(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년05월19일
(11) 등록번호 10-2400410
(24) 등록일자 2022년05월17일

(51) 국제특허분류(Int. Cl.)
H04N 19/593 (2014.01) *H04N 19/176* (2014.01)
H04N 19/70 (2014.01) *H04N 19/91* (2014.01)
(52) CPC특허분류
H04N 19/593 (2015.01)
H04N 19/176 (2015.01)
(21) 출원번호 10-2016-7036936
(22) 출원일자(국제) 2015년06월30일
심사청구일자 2020년06월12일
(85) 번역문제출일자 2016년12월29일
(65) 공개번호 10-2017-0028905
(43) 공개일자 2017년03월14일
(86) 국제출원번호 PCT/US2015/038629
(87) 국제공개번호 WO 2016/004086
국제공개일자 2016년01월07일
(30) 우선권주장
62/020,340 2014년07월02일 미국(US)
(뒷면에 계속)

(73) 특허권자
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
푸 웨이
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
조쉬 라잔 랙스맨
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
(74) 대리인
특허법인코리아나

(56) 선행기술조사문헌
Xiu X et al, Description of screen content coding technology proposal by InterDigital, JCT-VC of ITU-T and ISO/IEC 17th meeting, JCTVC-Q0037 (2014.03.29.) 1부.*
Zhu J et al, AHG10: Modified copy above mode for palette based coding, JCT-VC of ITU-T and ISO/IEC 17th meeting, JCTVC-Q0174 (2014.03.29.) 1부.*
*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 24 항 심사관 : 황수진

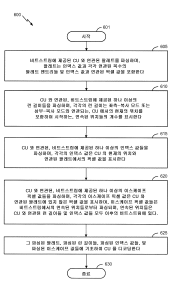
(54) 발명의 명칭 **팔레트 모드 코딩을 위한 방법**

(57) 요 약

비트스트림에 제공된 비디오 데이터를 디코딩하는 방법으로서, 비트스트림은 팔레트 모드에서 코딩된 코딩 유닛(CU) 을 포함하며, 상기 방법은, 비트스트림에 제공된 CU 와 연관된 팔레트를 파싱하는 단계; CU 와 연관된 비트스트림에 제공된 하나 이상의 런 길이들을 파싱하는 단계; CU 와 연관된 비트스트림에 제공된 하나 이상의 인덱

(뒷면에 계속)

대 표 도 - 도6



스 값들을 파싱하는 단계; 및 CU 와 연관된 비트스트림에 제공된 하나 이상의 이스케이프 픽셀 값들을 파싱하는 단계를 포함한다. 이스케이프 픽셀 값들은 비트스트림에서의 연속된 위치들로부터 파싱될 수도 있으며, 연속된 위치들은 CU 와 연관된 런 길이들 및 인덱스 값들 모두 후의 비트스트림에 있다. 본 방법은 그 파싱된 팔레트, 파싱된 런 길이들, 파싱된 인덱스 값들, 및 파싱된 이스케이프 값들에 기초하여 CU 를 디코딩하는 단계를 더 포함할 수도 있다.

(52) CPC특허분류
H04N 19/70 (2015.01)
H04N 19/91 (2015.01)
(30) 우선권주장
62/028,039 2014년07월23일 미국(US)
14/754,577 2015년06월29일 미국(US)

(72) 발명자
천 지안레
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
카르체비츠 마르타
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
시에 청-테
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
저우 펑
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
솔레 로할스 호엘
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명 세 서

청구범위

청구항 1

비트스트림에 제공된 비디오 데이터를 디코딩하는 방법으로서,

상기 비트스트림은 팔레트 모드에서 코딩된 코딩 유닛 (CU) 을 포함하며,

상기 방법은,

상기 비트스트림에 제공된 상기 CU 와 연관된 팔레트를 파싱하는 단계로서, 상기 팔레트는 인덱스 값 및 상기 인덱스 값과 연관된 픽셀 값과 각각 연관되는 복수의 팔레트 엔트리들을 포함하는, 상기 팔레트를 파싱하는 단계;

상기 CU 와 연관된 상기 비트스트림에 제공된 하나 이상의 런 길이들을 파싱하는 단계로서, 각각의 런 길이는 좌측-복사 모드 또는 상부-복사 모드와 연관되는, 상기 CU 에서의 현재의 위치를 포함하여 현재의 위치로부터 시작하는 연속된 위치들의 수를 표시하는, 상기 런 길이들을 파싱하는 단계;

상기 CU 와 연관된 상기 비트스트림에 제공된 하나 이상의 인덱스 값들을 파싱하는 단계로서, 각각의 인덱스 값은 상기 CU 에서의 상기 현재의 위치와 연관된 상기 팔레트에서의 픽셀 값을 표시하고, 상기 인덱스 값들은 상기 CU 와 연관된 모든 상기 런 길이들 후에 상기 비트스트림에서의 제 1 연속된 위치들로부터 파싱되는, 상기 인덱스 값들을 파싱하는 단계;

상기 CU 와 연관된 상기 비트스트림에 제공된 하나 이상의 이스케이프 픽셀 값들을 파싱하는 단계로서, 각각의 이스케이프 픽셀 값은 상기 팔레트에 있지 않은 픽셀 값을 포함하며, 상기 이스케이프 픽셀 값들은 상기 CU 와 연관된 모든 상기 런 길이들 및 상기 인덱스 값들 후에 상기 비트스트림에서의 제 2 연속된 위치들로부터 파싱되는, 상기 이스케이프 픽셀 값들을 파싱하는 단계; 및

상기 파싱된 팔레트, 파싱된 런 길이들, 파싱된 인덱스 값들, 및 파싱된 이스케이프 값들에 기초하여 상기 CU 를 디코딩하는 단계를 포함하는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 CU 와 연관된 모든 상기 런 길이들 및 상기 인덱스 값들을 파싱한 후, 상기 CU 와 연관된 현재의 간격의 범위를 규정하는 산술 코딩 변수를 리셋하는 단계를 더 포함하는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 CU 와 연관된 상기 이스케이프 픽셀 값들은, 상기 CU 와 연관된 모든 상기 런 길이들 및 상기 인덱스 값들을 파싱한 후 병렬로 파싱되는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 CU 와 연관된 모든 상기 런 길이들을 파싱한 후, 상기 CU 와 연관된 현재의 간격의 범위를 규정하는 산술 코딩 변수를 리셋하는 단계를 더 포함하는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 CU 와 연관된 상기 인덱스 값들은, 상기 CU 와 연관된 모든 상기 런 길이들을 파싱한 후 병렬로 파싱되는,

비트스트림에 제공된 비디오 데이터를 디코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 현재의 위치보다 바로 선행하는 상기 CU 에서의 제 1 위치가 상부-복사 런의 끝과 연관된다고 결정하는 단계; 및

상기 CU 에서의 상기 제 1 위치가 상부-복사 런의 끝과 연관된다고 결정하는 것에 응답하여, 상기 CU 에서의 상기 현재의 위치보다 바로 위에 있는 제 2 위치가 이스케이프 픽셀 값과 연관되는지 여부를 결정함이 없이, 상기 현재의 위치와 연관된 인덱스 값을 결정하는 단계를 더 포함하는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 CU 는 이스케이프 픽셀 값을 포함하는 상부-복사 런 또는 좌측-복사 런 중 하나를 포함하는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 방법.

청구항 8

비트스트림에 제공된 비디오 데이터를 디코딩하는 장치로서,

상기 비트스트림과 연관된 비디오 데이터를 저장하도록 구성된 메모리로서, 상기 비트스트림은 팔레트 모드에서 코딩된 코딩 유닛 (CU) 을 포함하는, 상기 메모리; 및

상기 메모리와 통신하는 프로세서를 포함하며,

상기 프로세서는,

상기 비트스트림에 제공된 상기 CU 와 연관된 팔레트를 파싱하는 것으로서, 상기 팔레트는 인덱스 값 및 상기 인덱스 값과 연관된 픽셀 값과 각각 연관되는 복수의 팔레트 엔트리들을 포함하는, 상기 팔레트를 파싱하고;

상기 CU 와 연관된 상기 비트스트림에 제공된 하나 이상의 런 길이들을 파싱하는 것으로서, 각각의 런 길이는 좌측-복사 모드 또는 상부-복사 모드와 연관되는, 상기 CU 에서의 현재의 위치를 포함하여 현재의 위치로부터 시작하는 연속된 위치들의 수를 표시하는, 상기 런 길이들을 파싱하고;

상기 CU 와 연관된 상기 비트스트림에 제공된 하나 이상의 인덱스 값들을 파싱하는 것으로서, 각각의 인덱스 값은 상기 CU 에서의 상기 현재의 위치와 연관된 상기 팔레트에서의 픽셀 값을 표시하고, 상기 인덱스 값들은 상기 CU 와 연관된 모든 상기 런 길이들 후에 상기 비트스트림에서의 제 1 연속된 위치들로부터 파싱되는, 상기 인덱스 값들을 파싱하고;

상기 CU 와 연관된 상기 비트스트림에 제공된 하나 이상의 이스케이프 픽셀 값들을 파싱하는 것으로서, 각각의 이스케이프 픽셀 값은 상기 팔레트에 있지 않은 픽셀 값을 포함하며, 상기 이스케이프 픽셀 값들은 상기 CU 와 연관된 모든 상기 런 길이들 및 상기 인덱스 값들 후에 상기 비트스트림에서의 제 2 연속된 위치들로부터 파싱되는, 상기 이스케이프 픽셀 값들을 파싱하고; 그리고

상기 파싱된 팔레트, 파싱된 런 길이들, 파싱된 인덱스 값들, 및 파싱된 이스케이프 값들에 기초하여 상기 CU 를 디코딩하도록 구성되는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 장치.

청구항 9

제 8 항에 있어서,

상기 프로세서는, 상기 CU 와 연관된 모든 상기 런 길이들 및 상기 인덱스 값들을 파싱한 후 상기 CU 와 연관된 현재의 간격의 범위를 규정하는 산술 코딩 변수를 리셋하도록 더 구성되는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 장치.

청구항 10

제 8 항에 있어서,

상기 프로세서는, 상기 CU 와 연관된 모든 상기 런 길이들 및 상기 인덱스 값들을 파싱한 후 상기 CU 와 연관된 상기 이스케이프 픽셀 값들을 병렬로 파싱하도록 구성되는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 장치.

청구항 11

제 8 항에 있어서,

상기 프로세서는, 상기 CU 와 연관된 모든 상기 런 길이들을 파싱한 후 상기 CU 와 연관된 현재의 간격의 범위를 규정하는 산술 코딩 변수를 리셋하도록 더 구성되는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 장치.

청구항 12

제 8 항에 있어서,

상기 프로세서는, 상기 CU 와 연관된 모든 상기 런 길이들을 파싱한 후 상기 CU 와 연관된 상기 인덱스 값들을 병렬로 파싱하도록 구성되는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 장치.

청구항 13

제 8 항에 있어서,

상기 프로세서는,

상기 현재의 위치보다 바로 선행하는 상기 CU 에서의 제 1 위치가 상부-복사 런의 끝과 연관된다고 결정하고; 그리고

상기 CU 에서의 상기 제 1 위치가 상부-복사 런의 끝과 연관된다고 결정하는 것에 응답하여, 상기 CU 에서의 상기 현재의 위치보다 바로 위에 있는 제 2 위치가 이스케이프 픽셀 값과 연관되는지 여부를 결정함이 없이, 상기 현재의 위치와 연관된 인덱스 값을 결정하도록 더 구성되는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 장치.

청구항 14

제 8 항에 있어서,

상기 CU 는 이스케이프 픽셀 값을 포함하는 상부-복사 런 또는 좌측-복사 런 중 하나를 포함하는, 비트스트림에 제공된 비디오 데이터를 디코딩하는 장치.

청구항 15

비트스트림에서의 비디오 데이터를 인코딩하는 방법으로서,

코딩 유닛 (CU) 에서의 복수의 픽셀들을 분석하는 단계로서, 각각의 픽셀은 그와 연관된 픽셀 값을 가지는, 복수의 픽셀들을 분석하는 단계;

상기 CU 에서의 상기 복수의 픽셀들에 기초하여 팔레트를 발생시키는 단계로서, 상기 팔레트는 인덱스 값 및 상기 인덱스 값과 연관된 픽셀 값과 각각 연관되는 복수의 팔레트 엔트리들을 포함하는, 상기 팔레트를 발생시키는 단계;

상기 비트스트림에서의 상기 CU 와 연관된 하나 이상의 런 길이들을 결정하는 단계로서, 각각의 런 길이는 좌측-복사 모드 또는 상부-복사 모드와 연관되는, 상기 CU 에서의 현재의 위치를 포함하여 현재의 위치로부터 시작하는 연속된 위치들의 수를 표시하는, 상기 런 길이들을 결정하는 단계;

상기 비트스트림에서의 상기 CU 와 연관된 하나 이상의 인덱스 값들을 결정하는 단계로서, 각각의 인덱스 값은 상기 CU 에서의 상기 현재의 위치와 연관된 상기 팔레트에서의 픽셀 값을 표시하는, 상기 인덱스 값들을 결정하는 단계;

상기 비트스트림에서의 상기 CU 와 연관된 하나 이상의 이스케이프 픽셀 값들을 결정하는 단계로서, 각각의 이

스케이프 픽셀 값은 상기 팔레트에 있지 않은 픽셀 값을 표시하는, 상기 이스케이프 픽셀 값들을 결정하는 단계; 및

상기 발생된 팔레트, 결정된 런 길이들, 결정된 인덱스 값들, 및 결정된 이스케이프 픽셀 값들에 기초하여 상기 CU 를 인코딩하는 단계를 포함하며,

상기 인덱스 값들은 상기 CU 와 연관된 모든 상기 런 길이들 후에 상기 비트스트림에서의 제 1 연속된 위치들에서 인코딩되고, 상기 이스케이프 픽셀 값들은 상기 CU 와 연관된 모든 상기 런 길이들 및 상기 인덱스 값들 후에 상기 비트스트림에서의 제 2 연속된 위치들에서 인코딩되는, 비트스트림에서의 비디오 데이터를 인코딩하는 방법.

청구항 16

제 15 항에 있어서,

상기 현재의 위치보다 바로 선행하는 상기 CU 에서의 제 1 위치가 상부-복사 런의 끝과 연관된다고 결정하는 단계; 및

상기 CU 에서의 상기 제 1 위치가 상부-복사 런의 끝과 연관된다고 결정하는 것에 응답하여, 상기 CU 에서의 상기 현재의 위치보다 바로 위에 있는 제 2 위치가 이스케이프 픽셀 값과 연관되는지 여부를 결정함이 없이, 상기 현재의 위치와 연관된 인덱스 값을 결정하는 단계를 더 포함하는, 비트스트림에서의 비디오 데이터를 인코딩하는 방법.

청구항 17

제 15 항에 있어서,

상기 CU 는 이스케이프 픽셀 값을 포함하는 상부-복사 런 또는 좌측-복사 런 중 하나를 포함하는, 비트스트림에서의 비디오 데이터를 인코딩하는 방법.

청구항 18

제 15 항에 있어서,

상기 CU 와 연관된 상기 인덱스 값들 중에서 가장 빠른 위치를 가지는 인덱스 값에 대응하는 상기 비트스트림에서의 위치를 표시하는 제 1 오프셋을 결정하는 단계를 더 포함하는, 비트스트림에서의 비디오 데이터를 인코딩하는 방법.

청구항 19

제 15 항에 있어서,

상기 CU 와 연관된 상기 이스케이프 픽셀 값들 중에서 가장 빠른 위치를 가지는 이스케이프 픽셀 값에 대응하는 상기 비트스트림에서의 위치를 표시하는 제 2 오프셋을 결정하는 단계를 더 포함하는, 비트스트림에서의 비디오 데이터를 인코딩하는 방법.

청구항 20

비트스트림에서의 비디오 데이터를 인코딩하는 장치로서,

상기 비트스트림과 연관된 비디오 데이터를 저장하도록 구성된 메모리로서, 상기 비트스트림은 팔레트 모드에서 코딩된 코딩 유닛 (CU) 을 포함하는, 상기 메모리; 및

상기 메모리와 통신하는 프로세서를 포함하며,

상기 프로세서는,

코딩 유닛 (CU) 에서의 복수의 픽셀들을 분석하는 것으로서, 각각의 픽셀은 그와 연관된 픽셀 값을 가지는, 상기 복수의 픽셀들을 분석하고;

상기 CU 에서의 상기 복수의 픽셀들에 기초하여 팔레트를 발생시키는 것으로서, 상기 팔레트는 인덱스 값 및 상기 인덱스 값과 연관된 픽셀 값과 각각 연관되는 복수의 팔레트 엔트리들을 포함하는, 상기 팔레트를 발생시키

고;

상기 비트스트림에서의 상기 CU 와 연관된 하나 이상의 런 길이들을 결정하는 것으로서, 각각의 런 길이는 좌측-복사 모드 또는 상부-복사 모드와 연관되는, 상기 CU 에서의 현재의 위치를 포함하여 현재의 위치로부터 시작하는 연속된 위치들의 수를 표시하는, 상기 런 길이들을 결정하고;

상기 비트스트림에서의 상기 CU 와 연관된 하나 이상의 인덱스 값들을 결정하는 것으로서, 각각의 인덱스 값은 상기 CU 에서의 상기 현재의 위치와 연관된 상기 팔레트에서의 픽셀 값을 표시하는, 상기 인덱스 값들을 결정하고;

상기 비트스트림에서의 상기 CU 와 연관된 하나 이상의 이스케이프 픽셀 값들을 결정하는 것으로서, 각각의 이스케이프 픽셀 값은 상기 팔레트에 있지 않은 픽셀 값을 표시하는, 상기 이스케이프 픽셀 값들을 결정하고; 그리고

상기 발생된 팔레트, 결정된 런 길이들, 결정된 인덱스 값들, 및 결정된 이스케이프 픽셀 값들에 기초하여 상기 CU 를 인코딩하도록 구성되며,

상기 인덱스 값들은 상기 CU 와 연관된 모든 상기 런 길이들 후에 상기 비트스트림에서의 제 1 연속된 위치들에서 인코딩되고, 상기 이스케이프 픽셀 값들은 상기 CU 와 연관된 모든 상기 런 길이들 및 상기 인덱스 값들 후에 상기 비트스트림에서의 제 2 연속된 위치들에서 인코딩되는, 비트스트림에서의 비디오 데이터를 인코딩하는 장치.

청구항 21

제 20 항에 있어서,

상기 프로세서는,

상기 현재의 위치보다 바로 선행하는 상기 CU 에서의 제 1 위치가 상부-복사 런의 끝과 연관된다고 결정하고; 그리고

상기 CU 에서의 상기 제 1 위치가 상부-복사 런의 끝과 연관된다고 결정하는 것에 응답하여, 상기 CU 에서의 상기 현재의 위치보다 바로 위에 있는 제 2 위치가 이스케이프 픽셀 값과 연관되는지 여부를 결정함이 없이, 상기 현재의 위치와 연관된 인덱스 값을 결정하도록 더 구성되는, 비트스트림에서의 비디오 데이터를 인코딩하는 장치.

청구항 22

제 20 항에 있어서,

상기 CU 는 이스케이프 픽셀 값을 포함하는 상부-복사 런 또는 좌측-복사 런 중 하나를 포함하는, 비트스트림에서의 비디오 데이터를 인코딩하는 장치.

청구항 23

제 20 항에 있어서,

상기 프로세서는 상기 CU 와 연관된 상기 인덱스 값들 중에서 가장 빠른 위치를 가지는 인덱스 값에 대응하는 상기 비트스트림에서의 위치를 표시하는 제 1 오프셋을 결정하도록 더 구성되는, 비트스트림에서의 비디오 데이터를 인코딩하는 장치.

청구항 24

제 20 항에 있어서,

상기 CU 와 연관된 상기 이스케이프 픽셀 값들 중에서 가장 빠른 위치를 가지는 이스케이프 픽셀 값에 대응하는 상기 비트스트림에서의 위치를 표시하는 제 2 오프셋을 결정하는 단계를 더 포함하는, 비트스트림에서의 비디오 데이터를 인코딩하는 장치.

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

발명의 설명

기 술 분 야

[0001] 본 개시물은 비디오 코딩 및 압축의 분야, 특히 스크린 콘텐츠 코딩에 관한 것이다.

배 경 기 술

[0002] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기들 (PDA들), 랩탑 또는 데스크탑 컴퓨터들, 디지털 카메라들, 디지털 리코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 원격 화상회의 디바이스들 등을 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 MPEG-2 (Moving Picture Experts Group-2), MPEG-4, ITU-T (International Telegraph Union-Telecommunication Standardization Sector) H.263, ITU-T H.264/MPEG-4, 파트 10, AVC (Advanced Video Coding) 에 의해 정의된 표준들, HEVC (High Efficiency Video Coding) 표준, 및 이런 표준들의 확장판들에서 설명되는 것들과 같은, 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 코딩 기법들을 구현함으로써, 디지털 비디오 정보를 좀더 효율적으로 송신하거나, 수신하거나, 인코딩하거나, 디코딩하거나, 및/또는 저장할 수도 있다.

[0003] 고속 인터넷 액세스의 보급으로 인해, 원격 데스크탑 공유, 가상 데스크탑 기반구조, 및 무선 디스플레이와 같은, 최신 비디오 애플리케이션들은 스크린 콘텐츠의 높은 압축 효율을 요구한다. 그러나, 추가적인 인트라 및 인터 비디오 코딩 툴들은 주로 자연 콘텐츠들용으로 설계되었다. 스크린 콘텐츠는 그들 전통적인 코딩 툴들을 덜 충분하게 만드는, 자연 콘텐츠들과 비교하여 현저하게 상이한 특성들 (예컨대, 날카로운 에지들 및 적거나 전혀 없는 잡음) 을 갖는다.

발명의 내용

과제의 해결 수단

[0004] 본 개시물의 시스템들, 방법들 및 디바이스들은 여러 혁신적인 양태들을 각각 가지며, 그 중 어떤 단 하나의 양태도 본원에서 개시된 바람직한 속성들을 단독으로 담당하지 않는다.

[0005] 일 양태에서, 비트스트림에서의 비디오 데이터를 디코딩하는 방법으로서, 상기 비트스트림은 팔레트 모드에서 코딩된 코딩 유닛 (CU) 을 포함하며, 상기 방법은, 비트스트림에 제공된 CU 와 연관된 팔레트를 파싱하는 단계로서, 상기 팔레트가 인덱스 값과 각각 연관된 복수의 팔레트 엔트리들 및 인덱스 값과 연관된 픽셀 값을 포함하는, 상기 팔레트를 파싱하는 단계; CU 와 연관된 비트스트림에 제공된 하나 이상의 런 길이들을 파싱하는 단계로서, 각각의 런 길이가 좌측-복사 모드 또는 상부-복사 모드와 연관되는, CU 에서의 현재의 위치를 포함하여 시작하는, 연속된 위치들의 수를 표시하는, 상기 런 길이들을 파싱하는 단계; CU 와 연관된 비트스트림에 제공된 하나 이상의 인덱스 값들을 파싱하는 단계로서, 각각의 인덱스 값이 CU 에서의 현재의 위치와 연관되는 팔레트에서의 픽셀 값을 표시하는, 상기 인덱스 값들을 파싱하는 단계; CU 와 연관된 비트스트림에 제공된 하나 이상의 이스케이프 픽셀 값들을 파싱하는 단계로서, 각각의 이스케이프 픽셀 값이 팔레트에 있지 않은 픽셀 값을 표시하며, 이스케이프 픽셀 값들이 비트스트림에서 연속된 위치들로부터 파싱되며, 연속된 위치들이 CU 와 연관된 런 길이들 및 인덱스 값들 모두 후의 비트스트림에 있는, 상기 이스케이프 픽셀 값들을 파싱하는 단계; 및

그 파싱된 팔레트, 파싱된 런 길이들, 파싱된 인덱스 값들, 및 파싱된 이스케이프 값들에 기초하여, CU 를 디코딩하는 단계를 포함한다.

