Adaptive 3D Mesh Steganography Based on Feature-Preserving Distortion

Yushu Zhang, Senior Member, IEEE, Jiahao Zhu, Senior Member, IEEE, Mingfu Xue, Senior Member, IEEE, Xinpeng Zhang, Member, IEEE, and Xiaochun Cao, Senior Member, IEEE

Abstract—Current 3D mesh steganography algorithms relying on geometric modification are prone to detection by steganalyzers. In traditional steganography, adaptive steganography has proven to be an efficient means of enhancing steganography security. Taking inspiration from this, we propose a highly adaptive embedding algorithm, guided by the principle of minimizing a carefully crafted distortion through efficient steganography codes. Specifically, we tailor a payload-limited embedding optimization problem for 3D settings and devise a feature-preserving distortion (FPD) to measure the impact of message embedding. The distortion takes on an additive form and is defined as a weighted difference of the effective steganalytic subfeatures utilized by the current 3D steganalyzers. With practicality in mind, we refine the distortion to enhance robustness and computational efficiency. By minimizing the FPD, our algorithm can preserve mesh features to a considerable extent, including steganalytic and geometric features, while achieving a high embedding capacity. During the practical embedding phase, we employ the Q-layered syndrome trellis code (STC). However, calculating the bit modification probability (BMP) for each layer of the Q-layered STC, given the variation of Q, can be cumbersome. To address this issue, we design a universal and automatic approach for the BMP calculation. The experimental results demonstrate that our algorithm achieves state-of-the-art performance in countering 3D steganalysis.

Index Terms—3D mesh, 3D mesh steganography, syndrome trellis code, 3D steganalysis.

I. INTRODUCTION

STEGANOGRAPHY is both an art and a science that conceals confidential messages imperceptibly in innocent-looking digital media, such as images, videos, audio, and texts [1]. In recent years, the rapid advancement of 3D technologies has led to the widespread use of 3D models in various applications, such as virtual reality and 3D printing. This trend has made 3D models suitable hosts for steganography, similar to other digital media. Currently, 3D models are typically represented in three different ways: point clouds, polygon meshes, and voxels [2]. Among these representations, meshes are preferred by 3D steganographers because they provide a greater capacity for message embedding.

In 3D mesh steganography, most algorithms rely on moving mesh vertices to embed messages. Early algorithms [3], [4], [5], [6], [7], [8] focused on enlarging the embedding capacity while minimizing mesh distortion. However, with the advent of 3D steganalyzers [9], [10], [11], [12], [13], [14], [15], these algorithms are now easily detectable. Consequently, steganographers have shifted their focus to steganographic security, i.e., anti-steganalysis ability. Li et al. [16] were the first to do so, but their algorithm can resist only a few 3D steganalyzers. Numerous studies in 2D steganography have demonstrated that incorporating adaptability into algorithms is an effective means of enhancing steganographic security. This strategy relies on the principle of minimal impact embedding [17], which involves two core procedures: minimizing a suitably-defined distortion and practical steganography coding. Since Syndrome Trellis Code (STC) [18] has become the standard methodology for practical embedding, most approaches to adaptive steganography strive to find a distortion model that performs well experimentally. Examples include the classical spatial algorithms HUGO [19], WOW [20], and S-UNIWARD [21], as well as some algorithms tailored for JPEG images [21], [22], [23]. Building on this idea, Zhou et al. [24] applied the design framework for 2D adaptive steganography [25] to 3D settings. They proposed two geometric-feature-related distortion models: vertex normal based distortion and curvature based distortion. Although their algorithm achieves satisfactory results in countering steganalysis, it involves nonadaptive embedding, namely least significant bit replacement (LSBR), which somewhat limits its potential for improving steganographic security.

From the above short review, it is clear that the distortion model plays a crucial role in achieving effective concealment
of messages. A well-designed distortion can capture specific characteristics that impede detection by steganalyzers within a chosen feature space [24]. Building on this insight, we propose a highly adaptive 3D mesh steganography method based on feature-preserving distortion (FPD). Our algorithm comprises three key steps: embedding domain construction, distortion design, and practical message embedding. For the first step, we begin with a Decimal-to-Integer (D2I) conversion for vertex coordinates, and then use the integers for constructing an embedding domain composed of bitplanes. In the second step, we propose an additive distortion model FPD that measures the impact of message embedding with the assistance of 3D steganalytic features. Specifically, we evaluate the steganalysis ability of each subfeature from the state-of-the-art statistical 3D steganalyzer, and then, we select the most effective ones for designing the FPD. Furthermore, to make our algorithm practical, we improve upon FPD from the aspect of subfeature selection and computational efficiency. By minimizing the FPD, we can obtain the optimal vertex-changing distribution, which serves as a guide for the subsequent practical embedding. In the last step, we adopt the $Q$-layered STC for bit embedding due to its near-optimal performance for arbitrary additive distortion functions [18]. However, varying $Q$ makes the calculation of the bit modification probability (BMP) in each layer of $Q$-layered STC troublesome. To address this problem, we devise a universal and automatic BMP calculation approach, dubbed U & A-BMP. The contributions of our work are fourfold:

- We present a highly adaptive 3D mesh steganography algorithm that balances security and embedding capacity. It assigns a specific cost to each vertex coordinate change, enabling it to carefully consider the direction and distance of each vertex movement. To our knowledge, this is the first 3D steganography to achieve such a high level of adaptability.
- We assess the effectiveness of steganalytic features adopted by the state-of-the-art 3D steganalyzer and create the FPD to gauge the impact of message embedding. Our experimental results indicate that FPD significantly enhances the security of our approach while retaining mesh features, such as steganalytic and geometric features, to a certain extent.
- We fine-tune the FPD design, resulting in a significant enhancement of its robustness and computational efficiency. This improvement makes our algorithm practical and feasible for real-world scenarios.
- We provide a toolbox, U & A-BMP, which offers a more convenient and flexible method to calculate the BMP for each layer of $Q$-layered STC, especially when $Q$ varies.

The remainder of the article is organized as follows. In Section II, we introduce existing 3D mesh steganography and steganalysis. In Section III, we first describe the overall framework of our algorithm and then elaborate on it from three aspects, i.e., embedding domain construction, distortion design, and message embedding and retrieval. In Section IV, we report the experimental results and discuss the limitations of our algorithm. Finally, we conclude our work in Section V and offer several future research topics.

II. RELATED WORK

A. 3D Mesh Steganography

Unlike digital watermarking, steganography was developed for covert communication. Existing 3D mesh steganography can be roughly divided into the following four categories [26].

Two-State Steganography: Cayre and Macq [3] proposed a substitutive blind 3D mesh steganographic scheme in the spatial domain, the main idea of which is to regard a triangular surface as a two-state geometrical object. During data embedding, the states of triangles are geometrically modulated in terms of the message to hide. To enlarge the embedding capacity, Wang and Cheng [4] proposed a multi-level embedding procedure (MLEP). They built a triangle neighbor table and heuristically constructed a $k$-D tree to improve efficiency. Subsequently, they proposed a modified MLEP and combined the spatial domain with representation domain for a larger embedding capacity [5]. Later, Chao et al. [6] proposed a novel multilayered embedding scheme, which increases the embedding capacity up to 21 $\sim$ 39 bits per vertex (bpv). Unlike the prior work, Itier et al. [7] provided a novel scheme that synchronizes vertices along a Hamiltonian path and displaces them with static arithmetic coding. Recently, Li et al. [8] developed an embedding strategy that allows us to adjust embedding distortion below a specified level by defining a truncated space. With the emergence of modern 3D steganalyzers, Li et al. [16] were the first to consider the anti-steganalysis ability in 3D mesh steganography, and they reinvestigated and modified the prior work [7] to make it more resistant against steganalysis.

