FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Automotive Interior Sensing - Temporal Consistent Human Body Pose Estimation

José Martinho Oliveira Peres

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor (FEUP): Prof. Jaime Cardoso

Supervisor (Bosch): Eng.º Joaquim Fonseca

July 29, 2020

Resumo

Com o surgimento e desenvolvimento de veículos autónomos, foi igualmente criada uma necessidade de monitorizar e identificar objetos e ações que ocorrem no ambiente que rodeia o veículo. Este tipo de monitorização é particularmente importante no caso de veículos partilhados, dada a necessidade de identificar ações não só no exterior mas também no interior do veículo devido à ausência de um condutor humano que possa detetar, por exemplo, potenciais ações de violência entre passageiros e/ou situações onde estes necessitem de assistência.

Englobado neste contexto, a Bosch desenvolveu uma solução de estimação de postura humana com o objetivo de extrapolar a pose de todos os ocupantes presentes numa dada imagem, inferir o comportamento de cada passageiro e, consequentemente, identificar ações potencialmente maliciosas. Porém, para que este algoritmo possa ser aplicado não apenas a imagens isoladas mas também a vídeos é necessário adicionar contexto temporal entre frames. Por outras palavras, é necessário associar a estimação de pose de uma dada pessoa para uma dada frame às estimações de pose para a mesma pessoa em frames subsequentes de modo a que a identificação dessa pessoa (ou qualquer outra presente numa dada frame) ao longo do vídeo seja correta e consistente.

O tópico de associação temporal, também conhecido como "pose tracking", é abordado e desenvolvido ao longo do presente projeto, culminando na proposta e implementação de uma solução que melhora consideravelmente a consistência temporal do algoritmo de estimação de pose humana da Bosch. A solução desenvolvida utiliza uma mistura de abordagens clássicas e atuais de associação de informação, como por exemplo o "Hungarian algorithm", e abordagens de lógica de informação desenvolvidas especificamente para o caso em questão. A performance do algoritmo implementado no presente projeto é avaliada usando duas das mais recorrentes métricas de avaliação em casos de rastreamento de pose.

Palavras-chave: Autónomo, Estimação, Interior do veículo, Postura, Rastreamento

Abstract

With the emergence and development of autonomous vehicles, a necessity to constantly monitor and identify objects and action that occur in the surrounding environment of the vehicle itself was also created. This type of monitoring is particularly important in the case of shared vehicles, given the necessity to identify actions not only in the exterior but also in the interior of the vehicle due to the absence of a human driver that can detect, for instance, potential violent actions between passengers and/or cases where assistance is required.

Encompassed in this context, Bosch has developed a human body pose estimation solution in order to extrapolate the pose of all vehicle occupants present in a given image, infer the behaviour of each passenger and, consequently, identify potentially malicious actions. However, in order to apply this algorithm not only to isolated images but also to videos it is necessary to add temporal context between frames. In other words, an association is required between the body pose estimation for a given person in a given frame and the body pose estimations for the same person in subsequent frames in order to ensure that the identification of that passenger (or any other passenger present in the same frame) is accurate and consistent throughout the entire video.

The temporal association topic, also known as pose tracking, is addressed and developed during the present project, culminating in the proposal and implementation of a solution that considerably improves the temporal consistency of the human body pose estimation algorithm developed by Bosch. The implemented solution uses a mixture of currently relevant classical approaches for data association, such as the Hungarian algorithm, and approaches based on data logic developed specifically for the present case. Regarding performance, the developed algorithm is evaluated using two of the most recurrent metrics for pose tracking methods.

Keywords: Autonomous, Estimation, In-vehicle, Pose, Tracking

Agradecimentos

Em primeiro lugar, gostaria de agradecer tanto à empresa Bosch como à Faculdade de Engenharia da Universidade do Porto a oportunidade única de desenvolver a minha dissertação em ambiente empresarial. Adicionalmente, gostaria de expressar a minha mais sincera gratidão a todas as pessoas, especialmente colegas de trabalho, amigos e família, que me auxiliaram e apoiaram durante esta jornada.

Um especial agradecimento ao Prof. Jaime Cardoso e ao Eng.º Joaquim Fonseca pela oportunidade e excelente orientação; ao Eng.º Marco Prata pela indispensável ajuda e paciência; aos restantes membros da equipa de "In-Vehicle Sensing" da Bosch pela disponibilidade, motivação e ajuda no processo de integração; ao Pedro Augusto pelos conselhos e partilhas e aos meus pais pela motivação e apoio incondicional que demonstraram ao longo de todos estes anos, fulcral para poder seguir os meus sonhos.

Um sentido e sincero muito obrigado,

Martinho Peres

"No star is ever lost we once have seen,
we always may be what we might have been."

Adelaide Anne Procter

Contents

List of Figures

List of Tables

Acronyms and abbreviations

CNNs	Convolutional Neural Networks
CPU	Central Processing Unit
CVPR	Computer Vision and Pattern Recognition
GPU	Graphics Processing Unit
GRUs	Gated Recurrent Units
GT	Ground Truth
HPE	Human Pose Estimation
HRNet	High-Resolution Network
IoU	Intersection-over-Union
JPDAF	Joint Probabilistic Data Association Filters
LSTM	Long-Short Term Memory
mAP	mean Average Precision
MHT	Multi-Hypothesis Tracking
ML	Mostly Lost trajectories
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
MSCOCO	Microsoft Common Objects in Context
MT	Mostly Tracked trajectories
MTT	Multi-Target Tracking
NMS	Non-Maximum Suppression
OKS	Object Keypoint Similarity
PAFs	Part Affinity Fields
PCK	Percentage of Correct Keypoints
PCP	Percentage of Correctly estimated body Parts
RCNNs	Recurrent convolutional Neural Networks
SOTA	State-Of-The-Art
SVMs	Support-Vector Machines
TAFs	Temporal Affinity Fields
TFFs	Temporal Flow Fields
TOKS	Temporal OKS
VGG	Visual Geometry Group

Chapter 1

Introduction

1.1 Context

With the emergence and development of autonomous vehicles, a necessity to constantly monitor and identify objects and actions that occur in the surrounding environment of the vehicle itself was also created. In the case of shared autonomous vehicles, this monitoring process is also applied to the interior of the vehicle, which plays an important role in the surveillance of the behaviour of its passengers, given the lack of human intervention not only in the driving aspect but also in the management of potential malicious actions performed by the passengers [17]. Such behaviours can be detected or inferred through information obtained using body pose estimation algorithms. In most cases, this type of algorithms uses deep learning techniques, which provide a way to estimate the pose of the passengers through processing of images captured inside of the vehicle by using neural networks [17].

1.1.1 Human pose estimation

Encompassed in the field of computer vision, human pose estimation (HPE) can be defined as the task responsible for the identification/localisation of keypoints (figure 1.1) representative of human body joints (for example wrists, elbows, shoulders or knees) in a single image or video and subsequent estimation of the pose resulting from the spatial alignment of those keypoints [18].

The information provided by these techniques may then be used in a wide array of applications, such as human-robot interactions, virtual and augmented realities, sport analysis and video surveillance [18], making HPE an influential and important area of research and development in the field of computer vision. Despite of the ever-growing progress in this area, there are still challenges that prevent state-of-the-art (SOTA) methods from achieving optimal results [18, 19]. Examples of these constraints are: necessity to capture the context, variability in human physical appearance and in background features, occlusion of keypoints due to overlapping, structural complexity and information loss from 2D to 3D conversion [19, 20].

Regarding classification, HPE problems can be divided into distinct categories depending on the several features/factors of the estimation process. For instance, an important aspect that needs

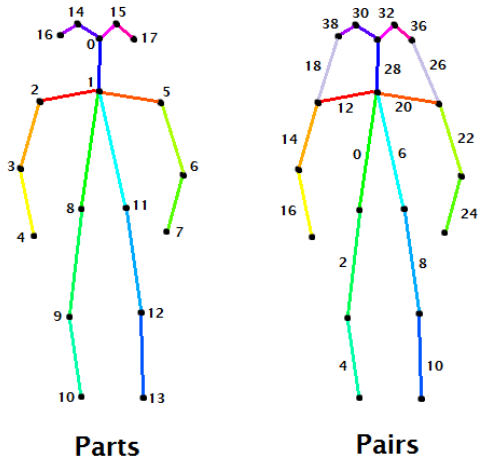

Figure 1.1: Representation of the keypoints that comprise the human pose skeleton from Microsoft Common Objects in Context dataset [1].

to be taken into consideration when performing pose estimation in an image is the number of people that are present in it. In this case, HPE can be either classified as single-person or multi-person pose estimation, with the latter being the most complex and demanding of the two processes due not only to the additional challenge of identifying multiple people and correctly matching them to their respective keypoints, but also to the possibility of the occurrence of inter-person occlusion of keypoints. Another important aspect is the dimension of the estimation output, which allows the classification of the estimation method into either 2D, which outputs X and Y coordinates for each keypoint detected, or 3D HPE, which adds a third coordinate (Z) to each keypoint in order to provide a three-dimensional prediction of the pose of each person represented in the image. Once again, the latter category (3D HPE) is harder to implement due to, among other problems, an inferior number of datasets (when comparing with the 2D alternative) for training of purely based 3D methods and the presence of spatial ambiguities when converting 2D poses into 3D equivalents [18]. Besides these two main classification factors, HPE can also be classified taking into account other aspects such as: input format (with RGB and Time of Flight being the most popular) or number of frames (single-frame or multi-frame input).

1.1.2 Multi-person tracking

The process of tracking multiple individuals in a video is comprised within the computer vision task of multi-target tracking (MTT) [5]. MTT is responsible for the detection and tracking of multiple objects (for instance vehicles) and/or humans present in a given video [5]. Similarly to pose estimation, this task can be applied to a vast set of fields such as video surveillance, action recognition and, as the objective of this dissertation, autonomous driving [5]. Regarding classification, MTT can be divided into two categories: 1) online methods, which provide an estimation of the movement of each individual based only on information from current and past

frames of a given video and 2) batch (or offline) methods, whose results for a given sequence are based on information provided by past, current and future (occur after the current sequence) frames [5, 21]. Whilst batch approaches often provide more accurate and temporal consistent results due to the access to a more complete set of information, which comprises data from frames subsequent to the one being analysed, this type of approach is not compatible with real-time due to the necessity of using information only available after the real-time event occurs [5, 21]. In contrast, online approaches are suitable for real-time tracking but their results are less consistent than the ones obtained through batch tracking [5, 21].

1.2 Objectives

Despite of the clear advantages that HPE algorithms provide, one limitation of this kind of approach is their lack of temporal consistency due to the processing of video footage occurring in a frame-by-frame basis. Given this limitation, it was proposed, within the scope of the dissertation, the development and implementation of an algorithm, auxiliary to the pose estimation process, capable of mitigating the effects of inconsistency currently observed, with the objective of improving the performance of the SOTA body pose estimation method developed by Bosch. Although this is the main objective of the present dissertation, it is also necessary to perform a thorough review of SOTA literature as well as a familiarisation with the currently relevant solutions used for temporal consistency in body pose estimation methods. Only with the combined knowledge gathered from the aforementioned tasks, it will be possible to fully understand the problems at hand, and develop a method, based on already implemented approaches, capable of solving it.

1.3 Contributions

The present dissertation theme, human pose estimation and tracking, is undoubtedly an influential and promising field of research, given, not only, its wide array of relevant applications, but also its potential influence for the growth of other research areas, such as artificial intelligence. Therefore, the possible results obtained at the end of the present dissertation may provide an important step towards the development of a temporal consistent method for pose estimation and tracking applied to surveillance in autonomous vehicles.

1.4 Document organisation

Following the brief introductory overview presented in this chapter, the subsequent chapters of the present document will aim towards providing a thorough description of the several key aspects necessary to fully understand the development process of the solution implemented within the scope of the dissertation. More specifically, in chapter 2, a literature review of the areas of human body pose estimation and tracking is provided in order to further contextualise the work described here and to familiarise the readers with these areas of research. In chapter 3, a more

detailed characterisation of the main problem that this thesis aims at solving is provided, as well as a description of the current implementation from which the solution builds upon, the proposed solution itself and its possible alternatives. Upon proposal of the approach, the next step is the implementation of the algorithm. This process is reported, as detailed as possible, in chapter 4. Following implementation, the main performance results obtained for the developed algorithm are analysed, compared with the benchmark performance and discussed also in chapter 4. Lastly, the main conclusions of the present dissertation are provided in chapter 5 through extrapolation of all the relevant information gathered throughout the duration of this project.

Chapter 2

Literature review

As previously stated in the chapter 1, the main objective of the present dissertation is the addition of a temporal component to the pose estimation pipeline created by Bosch. In order to fulfil this goal, it is necessary, firstly, to comprehend and consolidate the core concepts of the topics of pose estimation and tracking, acknowledge which are the main pipelines used in these situations and how the current main approaches of this area tackle the problem of pose tracking. Therefore, the present chapter will provide a detailed review of these concepts and methodologies, firstly, for the pose estimation topic (in sub-chapter 2.1), and secondly, for the pose tracking issue (in sub-chapter 2.2), complementing the introductory contextualisation already provided in chapter 1.

2.1 Human pose estimation

Human body pose estimation is an intriguing and versatile area of research that has the potential to grow exponentially in the following years given the most recent reported advancements, specially in the deep learning area. Nevertheless, it is still a very demanding and complex task given, not only, the non-linear and unpredictable nature of the actions and movements carried out by the human physiology, but also, due to the variety and complexity of the scenarios in which the individuals are inserted. In the following sections, a detailed description of the main approaches, pipelines and evaluation metrics currently used in order to overcome the previously mentioned difficulties of this topic is provided.

2.1.1 General methodologies

Based on the approach used to tackle a pose estimation problem, the vast majority of HPE solutions currently available fall into one of two main categories: classic generative methods and discriminative methods.

Regarding generative methods, probably the most well-known example of this category are the pictorial structure models introduced by [2]. These graphical models try to fit a pre-defined template/model represented by a deformable arrangement of parts that are linked by spring-like

spatial connections (figure 2.1) to an image, minimising the conformation and linkage energy costs in order to achieve the best match possible for object recognition and/or pose estimation [22].

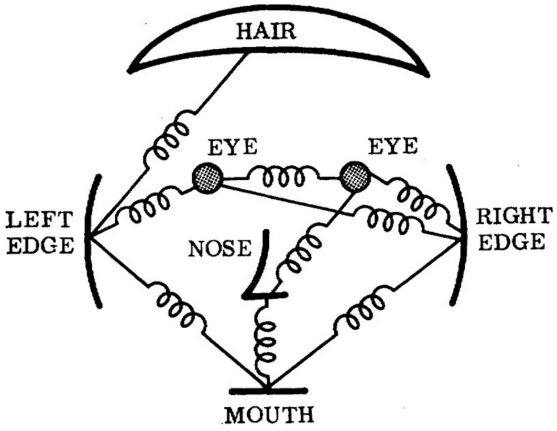

Figure 2.1: Pictorial structures representation of the facial components and their respective linkages (Figure courtesy of [2]).

Although the methods based on these approaches are able to successfully detect and predict body poses from images, their performance can also be impaired by, among other issues, the lack of modelling of interactions suited for the pose illustrated in the image, which can lead to misinterpretation of the spatial arrangement of parts [22, 23]. Moreover, generative methods require a high number of degrees of freedom (parameters) in order to translate the deformable nature of the object/body, which leads to high processing time per frame and slower computation times, making real-time scenarios more difficult to cope with [24].

In contrast, discriminative methods base their approach on the comparison of visible features in an image with body poses "learnt" by the method itself in an attempt to find a positive match between the pose observed in the image and the examples available [24]. These methods are, not only, more fitting for real-time applications than generative approaches, but also more robust, due to a better cope with anatomically viable poses that may not be covered by a given model [24]. Furthermore, the emergence of discriminative methods based on deep learning approaches, namely the use of neural networks for pose estimation introduced by [25], led to a paradigm shift on research and development of pose estimation solutions towards this new class of HPE methods. Therefore, it is not surprising to see that most of the recent HPE solutions [26, 27, 28, 29] resort to the use of convolutional neural networks in their estimation process.

Given the current importance of deep learning HPE methods and the fact that the present work will be based on improving a previously implemented HPE method based on this type of approach, the content of the following sections will focus more specifically towards deep learning methodologies and their characterisation.

2.1.2 Pipelines

As aforementioned, HPE approaches can be classified based on the number of people present in a given image to be analysed. In this case there are two main categories: single-person and multi-person pose estimation.

Single-person HPE methods perform pose estimation throughout the assumption that only one person is present in a given image and that its location within the image is known [30]. Given this information, the aim of these methods is to pinpoint the location of keypoints (i.e. human body joints) that allow a method to estimate the respective pose resulting from the correct keypoint conformation [30]. Based on this objective there are two main options for single-person pose estimation: 1) direct regression, which is only suitable for single-person cases, and 2) Heatmap-based, which first generates a map of the most probable areas for each keypoint and then regresses them based on the heatmaps created previously [30].

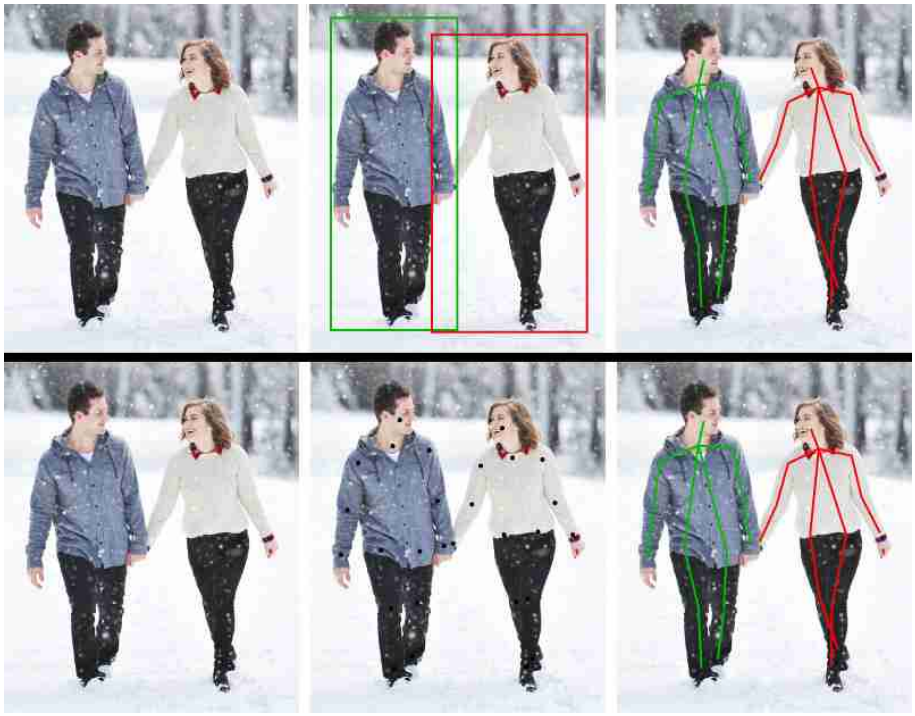

Figure 2.2: Comparison between a top-down approach (top) and a bottom-up approach (bottom) for multi-person pose estimation (Figure courtesy of [3]).

