



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0078707
(43) 공개일자 2022년06월10일

- (51) 국제특허분류(Int. Cl.)
G06V 10/24 (2022.01) G05D 1/00 (2006.01)
G06N 3/04 (2006.01) G06N 3/08 (2006.01)
G06V 10/44 (2022.01) G06V 20/56 (2022.01)
G06V 30/194 (2022.01) G10L 15/16 (2006.01)
- (52) CPC특허분류
G06V 10/242 (2022.01)
G05D 1/0088 (2013.01)
- (21) 출원번호 10-2022-7016207
(22) 출원일자(국제) 2020년10월26일
심사청구일자 2022년05월13일
(85) 번역문제출일자 2022년05월13일
(86) 국제출원번호 PCT/US2020/057408
(87) 국제공개번호 WO 2021/086796
국제공개일자 2021년05월06일
(30) 우선권주장
16/667,708 2019년10월29일 미국(US)
- (71) 출원인
엔비디아 코포레이션
미국 캘리포니아 95051 산타 클라라 산 토마스 익스프레스웨이 2788
- (72) 발명자
트렘블레이, 조나단
미국 95051 캘리포니아주 산타 클라라 산 토마스 익스프레스웨이 2788
리우, 밍-위
미국 95051 캘리포니아주 산타 클라라 산 토마스 익스프레스웨이 2788
(뒷면에 계속)
- (74) 대리인
양영준, 백만기

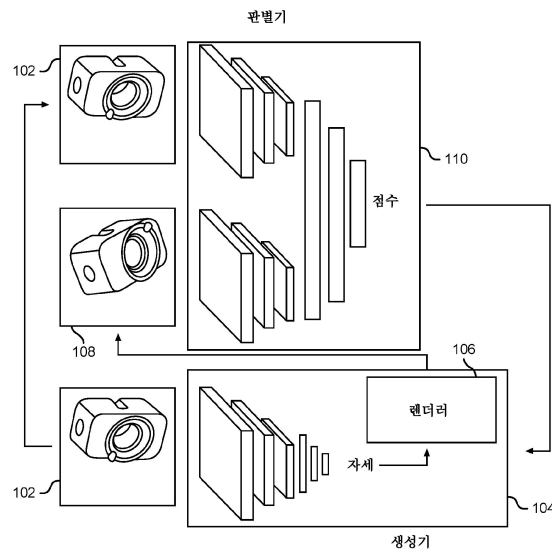
전체 청구항 수 : 총 24 항

(54) 발명의 명칭 머신 러닝을 이용한 이미지로부터의 객체 배향 결정

(57) 요약

이미지 내의 객체들의 배향을 결정하는 장치들, 시스템들, 및 기법들. 적어도 하나의 실시예에서, 이미지들은 객체의 배향을 결정하기 위해 훈련된 신경망을 사용하여 처리된다.

대표도 - 도1



(52) CPC특허분류

G06N 3/0454 (2013.01)

G06N 3/08 (2013.01)

G06V 10/454 (2022.01)

G06V 20/56 (2022.01)

G06V 30/194 (2022.01)

G10L 15/16 (2013.01)

G06T 2207/20084 (2013.01)

(72) 발명자

폭스, 디에터

미국 95051 캘리포니아주 산타 클라라 산 토마스
익스프레스웨이 2788

암미라토, 필립

미국 95051 캘리포니아주 산타 클라라 산 토마스
익스프레스웨이 2788

명세서

청구범위

청구항 1

프로세서로서, 2개 이상의 이미지 내의 제1 객체의 배향의 차이를 최소화하도록 훈련된 하나 이상의 신경망에 적어도 부분적으로 기초하여 하나 이상의 이미지를 수신하는 것에 응답하여 상기 하나 이상의 이미지 내의 하나 이상의 객체의 배향을 결정하는 하나 이상의 회로를 포함하는, 프로세서.

청구항 2

제1항에 있어서,

상기 2개 이상의 이미지는 상기 하나 이상의 이미지를 포함하고;

상기 2개 이상의 이미지는 상기 하나 이상의 신경망에 의해 결정된 배향에서의 하나 이상의 객체의 수학적 모델로부터 생성된 상기 하나 이상의 객체의 이미지를 포함하고;

상기 하나 이상의 신경망은 상기 2개 이상의 이미지 사이의 차이의 측정치를 생성하는 추가적인 신경망에 의해 생성된 값에 적어도 부분적으로 기초하는 보상을 사용하여 훈련되는, 프로세서.

청구항 3

제2항에 있어서, 상기 수학적 모델로부터 생성된 상기 하나 이상의 객체의 이미지는 미분불가능 렌더러에 의해 생성되는, 프로세서.

청구항 4

제1항에 있어서, 상기 하나 이상의 객체들은 텍스처 없는 대칭적인 객체를 포함하는, 프로세서.

청구항 5

제2항에 있어서, 상기 2개 이상의 이미지들 사이의 차이의 측정치를 생성하는 상기 추가적인 신경망은 이미지 및 상기 이미지의 교란된 버전을 상기 추가적인 신경망에 제공함으로써 훈련되는, 프로세서.

청구항 6

제2항에 있어서, 상기 하나 이상의 신경망은 상기 추가적인 신경망을 최대로 혼동시키도록 조정되는, 프로세서.

청구항 7

제2항에 있어서, 상기 수학적 모델로부터 생성된 상기 하나 이상의 객체의 이미지는 하나 이상의 데이터 증강에 따라 생성되는, 프로세서.

청구항 8

제7항에 있어서, 상기 하나 이상의 데이터 증강은 상기 이미지 내의 상기 하나 이상의 객체에 대한 광 위치, 조명 강도, 또는 정반사(specular reflection)를 포함하는, 프로세서.

청구항 9

프로세서에 있어서, 제2 배향을 갖는 객체의 제1 이미지에 적어도 부분적으로 기초하여 제1 배향을 갖는 객체의 제2 이미지를 생성하는 것을 돕고 상기 제2 배향과 동등하게 관찰될 때 때까지 제1 배향을 조정하는 하나 이상의 회로를 포함하는, 프로세서.

청구항 10

제9항에 있어서,

상기 제1 배향은 상기 제1 이미지를 사용하여 제1 신경망에 의해 결정되고;

상기 제1 배향은 상기 제1 이미지와 제2 이미지 사이의 차이에 적어도 부분적으로 기초하여 조정되고;

상기 제2 이미지는 상기 제1 배향에 있는 상기 객체의 수학적 모델을 사용하여 생성되는, 프로세서.

청구항 11

제10항에 있어서, 상기 제2 이미지는 미분불가능 렌더러를 사용하여 생성되는, 프로세서.

청구항 12

제10항에 있어서, 상기 제1 이미지와 상기 제2 이미지 사이의 차이는 제2 신경망을 사용하여 결정되는, 프로세서.

청구항 13

제12항에 있어서, 상기 제2 신경망은 제3 이미지 및 상기 제3 이미지의 교란된 버전을 상기 제2 신경망에 제공함으로써, 훈련되는 프로세서.

청구항 14

제10항에 있어서, 상기 제2 이미지는 상기 객체의 광 위치, 조명 강도 또는 정반사를 포함하는 하나 이상의 데이터 증강에 따라 생성되는, 프로세서.

청구항 15

시스템으로서,

제2 이미지 및 제3 이미지 내의 객체의 배향의 차이에 적어도 부분적으로 기초하여 훈련된 제1 신경망을 이용하여 제1 이미지를 수신하는 것에 응답하여 상기 제1 이미지 내의 객체의 배향을 결정하는 하나 이상의 프로세서; 및

상기 제1 신경망을 저장하기 위한 하나 이상의 메모리를 포함하는, 시스템.

청구항 16

제15항에 있어서,

제2 이미지는 상기 제1 신경망에 의해 결정된 배향에서 객체의 모델로부터 생성되고;

상기 제1 신경망은 차이를 추정하는 추가적인 신경망에 의해 생성된 값에 적어도 부분적으로 기초하여 훈련되는, 시스템.

청구항 17

제16항에 있어서, 상기 추가적인 신경망은, 상기 추가적인 신경망에, 제1 배향의 상기 객체의 이미지 및 상기 제1 배향을 섭동함으로써 결정된 제2 배향의 상기 객체의 이미지를 제공함으로써 훈련되는, 시스템.

청구항 18

제16항에 있어서, 상기 하나 이상의 신경망은 상기 추가적인 신경망을 최대로 혼동시키도록 조정되는, 시스템.

청구항 19

제16항에 있어서, 상기 모델로부터 생성된 상기 객체의 상기 제1 이미지는 하나 이상의 데이터 증강에 따라 생성되는, 시스템.

청구항 20

제19항에 있어서, 상기 하나 이상의 데이터 증강은 상기 제1 이미지에 대한 광학적 변경들을 포함하는, 시스템.

청구항 21

자율 차량에 있어서,

객체의 이미지를 획득하는 상기 자율 차량에 부착된 카메라;

2개 이상의 이미지 내의 제1 객체의 배향의 차이를 최소화하도록 훈련된 하나 이상의 신경망에 적어도 부분적으로 기초하여 이미지를 수신하는 것에 응답하여 상기 이미지 내의 객체의 배향을 결정하는 하나 이상의 회로를 포함하는 프로세서; 및

상기 객체의 결정된 배향에 적어도 부분적으로 기초하여 상기 자율 차량을 제어하는 내비게이션 시스템을 포함하는, 자율 차량.

청구항 22

제21항에 있어서,

상기 2개 이상의 이미지는 상기 하나 이상의 신경망에 의해 결정된 배향으로 객체의 모델로부터 생성된 상기 객체의 이미지를 포함하고;

상기 하나 이상의 신경망은 상기 2개 이상의 이미지 사이의 차이의 측정치를 생성하는 추가적인 신경망에 의해 생성된 값에 적어도 부분적으로 기초하여 훈련되는, 자율 차량.

청구항 23

제22항에 있어서, 상기 객체의 이미지는 하나 이상의 데이터 증강에 따라 생성되는, 프로세서.

청구항 24

제23항에 있어서, 상기 하나 이상의 데이터 증강은 광 위치, 조명 강도, 또는 정반사 파라미터를 포함하는, 프로세서.

발명의 설명

기술 분야

[0001] 관련 출원에 대한 상호 참조

[0002] 본 출원은 2019년 10월 29일자로 출원된, 발명의 명칭이 "DETERMINING OBJECT ORIENTATION FROM AN IMAGE WITH MACHINE LEARNING"인 미국 특허 출원 제16/667,708호에 대한 우선권을 주장하며, 이 미국 특허 출원의 전체 내용은 그 전체가 모든 목적을 위해 본 명세서에 참고로 포함된다.

[0003] 분야

[0004] 적어도 하나의 실시예는 이미지로부터 객체의 배향을 결정하는 데 사용되는 처리 리소스들에 관한 것이다. 예를 들어, 적어도 하나의 실시예는 객체의 배향을 추정하는 신경망들을 훈련하기 위해 사용되는 프로세서들 또는 계산 시스템들에 관한 것이다.

배경 기술

[0005] 적어도 하나의 실시예는 이미지로부터 객체의 배향을 결정하는 데 사용되는 처리 리소스들에 관한 것이다. 예를 들어, 적어도 하나의 실시예는 객체의 배향을 추정하는 신경망들을 훈련하기 위해 사용되는 프로세서들 또는 계산 시스템들에 관한 것이다.

도면의 간단한 설명

[0006] 도 1은 적어도 하나의 실시예에 따른, 이미지 판별기를 사용하여 객체의 자세를 추정하기 위해 신경망을 훈련시키는 시스템의 예를 예시한다.

도 2는 적어도 하나의 실시예에 따른, 다양한 대칭도를 갖는 다양한 객체의 예를 예시한다.

도 3은 적어도 하나의 실시예에 따라 판별기를 훈련하는 프로세스의 예를 예시한다.

도 4는 적어도 하나의 실시예에 따른, 생성 네트워크를 훈련하는 프로세스의 예를 예시한다.

도 5는 적어도 하나의 실시예에 따른, 판별기 출력 대 20도 자세의 객체의 이미지에 대한 자세의 예를 예시한다.

도 6은 적어도 하나의 실시예에 따른, 판별기 출력 대 70도 자세의 객체의 이미지에 대한 자세의 예를 예시한다.

도 7은 적어도 하나의 실시예에 따른, 생성기에 의해 생성된 이미지의 일 예를 예시한다.

도 8은 적어도 하나의 실시예에 따른, 도 7에 도시된 객체에 대해 생성된 생성기 및 판별기 출력의 예를 예시한다.

도 9는 적어도 하나의 실시예에 따른, 생성기에 의해 생성된 이미지의 일 예를 예시한다.

도 10은 적어도 하나의 실시예에 따른, 도 9에 도시된 객체에 대해 생성된 생성기 및 판별기 출력의 예를 예시한다.

도 11은 적어도 하나의 실시예에 따른, 생성기에 의해 생성된 이미지의 일 예를 예시한다.

도 12는 적어도 하나의 실시예에 따른, 도 11에 도시된 객체에 대해 생성된 생성기 및 판별기 출력의 예를 예시한다.

도 13은 적어도 하나의 실시예에 따른, 생성기에 의해 생성된 이미지의 예를 예시한다.

도 14는 적어도 하나의 실시예에 따른, 도 13에 도시된 객체에 대해 생성된 생성기 및 판별기 출력의 예를 예시한다.

도 15는 적어도 하나의 실시예에 따른, 상이한 모델들이 3-축 대칭을 갖는 객체의 회전에 어떻게 반응하는지의 예를 예시한다.

도 16은 적어도 하나의 실시예에 따른, 상이한 모델들이 반대칭 객체의 회전에 어떻게 반응하는지의 예를 예시한다.

도 17은 적어도 하나의 실시예에 따른, 상이한 모델들이 반대칭 객체들의 그룹의 회전에 어떻게 반응하는지의 예를 예시한다.

도 18은 적어도 하나의 실시예에 따른, 상이한 모델들이 2-축 대칭을 갖는 객체의 회전에 어떻게 반응하는지의 예를 예시한다.

도 19는 적어도 하나의 실시예에 따른, 이미지들의 쌍들에 대한 판별기 출력들의 시각화의 예를 예시한다.

도 20 내지 도 28은 적어도 하나의 실시예에 따른, T-LESS 데이터 세트에 대한 시점 예측기의 정성적 결과의 일 예를 예시한다.

도 29a는 적어도 하나의 실시예에 따른 추론 및/또는 훈련 로직을 예시한다.

도 29b는 적어도 하나의 실시예에 따른 추론 및/또는 훈련 로직을 예시한다.

도 30은 적어도 하나의 실시예에 따른, 신경망의 훈련 및 배치를 예시한다.

도 31은 적어도 하나의 실시예에 따른 예시적인 데이터 센터 시스템을 예시한다.

도 32a는 적어도 하나의 실시예에 따른 자율 차량의 예를 예시한다.

도 32b는 적어도 하나의 실시예에 따른, 도 32a의 자율 차량에 대한 카메라 위치 및 시야의 예를 예시한다.

도 32c는 적어도 하나의 실시예에 따른, 도 32a의 자율 차량에 대한 예시적인 시스템 아키텍처를 나타내는 블록도이다.

도 32d는 적어도 하나의 실시예에 따른, 도 32a의 클라우드 기반 서버(들)와 자율 차량 사이의 통신을 위한 시스템을 예시하는 도면이다.

도 33은 적어도 하나의 실시예에 따른, 컴퓨터 시스템을 나타내는 블록도이다.

도 34는 적어도 하나의 실시예에 따른 컴퓨터 시스템을 나타내는 블록도이다.

도 35는 적어도 하나의 실시예에 따른 컴퓨터 시스템을 예시한다.

도 36은 적어도 하나의 실시예에 따른, 컴퓨터 시스템을 예시한다.

도 37a는 적어도 하나의 실시예에 따른 컴퓨터 시스템을 예시한다.

도 37b는 적어도 하나의 실시예에 따른 컴퓨터 시스템을 예시한다.

도 37c는 적어도 하나의 실시예에 따른 컴퓨터 시스템을 예시한다.

도 37d는 적어도 하나의 실시예에 따른 컴퓨터 시스템을 예시한다.

도 37e 및 도 37f는 적어도 하나의 실시예에 따른, 공유 프로그래밍 모델을 예시한다.

도 38은 적어도 하나의 실시예에 따른, 예시적인 집적 회로들 및 연관된 그래픽 프로세서들을 예시한다.

도 39a 및 도 39b는 적어도 하나의 실시예에 따른, 예시적인 집적 회로들 및 연관된 그래픽 프로세서들을 예시한다.

도 40a 및 도 40b는 적어도 하나의 실시예에 따른 추가적인 예시적인 그래픽 프로세서 로직을 예시한다.

도 41은 적어도 하나의 실시예에 따른, 컴퓨터 시스템을 예시한다.

도 42a는 적어도 하나의 실시예에 따른 병렬 프로세서를 예시한다.

도 42b는 적어도 하나의 실시예에 따른 파티션 유닛을 예시한다.

도 42c는 적어도 하나의 실시예에 따른 처리 클러스터를 예시한다.

도 42d는 적어도 하나의 실시예에 따른 그래픽 멀티프로세서를 예시한다.

도 43은 적어도 하나의 실시예에 따른 다중-그래픽 처리 유닛(GPU) 시스템을 예시한다.

도 44는 적어도 하나의 실시예에 따른 그래픽 프로세서를 예시한다.

도 45는 적어도 하나의 실시예에 따른, 프로세서에 대한 프로세서 마이크로-아키텍처를 예시하는 블록도이다.

도 46은 적어도 하나의 실시예에 따른 딥 러닝 애플리케이션 프로세서를 예시한다.

도 47은 적어도 하나의 실시예에 따른 예시적인 뉴로모픽 프로세서를 예시하는 블록도이다.

도 48은 하나 이상의 실시예들에 따라, 그래픽 프로세서의 적어도 일부들을 예시한다.

도 49는 하나 이상의 실시예들에 따라, 그래픽 프로세서의 적어도 일부들을 예시한다.

도 50은 하나 이상의 실시예들에 따라, 그래픽 프로세서의 적어도 일부들을 예시한다.

도 51은 적어도 하나의 실시예에 따른 그래픽 프로세서의 그래픽 처리 엔진(5110)의 블록도이다.

도 52는 적어도 하나의 실시예에 따른, 그래픽 프로세서 코어의 적어도 일부의 블록도이다.

도 53a 및 도 53b는 적어도 하나의 실시예에 따른, 그래픽 프로세서 코어의 처리 엘리먼트들의 어레이를 포함하는 스레드 실행 로직(5300)을 예시한다.

도 54는 적어도 하나의 실시예에 따른 병렬 처리 유닛("PPU")을 예시한다.

도 55는 적어도 하나의 실시예에 따른, 일반 처리 클러스터(general processing cluster)("GPC")를 예시한다.

도 56은 적어도 하나의 실시예에 따른 병렬 처리 유닛("PPU")의 메모리 파티션 유닛을 나타낸다.

도 57은 적어도 하나의 실시예에 따른 스트리밍 멀티-프로세서를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0007] 적어도 하나의 실시예는 이미지로부터 객체의 배향을 결정하는 데 사용되는 처리 리소스들에 관한 것이다. 예를 들어, 적어도 하나의 실시예는 객체의 배향을 추정하는 신경망들을 훈련하기 위해 사용되는 프로세서들 또는 계산 시스템들에 관한 것이다.

[0008] 적어도 하나의 실시예에서, 로봇 공학 시스템에서의 문제는 장면 내의 객체들의 3D 위치 및 배향(방위각, 고도

및 사이클로로테이션(cyclorotation))을 아는 것이다. 적어도 하나의 실시예에서, 3D 배향은 6D 자세의 서브세트이고, 종종 3D 변환 및 3D 배향 양자 모두를 포함하는 6-DoF(자유도) 자세라고 지칭된다. 적어도 하나의 실시예에서, 배향 검출기들은 차단, 조명 및 클러터와 같은 다양한 인자들에 강건하다. 적어도 하나의 실시예에서, 대칭 배향은 주목을 덜 받아왔고, 과제에 남아 있다.

[0009] 적어도 하나의 실시예에서, 명시적인 대칭 라벨링 또는 손실 사양을 필요로 하지 않고 대칭적인 객체들의 배향을 검출하는 새로운 접근법이 제공된다. 적어도 하나의 실시예에서, 제안된 방법은 객체의 배향을 추정하기 위해 관심 객체의 CAD 3d 모델을 사용한다. 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 적대적 컴포넌트가 객체의 대칭들과 일치하는 손실 함수를 제공하도록 학습하는 적대적 훈련 패러다임을 사용하여 시점 회귀기를 훈련하는데 사용될 수 있다. 적어도 하나의 실시예에서, 생성기의 역할을 하는 3D 렌더러와 결합된 시점 회귀기는 판별기와 동시에 훈련된다. 적어도 하나의 실시예에서, 판별기의 목표는 생성기가 그 예측을 개선하도록 도전하게 하는 것이다. 적어도 하나의 실시예에서, 생성기는 관심 객체의 크로핑된 이미지를 입력으로서 취하고 예측된 자세를 반환한다. 적어도 하나의 실시예에서, 판별기는 입력으로서 2개의 이미지, 즉 크로핑된 이미지(생성기에 사용된 것과 유사한 입력) 및 생성기에 의해 출력된 배향에서의 관심 객체의 3D 렌더링을 취한다. 적어도 하나의 실시예에서, 판별기는 시각적 유사성에 기초하여 이미지 쌍을 점수화한다. 적어도 하나의 실시예에서, 렌더링된 이미지로부터의 경사가 이용 가능하지 않기 때문에, 무작위 샘플링 방법을 이용하여 판별기 및 생성기를 공동으로 훈련시킨다.

[0010] 적어도 하나의 실시예에서, 메트릭이 예측된 자세에서의 3D 모델 상의 각각의 포인트를 실측 자료 자세(ground truth pose)의 최근접 포인트와 일치시키기 때문에, 본 명세서에 설명된 기법들은 대략 대칭 형상을 갖는 객체들에 대한 자세를 결정하는데 탁월하다. 적어도 하나의 실시예에서, 객체의 크롭이 인코딩된 다음, 최종 배향을 검색하기 위해 최근접 이웃 검색을 통해 인코딩된 자세들의 사전과 비교된다. 적어도 하나의 실시예에서, 자세 결정은 기술적으로 연속적이고, 신경망은 자세를 직접 출력한다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 로봇이 조작 목적들을 위해, 특히 머그잔(mug)들과 같은 대칭성들을 갖는 객체들에 대해 알려진 객체들의 자세들을 추정할 수 있을 필요가 있는 로봇 공학 응용들에 중요하다.

[0011] 적어도 하나의 실시예에서, 객체의 자세를 예측하기 위해 컴퓨터 비전 시스템을 훈련시키는 것은 로봇 조작을 개선하는 데 결정적이며, 따라서, 로봇들이 객체들을 쉽게 위치확인한 이후에 파지할 수 있게 한다. 적어도 하나의 실시예에서, 자세 추정에서의 과제들은 라벨링된 데이터를 획득하고 대칭성들을 갖는 객체들을 핸들링하는 것에 있다. 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 관심 객체의 3D 모델을 사용하는 비지도 훈련 패러다임을 구현한다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 적대적 훈련 프레임워크에서, 단순히 RGB 이미지를 소비하는, 배향 검출기를 성공적으로 훈련시킬 수 있고, 여기서 판별기는 블랙박스 미분불가능 렌더러(black-box non-differentiable renderer)를 사용하여 객체 배향을 검색하기 위해 학습 신호를 제공하도록 학습한다. 적어도 하나의 실시예에서, 이러한 미분불가능성을 극복하기 위해, 랜덤화된 샘플링 방법이 훈련 경사들을 획득하기 위해 사용된다. 적어도 하나의 실시예에서, 대칭적인 객체들을 핸들링할 수 있는 시점 검출기를 성공적으로 훈련하기 위해 적대적 프레임워크가 이용된다. 적어도 하나의 실시예에서, 텍스처가 없고 대칭적인 객체들에 대한 도전적인 데이터 세트인 T-LESS 상의 3D 배향 예측에 대한 최신 기술의 결과들이 도시된다.

[0012] 도 1은 적어도 하나의 실시예에 따른, 이미지 판별기를 사용하여 객체의 자세를 추정하기 위해 신경망을 훈련시키는 시스템의 예를 예시한다. 적어도 하나의 실시예에서, 객체의 이미지(102)가 객체의 자세를 추정하는 생성기(104)에 제공된다. 적어도 하나의 실시예에서, 이미지(102)는 카메라, 스캐너, LIDAR, RADAR, 또는 다른 이미징 시스템에 의해 포착될 수 있다. 적어도 하나의 실시예에서, 생성기(104)는 객체의 자세를 추정하도록 훈련가능한 신경망이다. 적어도 하나의 실시예에서, 자세는 객체의 배향 및/또는 위치를 포함한다.

[0013] 적어도 하나의 실시예에서, 훈련 동안, 생성기(104)에 의해 생성된 자세 추정이 렌더러(106)에 제공된다. 적어도 하나의 실시예에서, 렌더러(106)는, 컴퓨터 시스템의 하나 이상의 프로세서 상에서 실행된 결과로서, 상기 컴퓨터 시스템으로 하여금 지정된 자세로 객체의 이미지(108)를 생성하게 하는 실행가능한 명령어들로 구성된다. 적어도 하나의 실시예에서, 렌더러(106)는 상기 객체의 수학적 모델을 이용하여 상기 객체의 이미지(108)를 생성한다. 적어도 하나의 실시예에서, 이미지(108)가 판별기(110)에 제공된다. 적어도 하나의 실시예에서, 판별기(110)는 이미지(108)와 이미지(102)를 비교하고, 2개의 이미지 사이의 차이들을 나타내는 점수를 생성한다. 적어도 하나의 실시예에서, 상기 점수는 생성기(104) 내의 신경망에 대한 보상 또는 훈련 입력으로서 사용된다.

- [0014] 적어도 하나의 실시예에서, 자세 회귀기 및 미분불가능 렌더러 양자 모두로 구성된 생성기는 3D 자세 예측을 수행하고 미리 결정된 객체를 예측된 자세로 렌더링한다. 적어도 하나의 실시예에서, 판별기는 이미지들에 나타난 자세들의 시각적 유사성을 점수화한다. 적어도 하나의 실시예에서, 이 점수는 생성기에 대한 훈련 신호로서 사용된다.
- [0015] 적어도 하나의 실시예에서, 로봇 공학 시스템에서의 문제는 6자유도(degrees of freedom)(6-DoF) 자세라 종종 지칭되는 장면 내의 객체들의 3D 위치 및 배향(방위각, 고도 및 사이클로로테이션)을 아는 것이다. 적어도 하나의 실시예에서, 본 명세서에 설명된 다양한 기술들은 객체 배향의 문제, 특히 대칭적인 객체 배향의 문제들을 해결한다.
- [0016] 적어도 하나의 실시예에서, 객체들은 그들의 형상, 텍스처 및 차단들에 따라 다양하고 복잡한 대칭성들을 가질 수 있다. 적어도 하나의 실시예에서, 3가지 상이한 타입의 대칭성이 있다: 점대칭(구), 축대칭(원통), 및 평면 대칭(입방체)- 이들은 임의의 주어진 객체 상에서 혼합되거나 부분적일 수 있음 -. 적어도 하나의 실시예에서, 텍스처가 없는 머그잔의 경우에, y-축 상에 대칭축이 존재하고, 예를 들어, 위를 바라보면, 손잡이가 가시화될 때 그 대칭성이 사라진다. 적어도 하나의 실시예에서, 객체 대칭성들은 대칭성들을 수동으로 특정하고 대응하는 훈련 손실을 제공함으로써 핸들링된다. 적어도 하나의 실시예에서, 객체 대칭성들은 텍스처와 같은 시각적 입력의 특정 양태들을 무시하는 손실 함수들을 통해 핸들링된다.
- [0017] 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 명시적 대칭성 라벨링 또는 손실 사양을 필요로 하지 않고 대칭적인 객체들의 배향을 검출한다. 적어도 하나의 실시예에서, 제안된 방법은 관심 객체의 CAD 3d 모델만을 필요로 한다. 적어도 하나의 실시예에서, 핵심 아이디어는 적대적 컴포넌트가 객체의 대칭성들과 일치하는 손실 함수를 제공하도록 학습하는 적대적 훈련 패러다임을 사용하여 시점 회귀기를 훈련하는 것이다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은, 판별기와 동시에, 생성기의 역할을 하는 3D 렌더러와 결합된 시점 회귀기를 이용한다. 적어도 하나의 실시예에서, 판별기의 목표는 생성기가 그 예측을 개선하도록 도전하게 하는 것이다. 적어도 하나의 실시예에서, 생성기는 관심 객체의 크로핑된 이미지를 입력으로서 취하고 예측된 자세를 반환한다. 적어도 하나의 실시예에서, 판별기는 입력으로서 2개의 이미지, 즉 크로핑된 이미지(생성기를 위해 사용된 동일한 입력) 및 생성기에 의해 출력된 배향에서의 관심 객체의 3D 렌더링을 취한다. 적어도 하나의 실시예에서, 판별기는 그들의 시각적 유사성에 기초하여 이미지 쌍을 점수화한다. 적어도 하나의 실시예에서, 렌더링된 이미지로부터의 경사가 이용 가능하지 않기 때문에, 판별기와 생성기를 공동으로 훈련하는 랜덤화된 샘플링 방법이 사용된다.
- [0018] 적어도 하나의 실시예에서, 기여들은 다음과 같다: 1) 임의의 종류의 객체 대칭성들(대칭성들이 없는 객체들을 포함함)로 시점 추정기를 훈련하는 방법, 2) 경사들이 액세스 가능하지 않을 때 GAN을 훈련하는 샘플링 방법, 및 3) 도전적인 T-LESS 데이터 세트에 대한 긍정적 결과들을 보여주는 실험적 데이터.
- [0019] 적어도 하나의 실시예에서, 시스템은 배향 자세(쿼터니언(quaternion))로 직접 회귀한다. 적어도 하나의 실시예에서, 특수 라벨링과 같은 대칭적인 객체들에 대한 특수한 핸들링이 요구되지 않는다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 기술적으로 연속적이고 자세를 직접 출력한다. 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 현실적 괴리 문제(reality gap problem)로도 알려져 있는, 실제 데이터에 적용하면서 시뮬레이션된 데이터에 대한 훈련을 달성하기 위한 더 큰 노력의 일부이다. 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 실제 이미지들에 대해 테스트되면서, 렌더링된 이미지들에 대해 훈련되며, 관심 객체의 3D CAD 모델만을 필요로 하며, 이러한 3D CAD 모델은 종종 수작업 라벨링 실세계 훈련 데이터보다 획득하기가 더 쉽다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 도메인 랜덤화를 이용하여 해당 괴리를 가교한다. 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 극단적인 시각적 다양성을 갖는 모델을 훈련하며, 따라서 실세계 이미지 모델과 함께 제공될 때, 이는 모델을 다른 시각적 변형으로서 취급한다.
- [0020] 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 생성 적대적 네트워크(generative adversarial network)("GAN")의 인기도에 의해 대중화된 비지도 적대적 훈련을 사용한다. 적어도 하나의 실시예에서, 경사 손실이 판별기로부터 생성기로 흐를 수 없다. 적어도 하나의 실시예에서, 이 문제는 강화 학습을 활용함으로써, 예를 들어, 보상으로 판별기 출력을 최대화함으로써 해결된다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법은 이미지가 주어진 경우 유효한 배향 자세의 분포를 검색하기 위해 GAN을 사용하는 것은 물론, 판별기로부터 신호를 획득하기 위해 랜덤화된 샘플링 방법을 활용한다.
- [0021] 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 임의의 수동 대칭성 라벨링 없이 객체들의 배향(시점 추정이라고도 알려짐)을 예측하도록 학습한다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들에서 자세

및 배향은 교환 가능하게 사용될 수 있다. 적어도 하나의 실시예에서, 생성기는 객체의 크로핑된 이미지를 처리하고 쿼터니언 자세를 예측한다. 적어도 하나의 실시예에서, 판별기는 입력으로서 2개의 이미지, 즉 크로핑된 이미지 및 생성기의 출력 자세의 렌더를 취하고, 이미지의 자세들이 일치할 가능성이 얼마나 큰지를 출력한다.

[0022] 적어도 하나의 실시예에서, 대칭적인 객체의 자세를 추정하는 것은 사소하지 않은 문제이다. 적어도 하나의 실시예에서, 도 2는 T-LESS의 일부 객체에 대한 2개의 상이한 실측 정보 자세(행 방향)를 도시한다. 적어도 하나의 실시예에서, 행 a)에서, 2개의 자세는 외관 및 형상이 거의 완벽하게 동일한 것처럼 보이고, 이 자세들이 대칭으로 라벨링되어야 한다는 것이 명백하다. 적어도 하나의 실시예에서, 행 b)에서, 2개의 자세는 시각적으로 명확히 다르지만, ADI 및 VSD(Visible Surface Discrepancy)와 같은 메트릭들이고, 이들에 기초한 손실 함수들은 형상이 거의 동일하기 때문에 이들 자세를 대칭으로 취급한다. 적어도 하나의 실시예에서, 행 c)에서, 자세들은 거의 동일하게 보이지만, 인간 라벨러는 제1 이미지(좌측)의 하단 우측 코너의 시각적 피치가 제2 이미지(우측)에서 상단 우측 코너로 이동된 것을 볼 수 있다. 적어도 하나의 실시예에서, 이는 대칭성이 어떻게 라벨링되어야 하는지에 대한 모호성을 초래한다. 적어도 하나의 실시예에서, 무엇을 라벨링할지에 관한 결정을 내려야 하거나 3D 모델들의 형상에 의존하는 대신에, 본 명세서에 설명된 기법들은 시각적 객체 대칭성들에 관해 적응적으로 학습한다. 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 도 2에 예시된 것들과 같은 모호한 '의사 대칭성들'을 핸들링할 수 있다.

[0023] 도 2는 적어도 하나의 실시예에 따른, 다양한 대칭도들을 갖는 다양한 객체들의 예를 예시한다. 적어도 하나의 실시예에서, 도 2는 T-LESS 데이터 세트 내의 '대칭' 객체들의 예를 예시한다. 적어도 하나의 실시예에서, 제1 객체(202)와 제2 객체(204)는 대부분 동일하다. 적어도 하나의 실시예에서, 제3 객체(206) 및 제4 객체(208)는 상이한 형상들을 갖지만 텍스처들이 일치된다. 적어도 하나의 실시예에서, 제5 객체(210) 및 제6 객체(212)는 텍스처에서 약간의 차이를 갖는다. 적어도 하나의 실시예에서, 대부분 동일한 (a)로부터 형상은 동일하나 텍스처는 그렇지 않은 (b), 매우 작은 텍스처 차이들 (c)까지의 대칭성들이 나타나 있다. 적어도 하나의 실시예에서, 이는 데이터를 라벨링하거나 자세 추정기를 훈련하려고 시도할 때 모호성들을 초래할 수 있다.

[0024] 조건부 생성 적대적 네트워크

[0025] 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 타겟 이미지 x_t 를 조건으로 하여 타겟 자세 p_d 를 예측할 수 있는 생성 모델을 구성한다. 이러한 타겟을 예측하기 위해, 자세 및 CAD 모델을 수용하는 블랙 박스 렌더($p_d|x_t$)가 사용된다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 입력 자세에서 모델을 렌더링한다. 적어도 하나의 실시예에서, 태스크는 $(p_d|x_t)$, 즉, $p_d \approx \mathcal{R}(p_g|x_t)$ 를 학습하는 것과 동등하다. 적어도 하나의 실시예에서, 생성 프로세스가 도 4에 도시된다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 적대적 네트워크를 이용함으로써 이 프로세스를 최적화한다. 적어도 하나의 실시예에서, 이 네트워크에서, 생성기 G 는 D 로 표시된 판별기가 최대로 혼동하게 하는 것을 목표로 하고, 판별기는 $(p_g|x_t)$ 로 표시된 생성기에 의해 생성된 샘플과 $p \sim p_d|x_t$ 사이를 구별하도록 훈련되어 있다. 적어도 하나의 실시예에서, 결과로서 타겟들은 점진적으로 점점 더 유사해지고, 예를 들어, $p_d \approx p_g$ 이다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 다른 관련 작업들에 비교하여 p_d 및 p_g 로부터의 정렬된 예들을 요구하지 않는다. 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 G 의 출력에 의해 생성된 이미지들을 '가짜'로, p_d 로부터의 이미지들을 '실제'로 지칭한다.

[0026] 훈련 절차

[0027] 적어도 하나의 실시예에서, 원래의 GAN에 의해 도입된 목적은 최적화하기 어렵고, 따라서, 제로-중심 경사 패널티와 결합되는, L1 손실 및 메트릭 학습 정보 이론을 사용한다.

[0028] 도 3은 적어도 하나의 실시예에 따라, 판별기를 훈련하는 프로세스의 예를 예시한다. 적어도 하나의 실시예에서, 훈련은 가짜 이미지들 및 실제 이미지들로 달성된다. 적어도 하나의 실시예에서, 가짜 이미지(302)가 생성기(304)에 제공된다. 적어도 하나의 실시예에서, 가짜 이미지(302)와 함께 판별기(306)에 제출되는 이

미지가 생성기(304)에 의해 생성된다. 적어도 하나의 실시예에서, 실제 이미지(308)는 교란된 이미지(310)를 생성하도록 교란된다. 적어도 하나의 실시예에서, 실제 이미지(308) 및 교란된 이미지(310)가 판별기(306)에 제공된다. 적어도 하나의 실시예에서, 판별기 손실은 다음과 같이 정의된다:

$$\mathcal{L}_D = \mathbb{E}_{p \sim p_d | x_t} [D(R(p), x_t) - a] +$$

$$\mathbb{E}_{p \sim p_d | x_t} [D(R(G(R(p))), x_t) - b] + Z \quad (1)$$

여기서, Z는 제로-중심 경사 패널티이고, a는 실제 데이터 타겟 라벨이고 b는 가짜 데이터 타겟 라벨이다. 적어도 하나의 실시예에서, 도 3은 판별기를 훈련하기 위한 개요를 나타낸다. 적어도 하나의 실시예에서, 생성기는 '가짜' 이미지들을 출력하는데 사용된다. 적어도 하나의 실시예에서, 렌더링된 샘플 자세에 대한 작은 섭동이 이미지들의 '실제' 세트로서 사용된다. 적어도 하나의 실시예에서, '실제' 이미지들은 서로 매우 가까운 자세들을 나타내어 낮은 타겟 라벨을 야기한다. 적어도 하나의 실시예에서, 생성기가 자세들을 예측함에 있어서 불량하지만, 매우 다른 자세들을 보여야 하는 '가짜' 이미지들을 훈련할 때, 타겟에는 하이 값의 라벨이 할당된다.

생성기: 적어도 하나의 실시예에서, 다음의 손실이 생성기에 대해 최대화된다:

$$\mathcal{L}_G = \mathbb{E}_{p \sim p_d | x_t} [D(R(G(R(p))), x_t) - c] \quad (2)$$

여기서 c 는 생성기 타겟이다. 적어도 하나의 실시예에서, R 로부터 경사가 취해질 수 없기 때문에, 훈련을 위해 샘플링 방법이 도입된다. 적어도 하나의 실시예에서, K개의 노이즈 벡터가 소정의 분포, 예를 들어 정규 또는 균일 분포로부터 샘플링되고, 이는 $p^1, p^2, \dots, p^K$ 으로 표시된다. 적어도 하나의 실시예에서, 이러한 노이즈 벡터들은 다음과 같이 예측된 자세를 교란하고 생성기 손실을 업데이트하는 데 사용된다:

$$\mathcal{L}_G = \mathbb{E}_{p \sim p_d | x_t} [G(R(p)) - \argmin_{k=1 \dots K} [D(R(G(R(p)) + p_k), x_t)]] \quad (3)$$

도 4는 적어도 하나의 실시예에 따라, 생성 네트워크를 훈련하는 프로세스의 예를 예시한다. 적어도 하나의 실시예에서, 도 4는 1) 이미지(402)로부터 자세를 생성하고(404), 2) 예측된 자세(406) 주위에서 자세들을 샘플링하는(408) 훈련 절차의 개요를 도시한다. 적어도 하나의 실시예에서, 이러한 자세들은 렌더링되고, 원래의 이미지와 쌍을 이루며, 판별기로부터 점수화된다. 적어도 하나의 실시예에서, 이들 점수들로부터, 알고리즘은 원본과 가장 시각적으로 유사한 자세를 식별한다. 적어도 하나의 실시예에서, 이는 경사 신호를 검색하기 위한 유일한 가능한 공식이 아니다. 적어도 하나의 실시예에서, k 포인트들을 샘플링함으로써 경사를 직접 추측하는 것을 실험하는 것이 가능하며, 여기서 k는 자유도의 수 - 1과 동일하다. 적어도 하나의 실시예에서, 표준 강화 학습 방법들을 사용하여 예상된 반환을 최대화하는 것으로서 이 문제를 표현하는 것이 또한 가능하다.

훈련 상세사항

적어도 하나의 실시예에서, 적대적 훈련이 제시되었지만, 실제로 훈련 GAN들은 불안정하고 도전적일 수 있다. 적어도 하나의 실시예에서, 더 강건한 훈련 절차가 후술되는 바와 같이 구현된다.

판별기

적어도 하나의 실시예에서, 수학적 1에서, '실제' 쌍들에 대한 타겟 값들 a 은 이미지들의 자세들 사이의 거리에 기초하며, 판별기가 메트릭 함수로서 작용하게 한다. 적어도 하나의 실시예에서, 더 큰 섭동(perturbation)들은 더 큰 타겟 값들을 초래한다. 적어도 하나의 실시예에서, '가짜' 쌍들에 대해, 타겟 값들 b 은 랜덤하게 샘플링되는데, 그 이유는 시스템이 알려지지 않은 대칭성들로 인해 자세들이 얼마나 상이한지를 확신할 수 없기 때문이다. 적어도 하나의 실시예에서, 실제 이미지들의 쌍에 대해, a 는 섭동에 기초하여 0과 0.3 사이에서 스케일링되며, b 는 0.8과 1 사이의 무작위 균일한 분포로부터 샘플링된다. 적어도 하나의 실시예에서, 훈련 동안 모드

붕괴를 회피하는 것을 돕기 위해, 생성기의 예측 시간의 20%는 가짜 쌍들을 구성할 때 랜덤 자세로 대체된다. 적어도 하나의 실시예에서, 이는 판별기가 특히 훈련의 초기에 좁은 범위를 가질 수 있는 생성기의 출력에 과적합(over-fit)되지 않는 것을 보장한다.

[0042] **도메인 랜덤화**

[0043] 적어도 하나의 실시예에서, 모델이 합성 전용 훈련 데이터로부터 실제 테스트 이미지들로 일반화될 수 있게 하기 위해 훈련 이미지들에 많은 데이터 증강들이 적용된다. 적어도 하나의 실시예에서, 본 명세서에 설명된 기법들은 (1) 합성 이미지들을 랜덤 광 위치, 강도, 및 정반사로 렌더링하고, (2) 파스칼 VOC 데이터 세트로부터 랜덤 배경들을 삽입하고, (3) 랜덤 마스크들을 적용하여 차단을 시물레이션하고, (4) 최종 이미지를 랜덤 대비, 밝기, 채도 및 색조 시프트로 증강한다.

[0044] **2D 객체들에 대한 실험들**

[0045] 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 간단한 2D 형상들에 적용되며, 기법들은 고정 좌표 프레임에 대한 이미지 평면 내의 형상의 회전을 나타내는 1D 자세를 예측하는 것을 목표로 한다. 적어도 하나의 실시예에서, 이는 SymGAN이 훈련 동안 어떻게 학습하는지 및 그의 최종 출력이 무엇처럼 보이는지에 대한 더 나은 이해를 가능하게 한다.

[0046] **데이터 세트 및 훈련**

[0047] 적어도 하나의 실시예에서, 4개의 형상들이 조사된다: 정사각형, 컬러 코너들을 갖는 정사각형, 삼각형, 및 10각형. 적어도 하나의 실시예에서, 각각의 객체는 각각 2, 1, 3, 및 5개의 대칭축을 갖는다. 적어도 하나의 실시예에서, 동일한 자세가 4개의 값, 예를 들어, 0° , 90° , 180° , 270° 로 표현될 수 있고, 이는 정사각형에 대해 동일한 시각적 평면내 회전을 제공한다. 적어도 하나의 실시예에서, 형상들은 도 7, 9, 11 및 13에 도시된다. 적어도 하나의 실시예에서, 32×32 이미지들은 다음과 같이 생성된다: 1) 랜덤 스케일(0.5 내지 1)을 고르는 것, 2) 객체를 랜덤 위치에 배치하는 것, 3) 랜덤 회전을 적용하는 것, 4) 그것을 흑색 배경에 붙이는 것, 및 5) 랜덤 블러를 추가하는 것.

[0048] 적어도 하나의 실시예에서, 생성기 및 판별기 양자 모두는 동일한 아키텍처를 사용한다. 적어도 하나의 실시예에서, 이는 2개의 풀링 계층을 갖는 4개의 컨볼루션 계층, 및 그에 후속하는 1개의 완전 연결 계층으로 구성된다. 적어도 하나의 실시예에서, 아담 최적화기(Adam optimizer)가 사용되고 64의 배치 크기(batch size)로 10,000 반복에 대해 훈련된다. 적어도 하나의 실시예에서, 동일한 아키텍처를 사용하여, 자세로 직접 회귀하는 자세 추정기가 훈련된다. 적어도 하나의 실시예에서, L2가 손실로서 사용된다.

[0049] **결과**

[0050] 적어도 하나의 실시예에서, 자세 예측기를 평가하기 위해, 대칭성을 고려하면서 실측 정보의 1도 이내에 있다면 예측은 성공이다. 적어도 하나의 실시예에서, 간단한 정사각형으로 베이스라인을 훈련할 때, 여기에 설명된 기법들은 10% 유효 예측을 획득하는 한편 제안된 방법은 98%를 달성한다. 적어도 하나의 실시예에서, 방법을 이용하여, 텍스처링된 정사각형, 삼각형, 및 10각형에 대해 각각 96%, 96%, 및 82% 커버리지가 제공된다. 적어도 하나의 실시예에서, L2를 이용하여 멀티모드 함수로 직접 회귀하는 것은 맞춤형 솔루션보다 나쁘게 기능할 것이다. 적어도 하나의 실시예에서, 낮은 차원수의 평면내 회전 문제가 주어지면, 판별기 및 생성기 양자 모두의 거동이 조사된다.

[0051] **판별기**

[0052] 적어도 하나의 실시예에서, 도 5 및 도 6은 하나의 이미지를 고정하고- 좌측은 20° 로 설정되고 우측은 70° 로 설정됨 -, 0° 에서 360° (x-축)까지 제2 이미지 회전을 변화시킬 때의 판별기 판정 경계를 도시한다. 적어도 하나의 실시예에서, 라인은 입력 이미지를 상이한 자세와 비교할 때의 판별기의 출력을 나타내고; 로우 값은 유사한 자세를 나타내고 하이 값은 상이한 자세를 나타낸다. 적어도 하나의 실시예에서, 이는 판별기가 상이한 대칭성들 사이를 구별하도록 시각적으로 학습되었음을 제시한다. 적어도 하나의 실시예에서, 예를 들어, 도 5에서, 입력에 대한 모든 대칭 자세들 $\{20^\circ, 110^\circ, 200^\circ, 290^\circ\}$ 에는 낮은 점수들이 할당되고, 한편 다른 곳에서 다른 비대칭 자세들에는 높은 점수들이 할당된다. 적어도 하나의 실시예에서, 이러한 이미지들에 대한 생성기의 출력은 훈련의 중간 스테이지들에서 수직 점선으로 나타나 있다.

[0053] 도 5는 적어도 하나의 실시예에 따른, 판별기 출력(502) 대 20도 자세의 객체의 이미지에 대한 자세의 예를 예시한다. 적어도 하나의 실시예에서, 도 5는 단색 백색 2d 정사각형에 대한 판별기의 출력을 예시한다. 적어도

하나의 실시예에서, 훈련을 통한 중간 생성기의 출력이 라인(504)으로서 도시된다. 도 6은 적어도 하나의 실시예에 따른, 판별기 출력(602) 대 70도 자세의 객체의 이미지에 대한 자세의 예를 예시한다. 적어도 하나의 실시예에서, 도 5는 단색 백색 2d 정사각형에 대한 판별기의 출력을 예시한다. 적어도 하나의 실시예에서, 훈련을 통한 중간 생성기의 출력이 라인(604)으로서 도시된다.

[0054] 적어도 하나의 실시예에서, 이전에 설명한 바와 같이, 이 입력 이미지에 대해 훈련할 때, 본 명세서에 설명되는 기법들은 예측된 자세 주위를 중심으로 k개의 랜덤 자세로 객체를 렌더링한다. 적어도 하나의 실시예에서, 이들은 입력 이미지와 쌍을 이루고, 판별기로부터 차이 점수들이 얻어진다. 적어도 하나의 실시예에서, 판별기로부터 최상의(가장 낮은) 차이 점수를 갖는 자세가 생성기에 대한 훈련 타겟으로서 선택된다.

[0055] 적어도 하나의 실시예에서, 도 5 및 도 6에 도시된 바와 같이, 최상의 점수를 갖는 자세는 생성기의 출력(180)보다 큰 무언가일 것이다. 적어도 하나의 실시예에서, 많은 훈련 반복들을 통해, 생성기의 출력은 200에서 계곡을 향해 느리게 푸시될 것이고, 여기서 이는 결국 수렴할 것이다.

[0056] 도 7은 적어도 하나의 실시예에 따른, 생성기에 의해 생성된 이미지의 일 예를 예시한다. 적어도 하나의 실시예에서, 도 7은 4개의 상이한 배향에서의 정사각형을 예시한다: 제1 배향(702), 제2 배향(704), 제3 배향(706), 및 제4 배향(708). 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 도 7에 도시된 객체들의 대칭성 및 가능한 배향들 양자 모두를 식별하는 데 사용될 수 있다.

[0057] 도 8은 적어도 하나의 실시예에 따른, 도 7에 도시된 객체에 대해 생성된 생성기 및 판별기 출력의 예를 예시한다. 적어도 하나의 실시예에서, 생성기(802) 및 판별기(804)의 출력은 다각형으로 표현된 4-축 대칭을 예시한다.

[0058] 도 9는 적어도 하나의 실시예에 따른, 생성기에 의해 생성된 이미지의 일 예를 예시한다. 적어도 하나의 실시예에서, 도 9는 4개의 상이한 배향의 패턴화된 정사각형을 예시한다: 제1 배향(902), 제2 배향(904), 제3 배향(906), 및 제4 배향(908). 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 도 9에 도시된 객체들의 대칭성 및 가능한 배향들 양자 모두를 식별하는 데 사용될 수 있다.

[0059] 도 10은 적어도 하나의 실시예에 따른, 도 9에 도시된 객체에 대해 생성된 생성기 및 판별기 출력의 예를 예시한다. 적어도 하나의 실시예에서, 생성기(1002) 및 판별기(1004)의 출력은 다각형으로 표현된 2-축 대칭을 예시한다.

[0060] 도 11은 적어도 하나의 실시예에 따른, 생성기에 의해 생성된 이미지의 일 예를 예시한다. 적어도 하나의 실시예에서, 도 11은 4개의 상이한 배향의 삼각형을 예시한다: 제1 배향(1102), 제2 배향(1104), 제3 배향(1106), 및 제4 배향(1108). 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 도 11에 도시된 객체들의 대칭성 및 가능한 배향들 양자 모두를 식별하는 데 사용될 수 있다.

[0061] 도 12는 적어도 하나의 실시예에 따른, 도 11에 도시된 객체에 대해 생성된 생성기 및 판별기 출력의 예를 예시한다. 적어도 하나의 실시예에서, 생성기(1202) 및 판별기(1204)의 출력은 다각형으로 표현된 3-축 대칭을 예시한다.

[0062] 도 13은 적어도 하나의 실시예에 따른, 생성기에 의해 생성된 이미지의 일 예를 예시한다. 적어도 하나의 실시예에서, 도 13은 4개의 상이한 배향에서의 다각형을 예시한다: 제1 배향(1302), 제2 배향(1304), 제3 배향(1306), 및 제4 배향(1308). 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 도 13에 도시된 객체들의 대칭성 및 가능한 배향들 양자 모두를 식별하는 데 사용될 수 있다.

[0063] 도 14는 적어도 하나의 실시예에 따른, 도 13에 도시된 객체에 대해 생성된 생성기 및 판별기 출력의 예를 예시한다. 적어도 하나의 실시예에서, 생성기(1402) 및 판별기(1404)의 출력은 다각형으로 표현된 10-축 대칭을 예시한다.

[0064] 생성기

[0065] 적어도 하나의 실시예에서, 도 8, 10, 12 및 14의 좌측 그래프는 훈련이 수렴된 후의 각각의 실측 정보 자세(x-축)에서의 생성기의 출력(y-축)을 나타낸다. 적어도 하나의 실시예에서, 플롯은 상기 플롯에 걸쳐 수평선을 그림으로써 해석된다. 적어도 하나의 실시예에서, 예를 들어 도 8에 대한 라인 $y = 200$ 에 대해, 이는 $\{20^\circ, 110^\circ, 200^\circ, 290^\circ\}$ 에서 모델 출력과 교차한다. 적어도 하나의 실시예에서, 이는, 도 7의 이러한 4개의 실측 정보 자세에 대해, 모든 4개의 자세가 서로 구별가능하지 않기 때문에, 모델이 원하는 거동인 200을 출력한다는 것을 의미한다. 적어도 하나의 실시예에서, 대칭 값들의 임의의 다른 것에 비해 200을 선택하는 것은 임의적이

고 이는 모델 초기화로부터의 결과이다.

[0066] 적어도 하나의 실시예에서, 도 8, 10, 12 및 14의 우측 그래프는 훈련이 수렴된 후의 모든 가능한 자세 쌍들에 대한 히트 맵으로서 판별기의 출력을 설명한다. 적어도 하나의 실시예에서, 도 8의 우측 그래프에서, 행, 포인 트($x = 30, y=50$)는 제1 이미지가 30° 의 정사각형 및 50° 의 제2 이미지를 갖는 이미지들의 쌍에 대한 판별기의 출력에 대응한다. 적어도 하나의 실시예에서, 이 플롯을 해석하기 위해, 다시, 그래프 상의 어디에서나 수평(또는 수직) 라인을 추적하는 것을 상상한다. 적어도 하나의 실시예에서, 도 8의 우측에서, 트레이스 $y = 200$ 이면, 예를 들어, 정사각형 상의 4개의 대칭성에 대응하는 {20, 110, 200, 290}에서 판별기의 점수가 낮은 4개의 벨리가 있을 것이다. 적어도 하나의 실시예에서, 다른 곳에서는 판별기의 점수가 높는데, 그 이유는 그것이 시각적으로 상이한 자세들에 높은 값들을 할당하도록 학습했기 때문이다.

[0067] 3D에서의 실험들: T-LESS

[0068] 적어도 하나의 실시예에서, 구현은 SixD Pose Challenge로부터의 T-LESS 데이터 세트에 대한 3D 배향 예측 문제에 대해 테스트된다. 적어도 하나의 실시예에서, T-LESS는 30개의 객체 인스턴스로 구성되며, 이들 중 대부분은 제한된 텍스처 및 소정 레벨의 대칭성을 갖는다. 적어도 하나의 실시예에서, 실제 객체들의 임의의 세트에 대한 하나의 도전은 대칭성들의 수동 라벨링을 훨씬 더 어렵고 모호하게 할 수 있는 다양한 대칭성들의 레벨이다.

[0069] 훈련 절차

[0070] 적어도 하나의 실시예에서, 훈련 구현은 3D 객체들에 대해 더 실용적하도록 수정된다. 적어도 하나의 실시예에서, 각각의 객체는 이용 가능한 재구성된 3D 모델을 가지며, 이로부터 훈련 이미지들이 렌더링된다. 적어도 하나의 실시예에서, 시스템 동작 동안 훈련 이미지들을 생성하는 렌더러가 훈련 루프에 존재한다. 적어도 하나의 실시예에서, 이는 여전히 3D 설정에서도 가능하지만, 훈련 이미지들을 사전 렌더링하는 것은 훈련 속도를 개선하였고 성능을 크게 감소시키지 않았다. 적어도 하나의 실시예에서, 객체당 10,000개의 이미지가 렌더링되고, 각각은 흑색 배경 상의 무작위 자세를 가지며, 도메인 랜덤화 접근법에 따라 조명 위치 및 강도를 랜덤하게 변경한다.

[0071] 적어도 하나의 실시예에서, 훈련 동안 렌더러를 시뮬레이션하기 위해, 최근접 이웃 그래프가 각각의 훈련 이미지의 실측 정보 자세에 기초하여 구성된다. 적어도 하나의 실시예에서, 자세 p 를 갖는 이미지를 생성하기 위해, p 의 최근접 이웃인 훈련 이미지에 대한 쿼리가 수행된다. 적어도 하나의 실시예에서, 소정의 자세 p 주위의 K 이미지들을 샘플링하기 위해, N - 여기서, $N > K$ -, p 의 최근접 이웃들 및 이들의 샘플 K 에 대한 쿼리가 수행된다. 적어도 하나의 실시예에서, 예를 들어, '실제' 이미지들의 쌍으로 판별기를 훈련할 때, 무작위 자세 p 를 선택하고, p 의 최근접 자세를 갖는 훈련 이미지를 찾는다. 적어도 하나의 실시예에서, $N=10$ 에서 p 의 최근접 훈련 이미지들을 쿼리하고, 그들 중 하나를 랜덤하게 선택한다. 적어도 하나의 실시예에서, 이는 객체가 유사하지만 정확하지 않은 자세를 갖는 한 쌍의 이미지를 생성한다. 적어도 하나의 실시예에서, 대조적으로, 한 쌍의 '가짜' 이미지들에 대해, 가능한 자세 공간을 탐색하기 위해 생성기의 출력으로부터 충분히 멀리 떨어져 샘플링하는 것이 중요하지만, 그렇게 멀리 떨어져 샘플링된 자세들이 단지 완전히 랜덤한 것은 아니다. 적어도 하나의 실시예에서, 따라서, 생성기로부터 출력되는 자세로서, p 는 $N = 200$ 개의 p 의 최근접 이웃들로부터 랜덤하게 $K=10$ 개의 이미지를 샘플링한다. 적어도 하나의 실시예에서, 최근접 이웃 그래프를 사용하여 렌더링된 이미지들을 인출한 이후에, 다른 데이터 증강들이 훈련 루프에서 적용된다.

[0072] 적어도 하나의 실시예에서, 이 프로세스는 판별기가 '실제' 쌍으로 보는 것을 제한하지 않는다는 점에 유의한다. (자세 공간에서 멀리 떨어진) 상이한 2개의 자세인 (p_1, p_2) 를 고려한다. (p_1, p_2) 이 시각적으로 모호한 경우, $(p_1 + \epsilon, p_2)$ 및 $(p_1 + f, p_2)$ 는 또한 충분히 작은 일부 ϵ 에 대해 시각적으로 모호하다. 따라서, $(p_1, p_1 + \epsilon)$ 는 (p_1, p_2) 와 유사한 시각적 정보를 제공한다. $(p_1, p_1 + \epsilon)$ 는 자세 공간에서 가깝고 훈련에서 '실제'

쌍으로서 이용될 것이다. (p_1, p_2) 가 $(p_1, p_1 + \epsilon)$ 와 유사하지 않은 경우, 이때, (p_1, p_2) 는 시각적으로 모호하지 않으며, 훈련 동안 그것을 '실제' 쌍으로서 포함하지 않는 것이 옳바르다.

[0073] **모델 상세사항**

[0074] 적어도 하나의 실시예에서, 생성기는 객체의 128x128 크롭을 입력으로서 취한다. 적어도 하나의 실시예에서, 이는 먼저 FLOWNetSimple 아키텍처에 기초한 피쳐 추출기를 통해 이미지를 전달하고, 이어서 객체의 자세를 쿼터니언(quaternion)으로서 예측하기 위해 3개의 완전히 연결된 계층들이 뒤따른다. 적어도 하나의 실시예에서, 판별기는 또한 동일한 아키텍처를 사용하지만, 단일 점수만을 출력한다. 적어도 하나의 실시예에서, 양자 모두는 $1e^{-4}$ 의 학습률 및 32의 배치 크기를 갖는 아담을 사용하여 훈련된다. 적어도 하나의 실시예에서, 일반적으로 50,000 내지 100,000 업데이트 반복들 내에서, 관심 객체에 대한 훈련 데이터에 성능 수렴이 존재할 때 훈련이 중단된다. 적어도 하나의 실시예에서, 객체마다 3개의 모델이 훈련되고, 테스트 데이터에 대해 평가하기 위해 훈련 데이터에 대해 최상으로 기능하는 것을 유지한다.

[0075] **평가**

[0076] 적어도 하나의 실시예에서, T-LESS에 대한 방법을 평가하기 위해, VSD(Visible Surface Discrepancy)가 사용되는데, 그 이유는 그것이 추정된 및 실측 정보 가시 객체 깊이 표면들 사이의 거리에 의해 결정되는 모호성-불변 자세 에러 함수이기 때문이다. 적어도 하나의 실시예에서, 방법은 객체 자세 및 증강 자동 인코더("AAE")로 직접 회귀하도록 훈련된 베이스라인과 비교된다. 적어도 하나의 실시예에서, 전자는 SymGAN 내의 자세 예측기와 동일한 아키텍처를 사용하지만, 이는 T-LESS로부터의 주석화된 실제 자세들을 사용하여 훈련되고 통합된 판별기 또는 렌더러를 갖지 않는다. 적어도 하나의 실시예에서, 후자는 자동 인코더 방법을 사용하여 3D 배향 검색을 위해 훈련된 방법이다.

[0077] 적어도 하나의 실시예에서, 초점은 6D 자세 챌린지의 3D 배향 부분에 있고, 그래서, 평가 동안 3D 배향 예측이 실측 정보 3D 변환과 조합된다. 적어도 하나의 실시예에서, 네트워크에 대한 입력으로서, 실측 정보 경계 박스들이 사용되고, 하나의 객체의 다수의 인스턴스가 하나의 장면에 존재할 때, 가장 높은 가시성을 갖는 인스턴스가 선택된다. 적어도 하나의 실시예에서, ICP와 같은 방법들을 사용하여 정제된 자세들을 보여주는 것은 시스템의 RGB 컴포넌트에 집중하기 위해 생략된다.

[0078] **결과**

[0079] 적어도 하나의 실시예에서, 도 20 내지 도 28은 T-LESS 장면 1로부터 랜덤하게 선택된 정성적 결과를 도시하고; 이는 4개의 객체: 02, 25, 29 및 30을 도시한다. 적어도 하나의 실시예에서, SixD 챌린지에서와 같이, $err_{vsd} < 0.3$ 에서의 정확한 6D 객체 자세들의 리콜이 20 mm로 설정된 허용오차 및 10%를 초과하는 가시성을 갖는 객체들로 보고된다. 적어도 하나의 실시예에서, 방법을 유사한 RGB 기반 방법과 비교할 때, 평균 50.74를 갖는 상기 방법은 평균 36.79를 갖는 이전 작업보다 더 나은 배향 검색을 학습한다. 적어도 하나의 실시예에서, 평균 45.74를 갖는 베이스라인 또한 더 강하다. 적어도 하나의 실시예에서, 베이스라인 회귀기는 테스트 데이터와 동일한 환경으로부터 도입되는 훈련 데이터를 오용할 수 있다.

[0080] 표 1 T-LESS Dataset T-LESS에 대한 결과: 모든 프라임센스(Primesense) 테스트 장면들에서 $err_{vsd} < 0.3$ 에 대한 객체 리콜. 이탤릭체 숫자는 대칭축을 갖는 객체를 나타낸다.

객체	베이스 라인	선더메이어	SymGAN (본 출원)
1	65.0	12.33	75.5
2	80.2	11.23	88.0
3	85.2	13.11	84.0
4	62.2	12.71	66.2
5	58.5	66.70	73.0
6	65.5	52.30	65.1
7	11.8	36.58	22.2
8	4.6	22.05	42.0
9	20.6	46.49	47.4
10	3.6	14.31	13.3
11	66.3	15.01	66.0
12	40.3	31.34	49.4
13	69.0	13.60	66.6
14	36.0	45.32	34.3
15	54.0	50.00	58.2
16	78.0	36.09	80.6
17	68.0	81.11	54.3
18	67.0	52.62	63.0
19	23.5	50.75	28.5
20	7.8	37.75	12.4
21	35.0	50.89	33.3
22	11.0	47.60	9.6
23	15.0	35.18	17.0
24.0	67.75	11.24	68.2
25	60.5	37.12	35.0
26	49.0	28.33	43.3
27	10.0	21.86	40.0
28	43.4	42.58	54.2
29	23.0	57.01	38.5
30.0	90.3	70.42	92.0
평균	45.74	36.79	50.74

[0081]

[0082]

적어도 하나의 실시예에서, 표 1은, 도 20 내지 도 28의 객체(02)와 같은, 대칭축을 갖는 객체를 이탤릭체를 사용하여 식별한다. 적어도 하나의 실시예에서, 이들을 격리시키고 그들의 *vsd* 를 계산하면 67.60 평균 및 15.8 표준편차를 생성하고, 이는 35.94 평균 및 18.6 표준편차를 갖는, 평면 대칭성을 갖는 객체들(비이탤릭 객체들)과 비교하여 더 나은 결과이다.

[0083]

적어도 하나의 실시예에서: 1) 대칭축들을 갖는 객체들은 판별기로 훈련될 때 그것이 차원들의 수를 감소시킴에 따라 학습하기가 더 쉽고; 및/또는 2) 판별기는 대칭 평면들을 갖는 객체들에 대해 더 많은 모드 붕괴를 겪는다. 적어도 하나의 실시예에서, SynGAN은, RGB 이미지로부터 직접, 객체의 시각적 외관에 기초하여 학습하므로, 전적으로 형상에만 의존하는 문제가 없다.

[0084]

대칭 뷰에 대한 결과

[0085]

적어도 하나의 실시예에서, 특정 대칭축에 대한 객체에 대한 SymGAN 성능이 검사된다. 적어도 하나의 실시예에서, T-LESS 데이터 세트로부터 4개의 객체가 선택되고, 각각의 객체에 대해 뷰들의 세트는 뷰들이 객체의 하향식 뷰 주위의 360도 회전을 포괄하도록 테스트 세트로부터 선택된다. 적어도 하나의 실시예에서, 이는 도 15 내지 도 18에서 시각화되고; 각각의 플롯의 중심에는 객체의 이미지가 있고, 선택된 테스트 뷰들은 페이지로부터 나오는 축 주위로 그 이미지를 회전시키는 것에 대략 대응한다. 적어도 하나의 실시예에서, T-LESS의 카메라는 객체들 바로 위에 있지 않지만, 그러나, 그에 가깝다.

[0086]

도 15는 적어도 하나의 실시예에 따른, 상이한 모델들이 3-축 대칭을 갖는 객체의 회전에 어떻게 반응하는지의 예를 예시한다. 적어도 하나의 실시예에서, 객체(1502)가 회전되고, 본 명세서에 설명된 기법들(SymGAN) 및 베이스라인을 사용하여 예측된 자세와 함께 실측 정보(GT POSE)가 플롯팅된다. 적어도 하나의 실시예에서, 객체(1502)는 대략 3-축 대칭을 나타낸다.

[0087]

도 16은 적어도 하나의 실시예에 따른, 상이한 모델들이 반대칭 객체의 회전에 어떻게 반응하는지의 예를 예시한다. 적어도 하나의 실시예에서, 객체(1602)는 회전되고, 실측 정보(GT POSE)는 본 명세서에 설명된 기법들

(SYMGAN) 및 베이스라인을 사용하여 예측된 자세와 함께 플롯팅된다. 적어도 하나의 실시예에서, 객체(1602)는 대략 2-축 대칭을 나타낸다.

[0088] 도 17은 적어도 하나의 실시예에 따른, 상이한 모델들이 반대칭 객체들의 그룹의 회전에 어떻게 반응하는지의 예를 예시한다. 적어도 하나의 실시예에서, 객체(1702)가 회전되고, 본 명세서에 설명된 기법들(SYMGAN) 및 베이스라인을 사용하여 예측된 자세와 함께 실측 정보(GT POSE)가 플롯팅된다. 적어도 하나의 실시예에서, 객체(1702)는 대략 7-축 대칭을 나타낸다.

[0089] 도 18은 적어도 하나의 실시예에 따른, 상이한 모델들이 2-축 대칭을 갖는 객체의 회전에 어떻게 반응하는지의 예를 예시한다. 적어도 하나의 실시예에서, 객체(1802)가 회전되고, 본 명세서에 설명된 기법들(SYMGAN) 및 베이스라인을 사용하여 예측된 자세와 함께 실측 정보(GT POSE)가 플롯팅된다. 적어도 하나의 실시예에서, 객체(1802)는 대략 1-축 대칭을 나타낸다.

객체	SymGAN	베이스라인
01	93.0	45.8
05	90.2	40.2
08	97.2	15.2
09	83.3	6.9

[0090]

[0091] 표 2는 테스트 세트로부터의 뷰들의 제한된 세트에 대한 $err_{vsd} < 0.3$ 에 대한 객체 리콜을 예시한다. 적어도 하나의 실시예에서, 객체들 및 선택된 뷰들은 도 15 내지 도 18에서 시각화된 것들과 일치된다. 적어도 하나의 실시예에서, SymGAN은 모든 테스트 뷰보다 더 큰 마진만큼 이들 대칭 뷰에 대해 직접 회귀 베이스라인(direct regression baseline)을 능가할 수 있으며, 이는 대칭적인 객체에 대한 SymGAN의 유효성을 나타낸다.

[0092] 도 19는 적어도 하나의 실시예에 따른, 이미지들의 쌍들에 대한 판별기 출력들의 시각화의 예를 예시한다. 적어도 하나의 실시예에서, 도 19는 표 2에서 사용된 제한된 테스트로부터의 이미지들의 쌍들에 대한 판별기 출력들의 시각화를 제공한다. 적어도 하나의 실시예에서, 도 8, 10, 12 및 14의 플롯들과 유사하게, 객체 01의 응답은 매끄럽고, 대부분의 뷰들이 대칭임을 나타내는 낮은 값들을 포함한다. 적어도 하나의 실시예에서, 객체 05의 응답은 180도 대칭성의 명확한 패턴 표시, 도 9로부터의 컬러 코너들을 갖는 정사각형의 노이즈가 많은 버전을 보여준다.

[0093] 적어도 하나의 실시예에서, 3D 예측 자세들, 및 실측 정보 자세가 용이한 시각화를 위해 2D 평면에 투영된다.

적어도 하나의 실시예에서, 이들 제한된 테스트 세트들에 대한 2개의 방법들의 정확도($err_{vsd} < 0.3$ 에 대한 객체 리콜) 결과들이 표 2에 도시된다. 적어도 하나의 실시예에서, SymGAN은 객체들 상에서 83% 이상을 달성할 수 있고, 도 16 내지 도 18은 직사각형 타입 객체들을 도시하며, 각각이 180도에서 대칭성을 가져서 자세들의 절반이 중복됨을 의미한다. 적어도 하나의 실시예에서, SymGAN은 이 분포를 학습할 수 있고, 그 출력은 지상 자세 공간의 약 절반만을 커버한다. 적어도 하나의 실시예에서, 베이스라인은 객체의 대칭성들을 학습할 수 없고 광범위한(종종 부정확한) 자세들을 출력한다. 적어도 하나의 실시예에서, 도 15는 이러한 제한된 테스트 세트의 모든 뷰에서 대칭인 객체를 도시한다. 적어도 하나의 실시예에서, 이상적으로는 SymGAN은 단일 포인트를 출력할 것이지만, 여기서 그 출력은 상당히 제한된다.

[0094] 적어도 하나의 실시예에서, 객체들 중 어느 것도 일부 매우 작은 피쳐들로 인해 정확히 대칭이 아니지만, 이들은 일반적으로 대칭이고 모델의 구별가능성에 대해 대칭이다. 적어도 하나의 실시예에서, 본 명세서에 설명되는 기법들은 사용되는 모델의 용량과 적응한다.

[0095] 적어도 하나의 실시예에서, SymGAN은 객체의 대칭성을 학습한다. 적어도 하나의 실시예에서, 제한된 테스트 세트에 대한 베이스라인에 대한 성능 이득은 전체 테스트 세트에 대한 성능 이득보다 더 크며, 대칭 뷰에 대한 SymGAN의 이점을 검증한다.

[0096] 적어도 하나의 실시예에서, 적대적 훈련을 활용하는 것은 객체 대칭성들(또는 자세들)의 임의의 수작업-라벨링 없이 텍스트가 없는 대칭적인 객체들에 대한 최신 시점(3D 배향) 회귀기를 훈련하는 것이 가능하다는 것을 보여주었다. 적어도 하나의 실시예에서, 방법은 3D 모델만을 요구하고 미분불가능한 블랙 박스를 수반하므로, 훈련 경사들을 검색하기 위한 샘플링 절차가 도입된다.

[0097] 도 20 내지 도 28은 적어도 하나의 실시예에 따른, T-LESS 데이터 세트에 대한 시점 예측기의 정성적 결과의 일

예를 예시한다. 적어도 하나의 실시예에서, 도 20에는, T-LESS 데이터 세트로부터의 제1 객체(2002), 제2 객체(2006), 제3 객체(2010), 및 제4 객체(2012)가 도시되어 있다. 적어도 하나의 실시예에서, 예측된 시점들은 각각의 객체에 대해 도시된다. 적어도 하나의 실시예에서, 제1 객체(2002)는 예측 시점(2004)을 갖고, 제2 객체(2006)는 예측 시점(2008)을 갖고, 제3 객체(2010)는 예측 시점(2012)을 갖고, 제4 객체(2014)는 예측 시점(2016)을 갖는다.

[0098] 적어도 하나의 실시예에서, 도 21에서, T-LESS 데이터 세트로부터의 제1 객체(2102), 제2 객체(2106), 제3 객체(2110) 및 제4 객체(2114)가 도시된다. 적어도 하나의 실시예에서, 예측된 시점들은 각각의 객체에 대해 도시된다. 적어도 하나의 실시예에서, 제1 객체(2102)는 예측 시점(2104)을 갖고, 제2 객체(2106)는 예측 시점(2108)을 갖고, 제3 객체(2110)는 예측 시점(2112)을 갖고, 제4 객체(2114)는 예측 시점(2116)을 갖는다.

[0099] 적어도 하나의 실시예에서, 도 22에는, T-LESS 데이터 세트로부터의 제1 객체(2202), 제2 객체(2206), 제3 객체(2210), 및 제4 객체(2214)가 도시되어 있다. 적어도 하나의 실시예에서, 예측된 시점들은 각각의 객체에 대해 도시된다. 적어도 하나의 실시예에서, 제1 객체(2202)는 예측 시점(2204)을 갖고, 제2 객체(2206)는 예측 시점(2208)을 가지며, 제3 객체(2210)는 예측 시점(2212)을 갖고, 제4 객체(2214)는 예측 시점(2216)을 갖는다.

[0100] 적어도 하나의 실시예에서, 도 23에서, T-LESS 데이터 세트로부터의 제1 객체(2302), 제2 객체(2306), 제3 객체(2310), 및 제4 객체(2314)가 도시되어 있다. 적어도 하나의 실시예에서, 예측된 시점들은 각각의 객체에 대해 도시된다. 적어도 하나의 실시예에서, 제1 객체(2302)는 예측 시점(2304)을 갖고, 제2 객체(2306)는 예측 시점(2308)을 갖고, 제3 객체(2310)는 예측 시점(2312)을 갖고, 제4 객체(2314)는 예측 시점(2316)을 갖는다.

[0101] 적어도 하나의 실시예에서, 도 24에는, T-LESS 데이터 세트로부터의 제1 객체(2402), 제2 객체(2406), 제3 객체(2410), 및 제4 객체(2414)가 도시되어 있다. 적어도 하나의 실시예에서, 예측된 시점들은 각각의 객체에 대해 도시된다. 적어도 하나의 실시예에서, 제1 객체(2402)는 예측 시점(2404)을 갖고, 제2 객체(2406)는 예측 시점(2408)을 가지며, 제3 객체(2410)는 예측 시점(2412)을 갖고, 제4 객체(2414)는 예측 시점(2416)을 갖는다.

[0102] 적어도 하나의 실시예에서, 도 25에서, T-LESS 데이터 세트로부터의 제1 객체(2502), 제2 객체(2506), 제3 객체(2510), 및 제4 객체(2514)가 도시되어 있다. 적어도 하나의 실시예에서, 예측된 시점들은 각각의 객체에 대해 도시된다. 적어도 하나의 실시예에서, 제1 객체(2502)는 예측 시점(2504)을 갖고, 제2 객체(2506)는 예측 시점(2508)을 갖고, 제3 객체(2510)는 예측 시점(2512)을 갖고, 제4 객체(2514)는 예측 시점(2516)을 갖는다.

[0103] 적어도 하나의 실시예에서, 도 26에서, T-LESS 데이터 세트로부터의 제1 객체(2602), 제2 객체(2606), 제3 객체(2610), 및 제4 객체(2614)가 도시되어 있다. 적어도 하나의 실시예에서, 예측된 시점들은 각각의 객체에 대해 도시된다. 적어도 하나의 실시예에서, 제1 객체(2602)는 예측 시점(2604)을 갖고, 제2 객체(2606)는 예측 시점(2608)을 가지며, 제3 객체(2610)는 예측 시점(2612)을 갖고, 제4 객체(2614)는 예측 시점(2616)을 갖는다.

[0104] 적어도 하나의 실시예에서, 도 27에서, T-LESS 데이터 세트로부터의 제1 객체(2702), 제2 객체(2706), 제3 객체(2710), 및 제4 객체(2714)가 도시되어 있다. 적어도 하나의 실시예에서, 예측된 시점들은 각각의 객체에 대해 도시된다. 적어도 하나의 실시예에서, 제1 객체(2702)는 예측 시점(2704)을 갖고, 제2 객체(2706)는 예측 시점(2708)을 가지며, 제3 객체(2710)는 예측 시점(2712)을 갖고, 제4 객체(2714)는 예측 시점(2716)을 갖는다.

[0105] 적어도 하나의 실시예에서, 도 28에서, T-LESS 데이터 세트로부터의 제1 객체(2802), 제2 객체(2806), 제3 객체(2810), 및 제4 객체(2814)가 도시되어 있다. 적어도 하나의 실시예에서, 예측된 시점들은 각각의 객체에 대해 도시된다. 적어도 하나의 실시예에서, 제1 객체(2802)는 예측 시점(2804)을 갖고, 제2 객체(2806)는 예측 시점(2808)을 가지며, 제3 객체(2810)는 예측 시점(2812)을 갖고, 제4 객체(2814)는 예측 시점(2816)을 갖는다.

[0106] 적어도 하나의 실시예에서, SymGAN은 대칭적 자세들에 걸친 완전한 다중 모드 분포의 추정이 아니라, 현재 자세의 포인트 추정을 제공한다. 그러나, 적어도 하나의 실시예에서, 분포는 판별기에 의해 학습된다. 적어도 하나의 실시예에서, 1회 고가의 오프라인 단계를 통해, 대칭 뷰들의 분포를 근사화하는 SymGAN을 훈련하는 데 사용되는 판별기를 사용하여 특정 테이블이 구성된다.

[0107] 추론 및 훈련 로직

[0108] 도 29a는 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용되는 추론 및/또는 훈련 로직(2915)을 예시한다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항은 도 29a 및/또는 도 29b와 관련하여 아래에 제공된다.

[0109] 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 제한없이, 하나 이상의 실시예의 양태들에서 추론

을 위해 훈련 및/또는 이용되는 신경망의 뉴런들 또는 계층들을 구성하기 위해 순방향 및/또는 출력 가중치 및/또는 입력/출력 데이터, 및/또는 다른 파라미터들을 저장하기 위한 코드 및/또는 데이터 저장소(2901)를 포함할 수 있다. 적어도 하나의 실시예에서, 훈련 로직(2915)은, 정수 및/또는 부동 소수점 유닛들(집합적으로, ALU들(arithmetic logic units))을 포함하는, 로직을 구성하기 위해 가중치 및/또는 다른 파라미터 정보가 로딩되는, 타이밍 및/또는 순서를 제어하는 그래프 코드 또는 다른 소프트웨어를 저장하기 위한 코드 및/또는 데이터 저장소(2901)를 포함하거나, 또는 이에 결합될 수 있다. 적어도 하나의 실시예에서, 그래프 코드 같은 코드는, 이 코드가 대응하는 신경망의 아키텍처에 기초하여, 프로세서 ALU들 내에 가중치 또는 다른 파라미터 정보를 로딩한다. 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2901)는 하나 이상의 실시예들의 양태들을 사용하여 훈련 및/또는 추론 동안 입력/출력 데이터 및/또는 가중치 파라미터들의 순방향 전파 동안 하나 이상의 실시예들과 관련하여 훈련되거나 사용되는 신경망의 각각의 계층의 가중치 파라미터들 및/또는 입력/출력 데이터를 저장한다. 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2901)의 임의의 부분은, 프로세서의 L1, L2, 또는 L3 캐시 또는 시스템 메모리를 포함한, 다른 온-칩 또는 오프-칩 데이터 저장소에 포함될 수 있다.

[0110] 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2901)의 임의의 부분은 하나 이상의 프로세서 또는 다른 하드웨어 로직 디바이스들 또는 회로들의 내부 또는 외부에 있을 수 있다. 적어도 하나의 실시예에서, 코드 및/또는 코드 및/또는 데이터 저장소(2901)는 캐시 메모리, 동적 랜덤 액세스 가능 메모리("DRAM"), 정적 랜덤 액세스 가능 메모리("SRAM"), 비휘발성 메모리(예를 들어, 플래시 메모리) 또는 다른 저장소일 수 있다. 적어도 하나의 실시예에서, 코드 및/또는 코드 및/또는 데이터 저장소(2901)가 프로세서의 내부에 있는지 또는 외부에 있는지, 예를 들어, 즉, DRAM, SRAM, 플래시 또는 일부 다른 저장소 타입으로 구성되는지의 선택은, 이용 가능한 저장소 온-칩 대 오프-칩, 수행되는 훈련 및/또는 추론 기능들의 레이턴시 요건들, 신경망의 추론 및/또는 훈련에 이용되는 배치 데이터 크기, 또는 이들 요인의 일부 조합에 의존할 수 있다.

[0111] 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 제한없이, 하나 이상의 실시예의 양태들에서 추론을 위해 훈련 및/또는 이용되는 신경망의 뉴런들 또는 계층들에 대응하는 역방향 및/또는 출력 가중치 및/또는 입력/출력 데이터를 저장하는 코드 및/또는 데이터 저장소(2905)를 포함할 수 있다. 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2905)는 하나 이상의 실시예의 양태들을 이용하여 훈련 및/또는 추론하는 동안에 입력/출력 데이터 및/또는 가중치 파라미터들의 역방향 전파 동안 하나 이상의 실시예와 관련하여 훈련되거나 이용되는 신경망의 각각의 계층의 가중치 파라미터들 및/또는 입력/출력 데이터를 저장한다. 적어도 하나의 실시예에서, 훈련 로직(2915)은, 정수 및/또는 부동 소수점 유닛들(집합적으로, ALU들(arithmetic logic units))을 포함하는, 로직을 구성하기 위해 가중치 및/또는 다른 파라미터 정보가 로딩되는, 타이밍 및/또는 순서를 제어하는 그래프 코드 또는 다른 소프트웨어를 저장하기 위한 코드 및/또는 데이터 저장소(2905)를 포함하거나, 또는 이에 결합될 수 있다. 적어도 하나의 실시예에서, 그래프 코드 같은 코드는, 이 코드가 대응하는 신경망의 아키텍처에 기초하여, 프로세서 ALU들 내에 가중치 또는 다른 파라미터 정보를 로딩한다. 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2905)의 임의의 부분은, 프로세서의 L1, L2, 또는 L3 캐시 또는 시스템 메모리를 포함한, 다른 온-칩 또는 오프-칩 데이터 저장소에 포함될 수 있다. 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2905)의 임의의 부분은 하나 이상의 프로세서 또는 다른 하드웨어 로직 디바이스들 또는 회로들의 내부 또는 외부에 있을 수 있다. 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2905)는, 캐시 메모리, DRAM, SRAM, 비휘발성 메모리(예를 들어, 플래시 메모리), 또는 다른 저장소일 수 있다. 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소 디바이스(2905)가 프로세서의 내부에 있는지 또는 외부에 있는지, 예를 들어, 즉, DRAM, SRAM, 플래시 또는 일부 다른 저장소 타입으로 구성되는지의 선택은, 이용 가능한 저장소 온-칩 대 오프-칩, 수행되는 훈련 및/또는 추론 기능들의 레이턴시 요건들, 신경망의 추론 및/또는 훈련에 이용되는 배치 데이터 크기, 또는 이들 요인의 일부 조합에 의존할 수 있다.

[0112] 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2901)와 코드 및/또는 데이터 저장소(2905)는 별개의 저장소 구조일 수 있다. 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2901)와 코드 및/또는 데이터 저장소(2905)는 동일한 저장소 구조일 수 있다. 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2901)와 코드 및/또는 데이터 저장소(2905)는 부분적으로 동일한 저장소 구조이고 부분적으로 별개의 저장소 구조일 수 있다. 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2901)와 코드 및/또는 데이터 저장소(2905)의 임의의 부분은, 프로세서의 L1, L2, 또는 L3 캐시 또는 시스템 메모리를 포함한, 다른 온-칩 또는 오프-칩 데이터 저장소에 포함될 수 있다.

[0113] 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 제한없이, 훈련 및/또는 추론 코드(예를 들어, 그래프 코드)에 적어도 부분적으로 기초하거나 이에 의해 표시되는 논리적 및/또는 수학적 연산들을 수행하고, 그

결과, 코드 및/또는 데이터 저장소(2901) 및/또는 코드 및/또는 데이터 저장소(2905)에 저장된 입력/출력 및/또는 가중치 파라미터 데이터의 함수인 액티베이션 저장소(2920)에 저장되는 액티베이션들(예를 들어, 신경망 내의 계층들 또는 뉴런들의 출력 값들)을 생성할 수 있는, 정수 및/또는 부동 소수점 유닛들을 포함한 하나 이상의 산술 로직 유닛(들)("ALU(들)")(2910)을 포함할 수 있다. 적어도 하나의 실시예에서, 액티베이션 저장소(2920)에 저장된 액티베이션들은 명령어들 또는 다른 코드를 수행하는 것에 응답하여 ALU(들)(2910)에 의해 수행되는 선형 대수 및/또는 행렬 기반의 수학에 따라 생성되며, 여기서, 코드 및/또는 데이터 저장소(2905) 및/또는 코드 및/또는 데이터 저장소(2901)에 저장된 가중치 값들, 바이어스 값들, 경사 정보, 모멘텀 값들, 또는 다른 파라미터들 또는 하이퍼파라미터들 같은 다른 값들과 함께 피연산자로 이용되며, 이들 중 임의의 것 또는 전부는 코드 및/또는 데이터 저장소(2905) 또는 코드 및/또는 데이터 저장소(2901) 또는 온-칩 또는 오프-칩의 또 다른 저장소에 저장될 수 있다.

[0114] 적어도 하나의 실시예에서, ALU(들)(2910)는 하나 이상의 프로세서 또는 다른 하드웨어 로직 디바이스들 또는 회로들 내에 포함되는 반면, 또 다른 실시예에서, ALU(들)(2910)는 이들을 이용하는 프로세서 또는 다른 하드웨어 로직 디바이스 또는 회로(예를 들어, 코프로세서)의 외부에 있을 수 있다. 적어도 하나의 실시예에서, ALU(들)(2910)은 프로세서의 실행 유닛들 내에 또는 그렇지 않으면 동일한 프로세서 내의 또는 상이한 타입들의 상이한 프로세서들(예를 들어, 중앙 처리 유닛들, 그래픽 처리 유닛들, 고정 기능 유닛들 등) 사이에 분산되어 있는 프로세서의 실행 유닛들에 의해 액세스가능한 ALU들의 뱅크 내에 포함될 수 있다. 적어도 하나의 실시예에서, 데이터 저장소(2901), 코드 및/또는 데이터 저장소(2905), 및 액티베이션 저장소(2920)는, 동일한 프로세서 또는 다른 하드웨어 로직 디바이스나 회로 상에 있을 수 있는 반면, 또 다른 실시예에서 이들은 상이한 프로세서들 또는 다른 하드웨어 로직 디바이스들이나 회로들, 또는 동일하거나 상이한 프로세서들 또는 다른 하드웨어 로직 디바이스들이나 회로들의 일부 조합에 있을 수 있다. 적어도 하나의 실시예에서, 액티베이션 저장소(2920)의 임의의 부분은, 프로세서의 L1, L2, 또는 L3 캐시 또는 시스템 메모리를 포함한, 다른 온-칩 또는 오프-칩 데이터 저장소에 포함될 수 있다. 또한, 추론 및/또는 훈련 코드는, 프로세서 또는 다른 하드웨어 로직 또는 회로가 액세스할 수 있고 프로세서의 인출, 디코딩, 스케줄링, 실행, 폐기 및/또는 다른 논리적 회로를 이용하여 인출 및/또는 처리될 수 있는 다른 코드와 함께 저장될 수 있다.

