



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년05월28일

(11) 등록번호 10-1523888

(24) 등록일자 2015년05월21일

(51) 국제특허분류(Int. Cl.)

G09G 5/36 (2006.01)

(21) 출원번호 10-2014-7003599

(22) 출원일자(국제) 2012년06월12일

심사청구일자 2014년02월11일

(85) 번역문제출일자 2014년02월11일

(65) 공개번호 10-2014-0039068

(43) 공개일자 2014년03월31일

(86) 국제출원번호 PCT/US2012/042089

(87) 국제공개번호 WO 2013/009421

국제공개일자 2013년01월17일

(30) 우선권주장

13/181,300 2011년07월12일 미국(US)

(56) 선행기술조사문헌

US20020163523 A1

US20060012714 A1

(73) 특허권자

퀄컴 인코포레이티드

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자

라비 코스로 웹

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(74) 대리인

특허법인코리아나

전체 청구항 수 : 총 34 항

심사관 : 김민수

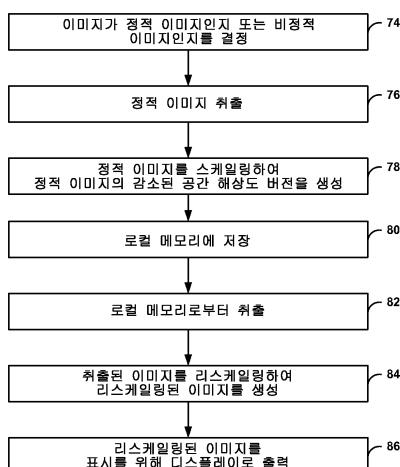
(54) 발명의 명칭 정적 이미지들의 디스플레이

### (57) 요 약

본 개시의 양태들은 감소된 소비 전력으로 정적 이미지를 디스플레이하는 기술들을 설명한다. 몇몇 양태들에서, 그래픽 프로세싱 유닛 (GPU)은 시스템 메모리로부터 정적 이미지를 취출하고, 정적 이미지의 감소된 공간 해상도 버전으로 정적 이미지를 스케일링하고, 정적 이미지의 감소된 공간 해상도 버전을 로컬 메모리에 저장한다. 디스플레이 프로세서는 로컬 메모리로부터 정적 이미지의 감소된 공간 해상도 버전을 취출할 수도 있다.

디스플레이 프로세서는 정적 이미지의 감소된 공간 해상도 버전을 리스케일링하고, 리스케일링된 이미지를 표시를 위해 디스플레이 상에 디스플레이할 수도 있다.

**대 표 도** - 도4



## 명세서

### 청구범위

#### 청구항 1

시스템 버스를 통해 액세스 가능한 시스템 메모리의 적어도 일부에 저장된 이미지가 정적 이미지 (static image) 인지 또는 비정적 이미지 (non-static image) 인지를 결정하는 단계;

상기 이미지가 상기 정적 이미지인 것으로 결정되면 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터, 그래픽 프로세싱 유닛 (GPU)에 의해, 상기 정적 이미지를 취출하는 단계;

상기 정적 이미지의 감소된 공간 해상도 버전을 생성하기 위해, 로컬 메모리에서의 이용 가능한 저장 공간의 양에 기초하여, 상기 GPU에 의해, 상기 정적 이미지를 스케일링하는 단계;

상기 시스템 메모리 외부에 있는 상기 GPU의 상기 로컬 메모리에, 상기 GPU에 의해 그리고 상기 시스템 버스를 사용하지 않고, 상기 정적 이미지의 상기 감소된 공간 해상도 버전을 저장하는 단계;

상기 로컬 메모리로부터 상기 정적 이미지의 상기 감소된 공간 해상도 버전을, 디스플레이에 커플링된 디스플레이 프로세서에 의해 그리고 상기 시스템 버스를 사용하지 않고, 취출하는 단계;

상기 정적 이미지의 상기 감소된 공간 해상도 버전을, 상기 디스플레이 프로세서에 의해, 리스케일링하여 리케일링된 이미지를 생성하는 단계; 및

상기 디스플레이 프로세서에 의해, 표시를 위해 상기 디스플레이에 상기 리스케일링된 이미지를 출력하는 단계;

및

상기 이미지가 상기 비정적 이미지인 것으로 결정되면, 상기 디스플레이 상에서의 표시를 위해 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터, 상기 디스플레이 프로세서에 의해, 상기 비정적 이미지를 취출하는 단계를 포함하는, 방법.

#### 청구항 2

삭제

#### 청구항 3

삭제

#### 청구항 4

제 1항에 있어서,

상기 로컬 메모리는 상기 GPU의 온칩 메모리를 포함하는, 방법.

#### 청구항 5

제 1항에 있어서,

상기 이미지가 상기 비정적 이미지인 것으로 결정되면, 상기 디스플레이 상에서의 표시를 위해 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터 상기 비정적 이미지를 제 1의 리프레시 레이트에서 반복적으로 취출하는 단계를 더 포함하고,

상기 리스케일링된 이미지를 출력하는 단계는, 상기 제 1의 리프레시 레이트보다 더 낮은 제 2의 리프레시 레이트에서 상기 리스케일링된 이미지를 반복적으로 출력하는 단계를 포함하는, 방법.

#### 청구항 6

제 1항에 있어서,

상기 시스템 메모리의 상기 일부에 저장된 이미지가 정적 이미지인지 또는 비정적 이미지인지를 결정하는 단계

는, 상기 이미지를 저장하는 상기 시스템 메모리의 상기 일부가 정의된 기간 내에 어떤 새로운 컨텐츠도 수신하는 않는 경우 상기 이미지가 상기 정적 이미지라고 결정하는 단계를 포함하는, 방법.

### 청구항 7

제 6항에 있어서,

주변 라이팅에서의 임의의 변화, 상기 GPU 및 상기 디스플레이 프로세서를 포함하는 디바이스의 방위 (orientation)에서의 변화들, 또는 상기 디바이스와 다른 디바이스 사이의 접속에서의 변화들이 있는지의 여부를 결정하는 단계를 더 포함하고,

상기 이미지가 상기 정적 이미지라고 결정하는 단계는, 상기 이미지를 저장하는 상기 시스템 메모리의 상기 일부가 상기 정의된 기간 내에 어떠한 새로운 컨텐츠도 수신하지 않는 경우 그리고 상기 주변 라이팅, 상기 디바이스의 상기 방위, 또는 상기 디바이스와 상기 다른 디바이스 사이의 상기 접속에서 상기 정의된 기간 내에 어떠한 변화도 없는 경우 상기 이미지가 상기 정적 이미지라고 결정하는 단계를 포함하는, 방법.

### 청구항 8

제 1항에 있어서,

상기 리스케일링된 이미지를 출력할 때 상기 디스플레이의 조도를 감소시키는 단계를 더 포함하는, 방법.

### 청구항 9

제 1항에 있어서,

상기 정적 이미지를 스케일링하는 단계는, 상기 정적 이미지의 상기 감소된 공간 해상도 버전을 생성하기 위해 상기 정적 이미지를 상기 GPU의 쉐이더에 의해 스케일링하는 단계를 포함하는, 방법.

### 청구항 10

제 1항에 있어서,

상기 리스케일링된 이미지의 선명도는 상기 이미지의 선명도보다 낮은, 방법.

### 청구항 11

디스플레이;

시스템 버스;

상기 시스템 버스를 통해 액세스 가능한 시스템 메모리;

상기 시스템 메모리 외부에 있는 로컬 메모리;

상기 시스템 메모리의 적어도 일부에 저장된 이미지가 정적 이미지인지 또는 비정적 이미지인지 결정하도록 구성된 하나 이상의 프로세싱 유닛들;

그래픽 프로세싱 유닛 (GPU) 으로서:

상기 이미지가 상기 정적 이미지인 것으로 결정되면 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터 상기 정적 이미지를 취출하고;

상기 정적 이미지의 감소된 공간 해상도 버전을 생성하기 위해, 상기 로컬 메모리에서의 이용 가능한 저장 공간의 양에 기초하여, 상기 정적 이미지를 스케일링하고; 그리고

상기 정적 이미지의 상기 감소된 공간 해상도 버전을, 상기 시스템 버스를 사용하지 않고 상기 로컬 메모리에 저장하도록 구성된,

상기 그래픽 프로세싱 유닛 (GPU); 및

디스플레이 프로세서로서:

상기 정적 이미지의 상기 감소된 공간 해상도 버전을, 상기 시스템 버스를 사용하지 않고 상기 로컬 메

모리로부터 취출하고;

상기 정적 이미지의 상기 감소된 공간 해상도 버전을 리스케일링하여 리스케일링된 이미지를 생성하고;  
그리고

표시를 위해 상기 디스플레이에 상기 리스케일링된 이미지를 출력하도록 구성된,  
상기 디스플레이 프로세서를 포함하며,  
상기 디스플레이 프로세서는 상기 디스플레이 상에서의 표시를 위해 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터 상기 비정적 이미지를 취출하도록 구성되는, 장치.

#### 청구항 12

삭제

#### 청구항 13

삭제

#### 청구항 14

제 11항에 있어서,

상기 로컬 메모리는 상기 GPU의 온칩 메모리를 포함하는, 장치.

#### 청구항 15

제 11항에 있어서,

상기 이미지가 상기 비정적 이미지인 것으로 결정되면, 상기 디스플레이 프로세서는 상기 디스플레이 상에서의 표시를 위해 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터 상기 비정적 이미지를 제 1의 리프레시 레이트에서 반복적으로 취출하고, 상기 디스플레이 프로세서는 상기 제 1의 리프레시 레이트보다 더 낮은 제 2의 리프레시 레이트에서 상기 리스케일링된 이미지를 반복적으로 출력하는, 장치.

#### 청구항 16

제 11항에 있어서,

상기 하나 이상의 프로세싱 유닛들은, 상기 이미지를 저장하는 상기 시스템 메모리의 상기 일부가 정의된 기간 내에 어떤 새로운 컨텐츠도 수신하지 않는 경우 상기 이미지가 상기 정적 이미지라고 결정하는, 장치.

#### 청구항 17

제 16항에 있어서,

상기 하나 이상의 프로세싱 유닛들은 주변 라이팅에서의 임의의 변화, 상기 장치의 방위에서의 변화들, 또는 상기 장치와 다른 디바이스 사이의 접속에서의 변화들이 존재하는지의 여부를 결정하고, 상기 하나 이상의 프로세싱 유닛들은, 상기 이미지를 저장하는 상기 시스템 메모리의 상기 일부가 상기 정의된 기간 내에 어떤 새로운 컨텐츠도 수신하지 않는 경우 그리고 상기 주변 라이팅, 상기 장치의 상기 방위, 또는 상기 장치와 상기 다른 디바이스 사이의 상기 접속에서 상기 정의된 기간 내에 어떤 변화도 없는 경우 상기 이미지가 상기 정적 이미지라고 결정하는, 장치.

#### 청구항 18

제 11항에 있어서,

상기 디스플레이 프로세서는 상기 디스플레이 프로세서가 상기 리스케일링된 이미지를 출력할 때 상기 디스플레이의 조도를 감소시키는, 장치.

#### 청구항 19

제 11항에 있어서,

상기 GPU는 쉐이더를 더 포함하고, 상기 쉐이더는 상기 정적 이미지의 상기 감소된 공간 해상도 버전을 생성하기 위해 상기 정적 이미지를 스케일링하는, 장치.

#### 청구항 20

제 11항에 있어서,

상기 리스케일링된 이미지의 선명도는 상기 이미지의 선명도보다 낮은, 장치.

#### 청구항 21

제 11항에 있어서,

상기 장치는, 텔레비전, 데스크탑 컴퓨터, 랩탑 컴퓨터, 이복 리더, 미디어 플레이어, 태블릿 컴퓨팅 디바이스, 모바일 수신 디바이스, PDA (personal digital assistant), 비디오 게이밍 콘솔, 모바일 화상회의 유닛 (mobile conferencing unit), 모바일 컴퓨팅 디바이스, 및 무선 핸드셋 중 적어도 하나를 포함하는, 장치.

#### 청구항 22

디스플레이;

시스템 버스;

상기 시스템 버스를 통해 액세스 가능한 시스템 메모리;

상기 시스템 메모리 외부에 있는 로컬 메모리;

상기 시스템 메모리의 적어도 일부에 저장된 이미지가 정적 이미지인지 또는 비정적 이미지인지를 결정하는 수단;

그래픽 프로세싱 유닛 (GPU) 으로서:

상기 이미지가 상기 정적 이미지인 것으로 결정되면 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터 상기 정적 이미지를 취출하는 수단;

상기 정적 이미지의 감소된 공간 해상도 버전을 생성하기 위해, 상기 로컬 메모리에서의 이용 가능한 저장 공간의 양에 기초하여, 상기 정적 이미지를 스케일링하는 수단; 및

상기 정적 이미지의 상기 감소된 공간 해상도 버전을, 상기 시스템 버스를 사용하지 않고 상기 GPU의 로컬 메모리에 저장하는 수단을 포함하는,

상기 그래픽 프로세싱 유닛 (GPU); 및

디스플레이 프로세서로서:

상기 정적 이미지의 상기 감소된 공간 해상도 버전을, 상기 시스템 버스를 사용하지 않고 상기 로컬 메모리로부터 취출하는 수단;

상기 정적 이미지의 상기 감소된 공간 해상도 버전을 리스케일링하여 리스케일링된 이미지를 생성하는 수단;

표시를 위해 상기 디스플레이에 상기 리스케일링된 이미지를 출력하는 수단; 및

상기 이미지가 상기 비정적 이미지인 것으로 결정되면, 상기 디스플레이 상에서의 표시를 위해 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터 상기 비정적 이미지를 취출하는 수단을 포함하는,

상기 디스플레이 프로세서를 포함하는, 장치.

#### 청구항 23

삭제

#### 청구항 24

삭제

### 청구항 25

제 22항에 있어서,

상기 로컬 메모리는 상기 GPU의 온칩 메모리를 포함하는, 장치.

### 청구항 26

제 22항에 있어서,

상기 이미지가 상기 비정적 이미지인 것으로 결정되면, 상기 디스플레이 상에서의 표시를 위해 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터 상기 비정적 이미지를 제 1의 리프레시 레이트에서 반복적으로 취출하는 수단을 더 포함하고,

상기 리스케일링된 이미지를 출력하는 수단은, 상기 제 1의 리프레시 레이트보다 더 낮은 제 2의 리프레시 레이트에서 상기 리스케일링된 이미지를 반복적으로 출력하는 수단을 포함하는, 장치.

### 청구항 27

제 22항에 있어서,

상기 시스템 메모리의 일부에 저장된 이미지가 정적 이미지인지 또는 비정적 이미지인지를 결정하는 수단은, 상기 이미지를 저장하는 상기 시스템 메모리의 상기 일부가 정의된 기간 내에 어떤 새로운 컨텐츠도 수신하는 않는 경우 상기 이미지가 상기 정적 이미지라고 결정하는 수단을 포함하는, 장치.

### 청구항 28

제 27항에 있어서,

주변 라이팅에서의 임의의 변화, 상기 장치의 방위에서의 변화들, 또는 상기 장치와 다른 디바이스 사이의 접속에서의 변화들이 존재하는지 여부를 결정하는 수단을 더 포함하고,

상기 이미지가 상기 정적 이미지라고 결정하는 수단은, 상기 이미지를 저장하는 상기 시스템 메모리의 상기 일부가 상기 정의된 기간 내에 어떠한 새로운 컨텐츠도 수신하지 않는 경우 그리고 상기 주변 라이팅, 상기 장치의 상기 방위, 또는 상기 장치와 상기 다른 디바이스 사이의 상기 접속에서 상기 정의된 기간 내에 어떠한 변화도 없는 경우 상기 이미지가 상기 정적 이미지라고 결정하는 수단을 포함하는, 장치.

### 청구항 29

제 22항에 있어서,

상기 디스플레이 프로세서는 상기 출력하는 수단이 상기 리스케일링된 이미지를 출력할 때 상기 디스플레이의 조도를 감소시키는 수단을 더 포함하는, 장치.

### 청구항 30

제 22항에 있어서,

상기 정적 이미지를 스케일링하는 수단은, 상기 정적 이미지의 상기 감소된 공간 해상도 버전을 생성하기 위해 상기 정적 이미지를 상기 GPU의 쉐이더에 의해 스케일링하는 수단을 포함하는, 장치.

### 청구항 31

제 22항에 있어서,

상기 리스케일링된 이미지의 선명도는 상기 이미지의 선명도보다 낮은, 장치.

### 청구항 32

제 22항에 있어서,

상기 장치는, 텔레비전, 데스크탑 컴퓨터, 랩탑 컴퓨터, 이북 리더, 미디어 플레이어, 태블릿 컴퓨팅 디바이스, 모바일 수신 디바이스, PDA (personal digital assistant), 비디오 게이밍 콘솔, 모바일 화상회의 유닛

(mobile conferencing unit), 모바일 컴퓨팅 디바이스, 및 무선 핸드셋 중 적어도 하나를 포함하는, 장치.

### 청구항 33

명령들을 포함하는 비일시적 컴퓨터 관독가능 저장 매체로서,

상기 명령들은 하나 이상의 프로세싱 유닛들로 하여금:

시스템 버스를 통해 액세스 가능한 시스템 메모리의 적어도 일부에 저장된 이미지가 정적 이미지인지 또는 비정적 이미지인지를 결정하게 하고;

상기 이미지가 상기 정적 이미지인 것으로 결정되면 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터, 그래픽 프로세싱 유닛 (GPU)에 의해, 상기 정적 이미지를 취출하게 하고;

상기 정적 이미지의 감소된 공간 해상도 버전을 생성하기 위해, 로컬 메모리에서의 이용 가능한 저장 공간의 양에 기초하여, 상기 GPU에 의해, 상기 정적 이미지를 스케일링하게 하고;

상기 시스템 메모리 외부에 있는 상기 GPU의 상기 로컬 메모리에, 상기 GPU에 의해 그리고 상기 시스템 버스를 사용하지 않고, 상기 정적 이미지의 상기 감소된 공간 해상도 버전을 저장하게 하고;

상기 로컬 메모리로부터 상기 정적 이미지의 상기 감소된 공간 해상도 버전을, 디스플레이에 커플링된 디스플레이 프로세서에 의해 그리고 상기 시스템 버스를 사용하지 않고, 취출하게 하고;

상기 정적 이미지의 상기 감소된 공간 해상도 버전을, 상기 디스플레이 프로세서에 의해, 리스케일링하여 리케일링된 이미지를 생성하게 하고;

상기 디스플레이 프로세서에 의해, 표시를 위해 상기 디스플레이에 상기 리스케일링된 이미지를 출력하게 하고; 그리고

상기 이미지가 상기 비정적 이미지인 것으로 결정되면 상기 디스플레이 상에서의 표시를 위해 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터, 상기 디스플레이 프로세서에 의해, 상기 비정적 이미지를 취출하게 하는, 명령들을 포함하는 비일시적 컴퓨터 관독가능 저장 매체.

