Robotics and Autonomous Systems 61 (2013) 1258–1276

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

A survey on coverage path planning for robotics

Enric Galceran *, Marc Carreras

University of Girona, Underwater Robotics Research Center (CIRS), Pic de Peguera, 13, 17003 Girona, Catalonia, Spain

HIGHLIGHTS

- A comprehensive literature survey on the coverage path planning problem is presented.
- This problem is key to many robotic tasks, and no updated surveys are available.
- A comprehensive review of the most relevant works in the problem is reported.

ARTICLE INFO

Article history:
Received 4 February 2013
Received in revised form
5 August 2013
Accepted 13 September 2013
Available online 20 September 2013

Keywords:
Coverage path planning
Path planning
Motion planning

ABSTRACT

Coverage Path Planning (CPP) is the task of determining a path that passes over all points of an area or volume of interest while avoiding obstacles. This task is integral to many robotic applications, such as vacuum cleaning robots, painter robots, autonomous underwater vehicles creating image mosaics, demining robots, lawn mowers, automated harvesters, window cleaners and inspection of complex structures, just to name a few. A considerable body of research has addressed the CPP problem. However, no updated surveys on CPP reflecting recent advances in the field have been presented in the past ten years. In this paper, we present a review of the most successful CPP methods, focusing on the achievements made in the past decade. Furthermore, we discuss reported field applications of the described CPP methods. This work aims to become a starting point for researchers who are initiating their endeavors in CPP. Likewise, this work aims to present a comprehensive review of the recent breakthroughs in the field, providing links to the most interesting and successful works.

1. Introduction

Coverage Path Planning (CPP) is the task of determining a path that passes over all points of an area or volume of interest while avoiding obstacles. This task is integral to many robotic applications, such as vacuum cleaning robots [1], painter robots [2], autonomous underwater vehicles creating image mosaics [3], demining robots [4–6], lawn mowers [7,8], automated harvesters [9,10], window cleaners [11] and inspection of complex underwater structures [12], just to name a few.

In one of the earliest works on CPP found in the literature, [7] defined the requirements a robot must meet to perform a coverage operation. Albeit the target application in the aforementioned paper is a mobile robot moving in a flat 2-dimensional environment, the same criteria are applicable to other coverage scenarios. The requirements are as follows:

1. robot must move through all the points in the target area covering it completely.
2. robot must fill the region without overlapping paths.
3. continuous and sequential operation without any repetition of paths is required.
4. robot must avoid all obstacles.
5. simple motion trajectories (e.g., straight lines or circles) should be used (for simplicity in control).
6. an "optimal" path is desired under available conditions.

However, it is not always possible to satisfy all these criteria in complex environments. Therefore, sometimes a priority consideration is required.

The CPP problem is related to the covering salesman problem, a variant of the well-known traveling salesman problem where, instead of visiting each city, an agent must visit a neighborhood of each city [13]. However, in CPP, the agent must pass over all points in the target area in contrast to visiting all the neighborhoods. Since the traveling salesman problem is NP-hard, the computational time required to solve the problem increases drastically when the dimension of the problem increases. Actually, finding a path to cut all the grass of a given region covered by grass, which is known as the "lawnmower problem", is proven to be NP-hard [14]. Notice that, the lawnmower problem does not account for obstacles. In fact, even the basic path planning problem, known as the "piano mover's problem", of finding a collision-free path from a start configuration to a goal configuration is shown to be PSPACE-hard, which implies NP-hard [15,16]. Two additional, similar problems

* Corresponding author.
E-mail addresses: enricgalceran@eia.udg.edu, enricgalceran@gmail.com (E. Galceran), marc.carreras@udg.edu (M. Carreras).

0921-8890/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.robot.2013.09.004

related to CPP are the art gallery problem and the watchman route problem. The art gallery problem calls for the minimum number of guards needed to station in a polygonal gallery so that each point in the gallery is visible to at least one guard [17]. The watchman route problem calls for the shortest route from a given point back to itself so that each point in a given space is visible from at least one point along the route [18]. Simple cases of the watchman route problem such as covering the interior of simple polygons can be achieved in polynomial time [19]. But, in general, both the art gallery problem and the watchman route problem are NP-hard [17,18]. Some coverage algorithms we discuss in Section 8.6 approach CPP as the art gallery problem and the watchman route problem.

Coverage algorithms can be classified as heuristic or complete depending on whether or not they provably guarantee complete coverage of the free space. Independently, they can be classified as either off-line or on-line. This classification was originally proposed by Choset [20]. Off-line algorithms rely only on stationary information, and the environment is assumed to be known in advance. However, assuming full prior knowledge of the environment might be unrealistic in many scenarios. On the other hand, on-line algorithms do not assume full prior knowledge of the environment to be covered and utilize real-time sensor measurements to sweep the target space. Thus, these later algorithms are also called sensor-based coverage algorithms.

In certain scenarios, a valid approach to solve the problem is to randomize. This is an approach that some floor-cleaning robots rely on: if the floor is swept randomly for long enough, it should become cleaned. Examples of commercial floor-cleaning robots based fully or partially on this strategy are the RC3000 by Karcher, Trilobite by Electrolux and Roomba by iRobot [21]. There are advantages to this approach, the main one being that no complex sensors for localization nor expensive computational resources are needed. However, for covering vast areas, and especially for underwater or aerial robotics operations which deal intrinsically with a 3-dimensional space, it is difficult to think that a randomized "algorithm" could be usable, as the cost of operating the vehicle (energy and time) would be unaffordable.

A considerable body of research addressing the CPP problem can be found in the literature. Choset [20] presented a survey on coverage path planning methods. However, no updated surveys on CPP reflecting the recent advances in the field have been presented in the past ten years. In this paper, we present a review of the most successful CPP methods, focusing in the achievements made in the past decade. Furthermore, we discuss reported field applications of the described CPP methods. This work aims to become a starting point for researchers who are initiating their endeavors in CPP. Likewise, this work aims to present a good review of the recent breakthroughs in the field, providing links to the most interesting and successful works.

Since most CPP algorithms decompose the target space in subregions (called cells) to achieve coverage, [20] classified coverage algorithms according to the type of decomposition used. Hence, his taxonomy comprises heuristic and randomized approaches (which typically do not use a representation of the environment and therefore neither use a decomposition), and approximate, semi-approximate and exact cellular decompositions. However, we argue that qualitatively different approaches can be distinguished among these categories. Thus, the sections in this article do not bear a one-to-one correspondence with Choset's taxonomy, but rather reflect the common underlying ideas used in the discussed approaches. Nonetheless, Choset's taxonomy is commonly used throughout the literature, and hence we provide the corresponding Choset's classification for the methods reviewed.

Resulting from these considerations, the remainder of this article is organized as follows. Section 2 reviews classical cellular decomposition algorithms which set the fundamentals of the cellular decomposition approach. Section 3 discusses the cellular decomposition based on critical points of Morse functions, which can deal with a more general class of obstacles. Section 4 presents a coverage approach based on detection of natural landmarks. A coverage algorithm for robots equipped with only contact sensors operating in rectilinear environments is discussed in Section 5. Methods based on a grid representation of the space are reviewed in Section 6. Work addressing the coverage problem in environments that can be represented as a graph, such as a street or road network, is briefly reviewed in Section 7. In Section 8 several methods for covering 3-dimensional environments are presented. Section 9 reviews some methods for achieving optimal coverage, and Section 10 focuses on methods that intend to reduce the accumulation of localization error while performing coverage. In Section 11 several multi-robot CPP methods are reviewed. Finally, concluding remarks including a summary table and directions for further research are given in Section 12.

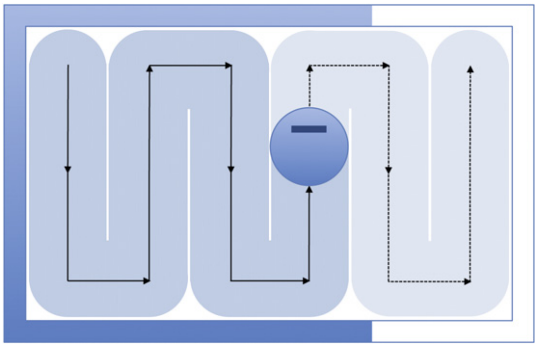


Fig. 1. Typical zigzag path. Shaded area indicates the area covered already (darker) and the area that will be covered (lighter) when the robot finishes following the zigzag path.

2. Classical exact cellular decomposition methods

Exact cellular decomposition methods break the free space (i.e., the space free of obstacles) down into simple, non-overlapping regions called cells. The union of all the cells exactly fills the free space. These regions, where no obstacles lay in, are "easy" to cover and can be swept by the robot using simple motions. For instance, each cell could be covered using a zigzag, "mowing the lawn" pattern like the one shown in Fig. 1. Generation of these zigzag motions, also called Seed Spreader motions, is well documented in the literature [22–24].

Two cells are said to be adjacent if they share a common boundary. An adjacency graph can be used to represent the cellular decomposition, where a node represents a cell and an edge represents an adjacency relationship between two cells (see Fig. 2). Exact cell decompositions can be generated by sweeping a line through the space (e.g. from left to right). The cell boundaries are then formed when some event is encountered by the sweep line. For instance, a change on the number of times the sweep line intersects with obstacle boundaries can be used as an event.

Typically, a planner based on exact cellular decomposition generates a coverage path in two steps. First, it decomposes the free space into cells and stores the decomposition as an adjacency graph. Next, it computes an exhaustive walk through the adjacency graph (i.e., a sequence that visits each node in the graph exactly once). It is worth noting that the exhaustive walk obtained is a sequence of nodes (i.e., a sequence of cells), and not an actual coverage path. Therefore, an explicit path for covering each cell must be derived.

Next, two popular off-line cellular decomposition approaches (Sections 2.1 and 2.2) that laid down the foundations for more novel approaches are discussed.

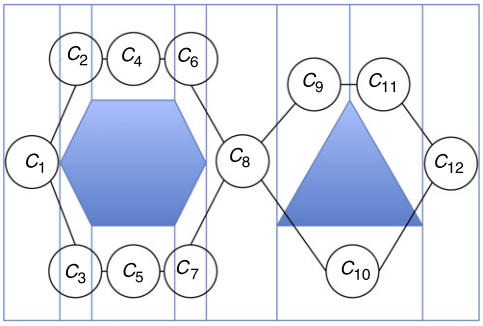


Fig. 2. Trapezoidal decomposition of an example workspace with its corresponding adjacency graph.

2.1. Trapezoidal decomposition

One of the simplest exact cellular decomposition techniques which can yield a complete coverage path is the trapezoidal decomposition [25,26], which handles only planar, polygonal spaces. This method can be classified as off-line. In the trapezoidal decomposition, each cell is a trapezoid, as shown in Fig. 2. Thereby, simple back-and-forth motions can be used to cover each cell. Complete coverage is guaranteed by finding an exhaustive walk through the adjacency graph associated to the decomposition (overlaid on the decomposition in Fig. 2). Finally, a specific zigzag path to cover each cell is generated.

The exhaustive walk determines the order in which the cells are visited to achieve complete coverage. Finally, specific paths to cover each cell are generated, typically using back-and-forth motions in a "mowing the lawn" manner.

As an application example, [27] proposed an off-line algorithm based on the trapezoidal decomposition for coverage path planning in the case of agricultural fields and agricultural machines. Their algorithm applies a trapezoidal decomposition of the field followed by a cell merging procedure. The resulting cells are similar to those produced by the boustrophedon decomposition, introduced in the next Section 2.2. To optimize the path, they use a path-based cost function to assess the largest cell arising in six different trapezoidal decompositions obtained by using a sweep line inclined at 30° intervals. Then, the three most favorable directions are selected and the process is repeated, with additional decompositions at 15° either side of the selected headings. The process continues iteratively until the improvement per step falls below a threshold, which for their application was achieved after 5 steps (about 1° accuracy), requiring 36 separate decompositions. Then, the largest cell in the minimum-cost decomposition is removed from the target area, and the process is repeated for the remainder of the field until all the area is covered by the path. This scheme produces effectively optimal coverage paths for a convex field and high-quality paths for a field whose boundaries consist of long, straight segments as well.

2.2. Boustrophedon decomposition

A drawback of the trapezoidal decomposition is that it generates many cells that, intuitively, can be merged together to form bigger cells. This is clearly inconvenient, as the more cells are present, the longer the final coverage path becomes, as shown in Fig. 3. This happens because the trapezoidal decomposition creates only convex cells. However, non-convex cells can also be completely covered by simple motions. To overcome this limitation, Choset and Pignon proposed the boustrophedon cellular decomposition [23,28]. The word "boustrophedon" comes from ancient Greek, and literally means "the way of the ox", signifying the pattern in which an ox drags a plow back and forth. The boustrophedon decomposition is similar to the trapezoidal decomposition

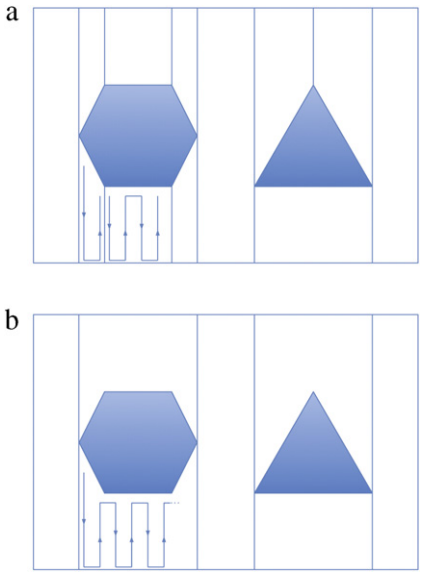


Fig. 3. A decomposition with less cells allows for shorter coverage paths. Note how an extra strip is needed in the trapezoidal decomposition in 3(a) with respect to the boustrophedon decomposition in 3(b).

introduced above, but it only considers vertices where a vertical segment can be extended both above and below the vertex. The vertices where this occurs are called critical points.

By adhering to this strategy, the boustrophedon decomposition effectively reduces the number of cells in trapezoidal decomposition. Hence, shorter coverage paths are obtained. Notice that, as the trapezoidal decomposition, this method assumes polygonal obstacles and the terrain to be known *a priori*, and thus classifies as an off-line method.

3. Morse-based cellular decomposition

Later [24] generalized the boustrophedon decomposition by proposing a novel cellular decomposition approach based on critical points of Morse functions [29]. In fact, they show that the boustrophedon decomposition is a particular case of Morse decomposition. With respect to the original boustrophedon decomposition, the Morse-based decomposition has the advantage of handling also non-polygonal obstacles. By choosing different Morse functions, different cell shapes are obtained, e.g. circular or spiked cells. Theoretically, Morse-decompositions can be applied to any n-dimensional space. Moreover, they presented a method to perform coverage of planar spaces by detecting the critical points using sensory range information, and a motion-template-based algorithm that ensures to encounter all the critical points in the target area. Therefore, this method allows complete coverage online [30,31].

