Generative Object Detection and Tracking in 3D Range Data

Ralf Kaestner, Jérôme Maye, Yves Pilat, and Roland Siegwart
Autonomous Systems Lab, ETH Zurich, Switzerland
email: {ralf.kaestner, jerome.maye, roland.siegwart}@mavt.ethz.ch, ypilat@student.ethz.ch

Abstract— **This paper presents a novel approach to tracking dynamic objects in 3D range data. Its key contribution lies in the generative object detection algorithm which allows the tracker to robustly extract objects of varying sizes and shapes from the observations. In contrast to tracking methods using discriminative detectors, we are thus able to generalize over a wide range of object classes matching our assumptions. Whilst the generative model underlying our framework inherently scales with the complexity and the noise characteristics of the environment, all parameters involved in the detection process obey a clean probabilistic interpretation. Nevertheless, our unsupervised object detection and tracking algorithm achieves real-time performance, even in highly dynamic scenarios covering a significant amount of moving objects. Through an application to populated urban settings, we are able to show that the tracking performance of the presented approach yields results which are comparable to state-of-the-art discriminative methods.**

I. Introduction

Understanding dynamic properties of the world has become an increasingly popular research topic in mobile robotics. The motivations for this popularity are manifold. They comprise unwanted corruption of the localization and map building process by non-stationary features or the endeavor to navigate platforms in highly dynamic settings.

A widely popular group of methods addressing the dynamic state estimation problem is committed to object tracking. Through object tracking, we seek to endow our robots with an understanding of the motion patterns displayed by the various types of objects in their environment. Whereas all these objects seem to compete for mobility, our platforms shall be able to safely interact with them or to confidently move by their sides.

In this paper, we therefore address the challenge of a generalized object detection and tracking framework. Our key contribution lies in the generative object detection algorithm which allows the tracker to robustly extract objects of varying sizes and shapes from the observations. In contrast to tracking methods using discriminative detectors, we are thus able to reason over a wide range of object classes. Our unsupervised approach is based on a 3-dimensional representation of the world as those acquired by state-of-the-art laser range sensors. However, the method does not put any general constraint on the origin and dimensionality of the input data.

The remainder of this paper is structured as follows: First, Section II will give a brief overview over related approaches and the work that majorly influenced the proposed method. With a strong emphasis resting on the probabilistic backgrounds, we will then introduce our generative object model in Section III. Section IV shall be dedicated to the probabilistic exploitation of the generative model, leading up to our object detection algorithm. In Section V, we furthermore want to discuss the tracking approach in brevity. The conception of the experiments and the evaluation of the results in Section VI will then conclude the paper.

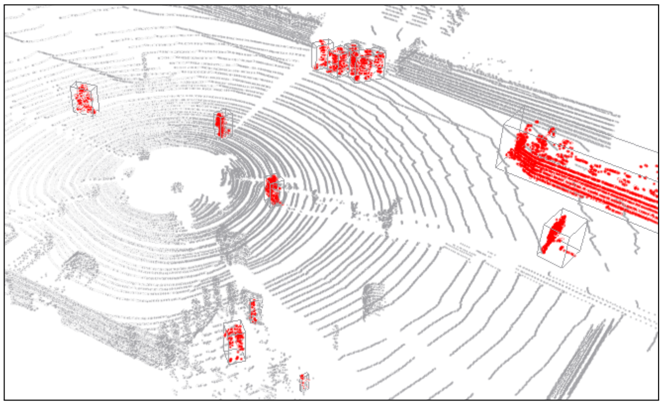


Fig. 1. A bird's-eye illustration showing the detection results of the proposed generative approach on the ETH Zurich *Tannenstrasse* dataset. In this populated urban setting, we found various types of moving objects, such as groups of pedestrians, cyclists, cars, and trams. Detected object points are surrounded by oriented bounding boxes and colored in red.

II. Related Work

The problem of detecting and tracking dynamic objects in 3D has been addressed using a variety of sensor setups and algorithms. Thereby, vision-based methods are widely dominant. The benefit of color features still prevails over the high noise characteristic of stereo vision approaches.

A purely vision-based effort which highly relates to the application scenario addressed here has recently been presented in [1]. In this work, the authors propose a technique to detect and track pedestrians and cars in populated urban scenes. A major claim of the paper indicates that the tracker should select specific motion parameters for different types of objects. Unfortunately, this procedure renders object classification an inevitable key problem, thus constraining the method in terms of its scalability.

To overcome the lack of precision of stereo vision trackers, a second class of approaches focuses on combined setups involving monocular cameras and range measurement devices. In [2], for instance, such an approach has been applied to the pedestrian detection problem. It utilizes supervised learning techniques and thus suffers from the aforementioned difficulties of discriminative detectors. Additionally, such hybrid methods inherently introduce the parameter calibration problem into the tracking framework.

