Squircular-CPP: A Smooth Coverage Path Planning Algorithm based on Squircular Fitting and Spiral Path

Mahdi Hassan[1], Dikai Liu[1], and Xiang Chen[2]

***Abstract*— Coverage path planning (CPP) is essential for applications such as robotic floor cleaning and high-pressure cleaning of surfaces. Smooth CPP algorithms have several benefits including smoother motion of the robot and the reduction of aggressive accelerations and decelerations resulting from sharp turns. In this paper, a novel smooth CPP algorithm is presented which is named Squircular-CPP. This algorithm proposes a squircular shape, which is an intermediate shape between the circle and the square, to fit a target area. Squircular-CPP can also fit a shape between the ellipse and the rectangle. The shape fitting is simple, fast, and analytical and doesn't require a preselection of the shape (i.e. square, circle, ellipse or rectangle). It enables and complements the creation of a smooth spiral path within the fitted shape. Several case studies are presented to demonstrate the effectiveness of the algorithm and to compare it against the popular boustrophedon-based coverage approach and the Deformable Spiral CPP (DSCPP) algorithm.**

I. INTRODUCTION

Many robotic applications, such as robotic floor cleaning and grit-blasting for surface preparation [1], require a coverage path planning (CPP) algorithm [2], [3] to plan a path over the target area. One important objective for many robotic coverage problems is the smoothness of the coverage path [3]–[6]. A sample of benefits that a smooth path might deliver include: (i) avoiding frequent or aggressive accelerations or decelerations of robot motion [4] (e.g. for energy efficiency [4] or to prevent long-term damage to certain robots [3]), and (ii) avoiding damage on the surfaces of structures (e.g. fatigue cracks from high-pressure blasting) due to operation below a velocity threshold resulting from sharp turns [7].

Suppose that a robot is tasked with covering the areas shown in Fig. 1(Left). For some coverage applications it may be desirable to cover such areas using smooth paths even if generating smooth paths may cause slight coverage outside the boundary of the target area. In other words, for some applications it is not a strict requirement for the robot to stay within the area for coverage; and therefore this flexibility can be exploited for the sake of generating smoother paths. An application here can be an Unmanned Aerial Vehicle (UAV) surveying an agricultural land where the UAV can go outside of the coverage area (since there may be no obstacles in the air) in favor of a smoother path. Note that this slight coverage outside of the boundary does not necessary create a longer or less efficient path, e.g. as shown in Fig. 2. Another sample application can

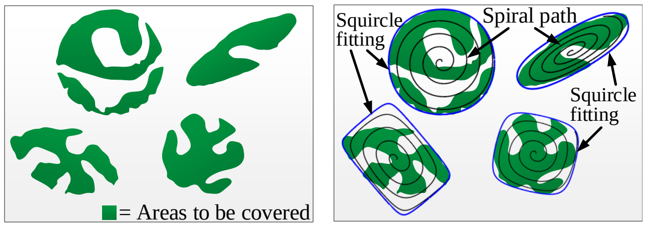

Fig. 1: (Left) Showing certain areas of a surface, colored in green, that are to be covered by a robot. (Right) Covering the target areas using the proposed smooth coverage path planning algorithm (Squircular-CPP) where first, a squircle (or a scaled squircle) fits the area of interest and then a spiral path is appropriately deformed to fit within the squircle.

be an intervention autonomous underwater vehicle (I-AUV), such as SPIR [7], [8], that is tasked with removing marine growth from surfaces of underwater structures using high-pressure blasting. Slightly covering outside of the target area is acceptable for this application as a trade-off to generating smoother path. Such a robot is required to generate smooth coverage paths so as to prevent damage to the structure [7]. As shown in Fig. 1(Right), spiral-like paths are appropriate; however, the creation of a smooth spiral path requires the target area to be approximated using a proper shape.

Therefore, the problem that this paper aims to address is as follows: *given a target area, how can a robot automatically determine an appropriate shape that approximates the area with the goal of enabling a smooth spiral path to be deformed to fit within the shape in a simple and computationally-efficient manner.* The deformation of the spiral path is to consider the maximum gap between spiral paths' laps and that the length of the spiral is not excessively long.

A novel algorithm is developed to address the above problem. The algorithm, named *Squircular-CPP*, uses the idea of fitting a squircle (an intermediate shape between the square and the circle) to the target area such that the fitted squircle is bounded by a Minimum Bounding Rectangle (MBR). The algorithm can also fit an intermediate shape between the ellipse and the rectangle.

The previous work [7] presented a Deformable Spiral CPP (DSCPP) algorithm that deforms a spiral path within a rectangle. However, using the proposed shape fitting algorithm, a shorter and smoother path can be generated. More specifically, the contributions of this paper are:

- This squircular shape fitting is done analytically and automatically without any prior selection of the shape (i.e. without preselection of square, circle, ellipse, or rectangle). To the best of authors' knowledge, this is the first squircular shape fitting algorithm. It is based on

[1] Mahdi Hassan and Dikai Liu are with the Centre for Autonomous Systems (CAS) at the University of Technology Sydney (UTS), 15 Broadway, Ultimo NSW 2007, Australia `Mahdi.Hassan@uts.edu.au`

[2] Xiang Chen is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada

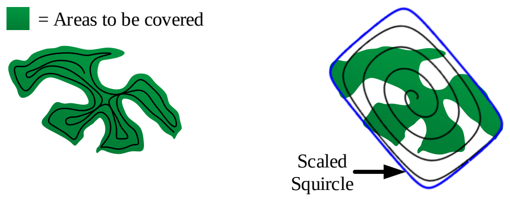

Fig. 2: (Left) An example coverage path to cover the target area while aiming to generate a smooth spiral-like path that is constrained within the area. (Right) Target area is approximated using a squircle as per the proposed Squircular-CPP algorithm to enable generating a smoother spiral path.

