UNIVERSIDADE D
COIMBRA

Guilherme de Sousa Carvalho

Kalman Filter-based Object Tracking Techniques for Indoor Robotic Applications

Dissertation supervised by Professor Doctor Urbano José Carreira Nunes
and submitted to the Electrical and Computer Engineering Department
of the Faculty of Science and Technology of the University of Coimbra,
in partial fulfilment of the requirements for the Master's Degree in
Electrical and Computer Engineering, specialization in Automation.

October of 2021

1 2  9 0

UNIVERSIDADE D
COIMBRA

Kalman Filter-based Object Tracking Techniques for Indoor Robotic Applications

Guilherme de Sousa Carvalho

October of 2021

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

Kalman Filter-based Object Tracking Techniques for Indoor Robotic Applications

Dissertation supervised by Professor Doctor Urbano José Carreira Nunes and submitted to the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra, in partial fulfillment of the requirements for the Master's Degree in Electrical and Computer Engineering, specialization in Automation.

Supervisor:
Prof. Dr. Urbano José Carreira Nunes

Co-Supervisor:
Master Ricardo Manuel Teixeira Pereira

Jury:
Prof. Dr. Rui Alexandre de Matos Araújo
Prof. Dr. Ana Cristina Barata Pires Lopes

Coimbra, October of 2021

Acknowledgments

Throughout the development of this dissertation I have received constant support and assistance.

I would like to express my sincere gratitude to my supervisor Dr. Professor Urbano Nunes, for providing me every material needed and feedback throughout the development of this dissertation. Furthermore, I would like to thank Master Ricardo Pereira for delivering me guidance, motivation and knowledge to complete and go through every challenge that was faced with the right attitude, and also Master Luis Garrote for his opinion and criticism.

I would like to acknowledge my team, Daniel Craveiro, Daniel Palaio, Francisco Alves, Gonçalo Lopes, Hugo Figueiras and João Duarte, for being my reference of work and closest friends throughout the development of this dissertation. We are slowly changing to the "Master Team Eletro".

The development of this dissertation wold not be the same without the support of my parents that always provided me comfort and motivation, and also Daniela Pereira that was always there throughout every moment and for providing me the strength to be a better person. Moreover, i would like to mention David Pereira, Diogo Almeida, Francisco Fernandes, João Pedro Almeida and Pedro Santos for their friendship and presence.

This work has been supported by MATIS-CENTRO-01-0145-FEDER-000014, Portugal, and by ISR-UC FCT through grant UIDB/00048/2020.

Abstract

The improvement of social robots have significantly increased, having in view an "intelligent" mobile robot system, that must be able to perform basic tasks, without compromising the human environment. Therefore, perception module has to be robust enough in object detection and tracking. Thus, the proposal of this dissertation, aims to integrate a multi-object tracking method in a mobile robotics context, mainly focusing on efficiency and performance, using the YOLOv3 object detector to acquire objects location in the image.

This dissertation presents a study and exploitation of the SORT and the Deep-SORT Multi-Object Tracking by Detection methods. Aiming to increase robustness of assigning measurements to existing tracks, are introduced different conjugation of similarity metrics, regarding the data association module. Furthermore, to avoid the association between tracks and measurements of different classes, an object class based constraint is applied. These proposed data association techniques, were incorporated in the SORT and the Deep-SORT methods.

The SORT, the Deep-SORT, and proposed data association techniques, were evaluated on the *MOT17* training set and on the *ISR Tracking Dataset* (dataset labeled in this study). Moreover, an experiment for evaluating the performance of each method on a lower frame rate condition was performed, showing a decrease of performance. Nevertheless, experimental results attained without using object detector, shown an improvement of performance, when formulating the association problem with different similarity metrics.

Throughout the development of this study, an indoor multi-class tracking dataset was labeled, providing useful conditions to validate the proposed framework. Therefore, a general evaluation of the SORT, the Deep-SORT and proposed data association techniques, using the YOLOv3 object detector, was performed in the referred labeled multi-class dataset.

Keywords : Multi-Object Tracking, Motion Estimation, Data Association, Autonomous Robotic Platforms

Resumo

A melhoria dos robôs sociais tem aumentado significativamente, tendo em vista um sistema robótico móvel "inteligente", que deve ser capaz de executar tarefas básicas, sem comprometer o ambiente humano. Portanto, o módulo de perceção tem de ser suficientemente robustos na deteção e rastreamento de objetos (rastreamento equivale à tradução portuguesa de *tracking*). Portanto, a proposta desta dissertação, pretende integrar um método de rastreamento de múltiplos objetos, num contexto de robótica móvel, focando-se em questões de eficiência e desempenho computacional, utilizando o detetor de objetos YOLOv3 para adquirir a localização de objetos na imagem.

Esta dissertação apresenta um estudo e exploração dos métodos de rastreamento por deteção, o SORT e o Deep-SORT. Com o objetivo de reforçar a robustez da atribuição de objetos medidos a objetos rastreados, são introduzidas conjugações diferentes de métricas de similaridade, no módulo de associação de dados. Adicionalmente, para evitar a associação de objetos medidos com objetos rastreados de diferentes classes, é aplicada uma restrição baseada em classes de objetos. Estas técnicas propostas de associação de dados, foram incorporadas nos métodos SORT e Deep-SORT.

O SORT, o Deep-SORT, e as técnicas de associação de dados propostas, foram avaliados nos dados de treino do *MOT17* e no *ISR Tracking Dataset* (conjunto de dados etiquetado neste estudo). Foram realizados testes ao desempenho dos métodos em condições de taxa reduzida de imagens, evidenciando uma diminuição do desempenho. Contudo, os resultados experimentais obtidos sem utilizar o detetor de objetos, mostram uma melhoria do desempenho, ao formular o problema de associação com métricas de semelhança diferentes.

Ao longo do desenvolvimento deste estudo, um conjunto de dados de ambientes multi-classe e de interiores, foi etiquetado com dados de rastreio, fornecendo condições úteis para validar a estrutura proposta. Consequentemente foi realizada uma avaliação geral dos métodos SORT, Deep-SORT e técnicas de associação propostas, utilizando o detetor de objetos YOLOv3, no referido conjunto de dados de rastreio multi-classe etiquetado.

Palavras-Chave : Rastreamento de Múltiplos Objectos, Estimação de Movimento, Associação de Dados, Plataformas Róboticas Autónomas

"The only way to do great work is to love what you do. If you haven't found it yet, keep looking. Don't settle."

Steve Jobs

Contents

List of Acronyms

CNN Convolutional Neural Network

CPU Central Processing Unit

CV Computer Vision

DL Deep Learning

FC Fully Connected

FM Fragmentation

FN False Negative

FP False Positive

FPS Frames Per Second

GPU Graphics Processing Unit

IDs Identification Switch

IoU Intersection over Union

KF Kalman Filter

LAP Linear Assignment Problem

ML Mostly Lost

MT Mostly Tracked

MOT Multi-Object Tracking

MOTA Multi-Object Tracking Accuracy

MOTP Multi-Object Tracking Precision

NMS Non-Maximum Suppression

NN Neural Network

TP True Positive

List of Figures

List of Tables

1

Introduction

In this dissertation, a study and exploitation of two Multi-Object Tracking (MOT) algorithms, the SORT [2] and the Deep-SORT [1], having in view indoor mobile robot applications, are presented.

1.1 Context and Motivation

In the past decade, service robots appeared in the scene, requiring complex and high level navigation autonomy [3]. Service robots typically share the human environment and exhibit "intelligent" behavior to accomplish assigned tasks [4]. Moreover, they have to collaboratively interact and navigate throughout the scene with the presence of humans, requiring explicit actions to fulfill their mission [5]. Therefore, sensors and perception submodules, have to be integrated, collecting and processing environment information. Multi-Object Tracking (MOT) is one of the most important subjects of Computer Vision (CV), aiming to estimate the state of multiple entities (including humans) in the scene. Moreover, MOT maintains the knowledge of objects exclusivity over time, identifying each object with an unique tracking ID [6]. MOT is applied in a variety of other applications, such as: surveillance [7, 8], traffic monitoring [9], medical instruments control [10, 11] and mobile robot navigation, including collision avoidance [12] or target following [13].

Throughout the years, MOT tasks were mainly performed in a tracking by detection paradigm, requiring detections of objects as measurements in frames, to process tracking tasks. This paradigm has the benefit to convert MOT to an association method between measurements and existing tracks [2]. Most of MOT algorithm proposals, are based on different methods of association. Some use the power of Deep Learning (DL) based algorithms computing similarities between appearance features [14, 1], others use linear assignment algorithms associating objects with state information [15, 2]. Nevertheless, a motion module is sometimes required to predict the position of objects of interest in the current frame, facilitating associations. Moreover, by integrating motion modules, a smoother trajectory is computed by the tracker, not relying only on measurements that could be inaccurate [1]. To ensure a good tracking performance, detection and tracking should be performed frame-by-frame, which is time-consuming and can lead to the inability

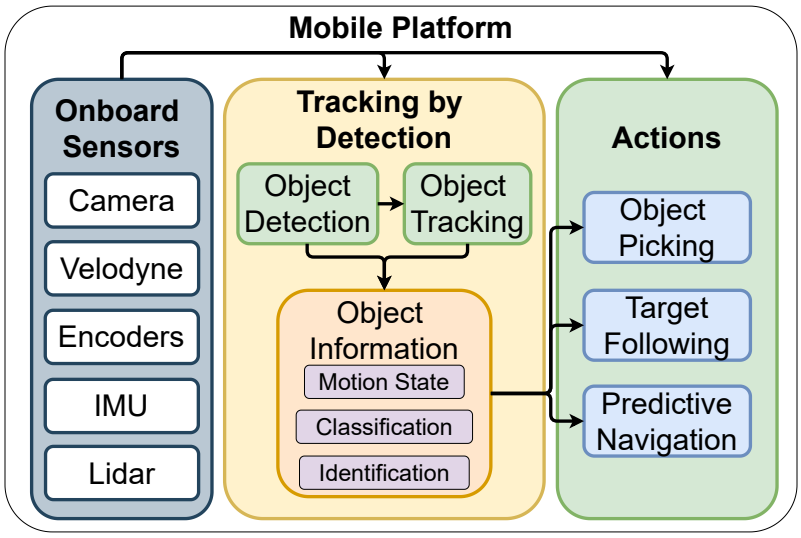


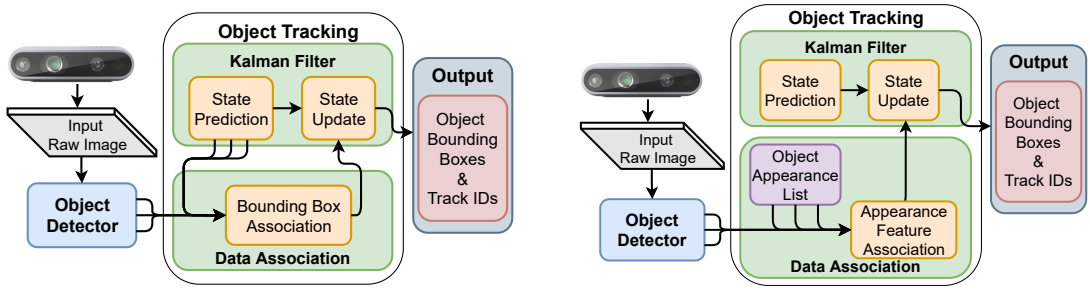
Figure 1.1: Example of a Tracking by Detection algorithm implemented in a mobile platform architecture.

of performing the MOT in real-time [16]. However, technology advance conjugated with the emergence of fast and accurate DL techniques to detect object in the scene [17, 18, 19], promoted the usage of DL techniques to be used as appearance descriptors, contributing with more accurate associations in MOT tasks [1, 20, 21, 22].

Unfortunately, there are significant number of people unable to perform daily tasks on their own, due to severe motor impairments. Service mobile robots can be used to assist motor impaired people, increasing their autonomy and mobility. For that, MOT methods are crucial to endow a robot to perform tasks such as object following [13], object picking [23] or navigating in dynamic environments [24]. Furthermore, is necessary for this algorithms to perform in real-time and to be able to module the motion of entities, limiting the the number of MOT solutions. The usage of Kalman Filter (KF) based algorithms, models trajectories robustly against sensor and modeling noises, and provide feasible predictions.

1.2 Problem Formulation and Proposed Framework

A real-time MOT by Detection algorithm for service robots, must have the ability to accurately estimate the state of every object of interest in real-time. Moreover, the hardware found onboard of a mobile robotic platform has a limited computational power, which implies the development of optimized algorithms. Hence, it is important to find a good compromise between model's accuracy and inference speed [25]. A representation of a tracking by detection method implemented in a mobile platform, is shown in Fig. 1.1. The proposal of a tracking by detection algorithm, requires the usage of an object detector model. Therefore, tracking tasks are limited by the performance of the detector, that is in charge of processing measurements. Occurrences of errors in bounding box detections,



(a) SORT method. **(b)** Deep-SORT method.

Figure 1.2: Representations of SORT and Deep-SORT MOT methods.

may lead to disturbances (not critical) to the tracking phase, however, a miss-detection can lead to the wrong conclusion that the object no longer exist. Thus, is imperatively to develop a MOT system that is able to handle miss-detections and accurately re-identify the track that is being processed.

With the aforementioned aspects, this dissertation focus on the exploitation of association modules, aiming to enhance the association of measurements to existing tracks of two state-of-the-art MOT by detection methods (SORT [2] and Deep-SORT [1]). To have a MOT end-to-end pipeline, the well known object detector network, the YOLOv3 [19], is used to recognize the objects available on images, and feeding the MOT methods. The YOLOv3 has been exploited and applied in works [25, 26] developed at the HCMR-ISR research lab, which makes the YOLOv3 well suited for the tasks proposed in this dissertation. Moreover, the training and evaluation of the YOLOv3's models are not part of the scope of this dissertation.

The SORT [2] and the Deep-SORT [1] are two tracking by detection methods that are focused on performance and inference speed, using the KF algorithm for object tracking. An overview representation of each method is shown in Fig. 1.2. The SORT is focused on associate objects using bounding box states to match measurements with predicted tracks, using the overlap of bounding boxes. On the other hand, to improve the bounding box association, the Deep-SORT uses a DL technique that extracts appearance features from objects bounding box images, in order to associate measurements to tracks. In this dissertation, the SORT and Deep-SORT MOT methods are exploited regarding the data association module, proposing different data association techniques.

1.3 Objectives and Key Contributions

In chronological order, the following list presents established objectives for the development of the proposed study:

1. Detailed analysis of SORT [2] method;
2. Implementation of SORT method, with different Linear Assignment Problem (LAP)

formulations;

3. Detailed analysis of Deep-SORT [1] method;
4. Implementation of Deep-SORT method, with different LAP formulations;
5. Labeling of ISR Tracking Dataset, using the indoor ISR RGB-D Dataset [25];
6. Evaluation and analysis of configurations for the SORT and Deep-SORT algorithms on the MOT17 Dataset [27], using dataset detection file as measurements;
7. Evaluation and analysis of configurations for the SORT and Deep-SORT algorithms on the ISR Tracking Dataset, using ground truth bounding boxes as measurements;
8. Validation and analysis of algorithm configurations on the ISR Tracking Dataset.

The main implementations and contributions of this study are described in the following Chapters of this dissertation:

Developed Work (Chapter 4)

A detailed analysis of the SORT and the Deep-SORT MOT methods is summarized in this chapter. Moreover, an alternative of LAP cost matrix formulation, that have in view to enhance the performance of each MOT method, is presented. Furthermore, specific conditions that were necessarily adopted in the labeling of the new ISR Tracking dataset are described.

Results and Discussion (Chapter 5)

Experimental validation of the SORT, the Deep-SORT and proposed association techniques is presented. Finally, a validation of an end-to-end tracking by detection pipeline using the YOLOv3 object detector on the ISR Tracking dataset is presented.

1.4 Concepts

In this dissertation context, are used the following concepts:

- Object state - Representation of an object, composed by bounding box coordinates and geometric shape. When estimated with a motion model, is also represented with velocity information.
- Appearance features - Vector containing deep appearance information of an image.
- Appearance descriptor - Model capable of extracting deep appearance features from images.
- Track - Object being tracked by a MOT algorithm, contains information of object state and tracking ID. Moreover, in DL-based MOT methods, a track has appearance features information.

- Detection - Object location and classification, attained by an object detector.
- Measurement - Object bounding box attained by an object detector, as a candidate to be associated to tracks on MOT methods. In the context of using class gate metrics, measurement also has the class information.
- Motion module - Module where object state estimation is modeled over time. In the context of this dissertation is denominated as KF Estimation module.
- Motion Model - Mathematical model that represents the evolution of the object's state. In the context of this dissertation, is used the KF.
- Data association module - Module responsible to associate measurements to existing tracks.
- Track management module - Module responsible to delete tracks, initialize new tracks and to establish the output of the MOT method.

2

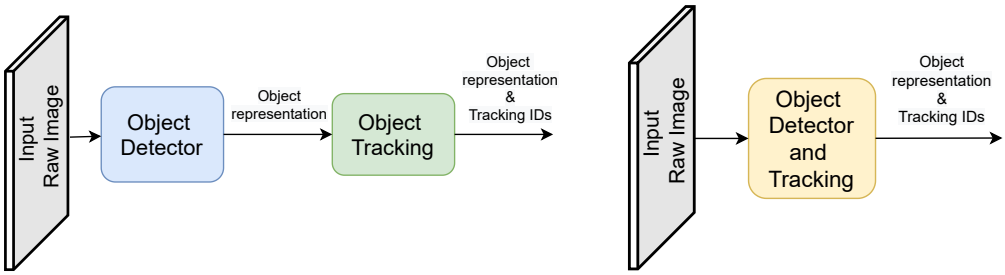
State of the Art

In this chapter, proposed works in the literature related to the tasks of Online Multi-Object Tracking (MOT), capable of performing in a real-time fashion will be summarized.

