Awareness of Road Scene Participants for Autonomous Driving

Anna Petrovskaya*, Mathias Perrollaz†, Luciano Oliveira‡, Luciano Spinello**, Rudolph Triebel††, Alexandros Makris†, John-David Yoder‡‡, Christian Laugier†, Urbano Nunes‡, Pierre Bessiere†

*Stanford University, USA, †INRIA, France, ‡Coimbra University, Portugal, **University of Frieburg, Germany, ††University of Oxford, UK, ‡‡Ohio Northern University, USA

October 12, 2011

Abstract

This chapter describes detection and tracking of moving objects (DATMO) for purposes of autonomous driving. DATMO provides awareness of road scene participants, which is important in order to make safe driving decisions and abide by the rules of the road. Three main classes of DATMO approaches are identified and discussed. First is the traditional approach, which includes data segmentation, data association, and filtering using primarily Kalman filters. Recent work within this class of approaches has focused on pattern recognition techniques. The second class is the model based approach, which performs inference directly on the sensor data without segmentation and association steps. This approach utilizes geometric object models and relies on non-parametric filters for inference. Finally, the third class is the grid based approach, which starts by constructing a low level grid representation of the dynamic environment. The resulting representation is immediately useful for determining free navigable space within the dynamic environment. Grid construction can be followed by segmentation, association, and filtering steps to provide object level representation of the scene. The chapter introduces main concepts, reviews relevant sensor technologies, and provides extensive references to recent work in the field. The chapter also provides a taxonomy of DATMO applications based on road scene environment and outlines requirements for each application.

Contents

1 Introduction

Autonomous driving in populated areas requires great situational awareness. The autonomous vehicle must perceive not only the stationary environment, but also dynamic objects such as vehicles and pedestrians. For each moving target, the autonomous vehicle needs to identify location and velocity, so that it can predict the target's position a few seconds later for planning purposes. Awareness of moving objects includes both detection of new targets and tracking of existing targets over time. For this reason it is often referred to as *detection and tracking of moving objects*, or *DATMO* for short.

The need for DATMO was born with the first interest in intelligent and autonomous vehicles in 1980s. A large exploratory project was launched in Europe in 1986 under the name PROMETHEUS, followed by a number of initiatives in Japan and United States (Bertozzi et al., 2000; Sun et al., 2006). Fueled by innovations in sensor technologies, recent DATMO advances have focused on improved detection using pattern recognition techniques, increased robustness using sensor fusion, and more accurate modeling of sensors and the scene. In this chapter we give an overview of the most prominent DATMO approaches and relevant concepts.

The remainder of this section provides a taxonomy of DATMO applications and gives a formal description of the DATMO problem. Sect. 2 introduces the required concepts as well as describes sensors and models. Sect. 3 discusses inference techniques for DATMO. In particular it outlines three main classes of DATMO approaches: traditional, model based, and grid based. Sect. 4 covers pattern recognition, and sensor fusion. The chapter concludes with a discussion in Sect. 5.

1.1 Taxonomy of DATMO Applications

Although detection of people and vehicles has received a lot of attention in other fields, this chapter focuses on techniques that satisfy the high demands of autonomous driving. To illustrate these demands, the driving applications are roughly grouped into three categories: (1) pedestrian zone driving, (2) freeway driving, and (3) mixed urban driving.

In the first category, the robot operates in a pedestrian zone, which can be crowded by bicyclists, adults, children, and pets. The robot has to operate in close proximity to these inhabitants in a safe and courteous manner. This situation is challenging because the targets come in all shapes and sizes, can change direction of motion unpredictably, and can partially occlude themselves or each other. The operational velocity has to be similar to pedestrian speed: 1–2m/s. Since the operational velocities are relatively low, the required range of target detection and tracking is relatively short: 10–20m is usually sufficient for safe operation. On the other hand, due to close proximity to targets, the robot's reaction time has to be similar to human reaction time, which is usually taken to be 1s. Hence detection and tracking have to happen fast enough, so that the entire perception-planning-control loop can be executed at 1Hz. However, to simplify the association stage tracking at 5–10Hz is desirable.

In the second category, the robot drives on a freeway, which is a much more structured environment than a pedestrian zone. The environment is populated only by vehicles, which have to travel within lanes most of the time. Oncoming vehicles are confined to a separate part of the freeway, so the robot only needs to concern itself with vehicles moving in the same direction. The motion of these vehicles is governed by non-holonomic constraints of vehicle dynamics and therefore is much more predictable than motion of pedestrians. The main challenge in this environment comes from high operational velocities: up to 35m/s. High operational velocity leads to longer stopping distances. Thus the required range for detection and tracking of targets is much greater than in the first category: 100m or more. In order to plan smooth maneuvers at high speeds, the planning loop runs at 10–20Hz and hence tracking also needs to happen at a comparable rate.

The third category lies in between the first two. The robot operates in an urban setting with mixed pedestrian and vehicle traffic. This category combines challenges of the first two categories. The environment is not as structured as on the freeway — vehicles and pedestrians can cross the robot's path in unpredictable ways. Oncoming traffic is usually present, which doubles the effective operational velocity from the perspective of object detection. If the speed limit is 15m/s (35mph), then an oncoming vehicle can move with a velocity of up to 30m/s with respect to the robot. Hence the detection range and tracking frequency rate have to be almost as high as on freeways: 60–80m and 10Hz respectively. In addition, the robot has to be able to distinguish between different types of targets: pedestrians, bicyclists, and vehicles. These three types of targets are governed by different traffic and dynamic laws and hence the robot needs to treat them differently.

1.2 DATMO Problem Statement

In DATMO, an autonomous vehicle (also called *robot* or *ego-vehicle*) navigates in a populated outdoor environment. Sensors are mounted on the ego-vehicle itself, which is potentially moving at high speeds. The robot is to perceive the environment around it based on its sensors and to detect and track all moving objects in its vicinity. For high level decision making, the robot needs to estimate pose and velocity of each moving object based on the sensor measurements. For low level trajectory planning, the robot needs to estimate the free space for driving.

Object pose estimates (also called *tracks*) need to be produced at a high enough rate to be useful for the perception-planning-control loop, which typically runs at 1–10Hz. *False negatives* (i.e. missing an object) tend to be very dangerous, whereas *false positives* (i.e. finding an object when one is not there) are slightly more acceptable. Note that for the applications in *advanced driver assistance systems* (*ADAS*), it is the opposite.

1.2.1 Coordinate Frames

This chapter assumes that a reasonably precise pose of the robot is always available. Further, it assumes the use of *smooth coordinates*, which provide a locally consistent estimate of the ego-vehicle motion. Smooth coordinates should not experience sudden jumps because jumps can greatly increase tracking uncertainty. In practice, smooth coordinates can be obtained by integrating the ego-vehicle velocity estimates from the inertial navigation system (Montemerlo et al., 2008; Leonard et al., 2008). To map from smooth coordinates to globally consistent GPS coordinates, one simply needs to add an offset, which is periodically updated to reflect the mismatch between the smooth and GPS coordinate systems. In the remainder of this chapter all operations are carried out in the smooth coordinate frame, which will also be called the *world frame*. The transformation from smooth to GPS coordinates is only needed when dealing with global features, such as the digital road map.

It is also common to use local coordinate systems, which are tied to the robot, the sensor, or tracked objects. These are called the *robot coordinate frame*, the *sensor coordinate frame*, and the *object coordinate frame* respectively.

1.2.2 Bayesian Formulation of DATMO

For a general Bayesian problem, the goal is to infer the state $\mathbf{S}$ of a system (or of the world) based on a set of measurements Z. Due to uncertainty, this information is best captured as a probability distribution $bel := p(\mathbf{S}|Z)$ called the *posterior distribution* or the *Bayesian belief*.

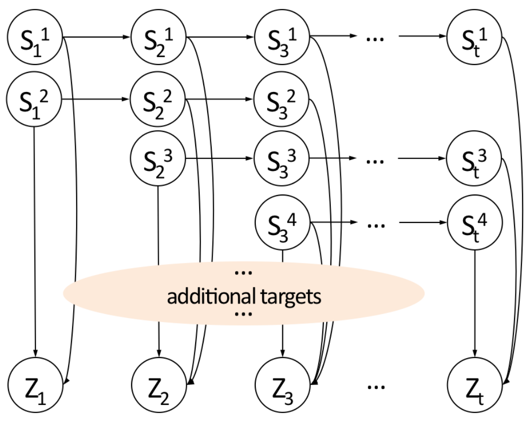
In a dynamic Bayesian system, the state changes over time, which is assumed to be discretized into small (although not necessarily equal) time intervals. The system is assumed to evolve as a *Markov process* with unobserved states. The goal is to estimate the belief at time t, $bel_t := p(\mathbf{S}_t|Z_1,\ldots,Z_t)$. The behavior of the system is described via two probabilistic laws: (i) the *measurement model* $p(Z|\mathbf{S})$ captures how the sensor measurements are obtained and (ii) the *dynamics model* $p(\mathbf{S}_t|\mathbf{S}_{t-1})$ captures how the system evolves between time steps. Given these two laws and measurements up to time t, the belief can be computed recursively using a *Bayesian filter* algorithm, which relies on the *Bayesian recursion* equation

$$bel_t = \eta\, p(Z_t|\mathbf{S}_t) \int p\,(\mathbf{S}_t|\mathbf{S}_{t-1})\, bel_{t-1}\, \mathrm{d}\mathbf{S}_{t-1}, \qquad (1)$$

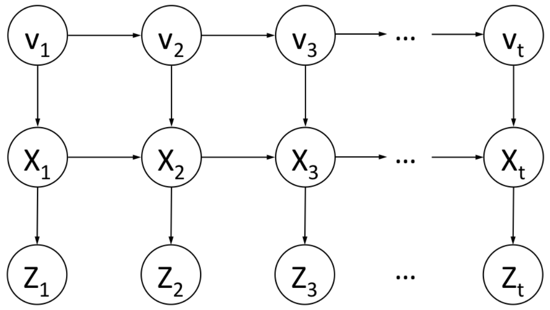
where η denotes the normalization constant.

For the DATMO problem, the system consists of a set of moving targets $T^1,\ldots,T^{K_t}$. The number of targets K_t changes over time as some targets leave and others enter the scene. For each target, the estimated parameters include its momentary 2D pose $X_t = (x_t, y_t, \theta_t)$ consisting of its 2D position (x_t, y_t) and orientation θ_t at time t. The parameters also include each target's forward velocity v_t, which is typically a scalar as the objects are assumed to move along vectors aligned with their orientation. Thus, the Bayesian state for a single target is usually $S_t := (X_t, v_t)$, although in some cases additional parameters may be added to describe shape, class, or motion of objects. The full Bayesian state consists of the states of all the targets: $\mathbf{S}_t := (S_t^1,\ldots,S_t^{K_t})$. At each time step we obtain a new sensor measurement Z_t.

A graphical representation of the resulting probabilistic model for DATMO is shown in Fig. 1. Note that during filtering the targets are usually considered independent of each other, although some relationships clearly exist in reality. The independence assumption allows the problem to be split into multiple smaller sub-problems: one per target. This reduces the effective dimensionality that the estimation algorithms need to cope with. Relationships between targets are often introduced as a set of constraints, which are imposed after each filtering step as we discuss in Sect. 3.1.



(a) for all targets



(b) for a single target

Figure 1: Graphical representation of the probabilistic model for DATMO. (a) For the full DATMO problem the graphical model shows a variable number of targets. (b) For a single target, the graphical model shows the relationships between the target's pose X_t, forward velocity v_t, and measurements Z_t.

For a single target, the dependencies between the parameters are typically modeled via several probabilistic laws: the *velocity model* $p(v_t|v_{t-1})$, the *motion model* $p(X_t|X_{t-1}, v_t)$, and the measurement model $p(Z_t|X_t)$. The velocity and motion models together comprise the dynamics model. Measurement and dynamics models for DATMO are discussed in Sect. 2.2.

2 Sensors and Models

2.1 Sensors

The most common sensors used in DATMO approaches are *optical cameras* and *laser range finders*, although some systems also incorporate *radar* sensors. In this subsection, we discuss all three types of sensors, their operating characteristics, advantages, and limitations.

2.1.1 Optical Cameras

Cameras are the most popular sensors due to their low cost and high information content. Two major camera technologies are available: *charge-coupled devices* (*CCD*) and *complementary metal oxide semiconductors* (*CMOS*). CCD cameras tend to have a higher output uniformity because all pixels can be devoted to capture light. In contrast, CMOS cameras are usually more compact and require less off-chip circuitry. An important parameter for cameras is their *field of view* (*FOV*), which is directly defined by the optics used and by the size of the sensor's matrix. A very narrow field of view is achieved with a *tele-lens* and a very wide field of view results from using a *fish-eye lens*. While in the first case, far objects can be observed at a higher resolution and nearly no line distortion, in the second case, a much larger fraction of the environment can be observed within a single image.

Cameras are very attractive sensors because they capture high resolution images of the environment at high frame rates, while consuming very little energy. These sensors produce high volumes of data with high information content. While this is an important advantage over other sensors, high volumes of data also lead to significant challenges in transmission and processing of the data stream. Moreover, since cameras are not able to capture range to objects, the data produced by cameras are more difficult to interpret than range finder data. Cameras are also greatly impacted by changes in lighting conditions, shadows, and other types of ambient effects. Finally, unlike radars and lasers, cameras do not provide useful data after dark, when the lighting of the scene is insufficient.

In addition to ordinary *monocular cameras*, several special camera types and configurations are employed. *Stereo cameras* make use of two (or more) cameras for binocular vision, which is able to produce range data in addition to ordinary monocular images. This is a cost-effective sensor for acquiring range data, but it can be very sensitive to calibration errors. One popular example is the Bumblebee camera by Point Grey. Omni-directional cameras capture 360° view of the environment. They are constructed either by using multiple mono cameras (e.g. the Ladybug camera) or mirrors. The multiple camera approach leads to even greater volumes of data, whereas the mirror approaches can lead to a loss of resolution. Infra red cameras perceive thermal energy emitted by objects in the environment. Although these sensors can simplify detection of people, the signal to noise ratio is very high and the range is limited to short distances.

2.1.2 Laser Range Finders

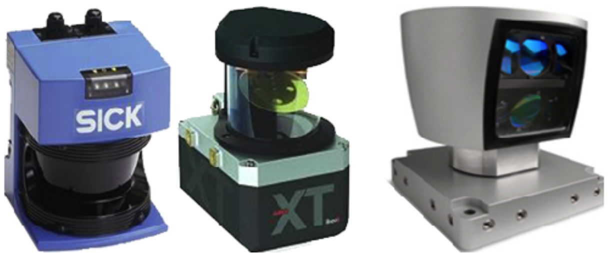


Figure 2: Laser range finders. From left to right: 2D laser (Sick LMS200), multilayer 2D laser (IBEO Alasca XT), 3D laser (Velodyne HDL-64E).

