Real-Time Dynamic Environment Perception in Driving Scenarios using Difference Fronts

Andrei Vatavu, Radu Danescu, and Sergiu Nedevschi, *Member, IEEE*

Abstract— **The environment representation is one of the main challenges of autonomous navigation. In the case of complex driving environments such as crowded city traffic scenarios, achieving satisfactory results becomes even more difficult. In this paper we propose a real-time solution for two main issues of advanced driver assistance systems: unstructured environment representation and extraction of dynamic properties of traffic participants. For the real-time environment representation we propose a solution to extract object delimiters from the traffic scenes and represent them as polygonal models. In order to track dynamic entities, an intermediate evidence map named "Stereo Temporal Difference Map" is proposed. This difference map is computed by comparing the occupancy of a cell between two consecutive frames. Based on the Stereo Temporal Difference Map information, difference fronts are extracted and are subjected to a particle based filtering mechanism. Finally, the provided dynamic features are associated to the extracted polygonal models. The result is a more compact representation of the dynamic environment.**

I. INTRODUCTION

In the context of Advanced Driver Assistance Systems, the perception of dynamic environments is still an open problem. In order to represent the knowledge about other moving traffic participants, first we have to choose adequate models that accurately describe dynamic evolution in time, and also their geometrical shapes. In the case of the most complex driving environments such as crowded city traffic scenarios, acquirement of satisfactory results becomes even more difficult. A driver assistance system should be able to provide a digital model of the surrounding world in real-time, and with a high accuracy and robustness. Also, the resulted representation should permit fast subsequent processing tasks.

In the case of stereovision systems, which rely on passive sensors, the motion information cannot be provided directly. A common approach for tracking solutions consists in extracting desired features and estimating their motion over time. Current solutions can be classified based on the level at which the tracking and representation is performed. In a simpler and clearly structured environment, the obstacles are usually modeled as 2D bounding boxes or 3D cuboids, and are described by their position, size and speed.

Manuscript received January 31, 2012.

Andrei Vatavu, Radu Danescu and Sergiu Nedevschi are with the Technical University of Cluj-Napoca, Computer Science Department (e-mail: {firstname.lastname}@cs.utcluj.ro). Department address: Computer Science Department, Str. Memorandumului nr. 28, Cluj-Napoca, Romania. Phone: +40 264 401484.

Model-based tracking may not be the best solution when the driving environment is more complex, as in the case of an off-road scenario, an intersection, or a crowded urban center. Even if parts of this environment can be tracked by estimating the parameters of the cuboidal model, many essential parts of the environment may not fulfill the constraints of the models. An occluded object, or an object that changes its size or shape will mislead the model-based tracking system, consequently the correct speed estimation will be impossible to achieve. For this reason, any perception system can be improved by estimating dynamic properties of the environment independently from the choice of object representation.

In order to achieve the goal of extracting the speed independently from object model, intermediate tracking solutions are devised. Such solutions can directly track 3D points (the 6D vision technique, presented in [1]), compact dynamic obstacle primitives such as the stixels [2], or they can use track the occupancy and speed of a cell in the map, such as in the case of occupancy grids.

The occupancy grid is a good option for traffic environments, as it is able of concisely describing the relevant features from the scene while maintaining a decent level of computation complexity. One of the first uses of occupancy grids is presented by Elfes in [3], in the context of sonar based robot navigation. A probability inference mechanism for handling the uncertainty of range sensors in computing the occupancy probability of each cell is presented in [4]. The initial occupancy grids, such as those described in [3] and [4], are simple 2D maps of the environment, each cell describing the probability of it being occupied or free. By adding the speed component in the environment estimation, the complexity increases significantly. A 4D occupancy grid, where each cell is described by a position and two speed components is presented by Coué et al in [5]. Another approach for the representation of speeds is described in [6] by Chen et al. Instead of having a 4D grid, this solution uses a distribution of speeds in the form of a histogram for each cell.

In order to provide a mean of identifying individual dynamic objects, we require a way to extract freeform models from the scene. A proper approach towards this goal is the extraction of polygonal models. This representation solution has the advantage of closely approximating the object contour by the polygonal model while having a number of vertices as small as possible, and it also includes the static and dynamic features from the associated objects. 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 [7], [8], sonar [9], or vision sensors [9]. The polyline representation was chosen in [8] 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.

