Adaptive Kernel Inference for Dense and Sharp Occupancy Grids

Youngsun Kwon[1], Bochang Moon[2] and Sung-Eui Yoon[1]

***Abstract*— In this paper, we present a new approach, AKIMap, that uses an adaptive kernel inference for dense and sharp occupancy grid representations. Our approach is based on the multivariate kernel estimation, and we propose a simple, two-stage based method that selects an adaptive bandwidth matrix for an efficient and accurate occupancy estimation. To utilize correlations of occupancy observations given sparse and non-uniform distributions of point samples, we propose to use the covariance matrix as an initial bandwidth matrix, and then optimize the bandwidth matrix by adjusting its scale in an efficient, data-driven way for on-the-fly mapping. We demonstrate that the proposed technique estimates occupancy states more accurately than state-of-the-art methods given equal-data or equal-time settings, thanks to our adaptive inference. Furthermore, we show the practical benefits of the proposed work in on-the-fly mapping and observe that our adaptive approach shows the dense as well as sharp occupancy representations in a real environment.**

I. Introduction

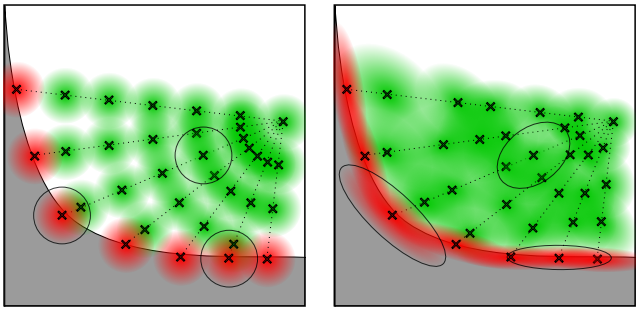
Various robotic applications, such as autonomous navigation, require a high-quality understanding of the surrounding environment for fulfilling their tasks. Well-known sensors (e.g., lidar) are often exploited as a means to estimate the surroundings, and these produce a noisy point cloud that represents a partial geometry.

As a notable example, occupancy maps have been widely used as a functional approximation of the geometry in the environment, and the maps are constructed from the partial data corrupted by noise. For example, the occupancy grids [1], [2], and tree-based structures [3] discretized the environment with multiple cells, each of which contains an occupancy probability. This simple representation allows updating occupancy states of the discretized environment in real-time [4], but the resolution (i.e., the number of cells) of such grid-based maps needs to be high in order to decrease its discretization error. In this case, sparse point clouds can lead to problematic holes in the resulting maps, where some cells do not contain any occupancy observations.

Learning-based approaches [5], [6] estimated the maps while addressing the sparsity issue of the observed points. Their high-level idea is to predict occupancy states in unmeasured regions by exploiting a spatial correlation among point samples. However, these techniques suffer from a high computation overhead compared to the grid-based approaches.

[1]Youngsun Kwon and [1]Sung-Eui Yoon are with School of Computing, Korea Advanced Institute of Science and Technology, Daejeon, South Korea. youngsun.kwon@kaist.ac.kr, sungeui@kaist.edu

[2]Bochang Moon is with School of Integrated Technology, Gwangju Institute of Science and Technology, Gwangju, South Korea. bmoon@gist.ac.kr



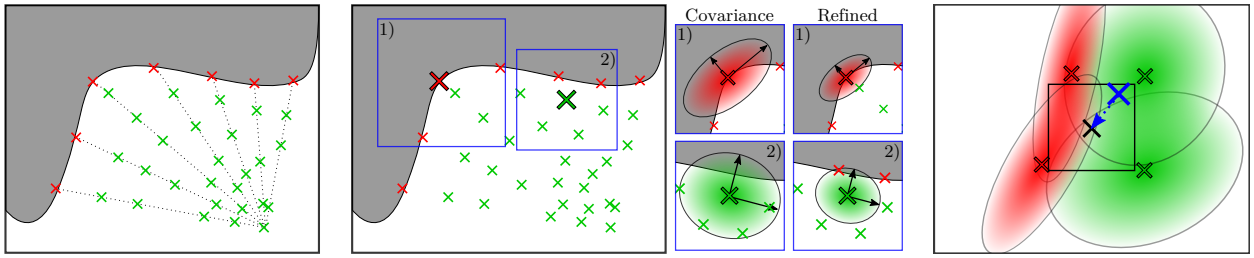
(a) Fixed kernel inference (b) Adaptive kernel inference

Fig. 1. These figures show (a) the fixed kernel inference using isotropic estimation and (b) the adaptive kernel inference on the occupancy samples. Outlines of the circles or ellipses represent each support region of kernel estimation at the sample. We denote the occupied and free states by the red and green colors, respectively, and the object is colored by the gray.

Recently, Doherty et al. [7] proposed a Bayesian nonparametric kernel inference model, which allows for estimating a dense occupancy map with a reduced computational cost. Nonetheless, we found that sharing the same isotropic bandwidth across the kernel can cause a false occupancy representation or unsupported region, especially when the distribution of given data is sparse and highly nonuniform.

Main contributions. In this paper, we propose a new adaptive kernel inference model, AKIMap, which reconstructs a sharp, but dense occupancy grid, while handling the nonuniformly distributed sparse data. As our main technical contribution, we leverage an anisotropic kernel in robust and efficient kernel inference model of occupancy mapping. In our approach, the kernel shape varies locally so that occupancy boundaries can be estimated appropriately. Specifically, our adaptive method optimizes the kernel shapes according to the distribution of occupancy observations. In addition, we present a two-stage optimization performed per kernel locally for the real-time occupancy mapping.