MOT consists of analyzing a video in order to preserve the exclusivity and track the displacement of objects belonging to one or more categories, without any prior knowledge about appearance or location of targets [6]. Online MOT represents the processing of MOT tasks using only past and present information [28]. Despite real-time MOT being based on online MOT, methods applied in real-time are limited by inference speed, which can directly affect the performance of the tracker. Recent progress on MOT has two focuses: Tracking by Detection and Joint Tracking and Detection (Fig. 2.1). Tracking by Detection [2, 1, 20, 29, 22, 30, 31, 21] makes usage of object detection algorithms [18, 17], before processing tracking tasks [6]. This simplifies tracking as an object association task over consecutive frames, receiving an array of measurements and outputting bounding boxes with a respective tracking ID (Fig. 2.1a). On the other hand, Joint Tracking and Detection methods [16, 32, 33, 34, 35, 36, 14, 37] are able to detect and track objects in a single model (Fig. 2.1b). This approach is an attempt to share detection and appearance embedding extraction tasks in a single model, which can decrease the running time of a tracker [33].

2.1 Tracking by Detection

Over the past years MOT was dominated by the tracking by detection paradigm [37]. This approach took the benefit of object location knowledge to generate a association model that would be able to associate objects over time. Multiple Hypothesis Tracking [15] is a known method capable of calculating hypotheses over measurements to estimate if they should be associated to a track, to be considered as a new track or if it is a miss measurement. The method uses the Kalman Filter (KF) algorithm to model the state of a track and a probabilistic distribution over hypotheses to associate measurements to tracks. In recent works, some algorithms use motion models to assist association of objects over time [2, 1]. Alex Bewley *et. al.* [2] proposed the SORT, a fast and efficient work based on KF to estimate the state of objects. Moreover, for data association, the



(a) Tracking by Detection. **(b)** Joint Tracking and Detection.

Figure 2.1: Different MOT approaches.

Hungarian algorithm [38] was used, associating KF predicted bounding boxes with measurements. An year later, Nicolai Wojke *et. al.* [1], improved the SORT algorithm to the Deep-SORT, by including a novel association step using a DL-based methodology as an appearance descriptor, describing each measurement with appearance features. Furthermore, the association algorithm combines appearance features similarity metric and the Mahalanobis distance between objects states. Despite the usage of DL architectures in detection and association phases, the Deep-SORT method achieved fast inference time and considerable performances on object tracking benchmarks. The Deep-SORT proposal opened a new strand to explore, introducing DL-based appearance descriptors to help association tasks in tracking. Some algorithms follow the CNN methodology to extract features [1, 20, 30, 21], while others use Siamese [39] architectures to compute similarities values between two images [29, 22, 31].

Long Chen *et. al.* [20] proposed the MOTDT, which uses as candidates to associate with existing tracks, a set of measurements and KF predicted object states of tracks. Moreover, a scoring function based on a fully convolutional neural network is used to apply optimal selection from candidates. Furthermore, an appearance descriptor is used, and the euclidean distance between appearance features is calculated to promote associations. Recently, Jiawei He *et. al.* [30] proposed the GMT-CT algorithm that incorporates Graph Partitioning with Deep Feature Learning. It uses appearance features to construct a graph of features. Furthermore, for association tasks, it matches a measurement graph and a track graph, modeling the relationship between measurements and tracks with higher accuracy. Xueqin Zhang *et. al.* [21], developed the DROP framework that consists in a fast object appearance descriptor and a confidence-based data association algorithm. The DROP algorithm extracts appearance features from measurements using a fast pedestrian re-identification network, and then each appearance feature score is used as input in an Hungarian matching algorithm.

Sangyun Lee *et. al.* [29] introduced the FPNS-MOT, which integrates a Siamese architecture and the Feature Pyramid Network [40], uses appearance features and motion features to the association stage. The method computes a similarity vector between features from two different inputs, and then updates tracks using an iterative selection of

Table 2.1: Description of Tracking-by-Detection methods.

Method	Year	DL-Based	KF-Based	Description
SORT [2]	2016		×	Simple and fast KF-based algorithm, that associate objects based on their bounding box appearance.
Deep-SORT [1]	2017	×	×	KF-based algorithm, associate objects based on their appearance description extracted by a CNN re-identification network.
MOTDT [20]	2018	×	×	Deep-SORT related algorithm that uses predicted bounding boxes as candidates for association, in an attempt to solve occlusion problem.
GMT-CT [30]	2021	×	×	Deep-SORT related algorithms, that solves association problem using a graph partitioning based on appearance features.
DROP [21]	2020	×	×	Associates objects using a confidence-based cost to construct the Hungarian algorithm solver. Furthermore, it uses appearance features to determine occlusions in the scene.
FPSN-MOT [29]	2019	×		Uses Siamese and Feature Pyramid based Networks addressing appearance and motion features in the association stage.
Jiating Jin *et. al.* [22]	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.

the maximum scored pair of tracks and measurements. Furthermore, the method outperformed previous proposed methods on the MOTChallenge benchmark [27] with inference time of 10Hz. Jiating Jin *et. al.* [22], developed a MOT method that uses Siamese architecture to enhance the performance of the feature extractor in Deep-SORT [1] algorithm. In addition, it introduced optical flow [41] in the motion module, in order to improve motion model estimation accuracy. Table 2.1 highlights the main characteristics of the aforementioned tracking by detection MOT methods.

2.2 Joint Tracking and Detection

The emergence and fast development of DL techniques helped combining both detection and tracking tasks in the same framework. Besides the aggregation of tasks, DL-based joint tracking and detection approaches, can also increase the performance of object detection [42]. Philipp Bergmann *et. al.* [32] proposed the Tracktor++. The method uses the power of the Faster-RCNN [17] object detection network, in the context of joint tracking and detection. It exploited the bounding box regression of the object detector to predict position of an object in the next frame. In the same year, Zhongdao Wang *et. al.* [33] proposed the joint detection and embedded learning (JDE). The method simultaneously output the location and appearance features of objects in a single forward pass, and for tracking purposes, it does a similar association of that used in Deep-SORT association method, combining appearance features similarity metric and the Mahalanobis distance between objects states. In a different approach, Xingyi Zhou *et. al.* [34] developed the CenterTrack that tracks objects based on their central point. It is a joint tracking and detection algorithm based on a CenterNet [43] object detector that produces detection and

Table 2.2: Description of Joint Tracking and Detection methods.

Method	Year	Description
Tracktor++ [32]	2018	CNN-based object detector converted into an object detector and tracking algorithm, using training methodologies based on tracking.
CenterTrack [34]	2020	Network that produces detection and tracking offsets in central point coordinates.
FairMOT [35]	2020	JDE related algorithm that addresses the unfairness of joint tracking and detection algorithms treating association as a secondary task.
CSTrack++ [36]	2020	Cross-correlation network and Scale-aware attention network to improve collaborations of detection and association sub-tasks.
Siamese Track-RCNN [14]	2020	Siamese network that unifies detection, tracking and association, and can be trained jointly in an end-to-end fashion.
TraDeS [37]	2021	Model that exploits tracking cues in order to improve detection tasks.
SiamMOT [31]	2021	Siamese-based algorithm that computes spacial matching and also models the object motion with an explicit motion model.

tracking offsets by receiving the current image frame, the past image frames, and an image of the previous frame, with objects centers highlighted similarly to an heat map. More recently, Yifu Zhang *et. al.* [35] proposed a anchor-free single-shot deep network, the FairMOT. FairMOT is a method based on previous JDE [33] and TrackRCNN [44] joint tracking and detection methods, and addressed the unfairness of other joint tracking and detection treating association task as a secondary task. A similar work, also based on JDE [33], Chao Liang *et. al.* [36] proposed the CSTrack++, which is a one-shot online model, with two branches. A novel cross-correlation network, to learn similarities of appearance features for detection and for association tasks, and a Scale-aware attention network to aggregate appearance features from different resolutions. Bing Shuai *et. al.* [14], developed a Siamese Track-RCNN that unifies detection, tracking and association in a single network architecture. This approach achieved the best published results on MOTChallenge [27], nonetheless it runs at around 5 FPS. Jialian Wu *et. al.* [37], proposed the Track to Detect and Segment (TraDeS) method, focusing on exploiting tracking cues to help detection. Furthermore, it uses a Cost Volume Association module that models object motion via a 4D cost volume. Furthermore, motion cues are used to enhance detections using a Motion-guided Feature Warper module. Bing Shuai *et. al.* [31] proposed the SiamMOT, through the use of Siamese architectures to compute a spatial matching between the object image and a region of interest. Furthermore, SiamMOT uses an explicit motion model integrated in the final layers of the Siamese network, to compute motion of objects. Aforementioned methods are summarized in detail in Table 2.2.

3

Background Material

In this chapter, a detailed description of the methods that support the development of this dissertation will be presented.

3.1 Kalman Filter

The Kalman Filter (KF) [45] is an optimal recursive data processing algorithm, that estimates the current value of variables of interest, by processing two phases, based on the system and the measurement models. The KF has a remarkable role in sensor fusion applications, once it is able to process measurements of variables of interest. It is also been widely used in robot localization applications [46].

The KF is divided into two phases: Prediction and Update. The prediction phase, uses information of the system model to compute an a priori estimation of new state variables ($\hat{x}_k^-$), and also the error covariance matrix (P_k^-) associated to the a priori estimation of the state. The system is modeled with the transition function (F), that represents the evolution of the state in optimal conditions, and with an input control matrix (B) applied to the control input (u_{k-1}). Furthermore, is assumed the existence of a zero-mean normal distribution process noise (w_{k-1}), with covariance Q, this is: $w_{k-1} \sim \mathcal{N}(0, Q)$. This assumption allows to calculate an a priori estimation of error covariance matrix (P_k^-). This phase is also processed, with information of the previous time-step state vector estimated ($\hat{x}_{k-1}$) and error covariance matrix estimated (P_{k-1}). It is represented by the following linear equations:

$$\hat{x}_k^- = F\hat{x}_{k-1} + Bu_{k-1} \tag{3.1}$$

$$P_k^- = FP_{k-1}F^T + Q \tag{3.2}$$

The KF update phase, uses a measurement (z_k) acquired by a measurement device, and the measurement model, to correct the a priori state estimated and respective covariance matrix. The measurement system is modeled by the state to measurement matrix (H), which relates the state vector to the measurement vector. Furthermore, is also assumed the existence of a zero-mean normal distribution measurement noise (v_k)

with covariance R, this is: $v_k \sim \mathcal{N}(0, R)$. Using the a priori estimation of error covariance matrix and the measurement model, the time-step Kalman Gain (K_k) is calculated. Moreover, the measurement residual ($\tilde{y}_k$) is computed, using the measurement acquired on the time-step (z_k) and the a priori estimation of the state ($\hat{x}_k^-$). Then, with the measurement residual and the Kalman Gain, the state estimation ($\hat{x}_k$) and respective error covariance matrix (P_k) are corrected. This phase is represented by the following linear equation:

$$K_k = P_k^- H^T (R + H P_k^- H^T)^{-1} \tag{3.3}$$

$$\tilde{y}_k = z_k - H\hat{x}_k^- \tag{3.4}$$

$$\hat{x}_k = \hat{x}_k^- + K_k \tilde{y}_k \tag{3.5}$$

$$P_k = (I - K_k H) P_k^- \tag{3.6}$$

3.2 Hungarian Algorithm

The Hungarian algorithm [38] is an approach to solve the Linear Assignment Problem (LAP). LAP can be defined as a problem to assign N individuals to N tasks. Moreover, every individual i ($i \in N$) has a different cost ($c_{i,j}$) to complete the task j ($j \in N$). In addition, the LAP requires a problem formulation, which is the formulation of the cost matrix ($C_{N \times N}$). The objective of the Hungarian algorithm is to associate each individual to a task using the minimum possible cost. Furthermore, each task can only be assigned to a single individual.

Hungarian algorithm steps are listed in Algorithm 1. The algorithm is based on the fact of the addition/subtraction of a constant from the lines/columns of the cost matrix, does not influence the final assignment result. Furthermore, by adding and subtracting constants (such as the smallest element of the matrix), it tries to find the N zero valued elements, which can associate N individuals (lines) to N tasks (columns). Hungarian algorithm receives as input a matrix of cost ($C_{N \times N}$) and outputs an $N \times 2$ array with individuals indices as first column and respective assigned task indices as second column. An overview example of each step of the Hungarian algorithm is shown on Fig. 3.1.

When the number of tasks is different from the number of individuals, additional columns/rows are added into the cost matrix to shape it as a squared matrix. Furthermore, values of those additional columns/rows should be filled by the maximum value of the initial cost matrix.

3.3 Online Multi-Object Tracking

Data Association and motion modules are key components for Online MOT approach, to understand the exclusivity of objects in the scene. Each object in the scene modeled by a MOT algorithm is denominated as track. The SORT [2] and the Deep-SORT [1] are two

Algorithm 1: Hungarian algorithm

Data: Cost matrix $C_{N\times N}$

Result: Assigned Indexes

1 Subtract the smallest entry in each row from each entry in that row;
2 Subtract the smallest entry in each column from each entry in that column;
3 Cover all zeros with the minimum number of lines over rows and columns;
4 **while** *number of lines* $< N$ **do**
5 | s = smallest element, uncovered by a line;
6 | Subtract s from all uncovered elements;
7 | Add s to elements covered by two lines;
8 | Cover all zeros with the minimum number of lines over rows and columns;
9 **end**
10 Optimal assignment is found by assigning indices lines with indices columns where the matrix has element value of 0 (start with lines covering less 0's).

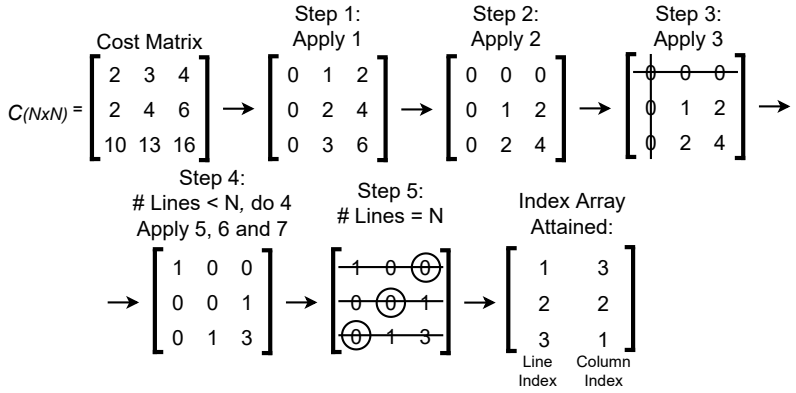


Figure 3.1: Representation of Hungarian algorithm steps described in Algorithm 1.

well known KF-based tracking methods. The Deep-SORT is an upgraded version of the SORT algorithm, as it introduces DL based association metrics on Data association module. Both methods have an identical structure with three main modules: KF Estimation, Data Association and Track Management. An overview of both methods is presented in Fig. 3.2.

3.3.1 SORT

The SORT[1] [2] algorithm recurrently calculates the state of the objects being tracked through a KF algorithm. Moreover it uses Hungarian algorithms to accurately associate modeled objects that are being tracked, to new measurements acquired by an object detector. A detailed overview of SORT algorithm is represented in Fig. 3.3.

[1] https://github.com/abewley/sort

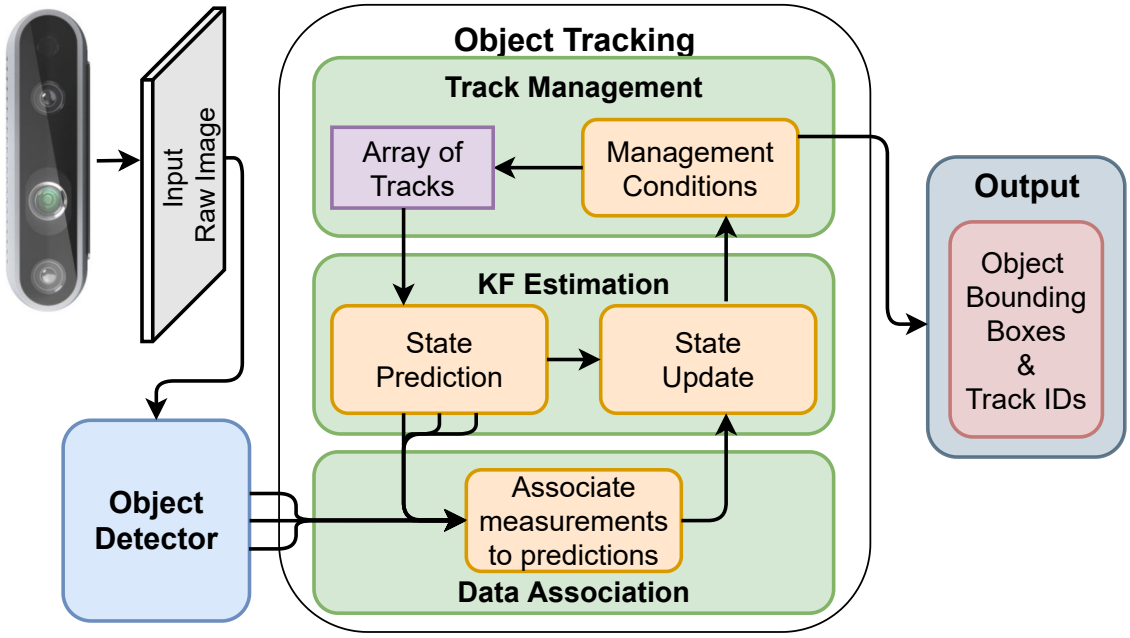


Figure 3.2: An overview example of SORT and Deep-SORT pipelines.

3.3.1.1 KF Estimation

The SORT KF Estimation module uses the KF for each track, to assign the state of the object's bounding box, with a linear constant velocity model, independent to other objects and camera motion. The track state vector ($x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$) is composed by central bounding box coordinates (u) and (v), bounding box scale and aspect ratio (s and r), velocities of central bounding box coordinates ($\dot{u}$ and $\dot{v}$) and bounding box scale velocity ($\dot{s}$). Aspect ratio is considered to be constant.

The KF Estimation module computes an a priori estimation of the state ($\hat{x}_k^-$) for every active track assigned on the previous frame step, using Eq. (3.1). Then, an association between KF predicted bounding boxes and measurements is performed. Matched measurements are used on the KF algorithm, to update the object state. Existing tracks that were not associated with measurements, do not go through update stage, instead prior estimation of the state ($\hat{x}_k^-$) is used as the state of the object in that frame.

