



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년07월01일
(11) 등록번호 10-1994986
(24) 등록일자 2019년06월25일

(51) 국제특허분류(Int. Cl.)
G06T 9/00 (2019.01) G06T 11/40 (2006.01)
(21) 출원번호 10-2013-7034226
(22) 출원일자(국제) 2012년06월14일
심사청구일자 2017년06월14일
(85) 번역문제출일자 2013년12월23일
(65) 공개번호 10-2014-0035438
(43) 공개일자 2014년03월21일
(86) 국제출원번호 PCT/US2012/042442
(87) 국제공개번호 WO 2012/174231
국제공개일자 2012년12월20일
(30) 우선권주장
13/163,071 2011년06월17일 미국(US)
(56) 선행기술조사문헌
JP2000105839 A
US6452602 A
US6959110 A
US20110063318 A1

(73) 특허권자
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 95054 산타 클라라 어거스틴 드
라이브 2485
(72) 발명자
이어사 콘스탄틴
미국 캘리포니아 95120 산호세 우디드 레이크 드
라이브 7186
브라더스 존 더블유.
미국 캘리포니아 94085 서니베일 레이크사이드 드
라이브 #1226 1257
(74) 대리인
박장원

전체 청구항 수 : 총 20 항

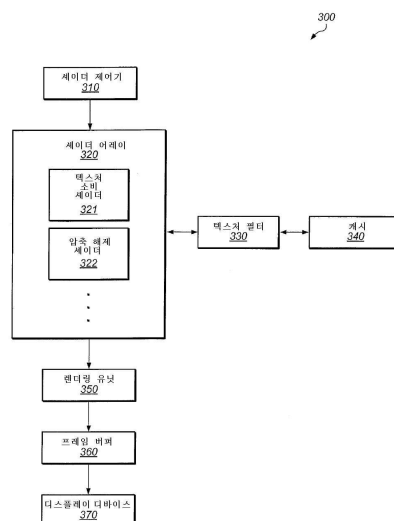
심사관 : 신재철

(54) 발명의 명칭 셰이더 프로세서를 사용한 실시간 온-칩 텍스처 압축 해제

(57) 요약

그래픽 처리 유닛(GPU) 내 텍스처를 압축 해제하거나 생성하는 처리 유닛, 방법 및 매체. 상기 텍스처는 JPEG과 같은 가변 율 압축 방식으로 압축된다. 상기 압축된 텍스처는 시스템 메모리로부터 검색되고 제일 먼저 압축 해제됨이 없이 상기 GPU에 있는 국부 캐시 메모리로 전달된다. 테이블은 상기 압축된 텍스처 내 개별 블록을 찾기 위해 상기 캐시에 의해 사용된다. 압축 해제 셰이더 프로세서는 압축된 블록을 수신하고 이후 진행 중에 상기 블록의 압축 해제를 수행한다. 상기 압축 해제된 블록은 이후 상기 GPU의 텍스처 소비 셰이더 프로세서에 의해 통상적으로 처리된다.

대표도 - 도3



명세서

청구범위

청구항 1

장치로서,

제 1 셰이더(shader); 및

상기 제 1 셰이더에 의한 사용을 위해 텍스처의 압축된 버전의 요청된 블록을 압축해제하도록 구성된 제 2 셰이더를 포함하고, 상기 텍스처의 압축된 버전은 가변율 압축(variable-rate compression)을 이용하여 압축되며,

상기 장치는,

상기 텍스처의 압축된 버전을 위해 제 1 가상 어드레스 공간을 할당하고, 상기 텍스처의 비압축된 버전을 위해 상기 제 1 가상 어드레스 공간과는 다른 제 2 가상 어드레스 공간을 할당하고;

상기 텍스처의 비압축된 버전 내에 있는 블록들의 위치들과 상기 텍스처의 압축된 버전 내의 블록들의 위치들을 맵핑하도록 테이블을 이용하고, 상기 테이블은 상기 텍스처의 압축된 버전의 각 블록의 위치를 저장하며;

상기 제 1 셰이더로부터 요청을 수신하고, 상기 제 1 셰이더는 상기 텍스처의 비압축된 버전 내의 요청된 블록의 가상 어드레스를 결정하며, 상기 요청은 상기 가상 어드레스를 포함하고;

상기 가상 어드레스를 이용하여 캐시에 액세스하고;

상기 블록이 캐시 내에 있으며 그리고 상기 가상 어드레스가 상기 제 1 가상 어드레스 공간에 대응된다라고 결정함에 응답하여, 상기 요청된 블록을 압축된 데이터로서 프로세스하고, 상기 요청된 블록을 압축된 데이터로서 프로세스하기 위해 상기 장치는:

상기 텍스처의 압축된 버전 내의 상기 요청된 블록의 위치를 상기 테이블로부터 결정하고;

상기 요청된 블록의 압축된 버전을 상기 제 2 셰이더에 제공하고;

상기 요청된 블록의 비압축된 버전을 상기 제 2 셰이더로부터 수신하고; 그리고

상기 요청된 블록의 비압축된 버전을 상기 제 1 셰이더에 제공하며;

상기 블록이 캐시 내에 있으며 그리고 상기 가상 어드레스가 상기 제 2 가상 어드레스 공간에 대응된다라고 결정함에 응답하여, 상기 블록을 비압축된 데이터로서 프로세스하고 그리고 비압축된 블록을 상기 제 1 셰이더에 제공하는 것을 특징으로 하는 장치.

청구항 2

제1항에 있어서,

상기 텍스처의 압축된 버전 내에 있는 블록들의 사이즈들은 변할 수 있으며, 상기 장치는 또한, 상기 텍스처의 압축된 버전 내에 있는 요청된 블록의 사이즈를 판별하도록 구성되는 것을 특징으로 하는 장치.

청구항 3

제2항에 있어서,

상기 제 2 셰이더는 또한, 상기 요청된 블록의 어드레스에 기초하여 상기 요청된 블록이 압축되었는지 또는 비압축되었는지를 판별하도록 구성되며, 제 1 어드레스 범위는 비압축된 블록들의 가상 어드레스들에 대응하고, 제 2 어드레스 범위는 압축된 블록들의 물리적 어드레스들에 대응하는 것을 특징으로 하는 장치.

청구항 4

제2항에 있어서,

상기 위치는 상기 텍스처의 압축된 버전의 시작부로부터 상기 요청된 블록까지의 오프셋(offset)에 대응하며,

상기 장치는 또한, 상기 텍스처의 압축된 버전의 복수의 블록들 중 각각의 블록에 대한 DC 계수값을 상기 테이블에 저장하도록 구성되는 것을 특징으로 하는 장치.

청구항 5

제4항에 있어서,

상기 장치는 또한, 상기 텍스처의 압축된 버전의 블록들에 대한 압축 해제를 용이하게 하기 위해, 허프만(Huffman) 코드들 및 상기 텍스처의 압축된 버전의 양자화(quantization) 테이블들을 상기 테이블에 저장하도록 구성되는 것을 특징으로 하는 장치.

청구항 6

제1항에 있어서,

상기 장치는 또한, 상기 텍스처의 압축된 버전을 JPEG-압축된 텍스처에서 하드웨어 내부 JPEG 포맷(hardware internal JPEG format)으로 재포맷하도록 구성되며, 상기 하드웨어 내부 JPEG 포맷은 상기 텍스처의 압축된 버전 내의 다양한 블록들의 위치들 및 사이즈들에 관한 정보를 구비한 테이블을 포함하는 것을 특징으로 하는 장치.

청구항 7

제1항에 있어서,

상기 위치는 상기 텍스처의 압축된 버전의 시작부로부터 상기 요청된 블록까지의 오프셋에 대응하며, 그리고 상기 장치는 상기 텍스처의 압축된 버전 내의 상기 요청된 블록의 오프셋과 인접한 블록의 오프셋 사이의 차이값으로부터 상기 요청된 블록의 사이즈를 계산하는 것을 특징으로 하는 장치.

청구항 8

제1항에 있어서,

상기 위치는, 캐시의 페이지에 상기 요청된 블록이 저장되는 오프셋에 대응하는 것을 특징으로 하는 장치.

청구항 9

제1항에 있어서, 상기 장치는,

상기 가상 어드레스를 상기 텍스처의 압축된 버전 내의 어드레스에 맵핑하고; 그리고

상기 텍스처의 압축된 버전 내의 상기 어드레스를 메모리로 전달하도록 구성되는 것을 특징으로 하는 장치.

청구항 10

제1항에 있어서,

오프-칩 시스템 메모리로부터 검색된 데이터를 저장하도록 된 온-칩 메모리를 더 포함하며, 상기 요청된 블록의 비압축된 버전이 상기 온-칩 메모리에 없다라고 판별함에 응답하여 상기 요청된 블록의 비압축된 버전에 대해 상기 온-칩 메모리에 저장부(storage)가 할당되는 것을 특징으로 하는 장치.

청구항 11

제1항에 있어서,

텍스처 데이터를 절차적으로(procedurally) 생성하도록 구성된 제 3 셰이더를 더 포함하며, 절차적으로 생성된 텍스처 데이터는 미리(in advance)가 아니라 동적으로(dynamically) 수행되는 것을 특징으로 하는 장치.

청구항 12

텍스처 데이터를 압축해제하는 방법으로서, 상기 방법은,

제 1 셰이더가 텍스처 블록을 요청하는 단계;

상기 제 1 셰이더에 의한 사용을 위해 제 2 셰이더가 가변율 압축(variable rate compressed) 텍스처 블록을 압축해제하고 그리고 텍스처의 압축된 버전 내에 있는 요청된 텍스처 블록의 위치를 결정하는 단계;

상기 텍스처의 압축된 버전을 위해 제 1 가상 어드레스 공간을 할당하고, 텍스처의 비압축된 버전을 위해 상기 제 1 가상 어드레스 공간과는 다른 제 2 가상 어드레스 공간을 할당하는 단계;

상기 제 2 셰이더에 의해서, 상기 텍스처의 비압축된 버전 내에 있는 블록들의 위치들과 상기 텍스처의 압축된 버전 내의 블록들의 위치들을 맵핑하는 테이블을 이용하는 단계, 상기 테이블은 상기 텍스처의 압축된 버전의 각 블록의 위치를 저장하며;

상기 제 1 셰이더로부터 요청을 수신하는 단계, 상기 제 1 셰이더는 상기 텍스처의 비압축된 버전 내의 요청된 블록의 가상 어드레스를 결정하며, 상기 요청은 상기 가상 어드레스를 포함하고;

상기 가상 어드레스를 이용하여 캐시에 액세스하는 단계;

상기 블록이 캐시 내에 있으며 그리고 상기 가상 어드레스가 상기 제 1 가상 어드레스 공간에 대응된다라고 결정함에 응답하여;

상기 요청된 블록을 압축된 데이터로서 프로세스하는 단계, 상기 요청된 블록을 압축된 데이터로서 프로세스하는 단계는,

상기 텍스처의 압축된 버전 내의 상기 요청된 블록의 위치를 상기 테이블로부터 결정하는 단계;

상기 요청된 블록의 압축된 버전을 상기 제 2 셰이더에 제공하는 단계;

상기 요청된 블록의 비압축된 버전을 상기 제 2 셰이더로부터 수신하는 단계; 및

상기 요청된 블록의 비압축된 버전을 상기 제 1 셰이더에 제공하는 단계를 포함하며,

상기 블록이 캐시 내에 있으며 그리고 상기 가상 어드레스가 상기 제 2 가상 어드레스 공간에 대응된다라고 결정함에 응답하여, 상기 블록을 비압축된 데이터로서 프로세스하고 그리고 비압축된 블록을 상기 제 1 셰이더에 제공하는 단계를 포함하는 것을 특징으로 하는 텍스처 데이터를 압축해제하는 방법.

