



Information and Communication Technologies

Collaborative Project

“V-Charge”

Autonomous Valet Parking and Charging for e-Mobility

Grant Agreement Number 269916



Deliverable Number	D3.4
Deliverable Name	Grid- and Object-based Data Fusion Concept
Dissemination Level	Restricted (Expected to be Public on 2014/07/15)
WP Number	WP 3
Date of Preparation	01/06/2014
Date of Submission	13/06/2014
Editor/Beneficiary	S. Wonneberger / Volkswagen AG

Executive summary

This deliverable reports on the progress within work package 3 of the V-Charge Project for Autonomous Valet Parking and Charging for e-Mobility. The main objective of the overall work package is to develop and test novel approaches towards online, robot-local localization and environment perception using camera-based, ultrasonic and probably radar based sensors. The sensors are selected with an eye kept to a resulting close-to-market setup.

This deliverable is focusing the online environment perception aspect of work package 3 while the parallel published deliverable 3.3 is about the innovative vision based localization approach within V-Charge.

In detail the deliverable addresses the following tasks from the description of work:

“Task 3.2: Online robot-centric depth-maps of the surrounding (RBG (lead), ETHZ)

Within this task, the depth maps provided from the front and rear stereo cameras, as well as the depth maps computed from multi-view stereo of the side cameras are fused into a robot-centric local 3D map of the surrounding. To this end, we will develop a multi-view stereo algorithm for side cameras which is able to deal with dynamic objects in the images. Furthermore, we will develop a module that fuses all depth maps and merges them into a 2D grid map for obstacle avoidance and path planning.”

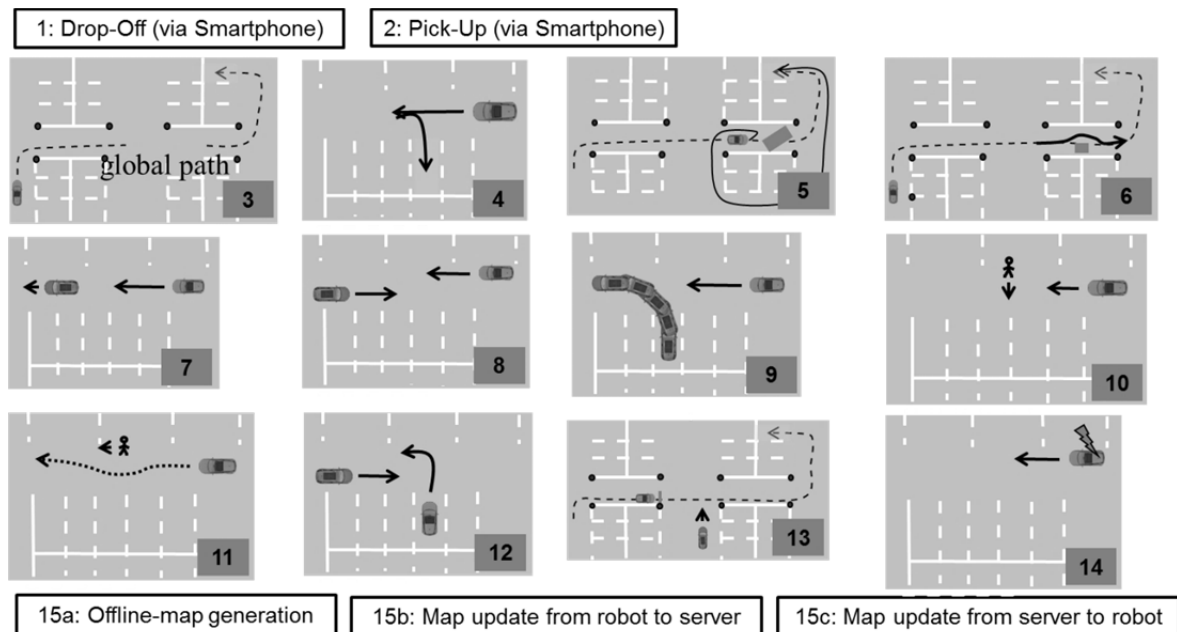
“Task 3.3: Detection and tracking of dynamic objects (UNIPR (lead), ETHZ, VW)

To detect dynamic objects in the environment, we will employ different strategies depending on the position of the objects in relation to the vehicle. For objects in the front and in the rear, the information obtained from stereo camera systems will be used. Then, based on the 3D features extracted in Task 2.2, the discrimination between static and dynamic objects will be performed. Furthermore, we will investigate tracking and motion stereo approaches to obtain a list of areas of interest prior to classification. For objects on the sides of the vehicle, motion stereo will be used to detect motions towards the vehicle. Given the vehicle ego-motion, the motion map will be used here to find dynamic objects. Again, a previous area-of-interest detector will be employed. Short term predictions of the future motion of identified dynamic objects will be generated through interaction models.”

“Task 3.4: Object-Classification (UNIPR (lead), RBG, ETHZ)

In the past, many different approaches have been presented to classify dynamic objects such as cars and pedestrians. However, the performance of these approaches is hard to quantify as the data used for evaluation varies strongly over the reported approaches. Also, the suitability of these methods for this project is hard to estimate beforehand due to the particular environment conditions (lighting, motion speed, etc.). Therefore, instead of investigating/developing a new approach, we will implement and adapt state-of-the-art techniques and combine them into a voting classification approach. Then, from a set of classification hypotheses obtained from the votes, a validation step will be used for the final classification.”

The project focuses a lot of scenarios with special focus and required impact in online-robot centric environment perception. The following figure shows the proposed scenarios from the consortium.



While scenarios 3 to 6 have a special focus on robust and accurate static environment reconstruction, the scenarios 7 to 13 require a detection and correct classification and tracking of relevant road users typically found on parking lots to achieve a natural integration of the v-charge research prototype in the normal traffic flow.

In the previous deliverable 3.2 a main focus has been spent on the first four scenarios building a basis for the more complex dynamic objects which are especially addressed in this deliverable. The contributions in this deliverable have been structured into logical sections of the overall system, starting with an update on the sensors and their current capabilities (Section 1). Here a special contribution has been made by the partner Bosch with the introduction of a rear-view Stereo Camera System capable to reconstruct a parking spot at a very high detail level

while at the same time provide a possibility to cluster and track dynamic objects such as pedestrians spontaneously walking in between car and parking spot (Section 2.3).

The next level of algorithm grouping is described as Sensor Preprocessing (Section 2) where camera and ultrasonic sensor raw data is processed sample by sample to extract relevant information on a per measurement level. Here major contributions are reported on all main algorithm components. ETH Zurich Computer Vision Group gives an in detail view of the Motion Stereo system, processed on the side cameras to achieve a dense and accurate reconstruction of the vehicle side area especially relevant when reconstructing the parking areas while passing parked cars (Section 3.2). The University of Parma / the VisLab Team gives a major contribution towards the scenarios 7, 8 and 10-13 (Section 3.3). In order to cope with pedestrians the detection framework has been extended for those objects. In addition solutions utilizing the full fisheye-cameras FoV to reconstruct objects at a better depth level are proposed, especially targeting the overlapping field-of-views of the cameras. In addition a detection of parking spots from mono vision is discussed. Special notes on the demonstration at Stuttgart Airport, April 2014 are given. Bosch provides a new Ultrasonic Processing System, in order to provide measurements with a higher spatial resolution, already triangulating single measurements on a raw signal level and providing data on an accurate point and line / wall representation level (Section 3.4).

The fusion logic of the intermediate sensor data is part of Section 4, where Volkswagen discusses new contributions to the static grid map fusion module and object tracking and fusion module. Special notes are also given here regarding the already mentioned Stuttgart Demo. As an overall result all teams are getting closer together and the objects and obstacles exchanged are better processed on all levels.

Finally the data needs to be provided and interpreted by the path planning, local navigation and obstacle / object avoidance algorithms as well as the parking algorithm. This scene interpretation part is grouped in Section 5. Here Volkswagen mentions the Road Graph Data Structure to combine local perception with the global localization and available semantic information of the parking area (Section 5.2). ETH Zürich discusses the usage and interpretation of tracked objects to achieve a robust prediction and thus natural driving maneuvers (Section 5.3). Finally, solutions to model based extraction of parking slots are presented by Bosch (Section 5.5) and Volkswagen (Section 5.4).

In a final conclusion (Section 6) a short view towards the remaining time in the V-Charge project and remaining sub-tasks and scenarios is given.

Table of Contents

1	Introduction	7
1.1	Perception Architecture	7
2	Sensors & Hardware	8
2.1	Vehicle Overview.....	8
2.2	Front Mid-Range Stereo-Camera.....	10
2.3	Rear Wide-Angle Fish-Eye Stereo Camera.....	11
2.4	Top-View Fisheye Mono Camera Rig.....	12
2.5	Ultrasonic Sensors	13
2.6	Processing Hardware	14
3	Sensor Preprocessing	15
3.1	Stereo Processing	15
3.2	Dense Motion Stereo.....	18
3.2.1	Depth Map Computation	19
3.2.2	Obstacle extraction	19
3.2.3	Results.....	20
3.3	Image-based Object Detection and Classification	22
3.3.1	Dynamic Object Detection - Roadmap.....	22
3.3.2	Multi View Detection	24
3.3.3	Pedestrian Detection.....	25
3.3.4	Cylindrical Images.....	29
3.3.5	Sparse Stereo Matching	30
3.3.6	Parking Spot Detection	33
3.3.7	Object-hypothesis Classification	34
3.4	Ultrasonic Triangulation.....	35
4	Sensor Data Fusion	35
4.1	Occupancy Grid Map-based Static Obstacle Fusion	35
4.1.1	Sensors and Sensor Models.....	36
4.1.2	Sensor Fusion.....	37
4.1.3	Compensation of Dynamic Objects.....	38
4.2	Dynamic Object Tracking & Fusion	39
4.2.1	Sensor Setup	39

4.2.2	Sensor fusion concept	40
5	Scene Interpretation & Information Extraction.....	44
5.1	Perception Interface for Path Planning & Parking	44
5.2	Roadgraph-based Scene Interpretation.....	45
5.3	Local Motion Planning in the Presence of Moving Objects.....	47
5.4	Parking-Spot Finder.....	48
5.5	Parking-Spot Tracking and Parking Position Refinement.....	49
6	Conclusion and Outlook.....	50
7	References.....	51

1 Introduction

V-Charge is a collaborative project to investigate and provide a profound impact on the state-of-art in automated driving in low-speed environments such as parking lots combined with the idea of an automatic valet parking and charging system for tomorrows E-Mobility life.

In combination of very restricted infrastructure integration and interaction this goal shall be achieved with entry-level, low-cost sensors for the task of online perception (Work Package 3). Only camera sensors and ultrasonic systems shall be used which are already available or expected to be available in series-production A/B-class car segment by end of the project.

This deliverable describes the current work in the field of sensors, sensor acquisition (Section 2), image and signal data processing (Section 3), sensor data fusion (Section 4) and scene interpretation (Section 5).

1.1 Perception Architecture

Stefan Wonneberger, Volkswagen AG

The perception architecture has been defined in clear modules each assigned to a specific partner or group of partners to work together. Figure 1 gives a brief overview of the complete sensor pre-processing & perception system in V-Charge.

Sensors may be shared between preprocessing modules. Different communication subsystems such as CAN, Ethernet-based Real-Time Data Transmission using the Data Distribution Service [1] Implementation of Company RTI as well as Inter/Intra-Process Communication using two main Frameworks [2], [3] ensure a real-time delivery of sensor and intermediate data to these and between those modules.

The architecture aims to provide basic semantic scene objects defined as

- Classified (Mainly Dynamic) Objects such as Pedestrians, Cars, Bicyclists the system has to cope with using the object class (*Object Fusion Module*)
- Unstructured Obstacles with no classification label but relevant for collision avoidance and further scene analysis – such as parking spot detection etc. (*Grid Fusion Module*)
- Road Network allowing a global navigation planning and defining areas where to drive, where to park and where to look especially for other road users (*Road Graph Module*)

Since the last module, the Road Graph is gathered from an infrastructure server in the V-Charge system proposal it is not furthered targeted in this report. The interested reader may refer to Work Packages 2 and 5.

In order to make these different modules transparent to the subsequent path planning, navigation and collision avoidance modules the main goal is to define an

object only in one representation. An object is either hold in object fusion or grid fusion, not both modules. Thus a classification of preprocessed data has to be performed. This concept is shown in Figure 1. Finally those objects may be related to the road graph to achieve a single point of access to the scene reconstruction and a combination of additional a-priori knowledge of the scene. This is especially relevant for the path planning algorithms while driving to the parking area, allowing for example an object prediction using lane information. As the parking maneuver is relying on a highly accurate contour reconstruction, here the grid fusion module is primarily processing all relevant sensor data. Dynamic Objects may be classified and used separately e.g. for maneuver interruption etc.

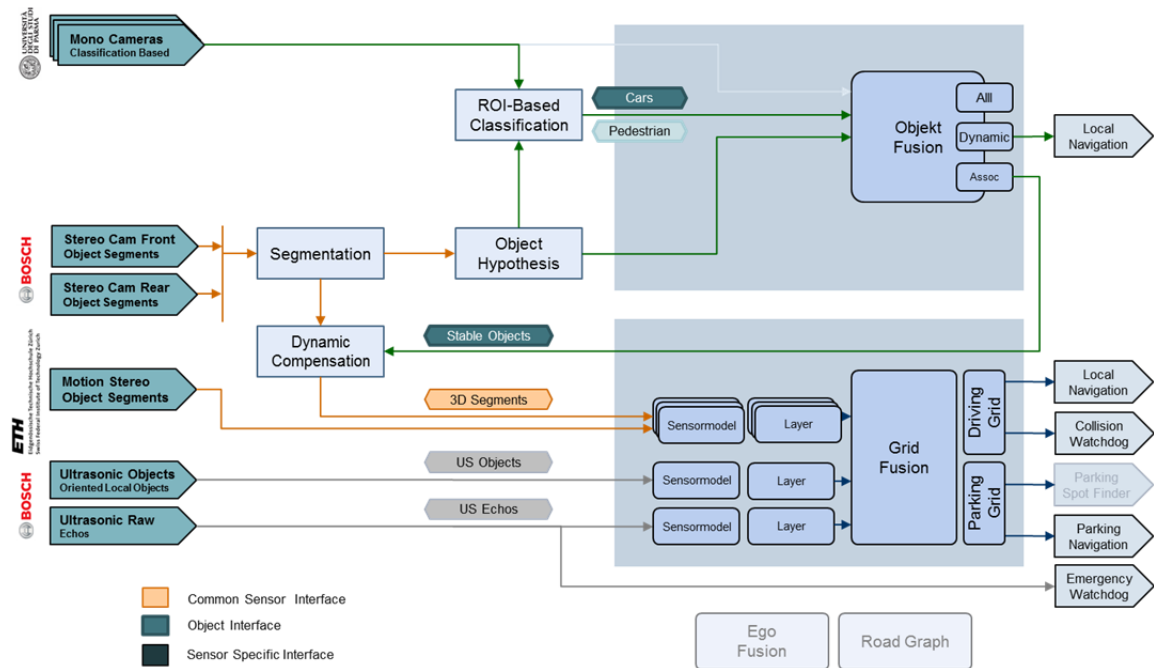


Figure 1: Perception Architecture showing modules per partner and data paths

2 Sensors & Hardware

This section gives an overview of the installed basic sensor hardware, camera and ultrasonic, and its detailed specification such as sample rate or field of view. The pre-processing of the sampled data towards a 3D Reconstruction or Object Detection is described in the next section.

