A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision

Sergiu Nedevschi, Radu Danescu, Tiberiu Marita, Florin Oniga,
Ciprian Pocol, Silviu Bota and Cristian Vancea
Technical University of Cluj-Napoca,
Romania

1. Introduction

Advanced driving assistance systems (ADAS) form a complex multidisciplinary research field, aimed at improving traffic efficiency and safety. A realistic analysis of the requirements and of the possibilities of the traffic environment leads to the establishment of several goals for traffic assistance, to be implemented in the near future (ADASE, INVENT, PREVENT, INTERSAFE) including: highway, rural and urban assistance, intersection management, pre-crash.

While there are approaches to driving safety and efficiency that focus on the conditions exterior to the vehicle (intelligent infrastructure), it is reasonable to assume that we should expect the best results from the in-vehicle systems. Traditionally, vehicle safety is mainly defined by passive safety measures. Passive safety is achieved by a highly sophisticated design and construction of the vehicle body. The occupant cell has become a more rigid structure in order to mitigate deformations. The frontal part of vehicles has been improved as well, e.g. it incorporates specially designed "soft" areas to reduce the impact in case of a collision with a pedestrian. In the recent decades a lot of improvements have been done in this field.

Similarly to the passive safety systems, primitive active safety systems, such as airbags, are only useful when the crash is actually happening, without much assessment of the situation, and sometimes they are acting against the well-being of the vehicle occupants. It has become clear that the future of the safety systems is in the realm of the artificial intelligence, systems that sense, decide and act.

Sensing implies a continuous, fast and reliable estimation of the surroundings. The decision component takes into account the sensorial information and assesses the situation. For instance, a pre-crash application must decide whether the situation is of no danger, whether the crash is possible or when the crash is imminent, because depending on the situation different actions are required: warning, emergency braking or deployment of irreversible measures (internal airbags for passenger protection, or inflatable hood for pedestrian protection). While warning may be annoying, and applying the brakes potentially dangerous, deploying non-reversible safety causes permanent damage to the vehicle, and therefore the decision is not to be taken lightly. However, in a pre-crash scenario it is even more damaging if the protection systems fail to act. Therefore, it is paramount that the

Source: Stereo Vision, Book edited by: Dr. Asim Bhatti,
ISBN 978-953-7619-22-0, pp. 372, November 2008, I-Tech, Vienna, Austria

protection systems act when needed, and only when needed, a decision that cannot be taken in the absence of reliable sensor data.

The sensorial systems for driving assistance (highway and urban) are today the focus of large, joint research projects, which combine active and passive sensors, GPS navigation, and telematics. Projects such as CARSENSE (www.carsense.org) INVENT (www.invent-online.de), PREVENT (www.prevent-ip.org), bring together car manufacturers and research partners for the common goal of solving the driving assistance problem.

In order to provide support for these applications, a sensorial system must provide an accurate and continuously updated model of the environment, fitted for high level reasoning. The environment description should include:

- Lane detection / Lane parameters estimation
- Navigable channel detection and channel parameters estimation in crowded environments
- Vehicle detection and tracking
- Detection of fixed (non-moving) obstacles
- Pedestrian detection and tracking.

There are many types of sensors that can be used for advanced driving assistance systems. The most known are:

- Long range radar: with a range of 1 to 200 m, and a response time of around 40 ms, it is a highly accurate ranging sensor, with a narrow field of view, suitable for detection of radar-reflecting targets such as vehicles in highway environments.
- Short/mid range radar: having a working range of 0-80 m, a fast response time, high accuracy and a medium width field of view, it is suitable for near range detection of vehicles in crowded urban scenarios. Both near range and far range radars have an increased reliability when detecting moving objects.
- Laser scanner: a high precision ranging sensor, working in near or far distance ranges, it is not limited to the metallic surfaces like the radar, but has considerable difficulty with low albedo objects.
- Monocular video sensors: employed in the visual or in the infrared light spectrum, the visual sensors can have a high field of view and can extract almost any kind of information relevant for driving assistance. The main problem of these sensors is that it cannot rely on accurate 3D information, having to infer it indirectly, usually with poor results.

A stereovision sensor adds the 3D information to the visual, thus becoming the most complex and complete sensor for driving assistance. It is capable of detecting any type of obstacle that falls inside its adjustable field of view, the road and lane geometry, the free space ahead, and it is also capable of visual classification, for pedestrian recognition.

The stereovision-based approaches have the advantage of directly estimating the 3D coordinates of an image feature, this feature being anything from a point to a complex structure. Stereovision involves finding correspondents from the left to the right image, and the search for correspondence is a difficult, time demanding task, which is not free from the possibility of errors. Obstacle detection techniques involving stereovision use different approaches in order to make some simplifications of the classic problem and achieve real-time capabilities. For instance, [1] uses stereovision only to measure the distance of an object after it has been detected from monocular images, [2] detects the obstacle points from their stereo disparity compared to the expected disparity of a road point, [3] detects obstacle features by performing two correlation processes, one under the assumption that the feature is part of a vertical surface and another under the assumption that it is part of a horizontal surface, and comparing the quality of the matching in each of the cases. A stereovision system that uses no correspondence search at all, but warps images instead and then performs subtraction is presented in [4].

Processing 3D data from stereo (dense or sparse) is a challenging task. A robust approach can prove of great value for a variety of applications in urban driving assistance. There are two main algorithm classes, depending on the space where processing is performed: disparity space-based and 3D space-based. Most of the existing algorithms try to compute the road/lane surface, and then use it to discriminate between road and obstacle points.

Disparity space-based algorithms are more popular because they work directly with the result of stereo reconstruction: the disparity map. The "v-disparity" [5] approach is well known and used to detect the road surface and the obstacles in a variety of applications. It has some drawbacks: it is not a natural way to represent 3D (Euclidian) data, it assumes the road is dominant along the image rows, and it can be sensitive to roll angle changes.

The 3D space algorithms have also become popular among the researchers in recent years. Obstacle detection and 3D lane estimation algorithms using stereo information in 3D space are presented in [6], [7], [8] and [9], ego pose estimation algorithms are presented in [10] and [11], and unstructured environment estimation algorithms are presented in [12], [13] and [14].

2. Stereo sensing solution for driving urban assistance

The research team of the Technical University of Cluj-Napoca has already implemented a stereovision-based sensor for the highway environment [6], [7] and [8]. This sensor was able to detect the road geometry and the obstacle position, size and speed from a pair of synchronized grayscale image pairs, using edge-based, general geometry, software stereo reconstruction.

The urban scenario required important changes in the detection algorithms, which in turn required more stereo information. Thus, the edge-based stereo engine was discarded, and replaced with a dense stereo system. A software dense stereo system being time consuming, a hybrid solution was chosen: software rectification and down sampling, followed by hardware correspondence search. The time gained by the use of a hardware board compensated the increase in complexity of the new algorithms.

The dense stereo information is vital for the new obstacle reconstruction module, which extracts oriented objects even in serious clutter, and also allows better shape segmentation for recognition of pedestrians. Dense stereo information allows us to compute and track an unstructured elevation map, which provides drivable areas in the case when no lane markings or other road delimiting features are present or visible.

Lane detection requires edges, but the vertical profile is better computed from dense stereo information. The edge based lane detection algorithms are completely changed, adapted to the limited and variable viewing distance of the urban environment. A freeform lane detection module was added, in order to solve the problem of the non-standard geometry roads.

The dense stereovision based sensor presented in this paper provides complex and accurate functionality on a conventional PC architecture, covering many of the problems presented by the urban traffic environment, and promising to be a valuable addition to a driving assistance system.

The hardware acquisition system (fig. 1) includes two grayscale digital cameras with 2/3" (1380x1030) CCD sensors and 6.5 mm fixed focal length lenses, allowing a horizontal field of view (HFOV) of 72 [deg]. The cameras are mounted on a rigid rig with a baseline of 320 [mm] (fig. 2). The images are acquired at full resolution with a digital acquisition board at a maximum frame rate of 24 fps.

