

Article

# DBSCAN-Based Tracklet Association Annealer for Advanced Multi-Object Tracking

Jongwon Kim  and Jeongho Cho \* 

Department of Electrical Engineering, Soonchunhyang University, Asan 31538, Korea; jw.kim@sch.ac.kr  
\* Correspondence: jcho@sch.ac.kr; Tel.: +82-41-530-4960

**Abstract:** Recently, as the demand for technological advancement in the field of autonomous driving and smart video surveillance is gradually increasing, considerable progress in multi-object tracking using deep neural networks has been achieved, and its application field is also expanding. However, various problems have not been fully addressed owing to the inherent limitations in video cameras, such as the tracking of objects in an occluded environment. Therefore, in this study, we propose a density-based object tracking technique redesigned based on DBSCAN, which has high robustness against noise and is excellent for nonlinear clustering. Moreover, it improves the noise vulnerability inherent to multi-object tracking, reduces the difficulty of trajectory separation, and facilitates real-time processing through simple structural expansion. Through performance test evaluation, it was confirmed that by using the proposed technique, several performance indices were improved compared to the existing tracking technique. In particular, when added as a post processor to the existing tracker, the tracking performance owing to noise suppression was considerably improved by more than 10%. Thus, the proposed method can be applied in industrial environments, such as real pedestrian analysis and surveillance security systems.

**Keywords:** object tracking; DBSCAN; video surveillance; trajectory separation; clustering



**Citation:** Kim, J.; Cho, J.

DBSCAN-Based Tracklet Association Annealer for Advanced Multi-Object Tracking. *Sensors* **2021**, *21*, 5715.  
<https://doi.org/10.3390/s21175715>

Academic Editor: Hanseok Ko

Received: 3 August 2021

Accepted: 24 August 2021

Published: 25 August 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



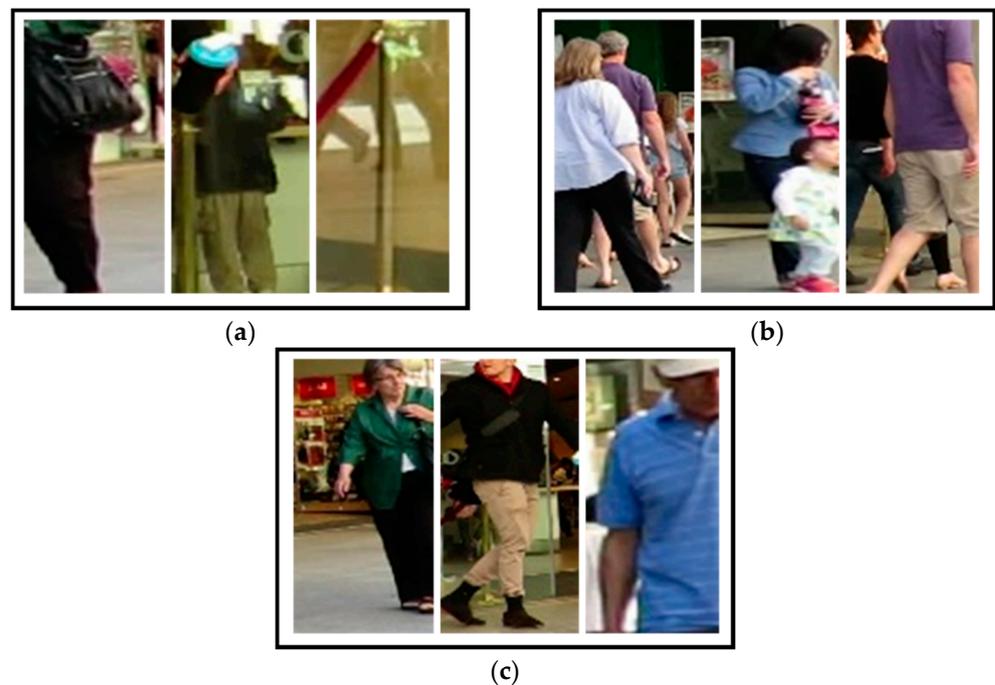
**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Recent advances in autonomous driving, intelligent robots, and smart video surveillance systems have evidenced multi-object tracking (MOT), which aims to estimate the trajectories of multiple objects of interest identified over time in a video sequence, as one of the most important computer vision tasks [1,2]. The predominant approach developed for MOT over the past few years is a new paradigm that identifies objects based on deep neural networks (DNNs) and draws trajectories by analyzing target identities in videos. Most MOT algorithms, designed based on a tracking-by-detection method, have achieved significant performance improvement over conventional methods using feature descriptors, such as the scale-invariant feature transform [3] or histogram of oriented gradients [4].

A tracking-by-detection framework is mainly divided into two phases: the object detection step, which locates the target in the current frame of a video, and the association of all detections throughout the frames and existing trajectories. Because DNNs can secure high object detection performance [5,6], most of the recently proposed MOTs focus on solving the data association problem to describe the path of each object over time by assigning an object identification number and clustering tracklets. Tracklets are a small set of paths associated with individual detections in consecutive frames [7,8]. Various techniques, such as prediction filters, graph-based optimizers, and shortest path methods, are generally combined to improve the efficiency of the data association process. In this process, the similarity of the predicted objects is analyzed, a trajectory is created, and an identity number is assigned when the objects detected in succession are the same. Recently, DNNs have been used as embedding vector extractors for object similarity determination [9,10].

Despite such significant technological advances, some challenges still need to be resolved. The first typical problem is false detection, which results in incorrectly drawing the trajectory of a detected object. This may be caused by the low reliability of the detector, where detected objects are regarded as noise in the association process, resulting in lower tracking performance and increased unnecessary computation. The use of DNNs has resulted in a decrease in the false detection rate, but false detection is still an important issue arising from the influence of the surrounding environment or noise. The second problem is the temporal occlusion caused by obstacles or joint paths, which is a consequence of the limited field of view (FoV) of a video camera. As a result, the trajectory of the object is separated owing to the loss of detections, which is called fragmentation (Frag), and two trajectories are created in one ground truth because of the transfer of the ID by an adjacent object, which is called ID switch (IDS). Finally, there is the noisy detection problem. The MOT estimates the trajectory of the target by connecting the bounding box of the object detected in the current frame with that of the object detected in the previous frame. In this process, owing to view variations due to weather, spatial misalignment, changes in the size and position of various objects, etc., the detector unintentionally predicts the bounding box, including noise. Figure 1 shows examples of the MOT related problems described above.



