Real-Time Environment Representation based on Occupancy Grid Temporal Analysis using a Dense Stereo-Vision System

Andrei Vatavu, Sergiu Nedevschi, and Florin Oniga
Computer Science Department
Technical University of Cluj-Napoca,
Cluj-Napoca, Romania
{Andrei.Vatavu, Florin.Oniga}@cs.utcluj.ro

***Abstract*—We propose an environment representation technique by Temporal Analysis of the Occupancy Grid using a Dense Stereo-Vision System. The proposed method takes into account both the 3D information provided by the Occupancy Grid and the ego-car parameters. We use a method for computing the differences between the previous and current frames and compute an evidence space called Occupancy Grid Difference Map. Based on the difference map we created a reasoning component to generate an improved 2.5D model by representing the environment as a set of polylines with the associated static and dynamic features.**

Keywords-environment representation; stereovision; difference map; occupancy grid; temporal difference; poligonal model;

I. INTRODUCTION

One of the main challenges when working in the field of autonomous navigation is the digital environment representation [17]. The environment modeling process has to be accurate and characterized by a low computational cost. However, the performances achieved in complex dynamic environments such as crowded city traffic scenarios are still unsatisfactory. Therefore, an Advanced Driver Assistance Systems must include an environment representation component, able to advise the driver and provide appropriate information about both its static and dynamic environment, achieving a high level of accuracy, confidence, and real-time capability.

Usually, the Driving Assistance Applications detect the objects through 2D or 3D points grouping processes. The detected objects are represented by geometric primitives such as 2D bounding boxes [2] or 3D cuboids [3] [13]. As an alternative approach, the objects may be represented by polylines. One of the advantages of the polyline based objects representation is the close approximation of the object contour by the polygonal model while having a number of vertices as small as possible. In the same time the polyline could inherit the type, position, height properties, and dynamic features of the associated object.

The polyline object representation may lead to the creation of subsequent algorithms that are computationally fast due to the fact that only a small subset of points is employed.

The road feature identification through the object delimiters detection can be used in the unstructured environments as an alternative solution to the lane detection algorithms.

The object delimiters extraction is studied in some areas like mobile robots [4], [5], [6], [7], [8] or autonomous vehicle systems [9], [10], [11]. The polyline representation is very common in many algorithms, such as localization and mapping [6], [7], [8], [10] contour tracking [12] and path planning [10].

The polyline extraction methods differ by the nature of the information as well as by the sensors used for data acquisition process. Current systems use laser [4], [8], [9], [10], sonar [11], [7] or vision sensors [11].

A method for map representation as a set of line segments or polylines is described in [7]. An occupancy grid is created here from sonar information. The data is converted to a list of vertices using the Douglas Peucker line reduction algorithm.

In [8] a method that learns sets of polylines from laser range information is presented. The polylines are iteratively optimized using the Bayesian Information Criterion.

The polyline representation was chosen in [10] for terrain-aided localization of autonomous vehicle. The new range data obtained from the sensor are integrated into the polyline map by attaching line segments to the end of the polyline as the vehicle moves gradually along the tunnel.

In [16] a vehicle detection algorithm using laser range finders is described. The notion of motion evidence is presented which allows the dynamic obstacle detection.

In our previous work [1] we presented and evaluated several methods for real-time environment representation by extracting object delimiters from the traffic scenes using a Dense Stereovision System [3]. The delimiters detection was based on processing the information provided by a 3D classified occupancy grid obtained from the raw dense stereo information. Two approaches to extract object delimiters were presented: an improved contour tracing called 3A Tracing, and a polyline extraction method through the radial scanning of the occupancy grid called Border Scanning.

In this paper we extend our system from [1] with a moving obstacle detection and representation technique by Temporal Analysis of the Occupancy Grid using a Dense Stereo-Vision System. The proposed algorithm takes in account both the 3D information provided by the Occupancy Grid and ego-car parameters.