A third class of detection and tracking approaches exclusively relies on range sensor data. Whereas benefiting from high accuracy, these methods widely lack sufficiently distinct features to alleviate the detection process. Thus, some authors propose to use 2D leg signatures in order to infer about the position of human objects [3]. Alternatively, [4] suggests to cede the feature selection problem in pedestrian detection problems to a boosted cascade of low-level classifiers. Just recently, an extension of this method to 3D range data has been presented in [5]. Plenty of similar methods exist that utilize different kinds of features and classifiers, all being designed to detect a limited number of specific objects [6], [7]. By approaching objection detection in a discriminative framework, robust detection results thus come at the cost of scalability towards additional object classes.

In contrast, very little work exists in the field of generative object detection for dynamic object tracking. An approach exploiting semantic information about groups of people has for instance been proposed in [8]. It uses 2D range readings from a stationary sensor to track complex human motion in a room. Aiming at the detection of cars [9], an alternative model-based technique uses geometric and dynamic features to achieve robustness. Whereas both these methods do not generalize to arbitrary object models, [10] have recently developed a class-independent approach that extracts objects using a local convexity criterion on a 2D projection of 3D range scans. Here, the tracking stage is composed of an involved combination of feature matching, ICP, and state filtering techniques.

III. Generative Object Model

We will first state a generative model that shall later be exploited in order to derive our object detection algorithm. As the formalism suggests, the model should describe how observable data can be generated given the hidden parameters of a system. A common modeling approach we want to adapt is to express the joint probability over observations, states, and labels as a factorization of conditional probabilities.

Our object detector operates on range readings acquired with a laser range scanner that continuously spins around its yaw axis. Throughout this paper, we therefore define an observation $\mathbf{z}$ as the random tuple $(\mathbf{r}, \vartheta, \varphi)$, with $\mathbf{r}$ being the measured range, and ϑ and φ are the pitch and yaw angles of the laser beam inducing $\mathbf{r}$. We assume that $\mathbf{r}$ is affected by Gaussian noise, but do not account for uncertainty in the acquisition angles ϑ and φ of our rotating laser sensor.

In a tracking scenario, the focus of interest rests on the dynamic aspects of the world. In fact, we want to limit the number of tracking hypothesis by disambiguation between static and dynamic objects. To make this explicit in our model, we therefore introduce a binary state variable $\mathbf{x}$, which is true if an observation $\mathbf{z}$ originates from a dynamic object and false otherwise.

In general, a simple range sensor cannot perceive entire objects. It rather draws regular-spaced random points from a surface distribution defining an object's boundary. However, we want to infer the origin of these points and therefore assign an object label $l \in [1, L]$ to each observation $\mathbf{z}$. With the "true" correspondence between measurements and objects being unknown, an L-dimensional random vector $\mathbf{o}$ shall be used to represent a 1-of-L choice of the label. Hence, $o_l \in \{0, 1\}$ such that $\sum_{l=1}^{L} o_l = 1$, and $p(\mathbf{o})$ is a categorical distribution.

Given a probabilistic representation of the observations, states, and object labels, we are now ready to state the joint distribution over these variables. Specifically, we define

$$p(\mathbf{z}, \mathbf{x}, \mathbf{o}) \quad = \quad p(\mathbf{x})\; p(\mathbf{o})\; p(\mathbf{z} \mid \mathbf{x}, \mathbf{o}). \tag{1}$$

According to the above factorization, our generative process induces observations in the following way: First, an object is randomly selected by sampling o_l from the categorical distribution over object labels $p(\mathbf{o})$. Independently, a state x_i is drawn from the binomial distribution $p(\mathbf{x})$ representing the probability of perceiving a dynamic object. The generation of a corresponding observation z_i is then governed by the conditional density $p(\mathbf{z} \mid \mathbf{x}, \mathbf{o})$ which inherently defines a classification of observations with respect to their object of origin and state. Thus, it shall henceforth be referred to as *classification likelihood*.

IV. Object Detection

As illustrated above, the observation process in an object tracking scenario can be interpreted as a generative process in which range measurements are produced by dynamic objects. The task of the probabilistic object detector hence boils down to estimating the unobserved process variables. We will therefore follow the generative approach by introducing model parameters and then fitting these parameters to maximize the likelihood of the observations.