LSB Steganography: Yang et al. [27] investigated the correlation between the spatial noise and normal noise in meshes. Additionally, they proposed a high-capacity 3D mesh steganography that calculates an appropriate quantization level for each vertex and replaces unused LSBs with bits to hide. Zhou et al. [24] analyzed the steganalytic features proposed in [11] and selected the most effective parts for their distortion model design. In addition, they combined STCs and LSBR for practical message embedding.

Permutation Steganography: Some researchers have explored alternative embedding domains to avoid the mesh distortion caused by embedding changes. Bogomjakov et al. [28] proposed a distortionless permutation steganographic scheme for polygon meshes, which hides messages by permutating the storage order of faces and vertices. Building upon the work of Bogomjakov et al. Huang et al. [29] designed a more efficient version. Moreover, they conducted a theoretical analysis that indicates that the expected embedding capacity of their algorithm is closer to optimal. Tu et al. [30] further improved upon the work [28] by making slight modifications to the mapping strategy. Later, Tu et al. [31] constructed a skewed binary coding tree rather than a complete binary tree for permutation steganography. However, the average embedding capacity of this algorithm is less than that reported in [30]. Therefore, Tu et al. [32] subsequently built a maximum expected level tree based on a novel message probability model, achieving a higher expected embedding capacity than that of all previous methods.
Transform-Domain-Based Steganography: Aspert et al. [33] utilized the watermarking technique presented in [34] and performed embedding operations by making slight displacements of vertices. Maret et al. [35] improved upon the method developed by Aspert et al. by creating a similarity-transform invariant space and by adapting the embedding procedure to the sample distribution in that space. Kaveh et al. [36] proposed 3D polygonal mesh steganography by using the surfacelet transform to achieve higher capacity and less geometric distortion.

B. 3D Mesh Steganalysis

Today, the majority of 3D steganalyzers are designed to detect steganography based on imperceptible geometric modifications. These types of steganalyzers can be implemented in three simple steps.

Mesh Smoothing and Normalization: First, the original mesh undergoes a smoothing procedure (e.g., Laplacian smoothing) to generate its reference counterpart. Subsequently, both meshes will be transformed by principal component analysis (PCA). Finally, the resulting meshes are scaled to fit within a unit cube centered at the point (0.5, 0.5, 0.5).

Calibrated Feature Extraction: The calibrated steganalytic features are extracted by computing the feature difference between the original mesh and its reference. Importantly, different steganalyzers may extract features in distinct ways. Yang and Ivriissimtzis [9] designed YANG208, the first 3D steganalytic feature set whose components are relevant to vertex coordinates, norms in Cartesian and Laplacian coordinate systems, dihedral angles of edges, and face normals. Li and Bors [10] reduced the dimension of YANG208 and considered three additional features, i.e., vertex normal, Gaussian curvature, and curvature ratio. Eventually, a 52-dimensional feature set, LFS52, was produced for steganalysis. Kim et al. [11] considered more geometric features, such as edge normals, average curvature, and total curvature, and they combined these additional features with LFS52 to create a new steganalytic feature set named LFS64. Since the features in LFS64 are not directly related to the embedding domain, it is likely that the classifier cannot achieve an ideal classification result if these features are directly input. Hence, they add a homogenous kernel map before the training step. The concept of utilizing the 3D spherical coordinate system for steganalytic feature extraction was initially suggested in LFS76 [12], followed by the inclusion of further edge-related geometric features as supplementary features in ELFS124 [13].

In contrast to the steganalysis algorithms outlined earlier, Zhou et al. [14] posited that message embedding can disrupt the interdependencies between neighboring faces. Based on this, they introduced neighborhood-level representation-guided normal voting tensor (NVT) features for 3D mesh steganalysis and integrated them with LFS64 to create a novel feature set named NVT+. Regarding mesh smoothing and normalization, Li and Bors [37] explored how the smoothing coefficient and iteration number of Laplacian smoothing impact steganalysis performance. More recently, they developed a wavelet-based steganalytic feature set, WFS228, for detecting stego meshes [15], which significantly enhanced their ability to detect meshes generated by wavelet-based watermarking algorithms.

Classification: The extracted steganalytic features are finally fed into a binary classifier. Currently, the Fisher linear discriminant (FLD) ensemble [38] has been widely adopted in contemporary 3D steganalyzers due to its fast training speed and highly reliable detection accuracy.

Aside from the steganalysis approaches described above, there are two distinct steganalysis algorithms created specifically for detecting PCA-based steganography [24] and permutation steganography [39].

III. PROPOSED ALGORITHM

A. Overall Framework

Our 3D mesh steganography follows the principle of minimal embedding impact [17] and builds upon the adaptive framework proposed in [25]. The message embedding pipeline is shown in Fig. 1. Specifically, given a mesh, we first perform the D2I conversion on the vertex coordinates for the embedding domain construction (see Section III-C). Second, we design the FPD with features selected from the state-of-the-art 3D steganalytic feature set NVT+ [14]. The specific form of the FPD and its advanced version are described in Sections III-D and III-E, respectively. To obtain the optimal vertex-changing distribution, we formulate a payload limited sender (PLS) problem tailored for 3D mesh steganography (see Section III-B) with the FPD. By using the optimal distribution obtained, we employ the Q-layered STC to embed the message bits into the cover bitplanes (see Section III-F), producing the stego bitplanes. Subsequently, the stego bitplanes undergo a Integer-to-Decimal (ID2) conversion and are combined with the topological information of the cover mesh to produce the stego mesh. The meshes used in this work are all composed of triangular faces unless otherwise stated.

B. Construction of the PLS Problem

Let $M = \{V, E, F\}$ be a cover mesh with vertex set $V = \{\mathbf{v}_i\}_{i=1}^{|V|}$, where $\mathbf{v}_i = [v_{ix}, v_{iy}, v_{iz}]^\top$, edge set $E = \{\mathbf{e}_j\}_{j=1}^{|E|}$, and face set $F = \{\mathbf{f}_k\}_{k=1}^{|F|}$. The impact of message embedding is measured by a distortion function $D$. For the sake of security, the sender should embed messages while minimizing $D$. In this work, $D$ is limited to the following additive form,

$$D(M, M') = \sum_{j \in \{x,y,z\}} D_j(M, M') = \sum_{j \in \{x,y,z\}} \sum_{i=1}^{|V|} \rho(\delta_{ij}).$$

Here $M' = \{V', E', F'\}$ is the stego counterpart of $M$, where $V' = \{\mathbf{v}'_i\}_{i=1}^{|V'|}$ and $\mathbf{v}'_i = [v'_{ix}, v'_{iy}, v'_{iz}]^\top$; and $\delta_{ij} = v_{ij}' - v_{ij}$ denotes a change quantity; $\rho : \mathbb{R} \to [0, \infty)$ defines a cost function that measures the cost of $\delta_{ij}$ occurring. Equation (1) indicates that $\rho(\delta_{ij})$ is independent of changes made at other vertices and coordinate channels, implying that the embedding changes do not interact: Equation (1) also shows the preliminary form
of the FPD, and obviously, the effectiveness for our algorithm relies on the design of \( \rho \).