We first demonstrate the robust occupancy estimation of our approach using two synthetic datasets. We compare our adaptive approach with the state-of-the-art techniques [7], [8] that employ an isotropic inference. Given the equal amount of sensor data as well as the equal time budget, our approach shows outstanding performance in the occupancy estimation over the tested methods. This result is mainly achieved by our simple, yet effective anisotropic kernel inference. We also test our method in a real environment. For on-the-fly mapping, our method shows an incremental reconstruction of the dense as well as sharp occupancy representations from noisy point clouds.



(a) Occupancy sampling (b) Adaptive bandwidth selection (c) Kernel estimation update

Fig. 2. These figures show an overview of our framework. (a) represents occupancy samples gathered from the sparse sensor data. In (b), each blue box represents a search region for finding neighbor samples of each kernel center. As an initial bandwidth matrix, our method computes a covariance matrix from the neighbor samples having the same occupancy state to the kernel center. We then refine the bandwidth matrix adaptive to the local distribution of positive and negative neighbor samples. (c) Finally, our model incrementally estimates and accumulates its information at cell centers. Once a query point (blue X mark) is given, the cell containing the point is used for final estimation.

II. Related Work

Machine-learning based approaches have been studied to reconstruct a dense occupancy map from noisy, sparse sensor data. For example, O'Callaghan and Ramos [5] proposed a non-parametric Gaussian process for the occupancy mapping problem. Ramos and Ott [6] designed a classification model, Hilbert-map, that employs a logistic classifier with stochastic gradient descent. While these models showed the dense occupancy representation of the environment thanks to learning an implicit correlation of occupancy observations, a high computation is required for on-the-fly mapping scenarios.

The learning-based approaches have been optimized with respect to computational overhead while preserving their estimation accuracy. For example, non-parametric regression models such as Gaussian process provide the occupancy estimation with predictive variance resulting in the accurate representation of the map. However, the well-known drawback of the Gaussian process is the cubic of time complexity in the number of data. To overcome this problem, some variations of this regression model have been proposed. For example, Kim and Kim [9] trained sparse Gaussian process regressors using partitioned training subsets. Wang and Englot [10] exploited a parallel computing when occupancy predictions are merged from local Gaussian process models. These works achieved a substantial performance gain compared to the Vanilla Gaussian Process model [5], but the heavy time complexity of the regressor remained.

Parametric regression models based on Hilbert-map have been proposed to overcome the computational overhead of Gaussian process models. Doherty et al. [11] presented an incremental mapping that fuses overlapping scan data using Hilbert-maps. Given the dimension of a feature space of Hilbert-map, Guizilini and Ramos [12], and Senanayake et al. [13] proposed methods to estimate parameters constructing the feature space from sensor measurements. Zhi et al. [14] improved the processing speed of a Bayesian Hilbert-map [15] by fusing local models into a global model. These parametric models need to carefully determine the number of feature space dimensions for a trade-off between representation accuracy and computational efficiency.

Recently, various approaches based on different models have been studied for improving the performances of occupancy mapping. O'Meadhra et al. [16] used parametric Gaussian Mixture Model (GMM) that learns the distribution of sensor data. The work showed robust reconstruction of occupancy grid through ray-casting from the sensor origin to points resampled from the mixture model. Doherty et al. [7], [8] proposed a Bayesian non-parametric kernel inference model for updating kernel estimation results in the occupancy grid incrementally. The work showed real-time performance using an isotropic kernel that shares a bandwidth matrix globally.

In this work, we propose a technique that enables an adaptive kernel inference, where the bandwidth matrix of each kernel varies locally to estimate occupancy robustly while maintaining computational efficiency. Compared to the parametric methods [12], [13], [16] using global data distribution, our non-parametric model exploits local distributions of each occupancy observation directly to varying kernel signals for occupancy predictions.

III. Multivariate Kernel Estimator

Our goal is to predict occupancy states at the unmeasured regions from sparse sensor data on-the-fly. To estimate a high-quality occupancy map, we adapt the well-established multivariate kernel estimator so that its kernels can be optimized in a data-driven way. The mathematical framework of the kernel estimator is discussed in this section, followed by our novel adaptation for occupancy mapping in Sec. IV.

A. Multivariate Kernel Estimator

Let $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$ be a d-dimensional input point and its response, respectively, which is an observation pair in a multivariate system, i.e., $d > 1$. Given the observations, the Nadaraya-Watson estimator [17], [18], one of the well-known kernel estimators, predicts an output, y_q, at a query point, $\mathbf{x}_q$, as the conditional expectation:

$$E[y_q|\mathbf{x}_q] = \frac{\sum |\mathbf{H}_i|^{-\frac{1}{2}} k(\mathbf{x}_q, \mathbf{x}_i, \mathbf{H}_i) y_i}{\sum |\mathbf{H}_i|^{-\frac{1}{2}} k(\mathbf{x}_q, \mathbf{x}_i, \mathbf{H}_i)}, \tag{1}$$

where $k(\cdot)$ is a kernel function that defines the weight of the response y_i from the nearby input sample $\mathbf{x}_i$ to the query point $\mathbf{x}_q$. The bandwidth matrix $\mathbf{H}_i$, which is a positive definite matrix, controls shape of the kernel function.

In the 3-D occupancy mapping, let $\mathbf{x} \in \mathbb{R}^3$ be a point in 3-D space, and $y \in \{0, 1\}$ be an occupancy state, where 0 and 1 represent a free and an occupied state, respectively. We define an occupancy sample, $\{\mathbf{x}_i, y_i\}$, as an i-th observation pair with the occupancy state in our algorithm; we denote such occupancy samples as $occupied\ sample$ or $free\ sample$ depending on their observations of occupancy states. Each sample is associated with a kernel bandwidth, $\mathbf{H}_i$, adaptive to the distribution of occupancy samples, as shown in Fig. 1.