2.1 Vehicle Overview

Stefan Wonneberger, Volkswagen AG

The current research vehicles available for evaluation are two Volkswagen Golf Mark VI both equipped with an identical sensor system. Figure 2 shows one of them uniquely defined by different colors – blue and green.

The sensors have been integrated with respect to keep the typical outlook of the series version. Figure 3 shows the mounting positions of the sensors. The car is equipped with four sensor arrays:

- Stereo Front Camera Rig
- Stereo Rear Wide-Angle Fisheye Camera Rig
- Quad Mono Wide-Angle Fisheye Camera Rig, typically used for Top View
- Ultrasonic Sensor Array



Figure 2: One of the two V-Charge Research Vehicles

Also the four main Sensor Preprocessing Modules are assigned to the sensors:

- Dense Stereo Disparity Map Generation and 3D Segment Clustering
- Dense Motion Stereo and Disparity Map Generation on Mono Cameras as well as 3D Segment Clustering
- Object Detection on Fisheye Cameras classifying Vehicles and Pedestrians in near surroundings
- Ultrasonic System providing raw echoes as well as triangulated obstacle points

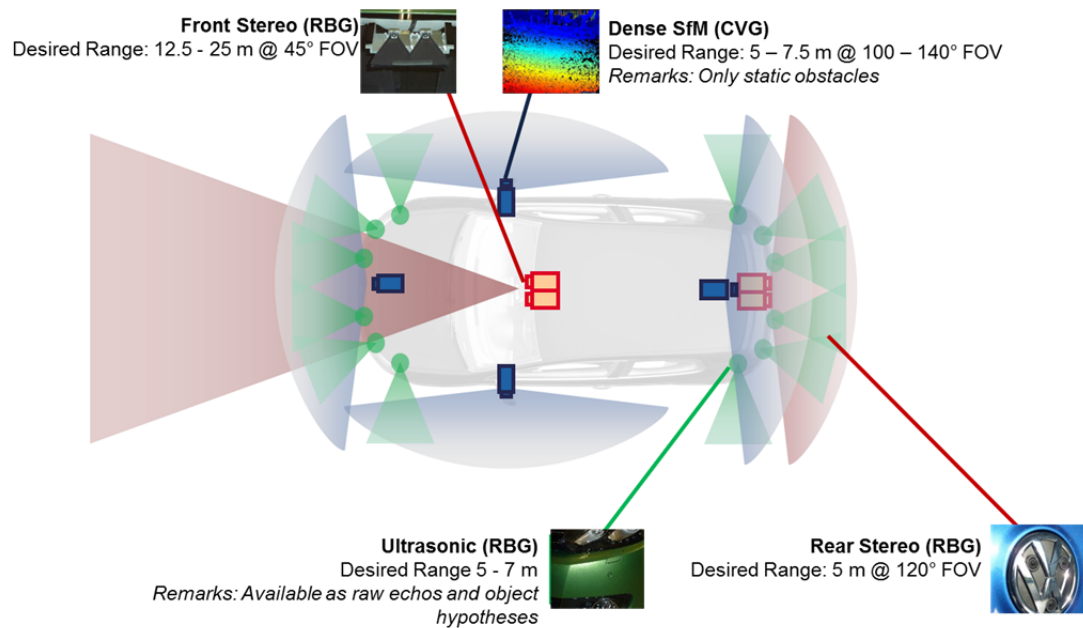


Figure 3: Sensors and Mounting Position,
also shown the processing algorithms and basic key facts (FoV, Range)

The sensors are described within the next subsections. The Preprocessing is described separately in the next section.

2.2 Front Mid-Range Stereo-Camera

Keiji Mimura, Robert Bosch GmbH

A front stereo-camera is installed in the front to detect objects (Figure 4Figure 19). Its range is 45 m and field-of-view (FoV) is 45 degree (Figure 5). There are two operation modes for the front stereo-camera:

- Deliver distance data readings + left reference image
- Deliver left and right reference images

These two operation modes are switchable online per software. Reference images have resolution of 512 x 219 Pixel. Stereo-cameras (front and rear) deliver distance data readings at 12.5 [Hz].

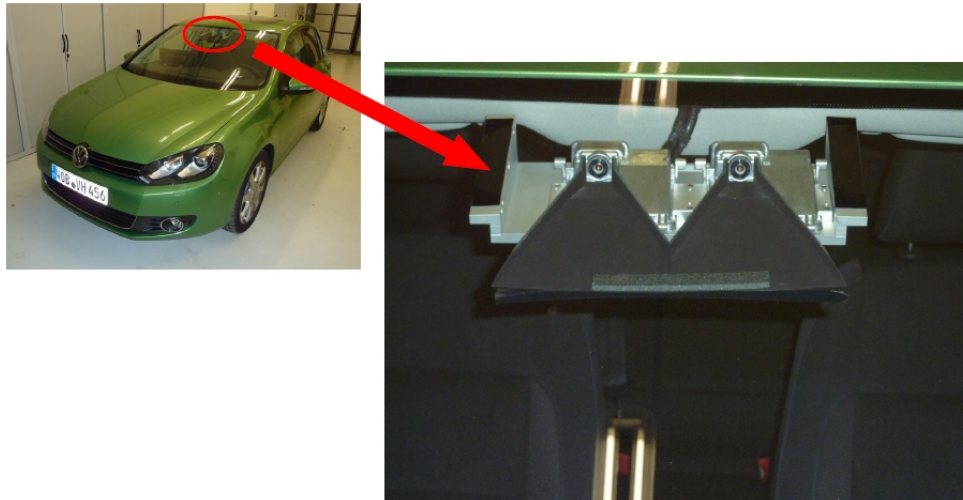


Figure 4: Stereo-Camera Front

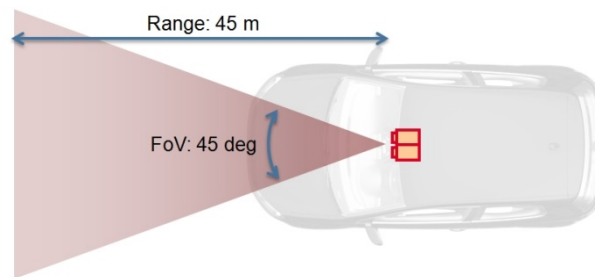


Figure 5: Range and Field-of-View (FoV) of Stereo-Camera Front

2.3 Rear Wide-Angle Fish-Eye Stereo Camera

Keiji Mimura, Robert Bosch GmbH

A prototype rear stereo-camera module (Figure 6, left and middle) is installed in the rear. The lenses for the rear stereo-camera are same fisheye lenses as ones for the mono cameras. Its stereo baseline is 5 cm so that the module can be integrated into the VW-logo in the rear (Figure 6, right). In order not to make the overall system complex, the mono cameras and the rear stereo-camera remain separate sub-systems. As a consequence, there are 3 lenses integrated in the rear VW-logo: 1 lens for the mono cameras and 2 for the rear stereo-camera. This new rear stereo-camera is connected to the same PC as the front stereo-camera which results in no additional components except the stereo-camera module itself. Its range is 5 m and field-of-view (FoV) for the stereo-camera function is 120 [deg] (Figure 7).

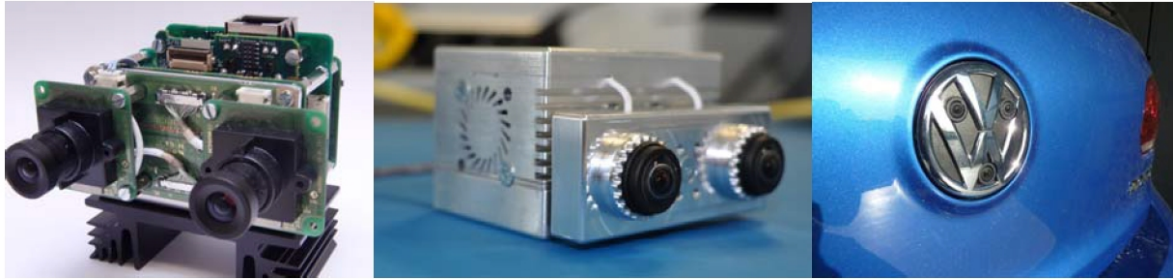


Figure 6: Stereo-Camera Rear

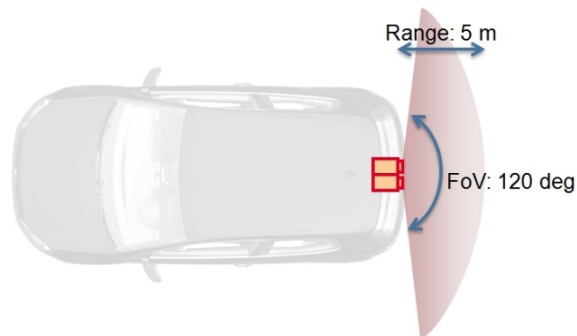


Figure 7: Range and Field-of-View (FoV) of Stereo-Camera Rear

2.4 Top-View Fisheye Mono Camera Rig

Keiji Mimura, Robert Bosch GmbH

Four fisheye cameras are installed in the front VW-logo, in the rear VW-logo and in both side mirrors (Figure 8). They have a nominal FoV of 185 degree and 1.3 Megapixel resolution and synchronously triggered at 12.5 Hz. Partners have access to raw image data.



Figure 8: Mono Cameras

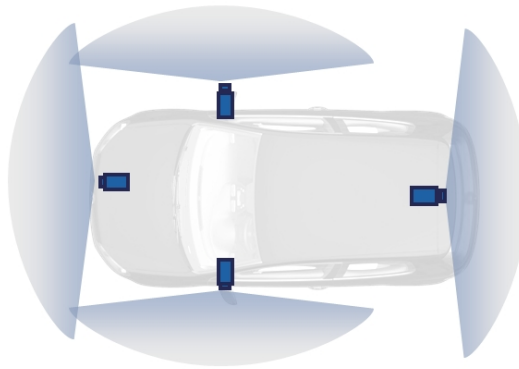


Figure 9: Positions and Coverage of Mono Cameras

2.5 Ultrasonic Sensors

Keiji Mimura, Robert Bosch GmbH

Twelve ultrasonic sensors are installed in front-/rear-bumpers (6 in the front, 6 in the rear) (Figure 10) to detect objects in short range. Each sensor has an aperture angle of horizontally 120 degree and vertically 60 degree and a range up to 4.2 m (Figure 11). Partners have access to raw distance readings and preprocessed object distance readings.



Figure 10: Ultrasonic Sensors

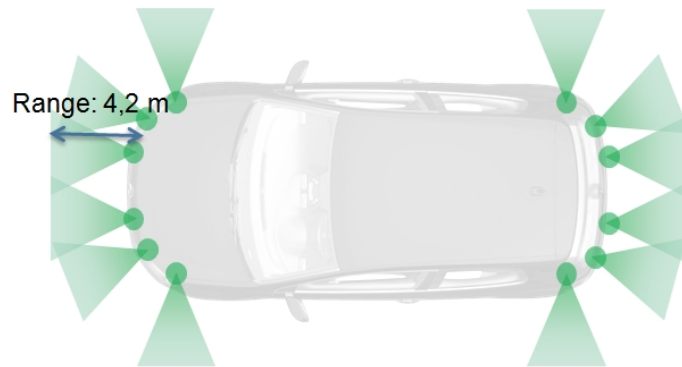


Figure 11: Range and Field-of-View (FoV) of Ultrasonic Sensors

2.6 Processing Hardware

Stefan Wonneberger, Volkswagen AG

For the data processing of the raw sensor data a distributed computing system has been installed in the trunk of the vehicle shown in Figure 12.



Figure 12: Computing system installed in vehicle trunk, 6 Rack PCs, two separated computing systems for mono and stereo processing

Overall there are 8 Computing nodes installed. Six Industrial PC Low-Size Low-Power Nodes having 2 Cores with 2.6 GHz and 8 GB of Memory. In addition two high-performance computing systems one designated for Mono Image Processing including a GPU for Dense Motion Stereo Matching and one for Stereo Processing allowing a fast adaptation of the Matching Algorithm allowing a specific adaptation towards the parking scenarios. The processing nodes have been partitioned as follows

- *Industrial PC 1:*
CAN-Communication Gateway and Real-Time Middleware Control
- *Industrial PC 2:*
Motion Control, Local Path Planning for Driving Maneuvers
- *Industrial PC 3:*
Mission Executive & Supervisor, Local Planning for Parking, Global Navigation Planning on Road Graph
- *Industrial PC 4:*
Static Grid Fusion and Dynamic Object Tracking & Fusion
- *Industrial PC 5:*
Image-based Object Detection and Classification
- *Industrial PC 6:*
<Backup>
- *Mono Image Processing PC*
Visual Localisation (on CPU)
Dense Motion Stereo (on GPU)
- *Stereo Image Processing PC*
Disparity Map Estimation for Front and Rear Stereo + 3D Segment Clustering

3 Sensor Preprocessing

Since the previous section lists the basic aspects of the sensor hardware, this section provides a detailed view on the preprocessing such as the image processing algorithms and signal analysis of the ultrasonic systems.

3.1 Stereo Processing

Keiji Mimura, Robert Bosch GmbH

Stereo-cameras (front and rear) deliver up to several hundreds of preprocessed object distance readings, together with additional attributes like height over ground or instantaneous motion direction (Figure 13). The rear stereo-camera is conceptually similar to the front stereo-camera. Main difference is its wide FoV. As lenses with wide FoV used in the rear stereo-camera, necessary adaptations to manage visual distortions are included in the processing for the rear stereo-camera.



Figure 13: Example of Objects detected by Stereo-Camera Front

Both of front / rear stereo-cameras are connected to the same PC. A processing pipeline of either front or rear stereo-camera is selected through a DDS signal. The DDS signal is set based on a position of a gear-selector. Then, the front stereo-camera is selected for driving forward and the rear stereo-camera is engaged when the gear-selector is set in reverse (Figure 14).

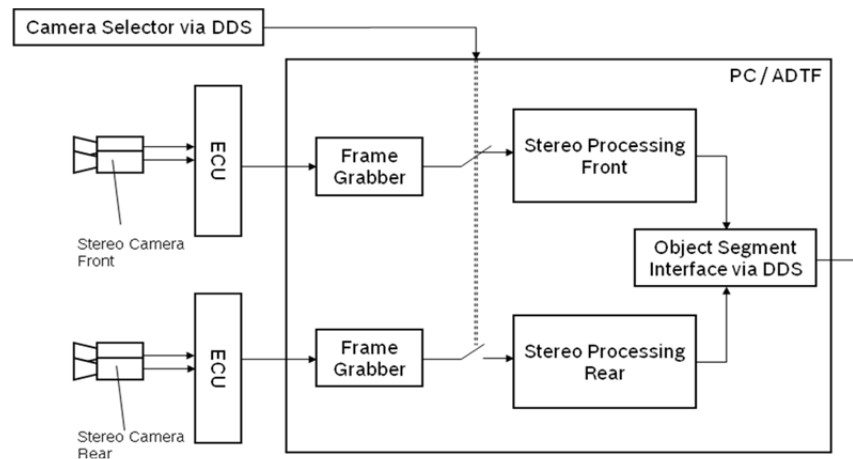


Figure 14: Architecture of Stereo-Camera System with Driving-Direction dependent Selector

Some examples of a grid-map based on front or rear stereo-camera are shown in Figure 15 – Figure 18. The scene for these examples is: an ego-car is driving on a lane in a parking garage and approaching to a target parking-spot → then, the car parks into the target parking-spot backwards (Figure 18). A grid-map based on the front stereo-camera during driving on a lane is shown in Figure 16. Then, the car changes its driving-direction to park backwards. When the gear-selector is set to reverse, then the rear stereo-camera is activated. Some grid-maps based on the rear stereo-camera during parking are shown in Figure 17. Just for information, an example of a fused grid-map based on the front and rear stereo-camera during parking is shown in Figure 18. A method to combine inputs from various sensors is mentioned in another section.

