applied sciences

MDPI

Article

Multiple Object Tracking in Robotic Applications: Trends and Challenges

Abdalla Gad [1,†], Tasnim Basmaji [1,†], Maha Yaghi [1], Huda Alheeh [1,2], Mohammad Alkhedher [3] and Mohammed Ghazal [1,*]

[1] Electrical, Computer and Biomedical Engineering Department, College of Engineering, Abu Dhabi University, Abu Dhabi 59911, United Arab Emirates
[2] Network and Communication Engineering Department, College of Engineering, Al Ain University, Abu Dhabi 112612, United Arab Emirates
[3] Mechanical and Industrial Engineering Department, College of Engineering, Abu Dhabi University, Abu Dhabi 59911, United Arab Emirates
* Correspondence: mohammed.ghazal@adu.ac.ae
† These authors contributed equally to this work.

Abstract: The recent advancement in autonomous robotics is directed toward designing a reliable system that can detect and track multiple objects in the surrounding environment for navigation and guidance purposes. This paper aims to survey the recent development in this area and present the latest trends that tackle the challenges of multiple object tracking, such as heavy occlusion, dynamic background, and illumination changes. Our research includes Multiple Object Tracking (MOT) methods incorporating the multiple inputs that can be perceived from sensors such as cameras and Light Detection and Ranging (LIDAR). In addition, a summary of the tracking techniques, such as data association and occlusion handling, is detailed to define the general framework that the literature employs. We also provide an overview of the metrics and the most common benchmark datasets, including Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI), MOTChallenges, and University at Albany DEtection and TRACking (UA-DETRAC), that are used to train and evaluate the performance of MOT. At the end of this paper, we discuss the results gathered from the articles that introduced the methods. Based on our analysis, deep learning has introduced significant value to the MOT techniques in recent research, resulting in high accuracy while maintaining real-time processing.

Keywords: multiple object tracking; MOT; self-driving; autonomous vehicle; autonomous navigation; SLAM; KITTI; MOTChallenges; MOT15; MOT16; MOT17; UA_DETRAC

check for updates

Citation: Gad, A.; Basmaji, T.; Yaghi, M.; Alheeh, H.; Alkhedher, M.; Ghazal, M. Multiple Object Tracking in Robotic Applications: Trends and Challenges. *Appl. Sci.* **2022**, *12*, 9408. https://doi.org/10.3390/app12199408

Academic Editor: Luis Gracia

Received: 18 August 2022
Accepted: 14 September 2022
Published: 20 September 2022

1. Introduction

Integrating computer vision and deep learning-based systems in the robotics field has led to a massive leap in the advancement of autonomous feature. The utilization of different sensors, such as cameras and LIDARs, and the progress established by the recent research on processing this data, have introduced multiple object tracking techniques in autonomous driving and robotics navigation systems. Multiple object tracking has been one of the most challenging topics researched through computer vision techniques. The reasons behind this are due to: (1) multiple object tracking (MOT), an essential tool that can be used in enhancing security and automating robotics navigation, and (2) occlusion, which is the main obstacle standing in the path of reaching a reliable accuracy and one issue that is difficult to tackle. In this paper, we aim to survey the different approaches of MOT introduced recently in autonomous robotics.

Much research has been done to enhance the performance of tracking in SLAM applications [1,2]. It is simply difficult to navigate through the environment while the positions of the robot itself and other objects in the surrounding are neglected. Tracking is used to estimate the relative location of the robot to other components in the environment. The most challenging part of this process is the existence of highly dynamic objects [3] such as people or vehicles. SLAM-based autonomous navigation in robots has been regarded as essential in development and research primarily because of its potential in many aspects. One example is the autonomous wheelchair systems reviewed in Refs. [4,5]. The authors in Ref. [6] provided a survey of the mobile devices that assist people with disability. Autonomous driving can cause a reduction in the number of accidents that occur due to fatigue and distractions [7]. Although that might be the case, the public opinion about autonomous vehicles is hesitant about whether to consider the technology trustworthy. Providing awareness and understanding of the capabilities of the sensors in autonomous vehicles to the drivers is vital to reaching the proper employment of the technology in our daily lives. These sensors should not be disregarded or become entirely dependable on them [8]. The approach introduced in Ref. [9] aims to reduce the risks firefighters encounter by deploying a team of UAVs with an MOT system to track wildfires and control the situation. The authors in Ref. [10] employ MOT to guide a swarm of drones and control them. Similarly, MOT is utilized with UAV for collision avoidance in Ref. [11]

As has been discussed in Refs. [12–14], the general framework for MOT is shown in Figure 1. The input frame is subjected to an object detection algorithm. Then, the detections from the current frame and the previous frames are used to match the similar trajectories either by motion, appearance, and/or other features. This process would generate tracks presenting the objects through the sequence of frames. Some data association between multiple frames is applied to track an object through multiple frames. A reliable MOT system should be able to handle the new tracks as well as the lost ones. Here, the occlusion issue is where the lost tracks reappear again because they did not move out of the sensor's view but were hidden by other objects.

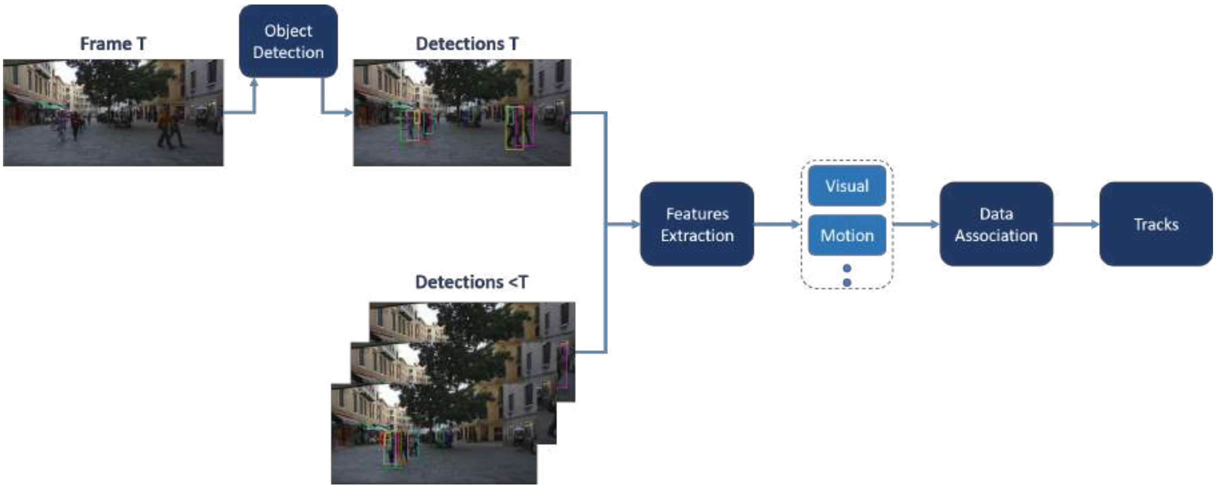

Figure 1. General framework of MOT systems. Visual and motion features of the detected objects at frame T are extracted and compared to those detected from previous frames. A robust data association algorithm would be able to match the features of the same objects. The final output of the system would be tracked with unique IDs identifying the multiple objects detected and tracked over the multiple frames.

1.1. Challenges

To effectively perform object tracking, one must develop a robust and efficient model that the users can effectively use. This section aims to provide a comprehensive overview of the various challenges facing developing and optimizing such models.

The first challenge a model must face is the quality of the input video [15]. If the model cannot process the video properly, it will require additional work to convert it into a clear form so that it can be used to detect objects. The classification system must first identify the objects that fall under a particular class. It then gives them IDs based on their shape and form, which raises the issue that objects of this class come in varying shapes and sizes [16]. After the objects are detected, they must be assigned IDs with a bounding box to identify them to ensure that the model can identify multiple similar objects in the coverage area. The next challenge is to identify the objects that are moving in the ROI of the camera. This phenomenon can cause the classification system to misclassify the objects or even identify them as new ones. Aside from the quality of the video input, other factors that affect the classification of objects are also considered. For instance, the illumination conditions can significantly influence the model's accuracy [12–14,16]. The model may not be able to detect objects that are blunt with the environment or have background conditions. It also needs to be able to identify them at varying speeds. One of the most challenging issues in object tracking is the Occlusion issue, where the object movement gets interrupted by other objects in the scene [12–14,16]. It can be caused by various factors such as natural conditions or the object's movement out of the camera's ROI. Another reason is that other objects might block the visual of the object if the object is in the camera's ROI. Therefore, the system must be trained to identify and track the objects in motion. It also needs to be able to re-identify the IDs of the captured images with the same ones already used by the cameras. Figure 2 shows an example of the occlusion issue. The yellow arrow follows one of the tracks that maintains its ID after experiencing full occlusion. The problem of occlusion is minimized in bird-eye view tracking [17]. However, other challenges arise, such as the low resolution of objects and misclassifications. Another obstacle is related to onboard tracking in self-driving applications. The issue is that the tracking process needs to be quick and accurate for an efficient assistant driving system. The FPS is one of the essential factors determining the tracking quality in this case [18].

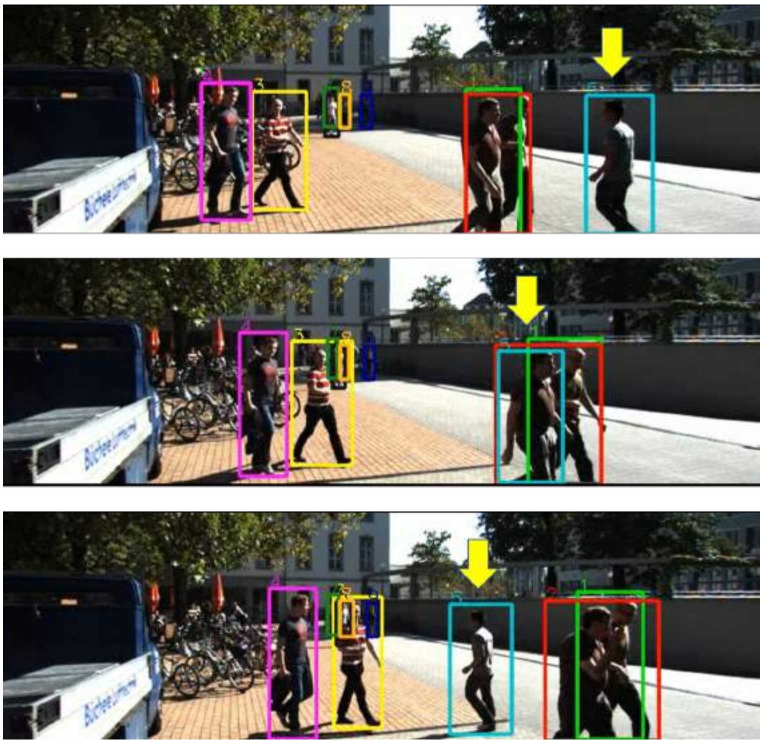

Figure 2. Preserving the ID during full occlusion. The frames are obtained from the MOT15 dataset [19]. The yellow arrow is pointing towards a track (**top image**) that experiences full occlusion (**middle image**). The objective of the MOT system is to preserve the ID of the track (**bottom image**) and matches the previously detected object with the reappeared one.

1.2. Related Work

The authors in Ref. [20] provided a summary of the techniques developed for SLAM-MOT (combination of SLAM and MOT systems) that utilize the dynamic features to construct 3D object tracking and multi-motion segmentation systems. In Ref. [20], 3D tracking of dynamic objects techniques was categorized into trajectory triangulation, particle filter, and factorization-based approaches. The authors in Ref. [20] discussed the data fusion problem in autonomous vehicles. The perception of data from different sensors such as RGB cameras, LIDAR, and depth cameras provides more knowledge and understanding of the surrounding environment and increases the navigation system's robustness. In Ref. [20], a survey is conducted on the techniques used for SLAM in autonomous driving applications and the limitations of the current research. The evaluation in this paper was done on the KITTI dataset.

