(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2020 년 9 월 10 일 (10.09.2020)

WIPO | PCT

(10) 국제공개번호
WO 2020/180014 A2

(51) 국제특허분류:
G05D 1/02 (2020.01)

(21) 국제출원번호: PCT/KR2020/001692

(22) 국제출원일: 2020 년 2 월 6 일 (06.02.2020)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:
10-2019-0025284 2019 년 3 월 5 일 (05.03.2019) KR

(71) 출원인: 네이버랩스 주식회사 (**NAVER LABS CORPORATION**) [KR/KR]; 13638 경기도 성남시 분당구 구미로 8 분당엠타워 4층, Gyeonggi-do (KR).

(72) 발명자: **최진영 (CHOI, Jinyoung)**; 13638 경기도 성남시 분당구 구미로 8 분당엠타워 4층, Gyeonggi-do (KR). **박경식 (PARK, Kay)**; 13638 경기도 성남시 분당구 구미로 8 분당엠타워 4층, Gyeonggi-do (KR). **김민수 (KIM, Minsu)**; 13638 경기도 성남시 분당구 구미로 8 분당엠타워 4층, Gyeonggi-do (KR). **석상옥 (SEOK, Sangok)**; 13638 경기도 성남시 분당구 구미로 8 분당엠타워 4층, Gyeonggi-do (KR). **서준호 (SEO, Joonho)**; 13638 경기도 성남시 분당구 구미로 8 분당엠타워 4층, Gyeonggi-do (KR).

(74) 대리인: **양성보 (YANG, Sungbo)**; 06099 서울시 강남구 선릉로125길 14 삼성빌딩 2층 피앤티특허법률사무소, Seoul (KR).

(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

공개:

— 국제조사보고서 없이 공개하며 보고서 접수 후 이를 별도 공개함 (규칙 48.2(g))

WO 2020/180014 A2

(54) **Title:** METHOD AND SYSTEM FOR TRAINING AUTONOMOUS DRIVING AGENT ON BASIS OF DEEP REINFORCEMENT LEARNING

(54) **발명의 명칭:** 심층 강화 학습에 기반한 자율주행 에이전트의 학습 방법 및 시스템

(57) **Abstract:** Disclosed are a method and a system for training an autonomous driving agent on the basis of deep reinforcement learning (DRL). The agent training method according to one embodiment may comprise a step of training an agent through an actor-critic algorithm in a simulation for DRL. The step of training may be characterized by inputting first information into an actor network that is an evaluation network for determining an agent's behavior in the actor-critical algorithm, and inputting second information into a critique that is a value network for evaluating how helpful the behavior is so as to maximize a predetermined reward. Here, the second information may include the first information and additional information.

(57) **요약서:** 심층 강화 학습에 기반한 자율주행 에이전트의 학습 방법 및 시스템을 개시한다. 일실시예에 따른 에이전트 학습 방법은, 심층 강화 학습(Deep Reinforcement Learning, DRL)을 위한 시뮬레이션상에서 액터-크리틱(actor-critic) 알고리즘을 통해 에이전트를 학습시키는 단계를 포함할 수 있다. 이때, 학습시키는 단계는, 상기 액터-크리틱 알고리즘에서 에이전트의 행동을 결정하는 평가망인 액터 네트워크에 제1 정보를, 상기 행동이 기설정된 보상을 최대화하는데 얼마나 도움이 되는가를 평가하는 가치망인 크리틱에 제2 정보를 입력하는 것을 특징으로 할 수 있다. 여기서, 상기 제2 정보는 상기 제1 정보와 추가 정보를 포함할 수 있다.

명세서

발명의 명칭: 심층 강화 학습에 기반한 자율주행 에이전트의 학습 방법 및 시스템

기술분야

[1] 아래의 설명은 심층 강화 학습에 기반한 자율주행 에이전트의 학습 방법 및 시스템에 관한 것이다.

배경기술

[2] 최근, 점점 더 많은 수의 모바일 로봇들이 생활 공간에 배치되고 있다. 모바일 로봇은 배달, 감시, 안내와 같은 서비스를 제공한다. 이러한 서비스를 제공하기 위해서는 복잡하고 혼잡한 환경에서 안전한 자율주행이 필수적이다. 예를 들어, 한국등록특허 제10-1539270호는 충돌회피 및 자율주행을 위한 센서융합 기반 하이브리드 반응 경로 계획 방법을 개시하고 있다.

[3] 대부분의 모바일 로봇 자율주행 방법은 글로벌 플래너 및 로컬 플래너/컨트롤 정책으로 구성된다. 글로벌 플래너는 전체 환경의 글로벌 구조를 사용하여 궤적 또는 경유지를 생성한다. 그리고 나서, 로컬 플래너나 컨트롤 정책은 보행자들과 같은 예기치 않은, 역동적인 장애물과의 충돌을 피하면서 글로벌 플랜을 따른다.

[4] 로컬 플래너(또는 컨트롤 정책)의 경우 인공 포텐셜 필드, 동적 윈도우 접근과 같은 접근방식이 널리 사용된다. 그러나 이러한 규칙 기반 알고리즘의 대부분은 국소 최소치(local minima)에 고착되거나, 정확한 지도에 대한 과도한 의존성, 그리고 다양한 환경에서의 일반화 결여 등과 같은 문제를 겪는 것으로 알려져 있다.

[5] 이러한 문제를 극복하기 위해, 심층 강화 학습(Deep Reinforcement Learning, DRL) 기반의 컨트롤 접근방식들이 제안되었다. 이러한 접근법은 환경과 상호작용을 통해 센서 입력을 로봇 속도에 직접 매핑하는 최적의 파라미터를 학습할 수 있다. 이러한 심층 강화 학습 접근방식이 유망한 결과를 보여주었지만, 기존 방법에서는 오직 통계적이고 시뮬레이션된 환경만 고려하거나 넓은 시야(Field Of View, FOV)를 필요로 하기 때문에 비싼 라이다 장치를 사용해야 하는 문제점이 있다.

발명의 상세한 설명

기술적 과제

[6] 심층 강화 학습(Deep Reinforcement Learning, DRL)을 위한 시뮬레이션상에서, 액터-크리틱 알고리즘의 정책망과 가치망 중 가치망에 실세계에서 얻기 힘들지만 학습에 도움이 되는 정보를 시뮬레이션의 상태에서 직접 추출해 제공함으로써, 학습 시 사용되는 가치망에서는 에이전트의 행동의 가치에 대한 더 정확한 평가를 내릴 수 있도록 하여 정책망의 성능을 향상시킬 수 있는 에이전트 학습 방법 및 시스템을 제공한다.

[7] LSTM(Long-Short Term Memory)과 같은 순환 신경망(Recurrent Neural Network)의 메모리를 활용하여 에이전트가 현재 시야 밖의 환경에 대한 정보를 순환 신경망에 저장된 이전의 센서 값을 통해 획득할 수 있도록 함으로써, 제한된 시야를 갖는 에이전트도 보다 효과적으로 자율주행이 가능하도록 할 수 있는 에이전트 학습 방법 및 시스템을 제공한다.

과제 해결 수단

[8] 적어도 하나의 프로세서를 포함하는 컴퓨터 장치의 에이전트 학습 방법에 있어서, 상기 적어도 하나의 프로세서에 의해, 심층 강화 학습(Deep Reinforcement Learning, DRL)을 위한 시뮬레이션상에서 액터-크리틱(actor-critic) 알고리즘을 통해 에이전트를 학습시키는 단계를 포함하고, 상기 학습시키는 단계는, 상기 액터-크리틱 알고리즘에서 에이전트의 행동을 결정하는 평가망인 액터 네트워크에 제1 정보를, 상기 행동이 기설정된 보상을 최대화하는데 얼마나 도움이 되는가를 평가하는 가치망인 크리틱에 제2 정보를 입력하고, 상기 제2 정보는 상기 제1 정보와 추가 정보를 포함하는 것을 특징으로 하는 에이전트 학습 방법을 제공한다.

[9] 컴퓨터 장치와 결합되어 상기 방법을 컴퓨터 장치에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램을 제공한다.

[10] 상기 방법을 컴퓨터 장치에 실행시키기 위한 프로그램이 기록되어 있는 컴퓨터 판독 가능한 기록매체를 제공한다.

[11] 상기 방법을 통해 학습된 에이전트가 탑재된 모바일 로봇 플랫폼을 제공한다.

[12] 컴퓨터에서 판독 가능한 명령을 실행하도록 구현되는 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서에 의해, 심층 강화 학습(Deep Reinforcement Learning, DRL)을 위한 시뮬레이션상에서 액터-크리틱(actor-critic) 알고리즘을 통해 에이전트를 학습시키고, 상기 에이전트를 학습시키기 위해, 상기 액터-크리틱 알고리즘에서 에이전트의 행동을 결정하는 평가망인 액터 네트워크에 제1 정보를, 상기 행동이 기설정된 보상을 최대화하는데 얼마나 도움이 되는가를 평가하는 가치망인 크리틱에 제2 정보를 입력하고, 상기 제2 정보는 상기 제1 정보와 추가 정보를 포함하는 것을 특징으로 하는 컴퓨터 장치를 제공한다.

발명의 효과

[13] 심층 강화 학습(Deep Reinforcement Learning, DRL)을 위한 시뮬레이션상에서, 액터-크리틱 알고리즘의 정책망과 가치망 중 가치망에 실세계에서 얻기 힘들지만 학습에 도움이 되는 정보를 시뮬레이션의 상태에서 직접 추출해 제공함으로써, 학습 시 사용되는 가치망에서는 에이전트의 행동의 가치에 대한 더 정확한 평가를 내릴 수 있도록 하여 정책망의 성능을 향상시킬 수 있다.

[14] LSTM(Long-Short Term Memory)과 같은 순환 신경망(Recurrent Neural Network)의 메모리를 활용하여 에이전트가 현재 시야 밖의 환경에 대한 정보를

순환 신경망에 저장된 이전의 센서 값을 통해 획득할 수 있도록 함으로써, 제한된 시야를 갖는 에이전트도 보다 효과적으로 자율주행이 가능하도록 할 수 있다.

도면의 간단한 설명

[15] 도 1은 본 발명의 일실시예에 따른 모바일 로봇 플랫폼의 예를 도시한 도면이다.

[16] 도 2는 본 발명의 일실시예에 따른 LSTM-LMC 아키텍처의 예를 도시한 도면이다.

[17] 도 3은 본 발명의 비교예에 따른 CNN 기반 메모리리스 모델의 예를 도시한 도면이다.

[18] 도 4는 본 발명의 일실시예에 따른 SUNCG 2D 시뮬레이터의 예를 나타내고 있다.