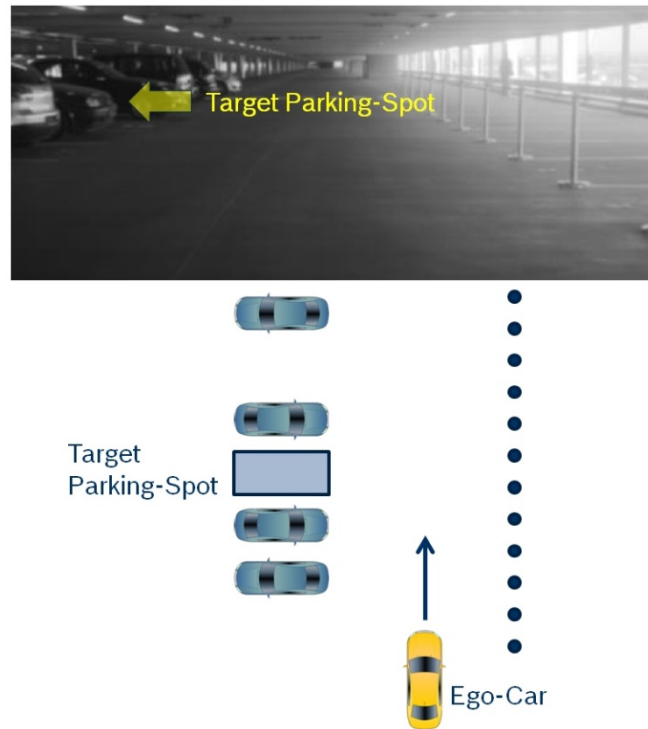


Figure 15: An Example Scene: Approaching to a Parking-Spot and Parking

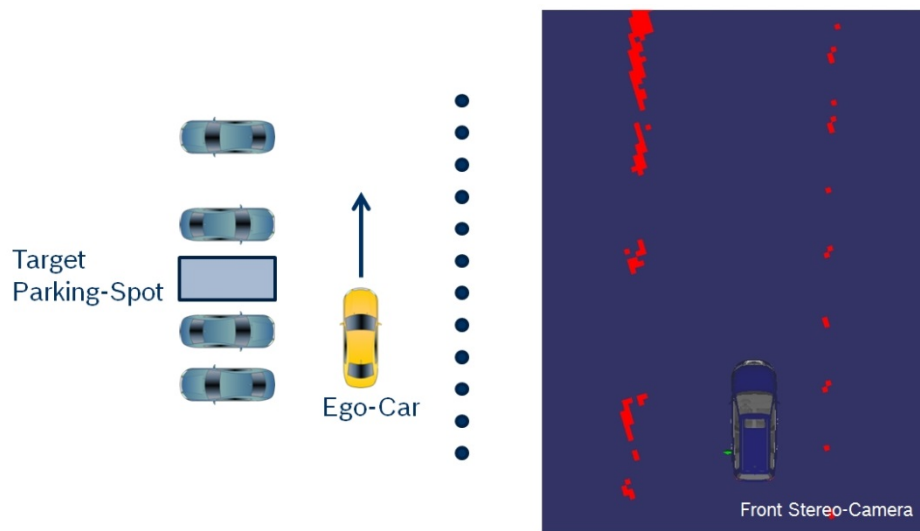


Figure 16: An Example of a Grid-Map based on the Stereo-Camera Front

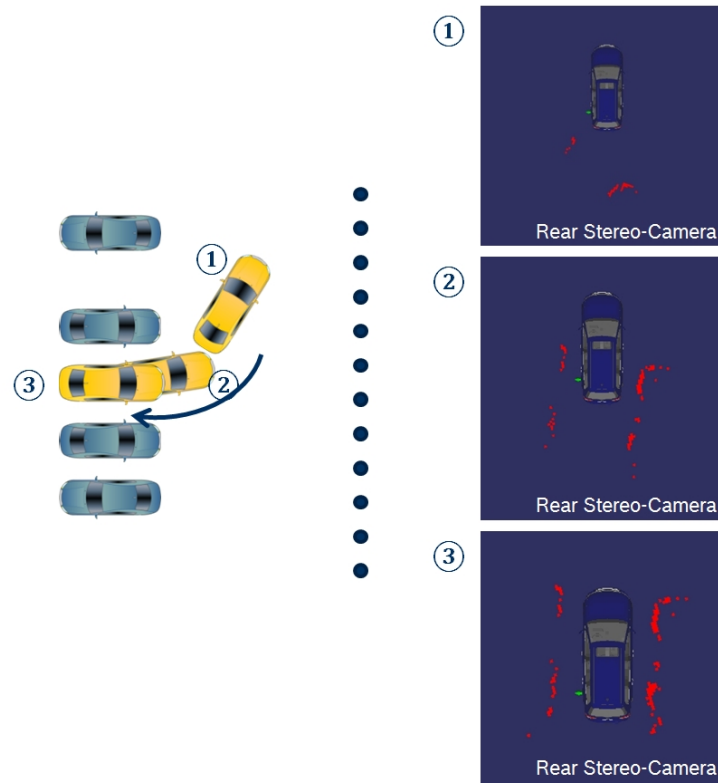


Figure 17: An Example of a Grid-Map based on the Stereo-Camera Rear

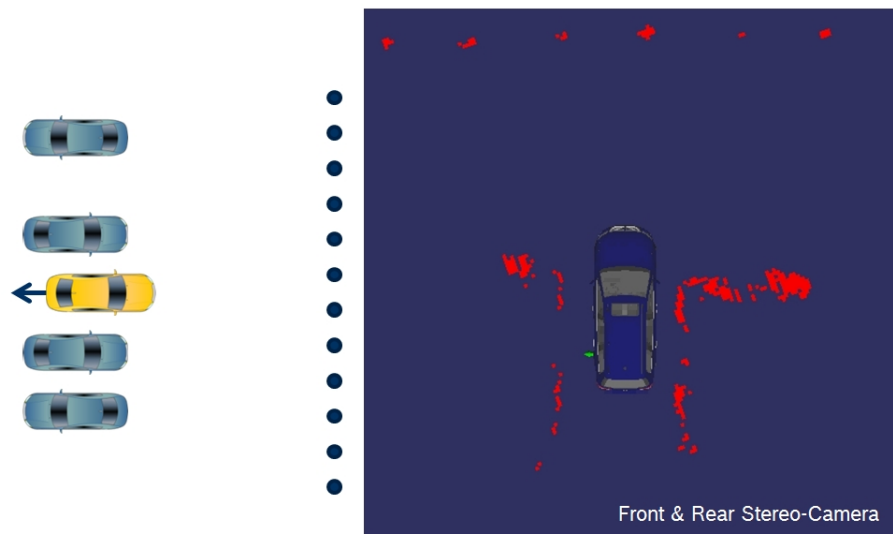


Figure 18: An Example of a Fused Grid-Map based on the Stereo-Camera Front & Rear

3.2 Dense Motion Stereo

Christian Häne, Computer Vision and Geometry Group, ETH Zurich

The goal of our online motion stereo module is the computation of obstacle maps given the mono fisheye camera images and the vehicle poses from the wheel odometry.

Our system works by first computing depth maps and then subsequently analyze the depth maps to find the relevant obstacles. In order to compute depth maps we

need several images of the same objects with a slightly different known viewpoint. We achieve this by treating each camera independently and using several subsequent camera images while the car moves. The camera pose is computed by applying the extrinsic camera calibration on the vehicle pose from the wheel odometry.

3.2.1 Depth Map Computation

There is a vast literature on dense stereo matching from images for example [4]. One especially suitable approach for our purpose is the plane sweeping algorithm [5]. It does not require stereo rectification of the images and is capable of directly matching more than two images without the need of a special camera configuration. This algorithm can be implemented in real-time on the GPU [6], [7] .

The algorithm works as follows. One image is chosen as a reference image for which a depth map is computed. A plane is swept through space, which serves as a local proxy geometry. For each pixel it is evaluated whether the proxy plane matches to the observed geometry. In order to do this efficiently for each plane position all the images are projected onto the proxy plane and back into the reference view. This two projection steps can be done at once by applying a planar homography. The warped images are then compared to the reference image using an image dissimilarity measure in order to evaluate, if the given proxy plane locally, per pixel, matches the observed geometry.

Currently we first extract pinhole images with a field of view of 129.8° horizontally and 106.3° vertically, out of the fisheye images. The pinhole images and the camera poses from the wheel odometry are then fed into our Nvidia CUDA implementation. We achieve real-time performance for the 12.5Hz capturing rate on the left and right camera simultaneously.

In the future we plan to change to stereo matching directly on the fisheye images to extend the field of view.

3.2.2 Obstacle extraction

Depth maps computed by stereo matching often contain a lot of noise. Therefore we need a way to robustly extract strong obstacles out of the noisy depth maps. We do this by building an occupancy grid and then check each ray for a strong obstacle within the occupancy grid.

The environment is described by a segmentation of the space, into an area where the car can move, called free space, and obstacles that stand on the ground (see for example [8]). For this representation we need to define the ground plane as a first step. In our current system we extract the ground plane based on the extrinsic calibration of the 4 fish-eye cameras. The position of each camera in the vehicle coordinate system directly gives us a good estimate for the ground plane with respect to the local camera coordinate frame.

Having the ground estimate we generate a 2D occupancy grid for each camera independently. This is done by projecting each 3D point that was reconstructed in the depth map into the 2D occupancy grid and check if it lies close to the ground or above the ground. If a point lies close to the ground we consider it as lying on the ground plane and hence add a negative weight to the occupancy grid cell. If the point lies above the ground we add a positive weight to the grid indicating that we saw an obstacle. To be able to handle overhanging structures we ignore points above a certain height over ground.

After having filled in the occupancy grid we try to find an obstacle for each ray. We do this by minimizing an energy over the ray which gives us the most likely obstacle position for each ray. The energy is given as

$$E(z) = \sum_{i=0 \dots z} \text{occupancy}(i) - \sum_{i=z \dots z+d} \text{occ}(i),$$

where d is a parameter which is the expected thickness of an obstacle. Finding the distance z which minimizes the energy can be done quickly in linear time by an exhaustive search over all the discretized depth values. As there is often no strong obstacle observed, we only consider obstacles which have a sufficiently low energy.

3.2.3 Results

In this section we show some results of our dense online motion stereo module. The extracted pinhole images have a resolution of 640x400 pixels and an opening angle of 129.8° in horizontal direction and 106.3° in vertical direction, respectively. The plane sweep is done using three consecutive images with the middle one as reference. As image dissimilarity we use a ZNCC matching score window. To be robust against occlusions we also use half sequence occlusion handling.

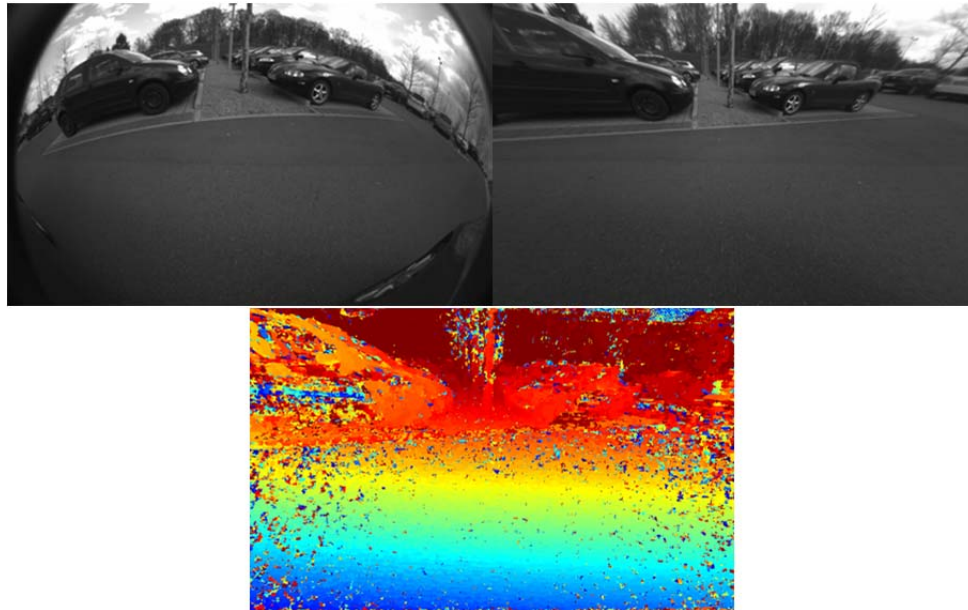


Figure 19: Original Image, Rectified Image, Resulting Depth Map

The following images show the depth map on the upper left, a depth map with the ground removed on the upper right. The lower half shows on the left hand side the occupancy grid and one of the input images and on the lower right hand side the extracted obstacles in 3D.

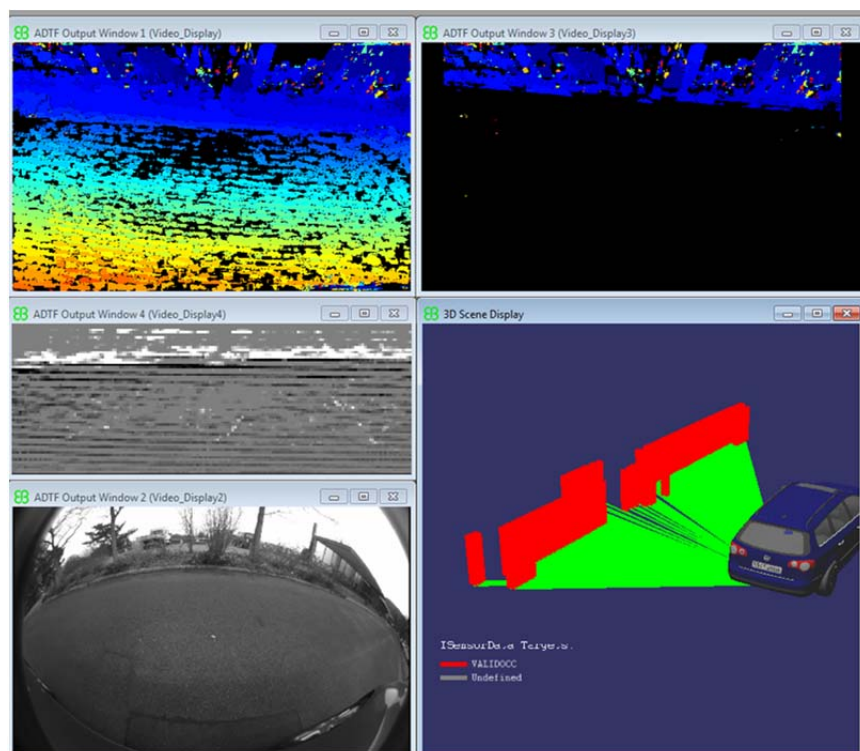


Figure 20: Computation of Stixel-based 2D Segmentation and 3D Position Estimation

3.3 Image-based Object Detection and Classification

Main activities during the third year in Work Package 3, **Task 3.3 “Detection and tracking of dynamic objects”** have been:

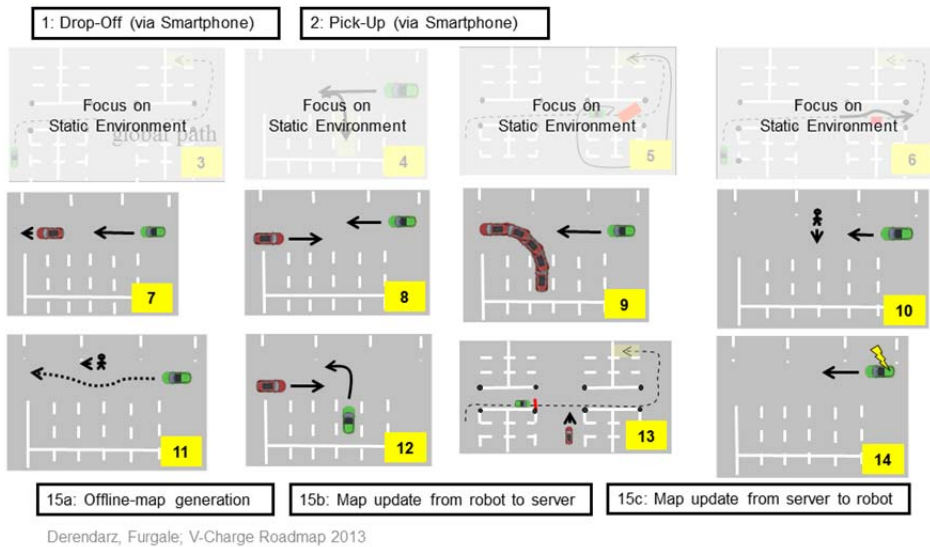
1. **integration** of classification and detection software into the vehicle framework; this represented the main activity for the first half of the V-Charge year, leading to have classification software running (in the loop) during automated driving sessions;
2. delivery of **pedestrian detection** and classification functionality; as today already integrated and in the loop;
3. continuous development and improvement of *classification based* algorithm aimed to detect specific classes of dynamic objects into the V-Charge scenario; in particular several different approaches have been studied and tested to improve detection and tracking, specifically designed to tackle the V-Charge scenarios.
4. re-engineering of the whole software into a more **generic framework**; this will allow to easily add additional classes of obstacles as well as new detection approaches, different than the current classification based one.