On the other hand, multi-person HPE presents a more demanding challenge when compared to single-person HPE, due to the presence of more than one person in a given image, which not only requires more computation time (due to a higher number of keypoints and possible associations) but also a way to distinguish and associate keypoints to their respective individual in a single image. For this type of HPE, one of the following two major approaches is usually utilised:

- **Bottom-up**: in this approach, a two-stage method is performed in order to obtain an estimation of the poses of all the people illustrated in a given image (figure 2.2). First, a detection

of all the displayed keypoints is executed and, afterwards, the keypoints are grouped by person and connected together resulting in an accurate pose for each person involved [30, 31].

- **Top-down**: although this approach is also a two-stage method, the way top-down estimations fulfil their objective is almost opposite to the process executed by bottom-up approaches (figure 2.2). In other words, top-down approaches firstly perform human detection by bounding each detected person to a box and then the keypoints associated to each box are pinpointed and connected in a manner similar to single-person estimation, with the objective of predicting an anatomically viable pose for each person [30, 31].

Whereas top-down approaches are simpler due to the breakdown of the pose estimation process into several easier-to-perform single-person estimation tasks, their computation time requirements scale up with the amount of people present in the image, making these approaches more time consuming [30, 32]. In contrast, bottom-up approaches computation time requirements almost remain constant with the increasing number of individuals in a given image [32]. Therefore, the balance between accuracy and computational requirements is better in the latter approaches [32].

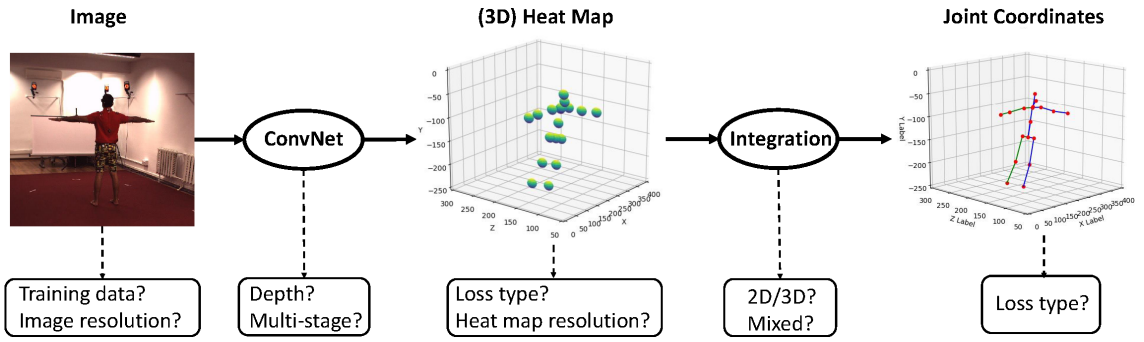

Figure 2.3: Example of a pose estimation pipeline from the work developed by [4] (Figure courtesy of [4]).

In a general manner, HPE approaches achieve their objectives through the implementation of a pipeline comprised of, at least, three phases:

- **Pre-processing** – comprised of tasks performed prior to the estimation process in order to prepare or normalise the input data for the next phases. Comprises tasks such as: background subtraction, which reduces the amount of noise in the image and improves keypoint detection or bounding box creation, which is a necessary task in top-down approaches of multi-person HPE (previously described in this section) [18, 31].

- **Feature extraction** – this is an important task given that not all the information present in an image or video is relevant for the process of pose estimation. In other words, its objective is to process input data, normally through the use of a convolutional neural network, in order to highlight and select useful features, reducing the size of the input for the pose estimation algorithms, which leads to a more time/resource-efficient process [18, 31].

- **Pose estimation** – this task aims at determine the most accurate location of the keypoints based on the feature gradients/heatmaps (as shown in figure 2.3) obtained on the previous phase. Furthermore, it predicts the most likely pose resulting from keypoint connections [18, 31]. The exact method used to obtain an accurate pose estimation varies with the approach used.

2.1.3 Datasets and metrics

Given the necessity of training the neural networks that are part of pose estimation methods, there is an increasing need for the creation of new and larger datasets. Furthermore, datasets also provide a way to evaluate the performance of HPE implementations in a wide range of situations. Microsoft Common Objects in Context (MSCOCO) [33], MPII [14] and PoseTrack [34] are among the most used datasets in the field of deep learning based HPE methods [30]. Whereas the MSCOCO dataset provides a large-scale framework, containing 330 000 context-rich images, for multi-object detection and segmentation in single images [33], the other two aforementioned datasets focus solely on the particular task of articulated HPE [14, 34]. Moreover, MPII and PoseTrack encompass less annotated information: the first dataset includes 25 000 images for evaluation of the HPE task (both single and multi-person) [14], whereas the latter has over 46 000 annotated video frames that can be used for evaluation of both HPE (single-shot) and pose tracking (sequential) approaches [34].

Additionally, the assessment of the overall performance of these pose estimation methodologies on datasets (such as the ones aforementioned), requires suitable evaluation metrics. The most common are Percentage of Correct Keypoints (PCK) [35] and Percentage of Correctly estimated body Parts (PCP) [36], which evaluate, given a pre-determined threshold, if the predicted location for, respectively, a keypoint or part corresponds to the real location [30], and mean Average Precision (mAP) of either Object Keypoint Similarity (OKS) or Intersection-over-Union (IoU), which were introduced by MSCOCO [1].

2.1.4 State-of-the-art approaches

Table 2.1: State-of-the-art single-person HPE methodologies based on the results available on the MPII Human Pose Dataset website [13]. PCKh @ 0.5: PCK with a threshold of 50% of the length of the head segment [14].

Reference	PCKh @ 0.5 (%)	Methodology/Novelty introduced
[37]	93.9	Cascade Feature Aggregation
[38]	92.5	Cascade Prediction Fusion & Pose Graph Neural Network
[39]	92.3	Deeply Learned Compositional Model
[40]	92.1	Multi-scale Structure-aware Neural Network
[41]	92.0	Pyramid Residual Module

Tables 2.1 and 2.2 present a summary of some of the main SOTA methodologies used for single and multi-person HPE, respectively. Furthermore, they are ordered based on their performance,

translated by a metric measurement (PCKh @ 0.5 for single HPE and mAP for multi-person HPE), on the MPII dataset. Additionally, it is also reported the main methodology/novelty introduced by those approaches in order to acknowledge the current most effective methods and innovations.

Table 2.2: State-of-the-art multi-person HPE methodologies based on the results available on the MPII Human Pose Dataset website [13].

Reference	mAP (%)	Methodology/Novelty introduced
[42]	78.0	Pose Refinement Network
[43]	77.5	Associative Embedding
[44]	76.7	Regional multi-person HPE
[45]	75.6	Part Affinity Fields
[46]	74.3	Articulated tracking

2.2 Multi-person tracking

One task closely related to HPE methodologies, that arises from the natural transition of pose estimation algorithms from single images to videos, is the pose tracking process. This task is responsible for adding a temporal component to HPE algorithms and to confer/improve the consistency of person identification throughout the several frames that constitute a video. Following the review provided in the previous sub-chapter regarding HPE methodologies, the aim of the next sections will be to provide a detailed overview and description of 1) the general pipelines used by pose tracking methods, 2) which are the most popular approaches and 3) how these methods are evaluated/validated.

2.2.1 Methodologies

Currently, the most prevalent used approach to MTT problems is tracking-by-detection, a process comprised by two distinct steps [5, 21, 47, 48]. Firstly, a detection algorithm is applied in a per-frame basis in order to highlight relevant features and identify all the individuals present in each frame, similarly to the process performed by pose estimation algorithms. Secondly, the resulting data from the detection step is submitted to an association algorithm, which is responsible for the link of all information corresponding to each individual across the sequence of frames in order to obtain a temporal consistent and accurate representation of the movement trajectory and/or actions performed by each specific person represented in the video input [5, 21, 47, 48]. This latter step is prone to incorrect results given the possibility of occurrence of misleading events such as occlusions and interactions among individuals, which can both lead to the disruption of the flow of one or more association sequences [48]. However, due to these same constraints, this step is also the current focus of most pose tracking algorithms [5], since it can be seen as the main performance bottleneck for this type of algorithms. Moreover, given the existence of estimation algorithms that already yield very accurate detection results (as shown in tables 2.1 and

2.2), the potential for improvement of the overall tracker performance is far greater in the case of the association process.

Over the years, the most predominant approaches used to tackle the tracking problems aforementioned were based on machine learning methodologies and/or other mathematical-based operations. Examples of these techniques are the Kalman filter [49], the particle filter, optical flow, IoU and OKS. These particular set of examples can be used for motion prediction of possible futures locations for the estimations obtained for a given current frame, or, in some cases, to produce spatial and/or temporal information that can be used as metrics for similarity computation among estimation candidates. On the other hand, techniques such as Multi-Hypothesis Tracking (MHT) [50], Joint Probabilistic Data Association Filters (JPDAF) [51], the Hungarian algorithm [52] and Support-vector machines (SVMs) [53] can use the information/metrics provided by the first set of examples and present a reliable and accurate way to associate/classify estimations throughout consecutive frames based on their similarity/affinity values.

As it happened in the field of pose estimation, the emergence of deep learning techniques in the last few years led to a significant increase on the number of tracking algorithms that encompass neural networks in their pipeline [5]. This popularity is justified by the ability of these networks to learn and extract features from input representations and by the SOTA results that these techniques provide in the detection process [5]. More specifically, deep learning techniques, such as convolutional neural networks (CNNs) and its derivatives can be used to extract temporal and/or spatial features from images that are then utilised as similarity metrics during the association process [5]. Furthermore, deep learning based tracking methods may also provide ways to store information throughout several frames through recurrent convolutional neural networks (RCNNs) and its derivatives, such as Long-short term memory (LSTM) networks and Gated recurrent units (GRUs) [5].

2.2.2 Pipelines

Regarding MTT pipelines, the vast majority follows, although with the possibility for small variations, a general sequence (figure 2.4) comprised of the following four stages [5]:

- **Detection** - as previously mentioned in subsection 2.2.1, in this stage, all objects/individuals present in a given frame are identified through the use of bounding boxes (similarly to the initial process used by top-down HPE approaches) [5]. Moreover, this step can be carried out by HPE algorithms, such is the case for the use case described in the present dissertation (more detailed in chapter 3), or by standard object/person detectors provided by benchmark datasets, which can enable tracking algorithms to focus more on the other stages described below [5].

- **Feature extraction** - upon identification of all individuals, the most relevant features of each individual, such as appearance or motion features, are highlighted, extracted and used by subsequent stages for tracking [5]. Common features retrieved in this step are IoU, OKS, optical flow vectors and, more recently, general visual features/similarity metrics through

the use of CNNs [5]. In fact, over the last few years this has been the main stage for application of deep learning methods due to their good capability to extract high-level features from images [5].

- **Affinity computation** - in this stage, the degree of similarity among two distinct detections, based on the features and visual/distance metrics previously extracted, is computed [5]. This step can also be merged with the association stage, since the affinity values obtained in this process are used in the step described below [5].

- **Association** - as previously mentioned, this last stage takes into the account the affinity values calculated for each set of two (or more) distinct detection candidates from two (or more) temporal adjacent frames in order to compute the optimal matching between those sets of candidates [5]. Furthermore, for each set that is considered as an optimal matching for any two given detections in consecutive frames, the same ID is assigned to both candidates as a way to signalise that they correspond to the same object/person [5].

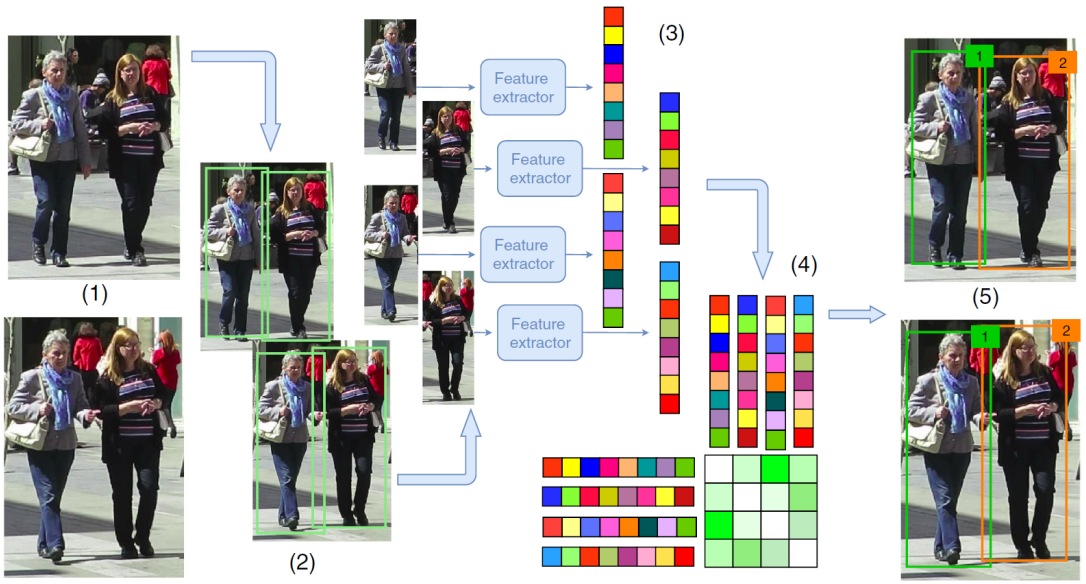

Figure 2.4: Representation of the general pipeline in which the majority of MTT algorithms is based upon. The main four steps that comprise this pipeline are: object detection (2), feature extraction (3), affinity computation (4) and association (5) (Figure courtesy of [5]).

2.2.3 Datasets and metrics

As it was the case of HPE methodologies, MTT approaches also require dedicated datasets in order to: 1) train and test the developed neural networks (if the approach is based in deep learning techniques), 2) validate their pipelines, 3) evaluate the overall performance of the methodology implemented in a given benchmark and compare it with the approaches developed by other research groups. Currently, MOTChallenge [54], KITTI [55] and PoseTrack [34] are the most used

datasets for this purpose, with the first two providing a more general framework with a wide array of annotated objects/targets [54, 55]. On the other hand, the latter dataset, as previously mentioned in section 2.1.3, focus solely on the specific tasks of human body pose estimation and tracking, and is regarded as one of the benchmark datasets for both tasks [34]. Furthermore, given its importance in this particular area of research (which coincides with the main theme of the present dissertation), PoseTrack dataset and the methodologies present on the leaderboard of its multi-person pose tracking challenge (PoseTrack 2017 Challenge 3) will be regarded as reference points and inspirational benchmarks for the development and implementation of the tracking algorithm proposed as the main objective for the present dissertation.

Regarding evaluation metrics, the most commonly used in MTT approaches are, the ID metrics [56], the CLEAR MOT metrics, namely Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) [6] and the set of metrics proposed by [57]: Mostly lost trajectories (ML), Mostly tracked trajectories (MT), Fragments, False trajectories and ID switches [5]. The latter two sets of metrics are the most popular, with the set proposed by [57] being used in both MOTChallenge and KITTI datasets, whereas the CLEAR MOT set is used to evaluate MTT approaches in all three of the datasets previously mentioned in this section. Once again, given the importance of this last set of metrics on the evaluation process for the different datasets previously listed, it will be regarded as the benchmark set of metrics for the present work and will be used as the main evaluation tool for the tracking performance of the solution developed throughout the duration of this dissertation. Therefore, in the next paragraphs, the two metrics that constitute the CLEAR MOT set (MOTA and MOTP) will be addressed and explained in more detail in order to understand how they perform the evaluation process of tracking methodologies.

Firstly introduced by [6], the CLEAR MOT metrics are based on a matching system between ground truth (GT) objects/persons and estimation candidates, which is currently carried out through the pairing of their respective bounding boxes based on the IoU values yielded, as established by the MOT15 dataset [58, 5]. Once the matching is computed, for a given frame, there are three possible situations that can be flagged as tracking errors (figure 2.5):

1. **Misses** (also known as false negatives): these errors occur when a GT object/person, for a given frame, does not have a corresponding match to one of the candidates produced for the same frame. In other words, the tracking algorithm fails to output that a candidate that corresponds and/or is close enough to a given GT object/person to be considered a positive match for this, which leads to a false negative situation given that for the particular frame being evaluated there is an object annotated in the position given by the GT information [6, 5].

2. **False positives**: this situation corresponds to the opposite behaviour described in the previous case, since for this situation it is the GT object that is missing (and not the candidate object, which was the case in the miss situation). This translates into an occurrence where a tracking candidate is proposed to be in a position where there is no matching GT object

[6, 5]. This is one of the most undesirable types of errors since it can give the wrong number, by excess, of passengers that are currently inside the vehicle. Consequently, it may impair the effectiveness/performance of the decision algorithms of action recognition methods, leading to the selection of incorrect outputs. Furthermore, in the case of the MSCOCO dataset evaluation, false positives are also less penalised than, for instance, misses by the mAP metric [1].

3. **Mismatches** (also known as ID switches): this last case occurs when a GT object has matches in two (or more) consecutive frames with candidates with different IDs. In other words, there is a switch in ID for a given GT object due to its pairing with a candidate with a given ID in one frame and in the next frame with a candidate that has a different ID from the previous one. This event can occur in situations where two candidates bounding boxes are too close to each other, which may lead to incorrect matching GT-candidate pairs [6, 5].

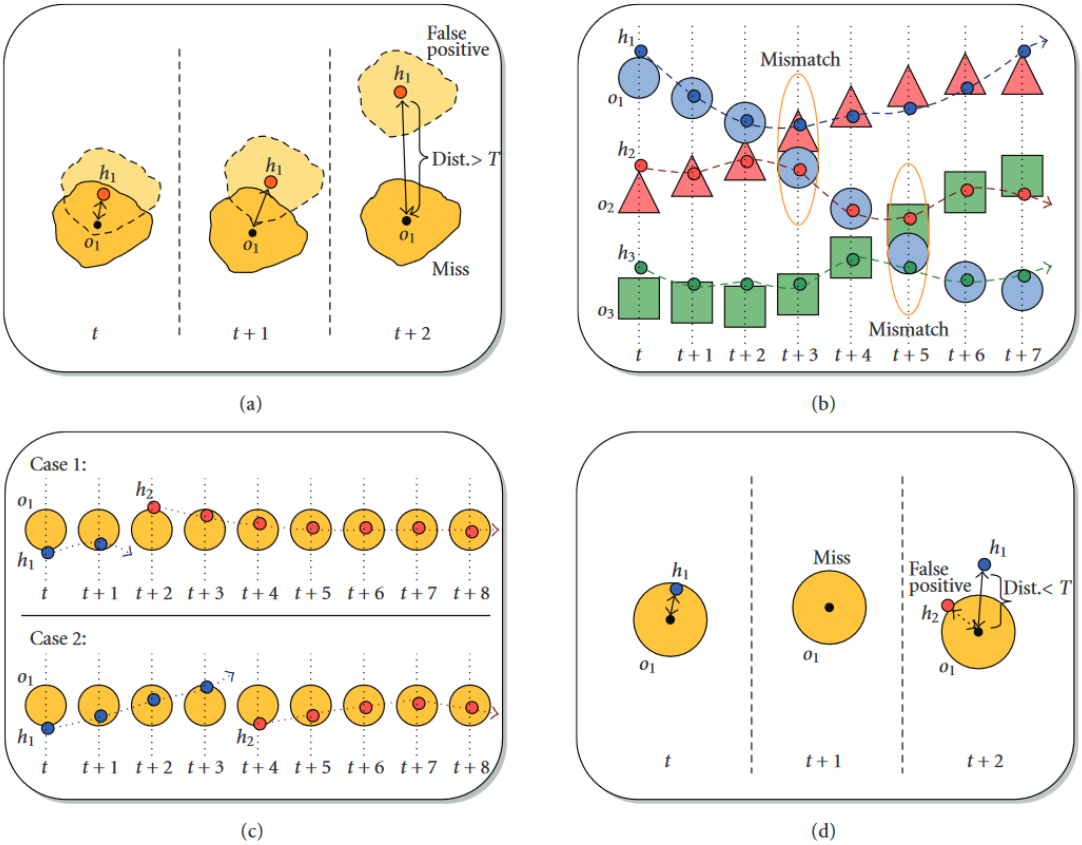

Figure 2.5: Representation of the possible errors that may occur during the tracking process: misses (a and d), false positives (a and d) and mismatches (b and c). The GT objects and the estimation candidates are represented by the letters o and h, respectively. (Figure courtesy of [6]).