Laser range finders retrieve distances by measuring *time-of-flight* of laser beam returns. Several types are available on the market: 2D, multilayer 2D, and 3D (Fig. 2). The principle of operation of 2D lasers is based on a rotating mirror that deflects a beam into a plane. The reflected beam is detected by a receiver circuit that also computes the travelled distance using the time elapsed from beam emission to detection. A very wide field of view is retrieved in this manner (at least 120°), but the scan only captures information about a 2D slice of the world. Measurements are very precise, but sparse due to limited angular resolution (0.25°–1°). Laser range finders work in any lighting conditions, although direct sunlight can cause false measurements and errors. For DATMO applications, the 2D lasers are typically mounted horizontally on the ego-vehicle. This produces scan lines parallel to the ground. A significant challenge is to tell the difference between ground readings vs. readings obtained from true obstacles.

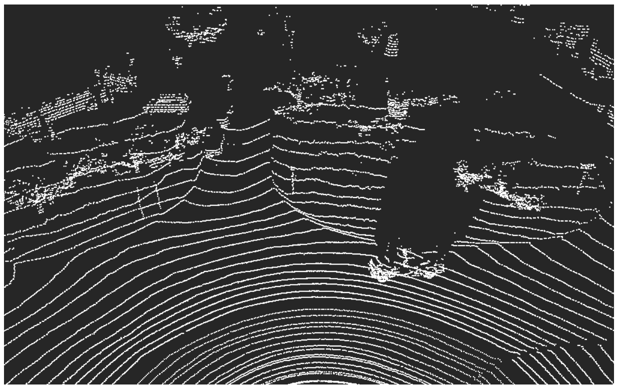


Figure 3: Example of a 3D scan obtained with a Velodyne HDL-64E scanner. See Fig. 5 for a picture of the scene.

Recently lasers have evolved from 2D to multilayer 2D devices, which are more suitable for driving applications. The principle of operation is similar to a standard 2D scanner, but more slices are retrieved (slices are almost parallel 1°– 2°) by employing more beams and more receivers. For example, IBEO Alasca sensors allow for easy ground filtering by collecting four nearly-parallel horizontal scan lines. Using information from all four scan lines it is easier to tell which readings likely came from the ground, by considering the vertical alignment of impacts and, hence, these sensors can provide automatic ground filtering.

Another recent innovation are 3D laser range finders, e.g. the Velodyne HDL-64E sensor. Data are retrieved at 5–15Hz with up to two million points per second (see Fig. 3). Even though angular resolution is not as fine as for a camera, this sensor has all the aforementioned advantages of a range device. This sensor has 64 lasers aligned at different vertical angles on a cylinder rotating around a vertical axis. On each rotation, lasers sweep the space and a 3D point cloud is received. The first successful usage of this device in the field of vehicle detection has been presented in the DARPA Urban Challenge in 2007. Given the rich data produced by 3D scanners, the challenge has become to process the readings in real time, as target tracking at 10–20Hz is desirable for driving decision making.

Overall, lasers can provide data at high frame rates and with high information content (especially 3D lasers). Unlike cameras, lasers give range information for every data point and can work in the dark. However, compared to cameras, lasers are much more expensive and consume more energy, which is required for sending out laser light pulses. Their angular resolution is not as high as cameras and they do not provide color information. Compared to radar, lasers can be affected by weather conditions, such as rain or fog.

2.1.3 Radars

Radar sensors measure distance to obstacles by sending out radio waves and measuring either time delay or phase shift of the reflected signals. The typical radars used for autonomous driving applications are *millimeter wave* (*MMW*) radars, which have started to appear on some upscale commercial vehicles for *adaptive cruise control*.

Although radar speed guns are commonly used by law enforcement officers to detect velocity of speeding vehicles, this technology is more difficult to use on an autonomous platform. Radars tend to have narrow field of view and low resolution for position measurements, although they are very precise for velocity estimation. The measured range can be corrupted by echoes from multiple objects in the path of the wave and even by highly reflective objects outside of the FOV. Some targets get multiple detections, others go undetected, and the bearing to each target is not very accurately determined. For these reasons, radars are often combined with vision sensors to obtain better accuracy (Lundquist and Schon, 2008; Richter et al., 2008). Since static clutter and infrastructure cause echoes, it is common to filter out static returns from radars, making these sensors better suited for highway rather than urban applications.

Despite all the challenges, radars perform better than other sensors in adverse weather conditions and have longer detection range. Hence, these sensors warrant further evaluation for the purposes of DATMO.

2.2 Models of Sensors and Motion

Measurement models (also called *sensor models* or *observation models*) define the measurement process in probabilistic terms. In other words, given a pose X_t of a tracked object, the measurement model defines the probability of obtaining a specific set of sensor measurements Z_t. Of course, it is not possible to model the sensor operation exactly as many unknown effects can impact the performance of the sensor: lighting conditions, dust or fog for example. Hence, measurement models explicitly represent some properties of the sensors and consider the unmodeled effects as uncertainty or noise.

Measurement models in DATMO fall within a spectrum between *physical sensor models* and *pseudo-position models*. The first type attempts to model the physical sensor operation directly by considering the physical phenomena that occur when the sensor interacts with the world (e.g. laser rays travel through space and get reflected from objects). These methods are appropriate when the physical phenomena are easy to model (i.e. usually for range sensors rather than classical vision). Examples of physical sensor models are discussed in Sect. 2.2.1 below.

On the opposite side of the spectrum are methods, which derive target positions (and sometimes orientations and velocities) from sensor data prior to the application of the Bayesian filter. Then the probabilistic sensor model within the Bayesian filter represents the resulting *pseudo sensor* of target positions (Sect. 2.2.2).

Most sensor model approaches will use at least some data pre-processing techniques to enhance the data. When the resulting data are still far from the quantities to be estimated, a lot of work still remains for DATMO algorithms. The sensor data at this stage shall be called *virtual sensor* data. Virtual sensor techniques can be very light (e.g. apply a Gaussian blur) or they can build a qualitatively different sensor (e.g. stereo vision). The important distinction from pseudo-position sensors, is that virtual sensors are still *relative sensors*, in that they do not provide direct measurements of quantities to be estimated. Virtual sensor techniques are discussed in Sect. 2.2.3.

2.2.1 Physical Models for Range Sensors

Physical sensor models are most common for range sensors. Two main types are prevalent: the *proximity model* and the *independent beam model*. In the proximity model, each range measurement is converted into a 3D point in the world frame. These points are considered to be caused by objects in the environment (both static and dynamic). The closer the points are to the surface of the objects the more likely the measurements are, hence the name proximity model. Proximity models are fast to compute, but they discard *negative information*, i.e. the information that the space the laser rays travelled through must be unoccupied. This information is taken into account by the *independent beam model* (*IB*). In the IB model, ray tracing is used to identify range to the closest obstacle along each laser ray. The obtained range is compared to the range returned by the sensor. The closer together these range values the more likely the measurements. Both of these models have been used with great success in SLAM, where proximity models are often called *likelihood fields*. See (Thrun et al., 2005) for more information.

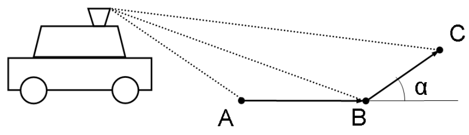


Figure 4: Ground readings can be determined by comparing angles between consecutive readings. If A, B, C are ground readings, then α is close to 0 and thus $\cos\alpha$ is close to 1.

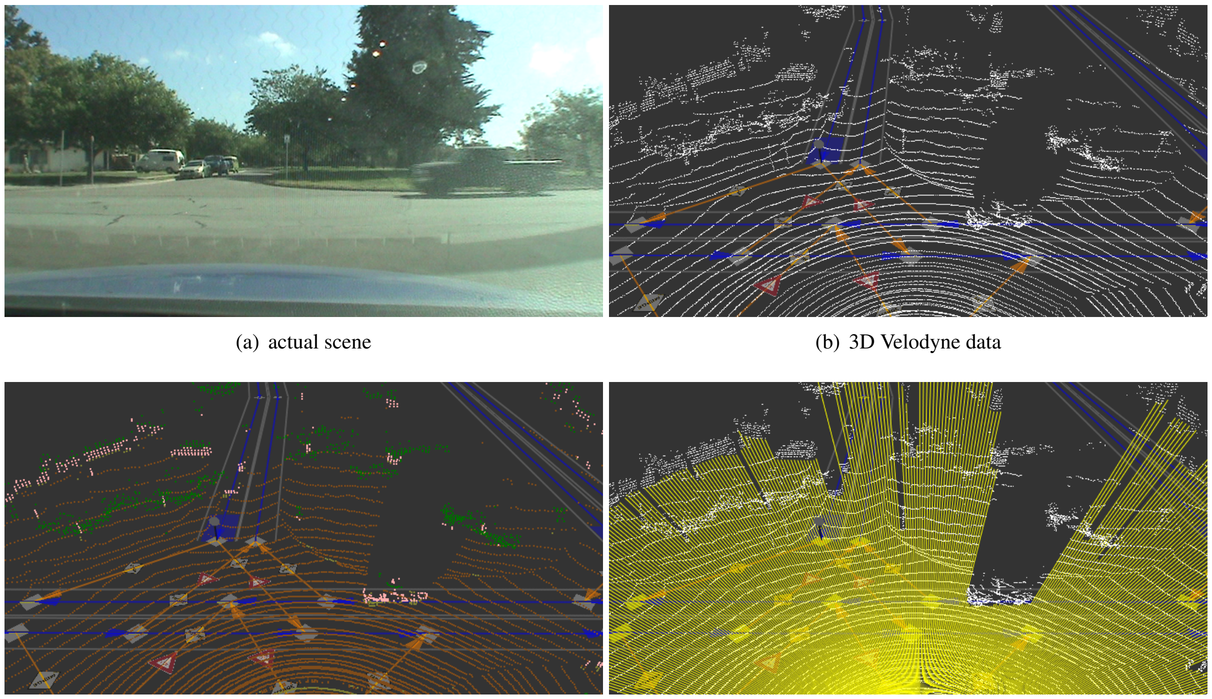
2.2.2 Pseudo-Position Sensors

Most of the sensors used for DATMO provide only relative information about the targets, e.g. individual range measurements or pixels. The quantities that need to be estimated are positions, orientations and velocities of targets. Many DATMO approaches augment the physical sensors with a set of algorithms (as discussed in Sect. 3.2) to produce pseudo measurements of target positions (and in some cases orientations and velocities). The resulting pseudo measurements are modeled as direct measurements of target positions, corrupted by zero-mean Gaussian noise.

2.2.3 Virtual Sensors

It is often useful to augment the physical sensor by a number of data pre-processing or clean-up techniques. A variety of methods can be used.

For range data it is common to sub-sample the rays,


(a) actual scene

(b) 3D Velodyne data

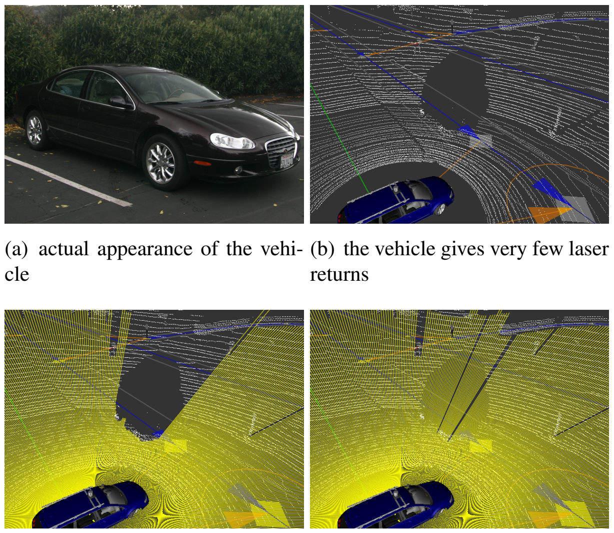
(c) after classification

(d) generated 2D virtual scan

Figure 5: In (c) Velodyne data is colored by type: orange — ground, yellow — low obstacle, red — medium obstacle, green — high obstacle. In (d) yellow lines denote the 2D virtual scan. Note the truck crossing the intersection, the cars parked on a side of the road and the white van parked on a driveway. On the virtual scan, all of these vehicles are clearly marked as obstacles, but ground, curbs and tree tops are ignored. Images are from (Petrovskaya and Thrun, 2009).

readjust origin point of rays, project from 3D to 2D, and reject outliers. It is especially important to filter out ground readings. Fig. 5 shows an example of a virtual sensor, where a 3D scan is converted to a 2D scan. With multi-layer and 3D laser sensors, ground filtering can be done based on the fact that impacts are aligned almost vertically on objects. Hence, ground filtering can be achieved by comparing relative angles between rays (see Fig. 4 for an example). When a vision sensor is available in addition to laser, the vision sensor can be used to detect the road surface, thus allowing to filter range data. Since lasers do not see black obstacles very well, some approaches also fill-in data for black objects (Fig. 6). If no readings are obtained along a range of vertical angles in a specific direction, the space must be occupied by a black obstacle. Otherwise the rays would have hit some obstacle or the ground.

For vision sensors, it is common to convert the image to greyscale, apply a Gaussian blur, and reduce the image resolution. When using stereo-vision, the pixels of the two images are matched. Then, a distance measurement is obtained through triangulation. With this technique, a pair of images can be transformed into a set of tridimensional points, represented in the *sensor coordinate system*, or in another space like the *disparity space* for some specific configuration of the sensor (Labayrade et al., 2002). Using successive images in an optical flow approach is another way to retrieve tridimensional data from a vision sensor.


(a) actual appearance of the vehicle (b) the vehicle gives very few laser returns

(c) virtual scan with black object detection (d) virtual scan without black object detection

Figure 6: Detecting black vehicles in 3D range scans. White points represent raw Velodyne data. Yellow lines represent the generated virtual scans. Images are from (Petrovskaya and Thrun, 2009).

2.2.4 Dynamics Models

Dynamics model describes motion of an object in probabilistic terms. Given a prior pose and velocity, the model defines a probability distribution over resulting poses at the next time step. In DATMO, it is common to assume a *constant velocity model*, in which the velocity of each tracked target stays constant for the duration of each time interval from $t-1$ to t. At each time step t, the velocity instantaneously evolves via addition of random noise based on maximum allowed acceleration a_{max} and the time delay Δt from the previous time step $t-1$. More specifically, Δv is either sampled from a normal distribution $\mathcal{N}(0, a_{max}\Delta t)$ or uniformly from $[-a_{max}\Delta t,\ a_{max}\Delta t]$.

Brownian Motion Model The simplest model of motion is *Brownian motion*. This model is often used for pedestrians because people can rapidly change velocity and direction of motion. In this model, the pose evolves via addition of zero-mean Gaussian noise. The variance of the noise grows with Δt.

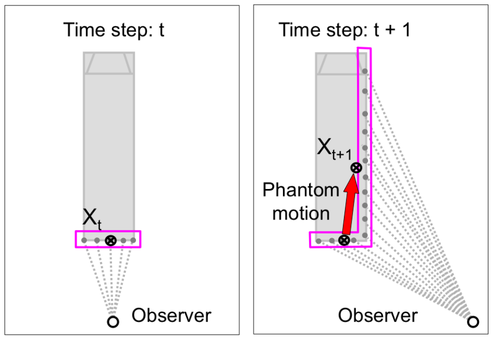
Linear Motion Model Vehicles have more predictable motion than pedestrians due to higher inertia and non-holonomic constraints. Since the exact dynamics of tracked vehicles are unknown, it is common to use the *linear motion* model. In this case, the motion consists of perturbing orientation by $\Delta\theta_1$, then moving forward according to the current velocity by $v_t\Delta t$, and making a final adjustment to orientation by $\Delta\theta_2$. Given a maximum allowed turning rate $d\theta_{max}$, $\Delta\theta_1$ and $\Delta\theta_2$ are sampled from a normal distribution $\mathcal{N}(0, d\theta_{max}\Delta t)$ or uniformly from $[-d\theta_{max}\Delta t,\ d\theta_{max}\Delta t]$.

