Mesh Watermark Removal Attack and Mitigation: A Novel Perspective of Function Space

Xingyu Zhu[1,2], Guanhui Ye[1], Chengdong Dong[2], Xiapu Luo[2], Shiyao Zhang[1], Xuetao Wei[1]*

[1]Department of Computer Science and Engineering, Southern University of Science and Technology, China
[2]Department of Computing, Hong Kong Polytechnic University, Hong Kong
12150086@mail.sustech.edu.cn, 12132370@mail.sustech.edu.cn, chengdong.dong@connect.polyu.hk, csxluo@comp.polyu.edu.hk, zhangsy@sustech.edu.cn, weixt@sustech.edu.cn

Abstract

Mesh watermark embeds secret messages in 3D meshes and decodes the message from watermarked meshes for ownership verification. Current watermarking methods directly hide secret messages in vertex and face sets of meshes. However, mesh is a discrete representation that uses vertex and face sets to describe a continuous signal, which can be discretized in other discrete representations with different vertex and face sets. This raises the question of whether the watermark can still be verified on the different discrete representations of the watermarked mesh. We conduct this research in an attack-then-defense manner by proposing a novel function space mesh watermark removal attack FUNCEVADE and then mitigating it through function space mesh watermarking FUNCMARK. In detail, FUNCEVADE generates a different discrete representation of a watermarked mesh by extracting it from the signed distance function of the watermarked mesh. We observe that the generated mesh can evade ALL previous watermarking methods. FUNCMARK mitigates FUNCEVADE by watermarking signed distance function through message-guided deformation. Such deformation can survive isosurfacing and thus be inherited by the extracted meshes for further watermark decoding. Extensive experiments demonstrate that FUNCEVADE achieves 100% evasion rate among all previous watermarking methods while achieving only 0.3% evasion rate on FUNCMARK. Besides, our FUNCMARK performs similarly on other metrics compared to state-of-the-art mesh watermarking methods.

1 Introduction

Triangle meshes are the primary representation of 3D geometry in computer graphics. They predominantly represent 3D assets used in video games, movies, manufacturing, and virtual reality interfaces (Siddiqui et al. 2023; Pavllo et al. 2020). Losing a high-fidelity mesh can raise ethical challenges. For example, a mesh stealer can falsely claim ownership of the high-fidelity mesh. One way to address such a challenge is by leveraging digital watermarking techniques (Zhu et al. 2024; Fernandez et al. 2023).

Digital watermarking hides secret messages on digital content like images and meshes for copyright protection, source tracking, and authentication. The watermark is traditionally embedded in the discrete representation of the digital contents. For example, image watermarking embeds a watermark by perturbating image pixels, and mesh watermarking embeds a watermark by perturbing vertex coordinates. However, the underlying signal represented by digital content is continuous. A mesh can be viewed as a discrete representation of a continuous signal, *i.e.*, signed distance function (SDF) (Dupont et al. 2022; Mescheder et al. 2019; Sitzmann et al. 2020), from which a high-quality mesh can be extracted as a different discrete representation. Such a different discrete representation shares the same perceptual quality while having a different vertex and face set (or topology). If we obtain a watermarked mesh, the question is: *can the watermark be verified on its other discrete representations?*

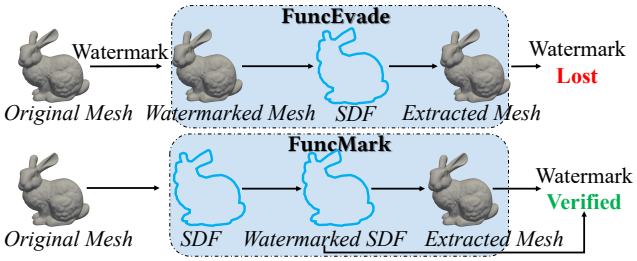


Figure 1: Top: FUNCEVADE fits the signed distance function (SDF) of the watermarked mesh. It gets an extracted mesh from SDF through isosurfacing, where the extracted mesh successfully evades ALL previous watermarking methods. Bottom: To mitigate FUNCEVADE, FUNCMARK watermark SDF instead of mesh, where the watermark can be verified on SDF and the mesh extracted from it.

A watermarking method (Zhu et al. 2024; Wang et al. 2022; Al-Khafaji and Abhayaratne 2019; Peng, Long, and Long 2021; Peng, Liao, and Long 2022; Tsai and Liu 2022; Tsai 2020; Hou, Kim, and Lee 2017; Hou et al. 2023) consists of embedding, decoding and detection phase. In the embedding phase, the mesh owner embeds a secret message into the original mesh to produce a watermarked mesh. In the decoding phase, a message is decoded from an input mesh (a watermarked mesh or an original mesh without a

*Corresponding Author.

watermark). In the detection phase, the input mesh is judged as watermarked if the bitwise accuracy of the decoded message is larger than a threshold τ, where bitwise accuracy is the fraction of matched bits in the decoded message and the ground-truth one. For non-learning-based watermarking methods (Al-Khafaji and Abhayaratne 2019; Peng, Long, and Long 2021; Peng, Liao, and Long 2022; Tsai 2020; Tsai and Liu 2022; Hou et al. 2023), the embedding and decoding algorithm are designed based on heuristics. For learning-based watermarking methods (Wang et al. 2022; Zhu et al. 2024), the embedding and decoding algorithm is built with a deep neural network-based encoder and decoder. Both types of methods embed secret messages on the vertex coordinates of meshes.

Robustness against watermark removal, which post-processes a watermarked mesh to evade watermark detection, is a crucial evaluation metric for watermarking methods. Popular post-processing methods such as Gaussian noise, affine transformation, Laplacian smoothing, and quantization post-process meshes in discrete space, where the post-processed meshes have the same topology and vertex number as the pre-processed meshes, and they only differ in vertex coordinates. **Research Gap:** existing works (Zhu et al. 2024; Peng, Long, and Long 2021; Peng, Liao, and Long 2022; Wang et al. 2022) only evaluated the robustness against methods which post-process meshes in discrete space, leaving their robustness against the technique which changes the mesh topology unexplored.