The first of the two metrics that compose the CLEAR MOT set, MOTA, takes into account the aforementioned incorrect situations in order to evaluate the tracking performance of a given algorithm. It is calculated according to the following mathematical expression [6]:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t gt} \tag{2.1}$$

where m_t represents the number of misses, fp_t the number of false positive cases, mme_t the number of mismatches that occur for the frame corresponding to time t. The sum of all occurrences for these three types of tracking errors is then divided by the sum of all GT objects for a given video and the result of 1 minus this ratio translates the MOTA result (between 1, or 100%, and -∞, since the number of errors can surpass the total number of GT objects) for the video analysed [6].

The second and last CLEAR MOT metric, MOTP, is more orientated towards the precision of the detector, rather than yielding detailed information about the performance of the tracking component itself [6, 5]. Moreover, it does not take into consideration any information regarding the three most common tracking errors previously described. Instead, the MOTP evaluation metric is calculated by using the following equation [6]:

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{2.2}$$

where d_t^i corresponds to the overlap value for a given matching pair between a GT object and the candidate i in a particular frame t, and c_t represents the number of matches performed for the same frame t [5]. As previously mentioned, MOTP is seen as more of a measurement of precision of the estimation/detection process. Therefore, it will not be regarded as relevant as MOTA for the evaluation process of the tracking algorithm developed in the present dissertation, given that the estimation/detection process, in this particular use case, is carried out by the pose estimation algorithm previously developed by Bosch, and not by the solution implemented in this dissertation.

2.2.4 State-of-the-art approaches

Regarding SOTA approaches for the task of MTT, a summary of the top performers in this area is provided in tables 2.3 and 2.4 for the MOTChallenge and the PoseTrack datasets, respectively. Furthermore, a more detailed overview of the methods listed in table 2.4 will be presented in the following paragraphs of the current section, since their purpose of human body pose tracking is more closely related with the theme and objectives of the present dissertation, than other MTT approaches.

Table 2.3: State-of-the-art MTT methodologies based on the Conference on Computer Vision and Pattern Recognition (CVPR) 2019 tracking results available on the MOTChallenge website [15].

Reference	MOTA (%)	Type	Detector	Open Source
Borysenko et al. (Submitted to ECCV'20)	54.8	Online	Public	No
[59]	51.3	Online	Public	Yes [60]
[61]	47.6	Online	Public	No
[62]	46.7	Batch	Public	No
[63]	43.0	Online	Public	No
[64]	35.8	Batch	Public	No [60]

Table 2.4: State-of-the-art MTT methodologies based on the PoseTrack 2017 multi-person tracking challenge leaderboard available on the PoseTrack website [16]. Only the top 3 methods for both top-down and bottom-up approaches are shown. Anonymous submissions present on the leaderboard are not taken into consideration in this table.

Reference	mAP (%)	MOTA (%)	Type of approach	Leaderboard position
[8]	74.14	64.09	Top-down	1^{st}
[9]	74.04	61.15	Top-down	2^{nd}
[65]	72.57	60.17	Top-down	3^{rd}
[10]	68.78	54.46	Bottom-up	9^{th}
[11]	70.28	53.81	Bottom-up	10^{th}
[12]	63.55	53.07	Bottom-up	11^{th}

The first approach analysed in this section is the top-down method proposed and developed by [8]. Currently, it is the highest performing approach, regarding both mAP and MOTA metrics, on the leaderboard for the PoseTrack 2017 multi-person challenge. Its overall architecture is comprised of three main components: 1) a Clip Tracking Network, based on the High-Resolution Network (HRNet) approach proposed by [7], 2) a Video Tracking Pipeline (figure 2.7) and 3) a Spatial-Temporal Merging component [8].

The aforementioned HRNet methodology (figure 2.6) is a popular top-down solution for the estimation process as it is used by several SOTA approaches present in the PoseTrack leaderboard, such as [8], [9] and [7]. This deep learning based network is comprised of multiple high-to-low resolution sub-networks, which are connected in a parallel configuration (figure 2.6) and exchange information amongst themselves [7]. Consequently, this allows the HRNet to maintain a high level of resolution throughout each step of the estimation process [7]. Therefore, it yields a set of heatmaps (one for each body joint detected) that possess the same high resolution as its respective input feature maps, which can potentially lead to more accurate predictions [8, 7].

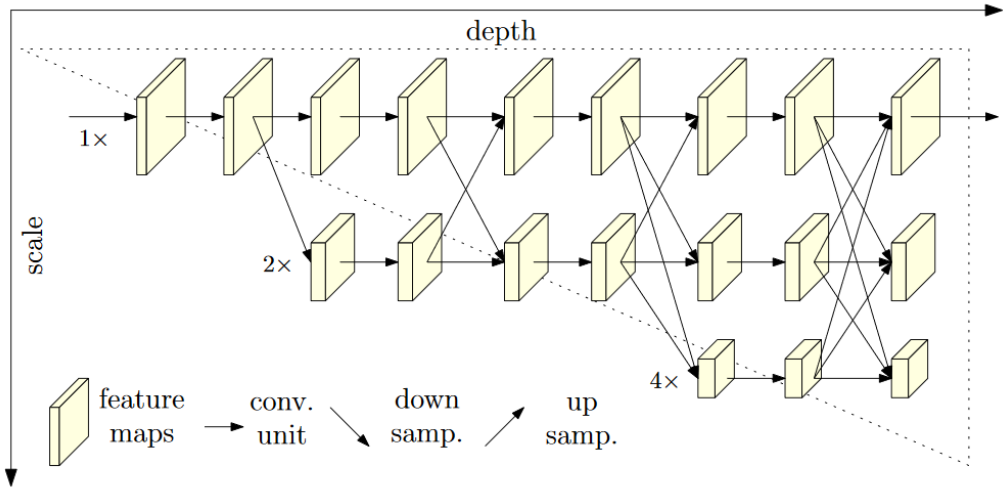

Figure 2.6: Representation of the HRNet architecture proposed by [7] (Figure courtesy of [7]).

Regarding the operation pipeline of the approach introduced by [8], this method starts by selecting the middle frame of a given video clip and by detecting all the persons present on that specific frame. It then propagates those persons throughout all the remaining frames in an attempt to identify and estimate their positions and poses on those frames [8]. These two tasks are performed by the first component of the algorithm, which is, as aforementioned, based on the HRNet architecture introduced by [7].

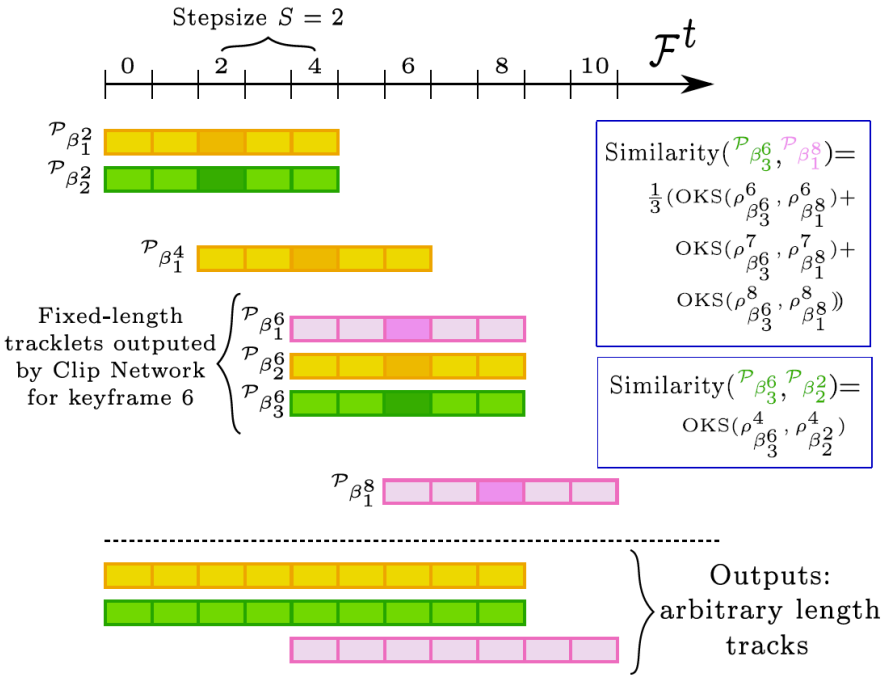

Figure 2.7: Representation of the video tracking pipeline, proposed by [8] and used for merging tracklets based on their similarity (Figure courtesy of [8]).

Afterwards, the tracklets produced by the Clip Tracking Network are passed to the pipeline depicted in figure 2.7, which is responsible for the association of those tracklets into arbitrary length tracks [8]. This association process is performed by the Hungarian algorithm [52] using the OKS as the similarity metric between tracklets 2.7. Finally, the Spatial-Temporal Merging task is responsible for the optimisation of each joint location from the previously yielded predictions, using the Dijkstra's algorithm [66], in order to produce the most consistent poses, both spatially and temporally-wise [8].

Proposed by [9], the runner-up method on the PoseTrack leaderboard is also a top-down approach that introduces several novelties, such as Pose Entailment and Temporal OKS (TOKS). As the first step of this method, the keypoints of each person are estimated using also a HRNet neural network and then these predictions are improved by generating boxes for the current and adjacent frames and by using OKS to decide which predictions are worth keeping [9]. Afterwards, each estimation from two consecutive frames are paired and then converted into tokens [9] as depicted in figure 2.8. Finally, using a transformer-based network, the tokens obtained in the previous step are classified based on whether the estimations that constitute those tokens are a temporal match

or not [9]. Based on this classification, an ID is assigned to the temporal accurate tokens (figure 2.8) [9].

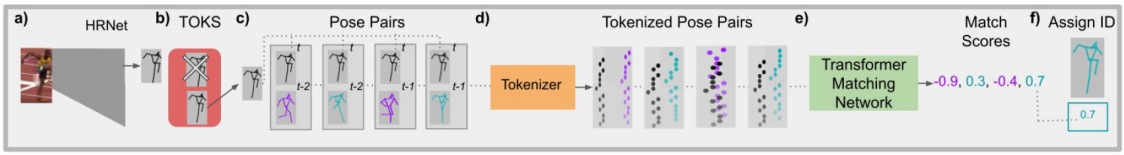

Figure 2.8: Representation of the general pipeline, from estimation to ID assignment, for the approach proposed by [9] (Figure courtesy of [9]).

The third position on the leaderboard is occupied by the approach proposed by [65]. Although the focus of this method is more towards introducing improvements on the estimation part of the algorithm, a new strategy, alternative to the Non-Maximum Suppression (NMS) technique, is proposed for the tracking component [65]. Moreover, the authors use a Mask R-CNN for the detection task, followed by a greedy box generator, which keeps redundant boxes as possible candidates that are sequentially filtered using, first, a box size threshold and then a box confidence threshold [65]. Afterwards, the remaining candidates are compared using the IoU metric and filtered once again based on the comparison results [65].

Contrasting with the previous three approaches, the following three solutions adopt a methodology based on a bottom-up approach. The first of the three methods is also the highest scoring approach regarding the MOTA metric and the second highest in terms of mAP. It was proposed by [10] and introduces a novelty based on temporal flow maps for limbs. Initially, the features extracted from two consecutive frames by a Visual Geometry Group (VGG) network [67] are fed into spatial network (figure 2.9), which then produces joint heatmaps and part affinity fields based on those features [10]. Afterwards, the resulting outputs are fed into the temporal network (figure 2.9), which then regresses the corresponding temporal flow maps [10]. These maps describe each limb movement and can be seen as a representation of the human body flow throughout a given video [10].

The last two bottom-up approaches described in table 2.4, are proposed by [11] and [12] and occupy, respectively, the 10^{th} and 11^{th} places on the PoseTrack leaderboard for multi-person tracking. These two approaches are described in more detail in chapter 3, due to their importance as alternative methods for the main approach proposed. Nevertheless, a general introduction to both approaches is still provided in the following paragraphs of this section.

The 10^{th} and 11^{th} best approaches on the PoseTrack leaderboard share several similarities on their methodologies. Both resort to the use of a deep learning approaches in order to extract temporal information from each estimation [11, 12], and are equally inspired by the work developed by [45], which introduces Part Affinity Fields (PAFs) for body parts association. In the case of [11], the use of specific neural networks regresses Temporal Affinity Fields (TAFs), which yield important information about keypoints connections across several frames [11], and the general idea of this approach is depicted in figure 2.10. On the other hand, the approach proposed by [12] resorts to deep learning networks in order to obtain movement information of every body part present in

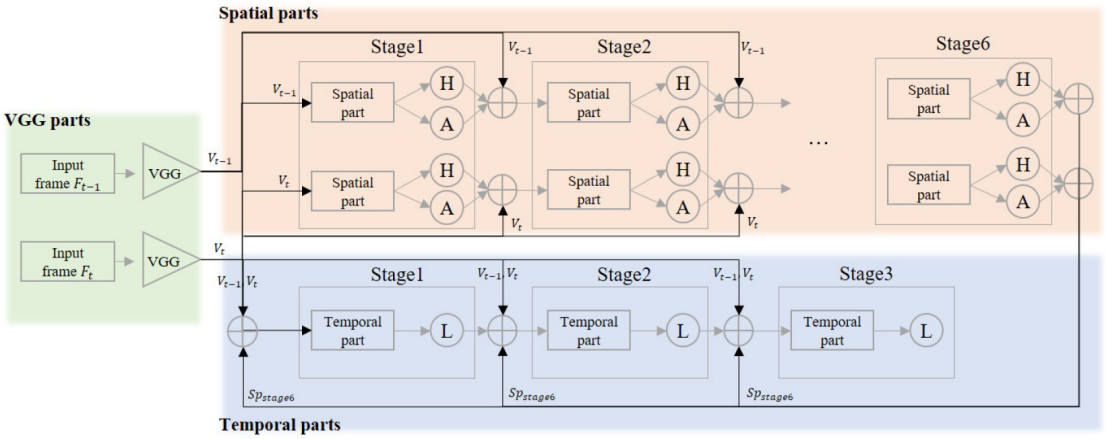

Figure 2.9: Representation of the architecture of the approach proposed by [10], comprised by a spatial network and a temporal network (Figure courtesy of [10]).

two consecutive frames, which is translated by the Temporal Flow Fields (TFFs) regressed by the dedicated temporal network depicted in figure 2.11 [12]. In both cases, the temporal information obtained is used to provide a more accurate and consistent assignment of estimations throughout a given video [11, 12].

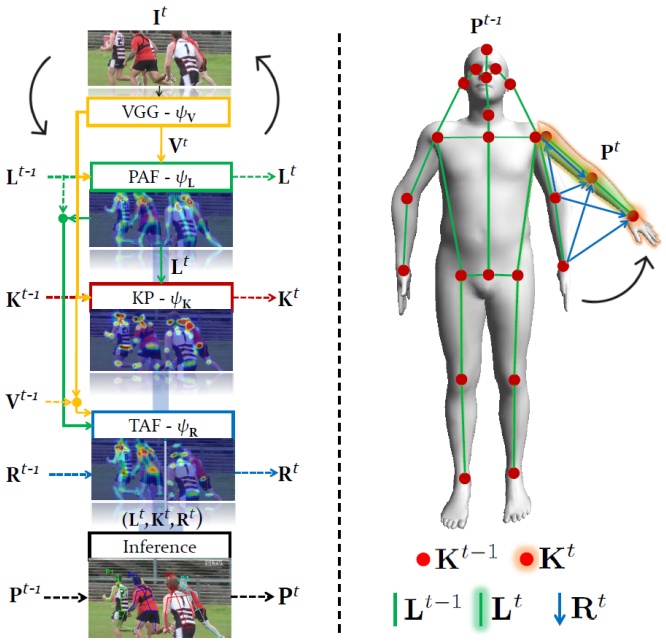

Figure 2.10: Representation of the operation mode of the network proposed by [11], during inference (left) and visual representation the outputs provided by the network: keypoints (red), PAFs (green) and TAFs (blue) (Figure courtesy of [11]).

Throughout the insightful summary of the SOTA approaches from the PoseTrack leaderboard, provided in the present section, it was possible to observe that these top performing algorithms use a wide variety of methodologies to solve, as accurately and efficiently as possible, the problem

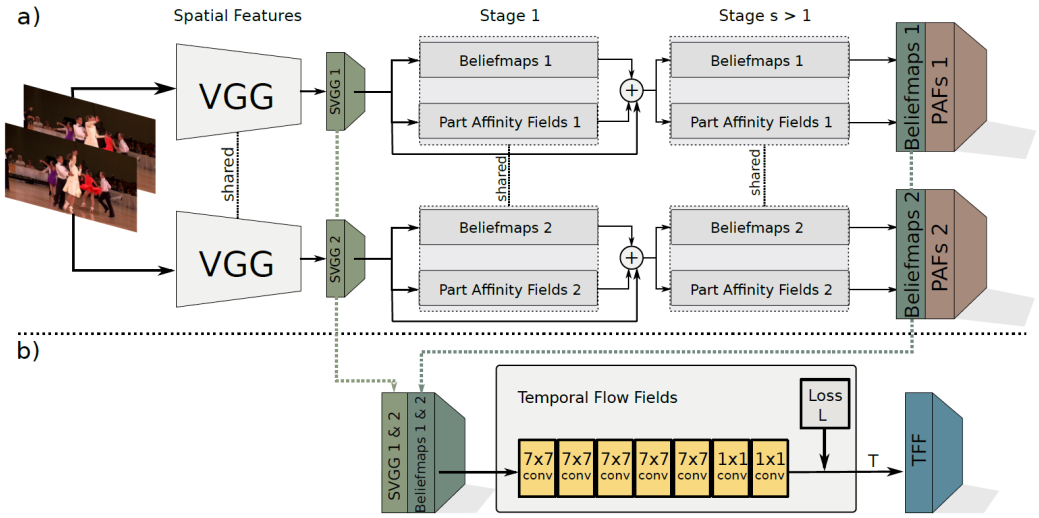

Figure 2.11: Representation of the model architecture proposed by [12]. The temporal network (b) receives information from the spatial networks (a) in order to regress TFFs (Figure courtesy of [12]).

of human body pose tracking. Furthermore, it is also possible to conclude that classical machine learning techniques and metrics are still currently relevant as they are still commonly used as part of the best solutions available for pose tracking. However, and as previously stated in both pose estimation and tracking sub-chapters, there has been an emergence and increasing interest, in the last few years, on the development of deep learning based approaches given the potential, advantages and insights that they provide in comparison to the more traditional methodologies. Therefore, it is very likely that in the next few years, the number of deep learning based approaches present in the SOTA leaderboards for pose tracking will significantly increase as the understanding and improvement of these techniques, undoubtedly, continues to grow.