As mentioned previously, the embedding changes are noninteracting. Therefore, the most straightforward approach to perform message embedding is to decompose the entire process into three subtasks according to the number of coordinate channels. For each channel, we hide a fixed average payload of \( m_j \) bits, where \( j \in \{x, y, z\} \). Considering that \( M \) is predetermined and our algorithm does not alter its topology (implying \( E = E' \) and \( F = F' \)), we can abbreviate \( D(M, M') \) as \( D(V') \). Consequently, the PLS problem for each channel can be defined as

\[
\begin{align*}
\text{minimize} \quad & E_p[D_j(V')] \\
\text{s.t.} \quad & H(p) = m_j,
\end{align*}
\]

(2)

where \( p \) denotes the vertex-changing distribution that we want to calculate and \( H \) is the information entropy function. The specific form and solution method of the PLS problem (2) are discussed in the next section.

### C. Construction of Embedding Domain

Most of the meshes used in prior works are sourced from public 3D datasets, and their vertex coordinates are typically represented in decimal form, such as \( v_i = [0.172345, 0.267324, 0.563521]^T \), with each coordinate value reserving a limited number of significant digits. As a result, it is always possible to find the minimum natural number \( k_{ij} \) that satisfies \( v_{ij} \times 10^{k_{ij}} \in \mathbb{Z} \). We then define \( k_d = \max(k_{ij}) \), and the D2I conversion shown in Fig. 1 can be achieved by \( \hat{v}_{ij} = v_{ij} \times 10^{k_d} \).

Likewise, we can easily determine the minimum positive integer \( h_b \) satisfying \( 0 \leq \hat{v}_{ij} \leq 2^{h_b} - 1 \) for any \( v_{ij} \). By letting \( b^{(l)}_{ij} \) be the \( l \)-th least significant bit of \( \hat{v}_{ij} \), we can construct a bitplane \( B^{(l)}_j = \{b^{(l)}_{ij} | v_i \in \mathbb{Z} \} \). Then the embedding domain can be defined as \( \{B^{(l)}_j | j \in \{x, y, z\}, l \in \{1, \ldots, h_b\}\} \).

As our message embedding involves modifying bitplanes, which is equivalent to adding an integer vector to \( \hat{v}_i \), we can generate the final stego vertices using the I2D conversion,

\[
\begin{align*}
\mathbf{v}'_i = \begin{bmatrix}
\hat{v}'_{ix} \\
\hat{v}'_{iy} \\
\hat{v}'_{iz}
\end{bmatrix} &= \begin{bmatrix}
(v_{ix} + \left\lfloor \delta_{ix} \times 10^{k_d}\right\rfloor)/10^{k_d} \\
(v_{iy} + \left\lfloor \delta_{iy} \times 10^{k_d}\right\rfloor)/10^{k_d} \\
(v_{iz} + \left\lfloor \delta_{iz} \times 10^{k_d}\right\rfloor)/10^{k_d}
\end{bmatrix}.
\end{align*}
\]

(3)

While in theory \( \delta_{ij} \) can take any value, in our constructed embedding domain, modifying the bitplanes causes \( \hat{v}_{ij} \) to vary by an integer only. Thus, we round down \( \delta_{ij} \times 10^{k_d} \), which can also be viewed as discretizing \( \delta_{ij} \) directly. We define \( \delta_{ij} \in \mathcal{I} \subset \{\delta/10^{k_d} \mid \delta \in \mathbb{Z}\} \), where the cardinality of \( \mathcal{I} \) is limited to avoid perceptible mesh distortion resulting from message embedding. This enables us to express problem (2) as follows:

\[
\begin{align*}
\text{minimize} \quad & E_p[D_j(V')] \\
\text{s.t.} \quad & H(p) = \sum_{i=1}^{n} \sum_{\delta_{ij} \in \mathcal{I}} p(\delta_{ij}) \ln \frac{1}{p(\delta_{ij})} = m_j.
\end{align*}
\]

(4)

The optimal \( p \) has a form of Gibbs distribution [25], namely

\[
p(\delta_{ij}) = \frac{e^{-\lambda_j} p(\delta_{ij})}{\sum_{\delta \in \mathcal{I}} e^{-\lambda_j} p(\delta)}, j \in \{x, y, z\}.
\]

(5)
The λ<sub>j</sub> is a scalar calculated through an iterative search with the condition H(ρ) = m<sub>j</sub>. Once ρ is determined, we can calculate the optimal p with (4) and (5).

**D. Design of FPD**

1) Selection of Efficient Steganalytic Features: The emergence of a new information security technique is inevitably accompanied by corresponding attacks, and the ensuing back-and-forth between attackers and defenders promotes progress on both sides. It is on this basis that we believe well-designed steganalyzers can contribute to the creation of more secure steganography.

In 3D steganalysis, NVT+, a 100-dimensional feature set Φ = \{φ<sub>i</sub><sub>j</sub><sup>100</sup>\}, has been demonstrated experimentally to possess the best steganalysis performance [14], [26]. Although the idea of preserving all steganalytic features during message embedding may be enticing, it is challenging to achieve in reality. Therefore, we need to identify which subfeatures of NVT+ are effective for steganalysis. The evaluation of subfeatures in NVT+ is presented in Appendix A, available online, and we conclude that subfeatures φ<sub>1</sub> ~ φ<sub>12</sub>, φ<sub>33</sub> ~ φ<sub>36</sub>, φ<sub>65</sub> ~ φ<sub>76</sub>, φ<sub>77</sub> ~ φ<sub>87</sub>, and φ<sub>88</sub> ~ φ<sub>100</sub> may be useful for designing a distortion model.

2) Design of Cost Function: Without loss of generality, we choose the first L best subfeatures from NVT+ to construct the cost function ρ as below.

\[
ρ(δ<sub>ij</sub>) = \mu \sum_{k=1}^{L} N_{or}(||S_{k}(M) - S_{k}(M(δ<sub>ij</sub>))||_1) \tag{6}
\]

where S<sub>k</sub> denotes an extractor w.r.t. the k-th subfeature, outputting a feature vector; M(δ<sub>ij</sub>) denotes that only v<sub>ij</sub> in M is altered and its change quantity is δ<sub>ij</sub>; N<sub>or</sub> is a normalization function; μ > 0 is a scaling factor set to prevent numerical calculation issues. Given that the steganalytic subfeatures mentioned above originate from different geometric features, and their values may be of diverse magnitude orders, N<sub>or</sub> plays a crucial role in balancing their contribution to ρ. Substituting (6) into (1) provides us with the specific form of the FPD. By minimizing the FPD, steganalytic, local, and global geometric features can be preserved to a certain extent, which is well-demonstrated in subsequent experiments.

**E. Improvement to FPD**

Although subfeatures φ<sub>1</sub> ~ φ<sub>12</sub>, φ<sub>33</sub> ~ φ<sub>36</sub>, φ<sub>65</sub> ~ φ<sub>76</sub>, φ<sub>77</sub> ~ φ<sub>87</sub>, and φ<sub>88</sub> ~ φ<sub>100</sub> exhibit satisfactory performance in steganalysis, their suitability for our distortion model requires further verification.

The specific analysis of φ<sub>1</sub> ~ φ<sub>12</sub> and φ<sub>33</sub> ~ φ<sub>36</sub> is presented in Appendix B, available online. The final result indicates that both of these subfeatures exhibit a lack of generalization against certain specifically designed steganography. Additionally, the utilization of φ<sub>33</sub> ~ φ<sub>36</sub> may result in ρ ≡ 0 for certain vertex movements. Thus, we decide not to consider them in ρ. The remaining three subfeatures do not expose the same defects as the first two. In this regard, it is reasonable to include them in (6).