청구항 13

제12항에 있어서,

상기 텍스처의 압축된 버전 내에 있는 상기 블록들의 사이즈들은 변환 수 있으며, 그리고 상기 방법은 제 2 셰이더가 상기 텍스처의 압축된 버전 내에 있는 요청된 블록의 사이즈를 판별하는 단계를 더 포함하는 것을 특징으로 하는 텍스처 데이터를 압축해제하는 방법.

청구항 14

제13항에 있어서,

상기 제 2 셰이더에 의하여, 상기 요청된 블록의 어드레스에 기초하여 상기 요청된 블록이 압축되었는지 또는 비압축되었는지를 판별하는 단계를 더 포함하며, 제 1 어드레스 범위는 비압축된 블록들의 가상 어드레스들에 대응하고, 제 2 어드레스 범위는 압축된 블록들의 물리적 어드레스들에 대응하는 것을 특징으로 하는 텍스처 데이터를 압축해제하는 방법.

청구항 15

제13항에 있어서,

상기 위치는 상기 텍스처의 압축된 버전의 시작부로부터 상기 요청된 블록까지의 오프셋(offset)에 대응하고,

상기 방법은, 상기 텍스처의 복수의 블록들 중 각각의 블록에 대한 DC 계수값을 상기 테이블에 저장하는 단계를 더 포함하며, 상기 텍스처의 복수의 블록들 중 각각의 블록은 8×8 픽셀 타일인 것을 특징으로 하는 텍스처 데이터를 압축해제하는 방법.

청구항 16

제12항에 있어서,

상기 방법은, 상기 텍스처의 압축된 버전을 JPEG-압축된 텍스처에서 하드웨어 내부 JPEG 포맷(hardware internal JPEG format)으로 재포맷하는 단계를 더 포함하며, 상기 하드웨어 내부 JPEG 포맷은 상기 텍스처의 압축된 버전 내의 다양한 블록들의 위치들 및 사이즈들에 관한 정보를 구비한 테이블을 포함하는 것을 특징으로 하는 텍스처 데이터를 압축해제하는 방법.

청구항 17

제12항에 있어서,

상기 위치는, 캐시의 페이지에 상기 요청된 블록이 저장되는 오프셋에 대응하는 것을 특징으로 하는 텍스처 데이터를 압축해제하는 방법.

청구항 18

프로그램 명령들을 포함하는 비-일시적인 컴퓨터 판독가능한 저장매체로서, 실행될 때 상기 프로그램 명령들은,

제 1 셰이더가 텍스처 블록을 요청하게 하고;

상기 제 1 셰이더에 의한 사용을 위해 제 2 셰이더가 가변율 압축(variable rate compressed) 텍스처 블록을 압축해제하고 그리고 텍스처의 압축된 버전 내에 있는 요청된 텍스처 블록의 위치를 결정하게 하고;

상기 텍스처의 압축된 버전을 위해 제 1 가상 어드레스 공간을 할당하고, 텍스처의 비압축된 버전을 위해 상기 제 1 가상 어드레스 공간과는 다른 제 2 가상 어드레스 공간을 할당하게 하고;

상기 제 2 셰이더가, 상기 텍스처의 비압축된 버전 내에 있는 블록들의 위치들과 상기 텍스처의 압축된 버전 내의 블록들의 위치들을 맵핑하는 테이블을 이용하게 하고, 상기 테이블은 상기 텍스처의 압축된 버전의 각 블록의 위치를 저장하며;

상기 제 1 셰이더로부터 요청을 수신하게 하고, 상기 제 1 셰이더는 상기 텍스처의 비압축된 버전 내의 요청된 블록의 가상 어드레스를 결정하며, 상기 요청은 상기 가상 어드레스를 포함하고;

상기 가상 어드레스를 이용하여 캐시에 액세스하게 하고;

상기 블록이 캐시 내에 있으며 그리고 상기 가상 어드레스가 상기 제 1 가상 어드레스 공간에 대응된다라고 결정함에 응답하여;

상기 요청된 블록을 압축된 데이터로서 프로세스하게 하고, 상기 요청된 블록을 압축된 데이터로서 프로세스하는 것은,

상기 텍스처의 압축된 버전 내의 상기 요청된 블록의 위치를 상기 테이블로부터 결정하고;

상기 요청된 블록의 압축된 버전을 상기 제 2 셰이더에 제공하고;

상기 요청된 블록의 비압축된 버전을 상기 제 2 셰이더로부터 수신하고; 및

상기 요청된 블록의 비압축된 버전을 상기 제 1 셰이더에 제공하는 것을 포함하며,

상기 블록이 캐시 내에 있으며 그리고 상기 가상 어드레스가 상기 제 2 가상 어드레스 공간에 대응된다라고 결정함에 응답하여, 상기 블록을 비압축된 데이터로서 프로세스하고 그리고 비압축된 블록을 상기 제 1 셰이더에 제공하게 하는 것을 포함하는 비-일시적인 컴퓨터 판독가능한 저장매체.

청구항 19

제18항에 있어서,

상기 텍스처의 압축된 버전 내에 있는 상기 블록들의 사이즈들은 변할 수 있으며, 그리고 상기 프로그램 명령들은 또한, 제 2 셰이더가 상기 텍스처의 압축된 버전 내에 있는 요청된 블록의 사이즈를 판별하게 하는 것을 특징으로 하는 비-일시적인 컴퓨터 판독가능한 저장매체.

청구항 20

제19항에 있어서,

상기 프로그램 명령들은 또한, 상기 제 2 셰이더가 상기 요청된 블록의 어드레스에 기초하여 상기 요청된 블록이 압축되었는지 또는 비압축되었는지를 판별하게 하며, 제 1 어드레스 범위는 비압축된 블록들의 가상 어드레스들에 대응하고, 제 2 어드레스 범위는 압축된 블록들의 물리적 어드레스들에 대응하는 것을 특징으로 하는 비-일시적인 컴퓨터 판독가능한 저장매체.

발명의 설명

기술 분야

[0001] 본 발명은 일반적으로 그래픽 처리에 관한 것으로, 보다 상세하게는 텍스처 압축 해제(texture decompression) 처리 유닛, 방법 및 매체에 관한 것이다.

배경 기술

[0002] 컴퓨터 그래픽 처리 시스템은 일반적으로 대부분의 처리를 수행하는 그래픽 처리 유닛(graphics processing unit: GPU)을 구비하며 많은 양의 데이터를 처리한다. GPU는 특히, 그래픽-처리 태스크(task)를 수행하도록 구성된 복잡한 집적 회로이다. 예를 들어, GPU는 비디오-게임 애플리케이션과 같은 최종-유저 애플리케이션에 의해 요구되는 그래픽-처리 태스크를 실행할 수 있다. GPU는 이산 디바이스이거나 또는 중앙 처리 유닛(central processing unit: CPU)과 같은 다른 프로세서와 동일한 디바이스에 포함될 수 있다.

[0003] GPU는 렌더링(rendering)으로 알려진 프로세스에서 그 구성요소에 대해 더 높은 레벨의 설명으로부터 이미지를 구성하는 픽셀을 생성한다. GPU는 일반적으로 컴퓨팅 요소를 사용하여 연속적으로 렌더링하여 픽셀, 텍스처 및 기하학적 데이터를 처리하는 개념을 사용한다. 컴퓨팅 요소는 래스터라이저(rasterizer), 설정 엔진(setup engine), 컬러 혼합기(color blender), 은닉된 표면 제거 및 텍스처 맵핑의 함수를 실행할 수 있다. 이들 컴퓨팅 요소는 종종 셰이더(shader), 셰이더 프로세서, 셰이더 어레이, 셰이더 파이프, 셰이더 파이프 어레이, 셰이더 파이프라인, 또는 셰이더 엔진으로 지칭되고, 여기서 "셰이더"는 기본적으로 렌더링 효과를 수행하도록 그래픽 자원에 의해 사용되는 소프트웨어 명령 또는 프로그램의 세트를 지칭하는 컴퓨터 그래픽 용어이다. "셰이더"는 또한 소프트웨어 명령을 실행하는데 사용되는 실제 하드웨어 구성요소 또는 프로세서를 지칭할 수 있다. 셰이더 프로세서 또는 프로그램은 데이터를 판독하고 렌더링하며 임의의 유형의 데이터 처리를 수행할 수 있다. 통일된 셰이더를 구비하는 GPU는 또한 픽셀, 정점(vertex), 프리미티브(primitive) 및 일반화된 컴퓨팅 처리로부터 많은 유형의 셰이더 처리를 동시에 지원한다.

[0004] 복잡한 그래픽 장면을 생성하는데 수반되는 많은 처리는 텍스처 데이터를 수반한다. 텍스처는 여러 유형의 데이터, 예를 들어 컬러, 투명도, 룩업 테이블(lookup table), 또는 다른 데이터 중 어느 것일 수 있다. 일부 실시예에서, 텍스처는 시각적 상세를 추가하기 위해 기하학적 형상에 그려지는 디지털화된 이미지일 수 있다. 텍스처를 사용하여 다수의 상세는 모델이 목적지 이미지(destination image)를 생성하도록 렌더링될 때 그래픽 모델의 표면으로 맵핑될 수 있다. 텍스처 맵핑의 목적은 객체의 표면에 현실감(realistic appearance)을 제공하는 것이다. 텍스처는 컬러, 거울 반사와 같은 표면 특성, 또는 통상(normal) 또는 범프(bump) 맵 형태인 정밀한(fine) 표면 상세를 포함하는 많은 특성을 지정할 수 있다. 텍스처는 또한 이미지 데이터, 컬러 또는 투명도 데이터, 러프니스(roughness)/평활도(smoothness) 데이터, 반사율 데이터 등일 수 있다. '텍셀(texel)'은 '픽셀'이 화상 요소인 것과 동일한 방식으로 텍스처 요소이다. '텍셀' 및 '픽셀'이라는 용어는 본 명세서에서 상호 교환가능하게 사용될 수 있다.

[0005] 3D 컴퓨터 그래픽에서, 객체의 표면 상세는 일반적으로 텍스처를 사용하여 추가된다. 예를 들어, 벽돌 벽의 2D 비트맵 이미지는 텍스처 맵핑을 사용하여, 빌딩의 3D 모델을 나타내는 다각형 세트에 적용되어 벽돌로 만들어진 외관을 이 객체의 3D 렌더링에 제공할 수 있다. 현실적인 컴퓨터 그래픽을 제공하려면 일반적으로 많은 고-품질, 상세 텍스처를 요구한다. 텍스처를 사용하는 것은 많은 양의 저장 공간 및 대역폭을 소비할 수 있고, 그 결과 텍스처는 저장 공간 및 대역폭 이용량을 감소시키기 위해 압축될 수 있다.

[0006] 텍스처 압축은 이에 따라 일반적으로 그래픽 하드웨어 및 특히 3D 그래픽 하드웨어의 널리 허용되는 특징이 되었다. 텍스처 압축의 목적은 가능한 한 원래의 텍스처의 품질을 유지하면서 그래픽 시스템에 저장 및 대역폭 비용을 감소시키는 것이다. 본 명세서에서 설명된 압축 및 압축 해제 방법은 이미지 데이터, 화상 데이터, 투명도 정보, 평활도 또는 러프니스 데이터, 또는 임의의 다른 유사하게 구조화된 데이터를 포함하는 여러 유형의 텍스

처 정보를 압축하는데 사용될 수 있다. 그리하여, 텍스처라는 용어는 GPU의 일부로 압축되거나 압축 해제된 데이터를 말하는 것으로 본 명세서에서 넓게 사용된다.

