

# Feature-Preserving Tensor Voting Model for Mesh Steganalysis

Hang Zhou<sup>1</sup>, Kejiang Chen<sup>1</sup>, Weiming Zhang<sup>1</sup>, Chuan Qin, and Nenghai Yu<sup>1</sup>

**Abstract**—The standard tensor voting technique shows its versatility in tasks such as object recognition and semantic segmentation by recognizing feature points and sharp edges that can segment a model into several patches. We propose a neighborhood-level representation-guided tensor voting model for 3D mesh steganalysis. Because existing steganalytic methods do not analyze correlations among neighborhood faces, they are not very effective at discriminating stego meshes from cover meshes. In this paper, we propose to utilize a tensor voting model to reveal the artifacts caused by embedding data. In the proposed steganalytic scheme, the normal voting tensor (NVT) operation is performed on original mesh faces and smoothed mesh faces separately. Then, the absolute values of the differences between the eigenvalues of the two tensors (from the original face and the smoothed face) are regarded as features that capture intricate relationships among the vertices. Subsequently, the extracted features are processed with a nonlinear mapping to boost the feature effectiveness. The experimental results show that the proposed feature sets prevail over state-of-the-art feature sets including LFS64 and ELFS124 under various steganographic schemes.

**Index Terms**—Mesh steganography, mesh steganalysis, normal voting tensor, feature extraction, ensemble classifier

## 1 INTRODUCTION

THREE-DIMENSIONAL (3D) technology is already mainstream for consumers, from virtual reality (VR), visual effects (VFX), 3D printing, animated movies, and video games to web integration, Facebook support and much more [1]. The growing number of 3D techniques facilitates the rapid progression of 3D-related applications, including 3D mesh watermarking and 3D mesh steganography. Whereas 3D watermarking focuses on protecting copyright ownership and reducing the counterfeiting of digital multimedia, 3D mesh steganography focuses on covert communication against steganalysis. The procedure of steganography targets the communications between two parties over covert channels, where the sender hides the message inside an innocuous-looking cover object using a steganographic method driven by a secret shared with the recipient such that a potential eavesdropper cannot detect its existence; 3D meshes are appropriate carriers for steganography.

Recently, 3D mesh steganography technologies have been actively investigated due to the rapid expansion of 3D techniques, and they can be mainly classified into two categories: low-capacity [2], [3], [4], [5], [6] and high-capacity steganography [7], [7], [8], [9], [10], [11], [12]. Correspondingly, to detect

whether a mesh contains hidden data, 3D mesh steganalysis algorithms [13], [14], [15], [16], [17] are being developed.

In this paper, we propose a new set of steganalytic features based on an adapted version of the normal tensor voting model that can measure the smoothness of the local region and the correlations of adjacent vertices. In contrast to previous feature sets such as LFS64 [16] and ELFS124 [17], the new feature extraction method contains three phases. First, to capture the local shape characteristics, normal voting tensors (NVTs) of each face are proposed with three diverse neighbor definitions. The process of extracting the normal voting tensor is similar to the process of calculating the side length or the vertex curvature, and it represents the essential attribute of a mesh. Second, three eigenvalues of each tensor are calculated, and they mirror the innate structural characteristics such as edges, flat faces or corners. Third, nonlinear mappings and several statistical moments are executed successively.

Whereas normal meshes have a strong neighboring relation, stego meshes have an inferior neighboring correlation because steganographic modifications partially destroy neighboring correlations. Moreover, our NVT features can measure the neighboring correlation more effectively than the existing methods. Therefore, we can effectively distinguish cover and stego meshes. Additionally, NVT features are extracted from adjacent faces, while existing features are extracted from single vertices, single edges or single faces. Therefore, the features are designed from a different perspective. When the NVT features are combined with other features such as LFS64, better performance can be obtained.

The rest of this paper is organized as follows. After introducing the basic notation and terminology in Section 2, we review the mesh steganalysis framework and prior mesh steganalytic features. In Section 3, we propose several new

• H. Zhou, K. Chen, W. Zhang, 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, chenkj}@mail.ustc.edu.cn, {zhangwm, ynh}@ustc.edu.cn.

• C. Qin is with the Information Security, University of Science and Technology of China, Hefei 230026, China. E-mail: qc94@mail.ustc.edu.cn.

Manuscript received 23 Nov. 2018; revised 30 June 2019; accepted 11 July 2019. Date of publication 16 July 2019; date of current version 24 Nov. 2020.

(Corresponding author: Weiming Zhang.)

Recommended for acceptance by T. Ju.

Digital Object Identifier no. 10.1109/TVCG.2019.2929041

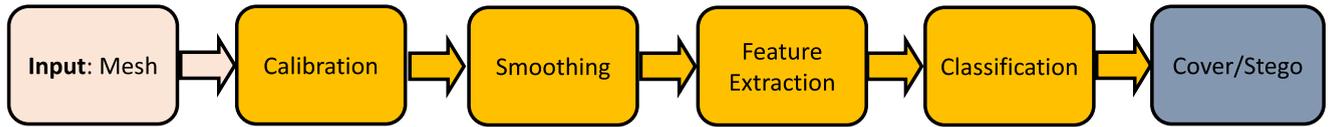


Fig. 1. The 3D mesh steganalysis framework based on learning from the statistics of residual features and classification via ensemble classifiers.

mesh features based on the normal voting tensor. The results of our experiments are detailed in Section 4 to demonstrate the effectiveness of the proposed schemes. The conclusion and future work are given in Section 5.

## 2 RELATED WORK

In this paper, capital and lowercase boldface symbols stand for matrices and vectors, respectively. We work with meshes  $\mathcal{M} = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$ . 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 the Cartesian coordinate system. Let  $\mathcal{E}$  be the edge set and  $\mathcal{F}$  be the face set.

### 2.1 3D Mesh Steganography

Cayre and Marcq [2] embed data by modulating the position of the orthogonal projection of a triangle summit on the opposite side. Wang and Cheng [3] improve the modulation by sliding, extending and rotating levels to embed data; they also combine both the spatial domain and the representation domain [4] to increase the embedding capacity. Other follow-ups with small embedding capacities mainly focus on perfecting visual distortions caused by modifications [5], [6].

High-capacity based 3D mesh steganography can be classified into two categories: distortionless steganography based on order permutation [18], [19], [20], and distorted steganography based on vertex shifting [7], [8], [9], [10], [11], [12]. For distortionless steganography, Bogomjakov et al. [18] propose to hide messages in the indexed representation of a mesh by permuting the order in which faces and vertices are stored. Tan et al. [19] consider the vertex index as a message block, and propose three embedding strategies: vertex index embedding, dynamic-length bit-string mapping, and repeated bits embedding to enlarge embedding capacity. Tu and Tai [20] propose to use maximum expected level tree based on a message probability model to embed data, which owns larger capacity than previous methods. For distorted steganography, Chao et al. [7] provide multilayered high-capacity reversible steganography with space modulation and demodulation techniques on the principal axes by vertex projection. Yang et al. [8] embed data by modifying the LSBs of selected vertex coordinates, where the embedding capacity depends on the shape of the mesh and cannot be known in advance. Itier and Puech [9] propose a steganographic method that hides data by the displacement of a vertex relative to its new position in the Hamiltonian path using static arithmetic coding. Li et al. [10] propose a key-modulation-based steganography with confined distortion. Li et al. [11] increase the resistance to steganalysis by means of the mesh steganography of [9]. Zhou et al. [12] put forward adaptive steganography together with coding techniques to enhance the security performance. The work also points out the defects of [7] and comes up with a targeted attack.

### 2.2 3D Mesh Steganalysis

To date, modern feature-based steganalysis [14], [21] on meshes has started by adopting a mesh model within which steganalyzers are built using machine learning classifiers [22], [23] operating on high-dimensional features. Fig. 1 shows the pipeline of the 3D mesh steganalysis framework based on learning from the statistics of residual features and classifications via ensemble classifiers, which includes calibration, smoothing, feature extraction and feature mapping.