Bicycle Motion Model A more refined motion model for vehicles is the *bicycle model*. This model uses angular velocity ω_t in addition to linear velocity v_t. As before, constant velocity model is assumed for both angular and linear velocities. The object moves along an arc, which is determined based on the linear and angular velocities. The equations of motion are more complex in this case. See (Thrun et al., 2005, Sec. 5.4) for details.

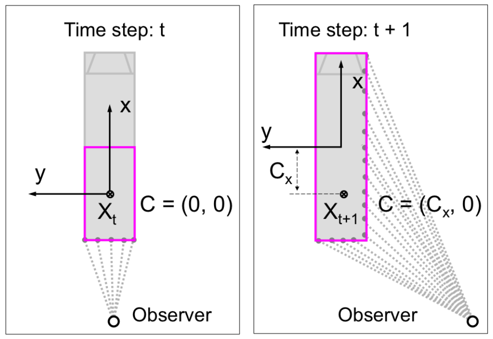
2.3 Scene Models

2.3.1 Cluster Based Models

One of the simplest and most common object models are cluster based representations. In this case, tracked objects are represented as clusters of observed data points or features. Target's pose X_t represents the position of the geometric mean of the cluster in the world coordinate frame. The main drawback of this representation is that it is viewpoint dependent because the geometric mean does not account for unobserved parts of the object. Once previously unobserved parts of the object come into view, the geometric mean shifts with respect to the object. This shift is seen as motion of the target because X_t moves with the geometric mean. This leads to incorrect estimates of the object's pose and velocity. For example, a robot can perceive a non-zero velocity for a stationary vehicle, simply because the robot moves around it and observes it from a different vantage point as Fig. 7 illustrates. This problem can be addressed using geometric object models, which are described below.



(a) cluster model

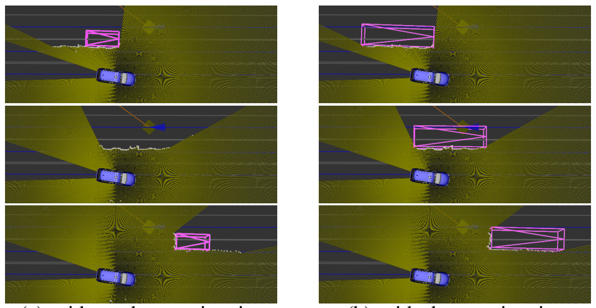


(b) geometric model

Figure 7: The effect of vantage point shift on estimated motion of the object. As we move to observe a different side of a stationary bus, the geometric mean of observed points shifts with respect to the bus. (a) With cluster models, X_t is tied to the geometric mean of the points. Hence, a change in vantage point leads to phantom motion (red arrow) of the object. (b) With geometric models, anchor point coordinates $C = (C_x, C_y)$ compensate for geometric mean shift, so that X_t remains stationary in world coordinates. The figure is from Petrovskaya (2011).

2.3.2 Geometric Models

As an alternative to clusters, objects can be represented by geometric shapes. Rectangular boxes are most common, although circles and ovals have also been used. Since one size and shape does not fit all targets well, some methods use several pre-defined shapes: one per class of objects. For example, pedestrians could be modeled as circles, bicycles as ovals, cars as rectangles, and buses as long rectangles. In this case object class variable is added to the state for each target.



(a) without shape estimation (b) with shape estimation

Figure 8: Shape inference on the example of a passing bus. Without shape estimation (a) the tracking results are poor because the geometric model does not fit the data well. Not only is the velocity estimated incorrectly, but the track is lost entirely when the bus is passing. With shape estimation (b) the bus is tracked successfully and the velocity is properly estimated. Images are from (Petrovskaya and Thrun, 2009).

To obtain an even more accurate representation of objects, some approaches parameterize the shape and infer the shape parameters. For example, objects could be modeled as rectangles, for which width and length are added to the state variables. Fig. 8 illustrates the impact of shape inference on a real example.

When unobserved parts of an object come into view, with a geometric model it is still possible to experience geometric mean shifts just as for the cluster based representations discussed above. To overcome this problem, let X_t be the pose of an *anchor point*, which is fixed somewhere on the object. Although the anchor point remains fixed on the object, the robot's belief of the anchor point position with respect to the object's center can change over time (Fig. 7(b)). Initially, the robot sets the anchor point to be the center of what it believes to be the object's shape and thus anchor point coordinates in the object's *local* coordinate system are $C = (0, 0)$. Assume that the object's local coordinate system is tied to its center with the x-axis pointing directly forward. As the robot revises its knowledge of the object's shape, the local coordinates of the anchor point need to be revised accordingly to $C = (C_x, C_y)$. Thus, C_x and C_y are added to the target's state variables.

2.3.3 Grid Based Models

Occupancy grids have been utilized to represent the scene for some time (Moravec (1988)). Occupancy grids divide the scene into a grid of cells. The cells may be uniform or varying in size. Typically each cell has a value $P(O_i)$, representing the probability that something is occupying cell i. $P(O_i) = 0$ indicates a cell that is certainly empty, $P(O_i) = 1$ indicates that it is certainly occupied, and $P(O_i) = 0.5$ is typically used to represent cells for which no information is available.

In the intelligent vehicle context, the grid is typically used to represent an area in front of the vehicle. That is to say that rather than the grid being used to represent the complete environment at a large scale, the grid is essentially attached to the vehicle, in a robot-based frame, as described in Sect. 1.2.1. This is because the grid is being used to model the scene and identify potential dangers, rather than building a global map. Work has been done using this moving grid along with local-slam to differentiate between moving and static obstacles.

In defining the grid, tradeoffs are made between accuracy and performance in specifying the grid size and cell size. Specifically, smaller cells allows for a more accurate representation of the scene. Similarly, a larger grid allows representation of more of the environment. However, additional cells also require additional computing resources, in terms of both memory for storing the grid, and computing power for updating the grids. The computing resources available and application-specific details (sensor range, sensor accuracy, accuracy requirements, etc.) should be used to define the grid.

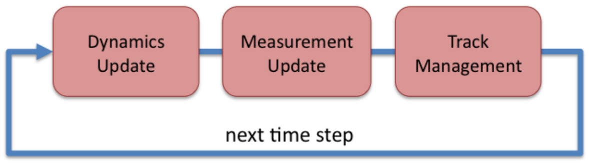
In the grid-based approaches, concepts such as *objects* or *tracks* do not exist; they are replaced by other properties such as occupancy or elevation, which are directly estimated for each cell of the grid using both sensor observations and some prior knowledge. It might seem strange to have no object representations when objects obviously exist in real life environments. However, an object-based representation is not required for all applications. Where object-based representations are not pertinent, we argue that it is more useful to work with a more descriptive, richer sensory representation rather than constructing object-based representations with their complications in data association. For example, to calculate the risk of collision for a mobile robot, the only properties required are the probability distribution on occupancy and velocities for each cell in the grid. Variables such as the number of objects are inconsequential in this respect.

Occupancy grids are especially useful to fuse information from several sensors. In standard methods for sensor fusion in tracking applications, the problem of track-to-track association arises where each sensor contains its own local information. Under the standard tracking framework with multiple sensors, the problem of data association will be further complicated: as well as the data association between two consecutive time instances from the same sensor, the association of tracks (or targets) between the different sensors must be taken into account as well.

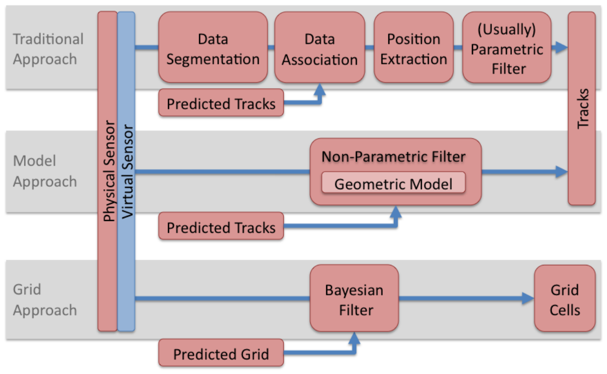
In contrast, the grid-based approaches will not encounter such a problem. A grid-based representation provides a conducive framework for performing sensor fusion Moravec (1988). Different sensor models can be specified to match the different characteristics of the different sensors, facilitating efficient fusion in the grids. The absence of an object-based representation allows easier fusing of low-level descriptive sensory information onto the grids without requiring data association.

Grid mapping from sensor data Occupancy grids are generally computed from the data provided by range finders. The classical approach is based on the *independent beam model*, taking into account both *positive* and *negative* information, see Sect. 2.2.1.

Grid mapping is somewhat less common with vision sensors. With a single camera, integration over time is necessary to retrieve distance measurements, leading to approaches like visual-SLAM. Stereo-vision can provide tri-dimensional data, so it is current practice to use it as a distance sensor, considering a metric point cloud Braillon et al. (2006). Stereo-specific methods also exist. In Matthies and Elfes (1988) or Murray and Little (2000), the authors consider the first detected object for each column and suppose that it occludes the field ahead. This is obtained by finding the highest disparity value for each column of the image. The result is a ternary grid (free/occupied/unknown) and the approach is sensitive to outliers in the disparity map. An improvement has been proposed in Badino et al. (2007), where the authors pro-



(a)



(b)

Figure 9: (a) Typical DATMO pipeline. (b) Three alternative approaches to the measurement update step in the DATMO pipeline. Each approach starts with sensor data on the left and produces tracks of moving objects or grids on the right. Predicted tracks and predicted grid are outputs of the dynamics update. Red boxes are mandatory, the blue box is optional.

pose to build such a grid in the u-disparity plan. Occupancy is then robustly estimated using dynamic programming. The result is a ternary grid, specifically designed for free field estimation. An improvement is proposed in Perrollaz et al. (2010), where the authors propose a visibility based approach to estimate the occupancy probability of each cell of the grid, in the u-disparity plane. This is done by estimating for each cell the probabilities of "being visible", of "containing an object", and of "containing the road surface".

3 Inference: Filtering and Tracking

In a Bayesian approach, the evolving state of a dynamic system is tracked using a *Bayesian filter*, which computes the current belief bel_t recursively from a prior belief bel_{t-1} using the Bayesian recursion equation (1). The filter consists of a *dynamics update*, which captures the evolution of the state between time steps, and a *measurement update*, which incorporates the most recent sensor measurements (Gordon, 1993; Diard et al., 2003). In DATMO, separate Bayesian filters are used to track the state of each moving object (or cell). Updates of the Bayesian filter are followed by a track management phase, during which tracks are created and deleted and dependencies between targets are enforced. The resulting pipeline for DATMO is summarized in Fig. 9(a).

There are two types of Bayesian filters: parametric and non-parametric (for an overview see Arulampalam et al. (2002)). *Parametric filters* include variants of *Kalman filters* (KF) and represent the belief by a parametric function, a *Gaussian*. Since the actual belief is usually non-Gaussian, approximations have to be made as in the extended Kalman filter (EKF) or the unscented Kalman filter (UKF). *Non-parametric filters* include *particle filters* (PF) and *histogram filters* (HF). They represent the belief by a set of points, which are positioned randomly in case of the PF and deterministically in HF. In PF these points are called *particles*, in HF they are *grid cells*.

Parametric methods have the advantage of being more efficient as their computational cost is polynomial in the dimensionality of the state. However, parametric methods are unable to represent complex beliefs, which often arise due to ambiguities in the data. They also often perform poorly if a sufficiently good estimate of the state is not available *a priori*. In contrast, non-parametric methods have the advantage of being able to represent an arbitrary belief even if it has multiple high probability regions (called *modes*). However, their computational cost is exponential in the dimensionality of the state and the base of the exponent tends to be large. This property is often referred to as the *curse of dimensionality* (MacKay, 1998).

Several types of DATMO approaches have evolved to build around these advantages and disadvantages. To help organize the discussion in this chapter, these approaches can be roughly divided into three main branches (Fig. 9(b)): the traditional approach, the model based approach, and the grid based approach approach. The first two branches represent two schools of thought for producing object level tracks: (i) build a pseudo-sensor of target positions by augmenting the physical sensor with a set of algorithms and then use a simple (usually parametric) filter (top branch), or (ii) use the raw sensor with a physical sensor model and employ a more advanced non-parametric filter (middle branch). In (i) most of the problem is solved while producing pseudo-measurements of target positions, whereas in (ii) most of the problem is taken care of by the inference method. Both of these approaches are discussed below in Sect. 3.2 and Sect. 3.3.

The third approach (bottom branch in Fig. 9(b)) has a different goal than the first two. Instead of object level tracks of targets, it produces tracks of grid cells — a lower level representation of moving obstacles, which can be used directly for low level trajectory planning. This approach is described in Sect. 3.4. To obtain object level tracks, the grid based approach can be followed by a process similar to the traditional branch, but replacing the physical sensor with the grid representation, which is built by the grid based approach. This method is described in Sect. 3.4.3.

3.1 Track Management and Dynamics Update

3.1.1 Imposing Track Constraints

Although during inference inter-track dependencies are ignored as was already mentioned in Sect. 1.2.2, these dependencies clearly exist. For example, no two vehicles can occupy the same space at the same time. These dependencies are enforced during the track management phase by imposing constraints. The constraints may include assumptions that targets do not overlap, are surrounded by some amount of free space, and are located in *regions of interest* (ROI, e.g. on or near roads).

Tracks violating constraints can be deleted right away, can have their existence score decreased (as discussed below), or can be merged with other tracks in the case of overlapping targets.

3.1.2 Track Existence

Creation and deletion of tracks relies on *track existence* probability, which can be derived using Bayes rule. This probability is usually kept in log form as an *existence score*, which represents the *log-likelihood ratio* (LLR) first proposed by Sittler (1964) and summarized by Blackman et al. (2004). This approach is an application of the classical *sequential probability ratio test* (SPRT).

The existence score takes into account whether or not the target has been observed, how well it matched the observations, how well it fits the dynamics model, as well as expected probability of detection and expected density of false alarms. It can also include terms for *signal-to-noise ratio* (SNR). In short, if a target is observed and matches the data well then the score is increased, otherwise it is decreased.

3.1.3 Track Creation

After a measurement update, the sensor data not well explained by existing tracks are examined to initialize new tracks. This process can take several frames of data into account (e.g. from time steps $t-2$, $t-1$, and t) by running a Bayesian filter over these frames of data for a prospective target. For each frame of data, the existence score is computed for the prospective target. If the score is sufficiently high in all frames, the prospective target may be added to the list of tracked targets.

Note that detection of new targets is a more challenging problem than tracking of existing targets. Since the initial pose uncertainty for each new target is very large, large areas of space need to be examined for possible existence of new targets. In contrast, during tracking of existing targets, a prior pose estimate is already available from the previous time step and hence the pose uncertainty of each target is lower.

