PartID – Individual Objects Tracking in Occupancy Grids Using Particle Identities

Radu Danescu
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
radu.danescu@cs.utcluj.ro

Razvan Itu
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
razvan.itu@cs.utcluj.ro

Mircea Paul Muresan
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
mircea.muresan@cs.utcluj.ro

***Abstract*— Occupancy grid tracking algorithms see the world as made out of cells that can be either free or occupied, and can have speed probability densities attached to each cell. These algorithms estimate the overall state of the environment based on sensor data, but they are not aware of, nor concerned with the identity of individual objects. This paper proposes a new approach for individual objects tracking using the dynamic occupancy grids, which will embed the identity of the objects in the grid state. The particle based dynamic occupancy grid is extended by attaching identity information to each particle, thus achieving individual object tracking at grid level without the need of explicit modeling of the object's shape. The position and dynamics of the world occupied cells are tracked independently of their identity, by the mechanism of the particle based occupancy grid. For achieving individual object tracking, this mechanism is extended with components for assigning and managing the identity of the particles. The designed system was tested on real world sequences, and was able to successfully track obstacles found on the road without making assumptions about their nature, shape or size.**

Keywords—object tracking, occupancy grid, particle filtering

I. INTRODUCTION

Tracking means continuous estimation of a system's state based on a sequence of measurements, sometimes called observations. When the system consists of multiple elements that can move independently, tracking also means following each element, and preserving its identity throughout the time it is observed. In robotics or driving assistance, the concept of tracking covers both objectives: each individual object of interest is assigned an identity, and the parameters of this object are estimated in time based on the sequence of measurements. The objects have a state, which describes static properties (size, position, shape, weight, etc.) and dynamic properties (speed, acceleration, etc.). The state of the object is not known directly, and has to be estimated based on imperfect measurements. All tracking solutions have to have a way of representing and handling the uncertainties in the object's state and the errors in measurement, in order to produce the best possible estimate. Models enforcing a specific object shape, or a specific type of movement will impose constraints on the estimation process and filter out the errors, but these models may impose rigid conditions that not all objects will satisfy, and therefore may cause errors themselves.

When tracking multiple objects, the problem of state estimation is complicated by the problem of knowing which tracked object is associated to each measurement. Assigning the wrong measurement to a track may lead to a false estimation of parameters, or, even worse, may cause the tracking system to confuse targets.

Most tracking solutions see the world as built from individual objects of specific geometry, and they focus on detecting these objects individually in each frame, and then track them by creating persistent objects from the first measurements and updating them with the subsequent measurements. An alternative to this view is the occupancy grid, which sees the world as made out of cells that can be either free or occupied. The dynamic occupancy grids allow a speed probability density to be associated to each grid cell, and can be used to track a dynamic world without concerning with individual objects. The dynamic occupancy are highly flexible, being able to track complex dynamic scenes such as the ones encountered in urban traffic, and can use as input data scans from active sensors such as laser or radar based, 3D points from stereovision, or even monocular images segmented by a convolutional neural network.

The occupancy grid based algorithms estimate the state of the environment based on a sequence of measurements, but they are not aware of the identity of individual objects. If individual object tracking is required, the cells of the occupancy grid can be clustered, and a geometric object model can be fit to the cell clusters. Unfortunately, this means that if objects with specific identity have to be tracked, another algorithm, such as a Kalman filter tracker, will have to be applied on top of the grid's results.

This paper proposes a new approach for individual objects tracking using the dynamic occupancy grids, which will embed the identity of the objects in the grid state. The dynamic occupancy grid addressed in this paper is the particle-based occupancy grid, which models occupancy inside a cell by a population of particles having position and speed, which can migrate from one cell to another to achieve state prediction, and can be deleted or multiplied to achieve state update based on measurement. The particle approach to dynamic grids allows us to attach object identity to particles, thus achieving individual object tracking at grid level without the need of explicit modeling of the object's shape. The position and dynamics of the world occupied cells are tracked independently of their identity, by the mechanism of the particle based occupancy grid, and this mechanism is not changed by adding identity to the particles. Thus, in order to achieve complete individual object tracking based on the particle grid results, the only thing we have to do is to manage the mechanism of assigning and maintaining the identity of the particles.

II. RELATED WORK

Multi object tracking (MOT) is a necessary enabling technology, which can be applied in very many settings and

978-1-7281-9080-8/20/$31.00 ©2020 IEEE

scenarios, in the field of robotics, autonomous driving and advanced driver assistance systems. Multiple papers in the literature tailor the target tracking algorithms for a variety of sensors like single cameras, stereo cameras, LIDARs, RADARs, thermal cameras or a combination of them.

The most popular approaches for object tracking are based on the Kalman filter [1], which represents the target state as a Gaussian probability density with mean and covariance matrix. The evolution of the state in time, and the observation process, are assumed to be linear, which allows for a very computationally efficient implementation. If the transition between the states, or the relation between measurement and state, are not linear, the Extended Kalman Filter (EKF) [2], can linearize the process functions or the measurement functions using the Taylor expansion. A more complex approach, the Unscented Kalman Filter [3] is also able to recover the Gaussian density and linearize the process or measurement models by propagating a set of sigma-points through the functions that need linearization.

