

Factor Graph based 3D Multi-Object Tracking in Point Clouds

Johannes Pöschmann, Tim Pfeifer and Peter Protzel
Dept. of Electrical Engineering and Information Technology
TU Chemnitz, Germany
Email: {firstname.lastname}@etit.tu-chemnitz.de

Abstract—Accurate and reliable tracking of multiple moving objects in 3D space is an essential component of urban scene understanding. This is a challenging task because it requires the assignment of detections in the current frame to the predicted objects from the previous one. Existing filter-based approaches tend to struggle if this initial assignment is not correct, which can happen easily.

We propose a novel optimization-based approach that does not rely on explicit and fixed assignments. Instead, we represent the result of an off-the-shelf 3D object detector as Gaussian mixture model, which is incorporated in a factor graph framework. This gives us the flexibility to assign all detections to all objects simultaneously. As a result, the assignment problem is solved implicitly and jointly with the 3D spatial multi-object state estimation using non-linear least squares optimization.

Despite its simplicity, the proposed algorithm achieves robust and reliable tracking results and can be applied for offline as well as online tracking. We demonstrate its performance on the real world KITTI tracking dataset and achieve better results than many state-of-the-art algorithms. Especially the consistency of the estimated tracks is superior offline as well as online.

I. INTRODUCTION

Robust tracking of multiple moving objects is a crucial component of urban scene understanding. Advances in autonomous driving are based on a reliable perception of dynamic objects around the ego-vehicle. Therefore, fast and robust 3D online multi-object trackers (MOT) are required. Another use case for 3D-MOT in urban scene understanding is the generation of reference data. This requires an even more accurate and reliable tracking to support or replace labeling by humans, but can be applied offline.

Due to recent advances in 2D and 3D object detection [1, 2, 3], the majority of multi-object tracking algorithms follow the tracking-by-detection paradigm [4, 5, 6, 7]. The key challenge in this approach is the data association step, where new detections get assigned to existing tracks. Offline methods attempt to find the global optimal solution, for example through the use of min-cost flow algorithms [4] over the whole sequence. Online tracking on the other hand must be fast and cannot wait for future detections. Therefore, a common approach is to consider only detections of the current frame and assign them directly via the Hungarian algorithm [5, 8].

Since data association is the most crucial step for current tracking-by-detection algorithms, a common way is to formulate an assignment strategy that is as robust as possible. In contrast to this approach, we propose a novel 3D multi-object tracking algorithm, which does not rely on explicit

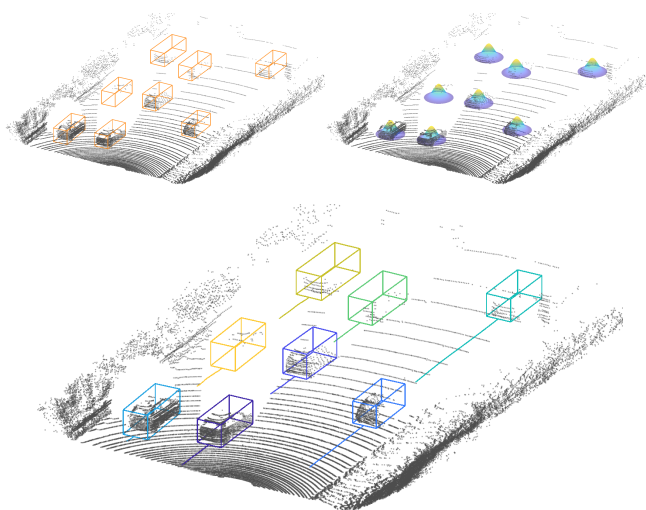


Fig. 1. Visualization of the data flow in our algorithm. We use an off-the-shelf 3D object detector to obtain 3D bounding boxes (top left). In our algorithm, each detection is represented as a Gaussian distribution (top right). The outcome of our 3D multi-object tracker are optimized tracks in 3D space (bottom).

data association. Instead, we propose a robust optimization back-end which is able to solve the assignment problem implicitly and jointly with the state estimation. Based on the 3D bounding boxes from a neural network [3], we create a Gaussian mixture model (GMM) to represent all detections simultaneously. By using a factor graph formulation, we can apply the Max-Mixture [9] to represent the full GMM in the state estimation process of each object. In combination with inter-object constraints and a simple motion model, the full 3D position of each object can be estimated without solving the assignment problem explicitly. This approach is motivated by the wide use of factor graphs in robotics for SLAM [10] and general sensor fusion [11].

Our main contributions are:

Hybrid online/offline tracker: Our 3D multi-object tracker can be applied for offline as well as online tracking without any adaptations due to the flexible structure of factor graphs and their ability to keep track of the entire history of all states. We demonstrate in our experiments, that the offline solution achieves better results by considering all detections of a sequence.

Implicit data association: We do not need an explicit data association step since it is managed inside the factor graph as part of the optimization problem. Furthermore, this association

is not fixed and can be changed during the optimization process, in contrast to most online trackers.

Optimization of state positions: We do not only estimate the association between states and detections, as min-cost flow approaches do, but also optimize the state positions over the whole sequence to deal with inaccuracies of the object detector.

