GridTrack: Detection and Tracking of Multiple Objects in Dynamic Occupancy Grids

Özgür Erkent, David Sierra Gonzalez, Anshul Paigwar, Christian Laugier

GridTrack: Detection and Tracking of Multiple Objects in Dynamic Occupancy Grids *

Özgür Erkent[1,2 [0000−1111−2222−3333]], David Sierra Gonzalez[1], Anshul Paigwar[1 [0000−0003−4312−5893]], and Christian Laugier[1]

[1] INRIA, Chroma Team, Rhône-Alpes, France.
name.surname@inria.fr
[2] Department of Computer Engineering, University of Hacettepe, Ankara, Turkey
namesurname@hacettepe.edu.tr

Abstract. Multiple Object Tracking is an important task for autonomous vehicles. However, it gets difficult to track objects when it is hard to detect them due to occlusion or distance to the sensors. We propose a method, "GridTrack", to overcome this difficulty. We fuse a dynamic occupancy grid map (DOGMa) with an object detector. DOGMa is obtained by applying a Bayesian filter on raw sensor data. This improves the tracking of the partially observed / unobserved objects with the help of the Bayesian filter on raw data, which has a powerful prediction capability. We develop a network to track the objects on the grid and fuse information from previous detections in this network. The experiments show that the multi-object tracking accuracy is high with the usage of the proposed method.

Keywords: tracking · autonomous vehicles · occupancy grids

1 INTRODUCTION

The ability of the autonomous vehicles (AVs) to navigate safely depends on their comprehension of the environment. For this reason, Advanced Driver Assistance Systems (ADAS) require reliable information about the locations and motion of the obstacles around the AV. Multiple Object Tracking (MOT) is one of the essential tasks for ADAS to estimate the poses of multiple objects for a given time interval. Due to recent advancements in object detection by using a variety of sensors, tracking algorithms heavily rely on object detection; either they use the output of a detection algorithm or integrate a detection algorithm into the tracking method. However, when the tracked object cannot be detected by the object detector, such as due to occlusions or distance to the sensor, the tracking cannot be achieved accurately.

In this work, we aim to increase the accuracy of the tracking performance by tracking objects that cannot be detected by the object detector due to poor observability. To be able to achieve this, we use Dynamic Occupancy Grid Maps (DOGMas) which are constructed by applying Bayesian filters on raw sensor data. Bayesian filters have a high capacity of predicting the occupancy and motion states of the grids, which are not directly observable by the sensors [22]. In MOT, when Bayesian filters are used,

* This work was supported by Toyota Motor Europe.

they are generally applied on the object detections to predict the next location of the detected object ([28, 6, 5]). We are proposing a new approach for MOT by integrating the Bayesian filter on raw sensor data with different off-the-shelf precise object detectors ([13, 26]). This will also result in enriched occupancy grid by integrating tracked object information into the grid.