**Figure 1.** Examples of cases that make object detection difficult when tracking multiple objects, causing problems with accuracy: (a) false detection, (b) temporal occlusion, (c) noisy detection.

Because the aforementioned problems are inevitable, creating an ideally compact bounding box to perfectly estimate when the detector finds the object and that is not affected by the FoV is not possible. Recent emerging state-of-the-art techniques concentrate on raising the trajectory completeness to the limit where the problems caused by Frag or IDS [11] can be offset. Nevertheless, these obstacles can still cause fatal statistical errors in practical fields that require specialized statistical analysis indicators, such as systems that analyze pedestrian behaviors or track specific objects. This should be a future direction for improving the MOT technique. In this regard, we designed a scheme similar to the existing graph model to solve the Frag and IDS problems in practical applications.

The graph-based tracking algorithm has been implemented by associating object trajectories between frames using graphs [12,13] or associating clustered trajectories for a short period of time [14]. The object in the graph-based model is defined as a node, and the similarity between objects is defined as the edge weight. Therefore, the similarity

between the interconnected nodes was determined based on their edge weights. The optimization of the edge weights has been conducted to minimize the total cost by applying the distance between objects [15,16], joints [17], and motions [18]. The graph-based tracking algorithm has been applied to various weighted graph models and has shown superiority in short tracklet connections. However, a limitation arises when trying to connect a path over a wide area, and the problem of overlapping objects remains because the pertinent feature of the object expressed in a real situation is not considered. In addition, the computational amount of the affinity matrix becomes excessive during the process of extracting and optimizing the edge weight, hindering real-time implementation [19].

The MOT technique based on the features of the detected object was proposed after the development of the DNN, and it was followed by feature vector embedding [20,21]. Although various similarity models have been proposed, such as modifying the convolutional neural network (CNN) structure [22,23] or vector embedding through fusion with long short-term memory [24,25], many studies still use feature vectors and graph models extracted through reidentification-based CNN [26,27]. They have higher robustness than graph models in an environment where temporal occlusion or noise exists, but vast amounts of data must be secured for learning. In addition, these models do not address the problem of changing IDs between objects when there is considerable overlapping of objects in a crowded environment within the frame or when objects have similar colors.

An IOU Tracker [28] was proposed for object tracking by dividing objects between frames using the intersection-over-union (IOU) threshold of the bounding box. Although its structure is simple, it highly depends on the detection result, and if there is a temporal occlusion, such as when the objects are too close to each other or are partially occluded by other objects, the reliability rapidly decreases. The simple online and real-time tracking (SORT) approach [29] was also proposed and used to estimate the velocity of an object and predict its position in the next frame. After prediction through the Kalman filter, the predicted and detected values are associated with the Hungarian algorithm [30] to complete the trajectory of the object. This approach is similar to the IOU tracker, but it solves the temporal occlusion problem by predicting the object location. Later, similar model-based methods using feature vectors, such as object motion, joints, and optical flows, were proposed and advanced [31–33]. However, limitations in the predictive model make it vulnerable to noisy environments and false detections, and the structural complexity is increased.

In this study, we propose a density-based tracklet association annealer (DTAA)—a novel tracking-by-detection technique. This technique aims to improve the vulnerability to noise and difficulty of trajectory separation inherent to MOT and to secure real-time processing through simple structural expansion. In addition, it can effectively reduce IDS and Frag in tracking. The proposed DTAA extends the clustering mechanism of density-based spatial clustering of applications with noise (DBSCAN) [34], which has high noise robustness and is excellent for nonlinear clustering, to a graph model. Although DBSCAN is a spatial data clustering technique, we have proven in prior research [35] that an adaptive graph clustering technique based on DBSCAN shows high performance in clustering real-world data. In addition, we confirmed that the spatial data clustering technique could be analyzed in the coordinate system of two-dimensional (2D) images in a previous study [36]. Based on the results of these preliminary studies, the proposed DTAA combines the coordinates of the bounding box with the feature vectors of the detected objects and extracts them as embedding vectors for clustering. The reduced-dimensional feature vectors are clustered into highly similar tracklets via DBSCAN, and heterogeneous tracklets or single objects are eliminated. The workflow of the proposed tracking scheme is shown in Figure 2.

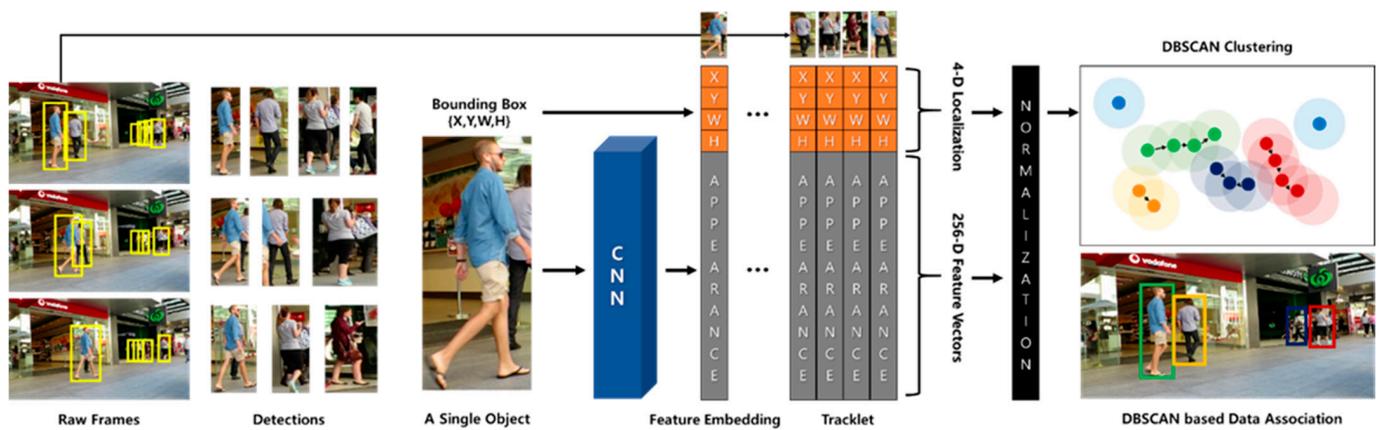


Figure 2. Workflow of the proposed MOT scheme.

