2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Integrated Mapping and Path Planning for Very Large-Scale Robotic (VLSR) Systems

Julian Morelli, Pingping Zhu, Bryce Doerr, Richard Linares, and Silvia Ferrari

Abstract— **This paper develops a decentralized approach for mapping and information-driven path planning for Very Large Scale Robotic (VLSR) systems. In this approach, obstacle mapping is performed using a continuous probabilistic representation known as a Hilbert map, which formulates the mapping problem as a binary classification task and uses kernel logistic regression to train a discriminative classifier online. A novel Hilbert map fusion method is presented that quickly and efficiently combines the information from individual robot maps. An integrated mapping and path planning algorithm is presented to determine paths of maximum information value, while simultaneously performing obstacle avoidance. Furthermore, the effect of how percentage communication failure effects the overall performance of the system is investigated. The approach is demonstrated on a VLSR system with hundreds of robots that must map obstacles collaboratively over a large region of interest using onboard range sensors and no prior obstacle information. The results show that, through fusion and decentralized processing, the entropy of the map decreases over time and robot paths remain collision-free.**

I. INTRODUCTION

Large multi-robot autonomous systems are highly desirable as their effectiveness of cooperatively performing a given task can far surpass that of a single robot. These systems, also known as Very Large Scale Robotic (VLSR) systems, are becoming increasingly more practical as the cost of small, but computationally powerful robots decreases, and onboard communications and functionalities allow them to collaborate and share information on common goals [1], [2]. Systems comprised of hundreds of small, homogeneous robots are being considered as a solution to tasks that were previously done by a single robot such as mapping, search and rescue, and other industrial and military applications. These systems are especially powerful when tasks involve large spatio-temporal operations in changing environments, such as ocean surface or underwater regions [3]. For example, the search for a lost vessel at sea can be performed more effectively with a large number of autonomous vehicles than with one or few vehicles, but the performance gains scales reasonably with cost only if the vehicles collaborate and share information effectively.

Julian Morelli is with Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, US jam888@cornell.edu
Pingping Zhu is with Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, US pingping.zhu@cornell.edu
Bryce Doerr is with the Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, US doerr024@umn.edu
Richard Linares is with the Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, US rlinares@umn.edu
Silvia Ferrari is with Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, US ferrari@cornell.edu

Many approaches to mapping and information-driven path planning and control have been explored over the past few decades, but most methods do not scale with the number of robots, or decompose the problem in such a way that information fusion and collaboration are not necessarily optimal [3]. Prioritized path planning, for example, plans the path of every robot individually and then accounts for mutual collisions at the expense of optimality [4]. Distributed optimal control (DOC) has been shown to optimize the performance of VLSR systems in tasks such as traversing an obstacle populated environment, maintaining a desired formation, or optimizing the coverage of a distributed sensor network [5]–[10]. Existing DOC approaches, however, are centralized and rely on prior information, such as known obstacle geometries, and sharing of robot information state. To move toward increased autonomy over large spatio-temporal scales, however, VLSR systems must be capable of planning with little or no prior information about the region of interest (ROI), and thus, the environment must be mapped through limited information exchange, such as bounded communication range.

Existing multi-robot mapping methods to date have considered teams of less than a dozen robots in a small environment, where each robot is capable of performing simultaneous localization and mapping (SLAM) [11]. To overcome drawbacks of classic occupancy grid methods, which assume cells in the map are independent, a novel continuous occupancy mapping approach has been recently presented in [12], by combining binary classification with a kernel logistic regression classifier, in what is now known as Hilbert mapping. Other continuous occupancy mapping approaches, such as Gaussian Process Occupancy Maps (GPOM) [13]–[15], are more computationally complex and cannot be implemented online. The Hilbert mapping approach is adopted and extended here because it uses efficient kernel approximations and stochastic gradient descent to train a discriminative classifier in a shorter amount of time, and thus, scales to large datasets obtained by VLSR systems.

This paper presents a novel kernel logistic regression fusion method for merging maps obtained by many distributed robots efficiently online, while simultaneously computing information-driven paths using the expected entropy reduction (EER) computed from the robot Hilbert maps. The integrated mapping and path-planning problem is described in Section II. The kernel logistic regression method for mapping, fusion, and path planning is described in Section IV.

978-1-5386-6027-0/19/$31.00 ©2019 IEEE

The effectiveness of the proposed approach is demonstrated numerically in Section V by controlling 200 robots that explore a large ROI by mapping unknown obstacles using only onboard range sensors.