A number of techniques can be used to make detection of new targets more efficient. For example, it is common to restrict areas in which new targets can enter the scene. These areas may be defined by boundaries of the observed space (i.e. image boundaries for cameras and point cloud boundaries for 3D sensors) and by features of the environment (e.g. entrance lanes on freeways or intersections). Another approach is to perform some fast preprocessing of the data to determine locations, in which new targets may have appeared. For example, sufficiently large change has to occur in an area if it contains a moving object.

3.1.4 Track Deletion

Like for track creation, existence score can be used to determine if a particular track should be discontinued. The existence score is computed for each target at each time step. Once the score falls below a certain threshold, the track is deleted.

3.1.5 Dynamics Update

Once track management phase completes, all surviving tracks are propagated to the next time step. The propagation is performed using the dynamics update of the Bayesian filter used to track the targets. The dynamics update relies on probabilistic laws of target dynamics, which we discussed in Sect. 2.2.4. In effect, the dynamics update computes the prediction distribution for each target $\overline{bel}_t := p(S_t|Z_1,\ldots,Z_{t-1})$, which is given by the integral in the Bayesian recursion equation (1).

3.2 Traditional DATMO

The traditional pipeline of DATMO corresponds to the top branch in Fig. 9(b). It usually relies on variants of Kalman filters, although some recent approaches have used particle filters. The distinguishing characteristic of the traditional approach is that most of the work is done prior to application of the filter. The data are segmented into meaningful pieces using clustering or pattern recognition methods as described in Sect. 3.2.1 and Sect. 4.1, respectively. The pieces of data are associated with targets using data association methods described below in Sect. 3.2.2. This stage can be challenging because of the association ambiguities that often arise. Finally, for each target a pseudo-measurement of position is produced by taking geometric mean of the data assigned to each target. This position estimate is then used in a Kalman filter variant (or a particle filter) to update the belief of each target's state.

A broad spectrum of DATMO approaches fall into the traditional approach category. While early work has used simple clustering techniques to segment data (Sect. 3.2.1), recent work within the traditional branch partitions data based on advanced pattern recognition techniques to improve detection of targets (Sect. 4.1).

3.2.1 Data Segmentation

Among the simplest methods for data segmentation are the *clustering methods*. These methods can be applied directly to range data. The data points are segmented into clusters based on range discontinuities. These clusters then need to be assigned to targets during the data association phase, however multiple clusters may need to be associated with the same target.

A slightly more complex approach is to look for specific features in the data. Straight lines and letter L's are common features for range data. Harris corners and edges are common for vision data. Again these features need to be associated to targets with possibly multiple features assigned to each target.

More sophisticated techniques for data segmentation rely on pattern recognition techniques, which are discussed in Sect. 4.1.

3.2.2 Data Association

Once the data have been segmented into pieces, these pieces need to be associated to targets. Below is a summary of data association methods. See Cox (1993) for a detailed review.

One of the simplest methods for data association is the *nearest neighbor* (*NN*) method, which assigns each piece of data to the closest predicted target. This method is widely used when update rates are high enough to make this assignment unambiguous. A more sophisticated method is the *global nearest neighbor* (*GNN*), which ensures that each piece of data is assigned to one target. It is useful for recognition based methods, which ensure that each piece of data represents the entire tracked object (e.g. pedestrian or vehicle) and hence a one-to-one correspondence between targets and pieces of data is appropriate.

More advanced methods have been developed for situations where ambiguities are plentiful. The two most common approaches are the *multiple hypothesis tracking* (*MHT*) algorithm and the *joint probabilistic data association filter* (*JPDAF*).

MHT Algorithm Originally developed by Reid (1979), this algorithm maintains multiple association hypotheses between targets and measurements. The MHT framework can handle situations where the measurements arise from a varying number of targets or from background clutter. For example in Fig. 10, there is a single hypothesis H^1_{t-1} at time $t-1$ containing targets T^1_{t-1} and T^2_{t-1}. At time t,

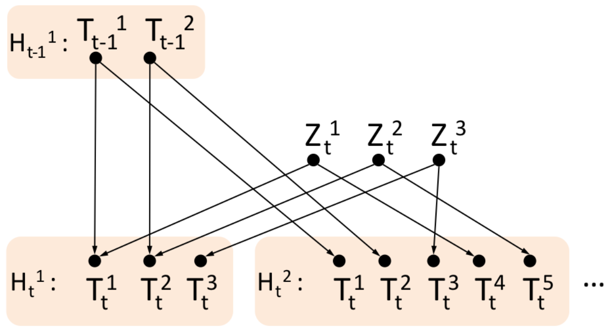

H_{t-1}^1: T_{t-1}^1 T_{t-1}^2
Z_t^1 Z_t^2 Z_t^3
H_t^1: T_t^1 T_t^2 T_t^3
H_t^2: T_t^1 T_t^2 T_t^3 T_t^4 T_t^5
...

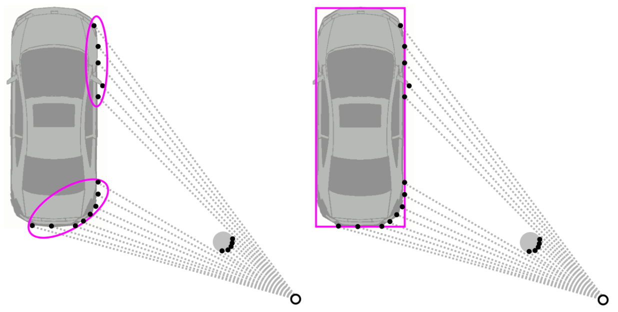
Figure 10: Example of a data association problem with two targets T_{t-1}^1, T_{t-1}^2 from time $t-1$ and three new measurements Z_t^1, Z_t^2, Z_t^3. A single hypothesis H_{t-1}^1 from the prior step splits into multiple hypotheses at time t, of which only H_t^1 and H_t^2 are shown. Arrows show the associations made.

the new observations are segmented into three pieces denoted by Z_t^1, Z_t^2, and Z_t^3. The new possible hypotheses are formed from the prior step hypothesis by associating the new measurements to the existing targets or by initializing new targets from each new measurement. Thus, these associations give rise to a number of hypotheses at time t, of which only two are shown in the figure.

For each of the obtained hypotheses, a hypothesis score is computed by summing the track existence scores (from Sect. 3.1.2) of all targets within it. The hypothesis probability can then be computed from the hypothesis score.

In practical situations, several issues arise from the application of this method. The most serious is the combinatorial increase in the number of generated tracks and hypotheses. Several techniques such as tracks clustering or pruning can be used to avoid this increase. For further information on MHT and its variants see Blackman et al. (2004).

JPDAF Algorithm The algorithm was proposed by Fortmann et al. (1983) based on the *probabilistic data association* concept introduced by Bar-Shalom and Jaffer (1972). Unlike the MHT algorithm, JPDAF does not suffer from the combinatorial explosion. It is similar to GNN in that a single association hypothesis is maintained. However, unlike GNN, JPDAF does not make a hard assignment of measurements to targets. Instead, it makes a soft assignment by considering the probability of each



(a) without geometric model (b) with geometric model

Figure 11: Scans from objects are often split up into separate clusters due to occlusion (a). Geometric model helps interpret the data properly (b). Purple shapes group together points that have been associated together. In (b) the purple rectangle also denotes the geometric model. Gray areas are objects. Gray dotted lines represent laser rays. Black dots denote laser data points. The figure is from Petrovskaya (2011).

measurement being assigned to each target.

More specifically, suppose we have K targets and M_t data segments $Z_t^1, \ldots, Z_t^{M_t}$. Let the probability of measurement m being caused by target k be denoted by β_{mk}. Then for target k, the measurement update is carried out considering all possible assignments of data segments

$$p\left(Z_t | S_t^k\right) = \eta \sum_{m=1}^{M_t} \beta_{mk}\, p\left(Z_t^m | S_t^k\right), \tag{2}$$

where η is a normalization constant. For details on how to approximate the probabilities β_{mk} see Schulz et al. (2003b).

A number of extensions and variants of JPDAF have been developed (Cox, 1993). Most methods utilizing JPDAF rely on parametric belief representations (i.e. Kalman filters). Such methods are widely used in DATMO approaches. Non-parametric variants of JPDAF have also been developed (Schulz et al., 2003b; Vermaak et al., 2005), although these methods have yet to find their way into DATMO for autonomous driving.

3.3 Model Based DATMO

The model based approach corresponds to the middle branch in Fig. 9(b). It works by directly modeling the physical sensor and the moving objects using a physical sensor model (Sect. 2.2.1) and geometric models of objects (Sect. 2.3.2). Most of the work here is done by the filter itself. Separate data segmentation and association steps are not required because the geometric object model helps associate data points to targets (see Fig. 11). The main challenge is to make the filter efficient enough to meet the high demands of autonomous driving.

3.3.1 Rao-Blackwellization

Inference efficiency can be improved with a hybrid representation of the belief using a *Rao-Blackwellized particle filter* (*RBPF*). The concept of Rao-Blackwellization dates back to Rao (1945) and Blackwell (1947) and has been used with great success in SLAM (Murphy and Russell, 2001; Montemerlo, 2003). In RBPF, the belief over some parameters of the state is represented by particles with each particle maintaining a Gaussian belief over the remaining parameters. This technique effectively reduces the dimensionality of the state from the perspective of the particle filter.

Let Ω denote the vector of object shape parameters (e.g. , $\Omega = (W, L, C_x, C_y)$). Then the full belief is $bel_t := p(X_{1:t}, v_{1:t}, \Omega | Z_{1:t})$, where $1 : t$ is a shorthand for $1, \ldots, t$. For Rao-Blackwellization, the belief is split into two conditional factors

$$bel_t = p(X_{1:t}, v_{1:t} | Z_{1:t}) \;\; p(\Omega | X_{1:t}, v_{1:t}, Z_{1:t}). \tag{3}$$

The first factor represents the belief about the object motion, whereas the second factor represents the belief about the object shape, conditioned on its motion. Let A_t denote the first factor and B_t denote the second factor. The motion belief is represented by particles and the shape belief is represented in Gaussian form. The dynamics update of RBPF is the same as standard particle filter (PF): the particles are resampled and propagated according to the dynamics model. Like for PF, the measurement update involves computation of the importance weights. For RBPF, the correct importance weights can be shown to be

$$w_t = \frac{p(X_{1:t}, v_{1:t} | Z_{1:t})}{p(X_{1:t}, v_{1:t} | Z_{1:t-1})} = \mathbb{E}_{B_{t-1}}\Big[\; p(Z_t | \Omega, X_t) \;\Big]. \tag{4}$$

In words, the importance weights are the expected value of the measurement likelihood with respect to the object shape prior. Using Gaussian approximations of B_{t-1} and the measurement model this expectation can be computed in closed form. See Petrovskaya (2011) for derivations.

3.3.2 Scaling Series Algorithm

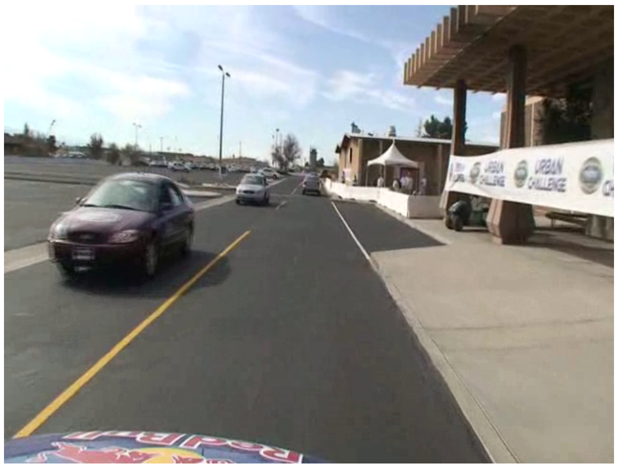
Efficiency can also be improved using an iterative annealing method, such as a *Scaling Series particle filter* (*SSPF*). SSPF gradually refines the belief from very coarse to fine resolution and simultaneously adjusts the annealing temperature. At each iteration it uses a divide-and-conquer strategy and thus it can be exponentially faster than the basic particle filter. Even though in principle SSPF can not escape the curse of dimensionality, it effectively reduces the base of the exponent (e.g. from 30 down to 6). Hence, the curse's effect is greatly diminished allowing the approach to outperform PF by several orders of magnitude. Detailed description of SSPF can be found in Petrovskaya and Khatib (2011).

An example combining the use of SSPF and RBPF techniques to solve the DATMO problem can be found in (Petrovskaya and Thrun, 2009), where the resulting approach tracks vehicles in real time (see Fig. 12). Since the model based approach leads to accurate estimation of motion, it is possible to build full 3D models of passing vehicles (Fig. 13) by collecting all scan points assigned to each target (Petrovskaya, 2011). The 3D models can in turn be used for even more accurate motion estimation, which is particularly useful for slow-moving vehicles near intersections.

3.3.3 MCMC Approach

The DATMO problem can be posed as a batch problem over a sliding window of a fixed number of frames. In this form, it can be solved using the *Markov Chain Monte Carlo* (*MCMC*) approach. See Andrieu et al. (2003) for an introduction to MCMC and related concepts.

MCMC methods produce samples from a desired probability distribution by constructing a *Markov chain*, for which the *equilibrium distribution* is equal to the desired distribution. When these methods "walk" the chain for a sufficiently large number of steps, the resulting state of the chain can be used as a sample from the desired distribution. While it is usually easy to construct a Markov chain with the desired distribution as its equilibrium, determin-


(a) actual scene

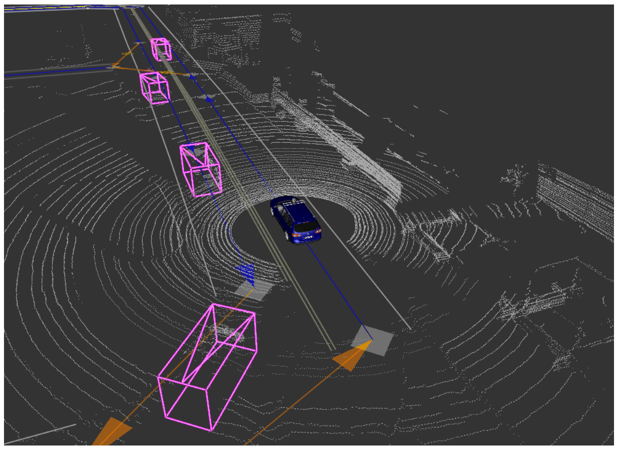

(b) Velodyne data and tracking results

Figure 12: Tracking results on course A at the 2007 Urban Grand Challenge. The purple boxes in (b) denote the tracked vehicles. Images are from Petrovskaya and Thrun (2009) and Petrovskaya (2011).