### 청구항 34

삭제

### 청구항 35

삭제

### 청구항 36

제 33항에 있어서,

상기 로컬 메모리는 상기 GPU의 온칩 메모리를 포함하는, 명령들을 포함하는 비일시적 컴퓨터 관독가능 저장 매체.

### 청구항 37

제 33항에 있어서,

상기 이미지가 상기 비정적 이미지인 것으로 결정되면 상기 디스플레이 상에서의 표시를 위해 상기 시스템 버스를 통해 상기 시스템 메모리의 상기 일부로부터 상기 비정적 이미지를 제 1의 리프레시 레이트에서 반복적으로 취출하기 위한 명령들을 더 포함하고,

상기 리스케일링된 이미지를 출력하기 위한 명령들은 상기 제 1의 리프레시 레이트보다 더 낮은 제 2의 리프레시 레이트에서 상기 리스케일링된 이미지를 반복적으로 출력하기 위한 명령들을 포함하는, 명령들을 포함하는 비일시적 컴퓨터 관독가능 저장 매체.

### 청구항 38

제 33항에 있어서,

상기 시스템 메모리의 상기 일부에 저장된 상기 이미지가 상기 정적 이미지인지 또는 상기 비정적 이미지인지를 결정하기 위한 상기 명령들은, 상기 이미지를 저장하는 상기 시스템 메모리의 상기 일부가 정의된 기간 내에 어떤 새로운 컨텐츠도 수신하는 않는 경우 상기 이미지가 상기 정적 이미지라고 결정하기 위한 명령들을 포함하는, 명령들을 포함하는 비일시적 컴퓨터 판독가능 저장 매체.

#### 청구항 39

제 38항에 있어서,

주변 라이팅에서의 임의의 변화, 상기 GPU 및 상기 디스플레이 프로세서를 포함하는 디바이스의 방위에서의 변화들, 또는 상기 디바이스와 다른 디바이스 사이의 접속에서의 변화들이 존재하는지의 여부를 결정하기 위한 명령들을 더 포함하고,

상기 이미지가 상기 정적 이미지라고 결정하기 위한 명령들은, 상기 이미지를 저장하는 상기 시스템 메모리의 상기 일부가 상기 정의된 기간 내에 어떠한 새로운 컨텐츠도 수신하지 않는 경우 그리고 상기 주변 라이팅, 상기 디바이스의 상기 방위, 또는 상기 디바이스와 상기 다른 디바이스 사이의 상기 접속에서 상기 정의된 기간 내에 어떠한 변화도 없는 경우 상기 이미지가 상기 정적 이미지라고 결정하기 위한 명령들을 포함하는, 명령들을 포함하는 비일시적 컴퓨터 판독가능 저장 매체.

#### 청구항 40

제 33항에 있어서,

상기 리스케일링된 이미지를 출력하고 있을 때 상기 디스플레이의 조도를 감소시키기 위한 명령들을 더 포함하는, 명령들을 포함하는 비일시적 컴퓨터 판독가능 저장 매체.

#### 청구항 41

제 33항에 있어서,

상기 정적 이미지를 스케일링하기 위한 명령들은, 상기 정적 이미지의 상기 감소된 공간 해상도 버전을 생성하기 위해 상기 GPU의 쉐이더에 의해 상기 정적 이미지를 스케일링하기 위한 명령들을 포함하는, 명령들을 포함하는 비일시적 컴퓨터 판독가능 저장 매체.

#### 청구항 42

제 33항에 있어서,

상기 리스케일링된 이미지의 선명도는 상기 이미지의 선명도보다 낮은, 명령들을 포함하는 비일시적 컴퓨터 판독가능 저장 매체.

### 발명의 설명

#### 기술 분야

[0001] 본 개시는 이미지 디스플레이에 관한 것으로, 보다 구체적으로는, 이미지를 디스플레이하는 것에 대한 전력 절감 기술에 관한 것이다.

#### 배경 기술

[0002] 많은 상이한 타입들의 디바이스들이 디바이스들의 디스플레이 상에 디스플레이하기 위한 이미지를 생성한다. 몇몇 실시형태들에서, 생성된 이미지는 디바이스들의 시스템 메모리에 저장될 수도 있다. 생성된 이미지를 디스플레이하기 위해, 디바이스들 내의 회로부는 시스템 메모리로부터 생성된 이미지를 추출하고 생성된 이미지를 디스플레이로 출력한다.

#### 발명의 내용

#### 과제의 해결 수단

[0003] 개요

[0004]

본 개시는 디바이스의 디스플레이 상에 정적 이미지들을 디스플레이하는 것에 대한 전력 절감 기술들을 설명한다. 몇몇 실시형태들에서, 회로부, 예컨대 디스플레이 프로세서는 시스템 메모리 대신 로컬 메모리로부터 정적 이미지를 취출하고, 그 정적 이미지를 디스플레이 상에 디스플레이할 수도 있다. 로컬 메모리로부터 정적 이미지를 취출하기 위해 활용되는 전력량은 시스템 메모리로부터 정적 이미지를 취출하기 위해 활용된 전력보다 더 적을 수도 있다.

[0005]

일 실시형태에서, 본 개시는 시스템 버스를 통해 액세스 가능한 시스템 메모리의 적어도 일부에 저장된 이미지가 정적 이미지인지 또는 비정적 이미지인지의 여부를 결정하는 것을 포함하는 방법을 설명한다. 상기 방법은 또한, 이미지가 정적 이미지인 것으로 결정되면 시스템 버스를 통해 시스템 메모리의 일부로부터, 그래픽 프로세싱 유닛 (GPU)에 의해, 정적 이미지를 취출하고, 정적 이미지의 감소된 공간 해상도 버전을 생성하기 위해 정적 이미지를, GPU에 의해, 스케일링하고, 시스템 메모리의 외부에 있는 GPU의 로컬 메모리에 정적 이미지의 감소된 공간 해상도 버전을, GPU에 의해, 저장하는 것을 포함한다. 상기 방법은, 로컬 메모리로부터 정적 이미지의 감소된 공간 해상도 버전을, 디스플레이에 커플링된 디스플레이 프로세서에 의해, 취출하고, 리스케일링된 이미지를 생성하기 위해 정적 이미지의 감소된 공간 해상도 버전을, 디스플레이 프로세서에 의해, 리스케일링하고, 리스케일링된 이미지를 표시를 위해 디스플레이로, 디스플레이 프로세서에 의해, 출력하는 것을 더 포함한다.

[0006]

다른 실시형태에서, 본 개시는 디스플레이, 시스템 버스, 시스템 버스를 통해 액세스 가능한 시스템 메모리, 시스템 메모리 외부에 있는 로컬 메모리, 하나 이상의 프로세싱 유닛들, 그래픽 프로세싱 유닛 (GPU), 및 디스플레이 프로세서를 포함하는 장치를 설명한다. 하나 이상의 프로세싱 유닛들은, 시스템 메모리의 적어도 일부에 저장된 이미지가 정적 이미지인지 또는 비정적 이미지인지 결정하도록 동작가능하다. GPU는 이미지가 정적 이미지인 것으로 결정되는 경우 시스템 버스를 통해 시스템 메모리의 일부로부터 정적 이미지를 취출하고, 정적 이미지의 감소된 공간 해상도 버전을 생성하기 위해 정적 이미지를 스케일링하고, 정적 이미지의 감소된 공간 해상도 버전을 로컬 메모리에 저장하도록 동작가능하다. 디스플레이 프로세서는, 로컬 메모리로부터 정적 이미지의 감소된 공간 해상도 버전을 취출하고, 리스케일링된 이미지를 생성하기 위해 정적 이미지의 감소된 공간 해상도 버전을 리스케일링하고, 리스케일링된 이미지를 표시를 위해 디스플레이로 출력하도록 동작가능하다.

[0007]

다른 실시형태에서, 본 개시는 디스플레이, 시스템 버스, 시스템 버스를 통해 액세스 가능한 시스템 메모리, 시스템 메모리 외부에 있는 로컬 메모리를 포함하는 장치를 설명한다. 상기 장치는 또한, 시스템 메모리의 적어도 일부에 저장된 이미지가 정적 이미지인지 또는 비정적 이미지인지 결정하는 수단을 포함한다. 상기 장치는 그래픽 프로세싱 유닛 (GPU) 및 디스플레이 프로세서를 더 포함한다. 그래픽 프로세싱 유닛 (GPU)은 이미지가 정적 이미지인 것으로 결정되면 시스템 버스를 통해 시스템 메모리의 일부로부터 정적 이미지를 취출하는 수단, 정적 이미지의 감소된 공간 해상도 버전을 생성하기 위해 정적 이미지를 스케일링하는 수단, 및 정적 이미지의 감소된 공간 해상도 버전을 GPU의 로컬 메모리에 저장하는 수단을 포함한다. 디스플레이 프로세서는 로컬 메모리로부터 정적 이미지의 감소된 공간 해상도 버전을 취출하는 수단, 정적 이미지의 감소된 공간 해상도 버전을 리스케일링하여 리스케일링된 이미지를 생성하는 수단, 및 표시를 위해 리스케일링된 이미지를 디스플레이로 출력하는 수단을 포함한다.

[0008]

다른 실시형태에서, 본 개시는 명령들을 포함하는 비일시적 컴퓨터 판독가능 저장 매체를 설명하는데, 상기 명령들은 하나 이상의 프로세싱 유닛들로 하여금 시스템 버스를 통해 액세스 가능한 시스템 메모리의 적어도 일부에 저장된 이미지가 정적 이미지인지 또는 비정적 이미지인지 결정하게 한다. 상기 명령들은 또한, 이미지가 정적 이미지인 것으로 결정되면 시스템 버스를 통해 시스템 메모리의 일부로부터, 그래픽 프로세싱 유닛 (GPU)에 의해, 정적 이미지를 취출하고, 정적 이미지의 감소된 공간 해상도 버전을 생성하기 위해 정적 이미지를, GPU에 의해, 스케일링하고, 시스템 메모리의 외부에 있는 GPU의 로컬 메모리에 정적 이미지의 감소된 공간 해상도 버전을, GPU에 의해, 저장하기 위한 명령들을 포함한다. 상기 명령은, 로컬 메모리로부터 정적 이미지의 감소된 공간 해상도 버전을, 디스플레이에 커플링된 디스플레이 프로세서에 의해, 취출하고, 리스케일링된 이미지를 생성하기 위해 정적 이미지의 감소된 공간 해상도 버전을, 디스플레이 프로세서에 의해, 리스케일링하고, 리스케일링된 이미지를 표시를 위해 디스플레이로, 디스플레이 프로세서에 의해, 출력하는 것을 더 포함한다.

[0009]

본 개시의 하나 이상의 양태들의 상세들은 첨부된 도면과 하기의 설명으로부터 설명된다. 본 개시에서 설명된 기술들의 다른 특징들, 목적들, 및 이점들은 하기의 상세한 설명, 도면들, 및 특허청구범위로부터 명확해질 것이다.

것이다.

### 도면의 간단한 설명

[0010] 도 1a 내지 도 1d는 본 개시에 따른 예시적인 디바이스를 예시하는 블록도이다.

도 2는 이미지가 동적 이미지 또는 정적 이미지인지를 프로세싱 유닛이 결정할 수도 있는 몇몇 예시적인 상태들을 예시하는 상태도이다.

도 3a 및 도 3b는 도 1a 내지 도 1d의 그래픽 프로세싱 유닛 (GPU)의 실시형태들을 더 상세히 예시하는 블로도들이다.

도 4는 본 개시에 따른 하나 이상의 프로세싱 유닛들의 예시적인 동작을 예시하는 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

#### 상세한 설명

[0012] 본 개시는 전력 절감을 촉진하는 정적 이미지들을 디스플레이하는 기술들에 관련된다. 본 개시의 기술들은, 비디오 또는 이미지 컨텐츠를 제공하는 텔레비전들, 데스크탑 컴퓨터들, 및 랩탑 컴퓨터들, 이북 리더들, 미디어 플레이어들, 태블릿 컴퓨팅 디바이스들, 모바일 수신 디바이스들, PDA들 (personal digital assistants), 비디오 디스플레이들을 포함하는 비디오 게임용 콘솔들, 모바일 화상회의 유닛들, 모바일 컴퓨팅 디바이스들, 무선 핸드셋들 등과 같지만 이들에 제한되지 않는 컴퓨팅 디바이스들에서 구현될 수도 있다.

[0013] 그래픽 프로세싱 유닛 (GPU)과 같은 컴포넌트들, 및 비디오 디코더와 같은 잠재적으로 다른 컴포넌트들은 디스플레이용 이미지를 생성하는 컨텐츠를 제공한다. 정적 이미지는 정의된 기간동안 그 컨텐츠가 변하지 않는 디스플레이된 이미지일 수도 있다. 예를 들면, 이미지에 기여하는 컴포넌트들 중 어느 것도, 정의된 기간동안 디바이스에 의해 디스플레이되고 있는 것을 변경시키는 어떠한 새로운 정보도 제공하지 않으면, 디바이스에 의해 디스플레이되고 있는 그 이미지는 정적 이미지로서 간주될 수도 있다. 예를 들면, 디바이스 상의 프로세서와 같은 하나 이상의 프로세싱 유닛들은, 임의의 컴포넌트, 예컨대 GPU가 디바이스에 의해 디스플레이되고 있는 것을 변경시키는 어떤 새로운 정보를 제공하는지의 여부를 모니터링할 수도 있다. 이러한 새로운 정보가 없다고 프로세서가 결정하면, 프로세서는 디스플레이되고 있는 이미지가 정적 이미지라고 결정할 수도 있다. 프로세서 이외의 컴포넌트가 임의의 새로운 정보가 존재하는지의 여부를 결정할 수도 있고, 그리고 디스플레이되고 있는 이미지가 정적 이미지라고 결정할 수도 있음이 이해되어야만 한다.

[0014] 몇몇 실시형태들에서, 이미지가 정적 이미지라고 결정될 수 있기 이전에 만족되어야만 하는 추가적인 조건들이 존재할 수도 있다. 예를 들면, 디바이스가 이미지를 디스플레이하고 있는 환경이 상대적으로 일정하게 유지되어야만 한다. 일 예로서, 디바이스에 의해 디스플레이되고 있는 이미지가 정적 이미지로 분류되기 위해서 정의된 기간 동안 주변 라이팅, 및 디바이스 방위 (orientation)가 일정하게 유지되어야 할 필요가 있을 수도 있다. 다른 예로서, 디바이스가 외부 비디오 인터페이스와 함께 사용되면 (예를 들면, HDMI를 통해 TV에 연결된 모바일 디바이스), 디바이스와 외부 디바이스 사이의 접속은 정의된 기간 내에 변경되지 않을 수도 있다. 이미지가 디스플레이되고 있는 환경에 대한 변경들은, 잠재적으로, 디스플레이되고 있는 이미지가 변경되도록 야기할 수도 있다. 이미지에서의 이러한 변경은 이미지가 정적 이미지가 되지 않도록 야기할 수도 있다.

[0015] 이미지가 정적 이미지인 것으로 결정될 수 있기 이전에 임의의 또는 모든 환경 조건들이 만족되어야 하는 것이 필요하지 않을 수도 있다. 몇몇 예들에서, 이미지가 정적 이미지라는 것을 결정하기 위해, 이미지에 기여하는 컴포넌트들 중 어느 것도, 정의된 기간 동안 디바이스에 의해 디스플레이되고 있는 것을 변경시키는 임의의 새로운 정보를 제공하지 않았다고 결정하는 것이 충분할 수도 있다.

[0016] 일 예로서, 이미지가 정적 이미지라고 결정되기 이전의 정의된 기간은 대략 15초일 수도 있다. 그러나, 본 개시의 양태들은 이렇게 제한되는 것은 아니다. 이미지가 정적 이미지라고 결정되기 이전의 정의된 기간은 프로그램될 수도 있고 상이한 상황들에 대해 상이할 수도 있다. 예를 들면, 이미지가 정적 이미지라고 결정되기 이전의 정의된 기간은, 이미지가 정적 이미지라고 결정되기 이전의 시간에 여러 변수들이 영향을 줄 수도 있다는 점에서, 애르고드적일 수도 있다. 일 예로서, 유저가 한 페이지에 얼마나 오래 머무르는지의 이력은, 이미지가 정적 이미지라고 결정되기 이전의 시간의 양에 영향을 끼칠 수도 있다. 다른 예로서, 유저에 의해 실행되는 어플리케이션의 타입은, 이미지가 정적 이미지라고 결정될 수 있기 이전에 얼마나 많은 시간이 경과해야하는지를 결정할 수도 있다. 이미지가 정적 이미지라고 결정될 수 있기 이전의 시간의 양을

결정하기 위해 활용되는 여러 다른 변수들이 존재할 수도 있고, 본 개시의 양태들은 임의의 이러한 상황들까지 확장될 수도 있다.

