 (140) 에 의해 시그널링되는 디코딩 능력 값에 상응할 수 있는 데이터의 예들을 나타낸다. 평균 비트레이트 필드 (154) 및 최대 비트레이트 필드 (156) 는 동작점 기술자 (140) 에 의해 시그널링되는 비트레이트 값에 상응한다.

[0140] 위에서 설명된 바와 같이, MPEG-2 시스템들의 사양서는 각 필드가 8비트 길이인 기술자 태그 필드 및 기술자 길이 필드를 각각의 기술자가 가짐을 명시한다. 그래서, 멀티플렉서 (30)(도 1) 는 MVC 동작점 기술자를 나타내는 값을 기술자 태그 필드 (142) 에 지정할 수 있다. 멀티플렉서 (30) 는 동작점에 대한 뷰들의 수 및 동작점 기술자에 대한 예약된 비트들의 수를 결정한 다음, 기술자 길이 필드 (144) 를 뒤따르는 동작점 기술자 (140) 의 바이트 길이를 계산한다. 멀티플렉서 (30) 는 동작점 기술자 (140) 를 인스턴스화할 때, 이 계산된 길이 값을 기술자 길이 필드 (144) 에 지정할 수 있다.

[0141] Profile\_IDC 필드 (146) 는 동작점 기술자 (140) 에서 주어진 정보에 의해 리어셈블링된 동작점의 profile\_idc 를 나타내는 8비트 필드를 포함할 수 있다. Level\_IDC 필드 (148) 는 동작점 기술자 (140) 에서 주어진 정보에 의해 리어셈블링된 동작점의 level\_idc를 나타내는 8비트 필드를 포함할 수 있다.

[0142] 프레임 레이트 필드 (149) 는 리어셈블링된 AVC 비디오 스트림의 최대 프레임 레이트를 256 초당 프레임으로 나타내는 16비트 필드를 포함할 수 있다. 다시 말하면, 멀티플렉서 (30) 는 256초 기간의 최대 프레임 레이트를 계산하여 프레임 레이트 필드 (149) 의 값을 설정할 수 있다. 프레임 레이트 필드 (124) 와 마찬가지로, 프레임 레이트 필드 (149) 에 대한 다른 예들에서, 256초 외의 기간들이 사용될 수 있다.

[0143] 디스플레이 뷰들의 수 필드 (150) 는 리어셈블링된 AVC 비디오 스트림의 출력 대상이 되는 뷰들의 수의 값을 나타내는 10비트 필드를 포함할 수 있다. 대체로, 디스플레이 뷰들의 수 필드 (150) 는 디스플레이될 뷰들의 수를 상응하는 동작점을 위해 나타낸다. 디코딩 뷰들의 수 필드 (152) 는 리어셈블링된 AVC 비디오 스트림을 디코딩하기 위해 요구되는 뷰들의 수의 값을 나타내는 10비트 필드를 포함할 수 있다. 이 값은 디스플레이 뷰들의 수 필드 (150) 에 의해 나타내어지는 디스플레이될 뷰들의 수와는 상이할 수 있다. 이는 예컨대, 디코딩 뷰들의 수 필드 (128) 에 관하여 위에서 설명된 바와 같이, 뷰 의존도들 때문에 디코딩을 위해 요구되지만 실제로는 디스플레이되지 않는 일정한 뷰들로 인한 것일 수 있다.

[0144] 평균 비트레이트 필드 (154) 는 리어셈블링된 AVC 비디오 스트림의 평균 비트 레이트를 초당 킬로비트로 나타내는 16비트 필드를 포함할 수 있다. 0으로 설정될 때, 평균 비트 레이트는 명시되지 않는다. 다시 말하



면, 평균 비트레이트 필드 (154) 에 대한 영(0)의 값은 평균 비트레이트 필드 (154) 가 리어셈블링된 AVC 비디오 스트림의 평균 비트레이트를 결정하는데 사용되지 않아야 한다는 것을 의미한다. 최대 비트레이트 필드 (156) 는 리어셈블링된 AVC 비디오 스트림의 최대 비트 레이트를 초당 킬로비트로 나타내는 16비트 필드를 포함할 수 있다. 0으로 설정될 때, 최대 비트 레이트는 명시되지 않는다. 다시 말하면, 최대 비트레이트 필드 (156) 에 대한 값이 영으로 설정될 때, 최대 비트레이트 필드 (156) 는 리어셈블링된 AVC 비디오 스트림의 최대 비트레이트를 결정하는데 사용되지 않아야 한다.

[0145] 시간적 식별자 필드 (158) 는 리어셈블링된 AVC 비디오 스트림의 프레임 레이트에 상응하는, temporal\_id의 값을 나타내는 3비트 필드를 포함할 수 있다. 다시 말하면, temporal\_id는 위에서 논의된 바와 같이 리어셈블링된 AVC 비디오 스트림의 프레임 레이트를 결정하는데 사용될 수 있다.

[0146] 동작점 기술자 (140) 는 또한 뷰 순서 색인 필드들 (162) 및 뷰 식별자 필드들 (164) 을 구비한다. 뷰 순서 색인 필드들 (162) 각각은 동작점에 포함되는 NAL 단위들의 뷰 순서 색인의 값을 나타내는 10비트 필드를 포함할 수 있다. 클라이언트 디바이스는 동작점 기술자 (140) 에서 view\_order\_index 필드들 (162) 에 의해 시그널링되는 모든 시그널링된 view\_order\_index 값들에 상응하는 NAL 단위들을 어셈블링할 수 있다. 뷰 순서 색인 필드들 (162) 은 디코딩될 뷰들 각각에 대한 뷰 순서 색인 필드들을 구비한다. 뷰 순서 색인 값이 주어지면, 클라이언트 디바이스는 상응하는 NAL 단위들을 기본 스트림들로부터 추출할 수 있는데, MVC 확장 기술자가 뷰 순서 색인 값들의 범위를 그 기본 스트림으로 알려 주고 그 범위는 동작점 기술자로 시그널링되는 view\_order\_index 값을 커버하기 때문이다.

[0147] 뷰 식별자 필드들 (164) 각각은 리어셈블링된 AVC 비디오 비트스트림에 들어 있는 NAL 단위들의 view\_id의 값을 나타내는 10비트 필드를 포함할 수 있다. 그래서, 동작점에 대한 각각의 디스플레이되는 뷰들의 뷰 식별자들은 뷰 식별자 필드들 (164) 을 이용하여 시그널링된다. 다시 말하면, 뷰 식별자 필드들 (164) 의 뷰 식별자들은 디스플레이되는 뷰들에 상응한다. 그래서, 디코딩되었지만 디스플레이되지 않는 뷰들은 도 5의 예에서 뷰 식별자 필드들 (164) 에 의해 시그널링되지 않는다.

[0148] 동작점 기술자 (140) 는 또한 예약된 트레일링 비트 필드들 (166) 을 구비한다. 동작점 기술자 (140) 는 동작점 기술자 (140) 의 비트들의 수가 8로써 균등하게 나누어지도록 트레일링 비트들을 패딩 (padding) 으로서 구비할 수 있다. 뷰 순서 색인 필드들 및 뷰 식별자 필드들의 수가 가변할 수 있기 때문에, 멀티플렉서 (30) 가 동작점 기술자 (140) 내에 포함시키는 트레일링 비트들의 수는 그에 따라 가변할 수 있다. 예를 들어, 트레일링 비트들의 수는 다음 수학식에 따라 결정될 수 있다

[0149] 
$$\text{트레일링 비트 수} = (6 * (\text{num\_display\_views} + \text{num\_decode\_views})) \% 8$$

[0150] 여기서 '%'는 모듈로 연산자를 나타낸다.

[0151] 표 5는 예의 동작점 기술자 (140)(도 5) 에 포함될 수 있는 데이터의 예시적인 집합을 요약하고 있다.