This paper presents a novel approach for real-time environment representation and tracking based on a dense stereo-vision system. Two main issues of advanced driver assistance systems are addressed: unstructured environment representation, and extraction of dynamic properties of traffic participants. The proposed methods take into consideration the 3D information provided by a Digital Elevation Map, as well as the ego-car parameters such as yaw rate and car speed.

For a real-time environment representation we developed a method that extracts free form object delimiters from the traffic scenes by radial scanning of the Elevation Map.

In order to track dynamic entities, an intermediate evidence space is generated by computing differences between the two consecutive elevation map representations at different moments of time. This evidence space will be called "Stereo Temporal Difference Map"; for simplicity, in the remainder of this paper we'll refer to it as Difference Map. Next, we use a probabilistic approach for modeling the extracted difference fronts by using particles that move from cell to cell and are created and destroyed based on new measurements provided by the Difference Map at each frame. Thus, instead of directly tracking all traffic entities, we focus only on analyzing and tracking the differences between two consecutive scenes without making assumptions about object shape or size. Finally, the provided dynamic features are associated to the extracted polygonal models. The result is a 2.5D compact representation of the dynamic environment.

In the next section, we describe the proposed system architecture. Section 3 presents the dynamic environment representation solution based on polylines, Difference Map representation and particle based modeling of difference fronts. The last two sections show the experimental results and the conclusion of this contribution.

II. System Architecture

Our method has been conceived and adapted for crowded unstructured environments such as urban city traffic scenes. The previously developed Dense Stereo-Based Object Recognition System (DESBOR) has been improved by including additional processing modules for Difference Map extraction and Particle Based Difference Fronts modeling. An overview about about the DESBOR system is presented in [10]. The Dynamic Environment Perception system consists in the following main modules (see figure 1):

Reconstructed 3D Points: the 3D reconstruction is performed in real time using a dense stereo algorithm implemented on a GPU board [11]. The reconstructed 3D points are used as primary information for computing the Digital Elevation Map.

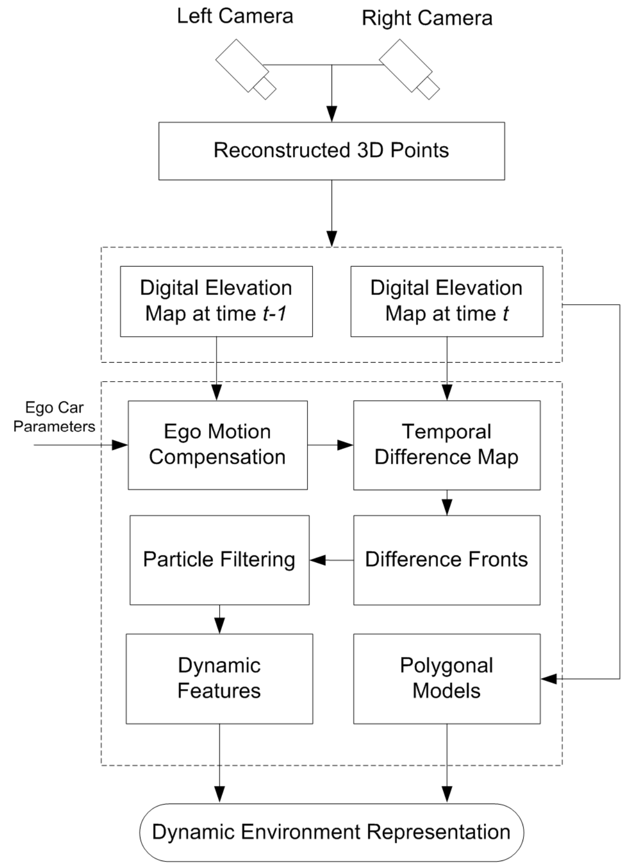

Figure 1. System Architecture.

Digital Elevation Map: the Elevation Map (see figure 2) represents an intermediary description of the scene and is computed from the raw dense stereo information. The Elevation Map contains three types of cells: road, traffic isle and object. More details about the Elevation Map are presented in [12].

Ego Motion Compensation: the Elevation Map's coordinates from the previous frame are transformed to the current frame, assuming that we know the ego car parameters. By compensating the ego motion we ensure that the two Elevation Map coordinate systems are aligned.

Stereo Temporal Difference Map: an evidence map is computed by comparing the presence or absence of an Elevation Map cell at different moments of time. This process classifies each Difference Map cell as direction, shadow, or core cell.