The Morse decomposition is based on a roadmap method for start-to-goal path planning proposed by Canny [32,33]. Critical points of a Morse function restricted to the obstacle boundaries are used to determine the cell decomposition. Recall that, given a real-valued function h: $\mathbb{R}^m \rightarrow \mathbb{R}$, its differential at $p \in \mathbb{R}^m$ is $dh(p) = [\frac{\partial h}{\partial x_1}(p) \ldots \frac{\partial h}{\partial x_m}(p)]$. A critical point is a value $p \in \mathbb{R}^m$ where either the function is not differentiable or all its partial derivatives are 0, i.e., $\frac{\partial h}{\partial x_1}(p) = \cdots = \frac{\partial h}{\partial x_m}(p) = 0$, and its Hessian $(\frac{\partial^2 h}{\partial x_i \partial x_j}(p))$ is non-singular. For instance, in the case of a single variable function, a critical point corresponds either to a local maximum, a local minimum or an inflection. A Morse function is one whose critical points are nondegenerate [29]. Practically speaking, this means that critical points are isolated from one another.

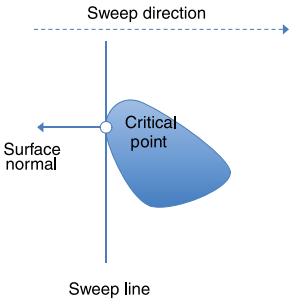


Fig. 4. Cell boundaries in Morse decomposition are placed at critical points, where the surface normal of the obstacle is perpendicular to the sweep slice, and parallel to the sweep direction.

To determine the cell decomposition, a slice is swept through the target space. Formally, the slice is a codimension one manifold defined in terms of the preimage of a real-valued Morse function, h: $\mathcal{W} \rightarrow \mathbb{R}$, where $\mathcal{W}$ is the robot's workspace, i.e., the space to be covered. For instance, in the plane ($\mathcal{W} = \mathbb{R}^2$), choosing $h(x, y) = x$ will make the slice be effectively a vertical line. Changes on the connectivity of the slice occur at critical points of the Morse function restricted to the obstacle boundaries. To put it more simply, at a critical point the sweep line encounters an obstacle whose surface normal is perpendicular to the sweep line, as shown in Fig. 4. Morse theory guarantees that, between critical points, the connectivity of the slice remains unchanged. Thus, as no obstacles lay between critical points, the space between them can be covered easily by simple motions, and critical points can be used to determine the cell boundaries.

Choosing different Morse functions produces different slice shapes and hence different cell decomposition patterns. For simplicity, we will describe the Morse-based boustrophedon decomposition [34], which happens in the plane. Later, we will give examples of different decomposition patterns obtained using different Morse functions.

In the boustrophedon decomposition, a vertical slice, defined in terms of the Morse function $h(x, y) = x$, is swept from left to right in the workspace, i.e., along the abscissae axis. Thus, the vertical slice is determined by the preimage of this Morse function, $\mathcal{W}_\lambda = h^{-1}(\lambda)$. The slice is parametrized by $\lambda \in \mathbb{R}$, which fixes its location in the target space. Increasing the value of the slice parameter, λ, sweeps the slice from left to right through the workspace.

As the slice sweeps the space it intersects (or stops intersecting) obstacles, which divide it into smaller pieces as the slice first encounters an obstacle, that is, the connectivity of the slice in the free space increases. Also, immediately after the slice leaves an obstacle, smaller slice pieces are merged into larger pieces (the connectivity of the slice in the free space decreases). The points where these connectivity changes occur are the critical points. (Recall that critical points are always located on the obstacle boundaries.) Thus, at critical points, the slice is used to determine the cells in the decomposition. Notice that within a cell, the slice connectivity remains constant. Fig. 5(a) shows how, at the critical point, the connectivity of the slice changes from one to two, and hence the old cell is closed and two new cells are created. In Fig. 5(b), at the critical point, the connectivity of the slice changes from two to one, and hence two old cells are closed and a new cell is created.

Once the cell decomposition is constructed, an exhaustive walk through its associated adjacency graph is determined by the planner. Then, it generates the explicit coverage path in each cell. The coverage pattern within each cell has three parts: motion along a slice, motion orthogonal to the slice, and motion along the cell boundary, as shown in Fig. 6. First, the robot laps along the current slice, $\mathcal{W}_\lambda$. Then the robot steps outward of the slice by going orthogonally to it by an inter-lap distance, typically by a distance of

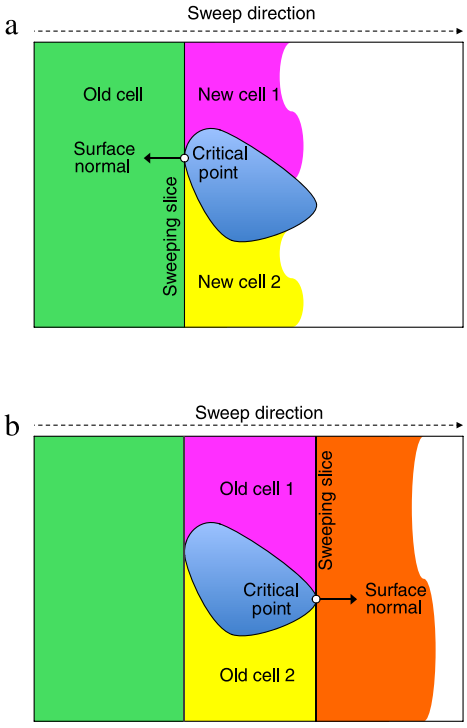


Fig. 5. Cell determination with the Morse-based boustrophedon cell decomposition method.

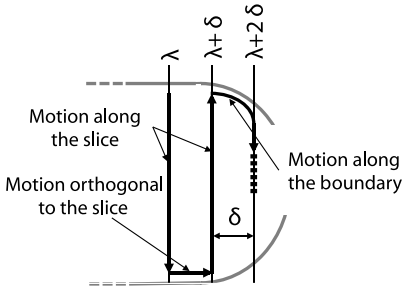


Fig. 6. Boustrophedon path construction process, where δ is the inter-lap spacing and λ is the slice parameter.

one robot sensor range; λ is also increased by this distance to form a new slice. If the robot encounters an obstacle (i.e., the cell boundary) while moving along the slice, the planner directs the robot to follow the obstacle boundary until it has moved an inter-lap distance and then a new lap along a new slice is started. The process repeats until the cell is completely covered.

Fig. 7 shows the Morse-based boustrophedon cell decomposition of an example workspace with its associated adjacency graph overlaid.

A key point of Morse decompositions is that, by choosing different Morse functions to define the slice that is swept through the space, different decomposition and coverage path patterns can be generated, like the spiral pattern [24]. Fig. 8 shows a spiral pattern obtained using the Morse function $h(x, y) = \sqrt{x^2 + y^2}$. Allowing different coverage patterns is useful for vehicles with kinematic constraints. For instance, a spiral path can be easily followed by an underactuated car-like vehicle unable to make hard turns [8].

A limitation of the Morse decomposition method is that it cannot handle rectilinear environments. This is because it is not possible to determine critical points in those environments which correspond to a change in the topology of the space (the critical points are said to be degenerate in this case [29]).

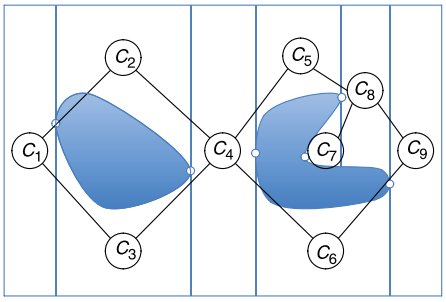


Fig. 7. Morse decomposition of an example workspace with its associated adjacency graph. .

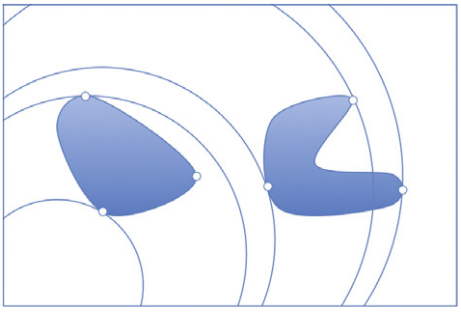


Fig. 8. Spiral Morse decomposition obtained using the Morse function $h(x, y) = \sqrt{x^2 + y^2}$.

Galceran and Carreras [35] presented a CPP method based on Morse decomposition to minimize redundant coverage for autonomous underwater vehicles and autonomous surface vehicles imaging the ocean floor. To this aim, this method determines a sweep direction on each cell of the decomposition and adjusts the inter-lap spacing of the boustrophedon paths on a lap by lap basis according to the ocean depth. The efficacy of the method is demonstrated in simulation using a real-world dataset.

3.1. On-line Morse-based boustrophedon decomposition

To face the sensor-based coverage problem, Acar and Choset gave a method to detect the critical points of a Morse-based boustrophedon decomposition on-line using range sensor information. Furthermore, they presented an algorithm that ensures to encounter all the critical points while performing coverage [31,36]. To detect the critical points, they use an omni-directional range sensor to look for points where the surface normals $\nabla m(x)$ of obstacles and the sweep direction are parallel. Given a robot located at point x, let c_0 be the closest point to x on the surface of obstacle C_i:

$$c_0 = \arg\min_{c \in C_i} \|x - c\|, \tag{1}$$

and let $d_i(x)$ be the distance between point x and the obstacle C_i. Now, the gradient of $d_i(x)$, $\nabla d_i(x)$ can be calculated as

$$\nabla d_i(x) = \frac{x - c_0}{\|x - c_0\|}. \tag{2}$$

Recall that, by definition, a gradient is a unit vector normal to a surface at a given point. In Eq. (2), as c_0 is a point laying on the surface of the obstacle C_i, $x - c_0$ is a vector pointing outward from c_0 towards x. Given that c_0 is the closest point to x on the obstacle surface, the vector $x - c_0$ is hence normal to the surface of the obstacle. By dividing by its norm, $\|x - c_0\|$, the result is turned into a unit vector.

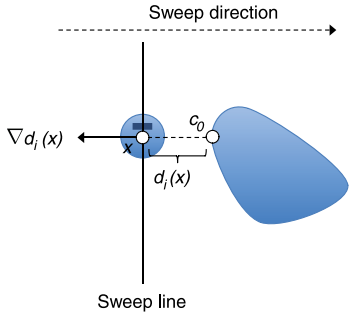


Fig. 9. Critical point detection occurs on the side of the range sensing robot, whose heading is indicated by the rectangular mark on the circle representing the robot.

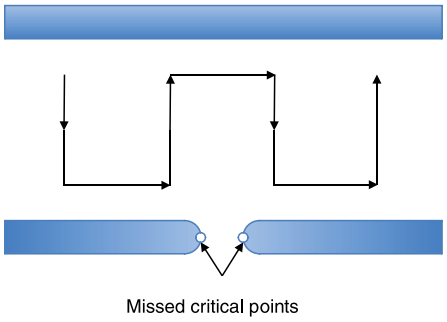


Fig. 10. With Morse decomposition, a range sensing robot following a simple zigzag path will miss the critical points in the figure unless it performs wall following both on the top and the bottom of a cell.

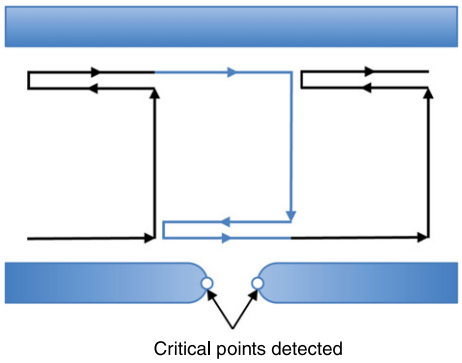


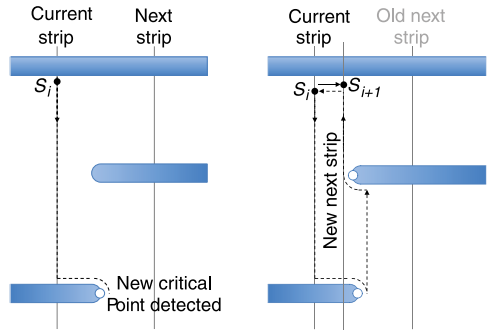
Fig. 11. A path composed of rectangular cycles allows detection of all the critical points. This pattern is used in on-line Morse decomposition and the CC_R algorithm as well.

A detection of a critical point occurs when $\nabla d_i(x)$ is parallel to the sweep direction. Or, in other words, when the sweep direction and the obstacle surface normal are parallel. Fig. 9 graphically shows this situation.

Notice the fact that a critical point can only be detected when it is the closest point to the robot on the obstacle surface. This implies that they can only be detected when the robot is performing wall following. Therefore, using a simple zigzag can result in some critical points being missed, as those shown in Fig. 10.

To address this issue, [24] presented an algorithm that uses repeated rectangular motion cycles with wall following on both ends of a lap, as shown in Fig. 11. The algorithm is termed "Cycle Algorithm". Notice that this cyclic path includes retracing, and hence is longer than the simple zigzag path.

The cycle path generation process of their proposed algorithm is shown in Fig. 12. Initially (Fig. 12(a)), the robot starts a forward phase at point S_i and moves downward. When it encounters an obstacle, it performs wall following until it has reached the next



(a) Forward phase. (b) Reverse phase.

Fig. 12. Critical point detection using the cycle algorithm.

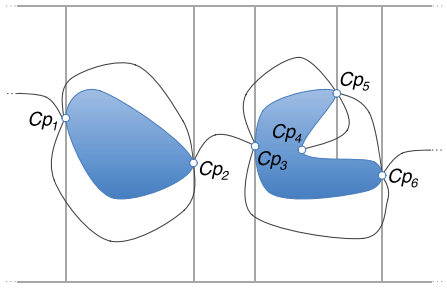


Fig. 13. Morse decomposition of an example workspace with its associated Reeb graph. $C_{p1} \ldots C_{p6}$ are the critical points.

strip or until a critical point is detected. If a critical point is detected, the robot then enters in a reverse phase where it moves upward (see Fig. 12(b)). In this phase, if an obstacle is detected, the robot wall-follows it. If a critical point is detected on the obstacle boundary, the next strip is moved to where the critical point is. The motion continues until S_i is reached again on the current strip. Then, the robot starts a new cycle at point S_{i+1}. The algorithm is proven to detect all the critical points.

