Particle-based Dynamic Semantic Occupancy Mapping using Bayesian Generalized Kernel Inference

Felix Neumann[1], Frederik Deroo[1], Georg von Wichert[1], and Darius Burschka[2]

***Abstract*— A representative and accurate environment model is essential for the safe navigation and operation of intelligent transportation systems, such as autonomous vehicles and mobile robots. This paper presents a semantic occupancy grid mapping approach that uses a particle-based map representation to approximate continuous dynamic environments. The proposed approach recursively updates occupancy, velocity and semantic class estimates using the Bayesian Generalized Kernel Inference (BGKI) framework to maintain a local occupancy map in real time. The novelty of this approach lies in its combination of the continuous static semantic mapping capabilities of BGKI with the recursive dynamic state estimation of Dynamic Occupancy Grid Maps (DOGMs) in the 3D domain. We demonstrate that the approach maintains the semantic mapping capabilities of BGKI while providing more accurate velocity estimates than previous particle-based three dimensional DOGMs on real and simulated automotive datasets, including Semantic KITTI. We show that our approach outperforms the current state of the art on both semantic mapping and velocity estimation.**

I. Introduction

Intelligent transportation systems such as autonomous vehicles or mobile robots often operate in dynamic environments and must be able to perceive and navigate in them. An accurate, uncertainty aware model of the surrounding world is needed to achieve safe operation. Occupancy Grid Maps (OGMs) are a popular map representation to address these tasks, because they explicitly model free, occupied and unknown space, and can represent arbitrarily shaped objects.

Recently, two-dimensional (2D) Dynamic Occupancy Grid Maps (DOGMs) [1], [2] have been extended to the three-dimensional (3D) domain [3], [4], [5]. One prevalent feature of these approaches is the use of particles to estimate velocity. Moreover, Chen *et al.* [3] propose to use particles as an approximation of a continuous dynamic occupancy map instead of discrete grids.

With advances in deep learning, the integration of richer information, such as semantic classes, into 3D OGMs has been pursued [6], [7], where the use of Bayesian Generalized Kernel Inference (BGKI) [8], [9] is popular due to its spatial smoothing and continuous map update properties.

However, the integration of semantic information has this far been restricted to static OGMs. Some works [10], [11] have used explicit scene flow estimation to propagate the semantic information of these maps over time, but they miss one defining feature of DOGMs, namely a recursively updated velocity estimation that is maintained in the map.

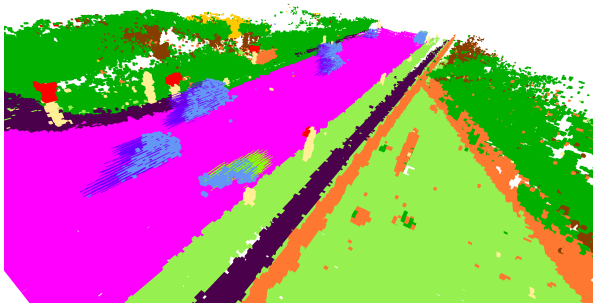


Fig. 1: Semantic dynamic occupancy estimations generated by our method on Semantic KITTI sequence 04. Velocity estimates are indicated by colored lines.

This motivates us to combine the dynamic mapping capabilities of particle-based DOGMs with the rich features of semantic OGMs. We base our work on DSP-Map [3], to produce a continuous dynamic semantic occupancy map as shown in Figure 1. We notice several limitations of DSP-Map, which we explore in Section III and propose to integrate its particle map structure with the BGKI framework to overcome them.

Our formulation generalizes semantic BGKI to dynamic scenes and can be reduced to static semantic BGKI [6] and binary occupancy mapping with BGKI [8] by imposing certain constraints, as described in Section III-B.

Our main contributions are:

- We extend DSP-Map to aggregate information from sparse LiDAR scans over longer time horizons in a continuous particle map.
- Using this adaptation of DSP-Map, we generalize the semantic BGKI framework to dynamic scenes without the use of explicit scene flow information.
- We evaluate the semantic mapping capability, as well as the velocity estimation accuracy of our approach on real and synthetic datasets.

The remainder of this work is structured as follows: Section II reviews current DOGMs and semantic mapping OGMs in the 3D domain. We briefly outline the shortcomings of the state of the art, represented by DSP-Map on sparse LiDAR data and then introduce our proposed integration and generalization of the semantic BGKI framework in Section III. Section IV presents and discusses quantitative and qualitative experimental results. Section V concludes the paper and offers potential future research directions.

This research has received funding from the Federal Ministry for Economic Affairs and Climate Action under grant agreements 19I21039A

[1]Siemens Technology, Friedrich-Ludwig-Bauer-Straße 3, 85748 Garching, Germany, {neumann.felix, frederik.deroo, georg.wichert}@siemens.com

[2]Machine Vision and Perception Group, School of Computation, Information and Technology, Technical University of Munich, Friedrich-Ludwig-Bauer-Straße 3, 85748 Garching, Germany burschka@tum.de

II. Related Work

A. Bayesian Generalized Kernel Inference occupancy maps

Occupancy mapping using BGKI was first introduced by Doherty *et al.* [8], [9] to circumvent discretizing measurements. Votes for free space or occupied space from surrounding measurements are aggregated using nonparametric bayesian inference [12] with a sparse kernel function. The map represents occupancy at a location as a beta distribution, the parameters of which represent evidence for occupied and free space. This evidence is usually gathered from range sensors that implicitly measure free space as rays cast from the sensor origin to each measurement point.

