A Flexible Solution for Modeling and Tracking Generic Dynamic 3D Environments*

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

***Abstract*— The traffic environment is a dynamic and complex 3D scene, which needs accurate models to represent it and reliable algorithms to perceive it. This paper presents a new model for representing the dynamic 3D environment of the traffic scene, the particle based dynamic elevation map enhanced with gray level information. A new tracking algorithm, based on the measurement cues extracted from dense stereovision, is employed for estimating the state of the proposed model. The multimodal probability density of a map cell's state concerning the cell's speed, height and gray level, is approximated by a population of particles, which can migrate from one cell to another using their speeds. The particle population is updated using a weighting-resampling mechanism based on the particle's fitness with the measurement data, which consists of the raw heights obtained from stereovision and the pixels of the grayscale left image. The measurement model used for particle weighting is designed by taking into account the specifics of the stereovision sensor. The dynamic elevation map tracking system is able to provide a dense and accurate representation of the observed scene, to improve the density and accuracy of the stereo-based environment perception, and to enhance it with dynamic information. The final result is an accurate virtual representation of the perceived 3D scene.**

I. Introduction

A complex and dynamic 3D scene, such as the ones encountered in the urban traffic, may be hard to model with oriented boxes [1, 2]. Some shapes are not well suited for box representation, or the occlusions and sensor limitations prevent a successful box fit. The traffic scene is composed of obstacles, which are static or dynamic, road surface, curbs, sidewalks, generic 3D structures that sometimes defy our attempt to model them with simple geometrical primitives. Many researchers try to overcome this problem by designing generic solutions for representation and tracking of 3D environments, models that are not bound to particular object types. The simplest approach is to store raw sensorial points, and use them to discriminate against accessible areas and obstacles. These points can also be used for map building, if they are highly accurate, delivered by a high accuracy high density laser scanner [3]. An ambitious attempt to model and track the generic dynamic 3D environment is the 6D vision, each point of the scene having position and a 3D speed vector [4]. A more compact representation is the dynamic stixel set [5], which models the visible sides of obstacles as a set of dynamic vertical structures, the speeds being confined to the road surface.

For the purpose of navigation, complex 3D environments can be modeled as digital elevation maps (DEM). The map is a 2D grid, each cell in the grid having a height above a globally or locally defined zero level. The DEMs can be large data structures, when they are used for terrain mapping [6], a function which makes them extremely useful for planetary exploration tasks [7], but they can also be more compact structures, used for the navigation of autonomous robots [8] or intelligent vehicles [9].

The elevation map's nature of environment representation makes it well suited to be estimated in real time, using sensors that are capable of delivering dense 3D data, such as the laserscanner [2] and the stereovision [9] [10]. For real-time navigation assistance, the cells of the elevation map can be quickly analyzed and classified into traversable, obstacles and others [8] [9].

The elevation map is not a complete description of the 3D environment – bridges and tunnels, for example, cannot be fully represented, and for this reason this kind of representation is said to be 2.5 dimensional [11]. Some extensions, such as the Multi-Level surface maps [12] or the multi-volume occupancy grid [13], offer some workarounds for this problem, by storing multiple heights for the same map cell.

Occupancy grids are another way of representing the perceived environment in the form of a 2D, bird-eye view map. However, instead of heights, the occupancy grid stores in each cell a probability of this cell to be occupied by an obstacle. Many occupancy grid solutions provide a probabilistic reasoning framework for using uncertain sensorial data [14], and recent ones can also handle the presence of dynamic obstacles [2][15].

The elevation map and the occupancy grid both have desirable properties that make them useful for driving environment representation and tracking, and thus a combination of the two seems to be a natural extension. In [16], we find a stereovision-based solution which maintains two maps, one for occupancy and one for height, while in [17] the elevation map is used as an intermediary processing step towards achieving the occupancy grid representation.

*This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-ID-PCE-2011-3-1086.

R. Danescu is with the Technical University of Cluj-Napoca, Cluj-Napoca, 400114 Romania (phone: +40-264-401457; fax: +40-264-594491; e-mail: radu.danescu@cs.utcluj.ro).

S. Nedevschi is with the Technical University of Cluj-Napoca, Cluj-Napoca, 400114 Romania (e-mail: sergiu.nedevschi@cs.utcluj.ro).

.