Fernandez-Guasti squircle and mapping, the shrunken squircle, fitting an MBR, and obtaining the closest point to the center of the MBR.

- The squircular shape fitting enables and complements the creation of a smoother and shorter spiral path not only because the squircle can better fit the target area but also because the spiral path can be made to gradually morph into a smoother shape.
- A measure is designed for quantifying smoothness which uses the curvature of path segments, similar to the work in [9]. Extensive comparison and analysis are carried out to quantify smoothness with respect to many squareness values (explained later) and MBR aspect ratios. Comparisons in terms of path length are also carried out. The resulting paths are compared to the simple boustrophedon path (also known as back-and-forth path and lawn-mover path) and the path generated through the DSCPP algorithm [7].

II. RELATED WORKS

Spiral paths are used for path smoothing and even though smooth spiral paths can be longer than square spiral paths or scan lines (boustrophedon paths), they are more energy-efficient since they don't contain sharp turns [4]. Spiral paths have been used for unmanned aircraft systems (UAS) [10], agricultural vehicles [11], autonomous underwater vehicles (AUV) [7], and others.

The work in [12] utilizes a backtracking spiral algorithm. The algorithm generates spiral paths for simple regions which are then linked together using a backtracking mechanism that ensures completeness. In a later work [13], procedures such as wall-following and return path by a virtual pipe were introduced to the original algorithm.

In [6], the aim is to reduce the number of turns to improve the completion time. The work guarantees completeness through linking simple spiral paths together. This linking is conducted using a Constrained Inverse Distance Transform (CIDT). An improved version of this algorithm is presented in [9] where a high-resolution grid-map is used to eliminate constraints on mobility, and a cardinal spline curve-model is used to generate spiral paths that are continuous and smooth.

In [5], in addition to spiral paths, two other behaviors, namely wall-following and virtual wall path tracking, are considered. A Coarse-to-Fine Constrained Inverse Distance Transform (CFCIDT) is used to link paths together. Particle swarm optimization is utilized for smoothing the path.

The major difference between existing work, such as those mentioned above, and the work in this paper is that existing work consider the coverage path to be constrained within the target area for coverage whereas this paper considers flexibility in allowing slight coverage outside of the boundary for the sake of smoother paths. This flexibility can be an acceptable trade-off to a smoother path for many coverage applications. As shown in Fig. 2, this can lead to a smoother path being generated which in turn may result in various other benefits. However, the challenge is to approximate the target area with an appropriate shape to enable generation of a smoother path; and the main contribution of this paper lies in addressing this challenge. To this end, an analytical method of fitting a squircle to the target area is proposed to enable the creation of a smoother path. Unlike above algorithms, the Squircular-CPP doesn't require large computation time, wall-following, grid representation of the target area, and iterative optimization for smoothing or joining paths.

III. PROBLEM DEFINITION

Let $\mathcal{X} \subset \mathbb{R}^2$ represent a target area that is given to a robot for coverage using a smooth path. In this work, the coverage problem is made flexible in terms of allowing the robot to cover a larger area $\Omega \subset \mathbb{R}^2$, $\mathcal{X} \subseteq \Omega$. However, this flexibility comes with an expectation of generating a path within Ω that is smooth while ensuring that the path is not excessively long.

The above flexibility in allowing the robot to cover outside of the boundary, although not acceptable for some robotic coverage applications, is acceptable for other applications. Example applications include but not limited to unmanned aerial vehicles (UAVs) performing aerial surveying, autonomous underwater vehicles (AUVs) surveying the seabed or an intervention AUV (I-AUV) removing marine growth from surfaces of underwater structures, and floor cleaning robots performing targeted cleaning of dirty areas. For such robots and applications, there is no harm for the robot to slightly and temporarily exit the area for coverage. Thus, the goal is to create an algorithm that exploits this flexibility as a way to generate smoother paths.

Given $\mathcal{X}$ which represents a target area on a surface, the problem is to find an Ω, $\mathcal{X} \subseteq \Omega$, that approximates $\mathcal{X}$ and enables a smooth path to be generated within Ω.

For the area Ω, let P be a generated smooth path that is discretized into points, $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_N$, with infinitesimal distance between any adjacent points along the path. Suppose that the end-effector tool of the robot covers an effective circular area of radius r, $\forall \boldsymbol{p}_i$ $i = 1, 2, \ldots, N$; and let V_i denote the coverage region by the end-effector tool at point $\boldsymbol{p}_i$. Thus, it is necessary to obtain $(\bigcup_i^N V_i) \cap \mathcal{X} = \mathcal{X}$ to prevent missing areas of coverage. In addition, aiming to shorten the length of the smooth path should be taken into account.