[0007] 고정 율 압축 방식(fixed-rate compression scheme)은 전통적으로 텍스처를 압축하는데 사용되어 왔고 일반적으로 가변 율 방식(variable-rate scheme)에 비해 여러 단점을 나타낼 수 있다. 고정 율 압축과는 달리, 가변 율 압축은 보다 융통성이 있어서 원하는 품질로 조정을 할 수 있다. 예를 들어, 가변 율 압축은 무손실 압축(lossless compression)을 달성하도록 설정될 수 있다. 일부 경우에, 가변 율 압축 방식을 사용하면 전통적인 고정 율 압축 방식에 비해 더 우수한 압축을 제공할 수 있다. JPEG(Joint Photographic Experts Group)과 같은 가변 율 압축 방식은, 높은 복잡성 및 구현 비용으로 인해 진행 중에(on-the-fly) 압축 해제가 요구될 때에는 일반적으로 텍스처 압축에 사용되지 않는다. 그리하여, 가변 율 압축된 텍스처를 낮은 비용으로 진행 중에 압축 해제하는 방법 및 메커니즘이 이 기술 분야에 요구된다.

[0008] 전술된 사항을 감안하여, 압축된 텍스처의 실시간 압축 해제를 수행하는 개선된 처리 유닛, 방법, 및 매체가 요구된다.

발명의 내용

[0009] 텍스처 데이터를 압축 해제하는 처리 유닛, 방법 및 매체의 여러 실시예가 고려된다. 일 실시예에서, 복수의 셰이더 중 제1 셰이더는 디스플레이 디바이스에 의해 사용될 데이터를 생성하거나 추가적인 처리를 하는데 텍스처의 블록을 요구할 수 있다. 제1 셰이더는 텍스처의 비압축된 버전(version) 내 블록의 가상 어드레스(virtual address)를 계산하고 이 가상 어드레스를 이 블록 요청과 함께 캐시 메모리 디바이스로 운반하도록 구성될 수 있다. 블록의 비압축된 버전이 캐시에 저장되어 있지 않은 것으로 결정된 것에 응답하여, 복수의 셰이더 중 제2 셰이더는 압축 해제 셰이더로 개시될 수 있고, 블록의 비압축된 버전의 가상 어드레스는 압축 해제 셰이더로 전달될 수 있다. 또한, 블록의 비압축된 버전이 캐시에 있지 않은 것으로 결정된 것에 응답하여, 캐시 라인이 요청된 블록에 할당될 수 있다.

[0010] 제2 셰이더는 캐시로부터 블록의 압축된 버전을 수신하도록 구성될 수 있다. 캐시는 텍스처의 비압축된 버전의 가상 어드레스 공간을 텍스처의 압축된 버전의 어드레스 공간으로 맵핑하는 테이블을 사용하도록 구성될 수 있다. 캐시 및/또는 제2 셰이더는 테이블로부터 블록의 압축된 버전의 위치 및 사이즈를 결정하도록 구성될 수 있다. 테이블은 또한 텍스처의 각 블록의 압축된 버전의 DC 계수값과 같은 추가적인 정보를 더 포함할 수 있다.

[0011] 캐시로부터 블록의 압축된 버전을 수신한 후에, 제2 셰이더는 블록의 압축된 버전을 압축 해제하고, 이후 블록의 압축 해제된 버전을 캐시에 기록하도록 구성될 수 있다. 블록의 압축 해제된 버전이 캐시에 기록된 후에, 제1 셰이더는 캐시로부터 블록의 압축 해제된 버전을 수신하도록 구성될 수 있다. 제1 셰이더는 이후 블록의 압축 해제된 버전을 처리하여 디스플레이를 위한 렌더링된 표면에 적용될 수 있도록 구성될 수 있다.

[0012] 이들 및 다른 특징 및 장점은 본 명세서에 제시된 방법의 이하 상세한 설명을 참조하여 이 기술 분야에 통상의 지식을 가진 자에게 명백하게 될 것이다.

도면의 간단한 설명

[0013] 시스템, 방법, 및 메커니즘의 상기 및 추가적인 장점은 첨부 도면과 관련하여 이하 상세한 설명을 참조하여 더 잘 이해될 수 있을 것이다.

도 1은 컴퓨터 그래픽 시스템의 일 실시예를 도시한 도면;

도 2는 하나 이상의 실시예에 따른 GPU의 블록도;

도 3은 그래픽 처리 시스템의 일 실시예의 블록도;

도 4a는 데이터 캐시의 일 실시예의 블록도;

도 4b는 하나 이상의 실시예에 따른 블록 맵핑 테이블을 도시한 도면;

도 5는 텍셀의 8×8 블록을 위한 가상 어드레스 공간의 일 실시예를 도시한 도면;

도 6은 데이터의 일부분의 일 실시예의 블록도;

도 7은 텍스처의 압축된 블록을 압축 해제하는 방법의 일 실시예를 도시한 일반화된 흐름도.

발명을 실시하기 위한 구체적인 내용

- [0014] 이하 상세한 설명에서, 다수의 특정 상세들이 본 명세서에 제시된 방법 및 메커니즘을 보다 잘 이해할 수 있도록 하기 위하여 제시된다. 그러나, 이 기술 분야에 통상의 지식을 가진 자라면 여러 실시예는 이들 특정 상세 없이 실시될 수 있다는 것을 이해할 수 있을 것이다. 일부 경우에, 잘 알려진 구조, 구성요소, 신호, 컴퓨터 프로그램, 및 기술은 본 명세서에 설명된 방법을 불명료하게 하는 것을 피하기 위해 상세히 도시되지 않았다. 설명의 간략함과 명료함을 위하여 도면에 도시된 요소는 반드시 축척에 맞게 그려진 것은 아닌 것으로 이해된다. 예를 들어, 이들 요소의 일부 치수는 다른 요소에 대해 과장되어 있을 수 있다.
- [0015] 본 명세서는 "일 실시예" 또는 "실시예"라는 언급을 포함한다. "일 실시예에서" 또는 "실시예에서"라는 어구의 사용은 반드시 동일한 실시예를 말하는 것은 아니다. 특정 특징, 구조, 또는 특성은 본 발명과 매칭하는 임의의 적절한 방식으로 조합될 수 있다.
- [0016] 용어. 다음 문단은 본 명세서(첨부된 청구범위를 포함하여)에 나오는 용어에 대한 정의 및/또는 문맥을 제공한다.
- [0017] "포함하는". 이 용어는 개방형 용어이다. 첨부된 청구범위에 사용된 바와 같이, 이 용어는 추가적인 구조 또는 단계를 배제하지 않는다. "복수의 세이더 . . . 를 포함하는 GPU"라는 청구항을 고려해보자. 이 청구항은 GPU가 추가적인 구성요소(예를 들어, 텍스처 유닛, 입력/출력 회로 등)를 포함하는 것을 배제하지 않는다.
- [0018] "~하도록 구성된". 여러 유닛, 회로, 또는 다른 구성요소는 태스크 또는 태스크들을 수행"하도록 구성된" 것으로 설명되고 청구될 수 있다. 이 문맥에서, "~하도록 구성된"은 유닛/회로/구성요소가 동작 동안 이 태스크 또는 태스크들을 수행하는 구조(예를 들어, 회로)를 포함하는 것을 나타내는 것에 의해 구조를 의미하는 데 사용된다. 그리하여, 이 유닛/회로/구성요소는 특정된 유닛/회로/구성요소가 현재 동작하지 않는 경우에도 (예를 들어, 온(on)이 아닌 경우에도) 태스크를 수행하도록 구성된 것이라고 말할 수 있다. "~하도록 구성된"이라는 언어로 사용된 유닛/회로/구성요소는 하드웨어-예를 들어, 회로, 동작을 구현하도록 실행가능한 프로그램 명령을 저장하는 메모리 등을 포함한다. 유닛/회로/구성요소가 하나 이상의 태스크를 수행"하도록 구성된" 것이라는 언급은 이 유닛/회로/구성요소에 대해 35 U.S.C. § 114의 제6절을 상기시키고자 명시적으로 의도한 것이 아니다. 추가적으로, "~하도록 구성된"은 소프트웨어 및/또는 펌웨어(예를 들어, 소프트웨어를 실행하는 FPGA 또는 일반 목적 프로세서)에 의해 조작되어 해당 태스크(들)를 수행할 수 있는 방식으로 동작하도록 하는 일반 구조(예를 들어, 일반 회로)를 포함할 수 있다. "~하도록 구성된"은 또한 하나 이상의 태스크를 구현하거나 수행하도록 적응된 디바이스(예를 들어, 집적 회로)를 제조하도록 제조 프로세스(예를 들어, 반도체 제조 시설)를 적용하는 것을 더 포함할 수 있다.
- [0019] "제1", "제2" 등. 본 명세서에 사용된 바와 같이, 이들 용어는 선행하는 명사에 대한 레이블(label)로 사용되고, 어느 유형의 순서(예를 들어, 공간적, 시간적, 논리적 등)를 의미하는 것이 아니다. 예를 들어, 8개의 처리 요소 또는 코어를 구비하는 프로세서에서, "제1" 및 "제2" 처리 요소라는 용어는 8개의 처리 요소 중 임의의 2개를 지칭하는 데 사용될 수 있다. 다시 말해, "제1" 및 "제2" 처리 요소는 논리적 처리 요소(0 및 1)로 제한되지 않는다.
- [0020] 도 1을 참조하면, 컴퓨터 그래픽 시스템의 일 실시예의 블록도가 도시된다. 컴퓨터 그래픽 시스템(100)은 컴퓨팅 시스템(102) 및 디스플레이 디바이스(114)를 포함한다. 컴퓨팅 시스템(102)은 그래픽 데이터를 처리하기 위한 그래픽 처리 유닛(GPU)(104)을 포함한다. 일부 실시예에서, GPU(104)는 컴퓨팅 시스템(102) 내에 그래픽 카드에 상주할 수 있다. GPU(104)는 그래픽 데이터를 처리하여 디스플레이 디바이스(114)에 디스플레이하기 위해 프레임의 각 픽셀을 위한 컬러 및 휘도값을 생성할 수 있다. GPU(104)는 픽셀 조작을 수행하기 위해 하나 이상의 처리 코어 및/또는세이더 어레이를 포함할 수 있다.
- [0021] 컴퓨팅 시스템(102)은, CPU(미도시)에서 실행될 수 있는, 소프트웨어 프로그램 애플리케이션(108), 애플리케이션 프로그래밍 인터페이스(application programming interface: API)(110), 및 드라이버(112)를 포함할 수 있다. API(110)는 OpenGL 또는 DirectX와 같은 산업 표준 사양에 따를 수 있다. API(110)는 드라이버(112)와 통신할 수 있다. 드라이버(112)는 API(110)로부터 수신된 표준 코드를 GPU(104)에 의해 이해되는 네이티브 명령 포맷으로 변환할 수 있다. GPU(104)는 이후 드라이버(112)로부터 수신된 명령을 실행할 수 있다.
- [0022] 텍스처는 컴퓨팅 시스템(102)의 시스템 메모리(미도시) 또는 다른 저장 디바이스로부터 GPU(104)로 전달될 수 있다. 일 실시예에서, 텍스처는 JPEG 압축을 사용하여 압축될 수 있다. 다른 실시예에서, 다른 유형의 가변 율 압축은 텍스처를 압축하는데 사용될 수 있다. 본 명세서의 나머지에서는, JPEG 유형 인코딩의 예들을 사용하여

여러 실시예를 기술할 것이다. 그러나, 이것은 단지 예시를 위한 것일 뿐, 다른 유형의 가변 을 압축이 또한 본 명세서에 설명된 방법 및 메커니즘에 사용될 수 있다.