We use a method for computing the differences between the previous and current frames and generate an evidence space called Occupancy Grid Difference Map. A problem in using the Temporal Difference approach is that object forms are influenced by the occluded points as well as by the dynamic nature of traffic scenes. For this we developed a method that does not take into consideration the occluded points by exploring the Difference Map through the radial scanning.

Finally, we compute circular histograms from object difference fronts in order to calculate object resultant directions and to separate obstacles into static and dynamic obstacles. Kalman filtering is employed for tracking these features.

As the result an improved 2.5D model is computed by representing the environment as a set of polylines with the following associated static and dynamic features: a list of vertices, the delimiter type (object/curb, moving obstacle/static obstacle), the object height, Occupancy Grid blobs' features (center of mass, axis of elongation), and the dynamic features (a motion vector associated to the moving obstacle).

In the next section, we describe the proposed System Architecture. The Temporal Analysis of the Occupancy Grid technique is presented in section III. The last two sections show the experimental results and conclusion about the Environment Representation System we have developed.

II. System Architecture

Our Environment Representation System has been developed for an urban driving assistance system. We extended our Dense Stereo-Based Object Recognition System (DESBOR) by developing a representation component by Temporal Analysis of the Occupancy Grid. A detailed description about the DESBOR system is presented in [3].

The Environment Representation system architecture consists in the following modules (see Fig. 1):

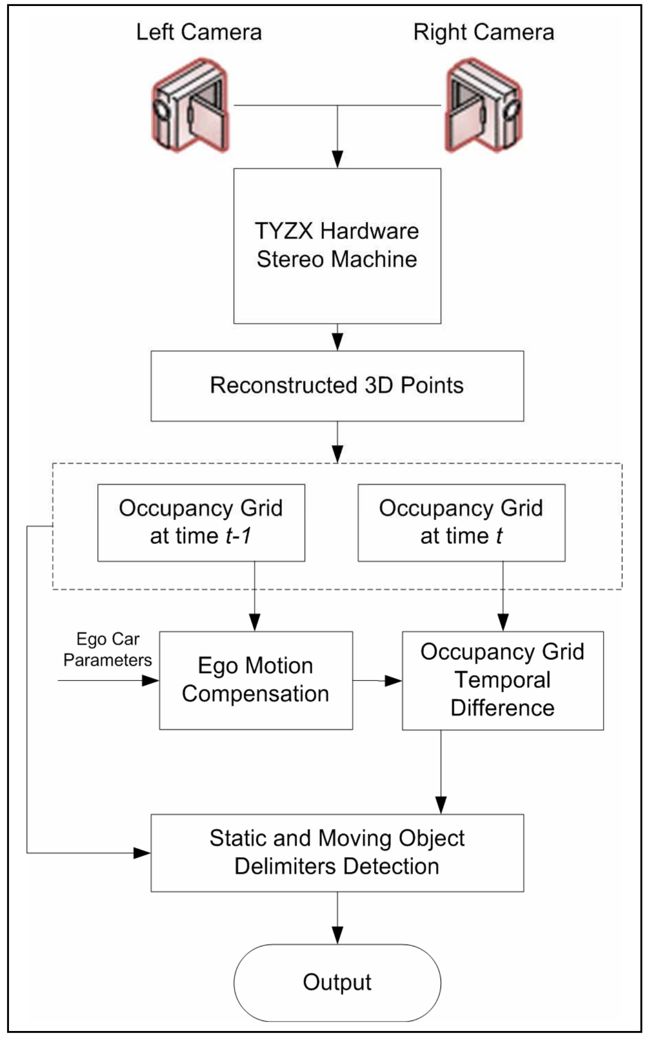

Figure 1. System Architecture

A. TYZX Hardware Stereo Machine

The 3D reconstruction is performed by hardware, a specialized PCI board ("TYZX") [14].

B. Reconstructed 3D Points

The reconstructed 3D points are used for the occupancy grid generation.