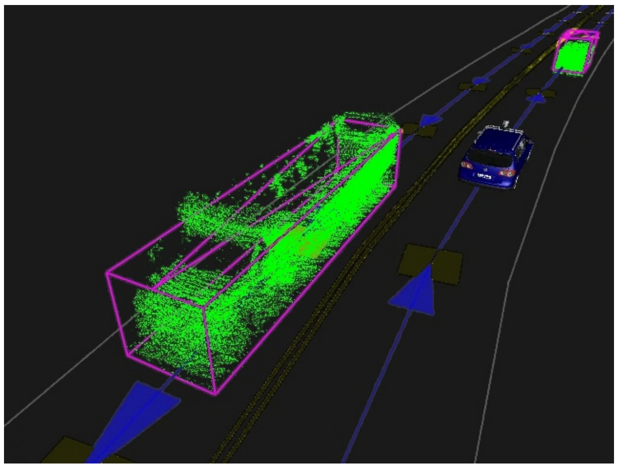


Figure 13: 3D shape inference for vehicle tracking. The green points show the accumulated Velodyne points assigned to each vehicle over multiple frames. Tracking with anchor point coordinates allows us to align the points from different frames on per-vehicle basis. Blue car shows the position of the ego-vehicle. Image is from Petrovskaya (2011).

ing the required number of steps (known as the *mixing time*) can be difficult.

The *Metropolis-Hastings* version of MCMC was proposed by Metropolis et al. (1953) and Hastings (1970). It uses a proposal distribution and accepts or rejects the next state based on a ratio test, which takes into account both the desired distribution and the proposal distribution. Although this algorithm will eventually converge for any proposal distribution, a more informed proposal distribution greatly improves efficiency of the approach. For this reason, Vu and Aycard (2009) used a *data driven* MCMC technique and generated object hypotheses with a detection module, which identifies moving parts of dynamic objects.

Further, their approach uses discrete models of objects: rectangles of several fixed sizes for cars, busses, and bikes, and points for pedestrians. Each class of objects also has its own dynamics model, which is integrated into the framework using an *interacting multiple model* (*IMM*) filter (Vu, 2009).

3.4 Grid Based DATMO

The grid based DATMO approach corresponds to the bottom branch in Fig. 9(b). The Bayesian filtering paradigm can be applied for estimation and filtering in the occupancy grid framework. This provides multiple advantages compared to static grids estimated independently for each frame of sensor data:

- it allows estimation of the velocity of each cell of the grid, hence modeling the dynamic environment;

- since it takes into consideration successive measurements over time, it allows retention of information about occupancy for regions of the scene that are occluded;
- through filtering, it can remove errors that are present on a unique frame of measurements.

Moreover, in the classical methodology presented in the top branch of Fig. 9(b), the problems of data association and state estimation are highly coupled, and an error in either component leads to erroneous outputs. The grid filtering methodology makes it possible to decompose this highly coupled relationship by avoiding the data association problem, in the sense that the data association is handled at a higher level of abstraction. A few methods have been proposed in the literature in order to adapt Bayesian filtering to the occupancy grids, the first and more popular approach is the *Bayesian Occupancy Filter* (*BOF*) which we will present briefly below.

3.4.1 The Bayesian Occupancy Filter

The *Bayesian Occupancy Filter* (*BOF*) addresses the dynamics of the environment using a two-step mechanism, derived from the Bayesian filter. This mechanism estimates, at each time step, the state of the occupancy grid by combining a prediction step (history) and an estimation step (incorporating new measurements). The consideration of sensor observation history enables robust estimation in changing environments (i.e. it allows processing of temporary objects, occlusions and detection problems).

The state space of the BOF is a 2-dimensional occupancy grid. Each cell of the grid is associated with a probability of occupancy and a probability distribution on the velocity of the occupancy associated with the cell. Contrary to the initial formulation presented in Coué et al. (2006), this formulation does not allow for overlapping objects. On the other hand, it allows for inferring velocity distributions and reduces the computational complexity.

Define the following variables:

- C is an index that identifies each 2D cell of the grid.
- A is an index that identifies each possible antecedent of the cell c over all the cells in the 2D grid.
- $Z_t \in \mathcal{Z}$ where Z_t is the random variable of the sensor measurement relative to the cell c.
- $V \in \mathcal{V} = \{v_1, \ldots, v_n\}$ where V is the random variable of the velocities for the cell c and its possible values are discretized into n cases.
- $O, O^{-1} \in \mathcal{O} \equiv \{occ, emp\}$ where O represents the random variable of the state of c being either "occupied" or "empty". O^{-1} represents the random variable of the state of an antecedent cell of c through the possible motion through c. For a given velocity $v_k = (v_x, v_y)$ and a given time step δt, it is possible to define an antecedent for $c = (x, y)$ as $c^{-k} = (x - v_x\delta t, y - v_y\delta t)$.

The decomposition of the joint distribution of the relevant variables according to Bayes' rule and dependency assumptions can be expressed as:

$$P(C, A, Z, O, O^{-1}, V) \\ = P(A)P(V|A)P(C|V, A)P(O^{-1}|A)P(O|O^{-1})P(Z|O, V, C)$$

with this decomposition, each component can be associated with a semantic and a parametric form. Particularly, $P(O|O^{-1})$ is the conditional distribution over the occupancy of the current cell, which depends on the occupancy state of the previous cell. It is defined as a transition matrix:

$$T = \begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix},$$

which allows the system to use a constant velocity hypothesis as an approximation; in this matrix, ϵ is a parameter representing the probability that the object in c does not follow the null acceleration model, i.e. ϵ models the prediction error.

The aim of filtering in the BOF grid is to estimate the occupancy and grid velocity distributions for each cell of the grid, $P(O, V|Z, C)$. The two stages of prediction and estimation are performed for each iteration. In the context of the BOF, prediction propagates cell occupancy probabilities for each velocity and cell in the BOF ($P(O, V|C)$). During estimation, $P(O, V|C)$ is updated by taking into account its observation $P(Z|O, V, C)$ to obtain its final Bayesian filter estimation $P(O, V|Z, C)$. The result from the Bayesian filter estimation is then used for prediction in the next iteration.

3.4.2 BOF Implementation

When implementing the BOF, the set of possible velocities is discretized. One way of implementing the computation of the probability distribution is in the form of histograms. The following equations are based on the discrete case. Therefore, the global filtering equation can be obtained by:

$$P(V,O|Z,C) = \frac{\sum_{A,O^{-1}} P(C,A,Z,O,O^{-1},V)}{\sum_{A,O,O^{-1},V} P(C,A,Z,O,O^{-1},V)}, \quad (5)$$

The global filtering equation (eqn. 5) can actually be separated into three stages. The first stage computes the prediction of the probability measure for each occupancy and velocity:

$$\begin{aligned}&\alpha(occ,v_k)\\ &= \textstyle\sum_{A,O^{-1}} P(A)P(v_k|A)P(C|V,A)P(O^{-1}|A)P(occ|O^{-1}),\\ &\alpha(emp,v_k)\\ &= \textstyle\sum_{A,O^{-1}} P(A)P(v_k|A)P(C|V,A)P(O^{-1}|A)P(emp|O^{-1})\end{aligned}$$

This is performed for each cell in the grid and for each velocity: prediction for each cell is calculated by taking into account the velocity probability and occupancy probability of the set of antecedent cells.

With the prediction of the grid occupancy and its velocities, the second stage consists of multiplying by the observation sensor model, which gives the unnormalized Bayesian filter estimation on occupancy and velocity distribution:

$$\begin{aligned}\beta(occ,v_k) &= P(Z|occ,v_k)\alpha(occ,v_k),\\ \beta(emp,v_k) &= P(Z|emp,v_k)\alpha(emp,v_k).\end{aligned}$$

Similarly to the prediction stage, these equations are performed for each cell occupancy and each velocity. The marginalization over the occupancy values gives the likelihood of a certain velocity:

$$l(v_k) = \beta(occ,v_k)+\beta(emp,v_k).$$

Finally, the normalized Bayesian filter estimation on the probability of occupancy for a cell C with a velocity v_k is obtained by:

$$P(occ,v_k|Z,C) = \frac{\beta(occ,v_k)}{\sum_{v_k} l(v_k)}. \quad (6)$$

The occupancy distribution in a cell can be obtained by the marginalization over the velocities and the velocity distribution by the marginalization over the occupancy values:

$$P(O|Z,C) = \sum_V P(V,O|Z,C), \quad (7)$$

$$P(V|Z,C) = \sum_O P(V,O|Z,C). \quad (8)$$

3.4.3 Object Level Tracking

There are often times where object level representations are required. The philosophy of the BOF is to delay the problem of data association and as a result does not contain information with respect to objects. A natural approach to obtain an object level representation from the BOF grids is to introduce grid based clustering to extract object hypothesis and an object level tracker to handle the hypothesis extracted.

Approaches similar to image processing can be advantageously used for the clustering the grid, e.g. a search for connected components after thresholding. When using a dynamic grid estimation algorithm, like the Bayesian Occupancy Filter, it is possible to take advantage of the estimated velocities, in order to improve the clustering. For instance, two very close objects with different velocities would be in the same cluster without velocity estimation, while they could be separated.

After segmentation of the grid, clusters representing the different objects of the scene can be tracked over time. A classical algorithm, like the JPDAF presented in Sect. 3.1 can be used for tracks management. However, in the cluttered environment with numerous moving objects, the JPDAF suffers from the combinational explosion of hypotheses. The *Fast Clustering and Tracking Algorithm* (*FCTA*) is more adapted to the BOF framework and to real time processing, since it uses a ROI based approach to facilitate the association stage (Mekhnacha et al., 2008).

FCTA could be roughly divided into a clustering module, an ambiguous association handling module and a tracking and track management module.

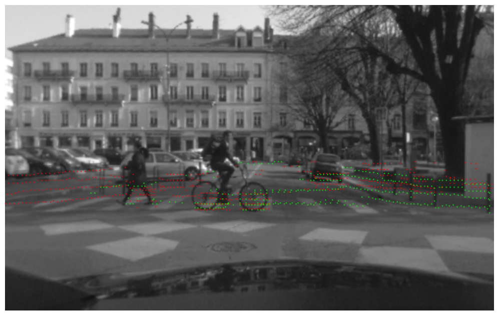
Clustering The clustering module takes the occupancy/velocity grid of the BOF as the input and extracts object level reports from it. A natural algorithm to achieve this is to connect the eight-neighbor cells according to an occupancy threshold. In addition to the occupancy values, a threshold of the Mahalanobis distance between the velocity distributions is also employed to distinguish the objects that are close to each other but with different moving velocities. In order to avoid searching for clusters in the whole grid, the predicted targets states are used as a form of feedback, by predicting regions of interest (ROI) in which the clustering process starts.

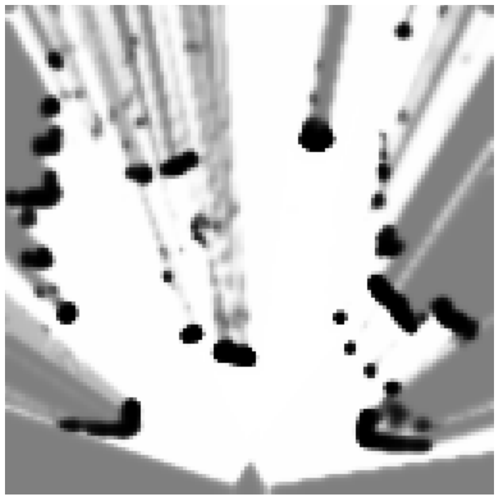
A report for the tracker is a 4-dimensional observation corresponding to the position and the velocity of an extracted cluster. The 2D position component of this vector is computed as the mass center of the region, and the 2D velocity component is the weighted mean of the estimated velocities of all cells of the cluster. These estimations come with a covariance matrix representing the uncertainty of the observation.

Re-clustering and track merging The output of this clustering module leads to three possible cases (a) no observation, where the object is not observed in the ROI, (b) ambiguity free, where one and only one cluster is extracted and is implicitly associated with the given object, and (c) ambiguity, where the extracted cluster is associated with multiple objects. Therefore, a Bayesian algorithm is used at this stage for tracks splitting and merging.

Tracks management After all the existing tracks are processed, the non-associated cells are processed to extract clusters as observations for potential new targets, using a region growing strategy from some "cluster seeds". Classically in the FCTA, the prediction and estimation of the targets are accomplished by attaching a Kalman filter with each track, similar to the traditional DATMO approach. A confidence on the existence of each track is continuously estimated, thus allowing to delete tracks with low confidence.

An example of result using the complete grid based DATMO approach is presented in Fig. 14.


(a) Lidar impacts


(b) BOF

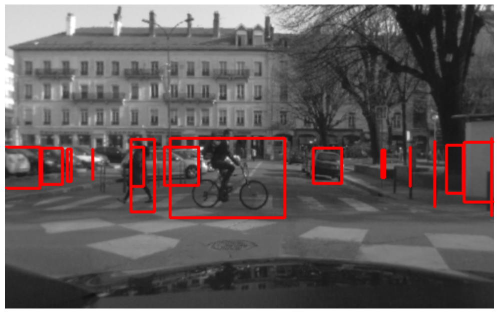

(c) FCTA

Figure 14: Example of lidar based BOF DATMO: (a) a urban scene, with dots representing the laser impacts, (b) the local occupancy grid estimated by BOF (c) tracked objects extracted by FCTA.

4 Pattern Recognition and Sensor Fusion

4.1 Pattern Recognition

4.1.1 Vision-based approaches

Object and people detection from vision data has a long history and a large body of literature exists in this area. Most of the existing approaches can be classified into one of two major kinds of methods: window scrolling approaches and parts-based approaches (see Fig. 15).

- Window scrolling approaches use a rectangular frame denoted as window that is shifted over a given image at all possible positions and scales to search for potential object occurences. For training, usually a fixed-size window is used in which features are computed and a large hand-labeled data set containing positive (people) and negative data (background) is used. In a similar way, *silhouette matching* works by matching a contour of a person at different scales and positions using an edge-to-contour metric such as the Chamfer distance Borgefors (1986). Seminal works in this area are Gavrila (2000) and Papageorgiou and Poggio (2000).

- Parts-based approaches use the fact that the most objects as well as the human body consist of connected parts. Therefore, part appeareances are learned and their geometrical relation is stored in a probabilistic way. This method produces detections when parts are represented in a geometrically consistent way in the image. Seminal works in this area are Viola et al. (2003), Mohan et al. (2001), and Mikolajczyk et al. (2004).

Recently, more mature methods that achieve high detection rates in crowded scenes have been presented (see Felzenszwalb et al. (2008); Tuzel et al. (2007); Dalal and Triggs (2005a); Leibe et al. (2005a); Wu and Nevatia (2005)). Two people detection data sets from a moving car (Enzweiler and Gavrila, 2009; Dollar et al., 2009) have been published in order to assess the performance of several detection algorithms. The conceptually simple Histogram-of-Oriented-Gradient detector by Dalal and Triggs (2005a) performs very well, especially for frontal views and intermediate sizes. Haar-based detectors such as the one by Viola et al. (2003) are more suited in detecting smaller sizes. If pedestrians are represented at bigger scales and if the aim is to also extract the articulation pose then component-based methods are expected to perform better (see Leibe et al. (2005a); Felzenszwalb et al. (2008)). We note that many of these algorithms have been designed for people detection, but they also work for any other object category, for example cars.

