Multi-Object tracking of 3D cuboids using aggregated features

Mircea Paul Muresan, Sergiu Nedevschi
Computer Science Department
Technical University of Cluj-NapocaCluj-Napoca, Romania
{mircea.muresan, sergiu.nedevschi}@cs.utcluj.ro

***Abstract*— the unknown correspondences of measurements and targets, referred to as data association, is one of the main challenges of multi-target tracking. Each new measurement received could be the continuation of some previously detected target, the first detection of a new target or a false alarm. Tracking 3D cuboids, is particularly difficult due to the high amount of data, which can include erroneous or noisy information coming from sensors, that can lead to false measurements, detections from an unknown number of objects which may not be consistent over frames or varying object properties like dimension and orientation. In the self-driving car context, the target tracking module holds an important role due to the fact that the ego vehicle has to make predictions regarding the position and velocity of the surrounding objects in the next time epoch, plan for actions and make the correct decisions. To tackle the above mentioned problems and other issues coming from the self-driving car processing pipeline we propose three original contributions: 1) designing a novel affinity measurement function to associate measurements and targets using multiple types of features coming from LIDAR and camera, 2) a context aware descriptor for 3D objects that improves the data association process, 3) a framework that includes a module for tracking dimensions and orientation of objects. The implemented solution runs in real time and experiments that were performed on real world urban scenarios prove that the presented method is effective and robust even in a highly dynamic environment.**

Keywords—multi-target tracking; data association; MDP; smoothing trajectories, feature engineering;

I. INTRODUCTION

Efficient and reliable perception is one of the core functions for representing the dynamic environment by autonomous vehicles. The ability to effectively detect the surrounding traffic scenarios plays an important role for many of the self-driving car components such as collision avoidance, path planning or localization. In order to navigate successfully several complex situations have to be addressed. The most difficult being the crowded places where multiple static and dynamic objects are present which may exhibit various motion behaviors. For the problem of environment perception, the target tracking process is essential since provided measurements are useful only if they are filtered (not noisy) and identifiable in occluded situations such that higher level modules from the processing pipeline can transform each measurement in an actionable information.

To address such complex scenarios which may occur in various weather conditions multiple types of complementary sensors are usually employed. Chiefly among them the LIDAR (Light Detection and Ranging) sensor is used because of its ability to provide an accurate position [1, 2]. Other sensors like stereo cameras can also be used because of their ability to provide the semantic class additionally to the position estimate of objects [3]. The main issue that appears with stereo sensors is that they may not work well in case of bad illumination, perspective effect or lack of texture among others [4]. Radars are another category of range sensors which are used in autonomous vehicles because of their long range detection ability and capacity to accurately detect motion. The drawback of radars is that they have a reduced field of view and are not able to reliably detect static objects or objects made of porous plastic [5]. Modern perception and tracking architectures usually fuse all the available sensor data to obtain a more comprehensive understanding of the environment. However one key aspect of any modern architecture is adaptability in case of sensor failure. In such a case the remaining sensors should be able to accurately detect and track the road objects. In this paper we address the problem of target tracking using a LIDAR sensor.

We split the challenges of developing a robust tracking algorithm in three categories: high level processing pipeline related challenges, target tracking related issues and time constraints.

The high-level processing pipeline challenges refers to the errors introduced in the target tracking module by the output provided by other modules from the self-driving car processing pipeline, inefficient sensor calibration or bad sensor synchronization. The general pipeline of the detection and tracking procedure includes steps like point cloud segmentation, candidate matching and motion estimation [6]. The quality of the point cloud segmentation algorithm impacts the quality of the tracking results. Existing methods in the literature work either on 2D [7] grid maps or 3D occupancy grid maps with higher computational burden [8]. Incorrect segmentation leads to a difficult candidate matching and tracking in a cluttered scenario. Some of the common issues of objects obtained by incorrect point cloud segmentation are change in appearance, unreliable dimensions and fluctuating positions in consecutive frames.

On the other hand, poor synchronization of LIDAR and camera may lead to bad point cloud projection in the image. Which may result in 3D points with an erroneous semantic class. In figure 1 we can see such a scenario. In the left-hand side of the image, we can observe the semantic class of each projected 3D LIDAR point in the semantic image. As we can

978-1-7281-4914-1/19/$31.00 ©2019 IEEE

Fig.1 Erroneous semantic class of point cloud projection

intuitively see these points do not fall on the correct object which may infer a poor sensor synchronization or an erroneous motion correction.

The second challenge refers to the issues one may have when developing a tracking algorithm. Such a challenge can include issues like the motion uncertainty, the origin uncertainty or the presence of heavy clutter among others. The motion uncertainty refers to the fact that real objects can have a complex motion that cannot be described by single motion model. One of the key challenges is represented by the data association step which aims to match identified targets to oncoming measurements in order to maintain the object identity. Mistakes made in the identity maintenance could result in a catastrophic failure in many high-level reasoning tasks. To obtain a highly accurate multi target tracking solution, a robust data association model and an accurate measure to compare detections over time is necessary. Popular data association and motion estimation methods include steps like feature-based matching followed by a filtering step using the Kalman or Extended Kalman filter. Lastly the time constraint refers to the fact that the target tracking module is useful only if it is running in real time. To address the above-mentioned issues in this paper we propose the following contributions:

- The implementation of an affinity measurement function and the development of a positional descriptor that exploits multiple features coming from LIDAR, camera and from a semantic segmented image, to find the best track-measurement correspondences
- We modeled the tracking problem as a Markov decision process that improves the quality of the data association and tracking process
- The inclusion of a module for filtering the orientation and dimension properties of tracked objects

The rest of the paper is organized as follows: In section II we overview the literature on data association and tracking and in Section III, the proposed solution is described. In Section IV we evaluate the obtained results using multiple metrics. Section V concludes the paper.

