

Tracking Multiple Objects in Traffic Scenarios using Free-Form Obstacle Delimiters and Particle Filters

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

Abstract—Dynamic environment representation is an important research task in the field of advanced driving assistance systems. Usually, the tracking process is influenced by several factors, such as the unpredictable and deformable nature of the obstacles, the measurement uncertainties or the occlusions. This paper presents a stereo-vision based approach for tracking multiple objects in unstructured environments. The proposed technique relies on measurement data provided by an intermediate grid map and the object delimiters extracted from this grid. We present a particle filter based tracking solution in which a particle state is described by two components: the dynamic object parameters, and the object's geometry. In order to solve the high dimensionality state space problem a Rao-Blackwellized Particle Filter is used. The proposed method takes into consideration the stereo uncertainties and relies on a weighting mechanism based on the particle alignment error.

I. INTRODUCTION

Modeling and tracking of dynamic entities is an important research task in the field of driving assistance systems. Typically, the tracking mechanism relies on extracting a set of relevant features from the scene and estimating their state over time. Despite the simplicity of the general idea, the dynamic environment representation remains a challenging problem. Usually, the surrounding world is more complex and the tracking process is influenced by several factors such as the unpredictable and deformable nature of the obstacles, the measurement uncertainties or the occlusions. Considering the above, an environment perception system must be able to track multiple objects at the same time, with high accuracy and confidence.

The tracking systems can be classified by the type of sensors they use. Most techniques rely on the use of ultrasound [4], laser [2] or vision sensors [5][6]. Some of the existing strategies imply directly tracking 3D point clouds [5] by treating each point independently, whereas other motion estimation techniques try to minimize the computational cost by using intermediate representations. The 3D information is transformed into digital elevation maps [14], octrees [4][10], occupancy grids [13] or Stixel Maps[15].

Many of the tracking solutions use high level attributes, including polygonal models [7], difference fronts [8], voxels [10], 2D boxes, 3D cuboids [6] or object contours [11]. Most of them work well in structured environments, where the obstacle's geometry is known. Usually, the traffic entities are represented by simplistic models such as bounding boxes and

the obstacle's position is determined by its center of mass. However, in the case of unstructured environments, it is difficult to use constrained models. The tracking process may lead to incorrect results when the target pose estimation is affected by occlusions or by changes in its geometry. In order to overcome this problem, various algorithms for moving and deforming objects are proposed [16][17]. Typically, the model shape is represented implicitly [17], or by a set of fixed number of points. In particular, the authors in [17] describe a tracking method for slowly deforming and moving contours that are represented implicitly. Isard and Blake [18] propose the CONDENSATION algorithm for tracking parametric curves.

Most often, the object tracking approaches rely on Kalman filters [6], Particle filters [14][16][18] or hybrid methods [20][21]. The traditional Kalman filter represents an optimal estimator in which the posterior distribution is modeled by a Gaussian function. However, the classical Kalman filter solutions are only applicable on linear systems with unimodal distributions. As an alternative, the particle filter approaches approximate the state space by a collection of N discrete samples, called "particles". Each particle represents a hypothesis about the system state. One of the main advantages of the particle filter based solutions is the ability to handle non-linear systems and multi-modal distributions. However, particle filters are not suitable for high-dimensional state spaces as their computational complexity tends to grow exponentially with the number of state parameters. In order to handle this problem, different strategies can be found in the literature. For example, in [19] the Unscented Kalman Filter is used to propagate the proposal distribution so that the number of sampled particles is reduced. In [20] the Rao-Blackwellized Particle Filter (RBPF) is introduced. The key idea of the RBPF approach is that a part of the state space can be updated analytically, while another part of the state is sampled. In [21], the RBPF is applied for Simultaneous Localization and Mapping (SLAM). The robot pose is estimated with a particle filter. In addition, the state vector is represented by N landmarks. Each landmark position is updated by using a 2x2 Extended Kalman Filter (EKF). In [2], a RBPF technique is applied for model based vehicle tracking. For simplicity the vehicle shape is approximated by a rectangle.

