Pedestrian Models for Autonomous Driving
Part I: low level models, from sensing to tracking

Fanta Camara\textsuperscript{1,2}, Nicola Bellotto\textsuperscript{2}, Serhan Cosar\textsuperscript{3}, Dimitris Nathanael\textsuperscript{4}, Matthias Althoff\textsuperscript{5}, Jingyuan Wu\textsuperscript{6}, Johannes Ruenz\textsuperscript{6}, André Dietrich\textsuperscript{7} and Charles W. Fox\textsuperscript{1,2,8}

\textbf{Abstract—}Autonomous vehicles (AVs) must share space with human pedestrians, both in on-road cases such as cars at pedestrian crossings and off-road cases such as delivery vehicles navigating through crowds on high-streets. Unlike static and kinematic obstacles, pedestrians are active agents with complex, interactive movements. Planning AV actions in the presence of pedestrians thus requires modelling of their probable future behaviour as well as detection and tracking which enable such modelling. This narrative review article is Part I of a pair which together survey the current technology stack involved in this process, organising recent research into a hierarchical taxonomy ranging from low level image detection to high-level psychology models, from the perspective of an AV designer. This self-contained Part I covers the lower levels of this stack, from sensing, through detection and recognition, up to tracking of pedestrians. Technologies at these levels are found to be mature and available as foundations for use in higher level systems such as behaviour modelling, prediction and interaction control.

\textbf{Index Terms—}Review; Survey; Pedestrians; Autonomous Vehicles; Detection; Tracking; Trajectory Prediction; Interaction; Behaviour Models; Game Theoretic Models; Signalling Models; eHMI.

\section{I. INTRODUCTION}

Many organisations are vigorously developing autonomous vehicles (AVs). The technology for vehicles moving in static environments – localising, mapping, planning, and controlling – is well developed \cite{216} and is now available as open-source software \cite{112}. However, in real-world driving environments, human drivers regularly make decisions involving social decision-making that are harder to automate \cite{104}. Autonomous vehicles need additional social intelligence to operate in these complex social environments.

Interacting with pedestrians is a particular type of social intelligence. Autonomous vehicles will need to utilise many very different levels of models of pedestrians, each addressing different aspects of perception and intelligence from low level machine vision detection to high level psychological and social reasoning. Each of these models can be based on empirical science results or obtained via machine learning. So far, the required models have typically been developed by different research communities, so their integration is currently premature.

At the lower levels of the technology stack, pedestrian modelling requires perceptual methods to detect pedestrians, track their positions and velocities over time, and predict their movements to avoid colliding with them. These methods mostly originate from computer vision and robotics.

At the higher levels, as researched by psychologists and taught in advanced driver training programmes \cite{104}, drivers often try to infer the personality of other humans, predict their likely behaviours, and interact with them to communicate mutual intentions. Between the high and low levels, researchers infer psychological information from perceptual information, for example recognizing pedestrian body language, gestures, and demographics information, to better predict their likely goals and behaviours. Despite the importance of bridging the research between the higher and lower levels, their connection is still thin, both conceptually and in terms of actual implementations.

A promising method to bridge the higher and the lower levels is probability theory, providing possibilities for quantitative computational interfaces: for example, a pedestrian detector can pass a detection probability to a gesture recognizer, which computes probabilities of particular gestures based on this information, which in turn can be passed to a psychological or game-theoretic behaviour predictor, before the information is finally used to probabilistically compute optimal steering and speed values. Such a unified probabilistic stack requires models at all levels to realise quantitative, probabilistic inferences and predictions. Besides surveying the required building blocks, we also examine the maturity of each required level.

Many papers have been published presenting pedestrian models at various levels, but no unifying theory to connect them has yet been produced. The present study is Part I of a linked pair which together survey and unify the stack of required skills from engineered low-level aspects up to high-level aspects involving social decision-making. This Part I reviews the lower-level parts of the stack from sensing, through detection and recognition, to tracking, which together create the required inputs for higher level AI systems to control interactions reviewed in Part II \cite{29}.

Together, these two reviews contribute steps towards such a theory by bringing together, and organising into a new taxonomy, research from different fields including machine
vision, robotics, data science, psychology and game theory. We show how models from these fields could be linked together into a single technology stack by probability theory. Furthermore, we support this goal by summarizing methods for translating qualitative concepts into simple quantitative statistical models. Fig. 1 provides an overview of the main structure of the review.

The Society of Automotive Engineers (SAE) has defined five levels of driving automation, ranging from simple driver assistance tools to full self-driving [105] (cf. Table I). In our taxonomy, we approximately map requirements for pedestrian modelling to each of these levels, with requirements increasing as levels increase. Table II gives an overview of pedestrian models required for the SAE levels.

To reach SAE level 0, no automation is required, but some basic sensing is needed to inform the human driver. Very simple sensors can be used, such as the ultrasonic reverse parking sensors currently available commercially, together with very basic signal processing such as distance thresholds causing an audible signal. More complex concepts from our reviews may also be added to inform the driver of higher level information such as the identity of the particular pedestrian they are about to hit, but this is not necessary to reach level 0.

To reach SAE level 1, the AV needs to provide driving assistance tools such as lane keeping and Adaptive Cruise Control (ACC). To do this, it needs to detect the road structure and the surrounding objects and add these inputs to help the driver in his driving task. It needs to detect these objects in order to avoid them, but does not yet need to recognize them as specific individuals because this is not needed for obstacle avoidance.

To reach SAE level 2, the AV and the driver must share the driving task, with the vehicle taking full control of the vehicle at certain times. To take full control, it is not sufficient to only detect objects, but it is also necessary to recognize and track them over time in order to make short-term predictions of their motion and safely avoid them, passing control the to the human often, when these simple predictions do not work.

To reach SAE level 3, the driver can turn their attention away from the driving task, but must be prepared to take control occasionally within a certain time. This requires better prediction of pedestrian motion than level 2 in order to avoid asking the human for help as often. For example, adding concepts of likely routes and destinations to pedestrian models reduces the human intervention requests.

Finally, to reach levels 4 and 5, we believe that the AV must have a level of understanding of the driving task as good as a good human driver. Human drivers use complex psychology of pedestrian behaviour as well as their negotiating and signalling behaviours in their driving, so these must be replicated by the AV.

This Part I begins at the lowest levels of machine vision sensing (SAE level 0) and detection (SAE level 1), and considers recognition and tracking (SAE level 2) based on them. This Part I is intended to lay the foundations for Part II [29] which then moves up the stack to consider SAE levels 3-5. Part II also reviews data sources and other experimental resources useful for building and testing models at all levels.

Pedestrians are here defined as humans moving on and near public highways including roads and pedestrianised areas, who walk using their own locomotive power. This excludes, for example, humans moving on cycles, wheelchairs and other mobility devices, skates and skateboards, or those transported by other humans. This review does not cover interactions of traffic participants without pedestrians: a survey on trajectory prediction of on-road vehicles is provided in [131] and a survey on vision-based trajectory learning is provided in [153].

The review organizes research in pedestrian modelling into a new progressive taxonomy from relatively well understood quantitative engineering methods at the lower levels, towards less clear qualitative psychological theories of behaviour and interaction. It summarizes some progress in translating these qualitative concepts into simple quantitative statistical models, and identifies a strong need for this process to continue up the stack towards quantifying psychological, social, group and interactive models into algorithms for real-world AV control. Each section has an introduction and discussion which should be readable by researchers from other, especially neighbouring, fields who would like to get an overview of the state of the art and consider how their own field could connect both conceptually and computationally to it. Statistics on included papers are shown in the supplementary material Sect. I. The remainder of this Part I is organized as shown in Fig. 2.

II. PEDESTRIAN SENSING

Any pedestrian modelling system must begin by sensing data about pedestrians. Detection, tracking and higher level models may all depend on what information is present at this low level, so a brief review is provided here. More details are available in [36]. We classify our review into passive and active sensors. Active sensors send some signals into the environment that are reflected and detected while passive sensors receive information already present in the environment. A summary of common AV sensors with their range and accuracy is provided in Table III.

A. Passive Sensors

a) Manual Detection and Labelling: The most basic method of sensing pedestrians is to ask another human to
TABLE I

| SAE LEVELS OF AUTOMATION |
|---------------------------|
| **Level 0: No Automation** | Automated system issues warnings and may momentarily intervene, but has no sustained vehicle control. |
| **Level 1: “Hands on”** | The driver and the automated system share control of the vehicle. For example, Adaptive Cruise Control (ACC), where the driver controls steering and the automated system controls speed. The driver must be ready to resume full control when needed. |
| **Level 2: “Hands off”** | The automated system takes full control of the vehicle (steering and speed). The driver must monitor and be prepared to intervene immediately. Occasional contact between hand and wheel is often mandatory to confirm that the driver is ready to intervene. |
| **Level 3: “Eyes off”** | Driver can safely turn attention away from the driving tasks, e.g., use a phone or watch a movie. Vehicle will handle situations that call for an immediate response, like emergency braking. The driver must still be prepared to intervene within some limited time. |
| **Level 4: “Mind off”** | No driver attention is required for safety, except in limited spatial areas (geofenced) or under special circumstances, like traffic jams. Outside of these areas or circumstances, the vehicle must be able to safely abort or transfer control to the human. |
| **Level 5: Full automation** | No human intervention is required at all, fully automated driving. |

TABLE II

| PEDESTRIAN MODELS REQUIRED FOR SAE LEVELS |
|-------------------------------------------|
| **Models**                               | **Level 0: No Automation** | **Level 1: “Hands on”** | **Level 2: “Hands off”** | **Level 3: “Eyes off”** | **Level 4: “Mind off”** & **Level 5: Full automation** |
| Sensing | X | X | X | X |
| Detection | X | X | X | X |
| Recognition | X | X | X |
| Tracking | X | X | X |
| Single pedestrian: Unobstructed walking paths | X | X |
| Single pedestrian: Route prediction around obstacles | X | X |
| Single pedestrian: Uncertain destination models | X | X |
| Single pedestrian: Event/activity models | X |
| Single pedestrian: Effects of Effects of pedestrian class on trajectory | X |
| Pedestrian interaction: microscopic models. | X |
| Pedestrian interaction: macroscopic models. | X |
| Game theoretic and signalling models | X |

Fig. 2. Structure of the paper
do it. This may be used in low SAE level systems but also in offline studies, such as using trained human psychologists to conduct on-street surveys or annotate recordings of such surveys made with other sensors. Humans still have advantages over automated systems since they can use their full intelligence to subjectively annotate otherwise difficult events, such as the meanings of body language, emotions, and gestures. In particular, manual detection of pedestrians is needed and used as ground truth data for machine learning algorithms as in [245] where “experts” were asked to detect people as a baseline for a comparison against machine algorithms. In Camara et al. [30] human observers manually annotated the sequences of actions occurring during interactions between human drivers and pedestrians at an unmarked intersection.

b) Video: One of the most commonly used automated sensors is the video camera, being typically cheap and easy to install. For example, Zhang et al. [247] proposed a real-time pedestrian detector using vehicle cameras. A drawback of video sensing is that lightning conditions change colours and shadow due to weather and time of day. The human visual system is very good at compensating for these changes, but it is difficult instead to build machine vision systems able to perform similar compensation. For example in [252], the shadow of moving objects is removed from the foreground images in order to improve the accuracy of the detection. In [103], shadows are automatically removed from the images in HSV color space. On the contrary, Wang and Yagi [225] treated shadow as helpful information for their appearance-based pedestrian detector.

c) Stereo Pair Video Cameras: Traditionally, 3D machine vision was a higher-level research field than 2D image processing [98]. It uses two (or more) images from cameras, placed some distance apart, to estimate the stereo disparity between them and, from this, the 3D distance of the observed scene. Disparity describes the difference in location of corresponding features seen by the left and right cameras [210]. Disparity estimation methods fall into two classes: pixel-based methods (similar to optical flow), which estimates disparity at each pixel based on colour similarity to its neighbours; feature-based methods, which find a smaller number of statistically “interesting” points in the image (such as corners) and compute only their disparities. In recent years these algorithms have become standard and very fast hardware implementations [108] have enabled both real-time use and integration into consumer-style camera products. Hence, it is now possible to consider a “stereo camera” as a single device at the sensor level for human detection. For example, in [113], pedestrians are detected using dense (i.e. pixel-based) stereo camera images. Ess et al. [76], instead, implemented a stereo vision-based detection algorithm that extracts visual features and performs pedestrian detection from a mobile platform.

d) Passive Infrared Imaging: Pedestrians bodies radiate heat in the infrared (IR) spectrum, which may be easier to detect than the visible one. For example, Shetty et al. [198] proposed to use Grid-EYE, a thermal infrared sensor composed of multiple thermophile elements capable of measuring the temperature of static and moving people. Some infrared cameras may detect only a single frequency of light, or a small set (similar to R,G,B but selecting three other frequencies). In some cases, an IR channel is added to an existing RGB camera as a fourth channel. For example, Cieleniak et al. [48] presented a technique that combines color and thermal vision sensors data to track multiple people. Unlike visible light, IR does not allow to distinguish a single body from a group of pedestrians. This technology is dependent on weather conditions as IR may be absorbed by water, such as fog and rain, before reaching the sensor [201].

e) Passive Ultrasonic Sensor: When a moving object enters and then leaves the detection area, the sound energy increases and then decreases: the role of a passive ultrasonic sensor is to measure the produced acoustic energy [73]. This technique is not very reliable, as it might not be able to detect single moving objects from groups, and it is also dependent on weather conditions.

f) Piezoelectric Sensors: A review on tactile sensor detection of humans is provided in [61] and [215]. Piezoelectric sensors generate an electric impulse on touch contact, such as pedestrians stepping onto a sensed ground region, or making contact with an AV itself. This technique can become very expensive because it requires the installation of many piezoelectric sensors in the study area, for instance on the floor of the pedestrian infrastructure. It is useful as a last-resort sensor to detect actual collisions when other sensors have failed (i.e. safety bumper). In some limited (small but very high density) environments, it may be useful to monitor pedestrian movements around a sensor-filled floor. For example, Alwan et al. [4] presented a vibration-based floor to detect the fall of elderly people in a dwelling.

g) ID Sensors: These devices are attached to or carried by pedestrians and they transmit unique identifying tags as well as simplifying localization, and include infrared and RFID (Radio-Frequency IDentification) badges. Schulz et al. [194] developed a tracking system which combines ID sensor information with anonymous ones, such as LIDAR (see section II-B0a), in order to improve tracking accuracy. Versichele et al. [222] proposed to use Bluetooth for person tracking based on unique MAC addresses emitted continually by many personal devices already carried by pedestrians such as mobile phones. In [91], camera images are fused with omnidirectional (360 degree) RFID detection system using particle filter in order to enable a mobile robot to track people in crowded environments.

