This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

Distortion Design for Secure Adaptive 3D Mesh Steganography

Hang Zhou, Kejiang Chen, Weiming Zhang, Yuanzhi Yao and Nenghai Yu

***Abstract*—We propose a novel technique for steganography on 3D meshes so as to resist steganalysis. The majority of existing methods modulate vertex coordinates to embed messages in a non-adaptive way. We take account of complexity of local regions as joint distortion of a triple unit (vertice) and coding method such as Syndrome Trellis Codes (STCs) to adaptively embed messages, which owns stronger security with respect to existing steganalysis. Key to the distortion is a novel formulation of adaptive steganography, which relies on some effective steganalytic features such as variation of vertex normal. We provide quantitative and qualitative comparison of our method with several baselines against steganalytic features LFS64, LFS76 and ensemble classifiers and show that it outperforms the current state of the art. Meanwhile, we proposed an attacking method on steganography proposed by Chao et al. (2009) with a high detection rate.**

***Index terms*—**mesh steganography, mesh steganalysis, vertex normal, least significant bit replacement

I. Introduction

STEGANOGRAPHY is a fundamental task in the field of information security which hides secret data into a digital multimedia such as digital image, audio, video, texts, 3D meshes, to name a few, without arousing suspicion. It targets the communication between two parties over covert channels such that a potential eavesdropper cannot detect its existence. With the intention of minimizing statistical detectability, modern steganography can be formulated as a source coding problem that minimizes embedding distortion [1]. Syndrome-Trellis Codes (STCs), proposed by Filler et al. in [2], are nowadays a standard methodology for embedding while minimizing an arbitrary additive distortion function with a performance near the theoretical bound.

Distortion acts as the pivotal element of this general framework. Hitherto steganographic distortion for cover source is heuristically and empirically designed, which are considerably challenging due to lack of accurate models. A fine distortion grasp characteristics that can restrain detectability in a selected feature space by steganalysts. Of late, methods on devising distortion for spatial images [3]–[8] and JPEG images [5], [9], [10] have been proposed, bringing great improvements on the security of image steganography.

This work was supported in part by the Natural Science Foundation of China under Grant 61572452 and Grant U1636201.

H. Zhou, K. Chen, W. Zhang, Y. Yao and N. Yu are with CAS Key Laboratory of Electromagnetic Space Information, University of Science and Technology of China, Hefei, 230026, China. (e-mail: zh2991@mail.ustc.edu.cn, chenkj@mail.ustc.edu.cn, zhangwm@ustc.edu.cn, yaoyz@mail.ustc.edu.cn, ynh@ustc.edu.cn).

Corresponding author: Weiming Zhang.

As for image steganalysis, much have been well-studied in the literature. Steganalyzer's features are usually generated by exploiting correlations between the predicted residuals of neighboring pixels [11]. Fridrich et al. [12] and Ker [13] propose methods specifically for the detection of Least Significant Bit Replacement (LSBR). The Subtractive Pixel Adjacency Model (SPAM) [15] set for the second-order Markov model of pixel differences has a dimensionality of 686. Whereafter, Spatial Rich Model (SRM) [16] is proposed with 34,671 dimensions to earn a better performance.

In Big Data era, the availability of 3D mesh models of real objects has become an essential prerequisite in a variety of applications domain, such as reverse engineering and industrial metrology, cultural heritage preserving and restoration, several biomedical fields including orthopedics and orthodontics [17], and thus they are befitted candidates and rich resources to serve as the innocuous-looking and ideal hosts for data hiding. Recent advancement on data hiding includes 3D mesh watermarking and 3D mesh steganography, where 3D mesh watermarking [18]–[21] are techniques focusing on protecting copyright ownership and reduce counterfeiting of digital multimedia and 3D mesh steganography focusing on covert communication against steganalysis.

As for 3D mesh steganography, Cayre and Marq [22] consider a triangle as a two-state geometrical object and embed data by the modulation of the position of the orthogonal projection of the triangle summit on the opposite side. Cheng and Wang [23] improve the modulation with sliding, extending and rotating levels to embed data; they also combine both the spatial domain and representation domain [24] to improve capacity. Other follow-ups on small capacity mainly focus on perfecting visual distortion caused by modification [25], [26]. Chao et al. [27] provide multilayered high-capacity reversible steganography with space modulation and demodulation technique on principle axes by vertex projection. Yang et al. [28] embed data by modifying the LSBs of selected vertex coordinates where the capacity depends on the shape of the mesh and cannot be known in advance. Itier et al. [29] propose a steganographic method which hides data by the displacement of a vertex relative to its new position in the Hamiltonian path using static arithmetic coding. Li et al. [30] propose a key modulation based steganography with confined distortion. Li et al. [31] increase the resistance to steganalysis of mesh steganography of [29].

In the meantime, modern feature-based steganalysis [32], [33] on meshes starts with adopting a mesh model within which steganalyzers are built using machine learning tools. The mesh steganalysis approach proposed in [32] considers

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

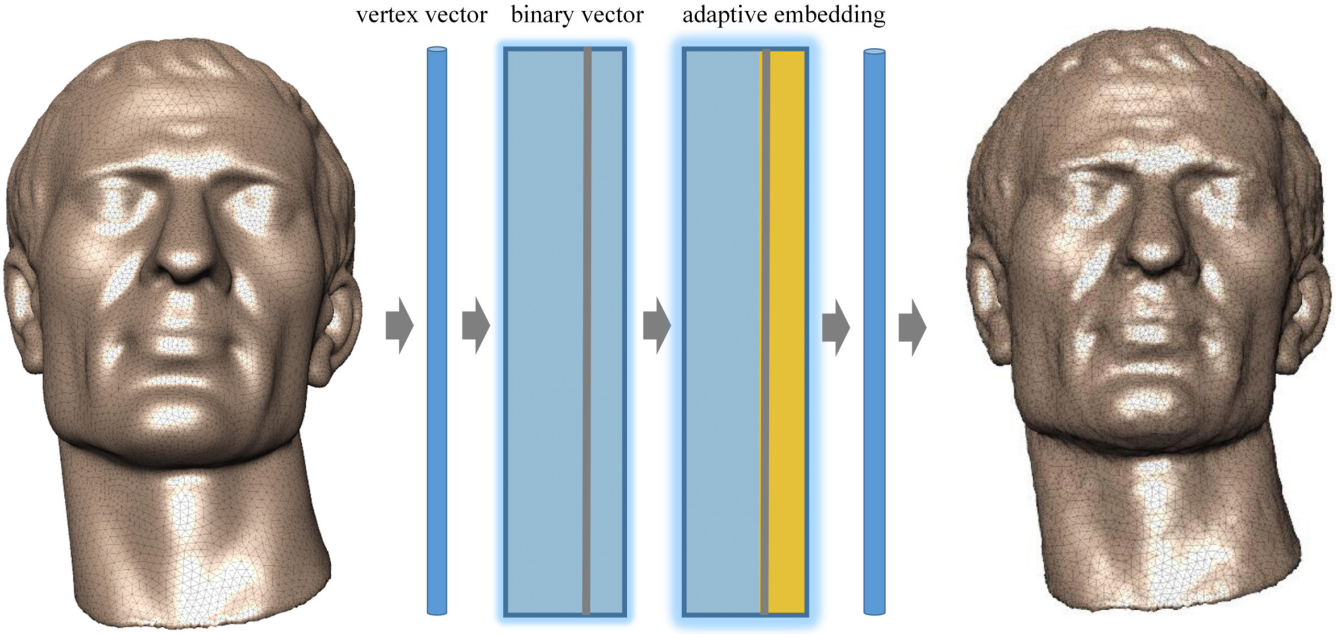


Fig. 1: Steganography on a statue with proposed method. The original cover mesh (*left*) and the corresponding stego mesh (*right*) are shown. The intensity of steganography is intensionally strengthened for visual representation.

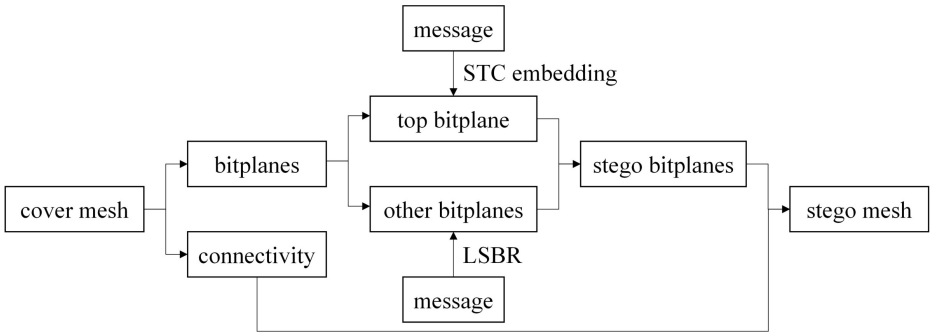


Fig. 2: Visual pipeline of the proposed method. Position of the top bitplane is acquired by the length of message and the number of vertices. The top bitplane is adaptively embedded by STCs. Note that the top bitplane is not equal to the highest bitplane. In the figure, we omit the bitplanes that cannot be modified.

various features including the norms of vertices in the Cartesian and Laplacian coordinate systems (where coordinates are linear transformed by Kirchhoff matrix) [34], the dihedral angle of faces adjacent to the same edge, and the face normal, which is called YANG208. Li et al. [33] dig deep in vertex normal and curvature features, combined with dimension-reduced YANG208, to obtain a more powerful feature set LFS52. Considering certain information hiding algorithms embedding data directly or indirectly in the spherical domain, they add another 24-dimensional feature vector extracted from the spherical coordinate system to obtain a more sophisticated LFS76 [35]. Kim et al. [36] combine LFS52 with edge normal, mean and total curvature features to form LFS64, which promotes detection performance.

Most of the above 3D mesh steganographic approaches modulate or shift vertex coordinates by their geometrical properties to embed messages, which belong to non-adaptive pattern. Early literatures embedding low-volume data disregarding steganalysis is more like fragile watermarking. In attempt to address the challenge of high-capacity data hiding, Chao's algorithm [27] is the first work that introduces multi-layered modulation technique into mesh steganography. While achieving excellent results on capacity and surface distortion facets, this approach has one important limitation: Chao et al. preprocess meshes by transforming the position of meshes into Principal Component Analysis (PCA) specified position. Such behavior will leak the position information and can be easily attacked by a specific classifier. Also, previous work [28]–[31] mention "large" capacity with data encoded into "float"-type 32-bit vertex, which covers some all-zero bits that should not carry message, and the capacity practically is smaller than stated.

In this paper, a novel 3D mesh steganography scheme is proposed. Different from all the previous algorithms, we straightforwardly operate on the binarized bitstream of vertices of meshes to implement adaptive steganography, which can avoid modifying all-zero bits. By analyzing varying steganalysis features, we take into account of prominent features to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

form distortion, and thus the ability to withstand malicious steganalysis is better than other methods. The main contributions of the paper are:

- Our proposed method avoids embedding data in invalid region. Existing schemes [30], [31] make modification of vertex coordinates in invalid region (which will be described in subsection III-C), which can be easily detected.
- In the study of image steganography, researchers focus on designing an effective distortion function for improving security. Yet 3D mesh steganography is at an early stage, and we are the first to design 3D steganographic distortion function to improve security performance. We bring in minimal distortion framework to embed data.
- Our proposed method avoids mesh rotation [27], a preprocessing before data embedding, which can be detected by a specifically designed detector.

After introducing the basic notation and terminology in Section II, we describe the distortion function in Section III. Section IV contains experimental results as well as the comparison with previous art. The security is measured empirically using classifiers trained with features on a range of relative payloads. Targeted attack of Chao's steganography is described in Section V. The paper is concluded in Section VI.