We aim to bridge this gap in an attack-then-defense manner. As shown in Fig. 1, we first propose FuncEvade, a function space post-processing method, to evade watermark detection. Given a watermarked mesh, FuncEvade generates its different discrete representation (with different vertex sets and topology) by extracting it from the watermarked mesh's signed distance function (SDF). Compared with discrete space post-process methods, FuncEvade achieves the highest evasion rate while keeping the highest post-processed mesh quality. Our experiment shows that FuncEvade successfully evades all mesh watermarking methods. To mitigate the function space watermark removal, we propose FuncMark, a function space watermarking method. Instead of watermarking a mesh, FuncMark watermarks its continuous signal (*i.e.* SDF). FuncMark embeds secret messages in SDF by spherical partitioning and local deformation. Spherical partitioning divides 3D space into multiple partitions to embed one bit in each partition. The local deformation is performed within each partition based on the bit embedded in the current partition. By doing so, 3D meshes extracted from the deformed SDF inherit such deformation and can be applied for further message decoding. Due to the nature of function space watermarking, FuncMark successfully mitigates FuncEvade and remeshing attacks. We summarize our contributions as the following:

- We take a first step towards investigating the low robustness of all previous mesh watermarking methods under topology attacks, and we propose FuncEvade, a new instance of topology attack that evades watermark detection by generating a different discrete representation of the watermarked mesh.
- To mitigate the function space watermark removal, we propose FuncMark, which embeds a secret message on the signed distance function through spherical partitioning and message-guided deformation. The secret message can be decoded from either the SDF or meshes extracted from the SDF.
- We conduct extensive experiments to show that FuncEvade achieves the highest evasion rate among popular removal methods while keeping the highest post-processed mesh quality, and FuncMark successfully mitigates FuncEvade while keeping similar performance with current SOTA on other metrics.

2 Related Work and Preliminary

2.1 Mesh Watermark

Encrypted domain watermarking methods (Jiang et al. 2017; Tsai 2020; Tsai and Liu 2022; Hou et al. 2023) encrypt vertex coordinates of a mesh by stream encryption before watermark embedding. The secret message embedding is conducted in the encrypted domain to generate a watermarked encrypted mesh, which can be further used for watermark decoding and plain-text mesh recovery. However, watermark decoding can only be conducted in the encrypted domain because the watermark cannot be verified on the recovered plain-text mesh. The function space of encrypted watermarked mesh is meaningless because encrypted meshes are not meaningful signals. Hence, our work focuses on plain-text domain mesh watermarking.

Plain-text domain watermarking methods can be categorized into learning-based and non-learning-based methods. In non-learning-based methods (Al-Khafaji and Abhayaratne 2019; Peng, Long, and Long 2021; Peng, Liao, and Long 2022; Praun, Hoppe, and Finkelstein 1999), watermark embedding and decoding are hand-crafted algorithms based on heuristics. In learning-based methods (Zhu et al. 2024), embedding and decoding algorithms use encoders and decoders built with deep neural networks, like Deep3DMark (Zhu et al. 2024). However, most of them only evaluated robustness against watermark removal methods which will not change the topology of the watermarked mesh, leaving the removal methods that will alter the mesh topology (such as remeshing) unexplored. Although (Praun, Hoppe, and Finkelstein 1999) has evaluated their robustness against remeshing, they cannot adapt to current deep learning trends, as they are traditional methods.

2.2 Signed Distance Function

The signed distance function (SDF) maps 3D locations $\mathbf{x}$ to a scalar, which represents the signed shortest distance to the 3D surface, with its first derivative representing the surface normal. Given a mesh $\mathcal{M}$ with vertex set V, all vertices $v \in V$ satisfy $F(v) = 0$, and $\nabla_v F(v)$ equals to the vertex normal at vertex v. In this work we follow the tradition (Atzmon and Lipman 2020; Michalkiewicz et al. 2019; Atzmon et al. 2019; Chen and Zhang 2019; Park et al. 2019; Sitzmann et al. 2020; Dupont et al. 2022) where an SDF is

parameterized by a deep neural network F_Θ with the following unified formulation:

$$F_\Theta(\mathbf{x}) = \mathbf{W}_n(f_{n-1} \circ f_{n-2} \circ ... \circ f_1)(\mathbf{x}), \\ f_i(\mathbf{x}) = \sigma_i(\mathbf{W}_i\mathbf{x} + \mathbf{b}_i), \tag{1}$$

where $\mathbf{W}_i, \mathbf{b}_i$ are the weight matrix and bias of the i-th layer, and σ_i is an element-wise nonlinear activation function. σ_i is either ReLU or sinusoidal function used in SIREN (Sitzmann et al. 2020). SIREN pioneeringly applied sine transform to the input coordinates, enabling SDFs to represent high-frequency details better. This work uses SIREN as the backbone for our SDF parameterization.

Given an SDF, a mesh $\mathcal{M}$ can be extracted from it through isosurfacing. Isosurfacing first divides 3D space with voxel cubes with resolution r^3, *i.e.*, the 3D space is divided by r^3 voxel cubes. Given a scalar function $F(\mathbf{x})$, isosurfacing fits a surface to the points whose sample values are of a specific isovalue and whose position is determined by the edges of voxel cubes. For example, to extract a mesh from a signed distance function is to sample points $\mathbf{x}$ whose scalar value $F(\mathbf{x}) = 0$. These sampled points are then polygonized to build a surface model through marching cube (Lorensen and Cline 1998).

3 Mesh Watermark Detection

Alice embeds n-bit binary message (as her signature) into a mesh. The watermark decoding algorithm then decodes messages from the mesh it receives and detects the watermark when the message is close to the ground truth message. We judge whether the watermark is detected through the following test.