C. Occupancy Grid Computation

The occupancy grid (see Fig. 2.c) represents a description of the scene, computed from the raw dense stereo information represented as a digital elevation map (see Fig. 2.b). The occupancy grid cells are classified into road, traffic isle and object cells. A detailed description about the occupancy grid computation is presented in [15].

D. Ego Motion Compensation

At each frame we keep the Occupancy Grid generated at the previous frame. The Occupancy Grid coordinates from the previous frame are transformed to the current frame, assuming the ego motion parameters are known. By applying the ego motion compensation we ensure that the coordinate systems of the two occupancy grids are aligned.

E. Occupancy Grid Temporal Difference

We use an original method for calculating the differences between the two frames by radial scanning of the occupancy grids. Each cell from the current occupancy grid is associated with the corresponding cell from the previous frame. Depending on the presence or absence of a cell in the two

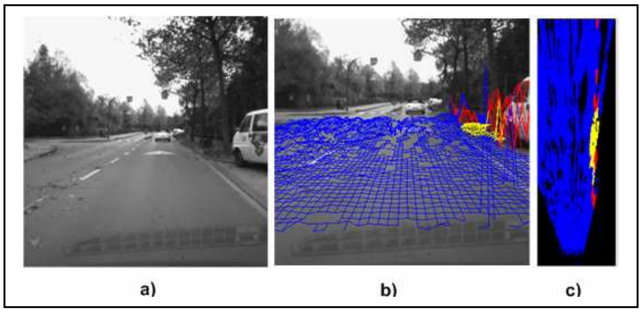

Figure 2. The Occupancy Grid (c) is computed from the Elevation Map (b) of a scene (a). The occupancy grid cells are roughly classified (blue – road, yellow – traffic isle, red – obstacles)

frames, we generate a specific label for it. As a result, we obtain so-called difference fronts that characterize moving objects between two frames. For each moving object, a motion vector is calculated by using the difference fronts as the relevant information. Each motion vector is described by its orientation and magnitude.

F. Object Delimiters detection

The Object Delimiters detection uses as the input the occupancy grid information and the results provided by de Temporal Difference module, generating a set of unstructured polygons approximated with the objects contour. Each polyline inherits from the associated objects the static as well as the dynamic features. For Delimiters Extraction we used the Border Scanning algorithm presented in [1].

G. Environment Representation Output

A polyline map is generated as the result of delimiters extraction and temporal difference analysis. For each polyline element we keep the following information:

- A set of vertices that describe the polygon.
- Object features: Center of Mass, Axis of Elongation, and Height.
- Type of the associated obstacle: Static or Moving obstacle, Object delimiter or Curb delimiter
- Dynamic Features: Orientation and Magnitude.

III. Temporal Analysis of the Occupancy Grid

We use the occupancy grid temporal analysis to keep an evidence of the occupancy grid cells and to detect differences both at the cell level, and the object level by computing a so called map of the difference. Based on the difference map we created a reasoning to detect static and dynamic obstacles features.

The Temporal Analysis of The Occupancy Grid (TAOG) approach consists in the following steps:

1. **Object Labeling.** In this step each object from the Occupancy Grid is labeled with a unique identifier. The object labels are subsequently used for object association in consecutive frames.
2. **Ego Motion Compensation.** The Ego Motion is compensated between successive frames in order to align the two Occupancy Grid coordinate systems. Coordinates from the previous frame are transformed straightforward into the current reference frame by a rotation and a translation assuming the ego car odometrical parameters are known.
3. **Difference Map Computation.** For each cell a corresponding flag is assigned (direction, shadow, and core) by comparing the previous and current occupancy grid.
4. **Circular Histogram Generation and Interpretation.** In this step we compute a circular histogram for each object from the Occupancy Grid by radial scanning of the Difference Map. Using circular histogram (also known as polar histogram) we generate a distribution model for moving object differences. We call these differences as front differences because usually they are clustered into a single front (Fig. 2.). The mean direction of an obstacle is estimated by using vector addition for all components accumulated into the histogram.
5. **Discrimination between static and dynamic obstacles.** For each moving object we compute a motion score knowing its Center of Mass Difference in successive frames and its Dominant Vector Magnitude.