II. Related Work

Existing tracking algorithms aim to model the environment at different abstraction levels, depending on the complexity of the surrounding world. Many video-based tracking solutions have been developed in recent years due to the low cost and availability of the video sensors [9, 10]. The problem with video-based tracking resides in the fact that all of these methods can be affected by environmental conditions such as weather or illumination. On the other hand, 3D LIDARs are not affected by the illumination conditions, the scale of their measurements are uniform despite their distance, and due to technological advancement, the sensors are becoming more affordable. An approach used in the research literature is the probabilistic data association (PDA) [11] filter which does not rely on a single measurement to estimate the state and covariance of an object but uses the set of validated measurements. Variations of the algorithm include the joint PDA [12] used in handling multiple targets or the integrated PDA in which the data association probability and the track existence are estimated jointly [13]. The JPDA (Joint Probabilistic Data Association) filter can exhibit poor performance when the objects are close to each other. Another class of more powerful algorithms use variations of algorithms like the multiple hypothesis tracking (MHT) [14, 15] to solve the multiple target tracking. This MHT solution retains all possible data association hypothesis until there is enough information to resolve the ambiguities that occurred in older associations. The issue with MHT is that the algorithm is computationally more expensive compared to JPDA or GNN. Some methods model objects at a higher level of abstraction using oriented cuboids [16] or L shaped models [17] due to the simplicity in which cars, pedestrians or other road users can be represented. The box representation is unable to represent complex structures like infrastructure or vegetation, yet it is often used due to the simplicity of implementation and high running time which is necessary in large computational pipelines. In order to improve the performance of the detection and tracking algorithm, various solutions try to find a tradeoff between more sophisticated representations and computational efficiency. For example, in [18] the dynamic objects are modeled as deforming contours, in [19] individual 3D points are tracked and in [20] boxes with adaptive sizes are used.

The task of improving the data association process has led some researchers to fuse the information coming from a camera with the 3D points. In this regard Held et al. [21] fused a 3D point cloud with a color image to obtain a colored 3D point cloud. The authors have used the 3D shape and color data to reconstruct and track the objects. They have showed that the usage of multiple features leads to an overall better velocity and position estimate. In [22] Asvadi et al. propose a 3D object tracking algorithm using a 3D LIDAR, an RGB camera and an GPS/IMU sensor. The solution starts with a known initial 3D bounding box for an object and then two parallel mean-shift algorithms are applied for object detection and localization in the 2D image and 3D point cloud, followed by a robust 2D/3D Kalman filter based fusion and tracking.

Other solutions use particle filter to estimate shape, velocity and object movement. The particles are independent instances each having their own position and velocity. In [23] the particle filter is employed to estimate velocities, while [24] applies a mix of static and dynamic particles to estimate position and velocities. In general grid-based tracking solutions are not able to accurately estimate the state of cells belonging to a large uniform area, and this leads to higher uncertainty due to incorrect data association. In [25] the authors present an interesting tracking solution that performs a probabilistic hierarchical object association based on 3D information provided by a stereo camera and optical flow data. This solution relies heavily on image quality to detect, associate and track objects and it is not suitable for scenarios with adverse weather conditions.

III. Proposed Solution

The most challenging aspect of object tracking is arguably that the associations between measurements and objects is unknown. The objective of multi-object tracking is to compute the posterior density as fast and as reliable as possible for each object of interest. Considering that the sensor, measurement and motion models are linear and Gaussian, the exact posterior density can be expressed as a Gaussian mixture with one term for every association at time k as seen in equation 1. The term $w_{k|k}^{\theta_{1:k}}$ is a probability mass function which denotes the probability of association to a measurement and $P_{k|k}^{\theta_{1:k}}$ represents a probability density function. We denote the fact that the Gaussian mixture spans over all associations that fall in the covariance ellipse of a target by the sum $\sum_{\theta_{1:k}}$

$$P_{k|k}(X_k) = \sum_{\theta_{1:k}} w_{k|k}^{\theta_{1:k}} P_{k|k}^{\theta_{1:k}}(X_k) \quad (1)$$

We try to find the best measurement association for each target, θ^* and prune all other associations that are situated in the covariance ellipse of the target object in a global nearest neighbor manner. Finding a single association will give a computationally cheap algorithm which can meet the real time performance requirement of a self-driving car. The posterior density can be approximated by $P_{K|K}^{GNN}(X_k)$ in (2), where $\theta_{1:k}^*$ is the sequence of optimal data associations from time 1 to time k

$$P_{K|K}^{GNN}(X_k) = P_{K|K}^{\theta_{1:k}^*}(X_k) \quad (2)$$

In order to make sure that an association is more probable, the correspondence between a measurement and a hypothesis is done using many aggregated features which will be described shortly. For the prior densities that are Gausian and the measurement model is linear and Gausian the Kalman update and prediction rules are used to find the posterior densities. Otherwise we linearize the predicted density using the sigma point sampling and the Unscented Kalman Filter for the update and prediction of the next state. The general pipeline of the tracking process is illustrated in figure 1. Two motion models were used in order to achieve a better modeling of the road users motion behavior. The motion models used are CTRV (constant turn rate and velocity model) and the CV (constant velocity model).

