



등록특허 10-2366842



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년02월24일
(11) 등록번호 10-2366842
(24) 등록일자 2022년02월18일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/463 (2014.01) *HO4N 19/186* (2014.01)
HO4N 19/30 (2014.01) *HO4N 19/50* (2014.01)
- (52) CPC특허분류
HO4N 19/463 (2015.01)
HO4N 19/186 (2015.01)
- (21) 출원번호 10-2021-7015575(분할)
- (22) 출원일자(국제) 2014년03월17일
심사청구일자 2021년06월23일
- (85) 번역문제출일자 2021년05월24일
- (65) 공개번호 10-2021-0064395
- (43) 공개일자 2021년06월02일
- (62) 원출원 특허 10-2015-7031727
원출원일자(국제) 2014년03월17일
심사청구일자 2019년03월15일
- (86) 국제출원번호 PCT/EP2014/055333
- (87) 국제공개번호 WO 2014/166705
국제공개일자 2014년10월16일

- (30) 우선권주장
13305453.6 2013년04월08일
유럽특허청(EPO)(EP)
(뒷면에 계속)
- (56) 선행기술조사문현
US05748176 A*
(뒷면에 계속)

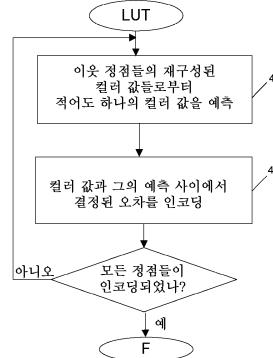
전체 청구항 수 : 총 10 항

심사관 : 조우연

(54) 발명의 명칭 LUT를 인코딩하는 방법, LUT를 디코딩하는 방법 및 대응하는 장치들

(57) 요 약

정점들의 격자로서 정의되는 LUT를 인코딩하기 위한 방법이 개시된다. 적어도 하나의 값이 격자의 각각의 정점에 속한다. 이 방법은 현재 정점에 대해: 상기 현재 정점의 적어도 하나의 값을, 예를 들어 이웃 정점들의 재구성 값들로부터 획득되는 다른 값으로부터 예측하는 단계; 및 현재 정점의 적어도 하나의 값과 비트스트림 내의 그의 예측 사이에서 계산된 적어도 하나의 오차를 비트스트림 내에 인코딩하는 단계를 포함한다.

대 표 도 - 도4

(52) CPC특허분류

HO4N 19/30 (2015.01)

HO4N 19/50 (2015.01)

(56) 선행기술조사문헌

US20130034166 A1*

KR1020090038374 A*

KR1020080094542 A*

US5049986 A

US20060268297 A1

Philippe Bordes et al., "AHG14: Color Gamut Scalable Video Coding using 3D LUT", (JCTVC-M0197), JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 13th Meeting: Incheon, KR, (2013.04.08)

*는 심사관에 의하여 인용된 문헌

(30) 우선권주장

13306010.3 2013년07월15일

유럽특허청(EPO)(EP)

14305109.2 2014년01월27일

유럽특허청(EPO)(EP)

명세서

청구범위

청구항 1

옥트리(octree)를 이용하여 정점들의 격자로서 정의되는 삼차원 컬러 탐색표를 인코딩하기 위한 방법으로서, 상기 삼차원 컬러 탐색표는 입력 컬러 값들의 세트를 표현하는 각각의 정점에 대한 적어도 하나의 값을 저장하고, 상기 방법은,

상기 옥트리의 현재 옥턴트(octant)에 대해 그의 직접 자식 옥턴트들 중 적어도 하나가 재귀적으로 인코딩되는지를 지시하는 분할 플래그를 인코딩하는 단계;

예측을 획득하기 위해 적어도 하나의 이웃 정점의 값(들)의 재구성으로부터 상기 현재 옥턴트 내의 정점에 대한 적어도 하나의 값을 예측하는 단계;

상기 정점에 대한 상기 적어도 하나의 값과 상기 예측 사이에서 계산된 오차(residue)를 결정하는 단계;

상기 오차를 양자화하여 양자화된 오차를 획득하는 단계;

상기 양자화된 오차를 엔트로피 코딩하는 단계; 및

상기 현재 옥턴트에 대한 분할 플래그가 참인 경우에 상기 현재 옥턴트의 직접 자식 옥턴트들을 인코딩하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서, 상기 정점에 대해 적어도 하나의 오차가 인코딩되는지의 여부를 지시하는 상기 정점에 대한 이진 플래그를 인코딩하는 단계를 더 포함하는 방법.

청구항 3

제1항에 있어서, 정점에 대한 상기 적어도 하나의 값 및 상기 이웃 정점에 대한 값(들)의 상기 재구성은 컬러 값들인 방법.

청구항 4

제3항에 있어서, 상기 컬러 값들은 컬러 변환을 위한 파라미터들인 방법.

청구항 5

제1항에 있어서, 상기 이웃 정점에 대한 값(들)의 상기 재구성을 획득하는 데 사용되는 상기 적어도 하나의 이웃 정점은 상기 현재 옥턴트의 부모 옥턴트에 속하는 방법.

청구항 6

옥트리를 이용하여 정점들의 격자로서 정의되는 삼차원 컬러 탐색표를 디코딩하기 위한 방법으로서, 상기 삼차원 컬러 탐색표는 입력 컬러 값들의 세트를 표현하는 각각의 정점에 대한 적어도 하나의 값을 저장하고, 상기 방법은,

상기 옥트리의 현재 옥턴트에 대해 그의 직접 자식 옥턴트들 중 적어도 하나가 재귀적으로 디코딩되어야 하는지를 지시하는 분할 플래그를 디코딩하는 단계;

상기 현재 옥턴트 내의 정점에 대한 적어도 하나의 값을, 적어도 하나의 이웃 정점에 대한 값(들)의 재구성으로부터 획득된 다른 값으로부터 예측하여 예측을 획득하는 단계;

상기 현재 옥턴트 내의 상기 정점에 대한 상기 적어도 하나의 값의 엔트로피 코딩되고 양자화된 오차를 엔트로피 디코딩하여 양자화된 오차를 획득하는 단계;

상기 현재 옥턴트 내의 상기 정점에 대한 상기 적어도 하나의 값의 상기 양자화된 오차를 역양자화하여 오차를 획득하는 단계;

상기 예측 및 상기 오차로부터 상기 정점에 대한 상기 적어도 하나의 값을 재구성하는 단계; 및

상기 현재 옥턴트에 대한 분할 플래그가 참인 경우에 상기 현재 옥턴트의 직접 자식 옥턴트들을 디코딩하는 단계

를 포함하는 방법.

청구항 7

제6항에 있어서, 상기 정점에 대해 적어도 하나의 오차가 디코딩되어야 하는지의 여부를 지시하는 상기 정점에 대한 이진 플래그를 디코딩하는 단계를 더 포함하는 방법.

청구항 8

제6항에 있어서, 정점에 대한 상기 적어도 하나의 값 및 상기 이웃 정점에 대한 값(들)의 상기 재구성은 컬러 값들인 방법.

청구항 9

제8항에 있어서, 상기 컬러 값들은 컬러 변환을 위한 파라미터들인 방법.

청구항 10

제6항에 있어서, 상기 이웃 정점에 대한 값(들)의 상기 재구성을 획득하는 데 사용되는 상기 적어도 하나의 이웃 정점은 상기 현재 옥턴트의 부모 옥턴트에 속하는 방법.

발명의 설명

기술 분야

[0001]

정점들의 격자로서 정의되는 탐색표를 인코딩하기 위한 방법이 개시되며, 적어도 하나의 값이 격자의 각각의 정점과 관련된다. 예를 들어, 값은 컬러 값이다. 대응하는 디코딩 방법, 인코딩 장치 및 디코딩 장치가 개시된다.

배경 기술

[0002]

도 1에 도시된 바와 같이, 스케일링 가능 비디오 디코딩은 기본 계층(BL) 비트스트림 및 적어도 하나의 향상 계층(EL) 비트스트림의 디코딩(각각의 인코딩)으로 구성된다. 통상적으로, EL 빙辍들은 (아마도 업샘플링된) 디코딩된 BL 빙辍들로부터 예측된다. 그러나, EL 빙辍들 및 BL 빙辍들이 상이한 컬러 공간들을 이용하여 표현되고/되거나, 상이하게 컬러 등급화된 때, 예측은 덜 효율적이다. 예측을 개선하기 위해, 디코딩된 BL 빙辍들에 대해 컬러 변환을 적용하는 것이 알려져 있다. 더 정확하게는, 컬러 변환은 컬러 정보를 이용하여 BL 컬러 공간(제1 컬러 공간)의 컬러들을 EL 컬러 공간(제2 컬러 공간)의 컬러들 상에 맵핑한다.

[0003]

도 2에 도시된 바와 같이, 비디오 콘텐츠 배포에서, 디코딩된 빙辍들에 대해 통상적으로 컬러 변환이 적용되며, 따라서 디코딩 및 변환된 빙辍들은 최종 장치 렌더링 능력에 적합해진다.

[0004]

이러한 컬러 변환은 컬러 맵핑 기능(CMF)으로도 알려져 있다. CMF는 예를 들어 3x3 이득 행렬 + 오프셋(이득 오프셋 모델)에 의해 근사화된다. 이 경우, CMF는 12개의 파라미터에 의해 정의된다. 그러나, CMF의 그러한 근사화는 선형 변환 모델을 가정하므로 매우 정확하지는 않다. 결과적으로, (3D LUT로도 알려진) 3D 탐색표가 CMF 모델에 대한 어떠한 선천성도 없이 그러한 CMF를 설명하는 데 사용된다. 3D LUT는 그의 크기가 필요한 정밀도에 따라 증가될 수 있으므로 훨씬 더 정밀하다. 그러나, 3D LUT는 그에 따라 거대한 데이터 세트를 표현할 수 있다. 따라서, 3D LUT를 수신기로 전송하는 것은 LUT의 인코딩을 필요로 한다.

[0005]

CMF를 근사화하는 LUT는 제1 컬러 공간 내의 적어도 하나의 컬러 값을 제2 컬러 공간 내의 다른 컬러 값과 관련시킨다. LUT는 제1 컬러 공간을 LUT의 정점들에 의해 정해지는 영역들의 세트로 분할하는 것을 가능하게 한다.

예를 들어, 3D LUT는 제1 컬러 공간 내의 컬러 값들의 삼중쌍(triplet)을 컬러 값들의 세트와 관련시킨다. 컬러 값들의 세트는 제2 컬러 공간 내의 컬러 값들의 삼중쌍, 또는 제1 컬러 공간 내의 컬러 값들을 제2 컬러 공간 내의 컬러 값들로 변환하는 데 사용되는 컬러 변환을 나타내는 컬러 값들의 세트(예로서, 논리적으로 정의된 CMF 파라미터들)일 수 있다. 도 3에서, 정사각형 3D LUT는 NxNxN 정점들의 격자로서 표현된다. 3D LUT의 각각의 정점 $V(c_1, c_2, c_3)$ 에 대해, 컬러 값들(V_{c1} , V_{c2} , V_{c3})의 대응하는 삼중쌍이 저장될 필요가 있다. 3D LUT와 관련된 데이터의 양은 $NxNxNxK$ 이며, 여기서 K 는 하나의 LUT 삼중쌍 값을 저장하는 데 사용되는 비트들의 양이다. 삼중쌍 값은 예를 들어 (R, G, B) 삼중쌍, (Y, U, V) 삼중쌍 또는 (Y, Cb, Cr) 삼중쌍 등이다. 모든 정점 값을 인코딩하는 것은 거대한 양의 데이터를 표현하므로 효율적이지 못하다.

발명의 내용

[0006]

발명의 요약

[0007]

본 발명의 목적은 종래 기술의 단점들 중 적어도 하나를 극복하는 것이다. 정점들의 격자로서 정의되는 LUT를 인코딩하기 위한 방법이 개시된다. 적어도 하나의 값이 상기 격자의 각각의 정점과 관련된다. 인코딩 방법은 현재 정점에 대해:

[0008]

- 상기 현재 정점과 관련된 상기 적어도 하나의 값을 다른 값으로부터 예측하는 단계; 및

[0009]

- 상기 현재 정점의 상기 적어도 하나의 값과 비트스트림 내의 그의 예측 사이에서 계산된 적어도 하나의 오차를 비트스트림 내에 인코딩하는 단계

[0010]

를 포함한다.

[0011]

유리하게는, 상기 다른 값은 이웃 정점들과 관련된 재구성 값들로부터 획득된다.