[0006] 다른 양태에서, 비트스트림에 제공된 비디오 데이터를 디코딩하는 장치는 메모리 및 메모리와 통신하는 프로세서를 포함한다. 메모리는 비트스트림과 연관된 비디오 데이터를 저장하도록 구성되며, 상기 비트스트림은 팔레트 모드에서 코딩된 코딩 유닛 (CU) 을 포함한다. 프로세서는, 비트스트림에 제공된 CU 와 연관된 팔레트를 파싱하는 것으로서, 상기 팔레트가 인덱스 값과 각각 연관된 복수의 팔레트 엔트리들 및 인덱스 값과 연관된 픽셀 값을 포함하는, 상기 팔레트를 파싱하고; CU 와 연관된 비트스트림에 제공된 하나 이상의 런 길이들을 파싱하는 것으로서, 각각의 런 길이가 좌측-복사 모드 또는 상부-복사 모드와 연관되는, CU 에서의 현재의 위치를 포함하여 시작하는, 연속된 위치들의 수를 표시하는, 상기 런 길이들을 파싱하고; CU 와 연관된 비트스트림에 제공된 하나 이상의 인덱스 값들을 파싱하는 것으로서, 각각의 인덱스 값이 CU 에서의 현재의 위치와 연관되는 팔레트에서의 픽셀 값을 표시하는, 상기 인덱스 값들을 파싱하고; CU 와 연관된 비트스트림에 제공된 하나 이상의 이스케이프 픽셀 값들을 파싱하는 것으로서, 각각의 이스케이프 픽셀 값이 팔레트에 있지 않은 픽셀 값을 표시하며, 이스케이프 픽셀 값들이 비트스트림에서 연속된 위치들로부터 파싱되며, 연속된 위치들이 CU 와 연관된 런 길이들 및 인덱스 값들 모두 후의 비트스트림에 있는, 상기 이스케이프 픽셀 값들을 파싱하고; 그리고, 그 파싱된 팔레트, 파싱된 런 길이들, 파싱된 인덱스 값들, 및 파싱된 이스케이프 값들에 기초하여 CU 를 디코딩하도록 구성된다.

[0007] 일 양태에서, 비트스트림에서의 비디오 데이터를 인코딩하는 방법은, 코딩 유닛 (CU) 에서의 복수의 픽셀들을 분석하는 단계로서, 각각의 픽셀이 그와 연관된 픽셀 값을 가지는, 상기 복수의 픽셀들을 분석하는 단계; CU 에서의 복수의 픽셀들에 기초하여 팔레트를 발생시키는 단계로서, 상기 팔레트가 인덱스 값과 각각 연관된 복수의 팔레트 엔트리들 및 인덱스 값과 연관된 픽셀 값을 포함하는, 상기 팔레트를 발생시키는 단계; 비트스트림에서 CU 와 연관된 하나 이상의 런 길이들을 결정하는 단계로서, 각각의 런 길이가 좌측-복사 모드 또는 상부-복사 모드와 연관되는, CU 에서의 현재의 위치를 포함하여 시작하는, 연속된 위치들의 수를 표시하는, 상기 런 길이들을 결정하는 단계; 비트스트림에서 CU 와 연관된 하나 이상의 인덱스 값들을 결정하는 단계로서, 각각의 인덱스 값이 CU 에서의 현재의 위치와 연관되는 팔레트에서의 픽셀 값을 표시하는, 상기 인덱스 값들을 결정하는 단계; 비트스트림에서 CU 와 연관된 하나 이상의 이스케이프 픽셀 값들을 결정하는 단계로서, 각각의 이스케이프 픽셀 값이 팔레트에 있지 않은 픽셀 값을 표시하는, 상기 이스케이프 픽셀 값들을 결정하는 단계; 및 그 발생된 팔레트, 결정된 런 길이들, 결정된 인덱스 값들, 및 결정된 이스케이프 픽셀 값들에 기초하여 CU 를 인코딩하는 단계를 포함하며, 상기 이스케이프 픽셀 값들은 비트스트림에서의 연속된 위치들에서 인코딩되며, 연속된 위치들은 CU 와 연관된 런 길이들 및 인덱스 값들 모두 후의 비트스트림에 있다.

[0008] 다른 양태에서, 비트스트림에서의 비디오 데이터를 인코딩하는 장치는 메모리 및 메모리와 통신하는 프로세서를 포함한다. 메모리는 비트스트림과 연관된 비디오 데이터를 저장하도록 구성되며, 상기 비트스트림은 팔레트 모드에서 코딩된 코딩 유닛 (CU) 을 포함한다. 프로세서는, 코딩 유닛 (CU) 에서의 복수의 픽셀들을 분석하는 것으로서, 각각의 픽셀이 그와 연관된 픽셀 값을 가지는, 상기 복수의 픽셀들을 분석하고; CU 에서의 복수의 픽셀들에 기초하여 팔레트를 발생시키는 것으로서, 상기 팔레트가 인덱스 값과 각각 연관된 복수의 팔레트 엔트리들 및 인덱스 값과 연관된 픽셀 값을 포함하는, 상기 팔레트를 발생시키고; 비트스트림에서 CU 와 연관된 하나 이상의 런 길이들을 결정하는 것으로서, 각각의 런 길이가 좌측-복사 모드 또는 상부-복사 모드와 연관되는, CU 에서의 현재의 위치를 포함하여 시작하는, 연속된 위치들의 수를 표시하는, 상기 런 길이들을 결정하고; 비트스트림에서 CU 와 연관된 하나 이상의 인덱스 값들을 결정하는 것으로서, 각각의 인덱스 값이 CU 에서의 현재의 위치와 연관되는 팔레트에서의 픽셀 값을 표시하는, 상기 인덱스 값들을 결정하고; 비트스트림에서 CU 와 연관된 하나 이상의 이스케이프 픽셀 값들을 결정하는 것으로서, 각각의 이스케이프 픽셀 값이 팔레트에 있지 않은 픽셀 값을 표시하는, 상기 이스케이프 픽셀 값들을 결정하고; 그리고, 그 발생된 팔레트, 결정된 런 길이들, 결정된 인덱스 값들, 및 결정된 이스케이프 픽셀 값들에 기초하여 CU 를 인코딩하도록 구성되며, 이스케이프 픽셀 값들이 비트스트림에서의 연속된 위치들에서 인코딩되며, 연속된 위치들이 CU 와 연관된 런 길이들 및 인덱스 값들 모두 후의 비트스트림에 있다.

도면의 간단한 설명

[0009] 도 1a 는 본 개시물에서 설명하는 양태들에 따른 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 1b 는 본 개시물에서 설명하는 양태들에 따른 기법들을 수행할 수도 있는 다른 예시적인 비디오 인코딩 및

디코딩 시스템을 예시하는 블록도이다.

도 2 는 본 개시물에서 설명하는 양태들에 따른 기법들을 구현할 수도 있는 예시적인 비디오 인코더의 예를 예시하는 블록도이다.

도 3 은 본 개시물에서 설명하는 양태들에 따른 기법들을 구현할 수도 있는 예시적인 비디오 디코더의 예를 예시하는 블록도이다.

도 4 는 입력 CU, 인덱스 블록, 이스케이프 픽셀, 및 CU 와 연관된 팔레트를 예시하는 블록도이다.

도 5 는 본 개시물에서 설명된 양태들에 따른, 비트스트림에서의 비디오 데이터를 코딩하는 방법을 예시하는 플로우차트이다.

도 6 은 본 개시물에서 설명된 양태들에 따른, 비트스트림에서의 비디오 데이터를 디코딩하는 방법을 예시하는 플로우차트이다.

도 7 은 본 개시물에서 설명된 양태들에 따른, 비트스트림에서의 비디오 데이터를 디코딩하는 다른 방법을 예시하는 플로우차트이다.

도 8 은 본 개시물에서 설명된 양태들에 따른, 비트스트림에서의 비디오 데이터를 코딩하는 다른 방법을 예시하는 플로우차트이다.

도 9 는 본 개시물에서 설명된 양태들에 따른, 비트스트림에서의 비디오 데이터를 인코딩하는 방법을 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0010] 스크린 콘텐츠 코딩의 기존 구현예들에서, 비트스트림에 일부 리던던시들이 있을 수도 있다. 이들 리던던시들은 어떤 조건들이 만족될 때 어떤 신택스 엘리먼트 시그널링을 스킵함으로써 제거될 수도 있다. 게다가, 일부 신택스 엘리먼트들은 파싱 의존성을 도입할 수도 있다. 예를 들어, 현재의 픽셀이 블록의 제 1 라인에 있으면, 디코더가 런 모드를 인덱스 복사 모드 (예컨대, 좌측 복사 모드) 인 것으로 추론할 수도 있기 때문에, 런 모드를 표시하는 신택스 엘리먼트가 시그널링될 필요가 없을 수도 있다. 게다가, 디코더가 인덱스 값을 먼저 디코딩하는 경우에, 그리고 디코딩된 인덱스 값에 의존하여, 디코더는 (예컨대, 그 인덱스 값이 이스케이프 인덱스 값을 나타내는지 여부에 기초하여) 그 모드가 인덱스 복사 모드 또는 이스케이프 모드인지 여부를 결정한다. 디코더가 그 모드를 인덱스 복사 모드인 것으로 결정하면, 디코더 파서는 런 길이를 계속 파싱한다. 디코더가 그 모드를 이스케이프 모드인 것으로 결정하면, 디코더 파서는 이스케이프 값들 및/또는 런 길이를 계속 파싱할 수도 있다. 그 파서들이 대개 디코더들보다 아주 더 높은 속도에서 동작하기 때문에, 디코딩 엔진과 파싱 엔진 사이의 이러한 의존성은 (예컨대, 디코딩 엔진이 파싱된 비트들을 디코딩하기를 파싱 엔진이 대기해야 할 수도 있기 때문에) 파서의 처리량에 영향을 미칠 수도 있다. 따라서, 팔레트 코딩 모드에서 코딩된 블록들을 프로세싱하는 향상된 방법이 요망된다. 본 출원에서, 팔레트 모드에서의 파싱 의존성을 피하거나 또는 감소시키기 위해 비트스트림에서의 팔레트 엘리먼트들을 편성하는 여러 신규한 방법들이 설명된다.

[0011] 하기 설명에서, 어떤 실시형태들에 관련된 H.264/AVC (Advanced Video Coding) 기법들이 설명되며; HEVC 표준 및 관련된 기법들이 또한 설명된다. 어떤 실시형태들은 본원에서 HEVC 및/또는 H.264 표준들의 상황에서 설명되지만, 당업자는 본원에서 개시된 시스템들 및 방법들은 임의의 적합한 비디오 코딩 표준에 적용가능할 수도 있음을 이해할 것이다. 예를 들어, 본원에서 개시된 실시형태들은 다음 표준들 중 하나 이상에 적용가능할 수도 있다: 국제 전기통신 연합 (ITU) 전기통신 표준화 부문 (ITU-T) H.261, ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) MPEG 1 Visual, ITU-T H.262 또는 ISO/IEC MPEG-2 Visual, ITU-T H.263, 범위 확장판을 포함한, ISO/IEC MPEG 4 Visual 및 (또한, ISO/IEC MPEG-4 AVC 로서 알려진) ITU-T H.264.

[0012] HEVC 는 일반적으로 많은 측면들에서 이전 비디오 코딩 표준들의 프레임워크를 따른다. HEVC 에서의 예측의 유닛은 어떤 이전 비디오 코딩 표준들에서의 예측의 유닛들 (예컨대, 매크로블록들) 과는 상이하다. 실제로, 매크로블록의 컨셉은 어떤 이전 비디오 코딩 표준들에서 이해되는 바와 같이 HEVC 에 존재하지 않는다. 매크로블록은 다른 가능한 이점들 중, 높은 유연성을 제공할 수도 있는 쿼드트리 방식에 기초하여, 계층적 구조로 대체된다. 예를 들어, HEVC 방식 내에서, 블록들의 3개의 유형들, 즉, 코딩 유닛 (CU), 예측 유닛

(PU), 및 변환 유닛 (TU) 이 정의된다. CU 는 영역 분할 (splitting) 의 기본적인 단위를 지칭할 수도 있다. CU 는 매크로블록의 컨셉과 유사한 것으로 간주될 수도 있으며, 그러나 HEVC 는 CU들의 최대 사이즈를 제한하지 않으며, 4개의 동일한 사이즈 CU들로의 재귀적인 분할을 가능하게 하여 콘텐츠 적응성을 향상시킬 수도 있다. PU 는 인터/인트라 예측의 기본적인 단위로서 간주될 수도 있으며, 단일 PU 는 불규칙적인 이미지 패턴들을 효과적으로 코딩하기 위해 다수의 임의의 형태 파티션들을 포함할 수도 있다. TU 는 변환의 기본적인 단위로서 간주될 수도 있다. TU 는 PU 와 독립적으로 정의될 수 있으며; 그러나, TU 의 사이즈는 TU 가 속하는 CU 의 사이즈에 한정될 수도 있다. 3개의 상이한 컨셉들로의 블록 구조의 이러한 분리는 각각의 유닛이 그 유닛의 개개의 역할에 따라서 각각 최적화될 수 있게 함으로써, 향상된 코딩 효율을 초래할 수도 있다.

[0013] **비디오 코딩 표준들**

[0014] 비디오 리코더 또는 컴퓨터에 의해 발생되는 비디오 이미지, TV 이미지, 정지 화상 또는 이미지와 같은, 디지털 이미지는 수평 및 수직 라인들에 배열된 픽셀들 또는 샘플들을 포함할 수도 있다. 단일 이미지에서 픽셀들의 개수는 일반적으로 수만개이다. 각각의 픽셀은 일반적으로 휘도 및 색차 정보를 포함한다. 압축이 없다면, 이미지 인코더로부터 이미지 디코더로 운반되는 정보의 전적인 양은 실시간 이미지 송신을 실행불가능하게 만들 것이다. 송신되는 정보의 양을 감소시키기 위해, JPEG, MPEG 및 H.263 표준들과 같은, 다수의 상이한 압축 방법들이 개발되었다.

[0015] 비디오 코딩 표준들은 ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 또는 ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual 및 (ISO/IEC MPEG-4 AVC 로서 또한 알려진) ITU-T H.264, 및 범위 확장판을 포함한 HEVC 를 포함한다.

[0016] 게다가, 비디오 코딩 표준, 즉 HEVC 는, ITU-T 비디오 코딩 전문가 그룹 (VCEG) 과 ISO/IEC MPEG 의 비디오 코딩에 관한 합동 작업팀 (JCT-VC) 에 의해 개발되었다. HEVC 초안 10 에 대한 전체 인용은 ITU-T SG16 WP3 와 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 12차 회의: 2013년 1월 14일 내지 2013년 1월 23일, 스위스, 제네바, 문서 JCTVC-L1003, Bross 등, "High efficiency video coding (HEVC) text specification draft 10" 이다. HEVC 에 대한 범위 확장판이 또한 JCT-VC 에 의해 개발되고 있다.

[0017] **비디오 코딩 시스템**

[0018] 신규한 시스템들, 장치들, 및 방법들은 이하에서 첨부 도면들을 참조하여 좀더 충분히 설명된다. 본 개시물은 그러나, 많은 상이한 형태들로 구현될 수도 있으며, 본 개시물을 통해서 제시되는 임의의 특정의 구조 또는 기능에 한정되는 것으로 해석되어서는 안된다. 대신, 이들 양태들은 본 개시물이 철저하고 완전하게 되도록, 그리고 본 개시물의 범위를 당업자들에게 충분히 전달하기 위해서 제공된다. 본원에서의 교시들에 기초하여, 당업자는 본 개시물의 범위가 본 개시물의 임의의 다른 양태와 독립적으로 구현되든 그와 결합되든, 본원에서 개시된 신규한 시스템들, 장치들, 및 방법들의 임의의 양태를 포괄하도록 의도되는 것으로 이해하여야 한다. 예를 들어, 본원에서 개시된 임의 개수의 양태들을 이용하여, 장치가 구현될 수도 있거나 또는 방법이 실시될 수도 있다. 게다가, 본 개시물의 범위는 본원에서 개시된 본 개시물의 여러 양태들에 추가해서 또는 이 이외에, 다른 구조, 기능, 또는 구조 및 기능을 이용하여 실행되는 장치 또는 방법을 포괄하도록 의도된다. 본원에서 개시된 임의의 양태는 청구항의 하나 이상의 엘리먼트들에 의해 구현될 수도 있는 것으로 이해되어야 한다.

[0019] 특정의 양태들이 본원에서 설명되지만, 이들 양태들의 많은 변형예들 및 치환물들은 본 개시물의 범위 이내이다. 바람직한 양태들의 일부 이익들 및 이점들이 언급되지만, 본 개시물의 범위는 특유의 이점들, 용도들, 또는 목적들에 한정되는 것으로 의도되지 않는다. 대신, 본 개시물의 양태들은 상이한 무선 기술들, 시스템 구성들, 네트워크들, 및 송신 프로토콜들에 넓게 적용가능한 것으로 의도되며, 이들 중 일부가 일 예로서 도면들에 그리고 바람직한 양태들의 다음 설명에 예시된다. 상세한 설명 및 도면들은 한정하기 보다는 단지 본 개시물의 예시이며, 본 개시물의 범위는 첨부된 청구범위 및 이의 균등물들에 의해 정의된다.

[0020] 첨부 도면들은 예들을 예시한다. 첨부 도면들에서 참조 번호들로 표시된 엘리먼트들은 다음 설명에서 유사한 참조 번호들로 표시된 엘리먼트들에 대응한다. 본 개시물에서, 서수의 단어들 (예컨대, "제 1", "제 2", "제 3", 및 기타 등등) 로 시작하는 이름들을 갖는 엘리먼트들은 엘리먼트들이 특정의 순서를 갖는다는 것을 반드시 암시하지는 않는다. 대신, 이러한 서수의 단어들은 동일한 또는 유사한 유형의 상이한 엘리먼트들을

지칭하기 위해 단지 사용된다.

[0021] 도 1a 는 본 개시물에서 설명된 양태들에 따른 기법들을 이용할 수도 있는 예시적인 비디오 코딩 시스템 (10) 을 예시하는 블록도이다. 설명되는 본원에서 사용될 때, 용어 "비디오 코더" 또는 "코더" 는 비디오 인코더들 및 비디오 디코더들 양쪽을 포괄적으로 지칭한다. 본 개시물에서, 용어들 "비디오 코딩" 또는 "코딩" 은 포괄적으로 비디오 인코딩 및 비디오 디코딩을 지칭할 수도 있다. 비디오 인코더들 및 비디오 디코더들에 더해서, 본 출원에서 설명된 양태들은 트랜스코더들 (예컨대, 비트스트림을 디코딩하고 다른 비트스트림을 재-인코딩할 수 있는 디바이스들) 및 미들박스들 (middleboxes) (예컨대, 비트스트림을 수정하고, 변환하고, 및/또는 아니면 조작할 수 있는 디바이스들) 과 같은, 다른 관련된 디바이스들로 확장될 수도 있다.

[0022] 도 1a 에 나타낸 바와 같이, 비디오 코딩 시스템 (10) 은 목적지 디바이스 (14) 에 의해 추후 디코딩될 인코딩된 비디오 데이터를 발생시키는 소스 디바이스 (12) 를 포함한다. 도 1a 의 예에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 별개의 디바이스들을 구성한다. 그러나, 소스 디바이스 (12) 및 목적지 디바이스 (14) 가 도 1b 의 예에 나타낸 바와 같이 동일한 디바이스 상에 또는 부분 상에 있을 수도 있다는 점에 유의한다.

[0023] 또 다시 도 1a 를 참조하면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 노트북 (예컨대, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한, 광범위한 디바이스들 중 임의의 디바이스를 포함할 수도 있다. 여러 실시형태들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신용으로 탑재될 수도 있다.

[0024] 목적지 디바이스 (14) 는 링크 (16) 를 통해서, 디코딩될 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의 종류의 매체 또는 디바이스를 포함할 수도 있다. 도 1a 의 예에서, 링크 (16) 는 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 실시간으로 송신할 수 있게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라서 변조되어 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들과 같은, 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 근거리 네트워크, 광역 네트워크, 또는 글로벌 네트워크, 예컨대 인터넷과 같은 패킷-기반 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0025] 대안적으로, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 (옵션적으로 존재하는) 저장 디바이스 (31) 로 출력될 수도 있다. 이와 유사하게, 인코딩된 데이터는 예를 들어, 목적지 디바이스 (14) 의 입력 인터페이스 (28) 에 의해 저장 디바이스 (31) 로부터 액세스될 수도 있다. 저장 디바이스 (31) 는 하드 드라이브, 플래시 메모리, 휘발성 또는 비-휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산된 또는 로컬 액세스되는 데이터 저장 매체들 중 임의의 데이터 저장 매체를 포함할 수도 있다. 추가 예에서, 저장 디바이스 (31) 는 소스 디바이스 (12) 에 의해 발생된 인코딩된 비디오를 유지할 수도 있는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 저장된 비디오 데이터에 저장 디바이스 (31) 로부터 스트리밍 또는 다운로드를 통해서 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신하는 것이 가능한 임의 종류의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버 (예컨대, 웹사이트용), 파일 전송 프로토콜 (FTP) 서버들, NAS (network attached storage) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해서, 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버 상에 저장되는 인코딩된 비디오 데이터에 액세스하는데 적합한, 무선 채널 (예컨대, 무선 로컬 영역 네트워크 (WLAN) 접속), 유선 접속 (예컨대, 디지털 가입자 회선 (DSL), 케이블 모뎀 등), 또는 양쪽의 조합을 포함할 수도 있다. 저장 디바이스 (31) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이 양쪽의 조합일 수도 있다.

[0026] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정들에 한정되지 않는다. 이 기법들은 오버-디-에어 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들 (예컨대, HTTP (Hypertext Transfer Protocol) 를 통한 동적 적응 스트리밍 등), 데이터 저장 매체 상에의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다

른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들 중 임의의 애플리케이션의 지원 하에 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화 통신과 같은, 애플리케이션들을 지원하기 위해 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0027] 도 1a 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 일부의 경우, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 소스 디바이스 (12) 에서, 비디오 소스 (18) 는 비디오 캡쳐 디바이스, 예컨대, 비디오 카메라, 이전에 캡쳐된 비디오를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하는 비디오 공급 인터페이스, 및/또는 컴퓨터 그래픽스 데이터를 소스 비디오로서 발생시키는 컴퓨터 그래픽스 시스템과 같은 소스, 또는 이런 소스들의 조합을 포함할 수도 있다. 일 예로서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 도 1b 의 예에 예시된 바와 같이, 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 본 개시물에서 설명하는 기법들은 비디오 코딩에 일반적으로 적용가능할 수도 있으며, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

[0028] 캡쳐되거나, 사전-캡쳐되거나, 또는 컴퓨터-발생된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (12) 의 출력 인터페이스 (22) 를 통해서 목적지 디바이스 (14) 에 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는, 대안적으로) 디코딩 및/또는 플레이백을 위해, 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 추후 액세스를 위해, 저장 디바이스 (31) 상에 저장될 수도 있다. 도 1a 및 도 1b 에 예시된 비디오 인코더 (20) 는 도 2 에 예시된 비디오 인코더 (20), 또는 본원에서 설명되는 임의의 다른 비디오 인코더를 포함할 수도 있다.

[0029] 도 1a 의 예에서, 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부의 경우, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 링크 (16) 를 통해서 및/또는 저장 디바이스 (31) 로부터, 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16) 를 통해서 통신되거나, 또는 저장 디바이스 (31) 상에 제공되는 인코딩된 비디오 데이터는, 비디오 데이터를 디코딩할 때에, 비디오 디코더 (30) 와 같은 비디오 디코더에 의한 사용을 위해, 비디오 인코더 (20) 에 의해 발생되는 다양한 신택스 엘리먼트들을 포함할 수도 있다. 이런 신택스 엘리먼트들은 통신 매체 상으로 송신되거나, 저장 매체 상에 저장되거나, 또는 파일 서버에 저장된 인코딩된 비디오 데이터와 함께 포함될 수도 있다. 도 1a 및 도 1b 에 예시된 비디오 디코더 (30) 는 도 3 에 예시된 비디오 디코더 (30), 또는 본원에서 설명되는 임의의 다른 비디오 디코더를 포함할 수도 있다.

[0030] 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나 이의 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함하며, 또한 외부 디스플레이 디바이스와 인터페이스하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 그 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 디바이스를 포함할 수도 있다.