[0017] 정적 이미지 또는 비정적 (non-static) 이미지, 예를 들면, 동적 이미지는, 최초, GPU 외부에 있는 시스템 메모리에 저장되고 시스템 버스를 통해 액세스가능할 수도 있다. 더 상세히 설명되는 바와 같이, 하나 이상의 프로세싱 유닛들, 예컨대 GPU는, GPU에 의해 활용되는 로컬 메모리 내에, 정적 이미지, 또는 정적 이미지의 스케일링된 버전을 저장할 수도 있다. 로컬 메모리는 GPU의 온칩 메모리일 수도 있다. 몇몇 실시형태들에서, 디스플레이 프로세서는 시스템 메모리로부터 비정적 이미지들을 취출할 수도 있고, 로컬 메모리로부터 정적 이미지들, 또는 정적 이미지들의 스케일링된 버전들을 취출할 수도 있다. 비정적 이미지들은 정의된 기간 내에 디스플레이에 의해 디스플레이되고 있는 것을 변경하는 이미지들일 수도 있고, 반면 정적 이미지들은 정의된 기간 내에 디스플레이 상에서 변경되지 않는 이미지들일 수도 있다. 예를 들면, 디스플레이가 재생 비디오를 표시하는 경우, 디스플레이되고 있는 비디오의 프레임은 정의된 기간 내에 변경될 수도 있다. 그러나, 비디오가 멈추면, 디스플레이되고 있는 비디오의 프레임은 정의된 기간 내에 변경되지 않을 수도 있다.

[0018] 디스플레이 프로세서는 제 1의 리프레시 레이트에서 시스템 메모리로부터 비정적 이미지들을 반복적으로 취출하고, 제 1의 리프레시 레이트에서 각각의 리프레시 싸이클 이후에 비정적 이미지들로 디스플레이를 업데이트할 수도 있다. 몇몇 실시형태들에서, 디스플레이 프로세서는 제 1의 리프레시 레이트보다 더 낮은 제 2의 리프레시 레이트에서 로컬 메모리로부터 정적 이미지들을 반복적으로 취출하고, 제 2의 리프레시 레이트에서 각각의 리프레시 싸이클 이후에 디스플레이로 정적 이미지를 반복적으로 출력한다. 몇몇 대안적 실시형태들에서, 제 1 및 제 2의 리프레시 레이트들이 동일할 수도 있다. 그러나, 몇몇 비제한적인 예시적인 구현예들에서, 제 2의 리프레시 레이트가 제 1의 리프레시 레이트보다 더 느리면 전력 소비에서의 절감이 있을 수도 있다.

[0019] 이미지가 정적 이미지인 것으로 결정되면, GPU는 제한적인 그래픽 프로세싱을 수행하든지 또는 그래픽 프로세싱을 수행하지 않을 수도 있다. 다시 말하면, 디스플레이가 정적 이미지를 디스플레이하고 있을 때, GPU는 휴면 상태 (dormant)에 있을 수도 있다. GPU가 휴면 상태에 있으면, GPU에 할당된 로컬 메모리의 일부들은 사용되지 않을 수도 있다. 더 상세히 설명되는 바와 같이, 그래픽 프로세싱에 대해 로컬 메모리가 GPU에 의해 사용되지 않는 경우, 본 개시의 양태들은 로컬 메모리 내에 정적 이미지의 스케일링된 버전을 저장할 수도 있다.

[0020] 몇몇 실시형태들에서, 정적 이미지인 것으로 결정된 이미지를 어떤 컴포넌트가 생성했는지는 관련되지 않을 수도 있다. 예를 들면, GPU 또는 다른 컴포넌트, 예컨대 비디오 디코더가 정적 이미지를 생성했을 수도 있다. 그러나, 이미지가 정적 이미지인 것으로 결정되면, 어떤 컴포넌트가 정적 이미지를 생성했는지에 무관하게, GPU에 할당된 로컬 메모리의 일부들은 사용되지 않을 수도 있다. 예를 들면, 어떤 컴포넌트가 이미지를 생성했는지에 무관하게, 이미지가 정적 이미지인 경우, GPU가 정적 이미지를 생성한 컴포넌트가 아닌 경우에도, GPU는 휴면 상태에 있을 수도 있다. 몇몇 실시형태들에서, 이미지가 정적 이미지인 것으로 결정되는 경우 GPU에 할당된 로컬 메모리의 일부가 사용되지 않을 수도 있기 때문에, GPU에 할당된 로컬 메모리의 일부는 정적 이미지의 스케일링된 버전을 저장하는 데 적합할 수도 있다.

[0021] 로컬 메모리는 디바이스의 여러 컴포넌트들에 대해 온칩 메모리로서 칭해질 수도 있지만, 시스템 메모리는 오프칩 (off-chip)이며 데이트 액세스를 위한 시스템 버스를 요구할 수도 있다. 일반적으로, GPU는 디바이스의 시스템 메모리보다는 로컬 메모리로부터 더 빨리 그리고 더 적은 소비 전력으로 데이터를 취출하고 로컬 메모리에 더 빨리 그리고 더 적은 소비 전력으로 데이터를 저장할 수도 있다. 마찬가지로, 다른 컴포넌트들, 예컨대 디스플레이 프로세서는, 디바이스의 시스템 메모리보다는 로컬 메모리로부터 더 빨리 그리고 더 적은 소비 전력으로 데이터를 취출하고 로컬 메모리에 더 빨리 그리고 더 적은 소비 전력으로 데이터를 저장할 수도 있다.

[0022] 위에서 설명한 바와 같이, 몇몇 실시형태들에서, 디스플레이 프로세서는 시스템 메모리로부터 디스플레이용 이미지를 취출할 수도 있다. 본 개시에서 설명된 몇몇 실시형태들에서, 정적 이미지의 스케일링된 버전이 로컬 메모리에 저장되면, 디스플레이 프로세서는 이러한 이미지를 시스템 메모리보다는 로컬 메모리로부터 취출할 수도 있다. 시뮬레이션에 의하면, 디스플레이 프로세서가, 예를 들면 시스템 버스를 통해 시스템 메모리로부터 정적 이미지를 취출하는 것에 비해, 로컬 메모리로부터 정적 이미지를 취출하는데 필요한 전력의 대략 1/10을 소비할 수도 있다는 것을 알게 되었다. 이와 같이, 본 개시의 양태들은 정적 이미지를 디스플레이하는데 소비되는 전력의 양을 절감할 수도 있다.

[0023] 몇몇 실시형태들에서, 하나 이상의 프로세싱 유닛들, 예를 들면, GPU는 정적 이미지의 스케일링된 버전, 즉 스케일링된 정적 이미지를 먼저 생성할 수도 있다. 정적 이미지의 스케일링된 버전은 감소된 공간 해상도를

갖는 정적 이미지의 버전일 수도 있다. 몇몇 실시형태들에서, 정적 이미지의 스케일링된 버전을 저장하는 데 필요한 스토리지의 양은 정적 이미지를 저장하는 데 필요한 스토리지의 양보다 더 적을 수도 있다. 로컬 메모리에 의해 제공되는 스토리지의 양이 전체 정적 이미지를 저장하는 데 필요한 스토리지의 양보다 더 적을 수도 있기 때문에 GPU가 스케일링된 정적 이미지를 생성하는 것이 적절할 수도 있다. 로컬 메모리에 의해 제공된 스토리지의 양이 전체 정적 이미지를 저장하는 데 필요한 스토리지의 양보다 더 크거나 동일한 경우, GPU는 정적 이미지를 스케일링할 필요가 없을 수도 있음이 이해되어야만 한다. 그러나, 예시적인 목적을 위해, GPU가 정적 이미지를 감소된 공간 해상도로 스케일링 할 수도 있음이 가정된다. 디스플레이를 위해, 디스플레이 프로세서는 정적 이미지를 리스케일링하고, 리스케일링된 이미지를 표시를 위해 디스플레이로 출력할 수도 있다.

[0024] 또한, 상이한 디바이스들에 대한 디스플레이들은 상이한 디스플레이 해상도들, 예를 들면, 디스플레이된 픽셀들의 수에 대해 구성될 수도 있다. 정적 이미지를 스케일링함으로써, 본 개시의 기술들은 상이한 디스플레이 해상도들을 갖는 디바이스들로 확장될 수도 있다.

[0025] 스케일링된 정적 이미지를 생성하기 위해, GPU는 시스템 메모리로부터 정적 이미지의 사본을 관독할 수도 있다. 그 다음 GPU는, 스케일링된 정적 이미지를 저장하는 데 필요한 스토리지의 양이 로컬 메모리에 의해 제공된 스토리지의 양보다 더 적거나 또는 동일하게 되도록 정적 이미지를 스케일링할 수도 있다. 예를 들면, GPU는 2X2 픽셀들의 블록에 대한 픽셀 값들을 단일 픽셀들에 대한 픽셀 값으로 대체할 수도 있다. 이런 식으로, GPU는 4의 인자에 의해 정적 이미지를 스케일링하여, 정적 이미지를 저장하는 데 필요한 스토리지의 양을 4의 인자만큼 감소시킬 수도 있다. 픽셀들의 블록에 대한 픽셀 값들을 단일 픽셀에 대한 픽셀 값으로 대체하는 기술은 테시메이션 (decimation)으로 칭해질 수도 있다.

[0026] GPU가 정적 이미지를 스케일링하는 다른 기술이 존재할 수도 있으며, 본 개시에서의 실시형태들은 본원에서 설명된 예시적인 스케일링 기술들에 제한되지 않는다. 또한, 정적 이미지를 스케일링하는 경우, GPU는 디스플레이되고 있는 이미지의 컨텐츠를 변경시키는 다른 그래픽 프로세싱 기능들을 수행하지 않을 수도 있다. 예를 들면, GPU가 다른 그래픽 프로세싱 기능들을 수행한다면, GPU의 출력은 디스플레이되고 있는 이미지를 변경할 수도 있고, 이것은 그 이미지가 더 이상 정적 이미지가 아니도록 할 수도 있다.

[0027] 몇몇 실시형태들에서, GPU는 스케일링된 정적 이미지, 예를 들면, 정적 이미지의 감소된 공간 해상도 버전을 로컬 메모리에 저장할 수도 있다. 몇몇 대안적 실시형태들에서, GPU는 스케일링된 정적 이미지를 시스템 메모리에 일시적으로 저장하고, 스케일링된 정적 이미지를 시스템 메모리로부터 취출하고, 스케일링된 정적 이미지를 로컬 메모리에 저장할 수도 있다.

[0028] 그 다음 디스플레이 프로세서는, 시스템 버스를 통해 시스템 메모리로부터의 정적 이미지 대신, 디스플레이를 위해 스케일링된 정적 이미지, 예를 들면 정적 이미지의 감소된 공간 해상도 버전을 로컬 메모리로부터 취출할 수도 있다. 디스플레이 프로세서는 시스템 메모리로부터 이미지를 취출하는 것에 비해 로컬 메모리로부터 이미지를 취출하는 데 더 적은 전력을 소비할 수도 있다. 몇몇 실시형태들에서, 디스플레이 프로세서는 스케일링된 정적 이미지를 리스케일하고 리스케일된 정적 이미지를 디스플레이에 제공할 수도 있다. 리스케일링된 정적 이미지의 해상도는 원래의 정적 이미지의 해상도처럼 풍부하거나 또는 조밀하지 않을 수도 있다. 그러나, 디스플레이를 보는 유저는 선명도에서의 감소를 구별하지 못할 수도 있다.

[0029] 위에서 설명된 바와 같이, 본 개시의 양태들은, 시스템 메모리로부터 풀해상도의 이미지를 취출하는 대신, 디스플레이용으로 로컬 메모리로부터의 스케일링된 정적 이미지를 취출함으로써 전력 절감을 촉진할 수도 있다. 본 개시의 양태들은 추가적인 전력 절감 기술들을 또한 제공할 수도 있다.

[0030] 예를 들면, 위에서 설명된 바와 같이, 디스플레이 프로세서는 미리 결정된 리프레시 레이트에서 시스템 메모리로부터 이미지들을 반복적으로 취출할 수도 있다. 미리 결정된 리프레시 레이트는 동적 이미지들 (디스플레이되고 있는 것을 변경시키고 있는 이미지들)을 디스플레이하기 위해 상대적으로 빠를 (예를 들면 120Hz) 수도 있다. 정적 이미지에 대해, 디스플레이의 컨텐츠가 변경되지 않고 있기 때문에, 이렇게 상대적으로 빠른 레이트에서 디스플레이를 리프레시할 필요가 없을 수도 있다. 몇몇 실시형태들에서, 디스플레이 프로세서는 제 1의 리프레시 레이트보다 더 낮은 제 2의 리프레시 레이트에서 리스케일링된 정적 이미지를 반복적으로 출력할 수도 있다. 디스플레이 프로세서가 단위 시간당 이미지를 취출하는 횟수가 감소될 수도 있기 때문에, 리프레시 레이트에서의 감소는 전력 절감을 또한 촉진할 수도 있다. 또한, 로컬 메모리에 저장된 스케일링된 이미지가 정적 이미지의 감소된 공간 해상도 버전이기 때문에, 디스플레이 프로세서가 로컬 메모리로부터 취출하는 비트들의 수는 리프레시 사이클당 감소될 수도 있다.

[0031] 다른 예로서, 디스플레이 프로세서는 이미지가 디스플레이되는 디스플레이 상에서의 픽셀들의 조도 (illumination intensity)를 감소시킬 수도 있다. 디스플레이 상에서의 픽셀들의 조도에서의 감소는 전력 절감을 또한 촉진할 수도 있다.

[0032] 도 1a 내지 도 1d는 디바이스 (10)의 예시적인 컴포넌트들을 예시하는 블록도들이다. 디바이스 (10)의 실시형태들은, 비디오 또는 이미지 컨텐츠를 제공하는 텔레비전들, 데스크탑 컴퓨터들, 및 랩탑 컴퓨터들, 이북 리더들, 미디어 플레이어들, 태블릿 컴퓨팅 디바이스들, 모바일 수신 디바이스들, PDA들 (personal digital assistants), 비디오 게임용 콘솔들, 모바일 화상회의 유닛들, 모바일 컴퓨팅 디바이스들, 무선 핸드셋들 등을 포함하지만 이들에 제한되지 않는다.

[0033] 도 1a 내지 도 1d에 예시된 바와 같이, 디바이스 (10)는 프로세서 (12), 그래픽 프로세싱 유닛 (GPU; 14), 로컬 메모리 (16), 디스플레이 프로세서 (18), 인코더/디코더 (코덱; 20), 비디오 프로세서 유닛 (22), 어플리케이션 데이터 무버 (application data mover; 24), 시스템 메모리 (26), 및 디스플레이 (28)와 같은 컴포넌트들을 포함할 수도 있다. GPU (14)와 로컬 메모리 (16) 주변의 접선들은, 몇몇 실시형태들에서, 하기에 더 상세히 설명되는 바와 같이, GPU (14)와 로컬 메모리 (16)가 공통 접적 회로 (IC) 상에 형성될 수도 있음을 나타낸다. 디바이스 (10)는 시스템 버스 (15)를 또한 포함할 수도 있다. 프로세서 (12), 그래픽 프로세싱 유닛 (GPU; 14), 디스플레이 프로세서 (18), 인코더/디코더 (코덱; 20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24)는 시스템 버스 (15)를 통해 시스템 메모리 (26)로부터의 데이터에 액세스할 수도 있다. 프로세서 (12), 그래픽 프로세싱 유닛 (GPU; 14), 디스플레이 프로세서 (18), 인코더/디코더 (코덱; 20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24)는 시스템 버스 (15)를 사용하지 않고 로컬 메모리 (16)로부터의 데이터에 액세스할 수도 있다.

[0034] 디바이스 (10)는 도 1a 내지 도 1d에 예시된 것들에 추가로 컴포넌트들을 포함할 수도 있다. 예를 들면, 디바이스 (10)가 모바일 무선 전화기인 경우의 실시형태들에서 전화 통신들을 실현하기 위해 또는 디바이스 (10)가 미디어 플레이어인 경우 스피커를 실현하기 위해, 디바이스 (10)는 스피커와 마이크를 포함할 수도 있는데, 이들 중 어느 것도 도 1a 내지 도 1d에 도시되지 않았다. 디바이스 (10)는 데이터의 송수신을 위한 트랜시버, 유저가 디바이스 (10)와 상호작용하기 위한 유저 인터페이스, 및 디바이스 (10)의 컴포넌트들에 전력을 제공하는 파워 서플라이를 또한 포함할 수도 있다. 몇몇 실시형태들에서, 디스플레이 (28)가 터치 스크린인 경우, 디스플레이 (28)는 적어도 부분적으로 유저 인터페이스로서 기능할 수도 있다.

[0035] 프로세서 (12), GPU (14), 로컬 메모리 (16), 디스플레이 프로세서 (18), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24)는 단일 접적 회로 (IC) 또는 IC들의 세트 (즉, 칩셋)에서의 컴포넌트들로서 형성될 수도 있다. 이를 실시형태들에서, 프로세서 (12), GPU (14), 디스플레이 프로세서 (18), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24)는 IC내에서 반드시 별개의 하드웨어 유닛들일 필요는 없다. 예시의 목적들을 위해, 이들 컴포넌트들 각각의 기능성은 개별적으로 설명된다.

그러나, 이러한 설명은 이해를 용이하게 하기 위해 제공된 것으로, 이들 컴포넌트들이 IC내에서 반드시 별개의 컴포넌트들이라는 것을 의미하는 것으로 해석되어선 안된다. 몇몇 대안적 실시형태들에서, 프로세서 (12), GPU (14), 디스플레이 프로세서 (18), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24)는 개개의 컴포넌트들, 예를 들면 개개의 IC들로서 형성될 수도 있다. 이를 대안적 실시형태들에서, 프로세서 (12), GPU (14), 디스플레이 프로세서 (18), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24)는 시스템 버스 (15)를 통해 서로 통신할 수도 있지만, 시스템 버스 (15)를 사용하지 않고 로컬 메모리 (16)와 통신할 수도 있다.

[0036] 프로세서 (12), GPU (14), 디스플레이 프로세서 (18), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24)는, 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 반도체들 (application specific integrated circuits; ASIC들), 필드 프로그래머블 로직 어레이들 (field programmable logic arrays; FPGA들), 또는 다른 등가의 접적 또는 이산 로직 회로부로서, 개별적으로 또는 조합하여 구현될 수도 있다. GPU (14)가 개개의 컴포넌트로서 형성된 실시형태들에서, 로컬 메모리 (16)는 GPU (14)에 형성될 수도 있다, 즉, 로컬한 온칩 메모리로서 GPU (14)와 함께 형성될 수도 있다. 예시의 목적들을 위해, 그리고 설명의 용이함을 위해, 로컬 메모리 (16)는 GPU (14) 외부에 있는 것으로 예시된다. 로컬 메모리 (16)는 GPU (14)의 로컬 메모리로서 칭해질 수도 있다.