[0115] 적어도 하나의 실시예에서, 액티베이션 저장소(2920)는, 캐시 메모리, DRAM, SRAM, 비휘발성 메모리(예를 들어, 플래시 메모리), 또는 다른 저장소일 수 있다. 적어도 하나의 실시예에서, 액티베이션 저장소(2920)는 완전히 또는 부분적으로 하나 이상의 프로세서 또는 다른 논리적 회로 내부에 또는 외부에 있을 수 있다. 적어도 하나의 실시예에서, 액티베이션 저장소(2920)가 프로세서의 내부에 있는지 또는 외부에 있는지, 예를 들어, 즉, DRAM, SRAM, 플래시 또는 일부 다른 저장소 타입으로 구성되는지의 선택은, 이용 가능한 저장소 온-칩 대 오프-칩, 수행되는 훈련 및/또는 추론 기능들의 레이턴시 요건들, 신경망의 추론 및/또는 훈련에 이용되는 배치 데이터 크기, 또는 이들 요인의 일부 조합에 의존할 수 있다. 적어도 하나의 실시예에서, 도 29a에 예시된 추론 및/또는 훈련 로직(2915)은, Google의 Tensorflow® Processing Unit, Graphcore™의 추론 처리 유닛(IPU), Intel Corp의 Nervana® 예를 들어, "Lake Crest") 프로세서 같은, 주문형 집적 회로("ASIC")와 관련하여 이용될 수 있다. 적어도 하나의 실시예에서, 도 29a에 예시된 추론 및/또는 훈련 로직(2915)은, 중앙 처리 유닛("CPU") 하드웨어, 그래픽 처리 유닛("GPU") 하드웨어, 또는 필드 프로그램가능 게이트 어레이("FPGA") 같은 다른 하드웨어와 관련하여 이용될 수 있다.

[0116] 도 29b는 다양한 적어도 하나의 실시예에 따른 추론 및/또는 훈련 로직(2915)을 예시한다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 제한없이, 신경망 내의 뉴런들의 하나 이상의 계층에 대응하는 가중치 값들 또는 다른 정보와 관련하여 계산 리소스가 전용되거나 다른 방식으로 배타적으로 이용되는 하드웨어 로직을 포함할 수 있다. 적어도 하나의 실시예에서, 도 29b에 예시된 추론 및/또는 훈련 로직(2915)은 Google의 Tensorflow® 처리 유닛, Graphcore™의 추론 처리 유닛(IPU) 또는 Intel Corp의 Nervana® 예를 들어, "Lake Crest") 프로세서와 같은 주문형 집적 회로(ASIC)와 관련하여 사용될 수 있다. 적어도 하나의 실시예에서, 도 29b에 예시된 추론 및/또는 훈련 로직(2915) 중앙 처리 유닛(CPU) 하드웨어, 그래픽 처리 유닛(GPU) 하드웨어, 또는 필드 프로그램가능 게이트 어레이(FPGA) 같은 다른 하드웨어와 관련하여 이용될 수 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 제한없이, 코드(예를 들어, 그래프 코드), 가중치 값들, 및/또는 바이어스 값들, 경사 정보, 모멘텀 값들, 및/또는 다른 파라미터 또는 하이퍼파라미터 정보를 포함하는 다른 정보를 저장하는데 이용될 수 있는, 코드 및/또는 데이터 저장소(2901)와, 코드 및/또는 데이터 저장소(2905)를 포함한다. 도 29b에 예시된 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2901) 및 코드 및/또는 데이터 저장소(2905) 각각은 계산 하드웨어(2902) 및 계산 하드웨어(2906)와 같은 전용 계산 리소스와 각각 연관된다. 적어도 하나의 실시예에서, 계산 하드웨어(2902) 및 계산 하드웨어(2906) 각각은 하나 이상의 ALU를 포함

하고, 하나 이상의 ALU는, 코드 및/또는 데이터 저장소(2901)와 코드 및/또는 데이터 저장소(2905)에 저장된 정보에 관해서만 선행 대수 함수들 같은 수학적 함수들을 수행하고, 각각의 결과는 액티베이션 저장소(2920)에 저장된다.

[0117] 적어도 하나의 실시예에서, 코드 및/또는 데이터 저장소(2901 및 2905) 및 대응하는 계산 하드웨어(2902 및 2906) 각각은 각각 신경망의 상이한 계층들에 대응하며, 따라서 코드 및/또는 데이터 저장소(2901) 및 계산 하드웨어(2902)의 하나의 "저장소/계산 쌍(2901/2902)"으로부터의 결과적인 액티베이션은 신경망의 개념적 구성을 미러링하기 위해 코드 및/또는 데이터 저장소(2905) 및 계산 하드웨어(2906)의 다음 "저장소/계산 쌍(2905/2906)"에 대한 입력으로서 제공된다. 적어도 하나의 실시예에서, 저장소/계산 쌍들(2901/2902 및 2905/2906) 각각은 2개 이상의 신경망 계층에 대응할 수 있다. 적어도 하나의 실시예에서, 저장소/계산 쌍들(2901/2902 및 2905/2906)에 후속하거나 이와 병렬인 추가적인 저장소/계산 쌍(도시되지 않음)이 추론 및/또는 훈련 로직(2915)에 포함될 수 있다.

[0118] 신경망 훈련 및 배치

[0119] 도 30은 적어도 하나의 실시예에 따른 심층 신경망의 훈련 및 배치를 예시한다. 적어도 하나의 실시예에서, 훈련되지 않은 신경망(93006)은 훈련 데이터 세트(3002)를 사용하여 훈련된다. 적어도 하나의 실시예에서, 훈련 프레임워크(3004)는 파이토치(PyTorch) 프레임워크인 반면, 다른 실시예들에서, 훈련 프레임워크(3004)는 텐서플로우(Tensorflow), 부스트(Boost), 카페(Caffe), 마이크로소프트(Microsoft) 인지 툴킷/CNTK, MXNet, 체이너(Chainer), 케라스(Keras), Deeplearning4j, 또는 다른 훈련 프레임워크이다. 적어도 하나의 실시예에서, 훈련 프레임워크(3004)는 훈련되지 않은 신경망(3006)을 훈련하고, 훈련된 신경망(3008)을 생성하기 위해 본 명세서에 설명된 처리 리소스들을 사용하여 훈련될 수 있게 한다. 적어도 하나의 실시예에서, 가중치들은 랜덤하게 선택되거나 심층 신뢰망(deep belief network)을 이용한 사전 훈련에 의해 선택될 수 있다. 적어도 하나의 실시예에서, 훈련은 지도 방식, 부분 지도 방식, 또는 비지도 방식 중 어느 하나로 수행될 수 있다.

[0120] 적어도 하나의 실시예에서, 훈련되지 않은 신경망(3006)은 지도 학습을 사용하여 훈련되고, 훈련 데이터 세트(3002)는 입력에 대한 원하는 출력과 쌍을 이루는 입력을 포함하거나, 또는 훈련 데이터 세트(3002)는 알려진 출력을 갖는 입력을 포함하고 신경망(3006)의 출력은 수동으로 등급화된다. 적어도 하나의 실시예에서, 훈련되지 않은 신경망(3006)은 지도 방식으로 훈련되어 훈련 데이터 세트(3002)로부터의 입력들을 처리하고 결과적인 출력들을 예상된 또는 원하는 출력들의 세트에 대해 비교한다. 그 다음, 적어도 하나의 실시예에서, 예러들은 훈련되지 않은 신경망(3006)을 통해 다시 전파된다. 적어도 하나의 실시예에서, 훈련 프레임워크(3004)는 훈련되지 않은 신경망(3006)을 제어하는 가중치들을 조정한다. 적어도 하나의 실시예에서, 훈련 프레임워크(3004)는, 훈련되지 않은 신경망(3006)이, 새로운 데이터(3012) 같은 알려진 입력 데이터에 기초하여 결과(3014) 같은 정확한 답변들을 생성하기에 적합한 훈련된 신경망(3008) 같은 모델을 향해 얼마나 잘 수렴하는지 모니터링하는 도구를 포함한다. 적어도 하나의 실시예에서, 훈련 프레임워크(3004)는 확률적 경사 하강(stochastic gradient descent)과 같은 손실 함수 및 조정 알고리즘을 사용하여 훈련되지 않은 신경망(3006)의 출력을 정제하기 위해 가중치들을 조정하면서 훈련되지 않은 신경망(3006)을 반복적으로 훈련한다. 적어도 하나의 실시예에서, 훈련 프레임워크(3004)는 훈련되지 않은 신경망(3006)이 원하는 정확도를 달성할 때까지 훈련되지 않은 신경망(3006)을 훈련시킨다. 적어도 하나의 실시예에서, 훈련된 신경망(3008)은 이후 임의의 수의 머신러닝 연산들을 구현하도록 배치될 수 있다.

[0121] 적어도 하나의 실시예에서, 훈련되지 않은 신경망(3006)은 비지도 학습을 사용하여 훈련되고, 훈련되지 않은 신경망(3006)은 라벨링되지 않은 데이터를 사용하여 자체적 훈련을 시도한다. 적어도 하나의 실시예에서, 비지도 학습 훈련 데이터 세트(3002)는 임의의 연관된 출력 데이터 또는 "실측 정보" 데이터가 없는 입력 데이터를 포함할 것이다. 적어도 하나의 실시예에서, 훈련되지 않은 신경망(3006)은 훈련 데이터 세트(3002) 내의 그룹화들을 학습할 수 있고, 개별 입력들이 훈련되지 않은 데이터 세트(3002)에 어떻게 관련되는지를 결정할 수 있다. 적어도 하나의 실시예에서, 비지도 훈련이 자기-조직화 맵을 생성하는데 이용될 수 있고, 이는 새로운 데이터(3012)의 차원수를 감소시키는 데 유용한 연산들을 수행할 수 있는 훈련된 신경망(3008)의 타입이다. 적어도 하나의 실시예에서, 비지도 훈련은 또한, 새로운 데이터 세트(3012)의 정상 패턴들로부터 벗어나는 새로운 데이터 세트(3012)에서의 데이터 포인트들의 식별을 허용하는 이상 검출을 수행하기 위하여 이용될 수 있다.

[0122] 적어도 하나의 실시예에서, 반-지도 학습이 사용될 수 있고, 이는 훈련 데이터 세트(3002)가 라벨링된 데이터와 라벨링되지 않은 데이터의 혼합을 포함하는 기법이다. 적어도 하나의 실시예에서, 훈련 프레임워크(3004)는 예컨대 전이 학습 기법을 통해 증분적 학습을 수행하는데 이용될 수 있다. 적어도 하나의 실시예에서, 증분 학습

은 훈련된 신경망(3008)이 초기 훈련 동안 네트워크 내에 주입된 지식을 잊지 않고 새로운 데이터(3012)에 적응할 수 있게 한다.

[0123] 데이터 센터

[0124] 도 31은 적어도 하나의 실시예가 사용될 수 있는 예시적인 데이터 센터(3100)를 예시한다. 적어도 하나의 실시예에서, 데이터 센터(3100)는 데이터 센터 인프라스트럭처 계층(3110), 프레임워크 계층(3120), 소프트웨어 계층(3130) 및 애플리케이션 계층(3140)을 포함한다.

[0125] 적어도 하나의 실시예에서, 도 31에 도시된 바와 같이, 데이터 센터 인프라스트럭처 계층(3110)은 리소스 오케스트레이터(3112), 그룹화된 계산 리소스들(3114), 및 노드 계산 리소스들("노드 C.R.들")(3116(1)-3116(N))을 포함할 수 있으며, 여기서 "N"은 임의의 전체, 양의 정수를 나타낸다. 적어도 하나의 실시예에서, 노드 C.R.들(3116(1)-3116(N))은 임의의 수의 중앙 처리 유닛("CPU") 또는 다른 프로세서들(가속기들, 필드 프로그램가능 게이트 어레이(FPGA)들, 그래픽 프로세서들 등을 포함), 메모리 디바이스(예를 들어, 동적 판독 전용 메모리), 저장 디바이스들(예를 들어, 솔리드 스테이트 또는 디스크 드라이브들), 네트워크 입력/출력("NW I/O") 디바이스들, 네트워크 스위치들, 가상 머신들("VMs"), 전력 모듈들, 및 냉각 모듈들 등을 포함할 수 있지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, 노드 C.R.들 3116(1)-3116(N) 중 하나 이상의 노드 C.R.들은 위에서 언급된 계산 리소스들 중 하나 이상을 갖는 서버일 수 있다.

[0126] 적어도 하나의 실시예에서, 그룹화된 계산 리소스들(3114)은, 하나 이상의 랙(도시되지 않음) 내에 수용된 노드 C.R.들의 별개의 그룹화들, 또는 다양한 지리적 위치(역시 도시되지 않음)의 데이터 센터들에 수용된 많은 랙을 포함할 수 있다. 그룹화된 계산 리소스들(3114) 내의 노드 C.R.들의 개별 그룹화들은 하나 이상의 작업부하들을 지원하도록 구성되거나 할당될 수 있는 그룹화된 계산, 네트워크, 메모리 또는 저장소 리소스들을 포함할 수 있다. 적어도 하나의 실시예에서, CPU들 또는 프로세서들을 포함하는 여러 노드 C.R.들은, 하나 이상의 작업부하를 지원하는 계산 리소스들을 제공하기 위해 하나 이상의 랙 내에 그룹화될 수 있다. 적어도 하나의 실시예에서, 하나 이상의 랙은 또한, 임의의 수의 전력 모듈, 냉각 모듈, 및 네트워크 스위치를 임의의 조합으로 포함할 수 있다.

[0127] 적어도 하나의 실시예에서, 리소스 오케스트레이터(3112)는 하나 이상의 노드 C.R.들(3116(1)-3116(N)) 및/또는 그룹화된 계산 리소스들(3114)을 구성하거나 달리 제어할 수 있다. 적어도 하나의 실시예에서, 리소스 오케스트레이터(3112)는 데이터 센터(3100)에 대한 소프트웨어 설계 인프라스트럭처("SDI") 관리 엔티티를 포함할 수 있다. 적어도 하나의 실시예에서, 리소스 오케스트레이터는, 하드웨어, 소프트웨어 또는 이들의 일부 조합을 포함할 수 있다.

[0128] 적어도 하나의 실시예에서, 도 31에 도시된 바와 같이, 프레임워크 계층(3120)은, 잡 스케줄러(3132), 구성 관리자(3134), 리소스 관리자(3136), 및 분산형 파일 시스템(3138)을 포함한다. 적어도 하나의 실시예에서, 프레임워크 계층(3120)은 소프트웨어 계층(3130)의 소프트웨어(3132) 및/또는 애플리케이션 계층(3140)의 하나 이상의 애플리케이션(들)(3142)을 지원하기 위한 프레임워크를 포함할 수 있다. 적어도 하나의 실시예에서, 소프트웨어(3132) 또는 애플리케이션(들)(3142)은, 각각, Amazon Web Services, Google Cloud 및 Microsoft Azure에 의해 제공되는 것들 같은, 웹 기반 서비스 소프트웨어 또는 애플리케이션들을 포함할 수 있다. 적어도 하나의 실시예에서, 프레임워크 계층(3120)은 대규모 데이터 처리(예를 들어, "빅 데이터")를 위해 분산형 파일 시스템(3138)을 이용할 수 있는 Apache Spark™(이하, "Spark")와 같은 자유 및 오픈 소스 소프트웨어 웹 애플리케이션 프레임워크의 타입일 수 있지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, 잡 스케줄러(3132)는 데이터 센터(3100)의 다양한 계층들에 의해 지원되는 작업부하들의 스케줄링을 용이하게 하기 위해 스파크 드라이버를 포함할 수 있다. 적어도 하나의 실시예에서, 구성 관리자(3134)는 대규모 데이터 처리를 지원하기 위한 스파크 및 분산형 파일 시스템(3138)을 포함하는 소프트웨어 계층(3130) 및 프레임워크 계층(3120)과 같은 상이한 계층들을 구성하는 것이 가능할 수 있다. 적어도 하나의 실시예에서, 리소스 관리자(3136)는 분산형 파일 시스템(3138) 및 잡 스케줄러(3132)의 지원을 위해 맵핑되거나 할당되는 클러스터링된 또는 그룹화된 계산 리소스들을 관리하는 것이 가능할 수 있다. 적어도 하나의 실시예에서, 클러스터링된 또는 그룹화된 계산 리소스들은 데이터 센터 인프라스트럭처 계층(3110)에 그룹화된 계산 리소스(3114)를 포함할 수 있다. 적어도 하나의 실시예에서, 리소스 관리자(3136)는 리소스 오케스트레이터(3112)와 조율하여 이들 맵핑되거나 할당된 계산 리소스를 관리할 수 있다.

[0129] 적어도 하나의 실시예에서, 소프트웨어 계층(3130)에 포함된 소프트웨어(3132)는, 프레임워크 계층(3120)의 노드 C.R.들(3116(1)-3116(N)), 그룹화된 계산 리소스(3114), 및/또는 분산형 파일 시스템(3138)의 적어도 일부

에 의해 이용되는 소프트웨어를 포함할 수 있다. 소프트웨어의 하나 이상의 타입에는, 인터넷 웹 페이지 검색 소프트웨어, 이메일 바이러스 스캔 소프트웨어, 데이터베이스 소프트웨어, 및 스트리밍 비디오 콘텐츠 소프트웨어가 포함될 수 있지만, 이에 제한되지 않는다.

[0130] 적어도 하나의 실시예에서, 애플리케이션 계층(3140)에 포함된 애플리케이션(들)(3142)은, 적어도 코드 C.R.들(3116(1)-3116(N)), 그룹화된 계산 리소스(3114), 및/또는 프레임워크 계층(3120)의 분산형 파일 시스템(3138)의 부분들에 의해 이용되는 하나 이상의 타입의 애플리케이션을 포함할 수 있다. 애플리케이션의 하나 이상의 타입에는, 임의의 수의 유전체학 애플리케이션, 인지 계산, 및 훈련 또는 추론 소프트웨어, 머신 러닝 프레임워크 소프트웨어(예를 들어, PyTorch, TensorFlow, Caffe 등) 또는 하나 이상의 실시예와 관련하여 이용되는 다른 머신 러닝 애플리케이션들을 포함한 머신 러닝 애플리케이션이 포함될 수 있지만, 이에 제한되지 않는다.

[0131] 적어도 하나의 실시예에서, 구성 관리자(3134), 리소스 관리자(3136), 및 리소스 오케스트레이터(3112) 중 임의의 것은 임의의 기술적으로 실현 가능한 방식으로 취득되는 임의의 양 및 타입의 데이터에 기초하여 임의의 수 및 타입의 자기-수정 액션들을 구현할 수 있다. 적어도 하나의 실시예에서, 자기-수정 액션들은 데이터 센터(3100)의 데이터 센터 운영자가 가능하게는 불량 구성 결정들을 행하는 것을 완화하고, 가능하게는 데이터 센터의 부족이용 및/또는 열악한 성능의 부분들을 방지할 수 있다.

[0132] 적어도 하나의 실시예에서, 데이터 센터(3100)는 본 명세서에 설명된 하나 이상의 실시예에 따라 하나 이상의 머신 러닝 모델을 훈련시키거나 하나 이상의 머신 러닝 모델을 사용하여 정보를 예측 또는 추론하기 위한 도구, 서비스, 소프트웨어 또는 다른 리소스를 포함할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 머신 러닝 모델은, 데이터 센터(3100)에 관하여 전술한 소프트웨어 및 계산 리소스를 이용하여 신경망 아키텍처에 따라 가중치 파라미터를 산출함으로써 훈련될 수 있다. 적어도 하나의 실시예에서, 하나 이상의 신경망에 대응하는 훈련된 머신 러닝 모델은, 본 명세서에 설명된 하나 이상의 훈련 기법을 통해 계산된 가중치 파라미터를 이용함으로써 데이터 센터(3100)에 관하여 전술한 리소스를 이용하여 정보를 추론 또는 예측하는데 이용될 수 있다.

[0133] 적어도 하나의 실시예에서, 데이터 센터는, CPU들, 주문형 집적 회로(application-specific integrated circuit)(ASIC)들, GPU들, FPGA들, 또는 다른 하드웨어를 이용하여, 전술한 리소스들을 이용해 훈련 및/또는 추론을 수행할 수 있다. 더욱이, 전술한 하나 이상의 소프트웨어 및/또는 하드웨어 리소스들은, 사용자가, 이미지 인식, 음성 인식 또는 다른 인공지능 서비스들 같은 정보의 추론을 훈련하거나 수행하는 것을 허용하는 서비스로서 구성될 수 있다.

[0134] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 신경망 훈련 연산들, 신경망 기능들 및/또는 아키텍처들, 또는 본 명세서에 설명된 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 연산들을 추론하거나 예측하기 위해 31의 시스템에서 이용될 수 있다. 적어도 하나의 실시예에서, 자율 차량을 제어 및 지향시키기 위해, 후술되는 컴퓨터 하드웨어로 구현되는 내비게이션 시스템에 객체 배향이 제공된다.

[0135] 자율 차량

[0136] 도 32a는 적어도 하나의 실시예에 따른 자율 차량(3200)의 예를 예시한다. 적어도 하나의 실시예에서, 자율 차량(3200)(대안적으로 본 명세서에서 "차량(3200)"으로 지칭됨)는, 제한없이, 승용차, 예컨대 자동차, 트럭, 버스, 및/또는 하나 이상의 승객들을 수용하는 다른 타입의 차량일 수 있다. 적어도 하나의 실시예에서, 차량(3200)은 화물을 운반하기 위해 사용되는 세미-트랙터-트레일러 트럭일 수 있다. 적어도 하나의 실시예에서, 차량(3200)은 비행기, 로봇 차량, 또는 다른 종류의 차량일 수 있다.

[0137] 자율 차량들은 미국 교통부의 부서인 미국 교통국(National Highway Traffic Safety Administration)("NHTSA")과 자동차 기술자 협회(Society of Automotive Engineers)("SAE")에서 정의한 자동화 레벨 "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles"의 관점에서 설명될 수 있다(예를 들어, 2018년 6월 15일자로 공개된 표준 번호 제J3016-201806호, 2016년 9월 30일자로 공개된 표준 번호 제J3016-201609호, 및 이 표준의 이전 및 미래의 버전들). 하나 이상의 실시예들에서, 차량(3200)은 자율 주행 레벨들의 레벨 1 - 레벨 5 중 하나 이상에 따른 기능성이 가능할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 차량(3200)은, 실시예에 따라, 조건부 자동화(레벨 3), 고도 자동화(레벨 4), 및/또는 완전 자동화(레벨 5)가 가능할 수 있다.

- [0138] 적어도 하나의 실시예에서, 차량(3200)은 차대, 차량 본체, 바퀴들(예를 들어, 2, 4, 6, 8, 18 등), 타이어들, 차축들, 및 차량의 다른 컴포넌트들과 같은 컴포넌트들을 제한없이 포함할 수 있다. 적어도 하나의 실시예에서, 차량(3200)은 내연 기관, 하이브리드 전력 플랜트, 완전 전기 기관(all-electric engine), 및/또는 다른 추진 시스템 타입과 같은 추진 시스템(3250)을 포함할 수 있지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, 추진 시스템(3250)은 차량(3200)의 추진을 가능하게 하는 변속기를 제한없이 포함할 수 있는 차량(3200)의 구동 트레인에 연결될 수 있다. 적어도 하나의 실시예에서, 추진 시스템(3250)은 스로틀/가속기(들)(3252)로부터 신호들을 수신하는 것에 응답하여 제어될 수 있다.
- [0139] 적어도 하나의 실시예에서, 제한없이, 조향 휠을 포함할 수 있는 조향 시스템(3254)은 추진 시스템(3250)이 동작하고 있을 때(예를 들어, 차량이 움직이고 있을 때) 차량(3200)을 (예를 들어, 원하는 경로 또는 루트를 따라) 조향하는 데 사용된다. 적어도 하나의 실시예에서, 조향 시스템(3254)은 조향 액추에이터(들)(3256)로부터 신호들을 수신할 수 있다. 조향 휠은 완전 자동화(레벨 5) 기능에서는 선택적일 수 있다. 적어도 하나의 실시예에서, 브레이크 센서 시스템(3246)은 브레이크 액추에이터(들)(3248) 및/또는 브레이크 센서들로부터 신호들을 수신하는 것에 응답하여 차량 브레이크들을 동작시키기 위해 사용될 수 있다.
- [0140] 적어도 하나의 실시예에서, 제어기(들)(3236)는 하나 이상의 시스템-온-칩("SoC")(도 32a에 도시되지 않음) 및/또는 그래픽 처리 유닛(들)("GPU(들)")을 포함할 수 있지만 이에 제한되지 않으며, (예를 들어, 커맨드들을 나타내는) 신호들을 차량(3200)의 하나 이상의 컴포넌트 및/또는 시스템에 제공한다. 예를 들어, 적어도 하나의 실시예에서, 제어기(들)(3236)는 브레이크 액추에이터들(3248)을 통해 차량 브레이크들을 동작시키기 위해, 조향 액추에이터(들)(3256)를 통해 조향 시스템(3254)을 동작시키기 위해, 스로틀/가속기(들)(3252)를 통해 추진 시스템(3250)을 동작시키기 위해 신호들을 전송할 수 있다. 제어기(들)(3236)는 자율 주행을 가능하게 하고/하거나 인간 운전자가 차량(3200)을 운전하는 것을 보조하기 위해 센서 신호들을 처리하고 동작 커맨드들(예를 들어, 커맨드들을 나타내는 신호들)을 출력하는 하나 이상의 온보드(예를 들어, 통합된) 계산 디바이스(예를 들어, 슈퍼컴퓨터)를 포함할 수 있다. 적어도 하나의 실시예에서, 제어기(들)(3236)는 자율 주행 기능들을 위한 제1 제어기(3236), 기능적 안전 기능들을 위한 제2 제어기(3236), 인공 지능 기능(예를 들어, 컴퓨터 비전)을 위한 제3 제어기(3236), 인포테인먼트 기능을 위한 제4 제어기(3236), 응급 상황들에서의 중복성을 위한 제5 제어기(3236), 및/또는 다른 제어기들을 포함할 수 있다. 적어도 하나의 실시예에서, 단일의 제어기(3236)는 상기 기능들 중 2개 이상을 핸들링할 수 있고, 2개 이상의 제어기(3236)는 단일의 기능, 및/또는 이들의 임의의 조합을 핸들링할 수 있다.
- [0141] 적어도 하나의 실시예에서, 제어기(들)(3236)는 하나 이상의 센서들로부터 수신된 센서 데이터(예를 들어, 센서 입력들)에 응답하여 차량(3200)의 하나 이상의 컴포넌트들 및/또는 시스템들을 제어하기 위한 신호들을 제공한다. 적어도 하나의 실시예에서, 센서 데이터는, 예를 들어 및 제한없이, GNSS(global navigation satellite systems) 센서(들)(3258)(예를 들어, Global Positioning System 센서(들)), RADAR 센서(들)(3260), 초음파 센서(들)(3262), LIDAR 센서(들)(3264), IMU(inertial measurement unit) 센서(들)(3266)(예를 들어, 가속도계(들), 자이로스코프(들), 자기 나침반(들), 자력계(들) 등), 마이크로폰(들)(3296), 스테레오 카메라(들)(3268), 와이드-뷰 카메라(들)(3270)(예를 들어, 어안 카메라), 적외선 카메라(들)(3272), 서라운드 카메라(들)(3274)(예를 들어, 360도 카메라), 장거리 카메라(도 32a에 도시되지 않음), (도 32a에 도시되지 않은) 중거리 카메라(들), (예를 들어, 차량(3200)의 속도를 측정하기 위한) 속도 센서(들)(3244), 진동 센서(들)(3242), 조향 센서(들)(3240), (예를 들어, 브레이크 센서 시스템(3246)의 일부로서의) 브레이크 센서(들), 및/또는 다른 센서 타입으로부터 수신될 수 있다.
- [0142] 적어도 하나의 실시예에서, 제어기(들)(3236) 중 하나 이상은 차량(3200)의 계기 클러스터(3232)로부터 입력들(예를 들어, 입력 데이터에 의해 표현됨)을 수신하고, 인간-머신 인터페이스("HMI") 디스플레이(3234), 가청 표시기, 스피커를 통해, 및/또는 차량(3200)의 다른 컴포넌트들을 통해 출력들(예를 들어, 출력 데이터, 디스플레이 데이터 등에 의해 표현됨)을 제공할 수 있다. 적어도 하나의 실시예에서, 출력들은, 차속, 속도, 시간, 지도 데이터(예를 들어, 고화질 지도(도 32a에 도시되지 않음)), 위치 데이터(예를 들어, 지도 상의 것 같은 차량(3200)의 위치), 방향, 다른 차량들의 위치(예를 들어, 점유 그리드), 제어기(들)(3236)에 의해 인지되는 객체들 및 객체들의 상태에 관한 정보 등과 같은 정보를 포함할 수 있다. 예를 들어, 적어도 하나의 실시예에서, HMI 디스플레이(3234)는 하나 이상의 객체들의 존재에 관한 정보(예를 들어, 거리 표지판, 주의 표지판, 트래픽 신호등 변경 등), 및/또는 운전 기동 차량이 행해졌거나, 행하고 있거나, 행할 것에 관한 정보(예를 들어, 지금 차선을 변경하는 것, 2 마일 떨어진 출구(34B)를 택하는 것 등)를 디스플레이할 수 있다.
- [0143] 적어도 하나의 실시예에서, 차량(3200)은 하나 이상의 네트워크들을 통해 통신하기 위해 무선 안테나(들)(3226)

및/또는 모뎀(들)을 사용할 수 있는 네트워크 인터페이스(3224)를 더 포함한다. 예를 들어, 적어도 하나의 실시예에서, 네트워크 인터페이스(3224)는 롱 텀 에볼루션(Long-Term Evolution)("LTE"), 광대역 코드 분할 다중 액세스(Wideband Code Division Multiple Access)("WCDMA"), 범용 이동 통신 시스템(Universal Mobile Telecommunications System)("UMTS"), 이동 통신을 위한 글로벌 시스템(Global System for Mobile communication)("GSM"), IMT-CDMA 멀티 캐리어(Multi-Carrier)("CDMA2000") 등을 통해 통신하는 것이 가능할 수 있다. 적어도 하나의 실시예에서, 무선 안테나(들)(3226)는 또한 블루투스, 블루투스 저 에너지("LE"), Z-Wave, ZigBee 등과 같은 근거리 통신망(들) 및/또는 LoRaWAN, SigFox 등과 같은 저전력 광역 네트워크(들)("LPWAN들")를 이용하여, 환경(예를 들어, 차량들, 모바일 디바이스들 등) 내의 객체들 사이의 통신을 가능하게 할 수 있다.

[0144] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 신경망 훈련 연산들, 신경망 기능들 및/또는 아키텍처들, 또는 본 명세서에 설명된 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 연산들을 추론하거나 예측하기 위해 도 32a의 시스템에서 이용될 수 있다. 적어도 하나의 실시예에서, 자율 차량을 제어 및 지향시키기 위해, 후술되는 컴퓨터 하드웨어로 구현되는 내비게이션 시스템에 객체 배향이 제공된다.

[0145] 도 32b는 적어도 하나의 실시예에 따른, 도 32a의 자율 차량(3200)에 대한 카메라 위치들 및 시야들의 예를 예시한다. 적어도 하나의 실시예에서, 카메라들 및 각각의 시야들은 하나의 예시적인 실시예이고 제한을 의도하지 않는다. 예를 들어, 적어도 하나의 실시예에서, 추가적인 및/또는 대안의 카메라들이 포함될 수 있고 및/또는 카메라들이 차량(3200) 상의 상이한 위치들에 위치될 수 있다.

[0146] 적어도 하나의 실시예에서, 카메라들에 대한 카메라 타입들은 차량(3200)의 컴포넌트들 및/또는 시스템들과 함께 사용하도록 구성될 수 있는 디지털 카메라들을 포함할 수 있지만, 이에 제한되지 않는다. 카메라(들)는 자동차 안전 무결성 레벨("ASIL") B 및/또는 다른 ASIL에서 동작할 수 있다. 적어도 하나의 실시예에서, 카메라 타입들은 실시예에 따라 초당 60 프레임(fps), 1220 fps, 240 fps 등과 같은 임의의 이미지 포착 레이트가 가능할 수 있다. 적어도 하나의 실시예에서, 카메라들은 롤링 셔터들, 글로벌 셔터들, 다른 타입의 셔터, 또는 이들의 조합을 사용하는 것이 가능할 수 있다. 적어도 하나의 실시예에서, 컬러 필터 어레이는 적색 투명 투명 투명("RCCC") 컬러 필터 어레이, 적색 투명 투명 청색("RCCB") 컬러 필터 어레이, 적색 청색 녹색 투명("RBGC") 컬러 필터 어레이, 포베온 X3 컬러 필터 어레이, 베이어 센서들("RGGB") 컬러 필터 어레이, 모노크롬 센서 컬러 필터 어레이, 및/또는 다른 타입의 컬러 필터 어레이를 포함할 수 있다. 적어도 하나의 실시예에서, RCCC, RCCB, 및/또는 RBGC 컬러 필터 어레이를 갖는 카메라들과 같은 투명 픽셀 카메라들은 광 감도를 증가시키기 위한 노력으로 사용될 수 있다.

[0147] 적어도 하나의 실시예에서, 카메라(들) 중 하나 이상은 (예를 들어, 중복 또는 장애 안전 설계의 일부로서) ADAS(advanced driver assistance systems) 기능들을 수행하는 데 사용될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 차선 이탈 경고, 교통 표지판 보조 및 지능형 전조등 제어를 포함하는 기능들을 제공하기 위해 다 기능 모노 카메라가 설치될 수 있다. 적어도 하나의 실시예에서, 카메라(들) 중 하나 이상(예를 들어, 카메라들 전부)은 이미지 데이터(예를 들어, 비디오)를 동시에 기록하고 제공할 수 있다.

[0148] 적어도 하나의 실시예에서, 카메라들 중 하나 이상은 카메라의 이미지 데이터 포착 능력들을 방해할 수 있는 자동차 내로부터의 미광 및 반사들(예를 들어, 윈드실드 미러들에서 반사되는 대시보드로부터의 반사들)을 차단하기 위해, 맞춤 설계된 (3차원("3D") 인쇄) 조립체와 같은 장착 조립체에 장착될 수 있다. 윈-미러 장착 조립체들을 참조하면, 적어도 하나의 실시예에서, 윈-미러 조립체들은 카메라 장착 플레이트가 윈-미러의 형상과 일치하도록 맞춤 3D 인쇄될 수 있다. 적어도 하나의 실시예에서, 카메라(들)는 윈-미러에 통합될 수 있다. 사이드-뷰 카메라들에 대해, 카메라(들)는 또한 적어도 하나의 실시예에서 객실의 각각의 코너에서 4개의 필러들 내에 통합될 수 있다.

[0149] 적어도 하나의 실시예에서, 차량(3200)의 전방에 있는 환경의 부분들을 포함하는 시야를 갖는 카메라들(예를 들어, 전면 카메라들)은 전방을 향한 경로들 및 장애물들을 식별하는 데 도움을 주는 것은 물론, 제어기들(3236) 및/또는 제어 SoC들 중 하나 이상의 도움을 받아, 점유 그리드를 생성하는 데 및/또는 선호된 차량 경로들을 결정하는 데 중요한 정보를 제공하는 데 도움을 주기 위한 서라운드 뷰를 위해 사용될 수 있다. 적어도 하나의 실시예에서, 전면 카메라들은, 비상 제동, 보행자 검출, 및 충돌 회피를 비롯한(이들로 제한되지 않음), LIDAR과

동일한 ADAS 기능들 중 다수를 수행하는 데 사용될 수 있다. 적어도 하나의 실시예에서, 전면 카메라들은 또한, 제한없이, "LDW"(Lane Departure Warnings), "ACC"(Autonomous Cruise Control), 및/또는 교통 표지판 인식과 같은 다른 기능들을 포함하는 ADAS 기능들 및 시스템들에 사용될 수 있다.

[0150] 적어도 하나의 실시예에서, 예를 들어, CMOS("complementary metal oxide semiconductor") 컬러 이미저(color imager)를 포함하는 단안 카메라 플랫폼(monocular camera platform)을 포함하는 다양한 카메라들이 전면 구성에서 사용될 수 있다. 적어도 하나의 실시예에서, 와이드-뷰 카메라(3270)는 주위로부터 시야에 들어오는 객체들(예를 들어, 보행자들, 교차하는 트래픽 또는 자전거들)을 인지하는데 사용될 수 있다. 하나의 와이드-뷰 카메라(3270)만이 도 32b에 예시되지만, 다른 실시예들에서, 차량(3200) 상에 임의의 수(0을 포함함)의 와이드-뷰 카메라(들)(3270)가 존재할 수 있다. 적어도 하나의 실시예에서, 임의의 수의 장거리 카메라(들)(3298)(예를 들어, 장거리-시야 스테레오 카메라 쌍)가 깊이 기반 객체 검출을 위해, 특히 신경망이 아직 훈련되지 않은 객체들에 대해 사용될 수 있다. 적어도 하나의 실시예에서, 장거리 카메라(들)(3298)는 또한 객체 검출 및 분류는 물론, 기본 객체 추적을 위해 사용될 수 있다.

[0151] 적어도 하나의 실시예에서, 임의의 수의 스테레오 카메라(들)(3268)가 또한 전면 구성에 포함될 수 있다. 적어도 하나의 실시예에서, 스테레오 카메라(들)(3268) 중 하나 이상은 단일 칩 상의 통합된 제어기 영역 네트워크("CAN") 또는 이더넷 인터페이스를 갖는 멀티 코어 마이크로프로세서 및 프로그램가능 로직("FPGA")을 제공할 수 있는 스케일러블 처리 유닛을 포함하는 통합된 제어 유닛을 포함할 수 있다. 적어도 하나의 실시예에서, 이러한 유닛은 이미지 내의 모든 포인트들에 대한 거리 추정을 포함하는, 차량(3200)의 환경의 3D 맵을 생성하기 위해 사용될 수 있다. 적어도 하나의 실시예에서, 스테레오 카메라(들)(3268) 중 하나 이상은, 제한없이, 2개의 카메라 렌즈(각각 좌측 및 우측에 하나씩) 및 차량(3200)으로부터 타겟 객체까지의 거리를 측정하고 생성된 정보(예를 들어, 메타데이터)를 사용하여 자율 비상 제동 및 차선 이탈 경고 기능들을 활성화할 수 있는 이미지 처리 칩을 포함할 수 있는 컴팩트 스테레오 비전 센서(들)를 제한없이 포함할 수 있다. 적어도 하나의 실시예에서, 다른 타입들의 스테레오 카메라(들)(3268)가 본 명세서에 설명된 것들에 추가로 또는 대안적으로 이용될 수 있다.

[0152] 적어도 하나의 실시예에서, 차량(3200)의 측면에 대한 환경의 부분들을 포함하는 시야를 갖는 카메라들(예를 들어, 사이드-뷰 카메라들)이 서라운드 뷰를 위해 사용되어, 점유 그리드를 생성하고 업데이트하는 것은 물론, 측면 충격 충돌 경고를 생성시키는 데 사용되는 정보를 제공할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 서라운드 카메라(들)(3274)(예를 들어, 도 32b에 예시된 바와 같은 4개의 서라운드 카메라(3274))는 차량(3200) 상에 위치될 수 있다. 서라운드 카메라(들)(3274)는 임의의 수 및 조합의 와이드-뷰 카메라(들)(3270), 어안 카메라(들), 360도 카메라(들) 등을 제한없이 포함할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 4개의 어안 카메라가 차량(3200)의 전방, 후방 및 측면에 위치될 수 있다. 적어도 하나의 실시예에서, 차량(3200)은 3개의 서라운드 카메라(들)(3274)(예를 들어, 좌측, 우측, 및 후방)를 사용할 수 있고, 하나 이상의 다른 카메라(들)(예를 들어, 전면 카메라)를 제4 서라운드 뷰 카메라로서 활용할 수 있다.

[0153] 적어도 하나의 실시예에서, 차량(3200)의 후방에 대한 환경의 부분들을 포함하는 시야를 갖는 카메라들(예를 들어, 후방-뷰 카메라들)은 주차 보조, 서라운드 뷰, 후방 충돌 경고들, 및 점유 그리드의 생성 및 업데이트를 위해 사용될 수 있다. 적어도 하나의 실시예에서, 본 명세서에 설명된 바와 같이, 전면 카메라(들)(예를 들어, 장거리 카메라들(3298) 및/또는 중거리 카메라(들)(3276), 스테레오 카메라(들)(3268), 적외선 카메라(들)(3272) 등)로서 또한 적합한 카메라들을 포함하지만 이에 제한되지 않는 매우 다양한 카메라들이 사용될 수 있다.

[0154] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 신경망 훈련 연산들, 신경망 기능들 및/또는 아키텍처들, 또는 본 명세서에 설명된 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 연산들을 추론하거나 예측하기 위해 도 32b의 시스템에서 이용될 수 있다. 적어도 하나의 실시예에서, 자율 차량을 제어 및 지향시키기 위해, 후술되는 컴퓨터 하드웨어로 구현되는 내비게이션 시스템에 객체 배향이 제공된다.

[0155] 도 32c는 적어도 하나의 실시예에 따른, 도 32a의 자율 차량(3200)에 대한 예시적인 시스템 아키텍처를 예시하는 블록도이다. 적어도 하나의 실시예에서, 도 32c에서의 차량(3200)의 컴포넌트들, 피쳐들, 및 시스템들 각각은 버스(3202)를 통해 접속되는 것으로 예시되어 있다. 적어도 하나의 실시예에서, 버스(3202)는, 제한없이, CAN 데이터 인터페이스(대안적으로 본 명세서에서 "CAN 버스"라고 지칭됨)를 포함할 수 있다. 적어도 하나의 실

시에에서, CAN은, 브레이크들, 가속, 제동, 조향, 윈드실드 와이퍼들(windshield wipers) 등의 작동과 같은, 차량(3200)의 다양한 피쳐들 및 기능의 제어를 돕는 데 사용되는 차량(3200) 내부의 네트워크일 수 있다. 적어도 하나의 실시예에서, 버스(3202)는 각각이 그 자신의 고유 식별자(예를 들어, CAN ID)를 갖는 수십 또는 심지어 수백 개의 노드를 갖도록 구성될 수 있다. 적어도 하나의 실시예에서, 버스(3202)는 조향 휠 각도, 지면 속도, 분당 엔진 회전수("RPM"), 버튼 위치들, 및/또는 다른 차량 상태 표시자들을 찾기 위해 판독될 수 있다. 적어도 하나의 실시예에서, 버스(3202)는 ASIL B를 준수하는 CAN 버스일 수 있다.

[0156] 적어도 하나의 실시예에서, CAN에 추가로 또는 대안적으로, FlexRay 및/또는 이더넷이 사용될 수 있다. 적어도 하나의 실시예에서, 0개 이상의 CAN 버스, 0개 이상의 FlexRay 버스, 0개 이상의 이더넷 버스, 및/또는 상이한 프로토콜을 사용하는 0개 이상의 다른 타입의 버스를 제한없이 포함할 수 있는 임의의 수의 버스(3202)가 있을 수 있다. 적어도 하나의 실시예에서, 2개 이상의 버스들(3202)이 상이한 기능들을 수행하는 데 사용될 수 있으며, 및/또는 중복성을 위해 사용될 수 있다. 예를 들어, 제1 버스(3202)는 충돌 회피 기능을 위해 사용될 수 있고 제2 버스(3202)는 작동 제어를 위해 사용될 수 있다. 적어도 하나의 실시예에서, 각각의 버스(3202)는 차량(3200)의 컴포넌트들 중 임의의 것과 통신할 수 있고, 2개 이상의 버스들(3202)은 동일한 컴포넌트들과 통신할 수 있다. 적어도 하나의 실시예에서, 임의의 수의 시스템(들) 온 칩(들)("SoC(들)") (3204) 각각, 각각의 제어기(들)(3236), 및/또는 차량 내의 각각의 컴퓨터는 동일한 입력 데이터(예를 들어, 차량(3200)의 센서들로부터의 입력들)에 액세스할 수 있고, CAN 버스와 같은 공통 버스에 접속될 수 있다.

[0157] 적어도 하나의 실시예에서, 차량(3200)은 도 32a와 관련하여 본 명세서에 설명된 것들과 같은 하나 이상의 제어기(들)(3236)를 포함할 수 있다. 제어기(들)(3236)는 다양한 기능들을 위해 사용될 수 있다. 적어도 하나의 실시예에서, 제어기(들)(3236)는 차량(3200)의 다양한 다른 컴포넌트들 및 시스템들 중 임의의 것에 결합될 수 있고, 차량(3200)의 제어, 차량(3200)의 인공 지능, 차량(3200)에 대한 인포테인먼트 등을 위해 사용될 수 있다.

[0158] 적어도 하나의 실시예에서, 차량(3200)은 임의의 수의 SoC(3204)를 포함할 수 있다. SoC들(3204) 각각은, 제한 없이, 중앙 처리 유닛들("CPU(들)") (3206), 그래픽 처리 유닛들("GPU(들)") (3208), 프로세서(들)(3210), 캐시(들)(3212), 가속기(들)(3214), 데이터 저장소(들)(3216), 및/또는 도시되지 않은 다른 컴포넌트들 및 피쳐들을 포함할 수 있다. 적어도 하나의 실시예에서, SoC(들)(3204)는 다양한 플랫폼들 및 시스템들에서 차량(3200)을 제어하는 데 사용될 수 있다. 예를 들어, 적어도 하나의 실시예에서, SoC(들)(3204)는 하나 이상의 서버들(도 32c에 도시되지 않음)로부터 네트워크 인터페이스(3224)를 통해 맵 리프레시들 및/또는 업데이트들을 획득할 수 있는 HD(High Definition) 맵(3222)과 시스템(예를 들어, 차량(3200)의 시스템)에서 조합될 수 있다.

[0159] 적어도 하나의 실시예에서, CPU(들)(3206)는 CPU 클러스터 또는 CPU 컴플렉스(본 명세서에서 대안적으로 "CCPLEX"라고 지칭됨)를 포함할 수 있다. 적어도 하나의 실시예에서, CPU(들)(3206)는 다수의 코어들 및/또는 레벨 2("L2") 캐시들을 포함할 수 있다. 예를 들어, 적어도 하나의 실시예에서, CPU(들)(3206)는 코히어런트 멀티-프로세서 구성에서 8개의 코어를 포함할 수 있다. 적어도 하나의 실시예에서, CPU(들)(3206)는 각각의 클러스터가 전용 L2 캐시(예를 들어, 2 MB L2 캐시)를 갖는 4개의 듀얼 코어 클러스터를 포함할 수 있다. 적어도 하나의 실시예에서, CPU(들)(3206)(예를 들어, CPLEX)는 CPU(들)(3206)의 클러스터들의 임의의 조합이 임의의 주어진 시간에 활성 상태일 수 있게 하는 동시 클러스터 동작을 지원하도록 구성될 수 있다.

[0160] 적어도 하나의 실시예에서, CPU(들)(3206) 중 하나 이상은, 제한없이, 다음의 피쳐들 중 하나 이상을 포함하는 전력 관리 능력들을 구현할 수 있다: 개별 하드웨어 블록들은 동적 전력을 절약하기 위해 유휴 상태일 때 자동으로 클록 게이팅될 수 있고; 각각의 코어 클록은 인터럽트를 위한 대기("WFI")/이벤트를 위한 대기("WFE") 명령어들의 실행으로 인해 코어가 명령어들을 능동적으로 실행하고 있지 않을 때 게이팅될 수 있고; 각각의 코어는 독립적으로 전력 게이팅될 수 있고; 모든 코어들이 클록 게이팅 또는 전력 게이팅될 때 각각의 코어 클러스터는 독립적으로 클록 게이팅될 수 있고; 및/또는 각각의 코어 클러스터는 모든 코어들이 전력 게이팅될 때 독립적으로 전력 게이팅될 수 있다. 적어도 하나의 실시예에서, CPU(들)(3206)는 전력 상태를 관리하기 위한 향상된 알고리즘을 더 구현할 수 있고, 여기서, 허용된 전력 상태와 예상된 각성 시간이 명시되며, 하드웨어/마이크로코드는 코어, 클러스터, 및 CPLEX에 진입할 최상의 전력 상태를 결정한다. 적어도 하나의 실시예에서, 처리 코어들은 마이크로코드에 작업을 오프로드하여 소프트웨어에서의 단순화된 전력 상태 진입 시퀀스들을 지원할 수 있다.

[0161] 적어도 하나의 실시예에서, GPU(들)(3208)는 통합 GPU(본 명세서에서 대안적으로 "iGPU"로 지칭됨)를 포함할 수 있다. 적어도 하나의 실시예에서, GPU(들)(3208)는 프로그램가능할 수 있고 병렬 작업부하들에 대해 효율적일 수 있다. 적어도 하나의 실시예에서, GPU(들)(3208)는, 적어도 하나의 실시예에서, 강화된 텐서 명령어 세트를

사용할 수 있다. 하나의 실시예에서, GPU(들)(3208)는 하나 이상의 스트리밍 마이크로프로세서를 포함할 수 있고, 여기서, 각각의 스트리밍 마이크로프로세서는 레벨 1("L1") 캐시(예를 들어, 적어도 96KB 저장 용량을 갖는 L1 캐시)를 포함할 수 있고, 스트리밍 마이크로프로세서들 중 2개 이상은 L2 캐시(예를 들어, 512KB 저장 용량을 갖는 L2 캐시)를 공유할 수 있다. 적어도 하나의 실시예에서, GPU(들)(3208)는 적어도 8개의 스트리밍 마이크로프로세서를 포함할 수 있다. 적어도 하나의 실시예에서, GPU(들)(3208)는 계산 애플리케이션 프로그래밍 인터페이스(들)(API(들))를 사용할 수 있다. 적어도 하나의 실시예에서, GPU(들)(3208)는 하나 이상의 병렬 계산 플랫폼 및/또는 프로그래밍 모델(예를 들어, NVIDIA의 CUDA)을 사용할 수 있다.

[0162] 적어도 하나의 실시예에서, GPU(들)(3208) 중 하나 이상은 자동차 및 임베디드 사용 사례들에서 최상의 성능을 위해 전력 최적화될 수 있다. 예를 들어, 하나의 실시예에서, GPU(들)(3208)는 핀 전계 효과 트랜지스터("FinFET") 상에 제조될 수 있다. 적어도 하나의 실시예에서, 각각의 스트리밍 마이크로프로세서는 다수의 블록들로 파티셔닝된 다수의 혼합 정밀도 처리 코어들을 통합할 수 있다. 예를 들어, 제한없이, 64개의 PF32 코어 및 32개의 PF64 코어는 4개의 처리 블록으로 파티셔닝될 수 있다. 적어도 하나의 실시예에서, 각각의 처리 블록은 16개의 FP32 코어, 8개의 FP64 코어, 16개의 INT32 코어, 딥 러닝 행렬 산술을 위한 2개의 혼합 정밀도 NVIDIA TENSOR CORE, 레벨 제로("L0") 명령어 캐시, 워프 스케줄러, 디스패치 유닛, 및/또는 64 KB 레지스터 파일을 할당받을 수 있다. 적어도 하나의 실시예에서, 스트리밍 마이크로프로세서들은 계산 및 어드레싱 산출들의 혼합으로 작업부하들의 효율적인 실행을 제공하기 위해 독립적인 병렬 정수 및 부동 소수점 데이터 경로들을 포함할 수 있다. 적어도 하나의 실시예에서, 스트리밍 마이크로프로세서들은 병렬 스레드들 사이의 더 미세한 입자의 동기화 및 협력을 가능하게 하는 독립적인 스레드 스케줄링 능력을 포함할 수 있다. 적어도 하나의 실시예에서, 스트리밍 마이크로프로세서들은 프로그래밍을 단순화하면서 성능을 개선하기 위해 조합된 L1 데이터 캐시 및 공유된 메모리 유닛을 포함할 수 있다.

[0163] 적어도 하나의 실시예에서, GPU(들)(3208) 중 하나 이상은, 일부 예들에서, 약 900 GB/초 피크 메모리 대역폭을 제공하기 위해 고대역폭 메모리("HBM") 및/또는 16 GB HBM2 메모리 서브시스템을 포함할 수 있다. 적어도 하나의 실시예에서, HBM 메모리에 추가로 또는 대안적으로, 그래픽 더블 데이터 레이트 타입 5 동기식 랜덤 액세스 메모리("GDDR5")와 같은 동기식 그래픽 랜덤 액세스 메모리("SGRAM")가 사용될 수 있다.

[0164] 적어도 하나의 실시예에서, GPU(들)(3208)는 통합 메모리 기술을 포함할 수 있다. 적어도 하나의 실시예에서, 어드레스 변환 서비스들("ATS") 지원은 GPU(들)(3208)가 CPU(들)(3206) 페이지 테이블들에 직접 액세스하는 것을 허용하기 위해 사용될 수 있다. 적어도 하나의 실시예에서, GPU(들)(3208) 메모리 관리 유닛("MMU")에 누락(miss)이 발생할 때, 어드레스 변환 요청이 CPU(들)(3206)로 송신될 수 있다. 이에 응답하여, 적어도 하나의 실시예에서, CPU(들)(3206)는 어드레스에 대한 가상-물리 맵핑을 위해 그의 페이지 테이블들을 조사할 수 있고 변환을 다시 GPU(들)(3208)로 송신한다. 적어도 하나의 실시예에서, 통합된 메모리 기술은 CPU(들)(3206) 및 GPU(들)(3208) 양자 모두의 메모리에 대한 단일의 통합된 가상 어드레스 공간을 허용할 수 있고, 이에 의해 GPU(들)(3208) 프로그래밍 및 GPU(들)(3208)에 대한 애플리케이션들의 포팅을 단순화시킨다.

[0165] 적어도 하나의 실시예에서, GPU(들)(3208)는 다른 프로세서들의 메모리에 대한 GPU(들)(3208)의 액세스 빈도의 추적을 유지할 수 있는 임의의 수의 액세스 카운터들을 포함할 수 있다. 적어도 하나의 실시예에서, 액세스 카운터(들)는 메모리 페이지들이 가장 빈번하게 페이지들에 액세스하고 있는 프로세서의 물리적 메모리로 이동되는 것을 보장하는 것을 도울 수 있으며, 이에 의해 프로세서들 사이에 공유되는 메모리 범위들에 대한 효율을 개선한다.

[0166] 적어도 하나의 실시예에서, SoC(들)(3204) 중 하나 이상은 본 명세서에 설명된 것들을 포함하는 임의의 수의 캐시(들)(3212)를 포함할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 캐시(들)(3212)는 CPU(들)(3206) 및 GPU(들)(3208) 양자 모두에 이용 가능한(예를 들어, CPU(들)(3206) 및 GPU(들)(3208) 양자 모두에 접속되는) 레벨 3("L3") 캐시를 포함할 수 있다. 적어도 하나의 실시예에서, 캐시(들)(3212)는 예컨대, 캐시 일관성 프로토콜(예를 들어, MEI, MESI, MSI 등)을 사용함으로써 라인들의 상태들을 계속 추적을 유지할 수 있는 후기입 캐시를 포함할 수 있다. 적어도 하나의 실시예에서, L3 캐시는 실시예에 따라 4 MB 이상을 포함할 수 있지만, 더 작은 캐시 크기들이 사용될 수 있다.

[0167] 적어도 하나의 실시예에서, SoC(들)(3204) 중 하나 이상은 하나 이상의 가속기(들)(3214)(예를 들어, 하드웨어 가속기들, 소프트웨어 가속기들, 또는 이들의 조합)를 포함할 수 있다. 적어도 하나의 실시예에서, SoC(들)(3204)는 최적화된 하드웨어 가속기들 및/또는 대형 온-칩 메모리를 포함할 수 있는 하드웨어 가속 클러스터를 포함할 수 있다. 적어도 하나의 실시예에서, 대형 온-칩 메모리(예를 들어, SRAM의 4MB)는 하드웨어 가속

클러스터가 신경망들 및 다른 산출들을 가속시킬 수 있게 할 수 있다. 적어도 하나의 실시예에서, 하드웨어 가속 클러스터는 GPU(들)(3208)를 보완하고 GPU(들)(3208)의 태스크들 중 일부를 오프로드(off-load)하기 위해(예를 들어, 다른 태스크들을 수행하기 위해 GPU(들)(3208)의 더 많은 사이클들을 해제하기 위해) 사용될 수 있다. 적어도 하나의 실시예에서, 가속기(들)(3214)는 가속에 순응할 만큼 충분히 안정적인 타겟화된 작업부하들(예를 들어, 인지, 컨볼루션 신경망들("CNN들"), 순환 신경망들("RNN들") 등)에 사용될 수 있다. 적어도 하나의 실시예에서, CNN은 영역-기반 또는 영역 컨볼루션 신경망들("RCNN들") 및 고속 RCNN들(예를 들어, 객체 검출을 위해 사용됨) 또는 다른 타입의 CNN을 포함할 수 있다.

[0168] 적어도 하나의 실시예에서, 가속기(들)(3214)(예를 들어, 하드웨어 가속 클러스터)는 딥 러닝 가속기(들)("DLA")를 포함할 수 있다. DLA(들)는 딥 러닝 애플리케이션들 및 추론을 위해 초당 추가적인 10조 연산들을 제공하도록 구성될 수 있는 하나 이상의 텐서 처리 유닛("TPU")을 포함할 수 있지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, TPU들은(예를 들어, CNN들, RCNN들 등에 대한) 이미지 처리 기능들을 수행하도록 구성되고 그에 대해 최적화된 가속기들일 수 있다. DLA(들)는 추론뿐만 아니라, 특정 세트의 신경망 타입들 및 부동 소수점 연산들에 대해 더 최적화될 수 있다. 적어도 하나의 실시예에서, DLA(들)의 설계는 전형적인 범용 GPU보다 밀리미터당 더 많은 성능을 제공할 수 있으며, 통상적으로 CPU의 성능을 크게 초과한다. 적어도 하나의 실시예에서, TPU(들)는, 예를 들어, 피쳐들 및 가중치들 양자 모두에 대한 INT8, INT16, 및 FP16 데이터 타입들은 물론, 포스트-프로세서 기능들을 지원하는, 단일-인스턴스 컨볼루션 기능을 비롯한, 여러 기능들을 수행할 수 있다. 적어도 하나의 실시예에서, DLA(들)는, 예를 들어 그리고 제한없이 다음을 포함하는 다양한 기능들 중 임의의 것에 대한 처리된 또는 처리되지 않은 데이터에 대해, 신경망들, 특히 CNN들을 신속하고 효율적으로 실행할 수 있다: 카메라 센서들로부터의 데이터를 사용하여 객체 식별 및 검출을 위한 CNN; 카메라 센서들로부터의 데이터를 사용하는 거리 추정을 위한 CNN; 마이크로폰들(3296)로부터의 데이터를 사용하는 음성 차량 검출 및 식별 및 검출을 위한 CNN; 카메라 센서들로부터의 데이터를 사용하여 얼굴 인식 및 차량 소유자 식별을 위한 CNN; 및/또는 보안 및/또는 안전 관련 이벤트들에 대한 CNN.

[0169] 적어도 하나의 실시예에서, DLA(들)는 GPU(들)(3208) 중 임의의 기능을 수행할 수 있고, 예를 들어, 추론 가속기를 사용함으로써, 설계자는 임의의 기능에 대해 DLA(들) 또는 GPU(들)(3208) 중 어느 하나를 타겟으로 할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 설계자는 CNN들 및 부동 소수점 연산들의 처리를 DLA(들)에 집중시키고 다른 기능들을 GPU(들)(3208) 및/또는 다른 가속기(들)(3214)에 남겨둘 수 있다.

[0170] 적어도 하나의 실시예에서, 가속기(들)(3214)(예를 들어, 하드웨어 가속 클러스터)는, 본 명세서에서 컴퓨터 비전 가속기라고 대안적으로 지칭될 수 있는, 프로그램가능 비전 가속기(들)(programmable vision accelerator)("PVA")를 포함할 수 있다. 적어도 하나의 실시예에서, PVA(들)는 고급 운전자 보조 시스템("ADAS")(3238), 자율 주행, 증강 현실("AR") 애플리케이션들, 및/또는 가상 현실("VR") 애플리케이션들을 위한 컴퓨터 비전 알고리즘들을 가속하도록 설계되고 구성될 수 있다. PVA(들)는 성능과 유연성 사이의 균형을 제공할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 각각의 PVA(들)는, 예를 들어, 제한없이, 임의의 수의 RISC(reduced instruction set computer) 코어, DMA(direct memory access), 및/또는 임의의 수의 벡터 프로세서를 포함할 수 있다.

[0171] 적어도 하나의 실시예에서, RISC 코어들은 이미지 센서들(예를 들어, 본 명세서에 설명된 카메라들 중 임의의 것의 이미지 센서들), 이미지 신호 프로세서(들) 등과 상호작용할 수 있다. 적어도 하나의 실시예에서, RISC 코어들 각각은 임의의 양의 메모리를 포함할 수 있다. 적어도 하나의 실시예에서, RISC 코어들은 실시예에 따라 다수의 프로토콜 중 임의의 것을 이용할 수 있다. 적어도 하나의 실시예에서, RISC 코어들은 실시간 운영 체제("RTOS")를 실행할 수 있다. 적어도 하나의 실시예에서, RISC 코어들은 하나 이상의 집적 회로 디바이스, 주문형 집적 회로("ASIC") 및/또는 메모리 디바이스를 이용하여 구현될 수 있다. 예를 들어, 적어도 하나의 실시예에서, RISC 코어들은 명령어 캐시 및/또는 밀접하게 결합된 RAM을 포함할 수 있다.

[0172] 적어도 하나의 실시예에서, DMA는 PVA(들)의 컴포넌트들이 CPU(들)(3206)와 독립적으로 시스템 메모리에 액세스하는 것을 가능하게 할 수 있다. 적어도 하나의 실시예에서, DMA는 다차원 어드레싱 및/또는 순환 어드레싱의 지원을 포함하지만 이에 제한되지 않는, PVA에 대한 최적화를 제공하는 데 사용되는 임의의 수의 피쳐들을 지원할 수 있다. 적어도 하나의 실시예에서, DMA는 블록 폭, 블록 높이, 블록 깊이, 수평 블록 스텝핑, 수직 블록 스텝핑 및/또는 깊이 스텝핑을 포함할 수 있지만 이에 제한되지 않는 최대 6개 이상의 어드레싱 차원을 지원할 수 있다.

[0173] 적어도 하나의 실시예에서, 벡터 프로세서들은 컴퓨터 비전 알고리즘들을 위한 프로그래밍을 효율적이고 유연하

게 실행하고 신호 처리 능력들을 제공하도록 설계될 수 있는 프로그램가능 프로세서들일 수 있다. 적어도 하나의 실시예에서, PVA는 PVA 코어 및 2개의 벡터 처리 서브시스템 파티션을 포함할 수 있다. 적어도 하나의 실시예에서, PVA 코어는 프로세서 서브시스템, DMA 엔진(들)(예를 들어, 2개의 DMA 엔진들), 및/또는 다른 주변기기들을 포함할 수 있다. 적어도 하나의 실시예에서, 벡터 처리 서브시스템은 PVA의 주 처리 엔진으로서 동작할 수 있으며, 벡터 처리 유닛("VPU"), 명령어 캐시 및/또는 벡터 메모리(예를 들어, "VMEM")를 포함할 수 있다. 적어도 하나의 실시예에서, VPU 코어는, 예를 들어, SIMD(single instruction, multiple data), VLIW(very long instruction word) 디지털 신호 프로세서와 같은 디지털 신호 프로세서를 포함할 수 있다. 적어도 하나의 실시예에서, SIMD와 VLIW의 조합은 처리량 및 속도를 향상시킬 수 있다.

[0174] 적어도 하나의 실시예에서, 벡터 프로세서들 각각은 명령어 캐시를 포함할 수 있고 전용 메모리에 결합될 수 있다. 그 결과, 적어도 하나의 실시예에서, 각각의 벡터 프로세서는 다른 벡터 프로세서와 독립적으로 실행되도록 구성될 수 있다. 적어도 하나의 실시예에서, 특정 PVA에 포함되는 벡터 프로세서들은 데이터 병렬성을 채용하도록 구성될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 단일 PVA에 포함된 복수의 벡터 프로세서는 동일한 컴퓨터 비전 알고리즘을, 그러나, 이미지의 상이한 영역들에 대해 실행할 수 있다. 적어도 하나의 실시예에서, 특정 PVA에 포함된 벡터 프로세서들은 동일한 이미지에 대해 상이한 컴퓨터 비전 알고리즘들을 동시에 실행하거나, 또는 심지어 순차적인 이미지들 또는 이미지의 부분들에 대해 상이한 알고리즘들을 실행할 수 있다. 적어도 하나의 실시예에서, 그 중에서도 특히, 임의의 수의 PVA가 하드웨어 가속 클러스터에 포함될 수 있고, 임의의 수의 벡터 프로세서가 각각의 PVA에 포함될 수 있다. 적어도 하나의 실시예에서, PVA(들)는 전체 시스템 안전을 향상시키기 위해 추가적인 에러 정정 코드("ECC") 메모리를 포함할 수 있다.

[0175] 적어도 하나의 실시예에서, 가속기(들)(3214)(예를 들어, 하드웨어 가속 클러스터)는 가속기(들)(3214)에 대해 고대역폭, 저레이턴시 SRAM을 제공하기 위해 컴퓨터 비전 네트워크 온-칩 및 정적 랜덤 액세스 메모리("SRAM")를 포함할 수 있다. 적어도 하나의 실시예에서, 온-칩 메모리는, 예를 들어, 제한없이, PVA와 DLA 양자 모두에 의해 액세스가능할 수 있는 8개의 필드-구성가능한 메모리 블록으로 구성된 적어도 4MB SRAM을 포함할 수 있다. 적어도 하나의 실시예에서, 각각의 메모리 블록 쌍은 APB(advanced peripheral bus) 인터페이스, 구성 회로, 제어기, 및 멀티플렉서를 포함할 수 있다. 적어도 하나의 실시예에서, 임의의 타입의 메모리가 사용될 수 있다. 적어도 하나의 실시예에서, PVA 및 DLA는 PVA 및 DLA에 메모리에 대한 고속 액세스를 제공하는 백본을 통해 메모리에 액세스할 수 있다. 적어도 하나의 실시예에서, 백본은 (예를 들어, APB를 사용하여) PVA 및 DLA를 메모리에 상호접속시키는 컴퓨터 비전 네트워크 온-칩을 포함할 수 있다.

[0176] 적어도 하나의 실시예에서, 온-칩 컴퓨터 비전 네트워크는 임의의 제어 신호/어드레스/데이터의 송신 전에, PVA 및 DLA 양자 모두가 준비(ready) 및 유효(valid) 신호들을 제공하는 것을 결정하는 인터페이스를 포함할 수 있다. 적어도 하나의 실시예에서, 인터페이스는 제어 신호들/어드레스들/데이터를 송신하기 위한 별도의 위상들 및 별도의 채널들, 뿐만 아니라 연속적인 데이터 전송을 위한 버스트-타입 통신들을 제공할 수 있다. 적어도 하나의 실시예에서, 인터페이스는 국제 표준화 기구("ISO") 26262 또는 국제 전기기술 위원회("IEC") 61508 표준들을 따를 수 있지만, 다른 표준들 및 프로토콜들이 사용될 수 있다.

[0177] 적어도 하나의 실시예에서, SoC(들)(3204) 중 하나 이상은 실시간 광선 추적 하드웨어 가속기를 포함할 수 있다. 적어도 하나의 실시예에서, 실시간 광선 추적 하드웨어 가속기는 (예를 들어, 월드 모델 내의) 객체들의 위치들 및 범위들을 신속하고 효율적으로 결정하기 위해, 실시간 시각화 시뮬레이션들을 생성하기 위해, RADAR 신호 해석을 위해, 음향 전파 합성 및/또는 분석을 위해, SONAR 시스템들의 시뮬레이션을 위해, 일반 파 전파 시뮬레이션을 위해, 위치확인 및/또는 다른 기능들의 목적들에 대해 LIDAR 데이터와 비교하기 위해, 및/또는 다른 용도들을 위해 사용될 수 있다.

[0178] 적어도 하나의 실시예에서, 가속기(들)(3214)(예를 들어, 하드웨어 가속기 클러스터)는 자율 주행을 위해 광범위하게 사용된다. 적어도 하나의 실시예에서, PVA는 ADAS 및 자율 차량들에서의 핵심 처리 스테이지들을 위해 사용될 수 있는 프로그램가능 비전 가속기일 수 있다. 적어도 하나의 실시예에서, PVA의 능력들은 낮은 전력 및 낮은 레이턴시에서, 예측가능한 처리를 필요로 하는 알고리즘 도메인들에 양호하게 일치한다. 다시 말해, PVA는 낮은 레이턴시 및 낮은 전력으로 예측가능한 런-타임들을 필요로 하는 반-밀집 또는 밀집 정규 계산에 대해, 심지어 작은 데이터 세트들에 대해서도 잘 기능한다. 적어도 하나의 실시예에서, 차량(3200)과 같은 자율 차량들, PVA들은 고전적인 컴퓨터 비전 알고리즘들을 실행하도록 설계되고, 그 이유는 이들이 객체 검출에서 효율적이고 정수 수학에서 연산하기 때문이다.

[0179] 예를 들어, 기술의 적어도 하나의 실시예에 따르면, PVA는 컴퓨터 스테레오 비전을 수행하는 데 사용된다. 적어

도 하나의 실시예에서, 세미-글로벌 매칭 기반 알고리즘이 일부 예들에서 사용될 수 있지만, 그러나, 이는 제한적인 것으로 의도되지 않는다. 적어도 하나의 실시예에서, 레벨 3-5 자율 주행을 위한 애플리케이션들은 온-더-플라이(on-the-fly) 모션 추정/스테레오 매칭(예를 들어, 모션으로부터의 구조, 보행자 인식, 차선 검출 등)을 사용한다. 적어도 하나의 실시예에서, PVA는 2개의 단안 카메라들로부터의 입력들에 대해 컴퓨터 스테레오 비전 기능을 수행할 수 있다.

[0180] 적어도 하나의 실시예에서, PVA는 밀집 광학 흐름을 수행하는 데 사용될 수 있다. 예를 들어, 적어도 하나의 실시예에서, PVA는 (예를 들어, 4D 고속 푸리에 변환을 사용하여) 원시 RADAR 데이터를 처리하여 처리된 RADAR 데이터를 제공할 수 있다. 적어도 하나의 실시예에서, PVA는 예를 들어 처리된 비행 시간 데이터를 제공하기 위해 원시 비행 시간 데이터를 처리함으로써 비행 시간 깊이 처리를 위해 사용된다.

[0181] 적어도 하나의 실시예에서, DLA는, 예를 들어 및 제한없이, 각각의 객체 검출에 대한 신뢰도 척도를 출력하는 신경망을 포함한, 제어 및 운전 안전성을 향상시키기 위해 임의의 타입의 네트워크를 실행하는데 이용될 수 있다. 적어도 하나의 실시예에서, 신뢰도는 확률로서, 또는 다른 검출들에 비교하여 각각의 검출의 상대적 "가중치"를 제공하는 것으로서 표현되거나 해석될 수 있다. 적어도 하나의 실시예에서, 신뢰도는 시스템이 어느 검출들이 위양성 검출들이 아니라 진양성 검출들로서 고려되어야 하는지에 관한 추가 판정들을 행하는 것을 가능하게 한다. 예를 들어, 적어도 하나의 실시예에서, 시스템은 신뢰도에 대한 임계값을 설정하고 임계값을 초과하는 검출들만을 진양성 검출들로서 고려할 수 있다. 자동 비상 제동("AEB") 시스템이 사용되는 실시예에서, 위양성 검출들은 차량이 비상 제동을 자동으로 수행하게 할 것이고, 이는 분명히 바람직하지 않다. 적어도 하나의 실시예에서, 매우 신뢰된 검출들은 AEB에 대한 트리거들로서 고려될 수 있다. 적어도 하나의 실시예에서, DLA는 신뢰도를 회귀시키기 위해 신경망을 실행할 수 있다. 적어도 하나의 실시예에서, 신경망은, 그 입력으로서, 특히, 경계 박스 치수, (예를 들어, 또 다른 서브시스템으로부터) 획득된 지상 평면 추정, 차량(3200) 배향과 상관되는 IMU 센서(들)(3266)로부터의 출력, 거리, 신경망으로부터 획득된 객체의 3D 위치 추정 및/또는 다른 센서(예를 들어, LIDAR 센서(들)(3264) 또는 RADAR 센서(들)(3260)) 같은, 적어도 일부 서브세트의 파라미터들을 취할 수 있다.

[0182] 적어도 하나의 실시예에서, SoC(들)(3204) 중 하나 이상은 데이터 저장소(들)(3216)(예를 들어, 메모리)를 포함할 수 있다. 적어도 하나의 실시예에서, 데이터 저장소(들)(3216)는 GPU(들)(3208) 및/또는 DLA 상에서 실행될 신경망들을 저장할 수 있는 SoC(들)(3204)의 온-칩 메모리일 수 있다. 적어도 하나의 실시예에서, 데이터 저장소(들)(3216)는 중복성 및 안전성을 위해 신경망들의 다수의 인스턴스들을 저장할 정도로 용량이 충분히 클 수 있다. 적어도 하나의 실시예에서, 데이터 저장소(들)(3212)는 L2 또는 L3 캐시(들)를 포함할 수 있다.

[0183] 적어도 하나의 실시예에서, SoC(들)(3204) 중 하나 이상은 임의의 수의 프로세서(들)(3210)(예를 들어, 임베디드 프로세서들)를 포함할 수 있다. 프로세서(들)(3210)는 부트 전력 및 관리 기능들 및 관련 보안 시행을 핸들링하기 위한 전용 프로세서 및 서브시스템일 수 있는 부트 및 전력 관리 프로세서를 포함할 수 있다. 적어도 하나의 실시예에서, 부트 및 전력 관리 프로세서는 SoC(들)(3204) 부트 시퀀스의 일부일 수 있고 런타임 전력 관리 서비스들을 제공할 수 있다. 적어도 하나의 실시예에서, 부트 전력 및 관리 프로세서는 클록 및 전압 프로그래밍, 시스템 저전력 상태 전이들에서의 보조, SoC(들)(3204) 열들 및 온도 센서들의 관리, 및/또는 SoC(들)(3204) 전력 상태들의 관리를 제공할 수 있다. 적어도 하나의 실시예에서, 각각의 온도 센서는 그 출력 주파수가 온도에 비례하는 링 발진기로서 구현될 수 있고, SoC(들)(3204)는 CPU(들)(3206), GPU(들)(3208), 및/또는 가속기(들)(3214)의 온도들을 검출하기 위해 링 발진기들을 사용할 수 있다. 적어도 하나의 실시예에서, 온도들이 임계값을 초과하는 것으로 결정되면, 이때, 부트 및 전력 관리 프로세서는 온도 결함 루틴에 진입하고 SoC(들)(3204)를 저전력 상태가 되게 하고 및/또는 차량(3200)을 안전 정지 모드로의 쇼퍼(chauffeur)에 배치할 수 있다(예를 들어, 차량(3200)을 안전 정지되게 함).

[0184] 적어도 하나의 실시예에서, 프로세서(들)(3210)는 오디오 처리 엔진으로서 역할할 수 있는 임베디드 프로세서 세트를 더 포함할 수 있다. 적어도 하나의 실시예에서, 오디오 처리 엔진은 다수의 인터페이스들을 통한 멀티-채널 오디오에 대한 완전한 하드웨어 지원, 및 넓고 유연한 범위의 오디오 I/O 인터페이스들을 가능하게 하는 오디오 서브시스템일 수 있다. 적어도 하나의 실시예에서, 오디오 처리 엔진은 전용 RAM을 갖는 디지털 신호 프로세서를 갖는 전용 프로세서 코어이다.

[0185] 적어도 하나의 실시예에서, 프로세서(들)(3210)는 저전력 센서 관리 및 웨이크 사용 사례들을 지원하기 위해 필요한 하드웨어 피쳐들을 제공할 수 있는 상시동작(always on) 프로세서 엔진을 더 포함할 수 있다. 적어도 하나의 실시예에서, 상시동작(always on) 프로세서 엔진은, 제한없이, 프로세서 코어, 밀접하게 결합된 RAM, 지원

주변기기들(예를 들어, 타이머들 및 인터럽트 제어기들), 다양한 I/O 제어기 주변기기들, 및 라우팅 로직을 포함할 수 있다.

[0186] 적어도 하나의 실시예에서, 프로세서(들)(3210)는 자동차 응용들에 대한 안전 관리를 핸들링하는 전용 프로세서 서브시스템을, 제한없이, 포함하는 안전 클러스터 엔진을 더 포함할 수 있다. 적어도 하나의 실시예에서, 안전 클러스터 엔진은, 제한없이, 2개 이상의 프로세서 코어들, 밀접하게 결합된 RAM, 지원 주변기기들(예를 들어, 타이머들, 인터럽트 제어기 등), 및/또는 라우팅 로직을 포함할 수 있다. 안전 모드에서, 2개 이상의 코어들은, 적어도 하나의 실시예에서, 락스텝 모드에서 동작할 수 있고, 그들의 동작들 사이의 임의의 차이들을 검출하는 비교 로직을 갖는 단일 코어로서 기능할 수 있다. 적어도 하나의 실시예에서, 프로세서(들)(3210)는 실시간 카메라 관리를 처리하는 전용 프로세서 서브시스템(이들로 제한되지 않음)을 포함할 수 있는 실시간 카메라 엔진을 더 포함할 수 있다. 적어도 하나의 실시예에서, 프로세서(들)(3210)는, 제한없이, 카메라 처리 파이프라인의 일부인 하드웨어 엔진인 이미지 신호 프로세서를 포함할 수 있는 하이 다이내믹 레인지 신호 프로세서를 더 포함할 수 있다.

[0187] 적어도 하나의 실시예에서, 프로세서(들)(3210)는 플레이어 윈도우에 대한 최종 이미지를 생성하기 위해 비디오 재생 애플리케이션에 의해 요구되는 비디오 후처리 기능들을 구현하는 (예를 들어, 마이크로프로세서 상에 구현되는) 처리 블록일 수 있는 비디오 이미지 합성기를 포함할 수 있다. 적어도 하나의 실시예에서, 비디오 이미지 합성기는 와이드-뷰 카메라(들)(3270), 서라운드 카메라(들)(3274), 및/또는 객실 내 모니터링 카메라 센서(들)에 대해 렌즈 왜곡 정정을 수행할 수 있다. 적어도 하나의 실시예에서, 객실 내 모니터링 카메라 센서(들)는 바람직하게는, 객실 내 이벤트들을 식별하고 그에 따라 응답하도록 구성된, SoC(3204)의 다른 인스턴스 상에서 실행되는 신경망에 의해 모니터링된다. 적어도 하나의 실시예에서, 객실 내 시스템은 셀룰러 서비스를 활성화하고 전화를 걸거나, 이메일을 지시하거나, 차량의 목적지를 변경하거나, 차량의 인포테인먼트 시스템 및 설정들을 활성화 또는 변경하거나, 음성 활성화 웹 서핑을 제공하기 위해 입술 판독을 수행할 수 있지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, 특정 기능들은 차량이 자율 모드에서 동작하고 있을 때 운전자에 의해 사용될 수 있고, 그렇지 않으면 불능화된다.

[0188] 적어도 하나의 실시예에서, 비디오 이미지 합성기는 공간 및 시간 노이즈 감소 양자 모두를 위한 향상된 시간 노이즈 감소를 포함할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 모션이 비디오에서 발생하는 경우, 노이즈 감소는 공간 정보를 적절히 가중하여, 인접 프레임들에 의해 제공되는 정보의 가중치를 감소시킨다. 이미지 또는 이미지의 일부가 모션을 포함하지 않는 적어도 하나의 실시예에서, 비디오 이미지 합성기에 의해 수행되는 시간 노이즈 감소는 이전 이미지로부터의 정보를 이용하여 현재 이미지 내의 노이즈를 감소시킬 수 있다.

[0189] 적어도 하나의 실시예에서, 비디오 이미지 합성기는 또한 입력 스테레오 렌즈 프레임들에 대해 스테레오 정류를 수행하도록 구성될 수 있다. 적어도 하나의 실시예에서, 비디오 이미지 합성기는 운영 체제 데스크탑이 사용 중일 때 사용자 인터페이스 합성에 더 사용될 수 있고, GPU(들)(3208)는 새로운 표면들을 연속적으로 렌더링하도록 요구되지 않는다. 적어도 하나의 실시예에서, GPU(들)(3208)의 전원이 켜지고 3D 렌더링을 능동적으로 수행할 때, GPU(들)(3208)를 오프로드하여 성능 및 응답성을 향상시키기 위해 비디오 이미지 합성기가 사용될 수 있다.

[0190] 적어도 하나의 실시예에서, SoC(들)(3204) 중 하나 이상은 카메라들로부터 비디오 및 입력을 수신하기 위한 모바일 산업 프로세서 인터페이스("MIPI") 카메라 직렬 인터페이스, 고속 인터페이스, 및/또는 카메라 및 관련 픽셀 입력 기능들을 위해 사용될 수 있는 비디오 입력 블록을 더 포함할 수 있다. 적어도 하나의 실시예에서, SoC(들)(3204) 중 하나 이상은 소프트웨어에 의해 제어될 수 있고 특정 역할에 커밋되지 않은 I/O 신호들을 수신하기 위해 사용될 수 있는 입력/출력 제어기(들)를 더 포함할 수 있다.

[0191] 적어도 하나의 실시예에서, SoC(들)(3204) 중 하나 이상은 주변기기들, 오디오 인코더들/디코더들("코덱들"), 전력 관리, 및/또는 다른 디바이스들과의 통신을 가능하게 하기 위해 광범위한 주변 인터페이스들을 더 포함할 수 있다. SoC(들)(3204)는 카메라들로부터의 데이터(예를 들어, 기가비트 멀티미디어 직렬 링크 및 이더넷을 통해 접속됨), 센서들(예를 들어, 이더넷을 통해 접속될 수 있는 LIDAR 센서(들)(3264), RADAR 센서(들)(3260) 등), 버스(3202)로부터의 데이터(예를 들어, 차량(3200)의 속도, 조향 휠 위치 등), GNSS 센서(들)(3258)로부터의 데이터(예를 들어, 이더넷 또는 CAN 버스를 통해 접속됨) 등을 처리하기 위해 사용될 수 있다. 적어도 하나의 실시예에서, SoC(들)(3204) 중 하나 이상은 그 자신의 DMA 엔진들을 포함할 수 있는 전용 고성능 대용량 저장소 제어기들을 더 포함할 수 있고, 이는 일상적인 데이터 관리 태스크들로부터 CPU(들)(3206)를 자유롭게 하기 위해 사용될 수 있다.

- [0192] 적어도 하나의 실시예에서, SoC(들)(3204)는 자동화 레벨들 3 내지 5에 걸쳐 있는 유연한 아키텍처를 갖는 종단간 플랫폼일 수 있고, 이에 의해 다양성 및 중복성을 위해 컴퓨터 비전 및 ADAS 기법들을 활용하고 이들을 효율적으로 사용하며, 딥 러닝 도구들과 함께, 유연하고 신뢰성 있는 운전 소프트웨어 스택을 위한 플랫폼을 제공하는 포괄적인 기능적 안전 아키텍처를 제공한다. 적어도 하나의 실시예에서, SoC(들)(3204)는 종래의 시스템들보다 더 빠르고, 더 신뢰성 있고, 훨씬 더 에너지 효율적이고 공간 효율적일 수 있다. 예를 들어, 적어도 하나의 실시예에서, 가속기(들)(3214)는, CPU(들)(3206), GPU(들)(3208), 및 데이터 저장소(들)(3216)와 조합될 때, 레벨 3-5 자율 차량들에 대한 빠르고 효율적인 플랫폼을 제공할 수 있다.
- [0193] 적어도 하나의 실시예에서, 컴퓨터 비전 알고리즘들은 CPU들 상에서 실행될 수 있고, 이것은 C 프로그래밍 언어와 같은 하이-레벨 프로그래밍 언어를 이용하여 다양한 시각적 데이터에 걸쳐 다양한 처리 알고리즘들을 실행하도록 구성될 수 있다. 그러나, 적어도 하나의 실시예에서, CPU들은 종종, 예를 들어, 실행 시간 및 전력 소비에 관련된 것들과 같은, 많은 컴퓨터 비전 응용들의 성능 요건들을 충족시킬 수 없다. 적어도 하나의 실시예에서, 많은 CPU들은 복잡한 객체 검출 알고리즘들을 실시간으로 실행할 수 없고, 이는 차량내 ADAS 응용들에서 그리고 실용적 레벨 3-5 자율 차량들에서 사용된다.
- [0194] 본 명세서에 설명된 실시예들은 다수의 신경망이 동시에 및/또는 순차적으로 수행될 수 있게 하고, 결과들이 레벨 3-5 자율 주행 기능을 가능하게 하기 위해 함께 조합될 수 있게 한다. 예를 들어, 적어도 하나의 실시예에서, DLA 또는 개별 GPU(예를 들어, GPU(들)(3220)) 상에서 실행되는 CNN은 텍스트 및 워드 인식을 포함할 수 있어, 슈퍼컴퓨터가, 신경망이 특별히 훈련되지 않은 표지판들을 포함한, 트래픽 표지판들을 판독하고 이해할 수 있게 한다. 적어도 하나의 실시예에서, DLA는 표지판의 의미론적 이해를 식별, 해석 및 제공하고, 그 의미론적 이해를 CPU 컴플렉스 상에서 실행되는 경로 계획 모듈들에 전달할 수 있는 신경망을 더 포함할 수 있다.
- [0195] 적어도 하나의 실시예에서, 예컨대, 레벨 3, 4, 또는 5 주행에 대해서 다수의 신경망들이 동시에 실행될 수 있다. 예를 들어, 적어도 하나의 실시예에서, "주의: 점멸등은 결빙 상태를 나타냄"으로 구성된 경고 표시는, 전기 조명과 함께, 여러 신경망에 의해 독립적으로 또는 집합적으로 해석될 수 있다. 적어도 하나의 실시예에서, 표지판 자체는 제1 배치된 신경망(예를 들어, 훈련된 신경망)에 의해 교통 표지판으로서 식별될 수 있고, 텍스트 "점멸등들은 결빙 상태들을 나타냄"은 제2 배치된 신경망에 의해 해석될 수 있고, 이는 점멸등이 검출될 때 결빙 상태가 존재한다는 것을 차량의 경로 계획 소프트웨어(바람직하게는 CPU 컴플렉스에서 실행됨)에 알려준다. 적어도 하나의 실시예에서, 점멸등은 점멸등의 존재(또는 부재)를 차량의 경로 계획 소프트웨어에 알리는, 다수의 프레임들에 걸쳐 제3 배치된 신경망을 동작시킴으로써 식별될 수 있다. 적어도 하나의 실시예에서, 3개 모두의 신경망은, DLA 내에서 및/또는 GPU(들)(3208) 상에서와 같이, 동시에 실행될 수 있다.
- [0196] 적어도 하나의 실시예에서, 얼굴 인식 및 차량 소유자 식별을 위한 CNN은 카메라 센서들로부터의 데이터를 사용하여 차량(3200)의 인가된 운전자 및/또는 소유자의 존재를 식별할 수 있다. 적어도 하나의 실시예에서, 상시동작(always on) 센서 처리 엔진은 소유자가 운전자 도어에 접근할 때 차량을 잠금해제하고 조명들을 켜고, 보안 모드에서, 소유자가 차량을 떠날 때 차량을 불능화시키기 위해 사용될 수 있다. 이러한 방식으로, SoC(들)(3204)는 도난 및/또는 카잭킹에 대한 보안을 제공한다.
- [0197] 적어도 하나의 실시예에서, 응급 차량 검출 및 식별을 위한 CNN은 응급 차량 사이렌들을 검출하고 식별하기 위해 마이크로폰들(3296)로부터의 데이터를 사용할 수 있다. 적어도 하나의 실시예에서, SoC(들)(3204)는 환경 및 도시 음향들을 분류할 뿐만 아니라 시각적 데이터를 분류하기 위해 CNN을 사용한다. 적어도 하나의 실시예에서, DLA 상에서 실행 중인 CNN은 (예를 들어, 도플러 효과를 사용함으로써) 응급 차량의 상대적인 근접 속도를 식별하도록 훈련된다. 적어도 하나의 실시예에서, CNN은 또한, GNSS 센서(들)(3258)에 의해 식별되는 바와 같이, 차량이 동작하고 있는 로컬 영역에 특정된 응급 차량들을 식별하도록 훈련될 수 있다. 적어도 하나의 실시예에서, 유럽에서 동작할 때, CNN은 유럽 사이렌들을 검출하려고 시도할 것이고, 미국 CNN에서, 북미 사이렌들만을 식별하려고 시도할 것이다. 적어도 하나의 실시예에서, 일단 응급 차량이 검출되면, 제어 프로그램은 응급 차량(들)이 통과할 때까지, 초음파 센서(들)(3262)의 보조로, 응급 차량 안전 루틴을 실행하고, 차량을 늦추고, 도로의 측면으로 정차하고, 차량을 주차하고, 및/또는 차량을 아이들링하기 위해 사용될 수 있다.
- [0198] 적어도 하나의 실시예에서, 차량(3200)은 고속 인터커넥트(예를 들어, PCIe)를 통해 SoC(들)(3204)에 결합될 수 있는 CPU(들)(3218)(예를 들어, 이산 CPU(들), 또는 dCPU(들))를 포함할 수 있다. 적어도 하나의 실시예에서, CPU(들)(3218)는, 예를 들어, X86 프로세서를 포함할 수 있다. CPU(들)(3218)는, 예를 들어, ADAS 센서들과 SoC(들)(3204) 사이의 잠재적으로 일관성 없는 결과들을 중재하는 것, 및/또는 제어기(들)(3236) 및/또는 인포

테인먼트 시스템-온-칩("인포테인먼트 SoC")(3230)의 상태 및 건강을 모니터링하는 것을 포함하는, 다양한 기능들 중 임의의 것을 수행하는 데 사용될 수 있다.

[0199] 적어도 하나의 실시예에서, 차량(3200)은 고속 인터커넥트(예를 들어, NVIDIA의 NVLINK)를 통해 SoC(들)(3204)에 결합될 수 있는 GPU(들)(3220)(예를 들어, 이산 GPU(들) 또는 dGPU(들))를 포함할 수 있다. 적어도 하나의 실시예에서, GPU(들)(3220)는 예컨대 중복 및/또는 상이한 신경망들을 실행하는 것에 의해, 추가적인 인공 지능 기능을 제공할 수 있고, 차량(3200)의 센서들로부터의 입력(예를 들어, 센서 데이터)에 적어도 부분적으로 기초하여 신경망들을 훈련 및/또는 업데이트하는 데 사용될 수 있다.