To obtain a tractable solution to the parameter estimation problem, further decomposition of the likelihood term will prove beneficial. We may therefore use Bayes' theorem to rewrite the classification likelihood as

$$\begin{aligned} p(\mathbf{z} \mid \mathbf{x}, \mathbf{o}) \quad &\propto \quad p(\mathbf{x} \mid \mathbf{z}, \mathbf{o})\; p(\mathbf{z} \mid \mathbf{o}) \\ &\propto \quad p(\mathbf{x} \mid \mathbf{z})\; p(\mathbf{z} \mid \mathbf{o}). \end{aligned} \tag{2}$$

Here, we explicitly assume that the probability of observing a static or dynamic reading does not depend on the object of origin. Put differently, we consider our observations to sufficiently discriminate between the binary state hypotheses, and thus $p(\mathbf{x} \mid \mathbf{z}, \mathbf{o}) = p(\mathbf{x} \mid \mathbf{z})$.

The above factorization essentially leaves us with two conditional probabilities we will henceforth consider independently. From the right-hand side of (2), we first note that $p(\mathbf{x} \mid \mathbf{z})$ states an assignment of states $\mathbf{x}$ to observations $\mathbf{z}$. The conditional shall therefore be coined as *state model*. The second density implies a segmentation of observations according to the object labels an will be termed the *clustering model*.

A. State Estimation

The first unobserved quantity we want to infer are the state variables $\mathbf{x}$. We will therefore adapt a recently presented method which utilizes *Gaussian mixture models* in order to learn 3D representations of dynamic environments [11]. In this approach, the continuous polar space around the sensor is discretized into $N \times M$ evenly spaced *range image* cells. Each cell contains a mixture of Gaussians, and the marginal $p(\mathbf{r})$ of such a cell mixture constitutes a distribution over the measured range $\mathbf{r}$.

By incorporating the mixture model, we may now extend our formalism to express how the state variables $\mathbf{x}$ can be inferred. Therefore, we introduce a 1-of-K assignment variable $\mathbf{g}$ and a set of latent model parameters $\Theta_{\mathbf{x}} = \{K, \mathbf{w}, \mu, \sigma\}$, where K is the overall number of mixture components, $\mathbf{w}$ denotes the weight vector with components $w_k = p(g_k = 1)$, and $N(\mu_k, \sigma_k)$ represents the k-th distribution in the mixture. For a detailed discussion of the theory behind Gaussian mixture models, the interested reader may refer to [12].

Following the approach in [11], we may estimate the parameters $\Theta_{\mathbf{x}}$ of the model that maximize the likelihood of the observed measurements in an online framework. We are thus able to approximate the marginal mixture density

$$p(\mathbf{z} \mid \Theta_{\mathbf{x}}) \quad = \quad \sum_{k=1}^{K} p(g_k = 1 \mid \Theta_{\mathbf{x}}) \; p(\mathbf{z} \mid g_k = 1, \Theta_{\mathbf{x}}), \quad (3)$$

where once again $\mathbf{z}$ are the observations composed of noisy range readings $\mathbf{r}$, and the acquisition angles ϑ and φ are mapped into the discrete coordinate vector of the range image cell generating $\mathbf{r}$.

The assignment variable $\mathbf{g}$ forms a K-dimensional binary random vector, where the component g_k is responsible for selecting the k-th Gaussian in the mixture according to the categorical probability w_k. Furthermore, the k-th Gaussian covers the expectation μ_k of a range reading along with its measurement noise σ_k. If both these parameters are considered to constitute a function of the sensing process and the environment, we can think of each Gaussian as a probabilistic representation of an object patch. Over time, such an object patch may repeatedly be sampled by our range sensor, and the associated Gaussian will be updated accordingly. Intuitively, each of the weight values w_k may hence be interpreted as the prior probability of occurrence of that specific object patch.

At this point, the reader should briefly recall that we effectively envision to estimate the state variables $\mathbf{x}$ from the observations $\mathbf{z}$. Following the generative paradigm, we once again use Bayes' rule to inversely relate these quantities and conclude that

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{z}, \Theta_{\mathbf{x}}) \quad &\propto \quad p(\mathbf{z} \mid \mathbf{x}, \Theta_{\mathbf{x}}) \; p(\mathbf{x} \mid \Theta_{\mathbf{x}}) \qquad (4) \\ &\propto \quad p(\mathbf{z} \mid \Theta_{\mathbf{x}}) \; p(\mathbf{x} \mid \Theta_{\mathbf{x}}). \end{aligned}$$

Here, we additionally assume that knowing our model parameters $\Theta_{\mathbf{x}}$, the probability of observing $\mathbf{z}$ is equal for both dynamic and static objects. Note that this assumption is inherently based on the correspondence information covered by the mixture model, and the background details are formally explained in [11].