The main drawback of these solutions is the uni-modal probability density assumed for the target state, meaning that they require a clear association between the track and the measurement. A false association, even for a single frame, can lead the tracker astray. For this reason, the association problem is usually modeled as a graph with cost links between measurements and tracks, as in [4], and significant effort is spent on finding the best cost functions using 2D or 3D features, as in [5][6][7]. Based on this graph, the association problem is usually solved using the Hungarian algorithm, as seen in [7].

Recently, due to the rising popularity of convolutional neural networks and the increase of processing power, more and more tracking solutions use a neural network to solve the track-measurement correspondence problem [8][9]. In [10] the authors use Siamese convolutional neural networks to extract deep features, and then using a cross-correlation layer, they perform an efficient sliding -window evaluation in the search region to find the best association. The patches having the same size as the target get a similar score and the one that obtains the highest score is identified as the new target location. The paper presented in [11] approaches the data association requirement and the identity maintenance issue in two different stages. For this purpose, the authors train two Siamese AlexNet [12] CNNs for feature extraction and fuse the results of the networks for the final decision regarding a tracked object.

The Particle filter is an algorithm that models an underlying distribution by a set of weighted particles, allowing to drop any assumption regarding the distribution itself [13][14]. Each particle is a complete target state description, allowing a multi-modal state representation. The state uncertainty is given by the distribution of the particles in the given state space. Similarly to the Kalman filters, the future states are predicted using motion models and the current state. The main differences are that the state representation is multi-modal, and that a clear association between the track and the measurement is not required.

Some of the most common sources of errors in tracking are the occlusions or partial observations. In order to cope with partially visible targets, the part-based particle filter was proposed in [15]. This solution decomposes the target in several parts, each having its own state characterized by multiple particles. The components that make up the target are strongly correlated using pre-trained probabilistic appearance and geometric models. In [16] the appearance model and the number of particles are adjusted dynamically. The method presented in [17] proposes a part-based filter for visual tracking which accurately captures the relation between the local parts. The authors use a top-down coarse-to-fine localization and a bottom-up adaptive update to increase the tracking robustness. Part based particle filters demonstrate a superior robustness in handling partial occlusion and deformations, due to their ability to divide the target into several local parts and track each of them individually.

Occupancy grids see the world as a bird's eye view of cells that can be either occupied or free, with a certain degree of probability. Initially static [18], the grids were extended to also track dynamic environments [19][20], by incorporating a speed probability density into the grid cells. These solutions can be seen as multi-object trackers as they estimate the position and speed of obstacle parts, but they don't assign identity to individual objects. The solution presented in this paper extends [20] to create and manage object identities within the grid itself.

III. Algorithm Description

A. Overview

This work is based on the dynamic particle-based occupancy grid described in [20], which models and tracks the 3D world by means of mobile particles. The mechanisms of the particle-based occupancy grid takes care of predicting and updating the occupancy of the cells, and estimating their speed probability density. In order to track individual objects, each particle is assigned a unique tracking ID, which will represent the object the particle is part of. At each frame, the following steps are performed:

a) Connected components labeling of occupied grid cells, to identify the objects of the current frame;

b) Many-to-many association between current objects and the particle tracking IDs;

c) Deactivation of the tracks that are too spread out, if they are associated to multiple objects;

d) Creation of new tracks from free objects;

e) Update of the tracking IDs of each particle;

f) Identification of tracked objects for output.

Each step will be detailed in the following sections.

B. The particle-label-track model

The dynamic environment is represented by particles having position and speed, which can move from one grid cell to another based on their speed vector. The number of particles in a cell indicates the probability of the cell to be occupied or free. For tracking individual objects, another parameter will be assigned for each particle: the tracking ID t. Now each particle π in the particle set S will be defined by its grid row r, grid column c, row speed v_r, column speed v_c, age since creation a, and tracking ID t:

$$S = \{\pi | \pi = (r, c, v_r, v_c, a, t)\} \quad (1)$$

The problem of tracking individual objects becomes the problem of assigning the proper tracking IDs to the particles.

An individual object identified by the unique tracking identifier τ will be made out of the particles of the set S_τ:

$$S_\tau = \{\pi | \pi \in S, \pi.t = \tau\} \quad (2)$$

After performing the steps of the particle based grid tracking algorithm (prediction by moving particles from one cell to another based on their speeds and the uniform motion model, update by multiplication and deletion based on measurement data), the densely populated cells are grouped into cell clusters using connected component labeling. These clusters can then be approximated by oriented cuboids, but this step is not necessary for the proposed tracking algorithm. In Fig.1 we see some steps leading to the cell clusters: finding the obstacle areas boundaries, continuous estimation of the cell occupancy by means of particles, and labeling of the most occupied cells.