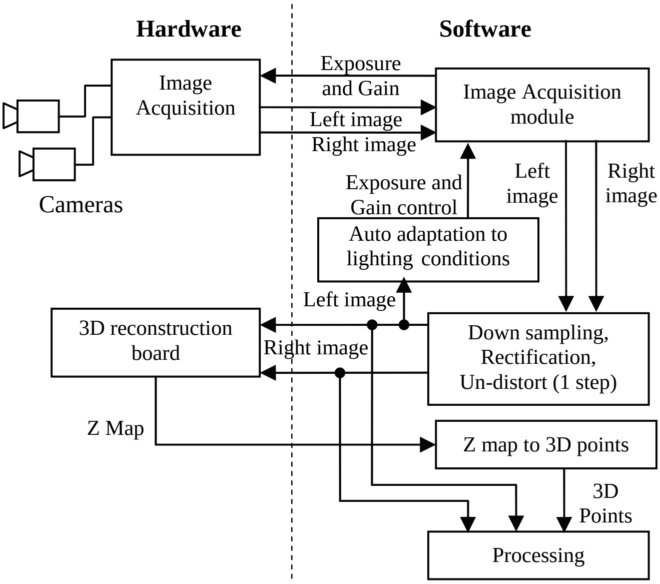


Fig. 1. The stereovision system architecture.

The camera parameters are calibrated using a dedicated method optimized for high accuracy stereovision [15],[16] and [17] using the full resolution images.

The images are further enhanced by lens distortion correction and rectified in order to fulfill the dense stereo reconstruction requirements (canonical images). A down-sampling step is used to adapt the image size to the dedicated hardware reconstruction board (512 pixels width) and to minimize the noise introduced by the digital rectification and image correction. The whole process is reduced to an image warping approach performed in a single step (fig. 1) using reverse mapping and bilinear interpolation [18]. An optimized implementation using MMX instructions and lookup tables was used in order to minimize the processing time.

The 3D reconstruction of the scene is performed using a dedicated hardware board. The input of the board consists in two rectified images and the output can be either a disparity or a Z map (left camera coordinate system). Our system uses 3D points set for scene representation; therefore the preferred output is the Z map. Using the Z coordinate value, the X and Y coordinate can be computed and then transformed into the car coordinate system.

With the current system setup a detection range optimally suited for the urban environments is obtained (fig. 2):

- minimum distance: 0.5 m in front of the ego car;

- delimiters of the current lane are visible at 1.0 m;
- reliable detection range: 0.5 … 35 m;

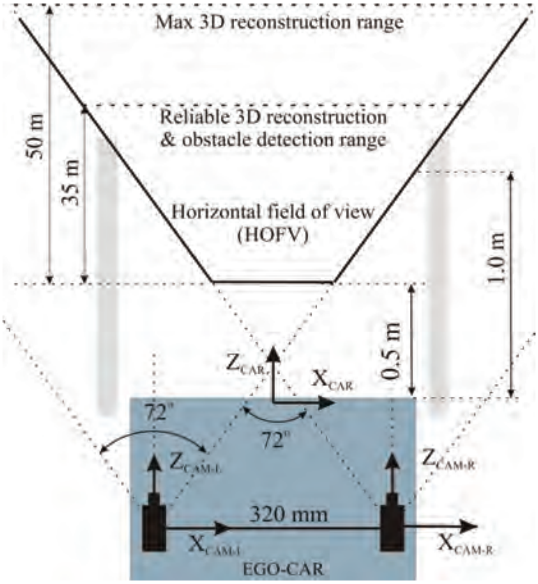


Fig. 2. Detection range of the current stereo system setup.

3. Stereovision-based lane detection

The urban lane detection system is organized as an integrator of multiple sensors, using a Kalman filter framework. Instead of having multiple physical sensors, we have multiple detection stages, which all deliver results that will be used to update the lane model state parameters. The cycle begins with the prediction, and continues with all the detection algorithms, until the final update. When one algorithm updates the lane state, the resulted estimation becomes the prediction for the next stage. In this way, we can insert any number of algorithms into the processing chain, or we can temporary disable some of them, for testing or speedup purposes.

Figure 3 shows the organization of the lane detection system, the main processing modules and the relationships between them. In what follows, we'll give a brief description for each module. A detailed description of the lane detection system is given in [19].

One of the most important advantages of stereovision-based lane detection is the possibility of direct detection of the vertical profile (pitch angle and vertical curvature). The projection of the 3D point set in the (YOZ) plane (height and distance) is analyzed by means of histograms. The pitch and the curvature range of values is divided into discrete hypotheses. First, a polar histogram counting the near range points along the lines described by each pitch angle hypothesis is built, and the lowest line having a significant number of points is selected. The vertical curvature histogram is built by taking into consideration the already detected pitch angle, and the curvature hypotheses, and counts the points in the far range. The method is somewhat similar to the Hough transform, and is described in detail in [7].

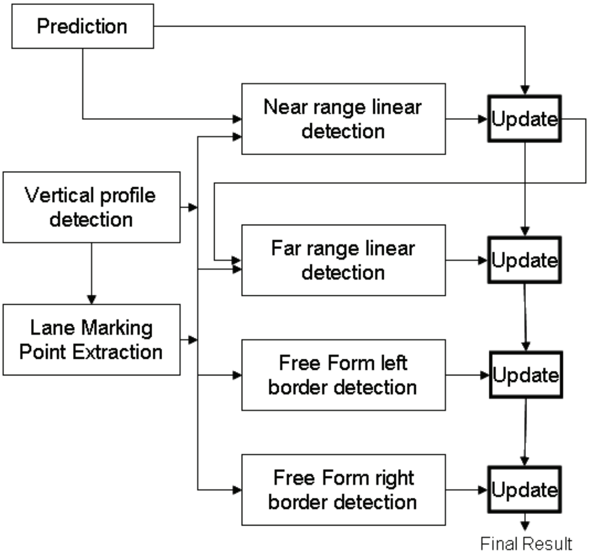

Prediction
Near range linear detection
Vertical profile detection
Far range linear detection
Lane Marking Point Extraction
Free Form left border detection
Free Form right border detection
Update
Update
Update
Update
Final Result

Fig. 3. Lane detection system architecture

After the vertical profile is detected, the 3D points can be labeled as belonging to the road or belonging to obstacle structures. The points belonging to the road are the main features used for lane geometry estimation. The next step is to extract, from the road surface point set, the points that have a higher relevance for lane delimitation, namely the lane markings. The highway lane detection approach required little information about lane markings, because it could rely greatly on the lane model. For the urban environment, however, we require a fast and robust lane marking extraction algorithm.

The lane marking extraction method relies on the well-known dark-light-dark transition detection [20]. We have to search for pairs of gradients of opposing sign and equal magnitude. We have improved the method by using a variable filter for computing the horizontal gradient. The size of the filter is the size of a standard width lane marking projected in the image space, and varies because of the perspective effect. The following equation shows the differentiation filter that is used:

$$\begin{aligned} G_N(x,y) &= \frac{\sum_{i=x+1}^{x+D} I(i,y) - \sum_{i=x-1}^{x-D} I(i,y)}{2D} \\ D &= KernelSize(y) \end{aligned} \qquad (1)$$

Applying the variable width filter we preserve the level of detail in the distance while filtering the noise in the near areas. The gradient maxima and minima are paired and the DLD pairs are extracted as lane markings. The complete technique is described in [21].

Although the clothoid model is not always accurate for the urban scenario, it has several benefits, such as good results when the lane is delimited by simple edges (unmarked roads). Due to the short visibility range, we have decided to avoid matching the whole clothoid model on the image data, but to match pairs of line segments instead, in two zones: near and far.

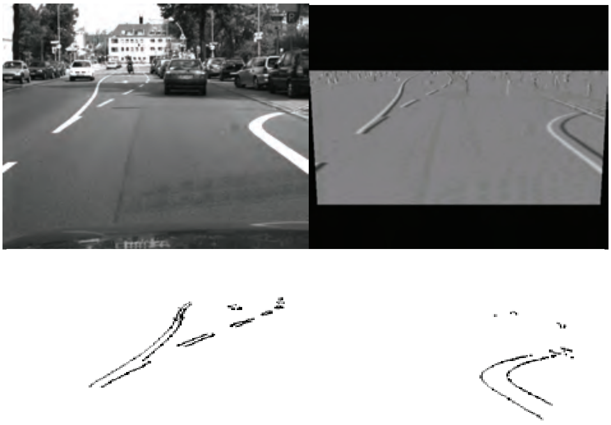


Fig. 4. Lane marking detection – top left, original image; top right, results of the adaptive gradient filter; bottom, lane marking results

First, we make an attempt for the near zone (2m to 5 m). Hough transform is performed on the image edges corresponding to the road points, and line segments are extracted. Lane markings will have a higher weight in the Hough bins, and therefore they will have a higher priority. We divide then the line segments in two sets – left and right. The segments on the left are paired with the segments on the right, and the best pair is selected as our lane measurement. The linear measurement will update the clothoidal model parameters using the Extended Kalman Filter. Depending on whether the detection has been successful on both sides, or only on one side, the measurement vector for the Kalman filter will have different configurations:

$$\mathrm{Y}_{both} = \begin{bmatrix} xbottomleft \\ xtopleft \\ xbottomright \\ xtopright \end{bmatrix} \quad \mathrm{Y}_{left} = \begin{bmatrix} xbottomleft \\ xtopleft \end{bmatrix} \quad \mathrm{Y}_{right} = \begin{bmatrix} xbottomright \\ xtopright \end{bmatrix} \tag{2}$$

If the linear fit for the near zone is successful, the same is done for the far zone, and the model parameters are updated again.

