Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle and Obstacle Detection

Florin Oniga and Sergiu Nedevschi
Technical University of Cluj–Napoca, Romania

***Abstract*— A new approach for the detection of the road surface and obstacles is presented. The high accuracy of the method allows the detection of traffic isles as distinct class. The 3D data inferred from dense stereo are transformed into a rectangular Digital Elevation Map (DEM). Two classifiers are proposed: density-based and road surface-based. The density-based obstacle classifier marks DEM cells as road or obstacles, using the density of 3D points as criterion. A quadratic road surface model is initially fitted, by a RANSAC approach, to the region in front of the ego vehicle. A region growing-like process refines this primary solution, driven by the 3D uncertainty model of the stereo sensor. A robust global solution for the road surface is obtained. The road surface is used for discrimination between road, traffic isle and obstacle points. Fusion and error filtering is performed on the results of the two classifiers. The proposed real-time algorithm was evaluated in an urban scenario and can be used in complex applications, from collision avoidance to path planning.**

***Index Terms*—stereovision, digital elevation maps, road surface detection, obstacle detection**

I. Introduction

Processing 3D data from stereo (dense or sparse) is a challenging task. A robust processing approach can be of great value for various applications in urban driving assistance. There are two main approaches (discussed next, and detailed in sections V and VI), depending on the space where processing is performed: disparity space-based and 3D space-based.

Disparity space-based algorithms are more popular because they work directly with the result of stereo reconstruction: the disparity map. The "V-disparity" approach [1] is widely used to detect the road surface. It has some drawbacks: 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 road profile becomes blurry and harder to detect on the "V-disparity" image). One way to avoid this problem is by using a specialized vehicle with small roll variations [2]. An extended "V-disparity" approach is presented in [3], where the roll angle can be computed assuming the scene has a planar road surface and assuming the presence of high-gradient road features (edges). The "U-V-disparity" concept, introduced in [4], is used to classify the 3D road scene into relative planar surfaces and to extract the features of roadside structures and obstacles.

3D space-based algorithms are used for ego-pose estimation ([5], [6]), lane and obstacle detection ([7], [8]), or path planning in unstructured environments ([9], [10], and [11]).

Digital Elevation Maps (DEM) are also used to represent 3D data from stereovision ([10] and [11]). A complex method for building the digital elevation map of a terrain (for a planetary rover) is proposed in [10]: 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 [11] 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.

A new road and obstacle detection algorithm will be presented in this paper. It transforms the 3D dense data from stereovision into a digital elevation map (DEM). The 3D uncertainty of the stereo sensor is modeled for improving the detection. A RANSAC-approach, combined with region growing, is used for the detection of the road surface. Obstacles and traffic isles are detected by using the road surface and the density of 3D points (Fig. 1). A temporal persistence filtering is proposed in order to have a robust detection of traffic isles.

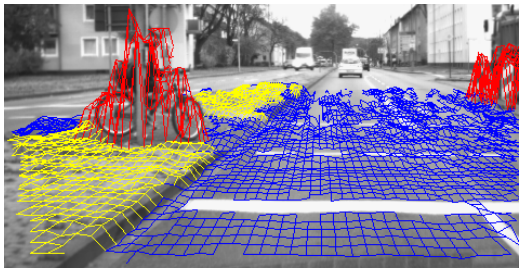


Fig. 1. The output of the algorithm is projected onto the left image: road area with blue, obstacles with red and traffic isles with yellow.

Manuscript received 13 December, 2008. This work was supported in part by Romanian Ministry of Education, Research and Youth under CNCSIS Grant ID_1522 no. 693/2009.

F. Oniga and S. Nedevschi are with the Computer Science Department, Technical University of Cluj-Napoca, Romania, email: florin.oniga, sergiu.nedevschi@cs.utcluj.ro, phone +40-264-401219.

This paper is organized as follows: section II presents an overview of the proposed algorithm and contributions; section III presents the mathematical support needed to model and fit the quadratic road surface; the main components of the algorithm are detailed in sections IV to VII; results, evaluation and failure cases are discussed in section VIII; and conclusions and future work in IX.

II. Overview of the Proposed Algorithm and Contributions

The road and obstacles detection algorithm (Fig. 2) presented in this paper takes as input dense 3D reconstructed points.

A Digital Elevation Map (DEM) and two density maps are computed from the set of 3D points (section IV). In contrast to the raw set of 3D points, the DEM provides a compact representation (grid), with explicit connectivity between adjacent 3D locations (easy to access). Furthermore, the set of 3D road points is reduced about 3-5 times (for multiple 3D points in a DEM cell, only one height is stored). Thus, the DEM representation helps to achieve real-time processing. To exploit this representation, we propose the following main contributions: an obstacle detection algorithm based on the density of 3D points per DEM cell (as a measure of the local slope), a combination of RANSAC, region growing and least square fitting for the computation of the quadratic road surface, and the detection of traffic isles.

A density-based algorithm for obstacle detection is proposed (section V): based on the density of 3D points, each DEM cell is classified as obstacle or road.

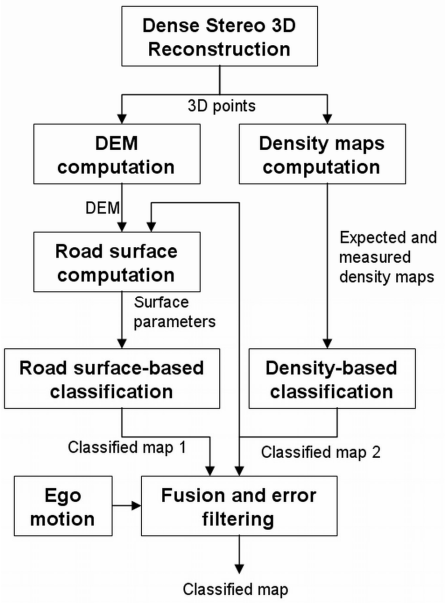


Fig. 2. Overview of the proposed algorithm.

The road surface is fitted, using a RANSAC approach, to a small patch in front of the ego vehicle. This primary surface is then refined through a region growing-like process. The robustness of the global road surface is insured in this way. Based on the road surface, DEM cells are classified as road, traffic isles or obstacles (section VI). Traffic isles (elevated surfaces parallel to the road, such as sidewalks) are detected as a distinct class, instead of being considered road inliers.

The results of the density-based and of the road surface-based classifiers are filtered and fused to obtain a robust result (section VII). The ego motion is used to validate traffic isles.

The output of the algorithm is multiple: the quadratic surface of the road in parametric form, 3D points classified as road/traffic isles/obstacles, and cell clusters representing individual traffic isles or obstacles.

III. The Quadratic Road Surface Model

The planar road model widely used can be less robust for obstacle/road separation, especially for the detection of road delimiters such as curbs (ex. border of a sidewalk). The road surface can present longitudinal and lateral curvatures, and the planar assumption cannot cope with these (vertical) curvatures (see figure 13 for results on real data). Even though the lateral curvature is small, it must be taken into account in order to detect small curbs of several centimeters in height (at least five).