II. Related Work

In this paper, capital and lowercase boldface symbols stand for matrices and vectors respectively.

A. Minimal distortion framework

A cover sequence is denoted by $\mathbf{c} = (c_1, c_2, ..., c_N)$, where c_i is its i-th element. Embedding operation on c_i is formulated by the range I. An embedding operation is called binary if $|I| = 2$ and ternary if $|I| = 3$. We consider the case of binary embedding and $c_i \in C \triangleq \{0, 1\}$, where the possible values of stego elements are restricted to $I_i = \{c_i, \bar{c}_i\}$, where $\bar{c}_i$ is the bit flip operation on c_i. The steganography sender modifies $\mathbf{c}$ to $\mathbf{s} = (s_1, s_2, ..., s_N) \in \mathcal{S} \triangleq \{0, 1\}$ with probability $\pi(\mathbf{s}) = P(S = \mathbf{s})$, where $\pi(s_i)$ is the probability of changing c_i to s_i.

The minimal distortion steganography is formulated as follows. In the model established by Filler et al. in [2], the additive distortion D is defined as

$$D(\mathbf{c}, \mathbf{s}) = \sum_{i=1}^{N} \rho_i(\mathbf{c}, s_i), \tag{1}$$

where $\rho_i(\mathbf{c}, s_i)$ are bounded functions expressing the cost of replacing the cover element c_i with s_i. Since the cover $\mathbf{c}$ is assumed to be fixed, the distortion introduced by changing $\mathbf{c}$ to $\mathbf{s}$ can be simply denoted by $D(\mathbf{c}, \mathbf{s}) = D(\mathbf{s})$. It is supposed that $\rho_i(\mathbf{c}, c_i) = 0$ and $\rho_i(\mathbf{c}, \bar{c}_i) = \rho_i \in [0, \infty)$. In [2], the overall distortion for binary embedding can be rewritten as follows:

$$D(\mathbf{s}) = \sum_{i=1}^{N} \rho_i \cdot [c_i \neq s_i]. \tag{2}$$

For a given message vector $\mathbf{m}$ with a length of R, the sender wants to minimize the average distortion of Equation (2). Following the maximum entropy principle, the optimal π has a Gibbs distribution, and one can simulate optimal embedding by assigning

$$\pi(s_i) = \frac{\exp(-\lambda \rho_i(s_i))}{\sum_{s_i \in C_i} \exp(-\lambda \rho_i(s_i))}, \qquad i = 1, 2, \ldots, N \tag{3}$$

where the scalar parameter $\lambda > 0$ determined by the payload constraint

$$R = \sum_{i=1}^{N} \sum_{s_i \in C_i} \pi(s_i) \log \frac{1}{\pi(s_i)}. \tag{4}$$

For additive distortion, there exist practical coding methods to embed messages, such as STCs [2], which can approach the optimal embedding performance. Its embedding and extraction can be formulated as

$$\mathrm{Emb}_{\mathrm{STC}}(\mathbf{c}, \mathbf{p}, \mathbf{m}) = \arg \min_{\mathbf{s} \in coset(\mathbf{m})} D(\mathbf{c}, \mathbf{s}), \tag{5}$$

$$\mathrm{Ext}_{\mathrm{STC}}(\mathbf{s}) = \mathbf{s}\mathbf{H}_{\mathrm{STC}}^{T} = \mathbf{m}, \tag{6}$$

where $\mathbf{p} = (\rho_1, \rho_2, \ldots, \rho_N)$ denotes the distortion scalar vector, $coset(\mathbf{m})$ is the coset corresponding to syndrome $\mathbf{m}$ and $\mathbf{H}_{\mathrm{STC}} \in \{0, 1\}^{R \times N}$ is the parity-check matrix of the used STC shared between the sender and the recipient. For more details of the STC, please refer to [2].

B. Regular meshes

We work with meshes $M = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$. Let vertex set $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{N}$ represent the sequence of vertices encountered as a mesh is being traversed, where $\mathbf{v}_i = [v_{i,x}, v_{i,y}, v_{i,z}]^T$ in Cartesian coordinates system.

Uncompressed representations of mesh models typically specify each vertex coordinate as a 32-bit IEEE 754 single precision standard format, and the number of significant precision of each vertex coordinate is 23 bits (about seven decimal digits) [27]. Decimal representations of vertex coordinate components are displaced by binary formats [21], and binary representation of x-component of a vertex $\mathbf{v}_i$ is expressed by $\mathbf{b}_{i,x} = [b_{i,x}^1, b_{i,x}^2, ..., b_{i,x}^L]^T$, where the number of bitplanes $L = 31$. Thus the x-component of l-th bitplane is formed by $\mathbf{c}_x^l = [b_{1,x}^l, b_{2,x}^l, ..., b_{N,x}^l]^T$. Concrete implementation process is taken as an iterative way of obtaining the binary element of each bitplane. Denote the residual error of l-th bitplane by r^l, thus the binary element of $v_{i,x}$ is acquired by

$$b_{i,x}^l = \lfloor 2r^{l+1} \rfloor, \tag{7}$$

and

$$r^l = 2r^{l+1} - \lfloor 2r^{l+1} \rfloor. \tag{8}$$

in an iterated way. The starting condition is set by $r^{L+1} = v_{i,x}$ and l decreases from L. The illustrative figure is shown in Figure 1 and the corresponding visual pipeline in Figure 2.

C. Description of steganalysis features

Many techniques extract statistical features from mesh models and conduct two-class classification [32], [33], [35], [36]. Before feature extraction, it is necessary to preprocess vertices to canonical version: the mesh object is rotated and aligned according to its first and second principal axes, given by PCA

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

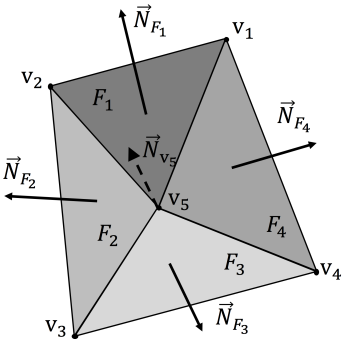


Fig. 3: Dihedral angles and vertex-based normals for representing local geometry properties of the surface.

algorithm. Afterwards, the object is scaled to fit inside a cube of sizes equalling to 1. The reference mesh M' that we use for calibration is produced by applying one iteration of Laplacian smoothing on the original mesh M, which updates the vertex $\mathbf{v}_i$ into $\mathbf{v}'_i$ as follows [37]:

$$\mathbf{v}'_i = \mathbf{v}_i + \frac{\tau}{\sum_{\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)} w_{ij}} \sum_{\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)} w_{ij}(\mathbf{v}_j - \mathbf{v}_i), \tag{9}$$

where τ is a scalar factor which is set by 0.2, and w_{ij} is the weight defined by

$$w_{ij} = \begin{cases} 1 & \text{if } \mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i) \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Features are designed by the differences between the mesh object and its smoothed version. All syntaxes of features in the paper follow the convention: *name* = $\{f\}\{\#\}$ where f represents feature and # is the sequence number of sub-features.

a) Yang40 features: The 40-dimensional feature vector YANG40 contains the most effective features from YANG208, used in [32], which corresponds to the statistics of features evaluated from the vertices, edges and faces that make up the given meshes. The first six components 'f1-f24' represent the absolute distance, measured along each coordinate axis x, y, z between the locations of vertices of the meshes M and M' after being normalized and aligned, in both the Cartesian and Laplacian coordinate systems. Next, the changes produced in the Euclidean distance between vertex location and the centre of the object, representing the vertex norms, are denoted by 'f25-f32'. 'f33-f36' evaluates the local mesh surface variation by calculating the changes in the orientations of faces adjacent to the same edge, which is measured by the absolute differences between the dihedral angles of neighbouring faces, calculated in the plane perpendicular on the common edge.

Likewise, shown in Figure 3, available features are extracted from faces. Changes in the local surface orientation are measured by calculating the angle between the surface normals $\overrightarrow{N}_{F_i}$ of the faces from the object $F_i \in \mathcal{F}$, and their correspondents $\overrightarrow{N}'_{F_i}$ from the smoothed object $F'_i \in \mathcal{F}'$. The absolute value of the angles between the two face normals is computed by $\arccos \frac{\overrightarrow{N}_{F_i} \cdot \overrightarrow{N}_{F'_i}}{\|\overrightarrow{N}_{F_i}\| \cdot \|\overrightarrow{N}_{F'_i}\|}$, where the features are denoted by 'f37-f40'.

b) Vertex normal features: The 'f41-f44' is the angle between the vertex normals of each two corresponding vertices in which a vertex normal is defined by the weighted sum of the normals of the faces that contain that vertex.

c) Curvature features: Local shape curvature is employed to measure the smoothness of mesh surface. Gaussian curvature and the curvature ratio formula used in [38] is considered here. The Gaussian curvature is defined by the product of the minimum principle curvature and the maximum principle curvature, and 'f45-f48' is evaluated by the absolute difference of two Gaussian curvature, while 'f49-f52' is acquired by two curvature ratios. Mean curvature and total curvature complement the aforementioned curvature information and enhance discrimination of features with 'f49-f52' and 'f53-f56' respectively.

d) Sphere coordinate features: Spherical coordinates provide a straight forward representation for most graphical objects in characterizing the distance from the centre and the location of each vertex on a sphere. The spherical coordinate system specifies a point in the space by a radius and two angles, totally forming 24 features.

III. Proposed Method

In this section, we first analyze the effect of different steganalysis features on cover and stego pair. Then we provide a general description of distortion function by taking into account of steganalytic features. A properly defined distortion function will improve the security of steganography. After that, we give a framework on how to embed messages.

A. From steganalysis to steganography

As Buckets effect reveals, the capacity of a bucket depends on the shortest board. Analogously, the security performance of steganography mainly depends on the most effective steganalytic features. Since we do not know which submodel of the steganalytic features is significant for designing steganography, the association of costs ρ_i to the features is generally very tough. By paying equal attention to each elements during steganography, we can acquire cover-stego pair that can be used to analyze which steganalytic features dominate in the discrimination of covers and stegos. Hereinto, the method Constant Distortion (CD) based matrix embedding proposed in [39] is used for paying equal attention to each elements on the operated bitplane for steganography, in which the goal is to solve the problem of minimizing the number of changed elements (the constant distortion profile) [2]. Afterwards, we evaluate the individual submodel of the steganalytic feature vector independently and set the costs ρ_i to reflect this ranking, meaning the optimality of submodel of features.

Our approach works as follows. First, we create a set of stego meshes embedded with CD method under certain payload. Then, we use Fisher Linear Discriminants (FLDs) criteria to evaluate, how good are individual features for detecting given embedding changes. The values of FLD criteria of individual elements may be either used directly to set the costs of embedding changes ρ_i or used to obtain insight into the problem and set the costs heuristically.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

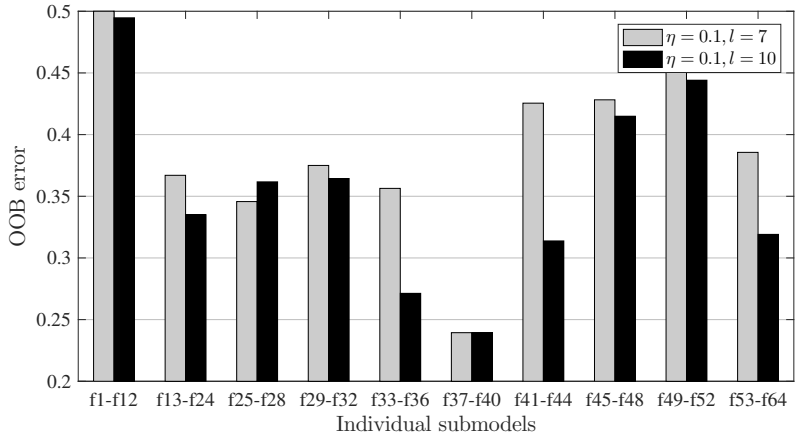


Fig. 4: OOB error estimates averaged over matrix embedding method and two payloads (0.1 bpv under 7 and 10 layers).