[0037] 디바이스 (10)의 여러 컴포넌트들은 로컬 메모리 (16)에 빠르게 그리고 낮은 소비 전력으로 액세스할 수도 있다. 예를 들면, 로컬 메모리 (16)는 프로세서 (12), GPU (14), 디스플레이 프로세서 (18), 코덱 (20), 비

디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 와 같은 컴포넌트들을 포함하는 IC에 대한 온칩 메모리일 수도 있다. 로컬 메모리 (16) 의 실시형태들은 캐시 메모리 또는 레지스터들, 또는 빠르게 액세스 될 수 있으며, 몇몇 실시형태들에서는 시스템 버스 (15) 를 사용하지 않고 액세스될 수 있는 임의의 다른 타입의 로컬 메모리를 포함한다. 프로세서 (12), GPU (14), 디스플레이 프로세서 (18), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 는, 시스템 버스 (15) 를 통해 시스템 메모리 (26) 에 데이터를 저장하거나 시스템 메모리 (26) 로부터 데이터를 취출하는 것에 비해, 훨씬 더 빠르게 그리고 더 낮은 소비 전력으로 로컬 메모리 (16) 로부터 데이터를 취출하고 로컬 메모리 (16) 에 데이터를 저장할 수도 있다.

[0038] 예시된 바와 같이, 시스템 메모리 (26) 는 프로세서 (12), GPU (14), 디스플레이 프로세서 (18), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 에 대해 외부에 있을 수도 있다. 시스템 메모리 (26) 가 외부에 있기 때문에, 프로세서 (12), GPU (14), 디스플레이 프로세서 (18), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 시스템 버스 (15) 를 통해 시스템 메모리 (26) 와 통신할 수도 있다. 대역폭 제한들 및 데이터 스케줄링으로 인해, 프로세서 (12), GPU (14), 디스플레이 프로세서 (18), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 와 시스템 메모리 (26) 사이의 통신은, 별도의 버스를 포함하지 않거나 과도한 스케줄링을 요구하지 않는 로컬 메모리 (16) 와의 통신보다 더 느릴 수도 있다. 또한, 시스템 버스를 따라 시스템 메모리 (26) 로 또는 시스템 메모리 (26) 로부터 데이터를 전송하는 데 소비되는 전력은, 별도의 버스를 포함하지 않는 로컬 메모리 (16) 로 또는 로컬 메모리 (16) 로부터 데이터를 전송하는 데 소비되는 전력보다 더 클 수도 있다.

[0039] 예를 들면, 시스템 메모리 (26) 로부터 데이터를 취출하기 위해, 디스플레이 프로세서 (18) 는, 디스플레이 프로세서 (18) 가 시스템 버스 (15) 를 통해 통신하도록 스케줄링되어 있다는 것을 보장할 필요가 있을 수도 있다. 디스플레이 프로세서 (18) 가 시스템 버스 (15) 를 통해 통신하도록 스케줄링되지 않으면, 디스플레이 프로세서 (18) 는 잠재적으로 아이들 상태를 유지할 수도 있다. 또한, 시스템 버스 (15) 를 통해 통신하기 위해 디스플레이 프로세서 (18) 에 의해 필요한 전력의 양은, 시스템 버스 (15) 를 사용하지 않고 로컬 메모리 (16) 와 직접적으로 통신하기 위해 디스플레이 프로세서 (18) 에 의해 필요한 전력의 양보다 더 클 수도 있다.

[0040] 프로세서 (12) 는 하나 이상의 어플리케이션들을 실행하는 프로세서일 수도 있다. 예를 들면, 프로세서 (12) 는 웹브라우저, 이메일 어플리케이션, 스프레드시트들, 비디오 게임들, 미디어 플레이어들, 또는 디스플레이에 대해 시청 가능한 컨텐츠를 생성하는 다른 어플리케이션들과 같은 어플리케이션들을 실행할 수도 있다. 프로세서 (12) 는 디바이스 (10) 의 중앙 처리 유닛 (CPU) 일 수도 있다. 이들 실시형태들에서, 프로세서 (12) 는 디바이스 (10) 의 여러 컴포넌트들에게 그들이 수행하도록 구성된 기능들을 수행하도록 명령할 수도 있다.

[0041] 일 예로서, 코덱 (20) 은 그것이 디코딩하여 실행을 위해 프로세서 (12) 로 제공하기 위한 명령들을 수신할 수도 있다. 코덱 (20) 은 인코더/디코더일 수도 있다. 예를 들면, 코덱 (20) 은 인코딩된 데이터를 수신하고, 인코딩된 데이터를 디코딩하고, 디코딩된 데이터를 프로세서 (12) 및/또는 시스템 메모리 (26) 로 제공할 수도 있다. 다른 예로서, 코덱 (20) 은 데이터를 수신하고, 그 데이터를 인코딩하고, 인코딩된 데이터를 송신할 수도 있다. 몇몇 실시형태들에서, 코덱 (20) 은 인코더 및 비디오 디코더일 수도 있다. 이들 실시형태들에서, 코덱 (20) 은 시스템 메모리 (26) 의 저장된 비디오의 일부들을 취출하고, 저장된 비디오의 그 일부들을 디코딩하고, 디코딩된 부분을 후속의 재생을 위해 다시 시스템 메모리 (26) 에 저장할 수도 있다.

[0042] 몇몇 실시형태들에서, 프로세서 (12) 에 의해 실행되는 어플리케이션들에 대한 명령들은 시스템 메모리 (26) 에 저장될 수도 있다. 시스템 메모리 (26) 의 실시형태들은, RAM (random access memory), ROM (read only memory), EEPROM (electrically erasable programmable read-only memory), CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 스토리지 디바이스들, 플래시 메모리, 또는 데이터 또는 명령들을 저장하기 위해 사용될 수 있는 임의의 다른 매체들을 포함하지만, 이들에 제한되는 것은 아니다. 몇몇 양태들에서, 시스템 메모리 (26) 는 여러 프로세싱 유닛들, 예를 들면, 도 1a 내지 도 1d에 예시된 예시적인 컴포넌트들이 그들의 설명된 기능들을 수행하게 하기 위한 명령들을 포함할 수도 있다. 따라서, 시스템 메모리 (26) 는 하나 이상의 프로세싱 유닛들로 하여금 여러 기능들을 수행하게 하기 위한 명령들을 포함하는 컴퓨터 판독가능 저장 매체일 수도 있다.

[0043] 시스템 메모리 (26) 는, 몇몇 실시형태들에서, 비일시적 저장 매체로서 간주될 수도 있다. 용어 "비일시적"이란 저장 매체가 반송파 또는 전파 신호로 구현되지 않는 것을 나타낼 수도 있다. 그러나, 용어 "비일

"시적"은 시스템 메모리 (26) 가 움직일 수 없다는 것을 의미하는 것으로 해석되어선 안된다. 일 예로서, 시스템 메모리 (26) 는 디바이스 (10) 로부터 제거되어, 다른 디바이스로 이동될 수도 있다. 다른 예로서, 시스템 메모리 (26) 와 실질적으로 유사한 시스템 메모리는 디바이스 (10) 에 삽입될 수도 있다. 어떤 실시형태들에서, 비일시적 저장 매체는 시간이 경과함에 따라 변할 수 있는 데이터를 (예를 들면, RAM에) 저장할 수도 있다.

[0044] GPU (14) 는 프로세서 (12) 에 의해 생성된 이미지들에 대한 속성들을 수신하고 수신된 속성들에 대해 그래픽 관련 프로세싱을 수행할 수도 있다. 예를 들면, GPU (14) 는 디스플레이 (28) 에 디스플레이될 이미지의 픽셀들의 각각에 대한 픽셀 값들을 결정할 수도 있다. 예를 들면, GPU (14) 는, 적용 가능하다면, 프로세서 (12) 로부터 수신된 이미지의 각각의 픽셀에 대해, 칼라 값들, 예를 들면, 레드-그린-블루 (RGB) 값들 또는 루마 및 크로미넌스 값들, 불투명값들, 예를 들면, 알파 값들, 및 텍스쳐 값들을 결정할 수도 있다. 일반적으로, GPU (14) 는 이미지 내의 각각의 픽셀에 대해 라이팅, 세이딩, 블렌딩, 쿨링 (culling), 및 다른 이러한 그래픽 관련 프로세싱과 같은 기능들을 수행할 수도 있다. GPU (14) 의 실시형태들은 도 3a 및 도 3b에 더 상세히 설명된다.

[0045] GPU (14) 가 이미지 내의 픽셀들에 대한 픽셀 값들을 결정한 이후, GPU (14) 는 그 이미지에 대한 픽셀 값들을 시스템 메모리 (26) 내에 저장할 수도 있다. 예를 들면, 도 1a에 예시된 바와 같이, 시스템 메모리 (26) 는 이미지 (30) 를 시스템 메모리 (26) 의 일부 (32) 내에 저장한다. 이미지 (30) 는 GPU (14) 에 의해 결정된 것과 같은 이미지 (30) 내의 픽셀들의 각각에 대한 픽셀 값들을 포함할 수도 있다.

[0046] 시스템 메모리 (26) 의 일부 (32) 는 이미지들, 예컨대 이미지 (30) 를 저장하기 위해 예약된 시스템 메모리 (26) 의 예약 부분일 수도 있다. 상기 일부 (32) 의 사이즈는 적어도 하나의 이미지의 픽셀 값들을 저장하기에 충분할 수도 있다. 예시의 목적들을 위해, 일부 (32) 는 디스플레이 버퍼 또는 프레임 버퍼로서 간주될 수도 있다. 그러나, 본 개시의 양태들은 그렇게 제한하는 것으로 간주되어선 안된다. 일부 (32) 는 하나 이상의 이미지들을 저장하기 위해 예약된 시스템 메모리 (26) 의 임의의 일부일 수도 있다.

[0047] 비디오 프로세서 유닛 (22) 은 디스플레이될 비디오에 대해 프로세싱 기능들을 수행할 수도 있다. 예를 들면, 비디오 프로세서 유닛 (22) 은 비디오 컨텐츠의 압축 및 압축해제와 같은 기능들을 수행할 수도 있다. 비디오 프로세서 유닛 (22) 은 비디오 컨텐츠에 대해 프리프로세싱 및 포스트프로세싱 기능들을 또한 수행할 수도 있다. 예를 들면, 비디오 프로세서 유닛 (22) 은 비디오 컨텐츠의 노이즈 감소, 스케일링, 및 회전과 같은 기능들을 수행할 수도 있다.

[0048] 어플리케이션 데이터 무버 (24) 는 시스템 메모리 (26) 에 저장된 데이터를 로컬 메모리 (16) 로 이동시킬 수도 있다. 예를 들면, 프로세서 (12), GPU (14), 디스플레이 프로세서 (18), 코덱 (20), 및/또는 비디오 프로세서 유닛 (22) 은 어플리케이션 데이터 무버 (24) 로 하여금 시스템 메모리 (26) 로부터 데이터를 취출하고 취출된 데이터를 로컬 메모리 (16) 에 저장하게 할 수도 있다.

[0049] 일반적으로, 프로세서 (12), GPU (14), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 각각은 이미지 (30) 와 같은 이미지를 생성하고, 시스템 메모리 (26) 의 일부 (32) 에 이미지 (30) 를 저장하는 컨텐츠에 어찌면 기억할 수도 있다. 프로세서 (12), GPU (14), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 가 이미지 (30) 생성을 위한 컨텐츠를 동시에 제공할 필요는 없을 수도 있다. 대신, 몇몇 실시형태들에서, 이를 컴포넌트들 중 단지 하나만이 이미지 (30) 를 생성하기 위한 컨텐츠를 제공하고 이미지 (30) 의 그 컨텐츠를 시스템 메모리 (26) 의 일부 (32) 에 저장할 수도 있다. 그러나, 본 개시의 양태들은 그렇게 제한되지 않으며, 예를 들면, 이들 컴포넌트들 중 2개 이상이 이미지 (30) 를 생성하기 위한 컨텐츠를 동시에 제공할 수도 있다.

[0050] 디스플레이 프로세서 (18) 는, 최초, 시스템 메모리 (26) 로부터 저장된 이미지 (30) 를 취출하고 이미지 (30) 를, 이미지 (30) 로부터 디스플레이 프로세서 (18) 를 통해 디스플레이 (28) 속으로, 그리고 디스플레이 (28) 에서의 이미지 (30) 의 점선 경계로 연장하는 점선에 의해 나타내어진 바와 같이, 디스플레이 (28) 로 출력하도록 구성될 수도 있다. 몇몇 실시형태들에서, 디스플레이 프로세서 (18) 는 전용 비디오 인식 프로그램가능 직접 메모리 액세스 엔진으로서 간주될 수도 있다. 예를 들면, 프로세서 (12), GPU (14), 코덱 (20), 및/또는 비디오 프로세서 유닛 (22) 은, 디스플레이 프로세서 (18) 가 이미지 (30) 를 취출해야 하는 위치를 디스플레이 프로세서 (18) 에게 가리킬 수도 있다. 프로세서 (12), GPU (14), 코덱 (20), 및/또는 비디오 프로세서 유닛 (22) 은 디스플레이 프로세서 (18) 에게, 스케일링, 회전, 오버레이, 및 다른 이러한 동작들과 같은 디스플레이 프로세서 (18) 가 어떤 기능들을 수행해야 하는지를 또한 가리킬 수도 있다. 일 예로서, 더 상세

히 설명되는 바와 같이, 프로세서 (12), GPU (14), 코덱 (20), 및/또는 비디오 프로세서 유닛 (22) 은 디스플레이 프로세서 (18) 로 하여금 스케일링된 이미지를 다시 스케일링하게 할 수도 있다.

[0051] 몇몇 실시형태들에서, 디스플레이 프로세서 (18) 는 미리 결정된 리프레시 레이트에서 디스플레이 (28) 를 리프레시할 수도 있다. 예를 들면, 디스플레이 프로세서 (18) 는 미리 결정된 리프레시 레이트에서 시스템 메모리 (26) 로부터 이미지 (30) 를 반복적으로 취출할 수도 있다. 예를 들면, 디스플레이 프로세서 (18) 는 120Hz, 예를 들면 초당 120회의 리프레시 레이트에서 시스템 메모리 (26) 로부터 이미지 (30) 를 취출할 수도 있다. 각각의 리프레시 싸이클 이후에, 디스플레이 프로세서 (18) 는 디스플레이 (28) 가 이미지 (30) 를 다시 디스플레이하게 할 수도 있다. 다시 말하면, 이 실시형태에서, 디스플레이 프로세서 (18) 는 디스플레이 (28) 상에 이미지 (30) 를 초당 120회 리프레시할 수도 있다.

[0052] 디스플레이 프로세서 (18) 는 다른 기능들을 또한 수행하도록 구성될 수도 있다. 예를 들면, 디스플레이 프로세서 (18) 는 주변 라이팅에 기초하여 디스플레이 (28) 의 픽셀들의 조도를 결정할 수도 있다. 픽셀들의 조도는 픽셀들이 디스플레이 (28) 상에서 얼마나 밝게 나타나는지를 가리킬 수도 있다. 더 높은 조도 레벨들은 디스플레이 (28) 가 더 많은 전력을 소모하게 할 수도 있다.

[0053] 몇몇 실시형태들에서, 이미지 (30) 의 컨텐츠는 정의된 기간 내에서 변경되지 않을 수도 있다. 예를 들면, 컴포넌트들, 가령 몇몇 예로서 코덱 (20) 또는 GPU (14) 의 어느 것도 정의된 기간 내에 이미지 (30) 를 저장하는 시스템 메모리 (26) 의 일부 (32) 에 임의의 새로운 정보를 제공하지 않으면, 디스플레이 (28) 에 의해 디스플레이되고 있는 이미지, 가령 이미지 (30) 의 컨텐츠는 정의된 기간 내에 변경되지 않을 수도 있다. 이미지 (30) 의 컨텐츠가 정의된 기간 동안 변경되지 않으면, 이미지 (30) 는 정적 이미지인 것으로 결정될 수도 있다. 예를 들면, 프로세서 (12), GPU (14), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 의 어느 것도 15초 내에 이미지 (30) 의 컨텐츠를 변경하는 임의의 새로운 정보를 제공하지 않으면, 이미지 (30) 는 정적 이미지로서 분류될 수도 있다. 다시 말하면, 디스플레이 (28) 에 의해 디스플레이되고 있는 이미지의 컨텐츠가 정의된 기간 내에 변경되지 않으면, 디스플레이 (28) 에 의해 디스플레이되고 있는 이미지는 정적 이미지인 것으로 결정될 수도 있다.

[0054] 이미지 (30) 를 정적 이미지로 분류하기 위한 기간에 대한 15초의 예는 예시적인 목적들을 위해 제공된 것으로, 제한하는 것으로 간주되어선 안된다. 이미지 (30) 가 정적 이미지로 분류되기 이전의 기간은 여러 기준들에 기초할 수도 있다. 예를 들면, 그 인자는 유저가 한 페이지에 이력적으로 머문 시간의 양일 수도 있다. 다른 인자들은 유저가 실행하고 있는 어플리케이션의 타입일 수도 있거나, 또는 유저가 사용하고 있는 디바이스의 타입일 수도 있다. 일반적으로, 이미지 (30) 가 정적 이미지로서 분류될 수 있기 이전에 경과해야 하는 시간의 양은 특정 구현예들에 의존할 수도 있는 관련 기준에 기초하여 프로그램 가능할 수도 있다. 몇몇 경우들에서, 이미지 (30) 가 정적 이미지로서 분류되기 이전의 정의된 기간에 대해 대략 15초 내지 60초가 적절한 범위일 수도 있다. 그러나, 본 개시의 양태들은 이렇게 제한되는 것은 아니다.