First, we will present how the surface is computed for a given set of 3D road points. Then, the region of road inliers is defined, as a function of the stereo configuration.

A. Fitting the surface to a set of 3D points

The following world reference frame is used: Z-axis represents the depth (longitudinal), Y-axis represents the height (vertical), and X-axis is the lateral offset. The origin is placed on the ground, at the center of the ego's front bumper.

We used a road model that allows quadratic variations of the height (Y) with the horizontal displacement and the depth. This limited model copes with normal driving on most urban roads. If a specific application needs more generality, it can be extended straightforward to cope with any orientation of the vehicle on the road surface (by adding the XZ term).

Equation (1) shows the algebraic form of the road model, by defining the height value Y with respect to Z and X.

$$Y = -a \cdot X - a' \cdot X^2 - b \cdot Z - b' \cdot Z^2 - c \,. \qquad (1)$$

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:

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

where Y_i is the height of the 3D point i and $\overline{Y_i}$ is the height of the surface at coordinates (X_i, Z_i).

Minimizing only along the Y-axis (instead of the surface normal) is common practice and provides enough accuracy. 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 (1) into (2), the function S is obtained, where the unknowns are *a, a', b, b',* and *c*:

$$S = \sum_{i=1}^{n} \left(Y_i + a \cdot X_i + a' \cdot X_i^2 + b \cdot Z_i + b' \cdot Z_i^2 + c \right)^2 . \quad (3)$$

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:

$$\left\{ \frac{\partial S}{\partial a} = 0, \frac{\partial S}{\partial a'} = 0, \frac{\partial S}{\partial b} = 0, \frac{\partial S}{\partial b'} = 0, \frac{\partial S}{\partial c} = 0 . \right. \quad (4)$$

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

$$\begin{bmatrix} S_{X^2} & S_{X^3} & S_{XZ} & S_{XZ^2} & S_X \\ S_{X^3} & S_{X^4} & S_{X^2Z} & S_{X^2Z^2} & S_{X^2} \\ S_{XZ} & S_{X^2Z} & S_{Z^2} & S_{Z^3} & S_Z \\ S_{XZ^2} & S_{X^2Z^2} & S_{Z^3} & S_{Z^4} & S_{Z^2} \\ S_X & S_{X^2} & S_Z & S_{Z^2} & n \end{bmatrix} \begin{bmatrix} a \\ a' \\ b \\ b' \\ c \end{bmatrix} = \begin{bmatrix} -S_{XY} \\ -S_{X^2Y} \\ -S_{ZY} \\ -S_{Z^2Y} \\ -S_Y \end{bmatrix}, \quad (5)$$

Generically, each sum is computed as $S_\alpha = \sum_{i=1}^{n} \alpha_i$ (for example $S_{XZ} = \sum_{i=1}^{n} X_i \cdot Z_i$). (6)

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

$$S_\alpha = \sum_{i=1}^{n} w_i \cdot \alpha_i \text{ (for example } S_{XZ} = \sum_{i=1}^{n} w_i \cdot X_i \cdot Z_i \text{)} . \quad (7)$$

System (5) 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 VI.B.

B. The 3D uncertainty from stereo. Road inliers and outliers

The 3D (localization) uncertainty is caused by a low accuracy evaluation of the disparity values (from the dense stereo module) and is present along the projection ray of each 3D point. The components of the 3D uncertainty along the Z-axis and along the Y-axis, respectively Z_{err} and Y_{err}, must be computed in order to estimate the vertical interval for the inliers of the road surface.

Let us consider a location $P(X_R, Y_R, Z_R)$ on the road surface (the depth Z is measured from the camera, Fig. 3.b). A simple model for the uncertainty of the depth was proposed in [12] for a canonical stereo system. The depth uncertainty Z_{err} of P is modeled as a function (8) of the system's parameters (baseline B and focal F, measured in pixels, are known from calibration) and of the disparity uncertainty D_{err}.

$$Z_{err} = \frac{-Z_R^2 \cdot D_{err}}{B \cdot F - Z_R \cdot D_{err}} . \quad (8)$$

We extended this model to compute the uncertainty Y_{err} of P, as in (9). H is the height of the camera in the world reference frame.

$$Y_{err} = \frac{(Y_R - H) \cdot Z_{err}}{Z_R} . \quad (9)$$

The maximum allowed vertical error for a point to be considered inlier is expressed as a function of D_{err}:

$$\Delta Y(D_{err}) = Y_{err} + Y_{errz} = Y_{err} - Z_{err} tg(\alpha) , \quad (10)$$

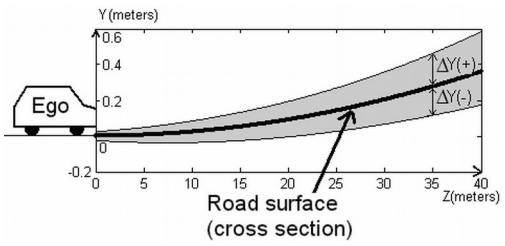
where the surface slope in P is computed by deriving the surface equation (1) with respect to Z:

$$tg(\alpha) = -2b' Z_R - b . \quad (11)$$

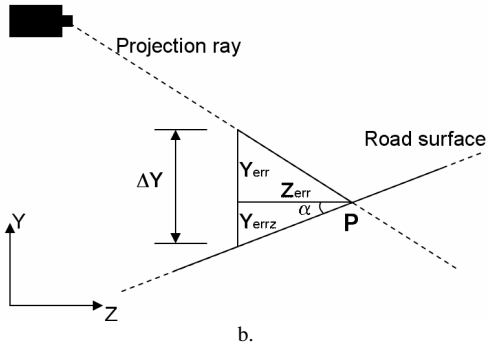
Estimating the disparity uncertainty D_{err} for each point is very difficult and time consuming. The road surface usually presents poor texture, corrupted by image Gaussian noise. Considering that the sub-pixel accuracy provided by the dense stereo matching is less reliable for the road surface, D_{err} is likely between -0.5 and +0.5 pixels. For future improvements, a matter worth investigating is the use of image regions with different D_{err} values.

An unknown point P_U at depth Z_R and height Y_U is considered road inlier if it is placed between the lower (for D_{err}=-0.5) and the upper bounds (D_{err}=+0.5) for inliers. The bounds (Fig. 3.a) can also be increased by small amount Δh (=2.5 cm) to compensate small road artifacts (bumps, leaves etc) at small depths.

$$Y_R + \Delta Y(-0.5) - \Delta h < Y_U < Y_R + \Delta Y(+0.5) + \Delta h . \quad (12)$$



a.



b.

Fig. 3. a. Lateral view (a slice from 3D with a constant value for the X coordinate) of the inliers (gray) region around a quadratic road. The ego position is shown (the ego size is over-scaled). b. The geometric construct used for computing the height uncertainty.