The implicit shape model (ISM) algorithm by Leibe et al. (2005a) is a generative detection method in which the detector is based on local features that independently cast votes for a common object center. The input for training is a set of images of people in which foreground and background is annotated via a binary mask. The first step of training is to collect all descriptors (SIFT, shape context, etc.) from interest points (Hessian-Laplace, Harris-Laplace etc). Then, a clustering algorithm is employed to reduce and generalize the numbers of descriptors present in the image set. The authors suggest using agglomerative clustering with average linkage, run with a fixed Euclidean distance θ. The resulting clustered centroids constitute the elements of the object *codebook*. Each element of the codebook is then matched to the training set images by using the same distance θ. The matched descriptors are used to note the relative position, with respect to the object center, of the associated interest points as a tag of the codebook element. To summarize: each element of the codebook is a generalized representation of an object patch and it is associated to a set of vectors indicating the positional displacement with respect to the object center.

During detection, a new image is acquired and interest points and associated descriptors are computed. Descriptors are matched with the codebook by using the distance θ. Matches are used to cast votes for object centers from the positions of the associated interest points. Votes are collected in a 3D voting space, defined by the x, y image coordinates and scale s. This procedures defines a generic Hough transform (GHT), in which the modes of the votes distribution define object detection hypotheses. A clever form of mode estimation, sensitive to scale errors amplification, is represented by a modification of the standard mean-shift technique (see Comaniciu and Meer (2002)), in which the kernel size is inflated anysotropically with respect to the scale (balloon kernel). In order to refine results in case of overlapping detection, an max-

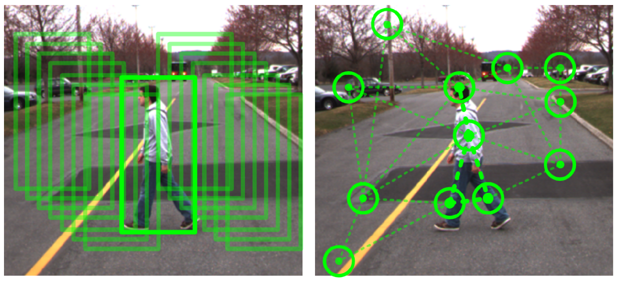


Figure 15: The main approach to object detection in computer vision. Window scrolling (**left**) and by-parts reasoning (**right**). The first classifies the image enclosed by the rectangle that has to be swept over the image at each position and at different scales. The latter classifies an object by reasoning on the geometrical disposition of object parts.

imum descriptor length (MDL) cost is used that takes into account that a feature is not sharable and it must belong to an object or to another. A quadratic boolean programming problem is formulated in order to solve this best assignment. Furthermore, this technique can be used to generate objects segmentations by associating bitmap patches to succesfully resolved feature assignments. One major disadvantage of ISM is that it relies on standard image features that have not been designed specifically for object detection. An solution to this drawbacks is provided by Felzenszwalb et al. (2008) that combines the advantages of ISM and HOG. It uses HOG-based classifiers to detect body parts and it assembles them in a probabilistic shape model.

A more classic window-based approach is used by the HOG (histogram of oriented gradients) detector of Dalal and Triggs (2005a). This detector is based on discriminative classification, thus negative and positive image training sets are needed. Each training sample is specified as a rectangle enclosing the object. This rectangle (or window) has fixed size and it is used for computing features at fixed locations. Precisely, the window is tessellated in evenly overlapping square cells. An histogram of gradients is collected from the content of each cell. Cells are organized 2×2 blocks for computing a histogram normalization that takes in account larger areas to encode robust illumination change. All the histograms of a window are concatenated in a single histogram, called HOG. All the HOG negative and positive features computed in training data are used for Support Vector Machine (SVM) classification with a linear kernel.

During detection, a new image is acquired and window is scrolled through the image at each position and scale. HOG is computed in each step and classified via the learned SVM. A non maxima suppression is used in order to refine results and solve overlapping detections ambiguity. A computationally faster version of HOG, based on Adaboost SVM cascades on a non-regular window tesselation has been developed by Zhu et al. (2006). Other variants leverage the parallel processing power of GPU (see Prisacariu and Reid (2009)) achieving $120Hz$ in classification speed. Occlusion handling and overall performance has been improved with the HOG-LBP detector, a novel detector that combines the HOG feature concept with local binary patterns classification (LBP) (see Wang et al. (2009)). People detection and very small scales has been addressed by Spinello et al. (2009), in which small edge segments, extracted via a superpixel segmentation, are assembled in a discriminative voting approach.

For car detection, simplistic computer vision approaches that exploit shadow detection (Dickmanns et al., 1994), trunk-frontal symmetries (Kuehnle, 1991), rectangular shapes (Bertozzi et al., 1997) or vehicle lights (Cucchiara and Piccardi, 1999) are nowdays overcame by HOG/ISM detectors trained with databases of car images (Ess et al., 2009; Leibe et al., 2007). Cars are learned with several viepoint-dependent classifiers in order to account for the difference in appearance with respect to the object viewpoint.

4.1.2 Object and People Detection from 2D Laser Data

To detect objects in 2D laser range scans, several approaches have been presented in the past (see, e.g. Fod et al. (2002); Kleinhagenbrock et al. (2002); Schulz et al. (2003a); Xavier et al. (2005); Arras et al. (2007)). These approaches extract features from the 2D scan data, which are used to classify the data points into people or background. The features used can be motion features, geometric features such as circles and lines, or a combination of these, and they are either predetermined or can be learned from hand-labeled data as in Arras et al. (2007). However, most of these approaches have the disadvantage that they disregard the conditional dependence between

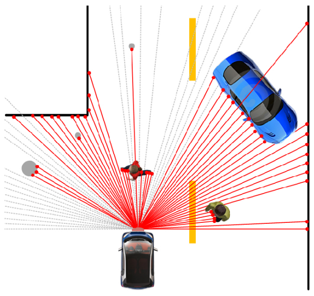


Figure 16: Example of a 2D laser scan. Laser beams are shown in red, while circles represent the measured points. Gray beams indicate out-of-range readings which, apart from the absence of an obstacle, can be caused by material reflections, sun light effects and a too large incidence angle between the laser beam and the surface normal.

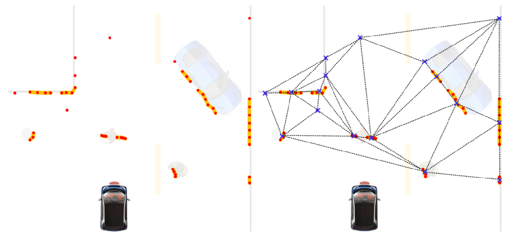


Figure 17: Object detection from 2D laser range scans using Conditional Random Fields (CRFs). The 2D data points shown here correspond to the scan depicted in Fig. 16. **Left**: First, the 2D scan points are clustered using jump distance clustering. The yellow lines correspond to the resulting clusters. **Right**: In a second step, a Delaunay triangulation is computed on the centroids of the segments. This triangulation defines the structure of the CRF.

data in a close neighborhood. In particular, they can not model the fact that the label l_i of a given laser segment $\mathcal{S}_i$ is more likely to be l_j if we know that l_j is the label of $\mathcal{S}_j$ and that $\mathcal{S}_j$ and $\mathcal{S}_i$ are neighbors. One way to model this conditional dependency is to use Conditional Random Fields (CRFs) (Lafferty et al., 2001), as shown by Douillard et al. (2008). CRFs represent the conditional probability $p(\mathbf{l} \mid \mathbf{s})$ using an undirected cyclic graph, in which each node is associated with a hidden random variable l_i and an observation $\mathbf{s}_i$. For example, $\mathbf{l}$ can be a vector of discrete labels that range over the 3 different classes 'pedestrian', 'car' and 'background', where $\mathbf{s}$ are the feature vectors extracted from the 2D segments in the laser scan. To obtain the structure of the CRF, the 2D scan needs to be clustered first, which can be done using jump distance clustering, a method that assigns two adjacent points to the same cluster if they are not further away from each other than a given threshold. As a result, all further reasoning is done on the clusters instead of the individual data points. Then, in a second step the graph structure needs to be computed. This can be done using a Delaunay triangulation on the centroids of each segment. The resulting graph connects clusters that are close to each other. The intuition behind this is that neighboring clusters have a higher likelihood to belong to the same class, which is modeled with a statistical dependency in form of an edge in the CRF (see Fig. 17).

Assuming that the maximal clique size of the graph is 2, one can compute the conditional probability of the labels $\mathbf{l}$ given the observations $\mathbf{s}$ as:

$$p(\mathbf{l} \mid \mathbf{s}) = \frac{1}{Z(\mathbf{s})} \prod_{i=1}^{N} \varphi(\mathbf{s}_i, l_i) \prod_{(i,j)\in\mathcal{E}} \psi(\mathbf{s}_i, \mathbf{s}_j, l_i, l_j), \qquad (9)$$

where $Z(\mathbf{s}) = \sum_{\mathbf{l}'} \prod_{i=1}^{N} \varphi(\mathbf{s}_i, l'_i) \prod_{(ij)\in\mathcal{E}} \psi(\mathbf{s}_i, \mathbf{s}_j, l'_i, l'_j)$ is usually called the *partition function*, $\mathcal{E}$ is the set of edges in the graph, and φ and ψ represent node and edge potentials. These potentials are positive functions that return large values if the labels l_i correspond to the correct labels of the segments $\mathbf{s}_i$ with a high probability. This means, that the potentials can be viewed as classifiers, where the node potential φ only uses the local information (i.e. features) of a segment for the classification, and the edge potential ψ measures the *consistency* of the labels between two adjacent segments. To define the node and edge potentials simple rules can be used that relate the feature vector with the assigned class label, but it turns out that using a classifier such as AdaBoost (Freund and Schapire, 1997) for the node potentials φ and a simple rule for the edge potentials gives very good classification results (see Spinello et al. (2010)). To achieve this, a *classification score* g_c is computed for each class based on the M weak classifiers h_i^c and their corresponding weight coefficients α_i^c as returned by the AdaBoost training algorithm:

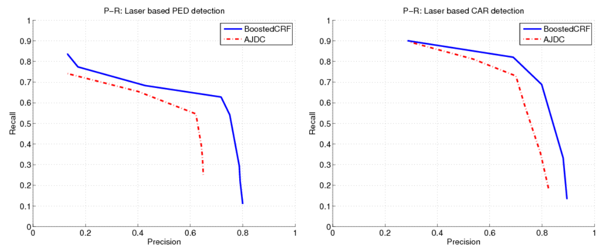


Figure 18: Classification results of pedestrians and cars from 2D laser range scans. The results obtained using a Conditional Random Field on node potentials using AdaBoost are compared with the results of an AdaBoost approach that is directly applied to the jump-distance clustered segments (AJDC). As can be seen, the boosted CRF performs better than AdaBoost alone (results taken from Spinello et al. (2010)).

$$g_c(\mathbf{s}_i) := \sum_{i=1}^{M} \alpha_i^c h_i^c(\mathbf{s}_i). \tag{10}$$

To obtain a classification *likelihood*, the logistic function $\lambda(x) = (1+e^{-x})^{-1}$ can be applied to g_c. For the edge potential, a good choice are rules such as "the closer two segments are to each other the higher is the chance that they correspond to the same class" or "the higher the classification score is for the two adjacent nodes on a given class, the higher is the probability that they have the same class labels". For details about how this can be expressed in a mathematical formulation see Spinello et al. (2010).

Fig. 18 shows precision-recall curves for results obtained with the boosted-CRF approach. As one can see, the additional information about the statistical dependence of labels from adjacent segments leveraged by the CRF approach effectively improves the classification results over the standard AdaBoost method.

4.1.3 Example of detection by classifier ensemble

A feature-based fusion can be accomplished by a monolithic classifier method or an ensemble of classifiers. Particularly, in the latter case, the decisions are spread into small specialized sub-modules, which are in charge of deciding over the feature representations. In this section, an ensemble fusion method, which performs over a set of features and image classifiers, are presented and discussed in a pedestrian detection task. In fact, concerning image classification, a single feature extractor-classifier is not usually able to deal with the diversity of multiple scenarios. Therefore, the integration of features and classifiers can bring benefits to cope with this problem, particularly when the parts are carefully chosen and synergistically combined.

Fusion of classifiers has been studied in the last few years with the aim of overcoming certain inabilities of single classifiers (Kuncheva (2004)). The objective is to explore diversity of the component classifiers in order to enhance the overall classification performance. In other words, since there is no perfect individual classifier yet, by assembling them, one can complement the other. If there are errors in the ensemble components, it is expected that they occur on different image objects in order to give rise to fusion methods that improve the performance of the whole system. Additionally, the rationale of building ensembles is also that most individual classifiers agree in a certain way, such that the whole system can be more successful than its component parts.

From experimental studies, a set of feature extractors and strong classifiers were chosen which could interact together to improve the general performance of the final classification system. In this sense, the choice of the feature extractors, histogram of oriented gradients (HOG) and local receptive fields (LRF), was motivated by the studies found in Dalal and Triggs (2005b), Munder and Gavrila (2006) and Szarvas et al. (2006). Dalal and Triggs (2005b) presented an experimental analysis demonstrating that HOG features outperforms PCA-SIFT, Haar wavelets and shape contexts in a complex data set. Munder and Gavrila Munder and Gavrila (2006) also experimentally showed that LRF features present superior performance in comparison with PCA and Haar wavelets, although computed from an MLP over Haar wavelets, or PCA features, and classified by SVM or Adaboost, which turned it to a method more sensitive to lighting variations. Szarvas et al. Szarvas et al. (2006) found that LRFs built on CNNs have great potential in pedestrian recognition. Our goal is thus to show that there is an opportunity to integrate synergistically the outputs of high performance classifiers performing over these two types of features.

After an extensive experimental work, the final architecture of the proposed synergistic method was built, and is illustrated in Fig. 19 (see Oliveira et al. (2010) for the complete description of the ensemble evaluation). In the feature level, HOG and LRF feature extractors are in charge of representing the objects in different ways. It was demonstrated that HOG are better to represent pedestrians, while LRF are better to represent background. Actually, they complement each other, in many circumstances with respect to pedestrian/non-pedestrian classification ability. Instead of employing weak classifiers in the fashion of boosting methods, strong classifiers were used in order to explore the complementarity of the features. It means that once the chosen features provide synergism in the way of acting, the lesser errors they commit individually, the better the integration made by the fusion method. The name of the method was coined as HLSM-FINT, standing for each initial letter of the feature extractors and classifiers used in the ensemble.

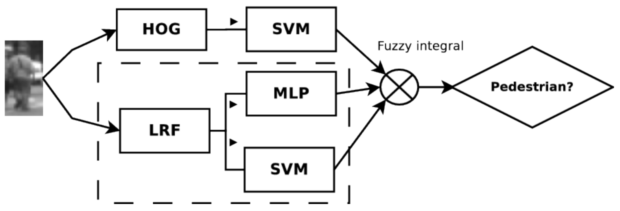


Figure 19: Classifier ensemble synergism: The component classifiers, SVM and MLP, runs over the feature extractors HOG and LRF. After a thorough evaluation of a bunch of methods, a Sugeno fuzzy integral method was chosen as the best one to combine the classifier results.

