



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2016년11월14일  
(11) 등록번호 10-1676059  
(24) 등록일자 2016년11월08일

- (51) 국제특허분류(Int. Cl.)  
H04N 13/00 (2016.01) H04N 19/112 (2014.01)  
H04N 19/132 (2014.01)
- (21) 출원번호 10-2011-7019803  
(22) 출원일자(국제) 2010년01월26일  
심사청구일자 2015년01월23일
- (85) 번역문제출일자 2011년08월25일  
(65) 공개번호 10-2011-0113752  
(43) 공개일자 2011년10월18일  
(86) 국제출원번호 PCT/US2010/000194  
(87) 국제공개번호 WO 2010/085361  
국제공개일자 2010년07월29일
- (30) 우선권주장  
61/205,938 2009년01월26일 미국(US)  
61/269,955 2009년07월01일 미국(US)
- (56) 선행기술조사문헌  
W02008127676 A2\*  
KR1020080112149 A\*  
KR1020080039797 A  
\*는 심사관에 의하여 인용된 문헌
- (73) 특허권자  
툼슨 라이선싱  
프랑스 이씨레플리노 뤼 잔 다르크 1-5 (우:  
92130)
- (72) 발명자  
티안, 동  
미국 뉴저지주 08536 플레인즈버로 소로 드라이브  
49  
라이, 폴린  
미국 뉴저지주 08536 플레인즈버로 웨일 릿지 드  
라이브 4906  
(뒷면에 계속)
- (74) 대리인  
양영준, 전경석, 백만기

전체 청구항 수 : 총 31 항

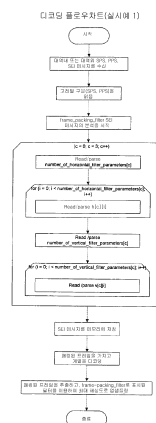
심사관 : 강성현

(54) 발명의 명칭 비디오 코딩을 위한 프레임 패킹

(57) 요약

예컨대 비디오 인코딩과 디코딩에서 뷰 타일링에 관한 구현이 제공된다. 특정 구현은 단일 화상으로 조합된 복수의 화상을 포함하는 비디오 화상에 액세스하며, 액세스된 비디오 화상 중의 복수의 화상을 어떻게 조합할 것인지 나타내는 부가 정보에 액세스한다. 액세스된 정보는 공간 인터리빙 정보와 샘플링 정보를 포함한다. 다른 구현은 단일 화상으로 조합된 복수의 화상을 포함하는 비디오 화상을 인코딩하며, 액세스된 비디오 화상 중의 복수의 화상을 어떻게 조합할 것인지를 나타내는 정보를 생성한다. 생성된 정보는 공간 인터리빙 정보와 샘플링 정보를 포함한다. 인코딩된 비디오 화상과 생성된 정보를 포함하는 비트스트림이 형성된다. 다른 구현은 생성된 정보를 전송하기 위한 데이터 구조를 제공한다.

대표도 - 도36



(72) 발명자

**루오, 지안콩**

미국 뉴저지주 08536 플레인즈버로 헌터스 글렌 드  
라이브 3404

**인, 평**

미국 뉴욕주 14850 이타카 존 스트리트 6

---

## 명세서

### 청구범위

#### 청구항 1

단일 화상으로 조합된 복수의 화상을 포함하며 수신된 비디오 스트림의 일부인 비디오 화상에 액세스하는 단계;

상기 수신된 비디오 스트림의 일부인 정보에 액세스하는 단계 - 상기 액세스된 정보는 상기 액세스된 비디오 화상 중의 복수의 화상이 어떻게 조합될 것인가를 나타내고, 상기 액세스된 정보는 SEI(Supplemental Enhancement Information) 메시지에 포함되고 공간 인터리빙 정보(spatial interleaving information)와 샘플링 정보를 포함하고, 상기 공간 인터리빙 정보는 상기 단일 화상을 구성하는데 있어 상기 복수의 화상에 적용된 공간 인터리빙을 나타내고, 상기 공간 인터리빙은 병렬 인터리빙(side-by-side interleaving), 상하 인터리빙(top-bottom interleaving), 행단위 인터리빙, 열단위 인터리빙 또는 체커판 인터리빙이고, 상기 샘플링 정보는 상기 복수의 화상 각각을 다른 해상도로 업샘플링하기 위한 업샘플링 필터에 관련된 하나 이상의 파라미터를 나타내고, 상기 업샘플링 필터에 관련된 상기 하나 이상의 파라미터는 필터링 방향의 표시를 포함하며, 상기 공간 인터리빙 정보는 관계 정보를 더 포함하고, 상기 관계 정보는, 상기 복수의 화상이 이미지의 입체 뷰인 것, 상기 복수의 화상이 관계가 없는 것, 상기 복수의 화상이 2D 이미지와 그 관련 깊이 맵(2D+Z)인 것, 상기 복수의 화상이 복수 세트의 2D+Z(MVD)인 것, 상기 복수의 화상이 이미지를 계층 깊이 비디오(layer depth video, LDV) 포맷으로 나타내는 것, 또는 상기 복수의 화상이 이미지를 2세트의 LDV(DES)로 나타내는 것을 나타냄 -; 및

상기 복수의 화상 중 적어도 하나의 디코딩된 표현(decoded representation)을 제공하기 위해 상기 비디오 화상을 디코딩하는 단계

를 포함하는 방법.

#### 청구항 2

제1항에 있어서,

상기 디코딩은 상기 공간 인터리빙 정보와 상기 샘플링 정보에 기초하여 수행되는 방법.

#### 청구항 3

삭제

#### 청구항 4

삭제

#### 청구항 5

삭제

#### 청구항 6

삭제

#### 청구항 7

제1항에 있어서,

상기 액세스된 정보는, 카메라의 초점 거리, 기준선 거리, 적어도 하나의 카메라 위치, 상기 복수의 화상 중의 이미지와 상기 카메라 간의 최소 거리를 나타내는 Znear 파라미터, 상기 복수의 화상 중의 상기 이미지와 상기 카메라 간의 최대 거리를 나타내는 Zfar 파라미터, 3×3 고유 행렬, 3×3 회전 행렬, 및 3D 변환 벡터 중 적어

도 하나를 포함하는, 카메라 파라미터 정보를 더 포함하는 방법.

#### 청구항 8

제7항에 있어서,

상기 디코딩은 상기 카메라 파라미터 정보에 더 기초하는 방법.

#### 청구항 9

제1항에 있어서,

상기 액세스된 정보는 상기 복수의 화상 중 적어도 하나의 배향(orientation)을 나타내는 배향 정보를 더 포함하는 방법.

#### 청구항 10

제9항에 있어서,

상기 디코딩은 상기 배향 정보에 더 기초하는 방법.

#### 청구항 11

제9항에 있어서,

상기 배향 정보는, 플립핑된 배향을 가진 화상이 없다는 것, 상기 복수의 화상 중 한 화상이 수직적으로 플립핑된 배향을 갖고 있다는 것, 및 상기 복수의 화상 중 한 화상이 수평적으로 플립핑된 배향을 갖고 있다는 것으로 이루어진 그룹 중에서 선택되는 방법.

#### 청구항 12

제11항에 있어서,

상기 배향 정보는 상기 배향 정보가 속하는 상기 복수의 화상 중 상기 하나 이상의 화상도 나타내는 방법.

#### 청구항 13

제1항에 있어서,

상기 샘플링 정보는 샘플링 표시, 샘플링 계수, 샘플링 필터 타입, 필터 계수의 수 및 하나 이상의 샘플링 필터 계수 값으로 이루어진 그룹 중에서 선택된 정보를 포함하는 방법.

#### 청구항 14

제13항에 있어서,

상기 샘플링 표시는 상기 샘플링 정보가 업샘플링 또는 다운샘플링에 관계하는지를 나타내는 방법.

#### 청구항 15

제13항에 있어서,

상기 샘플링 정보는 샘플링이 값 0, 2 및 4로 이루어진 그룹 중에서 선택되는 계수에 의해 수행된다는 것을 나타내는 샘플링 계수를 포함하는 방법.

#### 청구항 16

제13항에 있어서,

상기 샘플링 필터 타입은, 2차원 대칭 FIR 필터, 보간 필터, 바이리니어 필터, 큐빅 필터, 스플라인 필터, 비너(Wiener) 필터, 최소 제곱 필터, 6-탭 AVC 필터, 4-탭 SVC 필터, 2D 커스텀마이즈 필터를 포함하는 커스텀 필터, 1D 필터, 및 비선형 필터로 이루어진 그룹 중에서 선택된 샘플링 필터 타입을 나타내는 방법.

#### 청구항 17

제13항에 있어서,

상기 하나 이상의 샘플링 필터 계수 값은 업샘플링 필터 계수 값과 다운샘플링 필터 계수 값으로 이루어진 그룹 중에서 선택되는 방법.

#### 청구항 18

제13항에 있어서,

상기 디코딩은 상기 액세스된 정보에 포함된 다운샘플링 필터 계수 값으로부터 하나 이상의 업샘플링 필터 계수 값 세트를 도출하는 단계를 더 포함하는 방법.

#### 청구항 19

삭제

#### 청구항 20

제1항에 있어서,

필터링 방향의 상기 표시는 업샘플링이 수평 방향에서 수행되어야 하는지를 나타내는 방법.

#### 청구항 21

제1항에 있어서,

필터링 방향의 상기 표시는 업샘플링이 수직 방향에서 수행되어야 하는지를 나타내는 방법.

#### 청구항 22

제1항에 있어서,

필터링 방향의 상기 표시는 업샘플링이 수평 방향과 수직 방향에서 수행되어야 하는지를 나타내는 방법.

#### 청구항 23

제1항에 있어서,

상기 디코딩된 표현과 상기 액세스된 정보를 상기 액세스된 정보에 기초하여 상기 디코딩된 표현의 추가 처리를 위한 출력으로서 제공하는 단계를 더 포함하는 방법.

#### 청구항 24

삭제

#### 청구항 25

제1항에 있어서,

상기 디코딩된 표현을 디스플레이를 위해 제공하는 단계를 더 포함하는 방법.

#### 청구항 26

제1항에 있어서,

상기 업샘플링 필터에 관련된 상기 하나 이상의 파라미터는 상기 복수의 화상을 상기 단일 화상으로 조합하기 위해 상기 복수의 화상 중 하나 이상을 다운샘플링하는데 이용되는 필터 파라미터의 표시를 포함하는 방법.

#### 청구항 27

제26항에 있어서,

다운샘플링에 이용된 상기 필터 파라미터의 표시는 상기 디코딩된 표현을 제공하기 위한 업샘플링에 이용되는 업샘플링 필터 파라미터를 결정하는데 이용될 수 있는 방법.

#### 청구항 28

단일 화상으로 조합된 복수의 화상을 포함하며 수신된 비디오 스트림의 일부인 비디오 화상에 액세스하기 위한 수단;

상기 수신된 비디오 스트림의 일부인 정보에 액세스하기 위한 수단 - 상기 액세스된 정보는 상기 액세스된 비디오 화상 중의 복수의 화상이 어떻게 조합될 것인가를 나타내고, 상기 액세스된 정보는 SEI 메시지에 포함되고 공간 인터리빙 정보와 샘플링 정보를 포함하고, 상기 공간 인터리빙 정보는 상기 단일 화상을 구성하는데 있어 상기 복수의 화상에 적용된 공간 인터리빙을 나타내고, 상기 공간 인터리빙은 병렬 인터리빙, 상하 인터리빙, 행단위 인터리빙, 열단위 인터리빙 또는 체커판 인터리빙이고, 상기 샘플링 정보는 상기 복수의 화상 각각을 다른 해상도로 업샘플링하기 위한 업샘플링 필터에 관련된 하나 이상의 파라미터를 나타내고, 상기 업샘플링 필터에 관련된 상기 하나 이상의 파라미터는 필터링 방향의 표시를 포함하며, 상기 공간 인터리빙 정보는 관계 정보를 더 포함하고, 상기 관계 정보는, 상기 복수의 화상이 이미지의 입체 뷰인 것, 상기 복수의 화상이 관계가 없는 것, 상기 복수의 화상이 2D 이미지와 그 관련 깊이 맵(2D+Z)인 것, 상기 복수의 화상이 복수 세트의 2D+Z(MVD)인 것, 상기 복수의 화상이 이미지를 계층 깊이 비디오(LDV) 포맷으로 나타내는 것, 또는 상기 복수의 화상이 이미지를 2세트의 LDV(DES)로 나타내는 것을 나타냄 -; 및

상기 복수의 화상 중 적어도 하나의 디코딩된 표현을 제공하기 위해 상기 비디오 화상을 디코딩하기 위한 수단을 포함하는 장치.

#### 청구항 29

삭제

### 청구항 30

삭제

### 청구항 31

삭제

### 청구항 32

삭제

### 청구항 33

삭제

### 청구항 34

삭제

### 청구항 35

비디오 신호 구조를 저장한 프로세서 판독가능 매체로서,

상기 비디오 신호 구조는,

단일 화상으로 조합된 복수의 화상을 포함하는 비디오 화상의 인코딩을 포함하는 인코딩된 화상부; 및

상기 인코딩된 비디오 화상 중의 복수의 화상이 어떻게 조합되는가를 나타내는 발생된 정보의 인코딩을 포함하는 시그널링부를 포함하고,

상기 발생된 정보는 SEI 메시지에 포함되고 공간 인터리빙 정보와 샘플링 정보를 포함하고, 상기 공간 인터리빙 정보는 상기 단일 화상을 구성하는데 있어 상기 복수의 화상에 적용된 공간 인터리빙을 나타내고, 상기 공간 인터리빙은 병렬 인터리빙, 상하 인터리빙, 행단위 인터리빙, 열단위 인터리빙 또는 체커판 인터리빙이고, 상기 샘플링 정보는 상기 복수의 화상 각각을 다른 해상도로 업샘플링하기 위한 업샘플링 필터에 관련된 하나 이상의 파라미터를 나타내고, 상기 업샘플링 필터에 관련된 상기 하나 이상의 파라미터는 필터링 방향의 표시를 포함하며, 상기 공간 인터리빙 정보는 관계 정보를 더 포함하고, 상기 관계 정보는, 상기 복수의 화상이 이미지의 입체 뷰인 것, 상기 복수의 화상이 관계가 없는 것, 상기 복수의 화상이 2D 이미지와 그 관련 깊이 맵(2D+Z)인 것, 상기 복수의 화상이 복수 세트의 2D+Z(MVD)인 것, 상기 복수의 화상이 이미지를 계층 깊이 비디오(LDV) 포맷으로 나타내는 것, 또는 상기 복수의 화상이 이미지를 2세트의 LDV(DES)로 나타내는 것을 나타내고, 상기 발생된 정보는 상기 인코딩된 비디오 화상을 디코딩하는데 이용하기 위한 정보를 제공하는, 비디오 신호 구조를 저장한 프로세서 판독가능 매체.

### 청구항 36

단일 화상으로 조합된 복수의 화상을 포함하는 비디오 화상을 인코딩하는 단계(826);

상기 인코딩된 비디오 화상 중의 복수의 화상이 어떻게 조합되는가를 나타내는 정보를 발생시키는 단계 - 상기 발생된 정보는 SEI 메시지에 포함되고 공간 인터리빙 정보와 샘플링 정보를 포함하고, 상기 공간 인터리빙 정보는 상기 단일 화상을 구성하는데 있어 상기 복수의 화상에 적용된 공간 인터리빙을 나타내고, 상기 공간 인터리빙은 병렬 인터리빙, 상하 인터리빙, 행단위 인터리빙, 열단위 인터리빙 또는 체커판 인터리빙이고, 상기 샘플링 정보는 상기 복수의 화상 각각을 다른 해상도로 업샘플링하기 위한 업샘플링 필터에 관련된 하나 이상의 파라미터를 나타내고, 상기 업샘플링 필터에 관련된 상기 하나 이상의 파라미터는 필터링 방향의 표시를 포함하며, 상기 공간 인터리빙 정보는 관계 정보를 더 포함하고, 상기 관계 정보는, 상기 복수의 화상이 이미지의 입체 뷰인 것, 상기 복수의 화상이 관계가 없는 것, 상기 복수의 화상이 2D 이미지와 그 관련 깊이 맵(2D+Z)인 것, 상기 복수의 화상이 복수 세트의 2D+Z(MVD)인 것, 상기 복수의 화상이 이미지를 계층 깊이 비디오(LDV) 포맷으로 나타내는 것, 또는 상기 복수의 화상이 이미지를 2세트의 LDV(DES)로 나타내는 것을 나타냄 -;

및

상기 인코딩된 비디오 화상과, 상기 인코딩된 비디오 화상을 처리하는데 이용하기 위한 정보를 제공하는 상기 발생된 정보를, 포함하는 비트스트림을 형성하는 단계  
를 포함하는 방법.

#### 청구항 37

삭제

#### 청구항 38

삭제

#### 청구항 39

제36항에 있어서,

상기 발생된 정보는 상기 복수의 화상 중 적어도 하나의 배향을 나타내는 배향 정보를 더 포함하고,

상기 배향 정보는 상기 인코딩된 비디오 화상을 처리하는데도 이용되는 방법.

#### 청구항 40

제39항에 있어서,

상기 배향 정보는, 플립핑된 배향을 가진 화상이 없다는 것, 상기 복수의 화상 중 한 화상이 수직적으로 플립핑된 배향을 갖고 있다는 것, 및 상기 복수의 화상 중 한 화상이 수평적으로 플립핑된 배향을 갖고 있다는 것으로 이루어진 그룹 중에서 선택되는 방법.

#### 청구항 41

제36항에 있어서,

상기 필터링 방향의 표시는 업샘플링이 수평 방향에서 수행되어야 하는지를 나타내는 방법.

#### 청구항 42

제36항에 있어서,

상기 필터링 방향의 표시는 업샘플링이 수직 방향에서 수행되어야 하는지를 나타내는 방법.

#### 청구항 43

제36항에 있어서,

상기 필터링 방향의 표시는 업샘플링이 수평 방향과 수직 방향에서 수행되어야 하는지를 나타내는 방법.

#### 청구항 44

단일 화상으로 조합된 복수의 화상을 포함하는 비디오 화상을 인코딩하기 위한 수단;

상기 인코딩된 비디오 화상 중의 복수의 화상이 어떻게 조합되는가를 나타내는 정보를 발생시키기 위한 수단 -



상기 발생된 정보는 SEI 메시지에 포함되고 공간 인터리빙 정보와 샘플링 정보를 포함하고, 상기 공간 인터리빙 정보는 상기 단일 화상을 구성하는데 있어 상기 복수의 화상에 적용된 공간 인터리빙을 나타내고, 상기 공간 인터리빙은 병렬 인터리빙, 상하 인터리빙, 행단위 인터리빙, 열단위 인터리빙 또는 체커판 인터리빙이고, 상기 샘플링 정보는 상기 복수의 화상 각각을 다른 해상도로 업샘플링하기 위한 업샘플링 필터에 관련된 하나 이상의 파라미터를 나타내고, 상기 업샘플링 필터에 관련된 상기 하나 이상의 파라미터는 필터링 방향의 표시를 포함하며, 상기 공간 인터리빙 정보는 관계 정보를 더 포함하고, 상기 관계 정보는, 상기 복수의 화상이 이미지의 입체 뷰인 것, 상기 복수의 화상이 관계가 없는 것, 상기 복수의 화상이 2D 이미지와 그 관련 깊이 맵(2D+Z)인 것, 상기 복수의 화상이 복수 세트의 2D+Z(MVD)인 것, 상기 복수의 화상이 이미지를 계층 깊이 비디오(LDV) 포맷으로 나타내는 것, 또는 상기 복수의 화상이 이미지를 2세트의 LDV(DES)로 나타내는 것을 나타냄 -; 및

상기 인코딩된 비디오 화상과, 상기 인코딩된 비디오 화상을 처리하는데 이용하기 위한 정보를 제공하는 상기 발생된 정보를, 포함하는 비트스트림을 형성하기 위한 수단

을 포함하는 장치.

#### 청구항 45

삭제

#### 청구항 46

삭제

#### 청구항 47

삭제

#### 청구항 48

삭제

#### 청구항 49

단일 화상으로 조합된 복수의 화상을 포함하는 비디오 화상의 인코딩을 포함하는 인코딩된 화상부를 송신하기 위한 수단; 및

상기 인코딩된 비디오 화상 중의 복수의 화상이 어떻게 조합되는가를 나타내는 발생된 정보의 인코딩을 포함하는 시그널링부를 송신하기 위한 수단

을 포함하며,

상기 발생된 정보는 SEI 메시지에 포함되고 공간 인터리빙 정보와 샘플링 정보를 포함하고, 상기 공간 인터리빙 정보는 상기 단일 화상을 구성하는데 있어 상기 복수의 화상에 적용된 공간 인터리빙을 나타내고, 상기 공간 인터리빙은 병렬 인터리빙, 상하 인터리빙, 행단위 인터리빙, 열단위 인터리빙 또는 체커판 인터리빙이고, 상기 샘플링 정보는 상기 복수의 화상 각각을 다른 해상도로 업샘플링하기 위한 업샘플링 필터에 관련된 하나 이상의 파라미터를 나타내고, 상기 업샘플링 필터에 관련된 상기 하나 이상의 파라미터는 필터링 방향의 표시를 포함하며, 상기 공간 인터리빙 정보는 관계 정보를 더 포함하고, 상기 관계 정보는, 상기 복수의 화상이 이미지의 입체 뷰인 것, 상기 복수의 화상이 관계가 없는 것, 상기 복수의 화상이 2D 이미지와 그 관련 깊이 맵(2D+Z)인 것, 상기 복수의 화상이 복수 세트의 2D+Z(MVD)인 것, 상기 복수의 화상이 이미지를 계층 깊이 비디오(LDV) 포맷으로 나타내는 것, 또는 상기 복수의 화상이 이미지를 2세트의 LDV(DES)로 나타내는 것을 나타내고, 상기 발생된 정보는 상기 인코딩된 비디오 화상을 디코딩하는데 이용하기 위한 정보를 제공하는 장치.

#### 발명의 설명

#### 기술 분야

본 발명은 일반적으로 비디오 인코딩 및/또는 디코딩에 관한 것이다.

[0001]

[0002] 관련 출원의 상호 인용

[0003] 본 출원은 (1) 미국 임시특허 출원 제61/098,125호[출원일: 2009년 1월 26일, 발명의 명칭: "공간 인터리빙(Spatial Interleaving)", Attorney Docket PU090010] 및 (2) 미국 임시특허 출원 제61/269,955호[출원일: 2009년 9월 1일, 발명의 명칭: "프레임 패킹 응용을 위한 필터 파라미터 전달(Communicating Filter Parameters For Frame Packing Applications)", Attorney Docket PU090084]의 우선권을 주장한다, 이 임시특허 출원의 전체 내용은 본 명세서에 인용으로 포함된다.

[0004] 본 출원은 또한 국제특허 출원 PCT/US2008/004747[국제출원일: 2008년 4월 11일, 발명의 명칭: "비디오 인코딩 및 디코딩에서의 타일링(Tiling In Video Encoding And Decoding)"에 관계된다. 본 출원은 미국 특허출원 제 12/450,829호(Attorney Docket PU070078)로서 출원된 것이다.

## 배경 기술

[0005] 입체 디스플레이와 자동 입체 디스플레이와 같은 3D 디스플레이의 출시에 따라서 더 많은 3D 콘텐츠가 강하게 요구되고 있다. 보통은 다중뷰(multiple views)와 이에 대응하는 깊이 맵을 포함하는 3D 콘텐츠를 코딩하는 것은 통상적으로 어려운 작업이다. 3D 콘텐츠의 각 프레임에 대해서 시스템은 엄청난 양의 데이터를 처리해야 한다. 통상적인 3D 비디오 애플리케이션에서는 예컨대 전송 대역폭 한계, 저장 한계 및 처리 한계로 인해 다중뷰 비디오 신호를 효율적으로 전송 또는 저장할 필요가 있다. 다중뷰 비디오 코딩(MVC)은 다중뷰의 코딩을 용이하게 하는 고레벨 구문을 이용하여 H.264/Advanced Video Coding(AVC)을 확장한다. 이 구문은 이미지 프로세서에 의한 3D 이미지 처리를 지원한다.

[0006] H.264/AVC는 표면적으로 2D 비디오용으로 설계되어 있지만 프레임 패킹 기법을 이용하면 입체 콘텐츠를 전송하는데도 이용될 수 있다. 프레임 패킹 기법은 간단하게 설명하자면 다음과 같다. 인코더측에서는 2개의 뷰 또는 화상이 일반적으로 패킹을 위해 다운샘플링되어 하나의 단일 비디오 프레임이 되고, 이 프레임은 H.264/AVC 인코더에 공급되어 비트스트림으로서 출력되고, 디코더측에서 이 비트스트림은 디코딩되고, 복구된 프레임이 언패킹된다(unpacked). 언패킹은 복구된 프레임으로부터의 2개의 원래 뷰의 추출을 가능하게 하며 일반적으로 뷰가 디스플레이를 위해 렌더링될 수 있도록 각 뷰의 원래 크기를 복구하는 업샘플링 동작을 포함한다. 이 방식은 다중뷰 이미지, 깊이 정보 등과 같은 2개 이상의 뷰에 이용될 수 있다.

[0007] 프레임 패킹은 프레임 및 그 뷰와 연관된 보조 정보의 존재에 의존할 수 있다. SEI(Supplemental Enhancement Information) 메시지는 프레임 패킹 정보를 전달하는데 이용될 수 있다. 일례로서, AVC 초안 개정에서는 SEI 메시지를 디코더에게 구성 화상이 체커판 공간 인터리빙으로 구성되는 것을 포함하여 패킹된 화상의 여러 가지 공간 인터리빙 특성을 알리는데 이용할 수 있음을 제시하였다. SEI 메시지를 이용하면 AVC를 직접 이용하여 입체 비디오 이미지의 체커판 인터리빙된 화상을 인코딩할 수 있다. 도 26은 체커판 인터리빙의 알려진 예를 보여준다. 그러나 지금까지는 SEI 메시지 콘텐츠와 다른 고레벨 구문의 콘텐츠는 프레임 패킹되었던 화상이나 뷰에 관계된 정보를 전달하는데 제약이 있었다.

## 발명의 내용

### 해결하려는 과제

[0008] 본 발명의 목적은 종래의 문제점을 개선하는 데에 있다.

### 과제의 해결 수단

[0009] 일반적인 양상에 따라서, 단일 화상으로 조합된 복수의 화상을 포함하는 비디오 화상이 인코딩된다. 액세스된 비디오 화상 중의 복수의 화상을 어떻게 조합할 것인가를 나타내는 정보가 생성된다. 상기 생성된 정보는 공간 인터리빙 정보와 샘플링 정보를 포함한다. 상기 공간 인터리빙 정보는 단일 화상을 구성하는데 있어 복수의 화상에 적용된 공간 인터리빙을 나타낸다. 상기 샘플링 정보는 복수의 화상 각각을 원하는 해상도로 복구하기 위한 업샘플링 필터에 관련된 하나 이상의 파라미터를 나타낸다. 업샘플링 필터에 관련된 상기 하나 이상의 파라

미터는 필터링 방향의 표시를 포함한다. 상기 인코딩된 비디오 화상과 상기 생성된 정보를 포함하는 비트스트림이 형성된다. 상기 생성된 정보는 상기 인코딩된 비디오 화상을 처리하는데 이용되는 정보를 제공한다.

[0010] 다른 일반적인 양상에 따라서, 비디오 신호 또는 비디오 구조는 인코딩된 화상부와 시그널링부를 포함한다. 상기 인코딩된 화상부는 단일 화상으로 조합된 복수의 화상을 포함하는 비디오 화상의 인코딩을 포함한다. 상기 시그널링부는 상기 액세스된 비디오 화상 중의 복수의 화상을 어떻게 조합할 것인가를 나타내는 발생된 정보의 인코딩을 포함한다. 상기 발생된 정보는 공간 인터리빙 정보와 샘플링 정보를 포함한다. 상기 공간 인터리빙 정보는 상기 단일 화상을 구성하는데 있어 상기 복수의 화상에 적용된 공간 인터리빙을 나타낸다. 상기 샘플링 정보는 상기 복수의 화상 각각을 원하는 해상도로 복구하기 위한 업샘플링 필터에 관련된 하나 이상의 파라미터를 나타낸다. 업샘플링 필터에 관련된 상기 하나 이상의 파라미터는 필터링 방향의 표시를 포함한다. 상기 발생된 정보는 상기 인코딩된 비디오 화상을 디코딩하는데 이용하기 위한 정보를 제공한다.

[0011] 다른 일반적인 양상에 따라서, 단일 화상으로 조합된 복수의 화상을 포함하며 수신된 비디오 스트림의 일부인 비디오 화상이 액세스된다. 상기 수신된 비디오 스트림의 일부인 정보가 액세스되며, 상기 액세스된 정보는 상기 액세스된 비디오 화상 중의 복수의 화상을 어떻게 조합할 것인가를 나타낸다. 상기 액세스된 정보는 공간 인터리빙 정보와 샘플링 정보를 포함한다. 상기 공간 인터리빙 정보는 상기 단일 화상을 구성하는데 있어 상기 복수의 화상에 적용된 공간 인터리빙을 나타낸다. 상기 샘플링 정보는 상기 복수의 화상 각각을 원하는 해상도로 복구하기 위한 업샘플링 필터에 관련된 하나 이상의 파라미터를 나타낸다. 업샘플링 필터에 관련된 상기 하나 이상의 파라미터는 필터링 방향의 표시를 포함한다. 상기 비디오 화상은 상기 복수의 화상 중 적어도 하나의 디코딩된 표현을 제공하도록 디코딩된다.

[0012] 하나 이상의 구현에 대해 자세한 것은 첨부도면과 하기의 상세한 설명에서 설명된다. 구현들은 어떤 특정한 방식으로 설명되더라도 다른 여러 가지 방식으로 구성 또는 구체화될 수 있다. 예컨대 구현은 방법으로서 수행되거나, 일련의 동작을 수행하도록 구성된 장치로서 구체화되거나, 일련의 동작을 수행하기 위한 명령어를 저장하는 장치로서 구체화되거나, 어떤 신호로 구체화될 수 있다.

[0013] 다른 양상과 특징들은 첨부 도면 및 청구범위와 함께 하기의 상세한 설명으로부터 명백하게 드러날 것이다.

## 발명의 효과

[0014] 본 발명에 따르면 종래의 문제점을 개선할 수 있다.

## 도면의 간단한 설명

[0015] 도 1은 단일 프레임에 타일링된 4개의 뷰의 예를 보여주는 도.

도 2는 단일 프레임에 플립핑되고 타일링된 4개의 뷰의 예를 보여주는 도.

도 3은 본 발명의 원리가 적용될 수 있는 비디오 인코더의 구현을 도시한 블록도.

도 4는 본 발명의 원리가 적용될 수 있는 비디오 디코더의 구현을 도시한 블록도.

도 5는 MPEG-4 AVC 표준을 이용하여 복수의 뷰에 대한 화상을 인코딩하는 방법의 구현을 위한 흐름도.

도 6은 MPEG-4 AVC 표준을 이용하여 복수의 뷰에 대한 화상을 디코딩하는 방법의 구현을 위한 흐름도.

도 7은 MPEG-4 AVC 표준을 이용하여 복수의 뷰와 깊이에 대한 화상을 인코딩하는 방법의 구현을 위한 흐름도.

도 8은 MPEG-4 AVC 표준을 이용하여 복수의 뷰와 깊이에 대한 화상을 디코딩하는 방법의 구현을 위한 흐름도.

도 9는 깊이 신호의 예를 보여주는 도.

도 10은 타일로서 부가된 깊이 신호의 예를 보여주는 도.

도 11은 단일 프레임에 타일링된 5개의 뷰의 예를 보여주는 도.

도 12는 본 발명의 원리가 적용될 수 있는 예시적인 MVC(Multi-view Video Coding) 인코더의 블록도.

도 13은 본 발명의 원리가 적용될 수 있는 예시적인 MVC(Multi-view Video Coding) 디코더의 블록도.

- 도 14는 MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 화상을 인코딩하는 것에 대한 준비로서 복수의 뷰에 대한 화상을 처리하는 방법의 구현을 위한 흐름도.
- 도 15는 MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 복수의 뷰에 대한 화상을 인코딩하는 방법의 구현을 위한 흐름도.
- 도 16은 MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 화상을 디코딩하는 것에 대한 준비로서 복수의 뷰에 대한 화상을 처리하는 방법의 구현을 위한 흐름도.
- 도 17은 MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 복수의 뷰에 대한 화상을 디코딩하는 방법의 구현을 위한 흐름도.
- 도 18은 MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 화상을 인코딩하는 것에 대한 준비로서 복수의 뷰와 깊이에 대한 화상을 처리하는 방법의 구현을 위한 흐름도.
- 도 19는 MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 복수의 뷰와 깊이에 대한 화상을 인코딩하는 방법의 구현을 위한 흐름도.
- 도 20은 MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 화상을 디코딩하는 것에 대한 준비로서 복수의 뷰와 깊이에 대한 화상을 처리하는 방법의 구현을 위한 흐름도.
- 도 21은 MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 복수의 뷰와 깊이에 대한 화상을 디코딩하는 방법의 구현을 위한 흐름도.
- 도 22는 픽셀 레벨에서의 타일링 예를 보여주는 도.
- 도 23은 본 발명의 원리가 적용될 수 있는 비디오 처리 장치의 구현을 위한 블록도.
- 도 24는 예시적인 3D 비디오 시스템의 간략도.
- 도 25는 여러 가지 기준 뷰 중의 이미지에 대한 예시적인 좌측 및 우측 깊이 맵을 보여주는 도.
- 도 26은 체커판 인터리빙을 이용하여 2개의 구성 화상을 공간적으로 인터리빙하여 단일 화상 또는 프레임으로 만들기 위한 예시적인 블록도.
- 도 27은 2개의 구성 화상의 병렬적 공간 인터리빙의 예시적인 화상을 보여주는 도.
- 도 28은 2개의 구성 화상의 상하 공간 인터리빙의 예시적인 화상을 보여주는 도.
- 도 29는 2개의 구성 화상의 행단위 공간 인터리빙의 예시적인 화상을 보여주는 도.
- 도 30은 2개의 구성 화상의 열단위 공간 인터리빙의 예시적인 화상을 보여주는 도.
- 도 31은 좌측 화상이 수평적으로 플립되는 2개의 구성 화상의 병렬적 공간 인터리빙의 예시적인 화상을 보여주는 도.
- 도 32는 하단 화상이 수직적으로 플립되는 2개의 구성 화상의 상하 공간 인터리빙의 예시적인 화상을 보여주는 도.
- 도 33은 구성 화상이 LDV(Layer Depth Video) 포맷을 나타내는 예시적인 인터리빙된 화상 또는 프레임을 보여주는 도.
- 도 34는 구성 화상이 2D 플러스 깊이 포맷을 나타내는 예시적인 인터리빙된 화상 또는 프레임을 보여주는 도.
- 도 35 내지 38은 프레임 패킹 정보에 대한 SEI 메시지를 이용하여 비디오 이미지의 인코딩과 디코딩을 처리하는 여러 가지 실시예의 예시적인 플로우차트.
- 도 39는 본 발명의 원리가 적용될 수 있는 예시적인 비디오 송신 시스템을 도시한 도.
- 도 40은 본 발명의 원리가 적용될 수 있는 예시적인 비디오 수신 시스템을 도시한 도.
- 도 41은 본 발명의 원리가 적용될 수 있는 예시적인 비디오 처리 장치를 도시한 도.
- 여기서 설명되는 여러 가지 예시적인 실시예들은 본 발명의 범위를 한정하는 것으로 해석해서는 안 된다.