Statistical test. Let $w \in \{0,1\}^n$ be Alice's signature. We extract the message w' from a mesh (the mesh can be non-watermarked mesh $\mathcal{M}$ or watermarked mesh $\mathcal{M}_w$) and compare it to w. As done in previous works (Luo et al. 2023; Fernandez et al. 2023; Jiang, Zhang, and Gong 2023), the detection test relies on the number of bitwise accuracy $BA(w, w')$: if

$$BA(w, w') \geq \frac{\tau}{n} \quad \text{where} \quad \tau \in \{0, ..., n\}, \tag{2}$$

then the mesh is judged as watermarked. This provides a level of robustness to imperfections of the watermarking.

We test the statistical hypothesis H_1: "The given mesh was watermarked by Alice" against the null hypothesis H_0: "The given mesh was not watermarked by Alice". Under H_0, bits $w'_0, ..., w'_{n-1}$ are (i.i.d.) Bernoulli random variables with parameter 0.5 (*i.e.* $BA(w, w') \sim B(k, 0.5)/n$). The False Positive Rate under τ (denoted as $FPR(\tau)$) is the probability that $BA(w, w')$ takes a value bigger than the threshold $\frac{\tau}{n}$. Formally, $FPR(\tau)$ can be written as:

$$FPR(\tau) = \Pr(n \cdot BA(w, w') \geq \tau) \\ = \sum_{i=\tau}^{n} \binom{n}{i} \frac{1}{2^n}. \tag{3}$$

The key point is to set the threshold τ such that the $FPR(\tau)$, *i.e.*, the probability that an original mesh is falsely detected as watermarked is bounded by a small value η, *e.g.*, $\eta = 0.05$. To make $FPR(\tau) < \eta$, τ should be at least $\tau^* = \arg\min_\tau FPR(\tau) < \eta$. For instance, when $n = 48$ and $\eta = 0.05$, we have $\tau \geq \tau^* = 31$.

4 Watermark Removal Analysis

We first introduce FUNCEVADE, which generates a different discrete representation of the watermarked mesh by extracting it from the signed distance function of the watermarked mesh. We compare the removal effectiveness among FUNCEVADE, remeshing, and other popular watermark removal methods such as Gauss noise, quantization, smoothing, and rotation. Besides, we expose and explain the vulnerability of all previous watermarking methods when facing function space watermark removal methods.

Removal Objective. Given a watermarked mesh $\mathcal{M}_w$, watermark removal aims to get a post-processed mesh $\mathcal{M}'_w$ so that $\mathcal{M}'_w$ can evade watermark detection and $\mathcal{M}'_w$ and $\mathcal{M}_w$ are highly similar. We evaluate evasion effectiveness by **evasion rate** and mesh similarity by **Hausdorff distance**. In the watermark removal problem, we expect a higher evasion rate and lower Hausdorff distance.

4.1 Removal Implementation

Since the watermark removal method requires high similarity between the watermarked mesh $\mathcal{M}_w$ and post-processed $\mathcal{M}'_w$, and FUNCEVADE extracts $\mathcal{M}'_w$ from SDF, one challenge is to build a high-quality SDF from the watermarked mesh $\mathcal{M}_w$. SDF predicts the signed shortest distance to the 3D surface given arbitrary 3D points $\mathbf{x} \in \mathbb{R}^3$. Hence, the first step is to sample points $\mathbf{x}$ and their ground truth SDF value to supervise the fitting of SDF.

Sampling strategy on mesh. In practice, sampling strategy significantly impacts the fitted SDF quality. Uniform sampling across $[-1, 1]^3$ usually gives sparse on-surface points, thus resulting in low surface fitting quality. We expect on-surface points to be accurately supervised, and off-surface points far from the surface are roughly supervised because we only need to extract zero-isosurface in the mesh extraction stage. We use package *mesh-to-sdf*[1] to produce ground truth SDF for 5M off-surface point set V_1 and ground truth normal vectors for 5M on-surface point set V_0.

SDF fitting strategy. We supervise SDF fitting process with both its function values F_Θ and first-order derivatives ∇F_Θ. Besides, SDF fitting requires solving a particular Eikonal boundary value problem that constrains the norm of first-order derivatives $|\nabla F_\Theta|$ to be 1 everywhere. Hence, we supervise SDF with the following objectives.

$$\mathcal{L}_{F_\Theta} = \int_{\mathbf{x} \in V_0} \lambda_1 (1 - \langle \nabla F_\Theta(\mathbf{x}), \nabla F(\mathbf{x}) \rangle) d\mathbf{x} + \\ \int_{\mathbf{x} \in V_0 \cup V_1} \lambda_2 \|F_\Theta(\mathbf{x}) - F(\mathbf{x})\| + \lambda_3 \| |\nabla F_\Theta(\mathbf{x})| - 1 \| d\mathbf{x} \tag{4}$$

where $\lambda_1 = 100, \lambda_2 = 3000, \lambda_3 = 5$. For on-surface points V_0, their ground truth signed distance values $F(\mathbf{x})$ and first-order derivatives $\nabla F_\Theta(\mathbf{x})$ are assigned with zero and their

[1] https://github.com/marian42/mesh_to_sdf

Removal Methods	SPECTRAL		PENG2021		PENG2022		WANG2022		DEEP3DMARK	
	EVA	HD	EVA	HD	EVA	HD	EVA	HD	EVA	HD
Gauss Noise	98.3%	0.021	97.8%	0.021	98.2%	0.021	0.20%	0.021	0.1%	0.021
Rotation	80.2%	/	92.1%	/	0%	/	18.4%	/	15.2%	/
Quantization	94.5%	0.015	93.8%	0.014	91.2%	0.014	0%	0.015	4.5%	0.014
Smoothing	99.3%	0.108	99.7%	0.107	99.1%	0.112	14.6%	0.116	4.9%	0.118
Remesh	**100%**	0.005	**100%**	0.006	**100%**	0.005	**100%**	0.006	**100%**	0.005
FUNCEVADE (Ours)	**100%**	**0.004**	**100%**	**0.003**	**100%**	**0.004**	**100%**	**0.003**	**100%**	**0.004**

