Environment Perception Using Dynamic Polylines and Particle Based Occupancy Grids

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

Abstract— **Modeling and tracking dynamic entities in the driving environment is a complex task, as one has to accommodate multiple types of scenarios. Extraction of dynamic properties of obstacles becomes difficult when the measurement sensors do not provide speed directly. The dynamic polyline representation of obstacles is a compromise between the rigid model-based cuboid representation and the model-free representation of occupancy grids. This paper presents a polyline-based obstacle extraction system, based on a particle-based occupancy grid generated by processing of stereovision data. We have developed a real-time flexible method for occupancy grid modeling and representation using particles that move from cell to cell and are created and destroyed based on measurements derived from stereovision. The results of the occupancy grid processing are subjected to the Border Scanner algorithm, which extracts polylines from occupied cells, by taking into consideration only the most relevant information of the grid. The resulting system is able to extract individual objects from the occupancy grid, model them as polylines, and estimate their speed without making assumptions about their shape or size.**

Keywords-environment representation; stereovision; occupancy grid; polygonal model; particle filtering;

I. INTRODUCTION

Real-time perception of dynamic environments is a challenging research task, as it implies choosing adequate models for the dynamic and static entities, models that will accurately describe the geometrical shape and the dynamic evolution in time, and matching these models with the available, and often heterogeneous, sensorial data. One of the most complex and heterogeneous dynamic environment is the driving environment, especially when the scenario is urban. A simpler driving environment, clearly structured, is composed of mostly basic geometrical shapes. The obstacles can be modeled as cuboids having position, size and speed, and the driving surface delimiters can be modeled as parametrical curves. The highway and most of the urban and rural sections of road are usually suitable for geometrical modeling and tracking.

The cuboid-based tracking of obstacles may not be the best choice when the environment to be tracked is an intersection, a busy urban center, or an off-road scenario. Even if parts of this environment can be tracked by estimating the parameters of cuboidal model, many essential parts of the environment will not fulfill the constraints of the models. The main problem of a cuboid-based representation of a dynamic environment is that when the perception system relies on sensors that are unable to deliver dynamic information directly (such is the case of laser scanners or stereovision, capable of delivering 3D information, but not speed), the estimation of speed must rely on data association and cuboid following across frames. An object that is only partially visible, or an object that is not suitable for cuboidal representation, or an object that changes its size or shape will mislead the cuboid-based tracking system, and correct speed estimation will be impossible to achieve. For this reason, any perception system can be greatly improved if the dynamic properties of the environment can be estimated 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 dynamic occupancy grid is a good choice for the driving related dynamic environment, as it is capable of concisely describing the relevant aspects while maintaining a decent level of computation complexity. Maybe one of the first uses of occupancy grids, under the name of probabilistic local maps, is presented by Elfes in [3], in the context of sonar based robot navigation, and the probability inference mechanism for handling the uncertainty of range sensors in computing the probability of each cell's occupancy state is presented in [4]. The initial occupancy grids, such as those presented 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 factor in the environment estimation, the complexity increases significantly, as the cells are now strongly interconnected. The work of Coué et al, presented in [5], uses a 4D occupancy grid, where each cell has a position and two speed components along each axis. By estimating the occupancy of each cell in the 4D grid, the speeds for the classical cells in the 2D grid can be computed. Another solution for the representation of speeds is presented by Chen et al, in [6]. Instead of having a 4D grid, this solution comes back to 2D, but uses for each cell a distribution of speeds, in the form of a histogram.

While the occupancy grids may be a valuable tool for estimating speeds and occupancies at map cell level, it does not provide a means of identifying individual objects. What we need is a method for identifying these objects, which combines the advantages of an occupancy grid with the advantages of a cuboid model, without bringing along the latter's problems. Basically, we need a way to extract freeform objects from the occupancy grid cells. A good approach towards this goal is the extraction of 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. At the same time, the polyline could inherit the properties of type, position, and height, and the dynamic features of the associated object. 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 [9], [10], sonar [11], or vision sensors [11]. 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.

