




Article

Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics

Ricardo Pereira ^{*,†} , Guilherme Carvalho [†], Luís Garrote  and Urbano J. Nunes 

Department of Electrical and Computer Engineering, Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal; guicarvalho19@outlook.com (G.C.); garrote@isr.uc.pt (L.G.); urbano@isr.uc.pt (U.J.N.)

* Correspondence: ricardo.pereira@isr.uc.pt

† These authors contributed equally to this work.

Abstract: Multi-Object Tracking (MOT) techniques have been under continuous research and increasingly applied in a diverse range of tasks. One area in particular concerns its application in navigation tasks of assistive mobile robots, with the aim to increase the mobility and autonomy of people suffering from mobility decay, or severe motor impairments, due to muscular, neurological, or osteoarticular decay. Therefore, in this work, having in view navigation tasks for assistive mobile robots, an evaluation study of two MOTs by detection algorithms, SORT and Deep-SORT, is presented. To improve the data association of both methods, which are solved as a linear assignment problem with a generated cost matrix, a set of new object tracking data association cost matrices based on intersection over union, Euclidean distances, and bounding box metrics is proposed. For the evaluation of the MOT by detection in a real-time pipeline, the YOLOv3 is used to detect and classify the objects available on images. In addition, to perform the proposed evaluation aiming at assistive platforms, the ISR Tracking dataset, which represents the object conditions under which real robotic platforms may navigate, is presented. Experimental evaluations were also carried out on the MOT17 dataset. Promising results were achieved by the proposed object tracking data association cost matrices, showing an improvement in the majority of the MOT evaluation metrics compared to the default data association cost matrix. In addition, promising frame rate values were attained by the pipeline composed of the detector and the tracking module.

Keywords: multi-object tracking; data association; autonomous mobile robot platforms



Citation: Pereira, R.; Carvalho, G.; Garrote, L.; Nunes, U.J. Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics. *Appl. Sci.* **2022**, *12*, 1319. <https://doi.org/10.3390/app12031319>

Academic Editor: Alessandro Gasparetto

Received: 22 December 2021

Accepted: 18 January 2022

Published: 26 January 2022

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



Copyright: © 2022 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

Vision-based Multi-Object Tracking (MOT) methods analyze image sequences to establish object correspondences over the images [1,2]. Multiple MOT methods have been proposed over the years and have been widely used in applications, such as surveillance [3], traffic monitoring [4], autonomous driving [5], and mobile robot navigation, including object collision avoidance [6] or target following [7]. However, MOT results may be affected by difficult problem configurations due to crowded environments or occluded objects, which leads to limitations in performance for such scenarios. Moreover, due to a large number of applications where MOT methods can be applied, the importance of MOT is high and remains a challenging topic in the research community [1,2,8].

Throughout the years, MOT tasks were mainly performed by the tracking by detection paradigm [9], where objects were detected by an object detector and fed to the object tracking method, which then dealt with the object association between previous frames and the present one. Most methods proposed [10–12] use a Kalman Filter (KF) as a motion module to predict the position of objects of interest in the current frame. On the other hand, with the emergence of Deep learning-based Neural Networks (DNNs) [13,14], new state-of-the-art methods have been proposed in object vision-based tasks such as object classification [15], recognition [16], and tracking [11,17,18]. Therefore, to improve the object

association step of tracking algorithms, Convolutional Neural Networks (CNNs) have been applied to extract object appearance features, which are used to compute similarity values between two objects' feature maps, extracted over two consecutive images. On the other hand, CNNs have also been used to locate objects to track consecutive images [19,20].

MOT techniques can be employed to improve the motion planning behavior and safety on the navigation tasks of mobile robot platforms [6]. MOT techniques can also be an asset on assistive platforms for target-following tasks, where the platform follows a specific target (e.g., following a caregiver, reaching an object).

Due to several types of impairments, there are a significant number of people unable to perform daily tasks. Hence, a particular type of assistive mobile robot, robotic wheelchair platforms, has been researched aiming to increase the autonomy and mobility of such users [21,22]. Brain-actuated wheelchairs [21,23,24] have also received particular focus in research, with several promising techniques for severely motor disabled people who are unable to control a robotic platform by the conventional interfaces, such as joystick [21,25]. With the advances in Brain-Computer Interfaces (BCI) and shared control methods, new paradigms of the brain-computer interaction that allow the user to choose his navigation target have been proposed. The new paradigms can represent potential goals of interest to the user's navigation (e.g., objects) and can be empowered by considering the tracked objects from MOT methods. Once the user selects its navigation target, a MOT method is required to ensure that robotic wheelchair platforms navigate towards that specific target. However, to endow a mobile robot to pursue an object as its navigation target, a robust visual perception module, including an object tracking method, is required. Moreover, to ensure a robust object tracking performance, detection and tracking should be performed frame-by-frame, which is time-consuming and can lead to the inability of performing MOT in real-time [9].

In this work, considering navigation tasks in assistive platforms, an evaluation study of two multi-object tracking by detection algorithms, SORT [10] and Deep-SORT [11], using new data association metrics [26], is proposed. SORT and Deep-SORT methods were proposed with a focus on real-time object tracking tasks, both achieving state-of-the-art results with a high frame rate. The SORT and Deep-SORT methods share the same overall architecture, divided into three main modules, as shown in Figure 1: KF-based estimation, data association, and track management. To detect objects on the images, the YOLOv3 [16] network is used. Both methods use the KF algorithm to predict the position of the objects in the current frame, which are, as well as the object detections provided by the YOLOv3, the inputs of the data association module, which is a linear assignment problem with a cost matrix association. The SORT method associates objects using bounding box detections to match measurements with predicted tracks, using the overlap of bounding boxes. On the other hand, to improve the bounding box association step, the Deep-SORT uses a CNN to extract appearance features from the object bounding box images. For a detailed evaluation of the object tracking methods, a set of different types of data association cost matrices based on bounding boxes intersection over union, Euclidean distances, and bounding boxes ratios is proposed. To evaluate both tracking methods with the proposed cost matrices, considering an assistive robotics context, the ISR Tracking dataset is proposed. The dataset contains the object conditions from an assistive mobile robot's point of view. The dataset contains 329 object sequences of 9 different object classes. To complement the validation of the SORT and Deep-SORT methods with the proposed cost matrices, evaluation was also performed in the MOT17 [27] dataset.

The main contributions of this work can be summarized as follows:

- Eight new object tracking data association cost matrix formulations based on intersection over union, Euclidean distances, and bounding boxes ratio are proposed.
- The ISR Tracking dataset, presenting a mission performed by a mobile robot in a lab setting, represents the object conditions under which robotic platforms may navigate. It is a rearrangement of the ISR RGB-D dataset [28] with object tracking labels for multi-object tracking tasks.

- An evaluation, having in view navigation tasks for assistive mobile robot platforms, of two multi-object tracking by detection algorithms, SORT and Deep-SORT, is also presented. The proposed new data association cost matrices were integrated and evaluated on both tracking methods.

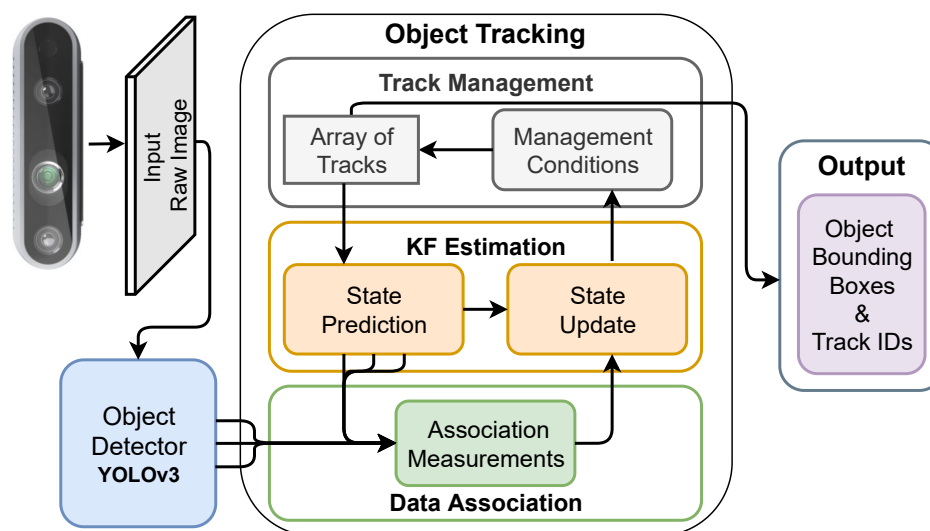


Figure 1. Overview of the Kalman Filter based object tracking algorithms used in this work.