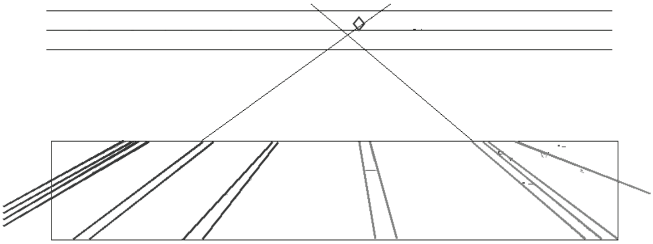


Fig. 5. The left and the right lines must intersect at the horizon line

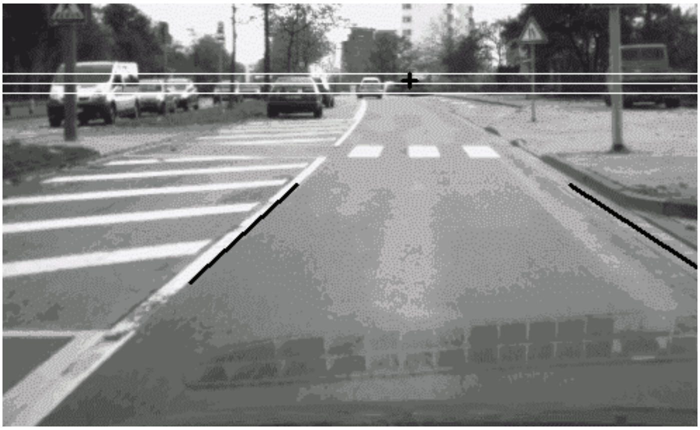


Fig. 6. Linear lane detection result

Sometimes the clothoid lane model is not suited for the road we wish to observe, and the detection will be incorrect. For these cases, a freeform lane detection system has been implemented. Because we don't have strong models to guide our search, we have to discard the non-marking delimiters, and work with lane markings only. The markings are projected onto a top view image, and then distance transform is performed, to facilitate the fitting of a lane border. The left and the right lane borders are represented as Catmull-Rom splines with four control points. The lateral coordinates of the four control points are the model parameters, and they are found using a simulated annealing search strategy in the model space.

The result of the freeform detection module is a chain of 3D X coordinates for a set of equally spaced fixed Z coordinates. The X values will form the measurement vector for the lane state update using again the Kalman filter.

$$\mathrm{Y}_{side} = \left[X_1, X_2, ..., X_n\right]^T \tag{3}$$

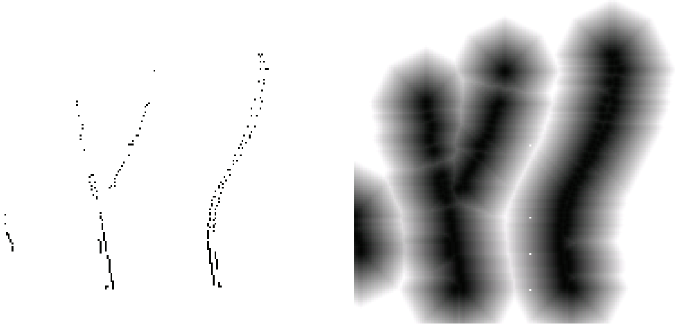


Fig. 7. Top view of the lane markings and the distance transform image used for freeform lane matching

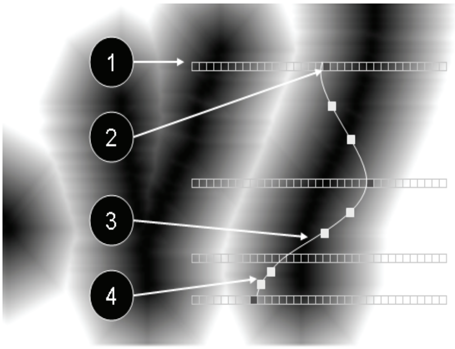


Fig. 8. The search problem for freeform detection: 1 - The search space for one of the control points. The y coordinate is fixed; 2 - A control point instance (hypothesis); 3- The Catmull-Rom spline generated by the control points hypotheses in the current iteration; 4 - The intermediate points, which, together with the control points, are used for evaluating the distance between the curve and the image

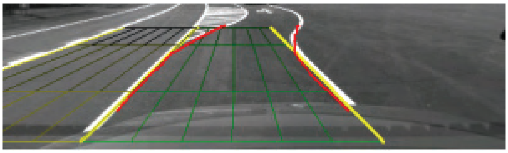


Fig. 9. Freeform lane detection succeeds in situations where the clothoid model fails

4. Stereovision-based obstacle detection and tracking

4.1 Stereovision-based obstacle detection

The obstacle detection algorithm is based on the dense stereo information, which provides a rich amount of 3D data for the observed scene and allows the possibility to carry out geometrical reasoning for generic obstacle detection regardless of the 2D appearance in images.

Starting from the obstacle points identified by the vertical profile provided by the lane detection subsystem, the target is to detect the obstacles as 3D boxes, having position, orientation and size. 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 criteria regarding the shape, the occupied areas may get fragmented into parts that obey the cuboidal shape. The orientation of the obstacles (on the road surface) is then extracted.

The only 3D points used by the obstacle detection algorithms are those situated above the road and below the height of the ego car (fig. 10.b). It is supposed that the obstacles do not overlap each other on the vertical direction. In other words, on a top view (fig. 10.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 our images, providing fewer pixels, and therefore, less, 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 of constant density is proposed (fig. 10.c). The horizontal field of view of the camera is divided into polar slices of 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. 11.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:

$$Column = ImageColumn/c \tag{4}$$

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. 10.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:

$$[Z_0 \ldots Z_0 + IntervalLength(Z_0)] = [Z_0 \ldots Z_1] \tag{5}$$

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

$$IntervalLength(Z) = k*Z \tag{6}$$

k being empirically chosen). Thus

$$Z_0 + IntervalLength(Z_0) = Z_0 + k*Z_0 = Z_0*(1+k) = Z_1 \tag{7}$$

b. The Cartesian interval corresponding to the nth row of the compressed spaces is $[Z_n \ldots Z_n + IntervalLength(Z_n)]$,
where

$$Z_n = Z_0*(1+k)^n \tag{8}$$

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 $[Z_i \ldots Z_i + IntervalLength(Z_i)] = [Z_i \ldots Z_{i+1}]$.
From equation (8), we can find the row number as the index of the point's interval

$$Row = i = [\log_{1+k} \frac{Z}{Z_0}] \tag{9}$$

Each 3D point, having the *(X, Z)* coordinates in the top view of the Cartesian space is transformed into a cell C(*Row, Column*) in the compressed space.

