



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년06월05일
(11) 등록번호 10-1986282
(24) 등록일자 2019년05월30일

(51) 국제특허분류(Int. Cl.)
G06T 9/00 (2019.01) H04N 19/00 (2014.01)
(21) 출원번호 10-2014-7029076
(22) 출원일자(국제) 2012년12월29일
심사청구일자 2017년12월08일
(85) 번역문제출일자 2014년10월16일
(65) 공개번호 10-2015-0008070
(43) 공개일자 2015년01월21일
(86) 국제출원번호 PCT/CN2012/087937
(87) 국제공개번호 WO 2013/155859
국제공개일자 2013년10월24일
(30) 우선권주장
PCT/CN2012/074388 2012년04월19일 중국(CN)
(56) 선행기술조사문헌
WO2010149492 A1
US20090080516 A1
US6438266 A
Exploiting repeated patterns for efficient
compression of massive models

(73) 특허권자
톱슨 라이센싱
프랑스 92130 이씨레블리노 루 잔다르크 1-5
(72) 발명자
카이 강잉
중국, 베이징 100190, 하이텐 디스트릭트, 치첸
로드, 루오추양썰리 레지던시, 6-2-901
지양 웬페이
중국, 베이징 100192, 하이텐 디스트릭트, 지산키
지안카이첵 웨스트 로드, 룡 지양 센트럴 에어리
어 14-3-301
티안 지양
중국, 베이징 100025, 카오양 디스트릭트, 칭니안
로드 29, 22-1002
(74) 대리인
김순웅, 안문환, 문경진

전체 청구항 수 : 총 12 항

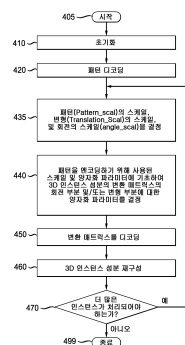
심사관 : 신재철

(54) 발명의 명칭 반복 구조 검색 기반의 3D 모델 압축을 위한 방법 및 장치

(57) 요약

3D 모델은 “패턴-인스턴스” 표현을 사용하여 모델링될 수 있으며, 여기서, 인스턴스 성분은 패턴의 변환(예를 들면, 회전, 변형, 및 스케일링)으로서 표현될 수 있다. 압축 효율을 향상시키기 위하여, 인스턴스의 변환을 위한 회전 부분 및 변형 부분의 양자화 파라미터는 해당하는 패턴을 엔코딩하기 위해 사용되는 양자화 파라미터에 기초하여 결정될 수 있다. 구체적으로, 회전 부분에 대한 양자화 파라미터는 인스턴스의 사이즈에 의존할 수 있고, 변형 부분에 대한 양자화 파라미터는 변형의 스케일에 의존할 수 있다. 즉, 큰 인스턴스는 회전 부분에 대한 미세한 양자화 파라미터를 사용할 수 있다. 따라서, 패턴, 변환의 변형 부분, 및 변환의 회전 부분을 압축함으로써 야기되는 양자화 에러들이 유사한 레벨에 있도록, 양자화 파라미터가 결정된다.

대표도



명세서

청구범위

청구항 1

3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법으로서,

3D 모델과 연관된 패턴을 엔코딩하기 위해 사용되는 제 1 양자화 파라미터를 액세스하는 단계(310, 410),

패턴의 인스턴스의 스케일과 패턴의 스케일을 결정하는 단계(350, 435)로서, 인스턴스는 패턴의 변환으로서 표현되는, 스케일을 결정하는 단계(350, 435),

제 1 양자화 파라미터, 인스턴스의 스케일, 및 패턴의 스케일에 따라 인스턴스에 대한 변환의 회전 부분의 제 2 양자화 파라미터를 결정하는 단계(360, 440), 및

제 2 양자화 파라미터에 따라 인스턴스에 대한 변환의 회전 부분의 엔코딩 또는 디코딩을 수행하는 단계(370, 450)를 포함하는, 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법.

청구항 2

제 1 항에 있어서,

제 2 양자화 파라미터는 인스턴스의 스케일과 패턴의 스케일 사이의 비율에 따라 결정되는, 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법.

청구항 3

제 1 항에 있어서,

인스턴스에 대한 회전 부분의 스케일을 결정하는 단계를 더 포함하고, 제 2 양자화 파라미터는 인스턴스에 대한 회전 부분의 스케일에 따라 더 결정되는, 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법.

청구항 4

제 1 항에 있어서,

패턴의 스케일은 3D 모델의 모든 패턴의 경계 박스의 대각선 거리에 따라 결정되는, 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법.

청구항 5

제 1 항에 있어서,

인스턴스의 스케일은 인스턴스의 경계 박스의 대각선 거리에 따라 결정되는, 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법.

청구항 6

제 1 항에 있어서,

인스턴스에 대한 변환의 변형 부분의 스케일을 결정하는 단계, 및

제 1 양자화 파라미터, 변형 부분의 스케일, 및 패턴의 스케일에 따라 변형 부분에 대한 제 3 양자화 파라미터를 결정하는 단계를 더 포함하는, 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법.

청구항 7

제 6 항에 있어서,

변형 부분에 대한 제 3 양자화 파라미터는 변형 부분의 스케일과 패턴의 스케일 사이의 비율에 따라 결정되는, 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법.

청구항 8

제 1 항에 있어서,

패턴, 변환의 회전 부분, 및 변환의 변형 부분을 엔코딩함으로써 야기되는 양자화 에러들은 동일한 것을 특징으로 하는, 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법.

청구항 9

제 1 항에 있어서,

패턴을 엔코딩하기 위해 사용되는 제 1 양자화 파라미터는 3D 모델의 스케일 및 패턴의 스케일 중 적어도 하나에 따라 결정되는, 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법.

청구항 10

제 1 항 내지 제 9 항 중 어느 한 항에 따른 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 수단을 포함하는 장치(500, 600).

청구항 11

제 1 항 내지 제 9 항 중 어느 한 항에 따른 비트스트림을 생성 또는 디코딩하기 위한 명령어가 저장된 컴퓨터 판독 가능한 저장 매체.

청구항 12

제 1 항 내지 제 9 항 중 어느 한 항에 따라 생성된 비트스트림이 저장된 컴퓨터 판독 가능한 저장 매체.

발명의 설명

기술 분야

[0001] 본 출원은 본 명세서에 참고로서 포함되어 있는 2012년 4월 19일자로 출원된 국제 특허 출원 번호 PCT/CN2012/074388의 이익을 주장한다.

[0002] 본 발명은 3D 모델을 표현하는 비트스트림을 생성하기 위한 방법 및 장치와, 그 비트스트림을 디코딩하기 위한 방법 및 장치에 관한 것이다.

배경 기술

[0003] 실제 응용에 있어서, 많은 3D 모델은 다수의 접속된 성분들로 구성된다. 이러한 다중 성분 3D 모델은, 도 1에 도시된 것처럼, 다양한 변환으로 많은 반복 구조를 일반적으로 포함한다.

[0004] 입력 모델에 있어 반복 구조의 장점을 이용하는 다중-성분 3D 모델에 대한 효율적인 압축 알고리즘은 공지되어 있다. 3D 모델의 반복 구조는 다양한 위치, 방향, 및 스케일링 요소에서 발견된다. 이때, 3D 모델은 "패턴-인스턴스" 표현으로 구성된다. 패턴은 해당하는 반복 구조의 대표적인 형상을 나타내기 위해 사용된다. 반복 구조에 속하는 성분은 해당하는 패턴의 인스턴스로서 표시되고, 패턴 ID와, 예를 들어, 패턴에 대해서 반영, 변형, 회전 및 가능한 스케일링과 같은 변환 정보에 의해 표현될 수 있다. 인스턴스 변환 정보는 예를 들어, 반영 부분, 변형 부분, 회전 부분, 및 가능한 스케일링 부분으로 구성될 수 있다. 고유 성분으로서 지칭되는 반복되지 않는 3D 모델의 일부 성분이 있을 수 있다.

[0005] 구체적으로 본 명세서에서 참고로서 포함되어 있고, K. Cai, W. Jiang, 및 T. Luo(PCT/CN2012/070877, 대리인 문서 번호 PA120001)에 의한 명칭이 "에러 제어 가능한 반복 구조 검색 기반 압축을 위한 시스템 및 방법"인 공동 소유의 PCT 출원은 인스턴스 성분들 간의 중복을 줄이고, 따라서, 압축 효율을 개선하기 위하여 3D 모델에서의 반복 구조를 식별하기 위한 방법 및 장치를 개시한다.

발명의 내용

해결하려는 과제

[0006] 본 발명은 3D 엔코더의 레이트-왜곡 성능을 고려하여 3D 인스턴스 성분에 대한 변환 정보를 효율적으로 양자화하기 위한 방법 및 장치를 제공하기 위한 것이다.

과제의 해결 수단

[0007] 본 발명은 3D 모델을 표현하는 비트스트림을 생성 또는 디코딩하기 위한 방법을 제공하며, 이 방법은, 3D 모델과 연관된 패턴을 엔코딩하기 위해 사용되는 제 1 양자화 파라미터를 액세스하는 단계, 패턴의 인스턴스의 스케일과 패턴의 스케일을 결정하는 단계로서, 인스턴스는 패턴의 변환으로서 표현되는, 스케일을 결정하는 단계, 제 1 양자화 파라미터, 인스턴스의 스케일, 및 패턴의 스케일에 따라 인스턴스에 대한 변환의 회전 부분의 제 2 양자화 파라미터를 결정하는 단계, 및 아래에 설명되는 제 2 양자화 파라미터에 따라 인스턴스에 대한 변환의 회전 부분의 엔코딩 또는 디코딩을 수행하는 단계를 포함한다. 본 발명은 또한 그 단계들을 수행하기 위한 장치를 제공한다.

[0008] 또한, 본 발명은 상술한 방법에 따른 비트스트림을 생성 또는 디코딩하기 위한 명령어가 저장된 컴퓨터 판독 가능한 저장 매체를 제공한다.

[0009] 또한, 본 발명은 상술한 방법에 따라 생성된 비트스트림이 저장된 컴퓨터 판독 가능한 저장 매체를 제공한다.

