



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2018년06월15일  
(11) 등록번호 10-1868200  
(24) 등록일자 2018년06월08일

(51) 국제특허분류(Int. Cl.)  
G06T 15/00 (2006.01)  
(21) 출원번호 10-2014-7016242  
(22) 출원일자(국제) 2012년10월16일  
심사청구일자 2014년06월13일  
(85) 번역문제출일자 2014년06월13일  
(65) 공개번호 10-2014-0098133  
(43) 공개일자 2014년08월07일  
(86) 국제출원번호 PCT/US2012/060368  
(87) 국제공개번호 WO 2013/074236  
국제공개일자 2013년05월23일  
(30) 우선권주장  
13/298,051 2011년11월16일 미국(US)  
(56) 선행기술조사문헌  
GB2480012 A\*  
Frederico Abraham, et al., 'A load-balancing  
strategy for sort-first distributed renderin  
g', IEEE, Proceedings of the XVII Brazilian  
Symposium on Computer Graphics and Image  
Processing, 2004.  
KR10001342797B1  
KR10001240815B1  
\*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
켈컴 인코퍼레이티드  
미국 92121-1714 캘리포니아주 샌 디에고 모어하  
우스 드라이브 5775  
(72) 발명자  
칼리오 끼아 까뽀  
미국 92121 캘리포니아주 샌디에고 모어하우스 드  
라이브 5775  
아르보 유까-빠까  
미국 92121 캘리포니아주 샌디에고 모어하우스 드  
라이브 5775  
(74) 대리인  
특허법인코리어나

전체 청구항 수 : 총 40 항

심사관 : 조우연

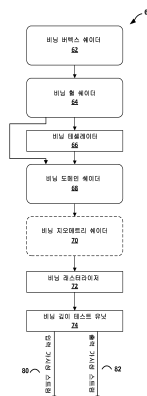
(54) 발명의 명칭 타일 기반 렌더링에서의 테셀레이션

(57) 요약

개시된 기술들은 프레임의 각각의 타일에 대한 입력 가시성 스트림을 생성하는 것, 및 프레임의 각각의 타일에 대한 출력 가시성 스트림을 생성하는 것을 포함하고, 입력 가시성 스트림은 렌더링시 각각의 타일에서 입력 프리미티브가 보이는지 아닌지의 여부를 나타내고, 출력 가시성 스트림은 렌더링시 출력 프리미티브가 각각의 타일에

(뒷면에 계속)

대표도 - 도4



서 보이는지 아닌지의 여부를 나타내며, 출력 프리미티브는 입력 프리미티브를 테셀레이팅하는 것에 의해 생성된다. 이러한 방식에서, 입력 가시성 스트림에 기초하여, 타일에서 보이지 않는 전체 입력 프리미티브에 대해 테셀레이션이 스킵될 수도 있다. 또한, 출력 가시성 스트림에 기초하여, 입력 프리미티브의 일부가 보이지 않더라도, 타일에서 보이지 않는 출력 프리미티브들 중 어떤 출력 프리미티브들에 대해 테셀레이션이 스킵될 수도 있다.

---

## 명세서

### 청구범위

#### 청구항 1

그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션 (tessellation) 을 위한 방법으로,

프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 입력 가시성 스트림을 생성하는 것으로서, 상기 입력 가시성 스트림은 입력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타내는, 상기 입력 가시성 스트림을 생성하는 것;

상기 프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 출력 가시성 스트림을 생성하는 것으로서, 상기 출력 가시성 스트림은 깊이 테스트에 기초하여 상기 타일에서 출력 프리미티브가 보이는지 아닌지의 여부의 표시를 포함하고, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되고, 상기 출력 프리미티브는 상기 입력 프리미티브를 테셀레이팅하는 것에 의해 생성되고, 상기 출력 가시성 스트림은 매 (every) 출력 프리미티브에 대해 프리미티브가 렌더링된 타일에 대해 기여하는지 아닌지의 여부를 나타내는 일련의 비트들을 포함하고, 상기 입력 가시성 스트림은 매 입력 프리미티브에 대해 프리미티브가 렌더링된 타일에 대해 기여하는지 아닌지의 여부를 나타내는 일련의 비트들을 포함하는, 상기 출력 가시성 스트림을 생성하는 것; 및

상기 입력 가시성 스트림 및 상기 출력 가시성 스트림에 기초하여 현재 타일에 대한 출력 프리미티브들을 생성하기 위해 입력 프리미티브들을 테셀레이팅하는 것을 포함하고,

상기 입력 프리미티브들을 테셀레이팅하는 것은:

상기 입력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들에 대해 테셀레이션을 스킵하는 것; 및

상기 출력 가시성 스트림에 기초하여, 상기 입력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하는 것으로 나타내어진 입력 프리미티브들에 대해 테셀레이션을 수행하는 것으로서, 대응하는 출력 프리미티브들이 상기 출력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들의 일부에 대해 테셀레이션이 스킵되는, 상기 입력 프리미티브들에 대해 테셀레이션을 수행하는 것

을 포함하는, 그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

#### 청구항 2

제 1항에 있어서,

상기 입력 가시성 스트림을 생성하는 것은:

상기 입력 프리미티브를 버텍스 (vertex) 셰이딩하는 것;

버텍스 셰이딩된 상기 입력 프리미티브를 헐 (hull) 셰이딩하는 것;

복수의 출력 프리미티브들을 생성하기 위해 헐 셰이딩된 상기 입력 프리미티브를 테셀레이팅하는 것;

상기 복수의 출력 프리미티브들을 도메인 셰이딩하는 것;

픽셀들을 생성하기 위해 상기 복수의 출력 프리미티브들을 래스터라이징하는 것;

프레임의 타일에 대해 상기 픽셀들이 보이는지를 결정하기 위해 깊이 테스트를 수행하는 것; 및

상기 깊이 테스트에 기초하여 상기 입력 프리미티브가 상기 타일에서 보이는지 아닌지의 여부의 표시 (indication) 를 상기 입력 가시성 스트림에서 생성하는 것으로서, 상기 입력 프리미티브와 관련된 상기 출력 프리미티브들을 래스터라이징하는 것에 의해 생성된 임의의 픽셀들이 보이는 것으로 결정되면 상기 입력 프리미티브는 보이는 것으로 결정되는, 상기 표시를 상기 입력 가시성 스트림에서 생성하는 것을 포함하는, 그래픽 프

로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

### 청구항 3

삭제

### 청구항 4

제 1항에 있어서,

상기 입력 프리미티브는 패치이고 상기 출력 프리미티브는 삼각형, 쿼드 (quad) 또는 등치선 (isoline) 인, 그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

### 청구항 5

제 1항에 있어서,

상기 프레임의 상기 현재 타일을 렌더링하는 것을 더 포함하고,

상기 입력 가시성 스트림에 의해 현재 타일에서 보이지 않는 것으로 나타내어진 입력 프리미티브들에 대해 렌더링이 스킵되는, 그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

### 청구항 6

제 1항에 있어서,

상기 프레임의 상기 현재 타일을 렌더링하는 것을 더 포함하고,

상기 출력 가시성 스트림에 의해 현재 타일에서 보이지 않는 것으로 나타내어진 출력 프리미티브들에 대해 렌더링이 스킵되는, 그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

### 청구항 7

삭제

### 청구항 8

그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법으로서,

프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 각각의 입력 프리미티브에 대한 가시성 스트림을 생성하는 것; 및

상기 가시성 스트림에 기초하여 현재 타일에 대한 출력 프리미티브들을 생성하기 위해 입력 프리미티브들을 테셀레이팅하는 것을 포함하고,

상기 가시성 스트림은 깊이 테스트에 기초하여 상기 타일에서 출력 프리미티브가 보이는지 아닌지의 여부의 표시를 포함하고, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되고, 상기 가시성 스트림은 N 비트들의 길이를 가지고, 각각의 비트는 각각의 출력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타내고, 상기 출력 프리미티브는 상기 입력 프리미티브를 테셀레이팅하는 것에 의해 생성되고,

상기 입력 프리미티브들을 테셀레이팅하는 것은, 대응하는 출력 프리미티브들이 상기 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들의 일부에 대해 테셀레이션을 스킵하는 것을 포함하는, 그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

### 청구항 9

제 8항에 있어서,

상기 길이 N은, 렌더링 파이프라인에 의해 정의된 바와 같은 입력 프리미티브마다의 출력 프리미티브들의 최대 수

에 의해 결정되는, 그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

#### 청구항 10

제 8항에 있어서,

상기 가시성 스트림을 생성하는 것은:

상기 입력 프리미티브를 버텍스 셰이딩하는 것;

버텍스 셰이딩된 상기 입력 프리미티브를 힐 셰이딩하는 것;

복수의 출력 프리미티브들을 생성하기 위해 힐 셰이딩된 상기 입력 프리미티브를 테셀레이팅하는 것;

상기 복수의 출력 프리미티브들을 도메인 셰이딩하는 것;

픽셀들을 생성하기 위해 상기 복수의 출력 프리미티브들을 래스터라이징하는 것;

프레임의 타일에 대해 상기 픽셀들이 보이는지를 결정하기 위해 깊이 테스트를 수행하는 것; 및

상기 깊이 테스트에 기초하여 상기 타일에서 상기 출력 프리미티브가 보이는지 아닌지의 여부의 상기 표시를 상기 가시성 스트림에서 생성하는 것으로서, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되는, 상기 표시를 상기 가시성 스트림에서 생성하는 것을 포함하는, 그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

#### 청구항 11

제 8항에 있어서,

상기 프레임의 상기 현재 타일을 렌더링하는 것을 더 포함하고,

상기 가시성 스트림에서의 매 비트가 보이지 않는 출력 프리미티브를 나타내는 경우 입력 프리미티브들에 대해 렌더링이 스킵되는, 그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

#### 청구항 12

제 8항에 있어서,

상기 프레임의 상기 현재 타일을 렌더링하는 것을 더 포함하고,

상기 가시성 스트림에 의해 현재 타일에서 보이지 않는 것으로 나타내어진 출력 프리미티브들에 대해 렌더링이 스킵되는, 그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

#### 청구항 13

제 8항에 있어서,

상기 타일 기반의 렌더링 아키텍처는 인에이블된 지오메트리 셰이더를 구비하고, 상기 가시성 스트림은  $N \times M$  비트들의 길이를 가지며,  $M$ 은 상기 지오메트리 셰이더에 의해 생성된 출력 프리미티브들의 최대 수이고, 상기 가시성 스트림을 생성하는 것은 상기 출력 프리미티브들을 도메인 셰이딩한 이후 상기 복수의 출력 프리미티브들을 지오메트리 셰이딩하는 것을 더 포함하는, 그래픽 프로세서에 의한 타일 기반의 그래픽스 렌더링 파이프라인에서의 테셀레이션을 위한 방법.

#### 청구항 14

타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치로서,

그래픽스 데이터를 저장하도록 구성된 메모리; 및

상기 메모리와 통신하는 그래픽스 프로세서를 포함하고,

상기 그래픽스 프로세서는:

상기 그래픽스 데이터 상에서 비닝 파이프라인을 실행하도록 구성되고, 상기 비닝 파이프라인은 프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 입력 가시성 스트림을 생성하도록 구성되고, 상기 입력 가시성 스트림은 입력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타내고, 상기 비닝 파이프라인은 상기 프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 출력 가시성 스트림을 생성하도록 더 구성되고, 상기 출력 가시성 스트림은 깊이 테스트에 기초하여 상기 타일에서 출력 프리미티브가 보이는지 아닌지의 여부를 표시를 포함하고, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되고, 상기 출력 프리미티브는 상기 입력 프리미티브를 테셀레이팅하는 것에 의해 생성되고, 상기 출력 가시성 스트림은 매 출력 프리미티브에 대해 프리미티브가 렌더링된 타일에 대해 기여하는지 아닌지의 여부를 나타내는 일련의 비트들을 포함하고, 상기 입력 가시성 스트림은 매 입력 프리미티브에 대해 프리미티브가 렌더링된 타일에 대해 기여하는지 아닌지의 여부를 나타내는 일련의 비트들을 포함하고, 그리고

상기 그래픽스 프로세서는 또한 렌더링 파이프라인을 실행하도록 구성되고, 상기 렌더링 파이프라인은 상기 입력 가시성 스트림 및 상기 출력 가시성 스트림에 기초하여 현재 타일에 대한 출력 프리미티브들을 생성하기 위해 입력 프리미티브들을 테셀레이팅하도록 구성되고, 상기 입력 프리미티브들을 테셀레이팅하는 것은, 상기 입력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들에 대해 테셀레이션을 스킵하는 것, 및 상기 출력 가시성 스트림에 기초하여, 상기 입력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하는 것으로 나타내어진 입력 프리미티브들에 대해 테셀레이션을 수행하는 것을 포함하고, 대응하는 출력 프리미티브들이 상기 출력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들의 일부에 대해 테셀레이션이 스킵되는, 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

#### 청구항 15

제 14항에 있어서,

상기 비닝 파이프라인은:

상기 입력 프리미티브에 대해 버텍스 셰이딩을 수행하도록 구성된 비닝 버텍스 셰이더;

버텍스 셰이딩된 상기 입력 프리미티브에 대해 헵 셰이딩을 수행하도록 구성된 비닝 헵 셰이더;

복수의 출력 프리미티브들을 생성하기 위해 헵 셰이딩된 상기 입력 프리미티브를 테셀레이팅하도록 구성된 비닝 테셀레이터 스테이지;

상기 복수의 출력 프리미티브들에 대해 도메인 셰이딩을 수행하도록 구성된 비닝 도메인 셰이더;

픽셀들을 생성하기 위해 상기 복수의 출력 프리미티브들을 래스터라이징하도록 구성된 비닝 래스터라이저; 및

상기 픽셀들이 프레임의 타일에 대해 보이는지를 결정하기 위해 상기 픽셀들에 대해 깊이 테스트를 수행하도록 구성된 비닝 깊이 테스트 스테이지를 포함하고,

상기 비닝 파이프라인은, 상기 깊이 테스트에 기초하여 상기 입력 프리미티브가 상기 타일에서 보이는지 아닌지의 여부를 표시를 상기 입력 가시성 스트림에서 생성하고, 상기 입력 프리미티브와 관련된 상기 출력 프리미티브들을 래스터라이징하는 것에 의해 생성된 임의의 픽셀들이 보이는 것으로 결정되면 상기 입력 프리미티브는 보이는 것으로 결정되는, 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

#### 청구항 16

제 14항에 있어서,

상기 입력 프리미티브는 패치이고 상기 출력 프리미티브는 삼각형, 쿼드 또는 등치선인, 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