Chapter 3

Characterisation of the problem

The main objective of the present dissertation is the development and implementation of a viable algorithm capable of adding a tracking component to the current pose estimation method developed by Bosch. Once implemented, the resulting algorithm will be able to, not only, assist in the estimation process (yielding more accurate and consistent pose results), but also add a temporal component to the approach, allowing it to track several poses throughout the multiple frames of a video (instead of analysing each frame as an individual image with no additional context).

In order to comprehend which is the best suited solution for the aforementioned scenario, it is of utmost importance to study: 1) how the current pose estimation approach is implemented, namely its inputs, outputs and general pipeline, 2) which requirements the tracking algorithm needs to fulfil and 3) how state-of-the-art algorithms tackle the pose tracking problem and which are the most popular approaches used in the literature. Since the latter point is already thoroughly addressed in sub-chapter 2.2.4, it will only be briefly mentioned throughout the present chapter, whereas the remaining two points will be introduced and described in more detail along the following sub-sections.

Afterwards, and taking into account the information collected from the literature, a main approach, as well as a few possible viable alternatives, will be proposed in order to tackle and solve the problem described in the present dissertation.

3.1 Current implementation

Due to the Bosch Group confidentiality policy, a thorough and detailed description of the currently implemented pose estimation algorithm is not possible in this document. Despite this fact, some general information about the approach can be disclosed for contextualisation of the present dissertation, namely the fact that it is a pose estimation algorithm based on a bottom-up deep learning approach. More specifically, it uses a convolutional neural network in order to convert an input image into visual features, which are posteriorly classified and translated into keypoints. These keypoints are assembled into viable sets (poses), which are the main output of the algorithm, and are assigned to their respective human representations in the original input image.

Apart from bottom-up approaches, human body pose estimation methods can also be classified as top-down approaches. In contrast with the first class, top-down methods firstly perform human detection by bounding each detected person to a box and then, in each box, pinpoint and connect the keypoints in a similar manner to single-person estimation, with the objective of predicting an anatomically viable pose for each person [30, 31]. In general, this type of approach yields better results in terms of accuracy than bottom-up approaches, which translates into most SOTA methods being top-down approaches. This behavior can be justified by the use of global and body structural information by top-down approaches (in opposition to bottom-up approaches that do not rely on that type of context), which results in less false positive detections [68]. Despite this fact, top-down approaches rely heavily on the performance of their human detectors and are, in general, more demanding in terms of computing processing given their two-step estimation processes [30, 32]. Furthermore, they are more prone to estimation errors due to occlusion, complex poses and/or overlapping than their counterparts [69, 68]. These are very common occurrences inside vehicles, with the examples of partial occlusion of a passenger behind a seat or overlapping due to close interaction with another passenger being the most relevant ones. Taking into account these facts, bottom-up methods, despite their lower accuracy, appear to be the most suitable approach for the specific problem of human body pose estimation inside vehicles, which supports the idea that not always the best overall performing solution is the best performing solution for a given use case.

3.2 Internal datasets

Before the development and implementation of the tracking algorithm itself, it is necessary, first and foremost, to select and characterise the set of videos that will be used to test the to-be-developed solution. For this purpose, a subset of the internal VideoPose dataset, which contains temporally annotated videos used by Bosch to test and validate possible sequential algorithms, is used. Even though, a detailed description of the content of each video present in the aforementioned subset cannot be disclosed, due to the Bosch Group confidentiality policy, it is still possible to provide a general overview of its organisation.

Regarding this subject, the subset used in this dissertation to test and validate the developed tracking algorithm is comprised of three video datasets, designated here as datasets #1, #2 and #3, that capture different in-vehicle perspectives of a wide variety of interactions between passengers. Moreover, this subset has a total of 72 temporally labelled videos, which are distributed by the three datasets (12, 39 and 21 videos respectively).

Additionally, the model of the pose estimation algorithm (responsible for providing the keypoints estimations that will be used as inputs for the pose tracking solution) developed by Bosch was previously trained using an internal single-shot dataset: BoschCOCO, which, as the name entails, is inspired by the MSCOCO dataset and is comprised by several images that depict a wide array of in-vehicle situations.

3.3 Approach proposal

Upon study of the use case at hand and the possible solutions available (reviewed in chapter 2), the next logical step is the definition of the approach in which the algorithm to be implemented will be based upon. For this selection, it is necessary to take into consideration a few important requirements:

1. Must be computationally efficient/light, given the necessity to run the algorithm in real time and, possibly, in restrictive hardware conditions, i.e., without a powerful central processing unit (CPU);

2. Must improve overall (non-tracking and tracking) performance of the current implementation;

3. Must promote/maintain person ID consistency throughout video frames, decreasing the number of ID switches, in order to improve the performance of pose estimation based algorithms, such as action recognition methodologies;

4. Must be a reproducible algorithm, i.e, an implementation and/or detailed description of the methodology used in the algorithm in question must be available online in order to validate and reproduce its results obtained.

Given the previous list, the solutions currently available and the nature of the use case of this dissertation, one main approach was proposed (figure 3.1).

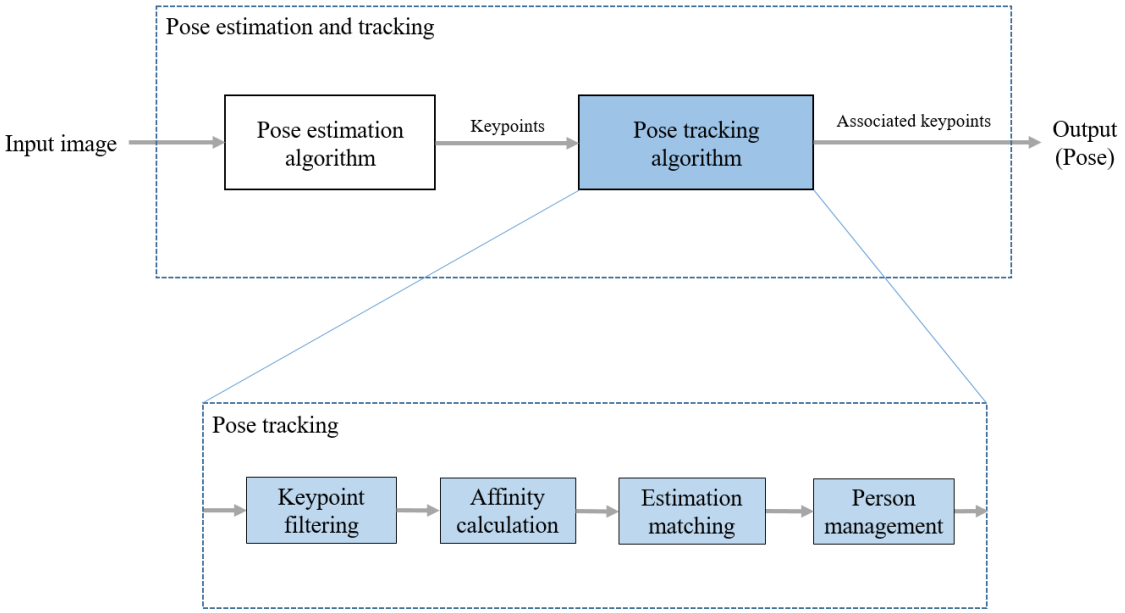

Figure 3.1: Representation of the main approach proposed for this dissertation. The pose tracking module to be developed is highlighted in blue and its general pipeline, comprised of four main steps is shown in more detail.

Moreover, two alternative solutions were also suggested as a precautionary measure, in case the main approach failed to produce the expected performance improvement or hit an unforeseen obstacle during implementation that led to the impairment/blocking of dissertation progress.

The main solution proposed is based in a modular approach, where the tracking algorithm is separated from the pose estimation algorithm, receiving only the keypoints produced by the latter as an input. The pose tracking module is responsible for the association of each set of keypoints, received from the pose estimation algorithm, with their respective person ID based on information gathered from previous and current frames, in order to yield updated human body poses that are consistent throughout time. In a general way, this approach features a pipeline (figure 3.1) mainly comprised of the following four steps:

1. **Keypoint filtering**: upon receiving a set of estimations for a given frame, the first task performed by the proposed tracking algorithm is the selection of only viable estimations through the application of filters that remove keypoints that do not meet certain criteria. This process not only reduces the probability of false positives but also decreases the computation time required for the following steps due to the reduction of the number of new estimations from the initial list received from the estimation algorithm.

2. **Affinity calculation**: the second step of the pipeline is responsible for the calculation of the similarity between the filtered estimations and the current keypoints for each person (stored from the previous frame). Popular distance metrics, such as IoU, OKS or PCKh [70] are used in order to compute the affinity level between two estimations/joints.

3. **Estimation matching**: the resulting values of each comparison performed in the last step are then used to perform an association between the new estimations set and the current set of tracked persons. This assignment process aims at finding the optimal combination of estimation/person pairs, based on the affinity values previously provided, that maximises the similarity of each pair in order to obtain the most accurate matching results possible. One of the most broadly used methods for this type of computation is the Hungarian algorithm [52].

4. **Person management**: lastly, once all viable estimations are matched with their respective person IDs, the latter are updated with the new information and stored in order to be used in the next frames. However, this process is not always this linear since there are cases in which: 1) the number of viable estimations is higher than the number of persons currently tracked or 2) not all persons possess a matching estimation for a given frame. Therefore, it is necessary to manage each person situation accordingly, based on the association process outcome. In the first case, it may be necessary the initialisation of a new person ID, if the particular estimation is viable enough, and update that person with the respective keypoints. On the other hand, in the second case, the lack of matching estimations throughout several frames should lead to the removal of that person from the tracking set since it is possible

the person in question may have already left the area or is not visible due to object/person occlusion.

This is a simple and relatively easy to implement solution for the problem at hand that combines several classical, yet still currently relevant and popular methods with data management and logical operations developed taking into account the overall system architecture. Moreover, it can be used in conjunction with different pose estimation algorithms given its modular nature (apart from the keypoints received, it is completely independent from these methods). This feature provides a higher level of freedom, given that any improvement/modification made in the pose estimation module does not require the alteration of the pose tracking algorithm (and vice-versa) in order for the latter to be compatible with the newer version. In other words, it creates an abstraction from the estimation algorithm (treating this algorithm almost as a black box) and eliminates most impairing dependencies/constraints that arise from developing a tracking algorithm embedded into an already existing estimation method.

Regarding the methodologies/metrics that are utilised in this proposed solution, they are inspired from several SOTA approaches, specially methods present in the PoseTrack leaderboard for multi-person tracking, which use classical techniques and/or metrics such as the Hungarian algorithm for data assignment ([8, 71]), NMS for box filtering ([72, 7]), optical flow ([72, 7]) and IoU measurements ([73, 65, 32]) or OKS distances ([8, 9]) for affinity calculation between estimation candidates and person IDs. Additionally, the considerable variety and quantity of machine learning-related techniques that are still used by SOTA approaches also adds more weight and support to the viability/credibility of the use of classical methods, in opposition to a deep learning approach, as the foundation for the main proposed approach for this dissertation.

3.4 Alternative approaches

Additionally, as previously mentioned, two alternative approaches were also considered in the scope of the present dissertation. The first is based on a recent article [11], in which a bottom-up estimation and tracking solution is documented (figure 3.2). This method has its foundations on the pose estimation algorithm developed by [45], which introduces the use of PAFs for the association of body parts [45], and builds upon it through the introduction of a temporal component: Temporal Affinity Fields (TAFs), which, combined with the keypoints and PAFs generated each frame, allow inference of human body pose throughout time [11]. Currently, it is the highest and second highest rated bottom-up approach regarding, respectively, mAP and MOTA metrics, in the PoseTrack 2017 challenge for multi-person tracking. Aside from being one of the benchmark methods for bottom-up pose tracking, an online repository containing an example implementation of this approach (available at: https://github.com/soulslicer/openpose/tree/staf) is also provided, which allows an easier reproducibility of this technique in other use cases. However, due to the implementation complexity, both in terms of comprehension and adaptability, of this solution to the current use case, it will be only considered as a strong alternative to the main solution proposed in the present dissertation.

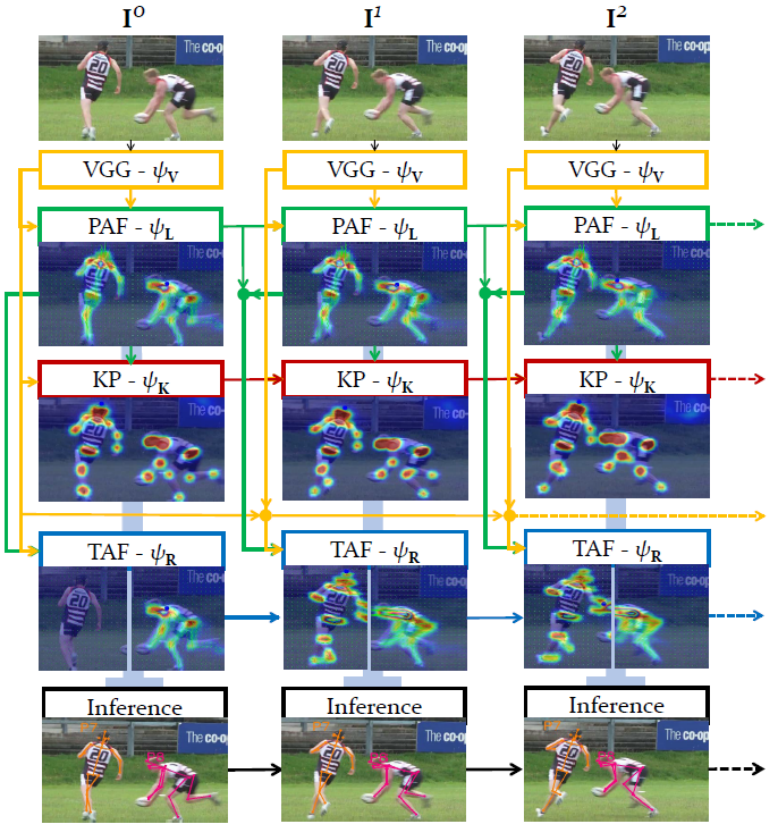

Figure 3.2: Representation of the architecture of one of the estimation and tracking models proposed by [11]. The interactions between the different modules in consecutive frames, regarding its inputs and outputs (keypoints, TAFs and PAFs) are depicted (Figure courtesy of [11]).

The second approach is based on the work developed by [12], which is currently classified one position below the method previously described ([11]) on the PoseTrack 2017 multi-person tracking leaderboard. As the aforementioned alternative, this online approach (figure 3.3) is also a bottom-up solution that takes inspiration on the PAFs method and introduces a novel tracking component based on TFFs. These fields are obtained through a neural network, that receives as input the predictions of two consecutive frames, and translate the movement direction of each body joint between those two frames [12]. Afterwards, the TFFs values are used as a similarity metric in a bipartite graph matching process in order to assign the estimate candidates to their respective person representations [12]. This is an interesting solution that presents a possible viable alternative to optical flow motion prediction approaches, which are known to be unreliable in moving scenarios, and introduces a fairly simpler deep learning approach than most other neural network based methods. However, despite these advantages, an online repository of this approach is not currently available, which lowers its reproducibility level considerably and means that in order to replicate or even adapt this methodology to the present case, the author of this dissertation has to rely solely in the respective article description of the implementation process, which is not an optimal solution in this context.

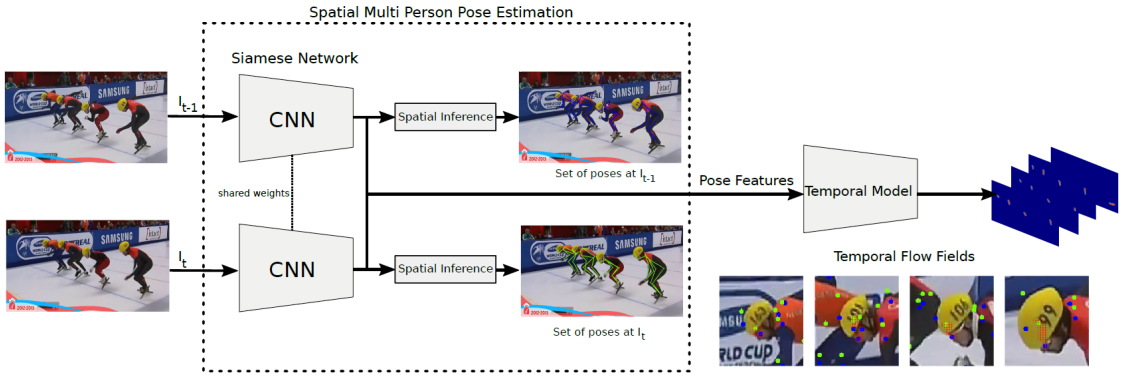

Figure 3.3: Representation of the approach proposed by [12]. The pose features from two consecutive frames are used to predict the TFFs for each joint, which are then used in the association process (Figure courtesy of [12]).

Despite the previously stated features and advantages of the two alternative approaches, both share a common factor that makes them less adequate for a viable solution of the present problem than the main approach selected: they incorporate deep learning elements, such as convolutional neural networks, in their methods. Normally, this factor would be extremely beneficial for an approach, given that the majority of SOTA approaches are based in deep learning. However, these approaches require a training process with temporal datasets, i.e., datasets that contain temporal annotations throughout the set of frames of a given video. Given the lack of internal datasets of this nature that are extensive enough to carry out an efficient training of the neural networks, it would be necessary to resort to public datasets or even synthesize artificial data using computing resources in order to perform this task. The use of public datasets could lead to the loss of specification since they lack the particular context (provided by internal data) for the problem of human pose tracking inside of a vehicle, which is a particular case of tracking given its unique environment conditions (for example: vehicle movement, closeness of the passengers to the camera in some situations, exchange of seats that leads to overlapping or occlusion behind front seats). Furthermore, the use of synthetic data also poses an issue given the complexity and amount of time required for creation of this type of dataset, which would impair severely the amount of time available for the implementation of the tracking approach itself. For the aforementioned reasons, the selection of deep learning methods as possible solutions for this specific problem of human pose tracking was less acknowledged in favor of more simple/classical approaches, such as the data association techniques previously mentioned throughout the present chapter.

Chapter 4

Implementation

Upon familiarisation with the core concepts of HPE and MTT methodologies, as well as the definition of the main approach pipeline (and its alternative solutions), the next logical step is the implementation of the proposed methodology. In the following sections of the present chapter, a thorough (when possible) description of each step of the aforementioned process will be provided in order to further comprehend, not only the architecture of the proposed solution, but also its general operation mode. Lastly, the performance evaluation process of the implemented approach, carried out in order to validate its improvements in regard to the initial algorithm and its compliance with the requirements stated in the previous chapter, will also be detailed in this chapter.

4.1 Tracking algorithm

In the present section, the general structure of the pose tracking algorithm proposed in chapter 3 and the reasoning behind each step of its pipeline will be discussed. Furthermore, it will be provided an overview of the methodologies adapted from the SOTA approaches and how they operate in order to obtain the desired results.

The general pipeline of the pose tracking algorithm implemented in the present dissertation is described in algorithms 1 and 2. The algorithm 1 encompasses the steps followed during the initialisation process of the tracking algorithm, more specifically, the handling and management procedures of the inputs received from the pose estimation algorithm after processing the first frame of a given video. On the other hand, the algorithm 2 is responsible for the processing of the inputs corresponding to the following frames, i.e., all the frames that have at least one preceding frame regarding the timeline of the video that they are part of. The implementation of both algorithms was performed using the version 3.6.7 of the programming language $Python^{TM}$ [74].