### 발명을 실시하기 위한 구체적인 내용

- [0016] 여러 가지 구현들은 비디오 인코딩과 디코딩에서 뷰 타일링 방법 및 장치에 관한 것이다. 따라서 당업자라면 여기에 명시적으로 기술되고 도시되지는 않아도 본 발명을 구현하고 그 본질과 범위 내에 있는 각종 구성을 고안해낼 수 있을 것이다.
- [0017] 여기서 설명되는 예와 조건적 표현들은 모두 본 발명자들이 기술 발전의 촉진에 기여한 본 발명의 원리와 개념을 이해시키는데 도움을 주고자 한 것이며, 본 발명은 그러한 특성의 예와 조건에 의해 제한되는 것으로 해석해서는 안된다.
- [0018] 더욱이 여기서 본 발명의 원리, 양상 및 실시에는 물론 그 구체적인 예에 대한 모든 설명은 그 구조적 기능적 등가물을 포함하는 것이다. 게다가 그와 같은 등가는 현재 알려져 있는 등가물은 물론이고 장래에 개발될 등가물, 즉 구조에 상관없이 동일 기능을 수행하는 개발될 구성요소를 모두 포함하는 것이다.
- [0019] 따라서, 예컨대 당업자라면 여기서 제시된 블록도들이 본 발명의 원리를 구체화하는 예시적인 회로의 개념도임을 잘 알 것이다. 마찬가지로 플로우차트, 흐름도, 상태 천이도, 의사코드 등은 컴퓨터 판독 매체에 실제적으로 표현되고, 컴퓨터나 프로세서에 의해(이런 컴퓨터나 프로세서가 명시적으로 나타나 있는지 여부에 관계없이) 실행될 수 있는 여러 가지 프로세스를 표현하는 것임을 잘 알 것이다.
- [0020] 도면에 도시된 여러 가지 구성요소의 기능은 전용 하드웨어는 물론 적당한 소프트웨어와 관련하여 소프트웨어를 실행할 수 있는 하드웨어를 통해 제공될 수 있다. 프로세서가 제공하는 이 기능들은 단일의 전용 프로세서, 단일의 공유 프로세서, 복수의 개별적 프로세서(그 일부는 공유될 수 있음)에 의해 제공될 수 있다. 더욱이 "프로세서"나 "컨트롤러"라는 용어를 명시적으로 사용하더라도 이는 소프트웨어를 실행할 수 있는 하드웨어에만 국한되는 것으로 해석해서는 안되고, 아무런 제한없이, DSP(digital signal processor) 하드웨어, 소프트웨어를 저장하는 ROM(read-only memory), RAM(random access memory), 및 불휘발성 저장장치를 암시적으로 포함하는 것이다.
- [0021] 기타 다른 종래의 그리고/또는 주문형 하드웨어도 여러 가지 구현에 포함될 수 있다. 예컨대 도면에 도시된 스위치들은 단지 개념적인 것이다. 이들의 기능은 프로그램 로직의 동작을 통해, 전용 로직을 통해, 프로그램 컨트롤 및 전용 로직의 상호작용을 통해, 또는 수동적으로도 실시될 수 있으며, 그 특정 기술은 본 명세서의 내용으로부터 더 구체적으로 이해되는 바와 같이 구현자에 의해 선택될 수 있다.
- [0022] 청구범위에서 특정 기능을 수행하기 위한 수단으로 표현된 구성요소는, 예컨대 a) 그 기능을 수행하는 회로 구성요소들의 조합, 또는 b) 소프트웨어를 실행하여 그 기능을 수행하는 적당한 회로와 결합된 펌웨어, 마이크로코드 등을 포함한 임의 형태의 소프트웨어를 포함하여 그 기능을 수행하는 임의의 방식을 포함하는 것이다. 그와 같은 청구범위에 의해 정해지는 본 발명의 원리는 이러한 여러 가지 설명된 수단이 제공하는 기능들이 청구범위가 필요로 하는 방식으로 조합되고 합체된다는 사실에 있다. 따라서 이들 기능을 제공할 수 있는 수단은 여기서 보여진 것들과 등가인 것으로 간주된다.
- [0023] 본 명세서에서 본 발명의 "일 실시예(또는 일 구현)"나 "소정 실시예(또는 소정 구현)"라는 말은 그 실시예와 관련하여 설명된 특정 형상, 구조, 특성 등이 본 발명의 적어도 하나의 실시예에 포함된다는 것을 의미한다. 따라서 본 명세서의 여러 곳에서 나타내는 "일 실시예에서"나 "소정 실시예에서"라는 구절은 반드시 같은 실시예를 말하는 것은 아니다.
- [0024] 본 발명의 하나 이상의 실시예는 여기서는 MPEG-4 AVC 표준과 관련하여 설명되지만 본 발명은 이 표준에만 한정되는 것은 아니고 본 발명의 본질을 그대로 유지하면서 다른 비디오 코딩 표준, 권고안, 그리고 MPEG-4 AVC 표준의 확장을 포함한 여러 확장과 관련해서 이용될 수 있음을 알아야 한다.
- [0025] 더욱이, 하나 이상의 다른 실시예는 여기서는 MPEG-4 AVC 표준의 다중뷰 비디오 코딩 확장에 대해 설명하지만, 본 발명은 이 확장 및/또는 이 표준에만 한정되는 것이 아니라, 본 발명의 본질을 그대로 유지하면서 다중뷰 비디오 코딩과 관한 다른 비디오 코딩 표준, 권고안 및 그 확장에 대해서도 이용될 수 있음을 알아야 한다. MVC(Multi-view Video Coding)는 다중뷰 계열의 인코딩을 위한 압축 체계이다. MVC(Multi-view Video Coding) 계열은 다른 시점(view point)으로부터 동일 장면을 포착하는 하나 이상의 비디오 시퀀스 세트이다.
- [0026] 또한, 하나 이상의 다른 실시예는 여기서는 비디오 콘텐츠에 대한 깊이 정보를 이용하는 것으로 설명하지만, 본 발명의 원리는 그와 같은 실시예에 한정되는 것이 아니며, 본 발명의 본질을 그대로 유지하면서 깊이 정보를 이



용하지 않는 다른 실시예도 구현될 수 있음을 알아야 한다.

- [0027] 그 외에도, 여기서 사용된 "고레벨 구분"이란 마크로블록 계층 위에 계층적으로 존재하는 비트스트림에 존재하는 구분을 말한다. 예컨대, 여기서 사용된 고레벨 구분은 슬라이스 헤더 레벨, SEI(Supplemental Enhancement Information) 레벨, PPS(Picture Parameter Set) 레벨, SPS(Sequence Parameter Set), VPS(View Parameter Set) 레벨 및 NAL(Network Abstraction Layer) 단위 헤더 레벨에서의 구분을 말할 수 있으나 이에 한정되는 것은 아니다.
- [0028] ISO/IEC(International Organization for Standardization/International Electrotechnical Commission) MPEG-4(Moving Picture Experts Group-1) Part 10 AVC(Advanced Video Coding) 표준/ITU-T(International Telecommunication Union, Telecommunication Sector) H.264 권고안(이하, "MPEG-4 VAC 표준")에 기초한 MVC(Multi-Video Coding)의 현재 구현에서는, 기준 소프트웨어는 단일 인코더를 가지고 각 뷰를 인코딩하고 크로스 뷰 기준을 고려함으로써 다중뷰 예측을 달성한다. 각 뷰는 그 원 해상도에서 인코더에 의해 별도의 비트스트림으로서 코딩되고, 그 후에 모든 비트스트림이 조합되어 단일의 비트스트림을 구성하여 디코딩된다. 각 뷰는 독립적인 YUV 디코딩된 출력을 생성한다.
- [0029] 3D 이미지의 작성과 이용을 지원하는 예시적인 비디오 시스템은 도 24에 개략적으로 도시되어 있다. 이 시스템의 콘텐츠 작성측은 입체 카메라, 깊이 카메라, 동시 작동하는 복수의 카메라, 및 2D 이미지의 3D 이미지로의 변환을 포함하는(이에 한정되지 않음) 여러 가지 수단에 의한 이미지 캡처를 보여준다. 동일 장면의 좌측 및 우측 뷰에 대해 캡처된 깊이 맵 정보(예컨대 Z 정보)의 일례는 도 25에 나타나 있다. 이들 방식 각각은 비디오 이미지 콘텐츠를 캡처하는 것 뿐만 아니라, 일부는 캡처된 비디오 이미지와 연관된 특정 깊이 정보를 생성한다. 이 모든 정보는 처리 및 코딩되면 분배, 전송, 최종적으로 렌더링될 수 있다. 메타데이터도 3D 비디오의 후속 렌더링에 이용하기 위해 비디오 콘텐츠와 함께 생성된다. 렌더링은 2D 디스플레이 시스템 또는 3D 디스플레이를 이용하여 행해질 수 있다. 3D 디스플레이는 도면에 도시된 바와 같이 입체 디스플레이부터 다중뷰 3D 디스플레이에 이르기까지 다양할 수 있다.
- [0030] 다중뷰 예측을 위한 다른 방식은 뷰 세트를 의사(pseudo) 뷰들로 그룹화하는 것이다. 이 방식의 일례로서, 다운샘플링 또는 다른 동작을 이용하여 더 큰 프레임 또는 슈퍼 프레임에서 (동시에 샘플링된) 총 M개의 뷰 중에서 N개의 뷰마다 화상을 타일링할 수 있다. 도 1을 참조로 설명하면, 단일 프레임 상에 타일링된 4개의 뷰의 일례가 도면부호 100으로 총괄 표시되어 있다. 4개 뷰 모두 정상 배향을 이루고 있다.
- [0031] 도 2를 참조로 설명하면, 단일 프레임 상에 플립핑(flippping) 및 타일링된 4개의 뷰의 일례가 도면부호 200으로 총괄 표시되어 있다. 상단 좌측 뷰는 정상 배향을 이루고 있다. 상단 우측 뷰는 수평적으로 플립핑되어 있다. 하단 좌측 뷰는 수직적으로 플립핑되어 있다. 하단 우측 뷰는 수평적으로 그리고 수직적으로 플립핑되어 있다. 따라서 4개의 뷰가 있다면 각 뷰로부터의 화상은 타일과 같은 슈퍼 프레임으로 배치된다. 이에 따라서 더 큰 해상도를 가진 단일의 미코딩 입력 계열이 생성된다.
- [0032] 대안으로서, 더 작은 해상도를 생성하기 위해 이미지를 다운샘플링할 수 있다. 이에 따라서 각각이 함께 타일링된 여러 가지 뷰들을 포함하는 복수의 계열을 생성할 수 있다. 그러면 그와 같은 각 계열은 N개의 서로 다른 타일링된 뷰를 포함하는 의사 뷰를 구성한다. 도 1은 한 의사 뷰를 보여주고, 도 2는 다른 의사 뷰를 보여준다. 이들 의사 뷰는 ISO/IEC MPEG-2 표준과 MPEG-4 AVC 표준과 같은 기존의 비디오 코딩 표준을 이용하여 인코딩될 수 있다.
- [0033] 다중뷰 코딩을 위한 또 다른 방식은 새로운 표준을 이용하여 여러 뷰를 독립적으로 인코딩하고, 디코딩 후에 플레이어의 필요에 따라서 뷰를 타일링하는 것이다.
- [0034] 더욱이, 다른 방식에서는 뷰는 픽셀 방식으로 타일링될 수도 있다. 예컨대, 4개의 뷰로 구성된 슈퍼 뷰에서, 픽셀(x,y)은 뷰 0부터, 픽셀(x+1,y)은 뷰 1부터, 픽셀(x,y+1)은 뷰 2부터, 픽셀(x+1,y+1)은 뷰 3부터일 수 있다.
- [0035] 많은 디스플레이 제조업자는 단일 프레임 상에 여러 개의 뷰를 배열 또는 타일링하고 그런 다음에 각자의 위치로부터 뷰를 추출하여 렌더링하는 체계를 이용한다. 그와 같은 경우에는 비트스트림이 그러한 특성을 갖는지를 판단하는 표준적인 방법은 없다. 따라서 시스템이 큰 프레임에서 여러 가지 뷰의 화상을 타일링하는 방법을 이용한다면 그 여러 가지 뷰를 추출하는 방법은 전유(專有)한 것이다.
- [0036] 그러나, 비트스트림이 그러한 특성을 갖는지를 판단하는 표준적인 방법은 없다. 본 발명에서는, 렌더러 또는 플레이어가 디스플레이 또는 다른 후처리를 지원하기 위해 그와 같은 정보를 용이하게 추출할 수 있도록 하는

고레벨 구문을 제시한다. 또한 서브화상들이 서로 다른 해상도를 갖도록 하는 것이 가능하며, 뷰를 최종적으로 렌더링하는데는 업샘플링이 필요할 수 있다. 사용자는 마찬가지로 고레벨 구문에 나타난 업샘플링 방법을 이용하기를 원할 수 있다. 그 외에도, 깊이 초점을 변경하는 파라미터도 전송될 수 있다.

[0037] 일 실시예에서, 각 화상이 다른 뷰에 속하는 서브화상들을 포함하는 MPEG-4 AVC 표준 양립 비트스트림에서 다중 뷰 정보를 발생하기 위한 새로운 SEI(Supplemental Enhancement Information) 메시지를 제시한다. 이 실시예는 예컨대 그와 같은 체계를 이용할 수 있는 3차원(3D) 모니터 상의 다중뷰 비디오 스트림의 디스플레이를 용이하고 편리하게 하기 위한 것이다. 이 개념은 고레벨 구문을 이용하여 그와 같은 정보를 발생하는 다른 비디오 코딩 표준과 권고안으로 확장될 수 있다.

[0038] 더욱이, 일 실시예에서, 뷰를 다중뷰 비디오 카메라 인코더 및/또는 디코더에 보내기 전에 뷰를 배열하는 시그널링(signaling) 방법을 제시한다. 양호하게는, 이 실시예는 다중뷰 코딩의 구현을 간단화할 수 있으며, 코딩 효율을 개선할 수 있다. 특정 뷰는 함께 묶여져 의사 뷰 또는 슈퍼 뷰를 구성할 수 있으며, 그러면 타일링된 슈퍼 뷰는 예컨대 다중뷰 비디오 코딩의 현재 MPEG-4 AVC 표준 기반 구현에 따라서 보통의 다중뷰 비디오 인코더 및/또는 디코더에 의해 정상적인 뷰로 취급된다. 표 1에 나타난 새로운 플래그는 의사 뷰 기법의 이용을 표시하는 다중뷰 비디오 코딩의 SPS(Sequence Parameter Set) 확장에 제시된다. 이 실시예는 그와 같은 체계를 이용할 수 있는 3D 모니터 상의 다중뷰 비디오 스트림의 디스플레이를 용이하고 편리하게 하기 위한 것이다.

[0039] 다중뷰 코딩을 위한 다른 방식은 다운샘플링 동작을 이용하여 더 큰 프레임 또는 슈퍼 프레임에서 (동시에 샘플링된) 각 뷰로부터 화상을 타일링하는 것이다. 도 1을 참조로 설명하면, 단일 프레임 상에 타일링된 4개의 뷰의 일례가 도면부호 100으로 총괄 표시되어 있다. 도 2를 참조로 설명하면, 단일 프레임 상에 플립핑 및 타일링된 4개의 뷰의 일례가 도면부호 200으로 총괄 표시되어 있다. 따라서 4개의 뷰가 있다면 각 뷰로부터의 화상은 타일과 같은 슈퍼 프레임으로 배치된다. 이에 따라서 더 큰 해상도를 가진 단일의 미코딩 입력 계열이 생성된다. 그러면 이 신호는 ISO/IEC MPEG-2 표준과 MPEG-4 AVC 표준과 같은 기존의 비디오 코딩 표준을 이용하여 인코딩될 수 있다.

[0040] 도 3을 참조로 설명하면, MPEG-4 AVC 표준에 따라서 비디오 인코딩을 수행할 수 있는 비디오 인코더는 도면부호 300으로 총괄적으로 나타낸다.

[0041] 비디오 인코더(300)는 결합기(385)의 비반전 입력부와 신호 통신하는 출력부를 가진 프레임 정렬 버퍼(310)를 포함한다. 결합기(385)의 출력부는 변환기 및 양자화기(325)의 제1 입력부와 신호 통신하도록 연결된다. 변환기 및 양자화기(325)의 출력부는 엔트로피 코더(345)의 제1 입력부 및 역변환기 및 역양자화기(350)의 제1 입력부와 신호 통신하도록 연결된다. 엔트로피 코더(345)의 출력부는 결합기(390)의 제1 비반전 입력부와 신호 통신하도록 연결된다. 결합기(390)의 출력부는 출력 버퍼(335)의 제1 입력부와 신호 통신하도록 연결된다.

[0042] 인코더 제어기(305)의 제1 출력부는 프레임 정렬 버퍼(310)의 제2 입력부, 역변환기 및 역양자화기(350)의 제2 입력부, 화상 타입 판정 모듈(315)의 입력부, 매크로블록(MB) 타입 판정 모듈(320)의 입력부, 인트라 예측 모듈(360)의 제2 입력부, 디블록킹 필터(365)의 제2 입력부, 동작 보상기(370)의 제1 입력부, 동작 추정기(375)의 제1 입력부 및 기준 화상 버퍼(380)의 제2 입력부와 신호 통신하도록 연결된다.

[0043] 인코더 제어기(305)의 제2 출력부는 SEI(Supplemental Enhancement Information) 삽입기(330)의 제1 입력부, 변환기 및 양자화기(325)의 제2 입력부, 엔트로피 코더(345)의 제2 입력부, 출력 버퍼(335)의 제2 입력부 및 SPS(Sequence Parameter Set) 및 PPS(Picture Parameter Set) 삽입기(340)의 입력부와 신호 통신하도록 연결된다.

[0044] 화상 타입 판정 모듈(315)의 제1 출력부는 프레임 정렬 버퍼(310)의 제3 입력부와 신호 통신하도록 연결된다. 화상 타입 판정 모듈(315)의 제2 출력부는 매크로블록 타입 판정 모듈(320)의 제2 입력부와 신호 통신하도록 연결된다.

[0045] SPS(Sequence Parameter Set) 및 PPS(Picture Parameter Set) 삽입기(340)의 출력부는 결합기(390)의 제3 비반전 입력부와 신호 통신하도록 연결된다. SEI 삽입기(330)의 출력부는 결합기(390)의 제2 비반전 입력부와 신호 통신하도록 연결된다.

[0046] 역변환기 및 역양자화기(350)의 출력부는 결합기(319)의 제1 비반전 입력부와 신호 통신하도록 연결된다. 결합기(319)의 출력부는 인트라 예측 모듈(360)의 제1 입력부 및 디블록킹 필터(365)의 제1 입력부와 신호 통신하도록 연결된다. 디블록킹 필터(365)의 출력부는 기준 화상 버퍼(380)의 제1 입력부와 신호 통신하도록 연결된다. 기준 화상 버퍼(380)의 출력부는 동작 추정기(375)의 제2 입력부 및 동작 보상기(370)의 제1 입력부와 신호 통

신하도록 연결된다. 동작 추정기(375)의 제1 출력부는 동작 보상기(370)의 제2 입력부와 신호 통신하도록 연결된다. 동작 추정기(375)의 제2 출력부는 엔트로피 코더(345)의 제3 입력부와 신호 통신하도록 연결된다.

[0047] 동작 보상기(370)의 출력부는 스위치(397)의 제1 입력부와 신호 통신하도록 연결된다. 인트라 예측 모듈(360)의 출력부는 스위치(397)의 제2 입력부와 신호 통신하도록 연결된다. 마크로블록 타입 판정 모듈(320)의 출력부는 스위치(397)에 제어 입력을 제공하기 위해 스위치(397)의 제3 입력부와 신호 통신하도록 연결된다. 스위치(397)의 제3 입력부는 (제어 입력, 즉 제3 입력과 비교해서) 스위치의 "데이터" 입력이 동작 보상기(370) 또는 인트라 예측 모듈(360)에 의해 제공될 것인지 여부를 판단한다. 스위치(397)의 출력부는 결합기(319)의 제2 비반전 입력부 및 결합기(385)의 반전 입력부와 신호 통신하도록 연결된다.

[0048] 프레임 정렬 버퍼(310) 및 인코더 제어기(305)의 입력부는 입력 화상(301)을 수신하는 인코더(300)의 입력부로서 이용될 수 있다. 더욱이 SEI(Supplemental Enhancement Information) 삽입기(330)의 입력부는 메타데이터를 수신하는 인코더(300)의 입력부로서 이용될 수 있다. 출력 버퍼(335)의 출력부는 비트스트림을 출력하는 인코더(300)의 출력부로서 이용될 수 있다.

[0049] 도 4를 참조로 설명하면, MPEG-4 AVC 표준에 따라 비디오 디코딩을 수행할 수 있는 예시적인 비디오 디코더는 도면부호 400으로 총괄적으로 나타낸다.

[0050] 비디오 디코더(400)는 엔트로피 디코더(445)의 제1 입력부와 신호 통신하도록 연결된 출력부를 가진 입력 버퍼(410)를 포함한다. 엔트로피 디코더(445)의 제1 출력부는 역변환기 및 역양자화기(450)의 제1 입력부와 신호 통신하도록 연결된다. 역변환기 및 역양자화기(450)의 출력부는 결합기(425)의 제2 비반전 입력부와 신호 통신하도록 연결된다. 결합기(425)의 출력부는 디블록킹 필터(465)의 제2 입력부 및 인트라 예측 모듈(460)의 제1 입력부와 신호 통신하도록 연결된다. 디블록킹 필터(465)의 제2 출력부는 기준 화상 버퍼(480)의 제1 입력부와 신호 통신하도록 연결된다. 기준 화상 버퍼(480)의 출력부는 동작 보상기(470)의 제2 입력부와 신호 통신하도록 연결된다.

[0051] 엔트로피 디코더(445)의 제2 출력부는 동작 보상기(470)의 제3 입력부 및 디블록킹 필터(465)의 제1 입력부와 신호 통신하도록 연결된다. 엔트로피 디코더(445)의 제3 출력부는 디코더 제어기(405)의 입력부와 신호 통신하도록 연결된다. 디코더 제어기(405)의 제1 출력부는 엔트로피 디코더(445)의 제2 입력부와 신호 통신하도록 연결된다. 디코더 제어기(405)의 제2 출력부는 역변환기 및 역양자화기(450)의 제2 입력부와 신호 통신하도록 연결된다. 디코더 제어기(405)의 제3 출력부는 디블록킹 제어기(465)의 제3 입력부와 신호 통신하도록 연결된다. 디코더 제어기(405)의 제4 출력부는 인트라 예측 모듈(460)의 제2 입력부, 동작 보상기(470)의 제1 입력부 및 기준 화상 버퍼(480)의 제2 입력부와 신호 통신하도록 연결된다.

[0052] 동작 보상기(470)의 출력부는 스위치(497)의 제1 입력부와 신호 통신하도록 연결된다. 인트라 예측 모듈(460)의 출력부는 스위치(497)의 제2 입력부와 신호 통신하도록 연결된다. 스위치(497)의 출력부는 결합기(425)의 제1 비반전 입력부와 신호 통신하도록 연결된다.

[0053] 입력 버퍼(410)의 입력부는 입력 비트스트림을 수신하는 디코더(400)의 입력부로서 이용될 수 있다. 디블록킹 필터(465)의 제1 출력부는 출력 화상을 출력하는 디코더(400)의 출력부로서 이용될 수 있다.

[0054] 도 5를 참조로 설명하면, MPEG-4 AVC 표준을 이용하여 복수의 뷰에 대한 화상을 인코딩하는 예시적인 방법은 도면부호 500으로 총괄 표시되어 있다.

[0055] 방법(500)은 기능 블록(504)으로 제어를 넘기는 시작 블록(502)을 포함한다. 기능 블록(504)은 특정 타임 인스턴스에서의 각 뷰를 타일 포맷의 부화상으로 배열하고, 제어를 기능 블록(506)으로 넘긴다. 기능 블록(506)은 구문 요소 num\_coded\_views\_minus1을 설정하고, 제어를 기능 블록(508)으로 넘긴다. 기능 블록(508)은 구문 요소 org\_pic\_width\_in\_mbs\_minus1과 org\_pic\_height\_in\_mbs\_minus1을 설정하고, 제어를 기능 블록(510)으로 넘긴다. 기능 블록(510)은 변수 i를 0으로 설정하고, 제어를 판단 블록(512)으로 넘긴다. 판단 블록(512)은 변수 i가 뷰 수보다 작은지를 판단한다. YES라면 제어는 기능 블록(514)으로 넘어간다. NO라면 제어는 기능 블록(524)으로 넘어간다.

[0056] 기능 블록(514)은 구문 요소 view\_id[i]를 설정하고, 제어를 기능 블록(516)으로 넘긴다. 기능 블록(516)은 구문 요소 num\_parts[view\_id[i]]를 설정하고, 제어를 기능 블록(518)으로 넘긴다. 기능 블록(518)은 변수 j를 0으로 설정하고, 제어를 판단 블록(520)으로 넘긴다. 판단 블록(520)은 변수 j의 현재 값이 구문 요소 num\_parts[view\_id[i]]의 현재 값보다 작은지를 판단한다. YES라면 제어는 기능 블록(522)으로 넘어간다. NO



라면 제어는 기능 블록(528)으로 넘어간다.

- [0057] 기능 블록(522)은 구문 요소 `depth_flag[view_id[i]][j]`; `flip_dir[view_id[i]][j]`; `loc_left_offset[view_id[i]][j]`; `loc_top_offset[view_id[i]][j]`; `frame_crop_left_offset[view_id[i]][j]`; `frame_crop_right_offset[view_id[i]][j]`; `frame_crop_top_offset[view_id[i]][j]`; 및 `frame_crop_bottom_offset[view_id[i]][j]`를 설정하고, 변수 `j`를 증분하고, 제어를 판단 블록(520)으로 넘긴다.
- [0058] 기능 블록(528)은 구문 요소 `upsample_view_flag[view_id[i]]`를 설정하고, 제어를 판단 블록(530)으로 넘긴다. 판단 블록(530)은 구문 요소 `upsample_view_flag[view_id[i]]`의 현재 값이 1인지를 판단한다. YES라면 제어는 기능 블록(532)으로 넘어간다. NO라면 제어는 판단 블록(534)으로 넘어간다.
- [0059] 기능 블록(532)은 구문 요소 `upsample_filter[view_id[i]]`를 설정하고, 제어를 판단 블록(534)으로 넘긴다.
- [0060] 판단 블록(534)은 구문 요소 `upsample_filter[view_id[i]]`의 현재 값이 3인지를 판단한다. YES라면 제어는 기능 블록(536)으로 넘어간다. NO라면 제어는 기능 블록(540)으로 넘어간다.
- [0061] 기능 블록(536)은 구문 요소 `vert_dim[view_id[i]]`; `hor_dim[view_id[i]]`; 및 `quantizer[view_id[i]]`를 설정하고, 제어를 기능 블록(538)으로 넘긴다.
- [0062] 기능 블록(538)은 각 YUV 성분에 대한 필터 계수를 설정하고, 제어를 기능 블록(540)으로 넘긴다.
- [0063] 기능 블록(540)은 변수 `i`를 증분하고, 제어를 판단 블록(512)으로 다시 넘긴다.
- [0064] 기능 블록(524)은 이들 구문 요소를 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), SEI(Supplemental Enhancement Information) 메시지, NAL(Network Abstraction Layer) 단위 헤더 및 슬라이스 헤더 중 적어도 하나에 기록하고, 제어를 기능 블록(526)으로 넘긴다. 기능 블록(526)은 각 화상을 MPEG-4 AVC 표준이나 기타 다른 단일 뷰 코덱을 이용하여 인코딩하고, 제어를 종료 블록(599)으로 넘긴다.
- [0065] 도 6을 참조로 설명하면, MPEG-4 AVC 표준을 이용하여 복수의 뷰에 대한 화상을 디코딩하는 예시적인 방법은 도면부호 600으로 총괄 표시되어 있다.
- [0066] 방법(600)은 기능 블록(604)으로 제어를 넘기는 시작 블록(602)을 포함한다. 기능 블록(604)은 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), SEI(Supplemental Enhancement Information) 메시지, NAL(Network Abstraction Layer) 단위 헤더 및 슬라이스 헤더 중 적어도 하나로부터 하기의 구문 요소들을 분석하고, 제어를 기능 블록(606)으로 넘긴다. 기능 블록(606)은 구문 요소 `num_coded_views_minus1`을 분석하고, 제어를 기능 블록(608)으로 넘긴다. 기능 블록(608)은 구문 요소 `org_pic_width_in_mbs_minus1`과 `org_pic_height_in_mbs_minus1`을 분석하고, 제어를 기능 블록(610)으로 넘긴다. 기능 블록(610)은 변수 `i`를 0으로 설정하고, 제어를 판단 블록(612)으로 넘긴다. 판단 블록(612)은 변수 `i`가 뷰 수보다 작은지를 판단한다. YES라면 제어는 기능 블록(614)으로 넘어간다. NO라면 제어는 기능 블록(624)으로 넘어간다.
- [0067] 기능 블록(614)은 구문 요소 `view_id[i]`를 분석하고, 제어를 기능 블록(616)으로 넘긴다. 기능 블록(616)은 구문 요소 `num_parts_minus1[view_id[i]]`를 분석하고, 제어를 기능 블록(618)으로 넘긴다. 기능 블록(618)은 변수 `j`를 0으로 설정하고, 제어를 판단 블록(620)으로 넘긴다. 판단 블록(620)은 변수 `j`의 현재 값이 구문 요소 `num_parts[view_id[i]]`의 현재 값보다 작은지를 판단한다. YES라면 제어는 기능 블록(622)으로 넘어간다. NO라면 제어는 기능 블록(628)으로 넘어간다.
- [0068] 기능 블록(622)은 구문 요소 `depth_flag[view_id[i]][j]`; `flip_dir[view_id[i]][j]`; `loc_left_offset[view_id[i]][j]`; `loc_top_offset[view_id[i]][j]`; `frame_crop_left_offset[view_id[i]][j]`; `frame_crop_right_offset[view_id[i]][j]`; `frame_crop_top_offset[view_id[i]][j]`; 및 `frame_crop_bottom_offset[view_id[i]][j]`를 분석하고, 변수 `j`를 증분하고, 제어를 판단 블록(620)으로 다시 넘긴다.
- [0069] 기능 블록(628)은 구문 요소 `upsample_view_flag[view_id[i]]`를 분석하고, 제어를 판단 블록(630)으로 넘긴다. 판단 블록(630)은 구문 요소 `upsample_view_flag[view_id[i]]`의 현재 값이 1인지를 판단한다. YES라면 제어는 기능 블록(632)으로 넘어간다. NO라면 제어는 판단 블록(634)으로 넘어간다.
- [0070] 기능 블록(632)은 구문 요소 `upsample_filter[view_id[i]]`를 분석하고, 제어를 판단 블록(634)으로 넘긴다.

- [0071] 판단 블록(634)은 구문 요소 upsample\_filter[view\_id[i]]의 현재 값이 3인지를 판단한다. YES라면 제어는 기능 블록(636)으로 넘어간다. NO라면 제어는 기능 블록(640)으로 넘어간다.
- [0072] 기능 블록(636)은 구문 요소 vert\_dim[view\_id[i]]; hor\_dim[view\_id[i]]; 및 quantizer[view\_id[i]]를 분석하고, 제어를 기능 블록(638)으로 넘긴다.
- [0073] 기능 블록(638)은 각 YUV 성분에 대한 필터 계수를 분석하고, 제어를 기능 블록(640)으로 넘긴다.
- [0074] 기능 블록(640)은 변수 i를 증분하고, 제어를 판단 블록(612)으로 다시 넘긴다.
- [0075] 기능 블록(624)은 각 화상을 MPEG-4 AVC 표준이나 기타 다른 단일 뷰 코덱을 이용하여 디코딩하고, 제어를 기능 블록(626)으로 넘긴다. 기능 블록(626)은 고레벨 구문을 이용하여 각 뷰를 화상으로부터 분리하고, 제어를 종료 블록(699)으로 넘긴다.
- [0076] 도 7을 참조로 설명하면, MPEG-4 AVC 표준을 이용하여 복수의 뷰와 깊이에 대한 화상을 인코딩하는 예시적인 방법은 도면부호 700으로 총괄 표시되어 있다.
- [0077] 방법(700)은 기능 블록(704)으로 제어를 넘기는 시작 블록(702)을 포함한다. 기능 블록(704)은 특정 타임 인스턴스에서의 각 뷰와 그 대응 깊이를 타일 포맷의 부화상으로 배열하고, 제어를 기능 블록(706)으로 넘긴다. 기능 블록(706)은 구문 요소 num\_coded\_views\_minus1을 설정하고, 제어를 기능 블록(708)으로 넘긴다. 기능 블록(708)은 구문 요소 org\_pic\_width\_in\_mbs\_minus1과 org\_pic\_height\_in\_mbs\_minus1을 설정하고, 제어를 기능 블록(710)으로 넘긴다. 기능 블록(710)은 변수 i를 0으로 설정하고, 제어를 판단 블록(712)으로 넘긴다. 판단 블록(712)은 변수 i가 뷰 수보다 작은지를 판단한다. YES라면 제어는 기능 블록(714)으로 넘어간다. NO라면 제어는 기능 블록(724)으로 넘어간다.
- [0078] 기능 블록(714)은 구문 요소 view\_id[i]를 설정하고, 제어를 기능 블록(716)으로 넘긴다. 기능 블록(716)은 구문 요소 num\_parts[view\_id[i]]를 설정하고, 제어를 기능 블록(718)으로 넘긴다. 기능 블록(718)은 변수 j를 0으로 설정하고, 제어를 판단 블록(720)으로 넘긴다. 판단 블록(720)은 변수 j의 현재 값이 구문 요소 num\_parts[view\_id[i]]의 현재 값보다 작은지를 판단한다. YES라면 제어는 기능 블록(722)으로 넘어간다. NO라면 제어는 기능 블록(728)으로 넘어간다.
- [0079] 기능 블록(722)은 구문 요소 depth\_flag[view\_id[i]][j]; flip\_dir[view\_id[i]][j]; loc\_left\_offset[view\_id[i]][j]; loc\_top\_offset[view\_id[i]][j]; frame\_crop\_left\_offset[view\_id[i]][j]; frame\_crop\_right\_offset[view\_id[i]][j]; frame\_crop\_top\_offset[view\_id[i]][j]; 및 frame\_crop\_bottom\_offset[view\_id[i]][j]를 설정하고, 변수 j를 증분하고, 제어를 판단 블록(720)으로 다시 넘긴다.
- [0080] 기능 블록(728)은 구문 요소 upsample\_view\_flag[view\_id[i]]를 설정하고, 제어를 판단 블록(730)으로 넘긴다. 판단 블록(730)은 구문 요소 upsample\_view\_flag[view\_id[i]]의 현재 값이 1인지를 판단한다. YES라면 제어는 기능 블록(732)으로 넘어간다. NO라면 제어는 판단 블록(734)으로 넘어간다.
- [0081] 기능 블록(732)은 구문 요소 upsample\_filter[view\_id[i]]를 설정하고, 제어를 판단 블록(734)으로 넘긴다.
- [0082] 판단 블록(734)은 구문 요소 upsample\_filter[view\_id[i]]의 현재 값이 3인지를 판단한다. YES라면 제어는 기능 블록(736)으로 넘어간다. NO라면 제어는 기능 블록(740)으로 넘어간다.
- [0083] 기능 블록(736)은 구문 요소 vert\_dim[view\_id[i]]; hor\_dim[view\_id[i]]; 및 quantizer[view\_id[i]]를 설정하고, 제어를 기능 블록(738)으로 넘긴다.
- [0084] 기능 블록(738)은 각 YUV 성분에 대한 필터 계수를 설정하고, 제어를 기능 블록(740)으로 넘긴다.
- [0085] 기능 블록(540)은 변수 i를 증분하고, 제어를 판단 블록(712)으로 다시 넘긴다.
- [0086] 기능 블록(724)은 이들 구문 요소를 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), SEI(Supplemental Enhancement Information) 메시지, NAL(Network Abstraction Layer) 단위 헤더 및 슬라이스 헤더 중 적어도 하나에 기록하고, 제어를 기능 블록(726)으로 넘긴다. 기능 블록(726)은 각 화상을 MPEG-4 AVC 표준이나 기타 다른 단일 뷰 코덱을 이용하여 인코딩하고, 제어를 종료 블록(799)으로 넘긴다.
- [0087] 도 8을 참조로 설명하면, MPEG-4 AVC 표준을 이용하여 복수의 뷰와 깊이에 대한 화상을 디코딩하는 예시적인 방법은 도면부호 800으로 총괄 표시되어 있다.