Parma team dedicated to WP3 has been significantly broadened and now includes 2 PhD students and 2 senior Post-doc.

3.3.1 Dynamic Object Detection - Roadmap

Starting from the V-Charge scenarios, a road map for Dynamic Object Detection has been defined, to cover the all possible cases where it is necessary to deal with moving obstacles. Particular attention has been paid to those scenarios where obstacles can be detected only by monocular cameras, due to the limited FOV and range of Stereo and Ultra Sonic.

Dynamic Scenarios



Functionality	Description	Useful when	Q2 '14	Q3 '14	Q4 '14
Multi-view Detection	Use of multiple pin-hole views instead of fish-eye images	Always	100%		
Pedestrian Detection	Standard training	10 and 11	100%		
Pedestrian Detection	New training with V-Charge images	10 and 11	90%		
Background subtraction	Detection of moving things when ego vehicle is standing	12 and 13	80%		
Sparse Stereo Matching	Match of sparse cross-camera features in corner areas (e.g. front-left, front-right, etc.) for a better estimation of obstacles position	12 and 13	80%		
Parking Spot Detection	Detect presence and position of parking spots by monocular images	Searching for parking spots		80%	

Mono Detection Tracking	Improvements in objects tracking using full fish-eye image	Always		X	
Vehicle Orientation1	To provide information about possible vehicle orientation. How: separate classifier	Always		X	
Stereo IPM	Difference in IPM images form different cameras. Works in corner areas.	12 and 13		X	
Vehicle Orientation2	To provide information about possible vehicle orientation. How: separate classifier + part based	Always			5%

The table above shows how we started to evaluate detection techniques that are not exclusively based on classification. While classification is a powerful approach in many cases, it is still weak in estimating position and speed, unless strong assumption and simplifications are adopted (e.g. highway driving).

3.3.2 Multi View Detection

The original 4 fish-eye images cover the complete 360° FOV around the vehicle; however, strongly distorted images are can be challenging for classifiers trained on standard pin-hole camera.

Exploiting the extrinsic, intrinsic and distortion cameras' parameters, each fish-eye image is projected onto several planes, obtaining a set of virtual unwarped pinhole camera (i.e. perspective) images, as shown in Figure 21. These virtual planes are fully customizable in terms of their number, size, orientation and field of view.

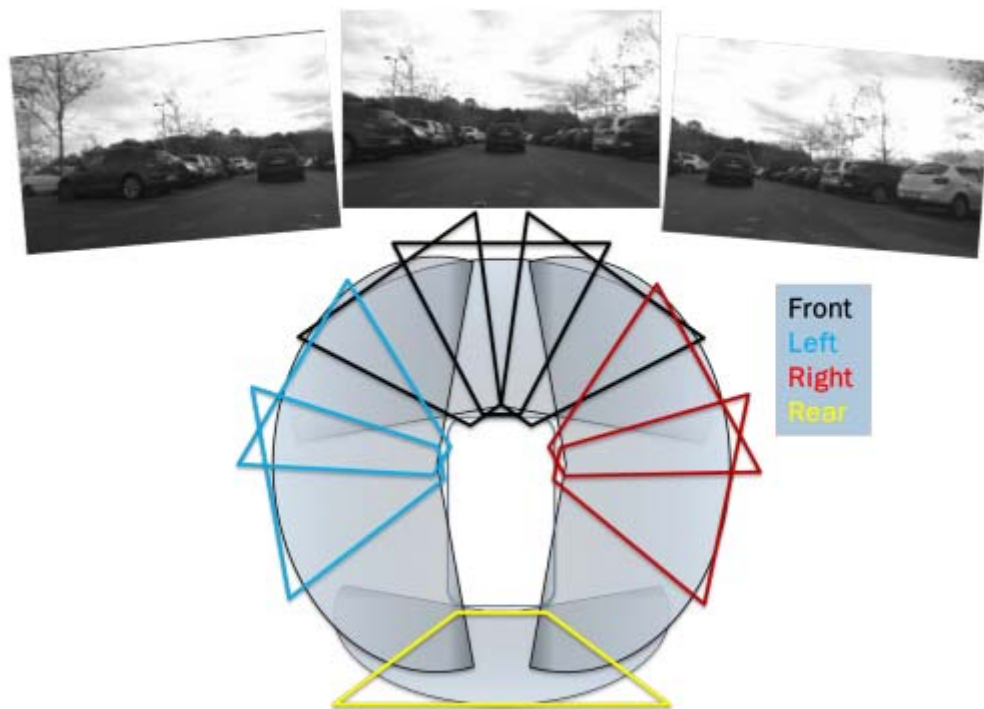


Figure 21: Virtual views layout: 3 virtual views from front camera (black), 2 virtual views from left(blue) and right(red), 1 single virtual view from rear, camera (yellow). On the upper row are shown the 3 virtual views obtained from the original fish-eye image,

This approach allowed a fast deployment and integration of the current classification algorithm, which has been designed for perspective images. However, it also brings some drawback:

1. classification is applied to a relatively large number of images, that, summed together, in general exceed the original fish-eye image size, posing computational power consumption issues;
2. objects needs to be tracked along the various virtual views, to obtain a single list of objects for each camera;
3. it is less straightforward to track obstacles from one camera to another, since there might be multiple views within a camera that overlap with multiple views on the adjacent ones.

For these reasons some experimentation has been carried on in order to find a trade-off between perspective and fish-eye images.

3.3.3 Pedestrian Detection

The pedestrian detection system developed within the V-Charge project uses a part-based classification for the candidates identification and a feature based tracking for increasing the result robustness.



Figure 22: example of monocular pedestrian detection.

Basically the system includes a novel part based pedestrian detection high performance approach with a combination of Haarcascade and HOG-SVM, together with a features-based tracking. Since missing detections are more likely than false positives due to the high robustness of the part-based system, a feature based tracking was introduced with the aim of counteracting a possible missing detection in two subsequent frames and of increasing the number of true positives and eliminate false positives appearing in single frame. A flowchart of the system is shown in Figure 23.

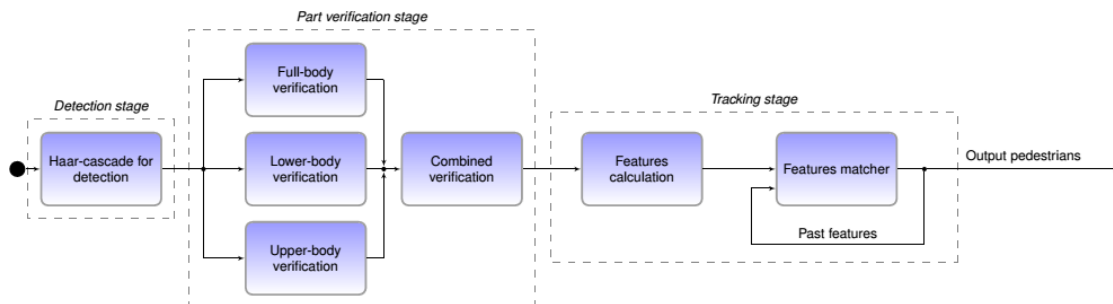


Figure 23: The flowchart of the part based pedestrian detection high performance system described in this paper. Output of detection stage and following stages are bounding boxes.

Detection Stage

The weak classifier used is Adaboost. The structure of a cascade classifier allows to speedup considerably the system filtering the most of pedestrians candidates in the first stage. The variance of false positives is closely linked to the classifier's stages, here denoted as k . The choice of the stages number is driven by the trade-off between the number of detection windows and, the quality of these windows.

Candidate Filtering

The bounding boxes of candidate pedestrians obtained in the detection are filtered and then passed to the verification stage. Filtering is applied to eliminate pedestrians candidate whose dimensions do not respect the human aspect ratio, removing people with size outside of a selected range. Using the Inverse Per-

spective mappings (**IPM**) technique it is possible to know the pedestrian height in the world knowing its height in image coordinates, using the flat road assumption.

Part Verification Stage

As previously described, a part-based approach has been introduced in this phase, evaluating two different compositions of body parts (shown in Figure 24):

- full body, upper body and lower body;
- full body, head, torso and legs.

Dense **HOG** descriptors, by integral images to speed up the system, are calculated for each part of the body and passed to the corresponding **SVM**. Two different types of SVM were tested, a linear SVM and a non-linear SVM. Each one have two variants: binary, providing only the classification (pedestrian or non-pedestrian) of the element and, regression, that provides the estimated function value. The use of different datasets has also been analyzed at this stage.



Figure 24: Example of part boundaries for the 2 and 3-part verification.

Combined Verification Stage

A combination of the three values returned by the verification stage is need to classify the detection window; two different approaches have been implemented:

- a majority vote approach, where at least two out of three classifier must label the window as pedestrian.
- a regression output classification approach, where the three float values coming from the detection stage are used to train a new classifier.

Several types of classifiers were tested: a linear SVM, a non-linear SVM, and a Bayesian classifier; in the results section the different performances of each one will be shown. The majority vote approach allows, unlike regression output classification approach, to detect pedestrians partially occluded but, with an higher false positive rate.

Tracking

To counteract the high selectivity of part-based approach, a feature-based tracking system has been introduced exploiting the features translation and light invariance. A set of features, based on multiple local convolution, key point and descriptor, are extracted from two different images: horizontal and vertical gradients. One of the features-based tracking system downsides is the strict dependence on the vehicle ego-motion. High vehicle speed, in conjunction with a low frame-rate, means large differences between the same pedestrian framed in consecutive frames and, therefore, higher complexity for the matching phase. However, this is not the case of V-Charge scenarios, where ego speed is typically pretty low.

The tracker considers a pedestrian candidate as true pedestrian and, inserts him in the tracking system, if it is detected for at least 250 milliseconds by the SVM. The pedestrian will be matched with new pedestrian candidates in the following frames using the features extracted from the images and then updating its position.

Assessment

A thorough evaluation of the different setups parameters is done before deploying the algorithm into the project. The training set has been chosen selecting 2400 images from the Daimler database Training Set, which comprehensively counts 26000 images; the remaining 2000 images of this database has been used for the parameters optimization. The final test has been performed again using the Daimler database Test Set, which is not included in the Training Set.

To compare each result with the ground truth we used the PASCAL VOC: a detection is considered correct if the overlap between the area of ground truth bounding box and the area of the bounding box obtained is more than 50%. The results are shown using ROC curves, created by plotting the fraction of true positives out of the positives (TPR = true positive rate) vs. the fraction of false positives out of the negatives (FPR = false positive rate), at various threshold settings.

Algorithm	Part-based	Average detection rate	Speed
Our Algorithm	yes	0.72	15.5 FPS
MultiFtr+Motion	no	0.71	0.004 FPS
LatSvm-V2	yes	0.62	0.164 FPS
MultiFtr+CSS	no	0.61	0.005 FPS
HogLbp	no	0.51	0.014 FPS
HikSvm	no	0.45	0.036 FPS
MultiFtr	no	0.43	0.017 FPS
LatSvm-V1	yes	0.42	0.098 FPS
HOG	no	0.40	0.054 FPS
Shapelet	no	0.06	0.010 FPS
VJ	no	0.05	0.089 FPS
FPDW	no	N/A for DaimlerDB dataset	2.670 FPS

Performance on Daimler-DB dataset are shown in Figure 25: an ROC curves is plotted reporting several results varying the number of cascade stages (k).

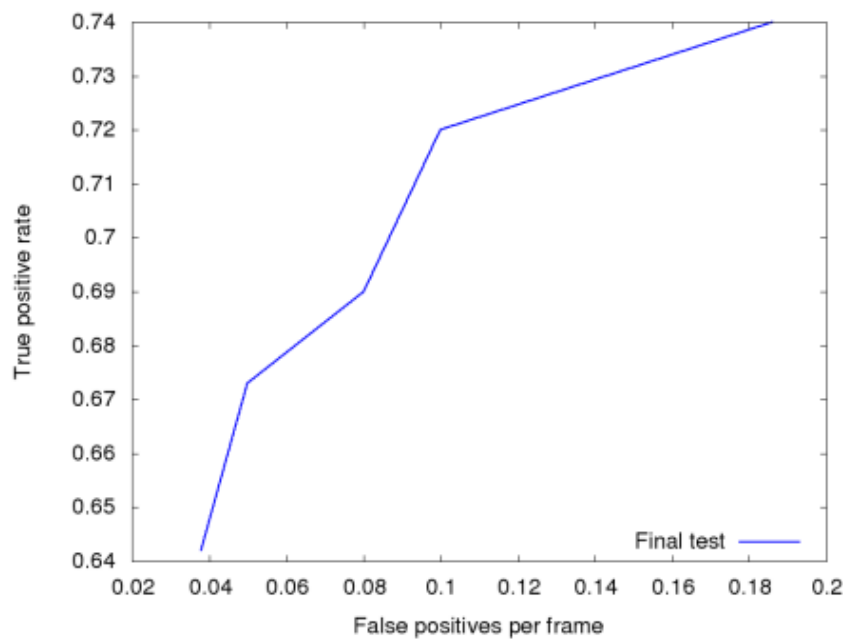


Figure 25: Final test detection

3.3.4 Cylindrical Images

As mentioned before, the current approach for detection via classification is to split each fish-eye image into several pin-hole virtual views, this to help the classifier to deal with very wide angle, distorted, images. However, some drawbacks exist: the number of images to be processed can become significantly large; multi view tracking might be not trivial, as well as multi camera tracking.