B. Active Sensors

a) Light Imaging Detection And Ranging (LIDAR): This sensor is mainly used for navigation and human detection purposes. It makes use of laser beams and calculates the distance to obstacles (objects, walls, people) by measuring the time gap between sending and receiving impulses. LIDARs can detect and classify different moving objects in the environment up to 360 degrees. It can be used to estimate pedestrian and/or vehicle traffic and determine the direction, speed and trajectory of the moving objects. For instance, Dewan et al. [68] presented a model-free detection and tracking of dynamic objects with 3D LIDAR data in complex environments. Objects are detected
and segmented thanks to multiple motion cues, then their estimated motion model is used for tracking. Arras et al. [7] proposed a similar supervised classifier to detect people using a 2D LIDAR. In this case, AdaBoost (Adaptive Boosting), a binary boosting algorithm that combines a set of weak classifiers into a strong classifier, is used to detect features of the laser beams corresponding to people’s legs in different environments. Gonzalez et al. [93] combined LIDAR and RGB camera data for pedestrian detection. The device can be used in any weather conditions but it can be quite expensive, especially when range longer than 30 m is needed [16]. Fig. 3(a) presents the working principle of a LIDAR and Fig. 3(b) shows the detection of road users with a LIDAR.

b) Radio Detection And Ranging (RADAR): This sensor was first used during World War II. Radars emit radiations from their antenna, which receives back the radiations reflected by passing objects. There are two types of radar: one which transmits continuous wave of constant frequency to determine the speed of moving objects based on the Doppler principle, where motionless objects are not detected [118]. The second type, FCWM (Frequency Modulated Continuous Wave), transmits a continuous changing frequency, which can detect static and moving objects [46].

c) Active Infrared Sensors: These sensors are composed of a transmitter that emits an infrared light in the direction of the detection area, a receiver that captures the reflected energy, and a data collection unit that measures the time of flight of the emitted infrared light. Objects speed can be detected by sending over two or more beams of infrared light. The range varies from a few to tens of meters. The Kinect sensor [248], a popular “depth” or “RGB-D” (red, green, blue, depth) camera, is a particular example of active infrared sensor. It uses a complex known pattern of thousands of rays and measures their movement in the reflected image to infer distance, similarly to a LIDAR. A review of computer vision techniques based on the Kinect sensor is proposed in [97].

d) Active Ultrasonic Sensors: They emit sound waves towards the detection environment. A detector senses the sound waves reflected by passing objects. This low cost sensing method is immune to lighting conditions and does not require significant maintenance. However, it can be seriously affected by weather conditions and it is typically not accurate enough for certain zones [201].

C. Discussions

Most autonomous vehicles today are using a mix of lidar and stereo vision. Visual RGB images are most commonly used as the base for detection, and feature-based localisation and mapping. Lidar or radar provide a more reliable but more expensive sensing capabilities for safety critical aspects such as collision avoidance. Traditionally, lidar was the main source of point cloud localisation and mapping but recent developments and price falls in millimeter radar and stereo cameras are making them increasingly competitive. Manual annotation of image data remains necessary for recognition of difficult detailed features such as pedestrian eye contact and body language meanings, but for other tasks even including the creation of training sets for machine learning, is now replaced by automated methods, including semi-supervised approaches which allow quite small manual training sets to be bootstrapped with much larger unannotated data.
III. PEDESTRIAN DETECTION MODELS

A previous review of pedestrian detection is presented in [72]. Here we summarize some of the key detection methods that are particularly relevant to AVs. Different techniques are used for detection, which can be classified into 6 main categories: visual appearance-based detection, motion-based detection, spatio-temporal features detection, 3D feature detection models, deep learning methods and attention windows detection. In computer vision, the detection problem can be viewed as a special case of image classification: given a candidate image window, the detection seeks to classify the latter as a pedestrian or non-pedestrian. The same concept applies to other types of sensors with their own detection “windows”. Fig 4 summarizes the sensing technologies and the pedestrian detection techniques described in this section.

A. Visual Appearance-Based Detection

Unlike motion-based methods, feature-based methods can operate with a single still image, as they look only for static patterns rather than changes over time.

a) HOG-SVM: One of the most commonly used pedestrian detectors is based on the combination of HOG (Histogram of Oriented Gradients) and SVM (Support Vector Machine). HOG [60] is a technique that was invented for the purpose of human detection (Fig. 5). After training, a classifier can determine whether a proposed HOG corresponds to a pedestrian or not. The OpenCV vision library [25] has a generic implementation of an object detector based on this method, which can be applied to pedestrian detection.

b) Alternative Features: Sometimes in place of HOG, alternative features including point descriptors, eg. BRISK (Binary Robust Invariant Scalable Keypoints) and SIFT (Scale Invariant Feature Transform), are used to detect characteristic features of an image such as corners, edges [190] [21]. Other forms of gradient features and edge detectors [33] are less sensitive to illumination compared to color descriptors. Texture features such as Local Binary Patterns (LBP) assign a class to each local window. Groups of classes in nearby windows can then be classified as pedestrians or non-pedestrians. Ahonen et al. [3] for example proposed a face recognition method based on the LBP feature descriptor. Paisitkriangkrai et al. [163] used LBP with spatial pooling for a robust pedestrian detection.

c) Segmentation Methods: These include methods such as the Mean-Shift clustering [28], watershed, and grab-cut, which divide the image into regions typically having similar or smoothly changing colour and texture characteristics. These regions can then be tested directly for pedestrians presence through shape, texture and other statistics as in [188] where people were detected and segmented based on a probabilistic method that describes the shapes of their different postures.

d) Cascade-based Detection: The detector proposed by [223] is composed of a sequence of classifiers, trained using Haar-like visual features, where each classifier can pass or not a sub-region to the following one. Zhu et al. [258] proposed a person detection method using a cascade (40 levels) of HOG-SVM detectors combined with Adaboost for feature selection. In [43], Chen et al. developed a person detection approach using a cascade classifier based on Adaboost with rectangle features and edge orientation histogram (EOH) features.

e) Deformable Part Model: Deformable Part Model (DPM) is a popular detection model. It has been originally proposed for the Pascal VOC challenge for object (including pedestrian) detection and recognition [77]. DPM splits an object into several parts arranged in a deformable configuration and can be used for pedestrian classification in [79]. This method can deal with significant variation in shape and appearance. A fast implementation of DPM applicable for person detection is proposed in [232].

B. Motion-based Detection

a) Frame Differencing: This method consists in computing the difference between the current frame and a reference one (usually the first frame). In [75], Elzein et al. developed a person detector using optical flow computed on regions selected by frame differencing on camera data recorded from a vehicle. The selected regions are then passed to a wavelet-based features classifier combined with template matching. Park et al. [166] proposed an approach that uses coarse-scale optical flow to stabilize camera frames with temporal difference features for pedestrian detection and human pose estimation, and tested on Caltech pedestrian benchmark [71].
b) Optical Flow: This technique assigns a direction and a velocity of motion to each pixel of two consecutive frames, as in [224]. Fernández-Caballero et al. [82] used optical flow and frame differencing for human detection on infrared camera images for a security mobile robot platform. Another use of optical flow for detection and tracking is proposed in [68] using 3D LIDAR data.

c) Background Subtraction: The method builds a background model used as a reference model in order to detect moving objects. This modeling is based on the assumption that the background is static. It consists in extracting an estimate of the background from the rest of the image by using some methods such as mean filter, running Gaussian average, etc. Background modeling has two variants: the recursive algorithm, which updates each frame with the estimate of the background, and the non-recursive algorithm, which stores a buffer with the previous frames and the background estimated from them. In [197], Sheikh et al. developed a background subtraction model that can detect human and objects in moving camera images. Their method builds background and foreground appearance models based on the background trajectory estimated by RANSAC algorithm.

C. Other Detection models

a) Spatio-Temporal Features: These features are commonly used in video codecs, such as Theora and h.264, because they have statistical properties of being the most efficient known summary descriptors of natural video. As such, they are also candidates for informative classification features. Oneata et al. [161] used these features with a supervoxel method for human detection in videos.

b) 3D Feature Detection: 3D feature detection models rely on 3D sensors such as depth cameras and 3D LIDARs. Depth information enables more robust detection algorithms. For example, the authors in [233] proposed an online learning method based on a 3D LIDAR cluster detector, a multi-target tracker, a human classifier and a sample generator. The cluster detection starts by removing the ground plane, then point clusters are extracted from the point clouds using the Euclidean distance in 3D space and finally a human-like volumetric model is fitted to the clusters for filtering. Yan et al. [234] took advantage of multiple (2D and 3D) sensor detectors to train an online semi-supervised human classifier for a mobile service robot. A depth-based person detector is presented in [151]. This detector applies template matching on depth images. To reduce the computational load, the detector first runs a ground plane estimation to determine a region of interest, which is the most suitable to detect the upper bodies of a standing or walking person. In [58], a mobile robot equipped with an RGB-D camera is used to detect people. Munaro and Menegatti [155] proposed a real-time detection and tracking system based on RGB-D camera data capable of detecting people within groups or standing near walls.

c) Deep Learning (DL): The 2D and 3D detection models presented so far rely on a two stage process of feature extraction followed by classification. In DL these are merged into a single step classifying the raw data directly using neural networks, which are organized in hierarchical models learned from data. These networks effectively learn their own feature sets in the lower layers of the hierarchy. ‘Deep learning’ is a contemporary term for hierarchical linear-in-the-parameters regression [94], [86], also known in previous decades as ‘neural networks’. These seek to minimise an error function between desired outputs $c_n$ and a model hierarchical linear-in-the-parameters function $F$, for inputs $x_n$ and parameters $\theta$,

$$E = \sum_n (c_n - F(x_n; \theta))^2.$$  \hspace{1cm} (1)

Hierarchical linear-in-the-parameters means that the model $F$ is comprised of layered hierarchy of ‘node’ functions of the form,

$$y_j = f(\sum_i w_{ji}z_i),$$  \hspace{1cm} (2)

where $f$ is any nonlinear function, $w_{ji} \in \theta$, and the outputs $y$ of one layer may form the inputs $z$ to the next layer. Such models have the special property that $E$ can be locally minimised by computing backpropagation terms for each node,

$$\Delta_j = \frac{\delta y_j}{\delta x_j} \sum_i \Delta_i w_{ji},$$  \hspace{1cm} (3)

then updating the parameters $w_{ji}$ as,

$$\frac{\delta E}{\delta w_{ji}} = \Delta_j(y_j).$$  \hspace{1cm} (4)

This algorithm dates from at least the 1970s [228] but has returned to popularity due to falls in prices of parallel hardware (specifically, graphics cards made by the company Nvidia), which enables to run larger models, together with algorithmic tweaks such as sharing weights (convolutional neural networks, CNN), pooling [127] and dropout [122] which exploit various symmetries and noise models relevant to the statistics of natural data. Angelova et al. [6] proposed a real-time pedestrian detector using deep network cascades. An advantage of the DL approach is that features can be learned, instead of manually “crafted”, which reduces errors on specific tasks rather than only on detection. For example, Ma [143] used an error function measuring correctness of prediction of pedestrian behaviors (as in later review sections, cf. Part II sections II and IV) to learn low-level visual features.

R-CNN [92] combines region proposals with CNNs in order to solve detection problem by operating within the “recognition using regions” paradigm. The system first takes an input image, extracts around 2000 bottom-up region proposals. Then, it computes features for each proposal using a large CNN. Finally, it classifies each region using class-specific linear SVMs. YOLO V3 [178] is a deep learning based software for object detection and classification. Unlike R-CNN, YOLO reframes object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. It is able to detect about twenty different classes including people, cats and dogs.

An extension of R-CNN is introduced in [180] by proposing a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling
nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. Mask R-CNN [99], extends Faster R-CNN by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression. The mask branch is a small fully convolutional network applied to each RoI, predicting a segmentation mask in a pixel-to-pixel manner.

**d) Attention Windows:** In their basic forms, the previous methods assume that every possible window of a 2D or 3D image will be tested for pedestrian detection. This can be computationally slow, unless the tests are performed in parallel (e.g. on a GPU) or some form of attention model is used to restrict the search. It is common to use a simple, fast and inaccurate detector set to have many false positives and few false negatives, to decide whether a window should be explored further or not [196]. In this case, a more advanced but slower method would be applied to test the most interesting windows. Prokhorov [174], for example, developed a road obstacle detector based on attention windows with potential application to pedestrian detection.