To maximize $p(\mathbf{x} \mid \mathbf{z}, \Theta_{\mathbf{x}})$, we propose to adopt the method described in [13]. It intuitively bases on the assumption that the majority of observations originate from static objects. Hence, knowing the correspondences between the Gaussians and their induced measurements, the weight parameters $\mathbf{w}$ practically yield a good approximation to $p(x = 0 \mid \mathbf{z}, \Theta_{\mathbf{x}})$.

B. Clustering

In the previous section, we have derived a formalism to estimate the state $\mathbf{x}$ associated with each observation $\mathbf{z}$. We have shown how the state model can be interpreted in order to probabilistically infer whether measurements originate from static or dynamic objects. By computing the maximum likelihood state estimates from this model, we may thus obtain a binary segmentation of static and dynamic observations.

In the following paragraphs, we will now discuss a generative interpretation of the clustering model used in this object detection approach. Generally speaking, such clustering models exploit the strong assumption that the data has been generated by a mixture of component probability distributions, where each component belongs to a different cluster [14]. Note that in our application, the components correspond to the objects $\mathbf{o}$, and the data simply is the sequence of range sensor observations $\mathbf{z}$. Then, model-based clustering is an approach to estimating an approximate likelihood of the data [15]. The likelihood of a point to belong to a specific cluster is thereby measured by a distance metric. And although such metrics are widely model-dependent, the affinity of points and clusters is commonly determined by spatial proximity.

In compliance with the generative paradigm, we borrow the probabilistic formalism from [14] and introduce an additional set of latent model parameters $\Theta_{\mathbf{o}} = \{L, \theta_1, ..., \theta_L\}$. As above, L refers to the number of mixture components or objects, and the θ_l are the parameters of the l-th cluster.

According to [14], we may thus express our clustering model by a factorization into independent assignments of observations to objects. And hence, we state that

$$p(\mathbf{z} \mid \mathbf{o}, \Theta_{\mathbf{o}}) \quad = \quad \prod_{i} p(z_i \mid \theta_{l_i}), \qquad (5)$$

where $l_i = l$, if the observed sample z_i has been generated by the l-th object.

In order to infer a labeling of our 3D range observations with respect to their objects of origin, we specifically suggest to adapt an efficient agglomerative clustering model which, according to the authors, can best be described as *radially bounded nearest neighbor* graph [16]. As inputs, we use the observations which have been segmented as dynamic.

Note that this corresponds to taking the maximum likelihood estimates of the state model for $x = 1$ and plugging them into the classification likelihood stated in (2).

The agglomerative clustering algorithm then labels observations as follows: It iterates over all unassigned data points whereas assigning them to an already existing cluster if a point in this cluster lies within a predefined distance d from the candidate point. If several points are located within the search radius and the labels of these points disagree, the affected clusters will be merged.

Even though the adopted method does not necessarily build a graph structure, in a model-based setting it would best be characterized as *single-link clustering* [17]. We justify the analogy by examination of the distance metric brought to bear: In all cases considered by the algorithm, a relabeling occurs if the minimum distance between points residing in different (or no) clusters evaluates to d or less. Probabilistically speaking, we may hence assume that a new observation z_i is generated by an object l according to a multivariate isotropic Gaussian with variance $\sigma_d \, I$ centered at the location of any observation $z_{m_l(i)}$ with the same object of origin. Here, we have used σ_d to express that the variance parameter of the Gaussian implicitly is a function of the search distance d. Furthermore, $m_l(i)$ is a mapping of the i-th observation z_i to that particular observation assigned to object l which was responsible for generating z_i.

Through the above arguments, we have illustrated how hierarchical clustering algorithms can be interpreted as purely probabilistic methods. In our approach, however, we want to stress the generative concept behind the single-link clustering model even further. The motivations for this extension are two-fold: On one hand, our rotating range sensor is characterized by a fixed angular resolution. As a consequence thereof, observations induced by distant objects suffer from poor sampling, whereas close objects cause comparably dense responses. The second justification is a direct implication of the sensor model which typically suggests increased measurement noise for distant range readings.

To reflect these insights, we propose to make the variance parameter of the point-generating Gaussians dependent on the distance between the cluster center and the sensor origin. Accordingly, we define that

$$p(t_c(z_i) \mid \theta_{l_i}) \quad = \quad \mathcal{N}(t_c(z_{m_{l_i}(i)}), \sigma_{l_i} \, I), \qquad (6)$$

where $t_c(z_i)$ denotes a Cartesian-space transform of the polar-space observation z_i, and σ_{l_i} corresponds to the suggested variance parameter of candidate cluster l_i.