- [0088] 방법(800)은 기능 블록(804)으로 제어를 넘기는 시작 블록(802)을 포함한다. 기능 블록(804)은 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), SEI(Supplemental Enhancement Information) 메시지, NAL(Network Abstraction Layer) 단위 헤더 및 슬라이스 헤더 중 적어도 하나로부터 하기의 구문 요소들을 분석하고, 제어를 기능 블록(806)으로 넘긴다. 기능 블록(806)은 구문 요소 num\_coded\_views\_minus1을 분석하고, 제어를 기능 블록(808)으로 넘긴다. 기능 블록(808)은 구문 요소 org\_pic\_width\_in\_mbs\_minus1과 org\_pic\_height\_in\_mbs\_minus1을 분석하고, 제어를 기능 블록(810)으로 넘긴다. 기능 블록(810)은 변수 i를 0으로 설정하고, 제어를 판단 블록(812)으로 넘긴다. 판단 블록(812)은 변수 i가 뷰 수보다 작은지를 판단한다. YES라면 제어는 기능 블록(814)으로 넘어간다. NO라면 제어는 기능 블록(824)으로 넘어간다.
- [0089] 기능 블록(814)은 구문 요소 view\_id[i]를 분석하고, 제어를 기능 블록(816)으로 넘긴다. 기능 블록(816)은 구문 요소 num\_parts\_minus1[view\_id[i]]를 분석하고, 제어를 기능 블록(818)으로 넘긴다. 기능 블록(818)은 변수 j를 0으로 설정하고, 제어를 판단 블록(820)으로 넘긴다. 판단 블록(820)은 변수 j의 현재 값이 구문 요소 num\_parts[view\_id[i]]의 현재 값보다 작은지를 판단한다. YES라면 제어는 기능 블록(822)으로 넘어간다. NO라면 제어는 기능 블록(828)으로 넘어간다.
- [0090] 기능 블록(822)은 구문 요소 depth\_flag[view\_id[i]][j]; flip\_dir[view\_id[i]][j]; loc\_left\_offset[view\_id[i]][j]; loc\_top\_offset[view\_id[i]][j]; frame\_crop\_left\_offset[view\_id[i]][j]; frame\_crop\_right\_offset[view\_id[i]][j]; frame\_crop\_top\_offset[view\_id[i]][j]; 및 frame\_crop\_bottom\_offset[view\_id[i]][j]를 분석하고, 변수 j를 증분하고, 제어를 판단 블록(820)으로 다시 넘긴다.
- [0091] 기능 블록(828)은 구문 요소 upsample\_view\_flag[view\_id[i]]를 분석하고, 제어를 판단 블록(830)으로 넘긴다. 판단 블록(830)은 구문 요소 upsample\_view\_flag[view\_id[i]]의 현재 값이 1인지를 판단한다. YES라면 제어는 기능 블록(832)으로 넘어간다. NO라면 제어는 판단 블록(834)으로 넘어간다.
- [0092] 기능 블록(832)은 구문 요소 upsample\_filter[view\_id[i]]를 분석하고, 제어를 판단 블록(834)으로 넘긴다.
- [0093] 판단 블록(834)은 구문 요소 upsample\_filter[view\_id[i]]의 현재 값이 3인지를 판단한다. YES라면 제어는 기능 블록(836)으로 넘어간다. NO라면 제어는 기능 블록(840)으로 넘어간다.
- [0094] 기능 블록(836)은 구문 요소 vert\_dim[view\_id[i]]; hor\_dim[view\_id[i]]; 및 quantizer[view\_id[i]]를 분석하고, 제어를 기능 블록(838)으로 넘긴다.
- [0095] 기능 블록(838)은 각 YUV 성분에 대한 필터 계수를 분석하고, 제어를 기능 블록(840)으로 넘긴다.
- [0096] 기능 블록(840)은 변수 i를 증분하고, 제어를 판단 블록(812)으로 다시 넘긴다.
- [0097] 기능 블록(824)은 각 화상을 MPEG-4 AVC 표준이나 기타 다른 단일 뷰 코덱을 이용하여 디코딩하고, 제어를 기능 블록(826)으로 넘긴다. 기능 블록(826)은 고레벨 구문을 이용하여 각 뷰를 화상으로부터 분리하고, 제어를 기능 블록(827)으로 넘긴다. 기능 블록(827)은 추출된 뷰 및 깊이 신호를 이용하여 뷰 합성을 잠재적으로 수행하고, 제어를 종료 블록(899)으로 넘긴다.
- [0098] 도 7과 8에서 이용된 깊이에 대해서 도 9는 깊이 신호(900)의 예를 보여준다. 이 예에서, 깊이는 이미지(미도시)의 각 대응 위치에 대한 픽셀값으로 제공된다. 더욱이, 도 10은 타일(1000)에 포함된 2개의 깊이 신호의 예를 보여준다. 타일(1000)의 상단 우측 부분은 타일(1000)의 상단 좌측 부분의 이미지에 대응하는 깊이 값을 가진 깊이 신호이다. 타일(1000)의 하단 우측 부분은 타일(1000)의 하단 좌측 부분의 이미지에 대응하는 깊이 값을 가진 깊이 신호이다.
- [0099] 도 11을 참조로 설명하면, 단일 프레임에 타일링된 5개의 뷰의 예가 도면부호 1100으로 총괄 표시되어 있다. 상단의 4개 뷰는 정상적으로 배향되어 있다. 다섯 번째 뷰도 정상적으로 배향되어 있지만 타일(1100)의 하단을 따라 2 부분으로 분할되어 있다. 다섯 번째 뷰의 좌측 부분은 다섯 번째 뷰의 "상단"을 보여주고, 다섯 번째 뷰의 우측 부분은 다섯 번째 뷰의 "하단"을 보여준다.
- [0100] 도 12를 참조로 설명하면, 예시적인 MVC(Multi-view Video Coding) 인코더가 도면부호 1200으로 총괄 표시되어 있다. 인코더(1200)는 변환기(1210)의 입력부와 신호 통신하도록 연결된 출력부를 가진 결합기(1205)를 포함한다. 변환기(1210)의 출력부는 양자화기(1215)의 출력부와 신호 통신하도록 연결된다. 양자화기(1215)의 출력부는 엔트로피 코더(1220)의 입력부 및 역양자화기(1225)의 입력부와 신호 통신하도록 연결된다. 역양자화기(1225)의 출력부는 역변환기(1230)의 입력부와 신호 통신하도록 연결된다. 역변환기(1230)의 출력부는 결합기

(1235)의 제1 비반전 입력부와 신호 통신하도록 연결된다. 결합기(1235)의 출력부는 인트라예측기(1245)의 입력부 및 디블록킹 필터(1250)의 입력부와 신호 통신하도록 연결된다. 디블록킹 필터(1245)의 출력부는 (뷰 i에 대한) 기준 화상 저장부(1255)의 입력과 신호 통신하도록 연결된다. 기준 화상 저장부(1255)의 출력부는 동작 보상기(1275)의 제1 입력부 및 동작 추정기(1280)의 제1 입력부와 신호 통신하도록 연결된다. 동작 추정기(1280)의 출력부는 동작 보상기(1275)의 제2 입력부와 신호 통신하도록 연결된다.

[0101] (다른 뷰에 대한) 기준 화상 저장부(1260)의 출력부는 디스패리티 추정기(1270)의 제1 입력부 및 디스패리티 보상기(1265)의 제1 입력부와 신호 통신하도록 연결된다. 디스패리티 추정기(1270)의 출력부는 디스패리티 보상기(1265)의 제2 입력부와 신호 통신하도록 연결된다.

[0102] 엔트로피 디코더(1220)의 출력부는 인코더(1200)의 출력부로서 이용될 수 있다. 결합기(1205)의 비반전 입력부는 인코더(1200)의 입력부로서 이용될 수 있고, 디스패리티 추정기(1270)의 제2 입력부 및 동작 추정기(1280)의 제2 입력부와 신호 통신하도록 연결된다. 스위치(1285)의 출력부는 결합기(1235)의 제2 비반전 입력부 및 결합기(1205)의 반전 입력부와 신호 통신하도록 연결된다. 스위치(1285)는 동작 보상기(1275)의 출력부와 신호 통신하도록 연결된 제1 입력부, 디스패리티 보상기(1265)의 출력부와 신호 통신하도록 연결된 제2 입력부, 및 인트라예측기(1245)의 출력부와 신호 통신하도록 연결된 제3 입력부를 포함한다.

[0103] 모드 판정 모듈(1240)은 스위치(1285)에 의해 어느 입력이 선택되는지를 제어하기 위해 스위치(1285)에 연결된 출력부를 갖고 있다.

[0104] 도 13을 참조로 설명하면, 예시적인 MVC(Multi-view Video Coding) 디코더가 도면부호 1300으로 총괄 표시되어 있다. 디코더(1300)는 역양자화기(1310)의 입력부와 신호 통신하도록 연결된 출력부를 가진 엔트로피 디코더(1305)를 포함한다. 역양자화기의 출력부는 역변환기(1315)의 입력부와 신호 통신하도록 연결된다. 역변환기(1315)의 출력부는 결합기(1320)의 제1 비반전 입력부와 신호 통신하도록 연결된다. 결합기(1320)의 출력부는 디블록킹 필터(1325)의 입력부 및 인트라예측기(1330)의 입력부와 신호 통신하도록 연결된다. 디블록킹 필터(1325)의 출력부는 (뷰 i에 대한) 기준 화상 저장부(1340)의 입력부와 신호 통신하도록 연결된다. 기준 화상 저장부(1340)의 출력부는 동작 보상기(1335)의 제1 입력부와 신호 통신하도록 연결된다.

[0105] (다른 뷰에 대한) 기준 화상 저장부(1345)의 출력부는 디스패리티 보상기(1350)의 제1 입력부와 신호 통신하도록 연결된다.

[0106] 엔트로피 코더(1305)의 입력부는 잔류 비트스트림을 수신하기 위해 디코더(1300)의 입력부로서 이용될 수 있다. 더욱이, 모드 모듈(1360)의 입력부도 스위치(1335)에 의해 어느 입력이 선택되는지를 제어하는 제어 구문을 수신하기 위해 디코더(1300)의 입력부로서 이용될 수 있다. 더욱이, 동작 보상기(1335)의 제2 입력부는 동작 벡터를 수신하기 위해 디코더(1300)의 입력부로서 이용될 수 있다. 또한, 디스패리티 보상기(1350)의 제2 입력부는 디스패리티 벡터를 수신하기 위해 디코더(1300)의 입력부로서 이용될 수 있다.

[0107] 스위치(1355)의 출력부는 결합기(1320)의 제2 비반전 입력부와 신호 통신하도록 연결된다. 스위치(1355)의 제1 입력부는 디스패리티 보상기(1350)의 출력부와 신호 통신하도록 연결된다. 스위치(1355)의 제2 입력부는 동작 보상기(1335)의 출력부와 신호 통신하도록 연결된다. 스위치(1355)의 제3 입력부는 인트라예측기(1330)의 출력부와 신호 통신하도록 연결된다. 모드 모듈(1360)의 출력부는 스위치(1335)에 의해 어느 입력이 선택되는지를 제어하기 위해 스위치(1355)와 신호 통신하도록 연결된다. 디블록킹 필터(1325)의 출력부는 디코더(1300)의 출력부로서 이용될 수 있다.

[0108] 도 14를 참조로 설명하면, MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 화상을 인코딩하는 것에 대한 준비로서 복수의 뷰에 대한 화상을 처리하는 예시적인 방법은 도면부호 1400으로 총괄 표시되어 있다.

[0109] 방법(1400)은 기능 블록(1410)으로 제어를 넘기는 시작 블록(1405)을 포함한다. 기능 블록(1410)은 특정 타임 인스턴스에서 총 M개의 뷰 중 N개 뷰마다 타일 포맷의 수퍼 화상으로 배열하고, 제어를 기능 블록(1415)으로 넘긴다. 기능 블록(1415)은 구문 요소 num\_coded\_views\_minus1을 설정하고, 제어를 기능 블록(1420)으로 넘긴다. 기능 블록(1420)은 모든 (num\_coded\_views\_minus1 + 1) 뷰에 대해 구문 요소 view\_id[i]를 설정하고, 제어를 기능 블록(1425)으로 넘긴다. 기능 블록(1425)은 앵커 화상에 대한 뷰간 기준 종속성 정보를 설정하고, 제어를 기능 블록(1430)으로 넘긴다. 기능 블록(1430)은 비앵커 화상에 대한 뷰간 기준 종속성 정보를 설정하고, 제어를 기능 블록(1435)으로 넘긴다. 기능 블록(1435)은 구문 요소 pseudo\_view\_present\_flag를 설정하고, 제어를 판단 블록(1440)으로 넘긴다. 판단 블록(1440)은 구문 요소 pseudo\_view\_present\_flag의 현



재 값이 참인지를 판단한다. YES이면 제어는 기능 블록(1445)으로 넘어간다. NO이면 제어는 종료 블록(1499)으로 넘어간다.

[0110] 기능 블록(1445)은 구문 요소 `tiling_mode`; `org_pic_width_in_mbs_minus1`; 및 `org_pic_height_in_mbs_minus1`를 설정하고, 제어를 기능 블록(1450)으로 넘긴다. 기능 블록(1450)은 각 코딩된 뷰에 대한 구문 요소 `pseudo_view_info(view_id)`를 호출하고 제어를 종료 블록(1499)으로 넘긴다.

[0111] 도 15를 참조로 설명하면, MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 복수의 뷰에 대한 화상을 인코딩하는 예시적인 방법은 도면부호 1500으로 총괄 표시되어 있다.

[0112] 방법(1500)은 입력 파라미터 `pseudo_view_id`를 가지며 기능 블록(1504)으로 제어를 넘기는 시작 블록(1502)을 포함한다. 기능 블록(1504)은 구문 요소 `num_sub_views_minus1`를 설정하고, 제어를 기능 블록(1506)으로 넘긴다. 기능 블록(1506)은 변수 `i`를 0으로 설정하고, 제어를 판단 블록(1508)으로 넘긴다. 판단 블록(1508)은 변수 `i`가 `sub_views`의 수보다 작은지를 판단한다. YES이면 제어는 기능 블록(1510)으로 넘어간다. NO이면 제어는 기능 블록(1520)으로 넘어간다.

[0113] 기능 블록(1510)은 구문 요소 `sub_view_id[i]`를 설정하고, 제어를 기능 블록(1512)으로 넘긴다. 기능 블록(1512)은 구문 요소 `num_parts_minus1[sub_view_id[i]]`를 설정하고, 제어를 기능 블록(1514)으로 넘긴다. 기능 블록(1514)은 변수 `j`를 0으로 설정하고, 제어를 판단 블록(1516)으로 넘긴다. 판단 블록(1516)은 변수 `j`가 구문 요소 `num_parts_minus1[sub_view_id[i]]`보다 작은지를 판단한다. YES이면 제어는 기능 블록(1518)으로 넘어간다. NO이면 제어는 판단 블록(1522)으로 넘어간다.

[0114] 기능 블록(1518)은 구문 요소 `loc_left_offset[sub_view_id[i]][j]`; `loc_top_offset[sub_view_id[i]][j]`; `frame_crop_left_offset[sub_view_id[i]][j]`; `frame_crop_right_offset[sub_view_id[i]][j]`; `frame_crop_top_offset[sub_view_id[i]][j]`; 및 `frame_crop_bottom_offset[sub_view_id[i]][j]`를 설정하고, 변수 `j`를 증분하고, 제어를 판단 블록(1516)으로 다시 넘긴다.

[0115] 기능 블록(1520)은 MVC(Multi-view Video Coding)을 이용하여 현재 뷰에 대한 현재 화상을 인코딩하고, 제어를 종료 블록(1599)으로 넘긴다.

[0116] 판단 블록(1522)은 구문 요소 `tiling_mode`가 0인지를 판단한다. YES이면 제어는 기능 블록(1524)으로 넘어간다. NO이면 제어는 기능 블록(1538)으로 넘어간다.

[0117] 기능 블록(1524)은 구문 요소 `flip_dir[sub_view_id[i]]`와 구문 요소 `upsample_view_flag[sub_view_id[i]]`를 설정하고, 제어를 판단 블록(1526)으로 넘긴다. 판단 블록(1526)은 구문 요소 `upsample_view_flag[sub_view_id[i]]`의 현재 값이 1인지를 판단한다. YES라면 제어는 기능 블록(1528)으로 넘어간다. NO라면 제어는 판단 블록(1530)으로 넘어간다.

[0118] 기능 블록(1528)은 구문 요소 `upsample_filter[sub_view_id[i]]`를 설정하고, 제어를 판단 블록(1530)으로 넘긴다. 판단 블록(1530)은 구문 요소 `upsample_filter[sub_view_id[i]]`가 3인지를 판단한다. YES이면 제어는 기능 블록(1532)으로 넘어간다. NO이면 제어는 기능 블록(1536)으로 넘어간다.

[0119] 기능 블록(1532)은 구문 요소 `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; 및 `quantizer[sub_view_id[i]]`를 설정하고, 제어를 기능 블록(1534)으로 넘긴다. 기능 블록(1534)은 각 YUV 성분 에 대한 필터 계수를 설정하고, 제어를 기능 블록(1536)으로 넘긴다.

[0120] 기능 블록(1536)은 변수 `i`를 증분하고, 제어를 판단 블록(1508)으로 다시 넘긴다.

[0121] 기능 블록(1538)은 구문 요소 `pixel_dist_x[sub_view_id[i]]`와 구문 요소 `flip_dist_y[sub_view_id[i]]`를 설정하고, 제어를 기능 블록(1540)으로 넘긴다. 기능 블록(1540)은 변수 `j`를 0으로 설정하고, 제어를 판단 블록(1542)으로 넘긴다. 판단 블록(1542)은 변수 `j`의 현재 값이 구문 요소 `num_parts[sub_view_id[i]]`의 현재 값보다 작은지를 판단한다. YES이면 제어는 기능 블록(1544)으로 넘어간다. NO이면 제어는 기능 블록(1536)으로 넘어간다.

[0122] 기능 블록(1544)은 구문 요소 `num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]`를 설정하고, 제어를 기능 블록(1546)으로 넘긴다. 기능 블록(1546)은 모든 픽셀 타일링 필터에 대한 계수를 설정하고, 제어를 기능 블록(1536)으로 넘긴다.

[0123] 도 16을 참조로 설명하면, MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 화상을 디코딩하

는 것에 대한 준비로서 복수의 뷰에 대한 화상을 처리하는 예시적인 방법은 도면부호 1600으로 총괄 표시되어 있다.

- [0124] 방법(1600)은 기능 블록(1615)으로 제어를 넘기는 시작 블록(1605)을 포함한다. 기능 블록(1615)은 구문 요소 `num_coded_views_minus1`를 분석하고, 제어를 기능 블록(1620)으로 넘긴다. 기능 블록(1620)은 모든  $(\text{num\_coded\_views\_minus1} + 1)$  뷰에 대해 구문 요소 `view_id[i]`를 분석하고, 제어를 기능 블록(1625)으로 넘긴다. 기능 블록(1625)은 앵커 화상에 대한 뷰간 기준 종속성 정보를 분석하고, 제어를 기능 블록(1630)으로 넘긴다. 기능 블록(1630)은 비앵커 화상에 대한 뷰간 기준 종속성 정보를 분석하고, 제어를 기능 블록(1635)으로 넘긴다. 기능 블록(1635)은 구문 요소 `pseudo_view_present_flag`를 분석하고, 제어를 판단 블록(1640)으로 넘긴다. 판단 블록(1640)은 구문 요소 `pseudo_view_present_flag`의 현재 값이 참인지를 판단한다. YES이면 제어는 기능 블록(1645)으로 넘어간다. NO이면 제어는 종료 블록(1699)으로 넘어간다.
- [0125] 기능 블록(1645)은 구문 요소 `tiling_mode`; `org_pic_width_in_mbs_minus1`; 및 `org_pic_height_in_mbs_minus1`를 분석하고, 제어를 기능 블록(1650)으로 넘긴다. 기능 블록(1650)은 각 코딩된 뷰에 대한 구문 요소 `pseudo_view_info(view_id)`를 호출하고 제어를 종료 블록(1699)으로 넘긴다.
- [0126] 도 17을 참조로 설명하면, MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 복수의 뷰에 대한 화상을 디코딩하는 예시적인 방법은 도면부호 1700으로 총괄 표시되어 있다.
- [0127] 방법(1700)은 입력 파라미터 `pseudo_view_id`를 가지고 시작하며 기능 블록(1504)으로 제어를 넘기는 시작 블록(1702)을 포함한다. 기능 블록(1704)은 구문 요소 `num_sub_views_minus1`를 분석하고, 제어를 기능 블록(1706)으로 넘긴다. 기능 블록(1706)은 변수 `i`를 0으로 설정하고, 제어를 판단 블록(1708)으로 넘긴다. 판단 블록(1708)은 변수 `i`가 `sub_views`의 수보다 작은지를 판단한다. YES이면 제어는 기능 블록(1710)으로 넘어간다. NO이면 제어는 기능 블록(1720)으로 넘어간다.
- [0128] 기능 블록(1710)은 구문 요소 `sub_view_id[i]`를 분석하고, 제어를 기능 블록(1712)으로 넘긴다. 기능 블록(1712)은 구문 요소 `num_parts_minus1[sub_view_id[i]]`를 분석하고, 제어를 기능 블록(1714)으로 넘긴다. 기능 블록(1714)은 변수 `j`를 0으로 설정하고, 제어를 판단 블록(1716)으로 넘긴다. 판단 블록(1716)은 변수 `j`가 구문 요소 `num_parts_minus1[sub_view_id[i]]`보다 작은지를 판단한다. YES이면 제어는 기능 블록(1718)으로 넘어간다. NO이면 제어는 판단 블록(1722)으로 넘어간다.
- [0129] 기능 블록(1718)은 구문 요소 `loc_left_offset[sub_view_id[i]][j]`; `loc_top_offset[sub_view_id[i]][j]`; `frame_crop_left_offset[sub_view_id[i]][j]`; `frame_crop_right_offset[sub_view_id[i]][j]`; `frame_crop_top_offset[sub_view_id[i]][j]`; 및 `frame_crop_bottom_offset[sub_view_id[i]][j]`를 설정하고, 변수 `j`를 증분하고, 제어를 판단 블록(1716)으로 다시 넘긴다.
- [0130] 기능 블록(1720)은 MVC(Multi-view Video Coding)을 이용하여 현재 뷰에 대한 현재 화상을 디코딩하고, 제어를 기능 블록(1721)으로 넘긴다. 기능 블록(1721)은 고레벨 구문을 이용하여 각 뷰를 화상으로부터 분리하고, 제어를 종료 블록(1799)으로 넘긴다.
- [0131] 디코딩된 화상으로부터의 각 뷰의 분리는 비트스트림에 나타난 고레벨 구문을 이용하여 행해진다. 이 고레벨 구문은 화상에 있는 뷰(및 그 대응 깊이)의 정확한 위치와 배향(orientation)를 나타낸다.
- [0132] 판단 블록(1722)은 구문 요소 `tiling_mode`가 0인지를 판단한다. YES이면 제어는 기능 블록(1724)으로 넘어간다. NO이면 제어는 기능 블록(1738)으로 넘어간다.
- [0133] 기능 블록(1724)은 구문 요소 `flip_dir[sub_view_id[i]]`와 구문 요소 `upsample_view_flag[sub_view_id[i]]`를 분석하고, 제어를 판단 블록(1726)으로 넘긴다. 판단 블록(1726)은 구문 요소 `upsample_view_flag[sub_view_id[i]]`의 현재 값이 1인지를 판단한다. YES라면 제어는 기능 블록(1728)으로 넘어간다. NO라면 제어는 판단 블록(1730)으로 넘어간다.
- [0134] 기능 블록(1728)은 구문 요소 `upsample_filter[sub_view_id[i]]`를 분석하고, 제어를 판단 블록(1730)으로 넘긴다. 판단 블록(1730)은 구문 요소 `upsample_filter[sub_view_id[i]]`가 3인지를 판단한다. YES이면 제어는 기능 블록(1732)으로 넘어간다. NO이면 제어는 기능 블록(1736)으로 넘어간다.
- [0135] 기능 블록(1732)은 구문 요소 `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; 및 `quantizer[sub_view_id[i]]`를 분석하고, 제어를 기능 블록(1734)으로 넘긴다. 기능 블록(1734)은 각 YUV 성분

에 대한 필터 계수를 분석하고, 제어를 기능 블록(1736)으로 넘긴다.

- [0136] 기능 블록(1736)은 변수  $i$ 를 증분하고, 제어를 판단 블록(1708)으로 다시 넘긴다.
- [0137] 기능 블록(1738)은 구문 요소 `pixel_dist_x[sub_view_id[i]]`와 구문 요소 `flip_dist_y[sub_view_id[i]]`를 분석하고, 제어를 기능 블록(1740)으로 넘긴다. 기능 블록(1740)은 변수  $j$ 를 0으로 설정하고, 제어를 판단 블록(1742)으로 넘긴다. 판단 블록(1742)은 변수  $j$ 의 현재 값이 구문 요소 `num_parts[sub_view_id[i]]`의 현재 값보다 작은지를 판단한다. YES이면 제어는 기능 블록(1744)으로 넘어간다. NO이면 제어는 기능 블록(1736)으로 넘어간다.
- [0138] 기능 블록(1744)은 구문 요소 `num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]`를 분석하고, 제어를 기능 블록(1746)으로 넘긴다. 기능 블록(1746)은 모든 픽셀 타일링 필터에 대한 계수를 분석하고, 제어를 기능 블록(1736)으로 넘긴다.
- [0139] 도 18을 참조로 설명하면, MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 화상을 인코딩하는 것에 대한 준비로서 복수의 뷰와 깊이에 대한 화상을 처리하는 예시적인 방법은 도면부호 1800으로 총괄 표시되어 있다.
- [0140] 방법(1800)은 기능 블록(1810)으로 제어를 넘기는 시작 블록(1805)을 포함한다. 기능 블록(1810)은 특정 타임 인스턴스에서 총  $M$ 개의 뷰 및 깊이 맵 중  $N$ 개 뷰 및 깊이 맵마다 타일 포맷의 수퍼 화상으로 배열하고, 제어를 기능 블록(1815)으로 넘긴다. 기능 블록(1815)은 구문 요소 `num_coded_views_minus1`를 설정하고, 제어를 기능 블록(1820)으로 넘긴다. 기능 블록(1820)은 모든 (`num_coded_views_minus1 + 1`) 뷰에 대해 구문 요소 `view_id[i]`를 설정하고, 제어를 기능 블록(1825)으로 넘긴다. 기능 블록(1825)은 앵커 깊이 화상에 대한 뷰간 기준 종속성 정보를 설정하고, 제어를 기능 블록(1830)으로 넘긴다. 기능 블록(1830)은 비앵커 깊이 화상에 대한 뷰간 기준 종속성 정보를 설정하고, 제어를 기능 블록(1835)으로 넘긴다. 기능 블록(1835)은 구문 요소 `pseudo_view_present_flag`를 설정하고, 제어를 판단 블록(1840)으로 넘긴다. 판단 블록(1840)은 구문 요소 `pseudo_view_present_flag`의 현재 값이 참인지를 판단한다. YES이면 제어는 기능 블록(1845)으로 넘어간다. NO이면 제어는 종료 블록(1899)으로 넘어간다.
- [0141] 기능 블록(1845)은 구문 요소 `tiling_mode`; `org_pic_width_in_mbs_minus1`; 및 `org_pic_height_in_mbs_minus1`를 설정하고, 제어를 기능 블록(1850)으로 넘긴다. 기능 블록(1850)은 각 코딩된 뷰에 대한 구문 요소 `pseudo_view_info(view_id)`를 호출하고 제어를 종료 블록(1899)으로 넘긴다.
- [0142] 도 19를 참조로 설명하면, MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 복수의 뷰 및 깊이에 대한 화상을 인코딩하는 예시적인 방법은 도면부호 1900으로 총괄 표시되어 있다.
- [0143] 방법(1900)은 기능 블록(1904)으로 제어를 넘기는 시작 블록(1902)을 포함한다. 기능 블록(1904)은 구문 요소 `num_sub_views_minus1`를 설정하고, 제어를 기능 블록(1906)으로 넘긴다. 기능 블록(1906)은 변수  $i$ 를 0으로 설정하고, 제어를 판단 블록(1908)으로 넘긴다. 판단 블록(1908)은 변수  $i$ 가 `sub_views`의 수보다 작은지를 판단한다. YES이면 제어는 기능 블록(1910)으로 넘어간다. NO이면 제어는 기능 블록(1920)으로 넘어간다.
- [0144] 기능 블록(1910)은 구문 요소 `sub_view_id[i]`를 설정하고, 제어를 기능 블록(1912)으로 넘긴다. 기능 블록(1912)은 구문 요소 `num_parts_minus1[sub_view_id[i]]`를 설정하고, 제어를 기능 블록(1914)으로 넘긴다. 기능 블록(1914)은 변수  $j$ 를 0으로 설정하고, 제어를 판단 블록(1916)으로 넘긴다. 판단 블록(1916)은 변수  $j$ 가 구문 요소 `num_parts_minus1[sub_view_id[i]]`보다 작은지를 판단한다. YES이면 제어는 기능 블록(1918)으로 넘어간다. NO이면 제어는 판단 블록(1922)으로 넘어간다.
- [0145] 기능 블록(1918)은 구문 요소 `loc_left_offset[sub_view_id[i]][j]`; `loc_top_offset[sub_view_id[i]][j]`; `frame_crop_left_offset[sub_view_id[i]][j]`; `frame_crop_right_offset[sub_view_id[i]][j]`; `frame_crop_top_offset[sub_view_id[i]][j]`; 및 `frame_crop_bottom_offset[sub_view_id[i]][j]`를 설정하고, 변수  $j$ 를 증분하고, 제어를 판단 블록(1916)으로 다시 넘긴다.
- [0146] 기능 블록(1920)은 MVC(Multi-view Video Coding)을 이용하여 현재 뷰에 대한 현재 깊이를 인코딩하고, 제어를 종료 블록(1999)으로 넘긴다. 깊이 신호는 그 대응 비디오 신호가 인코딩되는 방식과 유사하게 인코딩될 수 있다. 예컨대, 어떤 뷰에 대한 깊이 신호는 다른 깊이 신호만을, 또는 비디오 신호만을, 또는 깊이와 비디오 신호 모두를 포함하는 타일에 포함될 수 있다. 그러면 이 타일(의사 뷰)은 MVC를 위한 단일 뷰로서 취급되며, MVC를 위한 다른 뷰로서 취급되는 다른 타일들도 있을 수 있다.

- [0147] 판단 블록(1922)은 구문 요소 `tiling_mode`가 0인지를 판단한다. YES이면 제어는 기능 블록(1924)으로 넘어간다. NO이면 제어는 기능 블록(1938)으로 넘어간다.
- [0148] 기능 블록(1924)은 구문 요소 `flip_dir[sub_view_id[i]]`와 구문 요소 `upsample_view_flag[sub_view_id[i]]`를 설정하고, 제어를 판단 블록(1926)으로 넘긴다. 판단 블록(1926)은 구문 요소 `upsample_view_flag[sub_view_id[i]]`의 현재 값이 1인지를 판단한다. YES라면 제어는 기능 블록(1928)으로 넘어간다. NO라면 제어는 판단 블록(1930)으로 넘어간다.
- [0149] 기능 블록(1928)은 구문 요소 `upsample_filter[sub_view_id[i]]`를 설정하고, 제어를 판단 블록(1930)으로 넘긴다. 판단 블록(1930)은 구문 요소 `upsample_filter[sub_view_id[i]]`가 3인지를 판단한다. YES이면 제어는 기능 블록(1932)으로 넘어간다. NO이면 제어는 기능 블록(1936)으로 넘어간다.
- [0150] 기능 블록(1932)은 구문 요소 `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; 및 `quantizer[sub_view_id[i]]`를 설정하고, 제어를 기능 블록(1934)으로 넘긴다. 기능 블록(1934)은 각 YUV 성분 에 대한 필터 계수를 설정하고, 제어를 기능 블록(1936)으로 넘긴다.
- [0151] 기능 블록(1936)은 변수 `i`를 증분하고, 제어를 판단 블록(1908)으로 다시 넘긴다.
- [0152] 기능 블록(1938)은 구문 요소 `pixel_dist_x[sub_view_id[i]]`와 구문 요소 `flip_dist_y[sub_view_id[i]]`를 설정하고, 제어를 기능 블록(1940)으로 넘긴다. 기능 블록(1940)은 변수 `j`를 0으로 설정하고, 제어를 판단 블록(1942)으로 넘긴다. 판단 블록(1942)은 변수 `j`의 현재 값이 구문 요소 `num_parts[sub_view_id[i]]`의 현재 값보다 작은지를 판단한다. YES이면 제어는 기능 블록(1944)으로 넘어간다. NO이면 제어는 기능 블록(1936)으로 넘어간다.
- [0153] 기능 블록(1944)은 구문 요소 `num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]`를 설정하고, 제어를 기능 블록(1946)으로 넘긴다. 기능 블록(1946)은 모든 픽셀 타일링 필터에 대한 계수를 설정하고, 제어를 기능 블록(1936)으로 넘긴다.
- [0154] 도 20을 참조로 설명하면, MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 화상을 디코딩하는 것에 대한 준비로서 복수의 뷰 및 깊이에 대한 화상을 처리하는 예시적인 방법은 도면부호 2000으로 총괄 표시되어 있다.
- [0155] 방법(2000)은 기능 블록(2015)으로 제어를 넘기는 시작 블록(2005)을 포함한다. 기능 블록(2015)은 구문 요소 `num_coded_views_minus1`를 분석하고, 제어를 기능 블록(2020)으로 넘긴다. 기능 블록(2020)은 모든 (`num_coded_views_minus1 + 1`) 뷰에 대해 구문 요소 `view_id[i]`를 분석하고, 제어를 기능 블록(2025)으로 넘긴다. 기능 블록(2025)은 앵커 깊이 화상에 대한 뷰간 기준 종속성 정보를 분석하고, 제어를 기능 블록(2030)으로 넘긴다. 기능 블록(2030)은 비앵커 깊이 화상에 대한 뷰간 기준 종속성 정보를 분석하고, 제어를 기능 블록(2035)으로 넘긴다. 기능 블록(2035)은 구문 요소 `pseudo_view_present_flag`를 분석하고, 제어를 판단 블록(2040)으로 넘긴다. 판단 블록(2040)은 구문 요소 `pseudo_view_present_flag`의 현재 값이 참인지를 판단한다. YES이면 제어는 기능 블록(2045)으로 넘어간다. NO이면 제어는 종료 블록(2099)으로 넘어간다.
- [0156] 기능 블록(2045)은 구문 요소 `tiling_mode`; `org_pic_width_in_mbs_minus1`; 및 `org_pic_height_in_mbs_minus1`를 분석하고, 제어를 기능 블록(2050)으로 넘긴다. 기능 블록(2050)은 각 코딩된 뷰에 대한 구문 요소 `pseudo_view_info(view_id)`를 호출하고 제어를 종료 블록(2099)으로 넘긴다.
- [0157] 도 21을 참조로 설명하면, MPEG-4 AVC 표준의 MVC(Multi-view Video Coding) 확장을 이용하여 복수의 뷰 및 깊이에 대한 화상을 디코딩하는 예시적인 방법은 도면부호 2100으로 총괄 표시되어 있다.
- [0158] 방법(2100)은 입력 파라미터 `pseudo_view_id`를 가지고 시작하며 기능 블록(2104)으로 제어를 넘기는 시작 블록(2102)을 포함한다. 기능 블록(2104)은 구문 요소 `num_sub_views_minus1`를 분석하고, 제어를 기능 블록(2106)으로 넘긴다. 기능 블록(2106)은 변수 `i`를 0으로 설정하고, 제어를 판단 블록(2108)으로 넘긴다. 판단 블록(2108)은 변수 `i`가 `sub_views`의 수보다 작은지를 판단한다. YES이면 제어는 기능 블록(2110)으로 넘어간다. NO이면 제어는 기능 블록(2120)으로 넘어간다.
- [0159] 기능 블록(2110)은 구문 요소 `sub_view_id[i]`를 분석하고, 제어를 기능 블록(2112)으로 넘긴다. 기능 블록(2112)은 구문 요소 `num_parts_minus1[sub_view_id[i]]`를 분석하고, 제어를 기능 블록(2114)으로 넘긴다. 기능 블록(2114)은 변수 `j`를 0으로 설정하고, 제어를 판단 블록(2116)으로 넘긴다. 판단 블록(2116)은 변수 `j`가 구문 요소 `num_parts_minus1[sub_view_id[i]]`보다 작은지를 판단한다. YES이면 제어는 기능 블록(2118)으로



넘어간다. NO이면 제어는 판단 블록(2122)으로 넘어간다.

