



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0070007
(43) 공개일자 2017년06월21일

- | | |
|---|---|
| <p>(51) 국제특허분류(Int. Cl.)
 <i>H04N 19/46</i> (2014.01) <i>G06T 1/60</i> (2006.01)
 <i>H04N 19/12</i> (2014.01) <i>H04N 19/176</i> (2014.01)
 <i>H04N 19/42</i> (2014.01)</p> <p>(52) CPC특허분류
 <i>H04N 19/46</i> (2015.01)
 <i>G06T 1/60</i> (2013.01)</p> <p>(21) 출원번호 10-2017-7007039
 (22) 출원일자(국제) 2015년10월09일
 심사청구일자 없음
 (85) 번역문제출일자 2017년03월14일
 (86) 국제출원번호 PCT/US2015/054921
 (87) 국제공개번호 WO 2016/057908
 국제공개일자 2016년04월14일
 (30) 우선권주장
 62/062,211 2014년10월10일 미국(US)</p> | <p>(71) 출원인
 어드밴스드 마이크로 디바이시즈, 인코포레이티드
 미국 캘리포니아 94088-3453 서니베일 피.오.박스
 3453 원 에이엠디 플레이스</p> <p>(72) 발명자
 포미아노우스키 앤드류 에스.씨.
 미국 캘리포니아 94087 써니베일 캐릭 코트 577
 아이오우르차 콘스탄틴
 미국 캘리포니아 95120 산 호세 우디드 레이크 드
 라이브 786</p> <p>(74) 대리인
 박장원</p> |
|---|---|

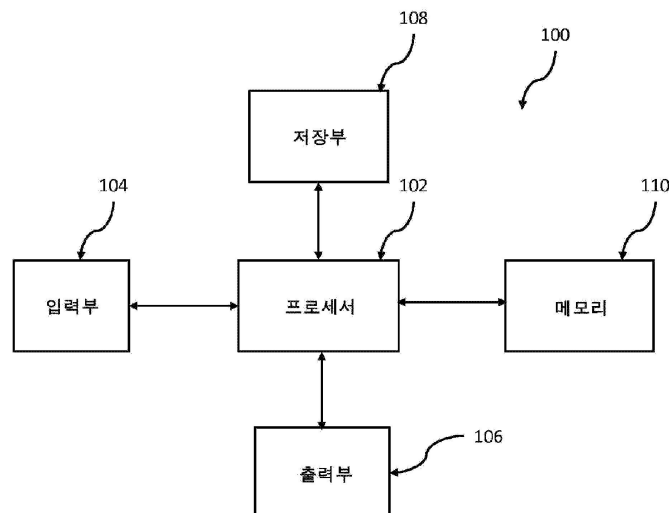
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 하이브리드 블록 기반 압축

(57) 요약

하드웨어 지원형 압축 포맷들을 사용하는 텍스처의 블록 기반 압축을 위한 방법 및 장치가 제공된다. 방법은, 텍스처를 복수의 블록들로 분할하는 단계, 각각의 블록에 대하여 오류 메트릭을 최소화하기 위해 블록과 함께 사용하기 위한 변환을 결정하는 단계, 변환의 적어도 하나의 특성을 그렇지 않았다면 기준 컴포넌트 값들을 나타내기 위하여 이용가능했을 복수의 비트들로 인코딩하는 단계, 및 블록을 압축하는 단계를 포함한다.

대표도 - 도1



(52) CPC특허분류

H04N 19/12 (2015.01)

H04N 19/176 (2015.01)

H04N 19/42 (2015.01)

명세서

청구범위

청구항 1

하드웨어 지원형 압축 포맷들을 사용하는 텍스처(texture)의 블록 기반 압축을 위한 방법으로서,

텍스처를 복수의 블록들로 분할하는 단계;

각각의 블록에 대하여, 오류 메트릭(error metric)을 최소화하기 위하여 상기 블록과 함께 사용하기 위한 변환을 결정하는 단계;

상기 변환의 적어도 하나의 특성을 그렇지 않았다면 기준 컴포넌트 값들을 나타내기 위하여 이용가능했을 복수의 비트들 내에 인코딩하는 단계; 및

상기 블록을 압축하는 단계를 포함하는, 방법.

청구항 2

청구항 1에 있어서, 각각의 블록은 RGB 컬러-공간 내의 2개의 종점들 및 상기 종점들 사이의 라인 상의 픽셀 값들을 나타내는 복수의 인덱스들을 포함하는 데이터를 포함하며, 상기 복수의 비트들은 상기 종점들의 상위 비트들인, 방법.

청구항 3

청구항 2에 있어서, 상기 종점들의 각각의 상기 상위 비트들이 동일할 때 상기 특성이 상기 복수의 비트들 내에 인코딩되는, 방법.

청구항 4

청구항 1에 있어서, 제 1 블록을 인코딩하기 위하여 사용되는 상기 복수의 비트들의 수는 제 2 블록을 인코딩하기 위하여 사용되는 상기 복수의 비트들의 수와는 상이한, 방법.

청구항 5

청구항 1에 있어서, 상기 블록과 연관된 데이터는 상기 하드웨어 지원형 압축 포맷에 의해 직접적으로 지원되지 않는, 방법.

청구항 6

청구항 1에 있어서, 상기 특성은 스케일링 인자(scaling factor) 및 지수(exponent) 중 하나를 나타내는, 방법.

청구항 7

이미지를 압축해제하기 위한 방법으로서,

상기 이미지의 선택된 영역과 연관된 복수의 블록들을 디코딩하는 단계;

각각의 블록에 대하여, 오류 메트릭을 최소화하기 위해 상기 블록과 함께 사용하기 위한 변환의 특성을 획득하는 단계로서, 상기 특성은 그렇지 않았다면 기준 컴포넌트 값들을 나타내기 위하여 이용가능했을 복수의 비트들 내에 인코딩된, 단계;

상기 블록을 압축해제하는 단계;

상기 블록에 적용하기 위한 상기 변환을 획득하기 위하여 상기 특성을 사용하는 단계; 및

상기 변환을 적용하는 단계를 포함하는, 방법.

청구항 8

청구항 7에 있어서, 각각의 블록은 RGB 컬러-공간 내의 2개의 중점들 및 상기 중점들 사이의 라인 상의 픽셀 값들을 나타내는 복수의 인덱스들을 포함하는 데이터를 포함하며, 상기 복수의 비트들은 상기 중점들의 상위 비트들인, 방법.

청구항 9

청구항 8에 있어서, 상기 중점들의 각각의 상기 상위 비트들이 동일할 때 상기 특성이 상기 복수의 비트들 내에 인코딩되는, 방법.

청구항 10

청구항 7에 있어서, 제 1 블록 내의 상기 특성을 인코딩하기 위하여 사용되는 상기 복수의 비트들의 수는 제 2 블록 내의 상기 특성을 인코딩하기 위하여 사용되는 상기 복수의 비트들의 수와는 상이한, 방법.