Gan *et al.* [6] proposed to extend BGKI to semantic mapping by employing Dirichlet distributions to accumulate evidence for each semantic class. Wilson *et al.* [7] extended this approach further by learning differently shaped kernels for different semantic classes.

As static semantic mapping approaches do not account for dynamic objects, Kochanov *et al.* [10] propose to use scene flow estimated by a neural network to propagate aggregated information in the map between time steps, which was extended to the BGKI framework by Unnikrishnan *et al.* [11].

While such approaches propagate information over time, they are dependent on the quality of the estimated scene flow. Since scene flow is commonly estimated using neural networks, this adds computational complexity and requires training data from the target domain. In contrast, we recursively update a velocity state including uncertainty in the map representation of BGKI without requiring scene flow.

B. Dynamic 3D occupancy mapping

Occupancy mapping in dynamic environments has been well studied for 2D DOGMs [1], [2], [13], [14], where particle representations show prevailing success in modeling the dynamic environment. However, the transfer of these methods to the 3D domain is challenging due to the higher dimensionality of the state space. Min *et al.* [4] have proposed K3DOM, which adds a layer of particles to estimate velocities in a voxel map built by binary BGKI, where dynamic objects are determined heuristically, which is extended to Dempster Shafer theory by Han *et al.* [5]. In contrast, our approach maintains a unified, continuous particle map without discretization, where each particle contains information about occupancy, velocity, and a distribution over semantic classes.

More recently, DSP-Map [3] proposed to approximate a continuous dynamic occupancy map using weighted particles without explicit voxel representation. Particles are only generated near sensor measurements and their velocity is initialized through cluster matching. A voxel map at arbitrary resolution can be derived from the aggregated weights of particles in each voxel. To facilitate updates of this map, the field of view of the sensor is partitioned in spherical space to produce the so called pyramid subspace.

Our work extends DSP-Map in the following ways: 1. We apply concepts of BGKI to the state update of particles, which resolves problems related to sparse measurements and allows us to retain information over longer periods of time. 2. We use this improved longevity of particles to integrate semantic class information into the map which allows us to focus dynamic particles on movable areas in the map.

III. Continuous Semantic Occupancy Mapping with Particles

In this section, we describe our extension of DSP-Map. Our goal is to build a vehicle centric DOGM which models the occupancy, semantic classes and velocities of the environment. Our online approach maintains a local map around the vehicle and is not used for global mapping.

In their work, Chen *et al.* [3] primarily test their approach using depth cameras to update the particle map. We observe that while the maps produced by DSP-Map using such sensors model the dynamic environment well, they lack persistence for sparse LiDAR measurements. Especially at longer distances, the sparsity of LiDAR scans leads to prematurely discarded particles when the vehicle moves. This occurs because the particle weights are reduced too much before they are confirmed by new measurements, which erases already mapped areas behind the vehicle. Figure 2 illustrates this effect and compares the results of DSP-Map and our approach on data from the automotive domain.

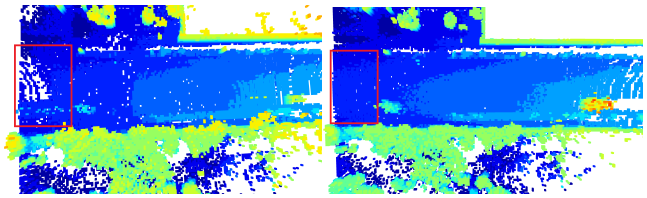
We validate this observation by analyzing the particle age of our approach and DSP-Map on the Semantic KITTI validation set, where we find that particles of our approach survive 55% longer on average. We furthermore find that for DSP-Map on average 24% of particles in the map are newly born at each iteration, while it is only 8% for our approach.

The transient nature of the particles of DSP-Map prevents the aggregation of information over longer time horizons from sparse measurements and motivates us to propose a BGKI based particle update mechanism, which builds on the map representation of DSP-Map.

A. Problem Formulation

We infer the dynamic semantic occupancy map from a sequence of measurements, where a measurement $\mathcal{X}^t := \{\mathbf{S}^t, \mathbf{C}^t\}$ at time t consists of a 3D point cloud $\mathbf{S}^t := \{\mathbf{s}_1^t, ..., \mathbf{s}_N^t\}$ with N points $\mathbf{s}_i^t \in \mathbb{R}^3$ and N associated semantic estimates over N_{cls} categories $\mathbf{C}^t := \{\mathbf{c}_1^t, ..., \mathbf{c}_N^t\}$ with $\mathbf{c}_i^t \in \mathbb{R}^{N_{\text{cls}}}$. A semantic estimate $\mathbf{c}_i^t$ is a multinomial distribution with $\sum_{k=1}^{N_{\text{cls}}} c_{i,k}^t = 1$. We assume a known transformation from world coordinates w to sensor coordinates l given by $\mathbf{T}_{w,l}^t \in SE(3)$ with rotational component $\mathbf{R}_{w,l}^t \in SO(3)$.

Our goal is to estimate a probability distribution $p(\omega^t, \mathbf{v}^t | \mathbf{x}, t, \mathcal{X}^{t_0}, ..., \mathcal{X}^t)$, for a sequence of measurements from time t_0 to t. The modeled state at a point $\mathbf{x}$ at time t is given by a semantic occupancy state ω^t and a velocity $\mathbf{v}^t$. The semantic occupancy of the map at location x is modeled by $\omega^t = \{\omega_0^t, ..., \omega_{N_{\text{cls}}}^t\}$, where ω_0^t represents free space and $\omega_1^t, ..., \omega_{N_{\text{cls}}}^t$ represent different semantic categories.