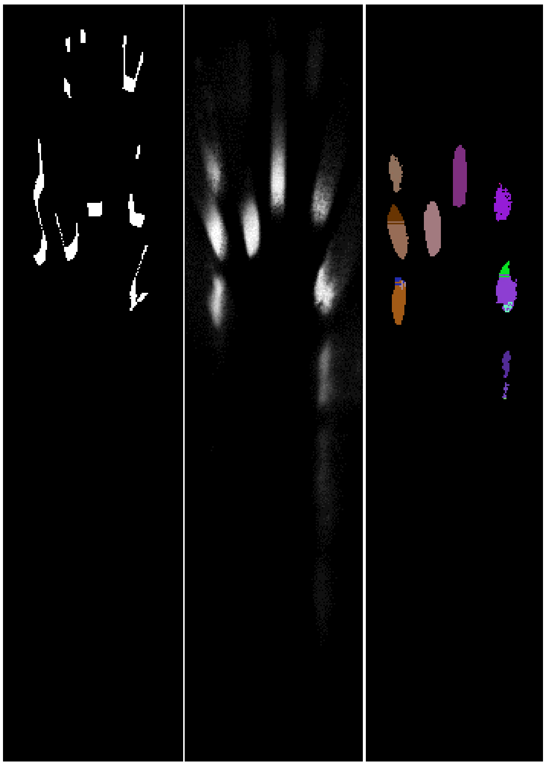

Fig. 1. Identifying connected occupied components in the grid. Left: polylines identifying obstacle boundaries from raw measurement data; middle: tracked occupancy of the cells; right: connected occupied components.

After identifying the connected components of the occupancy grid, we'll have the label grid $L(r, c)$, with a label assigned for each row r and column c. The labels are temporary, for a single frame.

A single particle can therefore be associated to a track τ and a label l. The set of particles for a particular combination (τ, l) is defined as:

$$S_{\tau,l} = \{\pi | \pi \epsilon S, \pi.t = \tau, L(\pi.r, \pi.c) = l\} \quad (3)$$

The cardinality of the set $S_{\tau,l}$ will be stored in the matrix M. This matrix, indexed by tracks on rows and by labels on columns, will be used as association measure between labels and tracks.

$$M(\tau, l) = |S_{\tau,l}| \quad (4)$$

If we set τ=0, the value of the matrix $M(\tau = 0, l)$ will tell us how many particles are located on a given label l, but are not associated to any track. If the value of l is set to zero, $M(\tau, l = 0)$ tells us how many particles are associated to the track τ, but have no label in the current frame.

C. Tracking algorithm description – updating existing tracks

The system maintains a list of active tracks that can be updated by new measurements. The actual state of the tracks is maintained by the particles associated to them by the track ID property. The position and speed of each particle are already updated from the particle occupancy grid tracking algorithm, therefore the individual trackers themselves are not concerned with position and motion parameters. The problem of track update becomes the problem of assigning particles to tracks – finding the value of $\pi.t$ for each particle π.

An optimal approach would be to compute for each particle a probability of belonging to a track, based on the particle's position, speed, past track ID and age. However, given that the particles are numbered in millions, the optimal approach will be very slow. Instead, a sub-optimal approach will be used, based on the label associated to the particle's position.

Based on the label grid and the position of the particle, the label of the particle is $L(\pi.r, \pi.c)$. Using the label to track association matrix M, the probability of a particle to belong to a track τ is computed as:

$$P(\pi.t \leftarrow \tau) = \frac{M(\tau, L(\pi.r, \pi.c))}{\sum_t M(t, L(\pi.r, \pi.c))} \quad (5)$$

Based on (5), the probabilities of association between a particle and all active tracks are computed. The track that is associated to the particle is selected randomly, based on the probability of association.

If a particle does not have a label, meaning that $L(\pi.r, \pi.c) = 0$, the particle's tracking ID will be kept unchanged. This is equivalent to the classical object tracking situation when there are no available measurements and the state of the object is given by the prediction. The only exception to this rule is when the track associated to the particle has been deactivated, as described in the next section. If the particle's track has been deactivated, the tracking ID of a zero label particle will be set to zero.

D. Deactivating the tracks

The tracks are deactivated naturally when they run out of particles, either because the particles are reassigned to other tracks or because the particles themselves become deleted due to lack of measurement data, or as they pass beyond the limits of the grid.

There is, however, a situation when the tracks must be deactivated even though they have plenty of particles. Due to the fact that particles having a single track ID can move in different areas of the grid based on their speeds, if the speeds are not convergent the track ID can spread around a large area. This case is shown in Fig. 2: two distinct groups of grid cells have the same tracking ID (the color in Fig. 2.a). If the cells are grouped based on their ID to generate individual objects, these objects will be abnormally large and will contain a lot of free space, as seen in Fig.2.b and Fig.2.c.

