



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0141074
(43) 공개일자 2024년09월25일

- (51) 국제특허분류(Int. Cl.)
G06V 20/58 (2022.01) G06N 3/08 (2023.01)
G06T 7/62 (2017.01) G06V 10/40 (2022.01)
G06V 10/62 (2022.01) G06V 10/74 (2022.01)
G06V 10/772 (2022.01) G06V 10/774 (2022.01)
G06V 20/17 (2022.01) G06V 20/40 (2022.01)
- (52) CPC특허분류
G06V 20/58 (2022.01)
G06N 3/08 (2023.01)
- (21) 출원번호 10-2023-0035071
(22) 출원일자 2023년03월17일
심사청구일자 2023년03월17일
- (71) 출원인
국립창원대학교 산학협력단
경상남도 창원시 의창구 창원대학교 20(퇴촌동)
- (72) 발명자
안동혁
경상남도 창원시 의창구 도계로 136, 205동 1004호 (명서동, 두산위브아파트)
- 박현진
경상남도 창원시 의창구 사림로99번길 18-22, 2층 202호 (사림동)
- (74) 대리인
양성보

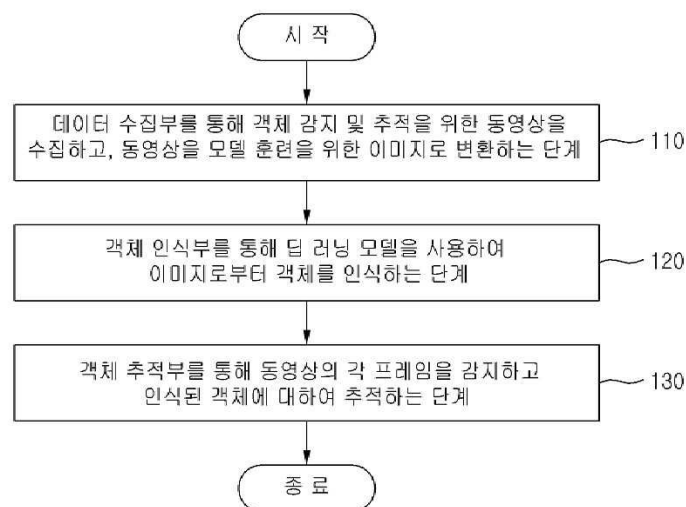
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 동영상 기반 장애물 추적 방법 및 시스템

(57) 요약

동영상 기반 장애물 추적 방법 및 시스템이 제시된다. 본 발명에서 제안하는 동영상 기반 장애물 추적 방법은 데이터 수집부를 통해 객체 감지 및 추적을 위한 동영상을 수집하고, 동영상을 모델 훈련을 위한 이미지로 변환하는 단계, 객체 인식부를 통해 딥 러닝 모델을 사용하여 이미지로부터 객체를 인식하는 단계 및 객체 추적부를 통해 동영상의 각 프레임을 감지하고 인식된 객체에 대하여 추적하는 단계를 포함한다.

대표도 - 도1



(52) CPC특허분류

G06T 7/62 (2017.01)
 G06V 10/40 (2023.08)
 G06V 10/62 (2023.08)
 G06V 10/761 (2023.08)
 G06V 10/772 (2023.08)
 G06V 10/774 (2023.08)
 G06V 20/17 (2023.08)
 G06V 20/40 (2023.08)
 G06T 2210/12 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711191293
과제번호	2020M3C1C2A01080838
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	DNA+드론기술개발
연구과제명	드론을 위한 AI 핵심 S/W개발
기 여 율	1/1
과제수행기관명	주식회사 아크틸
연구기간	2023.01.01 ~ 2023.12.31

명세서

청구범위

청구항 1

데이터 수집부를 통해 객체 감지 및 추적을 위한 동영상을 수집하고, 상기 동영상을 모델 훈련을 위한 이미지로 변환하는 단계;

객체 인식부를 통해 딥 러닝 모델을 사용하여 상기 이미지로부터 객체를 인식하는 단계; 및

객체 추적부를 통해 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 단계

를 포함하는 영상 기반 장애물 추적 방법.

청구항 2

제1항에 있어서,

상기 데이터 수집부를 통해 객체 감지 및 추적을 위한 동영상을 수집하고, 상기 동영상을 모델 훈련을 위한 이미지로 변환하는 단계는,

모델 훈련을 위한 데이터 입력이 이미지이므로 상기 수집된 동영상을 이미지로 변환하고, 변환된 이미지의 데이터셋은 연속 프레임 간의 유사도 때문에 발생하는 과적합을 방지하기 위해 동영상의 모든 프레임을 선택하지 않고, 미리 정해진 복수의 프레임마다 하나의 이미지가 저장되는 방식으로 생성되는

영상 기반 장애물 추적 방법.

청구항 3

제1항에 있어서,

상기 객체 인식부를 통해 딥 러닝 모델을 사용하여 상기 이미지로부터 객체를 인식하는 단계는,

BOS(Bag of Specials)의 데이터 확대 기법 중 모자이크 기법을 사용하여 복수의 개별 이미지를 경계 상자와 결합해 하나의 이미지를 생성하고, 하나의 입력만으로 복수의 이미지를 학습하며, k-평균 알고리즘을 사용하여 새로운 앵커 크기를 생성하고, 사전 훈련된 컨볼루션 가중치를 이용하여 딥 러닝 모델 훈련을 수행하는

영상 기반 장애물 추적 방법.

청구항 4

제1항에 있어서,

상기 객체 추적부를 통해 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 단계는,

상기 인식된 객체가 프레임 아웃되고 다시 프레임 인 되었을 때 동일한 객체로 식별되고, 복수의 객체가 겹쳐지거나 객체가 가려진 경우 해당 객체가 다시 나타날 때 동일한 객체로 식별되도록 동영상의 각 프레임을 감지하고 추적하는

영상 기반 장애물 추적 방법.

청구항 5

제4항에 있어서,

상기 객체 추적부를 통해 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 단계는,

현재 프레임의 객체의 경계 상자와 모든 이전 프레임의 객체의 경계 상자를 이용하여 IoU(intersection over union) 값을 계산한 후, 계산된 IoU 값에 따라 IoU 매칭을 수행하는

영상 기반 장애물 추적 방법.

청구항 6

제4항에 있어서,

상기 객체 추적부를 통해 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 단계는,

ORB(oriented FAST and rotated BRIEF) 기반 특징 매칭을 수행하고 객체 크기 및 객체 간 특징을 비교하여 객체 유사성을 결정하는

영상 기반 장애물 추적 방법.

청구항 7

객체 감지 및 추적을 위한 동영상을 수집하고, 상기 동영상을 모델 훈련을 위한 이미지로 변환하는 데이터 수집부;

딥 러닝 모델을 사용하여 상기 이미지로부터 객체를 인식하는 객체 인식부; 및

동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 객체 추적부

를 포함하는 영상 기반 장애물 추적 시스템.