청구항 11

하드웨어 지원형 압축 포맷들을 사용하는 텍스처의 블록 기반 압축을 위한 시스템으로서,

제 1 압축 포맷을 지원하도록 구성된 프로세서; 및

인코더로서,

각각의 블록에 대하여, 상기 제 1 압축 포맷에 의해 지원되지 않는 데이터를 포함하는 상기 블록에 기인하는 오류 메트릭을 최소화하기 위하여 상기 블록과 함께 사용하기 위한 변환의 특성을 결정하고;

상기 특성을 그렇지 않았다면 기준 컴포넌트 값들을 나타내기 위하여 이용가능했을 복수의 비트들 내에 인코딩하며; 및

상기 블록을 압축하도록 구성된, 상기 인코더를 포함하는, 시스템.

청구항 12

청구항 11에 있어서, 각각의 블록은 RGB 컬러-공간 내의 2개의 중점들 및 상기 중점들 사이의 라인 상의 픽셀 값들을 나타내는 복수의 인덱스들을 포함하는 데이터를 포함하며, 상기 복수의 비트들은 상기 중점들의 상위 비트들인, 시스템.

청구항 13

청구항 12에 있어서, 상기 중점들의 각각의 상기 상위 비트들이 동일할 때 상기 특성이 상기 복수의 비트들 내에 인코딩되는, 시스템.

청구항 14

청구항 11에 있어서, 제 1 블록을 인코딩하기 위하여 사용되는 상기 복수의 비트들의 수는 제 2 블록을 인코딩하기 위하여 사용되는 상기 복수의 비트들의 수와는 상이한, 시스템.

청구항 15

청구항 11에 있어서, 상기 시스템은,

디코더로서,

상기 압축된 블록을 압축해제하며;

상기 블록에 적용하기 위한 상기 변환을 획득하기 위하여 상기 특성을 사용하고; 및

상기 변환을 적용하도록 구성된, 상기 디코더를 더 포함하는, 시스템.

청구항 16

하드웨어 지원형 압축 포맷을 사용하여 텍스처의 블록 기반 압축을 수행하기 위하여 컴퓨팅 디바이스에 의한 실행을 위한 명령어들의 세트를 저장하는 비-일시적인 컴퓨터-판독가능 저장 매체로서, 상기 명령어들의 세트는, 상기 컴퓨팅 디바이스에 의해 실행될 때 상기 컴퓨팅 디바이스로 하여금 하기의 동작들을 수행하게끔 하는 명령

어들을 포함하며, 상기 동작들은,

텍스처를 복수의 블록들로 분할하는 동작;

각각의 블록에 대하여, 오류 메트릭을 최소화하기 위하여 상기 블록과 함께 사용하기 위한 변환을 결정하는 동작;

상기 변환의 적어도 하나의 특성을 그렇지 않았다면 기준 컴포넌트 값들을 나타내기 위하여 이용가능했을 복수의 비트들 내에 인코딩하는 동작; 및

상기 블록을 압축하는 동작을 포함하는, 비-일시적인 컴퓨터-판독가능 저장 매체.

청구항 17

청구항 16에 있어서, 각각의 블록은 RGB 컬러-공간 내의 2개의 종점들 및 상기 종점들 사이의 라인 상의 픽셀 값들을 나타내는 복수의 인덱스들을 포함하는 데이터를 포함하며, 상기 복수의 비트들은 상기 종점들의 상위 비트인, 비-일시적인 컴퓨터-판독가능 저장 매체.

청구항 18

청구항 17에 있어서, 상기 종점들의 각각의 상기 상위 비트들이 동일할 때 상기 특성이 상기 복수의 비트들 내에 인코딩되는, 비-일시적인 컴퓨터-판독가능 저장 매체.

청구항 19

청구항 16에 있어서, 제 1 블록을 인코딩하기 위하여 사용되는 상기 복수의 비트들의 수는 제 2 블록을 인코딩하기 위하여 사용되는 상기 복수의 비트들의 수와는 상이한, 비-일시적인 컴퓨터-판독가능 저장 매체.

청구항 20

청구항 16에 있어서, 상기 비-일시적인 컴퓨터-판독가능 저장 매체는, 상기 컴퓨팅 디바이스에 의해 실행될 때 상기 컴퓨팅 디바이스로 하여금 하기의 동작들을 수행하게끔 하는 명령어들을 더 포함하며, 상기 동작들은,

상기 블록을 압축해제하는 동작;

상기 블록에 적용하기 위한 상기 변환을 획득하기 위하여 상기 특성을 사용하는 동작; 및

상기 변환을 적용하는 동작을 포함하는, 비-일시적인 컴퓨터-판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 관련 출원들에 대한 상호 참조

[0002] 본 출원은 2014년 10월 10일자로 출원된 미국 가특허 출원번호 제62/062,211호에 대한 우선권을 주장하며, 이로써 이러한 출원의 전체 개시내용이 본원에 참조로서 명백하게 포함된다.

[0003] 본 개시는 전반적으로 하드웨어 지원형 압축 포맷들을 사용하는 텍스처(texture)의 블록 기반 압축을 위한 방법 및 장치에 관한 것으로서, 더 구체적으로는, 기존의 하드웨어 지원형 압축 포맷들과 호환이 불가능한 일부 방식의 데이터의 압축을 가능하게 하기 위하여 소프트웨어를 사용하는 방법 및 장치에 관한 것이다.

배경 기술

[0004] 컴퓨터 그래픽 프로세싱 시스템들은 전형적으로 대부분의 프로세싱을 수행하는 하나 이상의 그래픽 프로세싱 유닛(graphics processing unit; GPU)들을 가지고 많은 양의 데이터를 프로세싱한다. GPU는 렌더링(rendering)으로서 알려진 프로세스에서 그것의 컴포넌트들의 더 높은 레벨의 묘사(description)로부터 이미지를 구성하는 픽셀들을 생성한다. GPU들은 전형적으로 픽셀, 텍스처, 및 기하학적 데이터를 프로세싱하기 위한 컴퓨팅 엘리먼트들을 사용함으로써 연속적인 렌더링을 수행한다. 컴퓨팅 엘리먼트들은 래스터라이저(rasterizer)들, 셋업 엔진들, 컬러 블렌더(color blender)들, 히든 표면 제거, 및 텍스처 매핑의 기능들을 실행할 수 있다. 이러한 컴퓨팅 엘리먼트들은 흔히 셰이더(shader)들, 셰이더 프로세서들, 셰이더 어레이들, 셰이더 파이프, 셰이더 파이프

어레이들, 셰이더 파이프라인들, 또는 셰이더 엔진으로 지칭된다. "셰이더"는 또한 소프트웨어 명령어들(이는 보다 더 정확하게는 "셰이더 프로그램"으로 지칭됨)을 실행하기 위해 사용되는 실제 하드웨어 컴포넌트 또는 프로세서를 나타낼 수 있다. 셰이더 프로세서 또는 프로그램은 데이터를 판독하고 렌더링하며, 임의의 유형의 데이터의 프로세싱을 수행한다.