Table 1: Watermark removal results. EVA and HD indicate evasion rate and Hausdorff distance. HD is evaluated between watermarked mesh and post-processed meshes. Lower HD means higher post-processed mesh quality. Note that "/" means that using HD to evaluate rotated mesh similarity is unreasonable because the rotated mesh is the same as the original one.

normal vectors, respectively. For off-surface points V_1, their ground truth signed distance values are assigned with values obtained in the sampling process. We use SIREN (Sitzmann et al. 2020) as the backbone of F_Θ, which consists of four *Linear* layers with their channel size set to 512. We use Adam optimizer with the initialized learning rate 10^{-3} for 1000 epochs, and we decrease the learning rate by half every 200 epochs. After obtaining the fitted SDF, the final step is to obtain the post-processed mesh $\mathcal{M}'_w$ by extracting it from SDF through marching cube (Lorensen and Cline 1998).

4.2 Removal Evaluation

Metrics and Datasets. We evaluate the evasion rate for the effectiveness of watermark removal and Hausdorff distance for mesh similarity between the watermarked mesh $\mathcal{M}_w$ and the post-processed mesh $\mathcal{M}'_w$. Let w' be the message extracted from $\mathcal{M}'_w$. We judge the removal method successfully evades the detection if $BA(w, w_A) < \frac{\tau}{n}$. We select the message length $n = 48$ and $\tau = 31$ in our experiment. In this case, we have a false positive rate $FPR(\tau) < 0.05$. We normalize meshes in ShapeNet (Chang et al. 2015) and Stanford Repo (Laboratory 2023) to $[-1, 1]^3$. All watermark embedding, extraction, and removal experiments are conducted on normalized meshes.

Examined watermarking methods. We examined both learning-based and non-learning-based methods. We denote them as SPECTRAL (Al-Khafaji and Abhayaratne 2019), PENG2021 (Peng, Long, and Long 2021), PENG2022 (Peng, Liao, and Long 2022), WANG2022 (Wang et al. 2022) and DEEP3DMARK (Zhu et al. 2024) for brevity. In detail, SPECTRAL decomposes the Laplacian matrix of a mesh into eigenvalues and eigenvectors and hides secret messages in eigenvalues. Both PENG2021 and PENG2022 transform vertex coordinates into a new coordinate system before they embed or decode the watermark on the vertex coordinates. The new coordinate system of PENG2021 is built based on the largest face of the input mesh, while PENG2022 is built based on the bounding sphere of the input mesh. WANG2022 and DEEP3DMARK train the encoder and decoder to embed or decode the watermark on the vertex coordinates, where they increase their robustness by adding an attack layer between the encoder and decoder for adversarial training (Goodfellow, Shlens, and Szegedy 2014). All these methods watermark meshes in vertex coordinates without changing the topology of meshes.

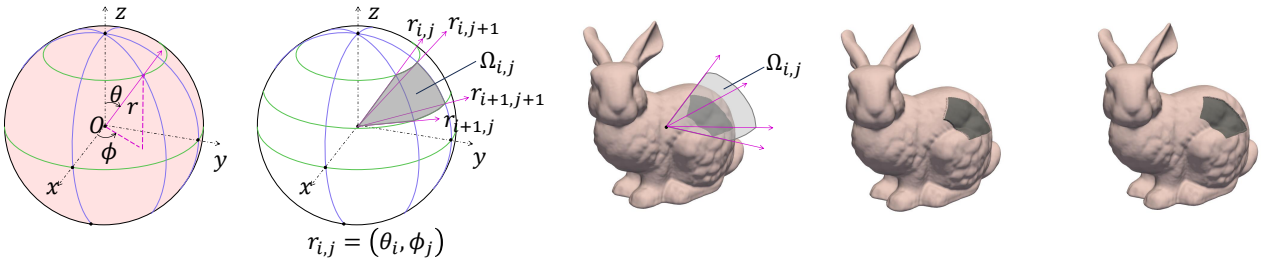
Examined post-processed methods. Following the above watermark process, we post-process watermarked mesh to remove the watermark. We first examine popular mesh processing methods such as Gauss noise, rotation, quantization, and smoothing, which are commonly used in previous works (Wang et al. 2022; Zhu et al. 2024) to evaluate the robustness of their watermark. These post-processed methods output $\mathcal{M}'_w$ sharing the same topology with $\mathcal{M}_w$. Our FUNCEVADE and remeshing (Botsch and Kobbelt 2004; Botsch et al. 2010) will output a $\mathcal{M}'_w$ with different topology with $\mathcal{M}_w$. In detail, remeshing incrementally performs simple operations such as edge splits, edge collapses, edge flips, and Laplacian smoothing. All the vertices of the remeshed patch are reprojected to the original surface to keep an approximation of the input. We set the target edge length of remeshing to be half of the average edge length of the input mesh. For other removal methods, we use Gauss noise with mean $\mu = 0$ and standard deviation $\sigma = 0.005$, rotation with rotation axis $(1, 0, 0)$ and rotation angle $\alpha < \frac{\pi}{3}$, quantization with bits $N_b = 6$, laplacian smoothing with $\lambda = 2$ and smoothing iteration 10.

Question 1. How is the quality of the mesh post-processed by FUNCEVADE?

Table 1 uses the previously mentioned removal parameters mentioned. Table 1 shows quantitative results of evasion rate and mesh similarity between $\mathcal{M}_w$ and $\mathcal{M}'_w$. We can observe that (1) Gauss noise, smoothing and quantization make visible distortions to $\mathcal{M}_w$ but cannot evade DEEP3DMARK and WANG2022; (2) meshes post-processed by FUNCEVADE achieves 100% evasion rate on all watermarking methods while keeping the lowest Hausdorff distance between $\mathcal{M}_w$ and $\mathcal{M}'_w$ (thus highest similarity).