We want to conclude the discussion of our clustering approach with the following insight: In order to find the maximum likelihood clustering, it will be perfectly sufficient to evaluate the Euclidean distances $\|t_c(z_i), t_c(z_{m_{l_i}(i)})\|$ between the observed points. A sketch of the according proof has been provided in [14].

C. Online Algorithm

Thus far, we have delivered a formal introduction to the probabilistic groundings of our generative object detection approach. This section will be dedicated to the algorithmic details, leading up to a brief discussion about computational complexity.

The proposed object detection method is formally stated in Alg. 1. At each iteration, it takes a set of range measurements z_i and the parameters of both the state model $\Theta_{\mathbf{x}}$ and the clustering model $\Theta_{\mathbf{o}}$ as input arguments. In return, the procedure outputs the sought correspondence mapping $c : z_i \mapsto l_i$ between the input observations and the detected objects. Additionally, the provided state model parameters will be modified with respect to the given input observations and made available for the next iteration of our object detector.

```
Algorithm 1: detectObjects($\mathbf{Z}_{1:N}$, $\Theta_{\mathbf{x}}$, $\Theta_{\mathbf{o}}$)
  Input: Set of 3D range observations $\mathbf{Z}_{1:N} = \{z_1, ..., z_N\}$
  Input: State model parameters $\Theta_{\mathbf{x}}$
  Input: Clustering model parameters $\Theta_{\mathbf{o}}$
  Output: Object mapping $c : z_i \mapsto l_i$
  Output: Updated state model parameters $\Theta'_{\mathbf{x}}$

  // State estimation
  $\mathbf{Z}_{dyn} \leftarrow \emptyset$
  foreach $z_i \in \mathbf{Z}_{1:N}$ do
      $\Theta'_{\mathbf{x}} \leftarrow$ updateStateModel($\Theta_{\mathbf{x}}$, $z_i$)
      Evaluate $p(x_i = 1 \mid z_i, \Theta'_{\mathbf{x}})$ using (4) and [13]
      if $p(x_i = 1 \mid z_i, \Theta'_{\mathbf{x}}) \geq p_{dyn}$ then
          $\mathbf{Z}_{dyn} \leftarrow \mathbf{Z}_{dyn} \cup \{z_i\}$
      end
  end
  // Clustering
  foreach $z_i \in \mathbf{Z}_{dyn}$ do
      $c(z_i) \leftarrow l_i$                    // Initialize labels
  end
  while $\mathbf{Z}_{dyn} \neq \emptyset$ do
      foreach $z_i \in \mathbf{Z}_{dyn}$ do
          $\mathbf{Z}_{dyn} \leftarrow \mathbf{Z}_{dyn} \setminus \{z_i\}$
          $z_j \leftarrow$ nearestNeighbor($\mathbf{Z}_{dyn}$, $z_i$)
          if $c(z_i) \neq c(z_j)$ and $\|t_c(z_i), t_c(z_j)\| \leq \sigma_{c(z_j)}$ then
              // Merge clusters
              $c(z_i) \leftarrow c(z_j)$
              $\mathbf{Z}_{dyn} \leftarrow \mathbf{Z}_{dyn} \cup \{z_i\}$
          end
      end
  end
```

Our algorithm makes frequent use of two auxiliary functions. Before evaluating the state x_i associated with an input reading z_i, `updateStateModel` reestimates the Gaussian mixture parameters by updating the underlying world representation. The modified model $\Theta'_{\mathbf{x}}$ then serves as an input to the state estimation step, where the z_i that have been classified as being caused by a dynamic object are accumulated in the set $\mathbf{Z}_{dyn}$ of dynamic observations. Note that the pseudo-parameter p_{dyn} acts as a probability threshold in that classification. It determines the minimal significance of a dynamic state hypothesis for an observation to be considered in the clustering step.

As the name suggests, the second helper function `nearestNeighbor` delivers the observation $z_j \in \mathbf{Z}_{dyn}$ with the smallest Euclidean distance to z_i. Therefore, it implicitly performs a Cartesian-space transform of the inputs.

D. Computational Complexity

We briefly want to examine the expected computational costs of the presented object detection algorithm.

The state estimation step of our implementation is dominated by the state model updates, and the costs of the updates are discussed in [11]. Following this discussion, we may roughly assess the update complexity by taking the average number of mixture components $\hat{K}$ into account. Obviously, $\hat{K}$ is dependent on the model resolution and can be thought of as a function of the range image discretization and the expected measurement noise. With N being the number of range observations in $\mathbf{Z}_{1:N}$, the updates hence take at most $O(N\hat{K}^2 log\hat{K})$. Note that, under practical considerations, this renders the procedure real-time.

