Stereovision-Based Sensor for Intersection Assistance

S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Bota, Technical University of Cluj-Napoca,

M.-M. Meinecke, M. A. Obojski, Volkswagen AG

Abstract

The intersection scenario imposes radical changes in the physical setup and in the processing algorithms of a stereo sensor. Due to the need for a wider field of view, which comes with distortions and reduced depth accuracy, increased accuracy in calibration and dense stereo reconstruction is required. The stereo matching process has to be performed on rectified images, by a dedicated stereo board, to free processor time for the high-level algorithms. In order to cope with the complex nature of the intersection, the proposed solution perceives the environment in two modes: a structured approach, for the scenarios where the road geometry is estimated from lane delimiters, and an unstructured approach, where the road geometry is estimated from elevation maps. The structured mode provides the parameters of the lane, and the position, size, speed and class of the static and dynamic objects, while the unstructured mode provides an occupancy grid having the cells labeled as free space, obstacle areas, curbs and isles.

1 Introduction

Intersections are the most complex traffic situations, as they can be both confusing and dangerous. The standard vehicle trajectories and the standard road geometries that are covered by most of the driving assistance systems do not apply in most of the intersections, but, on the other hand, most of the traffic accidents happen there. Due to the demanding nature of the scenario, the research community is increasingly focused on solving the perception and acting problems related to the intersection. The European research project INTERSAFE, part of the broader project PReVENT (http://prevent.ertico.webhouse.net/en/prevent_subprojects/intersection_safety/intersafe/), had the goal to develop a system that was able to perceive the relative vehicle localisation, path prediction of other objects and communication with traffic lights, in order to warn the driver and simulate active measures. A new joint research project, INTERSAFE-2 (www.intersafe-2.eu), aims at developing a system for cooperative sensor data fusion, based on state of the art passive and active on-board sensors, navigation maps, information from other traffic participants and from intelligent infrastructure, in order to generate a complete and accurate description of the complex intersection environment.

The dense stereovision sensor is maybe the sensor that provides the highest amount of usable information, as it combines the visual data with the dense 3D information that can be deduced through precise inference from the binocular view. A reliable stereovision sensor for urban driving assistance has been developed by our team [1]. However, the intersection scenario has some specific demands from a stereo sensor, demands that impose changes in the physical setup and in the software algorithms.

A wide field of view is essential for detecting and monitoring the relevant objects at an intersection, but that means lower focal length and consequently reduced working depth for stereo algorithms, and increased lens distortions in the images. The calibration algorithms, combined with an accurate image rectification step, must ensure that the quality of the reconstruction is not affected. The stereovision correlation process has to be performed on rectified images, by a dedicated stereo board to free processor time for the high-level algorithms. Even there are some methods reported in the literature for dense stereo systems calibration [2], [3], their accuracy is not well assessed. Therefore, a calibration method derived from previous approaches used for high precision and far range stereo-vision is proposed [4], [5], [6]. Camera parameters are estimated on the full-size images. For the extrinsic parameters calibration lenses distortion correction is performed in advance. This ensures the highest possible accuracy of the estimated parameters (which depends on the accuracy of the detected calibration features). The full-size images are further rectified, un-distorted and down sampled in a single step using reverse mapping and bilinear interpolation [7] in order to minimize the noise introduced by the digital rectification and image correction.

Due to the complex nature of the intersection, a solution for perceiving the environment in two modes is proposed: a structured approach, for the scenarios where the road geometry is estimated from lane delimiters, and an unstructured approach, where the road geometry is estimated from elevation maps.

Lane detection is one of the basic functions of any sensorial system dedicated to driving assistance and safety. The intersection scenario is the most difficult one for a lane tracking system, as many of the assumptions that the classical solutions rely on are not valid anymore. The good condition of the lane markings and the continuous nature of the road geometry on the highways lead to the first lane detection solutions, based on Kalman filtering, one of the first ones being presented in [8]. Since then, the Kalman filter was used, with some changes, for more difficult conditions. The solution presented in [9] used a backtracking system to select the measurement features in such a way that the false clues such as shadows or cracks in the road could be overcome (and thus the system could work with simpler delimiters, not only with markings). 3D lane estimation algorithms using stereo information in 3D space is presented in [10]. The solution presented in [11] uses the Kalman filter along with an elaborate selection of lane delimiters in the form of lines, in order to overcome the large quantity of possible lane delimiting cues and the short viewing distance that define the urban environment. All the Kalman filter solutions expect, more or less, a continuously varying road situation, and the discontinuities are usually handled by reinitializing the tracking process. The solution presented in [12] handles some particular case of road discontinuities by using two instances of the road model, but it is clear that the Kalman filter is not the best choice for tracking discontinuous roads.

Since the particle filter has been introduced in visual tracking under the name of condensation [13], several lane tracking solutions have been based on it. In [14] a lane detector based on a condensation framework, which uses lane marking points as measurement features is presented. Each point in the image receives a score based on the distance to the nearest lane marking, and these scores are used to compute the matching score of each particle. The system uses partitioned sampling (two-step sampling and measurement using subsets of the state space, achieving a multiresolution effect), importance sampling, and initialization samples (completely random samples from the whole parameter space) which cope faster with lane discontinuities. In [15] we find a lane detection system that uses the particle filtering framework to fuse multiple image cues (color, edges, Laplacian of Gaussian). This solution also uses initialization samples for faster lane relocation, and additional sampling around the best weighted particles for improvement of accuracy. The much simpler way in which a particle filter handles the measurement information allows the use of a wider range of cues. Such is the case of the lane detector for country roads, presented in [16], where the image space is divided into road and non-road areas and each pixel in these areas contribute to the final weight by its intensity, color, edge and texture information. The work presented in [17] also shows the value of the particle filtering technique for heterogeneous cue fusion, when image information is fused with GPS and map information for long distance lane estimation.

In [18] a lane tracking technique based on stereovision and particle filtering for handling discontinuous urban scenarios is described. For the intersection applications, the proposed lane detection system based on stereovision and particle filtering had to be extended and reorganized, so that it can handle multiple lane models and multiple measurement cues. These changes are designed to increase the robustness of lane tracking through the intersection, and give the system capability to handle various lane geometries. Due to the modular and configurable nature of the tracking system, we aim to extend it to detect other painted road structures, such as arrows and pedestrian crossing signs.

In structured environments, the obstacles are modeled as cuboids, having position, size, orientation and speed. From the top view of the dense stereo data, the cuboid reconstruction algorithms analyze the local density and vicinity of the 3D points, and determine the occupied areas which are then fragmented into obstacles with cuboidal shape: without concavities and only with 90^o convexities. The orientation of the obstacles also is determined in order to get a very good fitting of the cuboidal model to the obstacles in the scene ahead and, consequently, to minimize the free space which is encompassed by the cuboids. The cuboids are tracked using a Kalman filter-based approach, in order to reduce the measurement noise and improve the stability of the results, and to measure the object speed. An experimental tracking algorithm, based directly on 3D raw data, is also proposed in this paper.

An intersection assistance system is required to make decisions based on all the relevant factors present at the intersection environment. One important factor is the class of obstacles involved, either mobile or belonging to the intersection infrastructure. Relevant object classes include vehicles, pedestrians, bicyclists, curbs, pillars, trees and traffic signs. Discriminating between these object classes provides opportunities for implementation of comfort functions and active safety functions in intelligent vehicles.

Pedestrians are especially vulnerable traffic participants. Therefore, recognizing the pedestrian class is extremely important, and a lot of scientific research has been conducted in this area. Classification methods are traditionally limited to 2D image intensity information. A contour matching approach is described in [19]. Another technique using Adaboost with illumination independent features is described in [20]. There are some classification methods that use 3D information directly like [21] and [22]. The use of template matching in conjunction with 3D information has not been extensively explored. Some approaches aimed at pedestrian detection have used dense 3D information, but only as a validation method [23]. Another important feature for walking pedestrian detection is their walking pattern. There are a number of works, e.g. [24], [25] which have used motion cues in pedestrian detection. Other types of information such as IR images have also been used, as described [26].

In this paper we present a multi-feature, multi-class approach to classification. We use 2D intensity image information, 3D information and motion in order to obtain a reliable classification. Our detected classes include pedestrians, cars and poles. The architecture of our classifier allows us to easily incorporate more feature and more classes.

Intersections in urban areas often have no lane markings or these are occluded by crowded traffic. The lanes cannot be detected in such particular scenes, thus the obstacle/road separation becomes difficult. An alternative/complementary method must be used to detect elevated areas (obstacles), regions where the ego vehicle cannot be driven into. Complementary, the obstacle-free road areas can be considered as drivable. Starting from the representation of the dense 3D data as an elevation map, an occupancy grid which defines the drivable area, traffic isles and obstacles is proposed and implemented.

Digital elevation maps (DEMs) are less used to represent and process 3D data from stereovision, but they offer a compact way to represent 3D data, suitable for real-time processing. A complex method for building the digital elevation map of a terrain (for a planetary rover) is proposed in [27]: local planar surfaces are used to filter the height of each DEM cell, and the stereo correlation confidence for each 3D point is included in the filtering process. In [28] the elevation map is built straightforward from the disparity map. The authors avoid using a 3D representation of the reconstructed points by projecting a vertical 3D line for each DEM cell onto the left, disparity, and right image. Based on these projections, the disparity of the point associated with the cell is selected and possible occlusions are detected. The obtained DEM is used for a global path-planning algorithm.