3.3.1.2 Data Association

The Data association module is responsible for matching KF's predicted bounding boxes with measurements on the image. This module receives as input, N measurements and M predicted bounding boxes, acquired from each respective KF Estimation module. The module formulates the LAP by computing a cost matrix between each measurement and all predicted bounding boxes (respectively $D_i, i \in \{1 \dots N\}$ and $P_i, i \in \{1 \dots M\}$), with the IoU distance as cost, using the following equations:

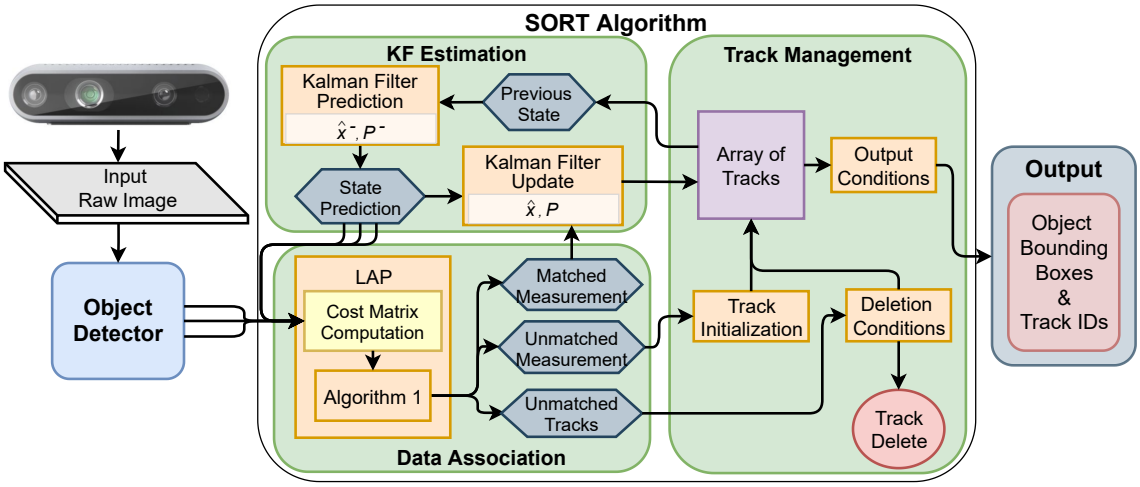


Figure 3.3: SORT detailed workflow representation.

$$IoU_{(D,P)} = \begin{bmatrix} iou_{(D_1,P_1)} & \dots & iou_{(D_1,P_M)} \\ iou_{(D_2,P_1)} & \dots & iou_{(D_2,P_M)} \\ \vdots & \ddots & \vdots \\ iou_{(D_N,P_1)} & \dots & iou_{(D_N,P_M)} \end{bmatrix} \tag{3.7}$$

where

$$iou_{(BB_d,BB_p)} = \frac{Area\ of\ intersection}{Area\ of\ union} \tag{3.8}$$

$$CostMatrix = -IoU_{(D,P)} \tag{3.9}$$

Then, the Hungarian algorithm (Algorithm 1) is computed, attaining an $N \times 2$ array, representing N measurements associated to N tracks. Moreover, associations made are filtered by the IoU threshold (IoU_{min}), discarding associations with IoU cost lower than the threshold. Furthermore, measurements and tracks which were not associated, are used to initialize new tracks and delete tracks.

3.3.1.3 Track Management

This module is responsible to delete and initiate tracks, and to generate the output composed by bounding boxes and respective tracking id, through predefined conditions. Tracks are initialized when a measurement was not matched in Data association module. A track is deleted when it has no matched measurement for a T_{Lost} consecutive number of frames, considering that the object has left the scene. This means that every other track that had matched measurements in less than T_{Lost} frames, will be saved to the next frame step array of tracks. In paper [2], experiments were performed using $T_{Lost} = 1$. Furthermore, the following output condition was applied to the final array of tracks:

1. Track was associated with measurements in every hit_{min} last frames.

3.3.2 Deep-SORT

The Deep-SORT [1] is an improvement of the SORT algorithm, integrating appearance information of objects to enhance associations. Association has an additional appearance metric based on pre-trained Convolutional Neural Network (CNN)s allowing re-identification of tracks, after a long period of occlusion. The KF Estimation and the Track management modules have the same flow as SORT. An overview of the method can be described as in Fig. 3.4.

3.3.2.1 KF Estimation and Track Management

The KF Estimation module is mostly identical to SORT algorithm. The KF has the same constant velocity and linear observation model. However, the state is now an eight-dimensional vector ($x = [u, v, r, h, \dot{u}, \dot{v}, \dot{r}, \dot{h}]^T$), represented by the bounding box center position (u and v), bounding box aspect ratio (r), bounding box height (h) and all respective velocities ($\dot{u}$, $\dot{v}$, $\dot{r}$, $\dot{h}$).

The Track management module is identical to the SORT algorithm (section 3.3.1.3), with an addition of deleting tracks that are not successfully associated in their first three frames. This procedure brings improvements, enabling the algorithm to anticipate a track deletion before it reaches a maximum of age. The method considers every initialized track as tentative, until it is successfully associated in its first three frames, changing its status from tentative to confirmed. Each track k has an age counter (a_k) that increments during KF prediction step and resets to 0 when it is successfully associated with a measurement. The method uses a maximum age of 30 frames ($A_{max} = 30$), meaning that every track that got the last thirty frames non associated, are considered to have left the scene.

3.3.2.2 Data Association

Assignment Problem Association of measurements to tracks is also solved by the Hungarian algorithm. In order to compute the cost matrix, motion and appearance metrics were combined. Motion information was incorporated by the (squared) Mahalanobis distance [47] between predicted states and arrived measurement states:

$$d^{(1)}(i, j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \tag{3.10}$$

where (y_i, S_i) is the projection of the i-th track distribution into measurement space and (d_j) is the j-th measurement. To evaluate if the association between the i-th track and j-th measurement is admissible, a 95% confidence interval computed from the Inverse Chi-Squared (χ^2) distribution is applied (3.11). Since the bounding box is represented in a

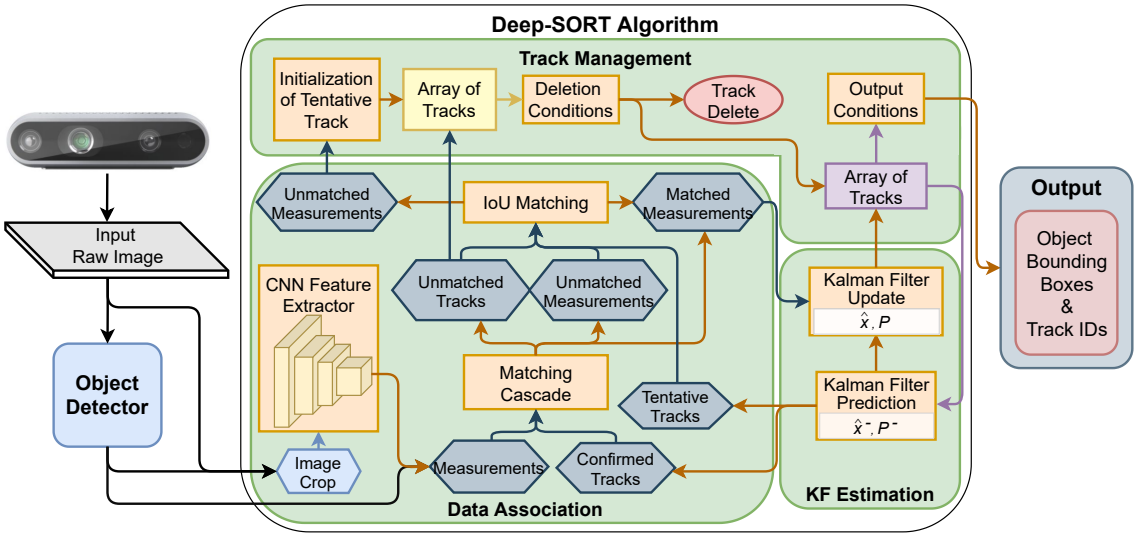


Figure 3.4: Deep-SORT detailed workflow representation.

4-dimensional space, the threshold of confidence is $t^{(1)} = 9.4877$:

$$b_{i,j}^{(1)} = \begin{cases} 1 & \text{if } d^{(1)}(i,j) \leq t^{(1)} \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

In addition to Mahalanobis metric, a second metric based on smallest cosine distance, measures the distance between the i-th track and j-th measurement appearance features:

$$d^{(2)}(i,j) = min\{1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in \mathcal{R}_i\} \tag{3.12}$$

The appearance features (r_j) are computed by a pre-trained CNN model, with structure shown in Table 3.1. The CNN was trained on a large-scale person re-identification dataset [48] using a deep cosine metric learning [49]. Furthermore, a pre-trained model is provided on the authors [1] GitHub[2] repository. For each track k a gallery $\mathcal{R}_i = \{r_k^{(i)}\}_{k=1}^{L_k}$ of the last $L_k = 100$ associated appearance features, are kept in memory. Moreover, a binary variable indicates if an association is admissible according to a $t^{(2)}$ threshold:

$$b_{i,j}^{(2)} = \begin{cases} 1 & \text{if } d^{(2)}(i,j) \leq t^{(2)} \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

To build the association problem, both metrics are combined using a weighted sum

[2]https://github.com/nwojke/deep_sort

Table 3.1: CNN structure of the Deep-SORT's appearance descriptor. Taken from [1].

Layer	Patch Size / Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
Batch and l_2 normalization		128

as follows:

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1 - \lambda) d^{(2)}(i,j) \tag{3.14}$$

Associations are only admissible if they are within the gating region of both metrics $b_{i,j} = 1$:

$$b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{(m)} = 1 \tag{3.15}$$

During experiments, the authors discovered, when there is substantial camera motion, the reasonable setting for this metric is to have $\lambda = 0$ [1]. Moreover, Mahalanobis gating metric (3.11) is still used in this setting [1].

Matching Cascade A matching cascade method is proposed to prioritize associations for more frequently seen objects. The Matching cascade algorithm relies on the iteration over tracks age to solve the LAP for tracks of increasing age. It is outlined on Algorithm 2. Receiving confirmed tracks ($T = \{1, \ldots, M\}$) and measurements ($D = \{1, \ldots, N\}$) indices as input, the algorithm starts by computing the cost and gated matrix (Lines 1 and 2). Then, iterates the age n from 1 to A_{max} and computes the linear assignment between confirmed tracks (T_n) and unmatched measurements ($\mathcal{U}$). This gives priority to tracks of smaller age. At last, it updates the set of unmatched and matched measurements ($\mathcal{U}, \mathcal{M}$). After matching cascade, remaining unmatched tracks of age $n = 1$ and tentative tracks, go through the association algorithm proposed in SORT (Section 3.3.1.2). This is a second association metric that helps to account for sudden appearance changes for tracks with no history.

Algorithm 2: Matching Cascade, from [1].

Data: Confirmed Track indices $T = \{1, \ldots, M\}$, Measurement indices $D = \{1, \ldots, N\}$, Maximum age A_{max}

Compute cost matrix $C = [c_{i,j}]$ using Eq. 3.14
Compute gate matrix $B = [b_{i,j}]$ using Eq. 3.15
Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
Initialize set of unmatched measurements $\mathcal{U} \leftarrow D$
for $n \in \{1, \ldots, A_{max}\}$ **do**
    Select tracks by age $T_n \leftarrow \{i \in T | a_i = n\}$
    $[x_{i,j}] \leftarrow$ min_cost_matching$(C, T_n, \mathcal{U})$
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j = |b_{i,j} \cdot x_{i,j} > 0\}$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
end

Result: M, U

3.3.3 Multi-Object Tracking Metrics

In a frame to frame evaluation, to recognize if an object is being correctly identified, an association algorithm is computed between bounding boxes resulted from MOT method and ground truths. This association algorithm can be replicated as the one described in sub-section 3.3.1.2. The association method is used with an IoU threshold of 0.5. True Positive (TP) are the number of correctly matched pairs of ground truths and bounding boxes. Is only a TP when the previous matched tracking ID is the same as the current one. If the ground truth sequence has no previous matched tracking ID associated, it also counts as TP. For ground truth sequences that are matched with a bounding box with a new tracking ID (previous matched tracking ID is different from the new one), they are counted as an Identification Switch (IDs). For ground truth objects without any matched bounding box resulted, they are counted as False Negative (FN). For bounding boxes that do not have any matched ground truth objects, they are counted as False Positive (FP).

The Multi-Object Tracking Accuracy (MOTA) (3.16), which is a score of accuracy, is interpreted as a sum of misses (FN, FP, and IDs) of the method in the video sequence, over the total number of ground truth bounding boxes (g):

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{IDs}_t)}{\sum_t g_t} \tag{3.16}$$

The Multi-Object Tracking Precision (MOTP) is a score of precision related to matched ground truths and bounding boxes (IoU greater than 0.5):

$$\text{MOTP} = \frac{\sum_{i,t} IoU_{i,t}}{\sum_t c_t} \tag{3.17}$$

where c_t is the number of matches in frame t and $IoU_{i,t}$ is the bounding box overlap of

bounding box i with its assigned ground truth. MOTP (3.17) represents the total position error for matched track-measurement pairs over all frames, averaged by the total number of matches. It quantifies the localization accuracy of the detector, therefore, it provides some information related to the state modeled by the KF Estimation.

Mostly Tracked (MT), Mostly Lost (ML) and Fragmentation (FM) define an evaluation correspondent to the quality of ground truth sequences tracked. Mostly Tracked (MT) represent the percentage of ground truth object sequence mostly tracked by the method. In other words, it is the percentage of the ground truth sequences, which have the same label for at least 80% of their life span. Otherwise, Mostly Lost (ML) is the percentage of sequences, which are tracked for at most 20% of their life span. Fragmentation (FM) is the total number of times a ground truth trajectory is interrupted, *i.e.*, a ground truth sequence changes status from tracked to untracked and later it resumes the tracked status.

Frames Per Second (FPS) metric evaluates the mean speed of the method. It is the total number of frames divided by the time spent to process all of them (3.18):

$$\text{FPS} = \frac{\text{Number of Frames}}{\text{Time}} \tag{3.18}$$

3.4 Deep Learning Approaches

3.4.1 Convolutional Neural Network

CNNs are a type of Neural Network (NN) with impressive success on top classification competitions [50, 51] and object detection tasks [18, 17]. CNN is a feed forward NN that use convolutional layers to extract features from input data. It is in charge of extracting appearance features, such as corners, lines, shapes and patterns among classes. A CNN architecture is composed by two main modules, as shown in Fig. 3.5: Feature extraction and classification. The feature extraction module is the stage were the CNN generates feature maps with information captured by convolutional layers and pooling layers. The classification module uses Fully Connected (FC) layers to accurately classify the outputted feature map into different predefined classes.

The feature extraction module is mostly composed by convolutional layers. In first convolutional layers, low level appearance features are extracted, such as corners, edges and lines, as more layers are added, more high level appearance features are captured, such as objects, structures or shapes, combining low level appearance feature information [52]. The extraction of features is made by kernels, also called filters. As more kernels are used in one layer, more feature maps are added to the output depth dimension, constructing a 3D output volume. A kernel is a weight matrix, with a much smaller dimension compared to input width and height. Also, they are applied systematically in a sliding window manner, across the width and height of the input volume. Each convolutional layer computes a

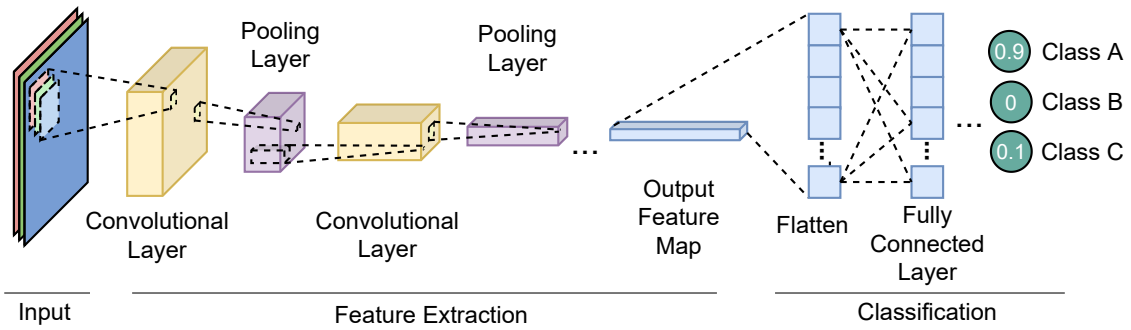


Figure 3.5: Overview of a CNN for 2D RGB image classification.

sum of the scalar product between input data local mask and the kernel matrices weights. To introduce non-linear proprieties, an activation function is used to transform the feature map data. Each activation function has its proprieties and they are used dependently to the objective of the layer [50].

Pooling layers are also used in CNNs to appropriately reduce feature map's spatial size and to decrease computational needs. Furthermore, pooling layers help to control the overfitting during training stage by extracting dominant features. There are some different pooling operations such as: Max pooling, Average pooling and Sum pooling [50].

As shown in Fig. 3.5, the classification stage is composed by the flatten layer, and FC layers. The flatten layer is responsible to reshape the feature map attained in the Feature extraction module, into a vector. FC layers can be added sequentially and process all the information from previous layers, aiming to score different labeled classes on an output vector.

3.4.2 Deep Residual Learning

As NN use more layers to get deeper features, accuracy gets saturated and then degrades rapidly [53]. Deep residual learning [53] was proposed in 2015 to counter this consequence and to ease the optimization of networks. Networks based on residual learning reached promising results and won first places on image classification, detection and segmentation contests [51, 54]. Residual block is represented in Fig. 3.6. Residual blocks compute a sum between the previous layer feature map with an early computed feature map. This maintains a reference to the block input, as the function of the output is based on it. Moreover, residual blocks allow memory to flow from early to future layers and also keep a reference throughout the network showing optimization and accuracy gain, by increasing networks number of residual layer [53].