Next we detail the main stages of the Occupancy Grid Temporal Analysis.

A. Ego Motion Compensation

To compensate the ego motion in the successive frames, for each given point $P_{t-1}(X_{t-1}, Y_{t-1}, Z_{t-1})$ in the previous frame the corresponding coordinates $P_t(X_t, Y_t, Z_t)$ in the current frame are computed by applying a rotation and a translation:

$$\begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix}^T = R_y(\psi)\begin{bmatrix} X_{t-1} \\ Y_{t-1} \\ Z_{t-1} \end{bmatrix}^T + \begin{bmatrix} 0 \\ 0 \\ T_z \end{bmatrix} \qquad (1)$$

Where:

$R_y(\psi)$ – is the rotation matrix around the Y axis with a given angle ψ.

T_z – is the point translation on the Z axis. It is considered that the translations on the X, Y axis are zero.

B. Difference Map Computation

For each cell in the previous frame we keep an evidence of its persistence at the corresponding position. Thus, based on the presence or absence of the cell in the current frame we build a

Difference Map that stores the point differences in the two frames.

A given cell in a certain position in the difference map may belong to the following categories:

Core cell – if the same cell is occupied in the both of the frames.

Direction cell – if a cell is empty in the previous frame, and occupied in the current frame.

Shadow cell – the cells that are occupied in the previous frame and are empty in the current frame at the corresponding position.

We use different flags for all of these categories.

There are cases when the same cell is occupied in successive frames by different objects (obstacle or traffic isle). This is due to the Occupancy Grid noise or to the dynamic nature of the environment. In this case we take into account only the persistence of the obstacles (not traffic isles). This can be explained by the fact that the traffic isles are static related to the road surface while we focus on the dynamic object features.

We must note that the obstacle association is limited by the obstacle velocity. The higher the object speed, the smaller the overlapped surface of an object in successive frames. However, the data acquisition is made at a frame rate of 20fps. Thus, for an 3 meters long and 2 meters wide obstacle, that has a velocity of 50km/h (generally this is the speed limit in urban areas), the overlapped surface in the consecutive frames is about 77%. This is more than enough for our approach, and even for higher speeds the overlapping is achieved.

After the computation of the difference map we observe that the dynamic obstacles are characterized by three types of areas (see Fig. 3): a direction front (the direction of the moving obstacles), a shadow front (usually located behind the moving obstacles), and a core area that remains unchanged in the consecutive frames.

Further, in this work we will use this information to compute the movement direction of the car through the creation and interpretation of circular histograms for each object.

C. *Circular Histogram Generation and Interpretation*

In this phase, for each object from the Temporal Difference Map, we calculate a dominant direction vector that is characterized by an orientation and a magnitude. The main idea consists in gathering a set of circular measurements for difference fronts and storing them into an angular histogram (also known as circular histogram or polar histogram). Thus, for a point situated at the object boundary we accumulate all the difference points (direction, and shadow) in a circular histogram by moving along a ray towards the object center of mass. The generated data sample in the histogram will have the same angle as the orientation of the processed ray.

The problem of this approach is that it's hard to keep evidence from the previous frame to the current one, because of the new information that influences the object shape. The object