However, directly employing the original subfeature extractors from [14] results in time-consuming calculations of ρ(δ<sub>ij</sub>) for all vertices and δ<sub>ij</sub> ∈ I, as presented in Table I. To alleviate this problem, we propose an accelerated method as follows.

Before going any further, we need a brief understanding of the original subfeature extractors w.r.t. φ<sub>65</sub> ~ φ<sub>76</sub>, φ<sub>77</sub> ~ φ<sub>87</sub>, and φ<sub>88</sub> ~ φ<sub>100</sub>. As described in [14], the three subfeature extractors rely heavily on their respective predefined neighborhood patterns, i.e., N<sub>1</sub>, N<sub>2</sub>, and N<sub>3</sub>, which are shown in Fig. 2(a), (c), and (e), respectively. Correspondingly, three types of NVTs of size 3 × 3 are defined by

\[
T_{1i} = \sum_{f \in N_{1}(v_i)} W_{1}(f)n(f) \cdot n(f)^{\top}, \quad i \in \{1, \ldots, |V|\},
\]

\[
T_{2i} = \sum_{f \in N_{2}(f_i)} W_{2}(f)n(f) \cdot n(f)^{\top}, \quad i \in \{1, \ldots, |F|\},
\]

\[
T_{3i} = \sum_{f \in N_{3}(f_i)} W_{3}(f)n(f) \cdot n(f)^{\top}, \quad i \in \{1, \ldots, |F|\},
\]

where n(f) denotes the face normal of f and the complete form of the weighting coefficient W<sub>k</sub>(f) is given in Appendix C, available online. Calculating ρ with the original subfeature extractors is also introduced in Appendix C, available online. Here we skip directly to the description of our accelerated method.

Equation (7) implies that each vertex or face of a mesh actually has a corresponding NVT: Let T<sub>1i</sub>, T<sub>2i</sub>, and T<sub>3i</sub> be associated with v<sub>i</sub>, f<sub>i</sub>, and f<sub>i</sub>, respectively. After establishing these associations, we introduce a new concept called vertex influence domain.

**Definition 1 (Vertex Influence Domain):** Given a neighborhood pattern N<sub>k</sub>, the influence domain of v<sub>i</sub>, denoted by ID<sub>k</sub>(v<sub>i</sub>), is a set of vertices or faces, whose corresponding tensors T<sub>k</sub> are changed when v<sub>i</sub> is moved.
Remark 1 are not uncommon, e.g., the classical Stanford bunny meshes.

Making full use of Remark 1 and $ID_k(v_i)$ can markedly speed up the calculation of $\rho(\delta_{ij})$ for each vertex in $M$ and $\delta_{ij} \in I$. This is because when only $v_i \in M$ is moved by $\delta_{ij}$, all we need to do is recalculate the eigenvalues of a tiny portion of NVTs that correspond to elements in $ID_k(v_i)$ to obtain $\delta_{ik}(M'(\delta_{ij}))$. The thorough explanation and relevant pseudo code are provided in Appendix E, available online.

F. Implementation of Message Embedding and Retrieval

1) Calculation of BMP: Prior to introducing our U & A-BMP, it is essential to provide a brief review of the pipeline of $Q$-layered STC. As illustrated in Fig. 3(a), the process of generating stego bitplanes involves calculating BMP and using (1-layered) STC in each layer. Additionally, it is crucial to note that the BMP in the layer $l$ is directly determined by the actual embedding results of all previous $l-1$ layers.

However, there are some problems with directly applying STC to 3D steganography. First, [18] provides a manual BMP calculation approach only, and only source code tailored for binary, ternary, and pentary embedding examples are given in their code repository, which are far from meeting the demand for embedding capacity in 3D settings; Second, the BMP calculation becomes enormously cumbersome as we have to rework our program when $Q$ varies.

To facilitate the application of STC in 3D mesh steganography, we design the U & A-BMP, the core of which is described in Algorithm 1. Within this algorithm, we employ an array $A$ (excluding its first row) to keep track of the BMP of each bit layer (Line 10) and use the check function $C(l)$ to represent the $l$-th least significant bit of an integer. The first row of $A$ is reserved for storing joint probability. On Lines 9 and 10, $b_{ij}^{(1)}, \ldots, b_{ij}^{(2)}$, and $b_{ij}^{(1)}$ are used to represent the actual embedding results of previous $l-2$ layers, with their values being omitted for simplicity. Furthermore, the 1-layered STC utilized on Line 11 follows the flipping lemma outlined in [18]. To explain Algorithm 1 more clearly, we present an example visualizing the BMP calculation for 2-layered STC in Fig. 3(b). The example begins with variable initialization in Lines 1–4, followed by the successive embedding of messages on $B_x^{(1)}$ and $B_x^{(2)}$ by using the for-loop present in Lines 5–14.

Notably, Algorithm 1 is just a pseudo code for the purpose of articulating the logic behind U & A BMP. During practical code implementation, we leverage some algorithmic acceleration techniques to ensure our U & A BMP is computationally efficient, such as the usage of vectorization to avoid an abuse of for-loops.

2) Message Embedding: We present the complete message embedding algorithm in Algorithm 2. We set $|I| = 2^Q$ and $\rho(\delta_{ij}) < \infty$ for $\delta_{ij} \in I$ to avoid wetness in each layer of $Q$-layered STC as much as possible. For more details about wet points in STC, we suggest [18].
3) **Message Retrieval:** Since we adopt the Q-layered STC, the whole embedding process can be considered as performing the 1-layered STC on a collection of bitplanes. Accordingly, the message retrieval on each stego bitplane can be implemented by

\[ m_j(l) = H^{STC} \cdot B_j(l), l \in \{1, 2, \ldots, Q\}, \ j \in \{x, y, z\}, \] (9)

where \( H^{STC} \in \{0, 1\}^{m(l) \times |V|} \) denotes the parity-check matrix derived from a predefined submatrix \( \tilde{H} \in \{0, 1\}^{h \times w} \). Typically, \( 6 \leq h \leq 15 \) and \( \frac{1}{w+1} < \frac{|m(l)|}{|V|} < \frac{1}{w} \). Note that \( H^{STC} \) is shared between the sender and recipient; the height of \( \tilde{H} \) is suggested to be set to a moderately large value according to [18].

### IV. EXPERIMENTAL RESULTS AND ANALYSIS

**A. Setup of Experiments**

1) **Datasets:** All experiments in this article are conducted on the following three 3D mesh datasets:

- **Princeton ModelNet (PMN)**: The dataset comprises an extensive assortment of manually-drawn 3D CAD models that cater to researchers in the fields of computer graphics, computer vision, robotics, and cognitive science.

- **Princeton Shape Benchmark (PSB)**: The dataset consists of 400 watertight mesh models, gathered from various web repositories and divided into 20 object categories, with each category comprising 20 meshes.

- **The Stanford Models (TSM)**: The dataset comprises multiple dense reconstructed mesh models, which were scanned by laser triangulation range scanners.