청구항 8

제7항에 있어서,

상기 데이터 수집부는,

모델 훈련을 위한 데이터 입력이 이미지이므로 상기 수집된 동영상을 이미지로 변환하고, 변환된 이미지의 데이터셋은 연속 프레임 간의 유사도 때문에 발생하는 과적합을 방지하기 위해 동영상의 모든 프레임을 선택하지 않고, 미리 정해진 복수의 프레임마다 하나의 이미지가 저장되는 방식으로 생성되는

영상 기반 장애물 추적 시스템.

청구항 9

제7항에 있어서,

상기 객체 인식부는,

BOS(Bag of Specials)의 데이터 확대 기법 중 모자이크 기법을 사용하여 복수의 개별 이미지를 경계 상자와 결합해 하나의 이미지를 생성하고, 하나의 입력만으로 복수의 이미지를 학습하며, k-평균 알고리즘을 사용하여 새로운 앵커 크기를 생성하고, 사전 훈련된 컨볼루션 가중치를 이용하여 딥 러닝 모델 훈련을 수행하는

영상 기반 장애물 추적 시스템.

청구항 10

제7항에 있어서,

상기 객체 추적부는,

상기 인식된 객체가 프레임 아웃되고 다시 프레임 인 되었을 때 동일한 객체로 식별되고, 복수의 객체가 겹쳐지거나 객체가 가려진 경우 해당 객체가 다시 나타날 때 동일한 객체로 식별되도록 동영상의 각 프레임을 감지하고 추적하는

영상 기반 장애물 추적 시스템.

청구항 11

제10항에 있어서,

상기 객체 추적부는,

현재 프레임의 객체의 경계 상자와 모든 이전 프레임의 객체의 경계 상자를 이용하여 IoU(intersection over

union) 값을 계산한 후, 계산된 IoU 값에 따라 IoU 매칭을 수행하는
영상 기반 장애물 추적 시스템.

청구항 12

제10항에 있어서,

상기 객체 추적부는,

ORB(oriented FAST and rotated BRIEF) 기반 특징 매칭을 수행하고 객체 크기 및 객체 간 특징을 비교하여 객체 유사성을 결정하는

영상 기반 장애물 추적 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 동영상 기반 장애물 추적 방법 및 시스템에 관한 것이다.

배경 기술

[0002] 최근 몇 년 동안 프로세싱 성능, 통신 및 센서 기술이 급속도로 발전하면서 자율 주행 기술이 많은 주목을 받고 있다. 자동차, 선박, 드론과 같은 이동 차량은 사람의 개입 없이 스스로 운전할 수 있도록 인접한 장애물을 신속하게 인식하고 대응할 수 있어야 한다. 자율주행 능력 향상을 위해 자율주행차에는 카메라, 레이더, LIDAR(Laser Imaging, Detection and Ranging), GPS(Global Positioning System), SONAR(Sound Navigation and Ranging) 등의 기술이 활용된다. 이들 외에도 해상 자율 수상선박은 자동식별시스템(Automatic Identification System; AIS)과 GPS를 이용해 주변 선박을 감지하고 다른 선박과의 충돌을 방지한다. 반면 부표, 소형 보트, 카약 등 소형 선박에는 AIS 송신기가 없다. 또 야간에 무단으로 조업하는 불법조업 어선은 AIS를 차단해 선박 사고가 발생할 가능성이 있다. 예상치 못한 선박 충돌 등 해양 불상사를 막기 위해서는 카메라 동영상을 이용해 물체를 인식하는 기술이 필요하다. 주변 선박을 파악한 뒤 다른 선박의 이동선과 속도, 방향 등의 정보가 있어야 충돌을 피할 수 있는 항로를 계획할 수 있다. 동영상에서의 객체 인식은 프레임별로 수행되기 때문에 시간에 따른 주어진 객체의 위치 변화를 감지하는 것이 불가능하다. 경로 탐색을 위해서는 이동선, 속도, 방향 등을 결정하는 객체 추적 기술도 필요하다.

[0003] 동영상 기반 물체 감지 및 추적에 대한 여러 연구가 수행되었다. R-CNN(Regions with Convolutional Neural Network Features) 및 YOLO(You Only Look Once)와 같은 딥 러닝 기반 객체 인식은 객체의 위치를 식별하고 카메라의 이미지 스트림에서 유형을 예측한다. 추적 알고리즘인 DeepSORT는 그것이 같은 객체인지를 평가한다. 그러나, 카메라를 배에 장착할 때, 배의 움직임으로 인해 이미지 구성을 안정적으로 유지할 수 없다. 대상이 프레임 아웃과 프레임 인을 지속적으로 경험하기 때문에 상황에서 동일한 대상으로 인식하기 어렵다. 또한, 여러 크기의 선박들이 서로 다른 방향으로 이동함으로써 동영상에서 서로 겹치기 때문에, 겹치는 물체들은 화면에서 사라진다. 숨겨진 객체가 다시 나타나면 일반적으로 다른 객체로 인식된다. 따라서, 동영상에서 이동하는 객체 인식 및 추적 알고리즘을 필요로 한다.

선행기술문헌

비특허문헌

[0004] (비특허문헌 0001) [1] Singapore Maritime Dataset. Available online: <https://www.kaggle.com/mmichelli/singapore-maritime-dataset> (accessed on 1 January 2022).

(비특허문헌 0002) [2] Capobianco, S.; Millefiori, L.M.; Forti, N.; Braca, P.; Willett, P. Deep learning methods for vessel trajectory prediction based on recurrent neural networks. IEEE Trans. Aerosp. Electron. Syst. 2021, 57, 4329-4346. [CrossRef]

(비특허문헌 0003) [3] Zamboni, S.; Kefato, Z.T.; Girdzijauskas, S.; Noren, C.; Dal Col, L. Pedestrian trajectory prediction with convolutional neural networks. Pattern Recognit. 2022, 121, 108252.

[CrossRef]

(비특허문헌 0004) [4] Jiang, H.; Chang, L.; Li, Q.; Chen, D. Trajectory prediction of vehicles based on deep learning. In Proceedings of the 2019 4th International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 5-7 September 2019; pp. 190-195.

(비특허문헌 0005) [5] Xu, Y.; Piao, Z.; Gao, S. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18-23 June 2018; pp. 5275-5284.

발명의 내용

해결하려는 과제

[0005] 본 발명이 이루고자 하는 기술적 과제는 훈련 중에 수천 개의 훈련 데이터를 활용하고 훈련 데이터의 부족과 레이블의 불균형을 피하기 위해 전이 학습 기술을 사용하는 동영상 기반 장애물 추적 방법 및 시스템을 제공하는 데 있다. 또한, 객체 인식을 위해 YOLO를 기반으로 하는 인식 모델을 이용하여 객체를 빠르게 감지하고, 동작이 높은 동영상에서의 객체 추적을 위해 IoU_matching과 ORB_and_size_matching을 활용하는 추적 알고리즘을 제안한다.

과제의 해결 수단

[0006] 일 측면에 있어서, 본 발명에서 제안하는 동영상 기반 장애물 추적 방법은 데이터 수집부를 통해 객체 감지 및 추적을 위한 동영상을 수집하고, 상기 동영상을 모델 훈련을 위한 이미지로 변환하는 단계, 객체 인식부를 통해 딥 러닝 모델을 사용하여 상기 이미지로부터 객체를 인식하는 단계 및 객체 추적부를 통해 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 단계를 포함한다.