#### 청구항 17

제 14항에 있어서,

상기 프레임의 상기 현재 타일을 렌더링하도록 구성된 상기 렌더링 파이프라인을 더 포함하고,

상기 입력 가시성 스트림에 의해 현재 타일에서 보이지 않는 것으로 나타내어진 입력 프리미티브들에 대해 렌더링이 스킵되고, 상기 출력 가시성 스트림에 의해 상기 현재 타일에서 보이지 않는 것으로 나타내어진 출력 프리미티브들에 대해 렌더링이 스킵되는, 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

#### 청구항 18

삭제

#### 청구항 19

타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치로서,

그래픽스 데이터를 저장하도록 구성된 메모리; 및

상기 메모리와 통신하는 그래픽스 프로세서를 포함하고,

상기 그래픽스 프로세서는 상기 그래픽스 데이터 상에서 비닝 파이프라인을 실행하도록 구성되고, 상기 비닝 파이프라인은 프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 각각의 입력 프리미티브에 대한 가시성 스트림을 생성하도록 구성되고, 상기 가시성 스트림은 깊이 테스트에 기초하여 상기 타일에서 출력 프리미티브가 보이는지 아닌지의 여부의 표시를 포함하고, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되고, 상기 가시성 스트림은 N 비트들의 길이를 가지며, 각각의 비트는 각각의 출력 프리미티브가 렌더링시 각각의 타일에서 보이는지의 여부를 나타내고, 상기 출력 프리미티브는 상기 입력 프리미티브를 테셀레이팅하는 것에 의해 생성되고,

상기 그래픽스 프로세서는 또한 렌더링 파이프라인을 실행하도록 구성되고, 상기 렌더링 파이프라인은 상기 가시성 스트림에 기초하여 현재 타일에 대한 출력 프리미티브들을 생성하기 위해 입력 프리미티브들을 테셀레이팅하도록 구성되고, 상기 입력 프리미티브들을 테셀레이팅하는 것은, 대응하는 출력 프리미티브들이 상기 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들의 일부에 대해 테셀레이션을 스킵하는 것을 포함하는, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

#### 청구항 20

제 19항에 있어서,

상기 길이 N은, 상기 렌더링 파이프라인에 의해 정의된 바와 같은 입력 프리미티브마다의 출력 프리미티브들의 최대 수에 의해 결정되는, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

#### 청구항 21

제 19항에 있어서,

상기 비닝 파이프라인은:

상기 입력 프리미티브에 대해 버텍스 셰이딩을 수행하도록 구성된 비닝 버텍스 셰이더;

버텍스 셰이딩된 상기 입력 프리미티브에 대해 헵 셰이딩을 수행하도록 구성된 비닝 헵 셰이더;

복수의 출력 프리미티브들을 생성하기 위해 헵 셰이딩된 상기 입력 프리미티브를 테셀레이팅하도록 구성된 비닝 테셀레이터 스테이지;

상기 복수의 출력 프리미티브들에 대해 도메인 셰이딩을 수행하도록 구성된 비닝 도메인 셰이더;

픽셀들을 생성하기 위해 상기 복수의 출력 프리미티브들을 래스터라이징하도록 구성된 비닝 래스터라이저; 및

상기 픽셀들이 프레임의 타일에 대해 보이는지를 결정하기 위해 상기 픽셀들에 대해 깊이 테스트를 수행하도록 구성된 비닝 깊이 테스트 스테이지를 포함하고,

상기 비닝 파이프라인은, 상기 깊이 테스트에 기초하여 상기 타일에서 상기 출력 프리미티브가 보이는지 아닌지의 여부의 상기 표시를 상기 가시성 스트림에서 생성하고, 상기 출력 프리미티브를 래스터라이징하는 것에 의해

생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되는, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

## 청구항 22

제 19항에 있어서,

상기 프레임의 상기 현재 타일을 렌더링하도록 구성된 상기 렌더링 파이프라인을 더 포함하고,

상기 가시성 스트림에서의 매 비트가 보이지 않는 출력 프리미티브를 나타내는 경우 입력 프리미티브들에 대해 렌더링이 스킵되는, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

## 청구항 23

삭제

## 청구항 24

제 19항에 있어서,

상기 렌더링 파이프라인은 지오메트리 셰이더를 포함하고, 상기 가시성 스트림은  $N \times M$  비트들의 길이를 가지며,  $M$ 은 상기 지오메트리 셰이더에 의해 생성된 출력 프리미티브들의 최대 수이고, 상기 비닝 파이프라인은 상기 비닝 도메인 셰이더 뒤에 상기 복수의 출력 프리미티브들에 대한 지오메트리 셰이딩을 수행하도록 구성된 비닝 지오메트리 셰이더를 더 포함하는, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

## 청구항 25

타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치로서,

프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 입력 가시성 스트림을 생성하는 수단으로서, 상기 입력 가시성 스트림은 입력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타내는, 상기 입력 가시성 스트림을 생성하는 수단;

상기 프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 출력 가시성 스트림을 생성하는 수단으로서, 상기 출력 가시성 스트림은 깊이 테스트에 기초하여 상기 타일에서 출력 프리미티브가 보이는지 아닌지의 여부의 표시를 포함하고, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되고, 상기 출력 프리미티브는 상기 입력 프리미티브를 테셀레이팅하는 것에 의해 생성되고, 상기 출력 가시성 스트림은 매 출력 프리미티브에 대해 프리미티브가 렌더링된 타일에 대해 기여하는지 아닌지의 여부를 나타내는 일련의 비트들을 포함하고, 상기 입력 가시성 스트림은 매 입력 프리미티브에 대해 프리미티브가 렌더링된 타일에 대해 기여하는지 아닌지의 여부를 나타내는 일련의 비트들을 포함하는, 상기 출력 가시성 스트림을 생성하는 수단; 및

상기 입력 가시성 스트림 및 상기 출력 가시성 스트림에 기초하여 현재 타일에 대한 출력 프리미티브들을 생성하기 위해 입력 프리미티브들을 테셀레이팅하는 수단을 포함하고,

상기 입력 프리미티브들을 테셀레이팅하는 수단은:

상기 입력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들에 대해 테셀레이션을 스킵하는 수단; 및

상기 출력 가시성 스트림에 기초하여, 상기 입력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하는 것으로 나타내어진 입력 프리미티브들에 대해 테셀레이션을 수행하는 수단으로서, 대응하는 출력 프리미티브들이 상기 출력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들의 일부에 대해 테셀레이션이 스킵되는, 상기 입력 프리미티브들에 대해 테셀레이션을 수행하는 수단

을 포함하는, 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

## 청구항 26

제 25항에 있어서,

상기 입력 가시성 스트림을 생성하는 수단은:

상기 입력 프리미티브를 버텍스 셰이딩하는 수단;

버텍스 셰이딩된 상기 입력 프리미티브를 힐 셰이딩하는 수단;

복수의 출력 프리미티브들을 생성하기 위해 힐 셰이딩된 상기 입력 프리미티브를 테셀레이팅하는 수단;

상기 복수의 출력 프리미티브들을 도메인 셰이딩하는 수단;

픽셀들을 생성하기 위해 상기 복수의 출력 프리미티브들을 래스터라이징하는 수단;

프레임의 타일에 대해 상기 픽셀들이 보이는지를 결정하기 위해 상기 픽셀들에 대해 깊이 테스트를 수행하는 수단; 및

상기 깊이 테스트에 기초하여 상기 입력 프리미티브가 상기 타일에서 보이는지 아닌지의 여부의 표시를 상기 입력 가시성 스트림에서 생성하는 수단으로서, 상기 입력 프리미티브와 관련된 상기 출력 프리미티브들을 래스터라이징하는 것에 의해 생성된 임의의 픽셀들이 보이는 것으로 결정되면 상기 입력 프리미티브는 보이는 것으로 결정되는, 상기 표시를 상기 입력 가시성 스트림에서 생성하는 수단을 포함하는, 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

## 청구항 27

삭제

## 청구항 28

제 25항에 있어서,

상기 입력 프리미티브는 패치이고 상기 출력 프리미티브는 삼각형, 쿼드 또는 등치선인, 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

## 청구항 29

제 25항에 있어서,

상기 프레임의 상기 현재 타일을 렌더링하는 수단을 더 포함하고,

상기 입력 가시성 스트림에 의해 현재 타일에서 보이지 않는 것으로 나타내어진 입력 프리미티브들에 대해 렌더링이 스킵되는, 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

## 청구항 30

제 25항에 있어서,

상기 프레임의 상기 현재 타일을 렌더링하는 수단을 더 포함하고,

상기 출력 가시성 스트림에 의해 현재 타일에서 보이지 않는 것으로 나타내어진 출력 프리미티브들에 대해 렌더링이 스킵되는, 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

## 청구항 31

삭제

## 청구항 32

타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치로서,

프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 각각의 입력 프리미티브에 대한 가시성 스트림을 생성하는 수단;

상기 가시성 스트림에 기초하여 현재 타일에 대한 출력 프리미티브들을 생성하기 위해 입력 프리미티브들을 테셀레이팅하는 수단; 및

상기 프레임의 상기 현재 타일을 렌더링하는 수단을 포함하고,

상기 가시성 스트림은 깊이 테스트에 기초하여 상기 타일에서 출력 프리미티브가 보이는지 아닌지의 여부의 표시를 포함하고, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되고, 상기 가시성 스트림은 N 비트들의 길이를 가지고, 각각의 비트는 출력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타내고, 상기 출력 프리미티브는 상기 입력 프리미티브를 테셀레이팅하는 것에 의해 생성되고, 상기 가시성 스트림에서의 매 비트가 보이지 않는 출력 프리미티브를 나타내는 경우 입력 프리미티브들에 대해 렌더링이 스킵되고,

상기 입력 프리미티브들을 테셀레이팅하는 것은, 대응하는 출력 프리미티브들이 상기 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들의 일부에 대해 테셀레이션을 스킵하는 것을 포함하는, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

### 청구항 33

제 32항에 있어서,

상기 길이 N은, 렌더링 파이프라인에 의해 정의된 바와 같은 입력 프리미티브마다의 출력 프리미티브들의 최대 수에 의해 결정되는, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

### 청구항 34

제 32항에 있어서,

가시성 스트림을 생성하는 수단은:

상기 입력 프리미티브를 버텍스 셰이딩하는 수단;

버텍스 셰이딩된 상기 입력 프리미티브를 헵 셰이딩하는 수단;

복수의 출력 프리미티브들을 생성하기 위해 헵 셰이딩된 상기 입력 프리미티브를 테셀레이팅하는 수단;

상기 복수의 출력 프리미티브들을 도메인 셰이딩하는 수단;

픽셀들을 생성하기 위해 상기 복수의 출력 프리미티브들을 래스터라이징하는 수단;

프레임의 타일에 대해 상기 픽셀들이 보이는지를 결정하기 위해 상기 픽셀들에 대해 깊이 테스트를 수행하는 수단; 및

상기 깊이 테스트에 기초하여 상기 타일에서 상기 출력 프리미티브가 보이는지 아닌지의 여부의 상기 표시를 상기 출력 가시성 스트림에서 생성하는 수단으로서, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되는, 상기 표시를 상기 출력 가시성 스트림에서 생성하는 수단을 포함하는, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

### 청구항 35

삭제

### 청구항 36

제 32항에 있어서,

상기 프레임의 상기 현재 타일을 렌더링하는 수단을 더 포함하고,

상기 가시성 스트림에 의해 현재 타일에서 보이지 않는 것으로 나타내어진 출력 프리미티브들에 대해 렌더링이 스킵되는, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

### 청구항 37

제 32항에 있어서,

상기 타일 기반의 렌더링 아키텍처는 인에이블된 지오메트리 셰이더를 구비하고, 상기 가시성 스트림은 NxM 비

트들의 길이를 가지며, M은 상기 지오메트리 셰이더에 의해 생성된 출력 프리미티브들의 최대 수이고, 상기 가시성 스트림을 생성하는 수단은 상기 출력 프리미티브들을 도메인 셰이딩하는 수단 뒤에 상기 복수의 출력 프리미티브들을 지오메트리 셰이딩하는 수단을 더 포함하는, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치.

### 청구항 38

명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은 실행시 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하는 디바이스의 하나 이상의 프로세서들로 하여금:

프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 입력 가시성 스트림을 생성하게 하는 것으로서, 상기 입력 가시성 스트림은 입력 프리미티브가 렌더링시 각각의 타일에서 보이는지의 여부를 나타내는, 상기 입력 가시성 스트림을 생성하게 하고,

상기 프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 출력 가시성 스트림을 생성하게 하는 것으로서, 상기 출력 가시성 스트림은 깊이 테스트에 기초하여 상기 타일에서 출력 프리미티브가 보이는지 아닌지의 여부를 표시를 포함하고, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되고, 상기 출력 프리미티브는 상기 입력 프리미티브를 테셀레이팅하는 것에 의해 생성되고, 상기 출력 가시성 스트림은 매 출력 프리미티브에 대해 프리미티브가 렌더링된 타일에 대해 기여하는지 아닌지의 여부를 나타내는 일련의 비트들을 포함하고, 상기 입력 가시성 스트림은 매 입력 프리미티브에 대해 프리미티브가 렌더링된 타일에 대해 기여하는지 아닌지의 여부를 나타내는 일련의 비트들을 포함하는, 상기 출력 가시성 스트림을 생성하게 하고, 그리고

상기 입력 가시성 스트림 및 상기 출력 가시성 스트림에 기초하여 현재 타일에 대한 출력 프리미티브들을 생성하기 위해 입력 프리미티브들을 테셀레이팅하게 하고,

상기 입력 프리미티브들을 테셀레이팅하는 것은:

상기 입력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들에 대해 테셀레이션을 스킵하는 것; 및

상기 출력 가시성 스트림에 기초하여, 상기 입력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하는 것으로 나타내어진 입력 프리미티브들에 대해 테셀레이션을 수행하는 것으로서, 대응하는 출력 프리미티브들이 상기 출력 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들의 일부에 대해 테셀레이션이 스킵되는, 상기 입력 프리미티브들에 대해 테셀레이션을 수행하는 것

을 포함하는, 명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체.