II. Problem Formulation

Consider the problem of optimally planning the trajectory of a VLSR system comprised of N cooperative robots engaged in environmental mapping through a large, obstacle-populated ROI. The robot workspace is denoted by $\mathcal{W} \subset \mathbb{R}^2$. A set of fixed, unknown rigid obstacles $\mathcal{B}_1, ..., \mathcal{B}_r \subset \mathcal{W}$ populate $\mathcal{W}$, where $\mathcal{B} = \bigcup_i^r \mathcal{B}_i$. Let $\mathcal{U} \subset \mathbb{R}^m$ denote the space of admissible actions or controls. The dynamics of each robot are governed by a stochastic differential equation (SDE),

$$\begin{aligned} \dot{\mathbf{x}}_i(t) &= \mathbf{f}[\mathbf{x}_i(t), \mathbf{u}_i(t), t] + \mathbf{G}\mathbf{w}(t), \\ \mathbf{x}_i(T_0) &= \mathbf{x}_{i_0}(T_0), \quad i = 1, ..., N \end{aligned} \tag{1}$$

where $\mathbf{x}_i(t) \in \mathcal{W}$ denotes the ith robot state, $\mathbf{u}_i(t) \in \mathcal{U}$ denotes the ith robot action or control, and $\mathbf{x}_{i_0}$ denotes the robot initial conditions at initial time T_0. The robot dynamic equation (1) is characterized by an additive Gaussian disturbance vector of independent and identically distributed (iid) random variables, denoted by $\mathbf{w}(t) \in \mathbb{R}^2$, and $\mathbf{G} \in \mathbb{R}^{2\times 2}$ is a constant matrix.

The objective of the VLSR system is to cooperatively map an environment by building a model of the obstacles $\mathcal{B}$ in $\mathcal{W}$. The quantity, location, and geometry of these obstacles are unknown *a priori*. Each robot constructs only a portion of the map using onboard (local) measurements and shares information only with neighbors within a limited communication range ρ, i.e. communication occurs only if $\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|_2 < \rho$, where $\|\cdot\|_2$ denotes the Euclidean norm. Every robot must be capable of building a local map for obstacle avoidance, while also improving it over time by communicating with other robots within a distance ρ. For simplicity, the state of each robot is assumed fully observable and known without error. All robots are equipped with identical omnidirectional range sensors, e.g. rotating laser range finders, characterized by operating conditions and parameters represented by $\lambda \in \mathbb{R}^{\wp}$. The field of view (FOV) of the sensor onboard robot i, denoted by $\mathcal{S}(\mathbf{x}_i) \subset \mathcal{W}$, can be defined as a disk of constant radius r centered at $\mathbf{x}_i$, where r is the sensor maximum range.

Measurements at any location $\mathbf{x}_M \in \mathcal{S}(\mathbf{x}_i)$ take the form $\mathbf{x}_M = \mathbf{x}_i + d\hat{\mathbf{e}}_r$, where $\hat{\mathbf{e}}_r = \hat{x}\cos\theta + \hat{y}\sin\theta$ is a unit vector pointing radially outwards from $\mathbf{x}_i$ and $\hat{x}$, $\hat{y}$ are unit vectors defining a coordinate frame $\mathcal{F}_{\mathcal{A}}$ fixed to the robot. θ is the angle of $\hat{\mathbf{e}}_r$ measured with respect to $\hat{x}$ and $d \leq r$ is the length of the vector. As shown by the schematic in Fig. 1, $\mathbf{x}_M$ can be localized in an inertial coordinate frame $\mathcal{F}_{\mathcal{W}}$, given $\mathbf{x}_i$, d, and θ. Let Y be a discrete random variable such that $Y : \mathcal{W} \to \mathcal{Y}$ where $\mathcal{Y} = \{0,\ 1\}$. Y is a discontinuous function with a finite range such that, for $\mathbf{x} \in \mathcal{W}$

$$Y(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{B} \\ 0, & \text{if } \mathbf{x} \notin \mathcal{B} \end{cases} \tag{2}$$

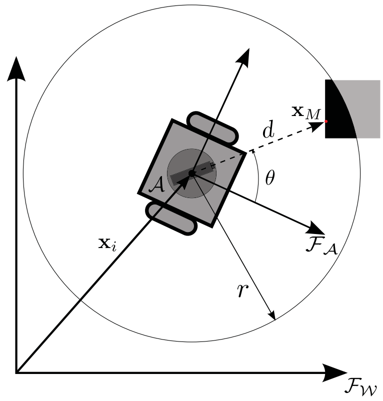


Fig. 1. Schematic of the ith robot and sensor

Y is assumed hidden, and thus, it can never be observed. Therefore, Y must be inferred from a set of measurements $\mathcal{M}_k = \{Z_1, ..., Z_k\}$ where Z_k denotes a measurement taken at a discrete time $k = k\Delta t$ for some small Δt. Measurements are modeled as a random vector $Z = [\hat{X}_M,\ \hat{Y}]^T$, which is used to estimate the true values $T = [X_M,\ Y]^T$, where $\hat{(\cdot)}$ represents an estimated quantity and $[\cdot,\ \cdot]^T$ indicates the matrix transpose. Z is related to Y through a known mixed joint probability function $p(Z,\ Y,\ \lambda)$. The joint probability function is known as *mixed* because it contains both continuous type random variables (X_M) and discrete type random variables (Y). Y can be thought of as a discriminative classification variable, partitioning $\mathcal{W}$ into two disjoint sets: $\mathcal{C}_{free} = \{\mathbf{x} \in \mathcal{W} : \mathbf{x} \notin \mathcal{B}\}$ and $\mathcal{B}$, representing the empty and occupied regions of $\mathcal{W}$.

