Robot Coverage Path Planning for General Surfaces Using Quadratic Differentials

Yu-Yao Lin¹, Chien-Chun Ni², Na Lei¹, Xianfeng David Gu¹ and Jie Gao¹

Abstract—Robot Coverage Path planning (i.e., the process of providing full coverage of a given domain by one or multiple robots) is a classical problem in the field of robotics and motion planning. The goal of such planning is to provide nearly full coverage while also minimize duplicated visited area. In this paper, we focus on the scenario of path planning on general surface, including planar domains with complex topology, complex terrain, and general surface in 3D space. Our approach described in this paper adopts a natural, intrinsic and global parametrization of the surface for robot path planning, namely the holomorphic quadratic differentials. We give each point on the surface a uv-coordinates naturally represented by a complex number, except for a small number of zero points (singularities). We show that natural, efficient robot paths can be obtained by using such coordinate systems. The method is based on intrinsic geometry and thus can be adapted to general surface exploration in 3D.

I. INTRODUCTION

The Coverage Path Planning (CPP) problem is to determine a path that passes through all points in a given geometric domain. It is a classical problem in robotics and motion planning and is of fundamental value to many applications that require a robot or multiple robots to sweep over the target area, such as vacuum cleaning robots, lawn mowers, underwater imaging/scanning robots, window cleaners, and many others.

In general, the CPP problem has multiple goals: full coverage (i.e., every point in the domain Ω is covered), no overlapping or repetition (no point is visited multiple times), and/or a variety of objectives on the simplicity or quality of the paths. Satisfying all such requirements is difficult if not impossible. Therefore priorities are often set on these possibly conflicting objectives and the goal is to obtain a good tradeoff.

The geometric shape of the domain to be covered is crucial in the design of coverage path planning algorithms. Simple shapes such as convex polygons can be covered by simple zigzag motion patterns (lawn mower patterns). Therefore, most algorithms for coverage path planning first decompose the target region into ‘simple cells’. The cell decomposition can be represented by a cell adjacency graph in which each cell is a vertex and two vertices are connected if they share common boundaries. Within each cell we can use a simple zig-zag pattern and to cover the entire domain we need to visit each cell at least once.

In all the decomposition methods, there are two general issues that may affect the final performance. First we need to find a path on the cell adjacency graph that visits each cell at least once – ideally exactly once (to keep the path short). Finding a path that visits each vertex of a graph exactly once is the well known Hamiltonian path problem, which is NP-hard [1]. The adjacency graph may not admit a Hamiltonian path – thus a robot may have to repeatedly visit some points just to get from one cell to the next cell. Second, all the algorithms above use the extrinsic coordinate system, i.e., the Euclidean coordinates representing the domain of interest. Such extrinsic coordinate systems, albeit being natural choices, are not the best to encode the complex geometric and topological features introduced by obstacles and boundaries. This is in fact the core challenge that the cell decomposition is mean to tackle. When the domain is not flat (e.g., on a terrain or as a general surface in 3D), the extrinsic coordinate system and the cell decomposition may lead to unnecessarily many pieces depending on the detailed implementation.

Our Contribution.

In this paper we focus on solving this problem using a generic solution that is applicable to general surfaces in 3D. The novelty of our method is to abandon the extrinsic Euclidean coordinates system and adopt the intrinsic coordinate system, i.e., a global parametrization of the domain of interest. To get an idea, consider a standard torus, one can slice the torus open along the two generators of the fundamental group of the torus and the torus can be flattened as a square. Thus, one can represent the points on the torus by a uv coordinate system, where the u coordinate represents the position of the point p along one generator of the fundamental group and v represents the position along the other generator. Both the geometry of the surface and the topology of the surface are inherently encoded in this new coordinate system. Finding a coverage path for the torus under the uv coordinate system is now trivial – one can simply zig-zag in the uv coordinate system which becomes a spiral motion on the torus.

We introduce the theory and algorithms for computing the intrinsic coordinate system using holomorphic quadratic differentials. Depending on the topology of the surface there are a constant number of zero points (also called critical points, singular points or singularities) which do not have such coordinates. But such singular points are

The authors would like to acknowledge support through NSF DMS-1418255, CCF-1535900, CNS-1618391 and AFSOR FA9550-14-1-0193.