(a) DSP-Map (b) Ours

Fig. 2: Map persistence comparison between DSP-Map and our approach after 50 frames on Semantic KITTI sequence 04. Highlighted in red: sparse LiDAR measurements cause increasing gaps behind the vehicle for DSP-Map, while our approach produces a more consistent, closed surface. The vehicle is driving towards right, the map is shown in birds eye view. Colors indicate the height of a grid cell.

B. Particle Map Representation

We represent a dynamic occupancy map using a set of particles $\mathbf{P}$, where each particle $\mathbf{p}_i$ represents a semantic occupancy hypothesis with N_{cls} possible semantic classes. The particle state is parameterized by a position $\mathbf{x}_i \in \mathbb{R}^3$, a velocity $\mathbf{v}_i \in \mathbb{R}^3$ and a Dirichlet distribution $\text{Dir}(\boldsymbol{\alpha})$ defined by vector $\boldsymbol{\alpha} \in \mathbb{R}^{N_{\text{cls}}+1}$ with parameters $\alpha_i > 0$. The parameters of the Dirichlet distribution represent observed evidence for each possible class, which includes the N_{cls} semantic classes, as well as one class representing free space, which we denote by α_0. The evidence accumulated in a particle for class agnostic occupancy is the sum of the accumulated evidence for all semantic classes $\alpha_{\text{occ}} = \sum_{n=1}^{N_{\text{cls}}} \alpha_n$.

This particle-based representation allows for a flexible transfer of the map between coordinate systems, including the transfer from cartesian to spherical coordinates. The representation reduces to static semantic BGKI as introduced by Gan *et al.* [6] if the particles are sampled in a regular grid pattern with zero velocity, and to the original binary BGKI formulation [8] if only one semantic class is considered.

C. Map Prediction

At each time step Δt from t to $t+1$, we predict the state of the particle map using a linear motion model with ego motion compensation. The relative ego motion of the sensor is given by $\mathbf{T}_l^{\Delta t} = (\mathbf{T}_{w,l}^{t+1})^{-1}\mathbf{T}_{w,l}^{t}$. The map is then propagated as: $\mathbf{x}_i^{t+1} = \mathbf{T}_l^{\Delta t}(\mathbf{x}_i^t + \mathbf{v}_i^t \Delta t) + \boldsymbol{\epsilon}_x$ and $\mathbf{v}_i^{t+1} = \mathbf{R}_l^{\Delta t}\mathbf{v}_i^t + \boldsymbol{\epsilon}_v$, with additive process noise of the position $\boldsymbol{\epsilon}_x \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_x)$ and velocity $\boldsymbol{\epsilon}_v \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_v)$ of particles. Under the assumption that a particle tracks a semantically consistent object in the scene, we do not use semantic process noise, therefore the Dirichlet distribution of the particle remains unchanged between two time steps.

D. Particle Map Update

In the following we describe how the particle map is updated with a new measurement $\mathcal{X}^t$. For compactness, we will omit t in our notation from here on. Since the Dirichlet distribution is a conjugate prior to the multinomial distribution, the class estimates can be used to update the parameters of the current particles in the map in closed form. To reduce computational complexity we subsample the input point cloud using mean voxel downsampling.

We employ BGKI with the sparse kernel proposed by Doherty *et al.* [8]

$$K(d) = \begin{cases} \sigma_0 \left(\frac{2+\cos(2\pi \frac{d}{l})}{3}(1-\frac{d}{l}) + \frac{1}{2\pi}\sin(2\pi\frac{d}{l}) \right) & d \leq l \\ 0 & d > l \end{cases}$$

to update the semantic and free space evidence of particles. Here d is the euclidean distance between a map particle and a measurement point, l defines the maximum range in which measurement points contribute information to the particle, and $\sigma_0 \in \mathbb{R}$ determines the magnitude of the update. In the following we separately consider the update of the free space parameter α_0 and the semantic parameters $\alpha_1, ..., \alpha_{N_{\text{cls}}}$.

To update the semantic parameters of a particle $\mathbf{p}_i$, the semantic evidence of each measurement point is weighted with the sparse kernel function to produce the update of the distribution parameters as $\Delta\boldsymbol{\alpha}_i = \sum_{j=1}^{N} K(d_{\mathbf{p}_i,\mathbf{s}_j})\mathbf{c}_j,\ i > 0$. Since the sparse kernel is zero for most measurement points, it is sufficient to consider only those points that lie within the distance l to the particle $\mathbf{p}_i$. The free space parameter update $\Delta\alpha_0$ cannot directly be calculated from the 3D point cloud, as free space measurements are implicitly given by rays traced between the sensor origin and the points of the observed point cloud. Commonly, free space measurement points are sampled along these rays [8], [11], [6], and the nearest point to the query point, in our case a particle, is used as a measurement. As these sampling methods only approximate the true measurement ray, we follow the method proposed by Doherty *et al.* [9] and directly calculate the distance between a particle $\mathbf{p}_i$ at position $\mathbf{x}_i$ and normalized ray $\mathbf{r}_j = \frac{\mathbf{s}_j - \mathbf{o}}{||\mathbf{s}_j - \mathbf{o}||}$ cast from the sensor origin $\mathbf{o}$ to measurement point $\mathbf{s}_j$. Since the ray terminates at $\mathbf{s}_j$, the distance between a ray $\mathbf{r}_j$ and particles that are farther from the sensor than $\mathbf{s}_j$ is calculated as the euclidean distance between the particle and the point. Thus, the distance between a particle $\mathbf{p}_i$ and ray $\mathbf{r}_j$ is calculated as