[0012]

본 발명의 특정 특징에 따르면, 상기 값은 컬러 값이다. 특정 실시예에 따르면, 상기 컬러 값은 컬러 변환을 나타낸다. 본 발명의 특정 특징에 따르면, 상기 컬러 값은 이득 파라미터 또는 오프셋이다.

[0013]

특정 실시예에 따르면, 상기 LUT는 옥트리(octree)를 이용하여 인코딩된 3D LUT이며, 3개의 값이 상기 격자의 각각의 정점과 관련된다. 이 경우, 예측에 사용되는 상기 이웃 정점들은 상기 현재 정점이 속하는 현재 옥텟트(octant)의 부모 옥텟트에 속한다.

[0014]

본 발명의 특정 특징에 따르면, 상기 현재 정점과 관련된 상기 적어도 하나의 값을 이웃 정점들과 관련된 재구성 값들로부터 예측하는 단계는 상기 이웃 정점들의 대응하는 재구성 값들로부터 상기 적어도 하나의 값을 보간하는 단계를 포함한다.

[0015]

특정 특징에 따르면, 상기 방법은 보간의 타입을 지시하는 인덱스를 상기 비트스트림 내에 인코딩하는 단계를 더 포함한다.

[0016]

특정 실시예에 따르면, 적어도 하나의 오차를 인코딩하는 단계는 양자화기를 이용하여 상기 적어도 하나의 오차를 양자화된 오차로 양자화하고, 상기 양자화된 오차를 상기 비트스트림 내에 엔트로피 코딩하는 단계를 포함하고, 상기 방법은 상기 양자화기를 상기 비트스트림 내에 인코딩하는 단계를 더 포함한다.

[0017]

본 발명의 특정 특징에 따르면, 플래그가 상기 현재 정점에 대해 인코딩되며, 상기 플래그는 적어도 하나의 오차가 상기 정점에 대해 인코딩되는지의 여부를 지시한다.

[0018]

일 변형에 따르면, 플래그가 상기 현재 정점의 각각의 값에 대해 인코딩되며, 상기 플래그는 오차가 해당 값에 대해 인코딩되는지 또는 상기 오차가 인코딩되지 않고 0인 것으로 추정되는지를 지시한다.

[0019]

유리하게는, 분할 플래그가 상기 현재 옥텟트에 대해 인코딩되며, 상기 분할 플래그는 그의 직접 자식들(immediate children)이 재귀적으로 인코딩되는지 또는 아직 인코딩되지 않은 모든 그의 자식들의 상기 정점들의 모든 상기 오차들이 0인 것으로 추정되는지를 지시한다.

[0020]

정점들의 격자로서 정의되는 LUT를 디코딩하기 위한 방법도 개시된다. 적어도 하나의 값이 상기 격자의 각각의 정점과 관련된다. 상기 디코딩 방법은 현재 정점에 대해:

[0021]

- 비트스트림으로부터 적어도 하나의 오차를 디코딩하는 단계;

[0022]

- 상기 현재 정점과 관련된 상기 적어도 하나의 값을 다른 값으로부터 예측하는 단계; 및

- [0023] - 상기 현재 정점의 상기 적어도 하나의 값을 그의 예측 및 상기 디코딩된 적어도 하나의 오차로부터 재구성하는 단계
- [0024] 를 포함한다.
- [0025] 유리하게는, 상기 다른 값은 이웃 정점들과 관련된 재구성 값들로부터 획득된다.
- [0026] 본 발명의 특정 특징에 따르면, 상기 값은 컬러 값이다. 특정 실시예에 따르면, 상기 컬러 값은 컬러 변환을 나타낸다. 본 발명의 특정 특징에 따르면, 상기 컬러 값은 이득 파라미터 또는 오프셋이다.
- [0027] 특정 실시예에 따르면, 상기 LUT는 옥트리를 이용하여 디코딩된 3D LUT이며, 3개의 값이 상기 격자의 각각의 정점과 관련되고, 상기 이웃 정점들은 상기 현재 정점이 속하는 현재 옥턴트의 부모 옥턴트에 속한다.
- [0028] 본 발명의 특정 특징에 따르면, 상기 현재 정점과 관련된 상기 적어도 하나의 값을 이웃 정점들과 관련된 재구성 값들로부터 예측하는 단계는 상기 이웃 정점들의 대응하는 재구성 값들로부터 상기 적어도 하나의 값을 보간하는 단계를 포함한다.
- [0029] 특정 특징에 따르면, 상기 방법은 보간의 타입을 지시하는 인덱스를 상기 비트스트림으로부터 디코딩하는 단계를 더 포함한다.
- [0030] 특정 실시예에 따르면, 적어도 하나의 오차를 디코딩하는 단계는 양자화된 오차를 상기 비트스트림으로부터 엔트로피 디코딩하고, 양자화기를 이용하여 상기 양자화된 오차를 디코딩된 오차로 역양자화하는 단계를 포함하고, 상기 방법은 상기 비트스트림으로부터 상기 양자화기를 디코딩하는 단계를 더 포함한다.
- [0031] 본 발명의 특정 특징에 따르면, 플래그가 상기 각각의 정점에 대해 디코딩되며, 상기 플래그는 적어도 하나의 오차가 상기 정점에 대해 인코딩되는지의 여부를 지시한다.
- [0032] 일 변형에 따르면, 플래그가 상기 현재 정점의 각각의 값에 대해 디코딩되며, 상기 플래그는 오차가 해당 값에 대해 디코딩되는지 또는 상기 오차가 0인 것으로 추정되는지를 지시한다.
- [0033] 유리하게는, 분할 플래그가 상기 현재 옥턴트에 대해 디코딩되며, 상기 분할 플래그는 그의 직접 자식들이 재귀적으로 디코딩되는지 또는 아직 디코딩되지 않은 모든 그의 자식들의 상기 정점들의 모든 상기 오차들이 0인 것으로 추정되는지를 지시한다.
- [0034] 정점들의 격자로서 정의되는 LUT를 인코딩하기 위한 인코더로서, 적어도 하나의 값이 상기 격자의 각각의 정점과 관련되며, 현재 정점에 대해:
- 현재 정점과 관련된 상기 적어도 하나의 값을 다른 값으로부터 예측하기 위한 수단; 및
- [0035] - 상기 현재 정점의 상기 적어도 하나의 값과 비트스트림 내의 그의 예측 사이에서 계산된 적어도 하나의 오차를 비트스트림 내에 인코딩하기 위한 수단
- [0036] 을 포함하는 인코더.
- [0037] 유리하게는, 상기 다른 값은 이웃 정점들과 관련된 재구성 값들로부터 획득된다.
- [0038] 상기 인코더는 인코딩을 위한 상기 방법의 상기 단계들을 실행하도록 구성된다.
- [0039] 정점들의 격자로서 정의되는 LUT를 디코딩하기 위한 디코더로서, 적어도 하나의 값이 상기 격자의 각각의 정점과 관련되며,
- 비트스트림으로부터 현재 정점과 관련된 적어도 하나의 오차를 디코딩하기 위한 수단;
- [0040] - 상기 현재 정점과 관련된 상기 적어도 하나의 값을 다른 값으로부터 예측하기 위한 수단; 및
- [0041] - 상기 현재 정점의 상기 적어도 하나의 값을 그의 예측 및 상기 디코딩된 적어도 하나의 오차로부터 재구성하기 위한 수단
- [0042] 을 포함하는 디코더.
- [0043] 유리하게는, 상기 다른 값은 이웃 정점들과 관련된 재구성 값들로부터 획득된다.
- [0044] 상기 인코더는 디코딩을 위한 상기 방법의 상기 단계들을 실행하도록 구성된다.
- [0045] 정점들의 격자로서 정의되는 적어도 LUT를 인코딩하는 비트스트림이 개시된다. 적어도 하나의 값이 상기 격자

의 각각의 정점과 관련되며, 상기 비트스트림은 현재 정점의 상기 적어도 하나의 값과 그의 예측 사이에서 계산되는 적어도 하나의 오차를 그 안에 인코딩하는 것을 포함한다.

도면의 간단한 설명

[0048]

본 발명의 다른 특징들 및 장점들은 그의 실시예들 중 일부에 대한 아래의 설명으로부터 명백해질 것이며, 이 설명은 도면들과 관련하여 이루어진다. 도면들에서:

도 1은 종래 기술에 따른, 범위 스케일링 가능성을 이용하는 스케일링 가능 비디오 디코더의 아키텍처를 나타낸다.

도 2는 종래 기술에 따른, 렌더링 표시 특성들에 적응하기 위한 컬러 변환을 포함하는 비디오 디코더의 아키텍처를 나타낸다.

도 3은 종래 기술에 따른, NxNxN 정점들의 격자로서 표현되는 직사각형 3D LUT를 개략적으로 나타낸다.

도 4는 본 발명의 일 실시예에 따른 인코딩 방법의 흐름도를 나타낸다.

도 5는 2D LUT를 개략적으로 나타낸다.

도 6은 본 발명의 다른 실시예에 따른 인코딩 방법의 흐름도를 나타낸다.

도 7은 좌측에 큐브의 옥텟트들로의 재귀적 세분을, 우측에 대응하는 옥트리를 나타낸다.

도 8은 본 발명에 따른, 정점의 컬러 값들의 보간을 나타낸다.

도 9는 불균일한 격자로서 표현되는 3D LUT를 개략적으로 나타낸다.

도 10은 본 발명의 다른 실시예에 따른 디코딩 방법의 흐름도를 나타낸다.

도 11은 본 발명의 다른 실시예에 따른 디코딩 방법의 흐름도를 나타낸다.

도 12는 본 발명에 따른, LUT를 인코딩하기 위한 인코더를 개략적으로 나타낸다.

도 13은 본 발명에 따른, LUT를 디코딩하기 위한 디코더를 개략적으로 나타낸다.

도 14는 옥텟트의 8개의 정점의 위치를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0049]

도 4는 본 발명의 일 실시예에 따른 인코딩 방법의 흐름도를 나타낸다. 방법은 정점들의 격자로서 정의되는 LUT를 인코딩하기 위한 것이며, 여기서는 적어도 하나의 값, 예로서 컬러 값이 격자의 각각의 정점과 관련된다. 디코딩과 관련하여, 용어 "재구성" 및 "디코딩"은 매우 자주 동의어로서 사용된다. 아래에서, 용어 컬러 값은 RGB, YUV 또는 Y, Cb, Cr 값들과 같은 주어진 컬러 공간의 컬러 값들을 포함하며, CMF 파라미터들, 즉 행렬 파라미터들 및 오프셋 값들과 같은 컬러 변환을 나타내는 값들을 더 포함한다.

[0050]

단계 40에서, 현재 정점의 적어도 하나의 컬러 값이 이웃 정점들과 관련된 재구성 컬러 값들로부터 예측된다.

[0051]

일례로서, 정점 $V_0(c_{10}, c_{20})$ 을 값들 (V_{0c1}, V_{0c2}), 예로서 컬러 값들의 대응하는 쌍과 관련시키는 2D LUT가 도 5에 도시된다. 정점 V_0 과 관련된 값들은 예를 들어 값들 ($V_{i_{c1}}, V_{i_{c2}})_{i=1\dots 4}$ 의 대응하는 쌍을 갖는 공간적으로 이웃하는 정점들 $V_1(c_{11}, c_{21}), V_2(c_{12}, c_{22}), V_3(c_{13}, c_{23})$ 및 $V_4(c_{14}, c_{24})$ 로부터 예측된다. 예측 $P(P_{c1}, P_{c2})$ 는 다음과 같이 보간을 이용하여 계산된다.

[0052]

$$P_{c1} = 0.25 * (V_{1c1} + V_{2c1} + V_{3c1} + V_{4c1})$$

[0053]

$$P_{c2} = 0.25 * (V_{1c2} + V_{2c2} + V_{3c2} + V_{4c2})$$

[0054]

일 변형에 따르면, 2D LUT는 정점 $V_0(c_{10}, c_{20})$ 을 값들 (V_{0c1}, V_{0c2})의 쌍 대신에 파라미터들 ($m_{11}, m_{12}, m_{21}, m_{22}, o_1, o_2$)의 세트와 관련시킨다. 이러한 파라미터들의 세트는 다음과 같이 V_0 의 값들 (c_{10}, c_{20})으로부터 값들 (V_{0c1}, V_{0c2})를 재구성하는 데 사용될 수 있다.