[19] 도 5는 본 발명의 일실시예에 있어서, 분석 시나리오들의 예를 도시한 도면이다.

[20] 도 6은 본 발명의 일실시예에 있어서, 컴퓨터 장치의 예를 도시한 블록도이다.

[21] 도 7은 본 발명의 일실시예에 따른 에이전트 학습 방법의 예를 도시한 흐름도이다.

발명의 실시를 위한 최선의 형태

[22] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[23] 모바일 로봇은 인간에게 서비스를 제공하기 위해 복잡하고 붐비는 환경에서 자유롭게 자율주행할 수 있어야 한다. 이러한 자율주행 능력을 위해 심층 강화 학습(Deep Reinforcement Learning, DRL) 기반 방식이 점점 주목받고 있다. 그러나 기존의 DRL 방식은 넓은 시야(Field Of View, FOV)가 필요하므로 비싼 라이다(lidar) 장치를 사용해야 한다. 본 명세서에서는 FOV가 제한된 저렴한 뎁스(depth) 카메라로 고가의 라이다 장치를 대체할 가능성에 대해 검토한다. 첫번째로 본 명세서에서는 DRL 에이전트에서 제한된 시야의 영향을 분석한다. 두번째로 FOV가 제한된 복잡한 환경에서 효율적인 자율주행을 학습하는 새로운 DRL 방법인 로컬맵 크리틱(Local-Map Critic)을 가진 LSTM(Long-Short Term Memory) 에이전트(이하, 'LSTM-LMC')를 제안한다. 마지막으로, 본 명세서에서는 다이내믹 무작위화(dynamics randomization) 방법을 도입하여 현실세계에서 DRL 에이전트의 견고성을 개선한다. 본 명세서에서는 FOV가 제한된 방법이 메모리가 한정되지만 FOV가 넓은 방법을 능가할 수 있다는 것을 보이며, 주변 환경과 다른 에이전트의 다이내믹을 암묵적으로 모델링하는 것을 학습한다는 것을 경험적으로 증명한다. 또한, 본 명세서에서는 하나의 뎁스 카메라를 가진 로봇이 본 발명의 실시예들에 따른 방법을 사용하여 복잡한 실세계를 자율주행할 수 있다는 것을 보여준다. 도 1은 본 발명의 일실시예에 따른 모바일 로봇 플랫폼의 예를 도시한 도면이다. 본 실시예에 따른 모바일

로봇 플랫폼(100)은 90° FOV를 탑재한 인텔 리얼센스(Intel Realsense) D435 뎁스 카메라 하나와 NVIDIA Jetson TX2 하나를 프로세서로 탑재한 예를 나타내고 있다.

[24]

[25] 1. 관련 연구

[26] A. 모바일 로봇 자율주행을 위한 DRL 방법

[27] 모바일 로봇 자율주행에 대한 종래의 접근법들은 인간공학적으로 설계된 하이퍼 파라미터와 규칙에 의존하기 때문에, 하이퍼 파라미터 또는 국소 최소치(local minima)에 대한 민감도와 같은 문제로 인해 복잡하고 다이내믹한 환경에서 종종 실패한다.

[28] DRL 기반 접근법이 이러한 문제를 해결하기 위해 광범위하게 연구되고 있으며, 이러한 DRL 접근방식에서 에이전트는, 환경과의 상호작용을 통해 수집된 데이터로부터 센서 입력을 로봇 속도에 직접 매핑하는 방법을 학습할 수 있다. 최근, 일부 종래기술에서는 RGB-D 이미지를 사용하여 복잡한 실내 환경을 자율주행할 수 있는 DRL 에이전트를 제안했다. 이러한 종래기술은 시뮬레이션 실험에서는 주목할 만한 결과를 보였음에도 불구하고, 다양한 환경에서 RGB-D 장면들 간의 큰 차이와 동적 장애물을 피할 수 있는 능력 부족으로 인해 실세계에 배치하기가 어렵다. 다른 종래기술들은 좀 더 현실적인 해결책을 제시했다. 사회적 인식 충돌 회피 방법을 제안한 종래기술은 실세계에서 강인한 성능을 보였음에도 불구하고, 다른 에이전트(또는 보행자)의 위치 및 속도에 대한 명확한 측정이 요구된다. 라이다의 원자료(raw lidar data)를 사용하는 종래기술의 DRL 에이전트는 확률론적 로드맵과 DRL을 결합하여, 복잡한 환경 전반에 걸쳐 장거리 자율주행을 가능케 하였으나, 정적인 장애물만을 고려한 관계로 복잡한 실제 환경에서는 사용이 어려웠다. 한편, 붐비는 환경에서 자율주행하는 방법을 학습할 수 있는 DRL 에이전트를 제안한 종래기술에서는 에이전트들을 실세계에 성공적으로 배치할 수 있었지만, 넓은 FOV(180° 내지 220°)를 유지하기 위해 고가의 라이다 장비를 요구한다.

[29] 본 발명의 실시예들에서는 고가의 라이다 장치 대신 FOV가 제한된 저가의 뎁스 카메라를 사용할 수 있다.

[30] B. 멀티-에이전트 DRL

[31] 최근 멀티-에이전트 설정에 대한 DRL 방법이 주목받고 있다. 복수의 에이전트 간의 암묵적인 통신 프로토콜을 학습할 수 있는 신경망 아키텍처는 에이전트가 통신이나 중앙집중식 컨트롤러가 없는 에이전트보다 더 나은 성능을 보였음에도 불구하고, 인간 로봇 상호작용 시나리오에서는 불가능한 다이렉트 메시징을 서로 필요로 한다. 다른 에이전트들의 정보를 크리틱에게만 제공하는 MADDPG(Multi-Agent Deep Deterministic Policy Gradient) 방법의 알고리즘은 테스트 시간에 명시적인 메시지 교환 없이 협력 행동이 나타날 수 있다는 것을 보여줌으로써, 붐비는 환경에서의 자율주행과 같은 인간-로봇 상호작용

상황에서 사용될 수 있는 가능성을 열어 주었다.

[32] 본 발명의 일실시예에서는 크리틱에게 다른 에이전트의 정보뿐만 아니라 환경에 대한 정보를 더 제공함으로써 MADDPG의 접근 방식을 확장한다.

[33] C. 다이내믹 무작위화를 이용한 실세계에서 DRL 에이전트의 직접 배치

[34] 게임 도메인에서 DRL 방법이 큰 성공을 하였지만, 실세계의 로봇 작업에 DRL 에이전트를 배치하는 것은 실세계와 시뮬레이터의 차이 때문에 더 어려운 것으로 여겨진다. 이 차이는 DRL 에이전트들이 시뮬레이터에서 훈련을 받은 후 정밀한 튜닝 없이 배치될 때에, 에이전트들의 성능을 크게 저하시킨다. 이 문제를 해결하기 위해, 시뮬레이터에서 다이내믹 무작위화가 사용되었다. 이러한 다이내믹 무작위화는 네발 달린 로봇의 운동이나 로봇 팔을 사용한 물체 조작과 같은 실제 로봇 작업에 있어 에이전트의 견고성을 향상시킬 수 있다. 본 발명의 일실시예에서는 시뮬레이션에서의 센서 노이즈, 휠 드리프트 및 컨트롤 주파수를 무작위화하여, 모바일 로봇 자율주행 작업에서 다이내믹 무작위화가 미치는 영향을 조사하였다.

[35]

[36] 2. 접근

[37] 이하에서는 심층 강화 학습 프레임워크에 대해 간략하게 설명한 후, 본 발명의 실시예들에 따른 LSTM-LMC 아키텍처를 설명한다. 그 후 본 발명의 일실시예에 따른 훈련 환경과 다이내믹 무작위화 기술에 대한 세부사항을 설명한다.

[38] A. 심층 강화 학습

[39] 강화 학습은 일례로, 로봇의 제어 알고리즘을 사람이 직접 만들지 않고 인공지능 에이전트가 시뮬레이션 또는 실세계에서 직접 상호작용하며 개발자가 지정해준 보상(reward)를 최대화 하도록 스스로 로봇의 제어 방법을 학습하는 방법이다. 심층 강화 학습은 심층 신경망(Deep Neural Network, DNN)를 사용하여 강화학습을 하는 모델을 말한다.

[40] 제한된 FOV와 다른 에이전트의 상태에 대한 불확실성으로 인한 부분적인 관찰 가능성(observability)으로 인해, 일실시예에 따른 환경은 POMDP(Partially Observed Markov Decision Process)로 모델링될 수 있다. POMDP는 6개의 튜플들(S, A, P, R, Ω, O)로 구성되며, 여기서 S는 상태 공간(state space), A는 동작 공간(action space), P는 전환 확률(transition probability), R은 보상 함수(reward function), Ω는 관측 공간(observation space), O는 관측 확률(observation probability)이다.

[41] 강화 학습의 목표는 아래 수학식 1의 감소된 리턴 G를 극대화하는 에이전트의 정책 $\pi(a, o)= p(a|o)$을 학습하는 것이다.

[42]

[수식1]

$$G = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}[r(s_t, a_t)]$$

[43] 여기서,

$\gamma \in [0, 1]$

은 미래의 보상에 대한 감소 요인이다.

[44] 최근, 심층 신경망은 강화 학습 에이전트의 정책 파라미터 또는 가치 함수를 학습하기 위해 널리 사용된다. 본 실시예에서는 아래 수학식 2와 같이 나타나는 리턴 G와 함께 확률론적 정책의 엔트로피를 공동적으로 최대화하는 SAC(Soft Action-Critical) 알고리즘을 사용한다.

[45] [수식2]

$$G = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}[r(s_t, a_t) + \alpha H(\pi(\cdot|s_t))]$$

[46] SAC 알고리즘은 하이퍼 파라미터에 대한 견고성, 연속적인 동작 공간에서 샘플 효율 학습(sample efficient learning), 바람직한 탐색 속성을 위해 선택될 수 있다.

[47] B. 문제 설정

[48] 1) 관측 공간 : 에이전트의 관측 o를 위해, 다양한 수평 FOV(90°, 120°, 180°)를 가진 라이다 데이터와 유사한 슬라이스 포인트 클라우드(Sliced Point Clouds)를 사용한다. 우선, 뎁스 이미지에서 포인트 클라우드를 계산하고, 포인트 클라우드를 수평으로 잘라 바닥과 천장을 제거하여 길이를 줄일 수 있다. 그런 다음, 잘린 클라우드 포인트를 5° 간격으로 수직으로 균일하게 자르고, 각 세그먼트에서 가장 가까운 점으로부터의 거리를 선택하여 ($\mathbb{R}^{18}$, $\mathbb{R}^{24}$, $\mathbb{R}^{36}$) 벡터를 만든다. 이 벡터를 이후 '뎁스 스캔(depth scan)'이라고 부를 것이다.