The road, traffic isle and obstacle detection algorithm presented in [29] uses digital elevation maps to represent 3D dense stereo data. This method is suitable for intersection scenarios, since it does not rely on high-gradient road features (lane markings) for the road surface detection. However, if the road surface is not detectable (poor texture resulting in few 3D points) an alternative to detect curbs (traffic isle borders) was designed and presented in [30]. A combination of these two approaches, more robust and suitable for intersection scenes, is presented in this paper.

The output provided by the stereo sensor supports intersection driving assistance systems in multiple ways: detection of the approaching intersection, intersection sensorial modeling, relative localization of the ego vehicle, precise alignment of the intersection sensorial model and the corresponding intersection map, obstacle detection, tracking and classification, collision prevention and collision mitigation.

2 Stereo system setup and calibration

2.1 Stereo system architecture

The proposed stereovision system architecture (Fig. 1) is derived from our previous experience gained with urban driving assistance application based on dense-stereovision [1]. Adjustments were made in order to fit the requirements of intersection assistance applications. E.g. a very large horizontal field of view is compulsory for such applications which imply short focal length lenses. Short focal lenses are introducing pronounced radial lens distortions which should be corrected before the stereo reconstruction. The intersection is mainly an unstructured road environment (sometimes, there are no lane delimiters, only curbs) and therefore a dense 3D points' map is required for providing rich information.

The architecture has a versatile and hybrid structure: some modules can be implemented on dedicated hardware boards or as software libraries in a conventional PC architecture.

Two monochrome (grayscale) imagers mounted in a quasi-canonical configuration are used. The image sensors can be integrated in a dedicated stereo-head or built from stand-alone cameras mounted rigidly on a stereo-rig. The stereo cameras (Fig. 1- module 1) can be interfaced (Fig. 1- module 2) trough a frame-grabber, a dedicated hardware interface or a common PC interface (Fierwire, USB). Image acquisition is controlled by a software interface (Fig. 1– module 3) specific for the used cameras. The quality of the acquired images is controlled by tuning the Gain and Exposure parameters of the cameras through an auto-adaptation to the lighting condition module (Fig. 1– module 4).

The stereo images are acquired at the full resolution of the image sensors. They are further rectified (to meet the canonical configuration), corrected (to eliminate the lens distortions) and down-sampled (to meet the processing capacity of the hardware dense-stereo reconstruction engine – e.g. 512 pixels in width). This process (Fig. 1 – module 5) can be performed on dedicated hardware or implemented in software. The software implementation of this process is reduced to an image warping approach performed in a single step using reverse mapping and bilinear

interpolation [7] in order to minimize the noise introduced by the digital rectification and image correction. The implementation is optimized for processing speed-up using MMX instructions and lookup tables.

The rectified (canonical) down-sampled images are fed to the stereo reconstruction module (Fig. 1 – module 6) which is implemented in hardware. From the depth map provided by the stereo-engine the 3D coordinates of reconstructed points (Fig. 1 – module 7) are computed knowing the camera parameters and are further provided to the processing modules (Fig. 1 – module 8).

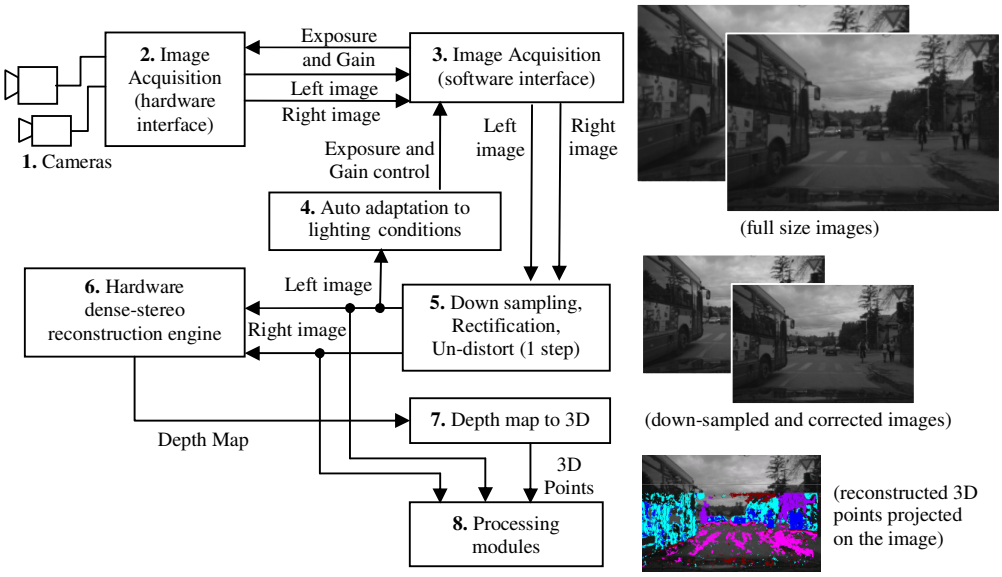


Fig. 1. The stereovision system architecture

2.2 Stereo cameras setup

The tuning of the image acquisition system to specific application requirements can be done by choosing the proper parameters of the stereo cameras: focal length of the lenses, imager resolution and size, baseline length.

Intersection assistance applications require a very large horizontal field of view (HFOV) but in the same time an accurate 3D reconstruction in a depth range from 0 m up to 20 … 25 m, measured from the front of the ego car. Spherical lenses (with HFOV > 90°) do not comply with this detection range. Therefore pinhole lenses with short focal lenses were chosen. Having the focal length as a known constant, the detection range is adjusted by choosing the proper baseline length (the horizontal displacement between the two cameras). The lowest limit of the depth range is computed with the formula bellow, while the upper limit is determined experimentally.

$$Z_{MIN} = \frac{b \cdot f}{d_{MAX}} \approx 0 \qquad (1)$$

where:

Z_{MIN} – lower limit of the detection range
b – is the baseline (horizontal displacement between the cameras) [mm]
f – focal length [number of pixels]
d – horizontal disparity between reconstructed features [pixels]. The values of d_{MAX} is stereo engine dependent.

Two setups are proposed (Table 1):

1. The first one uses an integrated stereo-head along with a dedicated board that performs the image acquisition along with the automatic exposure and gain control, image down-sampling, correction and rectification, and dense stereo reconstruction (Fig. 1 - modules 2, 4, 5 and 6). This setup has the advantage of using a compact size, cheap stereo head but provides medium quality images due to the CMOS imager and miniature lenses.
2. The second one uses a stereo-head built from stand-alone cameras with 1.3 megapixel CCD sensors and high quality lenses, mounted rigidly on a stereo-rig. The cameras are interfaced with the PC with a dedicated frame grabber trough the standard CameraLink interface. Functionalities of modules 4, 5 (Fig. 1), are implemented in

software libraries, while the dense stereo-reconstruction (Fig. 1 - module 6) is performed on the same dedicated hardware board. This second setup has the disadvantage of larger stereo-head size and higher system price, but provides the best image quality.

Table 1 Comparison of the two stereo setup parameters

System no.	**1**	**2**
Full size image [pixels]	752x480	1376x1030
Imager size (inch)	1/3	2/3
Pixel size [mm]	6.00	6.45
Lens focal [mm]	4.00	4.80
HFOV [deg]	75	90
Focal - full size image [pixels]	618	744
Focal – down-sampled images [pixels]	421	276
Baseline - b [mm]	220	340
Max. disparity - d_{MAX} [pixels]	52	52
$Depth_{MIN}$ (camera) [m]	1.78	1.81
$Depth_{MIN}$ (car) [m]	-0.02	0.01
$Depth_{MAX}$ (car) [m]	33.68	34.14
$Depth_{MAX}$ (car) (75% $Depth_{MAX}$) [m]	25.26	25.60

2.3 Camera calibration

In order to reconstruct and measure the 3D environment, the cameras must be calibrated. The calibration process estimates the camera's intrinsic and extrinsic parameters. Both intrinsic and extrinsic parameters are estimated on the full-size images. For the extrinsic parameters calibration lenses distortion correction is performed in advance. This ensures the highest possible accuracy of the estimated parameters (which depends on the accuracy of the detected calibration features/control points).

The intrinsic parameters of each camera are calibrated individually using the Bouguet's [31] method. The accuracy of the obtained intrinsic parameters depends on the quality of the calibration object (size, planarity, accuracy of the control points' positions) and on the calibration methodology used (number of views, number of control points on each view, coverage ratio of each of the view by the calibration object etc.). If the proper conditions are fulfilled, the uncertainties of the estimated parameters are below one pixel for the focal length and principal point.

For the extrinsic parameters estimation an original method dedicated for high range stereo-vision was developed [4]. The estimated parameters are the $\mathbf{T_L}$ and $\mathbf{T_R}$ translation vectors and $\mathbf{R_L}$ and $\mathbf{R_R}$ rotation matrices of each camera. The extrinsic parameters calibration method assures not only very accurate absolute extrinsic parameters of each camera individually but also very accurate relative extrinsic parameters, which allows a precise estimation of the epipolar lines (near 0 pixel drift) [5] with benefic effects for the stereo matching process. Therefore the method is suited for the calibration of any stereovision system used for far or mid range 3D reconstruction like in vision based driving assistance systems.

3 Structured approach for environment description

3.1 Lane detection

The lane tracking system for intersection safety applications was designed in a modular fashion, the equations for the vertical and horizontal profile being easily configurable, as lanes at the intersections have varying geometries. The measurement process is independent on the 3D model, as long as sets of 3D points for the delimiters are available.

