ECAI 2016
G.A. Kaminka et al. (Eds.)
© 2016 The Authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/978-1-61499-672-9-1256

Skeleton-Based Orienteering for Level Set Estimation

Lorenzo Bottarelli[1] and **Manuele Bicego**[1] and **Jason Blum**[2] and **Alessandro Farinelli**[1]

Abstract. In recent years, the use of unmanned vehicles for monitoring spatial environmental phenomena has gained increasing attention. Within this context, an interesting problem is *level set estimation*, where the goal is to identify regions of space where the analyzed phenomena (for example the PH value in a body of water) is above or below a given threshold level. Typically, in the literature this problem is approached with techniques which search for the most interesting sampling locations to collect the desired information (i.e., locations where we can gain the most information by sampling). However, the common assumption underlying this class of approaches is that acquiring a sample is expensive (e.g., in terms of consumed energy and time). In this paper, we take a different perspective on the same problem by considering the case where a mobile vehicle can continuously acquire measurements with a negligible cost, through high rate sampling sensors. In this scenario, it is crucial to reduce the path length that the mobile platform executes to collect the data. To address this issue, we propose a novel algorithm, called *Skeleton-Based Orienteering for Level Set Estimation* (SBOLSE). Our approach starts from the LSE formulation introduced in [10] and formulates the level set estimation problem as an orienteering problem. This allows one to determine informative locations while considering the length of the path. To combat the complexity associated with the orienteering approach, we propose a heuristic approach based on the concept of topological skeletonization. We evaluate our algorithm by comparing it with the state of the art approaches (i.e., LSE and LSE-batch) both on a real world dataset extracted from mobile platforms and on a synthetic dataset extracted from CO2 maps. Results show that our approach achieves a near optimal classification accuracy while significantly reducing the travel distance (up to 70% w.r.t LSE and 30% w.r.t. LSE-batch).

1 INTRODUCTION

The goal of environmental analysis is to collect information, generating an accurate model for a specific environmental process. For example when monitoring the quality of a body of water, operators might be interested in modeling how crucial parameters such as PH level, Dissolved Oxygen and temperature vary across time and space. These analyses usually require the collection of large data sets in harsh conditions, hence the use of mobile sensors such as unmanned ground vehicles (UGVs), unmanned aerial vehicles (UAVs) or autonomous surface vessels (ASVs). For an exhaustive overview on advancements and applications see [7].

A successful monitoring operation must acquire a sufficient amount of data to build an accurate model of the environmental phenomena of interest. However, the data collection process must consider limited resources such as energy and time. Therefore, it is crucial to carefully select measurement locations to acquire as much information as possible while minimizing energy and time required for data collection. An important aspect to consider is that the choice of the future locations to visit is dependent on previously collected data. Traditional, off-line sampling methods [6] do not represent a proper choice in this context – we refer to these processes as passive learning methods. Krause and Guestrin [14] survey advances to efficiently evaluate observation selection and illustrate the effectiveness of the approaches on environmental phenomena monitoring.

In contrast, active learning techniques [1, 16] aim to incrementally build a model of the environmental process during the data collection phase. Such techniques are very well suited for guiding mobile sensors and can be used to focus the data collection process on specific regions of the environment, so as to minimize the energy required for navigation [18].

In the simplest setting, the analysis process aims at collecting uniformly distributed data over a selected area. However, in many scientific and environmental monitoring applications, we are not interested in the precise value of the phenomena in every single location, rather we are interested in locating the regions of the space where the measurements exceed a given threshold level. This problem is typically referred to as the "level set estimation problem" [11]. For example, monitoring the water in a lake may require identification of the regions where the PH level of the water exceeds a critical value or to detect contours of biological and chemical plumes. Previous work on the level set estimation problem such as [5] focuses on a network composed by a combination of static and mobile sensors, while [20] focused on controlling the movement and communication of a sensor network without giving much attention to the choice of the sampling locations.

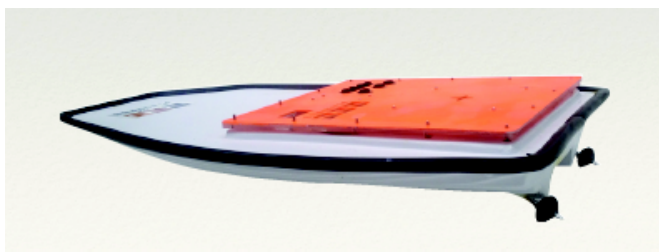

Figure 1. Platypus Lutra-T

In a more recent work on level set estimation [10] the data collection task is formalized as a classification problem with sequen-

[1] Computer Science Department, University of Verona, Italy name.surname@univr.it
[2] Robotics Institute, Carnegie Mellon University and Platypus LLC, Pittsburgh USA jasonblu@andrew.cmu.edu

tial measurements; the proposed LSE algorithm uses Gaussian Processes (GP) [17]) to estimate sampling points that reduce uncertainty around a given threshold level of the modeled function. The authors show a near-optimal classification for every region of the space with a reduced number of sampled locations compared to previous approaches. However, in the standard algorithm they do not explicitly take into account the path between the sampling locations, but they simply choose as the next point to be visited *the most informative* point, according to an ambiguity measure they derive from the Gaussian Process. They discuss the possibility to reduce the path length of the mobile sensor by proposing a *batch* version of their algorithm, where they determine a set of new sampling locations, again according to the the ambiguity measure derived from the Gaussian Process. Even if an efficient path between these points can be computed, again the choice of such points does not consider at all the distance the mobile sensor must cover to visit all such locations. Finally, more recently, [11] proposed a new receding horizon approach built on the LSE algorithm. Their method is designed for ASVs equipped with a probe that allows an aquatic sensor to be lowered into the water, and the algorithm uses a path planner to select sampling locations that lie on a feasible path for the probe within a predefined vertical transect plane.