HLSM-FINT was evaluated, by means of receiver operating characteristic (ROC) curves, over DaimlerChrysler Munder and Gavrila (2006), INRIA Dalal and Triggs (2005b) and Nature Inspired Smart Information System (NISIS) competition datasets Oliveira et al. (2010). Furthermore, the DaimlerChrysler datasets were changed to incorporate some usual situations involving lighting and contrast changes. At the end, HLSM-FINT presented an averaged performance of 94% of hit rate (HR) with 3% of false alarm rate (FAR) over all datasets evaluated.

4.2 Detection by sensor fusion

Actually, in the last decades, several researchers have been developing complete perception architectures for intelligent transportation systems (ITS) Reisman et al. (2004), Leibe et al. (2005b), Leibe et al. (2007).

In real life, perception systems have become a reality with the deployment of camera-based object-detection and lane departure systems, in top-of-the-line vehicles. Although these systems are intended to aid the driver in hazardous situations, much has yet to be done in order to make them completely reliable in several circumstances. In this regard, multi-sensor architectures may bring complementarity and redundancy, making perception systems more robust. The idea is to build a system to take advantage of the strengths of various sensors.

For intelligent vehicle applications, some sensors appear as the best choices to perceive the environment in order to provide sensing information and intelligent decisions. They are range finders like lasers and cameras. With those sensors, there are many approaches with the aim of getting the best of sensor data integration. For instance, Perrollaz et al. (2010) proposes a probabilistic representation of the uncertainty of the fusion of two cameras (stereo-vision); they build this representation into a grid framework for object detection. Oliveira et al. (2010) suggest that a semantic fusion of laser and vision in a feature level is highly advantageous for object detection, if it is possible to decouple the detection dependency of the vision system with respect to the laser detection Oliveira and Nunes (2010). Scheunert et al. (2008) integrate a stereo vision with a laserscanner for obstacle avoidance; that work is also structured in a occupancy grid framework as in Perrollaz et al. (2010). Spinello et al. (2009) propose a cooperative fusion of independent sensor detection of a laserscanner and a vision system; both sensors are fused by means of a tracking system used in each sensor space with respective data association.

A feature based fusion is characterized by an integration in the feature level. One should expect a general feature-based fusion framework depicted in Fig. 20 (for a broader view of various fusion architectures, see Dasarathy (1997)).

In Figure 20, it is shown how the feature-based fusion is accomplished. The figure depicts a general framework regardless of where the features are coming. Indeed, features may be obtained from a unique sensor or multiple ones. Also, the features can represent an object in different sensor spaces (for instance, laser and camera). For all those reasons, the choice of the fusion method is of underlying importance, and the fusion architecture ought to be designed taking into account the different characteristics

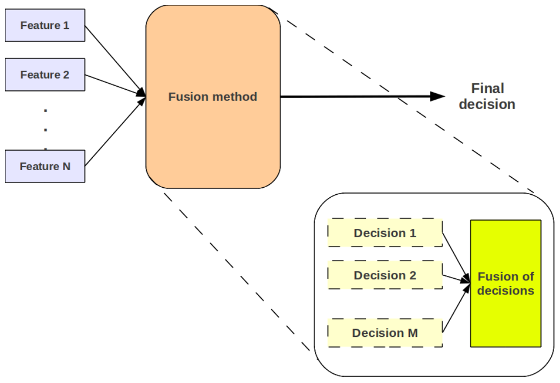


Figure 20: Feature based framework. Features 1...N are extracted in such a way that they could represent the object of interest in different manners. After that, the features are fused in the fusion method, which is ultimately structured as a monolithic decision module or a set of specialized decision sub-modules integrated in a more generic method.

brought for each feature extractor.

In turn, the feature extractors are in charge of getting good representations of an object of interest in the scene. For "good", one may expect that a particular object should be represented differently from other objects pertaining to a different category. Unfortunately, obtaining this unique representation for each different type of an object is cumbersome, if not impossible, which means that errors will be encountered. Those errors will be dealt with in the fusion method, which will later decide, in a certain level (monolithically or via a classifier ensemble methods), how to choose the best representation among the inputs (features 1...N).

The choice of the fusion methods can be made namely by considering the performance of the specialized decision sub-modules (if they are presented in the architecture), the characteristics of the feature extractors and/or how much information each feature carries on.

Another aspect of a fusion method design is the information that each feature or the specialized fusion sub-modules brings with them. The way to deal with this feature inherent information is crucial since it is expected that a fusion method explores not only redundancy but also complementarity of information; in the first case, the fusion will try to match similar characteristics that reinforces the decision, while in the latter one, the fusion method appeals to the fact that one feature representation can complement information missing in the others. The goal for the complementarity is also to decrease the decision uncertainty by building the final decision from small pieces of information coming from multiple sources or representations. Next, a sensor fusion method based on semantic information of parts-based detectors is described.

Semantic fusion So far, fusion of laserscanner and vision sensors has been performed by assuming that the probability to find an object is identical and usually independent, in both sensor spaces. There are two main approaches for this type of fusion:

- A laserscanner segmentation method is used to find most likely regions of interest (ROIs), where an image classification system is applied.
- Indentically and independent distribution (IID) integration of sensor-driven classifiers (posterior probabilities), or sensor-driven features.

Some of the works based on these ROI-driven methods are presented as follows. Szarvas et al. (2006) rely entirely on laser ROIs in order to find probable areas where a pedestrian might be. Each image projected ROI is classified by a convolutional neural network (CNN) Lecun et al. (1998). When the laser fails, no pedestrians are detected. Broggi et al. (2008) propose an on-spot pedestrian classification system, which first uses a laserscanner to determine areas between vehicles, and then a Haar-like feature/adaboost classification system to name the projection of these areas in the image. Mahlisch et al. (2006) propose a spatio-temporal alignment to integrate the laserscanner and the monocular camera. For that purpose, features in both sensor spaces are extracted, feeding a Bayesian classifier to detect cars. Douillard et al. Douillard et al. (2007) propose an approach similar to the previous one, but rather than a Bayesian rule they use a conditional random field conditional random fields (CRF). Additionally, they are able not only to classify laser and image features, but also to find a temporal relationship between features of sequential sensor readings. Premebida et al. (2007) propose to classify features in laser space with a Gaussian mixture model (GMM) while Haar-like features are

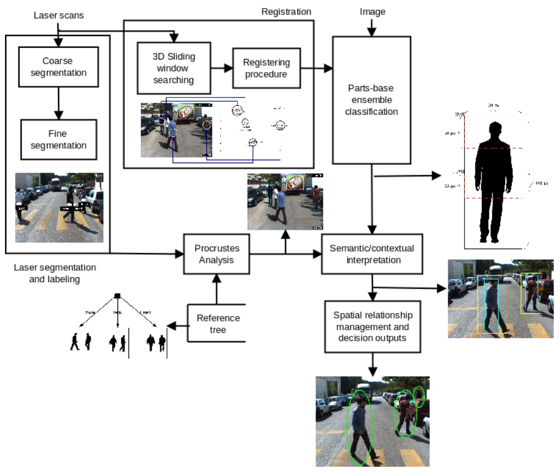

Laser scans
Registration
Image
Coarse segmentation
3D Sliding window searching
Registering procedure
Parts-base ensemble classification
Fine segmentation
Laser segmentation and labeling
Procrustes Analysis
Semantic/contextual interpretation
Reference tree
Spatial relationship management and decision outputs

Figure 21: The proposed framework is composed of 6 main blocks: The image object detection based on a parts-based ensemble detector using HLSM-FINT; laser segmentation and labeling; template matching; laser-vision registration, based on Zhang and Pless' method Zhang and Pless (2004); semantic and contextual interpretation; and the semantic fusion based on MLN.

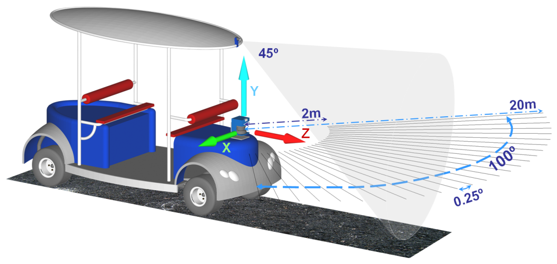


Figure 22: Setup vehicle: The perception system ranges from 2 up to 20m in both sensor spaces. A SICK 2D laserscanner (100^o degrees of aperture angle) and a PointGrey camera set to a resolution of 1024×768 pixels (45^o degrees of field of view) were used.

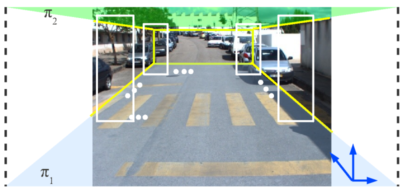


Figure 23: A window sized 1.0m $\times$ 1.8m is shifted onto horizontal and vertical directions in laser sensing space, with a stride of 0.20m, ranging over 2m up to 20m in depth. The searching area is constrained by the viewport formed by the planes π_1 and π_2.

extracted from the image objects and classified by an adaboost. The confidence scores of the classifiers feed a Bayesian rule to make the final decision. The goal is to classify vehicles and pedestrians.

Unlike the aforementioned approaches, the semantic fusion method Oliveira et al. (2010) deals with partial segments, it is able to recover depth information even if the laser fails, and the integration is modeled through contextual information. Figure 21 depicts the proposed framework, which is comprised of 6 main blocks: The image object detection based on the HLSM-FINT classifier ensemble (see Section 4.1.3 for more details); laser segmentation and labeling; template matching based on Procrustes analysis; a laser-vision registration procedure, proposed by Zhang and Pless Zhang and Pless (2004); semantic and contextual interpretation; and spatial relationship management and decision outputs based on Markov logic network (MLN) (Richardson and Domingos (2006)).

As it can be noticed, features are extracted in each sensor space. Here, it is important to say that there are 2 sensor geometric spaces, considering the laser and the camera spaces. However, there is an intermediary space where all resulting detections are unified, that is, the image space. In the image space, to guarantee that the individual detections in both sensors will be independent, that is, non-ROI based, a 2D window is slid in laser space, and subsequently projected into the image space. This has three-fold advantages: the economy on windows being slid in various scales and positions in the image space; a 3D control of the vision detector, which allows a depth information even if the laser fails; and, finally, the projections of the laser segments can also be treated independent in the same projections, in the image space. For those sliding windows in the 3D laser space, it is assumed that the laser is always parallel to the ground (see Fig. 22). Figure 23 illustrates the geometry of this process.

The segmentation, in laser space, uses a featureless approach. The rationale of this method is to segment coarsely the laser raw points. These coarse segments, $\{c_n\}$, where $n = 1, ..., N$, are posteriorly sub-segmented into fine segments, $\{f_m\}$, where $m = 1, .., 3$. This latter step is done by a β-skeleton random graph. As the main goal is to detect pedestrians in outdoor, the laser is mounted at the pedestrian-waist level (see Fig. 22), avoiding common problems as those observed in leg-level mounted systems. Therefore, the fine segmentation step expects at most 3 body parts (2 arms and the torso).

On the other hand, in image space, the HLSM-FINT is applied in a parts-based detector. Actually, the INRIA datasets Dalal and Triggs (2005b) were used to train a two-parts detector, which classifies the hip region and the shoulder-head region of the human body. The idea is to allow the image detector to cope with partial occlusion in the same way that the laser detector does. With that, there are parts being labeled in image and laser spaces. Each of these parts is then contextually interpreted, and finally modeled by an MLN.

In MLN framework (Richardson and Domingos (2006)), the information of the parts is modeled according to first-order clauses. The first-order rules are then structured as a Markov random fields (MRF) with a grounded first-order formula in each node, which is so called a ground MRF. After the inference process over the MRF, marginal probabilities are then computed and outputted for each test case. To build the MLN, we have used the Alchemy library, available at http://alchemy.cs.washington.edu/.

As mentioned before, the semantic fusion of laser and vision differs from previous fusion approaches on these sensors, in the following twofold reasons: non ROI-driven, and non-IID based fusion. These characteristics led to a very robust system with independent and synergistic sensor fusion, based on a semantic approach (see Fig. 21).

The rationale of the semantic fusion system is based on parts-based classifiers in both sensor spaces, which are integrated with the detachment of the vision detector from the laser detector. This independent fusion was achieved by the 2D sliding window running in 3D laser space and the processing of the object parts in a non-IID framework, that is, MLN. With that, it was possible to make the vision detection system to run regardless of the laser hypothesized ROI. The validation of the semantic method was made over collected datasets, which are available at http://www.isr.uc.pt/~lreboucas. Some results of the method application over the collected image frames is depicted in Fig. 24.

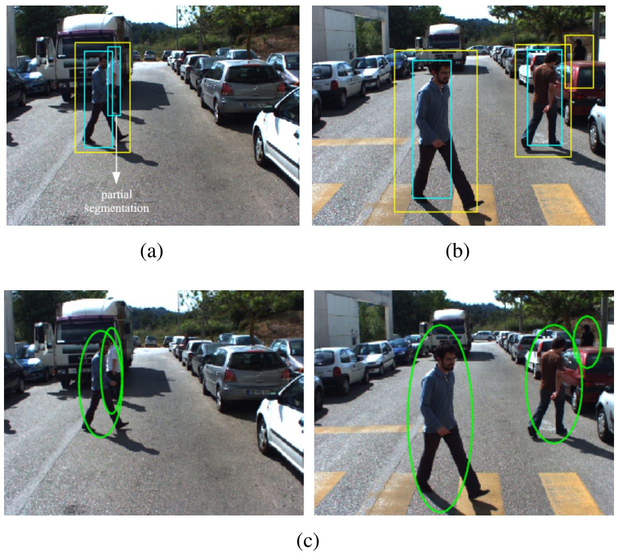


Figure 24: Samples of the semantic fusion in action. In (a) and (b), results of the individual classification in each sensor space (the yellow bounding box is given by an image detector, while the blue one is given by a laser segmentation). In (b), an example where the laser fails and the image detector complements the recognition process. Note that the depth information when laser fails is estimated from the 3D searching windows. In (c), the result of the semantic fusion (left to right) from (a) and (b), respectively.

5 Conclusions

This chapter provided an overview of the DATMO problem for autonomous driving and outlined three main classes of solutions: traditional, model based, and grid based.

The traditional approach consists of data segmentation, association, and filtering steps. Recent innovations within this class are lead by pattern recognition techniques, which are capable of recognizing scene participants (e.g. pedestrians and vehicles) from a single frame of data. These techniques are particularly useful for identifying potentially dynamic objects when these objects are not yet moving.

The model based approach is able to infer and use additional shape knowledge, which is particularly useful when working with large objects such as cars, buses, and trucks. These objects are often split up into separate clusters due to occlusion, making segmentation and association of data difficult. Using geometric models, the model based approach bypasses segmentation and association steps altogether. It also leads to more accurate motion estimation, which is particularly important for objects with non-holonomic motion and/or objects moving at high speeds.

The grid based approach delays data association decisions to later stages of the DATMO pipeline. At early stages it constructs a low level grid representation capturing occupancy and velocity estimates of grid cells. This representation can be used directly for motion planning in dynamic free space or supplemented by segmentation and association techniques to identify individual targets. The grid based approach is particularly useful for identifying a broad variety of objects regardless of their appearance. This makes it capable of identifying small participants such as young children and pets.