### 청구항 39

제 38항에 있어서,

상기 하나 이상의 프로세서들로 하여금 추가로:

상기 입력 프리미티브에 대해 버텍스 셰이딩을 수행하게 하고;

버텍스 셰이딩된 상기 입력 프리미티브에 대해 헵 셰이딩을 수행하게 하고;

복수의 출력 프리미티브들을 생성하기 위해 헵 셰이딩된 상기 입력 프리미티브를 테셀레이팅하게 하고;

상기 복수의 출력 프리미티브들에 대해 도메인 셰이딩을 수행하게 하고;

픽셀들을 생성하기 위해 상기 복수의 출력 프리미티브들을 래스터라이징하게 하고;

프레임의 타일에 대해 상기 픽셀들이 보이는지를 결정하기 위해 상기 픽셀들에 대해 깊이 테스트를 수행하게 하고;

상기 깊이 테스트에 기초하여 상기 입력 프리미티브가 상기 타일에서 보이는지 아닌지의 여부를 표시를 상기 입력 가시성 스트림에서 생성하게 하고, 상기 입력 프리미티브와 관련된 상기 출력 프리미티브들을 래스터라이징하는 것에 의해 생성된 임의의 픽셀들이 보이는 것으로 결정되면 상기 입력 프리미티브는 보이는 것으로 결정되

는, 명령들을 저장하는 비밀시적 컴퓨터 판독가능 저장 매체.

#### 청구항 40

제 38항에 있어서,

상기 입력 프리미티브는 패치이고 상기 출력 프리미티브는 삼각형, 쿼드 또는 등치선인, 명령들을 저장하는 비밀시적 컴퓨터 판독가능 저장 매체.

#### 청구항 41

제 38항에 있어서,

상기 하나 이상의 프로세서들로 하여금 추가로:

상기 프레임의 상기 현재 타일을 렌더링하게 하고,

상기 입력 가시성 스트림에 의해 현재 프레임에서 보이지 않는 것으로 나타내어진 입력 프리미티브들에 대해 렌더링이 스킵되고, 상기 출력 가시성 스트림에 의해 상기 현재 프레임에서 보이지 않는 것으로 나타내어진 출력 프리미티브들에 대해 렌더링이 스킵되는, 명령들을 저장하는 비밀시적 컴퓨터 판독가능 저장 매체.

#### 청구항 42

삭제

#### 청구항 43

명령들을 저장하는 비밀시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은 실행시 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하는 디바이스의 하나 이상의 프로세서들로 하여금:

프레임을 렌더링하기 전에 상기 프레임의 각각의 타일에 대한 각각의 입력 프리미티브에 대한 가시성 스트림을 생성하게 하고,

상기 가시성 스트림에 기초하여 현재 타일에 대한 출력 프리미티브들을 생성하기 위해 입력 프리미티브들을 테셀레이팅하게 하고,

상기 가시성 스트림은 깊이 테스트에 기초하여 상기 타일에서 출력 프리미티브가 보이는지 아닌지의 여부의 표시를 포함하고, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되고, 상기 가시성 스트림은 N 비트들의 길이를 가지고, 각각의 비트는 각각의 출력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타내고, 상기 출력 프리미티브는 상기 입력 프리미티브를 테셀레이팅하는 것에 의해 생성되고,

상기 입력 프리미티브들을 테셀레이팅하는 것은, 대응하는 출력 프리미티브들이 상기 가시성 스트림에 의해 상기 현재 타일에 대해 기여하지 않는 것으로 나타내어진 입력 프리미티브들의 일부에 대해 테셀레이션을 스킵하는 것을 포함하는, 명령들을 저장하는 비밀시적 컴퓨터 판독가능 저장 매체.

#### 청구항 44

제 43항에 있어서,

상기 길이 N은, 렌더링 파이프라인에 의해 정의된 바와 같은 입력 프리미티브마다의 출력 프리미티브들의 최대 수에 의해 결정되는, 명령들을 저장하는 비밀시적 컴퓨터 판독가능 저장 매체.

#### 청구항 45

제 43항에 있어서,

상기 하나 이상의 프로세서들로 하여금 추가로:

상기 입력 프리미티브에 대해 버텍스 셰이딩을 수행하게 하고;

버텍스 셰이딩된 상기 입력 프리미티브에 대해 힐 셰이딩을 수행하게 하고;

복수의 출력 프리미티브들을 생성하기 위해 힐 셰이딩된 상기 입력 프리미티브를 테셀레이팅하게 하고;

상기 복수의 출력 프리미티브들에 대해 도메인 셰이딩을 수행하게 하고;

픽셀들을 생성하기 위해 상기 복수의 출력 프리미티브들을 래스터라이징하게 하고;

프레임의 타일에 대해 상기 픽셀들이 보이는지를 결정하기 위해 상기 픽셀들에 대해 깊이 테스트를 수행하게 하고; 그리고

상기 깊이 테스트에 기초하여 상기 타일에서 상기 출력 프리미티브가 보이는지 아닌지의 여부의 상기 표시를 상기 가시성 스트림에서 생성하게 하고, 상기 출력 프리미티브를 래스터라이징하는 것에 의해 생성된 임의의 픽셀이 보이는 것으로 결정되면 상기 출력 프리미티브는 보이는 것으로 결정되는, 명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체.

#### 청구항 46

제 43항에 있어서,

상기 하나 이상의 프로세서들로 하여금 추가로:

상기 프레임의 상기 현재 타일을 렌더링하게 하고,

상기 가시성 스트림에서의 매 비트가 보이지 않는 출력 프리미티브를 나타내는 경우 입력 프리미티브들에 대해 렌더링이 스킵되는, 명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체.

#### 청구항 47

제 46항에 있어서,

상기 가시성 스트림에 의해 현재 타일에서 보이지 않는 것으로 나타내어진 출력 프리미티브들에 대해 렌더링이 스킵되는, 명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체.

#### 청구항 48

제 43항에 있어서,

상기 명령들은 상기 하나 이상의 프로세서들로 하여금 추가로, 지오메트리 셰이더를 포함하는 렌더링 파이프라인을 실행하게 하고,

상기 가시성 스트림은  $N \times M$  비트들의 길이를 가지며,  $M$ 은 상기 지오메트리 셰이더에 의해 생성된 출력 프리미티브들의 최대 수이고, 상기 명령들은 프로세서로 하여금 추가로, 도메인 셰이딩을 수행한 이후 상기 복수의 출력 프리미티브들에 대해 지오메트리 셰이딩을 수행하게 하는, 명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체.

### 발명의 설명

#### 기술 분야

[0001] 본 개시는 그래픽스 프로세싱을 위한 기술들에 관한 것으로, 특히 타일 기반의 렌더링 아키텍처를 사용하는 경우의 테셀레이션 (tessellation) 을 위한 기술들에 관한 것이다.

#### 배경 기술

[0002] 그래픽컬 유저 인터페이스들 및 비디오 게임들에 대한 콘텐츠와 같은, 디스플레이용의 시각적 콘텐츠는, 그래픽스 프로세싱 유닛 (graphics processing unit; GPU) 에 의해 생성될 수도 있다. GPU는 2차원 또는 3차원 (3D) 오브젝트들을, 디스플레이될 수도 있는 2차원 (2D) 픽셀 표현으로 변환할 수도 있다. 3D 오브젝트들에 관한 정보를 디스플레이될 수 있는 비트맵으로 변환하는 것은 픽셀 렌더링으로서 알려져 있고, 상당한 메모리와 프로세싱 파워를 필요로 한다. 과거, 3D 그래픽스 기능은 성능 좋은 (powerful) 워크스테이션들에서만 이용 가능했다. 그러나, 이제 3D 그래픽스 가속기들이, PC (personal computers) 뿐만 아니라 임베딩된 디바이스

들, 예컨대 스마트폰들, 태블릿 컴퓨터들, 휴대형 미디어 플레이어들, 휴대형 비디오 게임용 콘솔들 등에서도 일반적으로 발견된다. 통상적으로, 임베딩된 디바이스는 종래의 PC들과 비교하여 낮은 계산 파워와 메모리 성능을 갖는다. 이와 같이, 3D 그래픽스 렌더링 기술들에서의 증가된 복잡도는, 임베딩된 시스템 상에서 이러한 기술들을 구현하는 경우 어려움들을 야기시킨다.

## 발명의 내용

## 해결하려는 과제

## 과제의 해결 수단

- [0003] 개요
- [0004] 일반적으로, 본 개시는 그래픽스 프로세싱을 위한 기술들, 특히, 테셀레이션을 활용하는 타일 기반의 그래픽스 프로세싱 아키텍처를 갖는 컴퓨팅 디바이스에서의 그래픽스 프로세싱을 위한 기술들을 설명한다. 본 개시는 프레임의 타일에서의 그래픽스 프리미티브들의 가시성을 나타내는 출력 및/또는 입력 가시성 스트림들(visibility streams)의 생성을 제안한다. 이러한 가시성 스트림들은 타일 기반의 그래픽스 프로세싱 아키텍처가, 프레임에서 보이지 않는 그래픽스 프리미티브들의 렌더링을 스킵하는 것을 가능하게 한다. 이렇게, 불필요한 프로세싱이 방지될 수도 있고, 따라서 성능을 향상시키게 된다.
- [0005] 본 개시의 일 예에서, 타일 기반의 그래픽스에서의 테셀레이션을 위한 방법이 제안된다. 그 방법은 프레임의 각각의 타일에 대한 입력 가시성 스트림을 생성하는 것, 및 프레임의 각각의 타일에 대한 출력 가시성 스트림을 생성하는 것을 포함하고, 입력 가시성 스트림은 렌더링시 각각의 타일에서 입력 프리미티브가 보이는지 아닌지의 여부를 나타내고, 출력 가시성 스트림은 렌더링시 출력 프리미티브가 각각의 타일에서 보이는지 아닌지의 여부를 나타내며, 출력 프리미티브는 입력 프리미티브를 테셀레이팅하는 것에 의해 생성된다.
- [0006] 본 개시의 다른 예에서, 타일 기반의 그래픽스 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치가 제안된다. 그 장치는, 프레임의 각각의 타일에 대한 입력 가시성 스트림을 생성하도록 구성된 비닝 파이프라인을 포함한다. 입력 가시성 스트림은, 입력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타낸다. 비닝 파이프라인은 프레임의 각각의 타일에 대한 출력 가시성 스트림을 생성하도록 더 구성된다. 출력 가시성 스트림은 출력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타내고, 출력 프리미티브는 입력 프리미티브를 테셀레이팅하는 것에 의해 생성된다.
- [0007] 본 개시의 다른 예에서, 타일 기반의 렌더링 파이프라인에서의 테셀레이션을 위한 방법이 제안된다. 그 방법은, 프레임의 각각의 타일에 대한 각각의 입력 프리미티브에 대한 가시성 스트림을 생성하는 것을 포함하고, 가시성 스트림은 N 비트들의 길이를 가지고, 각각의 비트는 출력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타내고, 출력 프리미티브는 입력 프리미티브를 테셀레이팅하는 것에 의해 생성된다.
- [0008] 본 개시의 다른 예에서, 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하도록 구성된 장치가 제안된다. 그 장치는, 프레임의 각각의 타일에 대한 각각의 입력 프리미티브에 대한 가시성 스트림을 생성하도록 구성된 비닝 파이프라인을 포함하고, 가시성 스트림은 N 비트들의 길이를 가지고, 각각의 비트는 출력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타내고, 출력 프리미티브는 입력 프리미티브를 테셀레이팅하는 것에 의해 생성된다.
- [0009] 하나 이상의 예의 상세들은 첨부된 도면과 하기의 설명으로부터 설명된다. 다른 특징들, 목적들 및 이점들은 설명과 도면들, 및 특허청구범위로부터 명확해질 것이다.

## 도면의 간단한 설명

- [0010] 도 1은 타일 기반의 렌더링 아키텍처에서의 프레임의 타일들을 예시하는 개념도이다.
- 도 2는 일 예의 그래픽스 프로세싱 시스템을 예시하는 블록도이다.
- 도 3은 테셀레이션을 이용하는 일 예의 그래픽스 파이프라인을 예시하는 블록도이다.
- 도 4는 입력 가시성 스트림 및 출력 가시성 스트림을 생성하는 일 예의 비닝 파이프라인(binning pipeline)을

예시하는 블록도이다.

도 5는 일 예의 입력 가시성 스트림 및 일 예의 출력 가시성 스트림을 예시하는 개념도이다.

도 6은 입력 프리미티브를 테셀레이팅하는 일 예의 기술을 예시하는 개념도이다.

도 7은 입력 프리미티브를 테셀레이팅하는 일 예의 기술을 예시하는 개념도이다.

도 8은 입력 가시성 스트림을 생성하는 일 예의 비닝 파이프라인을 예시하는 블록도이다.

도 9는 일 예의 단일의 가시성 스트림을 예시하는 개념도이다.

도 10은 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하는 일 예의 방법을 예시하는 흐름도이다.

도 11은 타일 기반의 렌더링 파이프라인에서 비닝 패스의 일 예의 방법을 예시하는 흐름도이다.

도 12는 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하는 다른 예의 방법을 예시하는 흐름도이다.

도 13은 타일 기반의 렌더링 파이프라인에서 비닝 패스의 다른 예의 방법을 예시하는 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

#### 상세한 설명

본 개시는 그래픽스 프로세싱을 위한 기술들에 관한 것으로, 특히 타일 기반의 렌더링 아키텍처를 사용하는 경우 테셀레이션을 위한 기술들에 관한 것이다.

마이크로소프트의 다이렉트X11 (DX11) 어플리케이션 프로그래밍 인터페이스 (application programming interface; API) 를 포함하는 몇몇 그래픽스 렌더링 파이프라인들은 그래픽스 프리미티브들의 테셀레이션을 허용하는 추가적인 프로세싱 스테이지들을 포함한다. 테셀레이션은, 그래픽스 프리미티브들을 더 작은 그래픽스 프리미티브들로 분할하여 디스플레이된 이미지에서의 더 미세한 디테일을 가능하게 하는 프로세스이다.

테셀레이터 스테이지를 갖는 레퍼런스 구현에는 마이크로소프트 DX11 소프트웨어 레퍼런스 래스터라이저 (software reference rasterizer) 에 의해 잘 정의되지만, 이러한 구현에는, 임베딩된 디바이스들 (예를 들면, 태블릿 컴퓨터들, 모바일 폰들, 모바일 게임용 디바이스들, 모바일 미디어 플레이어들 등) 에서의 그래픽스 프로세서들에서 사용되는 것과 같은, 타일 기반의 렌더링 아키텍처에 아주 효율적으로 매핑되지 않는다. 타일 기반의 렌더링 (종종 비닝 (binning) 으로 칭해짐) 은 3D 장면의 한 프레임, 그 프레임을 더 작은 부분들 (예를 들면, 직사각형 타일들) 로 나누고 이들 부분들을 개별적으로 렌더링하는 것에 의해 렌더링한다. 타일 기반의 렌더링은, 전용 그래픽스 메모리가 거의 이용가능하지 않은 어플리케이션들에 대해, 예컨대 모바일 어플리케이션들에 대해 유용하다. 도 1은 타일 기반의 렌더링 아키텍처에서의 프레임의 타일들을 예시하는 개념도이다. 프레임 (22) 은, 타일 (24) 과 같은 복수의 타일들로 분할될 수도 있다. 그 다음, 그래픽스 프로세싱은 타일 단위 기준으로 수행되어 전체 프레임을 렌더링한다.