**표 5**

**표 5 - MVC 동작점 기술자**

[0152]

구분	비트 수	연상기호
MVC_op_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
profile_idc	8	uimsbf
level_idc	8	uimsbf
frame_rate	16	uimsbf
num_display_views	10	uimsbf
num_decode_views	10	uimsbf
average_bit_rate	16	
maximum_bitrate	16	uimsbf
temporal_id	3	uimsbf
reserved_bit	1	bslbf
for (i = 0 ; i< num_decode_views; i++){		
view_order_index	10	uimsbf
}		

for (i = 0 ; i< num_display_views; i++){		
view_id	10	uimsbf
}		
for (i = 0 ; i<6*(num_display_views + num_decode_views) %8; i++){		
reserved_bit	1	bslbf
}		
}		

- [0153] 도 6은 동작점 기술자들 (114)(도 3) 중의 하나에 포함될 수 있는 데이터의 다른 대체 예의 집합을 도시하는 블록도이다. 도 6의 예에서, 동작점 기술자 (170) 는 기술자 태그 필드 (172), 기술자 길이 필드 (174), profile\_IDC 필드 (176), level\_IDC 필드 (178), 프레임 레이트 필드 (180), 디스플레이 뷰들의 수 필드 (182), 디코딩 뷰들의 수 필드 (184), 평균 비트레이트 필드 (186), 최대 비트레이트 필드 (188), 시간적 식별자 필드 (190), 예약된 비트 필드 (192), 동작점 식별자 필드 (194), 동작점 종속 플래그 필드 (196), 옵션의 종속 동작점 식별자 필드 (198), 뷰 순서 색인 필드들 (200), 뷰 식별자 필드들 (202), 및 예약된 트레일링 비트 필드들 (204) 을 구비한다. 아래에서 설명되는 바와 같이, 동작점 기술자 (170) 는, 다른 동작점에 의존하고 디코딩을 위해 요구되는 과외의 뷰들을 시그널링하는 동작점에 대한 예시적인 동작점 기술자를 제공한다.
- [0154] 디스플레이 뷰들의 수 필드 (182) 및 프레임 레이트 필드 (180) 는 동작점 기술자 (140) 에 의해 시그널링되는 렌더링 능력 값에 상응한다. 프로파일\_IDC 필드 (176), level\_IDC 필드 (178), 및 디코딩 뷰들의 수 필드 (184) 는, 도 6의 예에서, 동작점 기술자 (140) 에 의해 시그널링되는 디코딩 능력 값에 상응할 수 있는 데이터의 예들을 나타낸다. 평균 비트레이트 필드 (154) 및 최대 비트레이트 필드 (156) 는 동작점 기술자 (140) 에 의해 시그널링되는 비트레이트 값에 상응한다.
- [0155] 위에서 설명된 바와 같이, MPEG-2 시스템들의 사양서는 각 필드가 8비트인 기술자 태그 필드 및 기술자 길이 필드를 각각의 기술자가 가짐을 명시한다. 그래서, 멀티플렉서 (30)(도 1) 는 MVC 동작점 기술자를 나타내는 값을 기술자 태그 필드 (172) 에 지정할 수 있다. 멀티플렉서 (30) 는 동작점에 대한 뷰들의 수 및 동작점 기술자에 대한 예약된 비트들의 수를 결정한 다음, 기술자 길이 필드 (174) 를 뒤따르는 동작점 기술자 (170) 의 바이트 길이를 계산한다. 멀티플렉서 (30) 는 동작점 기술자 (140) 를 인스턴스화할 때, 이 계산된 길이 값을 기술자 길이 필드 (174) 에 지정한다.
- [0156] Profile\_IDC 필드 (176) 는 동작점 기술자 (170) 에서 주어진 정보에 의해 리어셈블링된 동작점의 profile\_idc 를 나타내는 8비트 필드를 포함할 수 있다. Level\_IDC 필드 (178) 는 동작점 기술자 (170) 에서 주어진 정보에 의해 리어셈블링된 동작점의 level\_idc를 나타내는 8비트 필드를 포함할 수 있다.
- [0157] 프레임 레이트 필드 (180) 는 리어셈블링된 AVC 비디오 스트림의 최대 프레임 레이트를 256 초당 프레임으로 나타내는 16비트 필드를 포함할 수 있다. 다시 말하면, 멀티플렉서 (30) 는 256초 기간의 최대 프레임 레이트를 계산하여 프레임 레이트 필드 (149) 의 값을 설정할 수 있다. 프레임 레이트 필드 (124) 와 마찬가지로, 프레임 레이트 필드 (180) 에 대한 다른 예들에서, 256초 외의 기간들이 사용될 수 있다.
- [0158] 디스플레이 뷰들의 수 필드 (182) 는 리어셈블링된 AVC 비디오 스트림의 출력 대상이 되는 뷰들의 수의 값을 나타내는 10비트 필드를 포함할 수 있다. 대체로, 디스플레이 뷰들의 수 필드 (182) 는 디스플레이될 뷰들의 수를 상응하는 동작점을 위해 나타낸다. 디코딩 뷰들의 수 필드 (184) 는 리어셈블링된 AVC 비디오 스트림을 디코딩하기 위해 요구되는 뷰들의 수의 값을 나타내는 10비트 필드를 포함할 수 있다. 이 값은 디스플레이 뷰들의 수 필드 (182) 에 의해 나타내어지는 디스플레이될 뷰들의 수와는 상이할 수 있다. 이는 예컨대, 디코딩 뷰들의 수 필드 (128) 에 관하여 위에서 설명된 바와 같이, 뷰 의존도들 때문에 디코딩을 위해 요구되지만 실제로는 디스플레이되지 않는 일정한 뷰들로 인한 것일 수 있다.
- [0159] 평균 비트레이트 필드 (186) 는 리어셈블링된 AVC 비디오 스트림의 평균 비트 레이트를 초당 킬로비트로 나타내는 16비트 필드를 포함할 수 있다. 0으로 설정될 때, 평균 비트 레이트는 명시되지 않는다. 다시 말하면, 평균 비트레이트 필드 (186) 에 대한 영(0)의 값은 평균 비트레이트 필드 (186) 가 리어셈블링된 AVC 비디오 스트림의 평균 비트레이트를 결정하는데 사용되지 않아야 한다는 것을 의미한다. 최대 비트레이트 필드 (188) 는 리어셈블링된 AVC 비디오 스트림의 최대 비트 레이트를 초당 킬로비트로 나타내는 16비트 필드를 포함할 수 있다. 0으로 설정될 때, 최대 비트 레이트는 명시되지 않는다. 특히, 최대 비트레이트 필드 (188) 에 대한 값이 영으로 설정될 때, 최대 비트레이트 필드 (188) 는 리어셈블링된 AVC 비디오 스트림의 최대

비트레이트를 결정하는데 사용되지 않아야 한다.

[0160] 시간적 식별자 필드 (190) 는 리어셈블링된 AVC 비디오 스트림의 프레임 레이트에 상응하는, temporal\_id의 값을 나타내는 3비트 필드를 포함할 수 있다. 다시 말하면, temporal\_id는 위에서 논의된 바와 같이 리어셈블링된 AVC 비디오 스트림의 프레임 레이트를 결정하는데 사용될 수 있다. 예약된 비트 필드 (192) 는 장래의 사용을 위해 예약되는 단일 비트에 해당한다.

[0161] 동작점 기술자 (170) 는 또한 동작점 식별자 필드 (194) 및 동작점 종속 플래그 필드 (196) 를 구비한다. 동작점 식별자 필드 (194) 는 동작점 기술자 (170) 에 의해 기술되는 동작점의 식별자를 나타내는 10비트 필드를 포함할 수 있다. 동작점 종속 플래그 필드 (196) 는 현재 동작점의 다른 동작점에 대한 의존도가 시그널링되는지를 나타내는 단일 비트 플래그이다. 동작점 종속 플래그 (196) 가 1 (또는 참) 의 값을 가진다면, 의존도는 시그널링되며; 동작점 종속 플래그 (196) 의 값이 영 (또는 거짓) 이면, 의존도는 시그널링되지 않는다.

[0162] 동작점 종속 플래그 (196) 의 값이 참 또는 1일 때, 동작점 기술자 (170) 는 종속 동작점 식별자 필드 (198) 를 추가로 구비한다. 존재할 때, 동작점 식별자 필드 (198) 는 현재의 기술자가 종속하는 동작점의 식별자를 나타내는 10비트 필드를 포함할 수 있다. 다시 말하면, 동작점 기술자 (170) 가 다른 동작점에 종속하는 동작점에 상응한다고 동작점 멀티플렉서 (30) 가 결정하는 경우, 멀티플렉서 (30) 는 동작점 종속 플래그의 값을 참 또는 1로 설정하고, 그런 다음 동작점 기술자 (170) 에 상응하는 동작점이 종속하는 동작점의 식별자를 시그널링한다.

