UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXVIII Ciclo

MULTISENSORIAL OBSTACLES DETECTION AND CLASSIFICATION FOR ADAS AND AUTONOMOUS DRIVING

Coordinator:

Prof. Marco Locatelli

Supervisor:

Prof. Alberto Broggi

Tutor:

Prof. Pietro Cerri

PhD student: *Antonio Prioletti*

December 2015

Happiness must be sought in themselves

List of contents

List of Figures

List of Tables

Introduction

A reliable perception of the real world is a key-feature for an autonomous vehicle and the Advanced Driver Assistance Systems (ADAS). Obstacles detection (OD) is one of the main components for the correct reconstruction of the dynamic world. Historical approaches based on stereo vision and other 3D perception technologies (e.g. LIDAR) have been adapted to the ADAS first and autonomous ground vehicles, after, providing excellent results as shown in Fig. 1.

The obstacles detection is a very broad field and this domain counts a lot of works in the last years [32]. In academic research [121, 95], it has been clearly established the essential role of these systems to realize active safety systems for accident prevention, reflecting also the innovative systems introduced by industry [31, 10]. These systems need to accurately assess situational criticalities and simultaneously assess awareness of these criticalities by the driver; it requires that the obstacles detection algorithms must be reliable and accurate [118], providing: a real-time output, a stable and robust representation of the environment and an estimation independent from lighting and weather conditions.
Initial systems relied on only one exteroceptive sensor (e.g. radar or laser for ACC and camera for LDW) in addition to proprioceptive sensors such as wheel speed and yaw rate sensors. But, current systems, such as ACC operating at the entire speed range or autonomous braking for collision avoidance, require the use of multiple sensors since individually they can not meet these requirements. It has led the community to move towards the use of a combination of them in order to exploit the benefits of each one. [114, 85].

(a)

(b)

(c)

(d)

Figure 1: Examples of fully autonomous ground vehicle: (a) BRAiVE- VisLab, University of Parma; (b) Bertha - Daimler; (c) KITTI - Karlsruhe Institute of Technology (KIT); (d) Google Car - Stanford Artificial Intelligence Laboratory (SAIL), Stanford University.

Pedestrians and vehicles detection are ones of the major thrusts in situational criticalities assessment, still remaining an active area of research [34, 49, 102, 29, 66, 18, 48, 43, 113, 82]. ADASs are the most prominent use case of pedestrians and vehicles detection. Vehicles should be equipped with sensing capabilities able to detect and act on objects in dangerous situations, where the driver would not be able to avoid a collision. A full ADAS or autonomous vehicle, with regard to pedestrians and vehicles, would not only include detection but also tracking, orientation, intent analysis, and collision prediction [107, 77].

The system detects obstacles using a probabilistic occupancy grid built from a multi-resolution disparity map. Obstacles classification is based on an AdaBoost SoftCascade trained on Aggregate Channel Features. A final stage of tracking and fusion guarantees stability and robustness to the result.

The remainder of the dissertation is structured as follows: the chapter 1 provides a broad overview of the state of the art. The system is described, in detail, in the chapter 2 with quantitative and qualitative results reported in the chapter 3. At the end, results, conclusion and future works are presented in the chapter 4.

Chapter 1

State of the art

This chapter provides an overview of the different approaches presents in the community, divided according to the system steps.

1.1 Monocular obstacle detection

Vehicles and pedestrians detection is a very hard challenge. It is complex to find an exhaustive type of feature for the following reasons:

- high variability of obstacles appearance due they change pose, wear different clothes, carry objects and have different size. Erroneous detections can be introduced by shadows, man-made structures, and ubiquitous visual clutter.
- outdoor urban scenarios have clutter background and different illumination and weather conditions, that allow to make only weak assumptions about the scene structure [116].
- obstacles can be occluded, partially or totally. Training classifiers with just full obstacles shapes gives better detection results but also more false positives;
- high dynamic scenes where both the obstacles and camera are in motion. Obstacles also appear at a different viewing angles.

- high performance in detection rate and speed. Complex features are better but, at the same time require more computational resources and then are slow. We need a trade-off between this constraints: speed is fundamental for real-time processing and, detection rate is fundamental for decreasing the number not detected.

It is need to train a classifier for detecting obstacles which can be different in: carried objects, size and clothes. The more general classifier will be, the more obstacles it will detect but, also, will be easier to get false positives such as a tree, poles and guard rails.
A further challenge is to collect an extensive database; a large amount of images that allows to train a classifier that can better interact with the described problems. It is more important, as shown in the next chapters, which kind of subjects are used to train the classifier. The detection can be broken down into the generation of initial object hypotheses (ROI selection) and verification (classification).

1.1.1 ROI selection

ROI selection is one of the main and most important components of detection algorithms. This operation consists of selecting regions of interest where obstacles are present. The impact of this processing step on the computational time is huge; selecting more ROI requires more processing time. The simpler approach is the sliding window technique, where a detection window is shifted at various scales and location over the image. To give a sense of the computational complexity, let us take for example an image with resolution 640x480 pixels. Disregarding the varying window sizes for a moment, if a constant window size of 128x64 is used over the entire image, with a one pixel shift, more than 200.000 detection windows need to be analyzed requiring huge computational resources. Improvements can be achieved by combining the sliding window method with a cascade classifier of increasing complexity. This kind of approach is used in [48], where a combination of haar features and cascade classifier is used to obtain a detection area; also a cascade classifier is applied to understand the image regions where obstacles are more frequently located. An alternative to the use

of a cascade classifier for reducing the search area is to take advantage of the camera calibration knowledge and a priori information on target objects. Other assumptions such as float world objects, common pedestrians geometry, object height, and aspect ratio helps to reduce the number of possible ROIs. Other techniques are derived from the image data and object motion. In surveillance system the background subtraction can be used to obtain the ROI from an image or, also, compute the deviation of the observed optical flow from the expected ego-motion flow field. The use of an interesting point detector would help to extract the regions with high information content based on local discontinuities of the image brightness function. All these techniques allow to reduce the detection window and speed-up the system, which is a key feature for obtaining real-time behavior.

1.1.2 Classification

Once the ROIs of an image are determined, the next step is to understand whether obstacles are present in the ROIs. This process is referred to as classification (or verification), and it involves utilizing pedestrian/vehicle appearance and features. A good starting point is the separation of these models into generative and discriminative models [38]; this approach allows to classify an image subregion as pedestrian/vehicle. Discriminative models differ from generative models in that they do not allow one to generate samples from the joint distribution of x and y, where x is an observed variable and y in an unobserved variable. However, for tasks such as classification and regression that do not require the joint distribution, discriminative models can yield superior performance.

1.1.3 Generative Models

In the generative approach, the empirical observation are explained by a model that describes probabilistically the interaction between the variable quantities. This probabilistic system is specified by two components:

- a list of variables that quantify the status observed and supposed model;
- a joint probability defined over all these variables.

Differences between the generative models are in the nature of these specifics. The a-posteriori probability can be obtained from the a-priori probability and joint probability from the Bayesian approach.

Shape models

Shape cues are very important to reduce variability in pedestrian appearance due to lighting conditions and clothing; there are basically two type of shape representations: discrete and continue.

- Discrete approaches uses representative shapes to simplify complex shapes. This kind of approach requires high specificity for example-based models and consequentially an higher number of example shapes to cover the space of shape models. In this case, trade-off between specificity and compactness in order to use it in the real world must be found.

- In the continuous shape model case, a compact parametric representation for the class conditional density and learning from a set of training shapes are used. Forcing topologically different shapes (such as pedestrian with feet apart and closed) can give many intermediate physically implausible model instantiations. To recover physically plausible regions in the linear model space, conditional density models have been proposed. Nonlinear extensions can also be used jointly with a larger number of training shapes; an alternative solution is breaking the non linear model into piecewise linear patches. Using the continuous model, gaps in the discrete model representation can be filled using interpolation. In previous years, a two-layer statistical field model has been proposed [130] to improve the performance of detectors in presence of occlusions and cluttered background by representing shapes as a distributed connected model. An hidden Markov layer for capturing shapes prior is combined with an observation layer, which associates shape with the likelihood of image observations.

Combined shape and texture models

To create more complex representations, shape and texture information can be combined within a compound of parametric appearance model. These approaches involve separate statistical models for shape and intensity light variations [37]. Model fitting requires joint estimation of shape and texture parameters using iterative error minimization schemes. To reduce the parameter estimation complexity, the relation between fitting errors and the associated model parameters can be learned from examples.

1.1.4 Discriminative models

The discriminative models, on the other hand, directly compare the problem of finding criterias that allow grouping of empirical observations. To do this, usually , it is extracted some intrinsic features such as thickness, shape factors, brightness, from the object and mapped each observation to a point in a multidimensional space. The relationship between similar points of the same object or differences between points of different objects is searched in a second step.

Features

In computer vision and image processing the concept of feature is used to denote a piece of information which is relevant for solving the computational task related to a certain application. More specifically, features can refer to:

- the result of a general neighborhood operation (feature extractor or feature detector) applied to the image;
- specific structures in the image itself, ranging from simple structures such as points or edges to more complex structures such as objects.

Other examples of features are related to motion in image sequences, to shapes defined in terms of curves or boundaries between different image regions, or to properties of such a region. The feature concept is very general and the choice of features

in a particular computer vision system may be highly dependent on the specific problem at hand. Local filters on pixel intensities are frequently used for feature extraction. One of the most popular features are the non adaptative Haar wavelet features proposed by Papageorgiou and Poggio [98] and then used by many others. Haar features popularity is due to their simplicity and fast evaluation using integral images. Viola and Jones[126] adapted the idea of using Haar wavelets and developed the so called Haar-like features. An Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, summing up the pixel intensities in each region and calculating the difference between these sums. This difference is then used to categorize subsections of an image. For example, let say we have an images database of human faces. It is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore a common haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relatively to a detection window that acts like a bounding box to the target object (the face in this case). Due to overlapping spatial shifts, we have many times redundant representations and we need to select the most appropriate features from a large set of them. Initially, using the geometric configuration of human body, this procedure was done manually [90]; but, are used now automatic procedures of features selection, such as variants of AdaBoost. The automatic extraction can be seen as an optimization for the classification task. We can include particular configuration of spatial features in the optimization, leaving that the features set fits to the underlying data set during training. This type of approach, has been shown to be more effective than the non adaptative Haar wavelets features with regards to pedestrian classification. Other type of features are based on discontinuities in the brightness function in the image in terms of models of local edge structure. The most popular are the Histogram of Oriented Gradients(HOG) descriptors [29], well-normalized image gradient orientation histograms, calculated over local image blocks. They are implemented in dense and sparse representation, where the last one must be preceded by an interest point detector to know the relevant part of the images. Initially, the dense HOG descriptors were computed only at a single fixed scale, to obtain a smaller feature vector

and better performance in terms of speed. But, afterwards, variable size block versions has been implemented with better results than the original HOG descriptor. The local shape filters that explicitly incorporate the spatial configuration of salient edge-like structures have already been investigated by other people: Mikolajczyk et al. [89] introduced multiscale features based on horizontal and vertical co-occurrence groups of dominant gradient orientation. Also sets of edgelets, representing local line or curve segments, have been proposed. An extension has been recently introduced about adapting the local edgelet features to the underlying image data [106]; using the Adaptive Boosting these features are assembled from low-level oriented gradient responses to give us more discriminative local features. This is a very new approach because usually Adaboost is used to select the most discriminative subset of features. Also the outlines, with the extension to spatio-temporal features, have been used to capture the human motion, especially gait. Haar wavelets and local shape filters have been extended to the temporal domain by incorporating intensity differences over time or, also, HOGs have been extended to histograms of differential optical flow [28]. There are different papers comparing the performances of several techniques.

Classifier architectures

The goal of discriminative classification is to found an optimal decision boundary between pattern classes in a features space. Feed-forward multilayer neural networks [61] implement linear discriminant functions in the feature space in which input patterns have been mapped nonlinearly. The optimal boundary is reached minimizing an error criterion with respect to the network parameters, i.e., mean squared error. In the out context, feed-forward multilayer neural networks have been applied particularly in conjunction with adaptive local receptive field features as nonlinearities in the hidden network layer. Support Vector Machines(SVMs)[25] has become a powerful tool to solve pattern classification problem. At opposite of Neural Networks, SVMs do not minimize some error criteria but maximize the margin of a linear decision boundary (hyperplane) to achieve maximum separation between the object classes. In pedestrian/vehicle detection, linear SVM was combined with different type of features [28] [29]. Using non linear SVMs with polynomial or radial kernel showed to

improve considerably the performance but with a significant increase in computational cost and memory requirements. AdaBoost was used as both automatic features selection procedure and as constructor of strong classifiers as weighted linear combinations of the selected weak classifiers, each involving a threshold on a single feature. Viola et al. [126] adopted the boosted cascade detectors to incorporate nonlinearities and speed up the classification process. The main motivation for this type of approach is that the majority of the detection windows in an image are non-pedestrians, so a method is needed to keep those containing pedestrian and remove those containing non-pedestrian in the shortest possible time. AdaBoost performs in each layer using the error of the precedent layer to improve the performance and create a more complex detector. This increase the processing speed requiring just few features in the early cascade layer level to reject non-pedestrians.

1.2 Stereo obstacle detection

Unlike the monocular detection, stereo vision approaches relies mainly on motion-based than appearance-based approaches. The disparity map allows to obtain a 3-D world reconstruction, providing the understanding of scene, motion characteristics, and physical measurements. The 3-D information provide the ability to track points and distinguish moving from static objects, shifting the focus from appearance features and machine learning of monocular detection to motion features, tracking and filtering. Most of the system place stereo cameras looking forward out the front windshield to detect vehicles ahead of the ego vehicle; [51] is an example of cross traffic detection with stereo cameras looking sideways. Stereo matching, appearance-based approaches, and motion-based approaches to object detection using stereo vision are described below.

- **Stereo Matching**: the stereo cameras rectification transforms the epipolar lines into horizontal scan lines in the respective image planes. This allow to speed-up the searching of correspondences between two images, reducing the search area to the horizontal direction. The output of the stereo matching is a disparity map [84]. Dense matching techniques have been of great interest in the

intelligent vehicles community [41], allowing significant improvements of on-road scene interpretation. If correlation-based stereo has been widely studied and optimized [128], new approaches are actively searched in the computed vision communities. Mainly, it is possible to observe a transition from local correlation-based approaches [128] toward semiglobal matching [57, 46, 53], with denser maps and lower errors.

- **Compact Representations**: the community offers several approach of compact representations of measured data, including occupancy grids [100], elevation maps [96], free space understanding [6], ground surface modeling [68], and dynamic stixels [39]. This representation permits to facilitate segmentation of the scene [68], identify obstacles [123], and reduce computational load. In the following subsections, a more detailed explanation is provided, dividing them between appearance-based and motion-based methods.

- **Appearance-Based Approaches**: unlike as in monocular vision, these approaches are less common in stereo vision. However some motion-based approaches relies on some appearance-based stereo-vision techniques for initial scene segmentation. The most common technique to model the ground surface is the v-disparity [68]. The v-disparity forms a histogram of disparity values for pixel locations with the same v, i.e., vertical image coordinate. Starting with an $n \cdot m$ disparity map, the result is an image consisting of n stacked histograms of disparity for the image. Through the Hough transform [36] or the RANdom SAmple Consensus (RANSAC) [40], it is possible to model the disparity as a function of the v coordinate of the disparity map and classify the pixel locations as ground surface point if they fit this model [68]. In order to find the free space from disparity maps, instead, it has been used the u-disparity [99], forming also an histogram of stacked disparities, for pixel locations sharing the same u coordinate. This is used to infer directly the free space, instead of fitting a road model. Free space detection is widely addressed in the stereo-vision literature, mainly for scene segmentation and highlighting of potential obstacles. Dynamic programming is used in [6, 67] computing the free space directly

from the disparity and depth maps. In order to model the corresponding errors introduced to stereo matching and 3-D localization of tracked interest points, in [17], convolutional noise and image degradation are added to stereo image pairs. Monocular appearance features, as color [19] and image intensity [22], are less likely used for object detection using a stereo pairs compared to the disparity and depth ones. [22] size, width, height and image intensity are used as features and combined in a Bayesian model to detect object. [65], potential object are extracted from an histogram of depths, computed from stereo matching. Detection directly on the monocular images, include also Delaunay triangulation [64]. Clustering based on depth map and Euclidean distance to cluster point into object are common in various studies [8, 122]. A modified version of iterative closest point, with polar coordinates, has been used in [9]. The algorithm is able to detect objects and establish their pose respect to the ego vehicle. A combination of mean shift algorithm and clustering is used in [55, 72].