In this paper, we also address the problem of level set estimation by using active learning techniques with sequential measurements. However, in our case, we have a further objective, namely we aim also at determining efficient paths for mobile sensors (instead of determining single sampling locations) so to optimize the data collection process. Specifically our techniques are motivated by the recent development of low-cost, small mobile platforms that can perform continuous-sampling in various body of waters (lakes, rivers and coastal areas). For example, consider the autonomous surface vessel shown in Figure 1. This platform is small (about 1 meter long and 50 cm wide) and it is equipped with various probes that can measure parameters such as PH, Dissolved Oxygen, temperature, and electrical conductivity with sampling rate between 1 and 10 Hz. In this setting the cost to perform a single measurement is negligible, and the most crucial issue for the data collection process is the battery lifetime for the vessel. In fact, to meet the space constraint of this platform, batteries have a limited capacity that results in constraints on total path length. Hence, in this work we aim at optimizing the total path length required by the agent to achieve near optimal classification of the analyzed regions, rather than the number of samples extracted during the executions (which is an important criteria for previous works).

In this perspective, the approach we propose formulates the Level Set Estimation problem as an Orienteering Problem (OP) [22]. In the general formulation of the OP we start with a set of locations, each one associated to a given score, and the goal is to choose the locations to visit so to maximize the sum of the scores while keeping the time (or the distance travelled) below a given budget. In the level set estimation problem we can see the sampling candidates as the locations to be visited, each one equipped with an *informativeness* score. For example we can use the already introduced ambiguity [10] to measure the value of the points. In this case, the LSE solution introduced in [10] simply chooses the most ambiguous point as the next point to be visited. The batch variant simply selects few points, again without considering the path. In contrast, by solving the OP in this setting, we are now trying to maximize the total informativeness of the points visited while keeping the travelled distance below a given budget, i.e. while explicitly considering the *cost* of the exploration (i.e., the length of the path). The OP is known to be NP-Hard. While we can use heuristic approaches to solve the problem, to perform on-line exploration we need to reduce the size of the orienteering instance (i.e. the number of possible locations to be visited). To this end, we propose a heuristic based on topological skeletonization, a process introduced in the image processing communitiy [9] aimed at reducing regions in a binary image to a thin (skeletal) representation — the *skeleton*, sometimes also called *medial axis*. In particular, we approximate the regions of the possible points to be visited (i.e. the unclassified points) with their skeleton, thus drastically reducing the size of the orienteering instance. As we will show in our empirical evaluation, this heuristic does not significantly affect the accuracy of the classification.

As a final comment, it is important to note that a related approach has been proposed in [19], where an orienteering-inspired technique has been applied to a related but different problem concerning information gathering. However, there are several important differences with respect to our work: i) the technique introduced in [19] does not aim at solving the level set estimation problem; ii) they propose an algorithm to solve the *submodular orienteering* problem (a particular kind of OP introduced by Chekuri and Pal [3]); iii) finally, our main objective is to determine efficient paths for mobile sensors (instead of determining single sampling locations) so to optimize the data collection phase and reduce the energy required in this process.

The main contributions of this paper to the state of the art are:

- We propose a novel algorithm that uses an orienteering formulation to solve the level set estimation problem.
- We propose a topological skeletonization as a heuristic to reduce the number of points on which we apply the orienteering algorithm.
- We empirically evaluated our algorithm comparing it to the current state of the art approach (i.e., LSE and LSE-batch [10]). Specifically, we consider a real-world dataset composed of measurements of the PH level of the water acquired with our mobile watercraft, and a synthetic dataset based on publicly available CO2 maps. Results show an advantage in terms of total travel distance, hence proving the feasibility of a skeleton-based orienteering approach to solve the level set estimation problem.

2 PROBLEM STATEMENT AND BACKGROUND

In the same spirit of [10], we formalize our approach for the level set estimation problem as an active learning problem, where we want to select next measurement locations so to optimize the information gathering process.

The environmental phenomena of interest is represented by an unknown scalar field. The area of the environment is discretised in a matrix where each element represents a location with an associated scalar value. For example, in practical application each element could be associated to a squared meter of the environment's surface and each element represents a sampling location x_i that must be classified according to a threshold level.

Specifically, given a threshold level h and a set of locations $D \subseteq \mathbb{R}^d$, we want to infer knowledge about the unknown scalar field $f : \mathbb{R}^d \mapsto \mathbb{R}$ and then to classify all points $x \in D$ into either $H = \{x \mid f(x) > h\}$ (called superlevel set) or $L = \{x \mid f(x) \leq h\}$ (called sublevel set). The scalar field is modeled with a Gaussian Process (GP) [17]. The problem then is to select the best set of locations x_i where to perform (noisy) measurements $y_i = f(x_i) + e_i$ while optimizing the total path length required for the mobile agents to analyze these points. Our proposed approach faces this problem using an Orienteering-based approach. Since the Orienteering problem

is computationally heavy, reducing the number of candidate points to be considered is crucial: in our approach this is done by exploiting a skeletonization-based heuristic. In the remainder of this section we will summarize the needed background: the Gaussian Processes, the formulation of the Level Set problem with Gaussian Processes – together with the solutions proposed in [10]–, the Orienteering problem and the Topological skeletonization.

2.1 Gaussian Processes

Gaussian Processes are a very important and widely used tool in machine learning [17]. In probability theory, a Gaussian Process (GP) is a statistical distribution and offers a way to model an unknown function without using parameters. In our case the function to be modeled is the scalar field f. A GP is completely defined by its mean function $\mu(x)$ that formulates prior knowledge about the values of the function f [3], and its covariance function (also called kernel function) $k(x, x')$ which encodes the smoothness properties of the function samples. A GP can then be denoted as $\mathcal{GP}(\mu(x), k(x, x'))$. At a given time t, we consider a set of noisy measurements $Y_t = \{y_1, y_2, \cdots, y_t\}$ taken at locations $X_t = \{x_1, x_2, \cdots, x_t\}$. We assume that $y_i = f(x_i) + e_i$ where $e_i \sim \mathcal{N}(0, \sigma_n^2)$ (i.e., measurements noise with zero mean) and we consider a GP prior $\mathcal{GP}(0, k(x, x'))$. Under these assumptions, the posterior over f is still a GP and its mean and variance can be computed as follows [17]:

$$\mu_t(x) = \mathbf{k}_t(x)^T(\mathbf{K}_t + \sigma_n^2\mathbf{I})^{-1}Y_t \tag{1}$$

$$\sigma_t^2(x) = k(x, x) - \mathbf{k}_t(x)^T(\mathbf{K}_t + \sigma_n^2\mathbf{I})^{-1}\mathbf{k}_t(x) \tag{2}$$

where $\mathbf{k}_t(x) = [k(x_1, x), \cdots, k(x_t, x)]^T$
and $\mathbf{K}_t = [k(x, x')]_{x,x' \in X_t}$

Using the above equations, we can update the GP with the new knowledge acquired through the observations (i.e., the set of measurements).