3.4.3 YOLOv3

The YOLOv3 [19] is a successful real-time CNN-based object detector developed by *Joseph Redmon et. al.*. YOLOv3 uses the Darknet-53 [19] as its backbone network for fea-

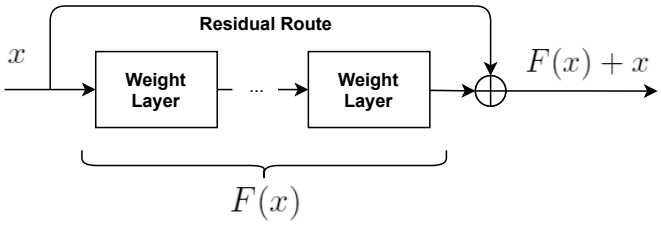


Figure 3.6: Overview of the Residual block.

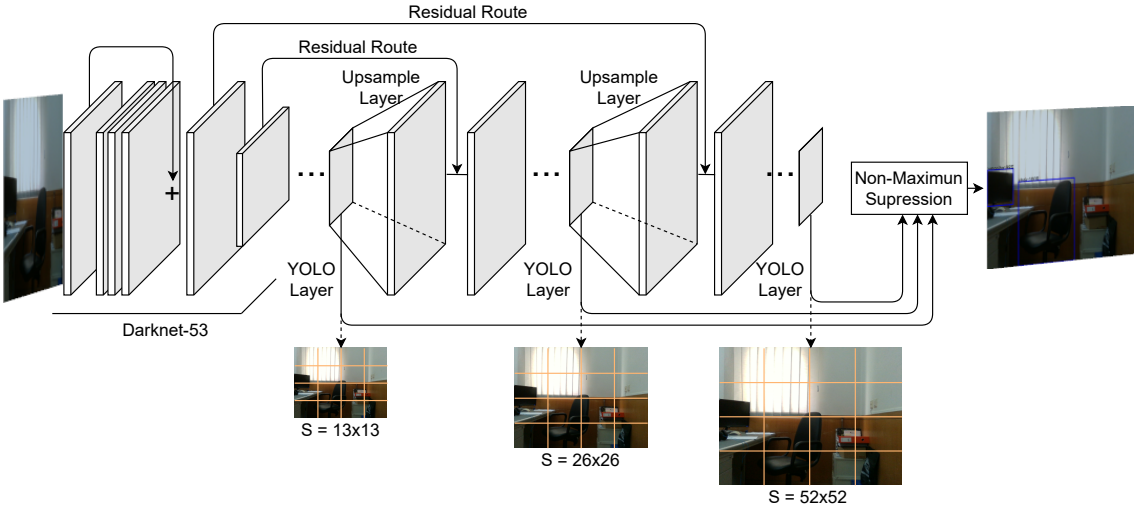


Figure 3.7: Representation of YOLOv3 architecture.

ture extraction. The Darknet-53 network is composed by convolutional layers and residual blocks having a total of 53 layers. For detection purposes, YOLOv3 integrates more 53 layers are added (totaling 106 layer), combining convolutional, upsampling, residual and detection layers. The YOLOv3 architecture, represented on Fig. 3.7, has three detection layers over the pipeline, allowing to recognize objects using different feature map sizes. Each detection layer, known as YOLO layer, divides the input image into $S \times S$ grid cells. Respectively they have strides of 32, 16 and 8, meaning that for an input of 416×416, detections will occur on 13×13, 26×26 and 52×52 scales. Each grid cell in the YOLO layer, is responsible to detect the object that has his center on it. This means that if a cell contains the center of an object, it must be able to detect that object, since other cells wont be able to detect it. Bounding boxes are predicted using dimensional clusters as anchor boxes. Furthermore, each cell predicts three anchor boxes, respective confidence scores and respective class prediction. Residual layers compute a concatenation between upsampled and early calculated feature maps, initializing new processes of detections for new scales. Each identity route used to compute residual layers are taken from different feature extraction layers, thus performing detection steps with different processed features for different scales. After all detection stages, to eliminate multiple detection of the same object, the Non-Maximum Suppression (NMS) method is used.

4

Developed Work

In this chapter, the developed work to fulfil the proposed dissertation, is presented in detail.

4.1 Methodology

In this work, the SORT [2] and the Deep-SORT [1] were exploited to operate as Multi-Object Tracking (MOT) algorithms. Also, a YOLOv3 [19] object detector network, was used to evaluate an end-to-end tracking by detection pipeline. The study got started by analyzing and comprehending each online code, for some important parameters not mentioned on each paper [2, 1]. Furthermore, the SORT and the Deep-SORT algorithms, were exploited with different data association techniques. New formulations of the LAP were implemented, by adding new distance metrics to establish the cost matrix for the LAP formulation. Moreover, in order to apply the SORT and the Deep-SORT methods in a multi-class environment, a class gate metric, to disallow association between objects from different classes, was implemented. For evaluation and validation tasks in indoor, multi-class and lowered point of view conditions, a labeling of an indoor tracking dataset was performed, using the ISR RGB-D Dataset [25]. An overview of the pipeline is described in Fig. 4.1.

4.2 Multi-Object Tracking Methods

4.2.1 SORT

To implement the SORT [2] algorithm, the main code available online[1] was used. Furthermore, new formulations of the LAP are implemented, by combining different metrics based on bounding box shape, euclidean distance and IoU, between bounding boxes over consecutive frames. Moreover, a class gate metric, indicating admissible associations according to the class of objects, is implemented. Also, to test different values of the variable T_{Lost}, a different approach, based on dividing the acquired scene image into critical and non critical zones, to delete tracks that could be out of the scene, is implemented.

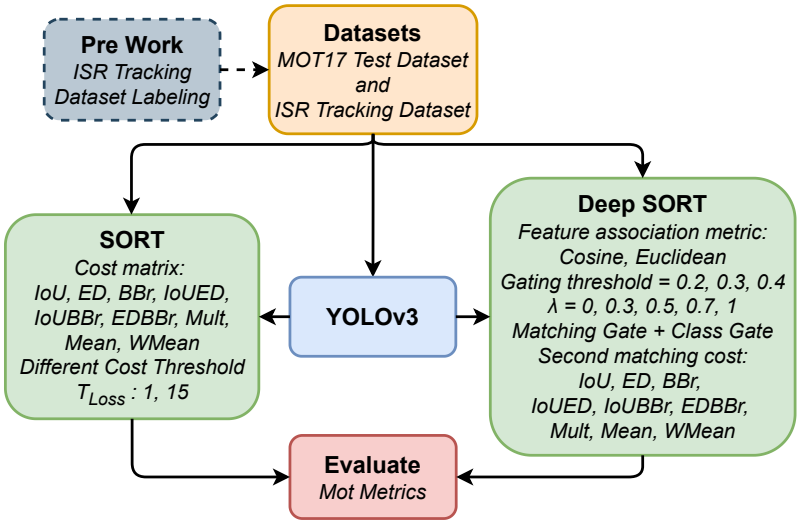


Figure 4.1: Multi-Object Tracking overview pipeline, using YOLOv3 as object detector, and the SORT and the Deep-SORT as MOT methods.

4.2.1.1 KF Estimation

An identical Kalman Filter (KF) Estimation module to the one online available[1] [2] is implemented. Each new object tracking sequence is initialized with a new KF. The initial state (x_0) is assembled with the central point of the bounding box in pixel coordinates (u, v), the bounding box scale ($s = width \times height$), the bounding box ratio ($r = \frac{width}{height}$) and null velocities:

$$x_0 = \Big[u, v, s, r, 0, 0, 0\Big]^T \tag{4.1}$$

Furthermore, the initial covariance matrix (P_0) is initialized with higher uncertainty to initial velocities, since the state is initialized with velocities at null value:

$$P_0 = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^4 \end{bmatrix} \tag{4.2}$$

The KF matrices are implemented as follows:

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-4} \end{bmatrix} \tag{4.4}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \tag{4.6}$$

4.2.1.2 Data Association

Different approaches to formulate an association problem with different bounding box distance metrics were studied. An euclidean distance based cost matrix ($ED_{(D,P)}$) is proposed:

$$ED_{(D,P)} = \begin{bmatrix} ed_{(D_1,P_1)} & \dots & ed_{(D_1,P_M)} \\ ed_{(D_2,P_1)} & \dots & ed_{(D_2,P_M)} \\ \vdots & \ddots & \vdots \\ ed_{(D_N,P_1)} & \dots & ed_{(D_N,P_M)} \end{bmatrix} \tag{4.7}$$

where

$$ed_{(BB_d,BB_p)} = 1 - \frac{\sqrt{(u_d - u_p)^2 + (v_d - v_p)^2}}{\frac{1}{2}\sqrt{image_{height}^2 + image_{width}^2}} \tag{4.8}$$

which is the distance between bounding box central points normalized into half of the image dimension. Furthermore, to incorporate the matrix into a maximization solution using the Hungarian algorithm, is subtracted the normalized euclidean distance to the value 1 (4.8). The bounding box ratio based cost matrix ($BBr_{(D,\ P)}$) is implemented as a ratio between the product of each width and height:

$$BBr_{(D,P)} = \begin{bmatrix} bbr_{(D_1,P_1)} & \dots & bbr_{(D_1,P_M)} \\ bbr_{(D_2,P_1)} & \dots & bbr_{(D_2,P_M)} \\ \vdots & \ddots & \vdots \\ bbr_{(D_N,P_1)} & \dots & bbr_{(D_N,P_M)} \end{bmatrix} \tag{4.9}$$

where

$$bbr_{(BB_d,BB_p)} = min(\frac{w_d * h_d}{w_p * h_p}, \frac{w_p * h_p}{w_d * h_d}) \tag{4.10}$$

Also, for boxes with similar shape, this metric outcomes with a value closer to 1, contrasting to values close to 0 or much greater than 1 otherwise. For that reason, the minimum between the ratio and its inverse is applied, to get a value that is within $[0, 1]$ range (4.10). Due to the implementation of new cost matrices, the IoU_{min} variable is changed to $Thresh_{cost}$, and will be denominated as so throughout the following chapters. With the aforementioned cost matrices and also the IoU cost matrix used by the SORT algorithm, the following six configurations are implemented:

$$IoUED_{(D,P)} = IoU_{(D,P)} \circ ED_{(D,P)} \tag{4.11}$$

$$IoUBBr_{(D,P)} = IoU_{(D,P)} \circ BBr_{(D,P)} \tag{4.12}$$

$$EDBBr_{(D,P)} = ED_{(D,P)} \circ BBr_{(D,P)} \tag{4.13}$$

where this first three, are the Hadamard product (element-wise product) between two matrices. Then, the other three conjugations get a cost based on every matrix. The Hadamard product of every cost matrix:

$$Mult_{(D,P)} = IoU_{(D,P)} \circ ED_{(D,P)} \circ BBr_{(D,P)} \tag{4.14}$$

The element-wise mean of every cost matrix:

$$Mean_{(i,j)} = \frac{IoU_{(i,j)} + ED_{(i,j)} + BBr_{(i,j)}}{3},\ i \in D,\ j \in P \tag{4.15}$$

And the element-wise weighted mean of every cost matrix:

$$WMean_{(i,j)} = \lambda_{IoU} \cdot IoU_{(i,j)} + \lambda_{ED} \cdot ED_{(i,j)} + \lambda_{BBr} \cdot BBr_{(i,j)}, \\ i \in D,\, j \in P,\, \lambda_{IoU} + \lambda_{ED} + \lambda_{BBr} = 1 \quad (4.16)$$

To improve tracking algorithms in multi-class environments, a new cost matrix is computed based on tracks and measurements object class:

$$Cost^{C}_{(i,j)}(Cost_{(i,j)}) = \begin{cases} Cost_{(i,j)} & \text{if } Class_i = Class_j \\ 0 & \text{otherwise} \end{cases}, i \in D,\, j \in P \quad (4.17)$$

Computing the final cost matrix as follows:

$$CostMatrix = -Cost^{C}_{(D,P)}(matrix_{(D,P)}) \quad (4.18)$$

where $matrix$ is selected form one of the following: IoU (3.7), ED (4.7), BBr (4.9), $IoUED$ (4.11), $IoUBBr$ (4.12), $EDBBr$ (4.13), $Mult$ (4.14), $Mean$ (4.15), $WMean$ (4.16).

4.2.1.3 Track Management

Deletion of tracks is computed for tracks that have a number of frames since the last association bigger than the maximum age of tracks ($time\_since\_update > T_{Lost}$). Furthermore, the output array containing bounding boxes and tracking identifications, is elaborated using following conditions:

1. Track is associated with measurement in the current frame;
2. The hit streak of the track is greater or equal to the number of minimum hits, or method is running for a shorter or equal period of hits margin.

Increasing the number of T_{Lost} frames had the drawback of trying to associate tracks that could already be out of the scene, or possibly associate lost tracks with recently entered measurements in the scene. Therefore, conditions to delete tracks that could possible go out of the scene are added to the implementation. The deletion approach consists in dividing the image in tracking zone and critical zone. It is considered a critical zone the two side margins of the frame that correspond to ρ % of the frame width, indicating the zone were objects normally appear or disappear. In example of $\rho = 20\%$, the frame had 10% of the width frame corresponding to the left side and 10% to the right side, has seen in Fig. 4.2. Furthermore, using "OR" logic operators, the following delete conditions are applied:

1. Track $skipped\_frames > T_{Lost}$.
2. Track central point is outside of frame.

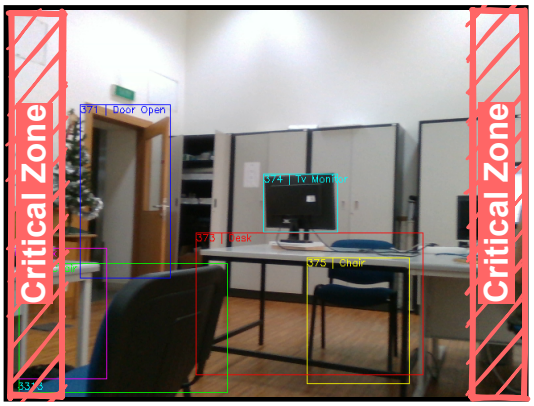


Figure 4.2: Representation of critical zone, for $\rho = 20\%$.

3. Track *skipped_frames* $> \frac{T_{Lost}}{2}$ and Track predicted central point is inside critical zone.

4.2.2 Deep-SORT

The Deep-SORT follows a publicly GitHub online code[1], with the feature extractor model modified to PyTorch library, and with a provided pre-trained model.

To the Deep-SORT implementation, Eq. (3.14) is added, since the code was only assembled for $\lambda = 0$. Furthermore, the second matching stage is implemented with different cost matrix formulation. Moreover, a class gate metric is added to the gate metric, and to the second matching stage problem formulation.

4.2.2.1 KF Estimation

Each new track is initialized with a KF. The initial state (x_0) is assembled with the central point of the bounding box in pixel coordinates (u, v), the bounding box ratio($r = \frac{width}{height}$), the bounding box height (h), and null velocities:

$$x_0 = \Big[u, v, r, h, 0, 0, 0, 0,\Big]^T \tag{4.19}$$

Furthermore, motion and observation uncertainty are chosen relative to the current aspect ratio (r) of the state estimate and uses two weights ($\lambda_{\sigma_{pos}}$ and $\lambda_{\sigma_{vel}}$) to control the amount of uncertainty in each model. Moreover, the initial error covariance matrix (P_0) is also projected with uncertainty weights, some element factors and the aspect ratio. The two weights used in the algorithm are: $\lambda_{\sigma_{pos}} = \frac{1}{20}$ and $\lambda_{\sigma_{vel}} = \frac{1}{160}$. This is an irregular projection of the KF, but is useful to discriminate objects further, from objects closer to the camera. This implies that each track has different process and measurement noise matrices over time, based on their aspect ratio (r) estimation. The KF matrices are projected similarly to the online code,[1] as following:

[1] https://github.com/ZQPei/deep_sort_pytorch

$$P_0 = diag\left(\begin{bmatrix} 2 \cdot \lambda_{\sigma_{pos}} \cdot x_0(r) \\ 2 \cdot \lambda_{\sigma_{pos}} \cdot x_0(r) \\ 10^{-2} \\ 2 \cdot \lambda_{\sigma_{pos}} \cdot x_0(r) \\ 10 \cdot \lambda_{\sigma_{vel}} \cdot x_0(r) \\ 10 \cdot \lambda_{\sigma_{vel}} \cdot x_0(r) \\ 10^{-5} \\ 10 \cdot \lambda_{\sigma_{vel}} \cdot x_0(r) \end{bmatrix}^2\right) \tag{4.20}$$

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.21}$$

$$Q_k = diag\left(\begin{bmatrix} \lambda_{\sigma_{pos}} \cdot x_k(r) \\ \lambda_{\sigma_{pos}} \cdot x_k(r) \\ 10^{-2} \\ \lambda_{\sigma_{pos}} \cdot x_k(r) \\ \lambda_{\sigma_{vel}} \cdot x_k(r) \\ \lambda_{\sigma_{vel}} \cdot x_k(r) \\ 10^{-5} \\ \lambda_{\sigma_{vel}} \cdot x_k(r) \end{bmatrix}^2\right) \tag{4.22}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.23}$$

$$R_k = diag\left(\begin{bmatrix} \lambda_{\sigma_{pos}} \cdot x_k(r) \\ \lambda_{\sigma_{pos}} \cdot x_k(r) \\ 10^{-1} \\ \lambda_{\sigma_{pos}} \cdot x_k(r) \end{bmatrix}^2\right) \tag{4.24}$$

4.2.2.2 Data Association

In the first stage association, the association problem is built with Eq. (3.14). Furthermore, the distance between track appearance features and measurement appearance features ($d^{(2)}$) can also be computed by the squared euclidean distance between appearance features, which is given by the follow equation:

$$d^{(2)}_{euc}(i,j) = \|A_i\|_2^2 - 2A_i^T F_j + \|F_j\|_2^2,\ i \in \{1,\dots,L\},\ j \in \{1,\dots,N\} \tag{4.25}$$

where A is an $L \times B$ matrix with L last associated appearance features of dimension B, F is an $N \times B$ matrix with N measurements with a B dimensional appearance features and A_i^T is the transpose vector of A_i. Moreover, a class gate metric is added to the gate metric as following:

$$b^{(3)}_{i,j} = \begin{cases} 1 & \text{if } Class_i = Class_j \\ 0 & \text{otherwise} \end{cases} \tag{4.26}$$

$$b_{i,j} = \prod_{m=1}^{3} b^{(m)}_{i,j} \tag{4.27}$$

to filter possible associations between tracks and measurements with different classes.