[0031] 관련된 양태들에서, 도 1b 는 예시적인 비디오 코딩 시스템 (10') 을 나타내며, 여기서 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 디바이스 (11) 상에 또는 부분에 있다. 디바이스 (11) 는 "스마트" 폰 또는 기타 등등과 같은, 전화기 핸드셋일 수도 있다. 디바이스 (11) 는 소스 디바이스 (12) 및 목적지 디바이스 (14) 와 통신 동작하는 제어기/프로세서 디바이스 (13) (옵션적으로 존재함) 를 포함할 수도 있다. 도 1b 의 비디오 코딩 시스템 (10'), 및 그의 구성요소들은, 다른 점에서는, 도 1a 의 비디오 코딩 시스템 (10), 및 그의 구성요소들과 유사하다.

[0032] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HEVC 표준과 같은 비디오 압축 표준에 따라서 동작할 수도 있으며, HEVC 테스트 모델 (HM) 을 따를 수도 있다. 이의 대안으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다른 독점 또는 산업 표준들, 예컨대, 대안적으로 MPEG4 로서 지칭되는, ITU-T H.264 표준, 파트 10, AVC, 또는 이런 표준들의 확장판들에 따라서 동작할 수도 있다. 본 개시물의 기법들은, 그러나, 임의의 특정의 코딩 표준에 한정되지 않는다. 비디오 압축 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263 을 포함한다.

[0033] 도 1a 및 도 1b 의 예에 나타내지는 않지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 오디오 인코더 및 디코더와 각각 통합될 수도 있으며, 오디오 및 비디오 양쪽의 인코딩을 공통 데이터 스트림

또는 별개의 데이터 스트림들로 처리하기에 적합한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능한 경우, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 다른 프로토콜들, 예컨대, 사용자 데이터그램 프로토콜 (UDP) 을 따를 수도 있다.

[0034] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 임의의 이들의 조합들과 같은, 다양한 적합한 인코더 회로 중 임의의 회로로 구현될 수도 있다. 이 기법들이 소프트웨어로 부분적으로 구현되는 경우, 디바이스는 본 개시물의 기법들을 수행하기 위해 소프트웨어용 명령들을 적합한 비일시성 컴퓨터-판독가능 매체에 저장하고, 그 명령들을 하드웨어에서 하나 이상의 프로세서들을 이용하여 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 쪽이든 각각의 디바이스에서, 결합된 인코더/디코더의 일부로서 통합될 수도 있다.

[0035] **비디오 코딩 프로세스**

[0036] 위에서 간단히 언급한 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 인코딩한다. 비디오 데이터는 하나 이상의 픽처들을 포함할 수도 있다. 픽처들의 각각은 비디오의 일부를 형성하는 정지 화상이다. 일부의 경우, 픽처는 비디오 "프레임" 으로서 지칭될 수도 있다. 비디오 인코더 (20) 가 비디오 데이터를 인코딩할 때, 비디오 인코더 (20) 는 비트스트림을 발생시킬 수도 있다. 비트스트림은 비디오 데이터의 코딩된 표현을 형성하는 비트들의 시퀀스를 포함할 수도 있다. 비트스트림은 코딩된 픽처들 및 연관된 데이터를 포함할 수도 있다. 코딩된 픽처는 픽처의 코딩된 표현이다.

[0037] 비트스트림을 발생시키기 위해, 비디오 인코더 (20) 는 비디오 데이터에서 각각의 픽처에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 픽처들에 대해 인코딩 동작들을 수행할 때, 비디오 인코더 (20) 는 코딩된 픽처들 및 연관되는 데이터의 시리즈를 발생시킬 수도 있다. 연관되는 데이터는 비디오 파라미터 세트들 (VPS), 시퀀스 파라미터 세트들 (SPS들), 픽처 파라미터 세트들 (PPS들), 적응 파라미터 세트들 (APS들), 및 다른 신택스 구조들을 포함할 수도 있다. SPS 는 제로 이상의 픽처들의 시퀀스들에 적용가능한 파라미터들을 포함할 수도 있다. PPS 는 제로 이상의 픽처들에 적용가능한 파라미터들을 포함할 수도 있다. APS 는 제로 이상의 픽처들에 적용가능한 파라미터들을 포함할 수도 있다. APS 에서의 파라미터들은 PPS 에서의 파라미터들보다 변할 가능성이 더 많은 파라미터들일 수도 있다.

[0038] 코딩된 픽처를 발생시키기 위해, 비디오 인코더 (20) 는 픽처를 동일-사이즈로된 비디오 블록들로 파티셔닝할 수도 있다. 비디오 블록은 샘플들의 2차원 어레이일 수도 있다. 비디오 블록들의 각각은 트리블록과 연관된다. 일부의 경우, 트리블록은 최대 코딩 유닛 (LCU) 으로서 지칭될 수도 있다. HEVC 의 트리블록들은 H.264/AVC 의 매크로블록들과 같은 이전 표준들과 대략적으로 유사할 수도 있다. 그러나, 트리블록은 특정의 사이즈에 반드시 제한되지 않으며, 하나 이상의 코딩 유닛들 (CU들) 을 포함할 수도 있다. 비디오 인코더 (20) 는 쿼드트리 파티셔닝을 이용하여, 트리블록들의 비디오 블록들을 CU들과 연관되는 비디오 블록들, 따라서 이름 "트리블록들" 로 파티셔닝할 수도 있다.

[0039] 일부 예들에서, 비디오 인코더 (20) 는 픽처를 복수의 슬라이스들로 파티셔닝할 수도 있다. 슬라이스들의 각각은 정수의 CU들을 포함할 수도 있다. 일부의 경우, 슬라이스는 정수의 트리블록들을 포함한다. 다른 경우, 슬라이스의 경계는 트리블록 내에 있을 수도 있다.

[0040] 픽처에 대해 인코딩 동작을 수행하는 것의 일부로서, 비디오 인코더 (20) 는 그 픽처의 각각의 슬라이스에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 슬라이스에 대해 인코딩 동작을 수행할 때, 비디오 인코더 (20) 는 그 슬라이스와 연관되는 인코딩된 데이터를 발생시킬 수도 있다. 슬라이스와 연관되는 인코딩된 데이터는 "코딩된 슬라이스" 로서 지칭될 수도 있다.

[0041] 코딩된 슬라이스를 발생시키기 위해, 비디오 인코더 (20) 는 슬라이스에서의 각각의 트리블록에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 트리블록에 대해 인코딩 동작을 수행할 때, 비디오 인코더 (20) 는 코딩된 트리블록을 발생시킬 수도 있다. 그 코딩된 트리블록은 트리블록의 인코딩된 버전을 나타내는 데이터를 포함할 수도 있다.

[0042] 비디오 인코더 (20) 가 코딩된 슬라이스를 발생시킬 때, 비디오 인코더 (20) 는 스캐닝 순서에 따라서 슬라이스에서의 트리블록들에 대해 인코딩 동작들을 수행할 수도 있다 (예컨대, 인코딩할 수도 있다). 예를 들어, 비디오 인코더 (20) 는 비디오 인코더 (20) 가 슬라이스에서 트리블록들의 각각을 인코딩 완료할 때까지, 슬라

이스에서 트리블록들의 최상부 로우를 가로질러 좌측에서 우측으로, 그후 트리블록들의 다음 하부 로우를 가로질러 좌측에서 우측으로, 그리고 기타 등등으로 진행하는 순서로, 슬라이스의 트리블록들을 인코딩할 수도 있다.

[0043] 래스터 스캔 순서에 따라서 트리블록들을 인코딩하는 것의 결과로서, 주어진 트리블록의 상부 및 좌측에 있는 트리블록들은 인코딩되었을 수도 있지만, 주어진 트리블록의 하부 및 우측에 있는 트리블록들은 아직 인코딩되어 있지 않다. 그 결과, 비디오 인코더 (20) 는 주어진 트리블록을 인코딩할 때, 주어진 트리블록의 상부 및 좌측에 있는 트리블록들을 인코딩함으로써 발생되는 정보에 액세스가능할 수도 있다. 그러나, 비디오 인코더 (20) 는 주어진 트리블록을 인코딩할 때, 주어진 트리블록의 하부 및 우측에 있는 트리블록들을 인코딩함으로써 발생되는 정보에 액세스불가능할 수도 있다.

[0044] 코딩된 트리블록을 발생시키기 위해, 비디오 인코더 (20) 는 비디오 블록을 계속해서 더 작은 비디오 블록들로 분할하기 위해 트리블록의 비디오 블록에 관해 쿼드트리 파티셔닝을 재귀적으로 수행할 수도 있다. 더 작은 비디오 블록들의 각각은 상이한 CU 와 연관될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 트리블록의 비디오 블록을 4개의 동일-사이즈로된 서브-블록들로 파티셔닝하고, 서브-블록들 중 하나 이상을 4개의 동일-사이즈로된 서브-서브-블록들로 파티셔닝하는, 등을 포함할 수도 있다. 파티셔닝된 CU 는 그의 비디오 블록이 다른 CU들과 연관되는 비디오 블록들로 파티셔닝되는 CU 일 수도 있다. 비-파티셔닝된 CU 는 비디오 블록이 다른 CU들과 연관되는 비디오 블록들로 파티셔닝되지 않는 CU 일 수도 있다.

[0045] 비트스트림에서 하나 이상의 신택스 엘리먼트들은 비디오 인코더 (20) 가 트리블록의 비디오 블록을 파티셔닝할 수도 있는 최대 횟수를 나타낼 수도 있다. CU 의 비디오 블록은 형태가 정사각형일 수도 있다. CU 의 비디오 블록의 사이즈 (예컨대, CU 의 사이즈) 는 8x8 픽셀들로부터 최고 64x64 픽셀들 이상의 최대치를 갖는 트리블록의 비디오 블록의 사이즈 (예컨대, 트리블록의 사이즈) 까지 일 수도 있다.

[0046] 비디오 인코더 (20) 는 z-스캐닝 순서에 따라서 트리블록의 각각의 CU 에 대해 인코딩 동작들을 수행할 수도 있다 (예컨대, 인코딩할 수도 있다). 즉, 비디오 인코더 (20) 는 최상부-좌측 CU, 최상부-우측 CU, 최하부-좌측 CU, 그리고 그후 최하부-우측 CU 를, 그 순서로 인코딩할 수도 있다. 비디오 인코더 (20) 가 파티셔닝된 CU 에 관해 인코딩 동작을 수행할 때, 비디오 인코더 (20) 는 z-스캐닝 순서에 따라서, 파티셔닝된 CU 의 비디오 블록의 서브-블록들과 연관되는 CU들을 인코딩할 수도 있다. 다시 말해서, 비디오 인코더 (20) 는 좌상단 서브-블록들과 연관되는 CU, 우상단 서브-블록들과 연관되는 CU, 좌하단 서브-블록들과 연관되는 CU, 그리고, 그후 우하단 서브-블록들과 연관되는 CU 를, 그 순서대로 인코딩할 수도 있다.

[0047] z-스캐닝 순서에 따라서 트리블록의 CU들을 인코딩하는 것의 결과로서, 주어진 CU 의 상부, 좌상부, 우상부, 좌측, 및 좌하부에 있는 CU들이 인코딩되었을 수도 있다. 주어진 CU 의 하부 및 우측에 있는 CU들은 아직 인코딩되지 않았다. 그 결과, 비디오 인코더 (20) 는 주어진 CU 를 인코딩할 때 주어진 CU 에 이웃하는 일부 CU들을 인코딩함으로써 발생되는 정보에 액세스가능할 수도 있다. 그러나, 비디오 인코더 (20) 는 주어진 CU 를 인코딩할 때 주어진 CU 에 이웃하는 다른 CU들을 인코딩함으로써 발생되는 정보에 액세스불가능할 수도 있다.

[0048] 비디오 인코더 (20) 가 비-파티셔닝된 CU 를 인코딩할 때, 비디오 인코더 (20) 는 그 CU 에 대한 하나 이상의 예측 유닛들 (PU들) 을 발생시킬 수도 있다. 그 CU 의 PU들의 각각은 CU 의 비디오 블록 내에서의 상이한 비디오 블록과 연관될 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 에 대해 예측된 비디오 블록을 발생시킬 수도 있다. PU 의 예측된 비디오 블록은 샘플들의 블록일 수도 있다. 비디오 인코더 (20) 는 인트라 예측 또는 인터 예측을 이용하여, PU 에 대한 예측된 비디오 블록을 발생시킬 수도 있다.

[0049] 비디오 인코더 (20) 가 PU 의 예측된 비디오 블록을 발생시키기 위해 인트라 예측을 이용할 때, 비디오 인코더 (20) 는 PU 와 연관되는 픽처의 디코딩된 샘플들에 기초하여 PU 의 예측된 비디오 블록을 발생시킬 수도 있다.

비디오 인코더 (20) 가 CU 의 PU들의 예측된 비디오 블록들을 발생시키기 위해 인트라 예측을 이용할 때, CU 는 인트라-예측된 CU 이다. 비디오 인코더 (20) 가 PU 의 예측된 비디오 블록을 발생시키기 위해 인터 예측을 이용할 때, 비디오 인코더 (20) 는 PU 와 연관되는 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여, PU 의 예측된 비디오 블록을 발생시킬 수도 있다. 비디오 인코더 (20) 가 CU 의 PU들의 예측된 비디오 블록들을 발생시키기 위해 인터 예측을 이용하면, 그 CU 는 인터-예측된 CU 이다.

[0050] 더욱이, 비디오 인코더 (20) 가 PU 에 대한 예측된 비디오 블록을 발생시키기 위해 인터 예측을 이용할 때, 비디오 인코더 (20) 는 PU 에 대한 모션 정보를 발생시킬 수도 있다. PU 에 대한 모션 정보는 PU 의 하나 이

상의 참조 블록들을 나타낼 수도 있다. PU 의 각각의 참조 블록은 참조 픽처 내 비디오 블록일 수도 있다.

참조 픽처는 PU 와 연관되는 픽처와는 다른 픽처일 수도 있다. 일부의 경우, PU 의 참조 블록은 또한 PU 의 "참조 샘플" 로서 지칭될 수도 있다. 비디오 인코더 (20) 는 PU 의 참조 블록들에 기초하여, PU 에 대한 예측된 비디오 블록을 발생시킬 수도 있다.

[0051] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 대해 예측된 비디오 블록들을 발생시킨 후, 비디오 인코더 (20) 는 CU 의 PU들에 대한 예측된 비디오 블록들에 기초하여, CU 에 대한 잔차 데이터를 발생시킬 수도 있다.

CU 에 대한 잔차 데이터는 CU 의 PU들에 대한 예측된 비디오 블록들 및 CU 의 원래 비디오 블록에서의 샘플들 사이의 차이들을 나타낼 수도 있다.

[0052] 더욱이, 비-파티셔닝된 CU 에 대해 인코딩 동작을 수행하는 것의 일부로서, 비디오 인코더 (20) 는 CU 의 잔차 데이터에 관해 재귀적인 쿼드트리 파티셔닝을 수행하여, CU 의 잔차 데이터를 CU 의 변환 유닛들 (TU들) 과 연관된 잔차 데이터의 하나 이상의 블록들 (예컨대, 잔차 비디오 블록들) 로 파티셔닝할 수도 있다. CU 의 각각의 TU 는 상이한 잔차 비디오 블록과 연관될 수도 있다.

[0053] 비디오 인코더 (20) 는 하나 이상의 변환들을 TU들과 연관되는 잔차 비디오 블록들에 적용하여, TU들과 연관되는 변환 계수 블록들 (예컨대, 변환 계수들의 블록들) 을 발생시킬 수도 있다. 개념적으로, 변환 계수 블록은 변환 계수들의 2차원 (2D) 매트릭스일 수도 있다.

[0054] 변환 계수 블록을 발생시킨 후, 비디오 인코더 (20) 는 변환 계수 블록에 대해 양자화 프로세스를 수행할 수도 있다. 양자화는 일반적으로 변환 계수들이 변환 계수들을 나타내는데 사용되는 데이터의 양을 가능한 한 감소시키도록 양자화되어 추가적인 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 그 변환 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 예를 들어, n-비트 변환 계수는 양자화 동안 m-비트 변환 계수로 절사될 수도 있으며, 여기서, n 은 m 보다 더 크다.

[0055] 비디오 인코더 (20) 는 각각의 CU 를 양자화 파라미터 (QP) 값과 연관시킬 수도 있다. CU 와 연관되는 QP 값은 비디오 인코더 (20) 가 CU 와 연관되는 변환 계수 블록들을 양자화하는 방법을 결정할 수도 있다. 비디오 인코더 (20) 는 CU 와 연관되는 QP 값을 조정함으로써, CU 와 연관되는 변환 계수 블록들에 적용되는 양자화의 정도를 조정할 수도 있다.

[0056] 비디오 인코더 (20) 가 변환 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수 블록에서 변환 계수들을 나타내는 신택스 엘리먼트들의 세트들을 발생시킬 수도 있다. 비디오 인코더 (20) 는 컨텍스트 적응 이진 산술 코딩 (CABAC) 동작들과 같은, 엔트로피 인코딩 동작들을 이들 신택스 엘리먼트들 중 일부에 적용할 수도 있다. 컨텍스트 적응 가변 길이 코딩 (CAVLC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩, 또는 다른 이진 산술 코딩과 같은 다른 엔트로피 코딩 기법들이 또한 사용될 수 있다.

[0057] 비디오 인코더 (20) 에 의해 발생된 비트스트림은 일련의 네트워크 추상화 계층 (NAL) 유닛들을 포함할 수도 있다. NAL 유닛들의 각각은 그 NAL 유닛에 데이터의 형태의 표시 및 그 데이터를 포함하는 바이트들을 포함하는 신택스 구조일 수도 있다. 예를 들어, NAL 유닛은 비디오 파라미터 세트, 시퀀스 파라미터 세트, 픽처 파라미터 세트, 코딩된 슬라이스, SEI, 액세스 유닛 구분문자, 필러 데이터, 또는 또다른 유형의 데이터를 나타내는 데이터를 포함할 수도 있다. NAL 유닛에서의 데이터는 여러 신택스 구조들을 포함할 수도 있다.

[0058] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 발생된 비트스트림을 수신할 수도 있다. 비트스트림은 비디오 인코더 (20) 에 의해 인코딩된 비디오 데이터의 코딩된 표현을 포함할 수도 있다. 비디오 디코더 (30) 가 비트스트림을 수신할 때, 비디오 디코더 (30) 는 비트스트림에 관해 파싱 동작을 수행할 수도 있다.

비디오 디코더 (30) 가 파싱 동작을 수행할 때, 비디오 디코더 (30) 는 비트스트림으로부터 신택스 엘리먼트들을 추출할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여, 비디오 데이터의 픽처들을 복원할 수도 있다. 신택스 엘리먼트들에 기초하여 비디오 데이터를 복원하는 프로세스는 일반적으로 신택스 엘리먼트들을 발생시키기 위해 비디오 인코더 (20) 에 의해 수행되는 프로세스와 반대일 수도 있다.

[0059] 비디오 디코더 (30) 가 CU 와 연관되는 신택스 엘리먼트들을 추출한 후, 비디오 디코더 (30) 는 신택스 엘리먼트들에 기초하여 CU 의 PU들에 대한 예측된 비디오 블록들을 발생시킬 수도 있다. 게다가, 비디오 디코더 (30) 는 CU 의 TU들과 연관되는 변환 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 변환 계수 블록들에 대해 역변환들을 수행하여, CU 의 TU들과 연관되는 잔차 비디오 블록들을 복원할 수도 있다. 예측된 비디오 블록들을 발생시키고 잔차 비디오 블록들을 복원한 후, 비디오 디코더 (30) 는 예측된 비디오 블

록들 및 잔차 비디오 블록들에 기초하여, CU 의 비디오 블록을 복원할 수도 있다. 이러한 방법으로, 비디오 디코더 (30) 는 비트스트림에서의 신택스 엘리먼트들에 기초하여, CU들의 비디오 블록들을 복원할 수도 있다.

[0060] **비디오 인코더**

[0061] 도 2 는 본 개시물에서 설명하는 양태들에 따른 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 의 예를 예시하는 블록도이다. 비디오 인코더 (20) 는 예컨대, HEVC 에 있어 비디오 프레임의 단일 계층을 프로세싱하도록 구성될 수도 있다. 비디오 인코더 (20) 는 본 개시물의 기법들 중 임의의 기법 또는 모두를 수행하도록 구성될 수도 있다. 일부 예들에서, 본 개시물에서 설명하는 기법들은 비디오 인코더 (20) 의 여러 구성요소들 사이에 공유될 수도 있다. 일부 예들에서, 추가적으로 또는 대안적으로, 프로세서 (미도시) 는 본 개시물에서 설명되는 기법들 중 임의의 기법 또는 모두를 수행하도록 구성될 수도 있다.

[0062] 설명의 목적을 위해, 본 개시물은 HEVC 코딩의 상황에서 비디오 인코더 (20) 를 기술한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다. 도 2 에 도시된 예는 단일 계층 코덱에 대한 것이다. 그러나, 어떤 실시형태들에서, 비디오 인코더 (20) 중 일부 또는 모두는 멀티-계층 코덱의 프로세싱을 위해 중복될 수도 있다.

[0063] 비디오 인코더 (20) 는 비디오 슬라이스들 내 비디오 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라 코딩은 주어진 비디오 프레임 또는 픽처 내 비디오에서 공간 리던던시를 감소시키거나 또는 제거하기 위해, 공간 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접 프레임들 또는 픽처들 내 비디오에서 시간 리던던시를 감소시키거나 또는 제거하기 위해, 시간 예측에 의존한다. 인트라-모드 (I 모드) 는 여러 공간 기반의 압축 모드들 중 임의의 코딩 모드를 참조할 수도 있다. 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 과 같은 인터-모드들은 여러 시간-기반의 코딩 모드들 중 임의의 모드를 참조할 수도 있다.

[0064] 도 2 의 예에서, 비디오 인코더 (20) 는 복수의 기능적 구성요소들을 포함한다. 비디오 인코더 (20) 의 기능적 구성요소들은 예측 프로세싱 유닛 (100), 잔차 발생 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역양자화 유닛 (108), 역변환 유닛 (110), 복원 유닛 (112), 필터 유닛 (113), 디코딩된 픽처 버퍼 (114), 및 엔트로피 인코딩 유닛 (116) 을 포함한다. 예측 프로세싱 유닛 (100) 은 인터 예측 유닛 (121), 모션 추정 유닛 (122), 모션 보상 유닛 (124), 인트라 예측 유닛 (126), 및 인터 예측 예측 유닛 (128) 을 포함한다. 다른 예들에서, 비디오 인코더 (20) 는 더 많거나, 더 적거나, 또는 상이한 기능적 구성요소들을 포함할 수도 있다. 더욱이, 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 은 고도로 통합될 수도 있지만, 도 2 의 예에서는 설명의 목적들을 위해 별개로 나타낸다.

[0065] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20) 는 여러 소스들로부터 비디오 데이터를 수신할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 (예컨대, 도 1a 또는 도 1b 에 도시된) 비디오 소스 (18) 또는 또 다른 소스로부터 비디오 데이터를 수신할 수도 있다. 비디오 데이터는 일련의 픽처들을 나타낼 수도 있다. 비디오 데이터를 인코딩하기 위해, 비디오 인코더 (20) 는 픽처들의 각각에 대해 인코딩 동작을 수행할 수도 있다. 픽처에 대해 인코딩 동작을 수행하는 것의 일부로서, 비디오 인코더 (20) 는 그 픽처의 각각의 슬라이스에 대해 인코딩 동작들을 수행할 수도 있다. 슬라이스에 대해 인코딩 동작을 수행하는 것의 일부로서, 비디오 인코더 (20) 는 그 슬라이스에서의 트리블록들에 대해 인코딩 동작들을 수행할 수도 있다.

[0066] 트리블록에 대해 인코딩 동작을 수행하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 비디오 블록을 계속해서 더 작은 비디오 블록들로 분할하기 위해 트리블록의 비디오 블록에 관해 쿼드트리 파티셔닝을 수행할 수도 있다. 더 작은 비디오 블록들의 각각은 상이한 CU 와 연관될 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100) 은 트리블록의 비디오 블록을 4개의 동일-사이즈로된 서브-블록들로 파티셔닝할 수도 있으며, 서브-블록들 중 하나 이상을 4개의 동일-사이즈로된 서브-서브-블록들로 파티셔닝할 수도 있으며, 기타 등등으로 파티셔닝할 수도 있다.