Difference Fronts: after the computation of the Difference Map, we define three types of areas for the moving objects: a direction front (the direction of the moving obstacle), a shadow front (usually located behind the moving obstacle), and a core area that remains unchanged in the consecutive frames.

Particle Based Filtering: the extracted difference fronts are subjected to the particle based filtering. As the result a dynamic grid based on particles is produced. Each particle has a position and speed, and can migrate in the grid from cell to cell depending on its motion model and motion parameters. Grid particles are also created and destroyed using a weighting-resampling mechanism. We extend the previously developed algorithm [13][14], by using the difference fronts as measurement information.

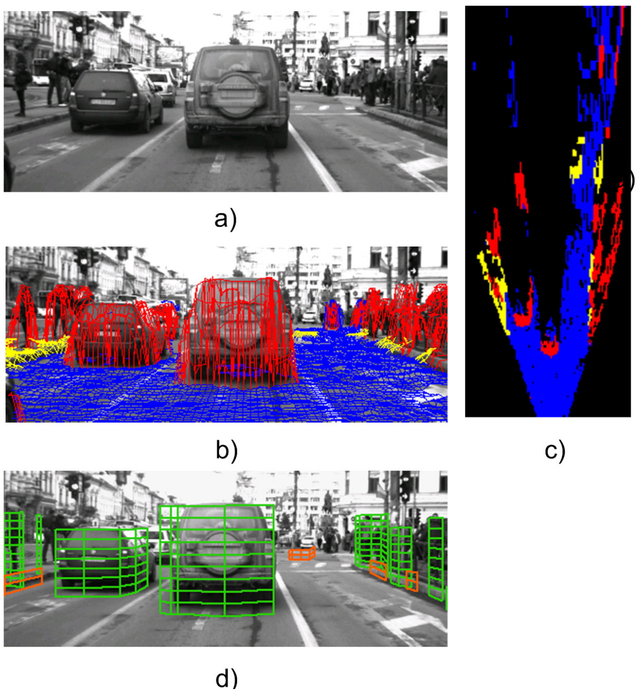

Figure 2. a) A traffic scene. b) The Elevation Map is projected on the left camera image. c) Elevation Map, top view. The Elevation Map cells are classified (blue – road, yellow – traffic isle, red – obstacles). d) Polyline Based Environment Representation. The object types are inherited from the Elevation Map information (green – obstacles, yellow – traffic isles).

Polygonal Models: the obstacles delimiters are extracted by radial scanning of the Elevation Map, and the more compact polygonal model is generated. For delimiter extraction we use the Border Scanning algorithm presented in [15]

Environment Representation Output: speed vectors, computed by the particle filtering step are associated to the static polygonal models. A dynamic polyline map is generated as the result. Each polyline element is characterized by a set of vertices, position, height, type (traffic isle, obstacle), orientation and magnitude.

III. Dynamic Environment Representation

In this section we present the main stages of the dynamic environment representation process. Most of the representation and tracking solutions in the literature rely on extracting an object model and subsequently inferring its motion over time. In our work we handle the unstructured environment representation problem and the motion estimation problem by independent modules. Thus, we avoid some additional intermediate processing steps for both cases. This approach allows us to extract dynamic features (speed vectors) regardless of the model chosen to represent the surrounding world and vice versa. Next, we describe each step of the proposed approach:

A. Polyline Based Environment Representation

For the polyline based object representation, we extend the Border Scanner algorithm described in [15]. The main idea is that we are taking into account only the most relevant scene information, by extracting object delimiters by radial scanning of the Elevation Map. Our method is based on a Ray-Casting approach, which determines the first occupied cell that intersects a virtual ray which is cast from the ego-car's reference frame origin. At each step we try to find the nearest visible point situated on the scanning ray. In this way, all subsequent cells P_i are accumulated into a Contour List C, as the scanning ray's angle changes:

$$C = \{P_1, P_2, \ldots P_n) \quad (1)$$

For each object O_i described by a contour C_i we apply a polygonal approximation of C_i, using a split-and-merge technique. The extracted polygon is used to build a compact 3D model based on the polyline set of vertices as well as on the object height. An example of the polyline representation is shown in figure 2.d.

B. Stereo Temporal Difference Map

We analyze the classified obstacle cells of the Elevation Map, in order to detect differences both at cell level, and at object level. The outcome of this analysis is the Difference Map.