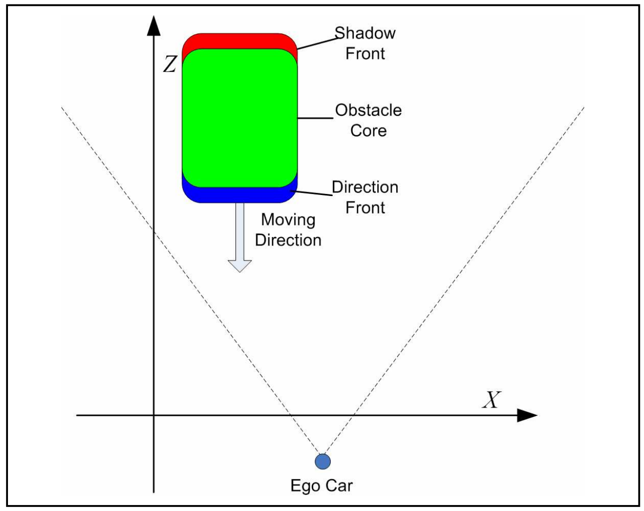

Figure 3. In a Difference Map, we have two types of difference fronts that characterize moving objects between two frames: in the direction front (with blue) the occupancy grid cells are present in the current frame but are empty in the previous frame, the shadow front flags the cells that are empty in the current frame but are filled in the previous one

forms are influenced by the occluded points as well as by the dynamic nature of traffic scenes.

For this we developed a technique that doesn't take into consideration occluded points by radial scanning of the Difference Map. The result of this scanning is a polygonal model of the environment with the obstacle dynamic features associated to it.

In this approach the scanning axis moves in the radial direction, having a fixed center at the Ego Car position (Fig. 4).

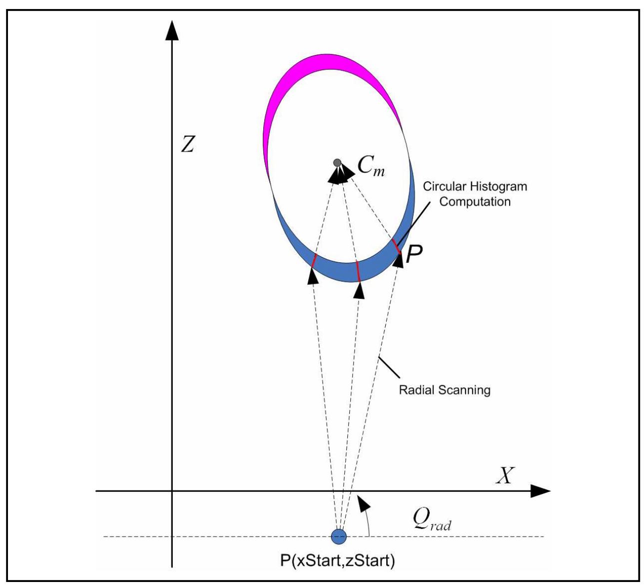

Figure 4. Accumulating the difference points that fall into the circular histogram associated to an object.

The scanning process is made within the limits of two given angles, thus only the interest area is scanned, where the object delimiters can be detected. Having a radial axis with a given slope we try to find the nearest Object point from the Ego Car situated on this axis. In this way, all subsequent points will be accumulated into a *Contour List*, moving the scanning axis in the radial direction. Once a point *P* of an object *O* is found (Fig. 4) we accumulate all the difference points situated on the axis that connect the point *P* and the center of mass C_m of the object *O*.

The accumulated points form a direction vector that is stored into a circular histogram associated to the object O.

The mean direction of an obstacle is estimated by using vector addition for all components accumulated into the histogram. Given a set of individual vectors with a direction θ_i and a magnitude M_i, The mean direction $\bar{\theta}$ is ocmputed by the following formula:

$$\bar{\theta} = atan2(R_x, R_z). \tag{2}$$

Where:

$$R_x = \sum_{i=1}^{n} M_i \cos(\theta_i),\ R_z = \sum_{i=1}^{n} M_i \sin(\theta_i). \tag{3}$$

$$R^2 = R_x^2 + R_z^2. \tag{4}$$

The magnitude of the mean vector is defined by:

$$\|\vec{r}\| = \frac{1}{n}\sqrt{R_x^2 + R_z^2}. \tag{5}$$

Where n is the number of individual vectors from the histogram. In our experiments we use R (equation 4) as the magnitude of the mean vector.

Fig.5.b shows the circular histograms (colored with orange) and the corresponding mean directions (colored with red) associated to the moving obstacles.