¹Department of Computer Science, Stony Brook University, Stony Brook, NY, USA {yyulin, chni, jgao, gu}@cs.stonybrook.edu
²School of Software, Dalian University of Technology, Liaoning, China nalei@dlut.edu.cn
of zero measure. The coordinate system is naturally represented by a complex number. One can trace out a curve by fixing the real/imaginary part of the coordinate, called the vertical/horizontal trajectory respectively. This coordinate system naturally produces a space decomposition by slicing along critical trajectories (i.e., trajectories that end at zero points). Each component is a simply connected piece with the complex coordinates as its natural parametrization. This decomposition can also be represented by a graph $G$ in which the vertices are the critical points and an edge represents a cell that touches two singular vertices. This graph and the coordinate system/parametrization are used to generate a coverage path. See Figure 1 for an example.

![Diagram](image)

**Fig. 1.** Example of a three holes donut with trapezoid decomposition (a) and our holomorphic quadratic differentials method (b). In trapezoid decomposition, the donut is decomposed into 11 cells, the CPP problem here is equivalent to finding a Hamiltonian path with these cells as vertices, which is NP-hard. Instead, our method simply cuts the donut into 6 cells, and the CPP problem in our setting is equivalent to finding an Euler cycle with these cells as edges, which can be easily achieved in polynomial complexity.

To generate a coverage path, we need to decide what order we use to visit the decomposed cells. Again we encounter the problem of visiting each cell at least once. In our setting we actually need to visit all the edges (representing the cells) in $G$, ideally once and only once. So this is in fact the Euler cycle problem, which is, fortunately, much easier than the Hamiltonian cycle problem. Any graph in which all vertices have even degree has an Euler cycle. In our case, the degree of a critical point may not be even. But we can simply have even degree has an Euler cycle. In our case, the degree of the critical points due to the special local structure. Fortunately our robot cover path avoids the zero points and all we need is to trace out the 3 critical trajectories through each critical point, which is carefully handled in our algorithm.

We evaluated the coverage path generated by our algorithms on a variety of different settings including flat domains with obstacles, non-flat terrains, as well as general high genus surfaces. Our method is an offline method and requires the domain to be known in advance.

### II. Theory on Holomorphic Quadratic Differentials

Our solution for the CPP problem is based on a global surface parameterization, namely the holomorphic quadratic differentials. Holomorphic quadratic differentials possess a good property that they inherently induce non-intersecting trajectories on a surface. This benefit prompts us to develop a path planning algorithm based on the trajectories. In this section, we briefly introduce some basics of holomorphic quadratic differentials. Then we design our path planning method on general surfaces. For detailed treatments, we refer readers to [3] for Riemann surface theory, [4] for complex analysis, and [5] for holomorphic quadratic differentials.

#### A. Riemann Surfaces

**Definition 1:** (Manifold). Let $M$ be a topological space. For each point $p \in M$, there is a neighborhood $U_\alpha$ and a continuous bijective map $\phi_\alpha : U_\alpha \rightarrow \mathbb{C}$ from $U_\alpha$ to an open set $\mathbb{C} \subset \mathbb{C}$, $(U_\alpha, \phi_\alpha)$ is called a local chart. If two neighborhoods $U_\alpha$ and $U_\beta$ intersect, then the transition map between the chart

$$
\phi_{\alpha \beta} = \phi_\beta \phi_\alpha^{-1} : \phi_\alpha(U_\alpha \cap U_\beta) \rightarrow \phi_\beta(U_\beta \cap U_\alpha)
$$

is a continuous bijective map. $M$ is an $n$ dimensional manifold, the set of all local charts \{$(U_\alpha, \phi_\alpha)$\} form an atlas.

**Definition 2:** (Holomorphic Function). A complex function $f : \mathbb{C} \rightarrow \mathbb{C} : x + iy \mapsto u(x, y) + iv(x, y)$ is holomorphic, if it satisfies the following Cauchy-Riemann equation $u_x = v_y, u_y = -v_x$. If $f$ is invertible and $f^{-1}$ is also holomorphic, then $f$ is called a bi-holomorphic function.