IV. THE SQUIRCULAR-CPP ALGORITHM

A flowchart of the overall procedure for generating a smooth coverage path is shown in Fig. 3. The proposed Squircular-CPP forms the second module of the flowchart.

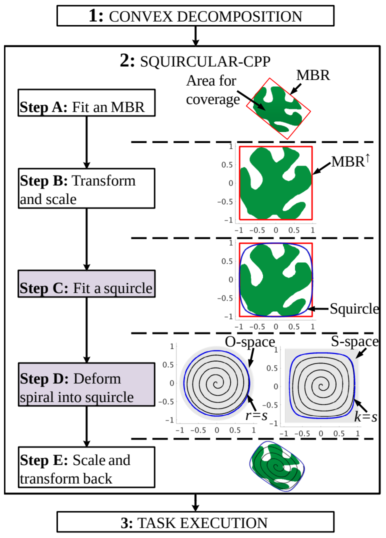

Fig. 3: A simplified flowchart of the overall procedure for generating a smooth coverage path is shown.

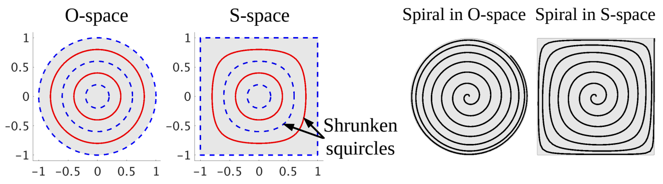

Fig. 4: (Left) FG-squircular mapping. (Right) Mapping of a spiral path.

A. Convex Decomposition

Prior to the implementation of the Squircular-CPP algorithm, convex decomposition of the target area may need to be carried out, as shown in Module 1 of the flowchart (Fig. 3). That is, the target area, which may be non-convex or contain obstacles and holes, will be decomposed into a number of obstacle-free and convex (or near-convex) subareas. There exists many algorithms for convex or near-convex decomposition; e.g. trapezoidal and slice decomposition [14] for polygonal and/or rectilinear environments with obstacles, determining minimum number of near-convex parts in a shape [15], and weak convex decomposition through lines-of-sight [16]. Investigating the limitation of various convex decomposition methods for different environments (e.g. rectilinear, polygonal, with/without obstacles, etc.) is left for future work. As the first paper to present the novel Squircular-CPP algorithm, the focus is on validating Squircular-CPP given convex subareas as inputs. Thus, the rest of the paper focuses on Module 2 of the flowchart in Fig. 3, and especially on the Steps C and D highlighted within Module 2. Note that the sub-area (input to the Squircular-CPP) can be approximately convex (e.g. have convexity ranks [16] above a threshold), or can be formed by combining multiple nearby and small convex subareas together.

B. Fitting an MBR

Let an area $A \subseteq \mathcal{X}$ be the input to Squircular-CPP for coverage and represented as a set of points $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_\eta\}$ (e.g. from point cloud or centers of a uniform grid). For Squircular-CPP it is sufficient for the points in X to represent only the boundary of A. Squircular-CPP starts with fitting a Minimum Bounding Rectangle (MBR) [17] to the points in X. This is shown in Step A of Module 2 in the flowchart (Fig. 3). Henceforth, the shortened notation M2:StepX is used to refer to step X of Module 2 in the flowchart (Fig. 3).

The MBR imposes an upper bound on the size of the squircle (or the scaled squircle). Another benefit of using MBRs is that if the area to be covered is large, then several MBRs can be stacked around each other to cover the area without overlap or gap between them. Additionally, the MBR helps with generating the deformed spiral within the bounding squircle. It defines the centroid of the squircle. Without the MBR, finding the centroid of the squircle is not trivial since the centroid of the bounding squircle is not necessarily the centroid of the target area for coverage. The spiral path may be allowed to exit the MBR to obtain a smoother path, e.g. if there is a constraint on the smoothness of the deformed spiral due to the type of robot.

C. Transforming and Scaling of the MBR and the Points

For convenience, the MBR is transformed such that it is axis-aligned and centered at the origin (M2:StepB).

Let S-space (square) be defined as $\mathrm{S} = \{(x,y) \in \mathbb{R}^2 \big| \ |x| \leq 1, |y| \leq 1\}$ where x and y are coordinates in S. The MBR is scaled along both axes to occupy the entire S-space (M2:StepB). Let the scaled MBR and the scaled points within this MBR be termed as MBR$^{\uparrow}$ and $X^{\uparrow}$, respectively. The relevance of this scaling procedure will become clear in the following subsections.

D. Fitting a Squircle

The scaling of the MBR helps with finding a bounding squircle within the MBR$^{\uparrow}$ (M2:StepC). A squircle is an intermediate shape between the square and the circle, as shown in Fig. 5. A squircle (based on Fernandez-Guasti squircle or FG-squircle in short) is defined as [18]:

$$x^2 + y^2 - \frac{s^2}{k^2}x^2y^2 = k^2 \qquad (1)$$

where $s \in [0,1]$ is the squareness parameter, and k, analogous to the radius of a circle, is the length from the centroid to the point on the boundary that intersects the x or y axis (considering axis-aligned squircle, centered at origin). As shown in Fig. 5, s defines how close a squircle is to a square or a circle. However, this closeness is not linear with respect to the increase in the s value (as shown in Fig. 5).