- **Motion-Based Approaches**: using motion for stereo-based object detection is a key point; the optical flow is the start for many stereo vision analysis of the on-road scene [83]. Several algorithms track interest points in the monocular image plan of one stereo cameras and localize the points using the disparity and depth maps [103]. Also [30, 69, 65, 64, 55, 103, 73, 78, 91, 16] use the optical flow as key component of their algorithm. [20], use a 3-D version of optical flow, based on least squares to solve the 3-D points' motion problem. The optical flow has several modification and uses in the community. A comparison of Lucas-Kanade optical flow and block-based coarse-to-fine one in [91] shows the more robustness to drifting of the second one. Differentiation between intersection of arterial road is performed in [45], by modeling the aggregate flow of the scene over time. In [73], instead, scene flow is used to detect candidate vehicles, modelling the motion of background and regions whose motion differs from the scene; geometric constraints are also included to improve the detection. In [64, 73], the tracking of feature points from optical flow is used to estimate the ground plane. The total least squares is used in [70]

to fit the ground plane model. In [65, 73], the ground plane fitting is based on RANSAC. A quadratic surface is estimated as ground in [96], and it is used as input for obstacle detection using digital elevation maps [123]. [125] improve this work using a radial scanning of the digital elevation map and detect static and dynamic objects after inserted in a Kalman filtering process. The tracking of interest points is also used in structure from motion techniques for scene reconstruction and understanding. The Longuet-Higgins equations are used for scene understanding in [13, 64, 97]. In [64], tracked interest points are used to estimate ego motion. In [8], ego-motion estimation is performed by tracking SURF interest points. The concept of stixels is introduced in [39]: tracked 3-D points, through 6D vision, grouped into an intermediate representation consisting of vertical columns of constant disparity. Starting with free space computation, stixels are formed assuming that structures of near-constant disparity stand upon the ground plane. This representation significantly reduces the computational time over tracking all the points individually. Classification is performed using probabilistic reasoning and fitting to a cuboid geometric model. Looking at stereo-vision literature, in scene segmentation and understanding, occupancy grid are widely used. The free space is extracted using the dynamic programming [6] on the occupancy grid, populated by the 3-D tracked points, static and moving. Also [71] use the dynamic programming to compute the free space and build the occupancy grid. Comparison of cartesian coordinates, column disparity, and polar coordinates is presented in a stochastic occupancy framework. The u-disparity representation of [99] is equivalent to the representation of [6]. Scene tracking through recursive Bayesian filtering is used in [99, 100] to populate the occupancy grid and object detection is performed via clustering. The state of the occupancy grid, in [70], is obtained through the sequential probability ratio test, a recursive estimation technique. A temporal and spatial filter on occupancy grid is applied in [100]. Polar coordinates and depth-adaptive dimensions are used to set up the occupancy grid in [97], to model the field of view and depth resolution of the stereo pair. Motion cues are used in [30], with cells represented as particles, the occupancy

defined by their particles probabilities and cells velocities estimated for object segmentation and detection.

1.3 Tracking

Stable and robust system needs a tracking stage, to re-identify and measure dynamics and motion characteristics and predict and estimate the upcoming position of obstacles on the road. Measurement and sensor uncertainty, data association, and track management are common issues of objects tracking. Below will be described the monocular and stereo-vision tracking approaches. Even if there are common estimation and filtering methods, depending on available measurements, the estimation parameters differ: often, monocular tracking are based on measurement and estimation in term of pixels, whereas stereo-vision methods estimate dynamics in meters. Combined approach and fusion with other sensing modalities will be also described.

1.3.1 Monocular Obstacles Tracking

Monocular tracking is typically based on image plane. Tracking using monocular vision serves two major purposes:

- facilitate estimation of motion and prediction of obstacles position in the image plane;
- help the temporal stability: filter spurious false positives [111] and maintain awareness of previously detected obstacles that were not detected in a given frame [54].

Measure the motion and predict the position of obstacles in pixel position and velocity is the goal of monocular tracking. The pixels observation space, leads to uniquely vision-based tracking methods, based on the objects appearance in the image plane. Template matching is an example of uniquely vision-based tracking. Object are detected using Haar wavelet coefficients and SVM classification in [80];

frame to frame tracking is performed taking a measurement of the similarity in appearance. Cross-correlation scores is often used in appearance-based tracking. A further step in the tracking process is represented by feature-based tracking [133]. Haar-like features combined with AdaBoost cascade classifier is used in [54], where the tracking is performed through a Kalman filter in the image plane. In order to have a measurement also in case of detector failure, it is exploited a local search over the image patch for similar feature score. In [134] the optical flow is used to track obstacles by directly measuring the new position and the displacement of interest points. Bayesian filter has been largely used in the monocular tracking literature. Typically, the state is formed by pixel coordinates of obstacle bounding box and the interframe pixel velocities [21]. In [22], [27], and [33], Kalman filtering was used to estimate the motion of detected vehicles in the image plane. In [23, 3, 5], Kalman filtering was used to estimate the motion of detected vehicles in the image plane. [21, 60, 111, 117, 92, 88] describe the use of particle filtering for monocular tracking. Several studies attempt to estimate longitudinal distance and 3-D information from monocular vision. Typically, it is assumed ground flat [27, 56] or it is used the interest point detection and a robust model fitting step to estimated its parameters [40]. In [94] the estimation of 3-D coordinates from monocular vision is made using a set of constraints and assumption and tracked through a Kalman filter. In [58], ground plane estimation is used to extract the 3-D information. Tracking is based on the interacting multiple models, each one consisting of a Kalman filter. Monocular vision is used in [132, 131] to estimate the ego motion and moving object were tracked using a 3-D Kalman filter. Even if the systems have estimated 3-D obstacle position and velocity with monocular data, a ground truth reference 3-D measurements, from radar, lidar, or stereo vision has been used to compare.

1.3.2 Stereo-Vision Obstacles Tracking

Stereo vision obstacles tracking deal the measurement and estimation of position and velocity, in meters, of detected obstacles on the road. The state vector is generally represented by obstacle's lateral and longitudinal position, width and height, and velocity. Assuming linear motion and Gaussian noise [41], Kalman filter is considered

optimal and often used for the estimation process. But, analyzing the problem, if the vehicle motion is represented by also its turning behavior using the vehicle's yaw rate, it is nonlinear. [12] show the use of the extended Kalman filter (EKF) for estimate the nonlinear parameters with their linearization. Alternative to EKF is the use of the Particle Filter, with sample importance re-sampling, to deal both with linear and nonlinear motion parameters [20]. Stereo-vision tracking based on Kalman filter and disparity filtering have been widely used [78]. Stereo matching noise is generally modeled as white Gaussian noise [17, 103] and cleaner disparity maps can be produced with time filtering [41]. [6] and [103] use the Kalman filter to track individual 3-D points. In [39] kalman filtering is performed to track stixels (vertical elements of near-constant depth). [112] and [65] combine monocular detection, based on AdaBoost classifier, and stereo information to track in 3-D using Kalman filtering. Vehicles's position and velocities are estimated in [16] using Kalman filter. [13] added also the vehicles' yaw rate to the Kalman filter state in addition to the position and velocity. Also in stereo-vision obstacles tracking, the EKF has been widely used, generally to take account of the nonlinear motion and observation model. In [13] and [11], the yaw rate and corresponding turning behavior is estimated using the EKF. Due to the camera positioning (side-mounted stereo rig), the motion of tracked obstacles with respect to the camera's frame of reference, is particularly nonlinear. In [69], the nonlinear transformation from objects' 3-D position into stereo image and disparity is modeled with the Extended Kalman filtering. The extended Kalman filtering was used to estimate the ego motion, with independently moving objects position and motion estimated using Kalman filtering in [64]. It has been widely used also the particle filter in stereo objects tracking. It is an alternative to the EKF for the estimation of nonlinear parameters; it is used a likelihood function to weight the filter's multiple hypotheses. [20] is an example of use of particle filter for estimate objects 3-D position and yaw rate. In [55], the particle filter is used to estimate the the motion of tracked obstacles, mapping the motion to full trajectories, which were learned from prior observational data. In [30], the particles has been used for multiple purpose: model the on-road environment as occupancy cells and represent the tracking states for the detected obstacles. Interacting multiple models have been used in

tracking to estimate the motion of an obstacles given different motion modes. [13] use four different predominating modes to describe the obstacles motion at intersection, in terms of their velocity and yaw rate characteristics. The goal is to identify whether the velocity is constant or accelerated and whether the yaw rate is constant or accelerated. Using the error covariance of each estimator it is possible for determined the model fit. A state transition probability is used to switch between competing modes after the model fit is determined. The interacting multiple models are becoming interesting, due to the more precise measurement and due to the best estimation of all the motion parameters that cannot be well estimated by a single linear or linearized filter [51].

1.4 Data Fusion

1.4.1 Vision sensors

Previously, they have been explained works based on stereo vision for obstacles detection and works based on monocular vision, mainly relying on machine learning algorithm. Actually, several systems exploit the benefits of both to on-road obstacles detection and tracking. Typically, the fusion of monocular and stereo vision, consists in using monocular vision for detection and stereo vision for 3-D localization and tracking. [115] shows as stereo-vision can merge objects if they lie close together in 3-D space (typically a group of pedestrians), whereas monocular vision can correctly detect them. Detection on monocular plane and localization based on stereo vision it has been used to address the problem. In [120], objects candidate regions are generate using symmetry on monocular image and vertical objects has been searched, in the 3-D domain, to verify those regions as obstacles. [112] use a monocular vehicle detector [111] to track objects in the image plane and stereo-vision and Kalman filtering to track them in the 3-D plane. [110] use the clustering technique presented in [111] to learn the typical vehicle behavior on highways. [76] use an AdaBoost classifier to detect object in the image plane. After, the ground surface is estimated using the v-disparity algorithm, and the tracking in the 3-D domain has been imple-

mented using an extended Kalman filtering. False alarms reduction and performance improving has been reached through a specific techniques of track management. A set of Adaboost classifiers, trained for multiple vehicle views, are used in [65] to extract vehicles' candidate regions. Peak in the disparity map are searched in order to verify the candidate regions. 3-D localization and ground plane estimation is based on stereo-vision.

1.4.2 Multi-sensors

In the last ten years, due to the lower cost, the intelligent vehicles have seen the increasing availability of a variety of sensors that pioneered for the driver assistance system and the autonomous driving. Sense, perceive and respond to the on-road environment in a safe and efficient manner requires that a full autonomous vehicle must be equipped with an advanced sensor suite, which must cover a variety of sensing modalities. A complete sensor suites, present in many leading autonomous vehicles, includes cameras, lidars, and radar sensing arrays [74]. A common way of sensor-fusion studies, is just an extension of vision-only based algorithms across multi-sensors, in order to reduce uncertainty, cover blind spots, or perform ranging with monocular cameras. In recent years, radar-vision fusion received a lot of attention for on-road vehicle detection and perception [81]. It is an interesting pair since the crude lateral resolution of radar can be balanced by monocular vision that, on the other hand, lacks in longitudinal range, that is the strength of the radar. In this way the weakness of each sensor is filled by the other [24, 93]. A probabilistic estimation of obstacles positions, fusing radar and vision sensors information, is performed in [108, 104]. The estimation uncertainty is then propagate into decision making, for lane change recommendations on the highway. In [44], overtaking vehicles on the highway are detected combining vision and radar; the optical flow has been used to detect vehicles entering the camera's field of view. In [2] radar and vision are combined with radar detecting side guardrails and vision detecting vehicle using symmetry cues. Extrinsic calibration between radar and camera sensor is covered by several studies. [15] detect obstacles using vision algorithms on the bird eye view image and distance is calculated using radar. Ranging with radar and detection with a boosted

classifier using Haar and Gabor features is performed in [63]. A common global occupancy grid, with projected camera and radar detections, is used in [24]; objects are tracked using Kalman filtering in a global frame of reference. [42] detects potential vehicles using saliency operations on the inverse perspective mapped image and combined with radar. In [79] a combination of optical flow, edge information, and symmetry are used to detect obstacles, then ranged with radar; tracking is based on interactive multiple models with Kalman filtering. [1] use symmetry to detect vehicles, with radar ranging. In [119] obstacles are detected using HOG features and SVM classifier and ranged using radar. In [127], structure from motion is performed using monocular vision and objects and ground surface probabilities are estimated with radar. A radar-vision online learning framework has been developed in [62], and used for vehicle detection. [109, 129] combine stereo vision and radar for obstacles detection. Also the fusion between lidar and monocular vision has been subject of study in recent years. Extrinsic calibration between lidar and camera sensors is performed by several studies, detecting vehicles using monocular vision and lidar for longitudinal ranging. A combination of Haar-like features on monocular vision to detect vehicles and lidar to range is performed in [59]. Similar system is also used in [87, 101]. A Bayesian framework is used to fuse lidar with saliency vision cue in [86]. Stereo vision and lidar fusion is performed in [105, 7, 4, 52].

Chapter 2

System Overview

As mentioned in the previous chapter, obstacles detection is a field that attracts much attention from the research community. Even when narrowed to applications in connection with cars and ADAS, a large body of work exist. ADAS is a challenging domain to work within. Braking system take a short while to be apply, and reaction times must be fast for driving, where fraction of second can be the deciding factor between a collision and a near-miss. At the same time, the system must be robust, so the braking system is not deployed mistakenly (due to a false positives detection), which could itself lead to accidents, or worse, not employ at all (due to a missed detection). In this chapter will be described each part of the system, illustrated in Fig.2.1:

- the obstacles detection and classification stage;
- the tracking and fusion stage.

A detailed description of each system modules is provided in Fig.2.2.

2.1 Obstacles detection and classification

Stereo-vision based obstacles detection is a wide and complex topic, specially referred to automotive applications. In this section, will be described the obstacles detection algorithm starting from a disparity map; as shown in Fig.2.3 and Fig.2.4 in

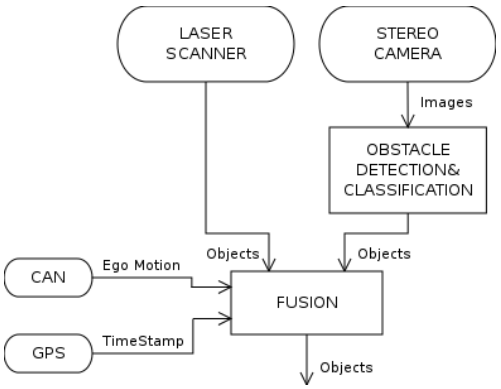


Figure 2.1: High-level schematic of the system.

more detail. A dense disparity map is built using a a pair of dedistorted and rectified images, as described in [57]. Images of different resolutions are used in this phase.

Input images have a resolution of $1280x960$ and, the disparity of the entire image is computed using a downsample of $4x$, while the disparity of the interest zone, as show in Fig.2.5, is computed using a downsample of $2x$. In this way it is possible to reduce the computational time withouth, at the sime time, lose accuracy in the disparity map. According the camera calibration parameters, a 3-D point cloud is derived respect to world reference system, shown in Fig.2.6. The 3-D world points are collected in a $2.5-D$ occupancy grid, called Digital Elevation Map (DEM), with user-defined dimensions. The density of each cell is represented by the number of contained points. In order to discriminate between obstacle and road cell, it is exploited the concept that a cell containing a vertical obstacle has a higher density respect a cell containing an horizontal surface. In this way, cells are classified as obstacle or road only evaluting their density. A more detailed description of the DEM can be found in [96]. This phase permits to assign a classification to each point with a valid disparity and discretize them in a grid. But, a discretization process, if on one hand allows to reduce the problem complexity, on the other hand introduce an error factor depending on the discretization.