$$d_{\mathbf{p}_i,\mathbf{r}_j} = \begin{cases} \mathbf{x}_i - \langle \mathbf{x}_i, \mathbf{r}_j \rangle \mathbf{r}_j & ||\mathbf{x}_i|| \leq ||\mathbf{s}_j||, \\ ||\mathbf{x}_i - \mathbf{s}_j|| & ||\mathbf{x}_i|| > ||\mathbf{s}_j||, \end{cases}$$

where $\langle \cdot, \cdot \rangle$ is the scalar product and $||\cdot||$ is the euclidean norm. Therefore, the free space measurement update for particle $\mathbf{p}_i$ is calculated as $\Delta\alpha_{0,i} = \sum_{j=1}^{N} K(d_{\mathbf{p}_i,\mathbf{r}_j})$. Unlike Doherty *et al.* [9], who use sampled points which link to the respective rays they were sampled from, we employ the pyramid structure proposed by Chen *et al.* [3]. As shown in Figure 3, this partitioning scheme can be used to limit the search space for rays and points to neighboring pyramids. We consider a sufficient number of neighboring pyramids to cover the radius of the kernel up to a minimum measurement distance $s_{\min}$ to the sensor.

LiDAR data is frequently provided as point clouds where invalid laser returns are filtered out and not as raw range images. This prevents ray casting for free space measurements in regions of the field of view without a returned laser measurement, as no measurement points are provided.

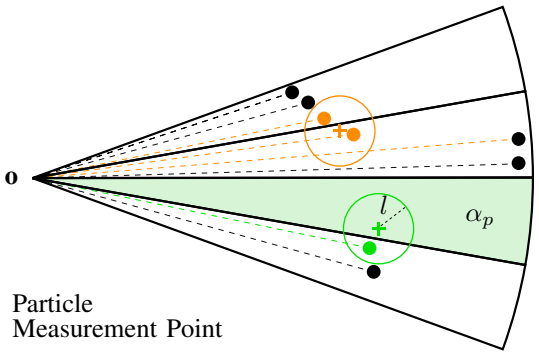


Fig. 3: Pyramid space partitioning for the particle update. A 2D slice of the 3D partition is shown. Particles are updated using measurement points and free space rays cast from sensor origin $\mathbf{o}$ that lie within the range l of the particle, as well as the free space prior α_p if no measurements are in a pyramid. The search space for these samples can be reduced to nearby pyramids.

One way to address this issue is to fill the missing measurements with rays cast to infinity, which requires exact specification for different LiDAR models. Instead, we address this issue more generally by imposing a prior free space measurement probability α_p for each empty pyramid that is in the field of view of the LiDAR. The free space parameter α_0 of a particle that lies in such a pyramid, e.g. the green particle in Figure 3 is calculated using the rays of points from neighboring pyramids and the free space prior as $\Delta\alpha_{0,i} = \sum_{j=1}^{N} K(d_{\mathbf{p}_i,\mathbf{r}_j}) + \alpha_p$.

The particle updates can be efficiently calculated on a GPU, where we parallelize the computation with a kernel that launches a block per pyramid to calculate the free space and semantic update for all particles in the pyramid.

E. Occupancy Calculation and Map Queries

We query a semantic map in two different ways once it is constructed. A finite volume may be queried, which can be used to construct a conventional voxel representation of the map. For a volume element $\mathbb{V}$, the semantic occupancy state is determined by the evidence of all particles contained in it. The occupancy state is represented by the Dirichlet distribution with parameters $\boldsymbol{\alpha}_\mathbb{V} = \frac{1}{N_\mathbb{V}} \sum_{i=1}^{N_\mathbb{V}} \boldsymbol{\alpha}_{\mathbf{p}_i}$, which is the mean Dirichlet distribution of all $N_\mathbb{V}$ particles $\mathbf{p}_i \in \mathbb{V}$. The velocity estimate for the volume element is similarly calculated, but weighted by the occupancy probability $p_{\text{occ},i} = \frac{\alpha_{\text{occ},i}}{\alpha_{\text{occ},i}+\alpha_{0,i}}$ as $\mathbf{v}_\mathbb{V} = \frac{1}{P_o} \sum_{i=1}^{N_\mathbb{V}} p_{\text{occ},i} \mathbf{v}_{\mathbf{p}_i}$ with $P_o = \sum_{i=1}^{N_\mathbb{V}} p_{\text{occ},i}$. For an occupied cell, the probability of being dynamic is calculated analogously to the Dempster Shafer approach presented in DSP-Map, but instead of the summed particle weights, the summed evidence for occupancy is considered.

The semantic map can also be queried using a point in 3D space through interpolation between nearby particles. To this end, we can also employ BGKI on the mapped particles to gather the evidence from the map in a local area. The estimated semantic occupancy at point $\mathbf{x}$ is calculated as $\boldsymbol{\alpha}_\mathbf{x} = \sum_{i=1}^{N} K(d_{\mathbf{p}_i,\mathbf{x}}) \boldsymbol{\alpha}_{\mathbf{p}_i}$. The velocity estimate of the point is similarly calculated to that of the volume element as $\mathbf{v}_\mathbf{x} = \frac{1}{P_k} \sum_{i=1}^{N} K(d_{\mathbf{p}_i,\mathbf{x}}) p_{\text{occ},i} \mathbf{v}_{\mathbf{p}_i}$ with $P_k = \sum_{i=1}^{N_\mathbb{V}} K(d_{\mathbf{p}_i,\mathbf{x}}) p_{\text{occ},i}$.