Formulas presented in this section will be applied in the elevation map space (it is still a 3D space, but with discrete locations, and each cell can be described 3-dimensionally as the map coordinates and height).

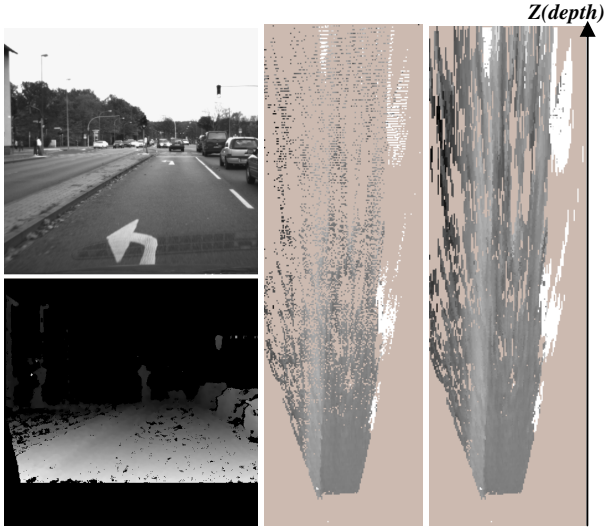
IV. Building the DEM and the Density Maps

A. Computing the DEM

The DEM is represented as a rectangular grid (matrix) of cells, with the same aspect ratio as the 3D space of interest. The DEM is built using the instant 3D data provided by stereo ([16]) for the current frame. A 3D space of interest (40 m x 13 m from bird-eye view) in front of the car is considered. The longitudinal Z and lateral X coordinates of each 3D point are scaled into the DEM space. Each DEM cell will store the highest height of the 3D points contained within the cell. A cell has a size of 10 cm x 10 cm in the horizontal XZ world plane (it can be adapted to different sizes, depending on the application requirements).

The world zero Y-level (road level at system calibration) is centered at 128 in the DEM for better visualization of the road and traffic isles heights. The height value is also scaled. Only heights around the road (1 m band) are displayed correctly in this paper due to a limited number of 256 gray values (Fig. 4.b, heights outside this interval are displayed saturated either to 0 or 255).

3D points higher than 2 meters from the ground (zero level from calibration) will not be stored because they are out of interest. For scenarios with extreme up-hills, the 2-meters interval can be measured relative to the road surface detected in the previous frame (between consecutive frames the road profile changes smoothly). Empty cells are marked and not used further.

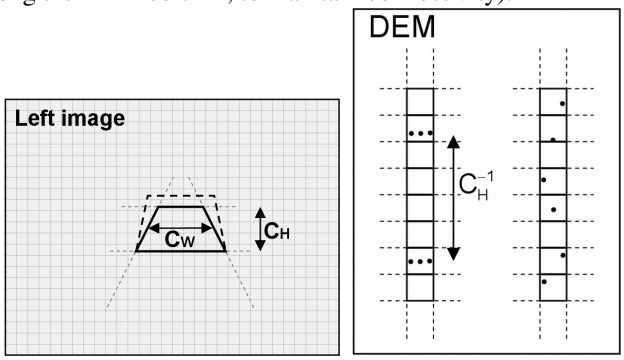


a. b. c.

Fig. 4. a. A common traffic scenario (top) and the disparity image (bottom), b. The DEM, empty cells are marked with color, c. The enhanced DEM, with improved connectivity along the depth.

B. Improving the connectivity of the DEM road cells

As seen in Fig. 4.b, the DEM presents poor connectivity between road points at far depths. Connectivity can be improved by propagating heights from each valid cell to adjacent empty cells, along the DEM column. The maximum propagation range is inversely related to the road depth resolution of the considered cell (if the cell is further away, its height data should be propagated to more adjacent empty cells, along the DEM column, to maintain connectivity).



a. b.

Fig. 5. a. The projections, with sub-pixel accuracy, of the flat road patch (solid line trapezoid) and of an uphill road patch (dashed line trapezoid) onto the left image. b. The distribution of road points from the same image row have the same depth (the left DEM column) assuming an ideal depth recovery, but are practically scattered along the depth (the right DEM column).

A DEM cell C covers a square (LxL) in the XZ plane, or a rectangle if the slope is not zero, of the 3D reference frame. The projection, with sub-pixel accuracy, of the square onto the left image is a trapezoid (Fig. 5.a, solid line) with height C_H and average width C_W, measured in pixels. The height C_H indicates how many image rows contribute to the 3D points from the cell. When C_H is below 1 then lack of connectivity might appear (empty cells along the depth).

Height data from valid cells can be propagated to empty cells, along the DEM column (Fig. 4.d). The distance along a DEM column (Fig. 5.b), in cells, between two consecutive DEM cells with valid data is $\Delta P = C_H^{-1}$. For each empty DEM cell C_1, the closest DEM cell C_2 with valid data is selected, on the same DEM column. If the distance between C_1 and C_2 is less than ΔP_h (half of ΔP) then the value of C_2 is copied into C_1.

This insures theoretically that all road DEM cells are filled with valid heights for a flat road surface (the XZ plane), even if the 3D reconstruction is ideal (road points from the same image row have the same depth, Fig. 5.b the left DEM column). With real images, the 3D road points from the same image row are spread along the true depth value (Fig. 5.b the right DEM column), so ΔP might actually be considered smaller (even divided by C_W). However, we did not reduce it in order to insure connectivity also for down hills. For stereo configurations with small focal length, if C_W drops below one from a certain depth, then ΔP can be even increased.

Connectivity is needed for the step of surface growing (section VI.B). The propagated heights will insure a uniform distribution of height data along the depth when computing the road surface.

C. Computing the density maps

Two maps, related to the density of 3D points, are computed for each cell: *expected road density map* and average *measured density map*. These two features are used for fast discrimination between road and non-road features (Section

V).

a. b. c. d.

Fig. 6. a. The average measured density map (3D points/cell), in a logarithmic form, b. The road expected density map in a logarithmic form, c. The difference between the measured and the expected density maps: only positives values are shown (vertical scene items are brighter), d. The left image with the vertical scene items numbered.

The expected road density of 3D points for a DEM cell C is the area of its trapezoidal image projection (Fig. 5. a). It can be approximated as:

$$erd(C_{Slope=0}) = C_W C_H \,. \tag{13}$$

The expected density has a large range of values from about 200 3D points/cell near the ego-vehicle, down to 0.1 points/cell at 35 meters depth (in Fig. 6.c, shown in a logarithmic form).