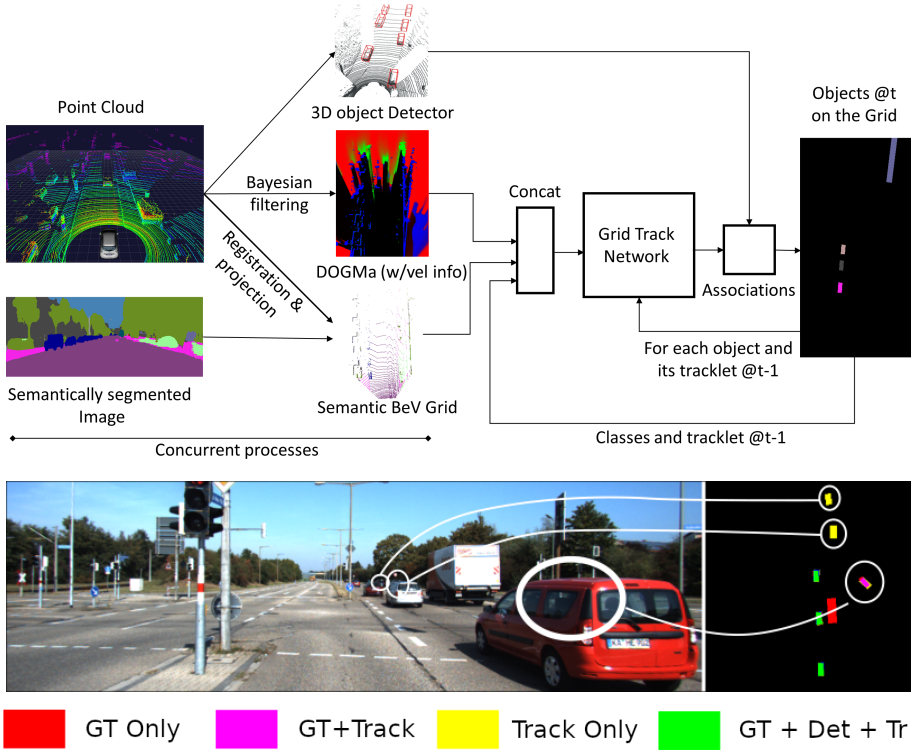


Fig. 1: On top, the general overview of our approach. In the bottom, tracked objects for the given image. GT: Ground truth, Track: Tracked object output from Grid Track Network (GTN), Det: Object output from Object Detector. The circles show the corresponding tracked objects.

The proposed framework is shown in Fig. 1. The GTN takes the input from processed data DOGMa, projected Semantic Bird's eye View (BeV) and output from the previous time step. In addition, the outputs of the GTN are associated with an object detector. The association performs three operations: a) it associates the detected objects from the object detector with the tracked objects from GTN; b) creates new objects; and c) represents tracked objects from GTN even if no corresponding object detection exists. The output of the framework consists of the poses of each object together with their instance id's in the previous frame and their class type. For illustration, we used pointcloud as input for DOGMa construction, but other sensor modalities such as stereo

cameras or radar can also be used as input. We use Conditional Monte Carlo Dense Occupancy Tracker (CMCDOT) to obtain DOGMa which can predict the occupation and motion states of each cell on the grid map [22]. To obtain the projected semantic BeV, we semantically segment RGB images with an off-the-shelf method whose details are given in the Sec. 3, and register the points with the point cloud and project them onto the grid. It should be noted that it is also possible to use other sensor modalities to obtain the semantic grid [9]. The losses are used during training while association is used only during testing. We have evaluated two different object detectors which use pointclouds; however, it should be reminded that an object detector method with other sensor modalities could also be used.

In summary, the contributions of our proposal can be listed briefly as follows:

- The usage of the Bayesian filter via DOGMa on the raw sensor data allows to track the objects that are heavily occluded and not detected by the object detector
- Integration with an off-the-shelf object detector as a module allows it to be used for different object types with different sensor modalities

An example is shown in Fig. 1. On the bottom right, the grid is shown with corresponding tracked objects. Three objects are detected as *car* and shown in green which means that they have labels in GT, Det and Tr. The truck is not detected since the object detector recognizes only cars; therefore it has only GT label. A circled object (pink) on the right side of the grid is not recognized by detector, it is tracked without anticipating the re-detection of the object. GT label also approves our tracker. On the RGB image, it can be observed that this object is heavily occluded; however, the tracker can still estimate its location on the grid. The same is true for two circled (yellow) *cars* at a distance. Note that these two objects are correctly tracked; however, their GTs are not labeled.

We evaluate our method on KITTI MOT dataset. It is shown both qualitatively and quantitatively that the method is capable of achieving high accuracy results.

In Section 2, we shortly review the literature related to MOT. In Section 3, we give a detailed explanation of our method. In Section 4, we show the results of our method on KITTI MOT dataset with comparisons. Finally, we conclude the paper with a brief summary and possible future directions.

2 Related Work

We discuss here the recent progress on MOT studies. Although 3D MOT is more relevant to our approach for AVs, we will cover both 2D and 3D MOT methods, since 3D methods are generally based on ideas from 2D MOT.

Early object trackers used hand-crafted features as points of interest and achieved a fast and robust performance for problems where the cues were evident such as corners, edges, etc. [25, 1]. For a detailed list of features used in trackers, interested readers are referred to [20]. However, these approaches were lacking in performance when the objects did not have sufficient features. As powerful deep learning object detectors emerged, instead of tracking the features, the objects were tracked. These methods, which are called as *tracking-by-detection*, use the output of the object detector and

predict the location of the object through time by associating the outputs of the object detector. One of the earliest works is SORT [4], which used the output of the object detector and a Kalman filter to predict the next state of the tracked object. The object association through time was achieved by Hungarian algorithm. DeepSORT [28] used deep learning re-identification network to distinguish similar appearances of the objects to overcome the Id switch (Ids) problem. Ids can be explained as the miss-switch of the instances of the objects at different time steps. If the object detector can detect the objects accurately for a given domain, the performance of *tracking-by-detection* can also be sufficient [5, 15]; however, the object detector cannot always be accurate for partially observed objects.

