Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Algorithm for 3D point cloud steganalysis based on composite operator feature enhancement*

Shuai REN†1, Hao GONG†‡1, Suya ZHENG2

[1]School of Information Engineering, Chang'an University, Xi'an 710064, China

[2]School of Geological Engineering and Geomatics, Chang'an University, Xi'an 710064, China

†E-mail: shuairen@chd.edu.cn; 2022124045@chd.edu.cn

Received May 7, 2024; Revision accepted June 24, 2024; Crosschecked July 29, 2024; Published online Aug. 28, 2024

Abstract: Three-dimensional (3D) point cloud information hiding algorithms are mainly concentrated in the spatial domain. Existing spatial domain steganalysis algorithms are subject to more disturbing factors during the analysis and detection process, and can only be applied to 3D mesh objects, so there is a lack of steganalysis algorithms for 3D point cloud objects. To change the fact that steganalysis is limited to 3D mesh and eliminate the redundant features in the 3D mesh steganalysis feature set, we propose a 3D point cloud steganalysis algorithm based on composite operator feature enhancement. First, the 3D point cloud is normalized and smoothed. Second, the feature points that may contain secret information in 3D point clouds and their neighboring points are extracted as the feature enhancement region by the improved 3DHarris-ISS composite operator. Feature enhancement is performed in the feature enhancement region to form a feature-enhanced 3D point cloud, which highlights the feature points while suppressing the interference created by the rest of the vertices. Third, the existing 3D mesh feature set is screened to reduce the data redundancy of more relevant features, and the newly proposed local neighborhood feature set is added to the screened feature set to form the 3D point cloud steganography feature set POINT72. Finally, the steganographic features are extracted from the enhanced 3D point cloud using the POINT72 feature set, and steganalysis experiments are carried out. Experimental analysis shows that the algorithm can accurately analyze the 3D point cloud's spatial steganography and determine whether the 3D point cloud contains hidden information, so the accuracy of 3D point cloud steganalysis, under the prerequisite of missing edge and face information, is close to that of the existing 3D mesh steganalysis algorithms.

Key words: Steganalysis; 3D point cloud; Feature enhancement; Feature set filtering
https://doi.org/10.1631/FITEE.2400360 **CLC number:** TP391

1 Introduction

Three-dimensional (3D) models are widely used as multimedia communication carriers due to their complex structural characteristics and various forms of expression, which have attracted the attention of scholars in the field of information hiding. Information hiding of 3D models can be effectively realized by encoding the secret information and expressing the encoded secret information by modifying the structure of some areas in 3D models. Compared with traditional methods of hiding information in two-dimensional (2D) images, 3D models have a larger capacity and more areas for information hiding.

With the development of 3D model information steganography field, some miscreants began to use 3D model information steganography algorithms to carry out illegal activities, which leads to a series of security and privacy issues. To prevent 3D

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 62372062)

ORCID: Shuai REN, https://orcid.org/0000-0001-8149-8602; Hao GONG, https://orcid.org/0009-0002-7798-8857

model steganography from being used maliciously, 3D model steganography analysis algorithms have been proposed. The spatial domain steganalysis method YANG208 for 3D mesh was first proposed in 2014, and feature detection and steganalysis were achieved by analyzing the coordinate changes under Cartesian and Laplace coordinate systems (Yang and Ivrissimtzis, 2014). Since then, 3D model steganalysis has been continuously studied with the goal of proposing new feature sets. Based on YANG208, further analysis and screening of features were carried out. Part of the feature set was retained, and vertex Euclidean distances and dihedral angle offsets were added to form the YANG40 feature set, which reduces the feature dimension under the premise of guaranteeing the accuracy. Based on the vertex normal vector and curvature, the detection ability of feature set LFS52 was significantly improved compared with that of the YANG40 feature set (Li and Bors, 2016). Based on LFS52, the spherical coordinate system and the edge pinch angle features were added to form the LFS76 feature set (Li and Bors, 2017), and the new edge vector features were added to form the ELFS124 feature set based on LFS76, which leads to a significant increase in the steganography analysis accuracy (Li et al., 2018a, 2018b; Li and Bors, 2020a). In addition to the spatial domain steganalysis algorithms, some studies have proposed the NVT+ steganalysis features based on the tensor matrix eigenroot (Zhou et al., 2021) and the WFS228 dataset based on the 3D wavelet decomposition which effectively expands the range of steganalysis in the transform domain (Li and Bors, 2020b). All the above 3D model steganalysis algorithms use 3D mesh as the carrier, which can effectively perform steganalysis on most of existing steganography algorithms. However, in reality, 3D mesh is difficult to obtain, and 3D point clouds are more often used as the carrier in the practical applications. Existing 3D model steganalysis algorithms all rely on the global information in the 3D mesh, point information, edge information, and surface information, and no research has proposed algorithms that can perform steganalysis on 3D point clouds.

Compared with 3D mesh, 3D point clouds lack edge information and face information. If the original 3D mesh information steganography algorithm is used for steganalysis of 3D point clouds, the accuracy will be greatly reduced, and it is not possible to effectively determine whether it contains steganographic information. Therefore, the key to improving the accuracy of 3D point cloud steganalysis is to analyze the key points that may contain steganographic information.

Currently, the local key point extraction operators for 3D point clouds are the scale-invariant feature transform (SIFT) operator, 3D Harris operator, and intrinsic shape signature (ISS) operator. The SIFT operator implements a scale-invariant key point detection method based on an image pyramid and local features, which is invariant to scale transformations and rotational transformations. This property can be used to realize cryptographic analysis of 3D point clouds at multiple resolutions (Lowe, 2004). Some studies have compared various subpixel scale-invariant region detectors, such as Harris-Affine and Hessian-Affine, and pointed out that the 3D Harris operator has lower algorithmic complexity than the existing SIFT operator in processing 3D point clouds (Mikolajczyk et al., 2005). The key point extraction method based on local surface curvature thresholding can be used to extract features such as edges and ridges of the point cloud (Pauly et al., 2003; Decker et al., 2023). In the earliest study on ISS operators, it is pointed out that ISS operator constructs stable target shape deformation feature vectors by analyzing the local surface information of point clouds (Zhong, 2009; Liu et al., 2024). Each of the above point cloud key point extraction algorithms has its own advantages in terms of algorithmic time complexity and feature point description area, and none of the current key point extraction algorithms can independently filter out all the possible dense key points. The accuracy of the SIFT operator decreases dramatically when dealing with a non-smooth 3D point cloud, and the local surface curvature thresholding based key point extraction method extracts only the protruding key points from the edge region. The 3D Harris operator can efficiently extract the edge key points of 3D point clouds with lower time complexity than the SIFT operator and local surface curvature thresholding based key point extraction methods, and can extract the edge key points with a lower degree of convexity. The ISS operator can extract the local surface key points, which meets the requirements of global key point coverage for steganalysis. Therefore, this paper adopts a combination of the 3D Harris operator and ISS operator

for key point extraction, named 3DHarris-ISS key points.

In this paper, we propose an algorithm for 3D point cloud steganalysis based on composite operator feature enhancement. The specific contributions are as follows:

1. The Gaussian pyramid theory is combined with the 3D Harris operator to realize the scale-invariant 3D Harris operator, and the ISS operator is combined on this basis to effectively delineate the feature-enhanced region of 3D point clouds.

2. A local feature enhancement method for 3D point clouds is proposed, and steganography analysis for 3D point clouds is realized for the first time.

3. The existing 3D mesh feature dataset is screened and new neighborhood features are added to form a 3D point cloud steganography feature set, which effectively improves the accuracy of the 3D point cloud steganography analysis.

2 Related works

The generic 3D mesh steganalysis algorithm consists of three stages: preprocessing, feature extraction, and learning (Zhou et al., 2022), as shown in Fig. 1. The basic principle of steganalysis is as follows: first the model is reduced to the unencrypted state using a smoothing operation, and then the original model and the smoothed model are normalized at the same time. Second, the steganalysis feature set is used to extract the features of the two models and calculate the residuals. If the original model is an encrypted one, the residual difference between the original model and the smoothed model is large; otherwise, the residual difference is small. Finally, the residuals are inputted into the classification learner for classification to complete the steganalysis.