A. Data association score

1) Aggregated Affinity score

In order to select the best measurement correspondent to a target, multiple discriminant features have to be considered. In general, the task of feature selection is a challenging endeavor. The choice of features varies depending on the tracking application. For example, in order to track an object which is very small, the centroid feature is usually used. On the other hand, for large objects a combination of various features may be more advantageous.

In this work, the object association position is considered by using the coordinates of the nearest corner, visible to the ego vehicle. Furthermore, a 3-channel reduced color histogram is used for each object. The color histogram has 8 bins per channel, and it is obtained by projecting the 3D points that correspond to an object onto the front RGB image. Each 3D point that falls in the image casts a vote in a specific bin from a channel inside the color histogram of the object. Each bin of the histogram can store 32 intensity values. The color score is computed as presented in equation (3).

$$ColorERROR = RMS(LIDAR.R, Track.R) + RMS(LIDAR.G, Track.G) + RMS(LIDAR.B, Track.B) \quad (3)$$

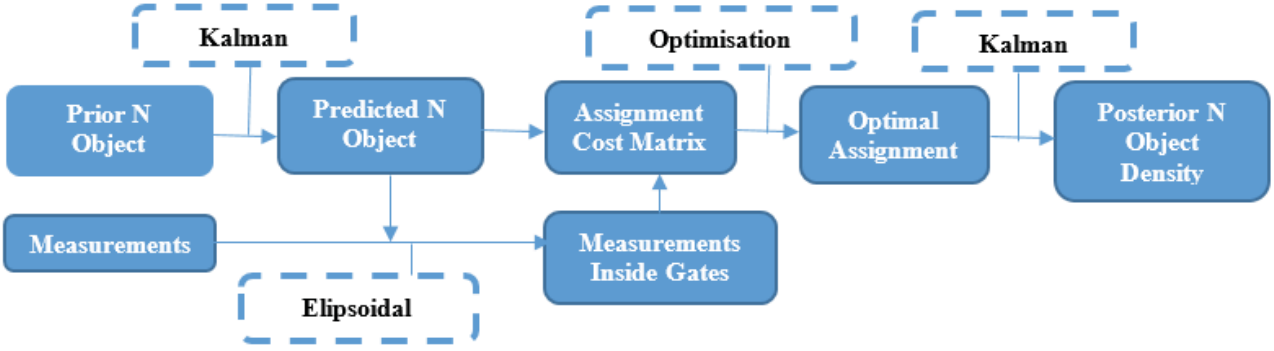

Fig. 2. Processing pipeline for obtaining the posterior N object density

RMS is the root mean square metric defined by equation (4), where *COL.HIST. BINS* is the number of bins of the color histogram for each channel.

$$RMSerror = \frac{1}{COL.HIST.BINS}\sum_{I=0}^{COL.HIST.BINS}(veloHist(i) - trackHist(i))^2 \quad (4)$$

The point cloud that corresponds to a measurement is also projected onto a semantic segmentation image, obtained using the ERF neural net [26] such that, semantic class is also extracted for each object. Since the semantic segmentation image is not perfect and the point projections may not fall entirely on the desired object in the image, due to motion or synchronization errors, the most probable 3 semantic classes are used together with their probabilities (5).

$$FrequencyCost = \sum_{i=0}^{3}\begin{cases}0, if\ w[i] = -1\\ |Measurement.P(i) - Hypothesis.P(w[i])|\end{cases} \quad (5)$$

In equation (5), w[i] takes the value of the position where semantic class of the hypothesis matches the semantic class of the measurement. If there is no class in the target that should match the semantic class of the measurement, w[i] takes the value -1. The symbol |a| refers to the absolute value of a. If two objects are similar the combined color and semantic segmentation cost will be very low (close to zero in case the two objects are identical). The cost function has been implemented such that two similar objects have a very high similarity score. Due to the fact that semantic and color information may become unreliable in case of bad weather conditions, we are taking the inverse of the computed scores for the features generated using a camera. The proposed methodology of computing the association score has the advantage that in adverse weather conditions when the camera information is no longer reliable, the terms in the score function corresponding to image and semantic segmentation will have a very low weight, and the final score will be computed based mostly on geometric features extracted from the LIDAR objects.