[0200] 적어도 하나의 실시예에서, 차량(3200)은 무선 안테나(들)(3226)(예를 들어, 셀룰러 안테나, 블루투스 안테나 등과 같은, 상이한 통신 프로토콜들에 대한 하나 이상의 무선 안테나(3226))를, 제한없이, 포함할 수 있는 네트워크 인터페이스(3224)를 더 포함할 수 있다. 적어도 하나의 실시예에서, 네트워크 인터페이스(3224)는 인터넷을 통한 클라우드와의(예를 들어, 서버(들) 및/또는 다른 네트워크 디바이스들과의), 다른 차량들과의, 및/또는 계산 디바이스들(예를 들어, 승객들의 클라이언트 디바이스들)과의 무선 접속성을 가능하게 하기 위해 사용될 수 있다. 적어도 하나의 실시예에서, 다른 차량들과 통신하기 위해, 차량(3200)과 다른 차량 사이에 직접 링크가 확립될 수 있고/있거나 간접 링크가 확립될 수 있다(예를 들어, 네트워크들을 통해 그리고 인터넷을 통해). 적어도 하나의 실시예에서, 직접 링크들은 차량 대 차량 통신 링크를 사용하여 제공될 수 있다. 차량 대 차량 통신 링크는 차량(3200)에 근접한 차량들(예를 들어, 차량(3200)의 전방, 측면, 및/또는 후방의 차량들)에 관한 차량(3200) 정보를 제공할 수 있다. 적어도 하나의 실시예에서, 전송한 기능은 차량(3200)의 협력 적응 크루즈 컨트롤 기능의 일부일 수 있다.

[0201] 적어도 하나의 실시예에서, 네트워크 인터페이스(3224)는 변조 및 복조 기능성을 제공하고 제어기(들)(3236)가 무선 네트워크들을 통해 통신할 수 있게 하는 SoC를 포함할 수 있다. 적어도 하나의 실시예에서, 네트워크 인터페이스(3224)는 기저대역으로부터 무선 주파수로의 상향 변환 및 무선 주파수로부터 기저대역으로의 하향 변환을 위한 무선 주파수 프론트엔드를 포함할 수 있다. 적어도 하나의 실시예에서, 주파수 변환들은 임의의 기술적으로 실현 가능한 방식으로 수행될 수 있다. 예를 들어, 주파수 변환들은 잘 알려진 프로세스들을 통해, 및/또는 슈퍼 헤테로다인 프로세스들을 이용하여 수행될 수 있다. 적어도 하나의 실시예에서, 무선 주파수 프론트엔드 기능은 별개의 칩에 의해 제공될 수 있다. 적어도 하나의 실시예에서, 네트워크 인터페이스는 LTE, WCDMA, UMTS, GSM, CDMA2000, 블루투스, 블루투스 LE, Wi-Fi, Z-Wave, ZigBee, LoRaWAN, 및/또는 다른 무선 프로토콜들을 통해 통신하기 위한 무선 기능성을 포함할 수 있다.

[0202] 적어도 하나의 실시예에서, 차량(3200)은, 제한없이, 오프-칩(예를 들어, 오프 SoC(들)(3204)) 저장소를 포함할 수 있는 데이터 저장소(들)(3228)를 더 포함할 수 있다. 적어도 하나의 실시예에서, 데이터 저장소(들)(3228)는 RAM, SRAM, DRAM(dynamic random-access memory), VRAM(video random-access memory), 플래시, 하드 디스크, 및/또는 적어도 하나의 비트의 데이터를 저장할 수 있는 다른 컴포넌트들 및/또는 디바이스들을 비롯한 하나 이상의 저장소 엘리먼트들을 포함할 수 있지만, 이에 제한되지 않는다.

[0203] 적어도 하나의 실시예에서, 차량(3200)은, 맵핑, 인지, 점유 그리드 생성, 및/또는 경로 계획 기능들을 보조하기 위해, GNSS 센서(들)(3258)(예를 들어, GPS 및/또는 보조 GPS 센서들)를 더 포함할 수 있다. 적어도 하나의 실시예에서, 예를 들어, 제한없이, 이더넷-직렬(예를 들어, RS-232) 브리지를 갖는 USB 커넥터를 사용하는 GPS를 포함하는 임의의 수의 GNSS 센서(들)(3258)가 사용될 수 있다.

[0204] 적어도 하나의 실시예에서, 차량(3200)은 RADAR 센서(들)(3260)를 더 포함할 수 있다. RADAR 센서(들)(3260)는 심지어 어둠 및/또는 심각한 기상 조건들에서도, 장거리 차량 검출을 위해 차량(3200)에 의해 사용될 수 있다. 적어도 하나의 실시예에서, RADAR 기능적 안전 레벨은 ASIL B일 수 있다. RADAR 센서(들)(3260)는, 일부 예들에서 원시 데이터에 액세스하기 위한 이더넷에 대한 액세스와 함께, 제어를 위해 그리고 객체 추적 데이터에 액세스하기 위해(예를 들어, RADAR 센서(들)(3260)에 의해 생성되는 데이터를 송신하기 위해) CAN 및/또는 버스(3202)를 사용할 수 있다. 적어도 하나의 실시예에서, 매우 다양한 RADAR 센서 타입들이 사용될 수 있다. 예를 들어, 그리고 제한없이, RADAR 센서(들)(3260)는 전방, 후방, 및 측면 RADAR 사용에 적합할 수 있다. 적어도 하나의 실시예에서, RADAR 센서들(들)(3260) 중 하나 이상은 펄스 도플러 RADAR 센서(들)이다.

[0205] 적어도 하나의 실시예에서, RADAR 센서(들)(3260)는 좁은 시야를 갖는 장거리, 넓은 시야를 갖는 단거리, 단거리 측면 커버리지 등과 같은 상이한 구성들을 포함할 수 있다. 적어도 하나의 실시예에서, 장거리 RADAR는 적응적 크루즈 컨트롤 기능을 위해 사용될 수 있다. 적어도 하나의 실시예에서, 장거리 RADAR 시스템들은 250m 범위 내에서와 같이, 2개 이상의 독립적인 스캔들에 의해 실현되는 넓은 시야를 제공할 수 있다. 적어도 하나의 실시

예에서, RADAR 센서(들)(3260)는 정적 객체와 움직이는 객체 사이를 구별하는 데 도움을 줄 수 있고, 비상 브레이크 보조 및 전방 충돌 경고를 위해 ADAS 시스템(3238)에 의해 사용될 수 있다. 장거리 RADAR 시스템에 포함된 센서들(3260(s))은, 제한없이, 다수의(예를 들어, 6개 이상) 고정 RADAR 안테나를 및 고속 CAN 및 FlexRay 인터페이스를 갖는 모노스테틱 멀티모드 RADAR을 포함할 수 있다. 6개의 안테나를 갖는 적어도 하나의 실시예에서, 중앙의 4개의 안테나는 인접 차선들에서의 트래픽으로부터의 최소 간섭으로 더 높은 속도로 차량(3200)의 주위를 기록하도록 설계된 집중된 빔 패턴을 생성할 수 있다. 적어도 하나의 실시예에서, 다른 2개의 안테나는 시야를 확장하여, 차량(3200)의 차선에 진입하거나 나가는 차량들을 신속하게 검출하는 것을 가능하게 할 수 있다.

[0206] 적어도 하나의 실시예에서, 중거리 RADAR 시스템들은, 예를 들어, 160m(전방) 또는 80m(후방)까지의 범위, 및 42도(전방) 또는 150도(후방)까지의 시야를 포함할 수 있다. 적어도 하나의 실시예에서, 단거리 RADAR 시스템들은 후방 범퍼의 양 단부에 설치되도록 설계된 임의의 수의 RADAR 센서(들)(3260)를 제한없이 포함할 수 있다. 후방 범퍼의 양 단부에 설치될 때, 적어도 하나의 실시예에서, RADAR 센서 시스템은 차량의 후방 및 옆에서 사각 지대를 지속적으로 모니터링하는 2개의 빔을 생성할 수 있다. 적어도 하나의 실시예에서, 단거리 RADAR 시스템들은 사각 지대 검출 및/또는 차선 변경 보조를 위해 ADAS 시스템(3238)에서 사용될 수 있다.

[0207] 적어도 하나의 실시예에서, 차량(3200)은 초음파 센서(들)(3262)를 더 포함할 수 있다. 차량(3200)의 전방, 후방, 및/또는 측면들에 위치될 수 있는 초음파 센서(들)(3262)는 주차 보조를 위해 및/또는 점유 그리드를 생성하고 업데이트하기 위해 사용될 수 있다. 적어도 하나의 실시예에서, 매우 다양한 초음파 센서(들)(3262)가 사용될 수 있고, 상이한 검출 범위들(예를 들어, 2.5m, 4m)에 대해 상이한 초음파 센서(들)(3262)가 이용될 수 있다. 적어도 하나의 실시예에서, 초음파 센서(들)(3262)는 ASIL B의 기능적 안전 레벨에서 동작할 수 있다.

[0208] 적어도 하나의 실시예에서, 차량(3200)은 LIDAR 센서(들)(3264)를 포함할 수 있다. LIDAR 센서(들)(3264)는 객체 및 보행자 검출, 비상 제동, 충돌 회피, 및/또는 다른 기능들을 위해 사용될 수 있다. 적어도 하나의 실시예에서, LIDAR 센서(들)(3264)는 기능적 안전 레벨 ASIL B일 수 있다. 적어도 하나의 실시예에서, 차량(3200)은 (예를 들어, 기가비트 이더넷 스위치에 데이터를 제공하기 위해) 이더넷을 사용할 수 있는 다수의 LIDAR 센서들(3264)(예를 들어, 2개, 4개, 6개 등)을 포함할 수 있다.

[0209] 적어도 하나의 실시예에서, LIDAR 센서(들)(3264)는 객체들의 목록 및 360도 시야에 대한 그들의 거리를 제공하는 것이 가능할 수 있다. 적어도 하나의 실시예에서, 상업적으로 이용 가능한 LIDAR 센서(들)(3264)는, 예를 들어, 대략 100m의 광고된 범위, 2cm-3cm의 정확도, 및 100 Mbps 이더넷 접속을 지원할 수 있다. 적어도 하나의 실시예에서, 하나 이상의 비돌출 LIDAR 센서(3264)가 사용될 수 있다. 이러한 실시예에서, LIDAR 센서(들)(3264)는 차량(3200)의 전방, 후방, 측면들, 및/또는 코너들에 내장될 수 있는 소형 디바이스로서 구현될 수 있다. 적어도 하나의 실시예에서, LIDAR 센서(들)(3264)는, 이러한 실시예에서, 심지어 저반사율 객체들에 대해서도 200m 범위로, 최대 120도 수평 및 35도 수직 시야를 제공할 수 있다. 적어도 하나의 실시예에서, 전방 장착 LIDAR 센서(들)(3264)는 45도와 135도 사이의 수평 시야를 위해 구성될 수 있다.

[0210] 적어도 하나의 실시예에서, 3D 플래시 LIDAR과 같은 LIDAR 기술들이 또한 사용될 수 있다. 3D 플래시 LIDAR은 레이저의 플래시를 송신 소스로서 사용하여, 차량(3200)의 주위를 대략 200m까지 조명한다. 적어도 하나의 실시예에서, 플래시 LIDAR 유닛은, 제한없이, 레이저 펄스 통과 시간 및 각각의 픽셀 상의 반사된 광을 기록하는 수용체를 포함하고, 이는 결국 차량(3200)으로부터 객체들까지의 거리에 대응한다. 적어도 하나의 실시예에서, 플래시 LIDAR은 모든 레이저 플래시로 주위의 매우 정확하고 왜곡 없는 이미지들이 생성되게 할 수 있다. 적어도 하나의 실시예에서, 차량(3200)의 각각의 측면에 하나씩, 4개의 플래시 LIDAR 센서가 배치될 수 있다. 적어도 하나의 실시예에서, 3D 플래시 LIDAR 시스템들은, 제한없이, 팬 이외의 이동 부분들(예를 들어, 비-스캐닝 LIDAR 디바이스)이 없는 고체 상태 3D 시작 어레이 LIDAR 카메라를 포함한다. 적어도 하나의 실시예에서, 플래시 LIDAR 디바이스는 프레임 당 5 나노초 클래스 I(눈-안전) 레이저 펄스를 사용할 수 있고, 3D 범위 포인트 클라우드 및 공동 등록된 강도 데이터의 형태로 반사된 레이저 광을 포착할 수 있다.

[0211] 적어도 하나의 실시예에서, 차량은 IMU 센서(들)(3266)를 더 포함할 수 있다. 적어도 하나의 실시예에서, IMU 센서(들)(3266)는, 적어도 하나의 실시예에서, 차량(3200)의 후방 차축의 중심에 위치될 수 있다. 적어도 하나의 실시예에서, IMU 센서(들)(3266)는 예를 들어 가속도계(들), 자력계(들), 자이로스코프(들), 자기 나침반(들) 및/또는 다른 센서 타입들을 포함할 수 있지만 이에 제한되지 않는다. 6축 응용들에서와 같은 적어도 하나의 실시예에서, IMU 센서(들)(3266)는 가속도계 및 자이로스코프를 포함할 수 있지만, 이에 제한되지 않는다. 9축 응용들에서와 같은 적어도 하나의 실시예에서, IMU 센서(들)(3266)는 가속도계, 자이로스코프 및 자력계를 포함할 수 있지만 이에 제한되지 않는다.

- [0212] 적어도 하나의 실시예에서, IMU 센서(들)(3266)는 위치, 속도 및 자세의 추정들을 제공하기 위해 마이크로-전자-기계 시스템("MEMS") 관성 센서들, 고감도 GPS 수신기 및 진보된 칼만 필터링 알고리즘들을 결합하는 소형 고성능 GPS 보조 관성 내비게이션 시스템("GPS/INS")으로서 구현될 수 있다. 적어도 하나의 실시예에서, IMU 센서(들)(3266)는 차량(3200)이 GPS로부터 IMU 센서(들)(3266)로의 속도의 변화들을 직접 관찰하고 상관시킴으로써 자기 센서로부터의 입력을 필요로 하지 않고 진로를 추정하는 것을 가능하게 할 수 있다. 적어도 하나의 실시예에서, IMU 센서(들)(3266) 및 GNSS 센서(들)(3258)는 단일 통합 유닛에 조합될 수 있다.
- [0213] 적어도 하나의 실시예에서, 차량(3200)은 차량(3200) 내에 및/또는 주위에 배치된 마이크로폰(들)(3296)을 포함할 수 있다. 적어도 하나의 실시예에서, 마이크로폰(들)(3296)은, 그 중에서도 특히, 응급 차량 검출 및 식별을 위해 사용될 수 있다.
- [0214] 적어도 하나의 실시예에서, 차량(3200)은 스테레오 카메라(들)(3268), 와이드-뷰 카메라(들)(3270), 적외선 카메라(들)(3272), 서라운드 카메라(들)(3274), 장거리 카메라(들)(3298), 중거리 카메라(들)(3276), 및/또는 다른 카메라 타입들을 포함하는 임의의 수의 카메라 타입들을 더 포함할 수 있다. 적어도 하나의 실시예에서, 차량(3200)의 주위부 전체 주위에서 이미지 데이터를 포착하기 위해 카메라들이 사용될 수 있다. 적어도 하나의 실시예에서, 사용되는 카메라들의 타입들은 차량(3200)에 의존한다. 적어도 하나의 실시예에서, 차량(3200) 주위에 필요한 커버리지를 제공하기 위해 카메라 타입들의 임의의 조합이 사용될 수 있다. 적어도 하나의 실시예에서, 카메라들의 수는 실시예에 따라 상이할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 차량(3200)은 6개의 카메라, 7개의 카메라, 10개의 카메라, 12개의 카메라, 또는 다른 수의 카메라를 포함할 수 있다. 카메라들은 예를 들어 그리고 제한없이, 기가비트 멀티미디어 직렬 링크("GMSL") 및/또는 기가비트 이더넷을 지원할 수 있다. 적어도 하나의 실시예에서, 카메라(들) 각각은 도 32a 및 도 32b와 관련하여 이전에 본 명세서에서 더 상세히 설명되었다.
- [0215] 적어도 하나의 실시예에서, 차량(3200)은 진동 센서(들)(3242)를 더 포함할 수 있다. 진동 센서(들)(3242)는 차축(들)과 같은 차량(3200)의 컴포넌트들의 진동들을 측정할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 진동들의 변화들은 도로 표면들의 변화를 나타낼 수 있다. 적어도 하나의 실시예에서, 2개 이상의 진동 센서(3242)가 사용될 때, 진동들 사이의 차이들은 도로 표면의 마찰 또는 미끄러짐을 결정하기 위해 사용될 수 있다 (예를 들어, 동력-구동 차축과 자유 회전 차축 사이에 진동의 차이가 있을 때).
- [0216] 적어도 하나의 실시예에서, 차량(3200)은 ADAS 시스템(3238)을 포함할 수 있다. ADAS 시스템(3238)은 일부 예들에서, 제한없이, SoC를 포함할 수 있다. 적어도 하나의 실시예에서, ADAS 시스템(3238)은 자율/적응/자동 크루즈 컨트롤("ACC") 시스템, 협력 적응 크루즈 컨트롤("CACC") 시스템, 포워드 충돌 경고("FCW") 시스템, 자동 비상 제동("AEB") 시스템, 차선 이탈 경고("LDW") 시스템, 차선 유지 보조("LKA") 시스템, 사각 지대 경고("BSW") 시스템, 후방 크로스-트래픽 경고("RCTW") 시스템, 충돌 경고("CW") 시스템, 차선 센터링("LC") 시스템, 및/또는 다른 시스템들, 피쳐들, 및/또는 기능성의 임의의 수 및 조합을 포함할 수 있지만, 이에 제한되지 않는다.
- [0217] 적어도 하나의 실시예에서, ACC 시스템은 RADAR 센서(들)(3260), LIDAR 센서(들)(3264), 및/또는 임의의 수의 카메라(들)를 이용할 수 있다. 적어도 하나의 실시예에서, ACC 시스템은 종방향 ACC 시스템 및/또는 측방향 ACC 시스템을 포함할 수 있다. 적어도 하나의 실시예에서, 종방향 ACC 시스템은 차량(3200)의 바로 앞의 차량까지의 거리를 모니터링하고 제어하며, 차량(3200)의 속도를 자동으로 조정하여 앞의 차량들로부터의 안전 거리를 유지한다. 적어도 하나의 실시예에서, 측방향 ACC 시스템은 거리 유지(distance keeping)를 수행하고, 필요할 때에 차선들을 변경하도록 차량(3200)에 조언한다. 적어도 하나의 실시예에서, 측방향 ACC는 LC 및 CW와 같은 다른 ADAS 애플리케이션들에 관련된다.
- [0218] 적어도 하나의 실시예에서, CACC 시스템은 무선 링크를 통해, 또는 간접적으로, 네트워크 접속을 통해(예를 들어, 인터넷을 통해), 다른 차량들로부터 네트워크 인터페이스(3224) 및/또는 무선 안테나(들)(3226)를 통해 수신될 수 있는 다른 차량들로부터의 정보를 사용한다. 적어도 하나의 실시예에서, 직접 링크들은 차량 대 차량("V2V") 통신 링크에 의해 제공될 수 있고, 한편 간접 링크들은 인프라스트럭처-대-차량("I2V") 통신 링크에 의해 제공될 수 있다. 일반적으로, V2V 통신 개념은 직전 차량들(예를 들어, 차량(3200)의 바로 앞에 있고 그와 동일한 차선에 있는 차량들)에 관한 정보를 제공하는 반면, I2V 통신 개념은 더 앞에 있는 트래픽에 관한 정보를 제공한다. 적어도 하나의 실시예에서, CACC 시스템은 I2V 및 V2V 정보 소스들 중 어느 하나 또는 양자 모두를 포함할 수 있다. 적어도 하나의 실시예에서, 차량(3200) 앞의 차량들의 정보가 주어지면, CACC 시스템은 더 신뢰적일 수 있고, 이는 트래픽 흐름 원활성을 개선하고 도로 상의 혼잡을 감소시킬 잠재력을 갖는다.
- [0219] 적어도 하나의 실시예에서, FCW 시스템은 운전자에게 위험을 경보하도록 설계되어, 운전자는 교정 조치를 취할

수 있다. 적어도 하나의 실시예에서, FCW 시스템은 디스플레이, 스피커, 및/또는 진동 컴포넌트와 같은 운전자 피드백에 전기적으로 결합되는 전용 프로세서, DSP, FPGA, 및/또는 ASIC에 결합된 전면 카메라 및/또는 RADAR 센서(들)(3260)를 이용한다. 적어도 하나의 실시예에서, FCW 시스템은 음향, 시각적 경고, 진동 및/또는 급속 제동 펄스의 형태와 같은 경고를 제공할 수 있다.

[0220] 적어도 하나의 실시예에서, AEB 시스템은 다른 차량 또는 다른 객체와의 임박한 전방 충돌을 검출하고, 운전자가 지정된 시간 또는 거리 파라미터 내에서 교정 조치를 취하지 않으면 자동으로 브레이크를 적용할 수 있다. 적어도 하나의 실시예에서, AEB 시스템은 전용 프로세서, DSP, FPGA 및/또는 ASIC에 결합된 전면 카메라(들) 및/또는 RADAR 센서(들)(3260)를 사용할 수 있다. 적어도 하나의 실시예에서, AEB 시스템이 위험을 검출할 때, AEB 시스템은 통상적으로 먼저 충돌을 피하기 위해 교정 조치를 취하도록 운전자에게 경고하고, 운전자가 교정 조치를 취하지 않으면, AEB 시스템은 예측된 충돌의 영향을 방지하거나 적어도 완화하려는 노력으로 브레이크를 자동으로 적용할 수 있다. 적어도 하나의 실시예에서, AEB 시스템은 동적 제동 지원 및/또는 충돌 임박 제동과 같은 기법들을 포함할 수 있다.

[0221] 적어도 하나의 실시예에서, LDW 시스템은 차량(3200)이 차선 마킹들과 교차할 때 운전자에게 경고하기 위해, 조향 휠 또는 시트 진동들과 같은 시각적, 청각적, 및/또는 촉각적 경고들을 제공한다. 적어도 하나의 실시예에서, LDW 시스템은 방향 지시등을 활성화함으로써, 드라이버가 의도적인 차선 이탈을 표시할 때에는 활성화되지 않는다. 적어도 하나의 실시예에서, LDW 시스템은 디스플레이, 스피커, 및/또는 진동 컴포넌트와 같은 운전자 피드백에 전기적으로 결합되는, 전용 프로세서, DSP, FPGA, 및/또는 ASIC에 결합되는, 전면 카메라들을 사용할 수 있다. 적어도 하나의 실시예에서, LKA 시스템은 LDW 시스템의 변형이다. LKA 시스템은 차량(3200)이 차선에서 벗어나기 시작하는 경우 차량(3200)을 교정하기 위해 조향 입력 또는 제동을 제공한다.

[0222] 적어도 하나의 실시예에서, BSW 시스템은 자동차의 사각 지대에서 차량들을 검출하고 운전자에게 경고한다. 적어도 하나의 실시예에서, BSW 시스템은 병합 또는 변화하는 차선들이 안전하지 않다는 것을 나타내기 위해 시각적, 청각적, 및/또는 촉각적 경보를 제공할 수 있다. 적어도 하나의 실시예에서, BSW 시스템은 운전자가 방향 지시등을 사용할 때 추가적인 경고를 제공할 수 있다. 적어도 하나의 실시예에서, BSW 시스템은, 디스플레이, 스피커, 및/또는 진동 컴포넌트와 같은, 운전자 피드백에 전기적으로 결합되는, 전용 프로세서, DSP, FPGA, 및/또는 ASIC에 결합되는, 후면 카메라(들) 및/또는 RADAR 센서(들)(3260)를 사용할 수 있다.

[0223] 적어도 하나의 실시예에서, RCTW 시스템은 차량(3200)이 후진하고 있을 때 후방 카메라 범위 밖에서 객체가 검출될 때 시각, 청각, 및/또는 촉각 통지를 제공할 수 있다. 적어도 하나의 실시예에서, RCTW 시스템은 차량 브레이크들이 충돌을 회피하기 위하여 적용되는 것을 보장하기 위한 AEB 시스템을 포함한다. 적어도 하나의 실시예에서, RCTW 시스템은 디스플레이, 스피커, 및/또는 진동 컴포넌트와 같은 운전자 피드백에 전기적으로 결합되는 전용 프로세서, DSP, FPGA, 및/또는 ASIC에 결합된 하나 이상의 후방 RADAR 센서(들)(3260)를 사용할 수 있다.

[0224] 적어도 하나의 실시예에서, 종래의 ADAS 시스템들은 운전자를 성가시고 산만해지게 할 수 있는 위양성 결과들(false positive results)에 취약할 수 있지만, 그러나, 종래의 ADAS 시스템들은 운전자에게 경보를 발하고 운전자가 안전 조건이 진정으로 존재하는지를 결정하고 그에 따라 조치할 수 있게 하기 때문에, 통상적으로 치명적이지 않다. 적어도 하나의 실시예에서, 결과들이 상충되는 경우에, 차량(3200) 자체가 주 컴퓨터 또는 보조 컴퓨터(예를 들어, 제1 제어기(3236) 또는 제2 제어기(3236)) 중 어느 것으로부터의 결과에 주의를 기울일지 여부를 결정한다. 예를 들어, 적어도 하나의 실시예에서, ADAS 시스템(3238)은 백업 컴퓨터 합리성 모듈에 인지 정보를 제공하기 위한 백업 및/또는 보조 컴퓨터일 수 있다. 적어도 하나의 실시예에서, 백업 컴퓨터 합리성 모니터는 인지 및 동적 운전 태스크에서의 결함을 검출하기 위해 하드웨어 컴포넌트 상에서 중복된 다양한 소프트웨어를 실행할 수 있다. 적어도 하나의 실시예에서, ADAS 시스템(3238)으로부터의 출력들은 감독 MCU에 제공될 수 있다. 적어도 하나의 실시예에서, 주 컴퓨터와 보조 컴퓨터로부터의 출력들이 충돌하면, 감독 MCU는 안전한 동작을 보장하기 위해 충돌을 어떻게 조정할지를 결정한다.

[0225] 적어도 하나의 실시예에서, 주 컴퓨터는 선택된 결과에 대한 주 컴퓨터의 신뢰도를 나타내는 신뢰도 점수를 감독 MCU에 제공하도록 구성될 수 있다. 적어도 하나의 실시예에서, 신뢰도 점수가 임계값을 초과하면, 감독 MCU는 보조 컴퓨터가 충돌하는 또는 불일치하는 결과를 제공하는지에 관계없이, 주 컴퓨터의 지시를 따를 수 있다. 적어도 하나의 실시예에서, 신뢰도 점수가 임계값을 충족시키지 못하고, 주 및 보조 컴퓨터가 상이한 결과들(예를 들어, 충돌)을 나타내는 경우, 감독 MCU는 컴퓨터들 사이를 중재하여 적합한 결과를 결정할 수 있다.

[0226] 적어도 하나의 실시예에서, 감독 MCU는 주 컴퓨터 및 보조 컴퓨터로부터의 출력들에 적어도 부분적으로 기초하

여, 보조 컴퓨터가 거짓 경보들을 제공하는 조건들을 결정하도록 훈련되고 구성된 신경망(들)을 실행하도록 구성될 수 있다. 적어도 하나의 실시예에서, 감독 MCU 내의 신경망(들)은 보조 컴퓨터의 출력이 언제 신뢰될 수 있는지, 그리고 언제 신뢰될 수 없는지를 학습할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 보조 컴퓨터가 RADAR 기반 FCW 시스템일 때, 감독 MCU 내의 신경망(들)은 FCW 시스템이 경보를 트리거하는 배수 그레이트(drainage grate) 또는 맨홀 커버(manhole cover)와 같은 사실상 위험하지 않은 금속 객체들을 식별하는 경우를 학습할 수 있다. 적어도 하나의 실시예에서, 보조 컴퓨터가 카메라-기반 LDW 시스템일 때, 감독 MCU 내의 신경망은 자전거 타는 사람들 또는 보행자들이 존재하고 차선 이탈이 실제로 가장 안전한 기동일 때 LDW를 오버라이드하도록 학습할 수 있다. 적어도 하나의 실시예에서, 감독 MCU는 연관된 메모리를 갖는 신경망(들)을 실행하기에 적합한 DLA 또는 GPU 중 적어도 하나를 포함할 수 있다. 적어도 하나의 실시예에서, 감독 MCU는 SoC(들)(3204)의 컴포넌트를 포함할 수 있고 및/또는 그 컴포넌트로서 포함될 수 있다.

[0227] 적어도 하나의 실시예에서, ADAS 시스템(3238)은 컴퓨터 비전의 전통적인 규칙들을 이용하여 ADAS 기능을 수행하는 보조 컴퓨터를 포함할 수 있다. 적어도 하나의 실시예에서, 보조 컴퓨터는 고전적인 컴퓨터 비전 규칙들(이프-덴(if-then))을 이용할 수 있으며, 감독 MCU 내의 신경망(들)의 존재는 신뢰성, 안전 및 성능을 개선할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 다양한 구현 및 의도적인 불일치(non-identity)는, 특히 소프트웨어(또는 소프트웨어-하드웨어 인터페이스) 기능에 의해 야기된 결함에 대해, 전체 시스템을 더 내결함성 있게 한다. 예를 들어, 적어도 하나의 실시예에서, 주 컴퓨터에서 실행되는 소프트웨어에 소프트웨어 버그(bug) 또는 에러가 존재하고, 보조 컴퓨터에서 실행되는 동일하지 않은 소프트웨어 코드가 동일한 전체 결과를 제공하는 경우, 이때, 감독 MCU는 전체 결과가 정확하다는 더 큰 신뢰를 가질 수 있고, 주 컴퓨터 상의 소프트웨어 또는 하드웨어의 버그는 중대한 에러를 야기하지 않는다.

[0228] 적어도 하나의 실시예에서, ADAS 시스템(3238)의 출력은 주 컴퓨터의 인지 블록 및/또는 주 컴퓨터의 동적 주행 태스크 블록에 공급될 수 있다. 예를 들어, 적어도 하나의 실시예에서, ADAS 시스템(3238)이 바로 앞의 객체로 인한 순방향 충돌 경고를 나타내는 경우, 인지 블록은 객체들을 식별할 때 이 정보를 사용할 수 있다. 적어도 하나의 실시예에서, 보조 컴퓨터는 본 명세서에 설명되는 바와 같이 훈련되는 그 자신의 신경망을 가질 수 있으며, 따라서 위양성들의 위험을 감소시킬 수 있다.

[0229] 적어도 하나의 실시예에서, 차량(3200)은 인포테인먼트 SoC(3230)(예를 들어, IVI(in-vehicle infotainment system))를 더 포함할 수 있다. SoC로서 예시되고 설명되었지만, 인포테인먼트 시스템(3230)은, 적어도 하나의 실시예에서, SoC가 아닐 수 있고, 제한없이, 2개 이상의 개별 컴포넌트들을 포함할 수 있다. 적어도 하나의 실시예에서, 인포테인먼트 SoC(3230)은 오디오(예를 들어, 음악, 개인 휴대 정보 단말기, 내비게이션 명령어들, 뉴스, 라디오 등), 비디오(예를 들어, TV, 영화들, 스트리밍 등), 전화(예를 들어, 핸드프리 통화), 네트워크 접속성(예를 들어, LTE, WiFi 등), 및/또는 정보 서비스들(예를 들어, 내비게이션 시스템들, 후방 주차 보조, 라디오 데이터 시스템, 연료 레벨, 총 주행 거리, 브레이크 연료 레벨, 오일 레벨, 도어 개방/폐쇄, 공기 필터 정보 등과 같은 차량 관련 정보)을 차량(3200)에 제공하기 위해 사용될 수 있는 하드웨어와 소프트웨어의 조합을 제한없이 포함할 수 있다. 예를 들어, 인포테인먼트 SoC(3230)은 라디오들, 디스크 플레이어들, 내비게이션 시스템들, 비디오 플레이어들, USB 및 블루투스 접속성, 카퓨터들, 차내 엔터테인먼트, WiFi, 조향 휠 오디오 제어들, 핸드프리 음성 제어, 헤드-업 디스플레이("HUD"), HMI 디스플레이(3234), 텔레매틱스 디바이스, 제어 패널(예를 들어, 다양한 컴포넌트들, 피쳐들, 및/또는 시스템들과 제어 및/또는 상호작용하기 위한), 및/또는 다른 컴포넌트들을 포함할 수 있다. 적어도 하나의 실시예에서, 인포테인먼트 SoC(3230)은, ADAS 시스템(3238)으로부터의 정보, 계획된 차량 기동들, 궤적들, 주위 환경 정보(예를 들어, 교차로 정보, 차량 정보, 도로 정보 등)와 같은 자율 주행 정보, 및/또는 다른 정보와 같은, 정보(예를 들어, 시각적 및/또는 청각적)를 차량의 사용자(들)에게 제공하는 데 추가로 사용될 수 있다.

[0230] 적어도 하나의 실시예에서, 인포테인먼트 SoC(3230)은 임의의 양 및 타입의 GPU 기능성을 포함할 수 있다. 적어도 하나의 실시예에서, 인포테인먼트 SoC(3230)은 버스(3202)(예를 들어, CAN 버스, 이더넷 등)를 통해 차량(3200)의 다른 디바이스들, 시스템들, 및/또는 컴포넌트들과 통신할 수 있다. 적어도 하나의 실시예에서, 인포테인먼트 SoC(3230)은, 주 제어기(들)(3236)(예를 들어, 차량(3200)의 주 및/또는 백업 컴퓨터들)가 고장나는 경우에 인포테인먼트 시스템의 GPU가 일부 자기 구동 기능들을 수행할 수 있도록 감독 MCU에 결합될 수 있다. 적어도 하나의 실시예에서, 인포테인먼트 SoC(3230)은 본 명세서에 설명된 바와 같이, 차량(3200)을 안전 정지 모드에 대한 쇼퍼에 배치할 수 있다.

[0231] 적어도 하나의 실시예에서, 차량(3200)은 계기 클러스터(3232)(예를 들어, 디지털 대시, 전자 계기 클러스터, 디지털 계기 패널 등)를 더 포함할 수 있다. 계기 클러스터(3232)는, 제한없이, 제어기 및/또는 슈퍼컴퓨터(예

를 들어, 개별 제어기 또는 슈퍼컴퓨터)를 포함할 수 있다. 적어도 하나의 실시예에서, 계기 클러스터(3232)는 속도계, 연료 레벨, 오일 압력, 타코미터, 주행기록계, 회전 지시기, 기어시프트 위치 지시기, 좌석 벨트 경고 등(들), 주차 브레이크 경고등(들), 엔진 오작동등(들), 보조 구속 시스템(예를 들어, 에어백) 정보, 조명 제어, 안전 시스템 제어, 내비게이션 정보 등과 같은 임의의 수 및 조합의 계기 세트를 포함할 수 있지만, 이에 제한되지 않는다. 일부 예들에서, 정보는 인포테인먼트 SoC(3230) 및 계기 클러스터(3232) 사이에 디스플레이 및/또는 공유될 수 있다. 적어도 하나의 실시예에서, 계기 클러스터(3232)는 인포테인먼트 SoC(3230)의 일부로서 포함될 수 있거나, 그 반대일 수 있다.

[0232] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 신경망 훈련 연산들, 신경망 기능들 및/또는 아키텍처들, 또는 본 명세서에 설명된 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 연산들을 추론하거나 예측하기 위해 도 32c의 시스템에서 이용될 수 있다. 적어도 하나의 실시예에서, 자율 차량을 제어 및 지향시키기 위해, 후술되는 컴퓨터 하드웨어로 구현되는 내비게이션 시스템에 객체 배향이 제공된다.

[0233] 도 32d는 적어도 하나의 실시예에 따른, 도 32a의 클라우드 기반 서버(들)와 자율 차량(3200) 사이의 통신을 위한 시스템(3276)의 도면이다. 적어도 하나의 실시예에서, 시스템(3276)은, 제한없이, 서버(들)(3278), 네트워크(들)(3290), 및 차량(3200)을 포함하는 임의의 수 및 타입의 차량들을 포함할 수 있다. 서버(들)(3278)는 복수의 GPU들(3284(A)-3284(H))(집합적으로 본 명세서에서 GPU들(3284)로 지칭됨), PCIe 스위치들(3282(A)-3282(D))(집합적으로 본 명세서에서 PCIe 스위치들(3282)로 지칭됨), 및/또는 CPU들(3280(A)-3280(B))(집합적으로 본 명세서에서 CPU들(3280)로 지칭됨)을 포함할 수 있지만, 이에 제한되지 않는다. GPU들(3284), CPU들(3280), 및 PCIe 스위치들(3282)은, 예를 들어 그리고 제한없이, NVIDIA 및/또는 PCIe 접속들(3286)에 의해 개발된 NVLink 인터페이스들(3288)과 같은 고속 인터커넥트들로 상호접속될 수 있다. 적어도 하나의 실시예에서, GPU들(3284)은 NVLink 및/또는 NVSwitch SoC를 통해 접속되고, GPU들(3284) 및 PCIe 스위치들(3282)은 PCIe 인터커넥트들을 통해 접속된다. 적어도 하나의 실시예에서, 8개의 GPU(3284), 2개의 CPU(3280), 및 4개의 PCIe 스위치(3282)가 예시되어 있지만, 이는 제한을 의도하지 않는다. 적어도 하나의 실시예에서, 서버(들)(3278) 각각은, 제한없이, 임의의 수의 GPU들(3284), CPU들(3280), 및/또는 PCIe 스위치들(3282)을 임의의 조합으로 포함할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 서버(들)(3278) 각각은 8개, 16개, 32개, 및/또는 그 이상의 GPU들(3284)을 포함할 수 있다.

[0234] 적어도 하나의 실시예에서, 서버(들)(3278)는, 네트워크(들)(3290)를 통해 그리고 차량들로부터, 최근에 시작된 도로 작업과 같은, 예상치 못한 또는 변경된 도로 상태들을 보여주는 이미지들을 나타내는 이미지 데이터를 수신할 수 있다. 적어도 하나의 실시예에서, 서버(들)(3278)는, 네트워크(들)(3290)를 통해 그리고 차량들로, 트랙픽 및 도로 상태들에 관한 정보(이들로 제한되지 않음)를 비롯한, 신경망들(3292), 업데이트된 신경망들(3292), 및/또는 지도 정보(3294)를 송신할 수 있다. 적어도 하나의 실시예에서, 지도 정보(3294)에 대한 업데이트들은 건설 현장, 포트홀, 우회로, 범람 및/또는 다른 장애물에 관한 정보와 같은 HD 지도(3222)에 대한 업데이트들을 포함할 수 있지만 이에 제한되지 않는다. 적어도 하나의 실시예에서, 신경망들(3292), 업데이트된 신경망들(3292), 및/또는 지도 정보(3294)는 환경 내의 임의의 수의 차량들로부터 수신된 데이터에서 표현되는 새로운 훈련 및/또는 경험들로부터, 및/또는 (예를 들어, 서버(들)(3278) 및/또는 다른 서버들을 사용하여) 데이터 센터에서 수행되는 훈련에 적어도 부분적으로 기초하여 생성되었을 수 있다.

[0235] 적어도 하나의 실시예에서, 서버(들)(3278)는, 훈련 데이터에 적어도 부분적으로 기초하여 머신 러닝 모델(예를 들어, 신경망)을 훈련시키는데 이용될 수 있다. 훈련 데이터는 차량들에 의해 생성될 수 있고, 및/또는 시뮬레이션에서 생성될 수 있다(예를 들어, 게임 엔진을 이용하여). 적어도 하나의 실시예에서, 임의의 양의 훈련 데이터가 태깅되고(예를 들어, 연관된 신경망에 지도 학습이 유익한 경우) 그리고/또는 다른 전처리를 거친다. 적어도 하나의 실시예에서, (예를 들어, 연관된 신경망이 지도 학습을 요구하지 않는 경우) 임의의 양의 훈련 데이터가 태깅 및/또는 전처리되지 않는다. 적어도 하나의 실시예에서, 일단 머신 러닝 모델들이 훈련되면, 머신 러닝 모델들은 차량들에 의해 사용될 수 있고(예를 들어, 네트워크(들)(3290)를 통해 차량들에 송신될 수 있고), 및/또는 머신 러닝 모델들은 차량들을 원격으로 모니터링하기 위해 서버(들)(3278)에 의해 사용될 수 있다.

[0236] 적어도 하나의 실시예에서, 서버(들)(3278)는 차량들로부터 데이터를 수신하고 실시간 지능형 추론을 위해 최신 실시간 신경망들에 데이터를 적용할 수 있다. 적어도 하나의 실시예에서, 서버(들)(3278)는, NVIDIA에 의해 개

발된 DGX 및 DGX 스테이션 머신들과 같은, GPU(들)(3284)에 의해 작동되는 딥 러닝 슈퍼컴퓨터들 및/또는 전용 AI 컴퓨터들을 포함할 수 있다. 그러나, 적어도 하나의 실시예에서, 서버(들)(3278)는 CPU-전원 데이터 센터들을 사용하는 딥 러닝 인프라스트럭처를 포함할 수 있다.

[0237] 적어도 하나의 실시예에서, 서버(들)(3278)의 딥 러닝 인프라스트럭처는 고속, 실시간 추론이 가능할 수 있고, 차량(3200) 내의 프로세서들, 소프트웨어, 및/또는 연관된 하드웨어의 건전성을 평가하고 검증하기 위해 그 능력을 사용할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 딥 러닝 인프라스트럭처는 (예를 들어, 컴퓨터 비전 및/또는 다른 머신 러닝 객체 분류 기술들을 통해) 차량(3200)이 그 이미지들의 시퀀스에 위치한 이미지들 및/또는 객체들의 시퀀스와 같은 주기적 업데이트들을 차량(3200)으로부터 수신할 수 있다. 적어도 하나의 실시예에서, 딥 러닝 인프라스트럭처는 객체들을 식별하고 이들을 차량(3200)에 의해 식별된 객체들과 비교하기 위해 그 자신의 신경망을 실행할 수 있고, 결과들이 일치하지 않고 딥 러닝 인프라스트럭처가 차량(3200) 내의 AI가 오작동하고 있다고 결론내리면, 이때, 서버(들)(3278)는 차량(3200)의 장애 안전 컴퓨터에 제어를 취하고, 승객들에게 통지하고, 안전한 주차 기동을 완료하도록 지시하는 신호를 차량(3200)에 송신할 수 있다.

[0238] 적어도 하나의 실시예에서, 서버(들)(3278)는 GPU(들)(3284) 및 하나 이상의 프로그램가능 추론 가속기(예를 들어, NVIDIA의 TensorRT 3)를 포함할 수 있다. 적어도 하나의 실시예에서, GPU-전원 서버들과 추론 가속의 조합은 실시간 응답성을 가능하게 할 수 있다. 성능이 덜 중요한 경우와 같은 적어도 하나의 실시예에서, CPU들, FPGA들, 및 다른 프로세서들에 의해 작동되는 서버들이 추론을 위해 사용될 수 있다. 적어도 하나의 실시예에서, 하드웨어 구조(들)(2915)는 하나 이상의 실시예를 수행하는데 이용된다. 하드웨어 구조(x)(2915)에 관한 상세는 도 29a 및/또는 도 29b와 관련하여 여기서 제공된다.

[0239] 컴퓨터 시스템들

[0240] 도 33은 예시적인 컴퓨터 시스템을 예시하는 블록도이고, 이는 적어도 하나의 실시예에 따라 상호접속된 디바이스들 및 컴포넌트들, 명령어를 실행하기 위한 실행 유닛들을 포함할 수 있는 프로세서로 형성된 시스템-온-칩(SOC) 또는 이들의 일부 조합(3300)을 갖는 시스템일 수 있다. 적어도 하나의 실시예에서, 컴퓨터 시스템(3300)은, 제한없이, 본 명세서에 설명된 실시예에서와 같이, 본 개시내용에 따른, 데이터를 처리하기 위한 알고리즘들을 수행하는 로직을 포함하는 실행 유닛들을 채용하는 프로세서(3302)와 같은 컴포넌트를 포함할 수 있다. 적어도 하나의 실시예에서, 컴퓨터 시스템(3300)은, 캘리포니아주, Santa Clara, Intel Corporation으로부터 입수가능한 PENTIUM® 프로세서 제품군, Xeon™, Itanium® XScale™ 및/또는 StrongARM™, Intel® Core™ 또는 Intel® Nervana™ 마이크로프로세서들 같은 프로세서들을 포함할 수 있지만, 다른 시스템들(다른 마이크로프로세서들, 엔지니어링 워크스테이션들, 셋톱 박스들 등을 갖는 PC들로 포함)도 역시 사용될 수 있다. 적어도 하나의 실시예에서, 컴퓨터 시스템(3300)은 미국 워싱턴주 레드몬드 소재의 Microsoft Corporation으로부터 입수가능한 WINDOWS의 운영 체제의 버전을 실행할 수 있지만, 다른 운영 체제들(예를 들어, UNIX 및 Linux), 임베디드 소프트웨어, 및/또는 그래픽 사용자 인터페이스들도 사용될 수 있다.

[0241] 실시예들은 핸드헬드 디바이스들 및 임베디드 애플리케이션들 같은 다른 디바이스들에서 이용될 수 있다. 핸드헬드 디바이스들의 일부 예들은 셀룰러 폰, 인터넷 프로토콜 디바이스, 디지털 카메라, PDA(personal digital assistant), 및 핸드헬드 PC를 포함한다. 적어도 하나의 실시예에서, 임베디드 애플리케이션들은 마이크로컨트롤러, 디지털 신호 프로세서("DSP"), 시스템-온-칩, 네트워크 컴퓨터("NetPC"), 셋톱 박스, 네트워크 허브, 광역 네트워크("WAN") 스위치, 또는 적어도 하나의 실시예에 따라 하나 이상의 명령어를 수행할 수 있는 임의의 다른 시스템을 포함할 수 있다.

[0242] 적어도 하나의 실시예에서, 컴퓨터 시스템(3300)은, 제한없이, 본 명세서에 설명된 기법에 따라 머신 러닝 모델 훈련 및/또는 추론을 수행하는 하나 이상의 실행 유닛(3308)을 제한없이 포함할 수 있는 프로세서(3302)를 포함할 수 있다. 적어도 하나의 실시예에서, 시스템(33)은 단일 프로세서 데스크탑 또는 서버 시스템이지만, 그러나, 다른 실시예에서, 시스템(33)은 멀티프로세서 시스템일 수 있다. 적어도 하나의 실시예에서, 프로세서(3302)는, 제한없이, 예를 들어, CISC(complex instruction set computer) 마이크로프로세서, RISC(reduced instruction set computing) 마이크로프로세서, VLIW(very long instruction word) 마이크로프로세서, 명령어 세트들의 조합을 구현하는 프로세서, 또는 디지털 신호 프로세서와 같은 임의의 다른 프로세서 디바이스를 포함할 수 있다. 적어도 하나의 실시예에서, 프로세서(3302)는 프로세서(3302)와 컴퓨터 시스템(3300) 내의 다른 컴포넌트들 사이에 데이터 신호들을 송신할 수 있는 프로세서 버스(3310)에 결합될 수 있다.

[0243] 적어도 하나의 실시예에서, 프로세서(3302)는, 제한없이, 레벨 1("L1") 내부 캐시 메모리("캐시")(3304)를 포함할 수 있다. 적어도 하나의 실시예에서, 프로세서(3302)는 단일 내부 캐시 또는 다수의 레벨의 내부 캐시를 가

질 수 있다. 적어도 하나의 실시예에서, 캐시 메모리는 프로세서(3302) 외부에 상주할 수 있다. 다른 실시예들도 역시, 특정한 구현 및 필요성에 따라 내부 및 외부 캐시 양자 모두의 조합을 포함할 수 있다. 적어도 하나의 실시예에서, 레지스터 파일(3306)은, 제한없이, 정수 레지스터, 부동 소수점 레지스터, 상태 레지스터, 및 명령어 포인터 레지스터를 포함한 다양한 레지스터에 상이한 타입의 데이터를 저장할 수 있다.

[0244] 적어도 하나의 실시예에서, 정수 및 부동 소수점 연산을 수행하는 로직(이들로 제한되지 않음)을 비롯한 실행 유닛(3308)도 역시 프로세서(3302)에 상주한다. 프로세서(3302)는 또한 특정 매크로 명령들에 대한 마이크로코드를 저장하는 마이크로코드("ucode") 판독 전용 메모리("ROM")를 포함할 수 있다. 적어도 하나의 실시예에서, 실행 유닛(3308)은 패키징된 명령어 세트(3309)를 핸들링하는 로직을 포함할 수 있다. 적어도 하나의 실시예에서, 명령어들을 실행하는 연관된 회로와 함께, 범용 프로세서(3302)의 명령어 세트에 패키징된 명령어 세트(3309)를 포함시킴으로써, 많은 멀티미디어 애플리케이션들에 의해 이용되는 연산들은 범용 프로세서(3302)에서 패키징된 데이터를 이용하여 수행될 수 있다. 하나 이상의 실시예에서, 많은 멀티미디어 애플리케이션은 패키징된 데이터에 관한 연산들을 수행하기 위해 프로세서의 데이터 버스의 전체 폭을 이용함으로써 가속되고 더 효율적으로 실행될 수 있어서, 하나의 데이터 엘리먼트에 관해 한번에 하나 이상의 연산들을 수행하기 위해 프로세서의 데이터 버스를 통해 더 작은 단위들의 데이터를 전송할 필요성을 제거할 수 있다.

[0245] 적어도 하나의 실시예에서, 실행 유닛(3308)은 또한 마이크로컨트롤러들, 임베디드 프로세서들, 그래픽 디바이스들, DSP들, 및 다른 타입들의 로직 회로들에서 사용될 수 있다. 적어도 하나의 실시예에서, 컴퓨터 시스템(3300)은, 제한없이, 메모리(3320)를 포함할 수 있다. 적어도 하나의 실시예에서, 메모리(3320)는 동적 랜덤 액세스 메모리("DRAM") 디바이스, 정적 랜덤 액세스 메모리("SRAM") 디바이스, 플래시 메모리 디바이스, 또는 다른 메모리 디바이스로서 구현될 수 있다. 메모리(3320)는 프로세서(3302)에 의해 실행될 수 있는 데이터 신호들에 의해 표현되는 명령어(들)(3319) 및/또는 데이터(3321)를 저장할 수 있다.

[0246] 적어도 하나의 실시예에서, 시스템 로직 칩은 프로세서 버스(3310) 및 메모리(3320)에 결합될 수 있다. 적어도 하나의 실시예에서, 시스템 로직 칩은, 제한없이, 메모리 제어기 허브("MCH")(3316)를 포함할 수 있고, 프로세서(3302)는 프로세서 버스(3310)를 통해 MCH(3316)와 통신할 수 있다. 적어도 하나의 실시예에서, MCH(3316)는 명령어 및 데이터 저장소를 위해 그리고 그래픽 커맨드들, 데이터 및 텍스처들의 저장을 위해 메모리(3320)에 고대역폭 메모리 경로(3318)를 제공할 수 있다. 적어도 하나의 실시예에서, MCH(3316)는 프로세서(3302), 메모리(3320), 및 컴퓨터 시스템(3300) 내의 다른 컴포넌트들 사이에 데이터 신호들을 지향시키고, 프로세서 버스(3310), 메모리(3320), 및 시스템 I/O(3322) 사이에 데이터 신호들을 가교할 수 있다. 적어도 하나의 실시예에서, 시스템 로직 칩은, 그래픽 제어기에 결합하기 위한 그래픽 포트를 제공할 수 있다. 적어도 하나의 실시예에서, MCH(3316)는 고대역폭 메모리 경로(3318)를 통해 메모리(3320)에 결합될 수 있고, 그래픽/비디오 카드(3312)는 가속 그래픽 포트("AGP") 인터커넥트(3314)를 통해 MCH(3316)에 결합될 수 있다.

[0247] 적어도 하나의 실시예에서, 컴퓨터 시스템(3300)은 MCH(3316)를 I/O 제어기 허브("ICH")(3330)에 결합하기 위해 독점 허브 인터페이스 버스인 시스템 I/O(3322)를 사용할 수 있다. 적어도 하나의 실시예에서, ICH(3330)는 로컬 I/O 버스를 통해 일부 I/O 디바이스들에 대한 직접 접속들을 제공할 수 있다. 적어도 하나의 실시예에서, 로컬 I/O 버스는 주변기기들을 메모리(3320), 칩셋 및 프로세서(3302)에 접속하기 위한 고속 I/O 버스를 포함할 수 있지만 이에 제한되지 않는다. 예들은, 제한없이, 오디오 제어기(3329), 펌웨어 허브("플래시 BIOS")(3328), 무선 트랜시버(3326), 데이터 저장소(3324), 사용자 입력 및 키보드 인터페이스들을 포함하는 레거시 I/O 제어기(3323), USB(Universal Serial Bus)와 같은 직렬 확장 포트(3327), 및 네트워크 제어기(3334)를 포함할 수 있다. 데이터 저장소(3324)는 하드 디스크 드라이브, 플로피 디스크 드라이브, CD-ROM 디바이스, 플래시 메모리 디바이스, 또는 다른 대용량 저장 디바이스를 포함할 수 있다.

[0248] 적어도 하나의 실시예에서, 도 33은 상호접속된 하드웨어 디바이스들 또는 "칩들"을 포함하는 시스템을 예시하는 반면, 다른 실시예들에서, 도 33은 예시적인 시스템-온-칩("SoC")을 예시할 수 있다. 적어도 하나의 실시예에서, 도 33에 예시된 디바이스들은 독점 인터커넥트들, 표준화된 인터커넥트들(예를 들어, PCIe) 또는 이들의 일부 조합으로 상호접속될 수 있다. 적어도 하나의 실시예에서, 시스템(3300)의 하나 이상의 컴포넌트들은 CXL(compute express link) 인터커넥트들을 사용하여 상호접속된다.

[0249] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 신경망 훈련 연산들, 신경망 기능들 및/또는 아키텍처들, 또는 본 명세서에 설명된 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도

부분적으로 기초하여 연산들을 추론하거나 예측하기 위해 도 33의 시스템에서 이용될 수 있다. 적어도 하나의 실시예에서, 객체의 배향을 결정하는 신경망은 도 29 내지 도 31에 설명된 하드웨어를 사용하여 구현될 수 있다. 적어도 하나의 실시예에서, 도 32a 내지 도 32d에 도시된 바와 같은 자율 차량은 차량 근처의 객체들의 배향을 결정하는 신경망을 포함할 수 있다. 적어도 하나의 실시예에서, 자율 차량을 제어 및 지향시키기 위해, 후술되는 컴퓨터 하드웨어로 구현되는 내비게이션 시스템에 객체 배향이 제공된다.

[0250] 도 34는 적어도 하나의 실시예에 따른, 프로세서(3410)를 이용하기 위한 전자 디바이스(3400)를 예시하는 블록도이다. 적어도 하나의 실시예에서, 전자 디바이스(3400)는, 예를 들어 그리고 제한없이, 노트북, 태워 서버, 랩 서버, 블레이드 서버, 랩탑, 데스크탑, 태블릿, 모바일 디바이스, 전화, 임베디드 컴퓨터, 또는 임의의 다른 적합한 전자 디바이스일 수 있다.

[0251] 적어도 하나의 실시예에서, 시스템(3400)은 임의의 적합한 수 또는 종류의 컴포넌트들, 주변기기들, 모듈들, 또는 디바이스들에 통신가능하게 결합된 프로세서(3410)를 제한없이 포함할 수 있다. 적어도 하나의 실시예에서, 프로세서(3410)는 1° C 버스, "SMBus"(System Management Bus), LPC(Low Pin Count) 버스, "SPI"(Serial Peripheral Interface), "HDA"(High Definition Audio) 버스, "SATA"(Serial Advance Technology Attachment) 버스, "USB"(Universal Serial Bus)(버전들 1, 2, 3), 또는 "UART"(Universal Asynchronous Receiver/Transmitter) 버스와 같은 버스 또는 인터페이스를 사용하여 결합된다. 적어도 하나의 실시예에서, 도 34는 상호접속된 하드웨어 디바이스들 또는 "칩들"을 포함하는 시스템을 예시하는 반면, 다른 실시예들에서, 도 34는 예시적인 시스템-온-칩("SoC")을 예시할 수 있다. 적어도 하나의 실시예에서, 도 34에 예시된 디바이스들은 독점 인터커넥트들, 표준화된 인터커넥트들(예를 들어, PCIe) 또는 이들의 일부 조합으로 상호접속될 수 있다. 적어도 하나의 실시예에서, 도 34의 하나 이상의 컴포넌트들은 CXL(compute express link) 인터커넥트들을 이용하여 상호접속된다.

[0252] 적어도 하나의 실시예에서, 도 34는 디스플레이(3424), 터치 스크린(3425), 터치 패드(3430), 근접장 통신 유닛("NFC")(3445), 센서 허브(3440), 열 센서(3446), 익스프레스 칩셋("EC")(3435), 신뢰 플랫폼 모듈("TPM")(3438), BIOS/펌웨어/플래시 메모리("BIOS, FW 플래시")(3422), DSP(3460), 솔리드 스테이트 디스크("SSD") 또는 하드 디스크 드라이브("HDD")와 같은 드라이브("SSD 또는 HDD")(3420), 무선 근거리 네트워크 유닛("WLAN")(3450), 블루투스 유닛(3452), 무선 광역 네트워크 유닛("WWAN")(3456), 글로벌 위치 결정 시스템(GPS)(3455), USB 3.0 카메라와 같은 카메라("USB 3.0 카메라")(3454), 또는 예를 들어, LPDDR3 표준으로 구현된 저전력 더블 데이터 레이트("LPDDR") 메모리 유닛("LPDDR3")(3415)을 포함할 수 있다. 이들 컴포넌트들은 각각 임의의 적합한 방식으로 구현될 수 있다.

[0253] 적어도 하나의 실시예에서, 다른 컴포넌트들은 전술한 컴포넌트들을 통해 프로세서(3410)에 통신가능하게 결합될 수 있다. 적어도 하나의 실시예에서, 가속도계(3441), 주변광 센서("ALS")(3442), 나침반(3443) 및 자이로스 코프(3444)는 센서 허브(3440)에 통신가능하게 결합될 수 있다. 적어도 하나의 실시예에서, 열 센서(3439), 팬(3437), 키보드(3446), 및 터치 패드(3430)는 EC(3435)에 통신가능하게 결합될 수 있다. 적어도 하나의 실시예에서, 스피커(3463), 헤드폰(3464), 및 마이크로폰("mic")(3465)은 오디오 유닛("오디오 코덱 및 클래스 d 앰프")(3462)에 통신가능하게 결합될 수 있고, 이는 차례로 DSP(3460)에 통신가능하게 결합될 수 있다. 적어도 하나의 실시예에서, 오디오 유닛(3464)은, 예를 들어 그리고 제한없이, 오디오 코더/디코더("코덱") 및 클래스 D 증폭기를 포함할 수 있다. 적어도 하나의 실시예에서, SIM 카드("SIM")(3457)는 WWAN 유닛(3456)에 통신가능하게 결합될 수 있다. 적어도 하나의 실시예에서, WLAN 유닛(3450) 및 블루투스 유닛(3452)뿐만 아니라 WWAN 유닛(3456)과 같은 컴포넌트들은 차세대 폼 팩터("NGFF")로 구현될 수 있다.

[0254] 도 35는 적어도 하나의 실시예에 따른 컴퓨터 시스템(3500)을 예시한다. 적어도 하나의 실시예에서, 컴퓨터 시스템(3500)은 본 개시내용 전반에 걸쳐 설명된 다양한 프로세스들 및 방법들을 구현하도록 구성된다.

[0255] 적어도 하나의 실시예에서, 컴퓨터 시스템(3500)은, PCI(Peripheral Component Interconnect), PCI-Express(peripheral component interconnect express), AGP(Accelerated Graphics Port), HyperTransport, 또는 임의의 다른 버스 또는 포인트-투-포인트 통신 프로토콜(들)과 같은, 임의의 적합한 프로토콜을 사용하여 구현되는 통신 버스(3510)에 연결되는 적어도 하나의 중앙 처리 유닛("CPU")(3502)를 포함하지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, 컴퓨터 시스템(3500)은, 제한없이, 메인 메모리(3504) 및 제어 로직(예를 들어, 하드웨어, 소프트웨어, 또는 이들의 조합으로서 구현됨)을 포함하고, 데이터는 랜덤 액세스 메모리("RAM")의 형태를 취할 수 있는 메인 메모리(3504)에 저장된다. 적어도 하나의 실시예에서, 네트워크 인터페이스 서브시스템("네트워크 인터페이스")(3522)은 컴퓨터 시스템(3500)으로부터 데이터를 수신하고 데이터를 다른

시스템들로 송신하기 위한 다른 계산 디바이스들 및 네트워크들에 대한 인터페이스를 제공한다.

- [0256] 적어도 하나의 실시예에서, 컴퓨터 시스템(3500)은, 적어도 하나의 실시예에서, 종래의 CRT(cathode ray tube), LCD(liquid crystal display), LED(light emitting diode), 플라즈마 디스플레이, 또는 다른 적합한 디스플레이 기술들을 사용하여 구현될 수 있는 입력 디바이스들(3508), 병렬 처리 시스템(3512), 및 디스플레이 디바이스들(3506)(이들로 제한되지 않음)을 포함한다. 적어도 하나의 실시예에서, 사용자 입력은 키보드, 마우스, 터치패드, 마이크론 등과 같은 입력 디바이스들(3508)로부터 수신된다. 적어도 하나의 실시예에서, 전술한 모듈들 각각은 처리 시스템을 형성하기 위해 단일 반도체 플랫폼 상에 위치할 수 있다.
- [0257] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 신경망 훈련 연산들, 신경망 기능들 및/또는 아키텍처들, 또는 본 명세서에 설명된 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 연산들을 추론하거나 예측하기 위해 도 35의 시스템에서 이용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 컴퓨터 시스템으로 구현된다.
- [0258] 도 36은 적어도 하나의 실시예에 따른 컴퓨터 시스템(3600)을 예시한다. 적어도 하나의 실시예에서, 컴퓨터 시스템(3600)은, 제한없이, 컴퓨터(3610) 및 USB 스틱(3620)을 포함한다. 적어도 하나의 실시예에서, 컴퓨터(3610)는 임의의 수 및 타입의 프로세서(들)(도시되지 않음) 및 메모리(도시되지 않음)를 제한없이 포함할 수 있다. 적어도 하나의 실시예에서, 컴퓨터(3610)는, 제한없이, 서버, 클라우드 인스턴스, 랩탑, 및 데스크탑 컴퓨터를 포함한다.
- [0259] 적어도 하나의 실시예에서, USB 스틱(3620)은, 제한없이, 처리 유닛(3630), USB 인터페이스(3640), 및 USB 인터페이스 로직(3650)을 포함한다. 적어도 하나의 실시예에서, 처리 유닛(3630)은 명령어들을 실행할 수 있는 임의의 명령어 실행 시스템, 장치, 또는 디바이스일 수 있다. 적어도 하나의 실시예에서, 처리 유닛(3630)은, 제한없이, 임의의 수 및 타입의 처리 코어들(도시되지 않음)을 포함할 수 있다. 적어도 하나의 실시예에서, 처리 코어(3630)는 머신 러닝과 연관된 임의의 양 및 타입의 동작들을 수행하도록 최적화되는 주문형 집적 회로("ASIC")를 포함한다. 예를 들어, 적어도 하나의 실시예에서, 처리 코어(3630)는 머신 러닝 추론 연산들을 수행하도록 최적화되는 텐서 처리 유닛("TPC")이다. 적어도 하나의 실시예에서, 처리 코어(3630)는 머신 비전 및 머신 러닝 추론 연산들을 수행하도록 최적화되는 비전 처리 유닛("VPU")이다.
- [0260] 적어도 하나의 실시예에서, USB 인터페이스(3640)는 임의의 타입의 USB 커넥터 또는 USB 소켓일 수 있다. 예를 들어, 적어도 하나의 실시예에서, USB 인터페이스(3640)는 데이터 및 전력을 위한 USB 3.0 타입-C 소켓이다. 적어도 하나의 실시예에서, USB 인터페이스(3640)는 USB 3.0 타입-A 커넥터이다. 적어도 하나의 실시예에서, USB 인터페이스 로직(3650)은 처리 유닛(3630)이 USB 커넥터(3640)를 통해 애플리케이션들 또는 디바이스들(예를 들어, 컴퓨터(3610))과 인터페이스할 수 있게 하는 임의의 양 및 타입의 로직을 포함할 수 있다.
- [0261] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 신경망 훈련 연산들, 신경망 기능들 및/또는 아키텍처들, 또는 본 명세서에 설명된 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 연산들을 추론하거나 예측하기 위해 도 36의 시스템에서 이용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 GPU 또는 다른 PPU로 구현된다.
- [0262] 도 37a는 복수의 GPU(3710 내지 3713)가 고속 링크(3740 내지 3743)(예를 들어, 버스, 포인트-투-포인트 인터랙트 등)를 통해 복수의 멀티 코어 프로세서(3705 내지 3706)에 통신가능하게 결합되어 있는 예시적인 아키텍처를 예시한다. 하나의 실시예에서, 고속 링크들(3740-3743)은 4GB/s, 30GB/s, 80GB/s 또는 그 이상의 통신 처리량을 지원한다. PCIe 4.0 또는 5.0 및 NVLink 2.0을 포함한 그러나 이것으로 제한되지 않는 다양한 상호접속 프로토콜이 이용될 수 있다.
- [0263] 또한, 하나의 실시예에서, GPU들(3710-3713) 중 2개 이상은 고속 링크들(3729-3730)을 통해 상호접속되고, 이는 고속 링크들(3740-3743)에 이용되는 것들과 동일하거나 상이한 프로토콜들/링크들을 이용하여 구현될 수 있다. 유사하게, 멀티 코어 프로세서들(3705-3706) 중 2개 이상은 20GB/s, 30GB/s, 120GB/s 또는 그 이상에서 동작하는 대칭 멀티 프로세서(SMP) 버스들일 수 있는 고속 링크(3728)를 통해 접속될 수 있다. 대안적으로, 도 37a에

도시된 다양한 시스템 컴포넌트들 사이의 모든 통신은 동일한 프로토콜들/링크들을 사용하여(예를 들어, 공통 상호접속 패브릭을 통해) 달성될 수 있다.

[0264] 하나의 실시예에서, 각각의 멀티 코어 프로세서(3705-3706)는 메모리 인터커넥트들(3726-3727)을 통해 프로세서 메모리(3701-3702)에 각각 통신가능하게 결합되며, 각각의 GPU(3710-3713)는 GPU 메모리 인터커넥트들(3750-3753)을 통해 GPU 메모리(3720-3723)에 각각 통신가능하게 결합된다. 메모리 인터커넥트들(3726-3727 및 3750-3753)은 동일하거나 상이한 메모리 액세스 기술들을 이용할 수 있다. 제한이 아닌 예로서, 프로세서 메모리들(3701-3702) 및 GPU 메모리들(3720-3723)은 동적 랜덤 액세스 메모리(DRAM)(스택형 DRAM을 포함함), 그래픽 DDR SDRAM(GDDR)(예를 들어, GDDR5, GDDR6) 또는 고대역폭 메모리(HBM)와 같은 휘발성 메모리들일 수 있고/있거나, 3D XPoint 또는 나노-램과 같은 비휘발성 메모리들일 수 있다. 하나의 실시예에서, 프로세서 메모리들(3701-3702)의 일부 부분은 휘발성 메모리일 수 있고, 다른 부분은 (예를 들어, 2LM(two-level memory) 계층구조를 이용하는) 비휘발성 메모리일 수 있다.

[0265] 본 명세서에 설명된 바와 같이, 다양한 프로세서들(3705-3706) 및 GPU들(3710-3713)이 특정 메모리(3701-3702, 3720-3723)에 각각 물리적으로 결합될 수 있지만, 동일한 가상 시스템 어드레스 공간("유효 어드레스" 공간이라고도 지칭됨)이 다양한 물리적 메모리들 사이에 분산되는 통합 메모리 아키텍처가 구현될 수 있다. 예를 들어, 프로세서 메모리들(3701-3702)은 각각 64GB의 시스템 메모리 어드레스 공간을 포함할 수 있고, GPU 메모리들(3720-3723)은 각각 32GB의 시스템 메모리 어드레스 공간을 포함할 수 있다(결과적으로 이 예에서는 총 256GB 어드레스 가능 메모리).

[0266] 도 37b는 하나의 예시적인 실시예에 따른 멀티 코어 프로세서(3707)와 그래픽 가속 모듈(3746) 사이의 상호접속에 대한 추가적인 상세사항들을 예시한다. 그래픽 가속 모듈(3746)은 고속 링크(3740)를 통해 프로세서(3707)에 결합되는 라인 카드 상에 통합된 하나 이상의 GPU 칩들을 포함할 수 있다. 대안적으로, 그래픽 가속 모듈(3746)은 프로세서(3707)와 동일한 패키지 또는 칩 상에 통합될 수 있다.

[0267] 적어도 하나의 실시예에서, 예시된 프로세서(3707)는 복수의 코어(3760A-3760D)를 포함하고, 각각은 변환 색인 버퍼(3761A-3761D) 및 하나 이상의 캐시(3762A-3762D)를 갖는다. 적어도 하나의 실시예에서, 코어들(3760A-3760D)은 도시되지 않은 명령어들을 실행하고 데이터를 처리하기 위한 다양한 다른 컴포넌트들을 포함할 수 있다. 캐시들(3762A-3762D)은 레벨 1(L1) 및 레벨 2(L2) 캐시들을 포함할 수 있다. 또한, 하나 이상의 공유 캐시(3756)가 캐시들(3762A-3762D)에 포함되고 코어들(3760A-3760D)의 세트들에 의해 공유될 수 있다. 예를 들어, 프로세서(3707)의 하나의 실시예는 24개의 코어를 포함하고, 각각은 그 자신의 L1 캐시, 12개의 공유 L2 캐시, 및 12개의 공유 L3 캐시를 갖는다. 이 실시예에서, 하나 이상의 L2 및 L3 캐시는 2개의 인접한 코어에 의해 공유된다. 프로세서(3707) 및 그래픽 가속 모듈(3746)은 도 37a의 프로세서 메모리들(3701-3702)을 포함할 수 있는 시스템 메모리(3714)와 접속된다.

[0268] 일관성 버스(3764)를 통한 코어 간 통신을 통해 다양한 캐시들(3762A-3762D, 3756) 및 시스템 메모리(3714)에 저장된 데이터 및 명령어들에 대해 일관성이 유지된다. 예를 들어, 각각의 캐시는 특정 캐시 라인들에 대한 검출된 판독들 또는 기입들에 응답하여 일관성 버스(3764)를 통해 통신하기 위해 그와 연관된 캐시 일관성 로직/회로를 가질 수 있다. 일 구현에서, 캐시 스누핑 프로토콜은 캐시 액세스들을 스누핑하기 위해 일관성 버스(3764)를 통해 구현된다.

[0269] 하나의 실시예에서, 프록시 회로(3725)는 그래픽 가속 모듈(3746)을 일관성 버스(3764)에 통신가능하게 결합시켜, 그래픽 가속 모듈(3746)이 코어들(3760A-3760D)의 피어로서 캐시 일관성 프로토콜에 참여할 수 있게 한다. 특히, 인터페이스(3735)는 고속 링크(3740)(예를 들어, PCIe 버스, NVLink 등)를 통해 프록시 회로(3725)에의 접속성을 제공하고, 인터페이스(3737)는 그래픽 가속 모듈(3746)을 링크(3740)에 접속시킨다.

[0270] 일 구현에서, 가속기 통합 회로(3736)는 그래픽 가속 모듈(3746)의 복수의 그래픽 처리 엔진(3731, 3732, N)을 대신하여 캐시 관리, 메모리 액세스, 컨텍스트 관리, 및 인터럽트 관리 서비스들을 제공한다. 그래픽 처리 엔진들(3731, 3732, N)은 각각 별개의 그래픽 처리 유닛(GPU)을 포함할 수 있다. 대안적으로, 그래픽 처리 엔진들(3731, 3732, N)은 그래픽 실행 유닛들, 미디어 처리 엔진들(예를 들어, 비디오 인코더들/디코더들), 샘플러들, 및 블릿 엔진들과 같은 GPU 내의 상이한 타입들의 그래픽 처리 엔진들을 포함할 수 있다. 적어도 하나의 실시예에서, 그래픽 가속 모듈(3746)은 복수의 그래픽 처리 엔진(3731-3732, N)을 갖는 GPU일 수 있거나, 또는 그래픽 처리 엔진(3731-3732, N)은 공통 패키지, 라인 카드 또는 칩 상에 통합된 개별 GPU들일 수 있다.

[0271] 하나의 실시예에서, 가속기 통합 회로(3736)는, 가상-대-물리 메모리 변환들(유효-대-실제 메모리 변환들이라고

도 지칭됨) 및 시스템 메모리(3714)에 액세스하기 위한 메모리 액세스 프로토콜들과 같은 다양한 메모리 관리 기능들을 수행하기 위한 메모리 관리 유닛(MMU)(3739)을 포함한다. MMU(3739)는 또한 가상/유효 대 물리적/실제 어드레스 변환들을 캐싱하기 위한 변환 색인 버퍼(TLB)(도시되지 않음)를 포함할 수 있다. 일 구현에서, 캐시(3738)는 그래픽 처리 엔진들(3731-3732, N)에 의한 효율적인 액세스를 위해 커맨드들 및 데이터를 저장한다. 하나의 실시예에서, 캐시(3738) 및 그래픽 메모리들(3733-3734, M)에 저장된 데이터는 코어 캐시들(3762A-3762D, 3756) 및 시스템 메모리(3714)와 일관성을 유지한다. 언급된 바와 같이, 이는 캐시(3738) 및 메모리들(3733-3734, M)을 대신하여 프록시 회로(3725)를 통해 달성될 수 있다(예를 들어, 프로세서 캐시들(3762A-3762D, 3756) 상의 캐시 라인들의 수정들/액세스들과 관련된 업데이트들을 캐시(3738)에 전송하고 캐시(3738)로부터 업데이트들을 수신한다).

[0272] 레지스터들의 세트(3745)는 그래픽 처리 엔진들(3731-3732, N)에 의해 실행되는 스레드들에 대한 컨텍스트 데이터를 저장하고, 컨텍스트 관리 회로(3748)는 스레드 컨텍스트들을 관리한다. 예를 들어, 컨텍스트 관리 회로(3748)는 컨텍스트 전환들 동안(예를 들어, 제2 스레드가 그래픽 처리 엔진에 의해 실행될 수 있도록 제1 스레드가 저장되고 제2 스레드가 저장되는 경우) 다양한 스레드들의 컨텍스트들을 저장 및 복원하기 위해 저장 및 복원 동작들을 수행할 수 있다. 예를 들어, 컨텍스트 전환 시에, 컨텍스트 관리 회로(3748)는 현재 레지스터 값들을 (예를 들어, 컨텍스트 포인터에 의해 식별되는) 메모리 내의 지정된 영역에 저장할 수 있다. 그 다음, 소정 컨텍스트로 복귀할 때 레지스터 값들을 복원할 수 있다. 하나의 실시예에서, 인터럽트 관리 회로(3747)는 시스템 디바이스들로부터 수신된 인터럽트들을 수신하고 처리한다.

[0273] 일 구현에서, 그래픽 처리 엔진(3731)으로부터의 가상/유효 어드레스들은 MMU(3739)에 의해 시스템 메모리(3714) 내의 실제/물리적 어드레스들로 변환된다. 가속기 통합 회로(3736)의 하나의 실시예는 다수의(예를 들어, 4, 8, 16) 그래픽 가속기 모듈들(3746) 및/또는 다른 가속기 디바이스들을 지원한다. 그래픽 가속기 모듈(3746)은 프로세서(3707) 상에서 실행되는 단일 애플리케이션에 전용될 수 있거나, 또는 다수의 애플리케이션들 사이에 공유될 수 있다. 하나의 실시예에서, 그래픽 처리 엔진들(3731-3732, N)의 리소스들이 다수의 애플리케이션 또는 가상 머신(VM)과 공유되는 가상화된 그래픽 실행 환경이 제공된다. 적어도 하나의 실시예에서, 리소스는, VM 및/또는 애플리케이션과 연관된 처리 요건 및 우선순위에 기초하여 상이한 VM 및/또는 애플리케이션에 할당되는 "슬라이스"로 세분될 수 있다.

[0274] 적어도 하나의 실시예에서, 가속기 통합 회로(3736)는 그래픽 가속 모듈(3746)을 위한 시스템에 대한 브리지로서 기능을 수행하고 어드레스 변환 및 시스템 메모리 캐시 서비스들을 제공한다. 또한, 가속기 통합 회로(3736)는 호스트 프로세서가 그래픽 처리 엔진들(3731-3732), 인터럽트들, 및 메모리 관리의 가상화를 관리하기 위한 가상화 설비들을 제공할 수 있다.

[0275] 그래픽 처리 엔진들(3731-3732, N)의 하드웨어 리소스들은 호스트 프로세서(3707)가 보는 실제 어드레스 공간에 명시적으로 맵핑되기 때문에, 임의의 호스트 프로세서는 유효 어드레스 값을 사용하여 이러한 리소스들을 직접 어드레싱할 수 있다. 하나의 실시예에서, 가속기 통합 회로(3736)의 하나의 기능은 그래픽 처리 엔진들(3731-3732, N)의 물리적 분리이며, 따라서 이들은 독립적인 유닛들로서 시스템에 나타난다.

[0276] 적어도 하나의 실시예에서, 하나 이상의 그래픽 메모리(3733-3734, M)가 그래픽 처리 엔진들(3731-3732, N) 각각에 각각 결합된다. 그래픽 메모리들(3733-3734, M)은 그래픽 처리 엔진들(3731-3732, N) 각각에 의해 처리되는 명령어들과 데이터를 저장한다. 그래픽 메모리들(3733-3734, M)은 DRAM들(스택형 DRAM들을 포함함), GDDR 메모리(예를 들어, GDDR5, GDDR6) 또는 HBM과 같은 휘발성 메모리들일 수 있고/있거나, 3D XPoint 또는 나노-램과 같은 비휘발성 메모리들일 수 있다.

[0277] 하나의 실시예에서, 링크(3740)를 통한 데이터 트래픽을 감소시키기 위해, 바이어싱 기법들이 그래픽 메모리들(3733-3734, M)에 저장된 데이터가 그래픽 처리 엔진들(3731-3732, N)에 의해 가장 빈번하게 사용되고, 바람직하게는 코어들(3760A-3760D)에 의해 사용되지 않는 (적어도 빈번하지 않음) 데이터인 것을 보장하기 위해 사용된다. 유사하게, 바이어싱 메커니즘은 코어들과 시스템 메모리(3714)의 캐시들(3762A-3762D, 3756) 내의 코어들(및 바람직하게는 그래픽 처리 엔진들(3731-3732, N)이 아님)에 의해 요구되는 데이터를 유지하려고 시도한다.

[0278] 도 37c는 가속기 통합 회로(3736)가 프로세서(3707) 내에 통합되는 또 다른 예시적인 실시예를 예시한다. 이 실시예에서, 그래픽 처리 엔진들(3731-3732, N)은 인터페이스(3737) 및 인터페이스(3735)(이는 다시, 임의의 형태의 버스 또는 인터페이스 프로토콜을 이용할 수 있음)를 통해 고속 링크(3740)를 통해 가속기 통합 회로(3736)와 직접 통신한다. 가속기 통합 회로(3736)는 도 37b와 관련하여 설명된 것들과 동일한 동작들을 수행할 수 있

지만, 일관성 버스(3764) 및 캐시들(3762A-3762D, 3756)에 대한 그 근접성이 주어지면 잠재적으로 더 높은 처리량에서 수행할 수 있다. 하나의 실시예에는 전용 프로세스 프로그래밍 모델(그래픽 가속 모듈 가상화 없음) 및 공유 프로그래밍 모델들(가상화를 가짐)을 포함하는 상이한 프로그래밍 모델들을 지원하며, 공유 프로그래밍 모델들은 가속기 통합 회로(3736)에 의해 제어되는 프로그래밍 모델들 및 그래픽 가속 모듈(3746)에 의해 제어되는 프로그래밍 모델들을 포함할 수 있다.

[0279] 적어도 하나의 실시예에서, 그래픽 처리 엔진들(3731-3732, N)은 단일 운영 체제 하에서 단일 애플리케이션 또는 프로세스에 전용된다. 적어도 하나의 실시예에서, 단일 애플리케이션이 다른 애플리케이션 요청을 그래픽 처리 엔진(3731 내지 3732, N)으로 보내어, VM/파티션 내의 가상화를 제공할 수 있다.

[0280] 적어도 하나의 실시예에서, 그래픽 처리 엔진들(3731-3732, N)은 다수의 VM/애플리케이션 파티션에 의해 공유될 수 있다. 적어도 하나의 실시예에서, 공유된 모델은 시스템 하이퍼바이저를 이용하여 그래픽 처리 엔진(3731-3732, N)을 가상화해 각각의 운영 체제에 의한 액세스를 허용할 수 있다. 하이퍼바이저가 없는 단일-파티션 시스템의 경우, 그래픽 처리 엔진(3731-3732, N)은 운영 체제에 의해 소유된다. 적어도 하나의 실시예에서, 운영 체제는 그래픽 처리 엔진들(3731-3732, N)을 가상화하여 각각의 프로세스 또는 애플리케이션에 대한 액세스를 제공할 수 있다.

[0281] 적어도 하나의 실시예에서, 그래픽 가속 모듈(3746) 또는 개별 그래픽 처리 엔진(3731-3732, N)은 프로세스 핸들을 이용하여 프로세스 엘리먼트를 선택한다. 하나의 실시예에서, 프로세스 엘리먼트들은 시스템 메모리(3714)에 저장되고, 본 명세서에 설명된 유효 어드레스 대 실제 어드레스 변환 기술들을 이용하여 어드레싱 가능하다. 적어도 하나의 실시예에서, 프로세스 핸들은 그래픽 처리 엔진(3731-3732, N)(즉, 프로세스 엘리먼트를 프로세스 엘리먼트 연결 목록에 추가하기 위해 시스템 소프트웨어를 호출함)에 그의 컨텍스트를 등록할 때 호스트 프로세스에 제공되는 구현 특정 값일 수 있다. 적어도 하나의 실시예에서, 프로세스 핸들의 하위 16-비트는 프로세스 엘리먼트 연결 목록 내의 프로세스 엘리먼트의 오프셋일 수 있다.

[0282] 도 37d는 예시적인 가속기 통합 슬라이스(3790)를 예시한다. 본 명세서에서 사용될 때, "슬라이스"는 가속기 통합 회로(3736)의 처리 리소스들의 지정된 부분을 포함한다. 시스템 메모리(3714) 내의 애플리케이션 유효 어드레스 공간(3782)은 프로세스 엘리먼트들(3783)을 저장한다. 하나의 실시예에서, 프로세스 엘리먼트들(3783)은 프로세서(3707) 상에서 실행되는 애플리케이션들(3780)로부터의 GPU 호출들(3781)에 응답하여 저장된다. 프로세스 엘리먼트(3783)는 대응하는 애플리케이션(3780)에 대한 프로세스 상태를 포함한다. 프로세스 엘리먼트(3783)에 포함된 작업 기술자(WD)(3784)는 애플리케이션에 의해 요청된 단일 잡일 수 있거나 잡들의 큐에 대한 포인터를 포함할 수 있다. 적어도 하나의 실시예에서, WD(3784)는 애플리케이션의 어드레스 공간(3782)에서의 잡 요청 큐에 대한 포인터이다.

[0283] 그래픽 가속 모듈(3746) 및/또는 개별 그래픽 처리 엔진들(3731-3732, N)은 시스템 내의 프로세스들의 전부 또는 서브세트에 의해 공유될 수 있다. 적어도 하나의 실시예에서, 가상화된 환경에서 잡을 시작하기 위해 프로세스 상태를 설정하고 WD(3784)를 그래픽 가속 모듈(3746)에 전송하기 위한 인프라스트럭처가 포함될 수 있다.

[0284] 적어도 하나의 실시예에서, 전용 프로세스 프로그래밍 모델은 구현 특정적이다. 이 모델에서, 단일 프로세스는 그래픽 가속 모듈(3746) 또는 개별 그래픽 처리 엔진(3731)을 소유한다. 그래픽 가속 모듈(3746)은 단일 프로세스에 의해 소유되기 때문에, 하이퍼바이저는 소유 파티션에 대해 가속기 통합 회로(3736)를 초기화하고 운영 체제는 그래픽 가속 모듈(3746)이 할당될 때 소유 프로세스에 대해 가속기 통합 회로(3736)를 초기화한다.

[0285] 동작 시에, 가속기 통합 슬라이스(3790) 내의 WD 인출 유닛(3791)은 그래픽 가속 모듈(3746)의 하나 이상의 그래픽 처리 엔진에 의해 행해질 작업의 표시를 포함하는 다음 WD(3784)를 인출한다. WD(3784)로부터의 데이터는 레지스터들(3745)에 저장되고, 예시된 바와 같이 MMU(3739), 인터럽트 관리 회로(3747) 및/또는 컨텍스트 관리 회로(3748)에 의해 이용될 수 있다. 예를 들어, MMU(3739)의 하나의 실시예는 OS 가상 어드레스 공간(3785) 내의 세그먼트/페이지 테이블들(3786)에 액세스하기 위한 세그먼트/페이지 워크 회로(page walk circuitry)를 포함한다. 인터럽트 관리 회로(3747)는 그래픽 가속 모듈(3746)로부터 수신된 인터럽트 이벤트들(3792)을 처리할 수 있다. 그래픽 연산을 수행할 때, 그래픽 처리 엔진(3731-3732, N)에 의해 생성된 유효 어드레스(3793)가 MMU(3739)에 의해 실제 어드레스로 변환된다.

[0286] 하나의 실시예에서, 레지스터들(3745)의 동일한 세트가 각각의 그래픽 처리 엔진(3731-3732, N) 및/또는 그래픽 가속 모듈(3746)에 대해 복제되고, 하이퍼바이저 또는 운영 체제에 의해 초기화될 수 있다. 이들 복제된 레지스터들 각각은 가속기 통합 슬라이스(3790)에 포함될 수 있다. 하이퍼바이저에 의해 초기화될 수 있는 예시적인

레지스터들이 표 1에 나타나 있다.

표 1

표 1 - 하이퍼바이저 초기화 레지스터들

1	슬라이스 제어 레지스터
2	실제 어드레스(RA) 스케줄링된 프로세스들 영역 포인터
3	권한 마스크 오버라이드 레지스터
4	인터럽트 벡터 테이블 엔트리 오프셋
5	인터럽트 벡터 테이블 엔트리 한계
6	상태 레지스터
7	논리적 파티션 ID
8	실제 어드레스(RA) 하이퍼바이저 가속기 이용 레코드 포인터
9	저장소 설명 레지스터

[0287]

[0288] 운영 체제에 의해 초기화될 수 있는 예시적인 레지스터들이 표 2에 나타나 있다.

표 2

표 2 - 운영 체제 초기화된 레지스터들

1	프로세스 및 스레드 식별
2	유효 어드레스(EA) 컨텍스트 저장/복원 포인터
3	가상 어드레스(VA) 가속기 이용 레코드 포인터
4	가상 어드레스(VA) 저장소 세그먼트 테이블 포인터
5	권한 마스크
6	작업 기술자

[0289]

[0290] 하나의 실시예에서, 각각의 WD(3784)는 특정 그래픽 가속 모듈(3746) 및/또는 그래픽 처리 엔진들(3731-3732, N)에 특정적이다. 이는 그래픽 처리 엔진(3731-3732, N)이 작업을 수행하기 위해 요구되는 모든 정보를 포함하거나 완료될 작업의 커맨드 큐를 애플리케이션이 설정한 메모리 위치에 대한 포인터일 수 있다.

[0291] 도 37e는 공유 모델의 하나의 예시적인 실시예에 대한 추가적인 상세사항을 예시한다. 이 실시예는 프로세스 엘리먼트 목록(3799)이 저장되는 하이퍼바이저 실제 어드레스 공간(3798)을 포함한다. 하이퍼바이저 실제 어드레스 공간(3798)은 운영 체제(3795)를 위한 그래픽 가속 모듈 엔진을 가상화하는 하이퍼바이저(3796)를 통해 액세스 가능하다.

[0292] 적어도 하나의 실시예에서, 공유 프로그래밍 모델들은 시스템 내의 파티션들의 전부 또는 서브세트로부터의 프

로세스들의 전부 또는 서브세트가 그래픽 가속 모듈(3746)을 이용하는 것을 가능하게 한다. 그래픽 가속 모듈(3746)이 다수의 프로세스들 및 파티션들에 의해 공유되는 2개의 프로그래밍 모델들이 있다: 시간-슬라이스 공유 및 그래픽 지시 공유.

[0293] 이 모델에서, 시스템 하이퍼바이저(3796)는 그래픽 가속 모듈(3746)을 소유하고 그의 기능을 모든 운영 체제들(3795)에 이용 가능하게 만든다. 그래픽 가속 모듈(3746)이 시스템 하이퍼바이저(3796)에 의한 가상화를 지원하기 위해, 그래픽 가속 모듈(3746)은 다음을 준수할 수 있다: 1) 애플리케이션의 잡 요청은 자율적이어야 하거나(즉, 잡들 사이에 상태가 유지될 필요가 없거나), 그래픽 가속 모듈(3746)은 컨텍스트 저장 및 복원 메커니즘을 제공해야 한다. 2) 애플리케이션의 잡 요청은 그래픽 가속 모듈(3746)에 의해 임의의 변환 결함들을 포함하는 지정된 시간량 내에 완료되는 것이 보장되거나, 그래픽 가속 모듈(3746)은 잡의 처리를 선점하는 능력을 제공한다. 3) 그래픽 가속 모듈(3746)은 지시 공유 프로그래밍 모델에서 동작할 때 프로세스들 사이의 공정성을 보장해야 한다.