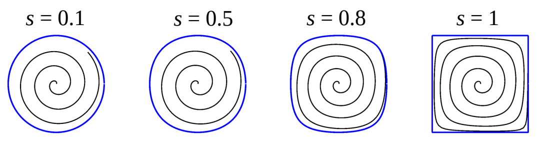

Fig. 5: Squircles at varying values of s (for a constant value of k), and a spiral path within each squircle.

The main contribution of the proposed algorithm is the squircular shape fitting which enables generating a smoother and shorter spiral path than the path generated through the DSCPP algorithm presented previously in [7]. The spiral path in Fig. 6(Left) is generated using the proposed Squircular-CPP algorithm whereas the spiral in Fig. 6(Right) is generated using the DSCPP algorithm [7]. The spiral generated using Squircular-CPP is clearly shorter and smoother for two reasons: (i) the squircle is a more accurate representation of the target area as opposed to an MBR (hence a shorter path is needed); and (ii) as shown in Fig. 5, the closer the bounding squircle is to a circle (i.e. the smaller the s value), then the smoother and shorter is the generated spiral path since the arithmetic spiral would need a lesser deformation to morph into the shape of the squircle. More examples of spiral paths generated through Squircular-CPP were shown in Fig. 1.

Given a set of points $X^{\uparrow}$ within the MBR$^{\uparrow}$, Squircular-CPP can analytically fit an FG-squircle with minimum s value that is axis-aligned with respect to MBR$^{\uparrow}$'s axes and that is upper bounded in size by the MBR$^{\uparrow}$.

Rearranging Eq. (1) to make s the subject gives:

$$s = \sqrt{\frac{(x^2 + y^2 - k^2)k^2}{x^2 y^2}}. \tag{2}$$

For a given value of k, the squareness value s of the squircle can be calculated by substituting the x and y coordinates of a point $\boldsymbol{b}$ on the boundary of the squircle. In Squircular-CPP, $\boldsymbol{b}$ is the farthest point, $\boldsymbol{x}^f \in X^{\uparrow}$, from the centroid.

The goal is to find the axis-aligned squircle with minimum s value such that the farthest point, $\boldsymbol{x}^f$, from the centroid is still within the squircle (i.e. the squircle is large enough to cover the farthest point, but not larger). The point $\boldsymbol{x}^f = \boldsymbol{x}^{\uparrow}_{l^*} \in X^{\uparrow}$ where l^* is simply

$$l^* = \underset{l \in \{1,2,\ldots,\eta\}}{\arg\max} \; \|\boldsymbol{x}^{\uparrow}_l - \boldsymbol{c}\| \tag{3}$$

where $\boldsymbol{c}$ is the centroid of the S-space which is at the origin.

By definition, the MBR$^{\uparrow}$ would have a side length of 2 since it occupies the entire S-space. The bounding squircle (henceforth referred to as BS$^{\uparrow}$), that fits the points $X^{\uparrow}$ in MBR$^{\uparrow}$ shares boundary with the MBR$^{\uparrow}$ and would always have a k value of 1 (half of MBR$^{\uparrow}$ side length). Thus, given $k = 1$, the s value of BS$^{\uparrow}$ can be calculated from Eq. (2) by substituting the x and y coordinates of $\boldsymbol{x}^f$. Once BS$^{\uparrow}$ is defined, a spiral path is generated within this BS$^{\uparrow}$ and later scaled and transformed to the original size and pose of the MBR (M2:StepE). If instead of complete coverage, an above-threshold coverage (user-defined) is acceptable for an application, then the path can be made smoother by incrementally reducing the s value of the bounding squircle while coverage is still above the user-defined threshold. Note that reducing the s value does not reduce the size of the bounding squircle significantly (since k is constant); however, it can improve path smoothness considerably.

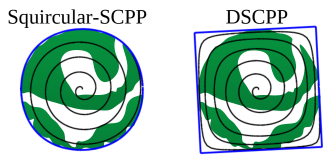

Fig. 6: Comparing Squircular-CPP to DSCPP [7]

E. Deforming a Spiral Path into a Squircle

1) Determining the Appropriate Circle Size in O-Space: The S-space (square) was previously defined as $\mathrm{S} = \{(x,y) \in \mathbb{R}^2 | \; |x| \leq 1, |y| \leq 1\}$ where x and y are coordinates in S. Similarly, let the O-space (circle) be defined as $\mathrm{O} = \{(u,v) \in \mathbb{R}^2 | u^2 + v^2 \leq 1\}$ where u and v are coordinates in O, as shown in Fig. 4(Left).