The initialisation procedure of the proposed algorithm, depicted in algorithm 1, is a fairly simple process comprised of two main steps: 1) storage of the information from the inputs (new

Algorithm 1: Proposed pose tracking algorithm (first frame)

Input: *Set of new estimations* (*Coordinates*, *Scores*, *Box area*, *Confidences*)
Output: *Set of poses*

```
Store new estimations
for each estimation in new estimations set do
    Initialise new person ID
    Update person with estimation information
```

Algorithm 2: Proposed pose tracking algorithm (following frames)

Input: *Set of new estimations* (*Coordinates*, *Scores*, *Box area*, *Confidences*)
Output: *Set of poses*

```
Store new estimations
for each estimation in new estimations set do
    if estimation's bounding box area or score below given threshold then
        Discard estimation
    end
for each person in current persons set do
    Compute affinity (IoU + OKS) between person and each estimation
Compute association (Hungarian algorithm)
if list of estimations without person match is not empty then
    for each estimation without person match do
        if estimation's average confidence and score above given threshold then
            Initialise new person ID
        else
            Discard estimation
for each person in current persons set do
    if person has estimation match for current frame then
        for each keypoint in matched estimation do
            if keypoint confidence below given threshold then
                Replace keypoint coordinates with last frame coordinates
        Update person with matched estimation information
    else
        Update person using last frame information
    if person has no match for a given time then
        Delete person ID
```

estimations) received from the pose estimation algorithm (line 1 of algorithm 1) and 2) initialisation of a new person ID (lines 2-4 of algorithm 1), for each new estimation received, and update of the newly created person with the information contained in its respective estimation.

In this case, a person is a data structure, with an unique ID number (used to differentiate each passenger in a given frame/video), that stores information from each estimation that is matched to a particular person ID up to a given number of frames. In other words, each person can store its respective assigned estimations from each of the last X frames, where the number X solely depends on the frame rate of the current video. Additionally, the information stored for each frame is comprised of:

1. Coordinates (x,y) for each keypoint location;
2. Estimation score, which translates the pose estimation algorithm confidence that a passenger is present in the predicted location;
3. Bounding box area, resulting from the box generated using the minimum and maximum coordinates locations;
4. Confidences of each keypoint predicted location.

The previously listed parameters are then used in the several stages of the algorithm 2, in order to compute the most accurate estimation assignment possible.

Once the initialisation process is completed, the following frames of a video are processed using the pipeline described in the algorithm 2. The aforementioned pipeline follows the logic previously described in chapter 3, which divides the proposed approach into four main steps: 1) keypoint filtering, 2) affinity calculation, 3) estimation matching and 4) person management. This four stages will be explained in the following sections, as well as the reasoning behind the development of each one of them.

4.1.1 Keypoint filtering

After receiving and storing new estimations for a given current frame, the next task performed by the proposed algorithm is the filtering of those estimations (algorithm 3) based on two parameters: score and bounding box area. This approach was inspired by a similar method proposed by [73]. The reasoning for the use of a score threshold lies on the removal of predictions that have a low probability to be viable representations of a person. This selection alone is able to remove several false positive detections that would be detrimental for the rest of the pipeline and could impair the matching process through the assignment of inaccurate estimations to a given person.

In order to complement the aforementioned filtering process, an area threshold for the bounding boxes generated during this stage (based on the minimum and maximum keypoints locations) for each estimation, was also utilised. However, this second approach was not the first choice for the designated process, given the existence of the fairly popular NMS method. This latter technique uses the IoU metric to, first, identify a set of detections with similar bounding boxes and

Algorithm 3: Keypoint filtering algorithm (from lines 2-5 of algorithm 2)

```
1 for each estimation in new estimations set do
2     if estimation's bounding box area or score below given threshold then
3         Discard estimation
4     end
```

then remove all but the bounding box with the highest confidence score from that set [75]. This process is repeated until only the highest confidence bounding boxes for each predicted location remain [75]. In most cases, NMS is an effective solution for the removal of ambiguous bounding boxes, that share the same general area of a correct prediction, but are seen as duplicates or false positives that impair the overall estimation score of an algorithm.

Despite its effectiveness in other cases, NMS failed to produce a positive impact on the particular use case addressed in this dissertation. This inability to positively affect the filtering process of the present approach was simply due to the much smaller size of the bounding boxes of potential false positives when compared with the bounding boxes from the correct predictions. This fact meant that, when applying NMS to the new estimations, the IoU threshold was never reached, given the difference in size between the bounding boxes. Moreover, the adjustment (decrease in this case) of the IoU threshold to increase the sensitivity to this particular case could not be performed. This was mainly due to the increase possibility of removing correct predictions in the neighbourhood of another correct bounding box if they had a low, yet above the IoU threshold, overlap between them.

Therefore, in order to remove the small incorrect detections from the set of estimations, a bounding box area filtering was applied, in conjunction with the prediction score threshold aforementioned. The fixed values used for both thresholds were based on visual inspection of the tested videos and analysis of the score and area values yielded by the pose estimation algorithm.

4.1.2 Affinity calculation

Following the last stage, the viable filtered estimations are then submitted to an affinity calculation process (algorithm 4). This procedure encompasses the computation of two popular distance similarity metrics: IoU and OKS. The IoU metric is calculated, for any two given bounding boxes, through the ratio between their respective overlap and union areas (figure 4.1) and allows the measurement/comparison of the similarity among two bounding boxes.

Algorithm 4: Affinity calculation algorithm (from lines 6 and 7 of algorithm 2)

```
1 for each person in current persons set do
2     Compute affinity (IoU + OKS) between person and each estimation
```

The second metric, OKS, provides an average measurement of keypoint similarity between estimation and persons and is, as stated by MSCOCO [1], calculated using the following equation:

$$OKS = \frac{\sum_i [exp(-d_i^2/2s^2k_i^2)\delta(v_i > 0)]}{\sum_i [\delta(v_i > 0)]} \tag{4.1}$$

where d_i is the Euclidean distance between an estimation keypoint i and the corresponding person parameter, s is the object scale, k_i is a constant linked to a given keypoint i and v_i is its visibility flag (i.e. if it is labelled or not). During the calculation of this metric, only the labelled keypoints ($v_i > 0$) are considered and the final result is obtained through the averaging the OKS of each visible keypoint.

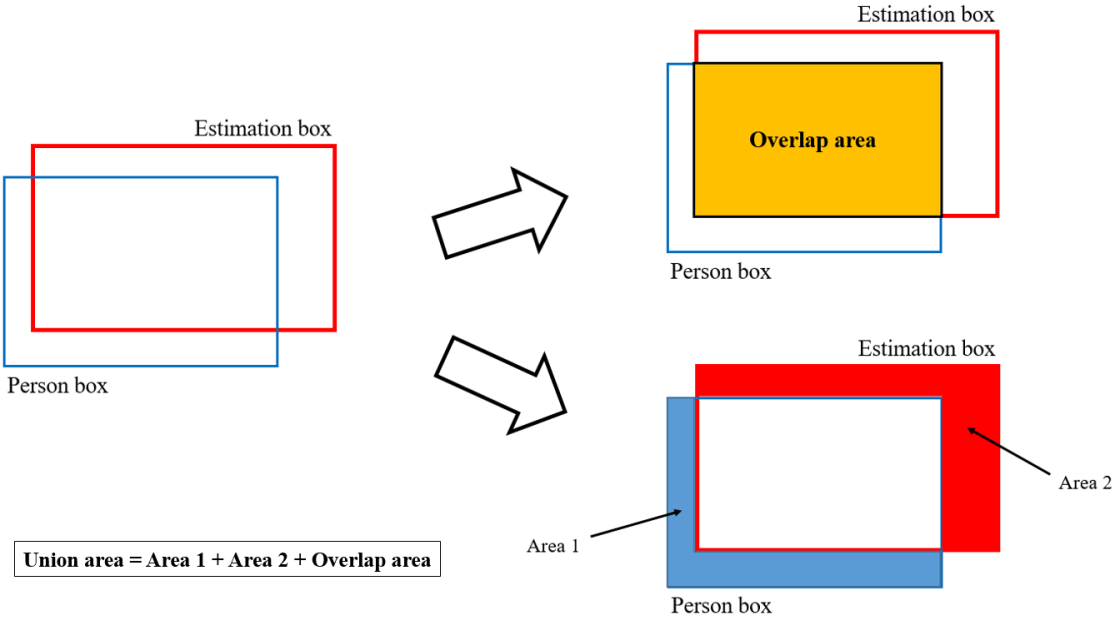

Figure 4.1: Visual representation of the method used to obtain the parameters (overlap and union areas of two bounding boxes) used for the calculation of the IoU metric. The ratio between the overlap area and the union area yields the IoU result for two given boxes.

Once both metrics are calculated for a given estimation-person pair, their values (ranging between 0 and 1) are complemented and the sum of those two values yields the affinity level (ranging between 0 and 2) for the estimation-person pair. The complement calculation is used to invert the logic of the metrics, i.e., the higher the metric value is, the worse the affinity level gets. Although this operation seems counter-intuitive, it is necessary for the next step of data association, since the method used for that purpose tries to minimise the overall cost of the association process. This means that estimations are assigned to a person based on how low their "inverted" affinity value is.

The calculation of IoU and OKS metrics is performed for each estimation-person pair and the results of this operation are stored in a cost matrix that is used as an input for the next tracking stage. Given the possibility of the estimation and person sets having a different size, the matrix

yielded in this step may have a square (if both sets have the same number of entries) or rectangular shape (if the number of new estimations is different from the number of currently tracked persons).

4.1.3 Estimation matching

Upon calculation of the affinity between the new estimations set and the current persons set, the resulting cost matrix is then used to associate those two sets and yield the combination with the lowest "inverted" affinity values possible (algorithm 5). In this case, the matching process is performed using the Hungarian algorithm, which is implemented in the proposed solution through the use of the *linear_sum_assignment* function from the SciPy [76] *optimize* library.

Algorithm 5: Estimation matching algorithm (from line 8 of algorithm 2)

1 Compute association (Hungarian algorithm)

The assignment problem is solved using the following equation [77]:

$$Optimal\ assignment = min \sum_{i} \sum_{j} C_{i,j} X_{i,j} \tag{4.2}$$

where $C_{i,j}$ is the cost ("inverted" affinity value) for the matching between an estimation i and a person j, whereas $X_{i,j}$ is a boolean variable that is 1 if the estimation i is assigned to the person j and 0 if it is not. The previous equation aims at minimising the matching cost, in order to assign the estimation with the best similarity score possible to each person. Based on the assignment results, the estimations are then paired with their respective person IDs and the latter are updated through the procedure described in the next section. Furthermore, the handling process of estimations without matching will also be addressed in the aforementioned section.

4.1.4 Person management

The last stage of the proposed tracking algorithm (lines 9-24 of algorithm 2) is also the most complex one, being responsible for the initialisation, update and removal of person IDs throughout the different frames of a given video. Moreover, this stage is very important for the coherence of the tracking process since it correlates all the information (for instance similarity metrics and matching results), from the previous three stages and decides the fate of each currently tracked person. It is mainly divided into two steps: 1) initialisation of new person IDs for viable new estimations that are not matched with any of the current persons (algorithm 6), and 2) update/removal of the person IDs based on different matching parameters (algorithm 7).

The first step previously enumerated is a situational event, i.e., it only occurs in the particular case where the number of new estimations exceeds the amount of currently tracked person IDs, resulting in estimations without any assigned person ID. In this step, a list of those non-matching estimations is retrieved and each of those predictions is submitted to a more restrictive filtering than the one previously described in section 4.1.1, in order to ensure that only viable estimations

Algorithm 6: First stage of the person management algorithm (from lines 9-14 of algorithm 2)

```
1 if list of estimations without person match is not empty then
2     for each estimation without person match do
3         if estimation's average confidence and score above given threshold then
4             Initialise new person ID
5         else
6             Discard estimation
```

lead to the initialisation of a new person ID. Moreover, this measure enforces one of the main goals of the proposed algorithm, which is to avoid the generation of false positive estimations that can result in an inaccurate/excessive representation of the real number of passengers inside a vehicle.

The filtering process, performed in this step, is carried out using the average confidence of the keypoints of the assessed estimation and, once again, its prediction score. The first, and new, parameter introduced here for filtering is, as previously stated, the average confidence of all the annotated/visible keypoints that comprise a given estimation. This parameter yields another possible metric for the evaluation of the prediction viability and is compared with a fixed threshold, which was defined based on visual inspection and data analysis of the aforementioned metric in several videos. The second parameter (score) follows the same reasoning as the one previously stated in section 4.1.1. However, for this particular step, its threshold is increased given the necessity to ensure that a new person is only initialised if there is a very high confidence that the prediction in question really represents a passenger. This is an important selection step since the initialisation of a false positive, that will be propagated throughout the following frames, is more detrimental to the overall algorithm performance than an error in the matching of an already tracked person for one frame.

Algorithm 7: Second stage of the person management algorithm (from lines 10-24 of algorithm 2)

```
1  for each person in current persons set do
2      if person has estimation match for current frame then
3          for each keypoint in matched estimation do
4              if keypoint confidence below given threshold then
5                  Replace keypoint coordinates with last frame coordinates
6          Update person with matched estimation information
7      else
8          Update person using last frame information
9      if person has no match for a given time then
10         Delete person ID
```

Afterwards, any person initialised through a viable estimation, filtered in the last step, is added to the currently tracked list of person IDs and the second step of the person management stage is started. During this step, each currently tracked person (including the new persons initialised in the last step) is verified taking into account its matching situation (i.e. if an estimation was assigned to its ID) for the current frame. Depending on the outcome of the assignment process described in section 4.1.3, three situations may arise:

1. **Update**: the person has a matching estimation, which means that it will be updated with new keypoint coordinates in the current frame. Additionally, this situation is always triggered for the person IDs newly initialised in the present frame, since they have a guaranteed matching in their first frame of existence.

    - Addressed in lines 1-6 of algorithm 7.

2. **Missing**: there is no new estimation assigned to the designated person ID. In this case, the person in question is flagged to acknowledge that, for a given frame, a matching estimation is not present. Additionally, the person is updated, for the current frame, with the keypoint information from the previous frame, albeit with all keypoints flagged as non-visible.

    - Addressed in lines 7 and 8 of algorithm 7.

3. **Deletion**: once a particular person ID is flagged for a given number (based on the video frame rate) of consecutive frames for not having a matching estimation, it is removed from the currently tracked persons and its ID and the stored information is deleted. This measure not only allows a person to recuperate from brief occlusions or errors from the estimation algorithm, but also ensures that persons that, for instance already left the vehicle, are not considered as currently tracked/active. Moreover, it also prevents the propagation of false positives that derive from the permanent tracking of persons that are no longer detected. Additionally, this idea to remove persons after a given number of frames was inspired by a similar approach developed by [68].

    - Addressed in lines 9 and 10 of algorithm 7.

Additionally, in the first situation previously listed, a filtering process is also performed (lines 3-5 of algorithm 7) before the occurrence of the update step. This process is executed in order to discard keypoints with low confidence from a viable matched estimation in order to promote more stability/consistency of keypoint locations throughout consecutive frames. Moreover, this approach compares each joint (keypoint) confidence with its respective dynamic threshold, which is computed using the average confidence of all matched estimations in the current frame for that given joint. Therefore, if a given joint has a confidence value below its corresponding threshold, it will be replaced with the respective joint coordinates from the previous frame. The idea of using a dynamic threshold for each joint/keypoints was inspired by the work developed by [68]. In this article, the authors reported that an adaptive pruner is more effective for keypoint filtering

and to maintain/increase tracking performance, since not all joints have the same confidence level of estimation/detection [68]. The previous affirmation was corroborated during the development of the present pose tracking algorithm, by the observation of a lower average confidence level of some particular joints, when compared with the remaining keypoints.

Throughout the previous sections of this chapter, the main parts of the pose tracking algorithm implemented in the present dissertation were introduced and described in order to provide a clear insight of the work developed in the current project. In the following section, a comprehensive description of the evaluation process used to test and validate the implemented solution will be provided.

4.2 Evaluation

In order to efficiently test and visualise the results obtained using the several versions of the implemented algorithm, it was necessary, first, to adapt one of the test algorithms provided by Bosch. This adaptation consisted on:

1. Implementation of a test batch mode, in which it was possible to evaluate the whole subset (or each dataset individually) and obtain the partial performance results for each video.

2. Addition of the bounding boxes from each person to a given video, using functions from the OpenCV library [78], in order to complement the keypoint visualisation tool already implemented prior to this dissertation by Bosch.

3. Implementation of the MOTA metric, using the py-motmetrics library [79], and inference timers, using *perf_counter* function from Python time module, for the performance evaluation of the developed algorithm. As it happened with the keypoint visualisation, the mAP metric was already implemented by Bosch prior to the start of the present project.

Regarding the MOTA evaluation, thanks to the py-motmetrics library [79], which offers the ability to generate partial results during the evaluation process, it was possible to obtain a detailed report, not only containing the MOTA values, but also the number of matches (comprises both true and false positives), misses (false negatives), ID switches and false positives. The posterior analysis of this thorough performance report was crucial to understand how each method implemented in the proposed solution affected/improved its overall tracking performance. Lastly, all evaluation tests involving this metric were performed using the library embedded IoU calculator with the default maximum tolerable overlap distance of 0.5, i.e., pairs of boxes with IoU below 0.5 were considered as a non-matching pair.

Regarding the measurement of the inference time necessary to perform the computation of the tracking task, this was performed, as previously stated using the time module from Python. More specifically, the *perf_counter* function was placed both immediately before the start of the pose estimation process and exactly after the tracking task was completed. The difference between those

two timestamps was then compared with the benchmark difference obtained by measuring (using the same method previously described) the start and finish timestamps of the pose estimation process alone.

Lastly, all the preliminary, validation and evaluation tests were performed under the same conditions, using a development cluster equipped with a NVIDIA® Tesla V100 graphics processing unit (GPU).

4.3 Results

Following the description of the implementation process, and the consequent validation/evaluation procedure performed in the current dissertation, the next sections will focus on unveiling the main results achieved in the present work. Moreover, the key highlights of the progress made throughout the dissertation will also be presented in the following sections.

4.3.1 Metric performance

The performance results for the final algorithm and the two most relevant preliminary versions of the work performed in the present dissertation are shown in figures 4.2, 4.3 and 4.4. Moreover, the visual information depicted on those three figures is summarised in table 4.1, in order to aid in the numerical analysis of the performance results. Additionally, these results are shown in relative comparison with the benchmark results obtained using solely the pose estimation algorithm (depicted as blue bars in figures 4.2, 4.3 and 4.4). Furthermore, in this section, only the global results for the complete video subset are shown. The partial performance results for each of the three datasets that comprise the video subset are addressed in appendices A, B and C.

