SEE-CSOM: Sharp-Edged and Efficient Continuous Semantic Occupancy Mapping for Mobile Robots

Yinan Deng, Meiling Wang, Yi Yang, Danwei Wang, *Life Fellow, IEEE*, and Yufeng Yue, *Member, IEEE*

***Abstract*—Generating an accurate and continuous semantic occupancy map is a key component of autonomous robotics. Most existing continuous semantic occupancy mapping methods neglect the potential differences between voxels, which reconstruct an overinflated map. What is more, these methods have high computational complexity due to the fixed and large query range. To address the challenges of overinflation and inefficiency, this article proposes a novel sharp-edged and efficient continuous semantic occupancy mapping algorithm (SEE-CSOM). The main contribution of this work is to design the Redundant Voxel Filter Model (RVFM) and the Adaptive Kernel Length Model (AKLM) to improve the performance of the map. RVFM applies context entropy to filter out the redundant voxels with a low degree of confidence, so that the representation of objects will have accurate boundaries with sharp edges. AKLM adaptively adjusts the kernel length with class entropy, which reduces the amount of data used for training. Then, the multientropy kernel inference function is formulated to integrate the two models to generate the continuous semantic occupancy map. The algorithm has been verified on indoor and outdoor public datasets and implemented on a real robot platform, validating the significant improvement in accuracy and efficiency.**

***Index Terms*—Mobile robots, semantic mapping, Bayesian rule, kernel inference.**

Manuscript received 13 April 2022; revised 7 September 2022 and 28 December 2022; accepted 22 March 2023. Date of publication 5 April 2023; date of current version 16 August 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62003039, Grant 62233002, Grant U193203, and in part by the CAST program under Grant YESS20200126, and Collective Intelligence & Collaboration Laboratory. *(Corresponding author: Yufeng Yue.)*

Yinan Deng, Meiling Wang, Yi Yang, and Yufeng Yue are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: dengyinan@bit.edu.cn; wangml@bit.edu.cn; yang_yi@bit.edu.cn; yueyufeng@bit.edu.cn).

Danwei Wang is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: edwwang@ntu.edu.sg).

The code is available at https://github.com/BIT-DYN/SEE-CSOM.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2023.3262857.

Digital Object Identifier 10.1109/TIE.2023.3262857

I. Introduction

THE essence of robot mapping is to employ sparse noisy sensor observations to construct a dense accurate representation, which is regarded as a fundamental problem in robotics [1]. As robots are required to perform more intelligent tasks, incorporating semantic information can further help them distinguish object categories and allow a higher level of environmental representation [2]. Currently, the most widely used mapping technique is the occupancy grid map [3]. Most grid mapping methods are noncontinuous, assuming that voxels are statistically independent, which contradicts the fact that real-world object surfaces are usually smooth. Recent success in Bayesian kernel inference has boosted the development of continuous mapping, such as BGKOctoMap [4], BGKOctoMap-L [5], and S-BKI [6]. They incorporate local spatial correlations into the mapping model, which can infer the continuous surface from sparse sensor data. However, the potential differences between voxels are not fully exploited and all voxels are treated equally, so the voxels next to the object are misclassified to be occupied, resulting in overinflated objects. Such an overinflated map is not suitable for robot navigation tasks, because traversable free space might be falsely blocked. Therefore, the main objective of this article is to design a novel continuous semantic occupancy mapping algorithm that can mitigate overinflation while improving efficiency.

The first challenge is to infer the voxels that are worth filling, so as to mitigate the overinflation phenomenon of existing continuous mapping methods [4], [5], [6]. As shown in Fig. 1, the discrete map only recovers the area hit by the sensor observations, leaving many loopholes. These mapped voxels are defined as *observed voxels* in this article. Other unknown voxels can be divided into two types: the voxels located in the loopholes due to the lack of observations are called *inactive voxels* (such as Fig. 1 yellow masks), while those in the free space outside the obstacle surface are called *redundant voxels* (such as Fig. 1 red masks). Continuous mapping is expected to only fill in *inactive voxels*, generating a representation similar to the ground truth map. However, SOTA continuous mapping method S-BKI [6] does not distinguish unknown voxels and builds an overinflated map due to falsely filling in redundant voxels (see Fig. 1). To accurately reconstruct the scene, the redundant voxel filter model is proposed to filter out redundant voxels by measuring

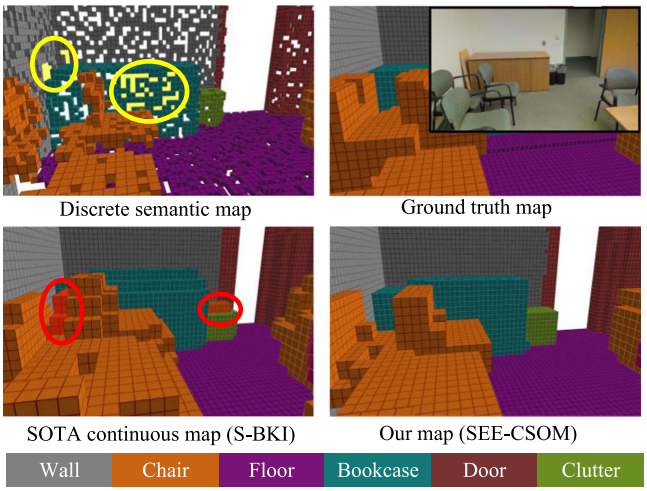

Fig. 1. Demonstration of a conference room of Stanford 2-D–3-D Semantic Dataset [7]. Our map restores smooth surfaces for the objects while generating precise boundaries with sharp edges.

context entropy, which aims to increase the confidence level in the inference process.

The second challenge is to reduce the computational complexity of continuous semantic occupancy mapping, thereby improving efficiency. Current continuous approaches [8], [9], [10] adopt fixed kernel length, which is n times of the voxel size. This operation will increase the computational complexity by n^3 compared to discrete approaches. To reduce the time cost, the Adaptive Kernel Length Model is proposed to adjust the kernel length adaptively by introducing class entropy, which is the measurement of the overall uncertainty of a voxel. A large class entropy usually indicates that the voxel is located at the junction of objects or contains noisy observations, therefore a large query range is needed to improve accuracy. When the class entropy is small, a small kernel length can satisfy the accuracy.

In summary, overinflation and inefficiency are two challenging problems of continuous semantic occupancy mapping. This article proposes a novel sharp-edged and efficient continuous semantic occupancy mapping algorithm (SEE-CSOM) by extending [11]. The overall continuous semantic occupancy mapping problem is mathematically formulated and its probabilistic model is derived. The main contributions of this work are listed as follows.

1) Redundant voxel filter model (RVFM) is proposed to filter out redundant voxels distinguished by context entropy, which moderates the overinflation phenomenon.
2) Adaptive kernel length model (AKLM) is proposed to assign an appropriate kernel length to each voxel by class entropy, which improves the mapping efficiency.
3) The proposed algorithm has been verified on indoor and outdoor public datasets and a real robot. Qualitative and quantitative results show the superiority of the algorithm in improving accuracy and efficiency.

The rest of this article organized as follows. Section II reviews the related works. Section III introduces the SEE-CSOM algorithm. Section IV shows the experimental results. Section V concludes this article.

II. Related Works

In this section, existing algorithms related to continuous semantic occupancy mapping are introduced, including semantic mapping and continuous mapping.

