Steganalysis of 3D objects using statistics of local feature sets

Zhenyu Li, Adrian G. Bors*

Department of Computer Science, University of York, York, YO10 5GH, UK

ARTICLE INFO

Article history:
Received 15 February 2017
Revised 12 May 2017
Accepted 14 June 2017
Available online 15 June 2017

Keywords:
3D Objects
Steganalysis
Information hiding
Watermarking
Steganography
Local feature

ABSTRACT

3D steganalysis aims to identify subtle invisible changes produced in graphical objects through digital watermarking or steganography. Sets of statistical representations of 3D features, extracted from both cover and stego 3D mesh objects, are used as inputs into machine learning classifiers in order to decide whether any information was hidden in the given graphical object. The features proposed in this paper include those representing the local object curvature, vertex normals, the local geometry representation in the spherical coordinate system. The effectiveness of these features is tested in various combinations with other features used for 3D steganalysis. The relevance of each feature for 3D steganalysis is assessed using the Pearson correlation coefficient. Six different 3D watermarking and steganographic methods are used for creating the stego-objects used in the evaluation study.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

3D graphics are increasingly used in a variety of applications including computer graphics, virtual reality, Computer Aided Design (CAD) and 3D printing among many others. Specific information is hidden into 3D objects through digital watermarking or steganography for copyright protection, relevant information storage and marketing among many other applications. While watermarking seeks to robustly embed rather smaller codes, steganography would seek to embed larger payload messages without enforcing robustness. All these approaches aim to hide information in such a way that the changes they produce to the 3D objects are not visible. On the other hand, steganalysis is being developed in order to find whether information is embedded in a certain media data or not. There are many steganalytic algorithms being proposed for audio signals [31,32,45], digital images [12,26,36,50] and video signals [37,44]. While 3D objects can be represented in various ways, their most usual data representation is by means of meshes. Such irregular representations, modelling complex 3D shapes, are very different from the regular structural arrays representing audio, digital images or video signals. Consequently, the existing image and video steganalytic algorithms cannot be successfully applied to 3D objects.

3D watermarking involves modifying certain geometrical properties of the object, most often in a statistical manner, in such a way that there are no visible changes, while the message can be successfully retrieved during the watermark extraction stage. Research on 3D watermarking started in 1997 when Ohbuchi et al. proposed two 3D information hiding algorithms using ratios of distances calculated from the local geometry [30]. Cho et al. [10] proposed two blind robust watermarking algorithms based on modifying the mean or variance of the distribution of the vertices’ radial distances in the spherical coordinate system. Cayre and Macq proposed a steganographic approach for 3D triangle meshes in [7] by considering each triangle from the mesh as a two-state geometrical object embedding a bit. The hierarchical 3D watermarking...
methodology based on the wavelet transform was developed in [42], which includes three different algorithms, each one enforcing one of the following requirements: robustness, high-capacity and fragility for authentication. Luo and Bors proposed changing the statistics of geodesic distance distributions in 3D objects [27]. The same authors proposed minimizing the surface distortions in 3D watermarking by using an optimization algorithm in [6]. Among the information hiding algorithms, we mention a multi-layer 3D steganographic method [8] which embeds large payloads using vertices’ projections onto the principal axis of the object. This steganographic method can make use of several layers for embedding the information increasing thus significantly the embedded payload. However, not all bits embedded through this method are retrievable and some are lost. More recently, Yang et al. [49] proposed a steganalysis-resistant watermarking algorithm which embeds the payload by changing the histogram of the radial coordinates of the vertices. This watermarking method produces less embedding distortion in the 3D objects when compared to the methods proposed in [10]. A distortion-free steganographic algorithm [3] embeds the information into the meshes by permuting the order in which faces and vertices are stored. However, this algorithm is not robust to the vertex-reordering attack.

3D steganalysis only received very recently the attention of the scientific community. The 3D steganalytic approach proposed in [47] considered various features including the norms of vertices in the Cartesian and Laplacian coordinate systems [46], the dihedral angle of faces adjacent to the same edge, and the face normal. Parameters representing the statistics of these features were used as inputs to a quadratic classifier. Yang et al. [48], proposed a new steganalytic algorithm, specifically designed for the mean-based watermarking algorithm from [10]. This steganalytic algorithm first estimates the number of bins through exhaustive search and then detects the presence of the secret message by a tailor-made normality test. The ability of a steganalyzer to generalize from the training set to a large testing set is a very demanding requirement. A steganalytic approach which is designed specifically for addressing the cover source mismatch scenario in 3D steganalysis by selecting the features which are robust to the variations of the cover source was proposed in [25]. Cryptanalysis aspects of 3D watermarking in a larger context have been discussed in the review paper from [18].