In order to store and incrementally construct the Morse decomposition on-line, it is stored as a Reeb graph [37]. The Reeb graph is dual to the adjacency graph in that the nodes of the Reeb graph are the critical points and the edges connect the neighboring critical points, i.e., correspond to cells. An example of Morse decomposition with its associated Reeb graph is shown in Fig. 13.

Whenever a critical point is encountered, the robot updates the Reeb graph. When a cycle path where critical points were found is finished, the robot looks for uncovered cells at the last encountered critical point. If the critical point is associated with two uncovered cells, the robot picks one of the cells associated as the next cell to cover. If there are no uncovered cells associated with the last encountered critical point, a depth-first search is performed on the Reeb graph. To travel to the selected uncovered cell, the robot follows the Reeb graph and plans a path that passes through cells and critical points. When no uncovered cells (edges) remain in the Reeb graph, the environment is completely covered.

The Cycle algorithm just described, however, may fail to detect the critical points on certain non-convex obstacles. In particular, the concave critical points such as *Cp*2 in Fig. 14 cannot be detected by range data when the boundary's curvature is smaller than the robot's periphery [38]. However, the closest convex critical point (*Cp*3 in the example shown in Fig. 14) to a critical point like *Cp*2 in Fig. 14 will get detected. This leads to adding a spurious edge to the Reeb graph that does not correspond to any existing cell. As a result, the algorithm will fail to detect the closing critical point for the newly added edge. Garcia and de Santos [38] propose a solution to this issue which involves associating each critical

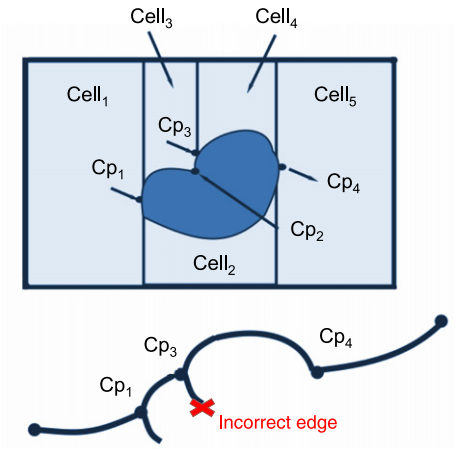


Fig. 14. A concave critical point will not be detected if the boundary's curvature is smaller than the robot's periphery and will lead to an incoherent Reeb graph. This is the case of *Cp*2 in this example environment, which does not get detected and produces an incorrect edge emanating from *Cp*3.

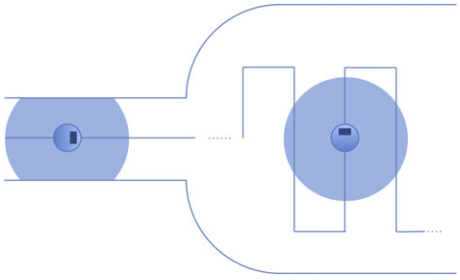


Fig. 15. Combination of Morse decomposition and GVD for extended range sensor coverage. In cluttered spaces (left) the robot just follows the GVD of that space. In vast areas (right), the robot follows the pattern generated using a Morse decomposition scheme.

point with its obstacle and defining unique entry and exit critical points for each obstacle. Additionally, their paper discusses several implementation details of the Cycle algorithm.

Acar et al. [6] discussed coverage path planning in relation to a demining application. In this article, they show that the Morse decomposition overcomes the random coverage approach in this task, which used to be considered the state of the art on demining operations.

3.2. Morse-based cellular decomposition combined with the generalized Voronoi diagram

Acar et al. [39,40] presented a sensor-based coverage approach with extended range sensors. As they point out, "prior work in coverage tends to fall into one of two extremes: coverage with an effector the same size of the robot, and coverage with an effector that has infinite range". In this work, they consider coverage in the middle of this spectrum: coverage with a detector range that goes beyond the robot, and yet is still finite in range. They term these sensors extended range sensors. In this work, coverage is achieved in two steps: the first step considers vast, open spaces, where the robot can use the full range of its detector; the robot covers these spaces as if it were as big as its detector range (see Fig. 15, on the right). Here the previous work in using Morse cell decompositions [24] is employed to cover unknown spaces. As explained above, cells in this decomposition can be covered via simple back-and-forth motions, and coverage of the vast space is then reduced to ensuring that the robot visits each cell in the vast space. The second step considers the narrow or cluttered spaces

where obstacles lie within the detector range, and thus the detector "fills" the surrounding area. In this case, the robot can cover the narrow space by simply following the Generalized Voronoi Diagram (GVD) of that space, which are sets of points equidistant to two obstacles [41,42] (see Fig. 15, on the left). The GVD can be constructed on-line using range sensor information and has been previously used for sensor-based exploration [43] and inspection of 3D structures [44]. A hierarchical decomposition that combines the Morse decompositions and the GVDs is introduced to ensure that the robot indeed visits all vast, open, as well as narrow, cluttered, spaces. In their article, it is shown how to construct this decomposition online using the sensor data accumulated while the robot covers the environment.

4. Landmark-based topological coverage

Wong and MacDonald [45,46] and Wong [47] presented an on-line topological coverage algorithm for mobile robots based on the detection of natural landmarks. This work is intended for simple planar environments. As in Morse decomposition, their method also uses concepts introduced by boustrophedon decomposition. However, the topological algorithm proposed here uses different events to determine cell boundaries. Morse decomposition places cell boundaries on the critical points on the obstacles surface. However, as commented before, rectilinear environments cannot be handled by Morse decomposition, as the critical points in such environments are degenerate. On the other hand, as critical points can only be discovered on the side of the robot while performing wall following, a rectangular coverage pattern which includes retracing is needed. In comparison, the topological approach presented here uses simpler landmarks to determine an exact cellular decomposition termed "slice decomposition". Due to the use of simpler landmarks, slice decomposition can handle a larger variety of environments, including ones with polygonal, elliptical and rectilinear obstacles. Moreover, obstacles can be detected from all sides of the robot, allowing a simpler zigzag pattern without retracing to be used. As a result, the generated coverage path is shorter with this method.

4.1. Slice decomposition

The slice decomposition is constructed by sweeping a line through the space. It uses five different events to determine the cell boundaries:

1. *Split event*: a free space segment in the previous strip is split into two by the emergence of an obstacle, as in Fig. 16(a).
2. *Merge*: two free space segments in the previous strip are merged into one by the disappearance of an obstacle, as in Fig. 16(b).
3. *Lengthen*: the current strip is much longer than the previous strip, as in Fig. 16(c).
4. *Shorten*: the current strip is much shorter than the previous strip, as in Fig. 16(d).
5. *End*: the previous free space segment is the final one in the current cell, as in Fig. 16(e).

All these events or landmarks can be detected using a combination of range measurements thresholding and temporal sequence comparisons (comparing current sensor reading with previous ones) and odometry (comparing length of consecutive strips). An alternative solution that uses a neural network to detect the events is also presented in this work [47].

Whenever one of the stated events occur, a cell boundary is placed along the sweep line where the event takes place. The slice decomposition can be encoded as a topological map. A topological map is represented as a planar graph, where the nodes represent landmarks (i.e., split, merge, end, lengthen or shorten events) and edges indicate the types of motion required to travel between nodes they are incident upon. For example, whether the edge is next to a wall and which side the wall is on. They also store the estimated distances separating the two nodes they connect.

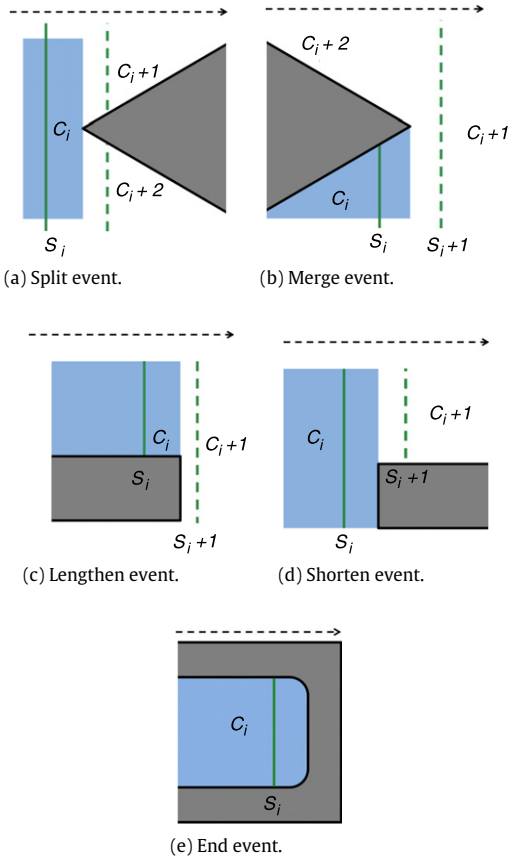


Fig. 16. Events (landmarks) in the slice decomposition. In all the events shown c_i is the current cell (shaded) and s_i is the current strip. The dashed arrow indicates the sweep direction.

4.2. On-line topological coverage algorithm

An algorithm that constructs the slice decomposition on-line while performing coverage is presented in Wong et al.'s work. The algorithm guarantees complete coverage. It iteratively constructs the topological map associated to the slice decomposition of the environment using a finite state machine with three states—boundary, normal, and travel. Fig. 17 shows its state transition diagram. The algorithm starts in the boundary state, as it is assumed that the robot is initially located in a corner of the environment. This assumption is not a shortcoming as it is easy to program a robot to look for a corner by following simple forward and wall following motions. In the boundary state, the robot explores the current cell boundary. The aim of the boundary exploration is to expose all cells neighboring the current border. Whenever the robot arrives at a landmark or at an end of the cell boundary, the topological map is updated. When the boundary exploration has finished, the algorithm switches to the travel state. In the travel state, the robot searches the topological map for an uncovered cell and it is directed to that cell. Then, the robot enters the normal state, where it follows a zigzag pattern to cover the current cell. Again, whenever a landmark is found the topological map is updated and the algorithm switches to the boundary state. The algorithm finishes when there are no more uncovered cells in the topological map.

5. Contact sensor-based coverage of rectilinear environments

Butler et al. [48] proposed CC_R, an exact cell decomposition algorithm for contact sensing robots (i.e., robots without range sensing capabilities) covering unknown rectilinear environments on-line. Their motivating application for coverage of rectilinear

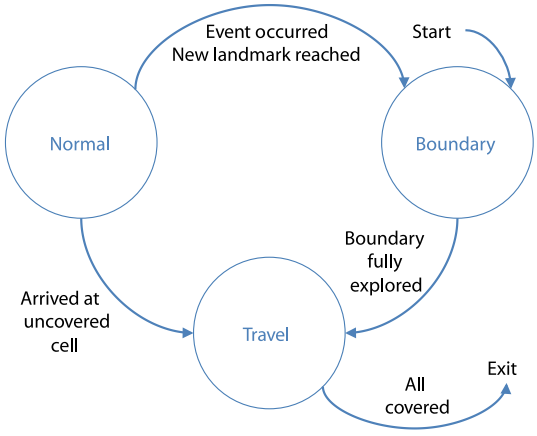


Fig. 17. State transition diagram of the topological coverage algorithm.

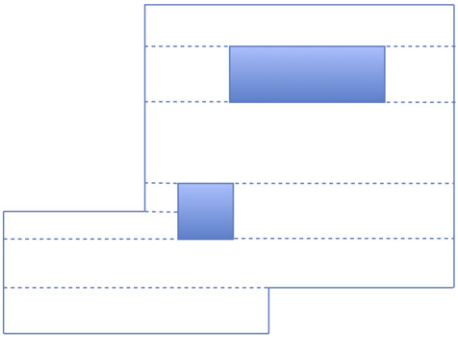


Fig. 18. CC_R uses an exact cell decomposition for rectilinear environments.

environments is calibration of an automated assembly system in which planar linear motors operate on table-like surfaces to transfer products through a factory.

In CC_R (for contact-based coverage of rectilinear environments), the robot follows a cyclic path with retracing as shown in Fig. 11. At the same time, it iteratively constructs a cellular decomposition of the environment. An example rectilinear decomposition produced by CC_R is shown in Fig. 18. In fact, the decomposition constructed by CC_R can be seen as the case of Morse decomposition where all the critical points are degenerate, as this is the case in rectilinear environments.

Normally, CC_R follows the cyclic path. An event (and hence a cell boundary) occurs whenever the robot is prevented from successfully executing a full path cycle. When such an event occurs, CC_R chooses a new trajectory based only the robot's environment and its current position. The next trajectory is determined by a list of rules that are designed to continue coverage in all possible cases.

A proof of completeness for CC_R is given by creating a finite state machine (FSM) that describes all possible events encountered by the robot, and demonstrating that the FSM has no infinite loops and that it stops only when coverage is complete.

6. Grid-based methods

Grid-based methods use a representation of the environment decomposed into a collection of uniform grid cells. This grid representation was first proposed by Moravec and Elfes [49] to map an indoor environment using a sonar ring mounted on a mobile robot. In this representation, each grid cell has an associated value stating whether an obstacle is present or if it is rather free space. The value can be either binary or a probability [50]. Typically, each grid cell is a square, but also different grid cell shapes can be used, such as triangles. As grid representations only approximate the shape

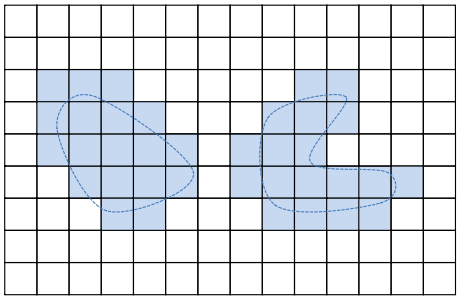


Fig. 19. An example grid map. Grid cells with obstacles present are shaded.

of the target region and its obstacles, Choset classified grid-based methods as approximate cellular decompositions [20]. As a result of this approximate representation, most grid-based methods are "resolution-complete", that is, their completeness depends on the resolution of the grid map. Fig. 19 shows an example grid map.

It is easy to create a grid map, as it can be represented as an array where each element contains occupancy information associated with a cell. On the other hand, it is simple to mark covered areas in a grid map. As a result, the grid-based representations are the most widely used for coverage algorithms. Nonetheless, grid maps suffer from exponential growth of memory usage because the resolution remains constant regardless of the complexity of the environment [51]. Also, they require accurate localization to maintain the map's coherency [52,53].

For these reasons, grid-based coverage methods are suited for indoor mobile robot operations, as the size of the area to be covered is typically relatively small.