In the second stage association, the different cost matrices formulated on SORT algorithm, are also applied. Small differences, such as formulation of a minimum cost LAP and the different approach to associate tracks to measurements and not otherwise, were taken into account. The following cost matrices are used:

$$C^{metric}_{(T,D)} = [1 - metric_{(T,D)}],\ metric \in [IoU,\ ED,\ BBr] \tag{4.28}$$

$$C^{Mult}_{(T,D)} = C^{iou}_{(T,D)} \circ C^{ED}_{(T,D)} \circ C^{BBr}_{(T,D)} \tag{4.29}$$

$$C^{Mean}_{(i,j)} = \frac{C^{iou}_{(i,j)} + C^{ED}_{(i,j)} + C^{BBr}_{(i,j)}}{3},\ i \in T,\ j \in D \tag{4.30}$$

$$C^{WMean}_{(i,j)} = \lambda_{IoU} \cdot C^{iou}_{(i,j)} + \lambda_{ED} \cdot C^{ED}_{(i,j)} + \lambda_{BBr} \cdot C^{BBr}_{(i,j)},$$
$$i \in T,\ j \in D,\ \lambda_{IoU} + \lambda_{ED} + \lambda_{BBr} = 1 \tag{4.31}$$

Paired with the threshold to associate tracks to measurements, an indicator to filter associations made with different object classes is implemented.

Since the Mahalanobis threshold ($t^{(1)}$) is a constant based on the the inverse Chi-Squared distribution, further mentions of the first stage threshold will be referred to the appearance-based association threshold ($t^{(2)}$), and will be introduced as $dist^1$. Moreover, the second association stage threshold will be mentioned as $dist^2$.

4.2.2.3 Track Management

Each track is represented by a state: Tentative, Confirmed or Deleted. Every track is initialized in the Tentative state. Tracks need to pass the Tentative stage by getting three consecutive associations ($hit_{min} = 3$) to be considered as Confirmed. Once a track is Confirmed, it only needs a single association to be considered as an output, on that current time-step. A track is Deleted if it is in the Tentative state and no association was made in the current time-step, or it is in the Confirmed state and no associations were made in the last $A_{max} = 30$ frames.

4.2.3 Multi-Object Tracking Evaluation Metrics

For quantitative evaluation of each method and proposed data association techniques, was implemented an evaluation script with MOTA, MOTP, TP, FN, FP, IDs, MT, ML and FM metrics. The script gets the result and ground truth *.txt* files and evaluates every sequence with a matching threshold of IoU = 0.5.

4.3 Object Detection + Object Tracking

As referenced in Fig. 4.1, the overall pipeline to fulfil this study, requires an object detector method to obtain objects bounding boxes and classifications in order to perform tracking methodologies. Therefore, YOLOv3 object detector network was used.

The overall pipeline is implemented to evaluate each SORT and Deep-SORT methods in a real-time strategy. Moreover, the best configuration for the two MOT methods, are chosen, to understand the importance of the object detector and how it affects tracking methodologies. Bounding boxes attained by the YOLOv3 method are filtered using a NMS algorithm, only for bounding boxes with confidence over 0.5 ($conf_{min} = 0.5$), and with an overlap threshold of 0.5 ($nms_{min} = 0.5$).

4.4 ISR Tracking Dataset Labeling

The ISR RGB-D Dataset [25], is an object-related dataset, recorded in the ISR facilities using a camera sensor onboard the InterBot [55] platform. The dataset presents a mission performed by the platform in a real scenario setting, representing the object conditions under mobile robot platforms may navigate. The ISR dataset was recorded at 30 FPS in 640×480 resolution. Moreover, this dataset contains a total of 10000 RGB-D raw images and labels of 10 object classes (unknown, person, laptop, tv-monitor, chair, toilet, sink, desk, door-open and door-closed) labeled in every 4^{th} frame achieving a total of 7832 object-centric RGB-D images. Since this dataset represents the object conditions from a mobile robot point-of-view, it is utmost important to evaluate object tracking methods in such conditions. To accomplish that, object tracking labels need to be add to the original

labels. Also, due to objects being labeled every 4^{th} frame, a procedure of labeling bounding boxes throughout every frame was necessary. Therefore, the YOLOv3 [19] network was trained using labeled frames and executed to detect objects in every other non-labeled frame. To label an indoor object tracking dataset based on detections, some conditions have to be defined. The reason is in respect of the same objects being present in the scene multiple different times. Moreover, an image detector does not handles occlusions, so this dataset has no labeled occluded objects. Also, the existence of miss detections of objects in a continuous frame sequence are recurrent in this dataset. Based on the existence of tracks fragmentation in ground truth parameters, the same tracking label was used to a maximum number of 15 frames without detections. Concerning the frame rate of the dataset being 30 FPS, the limit time that an object continue with its ground truth tracking ID, is if the object got detected every half a second.

The tracking labeling by hand was proceeded for the entire 10000 frame dataset, for every object except "unknown" labeled objects. The ISR Tracking dataset ground truth is included in a single ".txt" file, with rows corresponding to the amount of object instances in the dataset. Each line contains 7 values, respectively corresponding to: <Frame number>, <Tracking id>, <Bounding box center x>, <Bounding box center y>, <Bounding box width>, <Bounding box height>, <Object class>. The decision of maintaining the object class in the dataset, is for the possibility of understanding the behaviour of static and dynamic objects, in future works. After proper labeling, the dataset got a total of 32635 bounding boxes, 329 sequences of objects and a video length of 05 : 33 minutes. Furthermore, the ISR Tracking Dataset only contains RGB images.

4.5 Implementation Details

This study was performed by the usage of a NVIDIA GeForce RTX 2060 SUPER Graphics Processing Unit (GPU) and AMD Ryzen™ 5 3600 Central Processing Unit (CPU). Moreover, every algorithm is implemented in Python 3.8.5 object-oriented programming language, with help of Numpy, SciPy and OpenCV Python Libraries and the PyTorch framework. Also, to speed up computing applications using the power of GPU, the PyTorch with CUDA toolkit version 11.2.152 is used, that includes GPU-accelerated libraries. Furthermore, the PyCharm Community edition open sourced Python Integrated Development Environment, is used to elaborate the Python project in study.

5

Results and Discussion

This chapter presents the results of the SORT, the Deep-SORT and proposed data association techniques, in the MOT17 Training Sequences Dataset and ISR Tracking Dataset. Furthermore, a tracking by detection framework was evaluated, using the YOLOv3 object detector to obtain measurements for the MOT algorithms, in the context of indoor robotic platforms

5.1 Dataset

To evaluate the performance of object tracking methods, is fundamental to have a reliable dataset. This dissertation proposes a study on MOT algorithms for indoor, multi-class and recorded in mobile robotic platforms, in which datasets with this conditions are required. The MOT benchmarks do not fulfill the requirements for our study. Nonetheless, MOT benchmarks are useful in performing comparative analysis of results. Therefore, to discuss the performance of the adopted methodology, the MOT17 Training set [27] and the ISR Tracking Dataset (labeled to fulfill requirements of this study) were used.

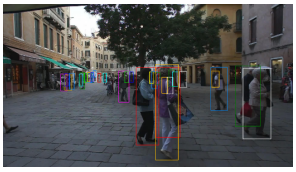
5.1.1 MOT17

The MOT17 [27][3] is a multi person tracking benchmark divided into 14 sequences with highly crowded scenarios, different viewpoints, weather conditions and camera motion. Each sequence comes in three different files that contain detections acquired from three different detectors: FasterRCNN [17], SDP [56] and DPM [57]. The MOT17 dataset is also divided into training and test sets, where training sequences are integrated with detection and ground truth files, whilst test sequences have only detection files. Consequently, we chose to only use the training dataset, in order to run our own evaluations. The dataset is labeled with visible and also occluded humans on the scene as previewed in Fig. 5.1. A brief description of each training sequence is presented on Table 5.1.

[3] https://motchallenge.net/data/MOT17/

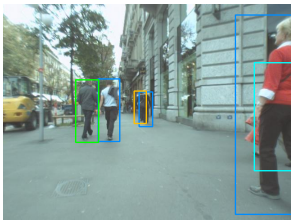
Table 5.1: Description of the MOT17 training sequences.

Sequence Number	FPS	Resolution	Length	Tracks	Boxes	Density	Description
02 (Fig. 5.1a)	30	1920 × 1080	600(00 : 20)	62	18581	31.0	People walking around a large square.
04 (Fig. 5.1b)	30	1920 × 1080	1050(00 : 35)	83	47557	45.3	Pedestrian street at night, elevated viewpoint.
05 (Fig. 5.1c)	14	640 × 480	837(01 : 00)	133	6917	8.3	Street scene from a moving platform.
09 (Fig. 5.1d)	30	1920 × 1080	525(00 : 18)	26	5325	10.1	A pedestrian street scene filmed from a low angle.
10 (Fig. 5.1e)	30	1920 × 1080	654(00 : 22)	57	12839	19.6	A pedestrian scene filmed at night by a moving camera
11 (Fig. 5.1f)	30	1920 × 1080	900(00 : 30)	75	9436	10.5	Forward moving camera in a busy shopping mall.
13	25	1920 × 1080	750(00 : 30)	110	11642	15.5	Filmed from a bus on a busy intersection.
Total			5316(03:25)	546	112297		



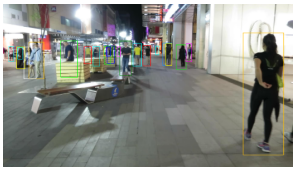
(a) MOT17 02 FRCNN.

(b) MOT17 04 FRCNN.



(c) MOT17 05 FRCNN.

(d) MOT17 09 FRCNN.



(e) MOT17 10 FRCNN.

(f) MOT17 11 FRCNN.

Figure 5.1: Image examples of the MOT17 Train sequences with ground truth bounding boxes.

5.1.2 ISR Tracking Dataset

The ISR Tracking Dataset was used in two different ways: To evaluate MOT algorithms using detections from the dataset and to evaluate MOT methods with YOLOv3 object detector, using images as input. For MOT methods evaluation, the ISR Tracking Dataset was used as it was labeled, containing a total of 32635 bounding boxes, 329 sequences of objects and a video length of 05 : 33 minutes in 30 FPS. It was also used in a lower frame rate condition, by collecting data from the dataset in a step of four frame gap, starting at the 2^{nd} frame. The dataset in this condition contains a total of 8434 bounding boxes, 321 sequences of objects and frame rate of 7.5 FPS. Moreover, ground truth will be used as detections, which will be favorable to the evaluation of each MOT method. Besides that, it is attainable an adequate form of comparison through the two MOT methods for conditions presented in the scope of this study. For the usage of the

Table 5.2: Evaluation of different weight conjugation, cost matrix computation, on the MOT17 training set (Detections acquired from FRCNN file). $T_{Lost} = 1$. $Thresh_{cost} = 0.3$. $hit_{min} = 3$.

Weights			Evaluation Metrics								
σ_{IoU}	σ_{Euc}	σ_{BBr}	MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓
5/10	4/10	1/10	44.93	87.84	56677	6223	54772	848	12.3%	33.0%	946
5/10	3/10	2/10	44.99	87.84	56713	6196	54757	827	12.3%	33.3%	932
4/10	3/10	3/10	44.85	87.76	56688	6323	54776	833	**13.0%**	**32.8%**	953
3/10	4/10	3/10	44.64	87.71	56613	6479	54832	852	12.8%	33.0%	948
3/10	3/10	4/10	44.58	87.70	56558	6492	54858	881	12.8%	33.0%	967
4/10	5/10	1/10	44.75	87.75	56627	6379	54793	877	12.6%	33.3%	961
6/10	3/10	1/10	45.25	87.90	**56803**	5984	54695	799	12.1%	33.0%	912
6/10	2/10	2/10	45.25	87.92	56801	5990	**54705**	791	12.1%	33.0%	907
7/10	**2/10**	**1/10**	**45.53**	**88.09**	56552	**5426**	54996	**749**	12.6%	33.5%	**853**

YOLOv3 object detector combined with MOT methods, the ISR Tracking Dataset was split in sequences of training and testing. The dataset was divided in sequences of 500 (ISR500 sub-dataset) and 200 (ISR200 sub-dataset) frames, respectively containing 20 and 50 sequences. Sequences are numbered from 1 to 20 in the ISR500 sub-dataset and 1 to 50 in the ISR200 sub-dataset. Furthermore, even numbered sequences integrate the Test split, and odd numbered sequences integrate the Training split.

5.2 SORT

The SORT algorithm has T_{Lost}, hit_{min} and $Thresh_{cost}$ constants that can be chosen. The main algorithm was configured with $T_{Lost} = 1$, $hit_{min} = 3$ and $Thresh_{cost} = 0.3$. Therefore, the following evaluation was primarily based on those values and without the usage of object detector algorithm.

5.2.1 *WMean* Cost Matrix Weights Selection

The different formulations of cost implemented in the SORT algorithm, were established using IoU (IoU), euclidean (ED) and bounding box ratio (BBr) distance metrics. Table 5.2 shows the evaluation of different weight conjugations (λ_{IoU}, λ_{ED} and λ_{BBr}) on the MOT17 Dataset for the $WMean$ cost matrix. As shown in Table 5.3, due to the lack of spatial information on the BBr distance metric, results of the BBr cost matrix are significantly unfavorable than using ED and IoU cost matrices. Based on the results of these three metrics, the best combination of weights to the weighted mean ($WMean$) cost matrix, was fond for values of $\lambda_{IoU} = \frac{7}{10}$, $\lambda_{ED} = \frac{2}{10}$ and $\lambda_{BBr} = \frac{1}{10}$ with MOTA score of 45.53. Aforementioned configuration has the less number of FP and IDs, despite the higher number of FNs. Furthermore, it has the less number of track FM by a large amount. Therefore, throughout the following evaluations of proposed data association techniques, $WMean$ cost matrix is configured with those aforementioned values.

Table 5.3: Evaluation of the SORT method and proposed data association techniques on the MOT17 training set (Detections acquired from FRCNN file). $T_{Lost} = 1$. $Thresh_{cost} = 0.3$. $hit_{min} = 3$.

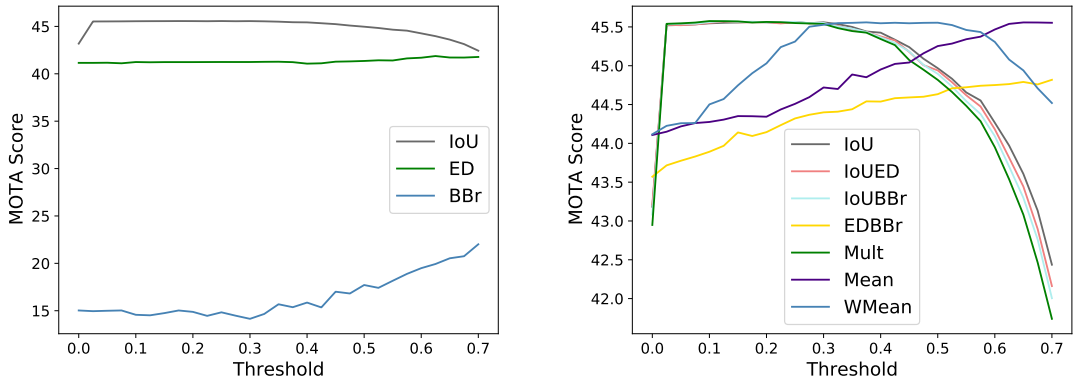
Cost Matrix	Evaluation Metrics									
	MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
IoU (SORT)	**45.56**	88.19	56298	5136	55281	718	11.5%	35.3%	798	**516**
ED	41.24	86.99	54292	7977	56271	1734	7.9%	33.7%	1915	500
BBr	14.15	83.06	37236	21351	70281	4780	4.9%	37.5%	4730	510
IoUED	45.55	88.20	56275	5126	55305	717	11.5%	35.3%	799	486
IoUBBr	45.55	**88.21**	56263	5111	55324	710	11.5%	35.5%	**797**	499
EDBBr	44.40	87.79	56329	6470	55028	940	11.7%	**32.4%**	1090	480
Mult	45.54	**88.21**	56245	**5107**	55344	**708**	11.5%	35.7%	**797**	469
Mean	44.72	87.72	**56636**	6417	**54811**	850	**13.0%**	33.0%	958	472
WMean	45.53	88.09	56552	5426	54996	749	12.6%	33.5%	853	473

Table 5.4: Best MOTA score of the SORT method and proposed data association techniques, on the MOT17 training set (Detections acquired from FRCNN file). $T_{Lost} = 1$. $hit_{min} = 3$.

Cost Matrix	$Thresh_{cost}$	Evaluation Metrics				
		MOTA↑	MOTP↑	% MT↑	% ML↓	Frag↓
IoU (SORT)	0.25	45.57	**88.16**	12.1%	34.8%	**808**
ED	0.625	41.87	87.17	8.1%	33.3%	1870
BBr	0.7	22.01	84.47	4.9%	36.1%	4077
IoUED	0.125	45.57	88.11	12.3%	33.9%	841
IoUBBr	0.125	45.58	88.11	12.3%	33.9%	838
EDBBr	0.7	44.82	87.97	12.1%	**33.0%**	1045
Mult	0.1	45.58	88.11	12.3%	33.7%	840
Mean	0.65	**45.59**	88.10	**12.5%**	33.7%	850
WMean	0.375	45.56	88.11	12.3%	33.9%	840

5.2.2 Evaluation on the MOT17 Dataset

All different formulations of cost matrices implemented, were evaluated on the MOT17 Dataset, with results shown in Table 5.3. While the SORT's *IoU* cost matrix formulation shows the best MOTA score, all the other evaluation metrics achieved a better performance on remaining cost matrices. The *Mult* cost matrix formulation has the lower number of FP, IDs and FM, which represents the most accurate tracking for sequences generated by the algorithm. On the other hand, the *Mean* cost matrix formulation has the higher number of TP and lower number of FN, which is proportional to the percentage of MT sequences. This represents a robust result when associating tracks, but has the downside of a lower MOTA score due to increased number of IDs and FP. In a general view, is observable an inverse proportionality between IDs and MOTP score, caused by the poor state representation of the KF when it gets shifted associations (compared to the ground truth) as observation or it is initialized as a new track. This evaluation shows positive results to the *IoU*, *IoUED*, *IoUBBr*, *Mult* and *Mean* cost matrices formulations and the fast frame processing of the method.