2) **Criteria for Steganography:** We evaluate our algorithm from three aspects:

- **Steganographic Security:** Steganography security can be understood as the ability of steganographic algorithms to resist steganalysis. In this article, we utilize the “out-of-bag” error estimate \( E_{OOB} \) [38] to measure steganography security. It is an unbiased estimate of the testing error, which is defined by

\[ E_{OOB} = \frac{1}{2N_{data}} \sum_{i=1}^{N_{data}} (B(n)(\tilde{\Phi}_i) + 1 - B(n)(\Phi'_i)), \] (10)

where \( \Phi_i \) and \( \Phi'_i \) are cover and stego steganalytic features respectively; \( n \) denotes the number of base learners used in the ensemble classifier, and the base learner is \( B(n)(\cdot) \in \{0, 1\} \).

- **Embedding Capacity:** Embedding capacity in this article is measured by the relative payload \( \alpha = \frac{\text{payload}}{\text{total size}} \) (bits per vertex), where \( \text{payload} \) is the size of the practical payload.

- **Robustness:** The ability to resist various digital attacks, such as affine transformation, vertex reordering, noise addition, smoothing and simplification.

3) **Experimental Setting:** All experiments are conducted in a computer equipped with an Intel Xeon 5218R processor, 64 GB of RAM, and MATLAB 2019(a).

**B. Evaluation of FPD**

1) **Time Complexity of Cost Calculation:** Based on the analysis in Sections III-D and III-E, we have identified that subfeatures \( \phi_{65} \sim \phi_{76}, \phi_{77} \sim \phi_{88}, \) and \( \phi_{89} \sim \phi_{100} \) are the most appropriate for the FPD design. However, computing \( p(\delta) \) for all vertices and \( \delta \in I \) with the original method in [14] is excessively time-consuming. In Section III-E, we further analyze the three subfeatures and propose an improved version called IFPD. We use the notations OFPD-S1, OFPD-S2, and OFPD-S3 to refer to the three original cost calculation methods based on \( \phi_{65} \sim \phi_{76}, \phi_{77} \sim \phi_{88}, \) and \( \phi_{89} \sim \phi_{100} \), respectively. Correspondingly, we denote the improved methods as IFPD-S1, IFPD-S2, and IFPD-S3. To showcase the effectiveness of the IFPD series, we randomly select several meshes from the
Algorithm 1. U & A-BMP

Input: The cover bitplane set \( B_j = \{ B_j^{(l)} \}_{l=1}^{|V|} \), the perturbation set \( I \), the optimal distribution set \( P_j = \{ P_{ij} \}_{j=1}^{|V|} \), where \( P_{ij} = \{ p(\delta_{ij}) \mid \delta_{ij} \in I \} \).

Output: Stego bitplane set \( B'_j \).
1. Set \( Q = \inf \arg \max_{q \in \mathbb{N}} \| I \| \leq 2^q \) // \([\cdot] \) denotes Iverson bracket;
2. Extend \( I \) until \( |I| = 2^Q \), with adding elements’ corresponding probability set to 0;
3. Create an array \( A \) of size \((Q + 1) \times |V|\) with all entries being 0;
4. Initialize \( \Pi_j = P_j ; \) and \( A_1 = 1 ; / / ^{"/"} \) means to take all indices in the current dimension.
5. for \( l = 2 \) to \( Q + 1 \) do
6. for \( i = 1 \) to \(|V|\) do
7. \( \Pi_{ij} = \{ p(\delta_{ij}) | \delta_{ij} \in I \} \); \( \Pi_{ij} \in C_{l-1} \left( i_{ij} + \delta_{ij} \times 10^d \right) = 0 \} ;
8. \( \Pi_{ij}^1 = \Pi_{ij} \cup \Pi_{ij}^1 \); \( P(b_{ij}^{(l-1)}) = \bar{P}(b_{ij}^{(l-1)} , b_{ij}^{(l-2)} ; \ldots , b_{ij}^{(1)}) = SUM(\Pi_{ij}^1) ;
9. \( A_{1i} = P(b_{ij}^{(l-1)} = 0 , b_{ij}^{(l-2)} , \ldots , b_{ij}^{(1)}) / A_1 ;
10. \) Perform 1-layered STC on \( B_j^{(l-1)} \) to generate \( B_j^{(l-1)} \);
11. for \( i = 1 \) to \(|V|\) do
12. \( \Pi_{ij} = \Pi_{ij}^1 \) if \( b_{ij}^{(l-1)} = 0 \), otherwise \( \Pi_{ij} = \Pi_{ij}^1 \); \( \Pi_{ij} = \Pi_{ij}^1 \); \( A_{1i} = P(b_{ij}^{(l-1)} | b_{ij}^{(l-2)} , \ldots , b_{ij}^{(1)}) \times A_{1i} ;
13. \( B'_j = \{ B_j^{(b)} , B_j^{(Q+1)} , B_j^{(Q)} , \ldots , B_j^{(1)} \} ;
14. \) return \( B'_j \).

PSB, PMN, and TSM datasets, and we calculate the vertex-changing cost by using various cost calculation methods with \( I = \{ 1 / 10^d , \ldots , 1 / 10^d , 0 , 1 / 10^d , \ldots , 6 / 10^d \} \). Table I presents the time consumed by each method on each mesh. It is evident from the table that the OFPD-based methods are excessively time-consuming, as processing meshes with more than 7000 vertices can take more than an hour. In contrast, the IFPD series significantly accelerates the cost calculation process.

2) Evaluations of Different Distortion Models: To evaluate the efficacy of the FPD, we opt to compare it with the vertex normal based-distortion (VND) and Gaussian curvature-based distortion (GCD) [24]. In [24], the two distortion models are defined by

\[
VND(v_i) = \frac{1}{\log(||n(v_i) - R(n(v_i))||_2 + 1) + \sigma}
\]

\[
GCD(v_i) = \frac{1}{K(v_i)^{\beta} + \sigma} \tag{11}
\]

where \( n(v_i) \) denotes the vertex normal of \( v_i \) and \( R(n(v_i)) \) is its reference counterpart; \( K(v_i) \) is the discrete Gaussian curvature at \( v_i \); and \( \sigma \) and \( \beta \) are constants stabilizing the numerical calculation. Notably, VND and GCD presume that the nonzero elements in \( I \) carry equal costs of change. In the following experiments, in addition to IFPD-S1, IFPD-S2, and IFPD-S3, we also evaluate their merged version IFPD-CS. Four relative payloads of 1.5, 3, 4.5, and 6 bpv are considered in the comparison experiments. Specifically, we set \( I = \{ 0 , 1 / 10^d \} \) for \( \alpha = 1.5 \) bpv and adopt the 1-layered STC for message embedding. For \( \alpha = 3 \) bpv, we restrict \( I = \{ -1 / 10^a , 0 , 2 / 10^d \} \) and perform embedding with 2-layered STC. \( I = \{ -1 / 10^a , 0 , 1 / 10^d , \frac{1}{3} / 10^d , \frac{2}{3} / 10^d , \frac{4}{3} / 10^d \} \) and \( I = \{ -1 / 10^a , 0 , 1 / 10^d , \frac{1}{3} / 10^d , \frac{2}{3} / 10^d , \frac{4}{3} / 10^d \} \) are used for relative payloads of 4.5 and 6 bpv, respectively. Correspondingly, 3-layered STC and 4-layered STC are utilized. We use the STC toolbox accessible on website1 and set the height of parity-check submatrix \( H \) as 10 and the width as 2. The normalization function’s form in (6) can be chosen, and in this work, it is selected as \( \rho_{\min} = \rho_{\min} \cdot \rho_{\max} \); where \( \rho_{\min} \) and \( \rho_{\max} \) represent the minimum and maximum values of \( \rho(\delta_{ij}) \), respectively. Additionally, the scalar \( \mu \) in (6) is set to 1, and the quantization parameter \( k_{id} \) is set to 6.