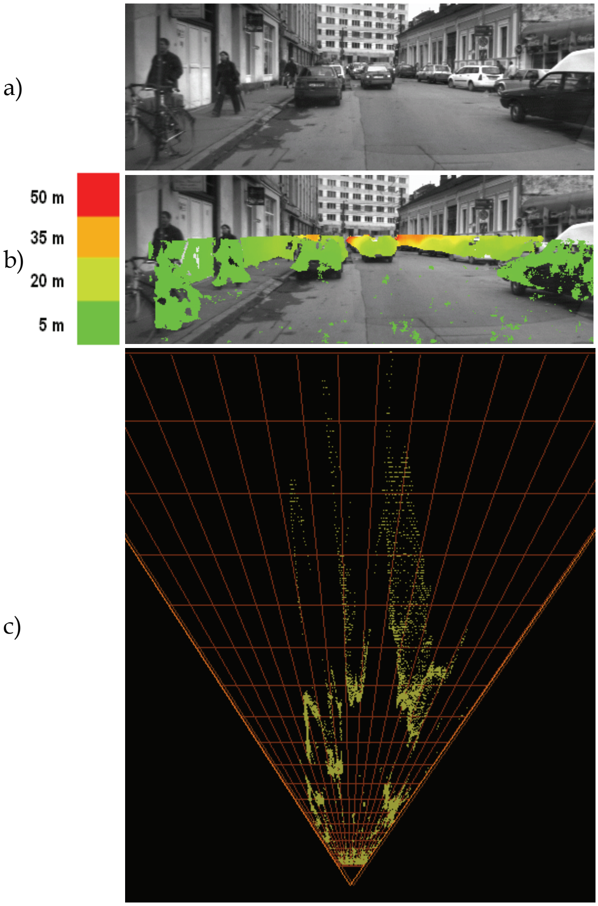


Fig. 10. Division into tiles. a) Gray scale image, b) 3D points – perspective view, c) 3D points – top view; the tiles are here considerably larger for visibility purpose; wrongly reconstructed points can be seen in random places; reconstruction error is visible as well.

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. 11.b). The small occupied areas are discarded in this phase.

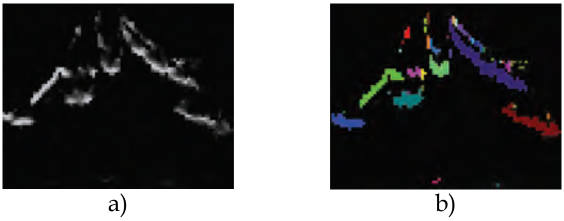


Fig. 11. The compressed space (for scene in fig. 7) – a bi-dimensional histogram counting 3D points. Occupied areas are identified by cell labeling

The occupied areas may contain several obstacles and by consequence they may have multiple 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 applied. The two sub-parts can be divided again and again as long as concavities are found.

In fig. 12.c the bottom side of the envelope for the cells in fig. 12.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.

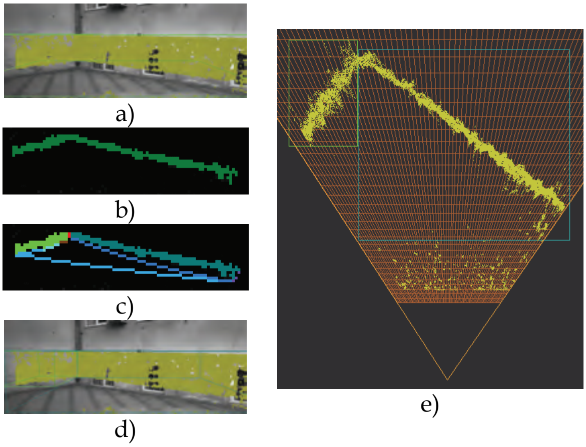


Fig. 12. Fragmentation of occupied areas into primitive obstacles. a) an occupied area, b) the labeling in the compresses space, c) sides of the envelope and the two primitive obstacles of the occupied area – compressed space, d) the two primitive obstacles – perspective view, e) the two primitive obstacles – 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. 12.d (perspective view) and fig. 12.e (top view). A more detailed description of the obstacle detection system is given in [22].

4.2 Stereovision-based obstacle tracking

Tracking is the final stage of the stereovision based object recognition system. The cuboids extracted using the methods described in the previous paragraphs represent the measurement data. The state of the object, composed of position, size, speed and orientation, is tracked using a Kalman filter based framework.

A measurement cuboid may initialize a track if several conditions are met: the cuboid is not associated to an existing track, the cuboid is on the road (we compare its Y position with the profile of the road), the cuboid's back side position in the image does not touch the image limits, and the height and width of the cuboid must be consistent to the standard size of the vehicles we expect to find on the road. The classification based on size is also useful for initialization of the object's length, as in most cases the camera cannot observe this parameter directly.

The core of the tracking algorithm is the measurement-track association process. The association (matching) process has two phases: a 3D matching of the predicted track cuboid against the measurements, which is performed as a simple intersection of rectangles in the top view space, and a corner by corner matching in the image space, when each active corner is matched against the corners of the measurement cuboids that passed the 3D intersection test.

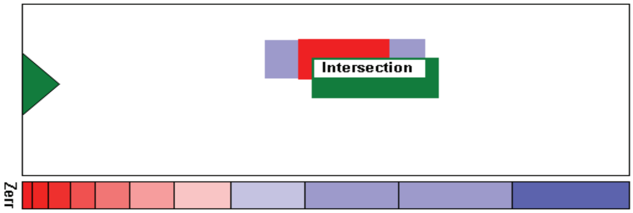


Fig. 13. Coarse association between prediction and measurement cuboids

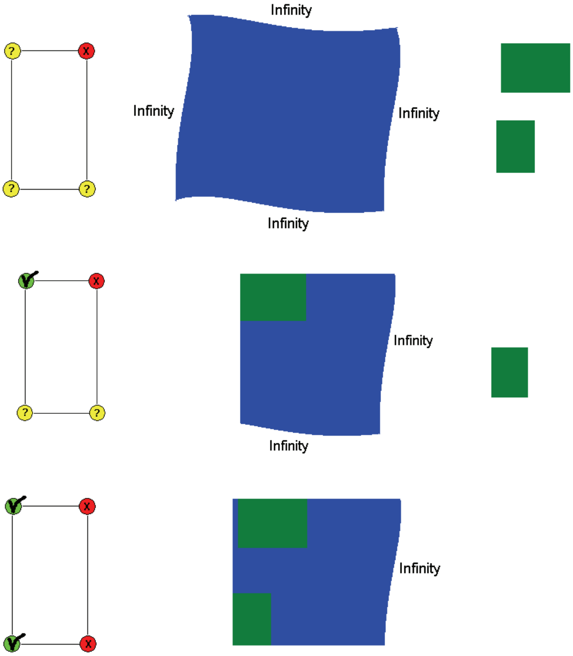


Fig. 14. Fine association at corner level and the building of a meta-measurement

The ratio between the area of the intersection and the area of the enhanced prediction is a measure of the quality of the 3D association. If a measurement intersects multiple predicted objects, it will be associated to the one it had the best intersection measure. A predicted object may associate to multiple measurement objects, but not the other way around.

After the 3D association is completed, each track prediction is compared to each of the associated measurements corner by corner, using their 2D projection, for the visible corners only. If a relevant corner of the prediction associates to at least one relevant corner of a measurement, this corner becomes an "active" corner. The active corners form a virtual object which we'll call "meta-measurement". The meta measurement is the sum of all measurement objects associated to a predicted object (Fig. 14). The meta measurement is used in the Kalman filter for updating the track state.

5. Description of unstructured environments

The 3D lane cannot be detected in urban scenes without lane delimiters (ex. intersections). An alternative method can 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.

The dense stereo engine normally reconstructs most of the road points even if lane markings are not present. Thus, the surface of the road can be computed by fitting a geometric model to the 3D data. Fitting the model to 3D data should be performed in a least-square fashion (LSQ), or, more robustly, using a statistical approach (ex. RANSAC). The model used for the road is quadratic (the vertical coordinate Y is a 2nd degree function of the depth Z and lateral displacement X).

$$Y = -a \cdot X - a' \cdot X^2 - b \cdot Z - b' \cdot Z^2 - c\,. \tag{10}$$

Fitting the quadratic surface to a set of *n* 3D points involves minimizing an error function. The error function S represents the sum of squared errors along the height:

$$S = \sum_{i=1}^{n} \left(Y_i - \overline{Y_i} \right)^2, \tag{11}$$

Where Y_i is the elevation of the 3D point *i* and $\overline{Y_i}$ is the elevation of the surface at coordinates (X_i, Z_i). Minimizing only along the Y-axis, instead of the surface normal, is acceptable. Even for curved roads, the normal of the surface is close to the Y-axis: for an extreme local slope of 20% (11.3 degrees), the residual of a 3D point along the vertical represents 98% of the residual along the normal. The computational complexity is highly reduced by avoiding minimization against the normal of the surface.