[49] 또한 에이전트의 현재 선형과 각속도로 구성된 $\mathbb{R}^2$ 벡터를 사용한다. 이러한 속도는 [-1,1] 범위로 표준화될 수 있다.

[50] 또한, r_i가 i 번째 경유지(waypoint)까지의 거리이고, θ_i가 극좌표 각도인 [r_1, sin(θ_1), cos(θ_1), r_2, … cos(θ_5)]의 형태에서 다음 5개 경유지의 상대 거리와 각도를

나타내는

$\mathbb{R}$

15 벡터가 사용될 수 있다.

[51] 2) 동작 공간 : 에이전트의 동작 a에 대해, 선속도 및 각속도를 구하기 위해

$\mathbb{R}$

2 벡터를 사용한다. 이 에이전트의 선속도는 [0, 1] m/s 범위이고 각속도는 [-90, 90] °/s 범위 내에 있다. [-1,1] 범위에 있는 표준화된 속도가 신경망의 출력으로 사용될 수 있다.

[52] 3) 보상 함수 : 보상 r은 다음 수학식 3과 같은 다섯 가지 용어로 구성될 수 있다.

[53] [수식3]

$$r = r_{base} + r_{collision} + r_{waypoint} + r_{rotation} + r_{safety}$$

[54] r_{base} = -0.05는 에이전트들이 가장 짧은 경로를 따르도록 하기 위해 모든 타임스텝에서 주어지는 작은 네거티브 기본 보상일 수 있다.

[55] $r_{collision}$ = -20은 에이전트들이 벽이나 다른 에이전트들과 충돌할 때 에이전트들에게 페널티를 주는 충돌 보상일 수 있다.

[56] $r_{waypoint}$ = 3은 에이전트와 다음 경유지 사이의 거리가 1 미터 미만일 때 에이전트에 주어질 수 있다. 최종 경유지(목표)의 경우, 임계값이 0.6 미터로 설정될 수 있다.

[57] $r_{rotation}$은 큰 각속도에 대한 패널티로서 아래 수학식 4와 같이 정의될 수 있다.

[58] [수식4]

$$r_{rotation} = \begin{cases} -0.15 * |w| & \text{if } \pi/4 \leq |w| \\ 0 & \text{else} \end{cases}$$

[59] 여기서 w는 에이전트의 라디안 각속도일 수 있다.

[60] r_{safety}는 에이전트들이 가능한 한 사전에 장애물을 피하도록 하는 작은 페널티이며 다음 수학식 5와 같이 정의될 수 있다.

[61] [수식5]

$$r_{safety} = min_{o_i \in Obs}(-0.15 * (score_x(o_i) + score_y(o_i)))$$

[62] 여기서 Obs는 다른 에이전트를 포함한 환경에서의 모든 장애물의 집합일 수 있다. s $core_x$ 및 $score_y$ 는 다음 수학식 6 및 수학식 7과 같이 정의될 수 있다.

[63] [수식6]

$$score_x(o_i) = \begin{cases} max(0, 1 - d_x/3) & \text{if } 0 \leq d_x \\ max(0, 1 + d_x/0.3) & \text{if } d_x < 0 \end{cases}$$

[64]

[수식7]

$$score_y(o_i) = max(0, 1 - |d_y|)$$

[65] 여기서, d_x와 d_y는 x축과 y축에서 에이전트와 o_i 사이의 상대 변위일 수 있다.

[66] C. LSTM-LMC

[67] FOV가 제한되면, DRL 에이전트에 대한 상당한 부분 관찰 가능성(patial observability)이 생긴다. 부분 관찰 가능성은 정확한 상태-동작 값의 추정을 어렵게 하며, 차선의 의사결정을 초래할 수 있다. 이러한 부분 관찰 가능성을 극복하기 위해, 본 실시예에 따른 LSTM-LMC가 활용될 수 있다. 도 2는 본 발명의 일실시예에 따른 LSTM-LMC 아키텍처의 예를 도시한 도면이다. 콘볼루션 레이어에서 'F'는 필터 사이즈, 'S'는 스트라이드(stride), 'O'는 출력 채널을 의미할 수 있다. 액터(actor) 네트워크, Q 네트워크 및 V 네트워크에 동일한 아키텍처를 사용할 수 있다. 액터 네트워크는 인공지능 에이전트의 행동을 결정하는 평가망을, Q 네트워크와 V 네트워크를 포함하는 크리틱은 해당 행동이 보상(reward)를 최대화하는데 얼마나 도움이 되는가를 평가하는 가치망을 의미할 수 있다. 로컬-맵 특징을 제공하기 위한 로컬-맵 브랜치(local-map branch)는 액터 네트워크에서 사용되지 않았다.

[68] 1) LSTM 에이전트 : 순환 신경망(Recurrent Neural Network)는 시계열 데이터(time-series data)와 같이 시간의 흐름에 따라 변화하는 데이터를 학습하기 위한 딥 러닝 모델로, 기준 시점(t)와 다음 시점(t+1)에 네트워크를 연결하여 구성한 인공 신경망이다. 그러나, 매 시점에 심층 신경망(DNN)이 연결되어 있을 경우, 오래 전의 데이터에 의한 기울기 값이 소실되는 문제(vanishing gradient problem)로 학습이 어려워진다. LSTM 방식의 순환 신경망은 이러한 문제를 해결하기 위한 대표적인 모델이다. 이러한 LSTM을 사용함에 따라 에이전트에 메모리 능력이 주어질 수 있다. 이후 실험에서 분석된 바와 같이, 메모리는 주변 환경의 표현과 움직이는 장애물의 다이내믹을 암묵적으로 구축함으로써 충돌 회피에 중요한 역할을 할 수 있다. LSTM 만으로도 이후 실험에서 FOV가 제한된 에이전트의 성능을 크게 향상시킬 수 있다. 경험 리플레이(experience replay)에서 200-스텝 궤적을 샘플링하여 LSTM(및 LSTM-LMC) 에이전트를 훈련시킬 수 있다. 궤적은 에피소드의 랜덤 포인트에서 샘플링될 수 있으며, LSTM의 상태는 각 궤적의 시작 부분에서 '0'으로 설정될 수 있다.

[69] 2) 로컬-맵 크리틱(Local-Map Critic, LMC) : 다른 에이전트의 동작과 같은 추가 정보를 크리틱에 포함시키면, 멀티-에이전트 DRL의 성능이 향상될 수 있다. 액터가 추가 정보를 요구하지 않고 크리틱은 대개 훈련이 완료된 후에 사용되지 않기 때문에, 비싼 추가 정보 없이 이 접근법으로 훈련된 에이전트를 배치할 수 있다. 단지 다른 에이전트들의 동작 대신에 주변 지역의 2D 로컬맵을 크리틱에게 줌으로써 이 접근법을 확장시킬 수 있다. 로컬맵 M은 에이전트 주변의 10m × 10m 영역을 다룬다. 이는 사이즈(39 × 39 × 4)인 텐서(tensor)로서,

$M_{i,j,k}$의 값은 다음 수학식 8과 같이 정의될 수 있다.

[70] [수식8]

$$M_{i,j,0} = \begin{cases} 1 & \text{if } M_{i,j} \text{ is movable} \\ 0 & \text{if } M_{i,j} \text{ is obstacle} \\ 0.33 & \text{if } M_{i,j} \text{ is other agent} \\ 0.66 & \text{if } M_{i,j} \text{ is self} \end{cases}$$

[71] $M_{i,j}$가 에이전트를 나타내는 경우, $M_{i,j,1:3}$는 표준화된 헤딩, 선속도 및 각속도를 인코딩할 수 있다.

[72] 3) 네트워크 아키텍처 : LSTM-LMC 모델의 네트워크 아키텍처는 앞서 설명한 도 2에 나타나 있다. 먼저 완전히 연결된 레이어를 사용하여 동일한 사이즈의 벡터에 뎁스 스캔과 속도를 투영하고, 이 두 벡터에 성분 내적(elementwise product)을 적용하여 관찰 특징을 얻을 수 있다. 크리틱(Q 네트워크 및 V 네트워크)에서 로컬 맵 텐서가 3개의 콘볼루션 레이어를 통과하고 글로벌 에버리지 풀링을 적용하여 로컬 맵 특징을 구현할 수 있다. 그런 다음 관찰 특징, 로컬 맵 특징 및 경유지의 연결(concatenation)이 LSTM의 입력으로 사용될 수 있다. LSTM 레이어의 출력은 완전히 연결된 레이어를 통과하고, 이어서 정책 출력 레이어 또는 가치 출력 레이어를 통과할 수 있다. 로컬 맵 특징은 액터에서 사용되지 않으며 Q 네트워크의 LSTM에는 추가의 동작 입력을 가질 수 있다. 정책 출력을 위해, 하이퍼볼릭 탄젠트 스쿼싱 함수(tanh squashing function)를 가진 가우시안 정책(Gaussian policy)이 사용될 수 있다.

[73] 또한 비교 실험을 위해 (90°, 120°, 180°)의 FOV를 가진 CNN 기반 메모리리스(memoryless) 모델이 구현되었다. 도 3은 본 발명의 비교예에 따른 CNN 기반 메모리리스 모델의 예를 도시한 도면이다. CNN 모델의 경우, d_{scan}이 뎁스 스캔 벡터의 크기일 때, 뎁스 스캔 벡터($\mathbb{R}^{d_{scan}\times 2}$)의 모양과 매치되도록 속도 벡터($\mathbb{R}$ [2])를 타일링할 수 있다. 그런 다음 이 타일링된 벡터는 뎁스 스캔 벡터($\mathbb{R}^{d_{scan}\times 1}$)와 연결되어 사이즈가 $\mathbb{R}^{d_{scan}\times 3}$ 인 매트릭스가 될 수 있다. 하나의 네트워크 입력 텐서($\mathbb{R}^{d_{scan}\times 3\times 3}$)를 얻기 위해 최근 세 개의 타임스텝에서 이 매트릭스가 쌓일 수 있다. 이 텐서는 3개의 콘볼루션 레이어를 통과하여 관찰 기능을 얻기 위해 편평화될 수

있다. 그런 다음 관찰 기능은 경유지에 연결되고, 완전히 연결된 2개의 레이어를 통과한 다음 출력 레이어를 통과할 수 있다.