Both query methods result in a Dirichlet distribution for semantic occupancy. The variance of this distribution can provide an estimate of the uncertainty of the predicted state. However, for an occupied cell classified as an occupied category ω_i, the variance value does not distinguish between uncertainty whether the cell is occupied or not, and the uncertainty of what category the cell is occupied by. Therefore, we propose to consider two separate variances, the occupancy variance, which provides information about the uncertainty of whether a region is occupied, as well as semantic variance for occupied regions, which represents the uncertainty over which semantic class occupies the region. To calculate the occupancy variance, we consider α_occ and α_0, which form a beta distribution with variance $\mathbf{Var}[\omega_\text{occ}] = \frac{\alpha_0 \alpha_\text{occ}}{(\alpha_0+\alpha_\text{occ})^2(\alpha_0+\alpha_\text{occ}+1)}$.

The semantic variance of an occupied query is the variance of the Dirichlet distribution $\text{Dir}(\alpha_1, ..., \alpha_{N_\text{cls}})$ over the semantic parameters of the queried Dirichlet distribution, without the free space evidence α_0. Thus, the semantic variance is expressed by $\mathbf{Var}[\omega_i] = \frac{\tilde{\alpha}_i(1-\tilde{\alpha}_i)}{\alpha_\text{occ}+1}$ with $\tilde{\alpha}_i = \frac{\alpha_i}{\alpha_\text{occ}}$.

F. Particle Management

Our approach maintains a local map of particles around the vehicle. At each iteration, all particles within update range of the sensor measurements are updated. The update range is determined by the field of view of the sensor and the range of the sparse kernel function.

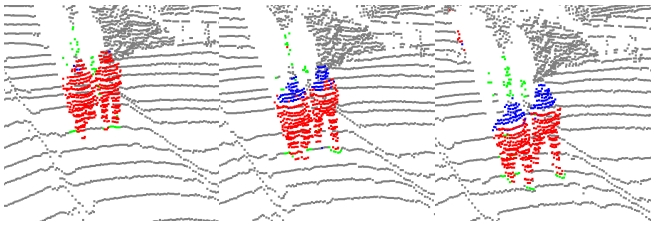
A particle representation of the map offers a high degree of flexibility in the allocation of processing to certain areas of the map. To minimize processing requirements, it is advantageous to keep the number of particles as small as possible. This is achieved by initializing static particles across the map with a uniform prior distribution of high variance $\boldsymbol{\alpha}_\text{prior}$, to represent unknown space. Each iteration, particles outside the map bounds are discarded and particles are initialized in newly developed areas of the map. Particles are sampled around measurement points to focus processing on interesting regions. We follow a similar approach to DSP-Map and initialize the velocity of particles using clustering and Hungarian matching between consecutive LiDAR scans. However, to improve the quality of matches we only consider points classified as movable classes, such as pedestrians and vehicles for clustering and matching. While this generally prevents the generation of dynamic particles for static objects, it also makes the velocity estimation prone to errors of the semantic segmentation network. Therefore, similarly to DSP-Map, we also initialize particles with randomly sampled velocities for all measurement points. Particles are initialized with the same semantic class distribution as the LiDAR scan point they originate from.

To further reduce computational load, we discard particles with low occupancy probability $\frac{\alpha_\text{occ}}{\alpha_\text{occ}+\alpha_0} < \epsilon_\text{occ}$. This means that free space is implicitly modeled by an absence of particles, which improves efficiency significantly, since usually free space comprises most of a scene. Furthermore,

TABLE I

COMPARISON OF SEMANTIC MAPPING PERFORMANCE ON THE SEMANTIC KITTI VALIDATION AND TEST SET.

Split	Method	Future Scans	Car	Bicycle	Motorcycle	Truck	Other Vehicle	Person	Bicyclist	Motorcyclist	Road	Parking	Sidewalk	Other Ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Sign	mIoU
Val.	Segmentation [15]	✓	91.0	25.0	47.1	40.7	25.5	45.2	62.9	0.0	93.8	46.5	81.9	**0.2**	85.8	54.2	84.2	52.9	72.7	53.2	40.0	52.8
	S-BKI (0.2) [7]	✓	92.6	30.3	55.3	43.1	25.0	51.9	69.9	0.0	93.6	46.8	81.9	0.1	87.9	57.5	86.0	59.8	74.0	60.0	43.2	55.7
	S-BKI (0.1) [6]	✓	93.5	33.5	57.3	44.5	27.2	52.9	72.1	0.0	**94.4**	**49.6**	**84.0**	0.0	**88.7**	**59.6**	**86.9**	**62.5**	**75.3**	**63.6**	45.1	57.4
	ConvBKI [7]	✓	**94.0**	**37.5**	**60.0**	33.3	**40.5**	**59.4**	74.4	0.0	93.3	49.0	81.2	0.1	88.5	59.5	86.8	62.2	75.0	59.9	**46.5**	**58.0**
	Ours	✗	92.8	26.2	53.3	**49.6**	25.7	52.6	**75.7**	0.0	93.4	46.6	81.8	0.1	87.7	59.3	85.7	58.9	74.1	59.7	40.9	56.0
Test	Segmentation [15]	✓	82.4	26.0	34.6	21.6	18.3	6.7	2.7	0.5	91.8	65.0	75.1	27.7	87.4	58.6	80.5	55.1	64.8	47.9	55.9	47.5
	S-BKI(0.2) [7]	✓	84.0	28.5	39.9	25.2	19.7	7.9	3.3	0.0	92.3	**67.5**	76.5	28.5	89.1	61.5	82.3	61.6	66.5	55.3	64.4	50.2
	S-BKI(0.1) [6]	✓	83.8	30.6	43.0	26.0	19.6	8.5	3.4	0.0	**92.6**	65.3	**77.4**	**30.1**	**89.7**	**63.7**	**83.4**	**64.3**	**67.4**	**58.6**	**67.1**	51.3
	ConvBKI [7]	✓	83.8	**32.2**	**43.8**	29.8	23.2	8.3	3.1	0.0	91.4	62.6	75.2	27.5	89.1	61.6	81.6	62.5	65.2	53.9	63.0	50.4
	Ours	✗	**93.0**	24.0	36.7	**30.0**	**26.3**	**43.7**	**47.7**	**5.0**	92.2	67.3	76.0	28.4	89.5	62.3	82.3	60.8	66.9	54.7	61.0	**55.2**