In order to verify the ideas behind our novel approach and to demonstrate its capability for multi-object tracking, we conduct real-world experiments on the challenging KITTI MOT benchmark [12]. The proposed algorithm achieves accurate, robust and reliable tracking results and performs better than many state-of-the-art algorithms. Especially the consistency of the estimated tracks is superior in the offline as well as in the online case. Considering the simplicity of our approach, these excellent results demonstrate the effectiveness of factor graph based 3D multi-object tracking.

II. RELATED WORK

A. 3D Object Detection

Reliable and accurate object detection is a crucial component of tracking algorithms, which follow the tracking-by-detection paradigm. Prior work in the domain of 3D object detection can be roughly categorized into three classes. The algorithms proposed in [13] and [14] use only 2D images to directly predict 3D object proposals using neural networks. Another common approach is the combination of 2D images and 3D point clouds through neural networks to obtain 3D bounding boxes [15, 16]. Methods of the third category solely rely on 3D point clouds and either project them to a 2D bird's-eye view [17], represent them as voxels [18] or directly extract 3D bounding boxes from raw point clouds [3].

B. Multi-Object Tracking

Following the common tracking-by-detection approach, offline trackers try to find the global optimal solution for the data association task over whole sequences. Typical methods are min-cost flow algorithms [4], Markov Chain Monte Carlo [19] or linear programming [20]. In contrast, online tracking algorithms only rely on past and current information and need to be real-time feasible. Common are filter-based approaches like Kalman [5] or particle filter [21]. The data association step is often formulated as a bipartite graph matching problem and solved with the Hungarian algorithm [5, 8]. In the domain of 3D multi-object tracking a lot of work focuses on neural network based approaches, especially end-to-end learned models like [22, 23]. Both jointly learn 3D object detection and tracking. The authors of [23] rely solely on point clouds as input, while [22] utilizes camera images and point clouds. Another approach is the incorporation of neural networks into filter-based solutions. The authors of [24] use 2D images as input for a deep neural network and combine it with a Poisson multi-Bernoulli mixture filter to obtain 3D tracks.

C. Factor Graphs

Although factor graphs are widely used in the field of robotics for [10, 11], they are not common in the tracking

community. The authors of [25] focus on solving the data association for 2D cell tracking, but do not optimize the cell positions. In [26] the data association for multi-object tracking is solved using a factor graph in a 2D simulation, but an extended Kalman filter is applied for track filtering and prediction. Therefore, the approach is not able to change the initial data association in past states. Other work focuses on tracking with multiple sensors [27] or between multiple agents [28] and solves it via particle-based belief propagation in 2D space. Their evaluation however, is limited to simulations and an application to real world use cases is unclear.

In contrast to prior work, we want to introduce online/offline capable, factor graph based multi-object tracking in 3D space. Furthermore, our approach is able to jointly describe data association and state positions for current and past states in a single factor graph and solve it effectively via non-linear least squares optimization.

III. MULTIMODALITY AND FACTOR GRAPHS

A. Factor Graphs for State Estimation

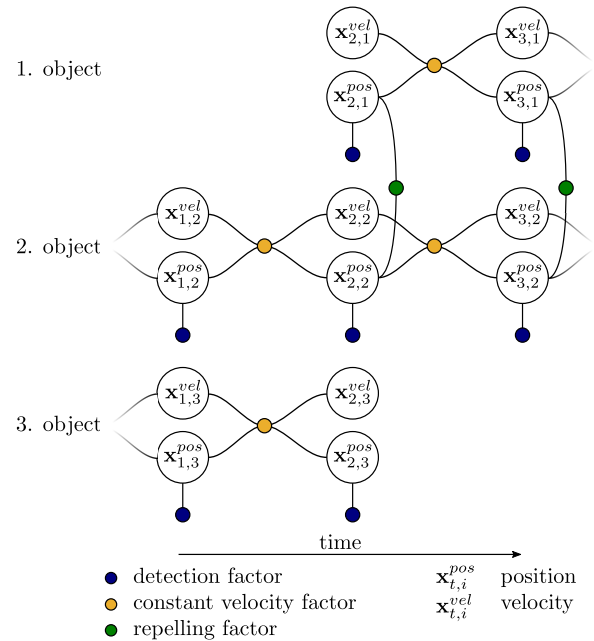


Fig. 2. Example of the factor graph representation of the proposed 3D multi-object tracking algorithm. Small dots represent error functions (factors) that define the least squares problem and big circles are the corresponding state variables. The set of tracked object varies over time due to the appearance and disappearance of objects from the field of view.

The factor graph, as a graphical representation of non-linear least squares optimization, is a powerful tool to solve complex state estimation problems and dominates today's progress in state estimation for autonomous systems [10].