[0163] 동작점 기술자 (170) 는 또한 뷰 순서 색인 필드들 (200) 및 뷰 식별자 필드들 (202) 을 구비한다. 뷰 순서 색인 필드들 (202) 각각은 operation\_point\_id의 식별자를 갖는 현재 동작점에는 들어 있지만 dependent\_operation\_point\_id의 식별자를 갖는 동작점에는 들어 있지 않은 NAL 단위들의 뷰 순서 색인의 값을 나타내는 10비트 필드를 포함할 수 있다. 클라이언트 디바이스는 동작점 기술자 (170) 에서 뷰 순서 색인 필드들 (200) 에 의해 시그널링되는 모든 시그널링된 view\_order\_index 값들에 상응하는 NAL 단위들을 어셈블링할 수 있다. 뷰 순서 색인 필드들 (200) 은 디코딩될 뷰들 각각에 대한 뷰 순서 색인 필드들을 구비한다. view\_order\_index 값이 주어지면, 클라이언트 디바이스는 상응하는 NAL 단위들을 기본 스트림들로부터 추출할 수 있는데, MVC 확장 기술자가 뷰 순서 색인 값들의 범위를 그 기본 스트림으로 알려 주고 그 범위는 동작점 기술자로 시그널링되는 view\_order\_index 값을 커버하기 때문이다. 동작점 기술자 (170) 로 시그널링되는 동작점은 뷰 순서 색인 필드들 (200) 의 모든 시그널링되는 view\_order\_index 값들에 상응하는 NAL 단위들 및 dependent\_operation\_point\_id의 식별자를 갖는 동작점에 의해 포함되는 NAL 단위들에 의해 리어셈블링된다.

[0164] 뷰 식별자 필드들 (202) 각각은 리어셈블링된 AVC 비디오 비트스트림에 들어 있는 NAL 단위들의 view\_id의 값을 나타내는 10비트 필드를 포함할 수 있다. 그래서, 동작점에 대한 각각의 디스플레이되는 뷰들의 뷰 식별자들은 뷰 식별자 필드들 (202) 을 이용하여 시그널링된다. 다시 말하면, 뷰 식별자 필드들 (164) 의 뷰 식별자들은 디스플레이되는 뷰들에 상응한다. 그래서, 디코딩되었지만 디스플레이되지 않는 뷰들은 도 5의 예에서 뷰 식별자 필드들 (202) 에 의해 시그널링되지 않는다.

[0165] 동작점 기술자 (170) 는 또한 예약된 트레일링 비트 필드들 (204) 을 구비한다. 동작점 기술자 (170) 는 동작점 기술자 (170) 의 비트들의 수가 8로써 균등하게 나누어지도록 트레일링 비트들을 패딩 (padding) 으로서 구비할 수 있다. 뷰 순서 색인 필드들 및 뷰 식별자 필드들의 수가 가변할 수 있기 때문에, 멀티플렉서 (30) 가 동작점 기술자 (170) 내에 포함시키는 트레일링 비트들의 수는 그에 따라 가변할 수 있다. 예를 들어, 트레일링 비트들의 수는 다음 수학식에 따라 결정될 수 있다

[0166] 
$$\text{트레일링 비트 수} = (6 * (\text{num\_display\_views} + \text{num\_decode\_views})) \% 8$$

[0167] 여기서 '%'는 모듈로 연산자를 나타낸다.

[0168] 표 6은 예의 동작점 기술자 (170)(도 6) 에 포함될 수 있는 데이터의 예시적인 집합을 요약하고 있다.

**표 6**

**표 6 - MVC 동작점 기술자**

[0169]

구문	비트 수	연상기호
MVC_op_descriptor(){		

descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
profile_idc	8	uimsbf
level_idc	8	uimsbf
frame_rate	16	uimsbf
num_display_views	10	uimsbf
num_decode_views	10	uimsbf
average_bit_rate	16	
maximum_bitrate	16	uimsbf
temporal_id	3	uimsbf
reserved_bit	1	bslbf
operation_point_id	10	uimsbf
op_dependent_flag	1	bslbf
if (op_dependent_flag)		
dependent_operation_point_id	10	10
for (i = 0 ; i < num_decode_views; i++){		
view_order_index	10	uimsbf
}		
for (i = 0 ; i < num_display_views; i++){		
view_id	10	uimsbf
}		
for (i = 0 ; i < 6*(num_display_views +		
num_decode_views) % 8; i++){		
reserved_bit	1	bslbf
}		
}		

[0170] 또 다른 대체예로서, 소스 디바이스 (20)(도 1) 는 동작점의 특성들을 동작점 기술자와는 다른 데이터 구조를 이용하여 시그널링한다. 예를 들어, 소스 디바이스 (20) 는 MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 랜더링 능력을 기술하는 랜더링 능력 값, MVC 동작점을 사용하기 위해 수신 디바이스에 의해 충족되어야 할 디코딩 능력을 기술하는 디코딩 능력 값, 및 MVC 동작점의 비트레이트를 기술하는 비트레이트 값을 수정된 MVC 확장 기술자를 이용하여 시그널링할 수 있다. 이하의 표 7이 이러한 수정된 MVC 확장 기술자의 일 예를 도시한다.

**표 7**

**표 7 - MVC 확장 기술자**

[0171]

구문	비트 수	연상기호
MVC_extension_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
frame_rate	16	uimsbf
num_display_views	16	uimsbf
num_decode_views	10	uimsbf
for (i = 0 ; i < num_display_views; i++){		
view_id	10	uimsbf
}		
average_bit_rate	16	uimsbf
maximum_bitrate	16	uimsbf
예약됨	4	bslbf
view_order_index_min	10	bslbf
view_order_index_max	10	bslbf
temporal_id_start	3	bslbf
temporal_id_end	3	bslbf
no_sei_nal_unit_present	1	bslbf

예약됨	1	bslbf
}		

- [0172] 멀티플렉서 (30)(도 2) 는 표 7에 의해 정의된 구문에 따라 MVC 확장 기술자들 (110) 을 구축할 수 있다. 대체로, 표 7의 구문론적 요소들의 의미들(semantics)은 위의 표 1에 관하여 설명된 공통의 이름의 요소들과 동일하다. 표 7의 예는 표 1의 그것들에 대한 부가적인 요소들, 즉 프레임 레이트 필드, 디스플레이 뷰들의 수 필드, 디코딩 뷰들의 수 필드, 및 MVC 확장 기술자에 상응하는 동작점 각각의 뷰에 대한 뷰 식별자 필드들을 구비한다.
- [0173] 프레임 레이트 필드는 리어셈블링된 AVC 비디오 스트림의 최대 프레임 레이트를 256 초당 프레임으로 나타내는 16비트 필드를 포함할 수 있다. 디스플레이 뷰들의 수 필드인 "num\_display\_views"는 리어셈블링된 AVC 비디오 스트림의 출력 대상이 되는 뷰들의 수의 값을 나타내는 10비트 필드를 포함할 수 있다. 디코딩 뷰들의 수 필드인 "num\_decode\_views"는 리어셈블링된 AVC 비디오 스트림을 디코딩하기 위해 요구되는 뷰들의 수의 값을 나타내는 10비트 필드를 포함할 수 있다. 뷰 식별자 필드들 "view\_id" 각각은 리어셈블링된 AVC 비디오 비트스트림에 들어 있는 상응하는 뷰에 대한 NAL 단위들의 view\_id의 값을 나타내는 10비트 필드를 포함할 수 있다.
- [0174] 일부 예들에서, 하나 이상의 동작점 기술자들은 비트스트림의 모든 MVC 동작점들의 최대 시간적 식별자 값 및 최대 프레임 레이트 값을 나타내는 값들을 구비할 수 있다. 일부 예들에서, 비트스트림의 모든 MVC 동작점들의 최대 시간적 식별자 값 및 최대 프레임 레이트 값은 MVC 동작점 기술자로 시그널링될 수 있다.
- [0175] 도 7은 예시적인 MVC 예측 패턴을 도시하는 개념도이다. 도 7의 예에서, 8개의 뷰들 (뷰 ID들인 "S0" 내지 "S7"를 가짐) 이 도시되고, 12개의 시간적 위치들 ("T0" 내지 "T11") 이 각각의 뷰를 위해 도시되어 있다. 다시 말하면, 도 7 각각의 행은 뷰에 해당하며, 각각의 열은 시간적 위치를 나타낸다.
- [0176] MVC가 H.264/AVC 디코더들에 의해 디코딩될 수 있는 이른바 기본 뷰(base view)를 가지고 스테레오 뷰 쌍은 MVC에 의해 지원될 수 있지만, MVC의 이점은 그것이, 2개를 넘는 뷰들을 3D 비디오 입력으로서 사용하고 다수의 뷰들에 의해 표현되는 이 3D 비디오를 디코딩하는 예를 지원할 수 있다는 것이다. MVC 디코더를 갖는 클라이언트의 렌더러(renderer)는 다수의 뷰들으로써 3D 비디오 콘텐츠를 예상할 수 있다.
- [0177] 도 7의 프레임들은 상응하는 프레임이 인트라 코딩된 (다시 말하면, I-프레임), 또는 한 방향으로 (다시 말하면, P-프레임으로서) 또는 다수의 방향으로 (다시 말하면, B-프레임으로서) 인트라 코딩된 것인지를 지시하는 문자를 포함하는 음영진 블록을 이용하여, 도 7의 각각의 행 및 각각의 열에 나타내어지고 있다. 대체로, 예측들은 화살표들로 나타내어지며, 가리켜진 프레임은 예측 참조에 대한 대상으로부터의 포인트를 이용한다. 예를 들어, 시간적 위치 T0에서의 뷰 S2의 P-프레임은 시간적 위치 T0에서의 뷰 S0의 I-프레임으로부터 예측된다.
- [0178] 단일 뷰 비디오 인코딩과 마찬가지로, 멀티뷰 비디오 코딩 비디오 시퀀스의 프레임들은 상이한 시간적 위치들에서 프레임들에 관하여 예측 인코딩될 수 있다. 예를 들어, 시간적 위치 T1에서의 뷰 S0의 b-프레임은 시간적 위치 T0에서 자신으로 향하는 뷰 S0의 I-프레임으로부터의 화살표를 가져서, b-프레임이 I-프레임으로부터 예측됨을 나타낸다. 그러나, 멀티뷰 비디오 인코딩의 측면에서는 부가적으로, 프레임들이 뷰 간 예측될 수 있다. 다시 말하면, 뷰 성분은 참조를 위해 다른 뷰들의 뷰 성분들을 이용할 수 있다. MVC에서는, 예를 들어, 뷰 간 예측은 다른 뷰의 뷰 성분이 인트라 예측 참조인 듯이 실현된다. 잠정적 뷰 간 참조들은 시퀀스 매개변수 집합 (SPS) MVC 확장물로 시그널링되고 참조 화상 목록 구축 프로세스에 의해 수정될 수 있는데, 이는 인트라 예측 또는 뷰 간 예측 참조들의 유연한 순서매김을 가능하게 한다. 아래의 표 8은 MVC 확장 시퀀스 매개변수 집합에 대한 예시적인 정의를 제공한다.