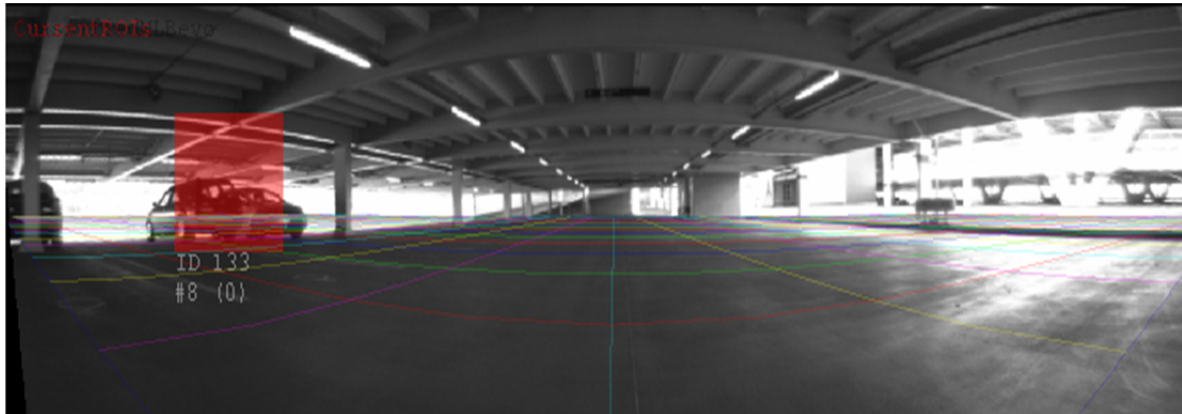


Figure 26: Cylindrical images, where vertical straight lines are still straight, while the horizontal ones are distorted.

For these reasons some experiment have been carried on to obtain a trade-off between perspective images and fish-eye images, in order have one single image to be processed and, at the same, to preserve the good performances delivered on perspective views. These trade-off images are called Cylindrical Images.

A perspective image is an image where straight lines in world are still straight as seen on the image; perspective images can have virtually any size and any resulting focal length, as far as it is greater than 0.

Cylindrical Images can still have virtually any size, and can have any horizontal and vertical FOVs, as far as they are within the original fish-eye FOVs. Basically, Cylindrical Images are obtained from the original fish-eye rectifying only column wise, and letting the rows distorted. The final result is shown in Figure 26: here vertical straight lines in world are straight also as seen on the image, while the horizontal are distorted.

Experimentations on this kind of images are promising: the classification algorithm seems to be much more sensitive to vertical edges than horizontal, leading to reasonable results. In addition we have all the advantages of a single, wide angle, image: easy inner-camera tracking, thanks to the long obstacles persistence on the image; improved cross-camera tracking, due to the large overlapping area, that allow to see the same obstacle on both images for a longer time.

3.3.5 Sparse Stereo Matching

Thanks to the very wide angle view allowed by fish-eye lenses, there is a large overlapping between the 4 cameras. In particular we have 4 corners covered by two cameras, meaning that it is possible to perform some **stereo matching**. Moreover, these are areas not covered by any other sensors on the vehicle, other than US in the short range.

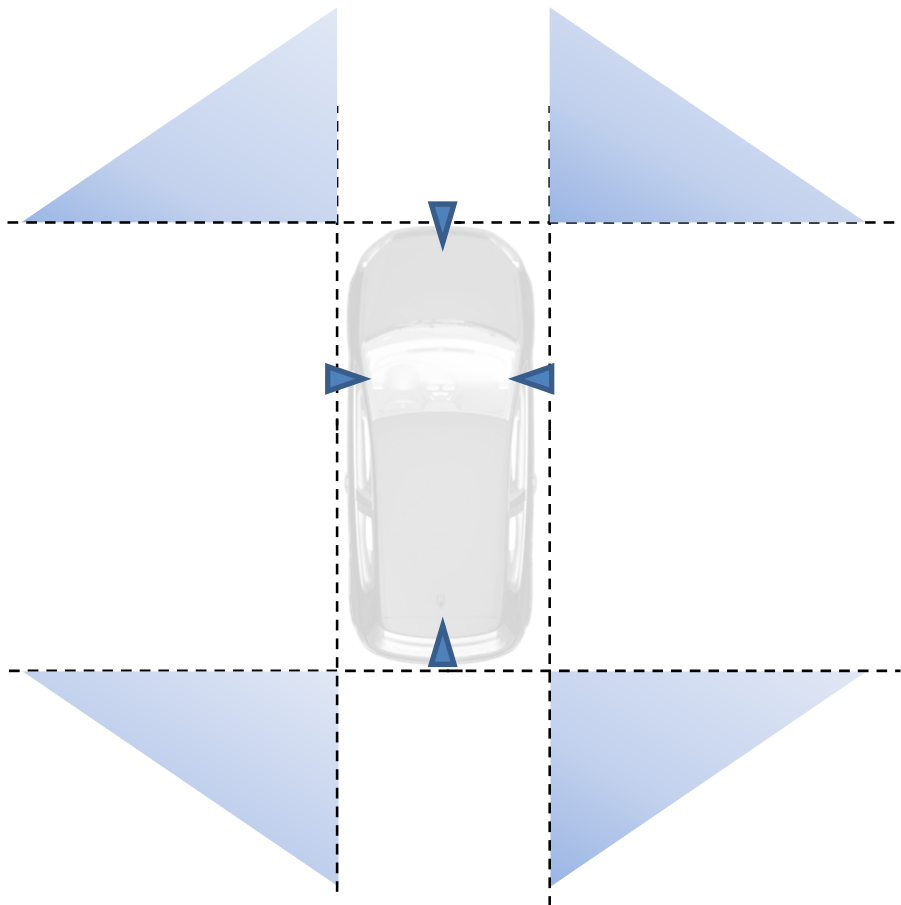


Figure 27: Areas of overlapping field of view of fisheye-cameras

In this configuration it is not possible to simply apply the most common dense stereo matching algorithms, like SGM, since the cameras are actually not rectified: in other words they do not represent canonical stereo pairs. However, it is always possible to apply *sparse stereo matching*, that does not require a perfect alignment between left and right camera.

The current version of the processing, still under development, consist of two separate phases:

1. Sparse feature matching: here key points are matched along two adjacent camera (e.g. front and left)
2. On the basis of the extrinsic and intrinsic camera calibration it is possible to find the intersection of the two projection rays, that always exist as far as calibration and pixel match are correct.

The obtained intersection represents the position of the point seen by the two cameras, in **world coordinates**

Sparse stereo matching will allow to estimate position and speed of objects detected in the corner areas, such as upcoming vehicles at intersection or when pulling out from a parking spots.

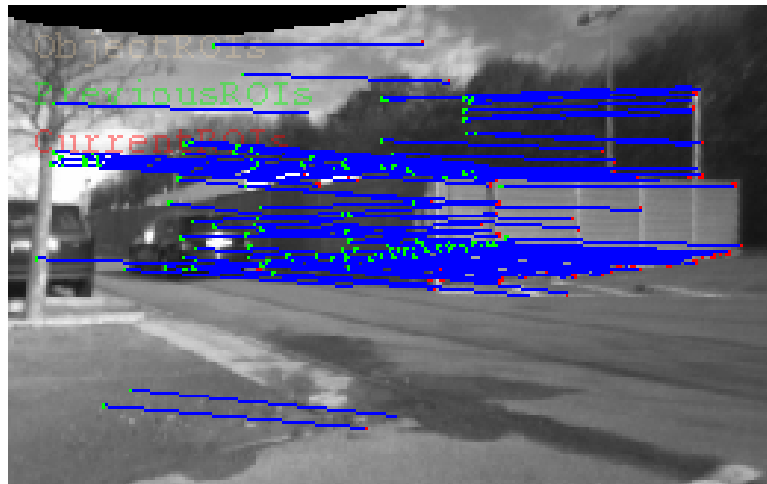


Figure 28: Sparse stereo matching: red dots are keypoints on the current left image, green dots represents the corresponding keypoints on current right image.

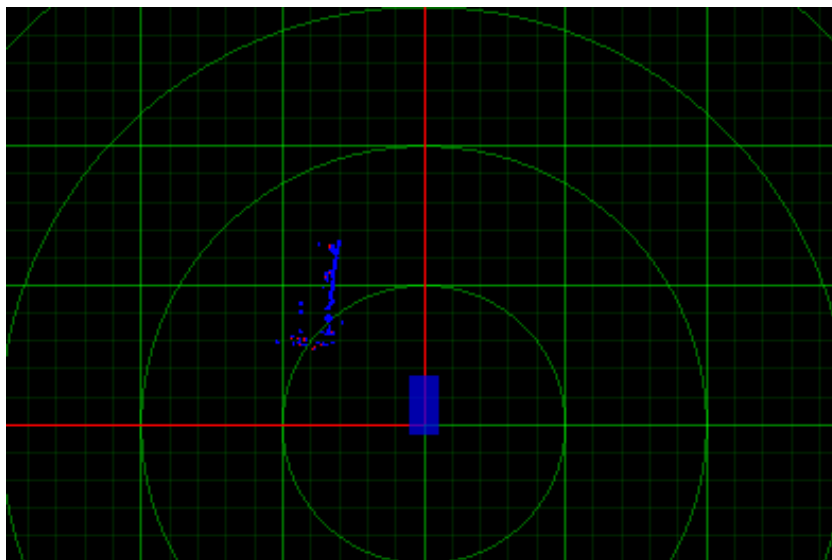


Figure 29: bird-eye view, where it is possible to see positions the keypoints detected, that actually correspond to the wall visible on the above image. The vehicle is represented by just a few points, but still enough to provide a reasonable estimation of its position.

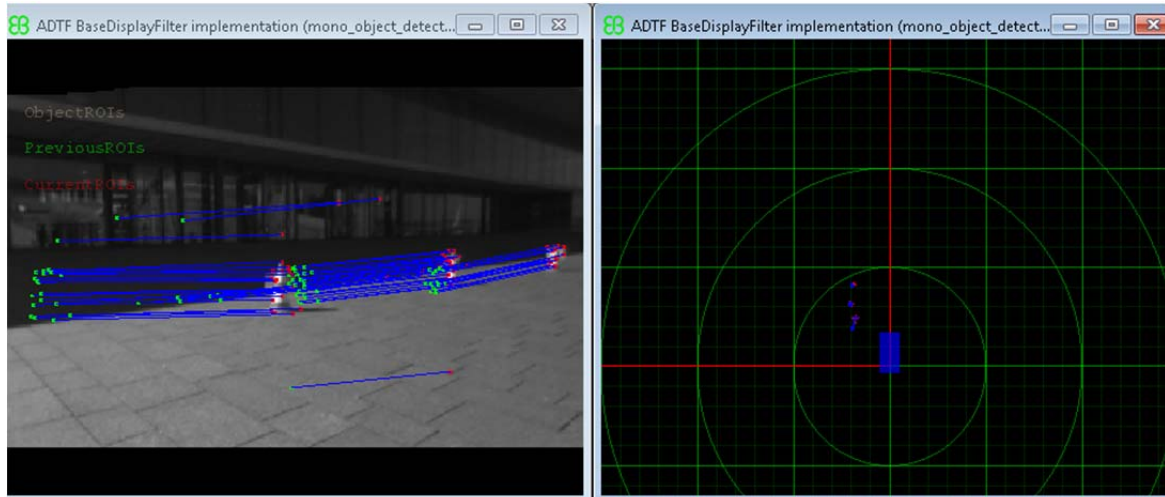
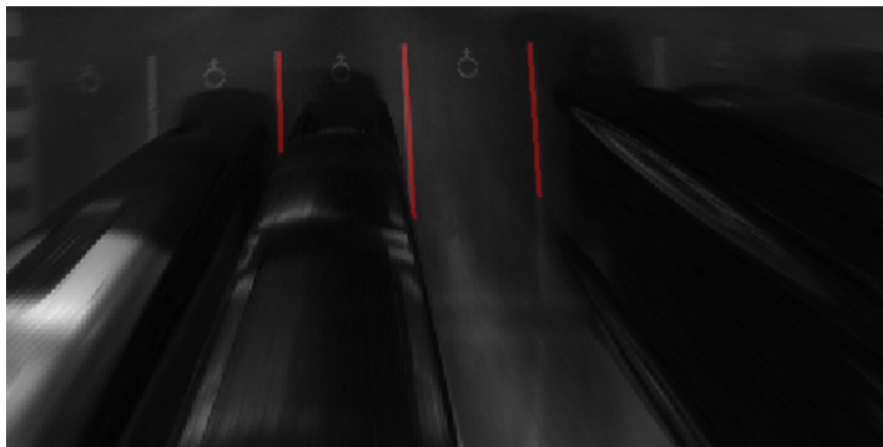


Figure 30: Another Sparse Stereo Matching example. Here are visible the matches obtained on the small concrete pillars and the corresponding 3D points in the tow view

3.3.6 Parking Spot Detection

To help the localization of parking spots, an additional vision based detection algorithm has been developed, based on mono cameras.

This kind of processing does not exactly falls under the “Dynamic Object Detection” definition, however, WP2 and WP3 partners recognized the usefulness of exploring additional detection features related with parking recognition and localization, especially thinking about the scenarios when the ego vehicle does not have a pre-assigned parking spot, but it is necessary to find a suitable free place.



The current version, still under development, consists of few straightforward steps:

1. A Low-High-Low filtering is applied, to highlight possible transitions from asphalt to white paint and asphalt again;
2. Straight, parallel, lines are then detected
3. Detected lines are tested with respect to the expected parking spot size, orientation, relative positions, etc.

Open problems are:

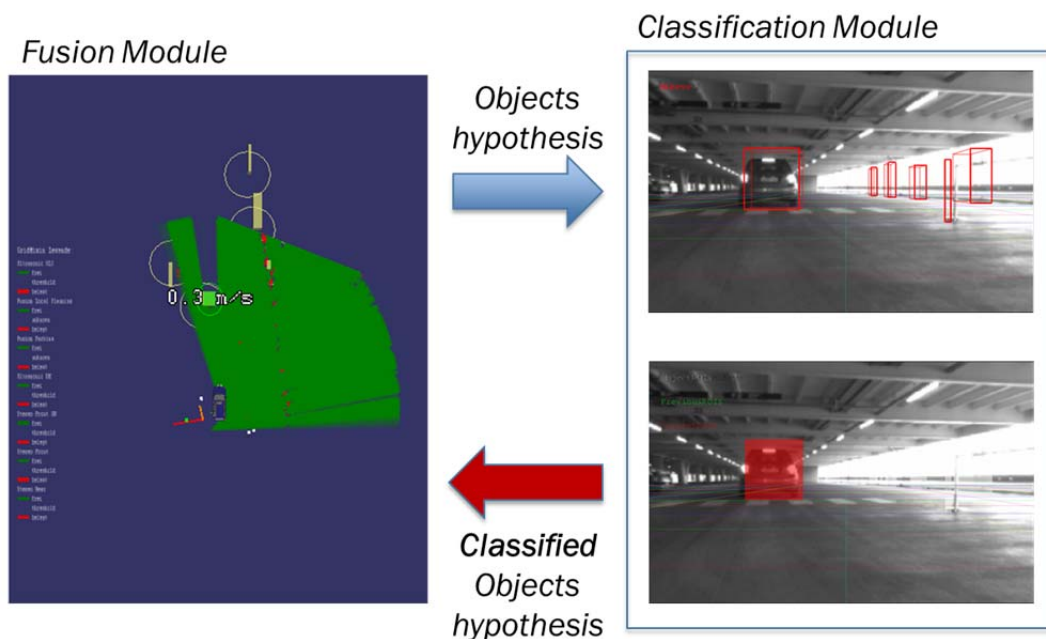
1. Parked cars occluding lines
2. Degraded lines

3.3.7 Object-hypothesis Classification

During the second year of activities in **Task 3.4 “Object Classification”** a lot of effort has been spent in **integration**.

Object classification consist of receiving a list of obstacles hypothesis from Stereo, or other sensors, and classify them on the basis of the visual information coming from the mono cameras. This processing provides useful information in the following scenarios:

1. ACC: here it is necessary to understand if an in-lane obstacle is a car, a parked car, or another kind of obstacles. On the basis of this information the Path Planner will adopt the proper driving behavior.
2. Pedestrian Crossing: if an obstacle is standing in front of a pedestrian crossing marking, it is important to understand if the obstacles is a pedestrian or not. In the first case the ego vehicle must yield way to the pedestrian, even when it is not occupying the lane.