For the clustering step, we refer to the cost analysis given by [16]. Due to the greedy nature of the agglomerative approach, the algorithm's non-optimized variant has expected complexity in $O(NlogN)$. The logarithmic costs of the nearest neighbor lookups can easily be obtained by applying efficient search structures such as the standard *kd-tree*.

In consequence, we may thus conclude that our generative object detection method can safely be assumed to achieve real-time performance in the envisioned online tracking framework.

V. Object Tracking

The generative object detector presented above produces dynamic object hypotheses from a set of range observations z_i acquired during a full 360° turn of our 3D laser sensor. These hypotheses are implicitly stated in the cluster mapping $c : z_i \mapsto l_i$ delivered by Alg. 1. We may thus further exploit the induced 3D point locations and their associated cluster assignments to infer a variety of geometric object parameters. Note that approaches to obtaining the centroid, the principal orientations, or the size of a point mass are standard techniques that will not further be explained in this paper.

To smooth the detection results by integration over time, we may then track the dynamic object hypotheses. This will not only enable us to continuously infer the full 6-DOF poses of the detected objects, but also provide estimates of their motion parameters.

With the major focus of this work being on the generative detection method, we do not intend to elaborate on a deep discussion about solutions to the object tracking problem. Instead, we will briefly summarize the procedure and explain the involved state quantities.

For tracking dynamic objects, we employ a standard multi-hypothesis approach [18] which essentially bases on the linear Kalman state estimation filter [19]. In brevity, the selected tracking algorithm has been proven to robustly associate between object hypotheses over time. It applies probabilistic reasoning techniques to maintain a multitude of statistically relevant tracks. Moreover, the method consistently deals with missing hypothesis updates, e.g. in the presence of occlusions. To further learn about the theoretic fundamentals of the matter, we suggest to the interested reader to consult the bibliographic literature.

One key insight distinguishing our generative formalism from discriminative techniques requires particular consideration: Since we do not make any assumptions about the shape or class of objects we seek to detect and track, the definition of an object's origin is whether obvious nor does it inherently result from the detection algorithm. This especially becomes evident in cases where objects undergo a substantially sparse sampling by the sensor or suffer from partial occlusion. We therefore propose to apply the following modifications to the multi-hypothesis tracking framework:

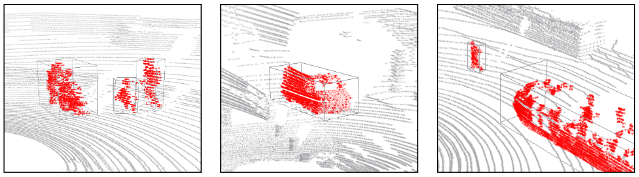
- *Bounding Box Estimation*: In addition to the 6-DOF pose variables, we incorporate bounding box parameters into the filter's state representation. Specifically, each hypothesis is extended by the size and orientation of the 3-dimensional volume enclosing all observations with coincident labels. The filters smoothing characteristics then respond to sudden and unexpected changes of an object's observed point distribution.
- *Track Splitting and Merging*: In analogy to the group tracking method presented in [8], we compute a distance measure to estimate the probability of separate hypotheses merging into or splitting away from larger hypotheses. This allows us to maintain a tractable number of state representations, even in highly dynamic environments containing significant amounts of moving objects. As a matter of fact, object grouping mainly is an implication of social behavior which can be witnessed in most robotic application scenarios.

Due to its generative nature, our modified tracking approach is able to estimate the filtered pose, size, and velocity of objects moving in 3-dimensional space. In addition, it allows for robust prediction of these quantities within in a limited time horizon.

VI. Experiments

In order to evaluate our generative approach to the object detection and tracking problem, we have conducted experiments on several outdoor datasets. These datasets were produced by a stationary Velodyne HDL 64E S2 laser range sensor in populated urban settings and contain 3D range samples acquired from various types of dynamic objects. Amongst these objects, we found single pedestrians as well as groups of pedestrians, but also cyclists, cars, and trams. Rotating at a frequency of about 5Hz, our sensor produced approximately 120,000 range observations per turn. Hence, moving objects suffer from minimal distortion which can usually be neglected in the detection process.

Our C++ implementation of the combined generative detection and tracking process runs at about 0.5Hz on a



(a) Pedestrians (b) Cars (c) Trams

Fig. 2. A selection of object detections from the ETH Zurich *Polyterrasse* and *Tannenstrasse* datasets. Again, observations corresponding to detected dynamic objects are enclosed in oriented bounding boxes and colored red.

standard PC, including visualization of the 3D scene and the resulting outputs.