A. Semantic Mapping

With the rapid development of deep learning, semantic mapping has attracted increasing attention. Early semantic mapping methods directly use semantic images for mapping. The authors in [12] use the Bayesian framework to filter probabilistic segmentation from multiple views in a voxel-based 3-D map. In [13], the street-level image label estimates are aggregated to annotate the 3-D volume. These methods are the pioneers of semantic mapping, but lack further optimization. To optimize incorrect voxel labels, CRF has become a research hotspot [14], which can simulate the long-distance relationships in a region, such as grids corresponding to 2-D superpixels [15] or grids within supervoxels [16]. In [17], a novel high-order CRF model is applied to optimize 3-D grid labels. Recently, a hierarchical framework for collaborative probabilistic semantic mapping is proposed in [18]. Besides, relative location among robots can be estimated by matching semantic maps [19], [20].

Various methods mentioned above promoted the development of semantic mapping. However, these methods assume that the voxels are independent and do not reconstruct the continuous surface of the objects.

B. Continuous Mapping

To construct a smoother occupancy map, many methods have attempted to relax the assumption that the voxels are independent, such as GPmap [8], Hilbert map [21], etc. GPmap [8] introduced a dependence relationship between points as the nonparametric Bayesian inference process, which has also been extended to semantic mapping [10]. However, $\mathcal{O}(n^3)$ computational complexity has limited its application to large-scale online mapping [9]. Hilbert map [21] makes use of fast kernel approximations to enable faster training in $\mathcal{O}(n)$ time. In addition, a real-time incremental 3-D Hilbert map has been proven to be feasible [22]. Recently, Bayesian kernel inference with $\mathcal{O}(\log n)$ computational complexity has begun to gain attention. BGKOctoMap [4] innovatively applies the sparse kernel and Bayesian nonparametric inference data structure to improve efficiency. Similar work is carried out in BGKOctoMap-L [5]. More recently, S-BKI [6] extends [5] to 3-D semantic mapping, which enriches the map information.

In summary, the above methods perform continuous mapping without considering voxel potential differences, which results in overinflated maps. These algorithms also have a large computational cost compared to discrete mapping methods. These have been the reasons for limiting the generalization of continuous mapping.

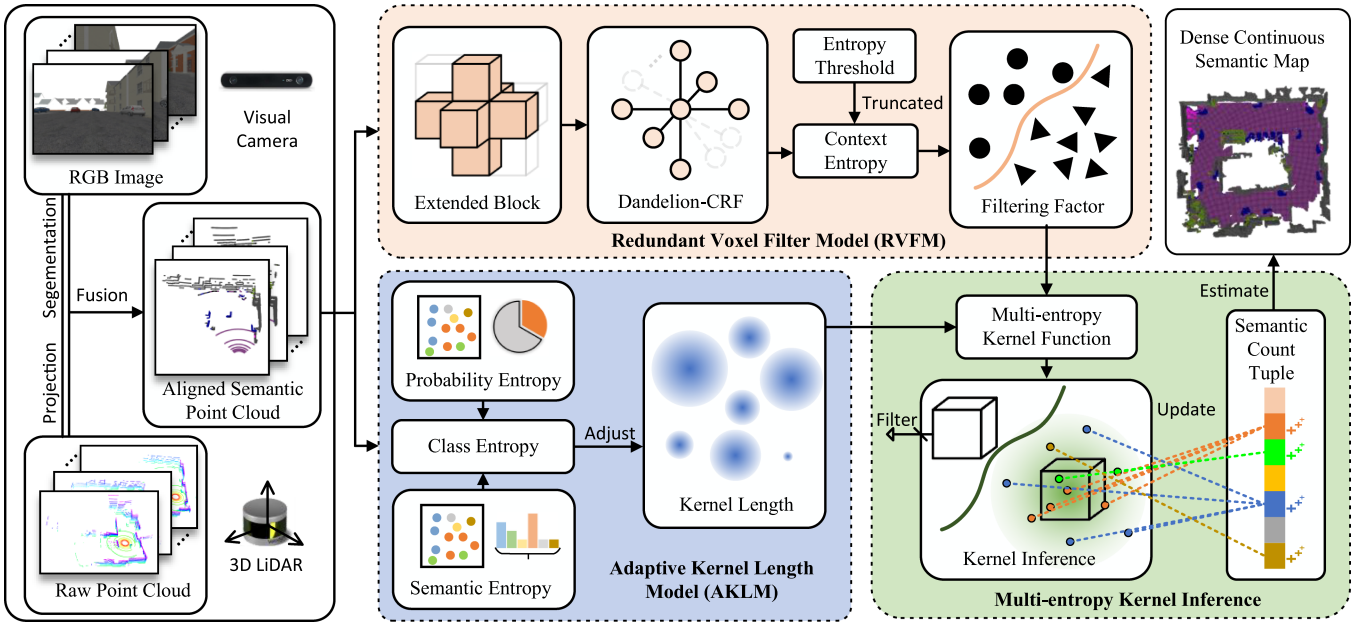

Fig. 2. Framework of sharp-edged and efficient continuous semantic occupancy mapping algorithm.

III. Sharp-Edged and Efficient Continuous Semantic Occupancy Mapping

This section describes and formulates the SEE-CSOM algorithm, divided into four sections: Algorithm framework and problem definition, redundant voxel filter model, adaptive kernel length model, and multientropy kernel inference.

A. Algorithm Framework and Problem Definition

The framework of the SEE-CSOM algorithm is depicted in Fig. 2, which consists of three main modules. In the redundant voxel filter model (RVFM), redundant voxels are distinguished from inactive and observed voxels by filtering factor. In the adaptive kernel length model (AKLM), class entropy composed of two subentropies is introduced to adjust the kernel length, which determines the range of local spatial associations. Finally, a continuous semantic map is estimated from sensor observations through multientropy kernel inference that combines the information conveyed by RVFM and AKLM.

Considering a robot operating in a completely unknown environment and attempting to reconstruct the surroundings, the problem can be defined as follows:

Problem Definition: Given a robot r with camera observations $I_{1:t}$, 3-D LiDAR observations $L_{1:t}$ and robot trajectory $O_{1:t}$, the objective is to estimate the continuous semantic occupancy map $\mathcal{M}_t$

$$p(\mathcal{M}_t|I_{1:t}, L_{1:t}, O_{1:t}). \tag{1}$$

The solution of the problem corresponds to the maximum a posterior (MAP) estimation of (1). For the input, there are $I_t \in \mathbb{R}^2$, $L_t \in \mathbb{R}^3$ and $O_t \in SE(3)$. For output, the dense semantic map $\mathcal{M} \triangleq \{v_j\}_{j=1}^{N_\mathcal{M}}$ consists of a set of voxels. Each voxel v_j contains the 3-D coordinate of the center (v_j^x, v_j^y, v_j^z), associated with a tuple $\lambda_j = (\lambda_j^1, \lambda_j^2, \cdots \lambda_j^K)$ to store probabilistic semantic labels, where K is the total number of semantic classes and $\sum_{k=1}^K \lambda_j^k = 1$.

At time t, the RGB image I_t is fed into the segmentation network [23]. For each pixel, the output is a one-hot encoded measurement tuple $c_i = (c_i^1, c_i^2, \cdots c_i^K)$. Due to differences in the sensor's field of perception, only 3-D LiDAR points within the camera perception area are collected. The semantic labels can be transmitted from pixels to LiDAR points by projection [24], where the parameters are calibrated by [25]. Therefore, (1) can be rewritten as

$$p(\mathcal{M}_t|I_{1:t}, L_{1:t}, O_{1:t}) = p(\mathcal{M}_t|L_{s_{1:t}}). \tag{2}$$

The semantic point cloud $L_s \triangleq \{p_i\}_{i=1}^{N_{Ls}}$ consists of a series of semantic points p_i referred by coordinates (p_i^x, p_i^y, p_i^z), which are associated with semantic label c_i. Alternatively, the problem can be refined to:

Given semantic points and labels $\{p_i, c_i\}_{i=1}^{N_{Ls}}$, the objective is to estimate probabilistic semantic labels λ_j of each voxel v_j

$$p(\mathcal{M}_t|L_{s_{1:t}}) = \prod_{j=1}^{N_\mathcal{M}} \prod_{i=1}^{N_{Ls}} p(\lambda_j|v_j, p_i, c_i). \tag{3}$$

B. Redundant Voxel Filter Model

As stated before, previous continuous mapping methods [5], [6] cannot clearly distinguish voxels, resulting in overfitting of the final continuous map. The redundant voxel filter model is designed to address this problem, distinguishing different types of voxels and filtering out the redundant voxels. Fig. 3 illustrates the significance of RVFM in a 2-D example.