[0023] 드라이버(112)는 압축된 텍스처를 타일링 프로세스의 일부로 재포맷할 수 있다. 이 재포맷은 JPEG-압축된 텍스처를 하드웨어 내부 JPEG 포맷으로 트랜스코딩(transcoding)하는 것을 수반할 수 있다. 다른 실시예에서, JPEG-압축된 텍스처는 다른 포맷으로 트랜스코딩될 수 있다. 하드웨어 내부 JPEG 포맷은 압축 해제 프로세스를 용이하게 하는 추가적인 정보를 포함할 수 있다. 예를 들어, 하드웨어 내부 JPEG 포맷은 JPEG-압축된 텍스처의 여러 블록의 위치 및 사이즈에 대한 정보를 구비하는 테이블을 포함할 수 있다. 테이블은 또한 JPEG-압축된 텍스처의 각 8×8 블록의 DC 계수에 대한 정보를 더 포함할 수 있다. 테이블은 압축된 텍스처의 압축 해제를 용이하게 하기 위해 허프만(Huffman) 코드, 양자화(quantization) 테이블, 및 다른 정보를 더 포함할 수 있다. 드라이버(112)는 또한 컴퓨팅 시스템(102)에 의해 사용된 압축된 텍스처 각각에 대해 가상 어드레스 공간을 할당할 수 있다. 각 가상 어드레스 공간의 사이즈는 비압축된 텍스처의 사이즈에 대응할 수 있다.

[0024] 컴퓨팅 시스템(102)은 일반적으로 CPU, 버스, 메모리, 주변 디바이스 등과 같은 도 1에 도시되지 않은 여러 다른 디바이스/구성요소를 구비할 수 있다. 예를 들어, 컴퓨팅 시스템(102)은 디스플레이 디바이스(114)에 더하여 다른 디바이스, 예를 들어, 키보드, 프린터, 및 마우스에 연결될 수 있는 I/O 인터페이스를 포함할 수 있다. 일부 실시예에서, 컴퓨팅 시스템(102)은 복수의 GPU를 포함할 수 있다.

[0025] 또 다른 실시예에서, GPU(104)와 같은 프로세서는 소프트웨어에 한정될 수 있다. 소프트웨어 명령은 컴퓨터 판독가능한 저장 매체에 저장될 수 있고, 컴퓨팅 디바이스에서 실행될 때, 프로세서를 한정할 수 있다. 다른 실시예에서, 프로세서는 GPU, CPU, 비디오 처리 유닛(video processing unit: VPU), 코프로세서, 및/또는 텍스처 데이터를 처리하도록 구성된 다른 유형의 프로세서를 포함할 수 있다. 여러 실시예에서, GPU 및 CPU는 별개의 집적 회로 디바이스/패키지일 수 있다. 여러 실시예에서, GPU 및 CPU는 단일 집적 회로 또는 패키지에 포함될 수 있다.

[0026] 도 2를 참조하면, GPU(200)의 일 실시예의 블록도가 도시된다. GPU(200)는 그래픽-처리와 관련된 태스크(예를 들어, 정점 셰이더, 기하학적 셰이더, 픽셀 셰이더 등을 사용하여) 및 일반 컴퓨팅 태스크(예를 들어, 수리적 알고리즘, 물리적 시뮬레이션 등)를 수행하도록 사용될 수 있다. 도시된 예에서, GPU(200)는 셰이더 프로세서 어레이(210), 커맨드 프로세서(212), 텍스처 메모리(220), 직접 메모리 액세스(DMA)를 지원하도록 구성될 수 있는 메모리 제어기(222)를 포함한다. 도 2에 도시된 GPU(200)의 실시예는 단지 예시를 위한 것일 뿐, 이 기술 분야에 통상의 지식을 가진 자라면 많은 다른 실시예들이 가능하다는 것을 인식할 수 있는 것으로 이해된다. 모든 이러한 다른 실시예들이 고려된다. 또한 GPU(200)는 도 2에 도시되지 않은 많은 다른 구성요소를 포함할 수 있는 것으로 이해된다.

[0027] 도시된 실시예에서, 셰이더 프로세서 어레이(210)는 병렬 수행할 수 있는 다수의 처리 유닛을 포함한다. 커맨드 프로세서(212)는 커맨드를 발행하고 처리 태스크를 셰이더 프로세서 어레이(210)의 개별 셰이더 프로세서에 할당할 수 있다. 일부 실시예에서, 커맨드 프로세서(212)는 수신된 작업부하(workload)를 스레드(thread)로 분할하고 셰이더 프로세서 어레이의 처리 유닛 중에 스레드를 분배하도록 구성된 디스패치(dispatch) 프로세서(미도시)를 포함할 수 있다. 셰이더 프로세서 어레이(210)는 처리 텍스처 데이터를 포함하는 여러 유형의 기능을 수행하고 3차원 텍스처 객체를 2차원 이미지로 변환하는 렌더링 알고리즘을 수행하도록 구성될 수 있다. 전술된 바와 같이, 셰이더 프로세서 어레이(210)는 복수의 셰이더 프로세서를 포함할 수 있고, 복수의 셰이더 프로세서는 정점 및 다른 텍스처 데이터에 대한 넓은 범위의 수리적 및 논리적 동작을 사용하는 알고리즘을 구현할 수 있다.

[0028] 일부 실시예에서, GPU(200)는 데이터를 일시적으로 저장하는 하나 이상의 온-칩 및/또는 오프 칩 메모리를 사용하도록 구성될 수 있다. 이러한 메모리는 본 명세서에서 "캐시"라고 지칭될 수 있으나, 이러한 용어를 사용하는 것이 이러한 메모리에 임의의 특정 구성, 구조, 또는 정책을 반드시 요구하는 것은 아닌 것으로 이해된다. 예를 들어, 이러한 메모리는 세트 연관 구성 및 대체 정책과 같은 중앙 처리 유닛(CPU) 캐시와 일반적으로 연관된 구성 및 정책을 사용할 수 있으나, 임의의 원하는 구성 및/또는 저장 정책이 사용될 수 있다. 여러 실시예에서, 텍스처 메모리(220)는 텍스처 데이터를 저장하는데 사용된다. 이러한 실시예에서, 텍스처 메모리(220)는 텍스처 데이터가 시스템 메모리(226) 또는 국부 메모리(230)에만 저장된 경우 가능할 수 있었던 것보다 자주 사용되는 텍스처 데이터와 같은 특정 텍스처 데이터에 더 빠른 액세스를 제공할 수 있다. 시스템 메모리(226)는 GPU(200) 및 중앙 처리 유닛(CPU, 미도시)에 의해 모두 액세스가능한 메모리를 나타낼 수 있으나, 국부 메모리는 GPU(200)에 의해서만 직접 액세스가능한 메모리를 나타낼 수 있다. 여러 실시예에서, 텍스처 메모리(220)는 캐

시 기술 분야에 일반적으로 알려진 계층 배열로 된 다수의 레벨을 포함할 수 있다. 텍스처 캐시 시스템(220)에 포함된 이러한 캐시 레벨의 수는 실시예마다 변할 수 있다. 텍스처 메모리(220)는 다양한 메모리 기술, 예를 들어 정적 메모리(예를 들어, SRAM), 동적 메모리(예를 들어, DRAM)를 사용하는 적층된 메모리, 또는 다른 것을 사용하여 구현될 수 있다. 텍스처 메모리(220)는 또한 캐싱 논리(caching logic)를 포함할 수 있다. 캐싱 논리는 데이터를 텍스처 메모리(220)에 캐싱하고 캐시 시스템(220) 대(versus) 시스템 메모리(226)의 상대적인 레이턴시(latency) 및/또는 대역폭을 고려하는 캐시 관리 정책을 구현하도록 구성될 수 있다.

[0029] GPU(200)는 또한 메모리 제어기(222)를 포함할 수 있다. 메모리 제어기(222)는 시스템 메모리(226) 및 국부 메모리(230)에 연결될 수 있다. 메모리 제어기(222)는 시스템 메모리(226) 내 압축된 텍스처(228)와 같은 데이터에 액세스할 수 있다. 압축된 텍스처(228)는 JPEG과 같은 여러 가변 압축 기술 중 어느 것을 사용하여 압축될 수 있는 복수의 텍스처를 포함할 수 있다. 압축된 텍스처(228), 또는 압축된 텍스처(228) 내 개별 텍스처의 일부분은 제일 먼저 압축 해제됨이 없이(메모리 제어기(222)를 통해) GPU(200)의 텍스처 메모리(220) 및 셰이더 프로세서 어레이(210)로 전달될 수 있다. 호스트 드라이버(240)는 커맨드 및 데이터를 시스템 메모리(226)를 통해 GPU(200)로 전달할 수 있다. 국부 메모리(230)는 GPU(200)에 의해 사용되는 정점 데이터 및 다른 데이터를 저장하는데 사용될 수 있고, GPU(200)는 프레임 데이터를 국부 메모리(230)에 기록할 수 있다.

[0030] 이제 도 3을 참조하면, 그래픽 처리 시스템의 일 실시예의 블록도가 도시된다. 그래픽 처리 시스템(300)은 셰이더 제어기(310)를 포함할 수 있고, 셰이더 제어기(310)는 특정 그래픽 처리 태스크를 셰이더 어레이(320) 내 개별 셰이더 컴퓨팅 유닛에 할당할 수 있다. 셰이더 제어기(310)는 그래픽-처리 태스크 및 일반 컴퓨팅 태스크에 전 처리를 수행하고, 이 태스크를 셰이더 어레이(320)에 발행할 수 있다. 셰이더 제어기(310)는 셰이더 어레이의 처리 요소 중 새로운 작업부하를 처리하는데 이용가능한 것을 식별할 수 있고, 셰이더 제어기(310)는 새로운 작업부하를 셰이더 어레이(320) 중 이용가능한 처리 요소에 송신할 수 있다. 셰이더 제어기(310)는 작업부하 중 셰이더 어레이의 상이한 처리 요소에 의해 처리되는 작업 부하를 추적하여, 복수의 스레드가 병렬로 실행될 수 있게 한다.

[0031] 셰이더 어레이(320)는 셰이더 어레이(320)에 포함될 수 있는 임의의 개수 및 유형의 셰이더 프로세서를 나타내는 텍스처 소비 셰이더(321) 및 압축 해제 셰이더(322)를 포함할 수 있다. 여러 실시예에서, 셰이더 어레이(320)는 텍스처 데이터를 절차적으로 생성하도록 구성될 수 있는 추가적인 셰이더 프로세서를 포함할 수 있다. 일반적으로 말하면, 절차적인 텍스처 생성은 텍스처를 알고리즘적으로 생성하는 공정을 말한다. 여러 실시예에서 텍스처의 이 절차적인 생성은 미리(in advance)가 아니라 동적으로 수행된다. 셰이더 어레이(320)는 다른 태스크 중에서 텍스처를 맵핑하고 디스플레이 디바이스를 위한 이미지 데이터를 생성하는데 사용될 수 있다. 이들 동작을 수행하는 일부로서, 텍스처 소비 셰이더(321)는 텍스처 요청을 텍스처 필터(330)에 발행할 수 있다. 텍스처 요청은 텍스처의 하나 이상의 부분(예를 들어, 블록, 텍셀)에 대한 것일 수 있다. 텍스처 필터(330)는 요청된 텍스처에 대해 가상 어드레스를 생성하고, 이 가상 어드레스를 이 요청과 함께 캐시(340)로 운반할 수 있다. 캐시(340)는 픽셀과 연관된 텍셀 데이터의 형태로 텍스처를 저장할 수 있다. 텍스처의 일부는 압축될 수 있고, 이 텍스처의 일부는 비압축될 수 있다.