**Definition 3:** (Riemann Surface). A Riemann surface is a surface with an atlas \{$(U_\alpha, \phi_\alpha)$\}, such that all chart transitions $\phi_{\alpha \beta}$ are bi-holomorphic. The atlas is called a conformal atlas and the local coordinates $\phi_\alpha(U_\alpha)$ are called holomorphic coordinates. The maximal conformal atlas is called a conformal structure of the surface.

On a Riemann surface, we can define a differential based on the conformal structure. Intuitively, a differential can be regarded as a vector field on a surface. The integration on a differential gives a surface parameterization. The holomorphic differentials and quadratic differentials we introduce below are curl and divergence free vector fields.
**Definition 4:** (Holomorphic Differential). Given a Riemann surface \( R \) with a conformal atlas \( \{ (U_\alpha, \phi_\alpha) \} \), a holomorphic differential \( \zeta \) is a complex differential form defined by a family \( \{ (U_\alpha, z_\alpha, \zeta_\alpha) \} \), such that \( \zeta_\alpha = \phi_\alpha(z_\alpha)dz_\alpha \), where \( \phi_\alpha \) is a holomorphic function on \( U_\alpha \), and if \( z_\alpha = \phi_\alpha^\beta(z_\beta) \) is the coordinate transformation on \( U_\alpha \cap U_\beta \), then \( \phi_\alpha(z_\alpha)\frac{dz_\alpha}{dz_\beta} = \phi_\beta(z_\beta) \).

According to the Poincaré-Hopf theorem [6], any vector field on a surface with non-zero Euler number must have the singularities where the vector field vanishes. Such singularities are called zero points. Here we define the zero points of a holomorphic differential.

**Definition 5:** (Zero Point). For a point \( p \) on a surface \( R \), if the local representation of a holomorphic differential \( \zeta \) around \( p \) is \( \zeta_\alpha = \phi_\alpha(z_\alpha)dz_\alpha \) and \( \phi_\alpha = 0 \) at \( p \), then \( p \) is called a zero point of \( \zeta \).

**B. Holomorphic Quadratic Differentials**

**Definition 6:** (Holomorphic Quadratic Differential). Given a Riemann surface \( R \). Let \( \Phi \) be a complex differential form with a conformal atlas \( \{ (U_\alpha, \phi_\alpha) \} \), such that on each local chart with the local parameter \( z_\alpha \), the differential can be represented by \( \phi_\alpha = \phi_\alpha(z_\alpha)dz_\alpha^2 \), where \( \phi_\alpha(z_\alpha) \) is a holomorphic function.

1) **Zero Points and Trajectories:** For a holomorphic quadratic differential \( \Phi \) on a surface \( R \), any point \( p \in R \) away from zero has the local coordinate defined as

\[
\xi(p) := \int_p^z \sqrt{\phi_\alpha(z_\alpha)}dz_\alpha. \tag{1}
\]

This is called the natural coordinate induced by \( \Phi \). The curves with constant real natural coordinates are called the vertical trajectories; while the curves with constant imaginary natural coordinates are called the horizontal trajectories. A trajectory which ends in zero points is called a critical trajectory, otherwise it is a regular trajectory. The horizontal trajectories of \( \Phi \) are either infinite spirals or finite closed loops. This means that the trajectories of holomorphic quadratic differentials are non-intersecting trajectories on a surface. This property is the key idea of our path planning algorithm.

**Definition 7:** (Genus). A genus \( g \) of a surface is the largest number of cuttings along non-intersecting simple closed curves on the surface without disconnecting it.

The local structure around a zero point of a holomorphic quadratic differential is a complex function \( z \rightarrow z^2 \). For any holomorphic quadratic differential \( \Phi \) on a closed surface with genus \( g \geq 1 \), there are \( 4g - 4 \) zero points. For a multiply-connected surface with \( n > 2 \) boundaries, there are \( 2g - 2 \) zero points of \( \Phi \). Zero points are also called critical points because they are the endpoints of critical trajectories.