We have designed a general-purpose particle filtering framework that is the basis for the lane tracking system, and that can be easily extended to cover multiple situations (Fig. 2). The *abstract particle* class contains the matrices and vectors that can describe any process, and provides interface for specific methods of initialization, measurement, drift and specifics. The *particle set class* aggregates particles regardless of their specifics, and implements the

Resample – Drift – Measure – Estimate cycle by calling individual particle methods. This class also provides abstract interface for specific calculations, such as computing the trajectory curvature from CAN sensor data.

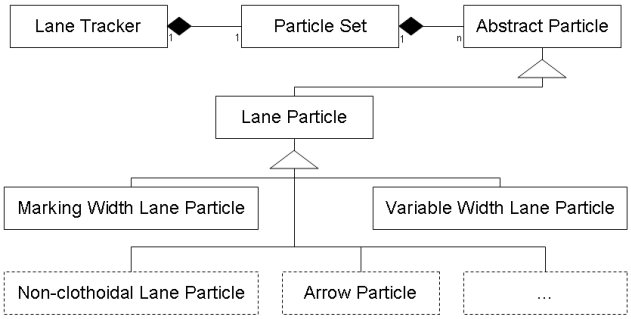


Fig. 2. Base hierarchy for developing lane detection solutions

Building a lane detection and tracking system based on this framework means extending the Abstract Particle class with a Lane Particle class, that will implement the lane specifics, such as projecting the 3D lane model into the measurement data space, or the measurement itself, which means weighting the lane hypothesis particle against the measurement data.

The model-independent measurement process relies on projecting chains of points for each lane delimiter in the image space, and use the distance transform for the measurement cues to compute a distance between the cues and the lane projection. Measurement cues can be lane markings, image edges that are located on the road surface, or projections of the curb points detected using dense stereo analysis. For each delimiter we use two rows of points, one row that is located on the delimiter (the delimiter points) and one row located inside the lane surface, near the delimiter (the inner points). Ideally, we would like to have a minimum distance for the delimiter points (they should fall on the delimiting features) and a maximum distance for the inner points. The two distances are combined, and a probability measure that is higher as the distance is lower is computed. A more detailed explanation of the measurement process is given in [18].

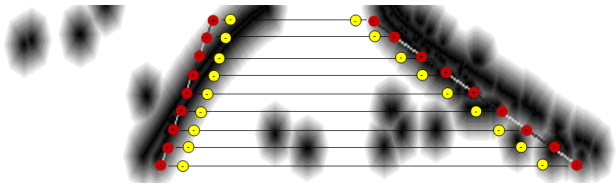


Fig. 3. Boundary and inner points. For the boundary points, the distance is added, and for inner points the distance is subtracted

Using a new 3D lane model implies that we modify only the initialization of the vectors, the building of the state transition matrices, and the factory method. A specific function GetLaneXY that retrieves the X and Y positions given Z and the model is used to decouple the measurement from the 3D model. Implementing the system based on the new model, by inheritance of the CLaneParticle class and modification of the specifics of the model takes about 100 lines of code, mainly declarations.

We have implemented two extensions from the classical lane description (as used in [18]): modeling the lane delimiter's marking width and modeling a variable lane width. For modeling the lane marking width, the lane state vector had to be enhanced with two more parameters, and the measurement process had to include an additional cue – a horizontal gradient image computed with a perspective-aware kernel, which allows for a smooth transition towards the maximum. The gradient image itself was used as a distance image. As a side effect, estimating the marking width in this manner improved the lane estimation stability in cases when the estimation was difficult (such as when only one delimiter is present).

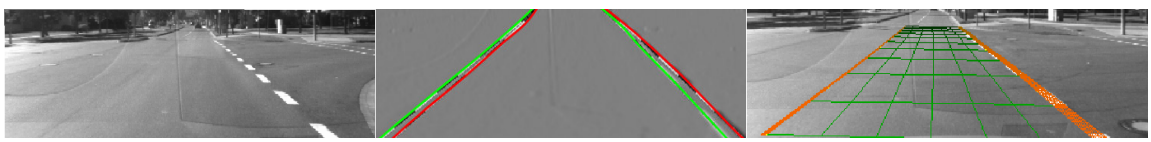


Fig. 4. Estimation of the lane marking width

Modeling and estimating a lane with variable width required much less than modeling the lane marking width, because the measurement system could be reused completely. Only the functions describing the 3D model needed to be overloaded.

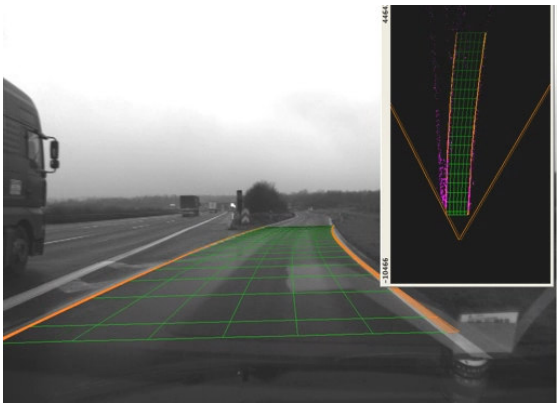


Fig. 5. Variable width model

The lane estimation framework based on particle filters will be extended in the future towards detecting target lanes (lanes we are not on yet), and this will require changes in the model's restrictions and changes in the measurement process (we have to rely more on horizontal features). Also, a future use will be to detect "Capped" lanes, which are lanes shorter than the detection distance, ending with a horizontal line, that usually mark the beginning of the intersection. They require a change in the model (addition of the length) and in the measurement (addition of the ending cue).

3.2 Obstacle detection and tracking

Starting from the 3D points situated above the 3D profile of the lane, obstacles described by cuboids, having position, orientation and size, are detected. The confident fitting of cuboids to obstacles is achieved in several steps. By analyzing the vicinity and the density of the 3D points, the occupied areas are located. An occupied area consists of one or more cuboidal obstacles that are close to each other. By applying shape criteria, the occupied areas may get fragmented into cuboidal parts. The orientation of the obstacles on the road surface is extracted as well.

The 3D points above the height of the ego vehicle are discarded (Fig. 6.b). It is supposed that the obstacles do not overlap each other on the vertical direction. In other words, on a top view (Fig. 6.c) the obstacles are disjoint. Consequently, in what follows the elevation of the 3D points will be ignored and all the processing is done on the top view.

Due to the perspective effect of the camera, further obstacles appear smaller in images, providing fewer pixels, and therefore, less and sparser 3D reconstructed points in the 3D space. On the other hand, the error of the depth reconstruction increases with the distance too, which, contributes to the 3D points sparseness as well. To counteract the problem of the points' density, a schema to divide the Cartesian top view space into tiles with constant density is proposed (Fig. 6.c). The horizontal field of view of the camera is divided into polar slices with constant aperture, trying to keep a constant density on the X-axis. The depth range is divided into intervals, the length of each interval being bigger and bigger as the distance grows, trying to keep a constant density on the Z-axis.

A specially compressed space is created, as a matrix (Fig. 7.a). The cells in the compressed space correspond to the trapezoidal tiles of the Cartesian space. The compressed space is, in fact, a bi-dimensional histogram, each cell counting the number of 3D points found in the corresponding trapezoidal tile. For each 3D point, a corresponding cell C (*Row*, *Column*) in the compressed space is computed, and the cell value is incremented.

The column number is found as:

$$Column = ImageColumn / c \quad (2)$$

where *ImageColumn* is the left image column of the 3D point and c is the number of adjacent image columns grouped into a polar slice as shown in Fig. 6.c ($c = 6$).

The depth transformation, from the Z coordinate of the Cartesian space into the *Row* coordinate of the compressed space has a formula obtained using the following reasoning:

a) The Cartesian interval corresponding to the first row of the compressed spaces is:
$[Z_0 \dots Z_0 + IntervalLength(Z_0)] = [Z_0 \dots Z_1]$ (3)

where $Z_0=Zmin$, the minimum distance available through stereo reconstruction. The length of the interval beginning at a certain Z is
$IntervalLength(Z) = k*Z$ (4)

k being empirically chosen). Thus
$Z_0 + IntervalLength(Z_0) = Z_0+k*Z_0 = Z_0*(1+k) = Z_1$ (5)

b) The Cartesian interval corresponding to the nth row of the compressed spaces is $[Z_n \dots Z_n + IntervalLength(Z_n)]$, where $Z_n=Z_0*(1+k)^n$ (6)

The above equation can be proven by mathematical induction.

c) For a certain 3D point, having depth Z, the ith interval it belongs to is $[Z_i \dots Z_i + IntervalLength(Z_i)] = [Z_i \dots Z_{i+1}]$.