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmax}} \mathbf{P}(\mathbf{X}|\mathbf{Z}) \quad (1)$$

To estimate the most likely set of states \mathbf{X}^* based on a set of measurements \mathbf{Z} , (1) is solved using the factorized conditional

probabilities:

$$\mathbf{P}(\mathbf{X}|\mathbf{Z}) \propto \prod_i \mathbf{P}(\mathbf{z}_i|\mathbf{x}_i) \cdot \prod_j \mathbf{P}(\mathbf{x}_j) \quad (2)$$

Please note that we omit the priors $\mathbf{P}(\mathbf{x}_j)$ in further equations for simplicity. By assuming a Gaussian distributed conditional $\mathbf{P}(\mathbf{z}_i|\mathbf{x}_i)$, the maximum-likelihood problem can be transformed to a minimization of the negative log-likelihood:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_i \frac{1}{2} \left\| \mathcal{I}_j^{\frac{1}{2}} (\mathbf{e}_i - \boldsymbol{\mu}) \right\|^2 \quad (3)$$

Therefore, the estimated set of states $\hat{\mathbf{X}}$ can be obtained by applying non-linear least squares optimization of the measurement function $\mathbf{e}_i = f(\mathbf{z}_i, \mathbf{x}_i)$. Additionally, the mean $\boldsymbol{\mu}$ and square root information $\mathcal{I}_j^{\frac{1}{2}}$ of a Gaussian distribution are used to represent the sensor's error characteristic. State-of-the-art frameworks like GTSAM [29] or Ceres [30] allow an efficient solution for online as well as offline estimation problems.

The major advantage over traditional filter-based solutions is the flexibility to re-estimate past states with the usage of current information. This enables the algorithm to correct past estimation errors and improves even the estimation of current states. Our algorithmic goal is the application of this capability to overcome the limitation of fixed assignments, that filters have to obey.

B. Solving the Assignment Problem

Despite their capabilities, factor graphs are facing the same challenges as filters when it comes to unknown assignments between measurements and states. The assignment can be represented by a categorical variable $\boldsymbol{\theta} = \{\theta_{i,j}\}$ which describes the probability of the j th detection to belong to i th object. To solve the assignment problem exactly, the following integral over $\boldsymbol{\theta}$ has to be solved:

$$\mathbf{P}(\mathbf{X}|\mathbf{Z}) = \int \mathbf{P}(\mathbf{X}, \boldsymbol{\theta}|\mathbf{Z}) d\boldsymbol{\theta} \quad (4)$$

Common filter-based solutions estimate $\boldsymbol{\theta}$ once, based on the predicted states, and assume it to be fixed in the following inference process. This leads to a decreased performance in the case of wrong assignments.

Instead of including wrong assumptions, we formulate the state estimation problem without any assumptions regarding the assignment. Therefore, we assume that $\boldsymbol{\theta}$ follows a discrete uniform distribution. This can be done by describing the whole set of measurements with an equally weighted Gaussian mixture model (GMM):

$$\mathbf{P}(\mathbf{z}_i|\mathbf{x}_i) \propto \sum_{j=1}^n c_j \cdot \exp\left(-\frac{1}{2} \left\| \mathcal{I}_j^{\frac{1}{2}} (\mathbf{e}_i - \boldsymbol{\mu}_j) \right\|^2\right) \quad (5)$$

with $c_j = w_j \cdot \det\left(\mathcal{I}_j^{\frac{1}{2}}\right)$

The GMM encodes that each measurement \mathbf{z}_j with mean $\boldsymbol{\mu}_j$ and uncertainty $\mathcal{I}_j^{\frac{1}{2}}$ is assigned to each state \mathbf{x}_i with the same probability. In our case, the error function is identical with the corresponding state ($\mathbf{e}_i = \mathbf{x}_i$).

Algorithm 1: Tracking Algorithm for the Offline Case

```

generate detections  $\mathbf{Z}$  using PointRCNN [3]
foreach time step  $t$  do
  create GMM based on  $\mathbf{z}_{t,j}$  and null-hypothesis
  if  $t == 0$  then
    | init  $\mathbf{x}_{t,i}^{pos}$  at  $\mathbf{z}_{t,j}^{pos}$  and  $\mathbf{x}_{t,i}^{vel} = [0, 0, 0]$ 
  else
    | propagate  $\mathbf{x}_{t-1,i}^{pos}$  to  $t$ 
    | get correspondence between  $\mathbf{x}_{t,i}$  and  $\mathbf{z}_{t,j}$ 
      | according to  $-\log(\mathbf{P}(\mathbf{z}_{t,j}|\mathbf{x}_{t,i}))$ 
    if  $\mathbf{x}_{t,i}$  does not correspond to any  $\mathbf{z}_{t,j}$  then
      | keep  $\mathbf{x}_{t,i}$  marked as lost or delete it
    end
    if  $\mathbf{z}_{t,j}$  does not correspond to any  $\mathbf{x}_{t,i}$  then
      | init  $\mathbf{x}_{t,i}^{pos}$  at  $\mathbf{z}_{t,j}^{pos}$  and  $\mathbf{x}_{t,i}^{vel} = [0, 0, 0]$ 
    end
  end
  add factors (6), (9) and (10)
  optimize factor graph
end
get association between  $\mathbf{X}$  and  $\mathbf{Z}$ 
if  $\mathbf{x}_{t,i}$  has associated  $\mathbf{z}_{t,j}$  then
  |  $\mathbf{x}_{t,i}^{dim} = \mathbf{z}_{t,j}^{dim}$  and  $x_{t,i}^{conf} = z_{t,j}^{conf}$ 
else
  |  $\mathbf{x}_{t,i}^{dim} = \mathbf{x}_{t-1,i}^{dim}$  and  $x_{t,i}^{conf} = 0$ 
end
if  $\operatorname{mean}(x_{t,i}^{conf} \forall t) < c_{min}$  then
  | delete  $\mathbf{x}_i$ 
end

```

By assigning all measurements to all states, the assignment problem has to be solved during inference by combining all available information. It also allows to re-assign measurements and correct wrong matches with future evidence, which relaxes the requirement of an optimal initial assignment.