II. Overview of the Proposed Solution

This paper presents a further extension of the occupancy grid/elevation map combinations, a solution for direct modeling and tracking fully dynamic elevation maps. While in [12] and [13] we can find sophisticated techniques for describing complex static 3D environments, and in [16] and [17] the elevation maps are used along with occupancy grids, this paper presents a unified solution for modeling and estimating both the static and the dynamic characteristics of the environment. Each cell in the dynamic elevation map has a probability distribution of height and speed, and these distributions will be updated according to the measurement data, which is available from dense stereo processing. Also, due to the fact that the stereovision system delivers not only 3D data, but image gray level data as well, we shall model, for each cell, a probability distribution of gray values, which will be useful in the state update process, allowing us to use an additional gray level cue, and will increase the fidelity of the estimated 3D world model.

The proposed dynamic elevation map modeling and tracking solution is based on moving particles. Each particle has a height, a speed and a gray value, and is located in a cell of the map. The particle set of a cell forms a multi-modal probability distribution over a 4-dimensional state space (height, forward speed, lateral speed, and gray value).

The particles can move from one cell to another, providing an elegant and intuitive mechanism for prediction, and can be created and destroyed based on their agreement with the measurement data (the state update). The same particle-based mechanism was used in [18], for modeling and tracking dynamic occupancy grids. However, the moving particle can carry more than position and speed information. Thus, the solution described in this paper is based on particles carrying information about position, speed, height and gray level, leading to a very flexible probabilistic model of a fully dynamic, gray level enhanced elevation map. The particle population is controlled by the measurement data, which consist of the instantaneous, static elevation map constructed from dense stereo, using a method described in [9], and the image gray levels associated to the elevation map cells.

The tracking algorithm continuously estimates the probability density of the height, speed and gray value for each cell in the map. As these probability densities are represented by particles, the purpose of the tracking algorithm is to manage the particle population, using the measurement data. A block diagram of the solution is presented in Fig. 1.

The first step of the tracking cycle is the *Particle Drift*, which applies motion equations to the particles' past states in order to predict their position at the present time. This step causes the particles to move from one cell to another. The drift is followed by the stochastic *Diffusion*, where the parameters of the particles (position, height, speed and gray level) are altered by random amounts, reflecting the uncertainties of the real world transformations. Together, *Drift* and *Diffusion* form the prediction.

The measurement data is represented by the *Raw Elevation Map*, derived from dense stereo data processing (and affected by stereo reconstruction specific errors), and by the *Grayscale left image* captured from the left camera. The raw elevation map is combined with the grayscale image to produce a *Raw Grayscale Map*, which assigns a gray level value for each cell in the raw elevation map that has a valid height.

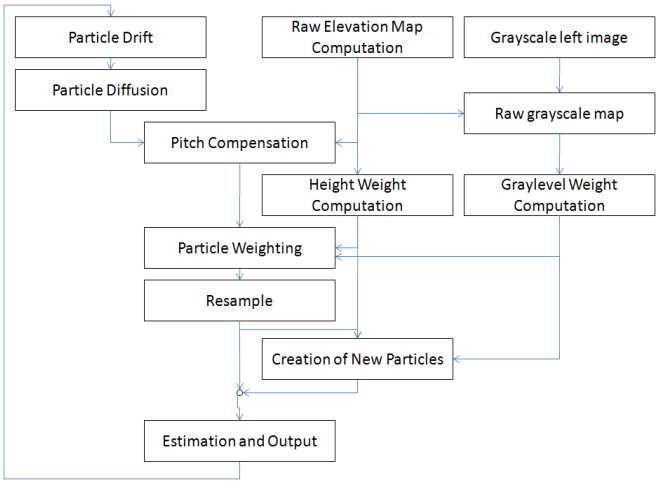

Fig. 1. Overview of the proposed tracking algorithm.

Due to the fact that the host vehicle can pitch randomly and sometimes abruptly between observation times, the height of the predicted particles has to be adjusted to compensate for this (*Pitch Compensation*). The pitch difference is estimated by comparing the predicted heights and the heights in the raw map, under the assumption that most of the scene is made up by static elements, whose heights should not change.

After the pitch-based height adjustment, the particles are subjected to *Particle Weighting*. The particles are compared to the measurement data, in terms of height and gray value. This process must take into account the uncertainty of the stereovision process. For speedup purposes, weight Look-Up Tables (LUTs) are computed for each map cell, for each possible particle height and gray level, in the process of *Height Weight Computation* and *Gray level Weight Computation*. Each particle will get a height weight and a gray level weight from the LUTs, and the final weight will be the product of the two.

After weighting, a new population of particles is generated for each map cell, using the process of *Resampling*, the weight of the particle influencing the chances of it being sampled.

If a cell in the grid has very few particles (or none), the measurement data, already pre-processed in the form of a height weight LUT and a gray level weight LUT, will be used to create new particles, with random speeds, and height and gray level distributions consistent with the LUTs.