[0294] 적어도 하나의 실시예에서, 애플리케이션(3780)은 그래픽 가속 모듈(3746) 타입, 작업 기술자(WD), 권한 마스크 레지스터(AMR) 값, 및 컨텍스트 저장/복원 영역 포인터(CSRP)로 운영 체제(3795) 시스템 호출을 하도록 요구된다. 적어도 하나의 실시예에서, 그래픽 가속 모듈(3746) 타입은 시스템 호출에 대한 타겟화된 가속 기능을 설명한다. 적어도 하나의 실시예에서, 그래픽 가속 모듈(3746) 타입은 시스템 특정 값일 수 있다. 적어도 하나의 실시예에서, WD는 그래픽 가속 모듈(3746)에 대해 구체적으로 포맷팅되며, 그래픽 가속 모듈(3746) 커맨드, 사용자-정의 구조에 대한 유효 어드레스 포인터, 커맨드들의 큐에 대한 유효 어드레스 포인터, 또는 그래픽 가속 모듈(3746)에 의해 수행될 작업을 설명하기 위한 임의의 다른 데이터 구조의 형태일 수 있다. 하나의 실시예에서, AMR 값은, 현재 프로세스에 이용하기 위한 AMR 상태이다. 적어도 하나의 실시예에서, 운영 체제에 전달된 값은 애플리케이션 설정 AMR과 유사하다. 가속기 통합 회로(3736) 및 그래픽 가속 모듈(3746) 구현이 UAMOR(User Authority Mask Override Register)을 지원하지 않는다면, 운영 체제는 하이퍼바이저 호출에서 AMR을 전달하기 전에 현재의 UAMOR 값을 AMR 값에 적용할 수 있다. 하이퍼바이저(3796)는 AMR을 프로세스 엘리먼트(3783)에 배치하기 전에 현재의 권한 마스크 오버라이드 레지스터(AMOR) 값을 선택적으로 적용할 수 있다. 적어도 하나의 실시예에서, CSRP는 그래픽 가속 모듈(3746)이 컨텍스트 상태를 저장하고 복원하기 위한 애플리케이션의 어드레스 공간(3782) 내의 영역의 유효 어드레스를 포함하는 레지스터들(3745) 중 하나이다. 이 포인터는, 잡들 사이에서 상태가 저장될 것이 요구되지 않거나 잡이 선점될 때 선택적이다. 적어도 하나의 실시예에서, 컨텍스트 저장/복원 영역은 고정된 시스템 메모리일 수 있다.

[0295] 시스템 호출을 수신하면, 운영 체제(3795)는 애플리케이션(3780)이 등록되었고 그래픽 가속 모듈(3746)을 사용할 권한을 부여받았음을 검증할 수 있다. 운영 체제(3795)는 이어서 표 3에 나타난 정보로 하이퍼바이저(3796)를 호출한다.

표 3

표 3 - OS-대-하이퍼바이저 호출 파라미터들

1	작업 기술자(WD)
2	권한 마스크 레지스터(AMR) 값(잠재적으로 마스킹됨)
3	유효 어드레스(EA) 컨텍스트 저장/복원 영역 포인터(CSRP)
4	프로세스 ID(PID) 및 선택적 스레드 ID(TID)
5	가상 어드레스(VA) 가속기 이용 레코드 포인터(AURP)
6	저장 세그먼트 테이블 포인터(SSTP)의 가상 어드레스
7	논리적 인터럽트 서비스 번호(LISN)

[0296]

[0297] 하이퍼바이저 호출을 수신하면, 하이퍼바이저(3796)는 운영 체제(3795)가 등록되었고 그래픽 가속 모듈(3746)을 사용할 권한을 부여받았음을 검증한다. 그 후 하이퍼바이저(3796)는 프로세스 엘리먼트(3783)를 대응하는 그래픽 가속 모듈(3746) 타입에 대한 프로세스 엘리먼트 연결 목록에 넣는다. 프로세스 엘리먼트는 표 4에 나타난 정보를 포함할 수 있다.

표 4

표 4 - 프로세스 엘리먼트 정보

1	작업 기술자(WD)
2	권한 마스크 레지스터(AMR) 값(잠재적으로 마스크됨).
3	유효 어드레스(EA) 컨텍스트 저장/복원 영역 포인터(CSRP)
4	프로세스 ID(PID) 및 선택적 스레드 ID(TID)
5	가상 어드레스(VA) 가속기 이용 레코드 포인터(AURP)
6	저장 세그먼트 테이블 포인터(SSTP)의 가상 어드레스
7	논리적 인터럽트 서비스 번호(LISN)
8	하이퍼바이저 호출 파라미터들로부터 도출된 인터럽트 벡터 테이블
9	상태 레지스터(SR) 값
10	논리적 파티션 ID(LPID)
11	실제 어드레스(RA) 하이퍼바이저 가속기 이용 레코드 포인터
12	저장소 기술자 레지스터(SDR)

[0298]

[0299] 적어도 하나의 실시예에서, 하이퍼바이저는 복수의 가속기 통합 슬라이스(3790) 레지스터(3745)를 초기화한다.

[0300] 도 37f에 예시된 바와 같이, 적어도 하나의 실시예에서, 물리 프로세서 메모리들(3701-3702) 및 GPU 메모리들(3720-3723)에 액세스하는 데 사용되는 공통 가상 메모리 어드레스 공간을 통해 어드레싱 가능한 통합 메모리가 사용된다. 이 구현에서, GPU들(3710-3713) 상에서 실행되는 동작들은 동일한 가상/유효 메모리 어드레스 공간을 이용하여 프로세서 메모리들(3701-3702)에 액세스하고, 그 반대도 가능하며, 이에 의해 프로그램 가능성을 단순화한다. 하나의 실시예에서, 가상/유효 어드레스 공간의 제1 부분은 프로세서 메모리(3701)에 할당되고, 제2 부분은 제2 프로세서 메모리(3702)에 할당되고, 제3 부분은 GPU 메모리(3720)에 할당되며, 기타 등등이다. 적어도 하나의 실시예에서, 전체 가상/유효 메모리 공간(때때로 유효 어드레스 공간이라고 지칭됨)이 이에 의해 프로세서 메모리들(3701-3702) 및 GPU 메모리들(3720-3723) 각각에 걸쳐 분산되어, 임의의 프로세서 또는 GPU가 그 메모리에 맵핑된 가상 어드레스를 갖는 임의의 물리 메모리에 액세스할 수 있게 한다.

[0301] 하나의 실시예에서, MMU들(3739A-3739E) 중 하나 이상 내의 바이어스/일관성 관리 회로(3794A-3794E)는 하나 이상의 호스트 프로세서(예를 들어, 3705)의 캐시들과 GPU들(3710-3713) 사이의 캐시 일관성을 보장하고, 특정 타입들의 데이터가 저장되어야 하는 물리적 메모리들을 나타내는 바이어싱 기법들을 구현한다. 바이어스/일관성 관리 회로(3794A-3794E)의 다수의 인스턴스가 도 37f에 예시되어 있지만, 바이어스/일관성 회로는 하나 이상의 호스트 프로세서(3705)의 MMU 내에 및/또는 가속기 통합 회로(3736) 내에 구현될 수 있다.

[0302] 하나의 실시예는 GPU-부착 메모리(3720-3723)가 시스템 메모리의 일부로서 맵핑되고, 공유된 가상 메모리(SVM)

기술을 사용하여 액세스되는 것을 허용하지만, 전체 시스템 캐시 일관성과 연관된 성능 단점들이 발생하지 않는다. 적어도 하나의 실시예에서, GPU-부착 메모리(3720-3723)가 부담스러운 캐시 일관성 오버헤드 없이 시스템 메모리로서 액세스되는 능력은 GPU 오프로드를 위한 유익한 동작 환경을 제공한다. 이 배열은 호스트 프로세서(3705) 소프트웨어가, 전통적인 I/O DMA 데이터 사본의 오버헤드 없이, 피연산자를 설정하고 계산 결과에 액세스할 수 있게 한다. 이러한 전통적인 사본들은, 모두 단순 메모리 액세스에 비해 비효율적인, 드라이버 호출들, 인터럽트들 및 메모리 맵핑된 I/O(MMIO) 액세스들을 수반한다. 적어도 하나의 실시예에서, 캐시 일관성 오버헤드 없이 GPU 부착 메모리(3720-3723)에 액세스하는 능력은 오프로드된 계산의 실행 시간에 중요할 수 있다. 상당한 스트리밍 기입 메모리 트래픽의 경우에, 예를 들어, 캐시 일관성 오버헤드는 GPU(3710-3713)가 보는 유효 기입 대역폭을 상당히 감소시킬 수 있다. 적어도 하나의 실시예에서, 피연산자 설정의 효율성, 결과 액세스의 효율성, 및 GPU 계산의 효율성은, GPU 오프로드의 유효성을 결정하는데 있어서 일조할 수 있다.

[0303] 적어도 하나의 실시예에서, GPU 바이어스 및 호스트 프로세서 바이어스의 선택은, 바이어스 추적기 데이터 구조에 의해 구동된다. 예를 들어, GPU-부착된 메모리 페이지당 1 또는 2 비트를 포함하는 페이지-입상 구조(즉, 메모리 페이지 입도로 제어됨)일 수 있는 바이어스 테이블이 이용될 수 있다. 적어도 하나의 실시예에서, 바이어스 테이블은 (예를 들어, 바이어스 테이블의 자주/최근에 사용된 엔트리들을 캐싱하기 위해) GPU(3710-3713) 내의 바이어스 캐시를 갖거나 갖지 않는 하나 이상의 GPU-부착 메모리(3720-3723)의 도난된 메모리 범위에서 구현될 수 있다. 대안으로서, 전체 바이어스 테이블이 GPU 내에 유지될 수 있다.

[0304] 적어도 하나의 실시예에서, GPU-부착된 메모리(3720-3723)에 대한 각각의 액세스와 연관된 바이어스 테이블 엔트리는 GPU 메모리에 대한 실제 액세스 이전에 액세스되어, 다음의 동작들을 야기한다. 먼저, GPU 바이어스에서 그 자신의 페이지를 찾는 GPU(3710-3713)로부터의 로컬 요청이 대응하는 GPU 메모리(3720-3723)로 직접 포워딩된다. 호스트 바이어스에서 그들의 페이지를 찾는 GPU로부터의 로컬 요청들은 (예를 들어, 전술한 바와 같은 고속 링크를 통해) 프로세서(3705)로 포워딩된다. 하나의 실시예에서, 호스트 프로세서 바이어스에서 요청된 페이지를 찾는 프로세서(3705)로부터의 요청들은 정상 메모리 판독과 같은 요청을 완료한다. 대안적으로, GPU-바이어싱된 페이지로 향하는 요청들은 GPU(3710-3713)로 포워딩될 수 있다. 적어도 하나의 실시예에서, GPU는, 그 후, 페이지를 현재 이용하고 있지 않다면 페이지를 호스트 프로세서 바이어스로 전이할 수 있다. 적어도 하나의 실시예에서, 페이지의 바이어스 상태는, 소프트웨어 기반 메커니즘, 하드웨어 보조 소프트웨어 기반 메커니즘, 또는 제한된 세트의 사례들에 대해, 순수 하드웨어 기반 메커니즘에 의해 변경될 수 있다.

[0305] 바이어스 상태를 변경하기 위한 한 메커니즘은, API 호출(예를 들어, OpenCL)을 이용하며, 이 API 호출은, 차례로, GPU의 디바이스 드라이버를 호출하고, 디바이스 드라이버는 차례로 메시지를 GPU에 전송하여(또는 커맨드 기술자를 인큐잉하여) 바이어스 상태를 변경할 것을 지시하고, 일부 전환에 대해서는, 호스트에서 캐시 플러싱 동작을 수행한다. 적어도 하나의 실시예에서, 캐시 플러싱 연산은 호스트 프로세서(3705) 바이어스로부터 GPU 바이어스로의 전이를 위해 사용되지만, 그러나, 반대 전이를 위한 것은 아니다.

[0306] 하나의 실시예에서, 호스트 프로세서(3705)에 의해 캐시될 수 없는 GPU-바이어싱된 페이지들을 일시적으로 렌더링함으로써 캐시 일관성이 유지된다. 이러한 페이지들에 액세스하기 위해, 프로세서(3705)는 즉시 액세스를 허가하거나 허가하지 않을 수 있는 GPU(3710)로부터의 액세스를 요청할 수 있다. 따라서, 프로세서(3705)와 GPU(3710) 사이의 통신을 감소시키기 위해, GPU-바이어싱된 페이지들이 호스트 프로세서(3705)가 아니라 GPU에 의해 요구되는 것들이며, 그 반대도 성립함을 보장하는 것이 유익하다.

[0307] 하드웨어 구조(들)(2915)는 하나 이상의 실시예를 수행하는데 이용된다. 하드웨어 구조(x)(2915)에 관한 상세는 도 29a 및/또는 도 29b와 관련하여 여기서 제공된다.

[0308] 도 38은 본 명세서에 설명된 다양한 실시예들에 따른, 하나 이상의 IP 코어를 이용하여 제조될 수 있는 예시적인 집적 회로들 및 연관된 그래픽 프로세서들을 예시한다. 예시된 것에 추가로, 추가적인 그래픽 프로세서들/코어들, 주변기기 인터페이스 제어기들 또는 범용 프로세서 코어들을 포함하는 다른 로직 및 회로들이 적어도 하나의 실시예에 포함될 수 있다.

[0309] 도 38은 적어도 하나의 실시예에 따라, 하나 이상의 IP 코어들을 사용하여 제조될 수 있는 예시적인 시스템-온-칩 집적 회로(3800)를 예시하는 블록도이다. 적어도 하나의 실시예에서, 집적 회로(3800)는 하나 이상의 애플리케이션 프로세서(들)(3805)(예를 들어, CPU들), 적어도 하나의 그래픽 프로세서(3810)를 포함하고, 이미지 프로세서(3815) 및/또는 비디오 프로세서(3820)를 더 포함할 수 있으며, 이들 중 임의의 것은 모듈러 IP 코어일 수 있다. 적어도 하나의 실시예에서, 집적 회로(3800)는 USB 제어기(3825), UART 제어기(3830), SPI/SDIO 제어기(3835) 및 I.sup.2S/I.sup.2C 제어기(3840)를 포함하는 주변기기 또는 버스 로직을 포함한다. 적어도 하나의 실

시예에서, 집적 회로(3800)는 고화질 멀티미디어 인터페이스(HDMI) 제어기(3850) 및 모바일 산업 프로세서 인터페이스(MIPI) 디스플레이 인터페이스(3855) 중 하나 이상에 결합된 디스플레이 디바이스(3845)를 포함할 수 있다. 적어도 하나의 실시예에서, 저장소는 플래시 메모리 및 플래시 메모리 제어기를 포함하는 플래시 메모리 서브시스템(3860)에 의해 제공될 수 있다. 적어도 하나의 실시예에서, 메모리 인터페이스는 SDRAM 또는 SRAM 메모리 디바이스들에 대한 액세스를 위해 메모리 제어기(3865)를 통해 제공될 수 있다. 적어도 하나의 실시예에서, 일부 집적 회로들은 임베디드 보안 엔진(3870)을 더 포함한다.

[0310] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 신경망 훈련 연산들, 신경망 기능들 및/또는 아키텍처들, 또는 본 명세서에 설명된 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 연산들을 추론하거나 예측하기 위해 집적 회로(3800)에서 이용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 그래픽 프로세서로 구현된다.

[0311] 도 39a 및 도 39b는 본 명세서에 설명된 다양한 실시예들에 따른, 하나 이상의 IP 코어를 이용하여 제조될 수 있는 예시적인 집적 회로들 및 연관된 그래픽 프로세서들을 예시한다. 예시된 것에 추가로, 추가적인 그래픽 프로세서들/코어들, 주변기기 인터페이스 제어기들 또는 범용 프로세서 코어들을 포함하는 다른 로직 및 회로들이 적어도 하나의 실시예에 포함될 수 있다.

[0312] 도 39a 내지 도 39b는 본 명세서에 설명된 실시예들에 따른, SoC 내에서 사용하기 위한 예시적인 그래픽 프로세서들을 예시하는 블록도들이다. 도 39a는 적어도 하나의 실시예에 따라, 하나 이상의 IP 코어들을 사용하여 제조될 수 있는 시스템-온-칩 집적 회로의 예시적인 그래픽 프로세서(3910)를 예시한다. 도 39b는 적어도 하나의 실시예에 따라, 하나 이상의 IP 코어들을 사용하여 제조될 수 있는 시스템-온-칩 집적 회로의 추가적인 예시적인 그래픽 프로세서(3940)를 예시한다. 적어도 하나의 실시예에서, 도 39a의 그래픽 프로세서(3910)는 저전력 그래픽 프로세서 코어이다. 적어도 하나의 실시예에서, 도 39b의 그래픽 프로세서(3940)는 고성능 그래픽 프로세서 코어이다. 적어도 하나의 실시예에서, 그래픽 프로세서들(3910, 3940) 각각은 도 38의 그래픽 프로세서(3810)의 변형들일 수 있다.

[0313] 적어도 하나의 실시예에서, 그래픽 프로세서(3910)는 정점 프로세서(3905) 및 하나 이상의 프래그먼트 프로세서(들)(3915A-3915N)(예를 들어, 3915A, 3915B, 3915C, 3915D, 내지 3915N-1, 및 3915N)를 포함한다. 적어도 하나의 실시예에서, 그래픽 프로세서(3910)는 별개의 로직을 통해 상이한 셰이더 프로그램들을 실행할 수 있고, 이에 의해 정점 프로세서(3905)는 정점 셰이더 프로그램들에 대한 동작들을 실행하도록 최적화되는 한편, 하나 이상의 프래그먼트 프로세서(들)(3915A-3915N)는 프래그먼트 또는 픽셀 셰이더 프로그램들에 대한 프래그먼트(예를 들어, 픽셀) 셰이딩 동작들을 실행한다. 적어도 하나의 실시예에서, 정점 프로세서(3905)는 3D 그래픽 파이프라인의 정점 처리 스테이지를 수행하고, 프리미티브들 및 정점 데이터를 생성한다. 적어도 하나의 실시예에서, 프래그먼트 프로세서(들)(3915A-3915N)는 정점 프로세서(3905)에 의해 생성된 프리미티브 및 정점 데이터를 이용하여 디스플레이 디바이스 상에 디스플레이되는 프레임버퍼를 생성한다. 적어도 하나의 실시예에서, 프래그먼트 프로세서(들)(3915A-3915N)는, Direct 3D API에서 제공되는 픽셀 셰이더 프로그램과 유사한 연산들을 수행하는데 이용될 수 있는, OpenGL API에서 제공되는 프래그먼트 셰이더 프로그램을 실행하도록 최적화된다.

[0314] 적어도 하나의 실시예에서, 그래픽 프로세서(3910)는, 하나 이상의 메모리 관리 유닛(MMU)(3920A-3920B), 캐시(들)(3925A-3925B), 및 회로 인터커넥트(들)(3930A-3930B)를 더 포함한다. 적어도 하나의 실시예에서, 하나 이상의 MMU(들)(3920A-3920B)는, 하나 이상의 캐시(들)(3925A-3925B)에 저장된 정점 또는 이미지/텍스처 데이터 외에도, 메모리에 저장된 정점 또는 이미지/텍스처 데이터를 참조할 수 있는 정점 프로세서(3905) 및/또는 프래그먼트 프로세서(들)(3915A-3915N)를 포함한, 그래픽 프로세서(3910)에 대한 가상-물리적 어드레스 맵핑을 제공한다. 적어도 하나의 실시예에서, 하나 이상의 MMU(들)(3920A-3920B)는, 도 38의 하나 이상의 애플리케이션 프로세서(들)(3805), 이미지 프로세서(3815), 및/또는 비디오 프로세서(3820)와 연관된 하나 이상의 MMU를 포함한, 시스템 내의 다른 MMU들과 동기화될 수 있으며, 따라서 각각의 프로세서(3805-3820)가 공유된 또는 통합된 가상 메모리 시스템에 참여할 수 있다. 적어도 하나의 실시예에서, 하나 이상의 회로 인터커넥트(들)(3930A-3930B)는 그래픽 프로세서(3910)가 SoC의 내부 버스를 통해 또는 직접 접속을 통해 SoC 내의 다른 IP 코어들과 인터페이스할 수 있게 한다.

[0315] 적어도 하나의 실시예에서, 그래픽 프로세서(3940)는 하나 이상의 MMU(들)(3920A-3920B), 캐시들(3925A-3925B), 및 도 39a의 그래픽 프로세서(3910)의 회로 인터커넥트들(3930A-3930B)을 포함한다. 적어도 하나의 실

시예에서, 그래픽 프로세서(3940)는 하나 이상의 셰이더 코어(들)(3955A-3955N)(예를 들어, 3955A, 3955B, 3955C, 3955D, 3955E, 3955F, 내지 3955N-1, 및 3955N)를 포함하고, 이는 단일 코어 또는 타입 또는 코어가 정점 셰이더들, 프래그먼트 셰이더들, 및/또는 계산 셰이더들을 구현하기 위한 셰이더 프로그램 코드를 포함하여, 모든 타입들의 프로그램가능 셰이더 코드를 실행할 수 있는 통합 셰이더 코어 아키텍처를 제공한다. 적어도 하나의 실시예에서, 셰이더 코어의 수는 달라질 수 있다. 적어도 하나의 실시예에서, 그래픽 프로세서(3940)는 실행 스레드들을 하나 이상의 셰이더 코어(3955A-3955N)에 디스패치하는 스레드 디스패처(thread dispatcher)로서 작용하는 인터-코어 태스크 관리자(3945) 및 타일 기반 렌더링을 위한 타일링 동작들을 가속하는 타일링 유닛(3958)을 포함하고, 여기서 장면에 대한 렌더링 동작들은 예를 들어 장면 내의 로컬 공간 일관성을 활용하거나 내부 캐시들의 사용을 최적화하기 위해 이미지 공간에서 세분된다.

[0316] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 본 명세서에 설명된 신경망 훈련 동작, 신경망 기능 및/또는 아키텍처, 또는 신경망 이용 사례를 이용하여 산출된 가중치 파라미터에 적어도 부분적으로 기초하여 동작을 추론 또는 예측하기 위해 집적 회로(39A 및/또는 39B)에서 이용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 그래픽 프로세서 로직으로 구현된다.

[0317] 도 40a 및 도 40b는 본 명세서에 설명된 실시예에 따른, 추가적인 예시적인 그래픽 프로세서 로직을 예시한다. 도 40a는 적어도 하나의 실시예에서, 도 38의 그래픽 프로세서(3810) 내에 포함될 수 있고, 적어도 하나의 실시예에서 도 39b에서와 같은 통합 셰이더 코어(3955A-3955N)일 수 있는 그래픽 코어(4000)를 예시한다. 도 40b는 적어도 하나의 실시예에서 멀티-칩 모듈 상에 배치하기에 적합한 고도 병렬 범용 그래픽 처리 유닛(4030)을 예시한다.

[0318] 적어도 하나의 실시예에서, 그래픽 코어(4000)는 그래픽 코어(4000) 내의 실행 리소스들에 공통인 공유 명령어 캐시(4002), 텍스처 유닛(4018), 및 캐시/공유된 메모리(4020)를 포함한다. 적어도 하나의 실시예에서, 그래픽 코어(4000)는 각각의 코어에 대해 다수의 슬라이스(4001A-4001N) 또는 파티션을 포함할 수 있고, 그래픽 프로세서는 그래픽 코어(4000)의 다수의 인스턴스를 포함할 수 있다. 슬라이스들(4001A-4001N)은 로컬 명령어 캐시(4004A-4004N), 스레드 스케줄러(4006A-4006N), 스레드 디스패처(4008A-4008N), 및 레지스터들의 세트(4010A-4010N)를 포함하는 지원 로직을 포함할 수 있다. 적어도 하나의 실시예에서, 슬라이스들(4001A-4001N)은, 한 세트의 추가 기능 유닛(AFU들 4012A-4012N), 부동 소수점 유닛(FPU 4014A-4014N), 정수 산술 로직 유닛(ALU들 4016-4016N), 어드레스 계산 유닛(ACU 4013A-4013N), 배정도 부동 소수점 유닛(DPFPU 4015A-4015N), 및 행렬 처리 유닛(MPU 4017A-4017N)을 포함할 수 있다.

[0319] 적어도 하나의 실시예에서, FPU들(4014A-4014N)은 단정도(32-비트) 및 반정도(16-비트) 부동 소수점 연산들을 수행할 수 있는 반면, DPFPU들(4015A-4015N)은 배정도(64-비트) 부동 소수점 연산들을 수행한다. 적어도 하나의 실시예에서, ALU들(4016A-4016N)은 8-비트, 16-비트, 및 32-비트 정밀도에서 가변 정밀도 정수 연산들을 수행할 수 있고, 혼합된 정밀도 연산들을 위해 구성될 수 있다. 적어도 하나의 실시예에서, MPU들(4017A-4017N)은 또한 반정도 부동 소수점 및 8-비트 정수 연산들을 포함하는 혼합된 정밀도 행렬 연산들을 위해 구성될 수 있다. 적어도 하나의 실시예에서, MPU들(4017-4017N)은, 가속된 일반 행렬 대 행렬 곱셈(GEMM)에 대한 지원을 가능하게 하는 것을 포함한, 머신 러닝 애플리케이션 프레임워크를 가속하기 위해 다양한 행렬 연산을 수행할 수 있다. 적어도 하나의 실시예에서, AFU들(4012A-4012N)은 삼각 연산들(예를 들어, 사인, 코사인 등)을 포함하는, 부동 소수점 또는 정수 유닛들에 의해 지원되지 않는 추가적인 로직 연산들을 수행할 수 있다.

[0320] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 본 명세서에 설명되는 신경망 훈련 동작들, 신경망 기능들 및/또는 아키텍처들, 또는 신경망 사용 사례들을 이용하여 산출되는 가중치 파라미터들에, 적어도 부분적으로, 기초하여 동작들을 추론 또는 예측하기 위해 그래픽 코어(4000)에서 사용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 범용 처리 유닛으로 구현된다.

[0321] 도 40b는 적어도 하나의 실시예에서, 그래픽 처리 유닛들의 어레이에 의해 고도 병렬 계산 연산들이 수행될 수 있게 하도록 구성될 수 있는 범용 처리 유닛(GPGPU)(4030)을 예시한 것이다. 적어도 하나의 실시예에서,

GPGPU(4030)는 GPGPU(4030)의 다른 인스턴스들에 직접 링크되어, 심층 신경망들에 대한 훈련 속도를 개선하기 위한 멀티-GPU 클러스터를 생성할 수 있다. 적어도 하나의 실시예에서, GPGPU(4030)는 호스트 프로세서와의 접속을 가능하게 하는 호스트 인터페이스(4032)를 포함한다. 적어도 하나의 실시예에서, 호스트 인터페이스(4032)는 PCI 익스프레스 인터페이스이다. 적어도 하나의 실시예에서, 호스트 인터페이스(4032)는 판매자 특정 통신 인터페이스 또는 통신 패브릭일 수 있다. 적어도 하나의 실시예에서, GPGPU(4030)는 호스트 프로세서로부터 커맨드들을 수신하고, 글로벌 스케줄러(4034)를 사용하여 그 커맨드들과 연관된 실행 스레드들을 계산 클러스터들(4036A-4036H)의 세트에 분배한다. 적어도 하나의 실시예에서, 계산 클러스터(4036A-4036H)는 캐시 메모리(4038)를 공유한다. 적어도 하나의 실시예에서, 캐시 메모리(4038)는 계산 클러스터들(4036A-4036H) 내의 캐시 메모리들에 대한 상위 레벨 캐시의 역할을 할 수 있다.

[0322] 적어도 하나의 실시예에서, GPGPU(4030)는 메모리 제어기들(4042A-4042B)의 세트를 통해 계산 클러스터들(4036A-4036H)과 결합된 메모리(4044A-4044B)를 포함한다. 적어도 하나의 실시예에서, 메모리(4044A-4044B)는 동적 랜덤 액세스 메모리(DRAM) 또는 그래픽 더블 데이터 레이트(GDDR) 메모리를 포함하는 동기식 그래픽 랜덤 액세스 메모리(SGRAM)와 같은 그래픽 랜덤 액세스 메모리를 포함하는 다양한 타입들의 메모리 디바이스들을 포함할 수 있다.

[0323] 적어도 하나의 실시예에서, 계산 클러스터들(4036A-4036H)은 각각 도 40a의 그래픽 코어(4000)와 같은 그래픽 코어들의 세트를 포함하며, 이는 머신 러닝 계산들에 적합한 것을 포함하는 정밀도들의 범위에서 계산 연산들을 수행할 수 있는 다수의 타입의 정수 및 부동 소수점 로직 유닛들을 포함할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 계산 클러스터들(4036A-4036H) 각각 내의 부동 소수점 유닛들의 적어도 서브세트는 16-비트 또는 32-비트 부동 소수점 연산들을 수행하도록 구성될 수 있으며, 한편 부동 소수점 유닛들의 상이한 서브세트는 64-비트 부동 소수점 연산들을 수행하도록 구성될 수 있다.

[0324] 적어도 하나의 실시예에서, GPGPU(4030)의 다수의 인스턴스들은 계산 클러스터로서 동작하도록 구성될 수 있다. 적어도 하나의 실시예에서, 동기화 및 데이터 교환을 위해 계산 클러스터들(4036A-4036H)에 의해 사용되는 통신은 실시예들에 걸쳐 달라진다. 적어도 하나의 실시예에서, GPGPU(4030)의 다수의 인스턴스는 호스트 인터페이스(4032)를 통해 통신한다. 적어도 하나의 실시예에서, GPGPU(4030)는 GPGPU(4030)의 다른 인스턴스들에의 직접 접속을 가능하게 하는 GPU 링크(4040)와 GPGPU(4030)를 결합하는 I/O 허브(4039)를 포함한다. 적어도 하나의 실시예에서, GPU 링크(4040)는 GPGPU(4030)의 다수의 인스턴스들 사이의 통신 및 동기화를 가능하게 하는 전용 GPU-대-GPU 브리지에 결합된다. 적어도 하나의 실시예에서, GPU 링크(4040)는 다른 GPGPU들 또는 병렬 프로세서들에 데이터를 송신 및 수신하기 위해 고속 인터커넥트와 결합된다. 적어도 하나의 실시예에서, GPGPU(4030)의 다수의 인스턴스는 별개의 데이터 처리 시스템에 위치하고 호스트 인터페이스(4032)를 통해 액세스가능한 네트워크 디바이스를 통해 통신한다. 적어도 하나의 실시예에서, GPU 링크(4040)는 호스트 인터페이스(4032)에 추가로 또는 그에 대한 대안으로서 호스트 프로세서에 대한 접속을 가능하게 하도록 구성될 수 있다.

[0325] 적어도 하나의 실시예에서, GPGPU(4030)는 신경망들을 훈련하도록 구성될 수 있다. 적어도 하나의 실시예에서, GPGPU(4030)는 추론 플랫폼 내에서 사용될 수 있다. GPGPU(4030)가 추론을 위해 이용되는 적어도 하나의 실시예에서, GPGPU는, GPGPU가 신경망을 훈련시키는데 이용되는 때에 비해 더 적은 수의 계산 클러스터(4036A-4036H)를 포함할 수 있다. 적어도 하나의 실시예에서, 메모리(4044A-4044B)와 연관된 메모리 기술은 추론 및 훈련 구성들 사이에서 상이할 수 있고, 더 높은 대역폭 메모리 기술들은 훈련 구성들에 전용된다. 적어도 하나의 실시예에서, GPGPU(4030)의 구성을 추론하는 것은 특정 명령어들을 추론하는 것을 지원할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 추론 구성은, 배치된 신경망들에 대한 추론 연산들 동안 이용될 수 있는 하나 이상의 8-비트 정수 내적 명령어들에 대한 지원을 제공할 수 있다.

[0326] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은 본 명세서에 설명된 신경망 훈련 동작들, 신경망 기능들 및/또는 아키텍처들, 또는 신경망 사용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 동작들을 추론 또는 예측하기 위해 GPGPU(4030)에서 사용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 컴퓨터 시스템으로 구현된다.

[0327] 도 41은 적어도 하나의 실시예에 따른 계산 시스템(4100)을 예시하는 블록도이다. 적어도 하나의 실시예에서, 계산 시스템(4100)은 메모리 허브(4105)를 포함할 수 있는 상호접속 경로를 통해 통신하는 하나 이상의 프로세서(들)(4102) 및 시스템 메모리(4104)를 갖는 처리 서브시스템(4101)을 포함한다. 적어도 하나의 실시예에서,

메모리 허브(4105)는 칩셋 컴포넌트 내의 별개의 컴포넌트일 수 있거나, 하나 이상의 프로세서(들)(4102) 내에 통합될 수 있다. 적어도 하나의 실시예에서, 메모리 허브(4105)는 통신 링크(4106)를 통해 I/O 서브시스템(4111)과 결합된다. 적어도 하나의 실시예에서, I/O 서브시스템(4111)은 계산 시스템(4100)이 하나 이상의 입력 디바이스(들)(4108)로부터 입력을 수신할 수 있게 할 수 있는 I/O 허브(4107)를 포함한다. 적어도 하나의 실시예에서, I/O 허브(4107)는 하나 이상의 프로세서(들)(4102)에 포함될 수 있는 디스플레이 제어기가 하나 이상의 디스플레이 디바이스(들)(4110A)에 출력들을 제공할 수 있게 할 수 있다. 적어도 하나의 실시예에서, I/O 허브(4107)와 결합된 하나 이상의 디스플레이 디바이스(들)(4110A)는 로컬, 내부, 또는 임베디드 디스플레이 디바이스를 포함할 수 있다.

[0328] 적어도 하나의 실시예에서, 처리 서브시스템(4101)은 버스 또는 다른 통신 링크(4113)를 통해 메모리 허브(4105)에 결합된 하나 이상의 병렬 프로세서(들)(4112)를 포함한다. 적어도 하나의 실시예에서, 통신 링크(4113)는 PCI 익스프레스와 같은, 그러나 이에 제한되지 않는 임의의 수의 표준 기반 통신 링크 기술들 또는 프로토콜들 중 하나일 수 있거나, 판매자 특정 통신 인터페이스 또는 통신 패브릭일 수 있다. 적어도 하나의 실시예에서, 하나 이상의 병렬 프로세서(들)(4112)는, MIC(many integrated core) 프로세서와 같은, 많은 수의 처리 코어들 및/또는 처리 클러스터들을 포함할 수 있는 계산적으로 집중된 병렬 또는 벡터 처리 시스템을 형성한다. 적어도 하나의 실시예에서, 하나 이상의 병렬 프로세서(들)(4112)는 I/O 허브(4107)를 통해 결합된 하나 이상의 디스플레이 디바이스(들)(4110A) 중 하나에 픽셀들을 출력할 수 있는 그래픽 처리 서브시스템을 형성한다. 적어도 하나의 실시예에서, 하나 이상의 병렬 프로세서(들)(4112)는 또한 하나 이상의 디스플레이 디바이스(들)(4110B)에 대한 직접 접속을 가능하게 하기 위해 디스플레이 제어기 및 디스플레이 인터페이스(도시되지 않음)를 포함할 수 있다.

[0329] 적어도 하나의 실시예에서, 시스템 저장소 유닛(4114)은 계산 시스템(4100)을 위한 저장소 메커니즘을 제공하기 위해 I/O 허브(4107)에 접속할 수 있다. 적어도 하나의 실시예에서, I/O 스위치(4116)는 I/O 허브(4107)와, 플랫폼에 통합될 수 있는 네트워크 어댑터(4118) 및/또는 무선 네트워크 어댑터(4119)와 같은 다른 컴포넌트들, 및 하나 이상의 애드-인 디바이스(들)(4120)를 통해 추가될 수 있는 다양한 다른 디바이스들 사이의 접속들을 가능하게 하는 인터페이스 메커니즘을 제공하기 위해 사용될 수 있다. 적어도 하나의 실시예에서, 네트워크 어댑터(4118)는 이더넷 어댑터 또는 다른 유선 네트워크 어댑터일 수 있다. 적어도 하나의 실시예에서, 무선 네트워크 어댑터(4119)는 Wi-Fi, 블루투스, 근접장 통신(NFC), 또는 하나 이상의 무선 라디오를 포함하는 다른 네트워크 디바이스 중 하나 이상을 포함할 수 있다.

[0330] 적어도 하나의 실시예에서, 계산 시스템(4100)은 또한 I/O 허브(4107)에 접속될 수 있는 USB 또는 다른 포트 접속들, 광학 저장 드라이브들, 비디오 포착 디바이스들 등을 포함하는, 명시적으로 도시되지 않은 다른 컴포넌트들을 포함할 수 있다. 적어도 하나의 실시예에서, 도 41의 다양한 컴포넌트들을 상호접속하는 통신 경로들은 PCI(Peripheral Component Interconnect) 기반 프로토콜들(예를 들어, PCI-Express)과 같은 임의의 적합한 프로토콜들, 또는 NV-링크 고속 인터넥트와 같은 다른 버스 또는 포인트-투-포인트 통신 인터페이스들 및/또는 프로토콜(들), 또는 상호접속 프로토콜들을 이용하여 구현될 수 있다.

[0331] 적어도 하나의 실시예에서, 하나 이상의 병렬 프로세서(들)(4112)는, 예를 들어, 비디오 출력 회로를 통합하는, 그래픽 및 비디오 처리에 최적화된 회로를 포함하고, 그래픽 처리 유닛(GPU)을 구성한다. 적어도 하나의 실시예에서, 하나 이상의 병렬 프로세서(들)(4112)는 범용 처리를 위해 최적화된 회로를 통합한다. 적어도 하나의 실시예에서, 계산 시스템(4100)의 컴포넌트들은 단일 집적 회로 상의 하나 이상의 다른 시스템 엘리먼트들과 통합될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 하나 이상의 병렬 프로세서(들)(4112), 메모리 허브(4105), 프로세서(들)(4102), 및 I/O 허브(4107)는 시스템-온-칩(SoC) 집적 회로에 통합될 수 있다. 적어도 하나의 실시예에서, 계산 시스템(4100)의 컴포넌트들은 단일 패키지로 통합되어 시스템 인 패키지(SIP) 구성을 형성할 수 있다. 적어도 하나의 실시예에서, 계산 시스템(4100)의 컴포넌트들의 적어도 일부는 멀티-칩 모듈(MCM)로 통합될 수 있으며, 이는 다른 멀티-칩 모듈들과 모듈러 계산 시스템 내에 상호접속될 수 있다.

[0332] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 신경망 훈련 연산들, 신경망 기능들 및/또는 아키텍처들, 또는 본 명세서에 설명된 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 연산들을 추론하거나 예측하기 위해 도 4100의 시스템에서 이용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 병렬 프로세서로 구현된다.

- [0333] 프로세서들
- [0334] 도 42a는 적어도 하나의 실시예에 따른 병렬 프로세서(4200)를 예시한다. 적어도 하나의 실시예에서, 병렬 프로세서(4200)의 다양한 컴포넌트들은 하나 이상의 집적 회로 디바이스, 예컨대 프로그램가능 프로세서들, 주문형 집적 회로들(ASIC들), 또는 필드 프로그램가능 게이트 어레이들(FPGA)을 사용하여 구현될 수 있다. 적어도 하나의 실시예에서, 예시된 병렬 프로세서(4200)는 예시적인 실시예에 따른 도 41에 도시된 하나 이상의 병렬 프로세서(들)(4112)의 변형이다.
- [0335] 적어도 하나의 실시예에서, 병렬 프로세서(4200)는 병렬 처리 유닛(4202)을 포함한다. 적어도 하나의 실시예에서, 병렬 처리 유닛(4202)은 병렬 처리 유닛(4202)의 다른 인스턴스들을 포함하는 다른 디바이스들과의 통신을 가능하게 하는 I/O 유닛(4204)을 포함한다. 적어도 하나의 실시예에서, I/O 유닛(4204)은 다른 디바이스들에 직접 접속될 수 있다. 적어도 하나의 실시예에서, I/O 유닛(4204)은 메모리 허브(4105)와 같은 허브 또는 스위치 인터페이스의 사용을 통해 다른 디바이스들과 접속한다. 적어도 하나의 실시예에서, 메모리 허브(4105)와 I/O 유닛(4204) 사이의 접속들은 통신 링크(4113)를 형성한다. 적어도 하나의 실시예에서, I/O 유닛(4204)은 호스트 인터페이스(4206) 및 메모리 크로스바(4216)와 접속하고, 여기서 호스트 인터페이스(4206)는 처리 연산들을 수행하는 것에 관한 커맨드들을 수신하고, 메모리 크로스바(4216)는 메모리 연산들을 수행하는 것에 관한 커맨드들을 수신한다.
- [0336] 적어도 하나의 실시예에서, 호스트 인터페이스(4206)가 I/O 유닛(4204)을 통해 커맨드 버퍼를 수신할 때, 호스트 인터페이스(4206)는 이러한 커맨드들을 수행하는 작업 동작들을 프론트엔드(4208)에 지시할 수 있다. 적어도 하나의 실시예에서, 프론트엔드(4208)는 커맨드들 또는 다른 작업 항목들을 처리 클러스터 어레이(4212)에 분배하도록 구성되어 있는 스케줄러(4210)와 결합된다. 적어도 하나의 실시예에서, 스케줄러(4210)는, 태스크들이 처리 클러스터 어레이(4212)의 처리 클러스터 어레이(4212)에 분배되기 전에 처리 클러스터 어레이(4212)가 적절하게 구성되고 유효한 상태에 있는 것을 보장한다. 적어도 하나의 실시예에서, 스케줄러(4210)는 마이크로컨트롤러 상에서 실행되는 펌웨어 로직을 통해 구현된다. 적어도 하나의 실시예에서, 마이크로컨트롤러 구현 스케줄러(4210)는 대략적 및 미세 입도로 복잡한 스케줄링 및 작업 분배 동작들을 수행하도록 구성가능하여, 처리 어레이(4212) 상에서 실행되는 스레드들의 신속한 선점 및 컨텍스트 스위칭을 가능하게 한다. 적어도 하나의 실시예에서, 호스트 소프트웨어는 다수의 그래픽 처리 도어벨들 중 하나를 통해 처리 어레이(4212) 상에 스케줄링하기 위한 작업부하들을 증명할 수 있다. 적어도 하나의 실시예에서, 작업부하는 이어서 스케줄러(4210)를 포함하는 마이크로컨트롤러 내의 스케줄러(4210) 로직에 의해 처리 어레이(4212)에 걸쳐 자동으로 분산될 수 있다.
- [0337] 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)는 최대 "N"개의 처리 클러스터(예를 들어, 클러스터(4214A), 클러스터(4214B), 내지 클러스터(4214N))를 포함할 수 있다. 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)의 각각의 클러스터(4214A-4214N)는 많은 수의 동시 스레드들을 실행할 수 있다. 적어도 하나의 실시예에서, 스케줄러(4210)는 다양한 스케줄링 및/또는 작업 분배 알고리즘들을 사용하여 처리 클러스터 어레이(4212)의 클러스터들(4214A-4214N)에 작업을 할당할 수 있는데, 이는 각각의 타입의 프로그램 또는 계산에 대해 발생하는 작업부하에 따라 달라질 수 있다. 적어도 하나의 실시예에서, 스케줄링은 스케줄러(4210)에 의해 동적으로 핸들링될 수 있거나, 또는 처리 클러스터 어레이(4212)에 의한 실행을 위해 구성된 프로그램 로직의 컴파일 동안 컴파일러 로직에 의해 부분적으로 보조될 수 있다. 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)의 상이한 클러스터들(4214A-4214N)이 상이한 타입들의 프로그램들을 처리하기 위해 또는 상이한 타입들의 계산들을 수행하기 위해 할당될 수 있다.
- [0338] 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)는 다양한 타입들의 병렬 처리 동작들을 수행하도록 구성될 수 있다. 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)는 범용 병렬 계산 연산들을 수행하도록 구성된다. 예를 들어, 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)는 비디오 및/또는 오디오 데이터의 필터링, 물리 동작들을 포함하는 모델링 동작들의 수행, 및 데이터 변환들의 수행을 포함하는 처리 태스크들을 실행하는 로직을 포함할 수 있다.
- [0339] 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)는 병렬 그래픽 처리 연산들을 수행하도록 구성된다. 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)는, 텍스처 연산들을 수행하는 텍스처 샘플링 로직뿐만 아니라 테셀레이션 로직 및 다른 정점 처리 로직을 포함하지만 이에 제한되지 않는, 이러한 그래픽 처리 연산들의 실행을 지원하는 추가 로직을 포함할 수 있다. 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)는 정점 셰이더들, 테셀레이션 셰이더들, 기하형상 셰이더들 및 픽셀 셰이더들과 같은, 그러나 이에 제한되지 않는 그래픽 처리 관련 셰이더 프로그램들을 실행하도록 구성될 수 있다. 적어도 하나의 실시예에서, 병렬 처리 유닛

(4202)은 처리를 위해 I/O 유닛(4204)을 통해 시스템 메모리로부터 데이터를 전달할 수 있다. 적어도 하나의 실시예에서, 처리 동안, 전달된 데이터는 처리 동안 온-칩 메모리(예를 들어, 병렬 프로세서 메모리(4222))에 저장된 다음, 시스템 메모리에 다시 기입될 수 있다.

[0340] 적어도 하나의 실시예에서, 병렬 처리 유닛(4202)이 그래픽 처리를 수행하는 데 사용될 때, 스케줄러(4210)는, 처리 클러스터 어레이(4212)의 다수의 클러스터들(4214A-4214N)로의 그래픽 처리 연산들의 분배를 더 양호하게 가능하게 하기 위해, 처리 작업부하를 대략 동일한 크기의 태스크들로 분할하도록 구성될 수 있다. 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)의 부분들은 상이한 타입들의 처리를 수행하도록 구성될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 제1 부분은 정점 셰이딩 및 토폴로지 생성을 수행하도록 구성될 수 있고, 제2 부분은 테셀레이션 및 기하형상 셰이딩을 수행하도록 구성될 수 있고, 제3 부분은 픽셀 셰이딩 또는 다른 스크린 공간 동작들을 수행하여, 디스플레이를 위한 렌더링된 이미지를 생성하도록 구성될 수 있다. 적어도 하나의 실시예에서, 클러스터들(4214A-4214N) 중 하나 이상에 의해 생성된 중간 데이터는 추가 처리를 위해 클러스터들(4214A-4214N) 사이에서 중간 데이터가 송신되는 것을 허용하기 위해 버퍼에 저장될 수 있다.

[0341] 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)는 스케줄러(4210)를 통해 실행될 처리 태스크들을 수신할 수 있고, 스케줄러는 처리 태스크들을 정의하는 커맨드들을 프론트엔드(4208)로부터 수신한다. 적어도 하나의 실시예에서, 처리 태스크들은, 처리될 데이터, 예를 들어 표면(패치) 데이터, 프리미티브 데이터, 정점 데이터, 및/또는 픽셀 데이터의 인덱스들뿐만 아니라, 또한 데이터가 어떻게 처리되어야 하는지(예를 들어, 어떤 프로그램이 실행될지)를 정의하는 상태 파라미터들 및 커맨드들을 포함할 수 있다. 적어도 하나의 실시예에서, 스케줄러(4210)는 태스크들에 대응하는 인덱스들을 인출하도록 구성될 수 있거나 또는 프론트엔드(4208)로부터 인덱스들을 수신할 수 있다. 적어도 하나의 실시예에서, 프론트엔드(4208)는 유입 커맨드 버퍼들(예를 들어, 일괄-버퍼들, 푸시 버퍼들 등)에 의해 특정된 작업부하가 개시되기 전에 처리 클러스터 어레이(4212)가 유효 상태로 구성되는 것을 보장하도록 구성될 수 있다.

[0342] 적어도 하나의 실시예에서, 병렬 처리 유닛(4202)의 하나 이상의 인스턴스 각각은 병렬 프로세서 메모리(4222)와 결합될 수 있다. 적어도 하나의 실시예에서, 병렬 프로세서 메모리(4222)는 처리 클러스터 어레이(4212)뿐만 아니라 I/O 유닛(4204)으로부터 메모리 요청들을 수신할 수 있는 메모리 크로스바(4216)를 통해 액세스될 수 있다. 적어도 하나의 실시예에서, 메모리 크로스바(4216)는 메모리 인터페이스(4218)를 통해 병렬 프로세서 메모리(4222)에 액세스할 수 있다. 적어도 하나의 실시예에서, 메모리 인터페이스(4218)는 병렬 프로세서 메모리(4222)의 일부(예를 들어, 메모리 유닛)에 각각 결합될 수 있는 다수의 파티션 유닛들(예를 들어, 파티션 유닛(4220A), 파티션 유닛(4220B) 내지 파티션 유닛(4220N))을 포함할 수 있다. 적어도 하나의 실시예에서, 다수의 파티션 유닛(4220A-4220N)은 메모리 유닛의 수와 동일하도록 구성되어, 제1 파티션 유닛(4220A)은 대응하는 제1 메모리 유닛(4224A)을 갖고, 제2 파티션 유닛(4220B)은 대응하는 메모리 유닛(4224B)을 갖고, N번째 파티션 유닛(4220N)은 대응하는 N번째 메모리 유닛(4224N)을 갖는다. 적어도 하나의 실시예에서, 파티션 유닛(4220A-4220N)의 수는 메모리 디바이스의 수와 같지 않을 수 있다.

[0343] 적어도 하나의 실시예에서, 메모리 유닛들(4224A-4224N)은 동적 랜덤 액세스 메모리(DRAM) 또는 그래픽 더블 데이터 레이트(GDDR) 메모리를 포함하는 동기식 그래픽 랜덤 액세스 메모리(SGRAM)와 같은 그래픽 랜덤 액세스 메모리를 포함하는 다양한 타입들의 메모리 디바이스들을 포함할 수 있다. 적어도 하나의 실시예에서, 메모리 유닛들(4224A-4224N)은 또한, 고대역폭 메모리(HBM)를 포함한 그러나 이것으로 제한되지 않는 3D 적층형 메모리를 포함할 수 있다. 적어도 하나의 실시예에서, 프레임 버퍼 또는 텍스처 맵과 같은 렌더 타겟은 메모리 유닛(4224A-4224N)에 걸쳐 저장될 수 있어, 파티션 유닛(4220A-4220N)이 병렬 프로세서 메모리(4222)의 가용 대역폭을 효율적으로 사용하기 위해 각각의 렌더 타겟의 일부를 병렬로 기입할 수 있게 한다. 적어도 하나의 실시예에서, 병렬 프로세서 메모리(4222)의 로컬 인스턴스는 로컬 캐시 메모리와 관련하여 시스템 메모리를 이용하는 통합 메모리 설계에 대해서는 제외될 수 있다.

[0344] 적어도 하나의 실시예에서, 처리 클러스터 어레이(4212)의 클러스터들(4214A-4214N) 중 임의의 하나는 병렬 프로세서 메모리(4222) 내의 메모리 유닛들(4224A-4224N) 중 임의의 것에 기입될 데이터를 처리할 수 있다. 적어도 하나의 실시예에서, 메모리 크로스바(4216)는 각각의 클러스터(4214A-4214N)의 출력을 임의의 파티션 유닛(4220A-4220N)에 또는 다른 클러스터(4214A-4214N)에 전달하도록 구성될 수 있으며, 이는 출력에 대해 추가적인 처리 연산들을 수행할 수 있다. 적어도 하나의 실시예에서, 각각의 클러스터(4214A-4214N)는 메모리 크로스바(4216)를 통해 메모리 인터페이스(4218)와 통신하여 다양한 외부 메모리 디바이스로부터 판독하거나 이에 기입할 수 있다. 적어도 하나의 실시예에서, 메모리 크로스바(4216)는 I/O 유닛(4204)과 통신하기 위한 메모리 인터페이스(4218)에 대한 접속뿐만 아니라, 또한 병렬 프로세서 메모리(4222)의 로컬 인스턴스에 대한 접속을 가져

서, 상이한 처리 클러스터들(4214A-4214N) 내의 처리 유닛들이 시스템 메모리 또는 병렬 처리 유닛(4202)에 로컬이 아닌 다른 메모리와 통신할 수 있게 한다. 적어도 하나의 실시예에서, 메모리 크로스바(4216)는 가상 채널들을 사용하여 클러스터들(4214A-4214N)과 파티션 유닛들(4220A-4220N) 사이의 트래픽 스트림들을 분리할 수 있다.

[0345] 적어도 하나의 실시예에서, 병렬 처리 유닛(4202)의 다수의 인스턴스들이 단일의 애드-인 카드 상에 제공될 수 있거나, 또는 다수의 애드-인 카드들이 상호접속될 수 있다. 적어도 하나의 실시예에서, 병렬 처리 유닛(4202)의 상이한 인스턴스들은 상이한 인스턴스들이 상이한 수의 처리 코어들, 상이한 양의 로컬 병렬 프로세서 메모리, 및/또는 심지어 다른 구성 차이들을 갖더라도 상호 동작하도록 구성될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 병렬 처리 유닛(4202)의 일부 인스턴스는 다른 인스턴스에 비해 더 높은 정밀도 부동 소수점 유닛을 포함할 수 있다. 적어도 하나의 실시예에서, 병렬 처리 유닛(4202) 또는 병렬 프로세서(4200)의 하나 이상의 인스턴스를 통합하는 시스템은, 데스크탑, 랩탑, 또는 핸드헬드 개인용 컴퓨터, 서버, 워크스테이션, 게임 콘솔, 및/또는 임베디드 시스템을 포함한 그러나 이것으로 제한되지 않는 다양한 구성 및 폼 팩터로 구현될 수 있다.

[0346] 도 42b는 적어도 하나의 실시예에 따른 파티션 유닛(4220)의 블록도이다. 적어도 하나의 실시예에서, 파티션 유닛(4220)은 도 42a의 파티션 유닛(4220A-4220N) 중 하나의 인스턴스이다. 적어도 하나의 실시예에서, 파티션 유닛(4220)은 L2 캐시(4221), 프레임 버퍼 인터페이스(4225) 및 ROP(4226)(래스터 연산 유닛)를 포함한다. L2 캐시(4221)는 메모리 크로스바(4216) 및 ROP(4226)로부터 수신되는 로드 및 저장 연산들을 수행하도록 구성되는 판독/기입 캐시이다. 적어도 하나의 실시예에서, 판독 누락들 및 긴급 라이트-백 요청들은 처리를 위해 L2 캐시(4221)에 의해 프레임 버퍼 인터페이스(4225)에 출력된다. 적어도 하나의 실시예에서, 업데이트는 또한 처리를 위해 프레임 버퍼 인터페이스(4225)를 통해 프레임 버퍼에 전송될 수 있다. 적어도 하나의 실시예에서, 프레임 버퍼 인터페이스(4225)는 (예를 들어, 병렬 프로세서 메모리(4222) 내의) 도 42의 메모리 유닛들(4224A-4224N)과 같은 병렬 프로세서 메모리 내의 메모리 유닛들 중 하나와 인터페이스한다.

[0347] 적어도 하나의 실시예에서, ROP(4226)는 스텐실, z 테스트, 블렌딩 등과 같은 래스터 연산들을 수행하는 처리 유닛이다. 적어도 하나의 실시예에서, ROP(4226)는 그 후 그래픽 메모리에 저장되는 처리된 그래픽 데이터를 출력한다. 적어도 하나의 실시예에서, ROP(4226)는 메모리에 기입되는 깊이 또는 컬러 데이터를 압축하고 메모리로부터 판독되는 깊이 또는 컬러 데이터를 압축해제하는 압축 로직을 포함한다. 적어도 하나의 실시예에서, 압축 로직은 다수의 압축 알고리즘 중 하나 이상을 이용하는 무손실 압축 로직일 수 있다. ROP(4226)에 의해 수행되는 압축의 타입은 압축될 데이터의 통계적 특성에 기초하여 달라질 수 있다. 예를 들어, 적어도 하나의 실시예에서, 델타 컬러 압축은 타일별 기반으로 깊이 및 컬러 데이터에 관해 수행된다.

[0348] 적어도 하나의 실시예에서, ROP(4226)는 파티션 유닛(4220) 내 대신에 각각의 처리 클러스터(예를 들어, 도 42의 클러스터(4214A-4214N)) 내에 포함된다. 적어도 하나의 실시예에서, 픽셀 데이터에 대한 판독 및 기입 요청들은 픽셀 프래그먼트 데이터 대신에 메모리 크로스바(4216)를 통해 송신된다. 적어도 하나의 실시예에서, 처리된 그래픽 데이터는 도 41의 하나 이상의 디스플레이 디바이스(들)(4110) 중 하나와 같은 디스플레이 디바이스 상에 디스플레이되거나, 프로세서(들)(4102)에 의한 추가 처리를 위해 라우팅되거나, 또는 도 42a의 병렬 프로세서(4200) 내의 처리 엔티티들 중 하나에 의한 추가 처리를 위해 라우팅될 수 있다.

[0349] 도 42c는 적어도 하나의 실시예에 따른 병렬 처리 유닛 내의 처리 클러스터(4214)의 블록도이다. 적어도 하나의 실시예에서, 처리 클러스터는 도 42의 처리 클러스터들(4214A-4214N) 중 하나의 인스턴스이다. 적어도 하나의 실시예에서, 처리 클러스터(4214)는 다수의 스레드들을 병렬로 실행하도록 구성될 수 있으며, 용어 "스레드"는 입력 데이터의 특정 세트 상에서 실행하는 특정 프로그램의 인스턴스를 지칭한다. 적어도 하나의 실시예에서, SIMD(single-instruction, multiple-data) 명령어 발행 기법들이, 다수의 독립적인 명령어 유닛을 제공하지 않고 많은 수의 스레드의 병렬 실행을 지원하는데 이용된다. 적어도 하나의 실시예에서, SIMT(single-instruction, multiple-thread) 기법들이, 각각의 처리 클러스터 내의 한 세트의 처리 엔진들에 명령어들을 발행하도록 구성된 공통 명령어 유닛을 이용하여, 많은 수의 일반적으로 동기화된 스레드들의 병렬 실행을 지원하는데 이용된다.

[0350] 적어도 하나의 실시예에서, 처리 클러스터(4214)의 동작은 처리 태스크들을 SIMT 병렬 프로세서들에 분배하는 파이프라인 관리자(4232)를 통해 제어될 수 있다. 적어도 하나의 실시예에서, 파이프라인 관리자(4232)는 도 42의 스케줄러(4210)로부터 명령어들을 수신하고, 그래픽 멀티프로세서(4234) 및/또는 텍스처 유닛(4236)을 통해 그 명령어들의 실행을 관리한다. 적어도 하나의 실시예에서, 그래픽 멀티프로세서(4234)는 SIMT 병렬 프로세서

의 예시적인 인스턴스이다. 그러나, 적어도 하나의 실시예에서, 상이한 아키텍처들의 다양한 타입들의 SIMT 병렬 프로세서들이 처리 클러스터(4214) 내에 포함될 수 있다. 적어도 하나의 실시예에서, 그래픽 멀티프로세서(4234)의 하나 이상의 인스턴스가 처리 클러스터(4214) 내에 포함될 수 있다. 적어도 하나의 실시예에서, 그래픽 멀티프로세서(4234)는 데이터를 처리할 수 있고, 데이터 크로스바(4240)는 처리된 데이터를 다른 셰이더 유닛들을 포함하는 다수의 가능한 목적지 중 하나에 분배하는 데 이용될 수 있다. 적어도 하나의 실시예에서, 파이프라인 관리자(4232)는 데이터 크로스바(4240)를 통해 분배될 처리된 데이터에 대한 목적지를 명시함으로써 처리된 데이터의 분배를 용이하게 할 수 있다.

[0351] 적어도 하나의 실시예에서, 처리 클러스터(4214) 내의 각각의 그래픽 멀티프로세서(4234)는 동일한 세트의 기능 실행 로직(예를 들어, 산술 로직 유닛들, 로드-저장 유닛들 등)을 포함할 수 있다. 적어도 하나의 실시예에서, 기능 실행 로직은 이전 명령어들이 완료되기 전에 새로운 명령어들이 발행될 수 있는 파이프라인 방식으로 구성될 수 있다. 적어도 하나의 실시예에서, 기능 실행 로직은, 정수 및 부동 소수점 산술, 비교 연산들, 부울 연산들, 비트 시프팅, 및 다양한 대수 함수들의 계산을 포함한 다양한 연산을 지원한다. 적어도 하나의 실시예에서, 동일한 기능 유닛 하드웨어가 활용되어 상이한 동작들을 수행할 수 있고 기능 유닛들의 임의의 조합이 존재할 수 있다.

[0352] 적어도 하나의 실시예에서, 처리 클러스터(4214)에 송신된 명령어들은 스레드를 구성한다. 적어도 하나의 실시예에서, 한 세트의 병렬 처리 엔진들에 걸쳐 실행되는 한 세트의 스레드들은 스레드 그룹이다. 적어도 하나의 실시예에서, 스레드 그룹은 상이한 입력 데이터에 관해 프로그램을 실행한다. 적어도 하나의 실시예에서, 스레드 그룹 내의 각각의 스레드는 그래픽 멀티프로세서(4234) 내의 상이한 처리 엔진에 할당될 수 있다. 적어도 하나의 실시예에서, 스레드 그룹은 그래픽 멀티프로세서(4234) 내의 처리 엔진들의 수보다 적은 수의 스레드들을 포함할 수 있다. 적어도 하나의 실시예에서, 스레드 그룹이 처리 엔진의 수보다 적은 수의 스레드를 포함할 때, 처리 엔진들 중 하나 이상은 그 스레드 그룹이 처리되고 있는 사이클들 동안 유휴 상태일 수 있다. 적어도 하나의 실시예에서, 스레드 그룹은 또한 그래픽 멀티프로세서(4234) 내의 처리 엔진들의 수보다 많은 스레드들을 포함할 수 있다. 적어도 하나의 실시예에서, 스레드 그룹은 그래픽 멀티프로세서(4234) 내의 처리 엔진들의 수보다 많은 스레드들을 포함하고, 처리는 연속적인 클록 사이클들에 걸쳐 수행될 수 있다. 적어도 하나의 실시예에서, 다수의 스레드 그룹이 그래픽 멀티프로세서(4234) 상에서 동시에 실행될 수 있다.

[0353] 적어도 하나의 실시예에서, 그래픽 멀티프로세서(4234)는 로드 및 저장 연산들을 수행하기 위한 내부 캐시 메모리를 포함한다. 적어도 하나의 실시예에서, 그래픽 멀티프로세서(4234)는 내부 캐시를 사용하지 않고 처리 클러스터(4214) 내의 캐시 메모리(예를 들어, L1 캐시(4248))를 사용할 수 있다. 적어도 하나의 실시예에서, 각각의 그래픽 멀티프로세서(4234)는 또한 모든 처리 클러스터들(4214) 사이에 공유되고 스레드들 사이에 데이터를 전달하는데 사용될 수 있는 파티션 유닛들(예를 들어, 도 42의 파티션 유닛들(4220A-4220N)) 내의 L2 캐시에 액세스할 수 있다. 적어도 하나의 실시예에서, 그래픽 멀티프로세서(4234)는 또한 로컬 병렬 프로세서 메모리 및/또는 시스템 메모리 중 하나 이상을 포함할 수 있는 오프-칩 글로벌 메모리에 액세스할 수 있다. 적어도 하나의 실시예에서, 병렬 처리 유닛(4202) 외부의 임의의 메모리가 글로벌 메모리로서 사용될 수 있다. 적어도 하나의 실시예에서, 처리 클러스터(4214)는 공통 명령어들 및 데이터를 공유할 수 있는 그래픽 멀티프로세서(4234)의 다수의 인스턴스들을 포함하며, 이는 L1 캐시(4248)에 저장될 수 있다.

[0354] 적어도 하나의 실시예에서, 각각의 처리 클러스터(4214)는 가상 어드레스들을 물리적 어드레스들로 맵핑하도록 구성된 MMU(4245)(메모리 관리 유닛)를 포함할 수 있다. 적어도 하나의 실시예에서, MMU(4245)의 하나 이상의 인스턴스는 도 42의 메모리 인터페이스(4218) 내에 상주할 수 있다. 적어도 하나의 실시예에서, MMU(4245)는 가상 어드레스를 타일의 물리적 어드레스(타일링에 관련하여 더 설명함) 및 선택적으로 캐시 라인 인덱스에 맵핑하는데 사용되는 페이지 테이블 엔트리(PTE)들의 세트를 포함한다. 적어도 하나의 실시예에서, MMU(4245)는 그래픽 멀티프로세서(4234) 또는 L1 캐시 또는 처리 클러스터(4214) 내에 상주할 수 있는 어드레스 변환 색인 버퍼들(TLB) 또는 캐시들을 포함할 수 있다. 적어도 하나의 실시예에서, 물리적 어드레스는 파티션 유닛들 사이에서 효율적인 요청 인터리빙을 허용하기 위해 표면 데이터 액세스 지역성을 분배하도록 처리된다. 적어도 하나의 실시예에서, 캐시 라인 인덱스는 캐시 라인에 대한 요청이 히트인지 미스인지를 결정하는데 이용될 수 있다.

[0355] 적어도 하나의 실시예에서, 처리 클러스터(4214)는, 각각의 그래픽 멀티프로세서(4234)가 텍스처 맵핑 동작을 수행하기 위해, 예를 들어, 텍스처 샘플 위치를 결정하고, 텍스처 데이터를 판독하고, 텍스처 데이터를 필터링하기 위해 텍스처 유닛(4236)에 결합되도록 구성될 수 있다. 적어도 하나의 실시예에서, 텍스처 데이터는 내부 텍스처 L1 캐시(도시되지 않음)로부터 또는 그래픽 멀티프로세서(4234) 내의 L1 캐시로부터 판독되고, 필요에 따라, L2 캐시, 로컬 병렬 프로세서 메모리, 또는 시스템 메모리로부터 인출된다. 적어도 하나의 실시예에서,

각각의 그래픽 멀티프로세서(4234)는 처리된 태스크들을 데이터 크로스바(4240)에 출력하여, 처리된 태스크를 추가 처리를 위해 다른 처리 클러스터(4214)에 제공하거나, 처리된 태스크를 메모리 크로스바(4216)를 통해 L2 캐시, 로컬 병렬 프로세서 메모리 또는 시스템 메모리에 저장한다. 적어도 하나의 실시예에서, PreROP(4242)(프리-래스터 연산 유닛)은 그래픽 멀티프로세서(4234)로부터 데이터를 수신하고, 데이터를 ROP 유닛들로 지향시키도록 구성되며, ROP 유닛들은 본 명세서에 설명된 바와 같은 파티션 유닛들(예를 들어, 도 42의 파티션 유닛들(4220A-4220N))과 함께 위치할 수 있다. 적어도 하나의 실시예에서, PreROP(4242) 유닛은 컬러 혼합을 위한 최적화를 수행하고, 픽셀 컬러 데이터를 조직화하고, 어드레스 변환을 수행할 수 있다.

[0356] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 본 명세서에 설명된 신경망 훈련 동작들, 신경망 기능들 및/또는 아키텍처들, 또는 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 동작들을 추론 또는 예측하기 위해 그래픽 처리 클러스터(4214)에서 이용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 그래픽 멀티프로세서로 구현된다.

[0357] 도 42d는 적어도 하나의 실시예에 따른 그래픽 멀티프로세서(4234)를 도시한다. 적어도 하나의 실시예에서, 그래픽 멀티프로세서(4234)는 처리 클러스터(4214)의 파이프라인 관리자(4232)와 결합된다. 적어도 하나의 실시예에서, 그래픽 멀티프로세서(4234)는 명령어 캐시(4252), 명령어 유닛(4254), 어드레스 맵핑 유닛(4256), 레지스터 파일(4258), 하나 이상의 범용 그래픽 처리 유닛(GPGPU) 코어(4262) 및 하나 이상의 로드/저장 유닛(4266)을 포함하지만 이에 제한되지 않는 실행 파이프라인을 갖는다. GPGPU 코어들(4262) 및 로드/저장 유닛들(4266)은 메모리 및 캐시 인터커넥트(4268)를 통해 캐시 메모리(4272) 및 공유된 메모리(4270)와 결합된다.

[0358] 적어도 하나의 실시예에서, 명령어 캐시(4252)는 파이프라인 관리자(4232)로부터 실행할 명령어들의 스트림을 수신한다. 적어도 하나의 실시예에서, 명령어들은 명령어 캐시(4252)에 캐싱되고 명령어 유닛(4254)에 의한 실행을 위해 디스패치된다. 적어도 하나의 실시예에서, 명령어 유닛(4254)은 명령어들을 스레드 그룹들(예를 들어, 워프들)로서 디스패치할 수 있고, 스레드 그룹의 각각의 스레드는 GPGPU 코어(4262) 내의 상이한 실행 유닛에 할당된다. 적어도 하나의 실시예에서, 명령어는, 통합된 어드레스 공간적 내의 어드레스를 명시함으로써 로컬, 공유, 또는 글로벌 어드레스 공간 중 임의의 것에 액세스할 수 있다. 적어도 하나의 실시예에서, 어드레스 맵핑 유닛(4256)은 통합된 어드레스 공간에서의 어드레스들을 로드/저장 유닛들(4266)에 의해 액세스될 수 있는 개별 메모리 어드레스로 변환하는 데 사용될 수 있다.

[0359] 적어도 하나의 실시예에서, 레지스터 파일(4258)은 그래픽 멀티프로세서(4234)의 기능 유닛들에 대한 레지스터들의 세트를 제공한다. 적어도 하나의 실시예에서, 레지스터 파일(4258)은 그래픽 멀티프로세서(4234)의 기능 유닛들(예를 들어, GPGPU 코어들(4262), 로드/저장 유닛들(4266))의 데이터 경로들에 접속된 피연산자들에 대한 임시 저장소를 제공한다. 적어도 하나의 실시예에서, 레지스터 파일(4258)은 각각의 기능 유닛이 레지스터 파일(4258)의 전용 부분을 할당받도록 각각의 기능 유닛들 사이에서 분할된다. 적어도 하나의 실시예에서, 레지스터 파일(4258)은 그래픽 멀티프로세서(4234)에 의해 실행되는 상이한 워프들 사이에서 분할된다.

[0360] 적어도 하나의 실시예에서, GPGPU 코어들(4262)은 각각 그래픽 멀티프로세서(4234)의 명령어들을 실행하는 데 사용되는 부동 소수점 유닛들(FPU들) 및/또는 정수 산술 로직 유닛들(ALU들)을 포함할 수 있다. GPGPU 코어들(4262)은 아키텍처가 유사할 수 있거나 아키텍처가 상이할 수 있다. 적어도 하나의 실시예에서, GPGPU 코어들(4262)의 제1 부분은 단정도 FPU 및 정수 ALU를 포함하는 한편, GPGPU 코어들의 제2 부분은 배정도 FPU를 포함한다. 적어도 하나의 실시예에서, FPU들은 부동 소수점 산술을 위한 IEEE 754-2008 표준을 구현하거나 가변 정밀도 부동 소수점 산술을 가능하게 할 수 있다. 적어도 하나의 실시예에서, 그래픽 멀티프로세서(4234)는 복사-직사각형 또는 픽셀 블렌딩 연산들과 같은 특정 기능들을 수행하기 위해 하나 이상의 고정 기능 또는 특수 기능 유닛들을 더 포함할 수 있다. 적어도 하나의 실시예에서, GPGPU 코어들 중 하나 이상은 또한 고정 또는 특수 기능 로직을 포함할 수 있다.

[0361] 적어도 하나의 실시예에서, GPGPU 코어들(4262)은 다수의 데이터 세트에 대해 단일 명령어를 수행할 수 있는 SIMD 로직을 포함한다. 적어도 하나의 실시예에서, GPGPU 코어들(4262)은 SIMD4, SIMD8, 및 SIMD16 명령어들을 물리적으로 실행하고 SIMD1, SIMD2, 및 SIMD32 명령어들을 논리적으로 실행할 수 있다. 적어도 하나의 실시예에서, GPGPU 코어들에 대한 SIMD 명령어들은 셰이더 컴파일러에 의해 컴파일 시간에 생성되거나, SPMD(single program multiple data) 또는 SIMT 아키텍처들에 대해 기입되고 컴파일된 프로그램들을 실행할 때 자동으로 생

성될 수 있다. 적어도 하나의 실시예에서, SIMT 실행 모델을 위해 구성된 프로그램의 다수의 스레드는 단일 SIMD 명령어를 통해 실행될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 동일하거나 유사한 동작들을 수행하는 8개의 SIMT 스레드는 단일 SIMD8 로직 유닛을 통해 병렬로 실행될 수 있다.

[0362] 적어도 하나의 실시예에서, 메모리 및 캐시 인터커넥트(4268)는 그래픽 멀티프로세서(4234)의 각각의 기능 유닛을 레지스터 파일(4258) 및 공유된 메모리(4270)에 접속하는 인터커넥트 네트워크이다. 적어도 하나의 실시예에서, 메모리 및 캐시 인터커넥트(4268)는 로드/저장 유닛(4266)이 공유된 메모리(4270)와 레지스터 파일(4258) 사이의 로드 및 저장 연산들을 구현할 수 있게 하는 크로스바 인터커넥트이다. 적어도 하나의 실시예에서, 레지스터 파일(4258)은 GPGPU 코어들(4262)과 동일한 주파수에서 동작할 수 있으며, 따라서 GPGPU 코어들(4262)과 레지스터 파일(4258) 사이의 데이터 전달은 초저레이턴시이다. 적어도 하나의 실시예에서, 공유된 메모리(4270)는 그래픽 멀티프로세서(4234) 내의 기능 유닛들 상에서 실행되는 스레드들 사이의 통신을 가능하게 하는 데 사용될 수 있다. 적어도 하나의 실시예에서, 캐시 메모리(4272)는 예를 들어 기능 유닛들과 텍스처 유닛(4236) 사이에서 통신되는 텍스처 데이터를 캐싱하기 위한 데이터 캐시로서 사용될 수 있다. 적어도 하나의 실시예에서, 공유된 메모리(4270)는 또한 프로그램 관리 캐시로서 사용될 수 있다. 적어도 하나의 실시예에서, GPGPU 코어들(4262) 상에서 실행되는 스레드들은 캐시 메모리(4272) 내에 저장된 자동으로 캐싱된 데이터 이외에 공유된 메모리 내에 데이터를 프로그램적으로 저장할 수 있다.

[0363] 적어도 하나의 실시예에서, 본 명세서에 설명된 병렬 프로세서 또는 GPGPU는, 그래픽 연산들, 머신 러닝 동작들, 패턴 분석 동작들, 및 다양한 범용 GPU(GPGPU) 기능들을 가속하기 위해 호스트/프로세서 코어들에 통신 가능하게 결합된다. 적어도 하나의 실시예에서, GPU는 버스 또는 다른 인터커넥트(예를 들어, PCIe 또는 NVLink 같은 고속 인터커넥트)를 통해 호스트 프로세서/코어들에 통신가능하게 결합될 수 있다. 적어도 하나의 실시예에서, GPU는 코어들과 동일한 패키지 또는 칩에 통합될 수 있고 내부 프로세서 버스/인터커넥트(즉, 패키지 또는 칩 내부)를 통해 코어들에 통신가능하게 결합될 수 있다. 적어도 하나의 실시예에서, GPU가 접속되는 방식에 관계없이, 프로세서 코어들은 작업 기술자에 포함된 커맨드들/명령어들의 시퀀스들의 형태로 작업을 GPU에 할당할 수 있다. 적어도 하나의 실시예에서, GPU는, 그 다음, 이들 커맨드들/명령어들을 효율적으로 처리하기 위해 전용 회로/로직을 이용한다.

[0364] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 본 명세서에 설명된 신경망 훈련 동작들, 신경망 기능들 및/또는 아키텍처들, 또는 신경망 이용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 동작들을 추론 또는 예측하기 위해 그래픽 멀티프로세서(4234)에서 이용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 다중-GPU 컴퓨터 시스템으로 구현된다.

[0365] 도 43은 적어도 하나의 실시예에 따른 다중 GPU 계산 시스템(4300)을 예시한다. 적어도 하나의 실시예에서, 다중 GPU 계산 시스템(4300)은 호스트 인터페이스 스위치(4304)를 통해 다수의 범용 그래픽 처리 유닛(GPGPU)(4306A-D)에 결합된 프로세서(4302)를 포함할 수 있다. 적어도 하나의 실시예에서, 호스트 인터페이스 스위치(4304)는 프로세서(4302)가 GPGPU들(4306A-D)과 통신할 수 있는 PCI 익스프레스 버스에 프로세서(4302)를 결합하는 PCI 익스프레스 스위치 디바이스이다. GPGPU들(4306A-D)은 GPU 대 GPU 링크들(4316)의 고속 포인트-투-포인트 세트를 통해 상호접속할 수 있다. 적어도 하나의 실시예에서, GPU 대 GPU 링크들(4316)은 전용 GPU 링크를 통해 GPGPU들(4306A-D) 각각에 접속된다. 적어도 하나의 실시예에서, P2P GPU 링크들(4316)은 프로세서(4302)가 접속되는 호스트 인터페이스 버스(4304)를 통한 통신을 요구하지 않고 GPGPU들(4306A-D) 각각 사이의 직접 통신을 가능하게 한다. 적어도 하나의 실시예에서, P2P GPU 링크들(4316)로 지향되는 GPU-대-GPU 트래픽으로, 호스트 인터페이스 버스(4304)는, 예를 들어, 하나 이상의 네트워크 디바이스들을 통해, 멀티-GPU 계산 시스템(4300)의 다른 인스턴스들과 통신하거나 시스템 메모리 액세스에 대해 이용 가능하게 유지된다. 적어도 하나의 실시예에서 GPGPU들(4306A-D)은 호스트 인터페이스 스위치(4304)를 통해 프로세서(4302)에 접속하지만, 적어도 하나의 실시예에서 프로세서(4302)는 P2P GPU 링크들(4316)에 대한 직접 지원을 포함하고 GPGPU들(4306A-D)에 직접 접속할 수 있다.

[0366] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은, 본 명세서에 설명되는 신경망 훈련 동작들, 신경망 기능들 및/또는 아키텍처들, 또는 신경망 사용 사례들을 이용하여 산출되는 가중치 파라미터들에, 적어

도 부분적으로, 기초하여 동작들을 추론하거나 또는 예측하기 위해 멀티-GPU 계산 시스템(4300)에서 사용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 그래픽 프로세서로 구현된다.

[0367] 도 44는 적어도 하나의 실시예에 따른 그래픽 프로세서(4400)의 블록도이다. 적어도 하나의 실시예에서, 그래픽 프로세서(4400)는 링 인터커넥트(4402), 파이프라인 프론트엔드(4404), 미디어 엔진(4437), 및 그래픽 코어(4480A 내지 4480N)를 포함한다. 적어도 하나의 실시예에서, 링 인터커넥트(4402)는 그래픽 프로세서(4400)를 다른 그래픽 프로세서들 또는 하나 이상의 범용 프로세서 코어를 포함하는 다른 처리 유닛들에 결합한다. 적어도 하나의 실시예에서, 그래픽 프로세서(4400)는 멀티 코어 처리 시스템 내에 통합된 많은 프로세서들 중 하나이다.

[0368] 적어도 하나의 실시예에서, 그래픽 프로세서(4400)는 링 인터커넥트(4402)를 통해 커맨드들의 배치를 수신한다. 적어도 하나의 실시예에서, 유입 커맨드들은 파이프라인 프론트엔드(4404)의 커맨드 스트리머(4403)에 의해 해석된다. 적어도 하나의 실시예에서, 그래픽 프로세서(4400)는 그래픽 코어(들)(4480A-4480N)를 통해 3D 기하형상 처리 및 미디어 처리를 수행하는 스케일러블 실행 로직을 포함한다. 적어도 하나의 실시예에서, 3D 기하형상 처리 커맨드의 경우, 커맨드 스트리머(4403)는 기하형상 파이프라인(4436)에 커맨드를 공급한다. 적어도 하나의 실시예에서, 적어도 일부 미디어 처리 커맨드들에 대해, 커맨드 스트리머(4403)는 커맨드들을 비디오 프론트엔드(4434)에 공급하며, 비디오 프론트엔드는 미디어 엔진(4437)과 결합된다. 적어도 하나의 실시예에서, 미디어 엔진(4437)은 비디오 및 이미지 후처리를 위한 VQE(Video Quality Engine)(4430) 및 하드웨어-가속된 미디어 데이터 인코딩 및 디코딩을 제공하는 MFX(multi-format encode/decode)(4433) 엔진을 포함한다. 적어도 하나의 실시예에서, 기하형상 파이프라인(4436) 및 미디어 엔진(4437) 각각은 적어도 하나의 그래픽 코어(4480A)에 의해 제공되는 스레드 실행 리소스에 대한 실행 스레드를 생성한다.

[0369] 적어도 하나의 실시예에서, 그래픽 프로세서(4400)는 다수의 서브코어들(4450A-550N, 4460A-4460N)(때때로 코어 서브-슬라이스들로 지칭됨)을 각각 갖는 모듈러 코어들(4480A-4480N)(때때로 코어 슬라이스들로 지칭됨)을 특징으로 하는 스케일러블 스레드 실행 리소스들을 포함한다. 적어도 하나의 실시예에서, 그래픽 프로세서(4400)는 임의의 수의 그래픽 코어들(4480A 내지 4480N)을 가질 수 있다. 적어도 하나의 실시예에서, 그래픽 프로세서(4400)는 적어도 제1 서브코어(4450A) 및 제2 서브코어(4460A)를 갖는 그래픽 코어(4480A)를 포함한다. 적어도 하나의 실시예에서, 그래픽 프로세서(4400)는 단일 서브코어(예를 들어, 4450A)를 갖는 저전력 프로세서이다. 적어도 하나의 실시예에서, 그래픽 프로세서(4400)는 다수의 그래픽 코어(4480A-4480N)를 포함하며, 이들 각각은 제1 서브코어들(4450A-4450N)의 세트 및 제2 서브코어들(4460A-4460N)의 세트를 포함한다. 적어도 하나의 실시예에서, 제1 서브코어들(4450A-4450N) 내의 각각의 서브코어는 적어도 제1 세트의 실행 유닛들(4452A-4452N) 및 미디어/텍스처 샘플러들(4454A-4454N)을 포함한다. 적어도 하나의 실시예에서, 제2 서브코어들(4460A-4460N) 내의 각각의 서브코어는 실행 유닛들(4462A-4462N) 및 샘플러들(4464A-4464N)의 적어도 제2 세트를 포함한다. 적어도 하나의 실시예에서, 각각의 서브코어(4450A-4450N, 4460A-4460N)는 공유 리소스들(4470A-4470N)의 세트를 공유한다. 적어도 하나의 실시예에서, 공유된 리소스들은 공유된 캐시 메모리 및 픽셀 동작 로직을 포함한다.

[0370] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)은 본 명세서에 설명된 신경망 훈련 동작들, 신경망 기능들 및/또는 아키텍처들, 또는 신경망 사용 사례들을 이용하여 산출된 가중치 파라미터들에 적어도 부분적으로 기초하여 동작들을 추론 또는 예측하기 위해 그래픽 프로세서(4400)에서 사용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 마이크로-아키텍처로 구현된다.

[0371] 도 45는 적어도 하나의 실시예에 따라, 명령어들을 수행하는 로직 회로들을 포함할 수 있는 프로세서(4500)에 대한 마이크로-아키텍처를 예시하는 블록도이다. 적어도 하나의 실시예에서, 프로세서(4500)는 x86 명령어들, ARM 명령어들, ASIC들(application-specific integrated circuits)에 대한 전문화된 명령어들 등을 포함하는 명령어들을 수행할 수 있다. 적어도 하나의 실시예에서, 프로세서(4510)는 캘리포니아 산타 클라라의 인텔 코퍼레이션으로부터의 MMX 기술로 강화된 마이크로프로세서들 내의 64-비트 폭의 MMX[™] 레지스터들과 같은 패키징된 데이터를 저장하기 위한 레지스터들을 포함할 수 있다. 적어도 하나의 실시예에서, 정수 및 부동 소수점 형태들 양자 모두로 이용 가능한 MMX 레지스터들은 단일 명령어, 다중 데이터("SIMD") 및 스트리밍 SIMD 확장들("SSE")

명령어들을 수반하는 패킹된 데이터 엘리먼트들로 동작할 수 있다. 적어도 하나의 실시예에서, SSE2, SSE3, SSE4, AVX, 또는 그 이상의(일반적으로 "SSEx"로 지칭됨) 기술에 관련한 128-비트 폭의 XMM 레지스터들이 이러한 패킹된 데이터 피연산자들을 보유할 수 있다. 적어도 하나의 실시예에서, 프로세서들(4510)은 머신 러닝 또는 딥 러닝 알고리즘들, 훈련 또는 추론을 가속화하기 위한 명령어들을 수행할 수 있다.

[0372] 적어도 하나의 실시예에서, 프로세서(4500)는 실행될 명령어들을 인출하고 프로세서 파이프라인에서 나중에 사용될 명령어들을 준비하는 순차 프론트엔드("프론트엔드")(4501)를 포함한다. 적어도 하나의 실시예에서, 프론트엔드(4501)는 여러 유닛들을 포함할 수 있다. 적어도 하나의 실시예에서, 명령어 프리페처(instruction prefetcher)(4526)는 메모리로부터 명령어들을 인출하고 명령어들을 명령어 디코더(4528)에 공급하며, 명령어 디코더는 차례로 명령어들을 디코딩하거나 해석한다. 예를 들어, 적어도 하나의 실시예에서, 명령어 디코더(4528)는 수신된 명령어를 머신이 실행할 수 있는 "마이크로-명령어들" 또는 "마이크로-연산들"("마이크로-op들" 또는 "uop들"이라고도 불림)이라고 지칭되는 하나 이상의 연산으로 디코딩한다. 적어도 하나의 실시예에서, 명령어 디코더(4528)는 적어도 하나의 실시예에 따른 연산들을 수행하기 위해 마이크로-아키텍처에 의해 사용될 수 있는 오퍼코드 및 대응하는 데이터 및 제어 필드들로 명령어를 파싱한다. 적어도 하나의 실시예에서, 트레이스 캐시(4530)는 디코딩된 uop들을 실행을 위해 uop 큐(4534) 내의 프로그램 순서 시퀀스들 또는 트레이스들로 어셈블링할 수 있다. 적어도 하나의 실시예에서, 트레이스 캐시(4530)가 복합 명령어를 만날 때, 마이크로코드 ROM(4532)은 연산을 완료하는데 필요한 uop들을 제공한다.

[0373] 적어도 하나의 실시예에서, 일부 명령어는 단일 마이크로-op로 변환될 수 있는 반면, 다른 것들은 전체 연산을 완료하기 위해 여러 마이크로-op들을 필요로 한다. 적어도 하나의 실시예에서, 명령어를 완료하는 데 4개 초과 마이크로-op가 필요한 경우, 명령어 디코더(4528)는 명령어를 수행하기 위해 마이크로코드 ROM(4532)에 액세스할 수 있다. 적어도 하나의 실시예에서, 명령어는 명령어 디코더(4528)에서의 처리를 위해 소수의 마이크로-op들로 디코딩될 수 있다. 적어도 하나의 실시예에서, 연산을 달성하는 데 다수의 마이크로-op가 필요한 경우, 명령어가 마이크로코드 ROM(4532) 내에 저장될 수 있다. 적어도 하나의 실시예에서, 트레이스 캐시(4530)는 적어도 하나의 실시예에 따라 마이크로코드 ROM(4532)으로부터 하나 이상의 명령어를 완료하기 위해 마이크로코드 시퀀스들을 판독하기 위한 정확한 마이크로-명령어 포인터를 결정하기 위해 진입점 프로그램가능 로직 어레이("PLA")를 참조한다. 적어도 하나의 실시예에서, 마이크로코드 ROM(4532)이 명령어에 대한 마이크로-op들의 시퀀싱을 완료한 이후에, 머신의 프론트엔드(4501)는 트레이스 캐시(4530)로부터 마이크로-op들을 인출하는 것을 재개할 수 있다.

[0374] 적어도 하나의 실시예에서, 비순차적 실행 엔진("비순차적 엔진")(4503)은 실행을 위한 명령어들을 준비할 수 있다. 적어도 하나의 실시예에서, 비순차적 실행 로직은, 명령어들이 파이프라인을 따라 내려가 실행을 위해 스케줄링될 때 성능을 최적화하기 위해 명령어들의 흐름을 평활화하고 재정렬하는 다수의 버퍼를 갖는다. 비순차적 실행 엔진(4503)은 할당기/레지스터 개명기(4540), 메모리 uop 큐(4542), 정수/부동 소수점 uop 큐(4544), 메모리 스케줄러(4546), 고속 스케줄러(4502), 저속/일반 부동 소수점 스케줄러("저속/일반 FP 스케줄러")(4504), 및 단순 부동 소수점 스케줄러("단순 FP 스케줄러")(4506)를 포함하지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, 고속 스케줄러(4502), 저속/일반 부동 소수점 스케줄러(4504), 및 단순 부동 소수점 스케줄러(4506)는 또한 본 명세서에서 집합적으로 "uop 스케줄러들(4502, 4504, 4506)"이라고 지칭된다. 할당기/레지스터 개명기(4540)는 각각의 uop가 실행하기 위해 필요로 하는 머신 버퍼들 및 리소스들을 할당한다. 적어도 하나의 실시예에서, 할당기/레지스터 개명기(4540)는 로직 레지스터들을 레지스터 파일 내의 엔트리들로 개명한다. 적어도 하나의 실시예에서, 할당기/레지스터 개명기(4540)는 또한, 메모리 스케줄러(4546) 및 uop 스케줄러들(4502, 4504, 4506) 앞에, 2개의 uop 큐들, 즉, 메모리 연산들을 위한 메모리 uop 큐(4542) 및 비-메모리 연산들을 위한 정수/부동 소수점 uop 큐(4544) 중 하나에서 각각의 uop에 대한 엔트리를 할당한다. 적어도 하나의 실시예에서, uop 스케줄러들(4502, 4504, 4506)은, 그들의 종속 입력 레지스터 피연산자 소스들의 준비 상태 및 uop가 그 연산을 완료하기 위해 필요로 하는 실행 리소스들의 가용성에 기초하여, uop가 실행될 준비가 된 시기를 결정한다. 적어도 하나의 실시예에서, 적어도 하나의 실시예의 고속 스케줄러(4502)는 메인 클록 사이클의 각각의 절반마다 스케줄링할 수 있는 반면, 저속/일반 부동 소수점 스케줄러(4504) 및 단순 부동 소수점 스케줄러(4506)는 메인 프로세서 클록 사이클마다 한번 스케줄링할 수 있다. 적어도 하나의 실시예에서, uop 스케줄러들(4502, 4504, 4506)은 실행을 위해 uop들을 스케줄링하기 위해 디스패치 포트들에 대해 중재한다.