Considering the couple of stereo images of Fig.2.7, in Fig.2.8 is shown the results

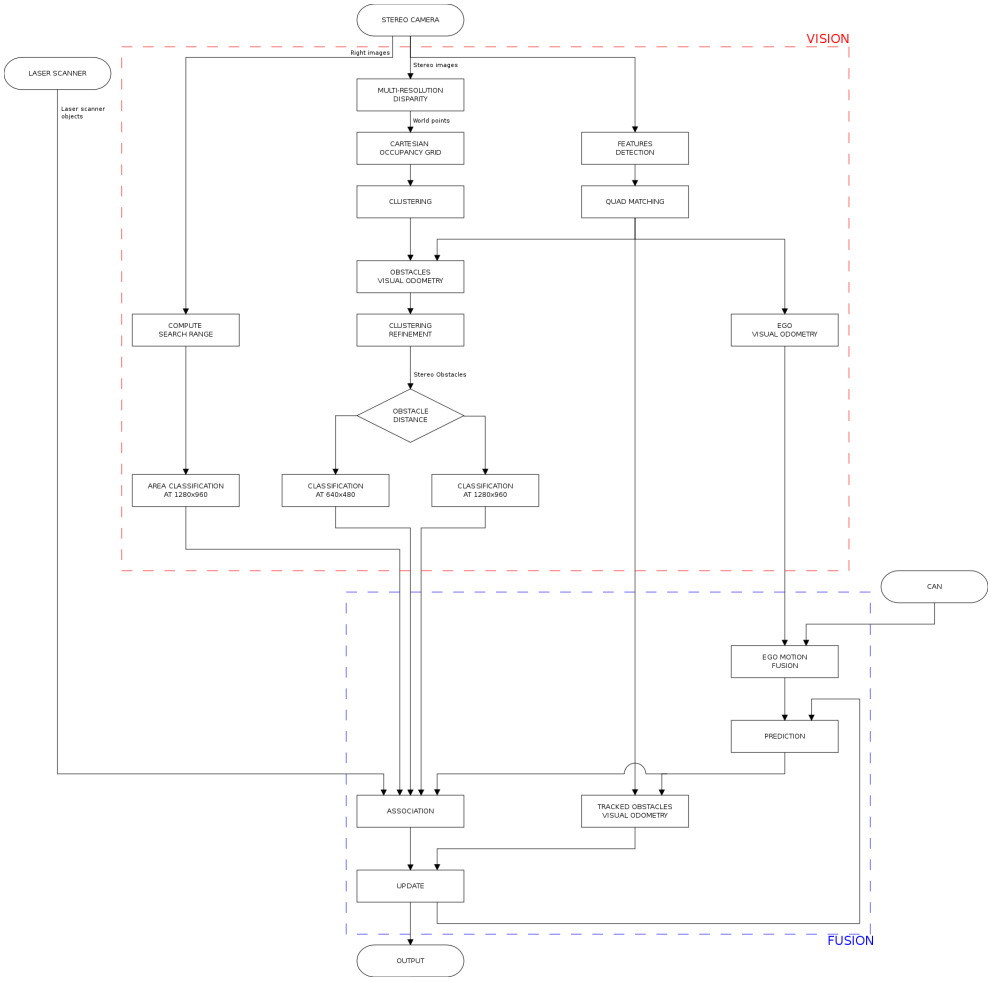


Figure 2.2: General schematic of the complete system.

of the density based classification process, where are reported only the points classified as obstacles. It is possible to see as not all the obstacles framed by the camera are reported in the grid. The grid dimension and position are used to define a region of interest where to detect the obstacles. The informations shown in Fig.2.8 are still not sufficient: it is provided the classification of each cell but it is not clear how many and which obstacles are present in the scene. It is required a further step aimed to

Figure 2.3: General schematic of the obstacles detection phase.

group grid cells belongig to the same obstacle.

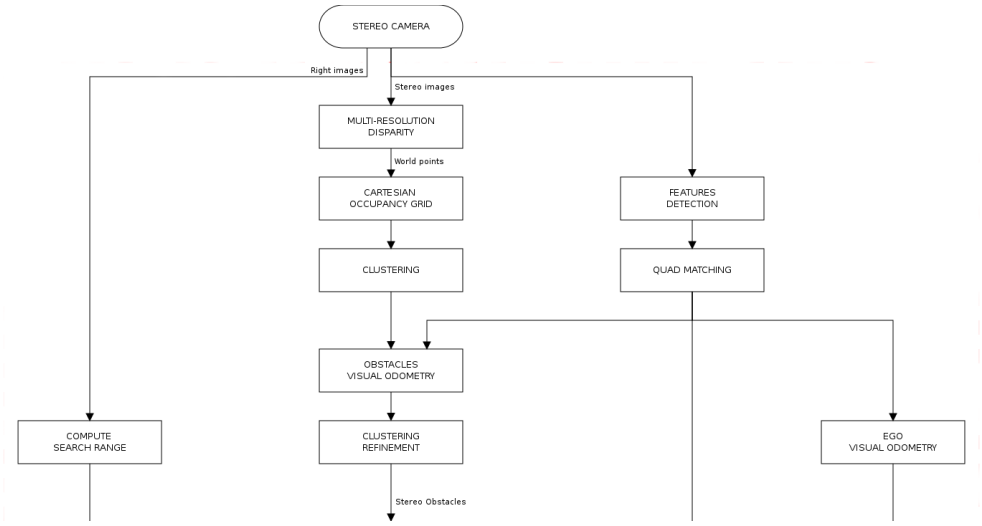


Figure 2.4: General schematic of the obstacles detection phase.

This operation is called *clustering* or *segmentation* and, according to the data distribution and their origin, can be carry out in several way. In this case, the clustering must be performed on 3-D point coming from a disparity map and grouped in a cartesian grid. The disparity computation results in an error of about 0.2 pixel, increasing with the distance, according to the formula 2.1, used to calculate the pixel depth knowing its disparity:

$$z = \frac{B * f}{d} \tag{2.1}$$

where B represent the *baseline* and f the pixel focal length of the stereo rig. The depth is represented by the z instead of x since this formula is used to obtain the

depth of a pixel in the *camera coordinate system*, rigid to the stereo rig. This reference system differ to the *world reference system* for the origin position and different orientation, as shown in Fig.2.6. The other two coordinates are computed using the following equations:

$$x = \frac{B * (u - u_0)}{d} \quad y = \frac{B * (v - v_0)}{d} \tag{2.2}$$

A rototranslation, according to the camera calibration parameters, permits to transform the camera coordinate in world coordinate. Defining with $m_k = (u, v, t)^T$ a generic point in the image plane, it is possible to consider this point as the projection of the world point $p_k = (x, y, z)^T$ on the image plane, so that $m_k = P(p_k)$.

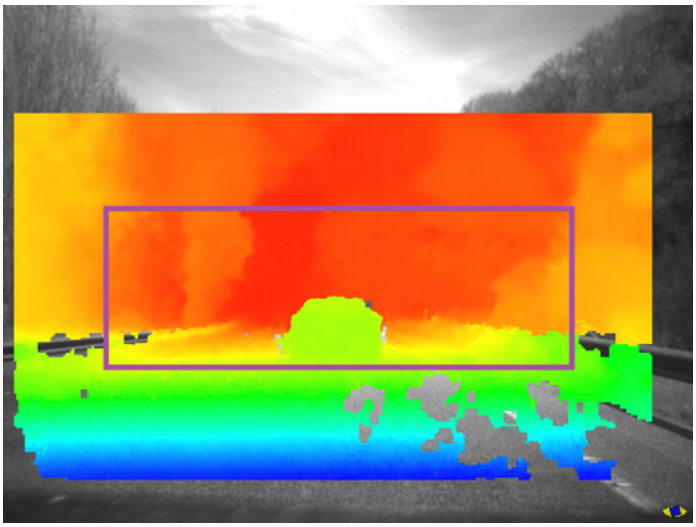


Figure 2.5: Example of multi-resolution disparity.

The measurement error of the sensor is reflected in the erroneous projection of the points in the grid. Since the error increases with the distance, farthest obstacles are more fragmented and difficult to cluster using only the concept of neighborhood. It has required to consider the error during the grid filling, using a process called *stochastic propagation*.

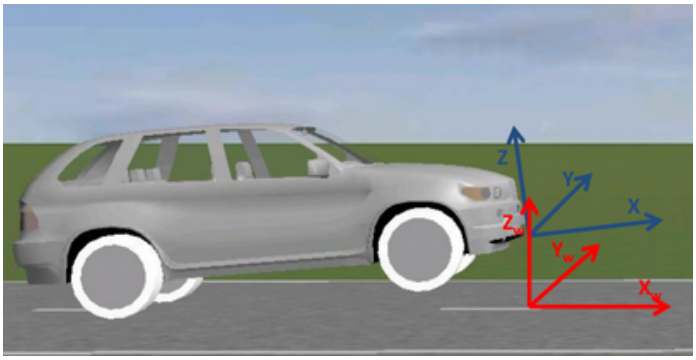


Figure 2.6: Reference systems. X_n, Y_n, Z_n: world reference system. X, Y, Z: vehicle reference system.

2.1.1 Stochastic Propagation

Considering a pixel with coordinate (u, v) with a disparity value of d. The error depends on the image resolution and on the used algorithm. In this case, we can set the error to 0.2 pixel.

$$\sigma_d = 0.2 \tag{2.3}$$

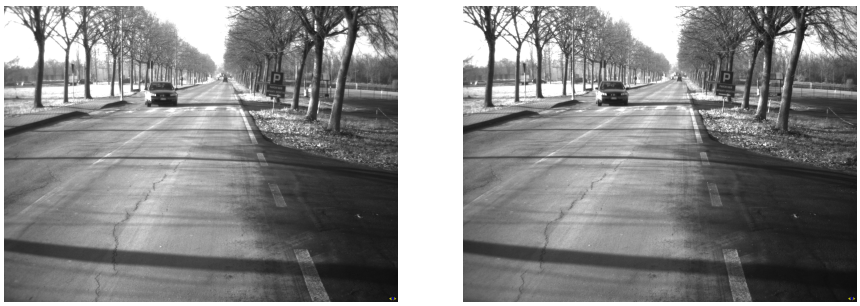
(x, y, z) is the 3-D world point in camera coordinate of the image point (u, v, d). According to [124], it is possible to approximate the longitudinal and lateral error as:

$$\sigma_z = \frac{z^2 \cdot \sigma_d}{B} \quad \sigma_x = \frac{\sigma_z \cdot x}{z} \tag{2.4}$$

The error described in 2.4 is expressed relative to the camera coordinate system. A transformation is required to be expressed relative to the world coordinate system. Considering the transfomation

$$\tilde{X} = T \cdot X \tag{2.5}$$

where $\tilde{X}$ is a point in the world coordinate obtained from the point X in camera coordinate. T is the rototraslation that convert from the camera coordinate system to the world coordinate system, expressed as $T = [R|t]$, with R is the rotation and t is



(a) Left image (b) Right image

Figure 2.7: Example input images.

the traslation. The uncertainty is propage through the Jacobian of the transformation T:

$$\Sigma_{\tilde{X}} = J \cdot \Sigma_X \cdot J^T \tag{2.6}$$

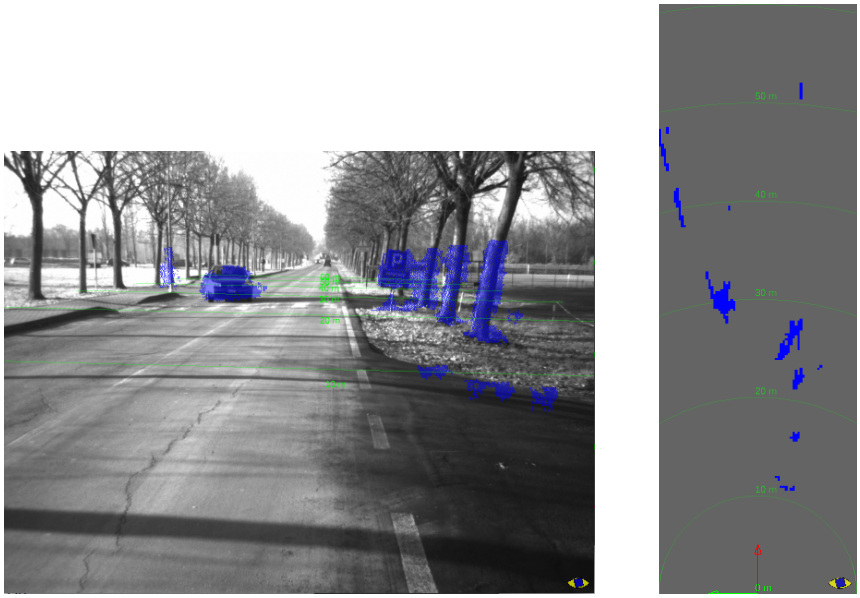
The Jacobian of a rototraslation is formed by the only rotation, so the transformation can be expressed as:

$$\Sigma_{\tilde{X}} = R \cdot \Sigma_X \cdot R^T \tag{2.7}$$

In this way it is possible to assign an uncertainty to each measured point. As described in [6], it is possible to define a density probability function to each 3-D point to be inserted in the grid. The function is a gaussian with mean the 3-D point and standard deviation described by 2.10. The p.d.f. of a multivariate normal distribution of order n can be used to described the probability to obtain an error ε_k given the real state $\overline{m_k}$:

$$G_{\overline{m_k}}(\varepsilon_k) = \frac{1}{(2\pi)^{\frac{n}{2}}|\overline{\Sigma_k}|} \exp(-\frac{1}{2}\varepsilon_k^T \overline{\Sigma_k}^{-1} \varepsilon_k) \tag{2.8}$$

Defined the p.d.f. for a single point, it is needed to define a function $L_{i,j}(p_k)$ which define the probability density of the cell (i, j), result of measurement m_k. In



(a) Image classification

(b) Topview classification

Figure 2.8: Density based classification on example images

this way is then possible to define the density probability of the entire occupancy grid as:

$$D(i,j) = \sum_{k=1}^{N} L_{i,j}(m_k) \tag{2.9}$$

Using the projection function defined before, the probability that in the cell (i,j) is present ad obstacle given a measurement m_k is:

$$L_{i,j}(m_k) = G_{m_k}(P(p_{i,j} - m_k)) \tag{2.10}$$

The formula shows as each point in the grid will affect all the other grid cells and not only his own. The influence is maximum in the cell corrispondent to the measurement and decrease with the distance. In order to guarantee a real-time processing, the

propagation has been confined to a limited neighborhood. An example of the density after the stochastic propagation process is shown in Fig. 2.9

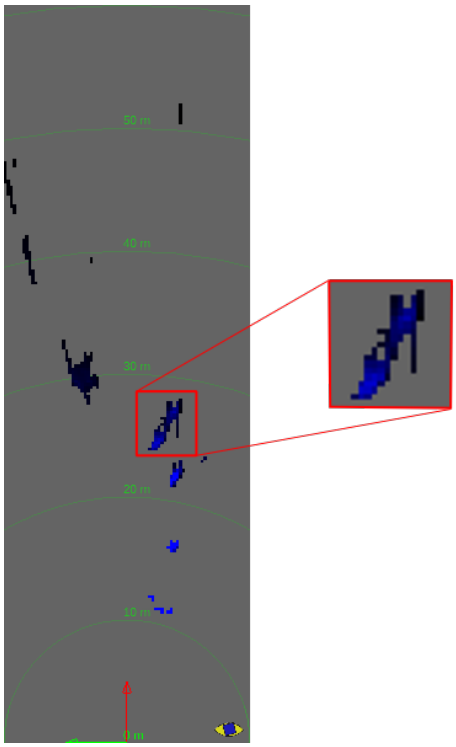


Figure 2.9: Example of density after stochastic propation.

The intensity of blu is directly proportional to the probability that in the cell is present an obstacle. Focusing on the image detail, it is possible to see as two close obstacles (a tree and a road sign) are distinguishable after the stochastic propagation process, unlike the density based classificaiton only, shown in 2.8.

2.1.2 Obstacles clustering

Two different method have been developed to extract the obstacles from the occupancy grid. The first approach is based on labeling of connected components and do not relies on the stochastic propagation phase. The second one, instead, starts from

the local maxima of the density probability function obtained in the stochastic propagation phase.