As shown in Fig. 4(a), in order to improve semantic accuracy and inference efficiency, *block* is introduced as an intermediate

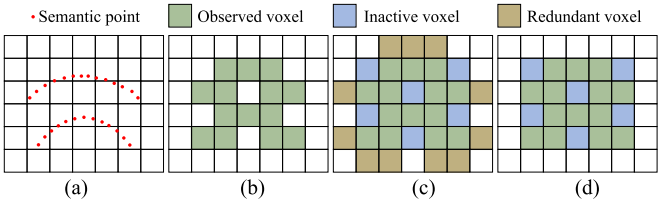

Fig. 3. 2-D demonstration of RVFM with the input of semantic point cloud. Colored boxes represent filled voxels. (a) Semantic point cloud inserted into the map. (b) Discrete Map with a obvious loophole. (c) Continuous map built without filtering. (d) Continuous map built with filtering.

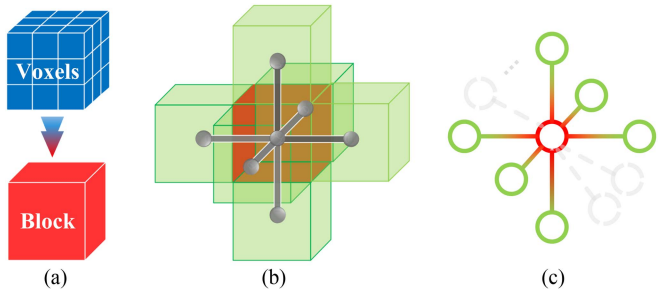

Fig. 4. Illustration of voxels and blocks. (a) Red is a block b_J composed of several black voxels $\{v_j\}$. (b) Green blocks $\{b_x\}$ and current block b_J together form the extended block B_J. (c) Dandelion-CRF extracted from B_J. The gray part implies its flexible scalability.

layer between *map* and *voxel*. Each block $b_J \triangleq \{v_j\}_{j\in J}$ is a small semantic octree, comprised of several adjacent voxels v_j. The depth of the octree is defined as block depth $\mathbf{D}_b$. In consideration of the time cost, the filtering of redundant voxels is performed in block units, so all the voxels v_j in the block b_J will inherit the attributes of the parent block. To clarify the state around the current block b_J, extended block $B_J \triangleq \{b_J, \{b_x\}\}$ [see Fig. 4(b)] is also introduced, which consists of the current block b_J and neighboring blocks $\{b_x\}$ around b_J. Each block b_J independently owns a unique extension block.

A graph model is extracted from the extended block [see Fig. 4(c)], with blocks in the extended block as nodes, and the connection between the current block b_J and the surrounding blocks $\{b_x\}$ as edges. This graph is called Dandelion-CRF because of its highly recognizable structure. It allows arbitrarily modifying the style of the expansion block to suit the map application and sensor resolution. Given the observation D, the context entropy $\mathbf{E}_{con}$ is described as the conditional probability of the central node b_J

$$\mathbf{E}^J_{con} = P(b_J \sim 1|D) \tag{4}$$

where $b_J \sim 1$ indicates that b_J should be filled to enhance the continuity of the map. It is important to note that filling does not set the state to be occupied, but instead uses the spatial association to populate current observation.

There are two kinds of cliques in Dandelion-CRF: One is a single node $\{b_k\}$ and the other is a pair of adjacent nodes $\{b_k, b_l\}$, where k and l are index variables. By selecting the exponential potential function and introducing the feature function, the conditional probability is defined as

$$P(b_J \sim 1|D) = \frac{1}{Z(D)} \exp(E(b_J \sim 1|D)) \tag{5}$$

$$E(b_J \sim 1|D) = \psi(b_J) + \sum_x (\psi(b_x)\psi(b_J, b_x)) \tag{6}$$

where $Z(D)$ is the partial function for normalization, the status feature function $\psi(b_k)$, and the transition feature function $\psi(b_k, b_l)$ describe the influence of the observation sequence and adjacent nodes, respectively. In the formulation, $\psi(b_k)$ obtains different values according to whether the block is observed. $\psi(b_k, b_l)$ takes the radial basis function (RBF) of the Euclidean distance between blocks. In (7) and (8), ω_1 and ω_2 are hyperparameters to control the amount of information transmitted, and s is the resolution of the block

$$\psi(b_k) = \begin{cases} \omega_1, & \exists p_i \in b_k \\ 0, & \forall p_i \notin b_k \end{cases} \tag{7}$$

$$\psi(b_k, b_l) = \frac{\omega_2}{\omega_1} \exp\left(-\frac{\|b_k - b_l\|^2}{2s^2}\right). \tag{8}$$

$\mathbf{E}_{con}$ reveals potential differences between voxels that can be used as indicators of differentiation. Taking $\mathbf{T}_{con}$ as the entropy threshold, the voxels contained in the block with context entropy less than $\mathbf{T}_{con}$ are redundant, often located in the gap between two objects or outside the boundaries of objects. RVFM will filter out these redundant voxels during continuous inference, while preserving observed and inactive voxels to estimate a more accurate map. This operation is realized by the filtering factor f_J transmitted to the multientropy kernel inference module

$$f_J = [\mathbf{E}^J_{con} \geq \mathbf{T}_{con}]. \tag{9}$$

C. Adaptive Kernel Length Model

The kernel length is the key to mapping efficiency, because it determines the query range. The adaptive kernel length model is designed to assign appropriate kernel lengths to voxels. Continuing in units of *block*, voxels in the same block are assigned with the same kernel length.

Class entropy $\mathbf{E}_{cla}$ is introduced to measure the overall uncertainty of the voxel. It contains two subentropies: one is the probability entropy $\mathbf{E}_p$, and the other is semantic entropy $\mathbf{E}_s$. On the one hand, probability entropy $\mathbf{E}_p$ reflects the proportion of number, which is defined as

$$\mathbf{E}_p = \frac{\mathfrak{n}_{all} - \mathfrak{n}_{\max}}{\mathfrak{n}_{all}} \tag{10}$$

where $\mathfrak{n}_{\max}$ is the number of semantic points that account for the largest number among all semantic classes, and $\mathfrak{n}_{all}$ is the total number of semantic points in block b_J. On the other hand, semantic entropy $\mathbf{E}_s$ describes the diversity of semantic labels in block b_J. Defining $\mathfrak{k}$ $(\mathfrak{k} < K)$ to indicate the number of semantic labels contained in block b_J, semantic entropy $\mathbf{E}_s$

$$\mathbf{E}_s = \log_K(\mathfrak{k}) = \frac{\ln(\mathfrak{k})}{\ln(K)}. \tag{11}$$

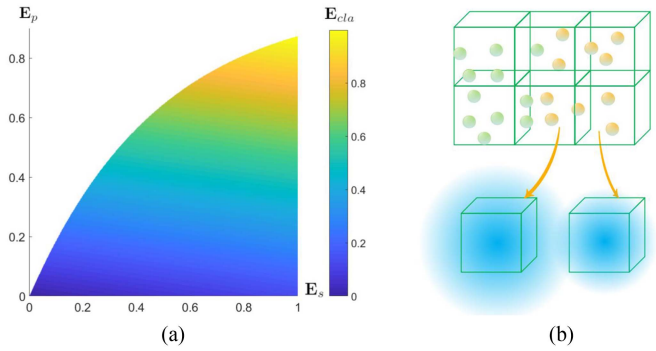

Fig. 5. (a) Value of class entropy $\mathbf{E}_{cla}$ in the subentropies plane. (b) Adaptive kernel length for different blocks.