In this paper we present a stereo-vision based approach for tracking multiple objects in unstructured environments. The proposed technique relies on measurement data provided by an intermediate grid map and the object delimiters extracted from this grid. Unlike other existing methods which track fixed models, we present a particle filter based solution for tracking freeform obstacle representations. At each step the particle state is described by two components: the object

dynamic parameters and its estimated geometry. We consider that each obstacle model is represented by a polyline with N vertices (control points). In order to solve the high-dimensionality state-space problem a Rao-Blackwellized Particle Filter is used. Therefore, in our case, the obstacle dynamic properties are estimated by importance sampling while the geometric properties are computed analytically by using a Kalman Filter for each key point. The proposed method takes into consideration the stereo uncertainties and relies on a weighting mechanism based on the particle alignment error.

The paper is structured as follows: the next chapter presents the overall system architecture, the object model is described in the chapter III, chapter IV shows how the data association is made, the proposed multiple object tracking approach and its main steps are detailed in the chapter V, while the last two sections show the experimental results and the conclusion about this work.

II. SYSTEM OVERVIEW

The system architecture (see Fig. 1) can be separated into two main stages: Preprocessing and Tracking.

The *Preprocessing* module performs a set of tasks prior to object tracking. First, image pairs are acquired from the two cameras. Then, stereo reconstruction is performed with a dedicated TYZX board [1]. The raw dense stereo information is then used to compute an intermediate classified grid map [12]. Each grid cell is labeled based on its height information as: road, traffic isle or obstacle. The intermediate representation is used to extract object delimiters and to compute a probabilistic measurement model.

The *Tracking* stage consists in estimating the optimal state parameters. First, the data association is performed in order to assign new measurements to the existing trackers and to initialize new ones. Then, for each existing individual tracker the following processing steps are applied: state prediction, Kalman filtering of object geometry, particle weighting, estimation, resampling and injection. These steps will be detailed in the next sections.

III. DATA ASSOCIATION

The data association consists in assigning new observations to the existing targets. In our case we perform the data association by computing overlapping scores w_{ij} between occupancy grid blobs (set of connected cells) at consecutive time steps. As the result, an association matrix $W = \{w_{ij}\}$ is formed. The cases when larger blobs are decomposed into many disjoint parts and vice versa are treated as separate tracking hypotheses. The data association approach is described in details in [3].

IV. OBJECT MODEL

Unlike other methods where objects are represented by fixed templates, we have adopted an approach in which we consider that the object geometry may change over time due to factors such as occlusions caused by crossing obstacles, the dynamic nature of the environment, or noisy

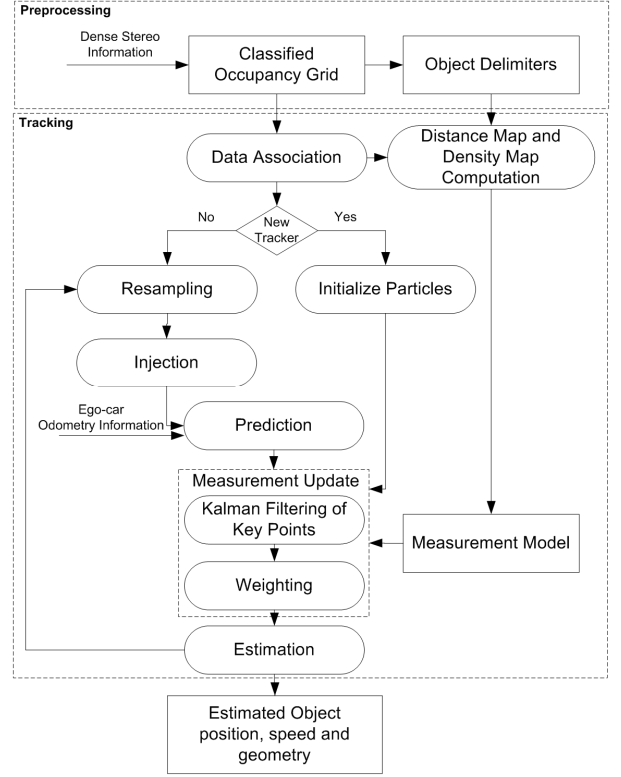


Figure 1. System Architecture

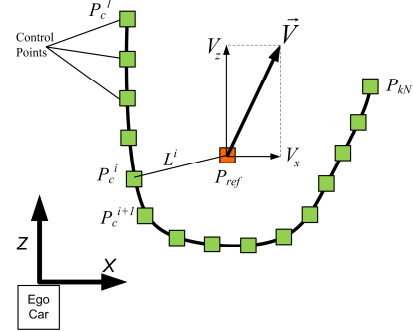


Figure 2. Object Model. An object from the traffic scene is represented by N control points P_c^i . (polygonal vertices), and a reference point P_{ref} .