[0067] CU들과 연관되는 비디오 블록들의 사이즈는 8x8 샘플들로부터 64x64 샘플들 이상의 최대치를 갖는 트리블록의 사이즈까지 이를 수도 있다. 본 개시물에서, "NxN" 및 "N 곱하기 N" 은 수직 및 수평 치수들의 관점에서 비디오 블록의 샘플 치수들, 예컨대, 16x16 샘플들 또는 16 곱하기 16 샘플들을 지칭하기 위해 상호교환가능하게 사용될 수도 있다. 일반적으로, 16x16 비디오 블록은 수직 방향으로 16개의 샘플들 (y = 16) 및 수평 방향으로 16개의 샘플들 (x = 16) 을 갖는다. 이와 유사하게, NxN 블록은 일반적으로 수직 방향으로 N 개의 샘플들 및 수평 방향으로 N 개의 샘플들을 가지며, 여기서 N 은 음이 아닌 정수 값을 나타낸다.

[0068] 더욱이, 트리블록에 대해 인코딩 동작을 수행하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 그 트리블록에 대한 계층적 쿼드트리 데이터 구조를 발생시킬 수도 있다. 예를 들어, 트리블록은 쿼드트리 데이터 구조의 루트 노드에 대응할 수도 있다. 예측 프로세싱 유닛 (100) 이 트리블록의 비디오 블록을 4개의 서브-블록들로 파티셔닝하면, 그 루트 노드는 쿼드트리 데이터 구조에서 4개의 자식 노드들을 갖는다. 자식 노드들 각각은 서브-블록들 중 하나와 연관되는 CU 에 대응한다. 예측 프로세싱 유닛 (100) 이 서브-블록들 중 하나를 4개의 서브-서브-블록들로 파티셔닝하면, 서브-블록들과 연관되는 CU 에 대응하는 노드는 4개의 자식 노드들을 가질 수도 있으며, 그 자식 노드들 각각은 서브-서브-블록들 중 하나와 연관되는 CU 에 대응한다.

[0069] 쿼드트리 데이터 구조의 각각의 노드는 대응하는 트리블록 또는 CU 에 대한 신택스 데이터 (예컨대, 신택스 엘리먼트들) 를 포함할 수도 있다. 예를 들어, 쿼드트리에서의 노드는 노드에 대응하는 CU 의 비디오 블록이 4개의 서브-블록들로 파티셔닝되는지 (예컨대, 분할되는지) 여부를 나타내는 분할 플래그를 포함할 수도 있다. CU 에 대한 신택스 엘리먼트들은 재귀적으로 정의될 수도 있으며, CU 의 비디오 블록이 서브-블록들로 분할되는지에 의존할 수도 있다. 비디오 블록이 파티셔닝되지 않는 CU 는 쿼드트리 데이터 구조에서 리프 노드에 대응할 수도 있다. 코딩된 트리블록은 대응하는 트리블록에 대한 쿼드트리 데이터 구조에 기초한 데이터를 포함할 수도 있다.

[0070] 비디오 인코더 (20) 는 트리블록의 각각의 비-파티셔닝된 CU 에 대해 인코딩 동작들을 수행할 수도 있다. 비디오 인코더 (20) 가 비-파티셔닝된 CU 에 대해 인코딩 동작을 수행할 때, 비디오 인코더 (20) 는 비-파티셔닝된 CU 의 인코딩된 표현을 나타내는 데이터를 발생시킨다.

[0071] CU 에 대해 인코딩 동작을 수행하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 CU 의 하나 이상의 PU들 사이에 CU 의 비디오 블록을 파티셔닝할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 여러 PU 사이즈들을 지원할 수도 있다. 특정의 CU 의 사이즈가 2Nx2N 이라고 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 2Nx2N 또는 NxN 의 PU 사이즈들, 및 2Nx2N, 2NxN, Nx2N, NxN, 2NxnU, nLx2N, nRx2N, 또는 유사한 것의 대칭적인 PU 사이즈들에서의 인터-예측을 지원한다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한 2NxnU, 2NxnD, nLx2N, 및 nRx2N 의 PU 사이즈들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (100) 은 기하학적 파티셔닝을 수행하여, CU 의 비디오 블록을, CU 의 PU들 사이에, CU 의 비디오 블록의 측면들과 직각으로 만나지 않는 경계를 따라서 파티셔닝할 수도 있다.

[0072] 인터 예측 유닛 (121) 은 CU 의 각각의 PU 에 대해 인터 예측을 수행할 수도 있다. 인터 예측은 시간 압축을 제공할 수도 있다. PU 에 대해 인터 예측을 수행하기 위해, 모션 추정 유닛 (122) 은 PU 에 대한 모션 정보를 발생시킬 수도 있다. 모션 보상 유닛 (124) 은 CU 와 연관되는 픽처 (예컨대, 참조 픽처들) 이외의 픽처들의 모션 정보 및 디코딩된 샘플들에 기초하여, PU 에 대한 예측된 비디오 블록을 발생시킬 수도 있다. 본 개시물에서, 모션 보상 유닛 (124) 에 의해 발생되는 예측된 비디오 블록은 인터-예측된 비디오 블록으로서 지칭될 수도 있다.

[0073] 슬라이스들은 I 슬라이스들, P 슬라이스들, 또는 B 슬라이스들일 수도 있다. 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 은 PU 가 I 슬라이스, P 슬라이스, 또는 B 슬라이스 내에 있는지에 따라서, CU 의 PU 에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU들은 인트라 예측된다. 그러므로, PU 가 I 슬라이스에 있으면, 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 은 PU 에 대해 인터 예측을 수행하지 않는다.

[0074] PU 가 P 슬라이스에 있으면, PU 를 포함하는 픽처는 "리스트 0" 으로서 지칭되는 참조 픽처들의 리스트와 연관된다. 리스트 0 에서의 참조 픽처들의 각각은 다른 픽처들의 인터 예측에 사용될 수도 있는 샘플들을 포함한다. 모션 추정 유닛 (122) 이 P 슬라이스에서의 PU 에 대해 모션 추정 동작을 수행할 때, 모션 추정 유닛 (122) 은 PU 에 대한 참조 블록에 대해 리스트 0 에서의 참조 픽처들을 탐색할 수도 있다. PU 의 참조 블록은 PU 의 비디오 블록에서의 샘플들에 가장 가깝게 대응하는 샘플들의 세트, 예컨대, 샘플들의 블록일 수도 있다. 모션 추정 유닛 (122) 은 다양한 메트릭들을 이용하여, 얼마나 참조 픽처에서의 샘플들의 세트가 PU 의 비디오 블록에서의 샘플들에 가깝게 대응하는지를 결정할 수도 있다. 예를 들어, 모션 추정 유닛 (122) 은 SAD (sum of absolute difference), SSD (sum of square difference), 또는 다른 차이 메트릭들에 의해 얼마나 참조 픽처에서의 샘플들의 세트가 PU 의 비디오 블록에서의 샘플들에 가깝게 대응하는지를 결정할 수도 있다.

[0075] P 슬라이스에서 PU 의 참조 블록을 식별한 후, 모션 추정 유닛 (122) 은 그 참조 블록을 포함하는 리스트 0 에서의 참조 픽처를 나타내는 참조 인덱스 및 PU 와 참조 블록 사이의 공간 변위를 나타내는 모션 벡터를 발생시킬 수도 있다. 여러 예들에서, 모션 추정 유닛 (122) 은 다양한 정밀도의 정도들에 대해 모션 벡터들을 발

생시킬 수도 있다. 예를 들어, 모션 추정 유닛 (122) 은 모션 벡터들을 1/4 샘플 정밀도, 1/8 샘플 정밀도, 또는 다른 분수 샘플 정밀도로 발생시킬 수도 있다. 분수 샘플 정밀도의 경우, 참조 블록 값들은 참조 픽처에서의 정수-위치 샘플 값들로부터 내삽될 수도 있다. 모션 추정 유닛 (122) 은 참조 인덱스 및 모션 벡터를 PU 의 모션 정보로서 출력할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 정보에 의해 식별되는 참조 블록에 기초하여, PU 의 예측된 비디오 블록을 발생시킬 수도 있다.

[0076] PU 가 B 슬라이스에 있으면, PU 를 포함하는 픽처는 "리스트 0" 및 "리스트 1" 로 지칭되는, 참조 픽처들의 2개의 리스트들과 연관될 수도 있다. 일부 예들에서, B 슬라이스를 포함하는 픽처는 리스트 0 과 리스트 1 의 조합인 리스트 조합과 연관될 수도 있다.

[0077] 더욱이, PU 가 B 슬라이스에 있으면, 모션 추정 유닛 (122) 은 PU 에 대한 단방향 예측 또는 양방향 예측을 수행할 수도 있다. 모션 추정 유닛 (122) 이 PU 에 대해 단방향 예측을 수행할 때, 모션 추정 유닛 (122) 은 PU 에 대한 참조 블록에 대해 리스트 0 또는 리스트 1 의 참조 픽처들을 탐색할 수도 있다. 모션 추정 유닛 (122) 은 그후 참조 블록을 포함하는 리스트 0 또는 리스트 1 에서의 참조 픽처를 나타내는 참조 인덱스 및 PU 와 참조 블록 사이의 공간 변위를 나타내는 모션 벡터를 발생시킬 수도 있다. 모션 추정 유닛 (122) 은 참조 인덱스, 예측 방향 표시자, 및 모션 벡터를 PU 의 모션 정보로서 출력할 수도 있다. 예측 방향 표시자는 참조 인덱스가 리스트 0 또는 리스트 1 에서 참조 픽처를 나타내는지를 표시할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 정보에 의해 표시되는 참조 블록에 기초하여, PU 의 예측된 비디오 블록을 발생시킬 수도 있다.

[0078] 모션 추정 유닛 (122) 이 PU 에 대해 양방향 예측을 수행할 때, 모션 추정 유닛 (122) 은 PU 에 대한 참조 블록에 대해서 리스트 0 에서의 참조 픽처들을 탐색할 수도 있으며, 또한 PU 에 대한 또 다른 참조 블록에 대해서 리스트 1 에서의 참조 픽처들을 탐색할 수도 있다. 모션 추정 유닛 (122) 은 그후 참조 블록들을 포함하는 리스트 0 및 리스트 1 에서 참조 픽처들을 나타내는 참조 인덱스들 및 참조 블록들과 PU 사이의 공간 변위들을 나타내는 모션 벡터들을 발생시킬 수도 있다. 모션 추정 유닛 (122) 은 PU 의 참조 인덱스들 및 모션 벡터들을 PU 의 모션 정보로서 출력할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 정보에 의해 표시되는 참조 블록들에 기초하여, PU 의 예측된 비디오 블록을 발생시킬 수도 있다.

[0079] 일부의 경우, 모션 추정 유닛 (122) 은 PU 에 대한 모션 정보의 전체 세트를 엔트로피 인코딩 유닛 (116) 으로 출력하지 않는다. 대신, 모션 추정 유닛 (122) 은 또 다른 PU 의 모션 정보를 참조하여 PU 의 모션 정보를 시그널링할 수도 있다. 예를 들어, 모션 추정 유닛 (122) 은 PU 의 모션 정보가 이웃하는 PU 의 모션 정보와 충분히 유사하다고 결정할 수도 있다. 이 예에서, 모션 추정 유닛 (122) 은 PU 과 연관되는 신택스 구조에서, PU 가 이웃하는 PU 와 동일한 모션 정보를 갖는, 비디오 디코더 (30) 에 표시되는, 값을 표시할 수도 있다. 또 다른 예에서, 모션 추정 유닛 (122) 은 PU 와 연관되는 신택스 구조에서, 이웃하는 PU 및 모션 벡터 차이 (MVD) 를 식별할 수도 있다. 모션 벡터 차이는 PU 의 모션 벡터와 표시된 이웃하는 PU 의 모션 벡터 사이의 차이를 표시한다. 비디오 디코더 (30) 는 표시된 이웃하는 PU 의 모션 벡터 및 모션 벡터 차이를 이용하여, PU 의 모션 벡터를 결정할 수도 있다. 제 2 PU 의 모션 정보를 시그널링할 때 제 1 PU 의 모션 정보를 참조함으로써, 비디오 인코더 (20) 는 더 적은 비트들을 이용하여 제 2 PU 의 모션 정보를 시그널링가능할 수도 있다.

[0080] CU 에 대해 인코딩 동작을 수행하는 것의 일부로서, 인트라 예측 유닛 (126) 은 CU 의 PU들에 대해 인트라 예측을 수행할 수도 있다. 인트라 예측은 공간 압축을 제공할 수도 있다. 인트라 예측 유닛 (126) 이 PU 에 대해 인트라 예측을 수행할 때, 인트라 예측 유닛 (126) 은 동일한 픽처에서의 다른 PU들의 디코딩된 샘플들에 기초하여 PU 에 대한 예측 데이터를 발생시킬 수도 있다. PU 에 대한 예측 데이터는 예측된 비디오 블록 및 여러 신택스 엘리먼트들을 포함할 수도 있다. 인트라 예측 유닛 (126) 은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서의 PU들에 대해 인트라 예측을 수행할 수도 있다.

[0081] PU 상에 인트라 예측을 수행하기 위해, 인트라 예측 유닛 (126) 은 다수의 인트라 예측 모드들을 이용하여 예측 PU 에 대한 다수의 데이터의 세트들을 발생시킬 수도 있다. 인트라 예측 유닛 (126) 이 인트라 예측 모드를 이용하여 PU 에 대한 예측 데이터의 세트를 발생시킬 때, 인트라 예측 유닛 (126) 은 샘플들을 이웃하는 PU들의 비디오 블록들로부터, PU 의 비디오 블록을 가로질러, 인트라 예측 모드와 연관되는 방향 및/또는 기울기로 확장할 수도 있다. PU들, CU들, 및 트리블록들에 대해 좌우, 상하 인코딩 순서를 가정하면, 이웃하는 PU들은 PU 의 상측에, 우상부에, 좌상부에, 또는 좌측에 있을 수도 있다. 인트라 예측 유닛 (126) 은 PU 의 사이즈에 따라서, 다수의 인트라 예측 모드들, 예컨대, 33개의 방향 인트라 예측 모드들을 이용할 수도 있다.

[0082] 예측 프로세싱 유닛 (100) 은 PU 에 대해 모션 보상 유닛 (124) 에 의해 발생된 예측 데이터 또는 PU 에 대해 인트라 예측 유닛 (126) 에 의해 발생된 예측 데이터 중에서, PU 에 대한 예측 데이터를 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (100) 은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여 PU 에 대한 예측 데이터를 선택한다.

[0083] 예측 프로세싱 유닛 (100) 이 인트라 예측 유닛 (126) 에 의해 발생된 예측 데이터를 선택하면, 예측 프로세싱 유닛 (100) 은 PU 에 대한 예측 데이터들, 예컨대, 선택된 인트라 예측 모드를 발생하는데 사용된 인트라 예측 모드를 시그널링할 수도 있다. 예측 프로세싱 유닛 (100) 은 그 선택된 인트라 예측 모드를 여러 방법들로 시그널링할 수도 있다. 예를 들어, 선택된 인트라 예측 모드가 이웃하는 PU 의 인트라 예측 모드와 동일할 가능성이 있을 수도 있다. 다시 말해서, 이웃하는 PU 의 인트라 예측 모드는 현재의 PU 에 대해 가장 가능성있는 모드일 수도 있다. 따라서, 예측 프로세싱 유닛 (100) 은 선택된 인트라 예측 모드가 이웃하는 PU 의 인트라 예측 모드와 동일하다는 것을 표시하는 신택스 엘리먼트를 발생시킬 수도 있다.

[0084] 위에서 설명한 바와 같이, 비디오 인코더 (20) 는 인터-계층 예측 유닛 (128) 을 포함할 수도 있다. 인터-계층 예측 유닛 (128) 은 SHVC 에 이용가능한 하나 이상의 상이한 계층들 (예컨대, 기초 또는 참조 계층) 을 이용하여 현재의 블록 (예컨대, EL 에서의 현재의 블록) 을 예측하도록 구성된다. 이러한 예측은 인터-계층 예측으로서 지칭될 수도 있다. 인터-계층 예측 유닛 (128) 은 예측 방법들을 이용하여 인터-계층 리던던시를 감소시켜, 코딩 효율을 향상시키고 컴퓨터 리소스 요구사항들을 감소시킨다. 인터-계층 예측의 일부 예들은 인터-계층 인트라 예측, 인터-계층 모션 예측, 및 인터-계층 잔차 예측을 포함한다. 인터-계층 인트라 예측은 기초 계층에서 병치된 블록들의 복원을 이용하여 향상 계층에서의 현재의 블록을 예측한다. 인터-계층 모션 예측은 기초 계층의 모션 정보를 이용하여 향상 계층에서의 모션을 예측한다. 인터-계층 잔차 예측은 기초 계층의 잔차를 이용하여 향상 계층의 잔차를 예측한다.

[0085] 예측 프로세싱 유닛 (100) 이 CU 의 PU들에 대한 예측 데이터를 선택한 후, 잔차 발생 유닛 (102) 은 CU 의 비디오 블록으로부터 CU 의 PU들의 예측된 비디오 블록들을 감산함으로써 (예컨대, 마이너스 부호로 표시됨으로써) CU 에 대한 잔차 데이터를 발생시킬 수도 있다. CU 의 잔차 데이터는 CU 의 비디오 블록에서의 샘플들의 상이한 샘플 구성요소들에 대응하는 2D 잔차 비디오 블록들을 포함할 수도 있다. 예를 들어, 잔차 데이터는 CU 의 PU들의 예측된 비디오 블록들에서의 샘플들의 휘도 성분들과, CU 의 원래 비디오 블록에서의 샘플들의 휘도 성분들 사이의 차이들에 대응하는 잔차 비디오 블록을 포함할 수도 있다. 게다가, CU 의 잔차 데이터는 CU 의 PU들의 예측된 비디오 블록들에서의 샘플들의 색차 성분들과, CU 의 원래 비디오 블록에서의 샘플들의 색차 성분들 사이의 차이들에 대응하는 잔차 비디오 블록들을 포함할 수도 있다.

[0086] 예측 프로세싱 유닛 (100) 은 쿼드트리 파티셔닝을 수행하여, CU 의 잔차 비디오 블록들을 서브-블록들로 파티셔닝할 수도 있다. 각각의 미분할된 잔차 비디오 블록은 CU 의 상이한 TU 와 연관될 수도 있다. CU 의 TU들과 연관되는 잔차 비디오 블록들의 사이즈들 및 위치들은 CU 의 PU들과 연관되는 비디오 블록들의 사이즈들 및 위치들에 기초하거나 또는 기초하지 않을 수도 있다. "잔차 쿼드 트리" (RQT) 로서 알려진 쿼드트리 구조는 잔차 비디오 블록들의 각각과 연관되는 노드들을 포함할 수도 있다. CU 의 TU들은 RQT 의 리프 노드들에 대응할 수도 있다.

[0087] 변환 프로세싱 유닛 (104) 은 하나 이상의 변환들을 TU 와 연관되는 잔차 비디오 블록에 적용함으로써 CU 의 각각의 TU 에 대한 하나 이상의 변환 계수 블록들을 발생시킬 수도 있다. 변환 계수 블록들의 각각은 변환 계수들의 2D 매트릭스일 수도 있다. 변환 프로세싱 유닛 (104) 은 여러 변환들을 TU 와 연관되는 잔차 비디오 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (104) 은 이산 코사인 변환 (DCT), 방향 변환, 또는 개념적으로 유사한 변환을 TU 와 연관되는 잔차 비디오 블록에 적용할 수도 있다.

[0088] 변환 프로세싱 유닛 (104) 이 TU 와 연관되는 변환 계수 블록을 발생시킨 후, 양자화 유닛 (106) 은 변환 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 유닛 (106) 은 CU 와 연관되는 QP 값에 기초하여, CU 의 TU 와 연관되는 변환 계수 블록을 양자화할 수도 있다.

[0089] 비디오 인코더 (20) 는 QP 값을 CU 와 여러 방법들로 연관시킬 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 와 연관되는 트리블록에 관해 레이트-왜곡 분석을 수행할 수도 있다. 레이트-왜곡 분석에서, 비디오 인코더 (20) 는 트리블록에 대해 인코딩 동작을 여러번 수행함으로써, 트리블록의 다수의 코딩된 표현들을 발생시킬 수도 있다. 비디오 인코더 (20) 는 비디오 인코더 (20) 가 트리블록의 상이한 인코딩된 표현들을 발생시킬 때, 상이한 QP 값들을 CU 와 연관시킬 수도 있다. 비디오 인코더 (20) 는 주어진 QP 값이 최저 비트레이트 및 왜곡 메트릭을 갖는 트리블록의 코딩된 표현에서 CU 와 연관될 때 주어진 QP 값이 CU 와 연관된다는 것

을 시그널링할 수도 있다.

[0090] 역양자화 유닛 (108) 및 역변환 유닛 (110) 은 역양자화 및 역변환들을 변환 계수 블록에 각각 적용하여, 변환 계수 블록으로부터 잔차 비디오 블록을 복원할 수도 있다. 복원 유닛 (112) 은 복원된 잔차 비디오 블록을 예측 프로세싱 유닛 (100) 에 의해 발생된 하나 이상의 예측된 비디오 블록들로부터의 대응하는 샘플들에 가산하여, TU 와 연관되는 복원된 비디오 블록을 발생시킬 수도 있다. CU 의 각각의 TU 에 대한 비디오 블록들을 이 방법으로 복원함으로써, 비디오 인코더 (20) 는 CU 의 비디오 블록을 복원할 수도 있다.

[0091] 복원 유닛 (112) 이 CU 의 비디오 블록을 복원한 후, 필터 유닛 (113) 은 CU 와 연관되는 비디오 블록에서 블록킹 아티팩트들을 감소시키기 위해 디블록킹 동작을 수행할 수도 있다. 하나 이상의 디블록킹 동작들을 수행한 후, 필터 유닛 (113) 은 CU 의 복원된 비디오 블록을 디코딩된 픽처 버퍼 (114) 에 저장할 수도 있다. 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 은 복원된 비디오 블록을 포함하는 참조 픽처를 이용하여, 후속 픽처들의 PU들에 관해 인터 예측을 수행할 수도 있다. 게다가, 인트라 예측 유닛 (126) 은 디코딩된 픽처 버퍼 (114) 에서의 복원된 비디오 블록들을 이용하여, CU 와 동일한 픽처에서의 다른 PU들에 대해 인트라 예측을 수행할 수도 있다.

[0092] 엔트로피 인코딩 유닛 (116) 은 비디오 인코더 (20) 의 다른 기능적 구성요소들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (116) 은 양자화 유닛 (106) 으로부터 변환 계수 블록들을 수신할 수도 있으며 예측 프로세싱 유닛 (100) 으로부터 신택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (116) 이 데이터를 수신할 때, 엔트로피 인코딩 유닛 (116) 은 하나 이상의 엔트로피 인코딩 동작들을 수행하여, 엔트로피 인코딩된 데이터를 발생시킬 수도 있다. 예를 들어, 비디오 인코더 (20) 는 그 데이터에 대해, CAVLC 동작, CABAC 동작, 변수-대-변수 (V2V) 길이 코딩 동작, 신택스-기반 컨텍스트-적응 이진 산술 코딩 (SBAC) 동작, 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 동작, 또는 또다른 유형의 엔트로피 인코딩 동작을 수행할 수도 있다. 엔트로피 인코딩 유닛 (116) 은 엔트로피 인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다.

[0093] 데이터에 대해 엔트로피 인코딩 동작을 수행하는 것의 일부로서, 엔트로피 인코딩 유닛 (116) 은 컨텍스트 모델을 선택할 수도 있다. 엔트로피 인코딩 유닛 (116) 이 CABAC 동작을 수행하고 있으면, 컨텍스트 모델은 특정의 값들을 가질 특정의 빈들의 확률들의 추정치들을 나타낼 수도 있다. CABAC 의 상황에서, 용어 "빈" 은 신택스 엘리먼트의 이진화된 버전의 비트를 지칭하기 위해 사용된다.