After the particle population is updated, it reflects the updated probability densities for the state of the environment, and the system is ready for a new cycle. A dynamic elevation map containing an estimated value of the

height, speed and gray level for each cell is the result of the *Estimation and Output* step.

III. The World Model

Assuming the coordinate system with the origin on the ground in front of the host vehicle, the *X* axis pointing forward, *Y* axis pointing to the right, and *Z* axis pointing up, the horizontal plane *XOY* is divided into cells of 20 cm x 20 cm, each cell *i* being identified by a row coordinate r_i and a column coordinate c_i. Each cell has an associated height value h_i, and a gray level value g_i. Also, since the 3D scene we want to model is dynamic, each cell has an assigned speed vector, which we'll confine to the horizontal plane, and therefore it will have two components, for the *X* and *Y* axes, or, in terms of rows and columns, $v_{r,i}(r_i,c_i)$ and $v_{c,i}(r_i,c_i)$ – the row speed and the column speed for each cell *i* in the map (Fig. 2).

Each cell *i* in the dynamic gray level elevation map can be described by four values, h_i, g_i, $v_{r,i}$ and $v_{c,i}$. Due to the fact that perfect sensing of a real traffic scene is impossible, one cannot have exact knowledge of these values. These limitations lead to uncertainties, which have to be included in the world model as probability densities.

A cell *i* in the dynamic elevation map is associated to a four-dimensional random variable $\mathbf{X}_i = (h_i, g_i, v_{r,i}, v_{c,i})^T$. The objective of the tracking algorithm is to compute the probability density of $\mathbf{X}_i$, for each cell *i* in the map, based on a sequence of measurements $\mathbf{Z}(0)\ldots\mathbf{Z}(t)$, *t* being the current observation time.

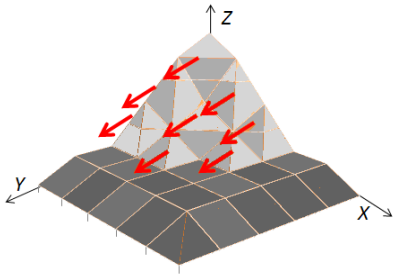

Fig. 2. The dynamic, gray level elevation map.

The dynamic elevation map will be described, at time *t*, by a set of particles $S(t)=\{\mathbf{x}_k(t) \mid \mathbf{x}_k(t)=(c_k(t), r_k(t), h_k(t), g_k(t), v_{c,k}(t), v_{r,k}(t))^T, k=1\ldots N_S(t)\}$, each particle *k* being located in the 2D grid, in the cell having the row r_k, and the column c_k. The grid is a map of 250 rows x 120 columns, and each cell in the grid is a rectangle of 20 cm x 20 cm. Each particle represents a hypothesis of the state of the cell: a possible height $h_k(t)$, a possible gray level $g_k(t)$, a possible forward speed $v_{r,k}(t)$ and a possible lateral speed $v_{c,k}(t)$, as depicted in Fig. 3. The row, column and speed of a particle are expressed as multiples of the cell size D_X and D_Y (currently 20 cm), and the height is expressed as a multiple of the height element of size D_H (currently D_H=1 cm).

The probability density of the state of a cell *i* is approximated by the speed, height and gray level values of the particles *k* located inside cell *i*, as shown by equation (1).

$$P(\mathbf{X}_i(t) \mid \mathbf{Z}(0), \mathbf{Z}(1), \ldots, \mathbf{Z}(t)) \approx \{\mathbf{x}_k(t) \in S(t) \mid r_k(t) = r_i, c_k(t) = c_i\} \quad (1)$$

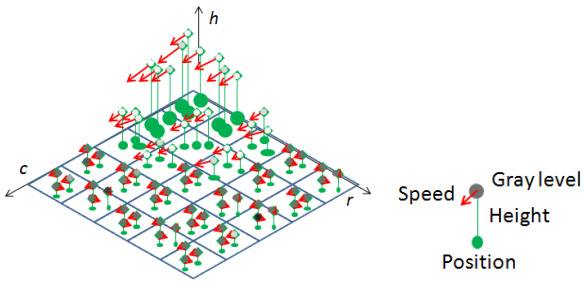

Fig. 3. Particle approximation of the uncertain state of the world.

IV. Algorithm Description

A. Prediction – Drift and Diffusion

The state transition probability model is implemented by the deterministic drift and the stochastic diffusion. The deterministic drift changes the state of the particles by taking into account two factors: the movement of the host vehicle, which causes a relative movement of the whole scene in the vehicle's coordinate systems, and the movement of the mobile particles, according to their speed. The host vehicle's movement can be computed from its speed and its yaw rate, which are read from the CAN bus and integrated over the time interval between two measurement frames. A detailed description of this process can be found in [18].