**D. Discussions**

Traditionally, a wide variety of image features have been developed by hand and matched with a wide variety of classifiers, to find good performance in pedestrian detection. Until recently, the HOG-SVM method was the best known [18]. Pedestrian detection, like most classification tasks, has however recently been revolutionized by price falls in parallel hardware such as GPUs, which have enabled classical neural network algorithms with small modifications (“deep learning”) to outperform hand-crafted methods for the first time. It seems likely that neural network methods will completely replace all others. The same GPU hardware also enables pixel-wise algorithms, such as optical flow, to be massively accelerated. They might not be necessary though if neural networks alone achieve the required performance.

The implementation of a person detection method for an AV is one of the major practical challenges. OpenCV\(^1\) library provides open-source implementation of many computer-vision algorithms (in C++ and Python), mainly aimed at real-time processing. It contains feature extraction methods such as HOG, SIFT, BRISK. It also includes a C++ implementation of DPM. In addition, LibSVM\(^2\) is a popular implementation of SVM classification algorithm. The Lidar-based leg detector in [7] is implemented as a Robot Operating System (ROS) module\(^3\). Again, the ROS implementation of the depth-based detector in [151] is implemented as a Robot Operating System (ROS) module\(^4\). The authors of the RGBD based detector in [155] provide the implementation of their algorithm\(^5\). Many DL-based approaches provide their code for reproducibility and comparison: YOLO\(^7\), R-CNN\(^8\), Faster R-CNN\(^9\), Mask R-CNN\(^10\).

The high performance of deep learning models comes at a price, they require larger training data (sometimes millions examples), longer training times (up to several days) and their computational cost is more important than for old-fashioned detectors e.g HOG-SVM [246]. In some cases, DL-based methods cannot reach real-time (i.e. 20-30 fps) performance [6] and are outperformed by a simple HOG-based detector [218].

**IV. PEDESTRIAN RECOGNITION MODELS**

While detection refers to finding the presence or absence of pedestrians at locations and scales in images, recognition here refers to the recognition of attributes of pedestrians given such detections. Recognition takes as input the localised window of visual or other sensor data forming the detection, and yields as output some information about the particular pedestrian detection. In some cases this could include their actual identity – identity recognition – but our use of the term here also includes recognition of attributes such as their body pose and facial features. Recognition refers to these tasks, while classification here refers to processes that perform recognition specifically by mapping inputs into discrete rather than continuous output classes. Figure 6 presents a set of attributes used for pedestrian recognition and a summary of the recognition models and papers reviewed in this paper is given in the supplementary material Sect. II.

---

\(^1\)https://opencv.org/
\(^2\)https://www.csie.ntu.edu.tw/~cjlin/libsvm/
\(^3\)https://github.com/wg-perception/people
\(^4\)https://github.com/strands-project/strands_perception_people/
\(^5\)https://github.com/LCAS/FLOBOT

---

![Fig. 6. Pedestrian attributes for recognition Models](image-url)
A. Recognition of body pose

While full-body tracking is discussed below, some methods may attempt to classify from single images some basic information on pose, such as the head direction of the pedestrian into facing AV / not facing AV. Where pedestrian body state is known – as resulting from skeleton and other body tracking – it may contain useful information about pedestrian’s goals and intentions, which may be extracted by classifiers operating at a higher level – on the tracked body configurations rather than on the raw images or sensor data. Cao et al. [34] [35] presented OpenPose, a real-time multi-person pose estimation software that uses CNNs to detect people in 2D images and Part Affinity Fields (PAF) is used to associate body parts to the detected people. Shotton et al. [200] developed a 3D human pose estimation method based on body parts representation. Their method, which relies on depth features and a randomized decision trees and forests algorithm for classification, outputs a proposal position for each detected body part. Their method is tested on motion capture data and synthetic data. Iqbal et al. [106] proposed a graphical model optimized by a linear integer programming (ILP) to estimate and track multiple people in videos that are made available as a new dataset called PoseTrack for multi-human pose estimation and tracking. Tompson et al. [217] combined a deep CNN with a Markov Random Field to estimate human pose from monocular images. Ma et al. [142] used a deep learning classifier to learn to recognize body heading direction. Manual annotations are not needed because a tracker gives the actual direction of movement which determines the heading. Kohari et al. [119] used a CNN model to estimate human body orientation for a service robot.

B. Recognition of head direction

Head direction’s primary use in pedestrian-AV interaction is epistemological: a pedestrian facing the AV – and/or establishing direct eye contact with it – is a good indicator that the pedestrian has seen the AV and knows it is there, and therefore will be planning their own behaviors on the assumption that they will have to interact with it. In contrast, a pedestrian who has not seen the AV, unless relying on auditory cues, may just step into the road with no idea that a potentially dangerous interaction is about to occur [229] [14]. Darrell et al. [62] developed a real-time human tracking and behaviour understanding system, called Pfinder. The system converts human head and hands into statistical model of color and shape in order to deal with different viewpoints. Schulz and Stiefelhagen [193] estimated pedestrian head pose (angle between -180 to +180 degrees) using multi-classifiers for different monocular grayscale images, depth information within the detection bounding box is also taken into account. Flohr et al. [84] proposed a model that can detect pedestrian body and head orientation from grayscale images based on a pictorial structure method. Microsoft Azure Face API [146] is a product that can report about gender, age, smile, body pose, head pose, facial features and wearing glasses.

C. Recognition of gaze direction and eye contact

Algorithms for gaze tracking and eye contact detection are not yet robust, and in laboratory eye tracking experiments require expensive precision equipment to be installed on the subject’s heads.

Benfold and Reid [19] proposed a method which infers the gaze direction from a head pose detector based on HOG and colour features. The head pose is classified using randomised ferns, i.e. similar to decision trees, and the tracking is done frame-by-frame based on the head detector using multiple point features. Baltrusaitis et al. [11] developed OpenFace, an open-source software that can run in real-time and used with a simple webcam. This tool is suitable for facial behavior analysis, in particular for facial landmark detections, head pose estimation, facial action recognition and eye-gaze estimation.

D. Emotion recognition

Pedestrian emotion recognition might be useful to inform about their crossing intention. For example, an angry pedestrian might be more likely to behave more assertive in crossing the road in front of an AV.

Cornejo et al. [56] and [55] developed a facial expression recognition method that is robust to occlusions. The occluded facial expression is reconstructed with a Robust Principal Component Analysis (PCA) method, facial features are extracted using Gabor wavelets and geometric features in [56] and using CENTRIST features in [55]. Then a dimensionality reduction of the feature vector is performed via PCA and LDA, finally recognition is performed with KNN and SVM as classifiers. Cambria et al. [32] proposed a new categorization model for emotion recognition systems and [31] reviewed sentiment analysis methods. Poria et al. [172] developed a CNN model with a convolutional recurrent multiple kernel learning (CRMLK) that can extract features from multimedia data such as audio, videos, and text. The method has been tested on Youtube videos and ICT-MMMD dataset. Horng et al. [101] presented a real-time time driver fatigue detection method which relies on eye tracking and dynamic template matching. Den Uyl and Van Kuilenburg [66] developed the FaceReader, an online facial expression recognition system, which is robust to the head pose, orientation and lighting conditions.

E. Recognition of pedestrian identity for re-identification

Person re-identification (re-ID) is the problem of recovering the identify of the same person with different clothing across different images, under different camera views, weather, lighting, and other environmental conditions. Ahmed et al. [2] developed a deep convolutional network that solves the re-identification problem by computing a similarity value between two image pairs. Their method has been tested on CUHK01, CUHK03 and VIPER datasets. Zheng et al. [253] proposed a person re-identification method based on the Bag-of-Words (BoW) model which extracts Color Names descriptor features from the input image, a Multiple Assignment (MA) is then used to find neighboring local features and finally
F. Gesture recognition

Deliberate gestures are the most obvious form of communication from body pose. For example a pedestrian may wave a vehicle on to show that they intend to give it priority in a crossing. A previous review on hand gesture recognition is provided in [150] and more recently Rautaray and Agrawal [177] presented a survey for interactions with a computer. Chen et al. [40] used a real-time tracker with Hidden Markov Models to recognize hand gestures. Freeman and Roth [87] used orientation histograms for gesture recognition. Their real-time method can recognize about 10 different hand gestures. Ren et al. [181] developed a robust hand gesture recognition system for active infrared (Kinect) sensors. Their method is based on template matching for part-hand gesture recognition and a new distance metric called Finger-Earth Mover’s Distance (FEMD) is used to measure the similarity between two hand shapes. Other gesture recognition methods based on hidden Markov models (HMM) are proposed in [24] [129].

G. Body language recognition

In addition to deliberate gestures, unconscious body language including stance and gait (walking style) may also be a predictor of pedestrian assertiveness in interactions, and of other behaviours. As with gesture recognition, body language recognition relies on recognition of body pose, followed by classification of this pose. Quintero et al. [175] proposed a Hidden Markov Model for pedestrian intention recognition based on 3D positions and joint displacements along the pedestrian body. In [226] a human gait recognition method is proposed, combining background subtraction with PCA for dimensionality reduction. A supervised pattern classification is finally performed to recognize the gait.

H. Activity recognition

Pedestrian activity recognition is of particular importance for autonomous mobile platforms. A lot of work is ongoing for service robots and AVs. Chaaraoui et al. [37] used contour points of human silhouette to do human action recognition for real-time scenarios. Dollar et al. [70] used spatio-temporal features for both human and rodent behaviour recognition. Vail et al. [219] compared hidden Markov models to conditional random fields for human activity recognition. In [137] a coupled conditional random field is used with RGB and depth sequential information. Coppola et al. [54] developed one of the first RGBD-based social activity recognition for multiple people. Their method learns spatio-temporal features from skeleton data which are fused using a probabilistic ensemble of classifiers called Dynamic Bayesian Mixture Model (DBMM). A more complete review on human activity recognition methods is proposed in [69].

I. Discussions

AVs need to recognize pedestrian attributes including pose and identity to help them make more accurate predictions about pedestrians likely future behaviours. Detection of pedestrians – i.e. determining the presence or absence of a pedestrian in an image window – is now mature technology, but recognizing the attributes of these pedestrians within these detections, such as body pose, is a harder and still open research area. Eye direction and eye contact remains particularly difficult as it requires very precise estimation of the positions of a very small pupil and iris at a long distance. Humans have evolved to be particularly good at recognizing gaze direction for social purposes but it is hard to automate. Recognition of emotions may be useful to inform predictions of pedestrians likely behaviours (e.g. an angry pedestrian may be more likely to push in front of us), and progress has been made in this area in offline systems such as social networks. But again, recognition at distances and speeds encountered by AVs remains challenging and open. It is likely that neural network approaches will come to dominate this area as with detection.

Open-source implementations of pedestrian recognition models include Openpose[11] for pose estimation, OpenFace[12] for head pose and eye-gaze estimation and OpenTrack[13] for head tracking. To our knowledge, there is no generally accepted single standard benchmark for pedestrian recognition models. Future research should thus explore the performance and computational efficiency of pedestrian recognition models in the context of autonomous driving.

Recognition of any attributes which enables recovery of a pedestrian’s name or other formal identification will fall under data protection laws in most jurisdictions, such as the GDPR across the EU. While re-ID might be particularly useful, for example for use in delivery robots to confirm recipients identities, the usage of this technology should be carefully assessed in regards to data privacy. The other recognition and tracking algorithms mentioned in this section extract features anonymously, i.e. extracted data does not allow the identification of individuals. Re-ID on the other hand can be used to record and store sensitive personal data, which yields the potential to be misused for public surveillance. For AVs centralized re-ID might be useful to link individual traffic participants to their previously observed behavior in traffic enhancing long term path prediction but at the cost of severe intrusion into the privacy of road users. This will raise a host of ethical and legal issues when such accuracy is reached by rapidly accelerating machine vision research, such as resale of data about individual’s locations and behaviours to insurance and advertising companies, or use by local authorities or law enforcement agencies [86].

V. PEDESTRIAN TRACKING MODELS

Pedestrian tracking is the process of updating belief about a pedestrians location from a temporal sequence of data. A pedestrian track is a sequence of a pedestrians locations over

11https://github.com/CMU-Perceptual-Computing-Lab/openpose
12https://cmusatyalab.github.io/openface/
13https://github.com/opentrack/opentrack
time. A pedestrian pose track is a sequence of a pedestrians body pose states over time. When multiple pedestrians are present, tracking requires knowing which pedestrian is which over time, associating the identities of the pedestrians with tracks. This is a challenging problem for humans if their tracks overlap or disappear behind obstacles, and appears to require high level social intelligence and knowledge to guess what most likely happened when tracks are temporarily hidden.

Pedestrian tracking plays an important role in many commercial applications such as security monitoring [141] and is also required in autonomous driving. To enable high level predictions pedestrian behaviour we first need to know which pedestrian is which, over time, and be able to see their historical tracks to use as evidence in our predictions of future behaviour. Tracking usually consists of two steps: a prediction step to determine several likely next possible pedestrian states, then a correction step to check each of these predictions and select the best one, to give an updated estimate of the new state. Tracking of pedestrians often requires the estimation of non-linear, non-Gaussian problems due to human motion, pedestrian scales, and posture changes. Tracking is a challenge for AVs because of the multiple uncertainties (e.g. occlusions etc.) due to complex environments. More generally, tracking consists in determining the position (and possibly orientation or velocity) of a given object at any time. Many techniques have been employed for pedestrian tracking in particular, and some are listed in [96] and [204]. For more details on the subject, Bar-Shalom et al. [13] presented state estimation algorithms and how they could be applied to tracking and navigation problems. Figure 7 summarizes single pedestrian tracking models.