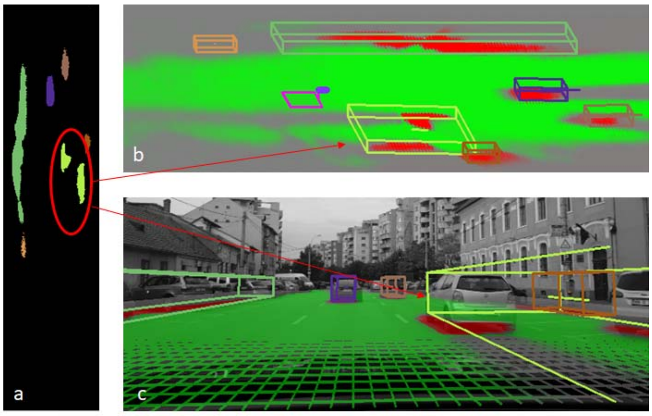

Fig. 2. Left unchecked, some tracks will spread over more cell groups: a – grid cells colored by the dominant tracking ID of their particles; b – cuboid objects extracted from cells with the same tracking ID, cabinet projection; c – cuboid objects extracted from cells with the same tracking ID, perspective projection.

In order to prevent this situation, the spread of the tracks on multiple labels is monitored. First, we define the dominant label for the track τ, $\lambda(\tau)$, as the label that contains the largest amount of particles that have the tracking ID τ. This label can be found using the association matrix M:

$$\lambda(\tau) = \arg\max{}_{l} M(\tau, l) \quad (6)$$

Based on the dominant label, the compactness of the track is defined as the ratio between the number of track particles associated to $\lambda(\tau)$ and the total number of particles of track τ.

$$k(\tau) = \frac{M(\tau,\lambda(\tau))}{\sum_l M(\tau,l)} \quad (7)$$

A track that is not spread out on multiple groups should have the compactness close to 1. A compactness that has a value of less than 0.5, indicating that more than 50% of the track's particles are outside of the dominating label, will signal a track that has spread out too much and needs to be deactivated. When a track is deactivated, all its particles will receive a tracking ID of zero, and its corresponding values in the association matrix, $M(\tau, l)$ will be added to $M(0, l)$, signaling that the number of free particles for label l is increased.

A list of deactivated tracks will be maintained for each measurement cycle. Due to the fact that these tracks often include valid objects, the system will try to re-activate them using one of the multiple measurement labels they span, so that the continuity of the tracking process can be maintained.

E. Creation of new tracks

New tracks are created from labels that have a higher degree of independence from existing tracks. Intuitively, we can consider labels that are not yet associated to tracks to be new objects. However, due to the fact that particles with non-zero tracking IDs can spread all over the map, they can be located on any labeled cell, even if this cell is not in the close proximity of the track dominated cells. Therefore, the condition of a label to be free from track association must become non-binary.

Based on the association matrix M, we can define the freedom factor of a label, $f(l)$, as:

$$f(l) = \frac{M(0,l)}{\sum_\tau M(\tau,l)} \quad (8)$$

If a label is not touched by any particle belonging to a track, $f(l)$ will ideally be 1. If a label is completely contained inside one or more tracks (meaning all particles that have the position on grid cells with label l are marked with non-zero tracking ID), the freedom factor f(l) will be zero.

If $f(l) > 0.5$, meaning that 50% of the particles associated to label l are not tracked, a new track is created. A new tracking ID, τ_{NEW}, is generated, and the track-label association matrix is updated:

$$M(\tau_{NEW}, l) = M(0, l) \quad (9)$$

The new track is not attached directly to the particles. The steps presented in section III.C, based on the association probability computed by means of (5), are performed uniformly, at the end of the track cycle, for all tracks, old and new.

F. Recovering deleted tracks

If a track is deleted due to low compactness, it can be re-initialized in the same frame, without losing the identity of the object. The first step is to define, for each label l, the track with the strongest association, $T(l)$.

$$T(l) = \arg\max{}_{\tau} M(\tau, l) \quad (10)$$

If the track $T(l)$ becomes deactivated due to low compactness, it will be marked as so. When a new track τ_{NEW} is created based on the current label l, the system checks whether the strongest associated track $T(l)$ is deactivated. If this is the case, a reassignment table is used to create the connection between the newly created track and the deactivated one:

$$\rho(\tau_{NEW}) = T(l) \quad (11)$$

A deleted track can be reassigned only once, otherwise it may be the strongest associated track to more than one label, and the track that spans more than one object will be created again. If for a given label, its $T(l)$ is deleted but has already been reassigned to a different label, the newly created track τ_{NEW} maintains its new identity.

When, at the end of the tracking cycle, the particles are assigned tracking IDs, the following assignment is performed if the selected track for assignment is a newly created track:

$$\pi.t = \begin{cases} \rho(\tau_{NEW}), & if\ \rho(\tau_{NEW}) > 0 \\ \tau_{NEW}, & otherwise \end{cases} \quad (12)$$

G. Object reconstruction

After each tracking cycle, the tracked particles can be grouped into objects. Each cell in the occupancy grid is labeled with the ID of the dominant track of the cell (the track that has the most particles in that cell), and similar cells are grouped into objects, which, for visualization purposes only, are represented as oriented cuboids.

One of the problems that we have encountered is that sometimes tracks tend to diffuse into others. Basically, isolated cells of one track can be found inside the cell cluster of another track, leading to superimposed or nested objects, as seen in Fig.3.