2. Related Work

2.1. Object Tracking

Object tracking techniques have become a fundamental task in real-time video-based applications that require establishing object correspondences between frames [8]. In the literature, proposed tracking techniques fall in two main categories [29]: Single-Object Tracking (SOT) and MOT. In SOT approaches, the appearance of the single target is known a priori, while in MOT techniques, the aim is to estimate trajectories of multiple objects of one or more categories without any prior knowledge about their appearance or location targets. For MOT, an object detection step is required across frames [1]. According to [1], applying multiple SOT models to perform MOT tasks generally leads to poor performance, often caused by similarly looking intra-class objects.

Recent advances in MOT literature have been focusing on two different approaches: tracking by detection and joint tracking and detection. Tracking by detection [10–12,30], as presented in Figure 1, makes use of object detection algorithms to detect and classify objects before performing the object association. This approach simplifies the tracking task as an object association task over consecutive frames. Methods receive an array of measurements and output bounding boxes with their respective tracking ID. On the other hand, joint tracking and detection methods [9,17,19,20] are able to detect and track objects in a single model. Generally, this approach uses visual appearance features of the object to track and locate it in the frames of interest. Joint tracking and detection techniques have become widely popular due to the emergence of the deep learning-based Siamese Networks [18,31].

Despite the promising results achieved by the joint tracking and detection approaches, for navigation tasks in assistive mobile robot platforms, an object detector method can already be available to provide knowledge of the surrounding environment for motion planning or localization methods. Hence, for the purpose of this work, tracking by detection methods are more suitable.

Tracking by Detection

With the emergence of deep learning-based object detectors, tracking by detection has become the most popular approach in the MOT research community [2]. This approach

takes the benefit of object location knowledge to generate an association model that would be able to associate objects over time. One of the first MOT methods found in the literature is Multiple Hypothesis Tracking [32], which calculates hypotheses over measurements to estimate if an object should be associated to a track, be considered as a new track, or if it is a mis-measurement. It uses the KF algorithm to estimate the object's states and a probabilistic distribution over hypotheses to associate measurements to tracks.

Recent works also employ the KF algorithm, as a motion model, to improve the association of objects over time [10–12,33]. Bewley et al. [10] proposed SORT, which is composed of a KF to estimate object states, and by the Hungarian [34] algorithm to associate the KF predictions with new object detections. A year later, Wojke et al. proposed an improvement of SORT, the Deep-SORT [11], by including a novel cascading association step that uses CNN-based object appearance features. The data association algorithm combines the similarity of the object appearance features with the Mahalanobis distance between object states and, at a later stage for unmatched states, uses the SORT's data association. Despite the usage of a CNN, the Deep-SORT method achieved a promising frame rate on the object tracking benchmarks. A method similar to Deep-SORT was proposed by Chen et al., the MOTDT [12]. MOTDT uses a fully CNN-based scoring function for an optimal selection of candidates. Euclidean distances between extracted object appearance features also are used to improve the association step. Recently, He et al. [33] proposed the GMT-CT algorithm that incorporates graph partitioning with deep feature learning. The graph was constructed through the extracted object appearance features, which was used in the association step to model the relationship between measurements and tracks with higher accuracy.

With the growth of deep learning-based Siamese networks in the object tracking community, a new paradigm has been proposed [1]. Lee et al. [35] introduced the FPNS-MOT, which integrates a Siamese architecture with a feature pyramid network [36]. It computes a similarity vector between features from two different inputs and then updates tracks using an interactive selection of the maximum scored pair of tracks and measurements. FPNS-MOT outperformed the aforementioned methods on the MOT challenge benchmarks [27] with an inference time of 10 Hz. Jin et al. [37] enhanced the performance of the Deep-SORT [11] object feature extractor with a Siamese architecture. In addition, it introduced optical flow [38] in the motion module, improving the object association accuracy.

In summary, Table 1 presents the main characteristics of the aforementioned tracking by detection MOT methods.

2.2. Tracking Applied in Mobile Robots

Object tracking techniques have been widely applied for navigation tasks in indoor mobile robot platforms, such as object collision avoidance [6], target following [7], and autonomous navigation [5]. Target detection and tracking have also been applied in robotic wheelchair platforms [39,40], which have been proposed to increase the mobility of people with motor impairments. Xiao et al. [39] proposed a visual-target detection and tracking method to detect and track people in the surroundings of an intelligent wheelchair. The visual tracking was implemented as a binary classification between the object and the background, and a semi-supervised online boosting approach was applied to solve the object drift problem. On the other hand, Lecrosnier et al. [40] proposed an advanced driver assistance system for a robotic wheelchair composed by the YOLOv3 [16] object detection algorithm and a 3D object tracking approach based on SORT [10] to detect and track doors and door handles.

Table 1. Review of state-of-the-art tracking by detection. (DL = Deep Learning-based; KF = Kalman Filter-based).

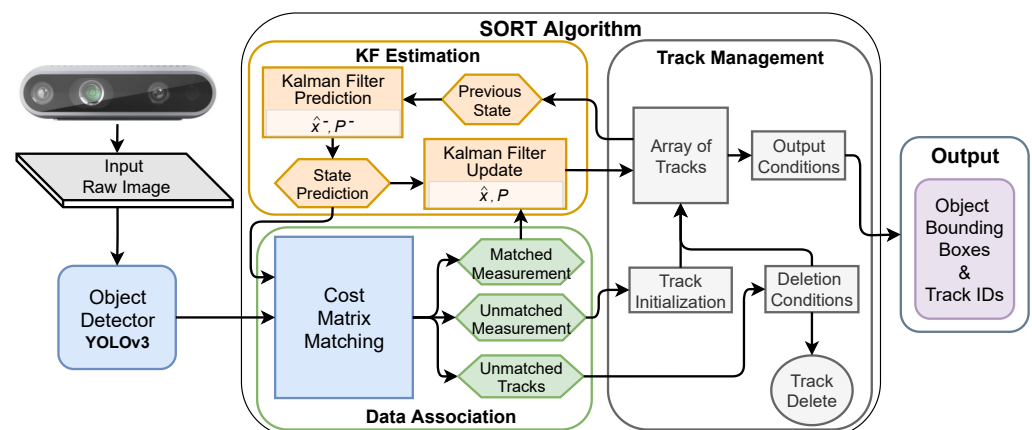
Method	Year	DL	KF	Description
SORT [10]	2016		×	Simple and fast KF-based algorithm that associates objects based on their bounding box appearance.
Deep-SORT [11]	2017	×	×	KF-based algorithm, associates objects based on their appearance description extracted by a CNN re-identification network.
MOTDT [12]	2018	×	×	Deep-SORT related algorithm that uses predicted bounding boxes as candidates for association, in an attempt to solve the occlusion problem.
GMT-CT [33]	2021	×	×	Deep-SORT related algorithm that solves association problems using graph partitioning based on appearance features.
DROP [30]	2020	×	×	Associates objects using a confidence-based cost to construct the Hungarian algorithm solver. Furthermore, it uses appearance features to determine occlusions in the environment.
FPSN-MOT [35]	2019	×		It uses Siamese and Feature Pyramid-based Networks addressing appearance and motion features in the association stage.
Jiating Jin et al. [37]	2020	×	×	Deep-SORT related algorithm that uses Siamese network to process association tasks and also introduce optical flow information to the motion model, in order to improve accuracy.