[0094] **비디오 디코더**

[0095] 도 3 은 본 개시물에서 설명하는 양태들에 따른 기법들을 구현할 수도 있는 예시적인 비디오 디코더 (30) 의 예를 예시하는 블록도이다. 비디오 디코더 (30) 는 예컨대, HEVC 에 있어 비디오 프레임의 단일 계층을 프로세싱하도록 구성될 수도 있다. 게다가, 비디오 디코더 (30) 는 본 개시물의 기법들 중 임의의 기법 또는 모두를 수행하도록 구성될 수도 있다. 일부 예들에서, 본 개시물에서 설명하는 기법들은 비디오 디코더 (30) 의 여러 구성요소들 사이에 공유될 수도 있다. 일부 예들에서, 추가적으로 또는 대안적으로, 프로세서 (미도시) 는 본 개시물에서 설명되는 기법들 중 임의의 기법 또는 모두를 수행하도록 구성될 수도 있다.

[0096] 설명의 목적을 위해, 본 개시물은 HEVC 코딩의 상황에서 비디오 디코더 (30) 를 기술한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다. 도 3 에 도시된 예는 단일 계층 코덱에 대한 것이다. 그러나, 어떤 구현예들에서, 비디오 디코더 (30) 중 일부 또는 모두는 멀티-계층 코덱의 프로세싱을 위해 중복될 수도 있다.

[0097] 도 3 의 예에서, 비디오 디코더 (30) 는 복수의 기능적 구성요소들을 포함한다. 비디오 디코더 (30) 의 기능적 구성요소들은 엔트로피 디코딩 유닛 (150), 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 유닛 (156), 복원 유닛 (158), 필터 유닛 (159), 및 디코딩된 픽처 버퍼 (160) 를 포함한다. 예측 프로세싱 유닛 (152) 은 모션 보상 유닛 (162), 인트라 예측 유닛 (164), 및 인터-계층 예측 유닛 (166) 을 포함한다. 일부 예들에서, 비디오 디코더 (30) 는 도 2 의 비디오 인코더 (20) 에 대해 설명된 인코딩 과정과는 일반적으로 반대인 디코딩 과정을 수행할 수도 있다. 다른 예들에서, 비디오 디코더 (30) 는 더 많거나, 더 적거나, 또는 상이한 기능적 구성요소들을 포함할 수도 있다.

[0098] 비디오 디코더 (30) 는 인코딩된 비디오 데이터를 포함하는 비트스트림을 수신할 수도 있다. 비트스트림은 복수의 신택스 엘리먼트들을 포함할 수도 있다. 비디오 디코더 (30) 가 비트스트림을 수신할 때, 엔트로피 디코딩 유닛 (150) 은 비트스트림에 대해 파싱 동작을 수행할 수도 있다. 비트스트림에 대해 파싱 동작을

수행하는 것의 결과로서, 엔트로피 디코딩 유닛 (150) 은 비트스트림으로부터 신택스 엘리먼트들을 추출할 수도 있다. 파싱 동작을 수행하는 것의 일부로서, 엔트로피 디코딩 유닛 (150) 은 비트스트림에서의 엔트로피 인코딩된 신택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 유닛 (156), 복원 유닛 (158), 및 필터 유닛 (159) 은 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여, 디코딩된 비디오 데이터를 발생하는 복원 동작을 수행할 수도 있다.

[0099] 위에서 설명한 바와 같이, 비트스트림은 일련의 NAL 유닛들을 포함할 수도 있다. 비트스트림의 NAL 유닛들은 비디오 파라미터 세트 NAL 유닛, 시퀀스 파라미터 세트 NAL 유닛들, 픽처 파라미터 세트 NAL 유닛들, SEI NAL 유닛들 등을 포함할 수도 있다. 비트스트림에 대해 파싱 동작을 수행하는 것의 일부로서, 엔트로피 디코딩 유닛 (150) 은 시퀀스 파라미터 세트 NAL 유닛들로부터 시퀀스 파라미터 세트들을, 픽처 파라미터 세트 NAL 유닛들로부터 픽처 파라미터 세트들을, SEI NAL 유닛들로부터 SEI 데이터를, 기타 등등으로 추출하여 엔트로피 디코딩하는 파싱 동작들을 수행할 수도 있다.

[0100] 게다가, 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림에 대해 파싱 동작을 수행하는 것의 일부로서, 엔트로피 디코딩 유닛 (150) 은 코딩된 슬라이스 NAL 유닛들로부터 코딩된 슬라이스들을 추출하여 엔트로피 디코딩하는 파싱 동작들을 수행할 수도 있다. 코딩된 슬라이스들 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 관련된 신택스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더에서의 신택스 엘리먼트들은 그 슬라이스를 포함하는 픽처와 연관된 픽처 파라미터 세트를 식별하는 신택스 엘리먼트를 포함할 수도 있다. 엔트로피 디코딩 유닛 (150) 은 슬라이스 헤더를 복원하기 위해, 코딩된 슬라이스 헤더에서의 신택스 엘리먼트들에 대해 CABAC 디코딩 동작들과 같은, 엔트로피 디코딩 동작들을 수행할 수도 있다.

[0101] 코딩된 슬라이스 NAL 유닛들로부터 슬라이스 데이터를 추출하는 것의 일부로서, 엔트로피 디코딩 유닛 (150) 은 슬라이스 데이터에서의 코딩된 CU들로부터 신택스 엘리먼트들을 추출하는 파싱 동작들을 수행할 수도 있다. 추출된 신택스 엘리먼트들은 변환 계수 블록들과 연관되는 신택스 엘리먼트들을 포함할 수도 있다. 엔트로피 디코딩 유닛 (150) 은 그후 신택스 엘리먼트들 중 일부에 대해 CABAC 디코딩 동작들을 수행할 수도 있다.

[0102] 엔트로피 디코딩 유닛 (150) 이 비-파티셔닝된 CU 에 대해 파싱 동작을 수행한 후, 비디오 디코더 (30) 는 비-파티셔닝된 CU 에 대해 복원 동작을 수행할 수도 있다. 비-파티셔닝된 CU 에 대해 복원 동작을 수행하기 위해, 비디오 디코더 (30) 는 CU 의 각각의 TU 에 대해 복원 동작을 수행할 수도 있다. CU 의 각각의 TU 에 대해 복원 동작을 수행함으로써, 비디오 디코더 (30) 는 CU 와 연관되는 잔차 비디오 블록을 복원할 수도 있다.

[0103] TU 에 대해 복원 동작을 수행하는 것의 일부로서, 역양자화 유닛 (154) 은 TU 와 연관되는 변환 계수 블록을 역양자화할 수도 있다, 예컨대, 양자화 해제할 수도 있다. 역양자화 유닛 (154) 은 변환 계수 블록을 HEVC 에 대해 제안되거나 또는 H.264 디코딩 표준에 의해 정의되는 역양자화 프로세스들과 유사한 방법으로 역양자화할 수도 있다. 역양자화 유닛 (154) 은 변환 계수 블록의 CU 에 대해 비디오 인코더 (20) 에 의해 계산된 양자화 파라미터 QP 를 이용하여, 양자화의 정도 및, 이와 유사하게, 적용할 역양자화 유닛 (154) 에 대한 역양자화의 정도를 결정할 수도 있다.

[0104] 역양자화 유닛 (154) 이 변환 계수 블록을 역양자화한 후, 역변환 유닛 (156) 은 변환 계수 블록과 연관되는 TU 에 대한 잔차 비디오 블록을 발생시킬 수도 있다. 역변환 유닛 (156) 은 TU 에 대한 잔차 비디오 블록을 발생시키기 위해 역변환을 변환 계수 블록에 적용할 수도 있다. 예를 들어, 역변환 유닛 (156) 은 역 DCT, 역 정수 변환, 역 Karhunen-Loeve 변환 (KLT), 역 회전 변환, 역 방향 변환, 또는 또 다른 역변환을 변환 계수 블록에 적용할 수도 있다. 일부 예들에서, 역변환 유닛 (156) 은 비디오 인코더 (20) 로부터의 시그널링에 기초하여, 변환 계수 블록에 적용할 역변환을 결정할 수도 있다. 이러한 예들에서, 역변환 유닛 (156) 은 변환 계수 블록과 연관되는 트리블록에 대한 쿼드트리의 루트 노드에서의 시그널링된 변환에 기초하여, 역변환을 결정할 수도 있다. 다른 예들에서, 역변환 유닛 (156) 은 하나 이상의 코딩 특성들, 예컨대 블록 사이즈, 코딩 모드 등으로부터 역변환을 추론할 수도 있다. 일부 예들에서, 역변환 유닛 (156) 은 캐스케이드된 역변환을 적용할 수도 있다.

[0105] 일부 예들에서, 모션 보상 유닛 (162) 은 내삽 필터들에 기초하여 내삽을 수행함으로써, PU 의 예측된 비디오 블록을 개선할 수도 있다. 서브-샘플 정밀도로 모션 보상에 사용되는 내삽 필터들에 대한 식별자들이 신택스 엘리먼트들에 포함될 수도 있다. 모션 보상 유닛 (162) 은 PU 의 예측된 비디오 블록의 발생 동안 비디오 인코더 (20) 에 의해 사용된 동일한 내삽 필터들을 이용하여 참조 블록의 서브-정수 샘플들에 대해 내삽된 값들을 계산할 수도 있다. 모션 보상 유닛 (162) 은 수신된 신택스 정보에 따라 비디오 인코더 (20) 에 의

해 사용되는 내삽 필터들을 결정하고 그 내삽 필터들을 이용하여 예측 비디오 블록을 발생시킬 수도 있다.

[0106] PU 가 인트라 예측을 이용하여 인코딩되면, 인트라 예측 유닛 (164) 은 인트라 예측을 수행하여, PU 에 대한 예측된 비디오 블록을 발생시킬 수도 있다. 예를 들어, 인트라 예측 유닛 (164) 은 비트스트림에서의 신택스 엘리먼트들에 기초하여, 그 PU 에 대한 인트라 예측 모드를 결정할 수도 있다. 비트스트림은 인트라 예측 유닛 (164) 이 PU 의 인트라 예측 모드를 결정하는데 사용될 수도 있는 신택스 엘리먼트들을 포함할 수도 있다.

[0107] 일부의 경우, 신택스 엘리먼트들은 인트라 예측 유닛 (164) 이 현재의 PU 의 인트라 예측 모드를 결정하기 위해 또 다른 PU 의 인트라 예측 모드를 이용해야 한다는 것을 표시할 수도 있다. 예를 들어, 현재의 PU 의 인트라 예측 모드가 이웃하는 PU 의 인트라 예측 모드와 동일할 가능성이 있을 수도 있다. 다시 말해서, 이웃하는 PU 의 인트라 예측 모드는 현재의 PU 에 대해 가장 가능성있는 모드일 수도 있다. 그러므로, 이 예에서, 비트스트림은 PU 의 인트라 예측 모드가 이웃하는 PU 의 인트라 예측 모드와 동일하다는 것을 나타내는 작은 신택스 엘리먼트를 포함할 수도 있다. 인트라 예측 유닛 (164) 은 그후 공간적으로 이웃하는 PU들의 비디오 블록들에 기초하여, PU 에 대한 예측 데이터 (예컨대, 예측된 샘플들) 을 발생시키기 위해, 인트라 예측 모드를 이용할 수도 있다.

[0108] 위에서 설명한 바와 같이, 비디오 디코더 (30) 는 또한 인터-계층 예측 유닛 (166) 을 포함할 수도 있다. 인터-계층 예측 유닛 (166) 은 SHVC 에 이용가능한 하나 이상의 상이한 계층들 (예컨대, 기초 또는 참조 계층) 을 이용하여 현재의 블록 (예컨대, 향상 계층에서의 현재의 블록) 을 예측하도록 구성된다. 이러한 예측은 인터-계층 예측으로서 지칭될 수도 있다. 인터-계층 예측 유닛 (166) 은 예측 방법들을 이용하여 인터-계층 리던던시를 감소시켜, 코딩 효율을 향상시키고 컴퓨터 리소스 요구사항들을 감소시킨다. 인터-계층 예측의 일부 예들은 인터-계층 인트라 예측, 인터-계층 모션 예측, 및 인터-계층 잔차 예측을 포함한다. 인터-계층 인트라 예측은 기초 계층에서 병치된 블록들의 복원을 이용하여 향상 계층에서의 현재의 블록을 예측한다. 인터-계층 모션 예측은 기초 계층의 모션 정보를 이용하여 향상 계층에서의 모션을 예측한다. 인터-계층 잔차 예측은 기초 계층의 잔차를 이용하여 향상 계층의 잔차를 예측한다. 인터-계층 예측 방식들의 각각이 더욱더 자세하게 아래에서 설명된다.

[0109] 복원 유닛 (158) 은 CU 의 TU들과 연관되는 잔차 비디오 블록들 및 CU 의 PU들의 예측된 비디오 블록들, 예컨대, 적용가능한 경우, 인트라-예측 데이터 또는 인터-예측 데이터를 이용하여, CU 의 비디오 블록을 복원할 수도 있다. 따라서, 비디오 디코더 (30) 는 비트스트림에서의 신택스 엘리먼트들에 기초하여, 예측된 비디오 블록 및 잔차 비디오 블록을 발생시킬 수도 있으며 그 예측된 비디오 블록 및 잔차 비디오 블록에 기초하여 비디오 블록을 발생시킬 수도 있다.

[0110] 복원 유닛 (158) 이 CU 의 비디오 블록을 복원한 후, 필터 유닛 (159) 은 CU 와 연관되는 블록킹 아티팩트들을 감소시키기 위해 디블록킹 동작을 수행할 수도 있다. 필터 유닛 (159) 이 CU 와 연관되는 블록킹 아티팩트들을 감소시키기 위해 디블록킹 동작을 수행한 후, 비디오 디코더 (30) 는 CU 의 비디오 블록을 디코딩된 픽처 버퍼 (160) 에 저장할 수도 있다. 디코딩된 픽처 버퍼 (160) 는 후속 모션 보상, 인트라 예측, 및 도 1a 또는 도 1b 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에의 프리젠테이션을 위해, 참조 픽처들을 제공할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 디코딩된 픽처 버퍼 (160) 에서의 비디오 블록들에 기초하여, 다른 CU들의 PU들에 관해 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다.

[0111] **팔레트 코딩 모드**

[0112] 상이한 코딩 유닛들 사이의 리던던시를 주로 제거하는 종래의 인트라 및 인터 예측과는 대조적으로, 팔레트 코딩은 코딩 유닛 내 반복적인 픽셀 값들/패턴들의 리던던시를 목표로 한다. 팔레트 코딩 모드에서, 픽셀 값들을 테이블 인덱스들 (또한, 팔레트 인덱스들로 불림) 에 맵핑하는 팔레트로서 지칭되는 룩업 테이블이 먼저 시그널링된다. 일부 구현예들에서, 팔레트는 규정된 최대 사이즈 (예컨대, 32 픽셀 값들) 를 갖는다. 팔레트는 블록 샘플들에 대한 예측자들로서 또는 최종 복원된 블록 샘플들로서 사용될 수 있는 칼라 성분 (예컨대, RGB, YUV, 등) 값들 또는 강도들을 나타내는 테이블 인덱스들로 넘버링된 엔트리들을 포함한다. 일부 구현예들에서, 팔레트 블록에서의 샘플들은 3개의 런-모드들, 즉 '좌측-복사 모드' (또는, 런 모드), '상부-복사 모드', 및 '이스케이프 모드' (또는, 픽셀 모드) 를 이용하여 코딩된다.

[0113] 좌측-복사 모드에서 코딩되는 팔레트 블록에서의 위치에 대해, 팔레트 인덱스가 먼저 시그널링되고 뒤이어서, "run_length" (또는, "palette_run") (예컨대, M) 이 시그널링된다. 팔레트 블록에서의 현재의 위치 및 다음 M 개의 위치들이 현재의 위치에 대해 시그널링되는 동일한 팔레트 인덱스를 가지기 때문에, 팔레트 블록에서

의 현재의 위치 및 다음 M 개의 위치들에 대해 어떤 추가 정보도 시그널링되도록 요구되지 않는다. 팔레트 인덱스 (예컨대, i) 는 모든 3개의 칼라 성분들에 의해 공유되는데, 이것은 복원된 픽셀 값들이 (칼라 공간이 YUV 라 가정하면) (Y, U, V) = ($palette_Y$[i], $palette_U$[i], $palette_V$[i]) 임을 의미한다.

[0114] 상부-복사 모드에서 코딩되는 팔레트 블록에서의 위치에 대해, 값 "run_length" (또는, "copy_run") (예컨대, N) 가 팔레트 블록에서 다음 N 개의 위치들 (현재의 위치를 포함하여, 전체에서 N + 1 개의 위치들) 에 대해, 팔레트 인덱스가 팔레트 블록에서 바로 상부에 있는 위치의 팔레트 인덱스와 동일하다는 것을 표시하기 위해, 시그널링된다.

[0115] 이스케이프 모드 (또는, 픽셀 모드) 에서 코딩되는 팔레트 블록에서의 위치에 대해, 팔레트 블록에서의 현재의 위치에 대응하는 픽셀 값이 시그널링된다. 이스케이프 모드는 이스케이프 플래그 (예컨대, 1 의 플래그 값은 현재의 위치가 이스케이프 모드에서 코딩된다는 것을 표시한다) 또는 팔레트 인덱스 (예컨대, 팔레트 엔트리들 중 임의의 팔레트 엔트리에 대응하지 않는 인덱스 값 또는 팔레트 사이즈보다 크거나 또는 동일한 인덱스 값) 를 이용하여 시그널링될 수도 있다.

[0116] **팔레트 비트스트림**

[0117] 기존 구현예들에서, 팔레트 비트스트림 (예컨대, 팔레트 코딩 모드에서 코딩된 코딩 유닛들을 포함하는 비트스트림) 은 다음과 같이 편성된다:

[0118]

palette_entries
palette_index_map

[0119] **테이블 1. 팔레트 모드 비트스트림**

[0120] palette_entries 는 테이블 인덱스에 각각 맵핑되는 하나 이상의 픽셀 값들을 포함한다. 예를 들어, 주어진 코딩 유닛이 3개의 고유한 픽셀 값들 (예컨대, 적색, 녹색, 및 청색) 을 포함하면, 팔레트 엔트리들은 3개의 엔트리들, (0, 적색), (1, 녹색), 및 (2, 청색) 을 포함할 수도 있다. palette_index_map 은 팔레트 엔트리들을 이용하여 코딩된 하나 이상의 팔레트 블록들을 포함하며, 여기서, 팔레트 테이블 인덱스들 (예컨대, 상기 예에서 0, 1, 및 2) 은 팔레트 블록에서의 픽셀 값들을 표시하기 위해 사용된다.

[0121] 도 4 는 입력 CU (410) 의 예시적인 구성, 인덱스 블록 (420), 이스케이프 픽셀 (430), 및 팔레트 (440) 를 예시한다. 도 4 에 나타낸 바와 같이, 입력 CU (410) 는 3개의 고유한 픽셀 값들: 백색, 그레이, 및 블랙을 포함한다. 백색 및 그레이의 빈도에 기초하여, 단지 백색 및 그레이 픽셀 값들만이 팔레트 (440) 에 포함되며, 여기서, 0 의 인덱스 값은 백색 픽셀 값과 연관되고 1 의 인덱스 값은 그레이 픽셀 값과 연관된다. 팔레트에 포함되지 않은 블랙 픽셀 값은 팔레트와 독립적으로 코딩되는 이스케이프 픽셀 (430) 로서 라벨링된다. 도 4 에 나타낸 바와 같이, 인덱스 블록 (420) 은 블록에서의 각각의 위치에 대한 인덱스 값을 포함한다. 인덱스 블록 (420) 에서의 2개의 위치들이 이스케이프 모드에서와 같이 (예컨대, 팔레트 인덱스들 0 또는 1 을 참조함이 없이) 코딩된다. 단지 단일 이스케이프 픽셀 및 단지 2개의 팔레트 엔트리들만이 도 4 의 예에서 사용되지만, 본 출원의 실시형태들은 이에 한정되지 않으며, 임의 개수의 이스케이프 픽셀들 및 팔레트 엔트리들이 사용될 수도 있다. 일부 실시형태들에서, 팔레트 사이즈는 32 개의 엔트리들로 제한되며, 32 개의 엔트리들 중 하나와 연관되지 않은 임의의 픽셀 값들이 이스케이프 픽셀들이 된다. 최대 팔레트 사이즈는 임의의 개수로 설정될 수도 있다. 또, CU 사이즈는 8 픽셀들 곱하기 8 픽셀들에 제한되지 않으며, 16x16 또는 임의의 다른 사이즈일 수도 있다.

[0122] **팔레트 비트스트림의 예시적인 신택스**

[0123] 일부 구현예들에서, 팔레트 인덱스 맵에서, 팔레트 코딩 모드에서 코딩된 블록은 비트스트림에서 다음 형태를 취할 수도 있다:

while (not end) {
run_mode_flag
if (run_mode_flag == COPY_ABOVE)
run_length
else if (run_mode_flag == COPY_LEFT) {
index
if (index == ESCAPE_INDEX)
escape_pixel_value
else
run_length
}

[0124]

[0125] **테이블 2. palette_index_map 비트스트림 (디폴트 설정)**

[0126] 테이블 2 에 예시된 예에서, run_mode_flag 의 값에 의존하여, 상이한 신택스 엘리먼트들이 비트스트림으로 시그널링된다. 팔레트 블록에서의 현재의 위치가 상부-복사 모드에서 코딩된다는 것을 런 모드 플래그가 표시하면, 런 바디는 런 길이 값 (상기 "run_length" 의 제 1 인스턴스) 을 포함한다. 팔레트 블록에서의 현재의 위치가 '좌측-복사' 모드에서 코딩된다는 것을 런 모드 플래그가 표시하면, 런 바디는 인덱스 값 ("인덱스") 과 뒤이어서 런 길이 값 (상기 "run_length" 의 제 2 인스턴스) 을 포함하며, 인덱스 값이 이스케이프 인덱스에 대응하지 않으면, 이 경우, 양자화된 이스케이프 픽셀 값들 ("escape_pixel_value") 이 시그널링된다.

[0127] 대안적인 구현예에서, 명시적인 이스케이프 플래그가 사용된다. 좀더 구체적으로, 팔레트 인덱스 맵은 비트스트림에서 다음 형태를 취할 수도 있다:

while (not end) {
escape_flag
if (escape_flag)
escape_pixel_value
else {
run_mode_flag
if (run_mode_flag == COPY_LEFT)
index
run_length
}

[0128]

[0129] **테이블 3. palette_index_map 비트스트림 (대안 설정)**

[0130] 테이블 3 에 예시된 예에서, escape_flag 의 값에 의존하여, 상이한 신택스 엘리먼트들이 비트스트림으로 시그널링된다. 이스케이프 플래그가 1 의 값을 가지면, 런 바디는 양자화된 이스케이프 픽셀 값들을 포함한다. 이스케이프 플래그가 0 의 값을 가지면, 런 모드 플래그가 '상부 복사' 모드와 '좌측 복사' 모드를 구별하기 위해 시그널링된다. 팔레트 블록에서의 현재의 위치가 '좌측-복사' 모드에서 코딩된다는 것을 런 모드 플래그가 표시하면, 비트스트림은 인덱스 값과, 뒤이어서, 런 길이 값을 포함한다. 그렇지 않으면, 단지 런 길이 값만이 비트스트림으로 시그널링된다.

[0131] 도 5 는 본 개시물의 양태들에 따른, 비트스트림에서의 비-자연 비디오 데이터를 코딩하는 방법 (500) 을 예시

하는 플로우차트이다. 도 5 에 예시된 단계들은 비디오 인코더 (예컨대, 비디오 인코더 (20)), 비디오 디코더 (예컨대, 비디오 디코더 (30)), 또는 임의의 다른 구성요소에 의해 수행될 수도 있다. 편의를 위해, 방법 (500) 은 비디오 인코더 (20), 비디오 디코더 (30), 또는 다른 구성요소일 수도 있는 비디오 코더 (또한, 코더로서 간단히 지칭됨) 에 의해 수행되는 것으로 설명된다.

[0132] 방법 (500) 은 블록 (501) 에서 시작한다. 블록 (505) 에서, 코더는 CU 와 연관된 팔레트에 기초하여 복수의 픽셀들을 가지는 코딩 유닛 (CU) 을 프로세싱한다. 위에서 설명한 바와 같이, 인코더는 CU 의 콘텐츠에 기초하여 팔레트를 유도하고, 디코더가 CU 와 연관된 팔레트를 이용하여 CU 를 프로세싱할 수 있도록, 비트스트림에서 팔레트를 시그널링할 수도 있다. 팔레트는 인덱스 값과 각각 연관된 복수의 팔레트 엔트리들 및 픽셀 값을 포함할 수도 있다. 픽셀 값은 CU 에서의 픽셀들 중 하나와 연관될 수도 있다. 일부 실시형태들에서, 각각의 팔레트 엔트리는 CU 에서 발견되는 고유한 픽셀 값과 연관된다. 블록 (505) 은 도 6 내지 도 8 을 참조하여 설명되는 하나 이상의 단계들 및/또는 방법들을 포함할 수도 있다. 이 방법은 블록 (510) 에서 종료한다.