It is worth pointing out that $\mathfrak{n}_{\max}/\mathfrak{n}_{all}$ has a bound $[1/\mathfrak{k}, 1]$. Converting this mathematical relationship to subentropies, the implicit constraint of subentropies can be obtained

$$\ln(1-\mathbf{E}_p)+K\ln(\mathbf{E}_s)\geq 0. \tag{12}$$

Class entropy $\mathbf{E}_{cla}$ is defined in (13) to combine two subentropies. Probability entropy $\mathbf{E}_p$ is dominant because it integrates part information of semantic entropy $\mathbf{E}_s$. Moreover, the weight of semantic entropy should be inversely proportional to the total number of semantic classes K. Coupled with the constraints of (12), the visualization of class entropy $\mathbf{E}_{cla}$ is illustrated in Fig. 5(a)

$$\mathbf{E}_{cla}^{J}=\mathbf{E}_p+\frac{1}{K}\mathbf{E}_s. \tag{13}$$

When there are no observations in the block, it will have the largest class entropy. This also occurs when points with any labels fall into the block evenly. Substituting (10), (11), into (13), class entropy $\mathbf{E}_{cla}^{J}$ is written as

$$\mathbf{E}_{cla}^{J}=\begin{cases}\frac{\mathfrak{n}_{all}-\mathfrak{n}_{\max}}{\mathfrak{n}_{all}}+\frac{\log_K(\mathfrak{k})}{K} & \exists p_i\in b_J\\ 1 & \text{otherwise.}\end{cases} \tag{14}$$

Larger class entropy means higher uncertainty, requiring a larger query range to ensure map accuracy. Therefore, for voxel v_j in block b_J, the kernel length with bounds $L_{\min}$ and $L_{\max}$ is adjusted to

$$L_J=L_{\min}+\mathbf{E}_{cla}^{J}(L_{\max}-L_{\min}). \tag{15}$$

D. Multientropy Kernel Inference

The efficacy of the RVFM and AKLM needs to be exerted through multientropy kernel inference, which essentially converts sensor observations into updated maps. Different from the classical voxel probability update model [3], the multientropy kernel inference model is derived based on the counting sensor model [26]. According to the Bayesian rule, (3) can be decomposed into

$$p(\lambda_j|\,v_j,p_i,c_i)\propto p(\lambda_j)p(\,c_i|\,v_j,p_i,\lambda_j). \tag{16}$$

For incremental Bayesian inference, likelihood probability is modeled as a categorical distribution $cat(\lambda_j^1,\lambda_j^2,\dots\lambda_j^K)$. And both prior probability and posterior probability satisfy Dirichlet distribution $\mathrm{Dir}(K,\sigma_0)$ and $\mathrm{Dir}(K,\sigma_j)$, where $\sigma_0=\{\sigma_0^1,\sigma_0^2,\dots\sigma_0^k\}$ and $\sigma_j=\{\sigma_j^1,\sigma_j^2,\dots\sigma_j^k\}$ are the distribution parameters.

To break the independence of voxels, an extended likelihood is introduced with a kernel function k that operates on 3-D space $\mathcal{X}\times\mathcal{X}\to[0,1]$. Then, (16) is rewritten as

$$\prod_{k=1}^{K}(\theta_j^k)^{\sigma_j^k-1}\propto\prod_{k=1}^{K}(\theta_j^k)^{\sigma_0^k-1}\left(\prod_{k=1}^{K}(\theta_j^k)^{c_i^k}\right)^{k(v_j,p_i)}. \tag{17}$$

After simplification, the relationship between the two Dirichlet distribution parameters σ_0 and σ_j can be obtained

$$\sigma_j=\sigma_0+k(v_j,p_i)c_i. \tag{18}$$

Because σ_0 usually takes a tiny value due to the lack of prior knowledge, σ_j is the weighted count of the semantic points, called the semantic count tuple of voxel v_j. It will be stored in the voxel and updated when a new semantic point cloud is inserted.

Equation (18) indicates that the semantic count tuple σ_j counts not only the semantic points that fall into the current voxel v_j but also the adjacent semantic points with the kernel function as the weight. In this way, the choice of kernel function has a pivotal influence on the quality and efficiency of semantic mapping. In order to reduce the computational complexity, the sparse kernel function $k_0(v_j,p_i)$ [27] is chosen as a template:

$$k_0(v_j,p_i)=\underset{d<L}{\mathbf{I}}\,\varepsilon_0\left[\frac{(2+\cos(2\pi\frac{d}{L})(1-\frac{d}{L}))}{3}+\frac{\sin(2\pi\frac{d}{L})}{2\pi}\right] \tag{19}$$

where $\mathbf{I}$ represents the indicator function, $d=\|v_j-p_i\|$, L is the kernel length, and ε_0 is the scale factor.

Incorporating the proposed RVFM (9) and AKLM (15) into the (19), the multientropy kernel function $k_e(v_j,p_i)$ is derived as (20). $k_e(v_j,p_i)$ improves the defects of indistinguishable redundant voxels and fixed kernel length in $k_0(v_j,p_i)$ by the integration of the two models, thus helping to greatly improve the mapping performance

$$k_e(v_j,p_i)=f_J k_0(v_j,p_i)_{L\to L_J}. \tag{20}$$

Inserting the obtained semantic point clouds $L_{s_{1:t}}$, the semantic count tuple σ_j of each voxel v_j in map $\mathcal{M}_t$ is denoted as

$$\sigma_j^k=\sigma_0^k+\sum_{i=1}^{N_{L_s}}k_e(v_j,p_i)c_i^k,k\in\mathcal{K} \tag{21}$$

where σ_j^k represents the count value of the 3-D point with semantic label k in voxel v_j.

Therefore, the probabilistic semantic label of the voxel v_j is the closed-form expected value of the posterior Dirichlet:

$$\lambda_j^k=\frac{\sigma_j^k}{\sum_{m=1}^{K}\sigma_j^m}=\frac{\sigma_0^k+\sum_{i=1}^{N_{L_s}}k_e(v_j,p_i)c_i^k}{\sum_{m=1}^{K}\left(\sigma_0^m+\sum_{i=1}^{N_{L_s}}k_e(v_j,p_i)c_i^m\right)}. \tag{22}$$

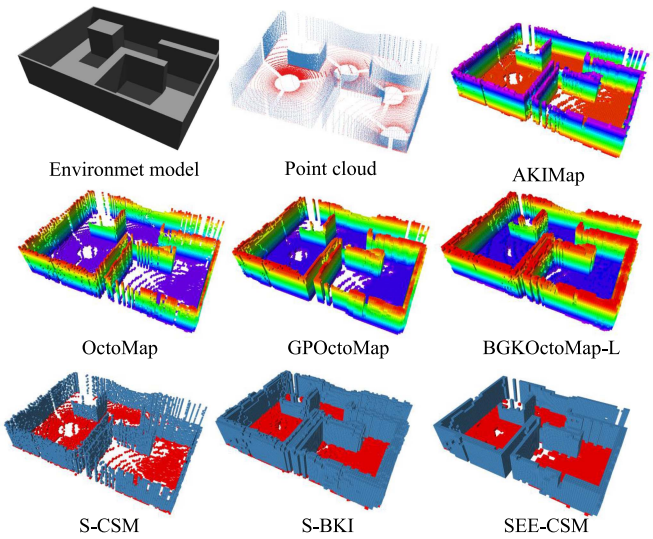

Fig. 6. Result of mapping on the structured dataset.