We use FLDs as base learners due to their simple and fast training. Denoting the cover and stego features from the training set as $\mathbf{x}^{(m)}$ and $\overline{\mathbf{x}}^{(m)}$, $m = 1, ..., N^{trn}$, respectively. The training makes use of the so-called "out-of-bag" (OOB) error estimate [16]:

$$E_{\text{OOB}}^{(L)} = \frac{1}{2N^{trn}} \sum_{m=1}^{N^{trn}} \left(B^{(L)}(\mathbf{x}^{(m)}) + 1 - B^{(L)}(\overline{\mathbf{x}}^{(m)}) \right). \tag{11}$$

We start by computing the OOB estimates for each submodel, including its different embedding payloads: 0.1 bpv (bit per vertices) under varying layers, for CD method. The intention is to investigate how the submodel ranking is affected by the payload on difference layers. To investigate the nature of submodels, in Figure 4 we plot for each submodel its OOB error estimate averaged over matrix embedding algorithm with 0.1 bpv under layer l. For example, 7 and 10 layers are selected. The fact that submodels from 'f37-f40' consistently provide lower OOBs than other submodels allows us to grasp the vulnerability of steganography to contend against 'f37-f40', which is the absolute value of angles between face normals. Figure 4 also nicely demonstrates that steganography on higher bitplane results in better detection rates, which should be attributed to deeper artifacts caused by modification. In summary, vertex normal features contributes to the design of steganography, which motivate us to design distortion function in the following subsection.

B. Distortion model

a) Vertex normal based distortion function: Following previous work, we use vertex normal of polygonal approximation which is defined by Nelson [40], the weighted sum of the normals of the faces that contain that vertex, to guide distortion definition, and it is formulated by

$$\overrightarrow{N}_{\mathbf{v}_i} = \sum_{F_j \in F_{\mathbf{v}_i}} \frac{S_j^{(i)} \cdot \overrightarrow{N}_{F'_j}}{\|e_{(\mathbf{v}_i, \mathbf{v}'_{F_j})}\| \cdot \|e_{(\mathbf{v}_i, \mathbf{v}''_{F_j})}\|}. \tag{12}$$

where $F_{\mathbf{v}_i}$ is the set of faces that contains the vertex $\mathbf{v}_i$, $\mathbf{v}'_{F_j}$ and $\mathbf{v}''_{F_j}$ are the two vertices adjacent to vertex $\mathbf{v}_i$ in the face F_j, $e_{(\mathbf{v}_1, \mathbf{v}_2)}$ represents the edge connecting vertices $\mathbf{v}_1$ and $\mathbf{v}_2$, and areas of an adjacent triangle $S_j^{(i)}$ is obtained by

$$S_j^{(i)} = \sqrt{q_j^{(i)} \left(q_j^{(i)} - e_j^{(i)}\right) \left(q_j^{(i)} - e_{j+1}^{(i)}\right) \left(q_j^{(i)} - p_j^{(i)}\right)}, \tag{13}$$

and semi perimeter $q_j^{(i)} = \left(e_j^{(i)} + e_{j+1}^{(i)} + p_j^{(i)}\right)/2$, as shown in Figure 5.

Therefore, the distortion design against targeted steganalytic features is considerately regarded as a major contribution to the distortion function. We quantify the distortion using outputs of a vertex normal to construct the distortion function.

The cost value is obtained by the reciprocal of absolute value of the ℓ_2 norm of vertex normal between cover mesh and Laplacian-smoothed mesh,

$$\rho_i = \frac{1}{\text{g}(\|\overrightarrow{N}_{\mathbf{v}_i} - \overrightarrow{N}_{\mathbf{v}'_i}\|_2) + \sigma}, \qquad i = 1, 2, \ldots, N \tag{14}$$

where $\text{g}(x)$ is a typically monotonous mapping function utilized for promoting steganography performance and $\sigma > 0$ is constants stabilizing the numerical calculations. For brevity, **V**ertex **N**ormal **D**istortion is abbreviated to VND.

b) Discrete Gaussian curvature based distortion function: Dialectically, we employ an inferior feature as distortion so as to have a comparative trial. Discrete Gaussian curvature of a vertex is related to angles and faces that are connected to that surface. As shown in Figure 5, the sharpness of the spherical polygon is approximated by the angle deficit of the polyhedron $\Delta(\mathbf{v}_i)$,

$$\Delta(\mathbf{v}_i) = 2\pi - \sum_{j=1}^{E} \theta_j^{(i)}, \tag{15}$$

where E is the number of adjacent triangles of the inspected point, and $\theta_j^{(i)}$ is the angle between two successive edges $e_j^{(i)}$ and $e_{j+1}^{(i)}$ of the i-th vertex, which is acquired by

$$\theta_j^{(i)} = \arccos\left[\frac{\left(e_j^{(i)}\right)^2 + \left(e_{j+1}^{(i)}\right)^2 - \left(p_j^{(i)}\right)^2}{2e_j^{(i)} e_{j+1}^{(i)}} \right], \qquad j = 1, 2, \ldots, E \tag{16}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

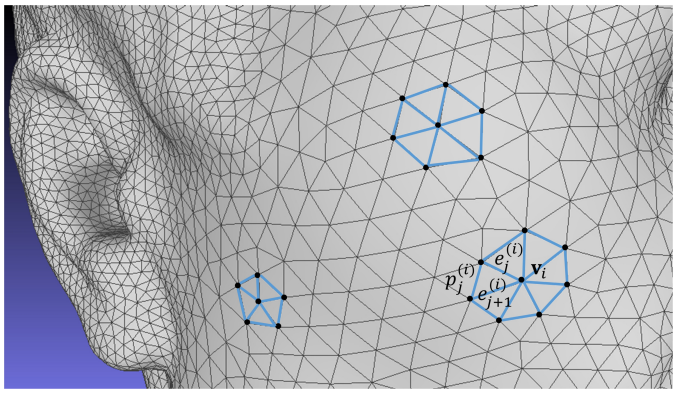


Fig. 5: Example of 1-ring neighbors of triangulation in Cartesian coordinate system, selected from zoomed region in the mesh in Figure 1. $\mathbf{v}_i$ is the selected vertices, which is surrounded by 7 triangles. Another two local regions each has 6 and 5 adjacent triangles[1].

where p_i is the side which is opposite of the angle and $e_1^{(i)} = e_{E+1}^{(i)}, i = 1, 2, \ldots, N$.

The area of each triangular face of the polyhedron can be partitioned into three equal parts, one corresponding to each of its vertices, so that the total area related to point $\mathbf{v}_i$ on the polyhedron is $\sum_{j=1}^{E} S_j^{(i)}/3$.

Assume that the curvatures are uniformly distributed around the vertex, discrete Gaussian curvature is determined as

$$K(\mathbf{v}_i) = \frac{\Delta(\mathbf{v}_i)}{\sum_{j=1}^{E} S_j^{(i)}/3}. \tag{17}$$

If a local region v_i is smooth, then $K(\mathbf{v}_i)$ tends to be a small value converging to zero.

Intuitively, a fine embedding algorithm embeds data into noisy areas with sharpness and irregularity that are not easily modellable or predictable, and these areas should be paid low costs. Hence, we introduce the following distortion function based on discrete Gaussian curvature as follows:

$$\rho_i' = \frac{1}{|K(\mathbf{v}_i)|^{\alpha} + \sigma}, \qquad i = 1, 2, \ldots, N \tag{18}$$

where α is scalar element. Likewise, **G**aussian **C**urvature **D**istortion is abbreviated to GCD.

C. Embedding strategy

We borrow LSB embedding technique from image steganography and convert vertex coordinates into multiple bitplanes to embed data. Given that embedding data in lower bitplane causes less artifact to the overall vertex coordinates, we consider embedding in lower bitplane first and then upper bitplane iteratively. We directly operate embedment of messages on bitplanes, which help avoid modifying invalid regions that cannot be embedded. The idea is simple yet effective in resistance against steganalysis.

As shown in Figure 6, the operation zone on a mesh consists of unmodified region, embedded region and invalid region. The invalid regions include 8 bitplanes that are all zeros, which cannot be embedded because once they are modified, a specifically designed detector can find the modification caused by steganography. Since the modification amplitude in lower layer is less than that in higher layer, we consider embedding message in lower layers as a priority. The embedded region and unmodified region are segmented by the actual message length. Because each element of any bitplane has two modes of modification, the maximum length of message that a bitplane carries is $N \cdot \log_2 2 = N$ bits. The number of layers in embedded region is acquired by $\lceil \frac{m}{N} \rceil$, and thus the number of layers in unmodified regions is $23 - \lceil \frac{m}{N} \rceil$. In the embedded region, we determine to adaptively embed messages on the top layer with the above designed distortion function and embed messages with LSBR on the remaining lower layers.

[1]Refer to [21] for more investigation of distributions of quantity of adjacent triangles.

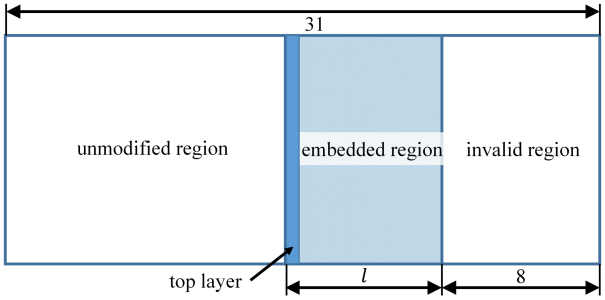


Fig. 6: Illustration of the operation zone on a regular mesh consisting of unmodified region, embedded region and invalid region. In the embedded region, the top layer is adaptive embedded and other layers are embedded with LSBR.

As mentioned, in additive distortion model, the modifications on elements are assumed to be independent and thus minimizing the overall costs is equivalent to minimizing the sum of costs of individual changed elements. The simplest way to conduct payload distribution in additive distortion rule is to serve each dimension of triple unit as the same cost and the previous cost value is evenly paid to x, y and z axes by

$$\begin{aligned} \rho^{(i)}(x) &= \rho_i, \\ \rho^{(i)}(y) &= \rho_i, \\ \rho^{(i)}(z) &= \rho_i. \end{aligned} \tag{19}$$

Note that each channel is individually embedded with STC without affecting the distortions of other channels.

D. Pseudo-code Procedure

To further clarify the scheme of steganography, in Algorithm 1 we provide a pseudo-code that describes the implementation procedure.

IV. Evaluation and results

A. Setups

Princeton Segmentation Benchmark[1] (PSB) is a mesh segmentation dataset with 354 objects [41]. 260 pairs of cover-objects are used for training and 94 pairs of stego-objects for testing, same in the configuration in the previous art [33].

Princeton ModelNet[2] (PMN) contains 12,311 mesh data for computer vision, computer graphics, robotics and cognitive science [42]. We take ModelNet40 with 40 categries for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

Algorithm 1: Distortion Based Steganography Procedure

Input: A cover mesh $\mathbf{X}$ with N vertices; m bits of message which determines the relative payload of target $\eta = m/N$.

Output: The stego mesh $\mathbf{Y}$.