#### 2.2.1 Calibration and Smoothing

Before feature extraction, it is necessary to preprocess vertices into the canonical version: the mesh object is rotated and aligned according to its first and second principal axes, which are given by the principal component analysis (PCA) algorithm. The guidance from image steganalysis indicates that the distinction between a stego image and its smoothed image is more significant than the distinction between a cover image and its smoothed image [24], [25]; Analogically, it is expected that the differences follow the same rule. The smoothed reference mesh  $\mathcal{M}'$  is produced by applying the umbrella operator or one iteration of Laplacian smoothing on the original mesh  $\mathcal{M}$ , which changes the vertex  $\mathbf{v}_i$  to  $\mathbf{v}'_i$  for  $i = 1, \dots, N$ , as follows [26]:

$$\mathbf{v}'_i = \mathbf{v}_i + \frac{\tau}{\sum_{\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)} w_{ij}} \sum_{\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)} w_{ij} (\mathbf{v}_j - \mathbf{v}_i), \quad (1)$$

where  $\tau$  is a scalar factor and  $w_{ij}$  is the weight defined by

$$w_{ij} = \begin{cases} 1 & \text{if } \mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i) \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

#### 2.2.2 Existing Features

Artifacts generated by steganography could be identified by the effective features. Moreover, these features are designated by the differences between the mesh object and its smoothed version. Detectors of such features range from simple vertex displacement measurements to algorithms that consider the local neighborhoods and the specific shape characteristics [15]. All of the syntaxes of the features in this paper follow the convention  $name = \{\phi\}_{\{\#\}}$ , where  $\phi$  represents the feature and  $\#$  is the sequence number; these features jointly constitute the multidimensional feature vector  $\Phi$ .

*YANG208 Features* [21]. Yang and Ivriissimtzis propose the first steganalytic algorithm for triangle meshes. For the vertex vectors, the absolute values of the differences of the  $xyz$  coordinates of  $\mathcal{M}$  and  $\mathcal{M}'$  as well as the lengths of the vector of the Cartesian coordinate differences are computed. Next, four more vectors are obtained by computing the same absolute differences but on the Laplacian rather than the Cartesian coordinates of  $\mathcal{M}$  and  $\mathcal{M}'$ . The computations are made

separately on vertices with valences less than, equal to, or greater than 6; all of the boundary vertices are excluded. The total number of vectors obtained from the mesh vertices is 24. For the edges of the mesh, one computes the vector of the absolute values of the differences of the dihedral angles between  $\mathcal{M}$  and  $\mathcal{M}'$ . Finally, for the mesh faces, the vector of the angles between the normals of  $\mathcal{M}$  and  $\mathcal{M}'$  are computed. The result is a total of 26 vectors from the vertices, edges, and faces. From each of the 26 vectors calculated in the previous section, 8 components of the feature vector are computed to create a vector of dimension 208 denoted by  $\Phi_{208}$ .

*YANG40 Features* [14]. Li and Bors propose the 40D feature vector YANG40, and it contains the most effective features from YANG208. The first 6 components  $\phi_1 - \phi_6$  represent the absolute distance, which is measured along each coordinate axis of  $xyz$  between the locations of the vertices of the meshes  $\mathcal{M}$  and  $\mathcal{M}'$  after being normalized and aligned in both the Cartesian and Laplacian coordinate systems. Next, the changes produced in the euclidean distance between the vertex location and the center of the object, which represent the vertex norms, are denoted by  $\phi_7$  and  $\phi_8$ .  $\phi_9$  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 the neighboring faces and is calculated on the plane perpendicular to the common edge.

Similarly, the available features are extracted from faces. Changes in the local surface orientation are measured by calculating the angle between the surface normals  $\mathbf{n}_{f_i}$  of the faces from the object  $f_i \in \mathcal{F}$  and their correspondents  $\mathbf{n}'_{f_i}$  from the smoothed object  $f'_i \in \mathcal{F}'$ . The absolute value of the angles between the two face normals is computed as follows:

$$\phi_{10}(i) = \arccos \frac{\mathbf{n}_{f_i} \cdot \mathbf{n}'_{f_i}}{\|\mathbf{n}_{f_i}\| \cdot \|\mathbf{n}'_{f_i}\|}, \quad i = 1, \dots, |\mathcal{F}|. \quad (3)$$

Note that  $\mathcal{F} = \mathcal{F}'$ , as the smoothing operation does not change the topologies among the vertices. The total feature vector of LFS40 is  $\Phi_{40} = \{\phi_1, \phi_2, \dots, \phi_{10}\}$ .

*LFS52 Features* [14]. Li and Bors propose the local feature set (LFS) based on the LFS52 features, which include the YANG40 features ( $\phi_1 \sim \phi_{10}$ ) and several local shape features ( $\phi_{11} \sim \phi_{13}$ ).  $\phi_{11}$  is the angle between the vertex normals of each of the two corresponding vertices, where vertex normal is defined by the weighted sum of the normals of the faces that contain that vertex

$$\mathbf{n}_{v_i} = \sum_{f_j \in \mathcal{F}(v_i^*)} \frac{A(f_j) \cdot \mathbf{n}_{f_j}}{\|e_{(v_1, v_2)}\| \cdot \|e_{(v_1, v_2)}\|}, \quad (4)$$

where  $\mathcal{F}(v_i^*)$  is the set of faces that contain the vertex  $v_i$ ;  $v'_i$  and  $v''_i$  are the two vertices adjacent to vertex  $v_i$  on face  $f_j$ ;  $e_{(v_1, v_2)}$  represents the edge that connects vertices  $v_1$  and  $v_2$ ; and  $A(f_j)$  represents the area of triangle  $f_j$ .

Thus, the absolute value of the angles between the two vertex normals is computed by

$$\phi_{11}(i) = \arccos \frac{\mathbf{n}_{v_i} \cdot \mathbf{n}'_{v_i}}{\|\mathbf{n}_{v_i}\| \cdot \|\mathbf{n}'_{v_i}\|}, \quad i = 1, \dots, N. \quad (5)$$

The local shape curvature is employed to measure the smoothness of the mesh surface. The Gaussian curvature  $\kappa_G$  and the curvature ratio  $\kappa_r$  used in [27] are considered. In differential geometry, the two principal curvatures  $\kappa_1, \kappa_2$  of a surface are provided by the eigenvalues of the shape operator, which are calculated at the location of a vertex using the vertices from its first neighborhood. These curvatures measure how the local surface bends by different amounts in the orthogonal directions at that point.

The Gaussian curvature is defined by the product of the minimum principal curvature and the maximum principal curvature:  $\kappa_G = \kappa_1 \cdot \kappa_2$ , and  $\phi_{12}$  is evaluated by the absolute difference of the two Gaussian curvatures

$$\phi_{12}(i) = |\kappa_G(\mathbf{v}_i) - \kappa_G(\mathbf{v}'_i)|, \quad i = 1, \dots, N. \quad (6)$$

$\kappa_r$  is acquired by two curvature ratios

$$\kappa_r = \frac{\min\{|\kappa_1|, |\kappa_2|\}}{\max\{|\kappa_1|, |\kappa_2|\}}. \quad (7)$$

Moreover,  $\phi_{13}$  is determined by the absolute difference between the two curvature ratios

$$\phi_{13}(i) = |\kappa_r(\mathbf{v}_i) - \kappa_r(\mathbf{v}'_i)|, \quad i = 1, \dots, N. \quad (8)$$

The total feature vector of LFS52 is  $\Phi_{52} = \{\phi_1, \phi_2, \dots, \phi_{13}\}$ .

*LFS64 Features* [16]. Kim et al. extend LFS52 and consider the edge normal, mean curvature and total curvature to enhance the discrimination between the cover and stego meshes with  $\phi_{14} - \phi_{16}$ . An edge normal  $\mathbf{n}_{e_i}$  is the weighted sum of the normals of the faces  $\mathbf{n}_{f_j}$  that have the edge and is defined as follows:

$$\mathbf{n}_{e_i} = \sum_{f_j \in \mathcal{F}(e_i^*)} A(f_j) \cdot \mathbf{n}_{f_j}, \quad (9)$$

where  $\mathcal{F}(e_i^*)$  represents the neighboring faces that contain edge  $e_i$ . Thus, the absolute value of the angles between the two edge normals is computed by

$$\phi_{14}(i) = \arccos \frac{\mathbf{n}_{e_i} \cdot \mathbf{n}'_{e_i}}{\|\mathbf{n}_{e_i}\| \cdot \|\mathbf{n}'_{e_i}\|}, \quad i = 1, \dots, |\mathcal{E}|. \quad (10)$$

The mean curvature  $\kappa_m = (\kappa_1 + \kappa_2)/2$  and total curvature  $\kappa_t = |\kappa_1| + |\kappa_2|$  contribute to two additional features

$$\begin{aligned} \phi_{15}(i) &= |\kappa_m(\mathbf{v}_i) - \kappa_m(\mathbf{v}'_i)| \\ \phi_{16}(i) &= |\kappa_t(\mathbf{v}_i) - \kappa_t(\mathbf{v}'_i)|, \quad i = 1, \dots, N. \end{aligned} \quad (11)$$

The total feature vector of LFS64 is  $\Phi_{64} = \{\phi_1, \phi_2, \dots, \phi_{16}\}$ .