(a) Each cost matrix. **(b)** Combination of cost matrices.

Figure 5.2: SORT method and proposed data association techniques MOTA Scores for different values of $Thresh_{cost}$ threshold on the MOT17 Train Dataset (Detections acquired from FRCNN file).

Table 5.5: Evaluation of the SORT method for different number of maximum age (T_{Lost}). $Thresh_{cost} = 0.3$. $hit_{min} = 3$.

Dataset	T_{Lost}	Evaluation Metrics			
		MOTA↑	MOTP↑	% MT↑	% ML↓
ISR	1	90.57	92.21	60.5%	1.5%
Tracking	15	90.57	92.21	60.5%	1.5%
MOT17	1	45.56	88.19	11.5%	35.3%
Training	15	45.56	88.19	11.5%	35.3%

Concerning the fact that each cost matrix is computed based on different operations, each should have its respective threshold that can better identify the limit between bad and good associations. Fig. 5.2, shows the MOTA score fluctuating with $Thresh_{cost}$ in a range of $[0.0, 0.7]$. Fig. 5.2b shows a decline on MOTA score for threshold values higher than 0.3, excluding the $EDBBr$, $Mean$ and $WMean$ cost matrices formulations, where the increase of threshold value helps the track association to be more accurate. The $Mult$ cost matrix formulation curve, shows a similar behaviour to the IoU cost matrix formulation curve, were a decrease of MOTA score is expected for thresholds higher than 0.3, since the cost of each metric had to be higher or around $\sqrt[3]{0.3} \simeq 0.66$. Therefore, for $Thresh_{cost} = 0.3$, the $Mult$ cost matrix formulation discards tracks that have lower values in at least one of the three distance metrics. Fig. 5.2 is complemented with Table 5.4, where is shown a promising result for the $Mean$ cost matrix formulation, with the best MOTA score of 45.59 for $Thresh_{cost} = 0.65$.

Table 5.5 shows attained results of the SORT algorithm using different values of T_{Lost}. The method was performed on the MOT17 Train and ISR Tracking Datasets to distinguish differences when the variable T_{Lost} was increased from 1 to 15. Without any changes observed in the evaluation of both variables, is assumed that the method

Table 5.6: Evaluation of the SORT method and proposed data association techniques, with new track management conditions, on MOT17 Train Dataset (Detections acquired from FRCNN file). $T_{Lost} = 15$. $Thresh_{cost} = 0.3$. $hit_{min} = 3$.

Cost Matrix	Evaluation Metrics								
	MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓
IoU (SORT)	**45.37**	88.20	56048	5095	55533	716	10.8%	35.9%	787
ED	41.16	87.01	54130	7910	56472	1695	7.3%	34.1%	**1870**
BBr	14.98	83.12	37612	20788	69925	4760	4.2%	38.5%	4601
$IoUED$	**45.37**	88.21	56032	5085	55549	716	10.8%	35.9%	788
$IoUBBr$	45.36	**88.22**	56011	5070	55576	**710**	10.8%	36.1%	787
$EDBBr$	44.21	87.81	56077	6430	55284	936	10.6%	**33.2%**	1085
$Mult$	45.35	**88.22**	55997	**5066**	55592	708	10.8%	36.3%	786
$Mean$	44.48	87.74	**56338**	6391	**55087**	872	**12.1%**	33.7%	950
$WMean$	**45.37**	88.11	56323	5373	55231	743	11.9%	33.9%	837

is not capable of re-encounter a track when it is not associated for more than 1 frame. Consequently, it is assumed that the KF cannot accurately predict the state of bounding boxes without observations in a short period of time. Furthermore, due to camera motion on some sequences of the MOT17 Train Dataset and throughout the entire ISR Tracking Dataset, is understandable that the KF is incapable of modeling the state of bounding boxes, without any camera motion information. Therefore, the new track management formulation implemented on the SORT algorithm, has no affect on these results. In fact, the new track management formulation has a slight negatively affect in the performance of the tracker, once it is more intrusive in early deletion of tracks. Evaluation of this experience can be visualized at Table 5.6.

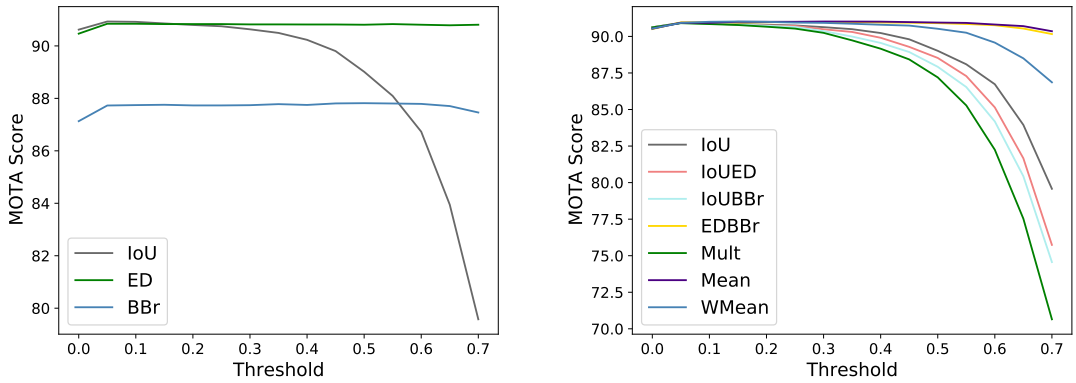
5.2.3 Class Gate Metric Evaluation on the ISR Tracking Dataset

The SORT class gate metric ($Cost^C$) was implemented to discard association of differentiated class objects. For that, due to the multi-class concept of the ISR Tracking Dataset, it is the one capable of evaluating such implementation. Moreover, it is achievable an evaluation of the tracking method in an indoor, multi-class, not crowded and lowered point of view scenario. Table 5.7 represents the evaluation of the SORT method and proposed data association techniques, in the ISR Tracking Dataset, with and without the usage of the $Cost^C$ class gate metric. Due to the usage of ground truth bounding boxes as measurements on the ISR Tracking Dataset, is observable an increase of MOTA and MOTP score in every evaluated data association technique. Furthermore, it is perceptible the effectiveness of the $Cost^C$ gate metric in such conditions, reaching the highest MOTA score of 91.02 and percentage of MT sequences of 69.3%, for the $Mean$ cost matrix formulation with $Cost^C$ gate metric. Excluding the usage of the class gate metric, $WMean$ cost matrix formulation shows the best MOTA score with 90.90. It is also noticeable the low number of FP of the $Mult$ cost matrix formulation, whilst for $Thresh_{cost} = 0.3$, it tend to easily reprove tracks that have at least one distance metrics with low value of cost.

To find the best $Thresh_{cost}$ for this conditions using the $Cost^C$ gate metric, Fig. 5.3

Table 5.7: Evaluation of SORT method and proposed data association techniques, on the ISR Tracking Dataset (Detections acquired from ground truth file). $T_{Lost} = 1$. $Thresh_{cost} = 0.3$. $hit_{min} = 3$.

Cost Matrix	Evaluation Metrics									
	MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
***IoU* (SORT)**	90.57	92.21	29589	31	2932	114	60.5%	1.5%	**556**	1368
$Cost^C$+*IoU*	90.63	92.20	29611	33	2926	98	61.7%	1.5%	**550**	1404
ED	88.85	91.48	29307	310	3031	297	59.0%	**0.9%**	618	**1389**
$Cost^C$+*ED*	90.82	92.00	29739	100	2830	66	66.9%	1.2%	566	**1408**
BBr	68.23	88.45	25015	2748	5550	2070	30.7%	3.6%	1179	1380
$Cost^C$+*BBr*	87.74	91.56	29134	500	3216	285	63.2%	1.2%	642	1425
IoUED	90.44	92.27	29538	22	2974	123	59.0%	1.5%	561	1317
$Cost^C$+*IoUED*	90.49	92.27	29553	22	2969	113	59.6%	1.5%	556	1337
IoUBBr	90.34	92.30	29491	9	3008	136	58.7%	1.8%	566	1377
$Cost^C$+*IoUBBr*	90.36	92.32	29497	8	3015	123	59.3%	1.8%	559	1392
EDBBr	90.77	92.00	29713	91	2827	95	65.0%	**0.9%**	562	1368
$Cost^C$+*EDBBr*	90.98	92.05	29767	77	2813	55	68.1%	**0.9%**	555	1375
Mult	90.21	**92.35**	29445	**5**	3053	137	57.4%	1.8%	564	1311
$Cost^C$+*Mult*	90.24	**92.36**	29456	**5**	3050	129	58.1%	1.8%	559	1307
Mean	90.87	91.96	**29756**	101	**2799**	**80**	**67.5%**	1.2%	558	1288
Cost*^{C}+*Mean	**91.02**	92.02	**29785**	81	**2799**	**51**	**69.3%**	1.2%	554	1292
WMean	**90.90**	92.10	29715	50	2832	88	64.4%	1.2%	558	1298
$Cost^C$+*WMean*	90.93	92.10	29727	53	2837	71	65.3%	1.2%	552	1305



(a) Each cost matrix. **(b)** Combination of cost matrices.

Figure 5.3: SORT method and proposed data association techniques MOTA Scores, for different values of $Thresh_{cost}$ using $Cost^C$ gate metric, on the ISR Tracking Dataset (Detections acquired from ground truth file).

shows the MOTA score of each metric throughout a range of thresholds, whilst Table 5.8 summarizes each best evaluation. In Fig. 5.3 is shown a decrease of MOTA values for higher thresholds. Moreover, *IoUBBr* and *Mean* cost matrices formulations curves, present a threshold independent behaviour. Table 5.8 exhibits the best MOTA score acquired from the range of different $Thresh_{cost}$ thresholds, for the SORT method and proposed data association techniques. Furthermore, it is noticeable that the *Mean* and

Table 5.8: Best MOTA score of the SORT method and proposed data association techniques, using $Cost^C$ gate metric, on the ISR Tracking Dataset (Detections acquired from ground truth file). $T_{Lost} = 1$. $hit_{min} = 3$.

Cost Matrix	$Thresh_{cost}$	Evaluation Metrics				
		MOTA↑	MOTP↑	% MT↑	% ML↓	FM↓
IoU	0.05	90.93	92.08	66.3%	1.2%	552
ED	0.05	90.84	91.99	67.8%	1.2%	565
BBr	0.5	87.82	91.60	62.9%	**0.9%**	639
$IoUED$	0.05	90.92	92.09	65.3%	1.2%	552
$IoUBBr$	0.05	90.93	92.08	66.3%	1.2%	**551**
$EDBBr$	0.25	90.99	92.04	68.4%	**0.9%**	555
$Mult$	0.05	90.91	**92.10**	65.0%	1.2%	**551**
$Mean$	0.3	**91.02**	92.02	**69.3%**	1.2%	554
$WMean$	0.15	**91.02**	92.02	**69.3%**	1.2%	554

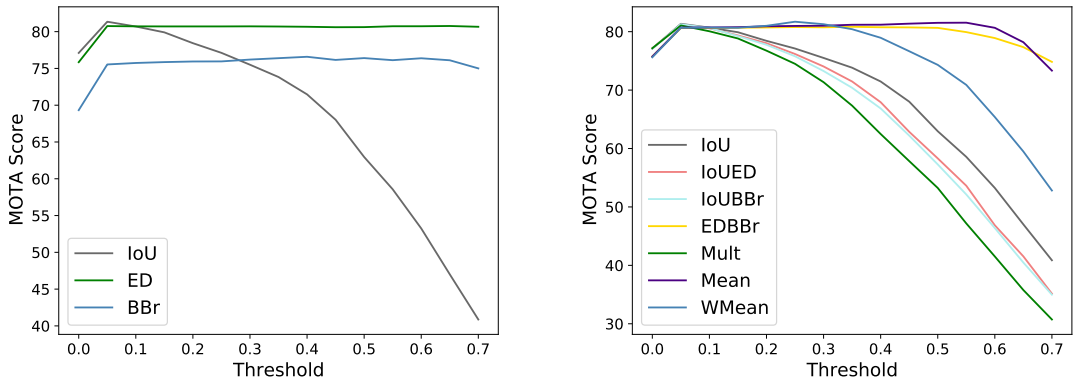
the $WMean$ cost matrices formulations outperformed remaining ones, achieving promising results comparing to the IoU cost matrix formulation.

5.2.4 Lower Frame rate Condition of the ISR Tracking Dataset

The ISR Tracking dataset lower frame rate condition was processed with the SORT algorithm acknowledging usefulness of the $Cost^C$ gate metric. The dataset was reduced by collecting frames in a four frame gap step. In this condition is expected an abrupt reduction of the MOTA score for cost matrices formulations that use the IoU distance metric, due to the spacial distance between bounding boxes in consecutive frames. In Table 5.9 is shown the results of the SORT and proposed data association techniques, using the $Cost^C$ gate metric. Is observable the decrease of MOTA score of IoU cost matrix formulation, and the outperform of ED and BBr cost matrices formulations. Despite BBr cost matrix formulation present flaws when used alone, in this experiment it achieved the second best result of MOTA score, compared to other single metric cost matrices formulations (IoU, ED). This conditions have to be tested with different $Thresh_{cost}$ to identify higher MOTA scores that could be shadowed by the threshold. Fig. 5.4 exhibits the MOTA scores throughout a range of thresholds ($Thresh_{cost} = [0, 0.7]$), with ED, BBr, $EDBBr$ and $Mean$ cost matrices formulations presenting a consistent behaviour. Each cost matrix formulation best MOTA score has their evaluations summarized at Table 5.10. Nonetheless, $WMean$ cost matrix formulation reaches the highest MOTA score with 81.71 for $Thresh_{cost} = 0.25$. Moreover, $Mean$ cost matrix formulation attain an optimistic MOTA score of 81.54, for a $Thresh_{cost} = 0.55$.

Table 5.9: Evaluation of proposed data association techniques, using $Cost^C$ gate metric, on the ISR Tracking Dataset (Gap = 4) (Detections acquired from ground truth file). $T_{Lost} = 1$. $Thresh_{cost} = 0.3$. $hit_{min} = 3$.

Cost Matrix	Evaluation Metrics									
	MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
IoU	75.65	86.99	6419	39	1886	129	23.1%	12.8%	82	1430
ED	80.63	84.29	7088	288	1308	38	**40.5%**	**6.2%**	51	1363
BBr	75.90	83.79	6831	430	1446	157	38.6%	6.5%	90	**1448**
IoUED	74.15	87.37	6268	14	2020	146	18.7%	14.3%	95	1376
IoUBBr	73.30	87.64	6193	11	2096	145	18.7%	15.0%	95	1403
EDBBr	80.63	84.48	7058	258	1339	**37**	38.3%	6.5%	54	1369
Mult	71.47	**88.05**	6030	**2**	2241	163	15.3%	15.6%	116	1354
Mean	80.97	84.41	**7093**	264	**1295**	46	40.2%	**6.2%**	47	1364
WMean	**81.16**	85.31	6932	87	1422	80	33.0%	8.7%	**42**	1356



(a) Each cost matrix. **(b)** Combination of cost matrices.

Figure 5.4: Proposed data association techniques MOTA Scores, for different values of $Thresh_{cost}$ using $Cost^C$ gate metric, on the ISR Tracking Dataset (Gap = 4) (Detections acquired from ground truth file).

Table 5.10: Best MOTA score of the SORT method and proposed data association techniques, using $Cost^C$ gate metric, on the ISR Tracking Dataset (Gap = 4) (Detections acquired from ground truth file). $T_{Lost} = 1$. $hit_{min} = 3$.

Cost Matrix	$Thresh_{cost}$	Evaluation Metrics				
		MOTA↑	MOTP↑	% MT↑	% ML↓	FM↓
IoU (SORT)	0.05	81.35	85.35	34.3%	8.4%	38
ED	0.65	80.77	84.62	**37.4%**	**6.2%**	46
BBr	0.4	76.58	84.01	36.4%	6.9%	89
IoUED	0.05	81.18	85.42	33.0%	8.4%	39
IoUBBr	0.05	81.27	85.43	34.0%	8.4%	**37**
EDBBr	0.35	80.84	84.73	**37.4%**	**6.2%**	49
Mult	0.05	81.06	**85.48**	33.0%	9.0%	38
Mean	0.55	81.54	85.14	33.3%	7.2%	42
WMean	0.25	**81.71**	84.98	34.3%	6.9%	42

Table 5.11: Evaluation of Deep-SORT method, for different values of λ and appearance feature association metric, on the MOT17 training set (Detections acquired from FRCNN file). $A_{max} = 30$. $dist^1_{max} = 0.2$. $dist^2_{max} = 0.7$. $hit_{min} = 3$.