A squircular mapping technique, named Fernandez-Guasti squircular mapping (FG-squircular mapping) [18], is used to map a non-uniform arithmetic spiral created within the O-space to a deformed spiral within the S-space, as shown in Fig. 4(Right). Using the FG-squircular mapping, given u and v coordinates of a point, $\boldsymbol{p}$, in O-space, the corresponding x and y coordinates of $\boldsymbol{p}$ in S-space can be calculated [18]:

$$\begin{aligned} \boldsymbol{p}_x = \mathrm{O2S}(u,v) &= \frac{\mathrm{sgn}(u,v)}{v\sqrt{2}} \sqrt{u^2 + v^2 - \sqrt{(u^2+v^2)(u^2+v^2-4u^2v^2)}}, \\ \boldsymbol{p}_y = \mathrm{O2S}(u,v) &= \frac{\mathrm{sgn}(u,v)}{u\sqrt{2}} \sqrt{u^2 + v^2 - \sqrt{(u^2+v^2)(u^2+v^2-4u^2v^2)}}, \end{aligned} \tag{4}$$

and conversely, for mapping from S-space to O-space:

$$\begin{aligned} \boldsymbol{p}_u &= \mathrm{S2O}(x,y) = x\sqrt{x^2 + y^2 - x^2y^2} / \sqrt{x^2+y^2}, \\ \boldsymbol{p}_v &= \mathrm{S2O}(x,y) = y\sqrt{x^2 + y^2 - x^2y^2} / \sqrt{x^2+y^2}. \end{aligned} \tag{5}$$

Unlike DSCPP algorithm [7], the goal here is to find the region of the O-space within which the arithmetic spiral will be generated such that when mapped into the S-space it can be made to occupy BS$^{\uparrow}$ (M2:StepD). This goal is achieved using the shrunken FG-squircle (Eq. (6)).

Let $k = s$ in Eq. (1), then the equation reduces to:

$$x^2 + y^2 - x^2y^2 = s^2 \tag{6}$$

which is referred to as the shrunken FG-squircle in [18]. Using this equation, both the size and the squareness of the squircle is controlled using the value of s. For s = 0.2 to 1 in steps of 0.2, the shrunken squircles shown in the S-space (Fig. 4(Left)) are generated. Hence, the larger the s value, the bigger the squircle and the closer it is to a square. Using the FG-squircular mapping, this squircles map to the circles generated in the O-space (Fig. 4(Left)), and vice versa.

The s values of these shrunken FG-squircles (in S-space) are in fact the radii of the circles (in O-space) [18]. Thus, a circle in the O-space with a radius of r would map to a shrunken FG-squircle (in S-space) with $s = r = k$. Thus, the spiral path is first generated within the circle of radius $r = s$ in the O-space where s equals to the s value of the BS$^{\uparrow}$. The spiral path is then mapped to the shrunken FG-squircle in the S-space. Since $k = s$ for shrunken FG-squircle, then the spiral needs to be scaled up by a factor of s to occupy the BS$^{\uparrow}$ (M2:StepE).

2) Generating the Deformed Spiral Path: The rest of the procedure for generating a spiral path is similar to DSCPP [7]. One main modification is that, instead of deforming the spiral path to occupy the entire S-space, the spiral path is deformed to fit within the BS$^{\uparrow}$. Since the BS$^{\uparrow}$ will be scaled back to fit within the original MBR, the final spiral path may no longer be within a bounding squircle but rather within an intermediate shape between the ellipse and the rectangle. This process of scaling back needs a special treatment for keeping the gaps between consecutive laps of the spiral to be less than or equal to the maximum allowed gap while ensuring that the path is not excessively longer than needed.

Let the maximum gap between consecutive laps of the spiral path be denoted as g^{max}. As explained in Section IV-B, the MBR (M2:StepB) is scaled to occupy the entire S-space (M2:StepB). The resulting MBR was referred to as MBR$^{\uparrow}$. This allows a bounding squircle, termed BS$^{\uparrow}$ above, to fit the points in MBR$^{\uparrow}$ (M2:StepC). The gap g^{max} is scaled similarly to become $g^{\uparrow}$. Finally, BS$^{\uparrow}$ is scaled to become the shrunken squircle (M2:StepD); thus, the maximum gap is scaled similarly to become $g = s\ g^{\uparrow}$ where s is the squareness value of BS$^{\uparrow}$. Therefore, the spiral path is first generated within the circle of radius $r = s$ in the O-space and then mapped to the shrunken squircle in the S-space (M2:StepD) such that the gap between any two consecutive laps does not exceed g^{max} and that the path length is not excessively long when the spiral is scaled back to fit the original MBR (M2:StepE). For details, readers are advised to refer to [7].

F. Task Execution

After generating the smooth coverage path using Squircular-CPP, the task is executed (Module 3) through the control of the robot to follow the coverage path. There are numerous control strategies for various robots and applications; details are beyond the scope of this paper.

V. CASE STUDIES

Four case studies are presented to demonstrate and compare the performance of Squircular-CPP algorithm. The work in [4] has already proved, through analytical and experimental studies, that "spirals become the most energy-efficient because the robot can continuously move without stopping and turning" as compared to square spiral paths and boustrophedon paths (lawn-mover paths). Hence, the work in this paper is complimentary with this respect and has been implemented in a real system for marine growth removal, as shown in Fig. 7. For this application, smooth spiral paths have the additional benefit of reducing the prolonged local exposure of the water jet on the pylon's surface as a result of sharp turns which can cause structural damage.