마이크로소프트 DX11 소프트웨어 레퍼런스 래스터라이저를 포함하는 테셀레이션 스테이지로써 그래픽스 데이터를 렌더링하는 종래의 기술들은, 3D 장면의 전체 프레임이 전체로서 렌더링될 것이라고 가정하여 설계된다. 이와 같이, 이러한 레퍼런스 래스터라이저에서 사용되는 테셀레이션 알고리즘은 전체 입력 프리미티브를 출력 프리미티브들의 전체 세트로 항상 테셀레이팅한다. 타일 기반의 렌더링 아키텍처에서, 이것은 많은 무관한 그래픽스 프로세싱 유닛 (GPU) 의 원인이 될 수 있고 테셀레이팅된 출력 프리미티브들의 부분으로서만의 테셀레이션 프로세싱은 현재 렌더링된 타일에 실제 기여할 수도 있다 (즉, 몇몇 출력 프리미티브들은 다른 타일에서의 현재 렌더링된 타일 외부에 있을 것이다).

이러한 관점에서, 본 개시는 타일 기반의 그래픽스 렌더링 아키텍처에서의 테셀레이션을 위한 방법 및 장치를 제안한다. 개시된 기술들은 프레임의 각각의 타일에 대한 입력 가시성 스트림을 생성하는 것, 및 프레임의 각각의 타일에 대한 출력 가시성 스트림을 생성하는 것을 포함하고, 입력 가시성 스트림은 렌더링시 각각의 타일에서 입력 프리미티브가 보이는지 아닌지의 여부를 나타내고, 출력 가시성 스트림은 렌더링시 출력 프리미티브가 각각의 타일에서 보이는지 아닌지의 여부를 나타내며, 출력 프리미티브는 입력 프리미티브를 테셀레이팅하는 것에 의해 생성된다. 이러한 방식에서, 입력 가시성 스트림에 기초하여, 타일에서 보이지 않는 전체 입력 프리미티브에 대해 테셀레이션이 스킵될 수도 있다. 또한, 출력 가시성 스트림에 기초하여, 입력 프리미티브의 일부가 보이지 않더라도, 타일에서 보이지 않는 출력 프리미티브들 중 어떤 출력 프리미티브들에 대해 테셀레이션이 스킵될 수도 있다. 이와 같이, 불필요한 테셀레이션이 방지된다. 또한, 테셀레이션이 스

킵되는 것으로 인해 몇몇 출력 프리미티브들이 생성되지 않기 때문에, 그래픽스 파이프라인의 나중의 스테이지들에서, 생성되지 않은 출력 프리미티브들의 프로세싱이 방지된다.

[0017] 본 개시의 다른 예에서, 프레임의 각각의 타일에 대한 각각의 입력 프리미티브에 대해 단일의 가시성 스트림이 생성된다. 가시성 스트림은 N 비트들의 길이를 갖는데, 각 비트는 출력 프리미티브가 렌더링시 각 타일에서 보이는지 아닌지의 여부를 나타내고, 출력 프리미티브는 입력 프리미티브를 테셀레이팅하는 것에 의해 생성된다.

[0018] 도 2는 본 개시의 기술들을 구현하도록 구성될 수도 있는 디바이스 (2) 를 예시하는 블록도이다. 디바이스 (2) 의 예들은, 무선 디바이스들, 이동 전화기들, PDA들 (personal digital assistants), 휴대형 미디어 플레이어들, (내장 디스플레이가 있거나 또는 없는) 비디오 게임용 콘솔들, 모바일 화상 회의 유닛들, 랩탑 컴퓨터들, 데스크탑 컴퓨터들, 텔레비전 셋탑 박스들, 태블릿 컴퓨팅 디바이스들, 이북 리더들 등을 포함하지만, 이들 에 제한되지는 않는다. 디바이스 (2) 는 프로세서 (4), 그래픽 프로세싱 유닛 (GPU; 6), 디스플레이 (8), 프레임 버퍼 (10), 저장 디바이스 (12), 트랜시버 모듈 (14), 유저 인터페이스 (16), 저해상도 버퍼 (18) (저해상도-z 버퍼 또는 LRZ (low resolution-z) 버퍼로서 칭해질 수도 있다), 및 타일 메모리 (20) 를 포함할 수도 있다. 디바이스 (2) 는, 명확화를 위해 도 2에 도시되지 않은 부가적인 모듈들 또는 유닛들을 포함할 수도 있다. 예를 들면, 디바이스 (2) 는, 디바이스 (2) 가 이동 무선 전화기인 경우의 예들에서 전화 통신들을 실현하기 위해 스피커 및 마이크를 포함할 수도 있는데, 이들 중 어느 것도 도 2에 도시되지 않았다. 또한, 디바이스 (2) 에서 도시된 다양한 모듈들 및 유닛들은 디바이스 (2) 의 모든 예에서 필수적이지 않을 수도 있다. 예를 들면, 유저 인터페이스 (16) 와 디스플레이 (8) 는, 디바이스 (2) 가 비디오 게임용 콘솔 또는 데스크탑 컴퓨터인 예들에서 디바이스 (2) 외부에 있을 수도 있다.

[0019] 프로세서 (4) 는 하나 이상의 어플리케이션들을 실행할 수도 있다. 어플리케이션들의 예들은 웹브라우저, 이메일 어플리케이션들, 스프레드시트들, 비디오 게임들, 또는 표시를 위해 가시적인 오브젝트들을 생성하는 다른 어플리케이션들 포함한다. 하나 이상의 어플리케이션들은 저장 디바이스 (12) 내에 저장될 수도 있다. 몇몇 경우들에서, 프로세서 (4) 는 (예를 들면, 인터넷 또는 다른 네트워크로부터) 트랜시버 모듈 (14) 을 통해 하나 이상의 어플리케이션들을 다운로드할 수도 있다. 프로세서 (4) 는 유저 인터페이스 (16) 를 통한 유저에 의한 선택에 기초하여 하나 이상의 어플리케이션들을 실행할 수도 있다. 몇몇 예들에서, 프로세서 (4) 는 유저 상호작용 없이 하나 이상의 어플리케이션들을 실행할 수도 있다.

[0020] 프로세서 (4) 의 예들은 디지털 신호 프로세서 (DSP), 범용 마이크로프로세서, 주문형 반도체 (ASIC), 필드 프로그램 가능 로직 어레이 (FPGA), 또는 다른 등가의 집적 또는 이산 로직 회로부를 포함하는데, 이들에 제한되는 것은 아니다. 저장 디바이스 (12) 는 하나 이상의 컴퓨터 판독가능 저장 매체들을 포함할 수도 있다. 저장 디바이스 (12) 의 예들은 RAM (random access memory), ROM (read only memory), EEPROM (electrically erasable programmable read-only memory), CD-ROM, 블루레이 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기적 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 이송 또는 저장하기 위해 사용될 수 있으며 컴퓨터 또는 프로세서에 의해 액세스 될 수 있는 임의의 다른 매체를 포함할 수 있다. 몇몇 양태들에 있어서, 저장 디바이스 (12) 는, 호스트 프로세서 (4) 및/또는 GPU (6) 로 하여금, 본 개시에서의 호스트 프로세서 (4) 및 GPU (6) 에 의한 기능들을 수행하게 하는 명령들을 포함할 수도 있다.

[0021] 저장 디바이스 (12) 는, 몇몇 예들에서, 비일시적 저장 매체로서 간주될 수도 있다. 용어 "비일시적"이란 저장 매체가 반송파 또는 전파 신호로 구체화되지 않는 것을 나타낼 수도 있다. 그러나, 용어 "비일시적"은 저장 디바이스 (12) 가 움직일 수 없다는 것을 의미하는 것으로 해석되어선 안된다. 일 예로서, 저장 디바이스 (12) 는 디바이스 (2) 로부터 제거되어, 다른 디바이스로 이동될 수도 있다. 다른 예로서, 저장 디바이스 (12) 와 거의 유사한 저장 디바이스가 디바이스 (2) 에 삽입될 수도 있다. 어떤 예들에서, 비일시적 저장 매체는 경시적으로 변할 수 있는 데이터를 (예를 들면, RAM에) 저장할 수도 있다.

[0022] 유저 인터페이스 (16) 의 예들은 트랙볼, 마우스, 키보드, 게임용 컨트롤러 및 다른 타입의 입력 디바이스들을 포함하는데, 이들에 제한되는 것은 아니다. 유저 인터페이스 (16) 는 터치스크린일 수도 있고 디스플레이 (8) 의 일부로서 내장될 수도 있다. 트랜시버 모듈 (14) 은, 디바이스 (2) 와 다른 디바이스 또는 네트워크 사이의 무선 또는 유선 통신을 가능하게 하는 회로부를 포함할 수도 있다. 트랜시버 모듈 (14) 은 변조기들, 복조기들, 증폭기들 및 유선 또는 무선 통신을 위한 다른 이러한 회로부를 포함할 수도 있다.

[0023] GPU (6) 는 고도의 병렬 구조를 사용하여 데이터를 빠르게 조작하도록 설계된 전용 프로세서이다. 특히,

GPU (6) 는 그래픽적 프로세싱을 수행하기 위해 렌더링 파이프라인의 프로그램 가능하고 고정된 기능 스테이지들을 실행하도록 구성될 수도 있다. 일 예에서, GPU (6) 는 3차원 오브젝트들을 디스플레이용의 2차원 공간으로 렌더링하기 위해 3차원 그래픽스 렌더링 파이프라인을 실행하도록 구성될 수도 있다. 예를 들면, GPU (6) 는 셰이딩, 블렌딩, 일루미네이팅과 같은 기능들, 및 디스플레이 (8) 상에 디스플레이될 픽셀들에 대한 픽셀 값들을 생성하기 위한 기능들을 수행할 수도 있다. GPU (6) 에 의해 생성된 픽셀들은 디스플레이 이전에 프레임 버퍼 (10) 에 저장될 수도 있다. 몇몇 예들에서, 셰이딩, 블렌딩, 및 일루미네이팅과 같은 기능들을 수행하기 이전에, GPU (6) 는 타일링 기능으로도 칭해지는 비닝 패스 (binning pass) 를 수행할 수도 있다.

[0024] 도 1에서 하나의 GPU (6) 가 예시되지만, 본 개시의 양태들은 그렇게 제한되지 않는다. 몇몇 예들에서, 디바이스 (2) 는 GPU (6) 와 유사한 복수의 GPU들 또는 GPU 코어들을 포함할 수도 있다. 그래픽스 프로세싱 태스크들은 이들 GPU들 또는 GPU 코어들 사이에서 분할될 수도 있다.

[0025] 비닝 패스를 실행하기 위해, 도 1에 도시된 바와 같이, GPU (6) 는 디스플레이 (8) 상의 픽셀들을, 타일들로 칭해지는, 픽셀들의 복수의 블록들로 분할될 수도 있다. 예를 들면, 디스플레이 (8) 는 640x480 픽셀들을 포함하도록 구성된다. 일 예로서, GPU (6) 는 디스플레이를 10x10 타일들로 분할할 수도 있는데, 각각의 타일은 64x48 픽셀들을 포함한다.

[0026] 타일 메모리 (20) 는, 비닝 패스 동안, 이미지 표면이 속하는 특정 타일들을 나타낼 수도 있다. 몇몇 예들에서, 타일 메모리 (20) 는 저장 디바이스 (12) 의 일부로서 형성될 수도 있지만, 본 개시의 양태들은 그렇게 제한되지 않는다. GPU (6) 는 이미지 표면에 대한 표면 식별자 값을, 그 이미지 표면에 대한 비닝 패스 동안, 그 이미지 표면이 어떤 타일들에 속하는지에 기초하여 타일 메모리 (20) 의 적절한 저장 위치들 내에 저장할 수도 있다. 타일 메모리 (20) 는 복수의 레이어들을 포함하는 것으로 간주될 수도 있다. 각각의 레이어는 이미지 표면들 중 하나에 대응할 수도 있다. 각각의 레이어는 복수의 저장 위치들을 포함하는 2차원 레이어일 수도 있다. 각각의 저장 위치는 디스플레이 (8) 의 타일들 중 하나에 대응할 수도 있다.

[0027] 저해상도 버퍼 (18) 는 복수의 저장 위치들을 갖는 2차원 버퍼일 수도 있다. 몇몇 예들에서, 저해상도 버퍼 (18) 는 저장 디바이스 (12) 의 일부로서 형성될 수도 있지만, 본 개시의 양태들은 그렇게 제한되지 않는다.

하기예 더 상세히 설명되는 바와 같이, 저해상도 버퍼 (18) 는 후속적으로 차단되는 픽셀들을 불필요하게 렌더링하는 것을 방지하기 위해 어떤 픽셀들이 렌더링되어야 하는지의 효율적인 표시 (indication) 를 촉진한다.

저해상도 버퍼 (18) 에서의 각각의 저장 위치는 디스플레이 (8) 상에서 표현되는 픽셀들의 블록에 대응할 수도 있다. 예를 들면, 디스플레이 (8) 상에서 표현되는 4x4 픽셀들의 블록은 저해상도 버퍼 (18) 에서의 하나의 저장 위치에 대응할 수도 있다. 4x4 픽셀들의 블록은 예시의 목적으로 제공되며, 제한하는 것으로 간주되어선 안된다. 저해상도 버퍼 (18) 는, 저해상도 버퍼 (18) 의 각각의 저장 위치가 단일의 픽셀보다는 복수의 픽셀들에 대응하기 때문에 저해상도 버퍼로서 간주될 수도 있다.

[0028] 도 3은 도 2의 GPU (6) 에 의해 실행될 수도 있는 예시적인 그래픽스 파이프라인 (30) 을 묘사한다. 위에서 언급된 바와 같이, 그래픽스 파이프라인 (30) 은 타일 기반의 렌더링 아키텍처에서 구현될 수도 있는데, 타일 기반의 렌더링 아키텍처에서는 전체 프레임의 타일들이 별도로 렌더링되고 그 다음 디스플레이용으로 결합된다.

그래픽스 파이프라인 (30) 은 마이크로소프트의 다이렉트X11 API에 따라 구현될 수도 있다. 다이렉트X11 API는 다이렉트X API들의 이전의 버전들 (iterations) 에서 발견되지 않는 3개의 스테이지들, 즉 프로그램 가능 쉐이더 (hull shader; HS; 36), 고정 기능 테셀레이터 (fixed function tessellator; TS; 38), 및 프로그램 가능 도메인 쉐이더 (domain shader; DS; 40) 를 포함한다. 쉐이더 (36), 테셀레이터 (38), 및 도메인 쉐이더 (40) 는 종종 함께 "테셀레이션 파이프라인"으로 칭해진다.