Before applying any reasoning about objects' state at different frames, the movement of the ego vehicle must also be taken into consideration. In order to compensate for the ego motion in 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:

$$\begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix}^T = R_y(\psi)\begin{bmatrix} X_{t-1} \\ Y_{t-1} \\ Z_{t-1} \end{bmatrix}^T + \begin{bmatrix} 0 \\ 0 \\ T_z \end{bmatrix} \quad (2)$$

Where $R_y(\psi)$ is the rotation matrix around the Y axis with a given angle ψ, and T_z – is the translation on the Z axis. We assume that the translations on the X and Y axis are zero.

For each cell in the previous frame we keep an evidence of its persistence at the corresponding position in the current frame. Thus, based on the presence or absence of the cell in the current frame, a Difference Map that stores the point differences between the two frames is built. We define three classes of cells (see figure 3):

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.

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

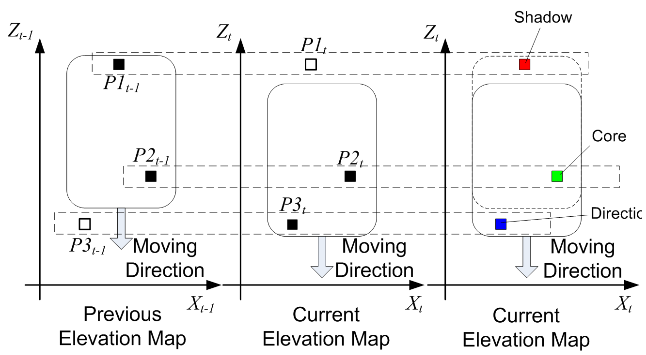

Figure 3. Difference Map Cells are classified: direction – blue, shadow – red, core - green.

C. Difference Fronts

After computing the Difference Map we define three types of areas that describe the moving obstacles (see figure 4): 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.

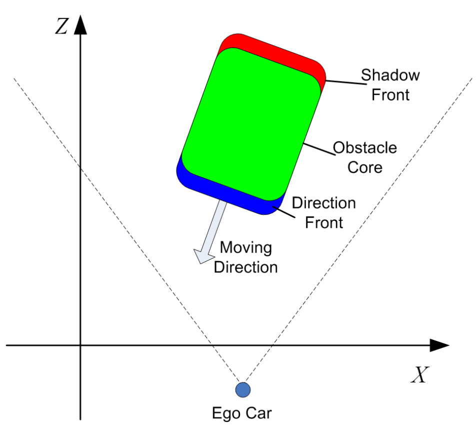

Figure 4. Difference Fronts: Direction Front – blue, Shadow Front – red and Obstacle Core - Green.

D. Particle Based Filtering

In this step we use a particle-based filtering mechanism in order to estimate the difference fronts speed components. We use a probability model to produce a fully dynamic grid based on particles. We consider that each grid cell has a population of particles that have a dual nature: they describe occupancy hypotheses, as in the particle filtering algorithms such as CONDENSATION [16], but can also be regarded as physical building blocks of our modeled world. The particles have position and speed, and they can migrate from cell to cell depending on their motion model and motion parameters, but they are also created and destroyed using the same logic as the weighting-resampling mechanism

Considering a coordinate system where the z axis points towards the direction of the ego-vehicle, and the x axis points to the right, the obstacles in the world model are represented by a set of particles:

$$S = \{ p_i \mid p_i = (c_i, r_i, vc_i, vr_i, a_i), i = 1 \ldots N_S \} \quad (3)$$

Each particle i has a position in the grid, described by the row r_i and the column c_i, and a speed, described by the speed components vc_i and vr_i. An additional parameter, a_i, describes the age of the particle, since its creation. The purpose of this parameter is to facilitate the validation and the speed estimation process, as only particles that survive in the field for several frames are taken into consideration. The total number of particles in the scene N_S dependent on the occupancy degree of the scene, that is, the number of obstacle cells in the real world. Having the population of particles in place, the occupancy probability of a cell C is estimated as the ratio between the number of particles whose position coincides with the position of the cell C and the total number of particles allowed for a single cell, N_C.

$$P_O(C) = \frac{|\{ p_i \in S \mid r_i = r_c, c_i = c_c \}|}{N_C} \quad (4)$$

The number of allowed particles per cell N_C is a constant of the system. In setting its value, a tradeoff between accuracy and time performance should be considered. The total number of particles in the scene will be directly proportional with N_C, and therefore the speed of the algorithm will be directly affected by its value.