3

This module was used during the automated driving session at the Stuttgart Demo, April 2014, allowing successful ACC behavior.

3.4 Ultrasonic Triangulation

For higher accuracy the raw ultrasonic echoes are preprocessed. Using neighbored sensors a higher spatial resolution is achieved. In addition already processed data is combined to a more complex environment reconstruction. As a result model hypothesis based on single obstacle points and line segments such as on contours of parked cars are sent to higher perception modules. Figure 31 visualizes a typical output of the ultrasonic preprocessing module.



Figure 31: Line and Point Objects triangulated immediately in the Ultrasonic ECU

4 Sensor Data Fusion

In this section the preprocessed sensor measurements, the object segments generated from stereo and motion stereo algorithms and the region of objects within the image space as well as ultrasonic echoes and triangulations are fused together. Here a clear separation is performed whether an obstacle may belong to an object capable of being a relevant road user or a single obstacle point to be verified over time and relevant for obstacle avoidance. Section 4.2 provides a detailed view on how objects are generated from sensor readings and tracked to verify their dynamic state. In the following section 4.1 the occupancy grid implementation for remaining static obstacles is presented and the logic to cope with obstacles turning between dynamic and static state.

4.1 Occupancy Grid Map-based Static Obstacle Fusion

Stefan Wonneberger, Volkswagen AG

When driving at low speed with high steering angles or maneuvering in limited space very precise, robust but fast obstacle detection is a mandatory component. Already back in the eighties, the occupancy grid map approach has been proposed to aggregate single 2D or 3D sensor measurements in a 2D-cell grid associated with the ground plane [9]. The grid origin is defined within a virtually abso-

lute coordinate frame, however only cells in the desired field of view of the vehicle are kept resulting typically in a rectangular area around the vehicle. While the vehicle moves the grid is extended in driving direction typically resulting in a memory clear and shift logic. With this in mind, the quality of the resulting grid map is based on the sensor model, describing how the information of single sensor measurement is transformed in a list of affected cells and which probability of the measurement can be derived for each affected cell. In addition, multiple sensor measurements over time or from different sensors need to be fused per cell robust enough to cope with outliers but fast converging to a significant state indication such as free or occupied.

In V-Charge occupancy maps are used for two main purposes. First the reconstruction of obstacles to avoid collision when driving from a pick-up zone to the area of parking lots and second to have a very precise reconstruction of obstacles around and within a designated parking spot.

As a result we compute two grid layers with different resolution and size. The layer relevant for parking spot reconstruction has a resolution of 10 cm currently at a length of 25 m per dimension. The layer accumulating obstacles for local planning while being in normal drive mode has a coarser resolution of 20 cm at a size of 50 m. As a result both layers can be used to have a single multi-resolution view of the scene while keeping simpler shift and sensor models.

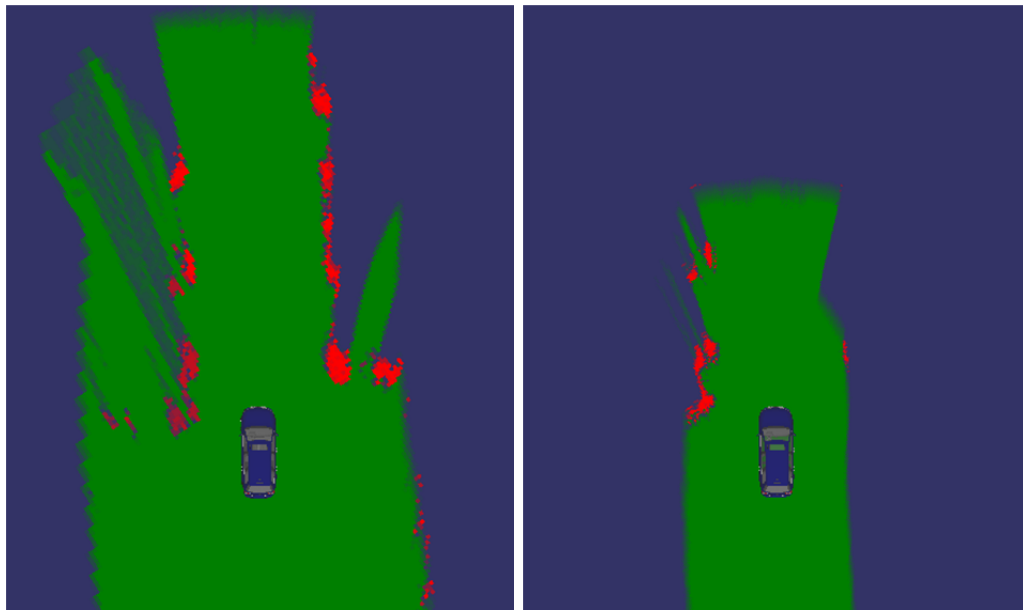


Figure 32: Driving Grid at coarse resolution with fast converging parameters and parking layer for accurate parking spot reconstruction

4.1.1 Sensors and Sensor Models

The stereo and motion stereo systems are sharing both a single data description allowing a single sensor model to be used for all these sensor sources (compare sections 3.1 and 3.2). The 3D information is clustered into single 2D obstacle point with a single confidence information (object segments, see Figure 33) com-

pared to [10]. This information reduction allows a straight-forward definition of the sensor model. Each obstacle with a given probability and position uncertainty can be transformed in an occupation region with given mass per cell. In addition each obstacle induces a line or polygon region of negative / free updates.

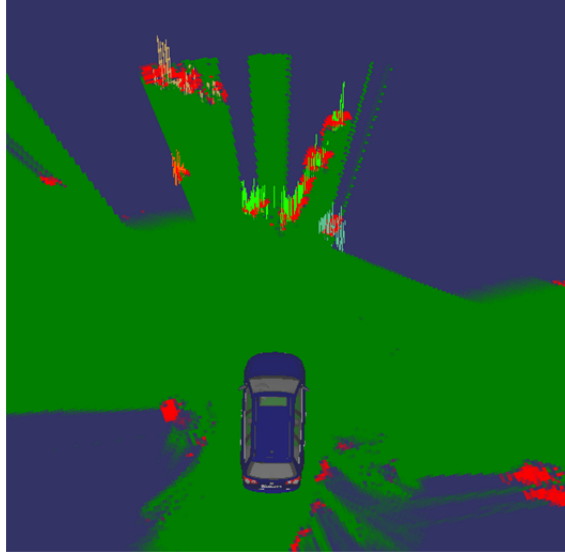


Figure 33: Object Segment and applied sensor model for Stereo and Motion Stereo Sensor Data

The ultrasonic information comes on two different levels. First the raw echoes can be used to generate a region of negative updates where no obstacle has been measured. However the raw echoes provide only a single distance to an obstacle at a drawback of no further information of the angle. The ultrasonic ECU used, performs an internal triangulation (See section 3.4) of the echoes to provide single points and line segments having a much higher resolution and typically fitting very well to the geometry of parked cars, walls and single obstacles. However this information is already tracked and aggregated over time conflicting with the primary principle of grid map fusion where each new information is considered as independent. As a result only the negative update is considered for the cell update. The preprocessed obstacle position is used in a later step.

4.1.2 Sensor Fusion

As already stated, two grid layers are computed at the same time. The layer for driving and obstacle avoidance has configured sensor models for front and rear stereo, motion stereo and raw ultrasonic echoes. Since the accuracy of the obstacle position is less crucial then for the exact parking spot reconstruction, obstacles are fused with a higher mass leading to a faster convergence and earlier recognition of obstacles within the path planning module.

For inter-cell information fusion a simple Bayesian fusion is used. This comes at cheaper computational costs. However only the distinguishing of cell states by a single probability for free, occupied or unknown state is possible. Especially an additional conflict state may be useful for different applications such as a wrong

dynamic detection and sensors with high noise. More complex fusion logics such as Dempster-Shafer would allow a modeling in this direction.

For the generation of the parking layer, a very precise modeling of the static obstacles is mandatory. Thus the sensor models are configured to add a significant less mass value requiring a larger amount of measurements per grid cell to either mark the cell as occupied, belonging to an object or definitely seen as free space.

In the previous section the pre-tracked ultrasonic information were mentioned. Since this data class conflicts with the update logic of the Bayesian approach, we currently only transform the obstacle points and line segments into a list of affected cells using simple line discretization. Depending on the current cell state and the probability of the ultrasonic object the final cell state provided to the path planning modules is changed to 'occupied' in a hard way.

For the decision of the occupation state per cell, we define a fixed threshold of the probability and convert periodically the probabilistic layer into a tristate logic layer providing the states free, unknown and occupied. Based on this information the path planning modules are able to avoid obstacles as well as may regulate speed information depending on a clear free space information or region of unknown or conflicting sensor results.

4.1.3 Compensation of Dynamic Objects

Both the grid map fusion and dynamic object fusion module shall provide a consistent view of the environment to the path planning modules. Thus dynamic objects shall be processed as box model description with the object tracking and fusion module and be filtered from the grid layers.

Here we follow an association and labeling approach for the raw sensor information before the new information is either processed by grid or object fusion.

The stereo sensors are currently the only sensors within V-Charge which are used for dynamic object detection, thus this raw sensor data need to be classified and send to the correct module. We first apply a clustering of object segments (see Figure 35, Page 40). This cluster is used to perform the generation of bounding boxes for object tracking (compare Section 4.2). The information between the generated bounding box and each raw object segment is kept via a unique object id for the current measurement. Before the box hypothesis is used for object tracking it is tried to be associated to an already valid object track. If found and a given a necessary confidence, the raw data segments are marked as invalid for the processing within the grid module. If the object track has a very low speed or is even still standing (such as a vehicle standing in front of the ego vehicle) the associated object segments are further marked as explicitly free update to cope with sensor noise. This leads to a fast clearance of grid cells which may have been marked as occupied while the object is initialized in the object fusion module. This can for example happen when an object is seen by stereo and later clas-

sified by mono camera object detection. In the next section the object tracking and fusion module is further described.



Figure 34: Driving Layer with Overlay Bound Box resulting from Object Fusion module already labeled dynamic e.g. by the image-based object detection and classification

4.2 Dynamic Object Tracking & Fusion

Dr. Thien-Nghia Nguyen, Volkswagen AG

In our designated scenarios it is essential to consider the interaction with other moving traffic participants such as pedestrians and other cars. In order for the motion planning algorithms to make right decision in such situations, we developed a reliable tracking framework of moving objects. Our object tracking and fusion framework gathers environment information from all sensors that are installed in the car and periodically provide a unified picture of the surrounding environment. In this chapter, we give an overview of the sensors and how we evaluate the sensor information, our information fusion algorithm, the current results and an outlook what we want to achieve until the end of the project.

4.2.1 Sensor Setup

Stereo camera

In 3.1, it was described how stereo segments can be extracted from stereo system. In Figure 35 we see that dynamic targets in the field of view are mostly covered by the segments. Based on that, we have developed an efficient algorithm to extract all kind of targets as bounding boxes from stereo segments, like shown in Figure 35. Our algorithm consists of two steps: clustering from segments and rectangle fitting on found segments.

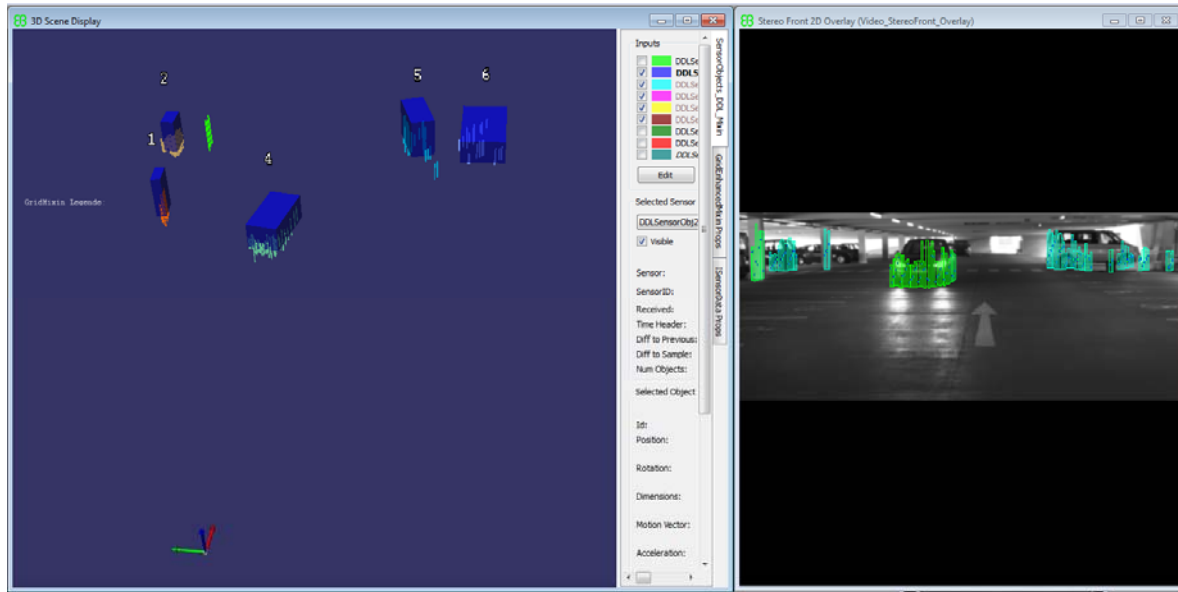


Figure 35: Segment Clustering and Object Hypothesis

Mono camera

Object detection techniques with mono camera was described in section 3.3, here the results have been mentioned which serve as an input for object fusion:

Classified stereo objects:

The extracted bounding boxes from stereo above (Figure 35) will be sent via DDS to the MonoDetection-PC in the car. There a classification from of these bounding boxes based on mono images is applied. Afterward a list of the same object, with classification information will be sent back to the Object Fusion System.

Mono objects

The mono objects detected by the Image-based Detection and Classification will be sent to Object Fusion as bounding box, with estimated angle, distance and size. The accuracy of the mono detection is to be verified until the end of the project.

4.2.2 Sensor fusion concept

In order to fuse the information from the sensors, we chose a probabilistic approach. Probabilistic approaches are widely used in robotic field. In [11] a probabilistic framework was introduced. The main idea is quite simple: we try to estimate the movement of the objects by using the difference between consecutive measurements and also by taking into account the movement of the ego car. The movements of the objects are assumed by well-known dynamic models, which correspond to simplified movement of cars or pedestrian. The updates of the objects' status consist of fusing the prediction, which based on prio estimation, with the new measurement, which is provided by sensors. The mathematical back-

ground for prediction and update steps are realized by an Extended Kalman Filter [12]. For our purpose we prefer the extended Kalman filter due to the non-linearity in object movements as well as in the projection of object status in measurement vector at the prediction step.

As an addition to the fusion of dynamic information of the object, we also combine objects classification information, for example, when an object has been detected by both sensors, mono and stereo. Due to the fact that mono object detection relies on classification, we can exploit this to update object information. Classification information helps us to find out which dynamic model we would use for a track. In the following we introduce in more details our dynamic models and the measurement models.

Dynamic model

It was already described how objects as bounding box were created based on 3D segments. If that object were a car, more stable results can be achieved if we use car-like model to track the objects.

Car-like model

This model assumes that a car state consists of following information:

$$[x, y, v_{abs}, a_{abs}, \varphi, \omega, W, L]$$

Whereas movements are reflected by a constant yaw rate ω and constant acceleration a_{abs} along the tangent of the “yaw arc”.