Steganography in 3D models can be regarded as the operation of vertices in a local region of a 3D model according to the steganography rules, and this process can be regarded as adding noise to the 3D models, which is essentially a noise-adding operation. The 3D models can be smoothed to eliminate the noise and steganographic information, but the vertex coordinate changes are small and not easily to be detected. In 3D mesh steganalysis algorithms, edge and face information is used to highlight the changes in point information, whereas 3D point clouds lack edge and face information, so it is necessary to consider other aspects to highlight the changes in point information.

The 3D point cloud steganalysis algorithm proposed in this paper is based on the general 3D mesh steganalysis algorithm, and is improved for the preprocessing and feature extraction.

1. In the preprocessing section, we consider using feature enhancement to highlight the point information in the model to compensate for the lack of edge and face information in the 3D point cloud model. Global feature enhancement can enhance the vertex changes caused by steganography, but at the same time, it will lead to the enhancement of non-steganographic regions, which will cause unnecessary interference. To perform steganalysis operations on

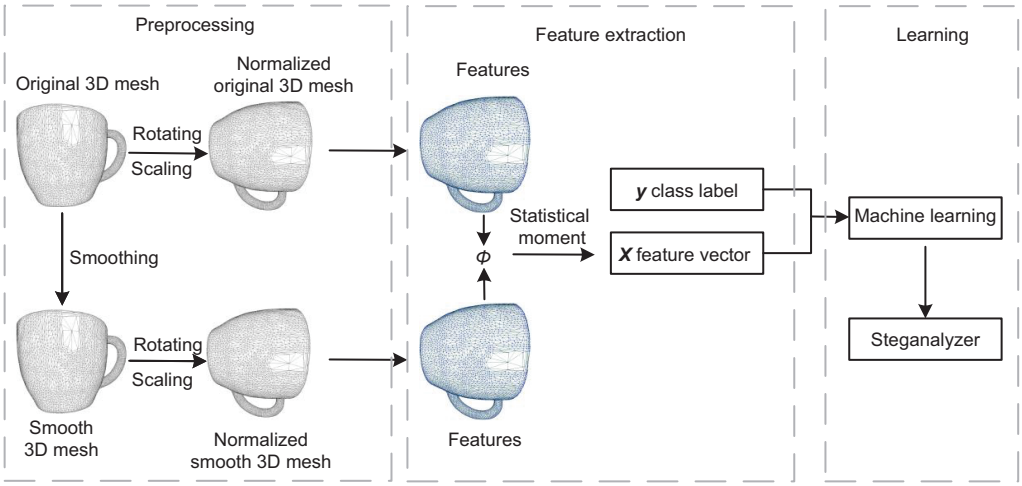


Fig. 1 Generic steganalysis flowchart

the model in a more targeted way, a local feature enhancement method is proposed in this paper.

2. In the feature extraction section, we describe how we carried out correlation analysis and screening on the basis of the original 3D mesh steganalysis feature set, and add new features that are more suitable for 3D point cloud steganalysis to form a 3D point cloud steganalysis feature set.

3 The proposed algorithm

3.1 Algorithmic framework

The point cloud steganalysis algorithm based on composite operator feature enhancement consists mainly of the following three stages, preprocessing based on feature enhancement, feature set screening and extraction, and steganalysis-based feature learning, as shown in Fig. 2.

3.1.1 Preprocessing based on feature enhancement

First, the model is smoothed using global smoothing, and the smoothed model is normalized and standardized. Second, key points are extracted for the normalized initial 3D point cloud and the normalized smoothed point cloud: the 3D Harris operator is used to extract the corner point information in the point cloud models, and then the ISS operator is combined to extract the significant feature points among the non-corner points as the key points in the feature extraction region. Finally, the multi-level neighborhoods around the key points are calculated, and the extracted region is processed with hierarchical feature enhancement using different feature enhancements in different neighborhoods to obtain the enhanced 3D point cloud.

3.1.2 Feature set screening and extraction

The existing 3D mesh feature set is filtered and added to form the 3D point cloud steganalysis feature set, and feature extraction is performed on the enhanced 3D point cloud.

3.1.3 Steganalysis-based feature learning

An integrated learner is built to perform steganalysis on the extracted feature information.

3.2 Local feature enhancement

3.2.1 Smoothing of 3D point clouds

Most of the 3D models containing secret information in steganalysis have very little noise as a result of steganographic operations; these models are essentially fine outliers, and the significance of smoothing operations in steganalysis models is to eliminate these fine outliers and restore the original model. The Laplace smoothing used in the smoothing part of the traditional 3D model steganalysis algorithm has a weak adaptability to different categories of models. It is difficult to eliminate the normal outliers of some complex 3D models, such as 3D human body models, and the stability of the

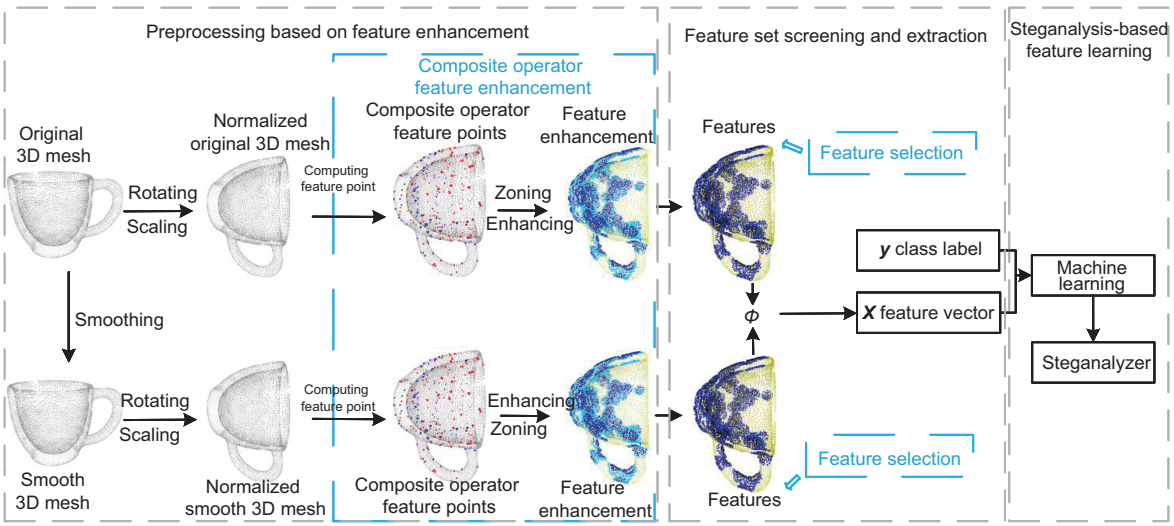


Fig. 2 The 3D point cloud steganalysis algorithm based on composite operator feature enhancement

smoothing effect is not good. Therefore, an adaptive bilateral filtering 3D point cloud smoothing algorithm based on normal outliers is used to perform smoothing operations on subtle outliers by automatically adjusting the filtering parameters through normal vector outliers (Chen et al., 2023). The algorithm improves the effectiveness and stability of the smoothing effect for different models while avoiding the influence of non-fine outliers to the greatest extent, which is more in line with the demands of steganalysis for the restoration of the original 3D point cloud.

After obtaining the smoothed 3D point cloud model, the smoothed model and the original model are normalized. Then the smoothing part of the local feature enhancement method can be accomplished by rotating the 3D models to the principal axis direction using principal component analysis.

3.2.2 Filtering feature-enhanced regions

The enhancement region is selected by first calculating the feature points using the improved 3DHarris-ISS composite operator, and second calculating the region that needs feature enhancement based on the feature points.

The traditional 3D Harris operator aims to extract single-scale corner points, which can effectively extract model edge feature points, but it does not have scale invariance. Improvement of the algorithm using the Gaussian pyramid theory can make the extracted feature points scale-invariant, to extract the key points under multi-resolution model for steganography analysis. The steps are as follows:

Step 1: constructing the Gaussian pyramid of the point cloud. The initial 3D point cloud and the smoothed 3D point cloud are used as the first level of the Gaussian pyramid, i.e., level G_0.