[0032] 텍스처 필터(330)로부터 가상 어드레스를 수신한 후, 캐시(340)는 요청된 텍스처가 캐시(340)에 저장되어 있는지 여부를 결정하기 위해 모든 알려진 가상 어드레스 범위에 대해 어드레스 체크를 수행할 수 있다. 요청된 텍스처의 비압축된 버전이 캐시(340)에 저장되어 있다면, 캐시(340)는 텍스처의 비압축된 버전을 텍스처 필터(330)로 리턴할 수 있다. 텍스처의 비압축된 버전이 캐시(340)에 저장되어 있지 않다면, 시도된 요청은 캐시 미스(cache miss)를 초래할 수 있다. 캐시 미스에 응답하여, 압축 해제 셰이더(322)는 텍스처의 압축된 버전을 압축 해제하기 위하여 개시될 수 있다. 여러 실시예에서, 셰이더 어레이(320)는 캐시(340)로부터 요청을 수신하거나, 또는 다른 방식으로, 압축 해제 셰이더를 개시하기 위하여 요청을 수신할 수 있다. 또한 캐시 미스에 응답하여, 텍스처 소비 셰이더(321)는 텍스처의 가상 어드레스를 압축 해제 셰이더(322)로 전달할 수 있다. 압축 해제 셰이더 프로그램을 위한 자원은 압축 해제 셰이더(322)에 미리 할당되어 셰이더 시작 레이턴시를 감소시키고 자원 관리를 간략화할 수 있다. 이 요청은 요청되는 블록의 가상 어드레스에 기초하여 셰이더 어레이(320)의 특정 셰이더 프로세서로 라우팅될 수 있다.

[0033] 캐시(340)는 텍스처의 압축된 버전에 대해 질의(queried)될 수 있고, 텍스처의 압축된 버전이 캐시(340)에 저장되어 있다면, 텍스처의 압축된 버전이 압축 해제 셰이더(322)로 리턴될 수 있다. 텍스처의 압축된 버전이 캐시(340)에 저장되어 있지 않다면, 텍스처의 압축된 버전이 시스템 메모리 또는 다른 위치로부터 검색될 수 있다. 압축 해제 셰이더(322)는 또한 압축 해제 동작을 가능하게 하기 위해 추가적인 테이블, 텍스처 및/또는 상수(constant)를 수신할 수 있다. 압축 해제 셰이더(322)는 요청된 텍스처를 압축 해제하는데 필요한 일부 추가적

인 압축된 데이터를 압축 해제할 수 있다. JPEG-압축된 텍스처의 경우에, 텍스처는 원래의 코드로부터 새로운 인코딩 방식으로 트랜스코딩될 수 있고, 새로운 인코딩 방식은 압축 해제를 보다 효율적으로 하기 위해 설계될 수 있다. 압축 해제 셰이더(322)가 텍스처의 압축된 버전을 수신하여 압축 해제된 후에, 텍스처 소비 셰이더(321)는 적절한 렌더링 계산을 위해 텍스처의 압축 해제된 버전을 사용할 수 있다. 이 프로세스는 복수의 텍스처 및/또는 텍스처의 일부분에 대해 계속될 수 있다. 또 다른 실시예에서, 텍스처 필터(330)에 의해 수행되는 것으로 설명된 기능은 셰이더 어레이(320)에 의해 수행될 수 있고, 셰이더 어레이(320)는 캐시(340)에 직접 연결될 수 있다.

[0034] 캐시(340)는 캐시(340)에 저장된 텍스처의 압축된 버전에 대해 소정의 가상 어드레스가 맵핑된 어드레스를 결정하는 테이블을 사용할 수 있다. 여러 실시예에서, 테이블(또는 그 일부분)은 캐시(340)에 또는 다른 곳에 저장될 수 있다. 일 실시예에서, 테이블은 가상 어드레스를 텍스처의 압축된 버전의 다른 어드레스로 맵핑할 수 있다. 가상 어드레스가 맵핑되는 어드레스는 그 자체가 가상 어드레스일 수도 있고 아닐 수도 있다. 사용되는 어드레스 방식의 유형에 대한 수많은 옵션은 가능하고 이에 고려된다. 테이블은 텍스처의 압축된 버전의 각 블록에 대해 오프셋(offset)을 저장할 수 있고, 여기서 오프셋은 텍스처의 압축된 버전의 시작부로부터 블록으로의 위치를 제공한다. 여러 실시예에서, 테이블은 하나 이상의 압축된 텍스처의 블록에 랜덤 액세스를 용이하게 한다. 캐시(340)의 캐시 논리는 블록의 압축된 버전에 대한 요청에 응답하여 소정의 블록의 어드레스를 결정할 수 있다. 캐시 논리는 이 테이블을 사용하여 원하는 블록이 캐시의 페이지 또는 페치 유닛(fetch unit) 내에 저장되어 있는 오프셋을 결정할 수 있다. 셰이더 어레이(320) 중 복수의 셰이더는 또한 이 테이블을 사용하여 텍스처의 요청된 블록의 오프셋을 결정할 수 있다. 여러 실시예에서, 캐시(340)는 복수의 텍스처에 대한 맵핑 정보를 구비하는 복수의 테이블을 사용할 수 있다.

[0035] 텍스처 데이터가 처리된 후에, 셰이더 어레이(320)는 이미지 데이터를 렌더링 유닛(350)으로 운반할 수 있다. 렌더링 유닛(350)은 이미지 프레임의 각 픽셀에 대해 고유한 컬러 속성을 한정하는 특정 수의 값을 할당할 수 있다. 이 수 값은 프레임 버퍼(360)로 전달될 수 있고 여기에 수 값이 저장되어 디스플레이 디바이스(370)에 렌더링될 때와 같이 적절한 시간에 사용할 수 있다.

[0036] 후속 동작에서, 텍스처 소비 셰이더(321)는 압축 해제 셰이더의 기능을 수행하도록 구성될 수 있고, 압축 해제 셰이더(322)는 텍스처 소비 셰이더의 기능을 수행하도록 구성될 수 있다. 셰이더 어레이(320) 중 각 셰이더 프로세서는 현재 동작의 요구조건에 따라 다양한 기능을 수행하도록 구성될 수 있다.

[0037] 여러 실시예에서, 부하 밸런싱(balancing)을 사용하여 압축 해제 태스크를 덜 이용된(underutilized) 셰이더에 할당할 수 있다. 또한, 일부 공간은 다수의 컴퓨트 유닛에 예비되어 압축 해제 셰이더가 다수의 컴퓨트 유닛에서 론칭(launched)될 수 있게 한다. 나아가, 다수의 압축 해제 요청은 단일 명령 다중 데이터(SIMD) 벡터로 패킹(packed)될 수 있다. SIMD 벡터는 일 벡터에서 다수의 블록의 압축 해제를 용이하게 할 수 있다. 일 실시예에서, 16개의 블록은 4개의 레인(lane)마다 하나의 블록을 가지는 일 벡터에서 압축 해제될 수 있다.

[0038] 여러 실시예에서, 그래픽 처리 시스템(300)은 텍스처 데이터의 진행 중에 절차적인 생성을 가능할 수 있다. 하나의 셰이더는 진행 중에 텍스처 데이터를 생성할 수 있고, 제2 셰이더는 렌더링 동작을 위해 생성된 텍스처 데이터를 사용할 수 있다. 압축 해제 셰이더는 압축된 데이터에 액세스할 수 있고, 또 다른 셰이더는 하나 이상의 테이블과 같은 추가적인 데이터를 압축 해제하는데 사용될 수 있다. 압축된 데이터의 일부는 다양한 압축 기술을 사용하여 압축될 수 있다. 여러 실시예에서, 압축 해제 셰이더는 캐시로부터 데이터를 요청할 수 있고, 캐시 미스에 응답하여, 다른 셰이더는 텍스처 데이터를 절차적으로 생성하도록 개시될 수 있다.

[0039] 이제 도 4a를 참조하면, 데이터 캐시의 일 실시예의 블록도가 도시된다. 캐시(410)는 캐시(410)에 저장될 수 있는 텍스처의 임의의 개수의 부분을 나타내는 텍스처(420 및 430)의 일부를 포함할 수 있다. 텍스처(420 및 430)는 압축된 텍스처일 수 있는 반면, 캐시(410)에 저장된 복수의 텍스처는 압축된 텍스처 및 비압축된 텍스처의 혼합일 수 있다. 텍스처(420)는 텍스처(420)의 임의의 개수의 블록을 나타내는 블록(422 및 423)을 포함할 수 있다. 텍스처(420)는 또한 텍스처(420)의 가상 어드레스 공간을 압축된 텍스처(420)의 어드레스 공간으로 맵핑할 수 있는 테이블(421)을 포함할 수 있다. 텍스처(430)는 텍스처(420)와 유사하게 구성될 수 있다. 또 다른 실시예에서, 테이블(421)은 텍스처(420)로부터 별개로 저장될 수 있다.

[0040] 텍스처 소비 셰이더가 캐시(410)로부터 텍스처의 블록을 요청하고 이 요청이 캐시 미스를 초래할 때, 캐시(410)는 요청된 블록에 대해 캐시 라인(440)을 할당할 수 있다. 캐시(410)는 할당된 캐시 라인의 어드레스를 압축 해제 셰이더에 운반할 수 있다. 압축 해제 셰이더가 요청된 블록에 대응하는 압축된 블록을 압축 해제한 후에, 압축 해제 셰이더는 압축 해제된 블록을 캐시 라인(440)에 기록하도록 구성될 수 있다. 대안적으로, 압축 해제

세이더는 압축 해제된 블록을 캐시(410) 내 여러 위치에 기록할 수 있다. 압축 해제 세이더가 압축 해제된 블록을 캐시 라인(440)에 기록한 것에 응답하여, 텍스트 소비 세이더는 캐시(410)로부터 압축 해제된 블록을 페치하도록 구성될 수 있다. 대응하는 레이턴시 보상 큐(compensation queue)는 압축된 블록의 진행 중에 압축 해제로부터 초래되는 더 큰 레이턴시를 수용하도록 연장될 필요가 있을 수 있다.

[0041] 블록의 압축 해제된 버전이 캐시 라인(cache line)(440)에 기록된 후에, 캐시(410)는 블록의 압축된 버전 및 블록의 압축 해제된 버전을 저장할 수 있다. 여러 실시예에서, 캐시(410)는 두 버전이 캐시(410)에 저장되어 있는 것으로 결정난 것에 응답하여 블록의 버전들 중 하나를 폐기하는 보유 정책(retention policy)을 실행할 수 있다. 일 실시예에서, 블록의 압축 해제된 버전은 텍스트 소비 세이더에 의해 페치(fetched)된 후에 폐기될 수 있다. 또 다른 실시예에서, 블록의 압축된 버전은 블록의 압축 해제된 버전이 캐시(410)에 기록된 후에 폐기될 수 있다. 다른 실시예에서, 블록의 압축된 버전 및 압축 해제된 버전은 모두 연장된 시간 기간 동안 캐시(410)에 유지될 수 있다.