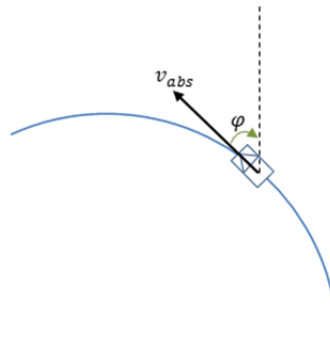


Figure 36: dynamic model constant acceleration constant turn (CA_CT)

Because this movement is not linear, we have derived the h- and H matrices correspondingly. At the project demonstration in April in Stuttgart, the ACC-scenario was accomplished using this car model. We had also applied the car model for oncoming traffic and achieved very good results.

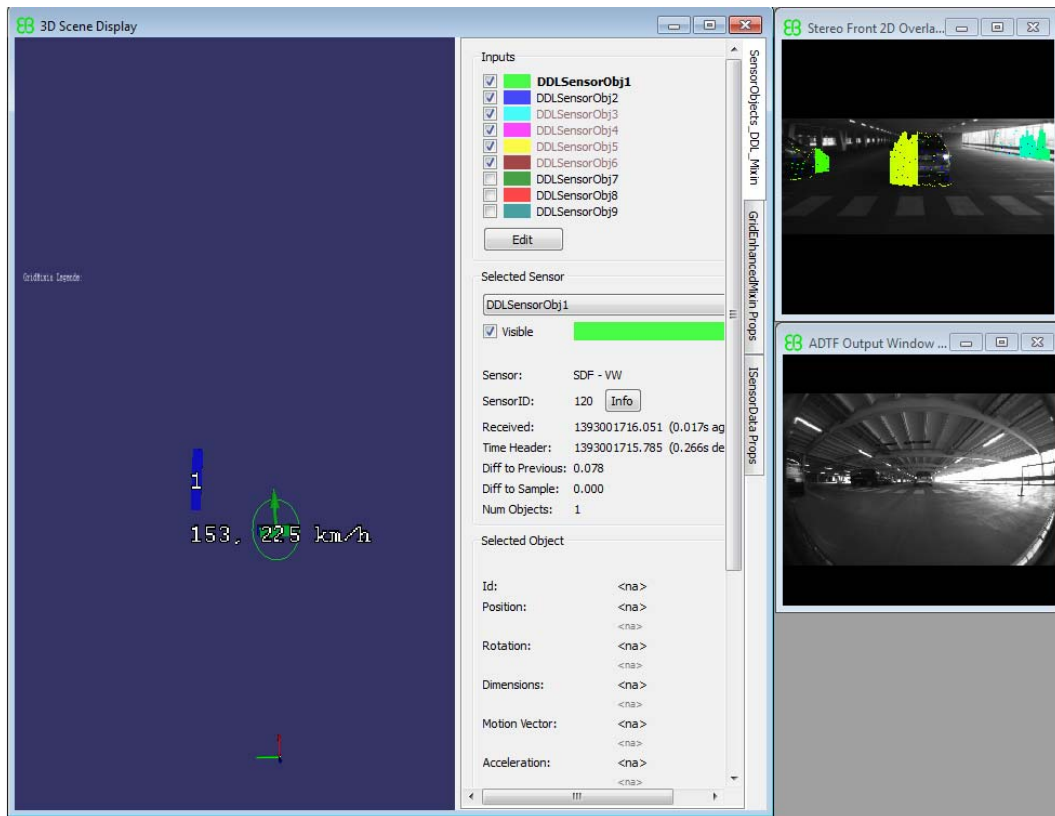


Figure 37: ACC drive

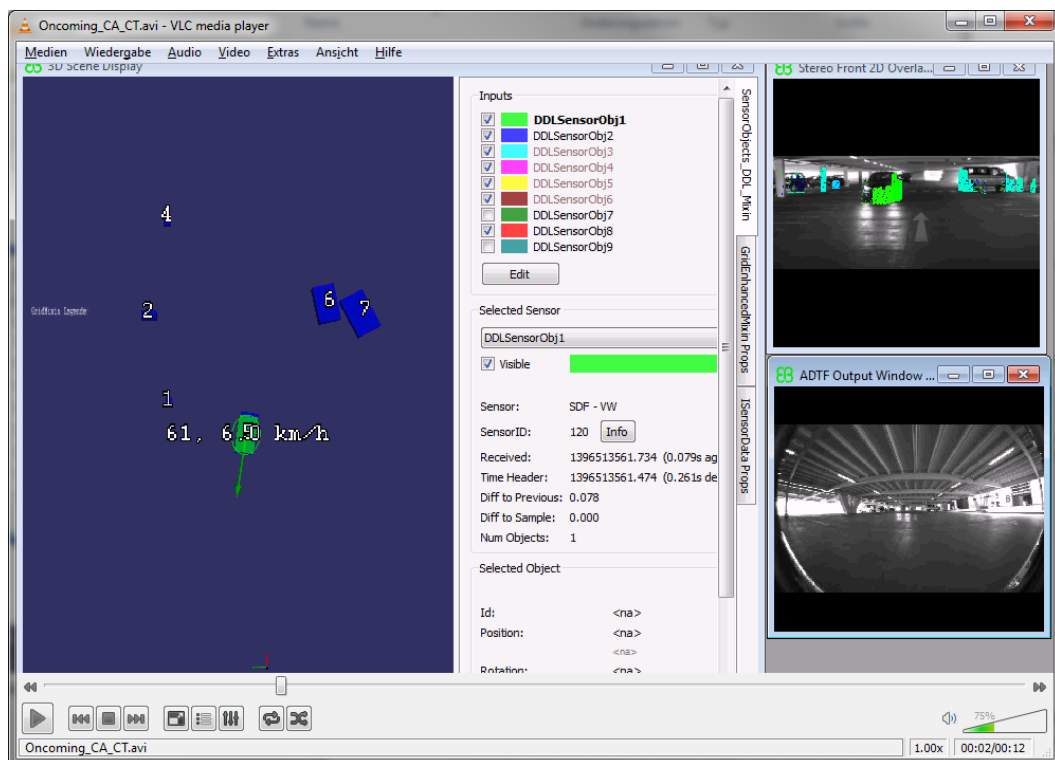


Figure 38: Oncoming traffic

Moving rectangle model

On the other side, tracking of pedestrian would not be effective with car model. Physically the pedestrian has the ability to move relatively freely to the side, so it is complicated if we always try to derive a yaw angle or yaw rate for the pedestrian. Easier would be, we assume that a pedestrian is a rectangle, with length and width and velocity to the x- and y-direction. The moving rectangle model therefore has following information $[x, y, v_x, v_y, W, L]$.

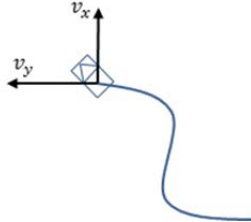


Figure 39: Moving rectangle model (Cvxy)

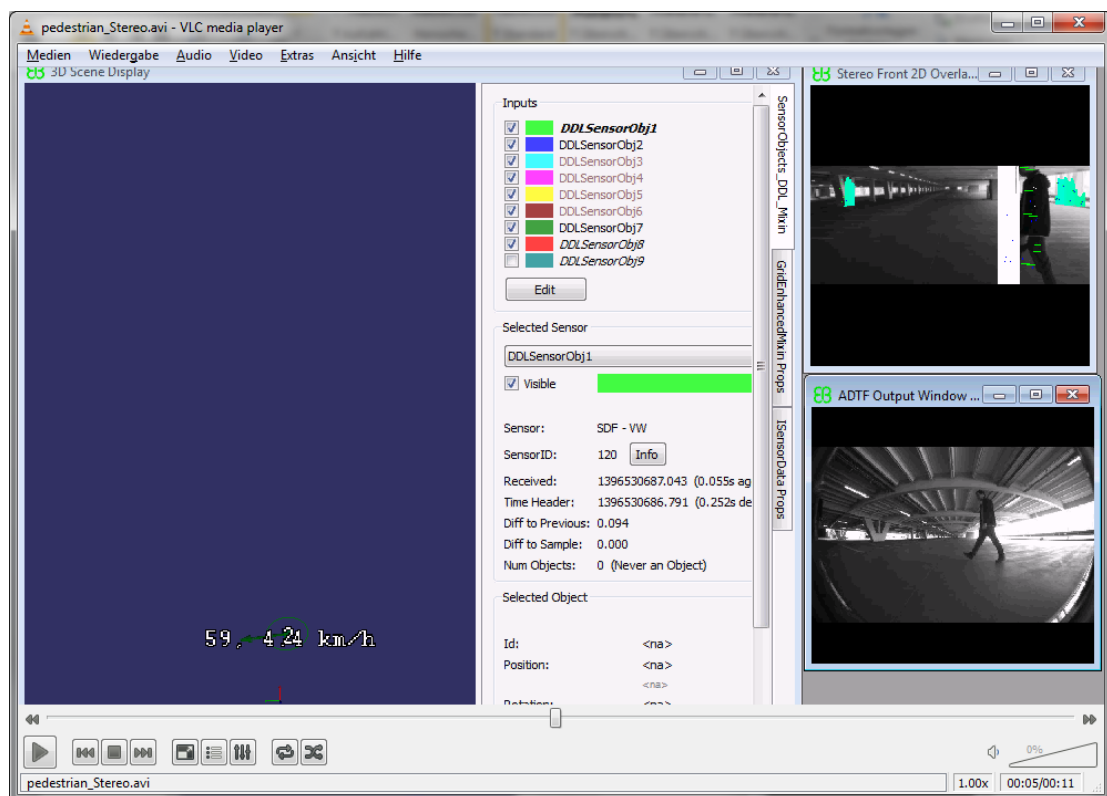


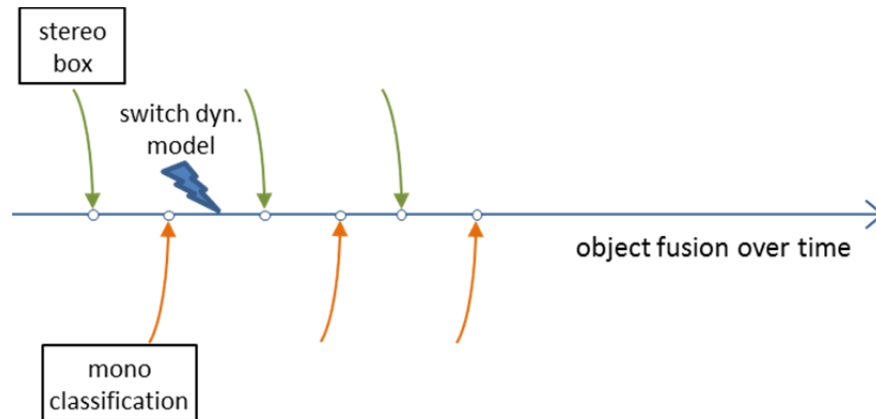
Figure 40: Pedestrian tracking with moving rectangle model

Dynamic model switching

One main issue of object fusion is that, to use the right dynamic model for the right object. So as soon as we received classification information from the mono pipeline, we switch either to car-like model or moving rectangle model. In case no classification information is available, the moving rectangle model will be applied for all objects. For targets which are passenger cars, this model performs not 100% like car-like model, but is still a very good compromise for lack of classification information.

Integration of classification information

One more main issue is not only to apply the right dynamic model, but also to identify when we can apply which model. As mentioned at the beginning of this chapter, we use mono classification information to determine what kind of object is involved and which dynamic model can be applied. In case object is passenger car, it will be switched to car-like model. In Fig. x it is illustrated how we integrate mono classification in the object fusion



Measurement model

According to the Kalman Filter theory, to update the object status, we need to define the appropriate Jacobi-Matrix, which helps to incorporate the measurement innovation (difference between measurement and prediction) into estimation.

5 Scene Interpretation & Information Extraction

In this section the different post processing steps are discussed to combine the output of the perception modules with further semantic information of the scene or to reconstruct a specific model information such as parking spots.

5.1 Perception Interface for Path Planning & Parking

The results of the grid and object fusion are provided to the path planning and parking module as a serialized data stream and can be subscribed by any other module as well, such as visualization, using the consortium wide chosen middle-ware solution DDS. The dynamic objects are transmitted as list of single bounding box representation including classification label, orientation, velocity vector and acceleration. For all values probabilistic information is given. The grid layer - parking and driving – can be subscribed as a raw probabilistic model, for fast transmission and low latency the layer are represented in bit-compressed and optionally runlength compressed manor to keep as much information as possible. The interested reader may refer to [13] for a further compression towards polygon line representation of free space and occupied areas.

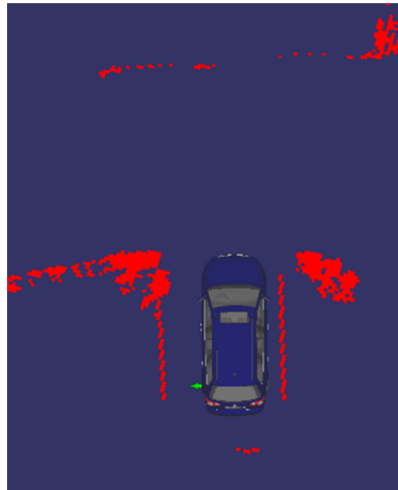


Figure 41: Bool Representation of the parking layer (Optionally a Tristate Version is available)

5.2 Roadgraph-based Scene Interpretation

Dr. Rene Waldmann, Stefan Wonneberger, Volkswagen AG

In the V-Charge Project semantic information for path planning and navigation is used in form of a graph network. The library and logic behind is called 'Roadgraph'. It includes a model description of roads and directed lanes on the parking area as well as a connection to adjacent areas to these lanes such as parking spots. Roads and their lanes are connected to each other indicating possible driving directions at crossroads or more complex intersections. Figure 42 shows a visualization of an example Road Graph for the Mobile Life Campus test site in Wolfsburg. As it can be seen, the Roadgraph is not fully populated for all possible driving areas. Automatic generation and extension of the Roadgraph, gathering semantic information from the vehicles while traversing the parking lot, is a stated goal of V-Charge which is further discussed in WP2. The interested reader may refer to Deliverable 2.3.

The Roadgraph may be either preinstalled in the vehicle and loaded depending on current coordinates of the vehicle, e.g. near to a known parking lot. It can be also received from a public infrastructure or single server: This aspect is covered within work package 5 (Deliverable 5.3).

When having a global localization such as DGPS or the developed visual localization (Deliverable 3.3) for indoor navigation, a relative localization to the absolute Roadgraph coordinates is possible and used for global navigation and path planning.

In addition this combination can also be used for matching the output of the perception modules to Roadgraph lanes or parking areas (Figure 43). With taken carefulness also a filtering can be applied, always taking in account that localization or Roadgraph network may be inaccurate. When both modules supply a sufficient quality level they can be used to filter false positives of detected objects in the scene for example. Either filtering the output of the perception framework but

also on the sensor input level can be applied. As already stated this is only relevant when considering objects for a smooth and natural integration in the traffic flow. In situations close to collision avoidance this filtering shall not be and is not applied.



Figure 42: Roadgraph of Mobile Life Campus Parking Lot, Wolfsburg. Individual Lanes are modelled and connected to allow a global path planning. In addition semantic details such as parking spaces can be stored in the graph.