Geometric properties such as object area, width, length and measurement-hypothesis overlapping are extracted, and used (6) for eliminating candidates that are not similar to the compared object.

$$objectDimension_{i,j} = \frac{coveredArea_{i,j}}{|areaLDIAR_i - areaTrack_j|} \quad (6)$$

Finally, we also use the cuboid orientation, because, we reason that object orientation cannot change drastically from one frame to the other (7).

$$orientationSimilarty = |orientationMeasurement - orientationTarget| \quad (7)$$

The final association score for a measurement, hypothesis pair is obtained by summing the values obtained in equations (3), (4), (5), (6) and (7) and weighting it by the distance (denoted by wd) between the measurement and target. The closer the target is to the measurement, the more reliable the aggregated score is. This aggregated score (AgSc) (8), which can also be considered the weight ($w^{\theta_{i,j}}$) of the measurement i – target j association for a validated measurement in the target covariance ellipse, is added to the cost matrix.

$$AgSc_{i,j} = \left(\left|\frac{1}{ColorERROR_{i,j}+0.001}\right| + \left|\frac{1}{FrequencyCost_{i,j}+0.001}\right| + objectDimension_{i,j} + orientationSimilarity_{i,j}\right)/wd \quad (8)$$

2) Positional Descriptor (PD)

A positional descriptor is used to better describe the relation between the neighborhood of each target and measurement. We make an assumption that the neighborhood of a measurement and its corresponding target should be similar to a certain degree. The PD descriptor is computed as a sum over the ColorERROR (CE) differences and objectDimension (OD) differences of a target vehicle and its neighboring target vehicles in a vicinity around the target up to 10 m (9). A similar step is performed for each descriptor.

$$PD(target_i) = \sum_{j=0}^{neighbouringTargets}(CE_i - CE_j + OD_i - OD_j) \quad (9)$$

The same positional descriptor is also computed for each measurement. The difference between the track positional descriptor and the measurement descriptor should be as small as possible for similar objects. The difference is added to the overall aggregated score. In case there is no available descriptor a penalty value is added to the final score (10). The penalty value has been found experimentally.

$$AgSc_{i,j} = \begin{cases}PD_{target^i} - PD_{measurement^j}, if\ \exists\ PD_{target^i}, PD_{measurement^j}\\ Penalty, otherwise\end{cases} \quad (10)$$

To describe the data association, we use the letter Θ , measurements are symbolized with the letter z and i is the position of the measurement in the queue where the gated measurements are stored (11).

$$\Theta = \begin{cases}i > 0, if\ z^i\ is\ an\ object\ detection\\ 0,\ \ if\ object\ is\ undetected\end{cases} \quad (11)$$

The goal of the optimal assignment stage of the tracking processing pipeline is to find a sequence $\Theta^* = [\Theta_1\ \Theta_2 \dots \Theta_n]$ such that the sum of negative log weights (12) is minimized.

$$\theta^* = argmin\ \sum_{i=0}^{n} -\log(w^{\theta_{i,j}}) \quad (12)$$

The optimal assignment can be achieved using the Hungarian [27] or Auction [28] algorithms, and will not be discussed in this paper.

B. The tracking processes

1) Tracking as a Markov Decision Process

In this section we introduce a Markov decision process formulation of the lifetime of a single target in the tracking process. The MDP consists of the tuple (S, A, T(*), R(*)):

- The target state s Є S encodes the status of the hypothesis
- The action a Є A, represents the action which can be performed to the target
- The state transition function T: SxA → S describes the effect of each action in each state
- The reward function is a real-valued function that defines the immediate reward received after executing action a in state s; R: SxA → ℝ

The state space contains five states: Initialized, Processed, Updated, Drifting and Absolute Death. Figure 3 shows the

transition between the five states. Beside the five states the target also memorizes other information such as RGB appearance, geometry, semantic class, orientation, velocity, localization, size, history and others.

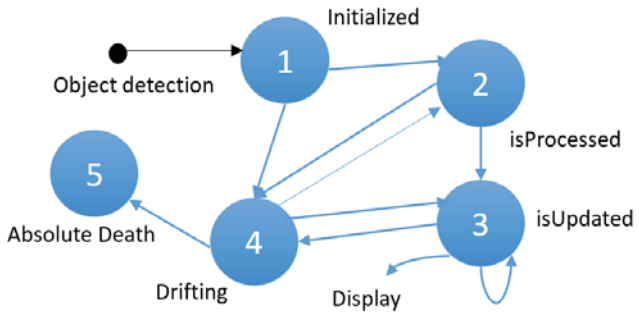

Fig. 3. State transition process

Initialized is the first state of any hypothesis, in this state no action is performed on the target. Whenever an object is detected by the object detector it enters the Initialized state. In the second state, Processed, a target was associated to a new measurement based on an affinity score. In this state, the MDP needs to decide whether to keep tracking the target or transfer it to a drifting state. If the track is associated in the next frame it is transferred into the third state. A track falls in the state, Updated, when it is constantly updated using the incoming measurements. If a hypothesis is updated for a number of three frames while being in the Updated state, the track is labeled stable and it is displayed. When a hypothesis has not been associated to any measurement for a number of three frames it enters the fourth state, the drifting state, and will not be displayed until new successful association are found in the next frames. The track is not removed because the target may be lost due to some reason, such as occlusion, or disappearance from the field of view. Finally, if a target has not been associated for a number of 15 frames, it has left the area of interest, or a time of 2 seconds has elapsed since it was last updated it will enter state five, absolute death, and it will be removed as soon as possible.