In accordance with the aforementioned settings, we conduct experiments on two datasets, namely, PSB and PMN. For the PSB dataset, 280 randomly-selected meshes and their stego counterparts are used as the training set, and the remaining pairs are used for testing purposes. For the PMN dataset, we choose 1280 median-volume meshes as cover meshes, with 1000 cover-stego pairs constituting the training set and the rest reserved for testing. The division of the training and test sets for each dataset is repeated 30 times. To evaluate the distortion models, we employ the 3D steganalyzer NVT+ as the evaluator. Furthermore, we train a separate evaluator for each distortion function and relative payload. The steganography security is assessed by calculating the average \( E_{OOB} \) over the 30 test sets, which is denoted by \( E_{OOB} \).

Algorithm 2: Adaptive 3D Mesh Steganography Based on FPD

Input: The cover mesh \( M = \{ V , E , F \} \), the quantization parameter \( k_{id} \), the bit number \( h_b \), the perturbation set \( I \), and the payload \( m \).

Output: Stego mesh \( M' \).
1: Set \( m_x = m_y = m_z = m \); \( \frac{m}{3} \);
2: for \( j \in \{ x , y , z \} \) do
3: Perform the D2I conversion for all vertices of \( M \) and obtain bitplane set \( B_j = \{ B_j^{(l)} \}_{l=1}^{|V|} \); \( B_j = \{ B_j^{(l)} \}_{l=1}^{|V|} \);
4: Calculate \( \rho(\delta_{ij}) \) for all vertices of \( M \) and \( \delta_{ij} \in I \) with (6);
5: Substitute \( m_j \) and \( \rho(\delta_{ij}) \) into (4) and obtain the optimal distribution \( P_j = \{ P_{ij} \}_{j=1}^{|V|} \).
6: Input \( B_j , I \), and \( P_j \) to the U & A-BMP algorithm and obtain the stego bitplane set \( B'_j \);
7: Perform the ID conversion on \( B'_j \) to generate \( V' \);
8: Concatenate \( V'_x \), \( V'_y \), and \( V'_z \) as \( V' \);
9: \( M' = \{ V' , E , F \} ;
10: \) return \( M' \).
Fig. 4 shows that the steganographic algorithm when using IFPD-S1 outperforms those when using VND and GCD in terms of security, with the difference becoming more significant as the relative payload increases. On the other hand, IFPD-S2, IFPD-S3, and VND perform similarly, while IFPD-CS exhibits remarkably better performance than that of VND and GCD, demonstrating IFPD’s effectiveness on the PSB dataset. In Fig. 4(b), IFPD-S1 also improves the steganographic security significantly, but its $E_{OOB}$ under each relative payload is lower than that on the PSB dataset. This is because, as described in [14], the NVT+ steganalyzer achieves better detection performance on the PMN dataset than that on the PSB dataset. Notably, IFPD-S1, IFPD-S2, and IFPD-S3 exhibit distinct performance on both datasets, with IFPD-S1 exhibiting an absolute advantage in anti-steganalysis, and the other two performing like VND and GCD. This suggests that using IFPD with subfeatures effective for steganalysis does not necessarily improve the algorithm’s security significantly, and preserving more features may not be as effective as anticipated. Despite these findings, to avoid overfitting to particular steganalytic features as much as possible [19], we still choose IFPD-CS as the distortion function for the subsequent experiments.

C. Comparison to State-of-the-Art Algorithms

To evaluate the anti-steganalysis ability of our algorithm, we compared it with four 3D mesh steganographic algorithms based on geometric modification, namely, VND/LSBR [24], Chao [6], Li [8], and HPQ-R [16]. All of these algorithms have high embedding capacities, with VND/LSBR and HPQ-R showing potential resistance to 3D steganalysis. We conducted relevant experiments on the PSB and PMN datasets, with relative payloads of 1.5, 3, 4.5, and 6 bpv. The division of the training and test sets aligns with the description in Section IV-B2. Since $k_0$ is set to 6, for fair comparison, we configure each algorithm as below. We set Chao’s interval number to $10^5$ and the maximum number of embedding layers to 2. For VND/LSBR, we set its initial embedding layer to 14 and reserve the last two layers for LSBR embedding. For Li, we fixed the size of the truncated space as 1 for each vertex. HPQ-R’s interval parameter is set to $10^{-5}$, and the number of subintervals is set to $2^{13}$. It’s worth noting that we did not consider HPQ-R at high relative payloads (i.e., 4.5 and 6 bpv) because a higher relative payload requires a larger subinterval number, which may lead to precision loss in vertex coordinates. We utilize the same embedding strategy for each relative payload as described in Section IV-B2.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 5. Average OOB error $E_{OOB}$ obtained across different steganalyzers as a function of the relative payload for IFPD-CS, Chao, Li, VND/LSBR, and HPQ-R on the PSB dataset and PMN dataset.

Fig. 6. Average OOB error $E_{OOB}$ obtained by IFPD-CS on different steganalyzers as a function of the relative payload on (a) the PSB dataset and (b) PMN dataset.
PMN. This is likely because these two steganalyzers both extract neighborhood-level representation-guided features for steganalysis, and our algorithm is indeed capable of preserving certain local geometric features. Furthermore, this also highlights a practical issue, namely, that current 3D steganalytic features may not be universal enough.

D. Statistical Significance Test

Considering that the test error distribution of Chao, Li, VND/LSBR, IFPD-CS, and HPQ-R is unknown, we utilize the Wilcoxon rank sum test to evaluate the statistical significance of the security improvement achieved by IFPD-CS compared to the other four algorithms under various relative payloads, steganalyzers, and datasets. The hypothesis test is constructed as follows.

\[ H_0 : \mu_1 = \mu_2, \quad H_1 : \mu_1 > \mu_2, \]  

where \( \mu_1 \) is the mean values of \( n_1 \) test OOB errors w.r.t. IFPD-CS; \( \mu_2 \) is the mean values of \( n_2 \) test OOB errors of the comparison counterpart. In this case, we set \( n_1 = n_2 = 30, \mu_1 = \mu_2 \) indicates that there is no significant difference between them. Let \( R \) denote the rank sum of the IFPD-CS test OOB error samples. Since \( n_1, n_2 \geq 10, R \sim N(\mu_R, \sigma^2_R) \) where \( \mu_R = \frac{n_1(n_1 + n_2 + 1)}{2} \frac{n_2(n_1 + n_2 + 1)}{12} \) and \( \sigma^2_R = \frac{n_1n_2(n_1+n_2+1)}{2} \). Therefore, we can use

\[ Z = \frac{R - \mu_R}{\sigma_R} \]  

as the test statistic. We use the built-in function \text{ranksum} of MATLAB to calculate the value of \( Z \) and obtain the corresponding p-value. The significance level is set to 0.05. If the p-value is less than the significance level, we reject the null hypothesis \( H_0 \), indicating that the security improvement of IFPD-CS is statistically significant. Experimental results on the PSB and PMN datasets are presented in Appendix F, available online. In general, our analysis reveals that when compared to Chao, VND/LSBR, Li, and HPD-R, IFPD-CS exhibits a significant improvement in steganography security.