[0042] 텍스트의 블록의 비압축된 버전에 대한 요청에 응답하여, 캐시(410)는 비압축된 버전이 캐시(410)에 저장되어 있지 않은 것으로 결정할 수 있다. 여러 실시예에서, 이러한 결정에 응답하여, 캐시(410)는 자동으로 블록의 압축된 버전을 자동으로 검색할 수 있다. 블록의 압축된 버전이 캐시(410)에 저장되어 있다면, 캐시(410)는 세이더 또는 다른 처리 유닛을 통지하거나 및/또는 캐시(410)는 블록의 압축된 버전을 세이더 또는 다른 처리 유닛으로 운반할 수 있다.

[0043] 일부 실시예에서, 비압축된 블록에 대한 요청이 캐시 미스인 것에 응답하여, 별개의 소프트웨어 스레드가 시작될 수 있고, 스레드는 압축 해제 세이더를 개시할 수 있다. 텍스트 소비 세이더는 블록의 가상 어드레스를 압축 해제 세이더로 운반할 수 있다. 여러 실시예에서, 세이더가 압축 해제 태스크를 종료할 때, 압축 해제 세이더는 비압축된 블록(들)을 캐시로 운반할 수 있다. 다른 실시예에서, 압축 해제 세이더가 압축 해제 동작을 종료할 때, 압축 해제 세이더는 세이더 출력을 텍스트 소비 세이더로 운반할 수 있다.

[0044] 이제 도 4b를 참조하면, 블록 맵핑 테이블의 일 실시예의 블록도가 도시된다. 테이블(421)은 텍스트(420)(도 4a)의 복수의 블록에 대한 맵핑 정보를 저장할 수 있다. 여러 실시예에서, 테이블(421)은 도 4b에 도시된 것에 더하여 다른 유형의 정보를 구비하게 여러 방식으로 구성될 수 있다. 예를 들어, 일 실시예에서, 테이블(421)은 텍스트(420)의 각 블록에 대한 DC 계수값을 포함할 수 있다.

[0045] 테이블(421)은 텍스트(420)의 가상 어드레스 공간을 압축된 텍스트(420)(도 4a)의 물리적 어드레스 공간으로 맵핑할 수 있다. 압축 해제 세이더(미도시)는 캐시(410)로부터 텍스트(420)의 하나 이상의 블록을 페치하거나 다른 방식으로 수신할 수 있고, 압축 해제 세이더는 테이블(421)로부터 압축된 블록의 위치 및 사이즈를 결정할 수 있다. 압축된 블록의 사이즈는 2개의 인접한 블록의 시작 물리적 어드레스들 사이에 차이를 계산하는 것에 의해 결정될 수 있다. 다른 실시예에서, 추가적인 데이터는 블록에 대한 사이즈 및/또는 위치 정보를 나타내도록 제공될 수 있다. 나아가, 압축 해제 세이더는 각 블록의 DC 계수값과 같은 테이블(421)로부터 추가적인 정보를 획득할 수 있다.

[0046] 일부 실시예에서, 텍스트는 슈퍼블록(superblock)에 따라 구성될 수 있다. 슈퍼블록은 총 1024개의 픽셀에 대해 32×32 픽셀의 타일인 16개의 8×8 블록의 세트일 수 있다. 텍스트에 대한 인덱스 테이블(index table)은 각 슈퍼블록에 대한 테이블 엔트리(entry)를 포함할 수 있고, 각 테이블 엔트리는 각 슈퍼블록에 시작 어드레스를 제공할 수 있다. 일 실시예에서, 이 어드레스는 텍스트 내 슈퍼블록의 위치일 수 있다. 또 다른 실시예에서, 이 어드레스는 텍스트의 시작으로부터 오프셋일 수 있다. 각 엔트리는 또한 슈퍼블록에 속하는 제1 8×8 블록의 4-비트 인덱스를 포함할 수 있다. 일부 실시예에서, 슈퍼블록은 캐시의 2 킬로비트(Kb) 경계와 정렬되어 있지 않을 수 있다. 각 엔트리는 또한 16-비트 마스크를 포함할 수 있다. 16-비트 마스크는 이 블록이 그 다음 2 Kb 워드에서 시작하는지 여부를 나타내는 블록당 일 비트를 포함할 수 있다.

[0047] 일부 실시예에서, 압축 해제 세이더는 8×8 블록의 가상 어드레스를 32×32 슈퍼블록의 가상 어드레스로 변환하여 룩업 목적을 위하여 인덱스 테이블의 엔트리 수를 계산할 수 있다. 압축 해제 세이더는 슈퍼블록에 대응하는 인덱스 테이블의 엔트리를 룩업할 수 있다. 인덱스 테이블은 다른 텍스트와 유사한 방식으로 세이더에 의해 처리될 수 있다. 인덱스 테이블의 엔트리는 캐싱되고 처리될 수 있다.

[0048] 각 인덱스 테이블 엔트리로부터 세이더는 가상 어드레스일 수 있는 베이스 어드레스를 획득할 수 있다. 베이스 어드레스는 압축된 슈퍼블록의 제1 페치 유닛으로 구성될 수 있다. 세이더는 또한 압축 해제될 필요가 있는 요청된 블록을 포함하는 페치 유닛의 오프셋을 획득할 수 있다. 세이더는 또한 블록이 압축되어 있거나 블록의 어

드레스에 기초하지 않은 경우 계산될 수 있다. 특정 어드레스 범위는 비압축된 블록의 가상 어드레스에 대응할 수 있고, 다른 어드레스 범위는 압축된 블록의 물리적 어드레스에 대응할 수 있다. 셰이더는 상이한 어드레스 범위들 사이를 구별할 수 있다.

[0049] 이제 도 5를 참조하면, 텍셀의 8×8 블록에 대한 가상 어드레스 공간의 일 실시예의 블록도가 도시된다. 각 텍셀은 가상 어드레스 공간(570) 내 고유한 어드레스로 맵핑될 수 있다. 텍셀 1은 어드레스(501)로 맵핑될 수 있고, 텍셀 2는 어드레스(502)로 맵핑될 수 있고, 8×8 블록(500)의 모든 64개의 텍셀에 대해 이와 같은 방식으로 맵핑될 수 있다. 블록(500)은 압축된 텍스처 내 블록일 수 있고, 가상 어드레스 공간(570)은 압축된 텍스처의 블록(500)에 대해 할당될 수 있다. 텍스처는 블록(500)에 더하여 복수의 블록을 포함할 수 있다. 가상 어드레스 공간(570)은 또한 텍스처 내 복수의 블록의 각 텍셀에 대해 고유한 어드레스를 포함할 수 있다.

[0050] 예시를 위하여, 비압축된 텍셀은 32-비트값(8-비트값의 4개의 세트)인 것으로 가정된다. 다른 사이즈의 비압축된 텍셀이 또한 본 명세서에 설명된 방법 및 메커니즘에 사용될 수 있다. 예를 들어, 24-비트값을 가지는 비압축된 텍셀은 유사한 방식으로 처리될 수 있다. 여러 실시예에서, 텍스처 소비 셰이더는 개별 텍셀에 대한 요청을 생성할 수 있다. 제일 먼저, 셰이더는 텍셀의 가상 어드레스를 컴퓨팅할 수 있다. 이후, 캐시는 텍셀에 대응하는 가상 어드레스에 대해 질의될 수 있다.

[0051] 이제 도 6을 참조하면, 압축된 데이터의 일 실시예의 블록도가 도시된다. 데이터 부분(605)은 압축된 데이터의 페치의 유닛일 수 있고, 데이터 부분(605)의 사이즈는 비압축된 블록의 사이즈에 기초할 수 있다. 일 실시예에서, 페치 유닛은 사이즈 2 Kb 사이즈로 구성될 수 있다. 다른 실시예에서, 페치 유닛은 여러 사이즈 중 어느 것일 수 있다. 복수의 압축된 블록은 페치 유닛으로 패킹될 수 있다. 일 실시예에서, 페치 유닛으로 패킹될 수 있는 블록의 최대 개수는 16인 것으로 가정될 수 있다. 다른 실시예에서, 다른 개수의 블록이 페치 유닛으로 패킹될 수 있다. 하나의 유형의 캐시 액세스 방식에서, 블록의 데이터는 페치 유닛의 경계를 횡단(cross)하지 않는 것으로 가정될 수 있다.

[0052] 블록은 JPEG과 같은 압축 포맷의 최소 디코딩가능한 유닛일 수 있다. JPEG에서, 블록은 (64개의 픽셀을 구비하는) 8×8 픽셀 타일이다. 텍스처가 압축되어 있고, 셰이더에 의해 요청된 텍스처의 블록은 압축 해제될 필요가 있을 때, 캐시 라인은 블록에 대해 캐시에 할당될 수 있다. 일 실시예에서, 캐시 라인 사이즈는 2 Kb이어서 전체 비압축된 블록($32 \text{ 비트} * 64 = 2 \text{ Kb}$)을 저장할 수 있다. 다른 실시예에서, 캐시 라인 사이즈는 여러 사이즈 중 어느 것일 수 있다.

[0053] 페치 유닛이 비압축된 블록을 포함하는 경우, 단 하나의 블록만이 페치 유닛에 맞을 수 있다. 압축된 블록을 포함하는 페치 유닛에서, 페치 유닛은 또한 176-비트 헤더를 포함할 수 있다. 페치 유닛은 16 블록의 용량을 구비하는 것으로 가정될 수 있다. 헤더는 페치 유닛 내 압축된 블록의 위치를 나타내도록 16개의 11-비트 오프셋값을 포함할 수 있다. 오프셋은 블록의 시작 비트 위치를 참조한다. 다른 실시예에서, 헤더에는 가변 개수의 오프셋 지시자(indicator)들이 있을 수 있다.

[0054] 도 6에 도시된 바와 같이, 데이터 부분(605)은 헤더(610) 및 블록(611-626)을 포함할 수 있다. 블록(611-626)은 압축된 텍스처의 16개의 상이한 블록일 수 있다. 헤더(610)는 오프셋(631-646)을 포함할 수 있다. 각 오프셋은 데이터 부분(605) 내 대응하는 블록의 위치에 대응하는 11-비트 오프셋값일 수 있다. 다른 실시예에서, 다른 비트-사이즈의 오프셋값이 사용될 수 있다. 오프셋(631)은 블록(611)의 시작 어드레스를 나타낼 수 있고, 오프셋(632)은 블록(612)의 시작 어드레스를 나타낼 수 있고, 이와 같이 계속될 수 있다. 일부 실시예에서, 마지막 블록의 마지막 비트를 나타내는 추가적인 오프셋이 있을 수 있어서, 캐시로부터 불필요한 페치를 감소시킬 수 있다.

[0055] 일부 실시예에서, 텍스처의 압축된 8×8 블록은 패킹될 수 있고 페치 유닛 경계를 횡단할 수 있다. 블록이 2개의 페치 유닛을 사용하는 것을 보여주는 대응하는 정보는 인덱스 테이블에 저장될 수 있고, 압축 해제 셰이더는 페치 유닛 경계를 횡단하는 블록에 대해 하나의 페치 대신 2개의 페치를 생성할 수 있다.