Obtain the trio binary cover bitplanes from the vertex sequence $\mathcal{V} = \{\mathbf{v}_i\}_{i=0}^{N}$;
Acquire the number l of bitplanes that need to be modified to embed messages;
Utilize the distortion function with Equation (14) to calculate mesh distortion ρ_i, $i = 1, 2, \ldots, N$;
Assign the distortion to the cover bitplane $\mathbf{c}_x^l$, $\mathbf{c}_y^l$ and $\mathbf{c}_z^l$ correspondingly;
Embed m_l bits of the message into cover bitplane with STCs according to the distortion and output the stego bitplane $\mathbf{s}_x^l$, $\mathbf{s}_y^l$ and $\mathbf{s}_z^l$;
for $j = l - 1$ **to** 1 **do**
    Embed N bits of the message into cover bitplane with LSBR and output the stego bitplane $\mathbf{s}_x^j$, $\mathbf{s}_y^j$ and $\mathbf{s}_z^j$;
end
Reconstruct and output the stego vertices $\mathcal{V}_s$;
Integrate the vertices $\mathcal{V}_s$ and other mesh structures $\{\mathcal{E}, \mathcal{F}\}$ to form the stego mesh $\mathbf{Y}$.

training and testing. A preprocessing with only 4,000 meshes are selected with a median-volume meshes in favor of time-saving. We use 50% for training and 50% for testing.

The detectors are trained as binary classifiers implemented using the FLD ensemble [43] with default settings. A separate classifier is trained for each embedding algorithm and relative payload. The ensemble classifier by default minimizes the total classification error probability under equal priors

$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2}(P_{\mathrm{FA}} + P_{\mathrm{MD}}), \tag{20}$$

where P_{FA} and P_{MD} are the false-alarm probability and the missed-detection probability respectively.

To precisely compare the steganographic method with prior art, we define relative payload of embedding η by the ratio of total length of message and number of vertices, $\eta = m/N$ bpv. The feature sets used are LFS64 [35] and LFS76 [36]. All tested embedding algorithms are conducted for small relative payload $\eta \in \{0.1, 0.2, ..., 1.0\}$ in top bitplane and large relative payload $\eta \in \{1, 2, ..., 23\}$ in remaining bitplanes. 30 different splits of the given mesh objects are considered into the training and testing data sets. The ultimate security is qualified by the average of all the error rates over all 30 trials, and larger P_{E} means stronger security.

B. Determining mapping function

Mapping function $\mathrm{g}(x)$ has two important attributes:

- The value domain of $\mathrm{g}(x)$ should be $[0, +\infty)$ theoretically. Since we are trying to give distortion profile, the mapped value should never be smaller than 0, that the function $\mathrm{g}(x)$ should be nonnegative.
- $\mathrm{g}(x)$ should be monotonically increasing. In the assumption of the VND rule, larger elements acquired by Equation (14) are those elements which steganalysis features are not sensitive to, and smaller they are, the more suitable they are for embedding. Under this assumption, $\mathrm{g}(x)$ is supposed to be monotonically increasing.

[1]http://segeval.cs.princeton.edu/
[2]http://modelnet.cs.princeton.edu/

TABLE I: Comparison of Average Testing Error $\overline{P}_{\mathrm{E}}$ of Mapping Functions under LFS64 Steganalyzer w.r.t. Relative Payload under Single Embedding Layer $l = 12$ and $\gamma = 0.2$ on PSB dataset.

Model	Function $\mathrm{g}(x)$	Average Testing Error $\overline{P}_{\mathrm{E}}$
Linear	x	.4388 ± .0203
Radical	$\sqrt{x}$	.4415 ± .0209
Exponential	$\exp(x)$	.4016 ± .0176
Logarithmic	$\ln(x+1)$	.4468 ± .0205

In following, we consider several possible monotone functions to search for the optimum. Besides linear model, exponential model, logarithmic model, and radical model are in an intercomparison of steganographic performance. Our experiment is configured according to the settings in Table I, and logarithmic model provides the largest $\overline{P}_{\mathrm{E}}$ than other models, which is taken as the mapping function of our proposed distortion scheme. Thus Equation (14) can be rewritten as

$$\rho_i = \frac{1}{\ln(\|\overrightarrow{N}_{\mathbf{v}_i} - \overrightarrow{N}_{\mathbf{v}'_i}\|_2 + 1) + \sigma}. \qquad i = 1, 2, \ldots, N \tag{21}$$

C. Comparison under single-layered steganography

We first test VND implemented with single-layered steganography on the top layer to see the improvement of the proposed distortion function. Figure 7 (*left*) shows the results of LFS64 testing error when conducting adaptive steganography on layer $l = 7$ w.r.t. varying relative payloads. Three comparison experiments are carried out, Gaussian curvature based adaptive steganography (GCD), constant distortion based steganography (CD) and LSB replacement (LSBR). The scalar factor α in Equation (18) is searched with $\alpha = 1$ performing the best. With better distribution of distortion costs, VND has better security against GCD, CD and LSBR. GCD and CD have comparable security, indicating that the performance of Gaussian curvature is akin to that of constant distortion of CD, and is not strong enough to depict vertex cost to resist these steganalysis.

On both ends of curvature, the averaged testing errors are almost the same among all steganographic algorithms because when on the left end, the payload is too small to distribute, resulting in a near 50% error rate; when on the right end, full payload makes all the adaptive steganography converting to the specific non-adaptive steganography, i.e. modifying cover bits directly by message bitstream. By the way, the maximum improvement against LSBR on layer $l = 7$ is 17% with single-layered steganography. Such intermediate results indirectly demonstrate the efficacy of proposed VND algorithm. Similarly, Figure 7 (*right*) shows the results of LFS64 testing error when conducting adaptive steganography on layer $l = 12$. The curvatures of the two figures have same

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

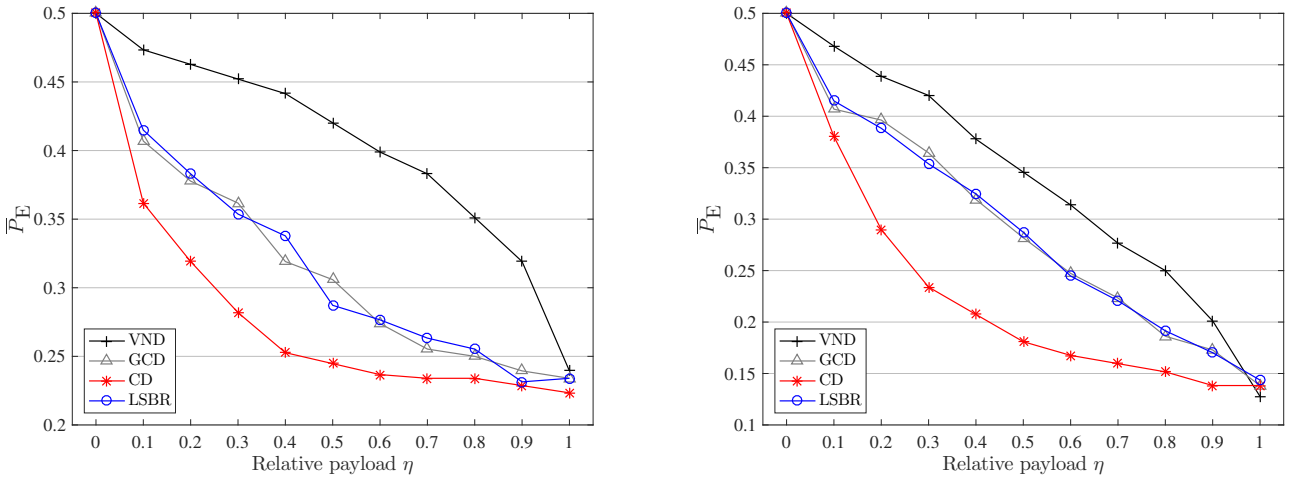


Fig. 7: The varying trends of LFS64 testing error w.r.t. relative payload under single embedding layer $l = 7$ (*left*) and $l = 12$ (*right*).

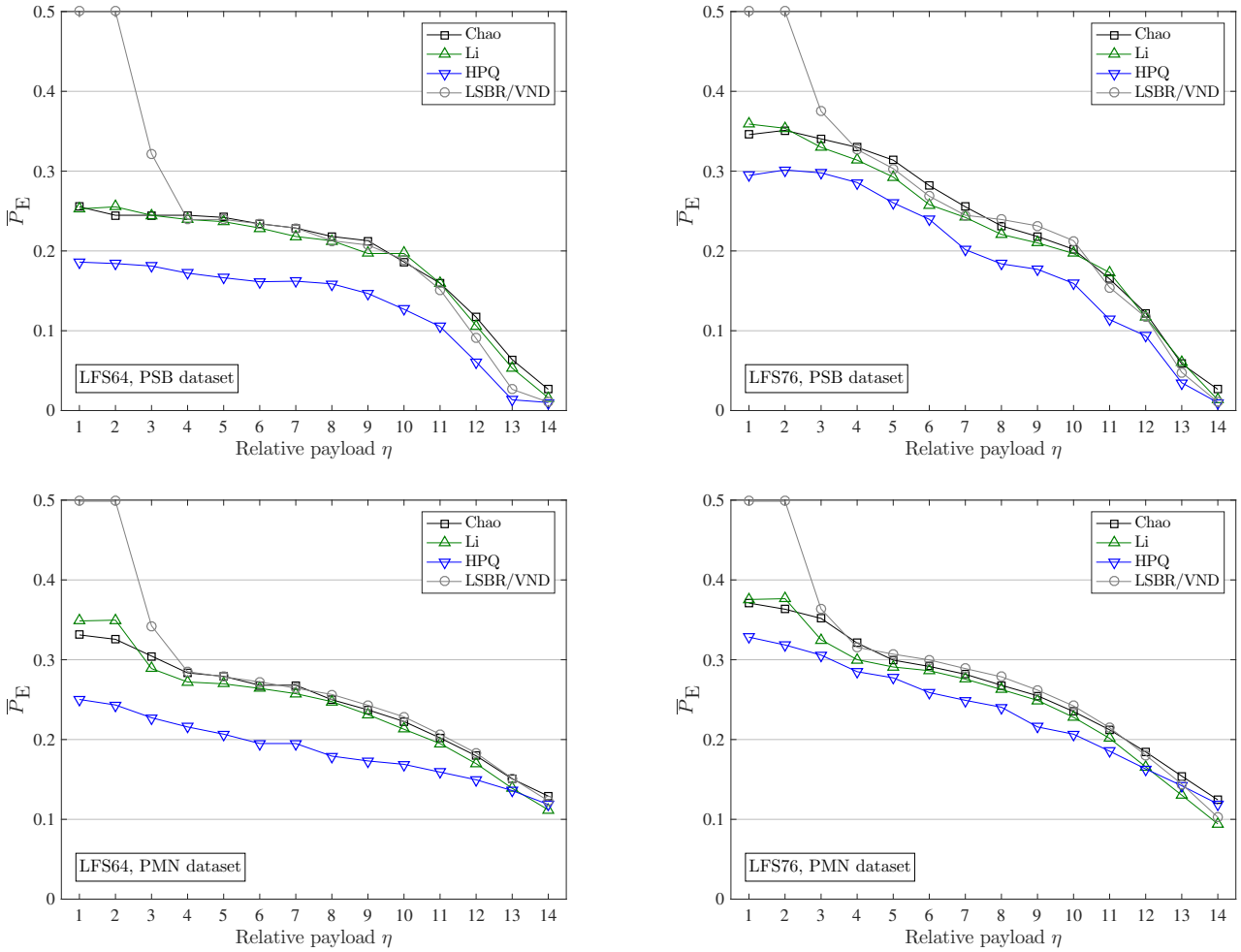


Fig. 8: The varying trends of average testing error w.r.t. integral relative payload. PSB dataset tested under LFS64 features (*top left*); PSB dataset tested under LFS76 features (*top right*); PMN dataset tested under LFS64 features (*bottom left*); PMN dataset tested under LFS76 features (*bottom right*).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