[0055] 상기 실시형태들에서, 프로세서 (12), GPU (14), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 는, 이미지 (30) 를 저장하는 시스템 메모리 (26) 의 일부 (32) 에 임의의 새로운 정보를 15초 내에 제공하지 않을 수도 있다. 그러나, 만약 프로세서 (12), GPU (14), 코덱 (20), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 중 임의의 하나 이상이 이미지 (30) 를 저장하는 시스템 메모리 (26) 의 일부 (32) 에 임의의 새로운 정보를 15초 내에 제공했다면, 이미지 (30) 는 정적 이미지가 아니라 동적 이미지인 것으로 간주될 수도 있다. 본 개시의 양태들은 이 실시형태에 제한되지 않는다. 위에서 설명된 바와 같이, 이미지 (30) 가 정적 이미지인 것으로 결정되기 이전에 경과해야만 하는 기간은 디바이스 (10) 의 상이한 실시형태들에 대해 선택 가능하고, 상이할 수도 있다.

[0056] 예시의 목적들을 위해, 하기에 정적 이미지의 몇몇 실시형태들을 설명한다. 일 예로서, 정적 이미지는 유저가 읽고 있는 디바이스 (10) 상의 페이지일 수도 있다. 그 페이지는 디바이스 (10) 가 이북 리더인 실시형태에서는 책의 페이지일 수도 있다. 그 페이지는 또한 이메일 또는 웹사이트일 수도 있다. 디스플레이 (28) 에 의해 디스플레이되고 있는 페이지는 유저가 그 페이지를 읽고 있을 때 정적으로 유지될 수도 있고, 유저가 이북 리더의 다른 페이지로 넘기거나, 현재의 이메일을 빼서 나오거나, 또는 다른 웹사이트를 로딩한 이후 변경될 수도 있다. 유저가 페이지를 읽는 데 걸리는 시간의 양은 그 페이지를 정적 이미지로서 분류하기에 충분한 것보다 더 길 수도 있다.

[0057] 다른 예로서, 정적 이미지는 디바이스 (10) 의 홈스크린일 수도 있다. 홈스크린은 유저가 디바이스 (10) 의 컨텐츠에 액세스할 수 있는 메인 시작 화면일 수도 있다. 홈스크린의 이미지 컨텐츠는 자주 변경되지 않을

수도 있다. 유저가 정의된 기간보다 길게 홈스크린을 보고 있을 때, 홈스크린은 정적 이미지로서 분류될 수도 있다.

[0058] 또 다른 실시형태로서, 유저는 비디오 예컨대 다운로드한 영화를 볼 수도 있거나 또는 디바이스 (10)에 커플링된 캠코더를 통해 비디오를 볼 수도 있다. 이 실시형태에서, 코덱 (20)은 이미지 데이터를 시스템 메모리 (26)의 일부 (32)에 기록하고 있을 수도 있다. 유저가 비디오를 일시정지하거나, 종료하거나, 또는 정지하는 경우, 디스플레이 (28) 상에 디스플레이되는 이미지는 정의된 기간보다 길게 일정하게 유지될 수도 있다. 이 실시형태에서, 디스플레이 (28) 상에 디스플레이되는 결과적인 이미지는 정적 이미지로서 분류될 수도 있다.

[0059] 이미지 (30) 가 정적 이미지로서 분류되는 다수의 상이한 원인들이 존재할 수도 있다. 본 개시의 양태들은 임의의 이러한 실시형태들로 확장될 수도 있으며, 상기의 실시형태들에 제한되는 것으로 간주되어선 안된다.

[0060] 프로세서 (12), GPU (14), 비디오 프로세서 유닛 (22), 및 어플리케이션 데이터 무버 (24) 중 어느 것도 이미지 (30)의 컨텐츠를 변경시키는 임의의 새로운 정보를 제공하지 않을 때, 프로세서 (12)는 이미지 (30)가 정적 이미지라고 결정할 수도 있다. 일 예로서, 프로세서 (12)는 시스템 메모리 (26)의 일부 (32)의 컨텐츠를 모니터링할 수도 있다. 시스템 메모리 (26)의 일부 (32)의 컨텐츠가 정의된 기간 내에 변경되지 않으면, 프로세서 (12)는, 일부 (32) 내에 저장된 이미지, 예를 들면 이미지 (30)가 정적 이미지라고 결정할 수도 있다.

[0061] 다른 예로서, 프로세서 (12)는 프로세서 (12), GPU (14), 비디오 프로세서 유닛 (22), 어플리케이션 데이터 무버 (24)의 출력들을 모니터링할 수도 있다. 프로세서 (12), GPU (14), 비디오 프로세서 유닛 (22), 어플리케이션 데이터 무버 (24) 중 어느 것도 일부 (32)의 컨텐츠를 변경시키는 임의의 새로운 정보를 출력하지 않으면, 프로세서 (12)는 일부 (32) 내에 저장된 이미지, 예를 들면 이미지 (30)가 정적 이미지라고 결정할 수도 있다. 그러나, 만약 시스템 메모리 (26)의 일부 (32)의 컨텐츠가 변경되면, 디스플레이 (28)에 의해 디스플레이된 이미지들이 정의된 기간 내에 변경되고 있기 때문에, 프로세서 (12)는 디스플레이 (28)에 의해 디스플레이된 이미지가 정적 이미지가 아니라고 결정할 수도 있다.

[0062] 몇몇 실시형태들에서, 이미지 (30)가 정적 이미지라고 프로세서 (12) 이외의 컴포넌트가 결정할 수도 있다. 예시의 목적들을 위해, 본 개시의 양태들은 이미지 (30)가 정적 이미지인지의 여부를 프로세서 (12)가 결정하는 상황에서 설명된다. 그러나, 몇몇 실시형태들에서, 이미지 (30)가 정적 이미지라는 것을 프로세서 (12), 또는 다른 컴포넌트가 결정할 수도 있다는 것을 나타내기 위해, 본 개시의 양태들은 이미지 (30)가 정적 이미지라고 결정하는 것으로서 하나 이상의 프로세싱 유닛들을 설명할 수도 있다.

[0063] 몇몇 실시형태들에서, 이미지 (30)가 정적 이미지라고 하나 이상의 프로세싱 유닛들, 예를 들면, 프로세서 (12)가 결정하기 이전에 만족되어야만 하는 추가적인 기준이 있을 수도 있다. 이를 추가적인 기준은 디바이스 (10)의 환경에 기초할 수도 있다. 예를 들면, 디스플레이 (28)가 이미지 (30)를 디스플레이하고 있는 환경이 상대적으로 일정하게 유지되어야만 한다. 일 예로서, 주변 라이팅 및 디바이스 방위는 이미지 (30)가 정적 이미지로서 분류되기 위해 정의된 기간 동안 일정하게 유지될 필요가 있을 수도 있다. 예를 들면, 디바이스 (10)는 주변 라이팅을 검출하는 하나 이상의 센서들을 포함할 수도 있다. 프로세서 (12)는 주변 라이팅에 어떤 변화가 있는지를 결정하기 위해 이를 센서들의 출력을 모니터링할 수도 있다. 다른 예로서, 디바이스 (10)는 디바이스 (10)의 방위를 결정하는 하나 이상의 가속도계들 또는 자이로스코프들을 포함할 수도 있다. 프로세서 (12)는 디바이스 (10)의 방위에서 어떤 변화가 있는지를 결정하기 위해 가속도계들 또는 자이로스코프들의 출력을 모니터링할 수도 있다. 다른 예로서, 디바이스 (10)는 다른 디바이스에 커플링될 수도 있는데, 예를 들면, 디바이스 (10)는 HDMI 케이블을 통해 TV에 연결된다. 이를 실시형태들에서, 디바이스 (10)와 다른 디바이스 사이의 연결은 변경되지 않을 수도 있으며, 예를 들면, 프로세서 (12)가 이미지 (30)를 정적 이미지로서 분류하는 기간 동안 HDMI 케이블은 제거되지 않을 수도 있다.

[0064] 이미지 (30)가 디스플레이되고 있는 환경에 대한 변경은, 잠재적으로, 이미지 (30), 또는 이미지 (30)의 적어도 겉보기 (appearance)가 디스플레이시 변경되게 할 수도 있다. 이미지 (30)에 대한 이러한 변경은 이미지 (30)가 정적 이미지가 되지 않도록 야기할 수도 있다. 예를 들면, 유저가 디바이스 (10)를 90도만큼 회전시키면, 프로세서 (12)는 이미지 (30)를 90도만큼 또한 회전시킬 수도 있다. 이러한 회전에서의 변경은 이미지 (30)를 변경시킬 수도 있어서 (예를 들면, 이미지 (30)의 컨텐츠를 리사이징), 이미지 (30)가 정적 이미지가 아니게 할 수도 있다.

[0065]

이미지 (30) 가 정적 이미지인 것으로 결정될 수 있기 이전에 임의의 또는 모든 환경 조건들이 만족되어야 하는 것이 필요하지 않을 수도 있다. 몇몇 실시형태들에서, 이미지 (30) 에 기여하는 컴포넌트들 중 어느 것도 정의된 기간 동안 이미지 (30) 를 변경하는, 예를 들면, 디스플레이 (28) 에 의해 디스플레이 되고 있는 것을 변경하는 임의의 새로운 정보를 제공하지 않는다고 하나 이상의 프로세싱 유닛들이 결정하는 것이 충분할 수도 있다.

[0066]

본 개시에서 설명된 몇몇 예시적인 구현예들에서, 이미지 (30) 가 정적 이미지로 분류되면, GPU (14) 는 그래픽 프로세싱을 거의 수행하지 않거나 또는 전혀 수행하지 않을 수도 있다. 예를 들면, 이미지 (30) 가 정적 이미지로 분류되기 위해서, GPU (14) 는 시스템 메모리 (26) 의 일부 (32) 로 임의의 새로운 정보를 출력하지 않을 수도 있다. GPU (14) 가 임의의 새로운 정보를 출력하지 않기 위해, GPU (14) 는 임의의 그래픽 관련 동작들을 수행하지 않을 수도 있다. 다시 말하면, 이미지 (30) 가 정적 이미지이면, GPU (14) 는 휴면 상태에 있거나 또는 시스템 메모리 (26) 의 일부 (32) 에 새로운 정보를 제공하는 그래픽 프로세싱 동작들을 적어도 액티브하게 수행하지 않을 수도 있다.

[0067]

몇몇 실시형태들에서, 로컬 메모리 (16) 의 적어도 일부는 GPU (14) 에 의해 생성된 그래픽 데이터를 저장하기 위해 예약될 수도 있다. GPU (14) 가 휴면 상태에 있으면, GPU (14) 에 의해 생성된 그래픽 데이터를 저장하기 위해 예약된 로컬 메모리 (16) 의 일부는 사용되지 않을 수도 있다. 따라서, 몇몇 실시형태들에서, 이미지 (30) 가 정적 이미지인 경우, GPU (14) 에 의해 생성된 그래픽 데이터를 저장하기 위해 예약된 로컬 메모리 (16) 의 일부는 사용되지 않을 수도 있다.

[0068]

GPU (14) 가 그래픽 관련 동작들을 수행하고 있지 않는 경우, 예를 들면, 이미지 (30) 가 정적 이미지인 경우, GPU (14) 는 GPU (14) 에 의해 생성된 그래픽 데이터를 저장하기 위해 예약된 로컬 메모리 (16) 의 일부 내에 이미지 (30) 의 한 버전을 저장할 수도 있다. 몇몇 실시형태들에서, 이미지 (30) 가 정적 이미지로서 분류된 이후 이미지 (30) 를 저장하기 이전에, GPU (14) 는 이미지 (30) 를 스케일링할 수도 있다. 이미지 (30) 를 스케일링하는 것은 이미지 (30) 의 공간적 해상도를 감소시키는 것으로 간주될 수도 있다. 그러나, 본 개시의 양태들은 GPU (14) 가 이미지 (30) 를 스케일링하는 것을 요구하는 것에 제한되는 것으로 간주되어선 안 된다. GPU (14) 가 로컬 메모리 (16) 에 저장하는 이미지 (30) 의 버전은 이미지 (30) 그 자체일 수도 있거나, 또는 이미지 (30) 의 스케일링된 버전일 수도 있다. 예시의 목적들을 위해, 본 개시에서 설명된 실시형태들은, 이미지 (30) 가 정적 이미지인 것으로 결정된 이후, 이미지 (30) 의 감소된 공간 해상도 버전을 생성하기 위해 GPU (14) 가 이미지 (30) 를 스케일링하는 상황에서 설명된다.

[0069]

이미지 (30) 가 정적 이미지로서 분류된 이후, GPU (14) 가 이미지 (30) 를 스케일링하고 이미지 (30) 의 스케일링된 버전을 로컬 메모리 (16) 에 저장하는 것이 어쩌면 적절할 수도 있는 적어도 2개의 상황들이 존재할 수도 있다. 일 예로서, 로컬 메모리 (16) 에서의, 또는 GPU (14) 에 대해 예약된 로컬 메모리 (16) 의 일부에 서의 저장 공간의 양이 이미지 (30) 의 전체를 저장하기에 불충분할 수도 있다. GPU (14) 는 로컬 메모리 (16) 에서 이용 가능한 저장 공간의 양에 기초하여 이미지 (30) 를 스케일링, 예를 들면, 이미지 (30) 의 해상도를 감소시킬 수도 있다. 예를 들면, GPU (14) 는, 이미지 (30) 의 감소된 공간 해상도 버전을 저장하는 데 필요한 저장 공간의 양이 로컬 메모리 (16) 에서의, 또는 GPU (14) 에 대해 예약된 로컬 메모리 (16) 의 일부에서의 저장 공간의 양보다 더 적도록 또는 동일하도록 이미지 (30) 의 감소된 공간 해상도 버전을 생성할 수도 있다. 그 다음 GPU (14) 는 이미지 (30) 의 스케일링된 버전을 로컬 메모리 (16) 에 저장할 수도 있다. 로컬 메모리 (16) 에 의해 제공된 스토리지의 양이 이미지 (30) 를 전체적으로 저장하는 데 필요한 스토리지의 양보다 더 크거나 또는 동일한 실시형태에서, GPU (14) 는 이미지 (30) 를 스케일링할 필요가 없을 수도 있다.

[0070]

다른 예로서, 이미지 (30) 의 사이즈는 디스플레이 (28) 의 사이즈에 기초할 수도 있다. 디스플레이 (28) 의 사이즈는 디바이스 (10) 의 상이한 타입들에 대해 상이할 수도 있다. 디스플레이 (28) 의 사이즈는 디스플레이 (28) 상의 광센들의 수를 가리킬 수도 있다. 예를 들면, 동일 해상도를 가정하면, 디스플레이 (28) 의 사이즈는, 디바이스 (10) 가 셀룰러 전화기인 실시형태들에서의 디스플레이 (28) 의 사이즈에 비해, 디바이스 (10) 가 태블릿 컴퓨팅 디바이스인 실시형태들에서 더 클 수도 있다. 몇몇 실시형태들에서, GPU (14) 는 디스플레이 (28) 의 사이즈에 무관하게 고정된 해상도로 이미지 (30) 를 스케일링할 수도 있다. 이와 같이, 본 개시의 양태들은 여러 사이즈들의 디스플레이들로 확장될 수도 있다.

[0071]

이미지 (30) 가 정적 이미지로 분류된 이후, GPU (14) 가 이미지 (30) 를 스케일링하는 여러 기술들이 존재할 수도 있다. 하나의 이러한 예시적인 기술은 데시메이션으로 칭해진다. 데시메이션 기술에서, GPU (14)

는 이미지 (30) 의 픽셀들의 블록에 대한 픽셀 값들을 단일 픽셀에 대한 픽셀 값으로 대체할 수도 있다. 일 예로서, 이미지 (30) 의 픽셀들의 블록은 픽셀들의 2X2 블록일 수도 있다. 이 실시형태에서, GPU (14) 는 픽셀들의 2X2 블록에서의 4개의 픽셀 값을 단일 픽셀 값으로 대체할 수도 있다. 이런 식으로, GPU (14) 는 4의 인자에 의해 정적 이미지를 스케일링하여, 이미지를 저장하는 데 필요한 스토리지의 양을 4의 인자만큼 감소시킬 수도 있다. GPU (14) 가 단일 픽셀 값으로 대체하는 이미지 (30) 의 픽셀들의 블록의 사이즈는 로컬 메모리 (16) 의 저장 능력들 및 디스플레이 (28) 의 사이즈에 기초하여 선택가능하게 될 수도 있다.

[0072] 위에서 설명된 데시메이션 기술에는 예시적인 목적들을 위해 그리고 이해를 돋기 위해 설명된 것이다. 이미지 (30) 가 정적 이미지로서 분류된 이후, GPU (14) 가 이미지 (30) 를 스케일링할 수도 있는 다른 기술들이 존재할 수도 있으며, 본 개시의 양태들은 데시메이션의 예시적인 기술에 제한되는 것으로 간주되어선 안된다. 또한, GPU (14) 가 이미지 (30) 를 스케일링할 때, GPU (14) 는 로컬 메모리 (16) 를 활용하는 다른 그래픽 프로세싱 기능들을 수행하지 않을 수도 있다.

[0073] 이미지 (30) 를 스케일링하는 것은 이미지 (30) 를 압축하는 것과 혼돈되어선 안된다. 압축에 있어서는, 이미지 (30) 의 픽셀 값을 나타내는 데 요구되는 비트들의 수가 감소되지만; 이미지 (30) 의 해상도는 일정하게 유지된다. 스케일링에 있어서는, 이미지 (30) 의 해상도가 감소될 수도 있다. 예를 들면, 스케일링에 있어서, 이미지 (30) 의 픽셀 값을 나타내는 데 요구되는 비트들의 수는 이미지 (30) 의 스케일링된 버전의 픽셀 값을 나타내는 데 요구되는 비트들의 수와 동일하지만; 그 픽셀 값을 저장하는 픽셀들의 수는 감소된다. 몇몇 실시형태들에서, GPU (14) 가 이미지 (30) 를 스케일링한 이후, GPU (14) 는 이미지 (30) 의 스케일링된 버전을 압축할 수도 있다.