발명의 효과

[0010] 본 발명에 따라, 3D 엔코더의 레이트-왜곡 성능을 고려하여 3D 인스턴스 성분에 대한 변환 정보를 효율적으로 양자화하기 위한 방법 및 장치를 제공한다.

도면의 간단한 설명

[0011] 도 1은 많은 접속 성분들과 반복 구조를 갖는 예시적인 3D 모델을 도시한 도면,
 도 2a는 패턴을 묘사하는 그림을 이용한 예를 도시한 도면,
 도 2b는 해당하는 인스턴스 및 재구성된 인스턴스를 묘사하는 그림을 이용한 예를 도시한 도면,
 도 2c는 해당하는 인스턴스와, 회전 파라미터의 재구성 오류를 갖는 인스턴스를 묘사하는 그림을 이용한 예를 도시한 도면,
 도 3은 본 발명의 실시예에 따라, 3D 모델의 패턴 및 인스턴스를 엔코딩하기 위한 예를 도시하는 흐름도,
 도 4는 본 발명의 실시예에 따라, 3D 모델의 패턴 및 인스턴스를 디코딩하기 위한 예를 도시하는 흐름도,
 도 5는 본 발명의 실시예에 따른 3D 모델의 예시적인 엔코더를 도시한 도면,
 도 6은 본 발명의 실시예에 따른 3D 모델의 예시적인 디코더를 도시한 도면, 및
 도 7은 본 발명의 실시예에 따라, 비트스트림 구조를 묘사한 예를 도시한 도면.

발명을 실시하기 위한 구체적인 내용

[0012] 도 1에 도시된 것처럼, 3D 모델의 많은 반복 구조가 있을 수 있다. 3D 모델을 효율적으로 엔코딩하기 위하여, 반복 구조는 패턴 및 인스턴스로 구성될 수 있는데, 여기서, 인스턴스는, 예를 들어 변형, 회전, 및 스케일링과 같은 정보를 포함하는 패턴 ID 및 변환 매트릭스를 사용하여, 해당 패턴의 변환으로서 표현될 수 있다.

[0013] 인스턴스가 패턴 ID 및 변환 매트릭스로 표현될 때, 패턴 ID 및 변환 매트릭스는 인스턴스를 압축할 때 압축되어야 한다. 따라서, 인스턴스는 패턴 ID 및 디코딩된 변환 매트릭스를 통해 재구성될 수 있는데, 즉, 인스턴스는 패턴 ID에 의해 색인된 디코딩된 패턴의 변환(디코딩된 변환 매트릭스로부터)으로서 재구성될 수 있다.

[0014] 도 2a 및 도 2b는 2D 표현에서 예시적인 성분을 도시하며, 여기서, 성분(210 및 220)은 패턴이며, 성분(230 및 250)(실선)은 압축될 원래의 인스턴스이고, 성분(240 및 260)(점선)은 재구성된 인스턴스이다. 특히, 인스턴스(230 및 250)는 패턴(210 및 220)의 변환(즉, 회전 및 변형)된 버전으로서 각각 표현될 수 있다. 도 2b에 있어서, 원래의 인스턴스와 재구성된 인스턴스 사이에는, 변형 및 회전 양자화 모두에 의해 야기되는 에러가 존재한다. 예시의 목적으로, 도 2c에는 단지 회전 에러를 갖는 인스턴스(270 및 280)를 제공한다.

[0015] 도 2c의 예에 있어서, 회전의 양자화는, 원래 인스턴스와 재구성된 인스턴스 사이의 차이를 일으키는 약 5°의

에러를 도입한다. 도 2c에서 알 수 있는 것처럼, 회전 에러(각도 측정으로)는 인스턴스(230 및 250)에 대해서 유사하지만, 회전 양자화에 의해 야기된 정점 좌표 에러(즉, 정점 변화, 예를 들어, 도 2c에서 A에서 A"까지, 그리고, B에서 B"까지, 원래의 인스턴스와 재구성된 인스턴스 사이)는 두 인스턴스 사이에서 상당히 변화하고, 인스턴스(250)는 훨씬 더 큰 정점 좌표 에러를 갖는다. 결과적으로, 동일한 양자화가 다양한 사이즈를 갖는 인스턴스에 대해 사용된다면, 재구성된 성분의 품질은 일치하지 않을 수 있는데, 예를 들어, 큰 인스턴스는 작은 인스턴스보다 낮은 재구성 품질을 가질 수 있다.

[0016] 본 발명은 3D 엔코더의 레이트-왜곡 성능을 고려하여 3D 인스턴스 성분에 대한 변환 정보를 효율적으로 양자화 하기 위한 방법 및 장치를 제공한다. 한 예시적인 실시예에 있어서, 최대 허용 왜곡을 고려하여, 인스턴스의 회전 부분 및 변형 부분에 대한 양자화 파라미터가 결정되어, 레이트-왜곡 성능이 향상된다. 이하, 양자화 비트의 수가 예시적인 양자화 파라미터로서 사용된다. 본 발명은 양자화 스텝 사이즈와 같은 다른 양자화 파라미터가 사용될 때 적용될 수 있다.

[0017] 변환의 스케일링 부분이 무손실(lossless) 부동 소수점 코덱에 의해 압축될 수 있기 때문에, 변환의 스케일링 부분은 다음 설명에 포함되지 않는다.

[0018] 발견된 "패턴-인스턴스" 표현에 기초하여, 원래의 인스턴스의 정점(v)은 다음 수학식 1로 표현될 수 있고,

수학식 1

$$v = R * p + T, \quad (1)$$

[0019]

[0020] 여기서, p는 해당 패턴의 v의 해당 정점이고, R 및 T는 각각 패턴과 인스턴스 사이의 회전 매트릭스 및 변형 벡터이다.

[0021] v, v_d의 디코딩된 위치는 다음 수학식 2로서 계산될 수 있고,

수학식 2

$$v_d = R_d * p_d + T_d, \quad (2)$$

[0022]

[0023] 여기서, p_d는 p의 디코딩된 위치이고, R_d 및 T_d는 각각 디코딩된 회전 매트릭스 및 디코딩 변형 벡터이다.

[0024] 이때, 정점(v)에 대한 압축에 의한 왜곡은 다음과 같이 계산될 수 있다.

수학식 3

$$\begin{aligned} v - v_d &= (R * p + T) - (R_d * p_d + T_d) \\ &= (R * p - R_d * p_d) + (T - T_d) = (R * p - R_d * p_d) + \Delta T \end{aligned} \quad (3)$$

[0025]

[0026] 수학적 이론에 기초하여, 다음이 유추될 수 있고,

수학식 4

$$\begin{aligned}
 \|v - v_d\| &= \|(R * p - R_d * p_d) + \Delta T\| \leq \|R * p - R_d * p_d\| + \|\Delta T\| \\
 &= \|R * p - R_d * p + R_d * p - R_d * p_d\| + \|\Delta T\| \\
 &\leq \|R * p - R_d * p\| + \|R_d * p - R_d * p_d\| + \|\Delta T\| \\
 &\leq \|(R - R_d) * p\| + \|p - p_d\| + \|\Delta T\| \\
 &= \|(R - R_d) * p\| + \|\Delta p\| + \|\Delta T\| \\
 &\leq \|(R - R_d) * p\| + \|\Delta p\|_{\max} + \|\Delta T\|_{\max}
 \end{aligned} \tag{4}$$

여기서, $\|\Delta p\|_{\max}$ 는 정점(p)의 양자화 에러의 상한이고, $\|\Delta T\|_{\max}$ 는 인스턴스 변형의 양자화 에러의 상한이다.

회전 매트릭스가 양자화 비트의 동일한 수를 사용하여 양자화될 수 있는 Euler 각도(α , β , γ)에 의해 표현된다고 가정하면,

수학식 5

$$\|(R - R_d) * p\| \leq \sqrt{3} \Delta \theta_{\max} \|p\|, \tag{5}$$

이며, 여기서, $\Delta \theta_{\max}$ 는 α , β 및 γ 에 대한 양자화 에러의 상한이다. 수학식 5는 다른 표현 또는 상이한 양자화 파라미터가 각도(α , β , γ)에 사용될 때 적용될 수 있다. 수학식 5를 사용하여, 수학식 4는 다음과 같이 되며,

수학식 6

$$\|v - v_d\| \leq \|\Delta p\|_{\max} + \sqrt{3} \Delta \theta_{\max} \|p\|_{\max} + \|\Delta T\|_{\max}, \tag{6}$$

여기서, $\|p\|_{\max}$ 는 해당 패턴의 임의의 정점과 해당 패턴의 중심 사이의 최대 거리이다.

QB_Pattern으로서 패턴을 엔코딩하기 위해 사용된 양자화 비트의 수를 나타내고, QB_Translation으로서 변형을 엔코딩하기 위한 양자화 비트의 수를 나타낸다면, $\|\Delta p\|_{\max}$ 는 다음과 같이 계산될 수 있다.

$$\|\Delta p\|_{\max} = \frac{Pattern_Scal}{2^{QB_Pattern+1}},$$

[0036] 및, $\|\Delta T\|_{\max}$ 는 다음과 같이 계산되며,

$$\|\Delta T\|_{\max} = \frac{Translation_Scal}{2^{QB_translation+1}},$$

[0038] 여기서, Pattern_Scal 및 Translation_Scal는 각각 패턴 정점 및 변형의 경계 박스의 스케일이다. 한 예에서, Translation_Scal은 모든 인스턴스의 모든 변형의 최대 및 최소 가능 값들 사이의 차이로서 측정될 수 있고, Pattern_Scal은 모든 패턴을 포함하는 경계 박스의 대각선 거리로서 측정될 수 있다. 다른 예에 있어서, Pattern_Scal은 x, y, 및 z 치수의 최대값으로서 측정될 수 있다.

[0039] 본 발명의 목표 중 하나는 최대 코딩 에러를 제어하면서 비트를 할당하기 위한 것이다. 최대 허용 코딩 에러가 MaxErr이고, 수학적 6은 다음의 부등식을 만족한다고 가정한다.