3. Methodology

In this section, a brief review of the SORT and Deep-SORT methods is presented. The proposed cost matrix formulations, which are part of the data association's linear assignment problem, inside the Cost Matrix Matching module (see Data Association—Cost Matrix Matching in Figures 2 and 3), are also presented.

3.1. SORT

SORT [10] iteratively computes the state of the objects being tracked through a KF. The method uses the Hungarian algorithm [34] to accurately associate detected objects (by an object detector) with objects that are being tracked. A detailed overview of the SORT algorithm is represented in Figure 2.

**Figure 2.** Overview of the object tracking SORT algorithm.

The SORT Data Association module, which is of particular interest in this work, is responsible for matching the KF's predicted bounding boxes with measured bounding boxes on the image, given by the object detector. This module receives, as input, N detected bounding boxes and M predicted bounding boxes (acquired from their respective KF). The module formulates a linear assignment problem by computing a cost matrix between each detected bounding box and all predicted bounding boxes (respectively $D_i, i \in \{1 \dots N\}$ and $P_i, i \in \{1 \dots M\}$), with the Intersection over Union (IoU) as metric:

$$IoU(D, P) = \begin{bmatrix} iou(D_1, P_1) & \dots & iou(D_1, P_M) \\ iou(D_2, P_1) & \dots & iou(D_2, P_M) \\ \vdots & \ddots & \vdots \\ iou(D_N, P_1) & \dots & iou(D_N, P_M) \end{bmatrix} \quad (1)$$

where the IoU between a detected bounding box and a predicted bounding box is given by

$$iou(D_i, P_i) = \frac{D_i \cap P_i}{D_i \cup P_i}. \quad (2)$$

After computing the cost matrix, the Hungarian algorithm [34] is used to associate the bounding boxes. The obtained associations are represented in a $N \times 2$ array, representing N measurements associated to N tracks. Associations are also filtered by considering a minimum IoU threshold, discarding associations with IoU lower than the threshold.

The KF Estimation module uses a linear constant velocity model to represent each object's motion model. When an object is associated with a tracked object (track), its bounding box is used to update the track state. If no object is associated with the track, then the track's state is only predicted. The Track management module is responsible for the creation and deletion of tracks. New Tracks are created when detections do not overlap or overlap with tracks below a minimum IoU threshold. The bounding box of the detection is used to initialize the KF state. Since the only data available are the object's bounding boxes, the object's velocity in the KF is set to zero and its covariance is set high to signal the uncertainty in the state. If a new track does not receive updates because it does not receive associations, or if a track stops receiving associations, they are deleted to avoid maintaining a high number of tracks to false positives or objects that left the scene, respectively.

3.2. Deep-SORT

Deep-SORT [11] is an improvement of the SORT algorithm, integrating appearance information of objects to enhance associations. Data association integrates an additional appearance metric based on pre-trained CNNs allowing re-identification of tracks, after a long period of occlusion. The KF Estimation and the Track management modules are similar to the corresponding SORT modules. An overview of the method is presented in Figure 3.

As in SORT, the association of detected bounding boxes to tracks is solved by the Hungarian algorithm, using a two-part matching cascade. In the first part, the Deep-SORT method uses motion and appearance metrics to associate valid tracks. The second part uses the same data association strategy as in SORT to associate unmatched and tentative tracks (recently created) with unmatched detections. Motion information is incorporated by the (squared) Mahalanobis distance between predicted states and detections. In addition to the metric computed with the Mahalanobis distance, a second metric based on the smallest cosine distance measures the distance between each track and each measurement appearance features. The appearance features are computed by a pre-trained CNN model. The CNN in the Deep-SORT method was trained on a large-scale person re-identification dataset [41] using deep cosine metric learning [42]. A pre-trained model is provided by the authors in their repository (https://github.com/nwojke/deep_SORT (accessed on 15 October 2021)).

2. Bounding box ratio based cost matrix ($R(D, P)$)—implemented as a ratio between the product of each width and height:

$$R(D, P) = \begin{bmatrix} r(D_1, P_1) & \dots & r(D_1, P_M) \\ r(D_2, P_1) & \dots & r(D_2, P_M) \\ \vdots & \ddots & \vdots \\ r(D_N, P_1) & \dots & r(D_N, P_M) \end{bmatrix} \quad (5)$$

$$r(D_i, P_i) = \min\left(\frac{w_{D_i}h_{D_i}}{w_{P_i}h_{P_i}}, \frac{w_{P_i}h_{P_i}}{w_{D_i}h_{D_i}}\right) \quad (6)$$

In addition, for boxes with similar shapes, this metric outcome with a value closer to 1 contrasts values close to 0 or much greater than 1 otherwise. For that reason, the minimum between the bounding box ratio and its inverse is applied, to get a value that is within the $[0, 1]$ range.

3. SORT’s IoU cost matrix combined with the Euclidean distance cost matrix:

$$E_D^{IoU}(D, P) = IoU(D, P) \circ D_E(D, P) \quad (7)$$

where \circ represents the Hadamard product (element-wise product) between two matrices.

4. SORT’s IoU cost matrix combined with the box ratio based cost matrix:

$$R^{IoU}(D, P) = IoU(D, P) \circ R(D, P) \quad (8)$$

5. Euclidean distance cost matrix combined with the box ratio based cost matrix:

$$R^{D_E}(D, P) = D_E(D, P) \circ R(D, P) \quad (9)$$

6. SORT’s IoU cost matrix combined with the Euclidean distance cost matrix and the box ratio based cost matrix:

$$M(D, P) = IoU(D, P) \circ D_E(D, P) \circ R(D, P) \quad (10)$$

7. Element-wise average of every cost matrix ($A(D, P)$):

$$A(D_i, P_i) = \frac{IoU(D_i, P_i) + D_E(D_i, P_i) + R(D_i, P_i)}{3}, \quad i \in D, j \in P \quad (11)$$

8. Element-wise weighted mean of every cost matrix value:

$$W_M(D_i, P_i) = \lambda_{IoU} \cdot IoU(D_i, P_i) + \lambda_{D_E} \cdot D_E(D_i, P_i) + \lambda_R \cdot R(D_i, P_i), \\ i \in D, j \in P, \lambda_{IoU} + \lambda_{D_E} + \lambda_R = 1 \quad (12)$$

To improve tracking performance in multi-class environments, cost matrices can be updated based on the match between predicted and detected object class (class gate):

$$C^*(C_{i,j}, D_i, P_i) = \begin{cases} C_{i,j} & \text{if Class } D_i = P_i, i \in D, j \in P \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

3.4. ISR Tracking Dataset

The ISR RGB-D Dataset [28] is a non-object centric RGB-D dataset, recorded at the Institute of Systems and Robotics (ISR-UC) facilities using a camera sensor onboard the ISR-InterBot [43] mobile platform. The dataset presents a mission performed by the platform in a real scenario setting, representing object conditions under which mobile robot platforms may navigate. The ISR RGB-D dataset contains a total of 10,000 RGB-D raw images captured

at 30 FPS with a resolution of 640×480 . Moreover, ten object classes (unknown, person, laptop, tvmonitor, chair, toilet, sink, desk, door-open, and door-closed) were annotated at every fourth frame, reaching a total of 7832 object-centric images.