[0007] 상기 데이터 수집부를 통해 객체 감지 및 추적을 위한 동영상을 수집하고, 상기 동영상을 모델 훈련을 위한 이미지로 변환하는 단계는 모델 훈련을 위한 데이터 입력이 이미지이므로 상기 수집된 동영상을 이미지로 변환하고, 변환된 이미지의 데이터셋은 연속 프레임 간의 유사도 때문에 발생하는 과적합을 방지하기 위해 동영상의 모든 프레임을 선택하지 않고, 미리 정해진 복수의 프레임마다 하나의 이미지가 저장되는 방식으로 생성된다.

[0008] 상기 객체 인식부를 통해 딥 러닝 모델을 사용하여 상기 이미지로부터 객체를 인식하는 단계는 BOS(Bag of Specials)의 데이터 확대 기법 중 모자이크 기법을 사용하여 복수의 개별 이미지를 경계 상자와 결합해 하나의 이미지를 생성하고, 하나의 입력만으로 복수의 이미지를 학습하며, k-평균 알고리즘을 사용하여 새로운 앵커 크기를 생성하고, 사전 훈련된 컨볼루션 가중치를 이용하여 딥 러닝 모델 훈련을 수행한다.

[0009] 상기 객체 추적부를 통해 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 단계는 상기 인식된 객체가 프레임 아웃되고 다시 프레임 인 되었을 때 동일한 객체로 식별되고, 복수의 객체가 겹쳐지거나 객체가 가려진 경우 해당 객체가 다시 나타날 때 동일한 객체로 식별되도록 동영상의 각 프레임을 감지하고 추적한다.

[0010] 상기 객체 추적부를 통해 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 단계는 현재 프레임의 객체의 경계 상자와 모든 이전 프레임의 객체의 경계 상자를 이용하여 IoU(intersection over union) 값을 계산한 후, 계산된 IoU 값에 따라 IoU 매칭을 수행한다.

[0011] 상기 객체 추적부를 통해 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 단계는 ORB(oriented FAST and rotated BRIEF) 기반 특징 매칭을 수행하고 객체 크기 및 객체 간 특징을 비교하여 객체 유사성을 결정한다.

[0012] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 동영상 기반 장애물 추적 시스템은 객체 감지 및 추적을 위한 동영상을 수집하고, 상기 동영상을 모델 훈련을 위한 이미지로 변환하는 데이터 수집부, 딥 러닝 모델을 사용하여 상기 이미지로부터 객체를 인식하는 객체 인식부 및 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 객체 추적부를 포함한다.

발명의 효과

[0013] 본 발명의 실시예들에 따른 사용하는 동영상 기반 장애물 추적 방법 및 시스템을 통해 훈련 중에 수천 개의 훈련 데이터를 활용하고 훈련 데이터의 부족과 레이블의 불균형을 피하기 위해 전이 학습 기술을 사용한다. 또한, 객체 인식을 위해 YOLO를 기반으로 하는 인식 모델을 이용하여 객체를 빠르게 감지하고, IoU_matching과 ORB_and_size_matching을 활용하는 추적 알고리즘을 통해 동작이 높은 동영상에서의 객체를 빠르게 추적할 수 있다.

도면의 간단한 설명

[0014] 도 1은 본 발명의 일 실시예에 따른 동영상 기반 장애물 추적 방법을 설명하기 위한 흐름도이다.
 도 2는 본 발명의 일 실시예에 따른 동영상 기반 장애물 추적 시스템의 구성을 나타내는 도면이다.
 도 3은 본 발명의 일 실시예에 따른 객체 인식을 위한 딥러닝 모델의 예시를 나타내는 도면이다.
 도 4는 본 발명의 일 실시예에 따른 객체 추적 알고리즘을 나타내는 도면이다.
 도 5는 본 발명의 일 실시예에 따른 IoU_matching 알고리즘을 나타내는 도면이다.
 도 6은 본 발명의 일 실시예에 따른 IoU_matching 알고리즘을 나타내는 도면이다.
 도 7은 본 발명의 일 실시예에 따른 영상 기반 장애물 추적 알고리즘과 DeepSORT를 이용한 선박 추적 결과를 비교하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0015] 객체 인식은 미리 감지할 객체의 특징을 추출한 다음 주어진 이미지 내에서 그 특징을 인식한다. 캐니(Canny) 에지 감지기, 해리스(Harris) 코너, HOG(Histogram of Oriented Gradient) 및 SIFT(scale-invariant feature transform)와 같은 다양한 물체 감지 기법이 과거에 제시되었다. 최근 몇 년 동안 CNN(Convolutional Neural Network)에 기반한 인식 알고리즘이 광범위하게 조사되었다. R-CNN(Regions with Convolutional Neural Network), VGG 시각 기하학 그룹 및 ResNet 잔차 네트워크와 같은 2단계 감지기가 객체 인식을 위해 제안되었다. 이러한 종래기술들은 영역 제안을 사용하여 객체를 찾고 지역 분류를 사용하여 객체를 분류한다. 2 단계 감지기의 단점은 물체를 천천히 감지하여 실시간 감지를 비현실적으로 만든다는 것이다. 영역 제안과 분류를 동시에 수행하는 1단계 감지기인 YOLO(You Only Look Once)는 실시간 감지를 용이하게 하기 위해 제안되었다.

[0016] GoogLeNet에 구축된 첫 번째 YOLO는 24개의 컨볼루션 계층과 초당 45프레임의 감지 속도로 2개의 완전 연결 계층으로 구성된 네트워크 구조가 특징이다. 종래기술에서, YOLO9000이라고도 하는 YOLOv2는 첫 번째 YOLO의 불량 감지율과 엄청난 수의 위치지정 오류를 해결하기 위해 10가지 성능 향상 요소를 제안하고, 초기 YOLO에 비해 mAP가 15% 증가했다. 그러나 백본 네트워크는 Darknet-53이며 ResNet short 연결이 사용된다. 종래기술에서 YOLOv4는 백본에 CSPDarknet53, 넥에 SPP(Spatial Pyramid Pooling)와 PAN(Path Aggregation Network), 헤드에 YOLOv3를 사용하였다. BOF(Bag of Freebies)와 BOS(Bag of Specials) 방식도 정확성과 성능을 향상시키기 위해 사용되었다. 또한, 병렬화된 계산 전략을 채택함으로써 단일 GPU 훈련 환경에서도 빠르고 정확한 학습을 달성할 수 있음을 보여준다.