Usually, cells in a grid map are square in shape and robot-sized. Oh et al. [54] proposed a coverage algorithm that uses a grid map in which the cells are triangles instead. The rationale behind the choice of triangular cells is that they offer a higher resolution in comparison to rectangular cells of similar size. However, the resolution of the grid can also be augmented by using finer-grained squared cells. In mobile robotics, the field for which the mentioned algorithm is intended, most mobile robots are not capable of making very fine movement adjustments, and hence there is no need for ultra high resolution in coverage path planning. Therefore, the extra effort devoted to implementing a triangular grid seems not to be worthwhile.

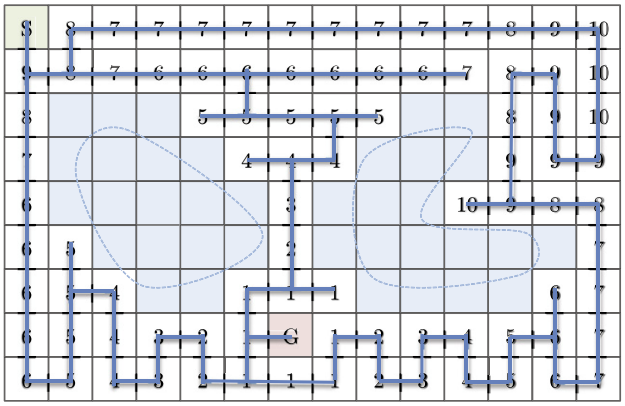
6.1. Grid-based coverage using the wavefront algorithm

Zelinsky et al. [55] presented the first grid-based method for coverage path planning. In their off-line method, they use a grid representation and apply a complete coverage path planning algorithm to the grid. The method requires a start cell and a goal cell. A distance transform that propagates a wave front from goal to start is used to assign a specific number to each grid element. That is, the algorithm first assigns a 0 to the goal and then a 1 to all its surrounding cells. Then, all the unmarked cells neighboring the marked 1 are labeled with a 2. The process repeats incrementally until the start cell is reached by the wavefront. Fig. 20(a) illustrates this procedure on an example environment.

Once the distance transform is calculated, a coverage path can be found by starting on the start cell and selecting the neighboring cell with the highest label that is unvisited. If two or more unvisited neighbors share the same label, one of them is selected randomly. This process to find a coverage path is equivalent to using pseudo-gradient descent from the start point on the numeric potential function constituted by the labeling, that is, following the equipotential curves from top to bottom. Fig. 20(b) shows the generated coverage path for the example environment on Fig. 20(a). A

S	8	7	7	7	7	7	7	7	7	7	8	9	10
9	8	7	6	6	6	6	6	6	6	7	8	9	10
8				5	5	5	5	5			8	9	10
7					4	4	4				9	9	9
6						3				10	9	8	8
6	5					2							7
6	5	4			1	1	1					6	7
6	5	4	3	2	1	G	1	2	3	4	5	6	7
6	5	4	3	2	1	1	1	2	3	4	5	6	7

(a) Wavefront distance transform for the selection of the start position (S) goal position (G).



(b) Coverage path generated using the wavefront distance transform with the selection of the start position (S).

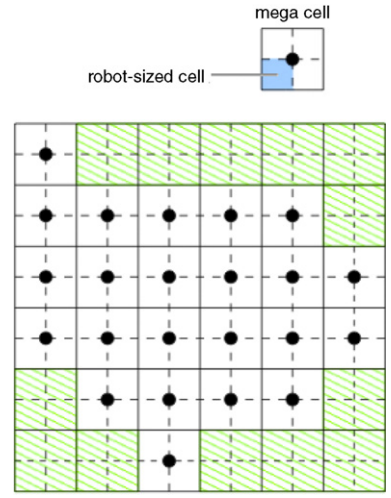
Fig. 20. Coverage path planning using the wavefront algorithm for an example environment.

unique feature of this coverage algorithm is that a start and a goal point can be specified.

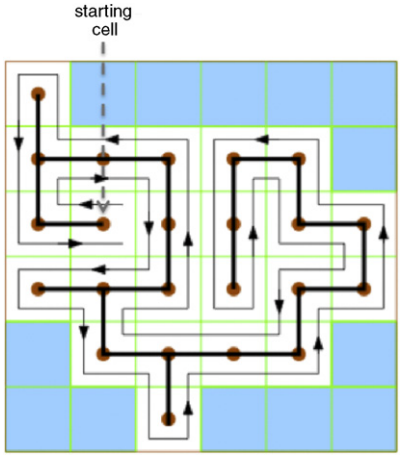
Shivashankar et al. [56] introduced a generalization of the wavefront algorithm to unknown environments to achieve on-line coverage with a mobile robot.

6.2. Grid-based coverage using spanning trees

Gabriely and Rimon [57] proposed the Spiral-STC (Spanning Tree Coverage) algorithm, an on-line approach for mobile robots which consists in subdividing the workspace into a grid map and following a systematic spiral path. This systematic spiral path is generated by following a spanning tree of the partial grid map that the robot incrementally constructs using its onboard sensors. The robot is able to cover every grid cell precisely once, and travel a complete coverage path. They validate the proposal in simulation. The Spiral-STC algorithm works as follows. Two different grid cell sizes are used. Bigger cells (so called mega cells) are divided in four smaller cells, which are the same size as the robot. This is shown in Fig. 21(a). To perform coverage, the robot executes the following procedure. Starting at the current cell, the robot chooses a new travel direction by selecting the first new mega cell in the free space in anti-clockwise direction. Then, a new spanning-tree edge is grown from the current mega cell to the new one. The algorithm is called recursively. The recursion stops only when the current cell has no new neighbors (a mega cell is considered old if at least one of its four smaller cells is covered, it is considered new otherwise). As a result of this recursion, the robot moves along one side of the



(a) Approximate cell decomposition in mega cells and robot-sized cells.



(b) Coverage path generated with the Spiral-STC algorithm.

Fig. 21. Coverage path planning using Spiral-STC algorithm.

spanning tree until it reaches the end of the tree. At that point, the robot turns around to traverse the other side of the tree. It is worth noticing that, when coverage is completed, the robot returns to the start cell, facilitating its collection and storage. On the other hand, STC never visits any small cell twice and thus minimizes the cover time. Fig. 21(b) shows an example of a coverage path generated by the Spiral-STC algorithm.

That is, when coverage is completed, the robot returns to the start cell. Fig. 21(b) shows an example of a coverage path generated by the Spiral-STC algorithm.

An extension to the Spiral-STC is the Backtracking Spiral Algorithm (BSA) [58], which is also an on-line approach intended for mobile robots. As an advantage in regard to the Spiral-STC algorithm, they proposed an extension to cover not only unoccupied cells, but also the partially occupied ones. This extension is based on the idea that the partially-occupied cells are part of the external ring of the spiral path. These cells are covered by a wall-following procedure. The proposed extension can be applied to most grid-based coverage algorithms. Simulation experiments validate the proposed algorithm.

Choi et al. [59] and Lee et al. [60] proposed an on-line complete coverage path planning solution based on the ideas introduced by the Spiral-STC algorithm and the BSA algorithm. They also use systematic spiral paths to achieve coverage, based on active

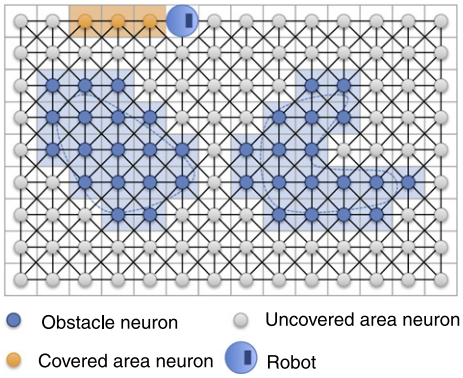


Fig. 22. Schematic of the neural network used by Luo, Yang and others to achieve coverage.

wall finding. Nonetheless, they introduce a map coordinate assignment scheme based on the history of sensor readings to improve the time-to-completion by reducing the number of turns on the generated path. The generated spiral paths are then linked by an inverse distance transform they introduce. This proposal is validated in simulation and also with real-world experiments conducted inside a room with a mobile robot.

6.3. Neural network-based coverage on grid maps

Luo et al. [61] and Yang and Luo [62] propose to use a neural network to achieve CPP on-line targeted to a floor cleaning application. They discretize a 2D space in a grid map where the diagonal length of each grid cell is equal to the robot sweeping radius, and then a neuron is associated to each and every grid cell. Each neuron has connections to its immediate 8 neighbors. These concepts are shown in Fig. 22.

A shunting equation based on the membrane equation by Hodgkin and Huxley [63] determines the dynamics of each neuron in the network. The activity landscape (i.e., the output value of all neurons at a given instant) of the shunting model used attracts the robot to unclean areas, while the robot is repulsed by already cleaned areas and obstacles. The next position of the robot is determined by the current position of the robot and the activity of the neuron associated to its current position, without any prior knowledge about the environment. It is assumed that the current state of the robot (if it is in a clean or dirty area, or in front of an obstacle, and its location) can be determined via sensory information. The state of the robot is an input to the neural network. The model used has 6 parameters that can be tuned in a wide range of values at the neural network design phase, and hence coverage is achieved without any learning procedures. An advantage of this method is that it can handle non-stationary environments (i.e., dynamically changing obstacles). The proposed neural network approach is validated in simulation. In [64] further simulation results are presented as well as a method to perform mapping on-line simultaneously with coverage navigation. In this later work, they consider a typical grid-based map and also a triangular mesh representation of the space, such as the one used in [54].

An application of this neural network-based method to an autonomous underwater vehicle covering a 2-D workspace in the seabed is proposed in [65]. The proposal is validated in simulation. However, a 2-D underwater workspace is rather unrealistic. Furthermore, discretization in a grid map of a vast environment such as the seabed presents a tough challenge in terms of computational burden.

Qiu et al. [66] add a local path planning technique on top of the biologically inspired neural network approach discussed above. In their approach, the next robot position is not determined immediately but rather a local path planning occurs in a window comprising a determined vicinity of the robot. By using this technique, they reduce the computational burden in comparison with the neural-network-only approach.

In a similar approach, [67] present a neural network-based method to generate continuous steering control for a robot to completely cover a bounded region over a finite time. First, they discretize the space in a regularly spaced, disk-shaped grid. Then, a neural network based on the same biologically inspired shunting equation as in the works discussed above is used to provide continuous steering to the robot. The algorithm works for car-like robots which have nonholonomic motion constraints. The approach is validated in simulation.

6.4. Hexagonal grid decomposition for robots equipped with side-looking sensors

Paull et al. [68,69] presented an on-line coverage method for robots equipped with side-looking sensors. Their target application is mine countermeasure operations using an autonomous underwater vehicle equipped with a side-scan sonar. This coverage method continuously directs the vehicle's heading using multi-objective optimization to maximize the information gain produced by the sensor measurements. The optimization procedure uses a grid decomposition composed of uniform hexagonal cells. The advantage of using a hexagonal grid is two-fold. On one hand, distance does not need to be taken into account in the objective functions, because the distance from a given cell to its neighboring cells is the same. On the other hand, assuming the hexagons are small enough, visiting one cell guarantees coverage of two neighboring cells by the side-looking sensor, minimizing the amount of partially covered cells. Although the proposed method is able to cover target areas with non-convex shapes, obstacles present amidst the workspace are not considered. The efficacy of this method is demonstrated in simulation and experimentally on an autonomous underwater vehicle conducting mine countermeasure operations.

7. Graph-based coverage

Xu [70] presented coverage algorithms for environments that can be represented as a graph, such as a street or road network. In particular, this work addresses the following issues in the coverage problem. First, it takes into account that the prior map information provided as a graph might be incomplete. Second, it accounts for environmental constraints, such as restrictions in certain directions in the graph (corresponding to a one-way street, for example). Third, it provides strategies for on-line re-planning when changes in the graph are detected by the robot's sensors when performing coverage. Finally, strategies for coverage using multiple robots are provided.

Graph search algorithms are proposed to solve the coverage problems considered. Optimality is addressed by assigning a cost to each edge in the graph and either looking for the optimal solution when the deliberation time allows or rather quickly finding an approximated solution when time constraints apply.

8. 3D coverage

Most coverage path planning methods, and in particular the methods reviewed so far in this article, assume that the environment can be modeled as a simple planar surface. This assumption is valid for floor cleaning, land mine detection, lawn mowing,

etc. However, some surfaces in nature are 3-dimensional, and 3-dimensional coverage path planning is required instead to cover these surfaces. This is the case of an autonomous underwater vehicle covering the seabed [3] or a robot spray-painting vehicle parts [2], for instance. Next, we review several 3-dimensional coverage methods. It is worth noticing that, except the algorithm discussed in Section 8.1, the methods discussed below actually focus on coverage of a surface of lower dimension than the robot's workspace. Indeed, in 3-dimensional coverage, covering 2-dimensional surfaces embedded in 3-dimensional space such as the boundaries of automotive parts, the boundaries of buildings, the ocean floor, rugged agricultural fields or the boundaries of the in-water part of a ship hull are the main focus. This contrasts with the standard CPP problem, in which all the free space must be covered.

8.1. 3D coverage using a planar coverage algorithm in successive horizontal planes

Hert et al. [3] presented a 3D coverage algorithm that is based on a planar 2-dimensional terrain-covering algorithm [22]. Their target application is an autonomous underwater vehicle imaging the sea bottom. Their solution applies to a 3D projectively planar environment by applying the planar terrain-covering algorithm in the successive horizontal planes laying at different depths. The restriction to projectively planar environments means that elements such as caves are not handled by this method. Their 2D terrain-covering algorithm uses a partial discretization of the space in where the space is divided in vertical slices of the same width, but where the top and bottom of each slice can have any shape. This discretization is classified as a semi-approximate cellular decomposition according to Choset's taxonomy [20]. A robot following this algorithm may start at an arbitrary point in the environment and will zigzag along parallel straight lines (grid lines) to cover the given area. Portions of the area that either would not be covered or would be covered twice using the zigzag procedure are detected by the robot and covered using the same procedure; that is, the procedure is applied recursively. These smaller areas, called inlets, are covered as soon as they are detected and inlets within inlets are treated in the same way. Hence, the inlets are covered in a depth-first order. By requiring the robot to remember the points at which it enters and exits every inlet it covers (which define the inlet doorways), the algorithm assures that each inlet is covered only once. When entering or exiting a certain type of inlet, the robot may cover the same area more than once, or miss some area at the inlet. Those inlets are called diversion inlets, and special procedures are necessary for covering them effectively. The robot enters a diversion inlet by moving along its boundary. After covering a given diversion inlet, the robot exits it by resuming its path as if the diversion inlet did not exist. When the area to be covered is not simply connected and contains islands as well as inlets, the same basic procedures are used, but with minor modifications to ensure that the area surrounding every island is covered. The robot is able to convert the part of the area around each island that would normally not be covered into an artificial inlet by remembering certain points along its path. Artificial inlets are covered in the same way that real diversion inlets are. Fig. 23 illustrates this procedure. It is worth noticing, however, that details on how to detect the inlets used by the algorithm using sensor information are not provided.