The smoothness of a spiral path generated through Squircular-CPP depends on two factors: (i) the squareness value s, and (ii) MBR's aspect ratio (ratio of the width to the height of the MBR). A visual demonstration of spiral paths for increasing values of s and aspect ratio is shown in Fig. 8. Hence, in the following case studies, the performance of Squircular-CPP is analyzed with respect to these two parameters. In doing so, both the path length as well as smoothness are considered for the analysis. Furthermore, Squircular-CPP is compared to two algorithms: (i) the DSCPP algorithm [7] to assess the extent of improvements on the smoothness, and (ii) the boustrophedon path since it is optimal in length considering that the path is generated for an MBR.

Similar to the work in [9], the curvature is used for quantifying smoothness. Suppose that the path is partitioned evenly into n^s small segments with a length l^s (0.01 m in this paper). The curvature of each segment is derived as [9]:

$$\kappa = \frac{|\Delta\vartheta_j|}{l^s} = \frac{|\vartheta_{j+1} - \vartheta_j|}{l^s} \qquad (7)$$

where ϑ_{j+1} and ϑ_j are the start and end orientation of the jth segment along the path. A smaller value of κ indicates a smaller turn (a smoother path segment), and vice versa.

The work in [9] considers the average curvature of the path. However, a path, such as a boustrophedon path, can have many sharp turns and yet provide a relatively small average curvature due to the straight line segments of the path. Thus, in this work, the sum of undesirable turns above a curvature threshold is considered. Let κ' be the curvature threshold for a path segment. Then, from Eq. (7), the maximum acceptable turn per path segment is $\vartheta^{max} = \kappa'\ l^s$. If $\kappa \leq \kappa'$ for a path segment, then the robot can smoothly execute the path segment (i.e. benefit from the smoothness of the path segment) otherwise the path segment is considered as a sharp turn. The sum of excessive undesirable turns above ϑ^{max} indicates the extent of decrease in the smoothness of

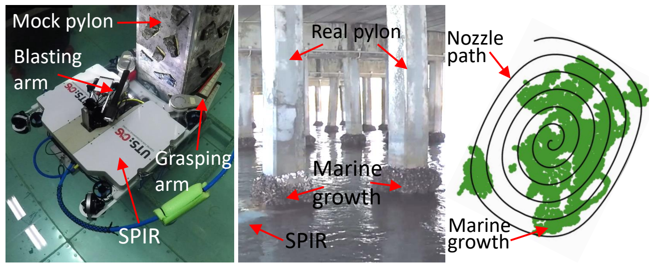

Fig. 7: (Left) Testing using a submersible pylon inspection robot (SPIR) [8] developed in the Center for Autonomous Systems at UTS. (Middle) SPIR in real-world environment. (Right) An example path using Squircular-CPP where the map generation is discussed in [7].

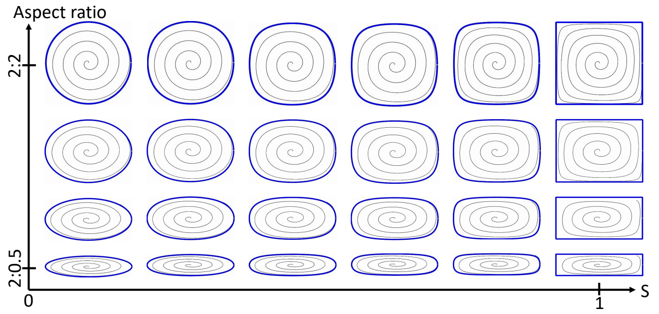

Fig. 8: Spiral paths generated through Squircular-CPP for increasing values of the squareness value s and MBR aspect ratios.

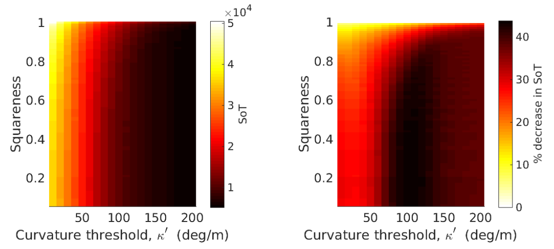

Fig. 9: (Left) SoT value for paths generated through Squircular-CPP for various κ' and squareness values. (Right) Percentage decrease in SoT relative to DSCPP [7].

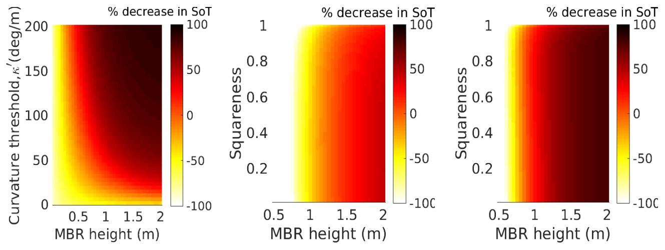

Fig. 11: (Left) Percentage decrease in SoT relative to the boustrophedon path when varying MBR aspect ratios (width fixed at 2 m) and κ'. (Middle) Percentage decrease in SoT for $\kappa' = 40$ deg/m. (Right) Percentage decrease in SoT for $\kappa' = 150$ deg/m.

the path. This sum is termed as Sum over Threshold (SoT):

$$\mathrm{SoT} = \sum_{j=1}^{n^s-1} \left(\begin{cases} 0, & \text{if } |\vartheta_{j+1} - \vartheta_j| \leq \vartheta^{max} \\ |\vartheta_{j+1} - \vartheta_j| - \vartheta^{max}, & \text{otherwise} \end{cases} \right). \tag{8}$$

where n^s is the total number of path segments. A smaller value of SoT indicates a smoother path.