In the current MDP all the actions are deterministic, i.e. given the current state and an action we can figure the new state for the target. The actions which determine the transitions between states are given by the data associations and their number. For example, if a target object is in state 1 and there has been a successful association in the next frame, the internal state will transition to state 2 otherwise it will fall in state 4. In state 1 and 2 the object receives a unitary reward, while in state 3 each time the target is successfully associated a unit is added to the target reward. No rewards are given in state 4 and 5.

2) Prediction and update of target states

The target state has the following constituents: the target position on the x and y dimensions, the velocity, the yaw and the yaw rate (13).

$$X_k = \begin{bmatrix} px \\ py \\ v \\ \Psi \\ \dot{\Psi} \end{bmatrix} \tag{13}$$

We are considering two motion models and a collaboration scheme that links the two models. The objects are tracked independently using the CTRV and the CV motion models. The results are then combined using a collaboration strategy based on the orientation as described in [4]. For tracking objects using the CV motion model, we are using the classical Kalman prediction and update rules. On the other hand, tracking objects having the non-liner motion model, CTRV, requires the usage of the Unscented Kalman filter. The process functions in the UKF depend on the type of motion used. In case the yaw angle is not 0, the process model can be seen in (14).

$$X_{k+1} = X_k + \begin{bmatrix} \frac{v_k}{\dot{\Psi}_k}(\sin(\Psi_k + \dot{\Psi}_k \Delta t) - \sin(\Psi_k)) \\ \frac{v_k}{\dot{\Psi}_k}(-\cos(\Psi_k + \dot{\Psi}_k \Delta t) + \sin(\Psi_k)) \\ 0 \\ \dot{\Psi}_k \Delta t \\ 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2}(\Delta t)^2 \cos(\Psi_k)\gamma_{a,k} \\ \frac{1}{2}(\Delta t)^2 \sin(\Psi_k)\gamma_{a,k} \\ \Delta t \gamma_{a,k} \\ \frac{1}{2}(\Delta t)^2 \gamma_{\dot{\Psi},k} \\ \Delta t \gamma_{\dot{\Psi},k} \end{bmatrix} \tag{14}$$

For rectilinear motion the state transition equation is described in (15)

$$X_{k+1} = X_k + \begin{bmatrix} v_k \cos(\Psi_k)\Delta t \\ v_k \sin(\Psi_k)\Delta t \\ 0 \\ \dot{\Psi}\Delta t \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}(\Delta t)^2 \cos(\Psi_k)\gamma_{a,k} \\ \frac{1}{2}(\Delta t)^2 \sin(\Psi_k)\gamma_{a,k} \\ \Delta t \gamma_{a,k} \\ \frac{1}{2}(\Delta t)^2 \gamma_{\dot{\Psi},k} \\ \Delta t \gamma_{\dot{\Psi},k} \end{bmatrix} \tag{15}$$

The UKF manages to recover the Gaussian density by propagating a set of sigma-points through the non-linear process function. The covariance matrix is recovered by using the sigma points and a set of weights, which have the role of inverting the spread of the sigma points (16) and (17). These weights depend on the spreading parameter lambda.

$$w_i = \frac{\lambda}{\lambda + n_a}, i = 0 \tag{16}$$

$$w_i = \frac{1}{2(\lambda + n_a)}, i = 2, \dots, n_a \tag{17}$$

The state mean and covariance are predicted using equations (18) and (19) below.

$$X_{k+1|K} = \sum_{i=1}^{n_\sigma} w_i X_{k+1|k,i} \tag{18}$$

$$P_{k+1|K} = \sum_{i=1}^{n_\sigma} w_i (X_{k+1|k,i} - x_{k+1|k})(X_{k+1|k,i} - x_{k+1|k})^T \tag{19}$$

Due to the fact that the measurement model is linear and Gaussian the update step is performed as described in the classical Kalman filter as described in (20), (21) and (22).

$$K = P_k H^T (H P_k H^T + R)^{-1} \tag{20}$$

$$X_k = X_k + K(z_k - H X_k) \tag{21}$$

$$P_k = (I - KH) P_k \tag{22}$$

In equation 20, K is the Kalman gain, X_k is the state of the target at time step k and P_k is the covariance at time k.

C. Filtering meta parameters

Parameters such as the width, height and orientation of a cuboid are referred to as meta parameters and are filtered in a separate module. The reason for doing this is that the three measured parameters fluctuate very violently in consecutive frames due to object segmentation problems.

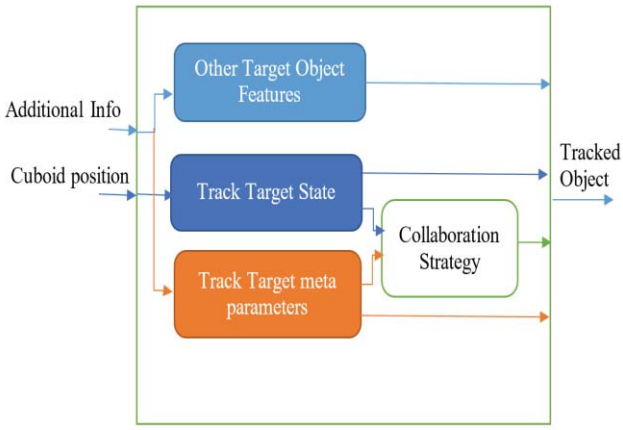

Fig.4. Components belonging to a tracked object

The usage of these parameter in the state vector presented in equation (11) would compromise other values that might depend on them, for example the object position. A view of the object tracking module can be seen in figure 4.
The state vector for the meta parameters tracker is illustrated in (23).