First, Gaussian blurring is performed on the point cloud. The point cloud model of the previous level is convolved with the 3D Gaussian kernel in a convolution operation to reduce the high-frequency information of the point cloud to obtain the blurred point cloud:

$$P'(x,y,z)=\frac{1}{(2\pi\sigma_x\sigma_y\sigma_z)^{3/2}}\iiint P(x,y,z)\cdot \mathrm{e}^{-\frac{(x-x')^2}{\sigma_x^2}-\frac{(y-y')^2}{\sigma_y^2}-\frac{(z-z')^2}{\sigma_z^2}}\mathrm{d}x'\mathrm{d}y'\mathrm{d}z', \tag{1}$$

where $P'(x,y,z)$ is the 3D point cloud after Gaussian blurring, $P(x,y,z)$ is the 3D point cloud at the previous level, and σ_x,σ_y, and σ_z represent the standard deviation of the Gaussian kernel on each coordinate axis.

Second, the Gaussian blurred point cloud $P'(x,y,z)$ is downsampled to reduce the 3D point cloud resolution to obtain the next-level point cloud:

$$P''(x,y,z)=P'\left(\frac{x}{2},\frac{y}{2},\frac{z}{2}\right), \tag{2}$$

where $P''(x,y,z)$ is the Gaussian pyramid point cloud at the next level.

Repeating the above Gaussian blurring and downsampling operations can obtain the point cloud model with different resolutions, i.e., the complete Gaussian pyramid. For the steganalysis neighborhood, the four-layer Gaussian pyramid can meet the demand of extracting 3D point cloud features at different resolutions, so this study chose the four-layer (G_0–G_3) Gaussian pyramid to achieve scale invariance, and the schematic of a Gaussian pyramid is shown in Fig. 3.

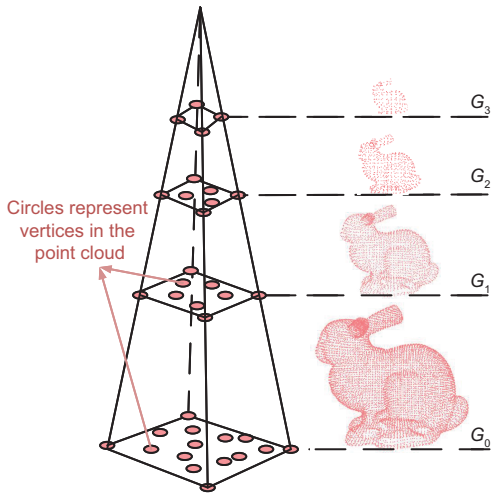


Fig. 3 Schematic of a point cloud Gaussian pyramid

Step 2: discretization of point cloud data at all levels. As shown in Fig. 4, the overall 3D point cloud models are voxelized into small voxel blocks using the octree algorithm for subsequent local feature extraction in voxel blocks.

Step 3: gradient calculation. The gradient of the vertices is calculated in each minimal voxel block after voxelization. The formulas for calculating the

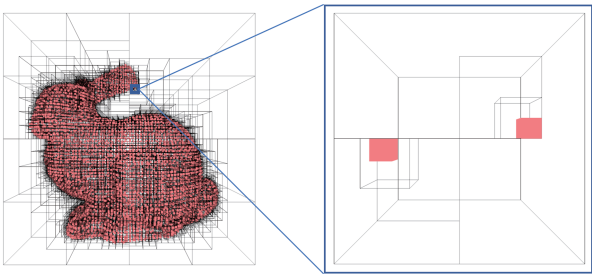


Fig. 4 3D point cloud voxelized raster schematic

gradient of the vertices in the 3D point cloud are

$$\begin{aligned} I_x &= \frac{\partial \iota}{\partial x} = \iota(x+1,y,z) - \iota(x-1,y,z), \\ I_y &= \frac{\partial \iota}{\partial y} = \iota(x,y+1,z) - \iota(x,y-1,z), \\ I_z &= \frac{\partial \iota}{\partial z} = \iota(x,y,z+1) - \iota(x,y,z-1), \end{aligned} \tag{3}$$

where ι represents the intensity or gray scale of the points in the point cloud. However, most of the 3D point cloud datasets in the field of information steganography, represented by the Model40 dataset proposed by Stanford University, contain only coordinate information and lack intensity or grayscale information, so a transformation is needed to calculate the gradient. The gradient of the vertices in a 3D point cloud can be approximated using the coordinate difference method:

$$\begin{aligned} I_x &= x_{\text{right}} - x_{\text{left}}, \\ I_y &= y_{\text{up}} - y_{\text{down}}, \\ I_z &= z_{\text{front}} - z_{\text{back}}, \end{aligned} \tag{4}$$

where x_{right}, x_{left}, y_{up}, y_{down}, z_{front}, and z_{back} are the coordinates of the nearest neighbors of the vertex in six directions, right, left, up, down, front, and back, respectively.

Step 4: constructing the Harris matrix. A 3×3 Harris matrix is constructed using the computed vertex gradients:

$$\boldsymbol{M} = \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{bmatrix}. \tag{5}$$

Step 5: corner point detection. To reduce redundancy and improve the efficiency and accuracy of the algorithm, non-maximum suppression (NMS) is used to screen the corner points. First, the Harris response value R is calculated to obtain the candidate feature points:

$$R = \det(\boldsymbol{M}) - s \cdot \text{trace}^2(\boldsymbol{M}), \tag{6}$$

where $\det(M)$ is the determinant of the Harris matrix $\boldsymbol{M}$, $\text{trace}(\boldsymbol{M})$ is the trace of the Harris matrix $\boldsymbol{M}$, and s is an empirical constant that has been verified by subsequent experiments to usually work best with $s = 0.04$.

Feature points are selected using the NMS method, with the aims of filtering out pixels with the highest response values from the point cloud and eliminating the response values of other neighboring pixels to ensure that the final retained corner points are the ones with the strongest local response values. A localization with radius r is defined to detect the feature points within the neighborhood of each candidate point. By comparing the Harris response values of other corner points in the neighborhood with the candidate points, only the point with the largest response value R in the domain is retained. The effect of different values of NMS radius r on the number of feature points and accuracy will be discussed in the subsequent experiments.

Step 6: multi-scale feature merging. The corner points are detected separately for each level of the point cloud in the Gaussian pyramid, and the feature points in each level are merged according to the Harris response value R of the corner points of the point cloud at each level. The final filtered feature points are the 3D Harris feature points with scale invariance.

The ISS operator can effectively extract the local feature points in the non-edge region of the 3D point cloud, which itself has good scale invariance, and therefore does not need to be processed in this aspect. The steps of the ISS operator are as follows:

Step 1: eigen-covariance matrix. Compute the eigen-covariance matrix of the points within the neighborhood of each point $\boldsymbol{p}_i$ in the 3D point cloud:

$$\boldsymbol{C}_i = \frac{1}{k} \sum_{j=1}^{k} (\boldsymbol{p}_j - \boldsymbol{p}_i)(\boldsymbol{p}_j - \boldsymbol{p}_i)^{\mathrm{T}}, \tag{7}$$

where k is the number of points in the neighborhood of $\boldsymbol{p}_i$, and $\boldsymbol{p}_j$ $(j = 1, 2, \cdots, k)$ are the points in the neighborhood of $\boldsymbol{p}_i$.

To accurately represent the local 3D point cloud features, after obtaining the covariance matrix, each

covariance matrix $\boldsymbol{C}_i$ is eigen-decomposed to find the eigenvectors $\boldsymbol{e}_1$–$\boldsymbol{e}_3$ and eigenvalues λ_1–λ_3.

Step 2: building the feature descriptor. Using the eigenvectors obtained from the feature decomposition and the eigenvalues, the feature descriptor **ISS** is constructed:

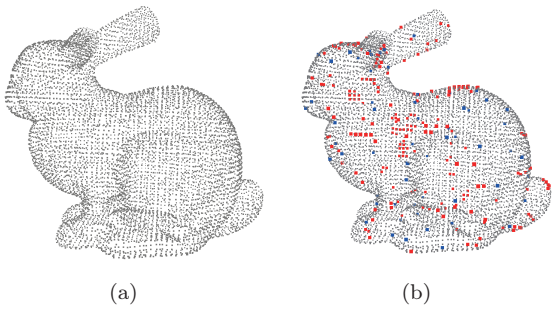
$$\mathbf{ISS}_i = [\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3, \lambda_1, \lambda_2, \lambda_3], \tag{8}$$

where the eigenvectors $\boldsymbol{e}_1$–$\boldsymbol{e}_3$ indicate the main direction of the local shape, and the eigenvalues λ_1–λ_3 express the scaling information of the local shape.

Step 3: feature matrix aggregation. Repeat steps 1 and 2 to obtain the feature description matrices of all the points and aggregate the feature description matrices of each point to obtain all the ISS feature points.