[0055] $V0_{c1}=m11* c1_0+m12* c2_0 +o1$

[0056] $V0_{c2}=m21* c1_0+m22* c2_0 +o2$

[0057] 정점 $V0$ 과 관련된 파라미터들은 예를 들어 파라미터 $(Vi_{c1}, Vi_{c2})_{i=1\dots4}$ 의 대응하는 쌍을 갖는 공간적으로 이웃하는 정점들 $V1(c1_1, c2_1)$, $V2(c1_2, c2_2)$, $V3(c1_3, c2_3)$ 및 $V4(c1_4, c2_4)$ 로부터 예측된다. 예측 $P(Pc1, Pc2)$ 는 예를 들어 아래와 같이 보간을 이용하여 계산된다.

[0058] $Pc1= 0.25*(V1_{c1}+ V2_{c1}+ V3_{c1}+V4_{c1})$

[0059] $Pc2=0.25*(V1_{c2}+ V2_{c2}+ V3_{c2}+V4_{c2})$

[0060] 단계 42에서, 적어도 하나의 오차가 현재 정점의 적어도 하나의 컬러 값과 그의 예측 사이에서 결정되며, 또한 비트스트림(F) 내에 인코딩된다. 오차는 현재 정점의 적어도 하나의 컬러 값으로부터 그의 예측을 뺀으로써 결정된다. 인코딩은 통상적으로 엔트로피 인코딩을 포함한다. 일 변형에 따르면, 인코딩은 양자화기(q)를 이용하여 오차를 양자화하고, 양자화된 오차를 엔트로피 인코딩하는 것을 포함한다.

[0061] 일례로서, 2D LUT의 경우, 제1 및 제2 오차들이 정점 $V0$ 에 대해 계산된다. 제1 오차는 $(V0_{c1} - Pc1)$ 과 동일하며, 제2 오차는 $(V0_{c2} - Pc2)$ 와 동일하다. 이어서, 오차들 또는 양자화된 오차들 $(V0_{c1} - Pc1)/q$ 및 $(V0_{c2} - Pc2)/q$ 는 비트스트림(F) 내에 엔트로피 코딩된다. 엔트로피 코딩은 지수-Golomb, Huffman, CABAC("Context Adaptive Binary Arithmetic Coding"의 영문 두문자어)와 같은 전통적인 이진 코딩 기술들을 이용한다.

[0062] 단계 40 및 42는 LUT의 모든 정점들이 인코딩될 때까지 LUT의 더 많은 정점을 인코딩하도록 반복된다.

[0063] 옵션으로서, 인코딩 방법은 양자화기 값(q)을 비트스트림(F) 내에 인코딩하는 단계를 포함한다. 현재 정점의 적어도 하나의 컬러 값이 다른 정점들의 예측에 사용되도록 더 재구성된다. 각각의 재구성된 적어도 하나의 컬러 값은 양자화기가 사용되지 않는 경우에, 즉 $q=1$ 인 경우에 대응하는 오리지널 컬러 값과 동일하다. 그렇지 않은 경우, 적어도 하나의 재구성 컬러 값은 오차를 역양자화하고, 역양자화된 오차를 예측에 더함으로써 결정된다.

[0064] 일 변형에 따르면, 인코딩 방법은 정점의 적어도 하나의 컬러 값을 예측하는 데 사용되는 보간 타입을 비트스트림(F) 내에 인코딩하는 단계를 더 포함한다. 더 정확하게는, 보간 타입을 식별하는 비트스트림으로부터 인덱스가 디코딩된다. 일례로서, 인덱스 0은 이선형 보간을 식별하고, 인덱스 1은 최근접 이웃 보간을 식별하고, 인덱스 2는 2개의 최근접 이웃을 이용하는 선형 보간을 식별한다.

[0065] 일 변형에 따르면, LUT의 모든 정점들이 비트스트림(F) 내에 인코딩되지는 않는다. 예를 들어, 정점의 모든 오차들 또는 모든 양자화된 오차들의 절대값이 임계치(TH) 아래인 경우에는 어떠한 오차도 그 정점에 대해 인코딩되지 않으며, 예로서 $TH=0$ 또는 $TH=1$ 이다. 따라서 이진 플래그가 각각의 정점에 대해 비트스트림 내에 인코딩되며, 이러한 이진 플래그는 적어도 하나의 오차가 그 정점에 대해 인코딩되는지 또는 오차들 중 어느 것도 인코딩되지 않고, 따라서 0인 것으로 추정되는지를 지시한다.

[0066] 일 변형에 따르면, 이진 플래그가 각각의 정점의 각각의 컬러 값에 대해 인코딩되며, 이러한 이진 플래그는 오차가 그 컬러 값에 대해 인코딩되는지 또는 오차가 인코딩되지 않고 0인 것으로 추정되는지를 지시한다.

[0067] LUT의 크기도 옵션으로서 비트스트림 내에 인코딩된다.

[0068] 도 6은 본 발명의 다른 실시예에 따른 인코딩 방법의 흐름도를 나타낸다. 이 방법은 옥트리를 이용하여 정점들의 격자로서 정의된 3D LUT를 인코딩하기 위한 것이며, 여기서는 컬러 값들의 삼중쌍과 같은 n 개의 컬러 값들의 세트가 격자의 각각의 정점과 관련되고, 여기서 n 은 정수 ≥ 1 이다. 옥트리는 도 7에 도시된 바와 같이 3D 컬러 공간을 8개의 옥텟트로 재귀적으로 세분함으로써 그를 분할하기 위한 것이다. 레벨 N 의 옥텟트는 레벨 $N-1$ 에서 그의 부모 옥텟트를 갖는다. 옥트리들은 큐드트리들의 3D 유사물이다. 이 인코딩 방법은 현재 옥텟트에 속하는 격자의 현재 정점 V 의 인코딩을 위해 개시된다. 아래에서 $n=3$ 이다.

[0069] 단계 50에서, 좌표들 (r, g, b)의 현재 정점 V 와 관련된 3개의 컬러 값 (V_r, V_g, V_b) 각각은 이웃 정점들, 즉 현재 정점의 부모 옥텟트에 속하는 정점들과 관련된 재구성 컬러 값들로부터 예측된다. 표기법을 간소화하기 위해 ($c1, c2, c3$) 대신에 (r, g, b)가 사용된다. 그러나, 본 발명은 (R, G, B) 컬러 공간으로 한정되지

않는다. 본 발명은 (Y,U,V), (Y,Cb,Cr) 등의 컬러 공간 표현들에 적용될 수 있다.

[0070] 따라서, 예를 들어 도 8에 도시된 바와 같은 삼선형 보간을 이용하여 각각의 컬러 값에 대해 예측이 결정된다.

$$\bar{V}_r = K \times \sum_{i=0,1} \sum_{j=0,1} \sum_{k=0,1} s_i(r) \times s_j(g) \times s_k(b) \times \text{LUT}[r_i][g_j][b_k].r$$

[0072] 여기서, $i=0, 1$, $j=0, 1$ 및 $k=0, 1$ 을 갖는 (r_i, g_j, b_k) 는 3D 컬러 공간 내의 부모 옥텟트의 정점들의 좌표들이고, (r, g, b) 는 현재 정점의 좌표들이고, $\text{LUT}[r_i][g_j][b_k].r$ 은 정점 (r_i, g_j, b_k) 와 관련된 제1 컬러 값이고,

$$K = \frac{1}{(r_1 - r_0) \times (g_1 - g_0) \times (b_1 - b_0)}$$

[0073] $s_0(t) = t_1 - t$ 및 $s_1(t) = t - t_0$ 이고, $t = r, g$ 또는 b 이다.

[0075] 동일한 식들이 g 및 b 에 대해 사용된다. 프리즘, 피라미드 또는 사면체와 같은 다른 타입의 보간이 사용될 수 있다.

[0076] 격자는 도 9에 도시된 바와 같이 반드시 균일하지는 않다.

[0077] 단계 52에서, 현재 정점에 대해, 각각의 컬러 컴포넌트에 대해 하나씩 3개의 오차: $\text{res}_r = (V_r - \bar{V}_r)$, $\text{res}_g = (V_g - \bar{V}_g)$ 및 $\text{res}_b = (V_b - \bar{V}_b)$ 가 계산된다.

[0078] 이어서, 오차들은 비트스트림 내에 엔트로피 코딩되거나, 엔트로피 코딩되기 전에 양자화된다. 엔트로피 코딩은 지수-Golomb, Huffman, CABAC("Context Adaptive Binary Arithmetic Coding"의 영문 두문자어)와 같은 전통적인 이진 코딩 기술들을 이용한다.

[0079] 레벨 0의 옥텟트는 부모 옥텟트를 갖지 않는다. 같은 방식으로, 코딩 옥텟트 리스트 내의 제1 옥텟트는 선행 옥텟트를 갖지 않는다. 결과적으로, 이러한 옥텟트의 현재 정점 V 와 관련된 3개의 컬러 값 (V_r, V_g, V_b)는 알려진 컬러 값, 예로서 값 128로부터 예측된다. 일 변형에 따르면, 상이한 알려진 값들이 레벨 0의 옥텟트의 상이한 정점들에 대해 사용된다. 다른 변형에 따르면, 알려진 컬러 값들로부터의 예측은 레벨 0의 옥텟트가 아닌 다른 옥텟트들에 대해서도 행해진다. 일 변형에 따르면, 이러한 옥텟트의 현재 정점 V 와 관련된 3개의 컬러 값 (V_r, V_g, V_b) 각각은 동일 옥텟트에 속하는 이웃 정점들과 관련된 이미 재구성된 컬러 값들로부터 예측된다.

[0080] 옵션으로서, 인코딩 방법은 양자화기 값(q)을 비트스트림 내에 인코딩하는 단계를 포함한다. 현재 정점의 3개의 컬러 값이 더 재구성되고, 다른 정점들의 예측에 사용될 수 있다. 각각의 재구성 컬러 값은 양자화기가 사용되지 않는 경우에 ($q=1$) 오리지널 컬러 값과 동일하다. 그렇지 않은 경우, 각각의 재구성 컬러 값은 대응하는 오차를 역양자화하고, 역양자화된 오차를 대응하는 예측에 더함으로써 결정된다.

[0081] 일 변형에 따르면, 인코딩 방법은 정점의 적어도 하나의 컬러 값을 예측하는 데 사용되는 보간 타입을 비트스트림 내에 인코딩하는 단계를 더 포함한다. 더 정확하게는, 보간 타입을 식별하는 인덱스가 비트스트림 내에 인코딩된다. 일례로서, 인덱스 0은 삼선형 보간을 식별하고, 인덱스 1은 프리즘 보간을 식별하고, 인덱스 2는 피라미드 보간을 식별하고, 인덱스 3은 사면체 보간을 식별한다.

[0082] 일 변형에 따르면, LUT의 모든 정점들이 비트스트림 내에 인코딩되지는 않는다. 예를 들어, 정점의 모든 오차들 또는 모든 양자화된 오차들의 절대값이 임계치(TH) 아래인 경우에는 어떠한 오차도 그 정점에 대해 인코딩되지 않으며, 예로서 $TH=0$ 또는 $TH=1$ 이다. 따라서 이진 플래그가 각각의 정점에 대해 비트스트림 내에 인코딩되며, 이러한 이진 플래그는 적어도 하나의 오차가 그 정점에 대해 인코딩되는지의 여부를 지시한다. 일 변형에 따르면, 이진 플래그가 각각의 정점의 각각의 컬러 값에 대해 인코딩되며, 이러한 이진 플래그는 오차가 그 컬러 값에 대해 인코딩되는지 또는 오차가 인코딩되지 않고, 0인 것으로 추정되는지를 지시한다.

[0083] LUT는 통상적으로 렌더링 표시 컬러 적용 또는 컬러 범위 스케일링 가능성과 같은 응용들에서 비디오의 픽처들을 변환하는 데 사용되도록 비트스트림 내에 인코딩된다. LUT는 인코딩되고, 비디오와 함께 전송될 수 있다. 인코딩 방법이 3D LUT의 일부 부분들이 응용에 의해 사용되지 않을 것이라는 것을 아는 경우, 3D LUT의 이러한

부분에 속하는 정점들은 인코딩되지 않는다. 같은 방식으로, 인코딩 방법이 3D LUT의 일부 부분들이 비디오의 최종 렌더링에 작은 영향을 준다는 것을 아는 경우, 3D LUT의 이러한 부분에 속하는 정점들은 인코딩되지 않는다.