The authors in Ref. [12] categorized the MOT approaches into three groups. The first is the initialization method, which defines whether the tracking would be detection-based or detection-free. Detection-based tracking or tracking-by-detection is the most common, where a detected object is connected to its trajectories from future frames. This connection can be applied by calculating the similarity based on appearance or motion. Detection-free tracking is where a set of objects is manually localized in the first frame and tracked through the future. This is not optimal in case new objects appear and is rarely applied. The second is based on the processing mode, either online or offline tracking. Online tracking is where objects are detected and tracked in real time. This is more optimal in the case of autonomous driving applications as offline tracking is where a batch of frames are processed at a low FPS. The final one is the type of output where it can be stochastic, in which the tracking varies at different running times, or deterministic, in which the tracking is constant. They would further define the components that are included in the MOT system. Appearance model is used to extract spatial information from the detections and then calculate their similarity. The visual features and representations extracted can be defined either locally or regionally.

Motion models are used to predict the future location of the detected object and hence, reduce the inspection area. A good model would have a good estimation after a certain number of frames as its parameters are tuned towards learning how the object moves. Linear (constant velocity) and non-linear are the two types of motion models. Although there has been a rapid advancement in multiple object detection and tracking for autonomous driving, it is still processing only a few objects. It will be a giant leap forward to have a system capable of tracking all types of objects in real time. This can be achieved by generating a great deal of data that tackles the problem at different perceptions such as camera, LIDAR, ultrasonic, etc. [21]. The issues related to the full deployment of MOT in autonomous vehicles lie in that its reliability heavily depends on many parameters, such as the camera view and the type of background (dynamic or static). This leads to difficulty in being entirely trustful towards MOT in different real scenarios and environments [12]. Tracking pedestrians is a far more difficult task than tracking vehicles whose motion is bounded by the road compared to the motion of people, which is very random and challenging for the system to learn. Another issue is the occlusion, which leads to high fragmentation and ID switches due to losing and re-initializing tracks every time they get lost. There have been very few systems that comprehensively tackle the problem, which leaves a huge space for improvements [22].

In Ref. [16], the tracking algorithms are categorized into two groups. The first is matching-based, which defines how features, such as appearance and motion, are first extracted and used to measure the similarity in the future frames. The second is filtering-based tracking, where Kalman and Particle filters are discussed. The authors in Ref. [13] comprehensively surveyed the deep learning-based methods for MOT. They also provided an overview of the data in MOTChallenges and the type of conditions included. An evaluation of the performance of some methods on this dataset is then listed. In Ref. [14], the deep learning-based methods for MOT were also reviewed. Similarly, the authors also provided an overview of the benchmark datasets, including MOTChallenges and KITTI, and presented the performance of some methods.

In Ref. [23], the vision-based methods used to detect and track vehicles at road intersections were discussed. The authors categorized those methods depending on the sensors used and the approach carried out for detection and tracking. On the other hand, the authors in Ref. [24] presented methods introducing vehicle detection and tracking in urban areas, and an evaluation was then discussed. UAVs' role in civil applications, including surveillance, have been surveyed in Refs. [25,26]. The authors discussed the characteristics and roles of UAVs in traffic flow monitoring. However, there have not been many contributions to vehicle tracking methods using an UAV.

The authors in Refs. [7,8], provided a detailed overview of the types of sensors mounted on autonomous vehicles, such as LIDAR, Ultrasonic, and cameras, for data perception. They also surveyed the current advancement in the autonomous driving field commercially and the type of technology associated with that. In Ref. [21], the authors studied the role of deep learning in autonomous driving including perception and path planning. In addition to deep learning approaches, a general review was introduced in Ref. [27]. In Ref. [28], the methods used to extract and match information from multiple sensors used for perception were reviewed. They also discussed how data association could be an issue in using multiple sensors to achieve reliable multiple object tracking. The authors in Ref. [29], surveyed the methods that utilize LIDAR in data perception and grouped the performance results on the KITTI dataset. Ref. [22] provided a comprehensive overview of the KITTI dataset's role in the autonomous driving application. The dataset can be used for training and testing pedestrians, vehicles, cyclists, and other objects that can be found on the road. Moreover, the dataset was extended to lane and road marks detection by Ref. [30].

Although the techniques mentioned above were very thorough in reviewing techniques, we aim in this paper to provide comprehensive research that surveys the techniques associated with autonomous robotics applications, provides an insight into the different tracking methods, gathers and compares the results from the different methods discussed in the paper, and evaluate the current work and find limitations that require future research. Table 1 lists the recent reviews, the year of publication, and the datasets used for comparing MOT methods.

Table 1. Recent reviews and the data used for evaluation and comparison.

Review	Year	Evaluation Dataset
Ciaparrone et al. [13]	2019	MOT 15, 16, 17
Xu et al. [14]	2019	MOT 15, 16
Luo et al. [12]	2022	PETS2009-S2L1
Ours	**2022**	**KITTI, MOT 15, 16, 17, 20 , and UA_DETRAC**

Section 2 discusses the state-of-art methods and techniques introduced by the literature. Section 3 discusses the benchmark datasets and evaluation metrics popularly used by the research for training and testing. Section 4 presents the evaluation results collected from the literature and discussion. Finally, Sections 5 and 6 provide the current study challenges and the future work that is required.

2. Mot Techniques

In this section, we go through the most recent MOT techniques and the common trends being followed for matching tracks across multiple frames.

Table 2 shows a summary of the components used in MOT techniques. It can be observed that the appearance cue is rarely neglected. Motion cue also shows presence a lot. Most approaches depend on deep learning for extracting visual features. CNNs are vital tools that can extract visual features from the tracks and achieve accurate detections of tracks matching [14]. The approaches introduced in Refs. [31,32] use Long Short Term Memory (LSTM) based networks for motion modeling. LSTM networks are considered in MOT for appearance and motion modeling as they can find patterns by efficiently processing the previous frames in addition to the current ones. On the other hand, The authors in Ref. [33] generated histograms from the detections and used them as the appearance features. As for data association, the Hungarian algorithm is common with MOT techniques, such as Refs. [33–36], for associating the current detections with the previous ones, although the performance of these techniques did not show much potential. Deep learning has rarely been utilized for data association. However, the best performing technique on MOT16 and MOT17 datasets relied on a prediction network to validate that the two bounding boxes are related. For occlusion handling, most approaches rely on feeding the history of tracks into the tracking system to validate the lost ones. The tracks absent for a specific number of frames would be considered lost and deleted from the history. This is to avoid processing a massive number of detections and reducing the FPS.

Table 2. Summary of the components used in MOT techniques.

Tracker	Appearance Cue	Motion Cue	Data Association	Mode	Occlusion Handling
Keuper et al. [37]	-	Optical flow	Correlation Co-Clustering	Online	Point trajectories
Fang et al. [38]	fc8 layer of the inception network	Relative center coordinates	Conditional Probability	Online	Track history
Xiang et al. [32]	VGG-16	LSTM network	Metric Learning	Online	Track history
Chu et al. [39]	CNN	-	Reinforcement Learning	Online	SVM Classifier
Zhu et al. [40]	ECO	-	Dual Matching Attention Networks	Online	Spatial attention network
Zhou et al. [41]	ResNet50	Linear Model	Siamese Networks	Online	Track history
Sun et al. [42]	VGG-like network	-	Affinity estimator	-	Track history
Peng et al. [43]	ResNet-50 + FPN	-	Prediction Network	Online	Track history
Wang et al. [44]	FaceNet	-	Multi-Scale TrackletNet	-	Track history
Mahmoudi et al. [34]	CNN	Relative mean velocity and position	Hungarian Algorithm	Online	Track history
Zhou et al. [35]	ResNet-50	Kalman Filter	Hungarian Algorithm	Online	Track history
Lan et al. [45]	Decoder	Relative position	Unary Potential	Near-online	Track history
Zhou et al. [46]	-	CenterTrack	2D displacement prediction	Online	Track history
Karunasekera et al. [33]	Matching histograms	Relative position	Hungarian Algorithm	Online	Grid Structure + Track history
Chu et al. [47]	Siamese-style network	-	R1TA Power Iteration layer	Online	Track history

Table 2. *Cont.*

Tracker	Appearance Cue	Motion Cue	Data Association	Mode	Occlusion Handling
Zhao et al. [36]	CNN + PCA + Correlation Filter	-	Hungarian Algorithm	Online	APCE + IoU
Chen et al. [48]	Faster-RCNN (VGG16)	-	Category Classifier	Online	Track history
Sadeghian et al. [31]	VGG16 + LSTM network	LSTM network	RNN	Online	Track history
Yoon et al. [49]	Encoder	-	Decoder	Online	Track history
Xu et al. [50]	FairMOT	Kalman Filter	Hungarian Algorithm	Online	Track history
Ye et al. [51]	LDAE	Kalman Filter	Cosine Distance	Online	Track History
Wang et al. [52]	Faster-RCNN (ResNet50)	Kalman Filter	Faster RCNN (ResNet50)	Online	Track history
Yu et al. [53]	DLA-34 + GCD + GTE	-	Hungarian Algorithm	Online	Track history
Wang et al. [54]	PCB	Kalman Filter	RTU++	Online	Track history
Gao et al. [55]	ResNet34	-	Depth wise Cross Correlation	Online	Track history
Nasseri et al. [56]	-	Kalman Filter	Normalized IoU	Online	Track history
Zhao et al. [57]	Transformer + Pixel Decoder	Kalman Filter	Cosine Distance	Online	Track history
Aharon et al. [58]	FastReID's SBS-50	Kalman Filter	Cosine Distance	Online	Track history
Seidenschwarz et al. [59]	ResNet50	Linear Model	Histogram Distance	Online	Track history
dai et al. [60]	CNN	-	Hungarian Algorithm	Online	Track history
Zhang et al. [61]	FairMOT	Kalman Filter	IoU + Hungarian Algorithm	Online	Track history
Hyun et al. [62]	FairMOT	-	Sparse Graph + Hungarian	Online	Track history

Table 2. *Cont.*

Tracker	Appearance Cue	Motion Cue	Data Association	Mode	Occlusion Handling
Chen et al. [63]	DETR Network	-	Hungarian Algorithm	Online	Track history
Cao et al. [64]	-	Kalman Filter + non-linear model + Smoothing	Observation Centric Recovery	Online	Track history
Wan et al. [65]	YOLOx	-	Self and Cross Attention Layers	Online	Track history
Du et al. [66]	-	NST Kalman Filter + ResNet50	AFLink	Online	Track history
Zhang et al. [67]	ResNet50	Kalman Filter	Hungarian Algorithm	Online	Track history

The general framework illustrated in Figure 1 is followed by most of the recent MOT techniques. The most common approach for extracting the visual features of an object is by using CNN. VGG-16 is very popular for this application, as in Refs. [32,42,48]. The issue with deploying CNN is the slow computation time due to the high dimensionality output. Zhao et al. [36] tackled this issue by applying PCA followed by a correlation filter for dimensionality reduction to the output of the CNN. An encoder with fewer parameters is introduced in Ref. [49] for faster computation. Another popular network for extracting appearance features is ResNet-50, as in Refs. [35,41,43], which resulted in a competing accuracy with fast computation. Peng et al. [43] extracted the appearance features from different layers of a ResNet-50 network forming a Feature Pyramid Network, which has the advantage of detecting objects at different scales. The LSTM network is an important concept for architecture design for processing a sequence of movies. It has been used for MOT application in multiple approaches such as Refs. [31,32]. The main approach taken by most current methods is to store the appearance features of the previous frames and retrieve them for comparison with the ones of the current frame. The important factor that affects the reliability of this comparison is the updating of the stored features. The object's appearance varies through the frames but not significantly between two adjacent frames; hence, constant updating can lead to a higher matching accuracy.

The second most common feature used for tracking is that related to the object's motion. This is specifically useful at full occlusion occurrence. In this case, the object's state can change significantly, and the appearance features will not be reliable for matching. A robust motion model can predict the object's location even if it disappears from the scene. The most common approach for motion tracking is the Kalman Filter, as in Refs. [50–52]. The authors in Refs. [34,45] use the relative position between two tracks in two adjacent frames and decide whether or not the two tracks are of the same object. The authors in Refs. [31,32,46], use deep learning approaches for motion tracking. The most common architecture for this approach is the LSTM network. The Kalman filter has a significantly lower computational cost than the deep learning approaches. The issue with utilizing only motion models for tracking is the random motion of objects. For instance, motion models would work better on cars where the motion is limited than on people. Zhou et al. introduced CenterTrack in Ref. [46] for tracking objects as points. The system is end-to-end, taking the current and the previous frames and outputting the matched tracks as illustrated in Figure 3.