In a number of other studies, Bayesian filters, such as the Kalman filter, have been replaced by recurrent neural networks (RNNs). The main reason is the end-to-end training capability of RNNs. An early work is by Sadeghian *et al*. [23] where a separate RNN is used for appearance, motion and re-identification. The advantage is that no extra network training is necessary such as an extra re-identification network and possible implementation difficulties of a Bayesian filter are avoided. Again, although they achieved a superior performance in certain domains [19], they are limited by the accuracy of the object detector.

Tracking-by-detection approaches have also been used in 3D MOT and they are similar to 2D MOT methods. They use a 3D object detection method and make a relation in between detections through time with an association method ([11, 27]). As mentioned, again the temporal data is not used for detection and if an object instance cannot be detected in a few number of frames (e.g. due to heavy occlusion), it is lost.

To be able to overcome the problem of using a single frame without any temporal information for object detection, recent studies focused on joint *detection-and-tracking*. In these approaches, generally more than one frame is provided as input and the detection is not independent of the tracking. In one of the early works, Feichtenhofer *et al*. [10] used two separate convolutional neural networks (CNNs) and multiple frames as input and then associated the outputs. Bergmann *et al*. [2] used a neural network for association of detected objects in a similar method. Center track [30] is one of the recent successful methods which achieved high performance results on benchmarks; however, one of the problems of this approach is that when an object is not detected for any reason, it is initialized as a new object when it is re-detected.

After the successful performance of joint detection-and-tracking in 2D images, a similar approach has been followed for 3D MOT. Early work by Luo *et al*. [18] used pointclouds from different frames similar to [10] and associated the outputs. In another similar study, Guo *et al*. [14] used keyframes instead of using all frames. Recently, Center Point [29] is inspired from the success of center track [30]. There are two main problems with these methods. The first one is that since they are data dependent, their training requires a long time for each new domain and object type. Secondly, if the objects are not detected due to any reason such as heavy occlusion or small scale at a distance, these methods cannot predict the location of the objects.

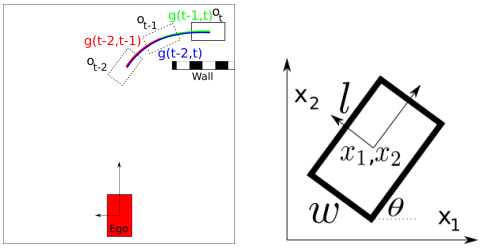
Finally, we will mention the methods where the grid based approach is used for tracking. Engel *et al*. [7] uses DOGMa as input and predicts the objects and their as-

sociations in time. However, the objects cannot be estimated precisely due to the noisy nature of the occupancy grids.

Our proposal is different from the aforementioned approaches in the sense that we continue tracking of the objects, even if the observability is poor due to occlusion or distance, by using the Bayesian filters on the raw sensor data to obtain DOGMa and integrate with a precise object detector for tracking on BeV.

3 METHOD

We formulate the detection and tracking of multiple objects in the grid as follows. We receive information from each sensor S_m at time t, detect objects for a given class type such as *car, cyclist* or *pedestrian* with respect to our ego vehicle and track this object instance for the given time interval $[t, t+T]$ while the object is in the perception range of the ego vehicle. The total number of class types will be denoted with N_c. Each object $o^i_{t+\delta t}$ is defined by its class type c, location $x = (x_1, x_2)$, dimension $d = (w, l)$, orientation θ and its unique instance id i at time $t+\delta t$. Initialy, objects are recognized by the detection method; after initialization they are tracked as explained in this section. Time depends on acquisition time of data from the sensors and will be annotated as discrete afterwards. The range of the tracking is equal to that of the occupancy grid map, which is selected by the user. The tracklet $g(t-k, t) = \{x(t-k), x(t-k+1), \cdots, x(t)\}$ is defined as the location information of an object in the time interval $[t-k, t]$ in this study. The approach is online as it does not use future information. An illustration is shown in Fig. 2 for clarification. An object is shown at three different time steps o_{t-2}, o_{t-1} and o_t. It has three possible tracklets: $g(t-1, t)$; $g(t-2, t)$ and $g(t-2, t-1)$ (trajectory from time step $t-1$ to t; $t-2$ to t; and $t-2$ to $t-1$ respectively).



(a) Range (b) Tracked Object

Fig. 2: On the left, the object is shown in the range of the ego vehicle on the grid for three different time steps: $\{t-2, t-1, t\}$ with all three possible tracklets. On the right, the location, dimension and orientation of the tracked object is illustrated.