Question 2. Why are previous watermarking methods vulnerable to function space removal?

The critical vulnerability of all previous methods is that their watermark detection is conducted on discretized space. Concretely, a mesh can be viewed as a discretization of a continuous surface. A continuous surface can be discretized in multiple ways. For example, meshes can be extracted from an SDF at different resolutions, resulting in various numbers of vertices and other topologies. However, previous watermarking methods are either vulnerable to topology changes (Wang et al. 2022; Zhu et al. 2024; Al-Khafaji and Abhayaratne 2019; Peng, Long, and Long 2021) or vulnerable to vertex number and vertex range changes (Peng,



(a) Spherical Coordinate (b) Partition $\Omega_{i,j}$ (c) Surface within $\Omega_{i,j}$ (d) $\Omega_{i,j}$ with $b = 1$ (e) $\Omega_{i,j}$ with $b = 0$

Figure 2: FUNCMARK overview. We convert vertex coordinates into (a) spherical coordinate system, which is further divided into $N_s * N_s$ partitions (denoted as $\Omega_{i,j}$). We further embed one bit in each (b) partition $\Omega_{i,j}$. The surface within $\Omega_{i,j}$ is deformed (c) outward if the embedded bit $b = 1$, or else deformed (d) inward. The deformation strength is $10\times$ of the default setting.

Liao, and Long 2022). For SPECTRAL, topology changes will result in different Laplacian matrices and thus affect the decomposed eigenvalue where the watermark is hidden. For WANG2022 and DEEP3DMARK, both methods rely on graph neural networks, which are sensitive to topology changes. For PENG2021 and PENG2022, they transform vertex coordinates into a new coordinate system before embedding or decoding the watermark. The watermark cannot be decoded if the coordinate system changes. PENG2021 builds the coordinate system based on the largest face of the input mesh, which is fragile to topology changes. PENG2022 builds the coordinate system based on the bounding sphere and the farthest vertex pair, which is fragile to vertex number and range changes.

If watermark embedding and decoding in discrete space is vulnerable to function space removal, *can we conduct watermark embedding and decoding in function space?*

5 Function Space Watermarking

The above analysis finds that current mesh watermarking methods are vulnerable to function space attacks. We address this issue by proposing FUNCMARK, which watermarks SDF (mesh in function space) such that the watermark can be verified on the given watermarked SDF or meshes extracted from it. The key idea of FUNCMARK is to make local deformations of SDF based on the binary message $w \in \{0,1\}^n$. As Fig. 2 shows, (1) we first divide 3D space and SDF into $N_s * N_s$ partitions under the spherical coordinate system (Fig. 2(a,b)); (2) within each partition $\Omega_{i,j}$ we embed one bit $w_k \in 0, 1$; (3) the surface within $\Omega_{i,j}$ is either deformed outward if $w_k = 1$, or else inward. We compare the effectiveness with previous watermarking methods and evaluate the robustness of FUNCMARK against FUNCEVADE and remeshing.

Watermark Objective. Given a mesh $\mathcal{M}$, watermark aims to get a watermarked mesh $\mathcal{M}_w$ such that $\mathcal{M}_w$ are highly similar to the original mesh $\mathcal{M}$ and the watermark can withstand arbitrary removal attacks. We evaluate robustness by **evasion rate** and mesh similarity by **Hausdorff distance**. Unlike the removal problem, we expect lower Hausdorff distance and lower evasion rate in the watermarking problem because a lower evasion rate means higher robustness against watermark removal attacks.

5.1 Spherical Partitioning

Our first step is to divide 3D space under a spherical coordinate system since we only embed one bit in each partition. A cartesian coordinate (x, y, z) and spherical coordinate (r, θ, ϕ) can be converted to each other by the following:

$$\begin{cases} r = \sqrt{x^2 + y^2 + z^2} \\ \theta = \arccos(\frac{z}{r}) \\ \phi = \arctan(\frac{y}{x}) \end{cases}, \quad \begin{cases} x = r\sin(\theta)\cos(\phi) \\ y = r\sin(\theta)\sin(\phi) \\ z = r\cos(\theta) \end{cases}. \tag{5}$$

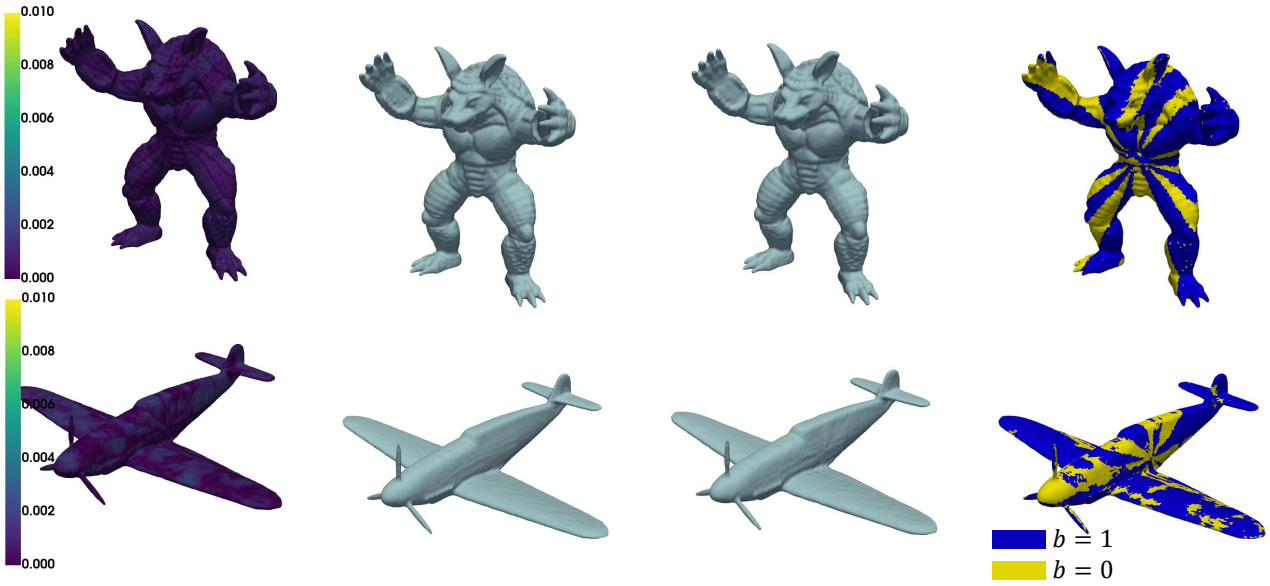
We equally divide θ and ϕ into N_s partitions. In this case, the 3D space is divided into $N_s * N_s$ partitions. As Fig. 2(b) shows, a spherical partition $\mathbf{\Omega}_{i,j}$ is a local region $\{(r, \theta, \phi) | \theta \in [\theta_i, \theta_{i+1}] \cap \phi \in [\phi_j, \phi_{j+1}]]\}$.