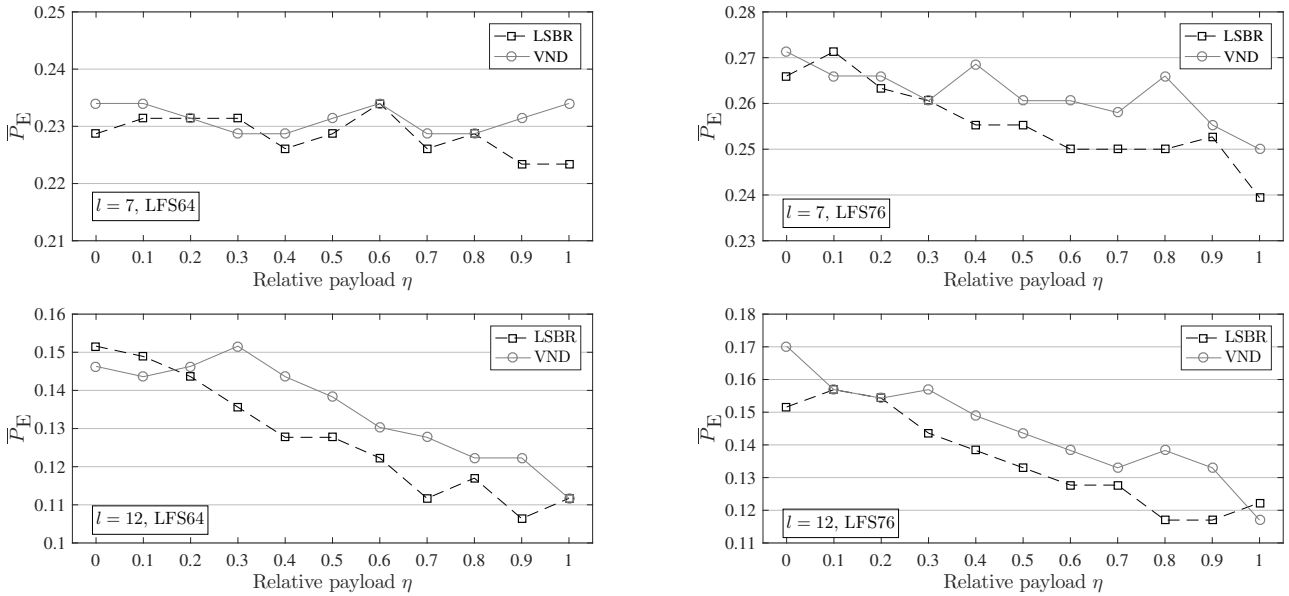


Fig. 9: The varying trends of LFS64 and LFS76 average testing error w.r.t. relative payload under multiple embedding layers (*top left*) $l = 7$ and LFS64 steganalyzer, (*top right*) $l = 7$ and LFS76 steganalyzer, (*bottom left*) $l = 12$ and LFS64 steganalyzer, (*bottom right*) $l = 12$ and LFS76 steganalyzer for VND and LSBR method tested in PSB dataset.

variation tendency and steganography on higher layer have fewer improvements than that on lower layer.

D. Comparison to Prior Art

Early mesh steganographic methods [22]–[24] embed data by simple vertex modulation, which own low capacity, thus we do not compare security performance with them. Recently, high-capacity steganographic schemes [27], [30], [31] have been developed. [30], [31] embed data in invalid regions, which can be easily detected by a specifically designed detector. Below, we compare the performance of proposed VND and LSBR with the state of the art.

To assess how much LSBR/VND embedding strategy improves the security of stego meshes, Figure 8 shows $\overline{P}_{\mathrm{E}}$ for integral relative payload η bpv together with the results for Chao [27], Li [30], HPQ [31], proposed LSBR and VND, where LSBR and VND have similar performance as the payload of top bitplane is fully embedded. The features are selected as LFS64 and LFS76 since they are by far the strongest detectors. The top two figures are the detection rates carried under PSB dataset while the bottom two are under PMN dataset. As apparent from all four subgraphs in Figure 8, with increasing η the performance of LSBR/VND decreases greatly while this decreases for Chao, Li and HPQ are rather gradual and very small under small payloads. When the payload is larger than 13 layers, Chao exceeds LSBR/VND, which attributes to the PCA preprocessing. The transformation by PCA makes the first principle component to have the largest possible variance while lower variance in other dimensions, the shifting on first and second components smartly mitigate the distortion on meshes, and reduce the detection by steganalyzers. However, Chao's algorithm leaks information when conducting preprocessing, which can be detected by targeted classifier, as explained in Section V. Numerical values of $\overline{P}_{\mathrm{E}}$ of Figure 8 are provided in the Appendix A.

In Figure 9 and Figure 10 we assess the improvement that VND has brought. The trend is similar to that in Figure 7 but the boost of VND is minor than that on the single layered embedding. We apply a z-test to evaluate the statistical significance of VND algorithm. The hypotheses are:

$$\boldsymbol{H}_0 : \mu_1 = \mu_2; \qquad \boldsymbol{H}_1 : \mu_1 \neq \mu_2,$$

in which μ_1 and μ_2 are the mean values of testing errors of VND and LSBR, $\mu_1 = \mu_2$ represents that there is no significant difference between them.

The z-score z is computed by

$$z = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1}{n_1} + \frac{\sigma_2}{n_2}}},$$

where n_1 and n_2 are the numbers of testing samples, σ_1 and σ_2 are standard deviations of VND and LSBR, respectively. By looking up the z-score in a table[3] of the standard normal distribution, the corresponding p-value can be obtained. A lower p-value indicates a lower probability that the null hypothesis $\boldsymbol{H}_0$ holds. If the p-value is less than a threshold, the null hypothesis $\boldsymbol{H}_0$ is rejected, and the improvement is deemed statistically significant and reliable. We set the level of significance z at 5%.

The numerical values of $\overline{P}_{\mathrm{E}}$ of Figure 9 and Figure 10 are provided in Appendix A. Bold font means the promotion is statistically significant, where we use the testing errors of VND and LSBR under LFS64 and LFS76 as examples. Under different payloads, layers and steganalyzer features, except for payloads converging to 0/1, the test statistic z

[3]http://math.arizona.edu/~rsims/ma464/standardnormaltable.pdf

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

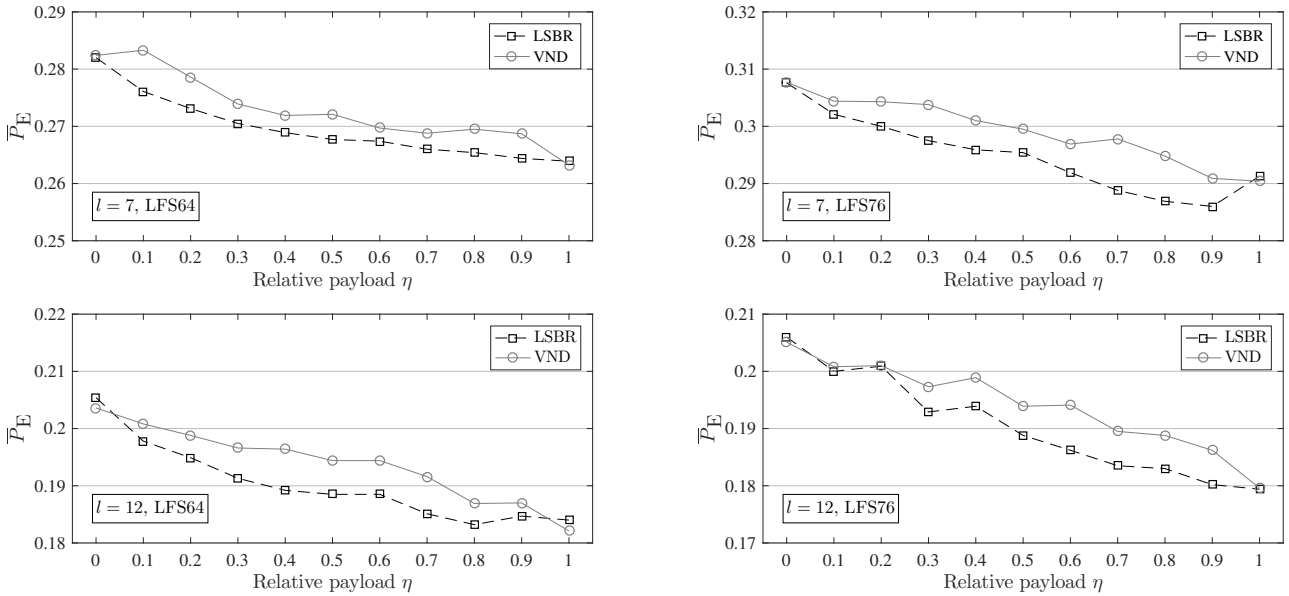


Fig. 10: The varying trends of LFS64 and LFS76 average testing error w.r.t. relative payload under multiple embedding layers (*top left*) $l = 7$ and LFS64 steganalyzer, (*top right*) $l = 7$ and LFS76 steganalyzer, (*bottom left*) $l = 12$ and LFS64 steganalyzer, (*bottom right*) $l = 12$ and LFS76 steganalyzer for VND and LSBR method tested in PMN dataset.

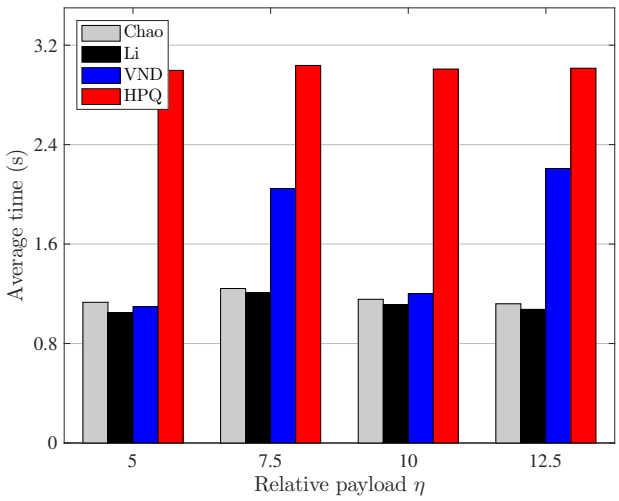


Fig. 11: Average computational time of 4,000 meshes from PMN dataset using Chao, Li, HPQ and VND under four different payloads, respectively.

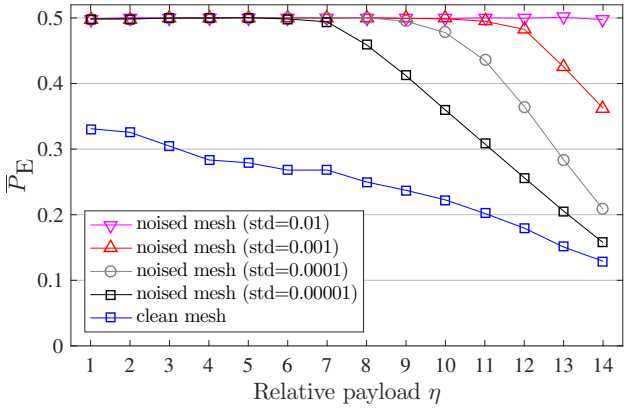


Fig. 12: The varying trends of average testing error w.r.t. payload under LFS64 features of clean and noised PMN dataset.

values are always larger than the corresponding quantile $z_{0.05/2}$, which implies the promotions have statistical significance. It is worth mentioning that lack of data in PSB dataset causes the fluctuation of $\overline{P}_E$ (large standard deviation) higher than that of PMN dataset, which is difficult to show the superiority of VND.