Using a GMM as probabilistic model breaks the least squares formulation derived in (3), which is limited to simple single Gaussian models. The authors of [9] proposed an approach to maintain this relationship by approximating the sum inside (5) by a maximum-operator. This allows to reformulate the weighted error function as follows:

$$\left\| \mathbf{e}_{t,i}^{det} \right\|^2 = \min_j \left\| \frac{\sqrt{-2 \cdot \ln \frac{c_j}{\gamma_m}}}{\mathcal{I}_j^{\frac{1}{2}}} (\mathbf{e}_i - \boldsymbol{\mu}_j) \right\|^2 \quad (6)$$

with $\gamma_m = \max_j c_j$

For a detailed explanation of this equation, we refer the reader to our previous work [11] and the original publication [9]. Due to the flexibility of factor graphs, additional information like different sensors or motion models can be added to the optimization problem.

This combination of factor graphs and multimodal probabilistic models allows to formulate a novel inference algorithm as robust back-end for different multi-object tracking applications, that can be applied online as well as offline.

IV. FACTOR GRAPH BASED TRACKING

An overview of our approach to 3D multi-object tracking is given in Fig. 3 and the data flow is shown in Fig. 1. First we apply an off-the-shelf 3D object detector to obtain 3D bounding boxes. Subsequently, all detections are represented as Gaussian distributions with the bounding box center as mean and Σ^{det} to describe the sensor's error characteristic. Our algorithm jointly estimates the state positions in 3D space and solves the data association implicitly as part of the optimization. In a postprocessing step the 3D bounding boxes are reconstructed through the combination of the states 3D positions and the association to the bounding boxes of the detector. Furthermore, we filter out tracks with low confidence. Details are explained in the following sections and the whole algorithm is shown in Alg. 1 for the offline case and in Alg. 2 for the online solution.

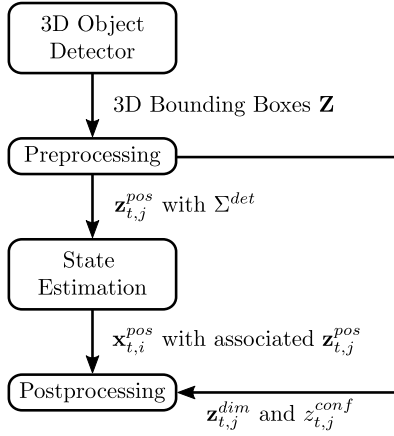


Fig. 3. Block diagram of the introduced algorithm. The bounding box center points and their assignment to detections are optimized using a factor graph approach. The bounding box dimensions and confidence are adopted after the assignment.

A. Detection and Preprocessing

We apply PointRCNN [3] to obtain the 3D bounding boxes \mathbf{Z} , which are composed of the 3D coordinates of the object's center, its dimensions, the rotation of the bounding box and its confidence and is defined at time step t as:

$$\mathbf{z}_{t,j} = \begin{bmatrix} \mathbf{z}_{t,j}^{pos} \\ \mathbf{z}_{t,j}^{dim} \\ \mathbf{z}_{t,j}^{conf} \\ z_{t,j} \end{bmatrix} \quad (7)$$

$$\text{with } \mathbf{z}_{t,j}^{pos} = \begin{bmatrix} z_{t,j}^x \\ z_{t,j}^y \\ z_{t,j}^z \end{bmatrix} \quad \mathbf{z}_{t,j}^{dim} = \begin{bmatrix} z_{t,j}^h \\ z_{t,j}^w \\ z_{t,j}^l \\ z_{t,j}^\theta \end{bmatrix}$$

Subsequently, all detections \mathbf{Z} are transformed into a global coordinate system, which is defined relative to the ego vehicles pose at the first frame of the sequence.

B. State Estimation

The estimated states within the factor graph are composed of the 3D position $\mathbf{x}_{t,i}^{pos}$ of the $i = 1 \dots M$ objects and their