[0005] 복잡한 그래픽 신(scene)들을 생성하는데 수반되는 프로세싱 중 다수는 텍스처 데이터를 수반한다. 텍스처들은, 높이 맵 데이터, 이미지 데이터, 컬러 또는 투명도 데이터, 거칠기/매끄러움(roughness/smoothness) 데이터, 반사율 데이터, 룩업(lookup) 테이블, 또는 다른 데이터와 같은 다양한 유형들의 데이터 중 임의의 데이터일 수 있다. 텍스처들은, 노멀(normal) 또는 뱀프(bump) 맵들의 형태의 미세 표면 세부사항들 또는 거울 반사와 같은 표면 속성들을 포함하는 다수의 속성들을 지정할 수 있다. 때때로 용어 '텍셀(texel)'이 당업계에서 사용되며, 여기에서 "텍셀"은 '픽셀'이 픽처 엘리먼트인 것과 마찬가지로 텍스처 엘리먼트이다. 그러나, 용어들 '텍셀' 및 '픽셀'은 ("텍셀"이 때때로 "픽셀"로서 간단하게 지칭된다는 점에 있어서) 흔히 상호교환적으로 사용된다. 따라서, 각각의 용어가 본 개시 내에서 상호교환적으로 사용될 수 있다.

[0006] 사실적인 컴퓨터 그래픽을 제공하는 것은 전형적으로 다수의 고-품질의 상세화된 텍스처들을 필요로 한다. 그러나, 텍스처들의 사용은 많은 양의 저장 공간 및 메모리-대-GPU 대역폭을 소비할 수 있으며, 따라서 텍스처들은 일반적으로 저장 공간 및 대역폭 사용을 감소시키기 위하여 압축된다.

[0007] 따라서, 텍스처 압축은 일반 및 특히 3D 그래픽 하드웨어에 있어서 광범위하게 인정되는 그래픽 하드웨어의 특징이 되었다. 텍스처 압축의 목표는, 가능한 한 원본 텍스처의 품질을 최대한 유지하면서 그래픽 시스템 상의 저장 및 대역폭 비용들을 감소시키는 것이다. 다수의 텍스처 압축 기법들은, 이하에서 추가적으로 설명되는 바와 같은 종점(endpoint)들 및 보간된 픽셀 값들을 인코딩하기 위하여 다양한 블록-기반 압축 방법들 중 하나를 사용한다. DirectX에서 사용되는 Microsoft[®] BCn이 이러한 기법들의 예들이다. 이러한 그리고 다른 포맷들을 보다 더 유연하게 사용하고 그럼으로써 원래 설계되었던 포맷들에 대한 데이터와는 상이한 새로운 유형의 데이터(예를 들어, 높이 맵 데이터)의 인코딩을 허용하기 위한 방식들을 개발하는 것이 바람직하다.

[0008] 더 구체적으로, 새로운 텍스처 압축 기술들을 개발하고 하드웨어 형태로 시장에 전달하기 위하여 긴 시간을 소요한다. 하드웨어 기반 압축의 하나의 버전으로부터 다음 버전으로의 전환 시간 동안, 소프트웨어 개발자들은, 기존의 압축 기법들이 이에 대하여 부적절한 결과들을 제공하는 새로운 압축 기술들을 필요로 하는 새로운 유형들의 데이터를 생성한다. 이는 이러한 새로운 기술들의 식별과 시장에 이들을 실제로 도입하는 것 사이의 공백을 생성한다.

[0009] 따라서, 이에 의해 커스텀(custom) 데이터가 기존의 압축 기법들을 사용하여 인코딩되도록 블록별로 변화될 수 있으며, 그럼으로써 개발될 하드웨어 기반 디코딩 기법들 및 소프트웨어의 유연한 하이브리드들을 가능하게 함으로써 기존의 하드웨어를 사용하여 새로운 텍스처 압축 가능성들을 제공하기 위한 방법에 대한 필요성이 존재한다.

발명의 내용

과제의 해결 수단

[0010] 본 개시의 일 실시예에 따르면, 하드웨어 지원형 압축 포맷들을 사용하는 텍스처의 블록 기반 압축을 위한 방법이 제공되며, 상기 방법은, 텍스처를 복수의 블록들로 분할하는 단계, 각각의 블록에 대하여 오류 메트릭(error metric)을 최소화하기 위해 블록과 함께 사용하기 위한 변환(transform)을 결정하는 단계, 변환의 적어도 하나의 특성을 그렇지 않았다면 기준 컴포넌트 값들을 나타내기 위하여 이용가능했을 복수의 비트들 내에 인코딩하는 단계, 및 블록을 압축하는 단계를 포함한다. 이러한 실시예의 하나의 측면에 있어서, 각각의 블록은 RGB 컬러-공간 내의 2개의 종점들 및 종점들 사이의 라인(line) 상의 픽셀 값들을 나타내는 복수의 인덱스들을 포함하는 데이터를 포함하며, 복수의 비트들은 종점들의 상위 비트들이다. 이러한 측면의 변형예에 있어서, 종점들의 각각의 상위 비트들이 동일할 때 특성이 복수의 비트들로 인코딩된다. 다른 측면에 있어서, 제 1 블록을 인코딩하기 위하여 사용되는 복수의 비트들의 수는 제 2 블록을 인코딩하기 위하여 사용되는 복수의 비트들의 수와는 상이하다. 또 다른 측면에 있어서, 블록과 연관된 데이터는 하드웨어 지원형 압축 포맷에 의해 직접적으로 지원되지 않는다. 또 다른 측면에 있어서, 특성은 스케일링 인자(scaling factor) 및 지수(exponent) 중 하나를 나타낸다.

[0011] 본 개시의 다른 실시예에 따르면, 이미지를 압축해제 하기 위한 방법이 제공되며, 상기 방법은, 이미지의 선택

된 영역과 연관된 복수의 블록들을 디코딩하는 단계, 각각의 블록에 대하여 오류 메트릭을 최소화하기 위하여 블록과 함께 사용하기 위한 변환의 특성을 획득하는 단계로서, 상기 특성은 그렇지 않았다면 기준 컴포넌트 값들을 나타내기 위하여 이용가능했을 복수의 비트들 내에 인코딩된, 단계, 블록에 적용하기 위한 변환을 획득하기 위하여 특성을 사용하여 블록을 압축해제하는 단계, 및 변환을 적용하는 단계를 포함한다. 이러한 실시예의 하나의 측면에 따르면, 각각의 블록은 RGB 컬러-공간 내의 2개의 중점들 및 2개의 중점들 사이의 라인 상의 픽셀 값들을 나타내는 복수의 인덱스들을 포함하는 데이터를 포함하며, 복수의 비트들은 중점들의 상위 비트들이다. 이러한 측면의 변형예에 있어서, 중점들의 각각의 상위 비트들이 동일할 때 특성이 복수의 비트들로 인코딩된다. 다른 측면에 있어서, 제 1 블록 내의 특성을 인코딩하기 위하여 사용되는 복수의 비트들의 수는 제 2 블록 내의 특성을 인코딩하기 위하여 사용되는 복수의 비트들의 수와는 상이하다.