To compute the actual density of 3D points (*measured density*), a counter is stored for each cell (when the DEM is built) that shows how many 3D points are contained by the cell. For road cells at far depths, this counter is not equivalent with the real density of 3D points. The measured density (Fig. 6.b, logarithmic form) can be estimated by averaging the map of counters with an adaptive mask, with the size equal to $2\Delta Ph+1$ (ΔPh is the distance used for connectivity, previous sub-section):

$$md(row,col) = \frac{1}{2\Delta P_h + 1} \sum_{k=-\Delta P_h/2}^{\Delta P_h/2} Counters(row+k, col) \,. \tag{14}$$

V. Density-based obstacle classification

The number of reconstructed 3D points is much greater if it contains a close to-vertical surface instead of the close-to horizontal road surface (due to the perspective projection). This is exploited in various ways in the literature. In [7], obstacles are detected as clusters of image edge points reconstructed in the 3D space. Road features as lane-markings can also be detected as obstacles. An algorithm for off-road navigation is presented in [9]: 3D points are clustered into obstacles based on their relative height and slope. A cone-shaped region is used to reduce the search space around each 3D point when performing clustering. The algorithm was fast but only close to real-time (0.5-1.5 Hz), and the response time was dependent to the scene's richness of 3D points. The gradient of the DEM is used in [11] to evaluate the navigability of the environment.

Our approach also relies on the fact that obstacles cells have much larger densities than the road for the same depth (Fig. 6. c, the difference between the expected road density and the measured density maps). There are two main reasons we preferred to use a planar (flat) road as the reference for comparison (road expected density). This classifier should work even for scenes where the road surface cannot be detected (poor 3D reconstruction, with no or few 3D road points). In addition, it should have a low computational complexity: the road expected density is computed offline, not for every frame (if the "per-frame" detected quadratic surface would be used).

If the road is non-flat, then cells containing the road surface will have a different density than the flat road: higher for uphills (Fig. 5.a, the dashed line trapezoid has a larger area), and lower for downhills. To avoid uphill road patches to be detected as obstacles, we propose a double thresholding technique:

- Cells are flagged as **Obstacle** if the measured density of the cell is higher by T_H times than the estimated road density,
- Other cells are flagged (recursively) as **Obstacle** if they are adjacent to an **Obstacle** cell and the measured density of the cell is higher by T_L ($=T_H/2$) times than the estimated road density.

The quotient T_H is computed as a function of the maximum slope allowed for the road surface at a particular DEM location. The expected road density for a cell containing a road patch with a particular slope is computed as shown in section II.B, as the area of its trapezoidal image projection.

$$T_H = \frac{erd(C_{Slope=40\%})}{erd(C_{Slope=0\%})} \,. \tag{15}$$

We used a value of 40% (about 22 degrees) for the slope, which covers even extreme uphills for urban roads. An additional constraint can applied at the first step of the double thresholding to reduce false detections: the height variation of each obstacle DEM cell must be above a threshold (the height uncertainty plus a minimum obstacle height value Δh_o of 5 to 10 cm).

The method is used as a stand-alone detector (if the road surface is not valid) or it can be fused with the road surface-based classification (section VI.C).

VI. Road surface detection and classification

The ego-pose is estimated relative to the road plane in [5]. The road plane is fitted (not in real-time) by a RANSAC-approach to the whole set of dense 3D points (after filtering non-road points). A constant band around the road is used to select inliers and outliers, even though this is against the fact that the 3D uncertainty from stereo increases with the depth. The assumption that most of the 3D points are road points is made again (if the ego car is close to a sidewalk with more 3D points that the road, it is likely to fail). A planar road surface is estimated from tracking in [6]. The method provides robust numerical results, but fails if occlusions (obstacles) are in front

of the ego car. Lack of high-gradient road features also leads to failure (lane markings, borders, etc.). A 3D lane model is proposed and used for obstacle/road points' separation in [8]. Again, the method requires high-gradient road features (edges) to be present and uses a constant band to select road inliers and outliers.

The RANSAC approach [13] is a robust method for fitting a model to a data set containing outliers. Instead of fitting the model to the whole set in a least square fashion (due to outliers the solution will not be accurate), the RANSAC approach chooses a number of samples (subsets of the data set). For each sample, the model is fitted and a score is computed. The sample with the highest score is selected.

Fitting the surface to the whole DEM can be very unstable and time consuming due to a large number of road outliers. Therefore, our solution involves the following steps:

- *Selection of cells for initial surface fitting*: a rectangular DEM region of interest, in front of ego, is analyzed to filter most road outliers; a RANSAC technique is then employed to detect the primary road surface.
- *Uncertainty model-driven surface growing*: the initial surface is refined gradually, by taking into account the uncertainty model of the stereo sensor.
- *Road surface-based classification*: cells are classified as road, traffic isles or obstacles, based on their position and density relative to the road surface.

A. *Selection of cells for initial surface fitting*

A rectangular patch from the DEM is selected in front of the ego car. Valid cells from this patch will represent the data set. Not all the valid cells will be used because there are two situations when even RANSAC can fail (Fig. 7): when traffic isles or obstacles are dominant in the selected patch.

Two constraints are used to filter the data set (Fig. 8.c) before applying RANSAC:

- All cells, previously classified as obstacle by the density-based classifier (section V), are rejected.
- Linear curbs are detected in the rectangular patch. Cells placed on the opposite side as the ego car, relative to the detected curbs, are rejected. Simple 2D geometry is employed, considering that the front of the ego car is placed, in the DEM space, on the middle column and the bottom row.

In [17] a Time-of-Flight camera is used for acquiring range data. Curbs are identified based on the detection of individual planar patches (adjacent to the curb) using a modified version of RANSAC. Image edge points are detected in [18]. A weight is computed, for each edge point, as a function of the image brightness gradient and the 3D elevation gradient (computed from stereo). These weights are used for voting in the Hough accumulator. One dominant straight curb (per scene) is extracted as the line with the maximum score in the accumulator. In [19] the detection starts alongside the vehicle, with a laser line striper, by searching the specific height variation of curbs. Then the curb is extended and tracked in front of the vehicle, based on a video camera and the ego motion.

Fig. 7. Two situations where RANSAC can fail: large traffic isles or obstacles in front of ego. The patch selected for initial fitting is shown as a white rectangle.

Our solution for curb detection is also based on the Hough transform [15]. Edge points are computed on the DEM, using the Canny edge detector [14] to detect specific height variations. The Hough transform is built and 5 relevant lines are selected, having the highest Hough scores. Each line is analyzed by counting how many of its points have a height variation between 5 cm and 35 cm (normal range for curbs). Lines with a score higher than 40% of the total number of line points are considered valid. At most two of the valid lines (left/right), with the highest scores, are selected as curbs.

False curbs may be detected, but they pose a problem (filter out most of the valid road cells) only if they have a lateral orientation and are placed close to the front of the ego car. This situation was extremely uncommon: from a large urban sequence of 10450 frames, such bad curbs were detected only in 2 frames.

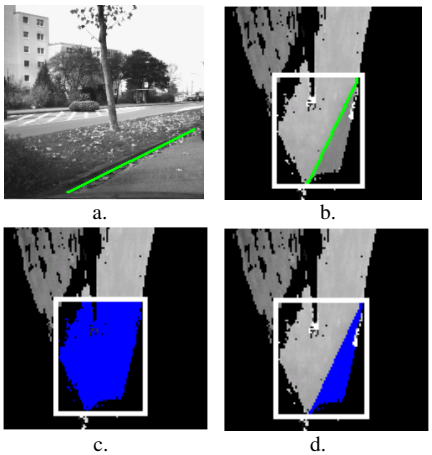


Fig. 8. a. The curb is projected onto the left image, b. The curb is detected on the DEM (green line – only the segment overlaying valid height data is shown), c. Cells with valid height are shown with blue, d. Highly probable road cells are selected for RANSAC.