As aforementioned, the main goal of this work is to study and compare the KF-based SORT and Deep-SORT object tracking methods to be applied in real-time mobile robot applications. To pursue that goal, a dataset representing the object conditions from the mobile robot platform's point of view during their navigation tasks is required. Due to the lack of publicly available datasets for such requirements, the labels of ISR RGB-D Dataset (https://github.com/rmca16/ISR_RGB-D_Dataset (accessed on 15 October 2021)) were rearranged to be used as a multi-object tracking dataset, the ISR Tracking Dataset. First, the labels for the remaining images were annotated for the described ten object classes. Then, a unique tracking ID was associated with the same objects throughout the images, except for the "unknown" object class that was not considered for tracking tasks. However, if an object disappeared or was occluded for more than 15 frames, it was considered as a new object, and a new tracking ID was associated. Each image has an associated ".txt" file that contains all object labels for that image, and each object label is organized as follows: <object class>, <tracking ID>, <bounding box center x>, <bounding box center y>, <bounding box width>, and <bounding box height>. ISR Tracking dataset has in total 32,635 object bounding boxes and 329 object sequences.

4. Experiments

The proposed study was evaluated on the MOT17 [27] dataset and also on the proposed ISR Tracking dataset. Moreover, to evaluate the proposed approaches on the used KF-based algorithms, the following standard evaluation metrics [1] were used: Multi-Object Tracking Accuracy (MOTA), Multi-Object Tracking Precision (MOTP), True Positives (TP), False Positives (FP), False Negatives (FN), Identification Switch (IDs) Mostly Tracked (MT), Mostly Lost (ML), Fragmentation (FM), and Frames Per Second (FPS).

4.1. Datasets

(1) MOT17 Dataset: It is a multi-person tracking benchmark dataset divided into 14 sequences with highly crowded scenarios, different viewpoints, weather conditions, camera motions, and indoor/outdoor environments. The dataset contains a public training/test split, where the training sequences have ground-truth files and detection files provided by three object detection state-of-the-art methods, while the test sequences just have the detection files. Hence, due to the scope of the performed experiments, and also due to the submission's constraints to obtain results on the test sequences, only the training sequences were used in this study. Since the multi-object tracking methods evaluated in this work do not require a training process, the training sequences were used as evaluation.

(2) ISR Tracking Dataset: It is composed of 10,000 RGB-D raw images acquired by an Intel RealSense D435 sensor onboard a mobile robot platform [43], representing the object conditions under which robotic platforms may navigate. Nine object classes were annotated for multi-object tracking tasks, achieving a total of 32,635 object bounding boxes and 329 object sequences. For evaluation, the ISR Tracking dataset was reorganized into two sub-datasets: ISR500 and ISR200. In the ISR500, the dataset was divided into sequences of 500 frames, which gives a total of 20 image sequences. On the other hand, the ISR200 contains 50 image sequences, which are the result of partitioning the dataset into sequences of 200 images. On both sub-datasets, the train/test image sequence split was performed by interleaving the sequences, i.e., the first sequence was used to train, the second sequence was used to test, the third sequence was used to train, and so on.

4.2. Implementation Details

All modules were implemented using the Python 3.8.5 programming language. Deep learning networks were also implemented using the PyTorch framework (version 1.8.0). YOLOv3 network was trained using an image size of 416×416 , a fixed learning rate of

10^4 over 50 iterations, a mini-batch of 6 images, and the ADAM optimizer. In addition, the YOLOv3 weights were initialized using the COCO pre-trained model. To perform evaluations on the SORT method [10], the number of frames to hold a track without associations before deleting that track was set with $T_{Lost} = 1$, the minimum number of object detections to start a new track was set with $hit_{min} = 3$, and the minimum threshold value for bounding box association was set with $th_{cost} = 0.3$. For the Deep-SORT, the following constant values were used: $\lambda = 0$ (hyperparameter to control the influence of each metric on the association cost), $T_{Lost} = 30$, and an association gating threshold, $dist_{max}^1 = 0.2$. Moreover, all experiments were performed using an Nvidia RTX 2060 super GPU, 32 GB RAM, and an AMD Ryzen 5 3600 CPU.

4.3. Results

The evaluation of the proposed work was divided as follows: evaluation of the SORT and Deep-SORT on both MOT17 and ISR Tracking datasets using all the available frames (ideal conditions); evaluation of the SORT and Deep-SORT on the ISR Tracking dataset skipping frames, representing real conditions when it is not possible to perform the default 30 FPS; and evaluation of the whole pipeline, YOLOv3 + object tracking method, evaluating also the influence that the YOLO's detection performance has on the object tracking method.

4.3.1. SORT and Deep-SORT Evaluation

The proposed W_M data association cost matrix formulation requires the selection of three constant values (weights) to control the influence of each data association cost matrix. Table 2 shows the evaluation performed on the MOT17 dataset with different weight value combinations of the W_M cost matrix. Based on the achieved results, the highest MOTA value was attained using: $\lambda_{IoU} = \frac{7}{10}$, $\lambda_{DE} = \frac{2}{10}$ and $\lambda_R = \frac{1}{10}$. The aforementioned weight configuration has the minimum number of FP and IDs, despite the higher number of FNs. Furthermore, it has the minimum number of FM by a large amount. Therefore, throughout the following evaluations, the aforementioned values were used for the W_M cost matrix.

Table 2. Evaluation of the W_M data association cost matrix using different weight combinations on the MOT17 dataset.

Weights			Evaluation Metrics								
λ_{IoU}	λ_{DE}	λ_R	% MOTA \uparrow	% MOTP \uparrow	TP \uparrow	FP \downarrow	FN \downarrow	IDs \downarrow	% MT \uparrow	% ML \downarrow	FM \downarrow
5/10	4/10	1/10	44.93	87.84	56,677	6223	54,772	848	12.3	33.0	946
5/10	3/10	2/10	44.99	87.84	56,713	6196	54,757	827	12.3	33.3	932
4/10	3/10	3/10	44.85	87.76	56,688	6323	54,776	833	13.0	32.8	953
3/10	4/10	3/10	44.64	87.71	56,613	6479	54,832	852	12.8	33.0	948
3/10	3/10	4/10	44.58	87.70	56,558	6492	54,858	881	12.8	33.0	967
4/10	5/10	1/10	44.75	87.75	56,627	6379	54,793	877	12.6	33.3	961
6/10	3/10	1/10	45.25	87.90	56,803	5984	54,695	799	12.1	33.0	912
6/10	2/10	2/10	45.25	87.92	56,801	5990	54,705	791	12.1	33.0	907
7/10	2/10	1/10	45.53	88.09	56,552	5426	54,996	749	12.6	33.5	853

The bold value highlights the best value on each column (in this case, each MOT evaluation metric).

Table 3 shows the results achieved on the MOT17 dataset. Regarding the SORT's results, the highest MOTA result was obtained using the default IoU cost matrix, being a similar result achieved by the E_D^{IoU} , R^{IoU} , M , and W_M cost matrices. However, on the remaining evaluation metrics, the default IoU cost matrix was outperformed by the proposed cost matrices. The M cost matrix had the lowest number of FP, IDs, and FM, which represents the most accurate tracking for sequences generated by the SORT. The A cost matrix had the highest number of TP and the lowest number of FN, which is proportional to the percentage of MT sequences. For this work, which has in view mobile robot navigation tasks, those metrics could impact performance, as it can ensure that the object is successfully tracked until the object leaves the scene. Regarding the Deep-SORT

results, the best MOTA result was achieved by the proposed W_M cost matrix with 45.67%. The W_M cost matrix reached the best results for the TP and MT evaluation metrics. The default IoU cost matrix achieved the best MOTP, FP, IDs, and FM results, which are very similar to the results attained by the proposed W_M cost matrix. Overall, promising results were achieved by the proposed cost matrices, being able to outperform the default IoU cost matrix. Moreover, the Deep-SORT with W_M cost matrix was able to obtain the highest MOTA and MT. Attained results show similar overall performances between SORT and Deep-SORT. However, as expected, SORT is much faster than Deep-SORT.

Table 3. Evaluation of the SORT, Deep-SORT, and proposed data association cost matrices on the MOT17 dataset.