[0012] 본 개시의 또 다른 실시예에 있어서, 제 1 압축 포맷을 지원하도록 구성된 프로세서, 및 인코더를 포함하는, 하드웨어 지원형 압축 포맷들을 사용하는 텍스처의 블록 기반 압축을 위한 시스템이 제공된다. 인코더는, 각각의 블록에 대하여, 제 1 압축 포맷에 의해 지원되지 않는 데이터를 포함하는 블록에 기인하는 오류 메트릭을 최소화하기 위하여 블록과 함께 사용하기 위한 변환의 특성을 결정하고, 특성을 그렇지 않았다면 기준 컴포넌트 값들을 나타내기 위해 이용가능했을 복수의 비트들 내에 인코딩하며, 및 블록을 압축하도록 구성된다. 이러한 실시예의 하나의 측면에 있어서, 각각의 블록은 RGB 컬러-공간 내의 2개의 중점들 및 중점들 사이의 라인 상의 픽셀 값들을 나타내는 복수의 인덱스들을 포함하는 데이터를 포함하며, 복수의 비트들은 중점들의 상위 비트들이다. 이러한 측면의 변형예에 있어서, 중점들의 각각의 상위 비트들이 동일할 때 특성이 복수의 비트들로 인코딩된다. 다른 측면에 있어서, 제 1 블록을 인코딩하기 위하여 사용되는 복수의 비트들의 수는 제 2 블록을 인코딩하기 위하여 사용되는 복수의 비트들의 수와는 상이하다. 다른 측면은 압축된 블록을 압축해제하고, 블록에 적용하기 위한 변환을 획득하기 위하여 특성을 사용하며, 변환을 적용하도록 구성된 디코더를 더 포함한다.

[0013] 본 개시의 또 다른 실시예는 하드웨어 지원형 압축 포맷을 사용하여 텍스처의 블록 기반 압축을 수행하기 위하여 컴퓨팅 디바이스에 의한 실행을 위한 명령어들의 세트를 저장하는 비-일시적인 컴퓨터-판독가능 저장 매체를 제공하며, 상기 명령어들의 세트는, 컴퓨팅 디바이스에 의해 실행될 때 컴퓨팅 디바이스로 하여금 텍스처를 복수의 블록들로 분할하고, 각각의 블록에 대하여, 오류 메트릭을 최소화하기 위해 블록과 함께 사용하기 위한 변환을 결정하며, 변환의 적어도 하나의 특성을 그렇지 않았다면 기준 컴포넌트 값들을 나타내기 위하여 이용가능했을 복수의 비트들 내에 인코딩하고, 및 블록을 압축하게끔 하는 명령어들을 포함한다. 이러한 실시예의 하나의 측면에 있어서, 각각의 블록은 RGB 컬러-공간 내의 2개의 중점들 및 중점들 사이의 라인 상의 픽셀 값들을 나타내는 복수의 인덱스들을 포함하는 데이터를 포함하며, 복수의 비트들은 중점들의 상위 비트들이다. 이러한 측면의 변형예에 있어서, 중점들의 각각의 상위 비트들이 동일할 때 특성이 복수의 비트들로 인코딩된다. 다른 측면에 있어서, 제 1 블록을 인코딩하기 위하여 사용되는 복수의 비트들의 수는 제 2 블록을 인코딩하기 위하여 사용되는 복수의 비트들의 수와는 상이하다. 또 다른 측면은, 컴퓨팅 디바이스에 의해 실행될 때, 컴퓨팅 디바이스로 하여금 블록을 압축해제하고, 블록에 적용하기 위한 변환을 획득하기 위하여 특성을 사용하며, 변환을 적용하게끔 하는 명령어들을 더 포함한다.

도면의 간단한 설명

[0014] 이하의 도면들을 동반할 때 실시예들이 다음의 설명의 관점에서 더 용이하게 이해될 것이며, 도면들에서 유사한 참조 번호들이 유사한 엘리먼트들을 나타낸다.

도 1은 하나 이상의 개시된 실시예들이 구현될 수 있는 예시적인 디바이스의 블록도이다.

도 2는 RGB 컬러 공간 내의 라인으로서 표현되는 데이터의 블록의 개념도이다.

도 3은 본 개시에 따른 텍스처를 압축하기 위한 방법의 순서도이다.

도 4는 본 개시에 따른 텍스처를 압축해제하기 위한 방법의 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0015] 간략하게, 일 예에 있어서, 하드웨어 지원형 압축 포맷들을 사용하는 텍스처의 블록 기반 압축을 위한 방법 및 장치가 제공된다. 방법은, 텍스처를 복수의 블록들로 분할하는 단계, 각각의 블록에 대하여 오류 메트릭을 최소화하기 위해 블록과 함께 사용하기 위한 변환을 결정하는 단계, 변환의 적어도 하나의 특성을 그렇지 않았다면 기준 컴포넌트 값들을 나타내기 위하여 이용가능했을 복수의 비트들로 인코딩하는 단계, 및 블록을 압축하는 단계를 포함한다. 따라서, 본 개시는, 이에 의해 커스텀 데이터가 기존의 압축 기법들을 사용하여 인코딩되도록

블록별로 변화될 수 있으며, 그럼으로써 개발될 하드웨어 기반 디코딩 기법들 및 소프트웨어의 유연한 하이브리드들을 가능하게 함으로써 기존의 하드웨어를 사용하여 새로운 텍스처 압축 가능성들을 제공하기 위한 접근방식을 제공한다.