Floodfill labeling

Given an occupancy grid where each cell is classified as obstacle or not, it is possible to extract clusters exploiting a common method used in computer vision to segment binary images, *the floodfill labeling*. The algorithm looks for an interesting data (a white or black pixel in a binary image, for example), and, if it is found, recursively analyze its neighborhood until no more interesting point are found. Each point detected during the recursive search is labeled and separeted from the other. Considering as binary input the grid of Fig. 2.8. The peculiarity of the algorithm resides in the definition of the *neighborhood*, that depends on the cell position respect to the grid. The different type of neighborhood are divided according to a polar logic, as shown in Fig. 2.10. However they have the same number of cells, eight. The algorithm looks for an obstacle cell in the grid. Starting from this cell, the floodfill labeling expands depeding on the cell position. Each classified cell is inserted in the *border* and will be removed if they are the starting point for a new expansion. The algorithm terminates if there are no more cells classified as obstacle.

A labeling example is shown in Fig.2.11. In this case, the road sign and the tree belong, wrongly, to the same obstacle. The probabilistic clustering has been introduced to overcome this problem.

Probabilisting labeling

The stochastic propagation allows to describe each cell in more details, as the probability that each cell contains an obstacle. This information is exploited to improve the floodfill labeling. As shown in Fig.2.8, is possible to look at the probability density function of each cell as a tridimensional function. Each maxima of the PDF represents an obstacle. The Non-Maxima Suppression algorithm allows to find the maxima of the function and, therefore, the obstacles. The only parameters to be setted is the minimum distance between two maxima, an example in shown in Fig.2.12.

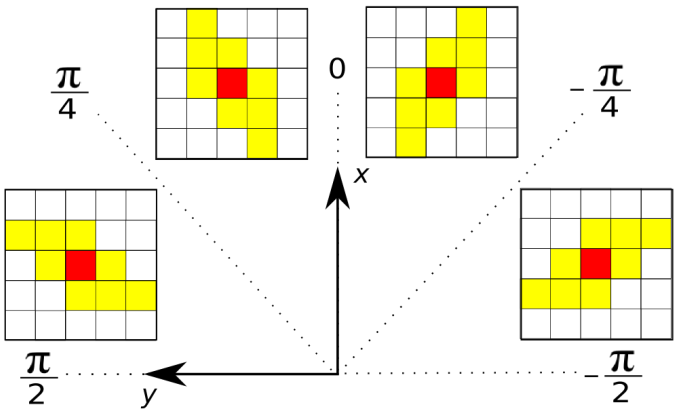
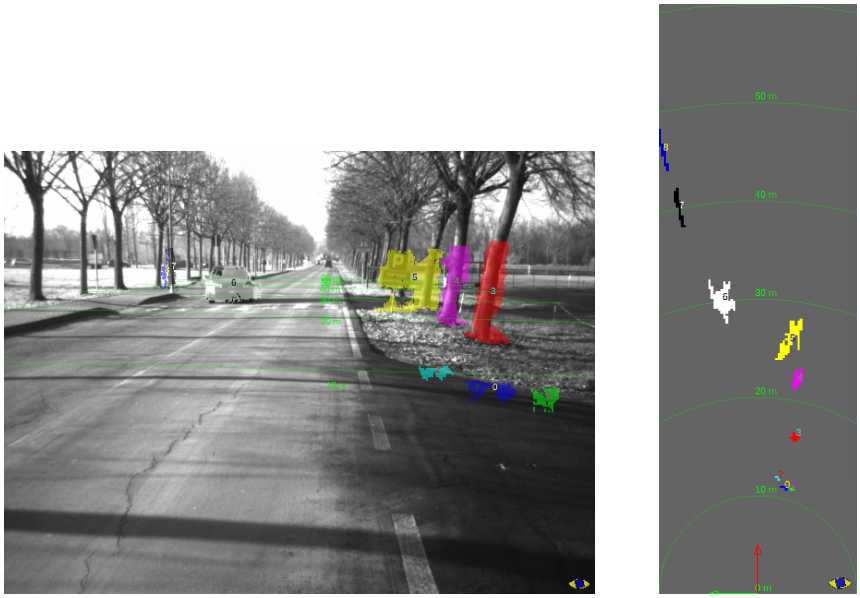


Figure 2.10: Neighborhood definition.

In this case, two maxima are found in the cell occupied by the road sign and the tree. After the maxima have been found, to assign the cells to the obstacles, it has been used a function that consider both the spatial distance and the relative density probability. The algorithm starts from a local maxima of the PDF, to which it is assigned an obstacle. Its neighborhood is analyzed, moving to each direction with lower density probability of the starting one. This condition is always true in the proximity of the maxima, according to its definition. Then, the algorithm continues from each cell inserted in the *border*, where is no longer guaranteed that they will have a lower density probability of the previous one. Each cell with lower probability is added to the *border* and the algorithm terminate when the *border* contains no more cell to be expanded.

The spatial distance constraints is respected due to the type of expansion used, as shown in Fig.2.13. The lower density constraint, instead, allows to improve the segmentation process since it better reflects the obstacle definition during the grid creation. A segmentation example can be seen in Fig.2.14. This technique well performs in the detection of small obstacles, since they are represented by only one maxima in the PDF. The performance decrease with medium and big obstacles, represented



(a) Image classification

(b) Topview classification

Figure 2.11: Floodfill labeling example.

by several maxima in the probability density function. A further stage is needed to merge obstacles wrongly separeted. In this case, close obstacles with same motion, are merged. The obstacles motion is calculated using the sparse optical flow.

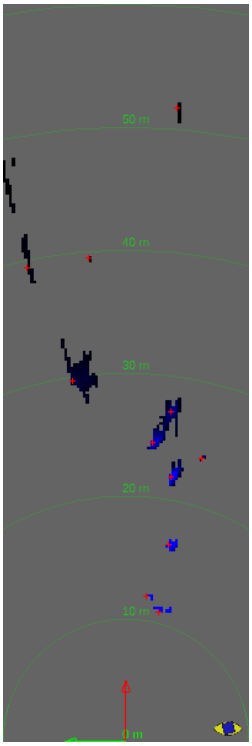


Figure 2.12: Maxima detection example.

2.1.3 Motion estimation

The object pose in the space is defined by six parameters, degrees of freedom, which represent rotation and traslation respect to a defined reference system.

$$x = \begin{bmatrix} \theta \\ \phi \\ \psi \\ t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.11}$$

The object motion is represented by the first order derivative, respect to the time,

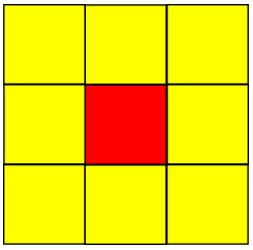


Figure 2.13: Neighborhood definition for stochastic clustering.

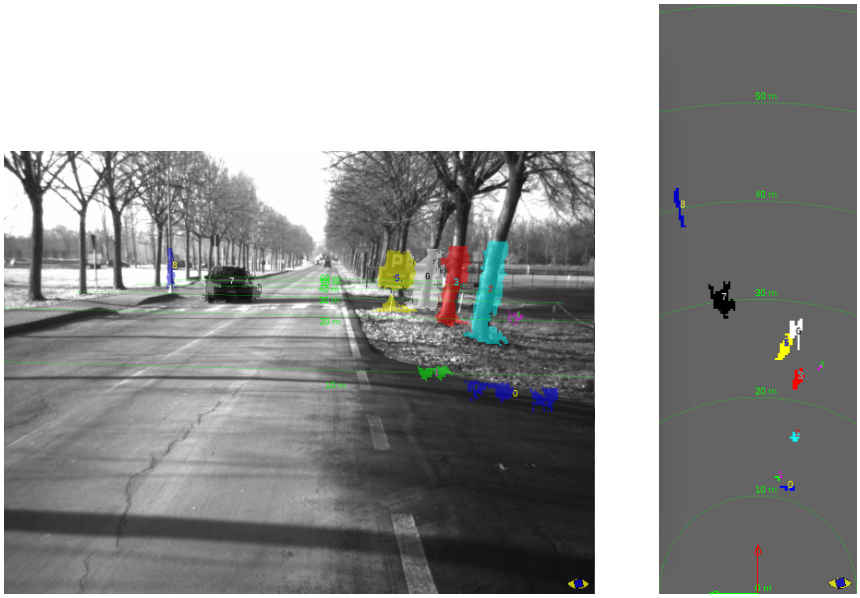
of the equation 2.11. The derivative is approximate with:

$$\dot{x} \approx \triangle x = x_t - x_{t-1} \tag{2.12}$$

The motion is estimated using the movement of the single image pixels. Two phase are required: the ego-motion estimation and then the object motion estimation. The ego-motion estimation is required since using the pixel movement between two frames it is possible only to obtain the relative motion between the host vehicle and the obstacles. Combining the ego-motion with the relative obstacle motion we can obtain the absolute obstacle motion. The pixel motion is obtained using the image pair of two subsequent frames.

Sparse optical flow

The optical flow is the most common technique, in the state of the art, to obtain motion information from subsequent images. The algorithm consists in the extraction of the image points with the most informative content, *keypoints*. The extraction is based on the algorithm described in [47], while a brute force search is used in the matching phase. An example is shown in Fig.2.15, where the motion is represented by a bidimensional vector. The image prove as the motion of the host vehicle affects also the motion of the obstacles.



(a) Image classification

(b) Topview classification

Figure 2.14: Stochastic labeling example

Host vehicle movement

Only the points belonging to the static obstacles can be used to estimate the motion of the host vehicle, since their movement is affects only by the movement of the vehicle. The obstacles detection algorithm, described in 2.1.2, is used to separate points belonging to the obstacles: not being able to distinguish between static or moving obstacles, it has been considered all moving obstacles. Using a stereo-pair images, each keypoint has associated with a 3-D world position. In this way it is possible, basing on keypoint position, to discriminate if the points belong to an obstacle or not. An example of this phase is shown in Fig.2.16.

The figure shows also as some points are wrongly not associated to obstacles, mainly the points close to the obstacle border. This error resides to the technique used to determine the feature membership to an obstacle: it has been used the perceived

Figure 2.15: Optical flow example.

dimensions instead of the real one, because it is still not possible to know the obstacle orientation. Basing the motion estimation algorithm on $LO-RANSAC^2$, described in [26], some erroneous points can be neglected. The algorithm evolves in subsequent iterations, trying to find the dominant movement. A movement based on random points is estimated on each iterations. The model is then evaluated on all the other points, with an appropriate error metrics, in order to discriminate the point as *inlier* or *outlier*. The error calculated using all the *inlier* points is used to determine if the algorighms terminates. An example result is shown in Fig.2.17.

Obstacles movement

The obstacle movement is higly dependent on the camera movement. Considering a static obstacle framed in subsequent frames. Two ipothesys are suitable: the obstacle and the camera are moving with the same motion, the obstacle and the camera are both still. The motion information extracted from subsequent images can be used to estimate only the *relative* motion of the obstacle. The combination of the host vehicle motion and the obstacle motion is needed to estimated the obstacle absolute motion.

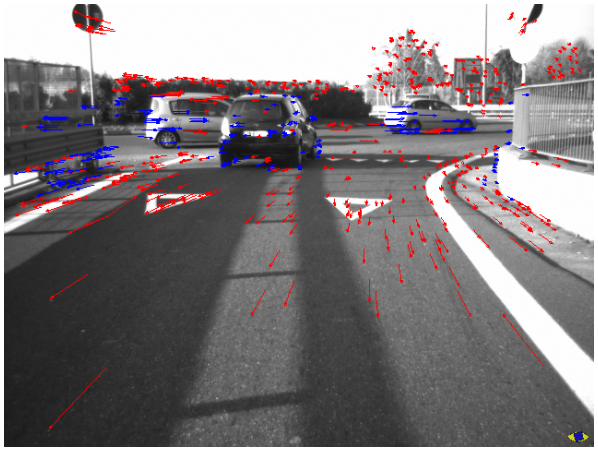


Figure 2.16: Features separation example.

2.1.4 Clustering refinement

The obstacle motion information can be used to improve the clustering approach described before, mainly to merge an obstacle wrongly splitted. Due to the limits of the perception system, it is possible that the clustering algorithm splits the hood and the side of the car in two obstacles. This happens when the same obstacles is perceived as two separates clouds. In this case, the stochastic propagation is not able to correctly cluster the obstacle. This step is need to overcome this problem and merge groups of cells close enough and with similar movement. The movement constraints has been introduced to fix problems as the one described before, and merge the hood and the side of the car in the same obstacle.

An example is shown in Fig.2.18. The truck is initially splitted in two obstacles, due to the inaccurate perception. Then, thanks to the camera and obstacle movement, it is possible the merge the front side and the back side of the truck in one obstacle.

2.1.5 Obstacles definition

Once defined the cells belonging to an obstacle, it is need to extract some informations in order to better characterize the obstacle: the contour, the dimensions and the

Figure 2.17: Host vehicle movement example.

position.

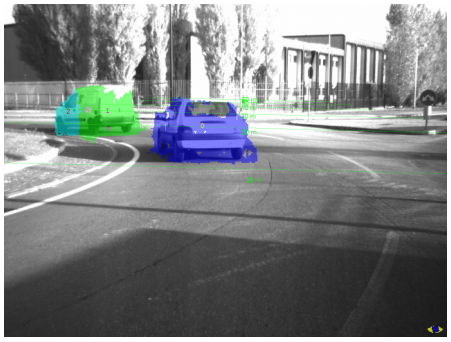
Contour

The contour represents the visible side of the obstacle at a given time, which allows to partially define its geometric properties. The contour definition is based on the analysis of the obstacle point cloud, according to the *Border Scanner* algorithm, that relies on the polar coordinates, as shown in Fig.2.19.

The point cloud is divided in several angolar section and for each one it is saved as contour the closest point. The angolar section dimension is based on step of the angular scansion: a lower value allows to a more accurate definition of the contour. The drawback of using a low angular scansion step is to introduce useless points in the contour that do not increase the information already present. Otherwise, a high angular scansion step can leads to introduce an excessive approximation.

Dimensions

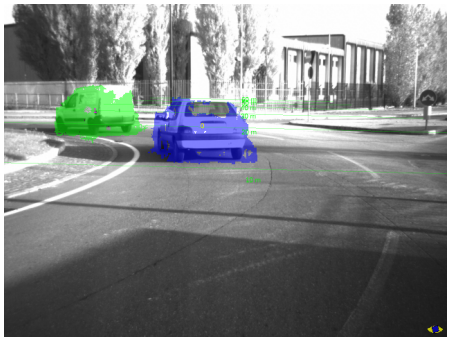
The obstacle dimensions could be defined directly on the point cloud, but with a high computational cost. The obstacle contour is then used to reduce the computational



(a) Erroneous clustering

(b) Optical flow

(c) Camera and obstacle movement extraction



(d) Clustering refinment

Figure 2.18: Clustering refinement.

time at the expense of an accuracy reduction. The dimension are the obstacle height, width and depth. The height is represented directly by the higher height of the point cloud points. The width and depth, instead, require to make a consideration: they make sense only if it is possible to define an orientation for the obstacle. Only in this case we can discriminate from width and depth. If the obstacle is moving, we can assume its orientation coincident with its moving direction; the visual odometry information described before can used in this case, as shown in Fig.2.20a.