[0017] 시간에 걸쳐 움직이는 객체를 감지하는 기술인 객체 추적은 연속적인 프레임 간의 시공간적 유사성을 추출하여 특정 객체 추적을 수행한다. FairMOT(Fair Multi-Object Tracking), SORT(Simple Online and Real-time Tracking), DeepSORT는 객체 추적 알고리즘이다. 종래기술에서 FairMOT는 초당 30프레임의 프레임 비율로 객체 감지 및 Re-ID 기능을 공정하게 수행한다. 계산 비용을 줄이기 위해 SORT는 칼만 필터와 Hungarian 알고리즘을 사용한다. 칼만 필터는 이전 속도를 기반으로 물체의 위치를 예측하기 때문에 두 물체가 겹치거나 방향이 빠르게 바뀌는 경우 추적이 어렵다. 종래기술에서 DeepSORT는 칼만 필터에 딥러닝 기능(Reflecting Deep Learning Features; Re-ID)을 반영함으로써 Hungarian 알고리즘을 사용하여 객체가 중첩되고 방향이 변경될 때에도 객체를 추적한다. 각 프레임에 대해 인식된 객체가 이전 프레임과 동일한지 여부를 평가하기 위해 객체 유사성을 기준으로 ID 레이블링을 수행한다. 그런 다음 DeepSort는 프레임의 여러 개체를 각각 추적할 수 있다.

[0018] 조선업계에서는 이미지 기반 선박 인식 및 추적에 대한 다양한 연구가 진행되고 있다. 종래기술에서 객체 인식 모델은 YOLOv2를 사용하고 신경망 토폴로지는 정밀도를 향상시키기 위해 조정된다. 훈련 데이터는 SMD 데이터셋에서 수집된 159개의 이미지로 구성되지만 레이블이 균일하게 분포되어 있지 않기 때문에 클래스별 감지 성능이

크게 다르다. 또한, 파생 모델은 초당 30프레임으로 실시간 처리가 가능한 반면, 리콜, 정밀도, mAP는 각각 76%, 66%, 69.79%의 상대적으로 낮은 성능 결과를 보인다. 종래기술에서는 선박 대상 감지의 정확도를 높이기 위해 Ship-YOLOv3가 제안되었다. YOLO3과 비교하면 정밀도와 리콜이 각각 12.5%, 11.5% 상승한다. 그러나 레이블이 3개뿐이고 브릿지에 장착된 카메라의 데이터를 활용하기 때문에 움직임이 높은 동영상에는 적합하지 않다. 종래기술에 있어서, 객체 감지 모델 및 객체 추적 방법은 각각 YOLOv3 및 DeepSORT를 이용하여 구현된다. 향상된 YOLOv3는 이전 버전과 비교하여 mAP와 FPS가 각각 5%, 2% 증가한다. 그것은 DeepSORT가 다른 6개의 추적 알고리즘을 능가한다는 것을 보여준다. 다만 해상 자율 수상함은 고정 카메라에서 4개의 동영상으로만 시험했기 때문에 적용하기에는 미흡하다. 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

도 1은 본 발명의 일 실시예에 따른 동영상 기반 장애물 추적 방법을 설명하기 위한 흐름도이다.

제안하는 동영상 기반 장애물 추적 방법은 데이터 수집부를 통해 객체 감지 및 추적을 위한 동영상을 수집하고, 상기 동영상을 모델 훈련을 위한 이미지로 변환하는 단계(110), 객체 인식부를 통해 딥 러닝 모델을 사용하여 상기 이미지로부터 객체를 인식하는 단계(120) 및 객체 추적부를 통해 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적하는 단계(130)를 포함한다.

단계(110)에서, 데이터 수집부를 통해 객체 감지 및 추적을 위한 동영상을 수집하고, 상기 동영상을 모델 훈련을 위한 이미지로 변환한다.

본 발명의 실시예에서는 객체 감지 및 추적을 위한 데이터셋으로 SMD[1]를 사용했다. 싱가포르 수로에서 밤낮으로 촬영한 동영상인 SMD가 훈련 및 테스트 데이터로 활용되었다. 동영상은 세 가지 범주로 나눌 수 있다. 육상에 탑재된 카메라로 촬영한 육상 영상 40편, 이동하는 선박에 탑재된 카메라로 촬영한 선상 영상 11편, 근적외선 영상 30편 등이다. SMD는 동영상 외에도 개별 프레임에 객체 정보가 포함된 실측 자료(ground truth) 레이블 파일을 제공한다. 이미지에서 볼 수 있는 물체의 10가지 분류는 페리, 부표, 선박, 쾌속정, 보트, 카약, 돛단배, 해엄치는 사람, 나는 새/비행기 등이다. 모델 훈련을 위해 실측 자료 레이블 파일을 레이블로 활용한다.

모델을 훈련하기 위한 데이터 입력이 이미지이기 때문에 동영상을 이미지로 변환해야 한다. 본 논문에서 이미지 데이터셋은 5프레임마다 하나의 이미지가 저장되는 방식으로 생성되었다. 연속 프레임은 유사도가 높기 때문에 동일한 형태로 노출된 물체에 대해 과적합이 발생할 수 있으며, 따라서 동영상의 모든 프레임이 선택되지 않는다. 59개 동영상(예를 들어, 육상 동영상 36개, 근적외선 동영상 23개)에서 47,705개의 객체 이미지를 추출했다. SMD에서 추출한 객체의 개수는 표 1과 같다.

<표 1>

	SMD	Crawling	Total
Ferry	2237	0	2237
Buoy	731	1	732
Vessel/ship	34,074	1	34,075
Speed boat	2193	0	2193
Boat	346	2	348
Kayak	865	1	866
Sail boat	616	2	618
Swimming person	0	120	120
Flying bird/plane	231	0	231
Other	6412	0	6412

표 1과 같이 선박 클래스는 레이블이 가장 많은 반면 수영하는 사람의 클래스는 SMD에 없다. 모델을 훈련할 때 레이블이 고르지 않으면 특정 등급에 대한 과적합 또는 편향이 나타날 수 있다. 과적합과 편향을 방지하기 위해 웹 크롤링을 이용하여 부표, 선박, 보트, 카약, 돛단배, 수영하는 사람의 클래스 등 127개의 객체 이미지를 수집하였다. 표 1은 SMD와 웹 크롤링이 클래스별로 수집한 총 47,832개의 객체 수를 보여준다. 표에 나타난 바와 같이 47,832개의 라벨링 대상 중 120개는 수영하는 사람이 차지한다. 모델 훈련 및 검증을 위해 전체 이미지를 각각 7:3으로 분할했다.

단계(120)에서, 객체 인식부를 통해 딥 러닝 모델을 사용하여 상기 이미지로부터 객체를 인식한다.

본 발명의 실시예에 따르면, BOS(Bag of Specials)의 데이터 확대 기법 중 모자이크 기법을 사용하여 복수의 개별 이미지를 경계 상자와 결합해 하나의 이미지를 생성하고, 하나의 입력만으로 복수의 이미지를 학습한다. 또

한, k-평균 알고리즘을 사용하여 새로운 앵커 크기를 생성하고, 사전 훈련된 컨볼루션 가중치를 이용하여 딥 러닝 모델 훈련을 수행한다.

[0030] 단계(130)에서, 객체 추적부를 통해 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적한다.