[74] D. SUNCG 2D 시뮬레이터 및 다이내믹 무작위화

[75] 1) SUNCG 2D 환경 : 2D 멀티-에이전트 자율주행 시뮬레이터가 본 발명의 일실시예에 따른 실험을 위해 구현되었다. 도 4는 본 발명의 일실시예에 따른 SUNCG 2D 시뮬레이터의 예를 나타내고 있다. 도 4에서 검은 영역은 장애물을 나타내고 있으며, 색을 갖는 서클은 에이전트(로봇)을 상징하며, 색을 갖는 선은 글로벌 플래너의 플랜들이다. 도 4는 빈 지도에서 0.33(오른쪽)의 확률로 에피소드를 시작한 예를 나타내고 있다. SUNCG 데이터셋에서 1,000개의 랜덤 층 플랜들이 추출될 수 있으며, 75개의 지도가 학습 환경으로서 수동으로 선택될 수 있다.

[76] 2) 훈련 시나리오 : 각 훈련 에피소드마다, 데이터셋의 75개 지도 중 무작위 환경이 샘플링될 수 있다. 초기 실험에서는 움직이는 장애물을 피하는 것이 정적인 장애물을 피하는 것보다 더 어렵다는 것이 발견되었다. 따라서, 움직이는 장애물만 있는 작은 빈 지도(도 3의 오른쪽)가 확률 0.33으로 선택되도록 하여 움직이는 장애물을 피하는 능력을 강화할 수 있다. 지도가 선택되면, 최대 20개의 에이전트가 임의의 위치에 배치되고 무작위 목표 위치가 에이전트에 지정될 수 있다. 그 다음, 환경은 (1m × 1m) 셀 격자 형태로 표시될 수 있으며, 각 에이전트에 대한 경유지를 dijkstra 알고리즘을 이용하여 추출할 수 있다. 각 에이전트에서, 이 에피소드는 장애물과 충돌하거나 1,000번의 타임스텝이 지나갈 때 끝이 나도록 설정되었다. 에이전트가 목표에 도달하면, 새로운 무작위 목표와 경유지를 에이전트에 할당하였다.

[77] 3) 다이내믹 무작위화 : 실세계의 다이내믹 및 관찰은 시뮬레이터의 다이내믹 및 관찰과는 다르다. 또한, 실세계의 다이내믹과 관찰은 노이즈가 매우 많다. 이러한 차이와 노이즈는 종종 시뮬레이터에서 훈련된 에이전트가 실제 환경에서 제대로 작동하지 못하게 한다. 이 문제를 해결하기 위해, 학습된 정책의 견고성을 개선하기 위해 시뮬레이터의 관찰과 다이내믹을 무작위화했다.

[78] 모바일 로봇 자율주행 또한 이러한 무작위화 기술의 혜택을 받을 수 있다. 시뮬레이터에는 다음과 같은 무작위화가 적용될 수 있다. 모바일 로봇이 마주칠 수 있는 실세계의 노이즈는 대개 한 에피소드 내에서 일관되지 않기 때문에, 모든 타임스텝의 노이즈를 다시 샘플링할 수 있다.

[79] · 스캔 노이즈 : 실세계 스캔 데이터는 시뮬레이터의 데이터보다 더 노이즈가 많으며, 뎁스 이미지는 라이다 데이터보다 더 노이즈가 많다고 알려져 있다. 따라서 모든 뎁스 스캔 값에 $N(0, 0.1)$을 더한다.

[80] · 속도 무작위화 : 실세계에서 로봇은 휠 드리프트, 모터 제어기 에러, 마찰 등으로 인해 입력과 동일한 속도로 이동하지 않는다. 이에 대처하기 위해, 로봇에 이를 적용하기 전에, 입력속도를 $N(1, 0.1)$과 곱할 수 있다. 또한, 실세계의

모터는 속도를 즉시 변경할 수 없으므로, 타임스텝 t에서의 에이전트의 속도를

$$\overline{v_t} = 0.8v_t + 0.2\overline{v_{t-1}}$$

로 설정할 수 있다. 여기서 v_t는 에이전트로부터의 커맨드를 노이즈와 곱한 값이고,

$\overline{v_t}$

는 로봇에 적용되는 실제 속도이다.

[81] · 타임스케일 무작위화 : 시뮬레이터에서 하나의 타임스텝을 0.15초로 설정할 수 있다. 그러나 실제 하드웨어에서는 정확한 제어 빈도를 기대할 수 없다. 이것은 타임스케일 노이즈가 로봇 자체를 포함한 움직이는 물체의 다이내믹을 잘못 추정하게 하기 때문에, 모바일 로봇 자율주행에 좋지 않을 수 있다. 이를 극복하기 위해 시뮬레이터의 모든 타임스텝에 $N(0, 0.05)$초를 추가할 수 있다.

[82] 실세계의 관찰 및 다이내믹 노이즈가 CNN 에이전트보다 LSTM-LMC 에이전트에 더 큰 영향을 미친다고 가정할 수 있다. 왜냐하면 LSTM-LMC 에이전트는 노이즈에서 발생하는 에러가 누적되도록 더 긴 히스토리를 고려하기 때문이다. 이후 실험 섹션에서 이러한 무작위화의 효과를 자세히 논의할 것이다.

[83]

[84] 3. 실험

[85] 표 I에 열거된 하이퍼 파라미터로 다섯 가지 유형의 에이전트(FOV가 90°, 120°, 180°인 CNN 에이전트, FOV가 90°인 LSTM 에이전트, FOV가 90°인 LSTM-LMC 에이전트)를 훈련시켰다.

[86] [표1]

Hyper parameter	Value
α in Equation 2	0.02
Mini-batch size	256
Trajectory sample length	200
Size of experience replay	5,000,000
Training iteration	3,000,000
Discount factor γ	0.99
Learning rate	3e-4
LSTM unroll	20
Target network update ratio	0.01
Activation functions	LeakyReLU

[87] 각 에이전트는 300만개의 환경 스텝에 맞게 훈련되었다.

[88] A. 성능

[89] 100회의 평가 에피소드에서 훈련된 에이전트가 평가되었다. 평가 세션의 무작위 시드를 수정하여 모든 에이전트가 동일한 출발 위치와 초기 목표 포지션을 가지고 동일한 맵에서 평가되도록 하였다. 평가 결과는 다음 표 2와

같이 요약될 수 있다.

[90] [표2]

Architecture	CNN	CNN	CNN	LSTM	LSTM-LMC (proposed)
FOV	90°	120°	180°	90°	90°
Mean number of passed waypoints	29.73	40.62	51.45	42.96	**53.64**
Mean number of passed goals	1.42	2.22	3.47	3.06	**3.63**
Survival rate until episode ends	13.53%	19.90%	27.67%	**35.80%**	26.90%

[91] 표 2는 다양한 FOV와 아키텍처를 갖는 에이전트들의 성능을 나타내고 있다. 표 2에 나타난 바와 같이, FOV가 감소함에 따라 CNN(메모리리스) 에이전트의 성능은 급격히 하락했다. 반면, FOV가 90°인 LSTM-LMC 에이전트는 통과된 경유지/목표의 개수 측면에서 다른 모든 에이전트, 심지어 FOV가 180°인 CNN 에이전트보다 성능이 우수했다. LSTM 에이전트는 120°인 CNN 에이전트는 능가했지만 180°인 에이전트를 능가하지는 못했다. 하지만, LSTM 에이전트는 에피소드가 끝날 때까지 가장 높은 생존율을 보였다.

[92] B. 분석

[93] 제안하는 방법이 암묵적으로 주변 환경 및 다른 에이전트의 다이내믹에 대한 강력하고 정확한 모델을 구축하기 때문에, 다른 방법보다 우수한 성능을 보여준다고 가설을 세운다. 이하에서는 다음과 같이 통제된 시나리오에서 훈련된 에이전트의 행동을 분석하여 가설을 검증한다.

[94] 도 5는 본 발명의 일실시예에 있어서, 분석 시나리오들의 예를 도시한 도면이다. 도 5에서 상단은 경로상에 예정에 없던 벽이 생겨 경로가 차단된 경우의 시나리오를, 중단은 다른 에이전트와 수직으로 가로질러 이동하는 교차 시나리오를, 하단은 마주보고 오는 다른 에이전트를 피하도록 하는 통과 시나리오에 따른 에이전트들의 움직임의 예를 나타내고 있다. 어두운 선들은 글로벌 플래너로부터의 경로를, 밝은 선들은 에이전트들의 궤도를, 숫자들은 타임스텝들을 각각 나타내고 있다. 본 발명의 일실시예에 따른 LSTM-LMC FOV 90°는 에이전트들 사이의 벽과 대칭 깨짐(symmetry breaking)을 우회하는데 있어 탁월한 성능을 보여준다.

[95] 1) 차단된 경로 시나리오: 제안된 에이전트가 환경 구조를 기억하는지 확인하기 위해 '차단된 경로 시나리오'를 설계했다. 도 5의 상단은 차단된 경로 시나리오에 관한 것으로, 차단된 경로 시나리오에서 글로벌 플래너의 경로는 벽에 의해 차단된다. 벽의 상단이나 하단에 무작위로 배치된 슬릿(slit)이 있어, 원래 경로가 차단된 것을 기억하면서, 에이전트는 어떤 면이 열려 있는지

탐색해야 한다. 50개의 에피소드 동안, 아래 표 3에 나타난 바와 같이, 본 발명의 일실시예에 따른 LSTM-LMC FOV 90°의 에이전트가 가장 높은 성공률을 달성했다.

[96] [표3]

		CNN FOV 90°	CNN FOV 120°	CNN FOV 180°	LSTM FOV 90°	LSTM-LMC FOV 90°
Blocked path	Success rate	0%	16%	82%	78%	92%
Crossing	Success rate	78%	92%	96%	100%	100%
Passing	Success rate	100%	98%	100%	100%	100%

[97] 정성적으로, 본 발명의 일실시예에 따른 LSTM-LMC FOV 90°의 에이전트는 벽의 양쪽을 효율적으로 탐색했고, 차단된 원래 경로가 FOV를 벗어날 때 원래 경로로 돌아가지 않았다. 반면 CNN 에이전트들은 차단된 원래 경로가 그들의 FOV 밖으로 벗어나자마자 그들의 원래 경로로 복귀하려 했다. LSTM 에이전트는 차단된 경로를 통과할 수 있었지만, 최고의 CNN 에이전트(CNN FOV 180°)를 능가하지는 못했다.