$$X_k = \begin{bmatrix} \Psi \\ \dot{\Psi} \\ width \\ length \end{bmatrix} \quad (23)$$

The transition equations for each member parameter are depicted in (24), (25) and (26) below

$$\Psi_k = \Psi_k + (\dot{\Psi})_{k+1}\Delta t_{k+1} \quad (24)$$

$$width_k = width_k + (width_{k+1} - width_k) \quad (25)$$

$$length_k = length_k + (length_{k+1} - length_k) \quad (26)$$

The Kalman filter is used to update and predict the next state for the meta parameters. The collaboration strategy between the tracked target state and the meta-parameter state is based on the level of maturity of each track, i.e. the number of successful associations. After computing the correspondence between the hypothesis obtained by each motion model as described in [4] the level of maturity analogous to the meta-parameters, is checked. The targets that have a higher level of maturity will be accorded a higher weight. The final orientation will be a weighted sum of the orientations obtained in the meta parameters associated to each motion model and the one inferred by the UKF as seen in (27).

$$final_{orientation} = w1 * orientation_{ukfInfered} + w2 * orientation_{CV_{meta}} + w3 * orientation\_CTRV_{meta} \quad (27)$$

IV. Experimental Results

In this section we will evaluate the results of the proposed solution with respect to two metrics MOTA (Multiple Tracking Accuracy) and MOTP (Multiple Object Tracking Precision). The system on which the method was tested contains an Intel i5-2500 CPU with 3 GHz frequency. The running time of the solution is 80ms. The characteristics of the 16L LIDAR, used to detect the objects, are illustrated in Table I bellow.

Table I. LIDAR characteristics

Features
Time of flight distance measurement with calibrated reflective
16 channels
Measurement range up to 100m
Accuracy +/- 3cm
Dual returns
Field of view (vertical): 30° (+15° to -15°)
Angular resolution (vertical): 2°
Field of view (horizontal/azimuth): 360°
Angular resolution (horizontal/azimuth): 0.1° - 0.4°
Rotation rate: 5 - 20 Hz

We have evaluated the proposed solution on 3000 frames, having objects with multiple classes.

To indicate the performance of a tracker, MOTA combines true positives, true negatives and *ID* switch (28). By t we indicate the timestamp and by *GT* we refer to the ground truth. The value of MOTA can also be negative if the number of errors exceeds the number of good object detections.

$$MOTA = 1 - \frac{\sum_t FN_t + FP_t + IDSW_t}{\sum_t GT_t} \quad (28)$$

The MOTP metric, on the other hand refers to the averaged differences between true positives and ground truth. It gives the average overlap between the correctly identified tracks and the detected objects.

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_i c_t} \quad (29)$$

In (29), c_t denotes the amount of tracker target match in frame t and $d_{t,i}$ is the bounding box overlap between tracked target i and the ground truth object. The scores of the evaluation are displayed in table II. The evaluation sequence has been recorded in the VW campus. It contains sequences in clear and rainy weather. The implemented solution is part of a bigger pipeline and it has been tailored to suit the needs of the general pipeline.

Table II. Tracking results

Metrics	**Value**
MOTA	86.86 %
MOTP	85.39 %
IDSW (sum)	130
Total Frames	3000
Total Objects	14317

The results indicate a relatively high degree of accuracy and precision for the tracker. The highest miss-rate as has been observed for fragmented objects, which present sporadic fluctuations with respect to their dimensions, fragmentation,

semantic class and position. It is important to mention the fact that the quality of the tracker depends on the quality of the segmentation and the input. In table III a comparison is presented with the traditional GNN and JPDA algorithm on the same dataset.

Table III. Tracking comparison

Name	MOTA (%)	MOTP (%)
Proposed Solution	86.86	85.39
JPDA	78.3	77.5
GNN	72.13	70.84

In figure 5 the measurements are illustrated with blue color and the corresponding hypothesis are depicted with red. The height of all the cuboids is received from the segmentation module and it is the same for all objects (2m).

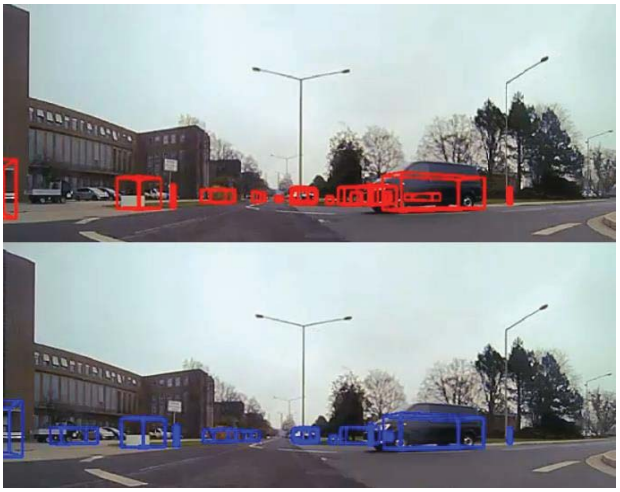

Fig. 5. Measurements and corresponding tracks

In figure 6 we observe the motion vector of an incoming vehicle as well as its trace in the right image. The ID of each object has been depicted with a different color in the right-hand side image.