Feature Association Metric	λ	Evaluation Metrics									
		MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
Cosine	0	**45.57**	88.26	**55689**	4513	**56161**	**447**	12.8%	**35.9%**	**655**	**13**
	0.3	45.23	88.32	55273	4484	56426	598	12.8%	36.1%	668	**13**
	0.5	45.23	88.32	55274	4486	56415	608	**13.0%**	36.1%	665	**13**
	0.7	45.13	**88.34**	55242	4561	56364	691	12.3%	36.1%	684	**13**
Euclidean	0	45.12	88.33	55113	4449	56559	625	11.7%	36.3%	664	**13**
	0.3	45.12	**88.34**	55108	**4436**	56571	618	11.7%	36.3%	659	**13**
	0.5	45.15	88.33	55148	4443	56538	611	12.1%	36.3%	660	**13**
	0.7	45.16	**88.34**	55182	4464	56502	613	12.5%	36.3%	658	**13**
	1	45.07	88.33	55242	4632	56322	733	11.9%	36.1%	697	**13**

5.3 Deep-SORT

The Deep-SORT algorithm is implemented with the possibility of selecting different constants and settings: The λ value for the weighted sum of object appearance feature and Mahalanobis distance metrics, the appearance feature distance metric (Euclidean or Cosine), association gating threshold for initial and second matching stage ($dist^1_{max}$ and $dist^2_{max}$), age of track without associations A_{max} and minimum of hits hit_{min} of a track to be considered as Confirmed. The main algorithm was configured with $\lambda = 0$, Cosine distance metric, $dist^1_{max} = 0.2$, $dist^2_{max} = 0.7$, $A_{max} = 30$ and $hit_{min} = 3$. Furthermore, it was used the appearance descriptor trained on a large-scale person re-identification dataset [48]. Therefore, in this experiments, was used the pre-trained network available online as appearance descriptor, and aforementioned constants as base configuration to evaluate different proposed data association techniques. Following evaluations on this section are attained without the usage of object detector algorithm.

5.3.1 λ Value and Feature Association

In order to obtain a solid performance of the Deep-SORT algorithm on the ISR Tracking Dataset, the λ values were ranged from 0 to 1 (with $\lambda = 1$ meaning no usage of feature association metric). Furthermore, in different occasions, the cosine and euclidean distance were applied, as appearance feature association metrics. Results of such experience in MOT17 training set are shown in Table 5.11. As expected, the method configured with $\lambda = 0$ and cosine distance as appearance feature association metric outperformed others. As reported in [1], $\lambda = 0$ is a reasonable choice when there is a notable camera motion, since the algorithm performs the first association stage metric only with appearance information. Moreover, since the appearance descriptor is trained using cosine metric learning, cosine distance is the adequate choice as feature association metric. Table 5.12 exhibits the evaluation of the aforementioned configurations indicated before, on the ISR Tracking

Table 5.12: Evaluation of Deep-SORT method, for different values of λ and appearance feature association metric, on the ISR Tracking Dataset (Detections acquired from ground truth file). $A_{max} = 30$. $dist^1_{max} = 0.2$. $dist^2_{max} = 0.7$. $hit_{min} = 3$.

Feature Association Metric	λ	Evaluation Metrics									
		MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
Cosine	0	**90.73**	89.65	**30974**	1366	**1439**	**222**	**71.4%**	**0.3%**	**147**	**152**
	0.3	87.35	**89.70**	29833	1328	2227	575	55.3%	0.6%	442	145
	0.5	86.97	**89.70**	29715	1332	2297	623	55.0%	0.6%	478	145
	0.7	86.76	**89.70**	29661	1347	2316	658	54.4%	0.6%	485	145
Euclidean	0	86.65	89.65	29539	**1262**	2478	618	53.8%	0.6%	533	144
	0.3	86.25	89.65	29427	1278	2525	683	53.2%	0.6%	549	142
	0.5	86.70	89.67	29571	1277	2432	632	53.5%	0.6%	510	143
	0.7	86.69	89.69	29604	1312	2380	651	53.8%	0.6%	500	143
	1	86.66	**89.70**	29635	1355	2323	677	53.8%	0.6%	484	143

Dataset. Reported results, manifest the same behaviour as in the MOT17 training set, moreover, it is detected a major increase of MT sequences and an impressive number of ML sequences of 0.3%, corresponding to only one sequence Mostly Lost. Despite the lower value of FPS on the MOT17 Dataset, this evaluation shows an optimistic result for not crowded scenarios.

5.3.2 First Stage Association Threshold Value

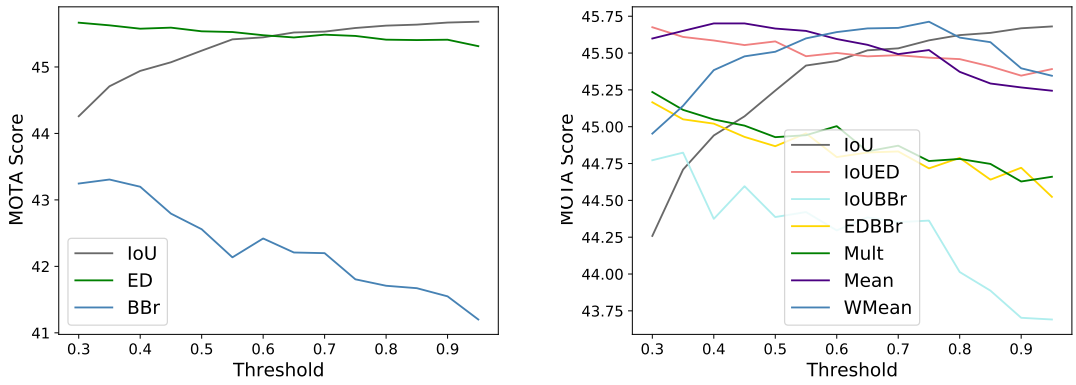
The $dist^1_{max}$ threshold was applied in different values to evaluate its influence on both datasets (Evaluation shown on Table 5.13). It is noticeable that the higher MOTA score has a different threshold for both datasets. Furthermore, the results are not capable of manifest any dependence with MOTP score, since the association is obtained based on the accuracy of detection and feature extraction, and trajectory modulation is achieved by the KF. $dist^1_{max} = 0.2$ is the configuration with lower number of FP on both datasets, which is an expected behaviour, since lower values of $dist^1_{max}$ are more critical to the feature association metric. This critical behaviour is also seen in the decreased number of TP and in the increased number of FN. However, $dist^1_{max} = 0.2$ configuration exhibits the higher value of IDs, which can be caused by the final stage association metric for objects that are not associated in the initial association stage. In results attained in the ISR Tracking dataset, configuration with $dist^1_{max} = 0.3$ outperforms remaining ones, despite the higher number of FP compared to $dist^1_{max} = 0.2$.

5.3.3 Second Stage Association Cost Matrix

The Deep-SORT second association stage was evaluated using different thresholds ($dist^2_{max}$) for different cost matrices formulations. As a backup association technique, it is expected to achieve similar MOTA results with minor changes on the number of TP, FP, FN and IDs. Fig. 5.5 shows the MOTA score variation for different cost matrices

Table 5.13: Evaluation of Deep-SORT different values of $dist^1_{max}$ on the MOT17 training (Detections acquired from FRCNN file) and on the ISR Tracking (Detections acquired from ground truth file) Datasets. $\lambda = 0$. $A_{max} = 30$. $dist^2_{max} = 0.7$. $hit_{min} = 3$. Appearance feature association: Cosine distance metric.

Dataset	$dist^1_{max}$	Evaluation Metrics									
		MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
MOT17 Training	0.2	45.57	**88.26**	55689	**4513**	56161	447	12.8%	35.9%	**655**	**13**
	0.3	45.94	88.17	56247	4659	55694	**356**	**15.8%**	34.2%	706	**13**
	0.4	**46.10**	88.12	**56492**	4722	**55423**	382	15.4%	**33.0%**	738	**13**
ISR Tracking	0.2	90.73	89.65	30974	**1366**	1439	222	71.4%	**0.3%**	147	152
	0.3	**91.24**	89.62	**31178**	1403	1272	**185**	**76.0%**	**0.3%**	**95**	**159**
	0.4	91.16	**89.67**	31176	1427	**1247**	212	75.4%	**0.3%**	97	156



(a) Each cost matrix.

(b) Combination of cost matrices.

Figure 5.5: Deep-SORT and proposed data association techniques MOTA Scores, for different values of $dist^2_{max}$ on the MOT17 training set (Detections acquired from FRCNN file). $\lambda = 0$. $A_{max} = 30$. $dist^1_{max} = 0.2$. $hit_{min} = 3$. Appearance feature association: Cosine distance metric.

formulations through a range of $dist^2_{max}$ values. In Table 5.14, is presented the best MOTA score with $dist^2_{max}$ value for the proposed data association techniques, with IoU, ED, $IoUED$, $Mean$ and $WMean$ cost matrices formulations outperforming remaining ones. It is noticeable the increase of the MOTA score, FP, IDs and FM, for the IoU cost matrix formulation when compared to the Table 5.13, Moreover, since the LAP is formulated for the minimization of cost, having an higher valued threshold is not the adequate choice. Therefore, ED, $IoUED$, $Mean$ and $WMean$ cost matrices formulations attained superior performance, whereas the percentage of MT sequences for each threshold exceed other evaluated proposals.

5.3.4 Class Gate Metric Evaluation

To use the Deep-SORT algorithm in a multi-class environment, a class gate metric was implemented to each stage association. This evaluation was proceeded in the ISR

Table 5.14: Best MOTA score of the Deep-SORT method and proposed data association techniques, on the MOT17 training set (Detections acquired from FRCNN file) $\lambda = 0$. $A_{max} = 30$. $dist^1_{max} = 0.2$. $hit_{min} = 3$. Appearance feature association: Cosine distance metric.

Cost Matrix	$dist^2_{max}$	Evaluation Metrics						
		MOTA↑	MOTP↑	FP↓	IDs↓	% MT↑	% ML↓	FM↓
IoU	0.95	45.68	88.21	4538	488	13.7%	34.8%	675
ED	0.3	45.67	88.18	4602	491	14.3%	34.6%	696
BBr	0.35	43.31	87.96	5649	946	11.0%	35.3%	1040
IoUED	0.3	45.68	88.18	4580	508	14.1%	**34.4%**	701
IoUBBr	0.35	44.82	88.17	5010	622	13.2%	35.2%	770
EDBBr	0.3	45.17	88.08	4863	582	13.2%	34.6%	778
Mult	0.3	45.24	88.10	4829	573	12.6%	34.6%	775
Mean	0.4	45.70	**88.22**	**4519**	**486**	14.3%	34.8%	**665**
WMean	0.75	**45.71**	88.20	4569	**486**	**14.5%**	34.8%	685

Table 5.15: Best MOTA score of the Deep-SORT method and proposed data association techniques using class gate metric, on the ISR Tracking Dataset (Detections acquired from ground truth file). $\lambda = 0$. $A_{max} = 30$. $dist^1_{max} = 0.2$. $hit_{min} = 3$. $dist^2_{max} = 0.7$. Appearance feature association: Cosine distance metric.

Cost Matrix	Evaluation Metrics									
	MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
IoU (Deep-SORT)	90.80	89.66	30989	1357	1447	199	72.3%	**0.3%**	142	163
ED	91.09	89.53	31100	1372	1367	168	76.3%	**0.3%**	131	167
BBr	89.12	**90.27**	30467	1384	1783	385	62.3%	**0.3%**	292	166
IoUED	91.15	89.52	**31124**	1376	**1350**	**161**	78.4%	**0.3%**	130	165
IoUBBr	90.90	89.86	30994	**1328**	1401	240	75.7%	**0.3%**	160	166
EDBBr	91.07	89.54	31087	1367	1381	167	76.3%	0.6%	134	**169**
Mult	91.15	89.55	31116	1370	1354	165	77.8%	0.6%	**125**	163
Mean	**91.23**	89.55	31123	1350	**1350**	162	**78.7%**	**0.3%**	126	168
WMean	91.09	89.56	31103	1376	1363	169	76.6%	**0.3%**	119	166

Tracking Dataset and achieved results using the $dist^2_{max} = 0.7$ are presented in Table 5.15. The decrease of FP and IDs support the benefit of using the class gate metric in multi-class environments. *IoUED* and *Mean* cost matrices formulations achieved the best performances, since they reached the 91 scored MOTA with more than 78% sequences MT. Besides the elevated value of FP acquired from *IoUED* cost matrix formulation, it attains the lower number of IDs. Furthermore, both *IoUED* and *Mean* cost matrices formulations, exhibit a comfortable value of FN and FM. Fig. 5.6 exhibits the variance of MOTA score for a range of $dist^2_{max}$ values, for the *IoU*, *IoUED* and *Mean* cost matrices formulations. Table 5.16 summarizes the best MOTA scored evaluations based on different values on $dist^2_{max}$, expressing the dominant performance of the *Mean* cost matrix formulation for a threshold of 0.65.

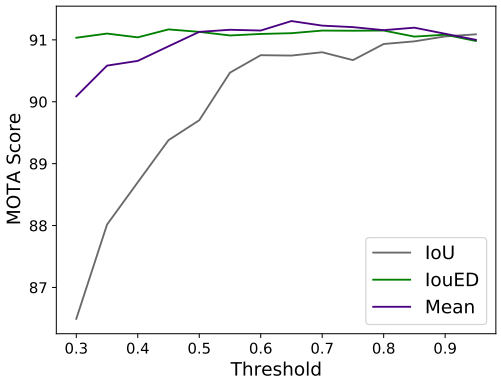


Figure 5.6: Proposed data association techniques MOTA Scores, for different values of $dist^2_{max}$ on the ISR Tracking Dataset (Detections acquired from ground truth file). $\lambda = 0$. $A_{max} = 30$. $dist^1_{max} = 0.2$. $hit_{min} = 3$. Appearance feature association: Cosine distance metric.

Table 5.16: Best MOTA score of *IoU*, *IoUED* and *Mean* cost matrices formulations, for different values of $dist^2_{max}$, on the ISR Tracking Dataset (Detections acquired from ground truth file). $\lambda = 0$. $A_{max} = 30$. $dist^1_{max} = 0.2$. $hit_{min} = 3$. Appearance feature association: Cosine distance metric.

Cost Matrix	$dist^2_{max}$	Evaluation Metrics						
		MOTA↑	MOTP↑	FP↓	IDs↓	% MT↑	% ML↓	FM↓
IoU	0.95	91.09	89.53	1367	164	76.6%	**0.3%**	129
IoUED	0.45	91.17	**89.54**	1376	161	77.5%	**0.3%**	127
Mean	0.65	**91.30**	**89.54**	**1358**	**150**	**79.3%**	**0.3%**	**124**

5.3.5 Best Deep-SORT Configuration

Based on results achieved in previous Deep-SORT and proposed data association techniques experiments, the *IoUED* and *Mean* cost matrices formulations for the second matching stage configurations, were chosen. Since the MOT17 dataset is a single class benchmark, the results using and not using the class gate metric, will be the same. Nevertheless, the selection of final stage association cost matrix and $dist^2_{max}$ was based on the higher MOTA score and percentage of MT sequences, with lower number of FP and IDs. Furthermore, results attained on the ISR Tracking dataset represented a substantial influence on this selection. Table 5.17 shows the contrast between the Deep-SORT method proposed data association techniques, on both datasets, including the ISR Tracking with four frame gap regime. *IoUED* and *Mean* cost matrices formulations have their respective $dist^2_{max}$ threshold, based on previous experiments performed on the MOT17 training and the ISR Tracking Datasets. *IoUED* cost matrix formulation evaluation was obtained for $dist^2_{max} = 0.3$ and *Mean* cost matrix formulation was evaluated for $dist^2_{max} = 0.5$. Furthermore, $\lambda = 0$, $A_{max} = 30$, cosine distance metric for appearance feature association, $hit_{min} = 3$ and $dist^1_{max} = 0.3$ configuration was used. The value of 0.3 for the $dist^1_{max}$ threshold, is based on the quality of the ISR Tracking Dataset ground truth bounding

Table 5.17: Evaluation of Deep-SORT, DSORT-CIoUED and DSORT-CMean methods on the ISR Tracking and MOT17 Datasets.

Dataset	Method	Evaluation Metrics									
		MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
MOT17 Training	Deep-SORT	45.57	**88.26**	55689	**4513**	56161	**447**	12.8%	35.9%	**655**	**13**
	DSORT-CIoUED	**46.13**	87.69	**57720**	5923	**54042**	535	16.1%	**31.5%**	891	**13**
	DSORT-CMean	46.07	87.71	57712	5972	54075	510	**16.3%**	31.9%	878	**13**
ISR Tracking	Deep-SORT	90.73	**89.65**	30974	**1366**	1439	222	71.4%	**0.3%**	147	152
	DSORT-CIoUED	**91.66**	89.56	**31310**	1397	**1199**	**126**	**81.8%**	**0.3%**	80	167
	DSORT-CMean	91.65	**89.65**	31292	1383	1203	140	80.5%	**0.3%**	**79**	**168**
ISR Tracking (Gap = 4)	Deep-SORT	69.30	**83.30**	6529	**684**	1719	186	30.8%	10.0%	108	148
	DSORT-CIoUED	74.85	80.95	7149	836	1196	89	48.9%	**2.5%**	58	**151**
	DSORT-CMean	**77.27**	81.49	**7231**	714	**1130**	**73**	**52.6%**	3.1%	**57**	**151**

boxes, implying substantial differences when comparing extracted appearance features (results reported on Table 5.13). For simplicity purposes, proposed data association technique with $IoUED$ cost matrix formulation and class gate metric, will be denominated as DSORT-CIoUED, while proposed data association technique with $Mean$ cost matrix formulation and class gate metric will be denominated as DSORT-CMean.

In Table 5.17 is presented the evaluation of proposed data association techniques and main Deep-SORT method. Each proposed data association technique achieve promising results in each dataset used. Nevertheless, for low frame rate conditions (ISR Tracking (Gap = 4)), DSORT-CMean outperform remaining methods. Since, neither Deep-SORT and DSORT-CIoUED have bounding box ratio associated in the second stage association cost matrix formulation. The bounding box ratio metric is exhibited as an important factor for the second stage association in lower frame conditions. Furthermore, as reported on section 5.2.4, in lower frame rate conditions, spacial distance between bounding boxes in consecutive frames, is increased, causing spacial distance based metrics to fail when using a lower $dist^2_{max}$.