(a) Frame 1291 (b) Frame 1292 (c) Frame 1293

Fig. 4: Segmentation estimates for the person category on three consecutive frames in sequence 16 of Semantic KITTI. Red: both Rangenet++ and our approach predict the person class, green: only Rangenet++ predicts person, blue: only our map predicts person. Our map is able to propagate semantic evidence for moving objects between frames using the velocity estimates, which compensates for noisy and inaccurate estimates by Rangenet++, while suppressing false positives on the ground due to accumulated evidence.

this prevents a conflict of dynamic objects moving into free space, where accumulated free space evidence may prevent the map state from changing to occupied. This is not a problem if free space is modeled by an absence of particles, as new particles can be generated around measurements. In ambiguous areas, such as near object boundaries, particles still collect evidence for both occupancy and free space.

Dynamic objects with changing velocity pose a challenge for our approach. Particles that track a dynamic object will accumulate increasing evidence for occupancy. If the velocity of the object changes, for example due to braking or turning, the linear motion model will cause the particle to move past the object boundary, usually into free or unobserved space. To address this, we decay the evidence of dynamic particles which do not receive an update to α_{occ} larger than a threshold $\Delta\alpha_{\text{occ,min}}$ over time as $\boldsymbol{\alpha}'_{\text{dyn}} = \gamma\boldsymbol{\alpha}_{\text{dyn}}$ with $\gamma \in [0, 1]$. This keeps the mean of the Dirichlet distribution, while increasing its variance and facilitates a faster state change in observed free space and a reversion to the unknown uniform prior in unobserved space.

IV. EXPERIMENTS AND RESULTS

We validate our approach by evaluating the accuracy of the estimated semantic class estimates and the velocity estimate of our approach against state of the art semantic mapping and DOGM methods.

A. Semantic Mapping

We evaluate the semantic mapping accuracy of our approach on Semantic KITTI [16], an automotive dataset consisting of 22 Sequences with semantic segmentation annotations in 3D for each LiDAR scan. We follow a similar evaluation methodology as Gan *et al.* [6] and Wilson *et al.* [7] and classify points of each LiDAR scan using our local semantic occupancy grid map. We use the semantic class predictions of Rangenet++ [15] and the sensor poses provided in the Semantic KITTI dataset as the input to our grid. Points that lie outside of the grid map bounds are classified using the estimates of Rangenet++. We maintain a local map with bounds [-50, -50, -2.6] to [50, 50, 2.6] m along the (X, Y, Z) axes of the LiDAR sensor and use a downsampling resolution of 0.2 m for the input point cloud. For estimating the semantic class of query points we use the point query approach described in Section III-E. We compare our approach with ConvBKI [7], as well as S-BKI [6] at two resolutions; the originally reported 0.1 m and the 0.2 m resolution reported by Wilson *et al.* [7]. For evaluation on the test set, we submit our results to the official evaluation server. These results are reported in Table I. While ConvBKI and S-BKI first construct a map using all scans in the sequence and then query this map, our approach is evaluated online without information from future measurements. Our approach runs in real time with 32 ms/scan on an Nvidia RTX 6000 Ada GPU, where S-BKI reports an inference time of 1670 ms/scan.

Despite not using data from future scans, we surpass the mapping performance of S-BKI with a voxel resolution of 0.2 m on the validation set based on the mean IoU. Compared to the input semantic segmentation by Rangenet++ [15], our approach improves the semantic segmentation of all categories except for road and sidewalk, where we report slightly decreased performance. We attribute this to the downsampling of the input point cloud, as even the global map by ConvBKI [7] with learned, class specific kernels shows lower performance on these categories while also downsampling the point cloud.

On the test set, we significantly outperform the static semantic mapping approaches. The largest improvements over the baselines appear on moving categories such as persons. We find that our approach effectively propagates accumulated evidence for dynamic objects between frames, as illustrated in Figure 4. Our dynamic map maintains the

TABLE II
RMSE OF VELOCITY ESTIMATES ON CARLASC (VAL).

Scenario	Class	Match	DSP-Map [3]	Ours
Light	Person	0.44	0.37	**0.19**
	Car	0.75	0.83	**0.58**
Medium	Person	0.44	0.49	**0.18**
	Car	0.64	1.11	**0.62**
Heavy	Person	0.63	0.76	**0.49**
	Car	**0.65**	1.07	**0.65**

correct classification of the shoulders and heads of two walking pedestrians based on prior collected evidence, while Rangenet++ misclassifies them. Static mapping methods are unable to compensate the misclassifications of Rangenet++, because the object motion prevents the accumulation of evidence in a static voxel.