The sensor data is processed to obtain a dynamic occupancy grid map, semantic representation of the grid and object detection in the grid. First, we will explain these processes briefly, then how we track objects with this data.

3.1 Bayesian Occupancy Grids

To obtain dynamic occupancy grid maps, we use a Bayesian Filter method. The occupancy probability of each cell of the grid is computed by using the sensor measurements and the previous states of the cells. We will very briefly explain the steps to compute the probability by using CMCDOT [22], which is a successful DOGMa approach. The state of each cell is defined as $\{$*free, statically occupied, dynamically occupied, unknown area*$\}$. First, the current state of the cell is predicted by using the probabilities from the previous states with transition probabilities. Transition probability is a value that represents the probability of transition of a state from one to another. Next, the probabilities from first step are evaluated with a probabilistic sensor model as the observation model. One of the advantages of the sensor models is that they can be used with a variety of sensors including LIDAR, radar, RGB cameras, etc. Once the evaluation is completed, the state distributions are estimated and the cycle is completed with a particle re-sampling step which assigns new particles to new dynamically occupied regions. For each cell with a high dynamic probability, a velocity related information of the motion state is also computed. The output consists of 4 channels for occupancy states and 2 channels for motion states. It should be noted that, although DOGMa does not have explicit object definition, it contains predictions for all grid cells regardless of their occlusion or distance.

3.2 Semantic Grid

Occupancy grids don't contain semantic information. To achieve this, we semantically segment the RGB images by using a deep learning method MobileNetV2 [24] which is adapted for KITTI dataset [8], register each pixel with a point in the pointcloud which is obtained from LIDAR data and project this information onto the grid. Although the data is sparse, it is a fast approximation of semantic grids. It should be noted that semantic grids can also be obtained in a dense manner and they would not require a specific sensor type as shown in [9]. The semantic grid is converted to a 1-hot vector to be used as input by the GTN. It consists of $N_c + 2$ channels where N_c is total number of classes, 1 additional channel to represent cells with no points and 1 channel for cells with unclassified points.

3.3 3D Object Detection

We use two different off-the-shelf 3D object detectors for estimating the 3D poses of the objects in the grid. Both of them use point clouds as input ([13], [26]). We require the following outputs from the detector: the class type of the detected object, its center location on the grid, dimension (width and length) and orientation. It should be reminded that the detector can use any sensor type as long as it provides the required outputs. Furthermore, we use it as an independent module in our framework so that it can be interchanged if necessary, such as to detect another object class, or interchange it with another sensor modality in another data domain.

3.4 Grid Tracking Network

An overview of the method is given in Fig. 3. As explained above, our method is not dependent on a specific sensor modality. Here we illustrate with LIDAR and RGB camera sensor types, which are available in most of the current datasets. Semantic grid is transformed into a 1-hot vector depending on the class types we will use. We use the *car*, *bike* and *pedestrian* classes. DOGMa, semantic grid maps, the class estimations on the grid and the tracklets $g(t-2, t-1)$ from the previous time step $t-1$ are concatenated and they are fed into a ResNet-34 backbone to extract the features. During training, the tracklets from previous steps are provided from the ground truth. The class estimation on the grid is simply the rendered tracking estimation where each cell contains information about the occupying object class type. The class estimations and tracklets have both $N_c + 1$ (class types and background) number of channels. Therefore, ResNet-34 has an input of $3 \times N_c + 10$ channels.

The features are processed by a feature pyramid network (FPN). We omit the details of FPN and refer the interested reader to [17]. The output of FPN is first input to the class convolutions. It has 5 layers each consisting of a convolution with a batch normalization with the corresponding channel sizes, $(192, 64, 16, 16, N_c)$. The output of class convolution is used to find the cross entropy loss for the class estimation on the grid $\mathcal{L}_C$ (Class loss). Then, for each object instance output at time step $t-1$, the output of the class convolution is concatenated with the binary grid estimation at $t-1$ where the cells that contain the corresponding object instance are 1, and 0 otherwise. The instance convolution has 12 convolution layers. The initial convolution layers consist of 8 channels. The instance convolution is applied for each instance at $t-1$ and N_I outputs are obtained for time step t where N_I is the number of instances at $t-1$. Therefore, for each object instance at $t-1$, their new location prediction is obtained at t. Finally, an instance loss (Object ID loss) is computed as the binary cross entropy loss for each instance estimation, $\mathcal{L}_I = \sum_i^{N_I} \mathcal{L}_i$; therefore the total loss is defined as $\mathcal{L} = \mathcal{L}_C + \mathcal{L}_I$.