The work of [3] was extended later in [71] to cover only areas that are close to the sea bottom surface. In this later work, it is assumed that the regions of interest in underwater environments are the ones close to the sea bottom. Therefore, aiming to make the robot navigate only in areas close to the surface, artificial obstacles (artificial islands) are introduced in the robot's map of the environment. This way, the volume of water at a certain distance from the seafloor surface is discarded and a more efficient exploration of the sea bottom is achieved.

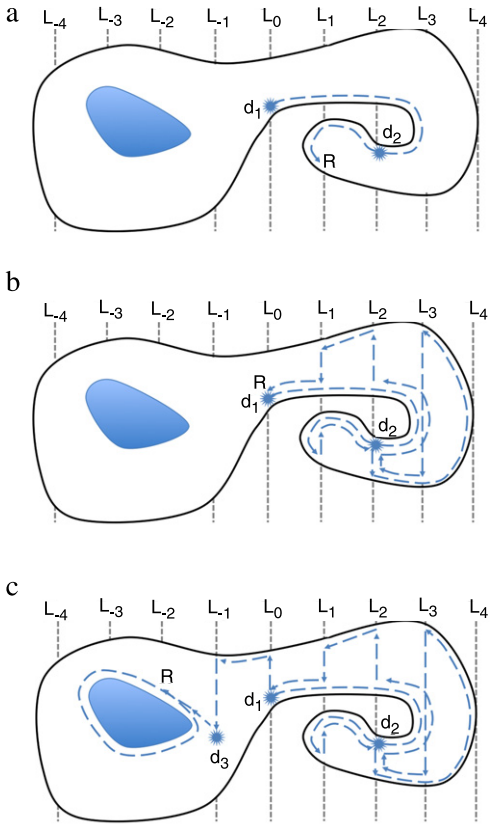
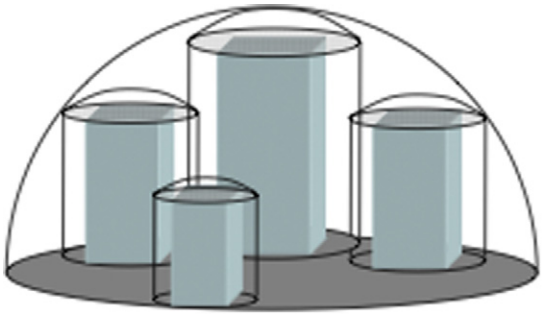


Fig. 23. The path a robot R follows in a non-simply connected environment when applying the algorithm proposed in [3]. First (a), the robot detects an inlet at d_1 and starts to cover it following a depth-first order. A second inlet is detected at d_2, and the robot starts covering it likewise. The robot continues to cover the rest of the inlets until it goes back to d_1 (b). Here, the robot continues to cover the main region until it detects an inlet at d_3. This inlet corresponds to an island, and hence the robot continues to circumnavigate it completely (c). Then, the robot will eventually pass through d_3 again and there it will resume the covering of the main area.

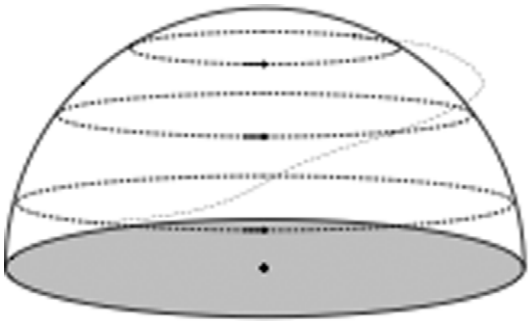
A theoretical proof of correctness of the algorithm is given in [3] and the extension proposed in [71] is validated in simulation.

8.2. 3D cellular decomposition

Atkar et al. considered the problem of trajectory generation for spray-painting robots. In their early work, they proposed an on-line, 3-dimensional, on-line CPP method for closed, orientable surfaces embedded in $\mathbb{R}^3$ [72]. The method extended the ideas of Morse decomposition to non-planar spaces. However, obstacles on the target surface are not considered in this work. Addressing their spray-painting target application, the method does not plan a coverage path on the target surface, but the coverage path is rather planned in an offset surface from which the end effector will spray the target surface. That is, the path is planned on a "virtual" surface that wraps the target object at a fixed offset distance. The coverage path is generated by intersecting a slice plane with the offset surface at equally spaced intervals. At each interval, the intersection of the slice plane with the offset surface forms a closed one-dimensional loop around the object. The robot traces this loop and moves to the next slice plane, and iteratively repeats the process. If the target surface is convex, the described process will achieve complete coverage. However, if the surface is non-convex and includes elements such as a bifurcation, the planner will use the critical points occurring in such shape changes to divide the surface in cells that will be covered individually. As in



(a) Simplified model of an urban environment.



(b) Illustration of a coverage path.

Fig. 24. Coverage scheme presented in [75].

the on-line Morse decomposition for planar spaces, a Reeb graph is used to encode the topology of the target surface. When all edges in the graph are covered, the coverage task is successfully completed. The method was validated in simulation using target surfaces constituted by polyhedra. It is worth noting that this method requires a robot equipped with a 2D omnidirectional range sensor in order to detect the critical points.

In a later work, they presented an off-line CPP method specifically targeted for spray-painting of automotive parts. They term such surfaces pseudo-extruded surfaces. It is worth noticing, however, that in this work the tackled problem is the uniform coverage problem, where the target surface not only needs to be completely covered but also the resulting paint deposition must meet certain uniformity requirements. To achieve uniform coverage, their proposed method takes a CAD model of the target automotive parts as input and segments their surface into topologically simple cells of similar curvature [73,74,2]. Then, individual, optimal paint-deposition coverage paths in each cell are determined. Simulations as well as experiments with real robots validate their proposal.

8.3. 3D urban structure coverage

Cheng et al. [75] presented an off-line approach for planning time-optimal trajectories for UAVs performing 3D urban structure coverage. First, they simplify the structures to be covered, namely buildings, into hemispheres and cylinders. Then, trajectories are planned to cover these simpler surfaces. Their proposal is validated in hardware-in-the-loop simulations using a fixed-wing aircraft. Fig. 24 illustrates this method.

8.4. Coverage of bathymetric surfaces

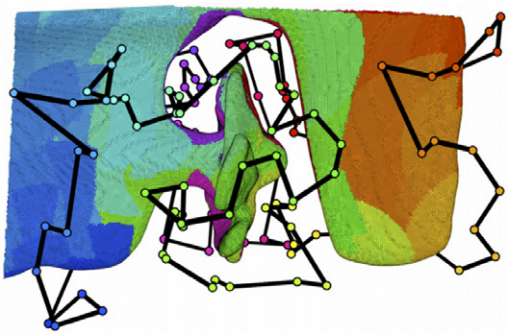
Bathymetric maps are elevation maps of the ocean floor. Such maps are used for the planning of ocean survey missions for autonomous underwater vehicles. A typical survey consists of a lawnmower-like pattern which follows the elevation profile of the terrain, imaging the bottom from an overhead point of view with some sensor. However, rugged, high-relief terrain cannot be properly imaged from such point of view since the imaging angle is too large on vertical protrusions. In contrast, an imaging angle close to the surface normal is desired. To address this issue, [76] presented a CPP method that plans a coverage path on a bathymetric map. The method determines regions that cannot be properly imaged using a traditional survey according to the terrain's gradient. Then, on each identified region, it plans a 3D coverage path that follows the horizontal contours of the surface, in a similar fashion as the method discussed in 8.2. On the remaining, effectively planar regions, it uses a standard lawnmower-like path for coverage. As a result, the entire target bathymetric surface is covered and a fair imaging angle is provided throughout the coverage path. The method is validated on a real-world bathymetric map comprising a large underwater volcanic area at 350 m depth.

8.5. 3D coverage for arable farming

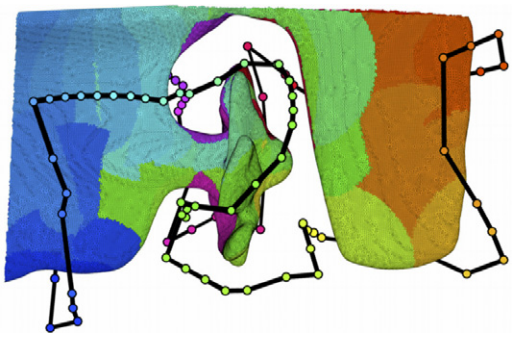
Jin and Tang [77] presented coverage algorithms for arable fields represented as elevation maps. Previous work in coverage for agricultural fields dealt with planar terrain, but many fields present 3-dimensional features that have an impact on coverage performance. Addressing this issue, this work provides coverage algorithms based on a seed curve that is incrementally offset on both sides to generate a coverage path. The method optimizes the seed curve selection by taking into account its associated number of turns, the soil erosion cost and the skipped area. The method is validated on real-world elevation maps of agricultural fields.

8.6. Random sampling-based coverage of complex 3D structures

In confined 3-dimensional areas where a robot cannot go through the spaces between component structures, or where occluded areas are only visible from a reduced set of viewpoints, modular approaches such as those described above are unsuitable. To handle this family of problems, global path planning strategies, utilizing sampling-based planning [78] have been applied to find feasible, collision-free, paths through confined areas and obtain full coverage of a 2-dimensional target structure. Their approach is based on the art gallery problem. Building upon a similar idea, [12] introduce an off-line, sampling-based coverage algorithm to achieve complete sensor coverage of complex, 3-dimensional structures. Their target application is autonomous ship hull inspection, in which the robot must cover the in-water part of the hull surface using a sensor such as a sonar. The sensory data collected in situ is later used to construct an accurate 3-dimensional model where anomalies in the hull surface can be searched for. They consider the planning problem with a fully-actuated, six degree-of-freedom hovering AUV that uses a bathymetry sonar to inspect the structures in the ship hull. The method requires a discrete model of the structure to be inspected provided in the form of a closed triangular mesh. The planning is performed in two steps. First, a graph of feasible paths for the robot is constructed using random sampling until the set of nodes of the graph allows complete coverage of the structure. This is equivalent to solving a variant of the art gallery problem. Then, a minimum-cost closed walk along the graph which fully covers the structure is searched for in the graph. This second step involves solving a variant of the traveling salesman problem. By favoring a random sampling method, they reduce the computational burden necessary to face the high-dimensionality of the problem. It should be noted that the generated paths cover cluttered spaces where complex structures such as shafts and rudders are present. The approach is validated using sensor imagery of real vessels and with experiments conducted at sea. Furthermore, a method for smoothing and shortening the paths initially generated



(a) Feasible tour for full coverage of a ship running gear. The tour is 176 m in length and contains 121 nodes.



(b) Tour of (a) after applying the refinement procedure. The shortened tour is 102 m in length and contains 97 configurations.

Fig. 25. Full-coverage inspection paths obtained with the method in [12].

is provided. This procedure can be incrementally applied while the computation time allows. Fig. 25 shows examples of planned inspection paths for a ship hull with and without smoothing.

As discussed above, the approach by Englot and Hover [12] first generates a set of view configurations that completely cover the target surface (by solving an instance of the art gallery problem) and then finds a path that connects them (by solving an instance of the traveling salesman problem). This might pose a problem for robots with differential constraints, given that a path connecting to a given view configuration might be infeasible. To tackle this problem, [79] presented a random sampling-based algorithm that incrementally explores the robot's configuration space while constructing an inspection path until all points on the target surface are guaranteed to be covered. In contrast to the aforementioned approaches, which first plan a set of view configurations that cover the target environment, their algorithm generates view configurations and at the same time validates the feasibility of the path connecting them. Only view configurations reached by feasible paths are incorporated in the final coverage path. Additionally, this method is probabilistically optimal with respect to a given cost function. The method is validated in simulation.

Given the wide variety of structures that are able to handle, these approaches constitute the state of the art in coverage of complex 3-dimensional structures.

9. Optimal coverage

Work addressing the optimality of the generated coverage paths, in terms such as path length and time to completion, appears in the CPP literature. Notice that it is only possible to find an optimal solution for an *a priori* known environment, or partially known at least, since an antagonistic example can always be found for a sensor-based approach. Hence optimal coverage methods are classified as off-line methods.

Huang presented an optimal line-sweep based method for cellular decomposition algorithms in planar spaces [80]. This approach produces an optimal length coverage path by allowing different sweep directions in the lawnmower paths used to cover each cell. The main idea is to minimize the number of turns in the path, as each turn typically implies the added cost of the robot decelerating and accelerating again after the turn. This is achieved by allowing a different sweep direction in each cell. The number of turns is minimized by sweeping each cell in parallel to its maximal altitude axis. That is, the method intends to maximize the length of the laps in the zigzag pattern in order to minimize the number of turns. However, this approach does not take into account the cost of traveling from cell to cell. The method is validated in simulation.

Jimenez et al. [81] proposed to use a genetic algorithm to achieve optimal coverage. In this proposal, the workspace and obstacles are assumed to be polygonal and known beforehand. Then, the free space is divided in subregions using the trapezoidal cellular decomposition method [25,26]. Finally, a genetic algorithm is used to plan an optimal path that covers all the subregions. This proposal is tested in simulation.

Mannadiar and Rekleitis proposed an algorithm based on the Boustrophedon cellular decomposition that achieves complete coverage of known spaces while minimizing the path of the robot [82]. The algorithm encodes the cells to be covered as edges of the Reeb graph. Then, the optimal solution to the Chinese Postman Problem is used to calculate an Euler tour, which guarantees complete coverage of the available free space while minimizing the coverage path length.

Xu et al. [83] presented an application of the optimal Morse-based boustrophedon decomposition method [82] for unmanned aerial vehicles (UAVs). First, they generate an optimal exhaustive walk through the adjacency graph of the cell decomposition of the terrain. Then, they cover each cell with zigzag motions taking into account the kinematic constraints of the vehicle, as the fixed-wing UAV they use has non-holonomic constraints. Extensive experimental results in simulation validate the presented system, along with data from over 100 km of coverage flights using a real fixed-wing aircraft.

10. Coverage under uncertainty

In many scenarios, the lack of an absolute localization system such as GPS makes the robot accumulate drift, and hence a growing uncertainty about its pose. Although the topological representations such as the adjacency graph are tolerant to localization error, the performance of coverage algorithms, even if using such representations, is still affected [84,20]. This is because the amount of coverage within a cell depends on the direction of the zigzag pattern.