Furthermore, visualizations of some commonly used models are shown in Figure 13. Each one from left to right columns corresponds to cover object, stego object with 9 layered embedding, 12 layered embedding and 15 layered embedding, respectively. The modification intensity is measured by ℓ_2 norm of coordinates between cover and stego objects. We restrict the maximum strength by the modification intensity between cover and stego objects under 15 layered embedding, thus for 9 and 12 layered embedding, the modification is barely seen. More messages we embed into the mesh, larger modification occurs on the mesh.

E. Comparison of computational complexity

We have analyzed computation complexity among four steganographic schemes, Chao, Li, HPQ and proposed VND. We use all the meshes in PMN dataset to test the average embedding time. Experimental environment: MATLAB R2017b under Windows 10, server configuration is Intel(R) Xeon(R) CPU E7-4820 and 32-GB RAM.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

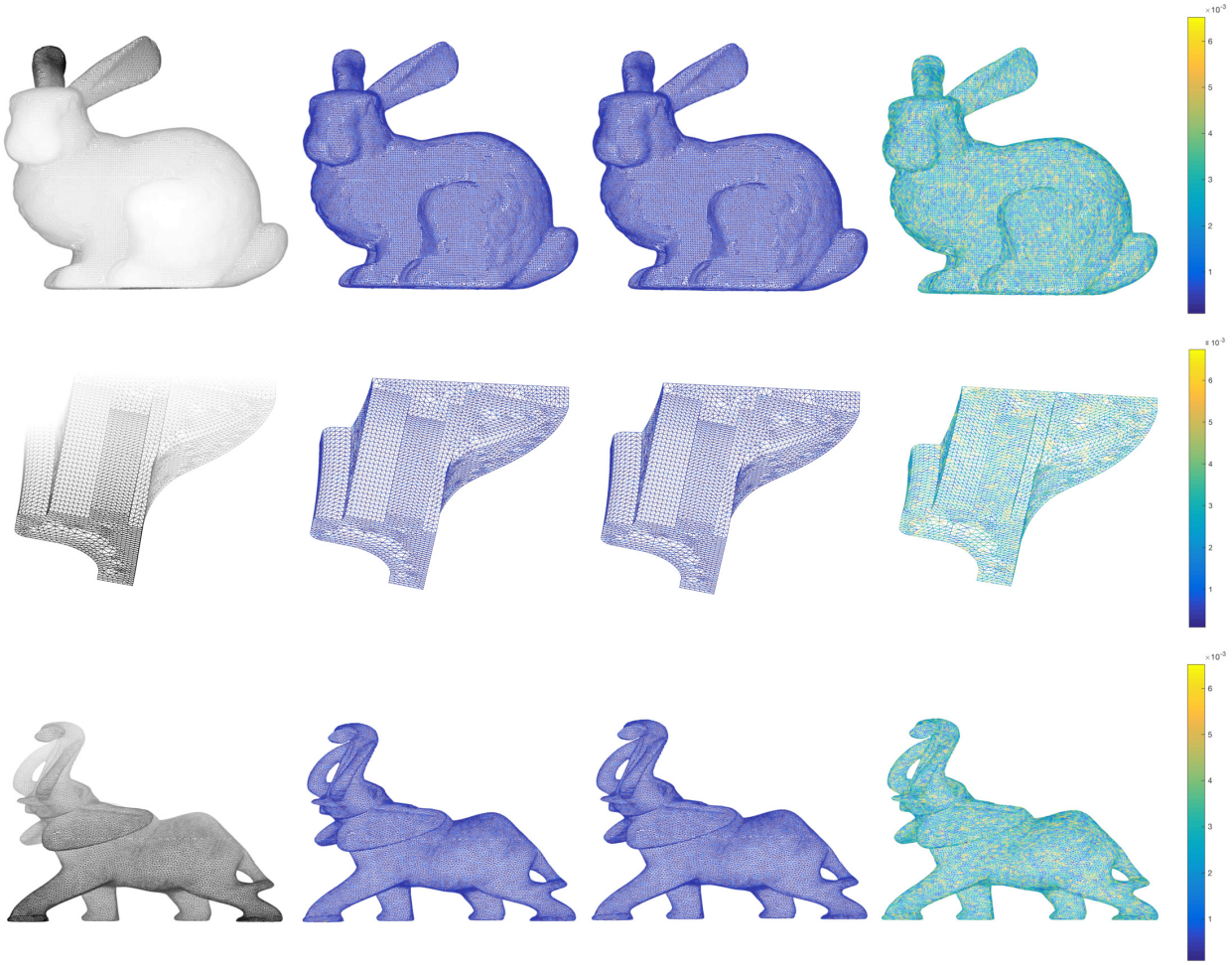


Fig. 13: Visualization of *Bunny*, *Fandisk*, *Elephant-50kv* models. Each one from left to right columns corresponds to cover object, stego object with 9 layered embedding, 12 layered embedding and 15 layered embedding, respectively. The modification intensity is measured by ℓ_2 norm of coordinates between cover and stego objects.

As shown in Figure 11, HPQ has longer computational time, for finding a Hamiltonian path is time-consuming. For payloads with fractions, the costs of VND is larger, which owns to the definition of distortion function. For payloads with integer, Li, HPQ and VND has similar computational time.

F. Comparison under noisy meshes

We have tested proposed method on noisy meshes injected with Gaussian noise with zero mean and varying standard deviations: $std \in \{0.00001, 0.0001, 0.001, 0.01\}$ under LFS64 steganalytic feature in PMN dataset. It is shown in Figure 12 that as a cover mesh, noisy mesh is more secure than clean mesh for steganography. When the noise degree increases, the mesh is more secure for data embedding.

V. Targeted attack on Chao's algorithm

A. Chao's algorithm

Chao's multilayered steganographic method [27] consists of two parts: model preprocessing and data embedding by modulation technique. Since there exist significant deficiencies in model preprocessing, we omit the description of data embedding hereinafter.

The preprocessing can be briefly described as follows. First, three end vertices of a cover model denoted as $\mathbf{v}_a$, $\mathbf{v}_b$, and $\mathbf{v}_c$ are selected by PCA. Given the first and second principal axes, orthogonally project all vertices onto these two axes. The vertices that fall on the two extreme ends of the first principal axis are selected as the end vertices $\mathbf{v}_a$ and $\mathbf{v}_b$. The vertex that fall on the furthest extreme end of the second principal axis is selected as the third end vertex $\mathbf{v}_c$. If an end vertex has multiple candidates, we simply select the nearest candidate (nearest to the principle axis) as the end vertex and slightly shift the other candidates in order to uniquely define an end vertex. The next step is to transform the cover model to align the vectors $\overrightarrow{\mathbf{v}_a\mathbf{v}_b}$, $\overrightarrow{\mathbf{v}_a\mathbf{v}_c}$ and $(\overrightarrow{\mathbf{v}_a\mathbf{v}_b} \times \overrightarrow{\mathbf{v}_a\mathbf{v}_c})$ with the x-axis, y-axis and z-axis, respectively, and to coincide vertex $\mathbf{v}_a$ with the origin of the Cartesian coordinate system. During mesh preprocessing, transformation matrix $\mathbf{T}$ is generated by

$$\mathbf{T} = \left[\overrightarrow{\mathbf{v}_a\mathbf{v}_b}, \overrightarrow{\mathbf{v}_a\mathbf{v}_c}, \overrightarrow{\mathbf{v}_a\mathbf{v}_b} \times \overrightarrow{\mathbf{v}_a\mathbf{v}_c}\right], \tag{22}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

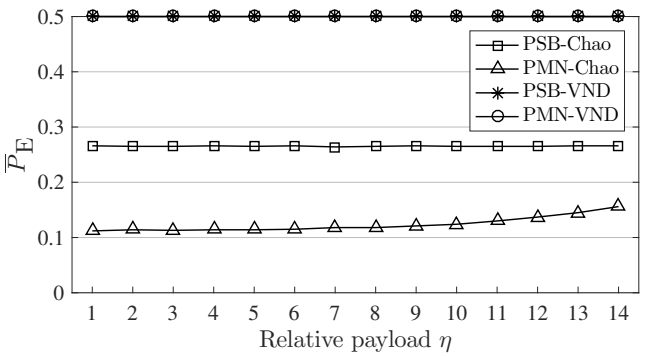


Fig. 14: Average testing error $\overline{P}_{\mathrm{E}}$ of targeted attack on Chao and VND under varying payloads with PSB and PMN dataset, respectively.

and thus the rotated mesh $\mathcal{V}_T$ is acquired by

$$\mathcal{V}_T = \mathbf{T}^T \cdot \mathcal{V}. \tag{23}$$

Experimental setup is described as follows: γ is integer so as to share the same meaning with the hiding layers in paper [13]. Because that the interval $I_w/2^\gamma$ must be greater than or equal to 2^{-23}, different from the parameter deployments that fix I_w to 10,000 in [33], in order to align the relative embedding payload to our proposed method, the interval width I_w is adaptively determined by the choice of γ, $I_w = 2^{\gamma-23}$. During the embedding, all the vertices in the mesh are carrying payloads, except for three vertices which are considered as the bases for the extraction process.

B. Targeted attack

Owing to the stego mesh with its first and second principle axes close to x-axis and y-axis correspondingly, such steganography with behavior disorder causes suspicion of attackers. Our detector algorithm is based on the observation that the transformation-based preprocessing of Chao's algorithm results in a variance of position between cover mesh and stego mesh, which is prone to be analyzed by some well-designed features and classifiers. A stego mesh has factitious position and its transformation matrix after PCA tends to be close to identity matrix $\mathbf{I} = \begin{bmatrix} 1,0,0 \\ 0,1,0 \\ 0,0,1 \end{bmatrix}$ while the matrix of a cover mesh is distant from $\mathbf{I}$ on most occasions. Specifically, the unidimensional feature is defined by the ℓ_1 norm between two matrices as

$$f_m = \|\mathbf{T} - \mathbf{I}\|_1. \tag{24}$$

Note that we have also tried cosine distance $(\sum_{j=1}^{3} \arccos(\mathbf{T}_j, \mathbf{I}_j))$ and ℓ_2 distance $(\|\mathbf{T} - \mathbf{I}\|_2)$ between two matrices as features where the performance does not exceed ℓ_1 norm measurement.

The steganalysis of Chao's method is evaluated empirically using binary classifiers trained on a given cover mesh and its stego version embedded with a fixed relative payload. Five-fold cross validation of Support Vector Machine (SVM) is employed to conduct training and classification. Each test is repeated 10 times, and results are averaged to evaluate the final performance. Soft-margin SVMs with the Gaussian kernel $k(x, y) = \exp(-\gamma_k \|x - y\|_2^2)$, $\gamma_k > 0$ is used. The values of the penalization parameter $C = 5$ and the kernel parameter $\gamma_k = 0.5$. Our experiments show that Radial Basis Function (RBF) SVM has competitive results, and LIBSVM [44] is utilized here as the classifier for low computing complexity.

We demonstrate a thorough experimental evaluation on the PSB and PMN dataset for targeted steganalysis on Chao and VND algorithms, as shown in Figure 14. Inspired by how steganalysis features are built by rotation preprocessing of meshes, we explored the difference between cover and stego meshes of Chao's algorithm. It is clear that the average testing error of Chao method is nearly a constant with 0.265 and 0.124 for PSB and PMN, respectively, while the results of two VND methods are approximately 0.5, informing that the defect of Chao's algorithm leaks the state of the meshes whether they are data-embedded.