[98] 2) 교차 & 통과 시나리오 : 움직이는 장애물의 다이내믹 모델링에 있어서 메모리와 로컬-맵 크리틱의 영향을 확인하기 위해, '교차'(도 5의 중단) 및 '통과'(도 5의 하단) 실험을 실시하였다. 교차 시나리오에서 두 명의 에이전트가 직교 경로(파란색 에이전트는 위나 아래 쪽에 무작위로 위치함)를 추구하고, 에이전트는 동일한 경로를 따르지만 통과 시나리오에서는 반대 방향으로 따른다. 에이전트들은 두 시나리오 모두에서 대칭을 깨기 위해 다른 에이전트의 미래 경로를 모델링해야 한다. 각 에이전트에 대해 각 시나리오를 50회 실시했으며, 결과는 앞서 표 3에 요약되어 있다. LSTM-LMC 및 LSTM 에이전트는 교차 시나리오에서 가장 높은 성공률을 달성했고, 모든 에이전트들이 통과 시나리오에서 성공률 측면에서 잘 수행되었다. 그러나 정성적으로 CNN 에이전트는 도 5의 중단 및 하단에 나타난 바와 같이 종종 양 시나리오(교차 및 통과)에서 대칭을 깨지 못했다. 반대로, 본 발명의 일실시예에 따른 LSTM-LMC FOV 90°에서는 모든 에피소드들에서 안정된 대칭을 보여주었다.

[99] C. 하드웨어 실험

[100] 실세계에서 본 발명의 일실시예에 따른 에이전트 학습 방법의 성능을 확인하기 위해 하드웨어 실험을 진행하였다.

[101] 1) 하드웨어 설정: 도 1을 통해 설명한 바와 같이 4개의 바퀴를 갖는 모바일 로봇 플랫폼을 구축했다. 이러한 모바일 로봇 플랫폼에는 NVIDIA Jetson TX-2가 메인 프로세서로 탑재됐으며, FOV가 90°인 Intel Realsense D435 RGB-D 카메라 1대가 장착됐다. 본 실험에서, 에이프릴태그(Apriltag)와 휠 주행거리 측정기가

로컬리제이션을 위해 사용되었다. 하지만, 로컬리제이션을 위해 GPS, 초광대역(ultrawideband) 또는 비주얼 로컬리제이션(visual localization)과 같은 다른 방법이 사용될 수도 있다. 이러한 모바일 로봇 플랫폼에는 본 발명의 실시예들에 따른 학습 방법에 의해 학습된 에이전트가 탑재될 수 있다.

[102] 2) 시뮬레이터에서 다이내믹 무작위화의 효과 : 실제 실내환경에서 무작위 훈련을 실시하거나 하지 않고, CNN 에이전트와 LSTM-LMC 에이전트들을 배치하였다. FOV가 제한된 에이전트들에게, 환경은 좁은 통로, 많은 커브, 그리고 계단이나 얇은 기둥과 같은 복잡한 장애물을 가지고 있기 때문에 환경은 상당히 어렵다. 또한, 노이즈가 많은 로컬리제이션은 안정적인 자율주행을 방해한다. 각 에이전트에 대해 3가지 실험을 수행했으며 결과는 아래 표 4와 같이 나타났다.

[103] [표4]

	CNN, FOV 90° No randomization			CNN, FOV 90° Randomization			LSTM-LMC, FOV 90° No randomization			LSTM-LMC, FOV 90° Randomization		
Trial	1	2	3	1	2	3	1	2	3	1	2	3
Passed Waypoints	20	51	55 (all)	24	23	55 (all)	55 (all)	44	24	55 (all)	55 (all)	51
Elapsed Time	-	-	63.969	-	-	59.711	64.093	-	-	49.493	50.391	-

[104] 무작위화를 하거나 하지 않은 CNN 에이전트는 둘 다 성능이 좋지 않고, 에피소드 초기 단계에서 장애물과 충돌한다. 또한, CNN 에이전트는 다이내믹 무작위화로부터 의미 있는 장점을 보여주지 못했다. 한편, 예상했던 대로, 다이내믹 무작위화가 없는 LSTM-LMC 에이전트는 실세계의 노이즈로부터 더 많은 어려움을 겪었다. 노이즈는 불안정한 움직임을 보이면서 충돌이나 느린 자율주행을 일으킨다. 다이내믹 무작위화를 사용하는 LSTM-LMC 에이전트는 안정적인 성능을 보인 유일한 에이전트였다.

[105] 3) 혼잡한 실세계 환경에서의 자율주행 : 실제 환경에서 본 발명의 일실시예에 따른 에이전트 학습 방법의 전반적인 성능을 확인하기 위해, 혼잡한 환경에 다이내믹 무작위화를 사용하는 LSTM-LMC 에이전트를 배치했다. 로봇은 7m의 직선 경로를 반복했고, 2명의 참가자가 교차, 통과하거나 로봇의 경로를 방해했다. 이 로봇은 방해를 받는 상황에서도 12개의 연속된 경로(약 84m)를 완주할 수 있었다.

[106] 도 6은 본 발명의 일실시예에 있어서, 컴퓨터 장치의 예를 도시한 블록도이다. 일례로, 본 발명의 실시예들에 따른 에이전트 학습 방법은 도 6을 통해 도시된 컴퓨터 장치(600)에 의해 실행될 수 있다. 이러한 컴퓨터 장치(600)는 도 6에 도시된 바와 같이, 메모리(610), 프로세서(620), 통신 인터페이스(630) 그리고 입출력 인터페이스(640)를 포함할 수 있다. 메모리(610)는 컴퓨터에서 판독 가능한 기록매체로서, RAM(random access memory), ROM(read only memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(permanent mass storage device)를 포함할 수 있다. 여기서 ROM과 디스크 드라이브와 같은 비소멸성

대용량 기록장치는 메모리(610)와는 구분되는 별도의 영구 저장 장치로서 컴퓨터 장치(600)에 포함될 수도 있다. 또한, 메모리(610)에는 운영체제와 적어도 하나의 프로그램 코드가 저장될 수 있다. 이러한 소프트웨어 구성요소들은 메모리(610)와는 별도의 컴퓨터에서 판독 가능한 기록매체로부터 메모리(610)로 로딩될 수 있다. 이러한 별도의 컴퓨터에서 판독 가능한 기록매체는 플로피 드라이브, 디스크, 테이프, DVD/CD-ROM 드라이브, 메모리 카드 등의 컴퓨터에서 판독 가능한 기록매체를 포함할 수 있다. 다른 실시예에서 소프트웨어 구성요소들은 컴퓨터에서 판독 가능한 기록매체가 아닌 통신 인터페이스(630)를 통해 메모리(610)에 로딩될 수도 있다. 예를 들어, 소프트웨어 구성요소들은 네트워크(660)를 통해 수신되는 파일들에 의해 설치되는 컴퓨터 프로그램에 기반하여 컴퓨터 장치(600)의 메모리(610)에 로딩될 수 있다.

[107] 프로세서(620)는 기본적인 산술, 로직 및 입출력 연산을 수행함으로써, 컴퓨터 프로그램의 명령을 처리하도록 구성될 수 있다. 명령은 메모리(610) 또는 통신 인터페이스(630)에 의해 프로세서(620)로 제공될 수 있다. 예를 들어 프로세서(620)는 메모리(610)와 같은 기록 장치에 저장된 프로그램 코드에 따라 수신되는 명령을 실행하도록 구성될 수 있다.

[108] 통신 인터페이스(630)은 네트워크(660)를 통해 컴퓨터 장치(600)가 다른 장치(일례로, 앞서 설명한 저장 장치들)와 서로 통신하기 위한 기능을 제공할 수 있다. 일례로, 컴퓨터 장치(600)의 프로세서(620)가 메모리(610)와 같은 기록 장치에 저장된 프로그램 코드에 따라 생성한 요청이나 명령, 데이터, 파일 등이 통신 인터페이스(630)의 제어에 따라 네트워크(660)를 통해 다른 장치들로 전달될 수 있다. 역으로, 다른 장치로부터의 신호나 명령, 데이터, 파일 등이 네트워크(660)를 거쳐 컴퓨터 장치(600)의 통신 인터페이스(630)를 통해 컴퓨터 장치(600)로 수신될 수 있다. 통신 인터페이스(630)를 통해 수신된 신호나 명령, 데이터 등은 프로세서(620)나 메모리(610)로 전달될 수 있고, 파일 등은 컴퓨터 장치(600)가 더 포함할 수 있는 저장 매체(상술한 영구 저장 장치)로 저장될 수 있다.

[109] 입출력 인터페이스(640)는 입출력 장치(650)와의 인터페이스를 위한 수단일 수 있다. 예를 들어, 입력 장치는 마이크, 키보드 또는 마우스 등의 장치를, 그리고 출력 장치는 디스플레이, 스피커와 같은 장치를 포함할 수 있다. 다른 예로 입출력 인터페이스(640)는 터치스크린과 같이 입력과 출력을 위한 기능이 하나로 통합된 장치와의 인터페이스를 위한 수단일 수도 있다. 입출력 장치(650)는 컴퓨터 장치(600)와 하나의 장치로 구성될 수도 있다.

[110] 또한, 다른 실시예들에서 컴퓨터 장치(600)는 도 6의 구성요소들보다 더 적은 혹은 더 많은 구성요소들을 포함할 수도 있다. 그러나, 대부분의 종래기술적 구성요소들을 명확하게 도시할 필요성은 없다. 예를 들어, 컴퓨터 장치(600)는 상술한 입출력 장치(650) 중 적어도 일부를 포함하도록 구현되거나 또는 트랜시버(transceiver), 데이터베이스 등과 같은 다른 구성요소들을 더 포함할

수도 있다.

[111] 통신 방식은 제한되지 않으며, 네트워크(660)가 포함할 수 있는 통신망(일례로, 이동통신망, 유선 인터넷, 무선 인터넷, 방송망)을 활용하는 통신 방식뿐만 아니라 블루투스(Bluetooth)나 NFC(Near Field Communication)와 같은 근거리 무선 통신 역시 포함될 수 있다. 예를 들어, 네트워크(660)는, PAN(personal area network), LAN(local area network), CAN(campus area network), MAN(metropolitan area network), WAN(wide area network), BBN(broadband network), 인터넷 등의 네트워크 중 하나 이상의 임의의 네트워크를 포함할 수 있다. 또한, 네트워크(660)는 버스 네트워크, 스타 네트워크, 링 네트워크, 메쉬 네트워크, 스타-버스 네트워크, 트리 또는 계층적(hierarchical) 네트워크 등을 포함하는 네트워크 토폴로지 중 임의의 하나 이상을 포함할 수 있으나, 이에 제한되지 않는다.

[112] 도 7은 본 발명의 일실시예에 따른 에이전트 학습 방법의 예를 도시한 흐름도이다. 본 실시예에 따른 에이전트 학습 방법은 일례로 앞서 설명한 컴퓨터 장치(600)에 의해 수행될 수 있다. 예를 들어, 컴퓨터 장치(600)의 프로세서(620)는 메모리(610)가 포함하는 운영체제의 코드나 적어도 하나의 프로그램의 코드에 따른 제어 명령(instruction)을 실행하도록 구현될 수 있다. 여기서, 프로세서(620)는 컴퓨터 장치(600)에 저장된 코드가 제공하는 제어 명령에 따라 컴퓨터 장치(600)가 도 7의 방법이 포함하는 단계들(710 내지 750)을 수행하도록 컴퓨터 장치(600)를 제어할 수 있다.