From equation (6), we can find the row number as the index of the point's interval

$$Row = i = [log_{1+k} \frac{Z}{Z_0}] \quad (7)$$

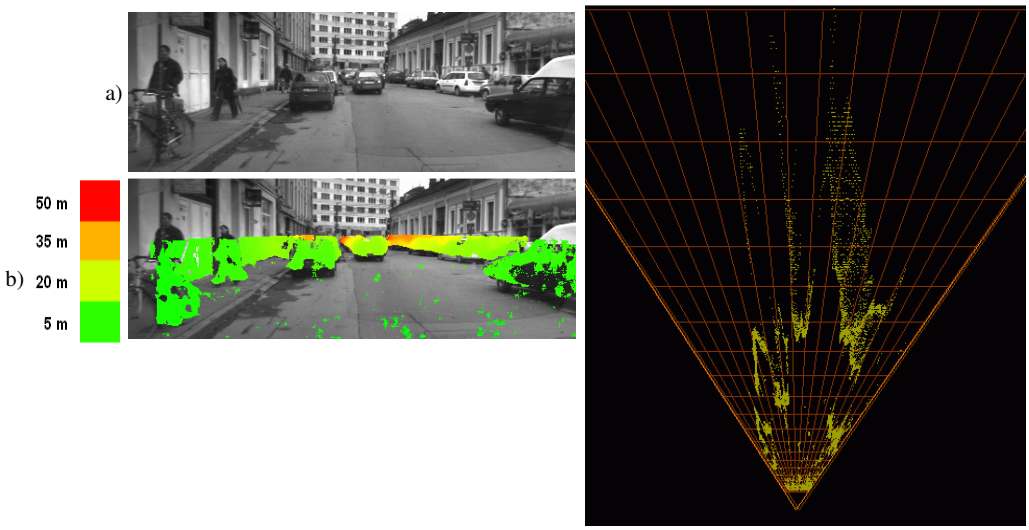


Fig. 6. Division into tiles. a) Gray scale image, b) 3D points – perspective view, c) 3D points – top view; the tiles are here considerably larger for visibility purpose; wrong reconstructed points can be seen in random places; reconstruction error is visible as well

The histogram cells that have a significant number of points indicate the presence of obstacle areas. On these cells, a labeling algorithm is applied, and the resulted clusters of cells represent occupied areas (Fig. 7.b). The small occupied areas are discarded in this phase.

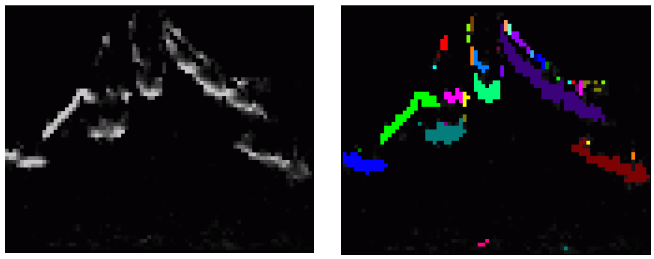


Fig. 7. The compressed space (for scene in Fig. 6) – a bi-dimensional histogram counting 3D points. Occupied areas are identified by cell labeling

The occupied areas may contain several obstacles and by consequence they may have miscellaneous shapes. The obstacle tracking algorithms as well as the driving assistance applications need the individual cuboidal obstacles. Therefore the fragmentation of the occupied areas into the individual cuboidal obstacles is required.

An indication that an area requires fragmentation is the presence of concavities. In order to detect the concavities, the envelope of the cells of an occupied area is drawn, and then for each side of the envelope, the gap between the side and the occupied cells is determined. If the gap is significant, we consider that two or more obstacles are present, and the deepest point of the concavity gives the column where the division will be performed. The two sub-parts are subject to be divided again and again as long they have concavities.
In Fig. 8.c the bottom side of the envelope for the cells in Fig. 8.b delimits a significant concavity. For each new sub-part, the envelope of the cells has been calculated again (and painted as well), but without revealing big concavities for new divisions to be performed.

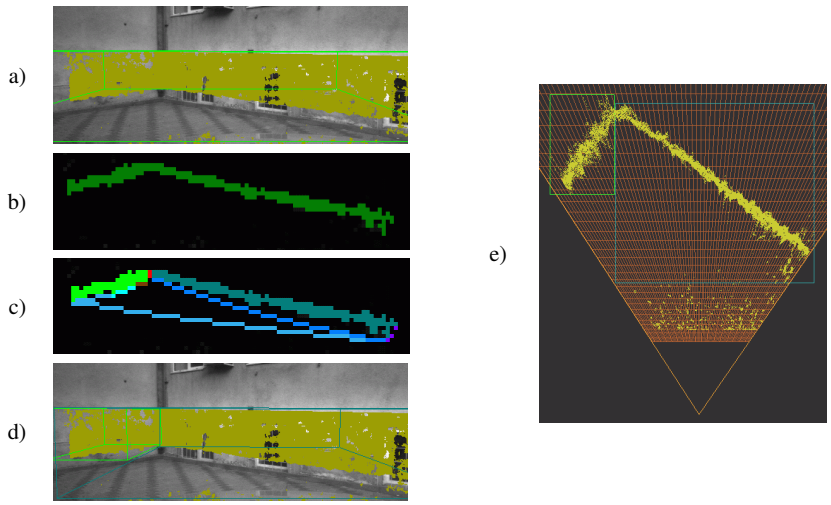


Fig. 8. Fragmentation of occupied areas into primitive obstacles. a) an occupied area, b) the labeling in the compresses space, c) sides of the envelope and the two primitive obstacles of the occupied area – compressed space, d) the two primitive obstacles – perspective view, e) the two primitive obstacles – top view

By reconsidering the coordinates (including the height) of the 3D points that have filled the cells of an obstacle, the limits of the circumscribing box are determined. Boxes are shown in Fig. 8.d (perspective view) and Fig. 8.e (top view).

The next step is obstacles orientation estimation. In Fig. 9.a, it can be observed that, even though the real obstacle is oblique oriented, the cuboid encompasses some free space because the cuboid is parallel with the coordinate system. The shape of the cloud of 3D points is modeled by their envelope and an analysis of its visible sides (towards the camera) can estimate the orientation of the obstacle. If the analysis cannot determine a preponderant orientation of these sides, the box remains un-oriented with sides parallel with the axes of the coordinates system.

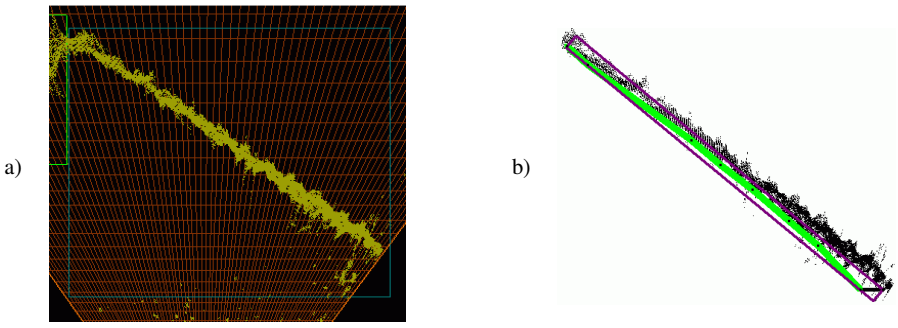
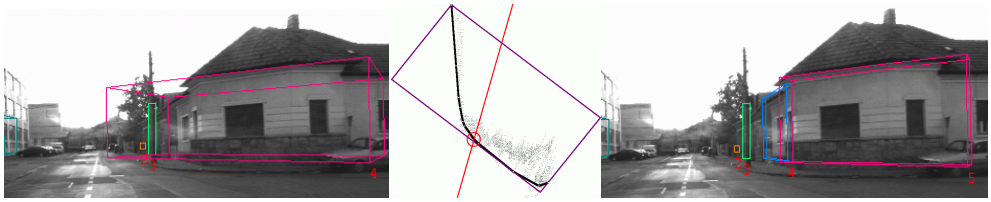


Fig. 9. Obstacle orientation (top view): a) un-oriented box,
b) the longest chain of visible envelope sides (green) and its surrounding rectangle

Another case when an occupied area must be fragmented is signaled by the following observation. A real obstacle is accurately detected when the space between the sides of its oriented cuboid and the visible shape of the cloud of its 3D points (on the top view) has a small area. When a free space of more than 10 cells (an empirically chosen threshold) between the cuboid corners and the inner cloud appears a fragmentation into two or more sub-obstacles is necessary.



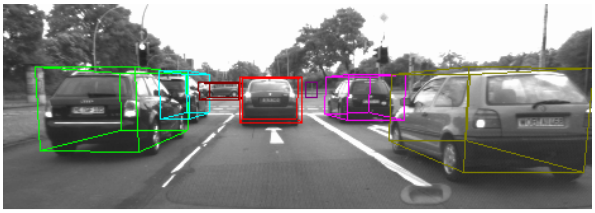
a) b) c)

Fig. 10. Fragmentation of an obstacle that doesn't have a cuboidal shape: a) initial cuboid, b) top view of the initial cuboid and the optical ray of the fragmentation, c) the two sub-obstacles obtained by fragmentation

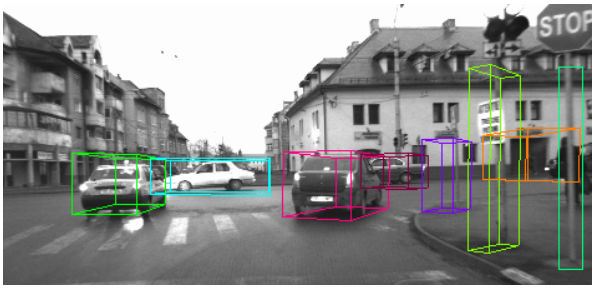
An example is shown in Fig. 10.a and b, where no single oriented cuboid would fit. The fragmentation is done by splitting along an optical ray that passes through one of the sharp vertices of the unoccupied triangle. The procedure is recursively applied on the newly obtained obstacles if needed. The fragmentation of the large cuboid in Fig. 10.a is shown in Fig. 10.c.

A more detailed description of the obstacle detection system is given in [32]. Some examples, showing obstacle detection in intersection scenarios, are presented in Fig. 11:

a)



b)



c)

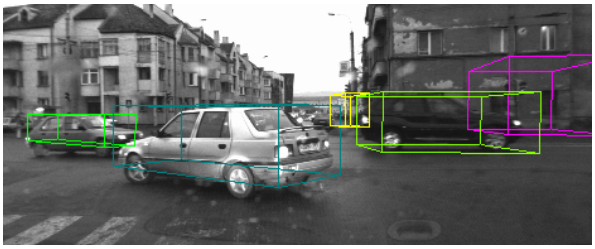


Fig. 11. Results on obstacle detection in intersection scenarios

As a major resource for improving the obstacle detection accuracy for complex intersection scenarios the use of color images and optical flow are being studied.