- [0160] 기능 블록(2118)은 구문 요소 `loc_left_offset[sub_view_id[i]][j]`; `loc_top_offset[sub_view_id[i]][j]`; `frame_crop_left_offset[sub_view_id[i]][j]`; `frame_crop_right_offset[sub_view_id[i]][j]`; `frame_crop_top_offset[sub_view_id[i]][j]`; 및 `frame_crop_bottom_offset[sub_view_id[i]][j]`를 설정하고, 변수 `j`를 증분하고, 제어를 판단 블록(2116)으로 다시 넘긴다.
- [0161] 기능 블록(2120)은 MVC(Multi-view Video Coding)을 이용하여 현재 화상을 디코딩하고, 제어를 기능 블록(2121)으로 넘긴다. 기능 블록(2121)은 고레벨 구문을 이용하여 각 뷰를 화상으로부터 분리하고, 제어를 종료 블록(2199)으로 넘긴다. 고레벨 구문을 이용한 각 뷰의 분리는 전술한 바와 같다.
- [0162] 판단 블록(2122)은 구문 요소 `tiling_mode`가 0인지를 판단한다. YES이면 제어는 기능 블록(2124)으로 넘어간다. NO이면 제어는 기능 블록(2138)으로 넘어간다.
- [0163] 기능 블록(2124)은 구문 요소 `flip_dir[sub_view_id[i]]`와 구문 요소 `upsample_view_flag[sub_view_id[i]]`를 분석하고, 제어를 판단 블록(2126)으로 넘긴다. 판단 블록(2126)은 구문 요소 `upsample_view_flag[sub_view_id[i]]`의 현재 값이 1인지를 판단한다. YES라면 제어는 기능 블록(2128)으로 넘어간다. NO라면 제어는 판단 블록(2130)으로 넘어간다.
- [0164] 기능 블록(2128)은 구문 요소 `upsample_filter[sub_view_id[i]]`를 분석하고, 제어를 판단 블록(2130)으로 넘긴다. 판단 블록(2130)은 구문 요소 `upsample_filter[sub_view_id[i]]`가 3인지를 판단한다. YES이면 제어는 기능 블록(2132)으로 넘어간다. NO이면 제어는 기능 블록(2136)으로 넘어간다.
- [0165] 기능 블록(2132)은 구문 요소 `vert_dim[sub_view_id[i]]`; `hor_dim[sub_view_id[i]]`; 및 `quantizer[sub_view_id[i]]`를 분석하고, 제어를 기능 블록(2134)으로 넘긴다. 기능 블록(2134)은 각 YUV 성분에 대한 필터 계수를 분석하고, 제어를 기능 블록(2136)으로 넘긴다.
- [0166] 기능 블록(2136)은 변수 `i`를 증분하고, 제어를 판단 블록(2108)으로 다시 넘긴다.
- [0167] 기능 블록(2138)은 구문 요소 `pixel_dist_x[sub_view_id[i]]`와 구문 요소 `flip_dist_y[sub_view_id[i]]`를 분석하고, 제어를 기능 블록(2140)으로 넘긴다. 기능 블록(2140)은 변수 `j`를 0으로 설정하고, 제어를 판단 블록(2142)으로 넘긴다. 판단 블록(2142)은 변수 `j`의 현재 값이 구문 요소 `num_parts[sub_view_id[i]]`의 현재 값보다 작은지를 판단한다. YES이면 제어는 기능 블록(2144)으로 넘어간다. NO이면 제어는 기능 블록(2136)으로 넘어간다.
- [0168] 기능 블록(2144)은 구문 요소 `num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]]`를 분석하고, 제어를 기능 블록(2146)으로 넘긴다. 기능 블록(2146)은 모든 픽셀 타일링 필터에 대한 계수를 분석하고, 제어를 기능 블록(2136)으로 넘긴다.
- [0169] 도 22를 참조로 설명하면, 픽셀 레벨에서의 타일링 예는 도면부호 2200으로 총괄 표시되어 있다. 도 22는 뒤에 더 자세히 설명한다.
- [0170] 다중뷰 비디오 코딩의 일 응용은 자유시점 TV(FTV)이다. 이 응용은 사용자가 2개 이상의 뷰 간에 자유로이 이동할 수 있는 것을 요한다. 이르 달성하기 위해 2개의 뷰 간에 "가상" 뷰가 보간 또는 합성되어야 한다. 뷰 보간을 실행하는 방법은 몇 가지 있다. 이들 방법 중 하나는 뷰 보간/합성에 깊이를 이용한다.
- [0171] 각 뷰는 관련 깊이 신호를 가질 수 있다. 따라서 깊이는 다른 형태의 비디오 신호로 생각할 수 있다. 도 9는 깊이 신호(900)의 일례를 보여준다. FTV와 같은 응용을 가능하게 하기 위해 깊이 신호는 비디오 신호와 함께 전송된다. 제시된 타일링 프레임워크에서는 깊이 신호는 타일들 중 하나로서 부가될 수도 있다. 도 10은 타일로서 부가된 깊이 신호의 일례를 보여준다. 이 깊이 신호/타일은 도 10의 우측에 도시되어 있다.
- [0172] 깊이가 전체 프레임의 타일로서 인코딩되면 고레벨 구문은 렌더러(renderer)가 깊이 신호를 적당히 이용할 수 있도록 어느 타일이 깊이 신호인지를 나타내어야 한다.
- [0173] (도 1에 도시된 것과 같은) 입력 계열이 MPEG-4 AVC 표준 인코더(또는 다른 비디오 코딩 표준 및/또는 권고안에 대응하는 인코더)를 이용하여 인코딩되는 경우에는 제시된 고레벨 구문은 예컨대 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), 슬라이스 헤더 및/또는 SEI(Supplemental Enhancement Information) 메시지에 존재할 수 있다. 제시된 방법의 실시예는 구문이 SEI(Supplemental Enhancement Information) 메시지에 존재하는 표 1에 나타낸다.

[0174] (도 1에 도시된 것과 같은) 의사 뷰의 입력 계열이 MPEG-4 AVC 표준 인코더의 MVC(Multi-view Video Coding) 확장(또는 다른 비디오 코딩 표준 및/또는 권고안에 대한 다중뷰 비디오 코딩 표준에 대응하는 인코더)를 이용하여 인코딩되는 경우에는 제시된 고레벨 구문은 SPS, PPS, 슬라이스 헤더, SEI 메시지, 또는 특정 프로파일에 존재할 수 있다. 제시된 방법의 실시예는 표 1에 나타낸다. 표 1은 본 발명의 원리에 따라 실현된 제시된 구문 요소를 포함하여 SPS(Sequence Parameter Set) 구조에 존재하는 구문 요소를 보여준다.

표 1

seq_parameter_set_mvc_extension( ) {	C	Descriptor
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for( j = 0; j < num_anchor_refs_l0[i]; j++ )		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for( j = 0; j < num_anchor_refs_l1[i]; j++ )		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
for( j = 0; j < num_non_anchor_refs_l0[i]; j++ )		
non_anchor_ref_l0[i][j]		ue(v)

[0175]

num_non_anchor_refs_l1[i]		ue(v)
for( j = 0; j < num_non_anchor_refs_l1[i]; j++ )		
non_anchor_ref_l1[i][j]		ue(v)
}		
pseudo_view_present_flag		u(1)
if (pseudo_view_present_flag) {		
tiling_mode		
org_pic_width_in_mbs_minus1		
org_pic_height_in_mbs_minus1		
for( i = 0; i < num_views_minus_1; i++)		
pseudo_view_info(i);		
}		
}		

[0176]

[0177] 표 2는 표 1의 pseudo\_view\_info 구문 요소에 대한 구문 요소를 보여준다.

표 2

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

[0178]

}		
}		
}		
} // if(tiling_mode == 0)		
else if (tiling_mode == 1) {		
pixel_dist_x[sub_view_id[ i ]]		
pixel_dist_y[sub_view_id[ i ]]		
for( j = 0; j <= num_parts[sub_view_id[ i ]]; j++ ) {		
num_pixel_tiling_filter_coeffs_minus1[sub_view_id[ i ]][j]		
for (coeff_idx = 0; coeff_idx <=		
num_pixel_tiling_filter_coeffs_minus1[sub_view_id[ i ]][j]; j++)		
pixel_tiling_filter_coeffs[sub_view_id[i]][j]		
} // for ( j = 0; j <= num_parts[sub_view_id[ i ]]; j++ )		
} // else if (tiling_mode == 1)		
} // for ( i = 0; i < num_sub_views_minus_1; i++)		
} // if (num_sub_views_minus_1 != 0)		
}		

[0179]

[0180] 표 1과 표 2에 나타낸 구문 요소의 의미는 하기와 같다.

[0181] pseudo\_view\_present\_flag는 참이라는 것은 어떤 뷰가 다중 서브뷰의 수퍼 뷰임을 의미한다.

[0182] tiling\_mode가 0이라는 것은 서브뷰가 화상 레벨에서 타일링되는 것을 의미한다. 값 1은 타일링이 픽셀 레벨에서 행해지는 것을 의미한다.

[0183] 새로운 SEI 메시지는 MPEG-4 AVC 표준 또는 MPEG-4 AVC 표준의 확장에서 이용되지 않았던 SEI 페이로드 타입에

대한 값을 이용할 수 있다. 새로운 SEI 메시지는 다음의 의미를 가진 몇 개의 구문 요소를 포함한다.

- [0184] **num-coded\_views\_minus1** 플러스 1은 비트스트림이 지원하는 코딩된 뷰의 수를 의미한다. **num-coded\_views\_minus1**의 값은 0부터 1023의 범위이다.
- [0185] **org\_pic\_width\_in\_mbs\_minus1** 플러스 1은 매크로블록 단위들에서 각 뷰의 화상의 폭을 의미한다.
- [0186] 매크로블록 단위들에서 화상 폭의 변수는 다음과 같이 도출된다.
- [0187]  $\text{PicWidthInMbs} = \text{org\_pic\_width\_in\_mbs\_minus1} + 1$
- [0188] 루마(luma) 성분에 대한 화상 폭의 변수는 다음과 같이 도출된다.
- [0189]  $\text{PicWidthInSamplesL} = \text{PicWidthInMbs} * 16$
- [0190] 크로마(chroma) 성분에 대한 화상 폭의 변수는 다음과 같이 도출된다.
- [0191]  $\text{PicWidthInSamplesC} = \text{PicWidthInMbs} * \text{MbWidthC}$
- [0192] **org\_pic\_height\_in\_mbs\_minus1** 플러스 1은 매크로블록 단위들에서 각 뷰의 화상의 높이를 의미한다.
- [0193] 매크로블록 단위들에서 화상 높이의 변수는 다음과 같이 도출된다.
- [0194]  $\text{PicHeightInMbs} = \text{org\_pic\_height\_in\_mbs\_minus1} + 1$
- [0195] 루마 성분에 대한 화상 높이의 변수는 다음과 같이 도출된다.
- [0196]  $\text{PicHeightInSamplesL} = \text{PicHeightInMbs} * 16$
- [0197] 크로마 성분에 대한 화상 높이의 변수는 다음과 같이 도출된다.
- [0198]  $\text{PicHeightInSamplesC} = \text{PicHeightInMbs} * \text{MbHeightC}$
- [0199] **num-sub\_views\_minus1** 플러스 1은 현재 뷰에 포함된 코딩된 서브뷰의 수를 의미한다. **num-sub\_views\_minus1**의 값은 0부터 1023의 범위이다.
- [0200] **sub\_view\_id[i]**는 i로 표시된 디코딩 순서로 서브뷰의 **sub\_view\_id**를 의미한다.
- [0201] **num\_parts[sub\_view\_id[i]]**는 **sub\_view\_id[i]**의 화상의 분할 수를 의미한다.
- [0202] **loc\_left\_offset[sub\_view\_id[i]][j]** 및 **loc\_top\_offset[sub\_view\_id[i]][j]**는 각각 좌측 및 상단 픽셀 오프셋에서의 위치를 의미한다. 여기서, 현재 부분 j는 **sub\_view\_id**가 **sub\_view\_id[i]**와 같은 상태에서 뷰의 최종 구성된 화상 내에 위치해 있다.
- [0203] **view\_id[i]**는 i로 표시된 코딩 순서를 가진 뷰의 **view\_id**를 의미한다.
- [0204] **frame\_crop\_left\_offset[view\_id[i]][j]**, **frame\_crop\_right\_offset[view\_id[i]][j]**, **frame\_crop\_top\_offset[view\_id[i]][j]**, 및 **frame\_crop\_bottom\_offset[view\_id[i]][j]**는 출력을 위한 프레임 좌표에서 특정된 직사각형 영역에 관하여 **num\_part j**와 **view\_id i**의 부분인 코딩된 비디오 계열 중의 화상의 샘플을 의미한다.
- [0205] 변수 **CropUnitX**와 **CropUnitY**는 다음과 같이 도출된다.
- [0206] - 만일 **chroma\_format\_idc**가 0이라면 **CropUnitX**와 **CropUnitY**는 다음과 같이 도출된다.
- [0207]  $\text{CropUnitX} = 1$
- [0208]  $\text{CropUnitY} = 2 - \text{frame\_mbs\_only\_flag}$
- [0209] - 그렇지 않으면(즉, **chroma\_format\_idc**가 1, 2, 또는 3이라면) **CropUnitX**와 **CropUnitY**는 다음과 같이 도출된다.
- [0210]  $\text{CropUnitX} = \text{SubWidthC}$
- [0211]  $\text{CropUnitY} = \text{SubHeightC} * (2 - \text{frame\_mbs\_only\_flag})$
- [0212] 프레임 크롭핑 직사각형은  $\text{CropUnitX} * \text{frame\_crop\_left\_offset}$ 부터  $\text{PicWidthSamplesL} - (\text{CropUnitX} * \text{frame\_crop\_right\_offset} + 1)$ 까지의 수평 프레임 좌표와  $\text{CropUnitY} * \text{frame\_crop\_top\_offset}$ 부터  $(16 * \text{frame\_crop\_bottom\_offset} + 1)$ 까지의 수직 프레임 좌표를 의미한다.

FrameHeightInMbs)-(CropUnitY \* frame\_crop\_bottom\_offset+1)까지의 수직 프레임 좌표를 가진 루마 샘플을 포함한다. frame\_crop\_left\_offset의 값은 0에서 (PicWidthSamplesL/CropUnitX)-(frame\_crop\_right\_offset+1)까지의 범위에 있고, frame\_crop\_top\_offset의 값은 0에서 (16 \* FrameHeightInMbs/CropUnitY)-(frame\_crop\_bottom\_offset+1)까지의 범위에 있을 것이다.

- [0213] chroma\_format\_idc가 0이 아닌 경우에는 2개의 크로마 배열의 대응 특정 샘플은 프레임 좌표(x/SubWidthC, y/SubHeightC)를 가진 샘플이다. 여기서 (x,y)는 특정 루마 샘플의 프레임 좌표이다.
- [0214] 디코딩된 필드에 대해서는, 디코딩된 필드의 특정 샘플은 프레임 좌표에서 특정된 직사각형 내에 있는 샘플이다.
- [0215] num\_parts[view\_id[i]]는 view\_id[i]의 화상의 분할 수를 의미한다.
- [0216] depth\_flag[view\_id[i]]는 현재 부분이 깊이 신호인지를 규정한다. depth\_flag가 0이면 현재 부분은 깊이 신호가 아니다. depth\_flag가 1이면 현재 부분은 view\_id[i]로 식별된 뷰와 연관된 깊이 신호이다.
- [0217] flip\_dir[sub\_view\_id[i][j]]는 현재 부분에 대한 플립핑 방향을 의미한다. flip\_dir이 0이면 플립핑이 없다는 것을 의미하고, flip\_dir이 1이면 수평 방향에서의 플립핑을 의미하고, flip\_dir이 2이면 수직 방향에서의 플립핑을 의미하고, flip\_dir이 3이면 수평과 수직 방향에서의 플립핑을 의미한다.
- [0218] flip\_dir[view\_id[i][j]]는 현재 부분에 대한 플립핑 방향을 의미한다. flip\_dir이 0이면 플립핑이 없다는 것을 의미하고, flip\_dir이 1이면 수평 방향에서의 플립핑을 의미하고, flip\_dir이 2이면 수직 방향에서의 플립핑을 의미하고, flip\_dir이 3이면 수평과 수직 방향에서의 플립핑을 의미한다.
- [0219] loc\_left\_offset[view\_id[i]][j] 및 loc\_top\_offset[view\_id[i]][j]는 픽셀 오프셋에서의 위치를 의미한다. 여기서, 현재 부분 j는 view\_id가 view\_id[i]와 같은 상태에서 뷰의 최종 구성된 화상 내에 위치해 있다.
- [0220] upsample\_view\_flag[view\_id[i]]는 view\_id[i]로 특정된 뷰에 속하는 화상이 업샘플링될 필요가 있는지를 나타낸다. upsample\_view\_flag[view\_id[i]]가 0이면 view\_id가 view\_id[i]와 같은 화상이 업샘플링되지 않을 것이라는 것을 의미한다. upsample\_view\_flag[view\_id[i]]가 1이면 view\_id가 view\_id[i]와 같은 화상이 업샘플링될 것이라는 것을 의미한다.
- [0221] upsample\_filter[view\_id[i]]는 업샘플링에 이용될 필터의 종류를 나타낸다. upsample\_filter[view\_id[i]]가 0이면 6-탭 AVC 필터가 이용되어야 하는 것을 의미하고, upsample\_filter[view\_id[i]]가 1이면 4-탭 SVC 필터가 이용되어야 하는 것을 의미하고, upsample\_filter[view\_id[i]]가 2이면 바이리니어(bilinear) 필터가 이용되어야 하는 것을 의미하고, upsample\_filter[view\_id[i]]가 3이면 커스텀 필터 계수가 전송되는 것을 의미한다. upsample\_filter[view\_id[i]]가 존재하지 않으면 이는 0으로 설정된다. 이 실시예에서는 2D 커스텀 마이즈된 필터를 이용한다. 이는 ID 필터와 기타 다른 비선형 필터에도 쉽게 확장될 수 있다.
- [0222] ver\_dim[view\_id[i]]는 커스텀 2D 필터의 수직 치수를 의미한다.
- [0223] hor\_dim[view\_id[i]]는 커스텀 2D 필터의 수평 치수를 의미한다.
- [0224] quantizer[view\_id[i]]는 각 필터 계수에 대한 양자화 계수를 의미한다.
- [0225] filter\_coefs[view\_id[i]][yuv][y][x]는 양자화된 필터 계수를 의미한다. yuv는 필터 계수가 적용되는 성분을 나타낸다. yuv가 0이면 Y 성분을 나타내고, yuv가 1이면 U 성분을 나타내고, yuv가 2이면 V 성분을 나타낸다.
- [0226] pixel\_dist\_x[sub\_view\_id[i]]와 pixel\_dist\_y[sub\_view\_id[i]]는 각각 sub\_view\_id가 sub\_view\_id[i]와 같은 뷰 중의 인접 픽셀들 간의 최종 재구성된 의사 뷰의 수평 방향과 수직 방향에서의 거리를 의미한다.
- [0227] num\_pixel\_tiling\_filter\_coefs\_minus1[sub\_view\_id[i][j]]는 타일링된 화상을 필터링하는데 이용될 수 있는 필터를 나타내는데 필요한 필터 계수를 의미한다.
- [0228] 도 22를 참조로 설명하면, 4개의 뷰로부터 픽셀들을 타일링함으로써 의사 뷰로 구성되는 것을 보여주는 2개의 예는 각각 도면부호 2210과 2220으로 표시된다. 4개의 뷰는 집합적으로 도면부호 2250으로 표시된다. 도 22에서 첫번째 예에 대한 구문 값은 하기 표 3에 나타낸다.

표 3

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

[0229]

[0230] 도 22에서 두 번째 예에 대한 구문 값들은 2개의 구문 요소, loc\_left\_offset[3][0]이 5이고 loc\_top\_offset[3][0]이 3이라는 것만 제외하고는 모두 같다.

[0231] 이 오프셋은 뷰에 대응하는 픽셀들이 특정 오프셋 위치에서 시작해야 하나를 것을 나타낸다. 이는 도 22에서 (2220)으로 나타나 있다. 이것은 예컨대 2개의 뷰가 공통의 오브젝트들이 뷰 간에 이동해보이는 이미지를 생성한다. 예컨대, (제1 및 제2 뷰를 나타내는) 제1 및 제2 카메라가 어떤 오브젝트의 화상을 찍는다면 그 오브젝트는 제1 뷰와 비교해 제2 뷰에서 5개의 픽셀이 우측으로 이동해 보일 수 있다. 이는 제1 뷰의 픽셀(i-5, j)이 제2 뷰의 픽셀(i, j)에 대응한다는 것을 의미한다. 이 2개 뷰의 픽셀이 단순히 픽셀 단위로 타일링되면 그 타일 내의 인접 픽셀들 간에는 상관이 별로 없고 공간 코딩 이득이 작을 수 있다. 반대로, 뷰 1로부터의 픽셀(i-5, j)이 뷰 2로부터의 픽셀(i, j) 다음에 위치하도록 타일링을 이동시키면 공간 상관이 증가하고 공간 코딩 이득도 증가할 수 있다. 이는 예컨대 제1 및 제2 뷰에서의 오브젝트에 대한 대응 픽셀이 서로 나란히 타일링되고 있기 때문이다.

[0232] 따라서, loc\_left\_offset과 loc\_top\_offset가 존재하면 코딩 효율이 개선될 수 있다. 이 오프셋 정보는 외부적 수단에 의해 얻을 수 있다. 예컨대, 카메라의 위치 정보 또는 뷰들 간의 전역적 디스패리티 벡터는 그와 같은 오프셋 정보를 결정하는데 이용될 수 있다.

[0233] 오프셋팅의 결과로서, 의사 뷰의 일부 픽셀은 뷰로부터 픽셀 값이 할당되지 않는다. 상기 예를 가지고 계속 설명하면, 뷰 1로부터의 픽셀(i-5, j)을 뷰 2로부터의 픽셀(i, j)(i는 0, ..., 4)과 동시에 타일링하면, 뷰 1로부터의 픽셀(i-5, j)은 타일링되지 않고, 따라서 이들 픽셀은 타일에서 비어있다. 뷰로부터 픽셀 값이 할당되지 않은 의사 뷰(타일) 중의 이들 픽셀에 대해서는 적어도 하나의 구현은 AVC에서 동작 보상 시의 서브픽셀 보간 절차와 유사한 보간 절차를 이용한다. 즉, 비어있는 타일 픽셀은 인접 픽셀들로부터 보간될 수 있다. 그와 같은 보간에 의해서 타일의 공간 상관이 더 커지고 타일의 코딩 이득도 더 커질 수 있다.

[0234] 비디오 코딩에서 I, P 및 B 화상과 같은 각 화상에 대해 서로 다른 코딩 타입을 선택할 수 있다. 다중뷰 비디오 코딩에 대해서는 앵커 및 비앵커 화상을 정의한다. 일 실시예에서, 화상 타입에 따라서 그룹핑 판정이 이루어질 수 있는 것을 제시한다. 이 그룹핑 정보는 고레벨 구문에 표시된다.

[0235] 도 11을 참조로 설명하면, 단일 프레임에 타일링된 5개의 뷰의 예가 도면부호 1100으로 총괄 표시되어 있다. 특히, 단일 프레임에 5개의 뷰가 타일링된 벌룸(ballroom) 시퀀스가 나타나 있다. 그 외에도, 다섯 번째 뷰가 직사각형 프레임 상에 배열될 수 있도록 2 부분으로 분할되어 있음을 볼 수 있다. 여기서 각 뷰는 총 프레임 크기가 640×600이 되도록 QVGA 크기를 갖고 있다. 600은 16의 배수가 아니므로 608로 확장되어야 한다.

[0236] 이 예에서는 가능한 SEI 메시지는 표 4에 나타낼 수 있다.

표 4

multiview_display_info( payloadSize ) {	Value
num_coded_views_minus1	5
org_pic_width_in_mbs_minus1	40
org_pic_height_in_mbs_minus1	30
view_id[ 0 ]	0
num_parts[view_id[ 0 ]]	1

[0237]



depth_flag[view_id[ 0 ]][ 0 ]	0
flip_dir[view_id[ 0 ]][ 0 ]	0
loc_left_offset[view_id[ 0 ]][ 0 ]	0
loc_top_offset[view_id[ 0 ]][ 0 ]	0
frame_crop_left_offset[view_id[ 0 ]][ 0 ]	0
frame_crop_right_offset[view_id[ 0 ]][ 0 ]	320
frame_crop_top_offset[view_id[ 0 ]][ 0 ]	0
frame_crop_bottom_offset[view_id[ 0 ]][ 0 ]	240
upsample_view_flag[view_id[ 0 ]]	1
if(upsample_view_flag[view_id[ 0 ]]) {	
vert_dim[view_id[0]]	6
hor_dim[view_id[0]]	6
quantizer[view_id[0]]	32
for (yuv= 0; yuv< 3; yuv++) {	
for (y = 0; y < vert_dim[view_id[i]] - 1; y++) {	
for (x = 0; x < hor_dim[view_id[i]] - 1; x++)	
filter_coeffs[view_id[i]] [yuv][y][x]	XX
view_id[ 1 ]	1
num_parts[view_id[ 1 ]]	1
depth_flag[view_id[ 0 ]][ 0 ]	0
flip_dir[view_id[ 1 ]][ 0 ]	0
loc_left_offset[view_id[ 1 ]][ 0 ]	0
loc_top_offset[view_id[ 1 ]][ 0 ]	0
frame_crop_left_offset[view_id[ 1 ]][ 0 ]	320
frame_crop_right_offset[view_id[ 1 ]][ 0 ]	640
frame_crop_top_offset[view_id[ 1 ]][ 0 ]	0
frame_crop_bottom_offset[view_id[ 1 ]][ 0 ]	320
upsample_view_flag[view_id[ 1 ]]	1
if(upsample_view_flag[view_id[ 1 ]]) {	
vert_dim[view_id[1]]	6
hor_dim[view_id[1]]	6
quantizer[view_id[1]]	32
for (yuv= 0; yuv< 3; yuv++) {	
for (y = 0; y < vert_dim[view_id[i]] - 1; y++) {	
for (x = 0; x < hor_dim[view_id[i]] - 1; x++)	
filter_coeffs[view_id[i]] [yuv][y][x]	XX
.....(similarly for view 2,3)	

[0238]



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

[0239]

[0240]

표 5는 표 4에 나타난 예에 대한 다중뷰 정보를 전송하기 위한 일반적인 구문 구조를 보여준다.

### 표 5

multiview_display_info( payloadSize ) {	C	Descriptor
num_coded_views_minus1	5	ue(v)
org_pic_width_in_mbs_minus1	5	ue(v)
org_pic_height_in_mbs_minus1	5	ue(v)
for( i = 0; i <= num_coded_views_minus1; i++ ) {		
view_id[ i ]	5	ue(v)
num_parts[view_id[ i ]]	5	ue(v)
for( j = 0; j <= num_parts[i]; j++ ) {		

[0241]

depth_flag[view_id[i]][j]		
flip_dir[view_id[i]][j]	5	u(2)
loc_left_offset[view_id[i]][j]	5	ue(v)
loc_top_offset[view_id[i]][j]	5	ue(v)
frame_crop_left_offset[view_id[i]][j]	5	ue(v)
frame_crop_right_offset[view_id[i]][j]	5	ue(v)
frame_crop_top_offset[view_id[i]][j]	5	ue(v)
frame_crop_bottom_offset[view_id[i]][j]	5	ue(v)
}		
upsample_view_flag[view_id[i]]	5	u(1)
if(upsample_view_flag[view_id[i]])		
upsample_filter[view_id[i]]	5	u(2)
if(upsample_fiter[view_id[i]] == 3) {		
vert_dim[view_id[i]]	5	ue(v)
hor_dim[view_id[i]]	5	ue(v)
quantizer[view_id[i]]	5	ue(v)
for (yuv= 0; yuv< 3; yuv++) {		
for (y = 0; y < vert_dim[view_id[i]] - 1; y++) {		
for (x = 0; x < hor_dim[view_id[i]] - 1; x++)		
filter_coefs[view_id[i]][yuv][y][x]	5	se(v)
}		
}		
}		
}		
}		

[0242]

[0243]

도 23을 참조로 설명하면, 비디오 처리 장치(2300)가 도시되어 있다. 비디오 처리 장치(2300)는 예컨대 인코딩된 비디오를 수신하고, 예컨대 디코딩된 비디오를 사용자에게 표시하거나 저장하기 위해 제공하는 셋톱 박스나 기타 다른 장치일 수 있다. 따라서 장치(2300)는 그 출력을 텔레비전, 컴퓨터 모니터, 컴퓨터 또는 기타 다른 처리 장치에 제공할 수 있다.

[0244]

장치(2300)는 데이터 신호(2320)를 수신하는 디코더(2310)를 포함한다. 데이터 신호(2320)는 예컨대 AVC 또는 MVC 호환 스트림을 포함할 수 있다. 디코더(2310)는 수신된 신호(2320)의 일부 또는 전부를 디코딩하여, 디코딩된 비디오 신호(2330) 및 타일링 정보(2340)를 출력으로서 제공한다. 디코딩된 비디오 신호(2330) 및 타일링 정보(2340)는 선택기(2350)에 제공된다. 장치(2300)는 사용자 입력(2370)을 수신하는 사용자 인터페이스(2360)도 포함한다. 사용자 인터페이스(2360)는 사용자 입력(2370)에 따라서 화상 선택 신호(2380)를 선택기(2350)에 제공한다. 화상 선택 신호(2380)와 사용자 입력(2370)은 사용자가 복수의 화상 중 어느 것을 디스플레이 하기를 바라는지를 나타낸다. 선택기(2350)는 선택된 화상(들)을 출력(2390)으로서 제공한다. 선택기(2350)는 화상 선택 정보(2380)를 이용하여 출력(2390)으로서 제공될 디코딩된 비디오(2330)의 화상 중 어느 것을 선택한다. 선택기(2350)는 타일링 정보(2340)를 이용하여 디코딩된 비디오(2330)의 선택된 화상(들)의 위치를 찾아낸다.

[0245]

여러 가지 구현에서, 선택기(2350)는 사용자 인터페이스(2360)를 포함하며, 다른 구현에서는, 선택기(2350)가 별도의 인터페이싱 기능을 수행하지 않고 바로 사용자 입력(2370)을 수신하기 때문에 사용자 인터페이스(2360)가 필요없다. 선택기(2350)는 예컨대 소프트웨어 또는 집적 회로에서 구현될 수 있다. 선택기(2350)는 디코더(2310)에 내장될 수도 있다.

[0246]

더 일반적으로 설명하면, 본 출원에서 설명된 여러 가지 구현의 디코더는 완전한 타일을 포함하는 디코딩된 출력을 제공할 수 있다. 그 외에도 또는 대안으로서, 디코더는 타일 중에서 선택된 하나 이상의 화상(예컨대 이미지 또는 깊이 신호)만을 포함하는 디코딩된 출력을 제공할 수 있다.

[0247]

전술한 바와 같이, 고레벨 구문은 본 발명의 하나 이상의 실시예에 따른 시그널링을 수행하는데 이용될 수 있다. 고레벨 구문은 예컨대, 더 큰 프레임에 존재하는 코딩된 뷰의 수; 모든 뷰의 원래 폭과 높이; 각 코딩된 뷰에 대해 그 뷰에 대응하는 뷰 식별자; 각 코딩된 뷰에 대해 뷰의 프레임 분할 수; 그 뷰의 각 부분에 대해 (예컨대 플립핑 없음, 수평적 플립핑, 수직적 플립핑, 또는 수평 및 수직 플립핑일 수 있는) 플립핑 방향; 그

뷰의 각 부분에 대해 현재 부분이 그 뷰에 대한 마지막 프레임에 속하는 매크로블록의 픽셀 또는 수에서의 좌측 위치; 그 뷰의 각 부분에 대해 현재 부분이 그 뷰에 대한 마지막 프레임에 속하는 매크로블록의 픽셀 또는 수에서의 그 부분의 상단 위치; 그 뷰의 각 부분에 대해 현재 큰 디코딩/인코딩된 프레임에서 매크로블록의 픽셀 또는 수에서의 크롭핑 윈도우의 좌측 위치; 그 뷰의 각 부분에 대해 현재 큰 디코딩/인코딩된 프레임에서 매크로블록의 픽셀 또는 수에서의 크롭핑 윈도우의 우측 위치; 그 뷰의 각 부분에 대해 현재 큰 디코딩/인코딩된 프레임에서 매크로블록의 픽셀 또는 수에서의 크롭핑 윈도우의 상단 위치; 그 뷰의 각 부분에 대해 현재 큰 디코딩/인코딩된 프레임에서 매크로블록의 픽셀 또는 수에서의 크롭핑 윈도우의 하단 위치; 및 각 코딩된 뷰에 대해 그 뷰가 출력 전에 업샘플링될 필요가 있는지 여부(업샘플링이 수행될 필요가 있다면 고레벨 구문을 이용하여 (AVC 6-탭 필터, SVC 4-탭 필터, 바이리니어 필터 또는 커스텀 1D, 2D 선형 또는 비선형 필터를 포함하여(이에 한정되는 것은 아님)) 업샘플링 방법을 나타내는데 이용될 수 있는 경우) 중 어느 것을 표시하는데(이에 한정되는 것은 아님) 이용될 수 있다.

[0248] 용어 "인코더"와 "디코더"는 일반적인 구조를 함축하며 특정 기능이나 특성에 한정되는 것은 아님에 유의해야 한다. 예컨대, 디코더는 인코딩된 비트스트림을 전달하는 변조 반송파를 수신하여 그 인코딩된 비트스트림을 디코딩하는 것은 물론 복조할 수 있다.