Previous reviews on tracking methods for pedestrians can be found in [132] [165]. In this section, we first review two classes of methods for single pedestrian location tracking relevant to autonomous vehicle interactions (as previously classified by Owczarek et al. [162]): point tracking and kernel-based tracking. We then review recent work in the more challenging tasks of body pose tracking and multiple pedestrian tracking. A summary of the tracking methods and papers reviewed in this Part I is provided in the supplementary material Sect. III.

A. Single pedestrian point tracking

Point tracking typically relies on probabilistic methods based on Bayes Filtering [207]. Based on Bayes rule (eq. 5), the filter is composed of an initial state, a prediction step and a correction step. The initial state (eq. 6) presented the initial belief about the system; the prediction step (eq. 7) consists in estimating the state of the system with the input data; finally, the correction step (eq. 8) updates the state estimate with sensor measurements [185], [216].

Bayes rule:

$$ p(x_t \mid z_t) = \frac{p(z_t \mid x_t)p(x_t)}{p(z_t)} $$

(5)

Initial belief:

$$ bel(x_0) = p(x_0) $$

(6)

Prediction:

$$ bel(x_t) = \int \eta \cdot p(x_t \mid u_t, x_{t-1}) \cdot bel(x_{t-1}) dx_{t-1} $$

(7)

Correction:

$$ bel(x_t) = \eta \cdot p(z_t \mid x_t) \cdot bel(x_t) $$

(8)

The transition probability $p(x_t \mid x_{t-1})$ is of crucial interest as it provides the mathematical bridge from low to high level pedestrian behavior models. In its lowest form – the standard Kalman filter – it may simply be a Gaussian with zero mean and variance set to model the scale of a (literal) random walk by the pedestrian. But we may have much more predictive information $\theta$ about the pedestrian behavior to form $p(x_t \mid x_{t-1}, \theta)$. Here $\theta$ could include mid-level information such as the pedestrian’s pose, heading, and location on a map. For example, if the pedestrian is standing at the edge of the road, he/she is more likely to wait and cross. Information about the pedestrian’s origin and destination could also help to predict the future trajectory. Further information about beliefs, intentions and desires of the pedestrian will also modify the trajectory probability. The transition probability thus provides the interface where all higher level models, discussed later in Part II, will link to low-level pedestrian models. The following are some of the most popular variants of Bayesian Filtering used for pedestrian point tracking:

a) Kalman Filter: A Bayes filter applied to linear systems with continuous states – i.e. its state transition probability $p(x_t \mid u_t, x_{t-1})$ (eq. 9) depends on a linear model with Gaussian noise $\epsilon_t$:

$$ x_t = A_t x_{t-1} + B_t u_t + \epsilon_t $$

(9)

where $A_t$ is the matrix that describes the evolution of the state $x_t$ up to time $t$ without controls or noise and $B_t$ is the matrix that describes how the control $u_t$ changes up to time $t$.

Also the measurement probability $p(z_t \mid x_t)$ (eq. 10) depends on a linear model with Gaussian noise $\delta_t$ as,

$$ z_t = C_t x_t + \delta_t $$

(10)

where $C_t$ is the matrix which describes how to map the state $x_t$ to an observation $z_t$. 

---

**Fig. 7. Single pedestrian tracking models**
The prediction step (control update step) increases the uncertainty in the robot’s belief, while the measurement update step decreases it.

b) Extended Kalman Filter (EKF): This EKF is derived from the Kalman Filter and approximates non-linear models via Taylor expansion. EKF is a tracking technique well performed in scenarios where there are few changes but it has a computational cost that could be not neglectable due to the linearization process, which can involve the calculation of big Jacobian matrices. One of the limitations of EKF is that the linearization decreases the certainty about the system and therefore the pedestrian tracking performance [17]. The Multi-Hypothesis Extended Kalman Filter (MHEKF), which is an EKF on multivariate Gaussians, tries to solve this problem. In [64], for example, a CNN detector is combined with an MHEKF for vehicle tracking using low resolution LIDAR data.

c) Unscented Kalman Filter (UKF): This is another Kalman Filter that avoids the linearization problem by a second-order approximation, called the Unscented Transformation. It approximates a probability distribution with chosen weighted points called sigma points and estimates its mean and covariance. This process is called the unscented transformation, and leads to better performance in pedestrian tracking, as the Jacobian computation is not necessary anymore, with no or minimum increase of the computational cost [17].

d) Particle Filter: Based on Monte Carlo methods, this is a sample-based estimator widely used for pedestrian tracking [132], [80], [144], [230]. Unlike EKF, which deals with Gaussian and linearized distributions, the Particle Filter can perform state estimation of non-linear and non-Gaussian distributions. It consists in representing the target distribution by a set of samples, also called particles. A particle is a hypothesis of the true world state at time $t$. An important step in particle filtering is the resampling, which consists in withdrawing “weak” particles with low weights from the sample set, and increase the number of “strong” particles with high weights. This keeps the strongest beliefs, or hypotheses, in the sample set and discards those with very low probabilities [216]. Particle Filtering demands high computation capabilities, especially with a large number of particles. In [8], the authors proposed several implementations of the particle filter for detection and tracking purposes. Moreover, Bellotto and Hu [17] evaluated different Bayesian filters, such as EKF, UKF and Sequential Importance Resampling (SIR) Particle Filter, for people tracking and analysed the trade-off between performance and computational cost of each method.

B. Single pedestrian kernel-based tracking

a) Simple Template Matching: This is a brute force method. The goal is to compare a region of an image to a reference template image by minimizing the sum-of-square-difference (SSD). For example, in [109], a template matching is proposed for real-time people tracking which is robust to occlusions and variations of the illumination. In the approach proposed by Lipton et al. [136], moving objects are detected in camera images using frame differencing. By combining temporal differencing and template matching, the classified objects are then tracked in real-time on video. In [111], a feature selection method in image sequences is proposed to improve the performance of template matching tracking.

b) Mean Shift Method: This is a visual tracking technique that tries to match objects in successive frames, where each track is represented by a histogram. The histogram of the region of interest is compared to the histogram of the reference model. The mean shift algorithm iteratively clusters data points to the average of the neighbouring points using a kernel function, similar to $k$-means clustering [44]. In [52], the authors proposed a real-time object tracking using the mean-shift algorithm and the Bhattacharyya coefficient to localize the targets. This method is applied to non-rigid objects tracking observed from a moving camera. Collins [50] applied the mean-shift algorithm to 2D blob tracking and proposed a method to select the kernel scale for an efficient tracking of blobs. In particular, a difference of Gaussian (DOG) mean-shift kernel is chosen to efficiently track the blobs through space.

c) Layering-based Tracking: Layering consists in splitting an image into several layers by compensating the background motion to estimate the state of a moving object with a 2D parametric model [165]. Each layer is represented by its shape, motion and appearance (based on intensity) [257]. For instance, in [212], the authors proposed a dynamic layering-based object tracker exploiting spatial and temporal information from its shape, motion and appearance. Their estimation is done using a Maximum-A-Posteriori (MAP) approach with the EM algorithm. Layering-based trackers can handle multiple moving objects and occlusion. In [231], a layering-based method is combined with optical flow. A Bayesian framework is used to estimate the layers’ appearance and a mixture model is used to segment the image into foreground/background regions. Other layering-based tracking methods applied to imaging sensors can be found in [74], [124].

C. Body pose state tracking

Tracking the whole state of a pedestrian’s body – including skeleton pose, head direction, feet and walking directions – may provide useful information about the pedestrians state and intention. These silhouette tracking methods are based on an accurate shape description of the pedestrian object. The general technique consists in finding the pedestrian region in each frame with an object model computed from the previous frames. The advantage is that it can cope with different types of shape, occlusion problems, etc.

a) Contour Matching Tracking: Tracking is performed considering the contours of objects which are dynamically updated in successive frames. Geiger et al. [90] proposed a contour tracking method that is based on Dynamic Programming (DP) to detect and track the contour of multiple shapes and provide the optimal solution to the problem. Techmer [214] developed a real-time approach to contour tracking relying on the distance transformation of contour images and tested it on real-world images. Baumberg and Hogg [15] proposed a method that combines dynamic filtering (Kalman
filter) with an active shape model to track a waking pedestrian in real-time. However, this tracking technique is very sensitive to the initialization, so other solutions have been developed to overcome that issue [238].

b) Region-based Tracking: This technique is based on the color distribution of objects. In [1], a tracking algorithm is proposed based on multiple fragments of object images, creating a histogram of the current frame that is compared to the histogram of the patches. Their method is able to handle occlusion and pose changes in an efficient manner. Other methods have employed depth, probabilistic occupancy maps and gait features to estimate a region’s features, but in some cases (e.g. depth features) this requires the computation of multiple views of the same scene. Meyer and Bouthemy [145] developed a method to track object over a sequence of images using a recursive algorithm based on image regions information, such as their position, shape and motion model.

c) Shape Matching Tracking: Shape matching tries to match silhouettes found in two consecutive frames. Performed with Hough transform, it can handle occlusion problems. For instance, in [51], a silhouette-based model is used to identify people from their body shape and gait.

d) Skeleton Tracking for Body Language and Gesture Recognition: Skeleton tracking, based on tracking human body parts, is a very popular technique [89], [237], [192], [152]. Schwarz et al. [195], for example, presented a full-body tracker using depth data from Kinect. The 3D data is represented by a graph structure and it allows to deal with variations in pose and illumination. A skeleton is then fitted to the 3D data by constrained inverse kinematics and geodesic distances between body parts. Sinthanayothin et al. [203] proposed a review on skeleton tracking methods using Kinect sensors and displaying 3D skeletons in virtual environments. Make Human Community [164] is an open-source project building parametric models of humans based on realistic skeleton structures, mainly targeted at video games users, but also used as a generative machine vision and tracking model for 3D sensor data.

D. Multiple Pedestrian Tracking

Multiple pedestrian tracking, commonly called MTT for Multi-Target Tracking, names the task of (rather than specific algorithm for) tracking the poses of several pedestrians (or other objects) at the same time. The pedestrians may be close, overlapping, or obstructing one another, and they may be indistinguishable from one another other than by their pose. This is required for AV interactions with multiple pedestrians, ranging from two well-separated pedestrians, to small groups of pedestrians (often crossing roads together) and to dense crowds.

MTT creates a data association problem: how to know which pedestrian detection belongs to which track? A probabilistic MTT model would maintain beliefs at each time step about the state of every track and consider every possible association of detections to tracks; then, it would perform inference accordingly. However, the number of associations grows exponentially with the number of pedestrians, so this approach is unlikely to work in very crowded scenarios. Standard approximations then include making hard “winner-take-all” assignments at each time step; maintaining search trees of recent possible assignments; and pruning association hypotheses. There are many possible variations on these approximations, all making use of basic individual-pedestrian trackers as components.

Leal-Taixé et al. [126] presented a benchmark for Multiple Object Tracking that was launched in 2014 and called “MOTchallenge”. This benchmark provides a framework for evaluating the performance of state-of-the-art MTT algorithms. About 50 methods have been tested up to now on this benchmark. However, [126] does not describe these algorithms, while Fan et al. [78] only presents a survey on visual methods. A previous review on multiple object tracking was proposed in [141]. The remainder of this section will therefore extend their work for multiple person tracking and try to give an overview of the main methods, challenges and future directions of MTT techniques, which Intelligent Transportation Systems heavily rely upon. Figure 8 summarizes the techniques described in this section.

1) Categories of MTT methods: The following will develop the different categories of multi-target tracking methods that are defined according to their initialization method used, the processing method or the tracking output.

a) Initialization method: The first category is characterised by the detection technique used before tracking. The most commonly used method is Detection-Based Tracking (DBT) where a program is trained in advance to detect the target object in the input data (e.g. images) [107]. This technique can deal with a variable number of target objects, but it cannot track unknown objects that were not part of the training. The other initialization method is Detection-Free Tracking (DFT), which requires manual initialization, i.e. an operator labels manually the target objects. In this case, the object detection is error-free but the tracking can usually only deal with a fixed
number of target objects. Neiswanger et al. [156] proposed a method to track multiple people in video sequences without any pre-defined person detector. A Dirichlet process is used to find the clusters in the images and then a Sequential Monte Carlo (SMC) method with local Gibbs iterations and a Particle Markov Chain Monte Carlo (PMCMC) are used to infer the posterior of targets. Lin et al. [134] developed a detection-free multiple target tracking method which relies on video bundle representation and a spatio-temporal graphical model to infer the trajectories of people.

b) Processing model: This second category refers to the information processing mode: online or offline tracking. Online tracking [116] is a sequential tracking, which relies on up-to-date information. It is a causal method where only past and current observations are used. Offline tracking [114] instead uses information both from past and future observations, therefore it is not causal. In order to estimate the output, offline tracking needs to evaluate all the observations from all the frames, which requires a high computation cost. The manual assignment guarantees a tracking process free of false detections, but is not suitable for real-time applications. Both online and offline tracking methods are proposed in [240].

c) Tracking output: Finally, MTT methods can be grouped according to their output. The tracking output results are fixed for MTT methods relying on deterministic optimization, i.e. there is no randomness when these methods are run many different times, whereas for probabilistic optimization methods, the tracking output may vary for several trials cf. section V-D3.