By replacing (10) into (11), the function *S* is obtained, where the unknowns are *a, a', b, b'*, and *c*:

$$S = \sum_{i=1}^{n} \left(Y_i + a \cdot X_i + a' \cdot X_i^2 + b \cdot Z_i + b' \cdot Z_i^2 + c \right)^2 . \tag{12}$$

For *S* to have a minimum value, its partial derivatives with respect to the unknowns must be 0. The following system of equations must be solved:

$$\left\{ \frac{\partial S}{\partial a} = 0,\ \frac{\partial S}{\partial a'} = 0,\ \frac{\partial S}{\partial b} = 0,\ \frac{\partial S}{\partial b'} = 0,\ \frac{\partial S}{\partial c} = 0\,. \right. \tag{13}$$

After writing explicitly each equation, the system (4) becomes (matrix form):

$$\begin{bmatrix} S_{X^2} & S_{X^3} & S_{XZ} & S_{XZ^2} & S_X \\ S_{X^3} & S_{X^4} & S_{X^2Z} & S_{X^2Z^2} & S_{X^2} \\ S_{XZ} & S_{X^2Z} & S_{Z^2} & S_{Z^3} & S_Z \\ S_{XZ^2} & S_{X^2Z^2} & S_{Z^3} & S_{Z^4} & S_{Z^2} \\ S_X & S_{X^2} & S_Z & S_{Z^2} & n \end{bmatrix} \begin{bmatrix} a \\ a' \\ b \\ b' \\ c \end{bmatrix} = \begin{bmatrix} -S_{XY} \\ -S_{X^2Y} \\ -S_{ZY} \\ -S_{Z^2Y} \\ -S_Y \end{bmatrix}, \tag{14}$$

Generically, each sum is computed as

$$S_\alpha = \sum_{i=1}^{n} \alpha_i \text{ (for example } S_{XZ} = \sum_{i=1}^{n} X_i \cdot Z_i \text{).} \tag{15}$$

If weights (w) for each point are available, then the following formulas will be applied:

$$S_\alpha = \sum_{i=1}^{n} w_i \cdot \alpha_i \text{ (example } S_{XZ} = \sum_{i=1}^{n} w_i \cdot X_i \cdot Z_i \text{).} \tag{16}$$

System (14) has 5 linear equation and 5 unknowns, therefore solving it is a trivial algebra problem. This explicit way of minimization was preferred instead of the pseudo-inverse matrix method. It allows real time re-computation (hundreds of times per frame) of the road surface during the surface growing step, as it will be explained later in section. This model allows the detection of road surfaces with non-zero pitch and roll angles relative to the ego car, and with vertical curvatures (lateral/longitudinal). The model can be extended to fit complex road surfaces, such as cubic or B-spline surfaces.

The 3D space available consists of a set of 3D points (80,000 to 120,000). Real-time fitting of the road surface to this set is not possible because it has a high computational complexity. A (bird-eye rectangular, 13x40 meters) region of interest of the 3D space can be represented similar to a digital elevation map (DEM). A DEM is formed: the value of each cell is proportional to the 3D height of the highest point (within the cell). The DEM presents poor connectivity between road points at far depths (due to the perspective effect, Fig. 15.b). Connectivity can be improved by propagating (to empty cells) the height of valid cell (with 3D points), along the depth (Fig. 15.c). The propagation amount is computed from the stereo geometry of the system, to compensate the perspective effect.

For the classification into drivable/non-drivable (road inliers/outliers, Fig. 16) areas, the depth uncertainty model was extended to a height uncertainty model (17). The expected height uncertainty *Yerr* is a function of the height *Y* and the depth Z of the 3D point, height of the cameras *Hcam*, and the estimated depth uncertainty *Zerr*. *Zerr* is a function of the stereo system parameters and the expected disparity uncertainty. The disparity uncertainty was chosen experimentally as 1 pixel, although a more complex model for estimating the correlation's accuracy can be developed.

$$Yerr = \left| \frac{(Y - Hcam) * Zerr}{Z} \right|. \tag{17}$$

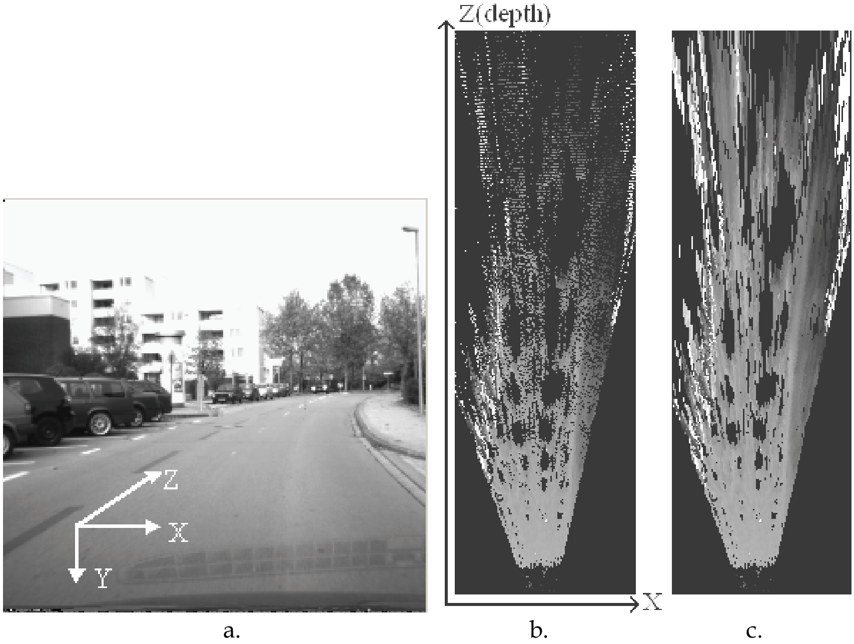


Fig. 15. The urban scenario (a) and the DEM (b. initial, c. with propagation of heights). Darker means more elevated.

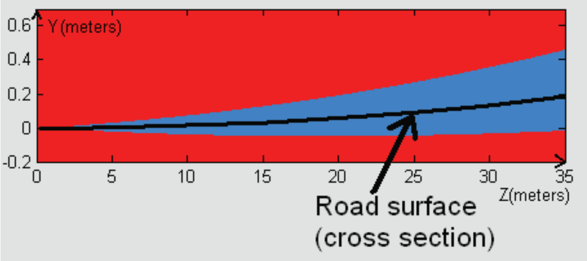
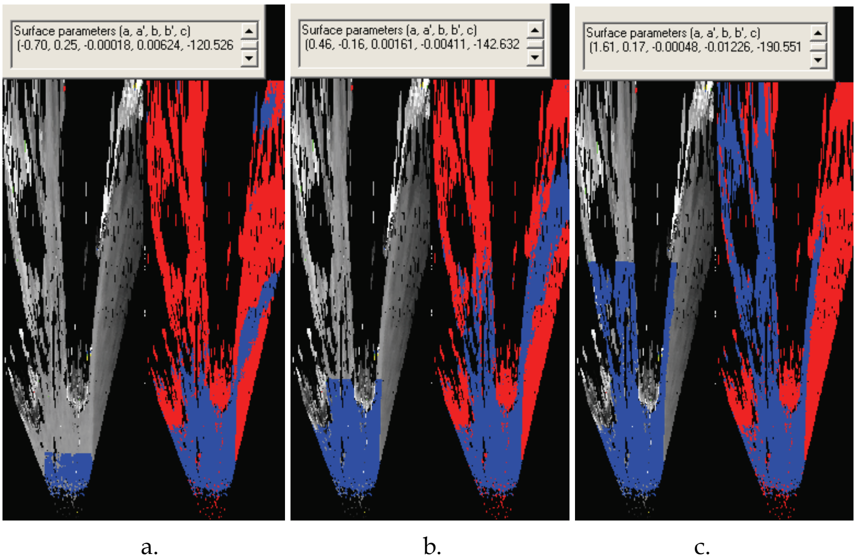


Fig. 16. Lateral view of the inliers (blue) region around a quadratic road surface, according to the proposed uncertainty model.

LSQ fitting is sensitive to rough noise in the data set. Due to road outliers, such as obstacle points, LSQ fitting is likely to fail when applied to the whole data set. We propose an optimal two-step approach for real-time detection of the road surface:

1. 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.
2. The primary solution is refined through a region growing process (Fig. 17) 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 the border of the region expands with 1-2 pixels (about 100 new cells).