[0084] 방법은 전체 3D LUT를 인코딩하기 위해 재귀적으로 적용된다. 이 경우, 옥트리의 모든 옥턴트들이 인코딩된다. 현재 옥턴트의 모든 정점들이 인코딩되면, 현재 옥턴트의 자식 옥턴트의 정점들이 인코딩된다.

[0085] 일 변형에서, 3D LUT는 인코딩 전에 사전 처리된다. 이러한 경우, 분할 플래그가 옥트리 내의 각각의 옥턴트와 관련되며, 처음에 거짓으로 설정된다. 전처리 단계 동안, 각각의 옥턴트에 대해 분할 플래그 값들이 결정된다. 현재 옥턴트의 인코딩될 적어도 하나의 정점이 아마도 TH보다 크게 양자화된 적어도 하나의 오차를 갖는 경우, 그의 부모 옥턴트의 분할 플래그는 "참"으로 설정된다. 따라서, 레벨 N의 현재 옥턴트의 분할 플래그는 그의 직접 자식들(즉, 레벨 N+1의 자식들)이 재귀적으로 인코딩되는지 또는 아직 인코딩되지 않은 모든 그의 자식들(즉, 레벨 N+k(k>0)의 자식들)의 정점들의 모든 오차들이 0인 것으로 추정되는지를 지시한다.

[0086] 인코딩 단계 동안, 분할 플래그들 및 오차들이 비트스트림 내에 인코딩된다. 현재 옥턴트의 모든 정점들이 인코딩되면, 현재 옥턴트의 자식 옥턴트의 정점들은 현재 옥턴트 분할 플래그가 참인 경우에 인코딩된다. 2개의 옥턴트에 속하는 정점들은 우선적으로 한 번만 인코딩된다.

[0087] 3D LUT는 예를 들어 VPS("비디오 파라미터 세트"), SPS("시퀀스 파라미터 세트"), PPS("픽처 파라미터 세트") 내에 또는 AVC, HEVC, SVC 또는 SHVC 비디오 코딩 표준들에서 정의되는 바와 같은 하나의 SEI 메시지("보완 향상 정보") 내에 인코딩된다. 3D LUT는 예를 들어 아래에 정의되는 것과 같은 SEI 메시지 내에 인코딩된다. LUT의 크기(S)도 옵션으로서 비트스트림 내에 인코딩된다. S는 하나의 방향에서의 정점들의 수이다.

[0088] 일 변형에 따르면, 위에서 개시된 바와 같이, n개의 관련 컬러 값을 갖는 정점들로 구성되는 하나의 3D LUT를 인코딩하는 대신에 1개의 컬러 값을 갖는 정점들로 구성되는 n개의 3D LUT가 인코딩되며, 예로서 n=3이다.

[0089] 일 변형에 따르면, 3D LUT 크기가 S=2일 때, 디코딩된 3D LUT로부터 3x3 이득 행렬 플러스 오프셋을 계산하여 3D LUT 대신에 CMF로서 사용하도록 디코더에 지시하는 플래그가 인코딩된다.

[0090] 다른 변형에 따르면, n개의 관련 컬러 값을 갖는 정점들로 구성되는 하나의 3D LUT를 이용하여, 국지적으로 변환된 컬러 변환의 파라미터들을 인코딩한다. 예를 들어, 3D LUT의 각각의 정점은 컬러 변환을 나타내는 12개의 컬러 값을 관련된다. 3개의 컬러 값 (V_r , V_g , V_b)를 현재 정점 V와 관련시키는 대신, 컬러 변환을 나타내는 12개의 컬러 값이 정점 $V(r, g, b)$ 와 관련되며, $i=0, 1$ 또는 2 를 갖는 12개의 컬러 값 $[a_i, b_i, c_i, o_i]$ 는 아래의 식이 입증되도록 정의된다.

$$\begin{pmatrix} V_r \\ V_g \\ V_b \end{pmatrix} = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} + \begin{pmatrix} o_0 \\ o_1 \\ o_2 \end{pmatrix}$$

[0091] 일 변형에 따르면, $i=0, 1$ 또는 2 를 갖는 파라미터들 $[a_i, b_i, c_i]$ 만이 정점들과 관련된다.

[0093] 사실상, 크기 2의 3D LUT(8개의 정점을 갖는 하나의 단일 옥턴트)의 경우, 3D LUT가 아래의 식에 의해 정의되는 이득-오프셋 모델과 동등하도록 정점들의 3개의 컬러 값이 선택될 수 있다.

$$\begin{pmatrix} y' \\ u' \\ v' \end{pmatrix} = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} y \\ u \\ v \end{pmatrix} + \begin{pmatrix} o_0 \\ o_1 \\ o_2 \end{pmatrix}$$

[0095] 그렇게 하기 위해, 8개의 정점 A 내지 H의 3개의 컬러 값 $(y_X, u_X, v_X)_{X=A, \dots, H}$ 를 아래의 값들로 설정해야 한다.

$$\left\{ \begin{array}{l} y_A = 255 \cdot a_0 \\ y_B = 255 \cdot a_0 + o_0 \\ y_C = 255 \cdot b_0 + o_0 \\ y_D = 255 \cdot (a_0 + b_0) + o_0 \\ y_E = 255 \cdot c_0 + o_0 \\ y_F = 255 \cdot (a_0 + c_0) + o_0 \\ y_G = 255 \cdot (b_0 + c_0) + o_0 \\ y_H = 255 \cdot (a_0 + b_0 + c_0) + o_0 \end{array} \right. \quad \left\{ \begin{array}{l} u_A = 255 \cdot a_1 \\ u_B = 255 \cdot a_1 + o_1 \\ u_C = 255 \cdot b_1 + o_1 \\ u_D = 255 \cdot (a_1 + b_1) + o_1 \\ u_E = 255 \cdot c_1 + o_1 \\ u_F = 255 \cdot (a_1 + c_1) + o_1 \\ u_G = 255 \cdot (b_1 + c_1) + o_1 \\ u_H = 255 \cdot (a_1 + b_1 + c_1) + o_1 \end{array} \right.$$

$$\left\{ \begin{array}{l} v_A = 255 \cdot a_2 \\ v_B = 255 \cdot a_2 + o_2 \\ v_C = 255 \cdot b_2 + o_2 \\ v_D = 255 \cdot (a_2 + b_2) + o_2 \\ v_E = 255 \cdot c_2 + o_2 \\ v_F = 255 \cdot (a_2 + c_2) + o_2 \\ v_G = 255 \cdot (b_2 + c_2) + o_2 \\ v_H = 255 \cdot (a_2 + b_2 + c_2) + o_2 \end{array} \right.$$

[0096]

K는 각각의 컬러 값을 나타내는 데 사용되는 비트들의 수에 의존하는 상수이다. 컬러 값들이 8개 비트로 표현되는 경우에 K=255이고, 컬러 값들이 10개 비트로 표현되는 경우에 K=1023이고, 기타 등등이다. 그러면, 삼선형 보간은 이득-오프셋 모델과 등가이다.

[0097]

도 10은 본 발명의 일 실시예에 따른 디코딩 방법의 흐름도를 나타낸다.

[0098]

단계 140에서, 적어도 하나의 오차가 비트스트림(F)으로부터 디코딩된다. 디코딩은 통상적으로 엔트로피 디코딩을 포함한다. 일 변형에 따르면, 디코딩은 양자화된 오차의 엔트로피 디코딩 및 양자화기(q)를 이용하는 양자화된 오차의 역 양자화를 포함한다. 엔트로피 디코딩은 지수-Golomb, Huffman, CABAC("Context Adaptive Binary Arithmetic Coding"의 영문 두문자어)와 같은 전통적인 이진 디코딩 기술들을 이용한다.

[0099]

옵션으로서, 디코딩 방법은 비트스트림(F)으로부터 양자화기 값(q)을 디코딩하는 단계를 포함한다.

[0100]

일 변형에 따르면, LUT의 모든 정점들이 비트스트림(F) 내에 인코딩되지는 않는다. 옵션으로서, 이진 플래그가 각각의 정점에 대해 비트스트림(F)으로부터 디코딩되며, 이 이진 플래그는 적어도 하나의 오차가 그 정점에 대해 인코딩되는지의 여부를 지시한다. 오차가 인코딩되지 않는 경우, 오차(들)는 이 정점에 대해 0인 것으로 추정된다.

[0101]

단계 142에서, 현재 정점의 적어도 하나의 컬러 값이 이웃 정점들과 관련된 재구성 컬러 값들로부터 예측된다. 일례로서, 정점 V0(c1, c2)을 컬러 값들 (V0_{c1}, V0_{c2})의 대응하는 쌍과 관련시키는 2D LUT가 도 5에 도시된다. 정점 V0와 관련된 컬러 값들은 공간적으로 이웃하는 정점들(V1, V2, V3, V4)로부터 예측된다. 일례로서, 예측자 P(Pc1, Pc2)가 아래와 같이 보간을 이용하여 계산된다.

[0102]

Pc1= 0.25*(V1_{c1}+ V2_{c1}+ V3_{c1}+V4_{c1})

[0103]

Pc2=0.25*(V1_{c2}+ V2_{c2}+ V3_{c2}+V4_{c2})

[0104]

단계 144에서, 정점이 재구성된다. 더 정확하게는, 현재 정점의 적어도 하나의 컬러 값이 그의 예측 및 디코딩된 적어도 하나의 오차로부터 재구성된다.

[0105]

일례로서, 2D LUT의 경우, 2개의 오차 R_{c1} 및 R_{c2}가 현재 정점 V0에 대해 디코딩된다. 따라서, 현재 정점은 그의 2개의 컬러 값을 (R_{c1}+Pc1) 및 (R_{c2}+Pc2)로서 계산함으로써 재구성된다.

- [0107] 일 변형에 따르면, 디코딩 방법은 정점의 적어도 하나의 컬러 값을 예측하는 데 사용되는 보간 탑입을 비트스트림(F)으로부터 디코딩하는 단계를 더 포함한다. 더 정확하게는, 보간 탑입을 식별하는 인덱스가 비트스트림으로부터 디코딩된다. 일례로서, 인덱스 0은 이선형 보간을 식별하고, 인덱스 1은 최근점 정점 값 보간을 식별한다.
- [0108] LUT의 크기도 옵션으로서 비트스트림으로부터 디코딩된다.
- [0109] 도 11은 본 발명의 다른 실시예에 따른 디코딩 방법의 흐름도를 나타낸다. 이 방법은 옥트리를 이용하여 정점들의 격자로서 정의된 3D LUT를 디코딩하기 위한 것이며, 여기서는 컬러 값들의 삼중쌍이 격자의 각각의 정점과 관련된다. 디코딩 방법은 현재 옥턴트에 속하는 격자의 현재 정점 V를 디코딩하기 위해 개시된다.
- [0110] 단계 150에서, 3개의 오차 res_r , res_g , res_b 가 비트스트림(F)으로부터 디코딩된다. 디코딩은 통상적으로 엔트로피 디코딩을 포함한다. 일 변형에 따르면, 디코딩은 양자화된 오차의 엔트로피 디코딩 및 양자화기(q)를 이용하는 양자화된 오차의 역 양자화를 포함한다. 엔트로피 디코딩은 지수-Golomb, Huffman, CABAC("Context Adaptive Binary Arithmetic Coding"의 영문 두문자어)와 같은 전통적인 이진 디코딩 기술들을 이용한다.
- [0111] 옵션으로서, 디코딩 방법은 비트스트림(F)으로부터 양자화기 값(q)을 디코딩하는 단계를 포함한다.
- [0112] 일 변형에 따르면, LUT의 모든 정점들이 비트스트림(F) 내에 인코딩되지는 않는다. 옵션으로서, 이진 플래그가 각각의 정점에 대해 비트스트림(F)으로부터 디코딩되며, 이 이진 플래그는 적어도 하나의 오차가 그 정점에 대해 인코딩되는지의 여부를 지시한다. 오차가 인코딩되지 않는 경우, 오차(들)는 이 정점에 대해 0인 것으로 추정된다.
- [0113] 단계 152에서, 좌표들 (r, g, b)의 현재 정점 V와 관련된 3개의 컬러 값 (V_r , V_g , V_b) 각각이 이웃 정점들, 즉 현재 옥턴트의 부모 옥턴트에 속하는 정점들과 관련된 재구성 컬러 값들로부터 예측된다. 표기법을 간소화하기 위해 (c1, c2, c3) 대신 (r, g, b)가 사용된다. 그러나, 본 발명은 (R, G, B) 컬러 공간으로 한정되지 않는다. 본 발명은 (Y, U, V), (Y, Cb, Cr) 등의 컬러 공간 표현들에 적용될 수 있다.
- [0114] 따라서, 각각의 컬러 값에 대해 예측이 결정된다.
- [0115] 따라서, 예를 들어 도 8에 도시된 바와 같은 삼선형 보간을 이용하여 각각의 컬러 값에 대해 예측이 결정된다.
- [0116]
$$\bar{V}_r = K \times \sum_{i=0,1} \sum_{j=0,1} \sum_{k=0,1} s_i(r) \times s_j(g) \times s_k(b) \times LUT[r_i][g_j][b_k].r$$
- [0117] 여기서 $i=0, 1$, $j=0, 1$ 및 $k=0, 1$ 을 갖는 (r_i , g_j , b_k)는 3D 컬러 공간 내의 부모 옥턴트의 정점들의 좌표들이고,
- [0118] (r , g , b)는 현재 정점의 좌표들이고,
- [0119] $LUT[r_i][g_j][b_k].r$ 는 정점 (r_i , g_j , b_k)와 관련된 제1 컬러 값이고,
- [0120]
$$K = \frac{1}{(r_1 - r_0) \times (g_1 - g_0) \times (b_1 - b_0)}$$
이고,
- [0121] $s_0(t) = t_1 - t$ 및 $s_1(t) = t - t_0$ 이고, $t = r$, g 또는 b 이다.
- [0122] 동일한 식들이 g 및 b 에 대해 사용된다. 프리즘, 피라미드 또는 사면체와 같은 다른 타입의 보간이 이용될 수 있다. 격자는 도 9에 도시된 바와 같이 반드시 균일하지는 않다.
- [0123] 따라서, 단계 154에서, 3개의 컬러 값이 현재 정점에 대해 그들의 예측 및 대응하는 디코딩된 오차들 (res_r , res_g , res_b)로부터 재구성된다. 따라서, 현재 정점은 3개의 컬러 값을 ($res_r + \bar{V}_r$), ($res_g + \bar{V}_g$) 및 ($res_b + \bar{V}_b$)로서 계산함으로써 재구성된다.
- [0124] 레벨 0의 옥턴트는 부모 옥턴트를 갖지 않는다. 같은 방식으로, 코딩 옥턴트 리스트 내의 제1 옥턴트는 선행 옥턴트를 갖지 않는다. 결과적으로, 이러한 옥턴트의 현재 정점 V와 관련된 3개의 컬러 값 (V_r , V_g , V_b)는 알려진 컬러 값, 예로서 값 128로부터 예측된다. 일 변형에 따르면, 이러한 옥턴트의 현재 정점 V와 관련된 3개의