- [0016] 도 1은 하나 이상의 개시된 실시예들이 구현될 수 있는 예시적인 디바이스(100)의 블록도이다. 특정 실시예들에 있어서, 디바이스(100)는, 게이밍 디바이스, 핸드헬드 디바이스, 셋-탑 박스, 텔레비전, 모바일 폰, 또는 태블릿 또는 다른 유형의 컴퓨터이다. 디바이스(100)는 일반적으로 프로세서(102), 하나 이상의 입력 디바이스들(104), 하나 이상의 출력 디바이스들(106), 하나 이상의 저장 디바이스들(108) 및 하나 이상의 메모리 디바이스들(110)을 포함한다.
- [0017] 프로세서(102)는 중앙 프로세싱 유닛(central processing unit; CPU), 그래픽 프로세싱 유닛(GPU)를 포함할 수 있으며, CPU 및 GPU는 동일한 다이(die) 또는 하나 이상의 프로세서 코어들 상에 위치되고, 여기에서 각각의 프로세서 코어는 CPU 또는 GPU일 수 있다. 디지털 신호 프로세서(digital signal processor; DSP)들, 가속 프로세싱 유닛(accelerated processing unit; APU)들, 필드 프로그램가능 게이트 어레이(field programmable gate array; FPGA)들 및 유사한 것을 포함하는 다른 유형들의 집적 회로들 및 프로세서들이 또한 본 개시의 측면들을 구현할 수 있다.
- [0018] 입력부(104)는, 키보드, 키패드, 터치 스크린/패드, 검출기, 마이크, 가속도계, 자이로스코프, 생체인식 스캐너, 또는 네트워크 연결부(예를 들어, 무선 IEEE 802 신호들의 송신 및/또는 수신을 위한 무선 근거리 영역 네트워크 카드) 중 하나 이상을 포함할 수 있다. 출력부(106)는, 디스플레이, 스피커, 프린터, 촉각 피드백 디바이스, 하나 이상의 표시기들, 안테나, 또는 네트워크 연결부(예를 들어, 무선 IEEE 802 신호들의 송신 및/또는 수신을 위한 무선 근거리 영역 네트워크 카드) 중 하나 이상을 포함할 수 있다.
- [0019] 저장부(108)는 고정형 또는 착탈가능 저장 컴포넌트, 예를 들어, 하드 디스크 드라이브, 고체 상태 드라이브, 광 디스크, 또는 플래시 디바이스를 포함할 수 있다. 메모리(110)는 프로세서(102)와 동일한 다이 상에 위치될 수 있거나, 또는 프로세서(102)와 별개로 위치될 수 있다. 메모리(110)는 휘발성 또는 비-휘발성 메모리, 예를 들어, 랜덤 액세스 메모리(random access memory; RAM), 동적 RAM, 또는 캐시를 포함할 수 있다.
- [0020] 디바이스(100)와 같은 디바이스들에 대한 다양한 GPU-지원형 텍스처 압축 포맷들이 수년에 걸쳐 개발되었으며, 이들은 7개의 표준 포맷들, 소위 BC1 내지 BC7(즉, 이상에서 언급된 Microsoft[®] BCn 포맷들)의 세트를 포함한다. 이러한 포맷들은, 예를 들어, 텍스처 맵들의 메모리 사용(저장 및 대역폭)을 감소시키기 위하여 사실적인 3D 게임들에서 광범위하게 사용된다. 이러한 애플리케이션들은 고 해상도 그래픽을 필요로 하며, 이는 디퓨즈 컬러(diffuse color), 노멀 맵, 정반사성 하이라이트, 광택, 방사성 글로우(emissive glow), 및 다른 것들과 같은 더 많은 수의 텍스처들을 필요로 한다. 예상된 시각적 세부사항들을 달성하기 위하여, 막대한 양의 데이터가 프로세서들에 제공되어야만 하며, 이는 매우 높은 메모리 대역폭을 필요로 한다. 텍스처 압축은, 그렇지 않았다면 셰이더 코어들에게 이미지 데이터를 제공하기 위하여 필요하였을 메모리 대역폭을 감소시킨다.
- [0021] BCn 포맷들의 전부에 의해 가능해진 텍스처 압축은 블록 압축에 기반하며, 더 구체적으로는, 픽셀들의 4x4 블록들에 기반한다. 프로세싱될 각각의 이미지는 이러한 블록들로 분할되며, 이들은 포맷에 따라서 8 또는 16 바이트 중 하나로 크기가 고정된다. 이러한 표준 레이아웃 및 메모리 내의 블록들의 연속적인 저장은, GPU가 텍스처의 임의의 부분을 포함하고 있는 임의의 블록을 빠르게 위치시키고 액세스할 수 있기 때문에 효율적인 GPU 렌더링을 가능하게 한다.
- [0022] 각각의 블록은 전체 이미지의 작은 영역을 나타낸다. 다수의 이미지들에 있어서, 임의의 특정한 작은 영역(또는 블록) 내에 매우 제한된 컬러 변화들이 존재한다. 일반적으로, 블록들은 단일 컬러 또는 2개의 컬러들 사이의 그라디언트(gradient)의 셰이드(shade)를 포함한다. BCn 포맷들은, 블록 내의 컬러들의 선명도(definition)를 그들의 공간적인 분포로부터 분리함으로써 이러한 사실을 활용한다.
- [0023] 본 개시의 일 실시예에 있어서, 이하에서 설명되는 바와 같은 압축은, 예를 들어, Microsoft[®] BC5 명세를 사용하여 단일 컴포넌트 높이 맵 텍스처에 대하여 수행된다. 이러한 일 실시예에 있어서, BC5를 사용하는 높이 맵 텍스처의 압축은, 높이 맵이 단일 컴포넌트 텍스처임에 따라 압축해제 동안 2-컴포넌트 텍스처를 일 컴포넌트 텍스처로 변환하는 변환(및 압축 동안의 대응하는 역 변환)을 필요로 한다. 이러한 경우에 있어서, 압축되는 데이터 및 하드웨어 지원형 포맷은 본 개시의 원리들을 적용함으로써 상이한 수의 컴포넌트들(채널들)을 가질 수 있다. 이러한 접근방식은, 높이 맵 텍스처가 하나의 채널(예를 들어, 적색 채널)로부터 상위 비트들을 사용하여 제어 비트들(즉, 변환의 적어도 하나의 특성)을 전파시킴으로써 16 비트들의 전체 동적 범위를 사용하지 않는다

는 사실을 활용한다. 인코딩된 특성(들)은 인코더가 블록에 대하여 사용하기 위한 최소화된 오류를 제공할 변환을 획득하는 것을 가능하게 한다.

[0024] 예로서, BC5 디코더의 2-컴포넌트 출력들이 C1 및 C2(즉, 8.6 채널의 2개의 무부호(unsigned) 채널들)라고 가정하면, 본 개시의 원리들에 따라서 다음의 명령어들이 구현될 수 있다(이하에서 라인들은 참조를 위하여 번호가 매겨진다):

```
float transform (float C1, float C2) { (1)
static int mask1 = (1<<7); (2)
static int mask2 = (1<<6); (3)
static int scale1 = 4; (4)
static int scale2 = 8; (5)
static int scale3 = 32; (6)
static int bias1 = 64; (7)
static int bias2 = 32; (8)
static int bias3 = 32; (9)
if ((int)C2 & mask1 == 0) { (10)
    return (11)
        C1 * 256 + (C2 - bias1) * scale1 (12)
    else if ((int)C2 & mask2 == 0) { (13)
        return (14)
            C1 * 256 + (C2 - mask1 - bias2) * scale2 (15)
    else { (16)
        return (17)
            C1 * 256 + (C2 - (mask1 + mask2) - bias3) * scale3 (18)
    } (19)
} (20)
```

[0025]

[0026] 라인들 (2) 및 (3)은 변환 파라미터들을 디코딩하기 위한 마스크들을 제공한다. 이러한 예에 있어서, 구현될 파라미터들의 코드를 제공하는 C2의 비트들이 존재한다(즉, 비트들 7 및 6은 파라미터 코드 정보를 포함할 수 있다). 라인들 (4) 내지 (9)는 각각의 변환에 대한 3개의 스케일들 및 각각의 변환에 대한 3개의 바이어스(bias)들의 예시적인 정의들을 제공한다. 라인들 (10) 내지 (20)은 변환들을 디코딩하기 위한 명령어들을 제공한다(즉, 적용될 변환의 코드를 결정한다). 이러한 실시예에 있어서, 코드는 가변 길이이다. 비트 7이 0인 경우, (그것이 무엇이든 간에) 변환 1이 적용되어야만 한다. 이는 라인 (12)에 반영된다. 이러한 구현예가 변환의 코드를 인코딩하기 위하여 단지 C2의 1 비트(이러한 예에 있어서, C2의 최상위 비트(most significant bit) - MSB -)만을 사용하기 때문에, 다른 7개의 비트들이 다른 정보를 전달하기 위하여 이용가능하다. MSB = 0인 경우에 있어서, 이를 마스킹하여 제거할 필요가 없다. 일반적으로, 코드를 디코딩한 이후에, 비트는 마스킹되어 제거되지만, 비트가 이미 0인 경우에는 이러한 것이 필요하지 않다. MSB = 1인 경우, 명령어들은 다음 비트를 평가한다. 이는 라인들 (13) 및 (14)에 반영된다. 제 2 비트가 0인 경우, 제 2 변환이 적용될 것이다(라인들 (15) 및 (16) 참조). 마지막으로, 비트들 둘 모두가 1인 경우, 제 3 변환이 사용될 것이며, 비트들 둘 모두가 마스킹된다(라인 (18) 참조).