[0031] 본 발명의 실시예에 따르면, 상기 인식된 객체가 프레임 아웃되고 다시 프레임 인 되었을 때 동일한 객체로 식별되고, 복수의 객체가 겹쳐지거나 객체가 가려진 경우 해당 객체가 다시 나타날 때 동일한 객체로 식별되도록 동영상의 각 프레임을 감지하고 추적한다.

[0032] 본 발명의 실시예에 따르면, 현재 프레임의 객체의 경계 상자와 모든 이전 프레임의 객체의 경계 상자를 이용하여 IoU(intersection over union) 값을 계산한 후, 계산된 IoU 값에 따라 IoU 매칭을 수행한다.

[0033] 본 발명의 실시예에 따르면, ORB(oriented FAST and rotated BRIEF) 기반 특징 매칭을 수행하고 객체 크기 및 객체 간 특징을 비교하여 객체 유사성을 결정한다.

[0035] 도 2는 본 발명의 일 실시예에 따른 동영상 기반 장애물 추적 시스템의 구성을 나타내는 도면이다.

[0036] 본 실시예에 따른 동영상 기반 장애물 추적 시스템(200)은 프로세서(210), 버스(220), 네트워크 인터페이스(230), 메모리(240) 및 데이터베이스(250)를 포함할 수 있다. 메모리(240)는 운영체제(241) 및 동영상 기반 장애물 추적 루틴(242)을 포함할 수 있다. 프로세서(210)는 데이터 수집부(211), 객체 인식부(212) 및 객체 추적부(213)를 포함할 수 있다. 다른 실시예들에서 동영상 기반 장애물 추적 시스템(200)은 도 2의 구성요소들보다 더 많은 구성요소들을 포함할 수도 있다. 그러나, 대부분의 종래기술적 구성요소들을 명확하게 도시할 필요성은 없다. 예를 들어, 동영상 기반 장애물 추적 시스템(200)은 디스플레이나 트랜시버(transceiver)와 같은 다른 구성요소들을 포함할 수도 있다.

[0037] 메모리(240)는 컴퓨터에서 판독 가능한 기록 매체로서, RAM(random access memory), ROM(read only memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(permanent mass storage device)를 포함할 수 있다. 또한, 메모리(240)에는 운영체제(241)와 SNS를 통한 개인간 거래에서 상대방의 신뢰도 측정 루틴(242)을 위한 프로그램 코드가 저장될 수 있다. 이러한 소프트웨어 구성요소들은 드라이브 메커니즘(drive mechanism, 미도시)을 이용하여 메모리(240)와는 별도의 컴퓨터에서 판독 가능한 기록 매체로부터 로딩될 수 있다. 이러한 별도의 컴퓨터에서 판독 가능한 기록 매체는 플로피 드라이브, 디스크, 테이프, DVD/CD-ROM 드라이브, 메모리 카드 등의 컴퓨터에서 판독 가능한 기록 매체(미도시)를 포함할 수 있다. 다른 실시예에서 소프트웨어 구성요소들은 컴퓨터에서 판독 가능한 기록 매체가 아닌 네트워크 인터페이스(230)를 통해 메모리(240)에 로딩될 수도 있다.

[0038] 버스(220)는 동영상 기반 장애물 추적 시스템(200)의 구성요소들간의 통신 및 데이터 전송을 가능하게 할 수 있다. 버스(220)는 고속 시리얼 버스(high-speed serial bus), 병렬 버스(parallel bus), SAN(Storage Area Network) 및/또는 다른 적절한 통신 기술을 이용하여 구성될 수 있다.

[0039] 네트워크 인터페이스(230)는 동영상 기반 장애물 추적 시스템(200)을 컴퓨터 네트워크에 연결하기 위한 컴퓨터 하드웨어 구성요소일 수 있다. 네트워크 인터페이스(230)는 동영상 기반 장애물 추적 시스템(200)을 무선 또는 유선 커넥션을 통해 컴퓨터 네트워크에 연결시킬 수 있다.

[0040] 데이터베이스(250)는 동영상 기반 장애물 추적을 위해 필요한 모든 정보를 저장 및 유지하는 역할을 할 수 있다. 도 2에서는 동영상 기반 장애물 추적 시스템(200)의 내부에 데이터베이스(250)를 구축하여 포함하는 것으로 도시하고 있으나, 이에 한정되는 것은 아니며 시스템 구현 방식이나 환경 등에 따라 생략될 수 있고 혹은 전체 또는 일부의 데이터베이스가 별개의 다른 시스템 상에 구축된 외부 데이터베이스로서 존재하는 것 또한 가능하다.

[0041] 프로세서(210)는 기본적인 산술, 로직 및 동영상 기반 장애물 추적 시스템(200)의 입출력 연산을 수행함으로써, 컴퓨터 프로그램의 명령을 처리하도록 구성될 수 있다. 명령은 메모리(240) 또는 네트워크 인터페이스(230)에 의해, 그리고 버스(220)를 통해 프로세서(210)로 제공될 수 있다. 프로세서(210)는 데이터 수집부(211), 객체 인식부(212) 및 객체 추적부(213)를 위한 프로그램 코드를 실행하도록 구성될 수 있다. 이러한 프로그램 코드는 메모리(240)와 같은 기록 장치에 저장될 수 있다.

[0042] 데이터 수집부(211), 객체 인식부(212) 및 객체 추적부(213)는 도 1의 단계들(110~130)을 수행하기 위해 구성될 수 있다.

- [0043] 동영상 기반 장애물 추적 시스템(200)은 데이터 수집부(211), 객체 인식부(212) 및 객체 추적부(213)를 포함할 수 있다.
- [0044] 데이터 수집부(211) 객체 감지 및 추적을 위한 동영상을 수집하고, 상기 동영상을 모델 훈련을 위한 이미지로 변환한다.
- [0045] 객체 인식부(212)는 딥 러닝 모델을 사용하여 상기 이미지로부터 객체를 인식한다.
- [0046] 본 발명의 실시예에 따르면, BOS(Bag of Specials)의 데이터 확대 기법 중 모자이크 기법을 사용하여 복수의 개별 이미지를 경계 상자와 결합해 하나의 이미지를 생성하고, 하나의 입력만으로 복수의 이미지를 학습한다. 또한, k-평균 알고리즘을 사용하여 새로운 앵커 크기를 생성하고, 사전 훈련된 컨볼루션 가중치를 이용하여 딥 러닝 모델 훈련을 수행한다.
- [0047] 객체 추적부(213)는 동영상의 각 프레임을 감지하고 상기 인식된 객체에 대하여 추적한다.
- [0048] 본 발명의 실시예에 따르면, 객체 추적부(213)는 상기 인식된 객체가 프레임 아웃되고 다시 프레임 인 되었을 때 동일한 객체로 식별되고, 복수의 객체가 겹쳐지거나 객체가 가려진 경우 해당 객체가 다시 나타날 때 동일한 객체로 식별되도록 동영상의 각 프레임을 감지하고 추적한다.
- [0049] 본 발명의 실시예에 따르면, 객체 추적부(213)는 현재 프레임의 객체의 경계 상자와 모든 이전 프레임의 객체의 경계 상자를 이용하여 IoU(intersection over union) 값을 계산한 후, 계산된 IoU 값에 따라 IoU 매칭을 수행한다.
- [0050] 본 발명의 실시예에 따르면, 객체 추적부(213)는 ORB(oriented FAST and rotated BRIEF) 기반 특징 매칭을 수행하고 객체 크기 및 객체 간 특징을 비교하여 객체 유사성을 결정한다.
- [0052] 도 3은 본 발명의 일 실시예에 따른 객체 인식을 위한 딥러닝 모델의 예시를 나타내는 도면이다.
- [0053] 본 발명의 실시예에 따르면, 네 가지 딥 러닝 모델을 사용하여 바다에 있는 10가지 유형의 물체를 실시간으로 높은 정확도로 인식할 수 있는 딥 러닝 모델을 찾는다. 표 2는 네 가지 인식 모델의 구성을 보여준다.
- [0054] <표 2>