[0074] 몇몇 실시형태들에서, GPU (14) 가 이미지 (30) 를 스케일링한 이후, GPU (14) 는 이미지 (30) 의 스케일링된 버전을 시스템 메모리 (26) 에 일시적으로 저장할 수도 있다. 예를 들면, GPU (14) 는 시스템 메모리 (26) 에 이미지 (30) 의 감소된 공간 해상도 버전을 일시적으로 저장할 수도 있다. 그 다음 GPU (14) 는 시스템 메모리 (26) 로부터 이미지 (30) 의 스케일링된 버전을 취출하고, 이미지 (30) 의 스케일링된 버전을 로컬 메모리 (16) 에 저장할 수도 있다. 대안적 실시형태에서, GPU (14) 는 이미지 (30) 의 스케일링된 버전을 시스템 메모리 (26) 에 먼저 저장하지 않고 이미지 (30) 의 스케일링된 버전을 로컬 메모리 (16) 에 저장할 수도 있다. 예를 들면, GPU (14) 는 로컬 메모리 (16) 에 이미지 (30) 의 감소된 공간 해상도 버전을 일시적으로 저장할 수도 있다.

[0075] 도 1b 및 도 1c는, 이미지 (30) 가 정적 이미지라고 프로세서 (12) 가 결정한 경우, GPU (14) 가 시스템 메모리 (26) 의 일부 (32) 로부터 이미지 (30) 를 취출하는 실시형태를 예시한다. 예를 들면, 도 1b 및 도 1c는 정적 이미지 (30A) 를 저장하는 것으로서의 시스템 메모리 (26) 의 일부 (32) 를 예시한다. 정적 이미지 (30A) 는 도 1a의 이미지 (30) 와 실질적으로 유사할 수도 있다. 도 1b 및 도 1c는, 도 1a의 이미지 (30) 가 정적 이미지라는 것을 도 1b 및 도 1c의 실시형태들에서 프로세서 (12) 가 결정했다는 것을 나타내기 위해 정적 이미지 (30A) 를 예시한다.

[0076] 정적 이미지 (30A) 로부터 GPU (14) 로 연장하는 도 1b의 점선에 의해 예시된 바와 같이, 일 예로서, GPU (14) 는 시스템 메모리 (26) 의 일부 (32) 로부터 정적 이미지 (30A) 를 취출할 수도 있다. GPU (14) 는 정적 이미지 (30A) 를 스케일링하여 스케일링된 정적 이미지 (34) 를 생성할 수도 있다. 스케일링된 이미지 (34) 는 정적 이미지 (30A) 의 감소된 공간 해상도 버전일 수도 있다. 그 다음 GPU (14) 는 스케일링된 정적 이미지 (34) 를 시스템 메모리 (26) 에 저장할 수도 있다. GPU (14) 는, 스케일링된 정적 이미지 (34) 를 저장하는 데 필요한 스토리지의 양이 로컬 메모리 (16) 의 스토리지의 양, 또는 GPU (14) 로부터의 데이터를 저장하기 위해 예약된 로컬 메모리 (16) 에서의 스토리지의 양보다 더 적거나 또는 동일하도록 정적 이미지 (30A) 를 스케일링할 수도 있다. 예를 들면, GPU (14) 는 로컬 메모리 (16) 에서 이용 가능한 저장 공간의 양에 기초하여 정적 이미지 (30A) 를 스케일링할 수도 있다.

[0077] 그 다음 GPU (14) 는 스케일링된 정적 이미지 (34) 를 로컬 메모리 (16) 에 저장할 수도 있다. 예를 들면, 스케일링된 정적 이미지 (34) 로부터 로컬 메모리 (16) 로 연장하는 도 1c의 점선에 의해 예시된 바와 같이, 일 예로서, GPU (14) 는 시스템 메모리 (26) 로부터 스케일링된 정적 이미지 (34) 를 취출하고 스케일링된 정적 이미지 (34) 를 로컬 메모리 (16) 에 저장할 수도 있다. 몇몇 대안적 실시형태들에서, GPU (14) 는 스케일링된 정적 이미지 (34) 를 시스템 메모리 (26) 에 먼저 저장하지 않고 스케일링된 정적 이미지 (34) 를 로컬 메모리 (16) 에 직접 저장할 수도 있다.

[0078] 도 1b 및 도 1c의 실시형태들이 시스템 메모리 (26) 의 일부 (32) 로부터 정적 이미지 (30A) 를 취출하고, 정적

이미지 (30A) 를 스케일링하여 스케일링된 정적 이미지 (34) 를 생성하고, 스케일링된 정적 이미지 (34) 를 로컬 메모리 (16) 에 저장하는 것으로서 GPU (14) 를 예시하지만, 본 개시의 양태들은 그렇게 제한되는 것은 아니다. 일반적으로, 디스플레이 (28) 가 정적 이미지를 디스플레이 하고 있을 때 GPU (14) 가 임의의 다른 기능들을 수행하지 않고 있을 수도 있기 때문에, GPU (14) 는, 시스템 메모리 (26) 의 일부 (32) 로부터 정적 이미지 (30A) 를 취출하고, 정적 이미지 (30A) 를 스케일링하여 스케일링된 정적 이미지 (34) 를 생성하고, 스케일링된 정적 이미지 (34) 를 로컬 메모리 (16) 에 저장하기 위한 적절한 컴포넌트일 수도 있다. 그러나, 몇몇 실시형태들에서, 프로세서 (12), 또는 잠재적으로 디바이스 (10) 의 다른 컴포넌트가 시스템 메모리 (26) 의 일부 (32) 로부터 정적 이미지 (30A) 를 취출하고, 정적 이미지 (30A) 를 스케일링하여 스케일링된 정적 이미지 (34) 를 생성하고, 스케일링된 정적 이미지 (34) 를 로컬 메모리 (16) 에 저장할 수도 있다. 예시적인 목적들을 위해, 본 개시에서 설명된 실시형태들은, GPU (14) 가 시스템 메모리 (26) 의 일부 (32) 로부터 정적 이미지 (30A) 를 취출하고, 정적 이미지 (30A) 를 스케일링하여 스케일링된 정적 이미지 (34) 를 생성하고, 스케일링된 정적 이미지 (34) 를 로컬 메모리 (16) 에 저장하는 상황에서 설명된다.

[0079] 정적 이미지 (30A) 의 한 버전이 로컬 메모리 (16) 에 저장된 이후, 디스플레이 프로세서 (18) 는 로컬 메모리 (16) 에 저장된 정적 이미지 (30A) 의 상기 버전, 예를 들면, 정적 이미지 (30A) 의 감소된 공간 해상도 버전일 수도 있는 스케일링된 정적 이미지 (34) 를 취출할 수도 있다. 예를 들면, 도 1d에서 스케일링된 정적 이미지 (34) 로부터 디스플레이 프로세서 (18) 연장하는 점선에 의해 예시된 바와 같이, 디스플레이 프로세서 (18) 는 로컬 메모리 (16) 로부터 스케일링된 정적 이미지 (34) 를 취출하고, 정적 이미지 (34) 를 리스케일링하여 리스케일링된 이미지 (36) 를 생성하고, 리스케일링된 이미지 (36) 를 표시를 위해 디스플레이 (28) 로 출력할 수도 있다. 몇몇 실시형태들에서, 디스플레이 프로세서 (18) 는, 시스템 버스 (15) 를 통해 시스템 메모리 (26) 로부터 이미지를 취출하는 것에 비해, 로컬 메모리 (16) 로부터 리스케일링된 정적 이미지 (34) 를 취출하는 데 더 적은 전력을 소비할 수도 있다. 몇몇 실시형태들에서, 전력 감소는 10배만큼의 전력 감소일 수도 있다. 이와 같이, 본 개시에서 설명된 몇몇 예시적인 구현예들은 소비 전력의 감소를 촉진할 수도 있다.

[0080] 몇몇 실시형태들에서, 하나 이상의 프로세싱 유닛들, 예를 들면, GPU (14) 가 정적 이미지 (30A) 의 한 버전을 로컬 메모리 (16) 에 저장한 이후, 프로세서 (12) 는 GPU (14) 를 슬립 모드에 둘 수도 있다. 예를 들면, 이미지 (30) 가 정적 이미지 (30A) 라고 프로세서 (12) 가 결정하는 경우, GPU (14) 는 어떠한 프로세싱도 수행하지 않을 수도 있기 때문에, 예를 들면, GPU (14) 는 휴면 상태에 있을 수도 있다. 위에서 설명된 바와 같이, GPU (14) 는 정적 이미지 (30A) 를 스케일링하여 스케일링된 정적 이미지 (34) 를 생성하고, 스케일링된 정적 이미지 (34) 를 로컬 메모리 (16) 에 저장할 수도 있다. 그 다음, 전력을 보존하기 위해, 프로세서 (12) 는 GPU (14) 를 슬립 모드에 둘 수도 있는데, 슬립 모드에서, GPU (14) 는 더 적은 전력을 소비한다. 그 다음, GPU (14) 의 가능성이 필요해지면, 예를 들면, 디스플레이 (28) 에 의해 디스플레이된 이미지가 변경되면, 프로세서 (12) 는 GPU (14) 가 임의의 필요한 그래픽 관련 작업들을 수행할 수 있도록 GPU (14) 를 깨울 수도 있다.

[0081] 디스플레이 프로세서 (18) 는 디스플레이 (28) 의 픽셀들의 각각에 픽셀 값들을 할당하기 위해 스케일링된 정적 이미지 (34) 를 리스케일링할 수도 있다. 예를 들면, 일 예로서, GPU (14) 는 정적 이미지 (30A) 의 2X2 픽셀들의 블록을 단일 픽셀 값으로 대체하여 스케일링된 정적 이미지 (34) 를 생성할 수도 있다. 스케일링된 정적 이미지 (34) 를 리스케일링하여 리스케일링된 이미지 (36) 를 생성하기 위해, 디스플레이 프로세서 (18) 는 정적 이미지 (30A) 의 2X2 픽셀들의 블록에 대응하는 디스플레이 (28) 의 2X2 픽셀들의 블록에 대한 픽셀 값들 각각에, 스케일링된 정적 이미지 (34) 를 생성하기 위해 사용된 단일 픽셀 값의 값을 할당할 수도 있다. 그 다음 리스케일링된 이미지 (36) 는 디스플레이 (28) 의 픽셀들의 각각에 대한 픽셀 값들을 포함할 수도 있다. 또한, 디스플레이 프로세서 (18) 는 스케일링된 정적 이미지 (34) 를 리스케일링하기 위해 다른 기술들을 적용할 수도 있다. 본 개시의 양태들은 위에서 설명된 예시적인 리스케일링 기술들에 제한되는 것으로 간주되어선 안된다.

[0082] 일 예로서, 예시적인 목적들을 위해 그리고 이해를 용이하게 하기 위해, 디스플레이 (28) 가 640x480 픽셀들을 포함한다고 가정한다. 이 실시형태에서, 정적 이미지 (30A) 도 또한 640x480 픽셀들을 포함할 수도 있다.

스케일링된 정적 이미지 (34) 를 생성하기 위해, GPU (14) 는 이미지 (30A) 의 640x480 픽셀들에서의 픽셀들의 2x2 블록의 각 픽셀들에 하나의 단일 픽셀 값을 할당할 수도 있다. 이 실시형태에서, 스케일링된 정적 이미지 (34) 는 320x240 픽셀 값들 (예를 들면 640x480을 2x2로 나눈 것) 을 포함할 수도 있다. 스케일링된 정적 이미지 (34) 를 리스케일링하여 리스케일링된 이미지 (36) 를 생성하기 위해, 디스플레이 프로세서 (18) 는 디스플레이 (28) 상의 픽셀들의 제 1의 2x2 블록에 대한 픽셀 값에, 320x240 픽셀 값들에서의 제 1의 픽셀

값들의 픽셀 값을 할당하고, 등등일 수도 있다. 따라서, 본 실시형태에 있어서, 디스플레이 (28) 상의 픽셀들의 2x2 블록에서의 4개의 픽셀들에 대해 동일한 픽셀 값이 할당되지만, 정적 이미지 (30A)에서의 픽셀들의 2x2 블록에서의 4개의 픽셀들에 대해서는 상이한 픽셀 값들이 할당될 수도 있다.

[0083] 몇몇 실시형태들에서, 리스케일링된 이미지 (36)의 해상도는 정적 이미지 (30A)의 해상도만큼 풍부하거나 조밀하지 않을 수도 있다. 예를 들면, 리스케일링된 이미지 (36)의 해상도는 정적 이미지 (30A)의 해상도보다 낮을 수도 있다. 그러나, 디스플레이 (28)를 보는 유저는 선명도에서의 감소를 구별하지 못할 수도 있다. 또한, 몇몇 실시형태들에서, 선명도에서의 감소는 유저의 경험에 부정적으로 영향을 끼치지 않을 수도 있다. 예를 들면, 유저가 영화를 일시정지하는 경우, 일시정지된 이미지의 선명도에서의 마이너한 감소는 유저에게 중요하지 않을 수도 있다. 다른 실시형태로서, 유저는 홈스크린 상의 그래픽 아이콘들의 위치를 일반적으로 알고 있을 수도 있다. 그래픽 아이콘들의 선명도에서의 마이너한 감소는 홈스크린 상의 그래픽 아이콘들 중 임의의 것을 선택하는 유저의 능력에 영향을 끼치지 않을 수도 있다.

[0084] 리스케일링된 이미지 (36)의 해상도에서의 감소의 양은 디바이스 (10)의 타입에 기초할 수도 있다. 비제한적인 실시형태로서, 디바이스 (10)가 모바일 폰이면, 정적 이미지 (30A)의 해상도와 비교하여, 리스케일링된 이미지 (36)의 해상도에서의 감소는 근사적으로 대략 2.5의 인자만큼의 감소일 수도 있다. 다른 비제한적인 실시형태로서, 디바이스 (10)가 태블릿 컴퓨팅 디바이스이면, 정적 이미지 (30A)의 해상도와 비교하여, 리스케일링된 이미지 (36)의 해상도에서의 감소는 근사적으로 대략 2의 인자만큼의 감소일 수도 있다. 그러나, 이들 실시형태들은 예시적인 목적들을 위해 제공된 것이며 제한적인 것으로 간주되어선 안된다. 리스케일링된 이미지 (36)의 해상도에서의 감소는 모바일 폰 또는 태블릿 컴퓨팅 디바이스에 대해, 각각, 2 또는 2.5의 인자로 제한되지 않을 수도 있다.

[0085] 몇몇 실시형태들에서, 디스플레이 프로세서 (18)는, 예를 들면, 로컬 메모리 (16)로부터 이미지를 취출하는 것에 더하여, 전력 소비에서의 감소를 촉진하기 위한 추가적 기능들을 수행할 수도 있다. 예를 들면, 디스플레이 프로세서 (18)는, 디스플레이 프로세서 (18)가 시스템 메모리 (26)로부터 이미지를 취출하는지 또는 로컬 메모리 (16)로부터 이미지를 취출하는지에 기초하여 상이한 리프레시 레이트들에서 디스플레이 (28)를 리프레시할 수도 있다. 디스플레이 프로세서 (18)가 디스플레이 (28) 상에 이미지를 나타낸 이후, 디스플레이 (28) 상의 픽셀들의 조명 레벨이 저하되기 시작한다. 예를 들면, 디스플레이 (28) 상의 픽셀들은 전하를 저장하는 커페시터들로서 유추될 수도 있고, 전하의 레벨은 조명 레벨에 상관될 수도 있다. 시간이 지남에 따라, 커페시터들은 방전되기 시작하여 조명 레벨을 저하시키게 된다. 저하에 대처하기 위하여, 디스플레이 프로세서 (18)는 이미지를 다시 나타냄으로써 디스플레이 (28)를 주기적으로 리프레시할 수도 있는데, 이것은 커페시터들을 재충전하는 것으로 유추될 수도 있다. 디스플레이 프로세서 (18)가 디스플레이 (28)를 초당 리프레시하는 횟수는 리프레시 레이트로서 칭해질 수도 있다.

[0086] 그 컨텐츠가 변경되는 비정적 이미지들, 예를 들면 동적 이미지들에 대해, 디스플레이 프로세서 (18)는 상대적으로 빠른 리프레시 레이트에서 디스플레이 (28)를 리프레시할 수도 있다. 예를 들면, 몇몇 텔레비전들은 120Hz의 리프레시 레이트들을 제공한다. 이렇게 빠른 리프레시 레이트들은, 동적 이미지들의 컨텐츠가 변경될 수도 있기 때문에, 동적 이미지들에 대해 유익할 수도 있다.

[0087] 그러나, 그 컨텐츠가 변경되지 않는 정적 이미지들에 대해, 상대적으로 빠른 리프레시 레이트에서 디스플레이 (28)를 리프레시함에 있어서는 이점이 없을 수도 있다. 예를 들면, 정적 이미지의 컨텐츠가 변경되지 않기 때문에, 정적 이미지의 동일한 이미지 컨텐츠를 1초에 120회 제공하는 것은 유저의 경험에 긍정적으로 영향을 주지 않을 수도 있다. 일 예로서, 유저가 영화를 재생하고 있는 경우, 영화의 이미지들은, 제공되고 있는 이미지들이 영화의 프레임마다 변경될 수도 있기 때문에, 동적 이미지들일 수도 있다. 이 경우, 디스플레이 프로세서 (18)가 상대적으로 빠른 리프레시 레이트에서 디스플레이 (28)를 리프레시하는 것이 유익할 수도 있다. 유저가 영화를 일시정지하는 경우, 디스플레이된 프레임에서 어떤 변화들도 없기 때문에, 일시정지된 장면은 정적 이미지일 수도 있다. 이 경우, 디스플레이 (28)의 컨텐츠가 변경되고 있지 않기 때문에 상대적으로 빠른 리프레시 레이트에서 디스플레이 프로세서 (18)가 디스플레이 (28)를 리프레시하는 것은 필요 없을 수도 있다.

