Efficiently Computing Shortest Paths on Curved Surfaces with Newton's Method

Ruyuan Liu, Fengyang Xiao and Wenlong Meng

Abstract—**Geodesics are important in the study of metric geometry. Although Euler–Lagrange equations are used to formulate geodesics, closed-form solutions are not available except in a few cases. Therefore, researchers have to seek for numerical methods instead of finding geodesics in computer vision and graphics. In this paper, we first formulate the computation of geodesics on a parametric surface into an optimization-driven problem and then propose an efficient solution to the optimization problem with a second-order Newton iteration method. The comparative study shows that our algorithm is an order of magnitude faster than the existing approaches for the same level of accuracy.**

Index Terms—**geometry processing, parametric surface, geodesic, shortest path, optimization, Newton's method.**

I. Introduction

In the field of Riemannian geometry, geodesics are characterized by having vanishing geodesic curvature [1], [2]. Suppose that there exists a geodesic curve Π on a Riemannian surface S equipped with Levi-Civita connection. Parallel transport of a vector along Π preserves the inner product of the transported vector and the tangent vector of the geodesic, as well as the norms of the transported vector and the tangent vector.

The existence of geodesic paths between any pair of points on a connected Riemannian manifold is guaranteed by the Hopf-Rinow theorem [3]. But it should be noted that the commonly used geodesics refer to minimal paths, i.e., curves globally minimizing the Riemannian length between two points. This local minimum curves are the generalization of straight lines in Euclidean geometry to the Riemannian manifolds. The computation of shortest paths is a fundamental task in computer vision and graphics. For example, many geometry processing tasks including segmentation [4], meshing [5], shape retrieval [6], [7] and geometric deep learning [8], [9] depend heavily on the geodesic distances and shortest paths.

Based on the theory of calculus of variations, the Euler–Lagrange equations of motion are often used to characterize the properties of geodesics. However, the PDE array does not have a closed-form solution on a general surface $S = (x(u,v), y(u,v), z(u,v))$. Therefore, researchers have to find numerical methods to compute discrete geodesic paths. Most of the existing approaches [10], [11], [12] assume that the input is a polygonal surface and aim at finding an as-short-as-possible path lying on the piecewisely-linear surface. Run-time performance and accuracy are often used as a pair of indicators to evaluate a geodesic algorithm.

Manuscript received July 13, 2022; revised December 12, 2022.

R. Liu is an associate professor of College of Information Engineering and Technology, Qingdao Vocational and Technical College of Hotel Management, Qingdao 266100, P.R. China. (e-mail: 1937504001@qq.com)

F. Xiao is a postgraduate student of School of Computer Science and Technology, Shandong University, Qingdao 266237, P.R. China. (e-mail: 1586225482@qq.com)

W. Meng is a lecturer of School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, P.R. China. (corresponding author to provide e-mail: mengwenlong_hit@163.com)

Most of the existing numerical approaches need to convert a parametric surface into a polygonal representation before computing geodesic paths or querying geodesic distances on the mesh surface. The main disadvantage lies in that it has to include a tedious and time-consuming discretization step of discretizing the continuous surface. For example, some known geodesic algorithms [10], [13], [14] have a worst-case $O(n^2 \log n)$ time complexity and an empirical $O(n^{1.5} \log n)$ time complexity, where n is the number of faces of the discrete mesh. The high time complexity limits their use in scenarios with a large number of polygonal faces.

In this paper, we advocate computing the geodesic path directly on a given parametric surface, without discretizing it into a polygonal mesh. First, we propose an optimization-driven method by minimizing a carefully designed geodesic energy functional, rather than finding the numerical solution to Euler–Lagrange equations. Note that explicit schemes for solving PDEs are stable only if the time step size is chosen sufficiently small, while implicit methods often need to solve a large algebraic system that must be solved (directly or iteratively) for the time integration on each of the space-time slices. Second, we suggest using Newton's method for quickly finding the minimizer of the optimization objective, which attains a second-order convergence rate. We conducted experiments on several commonly used parametric surfaces, and all the experimental results validate the effectiveness.

II. Related Work

A. Geodesics on smooth surfaces

Geodesic methods on smooth surfaces can be broadly classified as analytical and numerical methods. Since analytical approaches [15] are computationally expensive and have closed-form solutions that cannot be found for geodesic on general surfaces, the numerical methods are more widely used than the analytical methods. The Runge-Kutta method is a common numerical method to solve geodesic problems [16], [17], [18], [19] on smooth surfaces. It solves the non-linear differential geodesic equations and iteratively finds the approximate solution under the error threshold. Since the numerical solution involves computations at many points along the geodesic, it can be used as a convenient and efficient approach to trace the full path of the geodesic. This method is elegant and accurate, but the differential equations of geodesic are very complicated and generally not easy to solve.

In addition, geometric method [20], [21], [22] is a typical method of geodesic reporting on smooth surfaces. This method is based on the fundamental property that geodesics are a generalization of straight lines on plains. According to a point with a directional vector, the next point p_i can be recursively obtained by geometric operations. The geometric methods have the superiority that is independent of the surface complex description. However, the methods are complicated since they need to compute tangent planes of points in each step. Furthermore, they are only suitable for the "one point and one direction" situation. There are other elegant methods [23], [24], [25] with different techniques to compute geodesic on smooth surfaces. These methods fully utilize the property that the geodesic curvature at any point on the geodesic path is zero.

B. Geodesics on discrete surfaces

Computing a distance field rooted at a given source point aims to get distances from one source to any position of surfaces. No doubt that these methods can be used to solve the geodesic path problem. It is theoretically important since it serves as a base for solving the other variants of the geodesic problem. We refer to previous works [2] for comprehensive surveys. The global wavefront propagation methods [10], [26], [13], [14] inherit Dijkstra's spirit to propagate discrete wavefronts from near to far. Due to the global nature, these methods are able to find the global shortest path. The PDE methods [27], [28] for computing geodesic distance begin by formulating the problem in terms of partial differential equations (PDEs) on a smooth manifold, then discretizing and solving these PDEs via, e.g., finite element methods (FEM) or other numerical techniques. The graph-based methods [12], [29], [30], [31] rely on the assumption that the shortest geodesic distance/path between any pair of points p_s and p_t can be approximated with a chain of shortest distances/paths $(p_s, v_0, ..., v_k, p_t)$, where $v_0, ..., v_k$ belong to a finite set V_G of input polygonal S such that the shortest distance/path between pairs of V_G is precomputed and encoded in the edges E_G of a graph $G = (V_G, E_G)$.