	Model 1	Model 2	Model 3	Model 4
Model structure	22 convolutions 5 poolings	23 convolutions 5 poolings	CSPDarknet53, SPP & PAN, YOLOv3	CSPDarknet53, SPP & PAN, YOLOv3
Number of anchors	5	9	9	9
New anchor size	No	Yes	No	Yes
Mosaic	No	No	No	Yes
Transfer learning	No	Yes	No	Yes

- [0055]
- [0056] 모델 1은 22개의 컨볼루션 계층과 5개의 최대 풀링 계층이 있는 YOLOv2 모델이다. 모델 2는 성능 향상을 위해 모델 1을 수정한 버전이다. 다음은 모델 1과 모델 2의 차이점이다. 먼저, 13번째 컨볼루션 계층 이후에 원래의 YOLOv2 신경망에 1×1 컨볼루션 계층을 삽입한다.
- [0057] 도 3은 모델 2의 구조를 나타낸 것이다. 도 3에서 보는 바와 같이, 추가적인 컨볼루션 계층(310)은 상위 계층의 특징 맵과 하위 계층의 특징 맵을 혼합하는 지나가는 계층으로 작용함으로써 작은 객체 감지 정확도를 증가시킨다. 둘째, 셀당 예상 경계 상자(다시 말해, 앵커)의 수가 5개에서 9개로 증가하여 회수율은 높았으나 mAP는 유의하게 낮았다. 리콜과 mAP는 트레이드오프 관계를 가지므로 앵커 수에 따라 앵커 수가 증가하면 경계 상자가 더 많이 발견될수록 리콜이 증가하지만 mAP(평균 정밀도)는 감소한다. 셋째, k-평균 알고리즘을 사용하여 데이터셋에 맞게 앵커 크기를 조정했다. 마지막으로, 전이 학습은 더 빠른 학습과 더 나은 예측을 돕기 위해 사용된다. imageNet 데이터셋을 사용하여 사전 훈련된 컨볼루션 가중치인 Darknet19448.conv.23을 사용하여 모델을 훈련한다.
- [0058] 모델 3의 신경망 구조는 백본을 위한 CSPDarknet53, 넥을 위한 SPP(Spatial Pyramid Pooling)와 PAN(Path Aggregation Network), 헤드를 위한 YOLOv3이다. 성능을 향상시키기 위해 모델 4는 YOLOv4 신경망 구조에 여러 가지 기술을 사용한다. 첫째, BOS의 데이터 확대 기법 중 하나인 모자이크 기법을 사용한다. 모자이크 기법은 4

개의 개별 이미지를 경계 상자와 결합해 하나의 이미지를 만들고, 하나의 입력만으로 4개의 이미지를 학습할 수 있다. 이미지가 병합되면 물체의 크기가 줄어들기 때문에 학습은 자연스럽게 수많은 작은 것들과 마주치게 되고, 그 결과 작은 물체 감지 성능을 향상시킬 수 있다. 또한 앵커 수가 9개로 유지됨에도 불구하고 본 발명에서 활용한 데이터셋을 기반으로 모델 2에서와 같이 k-평균 알고리즘을 사용하여 새로운 앵커 크기를 생성하였다. 마지막으로, Microsoft COCO 데이터셋을 사용하여 사전 훈련된 컨볼루션 가중치인 yolov4.conv.137 이 모델을 훈련하는 데 사용된다. 본 발명의 실시예에서는 객체 추적을 위해 모델 4를 감지 모델로 사용한다.

[0060] 도 4는 본 발명의 일 실시예에 따른 객체 추적 알고리즘을 나타내는 도면이다.

[0061] 본 발명의 실시예에 따르면, 움직이는 선박에 의해 발생하는 움직이는 동영상의 객체를 추적하려면 다음 요건을 충족해야 한다. 첫째, 어떤 객체가 프레임 아웃되고 다시 프레임 인 되었을 때 동일한 객체로 식별되어야 한다. 둘째, 많은 객체가 겹쳐지고 객체가 가려질 때, 그것이 다시 나타날 때는 동일한 객체로 식별되어야 한다.

[0062] 이러한 요구 사항을 충족하기 위해 동영상의 각 프레임을 감지하고 추적한다. 상술된 모델 4는 인식을 위해 사용되며 경계 상자의 번호, 유형 및 좌표를 반환한다. 프레임에 둘 이상의 경계 상자가 포함된 경우 도 4에 도시된 알고리즘 1에 설명된 추적 알고리즘이 사용된다.

[0063] 각 경계 상자에 대해 IoU_matching이 수행된다. IoU_matching은 현재 프레임의 객체와 모든 이전 프레임의 객체의 IoU(intersection over union) 값을 계산한다. 그런 다음 IoU 값이 최대인 객체를 결정하고 IoU 값이 임계값보다 높은 객체는 overlapIDList에 저장된다. IoU_matching 결과값이 1이면 프레임에 객체가 하나 존재하므로 이전 프레임의 최대 IoU 값을 가진 객체와 동일하다고 판단된다. IoU_matching 결과가 0이거나 2보다 크면 ORB_size_matching이 수행된다.

[0065] 도 5는 본 발명의 일 실시예에 따른 IoU_matching 알고리즘을 나타내는 도면이다.

[0066] 본 발명의 실시예에 따른 IoU_matching에서, 현재 프레임 번호가 n이라고 가정하면, 비교 대상이 되는 선행 프레임은 n-1번째 프레임이다. IoU 값은 현재 프레임에 있는 객체의 경계 상자와 이전 프레임의 모든 경계 상자를 사용하여 계산된다. 객체 간의 중첩 영역은 0에서 1 사이의 IoU로 측정된다. 최대 IoU 값은 두 객체가 가장 가까운 위치에 있음을 나타낸다. 객체가 이전 프레임에서 인식되고 IoU 값이 0.2 이상인 경우 객체와 IoU 값 쌍이 overlapIDList에 저장된다. overlapIDList는 IoU 값이 임계값보다 큰 객체 및 IoU 값 쌍의 목록이다. 여러 객체가 겹칠 경우 overlapIDList는 하나의 객체에 대해 여러 객체 및 IoU 값 쌍을 저장한다. 계산된 IoU 값을 최대 IoU 값과 비교하여 최대 IoU 값이 업데이트된다. overlapIDList의 길이가 0이 아니므로 길이가 반환된다. overlapIDList 길이가 0이면 경계 상자가 반복적으로 있는 n-2번째 프레임에서 IoU 값이 계산된다.