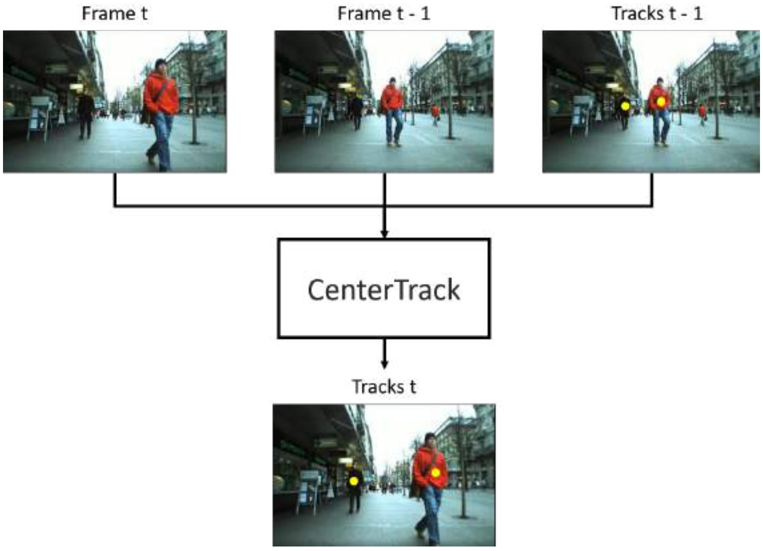

Figure 3. The point tracking approach in Ref. [46]. The current and previous frames are passed into the centerTrack network, which utilize the motion feature to detect and match tracks.

There are other features used for tracking. The authors in Refs. [56,61], added the IoU metric between the adjacent frames' detections to match the two tracks. The tracking of one object in the scene can be affected by other objects. For this reason, some methods have introduced the interactivity feature to the tracking algorithm. In Ref. [31], the interactivity features were extracted from an occupancy map using an LSTM network. The authors in Ref. [54] used a tracking graph and designed a set of conditions to measure the interactivity between two objects. Figure 4 illustrates the overview of tracking graph methods. The approach in Ref. [63], exploited the size and structure as features along with the appearance and motion for tracking. Increasing the number of features can improve the tracking process at the cost of computation time. The main issue would be the processing needed to fuse those features with different dimensionalities. The authors in Refs. [44,68] used IOU in addition to appearance and motion features to increase the reliability of the tracking. The authors in Ref. [44] further improved the model by adding epipolar constraints with the IOU and introduced a tracklenet to group similar tracks into a cluster. Ref. [69] added the deep_sort algorithm to the extracted features to reduce the unreliable tracks, and Ref. [36] added a correlation filter tracker to the CNN. Ref. [47] performed a similar approach of performing feature extraction and matching simultaneously by having affinity estimation and multi-dimensional assignment in one network. The authors in Refs. [70,71] experimented with 3D distance estimation from RGB frames. In Ref. [70], Poisson multi-Bernoulli mixture tracking filter was used to perform the 3D projections. In addition to CNN, Ref. [72] experimented visually with the track's Gaussian Mixture Probability Hypothesis Density. The authors in Ref. [37] introduced a motion segmentation framework using motion cues in addition to the IOU and bounding box clusters for object tracking through multiple frames. An interesting technique is established in Ref. [33], where one network has the current and prior frames as inputs and outputs point tracks. Those tracks are given to a displacement model to measure the similarity. Similarly, another approach to motion modeling is introduced in Ref. [45] to handle overlapping tracks by using the efficient quadratic pseudo-Boolean for optimization.

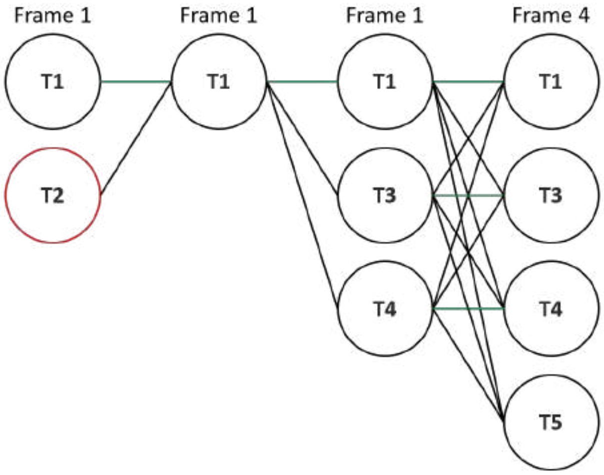

Figure 4. Track tree proposed in Track tree method system overview. The strength of each branch depends on a score evaluated by the matching algorithm. The green lines indicate matched tracks. The red circle indicate a lost track.

The features extracted from every detection in the current frame must be associated with those extracted from the previous frames. The most popular approach taken for data association in recent years would be the Hungarian algorithm, as in Refs. [50,53,60,63,67]. The advantage of this method is the accuracy accompanied by a fast computation time. Zhang et al. proposed ByteTrack in Ref. [50], where the Kalman filter is used for predicting the detection location followed by two levels of association. The first utilizes the appearance features in addition to the Intersection over Union (IoU) for matching tracks using the

Hungarian algorithm. The second level of association deals with the weak detections by utilizing only the IoU with the unmatched tracks remaining from the first level. The authors in Refs. [31,43,52] use deep learning networks for data association. The authors in Ref. [39] introduced a model trained using reinforcement learning. The metric learning concept was used to train the matching model in Ref. [32]. The authors in Ref. [62] take advantage of the object detection network for feature extraction. This would save computational costs from applying an appearance feature network on each detection in the current frame. The approaches in Refs. [43,52] apply the same concept in addition to an end-to-end system that takes the current frame and previous frames as input and outputs the current frame with tracks. The hierarchical single-branch network is an example of an end-to-end system proposed in Ref. [52] as illustrated in Figure 5. The P3AFormer tracker introduced in Ref. [57] uses the simultaneous detection and tracking framework where a decoder and a detector extract pixel-level features from the current and previous frames. The features are passed into a multilayer perceptron (MLP) that outputs the size, center, and class information. The features are then matched using the Hungarian algorithm. The system overview of the P3AFormer is shown in Figure 6.

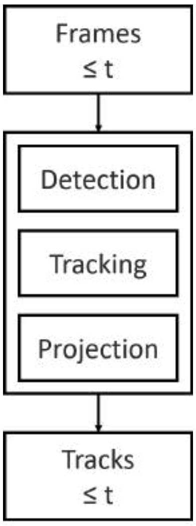

Figure 5. The hierarchical single-branch network proposed in Ref. [52]. The frames from a video source are passed into the network and outputs detections and tracks.

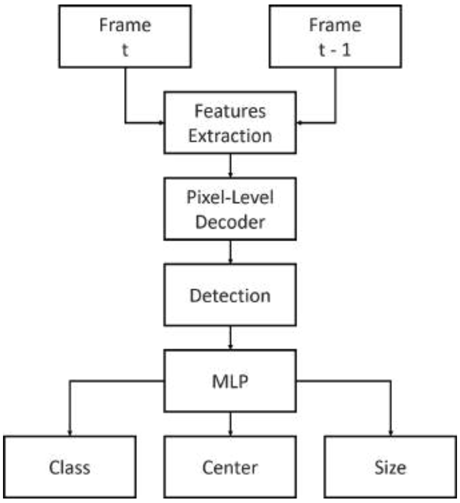

Figure 6. P3AFormer system overview [57]. The current and previous frames are passed for features extraction and detection. The extraction module consists of a backbone network, pixel-level decoder, and a detector, which is illustrated on the right. The features are passed to MLP heads to output the class, center, and size features.

The Recurrent Autoregressive Networks (RAN) approach introduced in Ref. [38] defines an autoregressive model that is used to estimate the mean and variance of appearance and motion features of all associated tracks of the same object stored in the external memory. The internal memory uses the model generated to compare with all upcoming detections. The one with the maximum score and above a certain threshold would then be considered the same object. The external and internal memory would then be updated with the new associated object. There are two types of independence in this approach. The first is that the motion and appearance models include different parameters, so they have different internal and external memories. The second is a new RAN model generated for every new object detected. The lost tracks are terminated after 20 frames. The visual features are extracted using the fully connected layer (fc8) of the inception network, and the motion feature is a 4-dimensional vector representing the width, height, and relative position from the previous detection. The CTracker framework [43] takes two adjacent frames as inputs and matches each detection with the other using Intersection over Union calculations. A constant velocity motion model is used to match the tracks up to a certain number of frames to handle the reappearance of lost tracks. This approach is end-to-end. It takes two frames as input and outputs two frames with detections and matching tracks.

The authors in Ref. [33] introduced an approach in which the detections' dissimilarity measures are computed and then matched. First is the dissimilarity cost computation. Next, the histogram of the H and S channels of the HSV colorspace of the previous detections is compared to the similar histogram of the current detections. A grid structure is used as in Refs. [73,74]. Hence, multiple histograms are used to match the appearance features. Furthermore, Linear Binary Pattern Histogram (LBPH), introduced in Ref. [75] and used for object recognition in Ref. [76], is utilized for computing the structure-based distance. The predicted and the measured position matching using the L2 norm is added as the motion-based distance. Finally, IoU calculates the size difference between the current detections and the previous tracks. The second step is using the Hungarian algorithm [77] to calculate the overall similarity using the four features calculated in the previous step.

The authors in Ref. [68] proposed V-IoU, an extension of IoU [78], for object tracking. The objective here is to reduce the number of ID switches and fragmentation by maintaining the location of lost tracks for a certain number of frames until it appears. A backtracking technique where the reappeared track is projected backward through the frames is implemented to validate that the reappeared track is, in fact, the lost one. In Ref. [46], CenterNet [79] is used to detect objects as points. Centertrack takes two adjacent frames as inputs in addition to the point detections in the previous frame. The tracks are associated using the offset between the current and previous point detections. The authors in Ref. [80] designed a motion segmentation model where the point clusters used for trajectory prediction were placed around the center of the detected bounding box. The approach in Ref. [37] employs optical flow and correlation co-clustering for projecting trajectory points across multiple frames, as illustrated in Figure 7.

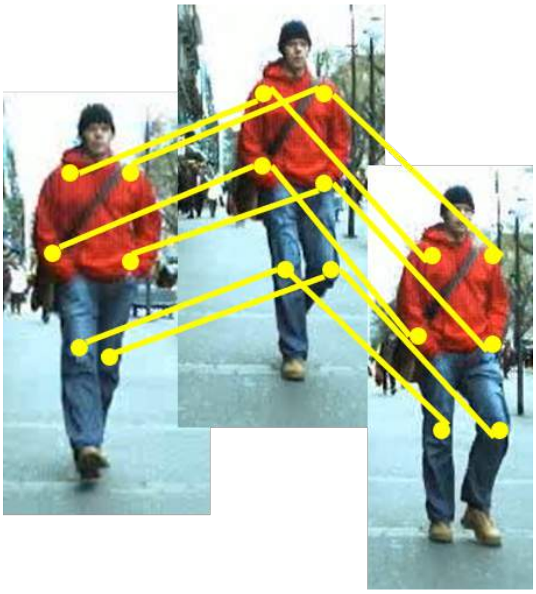

Figure 7. Motion segmentation. The trajectory points are projected across multiple frames.

There has been a recent advancement in multiple object tracking and segmentation (MOTS). This field tackles the issues related to the classic MOT, which are associated with the utilization of bounding box detection and tracking, such as background noises and loss of the shape features. The approach introduced in Ref. [81] used an instance segmentation mask on the extracted embedding. The method in Ref. [82] applies contrastive learning for learning the instance-masks used for segmentation. An offline approach introduced in Ref. [83] exploits appearance features for tracking. This method is currently at the top of the leader board at the MOTS20 challenge [84].