[113] 기본적으로 컴퓨터 장치(600)는 심층 강화 학습을 위한 시뮬레이션상에서 액터-크리틱 알고리즘을 통해 에이전트를 학습시킬 수 있다. 일례로, 컴퓨터 장치(600)는 액터-크리틱 알고리즘에서 에이전트의 행동을 결정하는 평가망인 액터 네트워크에 제1 정보를, 에이전트의 행동이 기설정된 보상을 최대화하는데 얼마나 도움이 되는가를 평가하는 가치망인 크리틱에 제2 정보를 입력할 수 있다. 이때, 제2 정보는 제1 정보와 추가 정보를 포함 이러한 에이전트의 학습을 위한 구체적인 일실시예로, 아래 단계들(710 내지 750)이 컴퓨터 장치(600)에 의해 수행될 수 있다.

[114] 단계(710)에서 컴퓨터 장치(600)는 뎁스 스캔, 에이전트의 속도 및 타임스케일 중 적어도 하나에 노이즈를 추가하는 다이내믹 무작위화를 통해 시뮬레이션을 위한 정보를 생성할 수 있다. 이러한 다이내믹 무작위화에 대해서는 앞서 자세히 설명한 바 있다.

[115] 단계(720)에서 컴퓨터 장치(600)는 생성된 정보 중 뎁스 스캔과 속도가 투영된 동일한 사이즈의 벡터들에 성분 내적을 적용하여 관찰 특징을 구현할 수 있다.

[116] 단계(730)에서 컴퓨터 장치(600)는 복수의 콘볼루션 레이어를 통과한 로컬 맵 텐서(tensor)에 글로벌 에버리지 풀링을 적용하여 로컬-맵 특징을 구현할 수 있다.

[117] 단계(740)에서 컴퓨터 장치(600)는 액터-크리틱 알고리즘에서 에이전트의 행동을 결정하는 평가망인 액터 네트워크에 관찰 특징 및 경유지를 입력할 수

있다. 여기서 관찰 특징과 경유지는 상술한 제1 정보에 대응할 수 있다.

[118] 단계(750)에서 컴퓨터 장치(600)는 액터-크리틱 알고리즘에서 에이전트의 행동이 기설정된 보상을 최대화하는데 얼마나 도움이 되는가를 평가하는 가치망인 크리틱에 관찰 특징, 경유지 및 로컬-맵 특징을 입력할 수 있다. 여기서, 로컬-맵 특징은 상술한 추가 정보에 대응할 수 있다. 다시 말해, 컴퓨터 장치(600)는 제1 정보로서 관찰 특징 및 경유지를 액터 네트워크에 입력하고, 제1 정보로서의 관찰 특징 및 경유지와 추가 정보로서의 로컬-맵 특징을 크리틱에 입력할 수 있다.

[119] 여기서, 로컬-맵 특징은 복수의 콘볼루션 레이어를 통과한 로컬 맵 텐서(tensor)에 글로벌 에버리지 풀링을 적용하여 구현될 수 있다. 예를 들어, 로컬-맵 특징은 전체 장애물 배치 상황, 이동하는 장애물의 속도 및 상기 이동하는 장애물의 목표 중 적어도 하나의 정보를 포함할 수 있다. 또한, 관찰 특징은 뎁스 스캔과 속도가 투영된 동일한 크기의 벡터들에 성분 내적(elementwise product)을 적용하여 구현될 수 있다. 경유지는 랜덤하게 설정될 수 있다.

[120] 단계(760)에서 컴퓨터 장치(600)는 액터 네트워크와 크리틱 각각에서 입력된 정보들이 연결(concatenation)된 시계열적인 데이터를 액터 네트워크와 크리틱 각각이 포함하는 순환 신경망에 입력할 수 있다. 이때, 컴퓨터 장치(600)는 순환 신경망에 저장된 이전의 센서 값을 통해 에이전트가 현재 시야 밖의 환경에 대한 정보를 획득하여 동작하도록 학습할 수 있다. 일례로, 순환 신경망은 LSTM(Long-Short Term Memory) 방식의 순환 신경망을 포함할 수 있다.

[121] 이처럼 본 발명의 실시예들에 따르면, 심층 강화 학습(Deep Reinforcement Learning, DRL)을 위한 시뮬레이션상에서, 액터-크리틱 알고리즘의 정책망과 가치망 중 가치망에 실세계에서 얻기 힘들지만 학습에 도움이 되는 정보를 시뮬레이션의 상태에서 직접 추출해 제공함으로써, 학습 시 사용되는 가치망에서는 에이전트의 행동의 가치에 대한 더 정확한 평가를 내릴 수 있도록 하여 정책망의 성능을 향상시킬 수 있다. 또한, LSTM(Long-Short Term Memory)과 같은 순환 신경망(Recurrent Neural Network)의 메모리를 활용하여 에이전트가 현재 시야 밖의 환경에 대한 정보를 순환 신경망에 저장된 이전의 센서 값을 통해 획득할 수 있도록 함으로써, 제한된 시야를 갖는 에이전트도 보다 효과적으로 자율주행이 가능하도록 할 수 있다.

[122] 이상에서 설명된 시스템 또는 장치는 하드웨어 구성요소, 또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를

이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 어플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[123] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록매체에 저장될 수 있다.

[124] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 매체는 컴퓨터로 실행 가능한 프로그램을 계속 저장하거나, 실행 또는 다운로드를 위해 임시 저장하는 것일 수도 있다. 또한, 매체는 단일 또는 수개 하드웨어가 결합된 형태의 다양한 기록수단 또는 저장수단일 수 있는데, 어떤 컴퓨터 시스템에 직접 접속되는 매체에 한정되지 않고, 네트워크 상에 분산 존재하는 것일 수도 있다. 매체의 예시로는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM 및 DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical medium), 및 ROM, RAM, 플래시 메모리 등을 포함하여 프로그램 명령어가 저장되도록 구성된 것이 있을 수 있다. 또한, 다른 매체의 예시로, 애플리케이션을 유통하는 앱 스토어나 기타 다양한 소프트웨어를 공급 내지 유통하는 사이트, 서버 등에서 관리하는 기록매체 내지 저장매체도 들 수 있다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

발명의 실시를 위한 형태

[125] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

[126] 그러므로, 다른 구현들, 다른 실시예들 및 청구범위와 균등한 것들도 후술하는 청구범위의 범위에 속한다.

청구범위

[청구항 1] 적어도 하나의 프로세서를 포함하는 컴퓨터 장치의 에이전트 학습 방법에 있어서,
상기 적어도 하나의 프로세서에 의해, 심층 강화 학습(Deep Reinforcement Learning, DRL)을 위한 시뮬레이션상에서 액터-크리틱(actor-critic) 알고리즘을 통해 에이전트를 학습시키는 단계
를 포함하고,
상기 학습시키는 단계는,
상기 액터-크리틱 알고리즘에서 에이전트의 행동을 결정하는 평가망인 액터 네트워크에 제1 정보를, 상기 행동이 기설정된 보상을 최대화하는데 얼마나 도움이 되는가를 평가하는 가치망인 크리틱에 제2 정보를 입력하고,
상기 제2 정보는 상기 제1 정보와 추가 정보를 포함하는 것을 특징으로 하는 에이전트 학습 방법.

[청구항 2] 제1항에 있어서,
상기 학습시키는 단계는,
상기 제1 정보로서 관찰 특징 및 경유지를 상기 액터 네트워크에 입력하고, 상기 제1 정보로서의 상기 관찰 특징 및 상기 경유지와 상기 추가 정보로서의 로컬-맵 특징을 상기 크리틱에 입력하는 것을 특징으로 하는 에이전트 학습 방법.

[청구항 3] 제2항에 있어서,
상기 로컬-맵 특징은 복수의 콘볼루션 레이어를 통과한 로컬 맵 텐서(tensor)에 글로벌 에버리지 풀링을 적용하여 구현되는 것을 특징으로 하는 에이전트 학습 방법.

[청구항 4] 제2항에 있어서,
상기 로컬-맵 특징은 전체 장애물 배치 상황, 이동하는 장애물의 속도 및 상기 이동하는 장애물의 목표 중 적어도 하나의 정보를 포함하는 것을 특징으로 하는 에이전트 학습 방법.

[청구항 5] 제2항에 있어서,
상기 관찰 특징은 뎁스 스캔과 속도가 투영된 동일한 크기의 벡터들에 성분 내적(elementwise product)을 적용하여 구현되는 것을 특징으로 하는 에이전트 학습 방법.

[청구항 6] 제1항에 있어서,
상기 액터 네트워크 및 상기 크리틱 각각은 시계열적인 데이터를 입력으로 받는 순환 신경망(Recurrent Neural Network)을 포함하고,
상기 학습시키는 단계는,
상기 순환 신경망에 저장된 이전의 센서 값을 통해 상기 에이전트가 현재

시야 밖의 환경에 대한 정보를 획득하여 동작하도록 학습시키는 것을 특징으로 하는 에이전트 학습 방법.

[청구항 7] 제6항에 있어서,
상기 순환 신경망은 LSTM(Long-Short Term Memory) 방식의 순환 신경망을 포함하는 것을 특징으로 하는 에이전트 학습 방법.

[청구항 8] 제1항에 있어서,
상기 학습시키는 단계는,
뎁스 스캔, 에이전트의 속도 및 타임스케일 중 적어도 하나에 노이즈를 추가하는 다이내믹 무작위화(dynamics randomization)를 통해 상기 시뮬레이션을 위한 정보를 생성하는 것을 특징으로 하는 에이전트 학습 방법.