[0068] 도 6은 본 발명의 일 실시예에 따른 IoU_matching 알고리즘을 나타내는 도면이다.

[0069] 본 발명의 실시예에 따른 ORB_size_matching은 ORB(oriented FAST and rotated BRIEF) 기반 특징 매칭을 수행하고 객체 크기를 비교한다. 도 6에 도시된 알고리즘 3은 ORB_size_matching을 설명한다. n - 1 및 n - 2 프레임에서 overlapIDList 길이가 0이면 IoU 값이 임계값보다 큰 객체는 없다. 그러나 빠른 이동으로 인해 IoU가 낮을 수 있기 때문에 현재 추적 중인 객체가 저장된 IDList의 객체를 사용하여 ORB 매칭을 수행한다. overlapIDList의 길이가 0이 아니라면 IoU 값이 임계값보다 큰 객체가 두 개 이상 있으므로 ORB 매칭은 overlapIDList의 모든 객체에 대해 수행된다. ORB 매칭은 객체 간 특징을 비교하여 객체 유사성을 결정한다. ORB 일치 유사성이 0.3보다 크면 크기 일치에 대한 비율이 계산된다. 비율이 0.9와 1.1 사이인 경우 입력 객체는 유사성이 가장 높은 객체와 동일하다. 비율이 0.9와 1.1 사이가 아니고 YOLO가 감지한 객체 정확도가 0.6 이상이면 유사점이 많지만 크기가 다른 새로운 객체로 간주된다. 또한 ORB 유사도가 0.3 미만이고 YOLO가 감지한 객체 정확도가 0.6 이상인 경우에도 새로운 객체로 인식된다.

[0070] 본 발명에서는 추적 알고리즘이 궤적을 예측하는 데 활용될 수 있을 것으로 예상된다. 현재 프레임의 객체가 이전 프레임의 객체와 동일한 경우 추적 객체의 좌표가 예측 목록에 저장된다. 앞서 언급한 바와 같이, 제안된 감지 모델은 추적 대상의 좌표 정보를 얻는다. [2-5]의 궤적 예측 접근법은 좌표 정보를 사용하여 궤적을 예측한다.

- [0072] 도 7은 본 발명의 일 실시예에 따른 영상 기반 장애물 추적 알고리즘과 DeepSORT를 이용한 선박 추적 결과를 비교하기 위한 도면이다.
- [0073] 도 7(a)는 70번째 프레임에서 제안된 추적 알고리즘, 도 7(b)는 70번째 프레임의 DeepSORT, 도 7(c)는 75번째 프레임에서 제안된 추적 알고리즘, 도 7(d)는 75번째 프레임의 DeepSORT, 도 7(e)는 80번째 프레임에서 제안된 추적 알고리즘, 도 7(f)는 80번째 프레임의 DeepSORT, 도 7(g)는 95번째 프레임에서 제안된 추적 알고리즘, 도 7(h)는 95번째 프레임의 DeepSORT의 결과를 각각 나타낸다.
- [0074] 본 발명의 실시예에 따른 테스트를 사용하여 제안하는 영상 기반 장애물 추적 알고리즘과 DeepSORT 간의 추적 정밀도 차이를 설명한다. 70, 75, 80, 95번째 프레임에서 제안하는 영상 기반 장애물 추적 알고리즘과 DeepSORT를 이용한 선박 추적 결과를 도 7에 도시하였다. 제안하는 영상 기반 장애물 추적 알고리즘과 DeepSORT는 70번째 프레임에서 각각 Tracking ID 1과 2를 할당하는데 이는 선박의 상하 움직임에 따라 추적된 선박이 75번째 프레임에서 프레임 경계에 위치하지만 Tracking ID는 동일하게 유지되기 때문이다.
- [0075] 선박은 80번째 프레임 이후 시야에서 사라졌다. 95번째 프레임에서 선박이 다시 나타나고 제안하는 영상 기반 장애물 추적 알고리즘은 이를 동일한 선박으로 인식하고 이전과 동일한 추적 ID를 제공한다. 반면 DeepSORT는 이를 새 선박으로 취급하고 새 추적 ID를 할당한다. DeepSORT는 객체의 위치를 기준으로 동일 여부를 판단하지만, 제안하는 영상 기반 장애물 추적 알고리즘은 객체의 특징을 기준으로 동일 여부를 판단한다. 결과적으로 SMD의 온보드 비디오와 같이 움직임이 많은 비디오의 경우 제안하는 영상 기반 장애물 추적 알고리즘의 성능이 DeepSORT를 능가한다.
- [0077] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 컨트롤러, 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 애플리케이션을 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 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.
- [0078] 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(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.
- [0079] 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. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를

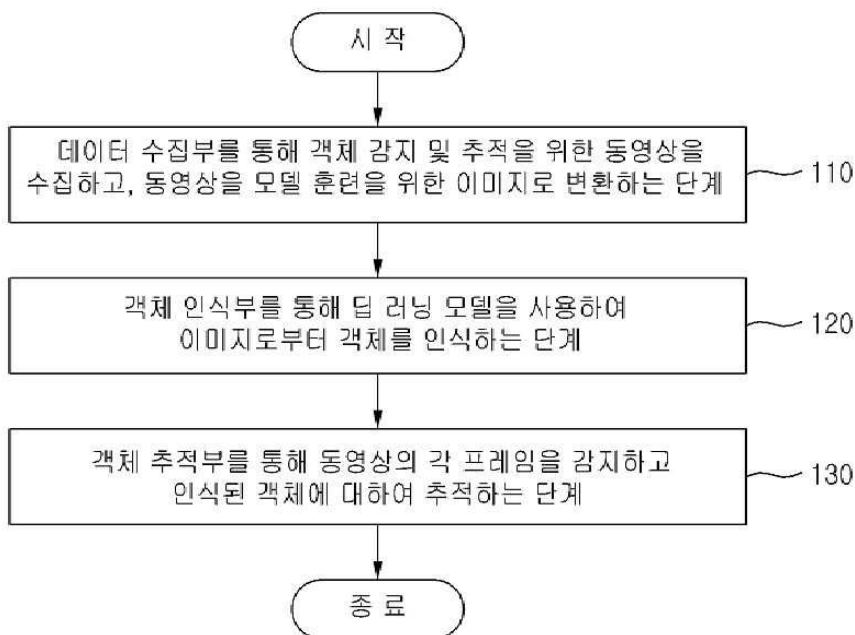
포함한다.

[0080] 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.