The system can achieve more reliability when utilizing multiple sensors to detect and track targets in addition to understanding their intentions [28]. The deep learning approaches are improving and showing promise in the LIDAR datasets. The issue with this is the bad running time, causing difficulty in real-time deployment [29]. The challenges facing 3D tracking are related to the fusion of the data perceived from LIDAR and RGB cameras. Table 3 lists the recent MOT techniques that utilize LIDAR and camera for tracking.

Table 3. Summary of the sensors fusion approaches used in 3D MOT techniques.

Tracker	RGB Camera	Point Cloud LIDAR	Data Fusion	Target Tracking
Simon et al. [85]	Enet	3D Voxel Quantization	Complex-YOLO	LMB RFS
Zhang et al. [86]	VGG-16	PointNet	Point-wise convolution + Start and end estimator	Linear Programming
Frossard et al. [87]	MV3D	MV3D	MV3D	Linear Programming
Hu et al. [71]	Faster R-CNN	34-layer DLA-up	Monocular 3D estimation	LSTM Network
Weng et al. [88]	ResNet	PointNet	Addition	Graph Neural Network
Sualeh et al. [89]	YOLOv3	IMM-UKF-JPDAF	3D projection	Munkres Association
Shenoi et al. [90]	Mask RCNN	F-PointNet	3-layered fully connected network	JPDA

Simon et al. [85] proposed complexer-YOLO, illustrated in Figure 8, for RGB and LIDAR data detection, tracking, and segmentation. A preprocessing step for the point cloud data input from LIDAR aims to generate a voxalized map of the 3D detections. The RGB frame is passed into ENet [91], which will output a semantic map. Both maps are matched and passed into the Complexer-YOLO network to output tracks. The approach in

Ref. [86] extracts features from the RGB frame and the point cloud data. An end-to-end approach was introduced in Ref. [87] for dealing with features extraction and fusing from RGB and Point Cloud Data, as illustrated in Figure 9. Point-wise convolution in addition to a start and end estimator are utilized for fusing both types of data to be used for tracking.

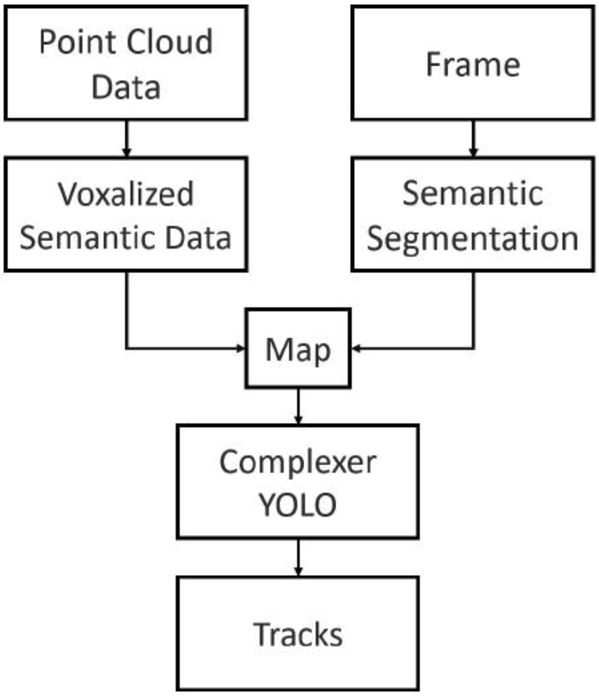

Figure 8. Complexer-YOLO [85]. Data from RGB frame and point cloud data are mapped and passed into Complexer-YOLO for tracking and matching.

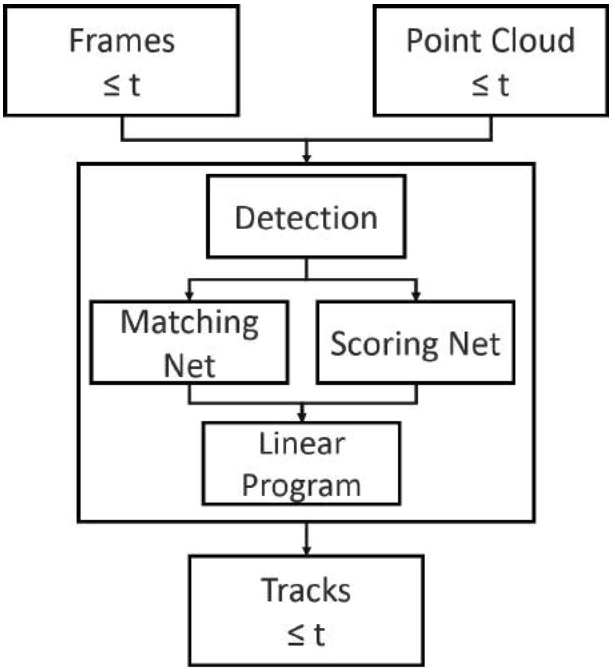

Figure 9. An end-to-end approach for 3D detection and tracking [87]. The RGB and point cloud data are passed into a detection network. Matching and scoring nets are then trained to generate trajectories across multiple frames.

3. Mot Benchmark Datasets and Evaluation Metrics

In this section, we review the most common datasets that are used for training and testing MOT techniques. We also provide an overview of the metrics used to evaluate the performance of these techniques.

3.1. Benchmark Datasets

Most research done in multiple object tracking uses standard datasets for evaluating the state of art techniques. In this way, we have a better view of what criteria, the new methodologies, have shown superiority. For this application, everyday moving objects could be pedestrians, vehicles, cyclists, etc. The most common datasets that provide a variation of those objects in the streets are the MOTChallenge collection and KITTI.

- MOTChallenge: The most common datasets in this collection are the MOT15 [19], MOT16 [92], MOT17 [92], and MOT20 [93]. There is a newly created set, MOT20, but it has not yet become a standard for evaluation in the research community to our current knowledge. The MOT datasets contain some data from existing sets such as PETS and TownCenter and others that are unique. Examples of the data included are presented in Table 4, where the amount of variation included in the MOT15 and MOT16 can be observed. Thus, the dataset is useful for training and testing using static and dynamic backgrounds and for 2D and 3D tracking. An evaluation tool is also given with the set to measure all features of the multiple object tracking algorithm, including accuracy, precision, and FPS. The ground truth data samples are shown in Figure 10.

Table 4. Examples of the types of data included in MOT15 and MOT16.

Dataset	Sequences	Length	Tracks	FPS	Platform	ViewPoint	Density	Weather
MOT 2015	TUD-Crossing	201	13	25	static	horizontal	5.5	cloudy
	PETS09-S2L2	436	42	7	static	high	22.1	cloudy
	ETH-Jelmoli	440	45	14	moving	low	5.8	sunny
	KITTI-16	209	17	10	static	horizontal	8.1	sunny
MOT 2016	MOT16-01	450	23	30	static	horizontal	14.2	cloudy
	MOT16-03	1500	148	30	static	high	69.7	night
	MOT16-06	1194	221	14	moving	low	9.7	sunny
	MOT16-12	900	86	30	moving	horizontal	9.2	indoor

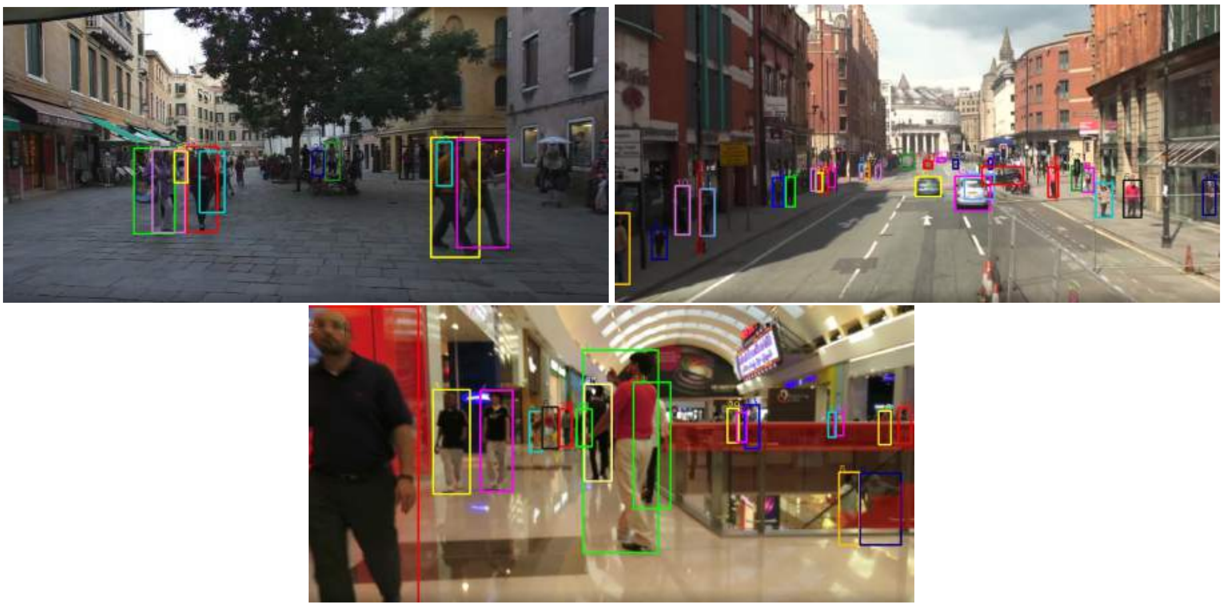

Figure 10. Samples from MOT 15-17 ground truth dataset. Samples of MOT15 (**top image**), MOT16 (**middle image**), and MOT17 (**bottom image**).

- KITTI [94]: This dataset is created specifically for autonomous driving. It was collected by a car driven through the streets with multiple sensors mounted for data collection. The set includes PointCloud data collected using LIDAR sensors and RGB video sequences captured by monocular cameras. It has been included in multiple research related to 2D and 3D multiple object tracking. Samples of the pointcloud and RGB data included in the KITTI dataset are shown in Figure 11.

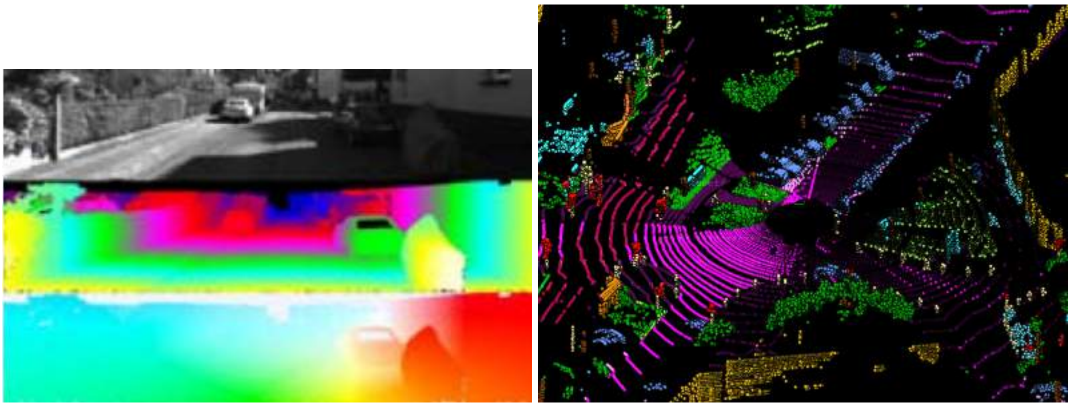

Figure 11. Samples of the KITTI dataset including Pointcloud and RGB. Visual odometry trajectory (**top left**), disparity and optical flow map (**top right**), visualized point cloud data [95] (**middle**), and 3D labels (**bottom**).

- UA-DETRAC [96–98]: The dataset includes videos sequences captured from static cameras looking at the streets at different cities. A huge amount of labeled vehicles can assist in training and testing for static background multiple object tracking in surveillance and autonomous driving. Samples of the UA-DETRAC dataset at different illumination conditions can be shown in Figure 12.