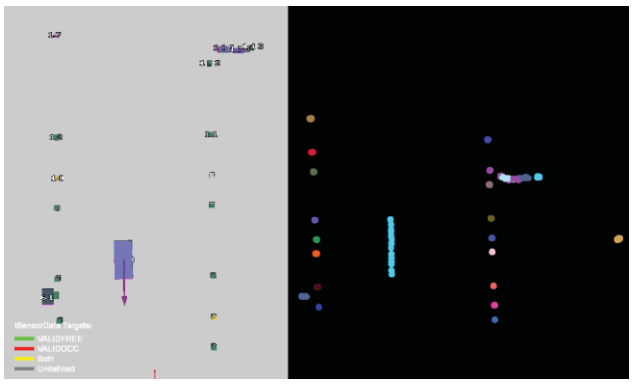

Fig. 6. Motion vector and target ID trail

Figure 7 depicts another scenario, when the ego vehicle is at an intersection near a parking lot. In the upper part of the

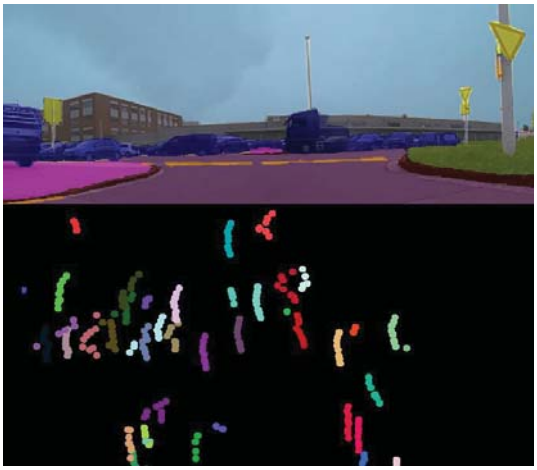

Fig. 7. Semantic Segmentation image and target ID trails

figure the segmentation image is overlapped over the intensity image. The lower part of the image graphically depicts the trails left by the targets that are approaching the ego vehicle with a speed similar to the ego but with negative sign.

Figure 8 illustrates the 3D cuboid measurements and targets with the corresponding motion vectors of the same scenario depicted as in figure 7.

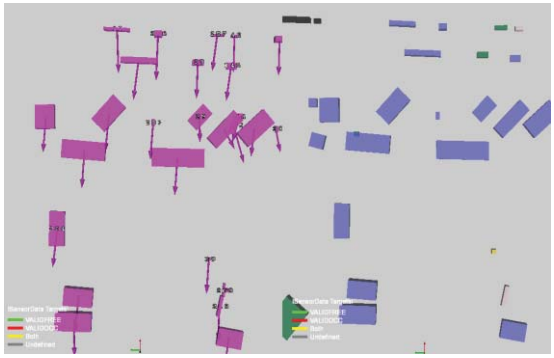

Fig. 8. Targets and measurements in a crowded intersection

Comparison with state-of-the-art methods is very difficult due to the fact that the proposed solution has been tailored to work with a particular input type used in the current project. Furthermore, the proposed method is implemented as a generic solution, being able to track any type of cuboid as long as it comes in the correct format, and contains all the required data.

V. Conclusion

This paper presented a novel multi-object tracking framework based on a Markov decision process, where the lifetime of an object is modeled with five states (Initialized, Processed, Updated, Drifting and Absolute Death). Furthermore, we presented a data association affinity function, which is based on multiple aggregated features like color, semantic class, geometric properties, orientation and positional

descriptor. The proposed data association score is able to make a good differentiation between objects that are similar or clustered even when some of the extracted features are inconsistent over consecutive frames. Two motion models were used to deal with the motion uncertainty and to describe the motion behavior of the tracks, the CTRV and the CV models. Our last contribution in this paper was a module for filtering the meta parameters like object dimensions and orientation. The rationale for making a separate module for the meta parameters is that, due to the violent fluctuations of meta parameters, variables that depend on these parameters would start to fluctuate as well which would lead to more problems overall. Therefore, a combination of the tracked meta parameters and the inferred ones using the two motion models offered a better result. The proposed solution runs in real time and was evaluated using MOTA and MOTP object tracking metrics achieving good results.

ACKNOWLEDGMENT

This work was supported by the "Automated Parking and Driving - **UP-Drive"**, EU H2020, project under grant nr. 688652, "Integrated Semantic Visual Perception and Control for Autonomous Systems – **SEPCA"**, CNCS- UEFISCDI, PN-III-P4-ID-PCCF-2016-0180, grant no. 9/2018 and "Multispectral environment perception by fusion of 2D and 3D sensorial data from the visible and infrared spectrum – **MULTISPECT"**, CNCS - UEFISCDI, PN -III -P4 -ID -PCE-2016-0727, grant no. 60/2017

REFERENCES

[1] B. Fortin, R. Lherbier and J. Noyer, "A Model-Based Joint Detection and Tracking Approach for Multi-Vehicle Tracking With Lidar Sensor," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 1883-1895, Aug. 2015

[2] W. Mei, G. Xiong, J. Gong, Z. Yong, H. Chen and H. Di, "Multiple moving target tracking with hypothesis trajectory model for autonomous vehicles," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, 2017, pp. 1-6.