One issue emerged regarding least square fitting of quadratic surfaces: numerical stability for a small set of cells. A planar surface would require at least 3 non-collinear cells. However, a quadratic surface requires at least 5 cells and the non-collinear constraint is not enough. Various configurations

of cells can lead to unstable fitting (ill-conditioned): quasi-circular cells, cells placed on two quasi-parallel lines, etc.

Therefore, a planar model was fitted in order to select the set of primary road inliers.

The RANSAC method is applied to the filtered set of cells. We used a number of K samples of 3 cells each. The value of K can be computed ([13]) as a function of the success probability p and the percentage of inliers w in the data set. A value of K=86 samples was obtained for p=0.99999 and w=50% (a very conservative value, since the data set is already filtered of most road outliers).

The planar surface was computed for each sample. The score for each sample was considered as the number of inliers from the whole set. The uncertainty model from section III.B was used to classify a cell as part of the road surface. The sample with the highest number of inliers is selected as the best planar road surface. If the total surface of inliers is less than (equivalent in 3D) 1 m^2, then the detection of the road surface is aborted and only the density-based method proposed in section V is used for obstacle detection.

The quadratic model is fitted (section III.A) to the set of inliers of the best planar surface, and the primary quadratic road surface is obtained.

B. *Uncertainty model-driven surface growing*

The primary road surface is detected correctly for the cells in front of the vehicle. Correctness is not granted for the whole scene because vertical curvatures of the road surface can be computed correctly only using large road patches.

The primary solution can be refined through a region growing process (Fig. 9) where the initial region is the set of road inliers from the initial rectangular patch. A new cell can be added to the current region if it fulfills the following conditions:

- It is adjacent to inliers from the current region,
- It verifies the current road surface equation according to the uncertainty model (section III.B).

The surface is re-computed, in a least square-fashion, each time the region has expanded its border with 1-2 pixels (about 50-100 new cells). This insures that the surface is refined gradually (Fig. 9.a). On average, the surface is recomputed about 150-200 times per frame. This can be very slow since it involves computing the sums defined by (6), in section III.A. A real-time implementation is possible by using the partial sums between two consecutive re-computations (only the amounts for the new cells are added).

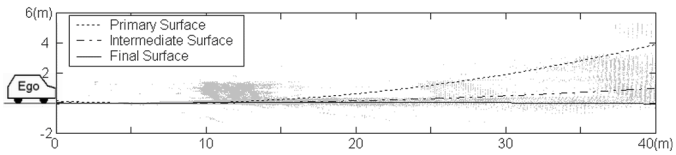
A global accurate solution for the road surface is obtained after this step. The iterative refinement of the surface is fast and stable to outliers and numerical errors: for each iteration only (most-likely) road cells are added to the current region and the parameters are re-computed using least square fitting.

C. *Road surface-based classification of cells*

Elevation map cells are classified based on the road surface, into road, traffic isles and obstacles (Fig. 10.c).

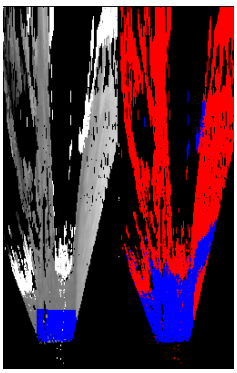
Both traffic isle and obstacle cells have the property of being elevated relative to the road surface. Two properties of traffic isles will help discriminate them from obstacles:

1. They have a height in a limited interval relative to the road.
2. They are mainly made of surfaces parallel to the road (next section).

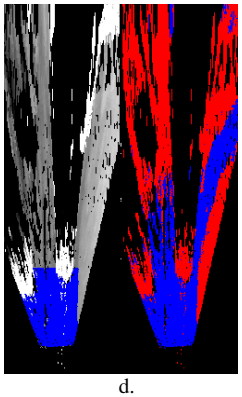


a.

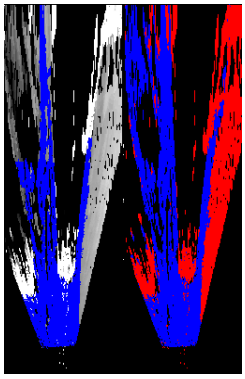
b.



c.



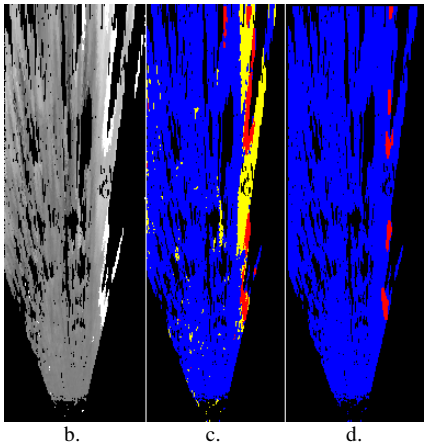
d.



e.

Fig. 9. a. The lateral road surface profile (X=0), 3D points are shown with gray, b. The left image, For each image (c, d, and e), 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: c. For the primary surface, d. An intermediate surface, e. The final surface.

a.



b. c. d.

Fig. 10. a. A traffic scene, b. The DEM, c. The result of the road surface based classification (blue - road, yellow - traffic isle, red - obstacles), d. The result of the density-based classification.

The following rules are used for the classification of a DEM cell P:

- If the height of P relative to the road is lower than the estimated height uncertainty for the cell location, then the cell is considered **Road,**
- Otherwise,
  - If the height of P relative to the road is in a specific interval (5-35 cm) and it was not classified as obstacle by the density based classifier, then P is considered **Traffic Isle,**
  - Otherwise,
    - If the height of P from the road is above a threshold (a minimum obstacle height plus the height uncertainty), the cell is considered **Obstacle**,
    - Otherwise P is considered garbage (contains erroneous 3D data).

VII. Fusion and Error Filtering

The road surface-based classifier (section VI.C) provides good results but also some false positives (both traffic isles and obstacles, Fig. 11.a). On the other hand, the density-based classifier (Fig. 11.b) provides good results for obstacles (especially for close-to vertical sides).