Recent advances in SLAM (simultaneous localization and mapping) have greatly improved robot localization. SLAM uses statistical techniques to correct the robot's pose (position and orientation) estimation. However, the problem of correcting the robot's pose while performing coverage has been only addressed in few work.

Acar and Choset [85], propose to plan the paths of their sensor-based Morse decomposition approach by relying on the boundaries of each cell, hence minimizing the dead-reckoning error.

Tully et al. [86] used a fleet of three robots, each one of them equipped with a red ball (easily detectable using standard computer vision techniques) to follow a strategic path in formation to minimize the localization error. The mentioned robot fleet is shown in Fig. 26. The path consists of a series of steps, or leaps. In each leap, two robots are static and serve the third one as beacons, and this later one advances. The robots successively interchange their roles. Real experiments show a minimization of the localization error, reporting one of the most successful 2-dimensional

Fig. 26. Three robots used for the experimental evaluation of the leap-frog localization and coverage strategy.

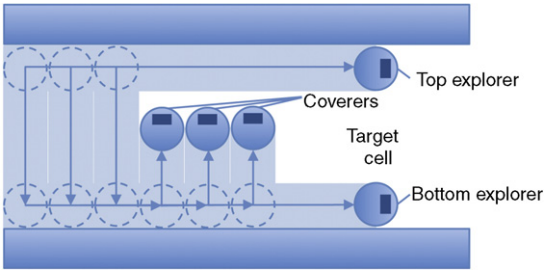


Fig. 27. Explorer/coverer approach, where two explorer robots outline the top and bottom boundaries while the remaining robots (coverers) execute simple back-and-forth coverage of the target cell.

coverage experiments to date. However, obstacles are not considered in this work.

Kim [87] propose an active SLAM approach to coverage path planning for ship hull inspection in a 3D scenario. The proposed algorithm drives the robot along a pre-planned coverage trajectory on the ship hull, and during trajectory execution the robot selects candidate locations that, once revisited, can help reduce the robot's pose uncertainty. The algorithm chooses to revisit a candidate location once the pose uncertainty surpasses a user-provided threshold, and otherwise follows the pre-planned path.

Bretl and Hutchinson [88] suggest a way to plan modified coverage paths for a mobile robot whose position and velocity are subject to bounded error. Assuming a worst-case model of uncertainty they are able to guarantee complete coverage. This guarantee comes at the cost of a longer path, since paths generated by their algorithm include retracing. Nonetheless, this work provides the first guaranteed coverage results for the case of bounded position and velocity error.

In the context of marine robotics, [89] presented a coverage path planning technique for vast, open sea areas which minimizes the robot's position uncertainty along the path. This technique is especially targeted for bathymetric mapping applications and respects application constraints such as the desire to survey in parallel tracks and to avoid turns in the target area to maximize sonar measurements quality. The method uses the saliency on an *a priori* map to predict how the terrain will affect the robots belief at every point on the target area. Based on this magnitude, an algorithm provided in this work computes the order in which to trace parallel tracks to cover the target area minimizing the overall uncertainty along the path. A particle filter keeps track of the robot's position uncertainty during the planning process and, in order to find useful loop- closures for mapping, crossing tracks that visit salient locations are added when the uncertainty surpasses a user-provided threshold. The method is tested on real-world datasets and results show that it offers benefits over a standard lawnmower-type path.

11. Multi-robot methods

There are advantages in using multiple robots in a CPP task. Using multiple robots clearly decreases the time to complete the task due to workload division. But a team of robots can go further, for example using each other as beacons to minimize localization error. Additionally, using multiple robots improves robustness, as failure of some members of the robot team can be compensated by others. There exist a number of multi-robot CPP proposals in the literature. Most approaches extend single-robot ideas presented above to multiple robots by using a strategy to divide the workload. In this section, we discuss multi-robot coverage methods based on the single-robot boustrophedon decomposition, on spanning trees, on the biologically inspired neural-network approach and on the graph-based approach. Nonetheless, there are genuine multi-robot approaches which are not based in any particular single-robot algorithm, which are also discussed below.

11.1. Multi-robot boustrophedon decomposition

Rekleitis et al. [90] presented a collection of algorithms for the complete coverage path planning problem using a team of mobile robots on an unknown environment (on-line). Their algorithms aim to minimize repeated coverage. The algorithms use the same planar cellular decomposition as the Boustrophedon single robot coverage algorithm, but provide extensions to handle how robots cover a single cell, and how robots are allocated among cells. Their solution takes into account communication restrictions among the members of the team. To achieve coverage in line-of-sight-only communications, the robots take two roles: some members, called explorers, cover the boundaries of the current target cell, while the other members, called coverers, perform simple back-and-forth motions to cover the remainder of the cell, as shown in Fig. 27. For task/cell allocation among the robots, a greedy auction mechanism is used. Experimental results from different simulated and real environments are provided to illustrate their proposed approach.

11.2. Multi-robot contact sensor-based coverage of rectilinear environments

A strategy for covering rectilinear environments using robots equipped only with contact sensors was proposed in [91]. The strategy is based on CC_R, discussed in Section 5, and is called DC_R. DC_R decouples cooperation and coverage by executing CC_R individually on each robot and adding a higher-level coordinator, termed the *overseer*, which is in charge of controlling the cooperation among the robots. The overseer operates in such a way that coverage directed by CC_R on each robot can continue without CC_R being aware that cooperation is occurring. A proof of completeness for DC_R is provided in this work.

11.3. Multi-robot spanning tree coverage

The STC method was generalized to multi-robot teams by Hazon and Kaminka [92] using an heuristic approach. They termed their algorithm MSTC. Zheng et al. [93] presented an improved (with respect to MSTC) method to find a coverage tree for a team of robots to cover known terrain with a team of robots. In this work they also provide an upper bound on the performance of a multi-robot coverage algorithm on known terrain, guaranteeing a performance of at most eight times the optimal cost. Their reported experimental results show their method to perform significantly better than MSTC. Agmon et al. [94] propose a spanning tree construction algorithm that provides efficient paths in terms of distance. The spanning tree construction algorithm can be used as a base for MSTC. An extension of MSTC to terrain with non-uniform traversability (that is, terrain where traversing certain areas is costlier than others) was presented in [95]. Hazon et al. [96] presented an on-line, robust version of MSTC. They show analytically that the algorithm

is robust, guaranteeing coverage as log as a single robot is able to move. Empirical results validating the algorithm are reported.

A more recent off-line spanning tree-based multi-robot coverage method presented by Fazli et al. [97] deals with the case where the robots have a limited visibility range. This approach is also shown to be complete and robust with respect to robot failure.

11.4. Multi-robot neural-network-based coverage

Luo and Yang [98] presented a straightforward adaptation of the biologically inspired neural network approach for coverage tasks to multi-robot scenarios where the robots see each other as moving obstacles. In a later work [99], an extension was provided to avoid deadlock situations between the robots. Their approach is validated in simulation.

11.5. Multi-robot graph-based and boundary coverage

Extensions of the graph-based techniques discussed in Section 7 for coverage using multiple robots are also provided in [70]. In a previous multi-robot graph-based coverage approach, [100] discuss a 2-dimensional boundary coverage algorithm for multiple robots. It is worth mentioning that, as in the majority of 3-dimensional coverage methods discussed in Section 8, this work focuses on covering only the boundary of the target environment. In the multi-robot boundary coverage problem, introduced in this work, a team of robots must inspect all points on the boundary of the 2-dimensional target environment. A motivating application of the multi-robot boundary coverage problem is inspection of separated blade surfaces inside a turbine. The boundary coverage problem is converted into an equivalent graph representation where a heuristic search is used to plan the inspection routes of every robot. The planned routes provide complete coverage of the boundary while balancing inspection load among the robots. The algorithm is validated in simulations.

11.6. Bio-inspired multi-robot coverage

Several multi-robot coverage path planning proposals have been presented which are inspired by behaviors found in nature. Many of them are inspired by the ant behavior, using evaporating traces to achieve an emergent coverage behavior [101–103]. In [104] two algorithms are presented which are based on the premise that to achieve coverage the team of robots must "spread out" over the environment. The authors note that "this premise is loosely inspired by the diffusive motion of fluid particles". Using these algorithms robots perform obstacle avoidance and at the same time are mutually repelled by each other within their sensor range. These bio-inspired works are validated in simulation, but their practical application has been very limited up to date.

11.7. Multi-robot coverage for aerial robotics

A considerable body of research has addressed multi-robot coverage path planning for fleets of aerial robots, taking into account the particulars of this domain. Notice that in the works discussed below it is assumed that the vehicles fly at a safe altitude, and as a result obstacles are not considered.

Ahmadzadeh et al. [105] proposed a coverage algorithm for surveillance using a fleet of UAVs. Their proposal takes into account the limited maneuverability of the aerial platforms and visibility constraints on the body-fixed cameras imposed by the application at hand. The problem is posed using the integer programming (IP) formalism, which provides a convenient representation for the aforementioned constraints. The solution of the IP problem instance produces a control policy for the UAV fleet to accomplish the surveillance task operating within the constraint limits. The efficacy of this approach is validated by simulation and experimental results.

Maza and Ollero [106] proposed a terrain coverage strategy using a heterogeneous fleet of UAVs. First, their method generates a polygonal partition of the target area. The partition takes into account the capabilities of each individual vehicle, such as flight endurance and range. Each polygon in the partition is assigned to a UAV which will cover it using a zigzag pattern. Each vehicle plans its zigzag pattern according to the geometric characteristics of its assigned polygonal area to determine a sweep direction that minimizes the number of turns. An important consideration in this work is low complexity of the algorithms used, seeking operation in near-real time. The proposed method is validated in simulation.

Targeting remote sensing in agriculture, [107] presented an approach to area coverage using fleets of mini aerial robots. Regarding multi-robot coverage, they first present a task scheduler to partition the global target area into k nonoverlapping subtasks for the k UAVs. In contrast with the previous work, this partition procedure is based on a negotiation process in which each robot claims covering as much area as possible, rather than on geometric considerations. Second, the wavefront algorithm discussed in 6.1 is used to cover each subarea.

12. Conclusion

In this paper, we have seen that the coverage path planning problem has been addressed using many different approaches. For planar spaces, the trapezoidal decomposition guarantees complete coverage for a known polygonal environment. An improvement to the trapezoidal decomposition is the "classical" boustrophedon decomposition, which generates shorter complete coverage paths for the same class of environments. The Morse-based cellular decomposition provides complete coverage paths for environments whose obstacle boundaries are differentiable. A method to detect the critical points that determine the cell boundaries using range sensor information allows to perform Morse-based cellular decomposition coverage on-line. Furthermore, Morse decomposition allows generation of different coverage patterns, such as spiral patterns, that can simplify the path following to vehicles with motion constraints.

However, the Morse-based cellular decomposition method cannot handle rectilinear environments, and the cyclic rectangular paths used to detect all the critical points include retracing, which makes them longer than a standard zigzag path. This limitations are overcome by the landmark-based topological coverage approach. This method uses a cellular decomposition based on natural landmarks of the environment which determine the cell boundaries. An algorithm is given to perform coverage on-line on unknown environments using this technique.

For the particular case of robots with only contact sensors (i.e., with no range sensing capabilities) operating in rectilinear environments, the CC_R algorithm guarantees complete on-line coverage.

Grid-based methods such as the wavefront algorithm, the Spiral-STC algorithm and its derivatives, and the described neural-network-based and hexagonal decomposition approaches, provide complete coverage on a discretized representation of the target environment. However, the grid representation of the environment used is highly sensitive to localization error and incurs an exponential memory consumption. On the other hand, it is easy to create and operate with a grid map. It is worth noticing, as a unique capability among the reviewed methods, that the discussed neural network-based methods are able to handle environments with moving obstacles.

Table 1
Survey summary.

Survey summary					
Category	Approach	Reference(s)	On/off-line	Environments handled	Remarks
Classical exact cell decomposition	Trapezoidal decomposition	[26]	Off-line	Polygonal	Lays the concept of using events to determine cell divisions, a concept a number of other approaches are based upon.
	Boustrophedon decomposition	[23]	Off-line	Polygonal	Generates less cells and hence shorter paths than the trapezoidal decomposition.
Morse-based cell decomposition	Morse decomposition and cycle algorithm	[24]	On-line	Polygonal and differentiable boundaries (non-rectilinear)	Allows for generation of different decomposition and coverage path patterns.
	Morse decomposition + GVD	[40]	On-line	Polygonal and differentiable boundaries (non-rectilinear)	Avoids generation unnecessary zigzag paths in narrow environments.
Natural landmark-based topological coverage	Landmark-based coverage algorithm	[45]	On-line	Generic planar obstacles	Handles a large variety of environments (including rectilinear environments).
Contact sensor-based coverage	Rectilinear decomposition (CC_R algorithm)	[48]	On-line	Rectilinear	Targeted for robots equipped only with contact sensors.
Grid-based coverage	Wavefront algorithm	[55]	Off-line	Grid-discretized	Simple, easy to implement algorithm.
	Spiral-STC algorithm	[57]	On-line	Grid-discretized	Minimizes repeated coverage by visiting each grid cell only once.
	Neural network	[61,62,64]	On-line	Grid-discretized	Handles dynamic obstacles.
	Hexagonal grid	[69]	On-line	Grid-discretized	Maximizes information gain along the path.
Graph coverage	Various graph search algorithms	[70]	On-line	Environmental constraints	Applies to environments that can be represented as a graph, such as a street or road network.
3-dimensional coverage	Planar terrain covering algorithm applied at successive depths	[3]	On-line	Projectively planar (2.5D)	Theoretically proven, however no details on how to detect the inlets used by the algorithm are provided.
	3D cellular decomposition	[72]	On-line	Closed, orientable surfaces embedded in $\mathbb{R}^3$	Theoretically proven. Demonstrated in simulation for simple 3D surfaces (polyhedra).
	Hierarchical segmentation of pseudo-extruded surfaces	[108]	Off-line	Car-like parts (pseudo-extruded surfaces)	Specifically targeted for coverage of automotive parts. Generates paths which optimize paint deposition uniformity.
	Coverage path generation on simplified 3D surfaces	[75]	On-line	Urban environments simplified as hemispheres and cylinders	Suitable for covering urban structures with sufficient clearance between them.
	Coverage of bathymetric surfaces	[76]	Off-line	Bathymetric (elevation) maps	Provides fair imaging angles on rugged terrain.
	Coverage for arable farming	[77]	Off-line	Elevation maps	Minimizes application-specific costs.
	Random sampling-based	[12]	Off-line	Complex 3D structures	Allows for coverage of complex 3-dimensional structures such as a ship propeller.
	Random sampling-based with differential constraints	[79]	Off-line	Complex 3D structures	Can handle differential constraints and probabilistically guarantees optimality.