2) Challenges of MTT approaches: There are multiple challenges with the tracking of multiple objects. Here we summarise the most important ones.

a) Similarity measurement: The first problem is how to measure the similarity between objects in different frames. Different models have been proposed to deal with the similarity measurement between objects. The most commonly used technique in visual tracking relies on the object’s appearance, i.e. its visual features. There are local features, which can be obtained by the KLT algorithm or optical flow (if we treat each pixel as the finest local range) to get information about object motion patterns [199]. Region features are extracted from an image and represented by a bounding box. Three main types of region features exist: zero-order, first-order and up-to-second-order type. The zero-order type represents region features as color histogram or raw pixel templates. Although color is a common similarity measure, the problem is that it does not take into account the spatial layout of the object region. A first-order type uses gradient-based representations or level-set formulation to deal with region features [47]. Gradient is a robust technique because it describes well the shape of the object and it is less dependent to illumination conditions, but it cannot handle occlusion problems. An up-to-second-order type computes region covariance matrices to model the observed features [173]. This is a robust strategy but it requires a high computation capability.

b) Track identification: The second problem consists in recovering the identity of objects from the similarity measurement across frames. Different strategies are used to compute the similarity between objects. A survey on similarity measure for probability density functions is provided in [250]. In case of a single cue, a distance measure (e.g. the Bhattacharyya distance) is computed from two color histograms and then transformed into similarity using the exponential function or an affinity measure such as the Normalized Cross Correlation (NCC). When multiple cues are available, there are several strategies used to fuse the information [27]. Boosting, for example, consists in selecting the most representative features from a large set of proposed features using a machine learning algorithm such as AdaBoost [235]. Concatenation uses features from different cues and concatenates them for computation. Summation takes affinity values from different features and adds a weight to each value. Product strategy assumes independence between affinity values and computes their weighted product. Cascading uses diverse visual representations and tries to determine the finest model appearance [223].

To improve tracking prediction, an exclusion model can be used to prevent physical collisions, e.g. it assumes that two distinct pedestrians cannot be at the same place at the same time. Two types of constraints can be applied to the trajectory hypotheses: detection-level exclusion and trajectory-level exclusion [141]. Detection-level exclusion assumes that two detections in a frame cannot be assigned to the same target. Trajectory-level exclusion means that two trajectories cannot be too close to each other. In order to avoid that, a penalty is assigned to two track hypotheses that are too close and which have different trajectories such that to suppress one of them.

c) Occlusion: The third problem is how to handle occlusions of tracking targets. Three major strategies are employed to face this challenge. Part-to-whole divides the object into several parts and then computes an affinity for each part. When an occlusion occurs, only the unoccluded parts are taken into account for estimation [208], [236]. In hypothesis-and-test, detection hypotheses are generated for two objects with different levels of occlusion, which are then tested for example using MAP or a multi-person detector [211]. The buffer-and-recover technique keeps the states of objects over several frames, before and during an occlusion. When it ends, the states of objects are recovered using the observations on the frame buffer [189].

3) Multi-tracks and data association methods: Probabilistic or deterministic optimization are the common methods to deal with multiple tracks and data association problems. Data association is about the uncertainty related to measurements, it aims at associating observed measurements with current known tracks or generate new tracks. Deterministic optimization methods are usually suitable for offline tracking, as they require observations from several or all the frames in advance [141], whereas probabilistic methods are commonly used for online or real-time tracking. Bar-Shalom and Li [12] presented several data association algorithms, such as Nearest Neighbors (NN), Multi-Hypothesis Tracking (MHT), Joint Probabilistic Data Association Filter (JPDAF), or Probability Hypothesis Density (PHD), and evaluated their performances.
a) Global Nearest Neighbour: GNN [23] is one of the simplest methods for data association. At every new time step, it “hardly” assigns each current observation to a single best object without revising the past. In [120], GNN is described as a 5-step algorithm:

1) Receive data for each scan
2) Each track is first defined as a cluster and if common observations are found for two tracks, they are merged into a “super cluster”
3) Observations are assigned to each cluster using Munkres algorithm [123]
4) Tracks’ states are updated using some estimation technique such as Kalman filter
5) Observations which are not associated to any existing tracks are used to create new tracks

In [10] developed a multiple person tracker where GNN is used for data association with a new distance function and a Kalman filter for state estimation. The proposed method is suitable for occlusion issues.

d) Probabilistic Hypothesis Density (PHD): This filter was introduced by [49]. It can track a variable number of tracks, estimating their number and their locations at each time step. There are different types of PHD filter, such as the Sequential Monte Carlo PHD filter (SMC-PHD) [186], the Gaussian Mixture PHD filter (GM-PHD) [242] and the Gaussian Inverse Wishart PHD filter (GIW-PHD) [95]. Zhang et al. [244] used a GMM-PHD (Gaussian Mixture Measurement PHD) tracker to tackle problems with bearing measurements. Khazaie et al. [115] developed a PHD filter in distributed camera network where each camera fuses its track estimates with its neighbors. Feng et al. [81] proposed a variational Bayesian PHD filter with deep learning update to track multiple persons. In [57], a PHD filter is used to track in real-time multiple people in a crowded environment. Yoon et al. [239] used hybrid observations in a PHD filter, where the filter observations are combined with local observations generated by on-line trained detectors. This method allows the missed detections and assigns an identity to each person.

e) Markov Chain Monte Carlo Data Association (MCMCDA): Introduced first by [169], this filter is an approximation of the Bayesian filter, derived from MCMC, which draws a set of samples and builds Markov chains over the target state space. A sampler moves from its current state to the next following the proposal distribution. The new state is accepted with an acceptance probability, otherwise the sampler stays at its current state. Oh et al. [158], [157] proposed an MCMCDA algorithm known as Metropolis-Hastings, where single-scan and multi-scan MCMCDA algorithms are used for known and unknown number of targets, respectively. A bipartite graph is used to represent possible associations between observations and targets. Their simulation results show better performances than MHT algorithms and their method has been tested on tracking people from video sequences. Yu et al. [241] proposed a data-driven MCMC (DD-MCMC) approach for sampling and incorporating a person motion and appearance information, using a joint probability model. Their method was tested in simulations and on real videos.

f) Bipartite graph matching: This method uses two sets of graph nodes representing existing trajectories and new detections in online tracking, or two sets of tracklets in offline tracking. The weights of the nodes model the affinities between trajectories and detections. Bipartite assignment algorithm or optimal Hungarian algorithm is used to find matching nodes in the two sets. A review on graph matching methods is presented in [53]. Chen et al. [41] used a dynamical graph matching method to track multiple people in order to dynamically change the graph nodes with the tracks movements.

g) Dynamic Programming: This method solves the data association problem by linking several detections over times. Pirsiavash et al. [171] used a greedy algorithm based on dynamic programming to find the global solution in a network flow. Another method is presented in [20] which can follow up to six people over several frames.

h) Min-Cost Max-Flow Network Flow: This is a popular method, which models the network flow as a directed graph.
A trajectory is represented by a start node and an end node (sink), and it corresponds to one flow path in the graph. The global optimal solution is obtained with the push-relabel algorithm. Zhang et al. [243] used min-cost flow algorithm combined with a recursive occlusion model to deal with occluded people. Their method does not require pruning. Chari et al. [38] proposed a new approach to the min-cost max-flow network flow optimization using pair-wise costs, which can deal with occluded people.

i) Conditional Random Field (CRF): A graph \( G = (V, E) \) is defined as a set of nodes \( V \) and a set of edges \( E \). The nodes represent observations and tracklets. A label is used to predict which track the observations are linked to. Sutton and McCallum [209] presented a tutorial on CRF including the methods, parameter estimation and implementations. Taycher et al. [213] proposed a person tracking method which learns observation from data: this is based on a CRF state-space estimation and a grid-filter with real-time capabilities. Milan et al. [147] developed a CRF-based multiple target tracker, which starts by detecting people using a HOG-SVM detector, and then defines two unary potentials for detection and superpixel nodes. Milan et al. [149] proposed a CRF-based multiple person tracker using discrete-continuous energy minimization, where the goal is to assign a unique trajectory to each detection.

j) Maximum-Weight Independent Set (MWIS): The MWIS graph is defined as \( G = (V, E, w) \). As in the CRF, the nodes \( V \) represent the pairs of tracklets in successive frames, which are given a weight \( w \) indicating the affinity of the tracklet pair. If two tracklets share the same detection, then their edges \( E \) are connected together. Brendel et al. [26] proposed a multi-target tracker based on MWIS data association algorithm. Their approach is as follows:

1) Detection of multiple targets in all frames using different object detectors;
2) Detections are considered as distinct tracks, with the assumption that one detection can only be one track;
3) A graph is built to match tracks over two consecutive frames;
4) An MWIS algorithm is used to perform the data association with guaranteed optimal solution;
5) Statistical and contextual properties of objects are learnt online for their similarity measurement using Mahalanobis distances;
6) Steps 2 to 5 are repeated over the frames to handle long-term occlusions by merging or splitting tracks.

In [100] is proposed a multi person tracker where the data association problem is modelled as a Connected Component Model (CCM) based on MWIS. A divide-and-conquer strategy is used to solve the Multi-Dimensional Assignment (MDA) problem.

4) Advanced MTT Techniques: “Advanced MTT” refers here to multi-target tracking that is performed at a higher level, simultaneously with other tasks.

a) MTT With Video Adaptation: MTT approaches rely on an object detector that is trained offline, so its performance can be totally different from a video to another. A possible solution is to create a generic detector adapted for a specific video by tuning some parameters. Previous works for multiple people tracking include [88], [39].

b) MTT With Deep Learning: Deep learning has proven a high performance method for classification, detection and many computer visions tasks. Applied to MTT, deep learning could provide a stronger observation model which could increase the tracking accuracy [240], [128]. In [160], Ondruska et al. introduced deep tracking, an end-to-end human tracking approach, based on recurrent neural network, using unsupervised learning on simulated data without dealing with the data association problem. In [67], Dequaire et al. used a similar method for static and dynamic person tracking in real-world environments. In [148], Milan et al. proposed a complete online multiple people tracking method based on recurrent neural networks.

c) MTT Under Multiple Cameras: Also called Multi-Target Multi-Camera (MTMC), this type of systems can be used to improve large tracking problems. Wang et al. [227] presented a survey on the challenges of MTMC. One problem would be if the cameras angles overlap, in which case it is necessary to find a good way to fuse multiple information. But if the cameras angles do not overlap, then the data association problem becomes an identification problem. In [183], Ristani et al. proposed different performance measures to test MTMC methods. In [184], they used neural networks to learn features from MTMC systems and for re-identification. In [139], Generalized Maximum Multi-Clique optimization, a graph-based method, is used for the MTMC problem. Munaro et al. [154] developed an open source software, called OpenPTrack, for multi-camera calibration and people tracking using RGB-D data.

d) Multiple 3D Object Tracking: This method could provide better position accuracy, size estimation and occlusion handling. The major problem for this technique is the camera calibration, to know its pose and scene layout. Park et al. [168] applied 3D object tracking from a monocular camera for augmented reality applications. Some other works on 3D visual tracking include [59], [167], [191], who used a single camera with a multi-Bernoulli mixture tracking filter. Some works with 3D lidar sensors include [110], [205], [233], who proposed online classification of human for 3D lidar tracking. In [176], both camera and lidar data are used to improve people tracking.

e) MTT With Scene Understanding: Scene understanding can provide contextual information and scene structure for the tracking algorithm, especially in crowded scenes. Leal-Taixé et al. [125] developed a model that decomposes an image and extracts features from the observed scene called “interaction feature strings”. These features are then used in a Random Forest framework to track human targets [65].

f) MTT With Other Computer Vision Tasks: Information from image segmentation or human pose estimation could not only improve the performance of multiple people target but also the computation of the tracking algorithm. For example, in [147], tracking is done with image segmentation and in [47] people are tracked for group activity recognition.
E. Discussions

Single pedestrian tracking is now a fully mature area with widely available open source and commercial implementations. Body pose tracking has made strong recent progress, likely to soon bring it to maturity, through the use of larger data sets and compute power.

Tracking multiple pedestrians requires additional algorithms which were major research areas until recently, but have largely matured in the last few years with methods such as MHT becoming standard. Tracking multiple pedestrians in the presence of occlusion by one another or by other objects remains a serious research problem, which requires the use of other data or prior information to compensate for the lack of purely visual data. We suggest that the higher level models from psychology and sociology discussed in the Part II of this review [29] should be used to provide such priors. Traditionally, tracking was a clearly separate task from both lower (detection) and higher (behaviour modelling) layers of pedestrian modelling, but a current trend is to merge it with nearby layers through neural network and probabilistic methods in this fashion to improve performance.

Practical implementation of tracking algorithms may be found in the Bayes Tracking library14 which provides open-source implementation of EKF, UKF and SIR Particle Filters with NN and JPDA data association algorithms. In addition, a detection and tracking pipeline15 contains an implementation of MHT. Choi et al. [45] proposed a fast tracker TRACA16 with a deep feature compression approach for single target tracking. In terms of computational efficiency, Bellotto and Hu [17] have shown that Kalman-based people tracking is much faster than particle-based, and in particular that UKF was faster and still almost as reliable as particle filter. Linder et al. [135] proposed a comparison (computation speed and other metrics) of various people tracking methods, including NN trackers, MHT and others. A common heuristic for some mobile robots is to run at 10Hz or more, i.e if the robot moves at 1m/s, a people tracker running at 10Hz will estimate the position of humans every 10cm, which is usually considered safe. But with cars moving much faster such as 10m/s (36km/h), the computational requirements would be greater, such as operating 100Hz to obtain the same 10cm accuracy.