The proposed DTAA has advantages comparable to or more than conventional density-based clustering techniques. First, it is robust against false positives. DTAA can filter false detections from clusters in real time. Second, with a relatively simple structure, it can be used independently or as a postprocessor to rebuild the existing MOT techniques, in which the Frag and IDS indicators can be significantly reduced. Finally, because the feature vectors are created through the fusion of the location and similarity of the identified object, spatial features and similarities are used for end-to-end clustering. To evaluate the robustness of the proposed tracking strategy, the MOT15-16 [37,38] benchmarks were used with the addition of various artificial noises, and a comparative evaluation was performed with the latest techniques using DNN. As a result, when the proposed DTAA was added to the existing tracking technique as a postprocessor, the performance of the existing tracking system was improved by more than 5% on average, and the Frag and IDS indices were decreased by more than 30–40%. It was further confirmed by testing in the presence of additional noise that the robustness against noise was improved compared to the existing tracking technique. In addition, in all experiments conducted in NVIDIA GTX 1080Ti and Intel Core i7-8700 CPU environments, it showed a near real-time processing speed of 20 frames per second (FPS) or more, and it is expected that the real-time anomaly detection system will be used in practical applications in the future.

The contributions of this study are summarized as follows:

1. A new density-based tracklet association enhancement method for improving the performance of “multi-object tracking by detection” is proposed.
2. A clustering technique suitable for multi-object tracking is introduced that combines the reduced-dimensional feature vectors obtained through CNN into very similar tracklets using DBSCAN.
3. When tracking multiple objects, the severe vulnerability to small external interference and the difficulty of separating multi-object trajectories were improved, and real-time multi-object tracking became possible through simple structure extension.

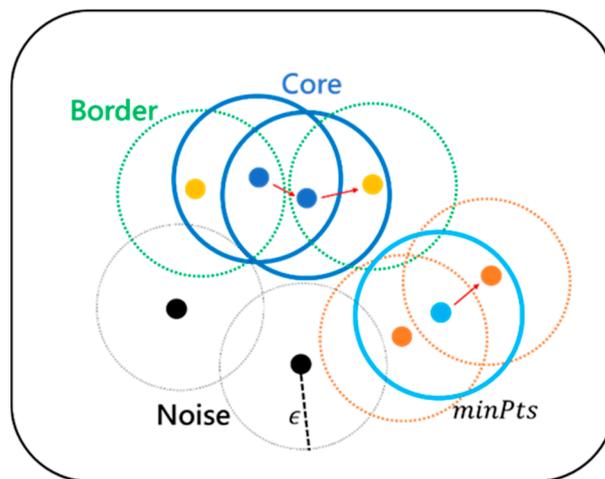
## 2. Density-Based Tracklet Association

The proposed DTAA extracts object features through conventional person re-identification (re-ID) utilizing a CNN. It is composed of the edges of the track graph by embedding the extracted features and object bounding box coordinates. In the process of data association, DBSCAN is applied instead of a graph model to solve the problem of computational complexity of the similarity matrix through edge clustering rather than graph model optimization. As a result, it is feasible to collect tracks for a longer period of time than that of the existing computational amount, and tracks that are considered unnecessary noise in real-time clustering are deleted. Therefore, the existing optimization problem can be approached more reasonably and efficiently.

### 2.1. DBSCAN

DBSCAN is a method of clustering data points that share common attributes based on the density of data, unlike most techniques that incorporate similar entities based on their data distribution. This means that clusters are defined as events occurring in the same space. This method is suitable for the clustering of multidimensional and spatial data based on the density correlation with neighboring clusters, and data containing various sizes, shapes, or noises can also be clustered [39]. Unlike the most commonly utilized k-means clustering, DBSCAN does not require the number of clusters in advance, and it receives only two hyperparameters. One is the minimum neighboring radius,  $\epsilon$ , which means the area in density and is defined as the distance from which data is viewed as a neighbor. Assuming that there are two data points  $x$  and  $x'$ , and the distance between them is  $d$ , if the condition  $d(x, x') \leq \epsilon$  is satisfied,  $x$  and  $x'$  are included in the same cluster. The other is  $minPts$ , which is the minimum number of points in the area defined above. A cluster is then formed when more than the minimum number of points is satisfied within a specific area based on the minimum neighboring radius.

The cluster is expanded by repeating this process around a vector neighboring the newly formed cluster. Here, a central vector that contributes to cluster formation is defined as a core vector, which refers to a case in which the number of neighboring vectors within the radius  $\epsilon$  from an arbitrary vector is higher than  $minPts$ . A vector that is a neighbor of the core vector but cannot itself become a central vector is defined as a border vector. It is located within the distance  $\epsilon$  from the core vector and is classified into the same cluster, but it is a vector located outside the cluster. A vector that is not a core or outer vector, that is, a vector whose number of neighboring vectors is lower than the number of  $minPts$  within  $\epsilon$  and does not belong to any cluster, is considered to be a noise vector. A description of these vectors is presented in Figure 3. Because DBSCAN clusters use the density of neighboring nodes, they are able to suppress noise by identifying them as noise if the data are outside the radius or do not satisfy the minimum number of neighbors.