Cost Matrix	Evaluation Metrics									
	% MOTA↑	% MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
SORT										
<i>IoU</i>	45.56	88.19	56,298	5136	55,281	718	11.5	35.3	798	516
D_E	41.24	86.99	54,292	7977	56,271	1734	7.9	33.7	1915	500
<i>R</i>	14.15	83.06	37,236	21,351	70,281	4780	4.9	37.5	4730	510
E_D^{IoU}	45.55	88.20	56,275	5126	55,305	717	11.5	35.3	799	486
R^{IoU}	45.55	88.21	56,263	5111	55,324	710	11.5	35.5	797	499
R^{D_E}	44.40	87.79	56,329	6470	55,028	940	11.7	32.4	1090	480
<i>M</i>	45.54	88.21	56,245	5107	55,344	708	11.5	35.7	797	469
<i>A</i>	44.72	87.72	56,636	6417	54,811	850	13.0	33.0	958	472
W_M	45.53	88.09	56,552	5426	54,996	749	12.6	33.5	853	473
Deep-SORT										
<i>IoU</i>	45.53	88.26	55,641	4510	56,187	469	13.0	35.7	666	57
D_E	45.49	88.13	55,768	4689	55,988	541	13.6	34.6	736	57
<i>R</i>	42.20	87.76	53,722	6334	57,604	971	9.7	37.2	1126	57
E_D^{IoU}	45.49	88.12	55,781	4702	55,995	521	13.9	34.2	724	57
R^{IoU}	44.35	88.06	55,106	5300	56,532	659	12.6	35.3	845	57
R^{D_E}	44.83	87.99	55,383	5038	56,278	636	12.3	34.4	821	57
<i>M</i>	44.87	87.97	55,408	5019	56,263	626	12.6	34.4	814	57
<i>A</i>	45.49	88.15	55,788	4701	56,001	508	13.9	34.6	712	57
W_M	45.67	88.23	55,834	4547	55,991	472	13.9	34.8	667	57

The bold value highlights the best value on each column (in this case, each MOT evaluation metric).

An evaluation of SORT and Deep-SORT, where the data association threshold is modified, was also performed on the MOT17 dataset, whose results are presented in Figure 4. As expected, as the threshold value increased, the MOTA score decreased for the majority of the cost matrices. As observed, no threshold value was found to be suitable for all evaluated cost matrices. Hence, the best results were obtained using a threshold value of 0.3, which was thereafter used for all the evaluations.

Table 4 presents the results attained on the ISR Tracking dataset. Due to the multi-class available on the ISR Tracking dataset, an evaluation on the SORT algorithm using and not using the class gate metric, to discard associations of objects with different object classes, was performed. Regarding the SORT's results, similar to the reported results on the MOT17 dataset, the proposed data association cost matrices outperformed the default IoU cost matrix. Moreover, the results of all evaluated data association cost matrices were slightly improved by using the class gate formulation, being able to reach the highest MOTA result with 91.02%. The *A* cost matrix using the class gate formulation was able to achieve the best result on the TP, FN, IDs, and MT with 29,785, 2799, 51, and 69.3%, respectively. The *A* data association cost matrix presents a significant improvement on the MT evaluation metric compared with the IoU cost matrix (61.7% to 69.3%), which can impact the performance of a mobile robot platform during navigation tasks. Regarding the Deep-SORT's results, once again, the proposed data association cost matrices outperformed the default *IoU* cost matrix. Moreover, the *A* cost matrix achieved the highest MOTA and MT values, while

the E_D^{IoU} achieved the best TP, FN, IDs, and ML results. A significant improvement of the MT evaluation metric was attained with the A cost matrix. Overall, on both SORT and Deep-SORT algorithms, the proposed data association cost matrices outperformed the default IoU cost matrix. The A cost matrix achieved the highest values on MOTA and MT evaluation metrics, showing that it could be the most suitable data association cost matrix to use. Regarding those evaluation metrics, the Deep-SORT outperformed the SORT algorithm, with a highlight on the MT evaluation metric (69.3% to 78.7%). Moreover, promising results were reached by both methods on the ISR Tracking dataset.

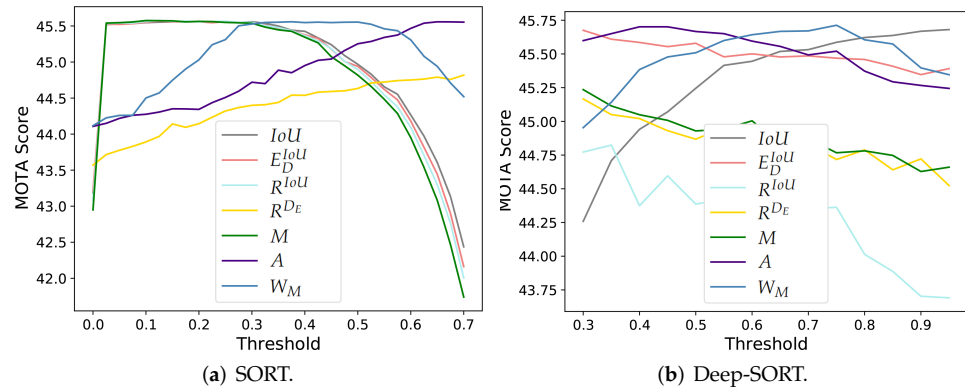


Figure 4. Tracking MOTA variation according to different data association thresholds on the MOT17 dataset.

For the following evaluations, based on the reported results, only the following data association cost matrices using the class gate metric were used: IoU , A , and W_M on the SORT algorithm and the IoU , E_D^{IoU} , and A on the Deep-SORT algorithm.

Table 4. Evaluation of the SORT, Deep-SORT, and proposed data association cost matrices on the ISR Tracking dataset.

Cost Matrix	Evaluation Metrics									
	% MOTA↑	% MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
SORT										
IoU	90.57	92.21	29,589	31	2932	114	60.5	1.5	556	1368
D_E	88.85	91.48	29,307	310	3031	297	59.0	0.9	618	1389
R	68.23	88.45	25,015	2748	5550	2070	30.7	3.6	1179	1380
E_D^{IoU}	90.44	92.27	29,538	22	2974	123	59.0	1.5	561	1317
R^{IoU}	90.34	92.30	29,491	9	3008	136	58.7	1.8	566	1377
R^{D_E}	90.77	92.00	29,713	91	2827	95	65.0	0.9	562	1368
M	90.21	92.35	29,445	5	3053	137	57.4	1.8	564	1311
A	90.87	91.96	29,756	101	2799	80	67.5	1.2	558	1288
W_M	90.90	92.10	29,715	50	2832	88	64.4	1.2	558	1298
SORT with Class Gate Metric										
IoU	90.63	92.20	29,611	33	2926	98	61.7	1.5	550	1404
D_E	90.82	92.00	29,739	100	2830	66	66.9	1.2	566	1408
R	87.74	91.56	29,134	500	3216	285	63.2	1.2	642	1425
E_D^{IoU}	90.49	92.27	29,553	22	2969	113	59.6	1.5	556	1337
R^{IoU}	90.36	92.32	29,497	8	3015	123	59.3	1.8	559	1392
R^{D_E}	90.98	92.05	29,767	77	2813	55	68.1	0.9	555	1375
M	90.24	92.36	29,456	5	3050	129	58.1	1.8	559	1307
A	91.02	92.02	29,785	81	2799	51	69.3	1.2	554	1292
W_M	90.93	92.10	29,727	53	2837	71	65.3	1.2	552	1305

Table 4. Cont.

Cost Matrix	Evaluation Metrics									
	% MOTA↑	% MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
Deep-SORT with Class Gate Metric										
<i>IoU</i>	90.80	89.66	30,989	1357	1447	199	72.3	0.3	142	163
<i>D_E</i>	91.09	89.53	31,100	1372	1367	168	76.3	0.3	131	167
<i>R</i>	89.12	90.27	30,467	1384	1783	385	62.3	0.3	292	166
<i>E_D^{IoU}</i>	91.15	89.52	31,124	1376	1350	161	78.4	0.3	130	165
<i>R^{IoU}</i>	90.90	89.86	30,994	1328	1401	240	75.7	0.3	160	166
<i>R^{D_E}</i>	91.07	89.54	31,087	1367	1381	167	76.3	0.6	134	169
<i>M</i>	91.15	89.55	31,116	1370	1354	165	77.8	0.6	125	163
<i>A</i>	91.23	89.55	31,123	1350	1350	162	78.7	0.3	126	168
<i>W_M</i>	91.09	89.56	31,103	1376	1363	169	76.6	0.3	119	166

The bold value highlights the best value on each column (in this case, each MOT evaluation metric).