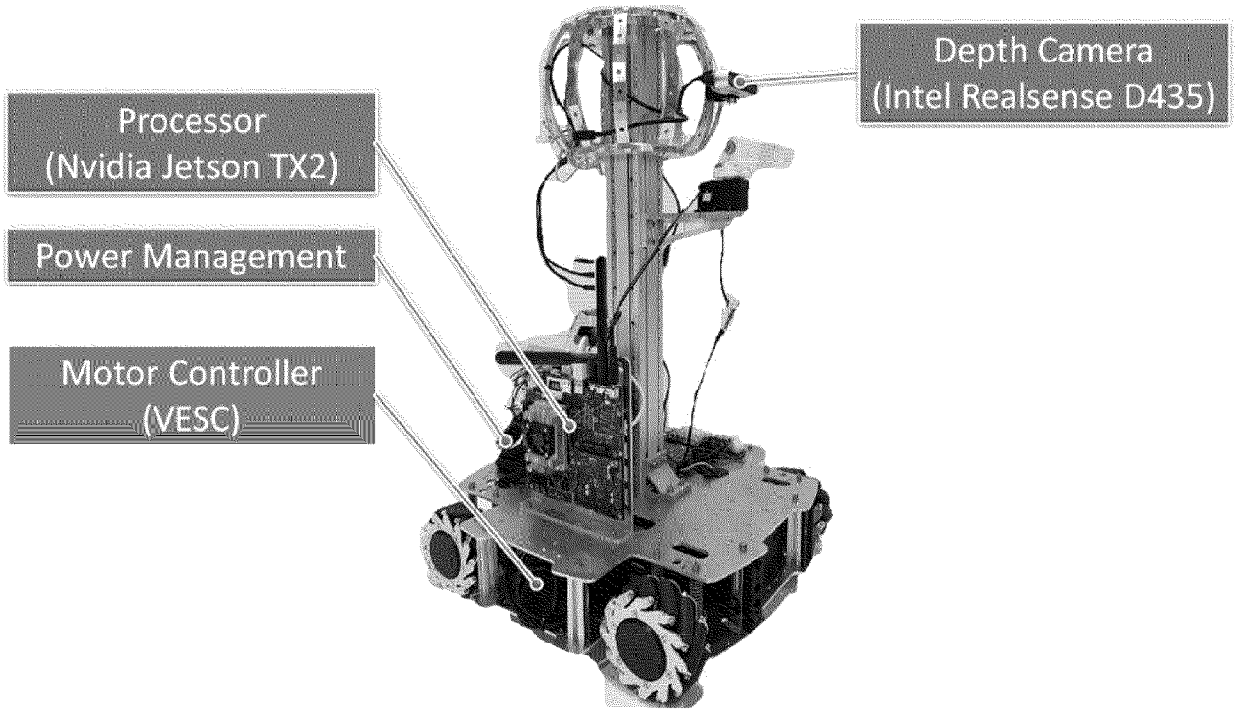
[청구항 9] 컴퓨터 장치와 결합되어 제1항 내지 제8항 중 어느 한 항의 방법을 컴퓨터 장치에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램.

[청구항 10] 제1항 내지 제8항 어느 한 항의 방법을 컴퓨터 장치에 실행시키기 위한 컴퓨터 프로그램이 기록되어 있는 컴퓨터 판독 가능한 기록매체.

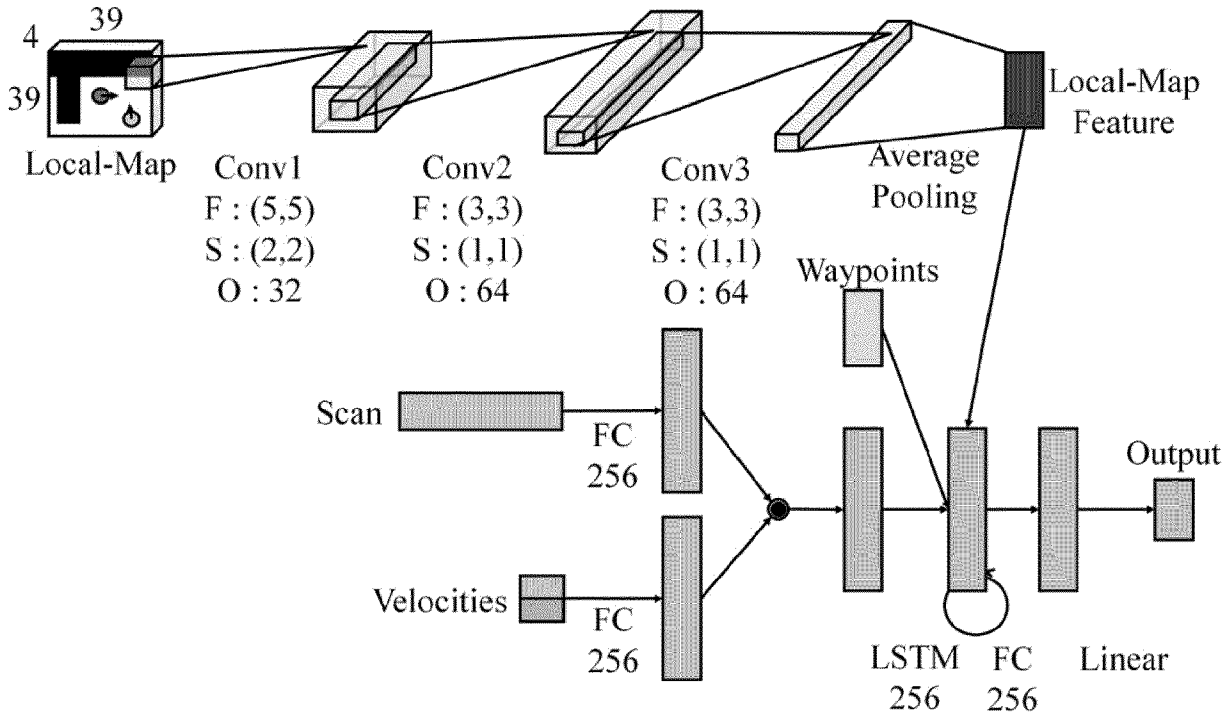
[청구항 11] 제1항 내지 제8항 어느 한 항의 방법을 통해 학습된 에이전트가 탑재된 모바일 로봇 플랫폼.

[청구항 12] 컴퓨터에서 판독 가능한 명령을 실행하도록 구현되는 적어도 하나의 프로세서
를 포함하고,
상기 적어도 하나의 프로세서에 의해,
심층 강화 학습(Deep Reinforcement Learning, DRL)을 위한 시뮬레이션상에서 액터-크리틱(actor-critic) 알고리즘을 통해 에이전트를 학습시키고,
상기 에이전트를 학습시키기 위해, 상기 액터-크리틱 알고리즘에서 에이전트의 행동을 결정하는 평가망인 액터 네트워크에 제1 정보를, 상기 행동이 기설정된 보상을 최대화하는데 얼마나 도움이 되는가를 평가하는 가치망인 크리틱에 제2 정보를 입력하고,
상기 제2 정보는 상기 제1 정보와 추가 정보를 포함하는 것
을 특징으로 하는 컴퓨터 장치.

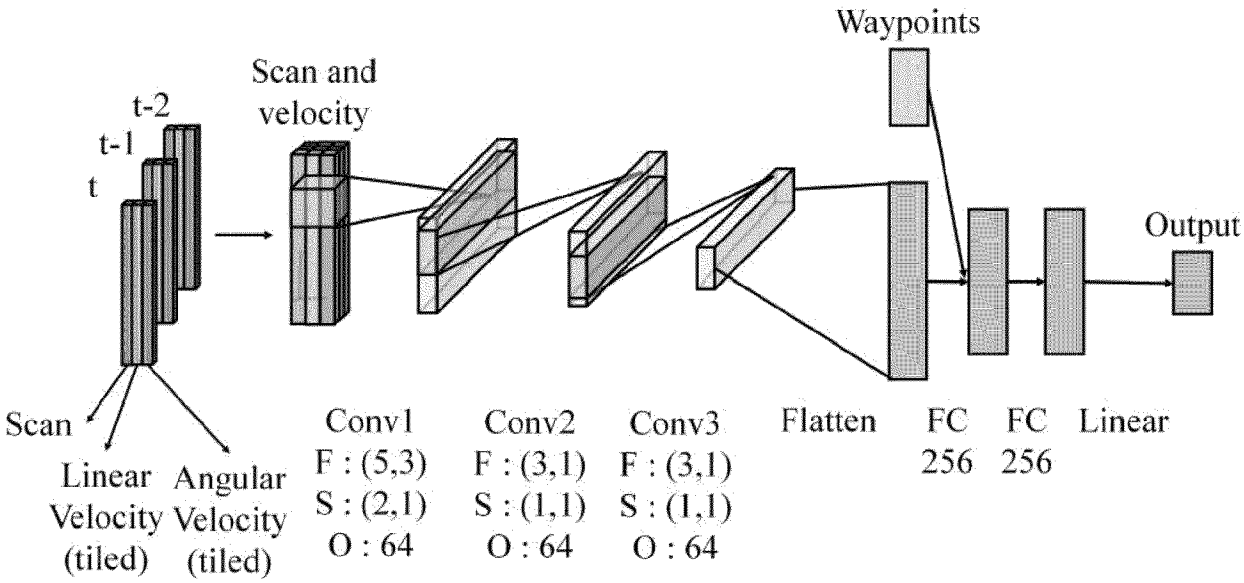
[도1]


Depth Camera
(Intel Realsense D435)
Processor
(Nvidia Jetson TX2)
Power Management
Motor Controller
(VESC)

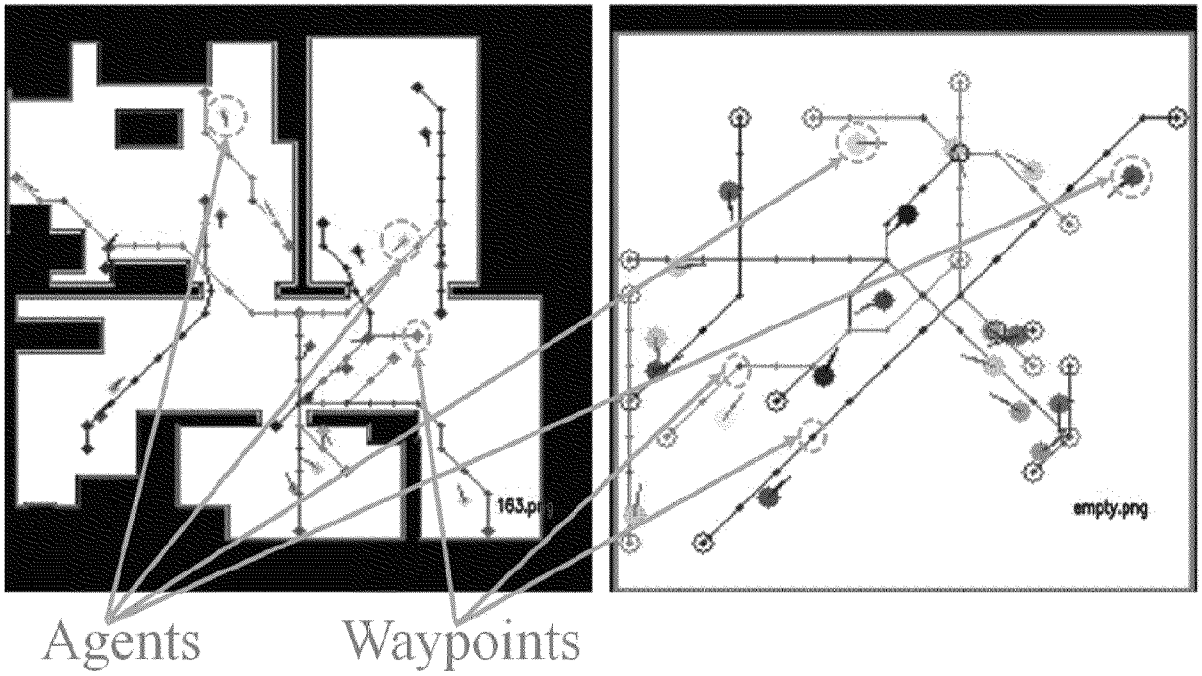
[도2]