In case the obstacle is not moving or it has not been possible to define its movement, the obstacle dimensions can be defined using the Principal Component Analy-

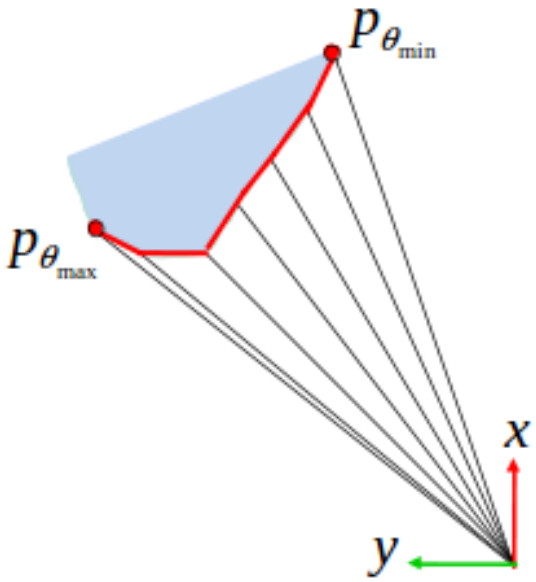
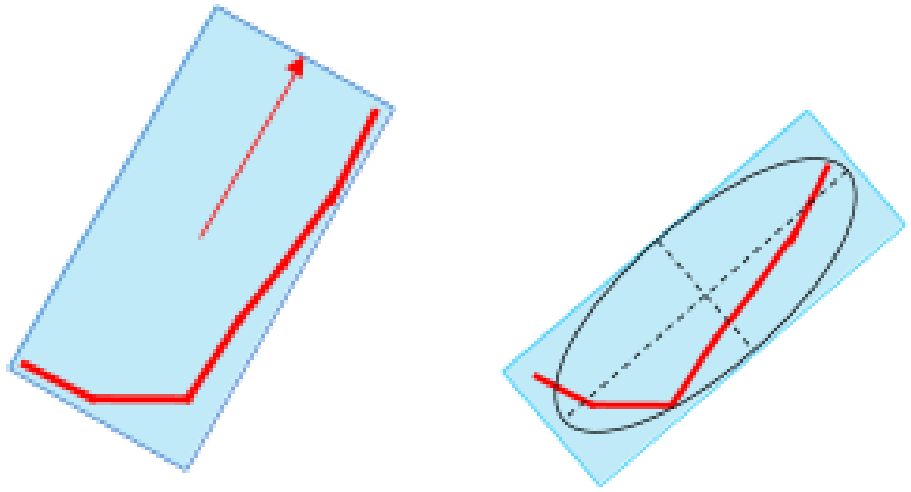


Figure 2.19: Contour definition based on Border Scanner algorithm.

sis (PCA), as shown in Fig.2.20b.

Position

Looking at the Fig.2.21a, it has beed defined nine key points for the obstacle, eight on the border and one in the center. The border points are directly observable and they are the best ones to describe the obstacle position. Two approaches has been investigated to define the obstacle position: the first one based on the most representative position and, the second one, on the closest position. The second approach, does not require any assumption for the position definition but, to be accurate, it requires a further tracking step. The first approach, instead, allow to have an initial guess regarding the most representative obstacle position. A interest region on the disparity map is used to built an histogram of the obstacle disparity values. The histogram is shown in Fig.2.21b, where the disparity has been replaced by the distance. The dominant



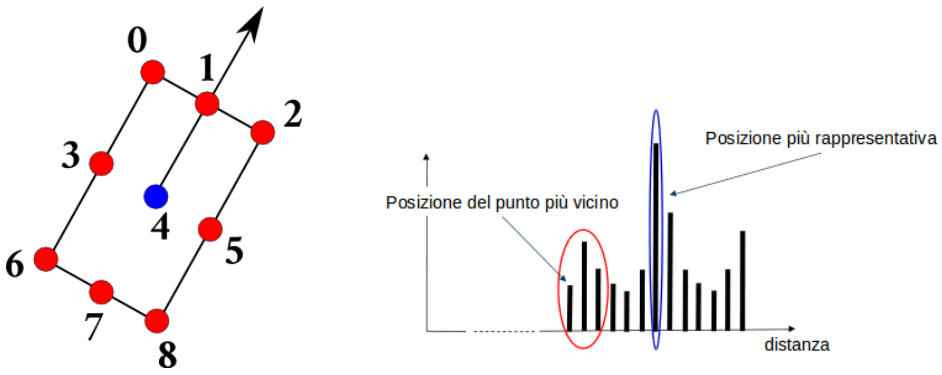
(a) Dimension definition based on visual odometry

(b) Dimension definition based on PCA

Figure 2.20: Obstacle dimensions definition

position is represented by the histogram maxima, highlighted in blu in the picture. According to the extracted distance, it is possible to define also the lateral and vertical distance analyzing the same interest region.

The second technique is based on the same approach, but focalized on the closest side of the obstacle. The histogram is replaced by a median filter in order to reduce the measurement error. Fig.2.21 shows the difference between the two approaches. The second approach relies on the knowledge of the obstacle orientation in order to define which of the 8 points has been observed. Due to the inaccurate estimation of the orientation withouth a tracking step, the obstacle position has been defined to the first approach leaving to the further tracking step the task of improving the obstacle position definition.



(a) Possible position observations

(b) Position definition approaches

Figure 2.21: Obstacle position definition

2.1.6 Obstacles classification

Two different input are used for this phase: the monocular image and the candidates obtained from the stereo obstacles detection. The monocular classification has been used to classify objects farthest than $60m$, don't detected by the stereovision.

Monocular classification

The monocular classification is based on the Aggregate Channel Features (ACF) [33] that is an extension and optimization of the Integral Channel Features (ICF) [35], and an AdaBoost cascade classifier. The classifier has been trained to detect vehicle and pedestrian, but can be easily extended to detect other classes. The full resolution image, $1280x960$, is used as input in this phase. The monocular classification, usually, relies on the sliding window approach: a window with fixed dimensions is shifted along the image where are extracted the features. These features are, then, used as input for the classifier. In order to reduce the number of windows to be analyzed, they have been introduced the search ranges: through geometrical considerations and the camera calibration parameters, it is possible to remove unconsistent windows and notevolly speed-up the entire classification process. In this case, only windows that

may contain objects farthest than $55m$ are passed to the classifier. A small overlapping zone has been mantained with the stereo obstacle detector, to avoid possible dark regions. Two ways can be used to calculate the obstacle distance: based on disparity map or based on the bird-eye view. If the classified region contains more than the 30% of valid disparity points, the median of these points are used to obtain the obstacle distance. Otherwise, considering a planar plane in front of the host vehicle, a bird-eye view has been used to estimate the obstacle distance. Obviously, this second approach, is much more uncertain than the first one, since a small variation on the base of the detected bounding box leads to a high variation of the obstacle distance.

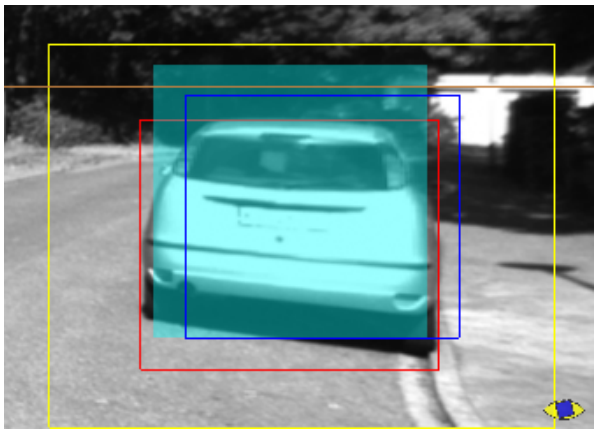


Figure 2.22: Example of stereo classification. The yellow box is the region where the classifier is run, the red one is the obstacle detected by the stereo, the blu one is the tracked obstacle and the filled blu box is the classifier output.

Stereo classification

The stereo classification relies also on the ACF and an AdaBoost cascade classifier as for the monocular classification. In this case, both the full and half resolution image are used as input, depending on the obstacle distance. The minimum image size of an object, detectable by the classifier, is known a priori. The stereo detector provides the distance of each candidate, which can be used to find its bounding box dimension.

If the dimension is lower than the minimum size of a detectable object, the classification is performed on the full resolution image. The use of both images, instead of classifying directly on the full resolution image, allows to speed-up the system. The classification is performed only on the bounding boxes provided by the stereo detection phase, as shown in Fig.2.22. Moreover, the distance and dimension information, are used to properly set the search ranges and limit the classifier detection to objects with a precise distance and dimension. The stereo obstacle can be also splitted in several obstacles in case of multiple classifier detection. This happens mainly in case of group of pedestrians where the clustering approach fails, as shown in Fig.2.23.

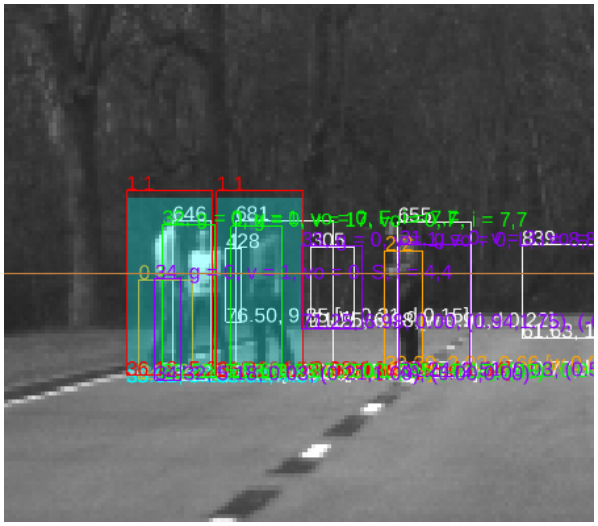


Figure 2.23: Example where the stereo detector fuse the group of pedestrian in only one obstacle but the classifier correctly split them in several object.

2.2 Obstacles tracking

The detection of a generic obstacle is a process of state observation affected by noise, which degrades the entire system performance. The errors can be attributed to the algorithm approximation or, simply, to the normal scene evolution. An example can

be a pedestrian between a vehicle and the camera, not observable in this case. These problems are overcome in the tracking stage, with approaches such those described in the Section 1.3. Each detected obstacle is associated with a state, which should provide a complete description on the object. The state is represented by:

$$S = \begin{bmatrix} x \\ y \\ v_x \\ v_y \\ \dot{\psi} \\ w \\ d \end{bmatrix} \tag{2.13}$$

where (x, y) are the centroid coordinate, v_x, v_y are the velocities on the $x-y$ plane, $\dot{\psi}$ is the yaw-rate and w and d are the object width and depth. The tracking is usually divided in two steps: the association and the filtering.

2.2.1 Association

The association process is responsible to create the corrispondences from the new detected obstacles and the previous tracked ones. Considering a case with N new obstacles and M tracked ones. The first step needed is to associate the new obstacles to the tracked ones. Often, $N \neq M$; this means both that not all the observations can be associated with a tracked object and both that not all the tracked objects can be associated with an observation. In the first case, the obstacle has appeared in the scene and it was observed for the first time, so it is inserted in the tracked list. In the second case, the tracked object has not been observed in this frame; if the event is repetead several time, the object is removed from the tracked list since it means that it disappeared from the scene.

In the described system, two groups of new obstacles can be associated with the tracked ones: the obstacle coming from the stereo obstacle detector and the ones coming from the laser. To account for all possible combination and obtain the best possible association, it has been introduced a multidimensional structure, as shown

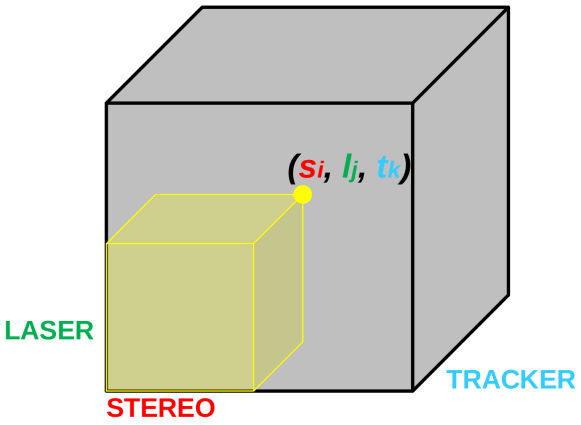


Figure 2.24: Multidimensional association structure.

in Fig.2.24. Each side of the structure represents a group of objects coming from the same source (laser, stereo, tracked list). An element of the structure represents, instead, the association confidence between all the combination of the source sensors. In Fig.2.24, for example, it is the sum of the contributes obtained from the association between the i-th stereo object and the j-th laser object, the i-th stereo object and the k-th tracked object and the j-th laser object and the k-th tracked object. Then, each side has a number of rows equal to the number of the sensor objects, plus an additional fake row. It is needed to manage the association between a subset of sensors: the fake stereo row, for example, is used to manage the association between the laser and tracked objects. The association process iteratively finds the maxima in the structure, until no more positive value are found. The association value between two objects, is a value ranging from -1 to 1 and it is estimated using a classifier. The classifier, based on AdaBoost [126], has been trained on a set of generic features extracted comparing some caractheristics of the two objects:

- cuboid overlap: overlap between the topview projection of the cuboid, estimated as the number of the overlapping cells;
- bounding box overlap: overlap between the objects bouding boxes in the im-

age;

- euclidean distance: distance between the objects centroids;
- dimension ratio: ratio between the objects width and ratio between the objects depth;
- dimension difference: difference between the objects with and difference between the objects depth.

Several classifiers has been trained since, in certain case, some features can not be extracted: if the laser object is outside the camera field-of-view, it is not possible to be projected on the image and obtain its bounding box. According to the availabile features, it is running the appropriate classifier.

2.2.2 Tracking

The tracking is based on the Unscented Kalman Filter, with the state described in the equation 2.13. It has been used the *vehicle reference system*, shown in Fig.2.6, in order to avoid distance errors when the host vehicle is pitching. Moreover, each tracked obstacle has its own pitch, that allows to obtain the *world reference system* coordinate withouth assuming a ground with costant slope. The tracking is divided in three main parts: initialization, matching and update.

Initialization

When an obstacle is observed for the first time, it is needed to initialize the filter, in order to be able to manage further observations. This happens when the detected obstacle has not been associated with previous tracked object. If two detected obstacles, from laser and stereo, has been associated with each other but not with a tracked object, the new observation will be formed by two objects. In this case, the tracked object is initialized using the covariance weighted mean of the obstacles parameters: centroid, dimension, pitch, speed and contour. The speed may take two steps to be correctly inizialized, depending if the new obstacle has also the speed information.

Otherwise, it will be used the position of the centroid between two observations, with the elapsed time, to inizialize the speed at the second step. An histrogram build over theta values is used, instead, to initialize the contour. The bin values are incremented with the covariance weighted rho. For each object, this phase is needed just the first time, while the prediction and observation phase are executed recursively untill the object comes out the tracked list.

Prediction

The prediction phase updates the state of each object to the current time. If we consider the execution flow as a sequence of discrete events, this step projects the previous observed obstacles to the actual time. This task allows to mantain consistent the filter state and, at the same time, to make more efficent and robust the association. A costant acceleration model has been used for the object motion, distinguishing between static and moving obstacle:

- static obstacle: prediction based only on ego-motion E, $\widehat{x} = E^{-1}x_{t-1}$, with obstacle speed fixed to zero;
- dynamic obstacle: prediction based on ego-motion E and obstacle absolute movement V: $\widehat{x} = VE^{-1}x_{t-1}$.

It is needed the combination of the ego-motion and the obstacle-motion for the dynamic object since the state is represented by the relative position of the obstacle respect to the host vehicle. This requires to take account of the movement of the host vehicle as well as the movement of the object.

Matching

After a candidate has been associated with the tracked object, the matching step consists in two steps: the *gross error* check and the candidate update. The gross error is a value that can be used to evaluate the goodness of the association according to the

state covariance and the observation noise:

$$g = e^T C_{RR}^{-1} e \tag{2.14}$$

Where e is the difference between the predicted observation vector and the observation vector:

$$e = \begin{bmatrix} \widehat{x} - x \\ \widehat{y} - y \\ \widehat{v_x} - v_x \\ \widehat{v_y} - v_y \\ \widehat{\dot{\psi}} - \dot{\psi} \\ \widehat{w} - w \\ \widehat{d} - d \end{bmatrix} \tag{2.15}$$

The matrix C_{RR}, instead, is formed by:

$$C_{RR} = C_{OBS} + H C_{XX} H^T \tag{2.16}$$

- C_{XX}: state covariance matrix;
- C_{OBS}: observation noise matrix;
- H: Jacobian to convert from the state space to the observation space.

The lower the value of the gross error the better is the association. Then it is possible to threshold this value in order to remove possible erroneous association. In a tracking system based on a Kalman filter, it is preferable to avoid wrong matching and evolve with the only prediction phase.