4.3.2. SORT and Deep-SORT on Skipped Frames

In real scenarios, sometimes due to hardware constraints, it is not always possible (or needed) to run the algorithms at 30 FPS, which is a standard value on image acquisition from cameras. Hence, to evaluate the tracking performance on such conditions, experiments by skipping 1, 2, and 3 images, representing an image acquisition at 15, 10, and 7.5 FPS, respectively, were performed.

Table 5 shows the SORT and Deep-SORT results attained on the ISR Tracking dataset using non-consecutive frames. As expected, as the image gap increased, the object tracking performance decreased. This happens due to a greater displacement of the objects, which increases the difficulty in predicting and associating objects. Nevertheless, promising results were achieved by the proposed *A* data association metric on both tracking methods, outperforming the default *IoU* data association metric, especially on MOTA, IDs, and MT evaluation metrics. The best overall performance was reached by the SORT method with the proposed *A* data association cost matrix with an accuracy of 86.43% and 58.2% of mostly tracked object sequences. The SORT method using the *IoU* data association metric attained the best results on the MOTP and FP evaluation metrics, while the Deep-SORT method with the proposed *A* data association metric achieved the best results on the TP and FN evaluation metrics. Note that, in these conditions, a significant improvement was achieved by the *A* data association cost matrix compared to the *IoU* association metric, showing its capacity to hold the object track.

4.3.3. Detection-Based MOT Pipeline

To evaluate the performance of the SORT and Deep-SORT object tracking methods in real scenarios, an evaluation using the YOLOv3 object detector algorithm feeding the tracking methods was performed. Moreover, to also evaluate the influence that the object detector performance may have over the object tracking performance, four YOLOv3 models with different performances were used. The four YOLOv3 models were trained on the same data (ISR RGB-D Dataset), and on the same conditions, varying only the number of training epochs. Each used YOLOv3 model has the following mean average precision: $Y_{M1, \dots, M4} = \{38\%, 60\%, 80\%, 90\%\}$.

Table 6 presents the detection-based MOT pipeline results achieved on the ISR200 sub-dataset. As expected, the YOLOv3's performance had a significant role in the overall pipeline. As the YOLOv3 performance increased, the object tracking performance also increased. In the case of a poor YOLOv3's performance, the number of FN was so high, especially on the Deep-SORT method, which achieved a negative accuracy (MOTA). Regardless of the YOLOv3's performance, in these conditions, SORT outperformed the Deep-SORT method. Moreover, the three data association cost matrices used on the SORT method reached similar results, being the default *IoU* cost matrix able to achieve the best MOTA and FP results, while the *A* data association metric got the best MT values. Note that using

an object detector may introduce additional errors to the object tracking pipeline, such as incorrect detections, shifted detections, miss detections, and wrong object classification. This can be observed by the obtained values of TP, FP, FN, and IDs, which directly influence the remaining evaluation metrics. As shown in Table 6, the object tracking performance increases as the YOLOv3's performance also increases, due to a large decrease in the FP values as well as the IDs values, which occurs due to an improvement of the object detection performance. Regarding the frame rate results, as expected, the SORT was faster than Deep-SORT since SORT does not have to extract visual features through a CNN.

Table 5. Evaluation of the SORT and the Deep-SORT on the ISR Tracking dataset using non-consecutive images. All data association cost matrices used the class gate formulation.

Gap	Tracking Method	Evaluation Metrics									
		% MOTA↑	% MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
1	SORT (<i>IoU</i>)	84.92	88.89	13,514	47	2247	97	48.3	9.2	178	1444
	SORT (<i>A</i>)	86.43	88.20	13,819	113	2012	27	58.2	8.3	160	1375
	SORT (<i>W_M</i>)	86.11	88.43	13,732	77	2073	53	53.2	8.9	154	1426
	Deep-SORT (<i>IoU</i>)	81.28	86.02	13,893	1003	1761	204	48.9	5.8	128	155
	Deep-SORT (<i>E_D^{IoU}</i>)	82.02	85.56	14,077	1070	1590	191	55.7	4.0	111	158
	Deep-SORT (<i>A</i>)	82.49	85.61	14,109	1027	1587	162	57.2	4.6	103	158
2	SORT (<i>IoU</i>)	79.73	87.75	8718	53	2022	128	35.8	12.3	151	1407
	SORT (<i>A</i>)	83.37	86.09	9239	178	1593	36	49.4	8.6	130	1413
	SORT (<i>W_M</i>)	83.08	86.61	9117	88	1674	77	43.5	9.0	127	1480
	Deep-SORT (<i>IoU</i>)	75.28	84.16	8975	794	1697	196	38.0	8.3	126	152
	Deep-SORT (<i>E_D^{IoU}</i>)	78.65	82.86	9414	866	1317	137	51.2	4.0	89	153
	Deep-SORT (<i>A</i>)	79.41	82.98	9470	840	1279	119	51.9	4.9	83	153
3	SORT (<i>IoU</i>)	75.50	87.23	6406	38	1897	131	23.4	13.1	84	1272
	SORT (<i>A</i>)	81.02	84.52	7094	261	1292	48	39.9	6.2	43	1338
	SORT (<i>W_M</i>)	81.28	85.44	6941	86	1414	79	33.0	8.4	39	1355
	Deep-SORT (<i>IoU</i>)	69.24	83.37	6528	688	1722	184	31.5	10.0	107	138
	Deep-SORT (<i>E_D^{IoU}</i>)	73.88	82.33	6947	716	1295	192	34.0	4.0	104	145
	Deep-SORT (<i>A</i>)	75.71	80.90	7185	800	1192	57	52.0	3.4	53	148

The bold value highlights the best value on each column (in this case, each MOT evaluation metric).

Table 6. Evaluation of the detection-based MOT pipeline on the ISR200 sub-dataset. All data association cost matrices used the class gate formulation.

YOLO	Tracking Method	Evaluation Metrics									
		% MOTA↑	% MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
<i>Y_{M1}</i>	SORT (<i>IoU</i>)	23.30	78.65	13,345	9665	1373	1078	42.6	6.6	167	47
	SORT (<i>A</i>)	22.39	78.61	13,384	9847	1350	1062	44.6	6.6	168	48
	SORT (<i>W_M</i>)	22.66	78.61	13,358	9778	1348	1090	44.6	6.6	166	49
	Deep-SORT (<i>IoU</i>)	−2.27	78.12	13,029	13,387	1338	1429	39.1	7.0	157	27
	Deep-SORT (<i>E_D^{IoU}</i>)	−10.16	78.08	13,090	14,695	1236	1470	40.3	7.4	109	28
	Deep-SORT (<i>A</i>)	−10.66	78.20	13,100	14,784	1218	1478	43.0	7.0	114	28
<i>Y_{M2}</i>	SORT (<i>IoU</i>)	41.93	81.03	14,191	7567	926	679	58.9	3.9	123	49
	SORT (<i>A</i>)	41.59	80.99	14,222	7652	895	679	58.5	3.9	114	49
	SORT (<i>W_M</i>)	41.54	81.00	14,203	7642	906	687	58.9	3.9	120	49
	Deep-SORT (<i>IoU</i>)	21.50	81.14	13,942	10,546	991	863	55.8	4.3	124	28
	Deep-SORT (<i>E_D^{IoU}</i>)	14.90	81.04	14,043	11,689	899	854	55.8	4.7	81	28
	Deep-SORT (<i>A</i>)	15.65	81.20	14,032	11,560	917	847	56.6	4.3	87	28
<i>Y_{M3}</i>	SORT (<i>IoU</i>)	65.43	81.64	14,623	4288	858	315	66.7	5.4	64	50
	SORT (<i>A</i>)	65.21	81.64	14,633	4333	833	330	66.7	5.4	71	50
	SORT (<i>W_M</i>)	65.40	81.64	14,644	4313	835	317	66.3	5.4	63	50
	Deep-SORT (<i>IoU</i>)	53.75	80.33	14,406	5916	975	415	61.2	5.8	101	30
	Deep-SORT (<i>E_D^{IoU}</i>)	49.28	80.32	14,430	6646	897	469	60.5	6.2	83	30
	Deep-SORT (<i>A</i>)	50.01	80.44	14,450	6551	890	456	62.4	6.2	75	30

Table 6. Cont.