수학적 7

$$\|\Delta p\|_{\max} + \sqrt{3}\Delta\theta_{\max} \|p\|_{\max} + \|\Delta T\|_{\max} \leq MaxErr. \quad (7)$$

[0041] 한 실시예에 있어서, 단순화를 위해, 모든 패턴은 양자화 비트의 동일 수를 사용하고, 모든 인스턴스 변형은 양자화 비트의 동일 수를 사용한다고 가정한다. 따라서, 패턴에 대한 양자화 에러는 다음을 만족하고,

수학적 8

$$\|\Delta p\|_{\max} = MaxErr/3, \quad (8)$$

[0043] 인스턴스 변형에 대한 양자화 에러는 다음을 만족할 수 있다.

수학적 9

$$\|\Delta T\|_{\max} = MaxErr/3. \quad (9)$$

[0045] 양자화 에러에 대한 상한이 수학적 8 및 수학적 9에서 추정된 이후에, 패턴 정점 및 인스턴스 변형의 경계 박스의 스케일에 기초하여 양자화 비트의 수를 추정할 수 있다.

[0046] 한 단순화된 예시적인 실시예에 있어서, 두 상한이 동일하고, 따라서, 수학적 8 및 수학적 9에 따라, 인스턴스 변형이 패턴과 동일한 양자화 비트의 수를 사용할 수 있다고 가정한다.

[0047] 다른 예시적인 실시예에 있어서, 패턴의 경계 박스와 인스턴스의 경계 박스 사이의 스케일 차이가 고려된다.

$\|\Delta p\|_{\max} = \|\Delta T\|_{\max}$ 임을 가정하면, 변형 부분에 대한 양자화 비트 수가 다음과 같이 계산될 수 있다.

수학식 10

$$\begin{aligned}
 \|\Delta p\|_{\max} &= \frac{Pattern_Scal}{2^{QB_Pattern+1}} = \frac{Translation_Scal}{2^{QB_translation+1}} = \|\Delta T\|_{\max} \\
 \Rightarrow \frac{2^{QB_translation}}{2^{QB_Pattern}} &= \frac{Translation_Scal}{Pattern_Scal} \\
 \Rightarrow 2^{QB_translation-QB_Pattern} &= \frac{Translation_Scal}{Pattern_Scal} \\
 \Rightarrow QB_translation - QB_Pattern &= \log_2\left(\frac{Translation_Scal}{Pattern_Scal}\right) \\
 \Rightarrow QB_translation &= QB_Pattern + \left\lceil \log_2\left(\frac{Translation_Scal}{Pattern_Scal}\right) \right\rceil
 \end{aligned} \tag{10}$$

[0048]

[0049]

파라미터 Translation_Scal는 비트스트림에 전송될 수 있고, 엔코더 및 디코더 모두에서 사용할 수 있게 되어, 수학식 10은 엔코더 및 디코더 모두에서 사용될 수 있다. 수학식 10에 따라, 모든 인스턴스 변형은 패턴의 스케일과 인스턴스 변형 사이의 비율에 관련된 동일한 양자화 파라미터를 사용한다.

[0050]

도 2에서 설명한 것처럼, 동일한 회전 양자화 에러(각도 측정에서)는 큰 성분에서 큰 정점 변화의 원인이 될 수 있다. 따라서, 회전 정보에 대한 양자화 파라미터는 인스턴스의 사이즈에 따라 적응적으로 양자화될 수 있다. 예를 들어, 비트의 많은 수 또는 작은 양자화 스텝 사이즈와 같은 미세한 양자화 파라미터는 큰 인스턴스의 회전 부분을 양자화하기 위해 사용될 수 있다.

[0051]

회전 양자화 에러에 의해 야기되는 정점 좌표 에러는 다음 수학식 11을 만족할 수 있다.

수학식 11

$$\sqrt{3}\Delta\theta_{\max_i}\|p\|_{\max} = MaxErr/3 \tag{11}$$

[0052]

[0053]

즉, i번째 인스턴스 성분의 회전 양자화 에러의 상한은 다음 수학식 12로서 추정될 수 있고,

수학식 12

$$\Delta\theta_{\max_i} = \frac{MaxErr}{(3\sqrt{3}\|p\|_{\max})} = \frac{MaxErr}{(3\sqrt{3} * HalfScal_i)} = \frac{\|p\|_{\max}}{(\sqrt{3} * HalfScal_i)}, \tag{12}$$

[0054]

[0055]

여기서, HalfScal_i는 i번째 인스턴스 성분의 절반 스케일(즉, 인스턴스의 경계 박스의 대각선 거리의 절반)이다. 한 실시예에 있어서, HalfScal_i는 해당 재구성된 패턴으로부터 추정될 수 있다. 엔코더 및 디코더 모두에 의해 액세스될 수 있는 재구성된 패턴을 사용하여 HalfScal_i를 추정함으로써, 회전 부분에 대한 양자화 파라미터는 디코더에서 계산될 수 있고, 따라서, 엔코더는 비트스트림에 회전 부분에 대한 양자화 파라미터를

명시적으로 표시할 필요가 없다. $\Delta\theta_{\max_i}$ 는 수학식 12로부터 상이하게 추정될 수 있다. 예를 들어, 엄격한 근사치가 수학식 5에서 추정될 때, 수학식 7 및 수학식 11, 뿐만 아니라 수학식 12는 그에 따라 갱신될 수 있다.

[0056]

압축 이전에, 모든 패턴은 세계 좌표 시스템과 함께 정렬된다. 이때, 모든 패턴은 동일한 양자화 파라미터를 사용하여 함께 엔코딩될 수 있다. 패턴의 위치 및 방향은 엔코딩 비트스트림에 또한 엔코딩 및 기록된다. 패턴

(QB_Pattern)을 엔코딩하기 위한 양자화 비트의 수를 사용하여, i번째 인스턴스(QB_Rot_i)의 회전 부분에 대한 양자화 비트의 수는 다음과 같은 회전 각도 및 패턴 정점의 경계 박스의 스케일에 기초하여 계산될 수 있고,

수학식 13

$$\begin{aligned}\Delta\theta_{\max_i} &= \frac{\text{angle_Scal}}{2^{QB_Rot_i+1}} = \frac{\text{Pattern_Scal}}{(\sqrt{3} * \text{HalfScal}_i) * 2^{QB_Pattern+1}} = \frac{\|\Delta p\|_{\max}}{(\sqrt{3} * \text{HalfScal}_i)} \\ \Rightarrow 2^{QB_Rot_i - QB_Pattern} &= \frac{\sqrt{3} * \text{HalfScal}_i * \text{angle_Scal}}{\text{Pattern_Scal}} \\ \Rightarrow QB_Rot_i &= QB_Pattern + \left\lceil \log_2 \left(\frac{\sqrt{3} * \text{HalfScal}_i * \text{angle_Scal}}{\text{Pattern_Scal}} \right) \right\rceil\end{aligned}\quad (13)$$

[0057]

[0058] 여기서, angle_Scal은 회전 각도의 스케일이다. 한 예에 있어서, angle_Scal은 모든 인스턴스 회전의 최대 및 최소 가능 각도들 사이의 차이로서 측정될 수 있다. 다른 예에 있어서는 angle_Scal를 2π 로 설정한다.

[0059] Pattern_scal은 모든 패턴의 경계 박스의 사이즈로서 측정되고, 각각의 Euler 각도의 가능한 값은 $0 \sim 2\pi$ 이고, 수학식 13은 다음과 같이 된다고 가정한다.

수학식 14

$$QB_Rot_i = QB_Pattern + \left\lceil \log_2 \left(\frac{2\sqrt{3}\pi * \text{HalfScal}_i}{\text{Pattern_Scal}} \right) \right\rceil\quad (14)$$

[0060]

[0061] 즉, 인스턴스가 클 때(즉, HalfScal_i가 클 때), 인스턴스에 대한 양자화 비트의 수(QB_Rot_i)는 큰 것으로 결정된다. 따라서, 본 발명에 따라, 더 큰 인스턴스는 그 회전 부분에 대한 더 미세한 양자화를 얻는다.

[0062] 여기서, 수학식 14에서, 변환에는 스케일링이 부분이 없는 것으로 가정한다. 수학식 14는 회전 각도의 스케일이 변할 때 그에 따라 조정되어야 하고, 변환의 스케일링 부분이 고려되거나, 패턴 및 인스턴스의 사이즈가 상이하게 측정된다.

[0063] 상술한 바와 같이, 인스턴스의 변환 매트릭스의 변형 부분 및 회전 부분에 대한 양자화 파라미터는 압축 효율을 개선하기 위하여, 패턴을 엔코딩하기 위해 사용되는 양자화 파라미터에 기초하여 결정될 수 있다. 한 예시적인 실시예에 있어서, 패턴(QB_Pattern)을 엔코딩하기 위해 사용된 양자화 비트의 수는 비트스트림에 표시될 수 있고, 인스턴스의 변환 매트릭스의 변형 부분은 아마도 수학식 10을 사용하여 계산될 수 있고, 회전 부분에 대한 양자화 비트의 수는 수학식 14를 사용하여 계산될 수 있다.

[0064] 다른 예시적인 실시예에 있어서, 패턴 정점의 재구성 에러의 상한, 인스턴스 변형 및 인스턴스 회전은 동일하다고 가정하는데, 즉,

$$\|\Delta P\|_{\max} = \frac{\|\Delta E\|_{\max}}{3},$$

[0065]

[0066] 여기서, $\|\Delta E\|_{\max}$ 는 전체 모델의 양자화 에러의 상한이고, $\|\Delta P\|_{\max}$ 는 모든 패턴의 양자화 에러의 상한이다. 결과적으로, 패턴을 엔코딩하기 위한 양자화 비트의 수는 다음과 같이 전체 3D 입력 모델의 양자화 비트의 수로부터 계산될 수 있고,

수학식 15

$$QB_Pattern = QB_EntireModel + \left\lceil \log_2 \left(\frac{3 * Pattern_Scal}{EntireModel_Scal} \right) \right\rceil \quad (15)$$

[0068] 여기서, QB_EntireModel은 전체 3D 입력 모델에 대한 양자화 비트의 수이고, EntireModel_Scal은 전체 3D 입력 모델의 사이즈를 반영한다.