[0133] **바이패스 (Bypass) 빈들을 그룹화하기**

[0134] H.264, HEVC, 및 많은 다른 현대의 비디오 코딩 표준들에서, 신택스 엘리먼트에 대해, 이진화 후, 0/1 빈 스트림이 컨텍스트 적응적 이진 산술 코더 (CABAC) 에 제공되며, 여기서, ("컨텍스트" 로 지칭되는) 확률 모델이 비-고정 확률 분포를 추적하도록 적응적으로 선택되어 업데이트된다. 특별한 경우로서, 확률 모델은 엔트로피 코더의 처리량을 향상시키기 위해 업데이트되지 않을 수도 있다. 컨텍스트 업데이트 없이, 이러한 단순화된 방법을 이용하여 코딩된 빈들은 바이패스 빈들로서 지칭된다.

[0135] 테이블 2 및 테이블 3 의 예들에서, 비트스트림에 일부 리던던시들이 있을 수도 있다. 이들 리던던시들은 어떤 조건들이 만족될 때 어떤 신택스 엘리먼트들을 시그널링하는 것을 스킵함으로써 제거될 수도 있다. 게다가, 일부 신택스 엘리먼트들은 파싱 의존성을 도입할 수도 있다. 예를 들어, 테이블 2 에서, 신택스 엘리먼트 run_mode_flag 는 현재의 픽셀이 블록의 제 1 라인에 있으면, 디코더가 런 모드를 인덱스 복사 모드 (예컨대, 좌측 복사 모드) 인 것으로 추론할 수도 있기 때문에, 시그널링될 필요가 없을 수도 있다. 게다가, 테이블 2 의 예에서, 디코더는 인덱스 값을 먼저 디코딩하고, 디코딩된 인덱스 값에 의존하여, 디코더는 (예컨대, 인덱스 값이 이스케이프 인덱스 값을 나타내는지 여부에 기초하여) 그 모드가 인덱스 복사 모드 또는 이스케이프 모드인지 여부를 결정한다. 디코더가 그 모드를 인덱스 복사 모드인 것으로 결정하면, 디코더 파서는 런 길이를 계속 파싱한다. 디코더가 그 모드를 이스케이프 모드인 것으로 결정하면, 디코더 파서는 이스케이프 값들 및/또는 런 길이를 계속 파싱할 수도 있다. 그 파서들이 대개 디코더들보다 아주 더 높은 속도에서 동작하기 때문에, 디코딩 엔진과 파싱 엔진 사이의 이러한 의존성은 (예컨대, 디코딩 엔진이 파싱된 비트들을 디코딩하기를 파싱 엔진이 대기해야 할 수도 있기 때문에) 파서의 처리량에 영향을 미칠 수도 있다. 따라서, 팔레트 코딩 모드에서 코딩된 블록들을 프로세싱하는 향상된 방법이 요망된다. 본 출원에서, 팔레트 모드에서의 파싱 의존성을 피하거나 또는 감소시키기 위해 비트스트림에서의 팔레트 엘리먼트들을 편성하는 여러 신규한 방법들이 설명된다.

[0136] **실시형태 #1: 양자화된 이스케이프 픽셀 값들을 팔레트 모드 블록의 끝에 배치하기**

[0137] 일부 실시형태들에서, 모든 양자화된 이스케이프 픽셀 값들이 비트스트림에서의 팔레트 모드 블록의 끝에서 시그널링된다. 이러한 실시형태들에서, 엔트로피 코더 리셋팅이 (인덱스, 런-길이) 코딩 후 적용될 수도 있다. 예를 들어, 블록에서 가능한 (인덱스, 런-길이) 쌍들 모두를 코딩한 후, 산술 코딩 엔진의 ivlCurrRange 변수 (예컨대, 현재의 산술 코딩 간격의 범위를 규정하는 변수) 는 256 으로 설정된다. 이 방법에 의해, 디코더는 비트스트림으로부터 비트들을 판독하고, CABAC 코더를 호출할 필요 없이, 그들을 있는 그대로 처리할 수도 있다. 변수를 256 으로 리셋하는 이 프로시저가 없다면, 컨텍스트는 업데이트될 필요가 없을 수도 있지만, 디코더는 이진 결정들을 행하기 위해 CABAC 코더를 여전히 호출해야 할 수도 있다. 따라서, 양자화된 이스케이프 픽셀 값들은 모든 (인덱스, 런-길이) 쌍들을 파싱 및/또는 디코딩한 후 병렬로 파싱될 수 있다. 일 실시형태에서, 이스케이프 픽셀들이 고정된 길이 코드를 이용하여 코딩되면, 이스케이프 픽셀들은 인덱스-런 블록을 파싱한 후 병렬로 파싱되어 디코딩될 수 있다.

[0138] 다른 실시형태에서, 이스케이프 픽셀들이 트렁케이트된 이진 코드를 이용하여 코딩되면, 이스케이프 픽셀의 각각의 칼라 성분은 그의 양자화된 강도에 의존하여 'k' 또는 'k + 1' 비트들을 취할 수도 있다. 예를 들어, 트렁케이트된 이진 인코딩에서, 값 X 를 가지는 신택스 엘리먼트에 대해, 그의 최대 가능한 값 Max 가 알려져

있고 그리고 $2^k \leq n < 2^{k+1}$ 인 경우 n = Max + 1 및 k = floor($\log_2(n)$) 이라고 가정하고 그리고 u = 2k + 1 - n 라 하자, X < u 이면, 트렁케이트된 이진 코드워드는 길이 k 를 가지는 X 의 이진 표현으로 규정된다. 그렇지 않으면, 트렁케이트된 이진 코드워드는 길이 k + 1 을 가지는 X + u 의 이진 표현으로 규정된다. 이러한 일 실시형태에서, 현재의 블록에서의 모든 이스케이프 픽셀들에 대한 각각의 칼라 성분의 제 1 'k' 비트들이 함께 그룹화되고, 뒤이어서, 옵션적인 (k + 1)번째 비트가 그룹화될 수도 있다. 이러한 편성에 의해, 모든 이스케이프 픽셀들에 대한 각각의 칼라 성분의 제 1 'k' 비트들이 병렬로 파싱되고 디코딩될 수 있다. 일부 의존성이 옵션적인 (k + 1)번째 비트를 파싱할 때에 여전히 존재할 수도 있다.

[0139] 도 6 은 본 개시물의 양태들에 따른, 비트스트림에서의 비-자연 비디오 데이터를 디코딩하는 방법 (600) 을 예시하는 플로우차트이다. 도 6 에 예시된 단계들은 비디오 디코더 (예컨대, 비디오 디코더 (30)) 또는 임의의 다른 구성요소에 의해 수행될 수도 있다. 편의를 위해, 방법 (600) 은 비디오 디코더 (30) 또는 다른 구성요소일 수도 있는, 비디오 코더 (또한, 코더로서 간단히 지칭됨) 에 의해 수행되는 것으로 설명된다.

[0140] 방법 (600) 은 블록 (601) 에서 시작한다. 블록 (605) 에서, 코더는 비트스트림에 제공된 CU 와 연관된 팔레트를 파싱한다. 팔레트는 인덱스 값과 각각 연관된 복수의 팔레트 엔트리들 및 인덱스 값과 연관된 픽셀 값을 포함할 수도 있다. 팔레트의 예시적인 실시형태가 도 4 에 예시된다.

[0141] 블록 (610) 에서, 코더는 CU 와 연관된 하나 이상의 런 길이들을 파싱한다. 위에서 설명한 바와 같이, 각각의 런 길이는 좌측-복사 모드 또는 상부-복사 모드와 연관되는, CU 에서의 현재의 위치를 포함하여 시작하는, 연속된 위치들의 개수를 표시한다.

[0142] 블록 (615) 에서, 코더는 CU 와 연관된 하나 이상의 인덱스 값들을 파싱한다. 위에서 설명한 바와 같이, 각각의 인덱스 값은 CU 에서의 현재의 위치와 연관되는 팔레트에서의 픽셀 값을 표시한다. 도 4 의 예에서, 0 의 인덱스 값은 CU 에서의 현재의 위치가 백색 픽셀 값을 갖는다는 것을 표시하며, 1 의 인덱스 값은 CU 에서의 현재의 위치가 그레이 픽셀 값을 갖는다는 것을 표시한다.

[0143] 블록 (620) 에서, 코더는 CU 와 연관된 하나 이상의 이스케이프 픽셀 값들을 파싱한다. 위에서 설명한 바와 같이, 각각의 이스케이프 픽셀 값은 CU 와 연관된 팔레트에 있지 않은 픽셀 값을 표시한다. 도 4 의 예에서, 블랙 픽셀 값들을 가지는 CU 에서의 2개의 위치들은 이스케이프 모드에서 코딩되며, 코더는 블랙 픽셀 값들을 비트스트림에서 이스케이프 픽셀 값들로서 시그널링한다. 일부 실시형태들에서, 이스케이프 픽셀 값들은 비트스트림에서의 연속된 위치들로부터 (예컨대, CU 와 연관된 비트스트림의 부분의 끝에서) 파싱된다. 예를 들어, 이스케이프 픽셀 값들의 연속된 위치들이 CU 와 연관된 런 길이들 및 인덱스 값들 모두 후의 비트스트림에서 나타난다. 이러한 실시형태들에서, 런 길이들 및 인덱스 값들 모두가 파싱된 후, 이스케이프 픽셀 값들은 병렬로 프로세싱될 (예컨대, 파싱될) 수 있다. 블록 (625) 에서, 코더는 파싱된 팔레트, 파싱된 런 길이들, 파싱된 인덱스 값들, 및 파싱된 이스케이프 값들에 기초하여 CU 를 디코딩한다. 이 방법은 블록 (630) 에서 종료한다.

[0144] 방법 (600) 에서, 도 6 에 도시된 블록들 중 하나 이상이 제거될 (예컨대, 수행되지 않을) 수도 있으며 및/또는 본 방법이 수행되는 순서는 스위칭될 수도 있다. 예를 들어, 블록 (610) 및 블록 (615) 은 CU 와 연관된 각각의 런 길이 및 인덱스 값 쌍을 파싱하기 위해 함께 수행될 수도 있다. 일부 실시형태들에서, 추가적인 블록들이 방법 (600) 에 추가될 수도 있다. 본 개시물의 실시형태들은 도 6 에 나타낸 예에 또는 그에 의해 한정되지 않으며, 다른 변형예들이 본 개시물의 정신으로부터 일탈함이 없이 구현될 수도 있다.

[0145] **실시형태 #2: 인덱스 값들을 팔레트 모드 블록의 끝에 배치하기**

[0146] 일부 실시형태들에서, 모든 인덱스 값들은 비트스트림에서 팔레트 모드 블록의 끝에서 시그널링된다. 일 실시형태에서, 모든 양자화된 이스케이프 값들은 비트스트림에서 팔레트 모드 블록의 끝에서 시그널링되며, 뒤이어서, 인덱스 값들의 모든 발생들의 그룹이 시그널링된다. 다른 실시형태에서, 모든 인덱스 값들은 비트스트림에서 양자화된 이스케이프 값들 직전에 시그널링된다.

[0147] 엔트로피 코더 리셋팅 (resetting) 이 런-길이 코딩 후 적용될 수도 있다. 예를 들어, 블록에서 가능한 런-길이들 모두를 코딩한 후, 산술 코딩 엔진의 ivlCurrRange 변수 (예컨대, 현재의 산술 코딩 간격의 범위를 규정하는 변수) 는 256 으로 설정된다. 따라서, 인덱스 값들 및/또는 이스케이프 값들은 팔레트 블록에서의 모든 런-길이들을 파싱하고 및/또는 디코딩한 후 병렬로 파싱될 수 있다. 일 실시형태에서, 인덱스 값들이 고정된 길이 코드를 이용하여 코딩되면, 인덱스 값들은 런-길이 블록을 파싱한 후 병렬로 파싱되어 디코딩될 수

있다.

[0148] 다른 실시형태에서, 인덱스 값들이 트렁케이트된 이진 코드를 이용하여 코딩되면, 인덱스들은 그의 값에 의존하여 'k' 또는 'k + 1' 비트들을 취할 수도 있다. 이러한 일 실시형태에서, 현재의 블록에서의 모든 인덱스 값들 및/또는 이스케이프 픽셀들에 대한 각각의 칼라 성분의 제 1 'k' 비트들이 함께 그룹화될 수도 있으며, 뒤이어서, 옵션적인 (k + 1)번째 비트가 그룹화될 수도 있다. 이러한 편성에 의해, 현재의 블록에서의 인덱스 값들 및/또는 이스케이프 값들 모두의 제 1 'k' 비트들이 병렬로 파싱되고 디코딩될 수 있다. 일부 의존성이 옵션적인 (k + 1)번째 비트를 파싱할 때에 여전히 존재할 수도 있다.

[0149] 도 7 은 본 개시물의 양태들에 따른, 비트스트림에서의 비-자연 비디오 데이터를 디코딩하는 방법 (700) 을 예시하는 플로우차트이다. 도 7 에 예시된 단계들은 비디오 디코더 (예컨대, 비디오 디코더 (30)) 또는 임의의 다른 구성요소에 의해 수행될 수도 있다. 편의를 위해, 방법 (700) 은 비디오 디코더 (30) 또는 다른 구성요소일 수도 있는, 비디오 코더 (또한, 코더로서 간단히 지칭됨) 에 의해 수행되는 것으로 설명된다.

[0150] 방법 (700) 은 블록 (701) 에서 시작한다. 블록 (705) 에서, 코더는 비트스트림에 제공된 CU 와 연관된 팔레트를 파싱한다. 팔레트는 인덱스 값과 각각 연관된 복수의 팔레트 엔트리들 및 인덱스 값과 연관된 픽셀 값을 포함할 수도 있다. 팔레트의 예시적인 실시형태가 도 4 에 예시된다.

[0151] 블록 (710) 에서, 코더는 CU 와 연관된 하나 이상의 런 길이들을 파싱한다. 위에서 설명한 바와 같이, 각각의 런 길이는 좌측-복사 모드 또는 상부-복사 모드와 연관되는, CU 에서의 현재의 위치를 포함하여 시작하는, 연속된 위치들의 개수를 표시한다.

[0152] 블록 (715) 에서, 코더는 CU 와 연관된 하나 이상의 인덱스 값들을 파싱한다. 위에서 설명한 바와 같이, 각각의 인덱스 값은 CU 에서의 현재의 위치와 연관되는 팔레트에서의 픽셀 값을 표시한다. 도 4 의 예에서, 0 의 인덱스 값은 CU 에서의 현재의 위치가 백색 픽셀 값을 갖는다는 것을 표시하며, 1 의 인덱스 값은 CU 에서의 현재의 위치가 그레이 픽셀 값을 갖는다는 것을 표시한다. 도 7 의 예에서, 인덱스 값들은 비트스트림에서의 연속된 위치들로부터 (예컨대, CU 와 연관된 런 길이들의 모두 후에) 파싱될 수도 있다. 이러한 실시형태들에서, 모든 런 길이들이 파싱된 후, 인덱스 값들은 병렬로 프로세싱될 (예컨대, 파싱될) 수 있다. 예를 들어, 인덱스 값들은 비트스트림에서 이스케이프 픽셀 값들 직전에 제공될 수도 있다.

[0153] 블록 (720) 에서, 코더는 CU 와 연관된 하나 이상의 이스케이프 픽셀 값들을 파싱한다. 위에서 설명한 바와 같이, 각각의 이스케이프 픽셀 값은 CU 와 연관된 팔레트에 있지 않은 픽셀 값을 표시한다. 도 4 의 예에서, 블랙 픽셀 값들을 가지는 CU 에서의 2개의 위치들은 이스케이프 모드에서 코딩되며, 코더는 블랙 픽셀 값들을 비트스트림에서 이스케이프 픽셀 값들로서 시그널링한다. 일부 실시형태들에서, 이스케이프 픽셀 값들은 비트스트림에서의 연속된 위치들로부터 (예컨대, CU 와 연관된 비트스트림의 부분의 끝에서) 파싱될 수도 있다. 예를 들어, 이스케이프 픽셀 값들의 연속된 위치들은 CU 와 연관된 런 길이들 및 인덱스 값들 모두의 후의 비트스트림에 나타날 수도 있다. 이러한 실시형태들에서, 런 길이들 및 인덱스 값들 모두가 파싱된 후, 이스케이프 픽셀 값들은 병렬로 프로세싱될 (예컨대, 파싱될) 수 있다. 블록 (725) 에서, 코더는 파싱된 팔레트, 파싱된 런 길이들, 파싱된 인덱스 값들, 및 파싱된 이스케이프 값들에 기초하여 CU 를 디코딩한다. 이 방법은 블록 (730) 에서 종료한다.

[0154] 방법 (700) 에서, 도 7 에 도시된 블록들 중 하나 이상이 제거될 (예컨대, 수행되지 않을) 수도 있으며 및/또는 본 방법이 수행되는 순서는 스위칭될 수도 있다. 일부 실시형태들에서, 추가적인 블록들이 방법 (700) 에 추가될 수도 있다. 본 개시물의 실시형태들은 도 7 에 나타낸 예에 또는 그에 의해 한정되지 않으며, 다른 변형예들이 본 개시물의 정신으로부터 일탈함이 없이 구현될 수도 있다.

[0155] **실시형태 #3: 인덱스 값 및 이스케이프 픽셀 값의 위치를 규정하기 위해 오프셋들을 이용하기**

[0156] 일부 실시형태들에서, 2개의 오프셋들이 비트스트림에서 팔레트 모드 블록의 처음에 시그널링될 수도 있으며, 여기서, 2개의 오프셋들은 예를 들어, S0 및 S1 로서 각각 표시된, 위에서 설명된 인덱스 값 그룹 및 이스케이프 픽셀 값 그룹의 시작 위치들을 규정한다. 인덱스 값 그룹이 이스케이프 픽셀 값 그룹보다 앞서는 경우, 단지 2개의 시작 위치들 사이의 delta 값만이 제 2 오프셋에 대해 시그널링될 수도 있다 (예컨대, 오프셋들 S0 및 S1-S0 이 시그널링될 수도 있다). 2개의 오프셋들은 여러 엔트로피 코딩 방법들, 예컨대, 트렁케이트된-Rice, 지수-Golomb, 트렁케이트된-이진수, 고정된 길이, 1진수, 트렁케이트된 1진수, 등을 이용하여 시그널링될 수 있다. 일부 실시형태들에서, 팔레트 모드 블록의 끝을 표시하는 오프셋 값 S2 가 또한 시그널링될 수도 있다. 대안적으로, 제 2 오프셋과 S2 사이의 delta 값이 시그널링될 수도 있다 (예컨대, S2-S1).

[0157] **바이패스 빈들을 그룹화하는 구현예**

[0158] 위에서 설명된 실시형태들 #1 및 #3 에 따르면, palette_index_map 비트스트림은 다음과 같이 수정될 수도 있다:

while (byte position != S0) {
run_mode_flag
if (run_mode_flag == COPY_ABOVE)
run_length
else if (run_mode_flag == COPY_LEFT) {
index
if (index != ESCAPE_INDEX)
run_length
}
while (not end of block) {
escape_pixel_value
}

[0159]

[0160] **테이블 4. 수정된 palette_index_map 비트스트림의 예**

[0161] 위에서 설명된 실시형태들 #1, #2, 및 #3 에 따르면, palette_index_map 비트스트림은 다음과 같이 수정될 수도 있다:

while (byte position != S0) {	
escape_and_run_mode_flag	/* '1' '01' '00' 일 수 있음 */
if (escape_and_run_mode_flag != 1)	/* 이스케이프 픽셀이 아님 */
run_length	
}	
while (byte position != S1) {	
index	
}	
while (byte position != S2) {	
escape_pixel_value	
}	

[0162]

[0163] **테이블 5. 예시적인 palette_index_map 비트스트림**

[0164] **인덱스 리던던시 체크**

[0165] 일부 실시형태들에서, 인덱스 값들을 코딩할 때, 리던던시 체크가 적용될 수도 있다. 예를 들어, (위치 'x - 1' 로서 표시되는) 래스터 스캔 순서에서의 이전 이웃하는 위치가 좌측-복사 런 모드의 끝이면, 현재의 인덱스 값은 그의 이전 이웃하는 위치의 인덱스 값과 동일하지 않을 수 있다. 다시 말해서, 위치 'x - 1' 이 유효하고 (예컨대, 현재의 블록 내에 있거나 또는 현재의 블록 외부에 있지만 예를 들어, 보더 패딩을 통해서, 결정론적인 값을 가지고) 좌측-복사 런의 끝이면, 위치 'x' 에 대한 인덱스 값은 위치 'x - 1' 에서의 인덱스 값과 동일하지 않을 수 있다 (경우 1). 그 이유는 2개의 인덱스 값들이 동일하였으면 그들이 더 긴 좌측-복사

런으로 병합되었을 것이라는 점이다. 다른 예에서, 래스터 스캔 순서에서의 이전 이웃하는 위치가 상부-복사 런 모드의 끝이면, 및/또는 현재의 위치의 상부 이웃이 이스케이프 픽셀이 아니라는 추가적인 제한 사항이 만족되면, 현재의 값은 그의 상부 이웃의 인덱스 값과 동일하지 않을 수 있다. 다시 말해서, 위치 'x - 1' 이 유효하고 상부-복사 런의 끝이면, 및/또는 위치 'x' 상부의 픽셀 값이 이스케이프 픽셀이 아니라는 추가적인 제한 사항이 만족되면, 위치 'x' 에 대한 인덱스 값은 그의 상부 이웃의 인덱스 값과 동일하지 않을 수 있다 (경우 2). 그 이유는 2개의 인덱스 값들이 동일하였으면 그들이 더 긴 상부-복사 런으로 병합되었을 것이라는 점이다. 따라서, 이들 예들은 인코더가 '가장 긴 가능한 런 (longest possible run)' 원리를 따른다고 가정한다. 이들 경우들 중 어느 경우든, 범위 (예컨대, 위에서 설명된 Max 값) 는 1 만큼 감소될 수 있으며, 비트 절감이 달성될 수도 있다.

[0166] 그러나, 경우 1 에 대해, 위치 'x' 에서의 인덱스 값의 디코딩은 (예컨대, 디코더가 위치 'x' 에서의 인덱스 값이 될 수 없는 것을 결정하기 위해 위치 'x - 1' 에서의 인덱스 값을 알아야 하기 때문에) 위치 'x - 1' 에서의 인덱스 값의 복원에 의존한다. 그러나, 위치 'x-1' 에 대한 인덱스 값은 위치 'x' 에서의 인덱스 값이 디코딩되고 있을 때까지 용이하게 이용가능하지 않을 수도 있다. 따라서, 이 의존성은 디코딩 프로세스에서 일부 지연을 야기할 수도 있다. 이 의존성을 제거하기 위해, 일부 실시형태들에서, 경우 1 에 대한 조건문 체크는 디스에이블될 수도 있다. 이러한 실시형태들의 일부, 경우 2 에 대한 조건문 체크가 (예컨대, 현재의 픽셀보다 위에 있는 위치의 인덱스 값이 좀더 이용가능성이 많기 때문에) 여전히 수행될 수도 있다. 이러한 실시형태들에서, 경우 1 에 대한 조건문 체크는 완전히 디스에이블된다. 대안적으로, 경우 1 에 대한 조건문 체크는 단지 특정의 경우에만 디스에이블될 수도 있다. 예를 들어, 경우 1 에 대한 조건문 체크는 오직 '제한된 런' 피쳐 (feature) 가 인에이블될 때에만 디스에이블될 수도 있으며, 여기서, 런 길이가 코딩되는 최대 팔레트 인덱스 값 (또는, 런 길이가 코딩되지 않는 최소 팔레트 인덱스 값) 이 표시된다.