[0056] 이제 도 7을 참조하면, 텍스처의 압축된 블록을 압축 해제하는 방법의 일 실시예가 도시된다. 설명의 목적을 위하여, 이 실시예에 있는 단계는 순차적으로 도시된다. 아래에 설명된 방법의 여러 실시예에서, 설명된 요소의 하나 이상이 동시에 수행될 수 있거나, 도시된 것과 상이한 순서로 수행될 수 있거나, 또는 전적으로 생략될 수 있는 것으로 이해된다. 다른 추가적인 요소는 또한 원하는대로 수행될 수 있다.

[0057] 본 방법(700)은 블록(705)에서 시작하고, 이후 블록(710)에서, 복수의 셰이더 중 제1 셰이더가 이미지에 대한 렌더링 동작의 일부로서 텍스처의 블록에 대한 필요성을 결정할 수 있다. 제1 셰이더는 텍스처 소비 셰이더일

수 있다. 다음으로, 제1 셰이더는 블록의 가상 어드레스를 계산할 수 있다(블록 715). 제1 셰이더는 비압축된 텍스처의 비압축된 버전에 대응하는 텍스처의 비압축된 뷰(view)를 가질 수 있고, 가상 어드레스는 비압축된 뷰 내에 요청된 블록의 위치에 대응할 수 있다. 블록(715) 후에, 제1 셰이더는 캐시로부터 블록을 요청하고 가상 어드레스를 요청과 함께 운반할 수 있다(블록 720). 다음으로, 캐시는 블록의 비압축된 버전이 캐시에 저장되어 있는지 여부를 결정할 수 있다(조건 블록 725). 블록의 비압축된 버전이 캐시에 저장되어 있는 경우, 제1 셰이더는 캐시로부터 블록의 비압축된 버전을 수신하여 블록을 처리할 수 있다(블록 770).

[0058] 블록의 비압축된 버전이 캐시에 저장되어 있지 않은 경우, 복수의 셰이더 중 제2 셰이더가 압축 해제 셰이더로 개시될 수 있다(블록 730). 압축 해제 셰이더를 위한 자원은 하나 이상의 셰이더 프로세서에 미리 할당되어 셰이더 시작 레이턴시를 감소시키고 자원 관리를 간략화할 수 있다. 또한, 요청된 블록의 가상 어드레스가 제1 셰이더로부터 제2 셰이더로 전달될 수 있다. 다음으로, 캐시 라인은 요청된 블록에 할당될 수 있다(블록 735). 이후, 캐시는 블록의 압축된 버전이 캐시에 저장되어 있는지 여부를 결정할 수 있다(조건 블록 740). 여러 실시예에서, 캐시는 제2 셰이더가 블록의 압축된 버전을 요청한 것에 응답하여 이 결정을 할 수 있다. 다른 실시예에서, 캐시는 블록의 비압축된 버전이 캐시에 저장되어 있지 않은 것으로 결정난 것에 응답하여 자동으로 이 결정을 할 수 있다(조건 블록 725).

[0059] 블록의 압축된 버전이 캐시에 저장되어 있는 경우(조건 블록 740), 캐시 및/또는 제2 셰이더는 테이블로부터 블록의 압축된 버전의 위치 및 사이즈를 결정할 수 있다(블록 750). 블록의 압축된 버전이 캐시에 저장되어 있지 않은 경우(조건 블록 740), 블록의 압축된 버전이 (예를 들어, 국부 메모리 또는 시스템 메모리로부터) 폐치되어 캐시에 저장될 수 있다(블록 745). 시스템 메모리로부터 블록의 압축된 버전을 폐치하는 것은 전체 압축된 텍스처 또는 이 텍스처의 일부 부분을 폐치하는 것을 수반할 수 있다. 캐시는 텍스처의 비압축된 버전의 가상 어드레스 공간을 텍스처의 압축된 버전의 어드레스 공간으로 맵핑하는 테이블을 사용하도록 구성될 수 있다. 캐시 및/또는 제2 셰이더는 테이블로부터 블록의 압축된 버전의 위치 및 사이즈를 결정할 수 있다(블록 750). 테이블은 또한 텍스처의 각 블록의 압축된 버전의 DC 계수값과 같은 추가적인 정보를 포함할 수 있다. 블록(750) 후에, 블록의 압축된 버전은 캐시로부터 제2 셰이더로 운반될 수 있다(블록 755).

[0060] 또 다른 실시예에서, 블록의 압축된 버전이 캐시에 저장되어 있지 않은 경우(조건 블록 740), 단계(745, 750, 및 755)는 대안 단계로 대체될 수 있다. 대안 단계에서, 블록의 압축된 버전은 시스템 메모리로부터 폐치될 수 있고 직접 제2 셰이더에 제공될 수 있다. 이들 대안 단계는 제2 셰이더가 캐시로부터 블록의 압축된 버전을 수신하는 것보다 더 효율적일 수 있다. 다른 실시예에서, 블록의 압축된 버전은 시스템 메모리로부터 폐치될 수 있고 또한 캐시에 기록되고 있는 동안 제 2 셰이더에 직접 제공될 수 있다.

[0061] 제2 셰이더가 블록의 압축된 버전을 수신한 후에(블록 755), 제2 셰이더는 블록의 압축된 버전을 압축 해제할 수 있다(블록 760). 다음으로, 제2 셰이더는 블록의 압축 해제된 버전을 캐시에 기록할 수 있다(블록 765). 이후, 제1 셰이더가 캐시로부터 블록의 압축 해제된 버전을 수신하고 현재 이미지에 대한 렌더링 동작의 일부로 블록을 처리할 수 있다(블록 770). 블록(770) 후에, 본 방법은 블록(775)에서 종료할 수 있다. 방법(700)은 복수의 텍스처로부터 복수의 블록에 대해 반복될 수 있다.

[0062] 특징 및 요소가 특정 조합으로 예시적인 실시예에 설명되었으나, 각 특징 또는 요소는 예시적인 실시예의 다른 특징 및 요소 없이 단독으로 사용되거나 또는 다른 특징 및 요소와 함께 또는 없이 여러 조합으로 사용될 수 있다. 본 발명은 임의의 비휘발성 메모리 디바이스에서 사용하거나 이 디바이스에 의해 사용하기 위한 기계, 프로세서, 및/또는 임의의 일반 목적 컴퓨터에 의해 실행하기 위한 기계 판독가능한 명령을 구비하는 비-일시적인 컴퓨터-판독가능한 저장 매체에 유형적으로 구현된 컴퓨터 프로그램 또는 펌웨어로 구현될 수 있다. 컴퓨터-판독가능한 저장 매체는 본 명세서에 설명된 기능, 방법, 및 동작을 가능하도록 동작가능한 프로그램 명령을 포함할 수 있다. 적절한 프로세서는 예로서 일반 목적 및 특수 목적 프로세서를 모두 포함한다.

[0063] 일반적으로, 프로세서는 저장된 소프트웨어 또는 펌웨어를 구비하는 판독 전용 메모리(ROM), RAM, 및/또는 저장 디바이스로부터 명령 및 데이터를 수신할 수 있다. 컴퓨터 프로그램 명령 및 데이터를 구현하기에 적절한 저장 디바이스는 예로서 반도체 메모리 디바이스, 판독 전용 메모리(ROM), 자기 매체, 예를 들어 내부 하드 디스크 및 이동식 디스크, 광자기 매체, 및 광 매체, 예를 들어 CD-ROM 디스크 및 디지털 다용도 디스크(DVD)를 포함하는 모든 형태의 비휘발성 메모리를 포함한다.

[0064] 전술된 실시예는 하드웨어 기술 언어(hardware description language: HDL), 예를 들어 베릴로그(Verilog) 또는 VHDL를 사용하여 소프트웨어로 설계될 수 있다. HDL-설계는 전자 시스템의 거동을 모델링할 수 있고, 이 설계는 합성되어 궁극적으로 하드웨어 디바이스로 제조될 수 있다. 게다가, HDL-설계는 컴퓨터 제품에 저장되어

하드웨어 제조 전에 컴퓨터 시스템에 로딩될 수 있다.

[0065] 본 발명에 의해 사용되거나 본 발명과 함께 사용될 수 있는 하드웨어 구성요소, 프로세서, 또는 기계의 유형으로는 응용 특정 집적 회로(ASIC), 전계 프로그래밍가능한 게이트 어레이(FPGA), 마이크로프로세서, 또는 임의의 집적 회로를 포함한다. 이러한 프로세서는 처리된 하드웨어 기술 언어(HDL) 명령(컴퓨터 판독가능한 매체에 저장될 수 있는 명령)의 결과를 사용하여 제조 프로세스를 구성하는 것에 의해 제조될 수 있다. 이러한 처리 결과는 본 명세서에 설명된 방법 및 메커니즘의 측면을 구현하는 프로세서를 제조하도록 반도체 제조 프로세스에 사용되는 마스크작업일 수 있다.

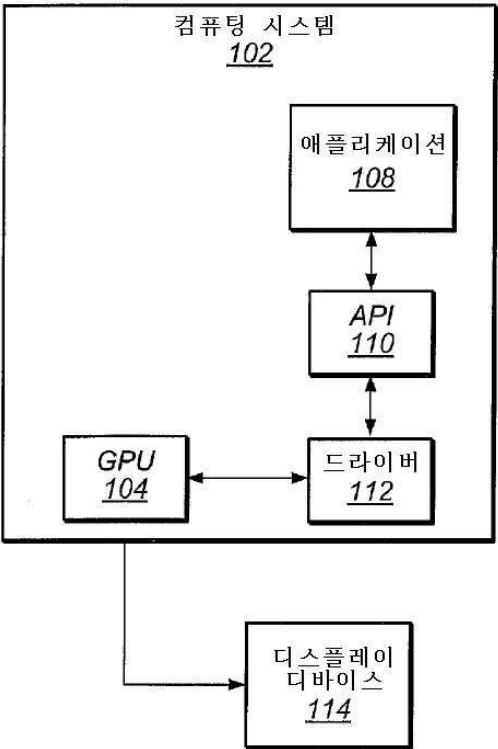
[0066] 이미지 렌더링 계산 및 셰이더 태스크를 구현하는데 사용되는 것과 같은 소프트웨어 명령은 컴퓨터-판독가능한 저장 매체에 저장될 수 있다. 컴퓨터-판독가능한 저장 매체는 기계(예를 들어, 컴퓨터)에 의해 판독가능한 형태(예를 들어, 소프트웨어, 처리 애플리케이션)로 정보를 저장하는 임의의 메커니즘을 포함할 수 있다. 컴퓨터-판독가능한 저장 매체는 자기 또는 광 매체(예를 들어, 디스크(고정식 또는 이동식), 테이프, CD-ROM, DVD-ROM, CD-R, CD-RW, DVD-R, DVD-RW, 또는 블루레이(Blu-Ray)), RAM(예를 들어, 동기 동적 RAM(SDRAM), 더블 데이터 레이트(double data rate)(DDR, DDR2, DDR3 등) SDRAM, 저전력 DDR(LPDDR2 등) SDRAM, 램버스 DRAM(RDRAM), 정적 RAM(SRAM)), ROM, USB 인터페이스, 마이크로-전자 기계 시스템(MEMS)과 같은 주변 인터페이스를 통해 액세스 가능한 비휘발성 메모리(예를 들어, 플래시 메모리), 및 네트워크 및/또는 무선 링크와 같은 통신 매체를 통해 액세스가능한 저장 매체를 포함할 수 있으나 이들로 제한되지 않는다.