4
39
39
Local-Map
Conv1
F : (5,5)
S : (2,2)
O : 32
Conv2
F : (3,3)
S : (1,1)
O : 64
Conv3
F : (3,3)
S : (1,1)
O : 64
Average
Pooling
Local-Map
Feature
Waypoints
Scan
FC
256
Velocities
FC
256
Output
LSTM
256
FC
256
Linear

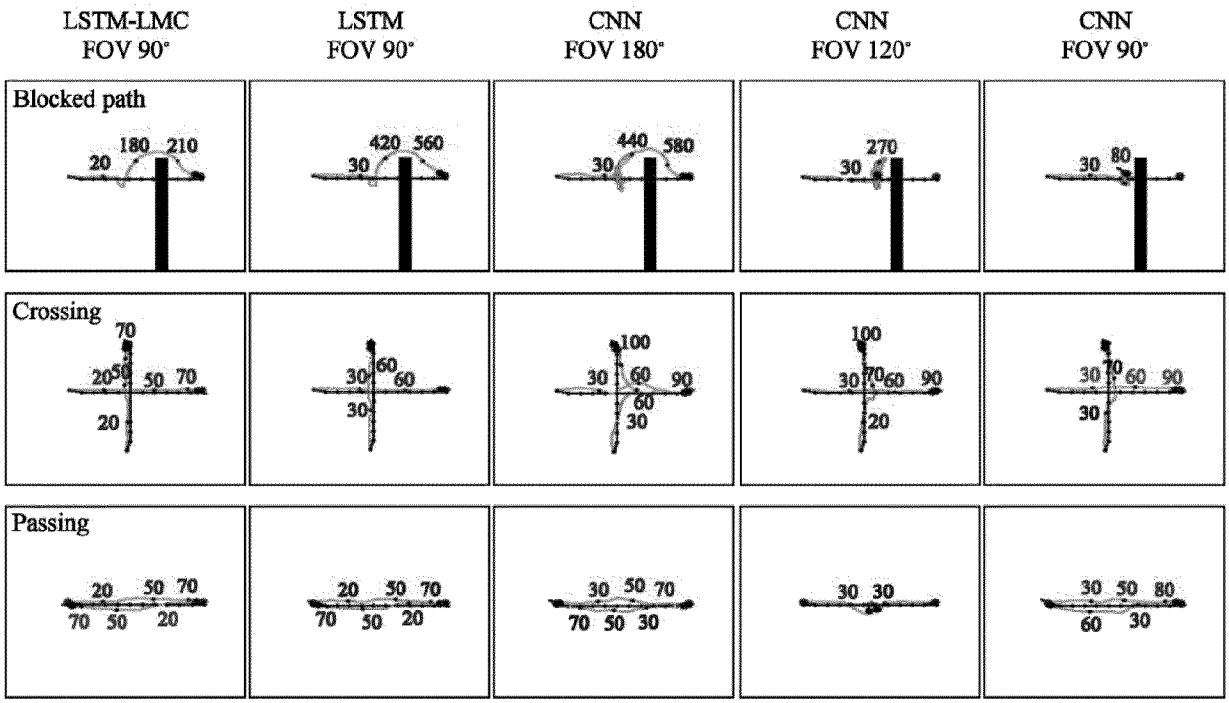
[도3]


Waypoints
Scan and velocity
t-2
t-1
t
Output
Scan
Linear Velocity (tiled)
Angular Velocity (tiled)
Conv1 F : (5,3) S : (2,1) O : 64
Conv2 F : (3,1) S : (1,1) O : 64
Conv3 F : (3,1) S : (1,1) O : 64
Flatten
FC 256
FC 256
Linear

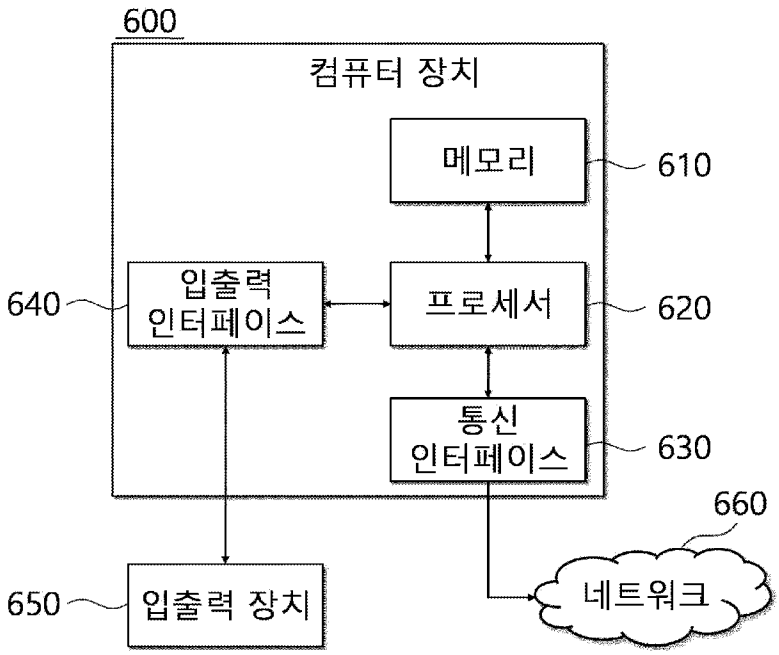
[도4]


163.png
empty.png
Agents
Waypoints

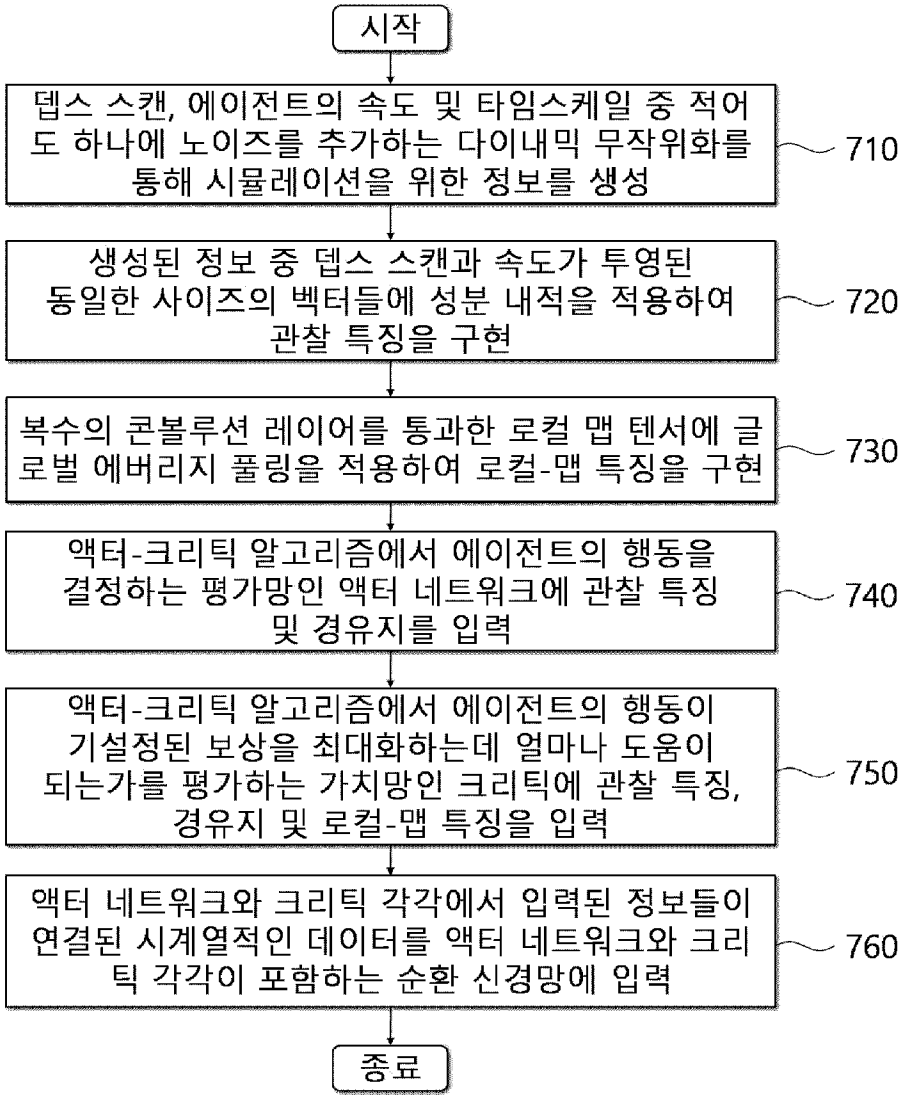
[도5]


LSTM-LMC
FOV 90°
LSTM
FOV 90°
CNN
FOV 180°
CNN
FOV 120°
CNN
FOV 90°
Blocked path
Crossing
Passing

[도6]


600
컴퓨터 장치
메모리
610
입출력
인터페이스
640
프로세서
620
통신
인터페이스
630
660
네트워크
650
입출력 장치

[도7]


시작
뎁스 스캔, 에이전트의 속도 및 타임스케일 중 적어도 하나에 노이즈를 추가하는 다이내믹 무작위화를 통해 시뮬레이션을 위한 정보를 생성
710
생성된 정보 중 뎁스 스캔과 속도가 투영된 동일한 사이즈의 벡터들에 성분 내적을 적용하여 관찰 특징을 구현
720
복수의 콘볼루션 레이어를 통과한 로컬 맵 텐서에 글로벌 에버리지 풀링을 적용하여 로컬-맵 특징을 구현
730
액터-크리틱 알고리즘에서 에이전트의 행동을 결정하는 평가망인 액터 네트워크에 관찰 특징 및 경유지를 입력
740
액터-크리틱 알고리즘에서 에이전트의 행동이 기설정된 보상을 최대화하는데 얼마나 도움이 되는가를 평가하는 가치망인 크리틱에 관찰 특징, 경유지 및 로컬-맵 특징을 입력
750
액터 네트워크와 크리틱 각각에서 입력된 정보들이 연결된 시계열적인 데이터를 액터 네트워크와 크리틱 각각이 포함하는 순환 신경망에 입력
760
종료