The uncertainty associated with the onboard sensors can be thought of as two-fold: uncertainty associated with misclassification, and uncertainty associated with the returned measurement location. The first type of uncertainty is specified by the discrete portion of $p(Z,\ Y,\ \lambda)$ which defines how often a sensor misclassifies a measurement location, $P(\hat{Y} = \hat{y} \mid Y = y)$. This PMF can be defined by two parameters β and $\gamma \in [0, 1]$ which are generally close to 1 and 0, respectively. The probabilities can be defined as $P(\hat{Y} = 1 \mid Y = 1) = \beta$ and $P(\hat{Y} = 1 \mid Y = 0) = \gamma$. The second type of uncertainty relating to the returned angle and distance can be modeled as an exponential power law. This formulation is widely applicable to sensors such as acoustic, magnetic, and optical sensors, where measurements are governed by linear wave propagation models [16]. The exponenital power law models the received isotropic energy generated by a constant target source level and attenuated by the environment [17].

Consider a measurement such that the true distance from the target to the sensor is d. The distance d can be estimated from a measurement D obtained according to the following power law

$$D = a\|d\|_2^{-\alpha} + \nu \tag{3}$$

where the true distance is altered by an attenuation coefficient α and a scaling constant a that are chosen based on the environmental conditions and target characteristics. Both the final distance and angle measurements are subject to zero-mean Gaussian noise $\nu \sim \mathcal{N}(0, \sigma_d)$ and $\upsilon \sim \mathcal{N}(0, \sigma_a)$, respectively, representative of sensor noise and accuracy. In this way, $\hat{X}_M$ returned by the sensor is subject to error with a known covariance. The sensor and environmental parameters can be represented by $\lambda = [a, \alpha, \sigma]^T$. If the sensor returns a

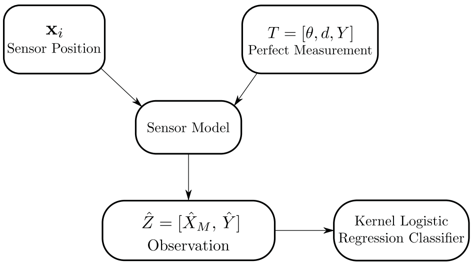


Fig. 2. Mapping Process Schematic

distance value $d < r$, this measurement is called a "hit" point and it is assumed reflected by the boundary of an obstacle and considered as "occupied". All other observed points can be considered "not occupied," or "empty." Essentially, a map of $\mathcal{W}$ is determined by where these hit points are located, determining the boundaries of obstacles $\mathcal{B}_i$. These measurements are stored onboard each robot and used to learn the parameters of the Hilbert map. A schematic of the mapping process can be seen in Fig. 2. Since each agent has a finite range FOV, every area of the ROI needs to be observed in order to accurately infer $Y(\mathbf{x})$. Areas of the map with little or no measurement data will not represent the ROI accurately. This motivates a path planning problem to maximize the information obtained by each onboard sensor.

The performance of the VLSR system over a fixed time interval $[T_0, \ T_f]$ is expressed as an integral cost function of the expected information gain, denoted $\hat{\varphi}(Y; Z \mid \mathcal{M}, \lambda)$, given all past measurements $\mathcal{M}$, an obstacle avoidance term, and control usage

$$J = \int_{\mathcal{W}} \hat{\varphi}(Y; Z \mid \mathcal{M}, \lambda) d\mathbf{x} + \int_{T_0}^{T_f} \mathbf{x}_i^T U(\mathbf{x}_i)\mathbf{x}_i + \mathbf{u}_i^T R \mathbf{u}_i \ dt \tag{4}$$

where $U(\mathbf{x}_i)$ is an obstacle repulsion term that penalizes robots for being in close proximity to an obstacle, and R is a positive definite matrix that weighs the importance of the elements of $\mathbf{u}$.

The VLSR optimal planning problem becomes finding the optimal robot state $\mathbf{x}_i$ and control $\mathbf{u}_i$, for all i, that minimizes the cost function J over the time interval $[T_0, \ T_f]$, subject to (1).

III. Information Theoretic Objective Functions

Information theoretic functions are very well suited to quantify the information gained through sensor measurements and can be used to plan the path of a robot in order to maximize the expected gain in information value of future measurements. A natural choice for measuring information value is entropy [18]. The entropy $H(X)$ of a discrete random variable X with finite range $\mathcal{X}$ is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{5}$$

and is a measure of the uncertainty of X. If the logarithm is to the base 2, this is called Shannon Entropy and is expressed in bits. Even though it is a great choice for measuring uncertainty, entropy is not well suited for estimating the information gain of future measurements, as computing it requires the probability mass function to be known. Entropy is also nonadditive, not a true metric, and myopic, meaning that it does not consider the effects of prior measurements to those that are performed subsequently [19].