[0069] 상기에서, Pattern_Scal은 3D 모델에서 모든 패턴을 사용하여 측정될 수 있다. 다른 실시예에 있어서, 패턴은 별도로 압축되고, 동일한 패턴의 인스턴스는 함께 압축된다. 모든 패턴은 동일한 양자화 파라미터를 사용할 수 있다. 또한, 수학식 10 및 수학식 13에서 Pattern_Scal은 해당 패턴의 경계 박스의 스케일에 대응하고, 수학식 10에서 Translation_Scal 및 수학식 13에서 angle_Scal은 각각 변형의 스케일 및 현재 패턴의 인스턴스의 회전 각도에 대응한다.

[0070] 도 3은 3D 모델의 인스턴스 및 패턴을 엔코딩하기 위한 예시적인 방법(300)을 도시한다. 방법(300)은 단계(305)에서 시작한다. 단계(310)에서, 3D 모델 데이터가 입력되어 초기화가 실행된다. 품질 파라미터와 같은 추가적인 데이터, 패턴을 엔코딩하기 위한 양자화 파라미터는 또한 입력되거나 또는 입력으로부터 유추될 수 있다. 한 예시적인 실시예에 있어서, 초기화 단계는 반복 구조를 패턴 및 인스턴스로 구성할 수 있고, 인스턴스에 대한 변환 매트릭스를 생성할 수 있다. 단계(320)에서, 패턴은 재구성된 패턴을 형성하기 위해 엔코딩된다. 단계(350)에서, 패턴, 변형, 회전의 스케일이 결정된다. 예를 들어, 패턴(Pattern_scal)의 스케일은 모든 패턴을 포함하는 경계 박스의 대각선 거리로서 측정될 수 있고, 변형(Translation_Scal)의 스케일은 메타데이터로부터 획득될 수 있고, 비트스트림에 전송되며, 회전 스케일(angle_scal)은 각도 표현에 기초하여 결정될 수 있으며, 예를 들어, angle_scal은 Euler 표현이 사용될 때 2π 로 설정된다.

[0071] 단계(360)에서, 인스턴스 성분에 대한 변환 매트릭스의 변형 부분에 대한 양자화 파라미터는 예를 들어, 수학식 10을 사용하여 결정될 수 있다. 변환 매트릭스의 회전 부분에 대한 양자화 파라미터는 예를 들어 수학식 14를 사용하여 결정될 수 있다. 다음에, 인스턴스에 대한 변환 매트릭스는 단계(370)에서 엔코딩된다. 단계(380)에서, 더 많은 인스턴스가 처리되어야 할 필요가 있는지의 여부를 확인한다. 더 많은 인스턴스가 처리되어야 한다면, 제어는 단계(350)로 복귀한다. 그렇지 않으면, 제어는 종료 단계(399)로 전달된다.

[0072] 도 4는 3D 모델의 인스턴스 및 패턴을 디코딩하기 위한 예시적인 방법(400)을 도시한다. 방법(400)의 입력은 비트스트림, 예를 들어, 방법(300)을 사용하여 생성된 비트스트림을 포함할 수 있다. 방법(400)은 단계(405)에서 시작한다. 단계(410)에서, 초기화가 실행되고, 예를 들어, 패턴을 엔코딩하기 위해 사용되는 양자화 파라미터는 입력 비트스트림으로부터 얻어진다.

[0073] 패턴은 단계(420)에서 디코딩된다. 단계(435)에서, 패턴, 변형, 및 회전의 스케일이 결정된다. 단계(440)에서, 인스턴스 성분에 대한 변환 매트릭스의 변형 부분 및/또는 회전 부분에 대한 양자화 파라미터가 결정될 수 있다. 변환 매트릭스는 단계(450)에서 디코딩된다. 다음에, 인스턴스는 단계(460)에서, 예를 들어, 해당 재구성된 패턴 및 디코딩된 변환 매트릭스를 사용하여 재구성된다. 단계(470)에서, 더 많은 인스턴스가 처리되어야 할 필요가 있는지를 확인한다. 긍정인 경우, 제어는 단계(435)로 복귀되고, 그렇지 않은 경우, 제어는 종료 단계(499)로 전달된다.

[0074] 인스턴스를 적절히 디코딩하기 위하여, 패턴, 변형, 및 회전의 스케일을 결정하고, 양자화 파라미터를 계산하는 해당 방법은 엔코더 및 디코더에서 사용되어야 한다는 것을 주목한다. 예를 들어, 방법(300)에 의해 생성된 비트스트림은 방법(400)에 입력으로서 사용되고, 단계(435 및 440)는 각각 단계(350 및 360)에 대응할 것이다.

[0075] 도 3 및 도 4에 있어서, 3D 모델에서 모든 패턴은 인스턴스 이전에 엔코딩 또는 디코딩된다. 다른 실시예에 있

어서, 각각의 패턴의 엔코딩/디코딩 이후에, 해당 인스턴스의 엔코딩/디코딩이 이어진다.

상기에서, 양자화 비트의 수를 할당하는 방법이 최대 허용 에러가 주어진 상태에서 설명되었다. 다른 한편, 수학식 7은 레이트 제어를 위해, 즉, 디코딩된 3D 모델의 품질을 최적화하기 위하여 비트의 수가 주어진 상태에서 양자화 파라미터를 결정하기 위해 또한 사용될 수 있다.

압축된 패턴의 비트의 수에서 측정된 사이즈가 다음과 같이 계산될 수 있고,

수학식 16

$$Compr_Pattern = QB_Pattern * Pattern_Ver_Num * Cpv + Pattern_Tri_Num * Cpt, (16)$$

여기서, Pattern_Ver_Num 및 Pattern_Tri_Num은 각각 정점의 수 및 모든 패턴의 삼각형의 수이고, Cpv 및 Cpt는, 예를 들어, 엔트로피 엔코딩의 압축 비율을 카운팅하기 위해 사용된 미리 결정된 값이다. 한 구현에 있어서, Cpv= 0.2이고, Cpt= 3이다. 압축된 인스턴스의 사이즈는 다음과 같이 계산될 수 있고,

수학식 17

$$Compr_Insta = (QB_translation * 3 + \sum_{i=1}^{InstanceNum} QB_Rot_i * 3) * C_{insta}, (17)$$

여기서, C_{insta}는 미리 결정된 값이다. 압축된 고유 성분에 대한 사이즈는 다음과 같이 계산될 수 있고,

수학식 18

$$Compr_UniComp = QB_UniComp * UniComp_Ver_Num * C_{pv} + UniComp_Tri_Num * C_{pt}, (18)$$

여기서, QB_UniComp는 고유 성분을 엔코딩하기 위한 양자화 비트의 수이고, UniComp_Ver_Num 및 UniComp_Tri_Num은 각각 정점의 수 및 고유 성분이 삼각형의 수이다. 한 예시적인 실시예에 있어서,

$$\|\Delta U\|_{\max} = \|\Delta P\|_{\max}$$

이고, QB_UniComp는 다음과 같이 계산될 수 있고,

수학식 19

$$QB_UniComp = QB_Pattern + \left\lceil \log_2 \left(\frac{UniComp_Scal}{Pattern_Scal} \right) \right\rceil, (19)$$

여기서, UniComp_Scal은 고유 부분의 경계 박스의 스케일이다. 고유 성분의 변형에 대한 양자화 비트는 다음과 같이 계산될 수 있고,

$$QB_UniComp_Transl = QB_Pattern + \left\lceil \log_2 \left(\frac{UniComp_Transl_Scal}{Pattern_Scal} \right) \right\rceil,$$

여기서, UniComp_Transl_Scal은 모든 고유 성분의 모든 변형의 경계 박스의 스케일이다.

다음에, 압축된 3D 모델의 전체 사이즈는 다음과 같이 계산될 수 있다.

수학식 20

$$Compr_EntireModel = Compr_Pattern + Compr_Insta + Compr_UniComp.(20)$$

[0089]

[0090]

수학식 10, 수학식 14 및 수학식 19에 기초하여, QB_Translation, QB_Rot_i 또는 QB_UniComp는 QB_Pattern의 함수로서 획득될 수 있다. 다음에, 수학식 10, 수학식 14, 및 수학식 16 내지 수학식 19에 따라, Compr_Pattern, Compr_insta, 또는 Compr_UniComp는 QB_Pattern의 함수로서 표현될 수 있다. 또한, 수학식 20을 사용하여, 압축된 3D 모델 Compr_EntireModel 및 QB_Pattern의 전체 사이즈 사이의 관계가 얻어질 수 있다. 이어서, 예측되는 압축된 모델 사이즈 Compr_Model가 주어지면, QB_Pattern이 결정될 수 있다. QB_Pattern으로부터, 코딩 에

$$MaxErr = 3 * Pattern_Scal / 2^{QB_Pattern+1}$$

러 MaxErr의 상한이

을 사용하여 계산될 수 있다.

[0091]

도 5는 예시적인 3D 모델 엔코더(500)의 블록도를 도시한다. 장치(500)의 입력은 3D 모델, 3D 모델을 엔코딩하기 위한 품질 파라미터, 및 다른 메타데이터를 포함할 수 있다. 먼저, 3D 모델은 패턴, 인스턴스 및 고유 성분에 대해서 3D 모델을 출력하는 반복 구조 검색 모듈(510)을 통과한다. 패턴 엔코더(520)는 패턴을 압축하기 위해 채용되고, 고유 성분 엔코더(550)는 고유 성분을 엔코딩하기 위해 채용된다. 인스턴스에 대해서, 인스턴스 성분 정보는 사용자가 선택한 모드에 기초하여 엔코딩된다. 인스턴스 정보 그룹 모드가 선택되면, 인스턴스 정보는 그룹화 인스턴스 정보 엔코더(540)를 이용하여 엔코딩되고, 그렇지 않으면, 기본 인스턴스 정보 엔코더(530)를 이용하여 엔코딩된다. 엔코딩된 성분은 반복 구조 검증기(560)에서 더 검증된다. 엔코딩된 성분이 품질 요구 조건을 충족시키지 않으면, 고유 성분 엔코더(550)를 사용하여 엔코딩될 것이다. 패턴에 대한 비트스트림, 인스턴스, 및 고유 성분은 비트스트림 어셈블러(570)에서 조립된다.