As can be observed, the majority of the existing discrete geodesic algorithms rely on a certain kind of discrete tessellation to function. Therefore, the run-time performance and the accuracy depend on the resolution/quality of the tessellation. Furthermore, the discrete geodesics can not be computed directly on the original smooth surfaces. The triangulation process for the target surface is a valid way to find shortest paths, but it will introduce errors depending on the resolution and add extra computational costs.

III. Formulation in the Continuous Setting

The typical geodesic problem is defined as follows. Given a smooth parametric surface $\mathcal{S} = \mathcal{S}(u, v)$ in $\mathbb{R}^3$, i.e.,

$$\begin{cases} x = x(u, v) \\ y = y(u, v) \\ z = z(u, v), \end{cases} \tag{1}$$

as well as a source point at (u_s, v_s) and a destination point at (u_t, v_t). The task is to find a path $\Pi \in \mathcal{S}$ such that Π is the shortest one among all the paths that connect the source point $\mathcal{S}(u_s, v_s)$ and the destination point $\mathcal{S}(u_t, v_t)$.

Let Π have a parametric form of

$$\begin{cases} u = u(w) \\ v = v(w), \end{cases} \tag{2}$$

where $w \in [0, 1]$ is the intrinsic parameter of Π such that $u(0) = u_s$, $u(1) = u_t$, $v(0) = v_s$, and $v(1) = v_t$. We denote

$$\frac{\partial(x, y, z)}{\partial w} = \begin{pmatrix} \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial w} \\ \frac{\partial z}{\partial w} \end{pmatrix}, \tag{3}$$

$$\frac{\partial(x, y, z)}{\partial(u, v)} = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{pmatrix}, \tag{4}$$

and

$$\frac{\partial(u, v)}{\partial w} = \begin{pmatrix} \frac{\partial u}{\partial w} \\ \frac{\partial v}{\partial w} \end{pmatrix}. \tag{5}$$

Then we have

$$\frac{\partial(x, y, z)}{\partial w} = \frac{\partial(x, y, z)}{\partial(u, v)} \cdot \frac{\partial(u, v)}{\partial w}. \tag{6}$$

The length of Π can be measured by

$$L(\Pi) = \int_0^1 \left\| \frac{\partial(x, y, z)}{\partial w} \right\| \mathrm{d}w \tag{7}$$

If the surface is not equipped with an anisotropic metric. Eq. (7) can be also written as:

$$L(\Pi) = \int_0^1 \sqrt{\left(\frac{\partial(x, y, z)}{\partial w}\right)^T \frac{\partial(x, y, z)}{\partial w}} \mathrm{d}w, \tag{8}$$

or

$$\int_0^1 \sqrt{\left(\frac{\partial(u, v)}{\partial w}\right)^T \left(\frac{\partial(x, y, z)}{\partial(u, v)}\right)^T \frac{\partial(x, y, z)}{\partial(u, v)} \frac{\partial(u, v)}{\partial w}} \mathrm{d}w, \tag{9}$$

where

$$\left(\frac{\partial(x, y, z)}{\partial(u, v)}\right)^T \frac{\partial(x, y, z)}{\partial(u, v)} \tag{10}$$

defines a 2×2 matrix at (u, v). Under the circumstance that $\mathcal{S}$ is equipped with an anisotropic metric $\mathbf{T} = \mathbf{T}(u, v)$, the length of Π, in the anisotropic sense, can be written as

$$L(\Pi) = \int_0^1 \sqrt{\left(\frac{\partial(u, v)}{\partial w}\right)^T \cdot \mathbf{T} \cdot \frac{\partial(u, v)}{\partial w}} \mathrm{d}w, \tag{11}$$

where $\mathbf{T}$ defines a 2×2 matrix at each parameter pair (u, v). To this end, $L(\Pi)$ defines a variational function about the unknown curve. The shortest one can be found by minimizing $L(\Pi)$. However, it is hard to find an efficient numerical method for achieving this purpose. As pointed out in [32], Eq. (11) can be promoted by changing the integrand a little bit:

$$E(\Pi) = \int_0^1 \left(\frac{\partial(u, v)}{\partial w}\right)^T \cdot \mathbf{T} \cdot \frac{\partial(u, v)}{\partial w} \mathrm{d}w. \tag{12}$$

It can be shown that the minimizer of $E(\Pi)$ can also report the geodesic path. In fact, the nice feature of $E(\Pi)$ lies in that only those constant-speed curves have a chance to be a minimizer, which constrains the potential solution in a smaller search space. Even so, the analytic solution to Eq. (12) is mostly not available, which motivates us to seek for the discrete formation instead.

IV. Formulation in the Discrete Setting

Rather than report an analytic solution to Eq. (12), we aim at developing a numerical algorithm for finding a polygonal path that sufficiently approximates the real geodesic path. Considering that the polygonal path $\widetilde{\Pi}$ can be defined by a sequence of vertices, we assume that the vertices are respectively given by the following parameter pairs:

$$(u_s, v_s), (u_1, v_1), (u_2, v_2), \cdots, (u_n, v_n), (u_t, v_t), \tag{13}$$

where n is the number of vertices for defining $\widetilde{\Pi}$. We denote

$$(u_0, v_0) \triangleq (u_s, v_s), \quad (u_{n+1}, v_{n+1}) \triangleq (u_t, v_t). \tag{14}$$