The aims of a steganalyzer are difficult to achieve because the stego-object and cover-object are supposed to be almost identical and undetectable visually at least. Different from the features and local descriptors [29] used for image and 3D object classification [19,41], recognition [13] and retrieval [17,35,51,52], which aim to recognise the content of the image or 3D object, the steganalytic features have to be oblivious to the content of the image or 3D object but sensitive to the subtle localized distortions. Nevertheless, there are links between the steganalytic features and the measurements used for image and 3D object quality assessment [23,43,53]. For example, the local roughness measure proposed in [23] can detect the geometric noise on the object's surface, which resembles the embedding distortions caused by the information hiding methods. Additionally, the image gradient magnitude is used for both image quality assessment in [53] and colour image steganalysis [1].

In this paper, we propose some new features to be used for 3D steganalysis. We use the Gaussian curvature and the curvature ratio in order to capture the changes of the surface’s curvatures. Furthermore, the vertex normal is also considered to find the changes in the orientation of polygon faces containing that vertex. We also propose to use the statistics of spherical coordinates, and the length of the edge defined in spherical coordinates, for 3D steganalysis. The spherical coordinate system is often utilized for hiding information in 3D objects, but it has not been previously used in 3D steganalysis. The statistics of sets of 3D features are then fed into machine learning algorithms, for example, the Fisher Linear Discriminant (FLD) ensemble [20], which was successfully used for image steganalysis as well. The discriminating ability for the 3D steganalyzer trained with the proposed feature set is then compared against the one proposed in [47].

The description of the 3D steganalysis framework formulated in this study is provided in Section 2. The 3D feature set, used by the steganalyzer is presented in detail in Section 3. The experimental results are provided in Section 4, while the conclusions of this study are outlined in Section 5.

2. 3D steganalysis framework

In this section, we provide a brief introduction of the 3D steganalysis framework. The steganalysis framework is treated as a machine learning problem, consisting of training and testing stages. The training of the steganalyzer has the following processing steps: preprocessing, feature extraction and learning, as illustrated in Fig. 1. The result of these processing steps consists of a parameter set discriminating between the 3D objects carrying hidden information and those that are not. The testing stage includes the same preprocessing and feature extraction steps as the training stage, while applying the parameters learnt during the training on the features extracted from various sets of test objects.

Firstly, during the preprocessing stage, Laplacian smoothing is applied on all the graphical objects. The idea of 3D object smoothing was borrowed from image steganalysis, where it was observed that the difference between the stego-image and its smoothed version is more significant than the difference between the cover-image and its corresponding smoothed version [15,21]. Similarly, it is expected that the difference between a mesh and its smoothed version is larger for a stego mesh than for a cover mesh. In most 3D watermarking algorithms, the changes produced to the stego-object, following the watermark embedding, can be associated to noise-like changes. Consequently, when smoothing a cover mesh, the resulting modifications will be smaller than those obtained when smoothing its corresponding stego mesh. We consider Laplacian smoothing for the object O, resulting in its smoothed version O′. Then the rotation and scaling are used to normalize the objects such that their size is constrained within a cube with sides of one, in order to eliminate the perturbation on the features, caused by the variation of the size in the objects from the training set.
Fig. 1. The 3D steganalysis framework based on learning from statistics of the local feature sets and classification by means of machine learning methods.

Features, characterizing the local geometry of 3D objects are extracted after the preprocessing stage. In Section 3 we propose to use a new set of 3D features for steganalysis. Then, $\Phi$ represents the absolute difference between the geometrical statistics extracted from the object $O$ and its smoothed version $O'$. In order to statistically model the difference, we consider the first four statistical moments, representing the mean, variance, skewness and kurtosis, of the logarithm of $\Phi$ as the feature vector $x$, which is then used as an input to a machine learning algorithm along with the class label $y$ for the object $O$.

In the supervised learning phase, a classifier is trained over the extracted feature vectors and the corresponding class labels of the objects. The classifier, which is also known as the steganalyzer, separates the feature space defining the stego-objects from that of the cover-objects. In this research study we propose to use the FLD ensemble [20] for training the steganalyzer.