After drift, the particles are subjected to diffusion. The state of each particle is altered by random quantities $\delta c(t)$, $\delta r(t)$, $\delta h(t)$, $\delta g(t)$, $\delta v_c(t)$ and $\delta v_r(t)$, drawn from Gaussian distributions of zero mean and experimentally adjusted covariance matrix $\mathbf{Q}_i(t)$.

B. The measurement data

The dense stereovision data is transformed into a raw elevation map. A detailed description of the process can be found in [9]. The raw map is described by two 2D arrays:

- Measured height of each cell, denoted by $z_i(r_i,c_i)$;
- Measured gray level of each cell, denoted by $b_i(r_i,c_i)$. The gray level of a map cell is found by transforming the row, column and height coordinates of the raw height map into *XYZ* coordinates, and projecting them into the left image plane.
- Data availability for each cell, denoted by $d_i(r_i,c_i)$. $d_i(r_i,c_i)$=1 means height for this cell is available, and $d_i(r_i,c_i)$=0 means that no measurement data is available for cell *i*.

A cell may have no valid measured height due to the fact that the stereovision engine may not be able to provide a 3D value for all areas in the image, or due to occlusions and

field of view limitations. The tracking algorithm is made aware of this situation by $d_i(r_i,c_i)$. If the cell has no measured height, we'll assume that it has no measured gray level, either. Figures 4-6 show the stages of the process of measurement data extraction.

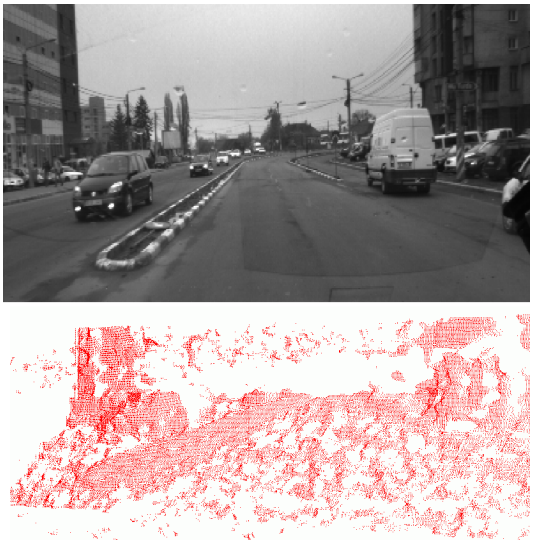

Fig. 4. Original grayscale left image (top), and dense stereo reconstructed 3D points (bottom).

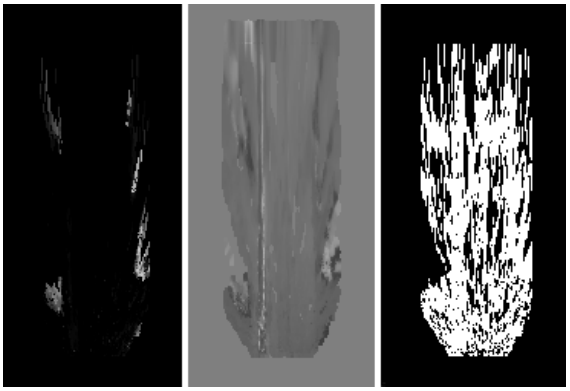

Fig. 5. Measurement data. Left – heights of each cell (a brighter value means a greater height of a cell); Middle – gray level values for the map cells; Right – Data availability for each cell – white means measurement data is available.

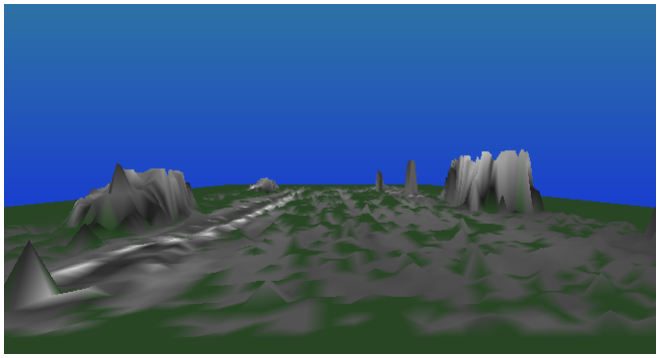

Fig. 6. Measurement data, represented as a 3D surface. The cells without data availability are shown in green.