2.2 Level Set Estimation using Gaussian Processes

The formulation of the level set estimation problem using Gaussian Processes has been introduced in [10]. We have a set of sample locations D (that represents our area of interest) and we want to classify each location $x_i \in D$ into the two sets H or L previously defined by a threshold level h. This formulation uses the inferred mean (1) and variance (2) from the GP learnt on the scalar field to construct an interval:

$$Q_t(x) = \left[\mu_{t-1}(x) \pm \beta_t^{1/2}\sigma_{t-1}(x)\right] \tag{3}$$

for any point $x \in D$, where the parameter β acts as a scaling factor for the interval. The algorithm uses the intersection of all previous intervals to define a confidence interval

$$C_t(x) = \bigcap_{i=1}^{t} Q_i(x) \tag{4}$$

The classification of a point x into H or L depends on the position of its confidence interval with respect to the threshold level h. Intuitively if the entire interval lies above h, then with high probability $f(x) > h$ and x should belong to H. Similarly if the entire interval lies below h then x should belong to the L set. These conditions are relaxed by introducing an accuracy parameter ϵ which acts as a trade-off parameter between classification accuracy and number of samples required. Specifically:

$$H_t = \{x \mid min(C_t(x)) + \epsilon > h\} \tag{5}$$

$$L_t = \{x \mid max(C_t(x)) - \epsilon \leq h\} \tag{6}$$

This confidence interval allows the algorithm to either classify a point into the superlevel or the sublevel set. However, this classification scheme does not permit classifying all locations $x_i \in D$ at time t when $|Y_t| \ll |D|$, leaving a set of unclassified locations:

$$U_t = D \setminus (L_t \cup H_t) \tag{7}$$

Hence for the points that belong to U_t, we have to defer the decision until enough information is available. Given this formulation the goal is to select new sampling locations $x_i \in U_t$ to acquire new data and classify the points in U_t according to the equations (5) and (6).

2.2.1 The solutions of [10]: the LSE algorithm and the LSE batch algorithm

[10] proposed two solutions to this problem, both based on the concept of *ambiguity* of the candidate points. In particular, at a given iteration, the algorithm exploits the confidence intervals $C_t(x)$ derived from the Gaussian Process to calculate the ambiguity of all unclassified points:

$$a_t(x) = min\{max(C_t(x)) - h, h - min(C_t(x))\} \tag{8}$$

Given this concept, two solutions are presented in [10] to select the set of next points to be sampled:

1. *LSE*: in this case, the set of the next points to be sampled is composed by only one point, in particular the one with the highest ambiguity. Clearly, this solution does not take into consideration the distance of the chosen point from the current position, i.e. it does not consider the length of the path.
2. *LSE Batch*: this second algorithm is aimed at alleviating this problem, and opts for the selection of multiple locations to be sampled next. In particular, such locations are sequentially selected by considering both their ambiguity defined in eq. (8) and their joint mutual information, as derived from the Gaussian Process – for more information see [10] and [11]. An efficient path between such locations is then computed.

Although the main goal of the LSE Batch approach is to select multiple locations and to compute an efficient path between them, this algorithm is far in spirit from what we propose in this paper: actually, as in the LSE algorithm, the length of the path is not a variable explicitly considered in the choice of the points to be sampled next. The main reason for this is that in both cases their assumption is that the process of acquiring new data is costly. Therefore their main goal is to minimize the number of sampling locations.

2.3 Orienteering

The Orienteering Problem (OP) originates from the sport game of orienteering. In this game, the start and end points are specified along with other locations (i.e., checkpoints) which have associated score. The players aim at visiting as many checkpoints as possible in order to maximize the total score and have to reach the end point within a given time frame. The same problem can model several different contexts. For example, consider the problem in which a traveling salesperson has a set of cities which he could visit. Assuming that the

[3] This can be assumed to be zero without loss of generality

saleperson knows the number of sales he/she can expect in each city, the goal is then to plan a route so as to maximize the total number of sales while keeping the total length of such route within a given budget (i.e., the maximum distance he/she can travel in one day).

More formally, the OP can be formulated in the following way: given a set of N locations each with a score $S_i \geq 0$, a starting location 1 , an ending location N and the travel time t_{ij} for all couples of locations i and j (with $i \neq j$), the goal is to determine a path, limited by a given budget T_{max}, that visits a subset of these locations, in order to maximize the total collected score.

The OP can intuitively be defined with the aid of a weighted undirected graph $G = (V, E)$ where $V = \{v_1, \ldots, v_N\}$ is the set of nodes (locations) and E is the set of edges. In this formulation the nonnegative score S_i of location i is associated with a vertex $v_i \in V$ and the travel time t_{ij} between location i and j is associated with each edge $e_{ij} \in E$. The OP consists of determining a Hamiltonian path over a subset of V, including the start node (v_1) and end node (v_N), and having a length not exceeding the bound T_{max}, in order to maximize the total collected score.

Therefore, the orienteering is a combination of node selection and shortest path computation between these nodes, hence it can be casted as a combination of the Knapsack Problem (KP) and the Traveling Salesman Problem (TSP) problems [4], where the KP goal is to maximize the total score collected while the TSP aims at minimizing the travel distance. This formulation is typically referred to as a generalized travelling salesman problem (GTSP) [8]. The orienteering problem is known to be an NP-hard problem, as it contains the well known traveling salesman problem as a special case.