measurements. Therefore, an object model (see Fig. 2) is described by the following properties:

- A local reference point P_{ref} denoting the obstacle position in the camera coordinate system. The reference point is initially set to the object center of mass, and is subsequently estimated by the tracking mechanism.
- The object speed vector $\vec{V}(v_x, v_z)$
- A set of control points $K = \{P_c^i(x_c^i, z_c^i) | i = [1..N_c]\}$ specifying the object shape, and defining the vertices of a polygonal line. In the initialization step the control points are determined by selecting N equidistant points along the object contour. For each control point $P_c^i(x_c^i, z_c^i)$ we also compute its relative position $L^i(l_x^i, l_z^i)$ to the reference point $P_{ref}(x_{ref}, z_{ref})$.

In our case the used coordinate system has its origin in front of the ego-car. The X axis points to the right while the Z axis points towards the ego-car moving direction. Considering the notations above, at a time t the obstacle state is defined as:

$$S_t = [x_{ref}, z_{ref}, v_x, v_z, l_x^1, l_z^1, l_x^2, l_z^2, \dots, l_x^N, l_z^N]^T \quad (1)$$

V. OBSTACLE TRACKING

A Bayesian solution to the tracking problem consists in estimating the current object state S_t from a set of noisy observations $Z_{1:t} = \{Z_1, Z_2, \dots, Z_t\}$ up to the time t :

$$p(S_t | Z_1, Z_2, \dots, Z_t) = \eta p(Z_t | S_t) \int_{S_{t-1}} p(S_t | S_{t-1}) p(S_{t-1} | Z_1, \dots, Z_{t-1}) \quad (2)$$

where η is a normalization constant. The $p(Z_t | S_t)$ term denotes the *measurement model* at a time t and $p(S_t | S_{t-1})$ describes the *state transition model* (motion model) from S_{t-1} to S_t .

A. Rao-Blackwellized Particle Filter

In a particle-based filtering solution [16][18] the object state probability is approximated by a set of N weighted particles $p(S_t) \approx \{S_t^i, w_t^i, i = [1..N]\}$. Each particle S_t^i represents a hypothesis of the state of the object at a given time t . Therefore, object tracking consists in estimating the best state by evaluating the samples S_t^i and their attached weights w_t^i , given a motion model and a measurement model.

A disadvantage of the classical particle filtering algorithm is that it is not suitable for high-dimensional state spaces. Usually, its computational complexity grows exponentially with the number of parameters. The ‘‘Rao-Blackwellization’’ process consists in estimating a part of the object state analytically, thus reducing the number of dimensions and the computational cost of the particle filter mechanism. In our case, the obstacle state S_t is split into two parts:

$$S_t = [X_t, G_t]^T \quad (3)$$

where the first component $X_t = [x_{ref}, z_{ref}, v_x, v_z]^T$ describes the obstacle position and speed and the component $G_t = [l_x^1, l_z^1, l_x^2, l_z^2, \dots, l_x^N, l_z^N]^T$ denotes the object geometry. The overall posterior distribution $p(X_t, G_t | Z_{1:t})$ defined by Equation (2) is factored as:

$$p(X_t, G_t | Z_{1:t}) = p(X_t | Z_{1:t}) p(G_t | X_t, Z_{1:t}) \quad (4)$$

The first probability distribution $p(X_t | Z_{1:t})$ denotes the object position and velocity, and is approximated by a set of weighted samples $\{X_t^i, w_t^i, G_t^i, i = [1..N]\}$. The second term $p(G_t | X_t, Z_{1:t})$ represents the object geometry posterior conditioned on X_t . Each control point in G_t is described by a mean value and a covariance matrix, $(\hat{L}^j, \Sigma^j)$ estimated by

using a 2x2 Kalman filter. The particle set can be now defined as:

$$\{q_t^i | q_t^i = [X_t^i, w_t^i, (\hat{L}^1, \Sigma^1), \dots, (\hat{L}^j, \Sigma^j)]^T\} \quad (5)$$

where $i = [1..N]$ and $j = [1..N_c]$. Next, we will present the main steps involved in our object tracking solution.