Combine the 3D Harris feature points and ISS feature points. Take the concatenation set of these points to obtain the points that need to be targeted for analysis in the 3D point cloud. Fig. 5 shows the extracted feature points, where the red points are the feature points extracted by 3D Harris, and the blue ones are the feature points extracted by the ISS operator.

To avoid omitting the neighborhood information of the feature points, a three-ring neighborhood expansion of the feature points is performed using the K-nearest neighbor algorithm to serve as the enhancement region for subsequent feature enhancement. The first-ring neighborhood is the q nearest points to the feature point, the second-ring domain is the $2q$ nearest points to the feature point, and the third-ring neighborhood is the $3q$ nearest points to the feature point. q is between 0.1% and 1% of the total number of vertices in the point cloud as derived from subsequent experiments, and the value of q will be explored in Section 4.3. Fig. 6 shows the schematic of the feature point and its three-ring neighborhood.



(a) (b)

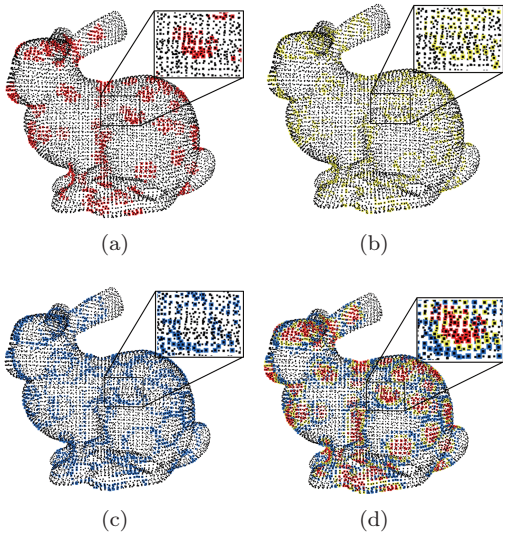
Fig. 5 The original model and the extracted feature points: (a) original 3D point cloud; (b) 3DHarris-ISS feature points (References to color refer to the online version of this figure)

3.2.3 Feature enhancement of feature points

Feature enhancement is often used for the preprocessing of point cloud data, but 3D models are rich in application scenarios, and it is difficult for general-purpose algorithms to perform targeted enhancement on dense 3D point clouds. In this study, we propose a 3D point cloud feature enhancement algorithm based on the position of the center feature point and normal guidance. The feature enhancement region in the 3D point cloud is adjusted by adding the priori information, such as the coordinates of the center feature point and the normal vector, before feature enhancement, to highlight the regions that may contain hidden information under the premise of ensuring the consistency and continuity of data (Nie et al., 2019).

First, an objective function is constructed for integrating the positions of the point cloud feature points and the normal vectors to form the priori information, which provides the basis for the subsequent feature enhancement:

$$\arg\min(\lambda E_{\mathrm{p}} + (1-\lambda)E_{\mathrm{n}}), \tag{9}$$



(a) (b) (c) (d)

Fig. 6 Schematic of the three-ring neighborhood segmentation: (a) neighborhood of the first ring; (b) neighborhood of the second ring; (c) neighborhood of the third ring; (d) integration of neighborhoods

where E_{p} denotes the positional constraints of the coordinate points of the point cloud model, E_{n} denotes the normal vector constraints, and λ is used to balance the importance of them.

The position constraint term is used to constrain the difference between the position of the enhanced feature point and the position of the center feature point:

$$E_{\mathrm{p}}=\sum_{i=1}^{n}\|\hat{\boldsymbol{p}}_i-\tilde{\boldsymbol{p}}_i\|^2, \tag{10}$$

where n is the number of points in all feature enhancement areas, $\hat{\boldsymbol{p}}_i$ is the position of the augmented post point, and $\tilde{\boldsymbol{p}}_i$ is the origin position.

The normal vector constraint term is used to cause the augmented point normal vector to be close to the normal vector of the origin:

$$E_{\mathrm{n}}=\sum_{i=1}^{n}\|\hat{\boldsymbol{n}}_i-\tilde{\boldsymbol{n}}_i\|^2, \tag{11}$$

where $\hat{\boldsymbol{n}}_i$ is the normal vector of the enhanced feature points and $\tilde{\boldsymbol{n}}_i$ is the normal vector of the origin.

Nonlinear optimization in the objective function makes the results unpredictable and greatly decreases the performance of the algorithm. The normal vector constraint term is expressed as a constraint in which the enhanced feature point-to-neighborhood point vectors are orthogonal to the desired normal vectors. Assuming that the normal vectors are parallel to the local tangent planes of the enhanced feature area, the objective function is optimized such that the normals are as perpendicular as possible to the normal vectors of the points in the neighborhood of the enhanced feature point, and the normal vector constraints are

$$\arg\min\Big(\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{k}\|\tilde{\boldsymbol{n}}_i(\hat{\boldsymbol{p}}_i-\hat{\boldsymbol{p}}_j)\|^2\Big), \tag{12}$$

where $\hat{\boldsymbol{p}}_j$ are the coordinates of the points in the neighborhood of $\hat{\boldsymbol{p}}_i$.

The new objective function is

$$\begin{aligned}\arg\min\Big(&\sum_{i=1}^{n}\Big(\lambda\|\hat{\boldsymbol{p}}_i-\tilde{\boldsymbol{p}}_i\|^2\\&+(1-\lambda)\sum_{j=1,j\neq i}^{k}\|\tilde{\boldsymbol{n}}_i(\hat{\boldsymbol{p}}_i-\hat{\boldsymbol{p}}_j)\|^2\Big)\Big).\end{aligned} \tag{13}$$

The optimized objective function is a system of linear equations containing the unknowns of the position of the feature point after enhancement $\hat{\boldsymbol{p}}_i$. The optimal location of the enhanced feature point $\hat{\boldsymbol{p}}_i$ is solved by linear least squares to realize feature enhancement.

When performing data enhancement, if enhancement is performed only once on the neighboring region, it will create a clear step between the enhanced and unenhanced regions. Doing stepwise reduction of enhancement intensity from the center of the enhanced region to the unenhanced region can effectively prevent the appearance of repeated enhancement points. Defining m as the iterative base of feature enhancement for different neighborhoods, the first-, second-, and third-ring neighborhoods of the 3D point cloud are enhanced for $3m$, $2m$, and m iterations, using the above enhancement method, and are defined as the large-, medium-, and small-amplitude enhancement regions, respectively. Except for the three-ring neighborhood region mentioned above, the points in the rest of the regions remain unchanged. Fig. 7 shows the model after feature enhancement.

Fig. 7 Schematic of feature enhancement

3.3 Feature set screening and analysis

The most widely used spatial domain feature set is ELFS124, containing 124-dimensional features, which are obtained by multiplying the model features with four-dimensional statistical moments of mean, variance, skewness coefficient, and kurtosis.

In the ELFS124 feature set, only 52-dimensional features are related to 3D model vertices, and the

rest are edge features or face features. These 52-dimensional features are the four-dimensional statistical moments of the vertex paradigm in the Cartesian coordinate system, the four-dimensional statistical moments in the Laplace coordinate system, the four-dimensional statistical moments of the vertex paradigm in the Laplace coordinate system, the four-dimensional statistical moments of the vertex normals, the four-dimensional statistical moments of the vertex Gaussian curvature, and the four-dimensional statistical moments of the vertex coordinates in the spherical coordinate system. Therefore, the 52-dimensional point features are retained and used as control variables for the bias correlation analysis of the remaining edge features and face features. The partial correlation coefficients and partial correlation matrices of the remaining face and edge features are calculated, and the features with correlation coefficients $\geq$0.7 are excluded. The remaining edge features and face features that need to be converted are the four-dimensional statistical moments of curvature ratios and the four-dimensional statistical moments of angles and lengths in the spherical coordinate system. At the same time, the 12-dimensional feature vectors of the nearest-neighbor edge length, nearest-neighbor normal vector, and four-dimensional statistics of domain density are added to compensate for the decrease in accuracy due to the loss of edge features.