D. Obstacle tracking

Each obstacle (non traffic isle) is tracked in order to compute its filtered position and speed, which will allow further the discrimination between static and dynamic obstacles.

A standard Kalman filter is employed in order to estimate the state of each obstacle:

$$x = \begin{bmatrix} X_m \\ Z_m \\ R_x \\ R_z \end{bmatrix}. \tag{6}$$

where:

X_m, Z_m are the coordinates of the obstacle mass center C_m,

$\vec{r} = (R_x, R_z)$ – the mean direction vector, which actually encodes the speed and the direction of movement.

For each new obstacle (that has no associated obstacles in previous frames) a new tracker is initialized. If the obstacle has an associated obstacle in the previous frame, then the associated tracker is updated with the current frame measurements and the state of the obstacle is estimated based on the current measurements. The measurement covariance matrix R is computed based on the stereo uncertainty model described in [15]. The localization uncertainty is estimated considering the stereo system parameters and a disparity uncertainty of a quarter of a pixel. The process covariance matrix Q is estimated considering a certain covariance for the obstacles' acceleration.

IV. Experimental Results

For the experimental results we tested a set of 10 scenarios form crowded urban environments using a 2.66GHz Intel Core 2 Duo Computer with 2GB of RAM.

Fig. 6 shows the Temporal Difference results (see Fig. 6.e) for the traffic scene from Fig. 6.a. The Occupancy Grid (see Fig. 6.b) is computed from scene (a). The temporal difference (Fig. 6.e) uses the previous occupancy grid (Fig. 6.c) corrected with the ego motion, and the current occupancy grid (Fig. 6.d). The Temporal Difference Maps Cells are classified in shadow cells (magenta), direction cells (blue), traffic isles core cells (yellow), and obstacles core cells (light green)

It can be observed in the Temporal Difference Map (Fig. 6.e) that despite the static nature of the traffic isles we obtain a shadow front at the Difference Map extremities. This is due the fact that after compensating the previous frame with the ego motion we obtain cells that have no correspondence in the previous frame. Therefore, shadow fronts are formed at the bottom of the Difference Map while direction fronts are formed at the top of the map. As a solution, we can use a Region of Interest for the processing algorithms.

The environment representation accuracy depends on the 3D reconstruction accuracy as well as on the results provided by the Occupancy Grid

Fig. 7 Shows the Environment Representation result in Urban Driving Scenarios.

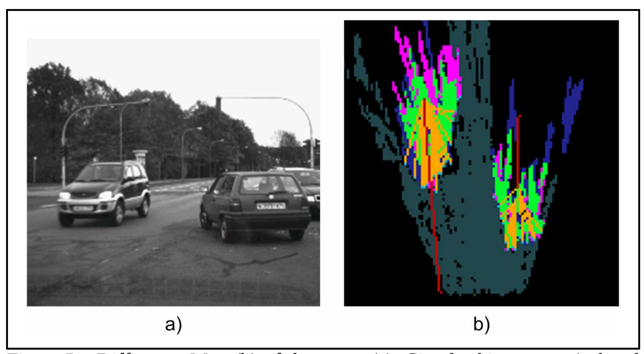

Figure 5. Difference Map (b) of the scene (a). Circular histograms (colored with orange) are computed. The mean directions (colored with red) are associated to the corresponding moving obstacles.