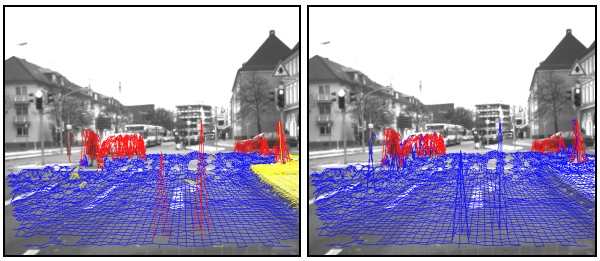
Fusion must be performed between the results of the two classifiers. Some basic constraints upon cell clusters can be used to filter false traffic isles. Traffic isles are placed above the road surface, thus cells containing traffic isles have lower densities of 3D points (compared to the underlying road surface). This is motivated by the fact that the solid angle (made by a close-to-horizontal surface patch and the camera location) decreases as the surface patch is translated upward on the vertical axis. Thus, fewer 3D points are reconstructed for a traffic isle (resulting lower density in the DEM cells), compared to the situation where the traffic isle is not present (the lower road surface patch is reconstructed, with a higher expected density).

The following rules are used for error filtering and fusion:

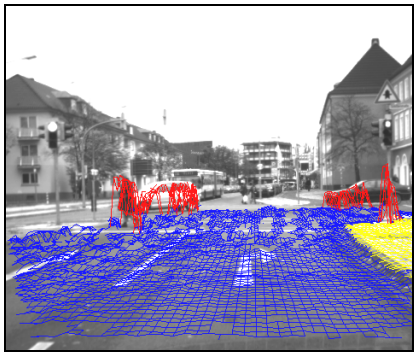
- Small **Traffic Isle** areas from the rough classification are discarded (less than 0.5 m^2),
- **Traffic Isle** areas having an average measured density higher than the average road expected density in the DEM cells are discarded,
- **Obstacle** areas from the road surface-based classification are marked as false elevations if they do not overlap **Obstacle** areas from the more robust density-based classification,
- For depths higher than 30 meters, only the result of the density-based classification is used.

The last rule is justified because the height uncertainty increases with the depth. For our stereo configuration, a 3D road point at a depth of 30 meters can have a height uncertainty of 17 centimeters (Fig. 3.a). Our experiments showed that, for depths higher than 30 meters, traffic isles detection is more or less random and provides many false positives. Detection of traffic isles is usually reliable up to 20-25 meters, depending on the quality of the stereo reconstruction.

False elevations can either be displayed as drivable (road, Fig. 11.b), or they can be invalidated and not displayed, as in Fig. 11.c.



a. b.



c.

Fig. 11. a. The output of the road surface-based classifier is projected onto the left image. Grid vertexes represent the centers of the DEM cells, b. The output of the density-based classifier, c. Results after fusion and error filtering (false 3D spikes are removed).

We evaluated the algorithm (with the steps described so far) and discovered that small false traffic isles can appear (11 false traffic isles/200 frames, see section VIII). Although less than 5% of the detected traffic isles are false, they can appear in front of the ego car and cause false collision situations.

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, we propose a fast (about 2 ms) and efficient approach to filter false traffic isles.

For each frame, 3D points are represented in a system of coordinates that moves with the ego car (translation and rotation). Coordinates of a DEM cell from the current reference frame O(t) can be transformed into the previous reference frame O(t-1), assuming the translation d and rotation angle α are known. The ego car's standard speed and yaw-rate sensors can be used to estimate these parameters. The following motion model is employed: the ego has a circular trajectory between successive frames, and the arc length and radius are computed based on speed v, yaw-rate angle γ and frame relative timestamp Δt.

Two steps are performed for validating a traffic isle cell over two consecutive frames (t and t-1). For each traffic isle cell C(i, j) of DEM(t):

1. Compute the coordinates of C in the previous frame DEM(t-1), as real numbers (i', j').
2. If at least one of the four DEM(t-1) closest cells of

(i',j') is a traffic isle then C is validated as a traffic isle cell in the current frame. Otherwise it is rejected (marked as invalid).

Using a 2x2 DEM cell neighborhood around the non-integer coordinates (i',j') is required to compensate some sources of errors: lack of accuracy from the ego sensors, pitch angle variation might occur between frames (due to road bumps etc), causing small depth shifts.

Extending the process to three (or more) consecutive frames (t, t-1 and t-2) is straightforward. We tested the transformation between consecutive frames for up to 5 frames (from t to t-4). The car sensors proved accurate enough so that an item (a thin pole was used for verification) is located in frame t-4 with an error (in the horizontal plane) less than the DEM cell size (10 cm). As it will be shown in the next section, the persistence-based filtering is robust.

VIII. Results and evaluation

The algorithm was implemented in C++. The dense 3D information was generated using a calibrated stereo rig with grayscale cameras and a commercial dense stereo board [16]. Image resolution was 498x468 pixels, with a focal length/FOV of 811 pixels/34 degrees, and a baseline of 220 mm.

Table 1. Main threshold values used

Derr	Δh	Max. road slope	Δh_o	Hough Accumulator (deg x pixels)	Curb/traffic isle interval
0.5 pixels	2.5 cm	40%	7.5 cm	360 x 90	5..35 cm

Due to the use of software-specific C optimizations and the DEM representation, an average processing time of 22 ms was achieved for the whole algorithm (on Pentium 4 at 2.6 Ghz). Overall, with the image acquisition and the dense hardware reconstruction, a sustained processing frame-rate of 23 fps is obtained. An average total processing time of 10 ms is achieved on a more recent processor (Pentium Dual Core).

A. Evaluation

The evaluation of the algorithm with real scenarios was time consuming. We were mainly interested in evaluating the capability of the algorithm to detect relevant scene items. For each test scenario, the classification result was projected from the DEM space onto the left image. Visual analysis (by a human observer) of each image was performed. Results were analyzed for each frame in terms of: missed obstacles, missed traffic isles, false obstacles, and false traffic isles.

A number of 200 stereo images of different (random) scenes were selected (out of two hours of stored stereo images, recorded while driving the ego-car in an urban environment).

Overall, for the 200 stereo images, the numbers obtained proved the robustness of the algorithm (only "per-frame" detection, without the step of persistence filtering for false traffic isles):

- Missed obstacles: 16 out of 484 - very small objects, such as (thin) poles smaller than 30 centimeters in height (Fig. 14.e); relevant obstacles, such as pedestrians or vehicles, were not missed.
- False obstacles: 1 small obstacle, at far depths (>30 meters) for sharp uphill roads (Fig. 14.d). These false obstacles were not persistent between frames, so they can be filtered through tracking. They only appeared at far depths so the navigability is not influenced.
- Missed traffic isles: 11 out of 234 - the isles height was too small (3-5 cm), not "sensed" by the stereo system, they are mainly drivable areas such as bus stations (Fig. 14.f).
- False traffic isles (Fig. 14.a): 11, small traffic isles that persist only few frames (1-2).

When the additional persistence-based filtering is used, the detection of traffic isles is more robust. For the 200 images, all false traffic isles are rejected. We performed an additional qualitative evaluation (only to count false traffic isles) on an offline sequence of 6250 frames. By using a 2-frames persistence condition, the rate of false traffic isles decreased to 1 at about 370 frames. The 3-frames persistence condition provided much better results: only two false traffic isles along the whole sequence.

The detection errors are not caused by the way dense stereo data is represented. They are caused by the low quality of the 3D reconstruction.