*LFS76 Features* [15]. Li and Bors extend LFS52 and put forward features extracted from spherical coordinates. Spherical coordinates provide a straightforward representation of most graphical objects to characterize the distance from the center and the location of each vertex on a sphere.

Similarly, the absolute difference between each coordinate and its corresponding smoothed version is regarded as features that totally form  $\phi_{17} - \phi_{19}$ . Similarly, the edges in the spherical coordinates compose another three features  $\phi_{20} - \phi_{22}$  [15]. Additionally, the total feature vector of LFS76 is  $\Phi_{76} = \{\phi_1, \phi_2, \dots, \phi_{13}, \phi_{17}, \dots, \phi_{22}\}$ .

*ELFS124 Features* [17]. Li et al. extend LFS52 and propose an extended local feature set with edge vectors for steganalysis. The edge vectors between the cover and smoothed meshes are denoted by  $e_{(v_i, v_j)}$  and  $e'_{(v_i, v_j)}$ . The absolute differences of the  $xyz$ -components, norm, absolute norm difference, and angle of the edge vector are calculated to form 6 features. Similarly, the features in the Laplacian coordinate system are counted. In total, we have 12 features together with those from the Cartesian coordinate system. The total feature vector of ELFS124 is  $\Phi_{124} = \{\phi_1, \phi_2, \dots, \phi_{13}, \phi_{17}, \dots, \phi_{34}\}$ .

### 2.2.3 Nonlinear Mapping

To better extract the available features, the feature mapping is executed after feature extraction [21]. The dynamic range of the nonzero elements of the aggregated features is large. For instance, the range is from  $10^{-5}$  to  $10^{-2}$ . Since large feature values dominate the classification performance, the logarithms of the features are computed to increase the efficiency of the small features and enforce the evenness of the feature distribution

$$x = \log(c + \epsilon), \quad (12)$$

where  $c$  is the feature value and  $\epsilon$  is a small constant to maintain a consistent logarithmic transformation for all of the values.

Then, quantitative analysis of the statistical distribution and the robustness of the different features are undertaken. Four statistical moments that represent the mean, variance, skewness and kurtosis of the nonlinear mapping of the obtained features are considered. Skewness and kurtosis are measures of the peakedness of the distribution. For a normal distribution, both of these measures are zero.

## 3 NORMAL VOTING TENSOR BASED FEATURES

In this section, we propose a neighborhood-level representation-guided tensor voting model for 3D mesh steganalysis. Conventional steganalytic features consist of low-level mesh features (points, edges, triangle faces, etc.), and it is difficult to extract the available features from the intrinsic shapes collected from the original meshes. Compared with the features extracted from the stego versions of the meshes, the differences are not significant. The normal voting tensor represents the local shape [28] and therefore can measure the local smoothness and neighborhood correlation. Because steganographic modification breaks the neighborhood correlations of vertices, the normal voting tensor can be used to extract steganalytic features. Motivated by this fact, we first introduce the definition of the second-order symmetric tensor; then, we present several neighborhood descriptions of meshes, propose a few normal voting tensors that reflect the local surface shapes and finally design new steganalytic features.

### 3.1 Second-Order Symmetric Tensor

A first-order local description of a surface patch is given by the point coordinates and its associated normal. A second-order description would also include the principal curvatures and their directions. To capture the first-order differential geometry information and its singularities, a second-order

symmetric tensor is used. This tensor captures both the orientation information and its confidence [29].

Intuitively, the shape of the tensor defines the type of information captured (points, curves, or surface elements). To express a second-order symmetric tensor  $\mathbf{T}$ , which is graphically depicted by an ellipsoid in 3D, we take the associated quadratic form and diagonalize it. This process leads to a representation based on the eigenvalues  $\lambda_1, \lambda_2, \lambda_3$  and the eigenvectors  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ . In a more compact form,

$$\mathbf{T} = \sum_{k=1}^3 \lambda_k \mathbf{e}_k \mathbf{e}_k^T = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T + \lambda_3 \mathbf{e}_3 \mathbf{e}_3^T, \quad (13)$$

where  $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ . Note that because  $\mathbf{T}$  is a second-order symmetric tensor, the eigenvalues are real and non-negative and the eigenvectors form an orthonormal basis. The above formula can be decomposed into

$$\mathbf{T} = (\lambda_1 - \lambda_2) \mathbf{e}_1 \mathbf{e}_1^T + (\lambda_2 - \lambda_3) (\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T) + \lambda_3 (\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T + \mathbf{e}_3 \mathbf{e}_3^T), \quad (14)$$

where  $\mathbf{e}_1 \mathbf{e}_1^T$  describes a stick,  $\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T$  describes a plate and  $\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T + \mathbf{e}_3 \mathbf{e}_3^T$  describes a ball [29].

### 3.2 Neighborhood Description

Steganographic modification breaks the neighborhood correlation of vertices; hence, by analyzing the local smoothness and neighborhood correlations, we can better discriminate stego meshes from cover meshes. Let us consider combinatorial neighborhoods of vertices and faces. There are four possible *neighbors*: neighborhood vertices of a vertex; neighborhood faces of a vertex; neighborhood faces of a face connected by mutual edges; and neighborhood faces of a face connected by mutual vertices. These neighborhoods are defined from different perspectives (vertex, edge and face), which can diversely describe local smoothness.

- The neighborhood (also referred to as a ring) of a vertex  $\mathbf{v}_i$  is the set  $\mathbf{v}_i^* = \{\mathbf{v}_j \in \mathcal{V} : \mathbf{v}_i \sim \mathbf{v}_j\}$ , as shown in Fig. 2a. The degree of a vertex  $\mathbf{v}_i$  is defined as the cardinality of  $\mathbf{v}_i^*$ , which is denoted by  $|\mathbf{v}_i^*|$ .
- We denote by  $\mathcal{F}(\mathbf{v}_i^*)$  the set of triangles of the ring  $\mathbf{v}_i^*$ , as shown in Fig. 2b.
- We denote by  $f_i^*$  the set of all triangles that share an edge with triangle  $f_i \in \mathcal{F}$  of a mesh; see Fig. 2c. For a closed mesh, the number of adjacent triangles of any triangle in the set  $f_i^*$  is 3, while for a nonclosed mesh, the triangles on the boundaries are partially defective. Thus, the number of adjacent triangles of each triangle is 1 or 2.
- We denote by  $\bar{f}_i^*$  the set of all triangles that share a vertex with a triangle  $\bar{f}_i \in \mathcal{F}$  of a mesh, as shown in Fig. 2d.

### 3.3 Normal Voting Tensor

Based on the above definition of a tensor and the neighborhood description, we propose three normal-voting-based tensors with different neighbors.