This output provides the coarse estimations at t for each instance from time step $t-1$. Finally, an association algorithm is applied that fuse coarse estimations with the outputs of the 3D Object detector at t.

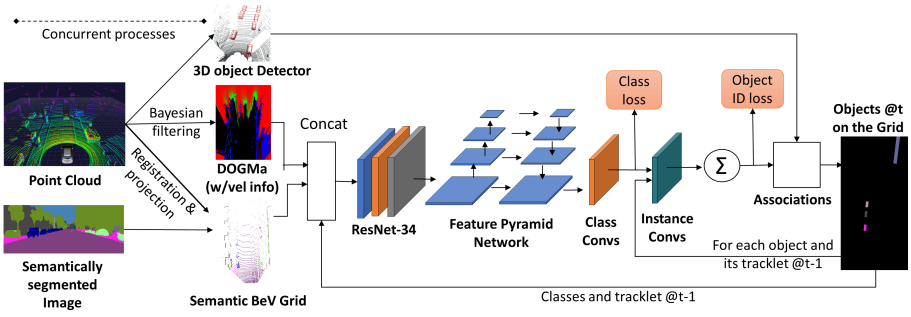


Fig. 3: Overview of the proposed framework.

Algorithm 1: Object Association

Definitions: $ct_{tr}(i)$: Estimation center of i^{th} object from assocation, $ct_{est}(i)$: Estimation center from detection network, $ct_{GTN}(i)$: Estimation center from GTN, τ_d: Distance threshold; $Id(i)$: Id of object i

Initialize $M = |ct_{GTN}|$; $N = |ct_{est}|$;

create simmat:

for $m = 1, \cdots, M$ **do**
  for $n = 1, \cdots, N$ **do**
    if $||ct_{GTN}(m) - ct_{est}(n)|| > \tau_d$ **then**
      $simmat(m, n) = \infty$
    else
      $simmat(m, n) = ||ct_{GTN}(m) - ct_{est}(n)||$

apply Hungarian algorithm:

$assoc = H(simmat)$

for $n = 1, \cdots, N$ **do**
  if $ct_{est}(n) \in assoc$ **then**
    $m = assoc(n)$
    $ct_{tr}(m) = ct_{est}(n)$
    $Id(m) = m$
  else
    $ct_{tr}(max(Id) + 1) = ct_{est}(n)$
    $Id(max(Id) + 1) = max(Id) + 1$

for $m = 1, \cdots, M$ **do**
  if $ct_{GTN}(m) \notin assoc$ **then**
    $ct_{tr}(m) = ct_{GTN}(m)$
    $Id(m) = m$

return ct_{tr}, Id

3.5 Association

We perform three operations in association; a) starting a new instance seed from 3D detection; b) associating an existing instance with the 3D detection; c) tracking the object detected by GTN but not by the object detector. Note that association is not used during training.

The algorithm is given in Algorithm.1. For each object detected by object detector and by GTN, the centers are found by simple morphological operations, ct_{est} and ct_{GTN}. A similarity matrix is obtained which represents the distance between the distances of the object detector and GTN detections. For association of the objects, we use Hungarian algorithm with a parameter which asserts that the distance between the centers of the associated objects should be smaller than a threshold τ_d. If the object found by the object detector is not associated with the detection from GTN, then it is initiated as a new instance. If GTN detection is not associated with an object detector detection, then it is included in the tracked objects list with the location found by GTN. The final output is the centers of the tracked objects with their Ids.

Removing of instances is automatic due to instance convolution layer of the proposed network which outputs no objects for objects that are not in the grid anymore; therefore, we don't have a specific algorithm for removal process.

3.6 Training

During training, we use the Ground Truth object locations instead of the 3D Object detector. For the semantic segmentation, we use the output of the off-the-shelf segmentation algorithm. We use SGD with a learning rate of 0.001 for optimization.

4 EXPERIMENTS

We have evaluated our approach on KITTI MOT dataset training data which is collected by using a vehicle equipped with various sensors [12]. We have used 15 sequences, 6523 frames, 20199 instances of car, 8304 instances of pedestrians and 1088 instances of cyclists for training; 6 sequences, 1481 frames, 7093 instances of car, 3166 instances of pedestrians and 850 instances of cyclists for validation. It has RGB camera, LIDAR and corresponding calibration parameters for the sensors and GPS data for the vehicle localization. The training and validation is made on Nvidia GTX 1080Ti. First, we give our qualitative results. Then, in the second part, we provide comparative results against other methods.

4.1 Qualitative Results

Fig. 4: Qualitative analysis.