**C. Surface Decomposition**

The path planning technique proposed in this work is applicable to both multiply-connected surfaces (surface with boundaries or obstacles) and general closed surfaces. For multiply-connected surfaces, we can directly decompose the surfaces along their critical trajectories. For general closed surfaces, the holomorphic quadratic differentials whose horizontal trajectories are closed loops induce the surface decomposition. The rationale of these properties are described as follows.

**Strebel Differentials.** For a closed surface with genus \( g > 1 \), holomorphic quadratic differentials induce the decomposition for the surface under some conditions. Those holomorphic quadratic differentials are called Strebel differentials.

**Definition 8:** (Strebel Differential [5, 7]). Suppose \( \Phi_s \) is a holomorphic quadratic differential on a surface \( R \) with genus \( g > 1 \). \( \Phi_s \) is called a Strebel differential, if all of its regular horizontal trajectories are closed loops.

Notice that for a Strebel differential \( \Phi_s \) on a closed surface \( R \) with genus \( g > 1 \), all the regular horizontal trajectories are closed loops as shown in Figure 2. The set of critical trajectories together with the critical points form the critical graph \( \Gamma \) of Strebel differential \( \Phi_s \). The critical graph \( \Gamma \) decomposes the surface \( R \) into \( 3g - 3 \) topological cylinders [5].

**Symmetric Quadratic Differentials.** For any given multiply-connected surface \( M \) with \( n > 2 \) boundaries, we can find a holomorphic quadratic differential which decomposes \( M \) into \( 3n - 3 \) simply-connected surfaces \( \{ d_1, d_2, \ldots, d_{3n-3} \} \).

According to the symmetric image property [5], \( M \) and its double \( M \) form a symmetric surface \( \tilde{M} = \{ M \cup M \} \) on which their corresponding boundaries are identified. Any holomorphic quadratic differential \( \Phi \) on \( \tilde{M} \) is reflected to \( M \). As a result, A symmetric surface \( \tilde{M} \) is with a symmetric holomorphic quadratic differential \( \Phi \). Because the boundaries \( \partial M \) and \( \partial \tilde{M} \) are identified, each horizontal (vertical) trajectory \( \gamma \) of \( M \) and its symmetric trajectory \( \tilde{\gamma} \) of \( \tilde{M} \) are connected and form a closed loop.

The symmetric surface \( \tilde{M} \) is, therefore, a closed surface with genus \( g = n \). The holomorphic quadratic differential \( \Phi \) on \( \tilde{M} \) is a Strebel differential, which means the critical graph decomposes \( \tilde{M} \) into \( 3n - 3 \) topological cylinders.

Each cylinder \( c_i \) is symmetric along the two curves which are some intervals of \( \partial \tilde{M} \). That is to say, \( c_i \) consists of two symmetric simply-connected domain \( d_i \) and \( \tilde{d}_i \). By considering \( \{ d_1, d_2, \ldots, d_{3n-3} \} \), we can conclude that the holomorphic quadratic differential \( \Phi \) decomposes \( M \) into \( 3n - 3 \) simply-connected surfaces.

**III. ALGORITHM**

The core idea of the proposed algorithm is the holomorphic quadratic differentials, which induce surface parameter-
izations for general surfaces. In brief, holomorphic quadratic differentials inherently induce non-intersecting trajectories on a surface as shown in Figure 3. This property provides us enough freedom on manipulating the trajectories, and motivates us to develop our path planning algorithm.

Holomorphic quadratic differentials can be obtained by multiplying two holomorphic differentials. III-C briefly lists the computational steps of holomorphic differentials. The parameterizations of holomorphic quadratic differentials should satisfy the property of being a curl-free vector field. It is challenging to control the numerical error around critical points due to the special local structure. As for our robot cover path, it avoids the zero points and all we need is to trace out the 3 critical trajectories through each critical point.

For a topological torus (closed surface with genus one) and an annulus, the holomorphic quadratic differentials and holomorphic differentials are equivalent. Therefore, by connecting each path induced by the trajectories of a holomorphic differential, a path planning is obtained. The algorithm described in this section focuses on the closed surfaces with genus $g > 1$, and the multiply-connected surface with $n > 2$ boundaries. For a closed surface with boundaries, we can double cover the surface to become a closed surface with genus $g > 1$. Then the algorithm can be directly applied.