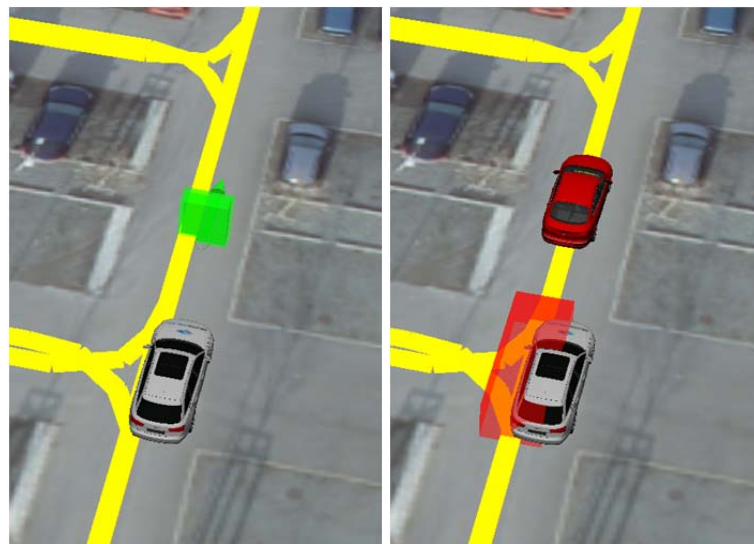


Figure 43: Association of Objects and Ego Vehicle to Roadgraph Lanes - Left: Tracked position from sensor readings of vehicle in front and position of own vehicle using global localization - Right: Successful matching to Roadgraph edges and applied classification

5.3 Local Motion Planning in the Presence of Moving Objects

Ulrich Schwesinger, ETHZ-ASL

Dynamic objects extracted by the classification and tracking pipeline are already utilized by the local motion planner. As the lack of information about objects' future motions yields poor navigation performance in environments with dynamic elements in the scene, the extraction of such dynamic elements and their velocities provides most valuable information for the planning stage.

Within the local motion planner, motion predictions based on a rudimentary constant velocity assumption are generated. Motion plans are generated that yield collision-free navigation via collision detection of candidate trajectories, explicitly considering time as an additional dimension. As within the V-Charge project the vehicle faces both a potentially large number of dynamic elements in the scene and little room for navigation, collision queries are accelerated with the help of a bounding volume hierarchy data structure (AABB-Tree, Deliverable 4.3) in workspace-time space. This greatly reduces the computational cost for collision detection via fast pruning of irrelevant collision pairs and enables the usage of arbitrary motions and shapes of objects. An exemplary AABB-Tree can be seen in Figure 44.

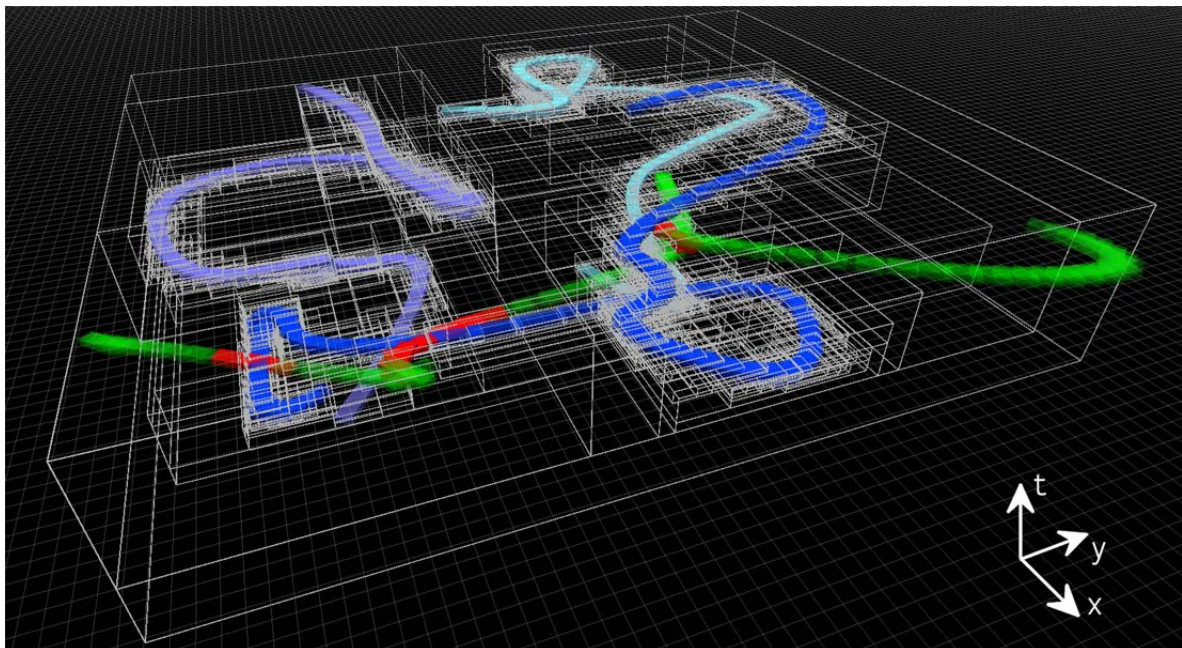


Figure 44: Visualization of an AABB-Tree in workspace-time space for three obstacle trajectories generated by a random walk (blue shades). The ego trajectory can be efficiently checked for collision with the help of the three structure (wireframes).

For a more detailed description of the treatment of dynamic objects within the local motion planner and experimental real-world navigation data refer to deliverable 4.3.

5.4 Parking-Spot Finder

Stefan Wonneberger, Dr. Rene Waldmann, Volkswagen AG

The a-priori Roadgraph information combined with a global visual localization (Deliverable 3.3) allows a semantic interpretation and filtering of the environment reconstruction. This is especially useful when looking for empty parking spaces.

In order to detect empty parking spaces only a region of neighbored parking spots needs to be defined in advance making the necessity of semantic mapping less complex and thus more robust.

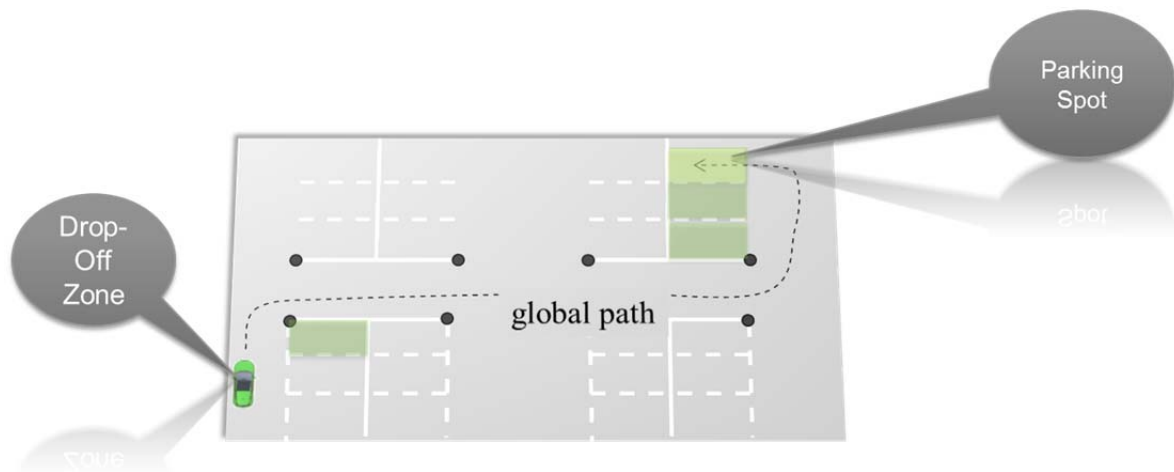


Figure 45: Global Path Planning with semantic information about areas of parking spaces

Like when using a fixed assigned parking spot, the car plans a global path to an area of possible available parking spaces or a traversal path through parking rows. The basic principle is shown in Figure 45.

When the car passes parking spaces the Parking-Spot Finder automatically scans in the background for enough space being probably an empty parking spot. Figure 46 sketches the algorithm.

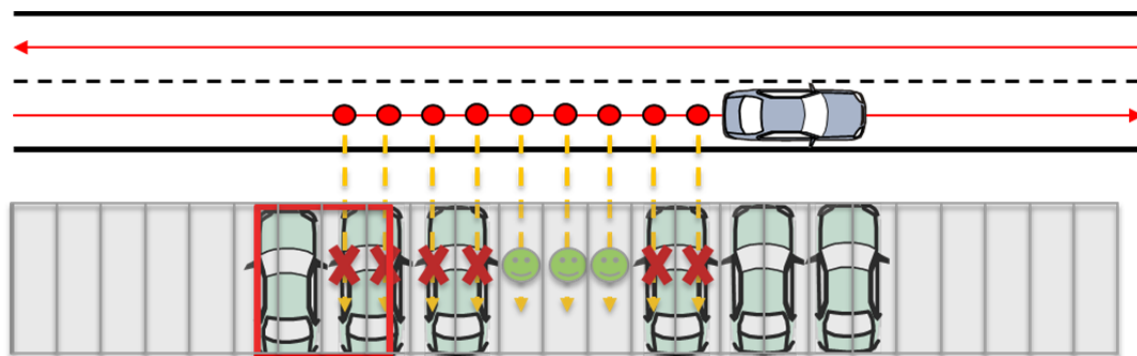


Figure 46: Slice/Scanning-Principle of Occupancy Grid when checking for empty Parking Spaces

Assuming the parking area has a known angle to the road, like 90°, the scanner splits the area of interest into small slices. Depending on the resolution of the oc-

cupancy grid, number of slices per parking spot width and initial accuracy of the detection taking non-centered parked cars in account the slice width can be configured.

The generated virtual slices are then used to accumulate the cells in the occupancy grid map and derive a probability of their overall occupancy status belonging to a parked car or not. This can also be used to achieve robustness against systematic outlier of sensor measurements when only a single sensor is facing the region of interest.

A simple sliding window operation is used to aggregate unoccupied empty slices to unoccupied parking slots. Using this principle and based on the slice resolution the detection is robust against wrongly parked cars, e.g. out of the slot markings.

As a drawback, this approach requires at least one parked car in an adjacent parking space, to adjust itself. Otherwise a very precise modeling of the first parking space using Roadgraph information is mandatory.

5.5 Parking-Spot Tracking and Parking Position Refinement

Keiji Mimura, Robert Bosch GmbH

In order to achieve a high robustness for parking regardless possible localization errors (based on visual localization or odometry) as well as a high flexibility to park in the middle of an available parking-spot regardless positions or sizes of neighbor cars, an algorithm to adjust an actual target-pose for parking task based on grid-map is implemented in the off-lane planner.

The main focus of this algorithm is to estimate the bounding lines of the parking-spot depending on the detected contour of neighbor cars located on both sides of a target parking-spot. Rough description of the algorithm is as follows (Fig. 5.4.1):

- Several scanning lines in a lateral direction to a heading of an originally assigned target-pose with a certain interval are extended from a center-line of the originally assigned target-pose to both (right and left) directions by a certain step.
- At each extending cycle of scanning lines, the number of object points included between origins of the scanning lines and the end of the lines are counted.
- Once enough scanning lines exceed a threshold for the number of object points and positions of these lines are spatially distributed, then extending is stopped.
- The frontier of scanning lines, when the extending is stopped, is defined as a bounding contour of a neighbor car.

- When surfaces on the both sides (right and left) are estimated, then the middle of these lines is defined as a candidate of an adjusted target-pose.
- If a distance between the newly defined target-pose and the originally assigned target-pose is larger than a threshold, and the distance keeps larger than the threshold for next some calculation cycles, then the newly defined target-pose is taken as an adjusted target-pose for the off-lane planner.

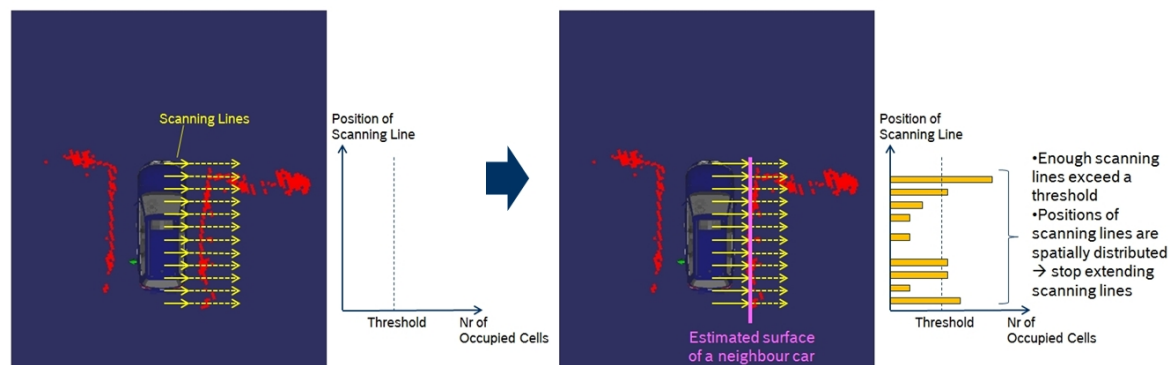


Fig. 1 Working Principle of Target-Pose-Shift Function

6 Conclusion and Outlook

This deliverable reports on the current status of the online perception and environment reconstruction within the V-Charge project. The development is controlled and verified using 15 predefined scenarios the research vehicles have to detect and react to in a most natural way. As a side aspect only low-cost sensors such as cameras and ultrasonic sensors shall be used, pushing the development of image processing and sensor fusion approaches to a new state-of-the art.

A lot of improvements have been shown especially in the sensor preprocessing domain such as the dense motion stereo or object detection and classification system on fisheye images. However in order to deal even with complex scenarios such the overtaking and continuously tracking of pedestrians around the car or detection of crossing traffic only based on wide angle mono cameras are challenging tasks. Especially when taking a robust integration and demonstration in the final presentation of the V-Charge project into account.

These are key aspects and sub-tasks for final year within this V-Charge work package regarding online perception.

7 References

- [1] "Data Distribution Service Portal," [Online]. Available: <http://portals.omg.org/dds/>.
- [2] Elektrobit, "ADTF Automotive Data and Time-Triggered Framework," Elektrobit, [Online]. Available: <http://automotive.elektrobit.com/driver-assistance/eb-assist-adtf>.
- [3] "The Robot Operating System (ROS)," [Online]. Available: <http://www.ros.org/>.
- [4] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.," *International Journal of Computer Vision*, no. 47, pp. 7-42, 2002.
- [5] R. T. Collins, "A space-sweep approach to true multi-image matching.," in *Computer Vision and Pattern Recognition - Proceedings CVPR'96*, 1996.
- [6] R. Yang and M. Pollefeys, "Multi-resolution real-time stereo on commodity graphics hardware.," in *Computer Vision and Pattern Recognition, Proceedings*, 2003.
- [7] D. Gallup, J.-M. Frahm, P. Y. Q. Mordohai and M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *In Computer Vision and Pattern Recognition, CVPR'07 - IEEE Conference on*, 2007, pp. 1-8.
- [8] D. Geronimo, A. M. Lopez, A. D. Sappa and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems.," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol. 32, no. 7, pp. 1239-1258, 2010.
- [9] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," *International Conference on Robotics and Automation - Proceedings of the IEEE* , pp. 116-121, 1985.
- [10] D. Pfeiffer, A. Barth and U. Franke, "A. Robust and Precise 3D-Modeling of Traffic Scenes based on Dense Stereo Vision," *6th Workshop Fahrerassistenzsysteme*, pp. 11-20, 2009.
- [11] Y. Bar-Shalom, K. Chang and H. Blom, "Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm,"

IEEE Transactions on Aerospace and Electronic Systems, vol. 25, no. 2, pp. 296-300, 1989.

- [12] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, p. 401–422, 2004.
- [13] T. Kubertschak, M. Maehlich and H.-J. Wuensche, "Fences - A Unified Architecture for Mapping Static Environments for Driver Assistance Systems," in *Tagungsband 9. Workshop Fahrerassistenzsysteme (FAS)*, Berthold Faerber, ed., 2014.