3.3.1 Curvature ratio feature transformation

The curvature ratio in the original ELFS124 is calculated by using the Gaussian curvature of the facet and the mean curvature, but there is no information that can be directly used to describe the curvature ratio in the 3D point cloud models. Therefore, we adopt the 3D point cloud coordinates of the points and their neighboring point information to calculate the curvature ratio; its calculation steps are as follows:

Step 1: compute the covariance matrix for each point. For each point $\boldsymbol{p}$ in the point cloud, the K-nearest neighbor algorithm can be used to derive the set of neighboring points around it and compute the covariance matrix $\boldsymbol{C}$ of $\boldsymbol{p}$.

Step 2: compute the eigenvalues and eigenvectors of the covariance matrix. Perform an eigenvalue decomposition of the covariance matrix $\boldsymbol{C}$ to obtain its eigenvalues λ_1 and λ_2.

Step 3: calculate the minimum and maximum principal curvatures. The minimum principal curvature $k_{\min}$ and the maximum principal curvature $k_{\max}$ can be calculated by Eqs. (14) and (15), respectively:

$$k_{\min}=\frac{1}{2h+\sqrt{4g^2+h^2}}, \tag{14}$$

$$k_{\max}=\frac{1}{2h-\sqrt{4g^2+h^2}}, \tag{15}$$

$$h=(\lambda_1+\lambda_2)/\,2, \tag{16}$$

$$g=\lambda_1\lambda_2, \tag{17}$$

where h is the mean curvature and g is the Gaussian curvature.

Step 4: calculate the curvature ratio. The curvature ratio is calculated by

$$k_{\mathrm{r}}=\frac{k_{\min}}{k_{\max}}. \tag{18}$$

The curvature ratio k_{r}, usually between 0 and 1, indicates the relative distribution of curvature around point $\boldsymbol{p}$. k_{r} close to 0 indicates a significant change in curvature in the vicinity of point $\boldsymbol{p}$. k_{r} close to 1 indicates a small change in curvature.

3.3.2 Spherical coordinate edge length feature transformation

In ELFS124, edge lengths are obtained from the coordinates of connected vertices using the Euclidean distance formula. The point cloud is composed of a series of discrete points, and there is no direct connection between the points, so it is impossible to obtain the effective edge information directly. Therefore, we propose to complete the computation of coordinate edge length information in spherical coordinates with the help of the neighborhood in the following steps:

Step 1: convert the point cloud coordinates to spherical coordinates (ρ, θ, φ) using the spherical coordinate formula.

Step 2: calculate the neighborhood of each feature point. The set of neighboring points around the feature point is computed using the K-nearest neighbor algorithm.

Step 3: calculate the edge length between the feature point and its neighbors. Calculate the Euclidean distance d between the point and the points in its neighborhood.

Step 4: calculate the average edge length of the neighborhood. Repeat step 3 to obtain the edge lengths of the back edges connected to all the points in the neighborhood and compute their average. The neighborhood average edge length $\bar{d}$ is the effective edge length in the spherical coordinate system of the point cloud model.

3.3.3 Neighborhood local features

Although the feature vectors with large correlation coefficients with the point features are removed in the correlation analysis, the large number of missing edge features will reduce the accuracy. To further deal with the problem of decreasing detection accuracy caused by missing edge features, we add new feature vectors that can represent the edge features, as shown in Table 1.

The 52-dimensional vertex features filtered from existing features, the transformed eight-dimensional features, and the newly added 12-dimensional neighborhood local features together form a 72-dimensional point cloud feature vector called POINT72, i.e., $\boldsymbol{X} = [x_1, x_2, ..., x_{72}]$.

POINT72 dataset is used to extract features from the initial 3D point cloud and the smoothed 3D point cloud, and the residuals are calculated to obtain the 3D point cloud feature vectors $\boldsymbol{X}$. The 3D point cloud steganalysis tool is designed by placing the set of feature vectors $\boldsymbol{X}$ and the data labels $\boldsymbol{y}$ into the integrated learner to train the model.

4 Experimental results and analysis

4.1 Experimental setup

The experiments were performed using 2000 pieces of data from the Princeton dataset for 3D point cloud conversion, which contains many types of 3D point cloud data including human body, animals, household items, and monsters, as shown in Fig. 8. The experiments focused on testing and analyzing the accuracy and area under the curve (AUC) of the 3D point cloud steganalysis algorithm based on feature enhancement by composite operators. The experiments were conducted on a Windows platform using Microsoft Visual Studio 2022, PyCharm Community 2022.03, and Meshlab 2021.

The experiments were divided into two parts: the effect of algorithm parameters on the accuracy and AUC and the comparison with other steganalysis algorithms. The algorithm parameters included NMS radius of the 3D Harris operator, feature point three-ring neighborhood range, and the effectiveness of feature enhancement and local features, as shown in Table 2.

Because the current steganalysis algorithms for 3D models are all based on 3D mesh models, there is no steganalysis algorithm based on 3D point cloud models. Therefore, four 3D mesh steganalysis algorithms are used for comparison: LFS76, LFS124, WFS228, and NVT+. In the 3D mesh steganalysis algorithm, the original 3D mesh model of the dataset is used for the experiments; in the 3D point cloud steganography algorithm proposed in this study, the 3D point cloud model with the edge information and face information removed based on the 3D mesh model is used for the experiments.

Table 2 Default parameters in the experiments

Notation	Meaning	Optimum value
r	Radius of non-maximum suppression	0.028
s	Empirical constant	0.04
q	Neighborhood range	0.003
λ	Equilibrium coefficient for feature enhancement	0.7
m	Iterative base for hierarchical enhancement	2

Table 1 Additional features and their definitions

New feature	Definition
Length of the nearest neighbor	Calculate the edge lengths of the constituent nearest-neighbor edges of each point and its nearest-neighbor points, i.e., the nearest-neighbor edge lengths. Compute the four-dimensional statistical moments of the nearest-neighbor edge lengths as a set of feature vectors
Nearest-neighbor normal vector	Compute the normal vectors of the above nearest-neighbor edges and use their four-dimensional statistical moments as a set of eigenvectors
Neighborhood density	Neighborhood density is represented by the number of points in a one-ring neighborhood, and its four-dimensional statistical moments are used as a set of feature vectors