B. Velocity Estimation

To evaluate the accuracy of the velocity estimation of our approach we use the synthetic CarlaSC dataset [17], because Semantic KITTI does not contain velocity ground truth information. CarlaSC contains LiDAR scans with pointwise semantic class and scene flow annotations.

The dataset contains two dynamic categories, persons and vehicles. To evaluate our approach, we use the ground truth class annotations to segment the LiDAR scan at each time frame. We obtain object instances by first segmenting the scan by class and then clustering each segment with DBScan [18]. The ground truth velocity of each cluster is calculated as the average scene flow of the cluster.

To evaluate our velocity estimates, we similarly segment the particle map using the current estimates of class probabilities and calculate the mean velocity for each cluster. These clusters are matched with the ground truth clusters using Hungarian matching based on the position of cluster centers. We evaluate the velocity estimates using the Root Mean Square Error (RMSE) for each movable category.

We compare our approach to the Hungarian matching described in Section III-F used to initialize the particle velocities and DSP-Map, which represents the state of the art of velocity estimation in dynamic 3D occupancy grids. For a fair comparison, we modify DSP-Map to also contain and update semantic class labels in each particle and cluster them analogously to our approach. Both the modified DSP-Map and our approach receive the same input. We evaluate on the validation set of CarlaSC, which consists of three scenarios with light, medium, and heavy traffic densities, respectively.

The results of our quantitative evaluation are presented in Table II. We observe that our approach improves the accuracy of the estimated velocities for both categories, although the improvements lessen in more densely populated scenes and are less substantial for the Car category than for the Person category. There are two reasons for this. Firstly, as cars are larger in volume, particles with an erroneous velocity can persist and be confirmed by measurements for several time steps before moving past the object boundary. This problem

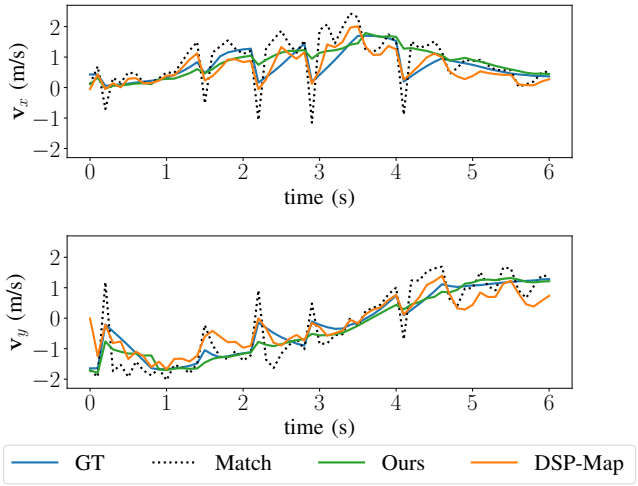


Fig. 5: Estimated velocity by Hungarian matching, DSP-Map and our approach for a turning pedestrian in CarlaSC.

also motivated Chen *et al.* [3] to initialize velocities using cluster matching. Secondly, cars pass each other more frequently than pedestrians, especially in the populated scenes of CarlaSC, which allows particles to transfer between object instances without being depleted.

We also compare our approach to DSP-Map and find that outside of the person category in the Light scenario, DSP-Map does not offer improvements and often worsens the initial velocity estimates. To gain more insight into this, we plot the velocity estimates in X and Y direction for a turning pedestrian in the Light Sequence in Figure 5. We observe that both DSP-Map and our approach offer a smoothing of the noisy Hungarian Matching initialization, but DSP-Map is more sensitive to changes in the initialization velocity. Due to the perfect scene flow annotations in the synthetic data, variations in the ground truth velocity from the walking pattern can be observed. DSP-Map tracks these patterns more closely, but at the cost of an overall higher degree of noise. Our approach does not closely track these patterns, but still follows the overall trend of the ground truth velocity closely.

The higher degree of noise in DSP-Map has a negative effect in more populated scenarios, because of more frequent occlusions and interfering particles. We visualize the difference in quality between our approach and DSP-Map in a scenario where two pedestrians closely pass each other in Figure 6. Our approach clearly distinguishes the individuals, and accurately estimates their velocities, while DSP-Map predicts almost static objects. The oncoming pedestrian (right) leaves a particle trail in the occluded area behind them, which can cause further particle interference. This occurs because DSP-Map does not update occluded particles past a certain distance to maintain consistent static elements in the map. Our approach is able to avoid such trails by decaying particles of movable categories in unobserved space quickly, while maintaining immovable categories.

C. Variance estimation

As described in Section III-E we calculate both the occupancy variance and the semantic variance separately. The

(a) DSP-Map (b) Ours

Fig. 6: Estimate by DSP-Map and our approach of two pedestrians walking past each other in opposite directions in CarlaSC. The velocity estimates of DSP-Map reduce to almost zero, as noisy particles with low weight interfere with one another. A trail of particles is left in the occluded space behind the right pedestrian. Our approach maintains an accurate velocity estimate without trailing particles, due to rapidly decaying particles past the object boundary.

benefit of this distinction is shown in Figure 7. Occupancy variance tends to be larger at the boundaries to free or unknown space. Semantic variance is high in semantically ambiguous areas and at class boundaries.