[0375] 적어도 하나의 실시예에서, 실행 블록 b11은, 제한없이, 정수 레지스터 파일/바이패스 네트워크(4508), 부동 소수점 레지스터 파일/바이패스 네트워크("FP 레지스터 파일/바이패스 네트워크")(4510), 어드레스 생성 유닛들("AGU들")(4512 및 4514), 고속 산술 로직 유닛들("고속 ALU들")(4516 및 4518), 저속 산술 로직 유닛("저속

ALU)(4520), 부동 소수점 ALU("FP")(4522), 및 부동 소수점 이동 유닛("FP 이동")(4524)을 포함한다. 적어도 하나의 실시예에서, 정수 레지스터 파일/바이패스 네트워크(4508) 및 부동 소수점 레지스터 파일/바이패스 네트워크(4510)는 본 명세서에서 "레지스터 파일들(4508, 4510)"이라고도 지칭된다. 적어도 하나의 실시예에서, AGUS들(4512 및 4514), 고속 ALU들(4516 및 4518), 저속 ALU(4520), 부동 소수점 ALU(4522), 및 부동 소수점 이동 유닛(4524)은 본 명세서에서 "실행 유닛들(4512, 4514, 4516, 4518, 4520, 4522, 및 4524)"이라고도 지칭된다. 적어도 하나의 실시예에서, 실행 블록(b11)은, 제한없이, 임의의 수(0 포함) 및 타입의 레지스터 파일들, 바이패스 네트워크들, 어드레스 생성 유닛들, 및 실행 유닛들을 임의의 조합으로 포함할 수 있다.

[0376] 적어도 하나의 실시예에서, 레지스터 파일들(4508, 4510)은 uop 스케줄러들(4502, 4504, 4506)과 실행 유닛들(4512, 4514, 4516, 4518, 4520, 4522, 및 4524) 사이에 배열될 수 있다. 적어도 하나의 실시예에서, 정수 레지스터 파일/바이패스 네트워크(4508)는 정수 연산들을 수행한다. 적어도 하나의 실시예에서, 부동 소수점 레지스터 파일/바이패스 네트워크(4510)는 부동 소수점 연산들을 수행한다. 적어도 하나의 실시예에서, 레지스터 파일들(4508, 4510) 각각은 레지스터 파일에 아직 기입되지 않은 방금 완료된 결과들을 새로운 종속 uop들로 바이패스 또는 포워딩할 수 있는 바이패스 네트워크를 포함할 수 있지만, 이에 제한되는 것은 아니다. 적어도 하나의 실시예에서, 레지스터 파일들(4508, 4510)은 서로 데이터를 통신할 수 있다. 적어도 하나의 실시예에서, 정수 레지스터 파일/바이패스 네트워크(4508)는 2개의 별개의 레지스터 파일, 즉 데이터의 하위 32-비트에 대한 하나의 레지스터 파일 및 데이터의 상위 32-비트에 대한 제2 레지스터 파일을 포함할 수 있지만 이에 제한되지 않는다. 적어도 하나의 실시예에서, 부동 소수점 레지스터 파일/바이패스 네트워크(4510)는, 부동 소수점 명령어들이 통상적으로 폭이 64 내지 128-비트인 피연산자들을 갖기 때문에, 128-비트 폭 엔트리들을 포함할 수 있지만, 이에 제한되지 않는다.

[0377] 적어도 하나의 실시예에서, 실행 유닛들(4512, 4514, 4516, 4518, 4520, 4522, 4524)은 명령어들을 실행할 수 있다. 적어도 하나의 실시예에서, 레지스터 파일들(4508, 4510)은 마이크로-명령어들이 실행할 필요가 있는 정수 및 부동 소수점 데이터 피연산자 값들을 저장한다. 적어도 하나의 실시예에서, 프로세서(4500)는 임의의 수 및 조합의 실행 유닛들(4512, 4514, 4516, 4518, 4520, 4522, 4524)을 포함할 수 있지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, 부동 소수점 ALU(4522) 및 부동 소수점 이동 유닛(4524)은 부동 소수점, MMX, SIMD, AVX 및 SSE, 또는 전문화된 머신 러닝 명령어들을 포함하는 다른 연산들을 실행할 수 있다. 적어도 하나의 실시예에서, 부동 소수점 ALU(4522)는, 계산, 제곱근, 및 나머지 마이크로 op들을 실행하는 64-비트 x 64-비트 부동 소수점 계산기를 포함할 수 있지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, 부동 소수점 값을 포함하는 명령어들은 부동 소수점 하드웨어로 핸들링될 수 있다. 적어도 하나의 실시예에서, ALU 연산들은 고속 ALU들(4516, 4518)에 전달될 수 있다. 적어도 하나의 실시예에서, 고속 ALU들(4516, 4518)은 1/2 클록 사이클의 유효 레이턴시로 고속 연산을 실행할 수 있다. 적어도 하나의 실시예에서, 가장 복잡한 정수 연산들은 저속 ALU(4520)로 가는데, 그 이유는 저속 ALU(4520)가, 제한없이, 곱셈기, 시프트, 플래그 로직, 및 분기 처리와 같은, 긴 레이턴시 타입의 연산들에 대한 정수 실행 하드웨어를 포함할 수 있기 때문이다. 적어도 하나의 실시예에서, 메모리 로드/저장 연산들은 AGUS(4512, 4514)에 의해 실행될 수 있다. 적어도 하나의 실시예에서, 고속 ALU(4516), 고속 ALU(4518), 및 저속 ALU(4520)는 64-비트 데이터 피연산자들에 대해 정수 연산들을 수행할 수 있다. 적어도 하나의 실시예에서, 고속 ALU(4516), 고속 ALU(4518), 및 저속 ALU(4520)는 16, 32, 128, 256 등을 포함한 다양한 데이터 비트 크기를 지원하도록 구현될 수 있다. 적어도 하나의 실시예에서, 부동 소수점 ALU(4522) 및 부동 소수점 이동 유닛(4524)은 다양한 폭들의 비트들을 갖는 피연산자들의 범위를 지원하도록 구현될 수 있다. 적어도 하나의 실시예에서, 부동 소수점 ALU(4522) 및 부동 소수점 이동 유닛(4524)은 SIMD 및 멀티미디어 명령어들과 관련하여 128-비트 폭의 패킹된 데이터 피연산자들에 대해 연산할 수 있다.

[0378] 적어도 하나의 실시예에서, uop 스케줄러들(4502, 4504, 4506)은 부모 로드가 실행을 완료하기 전에 종속 연산들을 디스패치한다. 적어도 하나의 실시예에서, uop들은 프로세서(4500)에서 추론적으로 스케줄링되고 실행될 수 있으므로, 프로세서(4500)는 또한 메모리 미스들을 핸들링하는 로직을 포함할 수 있다. 적어도 하나의 실시예에서, 데이터 로드가 데이터 캐시에서 미스되면, 스케줄러가 일시적으로 부정확한 데이터를 갖는 상태가 되게 하는 진행 중인 종속 연산들이 파이프라인 내에 있을 수 있다. 적어도 하나의 실시예에서, 리플레이 메커니즘은 부정확한 데이터를 이용하는 명령어들을 추적하고 재실행한다. 적어도 하나의 실시예에서, 종속 연산들이 리플레이될 필요가 있을 수 있고 독립적 연산들은 완료되도록 허용될 수 있다. 적어도 하나의 실시예에서, 프로세서의 적어도 하나의 실시예의 스케줄러 및 리플레이 메커니즘은 또한, 텍스트 스트링 비교 연산들을 위한 명령어 시퀀스들을 포착하도록 설계될 수 있다.

[0379] 적어도 하나의 실시예에서, 용어 "레지스터들"은 피연산자들을 식별하기 위한 명령어들의 일부로서 사용될 수

있는 온보드 프로세서 저장소 위치들을 지칭할 수 있다. 적어도 하나의 실시예에서, 레지스터들은 (프로그래머의 관점에서) 프로세서의 외부에서 이용할 수 있는 것들일 수 있다. 적어도 하나의 실시예에서, 레지스터들은 특정한 타입의 회로로 제한되지 않을 수 있다. 오히려, 적어도 하나의 실시예에서, 레지스터는 데이터를 저장하고, 데이터를 제공하고, 본 명세서에 설명된 기능들을 수행할 수 있다. 적어도 하나의 실시예에서, 본 명세서에 설명된 레지스터들은, 전용 물리적 레지스터들, 레지스터 리네이밍을 이용하여 동적으로 할당된 물리적 레지스터들, 전용 및 동적으로 할당된 물리적 레지스터들의 조합들 등과 같은, 임의의 수의 상이한 기법들을 이용하여 프로세서 내의 회로에 의해 구현될 수 있다. 적어도 하나의 실시예에서, 정수 레지스터들은 32-비트 정수 데이터를 저장한다. 적어도 하나의 실시예의 레지스터 파일은 또한, 패키징된 데이터를 위한 8개의 멀티미디어 SIMD 레지스터를 포함한다.

[0380] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)의 일부 또는 전부는 EXE 블록(4511) 및 도시되거나 도시되지 않은 다른 메모리 또는 레지스터에 통합될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 본 명세서에 설명된 훈련 및/또는 추론 기법은 EXE 블록(4511)에 예시된 ALU들 중 하나 이상을 사용할 수 있다. 더욱이, 본 명세서에 설명된 하나 이상의 머신 러닝 알고리즘, 신경망 아키텍처, 사용 사례, 또는 훈련 기법을 수행하도록 EXE 블록(4511)의 ALU를 구성하는 가중치 파라미터가 온-칩 또는 오프-칩 메모리 및/또는 레지스터(도시되어 있거나 도시되지 않음)에 저장될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 딥 러닝 애플리케이션 프로세서로 구현된다.

[0381] 도 46은 적어도 하나의 실시예에 따른 딥 러닝 애플리케이션 프로세서(4600)를 예시한다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서(4600)는, 딥 러닝 애플리케이션 프로세서(4600)에 의해 실행되는 경우, 딥 러닝 애플리케이션 프로세서(4600)로 하여금 본 개시내용 전반에 걸쳐 설명된 프로세스 및 기법의 일부 또는 전부를 수행하게 하는 명령어를 사용한다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서(4600)는 주문형 집적 회로(ASIC)이다. 적어도 하나의 실시예에서, 애플리케이션 프로세서(4600)는 하나 이상의 명령어 또는 양자 모듈을 수행한 결과로서 하드웨어에 "하드와이어드"되어 행렬 곱셈 연산들을 수행한다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서(4600)는, 제한없이, 처리 클러스터들(4610(1)-4610(12)), 칩간 링크들("ICL들")(4620(1)-4620(12)), 칩간 제어기들("ICC들")(4630(1)-4630(2)), 고대역폭 메모리 2세대("HBM2")(4640(1)-4640(4)), 메모리 제어기들("Mem Ctrlrs")(4642(1)-4642(4)), 고대역폭 메모리 물리적 계층("HBM PHY")(4644(1)-4644(4)), 관리-제어기 중앙 처리 유닛("관리-제어기 CPU")(4650), 직렬 주변기기 인터페이스, 집적 회로간, 및 범용 입력/출력 블록("SPI, I2C, GPIO")(4660), 주변기기 인터커넥트 익스프레스 제어기 및 직접 메모리 액세스 블록("PCIe 제어기 및 DMA")(4670), 및 16-레인 주변기기 인터커넥트 익스프레스 포트("PCI 익스프레스 x 16")(4680)를 포함한다.

[0382] 적어도 하나의 실시예에서, 처리 클러스터들(4610)은, 본 명세서에 설명된 것들을 포함하는, 하나 이상의 훈련 기법들에서 산출되는 가중치 파라미터들에 기초하는 추론 또는 예측 동작들을 포함하는, 딥 러닝 연산들을 수행할 수 있다. 적어도 하나의 실시예에서, 각각의 처리 클러스터(4610)는 임의의 수 및 타입의 프로세서를 포함할 수 있지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서(4600)는 임의의 수 및 타입의 처리 클러스터들(4600)을 포함할 수 있다. 적어도 하나의 실시예에서, 칩간 링크들(4620)은 양방향적이다. 적어도 하나의 실시예에서, 칩간 링크들(4620) 및 칩간 제어기들(4630)은 다수의 딥 러닝 애플리케이션 프로세서들(4600)이 하나 이상의 신경망들에서 구현되는 하나 이상의 머신 러닝 알고리즘들을 수행하는 것으로부터 기인하는 액티베이션 정보를 포함하는 정보를 교환할 수 있게 한다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서(4600)는 임의의 수(0을 포함함) 및 타입의 ICL들(4620) 및 ICC들(4630)을 포함할 수 있다.

[0383] 적어도 하나의 실시예에서, HBM2들(4640)은 총 32 기가바이트(GB)의 메모리를 제공한다. HBM2(4640(i))는 메모리 제어기(4642(i)) 및 HBM PHY(4644(i)) 양자 모두와 연관된다. 적어도 하나의 실시예에서, 임의의 수의 HBM2들(4640)은 임의의 타입 및 총량의 고대역폭 메모리를 제공할 수 있고, 임의의 수(0을 포함함) 및 타입의 메모리 제어기들(4642) 및 HBM PHY들(4644)과 연관될 수 있다. 적어도 하나의 실시예에서, SPI, I2C, GPIO(4660), PCIe 제어기 및 DMA(4670), 및/또는 PCIe(4680)는 임의의 수 및 타입의 통신 표준들을 임의의 기술적으로 실현 가능한 방식으로 가능하게 하는 임의의 수 및 타입의 블록들로 대체될 수 있다.

[0384] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어

있다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서는 딥 러닝 애플리케이션 프로세서(4600)에 제공된 정보를 예측 또는 추론하기 위해 신경망과 같은 머신 러닝 모델을 훈련하는데 사용된다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서(4600)는 다른 프로세서 또는 시스템에 의해 또는 딥 러닝 애플리케이션 프로세서(4600)에 의해 훈련된, 훈련된 머신 러닝 모델(예를 들어, 신경망)에 기초하여 정보를 추론 또는 예측하는데 사용된다. 적어도 하나의 실시예에서, 프로세서(4600)는 본 명세서에 설명된 하나 이상의 신경망 이용 사례를 수행하는데 이용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 뉴로모픽 프로세서로 구현된다.

[0385] 도 47은 적어도 하나의 실시예에 따른 뉴로모픽 프로세서(4700)의 블록도이다. 적어도 하나의 실시예에서, 뉴로모픽 프로세서(4700)는 뉴로모픽 프로세서(4700) 외부의 소스들로부터 하나 이상의 입력을 수신할 수 있다. 적어도 하나의 실시예에서, 이러한 입력들은 뉴로모픽 프로세서(4700) 내의 하나 이상의 뉴런(4702)으로 송신될 수 있다. 적어도 하나의 실시예에서, 뉴런(4702) 및 그 컴포넌트는 하나 이상의 산술 로직 유닛(ALU)을 포함하는 회로 또는 로직을 사용하여 구현될 수 있다. 적어도 하나의 실시예에서, 뉴로모픽 프로세서(4700)는 제한없이 수천 또는 수백만 개의 뉴런(4702)의 인스턴스를 포함할 수 있지만, 임의의 적합한 수의 뉴런(4702)이 사용될 수 있다. 적어도 하나의 실시예에서, 뉴런(4702)의 각각의 인스턴스는 뉴런 입력(4704) 및 뉴런 출력(4706)을 포함할 수 있다. 적어도 하나의 실시예에서, 뉴런(4702)은 뉴런(4702)의 다른 인스턴스의 입력에 송신될 수 있는 출력을 생성할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 뉴런 입력(4704)과 뉴런 출력(4706)은 시냅스(4708)를 통해 상호접속될 수 있다.

[0386] 적어도 하나의 실시예에서, 뉴런들(4702) 및 시냅스들(4708)은 뉴로모픽 프로세서(4700)가 뉴로모픽 프로세서(4700)에 의해 수신된 정보를 처리 또는 분석하도록 동작하도록 상호접속될 수 있다. 적어도 하나의 실시예에서, 뉴런(4702)은, 뉴런 입력(4704)을 통해 수신된 입력이 임계값을 초과할 때 출력 펄스(또는 "발화" 또는 "스파이크")를 송신할 수 있다. 적어도 하나의 실시예에서, 뉴런들(4702)은 뉴런 입력들(4704)에서 수신된 신호들을 합산 또는 통합할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 뉴런(4702)은 누설성 통합-및-발화 뉴런으로서 구현될 수 있고, 여기서 합("막 전위"로 지칭됨)이 임계값을 초과하면, 뉴런(4702)은 시그모이드 또는 임계값 함수와 같은 전달 함수를 사용하여 출력(또는 "발화")을 생성할 수 있다. 적어도 하나의 실시예에서, 누설성 통합-및-발화 뉴런은 뉴런 입력들(4704)에서 수신된 신호들을 막 전위로 합산할 수 있고, 또한 막 전위를 감소시키기 위해 감쇠 인자(또는 누설)를 적용할 수 있다. 적어도 하나의 실시예에서, 누설성 통합-및-발화 뉴런은 다수의 입력 신호들이 임계값을 초과하기에 충분히 빠르게(즉, 막 전위가 너무 낮게 감쇠하여 발화되기 전에) 뉴런 입력들(4704)에서 수신되는 경우에 발화될 수 있다. 적어도 하나의 실시예에서, 뉴런(4702)은, 입력을 수신하고, 입력을 막 전위에 통합하며, 막 전위를 감쇠시키는 회로 또는 로직을 이용하여 구현될 수 있다. 적어도 하나의 실시예에서, 입력들은 평균화될 수 있거나, 또는 임의의 다른 적합한 전달 함수가 사용될 수 있다. 또한, 적어도 하나의 실시예에서, 뉴런들(4702)은, 뉴런 입력(4704)에 전달 함수를 적용한 결과가 임계값을 초과할 때 뉴런 출력(4706)에서 출력 스파이크를 생성하는 비교기 회로들 또는 로직을 제한없이 포함할 수 있다. 적어도 하나의 실시예에서, 일단 뉴런(4702)이 발화하면, 예를 들어, 막 전위를 0 또는 다른 적합한 디폴트 값으로 리셋함으로써, 이전에 수신된 입력 정보를 무시할 수 있다. 적어도 하나의 실시예에서, 일단 막 전위가 0으로 리셋되면, 뉴런(4702)은 적합한 시간 기간(또는 불응 기간) 이후에 정상 동작을 재개할 수 있다.

[0387] 적어도 하나의 실시예에서, 뉴런들(4702)은 시냅스들(4708)을 통해 상호접속될 수 있다. 적어도 하나의 실시예에서, 시냅스들(4708)은 제1 뉴런(4702)의 출력으로부터 제2 뉴런(4702)의 입력으로 신호들을 송신하도록 동작할 수 있다. 적어도 하나의 실시예에서, 뉴런들(4702)은 시냅스(4708)의 2개 이상의 인스턴스를 통해 정보를 송신할 수 있다. 적어도 하나의 실시예에서, 뉴런 출력(4706)의 하나 이상의 인스턴스는, 시냅스(4708)의 인스턴스를 통해, 동일한 뉴런(4702)에서의 뉴런 입력(4704)의 인스턴스에 접속될 수 있다. 적어도 하나의 실시예에서, 시냅스(4708)의 인스턴스를 통해 송신될 출력을 생성하는 뉴런(4702)의 인스턴스는 시냅스(4708)의 그 인스턴스에 대해 "프리-시냅틱 뉴런"이라고 지칭될 수 있다. 적어도 하나의 실시예에서, 시냅스(4708)의 인스턴스를 통해 송신된 입력을 수신하는 뉴런(4702)의 인스턴스는 시냅스(4708)의 그 인스턴스에 대해 "포스트-시냅틱 뉴런"이라고 지칭될 수 있다. 뉴런(4702)의 인스턴스는 시냅스(4708)의 하나 이상의 인스턴스로부터 입력들을 수신할 수 있고, 또한 시냅스(4708)의 하나 이상의 인스턴스를 통해 출력들을 송신할 수 있기 때문에, 따라서, 뉴런(4702)의 단일 인스턴스는, 적어도 하나의 실시예에서, 시냅스들(4708)의 다양한 인스턴스들에 대해, "프리-시냅틱 뉴런" 및 "포스트-시냅틱 뉴런" 양자 모두일 수 있다.

[0388] 적어도 하나의 실시예에서, 뉴런들(4702)은 하나 이상의 계층으로 조직될 수 있다. 뉴런(4702)의 각각의 인스턴스는 하나 이상의 시냅스(4708)를 통해 하나 이상의 뉴런 입력(4704)으로 팬 아웃할 수 있는 하나의 뉴런 출력

(4706)을 가질 수 있다. 적어도 하나의 실시예에서, 제1 계층(4710)의 뉴런(4702)의 뉴런 출력(4706)은 제2 계층(4712)의 뉴런(4702)의 뉴런 입력(4704)에 접속될 수 있다. 적어도 하나의 실시예에서, 계층(4710)은 "피드-포워드 계층"이라 지칭될 수 있다. 적어도 하나의 실시예에서, 제1 계층(4710)의 인스턴스 내의 뉴런(4702)의 각각의 인스턴스는 제2 계층(4712) 내의 뉴런(4702)의 각각의 인스턴스로 팬 아웃할 수 있다. 적어도 하나의 실시예에서, 제1 계층(4710)은 "완전히 연결된 피드-포워드 계층"이라고 지칭될 수 있다. 적어도 하나의 실시예에서, 제2 계층(4712)의 인스턴스 내의 뉴런(4702)의 각각의 인스턴스는 제3 계층(4714) 내의 뉴런(4702)의 모든 인스턴스보다 적은 수의 인스턴스로 팬 아웃할 수 있다. 적어도 하나의 실시예에서, 제2 계층(4712)은 "최소하게 연결된 피드-포워드 계층"으로 지칭될 수 있다. 적어도 하나의 실시예에서, 제2 계층(4712)에서의 뉴런들(4702)은, (동일한) 제2 계층(4712)에서의 뉴런들(4702)을 포함한, 다수의 다른 계층들에서의 뉴런들(4702)로 팬 아웃할 수 있다. 적어도 하나의 실시예에서, 제2 계층(4712)은 "순환 계층"이라고 지칭될 수 있다. 뉴로모픽 프로세서(4700)는 최소하게 연결된 피드-포워드 계층들 및 완전히 연결된 피드-포워드 계층들 양자 모두를 포함하지만 이에 제한되지 않는 순환 계층들 및 피드-포워드 계층들의 임의의 적합한 조합을 포함할 수 있지만 이에 제한되지 않는다.

[0389] 적어도 하나의 실시예에서, 뉴로모픽 프로세서(4700)는, 제한없이, 시냅스(4708)를 뉴런들(4702)에 연결하기 위한 재구성가능한 인터커넥트 아키텍처 또는 전용 하드 와이어드 인터커넥트들을 포함할 수 있다. 적어도 하나의 실시예에서, 뉴로모픽 프로세서(4700)는, 신경망 토폴로지 및 뉴런 팬-인/아웃에 기초하여 필요에 따라 시냅스들이 상이한 뉴런들(4702)에 할당될 수 있게 하는 회로 또는 로직을 제한없이 포함할 수 있다. 예를 들어, 적어도 하나의 실시예에서, 시냅스들(4708)은 네트워크-온-칩과 같은 상호접속 패브릭을 이용하여, 또는 전용 접속들을 이용하여 뉴런들(4702)에 접속될 수 있다. 적어도 하나의 실시예에서, 시냅스 상호접속들 및 그 컴포넌트들은 회로 또는 로직을 이용하여 구현될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 처리 시스템으로 구현된다.

[0390] 도 48은 적어도 하나의 실시예에 따른 처리 시스템의 블록도이다. 적어도 하나의 실시예에서, 시스템(4800)은 하나 이상의 프로세서(4802) 및 하나 이상의 그래픽 프로세서(4808)를 포함하고, 단일 프로세서 데스크탑 시스템, 멀티프로세서 워크스테이션 시스템, 또는 많은 수의 프로세서들(4802) 또는 프로세서 코어들(4807)을 갖는 서버 시스템일 수 있다. 적어도 하나의 실시예에서, 시스템(4800)은 모바일, 핸드헬드, 또는 임베디드 디바이스들에서 사용하기 위한 시스템-온-칩(SoC) 집적 회로 내에 통합된 처리 플랫폼이다.

[0391] 적어도 하나의 실시예에서, 시스템(4800)은 서버 기반 게임 플랫폼, 게임 및 미디어 콘솔을 포함하는 게임 콘솔, 모바일 게임 콘솔, 핸드헬드 게임 콘솔 또는 온라인 게임 콘솔을 포함하거나 그 안에 통합될 수 있다. 적어도 하나의 실시예에서, 시스템(4800)은 이동 전화, 스마트 폰, 태블릿 계산 디바이스 또는 모바일 인터넷 디바이스이다. 적어도 하나의 실시예에서, 처리 시스템(4800)은 또한, 스마트 워치 착용형 디바이스, 스마트 안경류 디바이스, 증강 현실 디바이스, 또는 가상 현실 디바이스 같은 착용형 디바이스를 포함하거나, 이와 결합되거나, 이 내에 통합될 수 있다. 적어도 하나의 실시예에서, 처리 시스템(4800)은 하나 이상의 프로세서(4802) 및 하나 이상의 그래픽 프로세서(4808)에 의해 생성된 그래픽 인터페이스를 갖는 텔레비전 또는 셋톱 박스 디바이스이다.

[0392] 적어도 하나의 실시예에서, 하나 이상의 프로세서(4802) 각각은, 실행될 때, 시스템 및 사용자 소프트웨어에 대한 동작들을 수행하는 명령어들을 처리하는 하나 이상의 프로세서 코어(4807)를 포함한다. 적어도 하나의 실시예에서, 하나 이상의 프로세서 코어(4807) 각각은 특정 명령어 세트(4809)를 처리하도록 구성된다. 적어도 하나의 실시예에서, 명령어 세트(4809)는 CISC(Complex Instruction Set Computing), RISC(Reduced Instruction Set Computing), 또는 VLIW(Very Long Instruction Word)를 통한 계산을 용이하게 할 수 있다. 적어도 하나의 실시예에서, 프로세서 코어들(4807)은, 다른 명령어 세트들의 에뮬레이션을 용이하게 하는 명령어들을 포함할 수 있는, 상이한 명령어 세트(4809)를 각각 처리할 수 있다. 적어도 하나의 실시예에서, 프로세서 코어(4807)는 또한 디지털 신호 프로세서(DSP)와 같은 다른 처리 디바이스들을 포함할 수 있다.

[0393] 적어도 하나의 실시예에서, 프로세서(4802)는 캐시 메모리(4804)를 포함한다. 적어도 하나의 실시예에서, 프로세서(4802)는 단일 내부 캐시 또는 다수의 레벨의 내부 캐시를 가질 수 있다. 적어도 하나의 실시예에서, 캐시 메모리는 프로세서(4802)의 다양한 컴포넌트들 사이에서 공유된다. 적어도 하나의 실시예에서, 프로세서(4802)는 또한 공지된 캐시 일관성 기법들을 사용하여 프로세서 코어들(4807) 사이에 공유될 수 있는 외부 캐시(예를 들어, L3(Level-3) 캐시 또는 LLC(Last Level Cache))(도시되지 않음)를 사용한다. 적어도 하나의 실시예에서, 레지스터 파일(4806)은, 상이한 타입의 데이터를 저장하기 위한 상이한 타입의 레지스터(예를 들어, 정수 레지스터, 부동 소수점 레지스터, 상태 레지스터, 및 명령어 포인터 레지스터)를 포함할 수 있는 프로세서(4802)에

더 포함된다. 적어도 하나의 실시예에서, 레지스터 파일(4806)은 범용 레지스터들 또는 다른 레지스터들을 포함할 수 있다.

[0394] 적어도 하나의 실시예에서, 하나 이상의 프로세서(들)(4802)는 프로세서(4802)와 시스템(4800) 내의 다른 컴포넌트들 사이에서 어드레스, 데이터, 또는 제어 신호들과 같은 통신 신호들을 송신하기 위해 하나 이상의 인터페이스 버스(들)(4810)와 결합된다. 적어도 하나의 실시예에서, 인터페이스 버스(4810)는, 하나의 실시예에서, DMI(Direct Media Interface) 버스의 버전과 같은, 프로세서 버스일 수 있다. 적어도 하나의 실시예에서, 인터페이스(4810)는 DMI 버스로 제한되지 않으며, 하나 이상의 주변기기 인터커넥트 버스(예를 들어, PCI, PCI 익스프레스), 메모리 버스, 또는 다른 타입의 인터페이스 버스를 포함할 수 있다. 적어도 하나의 실시예에서, 프로세서(들)(4802)는 통합 메모리 제어기(4816) 및 플랫폼 제어기 허브(4830)를 포함한다. 적어도 하나의 실시예에서, 메모리 제어기(4816)는 메모리 디바이스와 시스템(4800)의 다른 컴포넌트들 사이의 통신을 용이하게 하는 한편, 플랫폼 제어기 허브(PCH)(4830)는 로컬 I/O 버스를 통해 I/O 디바이스들에 대한 접속들을 제공한다.

[0395] 적어도 하나의 실시예에서, 메모리 디바이스(4820)는 동적 랜덤 액세스 메모리(DRAM) 디바이스, 정적 랜덤 액세스 메모리(SRAM) 디바이스, 플래시 메모리 디바이스, 상변화 메모리 디바이스, 또는 프로세스 메모리로서 역할을 하기 위해 적합한 성능을 갖는 일부 다른 메모리 디바이스일 수 있다. 적어도 하나의 실시예에서, 메모리 디바이스(4820)는 시스템(4800)을 위한 시스템 메모리로서 동작하여, 하나 이상의 프로세서(4802)가 애플리케이션 또는 프로세스를 실행할 때 사용하기 위한 데이터(4822) 및 명령어들(4821)을 저장할 수 있다. 적어도 하나의 실시예에서, 메모리 제어기(4816)는 또한 그래픽 및 미디어 연산들을 수행하기 위해 프로세서들(4802) 내의 하나 이상의 그래픽 프로세서(4808)와 통신할 수 있는 선택적인 외부 그래픽 프로세서(4812)와 결합된다. 적어도 하나의 실시예에서, 디스플레이 디바이스(4811)는 프로세서(들)(4802)에 접속될 수 있다. 적어도 하나의 실시예에서, 디스플레이 디바이스(4811)는, 모바일 전자 디바이스 또는 랩탑 디바이스에서의와 같은 내부 디스플레이 디바이스 또는 디스플레이 인터페이스(예를 들어, DisplayPort 등)를 통해 부착된 외부 디스플레이 디바이스 중 하나 이상을 포함할 수 있다. 적어도 하나의 실시예에서, 디스플레이 디바이스(4811)는, 가상 현실(VR) 응용 또는 증강 현실(AR) 응용에서 이용하기 위한 입체 디스플레이 디바이스 같은 헤드 장착 디스플레이(HMD)를 포함할 수 있다.

[0396] 적어도 하나의 실시예에서, 플랫폼 제어기 허브(4830)는 주변기기들이 고속 I/O 버스를 통해 메모리 디바이스(4820) 및 프로세서(4802)에 접속할 수 있게 한다. 적어도 하나의 실시예에서, I/O 주변기기들은, 오디오 제어기(4846), 네트워크 제어기(4834), 펌웨어 인터페이스(4828), 무선 트랜시버(4826), 터치 센서들(4825), 데이터 저장 디바이스(4824)(예를 들어, 하드 디스크 드라이브, 플래시 메모리 등)를 포함하지만 이에 제한되지 않는다. 적어도 하나의 실시예에서, 데이터 저장 디바이스(4824)는 저장소 인터페이스(예를 들어, SATA)를 통해 또는 주변기기 인터커넥트 버스(예를 들어, PCI, PCI 익스프레스)와 같은 주변기기 버스를 통해 접속될 수 있다. 적어도 하나의 실시예에서, 터치 센서(4825)는, 터치 스크린 센서, 압력 센서, 또는 지문 센서를 포함할 수 있다. 적어도 하나의 실시예에서, 무선 트랜시버(4826)는 Wi-Fi 트랜시버, 블루투스 트랜시버, 또는 3G, 4G, 또는 롱 텀 에볼루션(LTE) 트랜시버와 같은 모바일 네트워크 트랜시버일 수 있다. 적어도 하나의 실시예에서, 펌웨어 인터페이스(4828)는 시스템 펌웨어와의 통신을 가능하게 하고, 예를 들어, UEFI(unified extensible firmware interface)일 수 있다. 적어도 하나의 실시예에서, 네트워크 제어기(4834)는 유선 네트워크로의 네트워크 접속을 가능하게 할 수 있다. 적어도 하나의 실시예에서, 고성능 네트워크 제어기(도시되지 않음)는 인터페이스 버스(4810)와 결합된다. 적어도 하나의 실시예에서, 오디오 제어기(4846)는 멀티-채널 고화질 오디오 제어기이다. 적어도 하나의 실시예에서, 시스템(4800)은 레저시(예를 들어, 개인 시스템 2(PS/2)) 디바이스들을 시스템에 결합하기 위한 선택적 레저시 I/O 제어기(4840)를 포함한다. 적어도 하나의 실시예에서, 플랫폼 제어기 허브(4830)는 또한 하나 이상의 USB(Universal Serial Bus) 제어기들(4842)에 접속하여, 키보드 및 마우스(4843) 조합들, 카메라(4844) 또는 다른 USB 입력 디바이스들과 같은 입력 디바이스들을 접속할 수 있다.

[0397] 적어도 하나의 실시예에서, 메모리 제어기(4816) 및 플랫폼 제어기 허브(4830)의 인스턴스는, 외부 그래픽 프로세서(4812)와 같은, 개별 외부 그래픽 프로세서에 통합될 수 있다. 적어도 하나의 실시예에서, 플랫폼 제어기 허브(4830) 및/또는 메모리 제어기(4816)는 하나 이상의 프로세서(들)(4802) 외부에 있을 수 있다. 예를 들어, 적어도 하나의 실시예에서, 시스템(4800)은 외부 메모리 제어기(4816) 및 플랫폼 제어기 허브(4830)를 포함할 수 있고, 이는 프로세서(들)(4802)와 통신하는 시스템 칩셋 내의 메모리 제어기 허브 및 주변기기 제어기 허브로서 구성될 수 있다.

[0398] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어

있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)의 일부 또는 전부는 그래픽 프로세서(4800)에 통합될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 본 명세서에 설명된 훈련 및/또는 추론 기법들은 3D 파이프라인(4812)에 구현된 ALU들 중 하나 이상을 사용할 수 있다. 더욱이, 적어도 하나의 실시예에서, 본 명세서에 설명된 추론 및/또는 훈련 동작들은 도 29a 또는 도 29b에 예시된 로직 이외의 로직을 이용하여 이루어질 수 있다. 적어도 하나의 실시예에서, 가중치 파라미터들은, 하나 이상의 머신 러닝 알고리즘, 신경망 아키텍처들, 이용 사례들, 또는 본 명세서에 설명된 훈련 기법들을 수행하도록 그래픽 프로세서(4800)의 ALU들을 구성하는 (도시되거나 도시되지 않은) 온-칩 또는 오프-칩 메모리 및/또는 레지스터들에 저장될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 프로세서를 이용하여 구현된다.

[0399] 도 49는 적어도 하나의 실시예에 따른, 하나 이상의 프로세서 코어(4902A-4902N), 통합 메모리 제어기(4914), 및 통합 그래픽 프로세서(4908)를 갖는 프로세서(4900)의 블록도이다. 적어도 하나의 실시예에서, 프로세서(4900)는 점선 박스들로 표현된 추가적인 코어(4902N)까지 및 이를 포함하는 추가적인 코어들을 포함할 수 있다. 적어도 하나의 실시예에서, 프로세서 코어들(4902A-4902N) 각각은 하나 이상의 내부 캐시 유닛(4904A-4904N)을 포함한다. 적어도 하나의 실시예에서, 각각의 프로세서 코어는 또한 하나 이상의 공유된 캐시 유닛들(4906)에 대한 액세스를 갖는다.

[0400] 적어도 하나의 실시예에서, 내부 캐시 유닛들(4904A-4904N) 및 공유된 캐시 유닛들(4906)은 프로세서(4900) 내의 캐시 메모리 계층구조를 나타낸다. 적어도 하나의 실시예에서, 캐시 메모리 유닛들(4904A 내지 4904N)은 각각의 프로세서 코어 내의 적어도 하나의 레벨의 명령어 및 데이터 캐시와, 레벨 2(L2), 레벨 3(L3), 레벨 4(L4), 또는 다른 레벨들의 캐시와 같은, 하나 이상의 레벨들의 공유된 중간 레벨 캐시를 포함할 수 있고, 여기서 외부 메모리 이전의 최상위 레벨의 캐시는 LLC로서 분류된다. 적어도 하나의 실시예에서, 캐시 일관성 로직은 다양한 캐시 유닛들(4906 및 4904A-4904N) 사이의 일관성을 유지한다.

[0401] 적어도 하나의 실시예에서, 프로세서(4900)는 또한 하나 이상의 버스 제어기 유닛들(4916)의 세트 및 시스템 에이전트 코어(4910)를 포함할 수 있다. 적어도 하나의 실시예에서, 하나 이상의 버스 제어기 유닛들(4916)은 하나 이상의 PCI 또는 PCI 익스프레스 버스들과 같은 주변기기 버스들의 세트를 관리한다. 적어도 하나의 실시예에서, 시스템 에이전트 코어(4910)는 다양한 프로세서 컴포넌트들에 대한 관리 기능을 제공한다. 적어도 하나의 실시예에서, 시스템 에이전트 코어(4910)는 다양한 외부 메모리 디바이스들(도시되지 않음)에 대한 액세스를 관리하기 위해 하나 이상의 통합 메모리 제어기(4914)를 포함한다.

[0402] 적어도 하나의 실시예에서, 프로세서 코어들(4902A-4902N) 중 하나 이상은 동시 멀티스레딩을 위한 지원을 포함한다. 적어도 하나의 실시예에서, 시스템 에이전트 코어(4910)는 멀티스레드 처리 동안 코어들(4902A-4902N)을 조정하고 동작시키기 위한 컴포넌트들을 포함한다. 적어도 하나의 실시예에서, 시스템 에이전트 코어(4910)는 전력 제어 유닛(PCU)을 더 포함할 수 있고, 이는 프로세서 코어들(4902A-4902N) 및 그래픽 프로세서(4908)의 하나 이상의 전력 상태를 조절하는 로직 및 컴포넌트들을 포함한다.

[0403] 적어도 하나의 실시예에서, 프로세서(4900)는 그래픽 처리 연산들을 실행하는 그래픽 프로세서(4908)를 더 포함한다. 적어도 하나의 실시예에서, 그래픽 프로세서(4908)는 공유된 캐시 유닛들(4906), 및 하나 이상의 통합 메모리 제어기(4914)를 포함하는 시스템 에이전트 코어(4910)와 결합된다. 적어도 하나의 실시예에서, 시스템 에이전트 코어(4910)는 또한 하나 이상의 결합된 디스플레이에 대한 그래픽 프로세서 출력을 구동하는 디스플레이 제어기(4911)를 포함한다. 적어도 하나의 실시예에서, 디스플레이 제어기(4911)는 또한 적어도 하나의 인터커넥트를 통해 그래픽 프로세서(4908)와 결합되는 개별 모듈일 수 있거나, 그래픽 프로세서(4908) 내에 통합될 수 있다.

[0404] 적어도 하나의 실시예에서, 링 기반 인터커넥트 유닛(4912)은 프로세서(4900)의 내부 컴포넌트들을 결합하는데 이용된다. 적어도 하나의 실시예에서, 포인트-투-포인트 인터커넥트, 스위칭형 인터커넥트, 또는 다른 기술들 같은 대안적인 인터커넥트 유닛이 이용될 수 있다. 적어도 하나의 실시예에서, 그래픽 프로세서(4908)는 I/O 링크(4913)를 통해 링 인터커넥트(4912)와 결합된다.

[0405] 적어도 하나의 실시예에서, I/O 링크(4913)는, 다양한 프로세서 컴포넌트들과 eDRAM 모듈과 같은 고성능 임베디드 메모리 모듈(4918) 사이의 통신을 용이하게 하는 온 패키지 I/O 인터커넥트를 포함하는, 다수의 다양한 I/O 인터커넥트들 중 적어도 하나를 나타낸다. 적어도 하나의 실시예에서, 프로세서 코어들(4902A-4902N) 및 그래픽 프로세서(4908) 각각은 임베디드 메모리 모듈들(4918)을 공유 최종 레벨 캐시로서 사용한다.

- [0406] 적어도 하나의 실시예에서, 프로세서 코어들(4902A-4902N)은 공통 명령어 세트 아키텍처를 실행하는 동종 코어들이다. 적어도 하나의 실시예에서, 프로세서 코어들(4902A-4902N)은 명령어 세트 아키텍처(ISA)의 관점에서 이질적이며, 여기서 프로세서 코어들(4902A-4902N) 중 하나 이상은 공통 명령어 세트를 실행하는 반면, 프로세서 코어들(4902A-4902N)의 하나 이상의 다른 코어는 공통 명령어 세트의 서브세트 또는 상이한 명령어 세트를 실행한다. 적어도 하나의 실시예에서, 프로세서 코어들(4902A-4902N)은 마이크로아키텍처의 관점에서 이질적이며, 여기서 비교적 더 높은 전력 소비를 갖는 하나 이상의 코어가 더 낮은 전력 소비를 갖는 하나 이상의 전력 코어와 결합한다. 적어도 하나의 실시예에서, 프로세서(4900)는 하나 이상의 칩 상에 또는 SoC 집적 회로로서 구현될 수 있다.
- [0407] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)의 일부 또는 전부는 그래픽 프로세서(4910)에 통합될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 본 명세서에 설명되는 훈련 및/또는 추론 기법들은 3D 파이프라인(4812), 그래픽 코어(들)(4915A), 공유 기능 로직(4916), 그래픽 코어(들)(4915B), 공유 기능 로직(4920) 또는 도 49의 다른 로직에서 구현되는 ALU들 중 하나 이상을 이용할 수 있다. 더욱이, 적어도 하나의 실시예에서, 본 명세서에 설명된 추론 및/또는 훈련 동작들은 도 29a 또는 도 29b에 예시된 로직 이외의 로직을 이용하여 이루어질 수 있다. 적어도 하나의 실시예에서, 가중치 파라미터들은, 하나 이상의 머신 러닝 알고리즘, 신경망 아키텍처들, 이용 사례들, 또는 본 명세서에 설명된 훈련 기법들을 수행하도록 그래픽 프로세서(4910)의 ALU들을 구성하는 (도시되거나 도시되지 않은) 온-칩 또는 오프-칩 메모리 및/또는 레지스터들에 저장될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 그래픽 프로세서로 구현된다.
- [0408] 도 50은 그래픽 프로세서(5000)의 블록도이고, 이는 개별 그래픽 처리 유닛일 수 있거나, 또는 복수의 처리 코어들과 통합된 그래픽 프로세서일 수 있다. 적어도 하나의 실시예에서, 그래픽 프로세서(5000)는 메모리 맵핑 I/O 인터페이스를 통해 그래픽 프로세서(5000) 상의 레지스터들과 통신하고, 커맨드들이 메모리에 배치된다. 적어도 하나의 실시예에서, 그래픽 프로세서(5000)는 메모리에 액세스하기 위한 메모리 인터페이스(5014)를 포함한다. 적어도 하나의 실시예에서, 메모리 인터페이스(5014)는 로컬 메모리, 하나 이상의 내부 캐시, 하나 이상의 공유 외부 캐시, 및/또는 시스템 메모리에 대한 인터페이스이다.
- [0409] 적어도 하나의 실시예에서, 그래픽 프로세서(5000)는 또한 디스플레이 출력 데이터를 디스플레이 디바이스(5020)에 구동하는 디스플레이 제어기(5002)를 포함한다. 적어도 하나의 실시예에서, 디스플레이 제어기(5002)는 디스플레이 디바이스(5020)에 대한 하나 이상의 오버레이 평면들에 대한 하드웨어 및 비디오 또는 사용자 인터페이스 엘리먼트들의 다수의 계층들의 구성을 포함한다. 적어도 하나의 실시예에서, 디스플레이 디바이스(5020)는 내부 또는 외부 디스플레이 디바이스일 수 있다. 적어도 하나의 실시예에서, 디스플레이 디바이스(5020)는 가상 현실(VR) 디스플레이 디바이스 또는 증강 현실(AR) 디스플레이 디바이스와 같은 헤드 장착 디스플레이 디바이스이다. 적어도 하나의 실시예에서, 그래픽 프로세서(5000)는, MPEG-2 같은 MPEG(Moving Picture Experts Group) 포맷, H.264/MPEG-4 AVC 같은 AVC(Advanced Video Coding) 포맷 뿐만 아니라 SMPTE(Society of Motion Picture & Television Engineers) 421M/VC-1, 및 JPEG 및 MJPEG(Motion JPEG) 포맷 같은 JPEG(Joint Photographic Experts Group) 포맷을 포함한 그러나 이것으로 제한되지 않는, 하나 이상의 미디어 인코딩 포맷으로, 이들로부터, 또는 이들 사이에서 미디어를 인코딩, 디코딩, 또는 트랜스코딩하는 비디오 코덱 엔진(5006)을 포함한다.
- [0410] 적어도 하나의 실시예에서, 그래픽 프로세서(5000)는, 예를 들어, 비트-경계 블록 전달들을 포함하는 2차원(2D) 래스터화기 동작들을 수행하기 위한 블록 이미지 전달(BLIT) 엔진(5004)을 포함한다. 그러나, 적어도 하나의 실시예에서, 2D 그래픽 연산들은 그래픽 처리 엔진(GPE)(5010)의 하나 이상의 컴포넌트를 이용하여 수행된다. 적어도 하나의 실시예에서, GPE(5010)는 3차원(3D) 그래픽 연산들 및 미디어 연산들을 포함하는 그래픽 연산들을 수행하기 위한 계산 엔진이다.
- [0411] 적어도 하나의 실시예에서, GPE(5010)는 3D 프리미티브 형상들(예를 들어, 직사각형, 삼각형 등)에 작용하는 처리 함수들을 이용하여 3차원 이미지들 및 장면들을 렌더링하는 것과 같은 3D 연산들을 수행하기 위한 3D 파이프라인(5012)을 포함한다. 3D 파이프라인(5012)은 다양한 태스크들을 수행하고/하거나 실행 스트림들을 3D/미디어 서브시스템(5015)에 생성하는 프로그램가능 및 고정 기능 엘리먼트들을 포함한다. 3D 파이프라인(5012)이 미디어 연산들을 수행하는데 사용될 수 있지만, 적어도 하나의 실시예에서, GPE(5010)는 또한 비디오 후처리 및 이

미지 강화와 같은 미디어 연산들을 수행하는데 사용되는 미디어 파이프라인(5016)을 포함한다.

- [0412] 적어도 하나의 실시예에서, 미디어 파이프라인(5016)은 비디오 코덱 엔진(5006) 대신에 또는 그를 대신하여 비디오 디코드 가속, 비디오 디인터레이싱(de-interlacing), 및 비디오 인코드 가속과 같은 하나 이상의 전문화된 미디어 연산을 수행하는 고정 기능 또는 프로그램가능 로직 유닛을 포함한다. 적어도 하나의 실시예에서, 미디어 파이프라인(5016)은 3D/미디어 서브시스템(5015) 상에서의 실행을 위해 스레드들을 생성하는 스레드 생성 유닛을 더 포함한다. 적어도 하나의 실시예에서, 생성된 스레드들은 3D/미디어 서브시스템(5015)에 포함된 하나 이상의 그래픽 실행 유닛들 상에서 미디어 연산들에 대한 계산들을 수행한다.
- [0413] 적어도 하나의 실시예에서, 3D/미디어 서브시스템(5015)은 3D 파이프라인(5012) 및 미디어 파이프라인(5016)에 의해 생성된 스레드들을 실행하기 위한 로직을 포함한다. 적어도 하나의 실시예에서, 3D 파이프라인(5012) 및 미디어 파이프라인(5016)은 스레드 실행 요청을 3D/미디어 서브시스템(5015)으로 전송하며, 이 3D/미디어 서브시스템(5015)은 다양한 요청을 중재하여 이용 가능한 스레드 실행 리소스로 디스패치하는 스레드 디스패치 로직을 포함한다. 적어도 하나의 실시예에서, 실행 리소스들은 3D 및 미디어 스레드들을 처리하기 위한 그래픽 실행 유닛들의 어레이를 포함한다. 적어도 하나의 실시예에서, 3D/미디어 서브시스템(5015)은 스레드 명령어 및 데이터에 대한 하나 이상의 내부 캐시를 포함한다. 적어도 하나의 실시예에서, 서브시스템(5015)은 또한 스레드들 사이에 데이터를 공유하고 출력 데이터를 저장하기 위해, 레지스터 및 어드레싱 가능 메모리를 포함하는, 공유된 메모리를 포함한다.
- [0414] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)의 일부 또는 전부는 그래픽 프로세서(5000)에 통합될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 본 명세서에 설명된 훈련 및/또는 추론 기법들은 3D 파이프라인(5012)에 구현된 ALU들 중 하나 이상을 사용할 수 있다. 더욱이, 적어도 하나의 실시예에서, 본 명세서에 설명된 추론 및/또는 훈련 동작들은 도 29a 또는 도 29b에 예시된 로직 이외의 로직을 이용하여 이루어질 수 있다. 적어도 하나의 실시예에서, 가중치 파라미터들은, 하나 이상의 머신 러닝 알고리즘, 신경망 아키텍처들, 이용 사례들, 또는 본 명세서에 설명된 훈련 기법들을 수행하도록 그래픽 프로세서(5000)의 ALU들을 구성하는 (도시되거나 도시되지 않은) 온-칩 또는 오프-칩 메모리 및/또는 레지스터들에 저장될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 그래픽 처리 시스템으로 구현된다.
- [0415] 도 51은 적어도 하나의 실시예에 따른 그래픽 프로세서의 그래픽 처리 엔진(5110)의 블록도이다. 적어도 하나의 실시예에서, 그래픽 처리 엔진(GPE)(5110)은 도 50에 도시된 GPE(5010)의 버전이다. 적어도 하나의 실시예에서, 미디어 파이프라인(5016)은 선택적이며, GPE(5110) 내에 명시적으로 포함되지 않을 수 있다. 적어도 하나의 실시예에서, 별개의 미디어 및/또는 이미지 프로세서가 GPE(5110)에 결합된다.
- [0416] 적어도 하나의 실시예에서, GPE(5110)는, 3D 파이프라인(5012) 및/또는 미디어 파이프라인(5016)에 커맨드 스트림을 제공하는 커맨드 스트리머(5103)에 결합되거나 이를 포함한다. 적어도 하나의 실시예에서, 커맨드 스트리머(5103)는 메모리에 결합되고, 이는 시스템 메모리, 또는 내부 캐시 메모리 및 공유된 캐시 메모리 중 하나 이상일 수 있다. 적어도 하나의 실시예에서, 커맨드 스트리머(5103)는 메모리로부터 커맨드들을 수신하고 커맨드들을 3D 파이프라인(5012) 및/또는 미디어 파이프라인(5016)에 전송한다. 적어도 하나의 실시예에서, 커맨드는 3D 파이프라인(5012) 및 미디어 파이프라인(5016)에 대한 커맨드를 저장하는 링 버퍼로부터 인출되는 명령어, 프리미티브 또는 마이크로-연산이다. 적어도 하나의 실시예에서, 링 버퍼는 또한 다수의 커맨드의 배치를 저장하는 배치 커맨드 버퍼(batch command buffer)를 더 포함할 수 있다. 적어도 하나의 실시예에서, 3D 파이프라인(5012)에 대한 커맨드는 또한 3D 파이프라인(5012)에 대한 정점 및 기하형상 데이터 및/또는 미디어 파이프라인(5016)에 대한 이미지 데이터 및 메모리 객체(이들로 제한되지 않음)와 같은 메모리에 저장된 데이터에 대한 참조를 포함할 수 있다. 적어도 하나의 실시예에서, 3D 파이프라인(5012) 및 미디어 파이프라인(5016)은 연산들을 수행함으로써 또는 하나 이상의 실행 스레드들을 그래픽 코어 어레이(5114)로 디스패치함으로써 커맨드들 및 데이터를 처리한다. 적어도 하나의 실시예에서, 그래픽 코어 어레이(5114)는 그래픽 코어들(예를 들어, 그래픽 코어(들)(5115A), 그래픽 코어(들)(5115B))의 하나 이상의 블록을 포함하며, 각각의 블록은 하나 이상의 그래픽 코어를 포함한다. 적어도 하나의 실시예에서, 각각의 그래픽 코어는, 도 29a 및 도 29b의 추론 및/또는 훈련 로직(2915)을 포함하는, 고정 기능 텍스처 처리 및/또는 머신 러닝 및 인공 지능 가속 로직뿐만 아니라, 또한 그래픽 및 계산 연산들을 수행하기 위한 범용 및 그래픽 특정 실행 로직을 포함하는 그래픽 실행 리소스들의 세트

를 포함한다.

- [0417] 적어도 하나의 실시예에서, 3D 파이프라인(5012)은, 명령어들을 처리하고 실행 스레드들을 그래픽 코어 어레이(5114)에 디스패치함으로써, 정점 셰이더들, 기하형상 셰이더들, 픽셀 셰이더들, 프래그먼트 셰이더들, 계산 셰이더들, 또는 다른 셰이더 프로그램들과 같은 하나 이상의 셰이더 프로그램을 처리하기 위한 고정 기능 및 프로그램가능 로직을 포함한다. 적어도 하나의 실시예에서, 그래픽 코어 어레이(5114)는 셰이더 프로그램들을 처리하는데 이용하기 위한 실행 리소스들의 통합된 블록을 제공한다. 적어도 하나의 실시예에서, 그래픽 코어 어레이(5114)의 그래픽 코어(들)(5115A-5115B) 내의 다목적 실행 로직(예를 들어, 실행 유닛들)은 다양한 3D API 셰이더 언어들에 대한 지원을 포함하고, 다수의 셰이더와 연관된 다수의 동시 실행 스레드를 실행할 수 있다.
- [0418] 적어도 하나의 실시예에서, 그래픽 코어 어레이(5114)는 비디오 및/또는 이미지 처리와 같은 미디어 기능들을 수행하기 위한 실행 로직을 또한 포함한다. 적어도 하나의 실시예에서, 실행 유닛들은 그래픽 처리 연산들 이외에 병렬 범용 계산 연산들을 수행하도록 프로그램가능 범용 로직을 추가로 포함한다.
- [0419] 적어도 하나의 실시예에서, 그래픽 코어 어레이(5114) 상에서 실행되는 스레드들에 의해 생성되는 출력 데이터는 통합 반환 버퍼(URB)(5118) 내의 메모리에 데이터를 출력할 수 있다. URB(5118)는 다수의 스레드에 대한 데이터를 저장할 수 있다. 적어도 하나의 실시예에서, URB(5118)는 그래픽 코어 어레이(5114) 상에서 실행되는 상이한 스레드들 사이에서 데이터를 전송하는 데 사용될 수 있다. 적어도 하나의 실시예에서, URB(5118)는 그래픽 코어 어레이(5114) 상의 스레드들과 공유 기능 로직(5120) 내의 고정 기능 로직 사이의 동기화를 위해 추가적으로 사용될 수 있다.
- [0420] 적어도 하나의 실시예에서, 그래픽 코어 어레이(5114)는 스케일러블하고, 그래서, 그래픽 코어 어레이(5114)는 GPE(5110)의 타겟 전력 및 성능 레벨에 기초하여 가변 수의 실행 유닛을 각각 갖는 가변 수의 그래픽 코어를 포함한다. 적어도 하나의 실시예에서, 실행 리소스들은 동적으로 스케일러블하고, 따라서, 실행 리소스들은 필요에 따라 가능화 또는 불가능화될 수 있다.
- [0421] 적어도 하나의 실시예에서, 그래픽 코어 어레이(5114)는 그래픽 코어 어레이(5114)의 그래픽 코어들 사이에 공유되는 다수의 리소스를 포함하는 공유 기능 로직(5120)에 결합된다. 적어도 하나의 실시예에서, 공유 기능 로직(5120)에 의해 수행되는 공유 기능들은 그래픽 코어 어레이(5114)에 전문화된 보충 기능을 제공하는 하드웨어 로직 유닛들로 구현된다. 적어도 하나의 실시예에서, 공유 기능 로직(5120)은 샘플러(5121), 수학(5122) 및 ITC(inter-thread communication)(5123) 로직을 포함하지만 이에 제한되는 것은 아니다. 적어도 하나의 실시예에서, 하나 이상의 캐시(들)(5125)는 공유 기능 로직(5120)에 포함되거나 또는 이에 결합된다.
- [0422] 적어도 하나의 실시예에서, 전문화된 기능에 대한 요구가 그래픽 코어 어레이(5114) 내에 포함되기에 불충분한 경우에 공유 기능이 사용된다. 적어도 하나의 실시예에서, 전문화된 함수의 단일 인스턴스가 공유 기능 로직(5120)에서 사용되고, 그래픽 코어 어레이(5114) 내의 다른 실행 리소스들 사이에서 공유된다. 적어도 하나의 실시예에서, 그래픽 코어 어레이(5114)에 의해 광범위하게 사용되는 공유 기능 로직(5120) 내의 특정 공유 기능들은 그래픽 코어 어레이(5114) 내의 공유 기능 로직(5116) 내에 포함될 수 있다. 적어도 하나의 실시예에서, 그래픽 코어 어레이(5114) 내의 공유 기능 로직(5116)은 공유 기능 로직(5120) 내의 일부 또는 모든 로직을 포함할 수 있다. 적어도 하나의 실시예에서, 공유 기능 로직(5120) 내의 모든 로직 엘리먼트들은 그래픽 코어 어레이(5114)의 공유 기능 로직(5116) 내에서 복제될 수 있다. 적어도 하나의 실시예에서, 공유 기능 로직(5120)은 그래픽 코어 어레이(5114) 내의 공유 기능 로직(5116)을 위해 제외된다.
- [0423] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)의 일부 또는 전부는 그래픽 프로세서(5110)에 통합될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 본 명세서에 설명되는 훈련 및/또는 추론 기법들은 3D 파이프라인(5112), 그래픽 코어(들)(5115A), 공유 기능 로직(5116), 그래픽 코어(들)(5115B), 공유 기능 로직(5120) 또는 도 51의 다른 로직에서 구현되는 ALU들 중 하나 이상을 이용할 수 있다. 더욱이, 적어도 하나의 실시예에서, 본 명세서에 설명된 추론 및/또는 훈련 동작들은 도 29a 또는 도 29b에 예시된 로직 이외의 로직을 이용하여 이루어질 수 있다. 적어도 하나의 실시예에서, 가중치 파라미터들은, 하나 이상의 머신 러닝 알고리즘, 신경망 아키텍처들, 이용 사례들, 또는 본 명세서에 설명된 훈련 기법들을 수행하도록 그래픽 프로세서(5110)의 ALU들을 구성하는 (도시되거나 도시되지 않은) 온-칩 또는 오프-칩 메모리 및/또는 레지스터들에 저장될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 그래픽 프로세서 코어로 구현된다.

- [0424] 도 52는 본 명세서에 설명된 적어도 하나의 실시예에 따른, 그래픽 프로세서 코어(5200)의 하드웨어 로직의 블록도이다. 적어도 하나의 실시예에서, 그래픽 프로세서 코어(5200)는 그래픽 코어 어레이 내에 포함된다. 적어도 하나의 실시예에서, 때때로 코어 슬라이스라고 지칭되는 그래픽 프로세서 코어(5200)는 모듈형 그래픽 프로세서 내의 하나 또는 다수의 그래픽 코어일 수 있다. 적어도 하나의 실시예에서, 그래픽 프로세서 코어(5200)는 하나의 그래픽 코어 슬라이스의 일레이고, 본 명세서에 설명된 그래픽 프로세서는 타겟 전력 및 성능 엔벨로프에 기초한 다수의 그래픽 코어 슬라이스를 포함할 수 있다. 적어도 하나의 실시예에서, 각각의 그래픽 코어(5200)는 범용 및 고정 기능 로직의 모듈형 블록들을 포함하는, 서브-슬라이스들이라고도 지칭되는, 다수의 서브코어들(5201A-5201F)과 결합된 고정 기능 블록(5230)을 포함할 수 있다.
- [0425] 적어도 하나의 실시예에서, 고정 기능 블록(5230)은, 예를 들어, 저성능 및/또는 저전력 그래픽 프로세서 구현들에서, 그래픽 프로세서(5200) 내의 모든 서브코어들에 의해 공유될 수 있는 기하형상/고정 기능 파이프라인(5236)을 포함한다. 적어도 하나의 실시예에서, 기하형상/고정 기능 파이프라인(5236)은 3D 고정 기능 파이프라인, 비디오 프론트엔드 유닛, 스투드 스투너 및 스투드 디스패처, 및 통합 반환 버퍼들을 관리하는 통합 반환 버퍼 관리자를 포함한다.
- [0426] 적어도 하나의 실시예에서, 고정 기능 블록(5230)은 또한 그래픽 SoC 인터페이스(5237), 그래픽 마이크로컨트롤러(5238), 및 미디어 파이프라인(5239)을 포함한다. 그래픽 SoC 인터페이스(5237)는 시스템-온-칩 집적 회로 내의 그래픽 코어(5200)와 다른 프로세서 코어들 사이의 인터페이스를 제공한다. 적어도 하나의 실시예에서, 그래픽 마이크로컨트롤러(5238)는 스투드 디스패치, 스케줄링 및 선점을 포함하는 그래픽 프로세서(5200)의 다양한 기능들을 관리하도록 구성가능한 프로그램가능 서브프로세서이다. 적어도 하나의 실시예에서, 미디어 파이프라인(5239)은 이미지 및 비디오 데이터를 포함하는 멀티미디어 데이터의 디코딩, 인코딩, 전처리, 및/또는 후처리를 용이하게 하는 로직을 포함한다. 적어도 하나의 실시예에서, 미디어 파이프라인(5239)은 서브코어들(5201-5201F) 내의 계산 또는 샘플링 로직에 대한 요청들을 통해 미디어 연산들을 구현한다.
- [0427] 적어도 하나의 실시예에서, SoC 인터페이스(5237)는 그래픽 코어(5200)가 범용 애플리케이션 프로세서 코어들(예를 들어, CPU들) 및/또는 공유 최종 레벨 캐시 메모리, 시스템 RAM, 및/또는 임베디드 온-칩 또는 온-패키지 DRAM과 같은 메모리 계층구조 엘리먼트들을 포함하는 SoC 내의 다른 컴포넌트들과 통신할 수 있게 한다. 적어도 하나의 실시예에서, SoC 인터페이스(5237)는 또한 카메라 이미징 파이프라인들과 같은 SoC 내의 고정 기능 디바이스들과의 통신을 가능하게 할 수 있고, 그래픽 코어(5200)와 SoC 내의 CPU들 사이에 공유될 수 있는 글로벌 메모리 원자들의 사용 및/또는 구현을 가능하게 한다. 적어도 하나의 실시예에서, SoC 인터페이스(5237)는 또한 그래픽 코어(5200)에 대한 전력 관리 제어를 구현할 수 있고, 그래픽 코어(5200)의 클록 도메인과 SoC 내의 다른 클록 도메인 사이의 인터페이스를 가능화할 수 있다. 적어도 하나의 실시예에서, SoC 인터페이스(5237)는 그래픽 프로세서 내의 하나 이상의 그래픽 코어들 각각에 커맨드들 및 명령어들을 제공하도록 구성되는 커맨드 스트리머 및 글로벌 스투드 디스패처로부터의 커맨드 버퍼들의 수신을 가능하게 한다. 적어도 하나의 실시예에서, 커맨드 및 명령어가 미디어 파이프라인(5239)으로 디스패치될 수 있고, 미디어 연산들이 수행되어야 할 때, 또는 그래픽 처리 연산들이 수행되어야 할 때의 기하형상 및 고정 기능 파이프라인(예를 들어, 기하형상 및 고정 기능 파이프라인(5236), 기하형상 및 고정 기능 파이프라인(5214))을 포함한다.
- [0428] 적어도 하나의 실시예에서, 그래픽 마이크로컨트롤러(5238)는 그래픽 코어(5200)에 대한 다양한 스케줄링 및 관리 태스크들을 수행하도록 구성될 수 있다. 적어도 하나의 실시예에서, 그래픽 마이크로컨트롤러(5238)는 서브코어들(5201A-5201F) 내의 실행 유닛(EU) 어레이들(5202A-5202F, 5204A-5204F) 내의 다양한 그래픽 병렬 엔진들에 대해 그래픽 및/또는 계산 작업부하 스케줄링을 수행할 수 있다. 적어도 하나의 실시예에서, 그래픽 코어(5200)를 포함하는 SoC의 CPU 코어 상에서 실행되는 호스트 소프트웨어는 적합한 그래픽 엔진에 대한 스케줄링 동작을 호출하는 다수의 그래픽 프로세서 도어벨 중 하나의 작업부하를 제출할 수 있다. 적어도 하나의 실시예에서, 스케줄링 동작들은 다음으로 실행할 작업부하를 결정하고, 작업부하를 커맨드 스트리머에 제출하고, 엔진 상에서 실행 중인 기존의 작업부하들을 선점하고, 작업부하의 진행상황을 모니터링하고, 작업부하가 완료될 때 호스트 소프트웨어에 통보하는 것을 포함한다. 적어도 하나의 실시예에서, 그래픽 마이크로컨트롤러(5238)는 또한 그래픽 코어(5200)에 대한 저전력 또는 유휴 상태들을 용이하게 하여, 운영 체제 및/또는 시스템 상의 그래픽 드라이버 소프트웨어와 독립적으로 저전력 상태 전이들에 걸쳐 그래픽 코어(5200) 내의 레지스터들을 저장 및 복원하는 능력을 그래픽 코어(5200)에 제공할 수 있다.
- [0429] 적어도 하나의 실시예에서, 그래픽 코어(5200)는 예시된 서브코어들(5201A-5201F)보다 더 많거나 더 적은, 최대 N개의 모듈러 서브코어를 가질 수 있다. N개의 서브코어의 각각의 세트에 대해, 적어도 하나의 실시예에서, 그래픽 코어(5200)는 또한 공유 기능 로직(5210), 공유 및/또는 캐시 메모리(5212), 기하형상/고정 기능 파이프라인

인(5214)뿐만 아니라, 또한 다양한 그래픽을 가속화하고 처리 연산들을 계산하기 위한 추가적인 고정 기능 로직(5216)을 포함할 수 있다. 적어도 하나의 실시예에서, 공유 기능 로직(5210)은 그래픽 코어(5200) 내의 각각의 N개의 서브코어에 의해 공유될 수 있는 로직 유닛들(예를 들어, 샘플러, 수학 및/또는 스레드-간 통신 로직)을 포함할 수 있다. 공유 및/또는 캐시 메모리(5212)는 그래픽 코어(5200) 내의 N개의 서브코어(5201A-5201F)에 대한 최종 레벨 캐시일 수 있고, 또한 다수의 서브코어에 의해 액세스 가능한 공유된 메모리로서 역할을 할 수 있다. 적어도 하나의 실시예에서, 기하형상/고정 기능 파이프라인(5214)은 고정 기능 블록(5230) 내의 기하형상/고정 기능 파이프라인(5236) 대신에 포함될 수 있으며, 동일하거나 유사한 로직 유닛들을 포함할 수 있다.

[0430] 적어도 하나의 실시예에서, 그래픽 코어(5200)는 그래픽 코어(5200)에 의해 사용되는 다양한 고정 기능 가속 로직을 포함할 수 있는 추가적인 고정 기능 로직(5216)을 포함한다. 적어도 하나의 실시예에서, 추가적인 고정 기능 로직(5216)은 위치 전용 셰이딩에서 사용하기 위한 추가적인 기하형상 파이프라인을 포함한다. 위치-단독 셰이딩에서는, 적어도 2개의 기하형상 파이프라인이 존재하는 반면, 전체 기하형상 파이프라인에서는 기하형상/고정 기능 파이프라인(5216, 5236) 내에, 추가적인 고정 기능 로직(5216) 내에 포함될 수 있는 추가적인 기하형상 파이프라인인 컬 파이프라인(cull pipeline)이 존재한다. 적어도 하나의 실시예에서, 컬 파이프라인은 전체 기하형상 파이프라인의 축소된 버전이다. 적어도 하나의 실시예에서, 전체 파이프라인 및 컬 파이프라인은 애플리케이션의 상이한 인스턴스들을 실행할 수 있고, 각각의 인스턴스는 별개의 컨텍스트를 갖는다. 적어도 하나의 실시예에서, 위치 전용 셰이딩은 폐기된 삼각형들의 긴 컬 런(long cull run)을 은닉할 수 있어서, 일부 경우에 셰이딩이 더 일찍 완료되는 것을 가능하게 한다. 예를 들어, 적어도 하나의 실시예에서, 추가적인 고정 기능 로직(5216) 내의 컬 파이프라인 로직은 메인 애플리케이션과 병렬로 위치 셰이더들을 실행할 수 있고, 일반적으로, 프레임 버퍼에 대한 픽셀들의 래스터화 및 렌더링을 수행하지 않고, 컬 파이프라인이 정점들의 위치 속성을 인출하고 셰이딩함에 따라, 전체 파이프라인보다 더 빠르게 중요한 결과들을 생성한다. 적어도 하나의 실시예에서, 컬 파이프라인은 생성된 중요 결과들을 이용하여 삼각형들이 컬링되는지 여부에 관계없이 모든 삼각형에 관련한 가시성 정보를 계산할 수 있다. 적어도 하나의 실시예에서, (이 경우 리플레이 파이프라인이라고 지칭될 수 있는) 전체 파이프라인은 가시성 정보를 소비하여 컬링된 삼각형들을 건너뛰고 최종적으로 래스터화 단계에 전달되는 가시적 삼각형들만을 셰이딩할 수 있다.

[0431] 적어도 하나의 실시예에서, 추가적인 고정 기능 로직(5216)은 또한 머신 러닝 훈련 또는 추론을 위한 최적화들을 포함하는 구현들을 위한 고정 기능 행렬 곱셈 로직과 같은 머신 러닝 가속 로직을 포함할 수 있다.

[0432] 적어도 하나의 실시예에서, 각각의 그래픽 서브코어(5201A-5201F) 내에는 그래픽 파이프라인, 미디어 파이프라인 또는 셰이더 프로그램들에 의한 요청들에 응답하여 그래픽, 미디어 및 계산 연산들을 수행하는 데 사용될 수 있는 실행 리소스들의 세트가 포함된다. 적어도 하나의 실시예에서, 그래픽 서브코어들(5201A-5201F)은 다수의 EU 어레이들(5202A-5202F, 5204A-5204F), 스레드 디스패치 및 스레드-간 통신(TD/IC) 로직(5203A-5203F), 3D (예를 들어, 텍스처) 샘플러(5205A-5205F), 미디어 샘플러(5206A-5206F), 셰이더 프로세서(5207A-5207F), 및 공유 로컬 메모리(SLM)(5208A-5208F)를 포함한다. EU 어레이들(5202A 내지 5202F, 5204A 내지 5204F) 각각은 다수의 실행 유닛들을 포함하고, 이들은 그래픽, 미디어, 또는 계산 셰이더 프로그램들을 포함하는, 그래픽, 미디어, 또는 계산 연산의 서비스에서 부동 소수점 및 정수/고정 소수점 로직 연산들을 수행할 수 있는 범용 그래픽 처리 유닛들이다. 적어도 하나의 실시예에서, TD/IC 로직(5203A-5203F)은 서브코어 내의 실행 유닛들에 대한 로컬 스레드 디스패치 및 스레드 제어 동작들을 수행하고, 서브코어의 실행 유닛들 상에서 실행되는 스레드들 사이의 통신을 용이하게 한다. 적어도 하나의 실시예에서, 3D 샘플러(5205A-5205F)는 텍스처 또는 다른 3D 그래픽 관련 데이터를 메모리로 판독할 수 있다. 적어도 하나의 실시예에서, 3D 샘플러는 구성된 샘플 상태 및 주어진 텍스처와 연관된 텍스처 포맷에 기초하여 텍스처 데이터를 상이하게 판독할 수 있다. 적어도 하나의 실시예에서, 미디어 샘플러(5206A-5206F)는 미디어 데이터와 연관된 타입 및 포맷에 기초하여 유사한 판독 동작들을 수행할 수 있다. 적어도 하나의 실시예에서, 각각의 그래픽 서브코어(5201A-5201F)는 대안적으로 통합 3D 및 미디어 샘플러를 포함할 수 있다. 적어도 하나의 실시예에서, 서브코어들(5201A-5201F) 각각 내의 실행 유닛들 상에서 실행되는 스레드들은, 스레드 그룹 내에서 실행되는 스레드들이 온-칩 메모리의 공통 풀을 사용하여 실행될 수 있게 하기 위해, 각각의 서브코어 내의 공유 로컬 메모리(5208A-5208F)를 사용할 수 있다.

[0433] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)의 일부 또는 전부는 그래픽 프로세서(5210)에 통합될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 본 명세서에 설명된 훈련 및/또는 추론 기법들은 3D 파이프라인(5210), 그래픽 마이크로컨트롤러(5238), 기하형상 & 고정 기능 파이프라인(5214 및 5236), 또는 도 49

의 다른 로직에 구현된 ALU들 중 하나 이상을 사용할 수 있다. 더욱이, 적어도 하나의 실시예에서, 본 명세서에 설명된 추론 및/또는 훈련 동작들은 도 29a 또는 도 29b에 예시된 로직 이외의 로직을 이용하여 이루어질 수 있다. 적어도 하나의 실시예에서, 가중치 파라미터들은, 하나 이상의 머신 러닝 알고리즘, 신경망 아키텍처들, 이용 사례들, 또는 본 명세서에 설명된 훈련 기법들을 수행하도록 그래픽 프로세서(5200)의 ALU들을 구성하는 (도시되거나 도시되지 않은) 온-칩 또는 오프-칩 메모리 및/또는 레지스터들에 저장될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 아래에 예시되고 설명되는 바와 같은 스레드 실행 로직으로 구현된다.

[0434] 도 53a 및 도 53b는 적어도 하나의 실시예에 따른 그래픽 프로세서 코어의 처리 엘리먼트들의 어레이를 포함하는 스레드 실행 로직(5300)을 예시한다. 도 53a는 스레드 실행 로직(5300)이 사용되는 적어도 하나의 실시예를 예시한다. 도 53b는 적어도 하나의 실시예에 따른 실행 유닛의 예시적인 내부 상세를 예시한다.

[0435] 도 53a에 예시된 바와 같이, 적어도 하나의 실시예에서, 스레드 실행 로직(5300)은 셰이더 프로세서(5302), 스레드 디스패처(5304), 명령어 캐시(5306), 복수의 실행 유닛(5308A-5308N)을 포함하는 스케일러블 실행 유닛 어레이, 샘플러(5310), 데이터 캐시(5312) 및 데이터 포트(5314)를 포함한다. 적어도 하나의 실시예에서, 스케일러블 실행 유닛 어레이는, 예를 들어, 작업부하의 계산 요건들에 기초하여 하나 이상의 실행 유닛들(예를 들어, 실행 유닛(5308A, 5308B, 5308C, 5308D, 내지 5308N-1 및 5308N) 중 임의의 것)을 가능화 또는 불능화함으로써 동적으로 스케일링할 수 있다. 적어도 하나의 실시예에서, 스케일러블 실행 유닛들은 각각의 실행 유닛에 링크되는 인터커넥트 패브릭을 통해 상호접속된다. 적어도 하나의 실시예에서, 스레드 실행 로직(5300)은 명령어 캐시(5306), 데이터 포트(5314), 샘플러(5310) 및 실행 유닛들(5308A-5308N) 중 하나 이상을 통해 시스템 메모리 또는 캐시 메모리와 같은 메모리에 대한 하나 이상의 접속을 포함한다. 적어도 하나의 실시예에서, 각각의 실행 유닛(예를 들어, 5308A)은 각각의 스레드에 대해 병렬로 다수의 데이터 엘리먼트를 처리하면서 다수의 동시 하드웨어 스레드를 실행할 수 있는 독립형 프로그램가능 범용 계산 유닛이다. 적어도 하나의 실시예에서, 실행 유닛들(5308A-5308N)의 어레이는 임의의 수의 개별 실행 유닛들을 포함하도록 스케일러블하다.

[0436] 적어도 하나의 실시예에서, 실행 유닛들(5308A-5308N)은 주로 셰이더 프로그램들을 실행하는데 이용된다. 적어도 하나의 실시예에서, 셰이더 프로세서(5302)는 다양한 셰이더 프로그램들을 처리하고, 셰이더 프로그램들과 연관된 실행 스레드들을 스레드 디스패처(5304)를 통해 디스패치할 수 있다. 적어도 하나의 실시예에서, 스레드 디스패처(5304)는 그래픽 및 미디어 파이프라인들로부터의 스레드 개시 요청들을 중재하고, 실행 유닛들(5308A-5308N) 내의 하나 이상의 실행 유닛들에서 요청된 스레드들을 인스턴스화하기 위한 로직을 포함한다. 예를 들어, 적어도 하나의 실시예에서, 기하형상 파이프라인은, 정점, 테셀레이션, 또는 기하형상 셰이더들을 처리를 위해 스레드 실행 로직에 디스패치할 수 있다. 적어도 하나의 실시예에서, 스레드 디스패처(5304)는 또한 셰이더 프로그램들을 실행하는 것으로부터의 런타임 스레드 생성 요청들을 처리할 수 있다.

[0437] 적어도 하나의 실시예에서, 실행 유닛들(5308A-5308N)은, 그래픽 라이브러리들(예를 들어, Direct 3D 및 OpenGL)로부터의 셰이더 프로그램들이 최소의 변환으로 실행되도록, 많은 표준 3D 그래픽 셰이더 명령어들에 대한 네이티브 지원을 포함하는 명령어 세트를 지원한다. 적어도 하나의 실시예에서, 실행 유닛들은, 정점 및 기하형상 처리(예를 들어, 정점 프로그램들, 기하형상 프로그램들, 정점 셰이더들), 픽셀 처리(예를 들어, 픽셀 셰이더들, 프래그먼트 셰이더들) 및 범용 처리(예를 들어, 계산 및 미디어 셰이더들)를 지원한다. 적어도 하나의 실시예에서, 하나 이상의 산술 로직 유닛(ALU)을 포함하는 실행 유닛들(5308A-5308N) 각각은 다중-발행 단일 명령어 다중 데이터(SIMD) 실행이 가능하고, 멀티스레드 연산은 더 높은 레이턴시 메모리 액세스에도 불구하고 효율적인 실행 환경을 가능하게 한다. 적어도 하나의 실시예에서, 각각의 실행 유닛 내의 각각의 하드웨어 스레드는 전용 고대역폭 레지스터 파일 및 연관된 독립 스레드 상태를 갖는다. 적어도 하나의 실시예에서, 실행은, 정수, 단정도 및 배정도 부동 소수점 연산들, SIMD 분기 능력, 논리적 연산들, 초월 연산들, 및 다른 잡다한 연산들이 가능한 파이프라인들에 대한 클럭당 다중 발행이다. 적어도 하나의 실시예에서, 메모리 또는 공유 기능들 중 하나로부터 데이터를 대기하는 동안, 실행 유닛들(5308A-5308N) 내의 종속성 로직은 요청된 데이터가 반환될 때까지 대기하는 스레드가 휴면하게 한다. 적어도 하나의 실시예에서, 대기 중인 스레드가 휴면 중인 동안, 하드웨어 리소스들은 다른 스레드들을 처리하는 데 전용될 수 있다. 예를 들어, 적어도 하나의 실시예에서, 정점 셰이더 연산과 연관된 지연 동안, 실행 유닛은, 픽셀 셰이더, 프래그먼트 셰이더, 또는 상이한 정점 셰이더를 포함한 또 다른 타입의 셰이더 프로그램에 대한 연산들을 수행할 수 있다.

[0438] 적어도 하나의 실시예에서, 실행 유닛들(5308A-5308N) 내의 각각의 실행 유닛은 데이터 엘리먼트들의 어레이에 관해 동작한다. 적어도 하나의 실시예에서, 데이터 엘리먼트의 수는 "실행 크기", 또는 명령어에 대한 채널의 수이다. 적어도 하나의 실시예에서, 실행 채널은, 명령어들 내의 데이터 엘리먼트 액세스, 마스크, 및 흐름 제

어를 위한 실행의 로직 유닛이다. 적어도 하나의 실시예에서, 채널 수는 특정한 그래픽 프로세서에 대한 물리적 산술 로직 유닛(ALU) 또는 부동 소수점 유닛(FPU)의 수와는 독립적일 수 있다. 적어도 하나의 실시예에서, 실행 유닛들(5308A-5308N)은 정수 및 부동 소수점 데이터 타입들을 지원한다.

[0439] 적어도 하나의 실시예에서, 실행 유닛 명령어 세트는 SIMD 명령어들을 포함한다. 적어도 하나의 실시예에서, 다양한 데이터 엘리먼트들은 레지스터에 패킹된 데이터 타입으로서 저장될 수 있고 실행 유닛은 엘리먼트들의 데이터 크기에 기초하여 다양한 엘리먼트들을 처리할 것이다. 예를 들어, 적어도 하나의 실시예에서, 256-비트 폭 벡터에 관해 동작할 때, 벡터의 256-비트는 레지스터에 저장되고, 실행 유닛은, 벡터에 관해 4개의 별개의 64-비트 패킹된 데이터 엘리먼트(Quad-Word(QW) 크기 데이터 엘리먼트들), 8개의 별개의 32-비트 패킹된 데이터 엘리먼트(더블 워드(DW) 크기 데이터 엘리먼트들), 16개의 별개의 16-비트 패킹된 데이터 엘리먼트들(Word(W) 크기 데이터 엘리먼트들), 또는 32개의 별개의 8-비트 데이터 엘리먼트(바이트(B) 크기 데이터 엘리먼트)로서 동작한다. 그러나, 적어도 하나의 실시예에서, 상이한 벡터 폭들 및 레지스터 크기들이 가능하다.

[0440] 적어도 하나의 실시예에서, 하나 이상의 실행 유닛은 융합된 EU들에 공통인 스레드 제어 로직(5307A-5307N)을 갖는 융합된 실행 유닛(5309A-5309N)으로 조합될 수 있다. 적어도 하나의 실시예에서, 다수의 EU가 한 EU 그룹으로 융합될 수 있다. 적어도 하나의 실시예에서, 융합된 EU 그룹 내의 각각의 EU는, 별개의 SIMD 하드웨어 스레드를 실행하도록 구성될 수 있다. 융합된 EU 그룹에서의 EU들의 수는 다양한 실시예들에 따라 달라질 수 있다. 적어도 하나의 실시예에서, SIMD8, SIMD16, 및 SIMD32를 포함한 그러나 이것으로 제한되지 않는 다양한 SIMD 폭이 EU마다 수행될 수 있다. 적어도 하나의 실시예에서, 각각의 융합된 그래픽 실행 유닛(5309A-5309N)은 적어도 2개의 실행 유닛을 포함한다. 예를 들어, 적어도 하나의 실시예에서, 융합된 실행 유닛(5309A)은 제1 EU(5308A), 제2 EU(5308B), 및 제1 EU(5308A) 및 제2 EU(5308B)에 공통인 스레드 제어 로직(5307A)을 포함한다. 적어도 하나의 실시예에서, 스레드 제어 로직(5307A)은 융합된 그래픽 실행 유닛(5309A) 상에서 실행되는 스레드들을 제어하여, 융합된 실행 유닛들(5309A-5309N) 내의 각각의 EU가 공통 명령어 포인터 레지스터를 이용하여 실행되는 것을 허용한다.

[0441] 적어도 하나의 실시예에서, 하나 이상의 내부 명령어 캐시(예를 들어, 5306)는 실행 유닛들을 위한 스레드 명령어들을 캐싱하기 위해 스레드 실행 로직(5300)에 포함된다. 적어도 하나의 실시예에서, 스레드 실행 동안 스레드 데이터를 캐싱하기 위해 하나 이상의 데이터 캐시(예를 들어, 5312)가 포함된다. 적어도 하나의 실시예에서, 샘플러(5310)가 포함되어 3D 동작을 위한 텍스처 샘플링과 미디어 연산을 위한 미디어 샘플링을 제공한다. 적어도 하나의 실시예에서, 샘플러(5310)는, 샘플링된 데이터를 실행 유닛에 제공하기 이전에 샘플링 프로세스 동안에 텍스처 또는 미디어 데이터를 처리하는 전문화된 텍스처 또는 미디어 샘플링 기능을 포함한다.

[0442] 실행 동안, 적어도 하나의 실시예에서, 그래픽 및 미디어 파이프라인들은 스레드 생성 및 디스패치 로직을 통해 스레드 개시 요청들을 스레드 실행 로직(5300)에 전송한다. 적어도 하나의 실시예에서, 기하학적 객체들의 그룹이 처리되어 픽셀 데이터로 래스터화되면, 셰이더 프로세서(5302) 내의 픽셀 프로세서 로직(예를 들어, 픽셀 셰이더 로직, 프래그먼트 셰이더 로직 등)이 호출되어, 출력 정보를 추가로 계산하고 결과들이 출력 표면들(예를 들어, 컬러 버퍼들, 깊이 버퍼들, 스텐실 버퍼들 등)에 기입되게 한다. 적어도 하나의 실시예에서, 픽셀 셰이더 또는 프래그먼트 셰이더는, 래스터화된 객체에 걸쳐 보간될 다양한 정점 속성들의 값들을 산출한다. 적어도 하나의 실시예에서, 셰이더 프로세서(5302) 내의 픽셀 프로세서 로직은 그 후 API(application programming interface)-공급 픽셀 또는 프래그먼트 셰이더 프로그램을 실행한다. 적어도 하나의 실시예에서, 셰이더 프로그램을 실행하기 위해, 셰이더 프로세서(5302)는 스레드 디스패처(5304)를 통해 스레드들을 실행 유닛(예를 들어, 5308A)에 디스패치한다. 적어도 하나의 실시예에서, 셰이더 프로세서(5302)는 샘플러(5310) 내의 텍스처 샘플링 로직을 사용하여 메모리에 저장된 텍스처 맵들 내의 텍스처 데이터에 액세스한다. 적어도 하나의 실시예에서, 텍스처 데이터 및 입력 기하형상 데이터에 관한 산술 연산들은, 각각의 기하학적 프래그먼트에 대한 픽셀 컬러 데이터를 계산하거나, 추가 처리로부터의 하나 이상의 픽셀을 폐기한다.

[0443] 적어도 하나의 실시예에서, 데이터 포트(5314)는 스레드 실행 로직(5300)이 그래픽 프로세서 출력 파이프라인 상에서의 추가 처리를 위해 처리된 데이터를 메모리에 출력하기 위한 메모리 액세스 메커니즘을 제공한다. 적어도 하나의 실시예에서, 데이터 포트(5314)는 데이터 포트를 통한 메모리 액세스를 위한 데이터를 캐싱하기 위해 하나 이상의 캐시 메모리(예를 들어, 데이터 캐시(5312))를 포함하거나 이에 결합된다.

[0444] 도 53b에 예시된 바와 같이, 적어도 하나의 실시예에서, 그래픽 실행 유닛(5308)은 명령어 인출 유닛(5337), 일반 레지스터 파일 어레이(GRF)(5324), 아키텍처 레지스터 파일 어레이(ARF)(5326), 스레드 중재기(5322), 전송 유닛(5330), 분기 유닛(5332), SIMD 부동 소수점 유닛들(FPU들)(5334)의 세트, 및 적어도 하나의 실시예에서 전

용 정수 SIMD ALU들(5335)의 세트를 포함할 수 있다. 적어도 하나의 실시예에서, GRF(5324) 및 ARF(5326)는 그래픽 실행 유닛(5308)에서 활성일 수 있는 각각의 동시 하드웨어 스레드와 연관된 일반 레지스터 파일들 및 아키텍처 레지스터 파일들의 세트를 포함한다. 적어도 하나의 실시예에서, 스레드별 아키텍처 상태는 ARF(5326)에서 유지되고, 반면에, 스레드 실행 동안 사용되는 데이터는 GRF(5324)에 저장된다. 적어도 하나의 실시예에서, 각각의 스레드에 대한 명령어 포인터들을 포함하는 각각의 스레드의 실행 상태는 ARF(5326) 내의 스레드 특정 레지스터들에 유지될 수 있다.

[0445] 적어도 하나의 실시예에서, 그래픽 실행 유닛(5308)은 SMT(Simultaneous Multi-Threading) 및 IMT(fine-grained Interleaved Multi-Threading)의 조합인 아키텍처를 갖는다. 적어도 하나의 실시예에서, 아키텍처는 동시 스레드들의 타겟 수 및 실행 유닛당 레지스터들의 수에 기초하여 설계 시간에 미세 튜닝될 수 있는 모듈러 구성을 갖고, 여기서, 실행 유닛 리소스들은 다수의 동시 스레드를 실행하는 데 사용되는 로직에 걸쳐 분할된다.