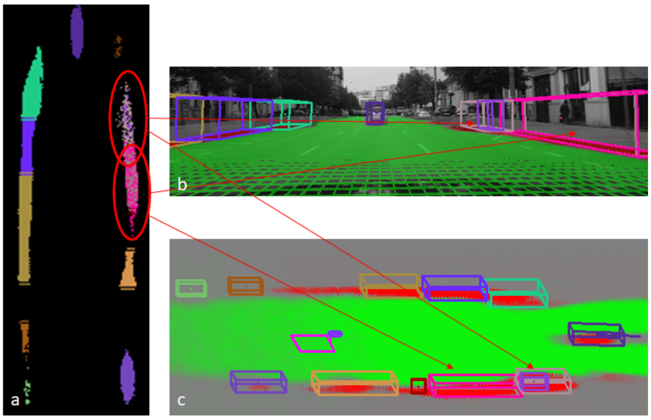

Fig. 3. Diffusing tracks: some track clusters may include cells dominated by other tracks, leading to superimposed objects (pink and brown labels) or nested objects (purple and beige). a – track labeled grid cells, b – perspective view of resulted cuboids, c – cabinet projection view of cuboids.

The solution to this problem is the use of track dilation. The tracks are sorted from largest to smallest in terms of dominated grid cells, and, in that order, the cells of the tracks are dilated to cover neighboring cells of different label. The structural element used for dilation is the 3x3 "cross" element, affecting the points in the 4-way neighborhood. This way, the large tracks become clearer and the smaller tracks, nested or overlapping the large ones, will be eliminated.

IV. Tests and Results

As the proposed tracking model is, for now, a concept demonstrator and not yet a refined system, the tests were aimed not to establish superiority over other systems, but to prove its viability to track real life targets and to identify the conditions that affect its operation. Two sequences were used in the tests:

- a 13500 frames monocular color sequence acquired by a mobile device, at 15 fps, with poor odometry data (speed read from the GPS receiver, accurate when traveling constantly, slow to update on accelerations, and very noisy yaw rate read from phone's gyroscope)
- a 9800 frames stereo monochrome sequence acquired by on-board industrial camera, at 12 fps, with accurate odometry data read from vehicle's CAN bus.

All sequences were processed using the particle-based occupancy grid tracking system described in [20]. Variations were attempted when selecting the primary input data for the particle grid tracking system. For the first sequence, the trackable obstacle areas were initially selected as all areas that are not road, segmented by a two-class U-Net semantic classification network, as seen in Fig. 4, and mapped in the bird's-eye view space by Inverse Perspective Mapping [21]. This approach ensured a large number of detected obstacles, including scenery outside of the road area, but also favored track confusion, especially for the objects on the side of the road or very near it.

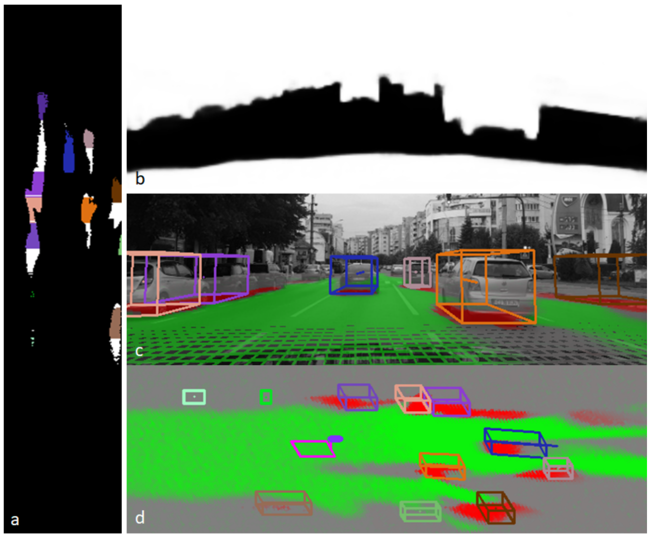

Fig. 4. Tracking using monocular images, segmented for road area: a – track labels, b – semantic segmentation, c – tracked objects, perspective view, d – track objects, cabinet projection.

In order to increase the stability of the tracks, the semantic classification was changed to only segment dynamic objects: vehicles, pedestrians, etc, excluding the scenery and buildings, as seen in Fig. 5. This restricted approach improves the stability of the tracks, but reduces the number of detected objects.