5.2 Watermark Embedding

Our watermark is embedded by deforming the SDF F_Θ based on the embedded bit w_k to get a deformed SDF G_Θ. Suppose the partition $\Omega_{i,j}$ is embedded with bit w_k, the surface within partition $\Omega_{i,j}$ will be deformed outward if $w_k = 1$, or else inward if $w_k = 0$. We define deformation function $\mathbf{x} = D(\mathbf{y})$, which moves the original $\mathbf{y}$ to target point $\mathbf{x}$ as follows:

$$D(\mathbf{y}) = \begin{cases} \mathbf{y} + \delta \nabla F_\Theta(\mathbf{y}) & w_k = 1 \\ \mathbf{y} - \delta \nabla F_\Theta(\mathbf{y}) & w_k = 0 \end{cases}, \tag{6}$$

where δ is a constant value representing the deformation strength. The deformed SDF G_Θ can be represented as $G_\Theta(\mathbf{x}) = F_\Theta(D^{-1}(\mathbf{x}))$ (Yang et al. 2021; Deng, Yang, and Tong 2021; Liu et al. 2021). One challenge is to make $D(\mathbf{y})$ invertible. One way to make D invertible is to approximate D with invertible residual blocks (Yang et al. 2021). However, in practice, we find the accuracy of invertible residual blocks cannot satisfy the accuracy requirements of watermarking. Fortunately, our $D(\mathbf{y})$ is explicitly defined, which makes it possible for us to derive $D^{-1}(\mathbf{x})$ via Newton's method.



(a) Shortest distance to original mesh for each vertex (b) The original mesh (c) Mesh extracted from G_Θ (d) Visualize vertex tags b

Figure 3: (a-c) Geometric differences between meshes extracted from F_Θ and G_Θ. (d) Vertex tags b on meshes from G_Θ.

Newton's method calculates $D^{-1}(\mathbf{x})$. We derive $\mathbf{y} = D^{-1}(\mathbf{x})$ by calcuating the zero point of $\mathbf{y} \pm \delta \nabla F_\Theta(\mathbf{y}) - \mathbf{x} = 0$. We apply Newton's method (Ypma 1995) to calculate the zero point as the following repeated process:

$$\mathbf{y}_{n+1} = \mathbf{y}_n - \mathcal{J}_D^{-1}(\mathbf{y}_n) \cdot D(\mathbf{y}_n), \tag{7}$$

where $\mathbf{y}_i$ is the guessed zero point at i-th iteration, $\cdot$ is the matrix multiplication operation, and $\mathcal{J}_D^{-1}(\mathbf{y})$ is the inverse of Jacobian matrix of $D(\mathbf{y})$. To calculate a correct zero point, this process is sensitive to the accuracy of $\mathcal{J}_D(\mathbf{y})$, which is derived from Hessian of the SDF: $\mathcal{H}_{F_\Theta}(\mathbf{y})$. However, $\nabla F_\Theta(\mathbf{y})$ is supervised with ground truth value only when $\mathbf{y}$ is on zero-isosurface, which means $\nabla F_\Theta(\mathbf{y})$ may not be accurate when $\mathbf{y}$ is far from surface. Thus, Newton's method may not find a zero point if $\mathbf{y}_0$ is initialized far from the surface. To address this issue, we randomly initialize multiple $\mathbf{y}_0$ within $[-1, 1]^3$. One hundred samples are enough to find the ground truth zero point.

5.3 Watermark Decoding

We decode each bit w'_k independently within each partition. A partition $\Omega_{i,j}$ is tagged with bit b if more than half of the points in that region are tagged with bit b. A point $\mathbf{x}$ is tagged with $b = 1$ if $F_\Theta(\mathbf{x}) > 0$, otherwise we tag it with bit 0. We can decode watermarks on both G_Θ and meshes extracted from G_Θ. Given a mesh extracted from G_Θ, we directly tag its vertex set with the binary bit. Given a watermarked SDF G_Θ, we can either extract a mesh from it through isosurfacing or sampling points on its zero-isosurface.

Sampling strategy on SDF. For zero-isosurface sampling, we adopt a sample-and-reject strategy. In detail, we sample points within $[-1, 1]^3$ and reject points whose distance to the surface is greater than 10^{-4}. We then run gradient descent to approximate on-surface points: $\mathbf{x}_{t+1} = \mathbf{x}_t - F_\Theta(\mathbf{x}_t)\nabla F_\Theta(\mathbf{x}_t)$.

5.4 Watermark Evaluation

Setup. Since we embed each spherical partition with one binary bit in the secret message w, if the number of partitions $N_s * N_s$ is greater than the length of the message, we repeatedly embed message bits until we consume all partitions. Unless explicitly mentioned, our experiment is conducted under the following settings. We set $N_s = 32$ (*i.e.*, the spherical system is divided into $32 * 32$ partitions). We set message length $n = 48$, and the detection threshold $\tau = 31$. We use marching cube (Lorensen and Cline 1998) as our default isosurfacing method and the default watermarking strength $\delta = 0.001$. We use the same dataset, evaluation metrics, and watermarking methods (as baselines) in Sec. 4.2 to evaluate watermark accuracy before any post-processing method is applied. We use the same set of attack settings for all post-processing methods in Sec. 4.2 to compare evasion rates between baselines and FUNCMARK.