3. Features for 3D steganalysis

3D watermarking and steganographic methods are specifically designed to embed information in a way that does not visibly alter the surface of the objects [6,27]. 3D steganalysis aims to find computationally such changes, separating the stego-objects from the cover-objects. Depending on the specific algorithm used, such changes could be randomly distributed on the surface of the 3D mesh [5] or they could be specifically located in certain regions of the object [2]. Artefacts produced in objects, following the information hiding embeddings, could be assimilated to low level protuberances on mesh surfaces and consequently could be identified by feature detection algorithms. In the following we outline some 3D local features which can be used for identifying whether objects have been watermarked or not. Such feature detectors range from very simple vertex displacement measurements to algorithms that take into account the local neighbourhoods and measure specific shape characteristics.

Let us assume that we have the shape of a 3D object, considered as a cover mesh $O = \{V,F,E\}$, containing the vertex set $V = \{v(i)|i = 1,2,\ldots,|V|\}$, where $|V|$ represents the number of vertices in the object $O$, its face set $F$, and its edge set $E$, respectively. We define the neighbourhood $N(v(i))$ of a vertex $v(i)$ as $\{v(j) \in N(v(i)) | e_{i,j} \in E\}$, where $e_{i,j}$ is the edge connecting vertices $v(i)$ and $v(j)$.

3.1. Preprocessing

As already mentioned in Section 2, the preprocessing of the 3D objects is an essential phase for extracting the steganalytic features. One iteration Laplacian smoothing is firstly applied to the 3D object $O$, which updates the vertex $v(i)$ into $v'(i)$ as follows, [38]:

$$v'(i) \leftarrow v(i) + \frac{\lambda}{\sum_{v(j) \in N(v(i))} w_{ij}} \sum_{v(j) \in N(v(i))} w_{ij} (v(j) - v(i)).$$

(1)

where $\lambda$ is a scale factor and $w_{ij}$ are the weights defined as:

$$w_{ij} = \begin{cases} 1 & \text{if } v(j) \in N(v(i)) \\ 0 & \text{otherwise} \end{cases}$$

(2)
The value of $\lambda$ is found empirically, such that the object is appropriately smoothed. The object is afterwards aligned according to its first and second principal axes, given by the Principal Component Analysis (PCA). Afterwards, the object is scaled to fit inside a cube of sides equal to 1.

3.2. The YANG40 features

The 40-dimensional feature vector YANG40 contains the most effective features from YANG208, used in [47], which correspond to the statistics of features evaluated from the vertices, edges and faces that make up the given meshes. For YANG40 we remove certain features, which provide lower performance, from YANG208 and abandon the strategy used in [47] which treats the vertices with valence less, equal, or greater than six separately for the sake of reducing the dimensionality.

Let us denote by $\Phi$, the feature set representing differences between the object $O$ and its smoothed version $O'$. The first six components of $\Phi$ represent the absolute distance, measured along each coordinate axis $x$, $y$, $z$ between the locations of vertices of the meshes $O$ and $O'$ after being normalized and aligned, in both the Cartesian and Laplacian coordinate systems [46]:

\begin{align}
\phi_1(i) &= |v_x(i) - v'_x(i)|, \\
\phi_2(i) &= |v_y(i) - v'_y(i)|, \\
\phi_3(i) &= |v_z(i) - v'_z(i)|, \\
\phi_4(i) &= |x_v(i) - x'_v(i)|, \\
\phi_5(i) &= |y_v(i) - y'_v(i)|, \\
\phi_6(i) &= |z_v(i) - z'_v(i)|,
\end{align}

where $v_x(i)$ and $v'_x(i)$ represent the $x$-coordinate of $v(i)$ in Cartesian and Laplacian coordinate systems, respectively, $i = 1, 2, \ldots, |V|$. The Laplacian coordinates of the object are the results of the Cartesian coordinates multiplied by the Kirchhoff matrix [4] of the object. Next, we evaluate the changes produced in the Euclidean distance between vertex locations and the centre of the object, representing the vertex norms. The absolute differences between the vertex norms of pairs of corresponding vertices in the meshes $O$ and $O'$ are calculated as:

\begin{align}
\phi_7(i) &= \|v(i)\| - \|v'_v(i)\| \\
\phi_8(i) &= \|v(i)\| - \|v'_v(i)\|,
\end{align}

where $\|v(i)\|$, $\|v'_v(i)\|$, represent the vector norms in Cartesian and Laplacian coordinates, respectively, for $i = 1, 2, \ldots, |V|$.