Algorithm 2: Tracking Algorithm for the Online Case

```

generate detections  $\mathbf{Z}$  using PointRCNN [3]
foreach time step  $t$  do
  create GMM based on  $\mathbf{z}_{t,j}$  and null-hypothesis
  if  $t == 0$  then
    | init  $\mathbf{x}_{t,i}^{pos}$  at  $\mathbf{z}_{t,j}^{pos}$  and  $\mathbf{x}_{t,i}^{vel} = [0, 0, 0]$ 
  else
    | propagate  $\mathbf{x}_{t-1,i}^{pos}$  to  $t$ 
    | get correspondence between  $\mathbf{x}_{t,i}$  and  $\mathbf{z}_{t,j}$ 
    | according to  $-\log(\mathbf{P}(\mathbf{z}_{t,j}|\mathbf{x}_{t,i}))$ 
    if  $\mathbf{x}_{t,i}$  does not correspond to any  $\mathbf{z}_{t,j}$  then
      | keep  $\mathbf{x}_{t,i}$  marked as lost or delete it
    end
    if  $\mathbf{z}_{t,j}$  does not correspond to any  $\mathbf{x}_{t,i}$  then
      | init  $\mathbf{x}_{t,i}^{pos}$  at  $\mathbf{z}_{t,j}^{pos}$  and  $\mathbf{x}_{t,i}^{vel} = [0, 0, 0]$ 
    end
  end
  add factors (6), (9) and (10)
  optimize factor graph
  get association between  $\mathbf{x}_{t,i}$  and  $\mathbf{z}_{t,j}$ 
  if  $\mathbf{x}_{t,i}$  has associated  $\mathbf{z}_{t,j}$  then
    |  $\mathbf{x}_{t,i}^{dim} = \mathbf{z}_{t,j}^{dim}$  and  $x_{t,i}^{conf} = z_{t,j}^{conf}$ 
  else
    |  $\mathbf{x}_{t,i}^{dim} = \mathbf{x}_{t-1,i}^{dim}$  and  $x_{t,i}^{conf} = 0$ 
  end
  if  $\text{mean}(x_{t,i}^{conf} \forall t) < c_{min}$  then
    | do not output  $\mathbf{x}_{t,i}$ 
  end
end
  
```

corresponding velocities $\mathbf{x}_{t,i}^{vel}$, both are defined at time step t as:

$$\mathbf{x}_{t,i}^{pos} = \begin{bmatrix} p_{t,i}^x \\ p_{t,i}^y \\ p_{t,i}^z \end{bmatrix} \quad \mathbf{x}_{t,i}^{vel} = \begin{bmatrix} v_{t,i}^x \\ v_{t,i}^y \\ v_{t,i}^z \end{bmatrix} \quad (8)$$

The detection factor is defined at each time step using (6) with the components mean $\boldsymbol{\mu}_j = \mathbf{z}_{t,j}^{pos}$ and a fixed uncertainty $\mathcal{I}_j^{\frac{1}{2}} = (\Sigma^{det})^{-\frac{1}{2}}$ which corresponds to the detector's accuracy. A generic null-hypothesis with a broad uncertainty Σ_0^{det} and mean $\boldsymbol{\mu}_0 = \text{mean}(\mathbf{z}_{t,j}^{pos} \forall j)$ is added to provide robustness against missing or wrong detections. We use a simple constant velocity factor to describe the vehicle's motion:

$$\|\mathbf{e}_{t,i}^{cv}\|_{\Sigma^{cv}}^2 = \left\| \begin{pmatrix} \mathbf{x}_{t,i}^{pos} - \mathbf{x}_{t+1,i}^{pos} \\ \mathbf{x}_{t,i}^{vel} - \mathbf{x}_{t+1,i}^{vel} \end{pmatrix} - \mathbf{x}_{t,i}^{vel} \cdot \Delta t \right\|_{\Sigma^{cv}}^2 \quad (9)$$

Please note, that $\|\cdot\|_{\Sigma}^2$ denotes the Mahalanobis distance with the covariance matrix Σ . To prevent two objects from occupying the same space, we add another simple constraint, which punishes the proximity of two objects. We call this factor the repelling factor:

$$\|\mathbf{e}_{t,n,m}^{rep}\|_{\Sigma^{rep}}^2 = \left\| \frac{1}{\|\mathbf{x}_{t,n}^{pos} - \mathbf{x}_{t,m}^{pos}\|} \right\|_{\Sigma^{rep}}^2 \quad (10)$$

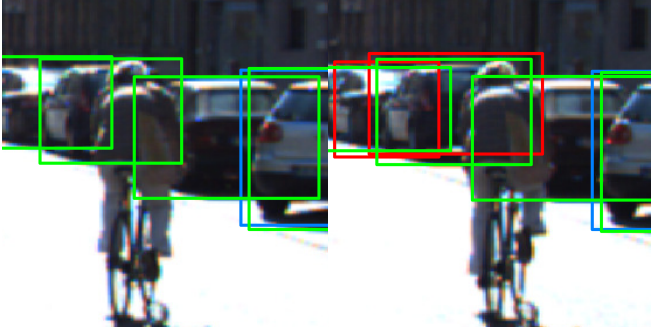


Fig. 4. Limitations of the KITTI Ground Truth: Visualized are *Don't Care* areas (red), *Cars* (blue) and the results of our algorithm for class *Car* (green). Due to the inconsistent labels we erroneously get 4 false positives (frames 109 and 110 from KITTI training sequence 0000).

The overall factor graph consists of one detection factor (6) per estimated object and corresponding constant velocity factors (9) that connect the states of one object over time. Repelling factors (10) are added between object pairs if the euclidean distance is below a defined threshold d_{min} . A visualization of the constructed factor graph is shown in Fig. 2.