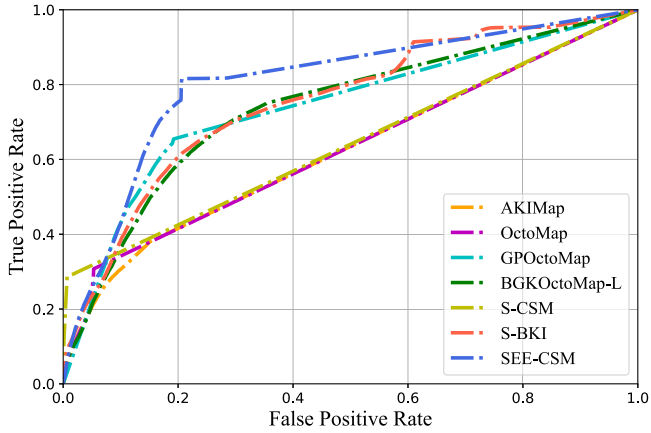

Fig. 7. ROC curve of structured toy dataset.

In summary, the mapping problem defined in (3) has been transformed into a probabilistic solution as (22). The map is updated by incrementally calculating (21) and (22) as new sensor observations are obtained.

IV. Experimental Results

In this section, the performance of the proposed SEE-CSOM algorithm is validated through experiments performed on multiple public datasets and a real robot platform.

Implementation Details: All experiments are conducted on AMD R7-5800H CPU @3.20 GHz, 16 GB RAM. Our code written in C++ has been made public, which is based on Robot Operating System, Point Cloud Library, and Semantic Bayesian Kernel Inference Library.

Comparison Baseline: S-CSM, S-BKI [6], OctoMap [3], GPOctoMap [9], BGKOctoMap-L [5], and AKIMap [28] are selected as baselines for occupancy accuracy and efficiency comparison, while S-CSM and S-BKI [6] are set as the baselines for semantic accuracy comparison. For these algorithms, the common hyperparameters follow the settings of S-BKI [6], while the unique ones are kept original or empirically adjusted to be appropriate.

Evaluation Metric: Accuracy is measured by occupancy AUC, voxel-IoU, w-average, and accurateness. Voxel-IoU extends the pixel-IoU from 2-D to 3-D, which is defined as TP/(TP+FP+FN). W-average is the weighted average of voxel-IoU. Accurateness is defined as the proportion of correctly classified voxels. Efficiency is measured in seconds.

A. Occupancy Evaluation

The structured toy dataset is collected from a closed space of 10.0 m × 7.0 m × 2.0 m in Gazebo, which is also used in many previous works [4], [5], [6]. Fig. 6 includes the reconstruction results for ours and all of the comparison baselines. Our map has a compact underlying representation even though the sensor data are sufficiently sparse, which is the closest to the environment model, while other methods suffer from underfitting or overfitting to a certain degree.

To quantitatively evaluate the maps, the receiver operating characteristic (ROC) curves are plotted in Fig. 7 and their numerical results are shown in Table II. By smoothing in the tangent plane of the object surface, the proposed algorithm achieves a promising occupancy classification effect, which is crucial for robots to avoid obstacles in unknown environments.

B. Stanford Indoor Dataset

Stanford 2-D–3-D Semantics Dataset [7] is a large indoor spatial dataset. It provides indoor data at multiple modalities, including annotated 3-D point clouds. A conference room, a lounge, an office, and a WC are selected as evaluation scenes, covering various indoor environments with different structures. The map resolution is set to 0.05 m for all algorithms.

Taking the lounge as an example, the mapping results are shown in Fig. 8. As can be seen, S-CSM generates a discrete semantic map by only predicting observed voxels. Objects in the S-BKI map are very thick, and a zoom-in view shows that the entire chair has been distorted and glued to the floor due to the blind filling of redundant voxels. In contrast, our proposed SEE-CSOM successfully filters out redundant voxels while filling in the inactive voxels, which builds a semantic map that visually has the most similar features to the ground truth.

The quantitative evaluation results of the mapping accuracy are summarized in Table I. SEE-CSOM has achieved significant advantages, consistent with the visual results. S-BKI has a higher IoU than S-CSM, but has the lowest accuracy due to the overfilling of a large number of redundant voxels.

C. SemanticKITTI Outdoor Dataset

SemanticKITTI dataset [29] is a large outdoor semantic point cloud dataset, collected from the real world. The semantic labels of the point clouds inserted into the map are obtained by RangeNet++ [30]. Sequences 02, 04, 06, and 08, four different

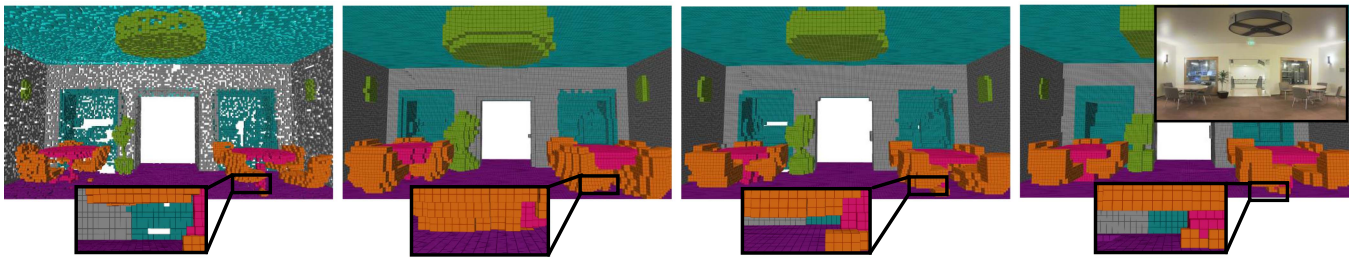

Fig. 8. Semantic mapping results on a lounge of Stanford 2-D–3-D Semantic Dataset.

TABLE I
QUANTITATIVE RESULTS ON FOUR SCENES OF STANFORD 2-D–3-D SEMANTIC DATASET

Scenes	Method	Wall	Chair	Table	Floor	Bookcase	Door	Beam	Clutter	Ceiling	W.-average	Accurateness
Conference room	S-CSM	30.94	19.91	16.54	36.47	21.55	15.98	22.84	24.75	39.84	27.97	88.13
	S-BKI	23.43	46.52	**40.16**	23.71	53.04	**51.92**	47.58	37.55	23.75	34.90	69.85
	SEE-CSOM	**36.68**	**56.40**	36.49	**60.57**	**62.33**	45.89	**48.44**	**51.10**	**64.53**	**52.67**	**90.02**
Lounge	S-CSM	**36.23**	20.82	15.85	33.76	12.68	-	-	21.04	38.43	27.36	80.32
	S-BKI	24.26	51.97	47.14	25.04	**56.32**	-	-	47.78	28.64	37.49	54.95
	SEE-CSOM	35.98	**62.01**	**49.77**	**62.71**	43.81	-	-	**60.07**	**78.59**	**54.24**	**84.78**
Office	S-CSM	34.64	17.46	7.18	**48.80**	15.74	-	27.13	20.51	46.50	23.14	79.72
	S-BKI	24.67	50.20	**34.05**	17.66	50.79	-	38.98	44.18	21.52	35.40	61.39
	SEE-CSOM	**51.54**	**57.20**	22.09	43.52	**58.41**	-	**45.95**	**49.22**	**59.57**	**44.81**	**83.16**
WC	S-CSM	29.45	-	-	29.49	-	16.39	23.69	26.39	25.79	27.58	85.84
	S-BKI	30.73	-	-	29.53	-	51.31	45.91	37.60	33.69	33.55	68.09
	SEE-CSOM	**45.82**	-	-	**55.54**	-	**48.56**	**59.25**	**50.46**	**61.22**	**51.37**	**88.38**

The bold entities indicate the best result of the comparison methods.

