Annual of Konstantin Preslavsky University of Shumen
Faculty of Mathematics and Informatics
XXVI C, 2025, pp. 41 – 47, https://doi.org/10.46687/DYNY4923

3D MESH WATERMARKING USING RADIAL VERTEX EMBEDDING[*]

DELYAN H. SARMOV

ABSTRACT: *This paper presents an algorithm for embedding digital watermarks into 3D mesh models. The method relies on radial displacement of vertices from the object's centroid and is evaluated for robustness against geometric transformations including rotation, scaling, and additive noise. The approach uses multiple radial layers with majority voting for error correction. Experimental results demonstrate high baseline extraction accuracy and robustness against moderate rotations and noise, while revealing sensitivity to scaling and rotations beyond certain thresholds.*

KEYWORDS: *3D watermarking, 3D mesh, Robust Watermarking, copyright protection*

2020 Math. Subject Classification: *65D17, 68U99, 93B51*

ВОДЕН ЗНАК ЗА 3D МОДЕЛ С РАДИАЛНО ВГРАЖДАНЕ ВЪВ ВЪРХОВЕ[†]

ДЕЛЯН Х. СЪРМОВ

1 Introduction

Digital watermarking has become a crucial technique for copyright protection, fingerprinting, and authentication of 3D content.

[*]This paper is partially supported by Scientific Research Grant РД-08-106/05.02.2025 of Konstantin Preslavsky University of Shumen

[†]Статията е частично финансирана по проект № РД-08-106/05.02.2025 г. на ШУ „Еп. Константин Преславски“

Unlike images or video, 3D meshes represent complex geometric structures that can undergo a wide range of transformations, making watermarking a challenging task.

Prior approaches often embed information in vertex positions, normals, or texture channels. However, normals can be recomputed or removed, and texture information may be absent or altered during processing. To address these challenges, this work proposes embedding watermark bits along radial displacement vectors from the mesh centroid. By doing so, the method leverages geometric consistency of the mesh to maintain robustness under transformations such as rotation, scaling, and small perturbations.

In this work, a multilayer radial embedding strategy is presented for distributing watermark bits into vertices. A strategy and algorithm for embedding and extraction by radial vectors are formulated. A robustness analysis against standard transformations with experimental validation is performed.

2 Related Work

Watermarking techniques on 3D meshes have been extensively studied. Different watermarking methods have been developed and distinguish fragile from robust techniques [1]. After embedding, their robustness to geometric distortions is compared. [2]. Other works investigate radial coordinate embedding strategies [3] and domain transformation techniques (wavelet, spectral) [4,5]. In comparison, this method focuses on direct radial embedding with shift with multiple radial layers and majority vote extraction, aiming for simplicity and robustness.

3 Watermarking Method

Normalization and radial vectors

A 3D mesh is represented as M = (V, F), where

(1) $$\boldsymbol{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \dots, \boldsymbol{v_N}\}$$

are the vertices and F are the faces.

The geometric centroid is

(2) $$c = \frac{1}{N}\sum_{i=1}^{N} v_i$$

For each vertex v_i, define the radial direction vector

(3) $$d_i = \frac{v_i - c}{||v_i - c||}$$

Normalization ensures ‖d‖ = i, for all i. By embedding along these vectors, the displacement is invariant to uniform scaling to first order.

Vertex Selection and Layering

First compute per-vertex distance

(4) $$dist_i = ||v_i - c||$$

and participation count (number of faces p_i incident to vertex i). Then define weights:

(5) $$w_i^{far} = dist_i \,.\, p_i$$

(6) $$w_i^{close} = \left(maxdist_j - dist_i\right) * p_i$$

(7) $$w_i^{med} = \left|dist_i - median_j\left(dist_j\right)\right| * p_i$$

For each of the three layers: far–layer, close–layer, median–layer, select the top-vertices with largest corresponding weight. This distributes embedding across vertices of varying radial positions and participation, improving robustness.

Embedding Procedure

Let the watermark bit-sequence be

(8) $$W = [w_1, w_2, \dots, w_L],\ w_k \in \{0, 1\}$$

For a selected vertex v_i (associated with bit index k), define

(9) $$s_i = \begin{cases} +1 & if\ w_k = 1 \\ -1 & if\ w_k = 0 \end{cases}$$

The new embedded vertexes are calculated in the following way

(10) $$v_i' = v_i + \alpha \,.\, s_i * d_i$$

where is the embedding strength α. Small enough to ensure imperceptibility, but large enough to tolerate noise.

Extraction Procedure

Given a possibly transformed mesh with vertices $v_i^{(T)}$, it is assumed that the original coordinates (or displacements) accessible.