Two other useful quantities are the conditional entropy and the expected entropy. Let X, Y, and Z be discrete random variables. The conditional entropy of Y given X can be expressed as the following

$$H(Y \mid X) = -\sum_{Y} P(Y \mid X) \log P(Y \mid X) \tag{6}$$

and the expected conditional entropy is given by

$$\mathbb{E}[H \mid X] = \sum_{Z} H(Y \mid Z) P(Z \mid X) \tag{7}$$

These quantities can be used to compute information theoretic objective functions such as the expected discrimination gain or the expected entropy reduction. Often, these functions are used in sensor planning applications to estimate the information gain of measurements that have not been taken yet. This allows for a planning method to be constructed to maximize the expected information gain over a trajectory.

IV. Methodology

In order to cooperatively map the ROI, each robot must be capable of constructing a map given data collected from its onboard sensor. The chosen map representation in this paper is a Hilbert map, originally presented by Fabio Ramos and Lionel Ott in [12]. Hilbert mapping formulates the mapping problem as a binary classification task using kernel logistic regression. Below is an in depth treatment of kernel logistic regression and how it can be used to construct a spatial map.

A. Mapping with Kernel Logistic Regression

A Hilbert map is a continuous occupancy map developed by formulating the mapping problem as a binary classification task. Let $\mathbf{x} \in \mathcal{W}$ be any point in $\mathcal{W}$ and $y \in \{0, 1\}$ be defined as a categorical variable such that

$$y = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{B} \\ 0, & \text{if } \mathbf{x} \notin \mathcal{B} \end{cases} \tag{8}$$

The classification task is as follows: given any point $\mathbf{x}$, predict which class y it belongs to.

There are many different classification methods, but not all can be applied to this task. The classifier must be able to learn non-linear decision boundaries, be updated incrementally and online, and give outputs with confidence levels. A logistic regression model is chosen. Although it does not supply any covariance information, it outputs a value in the range $[0, 1]$, which can be interpreted as the probability of $\mathbf{x}$ belonging to a certain class y.

Define the feature space to be the same space as the workspace, $\mathcal{W}$. The features used here are the spatial coordinates q_1 and q_2 that make up the vector $\mathbf{q} = [q_1,\ q_2]^T$ These spatial features are separate from the robot state, which is denoted by $\mathbf{x}_i = [x,\ y]$. This is an important distinction as a robot mitigates the risk of collision by not entering areas of the ROI with a high probability of collision. Therefore, a robot needs to know the probability of occupancy of points it will visit in the future and not only of its current position.

In order to capture the complex nonlinearities of the obstacles in $\mathcal{W}$, the feature space is enlarged using a basis expansion [20]. Define the lifting operator $\Phi : \mathcal{W} \to \mathbb{R}^{\mu}$, where $\mu \in \mathbb{N}$ and $\mu > 2$, as an operator that transforms any point in $\mathcal{W}$ to an enlarged feature space of dimension μ. As the robots navigate the environment, each capture a data set $\mathcal{D}_i = \{\mathbf{q}_j, y_j\}_{j=1}^{\mu_i}$, where $\mathcal{D}_i$ is the data set stored by robot i and μ_i is the current dimension of the training set of robot i. It is important to note that μ_i grows with time as the robots are constantly collecting data and any two robots will most likely have a different number of training samples. A $\mathbf{q}$ with a subscript j represents the jth data sample in a robot's data set, but since the following relations hold for *all* $\mathbf{q} \in \mathcal{W}$, not only the ones stored by a robot, the subscript is dropped for the following discussion.

The function defining the lifting operator $\Phi(\mathbf{q})$ never has to be explicitly specified for the following reasons. The equations in logistic regression depend only on the inner product between these functions, i.e. $\langle \Phi(\mathbf{q}), \Phi(\mathbf{q}') \rangle$, $\mathbf{q}, \mathbf{q}' \in \mathcal{W}$. Therefore, only knowledge of a Kernel function

$$K(\mathbf{q}, \mathbf{q}') = \langle \Phi(\mathbf{q}), \Phi(\mathbf{q}') \rangle \tag{9}$$

that computes the inner products in the transformed space needs to be known. A kernel function K must be a symmetric positive definite function, i.e. $K(x, y) = K(y, x)$ and $K(x, x') \geq 0\ \forall x, x'$. In this paper, the Radial Basis Function kernel (10) is used

$$K(\mathbf{q}, \mathbf{q}') = \exp\Big(-\frac{1}{2\sigma^2} \|\mathbf{q} - \mathbf{q}'\|_2^2 \Big) \tag{10}$$

where σ is the variance. The Radial Basis Function kernel is a Mercer kernel, as it is continuous, symmetric, and positive definite. According to Mercer's theorem, any Mercer kernel $K(\mathbf{q}, \mathbf{q}')$ induces a mapping $\Phi(\mathbf{q})$ from the input space $\mathcal{W}$ to a high-dimensional feature space such that the property seen in (9) holds [21]. Therefore, the lifting operator $\Phi(\mathbf{q})$ never has to be explicitly specified.