Our proposed algorithm is identical for offline and online tracking, except for the postprocessing step (compare Alg. 1 and Alg. 2). In the online use case it has to be done at each time step t . While it can be done once at the end for the offline solution. At each time step t new states can be added (creation of tracks) and existing states can be deleted (death of tracks) or carried over to the next time step (see Fig. 2). In order to create new tracks, we need to find detections $\mathbf{z}_{t,j}$ which do not correspond to any $\mathbf{x}_{t,i}$. Therefore, we create a similarity matrix between all $\mathbf{x}_{t,i}$ (column) and $\mathbf{z}_{t,j}$ (row), including null-hypothesis z_0 , by calculation of $-\log(\mathbf{P}(\mathbf{z}_{t,j}|\mathbf{x}_{t,i}))$. Subsequently, we find the minimum (best similarity) and delete its state (column) from the matrix. If the related measurement (row) is z_0 we mark the state as lost, since it does not correspond to any real measurement $\mathbf{z}_{t,j}$. Otherwise, we also delete the measurement from the matrix, since it has a corresponding state. These steps are repeated until all states are deleted from the matrix. In that way, we can simultaneously find $\mathbf{x}_{t,i}$ which correspond to the null-hypothesis z_0 and $\mathbf{z}_{t,j}$ which do not correspond to any $\mathbf{x}_{t,i}$. A new $\mathbf{x}_{t,i}$ is created for each unrelated $\mathbf{z}_{t,j}$ in order to track it. To suppress the tracking of false positive detections, all \mathbf{x}_i which have less consecutive detections than the defined threshold n_{det} are deleted from the factor graph. Thereby, the algorithm can simultaneously track objects from the first occurrence and suppress false positives in the offline use case. This is not possible for the online solution. Instead, states are only handed to the postprocessing step if they have n_{det} or more consecutive detections. In order to handle missing detections or occlusion of objects, a track \mathbf{x}_i has to be marked as lost (correspond to z_0) for more than n_{lost} consecutive time steps before it is terminated. In this case, the last n_{lost} states $\mathbf{x}_{t,i}$ are deleted, since they do not correspond to real measurements. If a track has a corresponding $\mathbf{z}_{t,j}$

TABLE I
PARAMETERS OF OUR FG-3DMOT ALGORITHM

Parameter Name	Symbol	Value
Detection Covariance	Σ^{det}	$\text{diag} \begin{pmatrix} 0.2 \text{ m} \\ 0.2 \text{ m} \\ 0.2 \text{ m} \end{pmatrix}^2$
Detection Null-Hypothesis Covariance	Σ_0^{det}	$\text{diag} \begin{pmatrix} 100 \text{ m} \\ 100 \text{ m} \\ 1 \text{ m} \end{pmatrix}^2$
Constant Velocity Covariance	Σ^{cv}	$\text{diag} \begin{pmatrix} 0.25 \text{ m} \\ 0.25 \text{ m} \\ 0.25 \text{ m} \\ 0.25 \text{ m s}^{-1} \\ 0.25 \text{ m s}^{-1} \\ 0.25 \text{ m s}^{-1} \end{pmatrix}^2$
Repelling Covariance	Σ^{rep}	0.5 m^2
Repelling Distance Threshold	d_{min}	4 m
Track Confidence Threshold (offline)	c_{min}	3.9
Track Confidence Threshold (online)	c_{min}	3.5
Number Consecutive Detections	n_{det}	2
Num. Con. Null-Hypothesis Detections	n_{lost}	5
Object Permanence (online)	n_{perm}	1

before hitting n_{lost} , all \mathbf{x}_i are kept in the factor graph and the track lives on. Again, this is not possible in the online use case. Instead, tracks \mathbf{x}_i marked as lost are only handed to the postprocessing step for a short duration of n_{perm} time steps t .

C. Postprocessing

After optimization, the factor graph returns the implicitly associated detection for each $\mathbf{x}_{t,i}$, which is either $\mathbf{z}_{t,j}$ or the null-hypothesis z_0 . Based on this, the optimized state positions $\mathbf{x}_{t,i}^{pos}$ are combined with the matched bounding box $\mathbf{x}_{t,i}^{dim} = \mathbf{z}_{t,j}^{dim}$ and confidence $x_{t,i}^{conf} = z_{t,j}^{conf}$ or, in the case of the matched null-hypotheses, with the last bounding box of the same track $\mathbf{x}_{t,i}^{dim} = \mathbf{x}_{t-1,i}^{dim}$ and $x_{t,i}^{conf} = 0$. Subsequently, we delete tracks \mathbf{x}_i with mean confidence below threshold c_{min} for the offline solution (see Alg. 1). In case of online tracking, the data association, bounding box fitting and track management has to be done at each time step t (see Alg. 2). Therefore, we do not delete \mathbf{x}_i from the factor graph if the mean confidence is below threshold c_{min} , since the track can get above c_{min} in the future. Instead, $\mathbf{x}_{t,i}$ is not handed to the postprocessing step and thus not part of the output.

By filtering the online and offline results based on their confidence we can effectively throw away tracks with a high number of matched detections with low confidence, which are most likely false positives, and tracks which are matched primarily to the null-hypotheses. As a result, our algorithm is robust against false positives from the object detector.