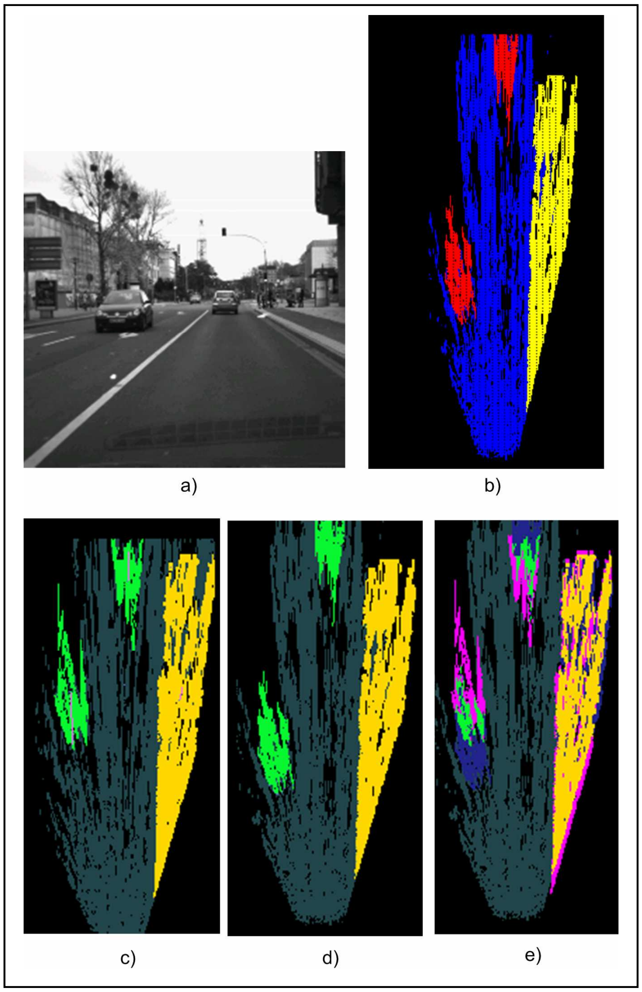

Figure 6. Temporal Difference. The Occupancy Grid (b) is computed from scene (a). The temporal difference (e) uses the previous occupancy grid (c) corrected with the ego motion, and the current occupancy grid (d). The Temporal Difference Maps Cells are classified in shadow cells (magenta), direction cells (blue), traffic isles core cells (yellow), and obstacles core cells (light green)

Difference maps are presented on the left side. The detected Object Delimiters are shown in the Virtual View of the Scene (middle) and are projected onto the Left Camera Image (right side). The Object Delimiters are represented as grids labeled as Traffic Isles (orange), Static Objects (light green), and Dynamic Objects (red). It can be observed that the moving obstacles have an associated direction represented as an orange line.

V. CONCLUSIONS

In this paper we develop a new technique for dynamic obstacles detection and representation using a Dense Stereovision System.

The proposed algorithm takes into account both the 3D information provided by the Occupancy Grid and the ego-car parameters.

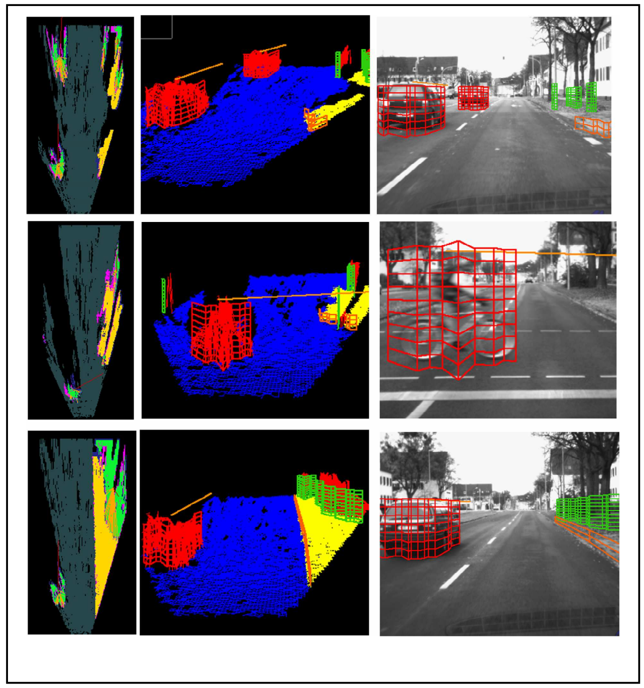

Figure 7. Dynamic Environment Representation.

We use a method for computing the differences between the two frames and generate an evidence space called Occupancy Grid Difference Map. A problem in using the Temporal Difference approach is that object forms are influenced by the occluded points as well as by the dynamic nature of traffic scenes. For this we developed a technique that doesn't take into consideration occluded points by radial scanning of the Difference Map.