B. Motivation

It is well-known that in the multivariate case, the accuracy of kernel inference highly depends on the bandwidth matrix $\mathbf{H}_i$ [19], as shown in Fig. 1. Unfortunately, the crucial parameter, bandwidth matrix, was not fully optimized for the reconstruction of the occupancy map. For example, the recent kernel inference method [7] used an identity matrix with a fixed scale for the bandwidth matrix. This isotropic kernel estimation is simple, but its estimation quality can be degraded when the map is reconstructed from samples on non-uniformly distribution (e.g., walls). Fig. 1-(a) illustrates this scenario. The fixed kernel inference produces dense estimation results nearby the sensor origin, but fails to estimate some occupied parts along the object's surface or does not preserve the boundary between occupied and unoccupied parts. We observe that this limitation is mainly caused by the isotropic kernel estimation, which does not take account for the non-uniform distribution of the occupancy samples adequately.

To mitigate this problem, we optimize the bandwidth matrix $\mathbf{H}_i$ at each sample according to the distribution of the samples so that the occupancy state can be estimated robustly given a non-uniformly distributed sample set. As shown in Fig. 1-(b), our approach adapts the size and orientation of the anisotropic kernels along the object surface by changing the matrix $\mathbf{H}_i$, and thus preserves the sharp occupancy boundary well, compared to the isotropic approach.

IV. Adaptive Kernel Inference for Grid-based Occupancy Map

A. Overview

To reconstruct a high-quality occupancy grid with dense, yet sharp properties from sparse data points with noise, we utilize the multivariate kernel estimation (Sec. III) in a data-driven way for our problem with the occupancy grid.

We first introduce procedures of our mapping framework, AKIMap, as depicted in Fig. 2 (Sec. IV-B). We describe a simple policy to extract observations about occupancy states from the sparse data as inputs to our inference model. Based on the distribution of the occupancy observations, our method optimizes the bandwidth matrix $\mathbf{H}_i$ so that the occupancy states can be estimated robustly using our anisotropic kernel estimation (Sec. IV-C). For efficient runtime updates of the occupancy grid, our adaptive kernel inference estimates and accumulates the occupancy states in cell centers of the occupancy grid incrementally (Sec. IV-D). Once a query point is given, we identify a cell containing the point and utilize its estimation information.

Note that we depict the 2-D examples in the figures for the simple description, but our method operates in the 3-D environment. Main notations are summarized in Table I.

TABLE I
Summary of Notations

$\mathbf{x} \in \mathbb{R}^3$	Observation point
$y \in \{0, 1\}$	Occupancy label; free state: 0, occupied state: 1
$\{\mathbf{x}_i, y_i\}$	i-th occupancy sample
$\mathbf{c}_m \in \mathbb{R}^3$	Center point of m-th cell
$\mathbf{H}_i$	Kernel bandwidth matrix of $\{\mathbf{x}_i, y_i\}$
$\mathbf{\Sigma}_i$	Covariance matrix of positive samples of $\mathbf{x}_i$
s_i	Bandwidth scale; $\mathbf{H}_i = s_i \mathbf{\Sigma}_i$
$k(\cdot)$	Kernel function

B. Kernel Inference for Occupancy Grid

The kernel inference, Eq. 1, requires the occupancy samples and their bandwidth matrices to estimate an occupancy state at a query point. However, it is intractable to store all of the data to the map during the on-the-fly mapping. Concerning this issue, a recent work [7] proposed a mapping framework that incrementally updates the kernel estimations of cells in the occupancy grid.

We extend the kernel inference framework with a fixed, isotropic bandwidth into the inference using our adaptive bandwidth matrices. Given the Bernoulli likelihood $p(y_i|\theta_m)$ of occupancy probability θ_m and its conjugate Beta prior $Beta(\alpha_0, \beta_0)$, the posterior at m-th cell of the framework follows $Beta(\alpha_m, \beta_m)$ expressed as:

$$\alpha_m = \alpha_0 + \sum |\mathbf{H}_i|^{-\frac{1}{2}} k(\mathbf{c}_m, \mathbf{x}_i, \mathbf{H}_i) y_i, \tag{2}$$

$$\beta_m = \beta_0 + \sum |\mathbf{H}_i|^{-\frac{1}{2}} k(\mathbf{c}_m, \mathbf{x}_i, \mathbf{H}_i)(1 - y_i), \tag{3}$$

where α_0 and β_0 originate in the Beta prior. Two parameters of this posterior, α_m and β_m, represent the accumulated kernel estimations from occupied and free samples, respectively. The maximum a posterior (MAP) at the cell then becomes:

$$\theta_m^{MAP} = \frac{\alpha_m}{\alpha_m + \beta_m} \approx \frac{\sum |\mathbf{H}_i|^{-\frac{1}{2}} k(\mathbf{c}_m, \mathbf{x}_i, \mathbf{H}_i) y_i}{\sum |\mathbf{H}_i|^{-\frac{1}{2}} k(\mathbf{c}_m, \mathbf{x}_i, \mathbf{H}_i)}, \tag{4}$$

where $\mathbf{c}_m$ is a center point of the cell. Note that the MAP, Eq. 4, approximates the multivariate kernel estimation, Eq. 1. Furthermore, the kernel estimations at the cell enable the efficient incremental updates and occupancy queries that we discuss in Sec. IV-D.

Kernel Function. For our mapping problem, we define the kernel function as a bounded kernel [20]:

$$k(\mathbf{c}_m, \mathbf{x}_i, \mathbf{H}_i) = \begin{cases} \frac{2+\cos(2\pi r)}{3}(1-r) + \frac{1}{2\pi}\sin(2\pi r) & \text{if } r < 1, \\ 0 & \text{if } r \geq 1, \end{cases} \quad (5)$$

where $r = \sqrt{(\mathbf{c}_m - \mathbf{x}_i)^T \mathbf{H}_i^{-1} (\mathbf{c}_m - \mathbf{x}_i)}$.