V. EXPERIMENTS

A. Setup

In order to evaluate our proposed algorithm, we use the KITTI 2D MOT benchmark [12]. It is composed of 21

TABLE II
RESULTS ON THE KITTI 2D MOT TESTING SET FOR CLASS CAR

Method	MOTA \uparrow	MOTP \uparrow	MT \uparrow	ML \downarrow	IDS \downarrow	FRAG \downarrow	FPS \uparrow
TuSimple [6]	86.62 %	83.97 %	72.46 %	6.77 %	293	501	1.7
MASS [31]	85.04 %	85.53 %	74.31 %	2.77 %	301	744	100.0
MOTSFusion [32]	84.83 %	85.21 %	73.08 %	2.77 %	275	759	2.3
mmMOT [22]	84.77 %	85.21 %	73.23 %	2.77 %	284	753	50.0
mono3DT [33]	84.52 %	85.64 %	73.38 %	2.77 %	377	847	33.3
MOTBeyondPixels [7]	84.24 %	85.73 %	73.23 %	2.77 %	468	944	3.3
AB3DMOT [5]	83.84 %	85.24 %	66.92 %	11.38 %	9	224	212.8
IMMDP [34]	83.04 %	82.74 %	60.62 %	11.38 %	172	365	5.3
aUToTrack [35]	82.25 %	80.52 %	72.62 %	3.54 %	1025	1402	100.0
JCSTD [36]	80.57 %	81.81 %	56.77 %	7.38 %	61	643	14.3
FG-3DMOT (offline)	88.01 %	85.04 %	75.54 %	11.85 %	20	117	23.8
FG-3DMOT (online)	83.74 %	84.64 %	68.00 %	9.85 %	9	375	27.1

training and 29 testing sequences, with a total length of 8008 respectively 11095 frames. For each sequence, 3D point clouds, RGB images of the left and right camera and ego motion data are given at a rate of 10 FPS. The testing split does not provide any annotations, since it is used for evaluation on the KITTI benchmark server. The training split contains annotations for 30601 objects and 636 trajectories of 8 classes. Since annotations are rare for a lot of classes, we only evaluate our algorithm on the car subset.

As previously mentioned, we use PointRCNN [3] as 3D object detector. Since [5] uses the same detector and made the obtained 3D bounding boxes publicly available, we use them for comparability reasons. We normalize the provided confidence to positive values by adding a constant offset. Since our algorithm is robust against false positive detections we achieve the best results by using all available detections. All other parameters are shown in Tab. I.

We construct the graph using the libRSF framework [37] and solve the optimization problem with Ceres [30], using the dogleg optimizer.

Since the KITTI 2D MOT benchmark is evaluated in the image space through 2D bounding boxes, we need to transform our 3D bounding boxes into the image space and flatten them into 2D bounding boxes. Furthermore, we only output 2D bounding boxes which overlap at least 25% with the image in order to suppress detections that are not visible in the image, but in the laser scan.

B. Results

We evaluated our tracking algorithm in online and offline use case on the KITTI 2D MOT testing set. Since offline results are rare and not among the best algorithms on the leaderboard, we compare our online and offline results against the 10 best methods on the leaderboard (accessed February 2020), which are summarized in Tab. II. The used metrics multi object tracking accuracy (MOTA), multi object tracking precision (MOTP), mostly tracked objects (MT), mostly lost objects (ML), id switches (IDS) and track fragmentation (FRAG) are defined in [38, 39]. The proposed algorithm

achieves accurate, robust and reliable tracking results in online as well as offline application, performing better than many state-of-the-art algorithms. In the offline use case, we improve the state-of-the-art in accuracy, mostly tracked objects and the fragmentation of tracks. Especially the fragmentation of our tracks is significantly lower than the previous state-of-the-art, since our algorithm can propagate tracks without any measurements far into the future and truncate them afterwards, if no measurements were associated. Even for the online solution, we achieve low track fragmentation and state-of-the-art id switches. Furthermore, the results demonstrate that our approach benefits from the optimization of past states and the ability to re-assign past detections based on future information, since it achieves considerably higher accuracy and lower fragmentation in the offline application. Although we did not optimize run time in any way, our algorithm is real-time feasible in online as well as offline use case. The computation time is similar in both cases and could be improved significantly, especially for the online use case. We provide a visualization of the results of our offline tracker in the accompanying video, which features KITTI testing sequence 0014¹.

Limitations of the KITTI Ground Truth: During the evaluation of our algorithm on the training split we discovered the limits of KITTI’s 2D bounding box ground truth. As an example we visualized a failure case of the ground truth in Fig. 4. The ground truth uses *Don’t Care* 2D bounding boxes to label regions with occluded, poorly visible or far away objects. Bounding boxes which overlap at least 50% with a *Don’t Care* area are not evaluated. As seen in Fig. 4, the ground truth is not labeled consistently and objects (here cars) are neither labeled *Don’t Care* nor *Car*. Therefore, we get a lot of false positives, since our algorithm can track objects from the first occurrence and from high distances. As a result, we assume that we could achieve significantly higher accuracy (MOTA) with a consistent labeled ground truth.