[3] L. Schneider et al., "Semantic Stixels: Depth is not enough," 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, 2016, pp. 110-117

[4] Muresan, Mircea Paul, Sergiu Nedevschi and Radu Danescu. "A Multi Patch Warping Approach for Improved Stereo Block Matching." VISIGRAPP (2017).

[5] D. Nuss, T. Yuan, G. Krehl, M. Stuebler, S. Reuter and K. Dietmayer, "Fusion of laser and radar sensor data with a sequential Monte Carlo Bayesian occupancy filter," 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, 2015, pp. 1074-1081.

[6] Cheng J, Xiang Z, Cao T, et al. Robust vehicle detection using 3D Lidarunder complex urban environment. Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014: 691-696

[7] Cheng J, Xiang Z, Cao T, et al. Robust vehicle detection using 3D Lidar under complex urban environment. Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014: 691-696.

[8] Himmelsbach M, Wuensche H. Fast segmentation of 3d point clouds for ground vehicles. Intelligent Vehicles Symposium (IV). IEEE, 2010: 560-565

[9] Xu, F., Huang, C.R., Wu, Z.J., and Xu, L.Z. (2011). Video multi-target tracking based on probabilistic graphic model. Journal of Electronics(China), 28(4), 548–557

[10] Benfold, B. and Reid, I. (2011).Stable multi-target tracking in real-time surveillance video. In CVPR, 3457–3464

[11] Y. Bar-Shalom, F. Daum and J. Huang, "The probabilistic data association filter," in *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82-100, Dec. 2009

[12] A. Çakıroğlu, "Tracking variable number of targets with Joint Probabilistic Data Association Filter," *2016 24th Signal Processing and Communication Application Conference (SIU)*, Zonguldak, 2016, pp. 2017-2020.

[13] E. Lee, D. Musicki and T. L. Song, "Multi-sensor distributed fusion based on integrated probabilistic data association," *17th International Conference on Information Fusion (FUSION)*, Salamanca, 2014, pp. 1-7

[14] B. Cheung, M. Rutten, S. Davey and G. Cohen, "Probabilistic Multi Hypothesis Tracker for an Event Based Sensor," *2018 21st International Conference on Information Fusion (FUSION)*, Cambridge, 2018, pp. 1-8

[15] C. G. Hempel, T. Luginbuhl and J. Pacheco, "Performance analysis of Adaptive Probabilistic Multi-hypothesis Tracking with the Metron data sets," *14th International Conference on Information Fusion*, Chicago, IL, 2011, pp. 1-5.

[16] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving", in Autonomous Robots, vol.26, no. 2-3, pp. 123-139, 2009.

[17] X. Zhang, W. Xu, C.Dong, and J.M. Dolan, "Efficient L-shape fitting for vehicle detection using laser scanners", in Proc. of Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, pp. 54-59, 2017

[18] Jackson, J.D.; Yezzi, A.J.; Soatto, S., "Tracking deformable moving objects under severe occlusions," Decision and Control, CDC. 43rd IEEE Conference on , vol.3, no., pp.2990,2995 Vol.3, 14-17 Dec. 2004

[19] 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

[20] Jeongju Choi, Jong Shik Kim, DEFORMABLE OBJECT TRACKING USING ACTIVE CONTOUR MODEL, IFAC Proceedings Volumes, Volume 35, Issue 1, 2002, Pages 31-36,

[21] D. Held, J. Levinson, and S. Thrun, "Precision tracking with sparse 3d and dense color 2d data," in Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013, pp. 1138–1145

[22] Asvadi, Alireza & Girão, Pedro & Peixoto, Paulo & Nunes, Urbano. (2016). 3D object tracking using RGB and LIDAR data. 10.1109/ITSC.2016.7795718

[23] A. Nègre, L. Rummelhard and C. Laugier, "Hybrid sampling Bayesian Occupancy Filter," 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, 2014, pp. 1307-1312.

[24] S. Steyer, G. Tanzmeister and D. Wollherr, "Object tracking based on evidential dynamic occupancy grids in urban environments," 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, 2017, pp.1064-1070

[25] S. Bota and S. Nedevschi, "Tracking multiple objects in urban traffic environments using dense stereo and optical flow," *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, DC, 2011, pp. 791-796.

[26] E. Romera, J. M. Álvarez, L. M. Bergasa and R. Arroyo, "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263-272, Jan. 2018.

[27] B. Sahbani and W. Adiprawita, "Kalman filter and Iterative-Hungarian Algorithm implementation for low complexity point tracking as part of fast multiple object tracking system," *2016 6th International Conference on System Engineering and Technology (ICSET)*, Bandung, 2016, pp. 109-115.

[28] Bertsekas, D.P., & Castañón, D.A. (1989). The auction algorithm for the transportation problem. Annals of Operations Research, 20, 67-96.