C. Pitch angle compensation

The world we observe will not change abruptly between measurements, therefore the heights of the map cells are expected to remain, on the average, constant. Abrupt changes of cell heights are expected when the host vehicle is pitching, the effect of a pitch angle variation $\Delta\alpha$ causing a height difference between the particles in the cell and the measurement data:

$$\Delta\alpha_k = \tan^{-1}\left(\frac{(h_k - z_i(r_k,c_k))D_H}{D_X r_k}\right) \quad (2)$$

Equation (2) is valid only when the particle belongs to a static structure, and its height resembles the true height of the structure in the scene. These conditions are not valid in the general case, but we can assume that most of the particles belong to static structures and the errors average out. By averaging the pitch candidate value of each particle, a pitch difference value is estimated for the entire scene.

After the pitch difference value is estimated, the height of each particle is corrected. Even if this correction is not perfect, as height changes may occur also from host vehicle rolling, and the pitch angle may be sometimes incorrectly estimated, it is enough to compensate for the most critical situations.

D. Particle weighting

The process of particle weighting is the particle filter instantiation of the measurement (observation) model $P(\mathbf{Z}(t) \mid \mathbf{X}_i(t) = \mathbf{x}_k(t))$. This model describes the conditional probability density of the measurement $\mathbf{Z}(t)$ given a possible cell state value $\mathbf{X}_i(t) = \mathbf{x}_k(t)$. A particle is a state hypothesis, and the probability of the measurement given this state will be encoded as a particle weight, which will describe how well the particle hypothesis matches the measurement data.

The raw elevation map is just a conveniently modified version of the stereovision-derived 3D point data, therefore the probabilistic observation model will be derived from the observation model of stereovision, a three-dimensional normal distribution centered in the real world 3D coordinate, and having a covariance matrix defined by the distance error standard deviation σ_X, the lateral error standard deviation σ_Y and the vertical error standard deviation σ_Z.

The expected error standard deviations of the three coordinates computed by a stereo reconstruction process depend on the system's baseline (distance between cameras) *b*, the focal distance in pixels *f*, and disparity computation uncertainty (matching uncertainty) σ_d. The error standard deviation for the distance coordinate *X* can be computed as [19]:

$$\sigma_X = X^2\sigma_d / bf \quad (3)$$

The error standard deviations for the lateral coordinate *Y* and for the vertical coordinate *Z*, σ_Y and σ_Z, can be computed using equations (4) and (5).

$$\sigma_Y = Y\sigma_X / X \quad (4)$$

$$\sigma_Z = Z\sigma_X / X \quad (5)$$

The uncertainties of *X*, *Y* and *Z* are then converted in uncertainties of row, column and height:

$$\sigma_r = \sigma_X / D_X + \sigma_{r0} \quad (6)$$

$$\sigma_c = \sigma_Y / D_Y + \sigma_{c0} \quad (7)$$

$$\sigma_h = \sigma_Z / D_H + \sigma_{h0} \quad (8)$$

The offsets σ_{r0}, σ_{c0} and σ_{h0} are added so that other measurement errors, besides matching uncertainty, can be accounted for. These offsets are tuned experimentally.

A particle will receive two weights, one which will reflect the agreement between the particle height and the measurement heights in the raw elevation map, and the other reflecting the agreement between the particle's gray level and the measurement gray levels.

Due to the uncertainties of the measurement, the best height and gray level measurement for a real world cell may not be found in the corresponding measured cell, but in neighboring cells. Thus, for a particle located in a cell *i*, its height and gray level must be compared to measured values in an uncertainty region of $4\sigma_r$ x$4\sigma_c$ around the central position (r_i, c_i). In order to make this process more efficient, Look-Up Tables (LUTs) are created for each cell, for each possible value of height and gray level, in the process of *Height Weight Computation* and *Gray level weight computation*.

For creating the height weight LUT, each measured height *z* inside the uncertainty region for a cell *i* votes in its corresponding height histogram position, but the weight of its vote will be generated by a Gaussian kernel Γ_i centered in (r_i,c_i), having the standard deviations $\sigma_r(i)$ and $\sigma_c(i)$. This is a straightforward application of the Gaussian observation model in the horizontal plane. Formally, the histogram value for a height candidate *h*, at coordinates (r_i, c_i), for the cell *i*, is computed using equation (9):

$$H_i(h) = \sum_{\tau=r_i-2\sigma_r}^{r_i+2\sigma_r} \sum_{\kappa=c_i-2\sigma_c}^{c_i+2\sigma_c} d(\tau,\kappa)\Gamma_i(\tau-r_i,\kappa-c_i)\delta(z(\tau,\kappa)-h) \quad (9)$$

The process of height histogram creation is depicted in Fig. 7.