The candidate update consists in the updating of its parameters according to the tracked object to which it has been associated. This step is performed to take advantage of the higher stability of the tracked object measurements in the calculation of the candidate parameters: dimensions and reference point. The dimensions are updated using the rotated contour according to the tracked orientation, as shown in

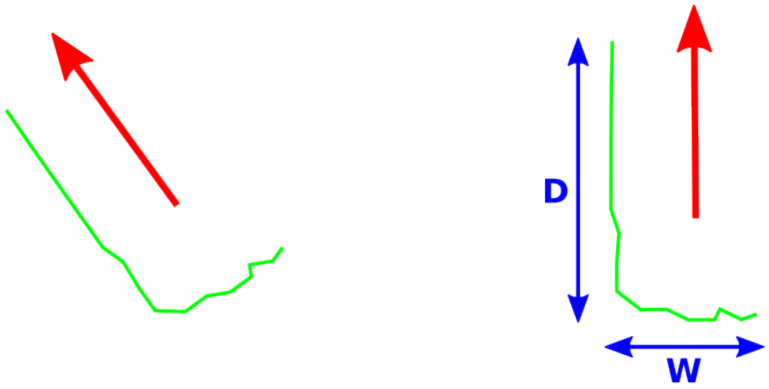


Figure 2.25: Dimensions update based on the rotated contour.

Fig.2.25. Regarding the reference point, instead, it is used the contour point closer to the cuboid vertex, actually tracked, of the tracked object as shown in Fig.2.26.

Update

After the matching process, the candidate is used to update the state of the associated tracked object. The tracking model is based on the *best knowledge model*. The idea is to describe each object respect to the point closer to the host vehicle, *the reference point*: it is the point with the lowest observation error and, therefore, ensure a reliable tracking.

The reference point is identified with an index ranging between 1 and 9, as shown in Fig.2.27. It can be obtained from the position *(x,y)*, the width w, the length l and the observation angle α. This angle is used to describe the object orientation respect to the host vehicle. It is estimated considering the yaw ψ and the angle β between the object position *(x,y)* and the reference system origin:

$$\alpha = \psi - \beta \tag{2.17}$$

In the Fig.2.27, for example, the reference point is represented by the point with the index number 5. The observation vector is built in a modular structure, which permits to manage and arbitraly combine different kind of observations. This is required

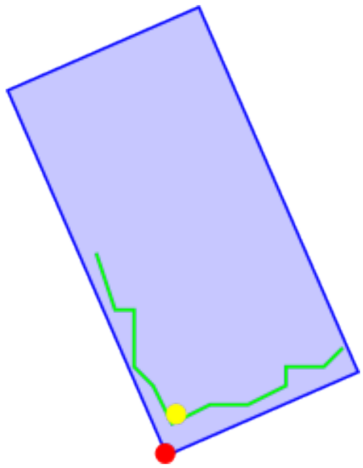


Figure 2.26: Reference point update: the red point is the actually tracked vertex of the tracked object, the yellow one is the closer contour point set a reference.

because, depending to the associated candidate, the observations vary each time. The handled observations are:

- reference point position in (u, v, d) coordinate;
- reference point position in (x, y, z) coordinate;
- reference point position in (u, v) coordinate, assuming flat ground;
- width in meters or pixels;
- depth in meters or pixels;
- speed along $x - y$ axes.

It has been used the following updating equations:

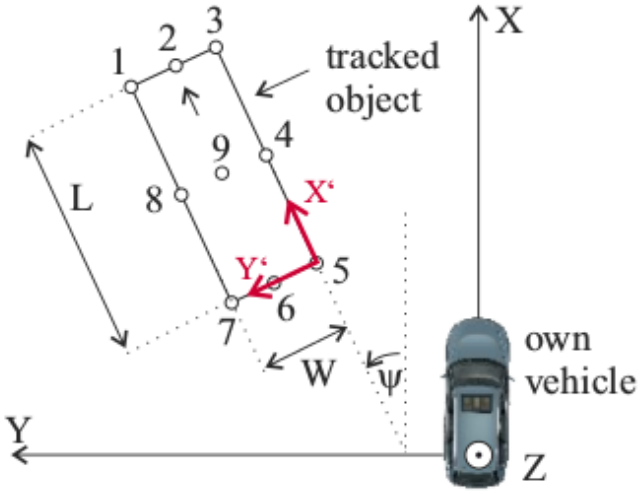


Figure 2.27: Best knowledge model.

Mono obstacles

$$y_m(t) = \begin{bmatrix} u_{pixel} \\ v_{pixel} \\ h_{pixel} \\ w_{pixel} \\ d_{pixel} \\ V^b(t) \\ \omega_\psi(t) \end{bmatrix} = h_m(x(t)) + v_m(t)$$

$$h_m(x(t)) = \begin{bmatrix} t\left(p^b(t)\right)_x \\ t\left(p^b(t)\right)_y \\ \left| t\left(p^b(t)\right)_y - t\left(p^b(t) - \begin{bmatrix} 0 & 0 & H(t) \end{bmatrix}^T\right)_y \right| \\ \frac{K_u(l_x(t)-r_x(t))}{(R_b^c g_{ctr}^{idx}(p^b(t)))_z} \\ \frac{K_u(f_x(t)-b_x(t))}{(R_b^c g_{ctr}^{idx}(p^b(t)))_z} \\ R_n^b V^n(t) \\ \omega_\psi(t) \end{bmatrix} \tag{2.18}$$

$$v_m = \begin{bmatrix} v_u(t) & v_v(t) & v_h(t) & v_w(t) & v_d(t) & v_v(t)^T & v_{\omega_\psi}(t) \end{bmatrix}^T$$

Stereo obstacles

$$
\begin{aligned}
y_s(t) &= \begin{bmatrix} disparity \\ u_{pixel} \\ h_{pixel} \\ w_{pixel} \\ d_{pixel} \\ V^b(t) \\ \omega_\psi(t) \end{bmatrix} = h_s(x(t)) + \nu_s(t) \\
h_s(x(t)) &= \begin{bmatrix} \frac{K_u B}{\left(R_b^c g_{ctr}^{idx}\left(p^b(t)\right)\right)_z} = d(t) \\ \frac{\left(R_b^c g_{ctr}^{idx}\left(p^b(t)\right)\right)_x d(t)}{B} + u_0 \\ \left| \frac{\left(R_b^c g_{ctr}^{idx}\left(p^b(t)\right)\right)_y d(t)}{B} + v_0 - \frac{\left(\left(R_b^c g_{ctr}^{idx}\left(p^b(t)\right)\right)_y - H(t)\right) d(t)}{B} + v_0 \right| \\ \frac{K_u (l_x(t) - r_x(t))}{\left(R_b^c g_{ctr}^{idx}\left(p^b(t)\right)\right)_z} \\ \frac{K_u (f_x(t) - b_x(t))}{\left(R_b^c g_{ctr}^{idx}\left(p^b(t)\right)\right)_z} \\ R_n^b V^n(t) \\ \omega_\psi(t) \end{bmatrix} \\
\nu_s &= \begin{bmatrix} \nu_{disparity}(t) & \nu_u(t) & \nu_h(t) & \nu_w(t) & \nu_d(t) & \nu_v(t)^T & \nu_{\omega_\psi}(t) \end{bmatrix}^T
\end{aligned}
\tag{2.19}
$$

Laser obstacles

$$
\begin{aligned}
y_l(t) &= \begin{bmatrix} p^b \\ w_{meters} \\ d_{meters} \\ V^b(t) \end{bmatrix} = h_l(x(t)) + \nu_l(t) \\
h_l(x(t)) &= \begin{bmatrix} g_{ctr}^{idx}\left(p^b(t)\right) \\ W(t) \\ D(t) \\ R_n^b V^n(t) \end{bmatrix} \\
\nu_l &= \begin{bmatrix} \nu_{p^b}(t)^T & \nu_w(t) & \nu_d(t) & \nu_v(t)^T \end{bmatrix}^T
\end{aligned}
\tag{2.20}
$$

Radar obstacles

$$
\begin{aligned}
y_r(t) &= \begin{bmatrix} p^b \\ d_{meters} \\ V^b(t) \end{bmatrix} = h_r(x(t)) + \nu_r(t) \\
h_r(x(t)) &= \begin{bmatrix} p^b(t) \\ D(t) \\ R_n^b V^n(t) \end{bmatrix} \\
\nu_r &= \begin{bmatrix} \nu_{p^b}(t)^T & \nu_d(t) & \nu_v(t)^T \end{bmatrix}^T
\end{aligned}
\tag{2.21}
$$

In the previous equations:

- $t(x)$ is the transformation from the body reference system to the image reference system;
- $g_{ctr}^{idx}(x)$ is the transformation from the obstacle centroid to the best point to track;
- K_u is the pixel focal length of the camera;
- B is the baseline of the stereo camera;
- $l(t)$ and $r(t)$ have different meanings depending on the specific scenario: when the obstacle is approaching the camera they represent the leftmost and rightmost points of the obstacle front, when the vehicle is getting away from the camera they represent the leftmost and rightmost points of the obstacle rear. In case of static obstacle there is no difference between the front and the rear part of the obstacle.
- $f(t)$ and $b(t)$ have different meanings depending on the specific scenario: when the obstacle is running from the left to the right of the camera they represent the frontmost and backtmost points of the vehicle right side, when the vehicle is running from the right to the left of the camera they represent the frontmost and backtmost points of the vehicle left side;

As said previously, the observations may vary time to time. This is because, first of all, the candidate can come from different sources like stereo or laser sensors and, moreover, it can be seen from different observation angle. If the candidate is originated by the only laser sensor, it will provided its position in (x, y, z) coordinate while, in case of stereo sensor, it will provide its position in (u, v, d) coordinate. As described in the Section 2.2.1, the candidate can be also provided by both sensors and, in this case, will be observed both the (x, y, z) and (u, v, d) position. The dimension observations, instead, depend on the observation angle. If the object is exactly in front of the host vehicle, for example, it is not possible to observe the object depth. The opposite situation is when the object is perpendicular to the host vehicle, in this case the width is not observabled. Then, according to the source sensors and the observation angle, the observation vector is modularly built.

Chapter 3

Results

The system evaluation has been divided in two parts: the obstacles detection and the obstacles tracking. A qualitative and quantitative analysis has been carried out, highlighting advantage e disadvantage. The Kitti dataset has been used as ground truth: it is composed by 21 annotated sequences in urban and extra-urban roads, provided by a collaboration between the Karlsruhe Institute of Technology and the Toyota Technological Institute of Chicago.

3.1 Obstacles detection

The obstacle has been defined as a zone or object that is an obstruction for the movement of the vehicle. According to this definition, the system aims to detect these zones and permits a safer planning. The system has paid particular attention to the moving obstacles as vehicles and pedestrians.

3.1.1 Qualitative results

False positives robustness and a correct segmentation are key points for a good obstacles detection system. Several scenarios will be presented in order to analyze the algorithm working under different conditions. In Fig 3.1, a far obstacle is divided in two separate obstacles due to a closer obstacle that partially occludes it. The segmen-

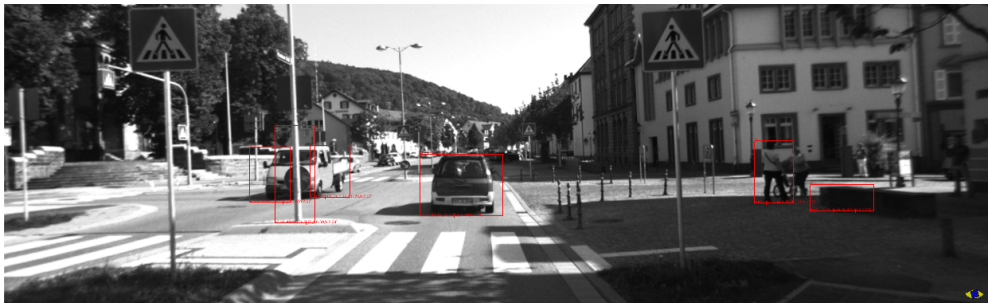


Figure 3.1: False positive example with partial interposition of an other obstacle.

Figure 3.2: True positive example with partial interposition of an other obstacle.

tation phase, based on motion information, would have to merge the two obstacle parts. But, because of the distance, it was not possible to obtain motion information and, therefore, to merge them. Different is the situation of Fig.3.2, where it was possible to recover motion information and correctly merge the obstacles parts initially divided.

In Fig. 3.3, is shown an example of erroneous segmentation of the two pedestrians on the right. The planner can neglect this error, since it will not affect the planning of a good trajectory; in case of surveillance, instead, would be interesting to split them in two separated pedestrians. In Fig. 3.4, instead, is shown a correct case of segmentation with the two pedestrians correctly separated from the car.

Illumination changes is a negative factor for the algorithm of computer vision. High illumination leads to image saturation and, then, to errors in the disparity map.

Figure 3.3: Example of pedestrian segmentation error.

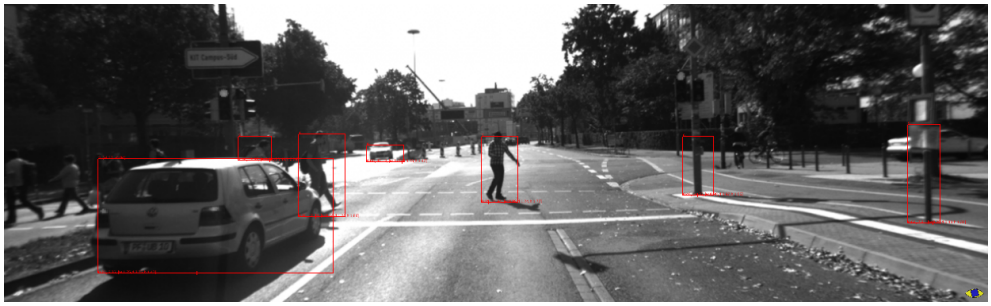


Figure 3.4: Example of correct pedestrian segmentation.

A sparse disparity map can produce false negatives and, on the other hand, wrong disparity values can produce false positives. An example of false positive is shown in Fig. 3.5. A correct detection and segmentation of vehicles can be seen in Fig.3.6 and Fig.3.7, where closer vehicles are correctly split in separated obstacles.

3.1.2 Quantitative results

A statistical analysis has been performed on the Kitti platform. This dataset provides annotated sequences, with obstacles position, dimension, orientation and classification. The true negative has not been considered due to the intrinsic nature of the system: the obstacles detection provides always a positive result and the negative one is not covered.

True positives and false positives and negatives are defined comparing the de-

Figure 3.5: Example of false positive due to the illumination condition.

Figure 3.6: Example of correct vehicles detection.

tected obstacles and the annotated ones at the time of the sequence. Looking at Fig.3.9, where the green obstacles are the detected ones and the red obstacles are the annotated ones. An association step is needed to relate the detected and annotated obstacles. Each detected obstacle is associated with the closest annotated one, considering a maximum distance more than which the association is not valid. An overlapping region is calculated for each association. Since the regions are represented as rectangles, considering the Fig.3.8, the overlapping between the areas A_1 and A_2 is calculated as:

$$\text{overlapping} = \frac{A_1 \cap A_2}{(A_1 \cup A_2)/(A_1 \cap A_2)} \tag{3.1}$$

Each association is considered a true positive if the overlapping is more than a certain threshold, for example 0.5, since the value is normalized. In Fig.3.9 the

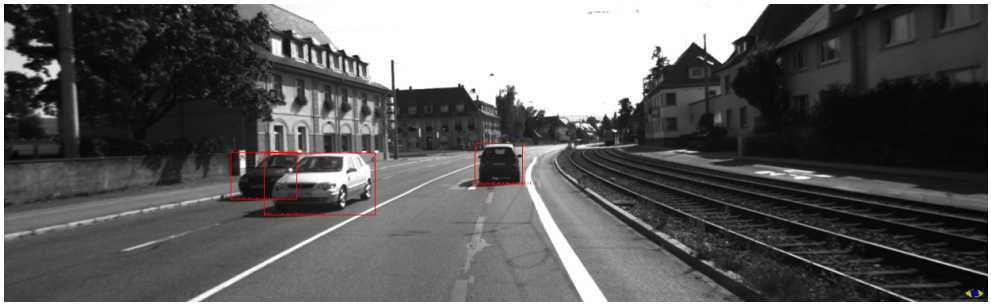


Figure 3.7: Example of correct vehicles detection.