[0167] 경우 2 에서, 이스케이프 픽셀이 좌측-복사 또는 상부-복사 런들로 허용되면, 위치 'x' 보다 위에 있는 픽셀이 이스케이프 픽셀인지 여부의 체킹이 제거될 수도 있다. 예를 들어, 이스케이프 픽셀들은 팔레트에 있지 않고 (예컨대, 마치 팔레트에서의 픽셀 값들처럼) 그들 자신의 런들을 가지는 하나 이상의 인덱스 값들을 할당받을 수도 있다. 이와 유사하게, 위치 'x' 의 좌측에 있는 픽셀이 이스케이프 픽셀인지 여부의 체킹 (예컨대, 이스케이프 픽셀들이 좌측으로부터 또는 상부로부터 복사되도록 주어진 구현예가 허용하지 않으면 위치 x 에서의 현재의 인덱스 값을 파싱하기 전에 수행되어야 할 수도 있는 단계) 은, 이스케이프 픽셀이 좌측-복사 또는 상부-복사 런들로 허용되면, 제거될 수 있다.

[0168] 도 8 은 본 개시물의 양태들에 따른, 비트스트림에서의 비-자연 비디오 데이터를 코딩하는 방법 (800) 을 예시하는 플로우차트이다. 도 8 에 예시된 단계들은 비디오 인코더 (예컨대, 비디오 인코더 (20)), 비디오 디코더 (예컨대, 비디오 디코더 (30)) 또는 임의의 다른 구성요소에 의해 수행될 수도 있다. 편의를 위해, 방법 (800) 은 비디오 인코더 (20), 비디오 디코더 (30), 또는 다른 구성요소일 수도 있는 비디오 코더 (또한, 코더로서 간단히 지칭됨) 에 의해 수행되는 것으로 설명된다.

[0169] 방법 (800) 은 블록 (801) 에서 시작한다. 블록 (805) 에서, 코더는 CU 에서 현재의 위치의 좌측에 있는 위치가 상부-복사 런의 끝과 연관된다고 결정한다. 위에서 설명한 바와 같이, CU 에서의 하나 이상의 위치들이 상부-복사 모드에서 코딩될 때, 상부-복사 모드와 연관된, CU 에서의 초기 위치를 포함하여 시작하는, 연속된 위치들의 개수를 표시하는 런 길이가 시그널링된다. 런 길이에 기초하여, 코더는 주어진 위치 (예컨대, CU 에서의 현재의 위치에 바로 선행하는 위치) 가 상부-복사 런의 끝이라고 결정할 수도 있다.

[0170] 블록 (810) 에서, 코더는, 현재의 위치의 좌측에 있는 위치가 상부-복사 런의 끝과 연관된다고 결정하는 것에 응답하여, 현재의 위치보다 위에 있는 위치가 이스케이프 픽셀 값과 연관되는지 여부를 결정함이 없이, 현재의 위치와 연관된 인덱스 값을 결정한다. 위에서 설명한 바와 같이, 일반적으로, 코더는 현재의 픽셀의 인덱스 값을 결정하기 전에, 현재의 위치보다 위에 있는 위치가 이스케이프 픽셀 값과 연관되는지 여부를 결정할 필요가 있다 (예컨대, 현재의 위치보다 위에 있는 위치가 이스케이프 픽셀과 연관되면, 현재의 위치의 인덱스 값이 현재의 위치보다 위에 있는 위치의 인덱스 값과 동일하지 않다는 가정이 부정확하게 될 수도 있다). 그러나, 본 개시물의 일부 실시형태들에서, 이스케이프 픽셀들은 좌측-복사 또는 상부-복사 런들의 부분일 수 있다. 따라서, 현재의 위치보다 위에 있는 위치에 대한 별개의 체크가 요구되지 않는다. 이 방법은 블록 (815) 에서 종료한다.

[0171] 방법 (800) 에서, 도 8 에 도시된 블록들 중 하나 이상이 제거될 (예컨대, 수행되지 않을) 수도 있으며 및/또는 본 방법이 수행되는 순서는 스위칭될 수도 있다. 일부 실시형태들에서, 추가적인 블록들이 방법 (800) 에

추가될 수도 있다. 본 개시물의 실시형태들은 도 8 에 나타낸 예에 또는 그에 의해 한정되지 않으며, 다른 변형예들이 본 개시물의 정신으로부터 일탈함이 없이 구현될 수도 있다.

[0172] **인덱스 빈들의 최대 개수를 상수로 유지하기**

[0173] 팔레트 모드 코딩의 일부 구현예들에서, 팔레트 인덱스들은 트렁케이트된 이진 코드를 이용하여 코딩된다. 현재의 CU 에서 가장 큰 인덱스가 N 이라고 (예컨대, 인덱스 값이 {0, 1, 2,..., N} 중에서 선택된다고) 가정하면, 각각의 인덱스를 코딩하기 위한 빈들의 개수는, 이들 2개의 값들이 동일하지 않으면, $\lfloor \log_2(N+1) \rfloor$ 또는 $\lceil \log_2(N+1) \rceil$ 일 수 있다. 이스케이프 픽셀이 가장 큰 인덱스를 할당받기 때문에 (예컨대, 팔레트에서의 모든 픽셀 값들이 그들의 인덱스 값들을 받은 후), 이스케이프 픽셀을 코딩하는 것은 $\lceil \log_2(N+1) \rceil$ 빈들을 취한다.

[0174] 일부의 경우, 위에서 설명된 방법들과 같은, 의존성들을 이용함으로써, 현재의 인덱스에 대한 가장 큰 심볼 값이 1 만큼 감소될 수도 있다. 다시 말해서, 이스케이프 픽셀은 리던던시 제거 조건이 인에이블되는지 여부에 의존하여, $\lceil \log_2 N \rceil$ 또는 $\lceil \log_2(N+1) \rceil$ 빈들을 취할 수도 있다. 그 결과, 디코더는 얼마나 많은 빈들이 인덱스를 디코딩하는데 필요로 하는지를 결정하기 위해 가장 큰 인덱스 심볼 값이 N 또는 N-1 인지 여부를 먼저 계산할 필요가 있을 수도 있다. 이것은 추가적인 온 칩 지연을 도입하였으며 디코더 파이프라이닝에 부정적인 영향을 미친다. 일부 실시형태들에서, 이 지연 및 디코더 파이프라이닝에 대한 임의의 부정적인 영향들을 제거하기 위해, 이 리던던시 제거 메커니즘은 제한될 수도 있다. 예를 들어, 인덱스 코딩에 사용되는 빈들의 최대 개수는 상수로 설정될 수도 있다. 일 예에서, 이스케이프 픽셀의 인덱스 값은 ceil(log2(N + 1)) 빈들을 항상 이용할 수도 있다. 다른 예에서, ceil(log2(N + 1)) 가 ceil(log2(N)) 과 동일하면, 이스케이프 픽셀들에 대한 리던던시 제거 프로시저가 인에이블되고, 그렇지 않으면, 이스케이프 픽셀들에 대한 리던던시 제거 프로시저가 디스에이블된다.

[0175] **인코더 측 플로우차트**

[0176] 도 9 는 본 개시물의 양태들에 따른, 비트스트림에서의 비디오 데이터를 인코딩하는 방법 (900) 을 예시하는 플로우차트이다. 예를 들어, 비디오 데이터는 컴퓨터-발생된 스크린 콘텐츠를 포함하는 비-자연 비디오 데이터일 수도 있다. 도 9 에 예시된 단계들은 비디오 인코더 (예컨대, 비디오 인코더 (20)) 또는 임의의 다른 구성요소에 의해 수행될 수도 있다. 편의를 위해, 방법 (900) 은 비디오 인코더 (20) 또는 다른 구성요소일 수도 있는, 비디오 코더 (또한, 코더로서 간단히 지칭됨) 에 의해 수행되는 것으로 설명된다.

[0177] 방법 (900) 은 블록 (901) 에서 시작한다. 블록 (905) 에서, 코더는 코딩 유닛 (CU) 에서의 복수의 픽셀들을 분석한다. CU 에서의 각각의 픽셀은 픽셀 값과 연관될 수도 있다. 예를 들어, CU 에서의 다수의 픽셀들은 동일한 픽셀 값을 가질 수도 있다.

[0178] 블록 (910) 에서, 코더는 CU 에서의 복수의 픽셀들에 기초하여 팔레트를 발생시킨다. 팔레트는 인덱스 값과 각각 연관된 복수의 팔레트 엔트리들 및 인덱스 값과 연관된 픽셀 값을 포함할 수도 있다. 팔레트의 예시적인 실시형태가 도 4 에 예시된다.

[0179] 블록 (915) 에서, 코더는 CU 와 연관된 하나 이상의 런 길이들을 결정한다. 위에서 설명한 바와 같이, 각각의 런 길이는 좌측-복사 모드 또는 상부-복사 모드와 연관되는, CU 에서의 현재의 위치를 포함하여 시작하는, 연속된 위치들의 개수를 표시한다.

[0180] 블록 (920) 에서, 코더는 CU 와 연관된 하나 이상의 인덱스 값들을 결정한다. 위에서 설명한 바와 같이, 각각의 인덱스 값은 CU 에서의 현재의 위치와 연관되는 팔레트에서의 픽셀 값을 표시한다. 도 4 의 예에서, 0 의 인덱스 값은 CU 에서의 현재의 위치가 백색 픽셀 값을 갖는다는 것을 표시하며, 1 의 인덱스 값은 CU 에서의 현재의 위치가 그레이 픽셀 값을 갖는다는 것을 표시한다.

[0181] 블록 (925) 에서, 코더는 CU 와 연관된 하나 이상의 이스케이프 픽셀 값들을 결정한다. 위에서 설명한 바와 같이, 각각의 이스케이프 픽셀 값은 CU 와 연관된 팔레트에 있지 않은 픽셀 값을 표시한다. 도 4 의 예에서, 블랙 픽셀 값들을 가지는 CU 에서의 2개의 위치들은 이스케이프 모드에서 코딩되며, 코더는 블랙 픽셀 값들을 비트스트림에서 이스케이프 픽셀 값들로서 시그널링한다.

[0182] 블록 (930) 에서, 코더는 그 발생된 팔레트, 결정된 런 길이들, 결정된 인덱스 값들, 및 결정된 이스케이프 픽셀 값들에 기초하여 CU 를 인코딩한다. 일부 실시형태들에서, 이스케이프 픽셀 값들은 비트스트림에서의 연속된 위치들에서 (예컨대, CU 와 연관된 비트스트림의 부분의 끝에서) 인코딩된다. 예를 들어, 이스케이프 픽셀 값들의 연속된 위치들이 CU 와 연관된 런 길이들 및 인덱스 값들 모두 후의 비트스트림에서 나타난다. 이러한 실시형태들에서, 런 길이들 및 인덱스 값들 모두가 디코더에 의해 파싱된 후, 이스케이프 픽셀 값들은 병렬로 프로세싱될 (예컨대, 파싱될) 수 있다. 이 방법은 블록 (935) 에서 종료한다.

[0183] 방법 (900) 에서, 도 9 에 도시된 블록들 중 하나 이상이 제거될 (예컨대, 수행되지 않을) 수도 있으며 및/또는 본 방법이 수행되는 순서는 스위칭될 수도 있다. 일부 실시형태들에서, 추가적인 블록들이 방법 (900) 에 추가될 수도 있다. 본 개시물의 실시형태들은 도 9 에 나타낸 예에 또는 그에 의해 한정되지 않으며, 다른 변형예들이 본 개시물의 정신으로부터 일탈함이 없이 구현될 수도 있다.

[0184] **다른 고려사항들**

[0185] 본원에서 개시된 정보 및 신호들은 다양한 상이한 기술들 및 기법들 중 어느 것을 이용하여서도 표현될 수도 있다. 예를 들어, 상기 설명 전반에 걸쳐서 인용될 수도 있는 데이터, 명령들, 지령들, 정보, 신호들, 비트들, 심볼들 및 칩들은, 전압들, 전류들, 전자기파들, 자기장들 또는 자기 입자들, 광학장들 또는 광학 입자들, 또는 이들의 임의의 조합으로 표현될 수도 있다.

[0186] 본원에서 개시된 실시형태들과 관련하여 설명된 여러가지 예시적인 로직 블록들, 및 알고리즘 단계들은 전자적 하드웨어, 컴퓨터 소프트웨어, 또는 양쪽의 조합들로서 구현될 수도 있다. 이러한 하드웨어와 소프트웨어의 상호 교환가능성을 명확히 예시하기 위하여, 여러가지 예시적인 구성요소들, 블록들, 및 단계들이 일반적으로 그들의 기능성의 관점에서 위에서 설명되었다. 이런 기능이 하드웨어 또는 소프트웨어로 구현되는지 여부는 특정의 애플리케이션 및 전체 시스템에 부과되는 설계 제한 사항들에 의존한다. 숙련자들은 각각의 특정의 애플리케이션마다 설명한 기능을 여러가지 방법으로 구현할 수도 있으며, 그러나 이런 구현 결정들은 본 개시물의 범위로부터의 일탈을 초래하는 것으로 해석되어서는 안된다.

[0187] 본원에서 설명된 기법들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 이러한 기법들은 범용 컴퓨터들, 무선 통신 디바이스 핸드셋들, 또는 무선 통신 디바이스 핸드셋들 및 다른 디바이스들에서의 애플리케이션을 포함한 다수의 용도들을 가진 집적 회로 디바이스들과 같은 다양한 디바이스들 중 임의의 디바이스에서 구현될 수도 있다. 디바이스들 또는 구성요소들로서 설명하는 임의의 특징들은 통합 로직 디바이스 내에 함께, 또는 별개의 공용가능한 로직 디바이스들로서 별개로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 이 기법들은 실행될 때, 위에서 설명된 방법들 중 하나 이상을 수행하는 명령들을 포함하는 프로그램 코드를 포함하는 컴퓨터-판독가능 데이터 저장 매체에 의해 적어도 부분적으로 실현될 수도 있다. 컴퓨터-판독가능 데이터 저장 매체는 패키징 재료들을 포함할 수도 있는 컴퓨터 프로그램 제품의 일부를 형성할 수도 있다. 컴퓨터-판독가능 매체는 메모리 또는 데이터 저장 매체들, 예컨대, 동기식 동적 랜덤 액세스 메모리 (SDRAM) 와 같은 랜덤 액세스 메모리 (RAM), 판독 전용 메모리 (ROM), 비-휘발성 랜덤 액세스 메모리 (NVRAM), 전기적으로 소거가능한 프로그래밍가능 판독 전용 메모리 (EEPROM), 플래시 메모리, 자기 또는 광학 데이터 저장 매체들 등을 포함할 수도 있다. 기법들은 추가적으로, 또는 대안적으로, 프로그램 코드를 명령들 또는 데이터 구조들의 유형으로 운반하거나 또는 통신하고 그리고 컴퓨터에 의해 액세스되거나, 판독되거나, 및/또는 실행될 수 있는, 전파 신호들 또는 파들과 같은, 컴퓨터-판독가능 통신 매체에 의해 적어도 부분적으로 실현될 수도 있다.

[0188] 프로그램 코드는 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 로직 어레이들 (FPGA들), 또는 다른 등가의 집적 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들을 포함할 수도 있는 프로세서에 의해 실행될 수도 있다. 이러한 프로세서는 본 개시물에서 설명하는 기법들 중 임의의 기법들을 수행하도록 구성될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있으며; 그러나 대안적으로는, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로제어기 또는 상태 머신일 수도 있다. 프로세서는 또한 컴퓨팅 디바이스들의 조합, 예컨대, DSP 와 마이크로프로세서의 조합, 복수의 마이크로프로세서들, DSP 코어와 결합된 하나 이상의 마이크로프로세서들, 또는 임의의 다른 이러한 구성으로서 구현될 수도 있다. 따라서, 용어 "프로세서" 는, 본원에서 사용될 때 전술한 구조 중 임의의 구조, 전술한 구조의 임의의 조합, 또는 본원에서 설명하는 기법들의 구현에 적합한 임의의 다른 구조 또는 장치를 지칭할 수도 있다. 게다가, 일부 양태들에서, 본원에서 설명하는 기능은 인코딩 및 디코딩하도록 구성된 전용 소프트웨어 또는 하드웨어 내에 제공되거나, 또는 결합된 비디오 인코더-디코더 (코덱) 에 포함될

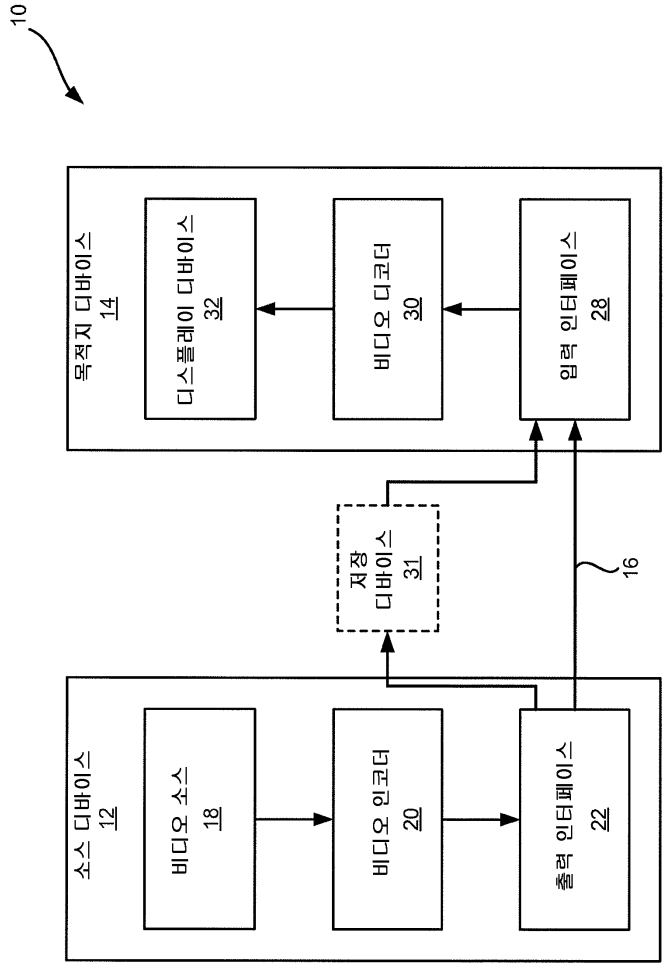
수도 있다. 또한, 이 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들로 전적으로 구현될 수 있다.

[0189] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예컨대, 칩 세트) 를 포함한, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 개시한 기법들을 수행하도록 구성되는 디바이스들의 기능적 양태들을 강조하기 위해서 여러 구성요소들, 또는 유닛들이 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하지는 않는다. 대신, 위에서 설명한 바와 같이, 여러 유닛들이 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명한 바와 같은 하나 이상의 프로세서들을 포함한, 상호작용하는 하드웨어 유닛들의 컬렉션으로 제공될 수도 있다.

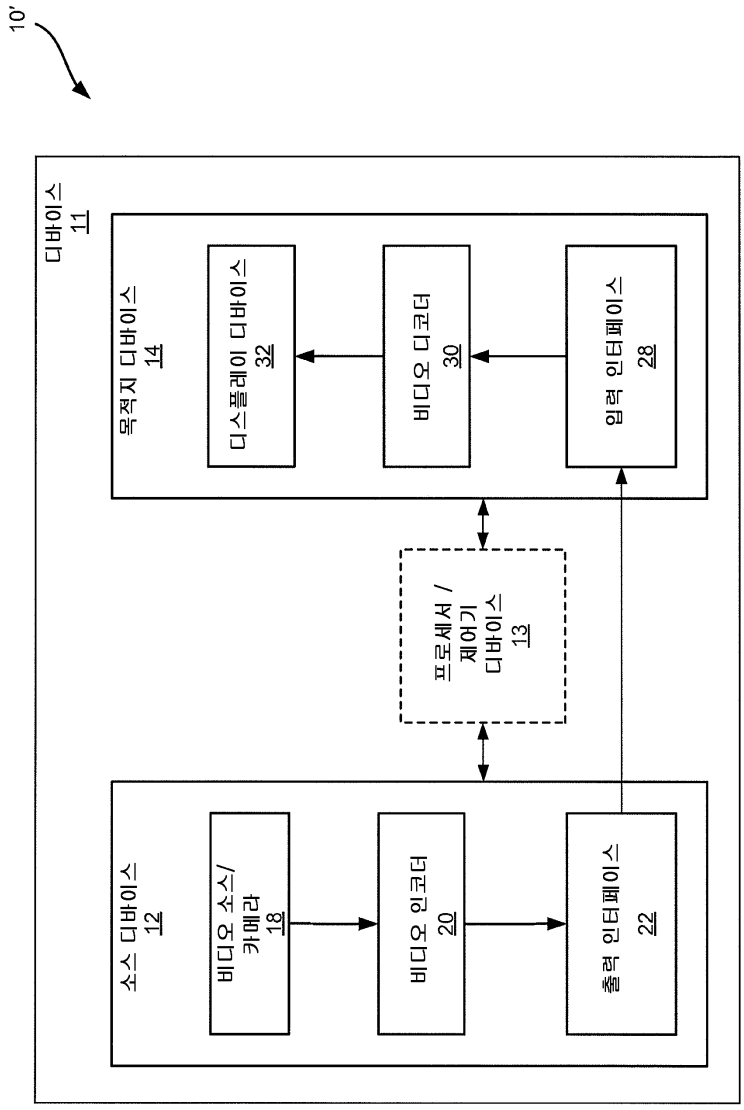
[0190] 상기는 여러 상이한 실시형태들과 관련하여 설명되었지만, 일 실시형태로부터의 특징들 또는 엘리먼트들은 본 개시물의 교시들로부터 일탈함이 없이 다른 실시형태들과 결합될 수도 있다. 그러나, 개개의 실시형태들 사이의 특징들의 조합들은 반드시 이에 한정되지 않는다. 본 개시물의 여러 실시형태들이 설명되었다. 이들 및 다른 실시형태들은 다음 청구항들의 범위 이내이다.