Let $Y(\mathbf{q})$ be a random variable defined as it is in Section II, i.e. representing the occupancy of $\mathbf{q}$, and define Q as a random variable such that realizations are spatial points in $\mathcal{W}$. Define $P(Y = 1 \mid Q = \mathbf{q}_*; \mathbf{w}_i)$ as the probability that queried point $\mathbf{q}_*$ is occupied ($\mathbf{q}_* \in \mathcal{B}$), given a vector of parameters $\mathbf{w}_i$, which is to be learned online. The subscript i indicates that vector $\mathbf{w}_i$ is stored by robot i. The probability of occupancy can now be formulated by the following

$$P(Y = 1 \mid Q = \mathbf{q}_*; \mathbf{w}_i) = 1 - \frac{1}{1 + e^{(\mathbf{w}_i^T \Phi(\mathbf{q}_*) + b)}} \tag{11}$$

where b is a scalar known as the intercept or bias. The bias can be made part of $\mathbf{w}_i$ by appending a 1 to the data, i.e. $[1 \quad \Phi(\mathbf{q})^T]^T$. This parameter will therefore be omitted as it is assumed to be absorbed in the data. Equation (11) is known as the *primal* form of the classifier.

Learning the parameters of the model presented in (11) can be accomplished as follows. Define the parameter vector $\mathbf{w}_i$ to be learned as:

$$\mathbf{w}_i = \sum_{j=1}^{\mu_i} \alpha_j \Phi(\mathbf{q}_j) \tag{12}$$

The basis expansion is clear here. $\Phi(\mathbf{q}_j)$ is a basis function constructed from the jth data sample and α_j (a scalar) is its coefficient. While this means learning μ_i parameters, which grows as the data grows, kernel approximation or feature selection methods can be used to limit the number of parameters needed to be learned. The kernel in (10) can be approximated by Nystroem Features or Random Fourier Features [22], [23], which will limit the number of basis functions to some user chosen number, or a feature selection method can be used such as quantization [21]. In this problem, a heuristic quantization that is similar to the method presented in [21] is used to limit the size of the kernel matrix used in learning the parameters α_j for computational reasons.

Plugging (12) into (11):

$$\begin{aligned} P(Y = 1 \mid Q = \mathbf{q}_*; \mathbf{w}_i) &= 1 - \frac{1}{1 + e^{(\sum_j^{\mu_i} \alpha_j \Phi(\mathbf{q}_j) \cdot \Phi(\mathbf{q}_*))}} \\ &= 1 - \frac{1}{1 + e^{(\sum_j^{\mu_i} \alpha_j K(\mathbf{q}_*, \mathbf{q}_j))}} \end{aligned} \tag{13}$$

The inner product $\langle \Phi(\mathbf{q}_j), \Phi(\mathbf{q}_*) \rangle$ is expressed in terms of the chosen kernel function. This is known as the *dual* form of the classifier.

The coefficients α_j that define the function can be learned by defining a loss function

$$L = \sum_{m=1}^{M} \ell_m + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2 \tag{14}$$

where $\ell_m = -\ln [P(Y = y_m | Q = q_m)]$ is the negative likelihood (NLL) of the data (q_m, y_m). The coefficents α_j can therefore be found by minimizing L with respect to α_j. This can be done using gradient descent or Newton-Rhapson method, both of which have their own advantages.

B. Map Fusion

Consider that there are two agents with learned functions $f_1$and f_2 from the data sets $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. Then, the posteriors of these two agents at the location $\mathbf{q}$ can be expressed by

$$P(Y = 1 | Q = \mathbf{q}, \mathcal{D}_1) = P(Y = 1 | \mathbf{q}, \mathcal{D}_1) = \frac{e^{f_1(\mathbf{q})}}{1 + e^{f_1(\mathbf{q})}} \tag{15}$$

$$P(Y=1|Q=\mathbf{q},\mathcal{D}_2)=P(Y=1|\mathbf{q},\mathcal{D}_2)=\frac{e^{f_2(\mathbf{q})}}{1+e^{f_2(\mathbf{q})}} \tag{16}$$