Occupancy Sampling. Our approach extracts the occupancy samples $\{\mathbf{x}_i, y_i\}_{i=1:N}$ from sparse sensor measurements captured by sensors. A point cloud data is simply observations about the occupied state at the points sampled from the surrounding objects. Therefore, we use the sensor data directly as the occupied samples. On the other hand, we can observe the free space on a sensor ray traversing from the sensor origin to each hit point of the point cloud. Specifically, our method randomly selects one free sample per sampling distance, e.g. 0.5 m, on sensor rays, similar to ones used in the prior approaches [6], [21]. The random free samples on the rays prevent that our adaptive shapes of kernels are overfitted to the sampling patterns, instead of the distribution of free space, occurring in a uniform sampling.

C. Adaptive Bandwidth Selection

In this section, we propose an adaptive technique that varies the kernel bandwidth matrix $\mathbf{H}_i$ of each occupancy sample for computing the occupancy map robustly. Our bandwidth selection reflects the following high-level observations to the adaptive bandwidth. 1) If the kernel center and its neighbor sample have the same occupancy state, we could observe another sample sharing the state in the space between them. 2) Otherwise, we could have a low chance of observing a new sample having the state of the kernel center.

A bandwidth matrix in a 3-D environment can be composed of six-parameters: three for a rotation matrix and the others for a length scale of each rotated basis. Ideally, one can opt to optimize all the parameters of the kernel bandwidth. We, however, observed that optimizing all the parameters simultaneously can be unstable for our case where only small numbers of occupancy samples are given within a local region.

We, therefore, propose an efficient, yet robust way to optimize the bandwidth matrix $\mathbf{H}_i$ with a reduced number of parameters. Our method utilizes a covariance matrix $\mathbf{\Sigma}_i$ so that the kernel can be adapted per sample according to its local distribution of nearby samples. Using the covariance as an initial guess, our bandwidth selector then finds the best scalar scale $\hat{s}_i$ to compose the bandwidth $\mathbf{H}_i = \hat{s}_i \mathbf{\Sigma}_i$.

In the first step, our method computes the covariance matrix $\mathbf{\Sigma}_i$ of an occupancy sample $\{\mathbf{x}_i, y_i\}$. We retrieve a set of the neighbor occupancy samples $\{\mathbf{x}_j, y_j\}_{j=1:M}$ of the kernel center $\mathbf{x}_i$ within a search box, i.e., the one colored by blue in Fig. 2-(b). Our approach computes the covariance matrix only from the positive neighbors, i.e., the neighbor samples having the same occupancy states with the occupancy sample $\{\mathbf{x}_i, y_i\}$ under the kernel estimation. Such covariance leads to a scaled and biased shape of the anisotropic kernel toward its positive neighbors. For example, the covariance of occupied samples makes the kernel estimation follows the local surface, which results in the sharp occupancy boundary (Fig. 2-(b)).

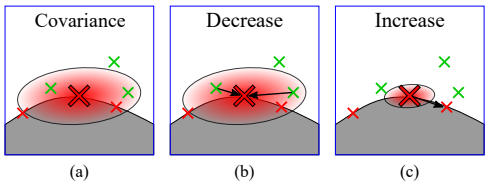


Fig. 3. These figures give an intuitive illustration of our adjustment process of kernel shapes with various neighbor samples. (a) represents the initial kernel shape using the covariance matrix. The negative neighbors in (b) decrease the scale, while the positive neighbor in (c) increases it. We find the best scale given the kernel shape of the covariance matrix by minimizing the estimation errors, Eq. 7.

While it is very efficient, a kernel estimation only using the covariance can be incomplete. For an example shown in Fig. 2-(b), there could be false estimation due to negative samples within the kernel support, once we do not consider them for computing the covariance matrix. Therefore, our approach refines the bandwidth matrix by re-scaling the covariance matrix to reduce such errors by considering such negative samples, while preserving the kernel shape.

At a high-level, our method finds a bandwidth scale s_i to remove the error caused by the false estimation, while keeping the kernel estimation towards the positive samples (Eq. 7). To achieve our goal, we design an estimation error of the kernel at a neighbor sample point, and then minimize it by an optimization technique (e.g. gradient descent). The false estimation at a negative neighbor, if existing, is caused by a kernel signal even reaching the negative neighbor, as shown in Fig. 3-(b). In our bandwidth optimization process, the negative neighbor shrinks the estimation range by pushing the kernel into its center. In contrast to the negative case, suppose a positive neighbor out of the kernel support, which causes no signal at the positive neighbor. Such a positive neighbor in our bandwidth refinement prevents the kernel estimation from having too small range, shown in Fig. 3-(c).

Specifically, we express a target estimation signal $t(\cdot)$ for computing the estimation error at the neighbor sample $\{\mathbf{x}_j, y_j\}$:

$$t(\cdot) \equiv \begin{cases} k(\mathbf{x}_j, \mathbf{x}_i, \mathbf{\Sigma}_i) & \text{if } y_i = y_j, \\ 0 & \text{if } y_i \neq y_j. \end{cases} \quad (6)$$

We set its target signal to be minimum signal value, i.e., 0, where an estimation error at the negative neighbor makes a decrease in the scale of kernel estimation. On the other hand, we set its target estimation signal at the positive neighbor to keep the estimation using the covariance matrix.

Given our definition of the target estimation signal and the estimation errors, our bandwidth refinement method finds the best bandwidth scale $\hat{s}_i$ based on M neighbor samples within the search region:

$$\hat{s}_i = \underset{s_i}{\operatorname{argmin}} \frac{|s_i \mathbf{\Sigma}_i|^{-\frac{1}{2}}}{2} \sum_{j=1}^{M} \left(t(\cdot) - k(\mathbf{x}_j, \mathbf{x}_i, s_i \mathbf{\Sigma}_i)\right)^2, \quad (7)$$

TABLE II

TOP ROWS OF EACH DATASET SHOW THE EQUAL-TIME COMPARISON W/ VARYING USAGES OF THE DATA, AND BOTTOM ROWS SHOW PERFORMANCE AS WE USE ALL THE DATA. BOLD NUMBERS REPRESENT THE BEST PERFORMANCES IN EACH CATEGORY OF COMPARISONS.