We extend our previous Border Scanning algorithm [1] by detecting moving obstacles difference fronts and collecting them into circular histograms, associated to each object. We use the histogram circular measurements to compute a resultant direction vector and to separate obstacles into static and dynamic obstacles.

As the result an improved 2.5D model is computed by representing the environment as a set of polylines with the associated static and dynamic features such as obstacle direction characterized by an orientation and a magnitude, type of the associated obstacle (Static or Moving obstacle, Object, Curb), a list of vertices, obstacle height etc.

As future work we propose to focus our research in improving the accuracy of the Environment Representation.

REFERENCES

[1] Andrei Vatavu, Sergiu Nedevschi, Florin Oniga, "Real Time Object Delimiters Extraction for Environment Representation in Driving Scenarios". In: ICINCO-RA 2009, Milano, Italy, 2009, pp 86-93.

[2] Dellaert, F., Thorpe, C.: Robust Car Tracking using Kalman filtering and Bayesian templates. In: proceedings of Conference on Intelligent Transportation Systems, vol. 3207, pp. 72--83. (1997)

[3] Nedevschi, S., Danescu, R., Marita, T., Oniga, F., Pocol, C., Sobol, S., Tomiuc, C. Vancea, C., Meinecke, M. M., Graf, T., To, T. B., Obojski, M. A.: A sensor for urban driving assistance systems based on dense stereovision. In: Intelligent Vehicles 2007, pp. 278--286, Istanbul (2007)

[4] Harati, A., Siegwart, R.: A new approach to segmentation of 2d range scans into linear regions. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA (2007)

[5] Magin, G., Russ, A.: Supporting real-time update of an environment representation for autonomous mobile robots real-time systems. In: EuroMicro Workshop on Real-Time Systems, pp. 34--39 (1994)

[6] Joshi, R. R.: Novel metrics for map-matching in in-vehicle navigation system. In: IEEE Intelligent Vehicle Symposium, Vol. 1, pp. 36--43 (2002)

[7] Laviers, K. R., Peterson, G. L.: Cognitive robot mapping with polylines and an absolute space representation. In: IEEE International Conference on Robotics and Automation, pp. 3771--3776. Hilton New Orleans Riverside, New Orleans, LA, USA (2004)

[8] Veeck M., Burgard, W.: Learning polyline maps from range scan data acquired with mobile robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2004)

[9] Kolski, S., Ferguson, D., Bellino, M., Siegwart, R.: Autonomous driving in structured and unstructured environments. In: IEEE Intelligent Vehicles Symposium (2006)

[10] Madhavan, R.: Terrain aided localization of autonomous vehicles. In: Symposium on Automation and Robotics in Construction, Gaithersburg (2002)

[11] Goncalves, A., Godinho, A., Sequeira, J.: Lowcost sensing for autonomous car driving in highways. In: ICINCO2007 - 4th International Conference on Informatics in Control, Automation and Robotics, Angers, France (2007)

[12] Prakash, S., Thomas, S.: Contour tracking with condensation/stochastic search. In: Dept. of CSE. IIT Kanpur (2007)

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

[14] Woodill, J. I., Gordon, G., Buck, R.: Tyzx deepsea high speed stereo vision system. In: IEEE Computer Society Workshop on Real Time 3-D Sensors and Their Use, Conference on Computer Vision and Pattern Recognition (2004).

[15] Oniga, F., Nedevschi, S., Meinecke, M. M., To, T. B.: Road surface and obstacle detection based on elevation maps from dense stereo. In: IEEE Intelligent Transportation Systems Conference, pp. 859--865, Seattle, WA, (2007).

[16] Petrovskaya, A., Thrun, S. Efficient techniques for dynamic vehicle detection. In ISER, Athens, Greece (2008).

[17] Pijpers, M.: Sensors in adas. Universiteit Twente (2007).