컬러 값 (V_r , V_g , V_b) 각각은 동일 옥텟트에 속하는 이웃 정점들과 관련된 이미 재구성된 컬러 값들로부터 예측된다.

[0125] 방법은 전체 3D LUT를 디코딩하기 위해 재귀적으로 적용된다. 2개의 옥텟트에 속하는 정점들은 우선적으로 한번만 디코딩된다.

[0126] 현재 옥텟트(레벨 N)의 모든 정점들이 디코딩되면, 현재 옥텟트의 자식(레벨 N+1) 옥텟트들의 정점들이 디코딩된다.

[0127] 일 변형에 따르면, 분할 플래그가 레벨 N의 현재 옥텟트에 대해 디코딩되며, 이 분할 플래그는 그의 직접 자식들(즉, 레벨 N+1의 자식들)이 재귀적으로 디코딩되는지 또는 아직 디코딩되지 않은 모든 자식들(즉, 레벨 N+k ($k > 0$)의 자식들)의 정점들의 모든 오차들이 0인 것으로 추정되는지를 지시한다.

[0128] 일 변형에 따르면, 디코딩 방법은 정점의 적어도 하나의 컬러 값을 예측하는 데 사용되는 보간 타입을 비트스트림(F)으로부터 디코딩하는 단계를 더 포함한다. 더 정확하게는, 보간 타입을 식별하는 인덱스가 비트스트림으로부터 디코딩된다. 일례로서, 인덱스 0은 삼선형 보간을 식별하고, 인덱스 1은 프리즘 보간을 식별하고, 인덱스 2는 피라미드 보간을 식별하고, 인덱스 3은 사면체 보간을 식별한다.

[0129] 3D LUT는 예를 들어 VPS, SPS, PPS로부터 또는 AVC, HEVC, SVC 또는 SHVC 비디오 코딩 표준들에서 정의되는 바와 같은 하나의 SEI 메시지 내에서 디코딩된다. LUT의 크기도 옵션으로서 비트스트림으로부터 디코딩된다. 3D LUT는 예를 들어 아래에 정의되는 것과 같은 SEI 메시지 내에서 디코딩된다.

[0130] 일 변형에 따르면, n 개의 컬러 값을 갖는 정점들로 구성되는 하나의 3D LUT를 디코딩하는 대신에 1개의 컬러 값을 갖는 정점들로 구성되는 n 개의 3D LUT가 디코딩되며, 예로서 $n=3$ 이다.

[0131] 일 변형에 따르면, 3D LUT 크기가 $S=2$ 일 때, 디코딩된 3D LUT로부터 3×3 이득 행렬 플러스 오프셋을 계산하여 3D LUT 대신에 CMF로서 사용하도록 디코더에 지시하는 플래그가 디코딩된다.

[0132] 도 4 및 6에 따른 인코딩 방법과 관련하여 개시되는 것과 동일한 변형들이 디코딩 측에 적용될 수 있다.

[0133] 도 4, 6, 10 및 11에서, 표시된 박스들은 순전히 기능적인 엔티티들이며, 이들은 물리적인 개별 엔티티들에 반드시 대응하지는 않는다. 이 분야의 기술자에 의해 인식되듯이, 본 발명의 원리들의 양태들은 시스템, 방법 또는 컴퓨터 판독 가능 매체로서 구현될 수 있다. 따라서, 본 발명의 원리들의 양태들은 완전 하드웨어 실시예, (펌웨어, 상주 소프트웨어, 마이크로코드 등을 포함하는) 완전 소프트웨어 실시예, 또는 본 명세서에서 모두가 일반적으로 "회로", "모듈" 또는 "시스템"으로 지칭될 수 있는 소프트웨어 및 하드웨어 양태들을 결합하는 실시예의 형태를 취할 수 있다. 또한, 본 발명의 원리들의 양태들은 컴퓨터 판독 가능 저장 매체의 형태를 취할 수 있다. 하나 이상의 컴퓨터 판독 가능 매체(들)의 임의의 조합이 이용될 수 있다.

[0134] 도면들 내의 흐름도 및/또는 블록도들은 본 발명의 다양한 실시예들에 따른 시스템들, 방법들 및 컴퓨터 프로그램 제품들의 가능한 구현들의 구성, 동작 및 기능을 예시한다. 이와 관련하여, 흐름도 또는 블록도들 내의 각각의 블록은 지정된 논리 기능(들)을 구현하기 위한 하나 이상의 실행 가능 명령어를 포함하는 모듈, 세그먼트, 또는 코드의 일부를 나타낼 수 있다. 일부 대안 구현들에서는 블록 내에 기재된 기능들이 도면들에 기재된 순서와 다르게 발생할 수 있다는 점에도 유의해야 한다. 예를 들어, 관련 기능에 따라, 연속적으로 도시된 2개의 블록은 실제로는 실질적으로 동시에 실행될 수 있거나, 블록들은 때때로 역순으로 실행될 수 있거나, 블록들은 대안 순서로 실행될 수 있다. 블록도들 및/또는 흐름도의 각각의 블록, 및 블록도들 및/또는 흐름도 내의 블록들의 조합들은 지정된 기능들 또는 동작들을 수행하는 특수 목적 하드웨어 기반 시스템들, 또는 특수 목적 하드웨어와 컴퓨터 명령어들의 조합들에 의해 구현될 수 있다는 점에도 유의할 것이다. 명확히 설명되지는 않지만, 본 발명의 실시예들은 임의의 조합 또는 하위 조합으로 이용될 수 있다.

[0135] 3D LUT와 같은 LUT를 인코딩하는 비트스트림도 개시된다. 도면의 인코딩 방법에 의해 생성되는 비트스트림은 정점들의 격자로서 정의되는 적어도 LUT를 인코딩하며, 적어도 하나의 컬러 값이 격자의 각각의 정점과 관련되고, 상기 비트스트림은 현재 정점의 적어도 하나의 컬러 값과 그의 예측 사이에서 계산되는 적어도 하나의 오차를 그 안에 인코딩하는 것을 포함한다.

[0136] ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11의 JCT-VC(Joint Collaborative Team on Video Coding)의 문서 JCTVC-L1003에서 정의되는 HEVC 코딩 표준 또는 ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11의 JCT-VC(Joint Collaborative Team on Video Coding)의 문서 JCTVC-L1008에서 정의되는 HEVC 코딩 표준의 스케일링 가능 확장

인 SHVC 코딩 표준의 프레임워크 내에서 일 실시예가 제안된다. 표준은 코딩된 데이터의 임의의 스트림이 이 표준에 맞도록 따라야 하는 선택지를 정의한다. 선택지는 특히 다양한 정보 아이템들(예로서, 시퀀스 내에 포함된 픽처들, 모션 벡터들 등에 관한 데이터)이 어떻게 코딩되는지를 정의한다. SHVC 코딩 표준과 관련하여, LUT는 PPS 또는 VPS 내에 인코딩될 수 있다. 선택지 요소 `use_color_prediction`은 표 1에 나타난 바와 같이 현재 픽처에서의 컬러 예측의 사용을 지시하는 데 사용된다.

[0137] 표 1: 예측 파라미터들의 시그널링

use_color_prediction
<code>if(use_color_prediction)</code>
3D_LUT_color_data()
<code>rbsp_trailing_bits()</code>

[0138]

`use_color_prediction` 플래그가 '1'인 경우, `3D_LUT_color_data` 함수가 표 2에 나타난 바와 같이 3D LUT 데이터를 시그널링하도록 호출된다.

[0140]

표 2: 3D LUT 컬러 데이터의 코딩

<code>3D_LUT_color_data()</code>	서술자
<code>nbpCode</code>	<code>u(3)</code>
<code>coding_octant(0,0,0,0)</code>	
}	

[0141]

`nbpCode`는 `nbpCode`의 주어진 값에 대한 표 4에 목록화된 바와 같은 3D LUT 크기를 지시한다. 양자화기 값은 `3D_LUT_color_data()` 함수에 의해 인코딩될 수 있다.

[0143]

일 변형에 따르면, `3D_LUT_color_data()`는 표 3에서 아래와 같이 정의된다.

[0144]

표 3: 3D LUT 컬러 데이터의 코딩

<code>3D_LUT_color_data()</code>	서술자
<code>nbpCode</code>	<code>u(3)</code>
<code>NbitsPerSample</code>	<code>u(5)</code>
<code>coding_octant(0,0,0,0)</code>	
}	

[0145]

`nbpCode`는 `nbpCode`의 주어진 값에 대한 표 4에 목록화된 바와 같은 3D LUT 크기를 지시한다. 양자화기 값은 `3D_LUT_color_data()` 함수에 의해 인코딩될 수 있다.

[0147]

`NbitsPerSample`은 컬러 값들을 나타내는 데 사용되는 비트들의 수를 지시한다.

[0148]

표 4: `nbpCode`의 해석

nbpCode	3D LUT 크기
0	2
1	3
2	5
3	9
4	17
5	33

[0149]

옥턴트(계층, y,u,v)의 디코딩은 표 5에 나타난 바와 같이 재귀 함수이다. 각각의 옥턴트는 오차 컬러 값들이 인코딩될지 또는 모두 0인 것으로 추정될지를 지시하는 플래그 (`encoded_flag[i]`)와 관련된 8개의 정점으로 구성된다. 컬러 값들은 컬러 값들의 예측에 오차들을 더함으로써 재구성된다. 컬러 값들의 예측은 예를 들어 `layer_id-1`의 8개의 이웃 정점의 삼선형 보간을 이용하여 계산된다.