[0249] 여러 가지 방법에 대해서 기술하였다. 이들 방법 중 많은 것은 충분한 공개를 제공하도록 상세히 설명된다. 그러나, 이들 방법에 대해 기술된 특정 특징들 중 하나 또는 여러 개를 변형할 수 있음을 알아야 한다.

[0250] 더욱이 몇 가지 구현에서는 특정 정보를 전송하기 위해 고레벨 구문을 이용하는 것에 대해 설명하였다. 그러나 다른 구현은 동일 정보(또는 그 정보의 변형)를 제공하기 위해 저레벨 구문 또는 실제로는(예컨대 정보를 인코딩된 데이터의 일부로서 전송하는 것과 같은) 다른 메카니즘을 함께 이용함을 알아야 한다.

[0251] 여러 가지 구현은 다중뷰(더 일반적으로는 화상)가 단일 화상으로 타일링되고 단일 화상으로서 인코딩되고 단일 화상으로서 전송될 수 있도록 타일링 및 적당한 시크널링을 제공한다. 이 시크널링 정보에 의해서 후처리기는 뷰/화상을 분리할 수가 있다. 또한, 타일링되는 복수의 화상은 뷰일 수 있지만, 적어도 하나는 깊이 맵일 수 있다. 이들 구현은 한 가지 이상의 이점을 제공할 수 있다. 예컨대 사용자는 다중뷰를 타일 형태로 디스플레이 하기를 원할 수 있으며, 이들 여러 가지 구현은 그러한 뷰를 타일 형태로 인코딩, 전송/저장하기 전에 타일링 함으로써 그러한 뷰를 인코딩 및 전송 또는 저장하는 효율적인 방법을 제공한다.

[0252] AVC 및/또는 MVC와 관련하여 다중뷰를 타일링하는 구현은 추가적인 이점을 제공한다. AVC는 표면적으로는 단일 뷰에만 이용되므로 추가적인 뷰는 예상되지 않는다. 그러나, 그와 같은 AVC 기반 구현은, 타일링된 뷰가 예컨대 디코더가 타일링된 화상이 여러 가지 뷰에 속한다는 것을(예컨대 의사 뷰 내의 상단 좌측 GYKT은 뷰 1이고, 상단 우측 화상은 뷰 2이고, 하는 등) 알 수 있도록 배열될 수 있기 때문에, AVC 환경에서 다중뷰를 제공할 수 있다.

[0253] 추가적으로, MVC는 이미 다중뷰를 포함하고 있으므로 단일 의사 뷰에 다중뷰가 포함될 것으로 예상되지 않는다. 더욱이, MVC는 지원가능한 뷰의 수가 한정되어 있으며, 그와 같은 MVC 기반 구현은 (AVC 기반 구현에서처럼) 추가적인 뷰가 타일링될 수 있도록 함으로써 지원가능한 뷰의 수를 효과적으로 증가시킨다. 예컨대 각 의사 뷰는 MVC의 지원된 뷰들 중 하나에 대응할 수 있으며, 디코더는 각 "지원된 뷰"가 실제로는 미리 배열된 타일링된 순서로 4개의 뷰를 포함한다는 것을 알 수 있다. 따라서 그와 같은 구현에서는 가능한 뷰의 수는 "지원된 뷰" 수의 4배이다.

[0254] 하기 설명에서는, SEI 메시지와 같은 고레벨 구문은 여러 가지 구현에서 복수의 공간 인터리빙 모드들 중 어느 것이 단일 화상으로서 타일링된 화상에 기인한 것인지에 대한 정보를 포함하도록 확장된다. 공간 인터리빙은 예컨대 병렬적, 상하, 수직 인터레이싱, 수평 인터레이싱 및 체커판과 같은 복수의 모드에서 일어날 수 있다. 그 외에도, 구문은 타일링된 화상 내의 콘텐츠에 대한 관계 정보를 포함하도록 확장된다. 관계 정보는 입체 영상 중의 좌측 및 우측 화상의 지정, 또는 2D 플러스 깊이가 화상에 이용되는 경우에는 깊이 화상으로서의 식별, 또는 화상이 계층적 깊이 비디오(LDV) 신호를 구성한다는 표시를 포함할 수 있다.

[0255] 전술한 바와 같이 구문은 슬라이스 헤더 레벨, PPS(Picture Parameter Set) 레벨, SPS(Sequence Parameter Set) 레벨, VPS(View Parameter Set) 레벨 및 NAL(Network Abstraction Layer) 단위 헤더 레벨에서의 구문과 같이, SEI 메시지 이외의 임의의 고레벨 구문에서 구현되는 것으로 생각된다. 그 외에도, 저레벨 구문은 정보를 표시하는데 이용될 수 있는 것으로 생각된다. 이 정보는 여러 가지 방식으로 대역 외에서 표시될 수 있는 것으로도 생각된다.

[0256] 전술한 AVC 초안 개정에서 설명한 바와 같은 체크판 공간 인터리빙을 위한 SEI 메시지 구문은 하기 표 6에서 정의된다.

표 6

spatially_interleaved_pictures( payloadSize ) {	C	Descriptor
spatially_interleaved_pictures_id	5	ue(v)
spatially_interleaved_pictures_cancel_flag	5	u(1)
if( !spatially_interleaved_pictures_cancel_flag ) {		
basic_spatial_interleaving_type_id	5	u(8)
spatially_interleaved_pictures_repetition_period	5	ue(v)
}		
additional_extension_flag	5	u(1)
}		

[0257]

[0258] 이 SEI 메시지는 디코더에게, 출력된 디코딩된 화상이 표시된 공간 인터리빙 방식을 이용하여 복수의 구별되는 화상의 공간적 인터리빙에 의해 구성된다는 것을 알려준다. 디코더는 이 SEI 메시지에서의 정보를 이용하여 디스플레이 또는 다른 목적으로 화상 데이터를 적절히 디인터리빙 또는 처리할 수 있다.

[0259] 표 6에서 공간적 인터리빙된 화상 SEI 메시지의 구문의 정의 값에 대한 의미는 다음과 같다.

[0260] spatially\_interleaved\_pictures\_id는 공간적으로 인터리빙된 화상 SEI 메시지의 용법을 확인하는데 이용될 수 있는 식별 번호를 포함한다.

[0261] spatially\_interleaved\_pictures\_cancel\_flag가 1인 것은 공간적으로 인터리빙된 화상 SEI 메시지가 이미 공간적으로 인터리빙된 화상 SEI 메시지의 유지를 출력 순으로 취소하는 것을 의미한다. spatially\_interleaved\_pictures\_cancel\_flag가 0인 것은 공간적으로 인터리빙된 화상 정보가 이어진다는 것을 의미한다.

[0262] basic\_spatial\_interleaving\_type\_id는 단일의 타일링된 화상에 포함된 복수의 화상의 공간적 인터리빙의 타입을 의미한다.

[0263] basic\_spatial\_interleaving\_type\_id가 0인 것은 디코딩된 화상의 각 성분 평면이 도 26에 도시된 바와 같은 2개 화상의 대응 평면의 "체크판" 방식 인터리빙을 포함한다는 것을 의미한다.

[0264] basic\_spatial\_interleaving\_type\_id가 1인 것은 디코딩된 화상의 각 성분 평면이 도 26에 도시된 바와 같은 2개 화상의 대응 평면의 "체크판" 방식 인터리빙을 포함한다는 것을 의미하고, 추가적으로 이 2개의 구성 화상이 도 26에 도시된 바와 같은 입체 뷰 장면의 좌측 뷰와 우측 뷰를 구성한다는 것을 의미한다.

[0265] spatially\_interleaved\_pictures\_repetition\_period는 공간적으로 인터리빙된 화상 SEI 메시지의 유지를 규정하고 spatially\_interleaved\_pictures\_id의 동일 값 또는 코딩된 비디오 계열의 끝을 가진 다른 공간적으로 인터리빙된 화상 SEI 메시지가 비트스트림 내에 존재하는 화상 순서 카운트 구간을 규정할 수 있다.

[0266] spatially\_interleaved\_pictures\_repetition\_period가 0인 것은 공간적으로 인터리빙된 화상 SEI 메시지가 현재 디코딩된 화상에만 적용된다는 것을 규정한다.

[0267] spatially\_interleaved\_pictures\_repetition\_period가 1인 것은 공간적으로 인터리빙된 화상 SEI 메시지가 하기 조건들 중 어느 것이 참일 때까지 출력 순서를 유지한다는 것을 규정한다.

[0268] - 새로운 코딩된 비디오 계열이 시작한다.

[0269] - spatially\_interleaved\_pictures\_id의 동일 값을 가진 공간적으로 인터리빙된 화상 SEI 메시지를 포함하는 액세스 단위의 화상이 PicOrderCnt(CurrPic)보다 큰 PicOrderCnt()를 가진 출력이다.

[0270] spatially\_interleaved\_pictures\_repetition\_period가 0또는 1인 것은 spatially\_interleaved\_pictures\_id의

동일 값을 가진 다른 공간적으로 인터리빙된 화상 SEI 메시지가 존재할 수도 하지 않을 수도 있다는 것을 의미한다.

- [0271] spatially\_interleaved\_pictures\_repetition\_period가 1보다 크다는 것은 공간적으로 인터리빙된 화상 SEI 메시지가 하기 조건들 중 어느 것이 참일 때까지 유지한다는 것을 규정한다.
- [0272] - 새로운 코딩된 비디오 계열이 시작한다.
- [0273] - spatially\_interleaved\_pictures\_id의 동일 값을 가진 공간적으로 인터리빙된 화상 SEI 메시지를 포함하는 액세스 단위의 화상이 PicOrderCnt(CurrPic)보다는 크고 PicOrderCnt(CurrPic) + spatially\_interleaved\_pictures\_repetition\_period보다 작거나 같은 PicOrderCnt()를 가진 출력이다.
- [0274] spatially\_interleaved\_pictures\_repetition\_period가 1보다 크다는 것은 비트스트림이 종료하지 않거나 새로운 코딩된 비디오 계열이 그러한 화상의 출력없이 시작하지 않는다면 spatially\_interleaved\_pictures\_id의 동일 값을 가진 다른 공간적으로 인터리빙된 화상 SEI 메시지가 PicOrderCnt(CurrPic)보다는 크고 PicOrderCnt(CurrPic) + spatially\_interleaved\_pictures\_repetition\_period보다 작거나 같은 PicOrderCnt()를 가진 출력인 액세스 단위의 화상에 대해 존재할 것이라는 것을 의미한다.
- [0275] additional\_extension\_flag가 0이라는 것은 공간적으로 인터리빙된 화상 SEI 메시지 내에 추가 데이터가 이어지지 않는다는 것을 의미한다.
- [0276] 표 6에 나타난 구문을 변경함이 없이 이 응용의 구현은 예시적인 SEI 메시지 내에 관계 정보와 공간 인터리빙 정보를 제공한다. basic\_spatial\_interleaving\_type\_id의 가능한 값의 범위는 이 구현에서 변경 확장되어 단 하나의 체커판 방법이 아니라 복수의 공간 인터리빙 방법을 나타낸다. 더욱이 파라미터 basic\_spatial\_interleaving\_type\_id는 특정 타입의 공간 인터리빙이 화상에 존재하고 구성 인터리빙된 화상들이 서로 관련이 있다는 것을 나타내는데 이용된다. 이 구현에서 basic\_spatial\_interleaving\_type\_id에 대한 의미는 다음과 같다.
- [0277] 값 2 또는 3은 단일 화상이 도 27에 도시된 2개 화상의 대응 평면의 "병렬적" 인터리빙을 포함한다는 것을 의미한다. 추가적으로 값 3은 2개의 구성 화상이 입체 뷰 장면의 좌측 및 우측 뷰를 구성한다는 것을 의미한다. 병렬적 인터리빙을 위해서 합성 화상이 2개의 이미지를 나란히 포함하도록 한 화상이 다른 화상 옆에 놓인다.
- [0278] 값 4 또는 5는 단일 화상이 도 28에 도시된 2개 화상의 대응 평면의 "상하" 인터리빙을 포함한다는 것을 의미한다. 추가적으로 값 5는 2개의 구성 화상이 입체 뷰 장면의 좌측 및 우측 뷰를 구성한다는 것을 의미한다. 상하 인터리빙을 위해서 합성 화상이 이미지를 상하로 갖는 것처럼 보이도록 한 화상이 다른 화상 위쳐 놓인다.
- [0279] 값 6 또는 7은 단일 화상이 도 29에 도시된 2개 화상의 대응 평면의 "행단위" 인터리빙 또는 단순히 행 인터리빙을 포함한다는 것을 의미한다. 추가적으로 값 7은 2개의 구성 화상이 입체 뷰 장면의 좌측 및 우측 뷰를 구성한다는 것을 의미한다. 행단위 인터리빙을 위해서 단일 화상의 연속한 행이 구성 화상들 간에 교호한다. 기본적으로 단일 화상은 구성 이미지들의 수평 슬라이스들이 교호하는 것이다.
- [0280] 값 8 또는 9는 단일 화상이 도 30에 도시된 2개 화상의 대응 평면의 "열단위" 인터리빙을 포함한다는 것을 의미한다. 추가적으로 값 9는 2개의 구성 화상이 입체 뷰 장면의 좌측 및 우측 뷰를 구성한다는 것을 의미한다. 열단위 인터리빙을 위해서 단일 화상의 연속한 열이 구성 화상들 간에 교호한다. 기본적으로 단일 화상은 구성 이미지들의 수직 슬라이스들이 교호하는 것이다.
- [0281] 관련 화상에 대한 정보를 전달하기 위해 예시적인 SEI 메시지에서 이용되는 구문의 다른 실시예에서는 추가 정보를 나타내기 위해 표 7에 몇 개의 추가적인 구문 요소를 포함시켰다. 이 추가적인 구문 요소는 예컨대 하나 이상의 구성 화상(예컨대 플립)의 배향을 나타내고, 이미지에 좌-우 입체쌍 관계가 존재하는지 여부를 별도로 나타내고, 어느 구성 화상에 업샘플링이 필요한지 여부를 나타내고, 업샘플링의 가능한 정도와 방향을 나타내기 위해 포함된다.



표 7

spatially_interleaved_pictures( payloadSize ) {	C	Descriptor
spatially_interleaved_pictures_id	5	ue(v)
spatially_interleaved_pictures_cancel_flag	5	U(1)
if( !spatially_interleaved_pictures_cancel_flag ) {		
basic_spatial_interleaving_type_id	5	U(8)
stereo_pair_flag	5	U(1)
upsample_conversion_horizontal_flag	5	U(1)
upsample_conversion_vertical_flag	5	U(1)
if( basic_spatial_interleaving_type_id == 1 or basic_spatial_interleaving_type_id == 2 ) {		
flip_flag	5	U(1)
}		
spatially_interleaved_pictures_repetition_period	5	ue(v)
}		
additional_extension_flag	5	U(1)
}		

[0282]

[0283]

의미는 다음과 같이 정의된다.

[0284]

**basic\_spatial\_interleaving\_type\_id**는 화상의 공간 인터리빙 타입을 나타낸다.

[0285]

**basic\_spatial\_interleaving\_type\_id**가 0이라는 것은 디코딩된 화상의 각 성분 평면이 앞서의 제안(도 26 참조)에서와 같이 2개 화상의 대응 평면의 "체커판" 방식 인터리빙을 포함한다는 것을 의미한다.

[0286]

**basic\_spatial\_interleaving\_type\_id**가 1이라는 것은 디코딩된 화상의 각 성분 평면이 도 27에 도시된 바와 같이 2개 화상의 대응 평면의 "병렬" 방식 인터리빙을 포함한다는 것을 의미한다.

[0287]

**basic\_spatial\_interleaving\_type\_id**가 2라는 것은 디코딩된 화상의 각 성분 평면이 도 28에 도시된 바와 같이 2개 화상의 대응 평면의 "상하" 방식 인터리빙을 포함한다는 것을 의미한다.

[0288]

**basic\_spatial\_interleaving\_type\_id**가 3이라는 것은 디코딩된 화상의 각 성분 평면이 도 29에 도시된 바와 같이 2개 화상의 대응 평면의 "행단위" 방식 인터리빙을 포함한다는 것을 의미한다.

[0289]

**basic\_spatial\_interleaving\_type\_id**가 4라는 것은 디코딩된 화상의 각 성분 평면이 도 30에 도시된 바와 같이 2개 화상의 대응 평면의 "열단위" 방식 인터리빙을 포함한다는 것을 의미한다.

[0290]

**stereo\_pair\_flag**는 2개의 구성 화상이 입체 뷰 장면의 좌측 및 우측 뷰를 구성하는 어떤 관계를 갖고 있다는 것을 나타낸다. 값 0은 구성 화상이 좌측 및 우측 뷰를 구성하지 않는다는 것을 의미한다. 값 1은 구성 화상이 이미지의 좌측 및 우측 뷰를 구성하는 것과 관련있다는 것을 의미한다.

[0291]

**upsample\_conversion\_horizontal\_flag**는 2개의 구성 화상이 디코딩 중에 단일 화상으로부터 추출된 후에 수평 방향에서 업샘플링을 필요로 하는지를 나타낸다. 값 0은 업샘플링이 필요없음을 의미한다. 이는 샘플링 계수가 0인 것에 해당한다. 값 1은 샘플링 계수 2에 의한 업샘플링이 필요함을 의미한다.

[0292]

**upsample\_conversion\_vertical\_flag**는 2개의 구성 화상이 디코딩 중에 단일 화상으로부터 추출된 후에 수직 방향에서 업샘플링을 필요로 하는지를 나타낸다. 값 0은 업샘플링이 필요없음을 의미한다. 이는 샘플링 계수가 0인 것에 해당한다. 값 1은 샘플링 계수 2에 의한 업샘플링이 필요함을 의미한다.

[0293]

뒤에 더 자세히 설명하겠지만, 업샘플링 동작의 많은 양상은 업샘플링 동작이 화상 디코딩 중에 적절하게 처리 되도록 SEI 메시지에서 전달될 수 있을 것으로 생각한다. 예컨대, 업샘플링을 위한 계수의 추가적 범위가 표시

될 수 있고, 업샘플링 필터의 타입도 표시될 수 있고, 디코더가 업샘플링을 위해 적당한 또는 최적의 필터를 결정할 수 있도록 다운 샘플링 필터도 표시될 수 있다. 또한 업샘플링 필터 계수의 수와 값을 포함하는 필터 계수 정보가 수신기가 바람직한 업샘플링 동작을 수행할 수 있도록 SEI 메시지에 더 표시될 수 있을 것으로 생각한다.

[0294] 샘플링 계수는 비디오 화상의 원래 크기와 샘플링된 크기 간의 비를 의미한다. 예컨대, 샘플링 계수가 2이면 원래 화상 크기는 샘플링된 화상 크기의 2배이다. 화상 크기는 보통은 픽셀의 해상도 척도이다. 그래서 수평적으로 다운샘플링된 화상은 원래 비디오 화상의 해상도를 복원하기 위해 같은 계수에 의한 대응 수평적 업샘플링을 필요로 한다. 예컨대 원래 화상의 폭이 1024 픽셀이라면 이는 샘플링 계수 2에 의해 수평적으로 다운샘플링되어 폭이 512 픽셀인 다운샘플링된 화상이 될 수 있다. 화상 크기는 보통은 픽셀의 해상도 척도이다. 수직 다운샘플링 또는 업샘플링에 대해서도 비슷한 분석이 보여질 수 있다. 샘플링 계수는 수평적 방식과 수직적 방식의 조합에 의존하는 다운샘플링 또는 업샘플링에 적용될 수 있다.

[0295] **flip\_flag**는 두 번째 구성 화상의 플립핑여부를 나타낸다. 값 0은 그 화상에 플립이 없다는 것을 의미한다. 값 1은 플립핑이 수행됨을 의미한다. 플립핑은 당업자라면 화상 평면에서 중심축에 대해 180° 회전시키는 것임을 잘 알 것이다. 플립핑 방향은 이 실시예에서는 인터리빙 타입에 따라서 결정된다. 예컨대, 구성 화상에 병렬 인터리빙(base\_spatial\_interleaving\_type\_id가 1인 것)이 있는 경우에는, flip\_flag의 적당한 값으로 표시된다면(도 31 참조), 우측 화상을 수평 방향에서(즉, 수직축에 대해) 플립핑하는 것이 바람직하다. 구성 화상에 상하 공간 인터리빙(base\_spatial\_interleaving\_type\_id가 2인 것)이 있는 경우에는, flip\_flag의 적당한 값으로 표시된다면(도 32 참조), 하단 화상을 수직 방향에서(즉, 중심 수평축에 대해) 플립핑하는 것이 바람직하다. 여기서는 두 번째 구성 화상을 플립핑하는 것에 대해 설명하지만, 다른 예시적인 실시예는 상하 인터리빙에서의 상단 화상 또는 병렬 인터리빙에서의 좌측 화상과 같은 첫 번째 화상을 플립핑하는 것일 수 있다고 생각한다. 물론 flip\_flag에 대한 이들 추가적인 자유도를 처리하기 위해서는 값 범위를 증가시키거나 이에 관계된 다른 의미를 도입할 필요가 있을 수 있다.

[0296] 또한 추가적인 실시예들은 수평 방향과 수직 방향 모두에서 화상을 플립핑할 수 있는 것으로 생각된다. 예컨대, 도 2에 도시된 바와 같이 4분면마다 하나의 뷰(즉, 화상)를 가진 단일 화상에 4개의 뷰가 함께 타일링되는 경우에는, 상좌 사분면은 플립핑되지 않는 제1 뷰를, 상우 사분면은 수평적으로만 플립핑된 제2 뷰를, 하좌 사분면은 수직적으로만 플립핑된 제3 뷰를, 하우 사분면은 수평적으로 그리고 수직적으로 모두 플립핑된 제4 뷰를 포함하도록 하는 것이 가능하다. 이런 식으로 타일링 또는 인터리빙함으로써 뷰들 간의 계면에 있는 경계는 그 화상 내의 인접 뷰들로부터 그 경계의 양측에 공통의 장면 콘텐츠를 가질 가능성이 큼을 알 수 있다. 이런 타입의 플립핑에 의해서 압축 효율이 더 개선될 수 있다. 이런 타입의 플립핑의 표시는 본 발명의 범위 내에 있다고 생각한다.

[0297] 더욱이 표 7의 구문은 2D 플러스 깊이 인터리빙 처리에 이용하는데 채택될 수 있다고 생각한다. 예컨대 그와 같은 포맷을 입력으로 받아들일 수 있는 상업적인 디스플레이가 개발될 수 있다. 그와 같은 용용을 위한 예시적인 구문은 표 8에 기술되어 있다. 의미들 중 많은 것은 이미 위에서 정의되어 있다. 새로 도입되는 의미에 대해서만 아래에 기술한다.

표 8

spatially_interleaved_pictures( payloadSize ) {	Descriptor
<b>spatially_interleaved_pictures_id</b>	ue(v)
<b>spatially_interleaved_pictures_cancel_flag</b>	u(1)
if( !spatially_interleaved_pictures_cancel_flag ) {	
<b>basic_spatial_interleaving_type_id</b>	u(8)
<b>semantics_id</b>	u(1)
<b>upsample_conversion_horizontal_flag</b>	u(1)
<b>upsample_conversion_vertical_flag</b>	u(1)
<b>spatially_interleaved_pictures_repetition_period</b>	ue(v)
if( basic_spatial_interleaving_type_id == 1 or basic_spatial_interleaving_type_id == 2 ) {	
<b>flip_flag</b>	u(1)
}	
if( semantics_id == 1 ) {	
camera_parameter_set()	
}	
}	
<b>additional_extension_flag</b>	u(1)
}	

[0298]

[0299]

camera\_parameter\_set는 카메라에 관련된 장면의 파라미터들을 나타낸다. 이들 파라미터는 이 동일 SEI 메시지에 나타날 수 있으며 또는 다른 SEI 메시지에서 전달 수 있으며, 또는 이들 파라미터는 SPS, PPS, 슬라이스 헤더 등과 같은 다른 구문 내에 나타날 수 있다. 카메라 파라미터는 일반적으로 적어도 초점 거리, 기준선 거리, 카메라 위치(들), Znear(장면과 카메라 간의 최소 거리) 및 Zfar(장면과 카메라 간의 최대 거리)를 포함한다. 이 카메라 파라미터 세트는 3x3 고유 행렬, 3x3 회전 행렬 및 3D 변환 벡터를 포함하는 카메라마다의 전체 파라미터 세트를 포함할 수도 있다. 카메라 파라미터는 예컨대 렌더링 및 코딩에 이용될 수 있다.

[0300]

표 8에 나타난 것과 비슷한 SEI 메시지 또는 다른 구문을 이용하는 경우에는 비디오와 깊이 간의 시스템 레벨에서의 동기화를 하지 않아도 될 수 있다. 이는 비디오와 그 관련 깊이가 이미 하나의 단일 프레임으로 타일링되어 있기 때문이다.

[0301]

도 33에는 화상에 대한 LDV(Layer Depth Video) 포맷이 나타나 있다. 이 도에서는 4개의 화상이 병렬적 및 상하 인터리빙되어 복합 화상을 구성한다. 상좌 사분면 화상은 중심 뷰 계층을 나타내고 상우 사분면 화상은 중심 깊이 계층을 나타낸다. 유사하게, 하좌 사분면 화상은 폐색(occlusion) 뷰 계층을 나타내고 하우 사분면 화상은 폐색 깊이 계층을 나타낸다. 도 33에 나타난 포맷은 상업적 자동 입체 디스플레이를 위한 입력 포맷으로서 이용될 수 있다.

[0302]

LDV의 존재는 앞서의 실시예들에서 보인 것과 유사한 구문을 이용하는 것을 나타내었다. 주로 semantics\_id의 의미는 다음과 같이 LDV 옵션을 도입하기 위해 확장된다.



표 9

spatially_interleaved_pictures( payloadSize ) {	C	Descriptor
spatially_interleaved_pictures_id	5	ue(v)
spatially_interleaved_pictures_cancel_flag	5	u(1)
if( !spatially_interleaved_pictures_cancel_flag ) {		
basic_spatial_interleaving_type_id	5	u(8)
semantics_id	5	u(1)
upsample_conversion_horizontal_flag	5	u(1)

[0303]

upsample_conversion_vertical_flag	5	u(1)
spatially_interleaved_pictures_repetition_period	5	ue(v)
if (basic_spatial_interleaving_type_id == 1 or basic_spatial_interleaving_type_id == 2) {		
flip_flag	5	u(1)
}		
if (semantics_id == 1    semantics_id == 3) {		
camera_parameter_set()		
}		
additional_extension_flag	5	u(1)
}		

[0304]

[0305]

semantics\_id는 화상들 간의 관계, 즉, 인터리빙된 화상이 서로에 대해 물리적으로 무엇을 의미하는지를 나타낸다.

[0306]

값 0 내지 2는 2개의 화상이 상기 정의된 바와 같이 관계가 있다는 것을 의미한다. 값 0은 2개의 화상이 좌측 및 우측 뷰를 구성한다는 것을 의미한다. 값 1은 두 번째 화상이 첫 번째 화상, 즉 2D+Z 화상 세트의 대응 깊이 맵을 나타냄을 의미한다. 값 2는 2개의 인터리빙된 화상 간의 관계가 특정되지 않음을 의미한다.

[0307]

값 3은 4개의 구성 화상이 인터리빙되고 이들이 예컨대 도 33에 도시된 바와 같이 LDV 표현의 4개의 구성 화상에 대응함을 의미한다. 3보다 큰 값은 추가적인 관계 표시에 이용될 수 있다.

[0308]

여기서 생각하는 추가적인 관계 표시는 다중뷰가 복수의 2D+Z 화상 세트라는 표시(MVD(MultiView plus Depth)라고 함)와, 다중뷰가 2개의 LDV 화상 세트 중의 이미지를 나타내는 표시(DES(Depth Enhanced Stereo)라고 함)를 포함하나 이에 한정되는 것은 아니다.

[0309]

semantics\_id가 3이고 basic\_spatial\_interleaving\_type\_id가 1(병렬) 또는 2(상하)인 경우에는 4개의 구성 화상은 도 33에 도시된 바와 같이 인터리빙된다.

[0310]

semantics\_id가 3이고 basic\_spatial\_interleaving\_type\_id가 0인 경우에는, 뷰 0에 대한 뷰 1, 2 및 3의 오프셋이 그 구문 내의 새로운 의미와 같은 것을 통해 특정 표시에 의해 추가적으로 표시된다면, 4개의 화상은 도 33에 도시된 바와 같이 예 1(요소(2210)) 또는 예 2(요소(2220))로서 인터리빙된다. semantics\_id가 3인 경우에는 도 22의 예 1(요소(2210))에 나타난 바와 같이 LDV 화상에 대한 인터리빙이 수행된다고 생각된다. 더욱이 semantics\_id가 4인 경우에는 도 22의 예 2(요소(2220))에 나타난 바와 같이 LDV 화상에 대한 인터리빙이 수행된다고 생각된다. 오프셋이 있는 LDV와 오프셋이 없는 LDV에 대한 인터리빙은 도 22와 관련하여 위에서 이미 설명하였다.

[0311]

전술한 바와 같이, 3D/스테레오 비디오 콘텐츠는 2D 비디오 코딩 표준을 이용하여 인코딩될 수 있다. 프레임 패킹 방법에 의하면 공간적으로 또는 시간적으로 다운샘플링된 뷰는 패킹되어 인코딩을 위한 하나의 프레임으로

될 수 있다. 디코딩된 프레임은 언패킹되어 구성 다중뷰로 되고, 이는 다시 일반적으로 원래의 해상도로 업샘플링된다.

[0312] 업샘플링과, 특히 업샘플링 필터는 구성 뷰 또는 화상의 재구성 품질에 중요한 역할을 한다. 업샘플링 필터는 통상적으로 인코더에서 이용된 다운샘플링 필터에 따라서 선택된다. 다운샘플링 또는 업샘플링 필터에 대한 정보가 비트스트림으로 또는 다른 비표준적 수단에 의해 디코더에 전달되면 각 업샘플링된 필터의 품질이 개선될 수 있다. 디코더에서의 업샘플링 필터의 특성은 역관계 또는 상호관계 같은 것에 의해서만 반드시 인코더에서의 다운샘플링 필터의 특성에 구속될 필요는 없지만, 매칭된 업샘플링 필터는 최적의 신호 재구성을 제공할 것으로 예상할 수 있다. 실험적 실습에 따른 적어도 한 가지 구현에서는 SEI 메시지는 업샘플링 파라미터를 디코더에 표시할 수 있다는 것이 확인되었다. 다른 구현에서는 업샘플링 및 다운샘플링 필터 파라미터는 SEI 메시지 내에서 비디오 디코더에 표시되는 것으로 생각된다. 전술한 바와 같이 도 3과 4에는 각각 H.264/MPEG-4 AVC 비디오 인코더와 디코더가 도시되어 있다. 이들 장치는 여기서 개시된 여러 가지 구현의 양상의 실행에 적당하다. 그 외에도 H.264/MPEG-4 AVC에 맞게 제작되지 않거나 이를 가지고 동작할 수 없는 다른 장치라도 본 출원에서 설명된 여러 가지 구현의 실행에 적합할 수 있다.

[0313] 전술한 바와 같이, 프레임 패킹 업샘플 SEI 메시지는 여러 가지 업샘플링 필터를 표시하기 위해 H.264/MPEG-4 AVC 프레임 패킷의 체계 내에서 하기와 같이 정의된다. 이 구문은 SEI 메시지에만 한정되는 것이 아니며, 예컨대 SPS, PPS, VPS, 슬라이스 헤더 및 NAL 헤더와 같은 다른 고레벨 구문에서 이용되는 것과 같이 다른 수단에 의해서도 메시지 정보가 제공될 수 있다고 생각된다.

[0314] 업샘플링 정보를 이용하는 예시적인 실시예에서 업샘플링 필터는 하기 형태:

[0315]  $c_k, 0, c_{k-1}, 0, \dots, c_1, 1, c_1, 0, \dots, 0, c_{k-1}, 0, c_k$  로 표현되는 k개의 비영 계수로 정의된 홀수개의 탭을 가진 2차원 대칭 FIR 필터로서 선택된다.

[0316] 필터 계수  $c_i$ 는 모든 계수  $c_i$ 의 합이 범위  $\{1, \dots, k\}$ 에서 i에 대해 0.5가 되도록 정규화된다.

[0317] 상기 필터 파라미터를 이용하는 예시적인 수평 업샘플링 프로세스는 하기에서 더 자세히 설명되는 바와 같이 일련의 단계를 통해 기술된다. 원 이미지는  $m \times n$  행렬 X로 나타낸다. 행렬 A는 하기 속성을 가진  $n \times (2k+n)$  행렬로서 정의된다.

[0318] 
$$A = \begin{bmatrix} D & | & 0_{(n-k),k} \\ 0_{(n-k),k} & | & D \end{bmatrix}$$

[0319] 
$$D = \text{adiag} \left( \overbrace{1, \dots, 1}^k \right)$$

여기서,  $\text{adiag}(\cdot)$ 는  $k \times k$  반대 대각 항등 행렬이다. 원 이미지 X는 다음과 같은 텐서 곱연산을 이용해 새로운 이미지 행렬 X'로 변환된다.

[0320] 
$$X' = (XA) \otimes [1 \ 0]$$

[0321] 행렬 H는 다음과 같이 각 연속 행을 따라 시프트되고 선단 및/또는 후단 제로가 채워진 상기 k개의 비영 계수 (즉,  $c_k, 0, c_{k-1}, 0, \dots, c_1, 1, c_1, 0, \dots, 0, c_{k-1}, 0, c_k$ )를 포함하는  $n \times (4k+n-1)$  행렬로서 정의된다.

[0322] 
$$H = \begin{bmatrix} c_k & 0 & \dots & c_1 & 1 & c_1 & \dots & c_k & 0 & 0 & \dots \\ 0 & c_k & 0 & \dots & c_1 & 1 & c_1 & \dots & c_k & 0 & \dots \\ \dots & & & & & & & & & & \\ \dots & 0 & c_k & 0 & \dots & c_1 & 1 & c_1 & \dots & c_k & 0 \end{bmatrix}$$

[0323] 그러면 수평 업샘플링 후의 출력 이미지 행렬 Y는 다음과 같이 나타난다.

[0324] 
$$Y^T = HX'^T$$

[0325] 유사하게, 예시적인 수직 업샘플링 프로세스는 다음과 같이 기술된다. 새로운 행렬 A'는 다음과 같은 행렬로 정의된다.

$$A' = \begin{bmatrix} D & \\ 0_{(m-k),k} & D \end{bmatrix} \begin{bmatrix} I_{m,m} \\ D \end{bmatrix} \begin{bmatrix} 0_{(m-k),k} \\ D \end{bmatrix}$$

[0327] 그러면 이미지 Y는 다음과 같은 텐서 곱연산을 이용해 새로운 이미지 행렬 Y'로 변환된다.

$$Y' = (YA') \otimes [1 \ 0]$$

[0329] 그러면 수직 업샘플링 후의 출력 이미지 행렬은 다음과 같이 나타난다.

$$F^T = HY'^T$$

[0331] 행렬 F는 수평 및 수직 업샘플링 변환 후의 최종 이미지 행렬이다.

[0332] 상기 예에서 업샘플링은 수평 동작과 수직 동작 모두로서 기술하였다. 동작 순서는 수직 업샘플링이 수평 업샘플링 전에 수행되도록 뒤바뀔 수 있다고 생각된다. 더욱이 특정 경우에는 한 종류의 업샘플링만이 수행될 수 있다고 생각된다.