A. Case Study 1: Comparison with DSCPP

This case study is designed to assess the extent of improvements obtained through the proposed Squircular-CPP algorithm relative to the DSCPP algorithm [7]. A 2 m by 2 m area is considered and paths are generated within the area while setting the maximum gap between laps to $g^{max} = 0.01$ m. Results are shown in Fig. 9 where large number of paths are generated with different squareness values and curvature threshold, κ'.

Figure 9(Left) show that as the squareness value decreases, so does the SoT, which improves the smoothness of the path. It also shows that setting a higher curvature threshold, κ', leads to a smoother path (lower SoT value).

Figure 9(Right) shows the percentage decrease in SoT value relative to the DSCPP algorithm. The Squircular-CPP achieves up to 45% improvement (reduction in SoT). The main reason for this improvements is that Squircular-CPP fits a better shape with smoother boundary, and therefore the deformed spiral path can be made smoother as it needs to gradually deform to a smoother boundary shape rather than an MBR. Better shape fitting can also reduce the path length, as shown in later case studies.

B. Case Study 2: Comparison with Boustrophedon Path

The improvements in smoothness through Squircular-CPP is shown relative to the boustrophedon path for the same setup as in Case Study 1. The percentage decrease in SoT

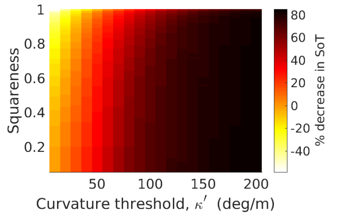

Fig. 10: Percentage decrease in SoT relative to the boustrophedon path for various κ' and squareness values.

value relative to the boustrophedon path is shown in Fig. 10. For κ' values of greater than approx. 35 deg/m, the Squircular-CPP algorithm performs significantly better in terms of smoothness, particularly for lower values of s. This is because as κ' is increased, more segments of the path generated through Squircular-CPP fall below the threshold whereas the sharp 90 degree turns of the boustrophedon path remain well above the threshold. Once again, lower squareness value improves the smoothness of the path.

C. Case Study 3: Effect of MBR's Aspect Ratio

The ratio of the width to the height of the MBR (aspect ratio) does affect the performance of the Squircular-CPP algorithm. Figure 11(Left) shows what happens to the percentage decrease in SoT value relative to the boustrophedon path when varying the aspect ratio (width is fixed at 2 m while varying the height from 0 to 2 m). For aspect ratios of 2:1 to 2:2, Squircular-CPP outperforms boustrophedon path, particularly for κ' values of greater than around 20 deg/m.

To construct the heatmap in Fig. 11(Left), for each pair of κ' and aspect ratio values, the average of the SoT values from all squareness values (0.01 to 1 in steps of 0.01) is considered. Then, the percentage decrease in average SoT (relative to the boustrophedon-based path) is calculated. To see the effect of aspect ratio for a particular κ', two example heatmaps are generated where in the first one (Fig. 11(Middle)) $\kappa' = 40$ deg/m and in the second (Fig. 11(Right)) $\kappa' = 150$ deg/m. In general, a higher κ' value and a smaller squareness value implies a smoother path.

D. Case Study 4: Comparisons with Respect to Path Length

The main objective in this paper is not to minimize path length but rather to achieve smooth paths. However, analyzing path length can provide more insight into the performance of Squircular-CPP. Same as previous case studies, maximum gap between laps is $g^{max} = 0.01$ m.

First, comparisons are made relative to the DSCPP and the boustrophedon algorithms. Note that in both DSCPP and boustrophedon algorithms, squircular shape fitting is not used; hence, the entire MBR is covered. Results are shown in Fig. 12(Left). Reducing the squareness value can improve the path length by up to 45% when compared to a path generated through the DSCPP algorithm. For squareness values of 0.85 and lower, Squircular-CPP outperforms boustrophedon path in terms of length and up to 20 % improvement is achieved.

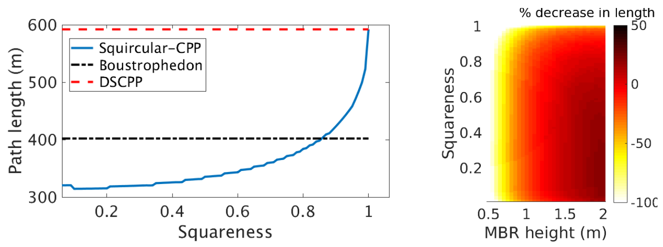

Fig. 12: (Left) Comparison of Squircular-CPP relative to the DSCPP and the boustrophedon algorithms in terms of path length. (Right) Percentage decrease in length relative to boustrophedon path when varying MBR aspect ratio (MBR width is fixed at 2 m).

Comparisons are also made relative to the boustrophedon path by varying the aspect ratio of the MBR as shown in Fig. 12(Right). Increasing the aspect ratio improves the path length in Squircular-CPP, particularly for lower values of squareness. However, for thin MBRs (aspect ratio of around 2:1 to 2:0), the boustrophedon path is better in terms of length, particularly for squareness values of close to 1. Nonetheless, the work in [4] shows that "if a shorter route contains several sharp turns, the robot may consume more energy due to frequent decelerations, changes of directions, and accelerations. A longer route may require less energy if the robot does not have to accelerate often". Note that the boustrophedon path can still benefit from using squircular shape fitting proposed in this paper to generate a shorter path within the squircle rather than the MBR.