V. Conclusion

This paper presented a unification of BGKI and DOGMs to produce a local semantic dynamic 3D occupancy grid map that is recursively updated. Experimental results on real and simulated data show that our method produces occupancy maps that are more persistent for static objects and estimate the velocity of dynamic objects more accurately than the current state of the art on sparse LiDAR data, without relying on prior scene flow estimation.

The constant velocity assumption limits the tracking capabilities of the current approach and requires re-initialization of particles to track changing velocities. Promising areas for future research include updating the velocity of particles using sensor measurements and integrating panoptic segmentation to further enrich the dynamic map.

$$\Delta\boldsymbol{\alpha}_i \;=\; \sum_{j=1}^{N} K(d_{\mathbf{p}_i,\mathbf{s}_j})\mathbf{c}_j,\; i \;>\; 0$$

References

[1] G. Tanzmeister, J. Thomas, D. Wollherr, and M. Buss, "Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 6090–6095.

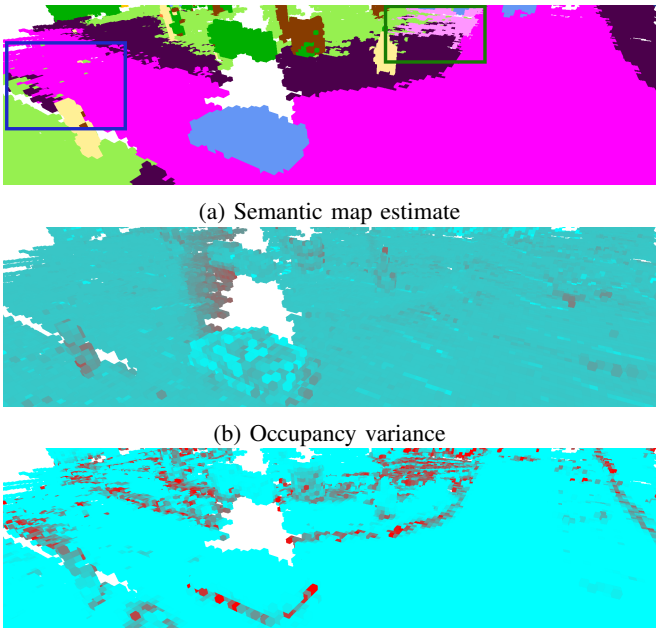
[2] D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, and K. Dietmayer, "A random finite set approach for dynamic occupancy grid maps with real-time application," *The Int. Journal of Robotics Research*, vol. 37, no. 8, pp. 841–866, 2018.

[3] G. Chen, W. Dong, P. Peng, J. Alonso-Mora, and X. Zhu, "Continuous occupancy mapping in dynamic environments using particles," *IEEE Transactions on Robotics*, 2023.

[4] Y. Min, D.-U. Kim, and H.-L. Choi, "Kernel-based 3-d dynamic occupancy mapping with particle tracking," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2021, pp. 5268–5274.

[5] J. Han, Y. Min, H.-J. Chae, B.-M. Jeong, and H.-L. Choi, "Ds-k3dom: 3-d dynamic occupancy mapping with kernel inference and dempster-shafer evidential theory," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2023, pp. 6217–6223.

[6] L. Gan, R. Zhang, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, "Bayesian spatial kernel smoothing for scalable dense semantic mapping," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 790–797, 2020.



(a) Semantic map estimate

(b) Occupancy variance

(c) Semantic variance

Fig. 7: Semantic and variance output of the map. Low variance is shown in cyan, high variance in red. Occupancy variance is high towards object boundaries and unexplored space, semantic variance is high at object boundaries and ambiguous areas, such as the ground in the background (green box) or the incorrectly classified road (blue box).

[7] J. Wilson, Y. Fu, A. Zhang, J. Song, A. Capodieci, P. Jayakumar, K. Barton, and M. Ghaffari, "Convolutional bayesian kernel inference for 3d semantic mapping," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2023, pp. 8364–8370.

[8] K. Doherty, J. Wang, and B. Englot, "Bayesian generalized kernel inference for occupancy map prediction," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 3118–3124.

[9] K. Doherty, T. Shan, J. Wang, and B. Englot, "Learning-aided 3-d occupancy mapping with bayesian generalized kernel inference," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 953–966, 2019.

[10] D. Kochanov, A. Ošep, J. Stückler, and B. Leibe, "Scene flow propagation for semantic mapping and object discovery in dynamic street scenes," in *Proc.IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 1785–1792.

[11] A. Unnikrishnan, J. Wilson, L. Gan, A. Capodieci, P. Jayakumar, K. Barton, and M. Ghaffari, "Dynamic semantic occupancy mapping using 3d scene flow and closed-form bayesian inference," *IEEE Access*, vol. 10, pp. 97 954–97 970, 2022.

[12] W. R. Vega-Brown, M. Doniec, and N. G. Roy, "Nonparametric bayesian inference on multivariate exponential families," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[13] S. Steyer, G. Tanzmeister, and D. Wollherr, "Grid-based environment estimation using evidential mapping and particle tracking," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 384–396, 2018.

[14] C. Buerkle, F. Oboril, J. Jarquin, and K.-U. Scholl, "Efficient dynamic occupancy grid mapping using non-uniform cell representation," in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, 2020, pp. 1629–1634.

[15] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2019, pp. 4213–4220.

[16] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2019.

[17] J. Wilson, J. Song, Y. Fu, A. Zhang, A. Capodieci, P. Jayakumar, K. Barton, and M. Ghaffari, "Motionsc: Data set and network for real-

time semantic mapping in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8439–8446, 2022.

[18] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.