B. Initialization

The initialization step is applied when new (not tracked) candidates are detected. This is achieved by comparing the list of associated blobs with the existing list of individual trackers. First, the motion parameters describing the initial state are estimated by applying a fast pairwise alignment of the associated delimiter pairs (from the previous and current frames). For this, we use the Iterative Closest Point (ICP) algorithm described in [3]. Then, a set of initial random object hypotheses are generated around the measurement position $\{q_0^i | q_0^i = [X_0^i, w_0^i, G_0^i]^T, i = [1..N]\}$. Each particle is initialized with the object geometry G_0 that is extracted from the measurement delimiters. It must be noted that a small amount of new particles (including new hypotheses for object position and geometry) are added in the Injection step.

C. Prediction

This step consists in predicting the current state S_t at time t given the previous information S_{t-1} and the motion model $p(S_t | S_{t-1})$. First, the particles are moved by applying a deterministic drift based on the target dynamics. Then, each predicted sample state is altered according to a random noise.

We also must take into account the ego-vehicle motion in order to extract its speed from the independent dynamics of the tracked objects. In our case, the vehicle speed v and the yaw rate $\dot{\psi}$ are obtained from the car sensors. By following the ego-vehicle motion model with constant yaw rate and constant speed, the particle positions (x_{ref}, z_{ref}) are transformed according to:

$$\begin{bmatrix} x_{ref-c} \\ z_{ref-c} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} x_{ref} \\ z_{ref} \end{bmatrix} - \begin{bmatrix} \frac{v\Delta t}{\psi} (1 - \cos \psi) \\ \frac{v\Delta t}{\psi} \sin \psi \end{bmatrix} \quad (6)$$

where Δt is the time delay between two frames and $\psi = \dot{\psi}\Delta t$ represents the vehicle rotation angle around the Y axis.

The position and velocity $X_t = [x_{ref}, z_{ref}, v_x, v_z]^T$ of each particle is predicted by using the standard constant velocity model:

$$\begin{bmatrix} x_{ref} \\ z_{ref} \\ v_x \\ v_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ref-c} \\ z_{ref-c} \\ v_x \\ v_z \end{bmatrix} + w \quad (7)$$

The matrix multiplication describes the deterministic drift component. The stochastic part is defined by the random

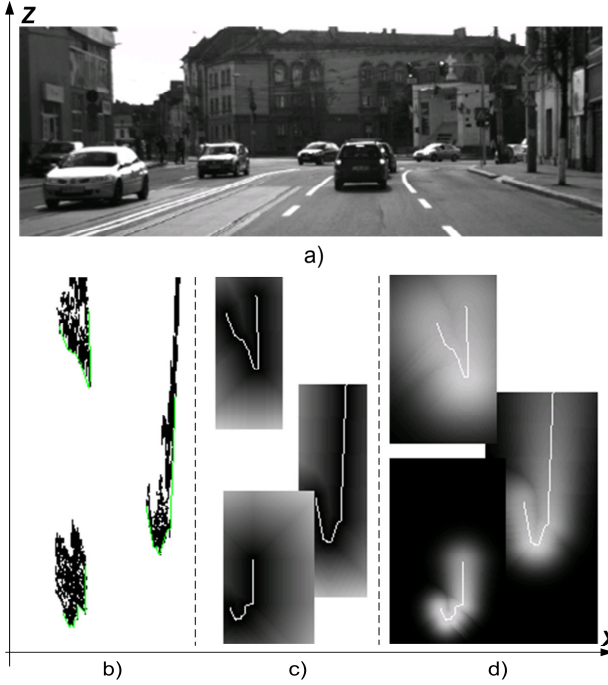


Figure 3. a) Left camera image. b) The occupancy grid projected on the ground plane. The obstacle delimiters are colored with green. c) The Distance Transform of the extracted delimiters. d) The density map is generated by taking into account stereo uncertainties and distances to the closest delimiter points. High intensities indicate high measurement probability.

noise $w \sim N(0, Q)$ which is drawn from a zero mean Gaussian distribution with covariance Q . The covariance matrix Q is estimated considering a certain covariance for the obstacles' acceleration.