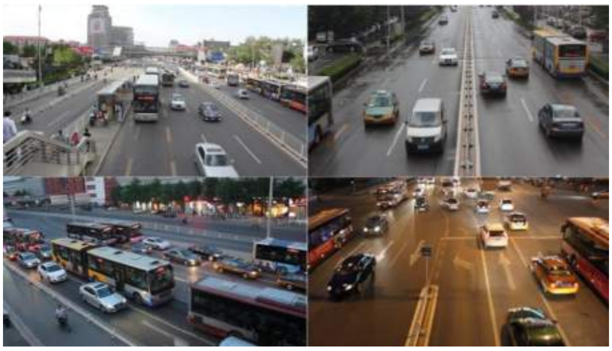

Figure 12. Samples of the UA-DETRAC dataset showing variation of illumination in the environment from a static camera.

3.2. Evaluation Metrics

The most common evaluation metrics are the CLEAR MOT metrics, which were developed by Refs. [22,23]. Mostly tracked objects (MT) and mostly lost objects (ML) in addition to IDF1 are uses to present the leaderboards in MotChallenges. False Positives (*FP*) is the number of falsely detected objects. false Negatives (*FN*) the number of falsely undetected objects. Fragmentation (Fragm) is the number of times a track gets interrupted. ID Switches (IDSW) is the number of times an ID changes. Multiple Object Tracking Accuracy (*MOTA*) is given by (1), whereas Multiple Object tracking Precision (*MOTP*) is given by (2). Finally, frames per second (Hz) and IDF1 given by (3).

$$MOTA = 1 - \frac{(FN + FP + IDSW)}{GT} \tag{1}$$

where GT is the total number of ground truth labels.

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{2}$$

where c is the number of matches in t frame and d is the correctly overlapping detections and tracks.

$$IdentificationF1(IDF1) = \frac{1}{\frac{1}{IDP} + \frac{1}{IDR}} \tag{3}$$

where identification precision is defined by:

$$IDP = \frac{IDTP}{IDTP + IDFP} \tag{4}$$

and identification recall is defined by:

$$IDR = \frac{IDTP}{IDTP + IDFN} \tag{5}$$

$IDTP$ is the sum of true positives edges weights, $IDFP$ is the sum of false positives edges weights, and $IDFN$ is the sum of false negatives edges weights

The metrics used for evaluating on the UA_DETRAC dataset utilize the precision recall (PR) for calculating the CLEAR metrics, as introduced in Ref. [96]. In addition to these metrics, the HOTA metric introduced in Ref. [99] is calculated by the formula in (6).

$$\int_{0<\alpha\leq 1} HOTA_\alpha \tag{6}$$

where

$$\sqrt{\frac{\sum_{c\epsilon TP_\alpha} (Ass\_IoU)_\alpha(c))}{|TP_\alpha| + |FN_\alpha| + |FP_\alpha|}} \tag{7}$$

where

$$Ass\_IoU = \frac{|TPA|}{|TPA| + |FNA| + |FPA|} \tag{8}$$

where TPA, FNA, and FPA are the association metrics.

There has been a recent advancement in the multiple object tracking and segmentation ($MOTS$). This field tackles the issues related to the classic MOT which are associated with the utilization of bounding box detection and tracking such as background noises and loss of the shape features. The MOTS20 Challenge [84] proposed metrics for evaluating methods that tackle this issue. The multi-object tracking and segmentation accuracy ($MOTSA$) is calculated using the formula in (9). Similarly, multi-object tracking and segmentation precision ($MOTSP$) and soft multi-object tracking and segmentation accuracy ($sMOTSA$) are found by the formulas in (10) and (11), respectively.

$$MOTSA = 1 - \frac{(|FN| + |FP| + |IDSW|)}{|M|} \tag{9}$$

where M is the ground truth masks.

$$MOTSP = \frac{\tilde{TP}}{|TP|} \tag{10}$$

where $\tilde{TP}$ is the soft version true positives (TP).

$$sMOTSA = \frac{\tilde{TP} - |FP| - |IDSW|}{|M|} \tag{11}$$

4. Evaluation and Discussion

In this section, we compare the MOT techniques based on the dataset used for evaluation. Then, analysis and discussion are conducted to provide insight for future work.

The performance of the most recent MOT techniques on MOT15, 16, 17, and 20 datasets are shown in Table 5. The ↑ stands for higher is better, and ↓ stands for lower is better. The protocol indicates the type of detector used for evaluating the results. The dataset provides the public detector and is typical for all methods. The private detector is designed by the method and is not shared. In MOT15, the tracker introduced in Ref. [38] has the highest accuracy and the lowest identity switches (IDs). It also maintained the highest percentage of tracks (MT) and has the lowest percentage of lost ones (ML). On the other hand, it had a significantly higher number of false positives and negatives compared to the method introduced in Ref. [42], which also performed with the highest FPS (Hz). The authors in Ref. [36] evaluated their system using the private detector protocol and have significantly lower fragmentation than all other methods. In Ref. [38], the tracker relied on appearance features extracted from the inception network layer and the position of the detections. The association process was done using conditional probability. The approach in Ref. [48] has the second best accuracy, where the motion features and the appearance features were used, as well as a category classifier for the association. The method with the lowest accuracy [80] only relied on the motion features, and the slowest performing method [39], where reinforcement learning was applied for data association. The approach in Ref. [62] has the highest accuracy on the MOT16 dataset using the private detector protocol. The method in Ref. [53] also used the private detector and has the highest HOTA. Both methods relied on appearance features for tracking and incorporated the Hungarian algorithm for matching. Similarly, a significantly faster-performing method [38] with slightly lower accuracy only used the appearance features and a prediction network for the association. The method in Ref. [35] used the Kalman filter for motion feature prediction and the appearance features extracted from the detection network for tracking. This method has a lower accuracy in comparison to other methods. In MOT17, the approach with the highest accuracy [57] has a significantly higher fragmentation than the one introduced in Ref. [64], which only used the motion feature for tracking. The method in Ref. [67] has slightly lower accuracy but with an acceptable FPS. This method used the appearance and motion features in addition to the Hungarian algorithm for matching. The approaches in Refs. [56,58] have an acceptable accuracy where both used Kalman filter for motion tracking and Ref. [56] neglected the appearance features. The method in Ref. [64] has the lowest number of fragmentation and only uses the motion features in tracking. The method in Ref. [57] has the highest accuracy on MOT20, although it has a significantly higher number of fragmentation compared to the one in Ref. [54]. The approach in Ref. [67] has the highest FPS, followed by the one in Ref. [54]. All of these methods relied on visual and motion features for tracking. The methods in Refs. [58,67] only used the motion features and had an acceptable accuracy. On the other hand, the ones that only relied on the visual features, such as Refs. [60,62,63,65] did not perform well according to the accuracy and other metrics. This evaluation shows that appearance features are essential for high accuracy, and other cues are used to boost performance. Based on the results from Table 5 and the summary presented in Table 2, the utilization of deep learning for data association reduces the processing time, as can be observed from Refs. [43,49]. On the other hand, including motion cues in the system drops the FPS significantly compared to only using the visual features as indicated by the results in Ref. [32]. Although adding complexity to the system drops the FPS, the IDS metric significantly improves when the motion features are included in the system. It can be concluded from these findings that to improve the FPS and the accuracy one should use deep learning in all MOT components. Although that might be the case, the end-to-end approaches introduced in Refs. [32,48] have used deep learning to extract appearance and motion features and data association, and the performance did not compete with other approaches. Deep learning approaches are data-driven, which means they are suitable for specific tasks but unsuitable or expected to perform poorly in real scenarios due to the variation from the data used in training [13].

Table 5. The performance using the MOT15, 16, 17, and 20 datasets.

Dataset	Tracker	MOTA ↑	MOTP ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Frag ↓	Hz ↑	HOTA ↑	Protocol
MOT15	Chen et al. [48]	53.0	**75.5**		29.1	20.2	5159	22,984	708	1476			Private
	Chen et al. [48]	38.5	72.6		8.7	37.4	4005	33,204	586	1263			Public
	Sadeghian et al. [31]	37.6	71.7		15.8	26.8	7933	29,397	1026	2024	1		Public
	Keuper et al. [37]	35.6		45.1	23.2	39.3	10,580	28,508	457	**969**			Private
	Fang et al. [38]	**56.5**	73.0	**61.3**	**45.1**	**14.6**	9386	16,921	**428**	1364	5.1		Public
	Xiang et al. [32]	37.1	72.5	28.4	12.6	39.7	8305	29,732	580	1193	1.0		Private
	Chu et al. [39]	38.9	70.6	44.5	16.6	31.5	7321	29,501	720	1440	0.3		Public
	Sun et al. [42]	38.3	71.1	45.6	17.6	41.2	**1290**	**2700**	1648	1515	**6.3**		Public
	Zhou et al. [46]	30.5	71.2	1.1	7.6	41.2	6534	35,284	879	2208	5.9		Private
	Dimitriou et al. [80]	24.2	70.2		7.1	49.2	6400	39,659	529	1034			Private
MOT16	Yoon et al. [49]	22.5	70.9	25.9	6.4	61.9	7346	39,092	1159	1538	**172.8**		Public
	Sadeghian et al. [31]	47.2		**75.8**	14.0	41.6	**2681**	92,856	774	1675	1.0		Public
	Keuper et al. [37]	47.1		52.3	20.4	46.9	6703	89,368	**370**	**598**			Private
	Fang et al. [38]	63	78.8		39.9	22.1	13,663	53,248	482	1251			Public
	Xiang et al. [32]	48.3	76.7		15.4	40.1	2706	91,047	543		0.5		Private
	Zhu et al. [40]	46.1	73.8	54.8	17.4	42.7	7909	89,874	532	1616	0.3		Private
	Zhou et al. [41]	64.8	78.6	**73.5**	**40.6**	22.0	13,470	49,927	794	1050	18.2		Private
	Chu et al. [39]	48.8	75.7	47.2	15.8	38.1	5875	86,567	906	1116	0.1		Public
	Peng et al. [43]	67.6	78.4	57.2	32.9	23.1	8934	48,305	1897		34.4		Private
	Henschel et al. [100]	47.8		47.8	19.1	38.2	8886	85,487	852				Private
	Wang et al. [44]	49.2		56.1	17.3	40.3	8400	83,702		882			Public
	Mahmoudi et al. [34]	65.2	78.4		32.4	21.3	6578	55,896	946	2283	11.2		Public
	Zhou et al. [35]	40.8	74.4		13.7	38.3	15,143	91,792	1051	2210	6.5		Private
	Lan et al. [45]	45.4	74.4		18.1	38.7	13,407	85,547	600	930			Public
	Xu et al. [50]	74.4		73.7	46.1	15.2						60.2	Private
	Ye et al. [51]	62.8	79.4	63.1	34.4	17.2	12,463	54,648	701	983			Private
	wang et al. [52]	50.4		47.5			18,370	69,800	1826				Private
	Yu et al. [53]	75.6	**80.9**	**75.8**	43.1	21.5	9786	**34,214**	448			**61.7**	Private
	Dai et al. [60]	63.8		70.6	30.3	30.6	7412	57,975	629		11.2	54.7	Private
	Hyun et al. [62]	**76.7**		73.1	**49.1**	**10.7**	10,689	30,428	1420				Private