This NP-hard problem arises in routing and scheduling applications and it is also known as the selective traveling salesperson problem ([15], [21]) or the maximum collection problem ([13]). A number of practical applications has been modeled as OP and many heuristic approaches have been developed to combat the inherent complexity of the OP. In most cases, the orienteering is defined as a path to be found between distinct locations, rather that a circuit where $v_1 \equiv v_N$. In some applications, however, v_1 can coincide with v_N but the difference between both formulation is not significant. For a general review we suggest the survey proposed by Vansteenwegen et al. [22].

2.4 Topological Skeletonization

In digital image processing and shape analysis, *skeletonization* is a process for reducing regions in a binary image to a thin (skeletal) representation while throwing away most of the original pixels (see example in Figure 2). The skeleton preserves and usually emphasizes the geometrical properties of the shape, such as its connectivity, topology, length and direction.

Skeletonization was first introduced by Blum [2], and it can be described by using an intuitive model of fire propagation. If one "sets fire" at all points on the boundary of an image the skeleton forms at the points in the region where two or more "fires" meet. This intuitive description has several different mathematical definitions and in the relevant literature it is sometimes referred to as *medial axis* or *thinning* [9].

Skeletonization has been used in several applications ranging from computer vision to image analysis and digital image processing. There are many algorithms that are tailored for different application contexts. Such approaches mainly vary in run time and properties of the produced skeleton (e.g., whether it is a connected component or not), however they all significantly compress the input. In this paper we are interested in skeletonization mainly to reduce the number of points that we must consider when planning the route for the robotic platforms. Hence, we select a basic approach based on morphological operators (see Section 4.3)

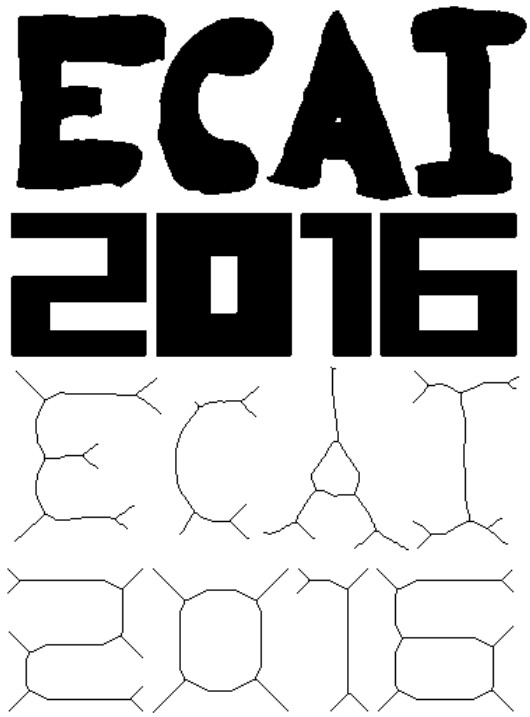

Figure 2. Example of a topological skeletonization applied to an image. On top the binarized input image and on the bottom the skeletonized version.

3 SBOLSE ALGORITHM

Using both the LSE solutions proposed in [10], the mobile sensor is guided toward the most informative points, without taking into account the path of the mobile sensor. For example, the LSE algorithm assumes that the mobile sensor moves from the current position to the next selected location following a straight line. Another issue is that the measure is collected only at the final location, without considering all the points traversed by the sensor along its path. On the contrary, here we consider applications where measuring devices can provide data while the robotic platform is moving. For example, the mobile platforms we use here are equipped with probes that measure various parameters (e.g., the PH level) with a given frequency while the platform is moving. In this scenario, our goal is then minimizing the path length while collecting as much information as possible to correctly classify all points $x_i \in D$.

In what follows we present our Skeleton-Based Orienteering for Level Set Estimation (SBOLSE) algorithm, which starts from the LSE framework but is specifically designed for continuous sampling sensors in which the cost required to extract a sample is negligible but it is necessary to optimize the total path of the mobile platform to minimize battery consumption.

The proposed algorithm considers the knowledge about unclassified locations $x_i \in U_t$ to build an OP instance and to select a sequence of visit locations (i.e., a path). The algorithm aims at optimizing the information that can be acquired along the path while

meeting the budget on the travel distance. Next, we propose a heuristic approach based on the topological skeletonization to combat the computational complexity associated with the OP, empirically showing that the classification accuracy does not suffer a significant degradation while greatly reducing the computation time.

Algorithm 1 SBOLSE algorithm

Input: set D, threshold h, accuracy parameter ϵ, prior known data $X \subset D$, starting location x_{start}
Output: sets H and L

1: $t \leftarrow 0$
2: $x_0 \leftarrow x_{start}$
3: $H_0 \leftarrow \emptyset, L_0 \leftarrow \emptyset, U_0 \leftarrow D$
4: **while** $H_t \cup L_t \neq D$ **do**
5: $t \leftarrow t + 1$
6: Compute GP posterior $\mu(x)$ and $\sigma^2(x)$ for all $x \in U_t$
7: Classify and update H_t, L_t, U_t according to LSE [10]
8: $x_c \leftarrow$ current position
9: $G \leftarrow buildGraph(x_c, U_t)$
10: $path \leftarrow orienteeringStep(G, budget)$
11: Execute $path$
12: $H \leftarrow H_t, L \leftarrow L_t$

The pseudo-code of Algorithm 1 describes the steps of our SBOLSE approach. The algorithm maintains three sets of points: the current superlevel H_t and sublevel L_t sets, as well as the set of unclassified points U_t. At each iteration t we update the Gaussian Process posterior by integrating the new information gathered at the preceding iteration (line 6). Then we compute the confidence intervals $C_t(x)$ for each point $x \in U_t(x)$, classify them in one of the three sets and then compute the sequence of locations to be visited. To compute such sequence of locations we consider the ambiguity defined by equation (8) of the unclassified points and build an OP instance. Specifically, in line 9 we create a graph from the unclassified points U_t (Algorithm 2) and then compute a path (line 10) using the orienteeringStep procedure (Algorithm 3). The algorithm terminates when $H_t \cup L_t = D$, i.e. when all points are classified and thus $U_t = \emptyset$. Note that during the execution of the path (Algorithm 1, line 11) if the platform moves over locations not yet analyzed but already classified according to LSE [10], these are evaluated and, in case, re-classified considering newly acquired data.