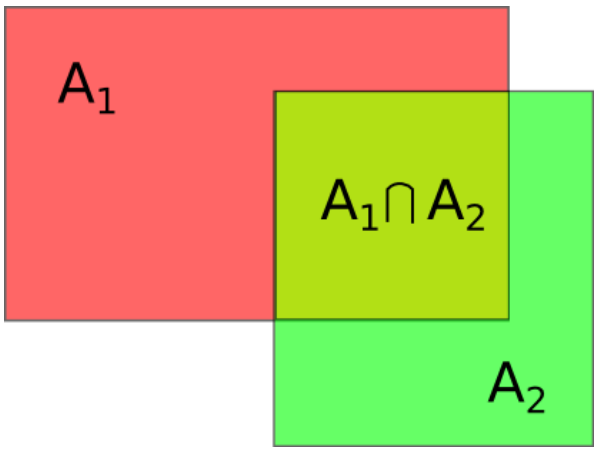


Figure 3.8: Regions overlapping.

central vehicle and the one parked on the right are considered true positives. A false negative, is represented by an annotated obstacle not associated. An example is shown in Fig.3.10, where the farther vehicle is not detected by the perception system.

The false positives, instead, has been defined as a not valid associations. Considering the Fig.3.11, where the two detected vehicles are not considered valid since the bounding boxes overlapping is not enough accurate. The results has been obtained on 20 sequences. True positive, false negative and false positive are reported. Moreover, it has been defined the *precision* and *recall*:

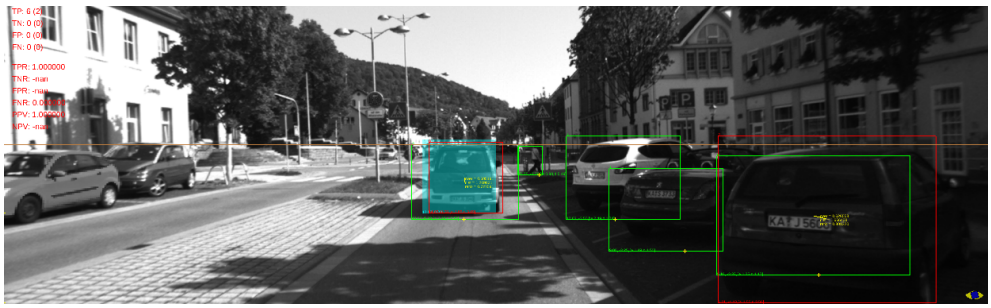


Figure 3.9: Example of true positive.

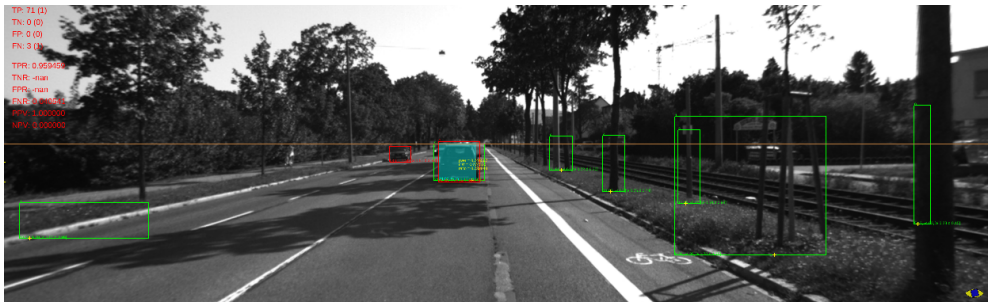


Figure 3.10: Example of false negative.

$$\text{Precision} = \text{PPV} = \frac{TP}{TP+FP} \tag{3.2}$$

$$\text{Recall} = \text{TPR} = \frac{TP}{TP+FN} \tag{3.3}$$

Table 3.1 reports the detailed results. It is possible to see as the results are not constant, and a description for each sequence would be helpful. The sequence 13,16 e 18 are obtained in a urban area. Therefore several group of pedestrians are present in this sequences, which make hard to distinguish them as separated obstacles. The poor results in the sequences 8 and 14, instead, are attributable to the adverse lighting.

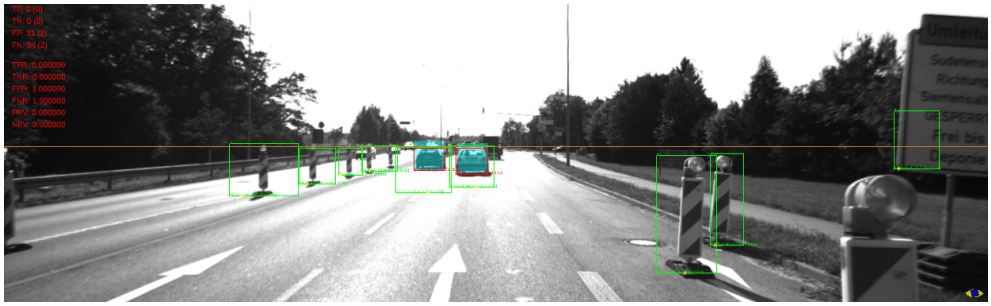


Figure 3.11: Example of false positive.

Sequenza	TP	FP	FN	PPV	TPR
00	58	41	71	0.586	0.450
01	144	64	121	0.693	0.543
02	102	74	150	0.580	0.405
03	112	14	16	0.890	0.875
04	236	6	71	0.975	0.769
05	121	4	37	0.968	0.766
06	90	12	148	0.882	0.378
07	416	102	103	0.803	0.718
08	73	250	429	0.222	0.145
09	154	72	98	0.681	0.611
10	340	7	33	0.980	0.912
11	540	33	33	0.942	0.942
12	31	0	74	0.295	1.000
13	188	443	551	0.298	0.254
14	48	55	85	0.467	0.361
15	50	25	38	0.667	0.568
16	157	386	467	0.289	0.252
17	405	83	117	0.830	0.776
18	117	2021	2231	0.356	0.334
19	1189	210	303	0.850	0.797

Table 3.1: Stereo obstacle detector results.

3.2 Obstacles tracking

In this section are shown the results comprehensive of the tracking stage. The performance has been evaluated on the Kitti dataset, the same used for the stereo obstacles detector testing. For the tracking system, the Kitti platform provides a software suite to compare the results with the state of the art systems. Several metrics are used:

- **MOTP**: Multiple object tracking precision, defined as the total error in estimated position for matched object-hypotesis pairs over all frames, averaged by the total number of matches made.
- **IDS**: ID switches, the total number of times that a tracked object changes its matched ground truth identity.
- **FRAG**: Fragments, the total number of times that an object tracking is interrupted in the tracking results.

A detailed description of the metrics can be found in [14, 75]. The table 3.2 shows the system results compared with the state of the art ones. It is possible to see as our system outperforms the others in all the metrics, with one of the lowest computational time.

Algorithm	MOTP	IDS	FRAG	Runtime
Our system	85.61%	15	250	0.092s
NOMT-HM	80.10%	109	378	0.09s
DCO_X	78.96%	327	996	0.9s
mbodSSP	78.83%	117	894	0.01s
TBD	78.47%	33	540	10s

Table 3.2: Obstacle tracking comparison with state of the art.

The error spreads over the distance is shown in Table 3.3. The distance has been choosed as the distance of the closest object between the detected and the ground truth one. The error percentual tends to be reduced with the increasing of the distance. This stems from the fact that the tracking improves the distance accuracy over the frames: a closer object just detected has a higher error that a farther one tracked for several frames.

Distance	GT Elements	Average Error	Percentual
0-10m	635	0.262m	5.62%
10-20m	3365	0.644m	4.72%
20-30m	5219	0.848m	3.82%
30-40m	3818	1.262m	3.90%
40-50m	2618	1.274m	3.15%
50-60m	1364	1.496m	2.99%
60-70m	803	1.525m	2.38%
70-80m	377	1.567m	2.33%

Table 3.3: Obstacle tracking results.

3.2.1 Computational time

Table 3.4 shows the time required by each stage of the system.

Section	Min [ms]	Max [ms]	Avg [ms]
Initialization	9.5	21.0	12.6
Features extraction	11.9	25.9	16.5
Disparity	23.8	42.1	29.6
Stereo OD	4.2	11.0	5.9
Visual Odometry	0.4	4.0	0.8
Stereo Classification	3.1	50.7	10.7
Mono Classification	8.9	22.3	14.7
Tracking	0.4	2.4	1.4
Total	78.8	125.1	92.2

Table 3.4: Computational time of the system.

Most of the time is spent in the creation of the disparity map and the features extraction process. For this reason has been introduced the multiresolution disparity, trying to reduce as much as possible the computational time of this part. A significant part of time is also spent in the classification steps: a threshold has been set, in order to control the elapsed time.

Chapter 4

Conclusions and Future directions

The dissertation closing arguments are presented in this chapter. A critical assessment with contributions and conclusions, future works and important directions will be presented.

4.1 Summary

The described system is able to detect obstacles from a stereo camera. The stereo obstacles are, then, tracked and fused with the ones provided by a Lidar system. Several automotive application can be derived by the proposed approach: the adaptive cruise control that automatically reduce the speed in order to maintain a safe distance with the front vehicle or the advanced emergency braking system that controls the proximity and automatically brake in case of dangerous situation. The algorithm starts from the depth information of the disparity map. The point cloud is used the built an occupancy grid, with fixed distance cells. The occupancy grid is fundamental to reduce the data complexity and analyze the problem from an higher level. The density based classification has been used to distinguish between free cells and cell occupied by obstacles. The grid construction has taken into account the measurement error, propagating the single measurement over multiple cells. A further segmentation step allow to cluster obstacles cells in single obstacles. The visual odometry on points not

belonging to an obstacles has been used to extract motion information of the host vehicle. The visual odometry on obstacles points, instead, has been used to obtain the relative objects motion information. Their combination has allowed to obtain the absolute motion information for the obstacles. This data are also used for a refinement of the segmentation step: close obstacles with same motion information are merged in a unique obstacle. The segmentation could be improved, especially in complex environment as the urban scenario with group of pedestrian. After the detection, the obstacles are classified as pedestrians or vehicles using a SofCascade Adaboost with ACF. Obstacles can be split in this phase if multiple detections are found in a single stereo object. An UKF based tracking step is performed after the classification. The *best knowledge model* approach has been used to track the best observable point. The tracking uses a modular structure that allows to arbitrarily combine different observations. The laser objects, once associated, are directly combined with stereo objects in the observation step. The association is based on an AdaBoost classifier trained on generic characteristics. The approach has been compared with the state of art, showing better results in all the analyzed metrics. Moreover, the system works at higher frame rate, making it appropriate for automotive applications.

4.2 Conclusions

The system has demonstrated a correct reconstruction of the dynamic world surrounding the vehicle, proving to be able to help the driver in the assessment of critical situations.
In particular, the developed algorithm provides a stable, robust and reliable detection, classification and tracking of the multiple targets coming from different sources. Moreover, the proposed approaches were seen to outperform the state of the art approaches on a public dataset.
The probabilistic DEM includes the measurements error of the disparity maps in the grid construction. This allows to probabilistically propagate the single cell measurement to the neighborhood, considering the real committed error, improving the original static propagation and obtaining remarkable advantages in the segmentation

phase. The PDF gradient is used to separate close obstacles.
A combination of monocular and stereo classification both on full and half-resolution images, in combination with an hard time constraints inserted in the classifier pipeline, has led to significantly speed up the system. This is fundamental for automotive applications, where real-time processing is a strong constraint; especially given that the classification step is, often, the most demanding in terms of time.
Assign a distance to the detected obstacle is a complex task, due to the different factors that can vary every time: orientation, form, viewpoint, etc.... The attention has been moved from find the distance of a point, to find the best point which we know with high certainty the distance. The *best knowledge model* aims to describe the object with respect to the closest point: it has the lowest uncertainty and it is not necessary to make any assumptions. Considerable benefits are obtained respect to the common approaches in the scientific community where it is tracked the obstacle centroid.
A fault tolerant and reliable system requires sensors redundancy and complementarity. Common approaches rely on object level fusion where only high-level information are used. This leads to a fast processing time but, at the same time, produces poor results being unable to exploit the specific sensor data. On the other hand, the low-level fusion is based on pixel-wise data, but it is time consuming and does not produce better results since it lacks general considerations at object level. For this reason, it has been introduced a medium-level fusion which take advantage from both the approaches. The fusion is performed at object level but preserving the low-level information; in this way it is guaranteed a real-time processing exploiting all available information.

4.3 Direction for Future works

The thesis has covered a large area in the field of detection, tracking and fusion of obstacles, which leaves considerable possible improvements. In the following, interesting directions for future works are sketched.

- In case of sensors group looking at same direction, significant improvements

can be obtained with a partial low-level fusion DEM. A unique DEM with sensors points is built for each group of sensors looking at the same direction. The DEMs and the detected obstacles are then fused with the approach described in the paper.

- The segmentation can be improved introducing further information such as color, texture etc... in addition to the 3-D data, extending the work of Nedevschi et al. [50].
- The association phase can be removed using the Probability Hypothesis Density Filter, which will improve also the tracking phase, pursuing different hypotheses otherwise discarded with this approach.
- The filter can be extended with a constrained Kalman filter with a kinematic evolution model, in order to force the vehicles to move under kinematic constraints. This requires to have a different filter and evolution for vehicles and pedestrians.

Bibliography

[1] B. Alefs, D. Schreiber, and M. Clabian. Hypothesis based vehicle detection for increased simplicity in multi-sensor acc. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 261–266, June 2005.

[2] G. Alessandretti, A. Broggi, and P. Cerri. Vehicle and guard rail detection using radar and vision data fusion. *Intelligent Transportation Systems, IEEE Transactions on*, 8(1):95–105, March 2007.

[3] J. Arrospide, L. Salgado, M. Nieto, and F. Jaureguizar. On-board robust vehicle detection and tracking using adaptive quality evaluation. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2008–2011, Oct 2008.

[4] O. Aycard, Q. Baig, S. Bota, F. Nashashibi, S. Nedevschi, C. Pantilie, M. Parent, P. Resende, and T.-D. Vu. Intersection safety using lidar and stereo vision sensors. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 863–869, June 2011.

[5] B. Aytekin and E. Altug. Increasing driving safety with a multiple vehicle detection and tracking system using ongoing vehicle shadow information. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 3650–3656, Oct 2010.

[6] H. Badino, U. Franke, and R. Mester. Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming. *Workshop on Dynamical Vision, ICCV*, 2007.

[7] Q. Baig, O. Aycard, T. D. Vu, and T. Fraichard. Fusion between laser and stereo vision data for moving objects tracking in intersection like scenario. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 362–367, June 2011.

[8] A. Bak, S. Bouchafa, and D. Aubert. Detection of independently moving objects through stereo vision and ego-motion extraction. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 863–870, June 2010.

[9] B. Barrois, S. Hristova, C. Wohler, F. Kummert, and C. Hermes. 3d pose estimation of vehicles using a stereo camera. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 267–272, June 2009.

[10] A. Bartels, M. Meinecke, and S. Steinmeyer. Lane change assistance. *Springer-Verlag*, 2012.

[11] A. Barth and U. Franke. Where will the oncoming vehicle be the next second? In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1068–1073, June 2008.

[12] A. Barth and U. Franke. Estimating the driving state of oncoming vehicles from a moving platform using stereo vision. *Intelligent Transportation Systems, IEEE Transactions on*, 10(4):560–571, Dec 2009.

[13] A. Barth and U. Franke. Tracking oncoming and turning vehicles at intersections. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 861–868, Sept 2010.

[14] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *J. Image Video Process.*, 2008:1:1–1:10, Jan. 2008.

[15] M. Bertozzi, L. Bombini, P. Cerri, P. Medici, P. Antonello, and M. Miglietta. Obstacle detection and classification fusing radar and vision. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 608–613, June 2008.

[16] S. Bota and S. Nedevschi. Tracking multiple objects in urban traffic environments using dense stereo and optical flow. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 791–796, Oct. 2011.

[17] A. Broggi, S. Cattani, E. Cardarelli, B. Kriel, M. McDaniel, and H. Chang. Disparity space image's features analysis for error prediction of a stereo obstacle detector for heavy duty vehicles. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 80–86, Oct 2011.

[18] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H. G. Jung. A new approach to urban pedestrian detection for automatic braking. *Intelligent Transportation Systems, IEEE Transactions on*, 10(4):594–605, Dec 2009.

[19] I. Cabani, G. Toulminet, and A. Bensrhair. Contrast-invariant obstacle detection system using color stereo vision. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 1032–1037, Oct 2008.

[20] G. Catalin and S. Nedevschi. Object tracking from stereo sequences using particle filter. In *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on*, pages 279–282, Aug 2008.