TABLE II
NUMERICAL COMPARISON OF ROC CURVES

Method	Area under the curve	TP detecetion rate for FP detection rate of 0.2
AKIMap	0.6224	0.4159
OctoMap	0.6260	0.4149
GPOctoMap	0.7408	0.6588
BGKOctoMap-L	0.7390	0.5865
S-CSM	0.6398	0.4249
S-BKI	0.7538	0.6060
SEE-CSM	**0.8062**	**0.7535**

The bold entities indicate the best result of the comparison methods.

types of outdoor scenes are extracted for evaluation. The map resolution is set to 0.3 m.

Taking Sequence 04 as an example, the comparison of the mapping results is shown in Fig. 9. The enlarged pictures present part of the ground. Due to inaccurate network segmentation, all the generated maps have some random noises. There are many loopholes and messy semantic labels in the S-CSM map. The reason is that S-CSM does not consider the spatial correlation. S-BKI can remove some noises and fill in the loopholes by smoothing, but it is still not comparable with ours. SEE-CSOM almost removes all noises by applying RVFM and AKLM, which is more in line with ground truth.

The quantitative results are summarized in Table III. It is obvious that SEE-CSOM has the best performance. Moreover, the accuracy of S-BKI has been greatly improved, and even surpasses S-CSM. The reason is that continuous mapping is more suitable for cluttered outdoor scenes with many unknown or ambiguous objects. Fig. 10 shows the confusion matrices. The diagonal (TP) of our confusion matrix has the darkest color, with the largest prediction and recall for each class. The highest F1 score also confirms the best spatial semantic classification effect of the proposed algorithm.

D. Efficiency Evaluation

To evaluate efficiency, the average runtime of both semantic and geometric mapping methods is reported in Table IV. For a fair comparison, all experiments use the same environment configuration on both hardware and software. In general, semantic mapping methods cost more time than geometric mapping methods. This is because the semantic map includes multiple classes of object labels, thus increasing the complexity of estimation and update. By adaptively assigning the kernel length, SEE-CSOM has the highest efficiency in semantic mapping

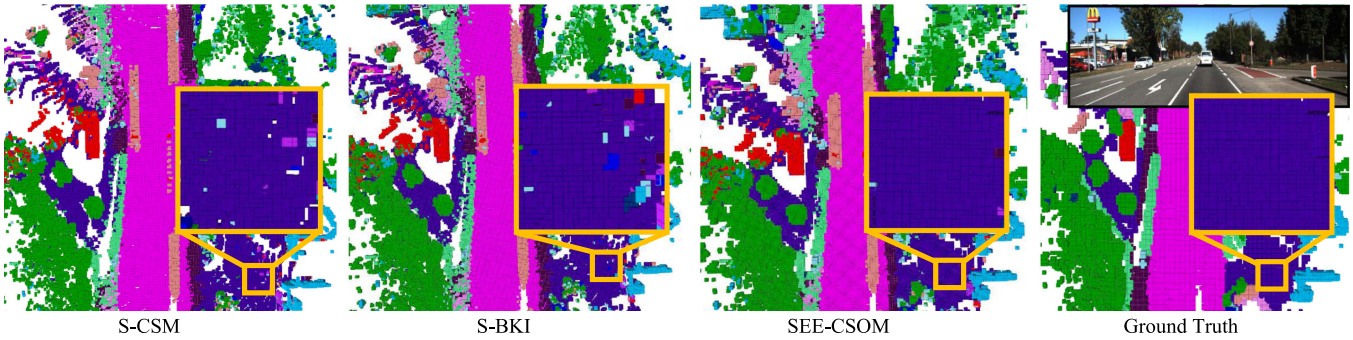

Fig. 9. Semantic mapping results on Sequence 04 of SemanticKITTI dataset.

TABLE III
QUANTITATIVE RESULTS ON FOUR SEQUENCES OF SEMANTICKITTI DATASET

Seq.	Method	Car	Other-vehicle	Person	Road	Parking	Sidewalk	Other-ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Traffic-sign	W-average	Accurateness
02	S-CSM	-	-	-	25.85	-	22.35	-	-	8.16	17.35	8.24	18.01	7.59	7.75	18.41	74.13
	S-BKI	-	-	-	48.78	-	41.17	-	-	12.46	29.52	8.46	34.24	7.27	11.27	32.81	78.33
	SEE-CSOM	-	-	-	**85.60**	-	**69.56**	-	-	**23.26**	**58.39**	**11.34**	**55.20**	**14.51**	**14.57**	**59.71**	**89.06**
04	S-CSM	7.97	18.67	6.82	25.59	7.13	6.41	20.98	14.23	9.70	17.49	5.32	12.78	8.59	9.86	17.29	76.20
	S-BKI	10.42	22.16	9.03	46.79	11.32	15.30	39.37	23.96	13.88	32.96	6.06	25.15	12.76	14.02	31.96	80.20
	SEE-CSOM	**16.91**	**26.60**	**9.17**	**79.79**	**21.60**	**26.39**	**60.92**	**42.77**	**23.84**	**64.67**	**8.85**	**43.22**	**23.42**	**23.67**	**57.36**	**89.92**
06	S-CSM	20.56	7.85	-	16.53	21.18	17.35	-	21.70	4.76	18.53	1.29	25.70	11.55	10.79	20.08	78.93
	S-BKI	30.81	8.16	-	32.29	29.93	28.89	-	29.08	6.63	27.67	1.42	42.57	15.96	12.54	31.92	81.73
	SEE-CSOM	**50.49**	**9.53**	-	**51.60**	**39.53**	**46.10**	-	**42.73**	**11.55**	**45.48**	**1.55**	**64.55**	**26.13**	**22.52**	**49.71**	**88.45**
08	S-CSM	9.04	-	0.00	23.53	2.17	17.25	-	13.86	6.23	20.35	6.75	17.96	6.73	5.47	18.49	82.86
	S-BKI	12.26	-	**0.74**	43.97	4.57	32.21	-	20.18	9.57	34.87	7.09	34.78	7.84	7.88	33.35	85.40
	SEE-CSOM	**18.59**	-	0.00	**72.34**	**6.71**	**51.22**	-	**31.14**	**11.53**	**64.72**	**9.18**	**63.27**	**14.66**	**13.65**	**58.89**	**92.13**

The bold entities indicate the best result of the comparison methods.

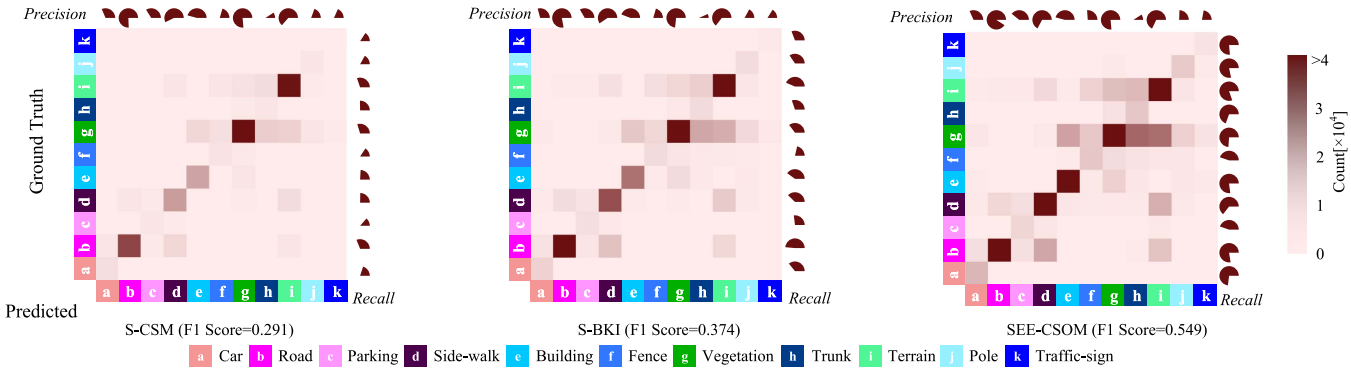

Fig. 10. Confusion matrix and F1 score of SemanticKITTI dataset. The brown sectors approximate the value of the prediction or recall.

methods and even slightly surpasses OctoMap in some scenes. Considering the importance of semantic information, the time cost of SEE-CSOM is acceptable.