3.1 Building the graph

In the buildGraph procedure we take all the unclassified locations U_t and we build an un directed weighted graph, where all locations are connected. This graph will then be used in the orienteering procedure.

As shown in Algorithm 2, the first node of the graph represents the current location of the mobile sensor (line 1). This location defines the starting position for the orienteering solver. Subsequently we build the nodes set V and the edges set E. The function $w(\cdot)$ denotes respectively the weight of a node or the weight of an edge. The weight of a node $w(v_i)$ (line 7) is the ambiguity measure (equation 8) of the location that the node represents. The weight of the first node is an exception as this represents the current position of the mobile sensor and hence the location has been visited and classified. The weight of the edges $w(e_{ij})$ (line 13) is the travel distance between the locations represented by the nodes v_i and v_j.

Algorithm 2 buildGraph procedure

Input: current position x_c, unclassified elements U_t
Output: weighted undirected graph G

1: $V \leftarrow v_1 \equiv x_c$
2: $w(v_1) \leftarrow 0$
3: $n \leftarrow 1$
4: **for all** $x_i \in U_t$ **do**
5: $n \leftarrow n + 1$
6: $V \leftarrow V \cup v_n \equiv x_i$
7: $w(v_n) \leftarrow a(x_i)$
8: $E \leftarrow \emptyset$
9: **for all** $v_i \in V$ **do**
10: **for all** $v_j \in V$ **do**
11: **if** $v_i \neq v_j$ **then**
12: $E \leftarrow E \cup e_{ij}$
13: $w(e_{ij}) \leftarrow dist(v_i, v_j)$
14: $G \leftarrow (V, E)$

3.2 Orienteering Step

Algorithm 3 orienteeringStep procedure

Input: graph $G = (V, E)$, budget B
Output: $bestPath$

1: $bestPath \leftarrow \emptyset$
2: $bestPathValue \leftarrow 0$
3: **for** i in range$(2, |V|)$ **do**
4: **if** $dist(v_1, v_i) \leq budget$ **then**
5: $path \leftarrow orienteeringHeuristic(G, v_1, v_i, B)$
6: **if** $value(path) > bestPathValue$ **then**
7: $bestPath \leftarrow path$
8: $bestPathValue \leftarrow value(path)$

In the orienteeringStep procedure we use the previously built undirected weighted graph G and consider this as the input to the orienteering problem. In particular we have a fixed starting point (i.e. the current location of the mobile agent), but we do not have an ending point (which is required in the classical formulation of the orienteering problem). It makes clearly sense that the starting point should be equal to the destination point, however in the classic orienteering problem the rewards of every node are fixed. In our case, rewards changes during the execution of the procedure since the information value of every location decreases while the robot acquires new data. Hence, making a single run of orienteering would not take into account the adaptivity required by such scenario. Therefore, we iterate the process for smaller segments. The choice of the length (budget) of these segments allows a tradeoff between adaptivity and computation requirements. To choose the destination we perform an orienteering heuristic multiple times (Algorithm 3, line 5), assuming as destination every unclassified location in the graph that is reachable with the given budget. Every time we solve an orienteering instance we obtain a new path. The procedure keeps track of the best discovered one and returns this as final route to be executed from the SBOLSE algorithm. Specifically with $value(path)$ (line 6 and 8) we indicate the summation of the nodes' weights in that route, that is $value(path) = \sum_{v_i \in path} w(v_i)$. Since the orienteering problem aims at maximizing the score within a given travel budget, using this procedure we obtain a path that maximizes the information selected among the unclassified locations for the level set estimation problem. In this work we did not focus on the computational efficiency

but rather on a novel formalization of the problem. The choice of having a completely connected graph and repeating the orienteering step n-times for every possible reachable destination represents the simplest choice for formalizing this problem. Improving these aspects to reduce the computation required is matter of future work. Current times do not prevent real-time operations.

3.3 Skeletonization

In most practical applications of level set estimation the input is a set of dense points that must be classified. Specifically, when we start the data acquisition process, we must consider the entire surface of the selected portion of the environment. These data are typically discretized and organized in a matrix where each entry represents a small portion of the surface (i.e., a square of 50 centimeters in our experiments).

Now, given some smoothness of the environmental phenomena, locations with higher classification uncertainty usually cluster into areas where the unknown scalar field has high probability to cross the threshold level. Considering all such points could be considered redundant. This motivates the use of a topological skeletonization algorithm to compress the input.

Specifically, we consider the matrix containing the information about the ambiguity measure (eq. 8) of the unclassified points U_t as a binary image, where unclassified points are set to 1 and classified points are 0. We then apply a topological skeletonization to such image, and we maintain as interesting points to be classified only the points of the skeleton. This greatly reduces the number of locations that we must consider in the $buildGraph$ procedure presented in section 3.1 (see the example in Figure 3).

3.4 Theoretical analysis

For what concerns the theoretical analysis of our approach, notice that Gotovos et. al. with Theorem 1 in [10] prove the convergence of the LSE algorithm. Even though the selection procedure of our SBOLSE algorithm differs from LSE, we used the same classification rules (Algorithm 1, lines 6-7). As in LSE, our technique iterates until every point is classified with respect to a threshold level h and with an accuracy parameter ϵ, hence we can ensure the convergence of the SBOLSE algorithm.