(*continued on next page*)

Some methods aimed to cover 3-dimensional environments have been reviewed. Hert's algorithm can completely cover projectively planar 3-dimensional environments. However, details on how to detect the inlets used by the algorithm using sensor data are not provided, making it difficult to implement. Modular approaches, such as the coverage methods targeted for spray-painting tasks proposed by Atkar et al. or the simplified model of a urban environment used by Cheng et al. can achieve complete coverage of certain 3-dimensional environments. The method for coverage of bathymetric surfaces by Galceran et al. and the coverage algorithm for arable farming by Jin et al. provide coverage of 3-dimensional environments taking into account application-specific constraints. However, in confined 3-dimensional areas where a robot cannot go through the spaces between component structures, or occluded areas only visible from a reduced set of viewpoints, these modular approaches do not suffice. To overcome this limitation, Englot et al. introduced a sampling-based coverage algorithm to achieve complete sensor coverage of complex, 3-dimensional structures. The paths generated using this method are able to cover cluttered spaces where complex structures such as shafts and rudders are present. The approach is validated using triangular mesh models constructed using sensor imagery of real-world vessels. Also building upon the idea of sampling-based coverage, Papadopoulos et al. presented an algorithm that generates coverage paths for complex 3-dimensional structures suitable for vehicles with differential constraints. Furthermore, their algorithm is proven probabilistically optimal with respect to a given cost function. This approaches constitute the state of the art in coverage of complex 3-dimensional structures.

We have discussed several approaches to generate optimal coverage paths in planar spaces and to reduce localization error while performing coverage.

Finally, we have reviewed some multi-robot coverage methods in which time to completion is reduced by dividing the workload among the individual robot team members, besides providing increased robustness guarantees.

The features of the most relevant CPP methods reviewed in this article are summarized in Table 1. For each method (rows), the table underlines (columns from left to right) its category, its approach, its main bibliographical reference, whether it can be used on-line or not, the kind of environments it can handle and some remarks.

Table 1 (*continued*)

Survey summary					
Category	Approach	Reference(s)	On/off-line	Environments handled	Remarks
Optimal coverage	Cell decomposition with variable sweep direction	[80]	Off-line	Polygonal	Takes into account "cell height" to select the optimal sweep direction.
	Trapezoidal decomposition + genetic algorithm	[81]	Off-line	Polygonal	A genetic algorithm quickly finds a specific coverage paths among the cells.
	Morse decomposition + optimal adjacency graph traversal	[82]	On-line	Polygonal and differentiable boundaries (non-rectilinear)	Finds an optimal walk through the adjacency graph.
Coverage under uncertainty	Exploiting critical points	[85]	On-line	Polygonal and differentiable boundaries (non-rectilinear)	Efficiently increases actual percent coverage achieved.
	Active SLAM	[87]	On-line	Ship hull	Decides when to revisit a salient feature when executing coverage to reduce uncertainty.
Multi-robot coverage	Leap-frog strategy	[86]	On-line	Obstacles not considered	Robots in a team use each other as beacons alternatively.
	Modified boustrophedon paths	[88]	Off-line	Planar	Guarantees complete coverage under bounded position and velocity error.
	Low uncertainty CPP for marine surveys	[89]	Off-line	Marine	Uses saliency to determine parallel track order and key salient points to revisit.
	Boustrophedon-based	[90]	On-line	Polygonal and differentiable boundaries (non-rectilinear)	Extension of Morse decomposition to multi-robot teams.
	Contact sensor-based	[91]	On-line	Rectilinear	Extension of CC_R using a decoupled high-level coordinator.
	Spiral-STC-based	[93]	Off-line	Grid-discretized	Extension of Spiral-STC to multi-robot teams.
	Spiral-STC-based	[96]	On-line	Grid-discretized	On-line extension of Spiral-STC to multi-robot teams.
	Neural-network-based	[99]	On-line	Grid-discretized	Robots see each other as moving obstacles.
	Boundary coverage	[100]	Off-line	2D	Focused on coverage of obstacle boundaries.
	Bio-inspired	[102] and others	On-line	2D	Validated in simulation, but of limited practical application.
	Aerial	[105] and others	Off-line	Aerial (obstacle-free)	Account for limited maneuverability and for load balancing in heterogeneous teams.

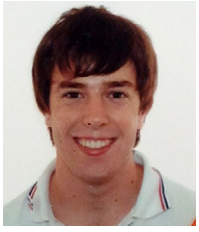
Probabilistic sampling-based algorithms have revolutionized the state of the art in path planning in the recent years, and they have proven to be extremely powerful as demonstrated in the work by Englot et al. on ship hull inspection. Therefore, exploiting these techniques opens the door to developing algorithms able to realize coverage tasks of unprecedented complexity. On the other hand, in real-world applications, a robot often does not have perfect knowledge about its location. In this situation, incorporating uncertainty in future location estimates in the planning phase can significantly improve motion performance. Although several research works have explored taking uncertainty into account in path planning problems, little attention has been given to incorporating uncertainty in coverage path planning methods. Hence, this remains as an open issue for further research.

References

[1] F. Yasutomi, M. Yamada, K. Tsukamoto, Cleaning robot control, in: Proc. Conf. IEEE Int Robotics and Automation, 1988, pp. 1839–1841.
[2] P. Atkar, A.L. Greenfield, D.C. Conner, H. Choset, A. Rizzi, Uniform coverage of automotive surface patches, The International Journal of Robotics Research 24 (11) (2005) 883–898.
[3] S. Hert, S. Tiwari, V. Lumelsky, A terrain-covering algorithm for an auv, Autonomous Robots 3 (1996) 91–119.
[4] D.W. Gage, Randomized search strategies with imperfect sensors, in: Proc. SPIE, Mobile Robots VIII—Int. Soc. Optical Engineering, Boston, MA, 1994, pp. 270–279.
[5] H. Najjaran, N. Kircanski, Path planning for a terrain scanner robot, in: Proc. 31st Int. Symp. Robotics, Montreal, QC, Canada, 2000, pp. 132–137.
[6] E.U. Acar, H. Choset, Y. Zhang, M. Schervish, Path planning for robotic demining: robust sensor-based coverage of unstructured environments and probabilistic methods, International Journal of Robotics Research 22 (7–8) (2003) 441–466.
[7] Z.L. Cao, Y. Huang, E.L. Hall, Region filling operations with random obstacle avoidance for mobile robotics, Journal of Robotic Systems 5 (2) (1988) 87–102.
[8] M. Bosse, N. Nourani-Vatani, J. Roberts, Coverage algorithms for an under-actuated car-like vehicle in an uncertain environment, in: Proc. IEEE Int. Robotics and Automation Conf., 2007, pp. 698–703.
[9] M. Ollis, A. Stentz, First results in vision-based crop line tracking, in: Proc. Conf. IEEE Int. Robotics and Automation, Vol. 1, 1996, pp. 951–956.
[10] M. Ollis, A. Stentz, Vision-based perception for an automated harvester, in: Proc. IEEE/RSJ Int. Intelligent Robots and Systems IROS'97. Conf., Vol. 3, 1997, pp. 1838–1844.
[11] M. Farsi, K. Ratcliff, J.P. Johnson, C.R. Allen, K.Z. Karam, R. Pawson, Robot control system for window cleaning, in: Proc. American Control Conf., Vol. 1, 1994, pp. 994–995.
[12] B. Englot, F. Hover, Sampling-based coverage path planning for inspection of complex structures, in: International Conference on Automated Planning and Scheduling, ICAPS, 2012.
[13] E.M. Arkin, R. Hassin, Approximation algorithms for the geometric covering salesman problem, Discrete Applied Mathematics 55 (3) (1994) 197–218.
[14] E.M. Arkin, S.P. Fekete, J.S. Mitchell, Approximation algorithms for lawn mowing and milling, Computational Geometry 17 (1–2) (2000) 25–50. URL: http://www.sciencedirect.com/science/article/pii/S0925772100000158.
[15] S.M. LaValle, Planning Algorithms, Cambridge University Press, 2006, pp. 130–131 The Configuration Sace (Chapter 4).
[16] J.H. Reif, Z. Sun, An efficient approximation algorithm for weighed region shortest path problem, in: Algorithmic and Computational Robotics: New Directions, A.K. Peters, 2001, pp. 191–203 (Chapter).
[17] T. Shermer, Recent results in art galleries (geometry), Proceedings of the IEEE 80 (9) (1992) 1384–1399.
[18] F. Li, R. Klette, An approximate algorithm for solving the watchman route problem, in: G. Sommer, R. Klette (Eds.), Robot Vision, in: Lecture Notes in Computer Science, vol. 4931, Springer, Berlin, Heidelberg, 2008, pp. 189–206.
[19] W.-P. Chin, S. Ntafos, Shortest watchman routes in simple polygons, Discrete & Computational Geometry 6 (1) (1991) 9–31. http://dx.doi.org/10.1007/BF02574671.
[20] H. Choset, Coverage for robotics—a survey of recent results, Annals of Mathematics and Artificial Intelligence 31 (2001) 113–126.
[21] J. Palacin, T. Palleja, I. Valganon, R. Pernia, J. Roca, Measuring coverage performances of a floor cleaning mobile robot using a vision system, in: Proceedings of the 2005 IEEE International Conference on, Robotics and Automation, 2005. ICRA 2005. 2005, pp. 4236–4241.
[22] V.J. Lumelsky, S. Mukhopadhyay, K. Sun, Dynamic path planning in sensor-based terrain acquisition, IEEE Transactions on Robotics and Automation 6 (4) (1990) 462–472.
[23] H. Choset, P. Pignon, Coverage path planning: the boustrophedon cellular decomposition, in: Proceedings of International Conference on Field and Service Robotics, 1997.
[24] E.U. Acar, H. Choset, A.A. Rizzi, P.N. Atkar, D. Hull, Morse decompositions for coverage tasks, International Journal of Robotics Research 21 (4) (2002) 331–344.