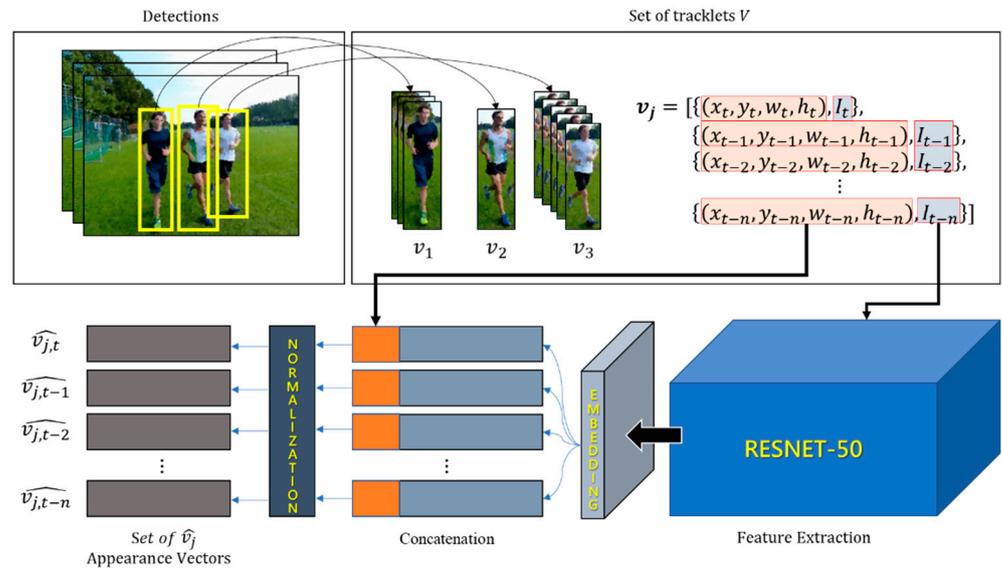


**Figure 3.** Clustering element vectors of DBSCAN.

### 2.2. Appearance Feature Extraction

The set  $V$  of the detected objects is defined as the union set of tracklets consisting of object  $j$  detected in frames  $t$  to  $t-n$ . Here,  $t$  and  $n$  are defined as the current frame index and the  $n$ -th previous index, respectively, and  $j$  is the index assigned to the object. The object's tracklet,  $v_j$ , consists of the coordinates  $(x, y, w, h)$  of the bounding box of the detected object and its raw image frame. RESNET-50 [40] was used to extract the appearance feature vectors for an application as inputs for DBSCAN. The size of the feature map created by RESNET-50 was  $IW \times IH \times 2048$ , in which  $IW$  and  $IH$  are the width and height of the output vector in proportion to the input image size, respectively. Although the size of

the input of DBSCAN should be fixed, the output of RESNET-50 is dependent on the size of the input image, and the extracted feature vectors have high-dimensional channels of 2048, which is not suitable for effective clustering of the tracklets because of its low computational efficiency. Therefore, to build an end-to-end learning system by extracting the feature map more efficiently and reducing the input dimension, we added the vector embedding layer proposed by the triplet network [41] to the output terminal of RESNET-50. Figure 4 shows the appearance feature vector extraction and preprocessing flow for use as the DBSCAN input.



**Figure 4.** Preprocessing of raw frames for tracklet association by DBSCAN.

The added neural network, a vector embedding layer, is a fully connected layer structure that is designed to reduce the output dimension to  $\mathbb{R}^{256 \times 1}$ , considering the computational efficiency of DBSCAN. Here, RESNET-50 uses previously learned weights, and the embedding layer performs fine-tuning based on the same triple loss used in the triplet network. In the track-clustering process, the extracted feature vector is concatenated with the bounding box coordinates of the object to include the location information of the object and then processed as normalization to generate  $\hat{v}_j \in \mathbb{R}^{260 \times 1}$  for DBSCAN input. Thereafter, for clustering using  $\hat{V}$ , which is a set of  $\hat{v}_j$  vectors, the input is assumed to be a spatial coordinate system and used in the data association step.

### 2.3. Tracklets Association Using DBSCAN

The proposed DTAA is a method of clustering object tracklets based on DBSCAN, as discussed above. After obtaining the input vector of  $\hat{v}_j \in \mathbb{R}^{260 \times 1}$  for DBSCAN, data association is performed by clustering using DBSCAN. The object tracklet association process using DBSCAN for the proposed DTAA is shown in Figure 5.

DTAA, which is transformed into a graph-based clustering method for MOT, uses  $\epsilon^*(t)$ . Temporal evolution is added to  $\epsilon^*(t)$ , as shown in (1), by considering the change over time to the minimum neighboring radius  $\epsilon$ , a parameter of the existing DBSCAN, in the clustering process.

$$\epsilon^*(t) = \mu e^{-\frac{(t-1)^2}{l^2}} \quad (1)$$

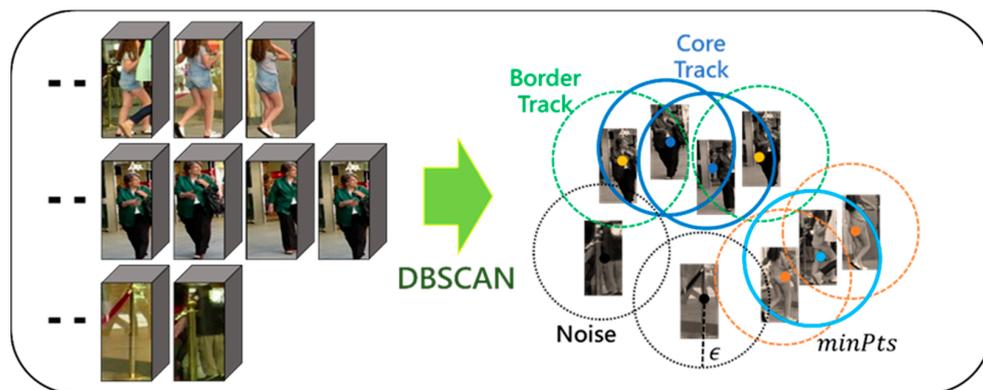


Figure 5. Illustration of object tracklets association process using DBSCAN.

Here,  $l$  is the number of tracks constituting the object set  $V$ , and  $\mu$  is the radius adjustment parameter. Objects detected long ago can be given low similarity by using  $\epsilon^*(t)$  and can be transformed into the form of a Gaussian distribution. A track that satisfies the minimum neighboring  $minPts$  or higher based on  $\epsilon^*(t)$  is defined as a core track, and a track that is clustered with the minimum neighboring but fails to expand below the standard is defined as a border track. The proposed tracklet association process is described in detail in Algorithm 1.

The algorithm attempts to cluster when a new object or tracklet  $v_j$  of an object  $j$  is applied as an input. In sequence, the algorithm calculates the distance to each object through the same RangeQuery function as in the conventional DBSCAN, and these objects are formed into a cluster when the number of neighboring objects is  $minPts$  or more within the radius  $\epsilon^*$ . After that, the objects not included in the cluster are regarded as noise and removed from set  $V$ . Figure 6 shows the flow of tracking multiple objects through the DTAA from object detection to clustering.