We show two sequences of images through time in Fig. 4 and Fig. 1 - bottom. On the left, RGB images are shown with circles for some of the detected objects to show their corresponding labels on the grid. On the right, the grids are shown with object labels. "GT only" means that the object is neither detected nor tracked but a GT label is provided in the dataset. "GT+Track" means that the object has a GT label and it is tracked by GTN although it was not detected by the object detector. "Track only" means there is a track of an object but there is no detection or GT label. "GT + Det + Tr" means that the object is in GT and it is detected and tracked while "Det + Tr" means that it is not given in GT, but an object is detected and tracked at that location.

On top image, 4 objects have correct common detection and tracking with a GT ("GT+Det+Tr"). A car on the lower left which is partially observable is tracked, but its location is estimated wrong. Two objects on the left and one on the upper right of the grid are detected and tracked, but they have no GT. The van is not detected; therefore it is not tracked yet. The top image is four time steps before ($t-4$) the image in the *middle*.

The *middle image* has 5 common GT+Track+Detections. As it can be seen, one of the cars which was detected in the $t-4$ frame is now also given in the GT. Two cars on the left and right side of the grid are not detected, but they are still tracked since they were detected in the $t-4$ frame. The van is tracked since it was probably detected as a car in between frames t and $t-4$. A car has GT, but it has not been detected yet on the right side of the grid.

In the bottom, two previous time steps $t-2$ of the image in the bottom of Fig. 1 is given. Here, the car on the far away right is observable and it is detected, tracked with a GT. The circled far away three cars are tracked even if they are not detected by the object detector. The most far away object is not tracked anymore on frame t in Fig. 1 since it is out of view. The truck is not detected nor tracked since the object detector is not trained for trucks.

These images help us to visualize that the objects can be tracked even if they are not detected due to heavy occlusion, distance, or any other reason. We attribute this mainly to the Bayesian filter process on the raw data, which has a strong predictive capacity for unobserved regions. However, as it can be seen, the ground truth of KITTI MOT dataset does not contain all the labels for the cars, cyclists and pedestrians available in the environment as it has been reported by other studies ([27, 21]). When the algorithm is used directly on this dataset, the false positives increase tremendously. To overcome this problem, we include an additional condition in our algorithm. We keep track of all object centers in Algorithm. 1, but report only the ones whose centers are close to the estimated tracked instance in the previous frame, $||ct_{tr}^{t} - ct_{tr}^{t-1}|| < \tau_{det}$ if it is not a new seed. This reduces false positives significantly.

4.2 Quantitative Results

For quantitative analysis, we have compared two different object detection methods, different grid sizes for our proposal and another tracking method on KITTI MOT for three object types; cars, pedestrians and cyclists. For the object detection, we have included two 3D object detection methods; one of them is GridPillars (GP) which uses a modified version of PointPillars [16] for occupancy grids ([13]). The other one is

PointRCNN (PR) ([26]). Both of them are reported to achieve high accuracy results for 3D object detection.

The results for the object detection without tracking is given in Table. 1. Here, we project the 3D bounding boxes onto the grid and assume that an object is correctly estimated if the projected bounding boxes of the ground truth and estimation overlaps more than 50 %. The results are given for a single threshold which gives the best result for AB3DMOT tracking method [27]. Here, we give the TP (True Positive), FP (False Positive), FN (False Negative) and precision and recall results for each object type. These will be useful in interpreting the results for tracking. One of the largest performance differences is in Cyclists, where [26] is more successful for both precision and recall.

Table 1: Object Detection Accuracy

Det.	**Object**	**TP**	**FP**	**FN**	**Prec**	**Recall**
PR [26]	Car	6022	1007	1071	0.8567	0.8490
GP [13]	Car	5799	383	1294	0.9380	0.8176
PR [26]	Ped.	1830	41	1336	0.9781	0.5780
GP [13]	Ped.	2470	402	696	0.8600	0.7802
PR [26]	Cyclist	720	84	130	0.8955	0.8471
GP [13]	Cyclist	363	80	487	0.8194	0.4271

Next, we compare two tracking methods with two object detection methods and two different grid sizes as shown in Table. 2-4 for Cars, Pedestrians and Cyclists respectively.