Then the posterior after these two agent sharing their information can be expressed by

$$P(Y=1|\mathbf{q},\mathcal{D}_1,\mathcal{D}_2)=\frac{e^{f_1(\mathbf{q})+f_2(\mathbf{q})-\ln\gamma}}{e^{f_1(\mathbf{q})+f_2(\mathbf{q})-\ln\gamma}+1} \tag{17}$$

where $\gamma = P(Y=1|\mathbf{q})/P(Y=0|\mathbf{q})$ is the ratio between the prior at the location $\mathbf{q}$. According the above equation, the posterior $P(Y=1|\mathbf{q},\mathcal{D}_1,\mathcal{D}_2)$ can also be expressed in a same form as the posterior before information sharing, such as

$$P(Y=1|\mathbf{q},\mathcal{D}_1,\mathcal{D}_2)=\frac{e^{f_F(\mathbf{q})}}{1+e^{f_F(\mathbf{q})}} \tag{18}$$

Here, f_F is the fusion function, which is updated by

$$f_F(\mathbf{q})=f_1(\mathbf{q})+f_2(\mathbf{q})-\ln\gamma. \tag{19}$$

Assuming that the prior is even, $\gamma = P(Y=1|\mathbf{q})/P(Y=0|\mathbf{q})=1$, then the fusion function is rewritten as

$$f_F(\mathbf{q})=f_1(\mathbf{q})+f_2(\mathbf{q}). \tag{20}$$

C. Path Planning on Hilbert Maps

From the measurement model $p(Z,Y,\lambda)$ presented in Section II, the expected benefit of future measurements can be computed using the prior belief,

$$P(Y\mid\mathcal{M}_{k-1},\lambda)=\frac{P(Z_{k-1}\mid Y,\lambda)P(Y\mid\mathcal{M}_{k-2},\lambda)}{\sum_{y\in\mathcal{Y}}P(Z_{k-1}\mid Y=y,\lambda)P(Y=y\mid\mathcal{M}_{k-2},\lambda)} \tag{21}$$

which assumes measurements obtained at different time instants are conditionally independent given the target state [17].

Then, as shown in [17], the posterior belief can be computed by applying Bayes' rule for every $z\in\mathcal{Z}$, i.e.

$$p(Y\mid Z=z,\mathcal{M}_{k-1},\lambda)=\frac{p(z\mid Y,\lambda)p(Y\mid\mathcal{M}_{k-1},\lambda)}{\sum_{y\in\mathcal{Y}}p(z\mid Y=y,\lambda)p(Y=y\mid\mathcal{M}_{k-1},\lambda)} \tag{22}$$

The terms $p(Z_{k-1}\mid Y,\lambda)$ and $p(z\mid Y,\lambda)$ are known from the measurement model and $p(Y\mid\mathcal{M}_{k-2},\lambda)$ and $p(Y\mid\mathcal{M}_{k-1},\lambda)$ are known from the Hilbert map. Therefore, the EER can be computed as

$$\begin{aligned}\Delta H(Z_k)&=H(Y\mid Z_{k-1},\lambda)-\mathbb{E}[H\mid Z_k,\lambda]\\&=-\sum_Y p(Y\mid\mathcal{M}_{k-1},\lambda)\log p(Y\mid\mathcal{M}_{k-1},\lambda)\\&-\sum_{z\in\mathcal{Z}}\int_{\mathcal{F}}H(Y\mid Z_k=z,\mathcal{M}_{k-1},\lambda)p(Z=z\mid\mathcal{M}_{k-1},\lambda)\end{aligned} \tag{23}$$

where

$$p(Z_k\mid\mathcal{M}_{k-1})=\sum_Y p(Z_k\mid Y)p(Y\mid\mathcal{M}_{k-1}) \tag{24}$$

and $\mathcal{F}$ denotes the area covered by the FOVs of all robots. Therefore, the EER can be computed prior to acquiring the next measurement Z_k, using the sensor model $p(Z,Y,\lambda)$, the PMFs in (21), (22), and the current Hilbert map.

Then, an information-driven path can be planned by each robot using the information roadmap method presented in [24]. Let $\mathcal{G}_i(V_i,E_i)$ represent a topological graph where V_i is a set of nodes or vertices and E_i is a set of edges connecting nodes. Each robot constructs a local roadmap by sampling nodes from a probability distribution constructed from the EER and the repulsive potential computed from the Hilbert map, with higher EER areas being sampled more often. Since some of the obstacles in $\mathcal{W}$ are unknown or uncertain, sampled nodes are connected by an edge only if the corresponding path intersects cells characterized by a probability of occupancy less than 0.5.

From $\mathcal{G}_i$, an adjacency matrix is constructed and a tree search algorithm is implemented to find the path of highest reward value. The branch lengths are limited to only a few nodes, as the total number of nodes in each local roadmap is small, and the algorithm is greedy in that it maximizes the information gain over a small horizon. A greedy algorithm is used because a large portion of the ROI displays high EER while unexplored, as illustrated in Fig. **??**. After the optimal path is found, each robot moves to the next node, obtains a new measurement, updates the map, and constructs a new optimal path.