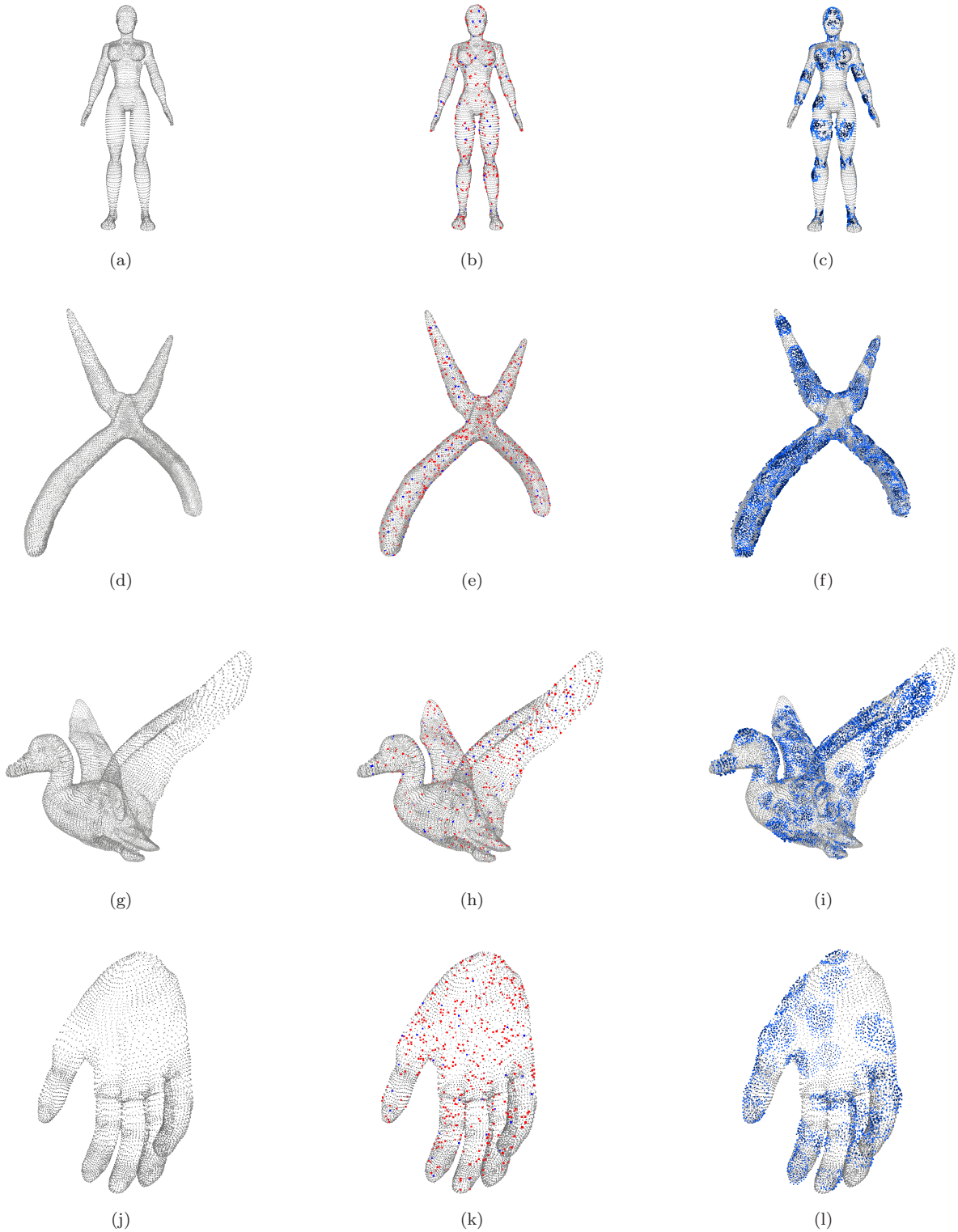


Fig. 8 Schematic of the point cloud model: (a) the original 3D point cloud of mannequin; (b) the characterization points of the mannequin; (c) the feature-enhanced mannequin; (d) the original 3D point cloud of tool model; (e) the characterization points of tool model; (f) the feature-enhanced tool model; (g) the original 3D point cloud of animal model; (h) the characterization points of animal model; (i) the feature-enhanced animal model; (j) the original 3D point cloud of palm model; (k) the characterization points of palm model; (l) the feature-enhanced palm model

4.2 Analysis of non-maximum suppression radius values

In the 3D Harris operator, the NMS method is used to filter the feature points, and the number of filtered feature points depends on the NMS radius r. The larger the radius r, the fewer the filtered feature points which can represent the overall features; the smaller the radius r, the more the filtered feature points which can represent the local features. To make the results easy to observe, we select the bunny point cloud containing 8171 points for the observation of the number of feature points, and use the Princeton dataset to check AUC of the algorithm. The relationship among the radius r, the number of feature points, and AUC is shown in Fig. 9. Since the number of feature points selected at $r = 0.010$ is close to the number of vertices (8171) and the number of feature points selected at $r = 0.030$ is $\leq 1\%$ of the total number of vertices, to determine the optimal suppression radius, the range of the radius r is selected as 0.010–0.030, and the number of feature points and AUC of the model at different suppression radii are calculated.

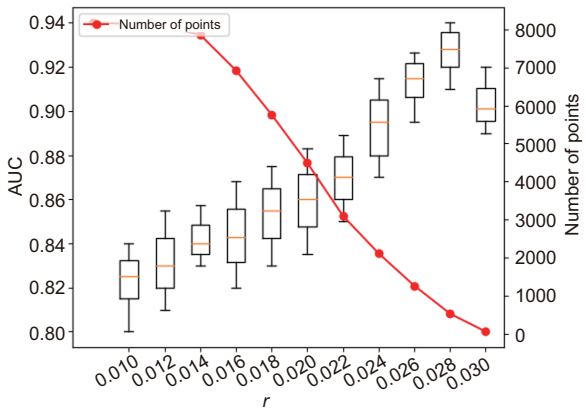


Fig. 9 Schematic of the relationship among the non-maximum suppression radius r, area under the curve (AUC), and the number of key points

As can be seen in Fig. 9, the number of feature points decreases as r increases. When $0.010 < r < 0.028$, AUC increases with increasing r. This is because there are too many feature points selected in this interval, which results in limited spacing between feature points. This makes part of the three-ring neighborhood and two-ring neighborhood overlap the enhancement region, which interferes with the detection of the feature region, and the overlapping region gradually decreases as the suppression radius r increases. As the suppression radius r increases, the overlapping region is gradually reduced and AUC is gradually improved. When $0.028 < r < 0.030$, the suppression radius r is not sufficient to screen out all the feature points, which results in a rapid decrease in AUC as r increases. When $r = 0.028$, the confidence interval is optimal and 531 feature points of the composite operator are extracted.

The effect of the NMS radius on the performance of the algorithm and the number of key points extracted is shown in Table 3.

Table 3 Analysis of the effect of non-maximum suppression radius on efficiency

r	AUC	Number of key points	Detection time (s)
0.010	0.8228	8171	4.51
0.012	0.8316	8066	4.35
0.014	0.8424	7855	4.22
0.016	0.8436	6921	4.10
0.018	0.8534	5759	3.93
0.020	0.8608	4492	3.78
0.022	0.8691	3098	3.56
0.024	0.8942	2113	3.32
0.026	0.9140	1264	3.21
0.028	**0.9267**	531	3.11
0.030	0.8997	58	3.02

Best result is in bold

As can be seen from Table 3, the time complexity of the algorithm increases tremendously when more keypoints are extracted, and the time required is 3.11 s at a non-maximum suppression radius $r = 0.028$.

4.3 Neighborhood range analysis

When selecting the three-ring neighborhood for feature enhancement, different neighborhood ranges determine the size of the enhancement range: a too small feature enhancement region will result in being unable to effectively enhance the feature points, and a too large feature enhancement region will result in the appearance of over-enhanced points. In this study, the neighborhood is divided into three-ring neighborhoods, and in the experiment, the number of K-nearest neighbor points in the first-ring neighborhood is q, the number of K-nearest neighbor points in the second-ring neighborhood is $2q$, and the number of K-nearest neighbor points in the

third-ring neighborhood is $3q$. Taking the current grid steganography detection algorithm's minimum detection accuracy of 84% as the baseline, q is considered to be invalid if it takes a value that makes the minimum accuracy $\leq$84%. Fig. 10 shows the valid values of q ranging from 0.1% to 0.7% of the total number of points in the point cloud.

From Fig. 10, it can be seen that when $0 < q < 0.003$, AUC is rapidly improved; with the expansion of the neighborhood, more local features are amplified, especially the difference between vertex normal vector and vertex normal form features caused by steganography algorithm. In addition, when $0.003 < q < 0.010$, some over-enhanced points appear as the neighborhood range increases, which leads to a gradual decrease in the confidence interval, due to the overlapping phenomenon mentioned above in some of the two-loop neighborhoods. When $q > 0.007$, some of the first-ring neighborhood areas overlap, which will cause severe interference. Interference causes the average accuracy to fall off a cliff after $q > 0.008$, which is even lower than the average accuracy without feature augmentation ($q = 0$). Fig. 11 shows a schematic of the interference region. The confidence interval reaches the optimum at $q = 0.003$, when the neighborhood range reaches the maximum range without overlapping interference.

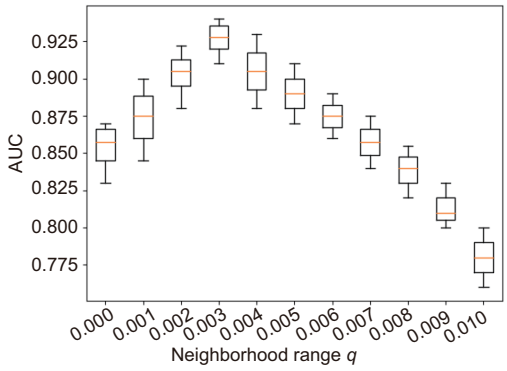


Fig. 10 Schematic of the relationship between neighborhood range and area under the curve (AUC)

Because the change of neighborhood range will cause the number of neighborhood points of the key point to change, thus affecting the algorithm's time complexity, the relationship between the neighborhood range and the algorithm's time complexity is experimentally explored. The results are shown in Table 4.

As can be seen from Table 4, the time complexity of the algorithm increases as the neighborhood range gradually increases. At neighborhood range $q < 0.003$, the time complexity increases slowly because no interference region occurs, while at $q > 0.003$, the time complexity increases rapidly as the interference region gradually increases.