Tracking is the final stage of the stereovision based object detection system. At the end of this step, the sensor outputs the cuboid list containing parameters of position, size, orientation and speed. For tracking, we have two methods: an established solution based on the Kalman filter [33], which uses as measurement the cuboids extracted using the methods described in the previous paragraphs, and an experimental one, based on particle filters, which uses raw 3D obstacle data.

The new solution is based on multiple particle filters, and does not require points grouping, working directly on the raw obstacle data projected on the top view, data which can be provided, for instance, by elevation map processing [29]. This solution is a multiple object tracking scheme that uses a two-level approach. First, a particle filter-based tracker that starts with an initial probability density function that covers the whole possible object parameter state space runs freely on the 3D data, and its particles will tend to cluster around one object. When the clustering is achieved, the particle state is passed to an individual object tracker, that will track the object until it disappears, and the initialization tracker is reset and will start searching for another object.

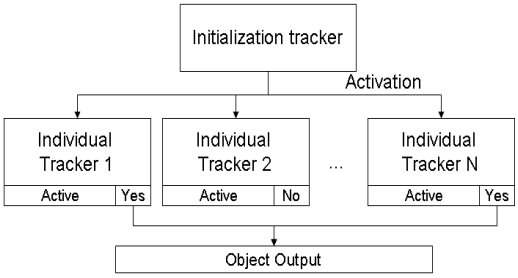


Fig. 12. Top-level architecture of the particle filter-based tracking system

Although the individual trackers and the initialization tracker are built around similar particle filters, their behavior is different. Fig. 13.a shows the flowchart for the initialization tracker. We'll briefly describe each block in the diagram.

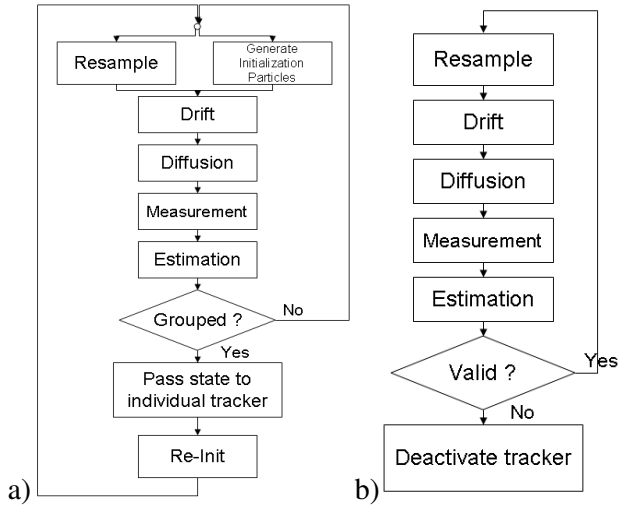


Fig. 13. a) Flowchart of operation for the initialization tracker, b) Flowchart of operation for the individual object tracker

Resample – the process that transforms the weighted particle set describing the past probability density into an equivalent set of weightless particles, by random weighted sampling.

Generation of initialization particles – a fraction of the particle set will be sampled from the initial probability distribution, that is, from the whole range of possible object states. These particles help the tracker evade a false hypothesis faster.

Drift – the uniform motion dynamic model is applied to the particles (details in the next section)

Diffusion – a random value is applied to each particle, accounting for the uncertainties of state transition.

Measurement – each particle is compared to the measurement data, and weighted according to a fitness score.

Estimation – an estimated object state is computed by weighted average of the particle values. This step also computes a standard deviation value for each estimated parameter.

"Grouped" decision - The initialization tracker uses only the standard deviations of the object position to decide whether the particles are grouped. If they are not grouped, the tracker starts another cycle. If they are grouped, the whole particle distribution is passed to an individual tracker, which is made active, and the initialization tracker is reset.

The operation of the active individual object tracker is depicted in Fig. 13.b. The basic blocks are similar to the initialization tracker, but there are some notable differences. First, there are no initialization particles, as these trackers operate at local level, their target already selected. Second, after estimation the next block is a "valid" condition, which tests the grouping of the particles, the position and size of the estimated object (if grows too big, or too small, or goes out of the field of view, it becomes invalid). If the tracker becomes invalid, it goes into the inactive state, and the tracked object is declared lost.

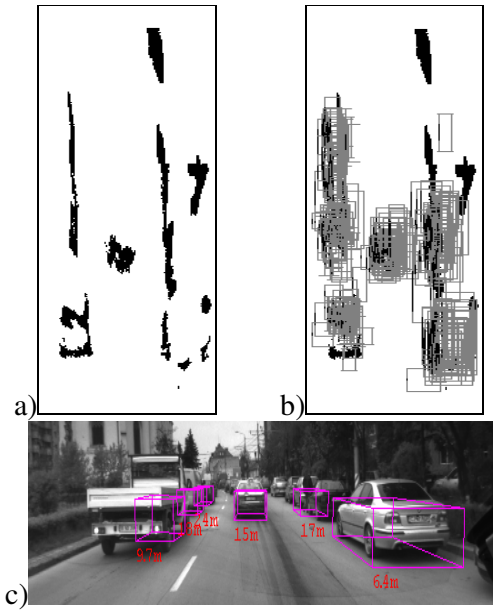


Fig. 14. a) Obstacle areas in 3D top view, b) Particles of the object trackers, c) Results (estimations) in perspective projection

3.3 Obstacle classification

Although recognition of various obstacle types is easy for humans (at least in normal visibility conditions) it proves to be an extremely challenging task for machine vision systems. In order to cope with the problem's difficulty we designed a complex, multi-class, multi-feature, multi-stage classification system. All the information supplied by the stereovision sensing approach: intensity images, range information and motion is used by our classifier. The training and assessment of the classifier was performed using an extremely large database of manually labeled stereo-data (approximately 50000 object instances). The architecture of our classifier is presented in Fig. 15.

The input of the classifier is a list of 3D tracked cuboids, the set of reconstructed 3D points and the left intensity image. The first step performed is to create depth masked images, i.e. images that capture the foreground pixels for each object (see Fig. 16). These images contain only the points which are inside the object box, thus eliminating background points. All subsequent feature extractors use only the foreground points.

A number of simple features are extracted in the first step: the 3D box dimensions (width, height & depth) and the lateral and longitudinal speed. For tracked, non-stationary objects, the motion signature feature [34] is extracted and computed. This feature represents the variance of the 3D motion vector fields. It is based on the computation of 2D optical flow vectors for corner points [35] which belong to the object (hence the need for masking) followed by their back-projection in 3D. The motion signature is large for articulated, moving objects such as pedestrians and small for rigid objects such as cars. Another feature related to the motion signature is its periodicity. Periodic motions are more reliable features, because occasionally, an object may have a large, spurious motion signature caused by noise in optical flow computation. An example of using the motion signature and motion periodicity features is presented in (Fig. 17).

All the above features apply to all object classes. The remaining features are more class specific in the sense that they help validating a hypothesized class or discriminate between two similar classes (dichotomizers). Therefore, it is unnecessary to extract these features for all objects. In order to choose which features to extract for which object, a coarse classification is performed, based on the features extracted thus far (object width, height, depth, lateral and longitudinal speed, motion signature and motion periodicity). The result of this coarse classification is a class probability distribution.

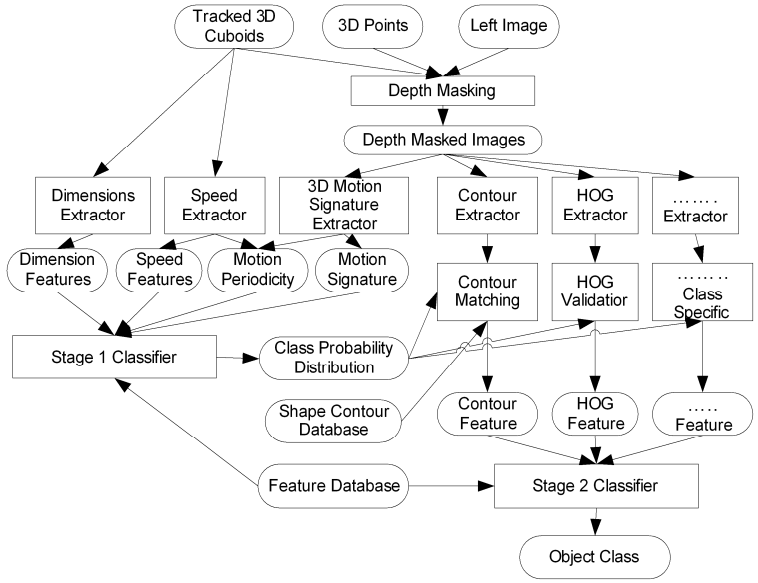


Fig. 15. Architecture of the object classifier

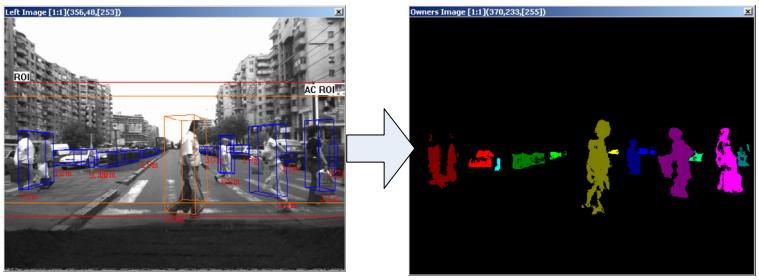


Fig. 16. Depth masking - background pixels are black, color pixels belong to objects

The remaining features are extracted only if the relevant class or classes for which they apply have a high enough probability according to the coarse classification. The first such feature is the object's contour, which is matched with a hierarchy of high-quality template contours extracted using a controlled fixed background scenario. The matching is performed using distance transform as presented in [19]. Both the template is matched with the distance transformed extracted contour and the distance transformed template is matched with the extracted contour. The two resulting distances are added together and form the contour matching feature. This ensures a more reliable matching process. We currently use contour matching only for pedestrians (Fig. 18).