D. Measurement update

The purpose of this stage is to assign new weights to the predicted particles based on the measurement model. First, the object delimiters are extracted from the current occupancy grid. Then, the new object geometry is updated by using a Kalman filter. Finally, new particle weights are computed by evaluating the alignment error between the measurement and existing hypotheses.

1) *Extracting Object Delimiters*: At each frame, object delimiters are extracted from the occupancy grid (see Fig. 3) by using the *BorderScanner* algorithm that is described in [9]. The main idea of the *BorderScanner* approach is to generate an object contour $C_{measurement}$ by selecting the most visible object cells c_i . This is achieved by considering a virtual ray which extends from the observation point and moves in a radial direction with fixed increments. At each step the closest occupied grid cell $Occ(c_i) = true$ is chosen as the delimiter point.

$$C_{measurement} = \{c_i \mid Occ(c_i) = true, i \in [1..M_c]\} \quad (8)$$

2) *Computing Stereo Uncertainties*: The next step is to compute the stereo uncertainties. If we consider that the stereo-vision system is rectified, then for each grid cell we can estimate a depth error σ_z and a lateral error σ_x :

$$\sigma_z = \frac{z^2 \cdot \sigma_d}{b \cdot f}, \quad \sigma_x = \frac{\sigma_z \cdot x}{z} \quad (9)$$

Where x and z are the real world coordinates of a point, f denotes the focal length, b is the stereo system baseline and σ_d represents the disparity error.

3) *Kalman Filtering*: This step consists in updating the particle geometry component G_t^i with the new measurements. For each control point we use a 2x2 Kalman filter to estimate its state $\hat{L}^j = [l_x^j, l_z^j]^T$ and covariance Σ^j . The Kalman filter observations are determined by selecting N equidistant points along the object delimiter extracted in the step 1. For each control point, the measurement covariance matrix R is computed, by considering the stereo uncertainties defined in the step 2.

4) *Computing the distance to the measurement*: The aim of this step is to determine the closest corresponding measurement points for each occupancy grid cell. First, we define a region of interest that covers all the particle space around the measurement $C_{measurement}$. Then we compute a modified Distance Transform (see Fig. 3.c). Each point $m_{dm}(x_{dm}, z_{dm})$ from the distance map will be described by two values: a distance $d_m = \sqrt{(x_{dm} - x_{del})^2 + (z_{dm} - z_{del})^2}$ to the nearest delimiter point $c_j(x_{del}, z_{del})$, and a position of the respective correspondence (x_{del}, z_{del}) . The probability density map (see Fig. 3.d) can be determined now for each cell m_{dm} by converting the distance values according to:

$$\pi_{xz} = \frac{1}{2\pi\sigma_x\sigma_z} e^{-\frac{1}{2} \left[\frac{(x_{dm} - x_{del})^2}{\sigma_x^2} + \frac{(z_{dm} - z_{del})^2}{\sigma_z^2} \right]} \quad (10)$$

where σ_x and σ_z represent the stereo uncertainties of the corresponding measurement point.

5) *Weighting*: This step consists in assigning new weights w_t^i to the delimiter hypotheses q_t^i based on their likelihood:

$$p(Z_t \mid X_t = X_t^i, G_t = G_t^i) \quad (11)$$

First, we need to define a distance metric between a given particle and a given observation. This is achieved by estimating an alignment error between object hypotheses and the measurement data. For each control point L^j from the particle q_t^i we determine the closest corresponding point c_k from the measurement $C_{measurement}$:

$$d(L^j, C_{measurement}) = \min_{k \in [1..N_c]} d(L^j, c_k) \quad (12)$$

In order to consider the stereo uncertainties we also assign a density value π_L^j to each corresponding pair (L^j, c_k) . The Euclidean distance $d(L^j, C_{measurement})$ and the weight π_L^j metrics are determined by superimposing the

particle model on the two maps estimated in the previous step. The alignment error is computed according to:

$$D_{\text{alignment}} = \frac{\sum_{j=1}^{N_s} \pi_L^j \cdot d(L^j, C_{\text{measurement}})}{\sum_{k=1}^{N_s} \pi_L^k} \quad (13)$$