VI. Conclusions

Autonomous vehicles must interact with pedestrians in order to drive safely and to make progress. It is not enough to simply stop whenever a pedestrian is in the way as this leads to the freezing robot problem and to the vehicle making no progress. Rather, AVs must develop similar interaction methods as used by human drivers, which include understanding the behaviour and predicting the future behaviour of pedestrians, predicting how pedestrians will react to the AVs movements, and choosing those motions to efficiently control the interaction.

This Part I review has surveyed the state of the art in the lower levels of machine perception and intelligence needed to enable such interaction control, namely: sensing, detection, recognition, and tracking of pedestrians. It has found that the level of maturity of these fields is high at the lowest levels but fades into current research areas at the higher levels. Sensing technology has progressed to maturity over the last decade so that lidars and stereo cameras are now reliable and cheap enough for use in research and even by hobbyist systems. Similarly, GPU compute hardware has fallen in price to enable both stereo camera processing and deep learning recognition to be run in these systems. Deep learning recognition has largely replaced classical feature-based for detection. Open source software is mature and freely available for these tasks. Beyond detection are areas with successful, open source, partial implementations but which require further research to become fully mature. Recognition of body pose and head direction are almost mature, including via deep learning methods. But recognition of higher level states, such as gestures used for explicit signalling, body language used as implicit signalling, actions as sequences of poses, and recognition of underlying emotional state, remain research areas.

Tracking is mature for single pedestrians but remains challenging for multiple pedestrians in the presence of occlusion. Algorithms to solve this task are known but require the use of extensive prior knowledge to predict behaviour in the absence of sensory information, which is not yet fully available. This includes information from recognition of poses, gestures, actions, and emotions, but also feedback information from very high level models of behaviour and psychology which will be studied in part II of this review.

REFERENCES

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), volume 1, pages 798–805, June 2006.
[2] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3908–3916, 2015.
[3] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(12):2037–2047, Dec 2006.
[4] M. Alwan, P. J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder. A smart and passive floor-vibration based fall detector for elderly. In Information and Communication Technologies, volume 1, pages 1003–1007, 2006.
[5] A. Amrdit, G. Thomaidis, P. Maroudis, P. Lytrivis, and G. Karaseitani-dis. Multiple Hypothesis Tracking Implementation, chapter 10. InTech, Rijeka, 2012.
[6] A. Angelova, A. Krizhevsky, V. Vanhoucke, A. Ogale, and D. Fergus. Real-time pedestrian detection with deep network cascades. In Proceedings of BMVC 2015, 2015.
[7] K. O. Aras, O. M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA), pages 3402–3407, 2007.
[8] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. Trans. Sig. Proc., 50(2):174–188, Feb. 2002.
[9] AutonomousStuff.com. Radar specification comparison chart, 2018.
[10] M. Azari, A. Seyhi, and A. H. Rezaie. Real time multiple object tracking and occlusion reasoning using adaptive kalman filters. In Machine Vision and Image Processing (MVIP), 2011 7th Iranian, pages 1–5. IEEE, 2011.
I. del Pino, V. Vaquero, B. Masini, J. Solà, F. Moreno-Noguer, V. Delaitre, J. Dequaire, P. Ondrka, D. Rao, D. Wang, and I. Posner.

Deep tracking.

P. Dollár, C. Wojek, B. Schiele, and P. Perona.

Pedestrian detection: from visual lifelogs: State of the art and future challenges. In 2015 IEEE International Conference on Computer Vision and Pattern Recognition, pages 151–158, Oct 2015.

P. Feng, W. Wang, S. M. Naqvi, and J. Chambers.

Variational bayesian phd filter with deep learning network updating for multiple human tracking. In 2015 Sensor Signal Processing for Defence (SSPD), pages 1–5, Sept 2015.

Fernández-Caballero, J. C. Castillo, J. Martínez-Cantos, and R. Martínez-Toms. Optical flow or image subtraction in human detection from infrared camera on mobile robot. Robotics and Autonomous Systems, 58(12):1273–1281, 2010.

Flur. Advanced thermal imaging cameras for r&d and science applications, 2012.

F. Flot, M. Dumitru-Guzu, J. F. Kooij, and D. M. Gavrilă. Joint probabilistic pedestrian head and body orientation estimation. In 2014 IEEE Intelligent Vehicles Symposium Proceedings, pages 617–622. IEEE, 2014.

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, Jul 1983.

C. Fox. Data Science for Transport: A Self-Study Guide with Computer Exercises. 2018.

W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In International workshop on automatic face and gesture recognition, volume 12, pages 296–301, 1995.

A. Gaidon and E. Vig. Online domain adaptation for multi-object tracking. CoRR, abs/1508.00776, 2015.

J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H. Seidel. Motion capture using joint skeleton tracking and surface estimation. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 1746–1753, June 2009.

D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos. Dynamic programming for detecting, tracking, and matching deformable contours. IEEE Transactions on Pattern Analysis & Machine Intelligence, (3):294–302, 1995.

T. Germa, F. Lerasle, N. Ouadah, and V. Cadetan. Vision and rfid data fusion for tracking people in crowds by a mobile robot. Computer Vision and Image Understanding, 114(6):641–651, 2010.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587, June 2014.

A. González, G. Villalonga, J. Xu, D. Vázquez, J. Amores, and A. M. López. Multiview random forest of local experts combining RGB and LIDAR data for pedestrian detection. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), pages 356–361, 2015.

I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016. http://www.deeplearningbook.org.

K. Granström and U. Orguner. A phd filter for tracking multiple extended targets using random matrices. IEEE Transactions on Signal Processing, 55(10):4825–4836, Oct 2007.

L. Guo, L. Li, Y. Zhao, and Z. Zhao. Pedestrian tracking based on camshift with kalman prediction for autonomous vehicles. International Journal of Advanced Robotic Systems, 13(3):120, 2016.

J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microvessel kinect sensor: A review. IEEE Transactions on Cybernetics, 43(5):1318–1334, Oct 2013.

R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2980–2988, Oct 2017.

Z. He, X. Li, X. You, D. Tao, and Y. Y. Tang. Connected component model for multi-object tracking. IEEE Transactions on Image Processing, 25(8):3698–3711, Aug 2016.

W.-B. Horng, C.-Y. Chen, Y. Chang, and C.-H. Fan. Driver fatigue detection based on eye tracking and dynamic, template matching. In Networking, Sensing and Control, 2004 IEEE International Conference on, volume 1, pages 7–12. IEEE, 2004.

P. Horridge and S. Maskell. Real-time tracking of hundreds of targets. In Proceedings of the 20th British Machine Vision Conference, 2009.

A. Gaidon and E. Vig. Online domain adaptation for multi-object tracking. CoRR, abs/1508.00776, 2015.

J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H. Seidel. Motion capture using joint skeleton tracking and surface estimation. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 1746–1753, June 2009.

D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos. Dynamic programming for detecting, tracking, and matching deformable contours. IEEE Transactions on Pattern Analysis & Machine Intelligence, (3):294–302, 1995.

T. Germa, F. Lerasle, N. Ouadah, and V. Cadetan. Vision and rfid data fusion for tracking people in crowds by a mobile robot. Computer Vision and Image Understanding, 114(6):641–651, 2010.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587, June 2014.

A. González, G. Villalonga, J. Xu, D. Vázquez, J. Amores, and A. M. López. Multiview random forest of local experts combining RGB and LIDAR data for pedestrian detection. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), pages 356–361, 2015.

I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016. http://www.deeplearningbook.org.

K. Granström and U. Orguner. A phd filter for tracking multiple extended targets using random matrices. IEEE Transactions on Signal Processing, 55(10):4825–4836, Oct 2007.

L. Guo, L. Li, Y. Zhao, and Z. Zhao. Pedestrian tracking based on camshift with kalman prediction for autonomous vehicles. International Journal of Advanced Robotic Systems, 13(3):120, 2016.

J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microvessel kinect sensor: A review. IEEE Transactions on Cybernetics, 43(5):1318–1334, Oct 2013.

R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2980–2988, Oct 2017.

Z. He, X. Li, X. You, D. Tao, and Y. Y. Tang. Connected component model for multi-object tracking. IEEE Transactions on Image Processing, 25(8):3698–3711, Aug 2016.

W.-B. Horng, C.-Y. Chen, Y. Chang, and C.-H. Fan. Driver fatigue detection based on eye tracking and dynamic, template matching. In Networking, Sensing and Control, 2004 IEEE International Conference on, volume 1, pages 7–12. IEEE, 2004.

P. Horridge and S. Maskell. Real-time tracking of hundreds of targets with efficient exact jpf implementation. In Information Fusion, 2006 9th International Conference on, pages 1–8. IEEE, 2006.

W. Huang, K. Kim, Y. Yang, and Y. S. Kim. Automatic shadow removal by illumination in hsv color space. Comput. Sci. Inf. Technol, 3(3):70–75, 2015.

I. Institute for Advanced Motorists. How to Be A Better Driver: Advanced Driving the Essential Guide. Motorbooks, 2007.

S. International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, 2018.

U. Iqbal, A. Milan, and J. Gall. Posetrack: Joint multi-person pose estimation and tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2011–2020, 2017.
[253] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In Computer Vision, IEEE International Conference on, 2015.

[254] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 3774–3783, Oct 2017.

[255] V. Zhmud, N. Kondratiev, K. Kuznetsov, V. Trubin, and L. Dmitrov. Application of ultrasonic sensor for measuring distances in robotics. In Journal of Physics: Conference Series, volume 1015, page 032189. IOP Publishing, 2018.

[256] B. Zhou and N. Bose. Multitarget tracking in clutter: Fast algorithms for data association. IEEE Transactions on aerospace and electronic systems, 29(2):352–363, 1993.

[257] Y. Zhou and H. Tao. A background layer model for object tracking through occlusion. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pages 1079–1085. IEEE, 2003.

[258] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), volume 2, pages 1491–1498. IEEE, 2006.

Fanta Camara (STMIEEE) is a PhD student at the Institute for Transport Studies (ITS), University of Leeds, UK, a Visiting Researcher in the School of Computer Science, University of Lincoln, UK, and a member of the Lincoln Centre for Autonomous Systems (L-CAS) and a member of Robotics at Leeds. She received her Robotics Engineering degree (Master’s) from Sorbonne Université (former Université Pierre et Marie Curie, UPMC), Paris, France. She worked as an intern at Airbus Defence and Space, National Taiwan University (Taipei, Taiwan). DLR (German Aerospace Center) and Robert Bosch GmbH. Her main research interests are in human-robot interaction for autonomous systems, game theory, computer vision, ML/DL and robotics.

Nicola Bellotto (SMIEEE) is an Associate Professor in the School of Computer Science, University of Lincoln, UK, and a member of the Lincoln Centre for Autonomous Systems. His main research interests are in machine perception, especially for human detection, tracking, identification and activity recognition with autonomous mobile robots. He has a Master in Electronic Engineering from the University of Padua, Italy, and a PhD in Computer Science from the University of Essex, UK. Before joining the University of Lincoln, he was a researcher in the Active Vision Lab at the University of Oxford. Dr Bellotto is the recipient of a Google Faculty Research Award and a PHCo-I in several research projects on autonomous mobile robots.

Serhan Cosar is currently a Senior Lecturer in Robotics at De Montfort University. He received his M.Sc and Ph.D. degrees of electronics Engineering and Computer Science from Sabanci University, Istanbul, Turkey in 2008 and 2013, respectively. Before joining De Montfort University, he worked as a post-doctoral research fellow at University of Lincoln, UK. He is the author of 3 book chapters, 9 international journal papers (CVIU, IEEE-TCSTV) and 16 international conference papers (IROS, ICCV, AVSS). He has participated in a number of European and National research projects. His research interest includes robot perception, human tracking, activity recognition, sparse representation, and distributed estimation.

Dimitris Nathanael is an assistant professor at the National Technical University of Athens, School of Mechanical Engineering, where he teaches ergonomics / human factors, design innovation and safety management courses. His scientific interests span across all facets of the interaction of humans with technology with an emphasis on cognition. Dimitris holds a diploma in Production Engineering & Management science, Technical University of Crete, a D.E.A. in Ergonomics, from the Conservatoire National des Arts et Metiers (C.N.A.M.) in France, and a PhD in the field of Cognitive Ergonomics, from the National Technical University of Athens. Before joining the academia, Dimitris has worked in for several years in industry as a consultant in ergonomics, human computer interaction and occupational safety. He is the President of the Hellenic Ergonomics Society and the national representative in the International Ergonomics Association.

Matthias Althoff is an assistant professor in computer science at Technische Universität München, Germany. He received his diploma engineering degree in Mechanical Engineering in 2005, and his PhD degree in Electrical Engineering in 2010, both from Technische Universität München, Germany. From 2010 to 2012 he was a postdoctoral researcher at Carnegie Mellon University, Pittsburgh, USA, and from 2012 to 2013 an assistant professor at Technische Universität Ilmenau, Germany. His research interests include formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, robotics, automated vehicles, and power systems.

Jingyuan Wu is pursuing his PhD degree with Chassis Systems Control, Robert Bosch GmbH in Abstatt, Germany, in cooperation with the Technical University of Munich, Germany. He received his B.Sc. degree in mechanical engineering from University of Science and Technology Beijing, China in 2013 and M.Sc. degree in mechanical engineering from Technical University of Kaiserslautern, Germany in 2017. His work is focused on signal processing and in particular pedestrian motion prediction in urban environments.