[0092]

압축 효율을 개선하기 위하여, 원래의 3D 모델의 모든 성분은 성분의 중심이 정렬되도록 변형될 수 있다. 중심의 변형은 압축되고 비트스트림에 표시될 필요가 있다. 레이트-왜곡 성능은 성분의 경계 박스가 훨씬 작게 될 수 있고(대부분의 경우에), 적은 수의 양자화 비트가 동일한 코딩 에러에 필요하기 때문에, 개선될 수 있다.

[0093]

또한, 패턴은 성분의 중심을 원점으로 변형함으로써 생성될 수 있고, 원래의 성분은 인스턴스로서 간주될 수 있다. 압축 효율을 개선하기 위하여, 패턴 성분(패턴에 해당하는 성분)에 대해서, 단지 변형은 압축된 비트스트림에서 표시된다. 패턴-인스턴스 표현으로부터 얻어진 비트레이트를 보장하기 위해 반복 구조가 없거나 충분하지 않은 경우, 입력 3D 모델의 모든 성분은 고유 성분으로서 간주될 수 있다.

[0094]

도 6은 예시적인 3D 모델 디코더(600)의 블록도를 도시한다. 장치(600)의 입력은 3D 모델의 비트스트림, 예를 들어, 엔코더(500)에 의해 생성된 비트스트림을 포함할 수 있다. 압축된 비트스트림에서 패턴에 관련된 정보는 패턴 디코더(620)에 의해 디코딩된다. 고유 성분에 관련된 정보는 고유 성분 디코더(650)에 의해 디코딩된다. 또한, 인스턴스 정보의 디코딩은 사용자가 선택한 모드에 의존한다. 인스턴스 정보 그룹 모드가 선택되면, 인스턴스 정보는 그룹화 인스턴스 정보 디코더(640)를 사용하여 디코딩되고, 그렇지 않으면, 기본 인스턴스 정보 디코더(630)를 사용하여 디코딩된다. 디코딩된 패턴, 인스턴스 정보 및 고유 성분은 재구성되어 모델 재구성 모듈(660)에서 출력 3D 모델을 생성한다.

[0095]

도 7은 압축된 비트스트림을 생성하기 위한 예시적인 방법(700)을 도시한다. 비트스트림은 압축된 스트림을 디코딩하기 위해 필요한 모든 정보를 포함하는 헤더 버퍼(PB3DMC_stream_header, 710)로 시작한다. 정보는,

[0096]

● 원래의 모델에서 고유 부분이 있는지의 여부,

[0097]

● 원래의 모델에서 적어도 하나의 반복 구조가 있는지의 여부,

[0098]

● "그룹화 인스턴스 변환 모드" 또는 "기본 인스턴스 변환 모드"가 이 비트스트림에 사용되는지의 여부,

[0099]

● 원래의 3D 모델에 관한 정보,

[0100]

● 인스턴스가 가질 수 있는 속성의 유형에 관한 정보,

[0101]

● 형상을 압축하기 위해 사용되는 3D 모델 압축 방법, 및

[0102] ● 모든 3D 객체(필요하다면, 패턴 및 다른 부분)의 연결 및 특성을 포함할 수 있다.

[0103] 원래의 모델에서 고유 부분 및 반복 구조가 없는 경우(`uni_part_bit == 0 && repeat_struct_bit == 0`), 비트스트림의 나머지 부분은 `PB3DMC_stream_header`에 표시되는 3D 모델 압축 방법을 사용하여 압축된 입력 3D 모델(730)이다. 그렇지 않은 경우에, 비트스트림에서 다음 부분은 일부가 있는 경우(740)에 모든 고유 성분의 압축된 결과이다(745). 적어도 하나의 반복 구조가 있는 경우(750), 다음 데이터 필드는 모든 패턴의 압축된 결과이다(760). 어느 인스턴스 변환 패킹 모드가 비트스트림에 선택되는가(770)에 따라, `compr_insta_grouped_data(780)` 또는 `compr_insta_elementary_data(785)`가 비트스트림에서 다음 부분이다.

[0104] 표 1에서, 헤더에 대한 예시적인 구문과 의미가 본 발명의 원리에 대해 설명되어 있다. 일부 데이터 필드가 3DMC 확장[w11455, Final text of ISO/IEC 14496-16 4th Edition, MPEG-3DGC, 93th MPEG meeting, 2011-02]의 비트스트림 정의로부터 선택되었음을 주목한다.

[0105] 표 1

class 3d_model_header{	Num. of Bits	Descriptor
ver_num	32	
tri_num	32	
default_coord_bbox	1	
if(default_coord_bbox=='0'){		
coord_bbox	6*32	
}		
QP_coord	5	
normal_binding		
if(normal_binding != 'not_found'){		
default_normal_bbox	1	
QP_normal	5	
}		
color_binding		
if(color_binding != 'not_found'){		
default_color_bbox	1	
if(defalut_color_bbox=='0'){		
color_bbox	6*32	
}		
QB_color	5	
}		
multi_texCoord_num	5	
if (multi_texCoord_num != 0){		
for (i=0; i < multi_texCoord_num; i++){		
texCoord_binding		
default_texCoord_bbox	1	

[0106]

if(default_texCoord_bbox=='0'){		
texCoord_bbox	4*32	
}		
QB_texCoord	5	
}		
}		
multi_attribute_num	5	
if (multi_attribut_num != 0){		
for (i=0; i < multi_attribute_num; i++){		
attribute_dim_num		
attribute_binding		
default_attribute_bbox	1	
if(default_attribute_bbox=='0'){		
attribute_bbox	2*attribute_dim_num*32	
}		
QB_attribute	5	
}		
}		
}		

[0107]

[0108]

ver_num: 이 32 비트 부호가 없는 정수는 전체 3D 모델의 정점의 수를 포함한다. 이 값은 디코딩된 3D 모델을 검증하는 데 사용될 수 있다.

[0109]

tri_num: 이 32 비트 부호가 없는 정수는 전체 3D 모델의 삼각형의 수를 포함한다. 이 값은 디코딩된 3D 모델을 검증하는 데 사용될 수 있다.

[0110]

default_coord_bbox: 이 1-비트 부호가 없는 정수는 디폴트 경계 박스가 전체 3D 모델의 형상에 사용되었는지 여부를 나타낸다. 0은 다른 경계 박스를 사용하는 것을 의미하고, 1은 디폴트 경계 박스를 사용하는 것을 의미한다. 디폴트 경계 박스는 $x_{min}=0.0$, $y_{min}=0.0$, $z_{min}=0.0$, $x_{max}=1.0$, $y_{max}=1.0$, 및 $z_{max}=1.0$ 으로서 정의된다.

[0111]

coord_bbox: 이 데이터 필드는 전체 3D 모델의 형상의 경계 박스를 포함한다. 형상의 경계 박스는 (x_{min} , y_{min} , z_{min} , x_{max} , y_{max} , z_{max})에 의해 정의된다.

[0112]

QP_coord: 이 5 비트 부호가 없는 정수는 3D 모델 형상의 품질 파라미터를 나타낸다. QP_coord의 최소값은 3이고, 최대값은 31이다. QP_coord는 수학적 10에서 QB_Pattern에 대응할 수 있거나, 또는 수학적 15에서 QB_EntireModel에 대응할 수 있다.

[0113]

normal binding: 이 2-비트 부호가 없는 정수는 3D 모델에 법선의 결합을 나타낸다. 허용값은 다음 표에 설명되어 있다.

normal_binding	Binding
0	not_bound
1	bound_per_vertex
2	bound_per_face
3	bound_per_corner

[0114]

[0115]

default_normal_bbox: 이 1-비트 부호가 없는 정수는 항상 '0' 이 되어야 하며, 이는 기본 경계 박스가 전체 3D 모델의 법선에 사용되는 것을 나타낸다. 법선의 디폴트 경계 박스는 $nx_{min}=0.0$, $ny_{min}=0.0$, $nz_{min}=0.0$, $nx_{max}=1.0$, $ny_{max}=1.0$, 및 $nz_{max}=1.0$ 으로서 정의된다.

[0116] **QP_normal**: 이 5-비트 부호가 없는 정수는 3D 모델 형상의 품질 파라미터를 나타낸다. QP_normal의 최소값은 3이고, 최대값은 31이다.

[0117] **color_binding**: 이 2-비트 부호가 없는 정수는 3D 모델에 색상의 결합을 나타낸다. 다음 표는 허용값을 나타낸다.

color_binding	Binding
0	not_bound
1	bound_per_vertex
2	bound_per_face
3	bound_per_corner

[0118]

[0119] **default_color_bbox**: 이 1-비트 부호가 없는 정수는 디폴트 경계 박스가 전체 3D 모델의 색상에 사용되었는지 여부를 나타낸다. 0은 다른 경계 박스를 사용하는 것을 의미하고, 1은 디폴트 경계 박스를 사용하는 것을 의미한다. 디폴트 경계 박스는 $r_{min}=0.0$, $g_{min}=0.0$, $b_{min}=0.0$, $r_{max}=1.0$, $g_{max}=1.0$, 및 $b_{max}=1.0$ 으로서 정의된다.

[0120] **color_bbox**: 이 데이터 필드는 전체 3D 모델의 색상의 경계 박스를 포함한다. 색상 경계 박스는 $(r_{min}, g_{min}, b_{min}, r_{max}, g_{max}, b_{max})$ 에 의해 정의된다.

[0121] **QP_color**: 이 5-비트의 부호가 없는 정수는 색상의 품질 파라미터를 나타낸다. QP_color의 최소값은 3이고, 최대값은 31이다.

[0122] **multi_texCoord_num**: 이 5-비트 부호가 없는 정수는 정점/모서리에 대한 텍스처 좌표의 수를 제공한다.

[0123] **texCoord_binding**: 이 2-비트 부호가 없는 정수는 3D 모델에의 텍스처 좌표의 결합을 나타낸다. 다음 표는 허용값을 나타낸다.

texCoord_binding	Binding
0	forbidden
1	bound_per_vertex

[0124]

2	forbidden
3	bound_per_corner

[0125]

[0126] **default_texCoord_bbox**: 이 1-비트 부호가 없는 정수는 디폴트 경계 박스가 텍스처 좌표에 사용되었는지 여부를 나타낸다. 0은 다른 경계 박스를 사용하는 것을 의미하고, 1은 디폴트 경계 박스를 사용하는 것을 의미한다. 디폴트 경계 박스는 $u_{min}=0.0$, $v_{min}=0.0$, $u_{max}=1.0$, 및 $v_{max}=1.0$ 으로서 정의된다.