The first significant breakthrough in the improvement process of tracking performance from the benchmark results was the implementation of a linear Kalman filter [49] for motion prediction. Its respective performance results are depicted in figures 4.2 to 4.4 by orange bars. This method has been an ever-lasting presence in object tracking approaches and is still currently considered a relevant technique in this area, with application on recent articles such as [80, 81, 82, 83]. The Kalman filter can be used to predict/determine the location of an object/target, by taking into account the position of a particular object in the previous timestamp and the position measurement for the current timestamp. As it is visible in figure 4.2, the implementation of this methodology in the tracking module led to an increase in the MOTA metric, which translates in the increase of matches seen in figure 4.3 and in the decrease of ID switches and false positives in 4.4. Despite these improvements, the use of the Kalman filter also resulted in an increase on the number of misses and had no effect in the mAP metric. Although this was a good start to the improvement of the tracking module, this approach was abandoned in favour of a more data association focused solution due to two main reasons:

1. Lack of a model representative enough of the particularly dynamic system addressed in this work. Despite the improvement previously observed using a linear Kalman filter, this approach is far from ideal, mainly due to the incorrect assumption that the movements carried out by the human body are linear. On the contrary, these movements are highly non-linear and extremely unpredictable, which impairs the definition of an equation accurate enough to describe the system in question. Furthermore, the use of extended or unscented versions of the Kalman filter, which provide an approximation of non-linear systems to a linear behaviour, is also not ideal since they both require the prior knowledge of the system equation.

2. The much higher importance given by SOTA tracking approaches to the data association process (when compared with its motion prediction counter-part). As previously stated in chapter 2 (more specifically in section 2.2.4), most of the best performing approaches focus their efforts mainly in the improvement of the efficiency of their data association/assignment methodologies. This fact highlights the key importance of this type of approaches towards the tracking process and was highly regarded factor during the approach definition process (as aforementioned in chapter 3).

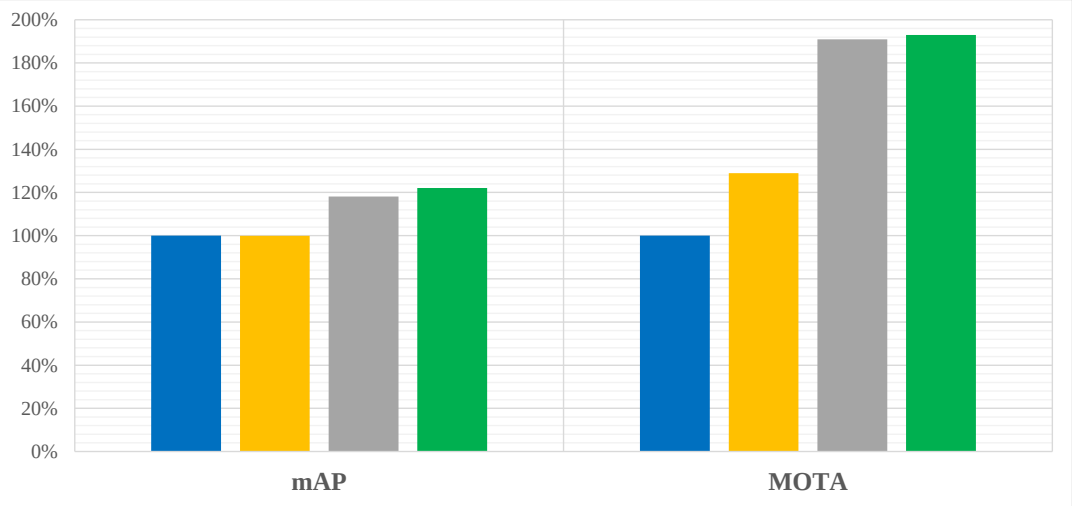

Figure 4.2: Subset (encompassing datasets #1, #2 and #3) performance results, regarding the evaluation metrics mAP and MOTA, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component.

Given the aforementioned reasons, the focus of the tracking solution shifted towards a more data association based approach, whose general pipeline was thoroughly described in the section 4.1 of the present chapter.

The performance results of this approach are shown in figures 4.2, 4.3 and 4.4, more specifically, by the depicted green bars. In the first figure, it is possible to observe a considerable mAP

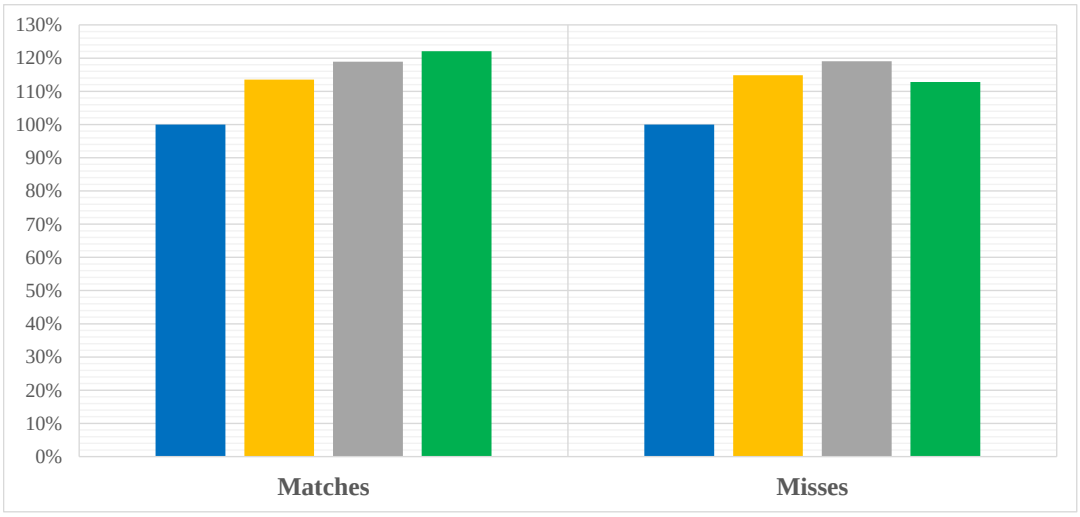

Figure 4.3: Subset (encompassing datasets #1, #2 and #3) performance results, regarding the number of matches and misses, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component. The total number of matches encompasses both true positives and false positives matches between GT bounding boxes and tracked bounding boxes.

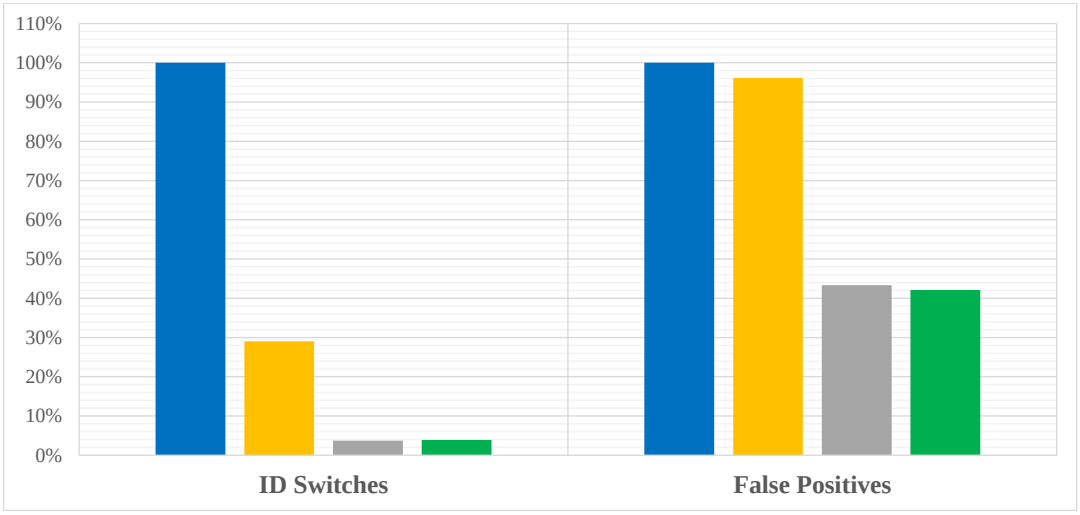

Figure 4.4: Subset (encompassing datasets #1, #2 and #3) performance results, regarding the number of ID switches and false positives, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component.

increase and a very significant boost in MOTA performance (almost a twofold increase comparing to the reference value). This tracking performance gain was mainly due to the considerable

decrease in ID switches and false positives (below 10% and 50% of the benchmark values, respectively). Additionally, the number of matches also increased, which, combined with the reduction of false positives cases, meant that the proposed algorithm was tracking more and with more accuracy than the benchmark and the previous Kalman filter based approach.

Despite the overall improvement of the final tracking algorithm, the number of misses suffered an unwanted increase (figure 4.3), similar to the behaviour previously reported for the Kalman-filter based approach (orange bars). This tendency may be explained, at least for the final algorithm, due to the "cautious" nature of this approach. In other words, the proposed final algorithm prefers/tends to promote a miss for a person in a given frame if the confidence levels of its matched estimation are not high enough, instead of generating a possible false positive for the same case. This behaviour is, as previously mentioned, enforced through the use of restrictive thresholds in different points of the tracking process. Moreover, another measure that may influence this increase in the number of misses is the handling of person IDs without matching for a given frame, since, although the keypoints are propagated, they are not visible during a frame with no matching and therefore, are considered a miss. Nevertheless, in most situations/videos, this last measure was seen yielding better tracking performance results than the case where keypoints are visible during a lack of estimation matching (data not shown).

Lastly, an approach combining both the data association based and the Kalman filter based algorithm was also tested and evaluated in terms of performance in an attempt to further improve the results obtained by the algorithm described in section 4.1. The performance results of the combined algorithm are present in figures 4.2, 4.3 and 4.4, depicted as grey bars. Despite the improvement shown by the two individual approaches separately, the conjunction of these two methodologies failed to yield better results than the approach solely based on data association. Nevertheless, these results prove the inadequacy of the linear Kalman filter for the present use case and corroborate the necessity of a method that is able to "learn" the system dynamics and model in order to efficiently implement a motion prediction methodology based on the Kalman filter.

Table 4.1: Subset (encompassing datasets #1, #2 and #3) performance results, regarding the two evaluation metrics (mAP and MOTA) and four partial metrics derived from MOTA, for the developed tracking algorithms using 1) Kalman filter (KF) only, 2) KF + data association (DA) and 3) DA only, relative to the benchmark values obtained using only the pose estimation algorithm (without any tracking component).

Metric \ Tracker	None	KF only	KF + DA	DA only
mAP	100%	100%	118%	122%
MOTA	100%	129%	191%	193%
Matches	100%	114%	122%	122%
Misses	100%	115%	114%	113%
ID switches	100%	29%	3.7%	3.9%
False positives	100%	96%	43%	42%

4.3.2 Visual analysis

Following the performance results previously presented, the aim of the present section is to provide a visual depiction and analysis of those improvements in the context of some of the subset videos used for algorithm evaluation. The visual comparison of the benchmark pose estimation algorithm with and without the implemented tracking algorithm is presented in figures 4.5 and 4.6, for two distinct scenes.

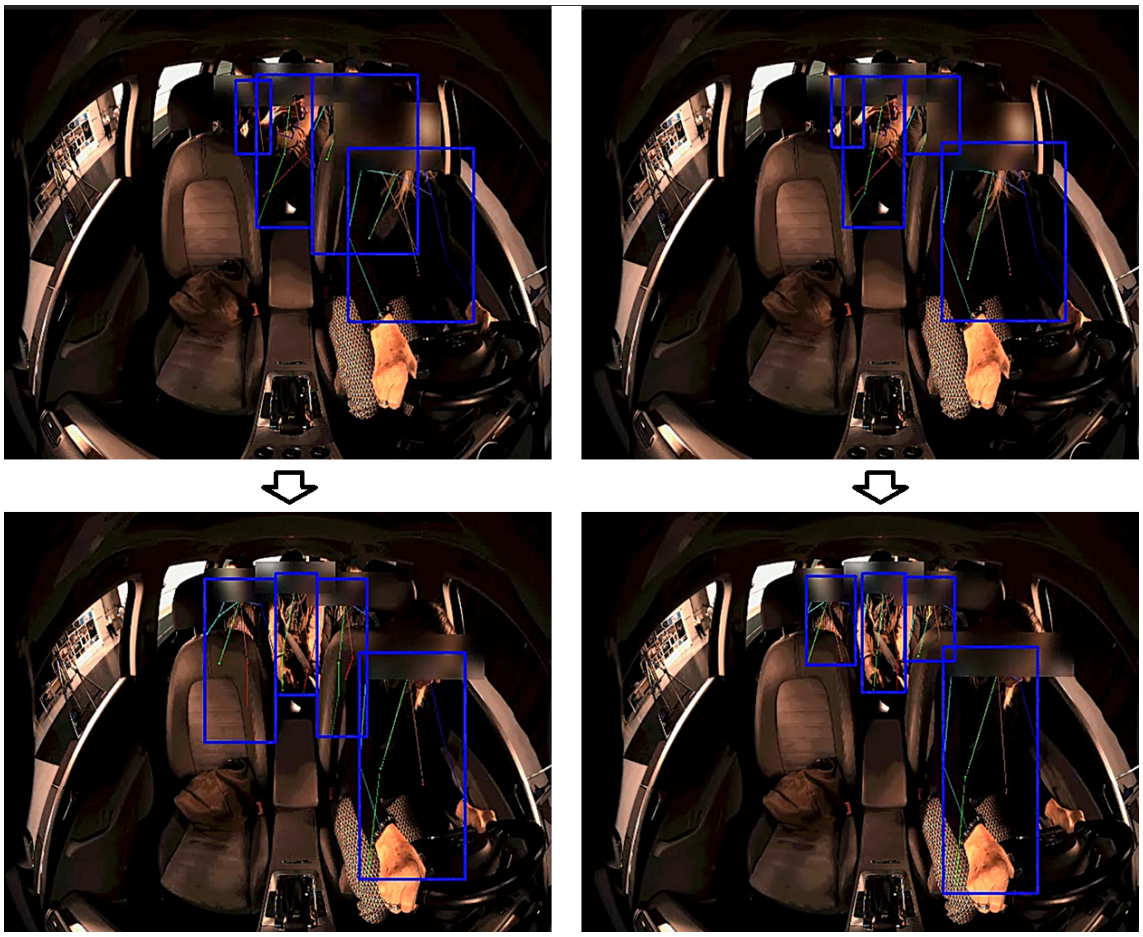

Figure 4.5: Visual comparison of the performance of the pose estimation algorithm without (left) and with (right) the proposed tracking algorithm in a representative video from dataset #1. The flow of the time-lapse (10 frames) is given by the arrows, i.e., for both cases (left and right), the top image represents the initial frame and the bottom image translates the scene evolution after 10 frames. Additionally, the blue outline rectangles depict the bounding boxes of each person identified by the corresponding algorithm.

In figure 4.5, it is depicted the evolution, for a short time-lapse, of an in-vehicle scene (recorded from a front row perspective) with four passengers (one in the front row and three in the second row) that interact with each other. In this case, the improvement provided by the tracking algorithm (figure 4.5, right) is mainly obtained through the removal of low confidence keypoints for the two persons seated in the left and right side of the second row (mostly occluded in this time segment). This allows the tracking algorithm to avoid propagation of bounding boxes with incorrect size

from the top frame to the bottom frame (figure 4.5, right), in contrast to the behaviour observed in the absence of this algorithm (figure 4.5, left). Additionally, this behaviour reduces the possibility of ID switches between persons, since there is less overlap between adjacent bounding boxes due to a more refined definition of their limits. One prominent example of the overlap area reduction is observed in the top frames of figure 4.5, between the right passenger from the second row and the passenger in the front row. In this case, the tracking algorithm is able to select only keypoints that are not occluded behind the seat (i.e. with high confidence), which translates into a smaller and more accurate bounding box for the occupant in the second row.

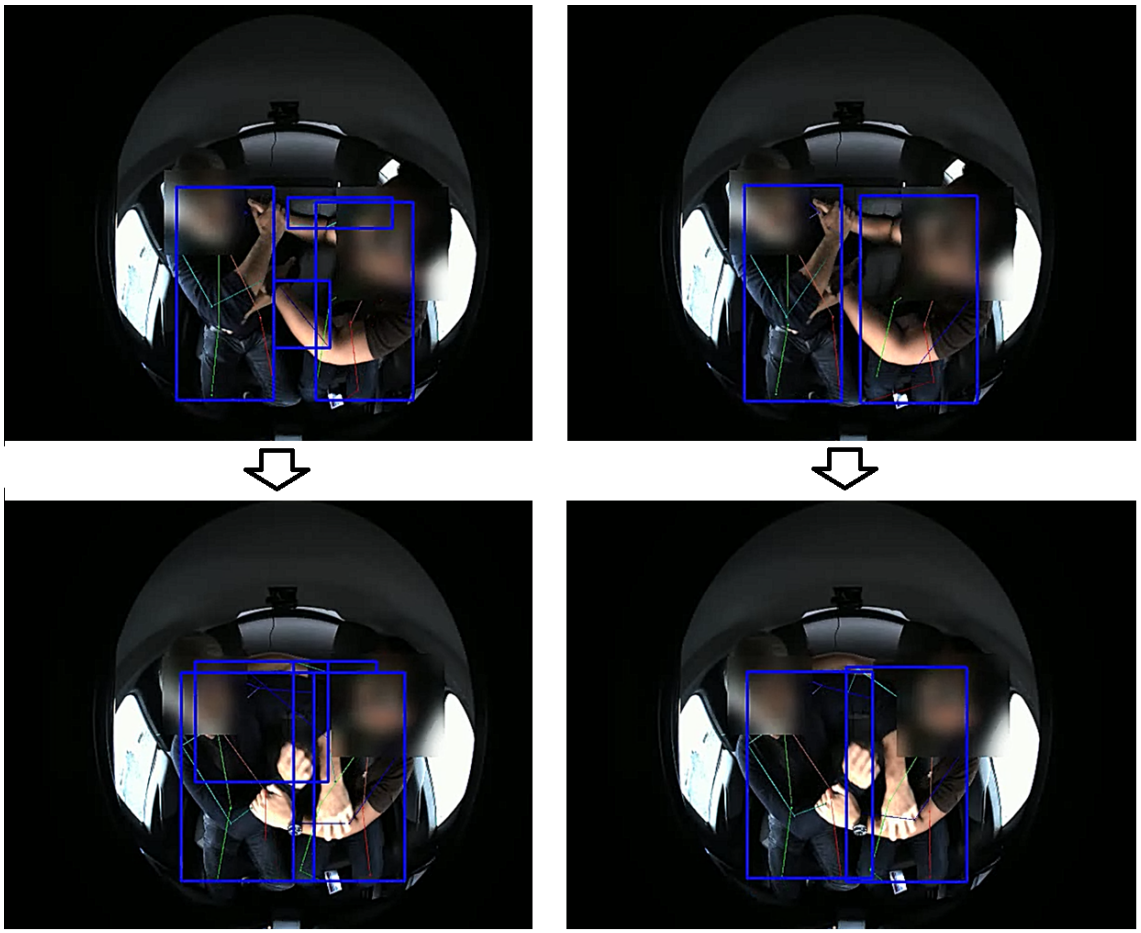

Figure 4.6: Visual comparison of the performance of the pose estimation algorithm without (left) and with (right) the proposed tracking algorithm in a representative video from dataset #3. The flow of the time-lapse (20 frames) is given by the arrows, i.e., for both cases (left and right), the top image represents the initial frame and the bottom image translates the scene evolution after 20 frames. Additionally, the blue outline rectangles depict the bounding boxes of each person identified by the corresponding algorithm.