[0088] 몇몇 실시형태들에서, 디스플레이 프로세서 (18)가 시스템 메모리 (26)로부터 이미지를 취출하는 경우 디스플레이 프로세서 (18)는 제 1의 리프레시 레이트에서 디스플레이 (28)를 리프레시할 수도 있다. 예를 들면, 여전히 정적 이미지로서 분류되어야 하는 이미지 또는 동적 이미지를 취출하는 경우, 디스플레이 (28)를 리프레시하기 위한 제 1의 리프레시 레이트에서 디스플레이 (28) 상에 표시하기 위해 디스플레이 프로세서 (18)는

이러한 이미지들을 시스템 메모리 (26)로부터 반복적으로 취출할 수도 있다. 디스플레이 프로세서 (18) 가로컬 메모리 (16)로부터 이미지를 취출하는 경우, 디스플레이 프로세서 (18)는 제 1의 리프레시 레이트보다 더 낮은 제 2의 리프레시 레이트에서 디스플레이 (28)를 리프레시할 수도 있다. 예를 들면, 디스플레이 프로세서 (18)는, 로컬 메모리 (16)로부터, 스케일링된 정적 이미지 (34)를 취출하고, 스케일링된 정적 이미지 (34)를 리스케일링하여 리스케일링된 이미지 (36)를 생성하고, 제 1의 리프레시 레이트보다 더 낮은 제 2의 리프레시 레이트에서 디스플레이 (28) 상에 표시하기 위해 리스케일링된 이미지 (36)를 디스플레이 (28)에 반복적으로 출력할 수도 있다.

[0089] 리프레시 레이트에서의 감소는 전력 소비에서의 감소를 또한 촉진할 수도 있다. 예를 들면, 디스플레이 프로세서 (18)는, 디스플레이 프로세서 (18)가 로컬 메모리 (16)로부터 정적 이미지 (30A)의 한 버전을 취출할 필요가 있는 초당 횟수들이, 디스플레이 프로세서 (18)가 시스템 메모리 (26)로부터 동적 이미지를 취출할 필요가 있는 초당 횟수들보다 더 적을 수도 있기 때문에, 디스플레이 프로세서 (18)는 더 적은 전력을 소비할 수도 있다. 또한, 스케일링된 정적 이미지 (34)의 픽셀들의 수는 정적 이미지 (30A)의 픽셀들의 수보다 더 적을 수도 있다. 리프레시 싸이클 당 디스플레이 프로세서 (18)가 취출해야 할 필요가 있는 픽셀 값들의 수에서의 감소로 인해, 디스플레이 프로세서 (18)는 정적 이미지 (30A)를 취출하는 것보다 스케일링된 정적 이미지 (34)를 취출하는 것에 더 적은 전력을 소비할 수도 있다.

[0090] 제 2의 리프레시 레이트는 여러 인자들에 기초할 수도 있다. 예를 들면, 제 2의 리프레시 레이트의 레이트는, 디스플레이 (28) 상의 픽셀들이 플리커하는 것으로 보이는 리프레시 레이트보다 더 높거나 동일할 수도 있다. 리프레시 레이트가 너무 느리면, 디스플레이 (28) 상의 픽셀들은 플리커하는 것으로 보일 수도 있고, 이것은 유저의 경험에 나쁜 영향을 끼칠 수도 있다. 플리커의 출현은 디스플레이 (28) 상의 픽셀들의 조명 레벨에 대한 빠른 변화들에 의해 야기될 수도 있다. 예를 들면, 상대적으로 느린 리프레시 레이트에서, 디스플레이 (28) 상의 픽셀들의 조명 레벨은 리프레시 싸이클들 사이에서 실질적으로 저하될 수도 있다. 그러면, 픽셀들의 조명 레벨이 원래의 조명 레벨로 재설정되는 각 리프레시 싸이클 이후, 조명 레벨에서의 빠른 증가는 디스플레이 (28) 상의 픽셀들로 하여금 그들이 플리커하는 것처럼 나타나게 할 수도 있다.

[0091] 디스플레이 (28) 상의 픽셀들이 플리커하는 것으로 나타나는 리프레시 레이트는 디스플레이 (28)의 설계에 기초할 수도 있다. 몇몇 실시형태들에서, 디스플레이 (28) 상의 픽셀들이 플리커하는 것처럼 나타나게 하는 것을 방지하기 위해서는 대략 15Hz 이상의 리프레시 레이트가 충분할 수도 있다. 이를 실시형태들에서, 제 2의 리프레시 레이트는 대략 15Hz로 설정될 수도 있다. 그러나, 본 개시의 양태들은 이렇게 제한되는 것으로 간주되어선 안되며, 제 2의 리프레시 레이트의 레이트는 디스플레이 (28)의 설계, 및 임의의 다른 가능한 적절한 인자들, 예를 들면, 디스플레이 프로세서 (18)가 제 1 및 제 2의 리프레시 레이트들을 생성할 수 있는 클록 신호의 주파수에 기초하여 선택 가능할 수도 있다.

[0092] 몇몇 실시형태들에서, 디스플레이 프로세서 (18)는 디스플레이 (28)의 픽셀들의 조도를 또한 결정할 수도 있다. 예를 들면, 주변광의 레벨이 상대적으로 높으면, 디스플레이 프로세서 (18)는 디스플레이 (28) 상의 픽셀들 각각의 조도를, 주변광의 레벨이 상대적으로 낮은 경우 디스플레이 프로세서 (18)가 설정할 조도보다 더 높게 설정할 수도 있다. 디스플레이 (28)의 픽셀들의 조도는 각 픽셀의 휙도로서 간주될 수도 있다. 몇몇 실시형태들에서, 디스플레이 (28)가 리스케일링된 이미지 (36)를 디스플레이하고 있는 경우 디스플레이 프로세서 (18)는 디스플레이 (28)의 픽셀들의 조도를 감소시킬 수도 있다.

[0093] 높은 조도 픽셀들을 디스플레이하기 위해 디스플레이 (28)에 의해 소비되는 전력은, 낮은 조도 픽셀들을 디스플레이하기 위해 디스플레이 (28)에 의해 소비되는 전력보다 더 클 수도 있다. 픽셀들의 조도를 감소시킴으로써, 디스플레이 (28)가 리스케일링된 이미지 (36)를 디스플레이하고 있는 경우, 디스플레이 (28)에 의해 소비되는 전력은 감소될 수도 있다. 이와 같이, 디스플레이 프로세서 (18)는 소비 전력에서의 감소를 더 촉진할 수도 있다.

[0094] 도 2는 이미지를 동적 이미지 또는 정적 이미지라고 프로세서 (12)가 결정하는 몇몇 예시적인 상태들을 예시하는 상태도이다. 도 2의 상태도에서 예시된 실시형태들은 예시적인 목적들을 위해, 그리고 이해를 돋기 위해 사용된 것이다. 본 개시의 양태들은 도 2의 실시형태들로 제한되는 것으로 간주되어선 안된다. 예를 들면, 하나 이상의 프로세싱 유닛들, 예를 들면, 프로세서 (12)로 하여금 이미지가 정적 이미지라는 것을 결정하게 할 수도 있는 몇몇 상황들을 예시하지만, 본 개시의 양태들은 도 2에 예시된 실시형태들로 그렇게 제한되는 것은 아니다.

[0095] 도 2는 동적 이미지 상태 (38) 및 정적 이미지 상태 (40)를 예시한다. 생성된 이미지들이 동적 이미지들일

수도 있는 상황들의 실시형태들은, 동적 이미지 상태 (38)에서 예시된 바와 같이, 시스템 구성 (system configuration) 동안, 어플리케이션이 실행될 준비가 된 경우, 그리고 어플리케이션이 정상 상태 (steady-state)에 도달하는 경우의 이미지들을 포함한다. 예를 들면, 디바이스 (10)의 시스템 구성 동안, 디스플레이 (28) 상에 디스플레이되는 임의의 이미지는 변경될 수도 있다. 또한, 시스템 구성 이후, 유저는 실행을 위한 어플리케이션, 예를 들면, 웹브라우저, 이메일 어플리케이션, 비디오를 재생하는 어플리케이션 등을 선택할 수도 있다. 이러한 선택들 동안, 디스플레이 (28) 상에 디스플레이된 이미지들은 변경될 수도 있다.

또한, 유저가 어플리케이션을 실행한 이후, 어플리케이션은 정상 상태에 도달할 수도 있다. 정상 상태에서, 디바이스 (10)는 어플리케이션의 액션들을 수행하고 있을 수도 있다. 예를 들면, 유저는 영화를 재생하는 어플리케이션을 실행할 수도 있다. 정상 상태에서, 디바이스 (10)는 영화들의 프레임들을 디스플레이 (28) 상에 제공할 수도 있다.

**[0096]** 정상 상태의 어플리케이션에 의해 생성된 이미지가 정적 이미지라고 결정되게 하는 여러 원인들이 존재할 수도 있다. 예를 들면, 정적 이미지 상태 (40)에서 예시된 바와 같이, 유저는 어플리케이션을 정지시키거나 또는 유저는 어플리케이션을 빠져나와 홈스크린으로 리턴할 수도 있다. 일 예로서, 유저는 영화를 일시정지시킬 수도 있다. 영화를 일시정지시킨 유저는 어플리케이션 인터럽트 (도 2에 예시된 바와 같은 앱-인터럽트)의 실시형태이다. 어플리케이션이 멈추면 (halted), 어플리케이션에 의해 생성된 이미지의 컨텐츠는 그 컨텐츠가 변경되지 않는 정적 이미지, 예를 들면, 일시정지된 이미지일 수도 있다. 그 다음, 유저가 어플리케이션을 재개한 이후 (도 2에 예시된 바와 같은 앱-재개), 어플리케이션은, 어플리케이션에 의해 생성된 이미지들이 변경되고 있는 정상 상태로 리턴할 수도 있다, 예를 들면, 동적 이미지 상태 (38)로 다시 전이할 수도 있다. 몇몇 실시형태들에서, 어플리케이션이 어떤 기간 동안 일시정지된 상태로 유지되면, 어플리케이션은 기한만료 (expire) (도 2에 예시된 바와 같은 앱-기한만료) 되고 유저는 어플리케이션을 다시 정상 상태로 리턴시킬 수 없을 수도 있다. 그러나, 어플리케이션에 의해 생성된 정적 이미지는 디스플레이 (28) 상에 여전히 남아 있을 수도 있고, 따라서 정적 이미지 상태 (40)로 유지될 수도 있다.

**[0097]** 몇몇 실시형태들에서, 유저는 어플리케이션을 정지시킬 수도 있는데 (도 2에서 예시된 바와 같은 앱-정지), 이것은 디스플레이 (28)로 하여금 정적 이미지를 디스플레이하게 할 수도 있다. 어플리케이션의 정지는 디스플레이 (28)로 하여금 홈스크린을 제공하게 할 수도 있다. 예를 들면, 어플리케이션의 정지는 어플리케이션으로 하여금 멈추어서(halt), 홈스크린으로 빠져 나오게 할 수도 있다. 홈스크린의 컨텐츠가 일반적으로 정적이기 때문에, 홈스크린은 정적 이미지일 수도 있다.

**[0098]** 도 3a 및 도 3b는 GPU (14)의 실시형태들을 상세히 예시하는 블록도들이다. GPU (14)가 시스템 메모리 (26)의 일부 (32)로부터 정적 이미지 (30A)를 취출하고, 정적 이미지 (30A)를 리스캐일링하여 스케일링된 정적 이미지 (34)를 생성하고, 스케일링된 정적 이미지 (34)를 로컬 메모리 (16)에 저장할 수도 있게 하는 예시적인 기술들을 설명하기 위한 GPU (14)의 실시형태들이 도 3a 및 도 3b에 상세히 예시된다.

**[0099]** 도 3a에 예시된 바와 같이, 몇몇 실시형태들에서, GPU (14)가 범용 GPU (general purpose GPU; GPGPU)인 경우에서와 같이, GPU (14)는 테셀레이션 쉐이더 (tessellation shader; 42), 지오메트리 쉐이더 (44), 프리미티브 어셈블리 유닛 (primitive assembly unit; 46), 삼각형 셋업 유닛 (50) 및 프래그먼트 쉐이더 (fragment shader; 52)를 포함하는 레스터라이저 (rasterizer; 48), 심도 스텐실 (depth stencil; 56), 칼라링 및 블렌딩 유닛 (58), 및 디더 유닛 (60)을 포함하는 텍스쳐링 및 팩셀 쉐이더 (54), 텍스쳐들 및 필터들 (64)을 포함하는 텍스쳐 엔진 (62), 및 컴포지션 및 오버레이 유닛 (66)을 포함할 수도 있다. 도 3b에 예시된 GPU (14)의 실시형태에서, GPU (14)는 도 3a에 예시된 GPU의 컴포넌트들과 실질적으로 유사한 컴포넌트들을 포함할 수도 있다. 그러나, 도 3b의 실시형태에서, GPU (14)는 테셀레이션 쉐이더 (42) 또는 지오메트리 쉐이더 (44)를 포함하지 않을 수도 있다. 도 3b의 실시형태에서, GPU (14)는 라이팅 유닛 (70)과 정점 변환 및 어셈블리 유닛 (72)을 포함하는 프리미티브 프로세서 (68), 및 정점 쉐이더 (74)를 포함할 수도 있다.

**[0100]** 도 3a 및 도 3b에 예시된 GPU (14)의 예시적인 유닛들은 하드웨어 유닛들, 하드웨어 유닛들 상에서 실행되는 소프트웨어 유닛들, 또는 이들의 조합으로서 구현될 수도 있다. 또한, 도 3a 및 도 3b에 예시된 바와 같은 GPU (14)는 반드시 도 3a 및 도 3b에 예시된 유닛들을 모두 포함하지 않을 수도 있다. 또한, GPU (14)는 도 3a 및 도 3b에 예시된 것들에 추가로 유닛들을 포함할 수도 있다.

**[0101]** 도 3a의 실시형태에서, 테셀레이션 쉐이더 (42)는 디스플레이될 이미지를 프로세서 (12)로부터 수신할 수도 있다. 테셀레이션 쉐이더 (42)는 수신된 이미지를 직사각형들 또는 삼각형들과 같은 복수의 폴리곤들로 분할할 수도 있다. 지오메트리 쉐이더 (44)는 테셀레이션 쉐이더 (42)로부터 폴리곤들을 수신하고 수신된

폴리곤들을 더 분할할 수도 있다. 예를 들면, 지오메트리 쉐이더 (44)는 수신된 폴리곤들을 퍼미터브들로 분할할 수도 있다. 퍼미터브들은 포인트들, 라인들, 또는 삼각형들과 같은 폴리곤들일 수도 있다.

몇몇 실시형태들에서, 지오메트리 쉐이더 (44)는 삼각형들의 정점들 각각의 칼라 및 텍스쳐 좌표들, 각 포인트의 좌표들, 및 각 라인의 좌표들을 결정할 수도 있다. 예를 들면, 지오메트리 쉐이더 (74)는 텍스쳐 엔진 (62)의 텍스쳐들 및 필터들 (64)로부터 텍스쳐 좌표들을 수신할 수도 있다.

[0102] 도 3b의 실시형태에서, 퍼미터브 프로세서 (68)는 디스플레이될 이미지를 프로세서 (12)로부터 수신할 수도 있다. 이미지는 3차원 이미지일 수도 있다. 정점 변환 및 어셈블리 유닛 (72)은 이미지를 삼각형들과 같은 복수의 폴리곤들로 분할하고, 삼각형들의 정점들의 좌표를 월드 스페이스 좌표들 (world space coordinates)로 변환할 수도 있다. 라이팅 유닛 (70)은 이미지에 대한 광원들, 및 광원들로 인해 발생할 수도 있는 쉐이딩을 결정할 수도 있다. 정점 쉐이더 (74)는 퍼미터브 프로세서 (68)로부터 삼각형들을 수신하고 3차원 좌표들을 디스플레이 (28)의 2차원 좌표들로 변환할 수도 있다. 정점 쉐이더 (74)는 각 정점에 대한 심도 값을 또한 결정할 수도 있다. 몇몇 실시형태들에서, 정점 쉐이더 (74)는 정점들 각각의 칼라 및 텍스쳐 좌표들을 결정할 수도 있다. 예를 들면, 정점 쉐이더 (74)는 텍스쳐 엔진 (62)의 텍스쳐들 및 필터들 (64)로부터 텍스쳐 좌표들을 수신할 수도 있다.

[0103] 도 3a 또는 도 3b 중 어느 한 실시형태에서, 퍼미터브 어셈블리 유닛 (46)은 수신된 퍼미터브의 좌표들을 합성할 수도 있다. 예를 들면, 정점 쉐이더 (74)는 6개의 정점들에 대한 데이터를 출력할 수도 있다. 퍼미터브 어셈블리 유닛 (46)은 6개의 정점들을 2개의 삼각형들 (예를 들면, 삼각형 당 세 개의 정점들)로 합성할 수도 있다.

[0104] 도 3a 또는 도 3b 중 어느 한 실시형태에서, 레스터라이저 (48)는 디스플레이 (28)의 어느 픽셀들이 어느 삼각형들에 속하는지를 결정할 수도 있고, 픽셀들에 대한 칼라 값들을 결정할 수도 있다. 예를 들면, 삼각형 세팅 유닛 (50)은 퍼미터브 어셈블리 유닛 (46)으로부터 수신된 삼각형들에 대한 라인 방정식들 (line equations)을 계산하여 디스플레이 (28)의 어느 픽셀들이 삼각형 내에 있는지, 그리고 디스플레이 (28)의 어느 픽셀들이 삼각형 밖에 있는지를 결정할 수도 있다. 프래그먼트 쉐이더 (52)는 삼각형들 각각 내에 있는 디스플레이 (28)의 픽셀들 각각에 대한 칼라 값들을 결정할 수도 있다. 몇몇 실시형태들에서, 프래그먼트 쉐이더 (52)는 텍스쳐들 및 필터들 (64) 내의 값들에 기초하여 칼라 값들을 결정할 수도 있다.

[0105] 도 3a 또는 도 3b 중 어느 하나의 실시형태에서, 텍스쳐링 및 픽셀 쉐이더 (54)는 레스터라이저 (48)로부터 픽셀들 각각의 좌표 값들과 칼라 값들을 수신할 수도 있다. 심도 스텐실 (56)은 임의의 수신된 픽셀들이 부분적으로 또는 전적으로 임의의 다른 픽셀들에 의해 차단되는지의 여부를 결정하고, 완전히 차단된 픽셀들을 추가의 프로세싱에서 제거할 수도 있다. 칼라링 및 블렌딩 유닛 (58)은 상이한 픽셀들의 칼라들을 서로 혼합한다. 디터 유닛 (60)은 프로세싱 동안의 디테일의 손실을 다루기 위해 픽셀들의 칼라 심도를 증가시킬 수도 있다. 텍스쳐링 및 픽셀 쉐이더 (54)의 출력은, 텍스쳐링 및 픽셀 쉐이더 (54)가 컴포지션 및 오버레이 유닛 (66)으로 출력하는 그래픽 프로세싱 이미지일 수도 있다.