[0127] **texCoord_bbox**: 이 데이터 필드는 전체 3D 모델의 텍스처 좌표의 경계 박스를 포함한다. 텍스처 좌표 경계 박스는 $(u_{min}, v_{min}, u_{max}, v_{max})$ 에 의해 정의된다.

[0128] **QP_texCoord**: 이 5-비트의 부호가 없는 정수는 텍스처 좌표의 품질 파라미터를 나타낸다. QP_texCoord의 최소값은 3이고, 최대값은 31이다.

[0129] **multi_attribute_num**: 이 5-비트 부호가 없는 정수는 정점/면/모서리에 대한 속성의 수를 나타낸다.

[0130] **attribute_binding**: 이 2-비트 부호가 없는 정수는 3D 모델에 대한 속성의 결합을 나타낸다. 다음 표는 허용값

을 나타낸다.

attribute_binding	Binding
0	forbidden
1	bound_per_vertex
2	bound_per_face
3	bound_per_corner

[0131]

[0132]

default_attribute_bbox: 이 1-비트 부호가 없는 정수는 디폴트 경계 박스가 속성에 사용되었는지의 여부를 나타낸다. 0은 다른 경계 박스를 사용하는 것을 의미하고, 1은 디폴트 경계 박스를 사용하는 것을 의미한다. 디폴트 경계 박스는 `attribute_min[1..attribute_dim]=0.0`, `attribute_max[1..attribute_dim]=1.0`로서 정의된다.

[0133]

attribute_bbox: 이 데이터 필드는 속성의 경계 박스를 포함한다. 텍스처 좌표 경계 박스는 `(attribute_min[1..attribute_dim], attribute_max[1..attribute_dim])`에 의해 정의된다.

[0134]

QP_attribute: 이 5-비트 부호가 없는 정수는 속성의 품질 파라미터를 나타낸다. QP_attribute의 최소값은 3이고, 최대값은 31이다.

[0135]

표 2에서, 고유 부분에 대한 예시적인 구문 및 의미가 본 발명에 대해 설명되어 있다.

[0136]

표 2

class compr_uni_part_data{	Num. of bits	Descriptor
compr_uni_comp_data		
compr_uni_comp_transl	<code>bit_num_uni_comp_transl()</code>	<code>f(bit_num_uni_comp_transl())</code>
}		

[0137]

[0138]

compr_uni_comp_data: 이 데이터 필드는 `3d_model_compr_mode`로 표시되는 압축 방법에 의해 엔코딩된 모든 고유 성분의 압축된 형상, 연결 및 특성을 포함한다. 모든 고유 성분은 압축 이전에 원점으로 변형된다.

[0139]

compr_uni_comp_transl: 이 데이터 필드는 모든 고유 성분에 대한 압축된 변형 벡터를 포함한다. 고유 성분 변형 벡터는 제 1 양자화 및 이후에 엔트로피 코딩에 의해 압축된다. 이 데이터 필드는 `compr_uni_comp_data`를 갖는 고유 성분의 동일한 순서를 사용한다.

[0140]

bit_num_uni_comp_transl(): 이 함수는 QP_coord에 기초하여 각각의 고유 성분 변형 벡터를 양자화하기 위해 사용되는 비트 수를 계산한다.

[0141]

표 3에서 반복 구조의 예시적인 구문과 의미가 본 발명에 대해 설명되어 있다.

[0142]

표 3

class compr_repeat_struct_data{	Num. of bits	Descriptor
compr_pattern_data		
compr_pattern_transl	bit_num_pattern_transl()	f(bit_num_pattern_transl())
if(insta_trans_elem_bit == 1){		
compr_insta_elementary_data		
}		
else{		
compr_insta_grouped_data		
}		
}		

[0143]

[0144]

compr_Pattern_data: 이 데이터 필드는 3d_model_compr_mode로 표시되는 압축 방법에 의해 엔코딩되는 압축된 형상, 모든 패턴의 연결 및 특성을 포함한다.

[0145]

compr_Pattern_transl: 이 데이터 필드는 모든 패턴 성분에 대한 압축된 변형 벡터를 포함한다. 패턴 변형 벡터는 제 1 양자화 및 이후의 엔트로피 코딩에 의해 압축된다. 이 데이터 필드는 compr_pattern_data의 패턴의 동일한 순서를 사용한다.

[0146]

compr_insta_elementary_data: 이 데이터 필드는 "기본 인스턴스 변환 모드"를 사용하여 모든 인스턴스에 대한 압축된 변환 데이터를 포함한다. 이는 바이트 정렬되는 방식으로 압축된다.

[0147]

compr_insta_grouped_data: 이 데이터 필드는 "그룹화 인스턴스 변환 모드"를 사용하여 모든 인스턴스에 대한 압축된 변환 데이터를 포함한다. 이는 바이트 정렬되는 방식으로 압축된다.

[0148]

bit_num_pattern_transl(): 이 함수는 QP_coord에 기초하여 각각의 패턴 성분의 변형 벡터를 양자화하기 위해 사용되는 비트 수를 계산한다.

[0149]

표 4에서, 인스턴스에 대한 예시적인 구문과 의미가 본 발명에 대해 설명되어 있다.

[0150] 표 4

class compr_insta_elementary_data{	Num. of bits	Descriptor
insta_transl_bbox	6*32	
for (i = 0; i < numofInstance; i++) {		
compr_elem_insta_patternID		ec(v)
elem_insta_flip_flag	1	
elem_insta_reflection_flag	1	
elem_insta_attribute_header		
compr_elem_insta_transl	bit_num_insta_transl()	f(bit_num_insta_transl(
))
compr_elem_insta_rotat_spherical		
if (use_scaling_bit){		
compr_elem_insta_scaling		ec(v)
}		
if (need_error_compensation()){		
compr_elem_insta_error_compen_data	bit_num_vertex_coding_error()	f(bit_num_vertex_coding_error())
}		
compr_elem_insta_attribute_data		
}		
}		

[0151]

[0152] **insta_transl_bbox**: 이 데이터 필드는 양자화가 인스턴스 변형 정보를 압축할 때 사용될 수 있도록 모든 변형 벡터의 경계 박스를 포함한다. 이 경계 박스는 **insta_transl_x_{min}**, **insta_transl_y_{min}**, **insta_transl_z_{min}**, **insta_transl_x_{max}**, **insta_transl_y_{max}**, **insta_transl_z_{max}**에 의해 정의된다.

[0153] **compr_elem_insta_patternID**: 이 데이터 필드는 i번째 인스턴스의 압축된 패턴 ID를 포함한다.

[0154] **elem_insta_flip_flag**: 이 1-비트 부호가 없는 정수는 i번째 인스턴스가 해당 패턴과 비교하여 반전(flipped)되었는지의 여부를 나타낸다. 반전된 인스턴스는 인스턴스 삼각형 법선이 해당 패턴 삼각형의 반대 방향으로 있는 것을 의미한다. 0은 i번째 인스턴스가 반전되지 않았다는 것을 의미하고, 1은 i번째 인스턴스가 반전되었음을 의미한다.

[0155] **elem_insta_reflection_flag**: 이 1-비트 부호가 없는 정수는 i번째 인스턴스의 변환이 반영을 포함하는지의 여부를 나타낸다. 0은 i번째 인스턴스의 변환이 반영을 포함하지 않았다는 것을 의미하고, 1은 i번째 인스턴스의 변환이 반영을 포함한다는 것을 의미한다.

[0156] **elem_insta_attribute_header**: 이 데이터 필드는 i번째 인스턴스의 속성 헤더를 포함한다.

[0157] **compr_elem_insta_transl**: 이 데이터 필드는 i번째 인스턴스의 압축된 변형 벡터를 포함한다.

[0158] **compr_elem_insta_rotat_spherical**: 이 데이터 필드는 구형 모드에서 i번째 인스턴스의 압축된 회전 변환을 포함한다.

[0159] **compr_elem_insta_scaling**: 이 데이터 필드는 i번째 인스턴스의 압축된 스케일링 요소를 포함한다.

[0160] **compr_elem_insta_error_compen_data**: 이 데이터 필드는 i번째 인스턴스의 압축된 코딩 에러 보상 데이터를 포함한다.

[0161] **compr_elem_insta_attribute_data**: 이 데이터 필드는 i번째 인스턴스의 압축된 속성 데이터를 포함한다.

[0162] **bit_num_insta_transl()**: 이 함수는 QP_coord에 기초하여 인스턴스 변형 벡터를 양자화하기 위해 사용되는 비트의 수를 계산한다.

[0163] **need_error_compensation()**: 이 함수는 i 번째 인스턴스의 코딩 에러가 보상될 필요성이 있는지의 여부를 판정한다.

[0164] **bit_num_vertex_coding_error()**: 이 함수는 QP_coord에 기초하여 i 번째 인스턴스의 각각의 정점의 코딩 에러를 양자화하기 위해 사용되는 비트의 수를 적응적으로 계산한다.

[0165] 표 5에서, 회전 각도에 대한 예시적인 구문과 의미가 본 발명에 대해 설명되어 있다. 구면 모드에서 i 번째 인스턴스의 회전은 3개의 각도: 알파, 베타 및 감마에 의해 표시된다.

[0166] 표 5

class compr_elem_insta_rotat_spherical{	Num. of bits	Descriptor
compr_elem_insta_rotat_alpha	bit_num_rotat_alpha()	f(bit_num_rotat_alpha())
compr_elem_insta_rotat_beta	bit_num_rotat_beta()	f(bit_num_rotat_beta())
compr_elem_insta_rotat_gamma	bit_num_rotat_gamma()	f(bit_num_rotat_gamma())
}		

[0167]

[0168] **compr_elem_insta_rotat_alpha**: 이 데이터 필드는 i 번째 인스턴스의 회전의 압축된 알파를 포함한다.