The speed of a grid cell can be estimated as the average speed of its associated particles, if we assume that only one obstacle is present in that cell.

$$(vc_C, vr_C) = \frac{\sum_{p_i \in S, x_i = x_c, z_i = z_c} (vc_i, vr_i)}{|\{ p_i \in S \mid r_i = r_c, c_i = c_c \}|} \quad (5)$$

Multiple speed hypotheses can be maintained simultaneously for a single cell, and the occupancy uncertainty is represented by the varying number of particles associated to the cell. The tracking algorithm can now be defined: using the measurement information in the form of elevation maps, it will create, update and destroy particles such that they accurately represent the real world.

The first step of the algorithm is the *prediction*, which is applied to each particle in the set. The positions of the particles are altered according to their speed, and to the motion parameters of the ego vehicle. Also, a random amount is added to the position and speed of each particle, for the effect of stochastic diffusion. The second step is the *processing of measurement* information. The raw measurement data is derived from difference fronts.

The measurement model information is used to *weight* the particles, and *resample* them in the same step. By weighting and resampling, the particles in a cell can be multiplied or reduced. The final step is to estimate the occupancy and speeds for each cell. A more detailed description of the particle grid tracking algorithm is given in [13] and [14].

E. Data Association

This stage consists in assigning the speed vectors derived from the particle-based filtering to the polygonal models extracted from the Elevation Map. As each polygonal model directly inherits the object position and type, the problem is reduced to associating the tracked direction fronts to the Difference Map measurements. For each direction Front F_j in the occupancy grid space and for each entity L_i in the Difference Map we calculate an overlapping score C_{ij}. The results are stored into a score matrix $C=\{C_{ij}\}$. Candidates with the highest score are taken into account in determining the associations between the two sets F and L.

IV. Experimental Results

The proposed representation and tracking technique has been tested in real traffic situations. For a more complete evaluation we have compared the obtained results with the Kalman filter-based, cuboidal model oriented tracking method presented in [17]. Figure 5 describes the dynamic environment representation steps, including the intermediate results. The Difference Map (figure 5.d) is obtained based on the Elevation Map results at different times (figure 5.a and b). The Difference Cells are classified as direction (blue), shadow (magenta), and core (light green). Figure 5.e shows the particle based occupancy grid obtained from Difference Map measurements. The extracted dynamic polylines and the associated speed vectors (yellow color) can be seen on the top view of the Elevation Map (figure 5.f). Figure 5.g shows the projection of the static (green color) and dynamic obstacles (red color) on the left camera image.

For the numerical evaluation we have included the following traffic scenarios: an incoming vehicle and a stationary lateral vehicle. The obtained speeds are compared to the speeds obtained with the Kalman filter-based tracking approach.