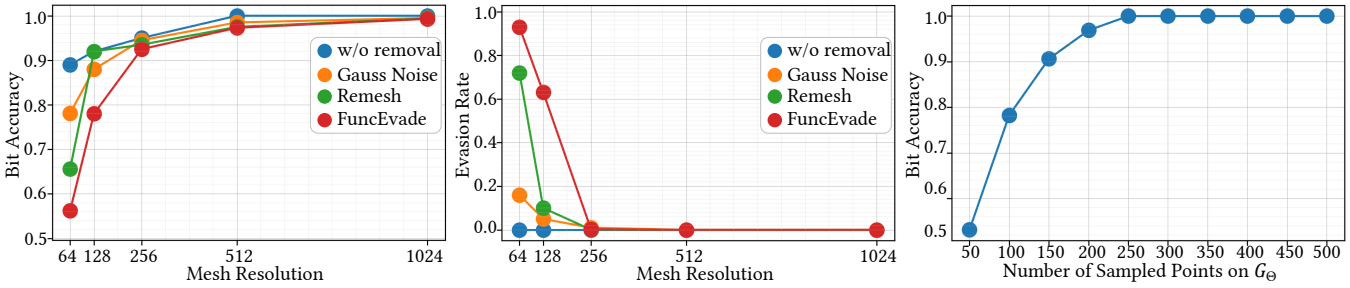
Question 1. How is the watermarked mesh quality?

We present visualizations of the shortest distance from vertices of the watermarked mesh to vertices of the original mesh (Figure 3(a)), the original mesh itself (Figure 3(b)), the watermarked mesh extracted from the watermarked SDF G_Θ (Figure 3(c)), and vertex tag b (Figure 3(d)). Visualizing vertex tags helps better understand the process of embedding the secret message w. Notably, it can be observed that the watermarked mesh extracted from the watermarked signed distance function G_Θ appears perceptually identical to the original mesh.

Question 2. How is the mesh quality, bit accuracy, and evasion rate compared with other watermarking methods?

Methods	HD↓	Accuracy↑	Evasion Rate↓				
			Gaussian	Rotation	Translation	Remesh	FUNCEVADE (Ours)
SPECTRAL	0.012	87.48%	100%	100%	100%	100%	100%
PENG2021	0.015	97.98%	100%	100%	100%	100%	100%
PENG2022	0.037	80.89%	100%	**0%**	**0%**	100%	100%
WANG2022	0.185	97.95%	0.2%	18.4%	**0%**	100%	100%
DEEP3DMARK	0.061	**98.17%**	**0.1%**	15.2%	**0%**	100%	100%
FUNCMARK (Ours)	**0.004**	95.69%	1.6%	14.8%	**0%**	**0.1%**	**0.3%**

Table 2: Watermark results. HD indicates Hausdorff distance, which is evaluated between the original and watermarked mesh. Accuracy is evaluated before the watermarked mesh is post-processed by removal methods. Lower HD means higher watermarked mesh quality, and a lower evasion rate means higher robustness.



(a) Bit accuracy with varied mesh resolution (b) Evasion rate with varied mesh resolution (c) Bit accuracy with varied sampled points

Figure 4: Our watermark can be verified on SDF G_Θ and extracted meshes. For watermark detection on extracted meshes, we evaluate (a) bit accuracy with varied mesh resolution and (b) evasion rate with varied mesh resolution. For watermark verification on SDF, we evaluate bit accuracy with a varied number of sampled points on G_Θ.

Mesh quality is evaluated by Hausdorff distance (HD), where lower HD indicates a higher similarity between the original and the watermarked mesh. Before the removal attack, we evaluate the bit accuracy of watermarking methods to prove they are all effective without attack. Then, we evaluate their robustness against attacks by evasion rate. Table 2 shows the performance results between FUNCMARK and previous watermarking methods. We can observe that (1) FUNCMARK achieves the lowest HD, which means we have the highest mesh quality. (2) Remeshing and FUNCEVADE have 100% evasion rate on other watermarking methods while having only 0.1% and 0.3% evasion rate on FUNCMARK, which means FUNCMARK is the only method robust against remeshing and FUNCEVADE.

Question 3. Since FUNCMARK is a function space watermarking method, can watermark be verified on meshes extracted at arbitrary resolutions?

The watermark of FUNCMARK can be verified on both SDF G_Θ and meshes extracted from G_Θ at arbitrary resolutions. For watermark decoding on a mesh, where we can directly tag $0/1$ on the mesh vertices, we evaluate the bit accuracy and evasion rate on meshes extracted at arbitrary resolution. For watermark verification on SDF G_Θ, where we have to sample points on the zero-isosurface of SDF such that we can tag $0/1$ on the sampled points, we evaluate the bit accuracy on varied sampled points. Figure 4 shows the bit accuracy and evasion rate with varied mesh resolution and sampled points. We can observe that (1) all the evasion rates drop to zero on meshes extracted at resolution 256, and meshes extracted at a resolution lower than 256 have bad quality, which means FUNCMARK mitigates all the removal methods because the attacker cannot evade watermark detection without damaging the resultant mesh quality. (2) Given a watermarked *sdf* G_Θ, our watermark can be verified with only 250 sampled points.