This paper presents a polyline-based obstacle extraction system, based on a particle-based occupancy grid generated by processing of stereovision data. We have developed a real-time flexible method for occupancy grid modeling and representation using particles that move from cell to cell and are created and destroyed based on measurements derived from stereovision. The results of the occupancy grid processing are subjected to the Border Scanner algorithm, which extracts polylines from occupied cells, by performing radial scanning using the position of the ego vehicle as the scan rotation center. For the resulted polylines an average speed is computed, as an average speed of the grid occupied cells neighboring the polyline.

II. System Architecture

Our Dynamic Environment Perception System has been projected for an urban driving assistance system where the surrounding world is crowded and unstructured. We extended our Dense Stereo-Based Object Recognition System (DESBOR) by consequently processing the Elevation Map and building a Particle Occupancy Grid representation with the occupancy and velocity probability distribution of each grid cell. A detailed description about the DESBOR system is presented in [13]. Using Particle Occupancy Grid as intermediate representation and tracking solution we extract dynamic obstacle primitives associated to each object from the scene. The result is a 2.5D compact representation of the environment and a more accurate estimation of the object speeds as only the relevant grid cells are processed. This is achieved by radial scanning of the particle occupancy grid and analyzing the grid delimiters' neighborhood.

The Dynamic Environment Perception system architecture consists in the following main components (see figure 1):

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

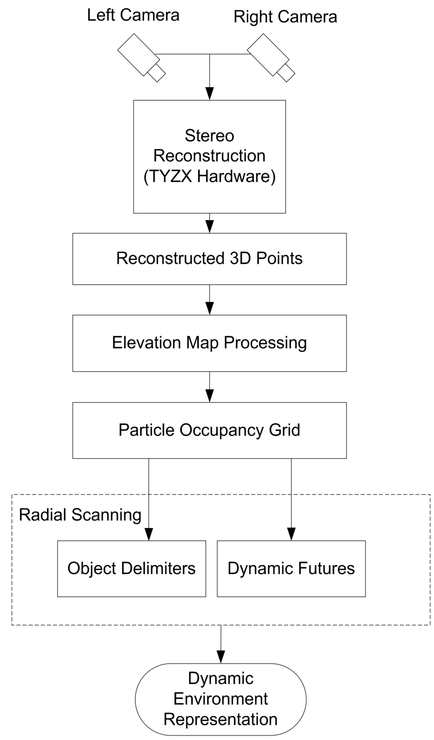


Figure 1. Dynamic Environment Perception System Architecture.

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

Elevation Map Processing Module: The Elevation Map (see figure 2.b) represents a description of the scene, computed from the raw dense stereo information. The Elevation Map cells are classified into road, traffic isle and object cells. A detailed description about the Elevation Map is presented in [8].

Particle Occupancy Grid: Is described as an intermediate Cartesian representation of the environment, using a forward sensor probability model, and producing a fully dynamic grid based on particles (see figure 2.c). 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. A grid particle will also be created and destroyed using a weighting-resampling mechanism specific to particle filter algorithms. A more detailed description of the particle grid tracking algorithm is given in [15] and [16].

Object Delimiters: The Object Delimiters are extracted by radial scanning of the Particle Occupancy Grid. A set of unstructured polygons approximated with the objects contour is generated. For Delimiters Extraction we used the Border Scanning algorithm presented in [12].

Environment Representation Output: A dynamic polyline map is generated as the result of delimiters extraction particle occupancy grid analysis. For each polyline element we keep the following information:

- A set of vertices that describe the polygon.

- Static Object features: position and height.
- Type of the associated obstacle: Static or Moving obstacle, Object delimiter or Curb delimiter.
- Dynamic Features: Object Orientation and Magnitude.

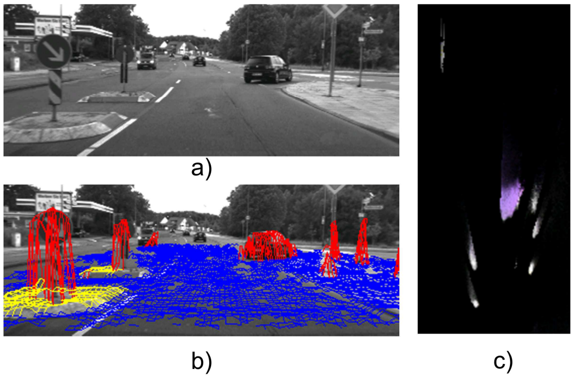


Figure 2. The Elevation map (b) and Particle Occupancy Grid (c) of a scene (a). The Elevation Map cells are roughly classified (blue – road, yellow – traffic isle, red – obstacles).