*Face Normal of a Vertex Neighbor.* Sun et al. [30] define the normal voting tensor of a vertex on a triangular mesh by

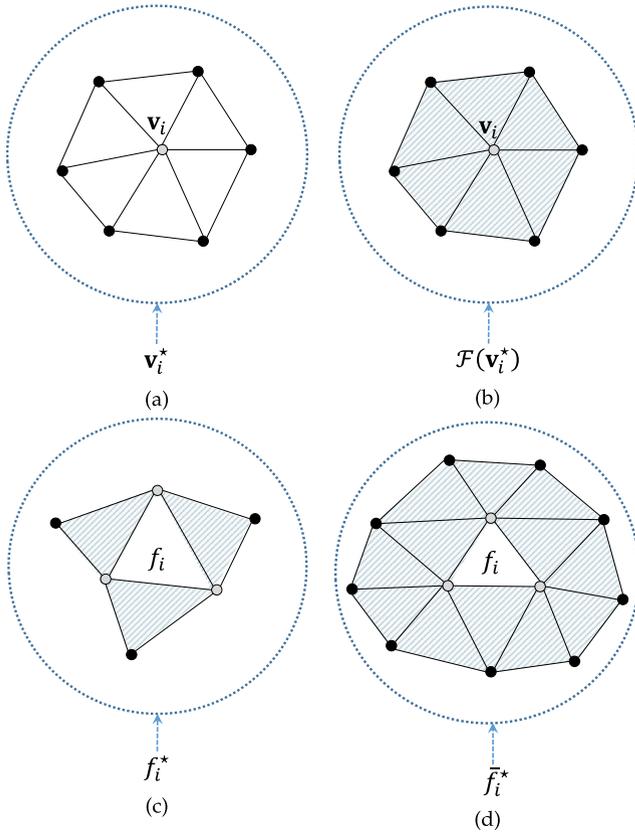


Fig. 2. Illustration of diverse neighbors. (a) Vertex neighborhood  $\mathbf{v}_i^*$ ; (b) Triangles of ring  $\mathbf{v}_i^*$  denoted by  $\mathcal{F}(\mathbf{v}_i^*)$ ; (c) Triangle neighborhood  $f_i^*$  is the set of all triangles sharing an edge with a triangle  $f_i \in \mathcal{F}$ ; (d) Triangle neighborhood  $\bar{f}_i^*$  is the set of all triangles sharing a vertex with a triangle  $f_i \in \mathcal{F}$ .

using the unit normal vectors of the neighbor triangles. If we consider the covariance matrix  $\mathbf{C}_{f_j}$  of each triangle  $f_j \in \mathcal{F}(\mathbf{v}_i^*)$  in Fig. 2b,  $\mathbf{C}_{f_j}$  can be written by the unit normal of the triangle as

$$\mathbf{C}_{f_j} = \mathbf{n}_{f_j} \cdot \mathbf{n}_{f_j}^T. \quad (15)$$

Then, the normal voting tensor  $\mathbf{T}_i$  of a vertex  $\mathbf{v}_i \in \mathcal{V}$  can be generated by accumulating the weighted covariance matrices of its neighbor triangles

$$\mathbf{T}_i = \sum_{f_j \in \mathcal{F}(\mathbf{v}_i^*)} \mu_{ij} \mathbf{n}_{f_j} \cdot \mathbf{n}_{f_j}^T = \sum_{f_j \in \mathcal{F}(\mathbf{v}_i^*)} \mu_{ij} \mathbf{C}_{f_j}, \quad i = 1, \dots, N, \quad (16)$$

where the weight  $\mu_{ij}$  is a vote decided by the area ratio among neighbor triangles and the distance between the vertex and barycenter  $\mathbf{c}_{f_j}$  of each triangle following [30]:

$$\mu_{ij} = \frac{A(f_j)}{\max(A(\mathcal{F}(\mathbf{v}_i^*)))} \exp\left(-\frac{\|\mathbf{c}_{f_j} - \mathbf{v}_i\|}{1/3}\right). \quad (17)$$

In a word, we denote by  $\xi_1$  the first tensor model, and the eigenvalues are computed from it. The eigenvalues are taken as the first part of the proposed features.

*Face Normals of a Face Neighbor.* The normal voting tensor for a mesh face  $f_i$  is formulated as the sum of the weighted covariance matrices from its 1-ring or 2-ring neighboring

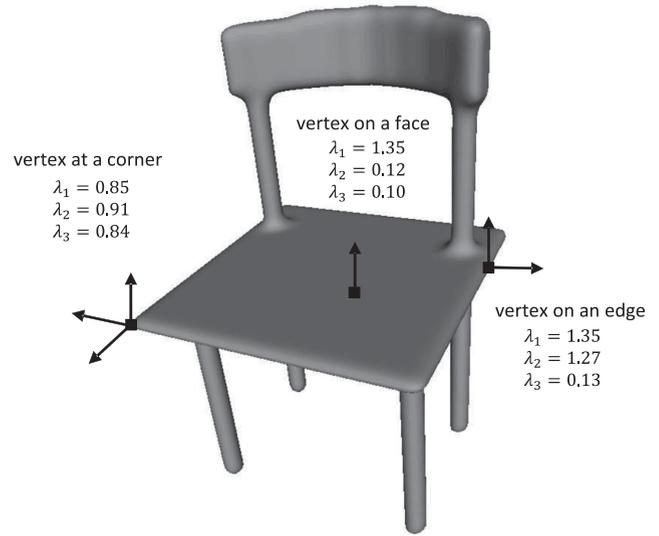


Fig. 3. Eigenvalues of the normal voting tensor for different features (a corner, an edge and a face).

faces [31]. One case is the neighborhood faces of one face conjoined by the shared edges, as shown in Fig. 2c

$$\mathbf{T}_i = \frac{1}{\sum_{f_j \in \bar{f}_i^*} w_{ij} A(f_j)} \sum_{f_j \in \bar{f}_i^*} w_{ij} A(f_j) \mathbf{n}_j \cdot \mathbf{n}_j^T. \quad (18)$$

where  $w_{ij}$  is the weighting function. Here, we simply set all  $w_{ij}$  equal to 1. Furthermore, we denote the second tensor model by  $\xi_2$ .

Another case is based on the neighborhood faces of one face conjoined by shared vertices, as shown in Fig. 2d

$$\mathbf{T}_i = \frac{1}{\sum_{f_j \in \bar{f}_i^*} w_{ij} A(f_j)} \sum_{f_j \in \bar{f}_i^*} w_{ij} A(f_j) \mathbf{n}_j \cdot \mathbf{n}_j^T. \quad (19)$$

Similarly, we denote the third tensor model by  $\xi_3$ .

Because the obtained tensor is a symmetric and positive semidefinite matrix, we can represent  $\mathbf{T}_i$  using an orthonormal basis of the eigenvectors  $\mathbf{e}_k$  and real eigenvalues  $\lambda_k$  by Equation (13).

Our geometrical interpretations are given below. By a 3D decomposition of the tensor  $\mathbf{T}_i$  with Equation (14), we can classify a vertex on the mesh as a corner, a sharp edge or a face by the eigenvalues of the tensor  $\mathbf{T}_i$ . In Fig. 3, the characteristics of the eigenvalues of the normal voting tensor are depicted. On a noise-free triangulated mesh, a planar area has only one dominant eigenvalue in the surface normal direction. Two dominant eigenvalues indicate edge features where the weakest eigenvector will be in the edge direction. At a corner, all three of the eigenvalues are dominant. For example, consider a cube model where the eigenvalues of the tensor are sorted in decreasing order  $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$  and normalized. Then, for the orthogonal features, we can write  $\{\lambda_1 = 1, \lambda_2 = \lambda_3 = 0\}$  (face),  $\{\lambda_1 = \lambda_2 = \frac{\sqrt{2}}{2}, \lambda_3 = 0\}$  (edge) and  $\{\lambda_1 = \lambda_2 = \lambda_3 = \frac{\sqrt{3}}{3}\}$  (corner) [31].

### 3.4 Feature Design

As explained above, the eigenvalues can reflect the shape of the normal voting tensor. Thus, the eigenvalues are effective

TABLE 1  
Results of the MMD between Cover Meshes and Stego Meshes at 2 bpv, 5 bpv and 10 bpv Against LFS64, ELFS124 and NVT+ under the Chao Steganographic Scheme Using the PSB Dataset

Relative payload	Steganalytic method	MMD
2	LFS64	.0396
	ELFS124	.0305
	NVT+	<b>.0816</b>
5	LFS64	.0503
	ELFS124	.0361
	NVT+	<b>.1008</b>
10	LFS64	.1026
	ELFS124	.0603
	NVT+	<b>.1706</b>

features for representing the shape of a local surface patch, as shown in Fig. 3. Formally, the absolute values of the differences between the eigenvalues of the two tensors (from the original face and the smoothed face) are regarded as features. The well-designed three-tensor models ( $\xi_1, \xi_2, \xi_3$ ) are such that each extracts three eigenvalue differences to form a total of 9 features, which are denoted by  $\phi_{23} - \phi_{31}$ .

Following the former convention, for each eigenvalue  $\lambda_k$  from each tensor voting model  $\xi_j$ ,

$$\phi_k(i) = |\lambda_k(f_i) - \lambda_k(f'_i)|, \quad i = 1, \dots, |\mathcal{F}|, \quad k = 1, 2, 3, \quad (20)$$

where  $f'$  is the triangle face from the smoothed mesh  $\mathcal{M}'$ . After the extraction of the statistical moments, the proposed features form  $9 \times 4 = 36$  features. We combine the proposed NVT features and LFS64 to form a new feature set NVT+, and the dimension of NVT+ reaches 100.