[0333] 디코더에 업샘플링 정보를 전달하는 SEI 메시지를 이용하기 위해서는 상기 설명한 구문들 내에 특정 의미가 포함되어야 한다고 생각된다. 하기의 표 10에 나타난 구문은 비디오 디코더에서의 업샘플링에 유용한 정보를 포함한다. 하기의 구문은 업샘플링 정보를 전달하는데 필요한 파라미터만을 보여주도록 축약된 것이다. 당업자라면 하기에 나타난 구문은 앞서의 구문들 중 하나 이상과 조합되어 예컨대 관계 표시, 배향 표시, 업샘플 표시 및 공간 인터리빙 표시에 관한 상당한 량의 프레임 패킹 정보를 전달할 수 있다는 것을 잘 알 것이다.

표 10

Frame_packing_filter( payloadSize ) {	Descriptor
for (c = 0; c<3; c++)	
{	
number_of_horizontal_filter_parameters[c]	ue(v)
for (i=0; i< number_of_horizontal_filter_parameters[c]; i++)	
{	
h[c][i]	s(16)
}	
number_of_vertical_filter_parameters[c]	ue(v)
for (i=0; i< number_of_horizontal_filter_parameters[c]; i++)	
{	
v[c][i]	s(16)
}	

[0334]

}	
}	

[0335]

[0336] 구문 요소들의 의미는 하기에 다음과 같이 정의된다.

[0337] number\_of\_horizontal\_filter\_parameters[c]는 색성분 c에 대한 (4k-1) 탭 수평 필터의 순서를 의미한다. 여

기서  $c$ 는 3개의 서로 다른 색성분 중 하나이다.

[0338]  $h[c][i]$ 는  $-2^{15}$  내지  $2^{15}-1$  범위에 있는 16 비트 정밀도의 색성분  $c$ 에 대한 수평 필터의  $i$ 번째 계수를 의미한다. 이 필터의 표준은  $2^{16}$ 이다.

[0339]  $number\_of\_vertical\_filter\_parameters$ 는 색성분  $c$ 에 대한  $(4k-1)$  탭 수직 필터의 순서를 의미한다.

[0340]  $v[c][i]$ 는  $-2^{15}$  내지  $2^{15}-1$  범위에 있는 16 비트 정밀도의 색성분  $c$ 에 대한 수직 필터의  $i$ 번째 계수를 의미한다. 이 필터의 표준은  $2^{16}$ 이다.

[0341] 이 구문에 대해서는 공지의 필터 타입이 디코더에 존재한다고 가정한다. 그러나 전문한 2차원 대칭 FIR 필터에 더하여 또는 이를 대신하여 여러 가지 다른 형태의 업샘플링 필터가 이용될 수 있다고 생각한다. 그와 같은 다른 업샘플링 필터는 바이리니어 필터, 큐빅(cubic) 필터, 스플라인(spline) 필터, 비너(Wiener) 필터, 또는 최소 제곱 필터와 같은 보간 필터 설계를 포함하나 이에 한정되는 것은 아니다. 다른 필터 형태를 이용할 수 있는 경우에는 상기 구문은 업샘플링을 위해 필터 형태에 대한 정보를 표시하는 하나 이상의 의미를 포함하도록 확장되어야 한다고 생각한다.

[0342] 당업자라면 업샘플링 필터 파라미터는 다운샘플링 필터 정보 등과 같은 여러 가지 입력에 기초하여 SEI 메시지에 포함되도록 인코더에서 도출될 수 있다는 것을 잘 알 것이다. 그 외에도 그와 같은 정보가 SEI 메시지에 포함되도록 단순히 인코더에 제공될 수 있다는 것을 잘 알 것이다.

[0343] 더욱이 표 10에 나타난 구문은 업샘플링 필터 파라미터 대신에 다운샘플링 필터 파라미터를 전송하도록 변경될 수 있다고 생각한다. 이 실시예에서, 다운샘플링 필터의 디코더에의 표시는 디코딩 응용에 다운샘플링 필터 파라미터에 기초하여 최적의 업샘플링 필터 파라미터를 결정할 수 있는 능력을 제공한다. 유사한 구문과 의미를 이용하여 다운샘플링 필터를 표시할 수 있다. 파라미터가 다운샘플링 필터 또는 업샘플링 필터를 나타내는 것을 표시하는 의미가 포함될 수 있다.

[0344] 최적 업샘플링 필터가 디코더에 의해 결정되어야 한다면 디코더는 그러한 결정을 종래 방식에 따라 할 수 있다. 그 외에도, 디코더는 최적이지 아니거나 반드시 최적인 것은 아닌 업샘플링 필터를 결정할 수 있다. 디코더에서의 그러한 결정은 처리 제한, 디스플레이 파라미터 고려 사항 등에 기초할 수 있다. 다른 예로서, 디코더는 처음에는 복수의 서로 다른 형태의 업샘플링 필터를 선택하고, 최종적으로 어떤 디코더 기준에 따라서 그 디코더에 최적인 것으로 결정된 결과를 생성하는 하나의 필터를 선택할 수 있다.

[0345] 업샘플링 필터 정보의 표시에 관련된 예시적인 인코딩 기법을 보여주는 플로우 차트는 도 35에 나타나 있으며, 이에 대해 이하에서 간략하게 설명한다. 처음에는 인코더 구성이 결정되고 고레벨 구문이 생성된다. 화상(예컨대 뷰)들은 프레임으로의 패키징을 위해 다운샘플링 필터에 의해 다운샘플링된다. 다운샘플링된 화상은 공간적으로 인터리빙되어 프레임으로 된다. 업샘플링 필터 파라미터는 인코더에 의해 결정되거나 인코더에 직접 공급된다. 일 실시예에서, 형태, 크기 및 계수 값과 같은 업샘플링 필터 파라미터는 다운샘플링 필터로부터 도출된다. 색성분(예컨대 YUV, RGB)마다 수평 및 수직 업샘플링 필터 파라미터의 수와 값이 결정된다. 이것은 도 35에서 3개의 루프, 즉 성분들에 대한 제1 루프, 어떤 성분에 대한 수평 필터 파라미터에 대한 제2 루프(제1 루프 내에 포함됨), 및 어떤 성분에 대한 수직 필터 파라미터에 대한 제3 루프(제1 루프 내에 포함됨)로 나타나 있다. 그 다음에 업샘플링 필터 파라미터는 SEI 메시지에 기록된다. 그런 다음에 SEI 메시지는 비디오 이미지 비트스트림과는 (대역외로) 별도로 또는 비디오 이미지 비트스트림과 (대역내로) 함께 디코더로 보내진다. 인코더는 필요에 따라 메시지를 패키징된 프레임 계열과 조합한다.

[0346] 도 35에 나타난 인코딩 방법과는 달리 도 37에 나타난 대안 실시예는 업샘플링 필터 파라미터 대신에 다운샘플링 필터 파라미터가 SEI 메시지에 포함된 것을 보여준다. 이 실시예에서, 업샘플링 필터 파라미터는 인코더에 의해 도출되지 않고, 패키징된 프레임과 함께 디코더에 전달되지도 않는다.

[0347] 업샘플링 필터 정보의 표시에 관련된 예시적인 디코딩 기법을 보여주는 플로우 차트는 도 36에 나타나 있으며, 이에 대해 이하에서 간략하게 설명한다. 먼저, 디코더에 의해 SEI 메시지와 기타 다른 관련 메시지가 수신되고, 디코더에 의해 수문이 읽혀진다. SEI 메시지는 샘플링 정보, 인터리빙 정보 등을 포함하는 정보를 결정하기 위해 분석된다. 이 예에서, 색성분마다의 업샘플링 필터 정보가 SEI 메시지에 포함되는 것으로 판단된다. 수직 및 수평 업샘플링 필터 정보는 추출되어 각 수평 및 수직 업샘플링 필터 계수의 수와 값을 얻는다. 이 추출은 도 36에서 3개의 루프, 즉 성분들에 대한 제1 루프, 어떤 성분에 대한 수평 필터 파라미터에 대한 제

2 루프(제1 루프 내에 포함됨), 및 어떤 성분에 대한 수직 필터 파라미터에 대한 제3 루프(제1 루프 내에 포함됨)로 나타나 있다. 그런 다음에 SEI 메시지는 저장되고, 패킹된 프레임 계열은 디코딩되어 내부의 패킹된 화상을 얻는다. 그런 다음에 이 화상은 복구된 업샘플링 필터 파라미터를 이용하여 업샘플링되어 화상을 그 원래의 완전 해상도로 복구한다.

[0348] 도 36에 나타난 디코딩 방법과는 달리 도 38에 나타난 대안 실시예는 업샘플링 필터 파라미터 대신에 다운샘플링 필터 파라미터가 SEI 메시지에 포함된 것을 보여준다. 이 실시예에서, 업샘플링 필터 파라미터는 인코더에 의해 도출되지 않고, 패킹된 프레임과 함께 디코더에 전달되지도 않는다. 그러므로, 디코더는 수신된 다운샘플링 필터 파라미터를 이용하여, 패킹된 비디오 프레임에서의 인터리빙으로부터 추출된 화상에 대한 원래의 완전 해상도를 복구하는데 이용될 다운샘플링 필터 파라미터를 도출한다.

[0349] 도 39는 본 발명의 구현에 따라서 본 발명의 원리가 적용될 수 있는 예시적인 비디오 송신 시스템(2500)을 보여준다. 비디오 송신 시스템(2500)은 예컨대 위성, 케이블, 전화선 또는 지상파 방송과 같은 여러 가지 매체를 이용하여 신호를 송신하는 헤드 엔드 또는 송신 시스템일 수 있다. 이 송신은 인터넷 또는 기타 다른 네트워크를 통해 제공될 수 있다.

[0350] 비디오 송신 시스템(2500)은 깊이를 가진 압축 비디오를 생성 및 송출할 수 있다. 이는 깊이 정보 또는 예컨대 디코더를 가질 수 있는 수신기단에서 이 깊이 정보를 합성하는데 이용될 수 있는 정보를 포함하는 인코딩된 신호(들)를 생성함으로써 달성된다.

[0351] 비디오 송신 시스템(2500)은 인코더(2510)와, 인코딩된 신호를 송신할 수 있는 송신기(2520)를 포함한다. 인코더(2510)는 비디오 정보를 수신하고 깊이를 가진 인코딩된 신호(들)를 생성한다. 인코더(2510)는 예컨대 여러 가지 정보를 수신하여 이를 저장 또는 송신을 위해 구조화된 포맷으로 조립하는 어셈블리 유닛을 포함하는 서브모듈을 포함할 수 있다. 이 여러 가지 정보는 예컨대 코딩 또는 미코딩 비디오, 코딩 또는 미코딩 깊이 정보, 및 예컨대 동작 벡터, 코딩 모드 표시기 및 구문 요소와 같은 코딩 및 미코딩 요소를 포함할 수 있다.

[0352] 송신기(2520)는 예컨대 인코딩된 화상 및/또는 그 관련 정보를 나타내는 하나 이상의 비트스트림을 가진 프로그램 신호를 송신하도록 구성될 수 있다. 통상적인 송신기는 예컨대 예러 정정 코딩, 신호 내의 데이터 인터리빙, 신호 내의 에너지 랜더마이징(randomizing) 및/또는 하나 이상의 반송파로의 신호 변조 중 한 가지 이상과 같은 기능을 수행한다. 이 송신기는 안테나(미도시)를 포함하거나 이와 인터페이스할 수 있다. 따라서 송신기(2520)의 구현은 변조기를 포함하거나 이에 한정될 수 있다.

[0353] 도 40은 본 발명의 실시예에 따라서 본 발명의 원리가 적용될 수 있는 예시적인 비디오 수신 시스템(2600)을 보여준다. 비디오 수신 시스템(2600)은 예컨대 위성, 케이블, 전화선 또는 지상파 방송과 같은 여러 가지 매체를 통해 신호를 수신하도록 구성될 수 있다. 이 신호는 인터넷 또는 기타 다른 네트워크를 통해 수신될 수 있다.

[0354] 비디오 수신 시스템(2600)은 인코딩된 비디오를 수신하는 예컨대 셀폰, 컴퓨터, 셋톱 박스, 텔레비전 또는 기타 다른 장치일 수 있으며, 예컨대 사용자에게 표시하거나 저장을 위해 디코딩된 비디오를 제공한다. 따라서 비디오 수신 시스템(2600)은 예컨대 텔레비전 스크린, 컴퓨터 모니터, (저장, 처리 또는 표시를 위한) 컴퓨터, 또는 기타 다른 저장, 처리 또는 표시 장치에 그 출력을 제공할 수 있다.

[0355] 비디오 수신 시스템(2600)은 비디오 정보를 포함하는 비디오 콘텐츠를 수신하여 처리할 수 있다. 비디오 수신 시스템(2600)은 예컨대 본 출원의 구현에서 설명된 신호와 같은 인코딩된 신호를 수신할 수 있는 수신기(2610)와, 수신된 신호를 디코딩할 수 있는 디코더(2620)를 포함한다.

[0356] 수신기(2610)는 예컨대 인코딩된 화상을 나타내는 복수의 비트스트림을 가진 프로그램 신호를 수신하도록 구성될 수 있다. 통상적인 수신기는 예컨대 변조되고 인코딩된 데이터 신호 수신, 하나 이상의 반송파로부터의 데이터 신호 복조, 신호 내의 에너지 디랜더마이징(de-randomizing), 신호 내의 데이터 디인터리빙 및/또는 신호 예러 정정 디코딩 중 한 가지 이상과 같은 기능을 수행한다. 수신기(2610)는 안테나(미도시)를 포함하거나 이와 인터페이스할 수 있다. 수신기(2610)의 구현은 복조기를 포함하거나 이에 한정될 수 있다. 디코더(2620)는 비디오 정보와 깊이 정보를 포함하는 비디오 신호를 출력한다.

[0357] 도 41은 본 발명의 실시예에 따라서 본 발명의 원리가 적용될 수 있는 예시적인 비디오 처리 장치(2700)를 보여준다. 비디오 처리 장치(2700)은 인코딩된 비디오를 수신하는 예컨대 셋톱 박스 또는 기타 다른 장치일 수 있으며, 예컨대 사용자에게 표시하거나 저장을 위해 디코딩된 비디오를 제공한다. 따라서 비디오 수신 장치(2700)는 텔레비전, 컴퓨터 모니터, 또는 컴퓨터 또는 기타 다른 처리 장치에 그 출력을 제공할 수 있다.



- [0358] 비디오 처리 장치(2700)는 프론트 엔드(FE) 장치(2705)와 디코더(2710)를 포함한다. 프론트 엔드 장치(2705)는 예컨대 인코딩된 화상을 나타내는 복수의 비트스트림을 가진 프로그램 신호를 수신하고 이 복수의 비트스트림 중에서 디코딩을 위한 하나 이상의 비트스트림을 선택하도록 구성된 수신기일 수 있다. 통상적인 수신기는 예컨대 변조되고 인코딩된 데이터 신호 수신, 데이터 신호 복조, 데이터 신호의 하나 이상의 인코딩(예컨대 채널 코딩 및/또는 소스 코딩)의 디코딩, 및/또는 데이터 신호 에러 정정 중 한 가지 이상과 같은 기능을 수행한다. 프론트 엔드 장치(2705)는 예컨대 안테나(미도시)로부터 프로그램 신호를 수신할 수 있다. 프론트 엔드 장치(2705)는 수신된 데이터 신호를 디코더(2710)에 제공한다.
- [0359] 디코더(2710)는 데이터 신호(2720)를 수신한다. 데이터 신호(2720)는 예컨대 하나 이상의 AVC(Advanced Video Coding), SVC(Scalable Video Coding) 또는 MVC(Multi-view Video Coding) 호환 스트림을 포함할 수 있다.
- [0360] 디코더(2710)는 수신된 신호(2720)의 전부 또는 일부를 디코딩하고, 디코딩된 비디오 신호(2730)를 출력으로서 제공한다. 디코딩된 비디오(2730)는 선택기(2750)에 제공된다. 장치(2700)는 사용자 입력(2770)을 수신하는 사용자 인터페이스(2760)도 포함한다. 사용자 인터페이스(2760)는 사용자 입력(2770)에 기초하여 화상 선택 신호(2780)를 선택기(2750)에 제공한다. 화상 선택 신호(2780)와 사용자 입력(2770)은 사용자가 복수의 화상, 계열, 스케일러블 버전, 뷰 또는 가용 디코딩된 데이터의 선택 중 어느 것을 표시하고자 하는 것을 나타낸다. 선택기(2750)는 선택된 화상(들)을 출력(2790)으로 제공한다. 선택기(2750)는 화상 선택 정보(2780)를 이용하여 디코딩된 비디오(2730) 중의 어느 화상을 출력(2790)으로서 제공할 것인지를 선택한다.
- [0361] 여러 가지 구현에서, 선택기(2750)는 사용자 인터페이스(2760)를 포함하고, 다른 구현에서는, 선택기(2750)가 별도의 인터페이스 기능을 수행하지 않고 바로 사용자 입력(2770)을 수신하기 때문에 사용자 인터페이스(2760)가 필요없다. 선택기(2750)는 예컨대 소프트웨어로 또는 집적 회로에서 구현될 수 있다. 일 구현에서, 선택기(2750)는 디코더(2710)와 통합되며, 다른 구현에서, 디코더(2710), 선택기(2750) 및 사용자 인터페이스(2760)이 모두 통합된다.
- [0362] 일 응용에서, 프론트 엔드(2705)는 각종 텔레비전 쇼의 방송을 수신하고 그 중 하나를 선택하여 처리한다. 한 쇼의 선택은 시청을 원하는 채널의 사용자 입력에 기초한다. 도 41에는 프론트 엔드 장치(2705)에의 사용자 입력이 나타나 있지 않지만, 프론트 엔드 장치(2705)는 사용자 입력(2770)을 수신한다. 프론트 엔드(2705)는 그 방송을 수신하여, 그 방송 스펙트럼의 관련 부분을 복조하고 그 복조된 쇼의 아우트(outer) 인코딩을 디코딩함으로써 원하는 쇼를 처리한다. 프론트 엔드(2705)는 디코딩된 쇼를 디코더(2710)에 제공한다. 디코더(2710)는 장치들(2760, 2750)을 포함하는 집적 유닛이다. 따라서 디코더(2710)는 그 쇼 중에서 시청하기를 원하는 뷰의 사용자 제공 표시인 사용자 입력을 수신한다. 디코더(2710)는 그 선택된 뷰는 물론 다른 뷰 중의 필요한 기준 화상을 디코딩하고, 디코딩된 뷰(2790)를 텔레비전(미도시)에 표시하기 위해 제공한다.
- [0363] 상기 응용을 계속해서 설명하면, 사용자는 디스플레이되는 뷰를 전환하고자 할 수 있으며, 그러면 디코더(2710)에 새로운 입력을 제공할 수 있다. 디코더(2710)는 사용자로부터 "뷰 변경"을 수신한 후에 이전 뷰와 새로운 뷰 뿐만 아니라 이전 뷰와 새로운 뷰 사이에 있는 뷰도 디코딩한다. 즉, 디코더(2710)는 이전 뷰를 촬영하는 카메라와 새로운 뷰를 촬영하는 카메라 사이에 물리적으로 위치한 카메라에 의해 촬영된 뷰를 디코딩한다. 프론트 엔드 장치(2705)는 또한 이전 뷰, 새로운 뷰, 및 이들 사이의 뷰를 식별하는 정보를 수신한다. 그와 같은 정보는 예컨대 뷰의 위치에 대한 정보를 가진 컨트롤러(도 41에는 미도시), 또는 디코더(2710)에 의해 제공될 수 있다. 다른 구현은 프론트 엔드 장치와 통합된 컨트롤러를 가진 프론트 엔드 장치를 이용할 수 있다.
- [0364] 디코더(2710)는 이들 디코딩된 뷰 모두를 출력(2790)으로서 제공한다. 후처리기(도 41에는 미도시)는 이전 뷰로부터 새로운 뷰로의 부드러운 전이를 제공하기 위해 뷰들 간을 보간하고, 이러한 전이를 사용자에게 표시한다. 새로운 뷰로의 전이 후에 후처리는 디코더(2710)와 프론트 엔드 장치(2705)에게 새로운 뷰만이 필요하다는 것을 (하나 이상의 통신 링크(미도시)를 통해) 알린다. 그 후, 디코더(2710)는 새로운 뷰를 출력(2790)으로서 제공하기만 한다.
- [0365] 시스템(2700)은 이미지 계열의 다중뷰를 수신하고, 디스플레이를 위해 단일 뷰를 제시하고, 여러 가지 뷰들 간을 부드럽게 전환하는데 이용될 수 있다. 이렇게 부드럽게 하는 것은 뷰들 간을 보간하는 것일 수 있다. 그 외에도, 사용자는 이 시스템(2700)을 이용하여 오브젝트 또는 장면을 회전시키거나 오브젝트 또는 장면의 3차원 표현을 볼 수 있다. 오브젝트의 회전은 예컨대 뷰들간에 이동하고 뷰들 간을 보간하여 뷰들 간에 부드러운 전이를 얻거나 단순히 3차원 표현을 얻는 것일 수 있다. 즉, 사용자는 보간된 뷰를 디스플레이될 "뷰"로서 선택할 수 있다.



- [0366] 여기서 설명된 바와 같이, 본 출원에서 설명된 구현과 특징들은 비디오 코딩, 깊이 코딩 및/또는 다른 타입의 데이터 코딩과 관련하여 이용될 수 있다. 그 외에도, 이들 구현과 특징은 H.264/MPEG-4 AVC(AVC) 표준, MVC 확장을 가진 AVC 표준, SVC 확장을 가진 AVC 표준, 3DV 표준 및/또는 (기존의 또는 미래의) 다른 표준과 관련하여, 또는 이들과 관련한 이용에 맞도록 구성되거나, 표준과 관계없는 상황에서 이용될 수 있다. 따라서, AVC에 따라서 작동하는 본 출원에서 설명된 특정 구현은 AVC에 한정되는 것이 아니며 AVC 밖의 이용에 적합하도록 구성될 수 있음을 알아야 한다.
- [0367] 또한 전술한 바와 같이, 구현들은 SEI 메시지, 슬라이스 헤더, 다른 고레벨 구문, 비고레벨(non-high-level) 구문, 대역외 정보, 데이터스트림 데이터, 및 함축적 시그널링을 포함하는(이에 한정되는 것은 아님) 다양한 기법을 이용하여 정보를 표시할 수 있다. 여기서 설명된 구현은 특정 상황에서 설명될 수 있지만, 그러한 설명은 특징과 개념을 그와 같은 구현 또는 내용으로 한정하는 것으로 보아서는 않된다.
- [0368] 여러 가지 구현은 디코딩을 포함한다. 본 출원에서 사용된 "디코딩"은 디스플레이에 적합한 최종 출력을 생성하기 위해 예컨대 수신된 인코딩된 계열에 대해 수행된 프로세스의 전부 또는 일부를 포함할 수 있다. 그와 같은 프로세스는 예컨대 엔트로피 디코딩, 역변환, 역양자화, 차등 디코딩과 같이 디코더에 의해 통상적으로 수행되는 프로세스를 포함할 수 있다. 그와 같은 프로세스는 예컨대 타일링된(패킹된) 화상으로부터 화상을 추출하는 것, 이용할 업샘플링을 결정한 다음에 화상을 업샘플링하는 것, 그리고 화상을 다시 그 의도한 배향으로 플립핑하는 것과 같이 본 출원에서 설명된 여러 가지 구현의 디코더에 의해 수행되는 프로세스를 추가적으로 또는 대안적으로 포함할 수 있다.
- [0369] 본 출원에서 설명된 여러 가지 구현은 복수의 화상을 조합하여 단일 화상으로 만든다. 화상을 조합하는 프로세스는 예컨대 화상들 간의 관계에 기초하여 화상을 선택하는 것, 공간 인터리빙, 샘플링, 및 화상 배향 변경을 포함할 수 있다. 따라서, 이 조합 프로세스를 기술하는 정보는 예컨대 관계 정보, 공간 인터리빙 정보, 샘플링 정보 및 배향 정보(orientation information)를 포함할 수 있다.
- [0370] 공간 인터리빙은 여러 가지 구현을 기술하는데 언급된다. 본 출원에서 사용된 "인터리빙"은 예컨대 "타일링" 및 "패킹"이라고도 한다. 인터리빙은 예컨대 화상의 병렬 배치, 화상의 상하 배치, (예컨대 4개 화상을 조합하는) 병렬과 상하의 조합, 행단위 교호, 열단위 교호, 여러 가지 픽셀 레벨(픽셀 방식 또는 픽셀 기반이라 함) 방식을 포함하는 다양한 형태의 인터리빙을 포함한다.
- [0371] 여기서 사용된 용어 "화상"과 "이미지"는 상호 교환적으로 사용되며, 예컨대 스틸 이미지의 전부 또는 일부 또는 비디오 계열 중의 화상의 전부 또는 일부를 말할 수 있다. 잘 알고 있는 바와 같이, 화상은 프레임 또는 필드일 수 있다. 그 외에도, 여기서 사용된 화상은 예컨대 프레임의 상단 절반, 단일 매크로블록, 교호하는 열, 교호하는 행, 또는 주기적 픽셀과 같은 프레임의 서브세트일 수도 있다. 다른 예로서, 깊이 화상은 예컨대 해당 비디오 프레임의 단일 매크로블록에 대한 깊이 정보만을 포함하는 완전 깊이 맵 또는 부분 깊이 맵일 수 있다.
- [0372] 본 명세서에서 본 발명의 "일 실시예"나 "소정 실시예" 또는 "일 구현"이나 "소정 구현"이라는 말과 그 변형은 그 실시예와 관련하여 설명된 특정 형상, 구조, 특성 등이 본 발명의 적어도 하나의 실시예에 포함된다는 것을 의미한다. 따라서 본 명세서의 여러 곳에서 나타내는 "일 실시예에서"나 "소정 실시예에서" 또는 "일 구현에서"나 "소정 구현에서"라는 구절과 그 변형은 반드시 같은 실시예를 말하는 것은 아니다.
- [0373] 예컨대 "A/B", "A 및/또는 B" 및 "A와 B 중 적어도 어느 하나"의 경우에서 "/", "및/또는" 및 "적어도 어느 하나"를 사용하는 것은 첫번째 옵션 (A)만, 두번째 옵션 (B)만, 양 옵션 (A 및 B)를 선택하는 것을 포함하는 것임을 알아야 한다. 추가 예로서, "A, B 및/또는 C" 및 "A, B 및 C 중 적어도 어느 하나" 및 "A, B 또는 C 중 적어도 어느 하나"의 경우에 이러한 구절은 첫번째 옵션 (A)만, 두번째 옵션 (B)만, 세번째 옵션 (C)만, 첫번째 옵션 (A)와 두번째 옵션 (B)만, 첫번째 옵션 (A)와 세번째 옵션 (C)만, 두번째 옵션 (B)와 세번째 옵션 (C)만, 세가지 옵션 모두(A, B 및 C)를 선택하는 것을 포함하는 것이다. 이것은 당업자라면 쉽게 알 수 있는 바와 같이 더 많은 옵션에 대해서도 확장될 수 있다.
- [0374] 그 외에도, 본 출원 또는 그 청구범위는 각종 정보를 "결정"하는 것을 말할 수 있다. 정보를 결정하는 것은 예컨대 정보를 추정하는 것, 정보를 계산하는 것, 정보를 예측하는 것 또는 정보를 메모리로부터 검색하는 것 중 하나 이상을 포함할 수 있다.
- [0375] 유사하게, "액세스하는 것"은 광의의 용어로 쓰인다. 정보에 액세스하는 것은 예컨대 정보를 이용, 저장, 송출, 송신, 수신, 검색, 변경 또는 제공하는 동작을 포함할 수 있다.

- [0376] 도면에서 도시된 구성요소들은 여러 가지 형태의 하드웨어, 소프트웨어, 또는 이들의 조합으로 구현될 수 있음을 알아야 한다. 바람직하게는 이들 구성요소는 프로세서, 메모리 및 입/출력 인터페이스를 포함할 수 있는 하나 이상의 적당히 프로그램된 범용 장치 상에서 하드웨어 및 소프트웨어의 조합으로 구현된다. 더욱이 여기서 설명되는 구현은 예컨대 방법 또는 프로세스, 장치 또는 소프트웨어 프로그램으로서 구현될 수 있다. 단일 형태의 구현과 관련하여 설명되더라도(예컨대 방법만으로 설명되더라도), 설명된 특징의 구현은 다른 형태(예컨대 장치나 프로그램)로도 구현될 수 있다. 장치는 전술한 바와 같이 구현될 수 있다. 방법은 예컨대 컴퓨터, 마이크로프로세서, 집적 회로 또는 프로그래머블 로직 장치를 포함하는 일반적으로 처리 장치를 말하는 프로세서와 같은 장치에서 구현될 수 있다. 처리 장치도 예컨대 컴퓨터, 셀폰, 휴대형/개인 정보 단말("PDA"), 기타 최종 사용자 간의 정보 통신을 가능하게 하는 다른 장치와 같은 통신 장치를 포함한다.
- [0377] 여기서 설명된 각종 프로세스와 특징의 구현은 특히 예컨대 데이터 인코딩 및 디코딩과 연관된 장비 또는 애플리케이션과 같은 여러 가지 장비나 애플리케이션에서 구체화될 수 있다. 장비의 예로는 비디오 코더, 비디오 디코더, 비디오 코덱, 웹 서버, 셋톱 박스, 랩톱 컴퓨터, 개인용 컴퓨터, 셀폰, PDA, 기타 다른 통신 장치, 개인 녹화 장치(예컨대 PVR, 녹화 소프트웨어 실행 컴퓨터, VHS 녹화 장치), 캠코더, 인터넷 또는 다른 통신 링크를 통한 데이터 스트리밍 및 비디오 온 디맨드가 있다. 명백히 장비는 모바일이며 차량에 설치될 수도 있다.
- [0378] 그 외에도, 방법은 프로세서에 의해 수행되는 명령어로 구현될 수 있으며, 그와 같은 명령어는 예컨대 집적 회로, 소프트웨어 캐리어, 또는 예컨대 하드 디스크, 콤팩트 디스켓, RAM(Random Access Memory) 또는 ROM(Read-Only Memory)과 같은 다른 저장 장치와 같은 프로세서 판독가능 매체에 저장될 수 있다. 명령어는 프로세서 판독가능 매체에 유형적으로 구체화되는 애플리케이션 프로그램을 구성할 수 있다. 명백하게 프로세서는 예컨대 프로세스를 실행하기 위한 명령어를 가진 프로세서 판독가능 매체를 포함할 수 있다. 그와 같은 애플리케이션 프로그램은 임의의 적당한 구조를 가진 기계로 업로드되어 실행될 수 있다. 바람직하게는, 기계는 하나 이상의 CPU(central processing unit), RAM(random access memory) 및 I/O(input/output) 인터페이스와 같은 하드웨어를 가진 컴퓨터 플랫폼 상에서 구현된다. 컴퓨터 플랫폼은 운영 체제와 마이크로명령어 코드도 포함할 수 있다. 여기서 설명된 여러 가지 프로세스와 기능들은 CPU에 의해 실행될 수 있는 마이크로명령어 코드의 일부, 애플리케이션 프로그램의 일부, 또는 이들의 조합일 수 있다. 게다가, 컴퓨터 플랫폼에는 부가 데이터 저장 장치나 프린트 장치와 같은 각종 다른 주변장치가 연결될 수 있다.
- [0379] 당업자라면 잘 알겠지만, 구현은 예컨대 저장 또는 전송될 수 있는 정보를 갖도록 포맷팅된 신호를 생성할 수도 있다. 이러한 정보는 예컨대 방법을 실행하기 위한 명령어, 또는 전술한 구현들 중 하나에 의해 생성된 데이터를 포함할 수 있다. 그와 같은 신호는 예컨대 (예컨대 스펙트럼의 무선 주파수 부분을 이용하는) 전자기파로서 또는 기저대역 신호로서 포맷팅될 수 있다. 이러한 포맷팅은 예컨대 데이터 스트림 인코딩, 구문 생성, 및 인코딩된 데이터 스트림 및 구문을 이용한 반송파 변조를 포함할 수 있다. 신호가 갖고 있는 정보는 예컨대 analog 로그 또는 디지털 정보일 수 있다. 이 신호는 공지의 다양한 유선 또는 무선 링크를 통해 전송될 수 있다.
- [0380] 더욱이 첨부도면에 나타난 시스템 구성성분과 방법들 중 일부는 바람직하게는 소프트웨어로 구현되기 때문에, 시스템 구성성분들 간 또는 프로세스 기능 블록들 간의 실제 연결은 본 발명이 프로그램되는 방식에 따라 다를 수 있다. 당업자는 여기서의 교시를 고려하여 본 발명의 이들 및 이와 유사한 구현이나 구성을 생각해낼 수 있을 것이다.
- [0381] 많은 구현에 대해서 설명하였다. 그럼에도 불구하고 여러 가지 변형이 가능함을 알아야 한다. 예컨대 여러 가지 구현의 구성요소들은 조합, 보충, 변형 또는 제거되어 다른 구현을 만들어 낼 수 있다. 그 외에도, 당업자라면 다른 구조와 프로세스가 개시된 구조와 프로세스를 대체할 수 있으며 이렇게 해서 만들어진 구현은 적어도 실질적으로 동일한 방식(들)으로 적어도 실질적으로 동일한 기능(들)을 수행하여 개시된 구현들과 적어도 실질적으로 동일한 결과(들)를 얻을 것이라는 것을 잘 알 것이다. 특히 지금까지 첨부도면을 참조로 예시적인 실시예들에 대해 설명하였지만, 본 발명은 이들 실시예들에만 한정되는 것이 아니며, 당업자라면 본 발명의 범위와 본질로부터 벗어남이 없이 여러 가지로 수정 및 변형을 가할 수 있음은 물론이다. 따라서 이들 구현 및 다른 구현도 본 출원에 의해 고려되는 것이며 하기 청구범위의 기술적 범위 내에 있는 것이다.