Another feature evaluates the local mesh surface variation by calculating the changes in the orientations of faces adjacent to the same edge. This is measured by the absolute differences between the dihedral angles of neighbouring faces, calculated in the plane perpendicular on the common edge $e(i) \in O$, $i = 1, 2, \ldots, |E|$, where $|E|$ represents the number of edges of the object $O$:

\begin{align}
\phi_9(i) &= |\theta_e(i) - \theta'_e(i)|,
\end{align}

where the calculation of the dihedral angle $\theta'_e(i)$ is illustrated in Fig. 2.

Changes in the local surface orientation are measured by calculating the angle between the surface normals $\hat{N}_{F(i)}$, of the faces from the object $F(i) \in O$, and their correspondents $\hat{N}'_{F(i)}$, from the smoothed object $F'(i) \in O'$:

\begin{align}
\phi_{10}(i) &= \arccos \frac{\hat{N}_{F(i)} \cdot \hat{N}'_{F(i)}}{\|\hat{N}_{F(i)}\| \cdot \|\hat{N}'_{F(i)}\|}
\end{align}

where $i = 1, 2, \ldots, |F|$. The 40-dimensional feature vector YANG40 represents the first four statistical moments, representing the mean, variance, skewness and kurtosis of the logarithm of the ten vectors $\{\phi_i | i = 1, 2, \ldots, 10\}$, described above.
3.3. The vertex normal and curvature features

In the following we propose to use some additional 3D features. These features model localized geometrical properties of the 3D shapes, and following extensive experimentation, they have shown to be efficient for steganalysis. The vertex normal is the weighted sum of the normals of all faces that contain the vertex [28]. A vertex normal is illustrated in Fig. 2 and is computed as:

\[
\vec{N}_{v(i)} = \sum \frac{A(F(j))}{\|e_{v(i),1}\|^2 \cdot \|e_{v(i),2}\|^2} \vec{N}_{F(j)}
\]

where \(F(j)\) represents the jth face that contains the vertex \(v(i)\), \(A(F(j))\) represents its area, \(e_{v(i),1}\) and \(e_{v(i),2}\) are the two edges containing \(v(i)\) in the face \(F(j)\). The change between two vertex normals is calculated as a dot product:

\[
\phi_{11}(i) = \arccos \frac{\vec{N}_{v(i)} \cdot \vec{N}_{v'(i)}}{\|\vec{N}_{v(i)}\| \cdot \|\vec{N}_{v'(i)}\|}
\]

where \(\vec{N}_{v'(i)}\) is the normal for a vertex from the smoothed object \(\{v'(i) \in \mathcal{O} | i = 1, 2, \ldots, |\mathcal{V}|\}\).

Next we consider the local shape curvatures, calculated according to the Gaussian curvature and the curvature ratio formula used in [33]. Most natural objects have shapes with many curvatures all over their surface. Steganographic algorithms would tend to embed changes that may influence the local curvatures. In differential geometry, the two principal curvatures of a surface are provided by the eigenvalues of the shape operator, calculated at the location of a vertex using the vertices from its first neighbourhood. Such curvatures measure how the local surface bends by different amounts in the orthogonal directions at that point. The Gaussian curvature is defined as:

\[
K_G = K_1 K_2,
\]

where \(K_1\) is the minimum principal curvature and \(K_2\) is the maximum principal curvature at a given point [34]. A special case is that of singularity in the shape operator, when we have a linear dependency in one direction or in both. In this case we have locally a planar region, which is characterized by a linear relationship among its coordinates and consequently by zero curvature. In our study we found that the curvature ratio proposed in [33], defined as

\[
K_i = \frac{\min(|K_1|, |K_2|)}{\max(|K_1|, |K_2|)},
\]

is effective to be used as a feature when training steganalyzers. The Gaussian curvature from Eq. (11) and the curvature ratio from Eq. (12) have been shown to be sensitive to very small mesh modifications and have been used to model 3D shape characteristics in various applications [39,40]. The two principal curvatures are evaluated at the location of each vertex in the object \(v(i) \in \mathcal{O}\) and for its corresponding vertex from the smoothed object \(v'(i) \in \mathcal{O'}\). Their absolute differences represent the features \(\phi_{12}\) and \(\phi_{13}\) used in the proposed set of features:

\[
\phi_{12}(i) = |K_G(v(i)) - K_G(v'(i))|,
\]

\[
\phi_{13}(i) = |K_i(v(i)) - K_i(v'(i))|,
\]

for \(i = 1, 2, \ldots, |\mathcal{V}|\).