[0027] 각각의 블록의 픽셀들 또는 텍셀들은 블록의 컬러 팔레트 내로의 인덱스들로서 코딩되며, 이는 전형적으로 이상에서 설명된 바와 같이 매우 제한된다. 픽셀 당 단지 몇 비트들만이 요구되며, 이는 팔레트가 매우 작기 때문이다. 그것의 모든 컬러들이 2개의 중점들 사이에서 연장하는 RGB 컬러 공간 내에서 라인 세그먼트(또는 보간축)를 따라서 균등하게 이격된 지점들에 속한다고 가정함으로써 팔레트가 압축된다. 이러한 가정을 사용하면,

라인의 종점들만이 저장되도록 요구되며, 다른 컬러들은 상이한 비율들로 2개의 종점들을 블렌딩(blend)함으로써 재구성된다.

[0028] 도 2는 이상에서 설명된 RGB 컬러 공간 내의 라인 세그먼트(200)의 개념도를 제공한다. 이러한 예가 예시적이지만, 본 개시의 하나의 실시예가, 예를 들어, Microsoft[®] BC5 명세를 사용하는 하드웨어에 의해 지원되는 2 컴포넌트 포맷을 사용하여 1 컴포넌트 (스칼라) 입력 데이터 압축을 사용한다는 것을 주의해야만 한다. 도시된 바와 같이, RGB 컬러 공간은 원점에서 흑색(무광)을 가지며 및 R, G 및 B 축들을 가지고 1차 컬러들(적색, 녹색 및 청색)을 매핑 아웃(map out)하기 위한 3D 직교 좌표계의 단위 정육면체 서브셋을 포함한다. 따라서, 모델이 단위 정육면체이기 때문에, 축들은 각기 0 내지 1 사이의 휘도(intensity) 범위를 갖는다. 라인(200)은 종점(202) 및 종점(204)을 포함한다. 이러한 예에 있어서, 라인(200)은, 선형적 보간을 통해 라인(200)을 따라 균일하게 이격된 2개의 중간 지점들 또는 픽셀 값들(206)을 더 포함한다. 구현예에 의존하여, 그리고 당업자들에 의해 이해되는 바와 같이, 픽셀 값들(206)의 수는 2보다 더 크거나 또는 더 작을 수 있다. 픽셀 값들(206)은 3-비트 인덱스의 형태의 데이터로서 라인(200)에 대응하는 블록 내에 포함된다. 각각의 픽셀 값에 대한 3 비트보다 더 많거나 또는 더 적은 인덱스가 본 개시에 따라서 사용될 수 있다는 것이 이해되어야만 한다.

[0029] 본 개시에 따르면, 종점들(202, 204)로부터 정밀도의 일부 비트들을 희생함으로써, 블록 단위로 추가적인 데이터를 인코딩하는 것이 가능하며, 이는 디코더에 의한 기존의 하드웨어 압축 기법이 수용하도록 설계되지 않은 커스텀 데이터 또는 데이터의 특수 처리를 가능하게 한다는 것이 인식될 것이다. 오로지 예시적인 목적들을 위하여, 3-비트 인덱스를 사용하여 선택된 픽셀 값들(206) 및 8-비트 값들에 의해 지정된 종점들(202, 204)을 갖는 데이터의 채널("적색" 채널)을 인코딩하는 단일 컴포넌트 포맷을 가정한다. 블록 내의 임의의 픽셀에 대한 가능한 컬러는 다음과 같이 유도된다:

```
red_0 = endpoint_0;
red_1 = endpoint_1;
red_2 = (6*red_0 + 1*red_1)/7.0f; // bit code 010
red_3 = (5*red_0 + 2*red_1)/7.0f; // bit code 011
red_4 = (4*red_0 + 3*red_1)/7.0f; // bit code 100
red_5 = (3*red_0 + 4*red_1)/7.0f; // bit code 101
red_6 = (2*red_0 + 5*red_1)/7.0f; // bit code 110
red_7 = (1*red_0 + 6*red_1)/7.0f; // bit code 111
```

[0030]

[0031] 블록 내의 각각의 픽셀에 대하여 어떠한 컬러를 사용할지 선택하기 위하여 3-비트 인덱스가 사용된다. 반환되는 최종 값들은 0 내지 1의 범위 내에 속한다.

[0032] 선형적인 보간을 통해 픽셀 값들(206)이 획득됨에 따라, 각각의 종점(202, 204)의 최상단(top) 비트들이 동일한 경우, 픽셀 값들(206)의 최상단 비트들이 블록에 대하여 일정할 것이다. 추가적으로, 종점들(202, 204)의 각각의 상위 비트들이 동일한 경우, 이러한 상위 비트들의 값들은 데이터의 하위 비트들 중 임의의 것의 보간된 값들에 영향을 주지 않는다. 결과적으로, 어떤 수의 상위 비트들(예를 들어, endpoint_0 및 endpoint_1의 각각으로부터 3 비트들)이 디코더에 의한 특수 처리를 위해 예비(reserve)될 수 있다. 예비된 비트들의 수가 블록별로 변화할 수 있다는 것이 이해되어야만 한다. 달리 말하면, 이러한 비트들은 블록에 대한 특수 데이터를 인코딩하기 위하여 사용될 수 있다.

[0033] 텍스트로부터 임의의 컬러 값의 검색 이후에, 이상에서 설명된 하드웨어 이후에 다음의 소프트웨어 디코딩 단계들이 부가될 수 있다.

```
color = returned_color_from_hardware_decoder; // range (0->1.0)
color = 255.0; // expand the color to match the original endpoint precision
reservedBitmask = 0xe0; // Binary mask 11100000
intColor = (int)color; // get rid of the fractional bits
specialBlockData = (intColor & reservedBitmask) >> 5; // Extract special block bits
color = (float)intColor - (specialBlockData << 5)
color /= 31.0; // rescale the remaining data back into the 0->1 range
```

[0034]

[0035] 이러한 소프트웨어 디코딩 단계는 세이더로 (인코딩에 대하여 8개의 상이한 수정들을 제공하기에 충분한) 3 비트들의 데이터를 통신한다. 본 개시의 일 실시예에 있어서, 데이터의 비트들은, 예컨대 데이터에 적용될 스케일링 인자 또는 지수를 표현함으로써 디코딩된 데이터를 변환하기 위해 사용될 수 있다. 당업자들에게 자명한 바

와 같이, 데이터의 비트들의 다수의 다른 사용들이 이루어질 수 있다. 또한, 이상에서 설명된 방법이 멀티-컴포넌트 압축형 포맷들에서 사용하기 위하여 당업자에 의해 용이하게 수정될 수 있다는 것(즉, 비트들이 이상에서 설명된 것과 동일한 방식으로 독립적인 컴포넌트들 중 임의의 컴포넌트 내에 인코딩될 수 있다는 것)이 이해되어야만 한다.