A. Discrete Approximation

The mathematical concepts on smooth surfaces are now transformed to the numerical procedures on triangular meshes. A smooth surface is approximated by a piecewise linear triangle mesh $T$. The half-edge data structure is adopted in our implementation. We denote a vertex by $v_i$, a half-edge by $[v_i, v_j]$, and an oriented triangle face by $[v_i, v_j, v_k]$. A discrete differential is a function defined on the edge $\omega : E \rightarrow \mathbb{C}$. The integration of a discrete differential, $f : V \rightarrow \mathbb{C}$, gives a complex number or a $uv$-coordinate to each vertex.

B. Algorithm Overview

The following pipeline shows a summary of the main procedures of the path planning in this paper. The input is a triangular mesh of a closed surface with genus $g > 1$, or a multiply-connected surface with $n > 2$ boundaries. We first compute the holomorphic differential basis on a surface, which is then used to compute the holomorphic quadratic differentials. The holomorphic quadratic differential induces a global parameterization, and the resulting critical trajectories naturally decompose the surface into $3g - 3$ ($3n - 3$) sub-surfaces. For each sub-surface, we can compute a number of paths by tracing regular trajectories. The paths are concatenated together to become a zig-zag path on the sub-surface. Finally, we combine the sub-surfaces back to get a continuous path on the whole surface.

C. Holomorphic Differentials

The computation of holomorphic differentials is to solve an elliptic partial differential equation on a triangle mesh using finite element method. The key step is to use piecewise linear functions defined on edges to approximate differentials. Furthermore, the differentials minimize the harmonic energy, the existence and the uniqueness are guaranteed by the Hodge theory [8]. Readers can refer to the works [2], [9] for more details.

D. Holomorphic Quadratic Differentials

The holomorphic quadratic differentials on a surface can be obtained from the products of any two holomorphic differentials of the surface, $\Phi = \{\zeta_i \cdot \zeta_j\}, i, j \in \{1,2,\ldots,2g\}$.

Algorithm 1: Coverage path planning

Input: A triangle mesh $T$
Output: A coverage path planning $P$ of $T$
1. Compute a holomorphic differential basis for $T$;
2. Compute a holomorphic quadratic differential $\Phi$ for $T$;
3. Locate zero points of $\Phi$ on $T$;
4. Trace the critical graph $\Gamma$ from zero points;
5. $T$ is decomposed along the critical graph $\Gamma$ and the sub-surfaces $T \setminus \Gamma = \{d_1, d_2, \ldots, d_{3n - 3}\}$ are obtained.
6. For each $\{d_i\}$, generate a path planning $P_i$;
7. The path planning of the whole surface is formed by $P_1 \cup P_2 \cup \cdots \cup P_{3n - 3} \cup \Gamma$.

Fig. 3. A three-hole donut with zero points(p1 ~ p4) and simply-connected surfaces(d1 ~ d6) decomposed by the critical trajectories(in blue).

E. Surface Decomposition

For a closed surface with genus $g > 1$, the surface decomposition is induced by Strebel differentials. Since holomorphic quadratic differentials $\zeta_i \cdot \zeta_j$ form a vector space, and Strebel differentials are the holomorphic quadratic differentials with closed horizontal trajectories. Therefore, a Strebel differential can be computed by the linear combination of holomorphic quadratic differentials. The surface is decomposed to $3g - 3$ topological cylinders with two boundaries $\{c_1, c_2, \ldots, c_{3g - 3}\}$. For any multiply-connected surface with $n > 2$ boundaries, the critical graph of a holomorphic quadratic differential decomposes the surface to $3n - 3$ simply-connected surfaces $\{d_1, d_2, \ldots, d_{3n - 3}\}$.