3.4. The spherical coordinates features

Spherical coordinates provide a straight forward representation for most graphical objects in characterizing the distance from the centre and the location of each vertex on a sphere. Certain 3D watermarking methods, such as those from [10,49], specifically embed changes into spherical coordinates. We convert the 3D objects from the Cartesian coordinate system to the spherical coordinate system, considering the centre of the object as its reference.

The spherical coordinate system specifies a point in the 3D space by a radius and two angles and the link to the Cartesian coordinate system is given by:

\[
\begin{align*}
    v_x &= R \cos(\phi) \cos(\theta) \\
    v_y &= R \cos(\phi) \sin(\theta) \\
    v_z &= R \sin(\phi)
\end{align*}
\]

where \(v = (v_x, v_y, v_z)\) represents the Cartesian coordinates of the vertex, and \((R, \theta, \phi)\) its spherical coordinates, representing \(R\), the Euclidean norm from a fixed origin, \(\theta\), the azimuth angle, while \(\phi\) is the elevation angle, as illustrated in Fig. 3. We
compute the absolute differences of the spherical coordinates of all vertices, \( \{ (R(i), \theta(i), \varphi(i)) \} \) between the original object \( O \) and the smoothed object \( O' \) in the spherical coordinate system:

\[
\begin{align*}
\phi_{14}(i) &= |\theta(i) - \theta'(i)|, \\
\phi_{15}(i) &= |\varphi(i) - \varphi'(i)|, \\
\phi_{16}(i) &= |R(i) - R'(i)|
\end{align*}
\]

where \( i = 1, 2, \ldots, |V| \). The centre of the spherical coordinate system is \( O \), representing the centre of the 3D object calculated by averaging all the vertices in the object, as shown in Fig. 3.

We also use the statistics of the edges, defined in the spherical coordinate system. In this case, edges are defined by the differences in the spherical coordinates of the two vertices that define the edge ends:

\[
\begin{align*}
K_\theta(e_{(i,j)}) &= |\theta(i) - \theta(j)|, \\
K_\varphi(e_{(i,j)}) &= |\varphi(i) - \varphi(j)|, \\
K_R(e_{(i,j)}) &= |R(i) - R(j)|
\end{align*}
\]

where \( e_{(i,j)} \) is the edge connecting vertices \( v(i) \) and \( v(j) \), and \( e_{(i,j)} \in E \). The corresponding features extracted from both the original object and its smoothed version are

\[
\begin{align*}
\phi_{17}(i) &= |K_\theta(i) - K_\theta'(i)|, \\
\phi_{18}(i) &= |K_\varphi(i) - K_\varphi'(i)|, \\
\phi_{19}(i) &= |K_R(i) - K_R'(i)|
\end{align*}
\]

where, for example, \( K_\theta(i) \) is obtained from the \( i \)th edge of the original object, while \( K_\theta'(i) \) from its corresponding edge in the smoothed object, for \( i = 1, 2, \ldots, |E| \), \( |E| \) is the total number of edges in object \( O \).

It was observed that most histograms of the features, such as those mentioned above, are highly uneven. In many cases such histograms are almost exponential. In order to introduce evenness in the distribution of the features, we apply the logarithm for all features. Then, we consider the first four statistical moments, representing the mean, variance, skewness and kurtosis, of the logarithm of all vertex normals, Gaussian curvatures, curvature ratios, and the spherical coordinate features calculated as indicated above, as it was done in the case of the feature set YANG40, presented in Section 3.2. In this way we define a vector \( \mathbf{x} \) of \( 19 \times 4 = 76 \) dimensions, which we call LFS76. The first four moments capture almost entirely the statistical characteristics of the distribution of the features, representing their centre and the deviation from the centre, as indicated by the mean and variance, respectively. The degree of symmetry in the logarithm of feature values is indicated by the skewness, while the level of peakedness and the presence of specific values in the statistical distribution is indicated by the kurtosis, representing the fourth statistical moment.