[0151] 표 5: coding_octant()에 대한 신택스 요소들

선택스	서술자
coding_octant (layer_id, v,u,v) {	
for (i = 0; i < 8 ; i++) {	
encoded_flag[i]	u(1)
if (encoded_flag[i]) {	
resY[i]	ue(v)
resU[i]	ue(v)
resV[i]	ue(v)
}	
}	
split_flag	u(1)
if (split_flag) {	
for (i = 0; i < 8 ; i++) {	
coding_octant (layer_id+1,	
v+dy[i], u+du[i], v+dv[i])	
}	
}	
}	

[0152]

[0153] 다른 유리한 실시예에 따르면, LUT는 SEI 메시지 내에 인코딩된다(SEI는 "보완 항상 정보"를 나타낸다). HEVC 표준은 그의 부록 D에서 SEI라고 하는 추가 정보를 코딩하는 방법을 정의한다. 이러한 추가 정보는 신택스에서 payloadType이라고 하는 필드에 의해 참조된다. SEI 메시지들은 예를 들어 표시와 관련된 프로세스들을 돋는다. 디코딩 장치가 그의 사용에 필요한 기능들을 갖지 않는 경우에는 이러한 정보가 무시된다는 점에 유의 한다. 본 발명의 특정 실시예에 따르면, 3D LUT에 관한 추가 정보를 코딩하기 위해 새로운 타입의 SEI 메시지가 정의된다. 이러한 목적으로, 필드 payloadType에 대한 새로운 값이 아직 사용되지 않은 값을 중에서 정의된다(예를 들어, payloadType은 24이다).

[0154] SEI 데이터의 신택스(즉, sei_payload)는 아래의 방식으로 확장된다.

[0155] 표 6: 컬러 맵핑 SEI 메시지

선택스	서술자
color_mapping_info(payloadSize) {	
color_map_id	ue(v)
color_map_cancel_flag	u(1)
if (!color_map_cancel_flag) {	
color_map_repetition_period	ue(v)
color_description_present_flag	u(1)
If (color_description_present_flag) {	
color_primaries_input_id	u(8)
color_primaries_output_id	u(8)
}	
color_output_rgb	u(1)
lut_bit_depth_minus8	u(4)
3D LUT color data()	
}	
}	

[0156]

[0157] 일 변형에 따르면, SEI 메시지는 예를 들어 color_description_present_flag 뒤에 지시자 color_interpolator_id를 더 포함하며, 그의 값은 표 7에 지정된 바와같은 보간의 타입을 지시한다.

[0158] 표 7: 보간의 타입

color_interpolator_id	3D LUT 크기
0	삼선형
1	사면체

[0159]

[0160] 이러한 SEI 메시지는 특정 표시 환경들에 대한 맞춤화를 위한 출력되는 디코딩된 픽처들의 컬러 샘플들의 리맵핑을 가능하게 하기 위한 정보를 제공한다. 리맵핑 프로세스는 RGB 컬러 공간 내의 코딩된 샘플 값을 타겟 샘플 값들로 맵핑한다. 맵핑들은 루마 또는 RGB 컬러 공간 도메인에서 표현되며, 따라서 루마 컴포넌트에 또는 디코딩된 픽처의 컬러 공간 변환에 의해 생성되는 각각의 RGB 컴포넌트에 적용되어야 한다.

- [0161] 3D_LUT_color_data()가 표 2 또는 3에서 정의된다.
- [0162] 디코딩된 3D LUT는 예를 들어 SEI 메시지의 NAL 유닛 헤더의 인덱스 **nuh_layer_id**에 의해 식별된 계층에 속하는 디코딩된 픽처들에 적용된다(IITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11의 JCT-VC(Joint Collaborative Team on Video Coding)의 문서 JCTVC-L1003에서 정의되는 문서 HEVC 코딩 표준의 섹션 7.3.1.2 참조).
- [0163] **color_map_id**는 컬러 맵핑 모델의 목적을 식별하는 데 사용될 수 있는 식별 번호를 포함한다. **color_map_id**의 값들은 응용에 의해 결정되는 대로 사용될 수 있다. **color_map_id**는 상이한 표시 시나리오들에 적합한 컬러 맵핑 동작들을 지원하는 데 사용될 수 있다. 예를 들어, **color_map_id**의 상이한 값들은 상이한 표시 비트 깊이들에 대응할 수 있다.
- [0164] 1과 동일한 **color_map_cancel_flag**는 컬러 맵핑 정보 SEI 메시지가 출력 순서에 있어서의 임의의 이전의 컬러 맵핑 정보 SEI 메시지의 지속성을 제거한다는 것을 지시한다. 0과 동일한 **color_map_cancel_flag**는 컬러 맵핑 정보가 이어진다는 것을 지시한다.
- [0165] **color_map_repetition_period**는 컬러 맵핑 정보 SEI 메시지의 지속성을 지정하며, 동일 값의 **color_map_id**를 갖는 다른 컬러 맵핑 정보 SEI 메시지 또는 코딩된 비디오 시퀀스의 끝이 비트스트림 내에 존재해야 하는 픽처 순서 카운트 간격을 지정할 수 있다. 0과 동일한 **color_map_repetition_period**는 컬러 맵 정보가 현재의 디코딩된 픽처에만 적용된다는 것을 지정한다.
- [0166] 1과 동일한 **color_map_repetition_period**는 아래의 조건들 중 임의의 조건이 참일 때까지 컬러 맵 정보가 출력 순서에 있어서 지속된다는 것을 지정한다.
- 새로운 코딩된 비디오 시퀀스가 시작된다.
- [0167] - 동일 값의 **color_map_id**를 갖는 컬러 맵핑 정보 SEI 메시지를 포함하는 액세스 유닛 내의 픽처가 PicOrderCnt(CurrPic)로 표시되는 현재의 디코딩된 픽처의 (POC로서 알려진) 픽처 순서 카운트보다 큰 POC를 갖고서 출력된다.
- [0168] 0과 동일한 또는 1과 동일한 **color_map_repetition_period**는 동일 값의 **color_map_id**를 갖는 다른 컬러 맵핑 정보 SEI 메시지가 존재하거나 존재하지 않을 수 있다는 것을 지시한다.
- [0169] 1보다 큰 **color_map_repetition_period**는 아래의 조건들 중 임의의 조건이 참일 때까지 컬러 맵 정보가 지속된다는 것을 지정한다.
- 새로운 코딩된 비디오 시퀀스가 시작된다.
- [0170] - 동일 값의 **color_map_id**를 갖는 컬러 맵핑 정보 SEI 메시지를 포함하는 액세스 유닛 내의 픽처가 PicOrderCnt(CurrPic)보다 크거나 PicOrderCnt(CurrPic) + **color_map_repetition_period** 이하인 POC를 갖고서 출력된다.
- [0171] 1보다 **color_map_repetition_period**는 동일 값의 **color_map_id**를 갖는 다른 컬러 맵핑 정보 SEI 메시지가 PicOrderCnt(CurrPic)보다 크거나 PicOrderCnt(CurrPic) + **color_map_repetition_period** 이하인 POC를 갖고서 출력되는 액세스 유닛 내의 픽처에 대해 존재해야 하며, 그렇지 않을 경우에는 그러한 픽처의 출력 없이 비트스트림이 종료되거나, 새로운 코딩된 비디오 시퀀스가 시작된다는 것을 지시한다.
- [0172] 1과 동일한 **color_description_present_flag**는 **colour_primaries_input_id** 및 **colour_primaries_output_id**가 존재한다는 것을 지정한다. 0과 동일한 **color_description_present_flag**는 **colour_primaries_input_id** 및 **colour_primaries_output_id**가 존재하지 않는다는 것을 지정한다.
- [0173] **color_primaries_input_id**는 ISO 11664-1에 의해 지정되는 바와 같은 x 및 y의 CIE 1931 정의와 관련하여 표 8에서 지정되는 바와 같은 소스 원색들의 색도 좌표들을 지시한다.
- [0174] **color_primaries_output_id**는 3D 컬러 LUT가 적용되는 경우에 ISO 11664-1에 의해 지정되는 바와 같은 x 및 y의 CIE 1931 정의와 관련하여 표 8에서 지정되는 바와 같은 컬러 맵핑된 원색들의 색도 좌표들을 지시한다.
- [0175] 1과 동일한 **color_output_rgb**는 출력 컬러 샘플들이 루마 및 크로마 신호들이라는 것을 지정한다. 0과 동일한 **color_output_rgb**는 출력 컬러 샘플들이 녹색, 적색, 청색 값들이라는 것을 지정한다.
- [0176] **lut_bit_depth_minus8**은 3D LUT 샘플들의 비트 깊이를 지정한다.

[0179] **nbp_code**는 nbp_code의 주어진 값에 대해 표 4에 목록화된 바와 같은 3D LUT 크기 nbp를 지시한다.

[0180] 3D LUT 디코딩의 출력은 크기 nbp x nbp x nbp의 삼차원 어레이 LUT이다. 각각의 LUT 어레이 요소는 정점으로 지정되며, (*lut_bit_depth_minus8+8*)과 동일한 비트 깊이의 3개의 재구성 샘플 값(*recSamplesY*, *recSamplesU*, *recSamplesV*)과 관련된다. 정점 *lut[i][j][k]*는 $i \%(nbp \gg \text{layer_id})$, $j \%(nbp \gg \text{layer_id})$, $k \%(nbp \gg \text{layer_id})$ 의 값들이 0과 동일한 경우에 계층 *layer_id*에 속하는 것으로 지정된다. 하나의 정점이 여러 계층에 속할 수 있다. 계층 *layer_id*의 옥텟트는 *layer_id*에 속하는 8개의 이웃 정점으로 구성된다(도 14).

[0181] 옥텟트 (*layer_id*, *y,u,v*)의 디코딩은 재귀 함수이다. 각각의 옥텟트는 오차 컴포넌트 값들 (*resY[i]*, *resU[i]*, *resV[i]*)가 인코딩되는지 또는 모두 0인 것으로 추정되는지를 지시하는 플래그 (*encoded_flag[i]*)와 관련된 8개의 정점 ($i = 0, \dots, 7$)로 구성된다. 컴포넌트 값들은 컴포넌트 값들의 예측에 오차들을 더함으로써 재구성된다. 컴포넌트 값들의 예측은 *layer_id-1*의 8개의 이웃 정점의 삼선형 보간을 이용하여 계산된다. 재구성된 정점은 재구성된 것으로서 마킹된다.

[0182] 여기서, (*y+dy[i]*), (*u+du[i]*) 및 (*v+dv[i]*)는 ((*y,u,v*)를 제1 정점 ($i=0$) 좌표들로서 갖는) 현재 옥텟트의 8개의 자식 옥텟트 좌표(제1 3D 컬러 정점의 좌표들)이다. 주어진 계층에 대한 값을 *dy[i]*, *du[i]* 및 *dv[i]*가 표 9에 표시된다.

[0183]

표 8: 원색들(color_primaries_input_id 및 color_primaries_output_id)

값	원색들	설명
0	예약	ITU-T ISO/IEC에 의해 후속 사용됨
1	원색 x 0.300 녹색 0.150 청색 0.640 적색 0.3127 백색 D65	ITU-R Rec. BT.709-5 ITU-R Rec. BT.1361 conventional colour gamut system and extended colour gamut system IEC 61966-2-1 (sRGB or sYCC) IEC 61966-2-4 Society of Motion Picture and Television Engineers RP 177 (1993) Annex B
2	미지정	이미지 특성들이 알려지지 않거나 응용에 의해 결정됨
3	예약	ITU-T ISO/IEC에 의해 후속 사용됨
4	원색 x 0.21 녹색 0.14 청색 0.67 적색 0.310 백색 C	ITU-R Rec. BT.470-6 System M (historical) United States National Television System Committee 1953 Recommendation for transmission standards for colour television United States Federal Communications Commission Title 47 Code of Federal Regulations (2003) 73.682 (a) (20)
5	원색 x 0.29 녹색 0.15 청색 0.64 적색 0.3127 백색 D65	ITU-R Rec. BT.470-6 System B, G (historical) ITU-R Rec. BT.601-6 625 ITU-R Rec. BT.1358 625 ITU-R Rec. BT.1700 625 PAL and 625 SECAM
6	원색 x 0.310 녹색 0.155 청색 0.630 적색 0.3127 백색 D65	ITU-R Rec. BT.601-6 525 ITU-R Rec. BT.1358 525 ITU-R Rec. BT.1700 NTSC Society of Motion Picture and Television Engineers 170M (2004) (값 7과 기능적으로 동일)
7	원색 x 0.310 녹색 0.155 청색 0.630 적색 0.3127 백색 D65	Society of Motion Picture and Television Engineers 240M (1999) (값 6과 기능적으로 동일)
8	원색 x 0.243 녹색 0.145 청색 0.681 적색 0.310 백색 C	일반 막(발광체 C를 이용하는 컬러 펌터들)
9	원색 x 0.170 녹색 0.131 청색 0.708 적색 0.3127 백색 D65	Rec. ITU-R BT.2020
10..255	예약	ITU-T ISO/IEC에 의해 후속 사용됨

[0184]

표 9: layer = layer_id에 속하는 정점들에 대한 인덱스 i의 함수 내의 값들 dy[i], du[i] and dv[i]

i	dy[i]	du[i]	dv[i]
0	0	0	0
1	0	0	nbp >> layer_id
2	0	nbp >> layer_id	0
3	0	nbp >> layer_id	nbp >> layer_id
4	nbp >> layer_id	0	0
5	nbp >> layer_id	0	nbp >> layer_id
6	nbp >> layer_id	nbp >> layer_id	0
7	nbp >> layer_id	nbp >> layer_id	nbp >> layer_id

[0186]

layer = layer_id의 옥텟트에 속하는 정점 ((y+dy[i]), (u+du[i]), (v+dv[i]))에 대한 재구성 3D 컬러 LUT 샘

[0187]

풀들 (recSamplesY[i], recSamplesU[i], recSamplesV[i])는 $recSamplesY[i] = resY[i] + predSamplesY[i]$ 에 의해 주어지며, 여기서 $predSamplesY[i]$ 의 값은 현재 옥텟트를 포함하는 $layer = layer_id-1$ 의 옥텟트의 정점들과 관련된 삼선형 보간을 이용하여 도출된다.