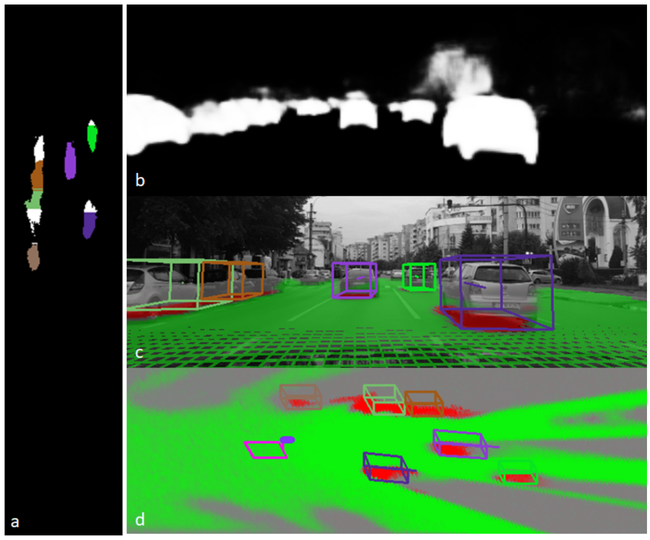

Fig. 5. Tracking using monocular images, segmented for dynamic objects only: a – track labels, b – semantic segmentation, c – tracked objects, perspective view, d – track objects, cabinet projection.

A comparison between the all-obstacle segmentation and dynamic only obstacles segmentation is shown in Table I. We can see that the number of obstacles detected is reduced to half, due to the selection of the obstacle types, but the deactivation rate was reduced four times, and the mean track life was extended from 22.52 frames to 34.54. The maximum track life registered a significant jump, from 809 to 1467 frames.

TABLE I. TRACK STABILITY VS. SEGMENTATION TYPE

	All obstacles segmentation	Dynamic obstacles segmentation
Detected obstacles (output cuboids)	70439	28463
Detected obstacles (output cuboids) / frame	5.22	2.11
Track deactivations	3972	938
Track deactivations/ frame	0.29	0.07
Mean track life	22.52	34.54
Median track life	11.12	12.23
Max track life	809	1467

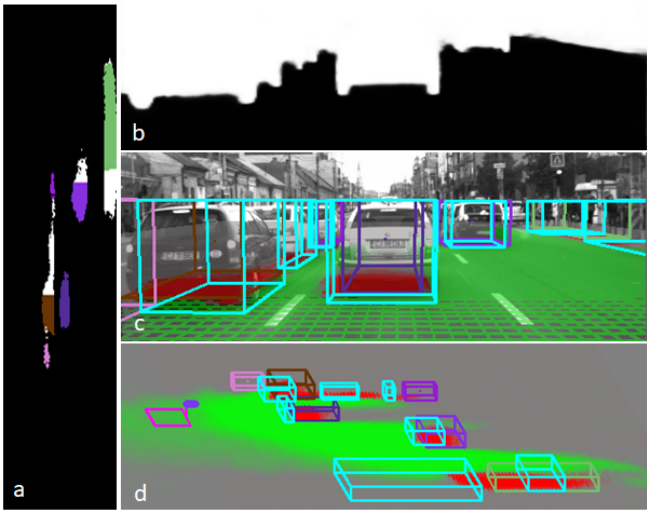

Fig. 6. Comparison with stereovision: monocular input for particle tracking, segmented for road areas. a – track labels, b – semantic segmentation, c- track results, perspective view, d – track results, cabinet projection. The cyan cuboids are results of stereovision-based tracking.

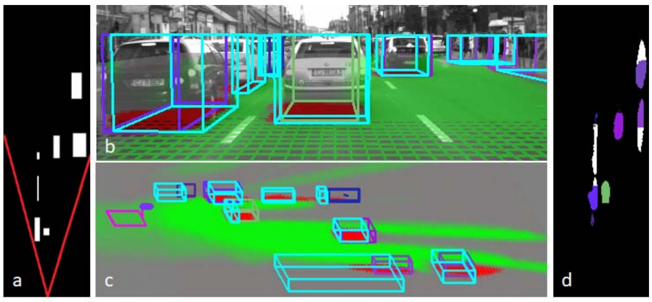

Fig. 7. Comparison with stereovision: stereovision generated 3D cuboids input for particle tracking. a – 3D input, b – track results, perspective view, c – track results, cabinet projection, d – track labels. The cyan cuboids are results of stereovision-based tracking.

The second sequence, consisting of monochrome frames which also have 3D information extracted by dense stereovision, was also processed in two ways. First, the sequence was processed as a monocular one, using road vs obstacles semantic segmentation (as shown in Fig. 6). Some minor errors were introduced by semantic segmentation due to lack of color, but overall the system was able to track the objects and produce boxes that were consistent with the boxes generated by a stereovision-based, Kalman filter cuboid tracking system [22].

The second approach was to use the stereovision 3D data to create the bird's eye view binary obstacle map subsequently used by the grid tracking algorithm (as shown in Fig. 7). As seen from Table II, the detection rate was only slightly reduced, but the deactivation rate was reduced to half, and the mean track life was extended from 19.99 to 30.74 frames, meaning that the tracks were a lot more stable.

TABLE II. MONO VS. STEREO

	Monocular input data	3D input data
Detected obstacles (output cuboids)	36873	34826
Detected obstacles (output cuboids) / frame	3.76	3.55
Track deactivations	2448	1195
Track deactivations/ frame	0.25	0.12
Mean track life	19.99	30.74
Median track life	11.56	18.25
Max track life	758	853

Another qualitative comparison between tracking on monocular images versus tracking on stereovision data can be made by observing the figures 8 and 9, which contain positions of detected objects for 1000 frames. We can see that both trackers follow pretty much the same trajectories, concentrated on the three directly visible lanes, with occasional lane changes. The main difference in the continuity and smoothness of the trajectories, especially in the case of smaller objects such as pedestrians.