Surface parameters (a, a', b, b', c)
(-0.70, 0.25, -0.00018, 0.00624, -120.526

Surface parameters (a, a', b, b', c)
(0.46, -0.16, 0.00161, -0.00411, -142.632

Surface parameters (a, a', b, b', c)
(1.61, 0.17, -0.00048, -0.01226, -190.551

Fig. 17. Road inliers (blue) / outliers (red) detected with the primary surface in a. Intermediate regions, growing from left to right (b and c). For each image, the left side shows the inliers used for surface fitting, while the right side shows the classification of the whole DEM based on the current surface parameters.

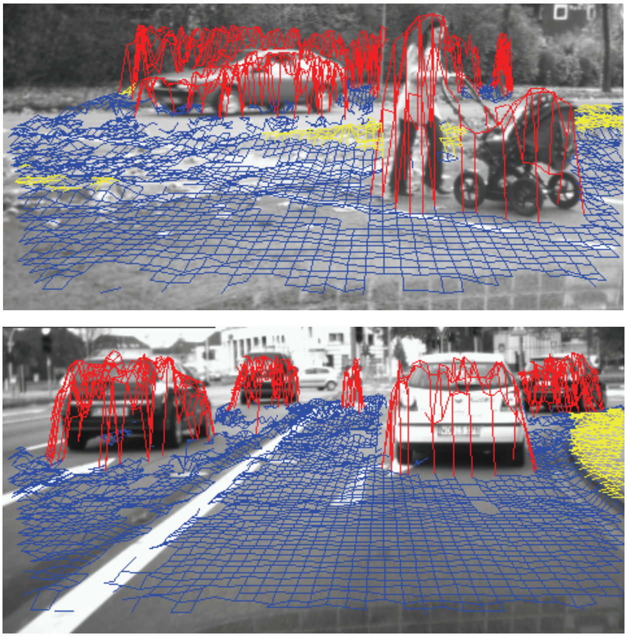


Fig. 18. The result for two scenes re-projected back as a grid onto the left image: obstacles (red) and traffic isles (yellow) are separated from the drivable road surface (blue).

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 and traffic isles, based on the density of 3D points. This approach is detailed in [13]. Some results are presented in figure 18, re-projected as a grid onto the left image.

6. Object classification

Out of the set of 3D cuboids depicting the obstacles in the urban traffic, one category of objects is of special importance: the pedestrians. The pedestrians are the most vulnerable traffic participants, and also the most undisciplined and having the most unpredictable behavior. For these reasons, the pedestrians need to be recognized as early and as reliably as possible.

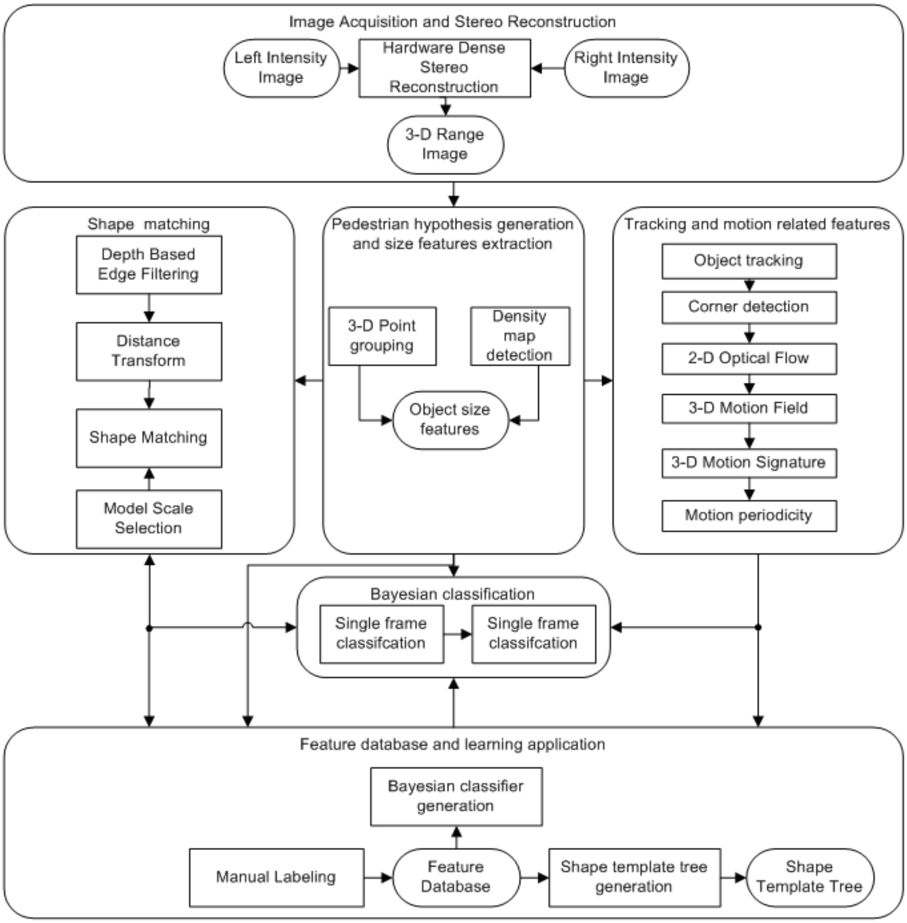


Fig. 19. Classification dataflow

In complex, urban scenarios, pedestrians are difficult to detect. This is because the variability of their appearance and the high clutter present in this type of scenarios. Therefore, as much information as possible must be used in order to correctly recognize them. A multi-feature dichotomizer was implemented [23], which can separate obstacles into pedestrians and non-pedestrians. The implementation contains the following modules (see Fig. 19):

- The *image acquisition and stereo reconstruction module*, described in section 2, supplies a dense range image, using the image pair from the stereo camera system.
- The *hypothesis generation and size features extraction* module generates pedestrian hypotheses, which will be classified into pedestrians and non-pedestrians. For short range (up to 10 meters) a specialized algorithm is used to generate reliable pedestrian hypotheses, containing only individual pedestrians. The algorithm is based on a "density map" built by accumulating 3D points, projected on the horizontal xOz plane [23]. As this algorithm does not give good results for distant objects, we use the generic obstacle detected by the algorithm described in section 4 to supply hypotheses which are further away than 10 meters. Because the objects are represented as cuboids, we can extract their height and base radius, and use them as features for classification.
- *Shape matching:* An important feature for pedestrian recognition is the specific pedestrian shape. Having a 3D cuboid, we obtain its projection as a 2D box in the image. Edges in the 2D box are extracted using a Canny filter. The extracted edges are filtered, using depth, and only those edges which have a correct depth are retained. These form the outline of the pedestrian hypothesis. We apply a distance transform on these outline edges. The shape matching algorithm uses a hierarchy (tree) of pedestrian shape templates, constructed offline. A top-down matching algorithm based on the Chamfer distance is used to select the shape template that has the best match with the outline of the pedestrian, and the matching score is supplied as a feature for classification. The scale of template is inferred from the distance of the 3D box. Our approach is similar to [24], but makes more use of the available 3D information.
- *Tracking and motion related features:* Another powerful feature for pedestrian recognition is the specific way in which pedestrians move. Pedestrians are articulated objects, as opposed to most objects present in traffic environments which are rigid. Pedestrians move their arms and legs while walking or running. By tracking the individual motions of different body parts, and observing motion variations across them, we can supply a motion based feature, called "motion signature".

The pedestrian hypothesis is tracked using the tracking algorithm described in section 4. This step is useful for both object association and to eliminate the global object motion. The next step is to select the foreground points of each object. Only points for which their 3D coordinates lie inside the tracked cuboid are considered. This step is important as it eliminates spurious background points and deals with partial occlusions. Corner points are detected, and 2D optical flow vectors computed using the approach described in [25]. The optical flow vectors are transformed from 2D to 3D using the available depth information supplied by stereo. Principal component analysis is used to find the principal direction of the 3D velocity field variation for each individual object. Variance is smoothed across frames, to increase its stability. The magnitude of this principal component represents the "motion signature". This motion signature is much smaller for non-pedestrians as compared to pedestrians, and thus it is a powerful feature for pedestrian detection. The spectrum of the motion signature variation in time is analyzed. Pedestrians display a typical periodic motion signature, while other types of objects display only impulsive noise. The cutoff frequency for the motion spectrum is much smaller for pedestrians as opposed to non-pedestrians. This makes the motion spectrum a powerful feature for pedestrian classification.