도면

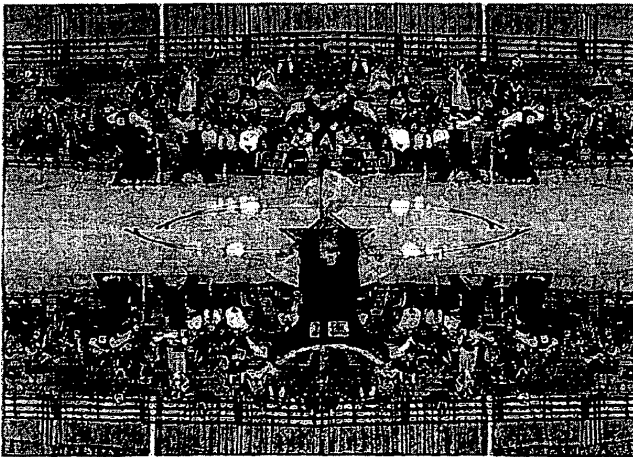
도면1

100

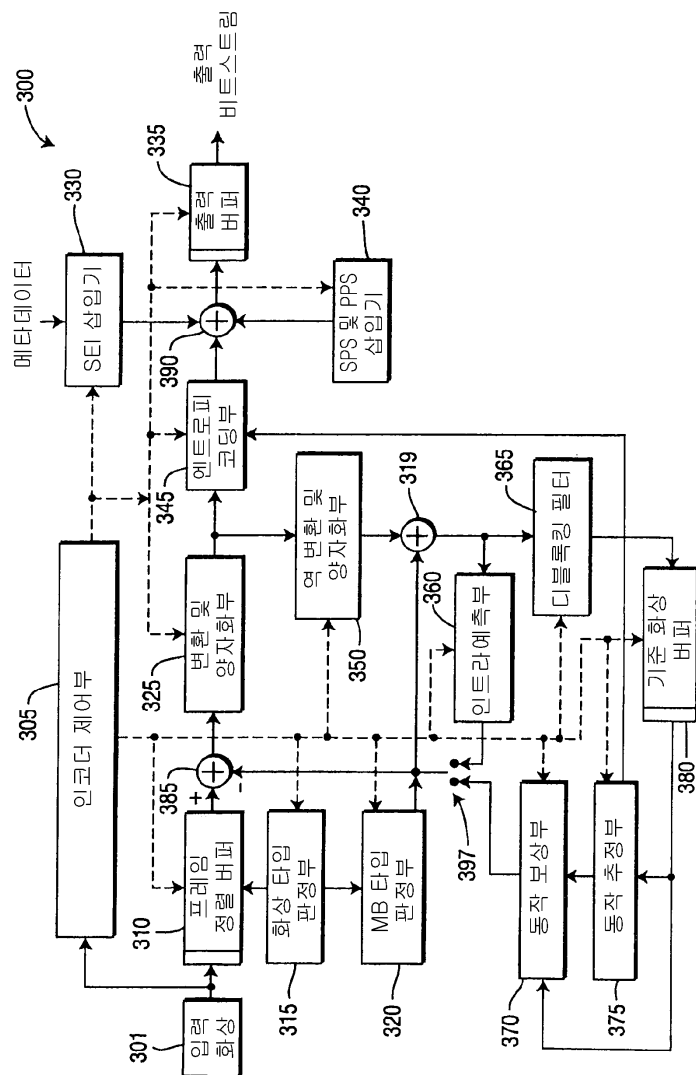


도면2

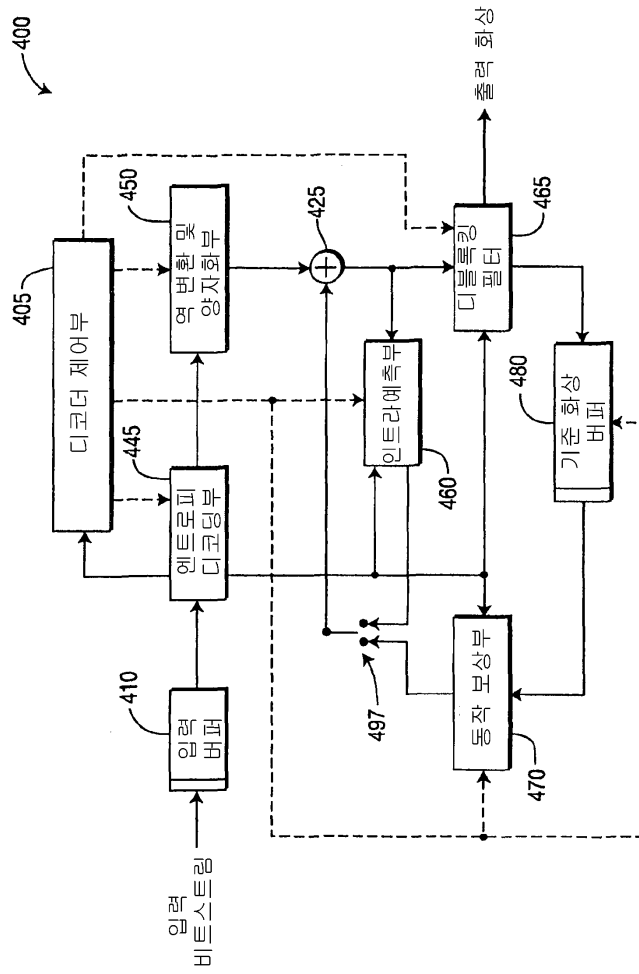
200



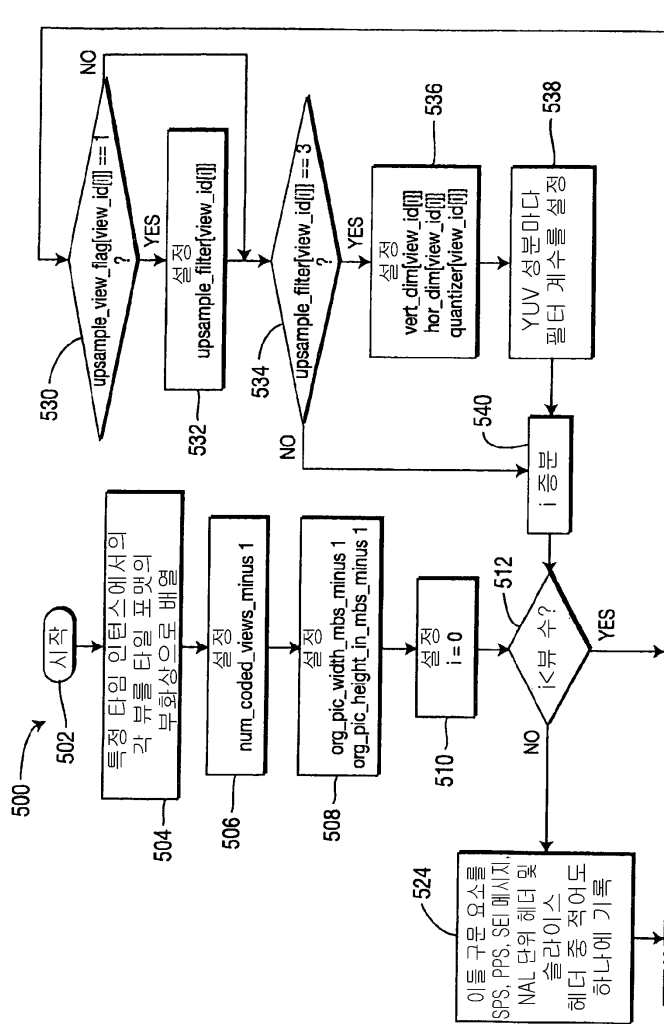
도면3



도면4

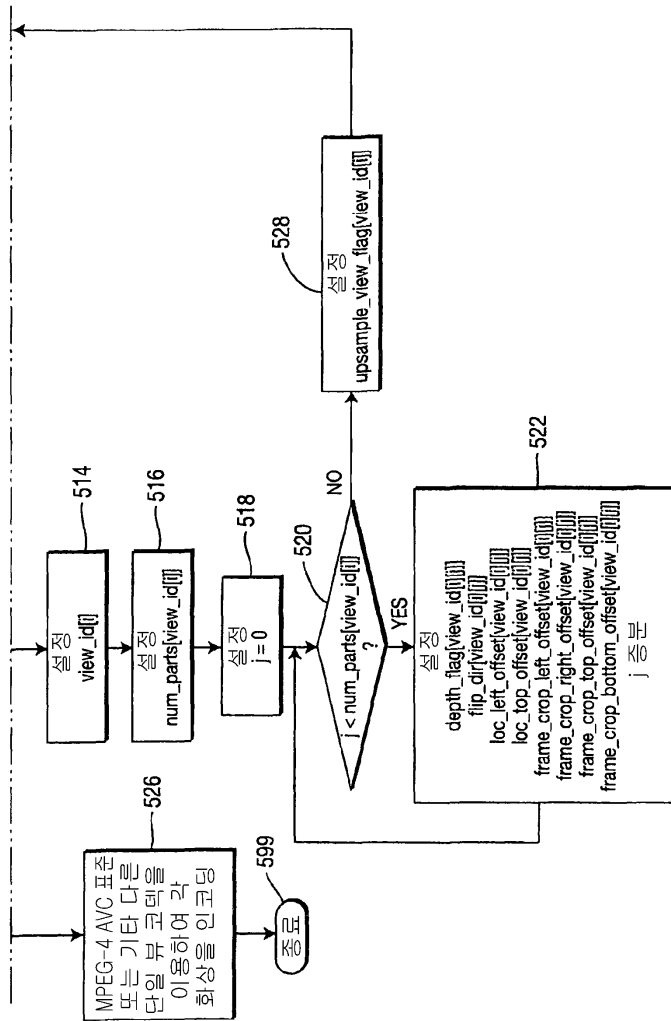


도면5a

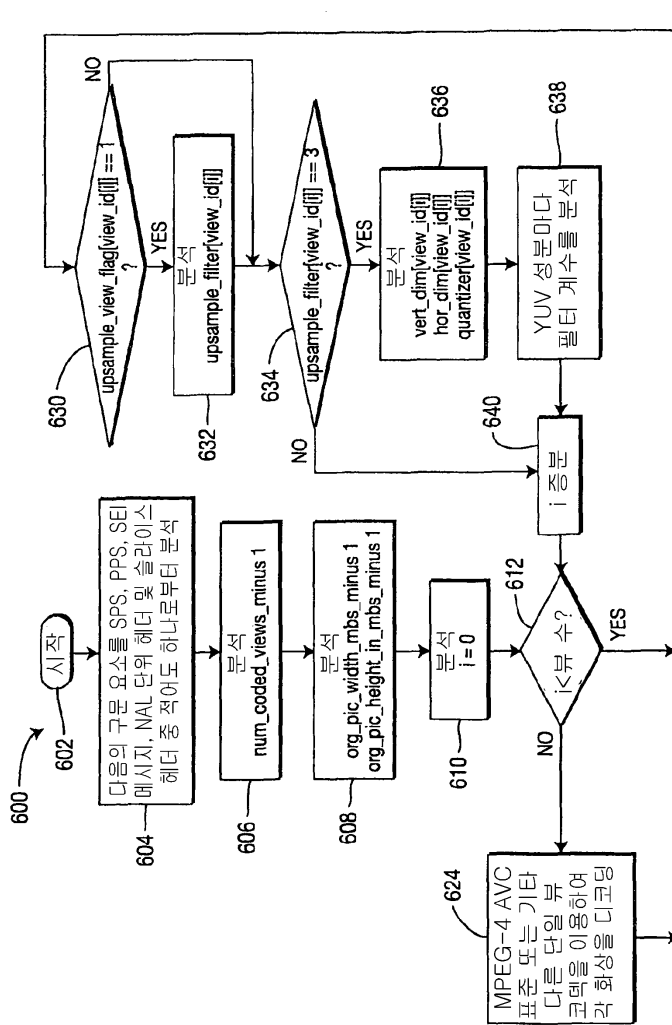




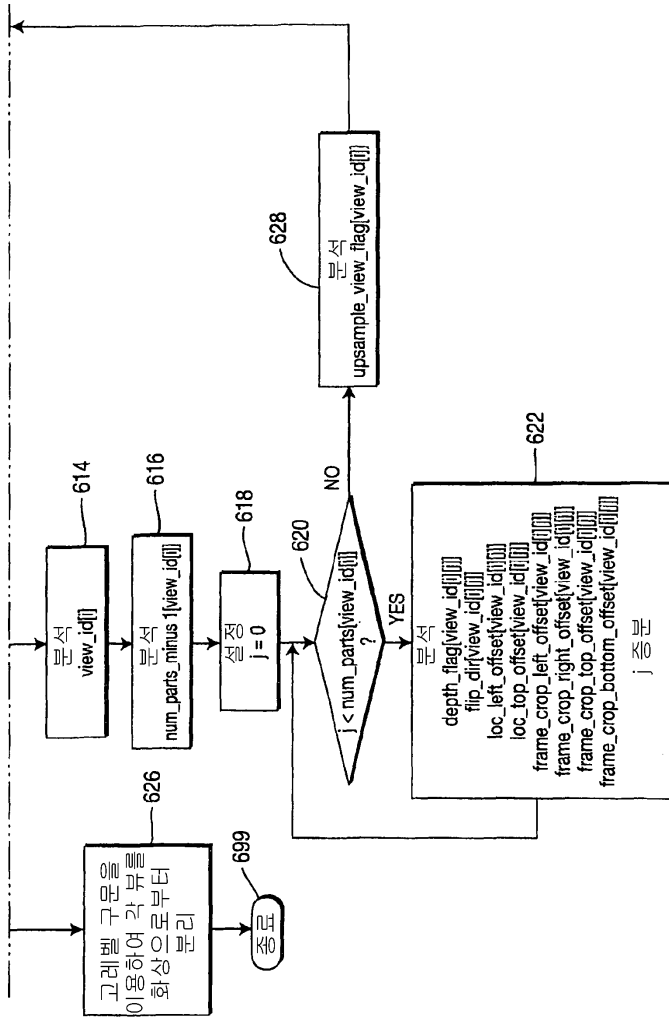
도면5b



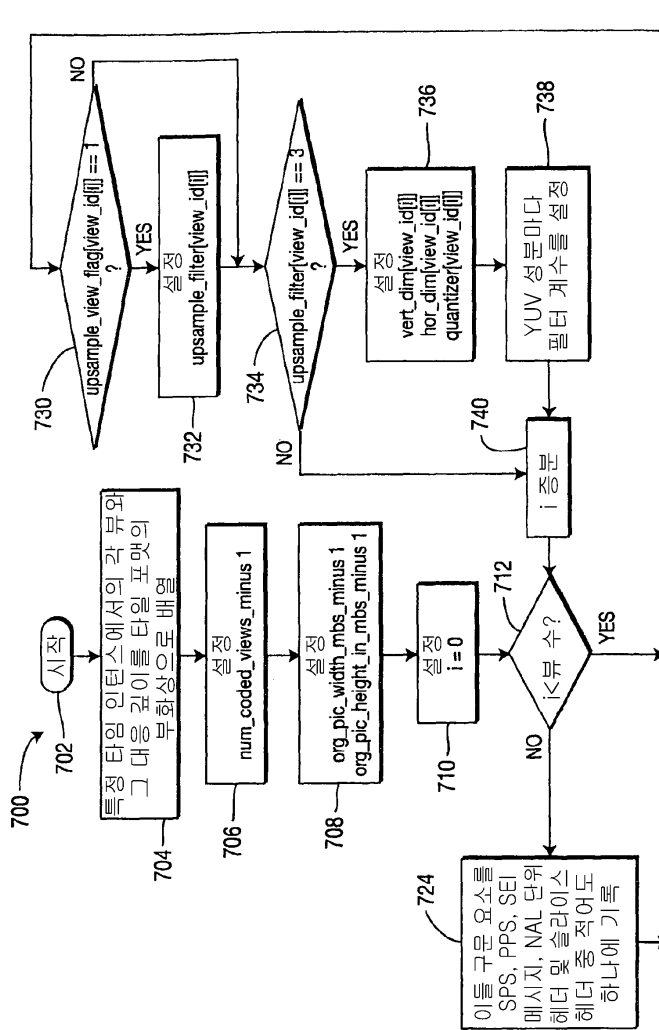
도면6a



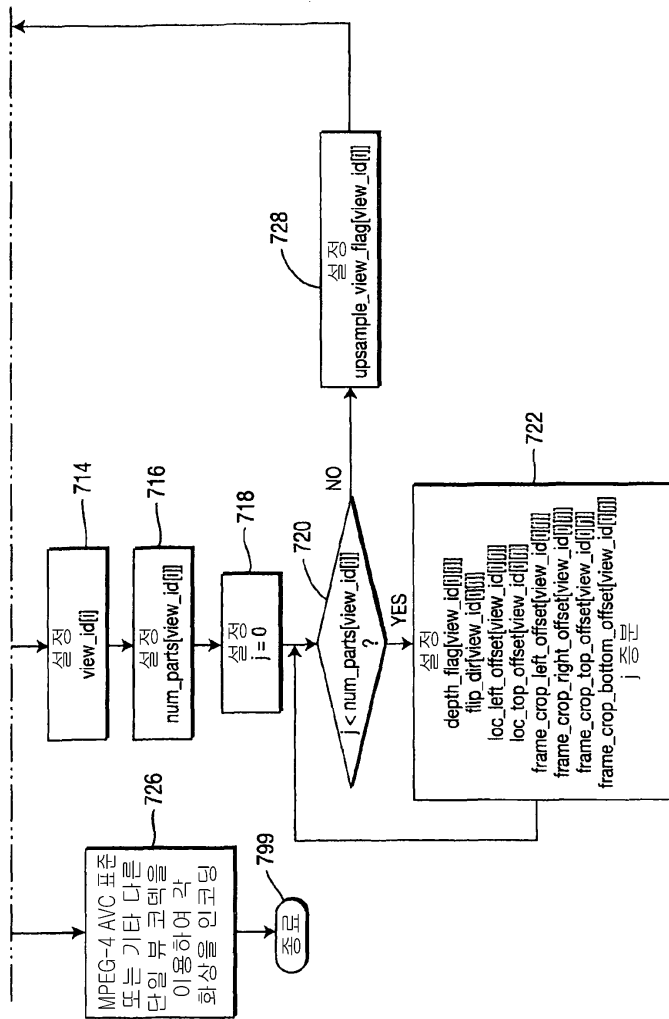
도면6b



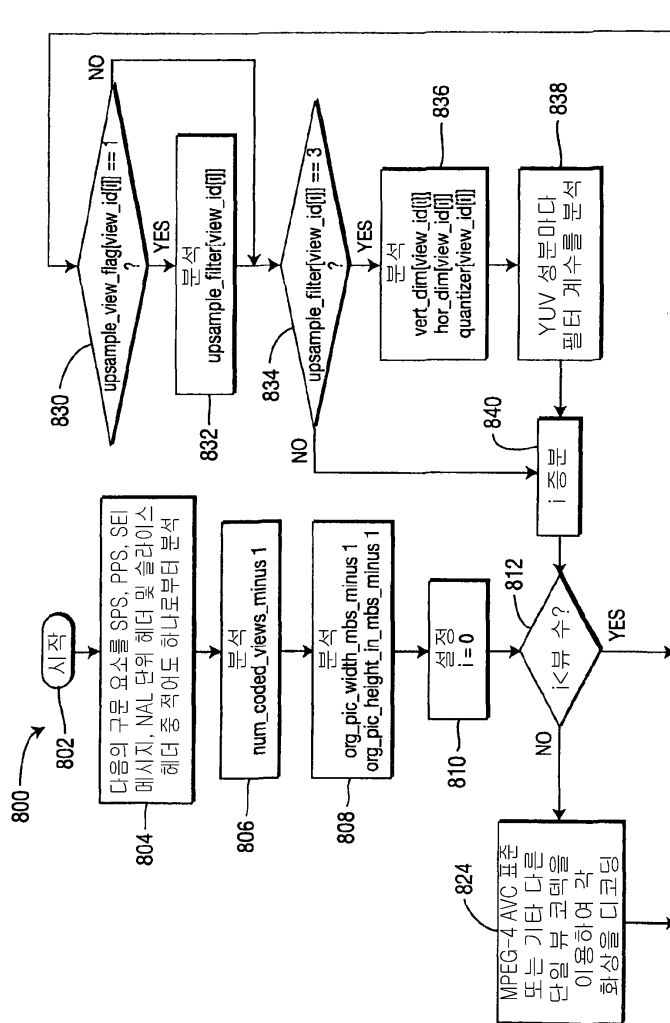
도면 7a



도면7b

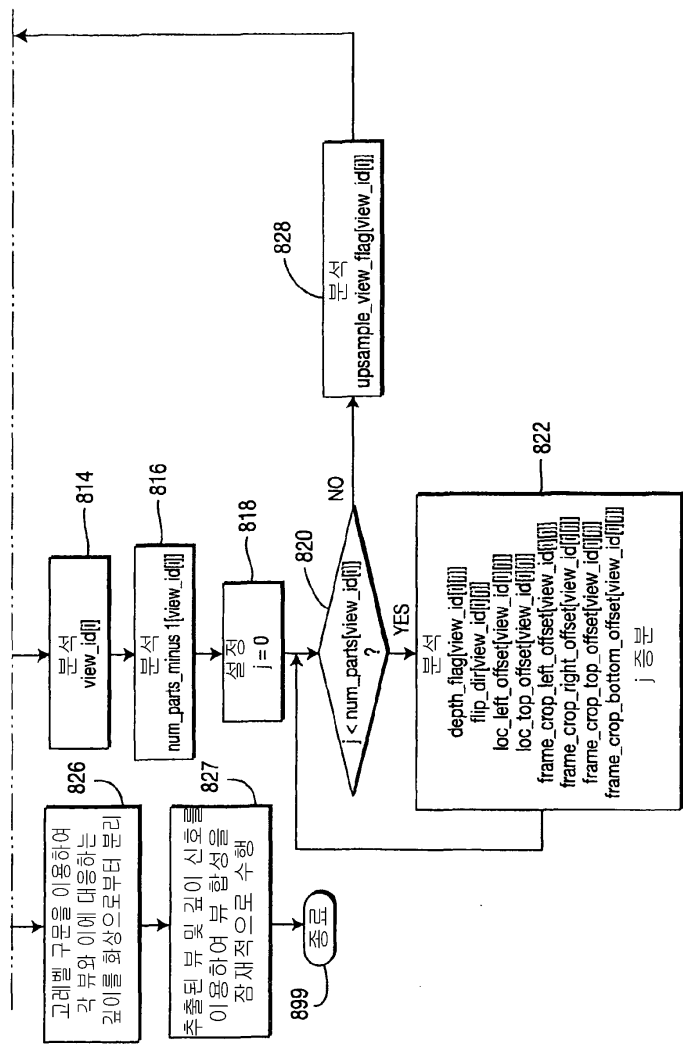


도면 8a





도면8b



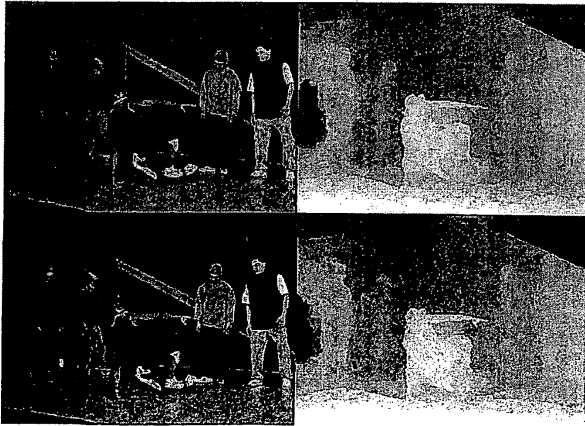
도면9



900

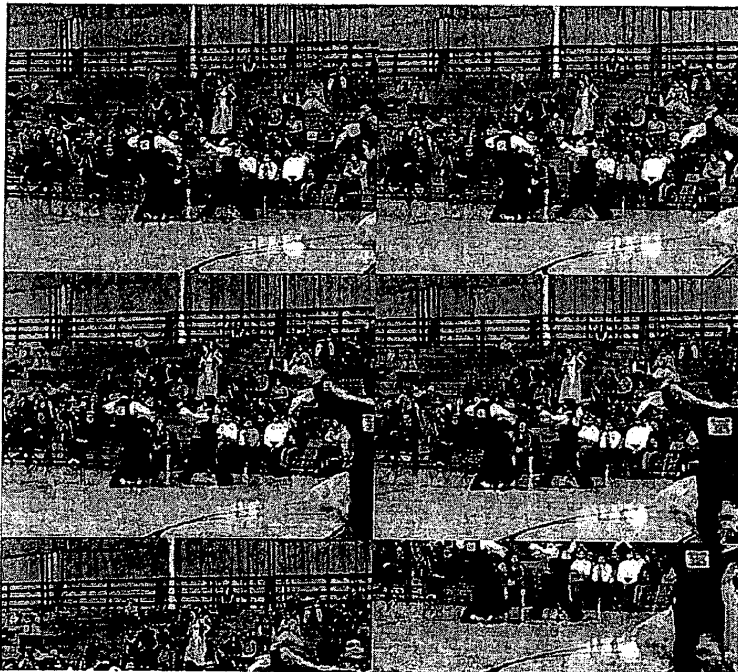
도면10

1000

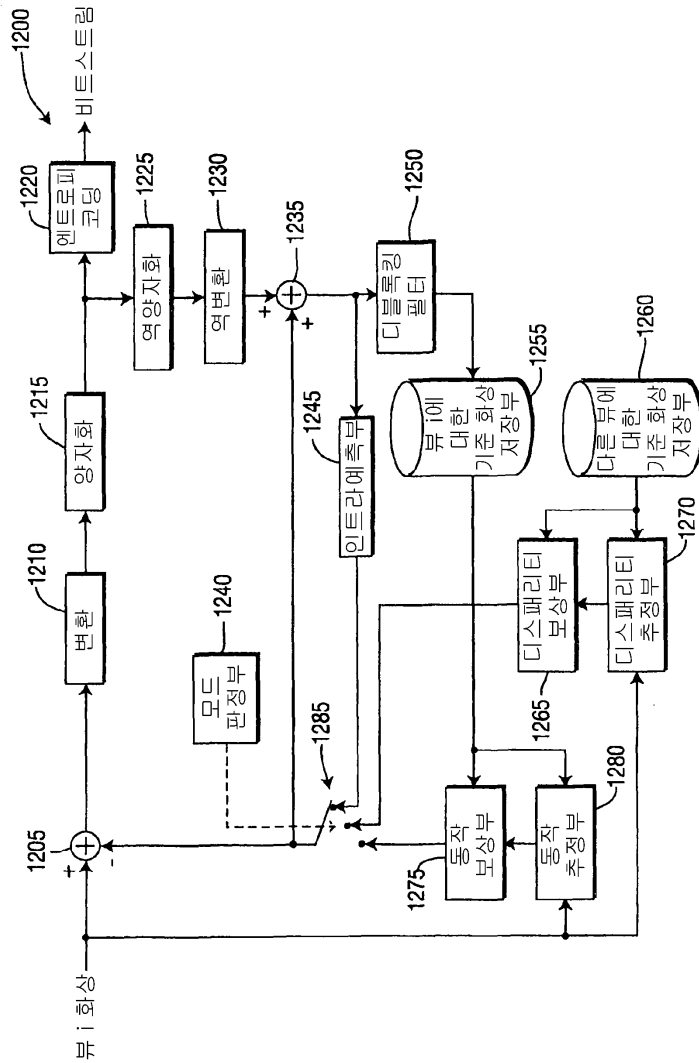


도면11

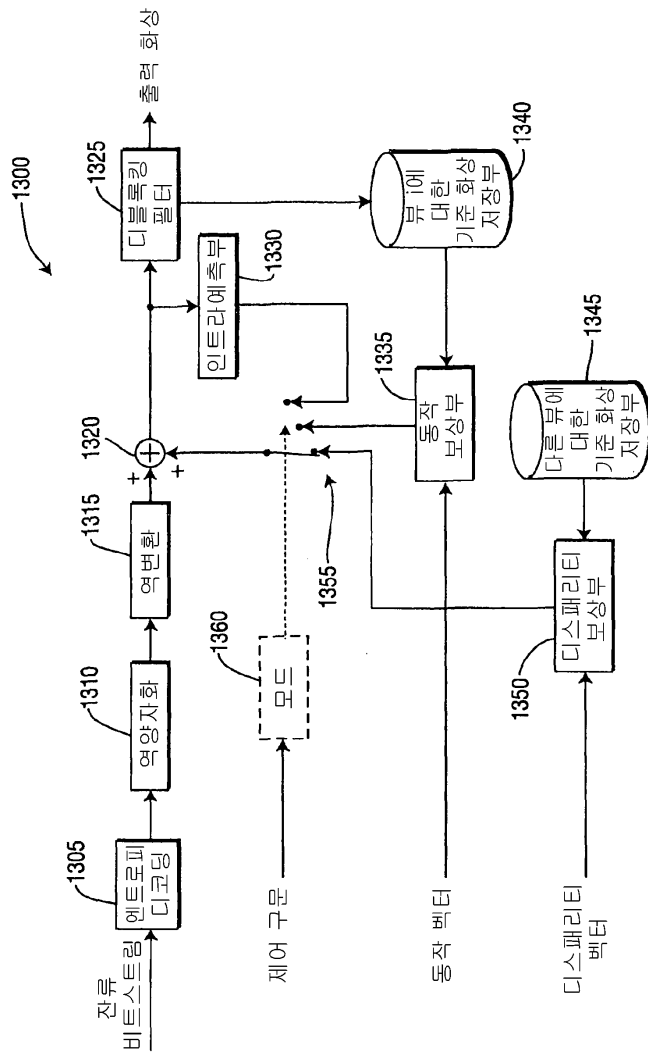
1100



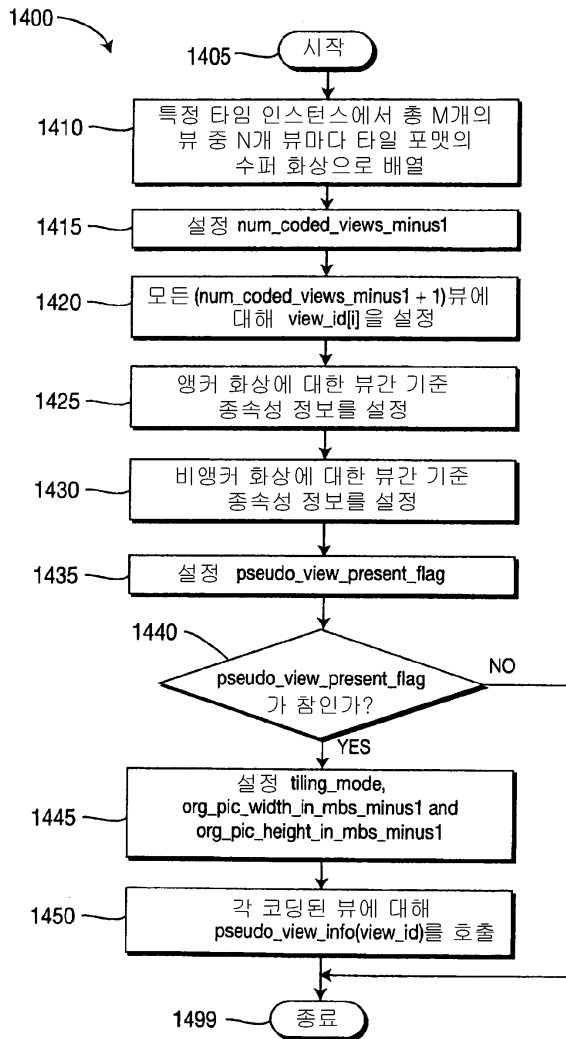
도면12



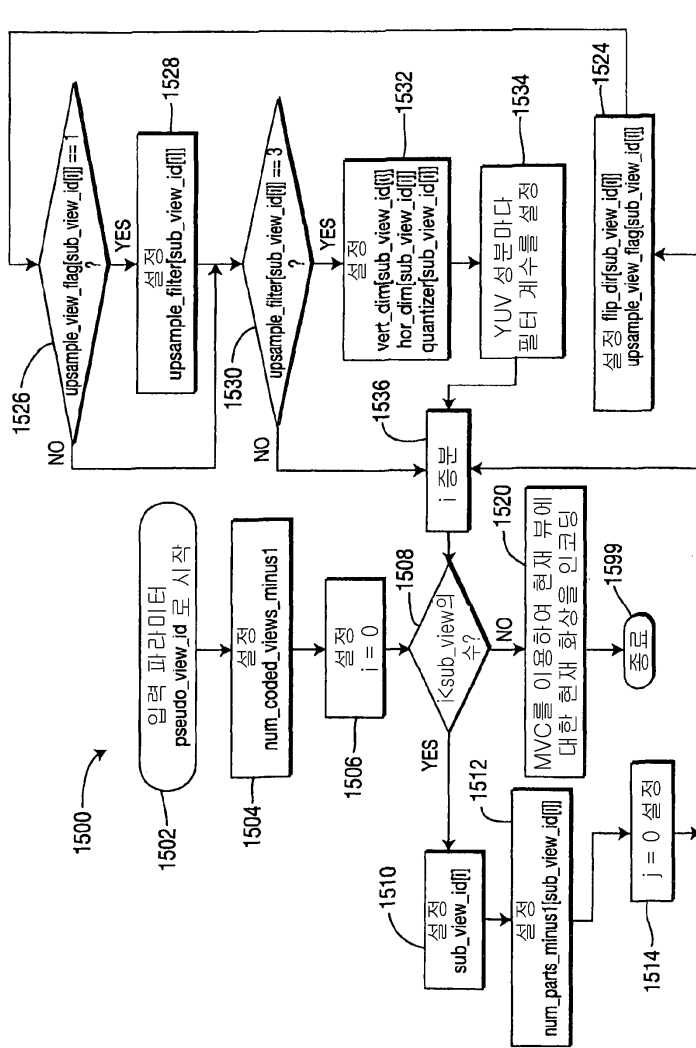
도면13



도면14

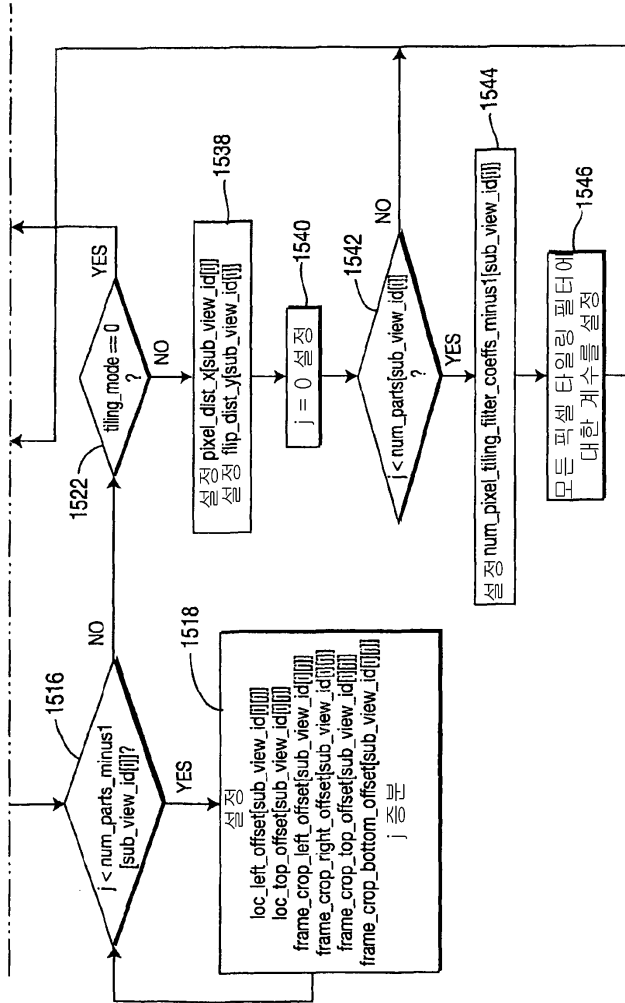


도면15a

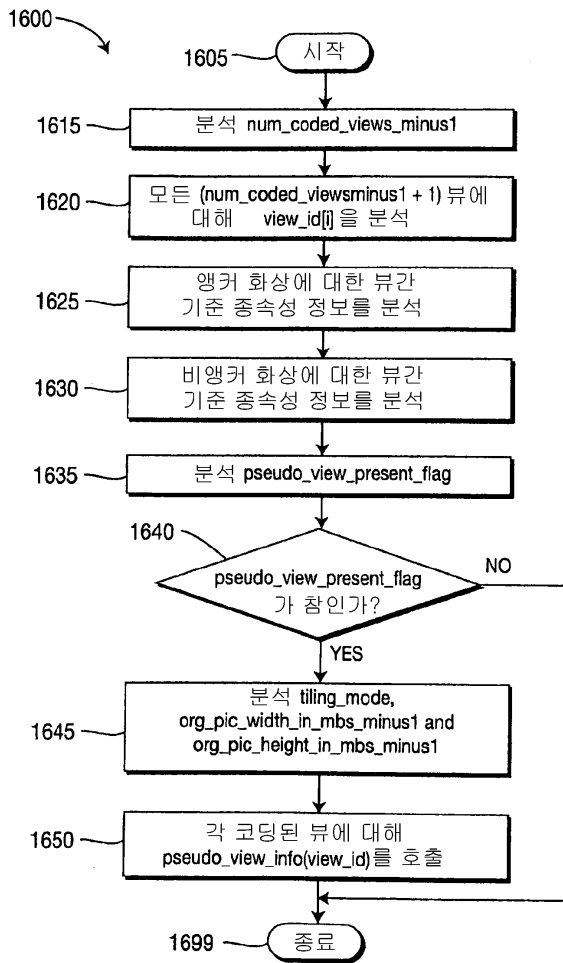




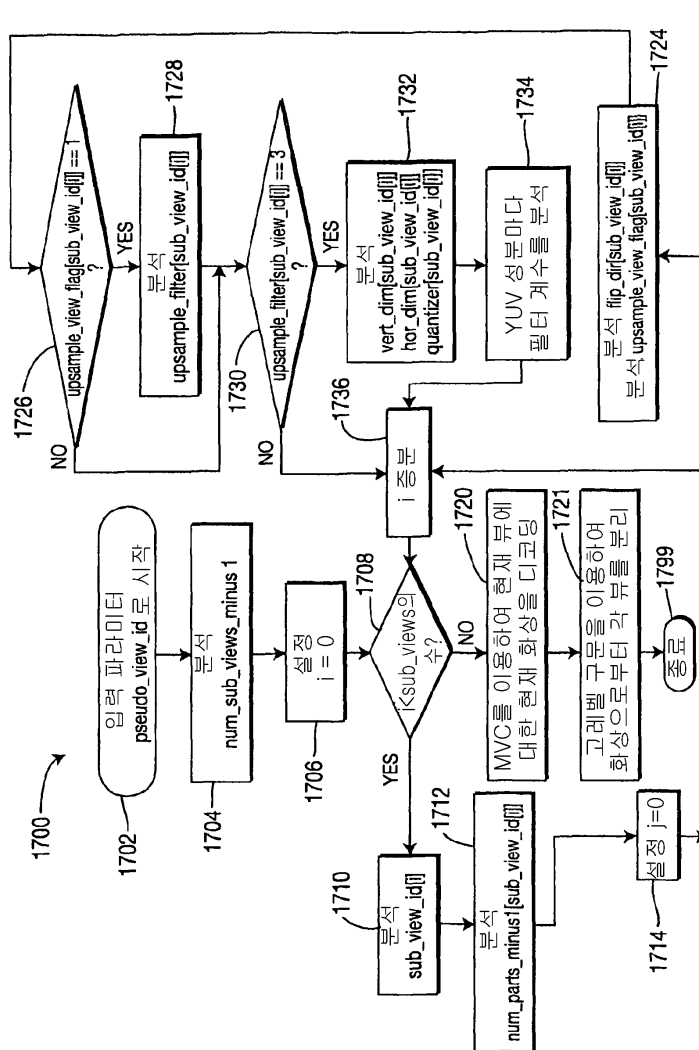
도면15b



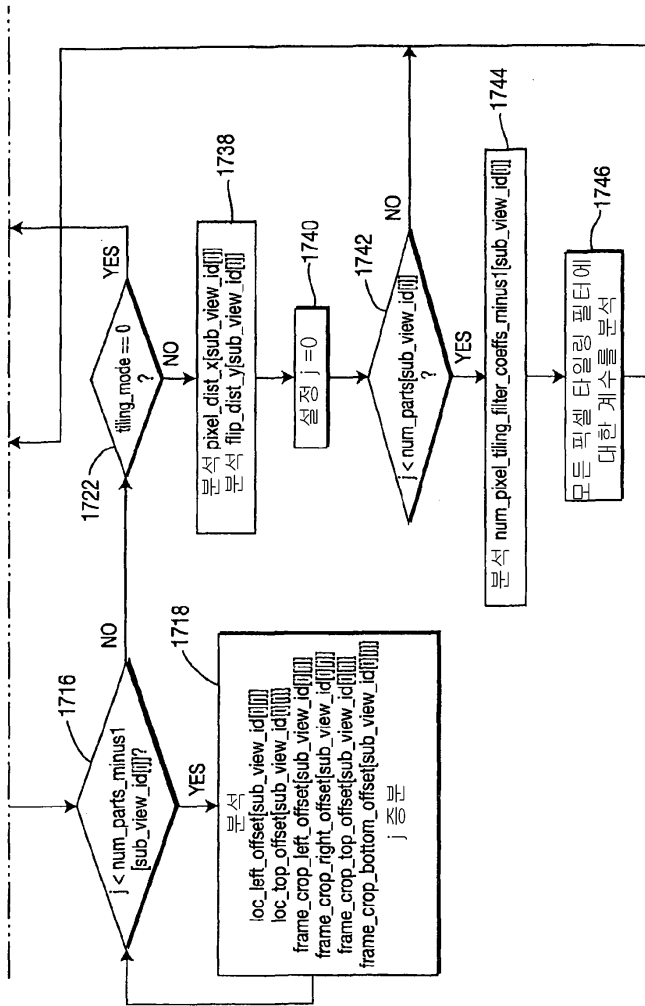
도면16



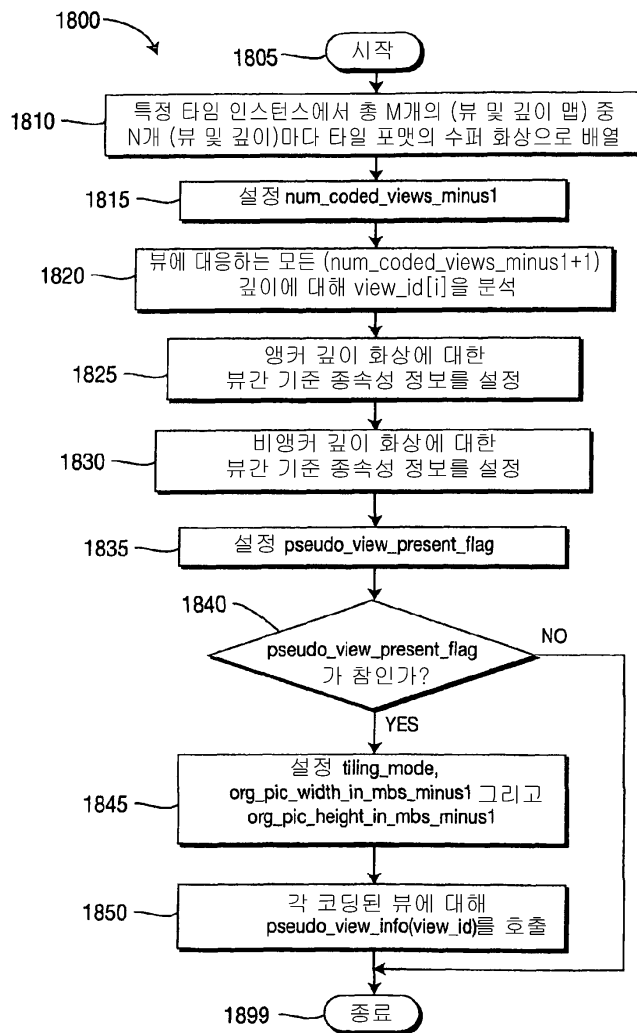
도면17a