The difference between a quadratic surface and a planar one is shown in figure 13 (both are detected with the proposed algorithm). The mean absolute error of the reconstructed 3D road points relative to the computed surface is about 4 times larger for the planar model, in contrast to the quadratic one.

The quadratic road model proves less effective in scenes with more complex road geometry. A large transversal false traffic isle appeared on the border between the quadratic surfaces (Fig. 14.b): the road surface was made of two main quadratic surfaces with different parameters. A more complex model should be investigated.

Modeling the 3D environment with a digital elevation map is not always optimal. Floating items (not rising from the road), such as tree branches (Fig. 14.c) or rough noise from dense stereo, can be a source for false obstacles. The DEM model must be extended to store the set of 3D points contained by each cell, for further processing. Additional noise filtering can be performed during the DEM building stage.

IX. Conclusions

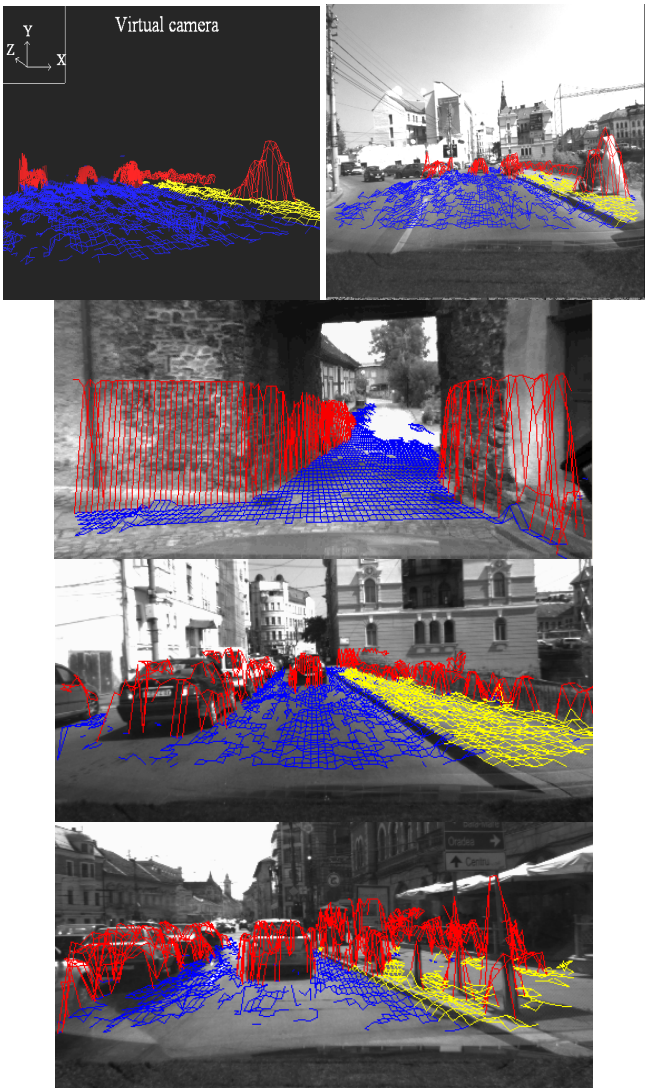
The main contributions presented in this paper are: an obstacle detection method based on the density of 3D points per DEM cell (as a measure of the local slope), a combination of RANSAC, region growing and least square fitting for the computation of the quadratic road surface, and the detection of traffic isles as a distinct class along with other relevant obstacles (pedestrians, vehicles, poles etc.).

A real-time algorithm was proposed based on these contributions. It takes as input dense 3D points, thus road edge features (high gradient) are not required. To achieve real-time

processing (and almost constant), the 3D set of points is transformed into a digital elevation map. The road is modeled as a quadratic surface to allow vertical curvatures, often present in urban scenarios and the 3D uncertainty increasing with the depth is taken into account. Possible lack of road dense 3D data, due to poor road texture, is compensated by the density-based obstacle detection method.

It has its own failure cases, as discussed in section VIII, and other future developments (in addition to those presented in sections V and VIII.B) are required:

- Additional evaluation of the proposed algorithm,
- Using tracking for obstacle blobs can greatly improved the robustness of the method,
- A more complex road model will be proposed and tested (cubic or 4th degree, or a cubic-spline surface).



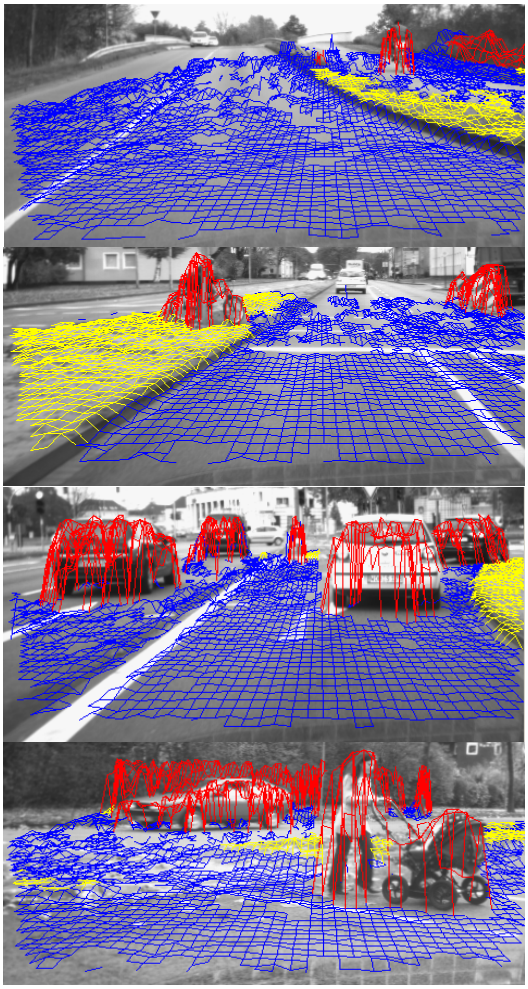


Fig. 12. Results for various scenes. The algorithm performs well even for scenes with sparse 3D road reconstruction and noisy 3d data (the grid is projected only where dense 3D data is provided by the dense stereo engine). The result for the first scene is displayed using a virtual camera. Grid colors represent blue - road, yellow - traffic isles, and red - obstacles.