André Dietrich is a research associate and PhD student at the Chair of Ergonomics at the Technical University of Munich. His research focuses on interaction between automated vehicles and pedestrians with an emphasis on the effects of visual communication. He currently leads Work package 2 “Psychological Models on Human Interaction & Intention Recognition Algorithms” in the European Project interACT. He graduated in Aerospace Engineering at the Technical University of Munich.

Johannes Ruenz is currently working in the advanced development department of Robert Bosch GmbHs Chassis Systems division. He studied Electrical Engineering and Information Technologies at the Karlsruher Institute of Technology (KIT) and received his master degree in 2014. His research topic is focused on active pedestrian protection systems using their smartphones and vehicles environment sensors.

Charles W. Fox is a Senior Lecturer in Robotics and Automation at the University of Lincoln and a Visiting Lecturer in Vehicle and Road Automation at the Institute for Transport Studies at the University of Leeds. He received his MA in Computer Science from the University of Cambridge, MSc in Cognitive Science from the University of Edinburgh, and DPhil in Engineering from the University of Oxford. He is the author of the Springer textbook “Data Science for Transport: a self-study guide with computer exercises”. 
Supplementary Material: Pedestrian Models for Autonomous Driving
Part I: low level models, from sensing to tracking

Fanta Camara\textsuperscript{1,2}, Nicola Bellotto\textsuperscript{2}, Serhan Cosar\textsuperscript{3}, Dimitris Nathanael\textsuperscript{4}, Matthias Althoff\textsuperscript{5}, Jingyuan Wu\textsuperscript{6}, Johannes Ruenz\textsuperscript{6}, André Dietrich\textsuperscript{7} and Charles W. Fox\textsuperscript{1,2,8}

I. QUALITY OF CITATIONS
These linked papers (Part I and II) review over 450 papers from high quality journals and conferences such as CVPR, ICRA, PAMI, IROS, ITSC, ECCV, IV. It is common in Computer Science fields including machine vision and machine learning for conferences to be considered higher quality or similar quality to journals, while psychology and sociology fields typically consider journals to be more authoritative. The following figures give some ideas about the quality of the cited papers.

![Fig. 1. Number of citations per paper](image1)

![Fig. 2. Number of citations per paper and per year of publication](image2)

![Fig. 3. Number of citations per paper](image3)

This project has received funding from EU H2020 interACT: Designing cooperative interaction of automated vehicles with other road users in mixed traffic environments under grant agreement No 723395.

1 Institute for Transport Studies (ITS), University of Leeds, UK
2 Lincoln Centre for Autonomous Systems, University of Lincoln, UK
3 Institute of Engineering Sciences, De Montfort University, UK
4 School of Mechanical Engineering, Nat. Tech. University of Athens
5 Department of Computer Science, Technische Universität München
6 Robert Bosch GmbH, Germany
7 Chair of Ergonomics, Technische Universität München (TUM), Germany
8 Ibex Automation Ltd., UK

Manuscript received March, 11, 2019; Revision 1: October, 21, 2019; Revision 2: February, 25, 2020
## II. Summary of Pedestrian Recognition Models

| Study/Paper       | Sensor input            | Method                                      | Pedestrian Recognition Type | Additional Info                        | SAE Level |
|-------------------|-------------------------|---------------------------------------------|-----------------------------|----------------------------------------|-----------|
| Cao et al. [1]    | Images                  | CNN model with Part Affinity Fields (PAF)     | Pose estimation             | OpenPose: open source software         | Level 2   |
| Shotton et al. [3] | Motion capture and synthetic data | Body parts representation model | 3D human pose estimation  |                                        | Level 2   |
| Iqbal et al. [4]  | Video data              | Graphical model                             | Pose estimation and tracking | Release of PoseTrack a new dataset     | Level [2,3]|
| Tompson et al. [5] | Monocular images        | Deep CNN model with Markov Random Field      | Pose estimation             |                                        | Level 2   |
| Ma et al. [6]     | Images                  | CNN model                                   | Body heading                | No annotations needed                   | Level 2   |
| Kohari et al. [7] | Video                   | CNN model                                   | Body orientation            | Service robot                          | Level 2   |
| Darrelli et al. [8] | Images            | Statistical model                           | Head direction              | From a mobile robot                     | Level [2,3]|
| Schulz et al. [9] | Grayscale images        | Multi-classifiers                           | Head pose                   |                                        | Level 2   |
| Microsoft Azure API [10] | Video                 | Head pose and facial features               | Commercial product          |                                        | Level 2   |
| Benfold et al. [11] | Video                | HOG and colour features                     | Gaze tracking               |                                        | Level 2   |
| Baltrusaitis et al. [12] | Video          | Deep learning model                         | Head pose and eye-gaze estimation | Openface: open source software     | Level 2   |
| Cornejo et al. [13] [14] | Images           | Principal Component Analysis + Gabor wavelets or CENTRIST features | Emotion recognition         |                                        | Level 2   |
| Cambria et al. [15] [16] | Images          | Review paper                                | Sentiment analysis          | [Level 2,3]                             |           |
| Poria et al. [17] | Videos                 | CNN model with recurrent multilayer kernel learning | Emotion recognition         |                                        | Level 2   |
| Horng et al. [18] | Video                  | Dynamic template matching                   | Driver fatigue detection    |                                        | Level 2   |
| Denisy et al. [19] | Video                  | Face expression recognition                 | FaceReader: commercial product |                                        | Level 2   |
| Ahmed et al. [20] | Images                  | Deep neural networks                        | Re-identification          |                                        | Level 2   |
| Zheng et al. [21] | Images                  | Bag of Words model                          | Re-identification          |                                        | Level 2   |
| Zheng et al. [22] | Images                  | CNN model                                   | Re-identification          | Unlabeled images                       | Level 2   |
| Li et al. [23]    | Images                  | Filter airing neural network (FPNN) model   | Re-identification          | Occlusion handling                     | Level 2   |
| Chen et al. [24]  | Images                  | Hidden Markov model (HMM)                   | Gesture recognition         |                                        | Level 2   |
| Freeman et al. [27] | Images            | Orientation histograms                      | Gesture recognition         | 10 different hand gestures recognition | Level 2   |
| Ren et al. [28]   | Images                  | Template matching with Finger-earth Mover’s Distance (FEMID) | Gesture recognition         |                                        | Level 2   |
| Quinteiro et al. [29] | Images              | Hidden Markov models (HMM)                  | Body language Recognition   |                                        | Level [2,3]|
| Wang et al. [30]  | Images                  | Background subtraction + PCA                | Body language Recognition   |                                        | Level 2   |
| Chaaraoui et al. [31] | Videos            | Contour points                              | Activity recognition       | Real-time method                       | Level 2   |
| Doolar et al. [32] | Videos                 | Spatio-temporal features                    | Activity recognition       |                                        | Level 2   |
| Vail et al. [33]  | Videos                  | Hidden Markov models and Conditional random field | Activity recognition |                                        | Level 2   |
| Liu et al. [34]   | RGB data                | Coupled conditional random field            | Activity recognition       |                                        | Level 2   |
| Coppola et al. [35] | RGB-D data            | Dynamic Bayesian mixture model (DBM)        | Activity recognition       |                                        | Level 2   |
## III. Summary Pedestrian Tracking Models

| Study/Paper               | Input/Evaluation          | Method                              | Tracking Models       | Additional Info | SAE Level |
|--------------------------|---------------------------|-------------------------------------|-----------------------|-----------------|-----------|
| Del Pino et al. [36]     | Low resolution LiDAR data | Multi-Hypothesis EKF (MHEKF)        | Point Tracking        |                 | Level 2   |
| Bellotto et al. [37]     | Robot with laser and camera | EKF, UKF, SIR Particle filter      | Point Tracking        | Trade-off between performance and computation cost | Level 2   |
| Arulampalam et al. [38]  | Example                   | Particle filter implementations    | Point Tracking        |                 | Level 2   |
| Fen et al. [39]          | Video data                | Color histogram based particle filter | Point Tracking       |                 | Level 2   |
| Jurie et al. [40]        | Video data                | Template matching with SSD          | Kernel-based Human Tracking | Real-time method and robustness to occlusions and illuminations | Level 2   |
| Lipton et al. [41]       | Video data                | Frame differencing + Template matching | Kernel-based Human Tracking | Real-time method | Level 2   |
| Kaneko et al. [42]       | Image sequences           | Template matching with a feature selection method | Kernel-based Human Tracking |                 | Level 2   |
| Comaniciu et al. [43]    | Moving camera data        | Mean-shift algorithm with Bhattacharyya coefficient | Kernel-based Human Tracking |                 | Level 2   |
| Collins et al. [44]      | Video data                | Mean-shift algorithm with 2d blob tracking | Kernel-based Human Tracking |                 | Level 2   |
| Tao et al. [45]          | Airborne vehicle tracking system | Dynamic layering method + MAP using EM algorithm | Kernel-based Human Tracking |                 | Level 2   |
| Yalcin et al. [46]       | Image sequences           | Layering method with optical flow   | Kernel-based Human Tracking |                 | Level 2   |
| Geiger et al. [47]       | Image sequences           | Contour matching method based on Dynamic programming | Tracking pedestrian body state |                 | Level 2   |
| Techmer et al. [48]      | Real-world images         | Contour tracking with distance transformations of contour images | Tracking pedestrian body state |                 | Level 2   |
| Baumberg [49]            | Image sequences           | Dynamic Kalman filter with active shape model | Tracking pedestrian body state | Method sensitive to initialization | Level 2   |
| Yilmaz [50]              | Image sequences           | Region color histogram method       | Tracking pedestrian body state | Occlusion and pose changes handling | Level 2   |
| Adam et al. [51]         | Image sequences           | Recursive algorithm using image regions information | Tracking pedestrian body state |                 | Level 2   |
| Meyer et al. [52]        | Image sequences           | Kalman filter with Global Nearest Neighbor (GNN) | Multi-Target Tracking (MTT) | Occlusion handling | Level 2, 3, 4 |
| Collins et al. [53]      | Gait databases. CMU, MIT, UMD, USH | Silhouette based model | Tracking pedestrian body state | To identify people from their body and gait | Level 2   |
| Schwarz et al. [54]      | Kinect data               | Graph method with skeleton fitting  | Tracking pedestrian body state | Full-body tracker | Level 2   |
| Sinthanayothin et al. [55] | Kinect data               | Skeleton tracking                   | Tracking pedestrian body state | Review paper    | Level 2   |
| Konstantinova et al. [56] | 5 test matrices          | Global Nearest Neighbor with Munkres algorithm | Multi-Target Tracking (MTT) |                 | Level 2, 3, 4 |
| Azari et al. [57]        | IBM, PETS2000 and PETS2001 databases | Kalman filter with Global Nearest Neighbor (GNN) | Multi-Target Tracking (MTT) | Occlusion handling | Level 2, 3, 4 |
| Reid et al. [58]         | Monte Carlo simulation   | Iterative algorithm with Multi-Hypothesis Tracking(MHT) | Multi-Target Tracking (MTT) | Occlusion handling | Level 2, 3 |
| Luber et al. [59]        | Two datasets collected in indoor and outdoor environments | Social force with Multiple Hypothesis Tracking (MHT) | Multi-Target Tracking (MTT) |                 | Level 3   |
| Kim et al. [60]          | PETS and MOTChallenge benchmarks | Multiple Hypothesis Tracking (MHT) | Multi-Target Tracking (MTT) |                 | Level 2   |
| Zhou et al. [61]         | Computer simulations      | Joint probabilistic data association filter (JPDAF) with a depth-search approach | Multi-Target Tracking (MTT) |                 | Level 2, 3 |
| Chen et al. [62]         | Video data                | Contour based tracker with JPDAF and IMM | Multi-Target Tracking (MTT) | Real-time method | Level 2, 3 |
| Liu et al. [63]          | Simulations and real robot environments | Sample-based JPDAF and multi-camera fusion | Multi-Target Tracking (MTT) | Real-time method | Level 2, 3 |
| Horridge et al. [64]     | Simulations               | JPDAF based tracker                 | Multi-Target Tracking (MTT) | 400 tracks in real-time | Level 2, 3, 4, 5 |
| Rezatofighi et al. [65]  | Fluorescence microscopy sequences and surveillance camera data | JPDAF based tracker | Multi-Target Tracking (MTT) |                 | Level 2, 3 |
| Zhang et al. [66]        | Simulations               | Gaussian Mixture Measurement PHD tracker (GMM-PHD) | Multi-Target Tracking (MTT) | Handle bearing measurements | Level 2, 3 |
| Khaare et al. [67]       | Data from a distributed network of cameras | Probabilistic Hypothesis Density (PHD) filter based tracker | Multi-Target Tracking (MTT) |                 | Level 2, 3 |
| Feng et al. [68]         | Simulations with sequences from CAVIAR dataset | Variational Bayesian PHD filter with deep learning updates | Multi-Target Tracking (MTT) |                 | Level 2, 3 |
| Correa et al. [14]       | Tested on a real-time crowded environment | PHD filter | Multi-Target Tracking (MTT) |                 | Level 4, 5 |
### TABLE II: Summary of pedestrian tracking models