[0029] 입력 어셈블러 (IA; 32) 는 베이직 프리미티브들에 대한 버텍스들 (vertices) 을 메모리로부터 판독한다. 베이직 프리미티브들은 리소스들 (50) 의 메모리 (예를 들면, 도 2의 타일 메모리 (20)) 에 저장될 수도 있다.

몇몇 예들에서, 그래픽스 파이프라인에 대한 베이직 프리미티브는 삼각형이다. 테셀레이션 스테이지들을 포함하는 그래픽스 파이프라인들의 경우, 그래픽스 파이프라인 (30) 과 마찬가지로, 베이직 프리미티브는 패치일 수도 있다. 패치는 본질적으로 메시 또는 표면의 표현이다. 패치는 표면을 정의하기 위한 1 내지 32 개의 상이한 컨트롤 포인트들을 포함할 수도 있다. 그래픽스 파이프라인의 나머지 논의는 베이직 프리미티브가 패치라고 가정할 것이지만; 그러나 본 개시의 기술들은 다른 베이직 프리미티브 타입들에 대해서도 적용가능하다.

[0030] 버텍스 쉐이더 (vertex shader; VS; 34) 는 패치의 각 컨트롤 포인트에 대한 동작들을 수행하는 프로그램 가능 스테이지이다. 버텍스 쉐이더 (34) 는 가상 공간에서의 각각의 컨트롤 포인트의 3D 포지션을, 각각의 컨트롤

를 포인트가 스크린 상에서 나타나는 2D 좌표뿐만 아니라 깊이 값으로 변환한다. 버텍스 셰이더들은 포지션, 칼라, 및 텍스처 좌표와 같은 특성들을 조작할 수 있지만, 새로운 컨트롤 포인트들을 생성하지는 않는다. 버텍스 셰이더 (34) 의 출력 컨트롤 포인트들은 헵 셰이더 (36) 로 간다.

[0031] 헵 셰이더 (36) 는 2개의 동작들을 수행한다. 대부분의 패치들에 대해, 버텍스 셰이더로부터의 변환된 컨트롤 포인트들은 단순히 도메인 셰이더 (40) 로 전달된다. 보다 복잡한 표면을 표현하는 패치들의 컨트롤 포인트들은, 도메인 셰이더 (40) 로 전달되기 이전에 헵 셰이더 (36) 에 의한 추가 프로세싱을 필요로 할 수도 있다. 또한 헵 셰이더는 각각의 패치 상에서 수행될 테셀레이션의 레벨을 결정하고 이 "테셀레이션 인자"를 테셀레이터 (38) 로 전송한다.

[0032] 테셀레이터 (38) 의 기본 기능은 3D 장면의 입력 프리미티브들 (예를 들면, 패치들) 을 다수의 출력 프리미티브들로 테셀레이팅 (즉, 세분) 하는 것이다. 테셀레이터 (38) 는 이들 패치들을 더 작은 출력 프리미티브들, 예컨대 헵 셰이더 (36) 에 의해 제공된 테셀레이션 인자에 따른 삼각형들, 쿼드들 (quads) 또는 등치선들 (isolines) 로 세분한다. 테셀레이터 (38) 의 출력은 출력 프리미티브들을 정의하는 버텍스들의 세트이다. 테셀레이션은 그래픽적으로 더 상세한 이미지들로 귀결되는 더 스무스한 곡면을 렌더링하는 것을 가능하게 한다.

[0033] 도메인 셰이더 (40) 는 테셀레이션 파이프라인의 최종 스테이지이다. 도메인 셰이더 (40) 는 테셀레이터 (38) 에 의해 생성된 출력 버텍스들에 대해 동작한다. 도메인 셰이더 (40) 는 최종 출력 버텍스 포지션들 및 헵 셰이더 (36) 로부터 수신된 컨트롤 포인트들로부터 속성들을 계산할 책임이 있다.

[0034] 지오메트리 셰이더 (geometry shader; GS; 42) 는 테셀레이션 파이프라인으로부터 수신된 테셀레이팅된 출력 프리미티브들로부터 추가적인 프리미티브들을 생성하기 위해 사용될 수도 있는 옵션적인 프로그램 가능 스테이지이다. 지오메트리 셰이더 (42) 는 그래픽적으로 한층 더 상세한 이미지들을 가능하게 한다.

[0035] 파이프라인의 나머지 스테이지들에서, 래스터라이저 (rasterizer; RS; 44) 는, 3D 공간으로부터의 지오메트리 셰이더 (42) 및/또는 테셀레이션 파이프라인으로부터의 출력 프리미티브들을 2D 공간에서의 픽셀 값들로 변환한다. 그 다음 프로그램 가능 픽셀 셰이더 (programmable pixel shader; PS; 46) 는, 깊이 테스트, 클리핑, 시저링 (scissoring), 블렌딩 등과 같은 추가 동작들을 픽셀들에 대해 수행할 수도 있다. 픽셀 셰이더 (46) 의 출력은 프레임 버퍼 (10) 에 저장되고 그 다음 디스플레이 (8) (도 2 참조) 와 같은 디스플레이 상에 디스플레이될 수도 있다.

[0036] 불필요한 GPU 프로세싱을 방지하고 타일 기반의 렌더링 아키텍처들의 제한된 그래픽스 메모리를 가장 효율적으로 사용하기 위해, 본 개시는 테셀레이션 스테이지들을 구비하는 그래픽스 파이프라인 (예를 들면, 도 3의 그래픽스 파이프라인) 을 갖는 가시성 스트림들의 사용을 제안한다. 가시성 스트림은 프리미티브가 렌더링된 타일에 기여하는지의 여부를 매 프리미티브마다 설명하는 일련의 비트들 (0/1) 이다. 즉, 가시성 스트림은, 프리미티브가 현재 렌더링된 프레임에서 보이는지 아닌지의 여부를 렌더링 파이프라인의 상이한 스테이지들에게 나타낸다. 가시성 스트림들은 타일들을 렌더링하기 이전에 실행되는 "비닝" 패스에서의 각각의 프레임의 시작에서 생성된다. 하나 이상의 가시성 스트림들은 프레임에서의 각각의 타일에 대해 생성된다. 타일 렌더링 동안, 렌더링된 타일에 기여하지 않는 무관한 프리미티브들 (즉, 이들은 보이지 않을 것이다) 을 컬링 (culling) 하기 위해, 그 타일에 대한 대응하는 가시성 스트림 (들) 이 사용된다.

[0037] 본 개시는, 테셀레이터 입력 프리미티브들 (즉, 패치들) 및 출력 프리미티브들 (즉, 삼각형들, 쿼드들, 또는 등치선들) 양자를 타일당 하나 이상의 가시성 스트림들로 인코딩하는 가시성 스트림들의 생성을 제안한다. 입력 및 출력 프리미티브들에 대한 가시성 정보를 활용함으로써, 이러한 테셀레이션 알고리즘은 (예를 들면, 입력 프리미티브 가시성 스트림을 사용하여) 현재 렌더링된 타일에 기여하지 않는 전체 입력 패치들을 컬링하고, 또한 (예를 들면, 출력 프리미티브 가시성 스트림을 사용하여) 현재 렌더링된 타일에 기여하지 않는 출력 프리미티브들의 테셀레이터 스테이지를 컬링한다. 테셀레이터 스테이지의 입력 및 출력 양자에서 이들 가시성 스트림들을 활용함으로써, 렌더링 파이프라인 (예를 들면, 도메인 셰이더 (40)) 에서의 나중의 스테이지들의 무관한 또는 중복적 호출들 (invocations) 의 수에서의 감소가 달성될 수도 있다. 본질적으로, 가시성 스트림들은, 입력 및 출력 가시성 스트림들에 의해 보이지 않는 것으로 나타내어진 입력 또는 출력 프리미티브들에 대해 렌더링 파이프라인의 하나 이상의 스테이지들이 스킵되는 것을 가능하게 한다. 추가적으로, 무관한 또는 보이지 않는 출력 프리미티브들을 프로세싱하는 것이 방지될 수 있기 때문에, 하나의 입력 패치를 테셀레이팅하기 위해 테셀레이터 하드웨어가 소비해야할 필요가 있는 클록 사이클들의 수에서의 감소가 달성된다.

- [0038] 본 개시의 일 양태에서, 2개의 가시성 스트림들은 렌더링 이전에 비닝 패스에서 타일마다 생성된다. 하나는 테셀레이션 입력 프리미티브들 (패치들) 에 대한 것이고 하나는 테셀레이션 출력 프리미티브들 (삼각형들) 에 대한 것이다. 본 개시의 다른 양태에서, 입력 및 출력 가시성 스트림들은 단일의 가시성 스트림으로 결합되거나 또는 지오메트리 셰이더와 함께 사용될 수도 있다.
- [0039] 도 4는 입력 가시성 스트림 및 출력 가시성 스트림을 생성하는 일 예의 비닝 파이프라인 (binning pipeline) 을 예시하는 블록도이다. 비닝 파이프라인은 도 2의 GPU (6) 에 의해 실행될 수도 있다. 가시성 스트림을 생성하기 위해, 비닝 파이프라인 (60) 은 전체 프레임에 대한 입력 프리미티브를 프로세싱하는 비닝 패스를 수행한다. 비닝 파이프라인은 도 3에 도시된 전체 렌더링 파이프라인의 단순화된 버전이다. 테셀레이션 스테이지들 (즉, HS/TS/DS) 및 옵션적 지오메트리 셰이더를 포함하는 그래픽스 파이프라인들에 대해, 비닝 파이프라인 (60) 은 비닝 버텍스 셰이더 (62), 비닝 헐 셰이더 (64), 비닝 테셀레이터 (66), 비닝 도메인 셰이더 (68), 옵션적 비닝 지오메트리 셰이더 (70), 비닝 래스터라이저 (72), 및 비닝 깊이 테스트 유닛 (74) 을 포함할 수도 있다.
- [0040] 비닝 버텍스 셰이더 (62) 는 비닝 셰이더로서 칭해질 수도 있다. 비닝 셰이더는 버텍스 셰이더의 수정된 버전일 수도 있는데, 여기서는 출력만이 버텍스 포지션이다. 일반적으로 버텍스 셰이더에 의해 수행되는 모든 파라미터 익스포트들 (exports) 은 비닝 셰이더뿐만 아니라 이들에 관련된 임의의 계산들로부터 제거될 수 있다.
- [0041] 비닝 버텍스 셰이더 (62) 에 의해 출력된 버텍스들로부터, 비닝 헐 셰이더 (64), 비닝 테셀레이터 (66), 및 비닝 도메인 셰이더는, 도 3의 테셀레이션 파이프라인에서와 동일한 방식으로, 테셀레이팅된 출력 프리미티브들을 생성한다. 비닝 헐 셰이더 (64) 와 비닝 테셀레이터 (66) 는 도 3의 테셀레이션 파이프라인에서의 대응하는 스테이지들에 대해 최적화될 수도 있다. 특히, 비닝 헐 셰이더 (64) 와 비닝 테셀레이터 (66) 는 포지션 관련 변수들만을 계산하도록 최적화될 수도 있다.
- [0042] 지오메트리 셰이더가 전체 렌더링 파이프라인에 대해 인에이블되면, 비닝 파이프라인 (60) 은 비닝 테셀레이터에 의해 생성된 테셀레이팅된 출력 프리미티브들로부터 추가적인 출력 프리미티브들을 생성하는 비닝 지오메트리 셰이더 (70) 를 또한 포함할 수도 있다.
- [0043] 비닝 래스터라이저 (72) 는 2D 공간들 (즉, 픽셀들) 에서의 출력 프리미티브들의 저해상도 표현을 생성한다. 각각의 픽셀은 최종 이미지에서의 픽셀 영역 (예를 들면, 4x4 픽셀 영역) 을 나타낸다. 생성된 저해상도 픽셀들은 (부분적으로 커버되거나 또는 완전히 커버되는) 2개의 값들을 가질 수 있다. 비닝 래스터라이저 (72) 는 통상의 래스터라이제이션 스테이지와 동일한 컬링 룰들 (페이스니스 (faceness), 프러스텀 (frustum) 등) 을 사용하고, 따라서 장면에 실제 보이는 그들 삼각형들만을 생성한다.
- [0044] 비닝 패스에서의 최종 스테이지는 비닝 깊이 테스트 유닛 (74) (저해상도 Z 테스트라고도 칭해짐) 이다. 비닝 깊이 테스트 유닛 (74) 은, 출력 프리미티브들로부터 래스터라이징된 픽셀들이 최종 장면에서 보이는지를 결정한다. 비닝 깊이 테스트 유닛 (74) 은, 픽셀들이 삼각형에 의해 완전히 커버되는 상황에서만 저해상도 버퍼 (도 2 참조, 저해상도 z-버퍼라고도 칭해짐) 에 기록된다. 부분적으로 커버되는 픽셀들은 저해상도 버퍼에 기록되지 않는다. 비닝 패스의 끝에서, 저해상도 버퍼 (18) 는 외부 메모리에 기록될 수 있고, 나중에, 렌더링 패스 동안 사용하기 위해 Z-버퍼를 초기화하는 데 사용될 수 있다. 이것은 (예를 들면, 도 3의 픽셀 셰이더 (46) 에서의) 렌더링 동안 향상된 깊이 테스트를 제공한다. GPU (6) 의 내부 그래픽스 메모리는 비닝 패스 동안 저해상도 버퍼로서 또한 사용될 수도 있다. 렌더링 패스들이 4x4 픽셀 블록들에서 행해지기 때문에, 그래픽스 메모리에서의 Z-버퍼도 또한 이 해상도에 있다. 또한, 그래픽스 메모리에서 어떠한 칼라 버퍼도 필요되지 않는다. 이것은, 저해상도 버퍼 (18) 가 저해상도와 비교하여 아주 큰 온스크린 영역을 커버할 수 있다는 것을 의미한다.
- [0045] 특정 타일에 대한 비닝 깊이 테스트를 통과하는 프리미티브들 (즉, 프리미티브를 래스터라이징하는 것에 의해 생성된 픽셀들이 그 타일에서 보인다) 은 가시성 스트림에서 보이는 것으로 마킹된다. 특정 타일에 대한 저해상도 Z-테스트를 통과하지 못한 프리미티브들 (즉, 프리미티브들을 래스터라이징하는 것에 의해 생성된 픽셀들이 그 타일에서 보이지 않는다) 은 가시성 스트림에서 보이지 않는 것으로 마킹된다. 대부분의 삼각형들로부터 래스터라이징된 대부분의 픽셀들이 임의의 주어진 타일에 대해 보이지 않을 것이기 때문에, 각각의 타일에 대한 가시성 스트림들은, 각각의 프리미티브들이 보이지 않는다는 것을 나타내는, 아주 많은 수의 0들로 구성될 수도 있다. 이와 같이, 가시성 스트림들은 런타임 압축 알고리즘을 사용하여 압축될 수도 있다.