V. Simulation and Results

The VLSR system consists of a network of $N = 200$ robots characterized by single integrator dynamics,

$$\dot{\mathbf{x}}_i=\mathbf{u}_i+\mathbf{I}_2\mathbf{w}_i,\quad i=1,\dots,N \tag{25}$$

where $\mathbf{x}_i=[x,\,y]^T$ is the robot state vector, x, y are the inertial coordinates, $\mathbf{u}_i=[u_1,\,u_2]^T$ is the robot control vector comprised of the x- and y-velocity components, and $\mathbf{w}_i$ is a random disturbance, with elements sampled from the bivariate normal distribution $\mathcal{N}(\mathbf{0},\Sigma)$ where $\Sigma=0.01\times\mathbf{I}_2$. The robot state is assumed fully observable and error free. The above VLSR system is tasked with mapping a ROI $\mathcal{W}=[0,L_x]\times[0,L_y]$ where $L_x=20$km and $L_y=16$km, with an unknown obstacle layout shown in black in Fig. 3, and over a fixed time interval $[0,\,T_F]$ where $T_F=100$.

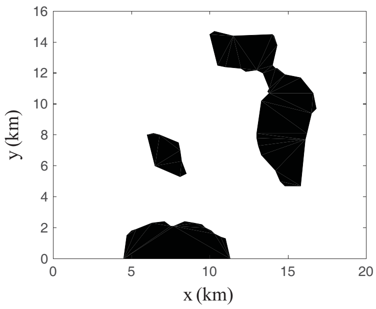
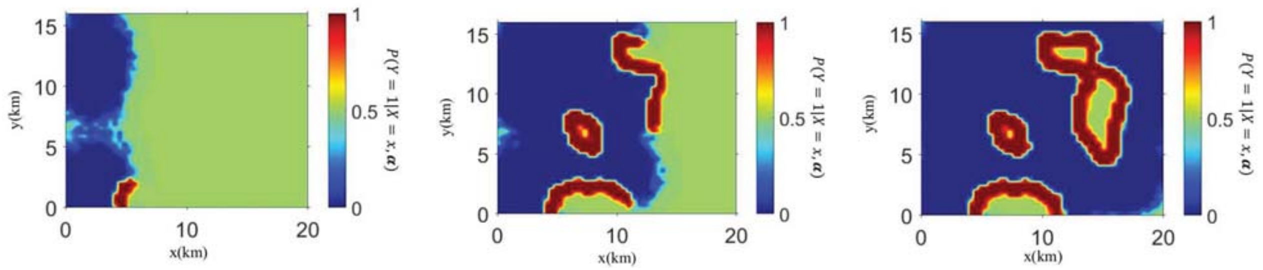


Fig. 3. ROI and obstacle layout.



(a) (b) (c)

Fig. 4. Hilbert map at three sample moments in time.

The robots are initialized by sampling from a given Gaussian Mixed Model with 2 components where $\mu = \{[2, 2], [2.5, 12]\}$ and the corresponding covariance matrices, where values are placed down the diagonal, are $\Sigma = \{[1, 1], [1, 4]\}$. The robots sense the environment with a simulated range finder with a maximum range $r = 2$km, and zero-mean Gaussian perturbations added to the final measurements with $\sigma_n = 0.05$. Each robot scans the environment individually, but for computational purposes, all data are stored by a centralized computer. As shown in Section IV-B, data can be fused by robots within a limited communication range to obtain the centralized map. From each scan, a dataset $\mathcal{D} = \{\mathbf{x}_p, y_p\}_{p=1}^{\mu}$ is used to construct a Hilbert map. A heuristic quantization algorithm is used to limit the size of the resulting kernel matrix. A set of quantization points $Q = \{q\}$ is chosen on a grid that is 100×100 points. For every $\mathbf{x}_p \in \mathcal{D}$, each quantization point q keeps a running average of the sum y_p divided by the number of times $q = \arg\min_{q\in Q} \|\mathbf{x} - q\|$. In this way, the number of data points used to train the kernel logistic regression classifier is limited, regardless of the size of $\mathcal{D}$, but redundant data is not discarded. The kernel logistic regression classifier is initialized with kernel size $\sigma = 0.03$, and a regularization parameter of $\lambda = 0.15$ with an L_2 penalty.

The expected entropy reduction is computed as a function of robot position over the entire workspace for the specific measurement model and robot FOV size and geometry. The measurement model uses parameters $\beta = 0.95$ and $\gamma = 0.01$. A cost function, $J = w_r \Delta H(Z_k) + w_b U_{rep}(\mathbf{x}_i)$ is constructed based on the EER and the Hilbert map, where w_r and w_b are user defined weighting terms. $U_{rep}(\mathbf{x}_i)$ is a potential field function similar to one used in traditional potential field methods [25](Chapter 7, Sec. 2.3), but altered to accept values in $[0, 1]$, as supplied by the Hilbert map. J depends on the EER and Hilbert map and is updated after each measurement. The probabilistic roadmap method uses J to sample roadmap points and only connects points that do not cross through the repulsion potential. Since measurements are constantly updating the Hilbert map, which updates J, this ensures that a robot will not collide with an obstacle.