## 4 EVALUATION AND RESULTS

### 4.1 Setups

In this paper, two disjoint mesh datasets are adopted:

The *Princeton Segmentation Benchmark*<sup>1</sup> (PSB) dataset is a mesh segmentation dataset with 354 objects [32]. Overall, 260 pairs of cover-objects are used for training, and 94 pairs of stego-objects are used for testing, which was also the configuration in the previous article [14].

The *Princeton ModelNet*<sup>2</sup> (PMN) dataset contains 12,311 mesh data for computer vision, computer graphics, robotics and cognitive science [33]. We take ModelNet40 with 40 categories for training and testing. A preprocessing with only 4,000 meshes is selected with median-volume meshes in favor of time savings. We use 50 percent for training and 50 percent for testing.

We investigate the effectiveness of the proposed method by using three state-of-the-art steganographic schemes by Chao [7], Li [10] and VND [12]; we employ a varying relative payload rate  $\eta \in \{1, 2, \dots, 14\}$ , which is defined by the ratio of the total length of a message and the number of vertices as  $\eta = m/N$  bpv (bit per vertices). The detectors are

trained as binary classifiers and implemented using the Fisher linear discriminant (FLD) ensemble [23] with the default settings. A separate classifier is trained for each embedding algorithm and relative payload. By default, the ensemble classifier minimizes the total classification error probability under equal priors

$$P_E = \min_{P_{FA}} \frac{1}{2} (P_{FA} + P_{MD}), \quad (21)$$

where  $P_{FA}$  and  $P_{MD}$  are the false-alarm probability and the missed-detection probability, respectively. The ultimate security  $\bar{P}_E$  is qualified by the average of all of the error rates over all 30 trials, and a larger  $\bar{P}_E$  implies stronger security.

### 4.2 Feature Validation

To justify our claim that tensor features are effective at detecting modifications caused by steganography, we implement a simulation with the following steps:

- Modify vertices of the mesh by the Chao steganographic method [7] under the relative payload  $\eta = 2, 5, 10$ . The meshes that we use are from the PSB dataset.
- Calculate LFS64 [16], ELFS124 [17] and the proposed NVT+ steganalytic feature vector for each 3D mesh. The obtained feature sets are denoted by  $\{feat_C(LFS64)\}$  and  $\{feat_S(LFS64)\}$ ,  $\{feat_C(ELFS124)\}$  and  $\{feat_S(ELFS124)\}$ , and  $\{feat_C(NVT+)\}$  and  $\{feat_S(NVT+)\}$  for each cover-stego pair.
- Calculate the MMD (maximum mean discrepancy [34]), which measures the distance between the feature set of the cover objects and the stego-objects) between  $\{feat_C(LFS64)\}$  and  $\{feat_S(LFS64)\}$ ,  $\{feat_C(ELFS124)\}$  and  $\{feat_S(ELFS124)\}$ , and  $\{feat_C(NVT+)\}$  and  $\{feat_S(NVT+)\}$ . Obtain the average value of the MMD over 30 distinct, independent tests on the dataset. Then, make a comparison.

The statistical results of the MMD are given in Table 1. In general, because the MMD represents the distance in the steganalytic feature space between the cover set and stego set, the larger the MMD is, the better the discriminant ability will be. From the statistical results, the MMD of NVT+ is larger than the MMDs of the other two methods, which demonstrates that NVT+ has better discriminant features.

### 4.3 Visualization of Steganalytic Features

We have compared the performance of our proposed normal voting tensor-based features with existing features by presenting the sample distributions after the features are projected onto the selected 2D subspace, as shown in Fig. 4. The stego meshes are generated by the Chao method in the PSB dataset.

In detail, first, we extract eigenvalue  $\lambda_1$  of the normal voting tensor feature from the neighbor of  $\xi_1$  by Equation (13), and we calculate the face normal features (which are demonstrated to be the most effective features in [12]) by Equation (3). After log-mapping is performed, skewness and kurtosis are measured for both steganalytic methods. In the two figures of Fig. 4, we can see the cover sample depicted with blue asterisks and the stego sample with red circles.

1. <http://segeval.cs.princeton.edu/>

2. <http://modelnet.cs.princeton.edu/>

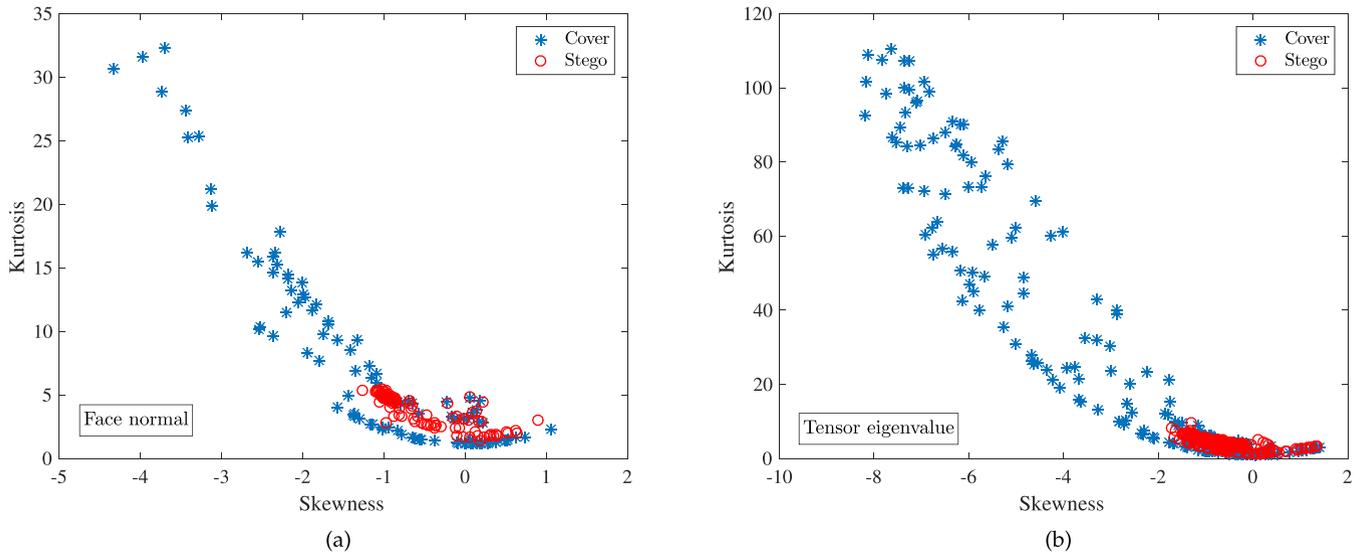


Fig. 4. Mesh object distributions in the selected 2D manifolds with different feature extraction algorithms on Chao steganographic method under 10 bpv relative payloads with PSB dataset. We take skewness and kurtosis as measurements for both steganalytic methods. (a) face normal features  $\phi_{10}$  are by far the strongest features, refer to [12]. Angles between cover meshes and stego meshes are taken as evaluation indicator. (b) tensor voting features. Variation of  $\lambda_1$  is evaluated.

Fig. 4a describes the sample distribution through the face normal features, and Fig. 4b describes the normal voting tensor features. It is harder for a linear classifier to partition the cover samples and stego samples in Fig. 4a than the samples in Fig. 4b, which implies that the normal voting tensor has better discriminative power with respect to cover meshes and stego meshes.

#### 4.4 Submodel Selection

We utilize the steganographic methods proposed by Chao et al. [7] and Zhou et al. [12] to validate our proposed steganalytic methods. Afterwards, we evaluate the performance of the individual submodel of the steganalytic feature vector independently.