VI. Conclusion

In this paper, we propose a new scheme on adaptive steganography in 3D meshes. The core of our approach is the design of distortion function, which utilizes vertex normal as a criteria after observing the effects of features and is called Vertex Normal Distortion (VND). Multiple bitplanes are modified to embed messages, from which the highest bitplane is adaptively data-embedded and other bitplanes are conducted with LSBR. Since the bitplanes are adequately utilized, the capacity of VND method increases greatly. The experiment results show that the security of VND based method outperforms other state-of-the-art methods. We also point out the deficiency of Chao's algorithm and implement a targeted attack.

We have discovered that the mutual dependencies among the three components of vertices are strong enough to affect the security of steganography, and better payload distribution can be made. To have a better security performance, we will try to generalize the cost function to steganography that minimizes a non-additive distortion function and distribute the total messages into several bitplanes following roles with the best security in our future study.

Appendix A

Numerical values of $\overline{P}_{\mathrm{E}}$ of Figure 8 are provided in Table II, and Table III, results of Figure 9 are provided in Table IV and Table V, and results of Figure 10 in Table VI and Table VII.

References

[1] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in *Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505. International Society for Optics and Photonics, 2007, p. 650502.

[2] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[3] T. Pevnỳ, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *International Workshop on Information Hiding*. Springer, 2010, pp. 161–177.

[4] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on*. IEEE, 2012, pp. 234–239.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

TABLE II
DETECTABILITY IN TERMS OF $\overline{P}_E$ VERSUS INTEGRAL EMBEDDING PAYLOAD SIZE IN BITS PER VERTEX (BPV) FOR PROPOSED VND AND PRIOR ART ON PRINCETON SEGMENTATION BENCHMARK (PSB) USING THE FLD ENSEMBLE CLASSIFIER WITH TWO FEATURE SETS.

Feature	Embedding method	1	2	3	4	5	6	7
LFS64	Chao	.2553 ± .0229	.2447 ± .0235	.2447 ± .0224	.2447 ± .0224	.2420 ± .0210	.2340 ± .0239	.2287 ± .0227
	Li	.2527 ± .0248	.2553 ± .0228	.2447 ± .0219	.2394 ± .0199	.2367 ± .0202	.2287 ± .0212	.2181 ± .0249
	HPQ	.1860 ± .0211	.1842 ± .0212	.1811 ± .0214	.1722 ± .0231	.1664 ± .0231	.1614 ± .0228	.1621 ± .0217
	LSBR/VND	.5080 ± .0358	.5080 ± .0358	.3218 ± .0231	.2394 ± .0214	.2394 ± .0196	.2340 ± .0174	.2287 ± .0230
LFS76	Chao	.3457 ± .0220	.3511 ± .0212	.3404 ± .0226	.3298 ± .0198	.3138 ± .0189	.2819 ± .0239	.2553 ± .0233
	Li	.3590 ± .0222	.3537 ± .0246	.3298 ± .0190	.3138 ± .0169	.2926 ± .0227	.2580 ± .0229	.2420 ± .0255
	HPQ	.2946 ± .0315	.3015 ± .0318	.2978 ± .0241	.2857 ± .0174	.2603 ± .0111	.2396 ± .0184	.2016 ± .0201
	LSBR/VND	.5080 ± .0358	.5080 ± .0358	.3617 ± .0200	.3245 ± .0163	.2872 ± .0174	.2686 ± .0199	.2247 ± .0238
Feature	Embedding method	8	9	10	11	12	13	14
LFS64	Chao	.2181 ± .0213	.2128 ± .0218	.1862 ± .0223	.1596 ± .0183	.1170 ± .0197	.0638 ± .0163	.0266 ± .0092
	Li	.2128 ± .0193	.1968 ± .0201	.1968 ± .0252	.1596 ± .0158	.1064 ± .0169	.0532 ± .0147	.0160 ± .0078
	HPQ	.1588 ± .0221	.1467 ± .0211	.1269 ± .0202	.1053 ± .0121	.0611 ± .0111	.0135 ± .0142	.0101 ± .0015
	LSBR/VND	.2128 ± .0227	.2074 ± .0203	.1888 ± .0231	.1516 ± .0136	.0904 ± .0135	.0266 ± .0160	.0106 ± .0048
LFS76	Chao	.2314 ± .0256	.2181 ± .0255	.2021 ± .0216	.1649 ± .0204	.1223 ± .0202	.0585 ± .0173	.0266 ± .0127
	Li	.2207 ± .0214	.2101 ± .0274	.1968 ± .0210	.1729 ± .0179	.1170 ± .0179	.0612 ± .0156	.0133 ± .0064
	HPQ	.1838 ± .0261	.1770 ± .0174	.1593 ± .0174	.1140 ± .0141	.0935 ± .0148	.0348 ± .0157	.0100 ± .0041
	LSBR/VND	.2394 ± .0254	.2314 ± .0192	.2128 ± .0198	.1543 ± .0172	.0904 ± .0188	.0319 ± .0117	.0106 ± .0047

TABLE III
DETECTABILITY IN TERMS OF $\overline{P}_E$ VERSUS INTEGRAL EMBEDDING PAYLOAD SIZE IN BITS PER VERTEX (BPV) FOR PROPOSED VND AND PRIOR ART ON PRINCETON MODELNET DATABASE (PMN) USING THE FLD ENSEMBLE CLASSIFIER WITH TWO FEATURE SETS.

Feature	Embedding method	1	2	3	4	5	6	7
LFS64	Chao	.3311 ± .0052	.3256 ± .0052	.3046 ± .0058	.2835 ± .0055	.2792 ± .0049	.2680 ± .0058	.2681 ± .0056
	Li	.3486 ± .0058	.3496 ± .0056	.2889 ± .0056	.2721 ± .0057	.2698 ± .0057	.2641 ± .0056	.2575 ± .0054
	HPQ	.2502 ± .0081	.2433 ± .0079	.2273 ± .0035	.2158 ± .0061	.2067 ± .0039	.1950 ± .0041	.1950 ± .0048
	LSBR/VND	.4986 ± .0083	.4986 ± .0083	.3425 ± .0041	.2850 ± .0053	.2786 ± .0040	.2722 ± .0044	.2644 ± .0050
LFS76	Chao	.3710 ± .0050	.3633 ± .0053	.3518 ± .0052	.3209 ± .0047	.2995 ± .0043	.2916 ± .0052	.2818 ± .0052
	Li	.3755 ± .0048	.3765 ± .0050	.3250 ± .0049	.3001 ± .0051	.2909 ± .0050	.2863 ± .0065	.2757 ± .0071
	HPQ	.3283 ± .0082	.3185 ± .0082	.3058 ± .0037	.2850 ± .0035	.2773 ± .0043	.2589 ± .0052	.2491 ± .0060
	LSBR/VND	.4986 ± .0082	.4986 ± .0082	.3635 ± .0037	.3155 ± .0035	.3070 ± .0043	.2998 ± .0052	.2889 ± .0060
Feature	Embedding method	8	9	10	11	12	13	14
LFS64	Chao	.2499 ± .0061	.2372 ± .0055	.2226 ± .0069	.2020 ± .0067	.1796 ± .0075	.1507 ± .0067	.1291 ± .0064
	Li	.2475 ± .0060	.2313 ± .0060	.2129 ± .0059	.1950 ± .0051	.1698 ± .0051	.1389 ± .0045	.1119 ± .0034
	HPQ	.1792 ± .0042	.1732 ± .0055	.1689 ± .0051	.1592 ± .0053	.1496 ± .0045	.1360 ± .0040	.1186 ± .0031
	LSBR/VND	.2565 ± .0051	.2430 ± .0068	.2284 ± .0057	.2066 ± .0058	.1831 ± .0042	.1514 ± .0041	.1232 ± .0036
LFS76	Chao	.2680 ± .0063	.2549 ± .0068	.2351 ± .0076	.2122 ± .0062	.1845 ± .0064	.1542 ± .0061	.1244 ± .0056
	Li	.2627 ± .0062	.2491 ± .0078	.2279 ± .0071	.2015 ± .0056	.1658 ± .0042	.1303 ± .0033	.0943 ± .0038
	HPQ	.2405 ± .0064	.2158 ± .0065	.2067 ± .0065	.1854 ± .0054	.1632 ± .0046	.1421 ± .0036	.1193 ± .0037
	LSBR/VND	.2786 ± .0074	.2620 ± .0069	.2421 ± .0067	.2149 ± .0059	.1804 ± .0042	.1442 ± .0032	.1027 ± .0034

TABLE IV
DETECTABILITY IN TERMS OF $\overline{P}_E$ VERSUS EMBEDDING PAYLOAD SIZE IN BITS PER VERTEX (BPV) ON 7 LAYERS FOR PROPOSED VND AND PRIOR ART ON PRINCETON SEGMENTATION BENCHMARK (PSB) USING THE FLD ENSEMBLE CLASSIFIER WITH TWO FEATURE SETS.

Feature	Embedding method	0	0.1	0.2	0.3	0.4	0.5
LFS64	LSBR	.2287 ± .0197	.2314 ± .0212	.2314 ± .0208	.2314 ± .0251	.2261 ± .0247	.2287 ± .0262
	VND	.2340 ± .0201	.2340 ± .0201	.2314 ± .0197	.2287 ± .0243	.2287 ± .0240	.2314 ± .0230
LFS76	LSBR	.2660 ± .0245	.2713 ± .0203	.2633 ± .0253	.2606 ± .0224	.2553 ± .0191	.2553 ± .0262
	VND	.2713 ± .0215	.2660 ± .0197	.2660 ± .0238	.2606 ± .0221	**.2686 ± .0240**	.2606 ± .0217
	Embedding method	0.6	0.7	0.8	0.9	1.0	
LFS64	LSBR	.2340 ± .0213	.2261 ± .0189	.2287 ± .0241	.2234 ± .0223	.2234 ± .0195	
	VND	.2340 ± .0182	.2287 ± .0218	.2287 ± .0269	.2314 ± .0243	.2340 ± .0266	
LFS76	LSBR	.2500 ± .0252	.2500 ± .0223	.2500 ± .0227	.2527 ± .0227	.2394 ± .0255	
	VND	.2606 ± .0237	.2580 ± .0238	**.2660 ± .0272**	.2553 ± .0254	.2500 ± .0232	

TABLE V
DETECTABILITY IN TERMS OF $\overline{P}_E$ VERSUS EMBEDDING PAYLOAD SIZE IN BITS PER VERTEX (BPV) ON 12 LAYERS FOR PROPOSED VND AND PRIOR ART ON PRINCETON SEGMENTATION BENCHMARK (PSB) USING THE FLD ENSEMBLE CLASSIFIER WITH TWO FEATURE SETS.