A subset of the proposed feature set, LFSS2, was used in [24]. That feature set did not include the 24-dimensional feature vector extracted in the spherical coordinate system of 3D objects. A higher dimensional feature set, used in [47], is represented by the 208-dimensional vector defined as YANG208. This feature set considers separately the statistics of the first eight features described above, distinctly on vertex sets with valences less, equal, or greater than six. Moreover, YANG208 feature set considers the histogram differences of the ten features defined in Section 3.2, as well. The features described in this section are mainly local and are centred on either the vertices or the edges or the faces forming the 3D meshes of the objects. During experimental studies we have tested other features, either local or global, representing larger regions of 3D objects, but the results have not been satisfactory.

The feature set described in this section is used as an input to a machine learning classifier. The machine learning classifier has two stages. During the first stage, it learns the feature spaces characterizing the stego-objects and the cover-objects respectively, and estimates the boundary between the two classes. Then, during a test stage, the parameters, learnt during the initial stage, are used for identifying new stego-objects, which have not been used during the training stage. A machine learning classifier is expected to provide a good generalization. For this study we are using the Fisher Linear
Discriminant (FLD) ensemble, and Quadratic Discriminant Analyst (QDA), as classifiers for 3D steganalysis. The quadratic discriminant fits multivariate normal densities with covariance estimates [22] and was used for 3D steganalysis in [47]. The FLD ensemble classifier was successfully used in image steganalysis [12,20,50].

4. Experimental Results

In the following we provide the experimental results for the proposed 3D steganalytic methodology, when detecting the stego-objects obtained by six different information hiding methods. For the experiments we use the Princeton Mesh Segmentation project [9] database, which consists of 354 3D objects represented as meshes. The shapes of ten objects from this database are shown in Fig. 4.

4.1. The 3D information hiding methods and their parameter settings

During the tests we consider detecting the 3D stego-objects obtained by hiding information using six different embedding algorithms: the Steganalysis-Resistant Watermarking (SRW) method proposed in [49]; the two blind robust watermarking algorithms based on modifying the Mean or the Variance of the distribution of the vertices’ Radial distances in the Spherical coordinate system, denoted as MRS and VRS, from [10]; the Wavelet-based High Capacity (WHC) watermarking method and Wavelet-based FRagile (WFR) watermarking method proposed in [42]; the Multi-Layer Steganography (MLS) provided in [8].

During the generation of the stego-objects using SRW method from [49], we consider multiple values for the parameter $k$ which determines the number of bins for the histogram of the radial distances for all vertices. According to Yang et al. [49], the upper bound of the embedding capacity is $(K-2)/2$. In our experiments we set the parameter $K = \{32, 64, 96, 128\}$ and thus obtain multiple sets of stego-objects. Another parameter in the watermarking method from [49] is $n_{th}$ which controls the robustness of the embedding method. In order to keep the distortion of the embedding to a relatively low level, we set the parameter $n_{th}$ as 20. If the smallest number of the elements in the bins from the objects is less than 20, we would choose $n_{th}$ equal to the smallest nonzero number of the elements in the bins. Examples of stego-objects obtained using SRW method are shown in Figs. 7(a) and 8(a), where $K = 128$.

For MRS and VRS watermarking methods from [10], we consider various values for the watermark strength, such as $\alpha = \{0.02, 0.04, 0.06, 0.08, 0.1\}$, while fixing the incremental step size to $\Delta k = 0.001$ and the message payload as 64 bits. Larger values of strength can increase the robustness of the watermark, but also enlarge the extent of the embedding modifications. An example of a stego-object obtained using MRS method is shown in Fig. 7(f), where the watermark strength factor is set as $\alpha = 0.04$.

In both WHC and WFR watermarking methods, proposed in [42], the information is embedded in the wavelet coefficient vectors obtained just after one wavelet decomposition of the original mesh, but the modifications are made in different ways for each of these algorithms. During the watermark embedding by WHC, the wavelet coefficient vectors’ norms are firstly divided by the parameter $p$. Then the resulting residues, representing the differences from the rounding error, are changed accordingly in order to generate a particular permutation which carries the watermark. The parameter $p$ is obtained by dividing the average edge length for the entire object by the control parameter $\epsilon_{hc}$. When using WFR to embed information, the angle between the wavelet coefficient vector and its associated edge is changed, where $\Delta_n$ is the quantization step used to establish the codebook. To investigate the influence of parameter $\Delta_n$ on the steganalysis results, we set it to the values from the set $\{\pi/6, \pi/4, \pi/3, \pi/2\}$. The other parameters involved in WHC and WFR are all exactly set to the values suggested in [42].