It is calculated d_i then

(11) $$\Delta \boldsymbol{v_i} = \boldsymbol{v_i^{(T)}} - \boldsymbol{v_i}$$

and project onto d_i

(12) $$\boldsymbol{p_i} = \Delta \boldsymbol{v_i} \, . \boldsymbol{d_i}$$

Extracted bit is

(13) $$\boldsymbol{w_i'} = \begin{cases} \boldsymbol{1} & \boldsymbol{if\ \ p_i > 0} \\ \boldsymbol{0} & \boldsymbol{otherwise} \end{cases}$$

Majority Voting Logic

It is embed three independent layers l=1,2,3. For each bit position i, define bits from each layer $w_i'^{(1)}$, $w_i'^{(2)}$, $w_i'^{(3)}$. Then define final extracted bit.

(14) $$\boldsymbol{w_i^{final}} = \boldsymbol{mode}\left\{\boldsymbol{w_i'^{(1)}}, \boldsymbol{w_i'^{(2)}}, \boldsymbol{w_i'^{(3)}}\right\}$$

This shows how majority voting improves reliability.

Algorithm Pseudocode

Embedding

Input: Mesh V={v_i}, Faces F, Watermark W, Strength α, TopN

Compute centroid c = (1/N) ∑ v_i

Compute participation p_i for each vertex

Compute distances dist_i = ||v_i – c||

For each layer L∈{far, close, med}:

Compute weights w_i^L

Select top-TopN indices idx^L

For each layer L:

For each j,k such that v_idx^L[j] corresponds to bit W[k]:

Compute d = (v_idx^L[j] – c) / ‖v_idx^L[j] – c‖

s = +1 if W[k]==1 else –1

v_idx^L[j] ← v_idx^L[j] + α * s * d

Return watermarked mesh V′

Extraction

Input: Original mesh V, Watermarked mesh V′, layers indices {idx^L}

For each layer L:

For each i in idx^L:

Compute d = (v_i – c) / ‖v_i – c‖

Δv = v′_i – v_i

If Δv · d > 0 then bit=1 else bit=0

Extract bits W_hat^(L)

W_hat_final = majority_vote(W_hat^(1),W_hat^(2),W_hat^(3))

Return W_hat_final

4 Experimental Setup and Results

A python program implementing the above algorithms has been developed. The program has been tested with a standard set of models used for validation of results in similar studies.

Dataset: A standard 3D meshes (Stanford Bunny 2503 verts, 4968 faces; armadillo 49990 verts, 99976 faces; Dragon 1180060 verts, 2349078 faces).

Watermark: 64-bit random binary sequence.

Embedding strength: $\alpha = 0.001$.

Vertices per layer (TopN): 256.

Evaluated transformations:
Rotations: 5°, 10°, 20°, 45° (around random axis).
Scaling factors: 0.90, 0.95, 1.05, 1.10.
Gaussian noise: σ in [α/3, α] (three steps).

Table 1. Results Summary

Transformation	Parameter	Accuracy Stanford Bunny	Accuracy armadillo	Accuracy Dragon
Baseline	-	1.000	1.000	1.000
Rotation	5°	0.969	0.906	0.875
Rotation	10°	0.531	0.750	0.750
Rotation	20°	0.500	0.500	0.531
Rotation	45°	0.500	0.500	0.500
Scaling	0.90	0.500	0.500	0.500
Scaling	0.95	0.500	0.500	0.531
Scaling	1.05	0.500	0.500	0.500
Scaling	1.10	0.500	0.500	0.500
Noise	0.000333	0.969	1.000	1.000
Noise	0.000667	0.938	0.969	1.000
Noise	0.001000	0.969	0.906	0.844

5 Conclusion and Future Work

Results showing baseline perfect extraction confirms embedding correctness. High resilience to small rotations (≤5°) and minor noise perturbations bellow α. Accuracy degrades under larger rotation (10°+) and more so under scaling beyond ±5%. Majority voting across three layers mitigates errors when single-layer reliability drops.

Limitations remain in handling larger scaling transformations and topological changes such as mesh simplification. Future work could explore inter-vertex distance embeddings, local coordinate frames, and graph-based or deep-learning methods for higher resilience [6].

REFERENCES:

[1] Wang K., Lavoué G., Denis F., Baskurt A. A Comprehensive Survey on Three-Dimensional Mesh Watermarking, IEEE Trans. Multimedia, 2008.

[2] Medimegh N., Belaid S., Werghi N. A Survey of the 3D Triangular Mesh Watermarking Techniques. Int. J. Multimedia, 2015.

[3] Liu J., Niu X. “A novel watermarking algorithm for three-dimensional point clouds.” SAGE Journals, 2019.

[4] Gul E. “3D watermarking techniques.” ScienceDirect, 2025.

[5] Vasic B., Vasic B. Simplification Resilient LDPC-Coded Sparse-QIM Watermarking for 3D-Meshes, 2012.

[6] Narendrra M. et al. Deep 3D mesh watermarking with self-adaptive robustness, 2022.

Delyan Hristov Sarmov
Konstantin Preslavsky University of Shumen
e-mail: d.sarmov@shu.bg