- [0046] 출력 가시성 스트림 (82) 은 출력 프리미티브들 (예를 들면, 삼각형들) 의 가시성을 인코딩한다. 출력 가시성 스트림은 출력 프리미티브마다, 대응하는 타일에서 출력 프리미티브가 보이는지의 여부를 나타내는 1비트 (예를 들면, 보이는 것에 대해 1을, 보이지 않는 것에 대해 0을, 또는 그 반대인) 를 포함한다. 입력 가시성 스트림 (80) 은 입력 프리미티브들 (예를 들면, DX11에서의 패치들) 의 가시성을 인코딩한다. 비닝 깊이 테스트 유닛 (74) 에 의해 패치로부터 형성된 적어도 하나의 출력 프리미티브가 보이는 것으로 결정되면, 패치는 보이는 것으로 간주된다. 패치로부터 형성된 어떠한 출력 프리미티브들도 보이지 않으면, 입력 가시성 스트림에서 전체 패치는 보이지 않는 것으로 나타내어진다. 이 경우, 입력 가시성 스트림에서의 대응하는 비트는 0으로 설정될 수도 있다. 대안적으로, 패치로부터 형성된 적어도 하나의 출력 프리미티브가 보인다면, 전체 패치는, 예를 들면, 대응하는 비트를 1로 설정하는 것에 의해, 입력 가시성 스트림에서 보이는 것으로 나타내어질 수도 있다.
- [0047] 도 5는 일 예의 입력 가시성 스트림 및 일 예의 출력 가시성 스트림을 예시하는 개념도이다. 입력 가시성 스트림 (80) 은 각각의 패치에 대해 1들과 0들의 스트림을 포함할 수도 있는데, 하나의 비트는 패치의 적어도 일부가 최종 장면에서 보인다는 것을 나타내고 (예를 들면, 패치 3에 대한 비트 1), 다른 비트는 최종 장면에서 전체 패치가 보이지 않는다는 것을 나타낸다 (예를 들면 패치 0에 대한 비트 0). 출력 가시성 스트림 (82) 은 각각의 삼각형에 대해 1들과 0들의 스트림을 포함할 수도 있는데, 하나의 비트는 최종 장면에서 보이는 픽셀에 대해 그 삼각형이 기여한다는 것을 나타내고 (예를 들면, 삼각형 0에 대한 비트 1), 다른 비트는 최종 장면에서 보이는 픽셀에 대해 그 삼각형이 기여하지 않는다는 것을 나타낸다 (예를 들면 삼각형 2에 대한 비트 0). 도 5에 도시된 바와 같이, 출력 가시성 스트림들은 패치 단위 기준으로 생성될 수도 있다. 즉, 각각의 타일에 대한 패치마다 하나의 출력 가시성 스트림이 존재한다. 다른 예에서, 각각의 타일은 입력 프리미티브들로부터 형성된 출력 프리미티브들을 모두 함께 연결하는 하나의 출력 가시성 스트림을 가질 수도 있다.
- [0048] 도 3으로 돌아가면, 출력 및 입력 가시성 스트림들은 메모리 (예를 들면, 타일 메모리 (20)) 에 저장될 수도 있고 리소스들 (50) 을 통해 파이프라인 스테이지들을 렌더링하는 것에 이용가능하게 될 수도 있다. 입력 가시성 스트림 및 출력 가시성 스트림에 의해 나타내어진 가시성에 각각 기초하여 입력 프리미티브 또는 출력 프리미티브를 렌더링할 때, 하나 이상의 렌더링 스테이지들은 스킵될 수도 있다.
- [0049] 일 예로서, 출력 가시성 스트림을 활용하면, 테셀레이터 (38) 는 아주 많은 수의 삼각형들이 현재 렌더링된 타일에서 보이지 않거나 현재 렌더링된 타일에 로케이팅되지 않는 경우 그 아주 많은 수의 삼각형들을 스킵할 수 있다. 마찬가지로, 패치가 보인다는 것을 입력 가시성 스트림이 나타내면, 그 패치는 렌더링 패스 동안 현재 타일에 대해 프로세싱되고, 그렇지 않으면 그 패치는 컬링된다 (즉, 그 패치에 대한 렌더링이 스킵된다).
- [0050] 특히, 보이지 않는 입력에 대해 렌더링을 스킵하는 것은, 입력 프리미티브가 보이지 않는다는 것을 입력 가시성 스트림이 나타낼 때 전체 입력 프리미티브의 테셀레이션을 방지하는 것을 수반할 수도 있다. 이와 같이, 출력 프리미티브들은 그 입력 프리미티브에 대해 생성되지 않고, 따라서 파이프라인의 추가 스테이지들에 의한 프로세싱에 이용가능하지 않다. 마찬가지로, 보이지 않는 출력 프리미티브들에 대해 렌더링을 스킵하는 것은, 보이지 않는 출력 프리미티브들을 생성하게 될 패치의 특정 부분에 대해 테셀레이션을 스킵하는 것을 수반할 수도 있다. 사용되는 스킵 프로세스는 테셀레이션 하드웨어에 의존적일 수도 있다. 도 6 및 도 7을 참조로 하기에 일 예가 설명된다. 도 6 및 도 7은 입력 프리미티브를 테셀레이팅하는 일 예의 기술을 설명하는 개념도이다.
- [0051] 일반적으로, 입력 패치 테셀레이션은 입력 삼각형 내부에 더 작은 삼각형을 추가하는 것, 및 두 삼각형들 쌍 단위 (wise) 로부터의 한 에지 (즉, 동일 방향에 있는 더 큰 삼각형으로부터의 한 에지와 더 작은 삼각형으로부터의 한 에지) 를 고려하는 것에 의해 2개의 삼각형들 사이의 영역을 테셀레이팅하는 것을 수반한다. 도 6에 도시된 바와 같이, 테셀레이션은 에지 쌍 (83a 및 83b), 에지 쌍 (84a 및 84b), 및 에지 쌍 (85a 및 85b) 사이에서 발생할 수도 있다. 3개의 에지 쌍들에 대해 테셀레이션이 완료된 후, 이전의 더 작은 삼각형 내부에 새로운 더 작은 입력 삼각형이 삽입된다 (도 7 참조). 도 7에 도시된 바와 같이, 에지들 (83b, 84b, 및 85b) 은 이제 에지들 (83c, 84c, 및 85c) 과 각각 쌍을 이루게 된다. 이 프로세스는 입력 삼각형이 완전히 테셀레이팅될 때까지 계속된다. 테셀레이션의 양 (즉, 테셀레이션 레벨) 은 힐 셰이더 (36) (도 3 참조) 에 의해 생성된 테셀레이션 인자에 의해 정의된다.
- [0052] 상기 예시적인 테셀레이션으로부터 알 수 있는 바와 같이, 테셀레이션 프로세스는 한 번에 하나의 에지 쌍만큼 진행한다. 출력 가시성 스트림으로부터 삼각형 가시성 정보를 활용함으로써, 향상된 테셀레이터 구현에는, 현재 테셀레이팅된 영역에 대한 출력 삼각형들의 어느 것도 현재 렌더링된 타일에서 보이지 않게 될 경우 전체 에

지 쌍들의 프로세싱을 스킵할 수 있다. 예를 들면, 에지 쌍 (85b 및 85c) 사이의 영역은 출력 가시성 스트림에 의해 보이지 않는 것으로서 나타내어진 출력 프리미티브들을 포함할 수도 있다. 이와 같이, 테셀레이션은 이 에지 쌍에 대해 스킵될 수도 있고, 도 7에 도시된 바와 같이, 어떠한 출력 삼각형들도 생성되지 않는다. 단일의 테스트로 여러 에지 쌍들을 스킵하는 것도 또한 가능한데, 그 이유는, 하나의 삼각형 쌍에 대해, 셋업 페이즈 이후에, 할 셰이더로부터 수신된 테셀레이션 인자에 기초하여, 두 삼각형들 사이의 영역에 얼마나 많은 삼각형들이 생성될 것인지를 알기 때문이다 (다수의 에지 쌍들이 병렬로 또는 잠재적으로 함께 고려될 수 있다).

[0053] 본 개시의 다른 예에서, 입력 프리미티브들에 대해 가시성 스트림을 출력 프리미티브들에 대해 별도의 가시성 스트림을 생성하는 대신, 단일의 가시성 스트림이 생성된다. 이러한 경우들에서, 단일의 가시성 스트림은 최대 테셀레이션 출력에 대해 인코딩될 것이다. 즉, 가시성 스트림의 길이는, 특정 렌더링 파이프라인에 대해 정의된 패치 사이즈 및 최대 가능한 테셀레이션 인자가 주어지면, 테셀레이팅된 프리미티브들의 최대 양에 의해 정의될 것이다. 예를 들면, 최대 테셀레이션 출력은 다이렉트X11 API에 의해 정의될 수도 있다. 단일의 가시성 스트림은 각각의 타일에 대한 패치 단위 기준으로 출력 프리미티브들의 가시성을 나타낼 것이다. 패치에 대한 단일의 가시성 스트림에서의 하나의 출력 프리미티브가 보이면, 그 패치도 또한 보인다. 각각의 패치에 대한 가시성 스트림은 길이 N을 가질 것인데, 여기서 N은 최대 테셀레이션 출력 (즉, 가능한 출력 삼각형들의 최대 수)에 대응한다.

[0054] 도 8은 단일의 가시성 스트림을 생성하는 일 예의 비닝 파이프라인을 예시하는 블록도이다. 비닝 파이프라인 (90)은 도 4의 비닝 파이프라인 (60)과 유사한 방식으로 동작할 것이다. 그러나, 2개의 가시성 스트림들을 생성하는 대신, 비닝 파이프라인 (90)은 단일의 가시성 스트림 (104)을 생성한다. 디폴트로, 전체 패치에 대한 가시성 스트림 (104)에서의 각각의 비트는 0들로 인코딩될 것이다 (즉, 테셀레이터의 어떤 출력 삼각형들도 보이지 않는다고 가정된다). 임의의 테셀레이터 출력 삼각형들이 비닝 깊이 테스트 유닛 (74)에 의해 현재 렌더링된 타일에서 보인다고 알려지면, 가시성 스트림으로부터의 대응하는 비트들은 1로 설정된다. 타일들이 렌더링되는 경우, 임의의 가시성 비트들이 단일의 가시성 스트림에서의 단일의 패치 영역 내부에서 1일 것인지를 결정하기 위해 가시성 스트림들은 단순히 체크될 수도 있다. 패치 내부의 모든 값들이 0이면, 이것은, 예를 들면 하나의 클록 사이클에서 결정될 수 있다. 패치에 대한 모든 가시성 비트들이 0이면, 버텍스 셰이더 및 테셀레이션 파이프라인 양자는 그 패치에 대해 스킵될 수도 있다. 가시성 스트림에서의 임의의 가시성 비트들이 1이면, 패치는 정상적으로 프로세싱된다. 그러나, 이제 테셀레이터는, 가시성 스트림에 의해 보이지 않는 것 (0들)으로 정의된 어떤 출력 프리미티브들의 테셀레이션을 스킵하도록 구성될 수도 있다. 이와 같이, 보이지 않는 패치들 및 삼각형들을 스킵하는 것에 의해, 버텍스 셰이더, 테셀레이션 파이프라인뿐만 아니라, 래스터라이저에 대해 불필요한 프로세싱이 방지된다.

[0055] 도 9는 일 예의 단일의 가시성 스트림을 예시하는 개념도이다. 가시성 스트림 (104)은 길이 N으로 이루어지는데, 여기서 N은 렌더링 파이프라인에 의해 정의된 바와 같은 패치당 출력 프리미티브들의 최대 수를 정의한다.

[0056] 도 8로 돌아가면, 본 개시의 다른 예에서, 지오메트리 셰이더가 인에이블되면 단일의 가시성 스트림이 또한 사용될 수도 있다. 위에서 언급된 바와 같이, 지오메트리 셰이더는 테셀레이션 파이프라인에 의해 출력된 프리미티브들로부터 더 많은 프리미티브들을 생성할 수도 있다. 단일의 지오메트리 셰이더 호출이 생성할 수 있는 데이터의 최대량은 DX11 API에 의해 4kB로 제한된다. 위에서 설명된 최대 테셀레이션 출력에 기초하여 입력 및 출력 가시성 스트림들을 함께 병합하는 것과 유사하게, 지오메트리 셰이더가 인에이블된 단일의 출력 가시성 스트림은, 지오메트리 셰이더가 생성한 프리미티브들의 최대 수에 의해 승산된 최대 수의 테셀레이팅된 프리미티브들을 인코딩할 것이다. 즉, 테셀레이션 출력 프리미티브들의 최대 수가 N이고, 지오메트리 셰이더로부터의 출력 프리미티브들의 최대 수가 M이면, 지오메트리 셰이더가 인에이블되는 경우의 가시성 스트림의 길이는 NxM이 될 것이다. 이 예에서, 비닝 지오메트리 셰이더 (70)는, 지오메트리 셰이더에 의해 생성된 프리미티브들이 최종 장면에서 보이는지를 비닝 깊이 테스트 유닛 (74)이 결정할 수도 있도록 인에이블될 것이다.

[0057] 다시, 이 단일의 가시성 스트림에 대한 디폴트 값들은 0들이다. 임의의 지오메트리 셰이더 출력 삼각형들이 비닝 깊이 테스트 유닛 (74)에 의한 비닝 패스 동안 현재 렌더링된 타일에서 보인다고 결정되면, 그것의 대응하는 가시성 비트는 1로 설정될 것이다. 타일이 렌더링될 때, 단일의 가시성 스트림에서의 단일의 패치 영역 내부에서 임의의 가시성 비트들이 1로 설정되는지를 결정하기 위해 가시성 스트림이 체크된다. 패치 내부의 모든 값들이 0이면, 이것은 하나의 클록 사이클에서 결정될 수 있다. 대응하여, 도메인 셰이더 (40)

또는 지오메트리 셰이더 (42) 에 의한 프로세싱을 위해 현재 테셀레이팅된 삼각형이 생성되어 전송될 필요가 있을 것인지의 여부를 결정하기 위해 임의의 지오메트리 셰이더 출력 프리미티브들이 보였다면, 테셀레이터 (38) (도 3 참조) 는 매 (every) 지오메트리 셰이더 출력 가시성 비트 범위 (범위 사이즈는 지오메트리 셰이더 출력 프리미티브들의 최대 수이다) 를 체크할 것이다.