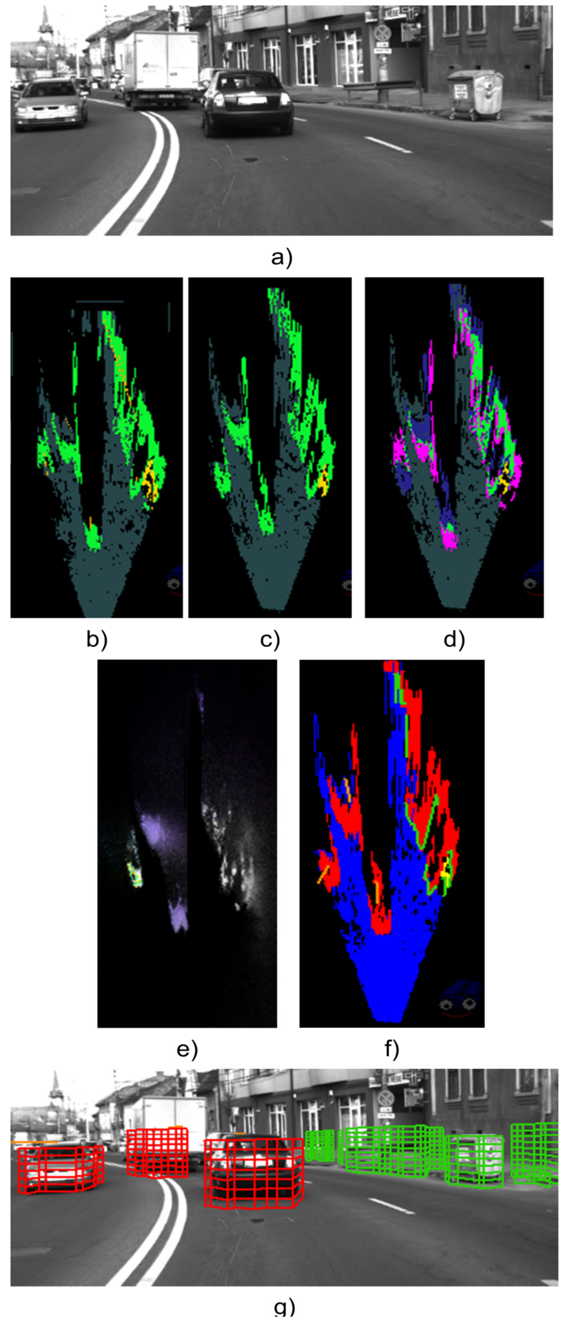

Figure 5. Dynamic environment representation with the intermediate stages. The Difference Map (d) is obtained based on the Elevation Map from previous frame (b) and current frame (c). The Difference Cells are classified as direction (blue), direction (magenta), and core (light green). The particle based occupancy (e) is grid obtained from Difference Map measurements. The extracted dynamic polylines and the associated speed vectors (yellow color) are shown on the top view of the Elevation Map (f). The static (green color) and dynamic obstacles (red color) are projected on the left camera image (g).

For the first test we have chosen a scenario with an incoming vehicle. The speed estimation values are shown in the figure 6. It can be observed that for this case, the values obtained by the particle filtering based technique (blue color) are close to the one obtained by a model based tracking method (magenta color).

The second test includes a stationary lateral vehicle (figure 7). The target speed of 0 is given as the ground truth for the measurements. The difference fronts tracking approach (blue color) proves to be more accurate having a lower mean absolute error (2.18 Km/h) than the Kalman filter cuboid-based tracking solution (7.5 Km/h) drawn with magenta color.

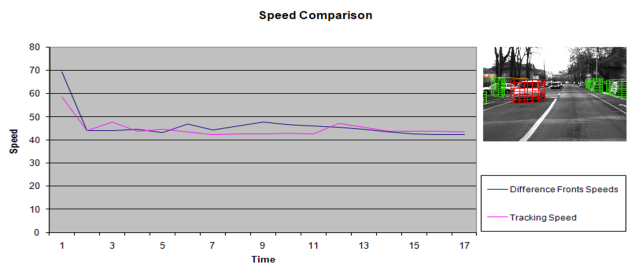

Figure 6. Speed Estimation for an incoming vehicle (green color). The particle based filtering of difference front method (blue color) is compared with a Kalman filter tracking solution (magenta color).

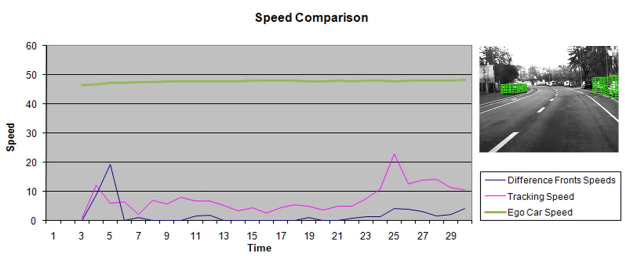

Figure 7. Speed Estimation for a stationary lateral vehicle (green color). The results are estimated with particle based tracking of difference fronts (blue color), Kalman filter tracking (magenta color). The ego car speed is colored with green.

V. Conclusions

In this paper a novel approach for stereo-based real-time environment representation and tracking is presented. Two main issues of advanced driver assistance systems are addressed: unstructured environment representation, and extraction of dynamic properties of traffic participants. For implementing our algorithms we use, as primary information, the Digital Elevation Map representation.

For the real-time environment representation the proposed solution extracts object delimiters from the traffic scenes, by radial scanning of the Elevation Map. In order to track dynamic entities, an intermediate evidence space is generated by computing differences between the two consecutive Elevation Map representations. We named this space the Stereo Temporal Difference Map. Further, we use a probabilistic approach for modeling the extracted difference fronts by using particles that move from cell to cell and are created and destroyed based on new measurements provided by the Difference Map at each frame. Instead of directly tracking all traffic entities, we focus only on the analysis and tracking of the differences between two consecutive scenes, without making assumptions about object shape or size. Finally, the provided dynamic features are associated to the extracted polygonal models. The result is a 2.5D compact representation of the dynamic environment. According to the experimental results the presented method achieves a high degree of accuracy.