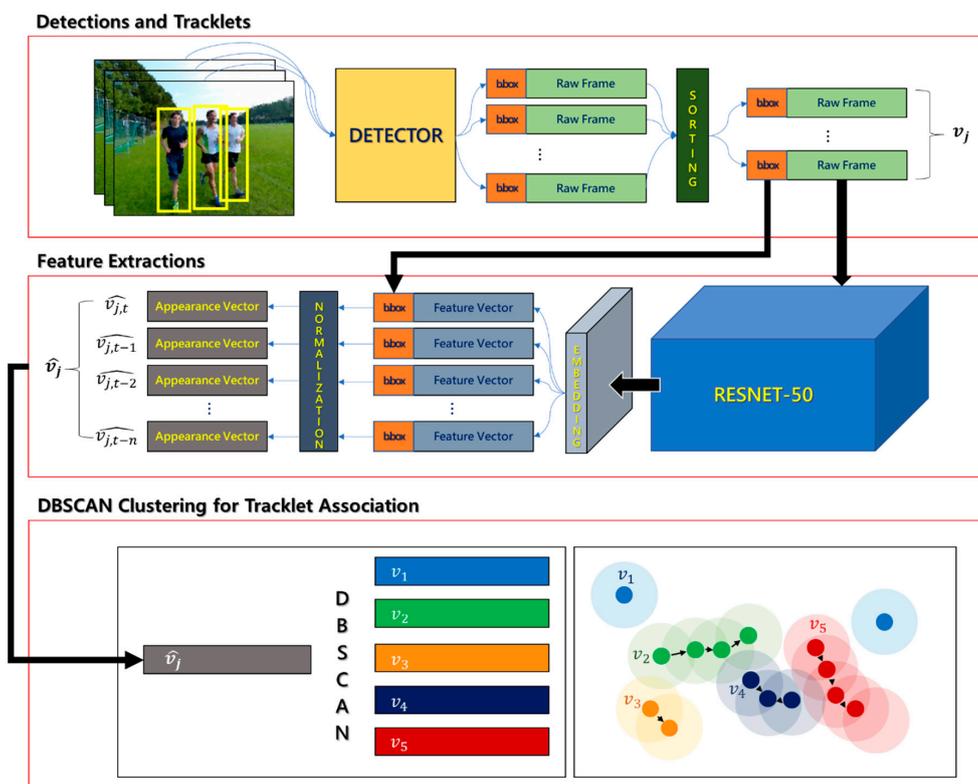


Figure 6. DBSCAN overall flow of multi-object tracking by the proposed DTAA.

**Algorithm 1.** Strengthening tracklet association process through DBSCAN clustering

---

```

Input: V (Set of tracklets)
Input: new v (tracklet)
Input: minPts (Density threshold)
Input:  $\epsilon^*$  (Neighboring radius)
Input: dist (L2 norm, Distance of function)
Input: label (Object labels, initially undefined)
01 if new v  $\neq$  undefined
02     V(end+1)  $\leftarrow$  new v // Append new tracklet
03     for each v in V // Iterate over every point
04         if label(v) undefined then continue
05         for each f in v
06             Neighbors N  $\leftarrow$  RangeQuery(V, dist, f,  $\epsilon^*$ ) // Find initial neighbors
07             if |N| < minPts then
08                 label(f)  $\leftarrow$  Noise
09                 continue
10             c  $\leftarrow$  next cluster label

11             label(f)  $\leftarrow$  c
12             Seed set S  $\leftarrow$  N \ {f} // Expand neighborhood
13             for each q in S do
14                 if label(q) = Noise then label(q)  $\leftarrow$  c
15                 if label(q)  $\neq$  undefined then continue
16                 Neighbors N  $\leftarrow$  RangeQuery(V, dist, q,  $\epsilon^*$ ) // Find initial neighbors
17                 label(q)  $\leftarrow$  c
18                 if |N| < minPts then continue // Core track check
19                 S  $\leftarrow$  S  $\cup$  N
20 V(find(label == Noise))  $\leftarrow$  None // Delete noise

```

---

**3. Experimental Results**

The MOTChallenge 2015-16 (MOT15-16) benchmark dataset was used to evaluate the performance of the DTAA proposed in this study. Owing to the lack of more accurate object detection results using DNNs, the MOT15 benchmark is generally rated for low performance, even with state-of-the-art methods showing excellent performance in object tracking [42]. This dataset contains 11 training and test scenarios and evaluates the trajectory completeness for pedestrians. The training data included 500 object tracks and 39,905 bounding boxes. In contrast, the MOT16 dataset consists of 14 scenarios, 1342 tracks, and 292,733 bounding boxes. Standard MOT metrics were used for the evaluation.

The MOT accuracy (MOTA), a representative MOT performance index, comprehensively evaluates the missed detection rate, false alarm rate (FAR), and IDS, as shown in (2).

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad (2)$$

Here,  $m_t$ ,  $fp_t$ , and  $mme_t$  are three error sources, defined as the number of missing objects, number of false positives, and number of mismatch errors in the  $t$ -th frame.  $g_t$  is the number of ground truths of the object to be detected. The MOTA metric is the most representative indicator for evaluating the MOT; however, as mentioned in [43], the following limitations exist. Because the threshold for object detection is very sensitive to the ground truth, it becomes too dependent on detections, and it is not easy to verify the overall robustness against noise such as Frag or IDS indices. Accordingly, we first compared the tracking performance based on MOTA and additionally conducted an evaluation focusing on the number of Frag, IDS, and FAR.

In addition, performance comparisons with conventional tracking methods were performed. To confirm the possibility of expanding the use of the proposed method, the DTAA was added to the end of the existing tracking algorithms in the form of late fusion as a postprocessor, and the verification task of its performance improvement was also conducted. In general, IOU Tracker and SORT, which can be easily used for object tracking, and tracking-by-detection techniques such as structural sparse tracking (SST) [44], joint detection and embedding (JDE) [45], and Adopting Tubes to Track Multi-Object in a One-Step Training Model (TubeTK) [46] based on DNNs were selected and compared with these prior works. IOU Tracker, SORT, and SST used the public detection results of the MOT15 benchmark dataset, whereas JDE and TubeTK used private detection results. Default values were used for the parameters and learning weights of the techniques used in the comparative evaluation. The learning for feature extraction through the proposed DTAA was conducted based on the MARKET-1501 dataset, and the following optimized values were derived through strategic learning for the parameters used in the data association: maximum track length of 20,  $\mu = 0.75$ ,  $minPts = 2$ . The comparative evaluation results of the proposed DTAA when used alone as a tracker in MOT and when incorporated with the existing trackers as a postprocessor are shown in Tables 1 and 2, respectively.