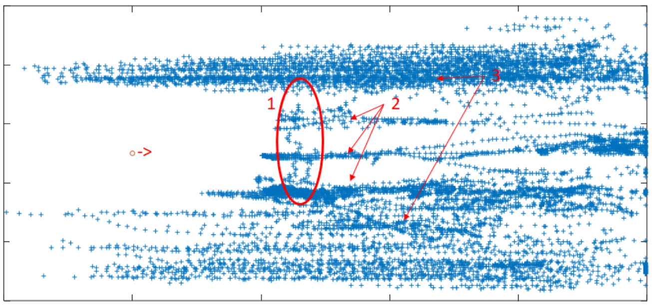

Fig. 8. Trajectories for 1000 frames, monocular processing of the stereo sequence: 1 – pedestrians crossing, 2 – vehicles on the current lane and two side lanes, 3 – stationary objects on the side of the road.

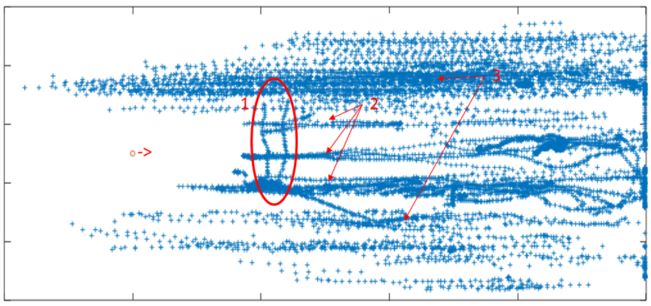

Fig. 9. Trajectories for 1000 frames, using stereo 3D data as input: 1 – pedestrians crossing, 2 – vehicles on the current lane and two side lanes, 3 – stationary objects on the side of the road.

The execution time of the particle-based tracking process is, on average, 45 ms. This time includes all the steps described in this paper, starting from the updated particle set *S* and finishing at oriented cuboids generation. The whole grid tracking processing cycle, starting from the segmented image, takes, on average, 110 ms. These times were measured on a Intel Core i7-8700 CPU, running at 3.2 GHz, on a single thread. These times indicate that a suitably optimized implementation, designed for parallelization, can easily achieve real time performance.

V. DISCUSSION

A conceptual comparison of our tracking method and model will be presented in this section. The most popular approaches are the Kalman Filter and the Particle Filter (where each particle is a complete hypothesis of the state of the target). Due to the fact that it is closely related to our work, we will also analyze a part-based particle filter method. All tracking systems estimate continuously the state of one or more targets based on a sequence of observations. The state and the observations have uncertainties which are usually modeled probabilistically. The specific tracking solutions differ in the ways they represent the target state, the uncertainties, and how they use the measurement to update this state.

TABLE III. COMPARISON OF GENERIC TRACKING SOLUTIONS

	Kalman filters	**Particle filters (particles as complete hypotheses)**	**Part based particle filter**	**Particle identities tracker (proposed method)**
Target state	- Mean value of a state vector - Fixed number of state parameters	- Any particle is a complete target state description - Multiple particles allow for a multi-modal state - Particle weights indicate that hypotheses have different importance	- Hierarchical decomposition of the targets in parts - Each part has its own state, represented by a population of hypotheses particles - Parts are strongly correlated by motion and appearance	- Individual particles with identity label, not constrained by their position to the whole object - Multiple particles allow for multi-modal state - Each particle has the same weight
State uncertainty	- Covariance matrix of the state vector	- Distribution of the particles in the target state space - Distribution of weights in the particle set	- Distribution of part particles in their own state space - Whole state uncertainty depends on part uncertainties and the part-whole model	- Distribution of the individual particles in the occupancy grid - Distribution of the particles' speed vectors
Prediction	- State vector predicted using motion model and the past mean - Covariance matrix predicted from past covariance matrix and transition the state transition uncertainty matrix	- Particle state vectors are predicted using motion models and their past state - The new state prediction is represented by the values of each particle	- Part particles are moved by taking into account their association with the whole object - The new state prediction is represented by the values of each particle	- Movement of particles based on their own speed vectors - The identity of the particle is carried along with it
Measurement to track association	- Requires clear association between any measurement and the track - Handles association uncertainty poorly	- No clear association is required. All particles are matched with the measurement and their weights are updated accordingly - Robust to association uncertainty	- No clear association is required. All part particles are matched with the measurement and their weights are updated accordingly - Robust to association uncertainty	- No clear association is required. Particles are multiplied or deleted based on their position with respect to the measurement data mapped to the occupancy grid - Robust to association uncertainty
State update	- Update of state mean and covariance matrix by means of Kalman filter equations	- Particles are resampled based on their weights	- Particles are resampled based on their weights - The whole object is inferred from the state of part particles and their relation to the whole	- Particle IDs are reassigned based on the particle's position with respect to track labels
Handling of irregular objects	- Possible, but all irregularities must be included in the model - A complex state model leads to complex prediction and association	- Possible, but all irregularities must be included in the model, for each particle - A complex state model leads to costly matching between particle and measurement	- Not possible, as the positions of the part particles with respect to the whole is a key element of the algorithm - The rigid shape offers resilience in case of occlusions.	- Natural, the particle set for a tracking ID is not constrained to any shape