When n is sufficiently large and the point sequence is sufficiently dense, we have

$$\sum_{i=1}^{n+1} d_{\mathbf{T}}(u_{i-1}, v_{i-1}; u_i, v_i) \geq L\left(1 - o(\frac{1}{n})\right), \tag{15}$$

where L is the length of the real geodesic path. According to the Cauchy-Schwarz inequality,

$$\sum_{i=1}^{n+1} d_{\mathbf{T}}^2(u_{i-1}, v_{i-1}; u_i, v_i) \geq \frac{1}{n+1}\left(\sum_{i=1}^{n+1} d_{\mathbf{T}}(u_{i-1}, v_{i-1}; u_i, v_i)\right)^2 \tag{16}$$

we further have

$$(n+1)\sum_{i=1}^{n+1} d_{\mathbf{T}}^2(u_{i-1}, v_{i-1}; u_i, v_i) \geq L^2\left(1 - o(\frac{1}{n})\right)^2, \tag{17}$$

where a necessary condition for making "=" hold is

$$d_{\mathbf{T}}(u_{i-1}, v_{i-1}; u_i, v_i) = d_{\mathbf{T}}(u_{j-1}, v_{j-1}; u_j, v_j) \tag{18}$$

for any $i \neq j$. Therefore, we define an optimization as follows:

$$\textbf{Minimize} \quad \widetilde{E} = (n+1)\sum_{i=1}^{n+1} d_{\mathbf{T}}^2(u_{i-1}, v_{i-1}; u_i, v_i), \tag{19}$$

where $d_{\mathbf{T}}^2(u_{i-1}, v_{i-1}; u_i, v_i)$ is given by

$$(u_i - u_{i-1}, v_i - v_{i-1})\, \widetilde{\mathbf{T}}\, (u_i - u_{i-1}, v_i - v_{i-1})^{\mathrm{T}} \tag{20}$$

and $\widetilde{\mathbf{T}}$ is the average anisotropic metric tensor defined between (u_{i-1}, v_{i-1}) and (u_i, v_i). Furthermore, all parameter pairs are constrained to be in the parameter domain. It's worth noting that the gradients of $\widetilde{E}$ can be computed easily:

$$\begin{aligned}\frac{\partial \widetilde{E}}{\partial(u_i, v_i)} =& 2\widetilde{\mathbf{T}}_i\,(u_i - u_{i-1}, v_i - v_{i-1})^{\mathrm{T}} \\ &+ 2\widetilde{\mathbf{T}}_i\,(u_i - u_{i+1}, v_i - v_{i+1})^{\mathrm{T}},\end{aligned} \tag{21}$$

where $\widetilde{\mathbf{T}}_i$ is the approximate anisotropic metric tensor in the small neighborhood of (u_i, v_i).

If we use isotropic density metric ρ, e.g, Gaussian density, to replace the anisotropic metric $\mathbf{T}$, it also works in the formulation. The Eq. 21 can be redefined as:

$$\begin{aligned}\frac{\partial \widetilde{E}}{\partial(u_i, v_i)} =& 2\widetilde{\rho}_i\,(u_i - u_{i-1}, v_i - v_{i-1})^{\mathrm{T}} \\ &+ 2\widetilde{\rho}_i\,(u_i - u_{i+1}, v_i - v_{i+1})^{\mathrm{T}},\end{aligned} \tag{22}$$

where $\widetilde{\rho}_i$ is the approximate isotropic density in the small neighborhood of (u_i, v_i).

a) Newton's iterative scheme: Let $\mathbf{x}$ be the combination of all variables. The iterative scheme of Newton's method [33], [34] proceeds based on the gradients $\nabla\widetilde{E}$ and :

$$\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} - \left(\nabla^2\widetilde{E}|_{\mathbf{x}^{[k]}}\right)^{-1}\nabla\widetilde{E}_{\mathbf{x}^{[k]}}. \tag{23}$$

Note that Newton's method is used to find stationary points. When one uses Newton's method [35] to solve the minimization problem, we have to enforce an additional constraint:

$$\widetilde{E}(\mathbf{x}^{[k+1]}) < \widetilde{E}(\mathbf{x}^{[k]}). \tag{24}$$

Since $-\nabla\widetilde{E}_{\mathbf{x}^{[k]}}$ defines the direction for decreasing the value of $\widetilde{E}$, we first check the assertion

$$\left(\nabla\widetilde{E}_{\mathbf{x}^{[k]}}\right)^{\mathrm{T}}\left(\nabla^2\widetilde{E}|_{\mathbf{x}^{[k]}}\right)^{-1}\nabla\widetilde{E}_{\mathbf{x}^{[k]}} \geq 0. \tag{25}$$

If the assertion is true, we perform a line search between $\mathbf{x}^{[k]}$ and $\mathbf{x}^{[k+1]}$. Otherwise, we change $\mathbf{x}^{[k+1]}$ as follows:

$$\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} + \left(\nabla^2\widetilde{E}|_{\mathbf{x}^{[k]}}\right)^{-1}\nabla\widetilde{E}_{\mathbf{x}^{[k]}}. \tag{26}$$

b) Line search: The purpose of line search is to find λ to minimize

$$g(\lambda) = \widetilde{E}\left((1-\lambda)\mathbf{x}^{[k]} + \lambda\mathbf{x}^{[k+1]}\right), \quad \lambda \in [0, 1]. \tag{27}$$

Based on Taylor expansion, we have

$$g(\lambda) \approx g(0) + g'(0)\lambda + \frac{g''(0)}{2}\lambda^2. \tag{28}$$

In this way, the best guess of λ can be obtained immediately. If the guess of λ cannot lead to a smaller objective value than $\mathbf{x}^{[k]}$, we then update

$$\mathbf{x}^{[k+1]} = (1-\lambda)\mathbf{x}^{[k]} + \lambda\mathbf{x}^{[k+1]} \tag{29}$$

and continue finding a different λ. It can be proved that this iterative scheme can report a monotonically decreasing sequence

$$\widetilde{E}(\mathbf{x}^{[1]}), \widetilde{E}(\mathbf{x}^{[2]}), \cdots, \tag{30}$$

which ensures the convergence. We summarize the pseudo-code of the algorithm in Algorithm 1.