도면

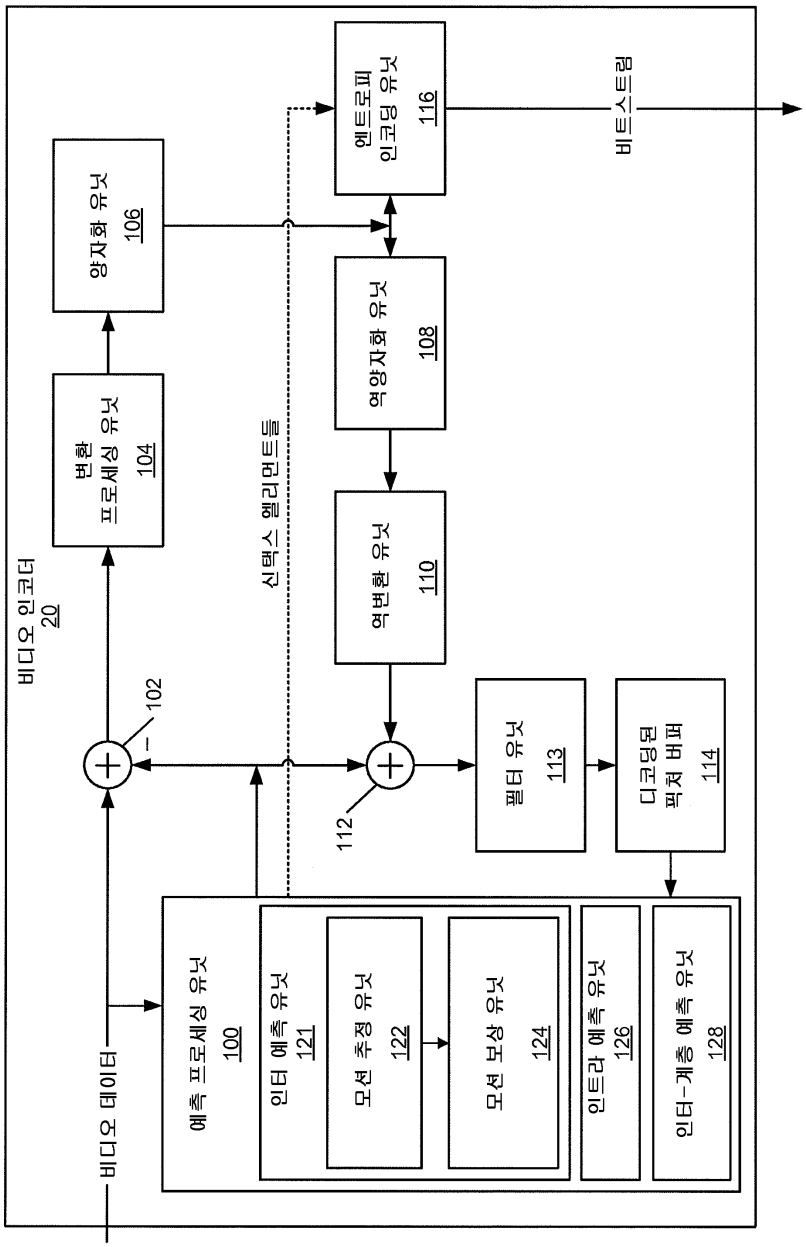
도면1a



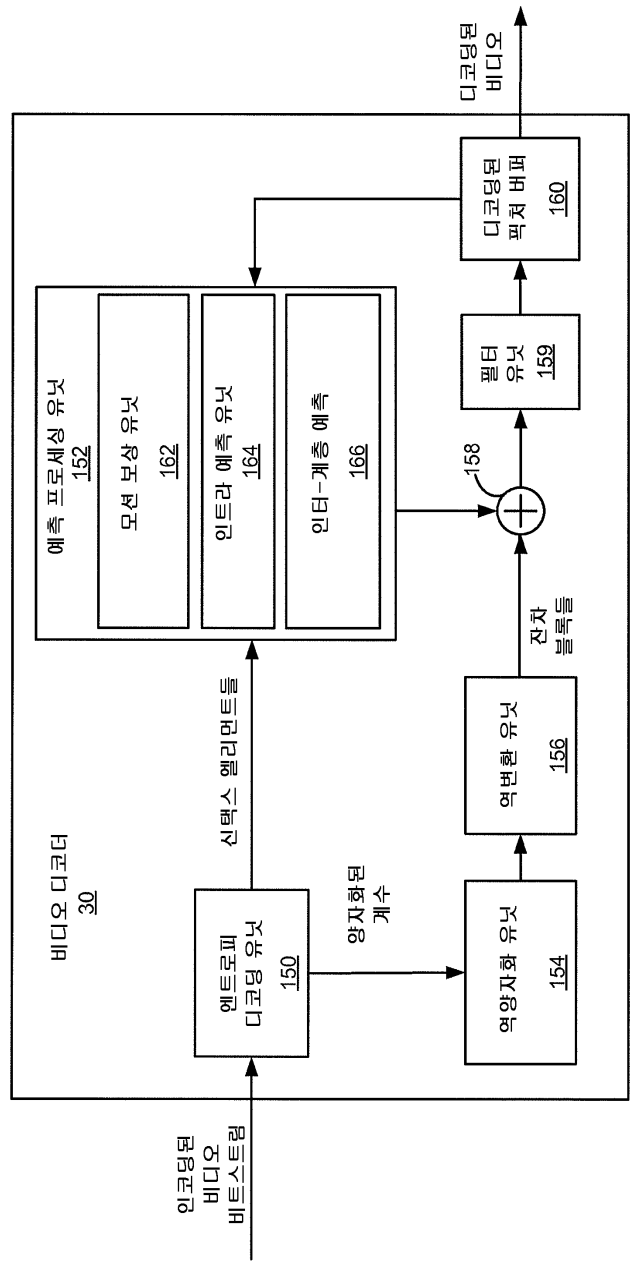
도면1b


10'
디바이스
11
목적지 디바이스
14
디스플레이 디바이스
32
비디오 디코더
30
입력 인터페이스
28
프로세서 /
제어기
디바이스
13
소스 디바이스
12
비디오 소스/
카메라
18
비디오 인코더
20
출력 인터페이스
22

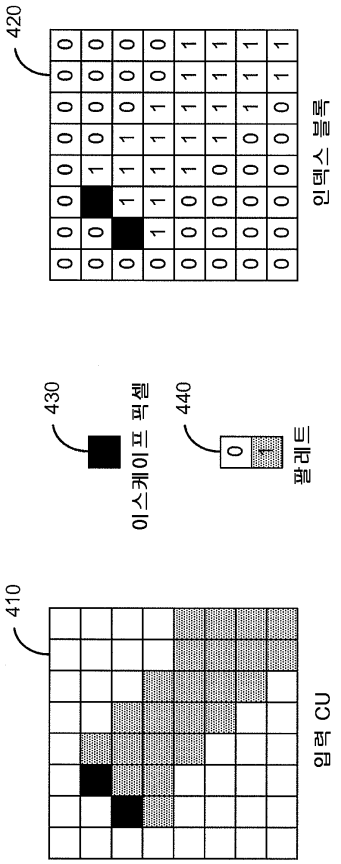
도면2


비디오 인코더
20
비디오 데이터
예측 프로세싱 유닛
100
인터 예측 유닛
121
모션 추정 유닛
122
모션 보상 유닛
124
인트라 예측 유닛
126
인터-계층 예측 유닛
128
디코딩된
픽처 버퍼
114
필터 유닛
113
역변환 유닛
110
역양자화 유닛
108
엔트로피
인코딩 유닛
116
비트스트림
양자화 유닛
106
변환
프로세싱 유닛
104
신택스 엘리먼트들
102
112

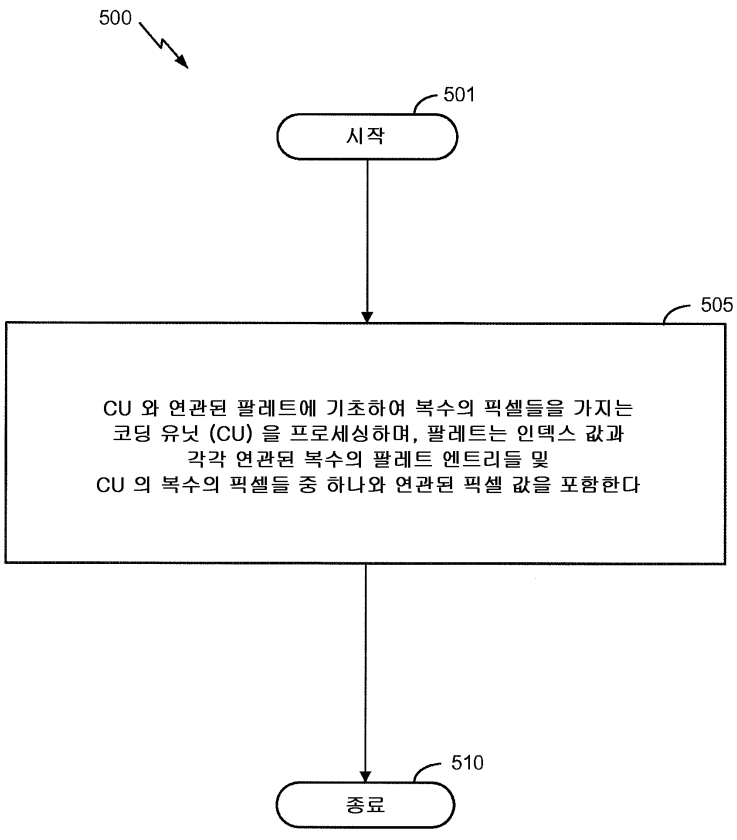
도면3


디코딩된 비디오
디코딩된 픽처 버퍼 160
필터 유닛 159
158
예측 프로세싱 유닛 152
모션 보상 유닛 162
인트라 예측 유닛 164
인터-계층 예측 166
잔차 블록들
역변환 유닛 156
신택스 엘리먼트들
양자화된 계수
엔트로피 디코딩 유닛 150
역양자화 유닛 154
비디오 디코더 30
인코딩된 비디오 비트스트림

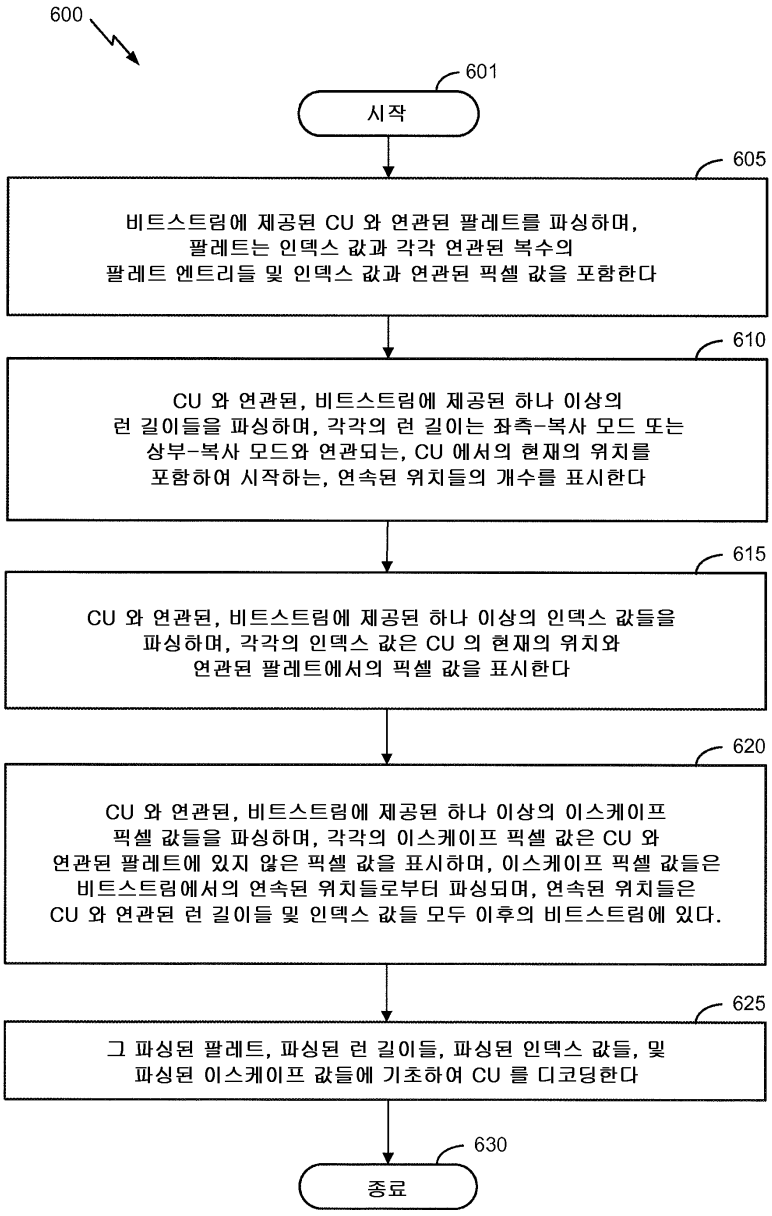
도면4


420
인덱스 블록
430
이스케이프 픽셀
440
팔레트
410
입력 CU

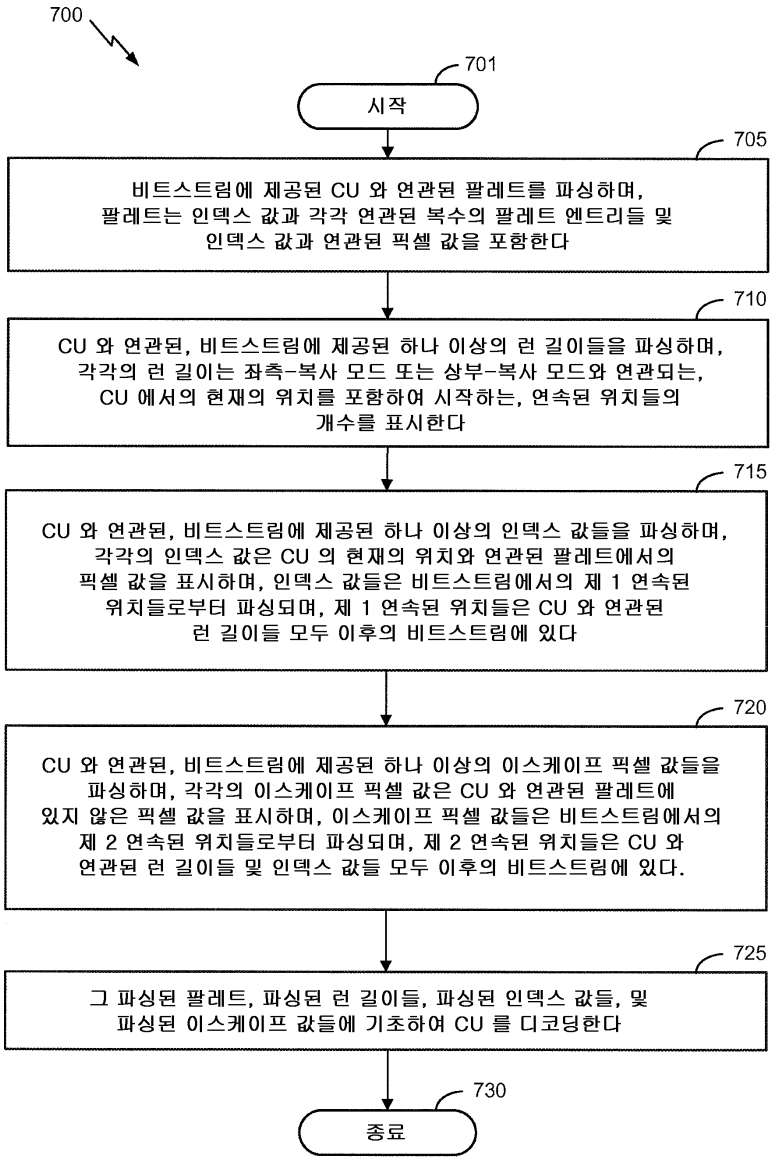
도면5


500
501
시작
505
CU 와 연관된 팔레트에 기초하여 복수의 픽셀들을 가지는
코딩 유닛 (CU) 을 프로세싱하며, 팔레트는 인덱스 값과
각각 연관된 복수의 팔레트 엔트리들 및
CU 의 복수의 픽셀들 중 하나와 연관된 픽셀 값을 포함한다
510
종료

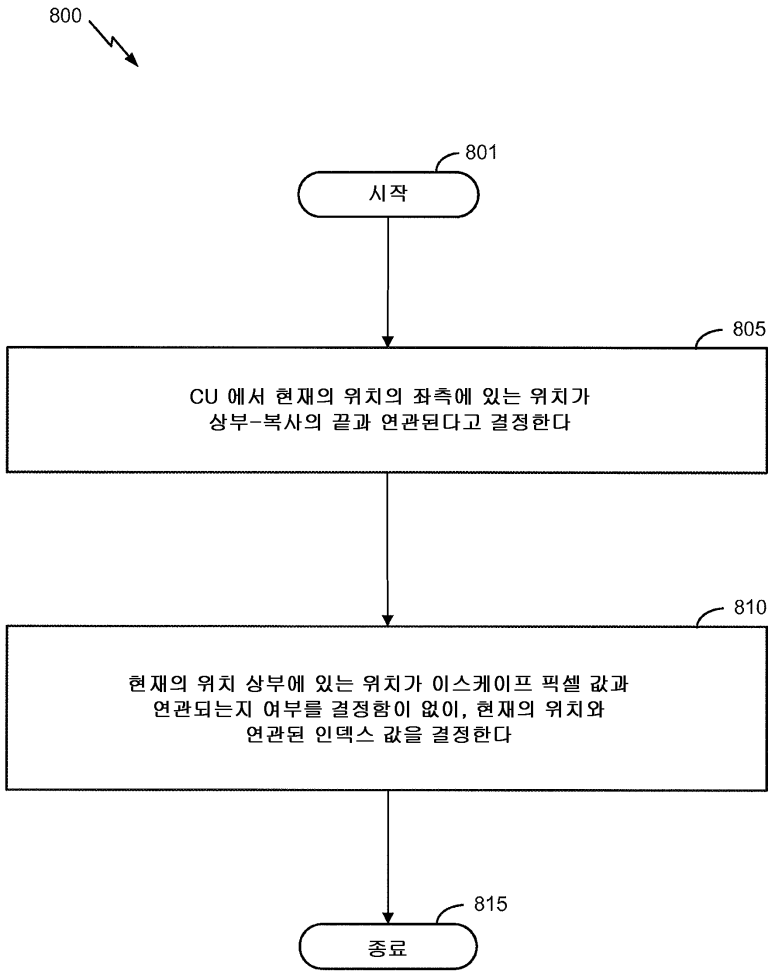
도면6


600
601
시작
605
비트스트림에 제공된 CU 와 연관된 팔레트를 파싱하며,
팔레트는 인덱스 값과 각각 연관된 복수의
팔레트 엔트리들 및 인덱스 값과 연관된 픽셀 값을 포함한다
610
CU 와 연관된, 비트스트림에 제공된 하나 이상의
런 길이들을 파싱하며, 각각의 런 길이는 좌측-복사 모드 또는
상부-복사 모드와 연관되는, CU 에서의 현재의 위치를
포함하여 시작하는, 연속된 위치들의 개수를 표시한다
615
CU 와 연관된, 비트스트림에 제공된 하나 이상의 인덱스 값들을
파싱하며, 각각의 인덱스 값은 CU 의 현재의 위치와
연관된 팔레트에서의 픽셀 값을 표시한다
620
CU 와 연관된, 비트스트림에 제공된 하나 이상의 이스케이프
픽셀 값들을 파싱하며, 각각의 이스케이프 픽셀 값은 CU 와
연관된 팔레트에 있지 않은 픽셀 값을 표시하며, 이스케이프 픽셀 값들은
비트스트림에서의 연속된 위치들로부터 파싱되며, 연속된 위치들은
CU 와 연관된 런 길이들 및 인덱스 값들 모두 이후의 비트스트림에 있다.
625
그 파싱된 팔레트, 파싱된 런 길이들, 파싱된 인덱스 값들, 및
파싱된 이스케이프 값들에 기초하여 CU 를 디코딩한다
630
종료

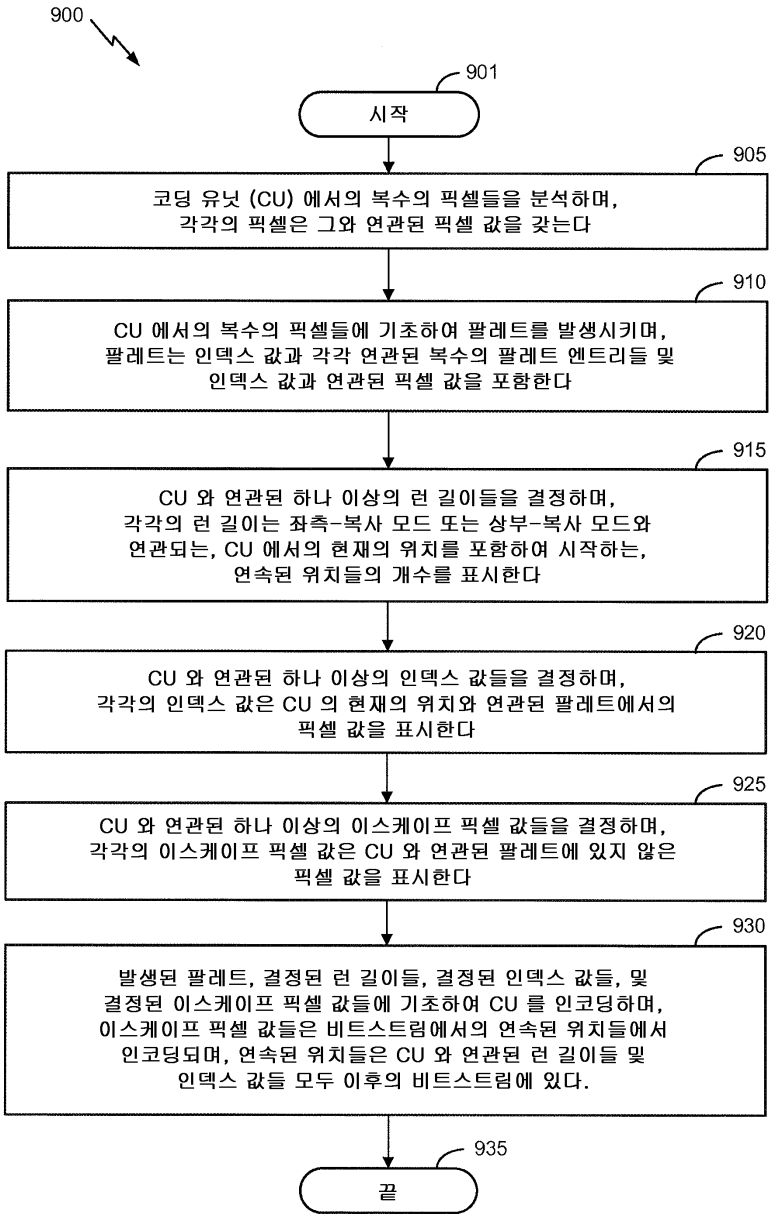
도면7


700
701
시작
705
비트스트림에 제공된 CU 와 연관된 팔레트를 파싱하며, 팔레트는 인덱스 값과 각각 연관된 복수의 팔레트 엔트리들 및 인덱스 값과 연관된 픽셀 값을 포함한다
710
CU 와 연관된, 비트스트림에 제공된 하나 이상의 런 길이들을 파싱하며, 각각의 런 길이는 좌측-복사 모드 또는 상부-복사 모드와 연관되는, CU 에서의 현재의 위치를 포함하여 시작하는, 연속된 위치들의 개수를 표시한다
715
CU 와 연관된, 비트스트림에 제공된 하나 이상의 인덱스 값들을 파싱하며, 각각의 인덱스 값은 CU 의 현재의 위치와 연관된 팔레트에서의 픽셀 값을 표시하며, 인덱스 값들은 비트스트림에서의 제 1 연속된 위치들로부터 파싱되며, 제 1 연속된 위치들은 CU 와 연관된 런 길이들 모두 이후의 비트스트림에 있다
720
CU 와 연관된, 비트스트림에 제공된 하나 이상의 이스케이프 픽셀 값들을 파싱하며, 각각의 이스케이프 픽셀 값은 CU 와 연관된 팔레트에 있지 않은 픽셀 값을 표시하며, 이스케이프 픽셀 값들은 비트스트림에서의 제 2 연속된 위치들로부터 파싱되며, 제 2 연속된 위치들은 CU 와 연관된 런 길이들 및 인덱스 값들 모두 이후의 비트스트림에 있다.
725
그 파싱된 팔레트, 파싱된 런 길이들, 파싱된 인덱스 값들, 및 파싱된 이스케이프 값들에 기초하여 CU 를 디코딩한다
730
종료

도면8


800
801
시작
805
CU 에서 현재의 위치의 좌측에 있는 위치가
상부-복사의 끝과 연관된다고 결정한다
810
현재의 위치 상부에 있는 위치가 이스케이프 픽셀 값과
연관되는지 여부를 결정함이 없이, 현재의 위치와
연관된 인덱스 값을 결정한다
815
종료

도면9


900
901
시작
905
코딩 유닛 (CU) 에서의 복수의 픽셀들을 분석하며,
각각의 픽셀은 그와 연관된 픽셀 값을 갖는다
910
CU 에서의 복수의 픽셀들에 기초하여 팔레트를 발생시키며,
팔레트는 인덱스 값과 각각 연관된 복수의 팔레트 엔트리들 및
인덱스 값과 연관된 픽셀 값을 포함한다
915
CU 와 연관된 하나 이상의 런 길이들을 결정하며,
각각의 런 길이는 좌측-복사 모드 또는 상부-복사 모드와
연관되는, CU 에서의 현재의 위치를 포함하여 시작하는,
연속된 위치들의 개수를 표시한다
920
CU 와 연관된 하나 이상의 인덱스 값들을 결정하며,
각각의 인덱스 값은 CU 의 현재의 위치와 연관된 팔레트에서의
픽셀 값을 표시한다
925
CU 와 연관된 하나 이상의 이스케이프 픽셀 값들을 결정하며,
각각의 이스케이프 픽셀 값은 CU 와 연관된 팔레트에 있지 않은
픽셀 값을 표시한다
930
발생된 팔레트, 결정된 런 길이들, 결정된 인덱스 값들, 및
결정된 이스케이프 픽셀 값들에 기초하여 CU 를 인코딩하며,
이스케이프 픽셀 값들은 비트스트림에서의 연속된 위치들에서
인코딩되며, 연속된 위치들은 CU 와 연관된 런 길이들 및
인덱스 값들 모두 이후의 비트스트림에 있다.
935
끝