[0067] 여러 예시적인 실시예가 도시되고 기술되었으나, 이 기술 분야에 통상의 지식을 가진 자에게는 설명된 방법에 다수의 변경, 변형 또는 대안이 이루어질 수 있다는 것은 자명하다. 그러므로, 이러한 변경, 변형 및 대안은 본 명세서에 설명된 방법 및 메커니즘의 범위 내에 있는 것으로 보아야 한다. 또한 기술된 실시예는 단지 본 발명의 구현을 예시하는 것으로 해석되어야 하는 것으로 이해되어야 한다.

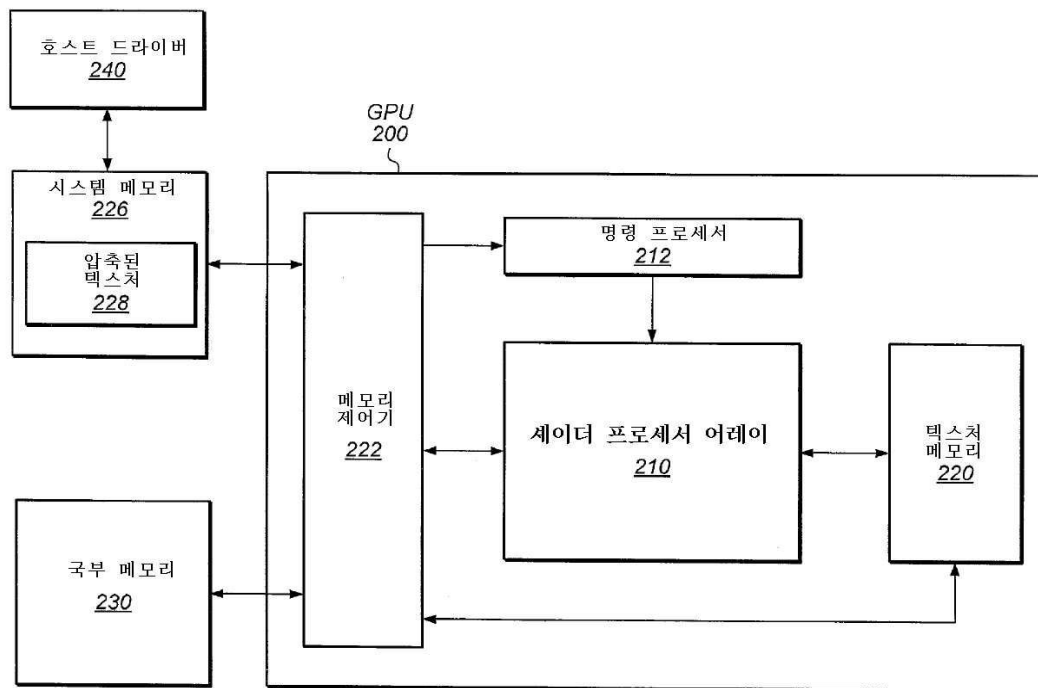
도면

도면1

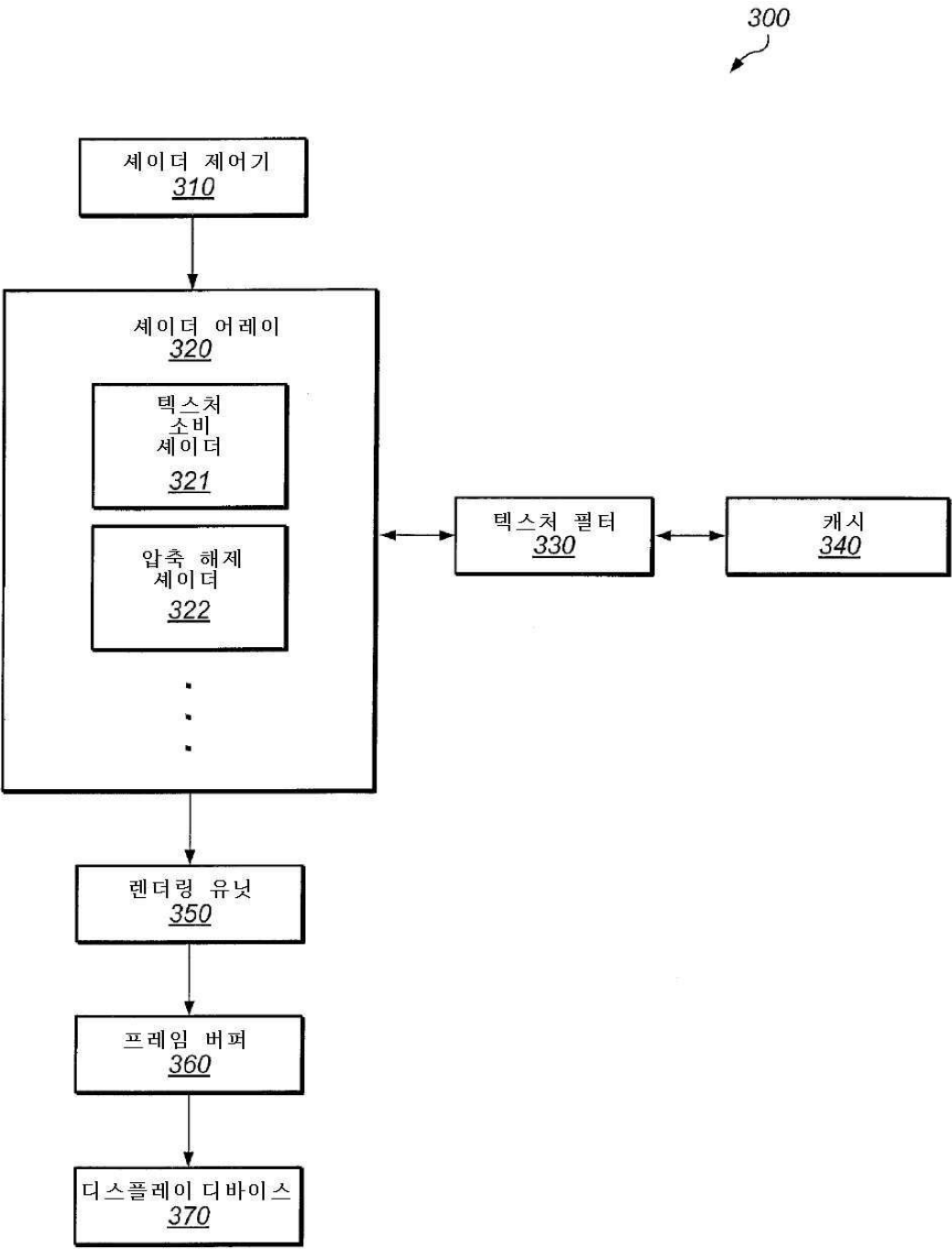
100



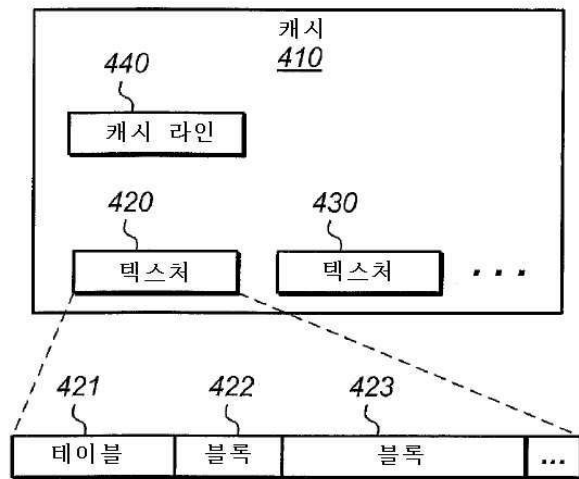
도면2



도면3



도면4a

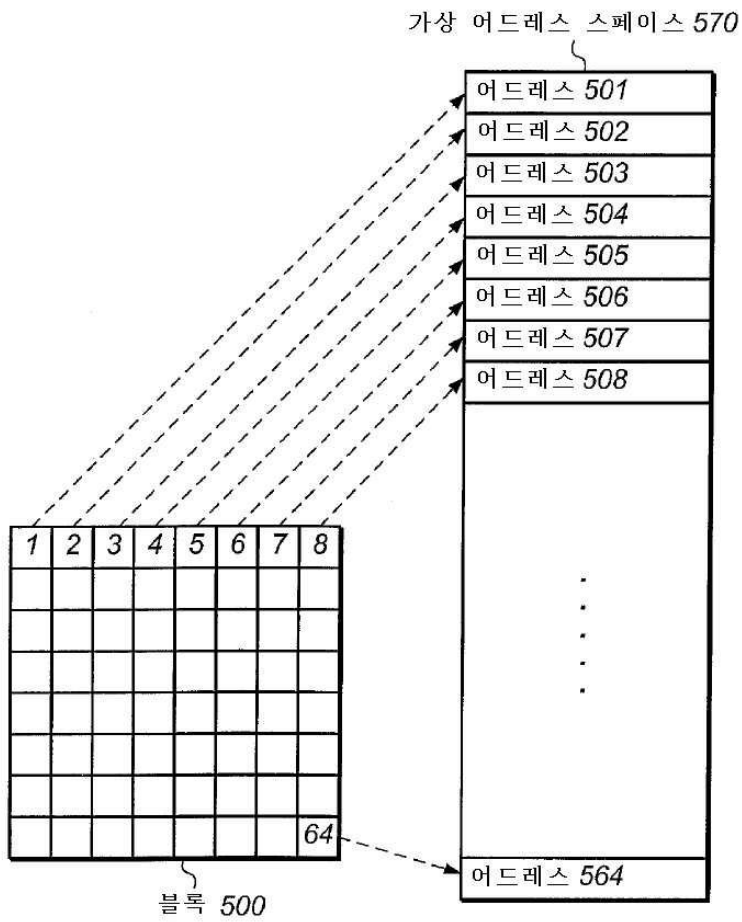


도면4b

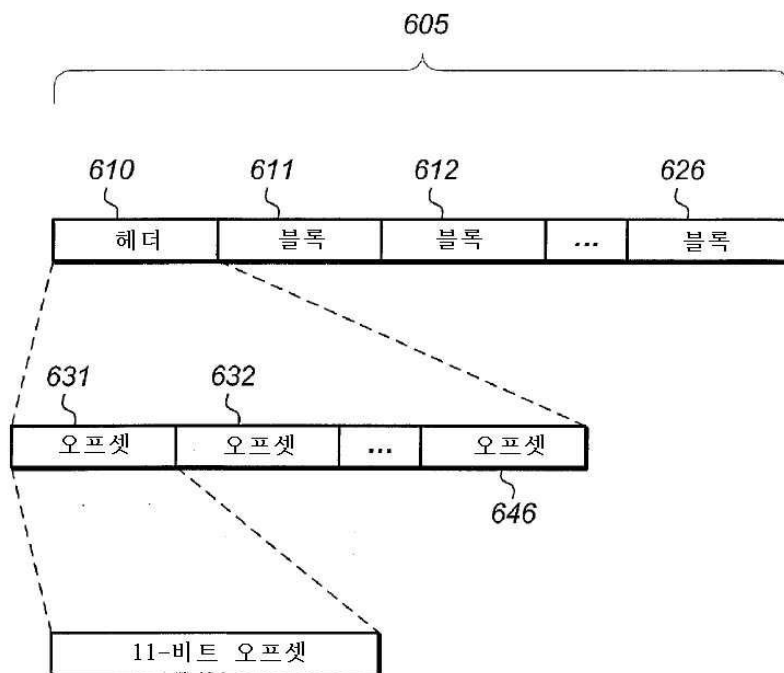
테이블
421

가상 어드레스	물리적 어드레스	블록
0x000	0x100	422
0x800	0x200	423
0x2000	0x800	
⋮	⋮	⋮

도면5



도면6



도면7