As future work we propose to focus our research in extending the concept of “Stereo Temporal Difference Map” by computing the evidence of the traffic scene over multiple frames.

References

[1] U. Franke, C. Rabe, H. Badino, and S. Gehrig, “6d-vision: Fusion of stereo and motion for robust environment perception,” in proc of *27th Annual Meeting of the German Association for Pattern Recognition DAGM ’05*, Vienna, October, 2005.

[2] D. Pfeiffer, U. Franke, "Efficient Representation of Traffic Scenes by Means of Dynamic Stixels", *IEEE Intelligent Vehicles Symposium (IEEE-IV)*, 2010, pp. 217-224.

[3] A. Elfes, “A Sonar-Based Mapping and Navigation System”, in proc of *IEEE International Conference on Robotics and Automation*, April 1986, pp. 1151-1156.

[4] A. Elfes, “Using Occupancy Grids for Mobile Robot Perception and Navigation”, *Computer*, vol. 22, No. 6, June 1989, pp. 46-57.

[5] C. Coue, C.Pradalier, C.Laugier, T.Fraichard, P.Bessiere, “Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application”, *The International Journal of Robotics Research*, 25(1):19, 2006.

[6] C. Chen, C. Tay, K. Mekhnacha, C. Laugier, “Dynamic environment modeling with gridmap: a multiple-object tracking application”, in proc of *International Conference on Automation, Robotics and Computer Vision (ICARCV)* 2006, pp. 1-6.

[7] S. Kolski, D. Ferguson, M. Bellino, R. Siegwart, "Autonomous driving in structured and unstructured environments", in proc of *IEEE Intelligent Vehicles Symposium (IEEE-IV)*, 2006, pp. 558–563.

[8] R. Madhavan, "Terrain aided localization of autonomous vehicles", in proc of *Symposium on Automation and Robotics in Construction*, Gaithersburg, 2002, pp. 511-518.

[9] A. Goncalves, A. Godinho, J. Sequeira, "Lowcost sensing for autonomous car driving in highways", in proc of *ICINCO2007 - 4th International Conference on Informatics in Control, Automation and Robotics*, Angers, France, 2007, pp. 370-377.

[10] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, M. A. Obojski, ”A sensor for urban driving assistance systems based on dense stereovision”, in proc of *IEEE Intelligent Vehicles Symposium (IEEE-IV)* 2007, pp. 278--286, Istanbul.

[11] I. Haller, C. Pantilie, F. Oniga, S. Nedevschi, “Real-time semi-global dense stereo solution with improved sub-pixel accuracy”, in proc of *IEEE Intelligent Vehicles Symposium 2010*, pp. 369-376.

[12] F. Oniga, S. Nedevschi, “Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection”, *IEEE Transactions on Vehicular Technology*, Vol. 59, No. 3, March 2010, pp. 1172-1182.

[13] R. Danescu, F. Oniga, S. Nedevschi, “Particle Grid Tracking System for Stereovision Based Environment Perception”, in proc of *IEEE Intelligent Vehicles Symposium (IEEE-IV),* 2010, pp. 987-992.

[14] R. Danescu, F. Oniga, S. Nedevschi, “Modeling and Tracking the Driving Environment with a Particle Based Occupancy Grid”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 4, 2011, pp. 1331-1342.

[15] A. Vatavu, Sergiu Nedevschi, Florin Oniga, “Real Time Object Delimiters Extraction for Environment Representation in Driving Scenarios”, in proc of *6th International Conference on Informatics in Control, Automation and Robotics,* Vol. *Robotics and Automation,* 2009, Milano, Italy, 2009, pp 86-93.

[16] M. Isard, A. Blake, “CONDENSATION -- conditional density propagation for visual tracking”, *International Journal of Computer Vision*, Vol. 29, No. 1, pp. 5-28, (1998).

[17] R. Danescu, S. Nedevschi, M.M. Meinecke, T. Graf, “Stereovision Based Vehicle Tracking in Urban Traffic Environments”, in proc of *IEEE Intelligent Transportation Systems Conference (ITSC 2007)*, Seattle, USA, 2007, pp. 400-404.