In order to decompose the given surface along the critical graph of a computed holomorphic quadratic differential, we first locate the zero points on the surface. Then we trace the critical trajectories from the zero points. Figure 3 illustrates the surface decomposition. For a surface with three holes (inner boundaries), there are four critical points (zero points) and six simply-connected domains.
we take a multiply-connected surface $\Phi$. Coverage Path

\[ \epsilon \]

density step $\{ \text{of the decomposed simply-connected surfaces denoted by} \{ \text{each zero point is dual to a node and each sub-surface is visited twice.} \]

sub-surface inspires the idea of finding an Euler cycle of $G_M$. By doubling each edge, it is guaranteed to find an Euler cycle which promises the visiting of every sub-surface.

Here we take the surface shown in Figure 3 as an example. Each zero point $p_i$ is dual to a node, and each decomposed simply-connected surface $d_i$ is dual to an edge. Figure 4(a) shows the doubled dual graph. For each edge $d_i$, the doubled edge $d_i$ is created. Figure 4(b) shows an Euler cycle of the doubled dual graph of Figure 3. In the dual graph, Euler cycle makes the navigation which starts and ends at the same point.

2) Path Interlacement: Euler cycle of the dual graph of a surface implies that every sub-surface is visited twice. By interleaving two zig-zag paths with same density step, each sub-surface can be covered nicely. Figure 3 illustrates the interleaving paths on the simply-connected domain $d_1$ in Figure 3. A robot can cover $d_1$ by traveling along both the blue and the orange paths which connect the zero points. By switching from one path to another at a zero point, a robot can transfer between adjacent sub-surfaces. The coverage path for the whole surface is therefore performed by following the path interlacement and the Euler cycle scheme.

IV. EXPERIMENTAL RESULTS

We evaluate our algorithm on various surfaces, and analyze the influence of different density step on coverage. We first demonstrate our algorithm on a 2D three holes donut as in Figure 5. The coverage path result is displayed in
The CPP problem has been studied extensively and one can refer to nice surveys [10], [11] for past work in this area. For 2D domains, most works use a cell decomposition to decompose the domain into simple shapes. Popular cell decomposition includes classical trapezoid decomposition [12], [13] and boustrophedon cellular decomposition [14]–[16], Morse decomposition [17], [18], slice decomposition [19], various grid-based algorithms [20]–[23], etc. Most of these approaches are concerned of producing a small number of cells in the decomposition, and whether the decomposition can be done in the online setting (when the target domain is unknown and to be discovered). Other methods include applying spanning tree coverage [24], [25] and neural network based coverage [26]–[28].

Coverage path planning for surfaces in 3D is less investigated. Hert et al. [29] considered coverage of a projectively planar 3D volume, they project the domain in 2D and then take advantages of the 2D planar terrain-covering algorithm to solve the problem. Atkar et al. [30] extended the Morse decomposition to non-planar surfaces but did not consider obstacles. In [31] Bhattacharya et al. extended their grid-based algorithm [23] into 3D cased; they first separated the domain into voronoi cells, then handled them by multiple robots. In [32], [33], the authors proposed a lawnmower type of algorithm on 3D planar domain, but the results only show terrains with boundary and without obstacles. More heuristic algorithms are adopted in application scenarios as [32]–[34].

The one most relevant was our earlier work for generating a space filling curve [35]. However, the focus in [35] was to find a curve with progressive density – that is, we want a path such that the distance from any point to the path to be shrinking progressively when the path gets longer. The same as in a followup work [36]. Although quadratic differentials were also used in [36] but both the theory and the algorithms for generating the curves are totally different from here.

The CPP problem is also related to various traveling salesman problem (TSP with neighborhoods [37]), the lawn-
mower problem (full cover of a region by a path with minimum length) [38], and the sweeping path problem (full coverage by a robot arm of fixed geometric degree of freedom) [39]. Since these problems are sufficiently different we skip the results here.

VI. CONCLUSION

In this paper, a brand new surface parameterization, holomorphic quadratic differentials, is adopted to perform the coverage path planning for general surfaces with complex topology. The natural coordinates of holomorphic quadratic differentials inherently induce non-intersecting trajectories on surfaces. This property inspires us to develop a robot coverage path planning algorithm. Moreover, holomorphic quadratic differentials intrinsically bring a regular number of surface decomposition. By converting the surface decomposition to its doubled dual graph, robots can travel on the whole surface according to the Euler cycle with great coverage.

REFERENCES