We include Multi-object tracking accuracy (MOTA) which is defined as $MOTA = 1-(FP+FN+IdS)/GT$ and Id switches (IdS) [3]. An object is correctly tracked only if it is correctly detected with the correct ID; therefore, MOTA also includes accuracy for location, dimension and orientation of the objects. The size depicts the number of grids. The range of the grid is same for both grid size ($92 \times 62m^2$), only the resolution doubles which is 0.15 m and 0.3 m respectively. The ego vehicle is in the middle lower edge of the grid. Size is Not Applicable (NA) for AB3DMOT [26] since it directly uses the exact location of the objects. For our method, after we detect and track the object on the grid, we find its center and bounding box, and convert it to the coordinates of the KITTI MOT for comparison. Since we find the objects in the grid, which is discrete, we have some error due to this. For small objects, the results are expected to improve on larger sized grids since the effect of the grid discretization will reduce.

First, we consider the cars. The MOTA measure for GridTrack is slightly higher for both object detection methods. It should be noted that AB3DMOT is a traditional tracking method and its success is limited by the detection accuracy. As it can be observed, the tracking performance of AB3DMOT is already close to object detection accuracy. For example for object detection method [13], the number of TPs is already 5790 for tracking, which is very close to 5799 TPs of the original object detection. It also reduces FPs. However, GridTrack has the capacity of increasing the number of TPs which reduces the FNs. Although the IdS and FP is higher for GridTrack, due to its increased performance in TPs, MOTA also increases. GridTrack achieves this higher TP number

by estimating objects even when they are not detected by using the GTN. The high IdS and FP values can be related to the track predictions of instances which have very close centers for different instances in consecutive frames. Another point of evaluation is the size of the grid. The value of MOTA is slightly higher for smaller grid size, which shows that for large objects such as cars, a smaller grid size can be selected without losing performance in the accuracy.

Table 2: Tracking Methods for Car in KITTI MOT

Tracker	**Size**	**MOTA**	**TP**	**FP**	**FN**	**IdS**
GridTrack + [26]	207x307	**0.7609**	5754	325	1339	32
GridTrack + [26]	415x615	0.7555	5760	329	1333	72
AB3DMOT + [26]	NA	0.7545	5970	618	1123	0
GridTrack + [13]	207x307	**0.7751**	5998	474	1095	26
GridTrack + [13]	415x615	0.7736	6015	462	1078	66
AB3DMOT + [13]	NA	0.7727	5790	308	1303	1

Next, we consider the pedestrians. The pedestrians are much smaller than the cars. The TPs are increased for [26] object detection. The MOTA increase is slightly higher in this case. The results are improved for the larger grid size in this case, which is expected since the pedestrians have a small size and a larger grid size reduces the errors in discretization. TPs are increased in PR-based detection [26] with the usage of GTN.

Table 3: Tracking Methods for Pedestrian in KITTI MOT

Tracker	**Size**	**MOTA**	**TP**	**FP**	**FN**	**IdS**
GridTrack + [26]	207x307	0.5648	1879	49	1287	42
GridTrack + [26]	415x615	**0.5774**	1936	93	1230	15
AB3DMOT + [26]	NA	0.5442	1762	39	1404	0
GridTrack + [13]	207x307	0.6412	2261	195	905	36
GridTrack + [13]	415x615	**0.6652**	2328	214	838	8
AB3DMOT + [13]	NA	0.6595	2383	294	783	1

Finally, we consider the cyclists. Their MOTA performance improves significantly, probably due to less number of cyclists in the scenes. The approach does not confuse the cyclists with each other and achieves a high performance. The size of the cyclists is also large; therefore, the performance is similar for both grid sizes. GTN successfully detects cyclists even if they are not detected by detection algorithm.

From these results it can be stated that GridTrack approach can be used with an object detector to achieve a SOTA performance on the multi-object tracking task.

Table 4: Tracking Methods for Cyclist in KITTI MOT

Tracker	Size	MOTA	TP	FP	FN	IdS
GridTrack + [26]	207x307	0.7576	686	38	164	4
GridTrack + [26]	415x615	**0.7659**	688	36	162	1
AB3DMOT + [26]	NA	0.6471	618	68	232	0
GridTrack + [13]	207x307	0.3294	362	80	488	2
GridTrack + [13]	415x615	**0.3294**	363	80	487	3
AB3DMOT + [13]	NA	0.2929	322	73	528	0

5 CONCLUSION

In this study, we have proposed a method which obtains a DOGMa by applying a Bayesian filter on raw sensor data and we integrated this information with a precise 3D object detector to track the objects even when they were hardly observable. We have used point clouds and RGB images to realize the framework; however, as it has been mentioned, other sensor modalities can also be used. The results suggest that the framework is capable of detecting partially observed objects even when they are not detected by an object detector. As part of the future work, we plan to integrate appearance classification into our framework so that the number of ID Switches are reduced. Another interesting direction would be to test the method with other sensor modalities.