III. Particle Occupancy Grid

Our occupancy grid solution is defined by a new and original method of representation of the occupancy and velocity probability distribution of each grid cell, and by the original updating algorithm derived from this representation. The occupancy probability of each grid cell is described by the number of particles in that cell, and the particles have a dual nature – they describe occupancy hypotheses, as in the particle filtering algorithms such as CONDENSATION [7], but can also be regarded as physical building blocks of our modeled world. The grid tracking algorithm is particle-oriented, not cell oriented. 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 described in [7]. The measurement data is the raw obstacle grid obtained by processing the elevation map.

The world is represented by a 2D grid, mapping the bird-eye view 3D space into discrete 20 cm x 20 cm cells. The size of the grid is 250 rows x 120 columns, corresponding to a scene size of 50x24 meters. The aim of the tracking algorithm is to estimate the occupancy probability of each grid cell, and the speed components on each axis. The tracking goals are achieved by the use of a particle-based filtering 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:

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

each particle *i* having a position in the grid, described by the row r_i (a discrete value of the distance in the 3D world *z*) and the column c_i (discrete value of the lateral position *x*), 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 is not fixed, but 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.

$$P_O(C) = \frac{|\{ p_i \in S \mid r_i = r_c, c_i = c_c \}|}{N_C} \tag{2}$$

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. A large number means that on a single cell multiple speed hypotheses can be maintained, and therefore the tracker can have a better speed estimation, and can handle fast moving objects better. However, the total number of particles in the scene will be directly proportional with N_C, and therefore the speed of the algorithm will decrease.

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. Of course, the particle population can handle the situation when multiple obstacles, having different speeds, share the same cell, and in this case the speed estimate of the cell must be computed by clustering.

$$(vc_C, vr_C) = \frac{\sum_{p_i \in S, x_i = x_c, z_i = z_c} (vc_i, vr_i)}{|\{ p_i \in S \mid r_i = r_c, c_i = c_c \}|} \tag{3}$$

Thus, the population of particles is sufficiently representative for the probability density of occupancy and speed for the whole grid. 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 (see figure 3). 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. This step is based on the raw occupancy cells provided by dense stereo processing, and provides the measurement model for each cell.

The measurement model information is used to *weight* the particles, and *resample* them in the same step (see figure 4). 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 [15] and [16].

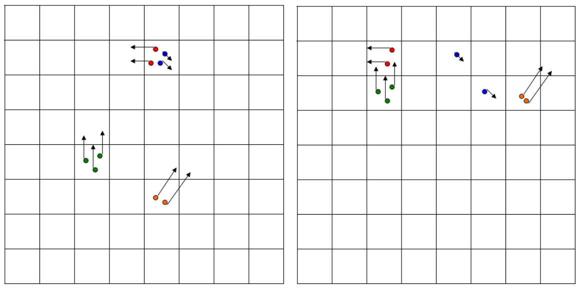


Figure 3. Migration of particles from one cell to another, as prediction is applied.

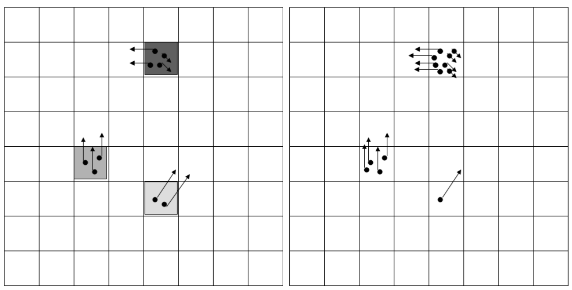


Figure 4. Weighting and resampling. The weight of the occupied hypothesis is encoded in the darkness of the cell of the left grid. In the right grid, the effect of resampling is shown, as particles are multiplied or deleted.

IV. Delimiters Extraction using Digital Occupancy Grid

Regardless the unstructured environment representation solutions, there are some basic problems which significantly influences the surrounding world modeling as well as the static or dynamic parameters computation. Beside the stereo reconstruction noises, an unstructured dynamic environment may include two types of errors:

- **Temporal errors**: the object shapes are influenced by the presence or absence of the information at different moments of time.
- **Spatial Errors**: when an object part is occluded it may lead to a noisy or a partial representation of that object. In this situation it's difficult to model and to track such an object at a high accuracy and confidence.