Table 5. *Cont.*

Dataset	Tracker	MOTA ↑	MOTP ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Frag ↓	Hz ↑	HOTA ↑	Protocol
MOT17	Keuper et al. [37]	51.2			20.7	37.4	24,986	248,328	1851	2991			Private
	Zhu et al. [40]	48.2	75.9	55.7	19.3	38.3	26,218	263,608	2194	5378	11.4		Private
	Sun et al. [42]	52.4	53.9	76.9	**49.5**	21.4	25,423	234,592	8431	14,797	6.3		Public
	Peng et al. [43]	66.6	78.2	57.4	32.2	24.2	22,284	160,491	5529		**34.4**		Private
	Henschel et al. [100]	51.3		47.6	21.4	35.2	24,101	247,921	2648				Private
	Wang et al. [44]	51.9		58.0	23.5	35.5	37,311	231,658		2917			Public
	Henschel et al. [101]	52.6		50.8	19.7	35.8	31,572	232,572	3050	3792			Private
	Zhou et al. [46]	61.5		59.6	26.4	31.9	**14,076**	200,672	2583				Private
	Karunasekera et al. [33]	46.9	76.1		16.9	36.3			4478		17.1		Private
	Xu et al. [50]	73.1	73.0	45	16.8							59.7	Private
	Ye et al. [51]	61.7	79.3	62.0	32.5	20.1	32,863	181,035	1809	3168			Private
	Yu et al. [53]	73.8	**81.0**	74.7	41.7	23.2	27,999	118,623	1374			61.0	Private
	Wang et al. [54]	79.5		79.1			29,508	84,618	1302	2046	24.7	63.9	Private
	Gao et al. [55]	70.2		65.5	43.1	15.4	30,367	115,986	3265		10.7		Private
	Nasseri et al. [56]	80.4		77.7			28,887	**79,329**	2325	4689		63.3	Private
	Zhao et al. [57]	**81.2**		78.1	54.5	13.2	17,281	86,861	1893				Private
	Aharon et al. [58]	80.5		**80.2**			22,521	86,037	1212		4.5	**65.0**	Private
	Seidenschwarz et al. [59]	61.7		64.2	26.3	32.2			1639			50.9	Private
	Dai et al. [60]	63.0		69.9	30.4	30.6	23,022	183,659	1842		10.8	54.6	Private
	Zhang et al. [61]	62.1		65.0			24,052	188,264	1768				Public
	Zhang et al. [61]	77.3		75.9			45,030	79,716	3255				Private
	Hyun et al. [62]	76.5		73.6	47.6	**12.7**	29,808	99,510	3369				Private
	Chen et al. [63]	76.5		72.6								59.7	Private
	Cao et al. [64]	78.0		77.5			15,100	108,000	1950	**2040**		63.2	Private
	Wan et al. [65]	67.2		66.5	38.3	20.5	17,875	164,032	2896				Public
	Wan et al. [65]	75.6		76.4	48.8	16.2	26,983	108,186	2394				Private
	Du et al. [66]	79.6		79.5					**1194**		7.1	64.4	Private
	Zhang et al. [67]	80.3		77.3			25,491	83,721	2196		29.6	63.1	Private

Table 5. *Cont.*

Dataset	Tracker	MOTA ↑	MOTP ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Frag ↓	Hz ↑	HOTA ↑	Protocol
MOT20	Xu et al. [50]	59.7		67.8	66.5	7.7						54.0	Private
	Ye et al. [51]	54.9	79.1	59.1	36.2	23.5	19,953	211,392	1630	2455			Private
	Wang et al. [52]	51.5		44.5			38,223	208,616	4055				Private
	Yu et al. [53]	67.2	**79.2**	70.5	62.2	8.9	61,134	104,597	4243			56.5	Private
	Wang et al. [54]	76.5		76.8			**19,247**	101,290	971	**1190**	15.7	62.8	Private
	Nasseri et al. [56]	76.8		76.4			27,106	91,740	1446	3053		61.4	Private
	Zhao et al. [57]	**78.1**		76.4	**70.5**	**7.4**	25,413	86,510	1332				Private
	Aharon et al. [58]	77.8		**77.5**			24,638	88,863	1257		2.4	**63.3**	Private
	Seidenschwarz et al. [59]	52.7		55.0	29.1	26.8			1437			43.4	Private
	Dai et al. [60]	60.1		66.7	46.5	17.8	37,657	165,866	2926		4.0	51.7	Private
	Zhang et al. [61]	55.6		65.0			12,297	216,986	**480**				Public
	Hyun et al. [61]	66.3		67.7			41,538	130,072	2715				Private
	Hyun et al. [62]	72.8		70.5	64.3	12.8	25,165	112,897	2649				Private
	Chen et al. [63]	67.1		59.1								50.4	Private
	Cao et al. [64]	75.5		75.9			18,000	108,000	913	1198		63.2	Private
	Wan et al. [65]	70.4		71.9	65.1	9.6	48,343	101,034	3739				Private
	Du et al. [66]	73.8		77.0			25,491	**83,721**	2196		1.4	62.6	Private
	Zhang et al. [67]	77.8		75.2			26,249	87,594	1223		**17.5**	61.3	Private

Similarly, the performance of the KITTI dataset is shown in Table 6. The dataset is divided into multiple sequences: car, pedestrian, and cyclist. The methods are either evaluated on all of them combined or individually. As it can be observed, the pedestrian data are the most difficult to process with a good performance. Only methods evaluated on all sequences are used for comparison, and accordingly, the values corresponding to the best performance according to the metric are in bold. The rest are listed for observation and analysis. The approach introduced in Ref. [46] showed superiority. Although it did not perform well on the MOT15 dataset, it showed a competing performance on the MOT17 dataset. The authors in Ref. [102] evaluated the proposed technique on each sequence individually. They have superior performance on all of them. The performance on the UA_DETRAC dataset is shown in Table 7. Although the UA_DETRAC dataset is of a static background type and does not include the challenge of dynamic background, the approach introduced in Ref. [47] performed better on the KITTI dataset. This variation in the performance of the MOT techniques on multiple datasets may indicate that the MOT techniques are data driven and difficult to generalize. Similar to 2D Tracking, the deep learning approach is utilized for visual feature extraction in 3D. For processing point cloud data, PointNet is the most popular method. The authors in Refs. [85,89] did not rely on deep learning for techniques for the processing of point cloud data and performed poorly in terms of accuracy on the KITTI dataset shown in Table 6. The approach with the highest accuracy in Table 3 introduced in Ref. [86] creates multiple solutions for data fusion problems. The features extracted from LIDAR and camera sensors are fused by concatenation, addition, or weighted addition and then passed into a custom-designed network to calculate the correlation between the features and output the linked detections. The approaches that depend on deep learning for data fusion, Refs. [85,87,90], have a low MOTA, although [90] has high accuracy on the car set. The localization-based Tracking introduced in Ref. [103] has the best accuracy on the UA_DETRAC dataset. Although the DMM-Net method in Ref. [104] has significantly lower accuracy, it showed superiority in identity switches and fragmentation.

Table 6. The performance using the KITTI dataset. We have marked the highest scores in bold for methods evaluated on all categories.

Tracker	MOTA ↑	MOTP ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Frag ↓	Type
Chu et al. [47]	77.1	78.8	51.4	8.9	760	6998	123		Car
Simon et al. [85]	75.7	78.5	58.0	5.1					All
Zhang et al. [86]	84.8	85.2	73.2	2.77	711	4243	284	753	All
Scheidegger et al. [70]	80.4	81.3	62.8	6.2			121	613	Car
Frossard et al. [87]	76.15	83.42	60.0	8.31			296	868	All
Hu et al. [71]	84.5	85.6	73.4	2.8	**705**	**4242**			All
Weng et al. [88]	82.2	84.1	64.9	6			142	416	Car
Zhao et al. [36]	71.3	81.8	48.3	5.9					All
Weng et al. [102]	86.2	78.43					0	15	Car
Weng et al. [102]	70.9								Pedestrian
Weng et al. [102]	84.9								Cyclist
Luiten et al. [105]	84.8				681	4260	275		Car
Wang et al. [106]	68.2	76.6	60.6	12.3			111	**725**	All
Sualeh et al. [89]	78.10	79.3	70.3	9.1			21	111	Car
Sualeh et al. [89]	46.1	67.6	30.3	38.7			57	500	Pedestrian
Sualeh et al. [89]	70.9	77.6	71.4	14.3			3	24	Cyclist
Shenoi et al. [90]	85.7	85.5					98		Car
Shenoi et al. [90]	46.0	72.6					395		Pedestrian

Table 6. *Cont.*

Tracker	MOTA ↑	MOTP ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Frag ↓	Type
Zhou et al. [46]	**89.4**	85.1	**82.3**	**2.3**			**116**	744	All
Karunasekera et al. [33]	85.0	**85.5**	74.3	2.8			**301**		All
Zhao et al. [57]	91.2		86.5	2.3					Car
Zhao et al. [57]	67.7		49.1	14.5					Pedestrian
Aharon et al. [58]	90.3						250	280	Car
Aharon et al. [58]	65.1						204	609	Pedestrian
Wang et al. [107]	90.4	85.0	84.6	7.38	2322	962			Car
Wang et al. [107]	52.2	64.5	35.4	25.4	1112	2560			Pedestrian
Sun et al. [108]	86.9	85.7	83.1	2.9			271	254	Car

Table 7. The performance using the UA_DETRAC dataset.

Dataset	Tracker	PR-MOTA ↑	PR-MOTP ↑	PR-MT ↑	PR-ML ↓	PR-FP ↓	PR-FN ↓	PR-IDS ↓	PR-FRAG ↓
UA_DETRAC	Bochinski et al. [68]	30.7	37	32	22.6	18,046	179,191	363	1123
	Kutschbach et al. [72]	14.5	36	14	18.1	38,597	174,043	799	1607
	Hou et al. [69]	30.3	36.3	30.2	21.0	20,263	179,317	389	1260
	Sun et al. [42]	20.2	26.3	14.5	18.1	9747	**135,978**		
	Chu et al. [47]	19.8	36.7	17.1	18.2	14,989	164,433	617	
	Gloudemans et al. [103]	**46.4**	**69.5**	**41.1**	16.3				
	Sun et al. [104]	12.2		10.8	14.9	36,355	192,289	**228**	**674**
	Messoussi et al. [109]	31.2	50.9	28.1	18.5	**6036**	170,700	252	
	Wang et al. [110]	22.5	35.2	15.5	**10.1**			1563	3186

Navigation and self-driving applications in robotics depend on the online feature. The system must be able to react in real time. Although most of the techniques discussed can process video sequences online, their FPS is not showing robust performance, according to the Hz metric, to be deployed in an application such as self-driving. The method introduced in Ref. [43] has the highest FPS overall and acceptable accuracy on the MOT16 and MOT17 datasets. However, the research utilizing LIDAR and RGB cameras show potential in robotics navigation and autonomous driving applications.

5. Current Research Challenges

Through this study, we gained insight into the current trends of online MOT methods that can be utilized in robotics applications and the challenges faced. The first challenge would be the online feature. The MOT algorithm should be able to operate in real-time for most robotics applications in order to be able to react to environmental change. The second challenge would be the accurate track trajectory across multiple frames. The lack of this problem can cause multiple identity switches and difficulty keeping a concise description of the surroundings. In addition, the issue is segmenting the detections at pixel level and tracking them. The bounding box provides wrong information about the object in shape and size, along with noises from the background. Finally, the motion feature has proven its value in tracking, but it is not simple to track random moving objects such as people, animals, cyclists, etc.

6. Future Work

The final objective of the research done on deploying MOT algorithms in autonomous robots is to have a reliable system that contributes to reducing accidents and facilitating tasks that might be difficult for humans to carry out. One aspect that we found the current research lacks is the generation of a new benchmark dataset that includes data collected by the standard sensors employed by the current industry. Sensors such as ultrasonic and LiDar are essential in today's autonomous robot manufacturing, and it is necessary to use the same tools to make the research on MOT up-to-date. Moreover, using deep learning algorithms for detection and tracking would face a massive problem due to the risk of meeting a variation that was not included in the training set. Thus, deep learning should be trained on segmenting the road regions and hence, be trained on any area before deployment. This is one of the approaches that can be researched to tackle the problem of dealing with new objects. The current research looks at appearance and motion models as necessary in MOT. They are further going into learning the behavior of objects in the scenes and the interactivity between those objects. For instance, two objects moving towards each other would lead to one of them getting covered, and a track would be lost. As the MOT system's complexity increases, it becomes more challenging to work in real-time. The research on embedded processors that can be utilized in autonomous robots will significantly contribute to increasing the accuracy while maintaining the online feature.

7. Conclusions

This paper aims to review the current trends and challenges related to the MOT for autonomous robotics applications. This area of study has been frequently researched recently due to its high potential and standards, which are difficult to achieve. The paper has discussed and compared the MOT techniques through a common framework and datasets, including MOTChallenges, KITTI, and UA_DETRAC. There is a vast area left to explore and investigate as well as multiple approaches created by the literature that has the potential to build into reliable and robust techniques. A summary of the components utilized in the general MOT framework, including appearance and motion cues, data association, and occlusion handling, has been listed and studied. In addition, the popular methods used for data fusion between multiple sensors, focusing on the camera and LIDAR, have been reviewed. The role that deep learning techniques are utilized in MOT approaches has been investigated thoroughly using quantitative analysis to evaluate its limitations and strong points.