YOLO	Tracking Method	Evaluation Metrics									
		% MOTA↑	% MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
Y _{M4}	SORT (<i>IoU</i>)	73.86	83.41	14,446	2779	1141	209	64.7	5.0	84	51
	SORT (<i>A</i>)	73.75	83.39	14,455	2806	1132	209	65.9	4.7	86	51
	SORT (<i>W_M</i>)	73.83	83.42	14,456	2794	1132	208	65.1	5.0	80	51
	Deep-SORT (<i>IoU</i>)	65.86	81.64	14,287	3884	1205	304	60.5	5.8	104	31
	Deep-SORT (<i>E_D^{IoU}</i>)	64.32	81.62	14,392	4232	1133	271	64.3	5.8	90	31
	Deep-SORT (<i>A</i>)	64.69	81.61	14,404	4186	1133	259	65.5	5.4	90	31

The bold value highlights the best value on each column (in this case, each MOT evaluation metric).

Table 7 presents the detection-based MOT pipeline results achieved on the ISR500 sub-dataset. Once again, the performance of the YOLOv3 is crucial for a promising object tracking performance. Overall, similar to the results attained on the ISR200 sub-dataset, the SORT method obtained the best results. However, the Deep-SORT method reached the best values on the FN, MT, and ML evaluation metrics, showing that the Deep-SORT could be most suitable for tracking larger object sequences. This happens due to an increased capability of the Deep-SORT method to re-identifying lost object sequences compared with the SORT method, which struggles to predict the position of the object when the track starts to miss. As observed in the previous evaluations, the *A* cost matrix, in both SORT and Deep-SORT, achieved the best MT result, meaning that the object sequence is, at least, tracked in 80% of its life span, which is very important to successfully perform mobile robot navigation tasks.

Table 7. Evaluation of the detection-based MOT pipeline on the ISR500 sub-dataset. All data association cost matrices used the class gate formulation.

YOLO	Tracking Method	Evaluation Metrics									
		% MOTA↑	% MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
Y _{M1}	SORT (<i>IoU</i>)	24.86	78.88	12,485	8798	1288	1056	38.3	2.3	156	50
	SORT (<i>A</i>)	23.68	78.82	12,497	8986	1255	1077	40.0	2.3	163	49
	SORT (<i>W_M</i>)	24.30	78.82	12,495	8891	1253	1081	39.4	2.3	153	49
	Deep-SORT (<i>IoU</i>)	−0.45	78.33	12,266	12,332	1178	1385	33.1	2.9	173	28
	Deep-SORT (<i>E_D^{IoU}</i>)	−8.75	78.31	12,334	13,631	1093	1402	38.9	2.3	125	28
	Deep-SORT (<i>A</i>)	−8.15	78.38	12,364	13,573	1060	1405	41.1	2.3	122	28
Y _{M2}	SORT (<i>IoU</i>)	43.49	81.20	13,242	6793	1001	586	53.1	2.9	108	50
	SORT (<i>A</i>)	43.16	81.19	13,250	6850	986	593	54.3	2.9	102	50
	SORT (<i>W_M</i>)	43.30	81.19	13,252	6831	980	597	54.9	2.9	104	50
	Deep-SORT (<i>IoU</i>)	23.61	81.36	13,041	9540	976	812	49.7	2.9	124	28
	Deep-SORT (<i>E_D^{IoU}</i>)	16.07	81.32	13,147	10,764	870	812	56.6	2.9	93	29
	Deep-SORT (<i>A</i>)	15.49	81.40	13,099	10,802	896	834	56.0	2.9	95	29
Y _{M3}	SORT (<i>IoU</i>)	65.25	81.34	13,637	3961	912	280	64.0	2.3	72	51
	SORT (<i>A</i>)	65.03	81.32	13,653	4009	887	289	63.4	2.3	76	51
	SORT (<i>W_M</i>)	65.20	81.33	13,652	3983	896	281	64.0	2.3	71	51
	Deep-SORT (<i>IoU</i>)	53.11	80.02	13,525	5649	932	372	58.9	1.1	124	30
	Deep-SORT (<i>E_D^{IoU}</i>)	49.77	80.01	13,628	6248	815	386	64.0	1.1	99	30
	Deep-SORT (<i>A</i>)	49.91	80.06	13,605	6204	825	399	66.3	1.1	109	30
Y _{M4}	SORT (<i>IoU</i>)	75.28	83.54	13,570	2406	1068	191	56.6	3.4	70	52
	SORT (<i>A</i>)	75.30	83.54	13,584	2418	1058	187	57.1	2.9	69	51
	SORT (<i>W_M</i>)	75.34	83.54	13,585	2413	1057	187	57.1	3.4	68	51
	Deep-SORT (<i>IoU</i>)	68.70	81.62	13,543	3355	1024	262	58.3	1.7	107	31
	Deep-SORT (<i>E_D^{IoU}</i>)	66.95	81.56	13,629	3701	950	250	66.9	2.3	93	31
	Deep-SORT (<i>A</i>)	66.82	81.58	13,629	3721	948	252	67.4	1.1	95	31

The bold value highlights the best value on each column (in this case, each MOT evaluation metric).

5. Conclusions

In this paper, having in view navigation tasks in assistive mobile robot platforms, an evaluation study of two MOTs by detection algorithms, SORT and Deep-SORT, was presented. Moreover, eight new tracking data association metrics based on intersection over union, Euclidean distances, and bounding boxes ratio were proposed. To evaluate both tracking methods with the proposed data association metrics, the ISR Tracking dataset, which represents the object conditions from an assistive mobile robot's point of view, was also proposed. The presented pipeline consists of using the YOLOv3 network to detect and classify the objects available on RGB images, feeding the tracking algorithm. Promising results were attained by the majority of the proposed tracking data association metrics on the SORT, and also on the Deep-SORT. Overall, based on the performed experiments, the SORT method was able to achieve higher results of accuracy and precision, while the Deep-SORT method obtained the best values of FN, IDs, and MT. Moreover, the proposed *A* data association metric achieved the best performance on both evaluated object tracking methods. The *A* data association metric showed a significant improvement on the MT evaluation metric, which could be crucial to successful navigation tasks on robotic platforms. The results showed, as expected, that the object tracking overall performance has a high dependency on the object detector performance. The SORT is faster than the Deep-SORT, reaching 50 FPS on the overall pipeline (YOLOv3 + SORT). Therefore, considering navigation tasks in assistive platforms, and also considering issues associated with an object detector algorithm, the SORT method using the *A* data association metric obtained more robust results and, as such, can be a more suitable approach.

As future work, it is intended to integrate the presented pipeline on the RobChair [21] platform for assistive navigation tasks.

Author Contributions: Conceptualization, methodology, software, validation, investigation: R.P. and G.C.; formal analysis, R.P., G.C. and L.G.; writing, R.P., G.C., L.G. and U.J.N.; supervision, funding acquisition: U.J.N. All authors have read and agreed to the published version of the manuscript.