**Table 1.** Tracking performance comparisons with MOT15 when the DTAA was used alone; best values are depicted in bold.

Tracker	MOTA $\uparrow$	FAR $\downarrow$	IDS $\downarrow$	Frag $\downarrow$	Hz $\uparrow$
<b>Public</b>					
SORT [27]	26.0	1.20	779	1171	100+
IOU Tracker [28]	25.8	1.53	689	1120	100+
SST [44]	31.5	1.86	1262	1542	10
DTAA(ours)	26.8	<b>0.93</b>	<b>421</b>	879	30
<b>Private</b>					
JDE [45]	<b>35.5</b>	3.68	520	<b>823</b>	15
TubeTK [46]	59.6	1.02	858	1103	6

When tracking objects using the DTAA alone, the highest level of MOTA could not be achieved, but FAR, IDS, and Frag were significantly lowered, which was our objective. In particular, the level of MOTA of the proposed tracker was similar or slightly higher than that of SORT and the IOU Tracker, whereas the indicators of FAR, IDS, and Frag showed significantly lower values, confirming that the tracking performance was considerably improved. Moreover, when DTAA was used as a postprocessor, the MOTA index improved by 2–10%, FAR decreased by up to 25% or more, and IDS and Frag decreased by almost 50%. In addition to improving the performance of existing tracking algorithms through the filtering effect, the FPS reduction rate as the amount of calculation increases was

approximately 10% to 20%, confirming that the calculation speed did not significantly decrease owing to the additional process of DTAA.

**Table 2.** Tracking performance comparisons when the DTAA was added to the state-of-the-art trackers as a postprocessor.

Tracker	MOTA $\uparrow$	FAR $\downarrow$	IDS $\downarrow$	Frag $\downarrow$
<b>MOT15 [36]</b>				
<b>Public</b>				
SORT [27] + DTAA	27.5 (+5.7%)	1.00 (−16.7%)	442 (−48.4%)	879 (−4.9%)
IOU Tracker [28] + DTAA	26.4 (+2.3%)	1.42 (−7.2%)	518 (−24.8%)	751 (−2.9%)
SST [44] + DTAA	33.7 (+7.0%)	1.39 (−25.3%)	717 (−3.2%)	1112 (−7.9%)
<b>Private</b>				
JDE [45] + DTAA	39.1 (+10.1%)	3.31 (−10.1%)	374 (−28.1%)	685 (−6.8%)
TubeTK [46] + DTAA	58.4 (−0.2%)	0.98 (−4.7%)	798 (−32.2%)	851 (−7.8%)
<b>MOT16 [37]</b>				
<b>Public</b>				
SORT [27] + DTAA	22.6 (−)	2.15 (−60%)	1366 (−65%)	4713 (−53%)
IOU Tracker [28] + DTAA	27.5 (+1.3%)	0.20 (−12%)	751 (−21%)	841 (−13%)
SST [44] + DTAA	27.9 (+3.4%)	1.30 (−14%)	1095 (−14%)	2786 (−6.2%)
<b>Private</b>				
JDE [45] + DTAA	72.8 (−0.9%)	1.05 (−16%)	1248 (−6.3%)	1510 (−32.1%)
TubeTK [46] + DTAA	73.5 (+0.1%)	1.02 (−12.1%)	653 (−12.2%)	1123 (−8.4%)

The result of expanding the proposed DTAA shown in Table 2 is remarkable. It is possible to perform more complete object tracking by removing noise with the DTAA from the trajectory of the object derived from the existing system without placing a large load on the data throughput. In all of the latest techniques, SORT and IOU Tracker, FAR, IDS, and Frag significantly decreased, whereas MOTA slightly increased. Even in the MOT techniques based on the private detection result, which have excellent detection performance, the tracking performance was similarly improved in terms of MOTA when the proposed DTAA was added as a postprocessor. Therefore, it was verified that even in the latest techniques, DTAA can be an augmentation system that can increase the robustness of the MOT algorithms and maximize the performance. Figure 7 shows some of the test evaluation results of the SST technique supplemented with DTAA.

An additional evaluation was conducted to verify the robustness against noise based on the processed data created by arbitrarily inserting noise into the previously used MOT15. To simulate the environment for temporal occlusion, data representing missed detection were created by randomly removing the detection results, and data representing noisy detection were created by adding noise to the bounding box of the detected object. In the data representing missed detection, the bounding box of the detected object was deleted with the probability of  $\alpha\%$ , and in the data representing the noisy detection,  $\beta\%$  pixels were added to the bounding box of the detected object according to the intensity. Tables 3 and 4 show the results of the robustness evaluation of SORT, IOU Tracker, and SST and their results when the proposed DTAA was added. As a result of all evaluations in a noisy environment, it was possible to maintain higher robustness, and the MOTA was increased by 5–10% on average compared to the existing method, regardless of the noise intensity or method, and IDS and Frag were reduced by approximately 30–40%. Although the overall performance was low owing to the effect of noise, practical applications are expected to achieve a large performance improvement.