Structured dataset					
Name (Usage)	AUC	MSE	Time [sec]	Accuracy (occupied)	Accuracy (free)
AKIMap (75%)	**0.912**	**0.063**	2.76	**0.678**	**0.981**
BGKOctoMap-L (100%)	0.892	0.109	2.82	0.476	**0.981**
BGKOctoMap (90%)	0.875	0.120	2.81	0.377	0.979
AKIMap (100%)	**0.937**	**0.060**	3.99	**0.678**	**0.983**
BGKOctoMap-L (100%)	0.892	0.109	**2.82**	0.476	0.981
BGKOctoMap (100%)	0.881	0.119	3.06	0.378	0.979
Unstructured dataset					
Name (Usage)	AUC	MSE	Time [sec]	Accuracy (occupied)	Accuracy (free)
AKIMap (90%)	**0.852**	**0.102**	2.58	**0.664**	**0.965**
BGKOctoMap-L (100%)	0.831	0.133	2.54	0.502	**0.965**
BGKOctoMap (95%)	0.816	0.145	2.52	0.391	0.962
AKIMap (100%)	**0.855**	**0.101**	2.86	**0.665**	**0.966**
BGKOctoMap-L (100%)	0.831	0.133	**2.54**	0.502	0.965
BGKOctoMap (100%)	0.817	0.145	2.61	0.391	0.962

where $|s_i\mathbf{\Sigma}_i|^{-\frac{1}{2}}$ is the signal weight of kernel estimation in Eq. 4. Based on the computed scale, we select the refined kernel bandwidth $\mathbf{H}_i = \hat{s}_i\mathbf{\Sigma}_i$ for the i-th sample.

D. Estimation Update on Occupancy Grid

Our framework aims to support the incremental updates of the occupancy map during on-the-fly mapping. The m-th cell of our map holds two kinds of values, α_m (Eq. 2) and β_m (Eq. 3). Given the N occupancy samples with their bandwidth matrices at a time step t, these equations can be re-formulated to the update rule that accumulates the adaptive kernel estimations during the time steps from 1 to t:

$$\alpha_m^{1:t} = \alpha_m^{1:t-1} + \sum_{i=1}^{N} |\mathbf{H}_i|^{-\frac{1}{2}} k(\mathbf{c}_m, \mathbf{x}_i, \mathbf{H}_i) y_i, \tag{8}$$

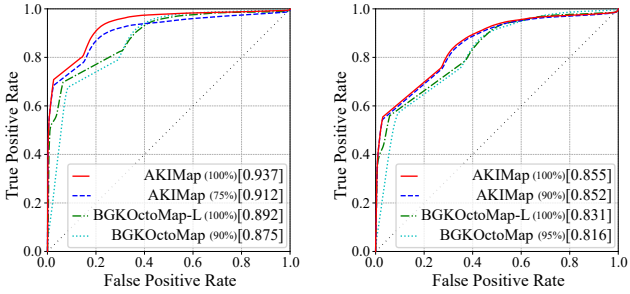
$$\beta_m^{1:t} = \beta_m^{1:t-1} + \sum_{i=1}^{N} |\mathbf{H}_i|^{-\frac{1}{2}} k(\mathbf{c}_m, \mathbf{x}_i, \mathbf{H}_i)(1 - y_i). \tag{9}$$

In a query step, we can achieve an occupancy probability of a cell based on Eq. 4 with these two equations, in a lazy evaluation manner.

V. RESULT

We have evaluated our approach using two synthetic scenes quantitatively and qualitatively, and have also tested for an on-the-fly mapping in the real environment. Specifically, we have compared our approach, AKIMap, with BGKOctoMap [8] and BGKOctoMap-L [7], which are the state-of-the-art techniques that rely on an isotropic kernel inference with a fixed bandwidth.

We have used a machine that has the 3.7GHz Intel i7-8700K CPU with 32GB memory for experiments, and all the tested methods were implemented in C++ and parallelized for exploiting 12 CPU cores. In all the following experiments, the parameters for BGKOctoMap and BGKOctoMap-L (e.g., the scale and the signal weight of isotropic kernel) were set by the values recommended in the previous papers.



(a) Structured dataset (b) Unstructured dataset

Fig. 4. Equal-time comparison. These graphs represent the ROC curves in structured (a) and unstructured (b) scenes of different methods running in the same time budget, except AKIMap (100%), as we vary the occupancy threshold. The number within the bracket is the AUC score of the map.