[0188] 도 12는 인코더(1)의 예시적인 아키텍처를 나타낸다. 인코더는 인코딩 방법의 단계들을 실행하도록 구성된다. 인코더(1)는 데이터 및 주소 버스(64)에 의해 함께 링크되는 아래의 요소들을 포함한다:

- 예를 들어 DSP(즉, 디지털 신호 프로세서)인 마이크로프로세서(61)(또는 CPU);

[0189] - ROM(즉, 판독 전용 메모리)(62);

[0190] - RAM(즉, 랜덤 액세스 메모리)(63);

[0191] - 예를 들어 키보드, 마우스, 웹캠 등과 같은 하나 또는 여러 개의 I/O(입출력) 장치(65); 및

[0192] - 전원(66).

[0193] 일 변형에 따르면, 전원(66)은 인코더 외부에 위치한다. 도 12의 이러한 요소들 각각은 이 분야의 기술자에게 잘 알려져 있으며, 더 설명되지 않을 것이다. 언급된 메모리 각각에서, 본 명세서에서 사용되는 단어 "레지스터"는 언급된 메모리를 각각에서 낮은 용량(몇몇 이진 데이터)의 메모리 존은 물론, (전체 프로그램이 저장되거나, 계산된 데이터를 나타내는 데이터의 전부 또는 일부가 표시되는 것을 가능하게 하는) 큰 용량의 메모리 존도 지시한다. ROM(62)은 프로그램 및 (임계치(TH)와 같은) 인코딩 파라미터들을 포함한다. 본 발명에 따른 인코딩 방법의 알고리즘은 ROM(62) 내에 저장된다. 스위치 온될 때, CPU(61)는 프로그램(620)을 RAM 내에 업로드하고, 대응하는 명령어들을 실행한다.

[0194] RAM(63)은 레지스터 내에, CPU(61)에 의해 실행되고 인코더(1)의 스위치 온 후에 업로드되는 프로그램, 레지스터 내의 입력 데이터, 레지스터 내의 인코딩 방법의 상이한 상태에서의 인코딩된 데이터 및 레지스터 내의 인코딩을 위해 사용되는 다른 변수들을 포함한다.

[0195] 도 13은 디코더(2)의 예시적인 아키텍처를 나타낸다. 디코더는 디코딩 방법의 단계들을 실행하도록 구성된다. 디코더(2)는 데이터 및 주소 버스(74)에 의해 함께 링크되는 아래의 요소들을 포함한다:

- 예를 들어 DSP(즉, 디지털 신호 프로세서)인 마이크로프로세서(71)(또는 CPU);

[0196] - ROM(즉, 판독 전용 메모리)(72);

[0197] - RAM(즉, 랜덤 액세스 메모리)(73);

[0198] - 애플리케이션으로부터의 전송할 데이터의 수신을 위한 I/O 인터페이스(75); 및

[0199] - 배터리(76).

[0200] 일 변형에 따르면, 배터리(76)는 인코더 외부에 위치한다. 도 13의 이러한 요소들 각각은 이 분야의 기술자에게 잘 알려져 있으며, 더 설명되지 않을 것이다. 언급된 메모리 각각에서, 본 명세서에서 사용되는 단어 "레지스터"는 적은 용량(몇몇 비트)의 영역 또는 매우 큰 영역(예로서, 전체 프로그램 또는 많은 양의 수신 또는 디코딩된 데이터)에 대응할 수 있다. ROM(72)은 적어도 프로그램 및 디코더 파라미터들을 포함한다. 본 발명에 따른 디코딩 방법의 알고리즘은 ROM(72) 내에 저장된다. 스위치 온될 때, CPU(71)는 프로그램(720)을 RAM 내에 업로드하고, 대응하는 명령어들을 실행한다.

[0201] RAM(73)은 레지스터 내에, CPU(71)에 의해 실행되고 디코더(2)의 스위치 온 후에 업로드되는 프로그램, 레지스터 내의 입력 데이터, 레지스터 내의 디코딩 방법의 상이한 상태에서의 디코딩된 데이터 및 레지스터 내의 디코딩을 위해 사용되는 다른 변수들을 포함한다.

[0202] 본 명세서에서 설명되는 구현들은 예를 들어 방법 또는 프로세스, 장치, 소프트웨어 프로그램, 데이터 스트림 또는 신호 내에 구현될 수 있다. 단일 형태의 구현과 관련해서만 설명되는 경우에도(예를 들어, 방법 또는 장치로서만 설명되는 경우에도), 설명되는 특징들의 구현은 다른 형태들(예로서, 프로그램)로도 구현될 수 있다. 장치는 예를 들어 적절한 하드웨어, 소프트웨어 및 펌웨어로 구현될 수 있다. 방법들은 예를 들어 일반적으로 컴퓨터, 마이크로프로세서, 집적 회로 또는 프로그래밍 가능 논리 장치를 포함하는 처리 장치들을 지칭하는 프로세서와 같은 장치에서 구현될 수 있다. 프로세서들은 또한 예를 들어 컴퓨터, 셀 폰, 휴대용/개인용 디지털 단말기("PDA"), 및 최종 사용자들 사이의 정보의 통신을 용이하게 하는 다른 장치들과 같은 통신 장치들을 포함

한다.

[0205]

본 명세서에서 설명되는 다양한 프로세스들 및 특징들의 구현들은 다양한 상이한 장비 또는 애플리케이션, 특히 예를 들어 장비 또는 애플리케이션에서 구현될 수 있다. 그러한 장비의 예는 인코더, 디코더, 디코더로부터의 출력을 처리하는 포스트-프로세서, 인코더에 입력을 제공하는 프리-프로세서, 비디오 코더, 비디오 디코더, 비디오 코덱, 웹 서버, 셋톱 박스, 랩탑, 개인용 컴퓨터, 셀 폰, PDA 및 다른 통신 장치들을 포함한다. 명백해야 하듯이, 장비는 이동할 수 있고, 이동 차량 내에 설치될 수도 있다.

[0206]

계다가, 방법들은 프로세서에 의해 실행되는 명령어들에 의해 구현될 수 있으며, 그러한 명령어들(및/또는 구현에 의해 생성되는 데이터 값들)은 예를 들어 집적 회로, 소프트웨어 캐리어 또는 다른 저장 장치, 예로서 하드 디스크, 컴팩트 디스크("CD"), 광 디스크(예로서, 종종 디지털 다기능 디스크 또는 디지털 비디오 디스크로 지칭되는 DVD 등), 랜덤 액세스 메모리("RAM") 또는 판독 전용 메모리("ROM")와 같은 프로세서 판독 가능 매체 상에 저장될 수 있다. 명령어들은 프로세서 판독 가능 매체 상에 유형적으로 구현되는 애플리케이션 프로그램을 형성할 수 있다. 명령어들은 예를 들어 하드웨어, 펌웨어, 소프트웨어 또는 이들의 조합 내에 존재할 수 있다. 명령어들은 예를 들어 운영 체제, 개별 애플리케이션 또는 이들의 조합 내에서 발견될 수 있다. 따라서, 프로세서는 예를 들어 프로세스를 수행하도록 구성되는 장치 및 프로세스를 수행하기 위한 명령어들을 갖는 프로세서 판독 가능 매체(예로서, 저장 장치)를 포함하는 장치 양자로서 특화될 수 있다. 또한, 프로세서 판독 가능 매체는 명령어들에 더하여 또는 그 대신에 구현에 의해 생성된 데이터 값들을 저장할 수 있다.

[0207]

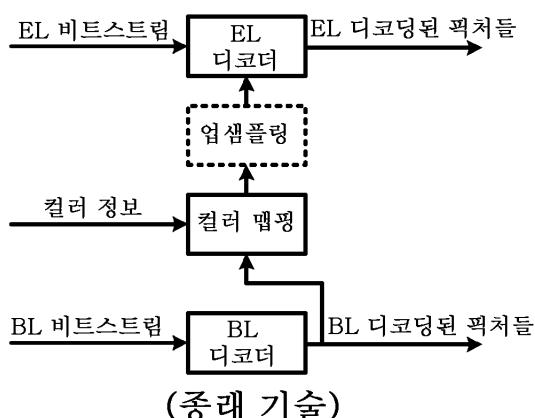
이 분야의 기술자에게 명백하듯이, 구현들은 예를 들어 저장 또는 전송될 수 있는 정보를 운반하도록 포맷팅되는 다양한 신호들을 생성할 수 있다. 정보는 예를 들어 방법을 수행하기 위한 명령어들, 또는 설명되는 구현들 중 하나에 의해 생성되는 데이터를 포함할 수 있다. 예를 들어, 신호는 설명되는 실시예의 신택스를 기록 또는 판독하기 위한 규칙들을 데이터로서 운반하거나 설명되는 실시예에 의해 기록되는 실제 신택스 값들을 데이터로서 운반하도록 포맷팅될 수 있다. 그러한 신호는 예를 들어 (예를 들어, 스펙트럼의 무선 주파수 부분을 이용하여) 전자기파로서 또는 기저대역 신호로서 포맷팅될 수 있다. 포맷팅은 예를 들어 데이터 스트림을 인코딩하고, 인코딩된 데이터 스트림으로 캐리어를 변조하는 것을 포함할 수 있다. 신호가 운반하는 정보는 예를 들어 아날로그 또는 디지털 정보일 수 있다. 신호는 공지된 바와 같은 다양한 상이한 유선 또는 무선 링크를 통해 전송될 수 있다. 신호는 프로세서 판독 가능 매체 상에 저장될 수 있다.

[0208]

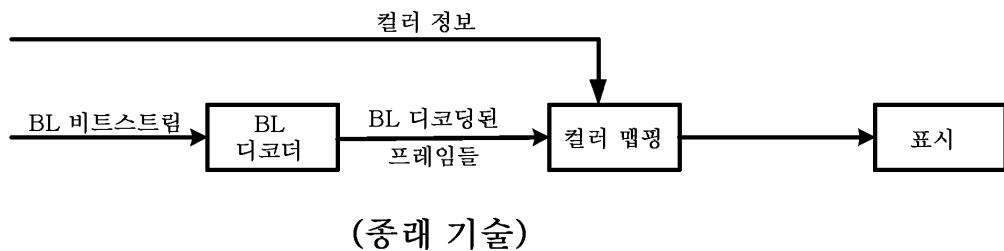
다양한 구현들이 설명되었다. 그러나, 다양한 변경들이 이루어질 수 있다는 것을 이해할 것이다. 예를 들어, 상이한 구현들의 요소들은 다른 구현들을 생성하도록 결합, 보완, 변경 또는 제거될 수 있다. 계다가, 통상의 기술자는 다른 구조들 및 프로세스들이 개시된 것들을 대체할 수 있으며, 결과적인 구현들은 적어도 실질적으로 동일한 방식(들)으로 적어도 실질적으로 동일한 기능(들)을 수행하여 개시된 구현들과 적어도 실질적으로 동일한 결과(들)를 달성할 것이라는 것을 이해할 것이다. 따라서, 이를 및 다른 구현들은 본원에 의해 고려된다. 특히, 본 발명에 따른 LUT를 인코딩하기 위한 방법은 각각의 정점과 관련된 값들의 타입들에 관계없이 임의 종류의 LUT를 인코딩하는 데 사용될 수 있다.