[0106] 도 3a 또는 도 3b 중 어느 한 실시형태에서, 컴포지션 및 오버레이 유닛 (66)은, 디터 유닛 (60)에 의해 생성된 이미지의 위에 오버레이되어야 할 필요가 있는 임의의 다른 이미지들이 존재하는지의 여부를 결정할 수도 있다. 예를 들면, 마우스 커서가 존재하는 경우, 컴포지션 및 오버레이 유닛 (66)은 디터 유닛 (60)에 의해 생성된 이미지의 위에 마우스 커서를 오버레이시킬 수도 있다. 결과적으로 생성되는 이미지는 시스템 메모리 (26)의 일부 (32)에 저장된 이미지의 일부 실시형태, 예를 들면, 이미지 (30)일 수도 있다. 이미지 (30)의 컨텐츠가 정의된 기간 동안 변경되지 않으면, 이미지 (30)은 정적 이미지 (30A)인 것으로 결정될 수도 있다.

[0107] 몇몇 실시형태들에서, GPU (14)의 텍스쳐링 및 픽셀 쉐이더 (54)는 시스템 메모리 (26)의 일부 (32)로부터 정적 이미지 (30A)를 취출하고, 정적 이미지 (30A)의 한 버전, 예를 들면, 정적 이미지 (30A) 그 자체, 또는 스케일링된 정적 이미지 (34)를 로컬 메모리 (16)에 저장할 수도 있다. 텍스쳐링 및 픽셀 쉐이더 (54)는 정적 이미지 (30A)를 스케일링하여 스케일링된 정적 이미지 (34)를 생성하기에 적합할 수도 있는데, 그 이유는, 몇몇 실시형태들에서, 텍스쳐링 및 픽셀 쉐이더 (54)가 다른 그래픽 관련 목적들을 위해 스케일링 유닛을 포함할 수도 있기 때문이다. GPU (14)는 정적 이미지 (30A)를 스케일링하여 스케일링된 정적 이미지 (34)를 생성하기 위해 텍스쳐링 및 픽셀 쉐이더 (54)의 스케일링 유닛을 활용할 수도 있다.

[0108] 도 4는 본 개시에 따른 하나 이상의 프로세싱 유닛들의 예시적인 동작을 예시하는 흐름도이다. 예시의 목적들을 위해, 도 1a 내지 도 1d, 도 3a, 및 도 3b를 참조한다.

[0109]

하나 이상의 프로세싱 유닛들, 예를 들면, 프로세서 (12)는, 시스템 메모리 (26)의 일부 (32)에 저장된 이미지 (30)가 정적 이미지인지 또는 비정적 이미지인지를 결정할 수도 있다 (74). 예를 들면, 위에서 설명된 바와 같이, 프로세서 (12)는, 임의의 컴포넌트, 예컨대 GPU (14), 비디오 프로세서 유닛 (22), 코덱 (20), 또는 어플리케이션 테이터 무버 (24)가 정의된 기간 내에 이미지 (30)의 컨텐츠를 변경하는 임의의 새로운 정보를 제공했는지의 여부를 결정하기 위해 시스템 메모리 (26)의 일부 (32)의 컨텐츠를 모니터링할 수도 있다. 시스템 메모리 (26)의 일부 (32)가 정의된 기간 내에 이미지 (30)의 컨텐츠를 변경하는 임의의 새로운 정보를 수신하지 않았다면, 프로세서 (12)는 이미지 (30)가 정적 이미지, 예를 들면, 정적 이미지 (30A)라고 결정할 수도 있다. 몇몇 실시형태들에서, 프로세서 (12)는 디바이스 (10)의 환경에서 어떤 변화가 있었는지의 여부를 또한 결정할 수도 있다. 예를 들면, 프로세서 (12)는 주변 라이팅에서, 디바이스 (10)의 디바이스 방위에서 임의의 변경이 있었는지, 또는 다른 외부 장치와의 디바이스 (10)의 접속에서 변경들이 있었는지의 여부를 결정할 수도 있다. 디바이스 (10)의 환경에서 어떠한 변경도 없었고, 이미지 (30)의 컨텐츠를 변경하는 새로운 정보를 어떠한 컴포넌트도 제공하지 않았다면, 프로세서 (12)는 이미지 (30)가 정적 이미지, 예를 들면, 정적 이미지 (30A)라고 결정할 수도 있다.

[0110]

이미지 (30)가 정적 이미지 (30A)라고 프로세서 (12)가 결정하면, GPU (14)는 시스템 버스 (15)를 통해 시스템 메모리 (26)의 일부 (32)로부터 정적 이미지 (30A)를 취출할 수도 있다 (76). GPU (14)는 정적 이미지 (30A)를 스케일링하여 정적 이미지 (30A)의 감소된 공간 해상도 버전, 예를 들면, 스케일링된 정적 이미지 (34)를 생성할 수도 있다 (78). 일 예로서, 텍스쳐 및 픽쳐 쉐이더 (54)와 같은 GPU (14)의 쉐이더는 정적 이미지 (30A)를 스케일링할 수도 있다. 몇몇 실시형태들에서, GPU (14)는 로컬 메모리 (16)에서 이용 가능한 저장 공간의 양에 기초하여 정적 이미지 (30A)를 스케일링할 수도 있다. GPU (14)는 스케일링된 정적 이미지 (34)를 로컬 메모리 (16)에 저장할 수도 있다 (80). 몇몇 실시형태들에서, GPU (14)는 GPU (14)로부터의 정보를 저장하기 위해 예약된 로컬 메모리 (16)의 일부에 스케일링된 정적 이미지 (34)를 저장할 수도 있다.

[0111]

디스플레이 프로세서 (18)는 스케일링된 정적 이미지 (34), 예를 들면, 정적 이미지 (30A)의 감소된 공간 해상도 버전을 로컬 메모리 (16)로부터 취출할 수도 있다 (82). 디스플레이 프로세서 (18)는 스케일링된 정적 이미지 (34)를 리스케일링하여 리스케일링된 이미지 (36)를 생성할 수도 있다 (84). 디스플레이 프로세서 (18)는 리스케일링된 이미지 (36)를 표시를 위해 디스플레이 (28)로 출력할 수도 있다 (86).

[0112]

하나 이상의 실시형태들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 상기 기능들은 비일시적 컴퓨터 판독가능 매체를 포함하는 제조물품 상에 하나 이상의 명령들 또는 코드로서 저장될 수도 있다. 컴퓨터 판독가능 매체들은 컴퓨터 데이터 저장 매체들을 포함할 수도 있다. 데이터 저장 디바이스는 본 개시에서 설명된 기법들의 구현을 위한 명령들, 코드, 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체들일 수도 있다. 비제한적인 실시형태로서, 그러한 컴퓨터 판독 가능 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 스토리지 디바이스들, 플래시 메모리, 또는 명령들이나 데이터 구조들의 형태로 원하는 프로그램 코드를 전하거나 저장할 수 있고, 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 본원에서 사용된 바와 같은 디스크 (disk) 와 디스크 (disc) 는, 컴팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피디스크 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 는 통상 자기적으로 데이터를 재생하고, 디스크 (disc) 는 레이저를 이용하여 광학적으로 데이터를 재생한다. 위의 조합들도 컴퓨터 판독가능 매체들의 범위 내에 또한 포함되어야 한다.

[0113]

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 모듈들에 의해 제공될 수도 있다. 또한, 본 기술들은 하나 이상의 회로들 또는 로직 소자들에서 완전히 구현될 수 있다.

[0114]

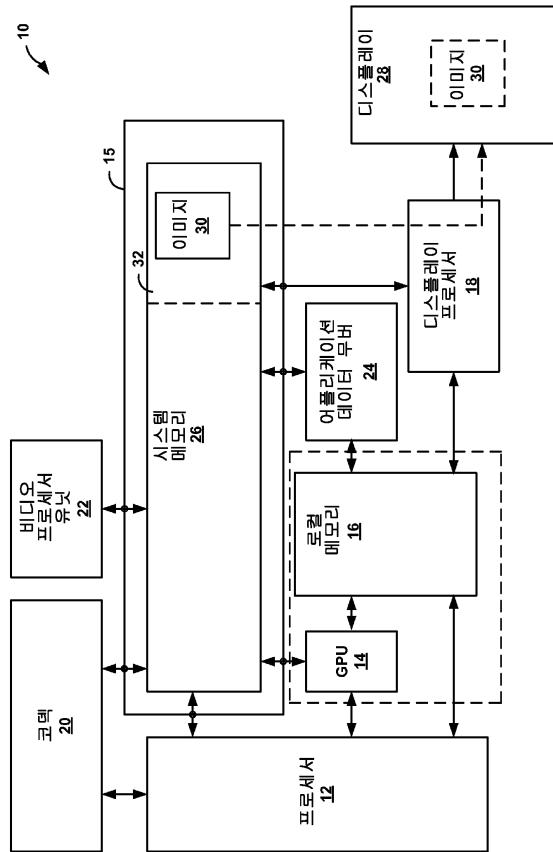
본 개시의 기술들은, 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예를 들면, 칩셋)를 포함하는 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 개시된 기술들을 수행하도록 구성된 디바이스들의 기능적 양태를 강조하기 위해 다양한 소자들, 모듈들, 또는 유닛들이 본 개시에서 설명되었지만, 반드시 상이한 하드웨어 유닛들에 의해 실현될 필요는 없다. 대신, 상술한 바와 같이, 다양한 유닛들은, 적절한 소프트웨어 및/또는 펌웨어와 연계하여, 하드웨어 유닛에 통합되거나 또는 상술한 하나 이상의 프로세서들을 포함하는, 상호 동작적

인 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

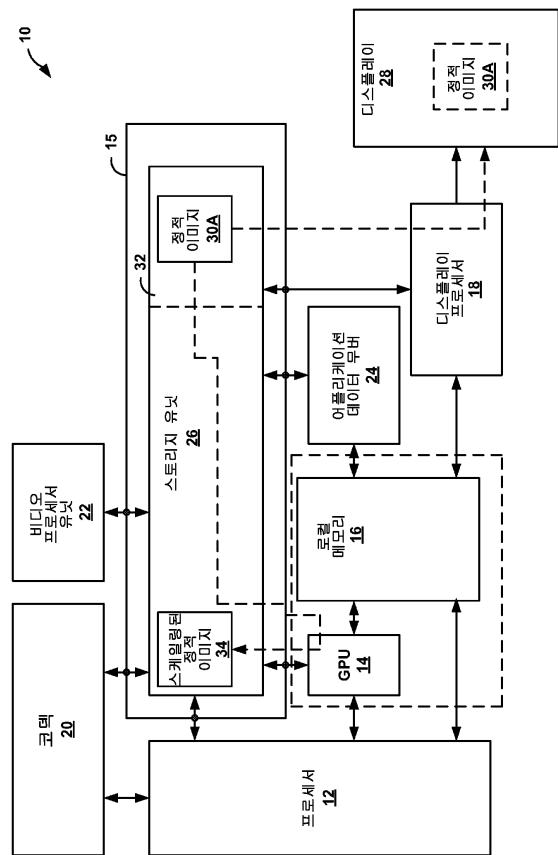
다양한 실시형태들을 설명하였다. 이들 및 다른 실시형태들은 하기의 특허청구범위 내에 있다.

## 도면

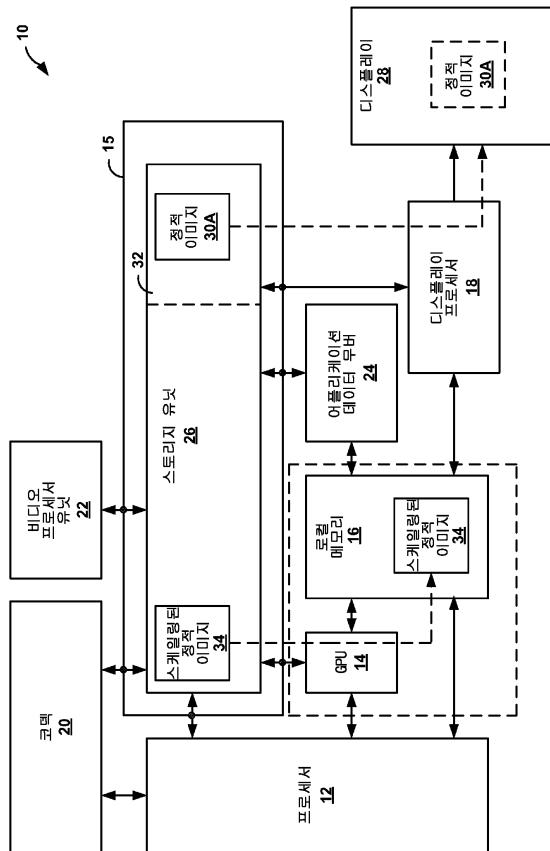
## 도면1a



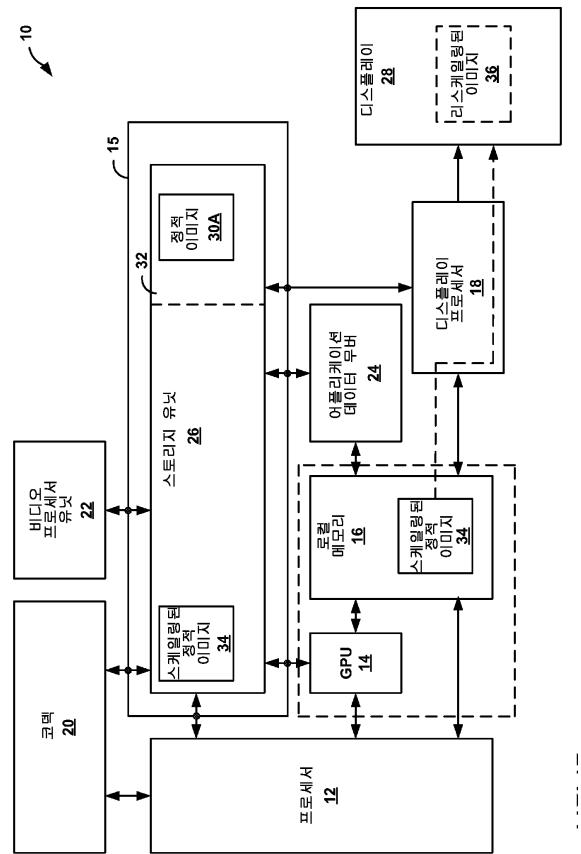
도면1b



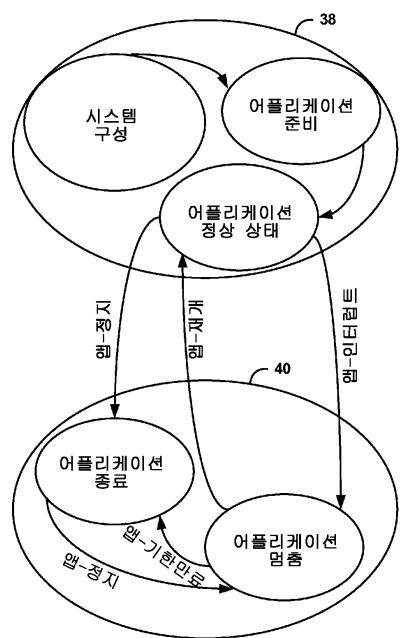
도면1c



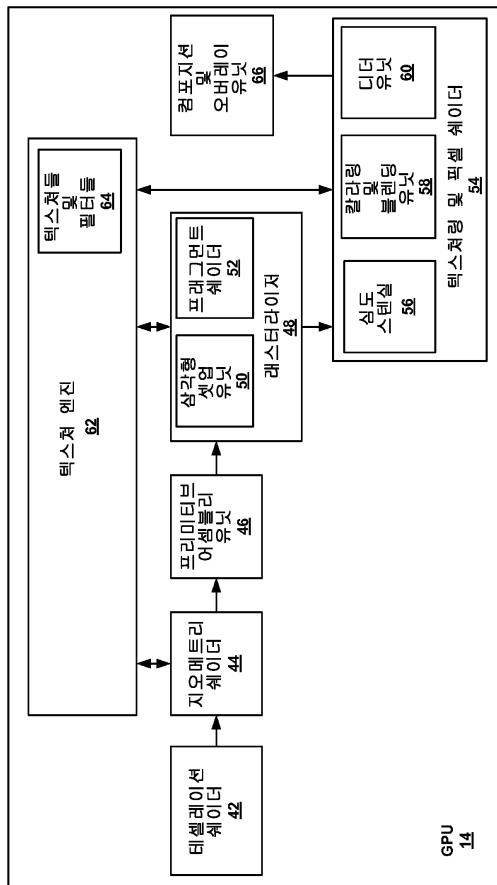
도면1d



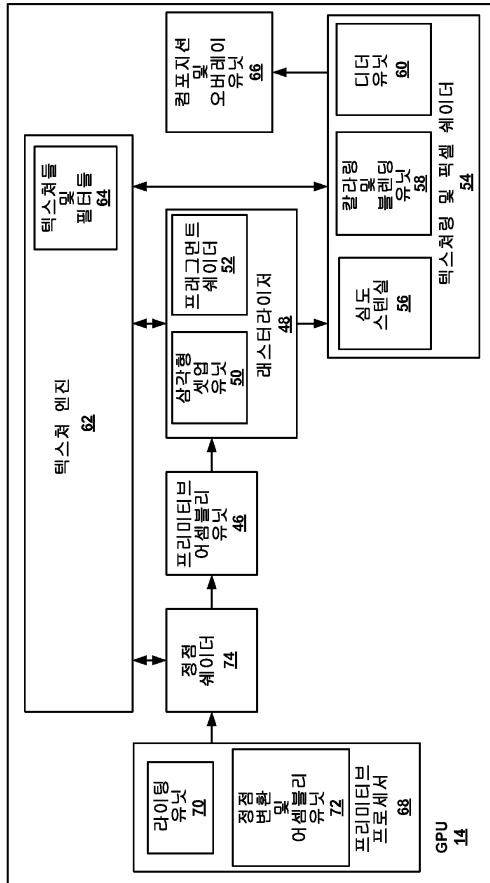
도면2



### 도면3a



도면3b



도면4