Author Contributions: Conceptualization, M.G., T.B. and A.G.; methodology, M.G., T.B. and A.G.; software, M.G., T.B. and A.G.; validation, M.Y., H.A., and M.A.; formal analysis, M.G., T.B. and A.G.; investigation, M.G., T.B. and A.G.; resources, M.G. and M.A.; data curation, M.G., T.B. and A.G.; writing—original draft preparation, T.B., A.G. and H.A.; writing—review and editing, M.G., T.B., A.G., M.Y., H.A. and M.A.; visualization, M.G., T.B., A.G. and M.A.; supervision, M.G. and M.A.; project administration, M.G. and M.A.; funding acquisition, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is part of a research project funded by the Office of Research and Sponsored Programs (ORSP) at Abu Dhabi University through a research fund (Grant number 19300564).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MOT	Multiple Object Tracking
LIDAR	Light Detection and Ranging
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
UA-DETRAC	University at Albany DEtection and TRACking
SLAM	Simultaneous localization and mapping
ROI	Region of Interest
IoU	Intersection over Union
HOTA	Higher Order Tracking Accuracy
SLAMMOT	Simultaneous localization and mapping Multiple Object Tracking
RTU	Recurrent Tracking Unit
LSTM	Long Short Term Memory
ECO	Efficient Convolution Operators
PCA	Principal Component Analysis
LDAE	Lightweight and Deep Appearance Embedding
DLA	Deep Layer Aggregation
GCD	Global Context Disentangling
GTE	Guided Transformer Encoder
DETR	DEtection TRansformer
PCB	Part-based Convolutional Baseline

References

1. Runz, M.; Buffier, M.; Agapito, L. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2018, Munich, Germany, 16–20 October 2019; pp. 10–20.
2. Zhou, Q.; Zhao, S.; Li, H.; Lu, R.; Wu, C. Adaptive Neural Network Tracking Control for Robotic Manipulators with Dead Zone. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3611–3620. [CrossRef] [PubMed]
3. Yu, C.; Liu, Z.; Liu, X.; Xie, F.; Yang, Y.; Wei, Q.; Fei, Q. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018; pp. 1168–1174.
4. Fehr, L.; Langbein, W.E.; Skaar, S.B. Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey. *J. Rehabil. Res. Dev.* **2000**, *37*, 353–360.
5. Simpson, R. Smart wheelchairs: A literature review. *J. Rehabil. Res. Dev.* **2005**, *42*, 423–436. [CrossRef]
6. Martins, M.M.; Santos, C.P.; Frizera-Neto, A.; Ceres, R. Assistive mobility devices focusing on Smart Walkers: Classification and review. *Robot. Auton. Syst.* **2012**, *60*, 548–562. [CrossRef]
7. Khan, M.Q.; Lee, S. A comprehensive survey of driving monitoring and assistance systems. *Sensors* **2019**, *19*, 2574. [CrossRef] [PubMed]
8. Van Brummelen, J.; O'Brien, M.; Gruyer, D.; Najjaran, H. Autonomous vehicle perception: The technology of today and tomorrow. *Transp. Res. Part C Emerg. Technol.* **2018**, *89*, 384–406. [CrossRef]
9. Pham, H.X.; La, H.M.; Feil-Seifer, D.; Deans, M.C. A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 1537–1548. [CrossRef]
10. Cai, Z.; Wang, L.; Zhao, J.; Wu, K.; Wang, Y. Virtual target guidance-based distributed model predictive control for formation control of multiple UAVs. *Chin. J. Aeronaut.* **2020**, *33*, 1037–1056. [CrossRef]
11. Huang, Y.; Liu, W.; Li, B.; Yang, Y.; Xiao, B. Finite-time formation tracking control with collision avoidance for quadrotor UAVs. *J. Frankl. Inst.* **2020**, *357*, 4034–4058. [CrossRef]
12. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T. Multiple object tracking: A literature review. *Artif. Intell.* **2021**, *293*, 103448. [CrossRef]
13. Ciaparrone, G.; Luque Sánchez, F.; Tabik, S.; Troiano, L.; Tagliaferri, R.; Herrera, F. Deep learning in video multi-object tracking: A survey. *Neurocomputing* **2020**, *381*, 61–88. [CrossRef]
14. Xu, Y.; Zhou, X.; Chen, S.; Li, F. Deep learning for multiple object tracking: A survey. *IET Comput. Vis.* **2019**, *13*, 411–419. [CrossRef]
15. Irvine, J.M.; Wood, R.J.; Reed, D.; Lepanto, J. Video image quality analysis for enhancing tracker performance. In Proceedings of the 2013 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA, 23–25 October 2013; pp. 1–9. [CrossRef]
16. Meng, L.; Yang, X. A Survey of Object Tracking Algorithms. *Zidonghua Xuebao/Acta Autom. Sin.* **2019**, *45*, 1244–1260.
17. Jain, V.; Wu, Q.; Grover, S.; Sidana, K.; Chaudhary, D.G.; Myint, S.; Hua, Q. Generating Bird's Eye View from Egocentric RGB Videos. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1–11. [CrossRef]

18. Yeong, D.J.; Velasco-Hernandez, G.; Barry, J.; Walsh, J. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. *Sensors* **2021**, *21*, 2140. [CrossRef]
19. Leal-Taixé, L.; Milan, A.; Reid, I.; Roth, S.; Schindler, K. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv* **2015**, arXiv:1504.01942.
20. Xu, Z.; Rong, Z.; Wu, Y. A survey: Which features are required for dynamic visual simultaneous localization and mapping? *Vis. Comput. Ind. Biomed. Art* **2021**, *4*, 20. [CrossRef]
21. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **2020**, *37*, 362–386. [CrossRef]
22. Janai, J.; Güney, F.; Behl, A.; Geiger, A. Computer vision for autonomous vehicles. *Found. Trends Comput. Graph. Vis.* **2020**, *12*, 1–308. [CrossRef]
23. Datondji, S.R.E.; Dupuis, Y.; Subirats, P.; Vasseur, P. A Survey of Vision-Based Traffic Monitoring of Road Intersections. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2681–2698. [CrossRef]
24. Buch, N.; Velastin, S.A.; Orwell, J. A review of computer vision techniques for the analysis of urban traffic. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 920–939. [CrossRef]
25. Otto, A.; Agatz, N.; Campbell, J.; Golden, B.; Pesch, E. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks* **2018**, *72*, 411–458. [CrossRef]
26. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. [CrossRef]
27. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Cardoso, V.B.; Forechi, A.; Jesus, L.; Berriel, R.; Paixão, T.M.; Mutz, F.; et al. Self-driving cars: A survey. *Expert Syst. Appl.* **2021**, *165*, 113816. [CrossRef]
28. Wang, Z.; Wu, Y.; Niu, Q. Multi-Sensor Fusion in Automated Driving: A Survey. *IEEE Access* **2020**, *8*, 2847–2868. [CrossRef]
29. Elhousni, M.; Huang, X. A Survey on 3D LiDAR Localization for Autonomous Vehicles. In Proceedings of the IEEE Intelligent Vehicles Symposium, Las Vegas, NV, USA, 9 October–13 November 2020; pp. 1879–1884.
30. Fritsch, J.; Kühnl, T.; Geiger, A. A new performance measure and evaluation benchmark for road detection algorithms. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013; pp. 1693–1700. [CrossRef]
31. Sadeghian, A.; Alahi, A.; Savarese, S. Tracking the Untrackable: Learning to Track Multiple Cues with Long-Term Dependencies. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 300–311. [CrossRef]
32. Xiang, J.; Zhang, G.; Hou, J. Online Multi-Object Tracking Based on Feature Representation and Bayesian Filtering Within a Deep Learning Architecture. *IEEE Access* **2019**, *7*, 27923–27935. [CrossRef]
33. Karunasekera, H.; Wang, H.; Zhang, H. Multiple Object Tracking With Attention to Appearance, Structure, Motion and Size. *IEEE Access* **2019**, *7*, 104423–104434. [CrossRef]
34. Mahmoudi, N.; Ahadi, S.M.; Mohammad, R. Multi-target tracking using CNN-based features: CNNMTT. *Multimed. Tools Appl.* **2019**, *78*, 7077–7096. [CrossRef]
35. Zhou, Q.; Zhong, B.; Zhang, Y.; Li, J.; Fu, Y. Deep Alignment Network Based Multi-Person Tracking With Occlusion and Motion Reasoning. *IEEE Trans. Multimed.* **2019**, *21*, 1183–1194. [CrossRef]
36. Zhao, D.; Fu, H.; Xiao, L.; Wu, T.; Dai, B. Multi-Object Tracking with Correlation Filter for Autonomous Vehicle. *Sensors* **2018**, *18*, 2004. [CrossRef]
37. Keuper, M.; Tang, S.; Andres, B.; Brox, T.; Schiele, B. Motion Segmentation & Multiple Object Tracking by Correlation Co-Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 140–153. [CrossRef]
38. Fang, K.; Xiang, Y.; Li, X.; Savarese, S. Recurrent Autoregressive Networks for Online Multi-Object Tracking. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018. [CrossRef]
39. Chu, P.; Fan, H.; Tan, C.C.; Ling, H. Online Multi-Object Tracking with Instance-Aware Tracker and Dynamic Model Refreshment. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 7–11 January 2019. [CrossRef]
40. Zhu, J.; Yang, H.; Liu, N.; Kim, M.; Zhang, W.; Yang, M.H. Online Multi-Object Tracking with Dual Matching Attention Networks. *arXiv* **2019**, arXiv:1902.00749.
41. Zhou, Z.; Xing, J.; Zhang, M.; Hu, W. Online Multi-Target Tracking with Tensor-Based High-Order Graph Matching. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 1809–1814. [CrossRef]
42. Sun, S.; Akhtar, N.; Song, H.; Mian, A.; Shah, M. Deep Affinity Network for Multiple Object Tracking. *arXiv* **2018**, arXiv:1810.11780.
43. Peng, J.; Wang, C.; Wan, F.; Wu, Y.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; Fu, Y. Chained-Tracker: Chaining Paired Attentive Regression Results for End-to-End Joint Multiple-Object Detection and Tracking. *arXiv* **2020**, arXiv:2007.14557.
44. Wang, G.; Wang, Y.; Zhang, H.; Gu, R.; Hwang, J.N. Exploit the Connectivity: Multi-Object Tracking with TrackletNet. *arXiv* **2018**, arXiv:1811.07258.