- *Bayesian classification:* A naïve Bayesian classifier is used to combine the extracted features (height, base radius, lateral and longitudinal speed, motion signature and the motion spectrum cutoff frequency). According to Bayes' formula, the probability of an object to belong to class C_1, if it displays features $F_1 \ldots F_n$ is:

$$P(C_1 \mid F_1, F_2, \ldots, F_n) = \frac{P(C_1)P(F_1, F_2, \ldots, F_n \mid C_1)}{P(F_1, F_2, \ldots, F_n)} \tag{18}$$

By assuming feature independence, switching from probability to likelihood and taking the logarithm we have:

$$\log\left(L(C_1 \mid F_1, F_2, \ldots, F_n)\right) = \frac{\log\left(P(C_1)P(F_1 \mid C_1)P(F_2 \mid C_1)\ldots P(F_n \mid C_1)\right)}{\log\left(P(\neg C_1)P(\neg F_1 \mid C_1)P(\neg F_2 \mid C_1)\ldots P(\neg F_n \mid C_1)\right)} \tag{19}$$

Finally we have:

$$\log\left(L(C_1 \mid F_1, F_2, \ldots, F_n)\right) = A + B_1 + B_2 + \ldots + B_n \tag{20}$$

Where A is the prior pedestrian likelihood and $B_1, \ldots, B_n$ are functions of feature values.

- *Feature Database and Learning Application*

In order to determine the A and B_k coefficients in equation 20, to obtain pedestrian shape templates and build the template tree, and to test our classification algorithm, a sufficiently large number of stereo image sequences had to be manually indexed. For each frame in an indexed sequence, the ground truth cuboids are stored, together with their class and extracted features. A manual labeling tool was developed, which uses the 3D cuboids detected by our hypothesis generation algorithms, and which asks the user to supply the ground truth class for each of them. Ground truth classes are tracked in order to allow more efficient labeling. From the set of indexed sequences, the subset of sequences where the ego vehicle was stopped is used to generate pedestrian shape templates for the pedestrian shape template tree. The tree is generated by an automatic clustering algorithm.

Figure 20 shows some results for pedestrian detection.

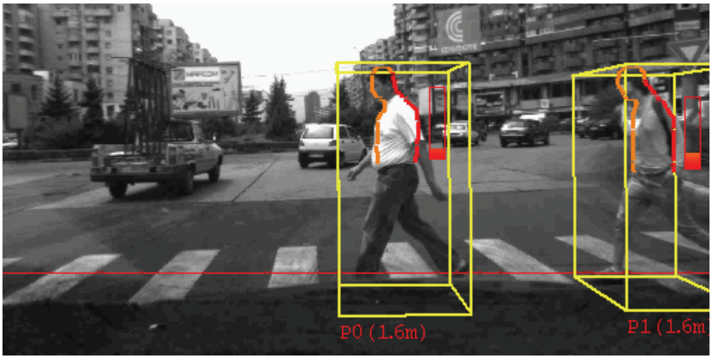


Fig. 20. Detected pedestrians

7. Possible use of stereovision in Driving Assistance Systems

The defining feature of a stereovision sensor is the capability of delivering rich meaningful data about the observed environment, combined with a reasonable precision of the 3D measurement. These features make the sensor suitable for a large variety of driving assistance applications.

The stereovision-based lane detection, capable of robust operation even when the lane markings are less than ideal or even absent, or when shadows or obstacles clutter the scene, can be used for *lane keeping and following assistance*, which can be implemented either as a form of warning or even as steering control.

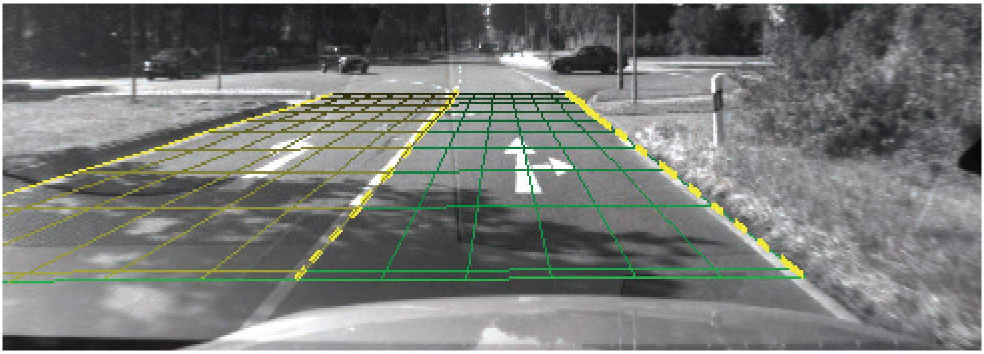


Fig. 21. Lane following assistance

Obstacle detection, in terms of 3D position, size and speed, provide the knowledge required for *collision avoidance*. The trajectory of the ego vehicle can be computed against the projected trajectory of the obstacles, the dangerous situations can be identified and appropriate measures such as emergency braking can be taken.

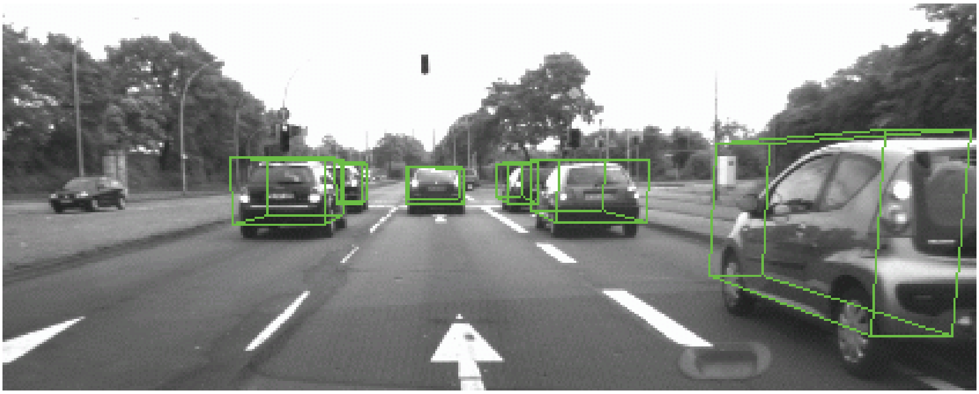


Fig. 22. Collision avoidance

A special type of collision is the one involving a pedestrian. The consequences of this collision are often very severe, and therefore it has to be avoided at (almost) all cost, even if some other type of unpleasant consequences may follow. This is the reason why the pedestrian has to be detected and classified as such as soon as it enters the field of view. The sensor we have described is thus suited for *pedestrian avoidance systems*.

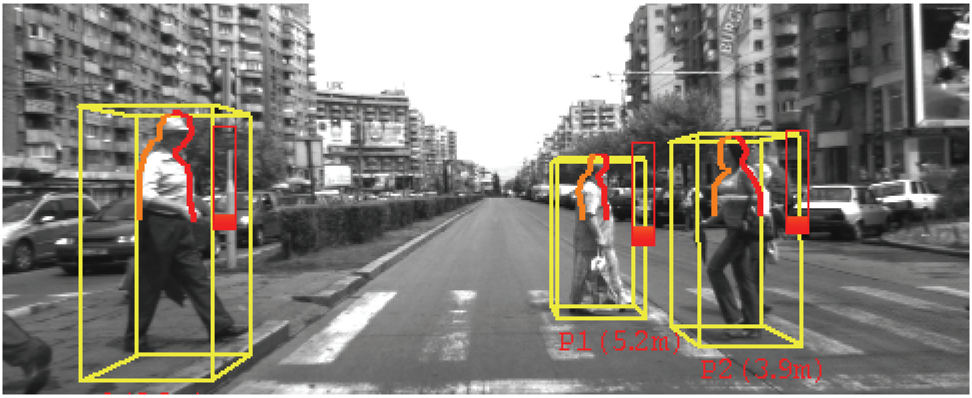


Fig. 23. Pedestrian avoidance

Accurate detection, measurement and tracking of the position, size and speed of the vehicle in front of us provides all the necessary information for a stop and go, (or follow the leader) assistance system, a helpful tool for the busy city traffic.