In order to reduce the errors described above we developed a method which takes into consideration only the most visible parts from the observation point (ego-car) by radial scanning of the occupancy grid. The result is a polygonal model of the environment with the obstacle dynamic features associated to it.

For the delimiters extraction we extend the Border Scanner algorithm described in [12]. In our case we use a Probabilistic Occupancy Grid as the input information. The main idea is that we are taking in account only the most relevant information by extracting object delimiters and its vicinity in order to estimate more accurate speed vectors corresponding to objects form a traffic scene. Our method is based on a Ray-Casting approach by determining the first occupied point intersected by a ray which extends from the ego-car position. The scanning axis moves in the radial direction, having a fixed center at the Ego Car position (see figure 5). The scanning process is made into the limits of two given angles, thus only the interest area are scanned, where the object delimiters can be detected. Having a radial axis with a given slope we try to find the nearest grid cell point from the Ego Car situated on this axis. In this way, all subsequent cells P_i will be accumulated into a *ContourList*, moving the scanning axis in the radial direction:

$$ContourList = \{P_1, P_2, ...P_n) \tag{4}$$

Once an object point P is reached we compute its speed (vr_i, vc_i) by averaging speed components on a connected neighborhood of W*W size, with W = 2k+1 (for our experiments we used a k=1):

$$\overline{vr_i} = \frac{1}{W^2}\sum_{l=-k}^{k}\sum_{m=-k}^{k} vr_{lm}\ ,\ \overline{vc_i} = \frac{1}{W^2}\sum_{l=-k}^{k}\sum_{m=-k}^{k} vc_{lm} \tag{5}$$

Thus, each delimiter point P_i is described by its position (r_i,c_i) as well as by its speed components ($\overline{vr}_i, \overline{vc}_i$).

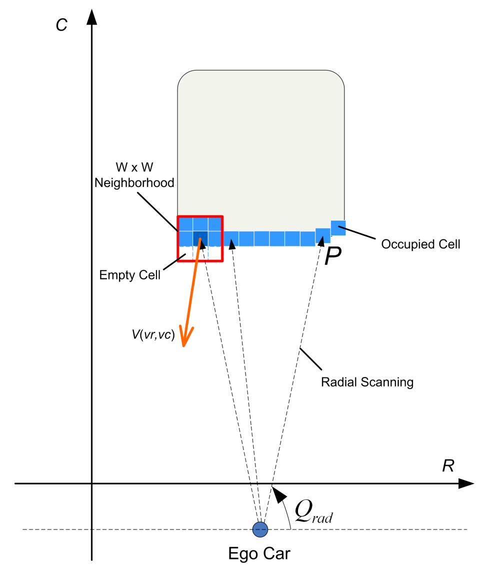


Figure 5. Occupancy Grid Radial Scanning. A mean speed V is computed for each scanned point by averaging speed components on a connected neighborhood of W*W size.

The mean vector of an obstacle is computed by using vector addition for all delimiter point speeds accumulated into the *ContourList*. Given a set of individual components ($\overline{vr}_i, \overline{vc}_i$), the object speed is calculated:

$$\overline{vr} = \frac{1}{n}\sum_{i=1}^{n}\overline{vr}_i\ ,\ \overline{vc} = \frac{1}{n}\sum_{i=1}^{n}\overline{vc}_i \tag{6}$$

Where n is the number of delimiter points.

The object speed magnitude M is defined by:

$$M = \frac{1}{n}\sqrt{\overline{vr}^2 + \overline{vc}^2} \tag{7}$$

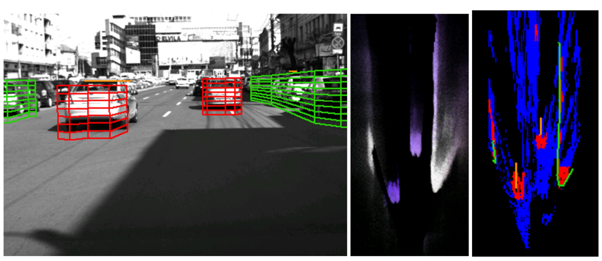
The mean direction $\overline{\theta}$ is described by the function:

$$\overline{\theta} = atan2\,(\overline{vr}\,, \overline{vc}\,). \tag{8}$$

V. Experimental Results

For the experimental results we have tested several scenarios from the urban traffic environment using a 2.66GHz Intel Core 2 Duo Computer with 2GB of RAM. Figure 6 illustrates the obtained results in a dynamic traffic scenario. The static obstacle delimiters are colored with green while dynamic obstacles are represented by red delimiters. We considered that the objects with a speed greater than 6km/h are dynamic. The intermediate representation by Particle Occupancy Grid is shown in the figure 6.b. Figure 6.c presents a top view of the Elevation Map and Polyline Representation based on Particle Occupancy Grid. The speed vectors associated to each dynamic obstacle are colored with yellow.

For the numerical evaluation of the results we have performed tests for the following scenarios: an incoming vehicle, a stationary lateral vehicle, and an outgoing dynamic motorcycle. The delimiters speeds are compared to the speeds obtained directly by the particle grid tracking method [15], and they are also compared to the object cuboidal model tracking [17]. The first case is of an incoming vehicle, which due to its high relative speed is seen only for a brief period of time. The results of the speed estimation are shown in figure 7.



a) b) c)

Figure 6. Dynamic Environment Perception (a), the Particle Occupancy Grid (b), and the top view of the Elevation Map (c) with the extracted polylines. Speed vectors are colored with yellow.

It can be seen that the obtained measurements by particle occupancy grid tracking (blue color) are available earlier than the object cuboid tracking speeds (yellow color), but the particle grid results converge slower to the model based measurements. Although the cuboid tracking speeds seem to be more stable, the results are available only for a short period of time. It can be observed that the delimiter based speed estimation (magenta color) converges almost immediately to the values estimated by the model tracking. It provides a better match to the real speed than the particle grid tracking method and the results are available earlier than the cuboid based tracking technique.

The second test was performed in a scenario with a stationary lateral obstacle. In this case the target speed of 0 is given as the ground truth for our measurements. Analyzing the results illustrated in the figure 8, we can see that the obstacle delimiter based speed estimation, converge more quickly towards the ground truth. The estimated speeds are more stable than the grid based tracking measurements (lower error standard deviation). The result is more accurate comparing to the other two outputs (lower mean absolute error). This fact is confirmed by the table 1.

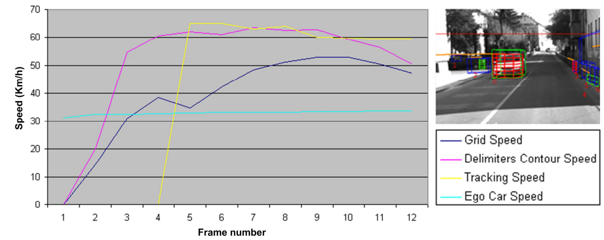


Figure 7. Speed Estimation. Incoming vehicle.

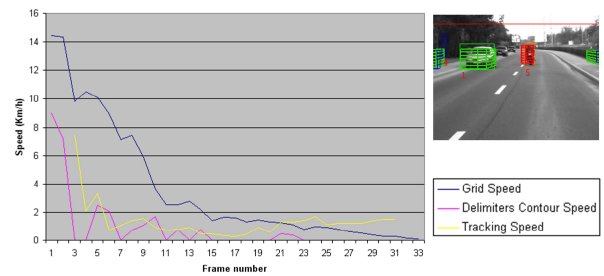


Figure 8. Speed Estimation. Stationary lateral vehicle (green color).

TABLE I. SPEED ESTIMATION ACCURACY

Accuracy Metrics	**Speed Estimation Methods**		
	Dynamic Polylines	***Particle Occupancy Grid***	***Cuboid-based Tracking***
STDEV (km/h)	1.99	4.23	1.32
MAE (km/h)	0.81	3.62	1.36