6 Conclusion

In this paper, we have noticed the research gap where no previous work investigated the mesh watermark robustness against function space removal. We have bridged this gap in an attack-then-defense manner by proposing FUNCEVADE and FUNCMARK. FUNCEVADE is a function space removal method that generates a different discrete representation of the watermarked mesh by extracting it from the signed distance function of the watermarked mesh. We have observed that FUNCEVADE evades all previous mesh watermarking methods. FUNCMARK is a function space mesh watermarking method which successfully mitigates FUNCEVADE by watermarking signed distance function. By doing so, the watermark can be verified on arbitrary mesh extracted from the watermarked signed distance function. Our experiments have shown that FUNCMARK mitigates FUNCEVADE by dropping its evasion rate from 100% to 0.3% while keeping similar performance on other metrics compared with current SOTA methods.

Acknowledgments

This work was supported in part by National Key R&D Program of China under Grant 2021YFF0900300, in part by Guangdong Key Program under Grant 2021QN02X166, and in part by Research Institute of Trustworthy Autonomous Systems under Grant C211153201. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

References

Al-Khafaji, H.; and Abhayaratne, C. 2019. Graph spectral domain blind watermarking. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2492–2496. IEEE.

Atzmon, M.; Haim, N.; Yariv, L.; Israelov, O.; Maron, H.; and Lipman, Y. 2019. Controlling neural level sets. *Advances in Neural Information Processing Systems*, 32.

Atzmon, M.; and Lipman, Y. 2020. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2565–2574.

Botsch, M.; and Kobbelt, L. 2004. A remeshing approach to multiresolution modeling. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 185–192.

Botsch, M.; Kobbelt, L.; Pauly, M.; Alliez, P.; and Lévy, B. 2010. *Polygon mesh processing*. CRC press.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; Xiao, J.; Yi, L.; and Yu, F. 2015. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Chen, Z.; and Zhang, H. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5939–5948.

Deng, Y.; Yang, J.; and Tong, X. 2021. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10286–10296.

Dupont, E.; Kim, H.; Eslami, S.; Rezende, D.; and Rosenbaum, D. 2022. From data to functa: Your data point is a function and you can treat it like one. *arXiv preprint arXiv:2201.12204*.

Fernandez, P.; Couairon, G.; Jégou, H.; Douze, M.; and Furon, T. 2023. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22466–22477.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Hou, G.; Ou, B.; Long, M.; and Peng, F. 2023. Separable Reversible Data Hiding for Encrypted 3D Mesh Models Based on Octree Subdivision and Multi-MSB Prediction. *IEEE Transactions on Multimedia*.

Hou, J.-U.; Kim, D.-G.; and Lee, H.-K. 2017. Blind 3D mesh watermarking for 3D printed model by analyzing layering artifact. *IEEE Transactions on Information Forensics and Security*, 12(11): 2712–2725.

Jiang, R.; Zhou, H.; Zhang, W.; and Yu, N. 2017. Reversible data hiding in encrypted three-dimensional mesh models. *IEEE Transactions on Multimedia*, 20(1): 55–67.

Jiang, Z.; Zhang, J.; and Gong, N. Z. 2023. Evading watermark based detection of AI-generated content. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 1168–1181.

Laboratory, S. U. C. G. 2023. The Stanford 3D Scanning Repository.

Liu, S.; Zhang, X.; Zhang, Z.; Zhang, R.; Zhu, J.-Y.; and Russell, B. 2021. Editing conditional radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5773–5783.

Lorensen, W. E.; and Cline, H. E. 1998. Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, 347–353.

Luo, Z.; Guo, Q.; Cheung, K. C.; See, S.; and Wan, R. 2023. Copyrnerf: Protecting the copyright of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22401–22411.

Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; and Geiger, A. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4460–4470.

Michalkiewicz, M.; Pontes, J. K.; Jack, D.; Baktashmotlagh, M.; and Eriksson, A. 2019. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4743–4752.

Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 165–174.

Pavllo, D.; Spinks, G.; Hofmann, T.; Moens, M.-F.; and Lucchi, A. 2020. Convolutional generation of textured 3d meshes. *Advances in Neural Information Processing Systems*, 33: 870–882.

Peng, F.; Liao, T.; and Long, M. 2022. A semi-fragile reversible watermarking for authenticating 3d models in dual domains based on variable direction double modulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12): 8394–8408.

Peng, F.; Long, B.; and Long, M. 2021. A general region nesting-based semi-fragile reversible watermarking for authenticating 3D mesh models. *IEEE transactions on circuits and systems for video technology*, 31(11): 4538–4553.

Praun, E.; Hoppe, H.; and Finkelstein, A. 1999. Robust mesh watermarking. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 49–56.

Siddiqui, Y.; Alliegro, A.; Artemov, A.; Tommasi, T.; Sirigatti, D.; Rosov, V.; Dai, A.; and Nießner, M. 2023. Meshgpt: Generating triangle meshes with decoder-only transformers. *arXiv preprint arXiv:2311.15475*.

Sitzmann, V.; Martel, J.; Bergman, A.; Lindell, D.; and Wetzstein, G. 2020. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33: 7462–7473.

Tsai, Y.-Y. 2020. Separable reversible data hiding for encrypted three-dimensional models based on spatial subdivision and space encoding. *IEEE transactions on multimedia*, 23: 2286–2296.

Tsai, Y.-Y.; and Liu, H.-L. 2022. Integrating Coordinate Transformation and Random Sampling Into High-Capacity Reversible Data Hiding in Encrypted Polygonal Models. *IEEE Transactions on Dependable and Secure Computing*.

Wang, F.; Zhou, H.; Fang, H.; Zhang, W.; and Yu, N. 2022. Deep 3D mesh watermarking with self-adaptive robustness. *Cybersecurity*, 5(1): 1–14.

Yang, G.; Belongie, S.; Hariharan, B.; and Koltun, V. 2021. Geometry processing with neural fields. *Advances in Neural Information Processing Systems*, 34: 22483–22497.

Ypma, T. J. 1995. Historical development of the Newton–Raphson method. *SIAM review*, 37(4): 531–551.

Zhu, X.; Ye, G.; Luo, X.; and Wei, X. 2024. Rethinking Mesh Watermark: Towards Highly Robust and Adaptable Deep 3D Mesh Watermarking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 7784–7792.