ACKNOWLEDGMENT

We thank Gabriel Othmezouri, Jérôme Lussereau and Lukas Rummelhard for their assistance in this study. Parts of the experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

References

1. Babenko, B., Member, S., Yang, M.h., Member, S.: Robust Object Tracking with Online Multiple Instance Learning **33**(8), 1619–1632 (2011)
2. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: ICCV (2019)
3. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The CLEAR MOT metrics. Eurasip Journal on Image and Video Processing **2008** (2008)
4. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: ICIP. pp. 3464–3468. IEEE (2016)
5. Chandra, R., Bhattacharya, U., Randhavane, T., Bera, A., Manocha, D.: RoadTrack: Realtime Tracking of Road Agents in Dense and Heterogeneous Environments. In: ICRA. pp. 1270–1277 (2020)
6. Ebert, J., Gumpp, T., Münzner, S., Matskevych, A., Condurache, A.P., Gläser, C.: Deep Radar Sensor Models for Accurate and Robust Object Tracking. In: ITSC. pp. 8–13 (2020)

7. Engel, N., Hoermann, S., Henzler, P., Dietmayer, K.: Deep Object Tracking on Dynamic Occupancy Grid Maps Using RNNs. In: ITSC (2018)
8. Erkent, O., Laugier, C.: Semantic segmentation with unsupervised domain adaptation under varying weather conditions for autonomous vehicles. IEEE Robotics and Automation Letters **5**(2), 3580–3587 (2020). https://doi.org/10.1109/LRA.2020.2978666
9. Erkent, O., Wolf, C., Laugier, C., Gonzalez, D., Cano, V.: Semantic grid estimation with a hybrid bayesian and deep neural network approach. In: IEEE IROS. pp. 888–895 (2018)
10. Feichtenhofer, C., Pinz, A., Zisserman, A.: Detect to Track and Track to Detect. In: ECCV. vol. 14, pp. 709–736 (2017)
11. Frossard, D., Urtasun, R.: End-to-end Learning of Multi-sensor 3D Tracking by Detection. In: ICRA. pp. 635–642 (2018)
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR. pp. 3354–3361. IEEE (2012)
13. González, D.S., Paigwar, A., Erkent, Ö., Dibangoye, J., Laugier, C.: Leveraging dynamic occupancy grids for 3d object detection in point clouds. In: 16th IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV) (2020)
14. Guo, X., Huang, K.: 3D Object Detection and Tracking on Streaming Data. In: ICRA. pp. 8376–8382 (2020)
15. Khalkhali, M.B., Vahedian, A., Yazdi, H.S.: Vehicle tracking with Kalman filter using online situation assessment. Robotics and Autonomous Systems **131**, 103596 (2020)
16. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: CVPR. pp. 12697–12705 (2019)
17. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. pp. 2117–2125 (2017)
18. Luo, W., Yang, B., Urtasun, R.: Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net. In: CVPR (2018)
19. Maksai, A., Fua, P.: Eliminating Exposure Bias and Loss-Evaluation Mismatch in Multiple Object Tracking. In: CVPR. pp. 4639–4648 (2019)
20. Miah, M., Pepin, J., Saunier, N., Bilodeau, G.A.: An empirical analysis of visual features for multiple object tracking in urban scenes. In: ICPR (2020)
21. Pöschmann, J., Pfeifer, T., Protzel, P.: Factor Graph based 3D Multi-Object Tracking in Point Clouds. In: IROS. pp. 10343–10350 (2020)
22. Rummelhard, L., Nègre, A., Laugier, C.: Conditional monte carlo dense occupancy tracker. In: ITSC. pp. 2485–2490. IEEE (2015)
23. Sadeghian, A., Alahi, A., Savarese, S.: Tracking The Untrackable: Learning To Track Multiple Cues with Long-Term Dependencies. In: ICCV (2017)
24. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: CVPR. pp. 4510–4520 (2018)
25. Shi, J., Tomasi, C.: Good Features. In: CVPR. pp. 593–600 (1994)
26. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: CVPR. pp. 770–779 (2019)
27. Weng, X., Wang, J., Held, D., Kitani, K.: AB3DMOT: A Baseline for 3D Multi-Object Tracking and New Evaluation Metrics. In: IROS. pp. 10359–10366 (2020)
28. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: ICIP. pp. 3645—-3650 (2017)
29. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. CVPR (2021)
30. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking Objects as Points. In: ECCV (2020), http://arxiv.org/abs/2004.01177