[25] J.C. Latombe, Robot Motion Planning, Kluwer Academic Publishers, 1991.
[26] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun, Principles of Robot Motion: Theory, Algorithms, and Implementation, The MIT Press, 2005.
[27] T. Oksanen, A. Visala, Coverage path planning algorithms for agricultural field machines, Journal of Field Robotics 26 (8) (2009) 651–668.
[28] H. Choset, E. Acar, A.A. Rizzi, J. Luntz, Exact cellular decompositions in terms of critical points of Morse functions, in: Proc. IEEE Int. Conf. Robotics and Automation ICRA'00, Vol. 3, 2000, pp. 2270–2277.
[29] J. Milnor, Morse Theory, Princeton University Press, 1963.
[30] E.U. Acar, H. Choset, Robust sensor-based coverage of unstructured environments, in: Proc. IEEE/RSJ International Intelligent Robots and Systems Conference, Vol. 1, 2001, pp. 61–68.
[31] E. Acar, H. Choset, Sensor-based coverage of unknown environments: incremental construction of morse decompositions, International Journal of Robotics Research 21 (4) (2002) 345–366.
[32] J.F. Canny, Constructing roadmaps of semi-algebraic sets i: completeness, Artificial Intelligence 37 (1–3) (1988) 203–222.
[33] J.F. Canny, An opportunistic global path planner, Algorithmica 10 (1993) 102–120.
[34] H. Choset, Coverage of known spaces: the boustrophedon cellular decomposition, Autonomous Robots 9 (3) (2000) 247–253.
[35] E. Galceran, M. Carreras, Efficient seabed coverage path planning for asvs and auvs, in: Proc. IEEE/RSJ International Intelligent Robots and Systems Conference, 2012.
[36] E.U. Acar, H. Choset, Critical point sensing in unknown environments, in: Proceedings of the 2000 IEEE International Conference on Robotics & Automation, 2000.
[37] G. Reeb, Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique, Comptes Rendus de l'Académie des Sciences 222 (1946) 847–849.
[38] E. Garcia, P.G. de Santos, Mobile-robot navigation with complete coverage of unstructured environments, Robotics and Autonomous Systems 46 (2004) 195–204.
[39] E. Acar, H. Choset, P. Atkar, Complete sensor-based coverage with extended-range detectors: a hierarchical decomposition in terms of critical points and voronoi diagrams, in: 2001 IEEE/RSJ International Conference on, Intelligent Robots and Systems, 2001. Proceedings, Vol. 3, 2001, pp. 1305–1311.
[40] E.U. Acar, H. Choset, J.Y. Lee, Sensor-based coverage with extended range detectors, IEEE Transactions on Robotics 22 (1) (2006) 189–198.
[41] H. Choset, J. Burdick, Sensor-based exploration: the hierarchical generalized voronoi graph, The International Journal of Robotics Research 19 (2) (2000) 96–125. URL: http://ijr.sagepub.com/content/19/2/96.abstract.
[42] M. de Berg, O. Cheong, M. v Kreveldand, M. Overmars, Computational Geometry, third ed., Springer, 2008.
[43] H. Choset, S. Walker, K. Eiamsa-Ard, J. Burdick, Sensor-based exploration: incremental construction of the hierarchical generalized voronoi graph, The International Journal of Robotics Research 19 (2) (2000) 126–148. URL: http://ijr.sagepub.com/content/19/2/126.abstract.
[44] H. Choset, D. Kortenkamp, Path planning and control for aercam, a free-flying inspection robot in space, ASCE Journal of Aerospace Engineering 12 (2) (1999) 74–81.
[45] S.C. Wong, B.A. MacDonald, A topological coverage algorithm for mobile robots, in: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, IROS 2003, Vol. 2, 2003, pp. 1685–1690.
[46] S.C. Wong, B.A. MacDonald, Complete coverage by mobile robots using slice decomposition based on natural landmarks, in: C. Zhang, H.W. Guesgen, W.-K. Yeap (Eds.), PRICAI 2004: Trends in Artificial Intelligence, in: Lecture Notes in Computer Science, vol. 3157, Springer, Berlin, Heidelberg, 2004, pp. 683–692. http://dx.doi.org/10.1007/978-3-540-28633-2_72.
[47] S.C. Wong, Qualitative topological coverage of unknown environments by mobile robots, Ph.D. Thesis, The University of Auckland, 2006.
[48] Z.J. Butler, A.A. Rizzi, R.L. Hollis, Contact sensor-based coverage of rectilinear environments, in: Proc. IEEE Int Intelligent Control/Intelligent Systems and Semiotics Symposium, 1999, pp. 266–271.
[49] H. Moravec, A. Elfes, High resolution maps from wide angle sonar, in: Proc. IEEE Int. Conf. Robotics and Automation, Vol. 2, 1985, pp. 116–121.
[50] A. Elfes, Sonar-based real-world mapping and navigation, IEEE Journal of Robotics and Automation 3 (3) (1987) 249–265.
[51] S. Thrun, Learning metric-topological maps for indoor mobile robot navigation, Artificial Intelligence 99 (1) (1998) 21–71. http://dx.doi.org/10.1016/S0004-3702(97)00078-7.
[52] J. Castellanos, J. Tardos, G. Schmidt, Building a global map of the environment of a mobile robot: the importance of correlations, in: 1997 IEEE International Conference on, Robotics and Automation, 1997. Proceedings, Vol. 2, 1997, pp. 1053–1059.
[53] S. Thrun, Robotic mapping: a survey, in: Exploring Artificial Intelligence in the New Millennium, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003, pp. 1–35. URL: http://portal.acm.org/citation.cfm?id=779345.
[54] J.S. Oh, Y.H. Choi, J.B. Park, Y. Zheng, Complete coverage navigation of cleaning robots using triangular-cell-based map, IEEE Transactions on Industrial Electronics 51 (3) (2004) 718–726.
[55] A. Zelinsky, R.A. Jarvis, J.C. Byrne, S. Yuta, Planning paths of complete coverage of an unstructured environment by a mobile robot, in: Proceedings of International Conference on Advanced Robotics, 1993, pp. 533–538.
[56] V. Shivashankar, R. Jain, U. Kuter, D. Nau, Real-time planning for covering an initially-unknown spatial environment, in: Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference, 2011.
[57] Y. Gabriely, E. Rimon, Spiral-stc: an on-line coverage algorithm of grid environments by a mobile robot, in: Proc. IEEE Int. Conf. Robotics and Automation, ICRA'02, Vol. 1, 2002, pp. 954–960.
[58] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra, C. Bustacara, Bsa: a complete coverage algorithm, in: Proc. IEEE Int. Conf. Robotics and Automation ICRA 2005, 2005, pp. 2040–2044.
[59] Y.-H. Choi, T.-K. Lee, S.-H. Baek, S.-Y. Oh, Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform, in: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2009, 2009, pp. 5788–5793.
[60] T.-K. Lee, S.-H. Baek, Y.-H. Choi, S.-Y. Oh, Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation, Robotics and Autonomous Systems 59 (10) (2011) 801–812. URL: http://www.sciencedirect.com/science/article/pii/S0921889011000996.
[61] C. Luo, S.X. Yang, D.A. Stacey, J.C. Jofriet, A solution to vicinity problem of obstacles in complete coverage path planning, in: Proc. IEEE Int. Conf. Robotics and Automation ICRA'02, Vol. 1, 2002, pp. 612–617.
[62] S.X. Yang, C. Luo, A neural network approach to complete coverage path planning, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34 (1) (2004) 718–724.
[63] L. Hodgkin, A.F. Huxley, A quantitative description of membrane current and its application to conduction and exitation in nerve, Journal of Physics London 117 (1952) 500–544.
[64] C. Luo, S. Yang, A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments, IEEE Transactions on Neural Networks 19 (7) (2008) 1279–1298.
[65] M. Yan, D. Zhu, An algorithm of complete coverage path planning for autonomous underwater vehicles, Key Engineering Materials 467–469 (2011) 1377–1385.
[66] X. Qiu, J. Song, X. Zhang, S. Liu, A complete coverage path planning method for mobile robot in uncertain environments, in: Proc. Sixth World Congress Intelligent Control and Automation WCICA 2006, Vol. 2, 2006, pp. 8892–8896.
[67] Y. Guo, M. Balakrishnan, Complete coverage control for nonholonomic mobile robots in dynamic environments, in: Proceedings 2006 IEEE International Conference on, Robotics and Automation, 2006, ICRA 2006, 2006, pp. 1704–1709.
[68] L. Paull, S. Saeedi, H. Li, V. Myers, An information gain based adaptive path planning method for an autonomous underwater vehicle using sidescan sonar, in: 2010 IEEE Conference on Automation Science and Engineering, CASE, 2010, pp. 835–840.
[69] L. Paull, S. SaeediGharahbolagh, M. Seto, H. Li, Sensor driven online coverage planning for autonomous underwater vehicles, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2012, pp. 2875–2880.
[70] L. Xu, Graph Planning for Environmental Coverage, Ph.D. Thesis, Carnegie Mellon University, 2011.
[71] T.-S. Lee, J.-S. Choi, J.-H. Lee, B.-H. Lee, 3-d terrain covering and map building algorithm for an auv, in: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2009, 2009, pp. 4420–4425.
[72] P.N. Atkar, H. Choset, A.A. Rizzi, E.U. Acar, Exact cellular decomposition of closed orientable surfaces embedded in r3, in: Proc. ICRA Robotics and Automation IEEE Int. Conf, Vol. 1, 2001, pp. 699–704.
[73] P. Atkar, H. Choset, A. Rizzi, Towards optimal coverage of 2-dimensional surfaces embedded in R3: choice of start curve, in: 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. IROS 2003. Proceedings, Vol. 4, 2003, pp. 3581–3587.
[74] P. Atkar, A. Greenfield, D. Conner, H. Choset, A. Rizzi, Hierarchical segmentation of surfaces embedded in r3 for auto-body painting, in: Proceedings of the 2005 IEEE International Conference on, Robotics and Automation, 2005, ICRA 2005, 2005, pp. 572–577.
[75] P. Cheng, J. Keller, V. Kumar, Time-optimal uav trajectory planning for 3D urban structure coverage, in: IEEE/RSJ International Conference on, Intelligent Robots and Systems, 2008, IROS 2008, 2008, pp. 2750–2757.
[76] E. Galceran, M. Carreras, Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor, in: Proc. International Conference on Robotics and Automation, 2013.
[77] J. Jin, L. Tang, Coverage path planning on three-dimensional terrain for arable farming, Journal of Field Robotics 28 (3) (2011) 424–440. http://dx.doi.org/10.1002/rob.20388. URL:.
[78] T. Danner, L. Kavraki, Randomized planning for short inspection paths, in: Proceedings. IEEE International Conference on Robotics and Automation 2000, ICRA'00, Vol. 2, 2000, pp. 971–976.
[79] G. Papadopoulos, H. Kurniawati, N.M. Patrikalakis, Asymptotically optimal inspection planning using systems with differential constraints, in: Proc. International Conference on Robotics and Automation, 2013.
[80] W.H. Huang, Optimal line-sweep-based decompositions for coverage algorithms, in: Proc. ICRA Robotics and Automation IEEE Int. Conf, Vol. 1, 2001, pp. 27–32.
[81] P.A. Jimenez, B. Shirinzadeh, A. Nicholson, G. Alici, Optimal area covering using genetic algorithms, in: Proc. IEEE/ASME International Conference Advanced Intelligent Mechatronics, 2007, pp. 1–5.

[82] R. Mannadiar, I. Rekleitis, Optimal coverage of a known arbitrary environment, in: Proc. IEEE Int Robotics and Automation (ICRA) Conf., 2010, pp. 5525–5530.
[83] A. Xu, P. Virie, I. Rekleitis, Optimal complete terrain coverage using an unmanned aerial vehicle, in: Proceedings of the 2011 IEEE International Conference on Robotics & Automation, 2011.
[84] M. Mazo, et al. Robust area coverage using hybrid control, in: TELEC 2004, Santiago de Cuba, 2004.
[85] E.U. Acar, H. Choset, Exploiting critical points to reduce positioning error for sensor-based navigation, in: Proc. IEEE Int. Conf. Robotics and Automation ICRA 2002, Vol. 4, 2002, pp. 3831–3837.
[86] S. Tully, G. Kantor, H. Choset, Leap-frog path design for multi-robot cooperative localization, in: A. Howard, K. Iagnemma, A. Kelly (Eds.), Field and Service Robotics, in: Springer Tracts in Advanced Robotics, vol. 62, Springer, Berlin, Heidelberg, 2010, pp. 307–317. http://dx.doi.org/10.1007/978-3-642-13408-1_28.
[87] A. Kim, Active visual SLAM with exploration for autonomous underwater navigation, Ph.D. Thesis, The University of Michigan, 2012.
[88] T. Bretl, S. Hutchinson, Robust coverage by a mobile robot of a planar workspace, in: Proc. International Conference on Robotics and Automation, 2013.
[89] E. Galceran, S. Nagappa, M. Carreras, P. Ridao, A. Palomer, Uncertainty-driven survey path planning for bathymetric mapping, in: Proc. IEEE/RSJ International Intelligent Robots and Systems Conference, 2013.
[90] I. Rekleitis, A. New, E. Rankin, H. Choset, Efficient boustrophedon multi-robot coverage: an algorithmic approach, Annals of Mathematics and Artificial Intelligence 52 (2009) 109–142. http://dx.doi.org/10.1007/s10472-009-9120-2.
[91] Z.J. Butler, A.A. Rizzi, R.L. Hollis, Complete distributed coverage of rectilinear environments, in: Proc. of the Workshop on the Algorithmic Foundations of Robotics, 2000.
[92] N. Hazon, G. Kaminka, Redundancy, efficiency and robustness in multi-robot coverage, in: Proceedings of the 2005 IEEE International Conference on, Robotics and Automation, 2005, ICRA 2005, 2005, pp. 735–741.
[93] X. Zheng, S. Jain, S. Koenig, D. Kempe, Multi-robot forest coverage, in: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, IROS 2005, 2005, pp. 3852–3857.
[94] N. Agmon, N. Hazon, G. Kaminka, Constructing spanning trees for efficient multi-robot coverage, in: Proceedings 2006 IEEE International Conference on, Robotics and Automation, 2006, ICRA 2006, 2006, pp. 1698–1703.
[95] X. Zheng, S. Koenig, Robot coverage of terrain with non-uniform traversability, in: IEEE/RSJ International Conference on, Intelligent Robots and Systems, 2007, IROS 2007, 2007, pp. 3757–3764.
[96] N. Hazon, F. Mieli, G. Kaminka, Towards robust on-line multi-robot coverage, in: Proceedings 2006 IEEE International Conference on, Robotics and Automation, 2006. ICRA 2006, 2006, pp. 1710–1715.
[97] P. Fazli, A. Davoodi, P. Pasquier, A. Mackworth, Complete and robust cooperative robot area coverage with limited range, in: 2010 IEEE/RSJ International Conference on, Intelligent Robots and Systems, IROS, 2010, pp. 5577–5582.
[98] C. Luo, S. Yang, A real-time cooperative sweeping strategy for multiple cleaning robots, in: Proceedings of the 2002 IEEE International Symposium on, Intelligent Control, 2002, 2002, pp. 660–665.
[99] C. Luo, S. Yang, D. Stacey, Real-time path planning with deadlock avoidance of multiple cleaning robots, in: IEEE International Conference on, Robotics and Automation, 2003. Proceedings. ICRA'03, Vol. 3, 2003, pp. 4080–4085.
[100] K. Easton, J. Burdick, A coverage algorithm for multi-robot boundary inspection, in: Proceedings of the 2005 IEEE International Conference on, Robotics and Automation, 2005, ICRA 2005, 2005, pp. 727–734.
[101] I.A. Wagner, M. Lindenbaum, A.M. Bruckstein, Distributed covering by ant-robots using evaporating traces, IEEE Transactions on Robotics and Automation 15 (5) (1999) 918–933.
[102] I.A. Wagner, Y. Altshuler, V. Yanovski, A.M. Bruckstein, Cooperative cleaners: a study in ant robotics, The International Journal of Robotics Research 27 (1) (2008) 127–151. URL: http://ijr.sagepub.com/content/27/1/127.abstract.
[103] R. Menezes, F. Martins, F.E. Vieira, R. Silva, M. Braga, A model for terrain coverage inspired by ant's alarm pheromones, in: Proceedings of the 2007 ACM Symposium on Applied Computing, SAC'07, ACM, New York, NY, USA, 2007, pp. 728–732. http://dx.doi.org/10.1145/1244002.1244164. URL:.
[104] M.A. Batalin, G.S. Sukhatme, Spreading out: a local approach to multi-robot coverage, in: Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems, 2002, pp. 373–382.
[105] A. Ahmadzadeh, J. Keller, A. Jadbabaie, V. Kumar, An optimization-based approach to time critical cooperative surveillance and coverage with unmanned aerial vehicles, in: International Symposium on Experimental Robotics, 2006.
[106] I. Maza, A. Ollero, Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms, in: Distributed Autonomous Robotic Systems, Vol. 6, Springer, 2007, pp. 221–230 (Chapter).
[107] A. Barrientos, J. Colorado, J. del Cerro, A. Martinez, C. Rossi, D. Sanz, J. Valente, Aerial remote sensing in agriculture: a practical approach to area coverage and path planning for fleets of mini aerial robots, Journal of Field Robotics 28 (5) (2011) 667–689.
[108] P. Atkar, D. Conner, A. Greenfield, H. Choset, A. Rizzi, Hierarchical segmentation of piecewise pseudoextruded surfaces for uniform coverage, IEEE Transactions on Automation Science and Engineering 6 (1) (2009) 107–120.



Enric Galceran is a Ph.D. student at the Underwater Robotics Research Center (CIRS) from the University of Girona (Spain). His thesis focuses in coverage path planning for marine vehicles. He holds B.S. (best GPA award) and M.S. degrees in Computer Engineering from the University of Girona. He was awarded the best final project prize in both degrees. Mr. Galceran participated in the 2010 (1st prize) and 2011 (2nd prize) editions of SAUC-E (Student Autonomous Underwater Challenge—Europe, a student competition of robotics). Currently, he has been a visiting researcher at NATO's Center for Maritime Research and Experimentation (La Spezia, Italy) and at the Robotics Institute in Carnegie Mellon University (Pittsburgh, PA, USA) under the supervision of Prof. Howie Choset.

Marc Carreras is an Associate Professor in the Computer Engineering Department at University of Girona (Spain), and member of the Computer Vision and Robotics Research Group (VICOROB) working in the Underwater Robotics Research Centre (CIRS). He holds a B.S. degree in Industrial Engineering (1998) and Ph.D. in Computer Engineering (2003, Best Ph.D. award) from the University of Girona. His research activity is mainly focused on underwater robotics in research topics such as intelligent control architectures, robot learning, path planning, AUV design, modeling and identification. He is currently involved in several European projects (FP7 STREP "TRIDENT", FP7 STREP "PANDORA", FP7 IP "MORPH") and National project (Spanish Ministry "TRITON") about autonomous underwater robotics. From 1999 till 2011, he has participated in 14 research projects (6 European and 8 National), he is author of more than 80 publications, and he has directed 2 Ph.D.s thesis. He led the participation of VICOROB in SAUC-E (Student European AUV competition) in 2006 (1st prize), 2010 (1st prize) and 2011 (2nd prize).