5.4 YOLOv3 + MOT Method

To validate the tracking by detection end-to-end pipeline, YOLOv3 object detector to detect objects in raw images, and SORT and Deep-SORT different proposals of data association techniques, that achieved the best performances on previous evaluations, were used. The YOLOv3 + MOT evaluation was divided into two different arrangements on the ISR Tracking Dataset, described in Subsection 5.1.2: ISR200 sub-dataset and ISR500 sub-dataset. YOLOv3 network was trained using training sequences of each ISR sub-dataset. Related to the SORT algorithm, the $Mean$ and $WMean$ cost matrix formulation proposals, with cost matrix gate metric, were chosen. For simplicity, SORT method data association proposals will be denominated as SORT-CMean and SORT-CWmean. SORT-CMean is configured for $Thresh_{cost} = 0.6$ and SORT-CWmean is configured for $Thresh_{cost} = 0.4$. Furthermore, Deep-SORT related data association proposals were the

ones chosen in section 5.3.5, denominated as DSORT-CIoUED and DSORT-CMean. To equally compare different data association technique proposals with the main methods, the SORT and the Deep-SORT were also evaluated with class gate metric.

Observing Table 5.18, the SORT, the SORT-CMean and the SORT-CWMean, outperformed other data association techniques, on MOTA score. Based on previous evaluations (where Deep-SORT related data association techniques outperformed SORT related data association techniques, in the ISR Tracking dataset), this experiment results, using the object detector algorithm, were caused by two factors: (1) Dividing the ISR Dataset into sequences of 200 or 500 images, may not be enough to properly represent the object conditions to train the YOLOv3 network in an efficient way, which can led to some miss-classifications or miss-detected bounding boxes. This is also seen when comparing results from each sub-dataset, where evaluations on the ISR200 sub-dataset outperformed those done on the ISR500 sub-dataset; (2) Using an object detector method instead of the dataset labels may introduce additional errors to the overall pipeline, since the tracking method can receive as input incorrect bounding boxes. Besides the higher MOTA score on SORT related data association techniques, a substantial increase of MT sequences is observable, indicating higher capacity of tracking sequences using Deep-SORT related data association techniques. On the other hand, the Deep-SORT related data association techniques, reached higher values of TP, which is directly related to the higher number of MT sequences. Deep-SORT related data association techniques, achieved more FP results, with minimum differentiation on IDs values, indicating more objects identified compared to SORT related data association techniques. This shows the capacity of Deep-SORT related data association techniques, to easily associate bounding boxes, which is caused by the continuous attempt to associate objects, after a track is marked as confirmed. In other words, a Deep-SORT related data association technique, after initiating and confirming a track (object got three consecutive matches), the track does not need another three hits to be identified as a possible track to be outputted by the method, instead, when it is associated, the output conditions integrate it on the output array in that frame. Comparing to SORT related data association techniques, where, despite the same need of three consecutive matches to be confirmed as a track, two miss association forces a track to be deleted, requiring a new track initialization with three consecutive associations and a new tracking ID. Each aforementioned track management description, imposes the SORT related data association techniques to be more critical to tracking tasks, despite their association algorithm being less critical. Furthermore, since the computation of TP, FP or FN is based on the Intersection over Union, which determines the overlap between the tracked bounding box and the ground truth, errors in detections and in KF state reproduction, cause tracked bounding boxes fail to be associated to the ground truth, producing the increase of FPs. Moreover, since Deep-SORT related data association techniques are capable of re-identifying a lost sequence, the KF is obligated to produce a state after some considerable frames without measurements, increasing state errors and consequently

Table 5.18: Evaluation of SORT, SORT-CMean, SORT-CWmean, Deep-SORT, DSORT-CIoUED and DSORT-CMean MOT data association techniques using YOLOv3 object detector, only accepting detections with confidence over $th_{conf} = 0.5$, on the ISR200 and the ISR500 Datasets.

Dataset	Tracking Method	Evaluation Metrics									
		MOTA↑	MOTP↑	TP↑	FP↓	FN↓	IDs↓	% MT↑	% ML↓	FM↓	FPS↑
ISR200	SORT	**64.33**	83.03	11881	**1719**	3770	**145**	45.3%	18.2%	142	**52**
	SORT-CMean	63.71	**83.04**	11828	1765	3782	186	42.2%	18.2%	162	**52**
	SORT-CWmean	63.78	83.03	11827	1753	3783	186	42.2%	18.2%	162	**52**
	Deep-SORT	61.07	81.38	12077	2430	3526	193	46.1%	17.1%	129	32
	DSORT-CIoUED	60.86	81.21	12188	2575	3431	177	48.8%	**16.7%**	125	33
	DSORT-CMean	61.16	81.30	**12202**	2541	**3430**	164	**50.0%**	**16.7%**	**123**	32
ISR500	SORT	**50.49**	79.95	9123	**1636**	5478	**228**	30.9%	24.0%	**238**	**53**
	SORT-CMean	49.65	79.94	9038	1676	5509	282	32.6%	24.0%	261	**53**
	SORT-CWmean	49.57	**79.99**	9034	1684	5517	278	30.9%	24.0%	264	**53**
	Deep-SORT	47.18	78.86	9561	2564	4993	275	33.1%	21.7%	266	33
	DSORT-CIoUED	46.33	78.68	9812	2942	4748	269	37.7%	**21.1%**	266	34
	DSORT-CMean	46.67	78.65	**9838**	2918	**4734**	257	**38.9%**	**21.1%**	269	34

FPs. Furthermore, sequence related metrics are outperformed by DSORT-CIoUED and DSORT-CMean configurations, with higher values of MT sequences, lower value of ML sequences, and lower values of FM, in the ISR200 sub-dataset.

The best arrangement of the ISR Tracking dataset, to evaluate tracking algorithms, is the ISR200 sub-dataset, where the ISR Tracking dataset was divided into more 30 sequences, providing more comparable object appearances between training and test sequences. There are some important concerns that must be taken into account, when choosing the best performance achieved in this experiment: a FP being the miss match of returned tracks in each frame; the critical behaviour to track sequences of SORT related data association techniques causing the increase of FN; the performance of the object detector in each sub-dataset. Observing Table 5.18, the SORT data association technique outperform others, by achieving the higher MOTA score, with less FP ans IDs, and best value of FPS. Nonetheless, DSORT-CMean technique, outperformed other techniques in number of TP, FN, percentage of MT and ML, and FM (in ISR200 sub-dataset), attaining the best performance on tracking ground truth sequences with the same tracking ID, with less FM. Moreover, the capacity of Deep-SORT related association techniques to re-identify and associate lost tracks, is crucial, indicating a promising usability in indoor multi-object tracking tasks.

6

Conclusion

This dissertation presented a study and an exploitation on Multi-Object Tracking by Detection algorithms (the SORT, the Deep-SORT, and proposed data association techniques), having in view indoor mobile robot applications, where the performance of associating measurements to existing tracks and inference speed are crucial aspects.

The SORT algorithm was enhanced by novel Linear Assignment Problem (LAP) formulation for the Hungarian Algorithm solver, using arrangements of different similarity metrics, such as the Intersection over Union (IoU), euclidean distance, and bounding box ratio. Also, a gate metric based on object classes, that disallowed measurements and tracks from different classes to be associated, was implemented. Moreover, in lower frame rate conditions, where objects bounding boxes differ in distance between frames, bounding box ratio metric was discovered to be successful, despite the metric compromises the association algorithm for not addressing spacial properties into the association. The IoU, the euclidean distance and the bounding box ratio metrics, showed promising results when assembled together, increasing the robustness of associating measurements to tracks. The SORT-CWMean data association technique, outperform other SORT related proposed data association techniques in most of experiments performed on this dissertation, showing the importance of the IoU distance conjugated with euclidean distance and bounding box ratio metrics.

The Deep-SORT was also enhanced with different approaches for the association problem. The same LAP formulations implemented in the SORT algorithm, were implemented for the second matching metric of the Deep-SORT method. Once again, an increase of performance was observed, when using cost formulation not only based on IoU metric. Also, a class gate metric was implemented, to disallow possible associations of objects with different classes. In most of experiments performed on this dissertation, the DSORT-CMean data association technique achieved better MOTA score compared to other Deep-SORT data association techniques. Moreover, it showed the importance of conjugating different association metrics when formulating the LAP. Also, when detection appearance features are dubious, the usage of alternative LAP formulations showed the importance to have a reliable backup association algorithm in the Deep-SORT method.

An end-to-end evaluation of a tracking by detection method was performed. Using the YOLOv3 object detector to attain measurements for the MOT algorithms, an evaluation on the ISR Tracking dataset was performed. Results achieved in this condition showed the importance of having a proper dataset, for adequate training of neural networks. Moreover, the importance of having a reliable object detector to perform tracking tasks, was also shown. Nonetheless, the SORT, the Deep-SORT and proposed association techniques evaluated on this condition, revealed a promising result of tracking ground truth sequences. Furthermore, a proposed data association technique, the DSORT-CMean, achieved the most ground truth sequences tracked with a single tracking id, and the less ground truth sequences without tracking.

Promising results were achieved by the SORT, the Deep-SORT and proposed data association techniques, performing experiments with promising inference speed. The capability of Deep-SORT related data association techniques, to easily re-identify lost tracks, show an importance of such algorithms to the motivation of this dissertation.

6.1 Future Work

This dissertation presented the potential of using YOLOv3 with DSORT-CMean framework for autonomous indoor mobile robot platforms. Moreover, the need of a reliable indoor dataset (ISR Tracking dataset) is also fundamental for such appliances. However, improvements can be made, to further improve the performance of such frameworks.

ISR Tracking Dataset

Improve the ISR Tracking Dataset with more labels and sequences.

Different Appearance Descriptor Network

Study different appearance descriptor networks for the appearance feature extraction in the Deep-SORT method. Train and apply the network using different re-identification datasets and the ISR Tracking dataset.

Camera Motion Information on Kalman Filter

Introduce camera motion information to the KF algorithm, improving the accuracy of predictions and consequentially the overall method.

Scene information

Use MOT methods to identifying dynamic or static objects.

3D Representation

Use MOT methods and other sensor information to generate a 3D representation of the scene.

Predictive Module

Add a predictive module that uses object states to accurately predict future states of objects. This strategy has benefits on occluded object tracking and also, can estimate the dynamical behaviour of objects and be merged with navigation units, to calculate more accurate paths, based on predictive scenes.

Bibliography

[1] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple Online and Realtime Tracking with a Deep Association Metric. In *IEEE International Conference on Image Processing (ICIP)*, 2017.

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE International Conference on Image Processing (ICIP)*, 2016.

[3] G. A. Zachiotis, G. Andrikopoulos, R. Gornez, K. Nakamura, and G. Nikolakopoulos. A Survey on the Application Trends of Home Service Robotics. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2018.

[4] Tamás Haidegger, Marcos Barreto, Paulo Gonçalves, Maki K. Habib, Sampath Kumar Veera Ragavan, Howard Li, Alberto Vaccarella, Roberta Perrone, and Edson Prestes. Applied Ontologies and Standards for Service Robots. In *Robotics and Autonomous Systems*, volume 61, pages 1215–1223, 2013.

[5] Félix Ingrand and Malik Ghallab. Deliberation for autonomous robots: A survey. In *Artificial Intelligence*, volume 247, pages 10–44, 2017.

[6] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. Deep learning in video multi-object tracking: A survey. In *Neurocomputing*, volume 381, pages 61–88, 2020.

[7] A. Gautam and S. Singh. Trends in Video Object Tracking in Surveillance: A Survey. In *Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2019.

[8] R. Kamal, A. J. Chemmanam, B. A. Jose, S. Mathews, and E. Varghese. Construction Safety Surveillance Using Machine Learning. In *International Symposium on Networks, Computers and Communications (ISNCC)*, 2020.

[9] D. J. R. Del Carmen and R. D. Cajote. Assessment of Vision-Based Vehicle Tracking for Traffic Monitoring Applications. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018.

[10] M. Yigitsoy, V. Belagiannis, A. Djurka, A. Katouzian, S. Ilic, E. Pernuš, A. Eslami, and N. Navab. Random ferns for multiple target tracking in microscopic retina image sequences. In *IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015.

[11] X. Li, R. Liu, M. Li, Y. Liu, L. Jiang, and C. Zhou. Real-Time Polyp Detection for Colonoscopy Video on CPU. In *IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2020.

[12] Shih-Yun Lo, Katsu Yamane, and Ken-ichiro Sugiyama. Perception of Pedestrian Avoidance Strategies of a Self-Balancing Mobile Robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[13] Md Jahidul Islam, Jungseok Hong, and Junaed Sattar. Person-following by autonomous robots: A categorical overview. In *The International Journal of Robotics Research*, volume 38, pages 1581–1618, 2019.

[14] Bing Shuai, Andrew G. Berneshawi, Davide Modolo, and Joseph Tighe. Multi-Object Tracking with Siamese Track-RCNN. In *arXiv*, 2020.

[15] D. Reid. An algorithm for tracking multiple targets. In *IEEE Transactions on Automatic Control*, volume 24, pages 843–854, 1979.

[16] Qiankun Liu, Bin Liu, Yue Wu, Weihai Li, and Nenghai Yu. Real-Time Online Multi-Object Tracking in Compressed Domain. In *IEEE Access*, volume 7, pages 76489–76499, 2019.

[17] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *arXiv*, 2015.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. In *arXiv*, 2018.

[20] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification. In *arXiv*, 2018.

[21] Xueqin Zhang, Xiaoxiao Wang, and Chunhua Gu. Online multi-object tracking with pedestrian re-identification and occlusion processing. In *The Visual Computer*, volume 37, pages 1089–1099, 2021.

[22] Jiating Jin, Xingwei Li, Xinlong Li, and Shaojie Guan. Online Multi-object Tracking with Siamese Network and Optical Flow. In *IEEE 5th International Conference on Image, Vision and Computing (ICIVC)*, 2020.

[23] Ryo Matsumura, Yukiyasu Domae, Weiwei Wan, and Kensuke Harada. Learning Based Robotic Bin-picking for Potentially Tangled Objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[24] D. N. Thang, L. A. Nguyen, P. T. Dung, T. D. Khoa, N. H. Son, N. T. Hiep, P. Van Nguyen, V. D. Truong, D. H. Toan, N. M. Hung, T. Ngo, and X. Truong. Deep Learning-based Multiple Objects Detection and Tracking System for Socially Aware Mobile Robot Navigation Framework. In *5th NAFOSTED Conference on Information and Computer Science (NICS)*, 2018.

[25] Ricardo Pereira, Tiago Barros, Luís Garrote, Ana Lopes, and Urbano J. Nunes. An Experimental Study of the Accuracy vs Inference Speed of RGB-D Object Recognition in Mobile Robotics. In *29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020.

[26] Ricardo Pereira, Luís Garrote, Tiago Barros, Ana Lopes, and Urbano J. Nunes. A Deep Learning-based Indoor Scene Classification Approach Enhanced with Inter-Object Distance Semantic Features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[27] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. In *arXiv*, 2016.

[28] Wenhan Luo, Xiaowei Zhao, and Tae-Kyun Kim. Multiple Object Tracking: A Review. In *arXiv*, 2014.

[29] Sangyun Lee and Euntai Kim. Multiple Object Tracking via Feature Pyramid Siamese Networks. In *IEEE Access*, volume 7, pages 8181–8194, 2019.

[30] Jiawei He, Zehao Huang, Naiyan Wang, and Zhaoxiang Zhang. Learnable Graph Matching: Incorporating Graph Partitioning with Deep Feature Learning for Multiple Object Tracking. In *arXiv*, 2021.

[31] Bing Shuai, Andrew G. Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe. SiamMOT: Siamese Multi-Object Tracking. In *arXiv*, 2021.

[32] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *arXiv*, 2019.

[33] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards Real-Time Multi-Object Tracking. In *arXiv*, 2019.

[34] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking Objects as Points. In *arXiv*, 2020.

[35] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. A Simple Baseline for Multi-Object Tracking. In *arXiv*, 2020.

[36] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and ReID in Multi-Object Tracking. In *arXiv*, 2020.

[37] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to Detect and Segment: An Online Multi-Object Tracker. In *arXiv*, 2021.

[38] H. W. Kuhn. The Hungarian method for the assignment problem. In *Naval Research Logistics Quarterly*, volume 2, pages 83–97, 1955.

[39] Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. Siamese network features for image matching. In *23rd International Conference on Pattern Recognition (ICPR)*, 2016.

[40] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature Pyramid Networks for Object Detection. In *arXiv*, 2017.

[41] Bruce Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of DARPA Image Understanding Workshop (IUW '81)*, 1981.

[42] Bharti Munjal, Abdul Rafey Aftab, Sikandar Amin, Meltem D. Brandlmaier, Federico Tombari, and Fabio Galasso. Joint Detection and Tracking in Videos with Identification Features. In *arXiv*, 2020.

[43] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as Points. In *arXiv*, 2019.

[44] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. MOTS: Multi-Object Tracking and Segmentation. In *CVPR*, 2019.

[45] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. In *Journal of Basic Engineering*, volume 82, pages 35–45, 1960.

[46] L. Jetto, S. Longhi, and G. Venturini. Development and Experimental Validation of an Adaptive Extended Kalman filter for the Localization of Mobile Robots. In *IEEE Transactions on Robotics and Automation*, volume 15, pages 219–229, 1999.

[47] Prasanta Chandra Mahalanobis. On the Generalised Distance in Statistics. In *Proceedings of the National Institute of Sciences of India*, volume 2, pages 49–55, 1936.

[48] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. MARS: A Video Benchmark for Large-Scale Person Re-identification. In *Computer Vision - ECCV*, 2016.

[49] Nicolai Wojke and Alex Bewley. Deep Cosine Metric Learning for Person Re-identification. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[50] Josh Patterson and Adam Gibson. *Deep Learning: A Practitioner's Approach*. O'Reilly, Beijing, 2017.

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. In *International Journal of Computer Vision (IJCV)*, volume 115, pages 211–252, 2015.

[52] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu. Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey. In *IEEE Signal Processing Magazine*, volume 35, pages 84–100, 2018.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *arXiv*, 2015.

[54] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *arXiv*, 2014.

[55] R. Cruz, L. Garrote, A. Lopes, and U. J. Nunes. Modular Software Architecture for Human-Robot Interaction applied to the InterBot Mobile Robot. In *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2018.

[56] F. Yang, W. Choi, and Y. Lin. Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[57] Junjie Yan, Zhen Lei, Longyin Wen, and Stan Z. Li. The Fastest Deformable Part Model for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.