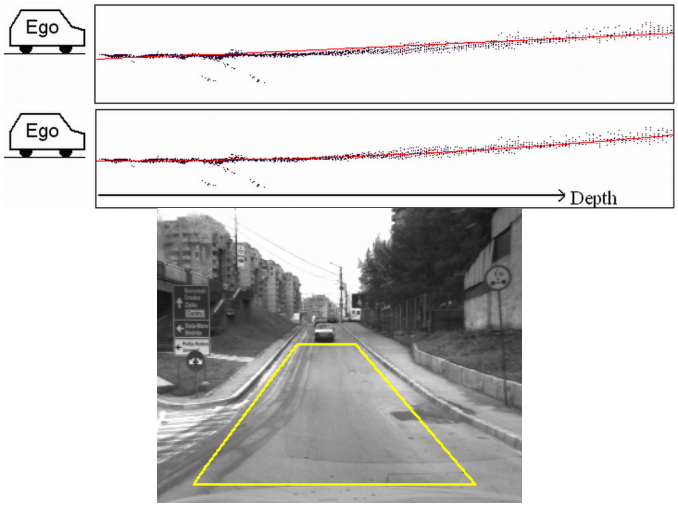


Fig. 13. The road surface is detected with the proposed algorithm. A quadratic model (middle) is more suitable than the planar model (top). The lateral view shows a 3D region with a depth range Z of 25 meters and Y from -1.5 to 1.5 meters. For better visualization, only 3D road points inside the yellow trapezoid (bottom image) are shown, and a cross-section (X=constant) of the surface is drawn.

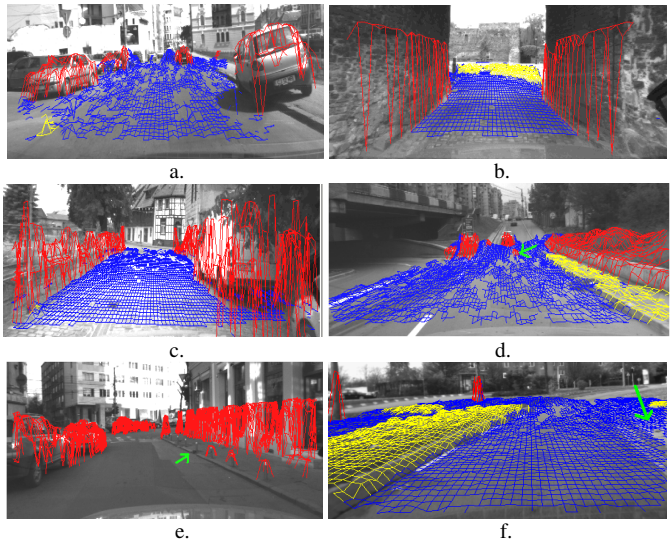


Fig. 14. a. False traffic isle appearing in the left-lower part of the image, b. Large false traffic isles are likely to appear if the road surface is not close to the quadratic road model, c. False obstacles appear on the right side due to floating tree branches, d. A small false object appears at far depths (34 meters, depicted with the green arrow), in a sharp uphill scene, e. Small traffic poles are sometimes not detected due to the lack of 3D data (usually this happens for mid- and far depths), f. Traffic isles with very small elevation from the road are classified as road inliers. The bus station shown by the arrow presents a 3 cm elevation at its border.

REFERENCES

[1] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection on non flat road geometry through V-disparity representation," in IEEE Intelligent Vehicles Symposium, Versailles, pp. 646–651, June 2002.

[2] A. Broggi, C. Caraffi, P. Paolo Porta, and P. Zani, "The Single Frame Stereo Vision System for Reliable Obstacle Detection used during the 2005 DARPA Grand Challenge on TerraMax", in IEEE Intelligent Transportation Systems Conference, Toronto, Canada, pp. 745-752, September 17-20, 2006.

[3] R. Labayrade and D. Aubert, "A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision," IEEE Intelligent Vehicles Symposium, Columbus, USA, pp. 31 – 36, June 2003.

[4] Z. Hu and K. Uchimura, "U-V-Disparity: An efficient algorithm for stereovision based scene analysis," in IEEE Intelligent Vehicles Symposium, Las Vegas, USA, pp. 48–54, June 2005.

[5] 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, Springer Verlag, Póvoa de Varzim, Portugal, pp. 205-216, September 18-20, 2006.

[6] M. Cech, W. Niem, S. Abraham, and C. Stiller, "Dynamic ego-pose estimation for driver assistance in urban environments", IEEE Intelligent Vehicles Symposium, Parma, Italy, pp. 43-48, 2004.

[7] 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", IEEE Intelligent Vehicles Symposium, Parma, Italy, pp. 292-297, 2004.

[8] S. Nedevschi, R..Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, "3D Lane Detection System Based on Stereovision", IEEE Intelligent Transportation Systems Conference (ITSC), Washington, USA, pp.161-166, 2004.

[9] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," Autonomous Robots, vol. 18, pp. 81–102, 2005.

[10] Z. Zhang, "A stereovision system for a planetary rover: Calibration, correlation, registration, and fusion," in IEEE Workshop on Planetary Rover Technology and Systems, April 1996.

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

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

[13] M. Fischler, R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", Graphics and Image Processing 24(6) (1981), pp. 381–395.

[14] J. Canny, "A computational approach to edge detection", IEEE Trans. Pattern Analysis and Machine Intelligence, pp. 679-698, Nov. 1983.

[15] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, 13(2):111-122, 1981.

[16] J. Woodfill, G. Gordon, D. Jurasek, T. Brown, and R. Buck, "The Tyzx DeepSea G2 Vision System, A Taskable, Embedded Stereo Camera", Conference on Computer Vision and Pattern Recognition, (New York, NY), June 2006.

[17] Gallo O., Manduchi R., and Rafii A., "Robust Curb and Ramp Detection for Safe Parking Using the Canesta ToF Camera". In IEEE Conference on Computer Vision & Pattern Recognition, Workshop on ToF-Camera based Computer Vision, June 2008.

[18] R. Turchetto, R. Manduchi, "Visual Curb Localization for Autonomous Navigation", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1336-1342, Las Vegas, October 2003.

[19] R. Aufrere, C. Mertz, and C. Thorpe, "Multiple sensor fusion for detecting location of curbs, walls, and barriers", IEEE Intelligent Vehicles Symposium, IV2003, pp. 126–131.

Florin Oniga received the Diplomat Engineer degree in computer science, from Technical University of Cluj-Napoca, Cluj-Napoca, Romania, in 2002. He is currently working toward the Ph.D. degree in computer science at Technical University of Cluj-Napoca, specializing in Computer Vision.

He is a Lecturer at Computer Science Department, Technical University of Cluj-Napoca, and he teaches image processing, pattern recognition, and computer architecture. His research interests include stereovision, digital elevation maps processing, and vision based automotive applications

Sergiu Nedevschi (M'99) received the M.S. degree in E.E. from Technical University of Cluj-Napoca, Romania, in 1975, and the Ph.D. degree in E.E. from the same university in 1993.

He worked as researcher at the Research Institute for Computer Technologies from Cluj-Napoca, between 1976-1983. In 1998 he was appointed Professor in computer science and he founded the Image Processing and Pattern Recognition Research Laboratory at the Technical University of Cluj-Napoca (TUCN). He was the head of Computer Science Department, between 2000-2004, and now is the dean of Faculty of Automation and Computer Science. His research interests include image processing, pattern recognition, computer vision, intelligent vehicles, signal processing and computer architecture. He has published nearly 200 technical papers and has edited over ten volumes including books and conference proceedings.