[0081] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

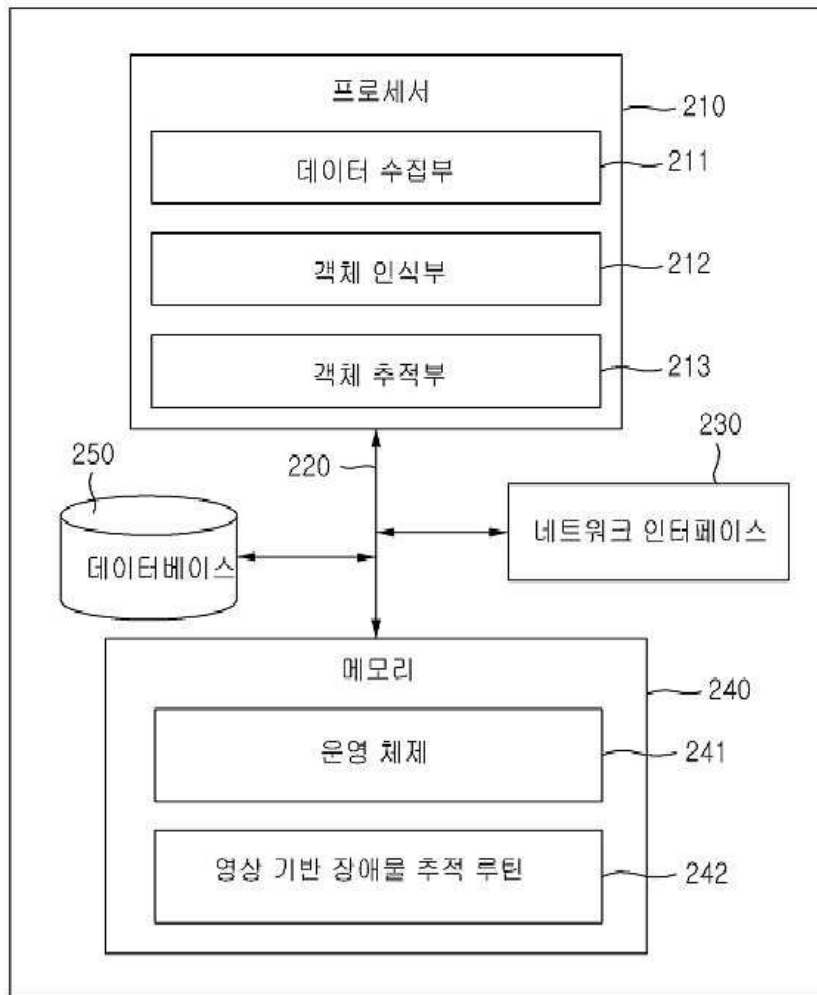
도면

도면1

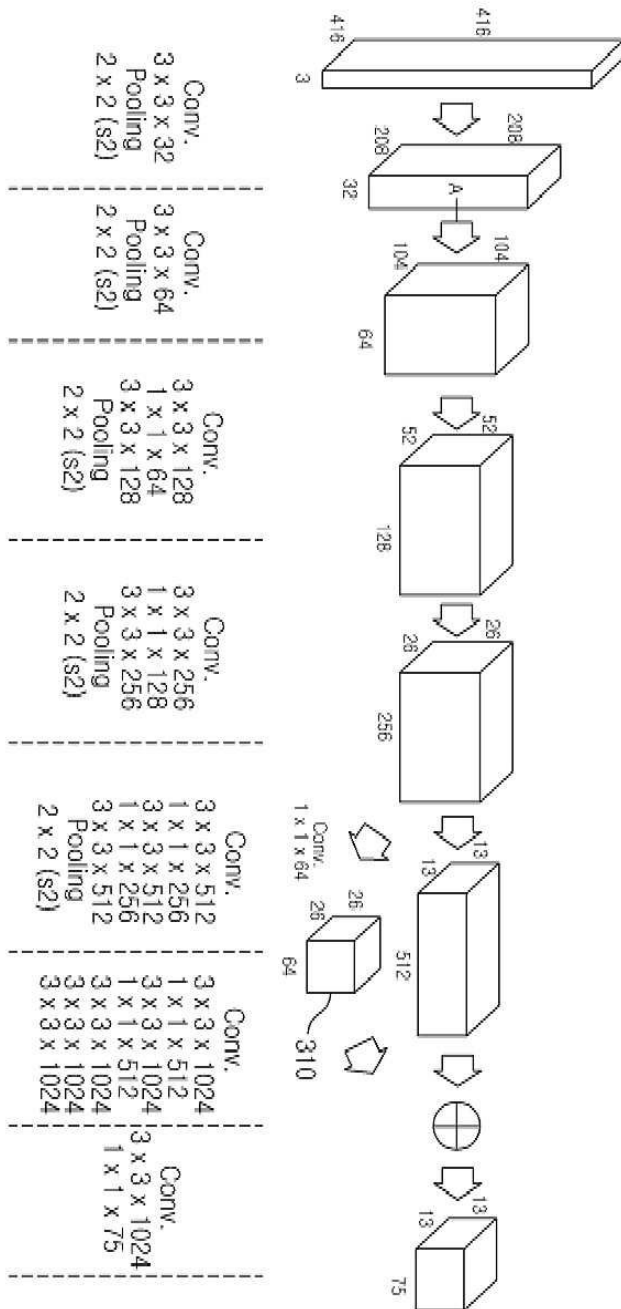


도면2

200



도면3



도면4

Algorithm 1 Tracking

```

1: for each bounding box do
2:   result = IoU_matching()
3:   if result == 1 then
4:     detected object is identical to the object
       with the highest IoU among the objects
       in the previous frame
5:   else
6:     ORB_size_matching()
7:   end if
8: end for

```

도면5

Algorithm 2 IoU_matching()

```

1: n: frame number currently in use
2: prev: previous frame number to be compared
3: Framen: nth frame
4: overlapIDList: list of objects having IoU values above the threshold
5:
6: for prev = n - 1 to n - 2 do
7:   for each bounding box of objects recognized in the Frameprev do calculates
       IoU value
8:     if IoU value ≥ 0.2 then
9:       the object is stored in overlapIDList
10:      if IoU value > maximum IoU then
11:        maximum IoU = IoU value
12:      end if
13:    end if
14:  end for
15:  if the length of overlapIDList is not 0 then
16:    return the length of overlapIDList
17:  end if
18: end for
19:
20: return the length of overlapIDList

```

도면6

Algorithm 3 ORB_size_matching()

```

1: IDList: list of the objects that are currently being tracked
2:
3: if the length of overlapIDList == 0 then
4:   List = IDList
5: else
6:   List = overlapIDList
7: end if
8:
9: The input object and all of the objects in the List are matched using ORB
10:
11: if the object with the highest similarity is recognized in the previous frame then
12:   if similarity > 0.3 then
13:     ratio =  $\frac{\text{the size of } \text{f bounding box of the input object}}{\text{the size of } \text{f bounding box of the highest accuracy object}}$ 
14:     if 0.9 <= ratio <= 1.1 then
15:       the input object is identical to the object with the highest matching accuracy
16:     else
17:       if YOLO's object detection accuracy >= 0.6 then
18:         the input object is recognized as new object
19:       end if
20:     end if
21:   else
22:     if YOLO's object detection accuracy >= 0.6 then
23:       the input object is recognized as new object
24:     end if
25:   end if
26: end if

```

도면7