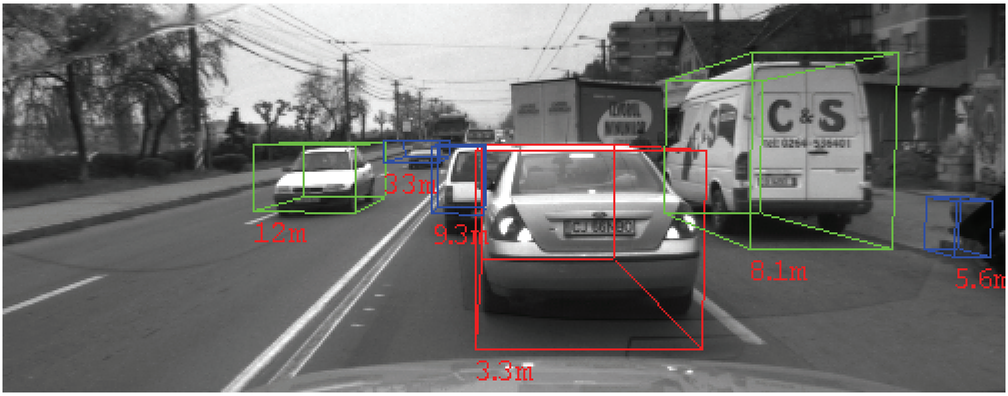


Fig. 24. Stop and go assistance

Sometimes, the environment is not easily represented in a structured way. In this case, the stereovision sensor can deliver information about areas that the vehicle is allowed to drive on, and about the forbidden areas, providing information for navigation assistance in unstructured environments.

8. References

[1] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, “Stereo Vision-based Vehicle Detection”, in *Proceedings of IEEE Intelligent Vehicles Symposium (IV 2000)*, October 2000, Detroit, USA, pp. 39-44.

[2] U. Franke, D. M. Gavrila, S. Görzig, F. Lindner, F. Paetzold and C. Wöhler, “Autonomous Driving Approaches Downtown”, *IEEE Intelligent Systems*, vol.13, no. 6, pp. 40-48, 1998.

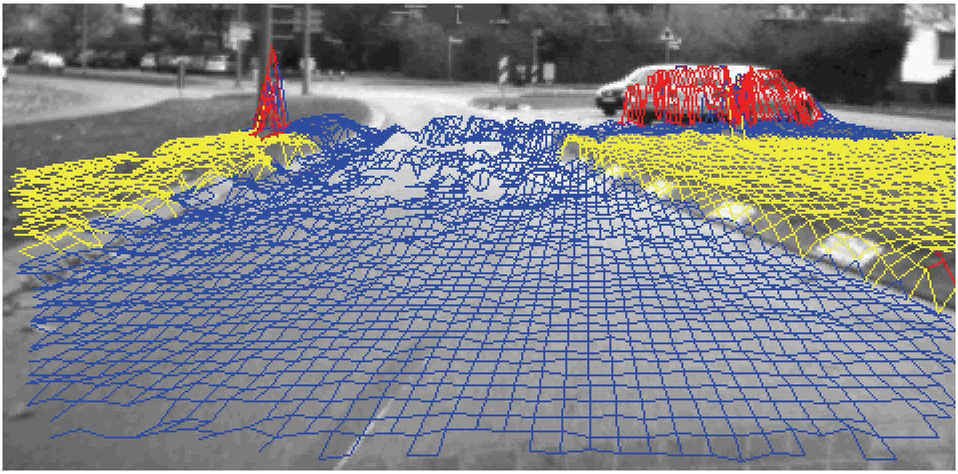


Fig. 25. Navigation assistance for unstructured environments

[3] T. A. Williamson, ″A high-performance stereo vision system for obstacle detection″, CMU-RI-TR-98-24, September 25, 1998, *Robotics Institute Carnegie Mellon University*, Pittsburg, PA 15123.

[4] M. Bertozzi and A. Broggi, “GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection”, *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62-81, January 1998.

[5] R. Labayrade, D. Aubert, and J.-P. Tarel, “Real time obstacle detection on non flat road geometry through V-disparity representation,” in *Proceedings of IEEE Intelligent Vehicles Symposium (IV 2002)*, June 2002, Versailles, France, pp. 646–651.

[6] 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.

[7] S. Nedevschi, R..Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, “3D Lane Detection System Based on Stereovision”, in *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC’04)*, October 2004, Washington, USA, pp. 161-166.

[8] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, T. Graf, R. Schmidt, “Driving Environment Perception Using Stereovision”, in *Proceedings of IEEE Intelligent Vehicles Symposium, (IV 2005)*, June 2005, Las Vegas, USA, pp.331-336.

[9] S. Nedevschi, F. Oniga, R. Danescu, T. Graf, R. Schmidt, Increased Accuracy Stereo Approach for 3D Lane Detection, *Proceedings of IEEE Intelligent Vehicles Symposium*, (IV2006), June 13-15, 2006, Tokyo, Japan, pp. 42-49.

[10] A. Sappa, D. Gerónimo, F. Dornaika, and A. López, “Real Time Vehicle Pose Using On-Board Stereo Vision System”, Int. Conf. on Image Analysis and Recognition, *LNCS*, Vol. 4142, September 18-20, 2006, Portugal, pp. 205-216.

[11] M. Cech, W. Niem, S. Abraham, and C. Stiller, “Dynamic ego-pose estimation for driver assistance in urban environments”, in *Proceedings of IEEE Intelligent Vehicles Symposium (IV 2004)*, Parma, Italy, 2004, pp. 43-48.

[12] Britta Hummel, Soeren Kammel, Thao Dang, Christian Duchow, Christoph Stiller, Vision-based Path Planning in Unstructured Environments, in *Proceedings of IEEE Intelligent Vehicles Symposium (IV 2006)*, June 13-15, 2006, Tokyo, Japan, pp. 176-181.

[13] 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'07)*, Sept. 30 - Oct. 3, 2007, Seattle, Washington, USA.

[14] F. Oniga, S. Nedevschi, M-M. Meinecke, "Curb Detection Based on Elevation Maps from Dense Stereo", in *Proceedings of the 3rd International IEEE Conference on Intelligent Computer Communication and Processing (ICCP 2007)*, Sept. 6-8, 2007, Cluj-Napoca, Romania, pp.119-125,.

[15] T. Marita, F. Oniga, S. Nedevschi, T. Graf, R. Schmidt, Camera Calibration Method for Far Range Stereovision Sensors Used in Vehicles, in *Proceedings of IEEE Intelligent Vehicles Symposium (IV 2006)* , June 13-15, 2006, Tokyo, Japan, pp. 356-363.

[16] S. Nedevschi, C.Vancea, T. Marita, T. Graf, "On-line calibration method for stereovision systems used in vehicle applications", in *Proceedings of 2006 IEEE Intelligent Transportation Systems Conference (ITSC'06)*, September 17-20, 2006, Toronto, Canada, pp. 957 – 962.

[17] 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, December, 2007.

[18] 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)*, September 1-2, 2006, Cluj-Napoca, Romania, pp. 135-142.

[19] R. Danescu, S. Nedevschi, M.M. Meinecke, T.B. To, "Lane Geometry Estimation in Urban Environments Using a Stereovision System", in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC'07)*, Sptember 30th- October 3rd, 2007, Seattle, USA, pp. 271-276.

[20] J. Goldbeck, B. Huertgen, "Lane Detection and Tracking by Video Sensors", in *Proceedings of IEEE International Conference on Intelligent Transportation Systems (ITSC'99), October 5-8, 1999, Tokyo Japan, pp. 74–79.*

[21] R. Danescu, S. Nedevschi, "Robust Real-Time Lane Delimiting Features Extraction", in Proceedings of *2nd IEEE International Conference on Intelligent Computer Communication and Processing (ICCP 2006),* September 1-2, 2006, Cluj Napoca, Romania, pp. 77-82.

[22] 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.

[23] S. Nedevschi, C. Tomiuc, S. Bota, "Stereo Based Pedestrian Detection for Collision Avoidance Applications", in *Proceedings of Workshop on Planning, Perception and Navigation for Intelligent Vehicle, at IEEE ICRA 2007*, April 10-14, 2007, Roma, Italy, pp. 39-44.

[24] D. M. Gavrila, J. Giebel, and S. Munder, "Vision-based pedestrian detection: the PROTECTOR system", in *Proceedings of Intelligent Vehicles Symposium (IV 2004)*, June, 2004, Parma, Italy, pp. 13–18.

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