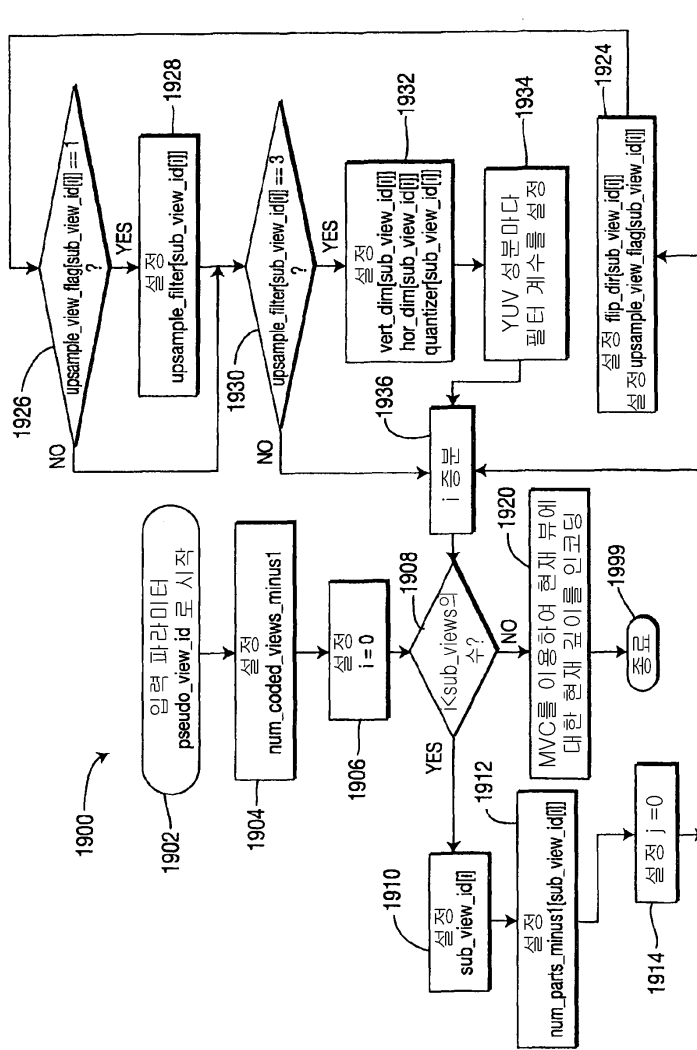
도면17b



도면18



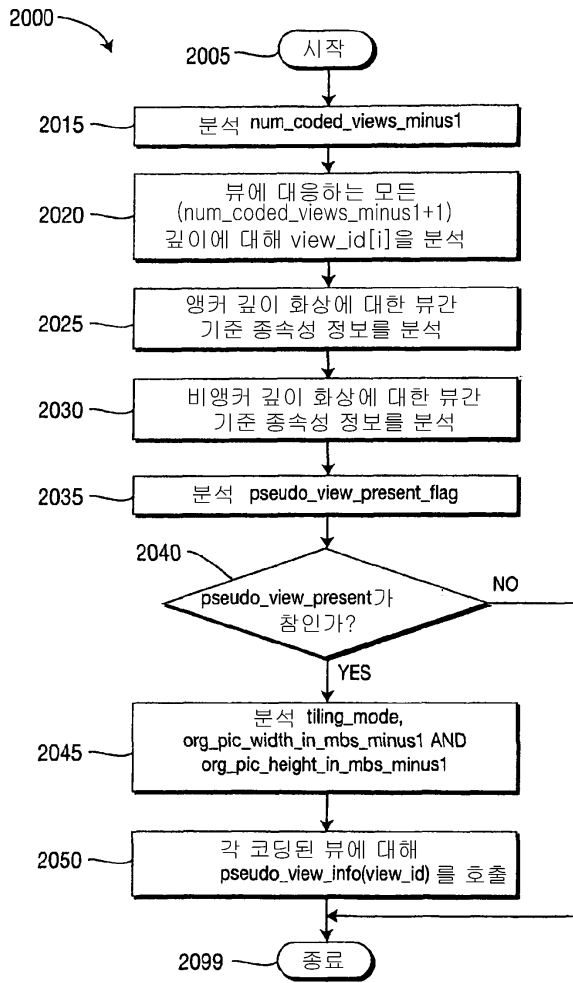
도면19a



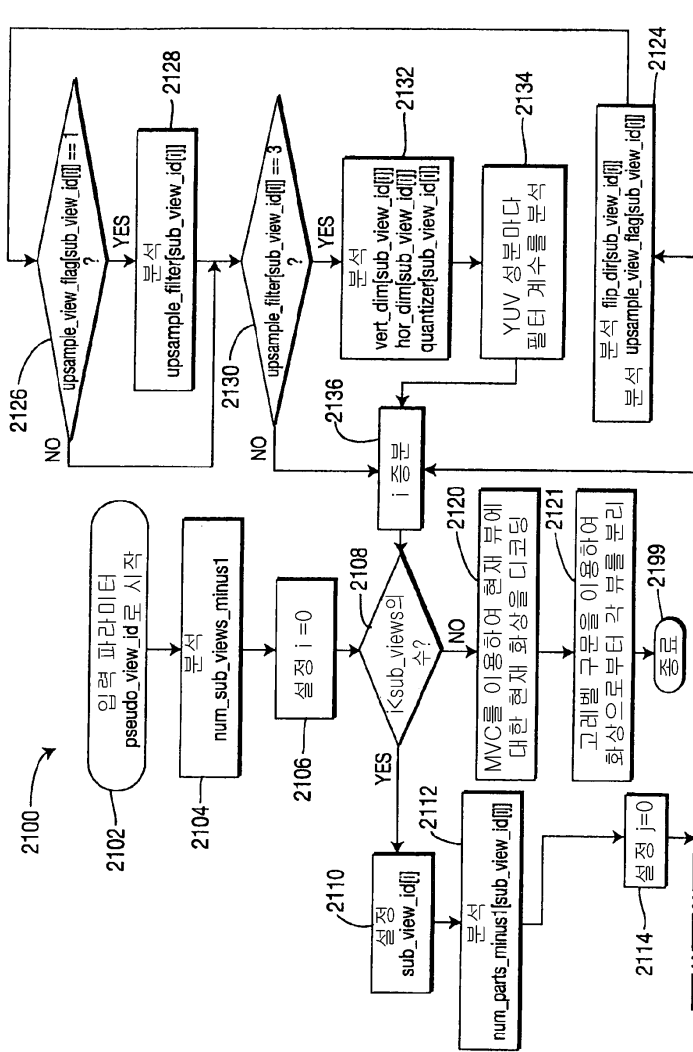




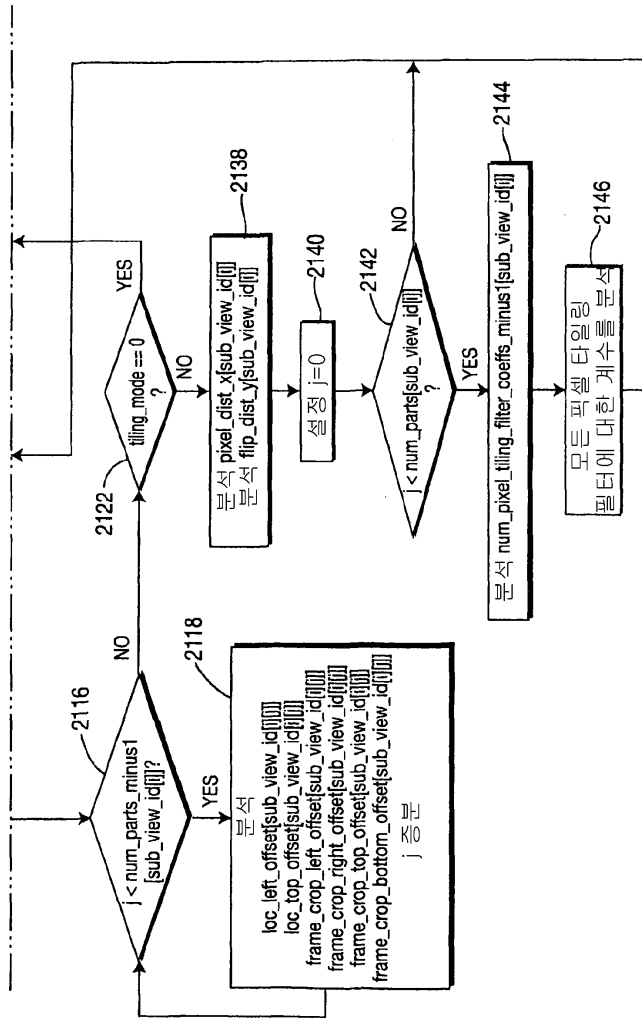
도면20



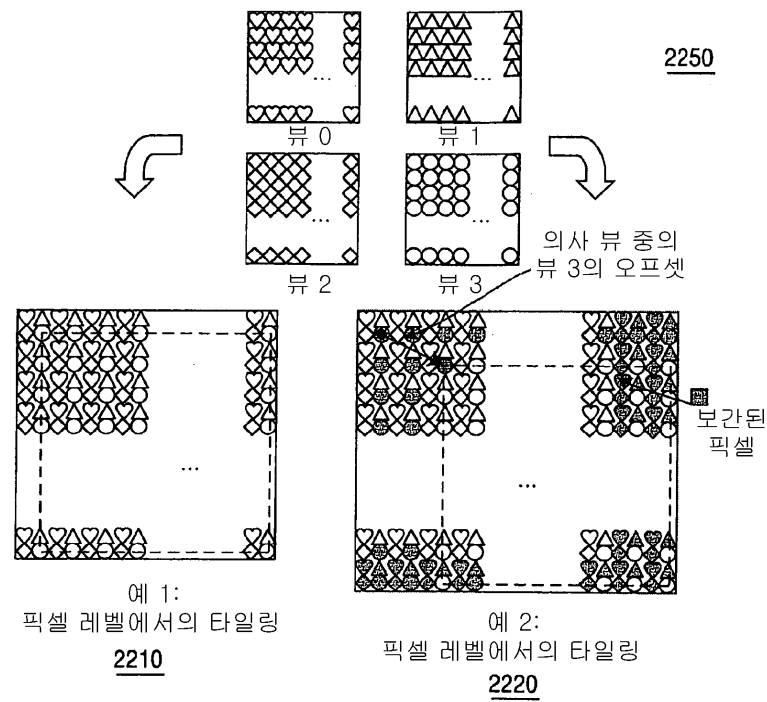
도면21a



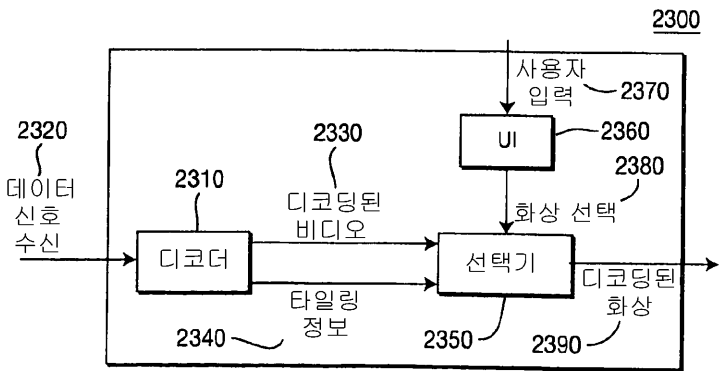
도면21b



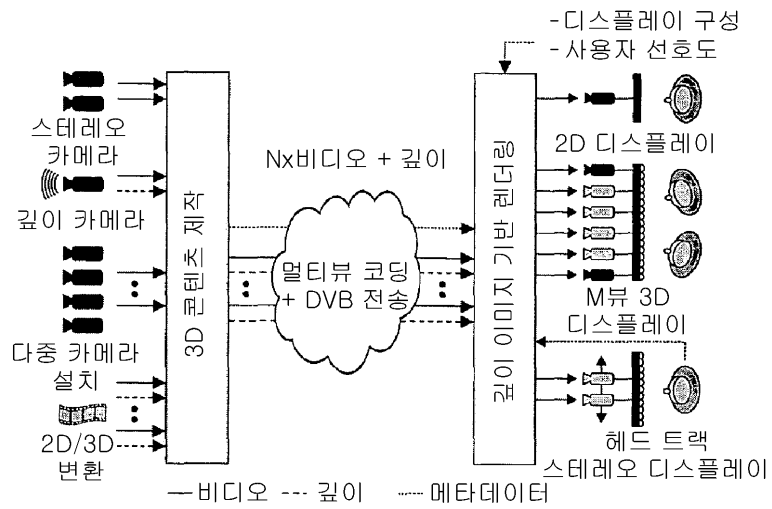
도면22



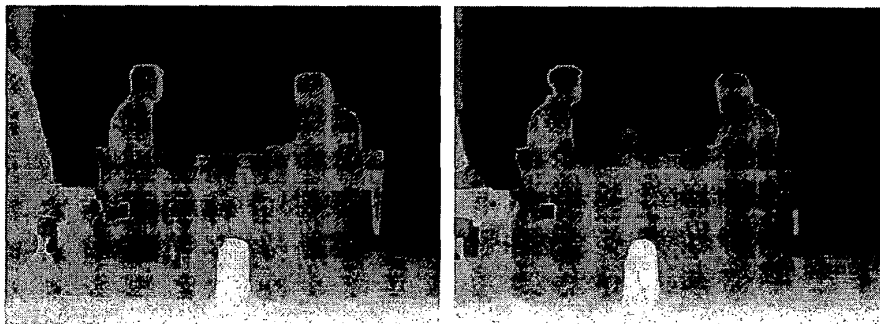
도면23



도면24



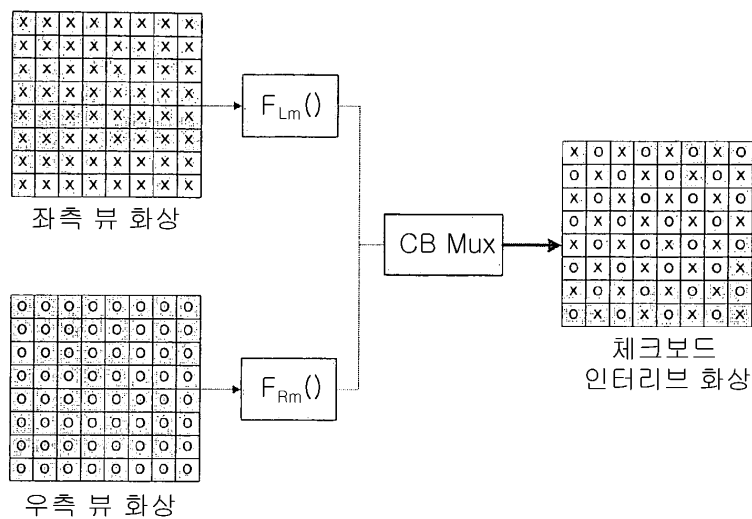
도면25



(a) : 좌측 깊이 맵

(b) : 우측 깊이 맵

도면26



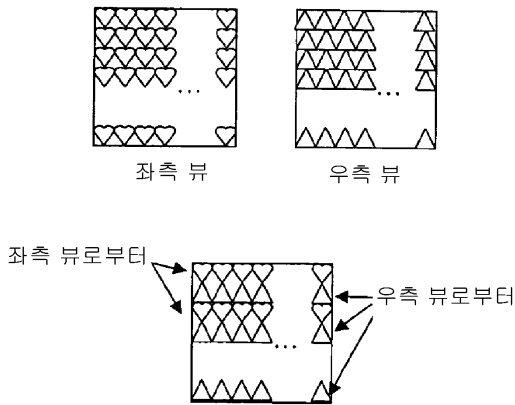
도면27



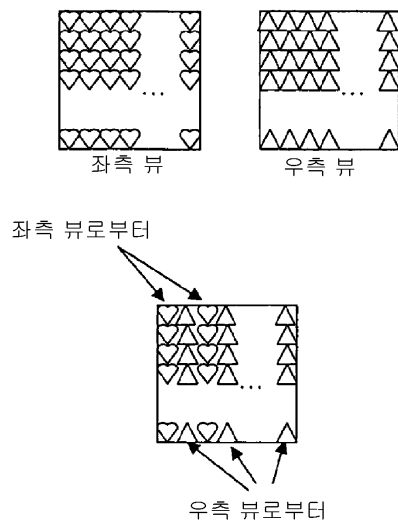
도면28



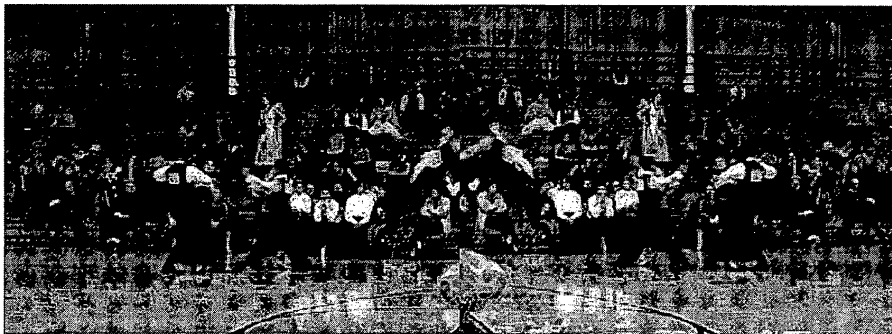
도면29



도면30

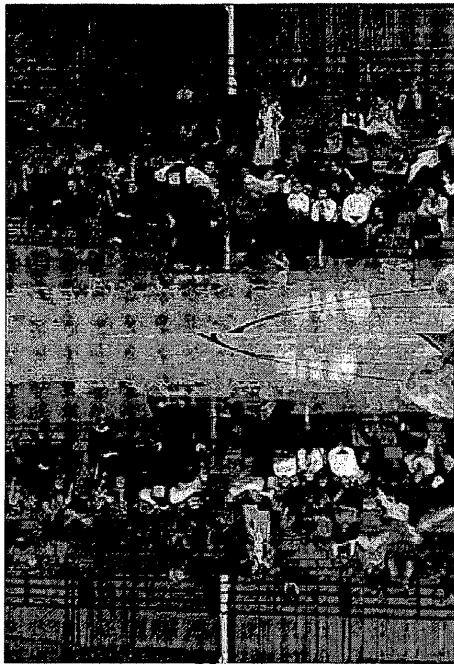


도면31

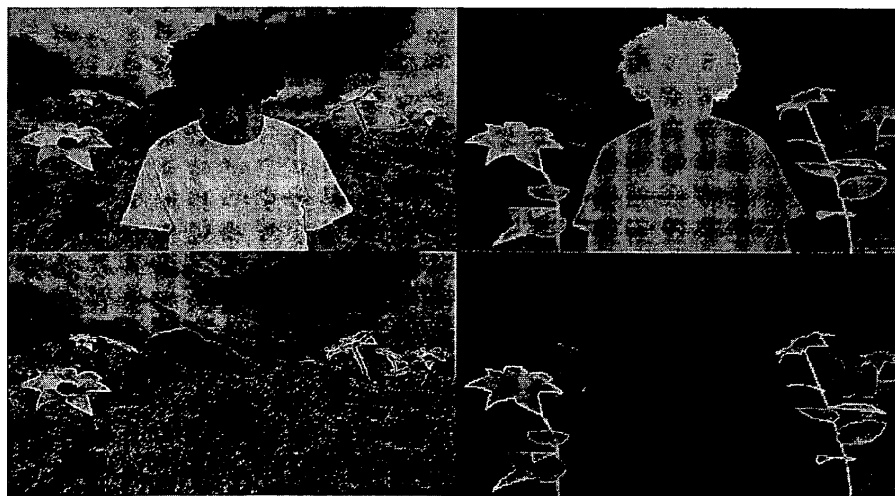




도면32



도면33

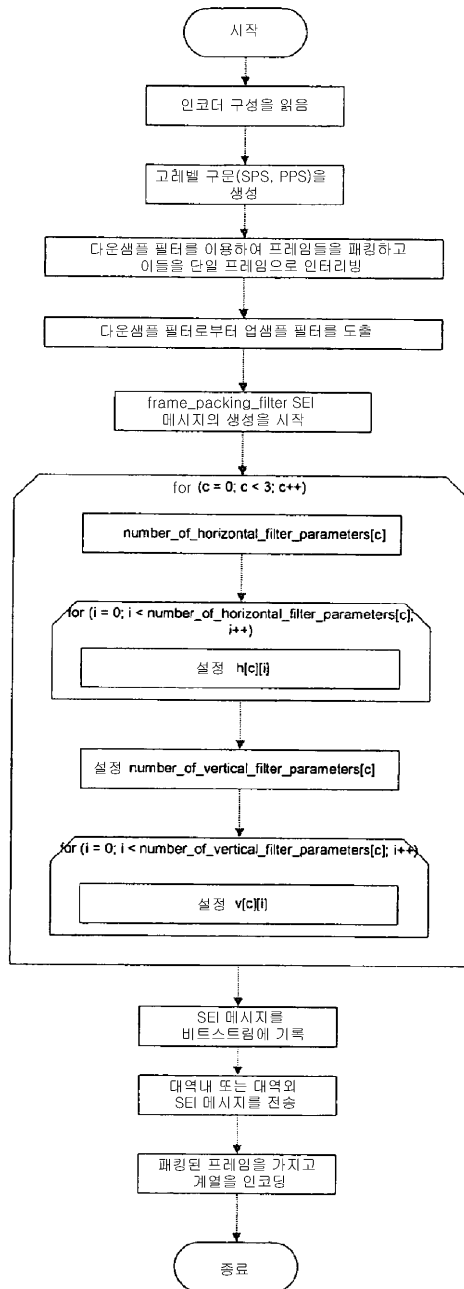


도면34



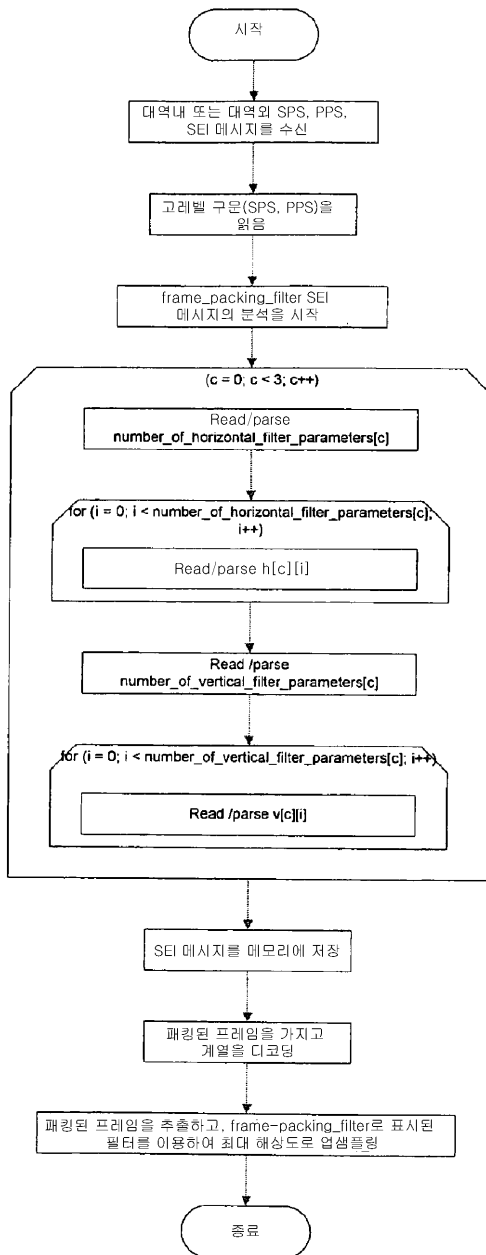
도면35

인코딩 플로우차트(실시예 1)



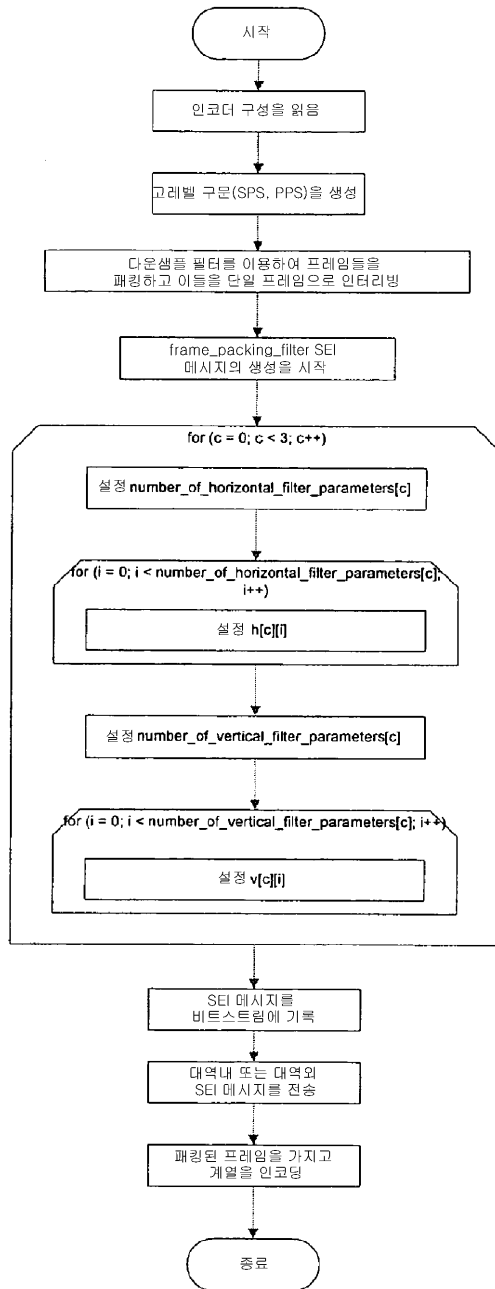
도면36

디코딩 플로우차트(실시예 1)



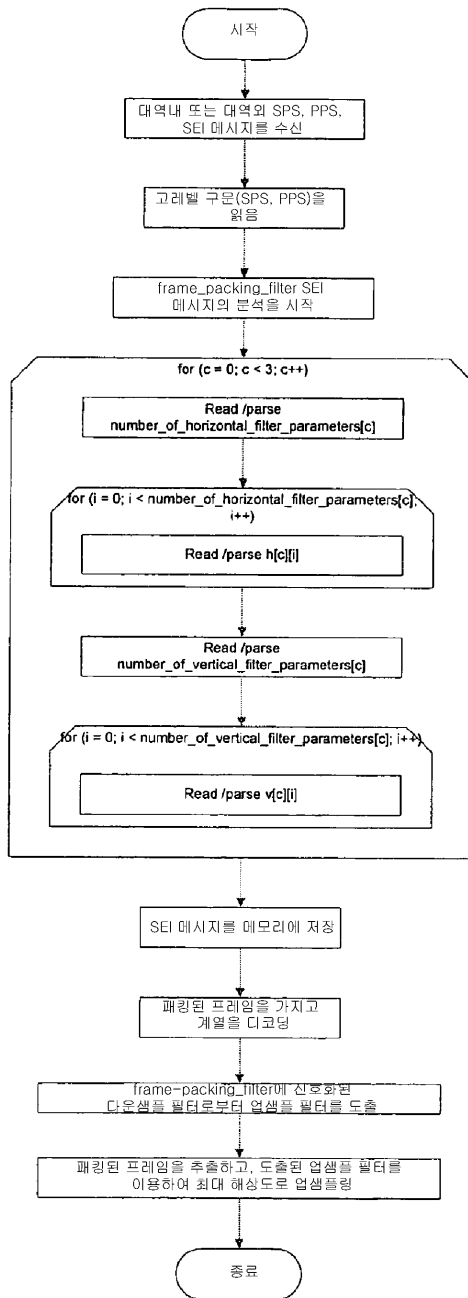
도면37

인코딩 플로우차트(실시예 2)

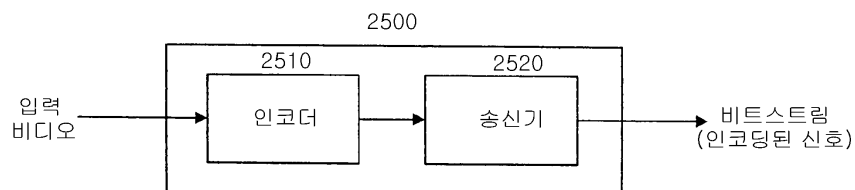


도면38

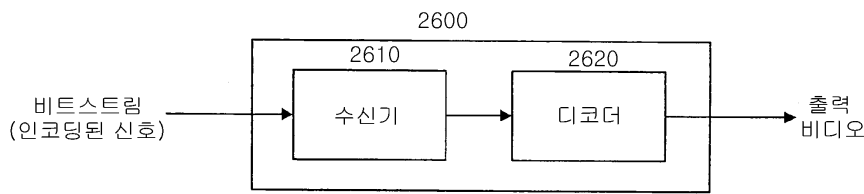
디코딩 플로우차트(실시예 2)



도면39



도면40



도면41