[0169] **compr_elem_insta_rotat_beta**: 이 데이터 필드는 i 번째 인스턴스의 회전의 압축된 베타를 포함한다.

[0170] **compr_elem_insta_rotat_gamma**: 이 데이터 필드는 i 번째 인스턴스의 회전의 압축된 감마를 포함한다.

[0171] **bit_num_rotat_alpha()**: 이 함수는 해당 패턴의 스케일 및 QP_coord에 기초하여 i 번째 인스턴스의 회전의 알파 값에 대한 비트의 수를 적응적으로 계산한다.

[0172] **bit_num_rotat_beta()**: 이 함수는 해당 패턴의 스케일 및 QP_coord에 기초하여 i 번째 인스턴스의 회전의 베타 값에 대한 비트의 수를 계산한다.

[0173] **bit_num_rotat_gamma()**: 이 함수는 해당 패턴의 스케일 및 QP_coord에 기초하여 i 번째 인스턴스의 회전의 감마 값에 대한 비트의 수를 계산한다.

[0174] 본 명세서에 기재된 구현은 예를 들어, 방법 또는 프로세스, 장치, 소프트웨어 프로그램, 데이터 스트림, 또는 신호로 구현될 수 있다. 심지어 단일 형태의 구현(예를 들어, 방법으로만 설명)의 컨텍스트에서 설명되었지만, 설명한 기능의 구현은 다른 형태(예를 들어, 장치 또는 프로그램)로 구현될 수도 있다. 한 장치는, 예를 들어, 적당한 하드웨어, 소프트웨어 및 펌웨어로 구현될 수 있다. 방법들은, 예를 들어, 컴퓨터, 마이크로프로세서, 집적 회로, 또는 프로그램 가능한 논리 디바이스를 포함하는 처리 디바이스를 일반적으로 지칭하는 프로세서와 같은 장치로 구현될 수 있다. 또한, 프로세서들은, 예를 들어, 컴퓨터, 셀 폰, 휴대용/개인용 디지털 어시스턴트("PDAs"), 및 단말 사용자들 간의 정보의 통신을 용이하게 하는 다른 디바이스와 같은 통신 디바이스를 포함한다.

[0175] "한 실시예" 또는 "실시예" 또는 "하나의 구현" 또는 본 원리의 "구현", 뿐만 아니라 이들의 다른 변형의 참조는 실시예와 함께 기재된 특정한 기능, 구조, 특성 등이 본 원리의 적어도 한 실시예에 포함된다는 것을 의미한다. 따라서, 명세서 전반에 걸쳐 다양한 곳에 나타나는 "한 실시예에서", 또는 "실시예에서", 또는 "한 구현에서", 또는 "구현에서"라는 문구의 출현뿐만 아니라 임의의 다른 변형의 출현이 반드시 동일한 실시예 모두를 참조하는 것은 아니다.

[0176] 추가적으로, 본 응용 또는 그 청구범위는 정보의 다양한 부분을 "결정하는"을 참조할 수 있다. 정보를 결정하는 것은, 예를 들어, 정보를 추정하고, 정보를 계산하고, 정보를 예측하고, 또는 메모리로부터 정보를 검색하는 것을 하나 이상 포함할 수 있다.

[0177] 또한, 본 응용 또는 그 청구 범위는 정보의 다양한 부분을 "액세스하는"을 참조할 수 있다. 정보를 액세스하는 것은, 예를 들어, 정보를 수신하고, (예를 들어, 메모리로부터) 정보를 검색하고, 정보를 저장하고, 정보를 처리하고, 정보를 전송하고, 정보를 이동시키고, 정보를 복사하고, 정보를 소거하고, 정보를 계산하고, 정보를 결정하고, 정보를 예측하고, 정보를 추정하는 것을 하나 이상 포함할 수 있다.

[0178] 추가적으로, 본 응용 또는 그 청구 범위는 정보의 다양한 부분들을 "수신하는 것"을 지칭할 수 있다. 수신하는

것은 "엑세스하는 것"과 같이, 폭넓은 용어인 것으로 의도된다. 정보를 수신하는 것은, 예를 들어 정보를 액세스하는 것, 또는 (예를 들어, 메모리로부터) 정보를 검색하는 것 중 하나 이상을 포함할 수 있다. 또한, "수신하는 것"은 일반적으로, 예를 들어 정보를 저장하는 것, 정보를 처리하는 것, 정보를 전송하는 것, 정보를 이동시키는 것, 정보를 복사하는 것, 정보를 소거하는 것, 정보를 계산하는 것, 정보를 결정하는 것, 정보를 예측하는 것, 또는 정보를 추정하는 것과 같은 작동 동안에, 하나의 방식 또는 또 다른 방식으로 수반된다.

[0179] 본 기술의 숙련된 사람에게 명백한 것처럼, 구현은 예를 들어, 저장 또는 전송될 수 있는 정보를 운반하도록 포맷된 다양한 신호를 생성할 수 있다. 정보는 예를 들어, 방법을 실행하기 위한 명령어, 또는 기재된 구현 중 하나에 의해 생성된 데이터를 포함할 수 있다. 예를 들어, 기재된 실시예의 비트스트림을 운반하도록 신호가 포맷될 수 있다. 이러한 신호는, 예를 들어, 전자기파(예를 들어, 스펙트럼의 무선 주파수 부분을 사용) 또는 기저대역 신호로서 포맷될 수 있다. 이러한 포맷은, 예를 들어, 데이터 스트림을 인코딩하고, 인코딩된 데이터 스트림으로 캐리어를 변조하는 것을 포함한다. 신호가 운반할 수 있는 정보는, 예를 들어, 아날로그 또는 디지털 정보가 될 수 있다. 이 신호는 공지된 것과 같은 다양한 상이한 유선 또는 무선 링크를 통해 전송될 수 있다. 신호는 프로세서-관독 가능한 매체에 저장될 수 있다.

[0180]

부호의 설명

[0181]