E. Visualization of Embedding Impact

We intuitively display several embedding visualization examples from three different datasets in Fig. 8. Specifically, we measure the change degree of each vertex by the root-mean-square error and represent it with colors ranging from blue to yellow. We do not show the mesh edges because our algorithm does not change the topology of meshes. To further demonstrate the superiority of our algorithm, we also visualize its message embedding results compared to those of Chao, Li, and VND/LSBR at various relative payloads and on different datasets, as shown in Fig. 8. It is evident from the figures in Figs. 7 and 8 that our algorithm produces more acceptable visual effects of embedding impact on both the PSB and PMN datasets. In particular, stego meshes generated by our method exhibit fewer visual differences than those generated by other algorithms, especially at low relative payloads. This indirectly indicates the capability of our algorithm to preserve certain local and global geometric features, such as sharp features, to a significant extent. Furthermore, it is evident that Chao and Li’s message embedding operation falls short in covering all the vertices at times. This observable embedding issue can be traced back to the nonadaptive nature of their algorithms. In sharp contrast to Chao and Li, our algorithm and VND/LSBR algorithms do not display such a phenomenon. Nonetheless, the algorithm VND/LSBR includes the nonadaptive operation LSBR, which makes its embedding impact more conspicuous. Our algorithm assigns each mesh vertex a corresponding cost for vertex modification. In other words, our approach accounts for all vertices and carefully evaluates which vertices can be modified more and which vertices should be modified less during the message embedding process.

F. Analysis of Computational Complexity

As outlined in Section III-F, the message embedding procedure comprises two distinct parts: vertex-changing cost calculation and \( Q \)-layered STC embedding. Without loss of generality, let the time taken to calculate each \( \rho(\delta_{ij}) \) be \( t \). For a mesh with \( |V| \) vertices, once \( I \) is determined, the time complexity of computing all \( \rho(\delta_{ij}) \) can be given by \( O(3|V||I|t) \). The calculation efficiency of \( \rho \), vertex number, and \( I \) are the three crucial factors determining the time consumption of the first part. Fortunately, we propose an accelerated cost calculation algorithm that greatly reduces \( t \), thereby enhancing the practicality of our algorithm. The second part of the message embedding procedure can be further divided into two parts: BMP calculation and 1-layered STC. Our adoption of acceleration
tricks in implementing U&A BMP can make the time taken by BMP calculation negligible. However, the execution time of the STC algorithm is highly dependent on the height of the selected parity-check submatrix. In our work, we select a submatrix with a height of 10, which results in a very short consumption time for STC.

G. Robustness Analysis

In traditional steganography, scholars are more concerned with steganographic security than robustness. However, to move steganography from the laboratory into the real world, a discussion on the robustness of steganography is essential. Zhou et al. [26] enumerated several digital attacks that target 3D mesh steganography, i.e., affine transformation, vertex reordering, smoothing, simplification, and noise addition. Unfortunately, we must admit that our algorithm is not robust enough against these attacks, primarily due to the embedding domain. As we learned from Section III-F3, our algorithm’s message retrieval procedure is highly sensitive to the modification of vertex coordinates and the change in vertex order, which we overlook when constructing the embedding domain.

V. CONCLUSION

In this article, we propose a highly adaptive 3D mesh steganographic algorithm based on FPD. Our innovations lie in the construction of the PLS problem, the embedding domain, the distortion function design and improvement, and the BMP calculation for \( Q \)-layered STC. In Section IV, we also present extensive experiments demonstrating the superiority of our algorithm in resisting steganalysis.

For a long time, research on 3D steganography has been overlooked. However, with the rapid advancement of 3D technology, 3D models are becoming increasingly popular in our daily lives. Hence, we believe that 3D steganography has a bright future. There are still many topics worth exploring in 3D steganography. For instance, the distortion function plays a crucial role in adaptive steganography, and a well-designed distortion function can significantly enhance steganography security. Although the experimental results presented in Fig. 6 show promising results, we are concerned that the proposed FPD may overfit certain steganalyzers. As a result, designing a more general distortion model is the focus of our future research. Additionally, as highlighted in Section IV-G, our algorithm is susceptible to common attacks due to our constructed embedding domain. Therefore, in our future work, we plan to focus on developing a more robust embedding domain.

REFERENCES

[1] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 77–85.
[3] F. Cayre and B. Macq, “Data hiding on 3-D triangle meshes,” *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 939–949, Apr. 2003.
[4] C.-M. Wang and Y.-M. Cheng, “An efficient information hiding algorithm for polygon models,” *Comput. Graph. Forum*, vol. 24, no. 3, pp. 591–600, Sep. 2005.
[5] Y.-M. Cheng and C.-M. Wang, “A high-capacity steganographic approach for 3D polygonal meshes,” *Vis. Comput.*, vol. 22, no. 9–11, pp. 845–855, Sep. 2006.
[6] M.-W. Chao, C.-H. Lin, C.-W. Yu, and T.-Y. Lee, “A high capacity 3D steganography algorithm,” *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 2, pp. 274–284, Mar./Apr. 2009.
[7] V. Itier and W. Puech, “High capacity data hiding for 3D point clouds based on static arithmetic coding,” *Multimedia Tools Appl.*, vol. 76, no. 24, pp. 1–25, Dec. 2017.
[8] N. Li, J. Hu, R. Sun, S. Wang, and Z. Luo, “A high-capacity 3D steganography algorithm with adjustable distortion,” IEEE Access, vol. 5, pp. 24457–24466, 2017.
[9] Y. Yang and I. Ivrissimtzis, “Mesh discriminative features for 3D steganalysis,” ACM Trans. Multimedia Comput. Commun. Appl., vol. 10, no. 3, Apr. 2014, Art. no. 27.
[10] Z. Li and A. G. Bors, “3D mesh steganalysis using local shape features,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., Barcelona, Spain, 2016, pp. 2144–2148.
[11] D. Kim, H.-U. Jang, H.-Y. Choi, J. Son, I.-J. Yu, and H.-K. Lee, “Improved 3D mesh steganalysis using homogeneous kernel map,” in Proc. Int. Conf. Inf. Sci. Appl., Macau, China, 2017, pp. 358–365.
[12] Z. Li and A. G. Bors, “Steganalysis of 3D objects using statistics of local feature sets,” in Inf. Sci., vol. 415/416, pp. 85–99, Nov. 2017.
[13] Z. Li, D. Gong, F. Liu, and A. G. Bors, “3D steganalysis using the extended local feature set,” in Proc. IEEE Int. Conf. Image Process., Athens, Greece, 2018, pp. 1683–1687.
[14] H. Zhou, K. Chen, W. Zhang, C. Qin, and N. Yu, “Feature-preserving tensor voting model for mesh steganalysis,” IEEE Trans. Vis. Comput. Graphics, vol. 27, no. 1, pp. 57–67, Jan. 2021.
[15] Z. Li and A. G. Bors, “Steganalysis of meshes based on 3D wavelet multiresolution analysis,” Inf. Sci., vol. 522, pp. 164–179, 2020.
[16] Z. Li, S. Beegnon, W. Puech, and A. G. Bors, “Rethinking the high capacity 3D steganography: Increasing its resistance to steganalysis,” in Proc. IEEE Int. Conf. Image Process., Beijing, China, 2017, pp. 510–514.
[17] J. Fridrich and T. Filler, “Practical methods for minimizing embedding impact in steganography electronic imaging 2007,” in Proc. Int. Soc. Opt. Photon., 2007, pp. 650–602.
[18] T. Filler, J. Judas, and J. Fridrich, “Minimizing additive distortion in steganography using syndrome-trellis codes,” IEEE Trans. Inf. Forensics Secur., vol. 6, no. 3, pp. 920–935, Sep. 2011.
[19] T. Pevný, T. Filler, and P. Bas, “Using high-dimensional image models to perform highly undetectable steganography,” in Proc. 12th Int. Workshop Inf. Hiding, Calgary, Canada, 2010, pp. 161–177.
[20] V. Holub and J. Fridrich, “Designing steganographic distortion using directional filters,” in Proc. IEEE Int. Workshop Inf. Forensics Secur., Tenerife, Spain, 2012, pp. 234–239.
[21] V. Holub, J. Fridrich, and T. Denemark, “Universal distortion function for steganography in an arbitrary domain,” Eurasip J. Inf. Secur., vol. 2014, no. 1, pp. 1–13, Jan. 2014.
[22] L. Guo, J. Ni, and Y. Shi, “Uniform embedding for efficient JPEG steganography,” IEEE Trans. Inf. Forensics Secur., vol. 9, no. 5, pp. 814–825, May 2014.
[23] L. Guo, J. Ni, W. Su, C. Tang, and Y. Shi, “Using statistical image model for JPEG steganography: Uniform embedding revisited,” IEEE Trans. Inf. Forensics Secur., vol. 10, no. 12, pp. 2669–2680, Dec. 2015.
[24] H. Zhou, K. Chen, W. Zhang, Y. Yao, and N. Yu, “Distortion design for feature-preserving 3-D mesh steganography,” IEEE Trans. Multimedia, vol. 21, no. 6, pp. 1384–1398, Jun. 2019.
[25] T. Filler and J. Fridrich, “Gibbs construction in steganography,” IEEE Trans. Inf. Forensics Secur., vol. 5, no. 4, pp. 705–720, Dec. 2010.
[26] H. Zhou, W. Zhang, K. Chen, W. Li, and N. Yu, “Three-dimensional mesh steganography and steganalysis: A review,” IEEE Trans. Vis. Comput. Graphics, vol. 28, no. 12, pp. 5006–5025, Dec. 2022, doi: 10.1109/TCG.2021.3075136.
[27] Y. Yang, N. Peyerimhoff, and I. Ivrissimtzis, “Linear correlations between spatial and normal noise in triangle meshes,” IEEE Vis. Comput. Graphics, vol. 19, no. 1, pp. 45–55, Jan. 2013.
[28] A. Bogomjakov, C. Gotsman, and M. Isenburg, “Distortion-free steganography for polygonal meshes,” Comput. Graph. Forum, vol. 27, no. 2, pp. 637–642, Apr. 2008.
[29] N.-C. Huang, M.-T. Li, and C.-M. Wang, “Toward optimal embedding capacity for permutation steganography,” IEEE Signal Process. Lett., vol. 16, no. 9, pp. 802–805, Sep. 2009.
[30] S.-C. Tu, W.-K. Tai, M. Isenburg, and C.-C. Chang, “An improved data hiding approach for polygon meshes,” Vis. Comput., vol. 26, no. 9, pp. 1177–1181, Sep. 2010.
[31] S. Tu, H. Hsu, and W. Tai, “Permutation steganography for polygonal meshes based on coding tree,” Int. J. Virtual Reality, vol. 9, no. 4, pp. 55–60, Nov. 2010.
[32] S.-C. Tu and W.-K. Tai, “A high-capacity data-hiding approach for polygonal meshes using maximum expected level tree,” Comput. Graph., vol. 36, no. 6, pp. 767–775, Oct. 2012.