Our approach works as follows. First, we create a set of stego meshes embedded with Chao and VND under a certain payload rate. Then, we use the FLDs criteria to evaluate the accuracy of the individual features for detection. Specifically, we use FLDs as base learners due to their simple and fast training. Denote the cover and stego features from the

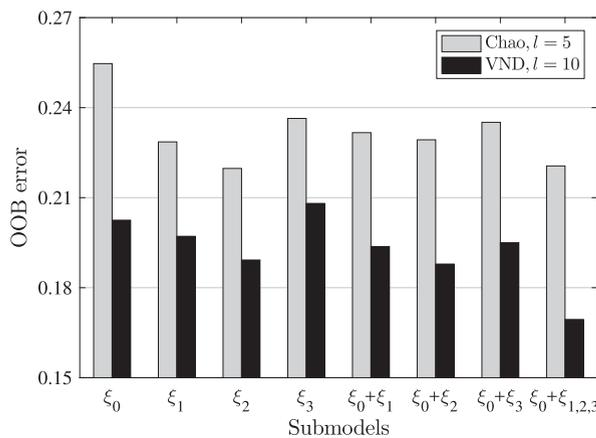


Fig. 5. OOB error estimates averaged over Chao under 5 bpv and VND under 10 bpv using the PSB dataset.

training set as  $\mathbf{x}^{(m)}$  and  $\bar{\mathbf{x}}^{(m)}$  for  $m = 1, \dots, N^{trn}$ , respectively. The training makes use of the so-called “out-of-bag” (OOB) error estimate [35]

$$E_{\text{OOB}}^{(L)} = \frac{1}{2N^{trn}} \sum_{m=1}^{N^{trn}} \left( B^{(L)}(\mathbf{x}^{(m)}) + 1 - B^{(L)}(\bar{\mathbf{x}}^{(m)}) \right), \quad (22)$$

which is an unbiased estimate of the testing error [36].

We compute the OOB estimates for each submodel under different embedding payload rates: 5 bpv for Chao [7] and 10 bpv for VND [12]. We investigate the classification performances of the submodels; in Fig. 5, for each submodel, we plot its OOB error estimate. We denote this estimate by  $\xi_0 = \text{LFS64}$  for readability.

The proposed NVT features are designed differently from the former features such as LFS64, as NVT features are extracted from adjacent faces, while the former features are extracted from a single vertex, single edge or single face. Specifically, the tensor voting operation is introduced before the eigenvalue extraction to characterize the vertex relationship.  $\xi_1$  is the normal voting tensor constructed from the vertex neighbor,  $\xi_2$  is the tensor constructed from the triangle neighbor connected by the edges and  $\xi_3$  is the tensor constructed from the triangle neighbor connected by the vertices. As shown in Fig. 5, all three models  $\xi_1, \xi_2, \xi_3$  individually provide lower OOBs than the state-of-the-art LFS64 model ( $\xi_0$ ). When cascading the proposed features  $\xi_1, \xi_2, \xi_3$  with the LFS64 feature  $\xi_0$  to form a 100D feature vector, better performance can be expected. Experimentally, we achieve the lowest OOB error when all of the models are combined together.

#### 4.5 Comparison with the State-of-the-Art Steganalytic Methods

We use five related steganalytic feature sets, namely, YANG208 [21], LFS52 [14], LFS64 [16], LFS76 [15] and ELFS124 [17], for comparison. Three steganographic schemes, namely, Chao [7], VND [12], and Li [10], are employed

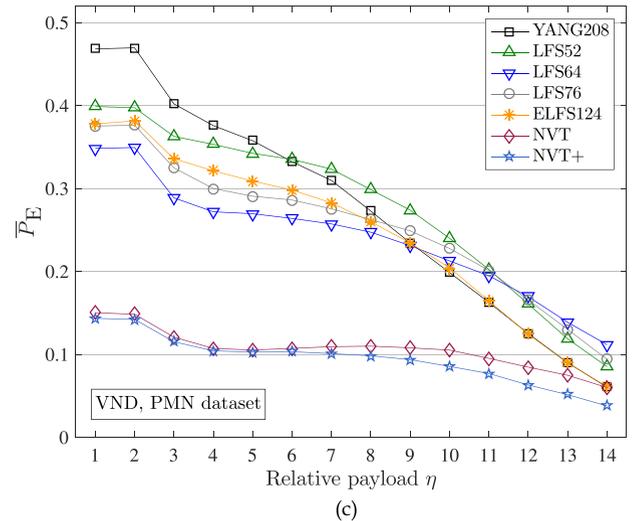
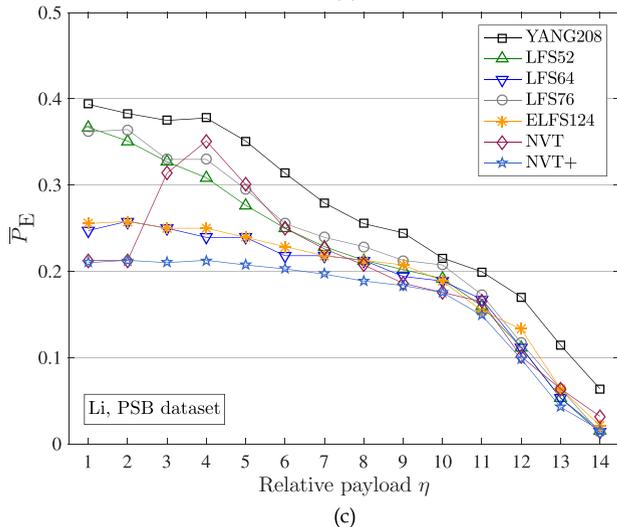
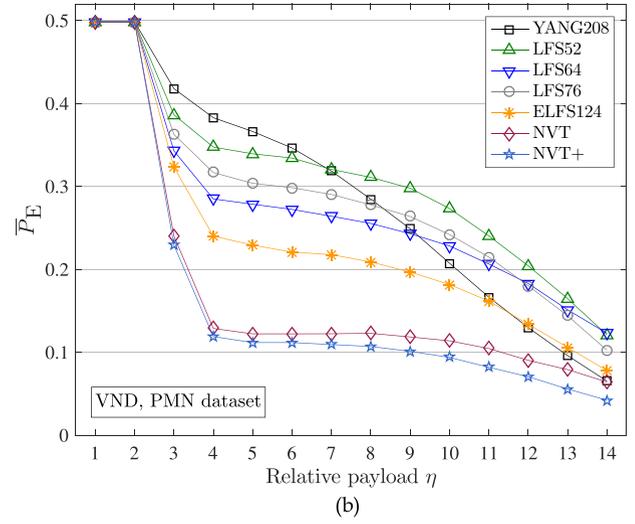
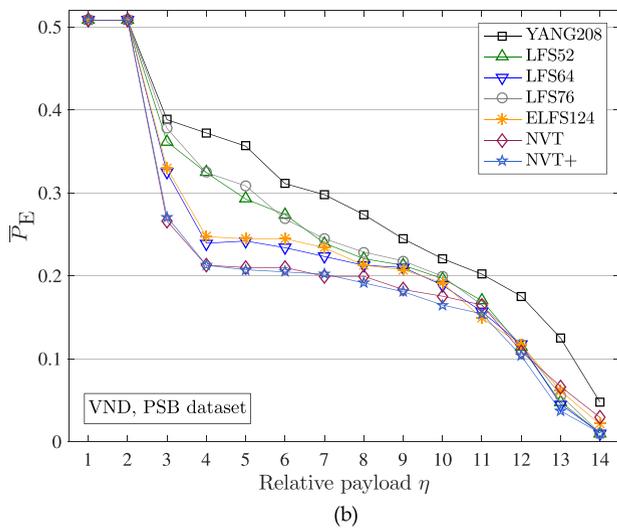
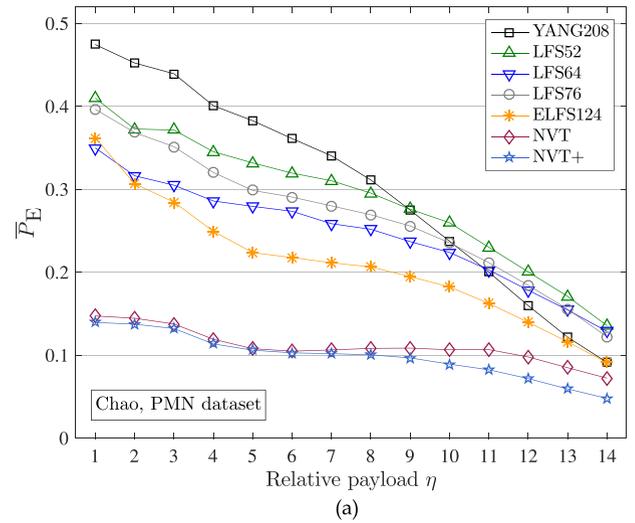
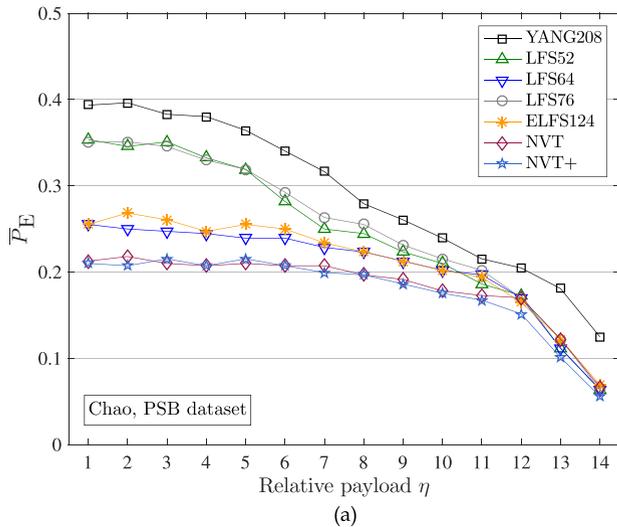


Fig. 6. Detection performance of YANG208, LFS52, LFS64, LFS76, ELFS124 and our proposed NVT+ on detecting each steganography under varying relative payload on the PSB dataset. (a) Chao method. (b) VND method. (c) Li method.