[0058] 도 9로 돌아가면, 테셀레이션 스테이지뿐만 아니라 인에이블된 지오메트리 셰이더를 구비하는 그래픽스 파이프라인에 대해 단일의 가시성 스트림 (104') 이 사용된다. 단일의 가시성 스트림 (104') 은 길이 NxM으로 이루어지는데, 여기서 N은 렌더링 파이프라인에 의해 정의된 것과 같은 패치마다의 출력 프리미티브들의 최대 수를 정의하고 M은 지오메트리 셰이더에 의해 생성될 수도 있는 출력 프리미티브들의 최대 수를 정의한다.

[0059] 도 10은 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하는 일 예의 방법을 예시하는 흐름도이다. 일 예로서, 이 방법은 도 2의 GPU (6) 에 의해 수행될 수도 있다. GPU (6) 는 프레임의 각각의 타일에 대한 입력 가시성 스트림을 생성하도록 (110), 그리고 프레임의 각각의 타일에 대한 출력 가시성 스트림을 생성하도록 (112) 구성될 수도 있다. 입력 가시성 스트림은, 입력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타낸다. 출력 가시성 스트림은, 출력 프리미티브가 렌더링시 각각의 타일에서 보이는지 아닌지의 여부를 나타낸다. 출력 프리미티브는 입력 프리미티브를 테셀레이팅하는 것에 의해 생성된다.

[0060] GPU (6) 는 프레임의 현재 타일을 렌더링하도록 더 구성될 수도 있는데, 여기서 렌더링은 입력 가시성 스트림 및 출력 가시성 스트림에 각각 기초하여 입력 프리미티브들 및 출력 프리미티브들에 대해 스킵된다 (114). 특히, 출력 프리미티브들에 대한 렌더링을 스킵하는 것은, 출력 가시성 스트림에 의해 현재 프레임에서 보이지 않는 것으로 나타내어진 출력 프리미티브들에 대해 테셀레이션을 스킵하는 것을 포함할 수도 있다.

[0061] 입력 가시성 스트림을 생성하고 (110) 그리고 출력 가시성 스트림을 생성하는 (112) 단계들은 비닝 패스 (116) 에서 함께 수행될 수도 있다. 도 11은 타일 기반의 렌더링 파이프라인에서 비닝 패스의 일 예의 방법을 예시하는 흐름도이다. GPU (6) 는 비닝 패스 (116) 를 수행하도록 구성될 수도 있다. GPU (6) 는, 입력 프리미티브를 벡터 셰이딩하고 (118), 복수의 출력 프리미티브들을 생성하기 위해 벡터 셰이딩된 입력 프리미티브를 테셀레이팅하고 (120), 복수의 출력 프리미티브들을 래스터라이징하고 (122), 프레임의 타일에 대해 출력 프리미티브들이 보이는지를 결정하기 위해 래스터라이징된 출력 프리미티브들에 대한 깊이 테스트를 수행하도록 (124) 더 구성될 수도 있다.

[0062] GPU (6) 는, 깊이 테스트에 기초하여 입력 프리미티브가 타일에서 보이는지 아닌지의 여부의 표시 (indication) 를 입력 가시성 스트림에서 생성하는데, 여기서 입력 프리미티브는, 그 입력 프리미티브를 테셀레이팅하는 것에 의해 생성된 임의의 출력 프리미티브들이 보이는 것으로 결정되면 보이는 것으로 결정된다 (126). 마찬가지로, GPU (6) 는 깊이 테스트에 기초하여 출력 프리미티브가 타일에서 보이는지 아닌지의 표시를 출력 가시성 스트림에서 생성한다 (128).

[0063] 도 12는 타일 기반의 렌더링 파이프라인에서 테셀레이션을 수행하는 다른 예의 방법을 예시하는 흐름도이다. 이 예에서, 단일의 가시성 스트림은, 입력 프리미티브로부터 형성될 수도 있는 출력 프리미티브들의 가시성을 나타내는 입력 프리미티브마다 생성된다. GPU (6) 는, 입력 프리미티브를 벡터 셰이딩하고 (140), 복수의 출력 프리미티브들을 생성하기 위해 벡터 셰이딩된 입력 프리미티브를 테셀레이팅하고 (142), 복수의 출력 프리미티브들을 래스터라이징하고 (144), 프레임의 타일에 대해 출력 프리미티브들이 보이는지를 결정하기 위해 래스터라이징된 출력 프리미티브들에 대한 깊이 테스트를 수행하도록 (146) 구성될 수도 있다.

[0064] GPU (6) 는 깊이 테스트에 기초하여 출력 프리미티브가 타일에서 보이는지 아닌지의 표시를 가시성 스트림에서 생성하도록 더 구성될 수도 있다 (148). 가시성 스트림은 N 비트들의 길이를 가질 수도 있는데, 각 비트는 렌더링시 각 타일에서 출력 프리미티브가 보이는지 아닌지의 여부를 나타낸다. 길이 N은, 렌더링 파이프라인에 의해 정의된 바와 같은 입력 프리미티브마다의 출력 프리미티브들의 최대 수에 의해 결정된다. GPU (6) 는 프레임의 현재 타일을 렌더링하도록 또한 구성될 수도 있는데, 여기서 가시성 스트림에서의 매 비트가 보이지 않는 출력 프리미티브를 나타내는 경우, 입력 프리미티브들에 대해 렌더링이 스킵되고, 가시성 스트림에 의해 현재 프레임에서 보이지 않는 것으로 나타내어진 출력 프리미티브들에 대해 렌더링이 스킵된다 (150).

[0065] 도 13은 타일 기반의 렌더링 파이프라인에서 비닝 패스의 다른 예의 방법을 예시하는 흐름도이다. 도 13에 도시된 프로세스는, 지오메트리 단계 (152) 의 추가를 제외하면 도 12에 도시된 것과 동일하다. GPU (6) 가 인에이블된 지오메트리 셰이더를 가지면, 복수의 출력 프리미티브들의 지오메트리 셰이딩을 수행하여 추가적인 출력 프리미티브들을 생성하는 추가적 단계 (152) 가 테셀레이션 단계 (142) 와 래스터라이제이션 단계 (144)

사이에서 수행된다. 이 예에서, 가시성 스트림은 NxM 비트들의 증가된 길이를 가질 것인데, 여기서 M은 지오메트리 셰이더에 의해 생성된 출력 프리미티브들의 최대 수이다.

[0066] 하나 이상의 실시형태들에서, 상술된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 상기 기능들은 비일시적 컴퓨터 판독가능 매체를 포함하는 제조물품 상의 하나 이상의 명령들 또는 코드로서 저장될 수도 있다. 컴퓨터 판독가능 매체들은 컴퓨터 데이터 저장 매체들을 포함할 수도 있다. 데이터 스토리지 매체들은 본 개시에서 설명된 기술들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용 가능한 매체들일 수도 있다. 비제한적인 예로서, 이러한 컴퓨터 판독가능 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기적 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 이송 또는 저장하기 위해 사용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 본원에서 사용된 바와 같이, 디스크 (disk) 와 디스크 (disc) 는 콤팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하며, 디스크 (disk) 는 일반적으로 데이터를 자기적으로 재생하고, 디스크 (disc) 는 데이터를 레이저들에 의해 광학적으로 재생한다. 위의 조합들도 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

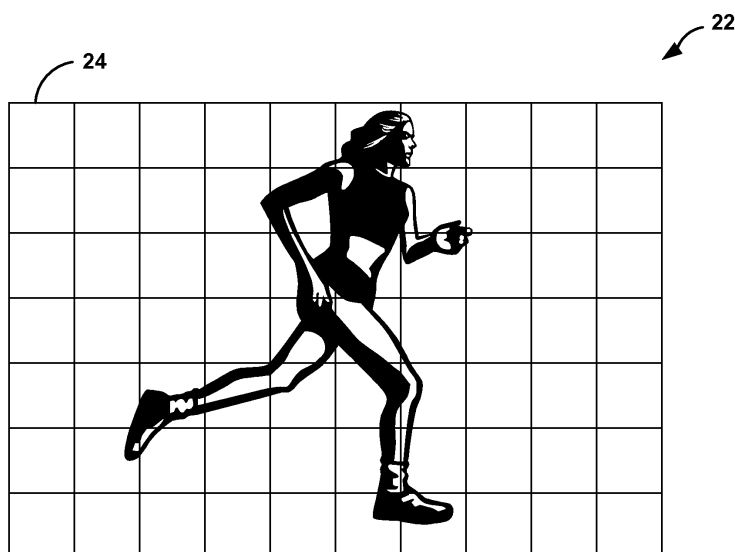
[0067] 하나 이상의 DSP들, 범용 마이크로프로세서들, ASIC들, FPGA들, 또는 다른 등가의 집적 또는 이산 로직 회로부와 같은 하나 이상의 프로세서들에 의해 코드가 실행될 수도 있다. 또한, 몇몇 양태들에서, 본원에서 설명된 기능성은 전용 하드웨어 및/또는 소프트웨어 모듈들 내에서 제공될 수도 있다. 또한, 본 기술들은 하나 이상의 회로들 또는 로직 소자들에서 완전히 구현될 수 있다.

[0068] 본 개시의 기술들은, 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예를 들면, 칩셋) 를 포함하는 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 개시된 기술들을 수행하도록 구성된 디바이스들의 기능적 양태를 강조하기 위해 다양한 소자들, 모듈들, 또는 유닛들이 본 개시에서 설명되었지만, 반드시 상이한 하드웨어 유닛들에 의해 실현될 필요는 없다. 대신, 상술한 바와 같이, 다양한 유닛들은, 적절한 소프트웨어 및/또는 펌웨어와 연계하여, 코덱 하드웨어 유닛에서 통합되거나 또는 상술한 하나 이상의 프로세서들을 포함하는 상호동작적인 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

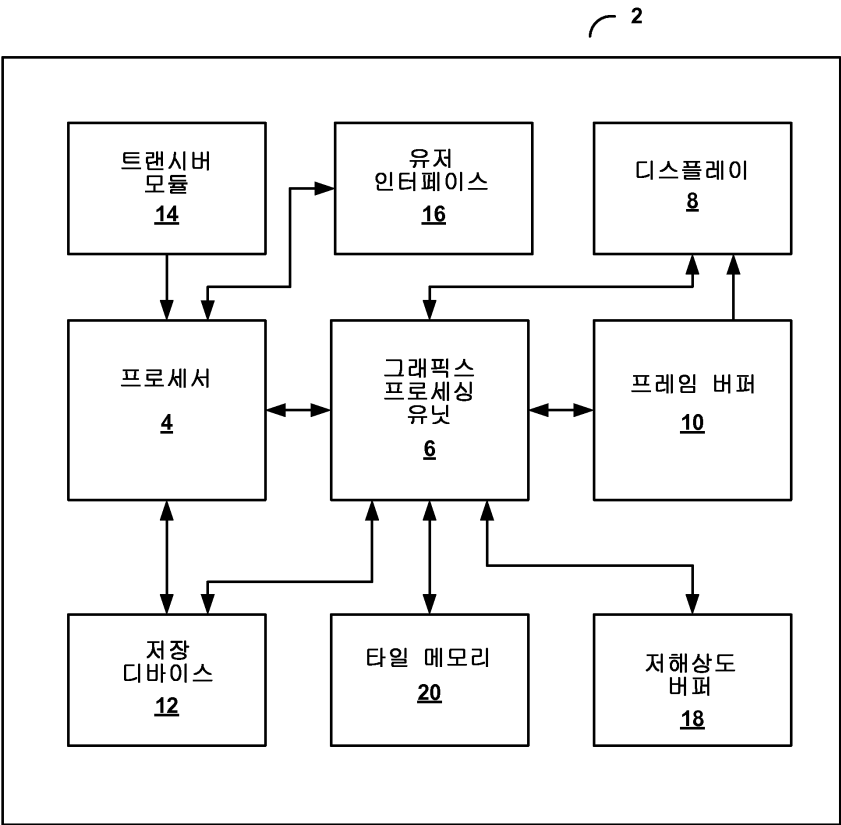
[0069] 다양한 예들을 설명하였다. 이들 및 다른 예들은 하기의 특허청구범위 내에 있다.