The mapping performance of the VLSR system is represented by the conditional entropy of Y, in (2), given the set of measurements $\mathcal{M}$. By using the EER as a "reward", every robot always moves towards a portion of the ROI that reduces the conditional entropy most dramatically. The conditional entropy over time is depicted in Fig. 5. From this figure, it is clear that the conditional entropy of the map is decreasing over time and far exceeds the performance of robots moving randomly in the same workspace. As the robots take more and more measurements and update their Hilbert maps, they are attracted to areas where entropy will be reduced the most, and therefore, the entropy is certain to decrease while robots are guaranteed to avoid collisions.

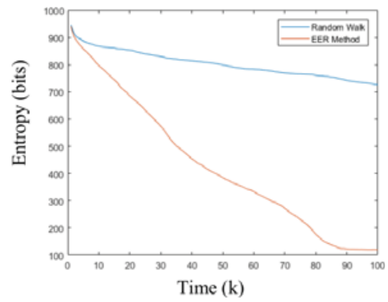


Fig. 5. Entropy of the map as the robots navigate the ROI

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a decentralized approach for mapping and information-driven path planning in Very Large Scale Robotic (VLSR) systems. Obstacle mapping is performed using a continuous probabilistic representation known as a Hilbert map and a novel Hilbert map fusion method is presented that quickly and efficiently combine information from many robots. The approach is demonstrated on a VLSR system with hundreds of robots that must map obstacles collaboratively over a large region of interest using onboard range sensors and no prior obstacle information. The results show that, through fusion and decentralized processing, the entropy of the map decreases over time and robot paths remain collision-free.

VII. ACKNOWLEDGMENTS

This research was funded by the Office of Naval Research, Code 321.

REFERENCES

[1] J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, no. 3, pp. 171–194, 1999.

[2] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, pp. 71–82, 2007.

[3] M. Steinberg, J. Stack, and T. Paluszkiewicz, "Long duration autonomy for maritime systems: challenges and opportunities," *Autonomous Robots*, vol. 40, no. 7, pp. 1119–1122, 2016.

[4] S. Thrun, M. Bennewitz, and W. Burgard, "Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots," *Robotics and Autonomous Systems*, vol. 41, no. 2, pp. 89–99, 2002.

[5] S. Ferrari, G. Foderaro, and P. Zhu, "Distributed optimal control of multiscale dynamical systems: A tutorial," *IEEE Contorl System Magazine*, p. in press, 2016.

[6] K. Rudd, G. Foderaro, P. Zhu, and S. Ferrari, "A generalized reduced gradient method for the optimal control of very large scale robotic (vlsr) systems," *IEEE Transactions on Robotics*, 2016.

[7] G. Foderaro, S. Ferrari, and M. M. Zavlanos, "A decentralized kernel density estimation approach to distributed robot path planning," 2012.

[8] G. Foderaro, P. Zhu, H. Wei, T. A. Wettergren, and S. Ferrari, "Distributed optimal control of sensor networks for dynamic target tracking," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 142–153, March 2018.

[9] G. Foderaro and S. Ferrari, "Necessary conditions for optimality for a distributed optimal control problem," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 4831–4838.

[10] S. Ferrari, G. Foderaro, P. Zhu, and T. A. Wettergren, "Distributed optimal control of multiscale dynamical systems: a tutorial," *IEEE Control Systems Magazine*, vol. 36, no. 2, pp. 102–116, 2016.

[11] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2016.

[12] F. Ramos and L. Ott, "Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1717–1730, 2015.

[13] C. E. Vido and F. Ramos, "From grids to continuous occupancy maps through area kernels," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1043–1048.

[14] S. T. OCallaghan and F. T. Ramos, "Gaussian process occupancy maps," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 42–62, 2012.

[15] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Gaussian process autonomous mapping and exploration for range sensing mobile robots," *arXiv preprint arXiv:1605.00335*, 2016.

[16] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 293–313, 2002.

[17] G. Zhang, S. Ferrari, and C. Cai, "A comparison of information functions and search strategies for sensor planning in target classification," *IEEE Transactions on systems, man, and Cybernetics*, vol. 42, no. 1, 2012.

[18] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal processing magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 1991.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. 223 Spring Street, New York, NY: Springer, 2009.

[21] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Quantized kernel least mean square algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, 2012.

[22] C. K. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*. MIT press, 2000, pp. 661–667.

[23] A. Rahimi, B. Recht *et al.*, "Random features for large-scale kernel machines." 2007.

[24] G. Zhang, S. Ferrari, and M. Qian, "An information roadmap method for robotic sensor path planning," *Journal of Intelligent and Robotic Systems*, vol. 56, pp. 69–98, 2009.

[25] J.-C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1991.