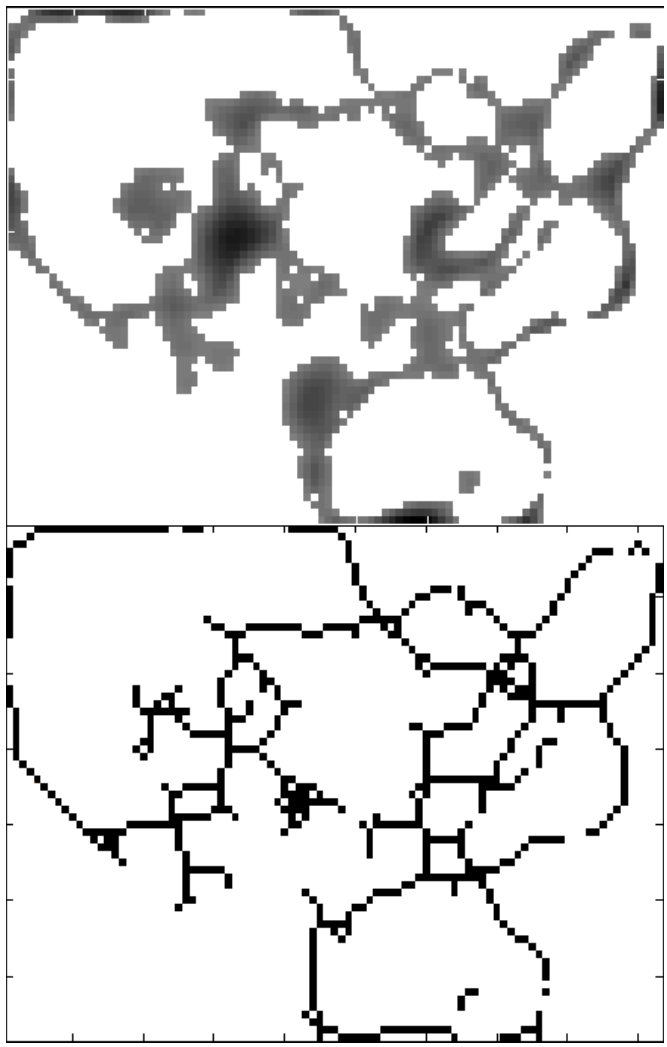

Figure 3. Example of the topological skeletonization applied to the data matrix containing the ambiguity measure for the unclassified points U_t. On top the data matrix before the skeletonization, a darker color corresponds to an higher value of ambiguity. On the bottom the skeletonized version.

4 EMPIRICAL EVALUATION

In this section we now present an empirical evaluation of our proposed technique. First (in sections 4.1 and 4.2) we present the comparison of our technique with state of the art approaches on two different datasets. Then, in section 4.3, we analyze the results of the topological skeletonization applied to the ambiguity measure (as explained in section 3.3), showing that this heuristic significantly reduces the size for the orienteering instances while preserving the overall classification accuracy.

Specifically we compare: i) Our SBOLSE technique with a variant of the LSE algorithm [10], which we designed to meet the continuous sampling setting; ii) The batch version of the two approaches tested in i). More specifically, the algorithms we compare are:

- **SBOLSE:** Our algorithm as explained in section 3.
- **CS:** This algorithm is a variant of the LSE as described in [10]. The classification and sample selection is the same except that all locations on the straight line between the last position and the next selected location are analyzed so to simulate a continuous sampling sensor.
- **CS$_{bXX}$:** This algorithm is a variant of the LSE$_{batch}$ as described in [10]. Similarly to CS we analyze all locations in the straight line between one location and the next one to simulate a continuous sampling sensor, XX identifies the dimension (i.e. the number of elements) of the batch. We performed tests with batches of different sizes and determined as optimal value a batch size of 30. We did not observe a significant reduction of the total path length with batches of size larger than 30.

Regarding our SBOLSE algorithm, we implemented a simple orienteering heuristic inspired by the *center of gravity* technique as proposed by Golden et al. [8]. We performed the skeletonization with a basic algorithm, based on morphological operators, as implemented in the MATLAB function `bwmorph`. We used the F_1-score as in [10] to assess the classification accuracy for all the results of our tests. The F_1-score is often used in information retrieval for measuring binary

classifications. In our case we consider the superlevel set as the positives and the sublevel set as the negatives elements.

4.1 Real data experiments

The first dataset consists of real-world data relative to the PH level extracted from waters of the Persian Gulf near Doha (Qatar) using the boat in Figure 1. The data collected has been aggregated in a 68×93 matrix where each element represents a sampling location x_i that must be classified according to a threshold level. In particular, each element represents 0.5 square meters of the analyzed surface. The value associated to that element is the average of all the samples extracted in that portion of the surface. An example of this dataset is shown in Figure 4.

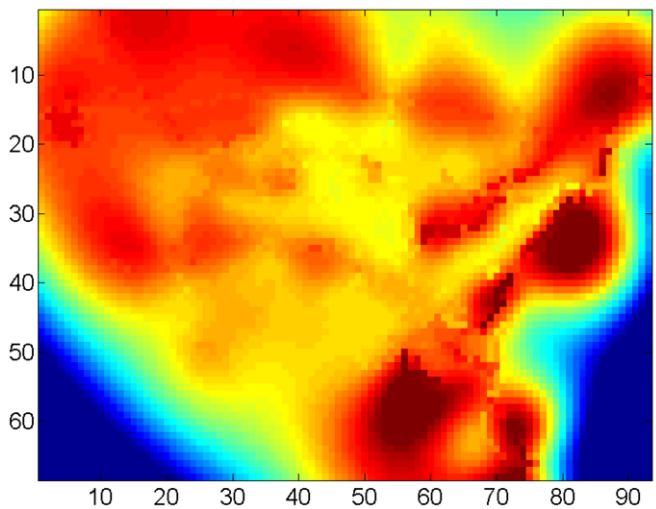

Figure 4. Scalar field of the real-world dataset, i.e. the PH level of waters extracted in the Persian Gulf near Doha (Qatar).

On this first set of experiments we considered three different thresholds to classify the PH scalar field, specifically 7.40, 7.42 and 7.44. We performed some test to determine the best parameter setting for β and ϵ in such a way that the classification accuracy is high for all the algorithms we compare.

Following standard approaches in the literature (e.g., [10]), we start from ten random initial prior composed by 10% of the elements of the matrix, for a total of 30 instances per algorithm. The priors were used to fit the hyperparameters of an isometric Matérn-3 covariance function. The results of this set of experiments are shown in Table 1. For a fair comparison the budget for the orienteering subroutine has been set to the same length that would have been traveled by the standard LSE algorithm, that is the distance between the current location and the selected sampling point. In this way both methods consider the knowledge about the environment with a new GP update after the same amount of traveled distance.