[0446] 적어도 하나의 실시예에서, 그래픽 실행 유닛(5308)은 각각이 상이한 명령어일 수 있는 다수의 명령어를 동시-발행할 수 있다. 적어도 하나의 실시예에서, 그래픽 실행 유닛 스레드(5308)의 스레드 중재기(5322)는 실행을 위해 전송 유닛(5330), 분기 유닛(5342) 또는 SIMD FPU(들)(5334) 중 하나에 명령어들을 디스패치할 수 있다. 적어도 하나의 실시예에서, 각각의 실행 스레드는 GRF(5324) 내의 128개의 범용 레지스터에 액세스할 수 있고, 여기서 각각의 레지스터는 32-비트 데이터 엘리먼트의 SIMD 8-엘리먼트 벡터로서 액세스가능한 32 바이트를 저장할 수 있다. 적어도 하나의 실시예에서, 각각의 실행 유닛 스레드는 GRF(5324) 내의 4 Kbytes에 액세스하지만, 실시예들은 그렇게 제한되지 않고, 더 많거나 더 적은 레지스터 리소스들이 다른 실시예들에서 제공될 수 있다. 적어도 하나의 실시예에서, 최대 7개의 스레드가 동시에 실행될 수 있지만, 실행 유닛당 스레드의 수도 역시 실시예들에 따라 달라질 수 있다. 7개의 스레드가 4 Kbytes에 액세스할 수 있는 적어도 하나의 실시예에서, GRF(5324)는 총 28 Kbytes를 저장할 수 있다. 적어도 하나의 실시예에서, 유연한 어드레싱 모드들은 레지스터들이 함께 어드레싱되는 것을 허용하여 효과적으로 더 넓은 레지스터를 구축하거나 스트라이드 직사각형 블록 데이터 구조들을 나타낼 수 있다.

[0447] 적어도 하나의 실시예에서, 메모리 연산들, 샘플러 동작들, 및 다른 더 긴 레이턴시 시스템 통신들은 전송 유닛(5330)을 통과하는 메시지에 의해 실행되는 "전송" 명령어들을 통해 디스패치된다. 적어도 하나의 실시예에서, 분기 명령어들은 SIMD 발산 및 최종 수렴을 용이하게 하기 위해 전용 분기 유닛(5332)에 디스패치된다.

[0448] 적어도 하나의 실시예에서, 그래픽 실행 유닛(5308)은 부동 소수점 연산들을 수행하기 위해 하나 이상의 SIMD 부동 소수점 유닛(FPU(들))(5334)을 포함한다. 적어도 하나의 실시예에서, FPU(들)(5334)는 또한 정수 계산을 지원한다. 적어도 하나의 실시예에서, FPU(들)(5334)는 최대 M개의 수의 32-비트 부동 소수점(또는 정수) 연산을 SIMD 실행하거나, 최대 2M 16-비트 정수 또는 16-비트 부동 소수점 연산을 SIMD 실행할 수 있다. 적어도 하나의 실시예에서, FPU(들) 중 적어도 하나는 고처리량 초월 수학 함수 및 배정도 64-비트 부동 소수점을 지원하는 확장된 수학 능력을 제공한다. 적어도 하나의 실시예에서, 8-비트 정수 SIMD ALU들(5335)의 세트가 또한 존재하고, 머신 러닝 계산들과 연관된 동작들을 수행하도록 구체적으로 최적화될 수 있다.

[0449] 적어도 하나의 실시예에서, 그래픽 실행 유닛(5308)의 다수의 인스턴스의 어레이들은 그래픽 서브코어 그룹화(예를 들어, 서브-슬라이스)에서 인스턴스화될 수 있다. 적어도 하나의 실시예에서, 실행 유닛(5308)은 복수의 실행 채널에 걸쳐 명령어들을 실행할 수 있다. 적어도 하나의 실시예에서, 그래픽 실행 유닛(5308) 상에서 실행되는 각각의 스레드는 상이한 채널 상에서 실행된다.

[0450] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 추론 및/또는 훈련 로직(2915)의 일부 또는 전부는 실행 로직(5300)에 통합될 수 있다. 더욱이, 적어도 하나의 실시예에서, 본 명세서에 설명된 추론 및/또는 훈련 동작들은 도 29a 또는 도 29b에 예시된 로직 이외의 로직을 이용하여 이루어질 수 있다. 적어도 하나의 실시예에서, 가중치 파라미터들은 본 명세서에 설명된 하나 이상의 머신 러닝 알고리즘, 신경망 아키텍처, 사용 사례, 또는 훈련 기법을 수행하도록 실행 로직(5300)의 ALU들을 구성하는 온-칩 또는 오프-칩 메모리 및/또는 레지스터들(도시되거나 도시되지 않음)에 저장될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 병렬 처리 유닛으로 구현된다.

[0451] 도 54는 적어도 하나의 실시예에 따른 병렬 처리 유닛("PPU")(5400)을 예시한다. 적어도 하나의 실시예에서, PPU(5400)는, PPU(5400)에 의해 실행되는 경우, PPU(5400)로 하여금 본 개시내용 전반에 걸쳐 설명된 프로세스

들 및 기법들의 일부 또는 전부를 수행하게 하는 머신 관독가능 코드로 구성된다. 적어도 하나의 실시예에서, PPU(5400)는 하나 이상의 집적 회로 디바이스 상에 구현되고 다수의 스레드 상의 컴퓨터 관독가능한 명령어들(머신 관독가능 명령어들을 또는 단순히 명령어들로도 지칭됨)을 병렬로 처리하도록 설계된 레이턴시 은닉 기법으로서 멀티스레딩을 이용하는 멀티스레드 프로세서이다. 적어도 하나의 실시예에서, 스레드는 실행 스레드를 지칭하고, PPU(5400)에 의해 실행되도록 구성된 명령어들의 세트의 인스턴스화이다. 적어도 하나의 실시예에서, PPU(5400)는 액정 디스플레이("LCD") 디바이스와 같은 디스플레이 디바이스 상에 디스플레이할 2차원("2D") 이미지 데이터를 생성하기 위해 3차원("3D") 그래픽 데이터를 처리하기 위한 그래픽 렌더링 파이프라인을 구현하도록 구성되는 그래픽 처리 유닛("GPU")이다. 적어도 하나의 실시예에서, PPU(5400)는 선형 대수 연산들 및 머신 러닝 연산들과 같은 계산을 수행하는 데 이용된다. 도 54는 단지 예시적인 목적을 위한 예시적인 병렬 프로세서를 예시하며, 본 개시내용의 범위 내에서 고려되는 프로세서 아키텍처들의 비제한적인 예를 들어 해석되어야 하며, 임의의 적합한 프로세서를 채용하여 이를 보완 및/또는 대체할 수 있다.

[0452] 적어도 하나의 실시예에서, 하나 이상의 PPU(5400)는 고성능 계산("HPC"), 데이터 센터, 및 머신 러닝 애플리케이션들을 가속화하도록 구성된다. 적어도 하나의 실시예에서, PPU(5400)는 다음의 비제한적인 예들을 포함하는 딥 러닝 시스템들 및 애플리케이션들을 가속하도록 구성된다: 자율 차량 플랫폼들, 딥 러닝, 고정밀 음성, 이미지, 텍스트 인식 시스템들, 지능형 비디오 분석, 분자 시뮬레이션들, 약물 발견, 질병 진단, 일기 예보, 빅 데이터 분석, 천문학, 분자 역학 시뮬레이션, 재무 모델링, 로봇 공학, 공장 자동화, 실시간 언어 번역, 온라인 검색 최적화, 및 개인화된 사용자 추천 등.

[0453] 적어도 하나의 실시예에서, PPU(5400)는 입력/출력("I/O") 유닛(5406), 프론트엔드 유닛(5410), 스케줄러 유닛(5412), 작업 분배 유닛(5414), 허브(5416), 크로스바("Xbar")(5420), 하나 이상의 일반 처리 클러스터("GPC")(5418) 및 하나 이상의 파티션 유닛("메모리 파티션 유닛")(5422)을 포함하지만 이에 제한되지 않는다. 적어도 하나의 실시예에서, PPU(5400)는 하나 이상의 고속 GPU 인터커넥트("GPU 인터커넥트")(5408)을 통해 호스트 프로세서 또는 다른 PPU들(5400)에 접속된다. 적어도 하나의 실시예에서, PPU(5400)는 인터커넥트(5402)를 통해 호스트 프로세서 또는 다른 주변기기 디바이스들에 접속된다. 적어도 하나의 실시예에서, PPU(5400)는 하나 이상의 메모리 디바이스("메모리")(5404)를 포함하는 로컬 메모리에 접속된다. 적어도 하나의 실시예에서, 메모리 디바이스들(5404)은, 제한없이, 하나 이상의 동적 랜덤 액세스 메모리("DRAM") 디바이스들을 포함한다. 적어도 하나의 실시예에서, 하나 이상의 DRAM 디바이스들은 고대역폭 메모리("HBM") 서브시스템들로서 구성되고 및/또는 구성가능하고, 다수의 DRAM 다이들이 각각의 디바이스 내에 적층된다.

[0454] 적어도 하나의 실시예에서, 고속 GPU 인터커넥트(5408)는 하나 이상의 중앙 처리 유닛들("CPU들")과 조합된 하나 이상의 PPU들(5400)을 스케일링하고 포함하며, PPU들(5400)과 CPU들 사이의 캐시 일관성을 지원하며, CPU 마스터링을 하기 위해 시스템들에 의해 사용되는 유선 기반 다중 차선 통신 링크를 지칭할 수 있다. 적어도 하나의 실시예에서, 데이터 및/또는 커맨드들은 고속 GPU 인터커넥트(5408)에 의해 허브(5416)를 통해 하나 이상의 복사 엔진, 비디오 인코더, 비디오 디코더, 전력 관리 유닛 및 도 54에 명시적으로 예시되지 않을 수 있는 다른 컴포넌트들과 같은 PPU(5400)의 다른 유닛들로/로부터 송신된다.

[0455] 적어도 하나의 실시예에서, I/O 유닛(5406)은 시스템 버스(5402)를 통해 호스트 프로세서(도 54에 도시되지 않음)로부터 통신들(예를 들어, 커맨드들, 데이터)을 송신 및 수신하도록 구성된다. 적어도 하나의 실시예에서, I/O 유닛(5406)은 시스템 버스(5402)를 통해 직접 또는 메모리 브리지와 같은 하나 이상의 중간 디바이스를 통해 호스트 프로세서와 통신한다. 적어도 하나의 실시예에서, I/O 유닛(5406)은 시스템 버스(5402)를 통해 PPU들(5400) 중 하나 이상과 같은 하나 이상의 다른 프로세서와 통신할 수 있다. 적어도 하나의 실시예에서, I/O 유닛(5406)은 PCIe(Peripheral Component Interconnect Express) 버스를 통한 통신을 위한 PCIe 인터페이스를 구현한다. 적어도 하나의 실시예에서, I/O 유닛(5406)은 외부 디바이스들과 통신하기 위한 인터페이스들을 구현한다.

[0456] 적어도 하나의 실시예에서, I/O 유닛(5406)은 시스템 버스(5402)를 통해 수신된 패킷들을 디코딩한다. 적어도 하나의 실시예에서, 적어도 일부 패킷들은 PPU(5400)로 하여금 다양한 동작들을 수행하게 하도록 구성된 커맨드들을 나타낸다. 적어도 하나의 실시예에서, I/O 유닛(5406)은 커맨드들에 의해 지정된 바와 같이 디코딩된 커맨드들을 PPU(5400)의 다양한 다른 유닛들로 송신한다. 적어도 하나의 실시예에서, 커맨드들은 프론트엔드 유닛(5410)으로 송신되고/되거나, 허브(5416) 또는 하나 이상의 복사 엔진, 비디오 인코더, 비디오 디코더, 전력 관리 유닛 등과 같은 PPU(5400)의 다른 유닛들로 송신된다(도 54에는 명시적으로 예시되지 않음). 적어도 하나의 실시예에서, I/O 유닛(5406)은 PPU(5400)의 다양한 로직 유닛들 사이에서 통신을 라우팅하도록 구성된다.

- [0457] 적어도 하나의 실시예에서, 호스트 프로세서에 의해 실행되는 프로그램은 처리를 위해 PPU(5400)에 작업부하들을 제공하는 버퍼에 커맨드 스트림을 인코딩한다. 적어도 하나의 실시예에서, 작업부하는 명령어들 및 이들 명령어들에 의해 처리될 데이터를 포함한다. 적어도 하나의 실시예에서, 버퍼는 호스트 프로세서 및 PPU(5400) 양자 모두에 의해 액세스(예를 들어, 판독/기입)될 수 있는 메모리 내의 영역이며, 호스트 인터페이스 유닛은 I/O 유닛(5406)에 의해 시스템 버스(5402)를 통해 송신되는 메모리 요청들을 통해 시스템 버스(5402)에 접속된 시스템 메모리 내의 버퍼에 액세스하도록 구성될 수 있다. 적어도 하나의 실시예에서, 호스트 프로세서는 커맨드 스트림을 버퍼에 기입한 다음, 커맨드 스트림의 시작에 대한 포인터를 PPU(5400)에 송신하여, 프론트엔드 유닛(5410)이 하나 이상의 커맨드 스트림에 대한 포인터들을 수신하고 하나 이상의 커맨드 스트림을 관리하며, 커맨드 스트림들로부터 커맨드들을 판독하고 커맨드들을 PPU(5400)의 다양한 유닛들에 포워딩하게 한다.
- [0458] 적어도 하나의 실시예에서, 프론트엔드 유닛(5410)은 하나 이상의 커맨드 스트림에 의해 정의된 태스크들을 처리하도록 다양한 GPC들(5418)을 구성하는 스케줄러 유닛(5412)에 결합된다. 적어도 하나의 실시예에서, 스케줄러 유닛(5412)은 스케줄러 유닛(5412)에 의해 관리되는 다양한 태스크들과 관련된 상태 정보를 추적하도록 구성되고, 여기서 상태 정보는 태스크가 GPC들(5418) 중 어느 것에 할당되는지, 태스크가 활성화인지 또는 비활성인지, 태스크와 연관된 우선순위 레벨 등을 나타낼 수 있다. 적어도 하나의 실시예에서, 스케줄러 유닛(5412)은 하나 이상의 GPC(5418) 상에서의 복수의 태스크의 실행을 관리한다.
- [0459] 적어도 하나의 실시예에서, 스케줄러 유닛(5412)은 GPC들(5418) 상에서의 실행을 위해 태스크들을 디스패치하도록 구성된 작업 분배 유닛(5414)에 결합된다. 적어도 하나의 실시예에서, 작업 분배 유닛(5414)은 스케줄러 유닛(5412)으로부터 수신된 다수의 스케줄링된 태스크들을 추적하고, 작업 분배 유닛(5414)은 GPC들(5418) 각각에 대한 계류 중인 태스크 풀 및 활성 태스크 풀을 관리한다. 적어도 하나의 실시예에서, 계류 중인 태스크 풀은 특정한 GPC(5418)에 의해 처리되도록 할당된 태스크들을 포함하는 다수의 슬롯(예를 들어, 32개의 슬롯)을 포함하고; 활성 태스크 풀은 GPC들(5418)에 의해 능동적으로 처리되고 있는 태스크들에 대한 다수의 슬롯(예를 들어, 4개의 슬롯)을 포함하되, GPC들(5418) 중 하나가 태스크의 실행을 완료할 때, 그 태스크가 GPC(5418)에 대한 활성 태스크 풀로부터 추출되고 계류 중인 태스크 풀로부터의 다른 태스크들 중 하나가 GPC(5418) 상에서의 실행을 위해 선택되고 스케줄링되게 할 수 있다. 적어도 하나의 실시예에서, 데이터 종속성이 해결되기를 대기하는 동안과 같이, 활성 태스크가 GPC(5418) 상에서 유희인 경우, 그 후, 활성 태스크는 GPC(5418)로부터 추출되고, 계류 중인 태스크 풀로 반환되는 한편, 계류 중인 태스크 풀 내의 다른 태스크가 선택되어 GPC(5418) 상에서의 실행을 위해 스케줄링된다.
- [0460] 적어도 하나의 실시예에서, 작업 분배 유닛(5414)은 XBar(5420)을 통해 하나 이상의 GPC(5418)와 통신한다. 적어도 하나의 실시예에서, XBar(5420)은 PPU(5400)의 유닛들 중 다수를 PPU(5400)의 다른 유닛들에 결합하는 인터커넥트 네트워크이고, 작업 분배 유닛(5414)을 특정 GPC(5418)에 결합하도록 구성될 수 있다. 적어도 하나의 실시예에서, PPU(5400)의 하나 이상의 다른 유닛도 허브(5416)를 통해 XBar(5420)에 접속될 수 있다.
- [0461] 적어도 하나의 실시예에서, 태스크들은 스케줄러 유닛(5412)에 의해 관리되고, 작업 분배 유닛(5414)에 의해 GPC들(5418) 중 하나로 디스패치된다. GPC(5418)는 태스크를 처리하고 결과들을 생성하도록 구성된다. 적어도 하나의 실시예에서, 결과는, XBar(5420)을 통해 상이한 GPC(5418)로 라우팅되거나 메모리(5404)에 저장되는, GPC(5418) 내의 다른 태스크에 의해 소비될 수 있다. 적어도 하나의 실시예에서, 결과는 파티션 유닛(5422)을 통해 메모리(5404)에 기입될 수 있고, 이는 메모리(5404)로/로부터 데이터를 판독 및 기입하기 위한 메모리 인터페이스를 구현한다. 적어도 하나의 실시예에서, 결과들은 고속 GPU 인터커넥트(5408)를 통해 다른 PPU(5404) 또는 CPU로 송신될 수 있다. 적어도 하나의 실시예에서, PPU(5400)는 PPU(5400)에 결합된 개별 및 개별 메모리 디바이스들(5404)의 수와 동일한 수 U의 파티션 유닛(5422)을 포함하지만 이에 제한되지 않는다. 적어도 하나의 실시예에서, 파티션 유닛(5422)은 도 56과 관련하여 본 명세서에서 더 상세히 설명된다.
- [0462] 적어도 하나의 실시예에서, 호스트 프로세서는 호스트 프로세서 상에서 실행되는 하나 이상의 애플리케이션이 PPU(5400) 상에서의 실행을 위한 동작들을 스케줄링하는 것을 가능하게 하는 애플리케이션 프로그래밍 인터페이스(API)를 구현하는 드라이버 커널을 실행한다. 적어도 하나의 실시예에서, 다수의 계산 애플리케이션은 PPU(5400)에 의해 동시에 실행되며, PPU(5400)는 다수의 계산 애플리케이션에 대한 격리, 서비스 품질("QoS") 및 독립 어드레스 공간들을 제공한다. 적어도 하나의 실시예에서, 애플리케이션은 드라이버 커널이 PPU(5400)에 의한 실행을 위한 하나 이상의 태스크를 생성하고 드라이버 커널이 PPU(5400)에 의해 처리되는 하나 이상의 스트림에 태스크들을 출력하게 하는 명령어들(예를 들어, API 호출들의 형태로)을 생성한다. 적어도 하나의 실시예에서, 각각의 태스크는 워프(warp)라고 지칭될 수 있는 관련 스레드들의 하나 이상의 그룹을 포함한다. 적어도 하나의 실시예에서, 워프는 병렬로 실행될 수 있는 복수의 관련된 스레드(예를 들어, 32개의 스레드)를 포함

한다. 적어도 하나의 실시예에서, 협력 스레드들이란, 태스크를 수행하고 공유된 메모리를 통해 데이터를 교환하는 명령어들을 포함하는 복수의 스레드를 지칭할 수 있다. 적어도 하나의 실시예에서, 스레드들 및 협력 스레드들은 도 56과 관련하여 적어도 하나의 실시예에 따라 더 상세히 설명된다.

[0463] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서는 PPU(5400)에 제공되는 정보를 예측 또는 추론하기 위해 신경망과 같은 머신 러닝 모델을 훈련하는데 사용된다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서(5400)는 다른 프로세서 또는 시스템에 의해 또는 PPU(5400)에 의해 훈련된, 훈련된 머신 러닝 모델(예를 들어, 신경망)에 기초하여 정보를 추론 또는 예측하는데 사용된다. 적어도 하나의 실시예에서, PPU(5400)는 본 명세서에 설명되는 하나 이상의 신경망 사용 사례들을 수행하는 데 사용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 컴퓨터 시스템으로 구현된다.

[0464] 도 55는 적어도 하나의 실시예에 따른, 범용 처리 클러스터("GPC")(5500)를 예시한다. 적어도 하나의 실시예에서, GPC(5500)는 도 54의 GPC(5418)이다. 적어도 하나의 실시예에서, 각각의 GPC(5500)는 작업을 처리하기 위한 다수의 하드웨어 유닛들을 포함하지만, 이에 제한되지 않고, 각각의 GPC(5500)는 파이프라인 관리자(5502), 프리-래스터 연산 유닛("PROP")(5504), 래스터 엔진(5508), 태스크 분배 크로스바("WDX")(5516), 메모리 관리 유닛("MMU")(5518), 하나 이상의 데이터 처리 클러스터들("DPC들")(5506), 및 부품들의 임의의 적합한 조합을 포함하지만, 이에 제한되지 않는다.

[0465] 적어도 하나의 실시예에서, GPC(5500)의 동작은 파이프라인 관리자(5502)에 의해 제어된다. 적어도 하나의 실시예에서, 파이프라인 관리자(5502)는 GPC(5500)에 할당된 태스크들을 처리하기 위한 하나 이상의 DPC(5506)의 구성을 관리한다. 적어도 하나의 실시예에서, 파이프라인 관리자(5502)는 그래픽 렌더링 파이프라인의 적어도 일부를 구현하도록 하나 이상의 DPC(5506) 중 적어도 하나를 구성한다. 적어도 하나의 실시예에서, DPC(5506)는 프로그램가능 스트리밍 멀티-프로세서("SM")(5514) 상에서 정점 셰이더 프로그램을 실행하도록 구성된다. 적어도 하나의 실시예에서, 파이프라인 관리자(5502)는, 적어도 하나의 실시예에서, 작업 분배 유닛으로부터 수신된 패킷들을 GPC(5500) 내의 적합한 로직 유닛들로 라우팅하도록 구성되고, 일부 패킷들은 PROP(5504) 및/또는 래스터 엔진(5508) 내의 고정 기능 하드웨어 유닛들로 라우팅될 수 있는 반면, 다른 패킷들은 프리미티브 엔진(5512) 또는 SM(5514)에 의한 처리를 위해 DPC들(5506)로 라우팅될 수 있다. 적어도 하나의 실시예에서, 파이프라인 관리자(5502)는 신경망 모델 및/또는 계산 파이프라인을 구현하도록 DPC들(5506) 중 적어도 하나를 구성한다.

[0466] 적어도 하나의 실시예에서, PROP 유닛(5504)은 적어도 하나의 실시예에서 래스터 엔진(5508) 및 DPC(5506)에 의해 생성된 데이터를 도 54와 관련하여 위에서 더 상세히 설명된 파티션 유닛(5422) 내의 래스터 연산("ROP") 유닛으로 라우팅하도록 구성된다. 적어도 하나의 실시예에서, PROP 유닛(5504)은, 컬러 혼합에 대한 최적화를 수행하고, 픽셀 데이터를 조직화하며, 어드레스 변환을 수행하는 등을 수행하도록 구성된다. 적어도 하나의 실시예에서, 래스터 엔진(5508)은, 제한없이, 적어도 하나의 실시예에서, 다양한 래스터 연산들을 수행하도록 구성된 다수의 고정 기능 하드웨어 유닛들을 포함하고, 래스터 엔진(5508)은, 제한없이, 설정 엔진, 개략 래스터 엔진, 컬링 엔진, 클리핑 엔진, 미세 래스터 엔진, 타일 통합 엔진, 및 이들의 임의의 적합한 조합을 포함한다. 적어도 하나의 실시예에서, 설정 엔진은 변환된 정점들을 수신하고 정점들에 의해 정의된 기하학적 프리미티브와 연관된 평면 방정식들을 생성하고; 평면 방정식들은 프리미티브에 대한 커버리지 정보(예를 들어, 타일에 대한 x, y 커버리지 마스크)를 생성하기 위해 개략 래스터 엔진에 송신되고; 개략 래스터 엔진의 출력은 z-테스트에 실패한 프리미티브와 연관된 프래그먼트들이 컬링되는 컬링 엔진에 전송되고, 시야 절두체(viewing frustum) 외부에 놓인 프래그먼트들이 클리핑되는 클리핑 엔진에 송신된다. 적어도 하나의 실시예에서, 클리핑 및 컬링에서 살아남은 프래그먼트들은 미세 래스터 엔진에 전달되어 설정 엔진에 의해 생성된 평면 방정식들에 기초하여 픽셀 프래그먼트들에 대한 속성들을 생성한다. 적어도 하나의 실시예에서, 래스터 엔진(5508)의 출력은 DPC(5506) 내에 구현된 프래그먼트 셰이더와 같은 임의의 적합한 엔티티에 의해 처리될 프래그먼트들을 포함한다.

[0467] 적어도 하나의 실시예에서, GPC(5500)에 포함된 각각의 DPC(5506)는, 제한없이, M-파이프 제어기("MPC")(5510); 프리미티브 엔진(5512); 하나 이상의 SM(5514); 및 이들의 임의의 적합한 조합을 포함한다. 적어도 하나의 실시예에서, MPC(5510)는 DPC(5506)의 동작을 제어하고, 파이프라인 관리자(5502)로부터 수신된 패킷들을 DPC(5506) 내의 적합한 유닛들로 라우팅한다. 적어도 하나의 실시예에서, 정점과 연관된 패킷은 메모리

로부터 정점과 연관된 정점 속성들을 인출하도록 구성된 프리미티브 엔진(5512)에 라우팅되고; 대조적으로, 세이더 프로그램과 연관된 패킷은 SM(5514)에 송신될 수 있다.

[0468] 적어도 하나의 실시예에서, SM(5514)은 다수의 스레드들에 의해 표현되는 태스크들을 처리하도록 구성되어 있는 프로그램가능 스트리밍 프로세서(이들로 제한되지 않음)를 포함한다. 적어도 하나의 실시예에서, SM(5514)은 멀티스레드이고 특정한 스레드 그룹으로부터의 복수의 스레드(예를 들어, 32개의 스레드)를 동시에 실행하도록 구성되고, 스레드 그룹(예를 들어, 워프) 내의 각각의 스레드가 동일한 명령어 세트에 기초하여 상이한 데이터 세트를 처리하도록 구성된 SIMD(Single-Instruction, Multiple-Data) 아키텍처를 구현한다. 적어도 하나의 실시예에서, 스레드 그룹 내의 모든 스레드는 동일한 명령어들을 실행한다. 적어도 하나의 실시예에서, SM(5514)은, "SIMT(Single-Instruction, Multiple Thread)" 아키텍처를 구현하며, 여기서 스레드 그룹 내의 각각의 스레드는 동일한 명령어 세트에 기초하여 상이한 데이터 세트를 처리하도록 구성되지만, 스레드 그룹 내의 개별 스레드들은 실행 동안에 발산하는 것이 허용된다. 적어도 하나의 실시예에서, 프로그램 카운터, 호출 스택, 및 실행 상태가 각각의 워프에 대해 유지되어, 워프 내의 스레드들이 발산할 때 워프들과 워프들 내의 직렬 실행 사이의 동시성을 가능케한다. 또 다른 실시예에서, 프로그램 카운터, 호출 스택, 및 실행 상태가 각각의 개별 스레드에 대해 유지되어, 워프들 내부 및 워프들 사이에서 모든 스레드들 사이에 동일한 동시성을 가능케한다. 적어도 하나의 실시예에서, 실행 상태가 각각의 개별 스레드에 대해 유지되고 동일한 명령어들을 실행하는 스레드들은 더 양호한 효율성을 위해 수렴되고 병렬로 실행될 수 있다. SM(5514)의 적어도 하나의 실시예가 본 명세서에서 더 상세히 설명된다.

[0469] 적어도 하나의 실시예에서, MMU(5518)는 GPC(5500)와 메모리 파티션 유닛(예를 들어, 도 54의 파티션 유닛(5422)) 사이의 인터페이스를 제공하고, MMU(5518)는 가상 어드레스들의 물리적 어드레스들로의 변환, 메모리 보호, 및 메모리 요청들의 중재를 제공한다. 적어도 하나의 실시예에서, MMU(5518)는 가상 어드레스의 메모리 내의 물리적 어드레스로의 변환을 수행하기 위한 하나 이상의 변환 색인 버퍼("TLB")를 제공한다.

[0470] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서는, 신경망 같은 머신 러닝 모델을 훈련하여, GPC(5500)에 제공된 정보를 예측 또는 추론하는데 이용된다. 적어도 하나의 실시예에서, GPC(5500)는 다른 프로세서 또는 시스템에 의해 또는 GPC(5500)에 의해 훈련된, 훈련된 머신 러닝 모델(예를 들어, 신경망)에 기초하여 정보를 추론 또는 예측하는 데 사용된다. 적어도 하나의 실시예에서, 본 명세서에 설명된 하나 이상의 신경망 이용 사례를 수행하기 위해 GPC(5500)가 이용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같은 메모리 파티션 유닛으로 구현된다.

[0471] 도 56은 적어도 하나의 실시예에 따른 병렬 처리 유닛("PPU")의 메모리 파티션 유닛(5600)을 예시한다. 적어도 하나의 실시예에서, 메모리 파티션 유닛(5600)은, 제한없이, ROP(Raster Operations) 유닛(5602); 레벨 2("L2") 캐시(5604); 메모리 인터페이스(5606); 및 이들의 임의의 적합한 조합을 포함한다. 메모리 인터페이스(5606)는 메모리에 결합된다. 메모리 인터페이스(5606)는 고속 데이터 전달을 위해 32, 64, 128, 1024-비트 데이터 버스 등을 구현할 수 있다. 적어도 하나의 실시예에서, PPU는 파티션 유닛(5600)의 쌍마다 하나의 메모리 인터페이스(5606)씩 U개의 메모리 인터페이스(5606)를 통합하며, 파티션 유닛(5600)의 각각의 쌍은 대응하는 메모리 디바이스에 접속된다. 예를 들어, 적어도 하나의 실시예에서, PPU는 고대역폭 메모리 스택들 또는 그래픽 터블 데이터 레이트, 버전 5, 동기식 동적 랜덤 액세스 메모리("GDDR5 SDRAM")와 같은 최대 Y개의 메모리 디바이스에 접속될 수 있다.

[0472] 적어도 하나의 실시예에서, 메모리 인터페이스(5606)는 고대역폭 메모리 2세대("HBM2") 메모리 인터페이스를 구현하고 Y는 U의 절반과 같다. 적어도 하나의 실시예에서, HBM2 메모리 스택들은 PPU와 동일한 물리적 패키지에 위치하여, 종래의 GDDR5 SDRAM 시스템에 비해 상당한 전력 및 면적 절약을 제공한다. 적어도 하나의 실시예에서, 각각의 HBM2 스택은, 제한없이, 4개의 메모리 다이를 포함하고 Y는 4이며, 각각의 HBM2 스택은 총 8개 채널 및 1024-비트의 데이터 버스 폭을 위해 다이당 2개의 128-비트 채널을 포함한다. 적어도 하나의 실시예에서, 메모리는 데이터를 보호하기 위해 단일 에러 정정 이중 에러 검출("SECCDED"; Single-Error Correcting Double-Error Detecting) 에러 정정 코드("ECC")를 지원한다. ECC는 데이터 손상에 민감한 계산 애플리케이션들에 대해 더 높은 신뢰성을 제공한다.

[0473] 적어도 하나의 실시예에서, PPU는 멀티-레벨 메모리 계층구조를 구현한다. 적어도 하나의 실시예에서, 메모리 파티션 유닛(5600)은 중앙 처리 유닛("CPU") 및 PPU 메모리를 위한 단일의 통합된 가상 어드레스 공간을 제공하

기 위해 통합된 메모리를 지원하여, 가상 메모리 시스템들 사이의 데이터 공유를 가능하게 한다. 적어도 하나의 실시예에서, 다른 프로세서들 상에 위치하는 메모리에 대한 PPU에 의한 액세스의 빈도는 메모리 페이지들이 더 자주 액세스하는 페이지들인 PPU의 물리 메모리로 이동되는 것을 보장하기 위해 추적된다. 적어도 하나의 실시예에서, 고속 GPU 인터커넥트(5408)는 PPU가 CPU의 페이지 테이블들에 직접 액세스하는 것을 허용하고 PPU에 의한 CPU 메모리로의 완전한 액세스를 제공하는 어드레스 변환 서비스들을 지원한다.

[0474] 적어도 하나의 실시예에서, 복사 엔진은 다수의 PPU들 사이에서 또는 PPU들과 CPU들 사이에서 데이터를 전달한다. 적어도 하나의 실시예에서, 복사 엔진들은 페이지 테이블들에 맵핑되지 않은 어드레스들에 대한 페이지 결합들을 생성할 수 있고 이때, 메모리 파티션 유닛(5600)은 페이지 결합들을 서비스하고 어드레스들을 페이지 테이블에 맵핑한 후, 복사 엔진이 전달을 수행한다. 적어도 하나의 실시예에서, 메모리는 다수의 프로세서들 사이의 다수의 복사 엔진 동작들을 위해 고정(즉, 페이지 불가)되어, 이용 가능한 메모리를 상당히 감소시킨다. 적어도 하나의 실시예에서, 하드웨어 페이지 결합시에, 메모리 페이지들이 상주하는지 여부에 관계없이 어드레스들이 복사 엔진들에 전달될 수 있고, 복사 프로세스는 투명하다.

[0475] 도 54의 메모리(5404) 또는 다른 시스템 메모리로부터의 데이터는 메모리 파티션 유닛(5600)에 의해 인출되고 L2 캐시(5604)에 저장되며, L2 캐시는 온-칩에 위치하고 적어도 하나의 실시예에 따라 다양한 GPC들 사이에서 공유된다. 적어도 하나의 실시예에서, 각각의 메모리 파티션 유닛(5600)은 대응하는 메모리 디바이스와 연관된 L2 캐시의 적어도 일부를 포함하지만 이에 제한되지 않는다. 적어도 하나의 실시예에서, 하위 레벨 캐시들은 GPC들 내에서 다양한 유닛으로 구현된다. 적어도 하나의 실시예에서, SM들(5514) 각각은 레벨 1("L1") 캐시를 구현할 수 있고, 여기서 L1 캐시는 특정한 SM(5514)에 전용인 사설 메모리이고 L2 캐시(5604)로부터의 데이터가 인출되어, SM(5514)의 기능 유닛들에서의 처리를 위해 L1 캐시들 각각에 저장된다. 적어도 하나의 실시예에서, L2 캐시(5604)는 메모리 인터페이스(5606) 및 XBar(5420)에 결합된다.

[0476] ROP 유닛(5602)은, 적어도 하나의 실시예에서, 컬러 압축, 픽셀 혼합 등과 같은, 픽셀 컬러에 관련된 그래픽 래스터 연산들을 수행한다. ROP 유닛(5602)은, 적어도 하나의 실시예에서, 래스터 엔진(5508)과 관련하여 깊이 테스트를 구현하며, 래스터 엔진(5508)의 컬러링 엔진으로부터의 픽셀 프래그먼트와 연관된 샘플 위치에 대한 깊이를 수신한다. 적어도 하나의 실시예에서, 깊이는 프래그먼트와 연관된 샘플 위치에 대한 깊이 버퍼에서의 대응하는 깊이와 대조하여 테스트된다. 적어도 하나의 실시예에서, 프래그먼트가 샘플 위치에 대한 깊이 테스트를 통과하면, 그 후, ROP 유닛(5602)은 깊이 버퍼를 업데이트하고 깊이 테스트의 결과를 래스터 엔진(5508)에 송신한다. 파티션 유닛(5600)의 수는 GPC의 수와 상이할 수 있고, 따라서, 각각의 ROP 유닛(5602)은, 적어도 하나의 실시예에서, 각각의 GPC에 결합될 수 있다는 것을 이해할 것이다. 적어도 하나의 실시예에서, ROP 유닛(5602)은 상이한 GPC들로부터 수신된 패킷들을 추적하고, ROP 유닛(5602)에 의해 생성된 결과가 XBar(5420)을 통해 라우팅되는지를 결정한다.

[0477] 도 57은 적어도 하나의 실시예에 따른, 스트리밍 멀티-프로세서("SM")(5700)를 예시한다. 적어도 하나의 실시예에서, SM(5700)은 도 55의 SM이다. 적어도 하나의 실시예에서, SM(5700)은, 제한없이, 명령어 캐시(5702); 하나 이상의 스케줄러 유닛(5704); 레지스터 파일(5708); 하나 이상의 처리 코어("코어")(5710); 하나 이상의 특수 기능 유닛("SFU")(5712); 하나 이상의 로드/저장 유닛("LSU")(5714); 인터커넥트 네트워크(5716); 공유된 메모리/레벨 1("L1") 캐시(5718); 및 이들의 임의의 적합한 조합을 포함한다. 적어도 하나의 실시예에서, 작업 분배 유닛은 병렬 처리 유닛들("PPU")의 일반 처리 클러스터들("GPC") 상에서의 실행을 위한 태스크들을 디스패치하고, 각각의 태스크는 GPC 내의 특정한 데이터 처리 클러스터("DPC")에 할당되고, 태스크가 셰이더 프로그램과 연관되는 경우, 태스크는 SM들(5700) 중 하나에 할당된다. 적어도 하나의 실시예에서, 스케줄러 유닛(5704)은 작업 분배 유닛으로부터 태스크들을 수신하고, SM(5700)에 할당된 하나 이상의 스레드 블록에 대한 명령어 스케줄링을 관리한다. 적어도 하나의 실시예에서, 스케줄러 유닛(5704)은 병렬 스레드들의 워프들로서 실행을 위한 스레드 블록들을 스케줄링하며, 각각의 스레드 블록은 적어도 하나의 워프를 할당받는다. 적어도 하나의 실시예에서, 각각의 워프는 스레드들을 실행한다. 적어도 하나의 실시예에서, 스케줄러 유닛(5704)은 복수의 상이한 스레드 블록을 관리하고, 상이한 스레드 블록들에 워프들을 할당한 다음, 각각의 클럭 사이클 동안 복수의 상이한 협력 그룹들로부터의 명령어들을 다양한 기능 유닛들(예를 들어, 처리 코어들(5710), SFU들(5712) 및 LSU들(5714))로 디스패치한다.

[0478] 적어도 하나의 실시예에서, 협력 그룹들이란, 개발자들이 스레드들이 통신하고 있는 입도를 표현하는 것을 허용하여 더 풍부하고 효율적인 병렬 분해들의 표현을 가능케하는 통신 스레드 그룹들을 조직화하기 위한 프로그래밍 모델을 지칭할 수 있다. 적어도 하나의 실시예에서, 협력 론칭 API들은 병렬 알고리즘들의 실행을 위한 스레드 블록들 사이의 동기화를 지원한다. 적어도 하나의 실시예에서, 종래의 프로그래밍 모델들의 애플리케이션들

은 협력 스레드들을 동기화하기 위한 단일의 단순한 구성: 스레드 블록의 모든 스레드들에 걸친 장벽(예를 들어, `syncthreads()` 함수)을 제공한다. 그러나, 적어도 하나의 실시예에서, 프로그래머들은, 스레드 블록 입도들보다 작은 스레드들의 그룹들을 정의하고 정의된 그룹들 내에서 동기화하여 집합적인 범그룹적 기능 인터페이스들의 형태로 더 큰 성능, 설계 유연성 및 소프트웨어 재사용을 가능케할 수 있다. 적어도 하나의 실시예에서, 협력 그룹들은 프로그래머들이 서브블록(즉, 단일 스레드만큼 작음) 및 다중 블록 입도들에서 명시적으로 스레드들의 그룹들을 정의하고 협력 그룹 내의 스레드들에 대한 동기화 같은 집합적 연산들을 수행할 수 있게 한다. 프로그래밍 모델은 소프트웨어 경계들을 가로지르는 클린 컴포지션(clean composition)을 지원하며, 따라서, 라이브러리 및 유틸리티 기능들은 수렴에 대한 가정들을 할 필요 없이 그들의 로컬 컨텍스트 내에서 안전하게 동기화할 수 있다. 적어도 하나의 실시예에서, 협력 그룹 프리미티브들은, 제한없이, 생산자-소비자 병렬성, 기회주의적 병렬성, 및 스레드 블록들의 전체 그리드에 걸친 글로벌 동기화를 포함한, 협력적 병렬성의 새로운 패턴들을 가능케한다.

[0479] 적어도 하나의 실시예에서, 디스패치 유닛(5706)은 하나 이상의 기능 유닛에 명령어들을 송신하도록 구성되고, 스케줄러 유닛(5704)은, 제한없이, 각각의 클록 사이클 동안 동일한 워프로부터의 2개의 상이한 명령어가 디스패치될 수 있게 하는 2개의 디스패치 유닛(5706)을 포함한다. 적어도 하나의 실시예에서, 각각의 스케줄러 유닛(5704)은 단일 디스패치 유닛(5706) 또는 추가적인 디스패치 유닛들(5706)을 포함한다.

[0480] 적어도 하나의 실시예에서, 각각의 SM(5700)은, 적어도 하나의 실시예에서, SM(5700)의 기능 유닛들에 대한 레지스터들의 세트를 제공하는 레지스터 파일(5708)을 포함하지만 이에 제한되지 않는다. 적어도 하나의 실시예에서, 레지스터 파일(5708)은 각각의 기능 유닛이 레지스터 파일(5708)의 전용 부분을 할당받도록 각각의 기능 유닛들 사이에서 분할된다. 적어도 하나의 실시예에서, 레지스터 파일(5708)은 SM(5700)에 의해 실행되는 상이한 워프들 사이에서 분할되고, 레지스터 파일(5708)은 기능 유닛들의 데이터 경로들에 접속된 피연산자들을 위한 임시 저장소를 제공한다. 적어도 하나의 실시예에서, 각각의 SM(5700)은, 제한없이, 복수의 L 처리 코어(5710)를 포함한다. 적어도 하나의 실시예에서, SM(5700)은, 제한없이, 많은 수(예를 들어, 128개 이상)의 별개의 처리 코어들(5710)을 포함한다. 적어도 하나의 실시예에서, 각각의 처리 코어(5710)는, 적어도 하나의 실시예에서, 제한없이, 부동 소수점 산술 로직 유닛 및 정수 산술 로직 유닛을 포함한 그러나 이것으로 제한되지 않는, 전체 파이프라인화된, 단정도, 배정도, 및/또는 혼합 정밀도 처리 유닛을 포함한다. 적어도 하나의 실시예에서, 부동 소수점 산술 로직 유닛은 부동 소수점 산술을 위한 IEEE 754-2008 표준을 구현한다. 적어도 하나의 실시예에서, 처리 코어(5710)는, 제한없이, 64개의 단정도(32비트) 부동 소수점 코어들, 64개의 정수 코어들, 32개의 배정도(64비트) 부동 소수점 코어들, 및 8개의 텐서 코어들을 포함한다.

[0481] 텐서 코어들은, 적어도 하나의 실시예에 따라 행렬 연산들을 수행하도록 구성된다. 적어도 하나의 실시예에서, 하나 이상의 텐서 코어가 처리 코어들(5710)에 포함된다. 적어도 하나의 실시예에서, 텐서 코어들은, 신경망 훈련 및 추론을 위한 컨볼루션 연산들 같은, 딥 러닝 행렬 산술을 수행하도록 구성된다. 적어도 하나의 실시예에서, 각각의 텐서 코어는 4x4 행렬 상에서 동작하고 행렬 곱셈 및 누산 연산 $D = A \times B + C$ 를 수행하며, 여기서 A, B, C, 및 D는 4x4 행렬이다.

[0482] 적어도 하나의 실시예에서, 행렬 곱셈 입력들 A 및 B는 16-비트 부동 소수점 행렬이고 누산 행렬들 C 및 D는 16-비트 부동 소수점 또는 32-비트 부동 소수점 행렬들이다. 적어도 하나의 실시예에서, 텐서 코어들은, 32-비트 부동 소수점 누산과 함께 16-비트 부동 소수점 입력 데이터에 관해 동작한다. 적어도 하나의 실시예에서, 16 비트 부동 소수점 곱은 64개의 연산을 이용하고, 전체 정밀도 곱을 생성하며, 이는 4x4x4 행렬 곱셈을 위한 다 른 중간 곱들과 함께 32비트 부동 소수점 덧셈을 이용하여 그 후 누산된다. 텐서 코어들은, 적어도 하나의 실시예에서, 이들 더 작은 엘리먼트들로부터 구축되는, 훨씬 더 큰 2차원 이상의 차원의 행렬 연산들을 수행하는데 이용된다. 적어도 하나의 실시예에서, CUDA 9 C++ API 같은 API는 전문화된 행렬 로딩, 행렬 곱셈 및 누산, 및 행렬 저장 연산들을 노출시켜 CUDA-C++ 프로그램으로부터 텐서 코어들을 효율적으로 이용한다. 적어도 하나의 실시예에서, CUDA 레벨에서, 워프-레벨 인터페이스는, 워프의 32개 스레드 모두에 걸쳐 있는 16x16 크기 행렬들을 가정한다.

[0483] 적어도 하나의 실시예에서, 각각의 SM(5700)은, 제한없이, 특수 기능들(예를 들어, 속성 평가, 역제곱근 등)을 수행하는 M개의 SFU(5712)를 포함한다. 적어도 하나의 실시예에서, SFU들(5712)은, 제한없이, 계층적 트리 데이터 구조를 순회하도록 구성된 트리 순회 유닛을 포함한다. 적어도 하나의 실시예에서, SFU들(5712)은, 제한없이, 텍스처 맵 필터링 연산들을 수행하도록 구성된 텍스처 유닛을 포함한다. 적어도 하나의 실시예에서, 텍스처 유닛들은 SM(5700)에 의해 실행되는 셰이더 프로그램들에서 이용하기 위한 샘플링된 텍스처 값들을 생성하기 위해 메모리 및 샘플 텍스처 맵들로부터 텍스처 맵들(예를 들어, 텍셀들의 2D 어레이)을 로딩하도록 구성

된다. 적어도 하나의 실시예에서, 텍스처 맵들은 공유된 메모리/L1 캐시(5718)에 저장된다. 적어도 하나의 실시예에서, 텍스처 유닛들은, 적어도 하나의 실시예에 따라, mip맵(mip-map)들(예를 들어, 다양한 레벨들의 상세의 텍스처 맵들)을 이용하는 필터링 연산들 같은 텍스처 연산들을 구현한다. 적어도 하나의 실시예에서, 각각의 SM(5700)은, 제한없이, 2개의 텍스처 유닛을 포함한다.

[0484] 적어도 하나의 실시예에서, 각각의 SM(5700)은 공유된 메모리/L1 캐시(5718)와 레지스터 파일(5708) 사이의 로드 및 저장 연산들을 구현하는 N개의 LSU(5714)를 포함하지만, 이에 제한되지 않는다. 적어도 하나의 실시예에서, 각각의 SM(5700)은 기능 유닛들 각각을 레지스터 파일(5708)에 그리고 LSU(5714)를 레지스터 파일(5708)에 그리고 공유된 메모리/L1 캐시(5718)에 접속시키는 인터커넥트 네트워크(5716)(이들로 제한되지 않음)를 포함한다. 적어도 하나의 실시예에서, 인터커넥트 네트워크(5716)는 기능 유닛들 중 임의의 것을 레지스터 파일(5708) 내의 레지스터들 중 임의의 것에 접속시키고 LSU들(5714)을 레지스터 파일(5708) 및 공유된 메모리/L1 캐시(5718) 내의 메모리 위치들에 접속시키도록 구성될 수 있는 크로스바이다.

[0485] 적어도 하나의 실시예에서, 공유된 메모리/L1 캐시(5718)는, 적어도 하나의 실시예에서, SM(5700)과 프리미티브 엔진 사이 및 SM(5700)의 스레드들 사이의 데이터 저장 및 통신을 허용하는 온-칩 메모리의 어레이이다. 적어도 하나의 실시예에서, 공유된 메모리/L1 캐시(5718)는, 제한없이, 128KB의 저장 용량을 포함하고, SM(5700)으로부터 파티션 유닛으로의 경로에 있다. 적어도 하나의 실시예에서, 공유된 메모리/L1 캐시(5718)는, 적어도 하나의 실시예에서, 판독들 및 기입들을 캐싱하는 데 사용된다. 적어도 하나의 실시예에서, 공유된 메모리/L1 캐시(5718), L2 캐시, 및 메모리 중 하나 이상은 백킹 저장들이다.

[0486] 데이터 캐시와 공유된 메모리 기능을 단일 메모리 블록으로 조합하는 것은, 적어도 하나의 실시예에서, 양자 모두의 타입의 메모리 액세스들에 대해 향상된 성능을 제공한다. 적어도 하나의 실시예에서, 예컨대, 공유된 메모리가 용량의 절반을 이용하도록 구성된 경우에, 용량은 공유된 메모리를 이용하지 않는 프로그램들에 의해 캐시로 이용되거나 캐시로서 이용 가능하며, 텍스처 및 로드/저장 동작들은 나머지 용량을 이용할 수 있다. 공유된 메모리/L1 캐시(5718) 내의 통합은, 적어도 하나의 실시예에 따라, 공유된 메모리/L1 캐시(5718)가 스트리밍 데이터를 위한 고처리량 도관으로서 기능하는 동시에, 빈번하게 재사용되는 데이터에 대한 고대역폭 및 저레이턴시 액세스를 제공하는 것을 가능케한다. 적어도 하나의 실시예에서, 범용 병렬 계산을 위해 구성될 때, 그래픽 처리에 비해 더 간단한 구성이 이용될 수 있다. 적어도 하나의 실시예에서, 고정 기능 그래픽 처리 유닛들은 바이패스되어, 훨씬 더 단순한 프로그래밍 모델을 생성한다. 범용 병렬 계산 구성에서, 작업 분배 유닛은, 적어도 하나의 실시예에서, 스레드들의 블록들을 DPC들에 직접 할당 및 분배한다. 적어도 하나의 실시예에서, 블록 내의 스레드들은, 각각의 스레드가 고유한 결과들을 생성하는 것을 보장하기 위해 산출에서 고유한 스레드 ID를 이용하여 동일한 프로그램을 실행하고, SM(5700)을 이용하여 프로그램을 실행하고 산출들을 수행하며, 공유된 메모리/L1 캐시(5718)를 이용하여 스레드들 사이에서 통신하고, LSU(5714)를 이용하여 공유된 메모리/L1 캐시(5718) 및 메모리 파티션 유닛을 통해 글로벌 메모리를 판독하고 기입한다. 적어도 하나의 실시예에서, 범용 병렬 계산을 위해 구성될 때, SM(5700)은 스케줄러 유닛(5704)이 DPC들 상에서 새로운 작업을 론칭하기 위해 사용할 수 있는 커맨드들을 기입한다.

[0487] 적어도 하나의 실시예에서, PPU는, 데스크탑 컴퓨터, 랩탑 컴퓨터, 태블릿 컴퓨터, 서버들, 슈퍼컴퓨터들, 스마트폰(예를 들어, 무선, 핸드헬드 디바이스), 개인 휴대 정보 단말기("PDA"), 디지털 카메라, 차량, 헤드 장착 디스플레이, 핸드헬드 전자 디바이스 등에 포함되거나 이에 결합된다. 적어도 하나의 실시예에서, PPU는 단일 반도체 기판 상에 구현된다. 적어도 하나의 실시예에서, PPU는, 추가적인 PPU들, 메모리, 축소된 명령어 세트 컴퓨터("RISC") CPU, 메모리 관리 유닛("MMU"), 디지털-아날로그 변환기("DAC") 등 같은 하나 이상의 다른 디바이스와 함께 시스템-온-칩("SoC")에 포함된다.

[0488] 적어도 하나의 실시예에서, PPU는 하나 이상의 메모리 디바이스를 포함하는 그래픽 카드에 포함될 수 있다. 그래픽 카드는 데스크탑 컴퓨터의 마더보드 상의 PCIe 슬롯과 인터페이스하도록 구성될 수 있다. 적어도 하나의 실시예에서, PPU는 마더보드의 칩셋에 포함된 통합된 그래픽 처리 유닛("iGPU")일 수 있다.

[0489] 추론 및/또는 훈련 로직(2915)은 하나 이상의 실시예와 연관된 추론 및/또는 훈련 연산들을 수행하는데 이용된다. 추론 및/또는 훈련 로직(2915)에 관한 상세사항이 도 29a 및/또는 도 29b와 관련하여 본 명세서에 제공되어 있다. 적어도 하나의 실시예에서, 딥 러닝 애플리케이션 프로세서는 SM(5700)에 제공되는 정보를 예측 또는 추론하기 위해 신경망과 같은 머신 러닝 모델을 훈련하는데 사용된다. 적어도 하나의 실시예에서, SM(5700)은 다른 프로세서 또는 시스템에 의해 또는 SM(5700)에 의해 훈련된, 훈련된 머신 러닝 모델(예를 들어, 신경망)에 기초하여 정보를 추론 또는 예측하기 위해 사용된다. 적어도 하나의 실시예에서, SM(5700)은 본 명세서에 설명

된 하나 이상의 신경망 이용 사례를 수행하는데 이용될 수 있다. 적어도 하나의 실시예에서, 이미지로부터 객체의 배향을 결정하는 신경망은 후술되는 바와 같이 반도체 플랫폼으로 구현된다.

[0490] 적어도 하나의 실시예에서, 단일 반도체 플랫폼이란, 유일한 단일 반도체 기반 집적 회로 또는 칩을 지칭할 수 있다. 적어도 하나의 실시예에서, 멀티-칩 모듈들은 온-칩 동작을 시뮬레이션하고, 종래의 중앙 처리 유닛("CPU") 및 버스 구현을 이용하는 것보다 상당한 개선을 이루는 증가된 집속성을 동반하여 이용될 수 있다. 적어도 하나의 실시예에서, 다양한 모듈은 또한, 사용자의 요구에 따라 별개로 또는 반도체 플랫폼들의 다양한 조합으로 위치할 수 있다.

[0491] 적어도 하나의 실시예에서, 머신 판독가능 실행가능 코드 또는 컴퓨터 제어 로직 알고리즘들의 형태의 컴퓨터 프로그램들은 메인 메모리(3504) 및/또는 보조 저장소에 저장된다. 컴퓨터 프로그램들은, 하나 이상의 프로세서들에 의해 실행되는 경우, 시스템(3500)이 적어도 하나의 실시예에 따라 다양한 기능들을 수행할 수 있게 한다. 메모리(3504), 저장소, 및/또는 임의의 다른 저장소는 컴퓨터 판독가능 매체의 가능한 예들이다. 적어도 하나의 실시예에서, 보조 저장소는, 플로피 디스크 드라이브, 자기 테이프 드라이브, 콤팩트 디스크 드라이브, DVD(digital versatile disk) 드라이브, 레코딩 디바이스, USB(universal serial bus) 플래시 메모리 등을 나타내는, 하드 디스크 드라이브 및/또는 착탈식 저장 드라이브와 같은 임의의 적합한 저장 디바이스 또는 시스템을 지칭할 수 있다. 적어도 하나의 실시예에서, 다양한 이전 도면들의 아키텍처 및/또는 기능은 CPU(3502); 병렬 처리 시스템(3512); 양자 모두의 CPU(3502)의 능력들의 적어도 일부가 가능한 집적 회로; 병렬 처리 시스템(3512); 칩셋(예를 들어, 관련 기능들을 수행하기 위한 유닛으로서 작동하고 판매하도록 설계된 집적 회로들의 그룹 등); 및 집적 회로(들)의 임의의 적합한 조합에 관련하여 구현된다.

[0492] 적어도 하나의 실시예에서, 다양한 이전 도면들의 아키텍처 및/또는 기능은, 일반 컴퓨터 시스템, 회로 기판 시스템, 엔터테인먼트 목적 전용의 게임 콘솔 시스템, 주문형 시스템 등에 관련하여 구현된다. 적어도 하나의 실시예에서, 컴퓨터 시스템(3500)은, 데스크탑 컴퓨터, 랩탑 컴퓨터, 태블릿 컴퓨터, 서버들, 슈퍼컴퓨터들, 스마트폰(예를 들어, 무선 핸드헬드 디바이스), 개인 휴대 정보 단말기("PDA"), 디지털 카메라, 차량, 헤드 장착 디스플레이, 핸드헬드 전자 디바이스, 이동 전화 디바이스, 텔레비전, 워크스테이션, 게임 콘솔, 임베디드 시스템, 및/또는 임의의 다른 타입의 로직의 형태를 취할 수 있다.

[0493] 적어도 하나의 실시예에서, 병렬 처리 시스템(3512)은, 제한없이, 복수의 병렬 처리 유닛("PPU")(3514) 및 연관된 메모리들(3516)을 포함한다. 적어도 하나의 실시예에서, PPU들(3514)은 인터커넥트(3518) 및 스위치(3520) 또는 멀티플렉서를 통해 호스트 프로세서 또는 다른 주변기기 디바이스들에 접속된다. 적어도 하나의 실시예에서, 병렬 처리 시스템(3512)은 병렬성될 수 있는 PPU들(3514)에 걸쳐 계산 태스크들을 분배한다 - 예를 들어, 다수의 그래픽 처리 유닛("GPU") 스레드 블록들에 걸친 계산 태스크들의 분배의 일부로서. 적어도 하나의 실시예에서, 메모리는 PPU들(3514)의 일부 또는 전부에 걸쳐(예를 들어, 판독 및/또는 기입 액세스에 대해) 공유되고 액세스가능하지만, 이러한 공유된 메모리는 PPU(3514)에 상주하는 레지스터들 및 로컬 메모리의 이용과 관련하여 성능 페널티를 초래할 수 있다. 적어도 하나의 실시예에서, PPU들(3514)의 동작은 `__syncthreads()`와 같은 커맨드의 사용을 통해 동기화되고, 여기서 블록 내의 모든 스레드들(예를 들어, 다수의 PPU들(3514)에 걸쳐 실행됨)은 진행 이전에 코드의 특정 실행 포인트에 도달한다.

[0494] 다른 변형들은 본 개시내용의 사상 내에 있다. 따라서, 개시된 기술들은 다양한 수정 및 대안적인 구성이 가능하지만, 그 특정 예시된 실시예들이 도면들에 도시되고 상세히 진술되었다. 그러나, 개시된 특정 형태 또는 형태들로 본 개시내용을 제한하려는 의도는 없지만, 첨부된 청구항들에 정의된 바와 같이 본 개시내용의 사상 및 범위 내에 속하는 모든 수정, 대안적 구성 및 균등물을 포함하고자 하는 의도임을 이해해야 한다.

[0495] 본 개시내용의 적어도 하나의 실시예는 다음의 조항들을 고려하여 설명될 수 있다:

[0496] 1. 프로세서로서, 2개 이상의 이미지 내의 제1 객체의 배향의 차이를 최소화하도록 훈련된 하나 이상의 신경망에 적어도 부분적으로 기초하여 하나 이상의 이미지를 수신하는 것에 응답하여 하나 이상의 이미지 내의 하나 이상의 객체의 배향을 결정하는 하나 이상의 회로를 포함한다.

[0497] 2. 조항 1의 프로세서에 있어서,

[0498] 2개 이상의 이미지는 하나 이상의 이미지를 포함하고;

[0499] 2개 이상의 이미지는 하나 이상의 신경망에 의해 결정된 배향에서의 하나 이상의 객체의 수학적 모델로부터 생성된 하나 이상의 객체의 이미지를 포함하고;

- [0500] 하나 이상의 신경망은 2개 이상의 이미지 사이의 차이의 측정치를 생성하는 추가적인 신경망에 의해 생성된 값에 적어도 부분적으로 기초하는 보상을 사용하여 훈련된다.
- [0501] 3. 조항 2의 프로세서에 있어서, 수학적 모델로부터 생성된 하나 이상의 객체의 이미지는 미분불가능 렌더러에 의해 생성된다.
- [0502] 4. 조항 1 내지 3 중 어느 하나의 프로세서에 있어서, 하나 이상의 객체는 텍스처 없는 대칭적인 객체를 포함한다.
- [0503] 5. 조항 2 내지 4 중 어느 하나의 프로세서에 있어서, 2개 이상의 이미지 사이의 차이의 측정치를 생성하는 추가적인 신경망은 이미지 및 이미지의 교란된 버전을 추가적인 신경망에 제공함으로써 훈련된다.
- [0504] 6. 조항 2 내지 5 중 어느 하나의 프로세서에 있어서, 하나 이상의 신경망은 추가적인 신경망을 최대도 혼동시키도록 조정된다.
- [0505] 7. 조항 2 내지 6 중 어느 하나의 프로세서에 있어서, 수학적 모델로부터 생성된 하나 이상의 객체의 이미지는 하나 이상의 데이터 증강에 따라 생성된다.
- [0506] 8. 조항 7의 프로세서에 있어서, 하나 이상의 데이터 증강은 이미지 내의 하나 이상의 객체에 대한 광 위치, 조명 강도, 또는 정반사를 포함한다.
- [0507] 9. 프로세서에 있어서, 제2 배향을 갖는 객체의 제1 이미지에 적어도 부분적으로 기초하여 제1 배향을 갖는 객체의 제2 이미지를 생성하는 것을 돕고 제2 배향과 동등하게 관찰될 때까지 제1 배향을 조정하는 하나 이상의 회로를 포함한다.
- [0508] 10. 조항 9의 프로세서에 있어서,
- [0509] 제1 배향은 제1 이미지를 사용하여 제1 신경망에 의해 결정되고;
- [0510] 제1 배향은 제1 이미지와 제2 이미지 사이의 차이에 적어도 부분적으로 기초하여 조정되고;
- [0511] 제2 이미지는 제1 배향에 있는 객체의 수학적 모델을 사용하여 생성된다.
- [0512] 11. 조항 10의 프로세서에 있어서, 제2 이미지는 미분불가능 렌더러를 사용하여 생성된다.
- [0513] 12. 조항 10 또는 11의 프로세서에 있어서, 제1 이미지와 제2 이미지 사이의 차이는 제2 신경망을 사용하여 결정된다.
- [0514] 13. 조항 12의 프로세서에 있어서, 제2 신경망은 제3 이미지 및 제3 이미지의 교란된 버전을 제2 신경망에 제공함으로써 훈련된다.
- [0515] 14. 조항 10 내지 12 중 어느 하나의 프로세서에 있어서, 제2 이미지는 객체의 광 위치, 조명 강도 또는 정반사를 포함하는 하나 이상의 데이터 증강에 따라 생성된다.
- [0516] 15. 시스템으로서,
- [0517] 제2 이미지 및 제3 이미지 내의 객체의 배향의 차이에 적어도 부분적으로 기초하여 훈련된 제1 신경망을 이용하여 제1 이미지를 수신하는 것에 응답하여 제1 이미지 내의 객체의 배향을 결정하는 하나 이상의 프로세서; 및
- [0518] 제1 신경망을 저장하기 위한 하나 이상의 메모리를 포함한다.
- [0519] 16. 조항 15의 시스템에 있어서,
- [0520] 제2 이미지는 제1 신경망에 의해 결정된 배향에서 객체의 모델로부터 생성되고;
- [0521] 제1 신경망은 차이를 추정하는 추가적인 신경망에 의해 생성된 값에 적어도 부분적으로 기초하여 훈련된다.
- [0522] 17. 조항 16의 시스템에 있어서, 추가적인 신경망은, 추가적인 신경망에, 제1 배향의 객체의 이미지 및 제1 배향을 섭동함으로써 결정된 제2 배향의 객체의 이미지를 제공함으로써 훈련된다.
- [0523] 18. 조항 16 또는 17의 시스템에 있어서, 하나 이상의 신경망은 추가적인 신경망을 최대도 혼동시키도록 조정된다.
- [0524] 19. 조항 16 내지 18 중 어느 하나의 시스템에 있어서, 모델로부터 생성된 객체의 제1 이미지는 하나 이상의 테

이터 증강에 따라 생성된다.

- [0525] 20. 조항 19의 시스템에 있어서, 하나 이상의 데이터 증강은 제1 이미지에 대한 광학적 변경들을 포함한다.
- [0526] 21. 자율 차량에 있어서,
- [0527] 객체의 이미지를 획득하는 자율 차량에 부착된 카메라;
- [0528] 2개 이상의 이미지 내의 제1 객체의 배향의 차이를 최소화하도록 훈련된 하나 이상의 신경망에 적어도 부분적으로 기초하여 이미지를 수신하는 것에 응답하여 이미지 내의 객체의 배향을 결정하는 하나 이상의 회로를 포함하는 프로세서; 및
- [0529] 객체의 결정된 배향에 적어도 부분적으로 기초하여 자율 차량을 제어하는 내비게이션 시스템을 포함한다.
- [0530] 22. 조항 21의 자율 차량에 있어서,
- [0531] 2개 이상의 이미지는 하나 이상의 신경망에 의해 결정된 배향에서 객체의 모델로부터 생성된 객체의 이미지를 포함하고;
- [0532] 하나 이상의 신경망은 2개 이상의 이미지 사이의 차이의 측정치를 생성하는 추가적인 신경망에 의해 생성된 값에 적어도 부분적으로 기초하여 훈련된다.
- [0533] 23. 조항 22의 프로세서에 있어서, 객체의 이미지는 하나 이상의 데이터 증강에 따라 생성된다.
- [0534] 24. 조항 23의 프로세서에 있어서, 하나 이상의 데이터 증강은 광 위치, 조명 강도, 또는 정반사 파라미터를 포함한다.
- [0535] 개시된 실시예들을 설명하는 정황에서(특히 이하의 청구항들에 관련하여) 용어 "a" 및 "an" 및 "the"의 이용 및 유사한 지시물들은, 용어의 정의로서가 아니라, 본 명세서에 달리 나타내거나 문맥상 명확히 상충되지 않는 한, 단수와 복수 양자 모두를 포괄하는 것으로 해석되어야 한다. 용어들 "~을 포함하는", "~을 갖는", "~을 내포하는", "~을 함유하는"은, 달리 언급되지 않는 한 ("~을 포함한 그러나 이것으로 제한되지 않는"을 의미하는) 개방형 용어들로서 해석되어야 한다. 용어 "접속된"은, 수정되지 않고 물리적 접속들을 지칭할 때, 중간에 무언가가 있더라도 부분적으로 또는 전체적으로 ~ 내에 포함되거나, ~에 부착되거나, ~함께 결합된 것으로서 해석되어야 한다. 본 명세서에서 값들의 범위를 열거한 것은, 본 명세서에서 달리 표시되지 않는 한, 그 범위 내에 속하는 각각의 개별 값을 개별적으로 언급하는 약식 방법의 역할을 의도할 뿐이며, 각각의 개별 값은 본 명세서에서 개별적으로 열거된 것처럼 본 명세서에 통합된다. 용어 "세트"(예를 들어, "한 세트의 항목들") 또는 "서브세트"의 사용은, 문맥상 달리 언급되거나 상충되지 않는 한, 하나 이상의 멤버를 포함하는 비어 있지 않은 집합으로서 해석되어야 한다. 또한, 문맥상 달리 언급되거나 상충되지 않는 한, 용어, 대응하는 세트의 "서브세트"란, 반드시 대응하는 세트의 적합한 서브세트를 의미하는 것은 아니지만, 서브세트 및 대응하는 세트는 동일할 수 있다.
- [0536] "A, B, 및 C 중 적어도 하나" 또는 "A, B 및 C 중 적어도 하나" 형태의 구문들 같은 연결성 언어들은, 구체적으로 달리 언급되지 않거나 문맥상 명백하게 상충되지 않는 한, 문맥과 함께, 항목, 조건 등이 A 또는 B 또는 C이거나, A와 B와 C로 이루어진 세트의 비어 있지 않은 임의의 서브세트일 수 있다는 것을 나타내기 위해 일반적으로 사용되는 것으로 이해되어야 한다. 예를 들어, 3개의 멤버를 갖는 세트의 예시적인 예에서, 연결성 구문들 "A, B, 및 C 중 적어도 하나"와 "A, B 및 C 중 적어도 하나"는 다음과 같은 세트들 중 임의의 것을 의미한다: {A}, {B}, {C}, {A, B}, {A, C}, {B, C}, {A, B, C}. 따라서, 이러한 연결성 언어는, 일반적으로, 특정 실시예들이, A 중 적어도 하나, B 중 적어도 하나, 및 C 중 적어도 하나가 각각 존재할 것을 요구한다는 것을 암시하도록 의도한 것은 아니다. 또한, 문맥상 달리 언급되거나 상충되지 않는 한, 용어 "복수"는 복수인 상태를 나타낸다(예를 들어, "복수의 항목"은 다수의 항목을 나타냄). 복수에서 항목들의 수는 적어도 2개이지만, 그러나, 명시적으로 또는 문맥에 의해 그렇게 표시될 때 더 많을 수 있다. 또한, 달리 언급되거나 달리 문맥상 명백하지 않는 한, "~에 기초한"이라는 구문은 "~에 전적으로 기초한"이 아니라 "~에 적어도 부분적으로 기초한"을 의미한다.
- [0537] 본 명세서에 설명된 프로세스들의 동작들은, 본 명세서에 달리 나타내지 않거나 달리 문맥상 명백히 상충되지 않는 한 임의의 적합한 순서로 수행될 수 있다. 적어도 하나의 실시예에서, 본 명세서에 설명된 프로세스들(또는 이들의 변형들 및/또는 조합들) 같은 프로세스는, 실행가능한 명령어들로 구성된 하나 이상의 컴퓨터 시스템의 제어하에 수행되고, 하나 이상의 프로세서 상에서 집합적으로 실행되는 코드(예를 들어, 실행가능한 명령어

들, 하나 이상의 컴퓨터 프로그램 또는 하나 이상의 애플리케이션)로서, 하드웨어에 의해 또는 이들의 조합으로 구현된다. 적어도 하나의 실시예에서, 코드는, 예를 들어, 하나 이상의 프로세서에 의해 실행가능한 복수의 명령어를 포함하는 컴퓨터 프로그램의 형태로, 컴퓨터 판독가능한 저장 매체 상에 저장된다. 적어도 하나의 실시예에서, 컴퓨터 판독가능한 저장 매체는, 일시적 신호들(예를 들어, 전파하는 과도적인 전기 또는 전자기 송신)을 배제하지만 일시적 신호들의 트랜시버들 내의 비일시적 데이터 저장 회로(예를 들어, 버퍼, 캐시 및 큐)를 포함하는 비일시적 컴퓨터 판독가능한 저장 매체이다. 적어도 하나의 실시예에서, 코드(예를 들어, 실행가능한 코드 또는 소스 코드)는, 컴퓨터 시스템의 하나 이상의 프로세서에 의해 실행될 때(즉, 실행의 결과로서), 컴퓨터 시스템으로 하여금 여기서 설명된 동작들을 수행하게 하는 실행가능한 명령어들이 저장된 한 세트의 하나 이상의 비일시적 컴퓨터 판독가능한 저장 매체(또는 실행가능한 명령어들을 저장하는 다른 메모리) 상에 저장된다. 적어도 하나의 실시예에서, 비일시적 컴퓨터 판독가능 저장 매체들의 세트는 다수의 비일시적 컴퓨터 판독가능 저장 매체들 및 다수의 비일시적 컴퓨터 판독가능 저장 매체들의 개별 비일시적 저장 매체들 중 하나 이상은 모든 코드가 없는 반면 다수의 비일시적 컴퓨터 판독가능 저장 매체들은 집합적으로 모든 코드를 저장한다. 적어도 하나의 실시예에서, 상이한 명령어들이 상이한 프로세서들에 의해 실행되도록 실행가능한 명령어들이 실행된다-예를 들어, 비일시적 컴퓨터 판독가능 저장 매체 저장 명령어들 및 메인 중앙 처리 유닛("CPU")은 명령어들 중 일부를 실행하는 반면 그래픽 처리 유닛("GPU")은 다른 명령어들을 실행한다. 적어도 하나의 실시예에서, 컴퓨터 시스템의 상이한 컴포넌트들은 별개의 프로세서들을 갖고 상이한 프로세서들은 명령어들의 상이한 서브세트들을 실행한다.

[0538] 따라서, 적어도 하나의 실시예에서, 컴퓨터 시스템들은 본 명세서에 설명된 프로세스들의 동작들을 단독으로 또는 집합적으로 수행하는 하나 이상의 서비스를 구현하도록 구성되고 이러한 컴퓨터 시스템들은 동작의 수행을 가능케하는 적용가능한 하드웨어 및/또는 소프트웨어로 구성된다. 또한, 본 개시내용의 적어도 하나의 실시예를 구현하는 컴퓨터 시스템은 단일 디바이스고, 또 다른 실시예에서는, 상이하게 동작하는 다수의 디바이스를 포함하는 분산 컴퓨터 시스템으로서, 분산 컴퓨터 시스템이 여기서 설명된 동작들을 수행하지만 단일 디바이스가 동작들 모두를 수행하지는 않는다.

[0539] 본 명세서에 제공된 임의의 그리고 모든 예들, 또는 예시적인 언어(예를 들어, "~와 같은")의 사용은 단지 본 개시내용의 실시예들을 더 잘 예시하도록 의도되며, 달리 청구되지 않는 한, 본 개시내용의 범위에 대한 제한을 제기하지 않는다. 명세서의 어떠한 용어도 본 개시내용의 실시예 필수적인 임의의 청구되지 않은 엘리먼트를 가리키는 것으로 해석되어서는 안 된다.

[0540] 본 명세서에서 인용된 간행물들, 특허 출원들, 및 특허들을 포함한 모든 참고 문헌은, 마치 각각의 참고 문헌이 개별적으로 그리고 구체적으로 참조에 의해 포함되고 그 전체내용이 여기에 기재된 것과 동일한 정도로 참조에 의해 본 명세서에 포함된다.

[0541] 설명 및 청구항들에서, "결합된" 및 "접속된"이라는 용어들은 그들의 파생어들과 함께 사용될 수 있다. 이들 용어들은 서로 동의어로서 의도한 것은 아님을 이해해야 한다. 오히려, 특정한 예들에서, "접속된" 또는 "결합된"은 2개 이상의 엘리먼트가 직접 또는 간접으로 물리적 또는 전기적으로 서로 접촉한다는 것을 나타내는데 사용될 수 있다. "결합된"은 또한, 2개 이상의 엘리먼트가 서로 직접 접촉하지 않고, 오히려 서로 협력하거나 상호작용한다는 것을 의미할 수도 있다.

[0542] 구체적으로 달리 언급되지 않는 한, 명세서 전체를 통해, "처리", "계산", "산출", "결정" 등과 같은 용어들은, 계산 시스템의 레지스터들 및/또는 메모리들 내의 전자적 같은 물리적 양으로서 표현된 데이터를, 계산 시스템의 메모리들, 레지스터들, 또는 다른 이러한 정보 저장, 전달 또는 디스플레이 디바이스들 내의 물리적 양으로서 유사하게 표현된 다른 데이터로 조작 및/또는 변환하는, 컴퓨터 또는 계산 시스템, 또는 유사한 전자 계산 디바이스의 액션 및/또는 프로세스들을 지칭한다는 것을 이해해야 한다.

[0543] 유사한 방식으로, 용어 "프로세서"란, 레지스터들 및/또는 메모리로부터의 전자 데이터를 처리하여 그 전자 데이터를 레지스터들 및/또는 메모리에 저장될 수 있는 다른 전자 데이터로 변환하는 임의의 디바이스 또는 디바이스의 일부를 지칭한다. 비제한적인 예로서, "프로세서"는 CPU 또는 GPU일 수 있다. "계산 플랫폼"은 하나 이상의 프로세서를 포함할 수 있다. 본 명세서에서 사용될 때, "소프트웨어" 프로세스는, 예를 들어, 태스크들, 스레드들, 및 지능형 에이전트들 같은, 시간이 지남에 따라 작업을 수행하는 소프트웨어 및/또는 하드웨어 엔티티들을 포함할 수 있다. 또한, 각각의 프로세스는, 명령어들을 순차적으로 또는 병렬로, 연속적으로 또는 간헐적으로 실행하기 위해 다수의 프로세스를 참조할 수 있다. 용어들 "시스템" 및 "방법"은, 시스템이 하나 이상의 방법을 구현할 수 있고 방법들이 시스템으로 고려될 수 있는 한 여기서는 서로 교환 가능하게 사용된다.

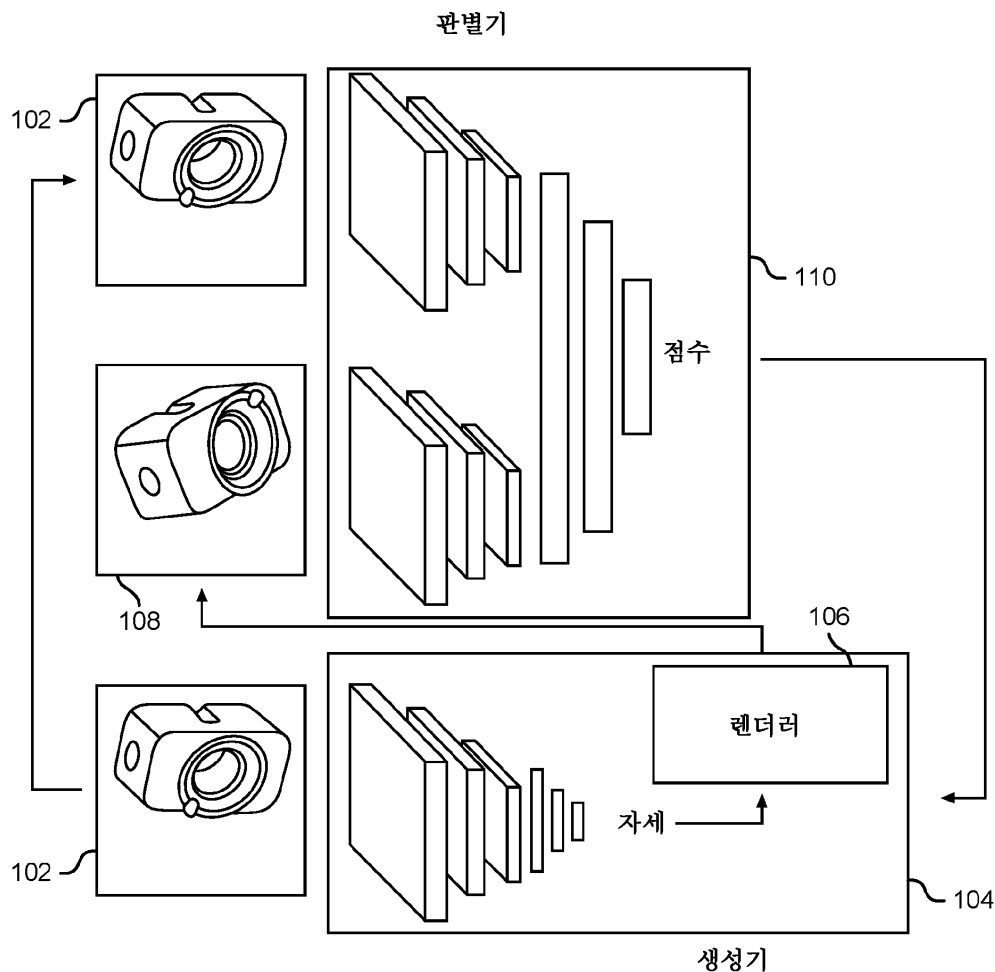
[0544] 본 문서에서, 아날로그 또는 디지털 데이터를 획득하거나, 취득하거나, 수신하거나, 또는 서비스시스템, 컴퓨터 시스템 또는 컴퓨터-구현된 머신에 입력하는 것에 대한 참조가 이루어질 수 있다. 아날로그 및 디지털 데이터의 획득, 취득, 수신 또는 입력하는 프로세스는, 함수 호출 또는 애플리케이션 프로그래밍 인터페이스에 대한 호출의 파라미터로서 데이터를 수신하는 것 같은 다양한 방식으로 달성될 수 있다. 일부 구현에서, 아날로그 또는 디지털 데이터를 획득, 취득, 수신 또는 입력하는 프로세스는, 직렬 또는 병렬 인터페이스를 통해 데이터를 전송함으로써 달성될 수 있다. 또 다른 구현에서, 아날로그 또는 디지털 데이터를 획득, 취득, 수신 또는 입력하는 프로세스는, 컴퓨터 네트워크를 통해 데이터를 제공측 엔티티로부터 취득측 엔티티로 전송함으로써 달성될 수 있다. 또한, 아날로그 또는 디지털 데이터를 제공, 출력, 전송, 송신, 또는 프리젠틱하는 것에 대한 참조가 이루어질 수 있다. 다양한 예에서, 아날로그 또는 디지털 데이터를 제공, 출력, 전송, 송신, 또는 프리젠틱하는 프로세스는, 함수 호출의 입력 또는 출력 파라미터로서, 애플리케이션 프로그래밍 인터페이스 또는 프로세스간 통신 메커니즘의 파라미터로서, 데이터를 전송함으로써 달성될 수 있다.

[0545] 이상의 논의가 설명된 기술들의 예시적인 구현들을 설명하고 있지만, 설명된 기능을 구현하기 위해 다른 아키텍처들이 사용될 수 있고 본 개시내용의 범위 내에 있는 것으로 의도된다. 또한, 논의의 목적으로 상기 설명에서는 구체적인 책임들의 분배들을 정의했지만, 상황에 따라 다양한 기능과 책임들이 상이한 방식으로 분배 및 분할될 수 있다.

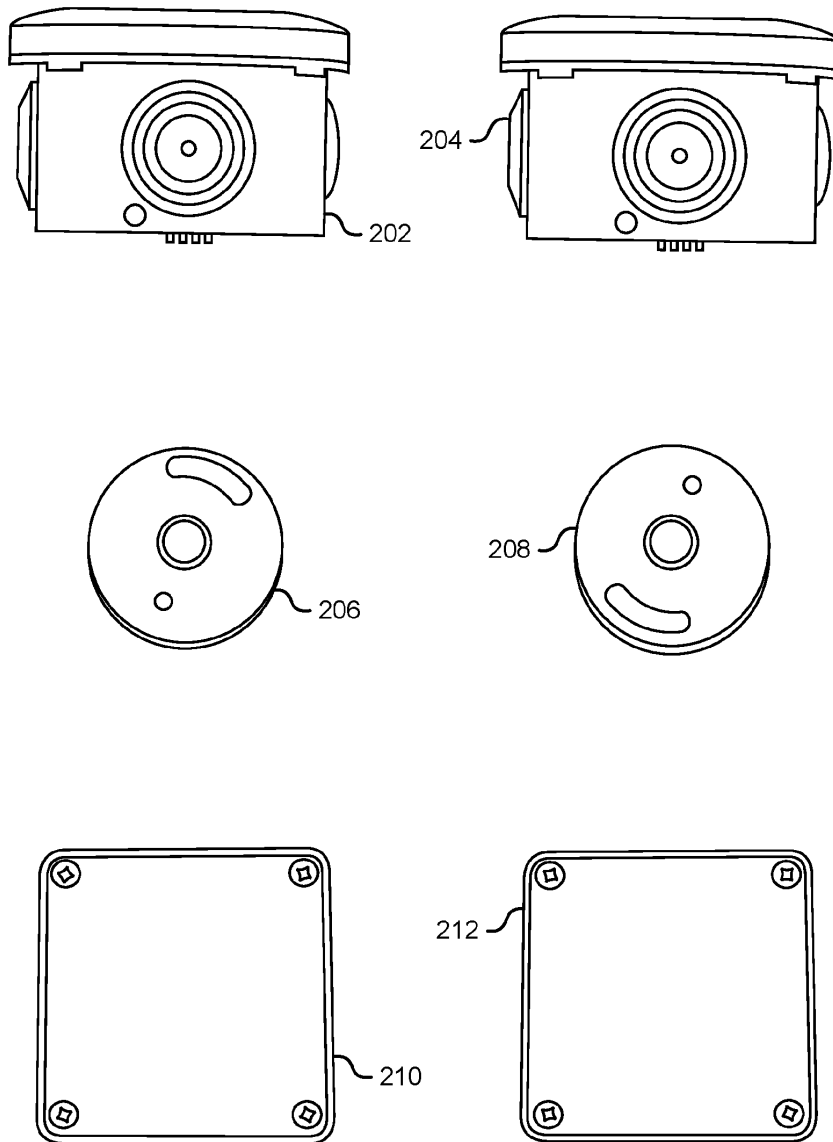
[0546] 또한, 구조적 피쳐들 및/또는 방법론적 작용들 특유의 용어로 주제가 설명되었지만, 첨부된 청구항들에서 청구되는 주제는 반드시 설명된 특정한 피쳐나 작용들로 제한되는 것은 아님을 이해해야 한다. 오히려, 개시된 특정한 피쳐들 및 작용들은 청구항들을 구현하는 예시적인 형태들로서 개시된 것이다.