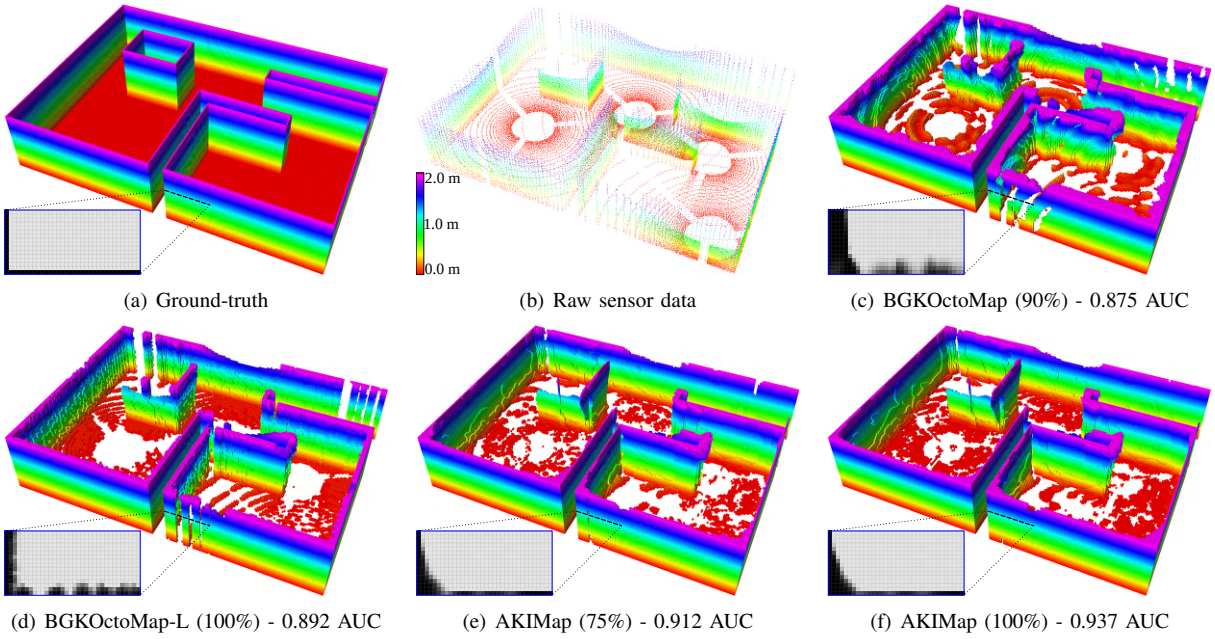
For our method, we set the sampling distance for free observations to 0.5 m used in BGKOctoMap and make one free sample per the sampling distance randomly. For the search range of neighbor samples, we can use a fixed search range of 0.5 m, the sampling distance. However, for a higher computational efficiency without degrading the estimation accuracy, we found that we can use a smaller search range near the sensor origin, since as the observed data is located close to the sensor origin, there is a higher chance that the data is densely populated. Based on this simple observation, we linearly decrease the search range from the maximum one of 0.5 m at the maximum sensor range to 0.05 m, the voxel size, at the sensor origin.

Note that the scales and signal weights of our anisotropic kernels are automatically adapted according to the distribution of the occupancy samples.

A. Performance comparison

To compare our method with the state-of-the-art techniques, we have used the two scenes, "structured" (Fig. 5) and "unstructured" (in our supplemental video), which were also used in the prior work [8]. Each virtual environment has dimensions of $10.0 \times 7.0 \times 2.0$ m in the Gazebo simulator. Both scenes consist of 12 scans captured at four different locations, and each scan has 3500 points. Fig. 5-(a) is the ground-truth for the benchmark scene. In this test, we aim to build the occupancy grid at a high resolution, i.e., 5 cm cell size, for observing the dense and sharp characteristics of the tested maps.

To evaluate our method quantitatively, we report the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve and the Mean Squared Error (MSE) of occupancy probability. The ROC curve shows the mapping ability for representing occupancy states given various occupancy thresholds; a higher AUC value represents higher robustness of the inference model. The MSE value of



(a) Ground-truth (b) Raw sensor data (c) BGKOctoMap (90%) - 0.875 AUC

(d) BGKOctoMap-L (100%) - 0.892 AUC (e) AKIMap (75%) - 0.912 AUC (f) AKIMap (100%) - 0.937 AUC

Fig. 5. These figures show the experimental results in the structured scene. We visualize the occupied cells classified based on the occupancy threshold (0.5), where the color represents the elevation from 0.0 m to 2.0 m. The grayscale 2-D image located in the left-bottom represents the occupancy probabilities of cells of a L-shape at the 1.0 m height; colors from white to black represent occupancy probability from the free to occupied states. The number in each parenthesis indicates the amount of used sensor measurements for each algorithm.

the occupancy map represents the average estimation error of the cells compared to the ground truth map. We report all the results on average in 10 experiments because our work uses a random approach to make free samples. Note that we do not show the standard deviation of performances explicitly, simply because the maximum variation, 0.002, of all the performance metrics is too small.

We first see the performances of different methods using all the sensor measurements in the structured and unstructured environments. As shown in the bottom rows of Table II, the proposed work reports the highest AUC and the lowest MSE compared to the other methods. For example, our method, AKIMap (100%), shows the robustness of occupancy estimation as reporting 0.937 AUC and 0.060 MSE in the structured scene.

These different methods take varying running times, and we thus see how they behave given the same running time budget. For the equal-time comparison, we decrease 5% of sensor data until the processing time of a mapping approach becomes similar to the time of the fastest one. To indicate how much of the original data is used for each method, we denote the usage of sensor measurements by the number in parenthesis next to the name of each mapping algorithm; e.g. (75%) means that its map uses 75% of the data, for reconstructing the occupancy representation.

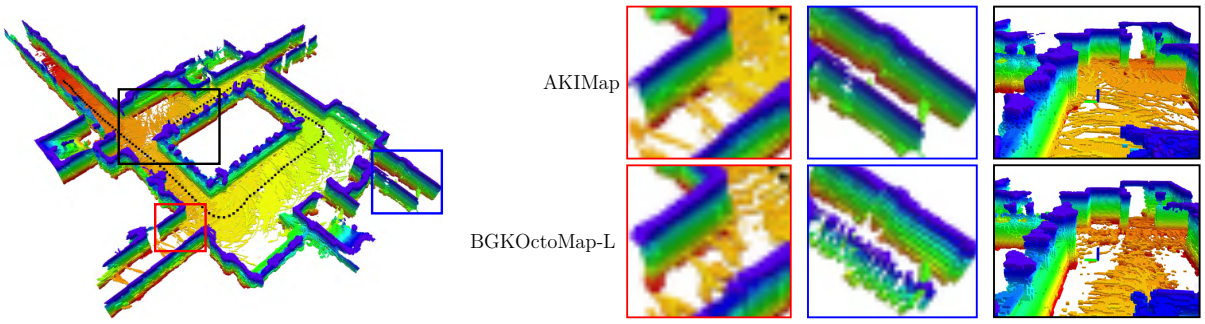
For the structured dataset, we pick 2.8 s that BGKOctoMap-L can process all the observation data. However, our method and BGKOctoMap can process only 75% and 90% of the data, due to their lower running performance, compared to BGKOctoMap-L. Nonetheless, our method shows the highest AUC result, thanks to its high estimation accuracy (Table II). For the unstructured dataset, we pick 2.5 s as the running time budget, since BGKOctoMap-L can process all the data given that budget. In this test data, ours achieves the highest estimation accuracy given the same running time. Fig. 4 shows ROC curves and their AUC values of the mapping approaches in the structured and unstructured environments in the equal-time comparison.