표 8

표 8

[0179]

seq_parameter_set_mvc_extension(){	C	기술자
num_views_minus1	0	ue(v)
for(i = 0; i <= num_views_minus1; i++)		
view_id[i]	0	ue(v)



for(i = 1; i <= num_views_minus1; i++){		
num_anchor_refs_l0[i]	0	ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++){		
anchor_ref_l0[i][j]	0	ue(v)
num_anchor_refs_l1[i]	0	ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++){		
anchor_ref_l1[i][j]	0	ue(v)
}		
for(i = 1; i <= num_views_minus1; i++){		
num_non_anchor_refs_l0[i]	0	ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++){		
non_anchor_ref_l0[i][j]	0	ue(v)
num_non_anchor_refs_l1[i]	0	ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++){		
non_anchor_ref_l1[i][j]	0	ue(v)
}		
num_level_values_signalled_minus1	0	ue(v)
for(i = 0; i <= num_level_values_signalled_minus1; i++){		
level_idc[i]	0	u(8)
num_applicable_ops_minus1[i]	0	ue(v)
for(j = 0; j <= num_applicable_ops_minus1[i]; j++){		
applicable_op_temporal_id[i][j]	0	u(3)
applicable_op_num_target_views_minus1[i][j]	0	ue(v)
for(k = 0; k <=		
applicable_op_num_target_views_minus1[i][j]; k++){		
applicable_op_target_view_id[i][j][k]	0	ue(v)
applicable_op_num_views_minus1[i][j]	0	ue(v)
}		
}		
}		

- [0180] 도 7은 뷰 간 예측의 다양한 예들을 제공한다. 뷰 S1의 프레임들은, 도 7의 예에서, 뷰 S1의 상이한 시간적 위치들에 있는 프레임들로부터 예측된 것으로 뿐만 아니라, 동일한 시간적 위치들에 있는 뷰들 (S0 및 S2) 의 프레임들로부터 뷰 간 예측된 것으로 도시되고 있다. 예를 들어, 시간적 위치 T1에 있는 뷰 S1의 b-프레임 은 시간적 위치들 (T0 및 T2) 에 있는 뷰 S1의 B-프레임들 각각 뿐만 아니라 시간적 위치 T1에 있는 뷰들 (S0 및 S2) 의 b-프레임들로부터 예측된다.
- [0181] 도 7의 예에서, 대문자 "B" 및 소문자 "b"는 상이한 인코딩 방법론들보다는 프레임들 사이의 상이한 계층구조적 관계들을 나타내는 것을 의도한다. 대체로, 대문자 "B"의 프레임들은 소문자 "b"의 프레임들보다 예측 계층 구조에서 상대적으로 상위에 있다. 도 7은 예측 계층구조에서의 변화들을 상이한 음영(shading) 레벨들을 이용하여 도시하는데, 음영 량이 더 큰 (다시 말하면, 상대적으로 더 어두운) 프레임들일수록 적은 음영을 갖는 (다시 말하면, 상대적으로 더 밝은) 프레임들보다 예측 계층구조에서 더 상위에 있다. 예를 들어, 도 7의 모든 I-프레임들은 완전한 음영으로 도시되는 반면, P-프레임들은 약간 더 밝은 음영을 가지고, B-프레임들 (및 소문자 b-프레임들) 은 서로 상대적이지만 P-프레임들 및 I-프레임들의 음영보다는 항상 더 밝은 여러 가지 음 영 레벨들을 가진다.
- [0182] 대체로, 예측 계층구조는 예측 계층구조에서 상대적으로 더 높은 프레임들이 그 계층구조에서 상대적으로 더 낮 은 프레임들을 디코딩하기 전에 디코딩되어야 한다는 점에서 뷰 순서 색인들에 관련되어서, 계층구조에서 상대 적으로 더 높은 프레임들은 계층구조에서 상대적으로 더 낮은 프레임들의 디코딩 동안에 기준 프레임들로서 사 용될 수 있다. 뷰 순서 색인은 액세스 단위에서 뷰 성분들의 디코딩 순서를 나타내는 색인들이다. 뷰 순서 색인들은 H.264/AVC (MVC 개정안) 의 부록 A에 명시된 바와 같이, SPS MVC 확장에서 암시된다. SPS에 서, 각각의 색인 i에 대해, 상응하는 view\_id가 시그널링된다. 뷰 성분들의 디코딩은 뷰 순서 색인의 오름 차순으로 일어날 것이다. 모든 뷰들이 표현된다면, 뷰 순서 색인들은 0부터 num\_views\_minus\_1로의 연속하 는 순서로 존재한다.
- [0183] 이런 방식으로, 참조 프레임들로서 사용되는 프레임들은 참조 프레임들을 참조하여 인코딩된 프레임들을 디코딩

하기 전에 디코딩될 수 있다. 뷰 순서 색인은 액세스 단위에서 뷰 성분들의 디코딩 순서를 나타내는 색인들이다. 각각의 뷰 순서 색인  $i$ 에 대해, 상응하는 `view_id`가 시그널링된다. 뷰 성분들의 디코딩은 뷰 순서 색인의 오름 차순으로 일어난다. 모든 뷰들이 표현된다면, 뷰 순서 색인들의 집합은 영부터 뷰들의 전체 수보다는 하나 작은 수까지의 연속 순서의 집합을 포함한다.

[0184] 계층구조의 동일한 레벨들에 있는 프레임들의 경우, 디코딩 순서는 서로에 대해 중요하지 않을 수 있다. 예를 들어, 시간적 위치  $T_0$ 에 있는 뷰  $S_0$ 의 I-프레임은 시간적 위치  $T_0$ 에 있는 뷰  $S_2$ 의 P-프레임에 대한 참조 프레임으로서 사용되며, 이 P-프레임은 시간적 위치  $T_0$ 에 있는 뷰  $S_4$ 의 P-프레임에 대한 참조 프레임으로서 사용된다. 따라서, 시간적 위치  $T_0$ 에 있는 뷰  $S_0$ 의 I-프레임은 시간적 위치  $T_0$ 에 있는 뷰  $S_2$ 의 P-프레임보다 먼저 디코딩되어야 하고, 이 P-프레임은 시간적 위치  $T_0$ 에 있는 뷰  $S_4$ 의 P-프레임보다 먼저 디코딩되어야 한다. 그러나, 뷰들 ( $S_1$  및  $S_3$ ) 사이에서 디코딩 순서는 중요하지 않은데, 뷰들 ( $S_1$  및  $S_3$ ) 이 예측을 위해 서로를 필요로 하지 않고 그 대신 예측 계층구조에서 더 상위의 뷰들로부터만 예측되기 때문이다. 더군다나, 뷰  $S_1$  이 뷰들 ( $S_0$  및  $S_2$ ) 뒤에 디코딩되는 한, 뷰  $S_1$ 은 뷰  $S_4$ 보다 앞서 디코딩될 수 있다.