Finally, the overall particle weight w_t^i is computed:

$$w_t^i = \frac{1}{2\pi\sigma_D} e^{-\frac{1}{2} \frac{D_{\text{alignment}}^2}{\sigma_D^2}} \quad (14)$$

E. Estimation

The current mean state at time t is estimated by using a weighted average:

$$\hat{S} = \sum_{i=1}^N w_t^i S_t^i \quad (15)$$

F. Resampling and Injection

The resampling step consists in drawing from the previous particle set with a sampling probability proportional to the assigned weights. Thus, the particles with low importance are removed while the samples with large weights are replicated.

However, there are cases when sharp changes in the traffic scene may lead to the estimation of erroneous states. This may happen due to the fact that there are no sufficient hypotheses in the vicinity of the true state (particle deprivation problem). As the solution, we introduced an Injection step where a small amount of particles with low importance are replaced with new completely random samples that are drawn around the measurement. Through the Injection step we also introduce new hypotheses for object geometry.

VI. EXPERIMENTAL RESULTS

The proposed object tracking approach was tested on various sequences of urban traffic scenes, including partially visible obstacles of different type, size and shape. We performed our experiments on a computer equipped with an Intel Core 2 Duo E6750 CPU at 2.66GHz and 4GB of RAM. The occupancy grid used in our solution has a resolution of 240 rows x 500 columns (0.1 m x 0.1 m cells).

Fig. 4 shows the results of the proposed object tracking method, including intermediate steps and the final results. In Fig. 4.b is presented a case when the initialization step is applied for all objects. Usually, this occurs at the beginning of a traffic sequence, when the list of individual trackers is empty. Fig. 4.c illustrates the same test scenario, two frames later. It can be observed that particles are clustered around each individual tracked object. The estimated mean state is colored with light blue. The predicted samples are colored with magenta. The picture also illustrates the influence of weighting and resampling steps (dark blue) on the predicted population of particles. Fig. 4.d shows a particular case when the initialization step is applied to a newly detected tracker. A set of initial random object hypotheses (red color) are generated around the measurement position. In Fig. 5.e is

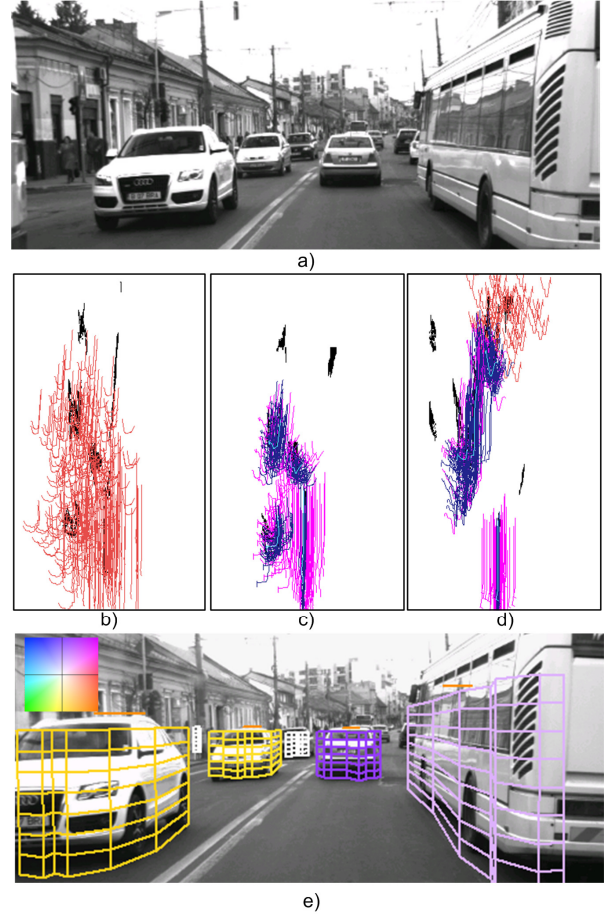


Figure 4. Multiple Object Tracking. a) An image of a traffic scene. b) Particle Filter Initialization step. c) Individual object tracker particles. d) The initialization step is applied to a new detected tracker. e) The resulting dynamic object representation.