In this test, our approach outperforms the state-of-the-art methods, although it uses less data than the other methods. Our work shows 0.912 and 0.852 AUC scores in the structured and unstructured scenes, respectively. The top rows of Table II show the reconstruction errors of different occupancy maps in the equal-time comparison. Similar to the AUC scores, our approach produces the lower MSE scores, 0.063 and 0.102 for the two different scenes. These improvements are achieved mainly thanks to our anisotropic model, which allows for estimating the occupancy states robustly.

B. Qualitative analysis

We visualize mapping results of the prior experiment using the occupancy threshold of 0.5 at the equal time comparison. Fig. 5 shows 3-D visualizations of the various maps for each scene. We only draw the occupied cells classified by the threshold, and the color indicates the elevation of the cell. In addition, the 2-D grayscale image located in the left-bottom of each sub-figure shows the occupancy estimations of the cells at a particular region of a specific height, 1.0 m. In this analysis, we report the accuracy scores of occupied and free representations of various methods (Table II).

The isotropic inference models (Fig 5-(c) and Fig 5-



(a) Corridor scene - AKIMap (b) Enlarged visualization

Fig. 6. These figures show the results for on-the-fly mapping test using a mobile robot. The dotted line in the color figure indicates a trajectory of the robot, and the red and blue boxes show enlarged regions. The image associated with the black box represents the visualization at the viewpoint of follower. As shown in these boxes, ours represents the more sharp and dense surfaces of the wall and the ground in the corridor, compared to the prior work.

(d)) produce dense estimation results given the original, sparse sensor measurements shown in Fig 5-(b). These maps, however, do not robustly estimate occupied regions far from the sensor origin, where the data is more sparse than others. On the other hand, our anisotropic model makes such regions (e.g., wall) to have the occupied state thanks to our adaptive bandwidth selection guided by the local distribution of samples, as shown in Fig. 5-(e).

Furthermore, our method reconstructs the occupied space more accurately than the previous techniques. As shown in the 2-D images of Fig. 5-(c) and (d), the isotropic kernel models tend to produce over-estimation results for the occupied space. On the other hand, our anisotropic kernel maintains the sharp representation for the occupied region, shown in Fig. 5-(f), by adapting its kernel shapes locally according to the distribution of the sensor data.

As a result, shown in Table II, the proposed method shows the highest accuracy of occupied representation, 0.678 (structured) and 0.664 (unstructured), compared to the prior, isotropic estimations. This result shows that our work based on adaptive kernel inference is able to handle the sparsely distributed data where its density varies in a non-uniform manner.

C. On-the-fly Mapping Scenario

We have tested ours and the prior work, BGKOctoMap-L, in the real environment ($74.4 \times 49.2 \times 2.0$ m). Fig. 6 shows the occupied cells of maps with 5 cm voxel size, in which we reconstruct using a mobile robot that equips with lidar and IMU sensors. While the robot roams a corridor in 163 seconds, we find its pose in real-time by a localization method [22] and update the occupancy map from sensor measurements on-the-fly. The lidar sensor provides a raw point cloud at 10 Hz, but the mapping approaches use a new sensor data right after finishing their map updates. Due to the computational cost, these maps discard the sensor data acquired during the map updates.

In this test, we set the maximum sensing range to 10.0 m and the others to be the same with the settings used in the synthetic datasets, e.g., the maximum search range of our approach is set to follow the sampling distance, 0.5 m. The whole process of on-the-fly mapping can be shown in our video attachment.

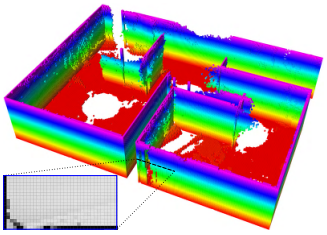
In the experiment, ours and the prior work use 130 and 134 scans for building and updating the occupancy maps, respectively. Fig. 6-(a) shows the visualization of occupied cells of our approach, where the color represents the relative elevation. Using sparse sensor measurements captured by the the lidar, our method produces the dense representations for the environment thanks to the inference model.

In Fig. 6-(b), we enlarge the 3-D visualizations at two regions within the red and blue boxes, and show a region within the black box at the viewpoint of follower. In the region within the red box, ours and the prior work show the dense reconstruction of an area in the corridor. But the proposed work makes the more sharp representation of the wall's surface compared to the prior work. In the cases of the blue and black boxes, the prior work cannot reconstruct the wall and the ground precisely, since it does not consider to adapt its kernel shape according to the density of data. On the other hand, ours enables to reconstruct the sharp and dense occupancy representations given the environment, and this is achieved mainly by our anisotrpic kernels that take account for the local distribution of samples.

D. Comparison of Gaussian Mixture Model

We compare the computational overhead of our work with the one of Gaussian Mixture Model based Occupancy Mapping (GMM-OM) [16], which controls the shape of each Gaussian distribution. This work, however, optimizes the covariance matrices globally with the other parameters such as means and weights of the Gaussian components, where the given number of components is much less than the occupancy samples. In this section, we test the computation efficiency of our local adaptation of the bandwidth matrix, comparing with the global optimization approach, GMM-OM.