V. Extension to Implicit Surfaces

Let $f(x, y, z) = 0$ be the implicit representation. Suppose that the points sequence is

$$(x_0, y_0, z_0), (x_1, y_1, z_1), \cdots, (x_n, y_n, z_n), (x_{n+1}, y_{n+1}, z_{n+1}). \tag{31}$$

The objective function, in this case, becomes

$$\widetilde{E} = (n+1)\sum_{i=1}^{n+1} d_{\mathbf{T}}^2(x_{i-1}, y_{i-1}, z_{i-1}; x_i, y_i, z_i), \tag{32}$$

where $(\{(x_i, y_i, z_i)\}_{i=1}^n)$ are constrained to be lying on the implicit surface, i.e.,

$$f(x_i, y_i, z_i) = 0, \quad i = 1, 2, \cdots, n. \tag{33}$$

Instead of adding hard constraints to Eq. (32), we simply solve an unconstrained optimization problem by projecting a moveable point $\mathbf{x} = (x, y, z)$ onto the implicit surface before evaluating the objective function of Eq. (32). Mathematically, the projection $\mathbf{x}^{\perp}$ of a point $\mathbf{x}$ can be defined as follows:

$$\mathbf{x}^{\perp} = \arg\min_{f(\mathbf{x}')=0}\|\mathbf{x}' - \mathbf{x}\|^2. \tag{34}$$

Algorithm 1 Computing Shortest Paths on Parametric Surfaces with Newton's Method

Input: A smooth parametric surface $\mathcal{S}=\mathcal{S}(u,v)$; A source p and a destination q; An initial path $\gamma^{[0]}(p,q)$; An error tolerance ε.

Output: A geodesic path $\gamma^{*}(p,q)$ on the parametric surface $\mathcal{S}$;

Extract a initial point sequence $\{(u_i^{[0]},v_i^{[0]})\}_{i=1}^{n}$ from the initial path $\gamma^{[0]}(p,q)$;
Compute the objective function $\widetilde{E}(\mathbf{x}^{[0]})$, as well as its gradient function $\nabla\widetilde{E}_{\mathbf{x}^{[0]}}$;
$j:=0$;
while $\left\|\nabla\widetilde{E}_{\mathbf{x}^{[j]}}\right\|\geq\varepsilon$ **do**
    Compute the gradient function $\nabla\widetilde{E}_{\mathbf{x}^{[j]}}$;
    Compute the hessian matrix $\nabla^2\widetilde{E}|_{\mathbf{x}^{[j]}}$;
    if $(\nabla\widetilde{E}_{\mathbf{x}^{[j]}})^{\mathrm{T}}(\nabla^2\widetilde{E}|_{\mathbf{x}^{[j]}})^{-1}\nabla\widetilde{E}_{\mathbf{x}^{[j]}}\geq 0$ **then**
        Perform one iteration of Newton's method by using $\mathbf{x}^{[j+1]}=\mathbf{x}^{[j]}-(\nabla^2\widetilde{E}|_{\mathbf{x}^{[j]}})^{-1}\nabla\widetilde{E}_{\mathbf{x}^{[j]}}$;
    else
        $\mathbf{x}^{[j+1]}=\mathbf{x}^{[j]}+(\nabla^2\widetilde{E}|_{\mathbf{x}^{[j]}})^{-1}\nabla\widetilde{E}_{\mathbf{x}^{[j]}}$;
    end if
    Generate a set of points $\{(u_i^{[j+1]},v_i^{[j+1]})\}_{i=1}^{n}$ according to $\mathbf{x}^{[j+1]}$;
    Compute the objective function $\widetilde{E}(\mathbf{x}^{[j+1]})$;
    $j:=j+1$;
end while
Transform the sequence $\{(u_i^{[j+1]},v_i^{[j+1]})\}_{i=0}^{n+1}$ onto the parametric surface.

We denote the initial position $\mathbf{x}$ by $\mathbf{x}^{(0)}$. According to

$$0=f(\mathbf{x}^{\perp})\approx f(\mathbf{x}^{(0)})+(\nabla f|_{\mathbf{x}^{(0)}})\cdot(\mathbf{x}^{\perp}-\mathbf{x}^{(0)}),\tag{35}$$

and

$$\mathbf{x}^{(0)}-\mathbf{x}^{\perp}\approx\lambda\nabla f|_{\mathbf{x}^{(0)}},\tag{36}$$

we repeat the following iterative scheme

$$\mathbf{x}^{(i+1)}=\mathbf{x}^{(i)}-\frac{f(\mathbf{x}^{(i)})}{\|\nabla f|_{\mathbf{x}^{(i)}}\|^2}\nabla f|_{\mathbf{x}^{(i)}},\tag{37}$$

and finally obtain

$$\mathbf{x}^{\perp}:=\lim_{i\to\infty}\mathbf{x}^{(i)}.\tag{38}$$

VI. Evaluation

A. Tests on parametric and implicit surfaces

In this paper, we mentioned geometric domains such as parametric surfaces and implicit surfaces. There are some classic surfaces as follows. The geodesic paths on these surfaces can be seen in Figure 1.

The Saddle surface model has a parametric form:

$$\begin{cases} x(u,v)=u\\ y(u,v)=v\\ z(u,v)=u^2-v^2\end{cases},\tag{39}$$

where $-1\leq u\leq 1$ and $-\pi/3\leq v<\pi/3$.

The Paraboloid model has a parametric form:

$$\begin{cases} x(u,v)=u\\ y(u,v)=v\\ z(u,v)=u^2+v^2\end{cases},\tag{40}$$

where $-1\leq u\leq 1$ and $-\pi/3\leq v<\pi/3$.

The Spiral surface model has a parametric form:

$$\begin{cases} x(u,v)=u\cos v\\ y(u,v)=u\sin v\\ z(u,v)=v\end{cases},\tag{41}$$

where $-1\leq u\leq 1$ and $-\pi\leq v<\pi$.

The Cylindrical surface model has a parametric form:

$$\begin{cases} x(u,v)=\cos u\\ y(u,v)=\sin u\\ z(u,v)=v\end{cases},\tag{42}$$

where $0\leq u\leq 2\pi$ and $-1\leq v\leq 1$.