We can observe that the F_1-score, (which indicates the accuracy of the classification) is higher than 97% for all the algorithms. Regarding the total path length, our SBOLSE algorithm performs very well, reducing the path required by the mobile sensor by about 70% compared to the standard LSE algorithm proposed in [10]. Also the comparison with the batch version of the LSE algorithm still show an advantage by about 32% in the total path length.

Table 1. Results of F_1-score and total path length using the real world PH dataset extracted from waters of the Persian Gulf near Doha (Qatar), $\overline{x}$ is the average of all experiments and $SE_{\overline{x}}$ is the standard error of the mean.

	F_1-score		Path Length	
	$\overline{x}$	$SE_{\overline{x}}$	$\overline{x}$	$SE_{\overline{x}}$
SBOLSE	97.23	0.066	**473.6**	6.203
CS	98.22	0.039	1560.8	18.582
CS_{b30}	97.54	0.061	687.9	14.296

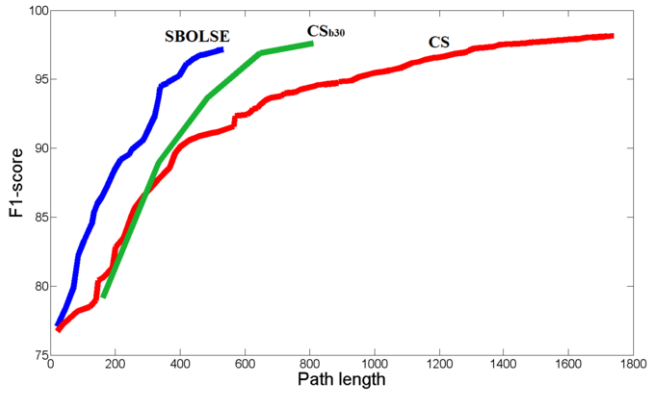

Figure 5. Runtime F_1-score comparison on the typical example instance of the real dataset, varying the path length between SBOLSE, CS and CS_{b30} algorithms.

4.2 Synthetic CO2 dataset experiments

The second dataset consists of 10 matrices 60×179. In this case we assume that each element represents 1 square meters of the surface. These matrices have been extracted from the color channels of portions of CO2 analysis images[4] to obtain a dataset with a scalar field consistent with a typical environmental topology. One example of a matrix from the dataset is presented in Figure 6. The main purpose of this dataset is to test our technique on bigger matrices (more that 10,000 elements to classify) and to assess the quality of the algorithm on data different than a scalar field extracted from a body of water.

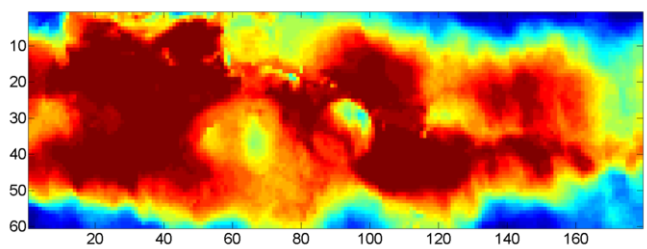

Figure 6. Example of one of the synthetic scalar fields extracted from a CO2 analysis map.

In our experiments we considered a threshold equals to 85% of the maximum value in the scalar field. Also in this case we determined

[4] http://oco.jpl.nasa.gov/galleries/gallerydataproducts/

the best parameter setting and then performed tests with all the three algorithms previously described with five random initial prior composed by 10% of the elements of the matrix, for a total of 150 experiments. Again the priors were used to fit the hyperparameters of an isometric Matérn-3 [17] covariance function. The results of these experiments are shown in table 2. We can observe that these are similar to what was obtained in the real-world dataset. Specifically we obtain a reduction of about 75% of the path length compared to the standard LSE algorithm and about 25% compared to the batch version.

Table 2. Results of F_1-score and total path length using the synthetic CO2 dataset, $\overline{x}$ is the average of all experiments and $SE_{\overline{x}}$ is the standard error of the mean.

	F_1-score		**Path Length**	
	$\overline{x}$	$SE_{\overline{x}}$	$\overline{x}$	$SE_{\overline{x}}$
SBOLSE	97.99	0.100	**1355.6**	26.156
CS	98.66	0.071	5588.1	136.864
CS_{b30}	98.25	0.089	1782.7	34.052

4.3 Topological skeletonization results

This test aims at computing some statistics on the unclassified points U_t during the execution of the algorithm before and after the operation of topological skeletonization. In particular, we empirically show that this heuristic significantly reduces the amount of points that need to be analyzed during the orienteering step. In Figure 7 we can observe the average reduction in the number of unclassified points after the skeletonization on a typical example instance of the real dataset. This directly translates in space reduction of the graph G used to perform the orienteering operation.

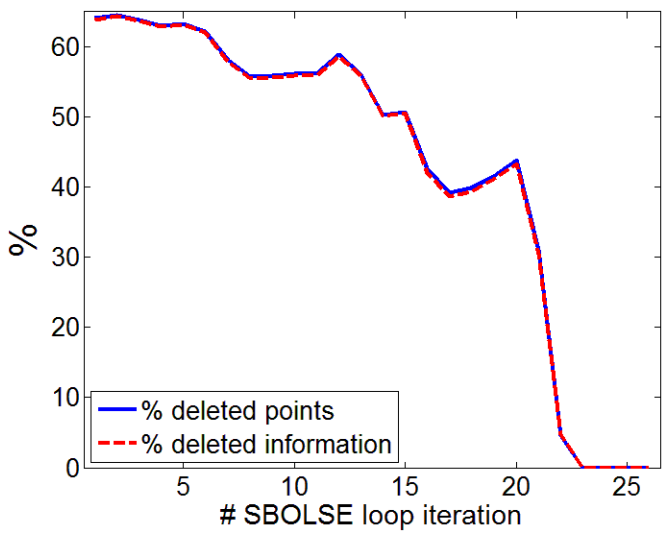

Figure 7. Reduction in the number of unclassified points and information after the topological skeletonization, during the execution of the SBOLSE algorithm

As reported in previous section, to perform the skeletonization we used a basic algorithm based on morphological operators as implemented in MATLAB function `bwmorph`. Although the operation is very simple and fast and with such a technique we do not take into account the amount of information in the deleted points, in Figure 7 we can observe that after this operation, statistically the percentage of points and information deleted are similar. This is due to the fact that, given the typical environmental phenomena, in the area representing the unclassified data, the highest amount of available information (points with the highest ambiguity value) is concentrated in the middle of the area itself. This suggests that the skeletonization is a good heuristic to apply in order to discard some of these points.