[0185] 이런 방식으로, 계층구조적 순서매김이 뷰들 ( $S_0$  내지  $S_7$ ) 을 기술하는데 사용될 수 있다. 표기  $SA > SB$ 는 뷰  $SA$ 가 뷰  $SB$ 보다 앞서 디코딩될 수 있다는 것을 의미한다. 이 표기법을 이용하면, 도 7의 예에서는,  $S_0 > S_2 > S_4 > S_6 > S_7$ 이다. 또한, 도 7의 예에 대하여,  $S_0 > S_1$ ,  $S_2 > S_1$ ,  $S_2 > S_3$ ,  $S_4 > S_3$ ,  $S_4 > S_5$ , 그리고  $S_6 > S_5$ 이다. 이 조건들을 위반하지 않는 뷰들에 대해 어떤 디코딩 순서라도 가능하다. 따라서, 많은 상이한 디코딩 순서들이 일정한 제한들만으로 가능하다. 2개의 예의 디코딩 순서들이 아래에서 제시되지만, 많은 다른 디코딩 순서들이 가능함이 이해될 것이다. 아래의 표 9에 도시된 하나의 예에서, 뷰들은 가능한 한 빨리 디코딩된다.

표 9

표 9

뷰 ID	S0	S1	S2	S3	S4	S5	S6	S7
뷰 순서 색인	0	2	1	4	3	6	5	7

[0186]

[0187] 표 9의 예는 뷰들 ( $S_0$  및  $S_2$ ) 이 디코딩된 직후에 뷰  $S_1$ 이 디코딩될 수 있으며 뷰들 ( $S_2$  및  $S_4$ ) 이 디코딩된 직후에 뷰  $S_3$ 이 디코딩될 수 있고, 뷰들 ( $S_4$  및  $S_6$ ) 이 디코딩된 직후에 뷰  $S_5$ 가 디코딩될 수 있음을 인정한다.

[0188] 아래의 표 10은 다른 예의 디코딩 순서를 제공하는데 이 디코딩 순서는 다른 뷰에 대한 참조로서 사용되는 어떤 뷰가 임의의 다른 뷰에 대한 참조로서 사용되지 않는 뷰들에 앞서 디코딩되도록 한다.

표 10

표 10

뷰 ID	S0	S1	S2	S3	S4	S5	S6	S7
뷰 순서 색인	0	5	1	6	2	7	3	4

[0189]

[0190] 표 10의 예는 도 7의 예에서, 뷰들 ( $S_1$ ,  $S_3$ ,  $S_5$ , 및  $S_7$ ) 의 프레임들이 임의의 다른 뷰들의 프레임들에 대한 참조 프레임들로서 역할을 하지 않고, 그러므로, 뷰들 ( $S_1$ ,  $S_3$ ,  $S_5$ , 및  $S_7$ ) 은 참조 프레임들로서 사용되는 그런 뷰들, 다시 말하면, 뷰들 ( $S_0$ ,  $S_2$ ,  $S_4$ , 및  $S_6$ ) 의 프레임들보다 뒤에 디코딩될 수 있음을 인정한다. 서로에 대해, 뷰들 ( $S_1$ ,  $S_3$ ,  $S_5$ , 및  $S_7$ ) 은 어떤 순서로도 디코딩될 수 있다. 따라서, 표 10의 예에서, 뷰  $S_7$ 은 뷰들 ( $S_1$ ,  $S_3$ , 및  $S_5$ ) 각각보다 먼저 디코딩된다.

[0191] 명확히 하기 위해, 각각의 뷰의 프레임들뿐만 아니라 각각의 뷰의 프레임들의 시간적 위치들 사이의 계층구조적 관계가 존재한다. 도 7의 예에 관하여, 시간적 위치  $T_0$ 에 있는 프레임들은 시간적 위치  $T_0$ 에 있는 다른 뷰들의 프레임들로부터 인트라 예측되거나 또는 뷰 간 예측된다. 마찬가지로, 시간적 위치  $T_8$ 에 있는 프레임들은 시간적 위치  $T_8$ 에 있는 다른 뷰들의 프레임들로부터 인트라 예측되거나 또는 뷰 간 예측된다. 따라서, 시간적 계층구조에 관하여, 시간적 위치들 ( $T_0$  및  $T_8$ ) 은 시간적 계층구조의 최상위에 있다.

[0192] 시간적 위치  $T_4$ 에 있는 프레임들은 도 7의 예에서, 시간적 위치들 ( $T_0$  및  $T_8$ ) 의 프레임들보다 시간적 계층구조에서 하위에 있는데 시간적 위치  $T_4$ 의 프레임들이 시간적 위치들 ( $T_0$  및  $T_8$ ) 의 프레임들을 참조하여 B로 인코

당되기 때문이다. 시간적 위치들 (T2 및 T6) 에 있는 프레임들은 시간적 위치 T4에 있는 프레임들보다는 시간적 계층구조에서 하위에 있다. 마지막으로, 시간적 위치들 (T1, T3, T5, 및 T7) 에 있는 프레임들은 시간적 위치들 (T2 및 T6) 에 있는 프레임들보다는 시간적 계층구조에서 하위에 있다.

[0193] MVC에서, 전체 비트스트림의 서브셋이 MVC를 여전히 준수하는 서브-비트스트림을 형성하도록 추출된다. 지정 애플리케이션들이 예를 들어, 서버에 의해 제공되는 서비스, 하나 이상의 클라이언트들의 디코더들의 용량, 지원, 및 능력들, 그리고/또는 하나 이상의 클라이언트들의 선호도에 기초하여 요구할 수 있는 많은 가능한 서브-비트스트림들이 존재한다. 예를 들어, 클라이언트가 3개의 뷰들만을 요구할 지도 모르고, 2개의 시나리오들만이 있을 지도 모른다. 하나의 예에서, 하나의 클라이언트는 원활한 시청 경험을 요구할 수 있고 view\_id 값들 (S0, S1, 및 S2) 을 갖는 뷰들을 선호할지도 모르는 반면, 다른 클라이언트는 뷰 확장성을 요구할 수 있고 view\_id 값들 (S0, S2, 및 S4) 을 갖는 뷰들을 선호할 수 있다. 원래 view\_id들이 표 9의 예에 관한 순서로 된다면, 뷰 순서 색인 값들은 이들 두 예들에서 각각 {0, 1, 2} 및 {0, 1, 4}이다. 이들 서브-비트스트림들의 양쪽 모두는 독립 MVC 비트스트림들로서 디코딩될 수 있고 동시에 지원될 수 있다는 것에 주의한다.

[0194] MVC 디코더들에 의해 디코딩 가능한 많은 MVC 서브-비트스트림들이 존재한다. 이론상으로는, 다음 2가지 속성들을 충족시키는 뷰들의 어떠한 조합이라도 일정한 프로파일 또는 레벨을 준수하는 MVC 디코더에 의해 디코딩될 수 있다: (1) 각 액세스 단위의 뷰 성분들이 뷰 순서 색인의 증가 순서로 정돈된다, (2) 조합의 각각의 뷰에 대해, 그것의 종속 뷰들 역시 그 조합에 포함된다.

[0195] 도 8은 동작점의 특성들을 시그널링하는 데이터 구조를 이용하기 위한 예시적인 방법을 도시하는 흐름도이다. 다시 말하면, 도 8의 방법은 소스 디바이스, 예컨대, 소스 디바이스 (20)(도 1) 에 의해 MPEG-2 시스템들의 비트스트림의 각각의 동작점에 대한 데이터 구조의 구축을 포함한다. 도 8의 방법은 또한 수신된 데이터 구조들을 이용하여, 목적지 디바이스, 이를테면 목적지 디바이스 (40)(도 1) 에 의해 디코딩되고 디스플레이할 멀티미디어 데이터를 검색하는 동작점을 선택하는 것을 포함한다.

[0196] 처음에, 도 8의 예에서, 소스 디바이스 (20) 는 프로그램에 대한 동작점들을 결정한다(210). 예를 들어, 소스 디바이스 (20) 는 프로그램의 다양한 서브셋들의 뷰들을 선택하여 클라이언트 디바이스들이 다양한 능력들, 예컨대, 렌더링 및 디코딩 능력들을 가짐을 나타내는 다양한 동작점들을 생성할 수 있다. 관리자는 소스 디바이스 (20) 와 상호작용하여, 예를 들어, 뷰들을 선택하고 동작점들을 생성할 수 있는데, 이 동작점들은 클라이언트 디바이스들이 가변하는 렌더링 및 디코딩 능력들을 가짐을 나타내거나, 또는 상이한 동작점들이 소스 디바이스 (20) 에 의해 자동 생성될 수 있음을 나타낸다.