| Study/Paper | Input/Evaluation | Method | Tracking Models | Additional Info | SAE Level |
|-------------|------------------|--------|-----------------|-----------------|-----------|
| Yoon et al. [69] | ETH dataset | Sequential Monte Carlo PHD filter (SMC-PHD) | Multi-Target Tracking (MTT) | Can handle missing detections | Level [2,3] |
| Oh et al. [70] [71] | Simulations | Markov Chain Monte Carlo Data Association (MCMCDA) Metropolis-Hastings method | Multi-Target Tracking (MTT) | | Level [2,3] |
| Yu et al. [72] | Simulations and video data | Data-driven Markov Chain Monte Carlo data association (DD-MCMCDA) | Multi-Target Tracking (MTT) | | Level [2,3] |
| Chen et al. [73] | Video data | Dynamical graph matching | Multi-Target Tracking (MTT) | Tracker can deal with interactions | Level [2,3] |
| Pirsiavash et al. [74] | Video data | Greedy algorithm based on Dynamic Programming | Multi-Target Tracking (MTT) | | Level [2,3] |
| Zhang et al. [75] | CAVIAR and ETHMS datasets | Min-Cost Flow algorithm with an explicit occlusion model (EOM) | Multi-Target Tracking (MTT) | Occlusion handling | Level [2,3] |
| Chari et al. [76] | PETS and TUD datasets | Min-Cost Max-Flow network optimization with pair-wise costs | Multi-Target Tracking (MTT) | Occlusion handling | Level [2,3] |
| Taycher et al. [77] | Video data | Conditional Random Field (CRF) state estimation and grid filter | Multi-Target Tracking (MTT) | Real-time capability | Level [2,3] |
| Milan et al. [78] | PETS 2010 Benchmark and TUD-Stadtmute dataset | CRF-based multiple tracker with HOG-SVM detector | Multi-Target Tracking (MTT) | | Level [2,3] |
| Milan et al. [79] | PETS, TUD, ETHMS datasets | CRF-based multi-target tracker using discrete continuous minimization | Multi-Target Tracking (MTT) | Trajectory estimation of targets | Level [2,3,4] |
| Brendel et al. [80] | ETHZ Central, TUD Crossing, i-Lids AB, UBC Hockey and ETHZ Soccer datasets | Maximum-weight independent set (MWIS) based tracker | Multi-Target Tracking (MTT) | Long-term occlusion handling | Level [2,3] |
| He et al. [81] | PEITSO, TUD Statmitte, TUD Crossing and ETHMS datasets | Connected component model with MWIS | Multi-Target Tracking (MTT) | | Level [2,3] |
| Gaidon et al. [82] | KITTI Benchmark and PASCAL-to-KITTI dataset | Online Domain Adaptation for Multi-Object Tracking | Multi-Target Tracking (MTT) | Generic detector and video adaptation for tracking | Level [2,3] |
| Ondruska et al. [83] | Simulated data | End-to-end recurrent neural network (RNN) tracker | Multi-Target Tracking (MTT) | No data association required | Level [2,3] |
| Dequaire et al. [84] | Real-world environment | End-to-end recurrent neural network (RNN) tracker | Multi-Target Tracking (MTT) | | Level [2,3] |
| Milan et al. [85] | MOTChallenge 2015 benchmark | Online recurrent neural network (RNN) tracker | Multi-Target Tracking (MTT) | | Level [2,3] |
| Ristani et al. [86] | Multi-cameras system data | Neural networks | Multi-Target Tracking (MTT) | Features learnt multi-cameras and Re-identification | Level [3, 4] |
| Liu et al. [87] | Multi-camera systems data | Generalized Maximum Multi-Clique optimization | Multi-Target Tracking (MTT) | | Level [2,3] |
| Park et al. [88] | Monocular camera data | 3D object tracking | Multi-Target Tracking (MTT) | 3D object Tracking for augmented reality applications | Level [2,3] |
| Schetsdeger et al. [89] | Single camera data | Multi-Bernoulli mixture tracking filter | Multi-Target Tracking (MTT) | | Level [2,3,4] |
| Yan et al. [90] | Lidar data | 3D LIDAR based tracking with Support Vector Machine (SVM) classifier | Multi-Target Tracking (MTT) | Online classification of humans | Level [2,3] |
| Leal-taixe et al. [91] | Camera data | Interaction feature strings with Random Forest method | Multi-Target Tracking (MTT) | Scene understanding | Level [2,3] |
| Choi et al. [92] | Video data | Discriminative model | Multi-Target Tracking (MTT) | Group activity recognition | Level [2,3] |
IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. X, NO. X, X 20XX

S. Oh, S. Russell, and S. Sastry, “Markov chain monte Carlo data association for general multiple-target tracking problems,” in 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601), vol. 1, pp. 735–742 Vol.1, Dec 2004.

Q. Yu, G. Medioni, and I. Cohen, “Multiple target tracking using spatio-temporal markov chain monte Carlo data association,” in 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2007.

H.-T. Chen, H.-H. Lin, and T.-L. Liu, “Multi-object tracking using dynamical graph matching,” in null, p. 210, IEEE, 2001.

H. Pirsavash, D. Ramanan, and C. C. Fowlkes, “Globally-optimal greedy algorithms for tracking a variable number of objects,” in CVPR 2011, pp. 1201–1208, June 2011.

L. Zhang, Y. Li, and R. Nevatia, “Global data association for multi-object tracking using network flows,” in 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2008.

V. Chari, S. Lacoste-Julien, I. Laptev, and J. Sivic, “On pairwise costs for network flow multi-object tracking,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

L. Taycher, D. Demirdjian, T. Darrell, and G. Shakhnarovich, “Conditional random people: Tracking humans with crfs and grid filters,” in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), vol. 1, pp. 798–805, June 2006.

F. Meyer and P. Bouthemy, “Region-based tracking in an image sequence,” in European Conference on Computer Vision, pp. 476–484, Springer, 1992.

R. T. Collins, R. Gross, and J. Shi, “Silhouette-based human identification from body shape and gait,” in Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition, pp. 366–371, May 2002.

L. A. Schwarz, A. Mkhitarian, D. Mateus, and N. Navab, “Human skeleton tracking from depth data using geodesic distances and optical flow,” Image and Vision Computing, vol. 30, no. 3, pp. 217–226, 2012.

C. Srinathanoyin, N. Wongwaen, and W. Bholsthi, “Skeleton tracking using kinect sensor & displaying in 3 d virtual scene,” International Journal of Advancements in Computing Technology, vol. 4, no. 11, 2012.

P. Konstantinova, U. Dvurev, and T. Semerdjiev, “A study of a target tracking algorithm using global nearest neighbor approach,” in Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech’01), pp. 290–293, 2001.

M. Azari, A. Seyfii, and A. H. Rezaie, “Real time multiple object tracking and occlusion reasoning using adaptive kalman filters,” in Machine Vision and Image Processing (MVIP), 2011 7th Iranian, pp. 1–5, IEEE, 2011.

D. Reid, “An algorithm for tracking multiple targets,” IEEE transactions on Automatic Control, vol. 24, no. 6, pp. 843–854, 1979.

M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, “People tracking with human motion predictions from social forces,” in Proc. of the Int. Conf. on Robotics & Automation (ICRA), (Anchorage, USA). 2010.

C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, “Multiple hypothesis tracking revisited,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 4866–4704, 2015.

B. Zhou and N. Bose, “Multitarget tracking in clutter: Fast algorithms for data association,” IEEE Transactions on Aerospace and electronic systems, vol. 29, no. 2, pp. 352–363, 1993.

Y. Chen, Y. Rui, and T. S. Huang, “Jpda based hmm or real-time contour tracking,” pp. 543, IEEE, 2001.

N. Liu, R. Xiong, Q. Li, and Y. Wang, “Human tracking using improved sample-based joint probabilistic data association filter,” in Intelligent Autonomous Systems 12 (S. Lee, H. Cho, K.-J. Yoon, and J. Lee, eds.), (Berlin, Heidelberg), pp. 293–302, Springer Berlin Heidelberg, 2013.

P. Horridge and S. Maskell, “Real-time tracking of hundreds of targets with efficient exact jpdaf implementation,” in Information Fusion, 2006 9th International Conference on, pp. 1–8, IEEE, 2006.

S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, “Joint probabilistic data association revisited,” in 2015 IEEE International Conference on Computer Vision (ICCV), pp. 3047–3055, Dec 2015.

Q. Zhang and T. L. Song, “Improved bearings-only multi-target tracking with gm-phd filtering,” Sensors, vol. 16, no. 9, p. 1469, 2016.

M. Khazaie and M. Jamzad, “Multiple human tracking using phd filter in distributed camera network,” in 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 569–574, Oct 2014.

P. Feng, W. Wang, S. M. Naqvi, and J. Chambers, “Variational bayesian phd filter with deep learning network updating for multiple human tracking,” in 2015 Sensor Signal Processing for Defence (SSPD), pp. 1–5, Sept 2015.

J. H. Yoon, K. J. Yoon, and D. Y. Kim, “Multi-object tracking using hybrid observation in phd filter,” in 2013 IEEE International Conference on Image Processing, pp. 3890–3894, Sept 2013.

S. Oh, S. Russell, and S. Sastry, “Markov chain monte Carlo data association for multi-target tracking,” IEEE Transactions on Automatic Control, vol. 54, no. 3, pp. 481–487, 2009.
Fanta Camara (STMIEEE) is a PhD student at the Institute for Transport Studies (ITS), University of Leeds, UK, a Visiting Researcher in the School of Computer Science, University of Lincoln, UK, a member of Lincoln Centre for Autonomous Systems (L-CAS) and a member of Robotics at Leeds. She received her Robotics Engineering degree (Master’s) from Sorbonne Université (former Université Pierre et Marie Curie, UPMC), Paris, France. She worked as an intern at Airbus Defence and Space, National Taiwan University (Taipei, Taiwan), DLR (German Aerospace Center) and Robert Bosch GmbH. Her main research interests are in human-robot interaction for autonomous systems, game theory, computer vision, ML/DL and robotics.

Nicola Bellotto (SMIEEE) is an Associate Professor in the School of Computer Science, University of Lincoln, UK, and a member of the Lincoln Centre for Autonomous Systems. His main research interests are in machine perception, especially for human detection, tracking, identification and activity recognition with autonomous mobile robots. He has a Master in Electronic Engineering from the University of Padua, Italy, and a PhD in Computer Science from the University of Essex, UK. Before joining the University of Lincoln, he was a researcher in the Active Vision Lab at the University of Oxford. Dr Bellotto is the recipient of a Google Faculty Research Award and a Ph.D/Co-I in several research projects on autonomous mobile robots.

Serhan Cosar is currently a Senior Lecturer in Robotics at De Montfort University. He received his M.Sc. and Ph.D. degrees of Electronics Engineering and Computer Science from Sabanci University, Istanbul, Turkey in 2008 and 2013, respectively. Before joining De Montfort University, he worked as a post-doctoral research fellow at University of Lincoln, UK. He is the author of 3 book chapters, 9 international journal papers (CVIU, IEEE-TCSVT) and 16 international conference papers (IROS, ICCV, AVSS). He has participated in a number of European and National research projects. His research interest includes robot perception, human tracking, activity recognition, sparse representation, and distributed estimation.

Dimitris Nathanael is an assistant professor at the National Technical University of Athens, School of Mechanical Engineering, where he teaches ergonomics / human factors, design innovation and safety management courses. His scientific interests span across all facets of the interaction of humans with technology with an emphasis on cognition. Dimitris holds a diploma in Production Engineering & Management science, Technical University of Crete, a D.E.A. in Ergonomics, from the Conservatoire National des Arts et Métiers (C.N.A.M.) in France, and a PhD in the field of Cognitive Ergonomics, from the National Technical University of Athens. Before joining the academia, Dimitris has worked in for several years in industry as a consultant in ergonomics, human-computer interaction and occupational safety. He is the President of the Hellenic Ergonomics Society and the national representative in the International Ergonomics Association.

Matthias Althoff is an assistant professor in computer science at Technische Universität München, Germany. He received his diploma engineering degree in Mechanical Engineering in 2005, and his Ph.D. degree in Electrical Engineering in 2010, both from Technische Universität München, Germany. From 2010 to 2012 he was a postdoctoral researcher at Carnegie Mellon University, Pittsburgh, USA, and from 2012 to 2013 an assistant professor at Technische Universität Ilmenau, Germany. His research interests include formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, robotics, automated vehicles, and power systems.

Jingyuan Wu is pursuing his PhD degree with Chassis Systems Control, Robert Bosch GmbH in Abstatt, Germany, in cooperation with the Technical University of Munich, Germany. He received his B.Sc. degree in mechanical engineering from University of Science and Technology Beijing, China in 2013 and M.Sc. degree in mechanical engineering from Technical University of Kaiserslautern, Germany in 2017. His work is focused on signal processing and in particular pedestrian motion prediction in urban environments.

Johannes Ruenz is currently working in the advanced development department of Robert Bosch GmbHs Chassis Systems division. He studied Electrical Engineering and Information Technologies at the Karlsruher Institute of Technology (KIT) and received his master degree in 2014. His research topic is focused on active pedestrian protection systems using their smartphones and vehicles environment sensors.

André Dietrich is a research associate and PhD student at the Chair of Ergonomics at the Technical University of Munich. His research focuses on interaction between automated vehicles and pedestrians with an emphasis on the effects of visual communication. He currently leads Work package 2 "Psychological Models on Human Interaction & Intention Recognition Algorithms" in the European Project interACT. He graduated in Aerospace Engineering at the Technical University of Munich.

Charles W. Fox is a Senior Lecturer in Robotics and Automation at the University of Lincoln and a Visiting Lecturer in Vehicle and Road Automation at the Institute for Transport Studies at the University of Leeds. He received his MA in Computer Science from the University of Cambridge, MSc in Cognitive Science from the University of Edinburgh, and DPhil in Engineering from the University of Oxford. He is the author of the Springer textbook “Data Science for Transport: a self-study guide with computer exercises".