¹<https://youtu.be/mvZmli4jrZQ>

Limitations of our Algorithm: After the evaluation on the testing set of the KITTI benchmark we discovered one main failure case of our algorithm. In testing sequences 0005 and 0006 the ego-vehicle captures a highway scene, hence it self and all other cars are moving with high velocities. Since our algorithm works in a static 3D coordinate system and we initialize new objects as a general assumption with zero velocity, these cases are hard to capture for our algorithm. Furthermore, such scenes with fast moving cars are not present in the training split, whereby we could not adopt our algorithm to handle these scenes. This issue will be addressed in future work.

VI. CONCLUSION

Our proposed 3D multi-object tracking algorithm is build around three main ideas: Implicit and variable data association using the factor graph formulation, optimization of object positions in 3D space and the application as online and offline solution. In order to prove the capabilities of our novel approach, we conducted real world experiments and achieved state-of-the-art results with a superior tracking consistency. Based on this minimalist approach a vast amount of optimization possibilities are given. Due to the flexibility of factor graphs, improvements like more sophisticated motion models or prior knowledge can be easily integrated into the current algorithm. The integration of additional sensors, like cameras with own detection pipelines, could also be a direction of future research. We will continue this work in the future, to exploit the capabilities of our multimodal measurement model for the even more difficult problem of extended object tracking. The ability of factor graphs to store the complete history reveals a great potential in this field.

REFERENCES

- [1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2015.
- [2] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [3] S. Shi, X. Wang, and H. Li, "Pointcnn: 3d object proposal generation and detection from point cloud," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [4] Li Zhang, Yuan Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [5] X. Weng and K. Kitani, "A baseline for 3d multi-object tracking," *CoRR*, vol. abs/1907.03961, 2019.
- [6] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proc. of Intl. Conf. on Computer Vision (ICCV)*, 2015.
- [7] S. Sharma, J. A. Ansari, J. Krishna Murthy, and K. Madhava Krishna, "Beyond pixels: Leveraging geometry and

- shape cues for online multi-object tracking," in *Proc. of Intl. Conf. on Robotics and Automation (ICRA)*, 2018.
- [8] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. of Intl. Conf. on Image Processing (ICIP)*, 2017.
- [9] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," in *Proc. of Robotics: Science and Systems (RSS)*, Sydney, Australia, 2012.
- [10] F. Dellaert, M. Kaess, *et al.*, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, 2017.
- [11] T. Pfeifer and P. Protzel, "Expectation-maximization for adaptive mixture models in graph optimization," in *Proc. of Intl. Conf. on Robotics and Automation (ICRA)*, 2019.
- [12] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [13] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3d object detection leveraging accurate proposals and shape reconstruction," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [14] B. Xu and Z. Chen, "Multi-level fusion based 3d object detection from monocular images," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [15] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *Proc. of Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.
- [16] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [17] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [18] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [19] W. Choi, C. Pantofaru, and S. Savarese, "A general framework for tracking multiple people from a moving camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, 2013.
- [20] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, 2011.
- [21] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proc. of Intl. Conf. on Computer Vision (ICCV)*, 2009.

- [22] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *Proc. of Intl. Conf. on Computer Vision (ICCV)*, 2019.
- [23] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [24] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, "Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering," in *Proc. of Intelligent Vehicles Symposium (IV)*, 2018.
- [25] M. Schiegg, P. Hanslovsky, C. Haubold, U. Koethe, L. Hufnagel, and F. A. Hamprecht, "Graphical model for joint segmentation and tracking of multiple dividing cells," *Bioinformatics*, vol. 31, no. 6, 2014.
- [26] H. Wang, J. Sun, S. Lu, and S. Wei, "Factor graph aided multiple hypothesis tracking," *Science China Information Sciences*, vol. 56, 2013.
- [27] F. Meyer, P. Braca, P. Willett, and F. Hlawatsch, "A scalable algorithm for tracking an unknown number of targets using multiple sensors," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, 2017.
- [28] F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, and F. Hlawatsch, "Distributed localization and tracking of mobile networks including noncooperative objects," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 1, 2016.
- [29] F. Dellaert and Others, "GTSAM," <http://research.cc.gatech.edu/borg/gtsam>.
- [30] S. Agarwal, K. Mierle, and Others, "Ceres Solver," <http://ceres-solver.org>.
- [31] H. Karunasekera, H. Wang, and H. Zhang, "Multiple object tracking with attention to appearance, structure, motion and size," *IEEE Access*, vol. 7, 2019.
- [32] J. Luiten, T. Fischer, and B. Leibe, "Track to reconstruct and reconstruct to track," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 2020.
- [33] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Krahenbuhl, T. Darrell, and F. Yu, "Joint monocular 3d vehicle detection and tracking," in *Proc. of Intl. Conf. on Computer Vision (ICCV)*, 2019.
- [34] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proc. of Intl. Conf. on Computer Vision (ICCV)*, 2015.
- [35] K. Burnett, S. Samavi, S. Waslander, T. Barfoot, and A. Schoellig, "autotrack: A lightweight object detection and tracking system for the sae autodrive challenge," in *Proc. of Intl. Conf. on Computer and Robot Vision (CRV)*, 2019.
- [36] W. Tian, M. Lauer, and L. Chen, "Online multi-object tracking using joint domain information in traffic scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, 2020.
- [37] T. Pfeifer and Others, "libRSF," <https://github.com/TUC-ProAut/libRSF>.
- [38] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [39] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, 2008.