The second visual example of the performance improvement provided by the tracking algorithm is presented in figure 4.6. Recorded from a different perspective (second row instead of front row), the second example introduced in this section depicts a negative interaction between two vehicle occupants, both seated in the second row. Similarly to the behaviour observed and stated

in the first example, the improvement of the tracking performance, by the proposed algorithm, in the scene depicted in figure 4.6 was mainly due to the removal of false positives and the yield of more stable and accurate keypoint locations. In this particular case, the reduction of the number of false positives was a direct consequence of a more efficient data association process that led to a decrease in person fragmentation (situation where parts of the same person are represented by different bounding boxes). These actions not only improve the overall tracking results of the pose estimation algorithm, but also provide a more accurate prediction of the real number of passengers present in the vehicle at a given time. This improvement allows action recognition methods (that receive the estimation outputs) to yield better decision results based on a more precise and much clearer input information.

4.3.3 Computation time

Upon validation of the proposed algorithm, and evaluation/confirmation of the tracking and non-tracking performance improvements that this approach was able to provide to the pose estimation process, the last evaluation test performed aimed at measuring its inference time. This evaluation was executed in order to assess if the first requirement described in chapter 3 was successfully accomplished. To that end, the inference time of the pose estimation module was compared with the inference time obtained for the pose estimation plus tracking modules. Additionally, this comparison was performed for videos containing one to five passengers in order to also assess the scalability of the tracking algorithm with the number of person IDs present in a given video. The computation results for the pose estimation algorithm alone (orange line) and with the proposed tracking algorithm attached to it (blue line) are shown in figure 4.7 (for visual inspection) and in table 4.2 (for numerical analysis).

Table 4.2: Variation of the inference time with the number of passengers present in a given video, for the pose estimation algorithm without any tracking component (None) and with the proposed tracking algorithm. The average values shown here are expressed in relation to the benchmark average results, obtained for the case of one passenger using the pose estimation algorithm without any tracking component.

Tracker / # Passengers	None	Proposed tracker
One	100%	105.5%
Two	104.6%	105.2%
Three	108.2%	108.4%
Four	115.0%	116.8%
Five	125.3%	126.5%

It is possible to observe, in figure 4.7, that both lines are dependent of the the number of passengers in a given video, i.e., the inference time increases with the increment in the number of passengers. However, this dependency is mainly due to the pose estimation algorithm, since both share a similar behaviour. Therefore, it is possible to assume that the pose tracking algorithm proposed and implemented in the present dissertation is invariant to the number of passengers

present inside a given monitored vehicle. Nevertheless, more tests are required to confirm this assumption/theory, for the specific use case analysed here but also for possible application outside a vehicle.

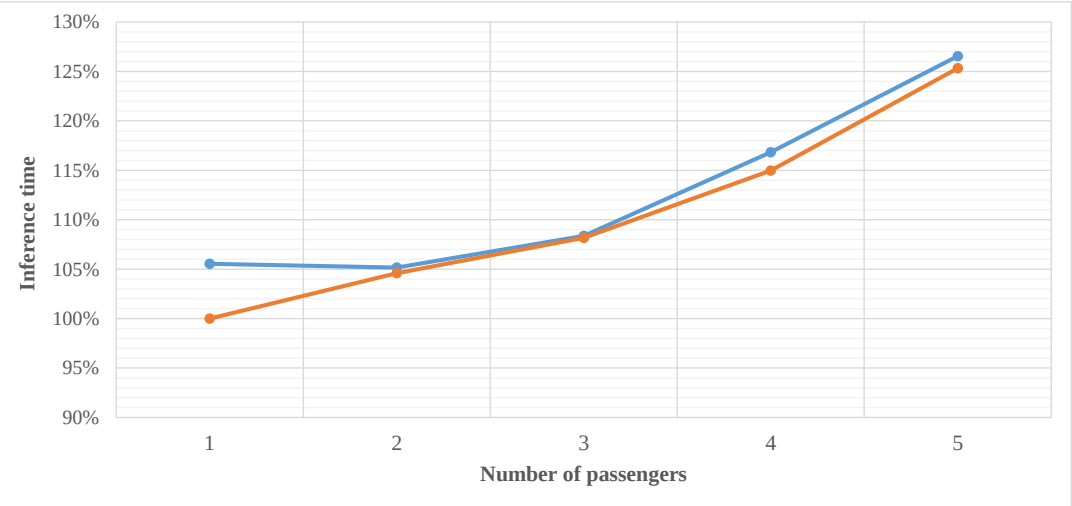

Figure 4.7: Variation of the inference time with the number of passengers present in a given video, for the pose estimation algorithm without any tracking component (orange line) and with the proposed tracking algorithm (blue line). The values shown here are expressed in relation to the benchmark results obtained for the case of one passenger using the pose estimation algorithm without any tracking component.

Chapter 5

Conclusions

The area of autonomous driving is a fairly new and fascinating field of research that is expected to grow and thrive in the next few years/decades. With the development of new and improved vehicles, comes also the necessity to create, improve or adapt several complementary technologies to ensure the safety and efficiency of those modes of transportation. One example of those technologies are the action recognition systems used to monitor the in-vehicle environment and to identify possible acts of violence that occur between passengers. This is an important research area, specially in the case of shared autonomous vehicles, given the lack of a human driver that could monitor/intervene to avoid potential dangerous situations and ensure the safety of the vehicle occupants. Most action recognition solutions resort to pose estimation algorithms to identify and characterise the human body pose of each person in order to then be able to extrapolate behavioural features, define the corresponding actions and classify those as violent or non-violent.

Despite the efficiency of most SOTA pose estimation approaches in single isolated images, they tend to struggle in scenarios comprised by videos (such is the case of surveillance and monitoring systems) due to the lack of context association between frames. One of the most commonly used solutions for this problem are the pose tracking algorithms, which provide the much needed temporal consistency to pose estimation methodologies through association and tracking of person estimations throughout consecutive frames. Therefore, the main objectives of the present dissertation were: 1) the familiarisation with the currently relevant and best performing techniques in the area of pose tracking and 2) the development and implementation of a solution, based on those techniques, that could provide an improvement in the temporal consistency of the pose estimation algorithm developed by Bosch. Moreover, this solution needed mainly to be computationally light, improve the tracking, as well as the non-tracking, performance of the pose estimation algorithm and promote coherence during the tracking process in order to improve the performance of subsequent action recognition algorithms.

Upon proposal, implementation and validation/evaluation of the pose tracking algorithm developed in the present dissertation, it is possible to conclude that the three aforementioned requirements were successfully fulfilled. Firstly, the inference time of the developed solution combined with the pose estimation algorithm yielded a fairly similar result to the reference time obtained

by the pose estimation algorithm alone. Additionally, the computation time of the tracking algorithm appeared to be invariant with the number of passengers, which means that it is possible for this solution to be scalable for scenarios with a higher number of targets. Secondly, the proposed tracking approach did improve the tracking performance of the benchmark estimation algorithm, which is reflected by the significant increase of the MOTA metric (translated by the reduction of the number of ID switches and false positives and the increment of the number of matches). Furthermore, it also improved the non-tracking performance, reflected by the mAP results, which is translated by the indirect increase of the estimations precision of keypoints locations. Lastly, the coherence of subsequent algorithms was also improved due to the aforementioned decrease in the number of ID switches and false positive, which are the two of the main sources of consistency impairment from estimation algorithms.

Beyond the fulfilment of the requirements previously mentioned, another positive aspect of the pose tracking algorithm developed in the present work is its modular nature, which means that the only dependence it has with the pose algorithm are the estimations yielded by the latter algorithm. This is a very helpful feature since any modification in the architecture and/or in the parameters of the pose estimation algorithm are inconsequential to the development and/or operation mode of the pose tracking algorithm and vice-versa. Furthermore, this level of freedom and independence may prove to be useful in the application of the developed tracking solution in the improvement of other pose estimation approaches and/or other implementations of the estimation algorithm used in the present dissertation.

All in all, the proposed and implemented pose tracking algorithm can be considered as a fairly solid and positive solution, in all aspects, for the problem of the particular use case addressed in the present dissertation.

5.1 Future work

Regarding future work, several aspects arise from the theme of the present dissertation that may require further investigation/development. The first aspect is the execution of more complete batch of comprehensive tests regarding the inference time of the proposed tracking algorithms. Although the invariance of its computation time with the number of passengers was proven with the test performed in chapter 4, its scalability for scenarios with a much higher number of estimations/targets remains to be assessed/confirmed.

The second aspect is the impact of the application of a Kalman filter with a more accurate representation of the system model, since the performance tests with a linear version of this method showed its ability to positively affect, although not as much as the data association component, the tracking performance of the estimation algorithm. Due to this fact, a more accurate representation of the dynamics seen in the various scenarios present on the video subset used for this particular case, may have the potential to further improve the final version of the proposed tracking algorithm. Approaches such as the ones proposed by [81] and [83], which resort to deep learning to obtain/learn information of the system dynamics and then use it to "tune" the Kalman filter, may be

the key to unlock the improvement potential of this technique for the particular use case reported here.

Following the work developed in the current dissertation, the last aspect that may have the most potential for improvement of the proposed tracking algorithm is the use of deep learning based methodologies. More specifically, the use of neural networks, either CNNs or RCNNs, which are already currently used in some SOTA approaches such as [10], [11] and [12], to extract visual information in the form of metrics (in the case of CNNs) or temporal context (in the case of RCNNs) to enhance the tracking process. In the case of visual metrics, these can be used to complement the affinity computation step and, consequently, improve the estimation assignment process, producing more accurate associations. On the other hand, the temporal context can be used in the data association process or to improve the detection and/or the tracking processes during scenarios with occlusion or with estimations errors that lead to misses.

Appendix A

Annex A: Dataset #1 results

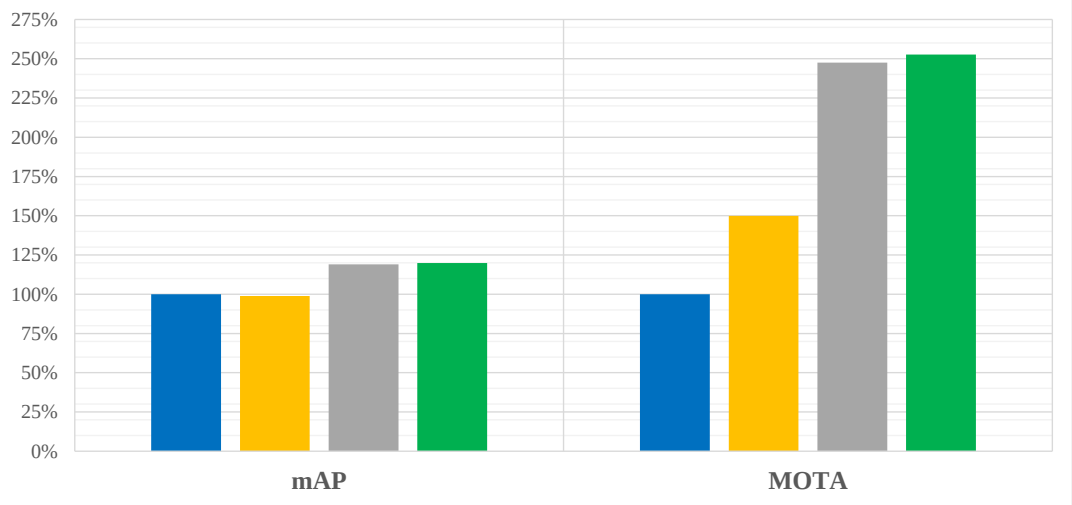

Figure A.1: Dataset #1 performance results, regarding the evaluation metrics mAP and MOTA, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component.

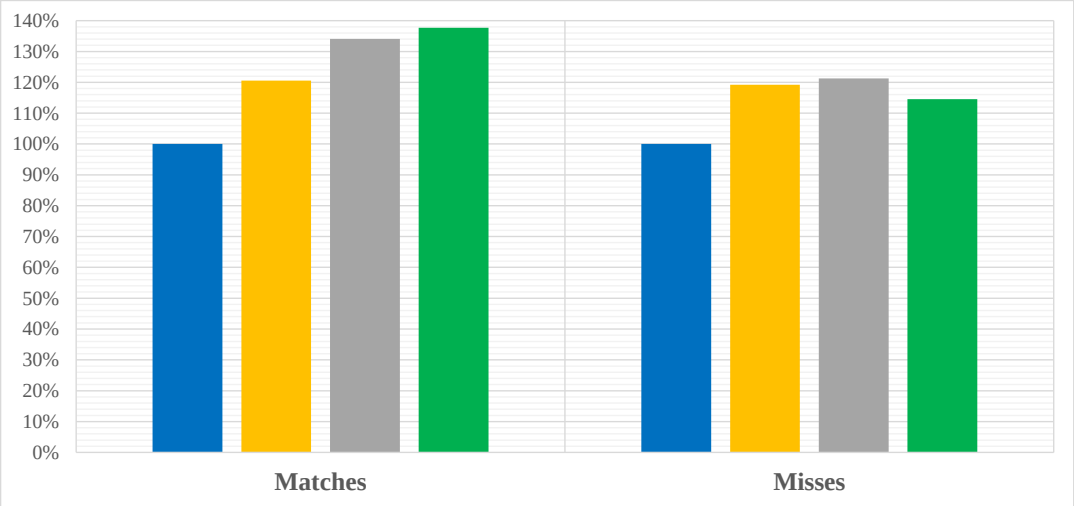

Figure A.2: Dataset #1 performance results, regarding the number of matches and misses, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component. The total number of matches encompasses both true positives and false positives matches between GT bounding boxes and tracked bounding boxes.

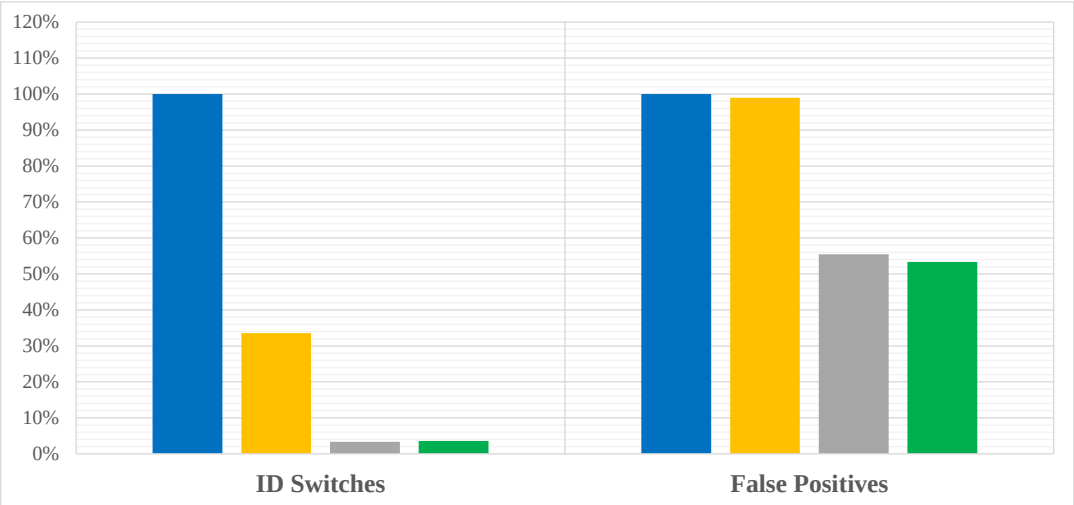

Figure A.3: Dataset #1 performance results, regarding the number of ID switches and false positives, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component.

Table A.1: Dataset #1 performance results, regarding the two evaluation metrics (mAP and MOTA) and four partial metrics derived from MOTA, for the developed tracking algorithms using 1) Kalman filter (KF) only, 2) KF + data association (DA) and 3) DA only, relative to the benchmark values obtained using only the pose estimation algorithm (without any tracking component).

Metric \ Tracker	None	KF only	KF + DA	DA only
mAP	100%	99%	119%	120%
MOTA	100%	150%	248%	253%
Matches	100%	121%	137%	138%
Misses	100%	119%	116%	115%
ID switches	100%	34%	3.3%	3.6%
False positives	100%	99%	55%	53%

Appendix B

Annex B: Dataset #2 results

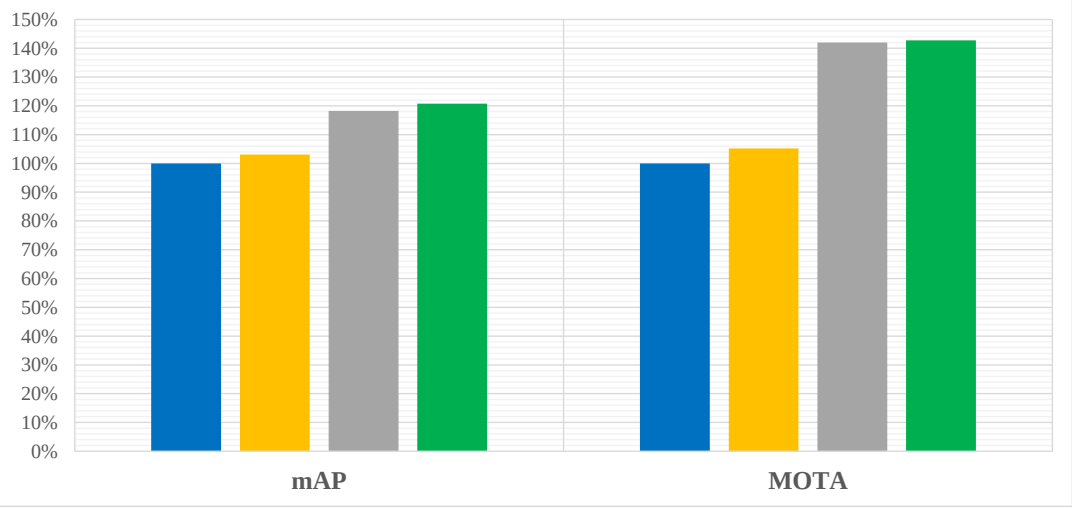

Figure B.1: Dataset #2 performance results, regarding the evaluation metrics mAP and MOTA, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component.

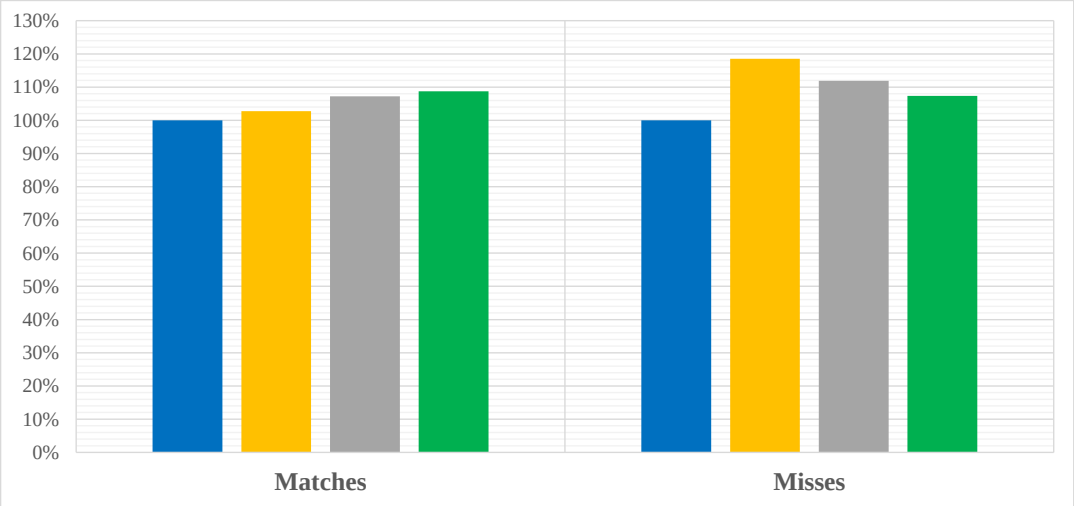

Figure B.2: Dataset #2 performance results, regarding the number of matches and misses, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component. The total number of matches encompasses both true positives and false positives matches between GT bounding boxes and tracked bounding boxes.