DATMO for autonomous driving is a young and rapidly developing field. Although many interesting and successful methods have been developed, a fully reliable autonomous DATMO system is yet to be built. For now, existing DATMO systems can not rival human capabilities in terms of detection range. Existing DATMO approaches have not yet been shown to be fully reliable with respect to participants of unusual appearance, for example unusual vehicles (e.g. construction equipment or parade floats), people wearing extravagant costumes or fully draped in loose clothing, animals, or people pushing various objects. Performance of DATMO techniques in adverse weather or lighting conditions is yet to be studied in depth.

References

Andrieu, C., N. De Freitas, A. Doucet, and M. Jordan (2003). An introduction to mcmc for machine learning. *Machine learning 50*(1), 5–43.

Arras, K., O. Mozos, and W. Burgard (2007). Using boosted features for the detection of people in 2d range data. In *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, Rome, Italy, pp. 3402–3407.

Arulampalam, S., S. Maskell, N. Gordon, and T. Clapp (2002). A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*.

Badino, H., U. Franke, and R. Mester (2007). Free space computation using stochastic occupancy grids and dynamic programming. In *IEEE Int. Conf. on Computer vision, Workshop on Dynamical Vision*, Rio de Janeiro, Brazil.

Bar-Shalom, Y. and A. Jaffer (1972). Adaptive nonlinear filtering for tracking with measurements of uncertain origin. In *Conference on Decision and Control and Symposium on Adaptive Processes, 11 th, New Orleans, La; United States*, pp. 243–247. New York, Institute of Electrical and Electronics Engineers, Inc.

Bertozzi, M., A. Broggi, and S. Castelluccio (1997, March). A real-time oriented system for vehicle detection. *J. Syst. Archit. 43*, 317–325.

Bertozzi, M., A. Broggi, and A. Fascioli (2000). Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous systems 32*, 1–16.

Blackman, S., R. Co, and C. El Segundo (2004). Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine 19*(1 Part 2), 5–18.

Blackwell, D. (1947). Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics 18*(1), 105–110.

Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics and Image Processing 34*, 344–371.

Braillon, C., C. Pradalier, K. Usher, J. Crowley, and C. Laugier (2006). Occupancy grids from stereo and optical flow data. In *Proc. Int. Symp. on Experimental Robotics*, Rio de Janeiro, Brazil.

Broggi, A., P. Cerri, S. Ghidoni, P. Grisleri, and J. Gi (2008). Localization and analysis of critical areas in urban scenarios. In *Proc. of IEEE Intl. Symp. on Intelligent Vehicles*, pp. 1074–1079.

Comaniciu, D. and P. Meer (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis & Machine Intelligence 24*, 603–619.

Coué, C., C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière (2006, January). Bayesian occupancy filtering for multitarget tracking: an automotive application. *Int. Journal of Robotics Research 25*(1), 19–30.

Cox, I. (1993). A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision 10*(1), 53–66.

Cucchiara, R. and M. Piccardi (1999). Vehicle detection under day and night illumination. In *International ICSC Symposium on Intelligent Industrial Automation*.

Dalal, N. and B. Triggs (2005a, Jun.). Histograms of oriented gradients for human detection. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, Volume 1, San Diego, USA, pp. 886–893.

Dalal, N. and B. Triggs (2005b). Histograms of oriented gradients for human detection. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, Volume 1, pp. 886–893.

Dasarathy, B. (1997). Sensor fusion potential exploitation – innovative and illustrative applications. In *Proceedings of the IEEE Special Issue on Sensor Fusion*, Volume 85, pp. 24–38. IEEE.

Diard, J., P. Bessiere, and E. Mazer (2003). A survey of probabilistic models using the bayesian programming methodology as a unifying framework. In *In The Second International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003)*.

Dickmanns, E., R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek, and J. Schiehlen (1994). The seeing passenger car 'vamors-p'. In *Intelligent Vehicles '94 Symposium, Proceedings of the*, pp. 68 – 73.

Dollar, P., C. Wojek, B. Schiele, and P. Perona (2009). Pedestrian detection: A benchmark. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, pp. 304 –311.

Douillard, B., D. Fox, and F. Ramos (2007). A spatiotemporal probabilistic model for multi-sensor object recognition. In *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 2402–2408.

Douillard, B., D. Fox, and F. Ramos (2008, Jun.). Laser and vision based outdoor object mapping. In *Proceedings of Robotics: Science and Systems (RSS)*, Zurich, Switzerland.

Enzweiler, M. and D. Gavrila (2009). Monocular pedestrian detection: Survey and experiments. *IEEE Trans. on Pattern Analysis & Machine Intelligence 31*(12), 2179–2195.

Ess, A., B. Leibe, K. Schindler, and L. V. Gool (2009). Moving obstacle detection in highly dynamic scenes. In *ICRA*.

Felzenszwalb, P., D. McAllester, and D. Ramanan (2008). A discriminatively trained, multiscale, deformable part model. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, pp. 1 –8.

Fod, A., A. Howard, and M. Mataric (2002, May). Laser-based people tracking. In *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, Volume 3, Washington, USA, pp. 3024 –3029.

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

Freund, Y. and R. E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences 55*(1), 119–139.

Gavrila, D. (2000). Pedestrian detection from a moving vehicle. In *European Conf. on Computer Vision (ECCV)*, Dublin, Ireland, pp. 37–49. IEEE Computer Society.

Gordon, N. (1993). *Bayesian methods for tracking*. Ph. D. thesis, University of London.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika 57*(1), 97–109.

Kleinhagenbrock, M., S. Lang, J. Fritsch, F. Lömker, G. Fink, and G. Sagerer (2002). Person tracking with a mobile robot based on multi-modal anchoring. In *IEEE International Workshop on Robot and Human Interactive Communication (ROMAN)*.

Kuehnle, A. (1991, April). Symmetry-based recognition of vehicle rears. *Pattern Recogn. Lett. 12*, 249–258.

Kuncheva, L. (2004). *Combining pattern classifiers: Methods and algorithms*. Wiley-Interscience.

Labayrade, R., D. Aubert, and J. Tarel (2002). Real time obstacles detection on non flat road geometry through v-disparity representation. In *IV*.

Lafferty, J., A. McCallum, and F. Pereira (2001, jun.). Conditional random fields: Probabilistic models for segmentation and labeling sequence data. In *Int. Conf. on Machine Learning (ICML)*, Williamstown, USA, pp. 282–289.

Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. *86*(11), 2279–2324.

Leibe, B., N. Cornelis, K. Cornelis, and L. V. Gool (2007, Jun.). Dynamic 3d scene analysis from a moving vehicle. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, Minneapolis, USA, pp. 1–8.

Leibe, B., N. Cornelis, K. Cornelis, and L. Van Gool (2007). Dynamic 3d scene analysis from a moving vehicle. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, Volume 1, pp. 1–8.

Leibe, B., E. Seemann, and B. Schiele (2005a, Jun.). Pedestrian detection in crowded scenes. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, Volume 1, San Diego, USA, pp. 878–885.

Leibe, P., E. Seemann, and B. Schiele (2005b). Pedestrian detection in crowded scenes. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, Volume 1, pp. 878–885. IEEE.

Leonard, J., J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, et al. (2008). A perception-driven autonomous urban vehicle. *Journal of Field Robotics 25*(10), 727–774.

Lundquist, C. and T. Schon (2008). Road geometry estimation and vehicle tracking using a single track model. In *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 144–149. IEEE.

MacKay, D. J. C. (1998). Introduction to Monte Carlo methods. In M. I. Jordan (Ed.), *Learning in Graphical Models*, NATO Science Series, pp. 175–204. Kluwer Academic Press.

Mahlisch, R. S., W. Ritter, and K. Dietmayer (2006). Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection. In *Proc. of IEEE Intl. Symp. on Intelligent Vehicles*, pp. 424–429.

Matthies, L. and A. Elfes (1988). Integration of sonar and stereo range data using a grid-based representation. In *IEEE Int. Conf. on Robotics and Automation*, Philadelphia, USA.

Mekhnacha, K., Y. Mao, D. Raulo, and C. Laugier (2008). Bayesian occupancy filter based Fast Clustering-Tracking algorithm. In *IEEE/RSJ Int. Conf. on Intelligent Robot and Systems*, Nice, France.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, et al. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics 21*(6), 1087.

Mikolajczyk, K., C. Schmid, and A. Zisserman (2004). Human detection based on a probabilistic assembly of robust part detectors. In *European Conf. on Computer Vision (ECCV)*, pp. 69–82.

Mohan, A., C. Papageorgiou, and T. Poggio (2001). Example-based object detection in images by components. *IEEE Trans. on Pattern Analysis & Machine Intelligence 23*(4), 349–361.

Montemerlo, M. (2003). *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. Ph. D. thesis, Robotics Institute, Carnegie Mellon University.

Montemerlo, M., J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Hähnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun (2008). Junior: The stanford entry in the urban challenge. *Journal of Field Robotics 25*(9), 569–597.

Moravec, H. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine 9*(2).

Munder, S. and D. Gavrila (2006). An experimental study on pedestrian classification. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 28, pp. 1863–1868.

Murphy, K. and S. Russell (2001). *Rao-blackwellized particle filtering for dynamic bayesian networks*. Springer.

Murray, D. and J. Little (2000, January). Using real-time stereo vision for mobile robot navigation. *Autonomous Robots, 8*.

Oliveira, L. and U. Nunes (2010). Context-aware pedestrian detection using lidar. In *Proc. IEEE Intl. Intelligent Vehicles Symposium*, pp. 773–778.

Oliveira, L., U. Nunes, and P. Peixoto (2010). On exploration of classifier ensemble synergism in pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 16–27.

Oliveira, L., U. Nunes, P. Peixoto, M. Silva, and F. Moita (2010). Semantic fusion of laser and vision in pedestrian detection. *Pattern Recognition 43*, 3648–3659.

Papageorgiou, C. and T. Poggio (2000). A trainable system for object detection. *Int. Journ. of Computer Vision 38*(1), 15–33.

Perrollaz, M., A. Spalanzani, and D. Aubert (2010). Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection. In *Proc. of IEEE Intl. Symp. on Symposium on Intelligent Vehicles*, pp. 313–318.

Perrollaz, M., J.-D. Yoder, and C. Laugier (2010). Using obstacle and road pixels in the disparityspace compputation of stereovision based occupancy grids. In *Proc. of IEEE International Conf. on Intelligent Transportation Systems*.

Petrovskaya, A. (2011, June). *Towards Dependable Robotic Perception*. Ph. D. thesis, Stanford University, Stanford, CA.

Petrovskaya, A. and O. Khatib (2011, June). Global localization of objects via touch. *IEEE Transactions on Robotics 27*(3).

Petrovskaya, A. and S. Thrun (2009). Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots 26*(2), 123–139.

Premebida, C., G. Monteiro, U. Nunes, and P. Peixoto (2007). A lidar and vision-based approach for pedestrian and vehicle detection and tracking. In *Proc. of IEEE Intl. Conf. on Intelligent Transportation Systems*, pp. 1044–1049.

Prisacariu, V. A. and I. Reid (2009). fasthog- a real-time gpu implementation of hog technical report no. 2310/09.

Rao, C. (1945). Information and accuracy obtainable in one estimation of a statistical parameter. *Bull. Calcutta Math. Soc 37*, 81–91.

Reid, D. B. (1979). An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control 24*, 843–854.

Reisman, P., O. Mano, S. Avidan, and A. Shashua (2004). Crowd detection in video sequences. In *Proc. of IEEE Intl. Symp. on Symposium on Intelligent Vehicles*, pp. 66–71. IEEE.

Richardson, M. and P. Domingos (2006). Markov logic networks. *Machine Learning 62*, 107–136.

Richter, E., R. Schubert, and G. Wanielik (2008). Radar and vision based data fusion-advanced filtering techniques for a multi object vehicle tracking system. In *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 120–125. IEEE.

Scheunert, U., N. Mattern, P. Lindner, and G. Wanielik (2008). Generalized grid framework for multi sensor data fusion. *Journal on Information Fusion*, 814–820.

Schulz, D., W. Burgard, D. Fox, and A. Cremers (2003a). People tracking with mobile robots using sample-based joint probabilistic data ass. filters. *Int. Journ. of Robotics Research (IJRR) 22*(2), 99–116.

Schulz, D., W. Burgard, D. Fox, and A. B. Cremers (2003b). People tracking with mobile robots using sample-based joint probabilistic data association filters. *I. J. Robotic Res. 22*(2), 99–116.

Sittler, R. (1964). An optimal data association problem in surveillance theory. *Military Electronics, IEEE Transactions on 8*(2), 125–139.

Spinello, L., A. Macho, R. Triebel, and R. Siegwart (2009, Oct.). Detecting pedestrians at very small scales. In *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*, St. Louis, USA, pp. 4313–4318.

Spinello, L., R. Triebel, and R. Siegwart (2009). A trained system for multimodal perception in urban environments. In *Proc. of the Workshop on Safe Nav. in Open and Dyn. Env.*

Spinello, L., R. Triebel, and R. Siegwart (2010). Multiclass multimodal detection and tracking in urban environments. *International Journal of Robotics Research (IJRR) 29*(12), 1498–1515.

Sun, Z., G. Bebis, and R. Miller (2006). On-road vehicle detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 694–711.

Szarvas, M., U. Sakai, and J. Ogata (2006). Real-time pedestrian detection using lidar and convolutional neural networks. In *Proc. of IEEE Intl. Symp. on Intelligent Vehicles*, pp. 213–218.

Thrun, S., W. Burgard, and D. Fox (2005). *Probabilistic Robotics*. MIT Press.

Tuzel, O., F. Porikli, and P. Meer (2007). Human detection via classification on riemannian manifolds. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, pp. 1 –8.

Vermaak, J., S. Godsill, and P. Perez (2005, January). Monte carlo filtering for multi target tracking and data association. *Aerospace and Electronic Systems, IEEE Transactions on 41*(1), 309 – 332.

Viola, P., M. J. Jones, and D. Snow (2003, Oct.). Detecting pedestrians using patterns of motion and appearance. In *IEEE Int. Conf. on Computer Vision (ICCV)*, Volume 2, Nice, France, pp. 734 –741.

Vu, T. (2009, September). *Vehicle Perception: Localization, Mapping with Detection, Classification and Tracking of Moving Objects*. Ph. D. thesis, INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE.

Vu, T. and O. Aycard (2009). Laser-based detection and tracking moving objects using data-driven markov chain monte carlo. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 3800–3806. IEEE.

Wang, X., T. X. Han, and S. Yan (2009). An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 32 –39.

Wu, B. and R. Nevatia (2005). Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *IEEE Int. Conf. on Computer Vision (ICCV)*.

Xavier, J., M. Pacheco, D. Castro, A. Ruano, and U. Nunes (2005, Apr.). Fast line, arc/circle and leg detection from laser scan data in a player driver. In *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, Barcelona, Spain, pp. 3930–3935.

Zhang, Q. and R. Pless (2004). Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Volume 3, pp. 2301–2306.

Zhu, Q., M. C. Yeh, K. T. Cheng, and S. Avidan (2006, Jun.). Fast human detection using a cascade of histograms of oriented gradients. In *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, Volume 2, New York, USA, pp. 1491–1498.

Index