The Three tori model has an implicit form:

$$\begin{cases} F(x,y,z)=F_1\cdot F_2\cdot F_3-r\\ F_1=\left(x^2+y^2+z^2+R^2-a^2\right)^2-4R^2\left(x^2+y^2\right)\\ F_2=\left(x^2+y^2+z^2+R^2-a^2\right)^2-4R^2\left(x^2+z^2\right)\\ F_3=\left(x^2+y^2+z^2+R^2-a^2\right)^2-4R^2\left(y^2+z^2\right)\\ F(x,y,z)=0\end{cases},\tag{43}$$

where $-2\leq x,y,z\leq 2$ and $R=1,a=0.2,r=0.01$.

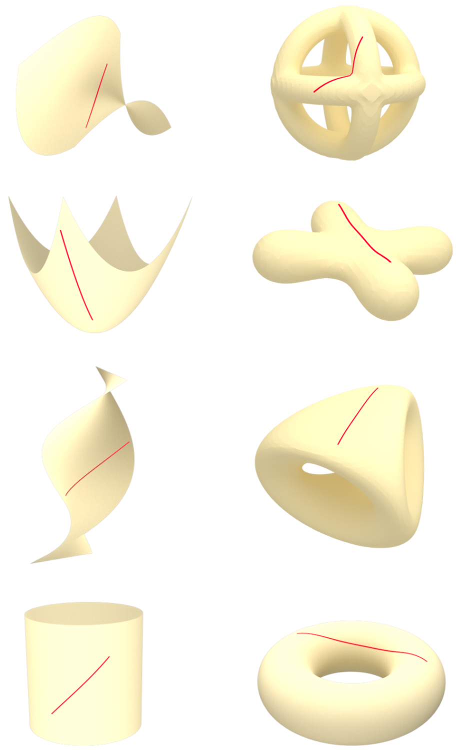


Fig. 1. Geodesic paths (red) on parametric surfaces (first column) and implicit surfaces (second column).

The Metamorphosis model has an implicit form:

$$\begin{cases} F(x,y,z)=F_1\cdot F_2\cdot F_3\cdot F_4-1.1\\ F_1=\sqrt{(x-1)^2+y^2+z^2}\\ F_2=\sqrt{(x+1)^2+y^2+z^2}\\ F_3=\sqrt{x^2+(y-1)^2+z^2}\\ F_4=\sqrt{x^2+(y+1)^2+z^2}\\ F(x,y,z)=0\end{cases},\tag{44}$$

where $-2\leq x,y,z\leq 2$.

The Genus 2 model has an implicit form:

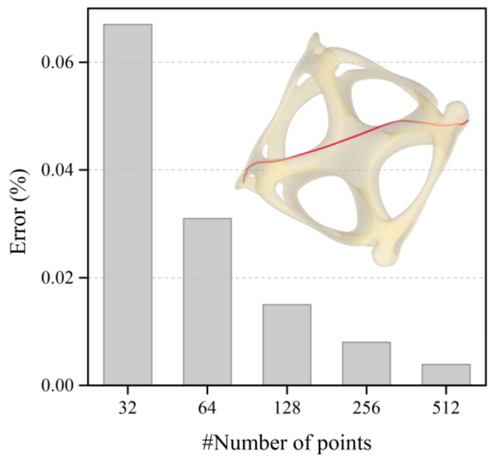
$$\left\{\begin{array}{l} F(x,y,z)=F_1+F_2-F_3 \\ F_1=2y\left(y^2-3x^2\right)\left(1-z^2\right) \\ F_2=\left(x^2+y^2\right)^2 \\ F_3=\left(9z^2-1\right)\left(1-z^2\right) \\ F(x,y,z)=0 \end{array}\right. , \quad (45)$$

where $-2 \leq x,y,z \leq 2$.

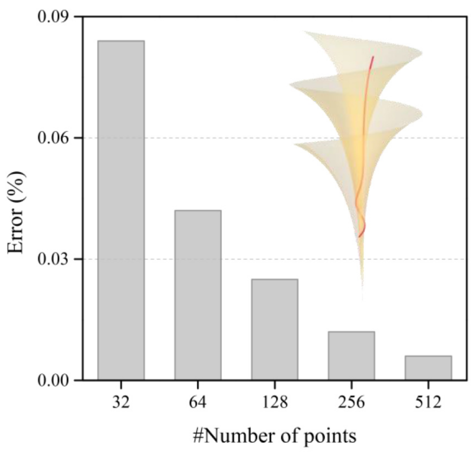
The Torus model has an implicit form:

$$\left(x^2+y^2+z^2+R^2-a^2\right)^2-4R^2\left(x^2+y^2\right)=0, \quad (46)$$

where $-4 \leq x,y,z \leq 4$ and $R=4, a=1.5$.



(a) An implicit surface



(b) A parametric surface

Fig. 2. Plot how relative errors of shortest paths depend on the number of inserted points. We employ 32, 64, 128, 256 and 512 points to describe the target paths on the implicit surface Banchoff-Chmutov in (a) and the parametric surface Dini in (b).

B. Error control

In the proposed method, we can set the number of intermediate points to obtain paths with varying precision. For example, we employ 32, 64, 128, 256 and 512 points to describe the target paths on the parametric surface Dini, shown in Figure 2(b). As the number of points increases, the accuracy of paths rapidly improves. When the inserted points number is 64, the relative error is only 0.042%, which is a competitive result in practical application scenarios [2], [36]. The Banchoff-Chmutov surface shown in Figure 2(a) has an implicit form:

$$\left\{\begin{array}{l} F(x,y,z)=F_1+F_2+F_3-F_4 \\ F_1=\left(3\left(x-1\right)x^2\left(x+1\right)+2y^2\right)^2 \\ F_2=\left(z^2-0.85\right)^2\left(3\left(y-1\right)y^2\left(y+1\right)+2z^2\right)^2 \\ F_3=\left(x^2-0.85\right)^2\left(3\left(z-1\right)z^2\left(z+1\right)+2x^2\right)^2 \\ F_4=0.12\left(y^2-0.85\right)^2 \\ F(x,y,z)=0 \end{array}\right. , \quad (47)$$

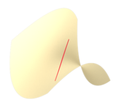
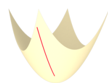
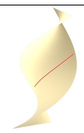
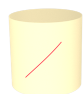
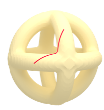
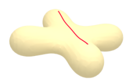
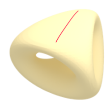
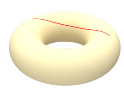
where $-5 \leq x,y,z \leq 5$.