[0036] 이상에서 설명된 방식에 있어서, (커스텀 데이터를 프로세싱하도록 설계되지 않은) 기존의 압축 기법으로부터 더 양호한 결과들을 제공하기 위하여 커스텀 데이터가 변환될 수 있다. 데이터는 압축 시에 변환되며, 반드시 압축 이후에(즉, 압축해제 및 렌더링 시에) 그것의 원래의 상태로 다시 변환되어야만 한다. 상이한 블록들에 대하여 변환이 상이할 수 있다는 것이 또한 이해되어야만 한다. 이러한 방식에 있어서, 오류 감소 및 품질 개선은 블록-바이-블록(block-by-block) 단위로 가장 적절한 것으로 결정된 다양한 기술들의 사용을 통해 실현될 수 있다. 이러한 블록-바이-블록 변환을 허용하기 위하여, 변환은 반드시 각각의 블록에 대하여 소프트웨어 디코더로 전달되어야만 한다. 이미지 내의 각각의 블록에 대하여, 예를 들어, 오류 메트릭을 최소화하기 위하여 블록과 함께 사용하기 위한 변환이 식별된다. 각각의 블록과 함께 사용하기 위한 변환을 결정하기 위한 방법들에 관한 추가적인 세부사항들은 "REGION-BASED IMAGE COMPRESSION"이라는 명칭으로 2012년 10월 12일자로 출원된 함께 계류 중인 미국 특허출원 번호 제13/651,020호에 제공되며, 이로써 이의 전체 개시내용이 본원에 참조로서 명백히 통합된다. 본 개시의 일 실시예에 있어서, 변환을 이상에서 설명된 종점들의 상위 비트들 내에 포함시킴으로써 변환이 디코더에 전달된다. 디코더가 블록의 인코딩된 픽셀들을 수신할 때, 픽셀의 MSB 내에 파라미터가 제공된다. 본원에서 설명되는 바와 같이, 본 변환들은 2 컴포넌트 포맷(BC5)의 하드웨어 압축해제로부터 2개의 컴포넌트들을 1개 컴포넌트로 결합하는 것을 포함한다. 이에 상응하여, 압축기는 단일 컴포넌트 데이터를 2 컴포넌트 데이터로 변환한다.

[0037] 도 3은 본 개시의 원리들에 따라 텍스처를 압축하는 방법(300)에 대한 순서도를 제공한다. 도시된 바와 같이, 블록(302)에서 압축될 텍스처가 선택된다. 블록(304)에서, 텍스처는 복수의 블록들(예컨대 이상에서 설명된 바와 같은 4x4 블록들)로 분할된다. 블록(306)에서, 현재 블록이 압축을 위해 선택된다. 다음으로, 블록(308)에서, 현재 블록의 종점들(202, 204)의 상위 비트들이 동일하고, 그림으로써 이러한 종점들이 본원에서 설명된 바와 같은 변환으로 대체되는 것을 허용하는지 여부가 결정된다. 상위 비트들이 동일한 경우, 블록(310)에서, 현재 블록과 함께 사용하기 위한 변환이 결정된다. 그렇지 않은 경우, 블록(309)에서, 데이터는 인코딩된 블록 내에서 사용하기 위한 비트들에 대하여 "공간을 만들기" 위하여 라운딩(round)되거나 또는 달리 변환된다. 블록(310)에서, 변환이 필수적이지는 않으며, 하나 이상의 비트들이 나머지가 변환될 필요가 없다는 것을 나타내기 위하여 사용될 수 있다는 것이 결정될 수 있다는 것을 주의해야만 한다. 특정 실시예들에 있어서, (적어도 하나의 비트는 항상 인코딩을 위해 사용되면서) "변환 비트들"의 수가 블록별로 상이하도록 가변 길이 코드들이 변환들을 인코딩하기 위하여 사용될 수 있다는 것이 또한 이해되어야만 한다.

[0038] 현재 블록의 종점들(202, 204)의 상위 비트들이 동일하고, 블록(310)에서 변환 파라미터들이 결정될 때, 블록(312)에서 변환은 당업계에서 공지된 원리들에 따라서 종점들(202, 204)의 상위 비트들 내로 인코딩된다. 그런 다음, 블록(314)에서 현재 블록이 압축된다. 다음으로, 블록(316)에서, 모든 블록들이 프로세싱되었는지 여부가 결정된다. 모든 블록들이 프로세싱된 경우, 블록(318)에서 프로세스가 종료한다. 그렇지 않은 경우, 블록(306)에서 새로운 블록이 선택되며, 방법의 나머지 부분이 반복된다.

[0039] 도 4는 본 개시의 원리들에 따라 텍스처를 압축해제하는 방법(400)에 대한 순서도를 제공한다. 도시된 바와 같이, 블록(402)에서 도 3의 방법(300)에 따라 압축된 블록이 수신된다. 블록(404)에서, 블록이 압축해제된다. 블록(406)에서, 압축해제된 블록이 변환을 포함하는지 여부가 결정된다. 그렇지 않은 경우, 블록(408)에서 블록이 디코딩된다. 블록(406)에서 결정되는 바와 같이 압축해제된 블록이 변환을 포함하는 경우, 블록(410)에서 변환은 블록에 대응하는 데이터를 변환하기 위하여 사용된다. 그런 다음, 블록(408)에서 블록이 디코딩된다.

[0040] 블록(412)에서, 모든 압축해제된 블록들이 프로세싱되었는지 여부가 결정된다. 그러한 경우, 프로세스는 블록(414)에서 종료한다. 그렇지 않은 경우, 블록(404)에서 프로세싱을 위한 다음 블록이 압축해제되며, 이상에서 설명된 프로세스가 반복된다. 이러한 설명에 있어서, 용어 "변환"은 매우 일반적인 의미로 사용되며 그리고 임의의 데이터 프로세싱을 의미한다.

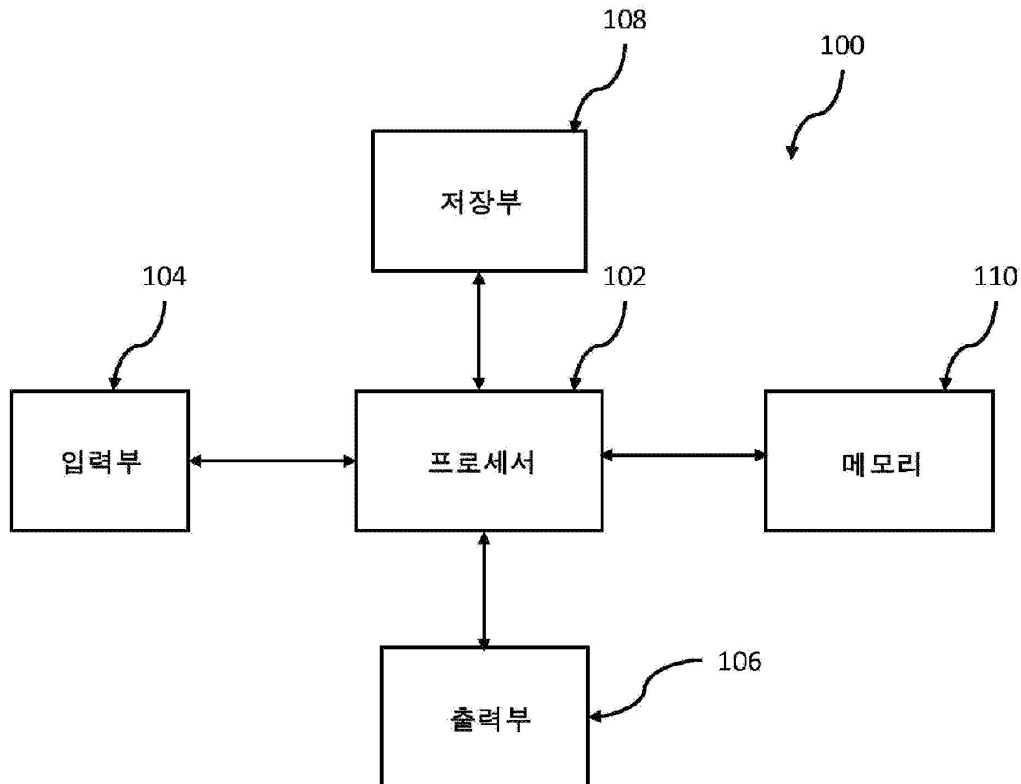
[0041] 이상에서 설명된 방식에 있어서, 더 복잡한 압축 기술들의 구현을 가능하게 하기 위하여 각각의 블록에 대하여 이용되는 압축 방법에 대한 미세(fine-grained) 수정들이 이루어질 수 있다. 따라서, 다른 이점들 중에서도 특히, 본 개시의 방법 및 장치는, 더 복잡한 압축 기법들이 더 복잡한 압축 기법들을 지원하도록 설계되지 않은 기존의 그래픽 하드웨어 상에서 합리적인 짧은 시간 내에 사용되는 것을 가능하게 하기 위한 광범위한 잠재적인