VI. Conclusion

The Squircular-CPP algorithm is developed with the aim of achieving smooth coverage paths. The novelty of the algorithm lies within the squircular shape fitting of the target area which is not only simple, fast and analytical, but also enables a smooth spiral path to be appropriately deformed within the fitted shape. The Squircular-CPP algorithm starts with fitting a minimum bounding rectangle (MBR) to the target area. However, instead of deforming the arithmetic spiral path to fit within the MBR, the aim is to fit it within a shape that is closer to a circle (or ellipse) so as to obtain a smoother spiral path. After appropriately scaling the MBR, the best fit squircle, which is an intermediate shape between the square and the circle, is determined analytically. The best fit squircle within an MBR is considered as the squircle that is closest to a circle while encompassing the target area. A simple arithmetic spiral path is generated within a circle and then deformed to fit within the squircle using Fernandez-Guasti squircular mapping. Four case studies are presented to validate the algorithm and to show its properties in terms of smoothness and length.

Future work includes extending the algorithm to be applicable to multi-robot coverage and surfaces with complex curvatures (i.e. three-dimensional coverage), performing analysis based on comprehensive real-world experiments, and incorporating task planning where areas to cover are constrained by complex tasks including priorities [19].

VII. Acknowledgements

This work is supported in part by the Australian Research Council (ARC) Linkage Project (LP150100935), the Roads and Maritime Services of NSW, and the Centre for Autonomous Systems at the University of Technology, Sydney. Authors thank Dr. Andrew To and Dr. Khoa Le for their help with implementation work in the SPIR system.

References

[1] M. Hassan and D. Liu, "Simultaneous area partitioning and allocation for complete coverage by multiple autonomous industrial robots," *Autonomous Robots*, vol. 41, no. 8, pp. 1609–1628, Dec 2017.

[2] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias, and G. Cai, "A survey on inspecting structures using robotic systems," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, pp. 1 – 18, Dec 2016.

[3] A. Khan, I. Noreen, and Z. Habib, "On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges." *Journal of Information Science & Engineering*, vol. 33, no. 1, pp. 101–121, Jan 2017.

[4] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee, "Energy-efficient motion planning for mobile robots," in *IEEE International Conference on Robotics and Automation*, vol. 5, April 2004, pp. 4344–4349.

[5] T.-K. Lee, S.-H. Baek, Y.-H. Choi, and S.-Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 801 – 812, 2011.

[6] Y. H. Choi, T. K. Lee, S. H. Baek, and S. Y. Oh, "Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform," in *International Conference on Intelligent Robots and Systems*, 2009, pp. 5788–5793.

[7] M. Hassan and D. Liu, "A deformable spiral based algorithm to smooth coverage path planning for marine growth removal," in *International Conference on Intelligent Robots and Systems*, 2018, pp. 1913–1918.

[8] J. Woolfrey, D. Liu, and M. Carmichael, "Kinematic control of an autonomous underwater vehicle-manipulator system (AUVMS) using autoregressive prediction of vehicle motion and model predictive control," in *International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4591–4596.

[9] T. K. Lee, S. H. Baek, S. Y. Oh, and Y. H. Choi, "Complete coverage algorithm based on linked smooth spiral paths for mobile robots," in *International Conference on Control Automation Robotics Vision*, Dec 2010, pp. 609–614.

[10] F. Balampanis, I. Maza, and A. Ollero, "Spiral-like coverage path planning for multiple heterogeneous UAS operating in coastal regions," in *International Conference on Unmanned Aircraft Systems*, June 2017, pp. 617–624.

[11] J. Backman, P. Piirainen, and T. Oksanen, "Smooth turning path generation for agricultural vehicles in headlands," *Biosystems Engineering*, vol. 139, pp. 76 – 86, 2015.

[12] E. Gonzalez, P. Aristizabal, and M. Alarcon, "Backtracking spiral algorithm: a mobile robot region filling strategy," in *International Symposium on Robotics and Automation*, 2002, pp. 261–266.

[13] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra, and C. Bustacara, "BSA: A complete coverage algorithm," in *IEEE International Conference on Robotics and Automation*, April 2005, pp. 2040–2044.

[14] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258 – 1276, 2013.

[15] Z. Ren, J. Yuan, and W. Liu, "Minimum near-convex shape decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2546–2552, Oct 2013.

[16] S. Asafi, A. Goren, and D. Cohen-Or, "Weak convex decomposition by lines-of-sight," *Computer Graphics Forum*, vol. 32, no. 5, pp. 23–31, 2013.

[17] D. Chaudhuri and A. Samal, "A simple method for fitting of bounding rectangle to closed regions," *Pattern Recognition*, vol. 40, no. 7, pp. 1981 – 1989, 2007.

[18] C. Fong, "Analytical methods for squaring the disc," *arXiv preprint arXiv:1509.06344*, 2015.

[19] C. Yoo, R. Fitch, and S. Sukkarieh, "Online task planning and control for fuel-constrained aerial robots in wind fields," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 438–453, 2016.