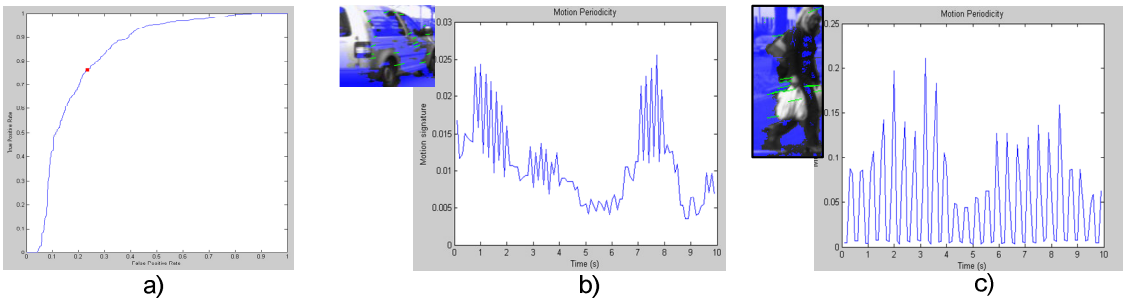


Fig. 17. a) ROC curve showing the relevance of motion signature for pedestrian recognition. b) Motion signature evolution in time for rigid objects. c) Motion signature evolution in time for pedestrians

Another feature we use is the histogram of oriented gradients [36]. This is applied for both pedestrians and cars. Because of the flexible architecture of our classifier, many more features can be added, as suggested in Fig. 15. After all these features are extracted, they enter the stage 2 (final) classifier, which picks up the class with the highest probability for each object, and outputs it.

Fig. 18. Pedestrian contour matching. Green contours are extracted from the images and yellow contours are templates matched with these contours

Both the coarse classifier and the final classifiers were trained using the WEKA environment [37] and are implemented by decision trees. We evaluated our classifier using two scenarios. In the first scenario 66% of the dataset was used for training and the remaining 34% was used for testing. The classes considered were car, pedestrian, pole and other. Out of a number of 3744 of instances, 3110 were correctly classified (83.0662%). The results are presented in Table 2. For the second scenario we performed a 10 fold cross validation using a number of 11013 instances. The "Other" class was eliminated. Out of 11013 instances, a number of 9223 (83.7465 %) instances were correctly classified. The results are summarized in Table 2.

Table 2 Classification results for scenario 1

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.899	0.095	0.823	0.899	0.859	0.966	Car
0.914	0.048	0.872	0.914	0.892	0.974	Pedestrian
0.815	0.009	0.864	0.815	0.839	0.978	Pole
0.706	0.096	0.797	0.706	0.748	0.895	Other

Table 3 Classification results for scenario 2

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.877	0.071	0.856	0.877	0.866	0.968	Car
0.909	0.048	0.869	0.909	0.889	0.974	Pedestrian
0.814	0.01	0.844	0.814	0.829	0.978	Pole

These are only preliminary results. The aim is to build a real-time classifier, with an extremely high classification rate of pedestrians, cars, poles, traffic signs, trees etc. that will provide the necessary information for an intersection assistance system.

4 Unstructured approach for environment description

Intersections in urban areas often have no lane markings or these are occluded by crowded traffic. For such scenarios a more general approach has to be used. An occupancy grid, which defines the drivable area, traffic isles and obstacles, should be computed, by using digital elevation maps to represent 3D data.

4.1 Computing the occupancy grid from elevation maps

The road, traffic isle and obstacle detection algorithm presented in [29] uses digital elevation maps (DEMs) to represent 3D dense stereo data. The DEM is enhanced based on the depth uncertainty and resolution models of the stereo sensor. A RANSAC-approach, combined with region growing, is used for the detection of the optimal road surface. Obstacles and traffic isles are detected by using the road surface and the density of 3D points. This algorithm outputs the road surface parameters and an occupancy grid (Fig. 19.c) with three distinct cell types: road, traffic isles and obstacles.

A region of interest of the 3D space is represented as a digital elevation map (DEM). The DEM is a rectangular grid (matrix) of cells and the value of each cell is proportional to the 3D elevation of the highest point (within the cell). The DEM presents poor connectivity between road points at far depths (due to the perspective effect). Connectivity is improved by propagating (to empty cells) the value of valid cells (with 3D points), along the depth (Fig. 19.b). The propagation amount is computed from the stereo geometry of the system, to compensate the perspective effect. Additional features are stored with each DEM cell, such as the density of 3D points per cell.

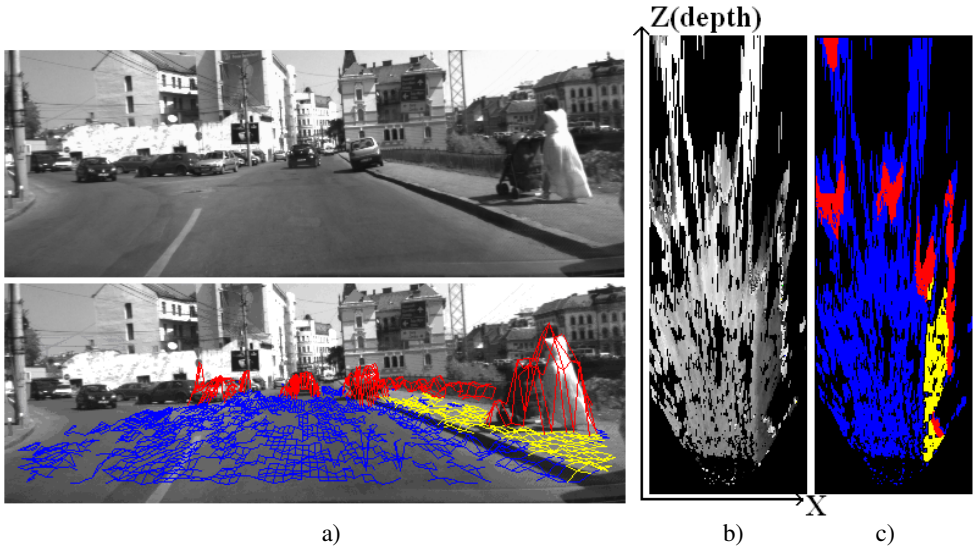


Fig. 19. a) top: A common urban scene, bottom: the occupancy grid projected onto the left image b) The elevation map c) The occupancy grid resulting after classification: road cells with blue, traffic isle cells with yellow, and obstacle cells with red

A quadratic surface is used to model the road, instead of the less general planar model. A RANSAC-approach, combined with region growing, is proposed to compute the parameters of the surface. First, the road model is fitted, using RANSAC, to a small rectangular DEM patch in front of the ego vehicle. A primary road surface is extracted optimally for the selected patch. Next, the primary solution is refined through a region growing process, where the initial region is the set of road inliers of the primary surface, from the initial rectangular patch. New cells are added to the region if they are on the region border and they are inliers of the current road surface. The road surface is recomputed (LSQ fitting to the current region) each time its border expands with 1-2 pixels.

The uncertainty of the stereo sensor is modeled and used to discriminate between road inliers and outliers (Fig. 20). Using a height uncertainty increasing with the depth is more reliable (compared to a constant uncertainty strip) and in accordance with the stereo system characteristics:

Obstacle / road separation is performed based on the detected road surface. Each DEM cell is labeled as drivable if it is closer to the road surface than its estimated height uncertainty or as non-drivable, otherwise. Small clusters of non-drivable cells are rejected based on criteria related to the density of 3D points. Remaining non-drivable clusters are classified into obstacles or traffic isles, based on the density of 3D points.

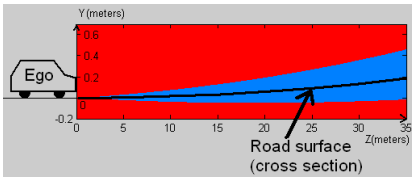


Fig. 20. Lateral view (a slice from 3D with a constant value for the X coordinate) of the inliers (blue) area around a quadratic road

This algorithm outputs the classified grid (Fig. 19.c) with three distinct cell types: road, traffic isles and obstacles. There are some difficult scenarios, with poor road texture or poor scene illumination, where only few 3D road points are reconstructed through stereovision. The road surface cannot be computed for such scenes, but the algorithm is still able to output an occupancy grid made of drivable and obstacle cells. In such scenes only the density of 3D points is used for discrimination, but the traffic isles cannot be identified.

The curb detection algorithm presented in [30] detects curbs as chains of segments on the DEM. Edge detection is applied to the DEM in order to highlight height variations. A method to reduce significantly the 3D noise from dense stereo was proposed, using a multi-frame persistence map. Temporal filtering was performed on edge points, based on the ego car motion, and only persistent points were validated. The Hough accumulator for lines was built with the persistent edge points. A scheme for extracting straight curbs (as curb segments) and curved curbs (as chains of curb segments) was proposed. Each curb segment was refined using a RANSAC approach to fit optimally the 3D data of the curb (Fig. 21).