[21] Y.-M. Chan, S.-S. Huang, L.-C. Fu, and P.-Y. Hsiao. Vehicle detection under various lighting conditions by incorporating particle filter. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 534–539, Sept 2007.

[22] P. Chang, D. Hirvonen, T. Camus, and B. Southall. Stereo-based object detection, classification, and quantitative evaluation with automotive applications.

In *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pages 62–62, June 2005.

[23] W.-C. Chang and C.-W. Cho. Real-time side vehicle tracking using parts-based boosting. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 3370–3375, Oct 2008.

[24] R. Chavez-Garcia, J. Burlet, T.-D. Vu, and O. Aycard. Frontal object perception using radar and mono-vision. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 159–164, June 2012.

[25] V. Cherkassky. The nature of statistical learning theory . *Neural Networks, IEEE Transactions on*, 8(6):1564–1564, Nov 1997.

[26] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In B. Michaelis and G. Krell, editors, *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 236–243. Springer Berlin Heidelberg, 2003.

[27] J. Cui, F. Liu, Z. Li, and Z. Jia. Vehicle localisation using a single camera. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 871–876, June 2010.

[28] N. Dalal and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European Conf. Computer Vision (ECCV), 2006. ECCV. IEEE Conference on*, pages 428–441, 2006.

[29] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

[30] R. Danescu, F. Oniga, and S. Nedevschi. Modeling and tracking the driving environment with a particle-based occupancy grid. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1331–1342, Dec 2011.

[31] T. Dang, J. Desens, U. Franke, D. Gavrila, L. Schafers, and W. Ziegler. Steering and evasion assist. *Springer-Verlag*, 2012.

[32] A. Discant, A. Rogozan, C. Rusu, and A. Bensrhair. Sensors for obstacle detection - a survey. In *Electronics Technology, 30th International Spring Seminar on*, pages 100–105, May 2007.

[33] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1532–1545, Aug 2014.

[34] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–761, April 2012.

[35] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. In *British Machine Vision Conference*, 2009.

[36] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, Jan. 1972.

[37] M. Enzweiler and D. Gavrila. A mixed generative-discriminative framework for pedestrian classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

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

[39] F. Erbs, A. Barth, and U. Franke. Moving vehicle detection by optimal segmentation of the dynamic stixel world. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 951–956, June 2011.

[40] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[41] U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6d-vision: Fusion of stereo and motion for robust environment perception. In W. Kropatsch, R. Sablatnig,

and A. Hanbury, editors, *Pattern Recognition*, volume 3663 of *Lecture Notes in Computer Science*, pages 216–223. Springer Berlin Heidelberg, 2005.

[42] J. Fritsch, T. Michalke, A. Gepperth, S. Bone, F. Waibel, M. Kleinehagenbrock, J. Gayko, and C. Goerick. Towards a human-like vision system for driver assistance. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 275–282, June 2008.

[43] T. Gandhi and M. Trivedi. Pedestrian protection systems: Issues, survey, and challenges. *Intelligent Transportation Systems, IEEE Transactions on*, 8(3):413–430, Sept 2007.

[44] F. Garcia, P. Cerri, A. Broggi, A. de la Escalera, and J. Armingol. Data fusion for overtaking vehicle detection based on radar and optical flow. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 494–499, June 2012.

[45] A. Geiger and B. Kitt. Object flow: A descriptor for classifying traffic motion. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 287–293, June 2010.

[46] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Proceedings of the 10th Asian Conference on Computer Vision - Volume Part I*, ACCV'10, pages 25–38, Berlin, Heidelberg, 2011. Springer-Verlag.

[47] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968, June 2011.

[48] P. Geismann and G. Schneider. A two-staged approach to vision-based pedestrian recognition using haar and hog features. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 554–559, June 2008.

[49] D. Geronimo, A. Lopez, A. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1239–1258, July 2010.

[50] I. Giosan and S. Nedevschi. Superpixel-based obstacle segmentation from dense stereo urban traffic scenarios using intensity, depth and optical flow information. In *Intelligent Transportation Systems (ITSC), 2014 17th International IEEE Conference on*, pages 1662–1668, Oct 2014.

[51] M. Grinberg, F. Ohr, and J. Beyerer. Feature-based probabilistic data association (fbpda) for visual multi-target detection and tracking under occlusions and split and merge effects. In *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pages 1–8, Oct 2009.

[52] M. Haberjahn and M. Junghans. Vehicle environment detection by a combined low and mid level fusion of a laser scanner and stereo vision. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1634–1639, Oct 2011.

[53] I. Haller, C. Pantilie, F. Oniga, and S. Nedevschi. Real-time semi-global dense stereo solution with improved sub-pixel accuracy. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 369–376, June 2010.

[54] A. Haselhoff and A. Kummert. An evolutionary optimized vehicle tracker in collaboration with a detection system. In *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pages 1–6, Oct 2009.

[55] C. Hermes, J. Einhaus, M. Hahn, C. Wohler, and F. Kummert. Vehicle tracking and motion prediction in complex urban scenarios. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 26–33, June 2010.

[56] C. Hilario, J. Collado, J. Armingol, and A. de la Escalera. Visual perception and tracking of vehicles for driver assistance systems. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 94–99, 2006.

[57] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814 vol. 2, June 2005.

[58] C. Hoffmann. Fusing multiple 2d visual features for vehicle detection. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 406–411, 2006.

[59] L. Huang and M. Barth. Tightly-coupled lidar and computer vision integration for vehicle detection. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 604–609, June 2009.

[60] C. Idler, R. Schweiger, D. Paulus, M. Mahlisch, and W. Ritter. Realtime vision based multi-target-tracking with particle filters in automotive applications. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 188–193, 2006.

[61] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, Jan 2000.

[62] Z. Ji, M. Luciw, J. Weng, and S. Zeng. Incremental online object learning in a vehicular radar-vision fusion framework. *Intelligent Transportation Systems, IEEE Transactions on*, 12(2):402–411, June 2011.

[63] U. Kadow, G. Schneider, and A. Vukotich. Radar-vision based vehicle recognition with evolutionary optimized and boosted features. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 749–754, June 2007.

[64] B. Kitt, B. Ranft, and H. Lategahn. Detection and tracking of independently moving objects in urban environments. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1396–1401, Sept 2010.

[65] T. Kowsari, S. Beauchemin, and J. Cho. Real-time vehicle detection and tracking using stereo vision and multi-view adaboost. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1255–1260, Oct 2011.

[66] S. Krotosky and M. Trivedi. On color-, infrared-, and multimodal-stereo approaches to pedestrian detection. *Intelligent Transportation Systems, IEEE Transactions on*, 8(4):619–629, Dec 2007.

[67] S. Kubota, T. Nakano, and Y. Okamoto. A global optimization algorithm for real-time on-board stereo obstacle detection systems. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 7–12, June 2007.

[68] R. Labayrade, D. Aubert, and J.-P. Tarel. Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 2, pages 646–651 vol.2, June 2002.

[69] H. Lategahn, T. Graf, C. Hasberg, B. Kitt, and J. Effertz. Mapping in dynamic environments using stereo vision. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 150–156, June 2011.

[70] H. Lategahn, T. Graf, C. Hasberg, B. Kitt, and J. Effertz. Mapping in dynamic environments using stereo vision. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 150–156, June 2011.

[71] K. Y. Lee, J. W. Lee, and M. R. Cho. Detection of road obstacles using dynamic programming for remapped stereo images to a top-view. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 765–770, June 2005.

[72] S. Lefebvre and S. Ambellouis. Vehicle detection and tracking using mean shift segmentation on semi-dense disparity maps. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 855–860, June 2012.

[73] P. Lenz, J. Ziegler, A. Geiger, and M. Roser. Sparse scene flow segmentation for moving object detection in urban environments. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 926–932, June 2011.

[74] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 163–168, June 2011.

[75] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2953–2960, June 2009.

[76] Y.-C. Lim, C.-H. Lee, S. Kwon, and J. Kim. Event-driven track management method for robust multi-vehicle tracking. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 189–194, June 2011.

[77] Y.-C. Lim, C.-H. Lee, S. Kwon, and J.-H. Lee. Position estimation and multiple obstacles tracking method based on stereo vision system. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 72–77, June 2009.

[78] Y.-C. Lim, C.-H. Lee, S. Kwon, and J.-H. Lee. A fusion method of data association and virtual detection for minimizing track loss and false track. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 301–306, June 2010.

[79] F. Liu, J. Sparbert, and C. Stiller. Immpda vehicle tracking system using asynchronous sensor fusion of radar and vision. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 168–173, June 2008.

[80] W. Liu, X. Wen, B. Duan, H. Yuan, and N. Wang. Rear vehicle detection and tracking for lane change assist. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 252–257, June 2007.

[81] X. Liu, Z. Sun, and H. He. On-road vehicle detection fusing radar and vision. In *Vehicular Electronics and Safety (ICVES), 2011 IEEE International Conference on*, pages 150–154, July 2011.

[82] Y. Liu, B. Tian, S. Chen, F. Zhu, and K. Wang. A survey of vision-based vehicle detection and tracking techniques in its. In *Vehicular Electronics and Safety (ICVES), 2013 IEEE International Conference on*, pages 72–77, July 2013.

[83] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint*

Conference on Artificial Intelligence - Volume 2, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[84] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[85] S. Matzka and R. Altendorfer. A comparison of track-to-track fusion algorithms for automotive sensor fusion. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 189–194, Aug 2008.

[86] S. Matzka, A. Wallace, and Y. Petillot. Efficient resource allocation for attentive automotive vision systems. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):859–872, June 2012.

[87] M. Mahlisch, R. Schweiger, W. Ritter, and K. Dietmayer. Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 424–429, 2006.

[88] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2259–2272, Nov 2011.

[89] C. Mikolajczyk and A. Zisserman. Human detection based on probabilistic assembly of robust part detectors. *ECCV*, pages 69–82, 2004.

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

[91] J. Morat, F. Devernay, and S. Cornou. Tracking with stereo-vision system for low speed following applications. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 955–961, June 2007.

[92] H. Niknejad, A. Takeuchi, S. Mita, and D. McAllester. On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):748–758, June 2012.

[93] M. Nishigaki, S. Rebhan, and N. Einecke. Vision-based lateral position improvement of radar detections. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 90–97, Sept 2012.

[94] J. Nuevo, I. Parra, J. Sjoberg, and L. Bergasa. Estimating surrounding vehicles' pose using computer vision. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1863–1868, Sept 2010.

[95] U. Nunes, M. Trivedi, and C. Laugier. Introducing perception, planning, and navigation for intelligent vehicles. *Intelligent Transportation Systems, IEEE Transactions on*, 10(3):375–379, September 2007.

[96] F. Oniga and S. Nedevschi. Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. *Vehicular Technology, IEEE Transactions on*, 59(3):1172–1182, March 2010.

[97] C. Pantilie and S. Nedevschi. Real-time obstacle detection in complex scenarios using dense stereo vision and optical flow. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 439–444, Sept 2010.

[98] C. Papageorgiou and T. Poggio. Trainable pedestrian detection. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 4, pages 35–39 vol.4, 1999.

[99] M. Perrollaz, A. Spalanzani, and D. Aubert. Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 313–318, June 2010.

[100] M. Perrollaz, J.-D. Yoder, A. Negre, A. Spalanzani, and C. Laugier. A visibility-based approach for occupancy grid computation in disparity space. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):1383–1393, Sept 2012.

[101] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto. A lidar and vision-based approach for pedestrian and vehicle detection and tracking. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 1044–1049, Sept 2007.

[102] A. Prioletti, P. Grisleri, M. Trivedi, and A. Broggi. Design and implementation of a high performance pedestrian detection. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 1398–1403, June 2013.

[103] C. Rabe, U. Franke, and S. Gehrig. Fast detection of moving objects in complex scenarios. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 398–403, June 2007.

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

[105] S. Rodriguez F, V. Fremont, P. Bonnifait, and V. Cherfaoui. Visual confirmation of mobile objects tracked by a multi-layer lidar. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 849–854, Sept 2010.

[106] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[107] U. Scheunert, H. Cramer, B. Fardi, and G. Wanielik. Multi sensor based tracking of pedestrians: a survey of suitable movement models. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 774–778, June 2004.

[108] R. Schubert, G. Wanielik, and K. Schulze. An analysis of synergy effects in an omnidirectional modular perception system. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 54–59, June 2009.

[109] T. Shen, G. Schamp, T. Cooprider, and F. Ibrahim. Stereo vision based full-range object detection and tracking. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 925–930, Oct 2011.

[110] S. Sivaraman, B. Morris, and M. Trivedi. Learning multi-lane trajectories using vehicle-based vision. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2070–2076, Nov 2011.

[111] S. Sivaraman and M. Trivedi. A general active-learning framework for on-road vehicle recognition and tracking. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):267–276, June 2010.

[112] S. Sivaraman and M. Trivedi. Combining monocular and stereo-vision for real-time vehicle ranging and tracking on multilane highways. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1249–1254, Oct 2011.

[113] S. Sivaraman and M. Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *Intelligent Transportation Systems, IEEE Transactions on*, 14(4):1773–1795, Dec 2013.

[114] D. Smith and S. Singh. Approaches to multisensor data fusion in target tracking: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 18(12):1696–1710, Dec 2006.

[115] G. Stein, Y. Gdalyahu, and A. Shashua. Stereo-assist: Top-down stereo for driver assistance systems. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 723–730, June 2010.

[116] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5):694–711, May 2006.

[117] A. Takeuchi, S. Mita, and D. McAllester. On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 1014–1021, June 2010.

[118] A. Talukder, R. Manduchi, A. Rankin, and L. Matthies. Fast and reliable obstacle detection and segmentation for cross-country navigation. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 2, pages 610–618 vol.2, June 2002.

[119] Y. Tan, F. Han, and F. Ibrahim. A radar guided vision system for vehicle validation and vehicle motion characterization. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 1059–1066, Sept 2007.

[120] G. Toulminet, M. Bertozzi, S. Mousset, A. Bensrhair, and A. Broggi. Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis. *Image Processing, IEEE Transactions on*, 15(8):2364–2375, Aug 2006.

[121] M. Trivedi, T. Gandhi, and J. McCall. Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *Intelligent Transportation Systems, IEEE Transactions on*, 8(1):108–120, March 2007.

[122] W. van der Mark, J. van den Heuvel, and F. Groen. Stereo based obstacle detection with uncertainty in rough terrain. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 1005–1012, June 2007.

[123] A. Vatavu, R. Danescu, and S. Nedevschi. Real-time dynamic environment perception in driving scenarios using difference fronts. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 717–722, June 2012.

[124] A. Vatavu, R. Danescu, and S. Nedevschi. Stereovision-based multiple object tracking in traffic scenarios using free-form obstacle delimiters and particle

filters. *Intelligent Transportation Systems, IEEE Transactions on*, 16(1):498–511, Feb 2015.

[125] A. Vatavu and S. Nedevschi. Real-time modeling of dynamic environments in traffic scenarios using a stereo-vision system. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 722–727, Sept 2012.

[126] P. Viola and M. Jones. Robust real-time face detection. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 747–747, 2001.

[127] A. Wedel and U. Franke. Monocular video serves radar-based emergency braking. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 93–98, June 2007.

[128] J. Woodlill, R. Buck, D. Jurasek, G. Gordon, and T. Brown. 3d vision: Developing an embedded stereo-vision system. *Computer*, 40(5):106–108, May 2007.

[129] S. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath. Collision sensing by stereo vision and radar sensor fusion. *Intelligent Transportation Systems, IEEE Transactions on*, 10(4):606–614, Dec 2009.

[130] Y. Wu and T. Yu. A field model for human detection and tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5):753–765, May 2006.

[131] K. Yamaguchi, T. Kato, and Y. Ninomiya. Moving obstacle detection using monocular vision. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 288–293, 2006.

[132] K. Yamaguchi, T. Kato, and Y. Ninomiya. Vehicle ego-motion estimation and moving object detection using a monocular camera. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 610–613, 2006.

[133] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), Dec. 2006.

[134] Y. Zhu, D. Comaniciu, V. Ramesh, M. Pellkofer, and T. Koehler. An integrated framework of vision-based vehicle detection with knowledge fusion. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 199–204, June 2005.

Thanks

My thanks go to my family and all the people who helped me in studies. I want to thank also the people of the Vislab and the professors Alberto Broggi and Pietro Cerri.