We can apply the same reasoning for the measured gray values *b*(*r*, *c*), and create a weight histogram G_i, for each candidate gray level *g*, using equation (10):

$$G_i(g) = \sum_{\tau=r_i-2\sigma_r}^{r_i+2\sigma_r} \sum_{\kappa=c_i-2\sigma_c}^{c_i+2\sigma_c} d(\tau,\kappa)\Gamma_i(\tau-r_i,\kappa-c_i)\delta(b(\tau,\kappa)-g) \quad (10)$$

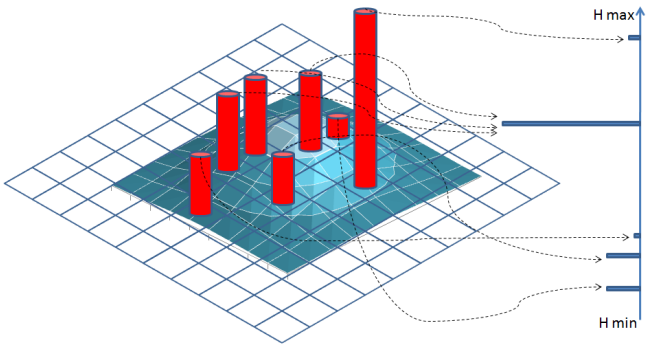

Fig. 7. Building the height histogram for a cell. A candidate height, located inside the stereo uncertainty region, will have a vote weighted by the value of the 2D Gaussian kernel centered in the current cell.

In order to account for the height uncertainty, the height histogram H_i will be convolved with a 1-dimensional Gaussian kernel $K_{H,i}$, of standard deviation $\sigma_H(i)$, obtaining the LUT for particle weighting with respect to height, $W_{H,i}$.

$$W_{H,i} = K_{H,i} * H_i \quad (11)$$

Similarly, the gray level histogram G_i will be convolved with a Gaussian kernel $K_{G,i}$, of standard deviation $\sigma_G(i)$, accounting for the uncertainty of gray level estimation (this includes uneven lighting, reflections, etc), obtaining the weight LUT for particle's gray levels, $W_{G,i}$.

$$W_{G,i} = K_{G,i} * G_i \quad (12)$$

After the weight LUTs are created, the weights of the particles can be easily computed. A particle *k*, in a cell *i*, will get the following weight:

$$w_k = W_{H,i}(h_k)W_{G,i}(g_k) \quad (13)$$

E. Resampling and creation of new particles

In order to update the state of the perceived environment in accordance with the measurement data, resampling is applied for each cell *i*, after the particles are weighted. The total number of allowed particles for a cell is N_C, a constant which is currently set at 200. The real number of particles in the cell, $N_{R,i}$, resulted by drift and diffusion, may be lower than N_C,, but never higher, as excess particles resulted from drift and diffusion are discarded. For re-sampling, we'll assume that the cell holds a higher number of particles, N_A = 1.25 N_C. The difference between the real number of particles in the cell, $N_{R,i}$, and the augmented maximum number of particles N_A is the number of "empty" particles, particles which are in fact empty places. The re-sampling mechanism will perform the following steps:

1. Weight the empty particles with a default low weight, which we chose to be the average value of the height weight LUT, $W_{H,i}$, multiplied by the average value of the gray level weight LUT, $W_{G,i}$. The number of empty particles multiplied by the default weight is the assumed probability of a complete mismatch between the predicted state and the measured data.

2. Normalize the weights of all N_A particles so that their sum becomes 1.
3. Perform N_C random extractions from the total particle population, real and empty, the weight of the particle controlling its chances of being selected [20].

The mechanism for decreasing the particle population of a cell when most of these particles do not match the measurement data is designed to speed up the convergence of the state estimation in the presence of moving obstacles. A moving obstacle's particles must replace ground particles, but if a cell is already full with ground particles, the moving obstacle particles have no place to go, most of them will be lost, and the moving obstacle will not be successfully tracked.

If a map cell has a low number of particles, the measurement data is used to create new particles to populate the cell. The speeds of the new particles will be sampled from a normal distribution centered in 0, the heights will be sampled from the multi-modal distribution represented by the weight LUT $W_{H,i}$ and the gray levels will be sampled from the gray level LUT $W_{G,i}$. The process of creating new particles is applied after each resampling step.

V. Results

In order to assess the world modeling and tracking capabilities of the proposed solution, multiple sequences acquired in the urban traffic of Cluj-Napoca, Romania, were used. In the absence of a definite ground truth, we made a subjective evaluation of how well the tracking system was able to provide a virtual 3D representation of the perceived environment, in comparison with the raw map used as measurement. Some results are shown in Fig. 8. The tracking system was able to considerably improve the density of the raw map, to filter out the reconstruction errors, especially in the case of the road surface and to correctly identify the moving elements in the scene and their direction of motion.