45. Lan, L.; Wang, X.; Zhang, S.; Tao, D.; Gao, W.; Huang, T.S. Interacting Tracklets for Multi-Object Tracking. *IEEE Trans. Image Process.* **2018**, *27*, 4585–4597. [CrossRef] [PubMed]
46. Zhou, X.; Koltun, V.; Krähenbühl, P. Tracking Objects as Points. *arXiv* **2020**, arXiv:2004.01177.
47. Chu, P.; Ling, H. FAMNet: Joint Learning of Feature, Affinity and Multi-dimensional Assignment for Online Multiple Object Tracking. *arXiv* **2019**, arXiv:1904.04989.
48. Chen, L.; Ai, H.; Shang, C.; Zhuang, Z.; Bai, B. Online multi-object tracking with convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 645–649. [CrossRef]
49. Yoon, K.; Kim, D.Y.; Yoon, Y.C.; Jeon, M. Data Association for Multi-Object Tracking via Deep Neural Networks. *Sensors* **2019**, *19*, 559. [CrossRef]
50. Xu, B.; Liang, D.; Li, L.; Quan, R.; Zhang, M. An Effectively Finite-Tailed Updating for Multiple Object Tracking in Crowd Scenes. *Appl. Sci.* **2022**, *12*, 1061. [CrossRef]
51. Ye, L.; Li, W.; Zheng, L.; Zeng, Y. Lightweight and Deep Appearance Embedding for Multiple Object Tracking. *IET Comput. Vis.* **2022**, *16*, 489–503. [CrossRef]
52. Wang, F.; Luo, L.; Zhu, E.; Wang, S.; Long, J. Multi-object Tracking with a Hierarchical Single-branch Network. *CoRR* **2021**, abs/2101.01984. Available online: http://xxx.lanl.gov/abs/2101.01984 (accessed on 7 September 2022).
53. Yu, E.; Li, Z.; Han, S.; Wang, H. RelationTrack: Relation-aware Multiple Object Tracking with Decoupled Representation. *CoRR* **2021**, abs/2105.04322. Available online: http://xxx.lanl.gov/abs/2105.04322 (accessed on 7 September 2022).
54. Wang, S.; Sheng, H.; Yang, D.; Zhang, Y.; Wu, Y.; Wang, S. Extendable Multiple Nodes Recurrent Tracking Framework with RTU++. *IEEE Trans. Image Process.* **2022**, *31*, 5257–5271. [CrossRef] [PubMed]
55. Gao, Y.; Gu, X.; Gao, Q.; Hou, R.; Hou, Y. TdmTracker: Multi-Object Tracker Guided by Trajectory Distribution Map. *Electronics* **2022**, *11*, 1010. [CrossRef]
56. Nasseri, M.H.; Babaee, M.; Moradi, H.; Hosseini, R. Fast Online and Relational Tracking. *arXiv* **2022**, arXiv:2208.03659.
57. Zhao, Z.; Wu, Z.; Zhuang, Y.; Li, B.; Jia, J. Tracking Objects as Pixel-wise Distributions. *arXiv* **2022**, arXiv:2207.05518.
58. Aharon, N.; Orfaig, R.; Bobrovsky, B.Z. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *arXiv* **2022**, arXiv:2206.14651.
59. Seidenschwarz, J.; Brasó, G.; Elezi, I.; Leal-Taixé, L. Simple Cues Lead to a Strong Multi-Object Tracker. *arXiv* **2022**, arXiv:2206.04656.
60. Dai, P.; Feng, Y.; Weng, R.; Zhang, C. Joint Spatial-Temporal and Appearance Modeling with Transformer for Multiple Object Tracking. *arXiv* **2022**, arXiv:2205.15495.
61. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. Robust Multi-Object Tracking by Marginal Inference. *arXiv* **2022**, arXiv:2208.03727.
62. Hyun, J.; Kang, M.; Wee, D.; Yeung, D.Y. Detection Recovery in Online Multi-Object Tracking with Sparse Graph Tracker. *arXiv* **2022**, arXiv:2205.00968.
63. Chen, M.; Liao, Y.; Liu, S.; Wang, F.; Hwang, J.N. TR-MOT: Multi-Object Tracking by Reference. *arXiv* **2022**, arXiv:2203.16621.
64. Cao, J.; Weng, X.; Khirodkar, R.; Pang, J.; Kitani, K. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking. *arXiv* **2022**, arXiv:2203.1662.
65. Wan, J.; Zhang, H.; Zhang, J.; Ding, Y.; Yang, Y.; Li, Y.; Li, X. DSRRTracker: Dynamic Search Region Refinement for Attention-based Siamese Multi-Object Tracking. *arXiv* **2022**, arXiv:2203.10729.
66. Du, Y.; Song, Y.; Yang, B.; Zhao, Y. StrongSORT: Make DeepSORT Great Again. *arXiv* **2022**. arXiv:2202.13514.
67. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *arXiv* **2021**, arXiv:2110.06864.
68. Bochinski, E.; Senst, T.; Sikora, T. Extending IOU Based Multi-Object Tracking by Visual Information. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; pp. 1–6. [CrossRef]
69. Hou, X.; Wang, Y.; Chau, L.P. Vehicle Tracking Using Deep SORT with Low Confidence Track Filtering. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–6. [CrossRef]
70. Scheidegger, S.; Benjaminsson, J.; Rosenberg, E.; Krishnan, A.; Granstrom, K. Mono-Camera 3D Multi-Object Tracking Using Deep Learning Detections and PMBM Filtering. *arXiv* **2018**, arXiv:1802.09975.
71. Hu, H.N.; Cai, Q.Z.; Wang, D.; Lin, J.; Sun, M.; Kraehenbuehl, P.; Darrell, T.; Yu, F. Joint Monocular 3D Vehicle Detection and Tracking. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 5389–5398. [CrossRef]
72. Kutschbach, T.; Bochinski, E.; Eiselein, V.; Sikora, T. Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–5. [CrossRef]
73. Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91. [CrossRef]
74. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. [CrossRef]

75. Ojala, T.; Pietikäinen, M.; Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.* **1996**, *29*, 51–59. [CrossRef]
76. Ahonen, T.; Hadid, A.; Pietikainen, M. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 2037–2041. [CrossRef]
77. Kuhn, H.W. The hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
78. Bochinski, E.; Eiselein, V.; Sikora, T. High-Speed tracking-by-detection without using image information. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–6. [CrossRef]
79. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
80. Dimitriou, N.; Stavropoulos, G.; Moustakas, K.; Tzovaras, D. Multiple object tracking based on motion segmentation of point trajectories. In Proceedings of the 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Colorado Springs, CO, USA, 23–26 August 2016; pp. 200–206. [CrossRef]
81. Cai, J.; Wang, Y.; Zhang, H.; Hsu, H.; Ma, C.; Hwang, J. IA-MOT: Instance-Aware Multi-Object Tracking with Motion Consistency. *CoRR* **2020**, abs/2006.13458. Available online: http://xxx.lanl.gov/abs/2006.13458 (accessed on 7 September 2022).
82. Yan, B.; Jiang, Y.; Sun, P.; Wang, D.; Yuan, Z.; Luo, P.; Lu, H. Towards Grand Unification of Object Tracking. *arXiv* **2022**, arXiv:2207.07078.
83. Yang, F.; Chang, X.; Dang, C.; Zheng, Z.; Sakti, S.; Nakamura, S.; Wu, Y. ReMOTS: Self-Supervised Refining Multi-Object Tracking and Segmentation. *CoRR* **2020**, abs/2007.03200. Available online: http://xxx.lanl.gov/abs/2007.03200 (accessed on 7 September 2022).
84. Voigtlaender, P.; Krause, M.; Osep, A.; Luiten, J.; Sekar, B.B.G.; Geiger, A.; Leibe, B. MOTS: Multi-Object Tracking and Segmentation. *arXiv* **2019**, arXiv:1902.03604.
85. Simon, M.; Amende, K.; Kraus, A.; Honer, J.; Sämann, T.; Kaulbersch, H.; Milz, S.; Gross, H.M. Complexer-YOLO: Real-Time 3D Object Detection and Tracking on Semantic Point Clouds. *arXiv* **2019**, arXiv:1904.07537.
86. Zhang, W.; Zhou, H.; Sun, S.; Wang, Z.; Shi, J.; Loy, C.C. Robust Multi-Modality Multi-Object Tracking. *arXiv* **2019**, arXiv:1909.03850.
87. Frossard, D.; Urtasun, R. End-to-end Learning of Multi-sensor 3D Tracking by Detection. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 635–642. [CrossRef]
88. Weng, X.; Wang, Y.; Man, Y.; Kitani, K. GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with Multi-Feature Learning. *arXiv* **2020**, arXiv:2006.07327.
89. Sualeh, M.; Kim, G.W. Visual-LiDAR Based 3D Object Detection and Tracking for Embedded Systems. *IEEE Access* **2020**, *8*, 156285–156298. [CrossRef]
90. Shenoi, A.; Patel, M.; Gwak, J.; Goebel, P.; Sadeghian, A.; Rezatofighi, H.; Martín-Martín, R.; Savarese, S. JRMOT: A Real-Time 3D Multi-Object Tracker and a New Large-Scale Dataset. *arXiv* **2020**, arXiv:2002.08397.
91. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *CoRR* **2016**, abs/1606.02147. Available online: http://xxx.lanl.gov/abs/1606.02147 (accessed on 7 September 2022).
92. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A Benchmark for Multi-Object Tracking. *arXiv* **2016**, arXiv:1603.00831.
93. Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.D.; Roth, S.; Schindler, K.; Leal-Taixé, L. MOT20: A benchmark for multi object tracking in crowded scenes. *CoRR* **2020**, abs/2003.09003. Available online: http://xxx.lanl.gov/abs/2003.09003 (accessed on 7 September 2022).
94. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
95. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
96. Wen, L.; Du, D.; Cai, Z.; Lei, Z.; Chang, M.; Qi, H.; Lim, J.; Yang, M.; Lyu, S. UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking. *Comput. Vis. Image Underst.* **2020**, *193*, 102907. [CrossRef]
97. Lyu, S.; Chang, M.C.; Du, D.; Li, W.; Wei, Y.; Del Coco, M.; Carcagnì, P.; Schumann, A.; Munjal, B.; Choi, D.H.; et al. UA-DETRAC 2018: Report of AVSS2018 & IWT4S challenge on advanced traffic monitoring. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; pp. 1–6.
98. Lyu, S.; Chang, M.C.; Du, D.; Wen, L.; Qi, H.; Li, Y.; Wei, Y.; Ke, L.; Hu, T.; Del Coco, M.; et al. UA-DETRAC 2017: Report of AVSS2017 & IWT4S Challenge on Advanced Traffic Monitoring. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–7.
99. Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P.H.S.; Geiger, A.; Leal-Taixé, L.; Leibe, B. HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking. *CoRR* **2020**, abs/2009.07736. Available online: http://xxx.lanl.gov/abs/2009.07736 (accessed on 7 September 2022).
100. Henschel, R.; Leal-Taixé, L.; Cremers, D.; Rosenhahn, B. Fusion of Head and Full-Body Detectors for Multi-Object Tracking. *arXiv* **2017**, arXiv:1705.08314.

101. Henschel, R.; Zou, Y.; Rosenhahn, B. Multiple People Tracking Using Body and Joint Detections. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 770–779. [CrossRef]
102. Weng, X.; Kitani, K. A Baseline for 3D Multi-Object Tracking. *CoRR* **2019**, abs/1907.03961. Available online: http://xxx.lanl.gov/abs/1907.03961(accessed on 7 September 2022).
103. Gloudemans, D.; Work, D.B. Localization-Based Tracking. *CoRR* **2021**, abs/2104.05823. Available online: http://xxx.lanl.gov/abs/2104.05823(accessed on 7 September 2022).
104. Sun, S.; Akhtar, N.; Song, X.; Song, H.; Mian, A.; Shah, M. Simultaneous Detection and Tracking with Motion Modelling for Multiple Object Tracking. *CoRR* **2020**, abs/2008.08826. Available online: http://xxx.lanl.gov/abs/2008.08826 (accessed on 7 September 2022).
105. Luiten, J.; Fischer, T.; Leibe, B. Track to Reconstruct and Reconstruct to Track. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1803–1810. [CrossRef]
106. Wang, S.; Sun, Y.; Liu, C.; Liu, M. PointTrackNet: An End-to-End Network For 3-D Object Detection and Tracking From Point Clouds. *arXiv* **2020**, arXiv:2002.11559.
107. Wang, L.; Zhang, X.; Qin, W.; Li, X.; Yang, L.; Li, Z.; Zhu, L.; Wang, H.; Li, J.; Liu, H. CAMO-MOT: Combined Appearance-Motion Optimization for 3D Multi-Object Tracking with Camera-LiDAR Fusion. *arXiv* **2022**, arXiv:2209.02540.
108. Sun, Y.; Zhao, Y.; Wang, S. Multiple Traffic Target Tracking with Spatial-Temporal Affinity Network. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–13. [CrossRef]
109. Messoussi, O.; de Magalhaes, F.G.; Lamarre, F.; Perreault, F.; Sogoba, I.; Bilodeau, G.; Nicolescu, G. Vehicle Detection and Tracking From Surveillance Cameras in Urban Scenes. *CoRR* **2021**, abs/2109.12414. Available online: http://xxx.lanl.gov/abs/2109.12414 (accessed on 7 September 2022).
110. Wang, G.; Gu, R.; Liu, Z.; Hu, W.; Song, M.; Hwang, J. Track without Appearance: Learn Box and Tracklet Embedding with Local and Global Motion Patterns for Vehicle Tracking. *CoRR* **2021**, abs/2108.06029. Available online: http://xxx.lanl.gov/abs/2108.06029 (accessed on 7 September 2022).