[33] N. Aspert, E. Galasca, Y. Maret, and T. Ebrahimi, “Steganography for three-dimensional polygonal meshes,” in Proc. SPIE Int. Soc. Opt. Eng., Seattle, WA, USA, 2002, pp. 211–219.
[34] M. G. Wagner, “Robust watermarking of polygonal meshes,” in Proc. Geometric Model. Process. Theory Appl., Hong Kong, China, 2000, pp. 201–208.
[35] Y. Maret and T. Ebrahimi, “Data hiding on 3D polygonal meshes,” in Proc. Multimedia Secur. Workshop, Magdeburg, Germany, 2004, pp. 68–74.
[36] H. Kaveh and M.-S. Moin, “A high-capacity and low-distortion 3D polygonal mesh steganography using surfacetetil transform,” Secur. Commun. Netw., vol. 8, no. 2, pp. 159–167, Jan. 2015.
[37] Z. Li and A. G. Bors, “Selection of robust and relevant features for 3-D steganalysis,” IEEE Trans. Cybern., vol. 50, no. 5, pp. 1899–2001, May 2020.
[38] J. Kodovsky, J. Fridrich, and V. Holub, “Ensemble classifiers for steganalysis of digital media,” IEEE Trans. Inf. Forensics Secur., vol. 7, no. 2, pp. 432–444, Apr. 2012.
[39] Y. Wang, L. Kong, Z. Qian, G. Feng, X. Zhang, and J. Zheng, “Breaking permutation-based mesh steganography and security improvement,” IEEE Access, vol. 7, pp. 183300–183310, 2019.

Yushu Zhang (Senior Member, IEEE) received the PhD degree in computer science and technology from the College of Computer Science, Chongqing University, China, in 2014. He held various research positions with Southwest University, City University of Hong Kong, University of Macau, and Deakin University, respectively. He is currently a professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. He has authored or coauthored more than 200 refereed journal articles and conference papers in these areas. His research interests include multimedia security and artificial intelligence, and blockchain. He is currently an associate editor for Information Sciences and Signal Processing.

Jiahao Zhu received the MS degree from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China, in 2020. He is currently working toward the Ph.D. degree with the School of Computer Science and Engineering, Sun Yat-sen University, China. His research interests include computer graphics and multimedia security.

Mingfu Xue (Senior Member, IEEE) received the PhD degree in information and communication engineering from Southeast University, Nanjing, China, in 2014. From 2011 to 2012, he was a visiting PhD student with Nanyang Technological University, Singapore. He is currently an Associate Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. He has been a technical program committee member for more than 20 international conferences. He has been the principal investigator of 11 research projects and participated in four other projects. He was the recipient of the Best Paper Award in ICCCS2015. His research interests include artificial intelligence security, secure and private machine learning systems, and hardware security.
Xinpeng Zhang (Member, IEEE) received the BS degree from Jilin University, China, in 1995, and the MS and PhD degrees from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a full time professor. He is also with the faculty of the School of Computer Science, Fudan University. From 2010 to 2011, he was with the State University of New York, Binghamton, as a visiting scholar, and also with Konstanz University as an experienced researcher, sponsored by the Alexander von Humboldt Foundation from 2011 to 2012. He has authored or coauthored more than 200 research papers. His research interests include multimedia security, image processing, and digital forensics. From 2014 to 2017, he was an associate editor of the IEEE Transactions on Information Forensics and Security.

Xiaochun Cao (Senior Member, IEEE) received the BE and ME degrees in computer science from Beihang University, Beijing, China, and the PhD degree in computer science from the University of Central Florida, Orlando, FL, USA. After graduation, he has spent about three years as a research scientist with Object Video Inc. From 2008 to 2012, he was a professor with Tianjin University, China. He is currently a professor with the School of Cyber Science and Technology, Sun Yat-sen University, Shenzhen Campus, Shenzhen, China. His dissertation was nominated for the University of Central Florida’s University Level Outstanding Dissertation Award. In 2004 and 2010, he was the recipient of the Piero Zamperoni Best Student Paper Award at the International Conference on Pattern Recognition. He is on the editorial board of the IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, and IEEE Transactions on Circuits and Systems for Video Technology.