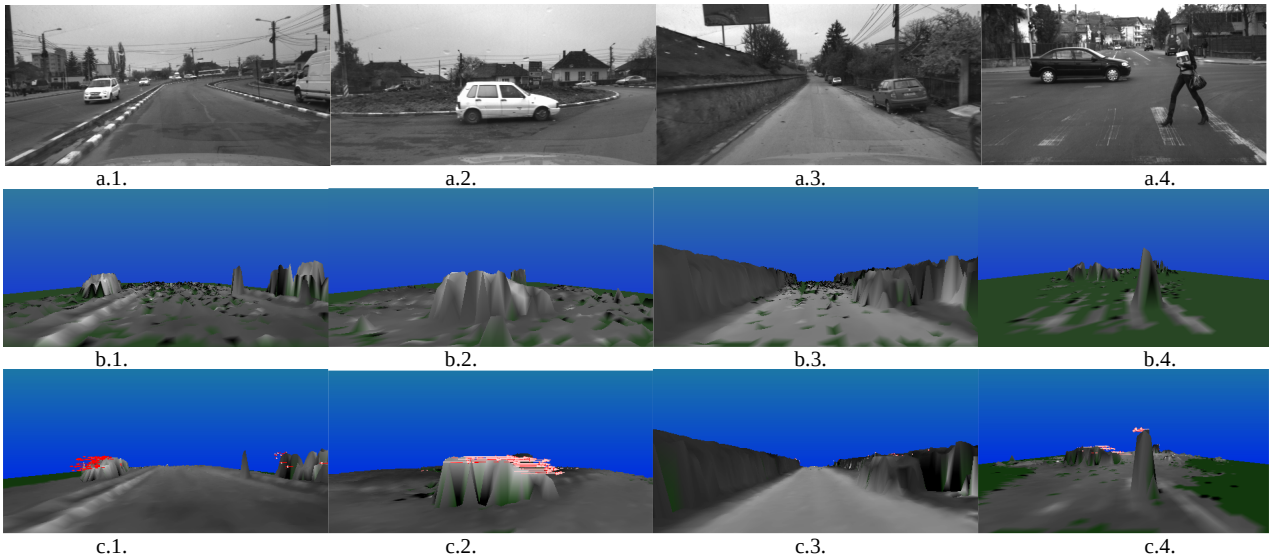

Fig. 8. Qualitative results in real urban traffic scenarios: a.1.-a.4, original grayscale image; b.1.-b.4, raw elevation map with grayscale information; c.1.-c.4, tracked dynamic elevation map.

The capability of the system to estimate the moving objects' speed was evaluated using sequences recorded in controlled scenarios. A test vehicle approaches at a 45 degrees angle from the left, passes through the field of view, and disappears towards our right (Fig. 9). The observed vehicle was recorded four times, with four different speeds: 30, 40, 50 and 60 km/h.


Fig. 9. Test scenario for speed measurement evaluation.

As the scene consists only of the test vehicle and the road, we analyze the speed of the particles that have a height higher than 50 cm, which we'll assume to belong to the obstacle. From the speeds of the particles, the speed of the moving obstacle is estimated. In figure 10, the estimated speed (in km/h) is plotted against the frame number. The system is able to quickly get an estimate of the object's speed, when the object becomes visible, but, as expected, the higher the magnitude of the speed, the more difficult it is to get the particles to converge to the correct value.

In Table I, the root mean square error (RMSE) of the speed estimation is shown. The estimated speed is compared to the ground truth only for the frames where the obstacle is in the field of view. The numbers confirm the results shown in the plots, the decrease of accuracy for

the higher speeds. This behavior is to be expected, as the particle population of a cell is initialized with random speeds from a distribution of zero mean.

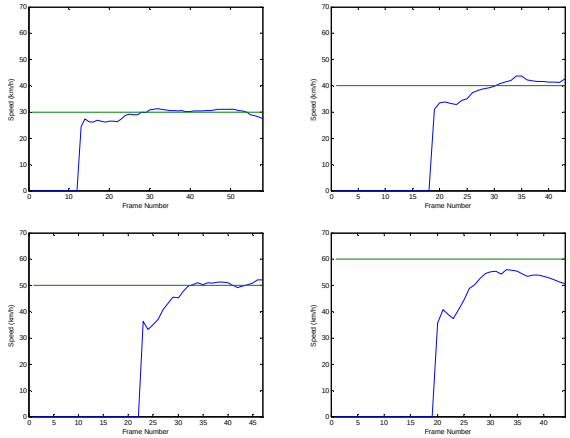

Fig. 10. Estimated object speed, for 30, 40, 50 and 60 km/h ground truth speeds.