Fig. 7. Detection performance of YANG208, LFS52, LFS64, LFS76, ELFS124 and our proposed NVT+ on detecting each steganography under varying relative payload on the PMN dataset. (a) Chao method. (b) VND method. (c) Li method.

for evaluation. Note that we do not consider distortionless schemes [18], [19], [20], though these steganalytic features cannot discriminate these stegos generated by order permutation based steganography from covers, they can be easily

detected by specifically designed detector. Because the orders of faces are generally highly relevant for normal meshes, when the stego meshes are created by distortionless methods, the orders of faces are uncorrelated. As an

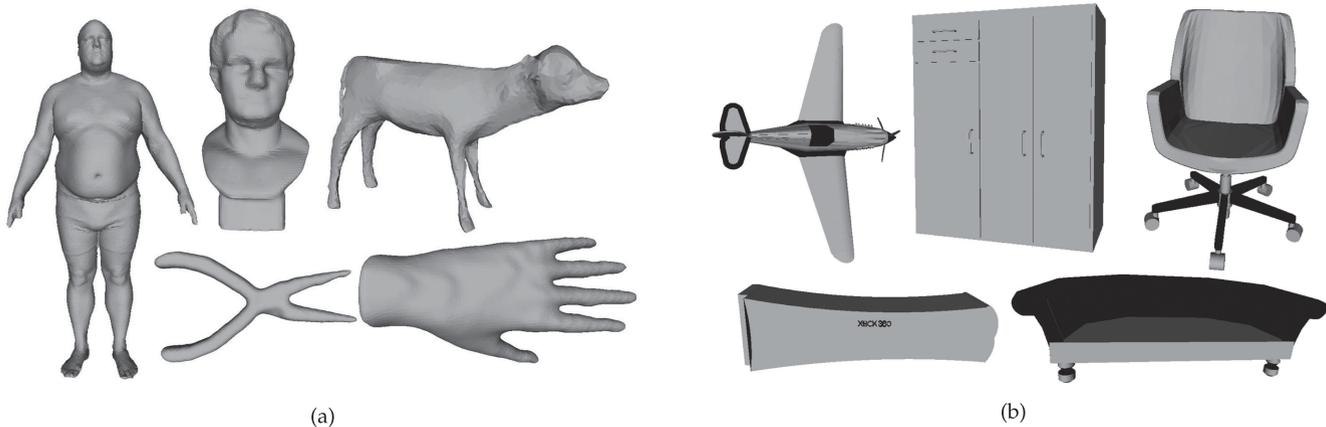


Fig. 8. 3D mesh objects used in the steganalytic training and testing phases. (a) The PSB dataset. (b) The PMN dataset.

example, we give some parts of the mesh structure in Fig. 9. It is shown that line 2,397 and line 2,398 have two common vertices, and line 2,398 and line 2,399 have two common vertices. Similarly, adjacent lines share some common vertices. But for stego meshes created by distortionless methods, the adjacent lines share no same vertices.

The detection error  $\bar{P}_E$  as a function of the embedded payload size for the PSB dataset is shown in Fig. 6. The gain of NVT+ w.r.t. both LFS64 and ELFS124 reaches a maximum of 5 percent. When the relative payload increases, the improvement decreases. When the payload rate is high for all of the steganographic schemes, the advantage of the NVT+ features is not prominent. For VND steganographic schemes with  $\gamma = 1, 2$ , the detection errors are nearly 50 percent because the modification of the vertices are too minor to be detected by any steganalytic methods.

To investigate the performance of NVT+ on a larger dataset, we have experimented on the PMN dataset. We show the results of the performance test in Fig. 7. We observe that except for the case of VND with a small relative payload, where none of the features can distinguish the difference between the cover mesh and stego mesh, the proposed NVT+ scheme always achieves lower detection error than the former steganalytic features. When the payload rate is low, NVT+ has more advantage in detection and its improvement can be as strong as 22 percent.

2390	-0.133675	-0.089779	-0.152190
2391	-0.151744	-0.089779	-0.152190
2392	-0.169812	-0.089779	-0.152190
2393	-0.187880	-0.089779	-0.152190
2394	-0.205949	-0.089779	-0.152107
2395	-0.223745	-0.089803	-0.150818
2396	-0.240709	-0.089956	-0.146631
2397	3 51 0 22		
2398	3 0 1 22		
2399	3 22 1 23		
2400	3 1 2 23		
2401	3 23 2 24		
2402	3 2 3 24		
2403	3 24 3 25		
2404	3 3 4 25		
2405	3 25 4 26		
2406	3 4 5 26		

Fig. 9. Structure of meshes.

The boost of the detection rate in the PMN dataset is more significant than in the PSB dataset, which we attribute to the data sources, as shown in Fig. 8. Whereas the shapes of the mesh objects from the PSB database are mostly acquired by reconstruction from natural 3D objects, the mesh objects from the PMN database are crafted by CAD techniques. The objects from PSB have more diverse local texture structures than objects from the PMN dataset. By extracting the neighborhood-level features such as NVTs, one is more likely to detect fixed modes in CAD-crafted objects than natural objects, which accounts for the different increases in the detection rates.

#### 4.6 Statistical Significance of the Improved Accuracy

To confirm the statistical significance of the improved accuracy, a z-test is realized between the error rates of the original and the improved NVT+ algorithms. The hypotheses are denoted as follows:

$$H_0 : \mu_1 = \mu_2; \quad H_1 : \mu_1 \neq \mu_2,$$

where  $\mu_1$  and  $\mu_2$  are the mean values of the testing errors of the original and NVT+ features, respectively, and  $\mu_1 = \mu_2$  indicates that there is no significant difference between them.

The z-score  $z$  is computed by

$$z = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}},$$

where  $n_1$  and  $n_2$  are the numbers of the testing samples and  $\sigma_1$  and  $\sigma_2$  are the standard deviations of the original and NVT+ features, respectively. By computing the z-score from statistical libraries or packages, the corresponding  $p$ -value can be obtained. A lower  $p$ -value indicates a lower probability that the null hypothesis  $H_0$  holds. If the  $p$ -value is less than a threshold, the null hypothesis  $H_0$  is rejected, and the improvement is deemed statistically significant and reliable. We set the level of significance  $z$  at 5 percent. Since we do multiple, independent, and post hoc statistical tests, we use the Bonferroni correction to adjust the threshold to account for spurious cases of high statistical significance. Since 30 hypotheses are being tested, the new critical

TABLE 2  
Computational Complexity of the Proposed NVT+ and Existing Methods on the PSB Dataset under an Embedding Payload Rate Size of 8 Bits per Vertex (bpv)

Time (s)	YANG208	LFS52	LFS64	LFS76	ELFS124	NVT+
Training	1.99	1.27	2.19	1.94	2.07	2.22
Feature extraction	17.13	82.13	84.33	106.72	107.04	1689.84
Classification	17.76	7.20	9.70	7.13	16.78	9.45

$p$ -value is  $0.05/30 = 0.0017$ , and the corresponding quantile  $z_{0.0017} = 2.93$ .

Under different payloads and steganographic schemes, in most cases, the test statistic  $z$  values are larger than the corresponding quantile  $z_{0.0017}$ , which implies the detection improvements have statistical significance.