도면

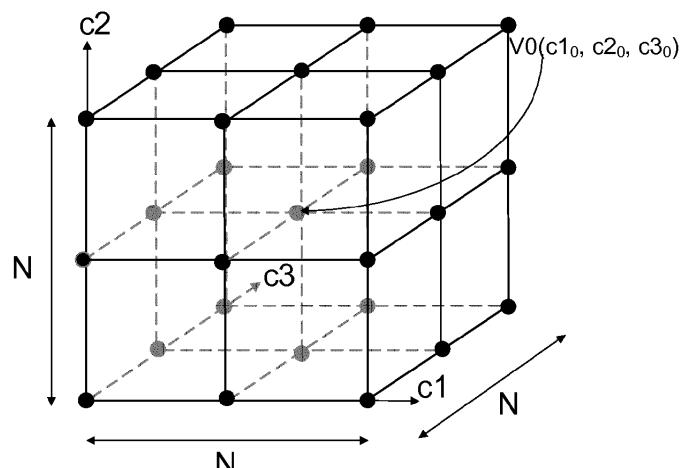
도면1



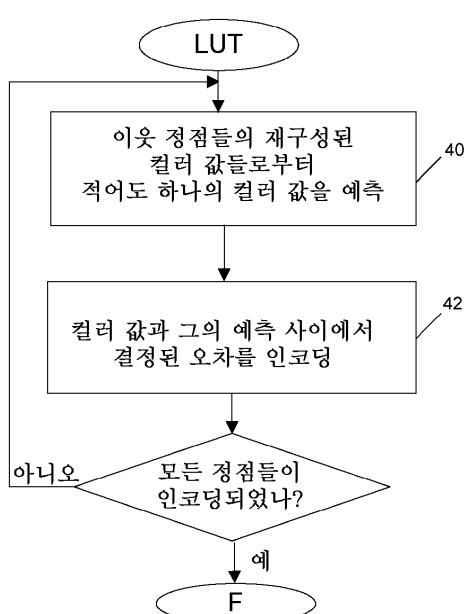
도면2



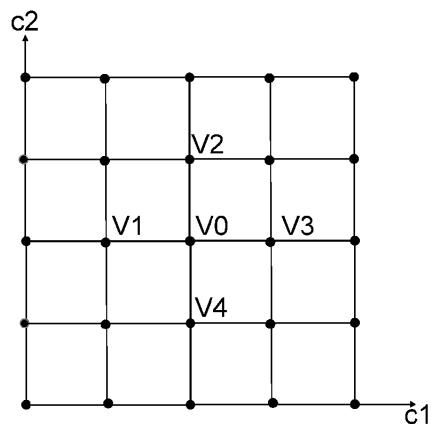
도면3



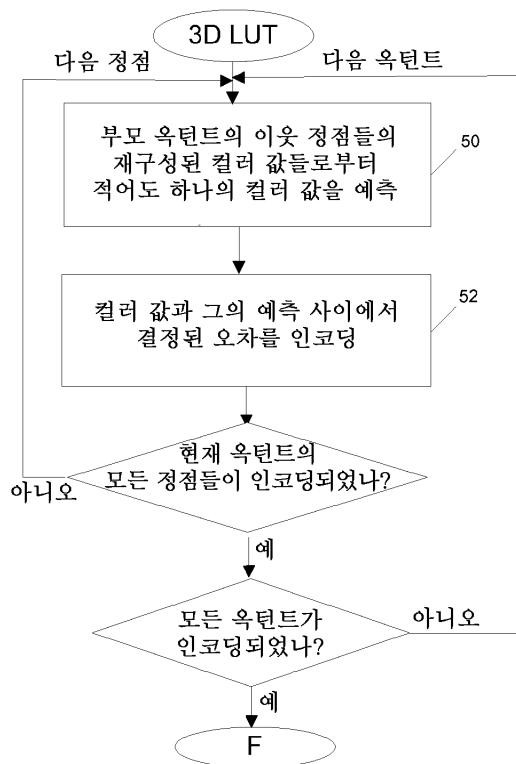
도면4



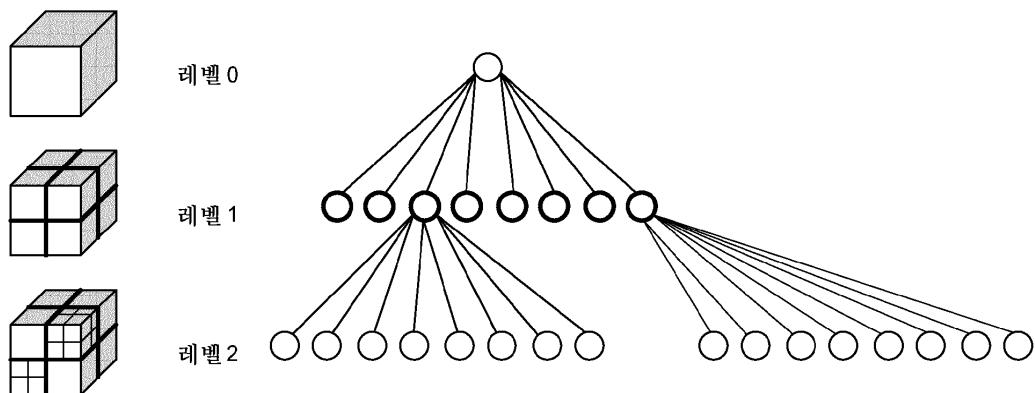
도면5



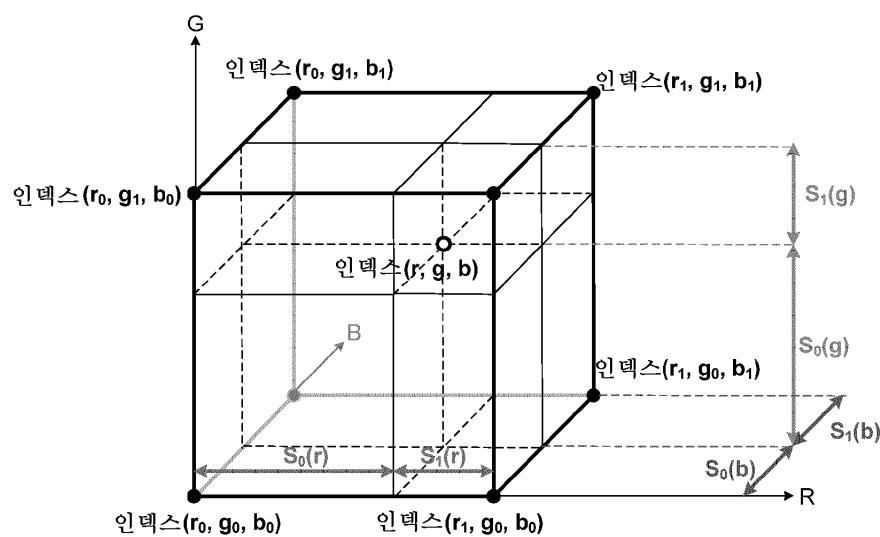
도면6



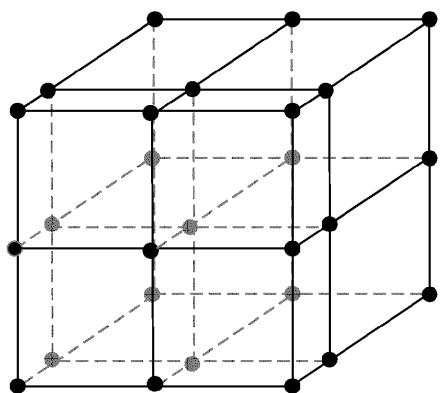
도면7



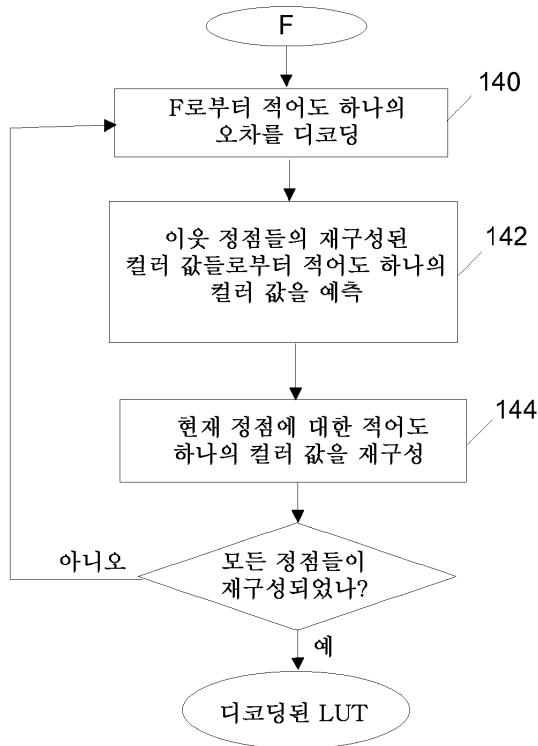
도면8



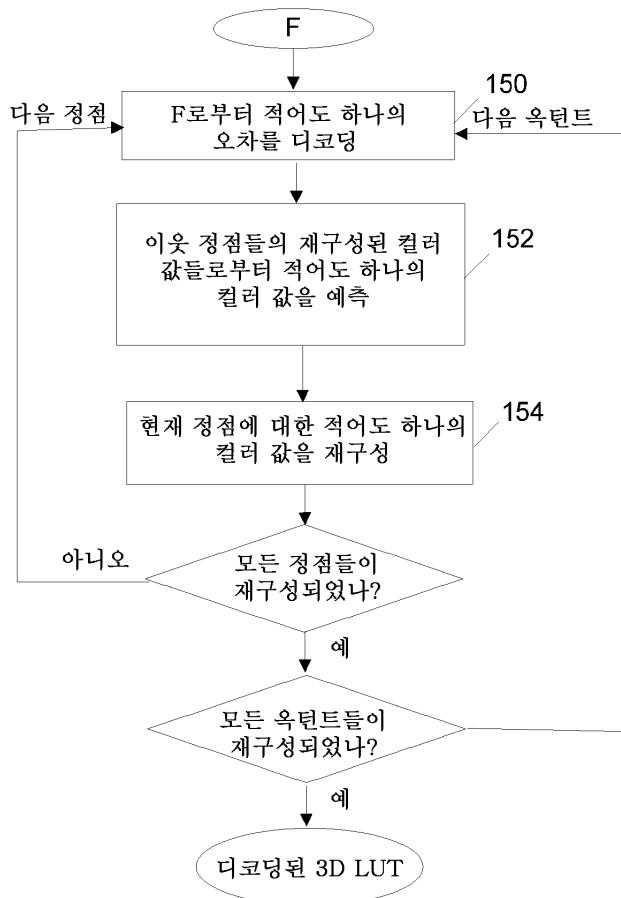
도면9



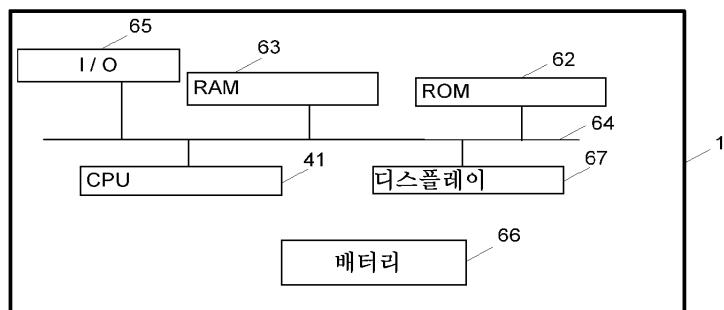
도면10



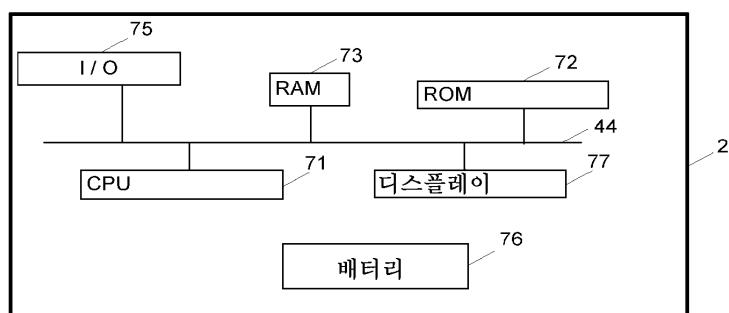
도면11



도면12



도면13



도면14