TABLE I. SPEED MEASUREMENT EVALUATION.

Ground truth speed (km/h)	***Mean estimated speed (km/h)***	***RMSE (km/h)***
30	29.2650	1.9720
40	38.9354	3.9316
50	46.9964	6.5184
60	50.0729	11.7318

VI. CONCLUSIONS AND FUTURE WORK

This paper describes a flexible method for modeling and tracking complex 3D environments - the particle-based dynamic elevation map enhanced with gray level information. The central element of the solution is the dynamic particle, having position, height, speed and gray value, which can migrate from one map cell to another. A particle population in a cell is a multi-modal probability density approximation for a 4-dimensional state space, an approximation that is updated by controlling the particle population using the cues obtained from measurement. The preliminary results show a system capable of reliably estimating the 3D static and dynamic characteristics of the observed traffic scenes, able to significantly increase the density and accuracy of the raw, stereovision-extracted elevation map, and to add dynamic information for each map cell.

The future work will be focused on increasing the time performance of the solution, as the current system is not yet a real-time algorithm. However, as the system's central concept is the particle, massive parallelization at the particle level and at the cell level can be employed, and the deployment on a CUDA-compatible GPU can lead to real-time performance.

REFERENCES

[1] B. Gassmann, L. Frommberger, R. Dillmann, and K. Berns, "Real-time 3D map building for local navigation of a walking robot in unstructured terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, 2003, pp. 2185-2190, vol.3.

[2] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application," *International Journal of Robotics Research*, vol. 25, pp. 19-30, 2006.

[3] C. Brenneke, O. Wulf, and B. Wagner, "Using 3D laser range data for SLAM in outdoor environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, 2003, pp. 188-193, vol.1.

[4] U. Franke, U. Franke, C. Rabe, H. Badino, and S. K. Gehrig, "6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception," in *DAGM-Symposium*, vol. 3663, ed: Springer, 2005, pp. 216-223.

[5] D. Pfeiffer and U. Franke, "Efficient representation of traffic scenes by means of dynamic stixels," in *IEEE Intelligent Vehicles Symposium (IV 2010)*, 2010, pp. 217-224.

[6] D. F. Huber and M. Hebert, "A new approach to 3-D terrain mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '99)*, 1999, pp. 1121-1127 vol.2.

[7] M. Vergauwen, M. Pollefeys, and L. Van Gool, "A stereo-vision system for support of planetary surface exploration," *Machine Vision and Applications*, vol. 14, pp. 5-14, 2003.

[8] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila, "Autonomous Rover Navigation on Unknown Terrains: Functions and Integration," *International Journal of Robotics Research*, vol. 21, pp. 917-942, 2002.

[9] F. Oniga and S. Nedevschi, "Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 1172-1182, 2010.

[10] M. Harville, "Stereo person tracking with adaptive plan-view templates of height and occupancy statistics," *Image and Vision Computing*, vol. 22, pp. 127-142, 2004.

[11] P. Pfaff, R. Triebel, and W. Burgard, "An Efficient Extension to Elevation Maps for Outdoor Terrain Mapping and Loop Closing," *International Journal of Robotics Research*, vol. 26, pp. 217-230, February 1, 2007.

[12] R. Triebel, P. Pfaff, and W. Burgard, "Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, 2006, pp. 2276-2282.

[13] I. Dryanovski, W. Morris, and X. Jizhong, "Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, 2010, pp. 1553-1559.

[14] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, pp. 46-57, 1989.

[15] C. Chen, C. Tay, C. Laugier, and K. Mekhnacha, "Dynamic Environment Modeling with Gridmap: A Multiple-Object Tracking Application," in 9th *International Conference on Control, Automation, Robotics and Vision (ICARCV '06)*, 2006, pp. 1-6.

[16] M. Kumar and D. P. Garg, "Three-Dimensional Occupancy Grids With the Use of Vision and Proximity Sensors in a Robotic Workcell," *ASME Conference Proceedings*, vol. 2004, pp. 1029-1036, 2004.

[17] H. Lategahn, W. Derendarz, T. Graf, B. Kitt, and J. Effertz, "Occupancy grid computation from dense stereo and sparse structure and motion points for automotive applications," in *IEEE Intelligent Vehicles Symposium (IV2010)*, 2010, pp. 819-824.

[18] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and Tracking the Driving Environment With a Particle-Based Occupancy Grid," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 1331-1342, 2011.

[19] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*: Mit Press, 1993.

[20] M. Isard and A. Blake, "CONDENSATION—Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, pp. 5-28, 1998.