Fig. 7 shows the GMM-OM in the structured scene when it makes 10^6 resamples from 70 component GMM. For comparing the computational efficiency, we choose the number of components at the setting that GMM-OM reports similar AUC and MSE scores with ours, AKIMap (100%)



(a) GMM-OM (100%)

Structured dataset		
	GMM-OM (100%)	AKIMap (100%)
Time [sec]	90.91	**3.99**
- training	32.86	**1.13**
- updating	58.05	**2.86**
AUC	0.939	0.937
MSE	0.065	0.060

(b) Performance comparison

Fig. 7. This figure (a) shows the visualization of 70-component GMM-OM in the structured scene, and the table (b) shows the computational performances when two methods report the similar AUC and MSE scores. In the test, our approach using local bandwidth optimization processes much faster than the prior work based on the global approach.

shown in Fig. 5-(f). For the number of resamples, we set it to the value recommended in the paper.

In this setting, GMM-OM reports 0.939 AUC and 0.065 MSE, while the global optimization for training the GMMs takes 32.86 seconds, shown in Fig. 7-(b). On the other hand, for achieving similar representation performances, our bandwidth optimization takes 1.13 seconds, and finally we reconstruct the map within 4 seconds. These results show that our local adaptive method is more computationally efficient than the GMM-OM based on the global approach, while producing a robust occupancy representation through the anisotropic kernel inference.

VI. Conclusion

We have proposed a new method, AKIMap, to use an adaptive kernel inference for the dense and sharp representation on the occupancy grid. For high estimation accuracy, our work has proposed to find the adaptive kernel bandwidth based on the local distribution of occupancy samples efficiently. As a result, our method has shown the robust occupancy representations with efficient computing in two synthetic scenes, compared to the prior methods. Furthermore, we have demonstrated the practical benefits of our adaptive approach in on-the-fly mapping. Compared to the isotropic approaches, our method has shown the accurate occupancy representations of the environment, thanks to the varying shape and size of our adaptive kernel estimations.

Limitations and future work. In this work, we used a scalar scale to adjust the bandwidth matrix. Because a full bandwidth matrix has 6 degrees-of-freedom (DOFs), our solution could be sub-optimal by exploring only 4 DOFs, while it is efficient and effective. For example, if we optimize three scales for all the rotated axes individually, the kernel estimation could follow the local surface of an object more accurately without false estimations. We would like to extend our current approach to explore the full search space effectively while keeping the computational overhead small, to broaden the impact of our adaptive kernel inference.

References

[1] H Moravec, "Robot spatial perceptionby stereoscopic vision and 3d evidence grids", *Perception,(September)*, 1996.

[2] Alberto Elfes, "Using occupancy grids for mobile robot perception and navigation", *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[3] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees", *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[4] Youngsun Kwon, Donghyuk Kim, Inkyu An, and Sung-eui Yoon, "Super rays and culling region for real-time updates on grid-based occupancy maps", *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 482–497, 2019.

[5] Simon T O'Callaghan and Fabio T Ramos, "Gaussian process occupancy maps", *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 42–62, 2012.

[6] Fabio Ramos and Lionel Ott, "Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent", *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1717–1730, 2016.

[7] Kevin Doherty, Tixiao Shan, Jinkun Wang, and Brendan Englot, "Learning-aided 3-d occupancy mapping with bayesian generalized kernel inference", *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 953–966, 2019.

[8] Kevin Doherty, Jinkun Wang, and Brendan Englot, "Bayesian generalized kernel inference for occupancy map prediction", in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3118–3124.

[9] Soohwan Kim and Jonghyuk Kim, "Gpmap: A unified framework for robotic mapping based on sparse gaussian processes", in *Field and service robotics*. Springer, 2015, pp. 319–332.

[10] Jinkun Wang and Brendan Englot, "Fast, accurate gaussian process occupancy maps via test-data octrees and nested bayesian fusion", in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1003–1010.

[11] Kevin Doherty, Jinkun Wang, and Brendan Englot, "Probabilistic map fusion for fast, incremental occupancy mapping with 3d hilbert maps", in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1011–1018.

[12] Vitor Guizilini and Fabio Ramos, "Towards real-time 3d continuous occupancy mapping using hilbert maps", *The International Journal of Robotics Research*, vol. 37, no. 6, pp. 566–584, 2018.

[13] Ransalu Senanayake, Anthony Tompkins, and Fabio Ramos, "Automorphing kernels for nonstationarity in mapping unstructured environments.", in *CoRL*, 2018, pp. 443–455.

[14] Weiming Zhi, Lionel Ott, Ransalu Senanayake, and Fabio Ramos, "Continuous occupancy map fusion with fast bayesian hilbert maps", in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4111–4117.

[15] Ransalu Senanayake and Fabio Ramos, "Bayesian hilbert maps for dynamic continuous occupancy mapping", in *Conference on Robot Learning*, 2017, pp. 458–471.

[16] Cormac O'Meadhra, Wennie Tabib, and Nathan Michael, "Variable resolution occupancy mapping using gaussian mixture models", *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2015–2022, 2018.

[17] Elizbar A Nadaraya, "On estimating regression", *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[18] Geoffrey S Watson, "Smooth regression analysis", *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372, 1964.

[19] Artur Gramacki, *Nonparametric kernel density estimation and its computational aspects*, Springer, 2018.

[20] Arman Melkumyan and Fabio Ramos, "A sparse covariance function for exact gaussian process inference in large datasets", in *IJCAI*, 2009, vol. 9, pp. 1936–1942.

[21] Vitor Guizilini and Fabio Ramos, "Large-scale 3d scene reconstruction with hilbert maps", in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 3247–3254.

[22] Tixiao Shan and Brendan Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain", in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.