[0197] 프로그램에 대한 동작점들을 결정한 후, 소스 디바이스 (20) 는 예컨대, 비트스트림이 MPEG-2 시스템들의 전송 스트림으로서 브로드캐스트될 때, 동작점들에 대한 데이터 구조들을 프로그램 맵 테이블 내에 생성할 수 있다 (212). 다르게는, 소스 디바이스 (20) 는 비트스트림이 MPEG-2 시스템들의 프로그램 스트림으로서 브로드캐스트될 때, 프로그램 스트림 맵 내에 데이터 구조들을 생성할 수 있다. 어쨌든, 소스 디바이스 (20) 는 각각의 동작점에 대해, 상응하는 동작점의 특성들을 나타내는 데이터 구조를 생성할 수 있다. 이 데이터 구조는 예를 들어, 도 4-6의 예들 중의 하나에 상응하는 동작점 기술자를 포함할 수 있다. 이런 방식으로, 데이터 구조는 렌더링 특성들, 디코딩 특성들, 및 비트레이트를 상응하는 동작점을 위해 시그널링할 수 있다.

[0198] 그 다음 소스 디바이스 (20) 는 예컨대, 도 8의 예에서의 PMT 내의 데이터 구조들을 클라이언트 디바이스, 예컨대, 목적지 디바이스 (40)(도 1) 에 출력할 수 있다(214). 이런 방식으로, 소스 디바이스 (20) 는 이 데이터 구조들을 비트스트림의 부분으로서 출력할 수 있다. 소스 디바이스 (20) 는 예컨대, 차세대 텔레비전 방식 위원회 (ATSC) 표준들 또는 전국 텔레비전 방식 위원회 (NTSC) 표준들을 준수하는 신호들에 따라, 네트워크, 예컨대 무선 또는 유선 네트워크를 통한 브로드캐스트, 유니캐스트, 멀티캐스트, 애니캐스트, 또는 기타 통신 프로토콜, 또는 텔레비전 주파수들을 통한 브로드캐스트의 형태로 비트스트림을 출력한다. 다르게는, 소스 디바이스 (20) 는 비트스트림을 컴퓨터 판독가능 저장 매체, 이를테면 DVD-ROM, 블루-레이 디스크, 플래시 드라이브, 자기 디스크, 또는 기타 저장 매체에 인코딩할 수 있으며, 이 경우 소스 디바이스 (20) 는 동작점들에 대한 데이터 구조들을 구비하는 PSM을 형성하고 이 PSM을 컴퓨터 판독가능 저장 매체에 인코딩할 수 있다.

[0199] 목적지 디바이스 (40) 는 궁극적으로 PMT (또는 PSM) 를 소스 디바이스 (20) 로부터 수신할 수 있다 (216). 그러면 목적지 디바이스 (40) 는 동작점들 중의 하나를 PMT 또는 PSM에 포함된 데이터 구조들에 의해 시그널링되는 동작점들의 특성들에 기초하여 선택할 수 있다(218). 대체로, 목적지 디바이스 (40) 가 상응하는 데이터 구조에 의해 시그널링되는 렌더링 및 디코딩 능력들을 충족하는 동작점을 목적지 디바이스 (40) 는 선택할



수 있다. 예를 들어, 목적지 디바이스 (40) 는 동작점을 위해 데이터 구조에 의해 시그널링되는 랜더링 능력 값에 일치하는 프레임 레이트에서, 데이터 구조에 의해 지시되는 수의 뷰들을 비디오 출력단 (44) 이 랜더링할 수 있는지를 결정한다. 비슷하게, 목적지 디바이스 (40) 는 동작점을 위해 디코딩될 수의 뷰들을 이 동작점의 디코딩 능력 값 데이터 구조에 의해 시그널링되는 대로 비디오 디코더 (48) 가 디코딩할 수 있는지를 결정할 수 있다. 더군다나, 일부 예들에서, 목적지 디바이스 (40) 는 데이터 구조로 시그널링되는 비트레이트를 이용하여 예컨대, 목적지 디바이스 (40) 가 비트스트림을 수신하는 전송 매체의 대역폭 제한에 기초하여, 전송 매체에 적합한 동작점을 선택할 수 있다.

[0200] 하나를 넘는 동작점을 목적지 디바이스 (40) 가 랜더링하고 디코딩할 수 있는지를 목적지 디바이스 (40) 가 결정할 때, 목적지 디바이스 (40) 는 디코딩 및 랜더링에 대한 최고 품질의 동작점을 선택할 수 있다. 예를 들어, 목적지 디바이스 (40) 는 선택할 동작점이 무엇인지를 결정하기 위해, 뷰들의 최고 수, 최고 비트레이트, 최고 프레임 레이트, 또는 동작점에 대한 기타의 품질의 지시들을 갖는 동작점을 선택할 수 있다.

[0201] 동작점을 선택한 후, 목적지 디바이스 (40) 는 동작점에 대한 데이터를 비트스트림으로부터 검색한다(220). 다시 말하면, 목적지 디바이스 (40) 는 비트스트림에 포함된 프로그램으로부터 동작점에 상응하는 뷰들 각각에 대한 데이터를 추출할 수 있다. 일부 예들에서, 목적지 디바이스 (40) 는 비트스트림의 하나 이상의 서브-비트스트림들로부터 데이터를 선택하여 동작점에 대한 데이터를 추출한다. 데이터를 추출한 후, 목적지 디바이스는 선택된 동작점에 대한 데이터를 디코딩하고 디스플레이할 수 있다(222). 비디오 디코더 (48) 는 동작점을 위해 디코딩될 뷰들 각각을 디코딩할 수 있는 반면, 비디오 출력단 (44) 은 동작점을 위해 디스플레이될 뷰들 각각을 디스플레이할 수 있다. 디스플레이되는 뷰들은 위에서 설명된 바와 같이, 디코딩되는 뷰들 일 필요가 없을 수도 있다.

[0202] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 어떤 조합으로도 구현될 수 있다. 소프트웨어로 구현된다면, 기능들은 하나 이상의 명령어들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 전송될 수 있다. 컴퓨터 판독가능 매체는 한 장소에서 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 어떠한 매체라도 포함하는 데이터 저장 매체 또는 통신 매체와 같은 컴퓨터 판독가능 저장 매체를 포함한다. 데이터 저장 매체는 이 개시물에서 설명된 기법들의 구현을 위한 명령어들, 코드들 및/또는 구조들을 검색하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 어떤 이용가능 매체라도 될 수 있다. 예로서, 및 비제한적으로, 이러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 기타의 광 디스크 스토리지, 자기 디스크 스토리지, 또는 기타 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령어들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 어떤 관련된 것이라도 사실상 컴퓨터 판독가능 매체라고 한다. 예를 들어, 명령어들이 웹사이트, 서버, 또는 다른 원격 자원으로부터 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 회선 (DSL), 또는 무선 기술들 이블테먼 적외선, 라디오, 및/또는 마이크로파를 이용하여 전송된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술은 매체의 정의에 포함된다. 그러나, 컴퓨터-판독가능 저장 매체 및 데이터 저장 매체는 접속선들, 반송파들, 신호들, 또는 기타 일시적인 매체들을 포함하지 않음이 이해되어야 한다. 디스크 (Disk 및 disc) 는 여기서 사용되는 바와 같이, 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하는데, disk들은 보통 데이터를 자기적으로 재생하지만, disc들은 레이저들로서 광적으로 데이터를 재생한다. 상기한 것들의 조합들도 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

[0203] 명령어들은 하나 이상의 프로세서들, 이를테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그램가능 로직 어레이들 (FPGA들), 또는 기타 동등한 집적 또는 개별 로직 회로에 의해 실행될 수 있다. 따라서, 여기서 사용된 용어 "프로세서"는 앞서의 구조 또는 여기서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중의 어느 것을 말할 수 있다. 또한, 일부 양태들에서, 여기서 설명된 기능은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱(codec) 으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수 있다. 또한, 기법들은 하나 이상의 회로들 또는 로직 요소들 내에 완전히 구현될 수 있다.