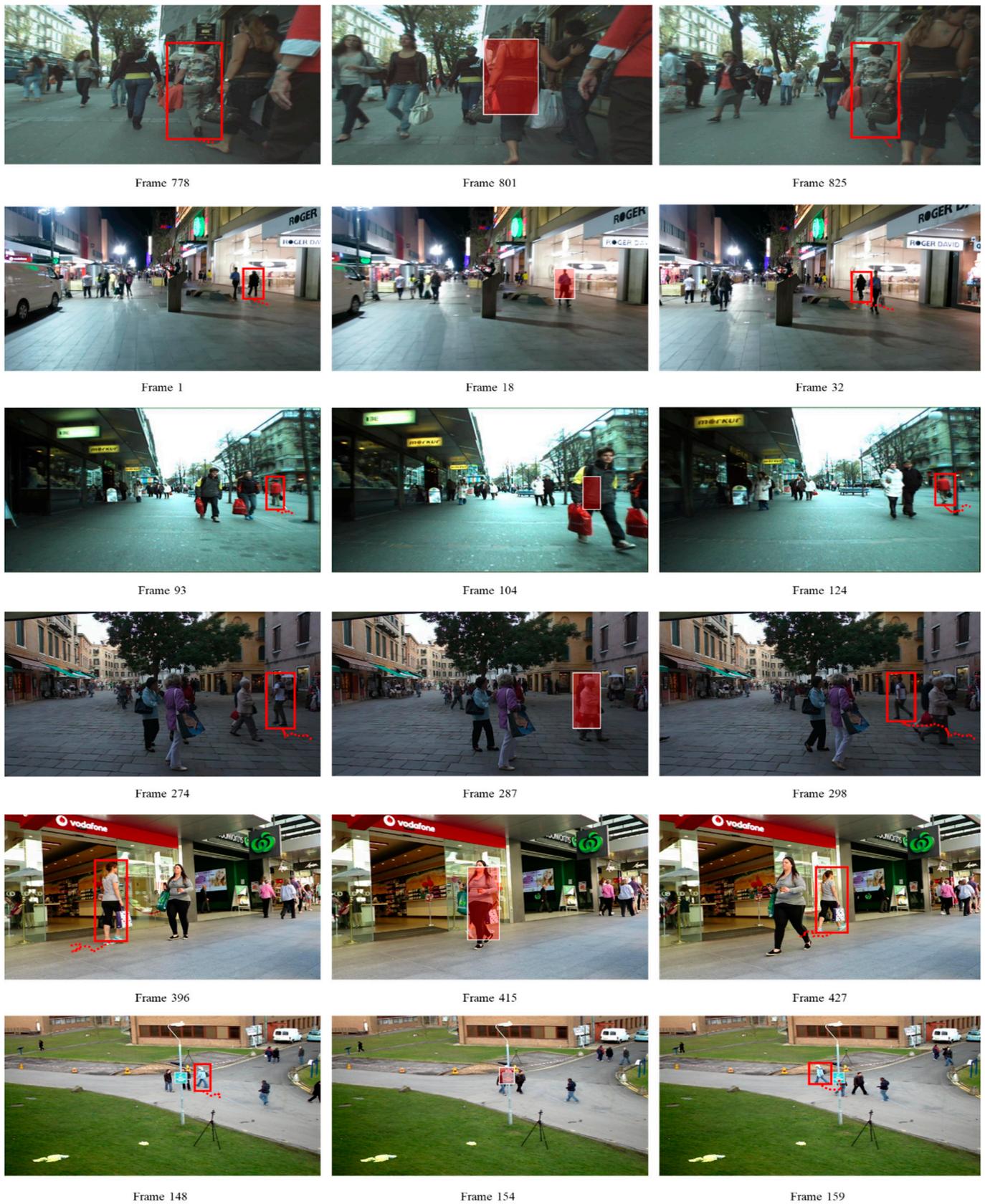


Figure 7. Exemplary tracking results using SST + DTAA (ours).

**Table 3.** Comparisons of tracking performance when the DTAA was added to the state-of-the-art methods under noisy conditions considering missed detection with  $\alpha = 10\%$ .

Tracker	MOTA $\uparrow$	FAR $\downarrow$	IDS $\downarrow$	Frag $\downarrow$
SORT [27]	18.6	0.90	859	1906
IOU Tracker [28]	19.9	1.17	1306	1722
SST [44]	23.2	1.38	1240	1682
SORT [27] + DTAA	18.6 (–)	0.90 (–)	707 (–17.7%)	1477 (–22.5%)
IOU Tracker [28] + DTAA	22.3 (+12.1%)	1.01 (–13.7%)	503 (–61.5%)	1566 (–9.1%)
SST [44] + DTAA	24.9 (+7.3%)	1.10 (–20.3%)	761 (–38.6%)	1432 (–14.9%)

**Table 4.** Comparisons of tracking performance when the DTAA was added to the state-of-the-art methods under noisy conditions considering noisy detection with  $\beta = \pm 15\%$ .

Tracker	MOTA $\uparrow$	FAR $\downarrow$	IDS $\downarrow$	Frag $\downarrow$
SORT [27]	12.8	0.70	954	1454
IOU Tracker [28]	7.5	1.41	1192	2340
SST [44]	13.8	0.84	1046	1762
SORT [27] + DTAA	14.4 (+12.5%)	0.40 (–42.8%)	399 (–58.2%)	788 (–45.8%)
IOU Tracker [28] + DTAA	10.8 (+44.0%)	1.12 (–20.6%)	676 (–43.3%)	1977 (–15.5%)
SST [44] + DTAA	15.1 (+9.4%)	0.76 (–9.5%)	785 (–24.9%)	1482 (–15.9%)

#### 4. Conclusions

Recently, MOT techniques have attracted attention owing to the breakthrough development of autonomous driving and smart video surveillance systems. The main approach developed for tracking is a paradigm that identifies objects based on DNNs and tracks objects by analyzing the target identity in a video. Although significant technological advances have been made, it is still difficult to track objects in an environment in which occlusion is present. In this paper, we propose a strategy to enhance the performance of existing tracking techniques while solving the problem of noise vulnerability and operational efficiency degradation occurring in MOT. The proposed DTAA integrates the object feature vector and bounding box coordinates extracted through the CNN into a low-dimensional vector, mimics the graph model at the data association step, and clusters it with DBSCAN to estimate the trajectory of the object. Through test evaluation, when the proposed technique was used as a tracker alone, its performance indices of IDS and Frag were improved when compared with the existing tracking techniques. In particular, when additionally incorporated with conventional trackers as a postprocessor, significant performance improvement and noise suppression were observed. Thus, applications in industrial environments, such as actual pedestrian analysis and surveillance security systems, are expected.

However, the computational efficiency decreases as the number of objects to be tracked increases, and the high IDS between similar objects remains a problem to be solved. In order to make the proposed tracker more stable and reliable, we plan to increase the learning efficiency of CNN through the development of the scheme that effectively reduces the dimension of the embedding vector, and continue research on how to incorporate graph-based data association techniques into the clustering process.