Fig. 21. A curb is detected in front of the ego while turning left in an intersection

Both algorithms for road, traffic isles and obstacles detection [29] and for curb detection [30] use the same elevation map representation as input data, and therefore their results can be integrated into a single occupancy grid. Scenes with poor road texture will have a better delimited drivable area (Fig. 22).

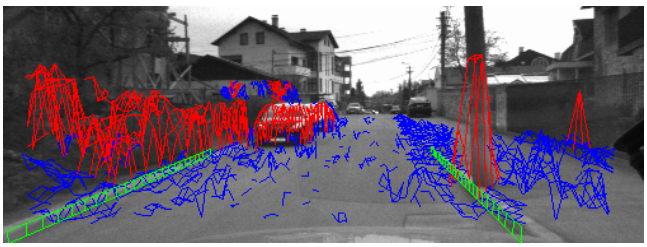


Fig. 22. The occupancy grid augmented with the detected curbs

The concept of multi-frame persistence map used for curb detection was also implemented for traffic isles, in order to remove false detections. Similar to false curbs, false traffic isles have an important feature: they persist only for a limited number of consecutive frames (mostly for two frames). Furthermore, traffic isles are static scene items. Based on these features, a fast and efficient approach can be used to filter false traffic isles: a multi-frame

persistence map is built (the ego motion between frames is taken into consideration) and only traffic isle cells that persist for several frames are validated.

Another improvement that can be done is the enhancement of the occupancy grid, to discriminate between static and dynamic obstacle cells. Again, the persistence map concept can be used. It shows the lifetime for each obstacle cell (for how many consecutive frames it was labeled as an obstacle cell, in a global reference system). Static obstacle cells will tend to have an increased lifetime while dynamic ones will have a shorter lifetime. However, depending on the obstacle size and speed, some of the obstacle cells might overlap cells from the same obstacle along several frames. This can offer false clues about the nature of individual cells. Therefore, the discrimination must be performed at blob level, as all the cells from the same obstacle are treated unitary. If the average persistence of the blob is above a threshold (a desired number of frames with no or negligible blob movement), the blob is considered static, otherwise dynamic.

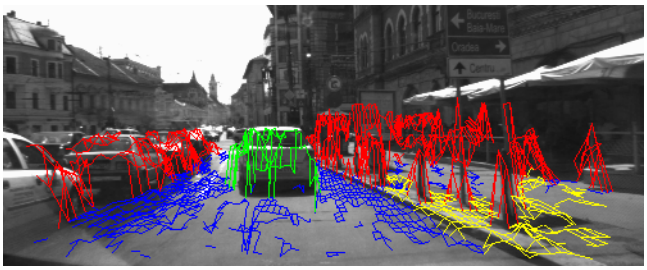


Fig. 23. The enhanced grid (with static / dynamic obstacles) projected onto the left image. The front car is the only moving obstacle; the vehicles on the left are waiting for the green light

4.2 Using the occupancy grid

The occupancy grid can be used for generating a compact, higher level representation of the environment components. A new representation was proposed and implemented [38], each scene component being represented as a 3D poly-line with additional information about its type (resulted from classification) and dynamic features.

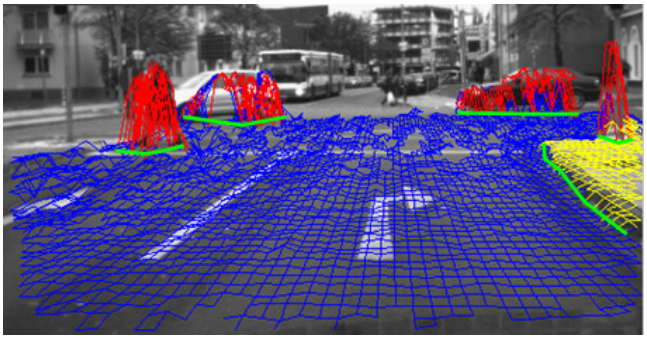


Fig. 24. Poly-line representation derived from the occupancy grid

The occupancy grid can be integrated over time using the ego car's motion parameters (yaw-rate and speed data), and the static grid cells. A global map is obtained (Fig. 265). The scene static features can be used for alignment with the intersection GPS or aerial maps (Fig. 26).

Since traffic isles (sidewalks) and/or curbs are included in the grid, these can be used as focus of attention for pedestrian detection. Another important issue in intersections is the detection of traffic lights and signs: static obstacle areas can be used for focusing attention for traffic lights / signs detection (Fig. 24).

Another way to use the occupancy grid is the detection (awareness) of the intersection situation (for passive intersections) based on: the configuration of static delimiters in the scene (ending traffic isles/curbs, enlarging drivable area, etc), and the configuration of dynamic obstacles (increased lateral motion).

Curbs can be used as measurement data for the lane detection algorithm, because the lane marking is usually not present near sidewalks, in urban scenes (Fig. 22).

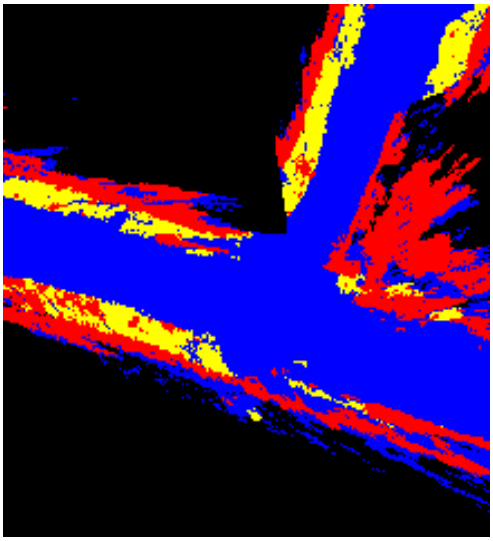


Fig. 25. The global occupancy grid

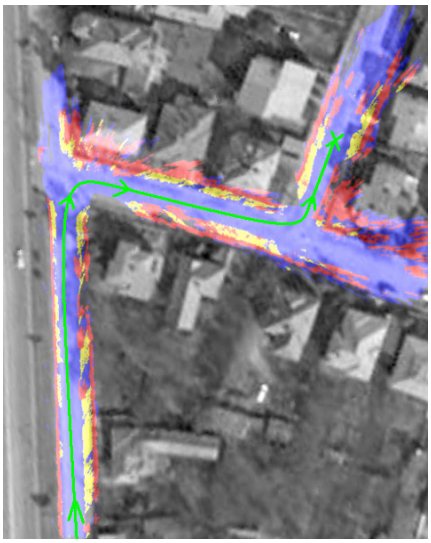


Fig. 26. Global occupancy grid superimposed to the satellite image (Google Earth) of the region. The trajectory of the ego car is shown with the green curve

5 Conclusions and future work

In this paper we have proposed an on-board stereovision sensor for the perception and description of urban and intersection environments. The sensor has as main features the exploitation of grayscale and stereo 3D information through specialized algorithms that provide structured and unstructured approaches for environment perception.

The system is able to detect the static and the dynamic components of the intersection environment. When the conditions allow it, the road geometry is estimated using a modular and configurable tracker based on particle filter, which allows the use of multiple lane models and the integration of multiple measurement cues. The obstacles are detected and tracked using a cuboid model, and subsequently classified to discriminate their type, and thus their importance in traffic. If no lane delimiters are available, or the geometry is unknown, the road surface and the free space ahead can be estimated in an unstructured fashion from the elevation map's occupancy grid. The obstacles are alternatively detected and described from the non-drivable cells of the occupancy grid. Thus, the system is able to detect, track and recognize most of the components of the traffic environment.

For the future, a fusion process between the structured and the unstructured descriptions is planned, in order to increase the detection robustness. Future work will include estimation of multiple types of road structures, detection and tracking of more object models, and enhancing the object classifier to handle more classes and improve the classification accuracy.

Cooperation between the stereo sensor and other types of data sources, such as the active sensors, or the GPS and map information is also planned for the future.

The stereovision process itself can be improved by using color information, and by experimenting with different camera and lens setups, to find the one that will provide the best coverage for the intersection.

Acknowledgement

This work was supported by Volkswagen AG Germany, Technical University of Cluj-Napoca Romania and recently is supported by the "INTERSAFE-2" EU founded project.

References

[1] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, M.M. Meinecke, T. Graf, T. B. To, M.A. Obojski, "A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision", *Proceedings of 2007 IEEE Intelligent Vehicles Symposium (IV2007)*, June 13-15, 2006, Istanbul, Turkey, pp 276-283.

[2] T. Dang, C. Hoffmann, "Stereo calibration in vehicles", *Proceedings of IEEE Intelligent Vehicles Symposium (IV2004)*, June 14-17, 2004, Parma, Italy, pp.268-272.

[3] T. Dang, C. Hoffmann, and C. Stiller, "Self-calibration for Active Automotive Stereo Vision", *Proceedings of IEEE Intelligent Vehicles Symposium (IV2006)*, June 13-15, 2006, Tokyo, Japan, pp.364-369.

[4] T. Marita, F. Oniga, S. Nedevschi, T. Graf, R. Schmidt, "Camera Calibration Method for Far Range Stereovision Sensors Used in Vehicles", *Proceedings of IEEE Intelligent Vehicles Symposium (IV2006)*, June 2006, Tokyo, Japan, pp. 356-363.