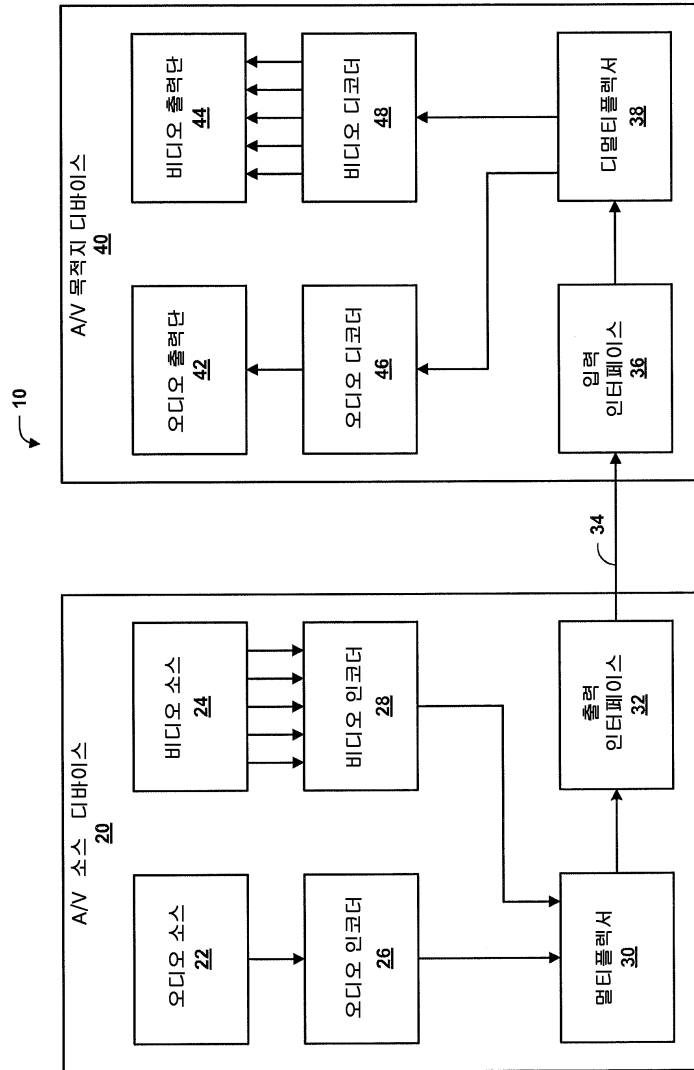
[0204] 이 개시물의 기법들은 무선 핸드셋, 집적회로 (IC) 또는 한 세트의 IC들 (예컨대, 칩 셋) 을 포함하여, 매우 다양한 디바이스들 또는 장치들에 구현될 수 있다. 다양한 구성요소들, 모듈들, 또는 유닛들이 개시된 기법들을 수행하도록 구성되지만 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는 디바이스들의 기능상 양태들을 강조하기 위해 이 개시물에서 설명된다. 약간 더 언급하면, 위에서 설명된 바와 같이, 다양한 유

닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수 있다.

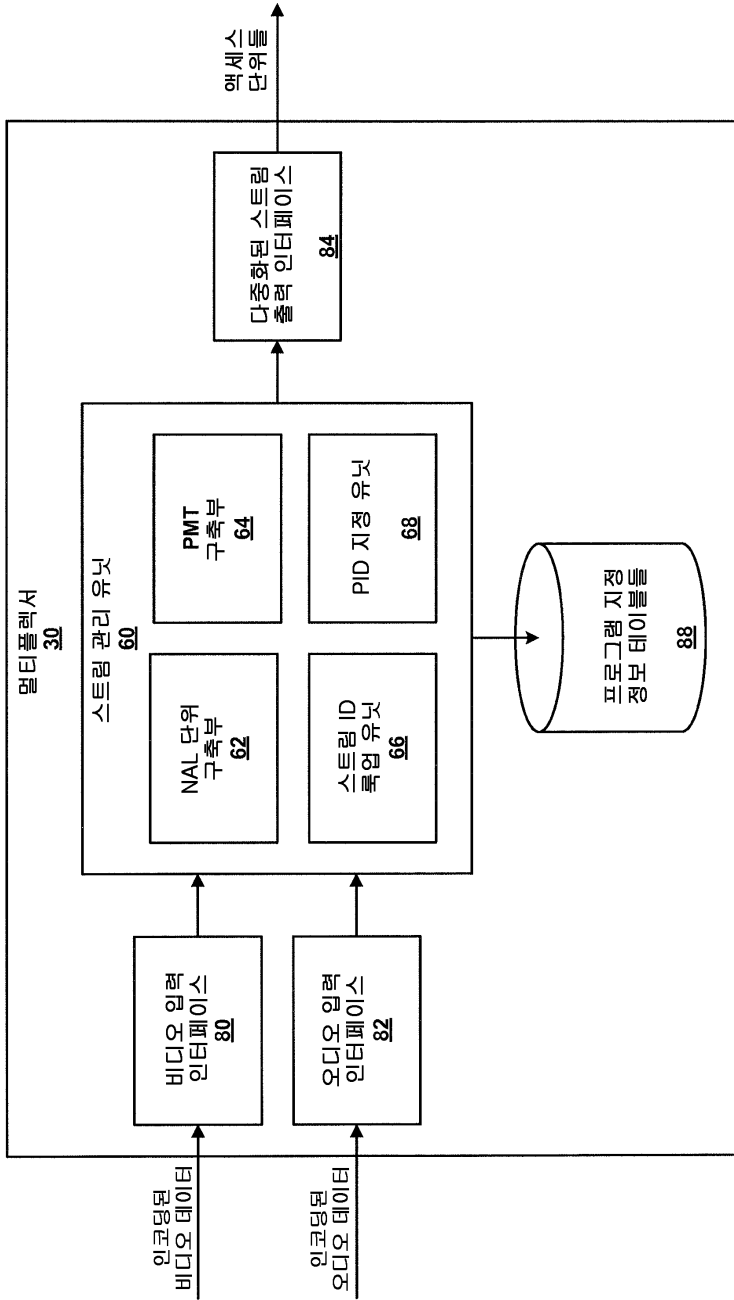
[0205] 여러 가지 예들이 설명되어 있다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