Funding: Ricardo Pereira has been supported by the Portuguese Foundation for Science and Technology (FCT) under a PhD grant with reference SFRH/BD/148779/2019. This work has been also supported by the projects B-RELIABLE with reference SAICT/30935/2017 and MATIS-CENTRO-01-0145-FEDER-000014. It was also partially supported by ISR-UC through FCT grant UIDB/00048/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository. Publicly available datasets were analyzed in this study. Data can be found here: <https://motchallenge.net/data/MOT17/> (accessed on 17 October 2021) and https://github.com/rmca16/ISR_RGB-D_Dataset (accessed on 20 December 2021).

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

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. Xu, Y.; Osep, A.; Ban, Y.; Horaud, R.; Leal-Taixe, L.; Alameda-Pineda, X. How To Train Your Deep Multi-Object Tracker. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
3. Kamal, R.; Chemmanur, A.J.; Jose, B.; Mathews, S.; Varghese, E. Construction Safety Surveillance Using Machine Learning. In Proceedings of the International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, Canada, 20–22 October 2020.
4. Behrendt, K.; Novak, L.; Botros, R. A Deep Learning Approach to Traffic Lights: Detection, Tracking, and Classification. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017.
5. Ess, A.; Schindler, K.; Leibe, B.; Gool, L.V. Object Detection and Tracking for Autonomous Navigation in Dynamic Environments. *Int. J. Robot. Res.* **2010**, *29*, 1707–1725. [CrossRef]
6. Lo, S.; Yamane, K.; Sugiyama, K. Perception of Pedestrian Avoidance Strategies of a Self-Balancing Mobile Robot. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019.

7. Islam, M.; Hong, J.; Sattar, J. Person-following by autonomous robots: A categorical overview. *Int. J. Robot. Res.* **2019**, *38*, 1581–1618. [[CrossRef](#)]
8. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H. Fast Online Object Tracking and Segmentation: A Unifying Approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
9. Liu, Q.; Liu, B.; Wu, Y.; Li, W.; Yu, N. Real-Time Online Multi-Object Tracking in Compressed Domain. *IEEE Access* **2019**, *7*, 76489–76499. [[CrossRef](#)]
10. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016.
11. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017.
12. Chen, L.; Ai, H.; Zhuang, Z.; Shang, C. Real-Time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018.
13. Pereira, R.; Gonçalves, N.; Garrote, L.; Barros, T.; Lopes, A.; Nunes, U.J. Deep-Learning based Global and Semantic Feature Fusion for Indoor Scene Classification. In Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal, 15–17 April 2020.
14. Pereira, R.; Garrote, L.; Barros, T.; Lopes, A.; Nunes, U.J. A Deep Learning-based Indoor Scene Classification Approach Enhanced with Inter-Object Distance Semantic Features. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021.
15. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
16. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
17. Zhang, Y.; Wang, C.; Wang, X.; Zenf, W.; Liu, W. A Simple Baseline for Multi-Object Tracking. *arXiv* **2020**, arXiv:2004.01888.
18. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
19. Wu, J.; Cao, J.; Song, L.; Wang, Y.; Yang, M.; Yuan, J. Track to Detect and Segment: An Online Multi-Object Tracker. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
20. Bergmann, P.; Meinhardt, T.; Leal-Taixé, L. Tracking without bells and whistles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
21. Lopes, A.; Rodrigues, J.; Perdigao, J.; Pires, G.; Nunes, U.J. A New Hybrid Motion Planner: Applied in a Brain-Actuated Robotic Wheelchair. *IEEE Robot. Autom. Mag.* **2016**, *23*, 82–93. [[CrossRef](#)]
22. Iturrate, I.; Antelis, J.M.; Kubler, A.; Minguez, J. A Noninvasive Brain-Actuated Wheelchair Based on a P300 Neurophysiological Protocol and Automated Navigation. *IEEE Trans. Robot.* **2009**, *25*, 614–627. [[CrossRef](#)]
23. Cruz, A.; Pires, G.; Lopes, A.; Carona, C.; Nunes, U.J. A Self-Paced BCI With a Collaborative Controller for Highly Reliable Wheelchair Driving: Experimental Tests with Physically Disabled Individuals. *IEEE Trans. Hum. Mach. Syst.* **2021**, *51*, 109–119. [[CrossRef](#)]
24. Lopes, A.; Pires, G.; Nunes, U.J. Assisted navigation for a brain-actuated intelligent wheelchair. *Robot. Auton. Syst.* **2013**, *61*, 245–258. [[CrossRef](#)]
25. Sinyukov, D.A.; Padir, T. A Novel Shared Position Control Method for Robot Navigation Via Low Throughput Human-Machine Interfaces. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018.
26. Carvalho, G. Kalman Filter-Based Object Tracking Techniques for Indoor Robotic Applications. Master’s Dissertation, University of Coimbra, Coimbra, Portugal, 2021.
27. Milan, A.; Leal-Taixé, L.; Reid, I.D.; Roth, S.; Schindler, K. MOT16: A benchmark for multi-object tracking. *arXiv* **2016**, arXiv:1603.00831.
28. Pereira, R.; Garrote, L.; Barros, T.; Lopes, A.; Nunes, U.J. An Experimental Study of the Accuracy vs Inference Speed of RGB-D Object Recognition. In Proceedings of the IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), Naples, Italy, 31 August–4 September 2020.
29. Fiaz, M.; Mahmood, A.; Javed, S.; Jung, S.K. Handcrafted and Deep Trackers: Recent Visual Object Tracking Approaches and Trends. *ACM Comput. Surv.* **2019**, *52*, 1–44. [[CrossRef](#)]
30. Zhang, X.; Wang, X.; Gu, C. Online multi-object tracking with pedestrian re-identification and occlusion processing. *Vis. Comput.* **2021**, *37*, 1089–1099. [[CrossRef](#)]
31. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning Dynamic Siamese Network for Visual Object Tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
32. Reid, D. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control.* **1979**, *24*, 843–854. [[CrossRef](#)]
33. He, J.; Huang, Z.; Wang, N.; Zhang, Z. Learnable Graph Matching: Incorporating Graph Partitioning with Deep Feature Learning for Multiple Object Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.

34. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [[CrossRef](#)]
35. Lee, S.; Kim, E. Multiple Object Tracking via Feature Pyramid Siamese Networks. *IEEE Access* **2019**, *7*, 8181–8194. [[CrossRef](#)]
36. Lin, T.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
37. Jin, J.; Li, X.; Li, X.; Guan, S. Online Multi-object Tracking with Siamese Network and Optical Flow. In Proceedings of the IEEE 5th International Conference on Image, Vision and Computing (ICIVC), Beijing, China, 10–12 July 2020.
38. Lucas, B.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), Vancouver, BC, Canada, 24–28 August 1981.
39. Xiao, H.; Li, Z.; Yang, C.; Yuan, W.; Wang, L. RGB-D Sensor-based Visual Target Detection and Tracking for an Intelligent Wheelchair Robot in Indoors Environments. *Int. J. Control Autom. Syst.* **2015**, *13*, 521–529. [[CrossRef](#)]
40. Lecrosnier, L.; Khemmar, R.; Ragot, N.; Decoux, B.; Rossi, R.; Kefi, N.; Ertaud, J.Y. Deep Learning-Based Object Detection, Localisation and Tracking for Smart Wheelchair Healthcare Mobility. *Int. J. Environ. Res. Public Health* **2021**, *18*, 521–529. [[CrossRef](#)] [[PubMed](#)]
41. Zheng, L.; Bie, Z.; Sun, Y.; Wang, J.; Su, C.; Wang, S.; Tian, Q. MARS: A Video Benchmark for Large-Scale Person Re-identification. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016.
42. Wojke, N.; Bewley, A. Deep Cosine Metric Learning for Person Re-identification. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018.
43. Cruz, R.; Garrote, L.; Lopes, A.; Nunes, U.J. Modular software architecture for human-robot interaction applied to the InterBot mobile robot. In Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Torres Vedras, Portugal, 25–27 April 2018.