The third case is of an outgoing motorcycle. First, the speed estimation was performed without using the polyline neighborhood by involving only the contour points (see figure 9, top diagram), then we have tested the same scenario by averaging contour speeds on a connected neighborhood of 3x3 size (see figure 9, bottom diagram). It can be observed that the particle occupancy grid and the object delimiter results converge more quickly towards the mean values estimated by cuboid based object tracking. The model based tracking is not able to provide speed values in a reasonable time because its object association stage is influenced by the high speed of the motorcycle (about 90 Km/h). Also we can see that the delimiter based estimation is closer to the cuboid based tracking results. However, in this case the contour based approach is not as stable as the particle occupancy grid method (a greater standard deviation). This is because the amount of information is smaller and inaccurate at far distances from the ego-car. A possible solution is to choose a larger neighborhood at once with the distance and the object size. The numerical evaluation demonstrates that, in the first case, by choosing a 3x3 neighborhood, we obtained a lower standard deviation, 9.2 km/h in comparison with the case when a neighborhood is not employed (10.9 km/h).

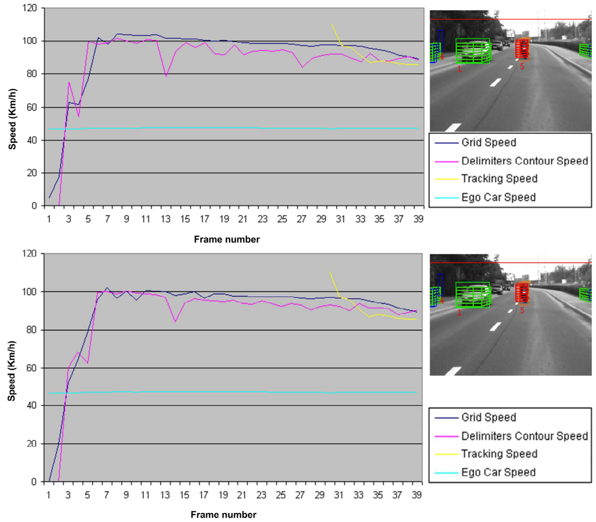


Figure 9. Speed Estimation. Outgoing motorcycle (red color). Top diagram: results are estimated by using a polyline neighborhood of 3x3 size. Bottom diagram: results are estimated without averaging speed with neighboring cells.

VI. CONCLUSIONS

In this paper we present a new fast converging technique for dynamic obstacles detection and representation using a particle-based occupancy grid generated by processing of stereovision data. We have developed a real-time flexible method for occupancy grid modeling and representation using particles that move from cell to cell and are created and destroyed based on measurements derived from stereovision. The results of the occupancy grid processing are subjected to the Border Scanner algorithm, which extracts polylines from occupied cells, by performing radial scanning using the position of the ego vehicle as the scan rotation center. For each extracted delimiter, an average speed is computed, as an average speed of the grid occupied cells neighboring the polyline. The result is a more compact representation of the environment and a more accurate estimation of the object speed.

For the numerical evaluation of the results we have performed tests for the following scenarios: an incoming vehicle, a stationary lateral vehicle, and an outgoing dynamic motorcycle. The delimiters speeds are compared to the speeds obtained directly by a particle grid tracking method, and they are also compared to the object cuboidal model tracking. The delimiter based speed estimation proved to be more accurate comparing to the other two outputs. As future work we propose to focus our research in improving the accuracy of the Environment Representation by using a fusion between our solution and others fast converging speed estimation methods.

VII. ACKNOLEDGMENT

This work was supported by CNCSIS –UEFISCSU, project number PNII – IDEI 1522/2008.

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

[8] 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, 2010.

[9] S. Kolski, D. Ferguson, M. Bellino, R. Siegwart, "Autonomous driving in structured and unstructured environments", in proc of *IEEE Intelligent Vehicles Symposium*, 2006.

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

[11] A. Goncalves, A. Godinho, J. Sequeira, "Lowcost sensing for autonomous car driving in highways", in proc of *ICINCO2007*, Angers, France, 2007.

[12] A. 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.

[13] 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: *Intelligent Vehicles Symposium* 2007, pp. 278—286.

[14] J. I. Woodill, G. Gordon, R. Buck, “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] R. Danescu, F. Oniga, S. Nedevschi, “Particle Grid Tracking System for Stereovision Based Environment Perception”, in proc of *IEEE Intelligent Vehicles Symposium* 2010.

[16] R. Danescu, F. Oniga, S. Nedevschi, “Modeling and Tracking the Driving Environment with a Particle Based Occupancy Grid”, *IEEE Transactions on Intelligent Transportation Systems*, in print.

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