The Dini surface shown in Figure 2(b) has a parametric form:

$$\left\{\begin{array}{l} x(u,v)=2\cos u\sin v \\ y(u,v)=2\sin u\sin v \\ z(u,v)=2\left(\cos v+\ln\left(\tan\left(\frac{v}{2}\right)\right)+0.4u\right) \end{array}\right. , \quad (48)$$

where $-2\pi \leq u \leq 2\pi$ and $0.1 < v < 2$.

TABLE I
The statistics of the average number of iterations and relative error on parametric and implicit surfaces shown in Figure 1

<Surfaces>	<Iterations*>	<Relative errors>
	2.4	0.017‰
	6.9	0.0051‰
	6	0.018‰
	4.7	0.012‰
	7.1	0.079‰
	4.3	0.024‰
	4.9	0.0069‰
	5.4	0.087‰

$< Iterations^* >$ indicates average number of iterations.

C. Empirical time complexity

With the support of gradients, we use Newton's method to minimize $\tilde{E}$. The termination condition is $\left\|\nabla\tilde{E}_{\mathbf{x}^{[k]}}\right\| \leq 10^{-6}$.

As is shown in Table I, we test the number of iterations on parametric and implicit surfaces in Figure 1. It can be seen that only $2 \sim 8$ iterations are required to achieve an accuracy level of 10^{-6} on these surfaces.

D. Run-time performance

Our method has an error-controlled feature using the number of inserted points. The parameter K indicates the number of inserted points on the initial path. Without a doubt, using more inserted points can increase accuracy but at the expense of higher computing costs. However, our method has good resistance to inserted points. It means that the time consumption grows slowly as the number of inserted points increases. For example, when we set K to be 50, the timing cost is about 0.007s. When K increases to 500, the timing cost is only about 0.091s. See Figure 3 for more details on time and parameter K.

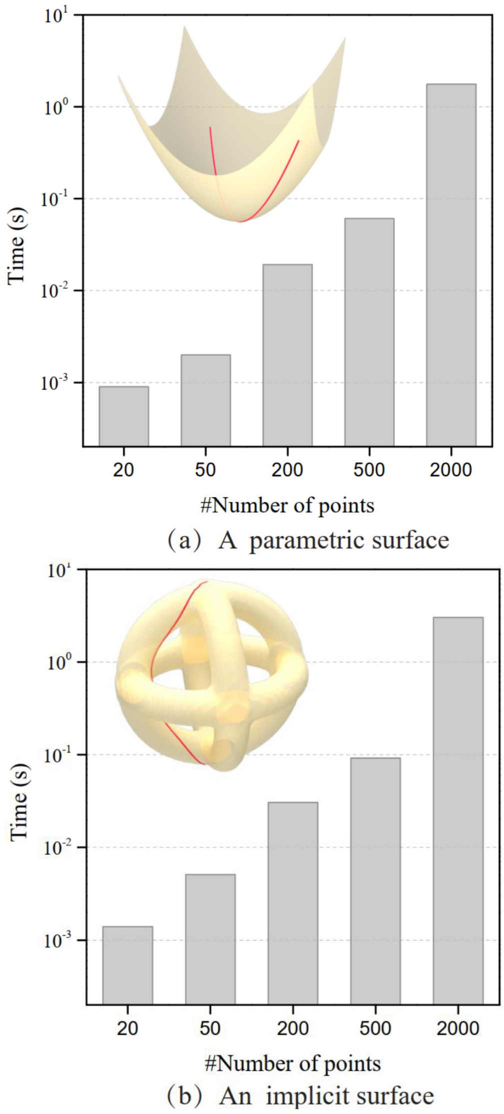


Fig. 3. Plot on how the timing cost depends on the number of inserted points. The timing cost slowly grows as the number of inserted points increases.

E. Path initialization

The initialization is vital for an optimization method to work successfully. When input surfaces have no local humps, the geodesic path is generally unique. In this situation, our algorithm is independent of initialization. In Figure 4, different initialization paths can obtain the same final path.

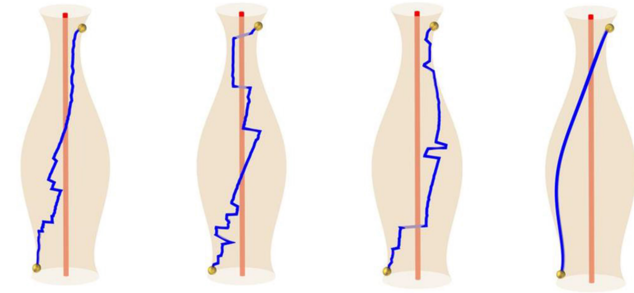
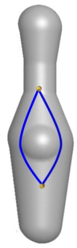


Fig. 4. Different initialization paths (blue) in (a)-(c) can obtain same final path in (d).



However, we can construct some examples where the geodesic paths are not unique. If the input model contains some spherical points, one can find two points on the surface such that the geodesics are not unique (the two equal-length geodesic paths are within the same equivalence class).

F. Qualitative comparison

In this section, we compare four typical and elegant methods that are most relevant to our method. Since our method focuses on solving the shortest paths rather than distance fields, we do not further compare with methods solving distance fields in this paper. In Table II, we show the qualitative comparison of four representative methods and discuss these methods as follows.

Geodesic trajectories on tubular surfaces [17]. They began by defining tubular surfaces, which are defined by specifying a centreline curve $X_0(t)$ and a radius function $R(t)$. It is easy to get a geodesic equation on the tubes given by a centreline curve and a radius function. Then the problem can be solved by geodesic equation using a second-order Runge-Kutta method. Compared with our algorithm, their method is quite complicated and, in general, not easy to solve. Since they used a second-order Runge-Kutta method to solve geodesic equation, the accuracy of the geodesic path computed is barely satisfactory and they are unable to control the path error. Moreover, they can only compute geodesics on tubes with a circular cross section perpendicular to the center curve, whereas our method is not limited in this way.