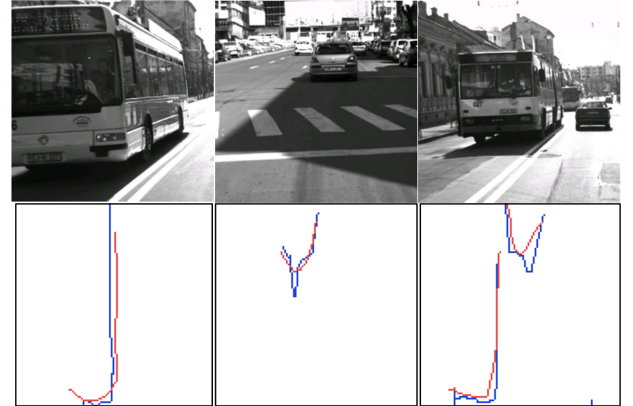


Figure 5. Comparative results between the measurement delimiter (blue) and the analytically tracked geometry (red).

shown the resulted dynamic object representation based on the extracted object delimiters. The object height properties are inherited from the occupancy grid blobs and are used as an additional cue when generating 3D polygonal models. Each detected obstacle is color coded, the color hue describing the orientation of a moving obstacle, while the saturation describes its speed (e.g. yellow – for incoming

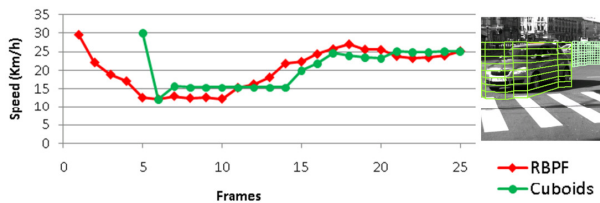


Figure 6. Object speed estimation. Comparison between RBPF based motion estimation (red color) and a cuboid based tracking method (green color).

objects, blue – for outgoing objects). Fig. 5 presents comparative results between the measurement delimiter (blue) and the analytically tracked geometry (red). Fig. 6 shows a comparative result between the RBPF based object motion estimation method and a Kalman filter based cuboidal tracking solution [6]. The test implies a crossing vehicle that is in the field of view for only 25 frames. It can be observed that the proposed RBPF tracking technique is able to provide the speed estimation results earlier, compared with the cuboidal based tracking method which uses a preselection step in order to validate a new tracker. The time performance of the algorithm depends on the number of tracked objects, the number of used particles per object and the number of control points per model. The algorithm complexity scales linearly with the number of trackers. In our tests sequences, the average number of tracked objects was 6. For each object we set up a fixed number of 80 particles and a fixed number of 20 control points per sample. The average processing time of the algorithm was about 99.83ms per frame.

VII. CONCLUSIONS

In this paper we proposed a stereo-vision based approach for tracking multiple objects in urban traffic scenarios. The solution is based on the information provided by a classified occupancy grid. Unlike the other existing methods that consist in tracking fixed models, we propose a particle filter based solution for tracking free-form obstacle representations. At each step the particle state is described by two components: the object dynamic parameters, and its estimated geometry. In order to solve the high-dimensionality state-space problem a Rao-Blackwellized solution is used, where the obstacle dynamic properties are estimated by importance sampling while the geometric properties are computed analytically by using a Kalman Filter for each key point. Inspired by the laser based scan matching techniques such as Iterative Closest Points (ICP) algorithm, we developed a weighting mechanism that relies on evaluating the particle alignment error by finding point-to-point correspondences between the particle model and the measurement. The presented probabilistic tracking solution takes into consideration the stereo uncertainties introduced by the sensorial system. As future work, we propose to improve the accuracy of our solution by including the intensity information, as in the optical flow techniques. We also intend to improve the system processing time by further optimizations.