**Author Contributions:** J.K. and J.C. took part in the discussion of the work described in this paper. Both authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MOE) (No. 2021R111A3055973) and the Soonchunhyang University Research Fund.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

**Conflicts of Interest:** The authors declare that they have no competing interests.

## References

1. Ciaparrone, G.; Sánchez, F.L.; Tabik, S.; Troiano, L.; Tagliaferri, R.; Herrera, F. Deep learning in video multi-object tracking: A survey. *Neurocomputing* **2020**, *381*, 61–88. [\[CrossRef\]](#)
2. Brasó, G.; Leal-Taixé, L. Learning a neural solver for multiple object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 6247–6257.
3. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [\[CrossRef\]](#)
4. 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, San Diego, CA, USA, 20–25 June 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 1, pp. 886–893.
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 7263–7271.
7. Zhang, Y.; Sheng, H.; Wu, Y.; Wang, S.; Lyu, W.; Ke, W.; Xiong, Z. Long-term tracking with deep tracklet association. *IEEE Trans. Image Process.* **2020**, *29*, 6694–6706. [\[CrossRef\]](#)
8. Peng, J.; Wang, T.; Lin, W.; Wang, J.; See, J.; Wen, S.; Ding, E. TPM: Multiple object tracking with tracklet-plane matching. *Pattern Recognit.* **2020**, *107*, 107480. [\[CrossRef\]](#)
9. Leal-Taixé, L.; Canton-Ferrer, C.; Schindler, K. Learning by tracking: Siamese CNN for robust target association. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 33–40.
10. Voigtlaender, P.; Krause, M.; Osep, A.; Luiten, J.; Sekar, B.B.G.; Geiger, A.; Leibe, B. Mots: Multi-object tracking and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 7942–7951.
11. Bergmann, P.; Meinhardt, T.; Leal-Taixé, L. Tracking without bells and whistles. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 941–951.
12. Janai, J.; Güney, F.; Behl, A.; Geiger, A. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Found. Trends@Comput. Graph. Vis.* **2020**, *12*, 1–308. [\[CrossRef\]](#)
13. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T.K. Multiple object tracking: A literature review. *Artif. Intell.* **2020**, *293*, 103448. [\[CrossRef\]](#)
14. Berclaz, J.; Fleuret, F.; Fua, P. Robust people tracking with global trajectory optimization. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; Volume 1, pp. 744–750.
15. Pirsiavash, H.; Ramanan, D.; Fowlkes, C.C. Globally-optimal greedy algorithms for tracking a variable number of objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1201–1208.
16. Zhang, S.; Huang, J.B.; Lim, J.; Gong, Y.; Wang, J.; Ahuja, N.; Yang, M.H. Tracking persons-of-interest via unsupervised representation adaptation. *Int. J. Comput. Vis.* **2020**, *128*, 96–120. [\[CrossRef\]](#)
17. Tang, S.; Andriluka, M.; Andres, B.; Schiele, B. Multiple people tracking by lifted multicut and person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 3539–3548.
18. Choi, W.; Savarese, S. A unified framework for multi-target tracking and collective activity recognition. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 215–230.
19. Sharma, S.; Ansari, J.A.; Murthy, J.K.; Krishna, K.M. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–26 May 2018; pp. 3508–3515.
20. Kim, C.; Li, F.; Ciptadi, A.; Reh, J.M. Multiple hypothesis tracking revisited. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4696–4704.
21. Mahmoudi, N.; Ahadi, S.M.; Rahmati, M. Multi-target tracking using CNN-based features: CNNMTT. *Multimed. Tools Appl.* **2019**, *78*, 7077–7096. [\[CrossRef\]](#)

22. Huang, C.L. Exploring effective data augmentation with TDNN-LSTM neural network embedding for speaker recognition. In Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, 14–18 December 2019; pp. 291–295.
23. Ullah, M.; Cheikh, F.A. Deep feature based end-to-end transportation network for multi-target tracking. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 3738–3742.
24. 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; pp. 466–475.
25. Kim, H.U.; Koh, Y.J.; Kim, C.S. Online Multiple Object Tracking Based on Open-Set Few-Shot Learning. *IEEE Access* **2020**, *8*, 190312–190326. [[CrossRef](#)]
26. Ristani, E.; Tomasi, C. Features for multi-target multi-camera tracking and re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6036–6046.
27. Zhang, Z.; Wu, J.; Zhang, X.; Zhang, C. Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project. *arXiv* **2017**, arXiv:1712.09531.
28. 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.
29. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, Arizona, 25–28 September 2016; pp. 3464–3468.
30. Zhu, H.; Zhou, M. Efficient role transfer based on Kuhn-Munkres algorithm. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2011**, *42*, 491–496. [[CrossRef](#)]
31. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
32. Fu, H.; Wu, L.; Jian, M.; Yang, Y.; Wang, X. MF-SORT: Simple online and Realtime tracking with motion features. In Proceedings of the International Conference on Image and Graphics, Beijing, China, 23–25 August 2019; pp. 157–168.
33. Xiao, B.; Wu, H.; Wei, Y. Simple baselines for human pose estimation and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 466–481.
34. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
35. Kim, J.; Cho, J. Delaunay triangulation-based spatial clustering technique for enhanced adjacent boundary detection and segmentation of LiDAR 3D point clouds. *Sensors* **2019**, *19*, 3926. [[CrossRef](#)] [[PubMed](#)]
36. Kim, J.; Cho, J. An online graph-based anomalous change detection strategy for unsupervised video surveillance. *EURASIP J. Image Video Process.* **2019**, *2019*, 76. [[CrossRef](#)]
37. 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.
38. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A benchmark for multi-object tracking. *arXiv* **2016**, arXiv:1603.00831.
39. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Trans. Database Syst. (TODS)* **2017**, *42*, 1–21. [[CrossRef](#)]
40. Peng, S.; Huang, H.; Chen, W.; Zhang, L.; Fang, W. More trainable inception-ResNet for face recognition. *Neurocomputing* **2020**, *411*, 9–19. [[CrossRef](#)]
41. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
42. Hornakova, A.; Henschel, R.; Rosenhahn, B.; Swoboda, P. Lifted disjoint paths with application in multiple object tracking. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 12–18 July 2020; pp. 4364–4375.
43. Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; Tomasi, C. Performance measures and a data set for multi-target, multi-camera tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 17–35.
44. Sun, S.; Akhtar, N.; Song, H.; Mian, A.; Shah, M. Deep affinity network for multiple object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 104–119. [[CrossRef](#)] [[PubMed](#)]
45. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards real-time multi-object tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 107–122.
46. Pang, B.; Li, Y.; Zhang, Y.; Li, M.; Lu, C. Tubetk: Adopting tubes to track multi-object in a one-step training model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 6308–6318.