Geometric method on parametric surfaces [21]. They proposed a geometric method for tracing geodesic on parametric surfaces. The presented approach is independent of the complex description of the geodesic equations which has been a common solution in previous works. In detail, they started to trace a geodesic on surface S using start point p_0 and directional vector t_0 as initial values. Then based on the previous point p_{i-1} and directional vector t_{i-1} the next p_i and t_i can be recursively got by solving a universal equation. The subsequent geodesic points can be computed until the boundary of the surface is reached or the number of computed points exceeds a maximal point number. Since the curvature k has a significant impact on the final geodesic

TABLE II
We list five features of geodesic algorithms to make a qualitative comparison with four representative methods.

Methods	Various input types	Geometric constraints	Density function	Anisotropy metric	Error control
Ours	✔	✔	✔	✔	✔
[17]	✗	✗	✗	✗	✗
[21]	✗	✗	✗	✗	✗
[37]	✗	✔	✔	✔	✗
[38]	✗	✔	✗	✗	✔

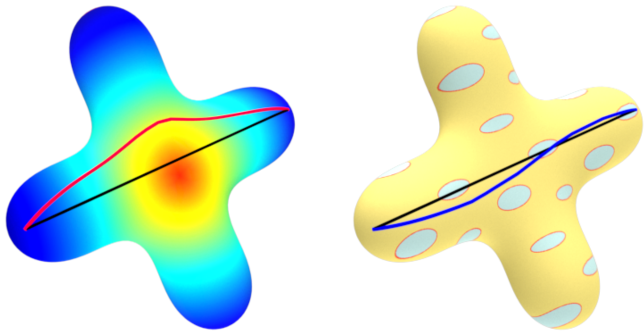
result, their method works better when the parametric surface is relatively flat. Besides, our method can get a better balance between accuracy and time compared with their approach.

Iterative unfolding method [37]. We compared the fast iterative unfolding scheme of Xin and Wang, which repeatedly computes the exact shortest path in unfolded triangle strips. This scheme traverses similar sequences of curves during the iterative procedure, generating increasingly short geodesic polylines between vertices, which is very time-consuming. In addition, since their algorithm cannot be applied directly to smooth surfaces, the target surface must first be transformed into a mesh before using algorithm. For a fair comparison, we use the March Cube (MC) method to extract the isosurface of the parametric surface with a uniform resolution of 1024 $\times$1024 $\times$ 1024. On a Saddle surface, our algorithm runs in 0.09 seconds with lower error than the iterative unfolding method, which takes 2.60 seconds.

Variational framework for geodesic paths [38]. In order to compute geodesic paths between two points on the sweep surfaces, they proposed a variational framework on the 2D parametric domain, instead of discretizing the surface into a polygonal mesh. A polyline curve with n vertices was used in the implementation to roughly depict the geodesic path, where n is a user-specified parameter for accuracy control. The optimal polyline curve can be found by minimizing the sum of the squared length of any two successive vertices. Different from the method that only supports sweep surfaces defined by straight guidelines, our approach is more general and can handle both parametric and implicit surfaces.

G. Geodesics with density/anisotropic metric

Most of the geodesic methods are designed for shortest distance paths/fields according to the standard Riemannian metric. However, geodesic distances based on other metrics, e.g., anisotropic metric [39], have gained more and more attention. The geodesic algorithm proposed in this paper can be easily updated to solve the problems such as paths with density metric or anisotropic metric mentioned in section IV. Figure 5(a) shows the shortest path results with the Gaussian density field. Different from the traditional path result, the path with density metric tries to avoid high-density areas that are very meaningful in many real applications [40], [41]. We assign a 2 $\times$ 2 matrix defined between (u_{i-1}, v_{i-1}) and (u_i, v_i) to denote the anisotropic metric $\widetilde{\mathbf{T}}$. According to the Eq.(21), our algorithm can efficiently compute the shortest paths under the configuration of anisotropic metric shown in Figure 5(b).



(a). Gaussian density metric (b). Anisotropic metric

Fig. 5. Geodesic paths with density/anisotropic metric. We use the black path to denote the geodesic result with normal metric. (a) the red path shows the geodesic result with the Gaussian density metric. The configuration of anisotropic metric shown in (b) and the blue path is the corresponding geodesic result.

VII. Conclusion

In this paper, we propose an optimized method to efficiently compute geodesics on the curved surface including parametric and implicit surfaces. Different from the traditional approaches, such as Runge-kutta methods, which need to handle complicated differential equations of geodesic, our method minimizes a carefully designed geodesic energy function rather than find the numerical solution to Euler–Lagrange equations. In detail, we suggest using Newton's method for quickly finding the minimizer of the optimization objective, which owns a second-order convergence rate. Besides, the proposed method is easily extended to compute the geodesic distances with density or anisotropic metrics, which is important in many real applications, e.g., navigation in motion planning. Experiments on several commonly used parametric and implicit surfaces validate the effectiveness of the proposed method in this paper.

References

[1] K. Polthier and M. Schmies, "Straightest geodesics on polyhedral surfaces," in *ACM SIGGRAPH 2006 Courses*, 2006, pp. 30–38.

[2] K. Crane, M. Livesu, E. Puppo, and Y. Qin, "A survey of algorithms for geodesic paths and distances," *Preprint ArXiv:2007.10430*, 2020.

[3] J. Jost, *Riemannian geometry and geometric analysis*. Springer, 2008, vol. 42005.

[4] B. L. Price, B. Morse, and S. Cohen, "Geodesic graph cut for interactive image segmentation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3161–3168.

[5] Y.-J. Liu, D. Fan, C.-X. Xu, and Y. He, "Constructing intrinsic Delaunay triangulations from the dual of geodesic Voronoi diagrams," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 2, pp. 1–15, 2017.