ACKNOWLEDGMENT

This work was supported by the Romanian Ministry of National Education, UEFISCDI, project number PNII-ID-PCE 344/2012

REFERENCES

- [1] J. I. Woodhill, G. Gordon, and R. Buck, "Tytx deepsea high speed stereo vision system". In *Proc. of IEEE Computer Society Workshop on Real Time 3-D Sensors and Their Use*, Washington, DC, 2004.
- [2] A. Petrovskaya and S. Thrun, "Model based vehicle tracking for autonomous driving in urban environments". In *Robotics: Science and Systems (RSS)*, 2008.
- [3] A. Vatavu and S. Nedevschi, "Real-time modeling of dynamic environments in traffic scenarios using a stereo-vision system", in *Proc. of IEEE ITSC 2012*, pp.722-727, 16-19 Sept. 2012.
- [4] N. Fairfield, G. A. Kantor, and D. Wettergreen, "Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels", in *Journal of Field Robotics*, 2007
- [5] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6d-vision: Fusion of stereo and motion for robust environment perception," in *27th Annual Meeting of the German Association for Pattern Recognition DAGM '05*, 2005, pp. 216-223.
- [6] R. Danescu, S. Nedevschi, M.M. Meinecke, and T. Graf, "Stereo-vision Based Vehicle Tracking in Urban Traffic Environments", in *Proc. of the IEEE ITSC 2007*, Seattle, USA, 2007
- [7] C.C. Wang, C. Thorpe, M. Hebert, S. Thrun, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking", in *The International Journal of Robotics Research*, 26(6), June 2007.
- [8] A. Vatavu, R. Danescu, S. Nedevschi, "Real-time dynamic environment perception in driving scenarios using difference fronts", in *Proc of IEEE IV 2012*, 3-7 June 2012, pp.717-722.
- [9] A. Vatavu, S. Nedevschi, and F. Oniga, "Real Time Object Delimiters Extraction for Environment Representation in Driving Scenarios", In *Proc. of ICINCO-RA 2009*, Milano, Italy, 2009, pp 86-93.
- [10] A. Azim, O. Aycard, "Detection, classification and tracking of moving objects in a 3D environment," in *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, 3-7 June 2012, pp.802-807
- [11] S. Prakash, S. Thomas, "Contour tracking with condensation / stochastic search", In *Dept. of CSE. IIT Kanpur*, 2007.
- [12] F. Oniga and S. Nedevschi, "Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection", in *IEEE TVT*, Vol. 59, No. 3, March 2010, pp. 1172-1182.
- [13] A. Elfes, "A Sonar-Based Mapping and Navigation System", in *Proc. of IEEE ICRA*, April 1986, pp. 1151-1156
- [14] R. Danescu, F. Oniga, S. Nedevschi, M-M. Meinecke, "Tracking Multiple Objects Using Particle Filters and Digital Elevation Maps", in *Proc. of the IEEE-IV 2009*, Xi'An, China, June 2009, pp. 88-93.
- [15] D. Pfeiffer and U. Franke, "Efficient Representation of Traffic Scenes by Means of Dynamic Stixels", in *Proc. of IEEE Intelligent Vehicles Symposium (IEEE-IV)*, 2010, pp. 217-224.
- [16] Y. Rathi, N. Vaswani, A. Tannenbaum, A. Yezzi, "Particle filtering for geometric active contours with application to tracking moving and deforming objects", in *Proc. of the IEEE Computer Vision and Pattern Recognition, CVPR 2005*, Vol. 2, 20-25 June 2005, pp. 2-9.
- [17] J.D. Jackson, A.J. Yezzi, S. Soatto, "Tracking deformable moving objects under severe occlusions", in *Proc. of 43rd IEEE Conference on Decision and Control, CDC*, Vol.3, 2004, pp.2990-2995
- [18] M. Isard and A. Blake. Condensation, "Conditional density propagation for visual tracking", in *International Journal of Computer Vision*, 29(1):5-28, 1998.
- [19] R. Merwe, A. Doucet, N. Freitas, E. Wan, "The Unscented Particle Filter", in *Advances in Neural Information Processing Systems*, vol. 8, 2000, pp 351-357
- [20] A. Doucet, N. De Freitas, K. Murphy, S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks" in *Proc. of the Sixteenth conference on Uncertainty in artificial intelligence*, 2000, pp. 176-183.
- [21] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem", in *Proc. of the AAAI National Conference on Artificial Intelligence*, pp. 593-598, 2002.