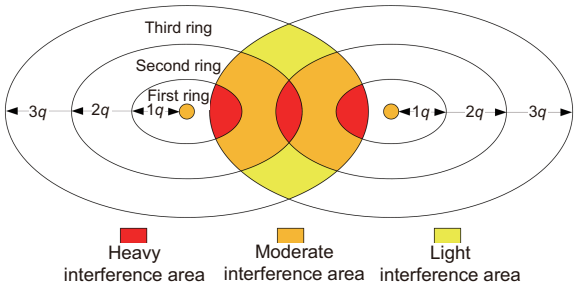


Fig. 11 Schematic of interference region

Table 4 Analysis of the effect of neighborhood range on efficiency

q	AUC	Detection time (s)
0.000	0.8537	2.64
0.001	0.8738	2.78
0.002	0.9030	2.99
0.003	**0.9267**	3.11
0.004	0.9035	3.34
0.005	0.8810	3.57
0.006	0.8759	3.89
0.007	0.8631	4.21
0.008	0.8383	4.53
0.009	0.8133	4.87
0.010	0.7812	5.26

Best result is in bold

Fig. 12 shows the average accuracy of the algorithm when considering the non-extremely large suppression radius and the neighborhood range. It can be seen from Fig. 12 that the 3D surface has two obvious ridges, which correspond to the average accuracy of the algorithm when the suppression radius r is 0.028 and the neighborhood range q is 0.003; the highest AUC of 0.9267 is achieved at the point $(r = 0.028, q = 0.003)$.

4.4 Experiments on the effectiveness of steganographic feature sets

The 3D point cloud steganalysis feature set consists of two parts: the original 52-dimensional vertex features in the 3D mesh feature set are retained and added to the converted eight-dimensional features to form POINT60, and the 12-dimensional local neighborhood features are added to form POINT72.

To analysis the effect of each feature set on AUC, Fig. 13 compares the four feature sets mentioned

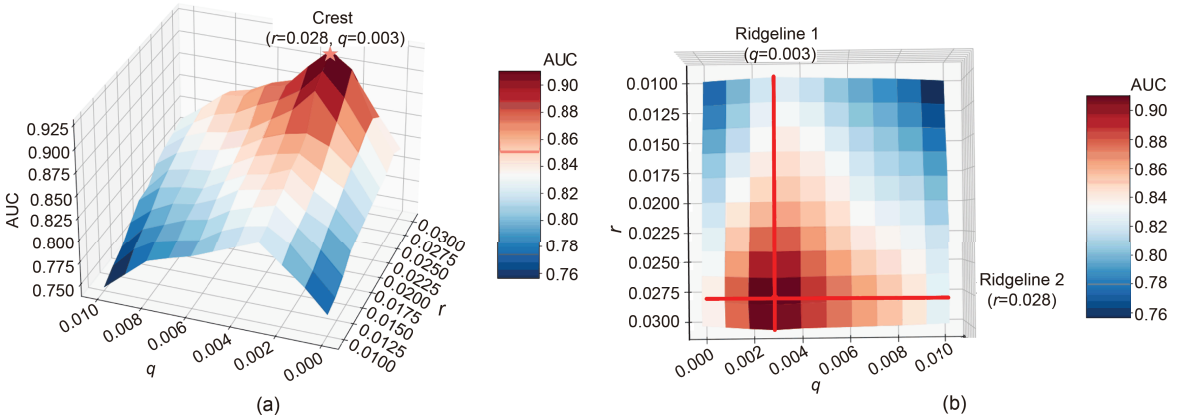


Fig. 12 Suppression radius and neighborhood range versus area under the curve (AUC): (a) crest; (b) ridgeline

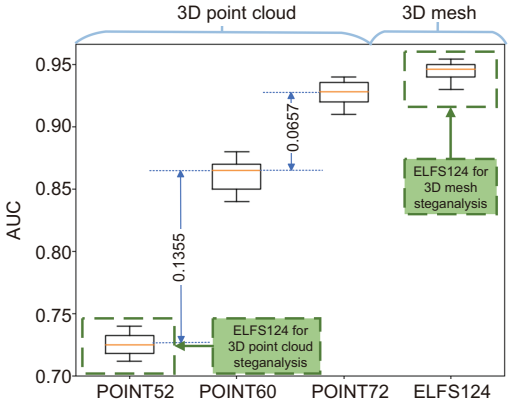


Fig. 13 Area under the curve (AUC) for different feature sets

above. Among them, POINT52 represents the detection capability of 3D point cloud by ELFS124, the best feature set of current 3D grid steganography. The AUC of the ELFS124 is derived using 3D mesh models as the dataset, and the rest of the feature sets use 3D point clouds as the dataset.

After the feature enhancement, the curvature features are greatly enhanced. Fig. 13 shows that after adding the converted curvature and the average side length of an eight-dimensional feature neighborhood to POINT60, AUC is improved by 18.67% compared with that based on the original POINT52. After adding 12-dimensional neighborhood local features, AUC is increased by 7.63% to compensate for the decrease in AUC caused by the lack of some edge features. AUC of the proposed algorithm is still lower than that of ELFS124 after adding the local neighborhood features, because the added neighborhood features cannot fully describe the edge features and surface features lost in the process of converting 3D mesh to 3D point cloud.

From the confidence interval of the receiver operating characteristic (ROC) curve of the POINT52 feature set for the steganalysis of 3D point cloud models, it can be seen that even the best 3D mesh steganalysis feature set, ELFS124, still has a low AUC for 3D point cloud steganalysis, whereas POINT72 proposed in this study greatly improves the AUC of 3D point cloud steganalysis.

The maximum mean difference (MMD) of the feature sets of the original model and the encrypted model is used to compare the effect of steganalysis for each feature set, where the larger the MMD is, the larger the gap is between the original model and the encrypted model, and the easier it is to detect the differences caused by steganography in feature sets.

As can be seen from Table 5, there is still a 0.007 gap between the MMDs of POINT72 and ELFS124. However, the MMD of POINT72 is 63.16% and 21.57% higher than that of POINT52 and POINT60, respectively, which indicates that the feature-converted features and the added features are effective and compensate for part of the AUC due to the missing edge features and face features.

Table 5 Feature set maximum mean difference (MMD) comparison

Steganalysis feature set	MMD
POINT52	0.038
POINT60	0.051
POINT72	0.062
ELFS124	0.069

4.5 Feature enhancement validity experiments

In the process of feature enhancement, AUC of the steganalysis model without feature-in-feature enhancement is compared with the AUC of the steganalysis model with feature enhancement. The results are shown in Fig. 14.

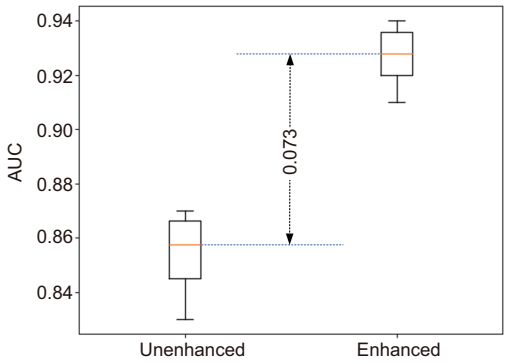


Fig. 14 Feature enhancement validity experiments

The role of feature enhancement for steganalysis is further analyzed using MMD and the results are shown in Table 6.

Table 6 Feature enhancement validity

Steganalysis feature set	MMD
Unenhanced POINT72	0.056
Enhanced POINT72	0.062

MMD: maximum mean difference

From Table 6, it can be seen that MMD of the feature vectors after enhancement is improved by 10.71% compared with that of the pre-enhancement vectors; that is, the steganalysis can be performed more efficiently.

4.6 Comparative experiments on 3D mesh steganalysis algorithms

To test the steganography algorithm proposed in this paper, LFS76, LFS124, WFS228, and NVT+ were selected as the comparison algorithms. Boxplot of the confidence interval area under ROC was plotted to compare the performance of each algorithm.

Fig. 15 shows that the AUC of POINT72 for the four steganographic algorithms has similar confidence areas as the existing 3D mesh steganalysis tools, which is still lower than that of LFS124. Compared with the other four 3D mesh steganalysis algorithms, POINT72 lacks edge and face information, which makes it extremely difficult to characterize the domain correlation between points, and therefore the AUC of POINT72 is lower than that of LFS124 among the four steganalysis algorithms. However, through the incorporation of neighboring features and feature enhancement, the AUC of 3D point cloud steganalysis is as close as possible to that of the existing 3D mesh steganalysis analyzers.