Feature	Embedding method	0	0.1	0.2	0.3	0.4	0.5
LFS64	LSBR	.1516 ± .0233	.1489 ± .0203	.1436 ± .0157	.1356 ± .0221	.1277 ± .0169	.1277 ± .0154
	VND	.1463 ± .0156	.1436 ± .0187	.1463 ± .0164	**.1516 ± .0191**	**.1436 ± .0198**	**.1383 ± .0183**
LFS76	LSBR	.1516 ± .0217	.1569 ± .0246	.1543 ± .0191	.1436 ± .0241	.1383 ± .0207	.1330 ± .0187
	VND	.1602 ± .0176	.1569 ± .0202	.1543 ± .0223	**.1569 ± .0260**	**.1489 ± .0197**	**.1436 ± .0228**
Feature	Embedding method	0.6	0.7	0.8	0.9	1.0	
LFS64	LSBR	.1223 ± .0205	.1117 ± .0170	.1170 ± .0196	.1064 ± .0175	.1117 ± .0157	
	VND	.1303 ± .0171	**.1277 ± .0185**	.1223 ± .0128	**.1223 ± .0199**	.1117 ± .0171	
LFS76	LSBR	.1277 ± .0194	.1277 ± .0171	.1170 ± .0192	.1170 ± .0181	.1223 ± .0189	
	VND	**.1383 ± .0211**	.1330 ± .0238	**.1383 ± .0192**	**.1330 ± .0170**	.1170 ± .0177	

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2882088, IEEE Transactions on Multimedia

TABLE VI
DETECTABILITY IN TERMS OF $\overline{P}_E$ VERSUS EMBEDDING PAYLOAD SIZE IN BITS PER VERTEX (BPV) ON 7 LAYERS FOR PROPOSED VND AND PRIOR ART ON PRINCETON MODELNET DATABASE (PMN) USING THE FLD ENSEMBLE CLASSIFIER WITH TWO FEATURE SETS.

Feature	Embedding method	0	0.1	0.2	0.3	0.4	0.5
	LSBR	.2820 ± .0052	.2761 ± .0045	.2731 ± .0058	.2705 ± .0048	.2689 ± .0048	.2677 ± .0043
LFS64	VND	.2824 ± .0045	**.2833 ± .0049**	**.2786 ± .0042**	**.2739 ± .0053**	**.2724 ± .0043**	**.2721 ± .0043**
	LSBR	.3076 ± .0053	.3021 ± .0051	.3000 ± .0067	.2975 ± .0054	.2959 ± .0065	.2954 ± .0049
LFS76	VND	.3077 ± .0049	.3044 ± .0046	**.3043 ± .0047**	**.3038 ± .0050**	**.3010 ± .0054**	**.2995 ± .0056**
	Embedding method	0.6	0.7	0.8	0.9	1.0	
	LSBR	.2674 ± .0055	.2660 ± .0050	.2654 ± .0055	.2644 ± .0041	.2639 ± .0041	
LFS64	VND	**.2710 ± .0047**	**.2698 ± .0053**	**.2695 ± .0033**	**.2687 ± .0051**	.2632 ± .0049	
	LSBR	.2919 ± .0058	.2888 ± .0052	.2869 ± .0060	.2860 ± .0043	.2913 ± .0058	
LFS76	VND	**.2969 ± .0055**	**.2978 ± .0055**	**.2948 ± .0054**	**.2909 ± .0054**	.2904 ± .0055	

TABLE VII
DETECTABILITY IN TERMS OF $\overline{P}_E$ VERSUS EMBEDDING PAYLOAD SIZE IN BITS PER VERTEX (BPV) ON 12 LAYERS FOR PROPOSED VND AND PRIOR ART ON PRINCETON MODELNET DATABASE (PMN) USING THE FLD ENSEMBLE CLASSIFIER WITH TWO FEATURE SETS.

Feature	Embedding method	0	0.1	0.2	0.3	0.4	0.5
	LSBR	.2054 ± .0040	.1978 ± .0041	.1949 ± .0046	.1913 ± .0050	.1892 ± .0042	.1885 ± .0051
LFS64	VND	.2036 ± .0045	**.2008 ± .0046**	**.1988 ± .0048**	**.1966 ± .0042**	**.1964 ± .0045**	**.1944 ± .0046**
	LSBR	.2060 ± .0046	.2000 ± .0045	.1969 ± .0043	.1929 ± .0033	.1939 ± .0045	.1888 ± .0044
LFS76	VND	.2051 ± .0035	.2008 ± .0041	**.2010 ± .0049**	**.1973 ± .0042**	**.1989 ± .0048**	**.1939 ± .0044**
Feature	Embedding method	0.6	0.7	0.8	0.9	1.0	
	LSBR	.1885 ± .0045	.1851 ± .0048	.1832 ± .0048	.1847 ± .0045	.1840 ± .0053	
LFS64	VND	**.1944 ± .0044**	**.1916 ± .0046**	**.1869 ± .0043**	.1870 ± .0053	.1821 ± .0053	
	LSBR	.1862 ± .0041	.1835 ± .0040	.1830 ± .0045	.1802 ± .0050	.1794 ± .0042	
LFS76	VND	**.1941 ± .0047**	**.1895 ± .0042**	**.1888 ± .0043**	**.1862 ± .0045**	.1796 ± .0048	

[5] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 1, 2014.

[6] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4206–4210.

[7] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, 2016.

[8] K. Chen, W. Zhang, H. Zhou, N. Yu, and G. Feng, "Defining cost functions for adaptive steganography at the microscale," in *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*. IEEE, 2016, pp. 1–6.

[9] L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient jpeg steganography," *IEEE transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 814–825, 2014.

[10] L. Guo, J. Ni, W. Su, C. Tang, and Y.-Q. Shi, "Using statistical image model for jpeg steganography: uniform embedding revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2669–2680, 2015.

[11] J. Kodovskỳ, T. Pevnỳ, and J. Fridrich, "Modern steganalysis can detect yass," in *Media Forensics and Security II*, vol. 7541. International Society for Optics and Photonics, 2010, p. 754102.

[12] J. Fridrich, M. Goljan, and R. Du, "Detecting lsb steganography in color, and gray-scale images," *IEEE multimedia*, vol. 8, no. 4, pp. 22–28, 2001.

[13] A. D. Ker, "Steganalysis of lsb matching in grayscale images," *IEEE signal processing letters*, vol. 12, no. 6, pp. 441–444, 2005.

[14] S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," in *International Workshop on Information Hiding*. Springer, 2002, pp. 340–354.

[15] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[16] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[17] M. Centin and A. Signoroni, "Mesh denoising with (geo) metric fidelity," *IEEE transactions on visualization and computer graphics*, 2017.

[18] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "A comprehensive survey on three-dimensional mesh watermarking," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1513–1527, 2008.

[19] B. Vasic and B. Vasic, "Simplification resilient ldpc-coded sparse-qim watermarking for 3d-meshes," *IEEE Transactions on Multimedia*, vol. 15, no. 7, pp. 1532–1542, 2013.

[20] X. Rolland-Neviere, G. Doërr, and P. Alliez, "Triangle surface mesh watermarking based on a constrained optimization framework," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 9, pp. 1491–1501, 2014.

[21] R. Jiang, H. Zhou, W. Zhang, and N. Yu, "Reversible data hiding in encrypted three-dimensional mesh models," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 55–67, 2018.

[22] F. Cayre and B. Macq, "Data hiding on 3-d triangle meshes," *IEEE Transactions on signal Processing*, vol. 51, no. 4, pp. 939–949, 2003.

[23] C.-M. Wang and Y.-M. Cheng, "An efficient information hiding algorithm for polygon models," in *Computer Graphics Forum*, vol. 24, no. 3. Wiley Online Library, 2005, pp. 591–600.

[24] Y.-M. Cheng and C.-M. Wang, "A high-capacity steganographic approach for 3d polygonal meshes," *The Visual Computer*, vol. 22, no. 9-11, pp. 845–855, 2006.

[25] P. Amat, W. Puech, S. Druon, and J.-P. Pedeboy, "Lossless 3d steganography based on mst and connectivity modification," *Signal Processing: Image Communication*, vol. 25, no. 6, pp. 400–412, 2010.

[26] V. Itier, W. Puech, G. Gesquière, and J.-P. Pedeboy, "Joint synchronization and high capacity data hiding for 3d meshes," in *Three-Dimensional Image Processing, Measurement (3DIPM), and Applications 2015*, vol. 9393. International Society for Optics and Photonics, 2015, p. 939305.

[27] M.-W. Chao, C.-h. Lin, C.-W. Yu, and T.-Y. Lee, "A high capacity 3d steganography algorithm," *IEEE transactions on visualization and computer graphics*, vol. 15, no. 2, pp. 274–284, 2009.

[28] Y. Yang, N. Peyerimhoff, and I. Ivrissimtzis, "Linear correlations between spatial and normal noise in triangle meshes," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 1, pp. 45–55, 2013.

[29] V. Itier and W. Puech, "High capacity data hiding for 3d point clouds based on static arithmetic coding," *Multimedia Tools and Applications*, vol. 76, no. 24, pp. 26 421–26 445, 2017.

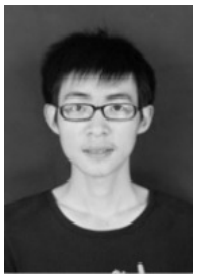
[30] N. Li, J. Hu, R. Sun, S. Wang, and Z. Luo, "A high-capacity 3d steganography algorithm with adjustable distortion," *IEEE Access*, vol. 5, pp. 24 457–24 466, 2017.

[31] Z. Li, S. Beugnon, W. Puech, and A. G. Bors, "Rethinking the high capacity 3d steganography: Increasing its resistance to steganalysis," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 510–514.

[32] Y. Yang and I. Ivrissimtzis, "Mesh discriminative features for 3d steganalysis," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 10, no. 3, p. 27, 2014.

[33] Z. Li and A. G. Bors, "3d mesh steganalysis using local shape features," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2144–2148.

[34] Y. Yang and I. Ivrissimtzis, "Polygonal mesh watermarking using laplacian coordinates," in *Computer Graphics Forum*, vol. 29, no. 5. Wiley Online Library, 2010, pp. 1585–1593.
[35] Z. Li and A. G. Bors, "Steganalysis of 3d objects using statistics of local feature sets," *Information Sciences*, vol. 415, pp. 85–99, 2017.
[36] 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 *International Conference on Information Science and Applications*. Springer, 2017, pp. 358–365.
[37] G. Taubin, "A signal processing approach to fair surface design," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 351–358.
[38] J. Rugis and R. Klette, "A scale invariant surface curvature estimator," in *Pacific-Rim Symposium on Image and Video Technology*. Springer, 2006, pp. 138–147.
[39] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 3, pp. 390–395, 2006.
[40] N. Max, "Weights for computing vertex normals from facet normals," *Journal of Graphics Tools*, vol. 4, no. 2, pp. 1–6, 1999.
[41] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3d mesh segmentation," in *Acm transactions on graphics (tog)*, vol. 28, no. 3. ACM, 2009, p. 73.
[42] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.
[43] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.
[44] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.



Hang Zhou received the B.S. degree in School of Communication and Information Engineering, Shanghai University, in 2015. He is currently pursuing the Ph.D. degree in Information Security in University of Science and Technology of China (USTC). His research interests include information hiding, image processing and computer graphics.

Kejiang Chen received the B.S. degree in School of Communication and Information Engineering, Shanghai University, in 2015. He is currently pursuing the Ph.D. degree in Information Security in University of Science and Technology of China (USTC). His research interests include information hiding, image processing and deep learning.

Weiming Zhang received his M.S. degree and Ph.D. degree in 2002 and 2005 respectively from the Zhengzhou Information Science and Technology Institute, P.R. China. Currently, he is a professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include information hiding and multimedia security.

Yuanzhi Yao received his Ph.D. degree in electronic engineering from the University of Science and Technology of China in 2017, where he is currently a postdoctoral researcher. His research interests include information hiding and video coding.

Nenghai Yu received his B.S. degree in 1987 from Nanjing University of Posts and Telecommunications, M.E. degree in 1992 from Tsinghua University and Ph.D. degree in 2004 from University of Science and Technology of China, where he is currently a professor. His research interests include multimedia security, multimedia information retrieval, video processing and information hiding.