#### 4.7 Comparison of the Computational Complexity

We evaluate the computational complexity of the PSB dataset under a relative payload of 8 bpv in Table 2. We observe that the proposed NVT+ has much higher computational complexity than the existing methods for feature extraction, as the search for adjacent triangles is time-consuming.

## 5 CONCLUSION

A 3D mesh steganalysis scheme based on the normal voting tensor (NVT) is proposed in this paper. We analyze the distortion caused by steganographic modifications and discover that the normal voting tensor could measure the smoothness of the local region. Here, three tensors with different neighbor patterns are used to extract the steganalytic features. Eigenvalues are obtained from each tensor, and nonlinear mappings and several statistical moments are subsequently calculated. The proposed NVT+ features achieve good detection performance against several state-of-the-art steganographic methods under various conditions. It is worth mentioning that the proposed scheme NVT+ has a maximum 22 percent improvement of the detection rate on the Princeton ModelNet dataset. In the future, we would like to investigate the mesh smoothing operation to extract more discriminant features. In addition, designing a better normal voting tensor is another part of our future work.

## ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China under Grant U1636201 and 61572452, by Anhui Initiative in Quantum Information Technologies under Grant AHY150400, and by the Fundamental Research Funds for the Central Universities under Grant WK6030000135 and WK6030000136.

## REFERENCES

- [1] J. Frank and L. Mensik, "Digital construction kit: 3D computer graphics for creative and accessible use of museum collections," *Biodiversity Inf. Sci. Standards*, vol. 2, 2018, Art. no. e26023.
- [2] 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.
- [3] C.-M. Wang and Y.-M. Cheng, "An efficient information hiding algorithm for polygon models," in *Proc. Eurographics*, 2005, vol. 24, pp. 591–600.
- [4] Y.-M. Cheng and C.-M. Wang, "A high-capacity steganographic approach for 3D polygonal meshes," *The Visual Comput.*, vol. 22, no. 9/11, pp. 845–855, 2006.
- [5] P. Amat, W. Puech, S. Druon, and J.-P. Pedeboy, "Lossless 3D steganography based on MST and connectivity modification," *Signal Process.: Image Commun.*, vol. 25, no. 6, pp. 400–412, 2010.
- [6] V. Itier, W. Puech, G. Gesquière, and J.-P. Pedeboy, "Joint synchronization and high capacity data hiding for 3D meshes," in *Proc. Three-Dimensional Image Process. Meas. Appl.*, 2015, vol. 9393, Art. no. 939305.
- [7] M.-W. Chao, C.-H. Lin, C.-W. Yu, and T.-Y. Lee, "A high capacity 3D steganography algorithm," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 2, pp. 274–284, Mar. 2009.
- [8] Y. Yang, N. Peyerimhoff, and I. Ivriissimtzi, "Linear correlations between spatial and normal noise in triangle meshes," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 1, pp. 45–55, Jan. 2013.
- [9] 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. 26 421–26 445, 2017.
- [10] 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.
- [11] Z. Li, S. Beugnon, W. Puech, and A. G. Bors, "Rethinking the high capacity 3D steganography: Increasing its resistance to steganalysis," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 510–514.
- [12] H. Zhou, K. Chen, W. Zhang, Y. Yao, and N.-H. Yu, "Distortion design for secure adaptive 3D mesh steganography," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1384–1398, Jun. 2019.
- [13] Y. Yang and I. Ivriissimtzi, "Polygonal mesh watermarking using laplacian coordinates," *Comput. Graph. Forum*, vol. 29, no. 5, pp. 1585–1593, 2010.
- [14] Z. Li and A. G. Bors, "3D mesh steganalysis using local shape features," in *Proc. Int. Conf. Acoustics Speech Signal Process.*, 2016, pp. 2144–2148.
- [15] Z. Li and A. G. Bors, "Steganalysis of 3D objects using statistics of local feature sets," *Inf. Sci.*, vol. 415, pp. 85–99, 2017.
- [16] 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.*, 2017, pp. 358–365.
- [17] 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.*, 2018, pp. 1683–1687.
- [18] A. Bogomjakov, C. Gotsman, and M. Isenburg, "Distortion-free steganography for polygonal meshes," *Comput. Graph. Forum*, vol. 27, no. 2, pp. 637–642, 2008.
- [19] C. I. Tan, C.-K. Lin, W.-K. Tai, and C.-C. Chang, "Hiding data: A high-capacity distortionless approach," *Multimedia Syst.*, vol. 15, no. 6, 2009, Art. no. 325.
- [20] 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, 2012.
- [21] Y. Yang and I. Ivriissimtzi, "Mesh discriminative features for 3D steganalysis," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 3, 2014, Art. no. 27.
- [22] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 1998.
- [23] 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.
- [24] J. Fridrich, M. Goljan, and D. Hoge, "Steganalysis of JPEG images: Breaking the F5 algorithm," in *Proc. Int. Workshop Inf. Hiding*, 2002, pp. 310–323.
- [25] J. Kodovský and J. Fridrich, "Calibration revisited," in *Proc. 11th ACM Workshop Multimedia Secur.*, 2009, pp. 63–74.

- [26] G. Taubin, "A signal processing approach to fair surface design," in *Proc. 22nd Annu. Conf. Comput. Graph. Interactive Techn.*, 1995, pp. 351–358.
- [27] J. Rugis and R. Klette, "A scale invariant surface curvature estimator," in *Proc. Pacific-Rim Symp. Image Video Technol.*, 2006, pp. 138–147.
- [28] H. S. Kim, H. K. Choi, and K. H. Lee, "Feature detection of triangular meshes based on tensor voting theory," *Comput. Aided Des.*, vol. 41, no. 1, pp. 47–58, 2009.
- [29] G. Medioni, C.-K. Tang, and M.-S. Lee, "Tensor voting: Theory and applications," in *Proc. 12eme Congres Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence in Artificielle*, 2000, [Online]. Available: [http://www.sci.utah.edu/~gerig/CS7960-S2010/handouts/Medioni\\_tensor\\_voting.pdf](http://www.sci.utah.edu/~gerig/CS7960-S2010/handouts/Medioni_tensor_voting.pdf). [Accessed on: Jun. 30, 2019].
- [30] Y. Sun, D. L. Page, J. K. Paik, A. Koschan, and M. A. Abidi, "Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing," in *Proc. IEEE Int. Conf. Image Process.*, 2002, vol. 3, pp. 825–828.
- [31] S. K. Yadav, U. Reitebuch, and K. Polthier, "Mesh denoising based on normal voting tensor and binary optimization," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 8, pp. 2366–2379, Aug. 2018.
- [32] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 73:1–73:12, 2009.
- [33] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.
- [34] T. Pevný and J. Fridrich, "Benchmarking for steganography," in *Proc. Int. Workshop Inf. Hiding*, 2008, pp. 251–267.
- [35] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 3, pp. 868–882, Jun. 2012.
- [36] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.



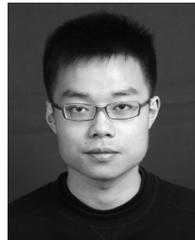
**Hang Zhou** received the BS degree from the School of Communication and Information Engineering, Shanghai University, in 2015. He is currently working toward the PhD degree in information security at the University of Science and Technology of China (USTC). His research interests include information hiding, image processing and computer graphics.



**Kejiang Chen** received the BS degree from the School of Communication and Information Engineering, Shanghai University, in 2015. He is currently working toward the PhD degree in information security at the University of Science and Technology of China (USTC). His research interests include information hiding, image processing and deep learning.



**Weiming Zhang** received the MS and PhD degrees from the Zhengzhou Information Science and Technology Institute, P.R. China, in 2002 and 2005, respectively. 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.



**Chuan Qin** received the BS degree from the School of Information Science and Technology, Northwest University, in 2016. He is currently working toward the PhD degree in information security at the University of Science and Technology of China (USTC). His research interests include steganography and steganalysis.



**Nenghai Yu** received the BS degree from the Nanjing University of Posts and Telecommunications, in 1987, the ME degree from Tsinghua University, in 1992, and the PhD degree from the University of Science and Technology of China, in 2004, where he is currently a professor. His research interests include multimedia security, multimedia information retrieval, video processing and information hiding.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/csdl](http://www.computer.org/csdl).