In preparation of the experiments, we did not have to perform any hand-labeling of training data. All models involved were instead initialized using prior parametric knowledge about the experimental setting. In other words, the model parameters were adapted to reflect our expectations about the measurement process and the occurrence of dynamic objects in the environment. Moreover, to assess the true potential of our method, we did not allow for a learning phase of the state model. As the reader may recall from the previous discussions, the state model represents the world using maximum likelihood estimates over Gaussian mixture models. It may thus be viewed as a map containing all previously observed locations of static or dynamic occurrences.

In a first set of experiment, we qualitatively investigated the robustness of our generative detector by visual inspection of the resulting bounding box estimates. We found that, after a short initialization phase of the state model, the approach was able to robustly discriminate static and dynamic observations and to cluster disconnected objects accordingly. For regions that were previously occluded due to the presence of moving objects, we would typically witness increased amounts of false positive estimates. Once such regions undergo an adequate sampling by our sensor, the false alarms will safely abate. In areas of high static measurement noise, such as those containing vegetation, we furthermore found widely scattered occurrences of misclassified readings. However, as these isolated observations will end up in their own singular object clusters, they may easily be discarded during the detection process. Note that in order to compute a valid oriented bounding box, three observations or more are required.

Fig. 2 shows an exemplary choice of object detections from the ETH Zurich *Polyterrasse* and *Tannenstrasse* datasets used in [5]. Additionally, a less detailed bird's-eye illustration of the detector output on the *Tannenstrasse* dataset is depicted in Fig. 1. Remarkably, our detector produced coherent boxes ranging from the size of a single pedestrian to the size of an entire tram. As intended, it thus showed the potential to seamlessly generalize over a wide range of object classes.

To further confirm the feasibility of the proposed method, we quantitatively evaluated the performance of the combined detection and tracking approach by use of the *CLEAR MOT* metric [20]. This metric accounts for three ratios that need to be determined for the life-cycle of the tracker from an available ground truth. In brevity, the ratios are:

- *FN*: The false negative rate, i.e. the fraction of missing tracks that exist according to the ground truth.
- *FP*: The false positive rate, i.e. the fraction of tracks that have mistakenly been detected but do not exist in the ground truth.
- *MME*: The percentage of wrongly performed track identity switches, i.e. the tracks that were confused with respect to the ground truth.

From these ratios, two additional measures are determined:

- *MOTP*: The average Euclidean distance between the estimated state hypotheses and the referenced ground truth positions.
- *MOTA*: The ratio between the number of correct track estimates and the number of ground truth states.

Since our approach is able to detect a variety of dynamic objects and comprehensive ground truth information for the testing datasets was not available, we instead deliver a quantitative evaluation against the *Polyterrasse* dataset. Therefore, we used the ground truth labeling provided to us by courtesy of the authors of [5]. Note that in the scope of our experiments, this manually crafted ground truth may be considered as complete. The dataset in fact does not contain ground truth annotation for any objects other than pedestrians. However, these pedestrians perform all kinds of activities, e.g., pushing a stroller or walking their bikes, which results in different shapes and sizes of their appearance. In order to cope with this variability, [5] simply negelected all non-pedestrian points during training of their detector. In our generative approach, however, we consider such "carriers" as a compound of objects with similar motion properties and hence represent them as single detections.

The obtained tracking results are depicted in Fig. 3, and the performance measures acquired using the proposed metric are listed in Table I. For the purpose of comparability, we plotted the measures against the results published in [5] for the *bottom-up* (BU) and the bottom-up top-down (BUTD) pedestrian tracker.

Detector	MOTP	MOTA	FN	FP	MME
BU	$< 0.16m$	23.1%	18.7%	57.7%	n/a
BUTD	$< 0.16m$	89.1%	2.6%	7.6%	n/a
Generative	$< 0.14m$	77.7%	8.5%	10.1%	3.6%

TABLE I

Comparison of CLEAR MOT measures: BU and BUTD vs. Generative

An investigation of the measures instantly reveals that our generative approach clearly supersedes the single-class bottom-up detector for pedestrians. In addition, it fairly competes against the more advanced bottom-up top-down detector which uses an involved framework of classifiers and voting schemes. In terms of the geometric precision, the proposed method slightly outperforms both approaches. Since discriminative models can generally be expected to constitute are more accurate representation of the sought