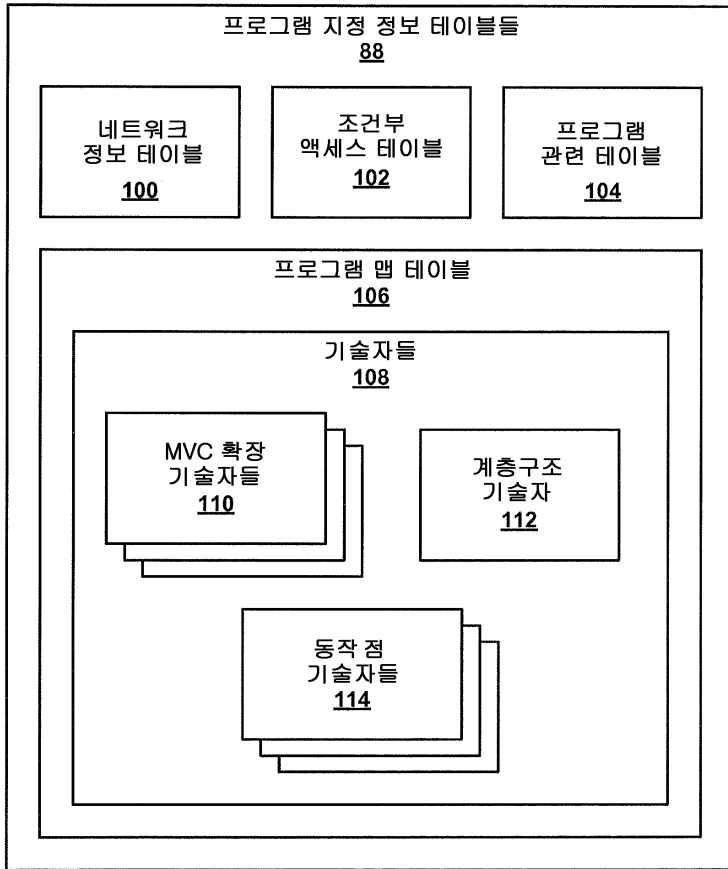
도면1



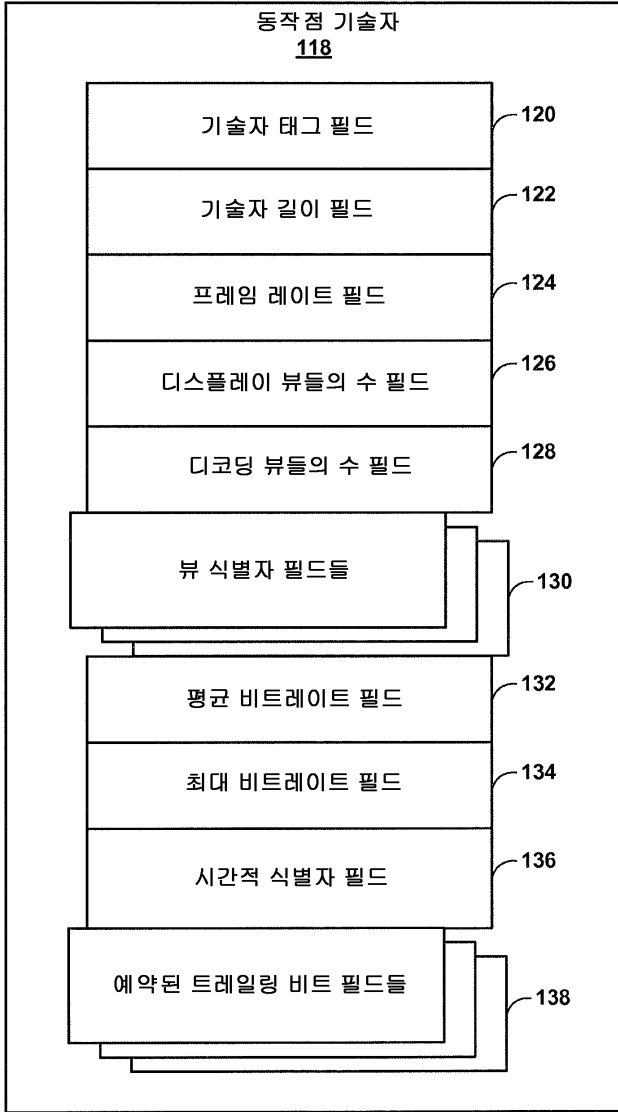
도면2



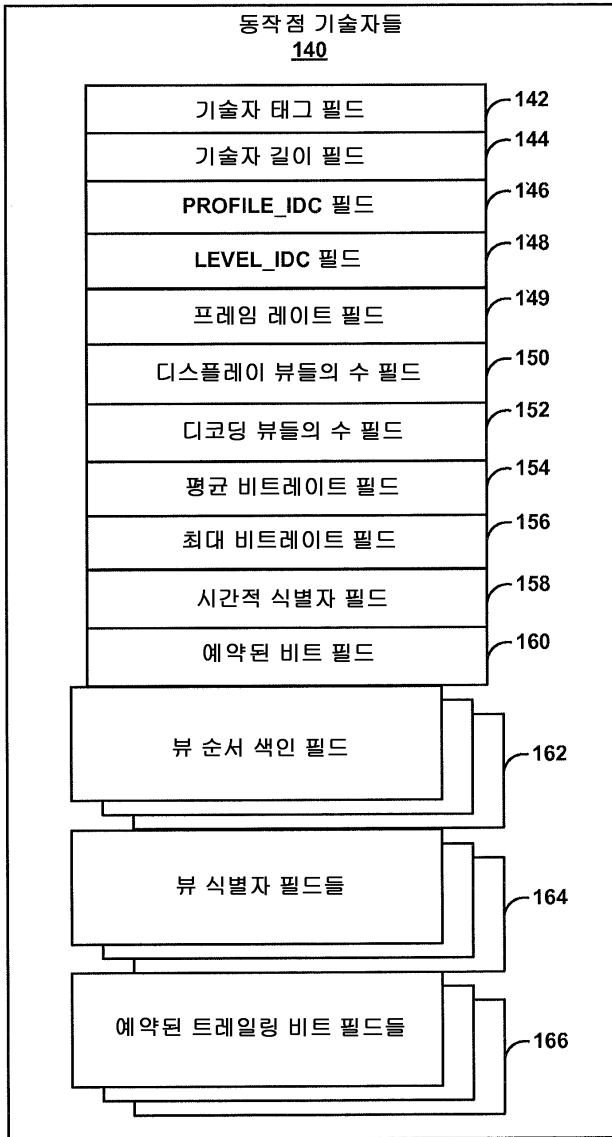
도면3



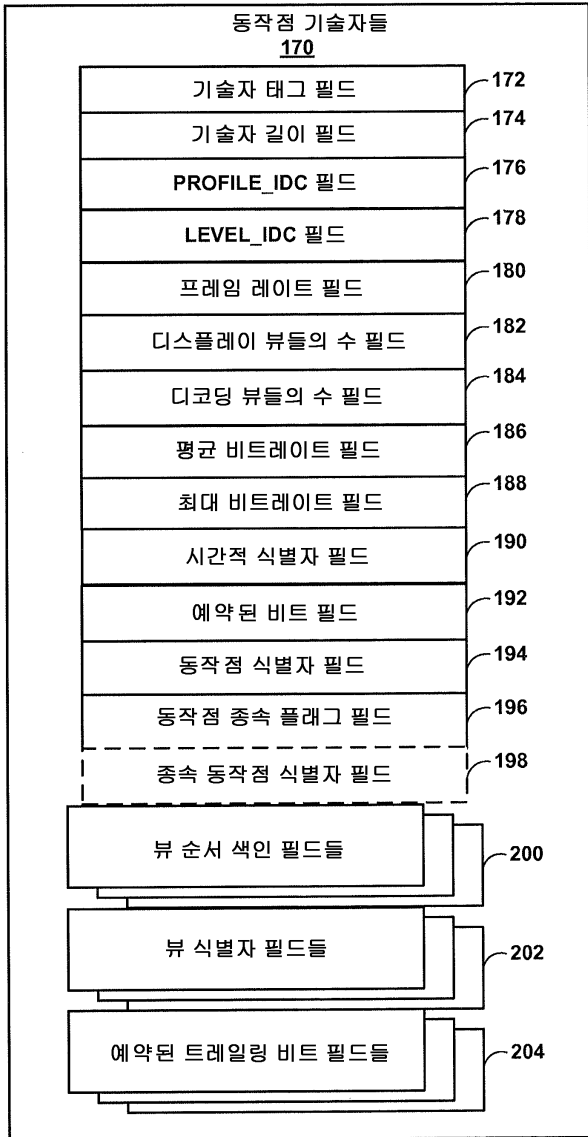
도면4



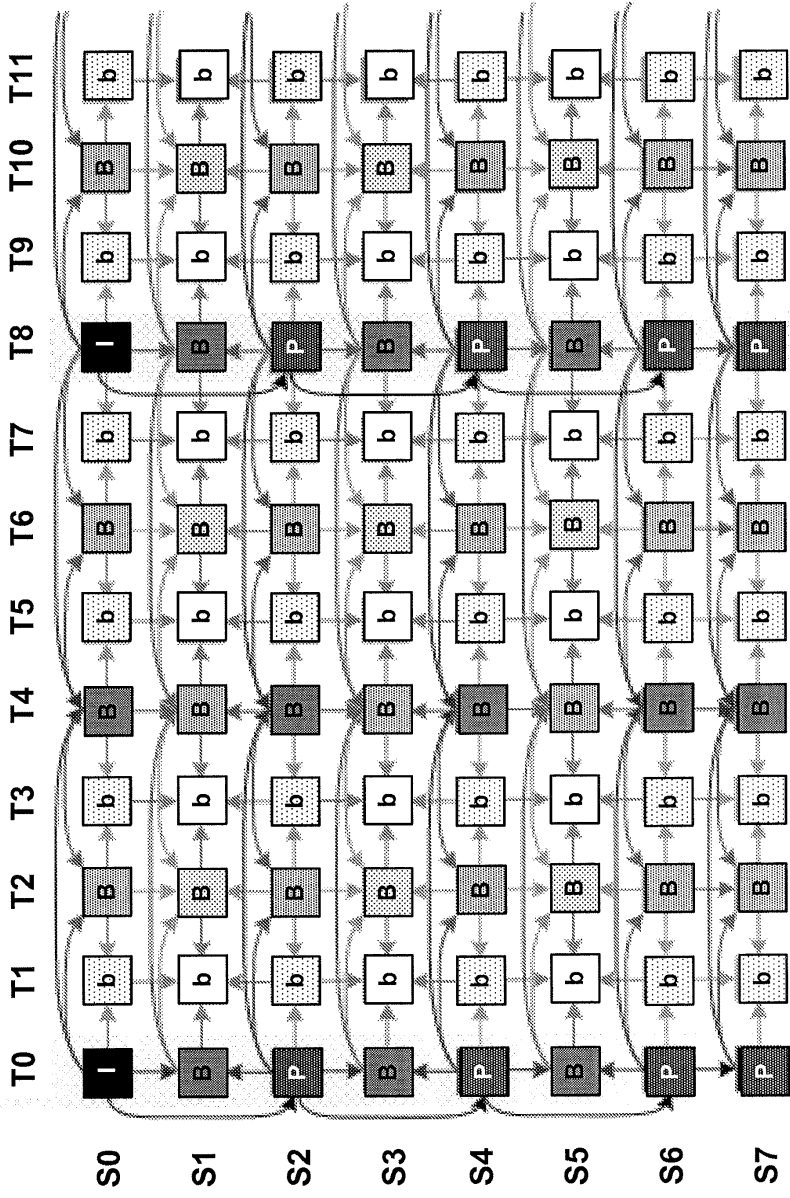
도면5



도면6



도면7





도면8