## 도면

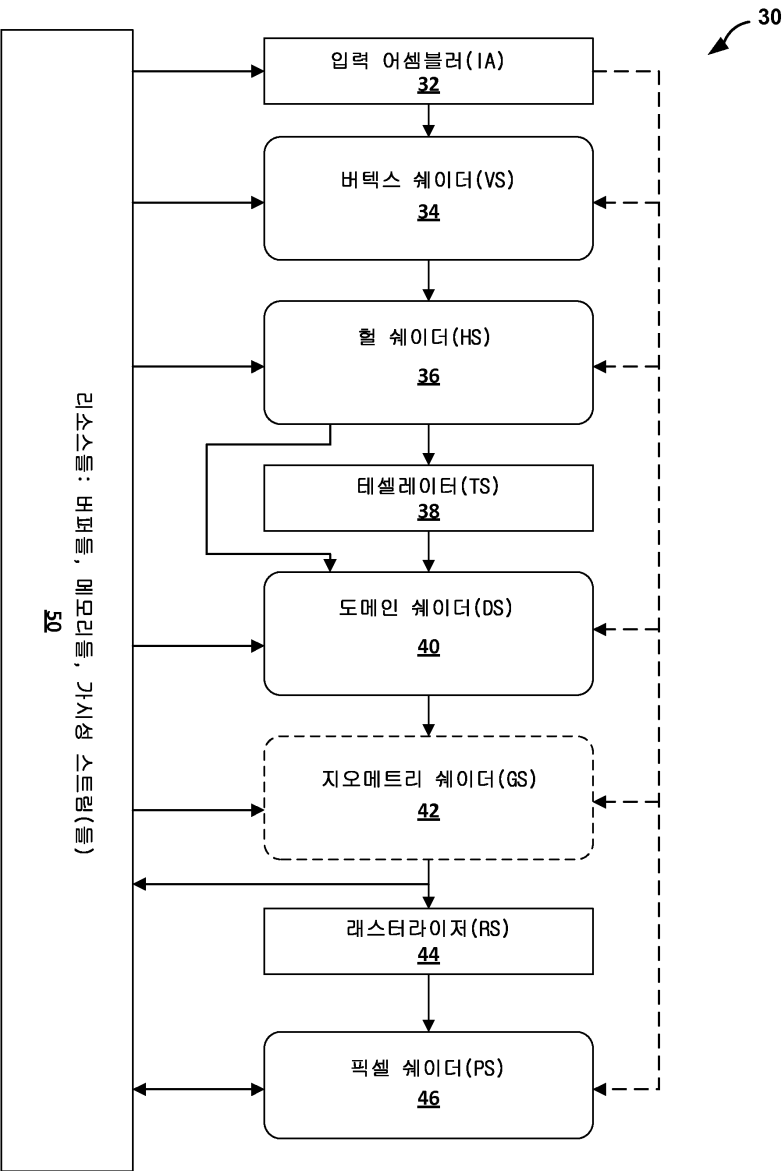
### 도면1



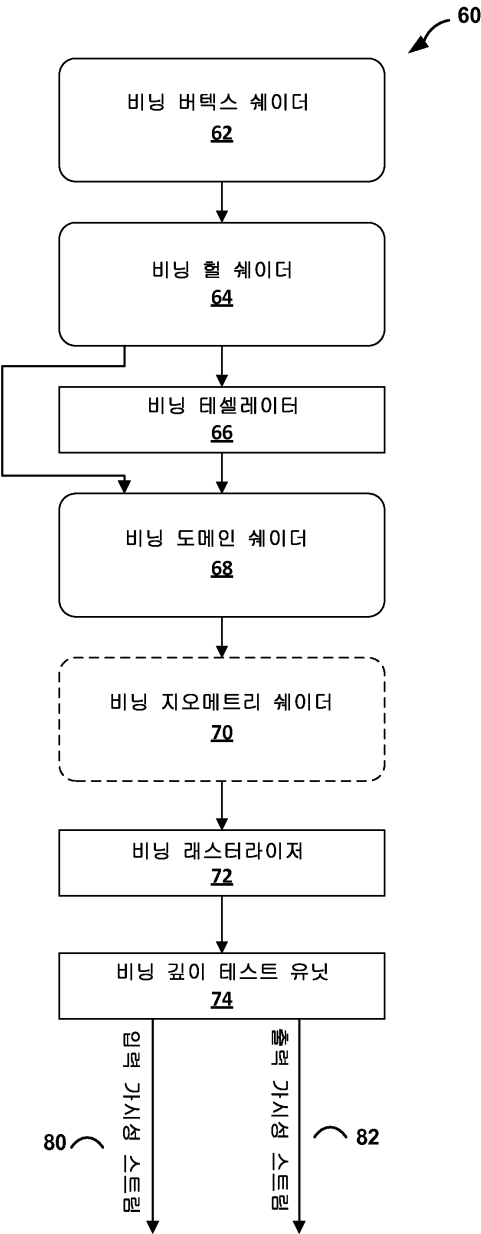
도면2



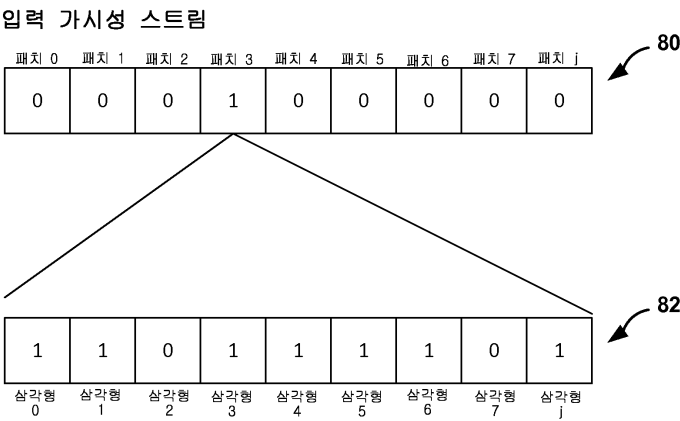
도면3



도면4

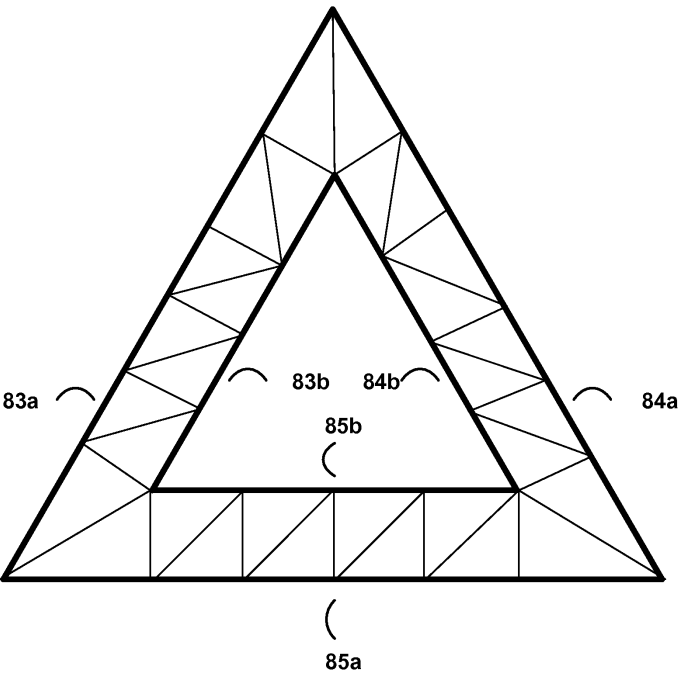


도면5

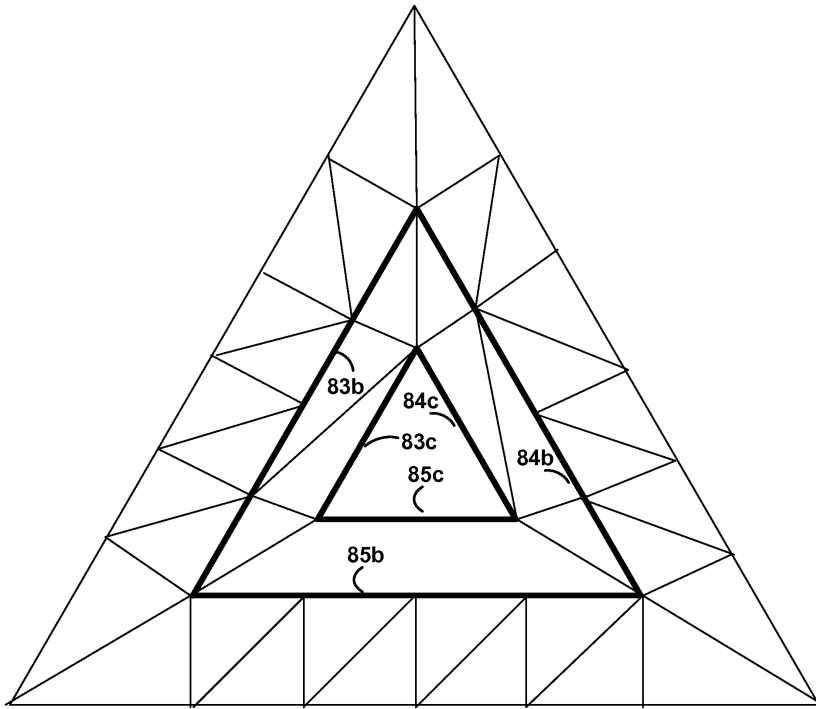


출력 가시성 스트림

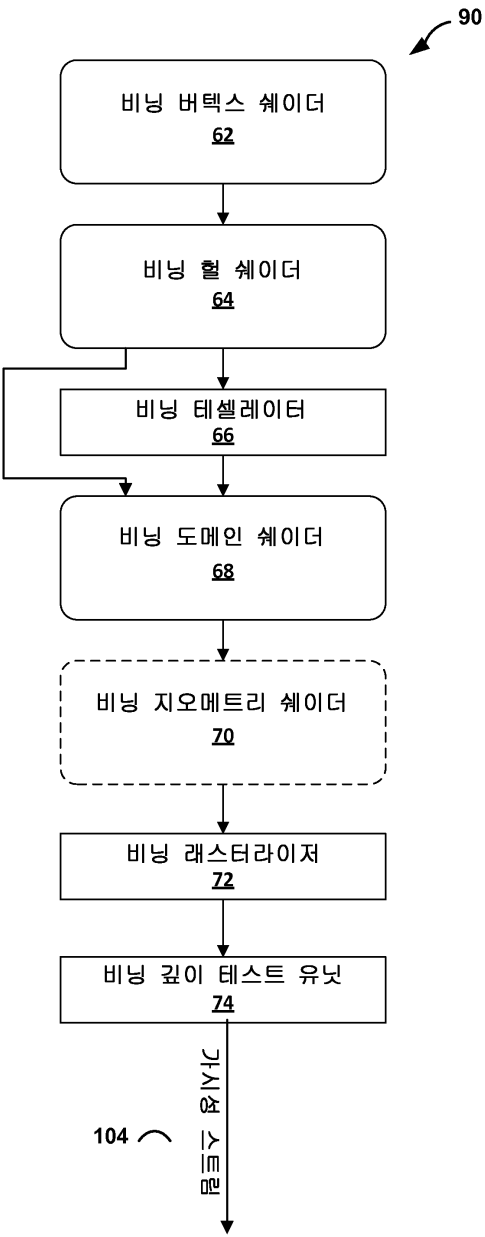
도면6



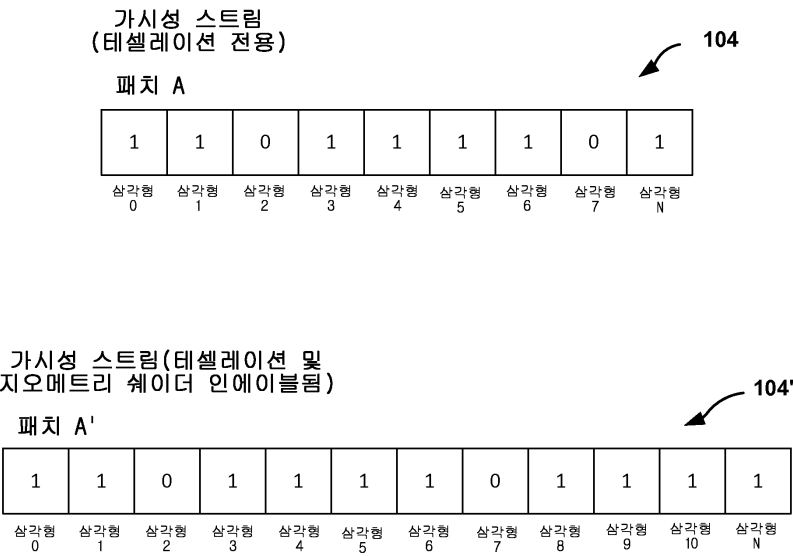
도면7



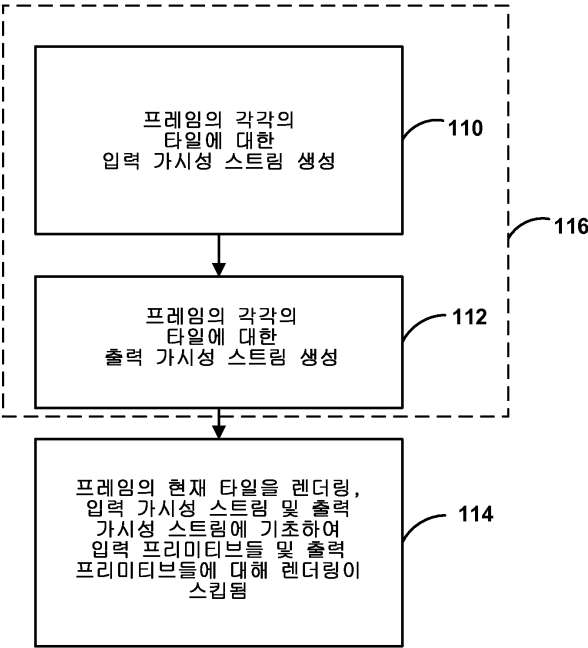
도면8



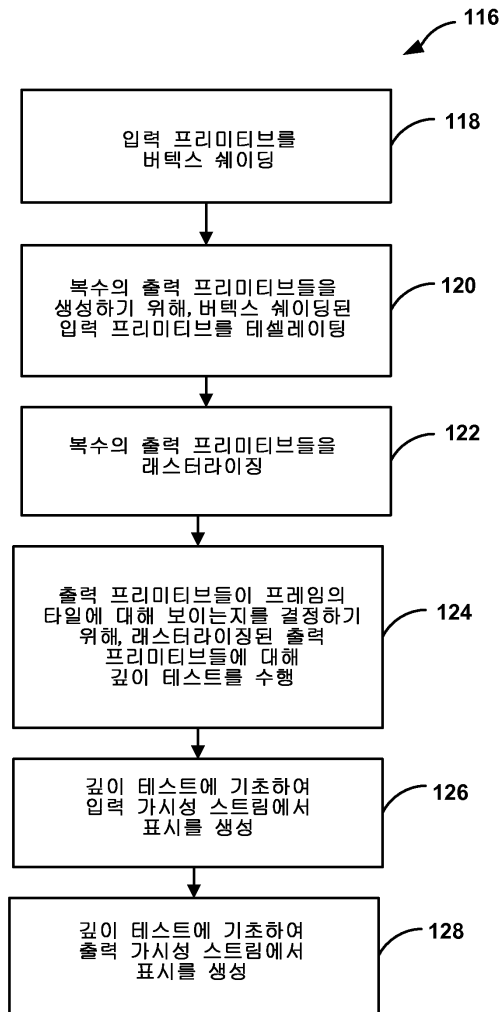
도면9



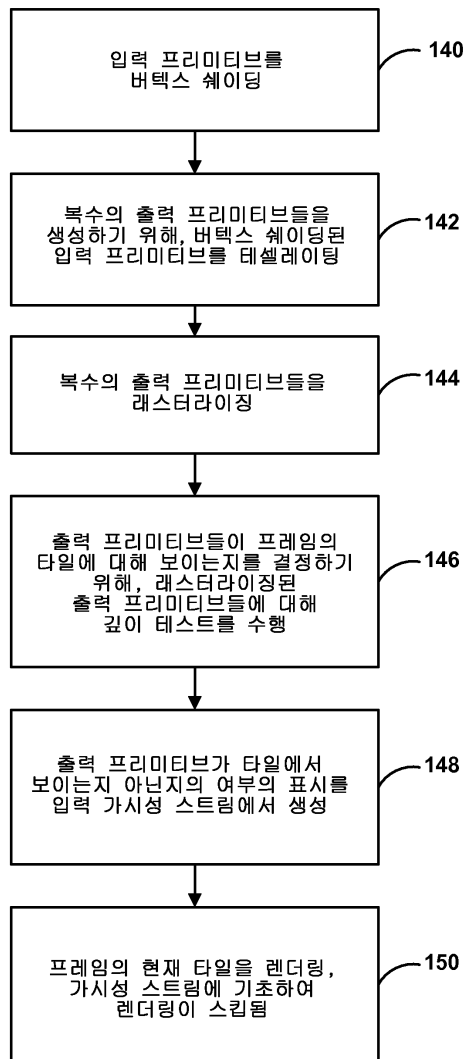
도면10



도면11



도면12



도면13