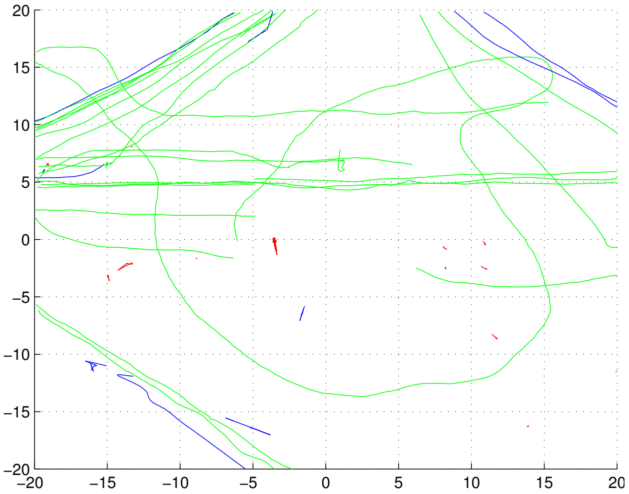


Fig. 3. Our tracking results matched against the ground truth for the ETH Zurich *Polyterrasse* dataset. Correctly estimated tracks are colored in green, and false positives show up in red. Blue tracks are non-false positive tracks which were not contained in the ground truth.

posteriors, the obtained results should be considered as being comparable to state-of-the-art discriminative methods. In contrast, the strength of the generative paradigm typically lies in the flexibility to generalize in complex learning tasks.

VII. Conclusion

In this paper, we have proposed a novel approach to tracking dynamic objects in 3D range data. Its generative object detection algorithm allows the tracker to robustly extract objects of varying sizes and shapes from the observations. Through experimental evaluation in populated urban settings, we have been able to show that the performance of the presented approach yields results which are comparable to state-of-the-art discriminative methods. Nevertheless, our detector generalizes over a wide range of object classes.

In the future, we envision ample opportunity for applications utilizing and refining our approach. A particular idea we are planning to investigate into is the bottom-up estimation of different object classes from the motion states recovered by our tracker. We would thus be able to further exploit our generative framework towards a deeper understanding of dynamic environments.

Acknowledgments

The authors would like to express their gratitude to Francis Colas for all the helpful discussions about probability theory and modeling approaches. This work has partly been supported by the EC under EUROPA (FP7-231888) and Natural Human-Robot Cooperation in Dynamic Environments (ICT-247870).

References

[1] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *The International Journal of Robotics Research*, 2010.

[2] L. Spinello, R. Triebel, and R. Siegwart, "Multimodal people detection and tracking in crowded scenes," in *Proc. of The AAAI Conference on Artificial Intelligence (Physically Grounded AI Track)*, 2008.

[3] K. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities," in *Proc. of The IEEE International Conference on Robotics and Automation (ICRA), Pasadena, CA, USA*, 2008.

[4] K. Arras, O. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Proc. of The IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2007, pp. 3402–3407.

[5] L. Spinello, M. Luber, and K. O. Arras, "Tracking people in 3d using a bottom-up top-down people detector." in *Proc. of The International Conference in Robotics and Automation (ICRA)*, 2011.

[6] L. Navarro-Serment, C. Mertz, and M. Hebert, "Pedestrian detection and tracking using three-dimensional ladar data," *The International Journal of Robotics Research (IJRR)*, vol. 29, no. 12, p. 1516, 2010.

[7] C. Premebida, O. Ludwig, and U. Nunes, "Exploiting lidar-based features on pedestrian detection in urban scenarios," in *Proc. of The 12th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2009, pp. 1–6.

[8] B. Lau, K. Arras, and W. Burgard, "Multi-model hypothesis group tracking and group size estimation," *International Journal of Social Robotics (SORO)*, vol. 2, no. 1, pp. 19–30, 2010.

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

[10] F. Moosmann and T. Fraichard, "Motion estimation from range images in dynamic outdoor scenes," in *Proc. of The IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 142–147.

[11] R. Kaestner, N. Engelhard, R. Triebel, and R. Siegwart, "A bayesian approach to learning 3d representations of dynamic environments," in *Proc. of The 12th International Symposium on Experimental Robotics (ISER)*, 2010.

[12] C. Bishop *et al.*, *Pattern Recognition and Machine Learning*, M. Jordan, J. Kleinberg, and S. B., Eds. Springer New York:, 2006.

[13] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[14] S. D. Kamvar, D. Klein, and C. D. Manning, "Interpreting and extending classical agglomerative clustering algorithms using a model-based approach," in *Proc. of The 19th International Conference on Machine Learning (ICML)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 283–290.

[15] G. Celeux and G. Govaert, "Comparison of the mixture and the classification maximum likelihood in cluster analysis," *Journal of Statistical Computation and Simulation*, vol. 47, no. 3, pp. 127–146, 1993.

[16] K. Klasing, D. Wollherr, and M. Buss, "A clustering method for efficient segmentation of 3d laser data," in *Proc. of The IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 4043–4048.

[17] J. Hartigan, *Clustering Algorithms*. John Wiley & Sons, Inc. New York, 1975.

[18] D. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 843–854, 1979.

[19] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[20] B. Keni and S. Rainer, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, 2008.