VI. CONCLUSION AND FUTURE WORK

This work describes a lightweight solution for individual object tracking in dynamic occupancy grids. The particle-based implementation of the dynamic grid allowed us to come up with the simple extension of assigning identity to particles and letting them move the identity around as the tracking system performs predictions and updates. The sub-optimal mechanism for assigning and maintaining particle identities, based on frame-by-frame labels, ensures that the execution time remains small, while the objects are reliably tracked for a large number of frames.

The resulted system is able to track objects independently of their type, size or shape. The same type, size and shape independence is specific to occupancy grid tracking solutions in general, and therefore we believe that this contribution completes one of the most popular 3D environment tracking solutions by enabling it to also be aware of individual objects identities.

The most important contribution of this paper is the development of the particle identity concept. The tracking algorithm developed around the particle identity model is only a concept demonstrator, and its performance can be significantly improved. For example, the age of the particle can be used in the process of identity reassignment, by making older particles less likely to change their identity. Particle level clustering can also be employed, instead of relying on labels, to maintain identities and avoid identity spread. There are many improvements that can be attempted, and the particle identity model is flexible enough to allow all these variations.

ACKNOWLEDGMENT

This work was supported by a grant of Ministry of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2016-0440, within PNCDI III.

REFERENCES

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Journal of Basic Engineering, Vol. 82, 1960, pp. 35–45.

[2] D. Mitzel and B. Leibe, "Real-time multi-person tracking with detector assisted structure propagation," IEEE International Conference on Computer Vision (ICCV) Workshops, 2011, pp. 974–981.

[3] J. S. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems", in Kadar, Ivan (ed.). Signal Processing, Sensor Fusion, and Target Recognition VI. Proceedings of SPIE, pp. 182–193, 1997.

[4] S. Wang and C. C. Fowlkes, "Learning optimal parameters for multitarget tracking with contextual interactions," International Journal of Computer Vision, Vol. 122, 2017, pp. 484-501.

[5] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna, "Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking", 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 3508–3515.

[6] M. P. Muresan, I. Giosan, and S. Nedevschi, "Stabilization and Validation of 3D Object Position Using Multimodal Sensor Fusion and Semantic Segmentation", Sensors, Vol. 20, 2020, article no. 1110.

[7] H. Karunasekera, H. Wang and H. Zhang, "Multiple Object Tracking with Attention to Appearance, Structure, Motion and Size," IEEE Access, vol. 7, 2019, pp. 104423-104434.

[8] K. Yoon, D. Y. Kim, Y.-C. Yoon, and M. Jeon, "Data association for multiobject tracking via deep neural networks," Sensors, Vol. 19, 2019, article no. 559.

[9] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[10] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. Torr, "Fully-convolutional siamese networks for object tracking", European Conference on Computer Vision (ECCV), 2016, pp. 850–865.

[11] G. Wang, C. Luo, Z. Xiong, and W. Zeng, "SPM-Tracker: Series-Parallel Matching for Real-Time Visual Object Tracking", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3643-3652.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", Neural Information Processing Systems, 2012, pp. 1097–1105.

[13] M. Isard, A. Blake, "CONDENSATION-- conditional density propagation of visual tracking", International Journal of Computer Vision, Vol. 29, 1998, pp. 5–28.

[14] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," IEEE International Conference on Computer Vision (ICCV), 2009, pp. 1515–1522.

[15] Y. Fang, C. Wang, W. Yao, X. Zhao, H. Zhao and H. Zha, "On-Road Vehicle Tracking Using Part-Based Particle Filter," IEEE Transactions on Intelligent Transportation Systems, Vol. 20, 2019, pp. 4538-4552.

[16] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," IEEE Transactions on Image Processing, Vol. 13, 2004, pp. 1491–1506.

[17] N. Wang, W. Zhou and H. Li, "Robust Object Tracking Via Part-Based Correlation Particle Filter," 2018 IEEE International Conference on Multimedia and Expo (ICME), 2018, pp. 1-6.

[18] A. Elfes, "A Sonar-Based Mapping and Navigation System", IEEE International Conference on Robotics and Automation, 1986, pp. 1151-1156.

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

[20] 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, 2011, pp. 1331-1342.

[21] R. Itu and R. G. Danescu, "A Self-Calibrating Probabilistic Framework for 3D Environment Perception Using Monocular Vision," Sensors, Vol. 20, 2020, article no. 1280.

[22] 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", 2007 IEEE Intelligent Vehicles Symposium, (IV2007), 2006, pp. 276-283.