E. Impact of Parameters

The sensitivity of SEE-CSOM to two important parameters is studied: block depth $\mathbf{D}_b$ and context entropy threshold $\mathbf{T}_{con}$. The experiments are conducted on the conference room dataset. In Fig. 11, w-average IoU and accurateness are utilized as evaluation indicators. The best mapping effect is achieved when the block depth $\mathbf{D}_b$ is set to 2. When it is smaller, the surrounding observation data are considered insufficient, and when it is larger, the details of the map will be ignored, both of which are not conducive to the accurate reconstruction of the scene. 0.1 is the best candidate for context entropy threshold $\mathbf{T}_{con}$, which filters out voxels that have no interior observations and have exterior observations in few directions.

TABLE IV
AVERAGE RUNTIME ON EIGHT SCENES (SECOND/SCAN)

	Method	Stanford				SemanticKITTI			
		Con.	Lou.	Off.	WC	02	04	06	08
W/O Sem.	OctoMap	0.188	0.241	0.151	0.198	1.205	1.246	1.133	1.088
	GPOctoMap	0.080	0.128	0.064	0.093	0.690	0.488	0.490	0.523
	BGKOctoMap-L	**0.079**	**0.126**	**0.063**	0.093	0.699	0.477	0.460	0.512
	AKIMap	**0.079**	**0.126**	**0.063**	**0.090**	**0.486**	**0.406**	**0.388**	**0.403**
W/ Sem.	S-CSM	0.236	0.277	0.242	0.227	4.320	4.803	3.750	5.831
	S-BKI	0.241	0.285	0.240	0.223	4.311	4.290	3.917	6.981
	SEE-CSOM	**0.117**	**0.169**	**0.118**	**1.161**	**1.640**	**1.814**	**1.242**	**1.454**

The bold entities indicate the best result of the comparison methods.

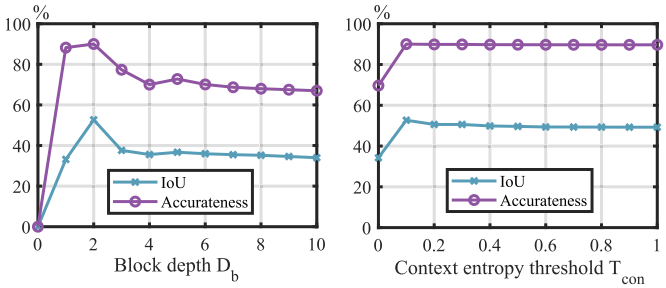

Fig. 11. Impact of parameters on mapping performance of SEE-CSOM.

F. Validation in the Real World

To verify the practicality in real applications, SEE-CSOM is tested by deploying a mobile robot. The robot is equipped with a 3-D Velodyne LiDAR and a visual camera, where the sensors have been accurately calibrated with [25]. The Cityscapes dataset [31] is used to train the semantic segmentation model.

The robot is teleoperated to traverse the campus to record raw sensor data, from which semantic point clouds and robot trajectories are generated. To verify the performance of the algorithms on sparse data, each scan of the point cloud is downsampled to a resolution of 0.2 m, the maximum perception range is 15 m, and the map resolution is set to 0.1 m. Using the exact same input and real-time playback, the qualitative results of the three semantic mapping algorithms are shown in Fig. 12. A top view of the SEE-CSOM map and an image of the robot are shown at the top. At the bottom is a comparison of the maps constructed by the three algorithms. By visually checking the generated map, it is found that the SEE-CSOM map strikes a balance between the sparse S-CSM map and the overinflated S-BKI map, filling the mapping loopholes without overfitting.

In addition, the numerical quantitative results are also reported in Table V. Since the ground-truth semantic labels are not available, the ground-truth geometric map is constructed offline using dense denoised point clouds. As indicated in the table, SEE-CSOM outperforms other algorithms on all metrics. S-CSM has the smallest occupied IoU due to its conservative estimation of occupancy, while S-BKI, on the opposite, is overinflated. In terms of runtime, SEE-SCOM achieves the best computational efficiency. Based on the real-world validation,

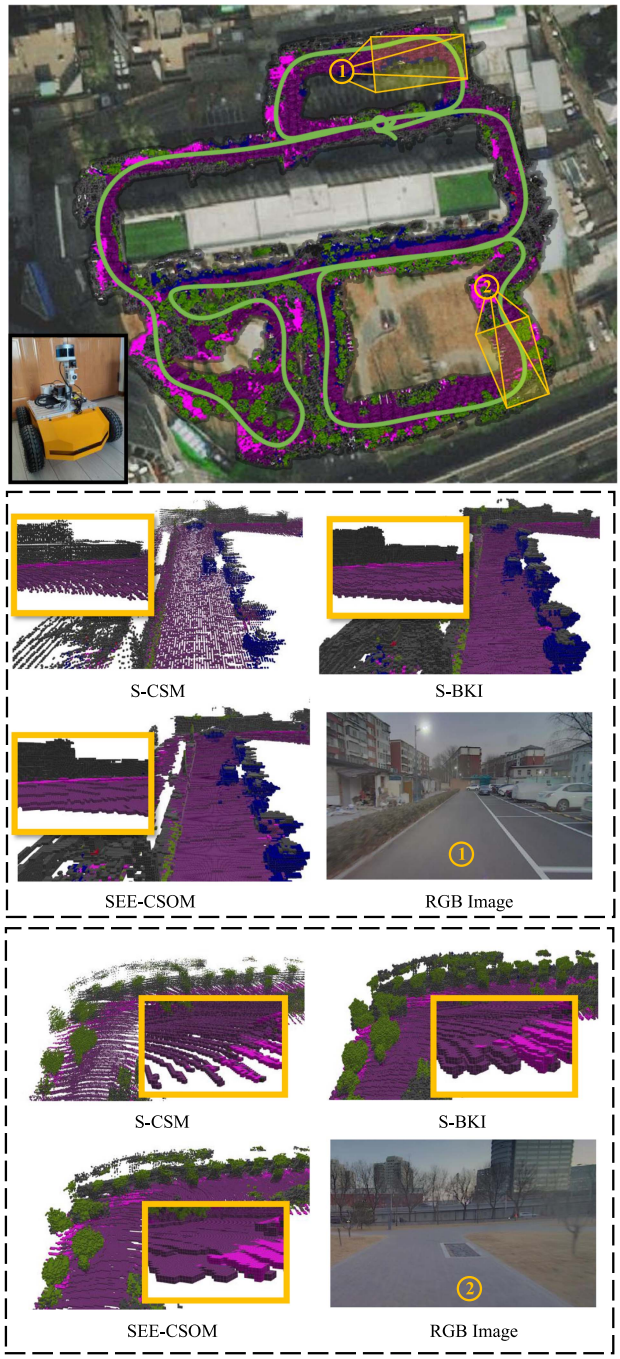

Fig. 12. Qualitative results of semantic mapping of large-scale real scenes. From top to bottom: The overall semantic occupancy map of SEE-CSOM, a comparison of three maps in two different regions.