[6] J. Rabin, G. Peyré, and L. D. Cohen, "Geodesic shape retrieval via optimal mass transport," in *European Conference on Computer Vision*. Springer, 2010, pp. 771–784.
[7] J. Ye and Y. Yu, "A fast modal space transform for robust nonrigid shape retrieval," *The Visual Computer*, vol. 32, no. 5, pp. 553–568, 2016.
[8] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 37–45.
[9] T. He, H. Huang, L. Yi, Y. Zhou, C. Wu, J. Wang, and S. Soatto, "Geonet: Deep geodesic networks for point cloud analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6888–6897.
[10] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe, "Fast exact and approximate geodesics on meshes," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 553–560, 2005.
[11] R. Kimmel and J. A. Sethian, "Computing geodesic paths on manifolds," *Proceedings of the National Academy of Sciences*, vol. 95, no. 15, pp. 8431–8435, 1998.
[12] W. Meng, S. Xin, C. Tu, S. Chen, Y. He, and W. Wang, "Geodesic tracks: Computing discrete geodesics with track based Steiner point propagation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 4887–4901, 2021.
[13] S.-Q. Xin and G.-J. Wang, "Improving Chen and Han's algorithm on the discrete geodesic problem," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 4, pp. 1–8, 2009.
[14] Y. Qin, X. Han, H. Yu, Y. Yu, and J. Zhang, "Fast and exact discrete geodesic computation based on triangle oriented wavefront propagation," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–13, 2016.
[15] M. P. Do Carmo, *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.
[16] J. M. Beck, R. T. Farouki, and J. K. Hinds, "Surface analysis methods," *IEEE Computer Graphics and Applications*, vol. 6, no. 12, pp. 18–36, 1986.
[17] J. Sneyd and C. S. Peskin, "Computation of geodesic trajectories on tubular surfaces," *SIAM Journal on Scientific and Statistical Computing*, vol. 11, no. 2, pp. 230–241, 1990.
[18] X. Chen, S.-Y. He, D.-F. Yu, H.-C. Yin, W.-D. Hu, and G.-Q. Zhu, "Geodesic computation on NURBS surfaces for UTD analysis," *IEEE Antennas and Wireless Propagation Letters*, vol. 12, pp. 194–197, 2013.
[19] E. Kasap, M. Yapici, and F. T. Akyildiz, "A numerical study for computation of geodesic curves," *Applied Mathematics and Computation*, vol. 171, no. 2, pp. 1206–1213, 2005.
[20] I. Hotz and H. Hagen, "Visualizing geodesics," in *Proceedings Visualization 2000*. IEEE, 2000, pp. 311–318.
[21] P. Zhang, R. Sun, and T. Huang, "A geometric method for computation of geodesic on parametric surfaces," *Computer Aided Geometric Design*, vol. 38, pp. 24–37, 2015.
[22] P. Zhang, J. Xiong, and L. Huang, "An efficient method for computation of geodesic on B spline surfaces," in *2019 International Conference on Advances in Construction Machinery and Vehicle Engineering (ICACMVE)*. IEEE, 2019, pp. 311–316.
[23] T. Maekawa, "Computation of shortest paths on free form parametric surfaces," 1996.
[24] L. Ying and E. J. Candes, "Fast geodesics computation with the phase flow method," *Journal of Computational Physics*, vol. 220, no. 1, pp. 6–18, 2006.
[25] S.-G. Chen, "Geodesic like curves on parametric surfaces," *Computer Aided Geometric Design*, vol. 27, no. 1, pp. 106–117, 2010.
[26] J. Chen and Y. Han, "Shortest paths on a polyhedron," in *Proceedings of the Sixth Annual Symposium on Computational Geometry*, 1990, pp. 360–369.
[27] J. Solomon, R. Rustamov, L. Guibas, and A. Butscher, "Earth mover's distances on discrete surfaces," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–12, 2014.
[28] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in heat: A new approach to computing distance based on heat flow," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 5, pp. 1–11, 2013.
[29] L. Aleksandrov, M. Lanthier, A. Maheshwari, and J.-R. Sack, "An ε approximation algorithm for weighted shortest paths on polyhedral surfaces," in *Scandinavian Workshop on Algorithm Theory*. Springer, 1998, pp. 11–22.
[30] S.-Q. Xin, X. Ying, and Y. He, "Constant time all pairs geodesic distance query on triangle meshes," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2012, pp. 31–38.
[31] X. Ying, X. Wang, and Y. He, "Saddle vertex graph (SVG) a novel solution to the discrete geodesic problem," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, pp. 1–12, 2013.
[32] L. P. Eisenhart, "Riemannian geometry," in *Riemannian Geometry*. Princeton university press, 2016.
[33] H. Wang, and M. Qin, "A modified regularized Newton method for unconstrained convex optimization," IAENG International Journal of Applied Mathematics, vol. 46, no. 2, pp130-134, 2016.
[34] R. Haelterman, A. Bogaers, J. Degroote, and N. Boutet, "Quasi-Newton methods for the acceleration of multi-physics codes," IAENG International Journal of Applied Mathematics, vol. 47, no. 3, pp352-360, 2017.
[35] H. Fu, "Improved convergence results of a BFGS trust region quasi-Newton method for nonlinear equations," IAENG International Journal of Applied Mathematics, vol. 50, no. 4, pp767-771, 2020.
[36] G. Peyré and L. D. Cohen, "Geodesic methods for shape and surface processing," *Advances in Computational Vision and Medical Image Processing*, pp. 29–56, 2009.
[37] S.-Q. Xin and G.-J. Wang, "Efficiently determining a locally exact shortest path on polyhedral surfaces," *Computer-Aided Design*, vol. 39, no. 12, pp. 1081–1090, 2007.
[38] W. Meng, S. Xin, J. Zhao, S. Chen, C. Tu, and Y. He, "A variational framework for computing geodesic paths on sweep surfaces," *Computer-Aided Design*, vol. 140, p. 103077, 2021.
[39] M. Campen, M. Heistermann, and L. Kobbelt, "Practical anisotropic geodesy," in *Computer Graphics Forum*, vol. 32, no. 5. Wiley Online Library, 2013, pp. 63–71.
[40] G. Peyré, M. Péchaud, R. Keriven, L. D. Cohen *et al.*, "Geodesic methods in computer vision and graphics," *Foundations and Trends® in Computer Graphics and Vision*, vol. 5, no. 3–4, pp. 197–397, 2010.
[41] S. K. Lee, S. P. Fekete, and J. McLurkin, "Structured triangulation in multi robot systems: Coverage, patrolling, voronoi partitions, and geodesic centers," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1234–1260, 2016.