510: 반복 구조 검색 모듈

520: 패턴 인코더

530: 기본 인스턴스 정보 인코더

540: 그룹화 인스턴스 정보 인코더

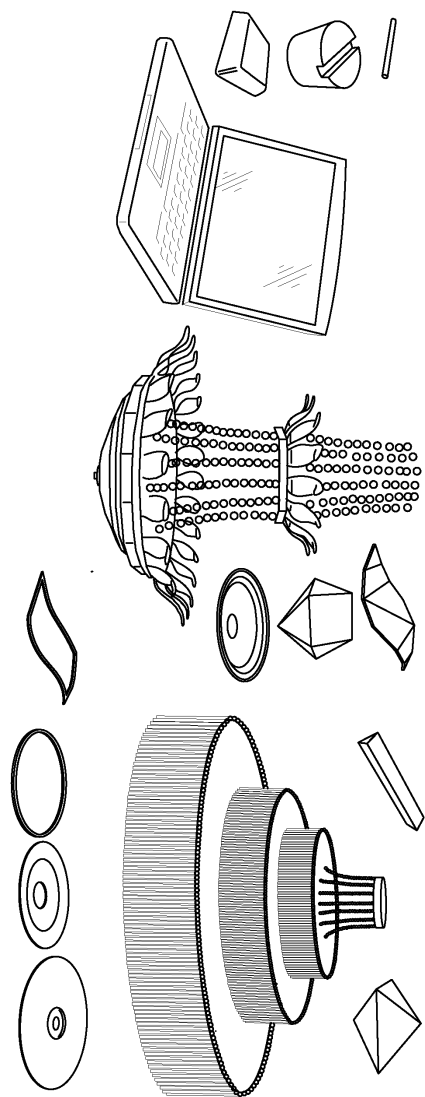
550: 고유 성분 인코더

560: 반복 구조 검증기

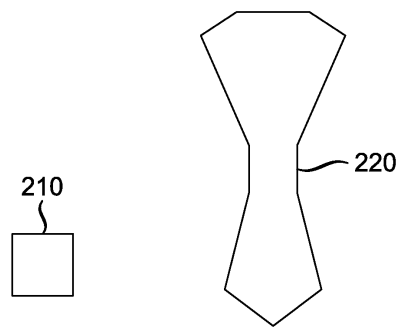
570: 비트스트림 어셈블러

도면

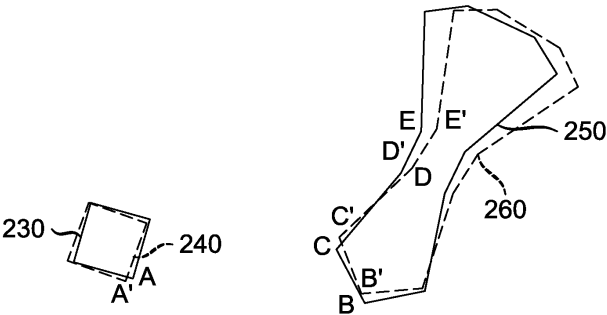
도면1



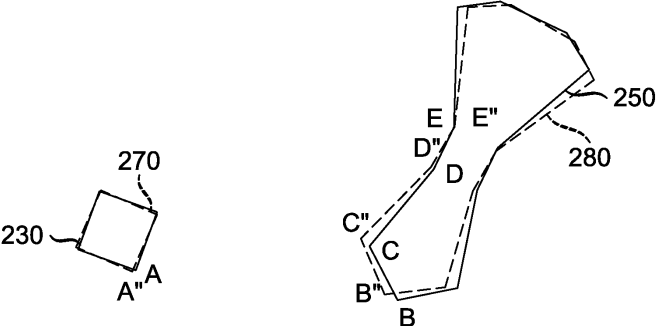
도면2a



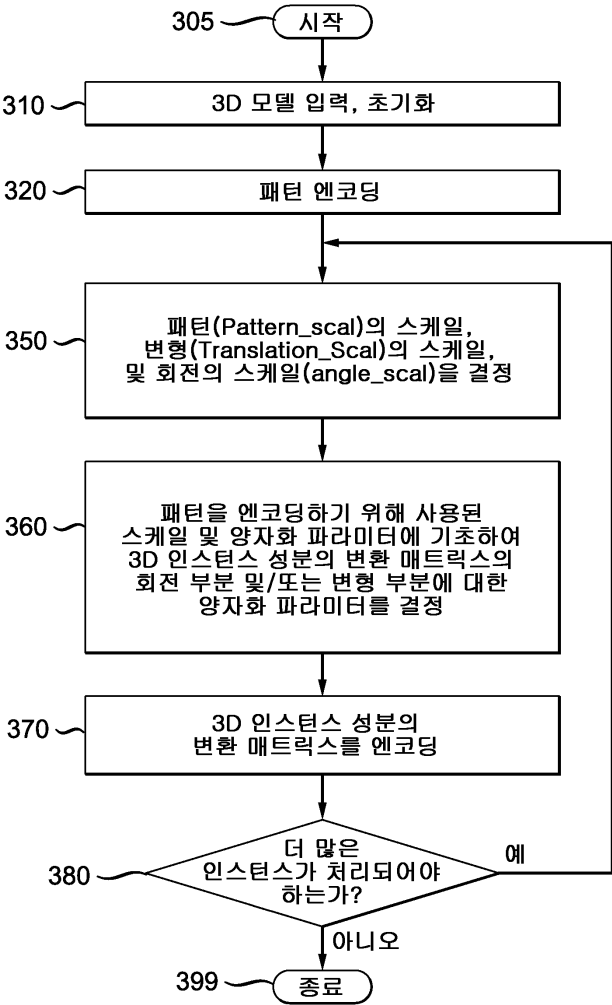
도면2b



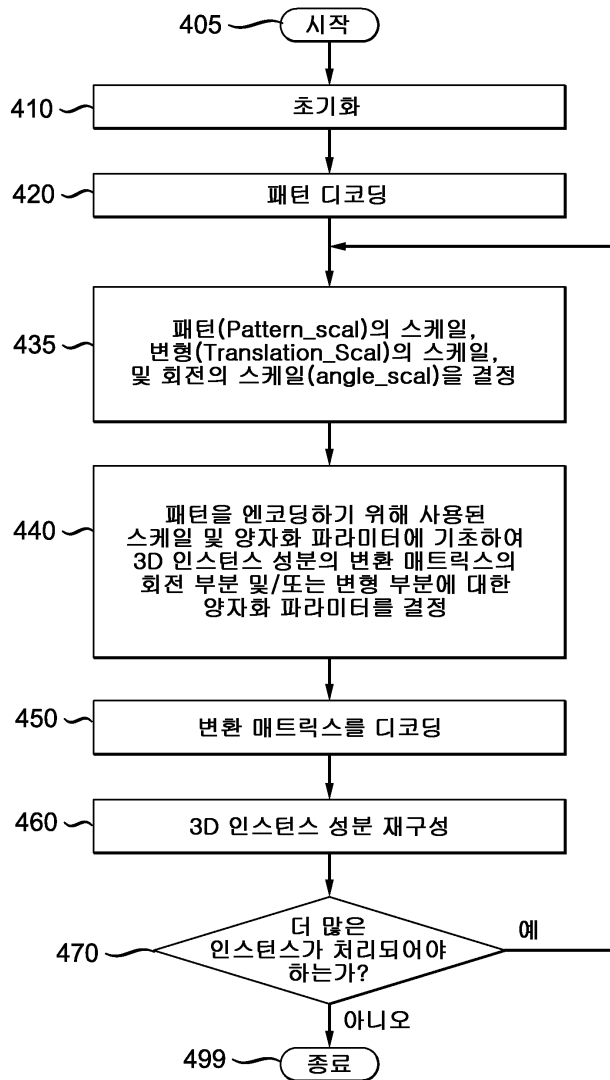
도면2c



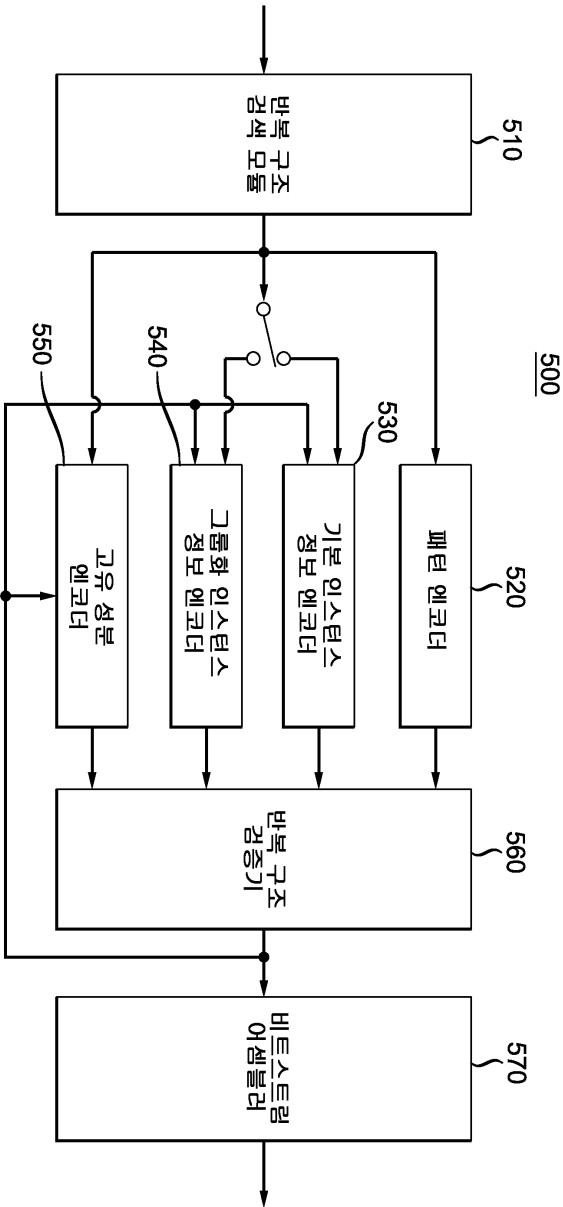
도면3



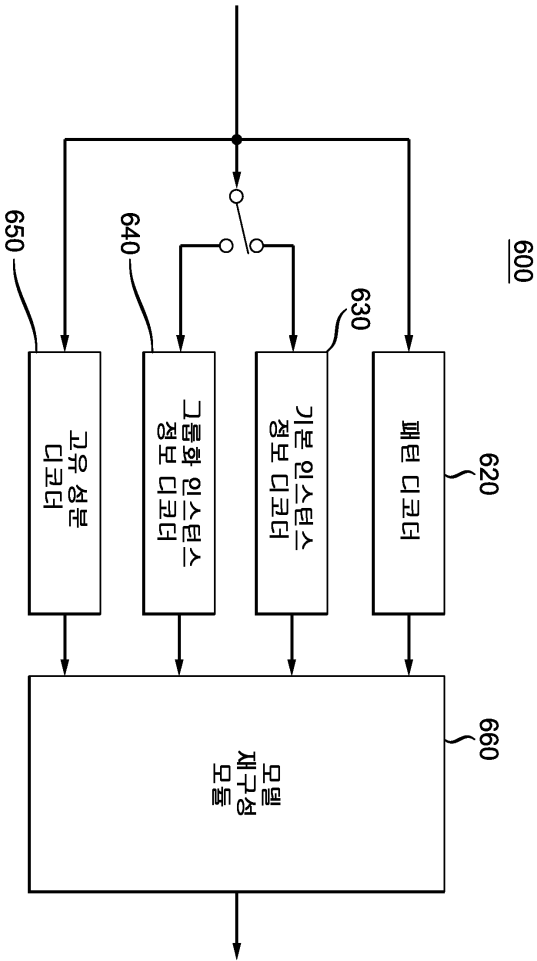
도면4

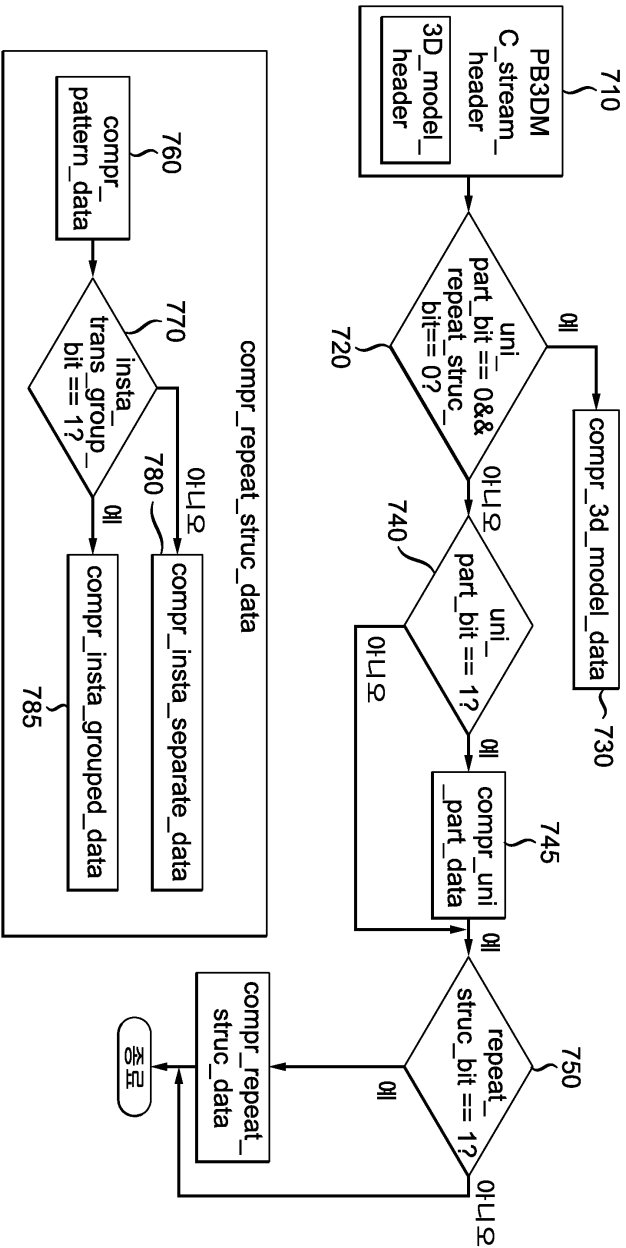


도면5



도면6





도면7