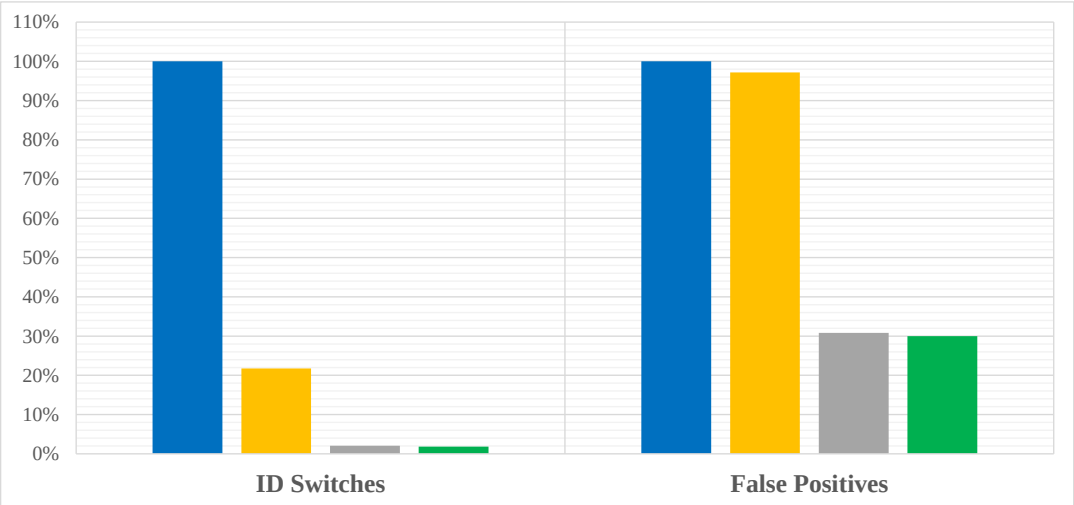

Figure B.3: Dataset #2 performance results, regarding the number of ID switches and false positives, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component.

Table B.1: Dataset #2 performance results, regarding the two evaluation metrics (mAP and MOTA) and four partial metrics derived from MOTA, for the developed tracking algorithms using 1) Kalman filter (KF) only, 2) KF + data association (DA) and 3) DA only, relative to the benchmark values obtained using only the pose estimation algorithm (without any tracking component).

Metric \ Tracker	None	KF only	KF + DA	DA only
mAP	100%	103%	118%	121%
MOTA	100%	105%	142%	143%
Matches	100%	103%	108%	109%
Misses	100%	119%	108%	107%
ID switches	100%	22%	2.1%	1.8%
False positives	100%	97%	31%	30%

Appendix C

Annex C: Dataset #3 results

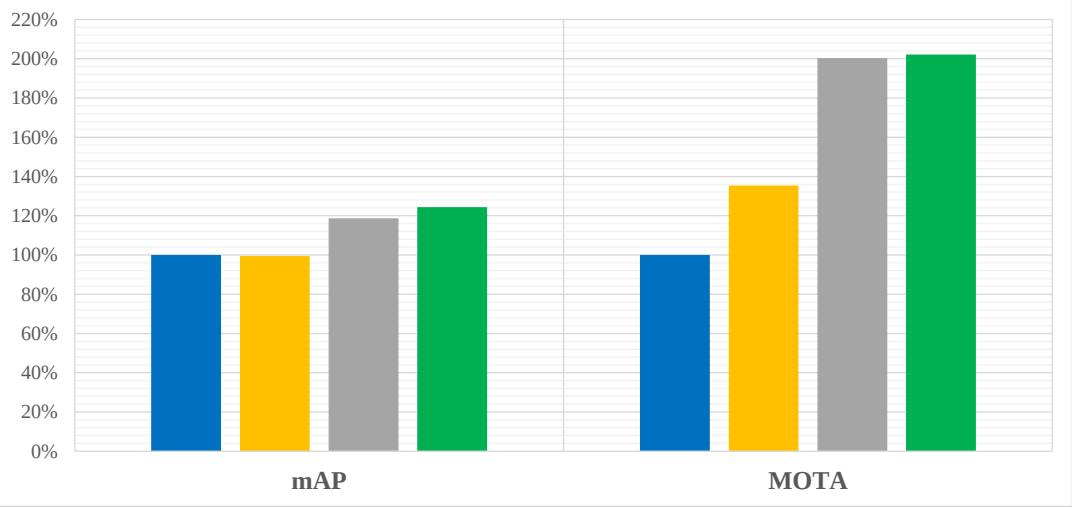

Figure C.1: Dataset #3 performance results, regarding the evaluation metrics mAP and MOTA, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component.

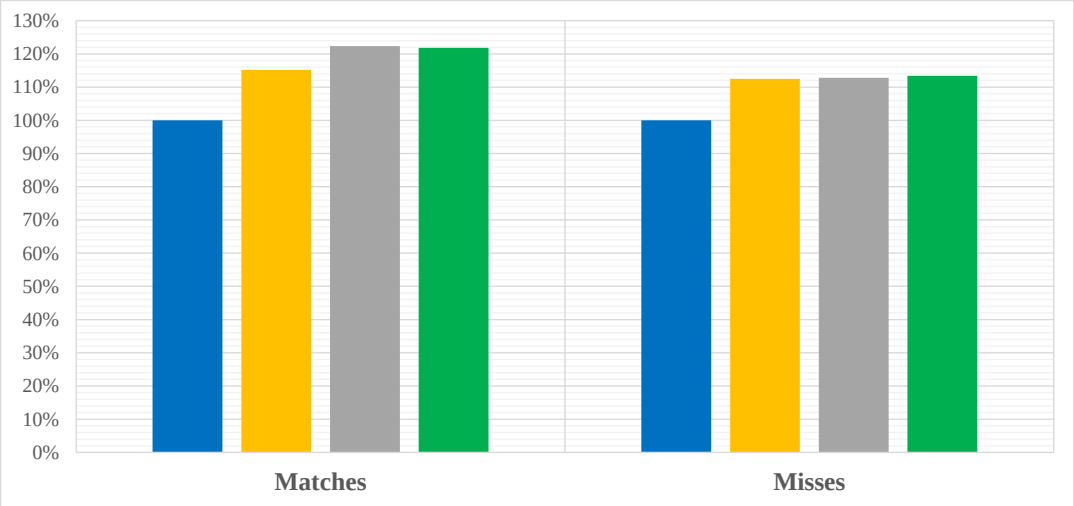

Figure C.2: Dataset #3 performance results, regarding the number of matches and misses, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component. The total number of matches encompasses both true positives and false positives matches between GT bounding boxes and tracked bounding boxes.

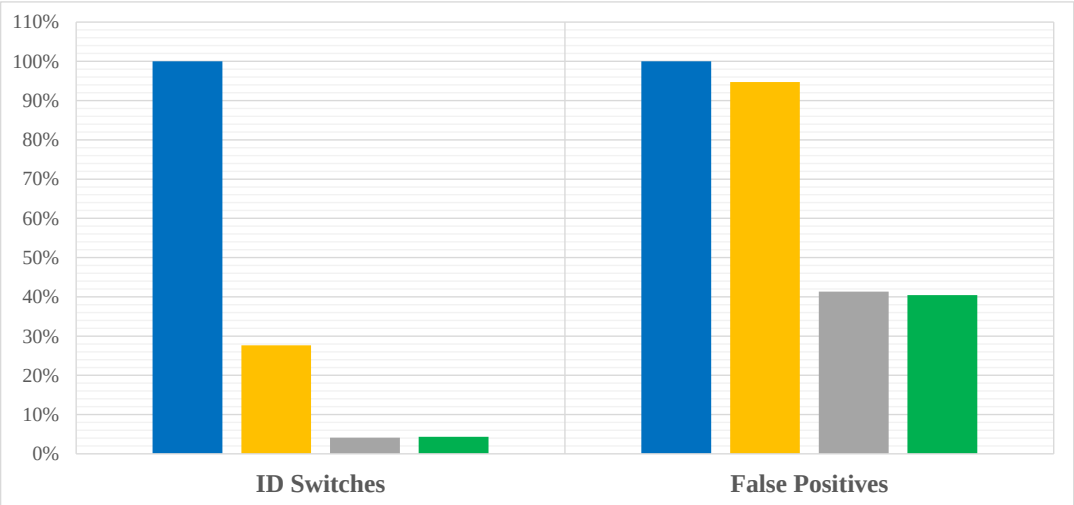

Figure C.3: Dataset #3 performance results, regarding the number of ID switches and false positives, for the Kalman filter only (orange), Kalman filter + data association (grey) and data association only (green) tracking algorithms. The values shown here are expressed in relation to the benchmark results obtained by solely using the pose estimation algorithm (blue), without any tracking component.

Table C.1: Dataset #3 performance results, regarding the two evaluation metrics (mAP and MOTA) and four partial metrics derived from MOTA, for the developed tracking algorithms using 1) Kalman filter (KF) only, 2) KF + data association (DA) and 3) DA only, relative to the benchmark values obtained using only the pose estimation algorithm (without any tracking component).

Metric \ Tracker	None	KF only	KF + DA	DA only
mAP	100%	99%	119%	124%
MOTA	100%	135%	200%	202%
Matches	100%	115%	122%	122%
Misses	100%	112%	114%	113%
ID switches	100%	28%	4.1%	4.3%
False positives	100%	95%	41%	40%

References

[1] COCO - Common Objects in Context website, (accessed: 15.06.2020). URL: http://cocodataset.org/.

[2] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22(1):67–92, Jan 1973. doi:10.1109/T-C.1973.223602.

[3] Bharath Raj. An overview of human pose estimation with deep learning, 2019. URL: https://mc.ai/an-overview-of-human-pose-estimation-with-deep-learning/.

[4] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression, 2017. arXiv:1711.08229.

[5] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, Mar 2020. URL: http://dx.doi.org/10.1016/j.neucom.2019.11.023, doi:10.1016/j.neucom.2019.11.023.

[6] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 01 2008. doi:10.1155/2008/246309.

[7] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation, 2019. arXiv:1902.09212.

[8] Manchen Wang, Joseph Tighe, and Davide Modolo. Combining detection and tracking for human pose estimation in videos, 2020. arXiv:2003.13743.

[9] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 keypoints is all you need, 2019. arXiv:1912.02323.

[10] Jihye Hwang, Jieun Lee, Sungheon Park, and Nojun Kwak. Pose estimator and tracker using temporal flow maps for limbs, 2019. arXiv:1905.09500.

[11] Yaadhav Raaj, Haroon Idrees, Gines Hidalgo, and Yaser Sheikh. Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields, 2018. arXiv:1811.11975.

[12] Andreas Doering, Umar Iqbal, and Juergen Gall. Joint flow: Temporal flow fields for multi person tracking, 2018. arXiv:1805.04596.

[13] MPII Human Pose Dataset website, (last accessed: 15.06.2020). URL: http://human-pose.mpi-inf.mpg.de.

[14] Mykhaylo Andriluka, Leonid Pishchulin, Peter V. Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[15] MOTChallenge website, (last accessed: 15.06.2020). URL: https://motchallenge.net.

[16] PoseTrack website, (last accessed: 15.06.2020). URL: https://posetrack.net.

[17] Helena Torres, Bruno Oliveira, Jaime Fonseca, Sandro Queirós, João Borges, Nelson Rodrigues, Victor Coelho, Johannes Pallauf, José Brito, and José Mendes. *Real-Time Human Body Pose Estimation for In-Car Depth Images*, pages 169–182. 04 2019. doi:10.1007/978-3-030-17771-3_14.

[18] Nikolaos Sarafianos, Bogdan Boteanu, Bogdan Ionescu, and Ioannis A. Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding*, 152:1–20, 2016.

[19] Naimat Ullah Khan and Wanggen Wan. A review of human pose estimation from single image. *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pages 230–236, 2018.

[20] Leonid Sigal. *Human Pose Estimation*, pages 362–370. Springer US, Boston, MA, 2014. URL: https://doi.org/10.1007/978-0-387-31439-6_584, doi:10.1007/978-0-387-31439-6_584.

[21] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. Deep network flow for multi-object tracking. 06 2017.

[22] Pedro Felzenszwalb and Daniel Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79, 01 2005. doi:10.1023/B:VISI.0000042934.15159.49.

[23] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732, June 2016. doi:10.1109/CVPR.2016.511.

[24] Michael Van den Bergh, Esther Koller-Meier, and Luc Van Gool. Real-time body pose recognition using 2d or 3d haarlets. *International Journal of Computer Vision*, 83:72–84, 06 2009. doi:10.1007/s11263-009-0218-0.

[25] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, June 2014. doi:10.1109/CVPR.2014.214.

[26] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2018. arXiv:1812.08008.

[27] Mihai Trăscău, Mihai Nan, and Adina Magda Florea. Spatio-temporal features in action recognition using 3d skeletal joints. *Sensors*, 19:423, 01 2019. doi:10.3390/s19020423.

[28] Muhammad Usman Khalid and Jie Yu. Multi-modal three-stream network for action recognition. *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3210–3215, 2018.

[29] J. Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. pages 4724–4733, 07 2017. doi:10.1109/CVPR.2017.502.

[30] Bin Wang Wenqing Zheng Qi Dang, Jianqin Yin. Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology*, 24(06):663, 2019. URL: http://tst.tsinghuajournals.com/EN/abstract/article_152468.shtml, doi:10.26599/TST.2018.9010100.

[31] Prakhar Ganesh. Human pose estimation : Simplified, 2019. URL: https://towardsdatascience.com/human-pose-estimation-simplified-6cfd88542ab3.

[32] Guanghan Ning and Heng Huang. Lighttrack: A generic framework for online top-down human pose tracking, 2019. arXiv:1905.02822.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[34] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking, 2017. arXiv:1710.10000.

[35] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:2878–2890, 2013.

[36] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. doi:10.1109/CVPR.2008.4587468.

[37] Zhihui Su, Ming Ye, Guohui Zhang, Lei Dai, and Jianda Sheng. Cascade feature aggregation for human pose estimation, 2019. arXiv:1902.07837.

[38] Hong Zhang, Hao Ouyang, Shu Liu, Xiaojuan Qi, Xiaoyong Shen, Ruigang Yang, and Jiaya Jia. Human pose estimation with spatial contextual information, 2019. arXiv:1901.01760.

[39] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[40] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[41] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation, 2017. arXiv:1708.01101.

[42] Mihai Fieraru, Anna Khoreva, Leonid Pishchulin, and Bernt Schiele. Learning to refine human pose estimation, 2018. arXiv:1804.07909.

[43] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping, 2016. arXiv:1611.05424.

[44] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation, 2016. arXiv:1612.00137.

[45] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields, 2016. arXiv:1611.08050.

[46] Eldar Insafutdinov, Mykhaylo Andriluka, Leonid Pishchulin, Siyu Tang, Evgeny Levinkov, Bjoern Andres, and Bernt Schiele. Arttrack: Articulated multi-person tracking in the wild, 2016. arXiv:1612.01465.

[47] Anton Milan, Hamid Rezatofighi, Anthony Dick, and Ian Reid. Online multi-target tracking using recurrent neural networks. 04 2016.

[48] Laura Leal-Taixé, Cristian Canton Ferrer, and Konrad Schindler. Learning by tracking: Siamese cnn for robust target association, 2016. arXiv:1604.07866.

[49] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960. URL: https://doi.org/10.1115/1.3662552, arXiv:https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf, doi:10.1115/1.3662552.

[50] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

[51] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, 1983.

[52] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109, doi:10.1002/nav.3800020109.

[53] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[54] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking, 2016. arXiv:1603.00831.

[55] Andreas Geiger, P Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: the kitti dataset. *The International Journal of Robotics Research*, 32:1231–1237, 09 2013. doi:10.1177/0278364913491297.

[56] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking, 2016. arXiv:1609.01775.

[57] Bo Wu and Ramakant Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 1:951–958, 2006.

[58] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking, 2015. arXiv:1504.01942.

[59] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles, 2019. arXiv:1903.05625.

[60] Phil Bergmann. Tracking without bells and whistles GitHub repository, (last accessed: 15.06.2020). URL: https://github.com/phil-bergmann/tracking_wo_bnw.

[61] Young-Chul Yoon, Du Yong Kim, Kwangjin Yoon, Young min Song, and Moongu Jeon. Online multiple pedestrian tracking using deep temporal appearance matching association, 2019. arXiv:1907.00831.

[62] E. Bochinski, T. Senst, and T. Sikora. Extending iou based multi-object tracking by visual information. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Nov 2018. doi:10.1109/AVSS.2018.8639144.

[63] Young-Chul Yoon, Abhijeet Boragule, Young min Song, Kwangjin Yoon, and Moongu Jeon. Online multi-object tracking with historical appearance matching and scene adaptive detection filtering, 2018. arXiv:1805.10916.

[64] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information [challenge winner iwot4s]. 08 2017. doi:10.1109/AVSS.2017.8078516.

[65] Rui Zhang, Zheng Zhu, Peng Li, Rui Wu, Chaoxu Guo, Guan Huang, and Hailun Xia. Exploiting offset-guided network for pose estimation and tracking, 2019. arXiv:1906.01344.

[66] E. W. Dijkstra. A note on two problems in connexion with graphs. *NUMERISCHE MATHEMATIK*, 1(1):269–271, 1959.

[67] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. arXiv:1409.1556.

[68] Guanghan Ning, Ping Liu, Xiaochuan Fan, and Chi Zhang. A top-down approach to articulated human pose estimation and tracking, 2019. arXiv:1901.07680.

[69] Miaopeng Li, Zimeng Zhou, Jie Li, and Xinguo Liu. Bottom-up pose estimation of multiple person with bounding box constraint, 2018. arXiv:1807.09972.

[70] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[71] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos, 2017. arXiv:1712.09184.

[72] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking, 2018. arXiv:1804.06208.

[73] Dongdong Yu, Kai Su, Jia Sun, and Changhu Wang. Multi-person pose estimation for pose tracking with enhanced cascaded pyramid network. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 221–226, Cham, 2019. Springer International Publishing.

[74] *Python*^{TM} website, (last accessed: 15.06.2020). URL: https://www.python.org/.

[75] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms – improving object detection with one line of code, 2017. arXiv:1704.04503.

[76] SciPy website, (last accessed: 15.06.2020). URL: https://scipy.org.

[77] SciPy linear_sum_assignment function web page, (last accessed: 15.06.2020). URL: https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.optimize.linear_sum_assignment.html.

[78] OpenCV website, (last accessed: 15.06.2020). URL: https://opencv.org/.

[79] Christoph Heindl. py-motmetrics GitHub repository, (last accessed: 15.06.2020). URL: https://github.com/cheind/py-motmetrics#py-motmetrics.

[80] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)*, Sep 2016. URL: http://dx.doi.org/10.1109/ICIP.2016.7533003, doi:10.1109/icip.2016.7533003.

[81] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric, 2017. arXiv:1703.07402.

[82] Caterina Buizza, Tobias Fischer, and Yiannis Demiris. Real-time multi-person pose tracking using data assimilation. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[83] Huseyin Coskun, Felix Achilles, Robert DiPietro, Nassir Navab, and Federico Tombari. Long short-term memory kalman filters:recurrent neural estimators for pose regularization, 2017. arXiv:1708.01885.