TABLE V
QUANTITATIVE RESULTS IN THE REAL WORLD

	F1 Score	Acc	Free IoU	Occupied IoU	Runtime (Second/Scan)
S-CSM	0.3213	0.7757	0.7631	0.1915	3.8735
S-BKI	0.5907	0.6400	0.5125	0.4192	3.2594
SEE-CSOM	**0.7686**	**0.8683**	**0.8309**	**0.6246**	**0.8804**

The bold entities indicate the best result of the comparison methods.

SEE-CSOM demonstrates its accuracy and efficiency in practical applications.

V. Conclusion

This article established a sharp-edged and efficient continuous semantic occupancy mapping algorithm. More specifically, the proposed redundant voxel filter model filtered out redundant voxels, therefore the representation of objects had accurate boundaries with sharp edges in our map. In addition, the proposed adaptive kernel length model adjusted kernel length adaptively, which greatly reduced the computational complexity. The multientropy kernel function integrated the two models to jointly reconstruct a dense accurate representation from sparse noisy sensor observations. The results demonstrated that the proposed algorithm achieved high accuracy and efficiency. In the future, the plan is to consider semantic consistency when filtering redundant voxels, so that a more accurate and smoother semantic map can be reconstructed.

References

[1] Y. Yue and D. Wang, *Collaborative Perception, Localization and Mapping for Autonomous Systems*, vol. 2. Berlin, Germany: Springer Nature, 2020.

[2] Y. Yue, C. Zhao, Z. Wu, C. Yang, Y. Wang, and D. Wang, "Collaborative semantic understanding and mapping framework for autonomous systems," *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 2, pp. 978–989, Apr. 2021.

[3] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[4] T. Shan, J. Wang, B. Englot, and K. Doherty, "Bayesian generalized kernel inference for terrain traversability mapping," in *Proc. Conf. Robot Learn.*, 2018, pp. 829–838.

[5] K. Doherty, T. Shan, J. Wang, and B. Englot, "Learning-aided 3-D occupancy mapping with Bayesian generalized kernel inference," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 953–966, Aug. 2019.

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

[7] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2D-3D-semantic data for indoor scene understanding," 2017, *arXiv:1702.01105*.

[8] S. T. O. Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *Int. J. Robot. Res.*, vol. 31, no. 1, pp. 42–62, 2012.

[9] J. Wang and B. Englot, "Fast, accurate Gaussian process occupancy maps via test-data octrees and nested Bayesian fusion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1003–1010.

[10] M. G. Jadidi, L. Gan, S. A. Parkison, J. Li, and R. M. Eustice, "Gaussian processes semantic map representation," 2017, *arXiv:1707.01532*.

[11] Y. Deng, M. Wang, D. Wang, and Y. Yue, "S-MKI: Incremental dense semantic occupancy reconstruction through multi-entropy kernel inference," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 3824–3829.

[12] J. Stückler, N. Biresev, and S. Behnke, "Semantic mapping using object-class segmentation of RGB-D images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 3005–3010.

[13] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. Torr, "Urban 3D semantic modelling using stereo vision," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 580–585.

[14] B.-S. Kim, P. Kohli, and S. Savarese, "3D scene understanding by voxel-CRF," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1425–1432.

[15] Z. Zhao and X. Chen, "Building 3D semantic maps for mobile robots using RGB-D camera," *Intell. Serv. Robot.*, vol. 9, no. 4, pp. 297–309, 2016.

[16] S. Sengupta and P. Sturgess, "Semantic octree: Unifying recognition, reconstruction and representation via an octree constrained higher order MRF," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 1874–1879.

[17] S. Yang, Y. Huang, and S. Scherer, "Semantic 3D occupancy mapping through efficient high order CRFs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 590–597.

[18] Y. Yue et al., "A hierarchical framework for collaborative probabilistic semantic mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 9659–9665.

[19] Y. Yue, M. Wen, C. Zhao, Y. Wang, and D. Wang, "Cosem: Collaborative semantic map matching framework for autonomous robots," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3843–3853, Apr. 2022.

[20] Q. Zhang, M. Wang, Y. Yue, and T. Liu, "LCR-SMM: Large convergence region semantic map matching through expectation maximization," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 5, pp. 3029–3040, Oct. 2022.

[21] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1717–1730, 2016.

[22] K. Doherty, J. Wang, and B. Englot, "Probabilistic map fusion for fast, incremental occupancy mapping with 3D hilbert maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1011–1018.

[23] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[24] J. Zhang, P. Siritanawan, Y. Yue, C. Yang, M. Wen, and D. Wang, "A two-step method for extrinsic calibration between a sparse 3D LiDAR and a thermal camera," in *Proc. IEEE 15th Int. Conf. Control, Autom., Robot. Vis.*, 2018, pp. 1039–1044.

[25] C. Guindel, J. Beltrán, D. Martín, and F. García, "Automatic extrinsic calibration for LiDAR-stereo vehicle sensor setups," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–6.

[26] D. Hähnel, "Mapping with mobile robots." Ph.D. dissertation, Univ., Freiburg, Freiburg im Breisgau, Germany, 2005.

[27] A. Melkumyan and F. T. Ramos, "A sparse covariance function for exact Gaussian process inference in large datasets," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1936–1942.

[28] Y. Kwon, B. Moon, and S.-E. Yoon, "Adaptive kernel inference for dense and sharp occupancy grids," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 4712–4719.

[29] J. Behley et al., "Semantickitti: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9297–9307.

[30] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet: Fast and accurate LiDAR semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4213–4220.

[31] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.

Yinan Deng received the B.S. degree in automation in 2021 from the Beijing Institute of Technology, Beijing, China, where he is currently working toward the M.S. degree for the first year in intelligent navigation with the School of Automation.

His research interests include semantic 3-D reconstruction and collaborative mapping of robotic systems in unknown environments.

Meiling Wang received the B.S. and M.S. degrees in automation and Ph.D. degree in navigation, guidance, and control from the Beijing Institute of Technology, Beijing, China, in 1992, 1995, and 2007, respectively.

She was with the University of California San Diego as a Visiting Scholar in 2004. She is currently a Professor and the Director of Integrated Navigation and Intelligent Navigation Lab. Her research interests include advanced technology of sensing and vehicle intelligent navigation.

Dr. Wang was elected as the Yangtze River Scholar Distinguished Professor in 2014.

Yi Yang received the Ph.D. degree in automation from the Beijing Institute of Technology, Beijing, China, in 2010.

He is currently a Professor with the School of Automation, Beijing Institute of Technology. His research interests include autonomous vehicles, bioinspired robots, intelligent navigation, semantic mapping, scene understanding, motion planning and control. He is the author or coauthor of more than 50 conference and journal papers in the area of unmanned ground vehicles.

Danwei Wang (Life Fellow, IEEE) received the B.E. degree from the South China University of Technology, China, and M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 1982, 1984, and 1989, respectively, all in robotics.

He is a Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. His research interests include robotics, control engineering, and fault diagnosis. He is a Fellow of The Academy of Engineering Singapore.

Yufeng Yue (Member, IEEE) received the B.Eng. degree in automation from the Beijing Institute of Technology, Beijing, China, in 2014, the Ph.D. degree in robotics from Nanyang Technological University, Singapore, in 2019.

He is currently a Professor with the School of Automation, Beijing Institute of Technology. He has published a book in Springer, and over 40 journal/conference papers, including TMECH, TIE, TCST, TMM, ICRA and IROS. His research interests include perception, mapping and navigation for collaborative robots.