5 Conclusions

In this paper, we propose a composite operator based feature enhancement algorithm for 3D point cloud steganalysis that uses an adaptive bilateral filtering point cloud smoothing algorithm based on normal outliers. An improved 3DHarris-ISS composite operator based on a Gaussian pyramid is included along with a point cloud feature enhancement algorithm based on the location of the center feature point and normal guidance to form a local feature enhancement method. This is used to enhance the local features of the 3D point cloud model. The 3D point cloud feature set POINT72 is constructed by filtering and transforming the feature set ELFS124 and adding neighborhood features. The steganalysis method in this paper compensates for the gap in previous steganalysis algorithms for 3D point clouds, and has strong detection ability for 3D point cloud steganalysis models. Due to the lack of edge and face information, the 3D point cloud steganalysis model has a slightly lower detection accuracy than the 3D mesh steganalysis model. Future research directions include expanding the feature enhancement approach to 3D meshes to improve the accuracy of 3D mesh steganalysis, expanding the feature set of 3D point cloud models to improve the accuracy of 3D point cloud steganalysis, and exploring end-to-end 3D point cloud steganalysis models.

Contributors

Shuai REN and Hao GONG designed the research and processed the data. Hao GONG and Suya ZHENG drafted the paper. Hao GONG and Shuai REN revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

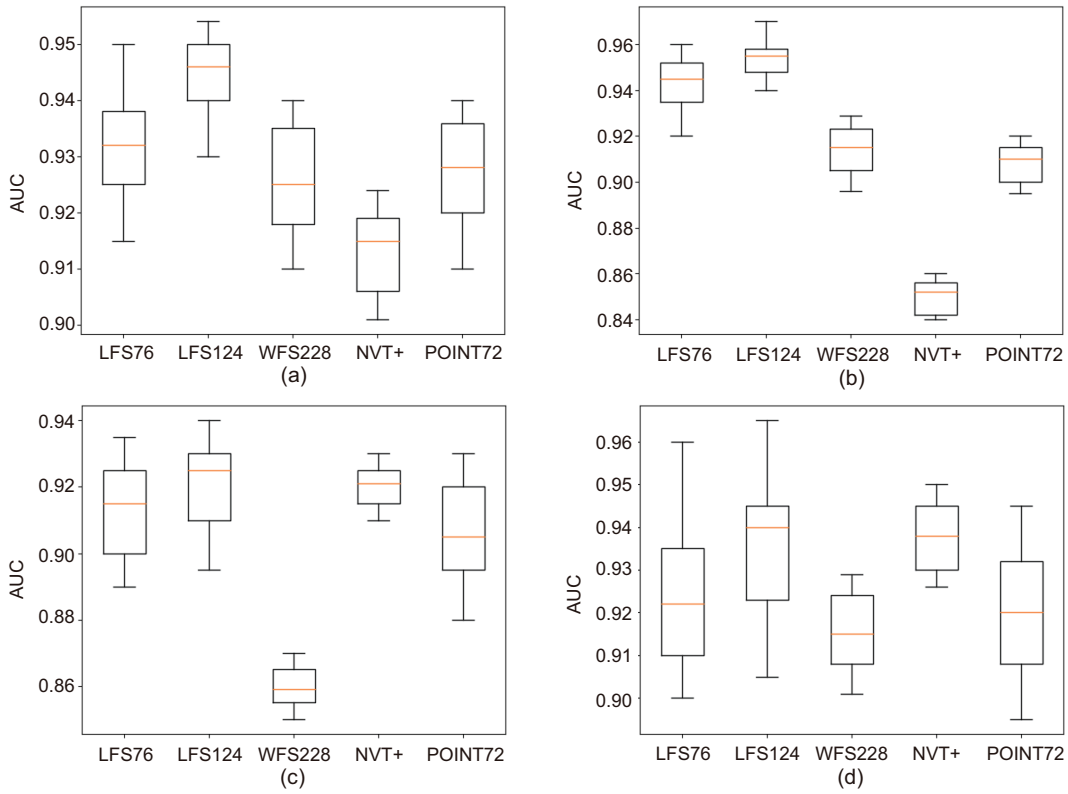


Fig. 15 AUC with 3D mesh steganalysis: (a) MRS; (b) VRS; (c) SRW; (d) MLS. AUC: area under the curve; MRS: mean of the distribution of the vertex radial distances in the spherical coordinate system; VRS: variance of the distribution of the vertex radial distances in the spherical coordinate system; SRW: steganalysis-resistant watermarking; MLS: multi-layer steganography

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Chen YC, Fan YG, Yu DF, et al., 2023. Adaptive bilateral filtering point cloud smoothing and IMLS evaluation method considering normal outliers. *J Graph*, 44(1):131-138 (in Chinese). https://doi.org/10.11996/JG.j.2095-302X.2023010131

Decker TG, Devillers RW, Gallier S, 2023. Detecting agglomeration patterns on solid propellant surface via a new curvature-based multiscale method. *Acta Astronaut*, 206:123-132. https://doi.org/10.1016/j.actaastro.2023.02.020

Li ZY, Bors AG, 2016. 3D mesh steganalysis using local shape features. Proc IEEE Int Conf on Acoustics, Speech and Signal Processing, p.2144-2148. https://doi.org/10.1109/ICASSP.2016.7472056

Li ZY, Bors AG, 2017. Steganalysis of 3D objects using statistics of local feature sets. *Inform Sci*, 415-416:85-99. https://doi.org/10.1016/j.ins.2017.06.011

Li ZY, Bors AG, 2020a. Selection of robust and relevant features for 3-D steganalysis. *IEEE Trans Cybern*, 50(5):1989-2001. https://doi.org/10.1109/TCYB.2018.2883082

Li ZY, Bors AG, 2020b. Steganalysis of meshes based on 3D wavelet multiresolution analysis. *Inform Sci*, 522:164-179. https://doi.org/10.1016/j.ins.2020.02.061

Li ZY, Gong DF, Liu FL, et al., 2018a. 3D steganalysis using the extended local feature set. Proc 25th IEEE Int Conf on Image Processing, p.1683-1687. https://doi.org/10.1109/ICIP.2018.8451643

Li ZY, Liu FL, Bors AG, 2018b. 3D steganalysis using Laplacian smoothing at various levels. Proc 4th Int Conf on Cloud Computing and Security, p.223-232. https://doi.org/10.1007/978-3-030-00021-9_21

Liu SJ, Luo FF, Li QS, et al., 2024. AWEDD: a descriptor simultaneously encoding multiscale extrinsic and intrinsic shape features. *Vis Comput*, 40:2537-2554. https://doi.org/10.1007/s00371-023-02935-6

Lowe DG, 2004. Distinctive image features from scale-invariant keypoints. *Int J Comput Vis*, 60(2):91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Mikolajczyk K, Tuytelaars T, Schmid C, et al., 2005. A comparison of affine region detectors. *Int J Comput Vis*, 65(1-2):43-72. https://doi.org/10.1007/s11263-005-3848-x

Nie JH, Zhang ZC, Liu Y, et al., 2019. Point cloud ridge-valley feature enhancement based on position and

normal guidance.
https://doi.org/10.48550/arXiv.1910.04942

Pauly M, Keiser R, Gross M, 2003. Multi-scale feature extraction on point-sampled surfaces. *Comput Graph Forum*, 22(3):281-289.
https://doi.org/10.1111/1467-8659.00675

Yang Y, Ivrissimtzis I, 2014. Mesh discriminative features for 3D steganalysis. *ACM Trans Mult Comput Commun Appl*, 10(3):27. https://doi.org/10.1145/253555

Zhong Y, 2009. Intrinsic shape signatures: a shape descriptor for 3D object recognition. Proc IEEE 12th Int Conf on Computer Vision Workshops, p.689-696.
https://doi.org/10.1109/ICCVW.2009.5457637

Zhou H, Chen KJ, Zhang WM, et al., 2021. Feature-preserving tensor voting model for mesh steganalysis. *IEEE Trans Vis Comput Graph*, 27(1):57-67.
https://doi.org/10.1109/TVCG.2019.2929041

Zhou H, Chen KJ, Zhang WM, et al., 2022. 3D mesh steganography and steganalysis: review and prospect. *J Image Graph*, 27(1):150-162 (in Chinese).
https://doi.org/10.11834/jig.210371