도면

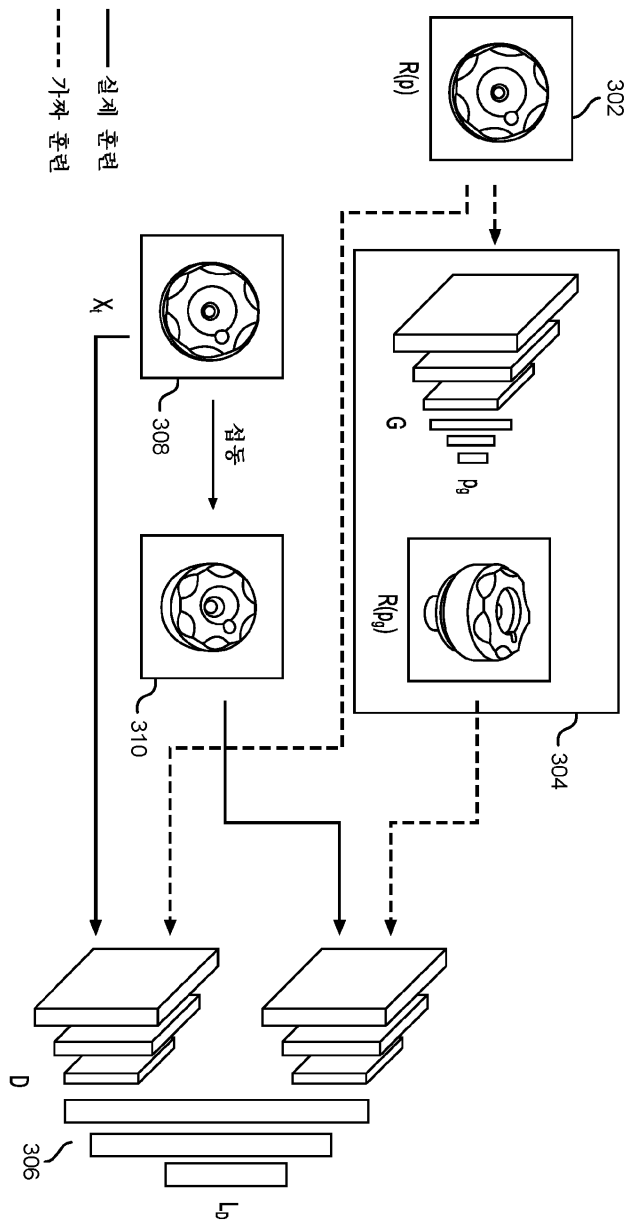
도면1



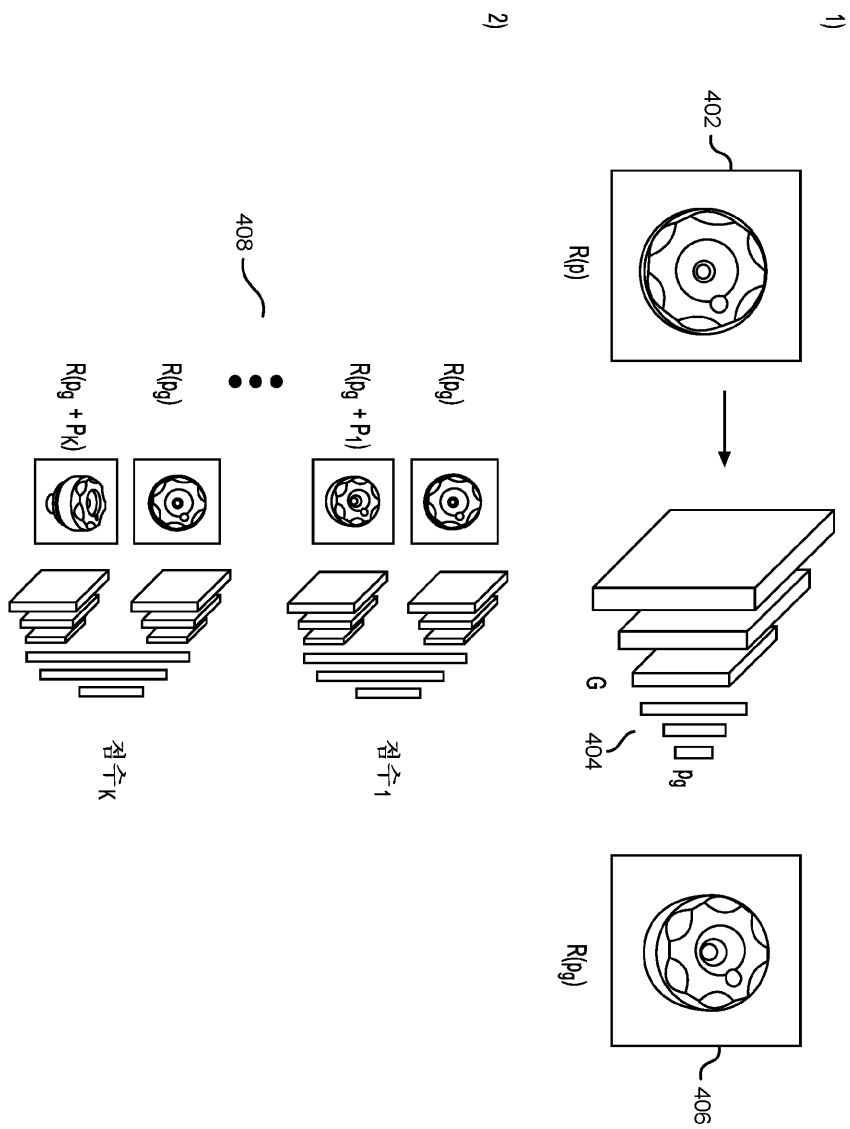
도면2



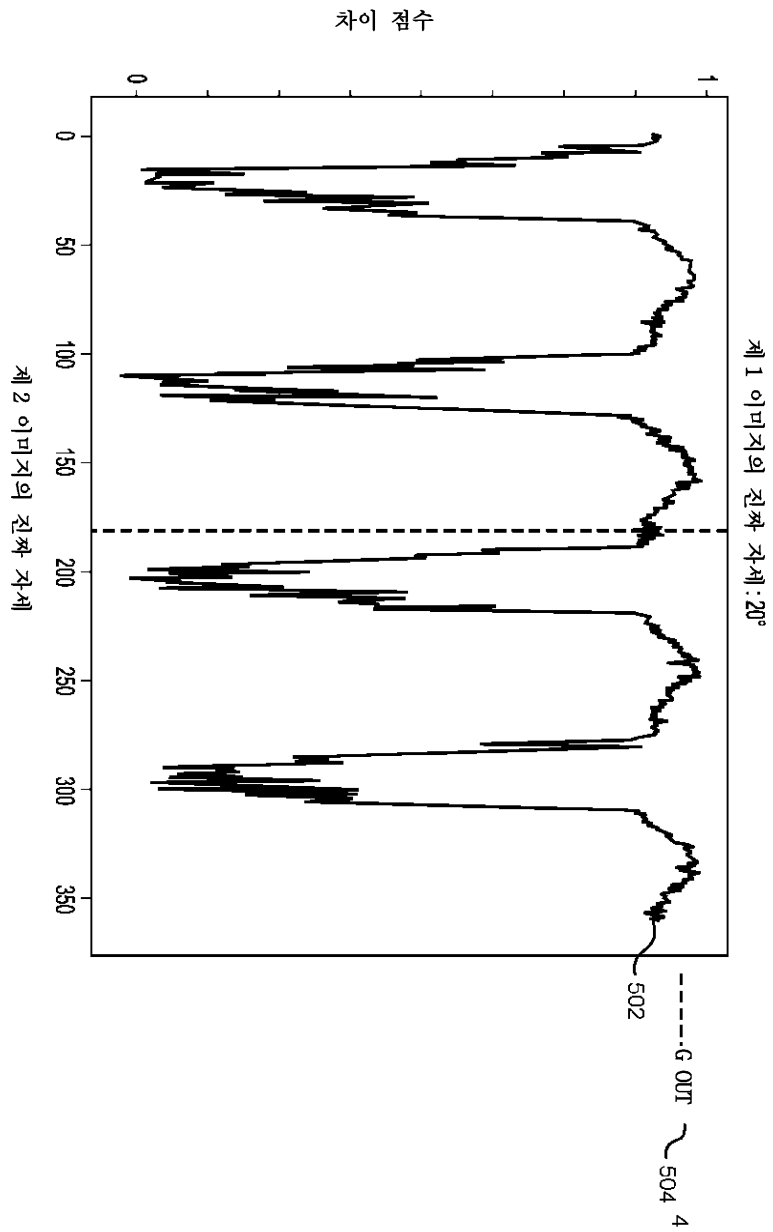
도면3



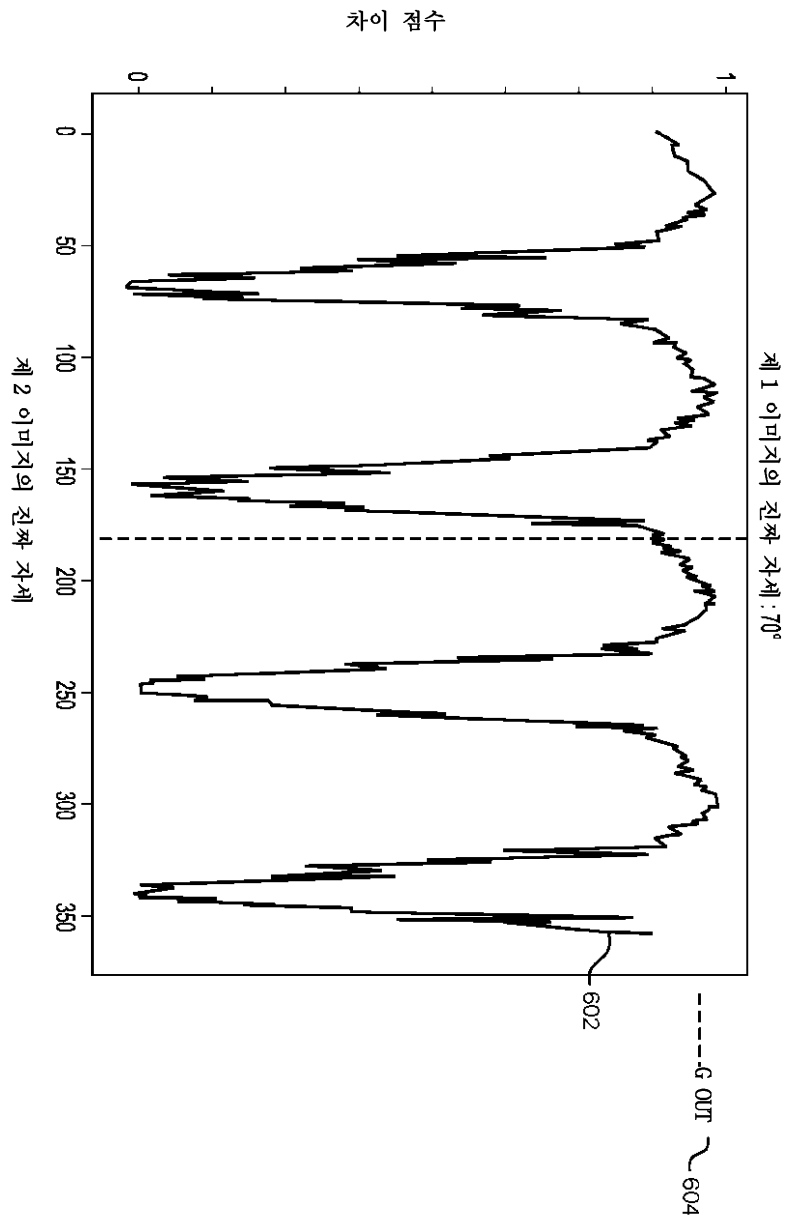
도면4



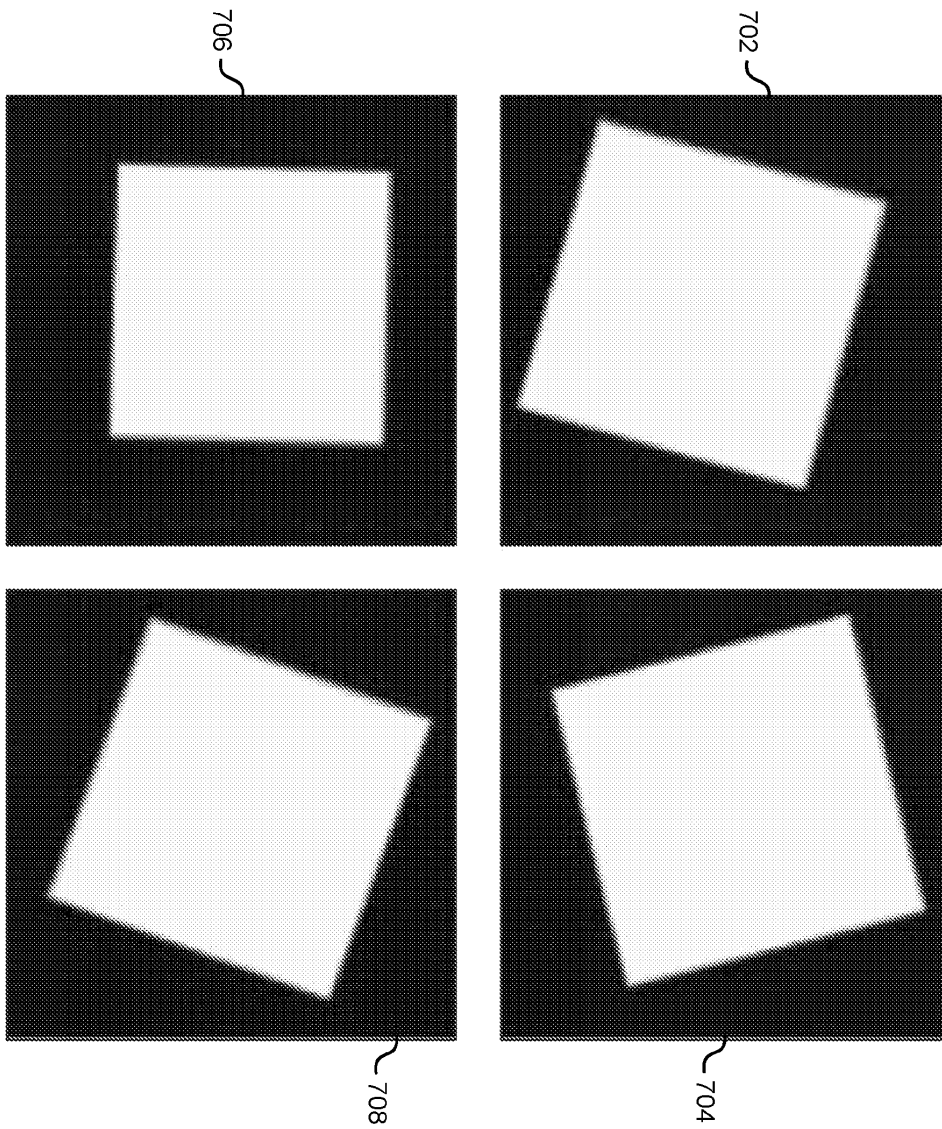
도면5



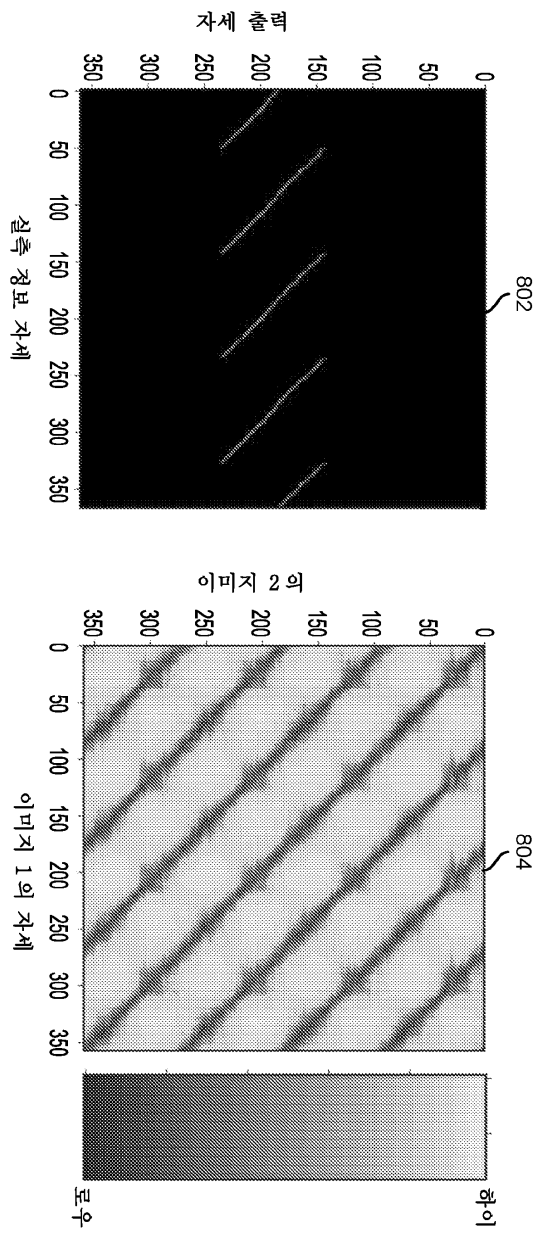
도면6



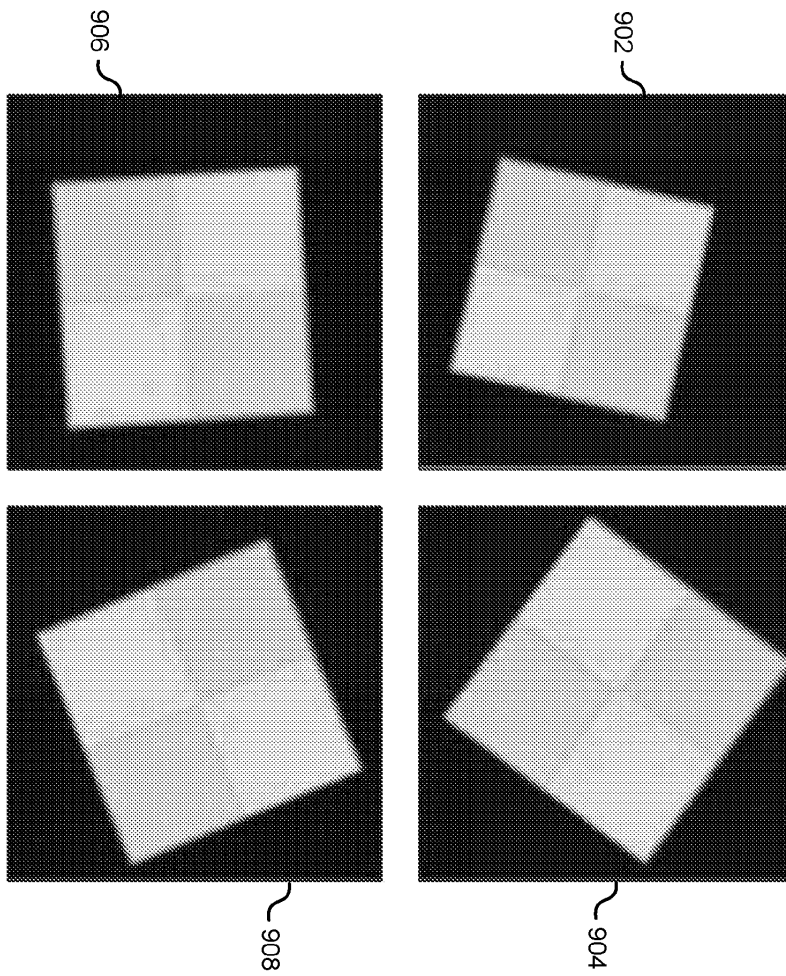
도면7



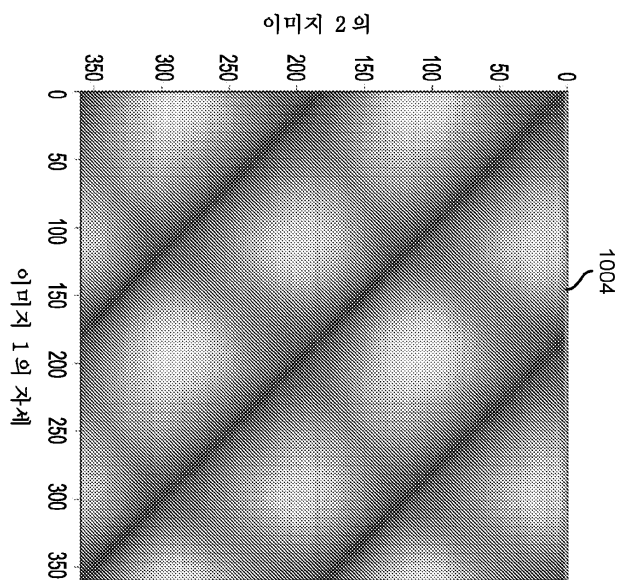
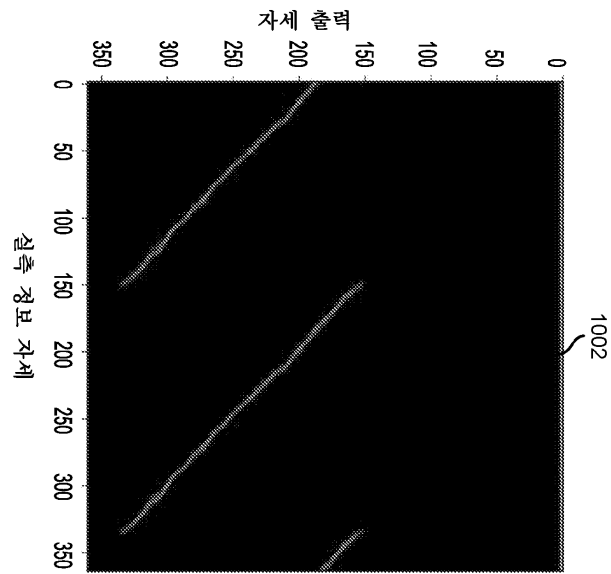
도면8