[5] S. Nedevschi, C. Vancea, T. Marita, T. Graf, "On-Line Calibration Method for Stereovision Systems Used in Far Range Detection Vehicle Applications", *IEEE Transactions on Intelligent Transportation Systems*, vol.8, no. 4, pp. 651-660, 2007.

[6] T. Marita, F. Oniga, S. Nedevschi, T. Graf, "Calibration Accuracy Assessment Methods for Stereovision Sensors Used in Vehicles", in *Proceedings of IEEE 3-rd International Conference on Intelligent Computer Communication and Processing (ICCP2007)*, 6-8 September 2007, Cluj-Napoca, Romania, pp. 111-118.

[7] C. Vancea, S. Nedevschi, "Analysis of different image rectification approaches for binocular stereovision systems", in *Proceedings of IEEE 2nd International Conference on Intelligent Computer Communication and Processing (ICCP 2006)*, vol. 1, September 1-2, 2006, Cluj-Napoca, Romania, pp. 135-142.

[8] E.D. Dickmanns, B.D. Mysliwetz, "Recursive 3-d road and relative ego-state recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no.2, pp. 199-213, 1992.

[9] R. Aufrere, R. Chapuis, F. Chausse, "A model-driven approach for real-time road recognition", *Machine Vision and Applications*, vol 13, no. 2, pp. 95-107, 2001.

[10] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, T. Graf, "High accuracy stereo vision system for far distance obstacle detection", in *Proceedings of IEEE Intelligent Vehicles Symposium (IV 2004)*, June 2004, Parma, Italy, pp. 292-297.

[11] R. Danescu, S. Nedevschi, M.M. Meinecke, T.B. To, "Lane Geometry Estimation in Urban Environments Using a Stereovision System", in *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC 2007)*, September 2007, Seattle, USA, pp. 271-276.

[12] R. Labayrade, J. Douret, D. Aubert, "A Multi-Model Lane Detector that Handles Road Singularities", in *Proceedings of IEEE Intelligent Transportation Systems Conference*, October 2006, Toronto, Canada, pp. 1143-1148.

[13] M. Isard, A. Blake, "CONDENSATION – conditional density propagation for visual tracking", *International Journal of Computer Vision*, vol. 29, nr. 1, pp. 5-28, 1998

[14] B. Southall, C.J. Taylor, "Stochastic road shape estimation", in *Proceedings of IEEE International Conference on Computer Vision*, 2001, Vancouver, Canada, pp. 205-212.

[15] K. Macek, B. Williams, S. Kolski, R. Siegwart, "A Lane Detection Vision Module for Driver Assistance", in *Proceedings of IEEE/APS Conference on Mechatronics and Robotics*, 2004

[16] U. Franke, H. Loose, C. Knoeppel, "Lane Recognition on Country Roads", in *Proceedings of IEEE Intelligent Vehicles Symposium*, June 2007, Istanbul, Turkey, pp.100-104.

[17] P. Smuda, R. Schweiger, H. Neumann, W. Ritter, "Multiple Cue Data Fusion with Particle Filters for Road Course Detection in Vision Systems", in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2006, Tokyo, Japan, pp. 400-405.

[18] R. Danescu, S. Nedevschi, M. M. Meinecke, T. B. To, "A Stereovision-Based Probabilistic Lane Tracker for Difficult Road Scenarios", in *Proceedings of IEEE Intelligent Vehicles Symposium 2008 (IV2008)*, Eindhoven, The Netherlands, June 4-6, 2008, pp.536-541.

[19] D. M. Gavrila, "Pedestrian detection from a moving vehicle," in *Proceedings of the European Conference on Computer Vision*, 2000,pp. 37–49.

[20] A. Khammari, F. Nashashibi, Y. Abramson and C. Laurgeau, "Vehicle detection combining gradient analysis and AdaBoost classification", in *Proceedings of Intelligent Transportation Systems Conference (ITSC 2005)*, pp. 61-71, 2005.

[21] D. Huber, A. Kapuria, R. Donamukkala and M. Herbert, "Parts-based 3D object classification", in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2004, vol 2, pp. 82-89.

[22] R. Osada, T. Funkhouser, B. Chazelle and D. Dobkin, "Matching 3D Models with Shape Distributions", Shape Modeling International, Genova, Italy, May 2001.

[23] D. M. Gavrila, "A Bayesian Exemplar-based Approach to Hierarchical Shape Matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28 No. 8, pp. 1408-1421, 2007.

[24] L. Havasi, Z. Szlávik and T. Szirányi, "Pedestrian Detection Using Derived Third-Order Symmetry of Legs", in *Proc. of the Int. Conference on Computer Vision and Graphics*, 2004, pp. 733-739.

[25] A. Shashua, Y. Gdalyahu and G. Hayun, "Pedestrian Detection for Driving Assistance Systems: Single-frame Classification and System Level Performance", in *Proceedings of IEEE Intelligent Vehicle Symposium(IV2004)*, June 2004, Parma, Italy, pp. 1-6.

[26] M. Bertozzi, E. Binelli, A. Broggi and M. D. Rose, "Stereo Vision-based approaches for Pedestrian Detection", in *Proceedings of the IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition - Workshops*, June 2005, San Diego, USA, vol. 3, pp. 16

[27] Z. Zhang, "A stereovision system for a planetary rover: Calibration, correlation, registration, and fusion," in *Machine Vision and Applications*, vol. 10, no. 1, pp. 27-34, 1997.

[28] M. Vergauwen, M. Pollefeys, and L. V. Gool, "A stereo-vision system for support of planetary surface exploration", *Machine Vision and Applications*, vol. 14, no.1, pp. 5–14, 2003.

[29] F. Oniga, S. Nedevschi, M-M. Meinecke, T-B. To, "Road Surface and Obstacle Detection Based on Elevation Maps from Dense Stereo", in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation* Systems (ITSC 2007), Sept. 30 - Oct. 3, 2007, Seattle, Washington, USA, pp. 859-865.

[30] F. Oniga, S. Nedevschi, M-M. Meinecke, "Curb Detection Based on a Multi-Frame Persistence Map for Urban Driving Scenarios", in *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems (ITSC 2008)*, Oct. 2008, Beijing, China, pp. 67-72.

[31] J.Y. Bouguet, Camera Calibration Toolbox for Matlab, www.vision.caltech.edu/bouguetj.

[32] C. Pocol, S. Nedevschi, M. M. Meinecke, "Obstacle Detection Based on Dense Stereovision for Urban ACC Systems", in *Proceedings of 5th International Workshop on Intelligent Transportation (WIT 2008)*, March 18-19, 2008, Hamburg, Germany, pp. 13-18.

[33] R. Danescu, S. Nedevschi, M.M. Meinecke, T. Graf, "Stereovision Based Vehicle Tracking in Urban Traffic Environments", in *Proceedings of the IEEE Intelligent*

Transportation Systems Conference (ITSC 2007), October 2007, Seattle, USA, pp.400-404.

[34] S. Bota, S. Nedevschi - Multi-Feature Walking Pedestrians Detection for Driving Assistance Systems, *IET Inteligent Transportation Systems Journal*, Volume 2, Issue 2, pp. 92-104, 2008.

[35] J.-Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker", available: http://mrl.nyu.edu/ bregler/classes/vision spring06/bouget00.pdf.

[36] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, June 2005, pp. 886–893.

[37] I. H. Witten, E. Frank "Data Mining: Practical machine learning tools and techniques, 2nd Edition", *Morgan Kaufmann*, San Francisco, 2007.

[38] S. Nedevschi, A. Vatavu, F. Oniga, M.-M. Meinecke, Forward Collision Detection using a Stereo Vision System, *Proceedings of IEEE 4-th International Conference on Intelligent Computer Communication and Processing (ICCP 2008)*, August 28-30, 2008, Cluj-Napoca, Romania, pp. 115-122.

Sergiu Nedevschi
Technical University of Cluj-Napoca, Str. C. Daicoviciu 15, 400 020 Cluj-Napoca, Romania
E-mail: Sergiu.Nedevschi@cs.utcluj.ro

Radu Danescu
Technical University of Cluj-Napoca, Str. C. Daicoviciu 15, 400 020 Cluj-Napoca, Romania
E-mail: Radu Danescu@cs.utcluj.ro

Tiberiu Marita
Technical University of Cluj-Napoca, Str. C. Daicoviciu 15, 400 020 Cluj-Napoca, Romania
E-mail: Tiberiu.Marita@cs.utcluj.ro

Florin Oniga
Technical University of Cluj-Napoca, Str. C. Daicoviciu 15, 400 020 Cluj-Napoca, Romania
E-mail: Florin.Oniga@cs.utcluj.ro

Ciprian Pocol
Technical University of Cluj-Napoca, Str. C. Daicoviciu 15, 400 020 Cluj-Napoca, Romania
E-mail: Ciprian.Pocol@cs.utcluj.ro

Silviu Bota
Technical University of Cluj-Napoca, Str. C. Daicoviciu 15, 400 020 Cluj-Napoca, Romania
E-mail: Silviu.Bota@cs.utcluj.ro

Marc Michael Meinecke
Volkswagen AG, Brieffach 01117770, 38436 Wolfsburg, Germany
E-mail: Marc-Michael.Meinecke@volkswagen.de

Marian Andrzej Obojski
Volkswagen AG, Brieffach 01117770, 38436 Wolfsburg, Germany
E-mail: Marian-Andrzej.Obojski@volkswagen.de

Keywords: stereovision, intersection assistance, lane detection, obstacle detection, tracking, classification, elevation maps, occupancy map