The applied algorithm based on morphological operators in some of the cases reduces the area giving a skeletonization centered where the information is concentrated (see Figure 3), whereas in other locations this is not the case. However this observation leaves open the possibility of applying different skeletonization techniques that better preserves this property, further increasing the usefulness of our technique. For example many definitions of skeleton make use of the concept of distance function [12], which is a function that for each point inside a shape gives its distance to the closest point on the boundary. Further investigations could use a similar concept in order to generate a skeletonization based on the amount of data present in the area.

We now show a comparison between two versions of our SBOLSE algorithm, with and without skeletonization of the orienteering instances, specifically we performed the experiments on the real world dataset. Results in table 3 show that the application of the skeletonization heuristic does not significantly influence the classification quality. At the same time, however, this method allows us to greatly reduce the complexity of the orienteering heuristic as previously shown in Figures 7

Table 3. Comparison between our SBOLSE algorithm using the orienteering subroutine with and without the topological skeletonization on the unclassified data U_t

	F_1-score		**Path Length**	
	$\overline{x}$	$SE_{\overline{x}}$	$\overline{x}$	$SE_{\overline{x}}$
with	97.37	0.152	449.0	7.414
without	97.70	0.075	525.0	13.891

5 CONCLUSION

In this paper we proposed a new algorithm (i.e., SBOLSE) for the level set estimation problem, considering mobile watercraft equipped with continuous-sampling sensors. Our technique formulates the level set estimation problem as an orienteering problem where the ambiguity about the classification of a location represents the score in the orienteering formulation. Our SBOLSE algorithm implements an orienteering heuristic solution as a subroutine to select an informative path that meets a given travel budget. Moreover, we present an approach based on the topological skeletonization to reduce the size of the orienteering instance we solve, allowing for on-line classification. Results show that our approach significantly outperforms the current state of the art algorithms for the level set estimation problem (i.e., LSE and LSE-batch) in terms of total travel distance, while maintaining a near-optimal classification quality.

ACKNOWLEDGEMENTS

This work was supported by the European Unions Horizon 2020 research and innovation programme under grant agreement No 689341. This work reflects only the authors' view and the EASME is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] M. A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. J. Kaiser, M. Hansen, G. J. Pottie, M. Srivastava, and D. Estrin, 'Call and response: Experiments in sampling the environment', in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pp. 25–38, New York, NY, USA, (2004). ACM.

[2] Harry Blum, 'A Transformation for Extracting New Descriptors of Shape', *Models for the Perception of Speech and Visual Form*, 362–380, (1967).

[3] Chandra Chekuri and M. Pal, 'A recursive greedy algorithm for walks in directed graphs', in *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pp. 245–253, (Oct 2005).

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, MIT Press, third edn., 2009.

[5] K. Dantu and G. Sukhatme, 'Detecting and tracking level sets of scalar fields using a robotic sensor network', in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3665–3672, (April 2007).

[6] A. Dhariwal, B. Zhang, B. Stauffer, C. Oberg, G. S. Sukhatme, D. A. Caron, and A. A. Requicha, 'Networked aquatic microbial observing system', in *International Conference on Robotics and Automation*, pp. 4285–4287, Orlando, FL, (May 2006). IEEE.

[7] M. Dunbabin and L. Marques, 'Robots for environmental monitoring: Significant advancements and applications', *Robotics Automation Magazine, IEEE*, **19**(1), 24–39, (March 2012).

[8] Bruce L. Golden, Larry Levy, and Rakesh Vohra, 'The orienteering problem', *Naval Research Logistics (NRL)*, **34**(3), 307–318, (1987).

[9] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing (3rd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[10] Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause, 'Active learning for level set estimation', in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 1344–1350. AAAI Press, (2013).

[11] G. Hitz, A. Gotovos, F. Pomerleau, M.-E. Garneau, C. Pradalier, A. Krause, and R.Y. Siegwart, 'Fully autonomous focused exploration for robotic environmental monitoring', in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 2658–2664, (May 2014).

[12] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[13] S. Kataoka and S. Morito, 'An algorithm for single constraint maximum collection problem', *Journal of the Operations Research Society of Japan*, **31**(4), 515–31, (1988).

[14] Andreas Krause and Carlos Guestrin, 'Near-optimal observation selection using submodular functions', in *National Conference on Artificial Intelligence (AAAI), Nectar track*, (July 2007).

[15] Gilbert Laporte and Silvano Martello, 'The selective travelling salesman problem', *Discrete Applied Mathematics*, **26**(2), 193 – 207, (1990).

[16] M. Rahimi, R. Pon, W. J. Kaiser, G. S. Sukhatme, D. Estrin, and M. Srivastava, 'Adaptive sampling for environmental robotics', in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 4, pp. 3537–3544 Vol.4, (April 2004).

[17] C. E. Rasmussen and Williams C. K. I., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, USA, 2006.

[18] Aarti Singh, Robert Nowak, and Parmesh Ramanathan, 'Active learning for adaptive mobile sensing networks', in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, IPSN '06, pp. 60–68, New York, NY, USA, (2006). ACM.

[19] Amarjeet Singh, Andreas Krause, and William J. Kaiser, 'Nonmyopic adaptive informative path planning for multiple robots', in *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pp. 1843–1850, San Francisco, CA, USA, (2009). Morgan Kaufmann Publishers Inc.

[20] S. Srinivasan, K. Ramamritham, and P. Kulkarni, 'Ace in the hole: Adaptive contour estimation using collaborating mobile sensors', in *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, pp. 147–158, (April 2008).

[21] T. Thomadsen and T. Stidsen, 'The quadratic selective travelling salesman problem', Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 305, DK-2800 Kgs. Lyngby, (2003).

[22] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden, 'The orienteering problem: a survey', *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*, **209**(1), 1–10, (2011).