애플리케이션들을 가능하게 한다. 이러한 방식으로, 현재에 이용가능한 하드웨어 압축 기법이 없는 새로운 유형의 데이터(또는 커스텀 데이터), 예컨대 높이 맵 데이터 또는 다른 데이터가 기존의 하드웨어를 사용하여 프로세싱될 수 있다. 다른 이점들이 당업자들에게 자명해질 것이다.

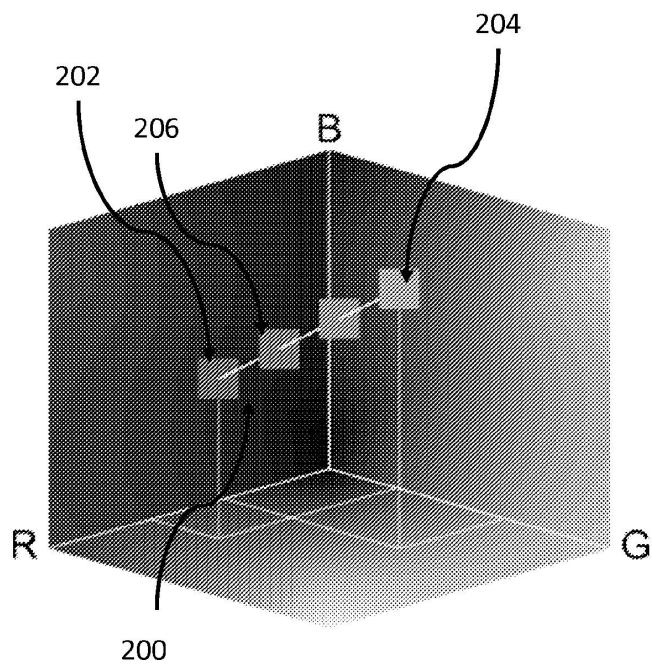
[0042] 본원에서 설명된 예들 및 이상의 상세한 설명은 제한적인 목적이 아니라 오로지 예시 및 설명의 목적들을 위하여 제공되었다. 따라서, 본 개시가 본원에서 청구되고 이상에서 개시된 기초 기본 원리들의 사상 및 범위 내에 속하는 임의의 그리고 모든 수정예들, 변형예들 또는 등가물들을 포괄하는 것으로 고려되어야 한다.

도면

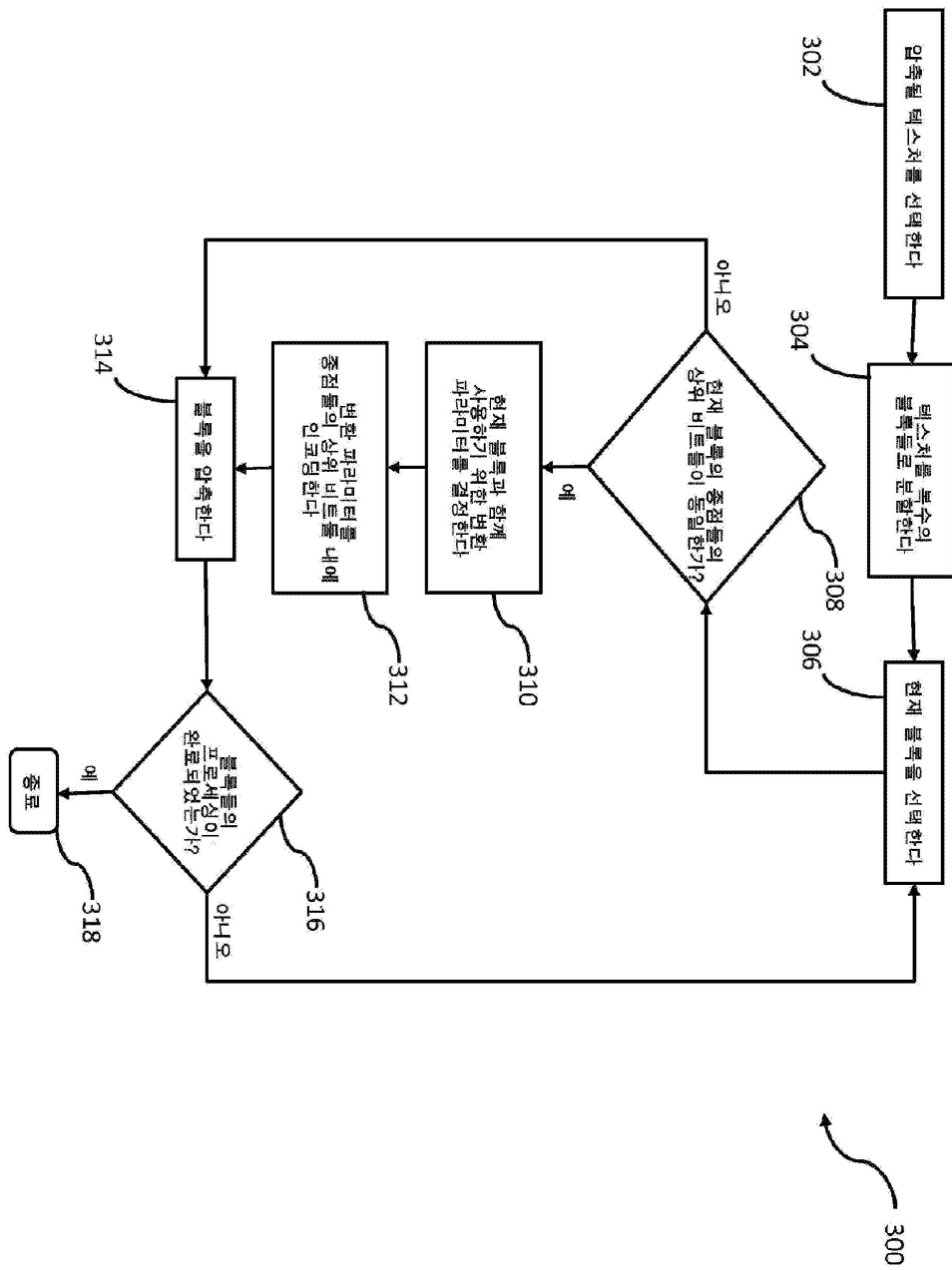
도면1



도면2



도면3



도면4