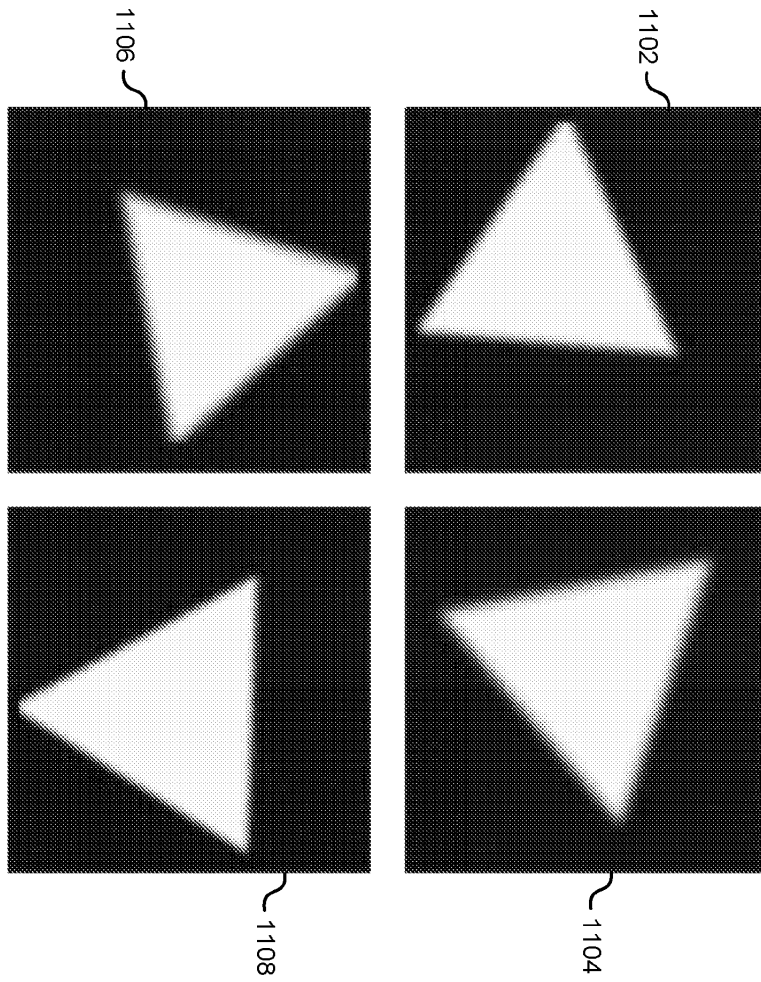
도면9



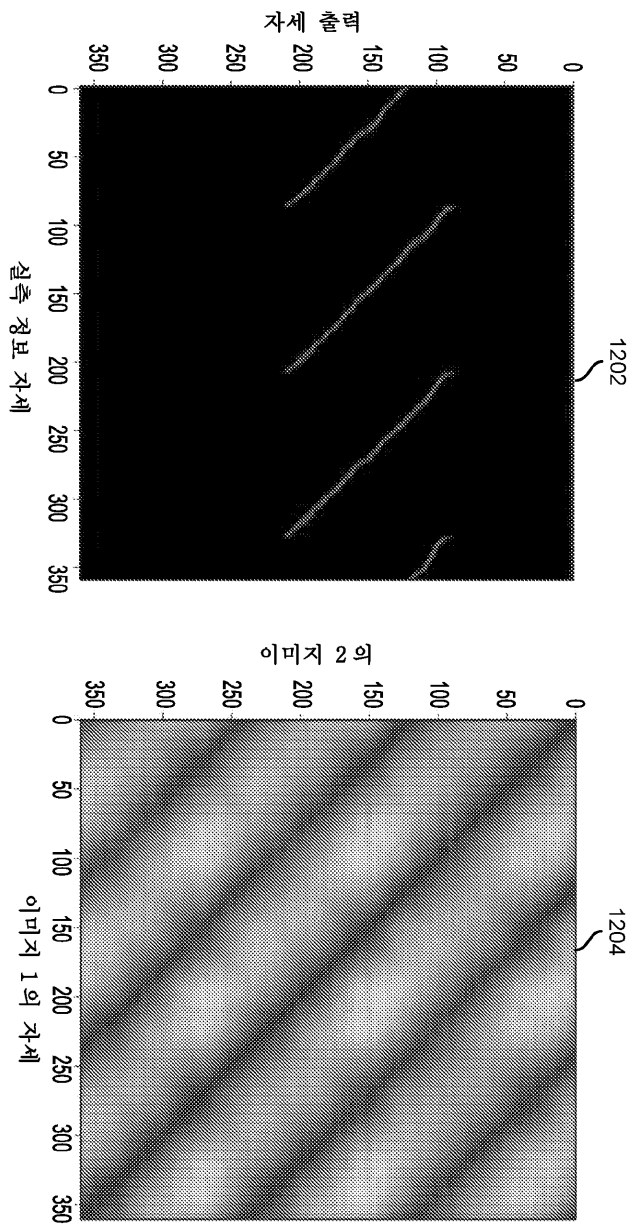
도면10



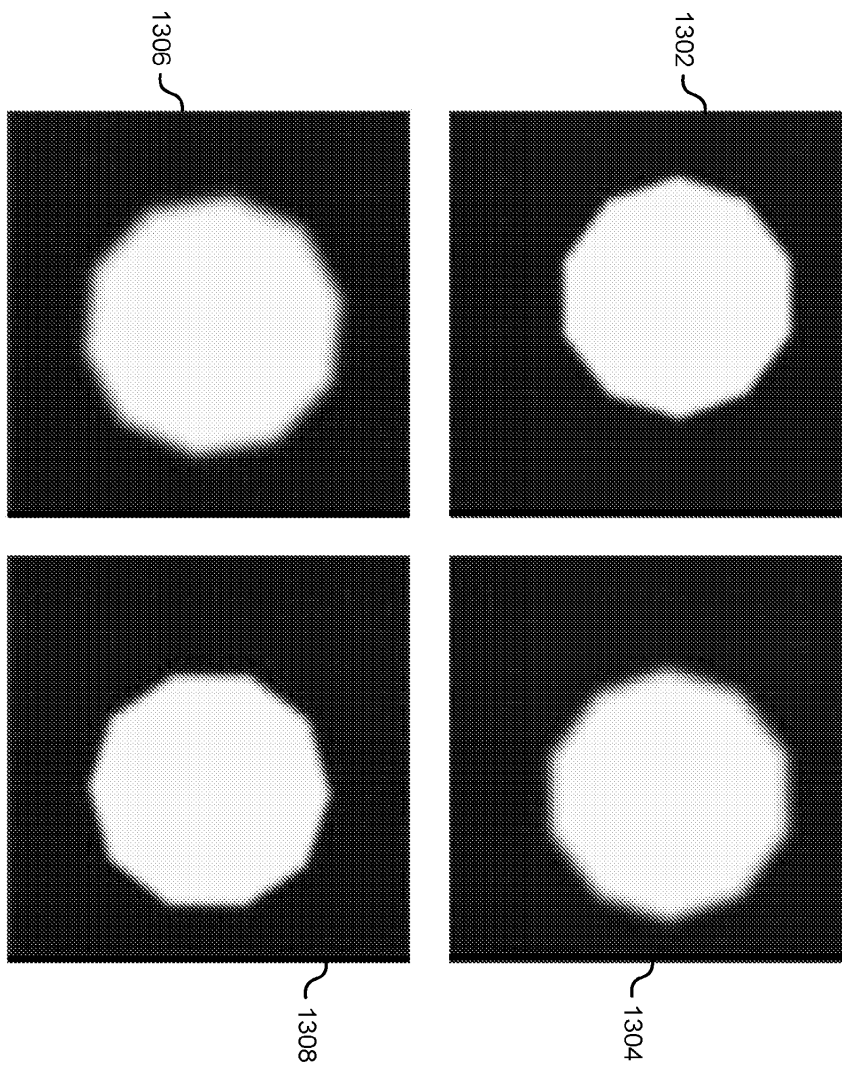
도면11



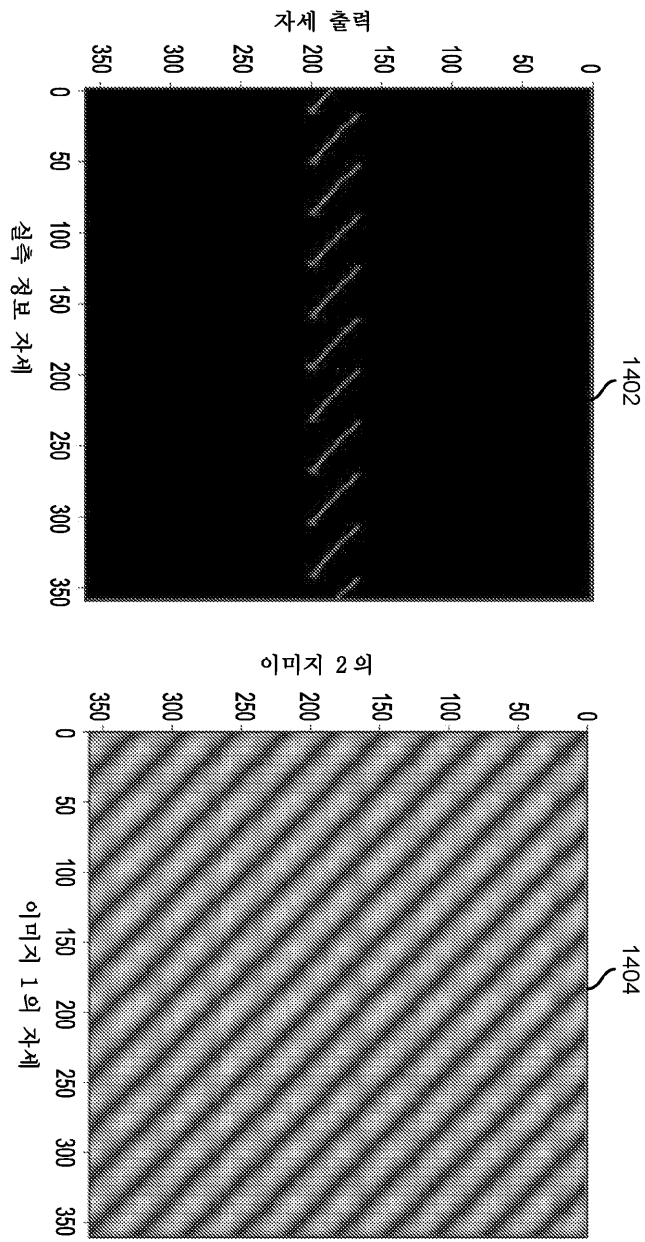
도면12



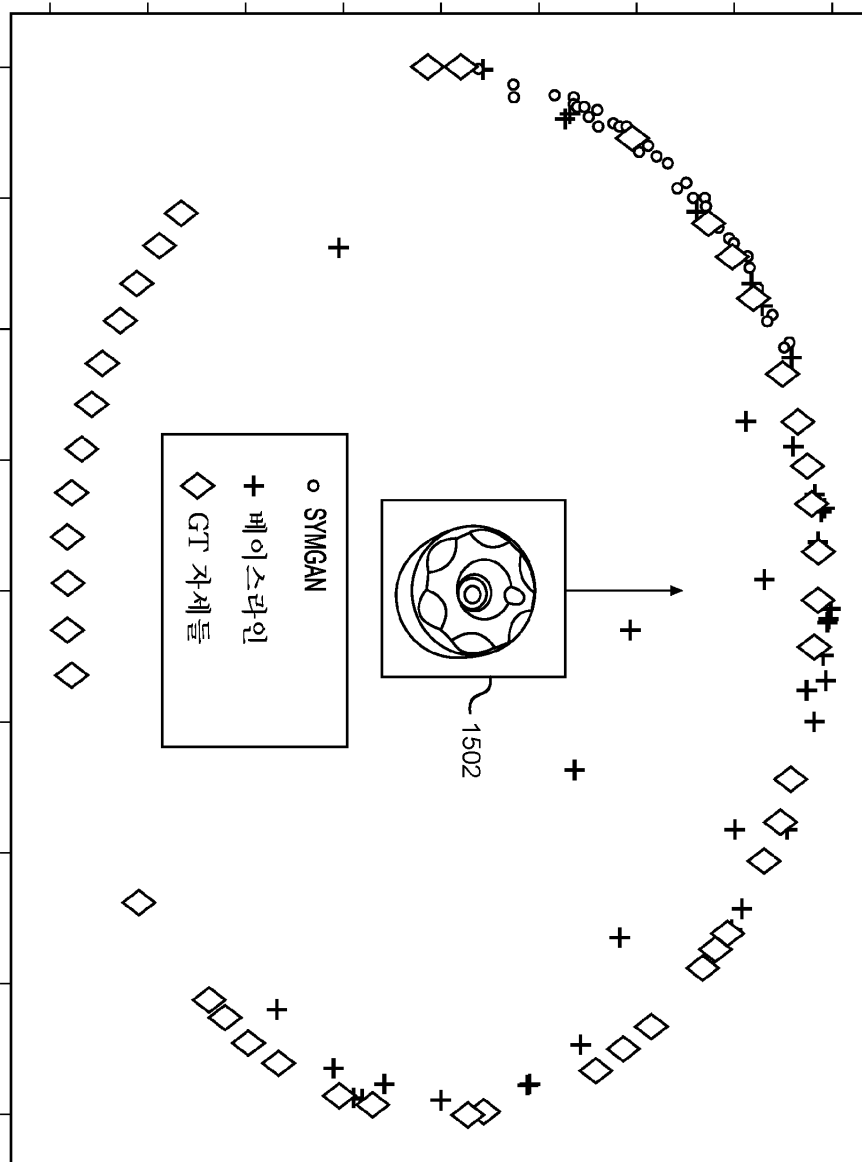
도면13



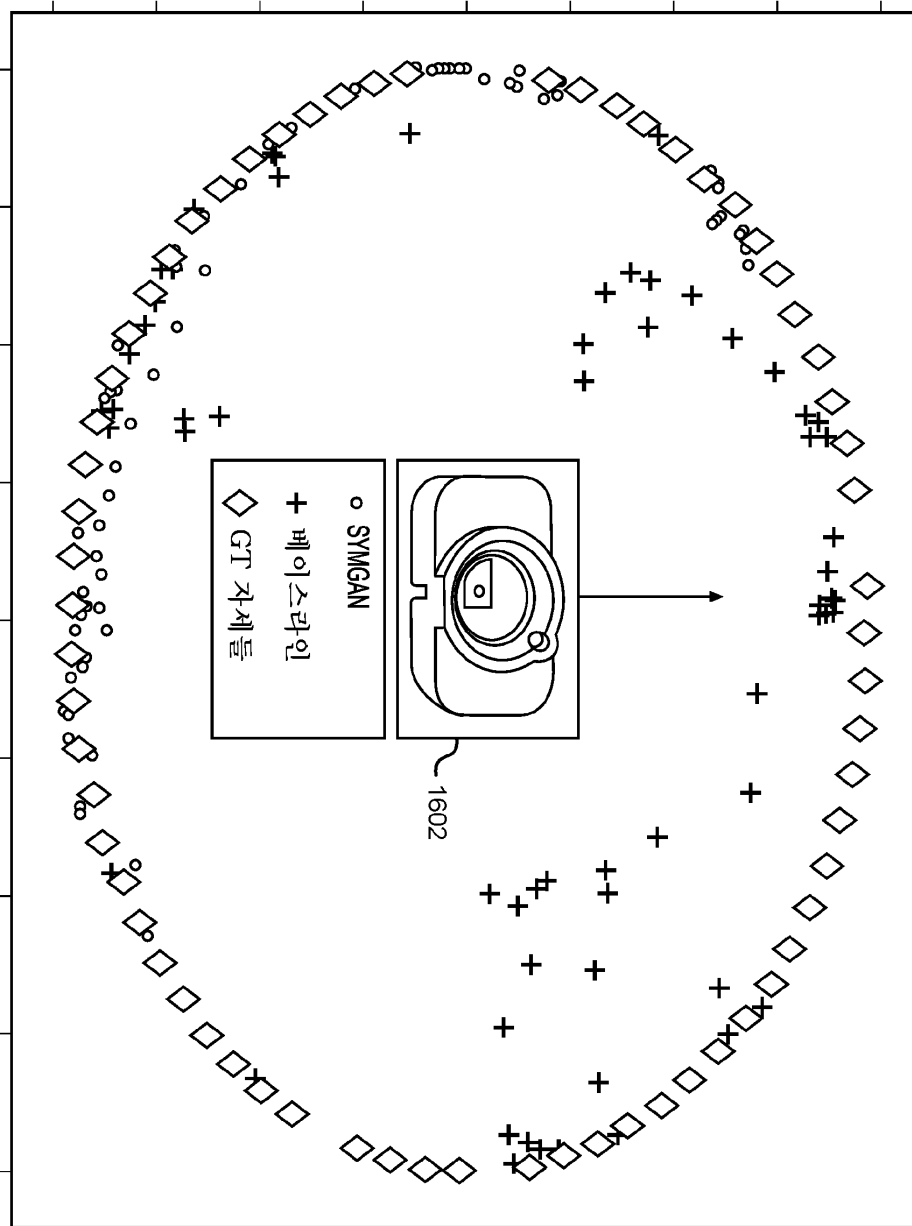
도면14



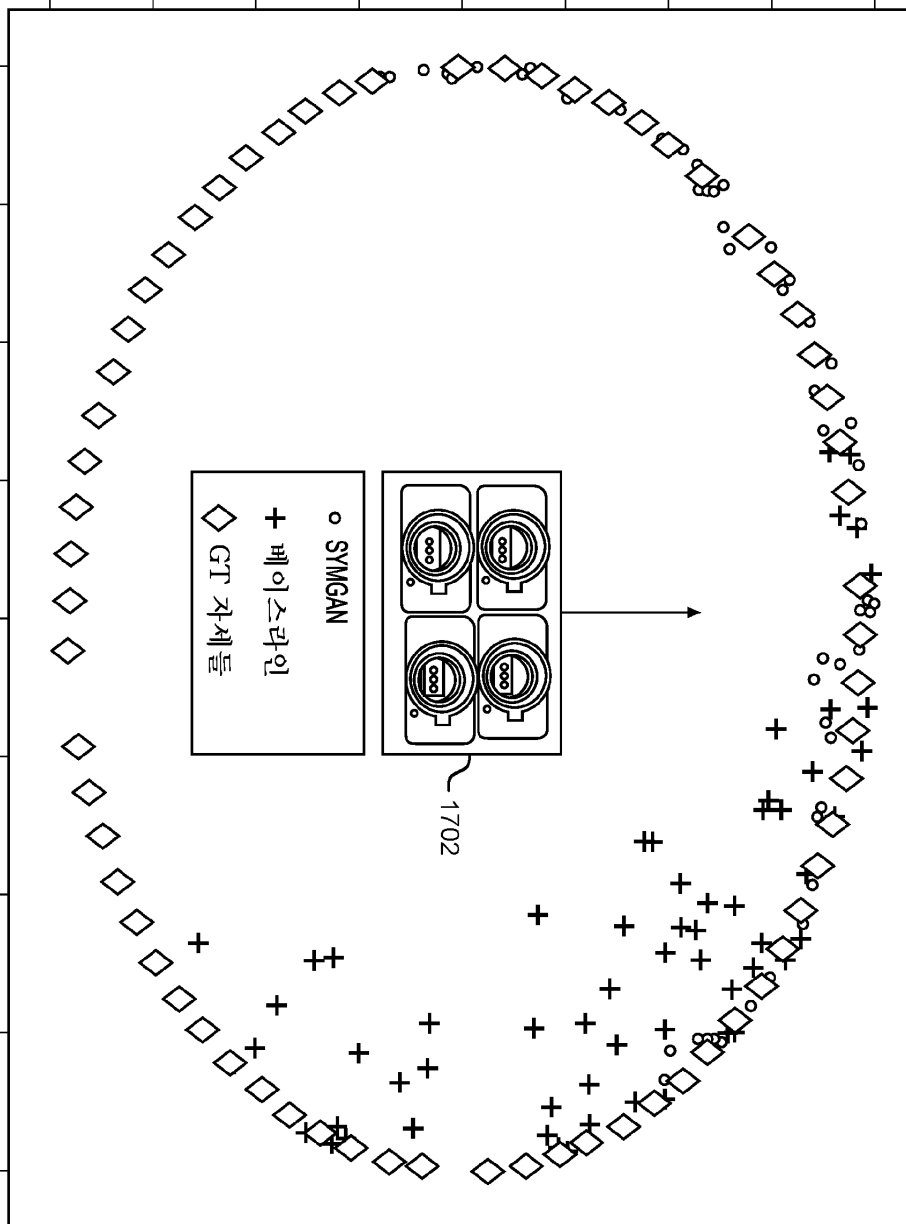
도면15



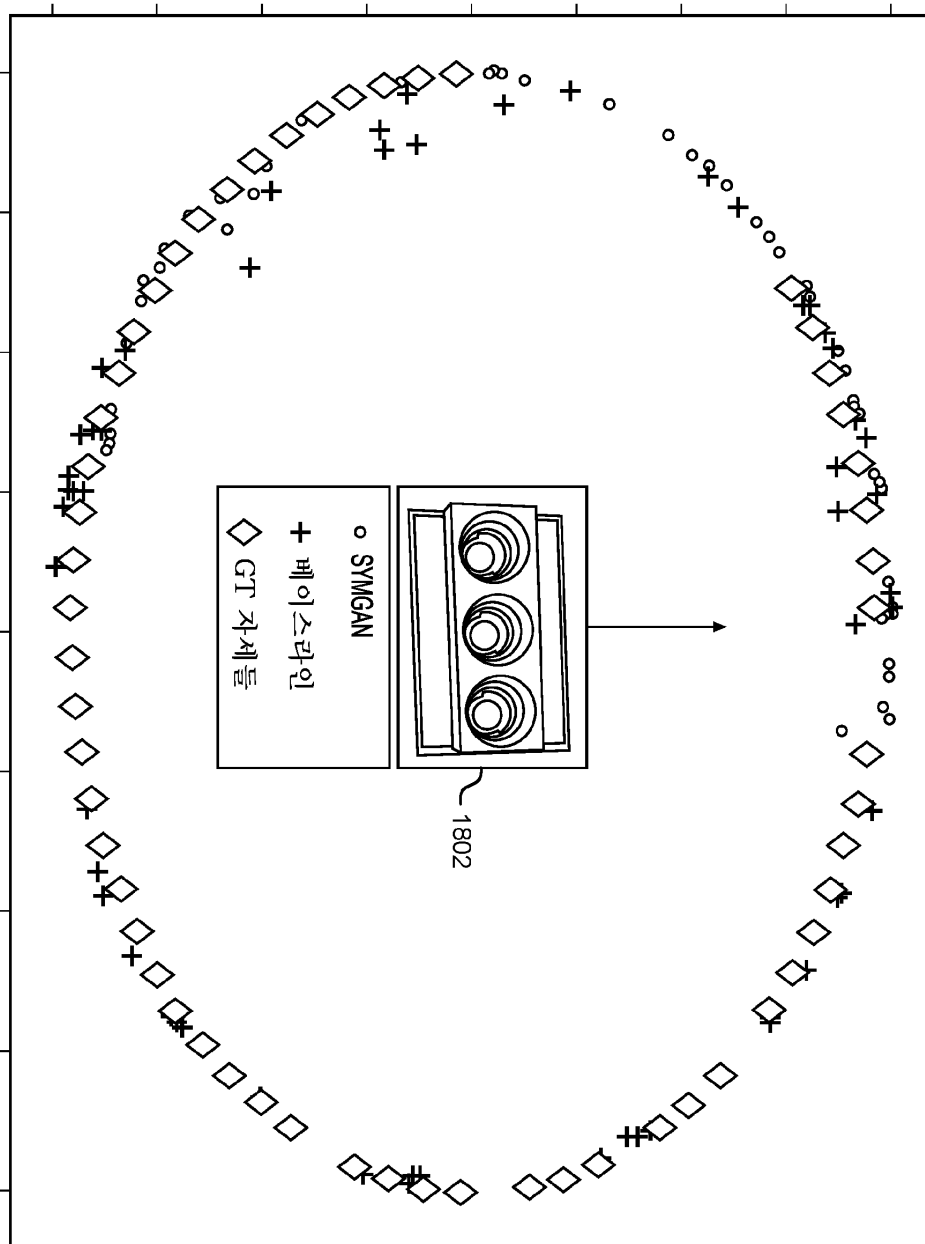
도면16



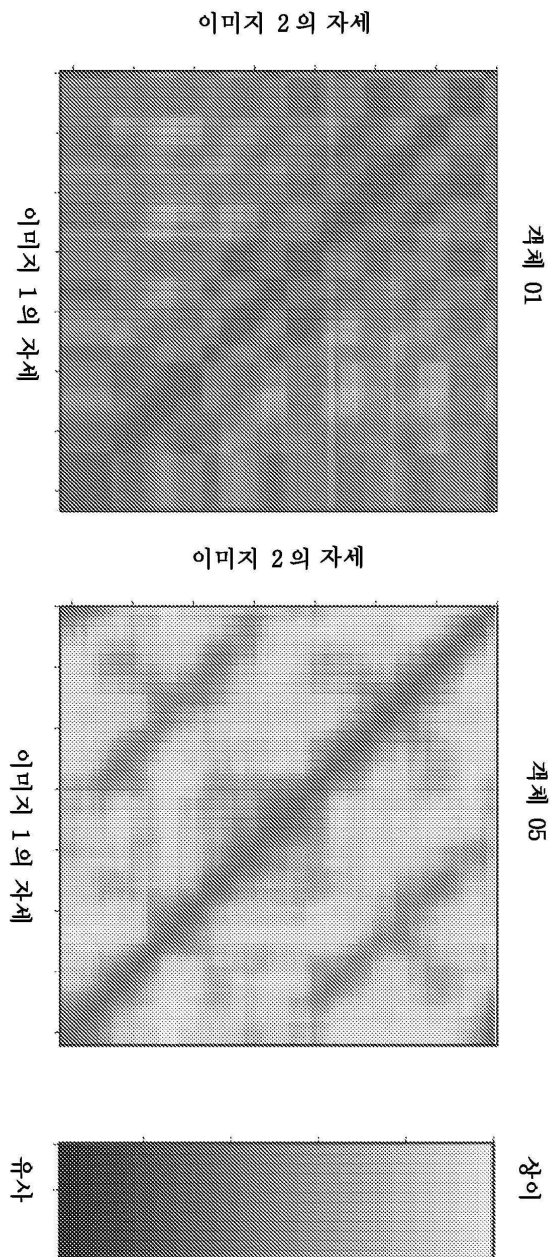
도면17



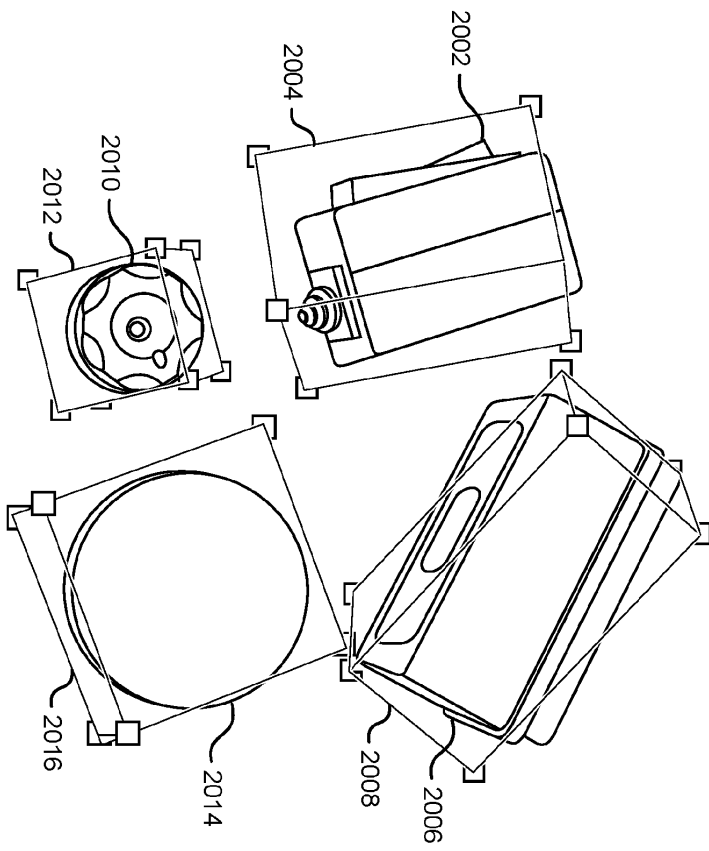
도면18



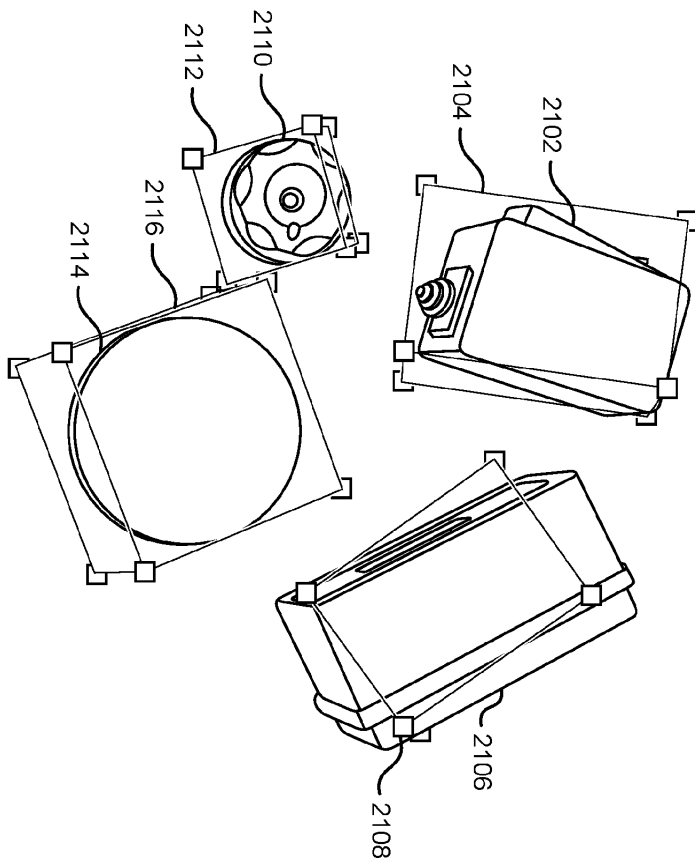
도면19



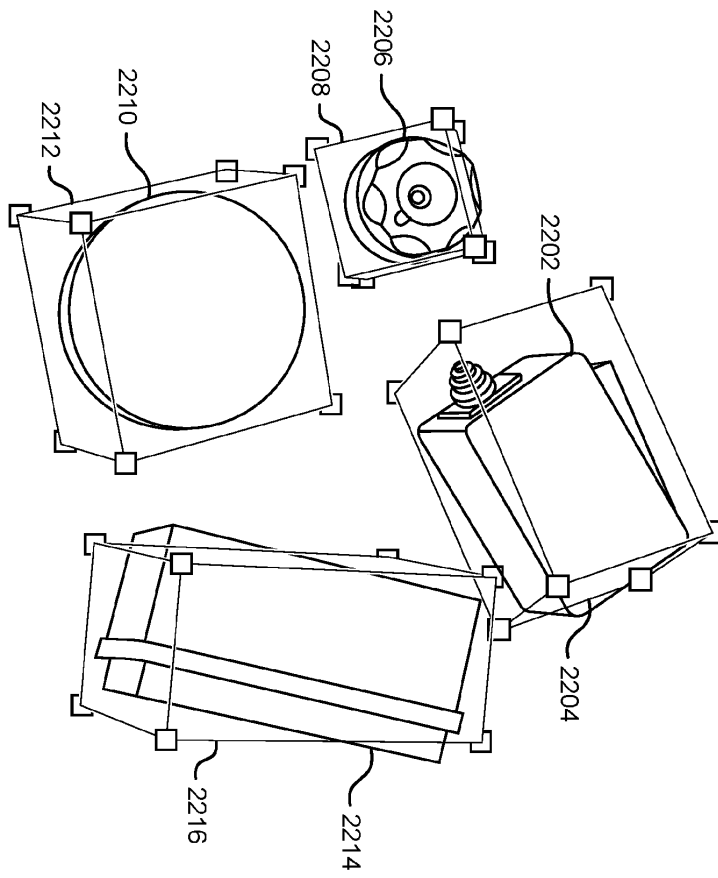
도면20



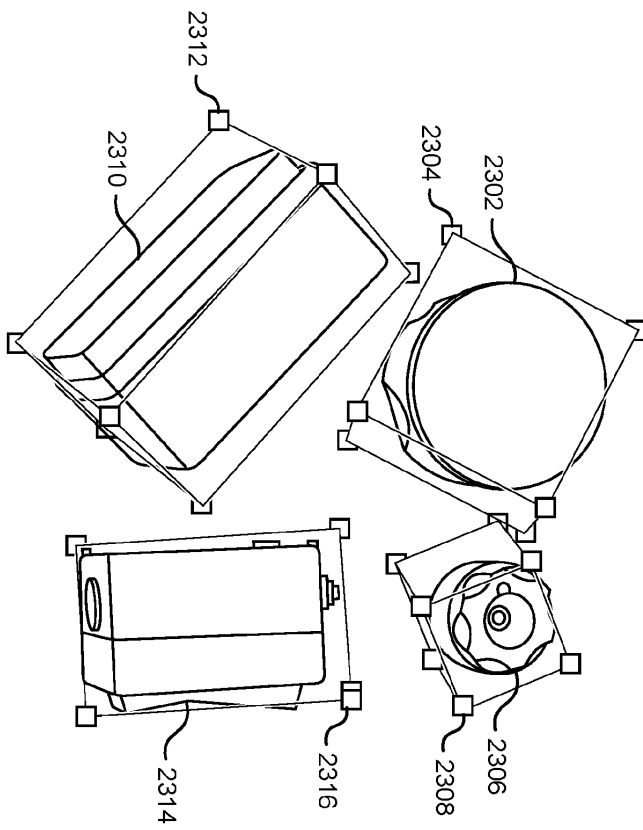
도면21



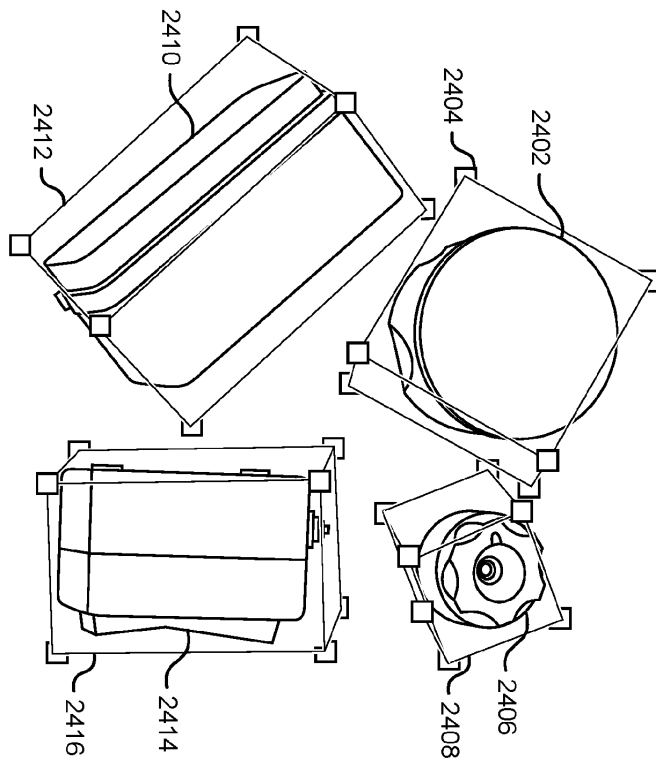
도면22



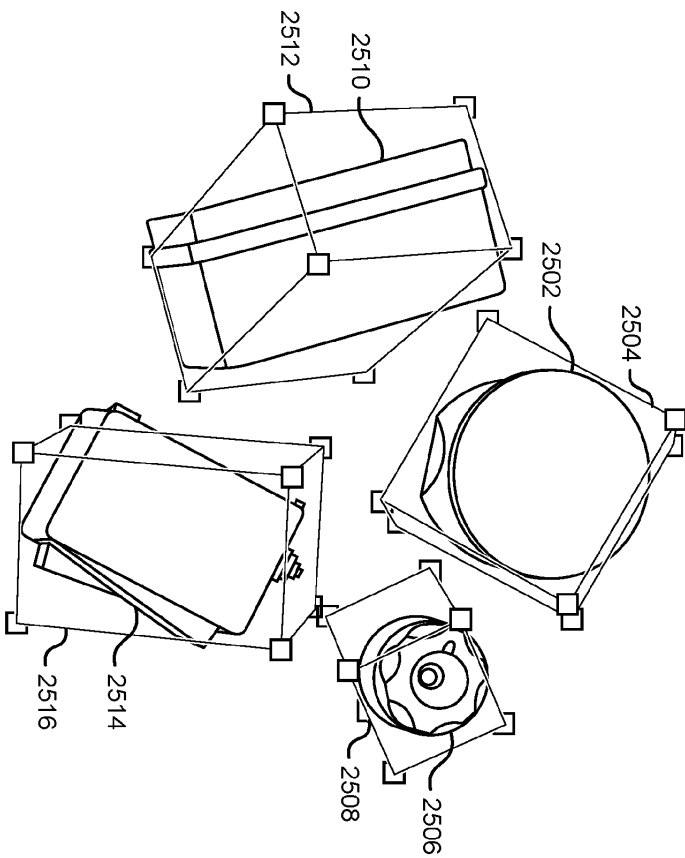
도면23



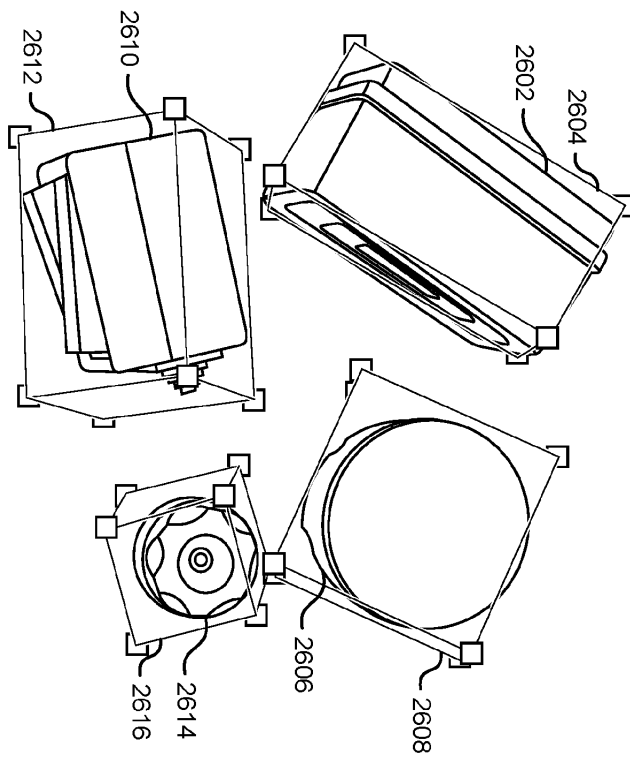
도면24



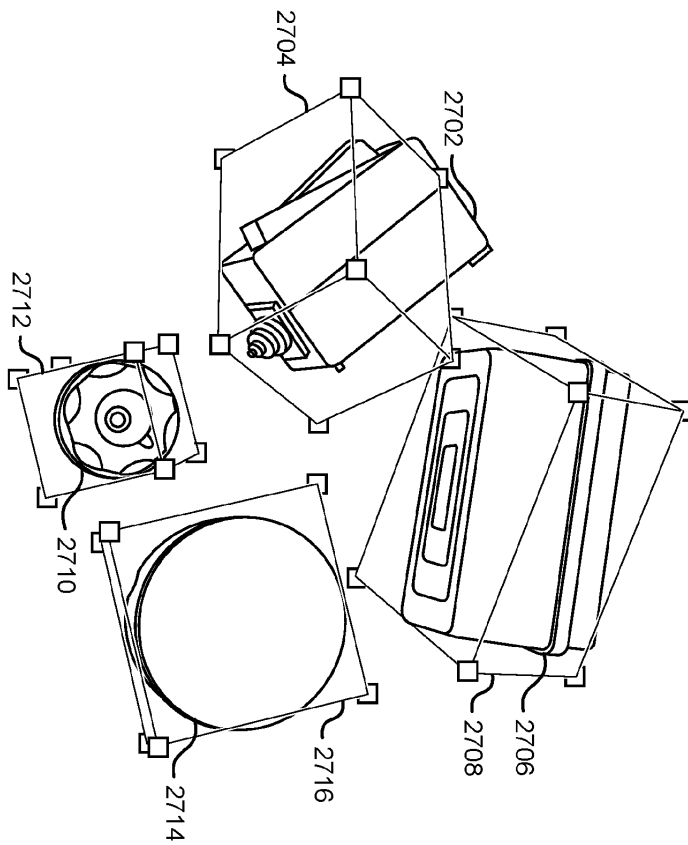
도면25



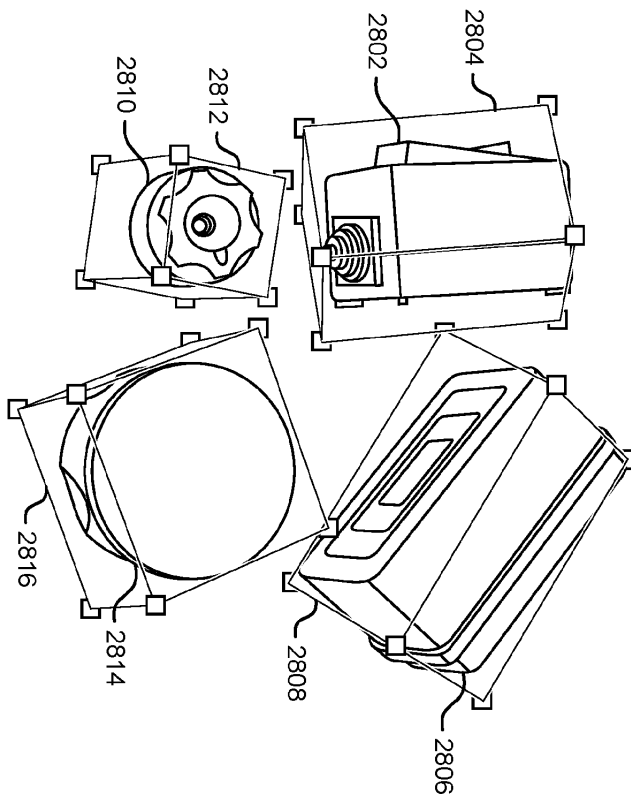
도면26



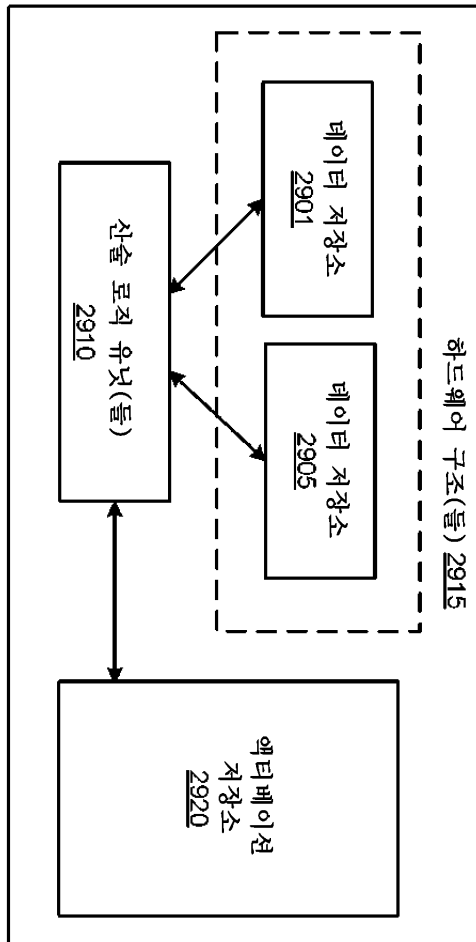
도면27



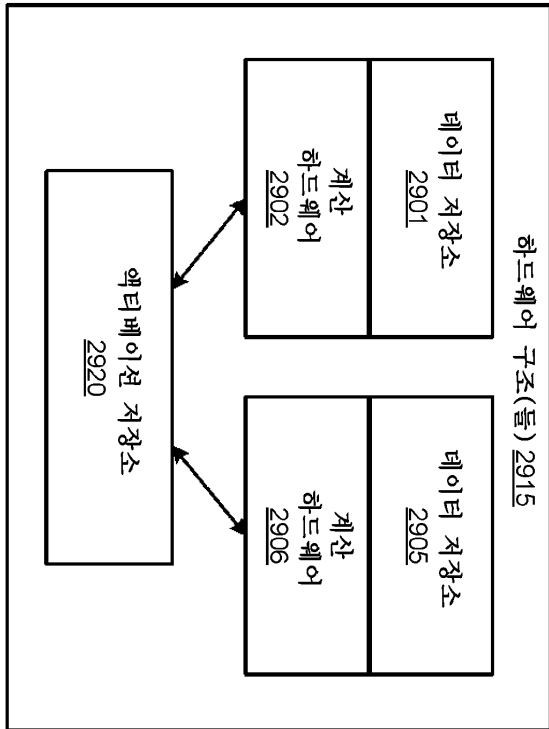
도면28



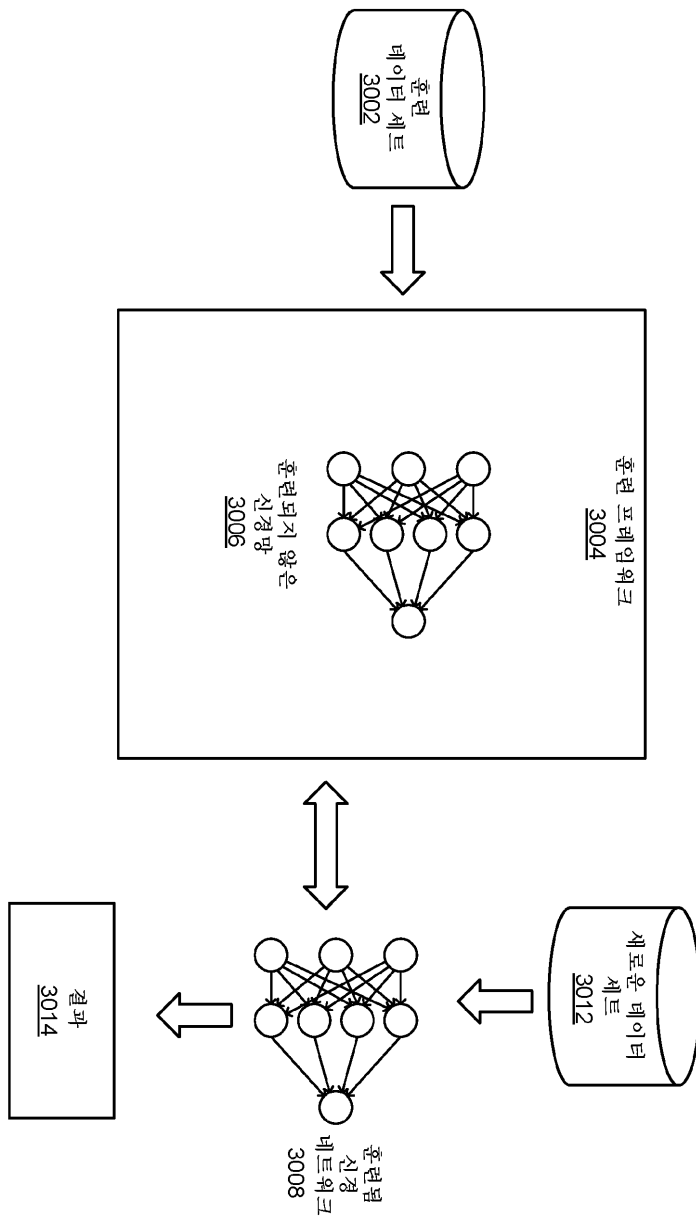
도면29a



도면29b

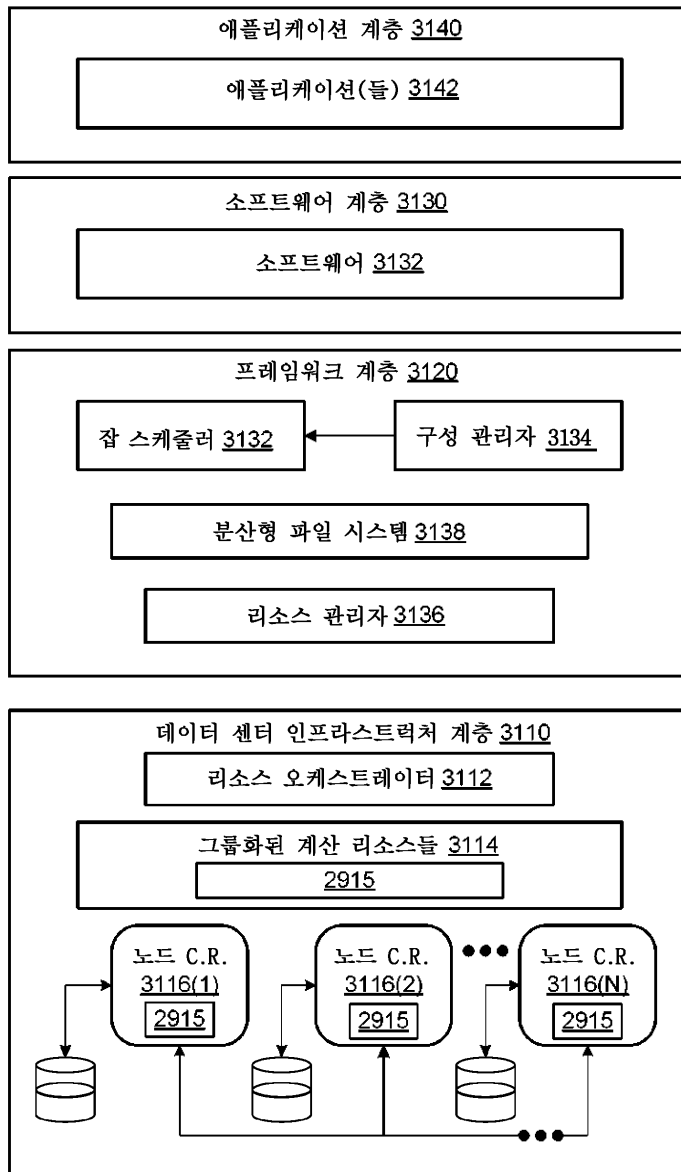


도면30

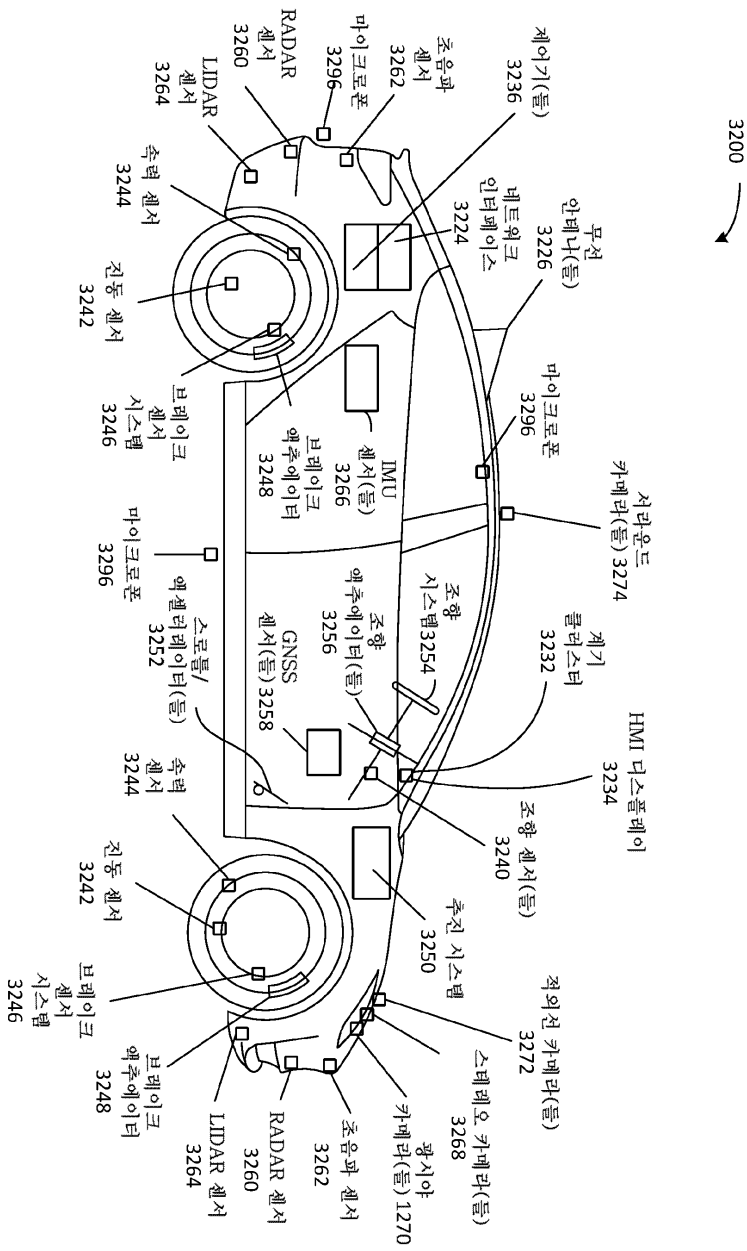


도면31

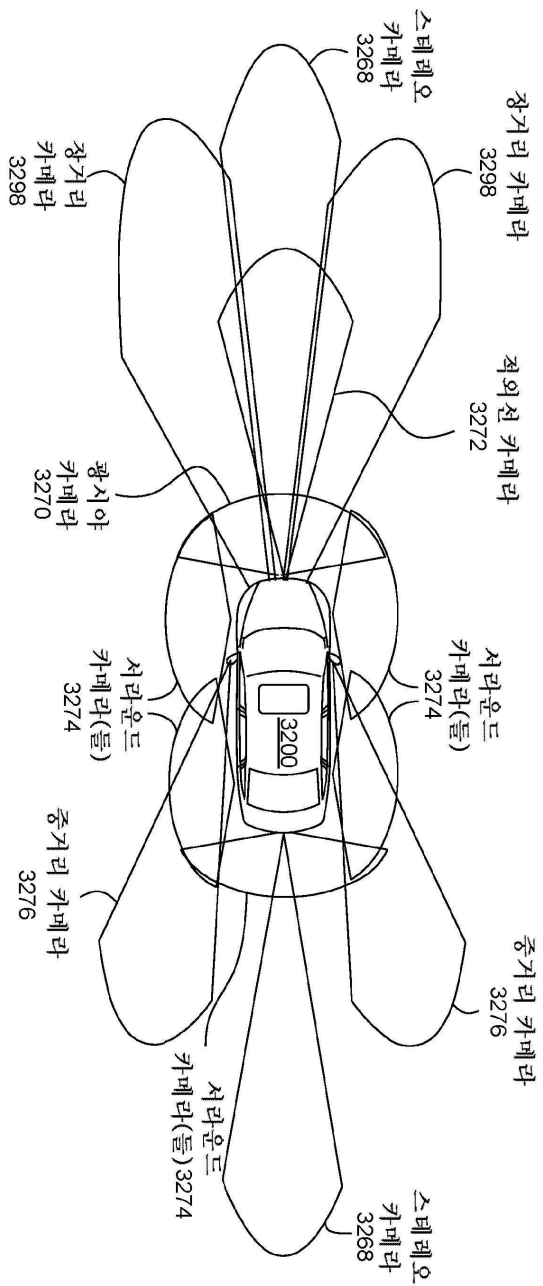
데이터 센터
3100



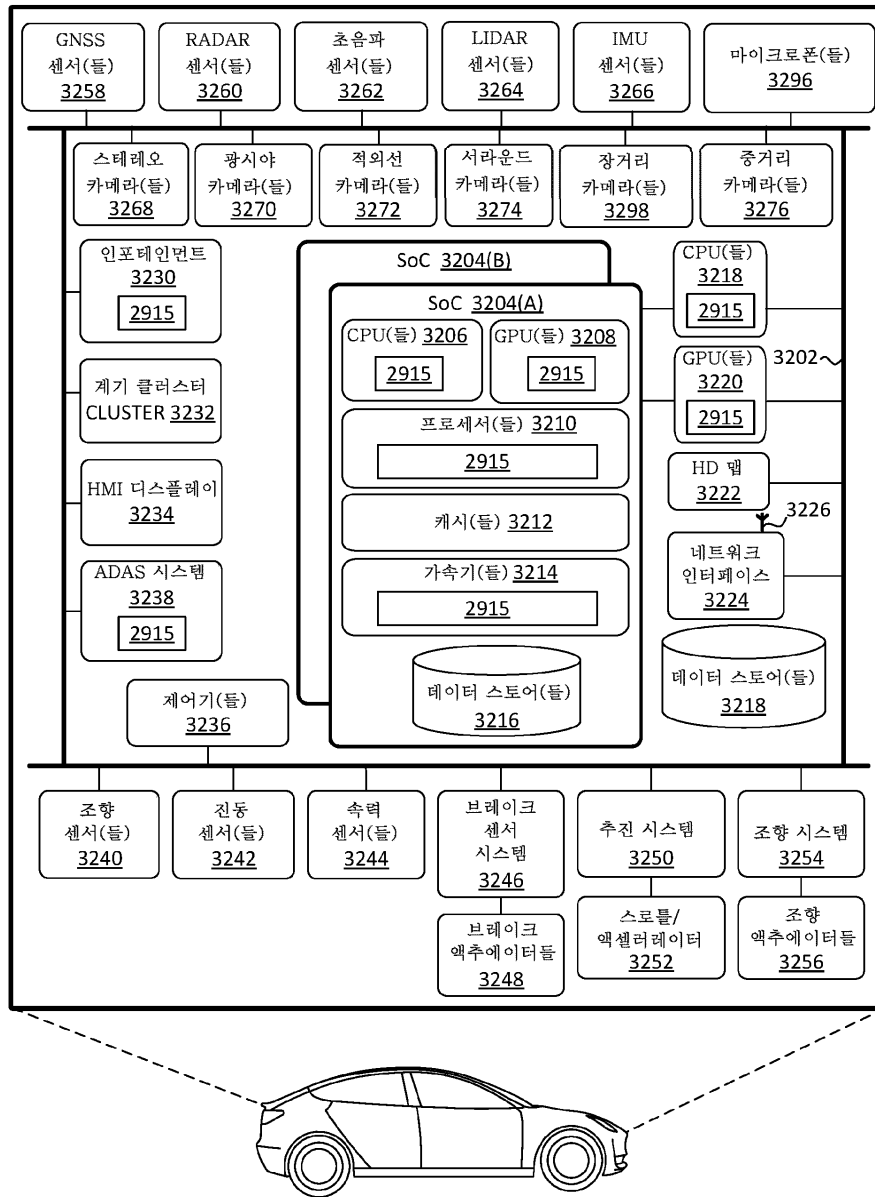
도면32a



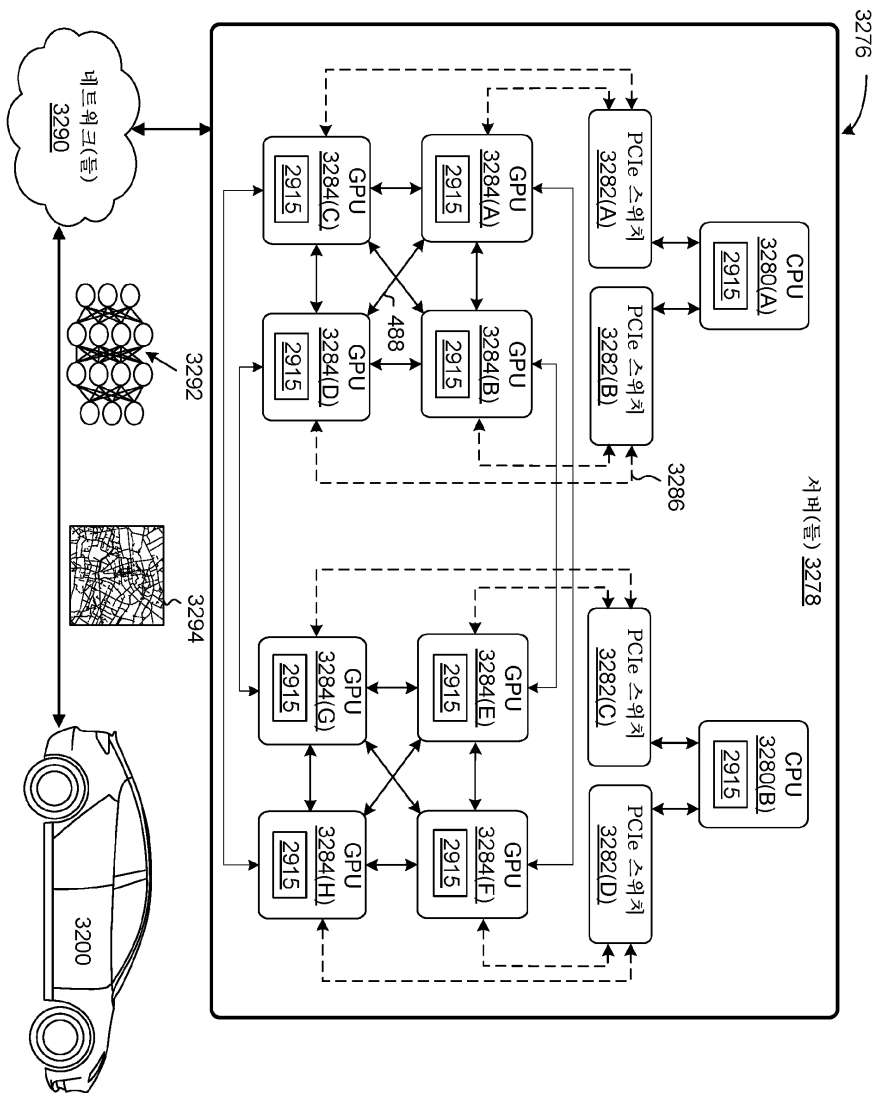
도면32b



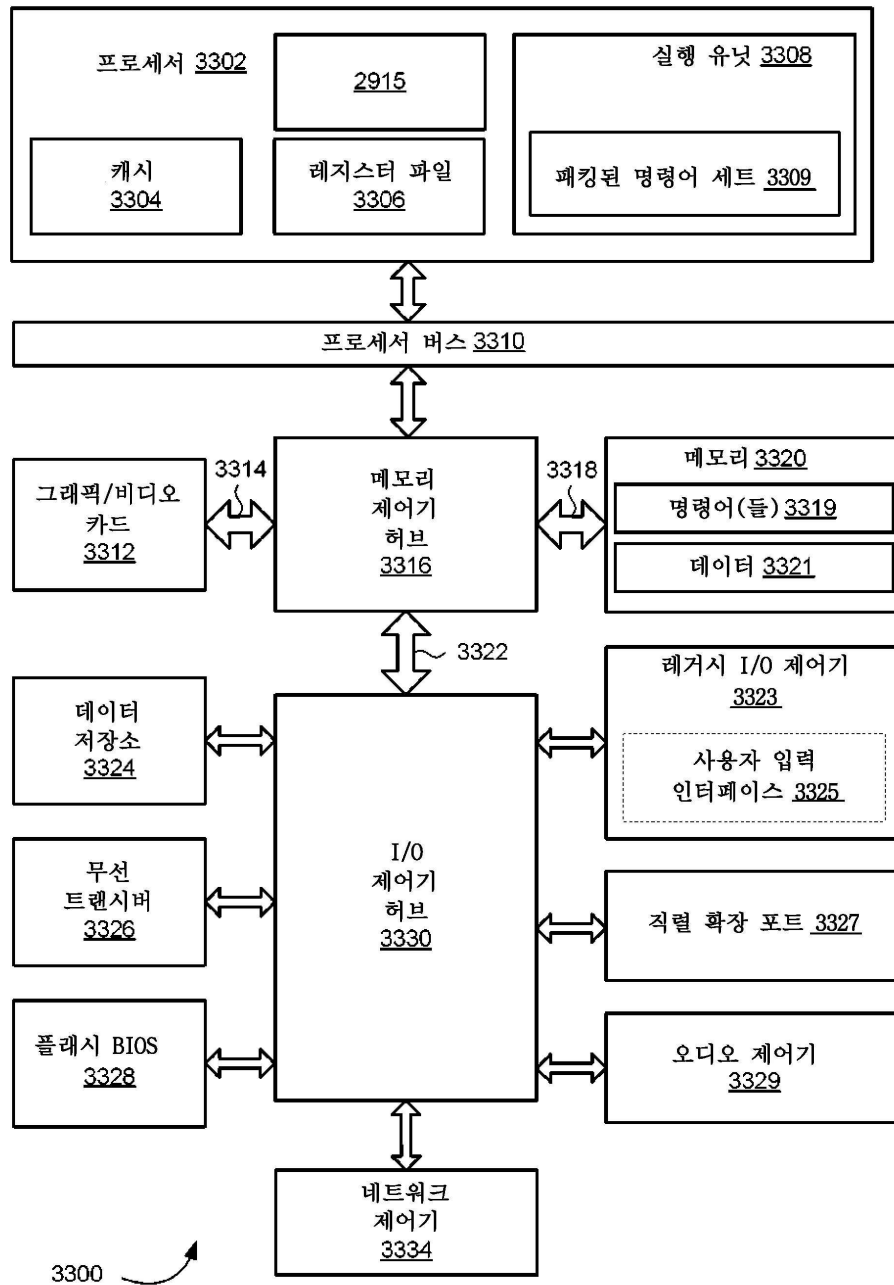
도면 32c



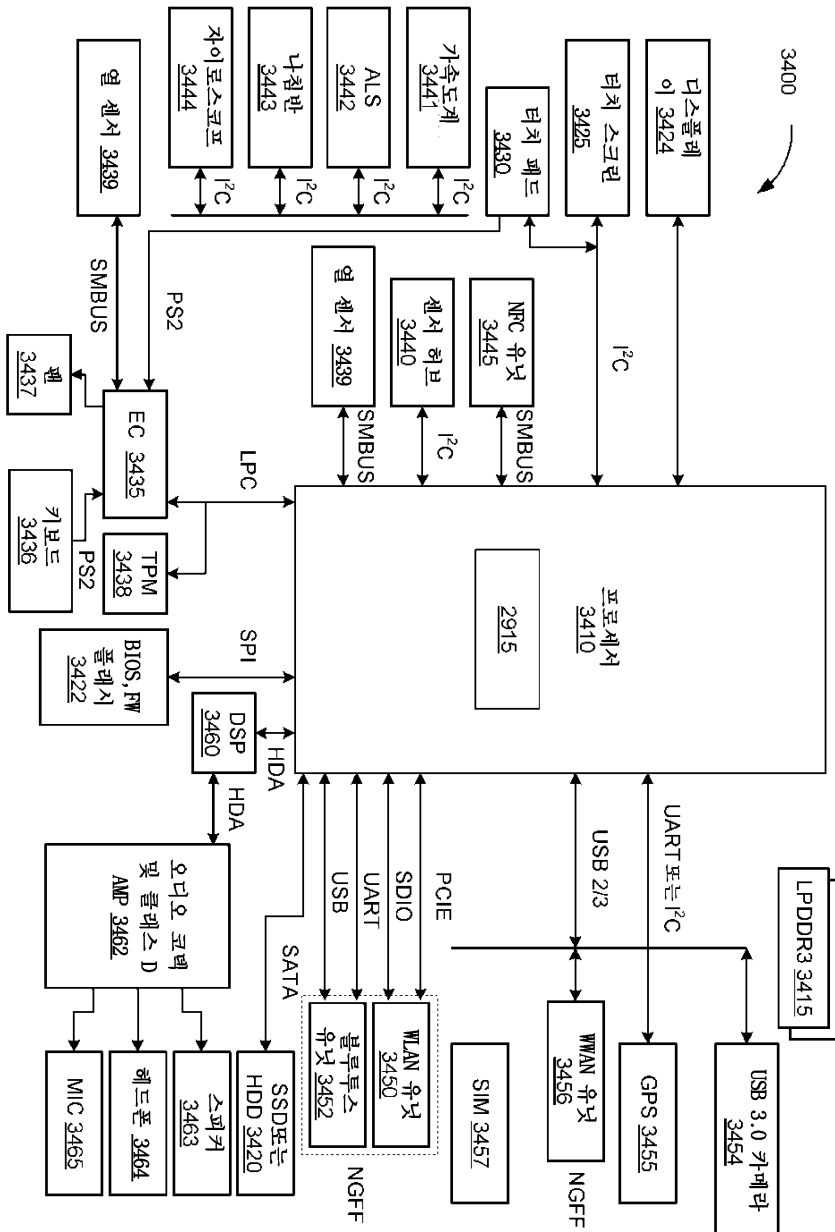
도면32d



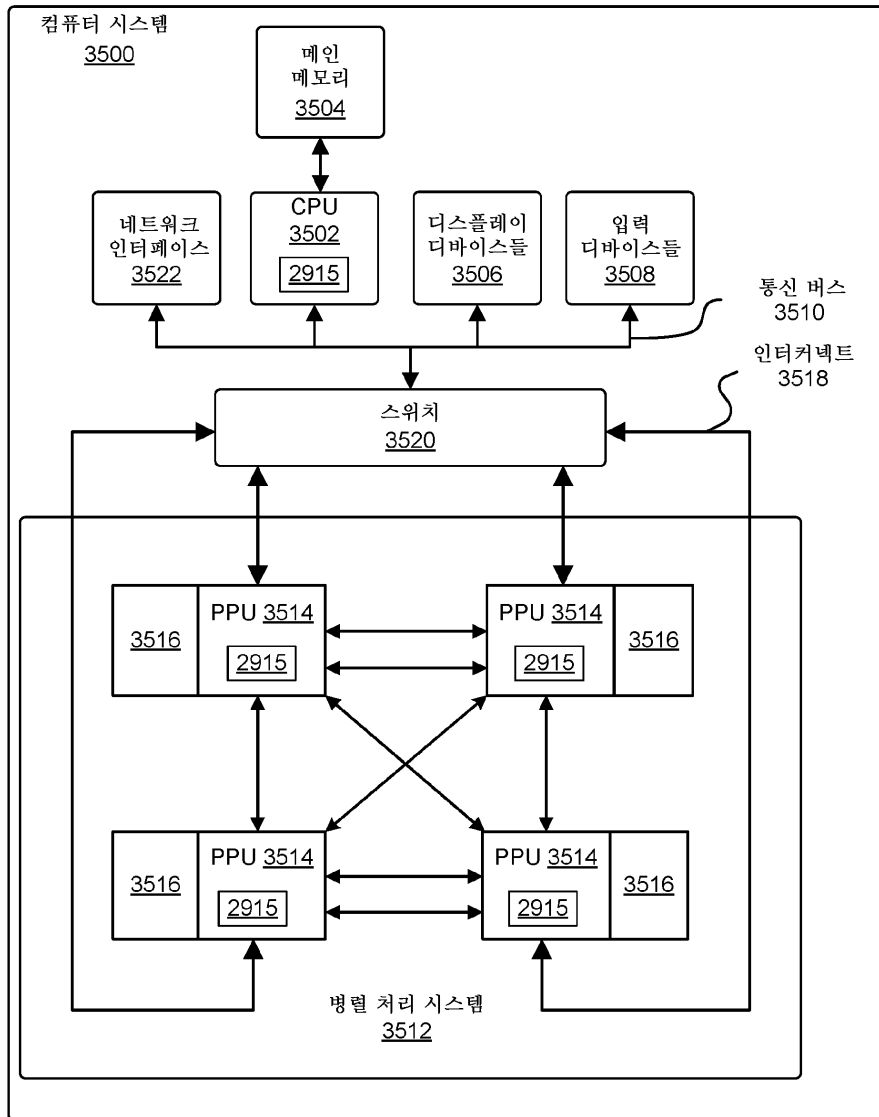
도면33



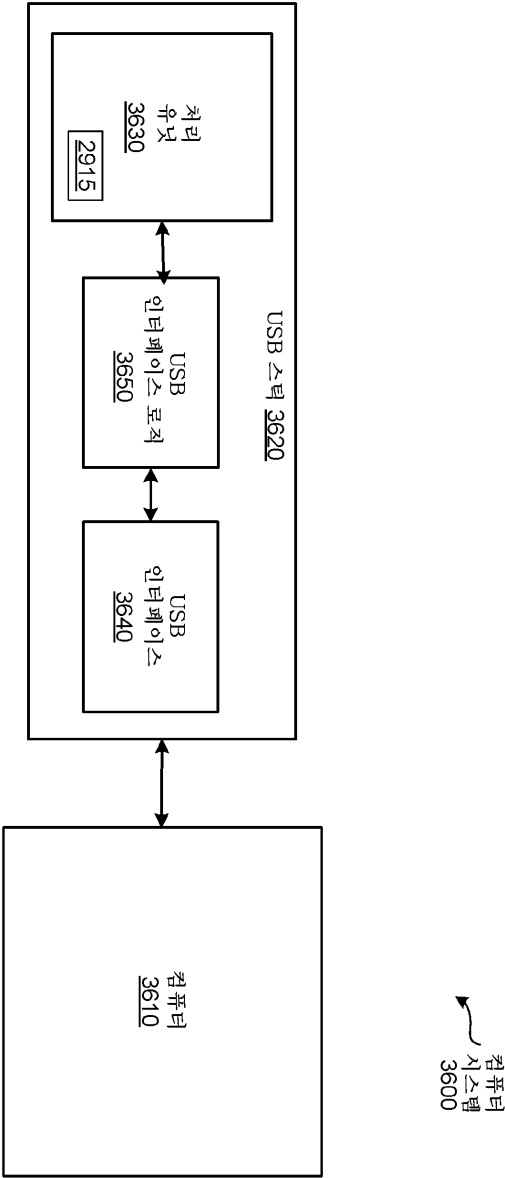
도면34



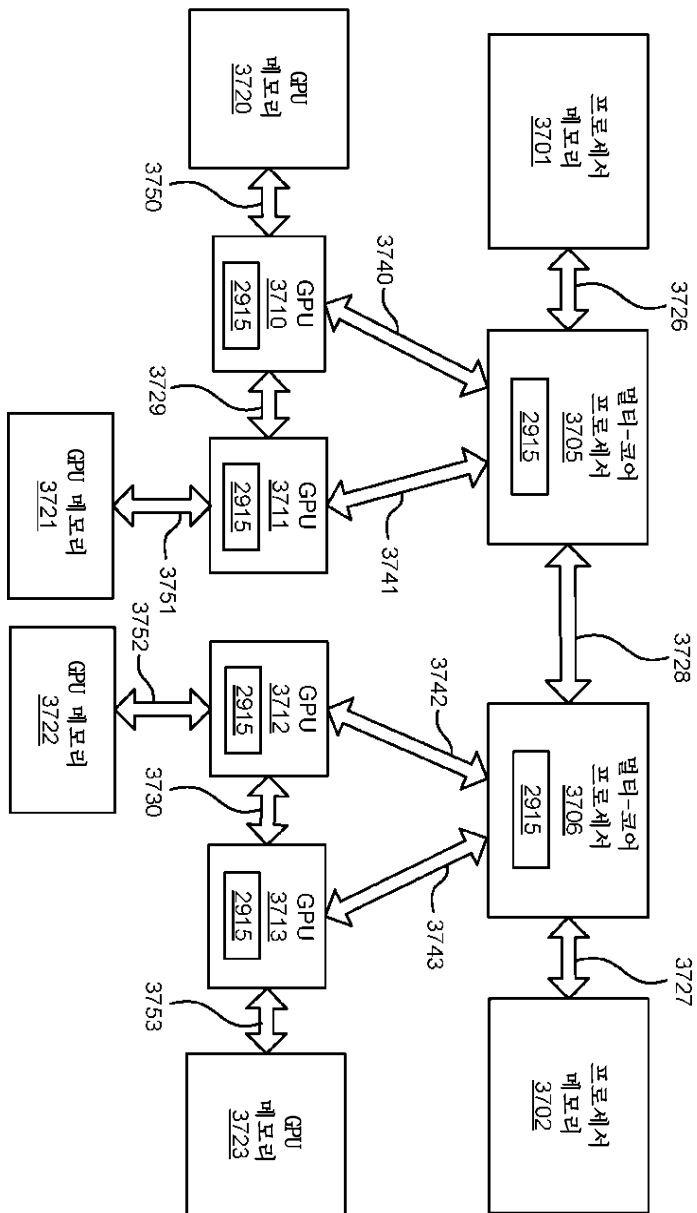
도면35



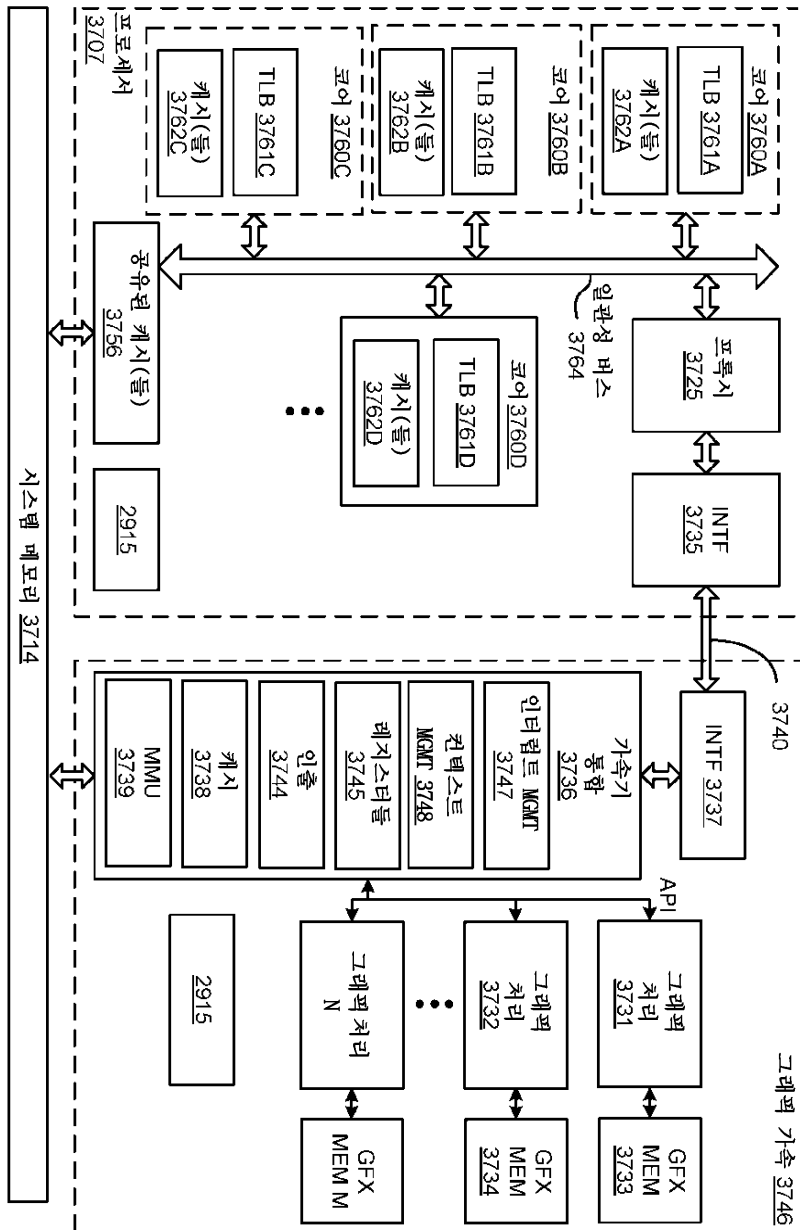
도면36



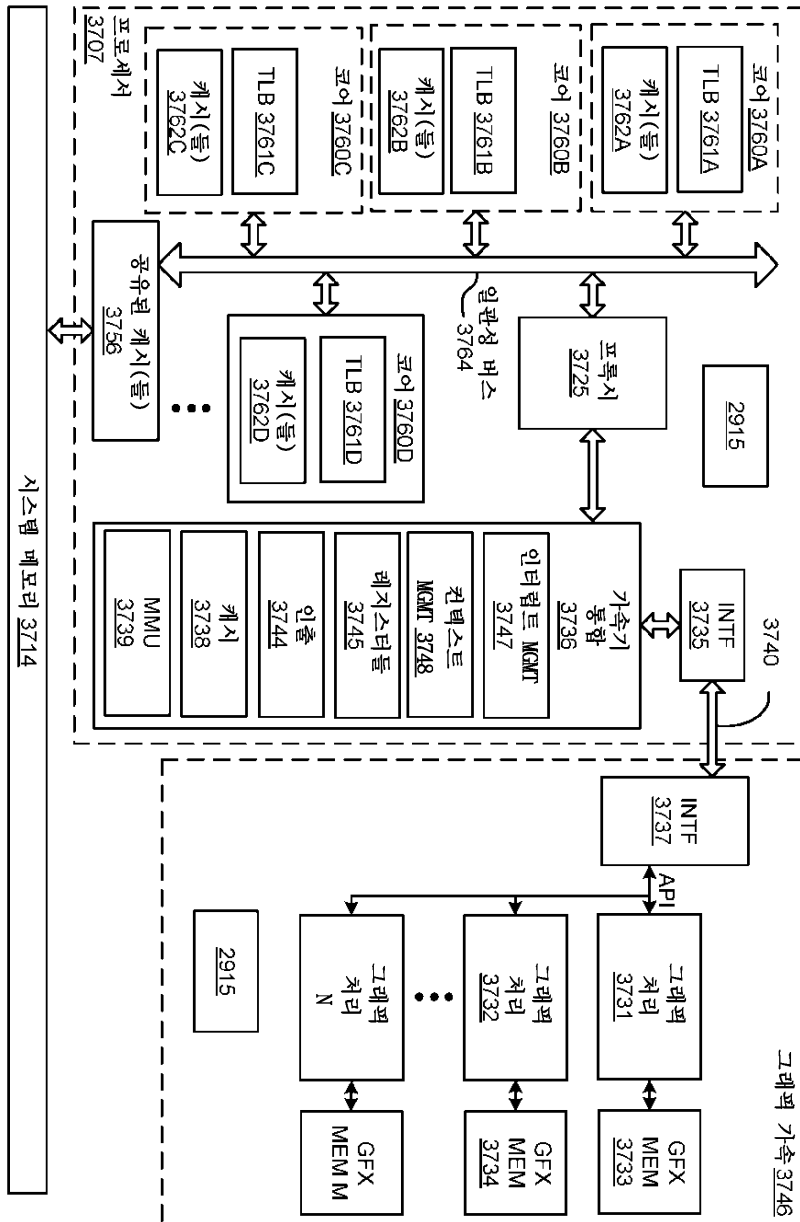
도면37a



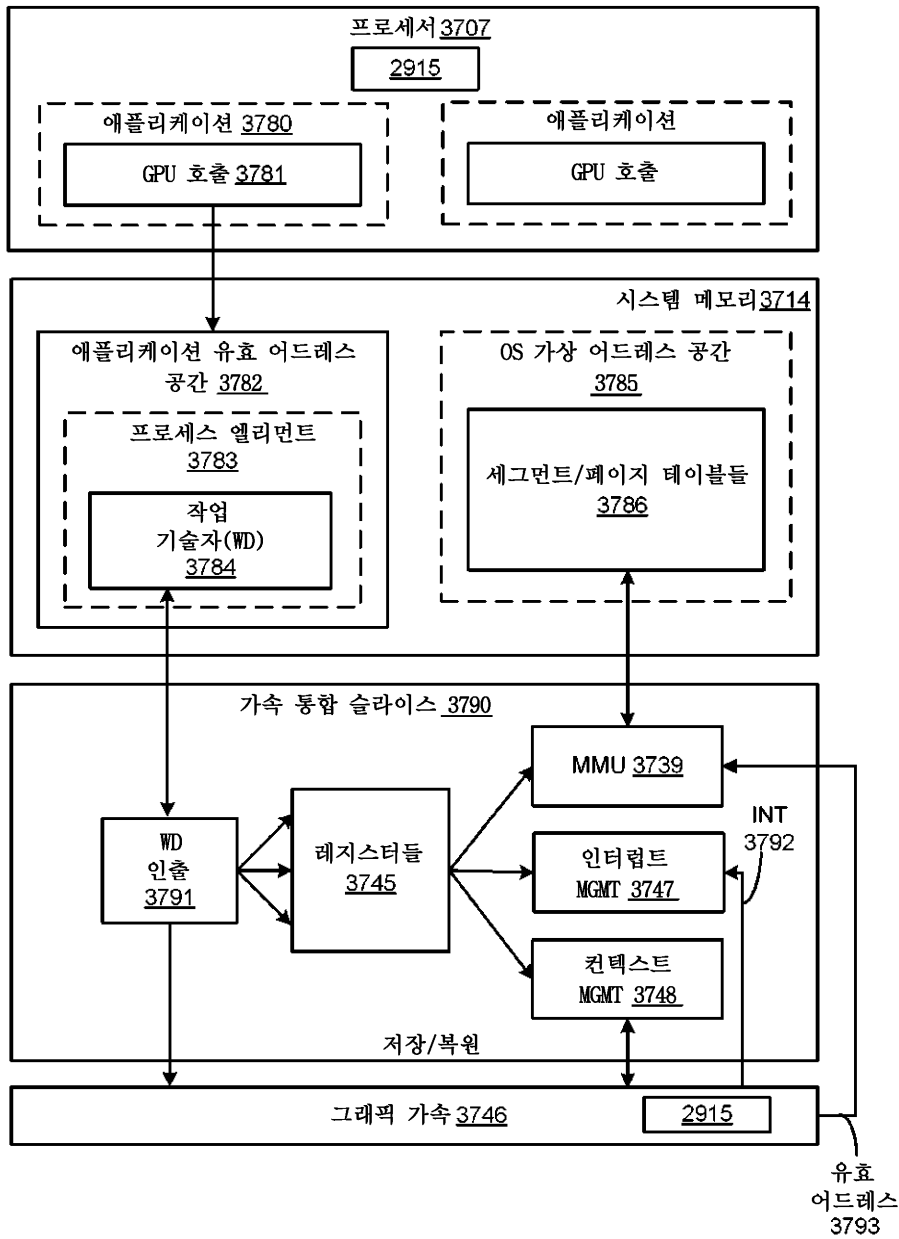
도면37b



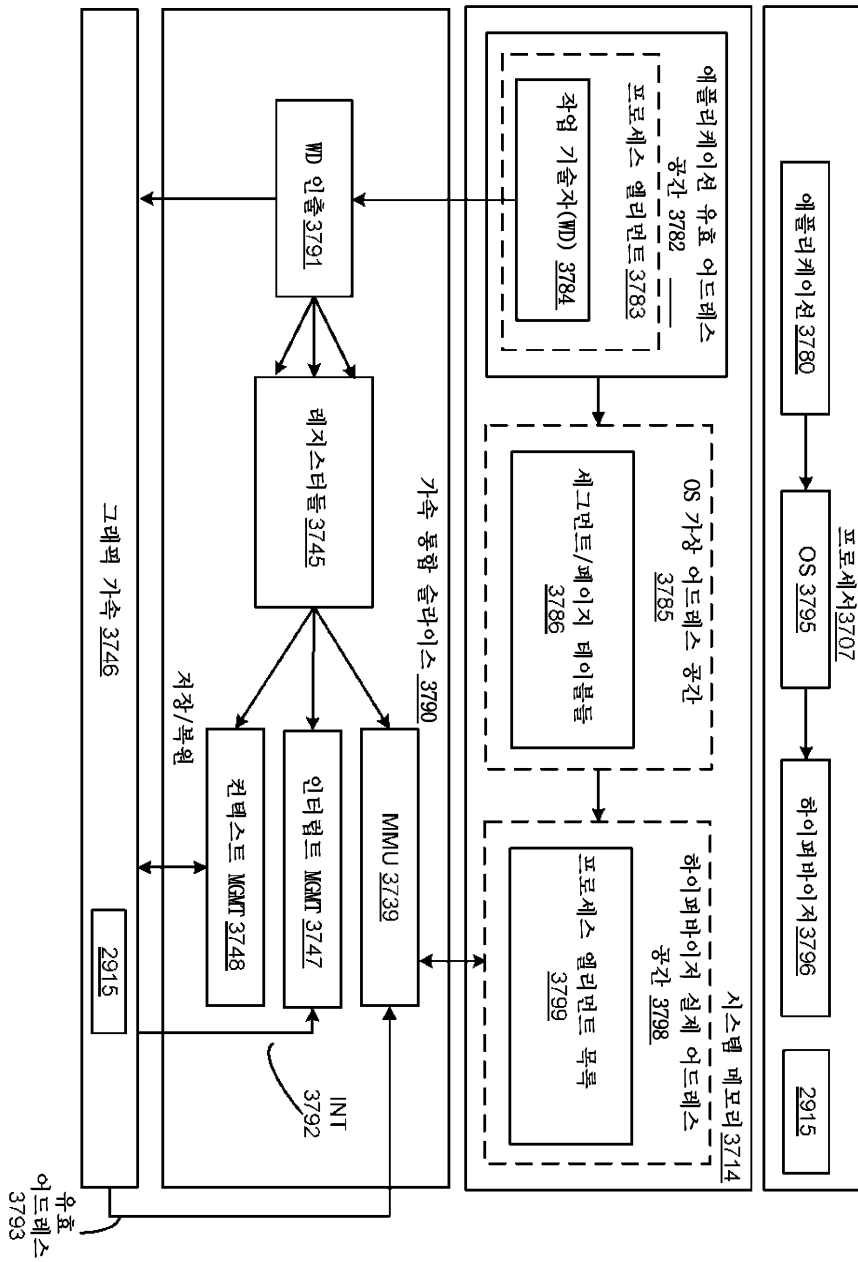
도면37c



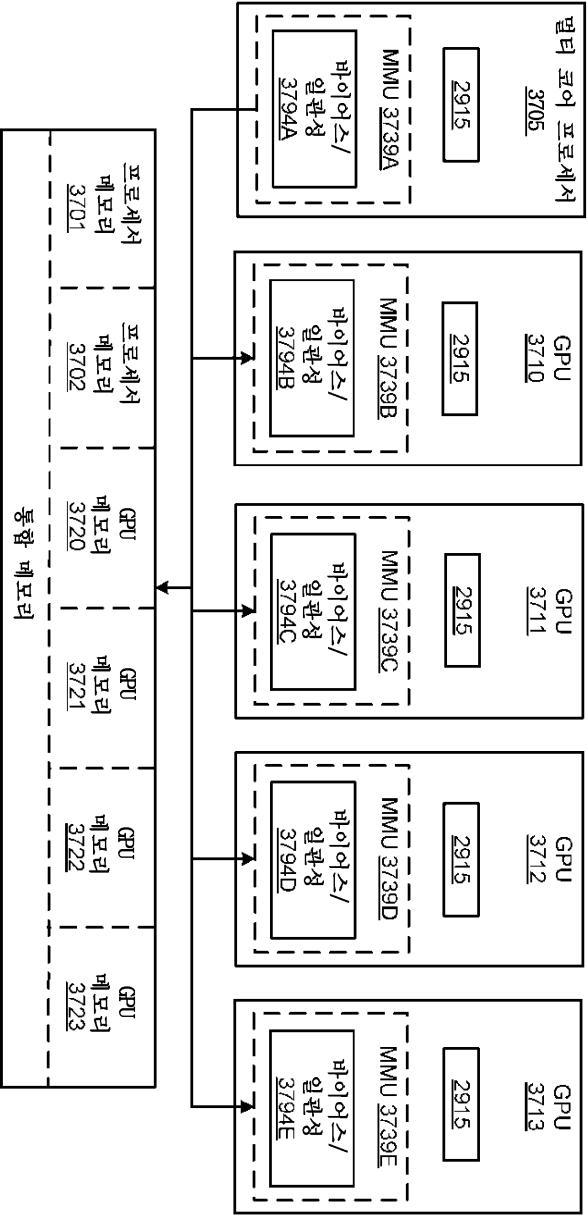
도면37d



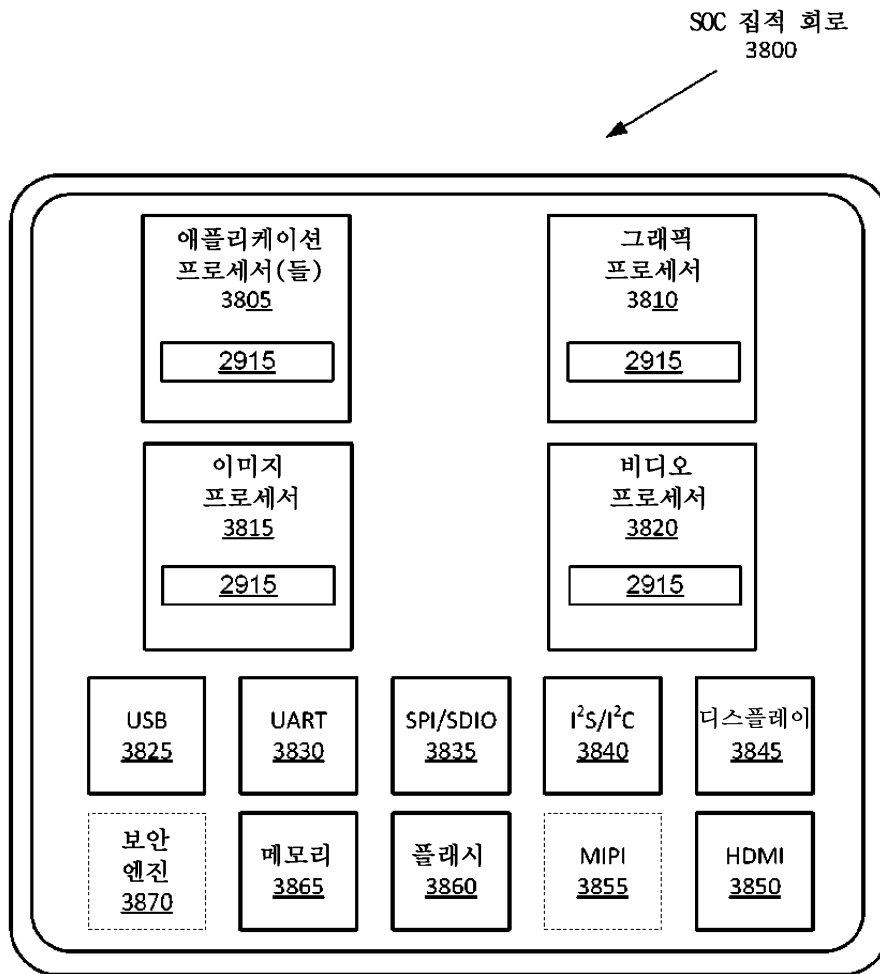
도면37e



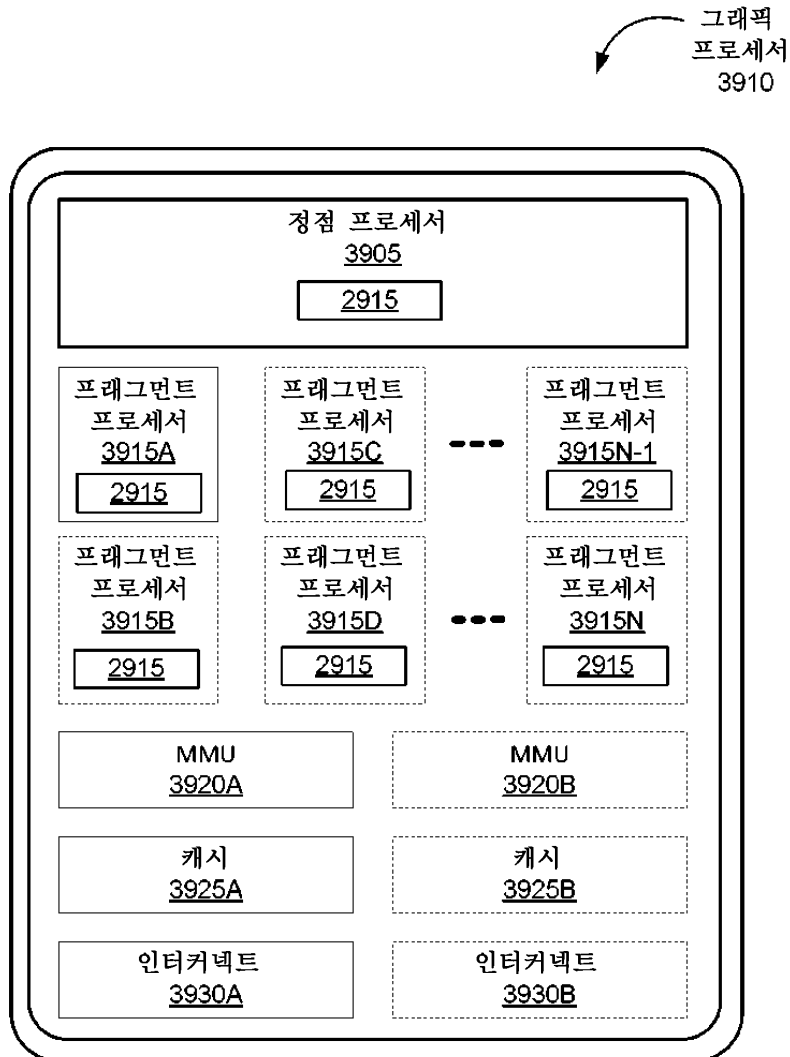
도면37f



도면38



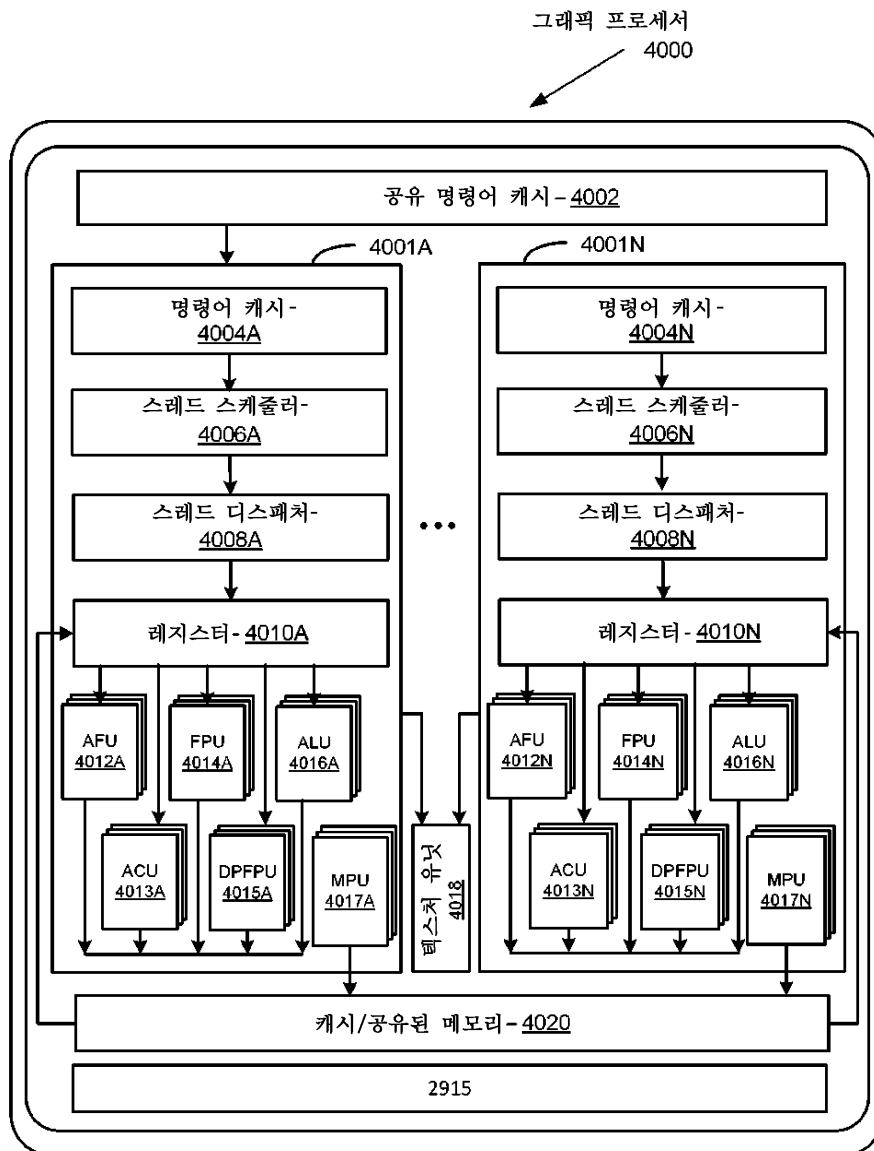
도면 39a



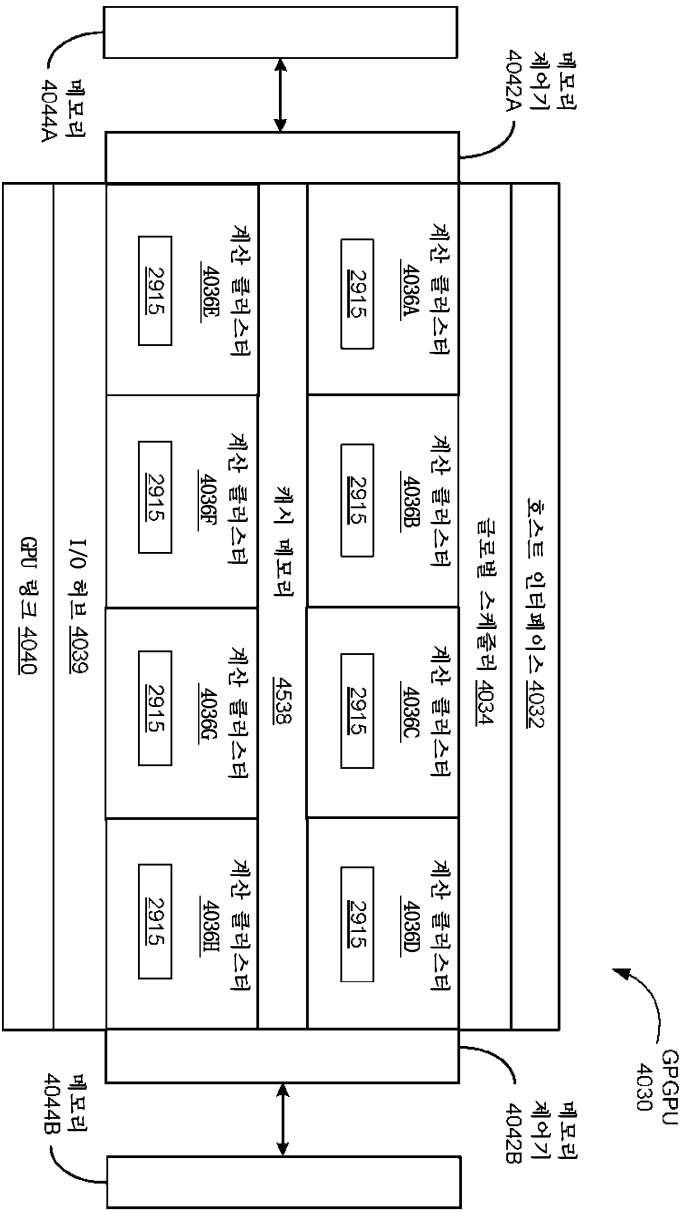
도면39b



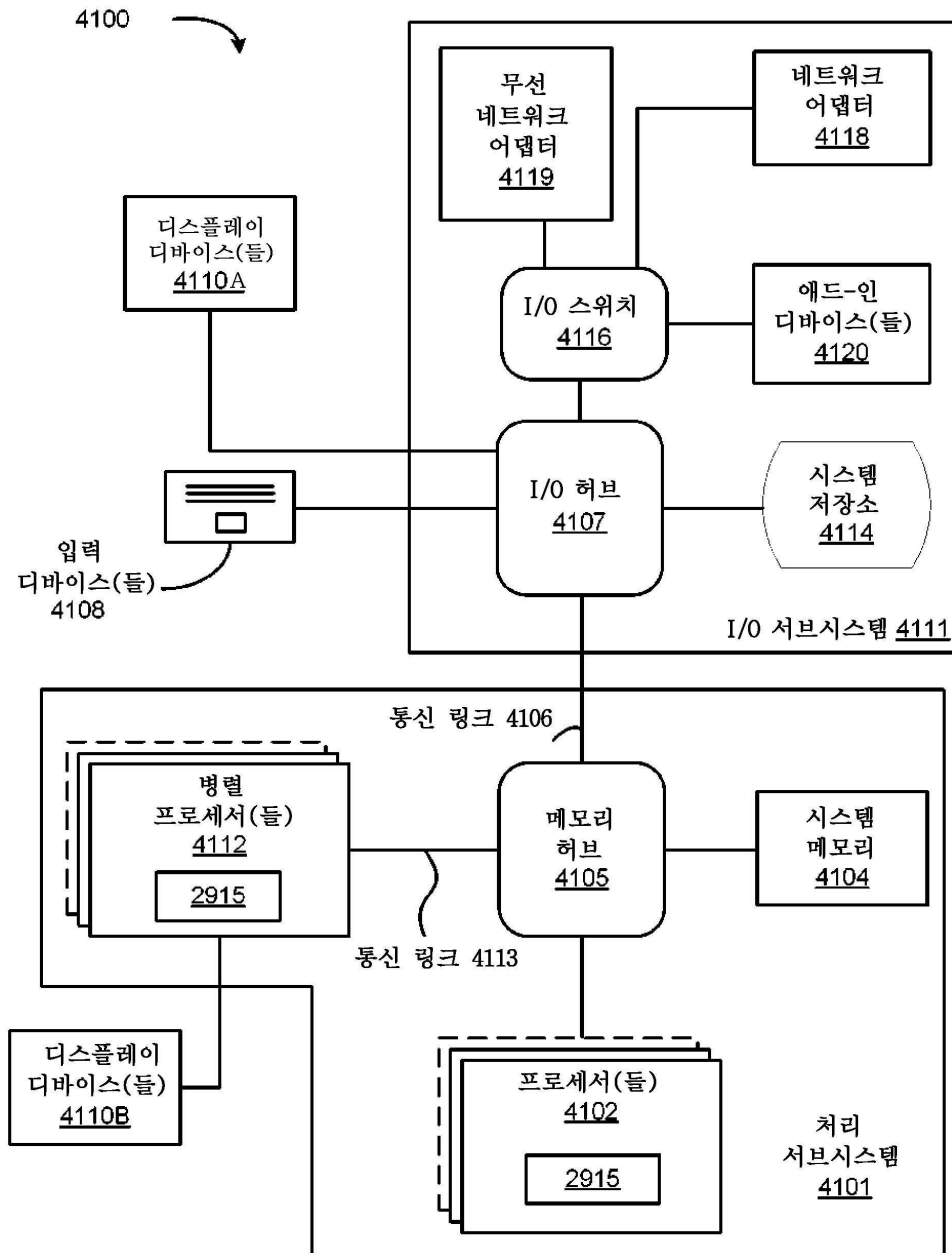
도면 40a



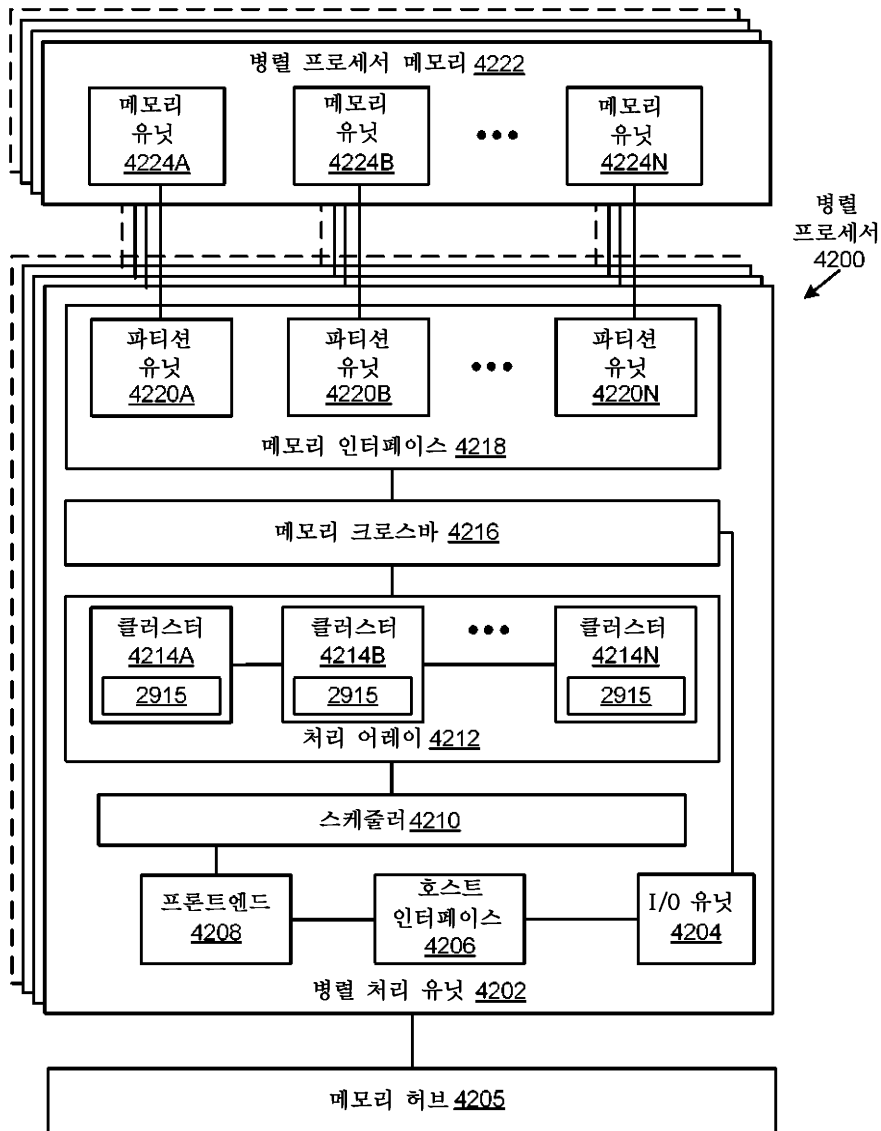
도면40b



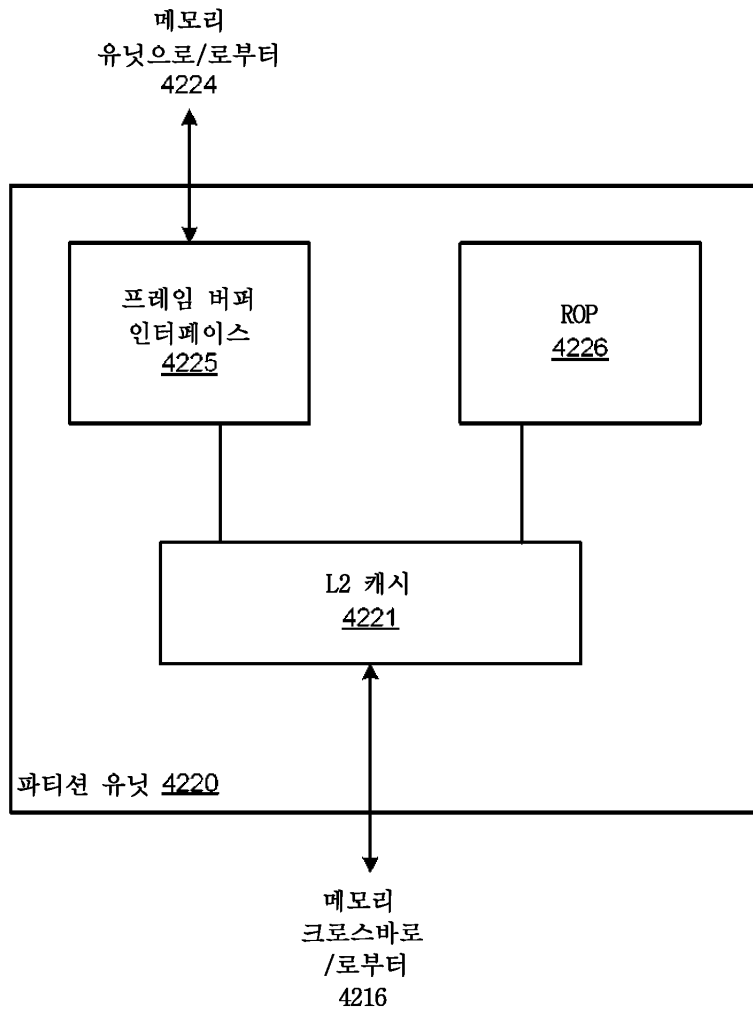
도면41



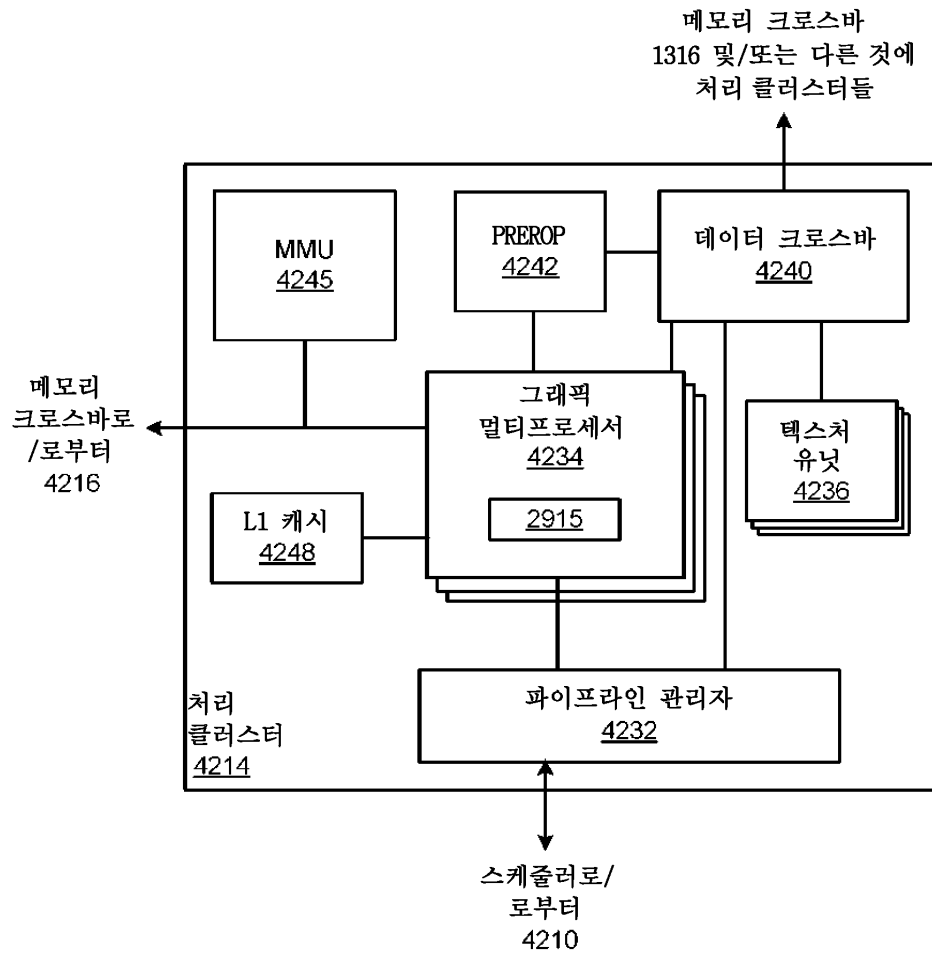
도면42a



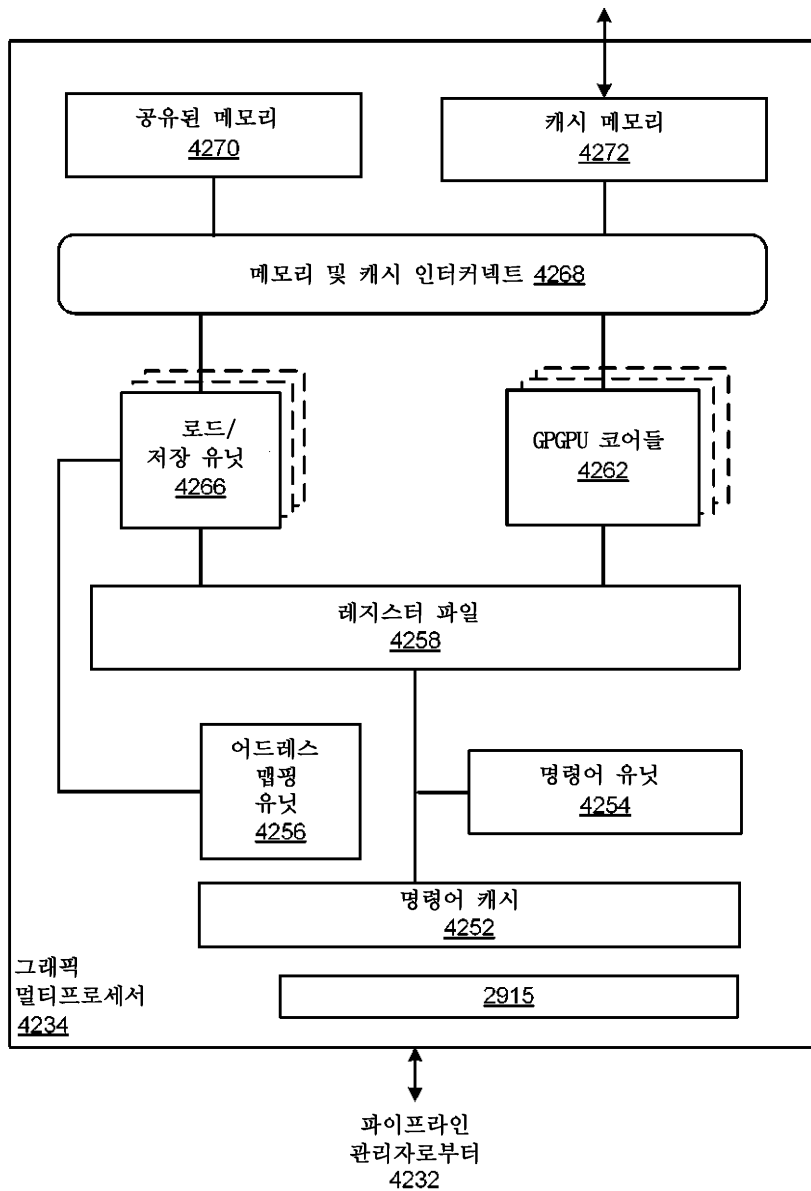
도면42b



도면42c

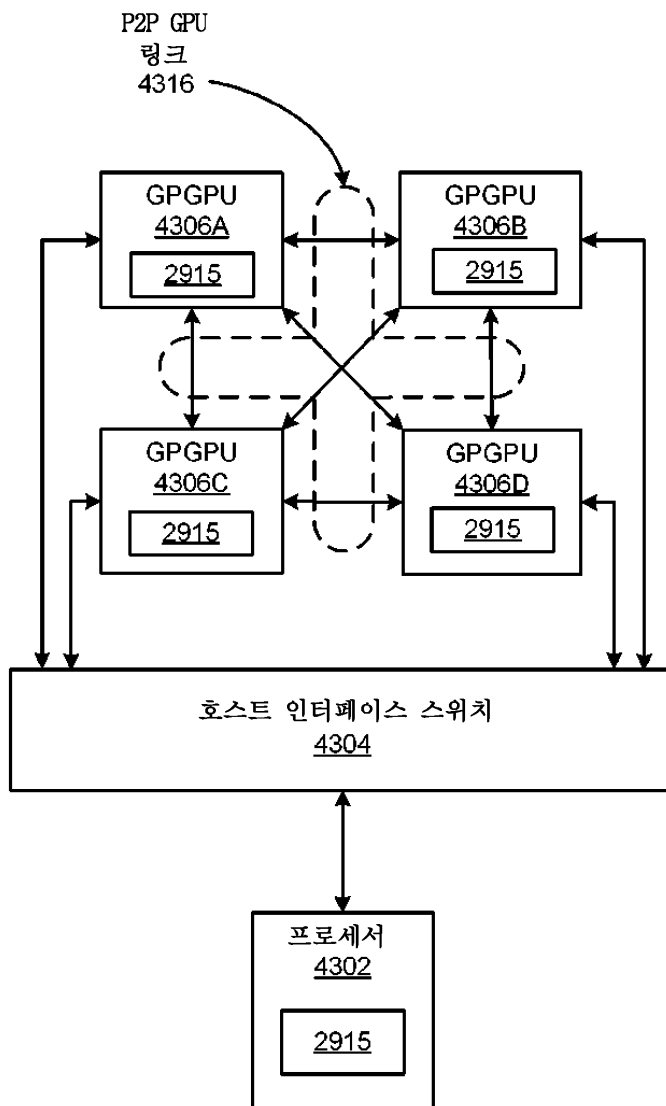


도면42d

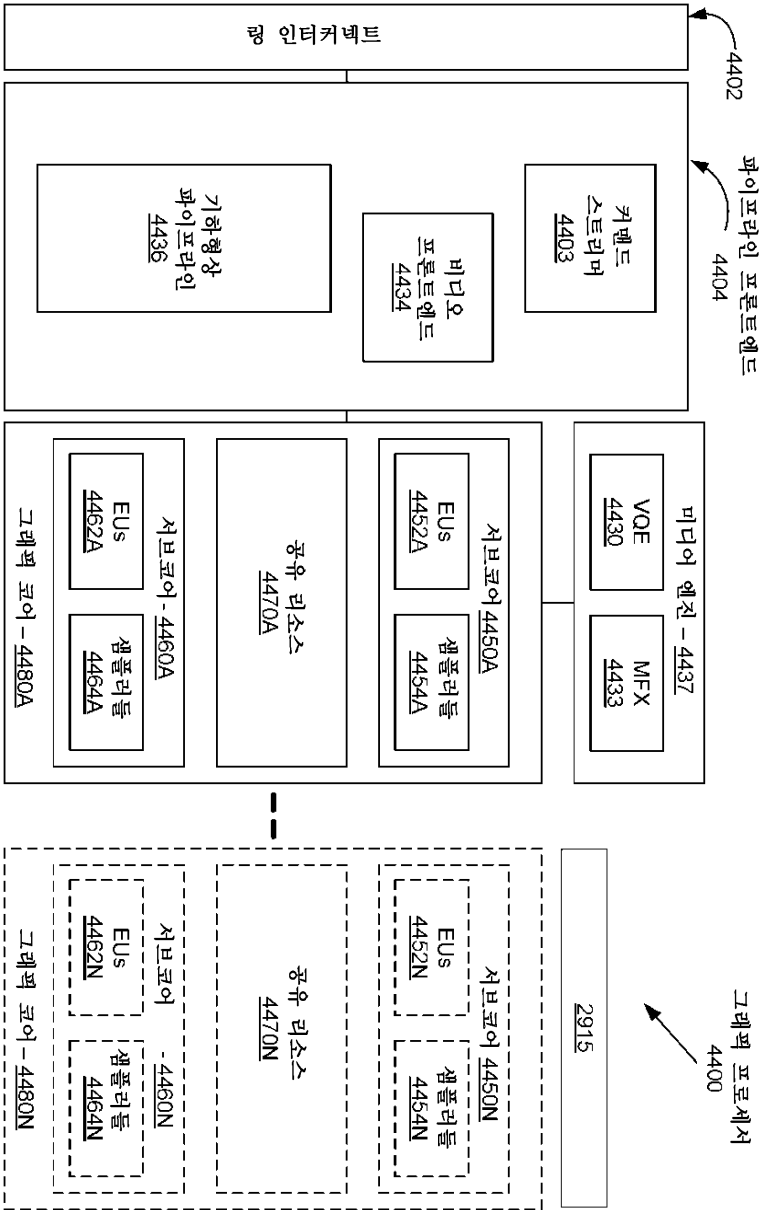


도면43

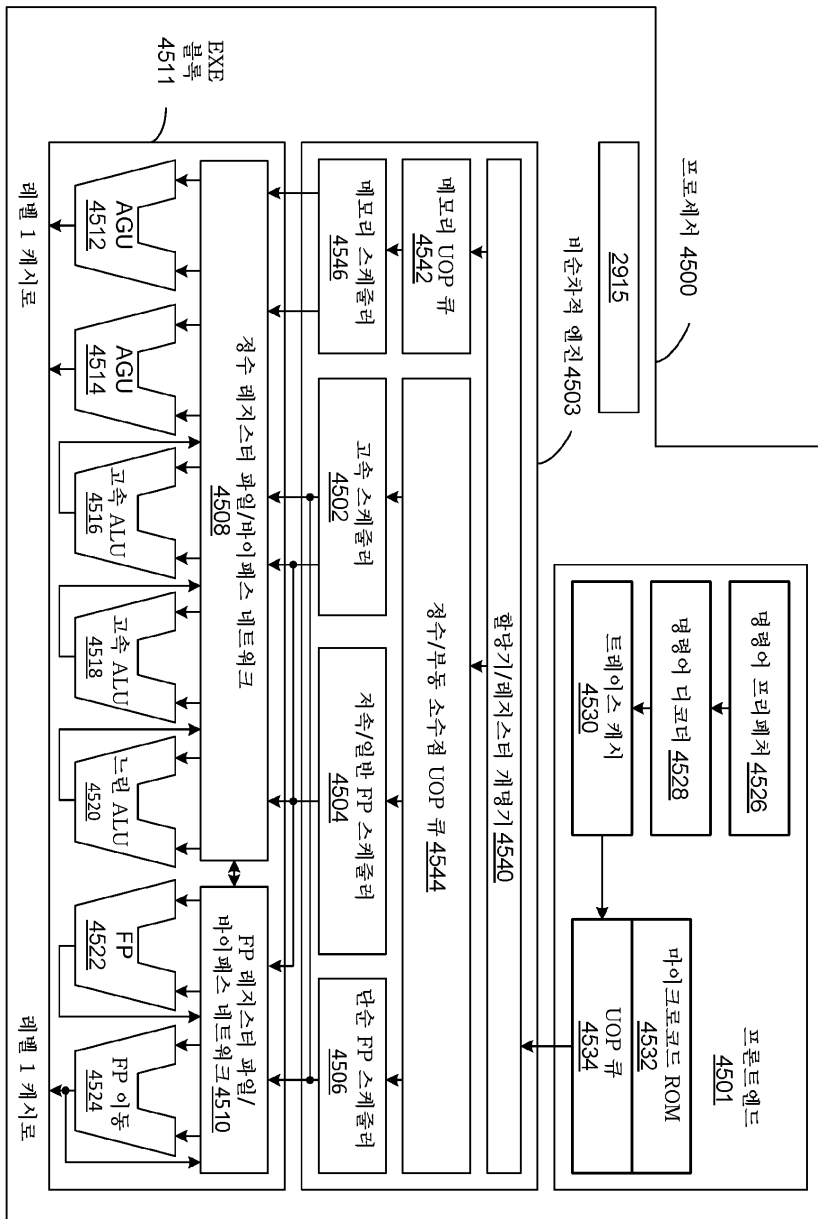
4300



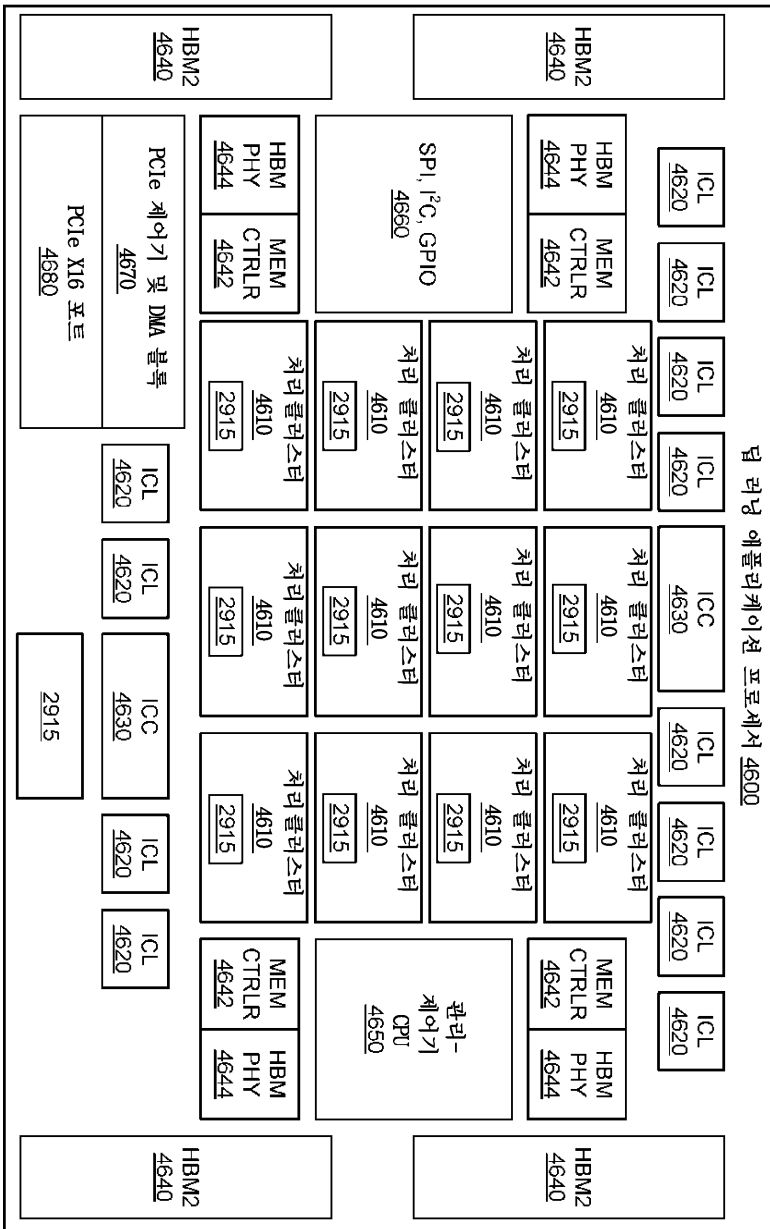
도면44



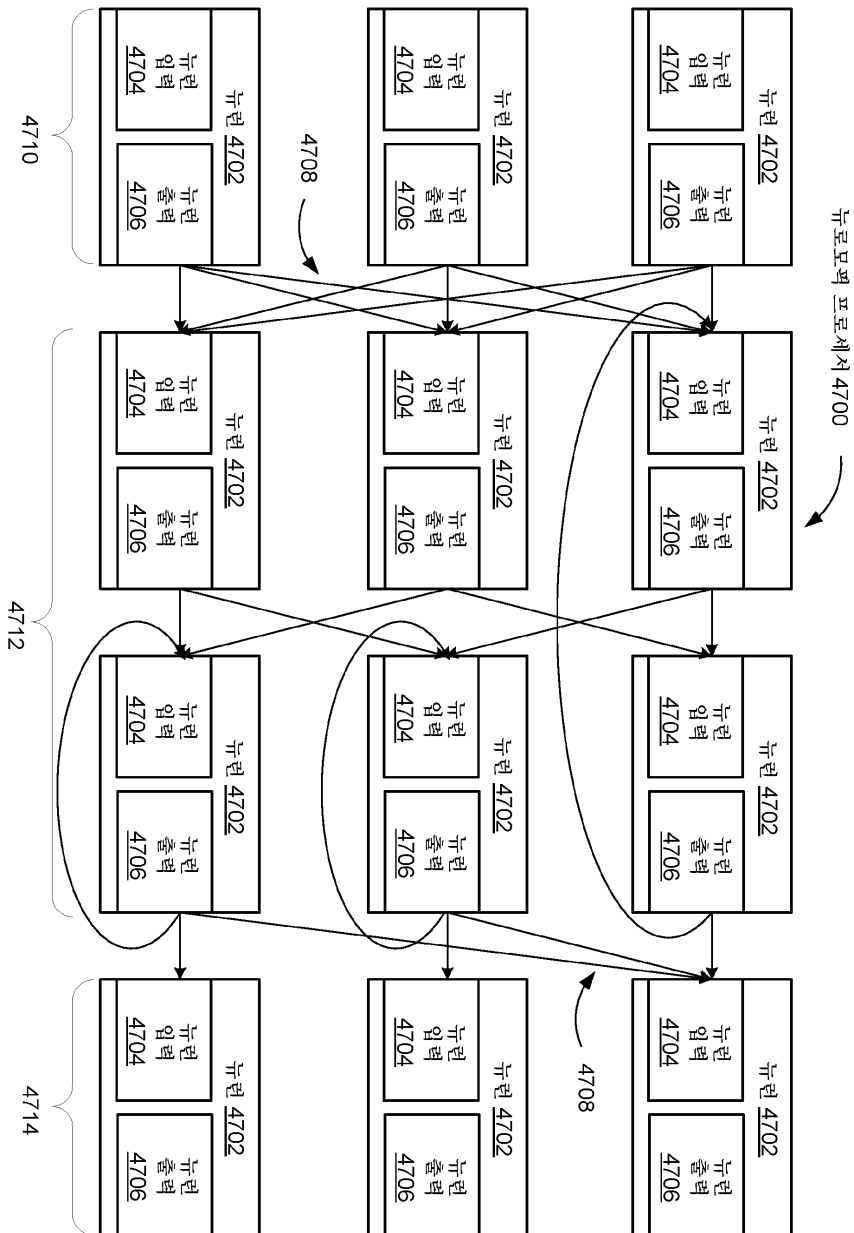
도면45



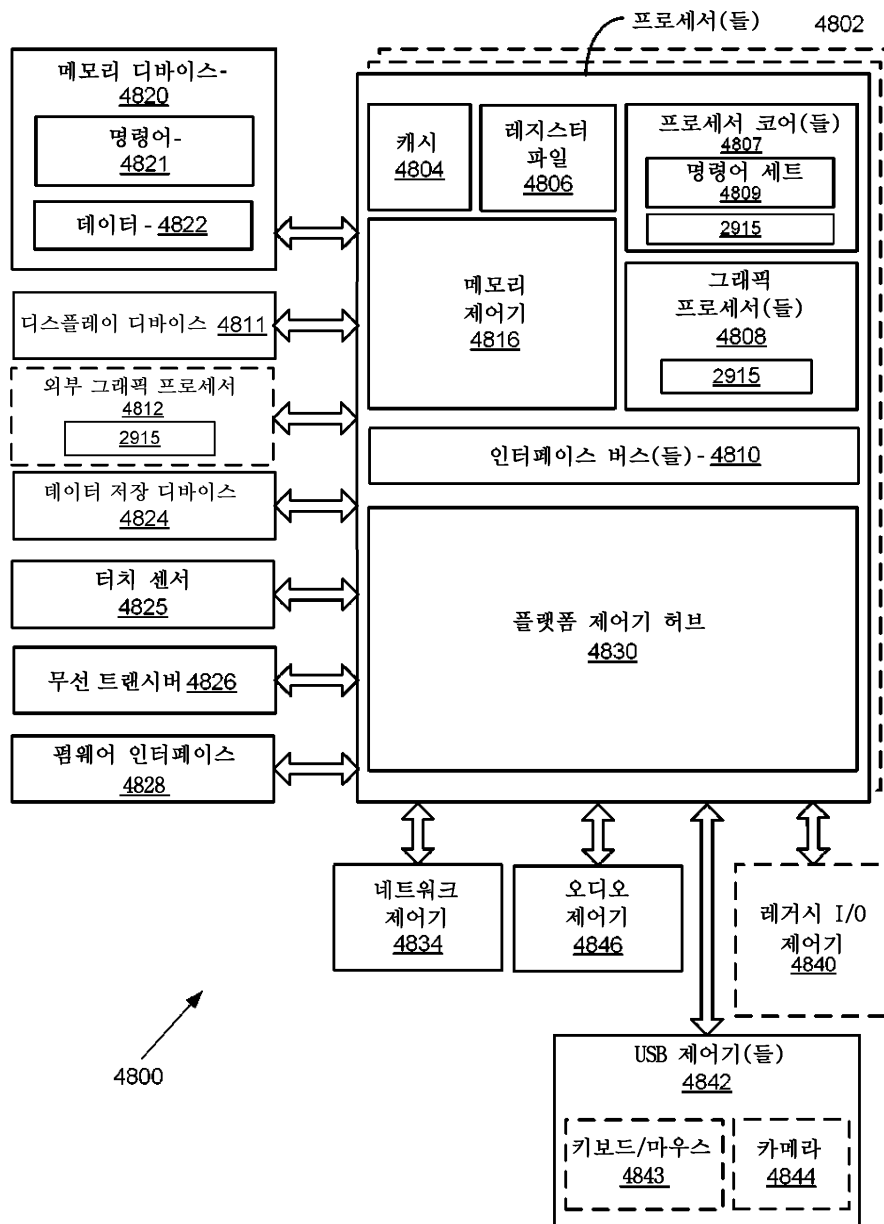
도면46



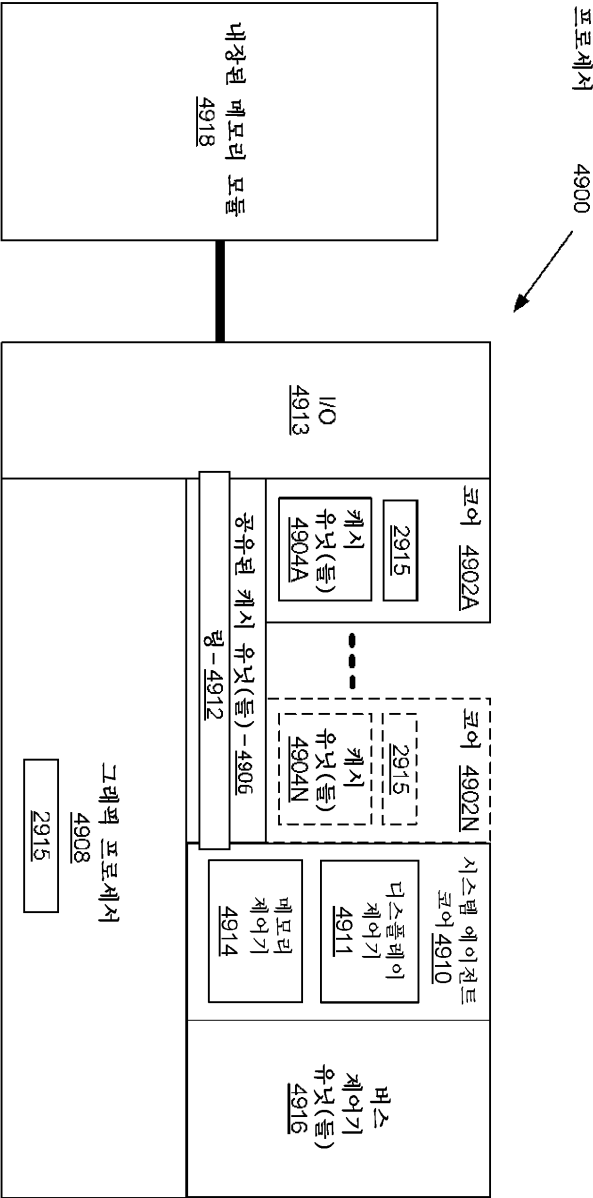
도면47



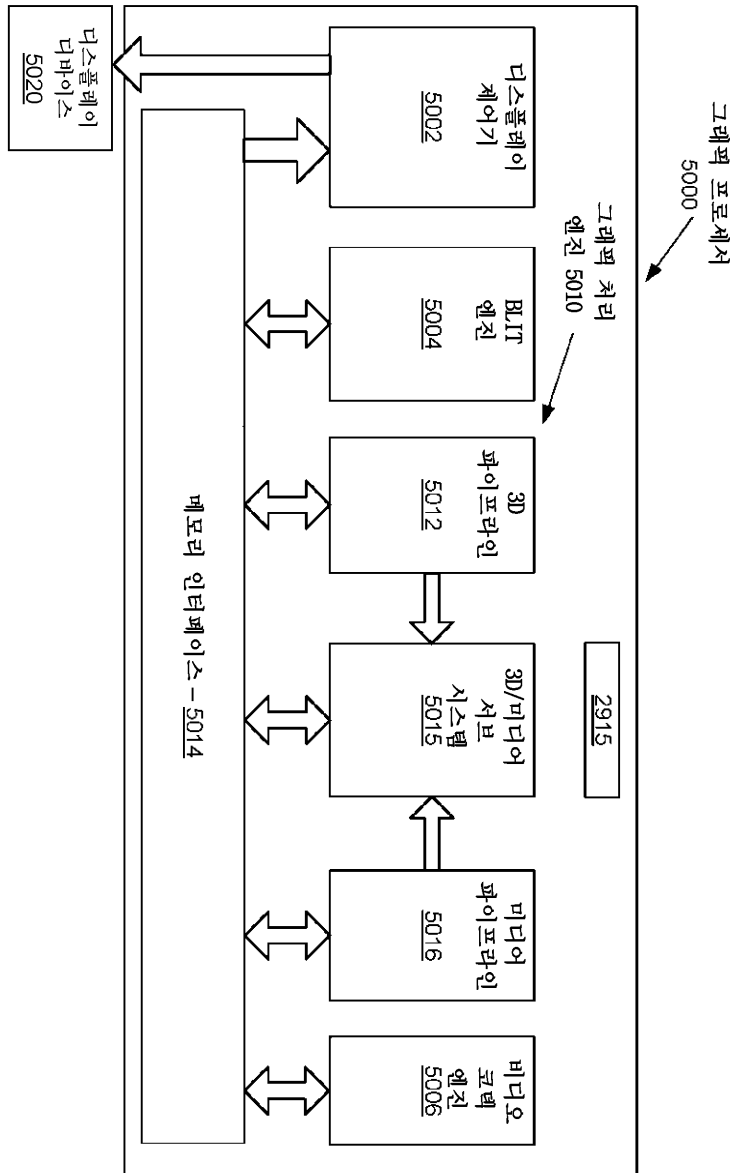
도면48



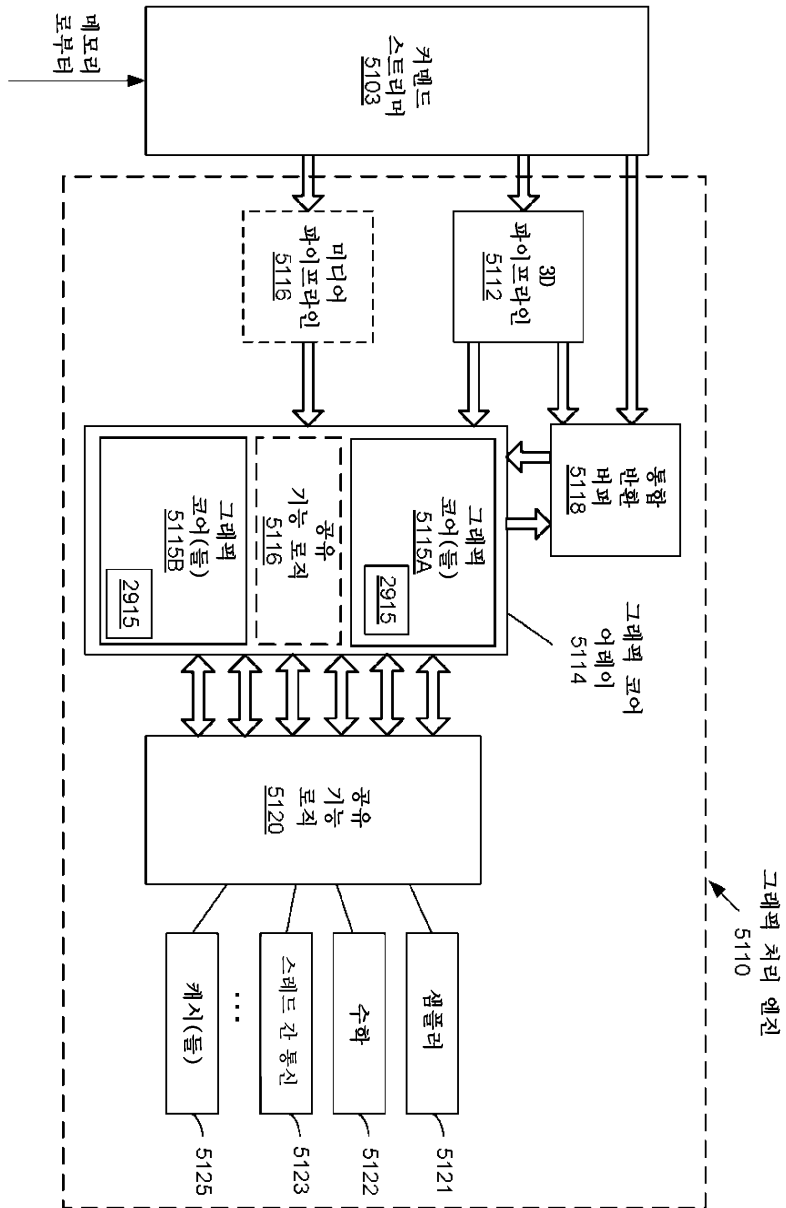
도면49



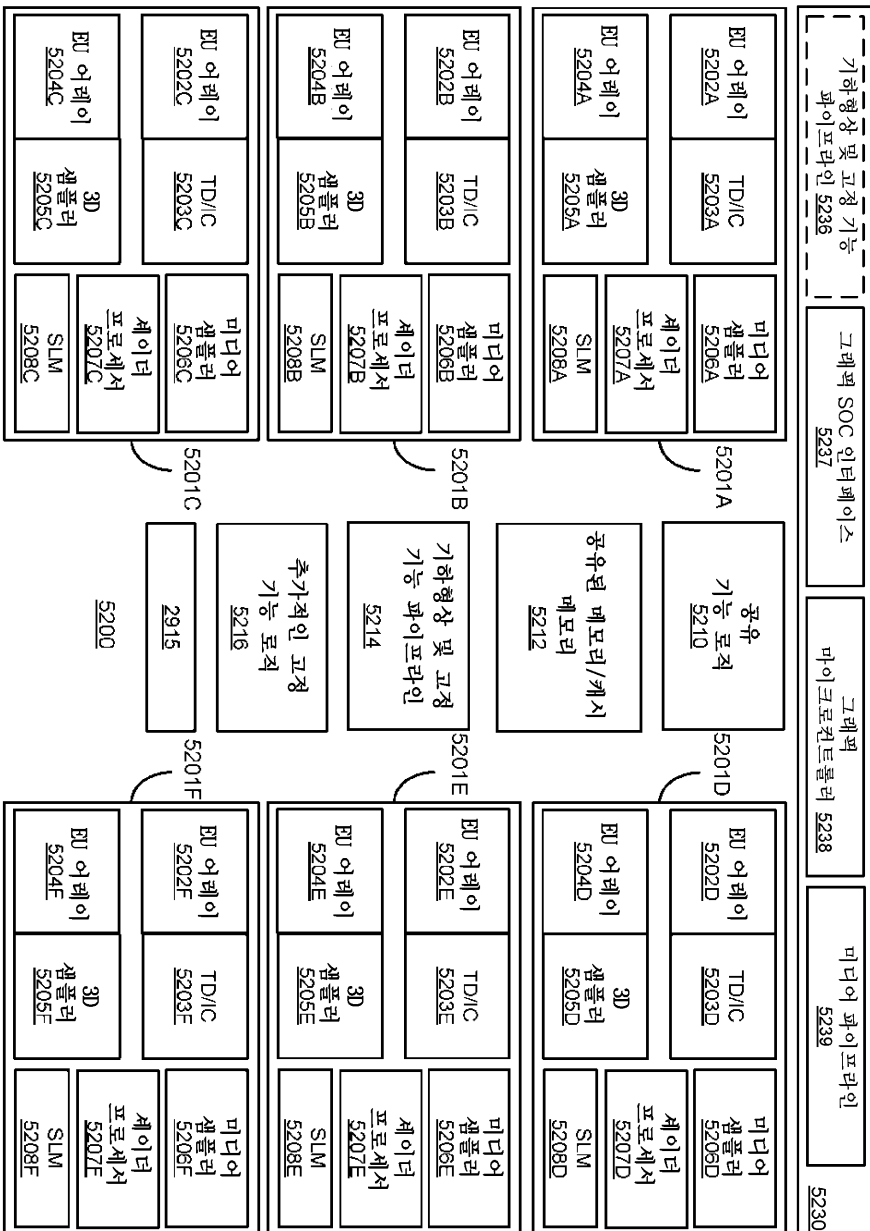
도면50



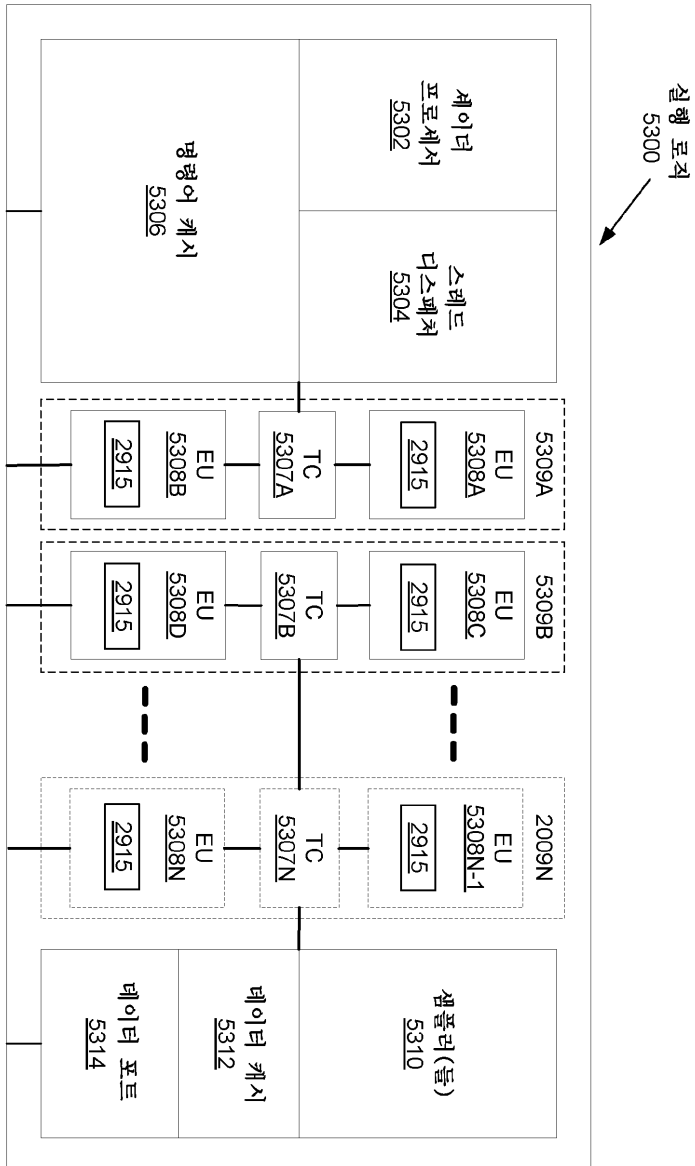
도면51



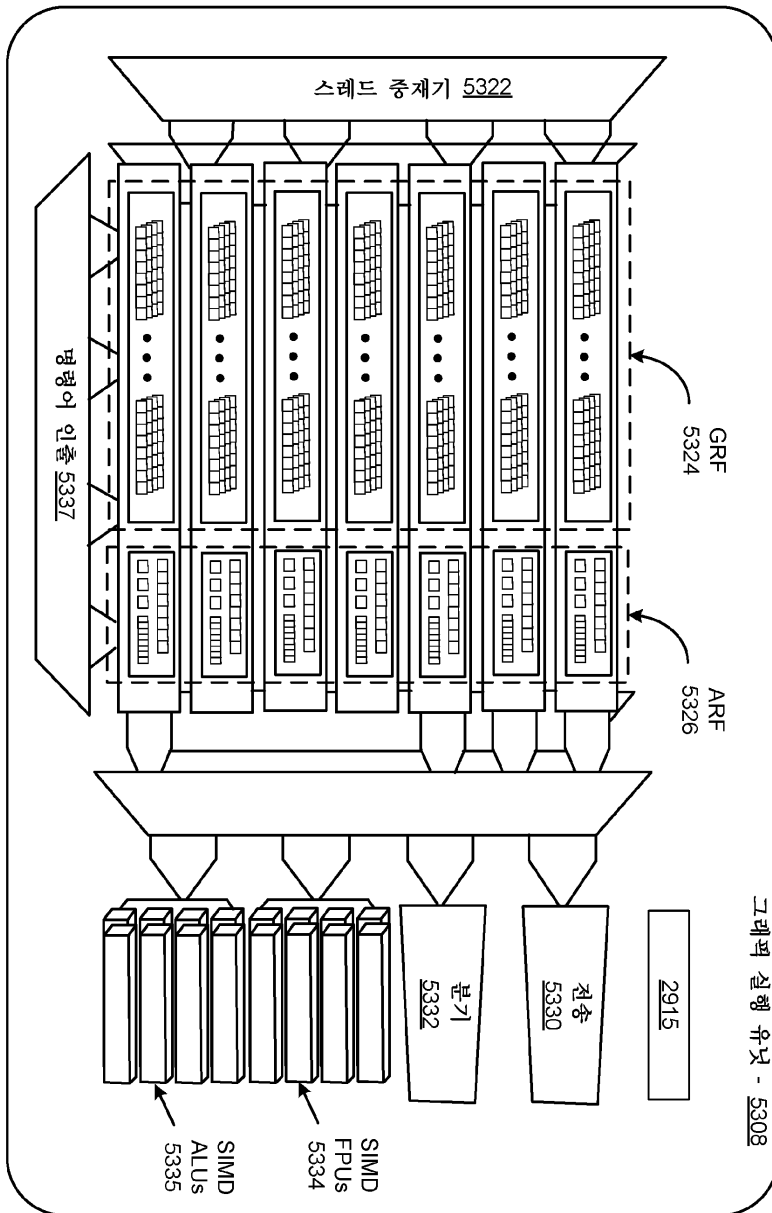
도면52



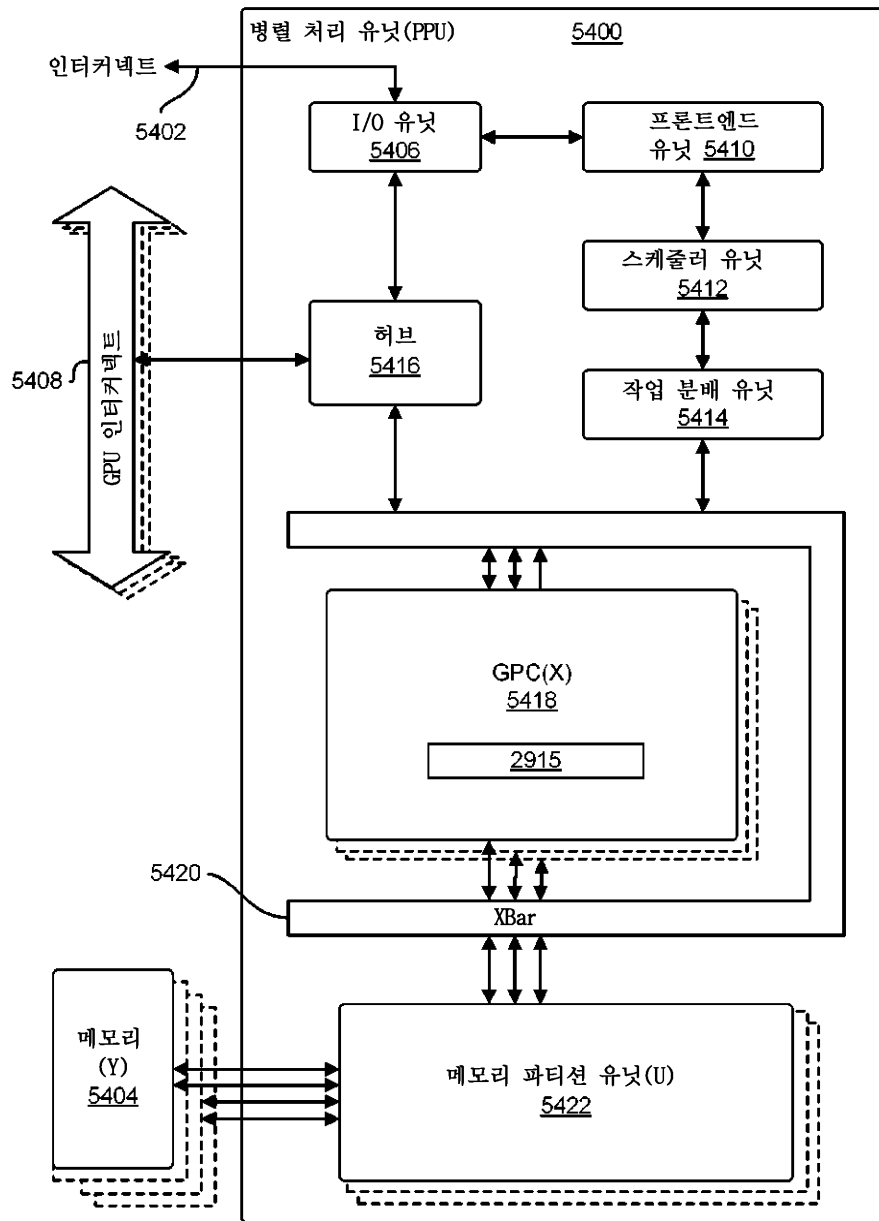
도면53a



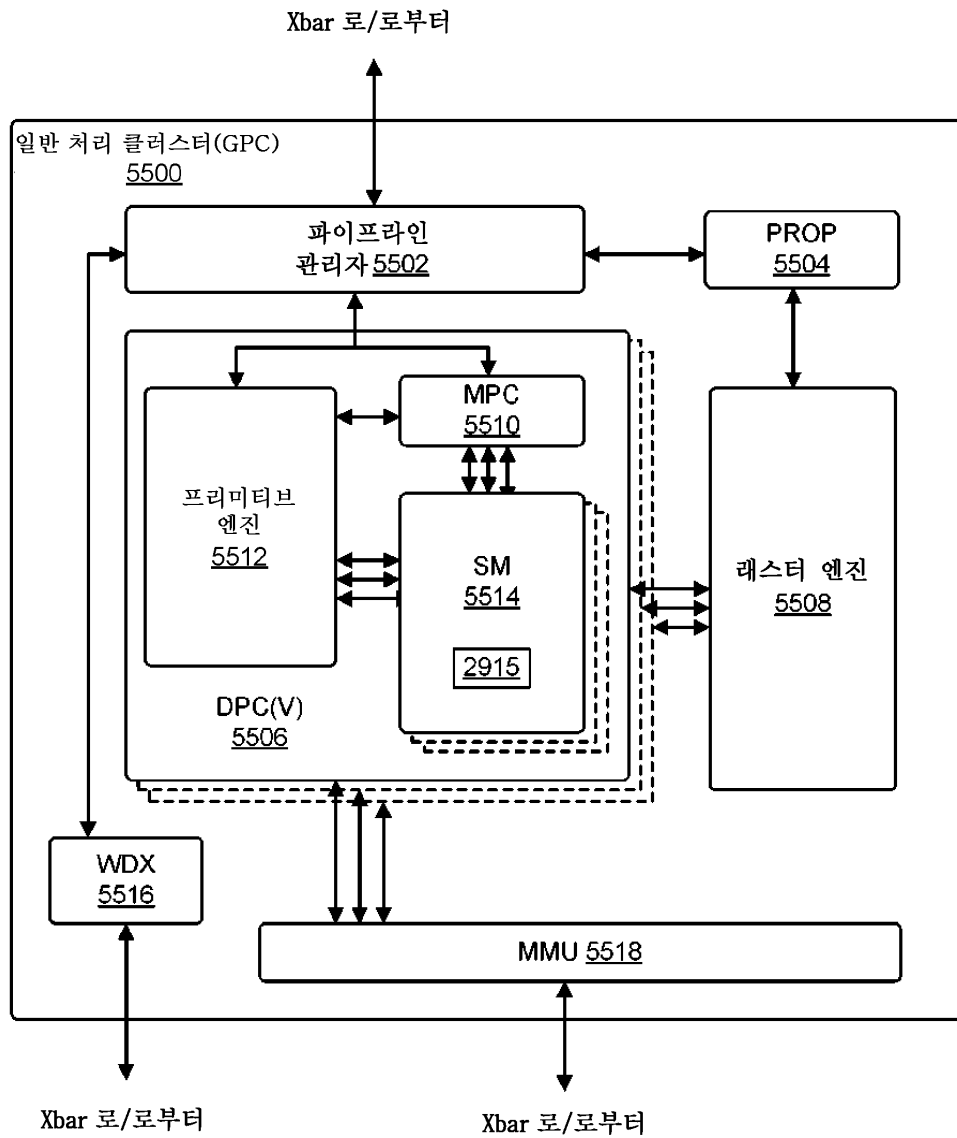
도면53b



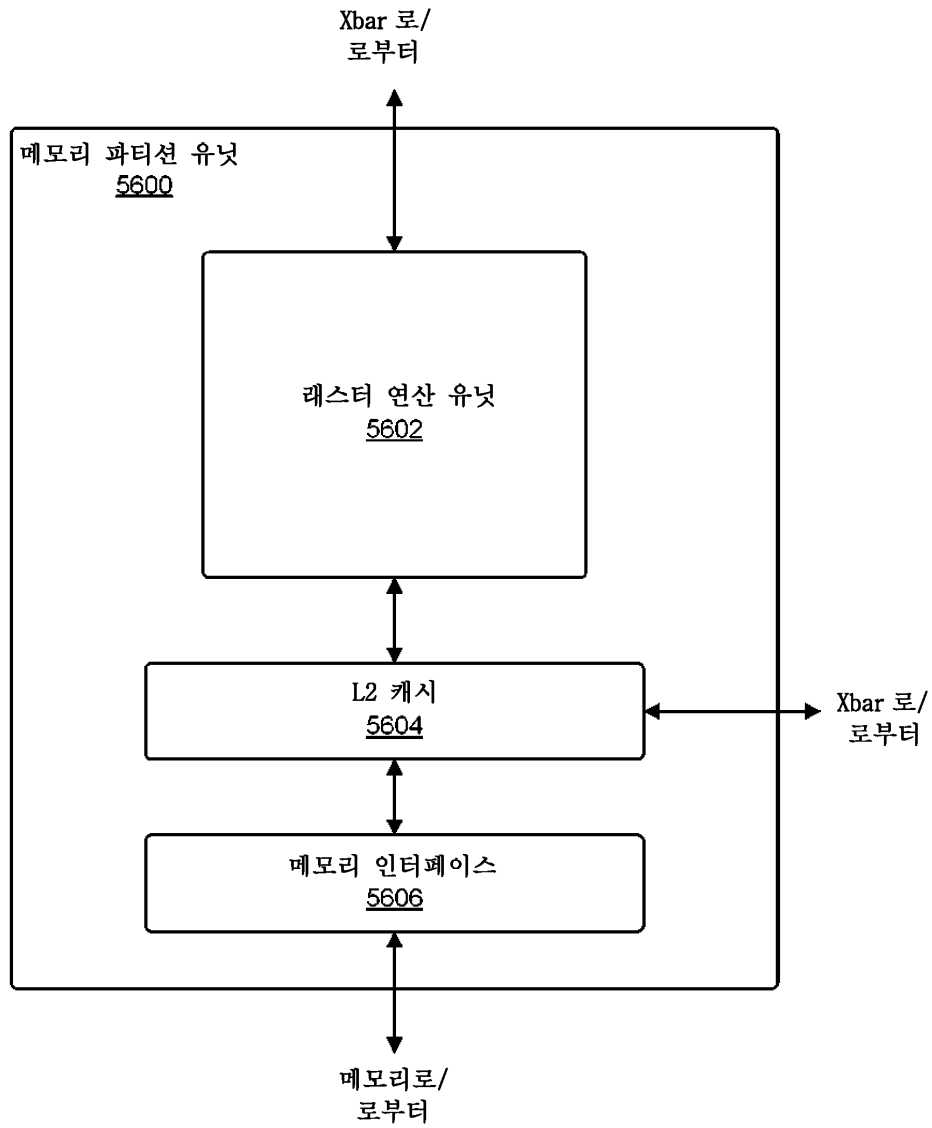
도면54



도면55



도면56



도면57