When using MLS method from [8], we increase the number of layers from 2 to 10, with a step of 2, and we consider the number of intervals as 10,000. Increasing the number of embedding layers in this steganographic method corresponds to

![Fig. 4. 3D objects used in the steganalytic tests.](image-url)
increasing the payload capacity. During the embedding, all the vertices in the mesh are used as payload carriers, except for three vertices which are used as references for the extraction process. A stego-object obtained using MLS method is shown in Fig. 7(k), where the number of layers is 10.

4.2. Feature extraction

The steganalytic features are extracted from the cover-objects and the corresponding stego-objects obtained after embedding the information by using the six steganographic algorithms mentioned above. During the preprocessing, we first apply one iteration of Laplacian smoothing on both cover-objects and stego-objects, by setting the scale factor $\lambda = 0.2$. We consider the proposed feature set LFS76, discussed in Section 3 and compare their results against YANG208, proposed in [47], its simplified version, called YANG40, and the feature set LFS52, which was proposed in our previous work [24]. We also consider the feature sets combining LFS52 and the features defining the Vertices’ Spherical coordinates, VS12, representing the mean, variance, skewness and kurtosis of $\phi_{14}$, $\phi_{15}$ and $\phi_{16}$ from Eq. (16), the combination of LFS52 and the features defining the Edge length in the Spherical coordinate system, ES12, representing the mean, variance, skewness and kurtosis of $\phi_{17}$, $\phi_{18}$ and $\phi_{19}$ from Eq. (18).

Fig. 5(a) and (b) show the histograms of the dihedral angle feature $\phi_9$, calculated according to Eq. (7), for the cover-object and stego-object, respectively, for the object “Head statue”, shown in Fig. 7(f). The histograms of the logarithm of $\phi_9$ are shown in Fig. 5(c) and (d), for the cover-object and stego-object, respectively. Fig. 6(a) and (b) show the histograms of the vertex normal feature $\phi_{11}$ calculated according to Eq. (10), while Fig. 6(c) and (d) show the corresponding histogram of logarithms for the cover-object “Horse” shown in Fig. 4, and its corresponding stego-object embedded by MRS method from [10]. From these figures, we can observe that following the application of the logarithm, the distributions of feature components $\phi_9$ and $\phi_{11}$ become similar to normal distributions, where it is easier to model the differences between the geometrical statistics of the cover-object and stego-object using the first four statistical moments of mean, variance, skewness and kurtosis.

Fig. 7(a) (f) and (k) show the stego-objects embedded by the information hiding methods, SRW [49], MRS [10] and MLS [8], respectively. Fig. 8(a) represents the stego-object obtained by using SRW method. From the second to the fifth columns of Figs. 7 and 8 illustrate the absolute differences of the features between the cover-object and its corresponding stego-object, namely, vertex normals $\phi_{11}$, the curvature ratios $\phi_{13}$, the azimuth angles $\phi_{14}$ and the radial distances $\phi_{16}$, depicted on the stego-objects. From these figures it can be observed that each feature identifies specific differences between the cover-object and stego-object, which usually does not overlap with those identified by the others.
Fig. 7. Stego-objects and the visualization of differences in the detection of features used for steganalysis. (a), (f) and (k) are the stego-objects obtained after using the information hiding algorithms, SRW [49], MRS [10] and MLS [8], respectively; (b), (g) and (l) show the absolute differences of vertex normals $\phi_{11}$ between those stego-objects and their corresponding cover-object, respectively; (c), (h) and (m) for the curvature ratios $\phi_{13}$; (d), (i) and (n) for the azimuth angle $\phi_{14}$; (e), (j) and (o) for the radial distance $\phi_{16}$.

Fig. 8. Stego-object and the visualization of differences in the detection of features used for steganalysis. (a) The stego-object obtained after using SRW algorithms described in [49]; (b) The absolute differences of vertex normals $\phi_{11}$ between the stego-objects and their corresponding cover-object; The absolute differences in (c) for the curvature ratios $\phi_{13}$; (d) for the azimuth angle $\phi_{14}$; (e) for the radial distance $\phi_{16}$. 
4.3. Training the steganalyzers

The steganalyzers are trained as binary classifiers implemented using two methods: the Quadratic Discriminant Analysis (QDA) and the Fisher Linear Discriminant (FLD) ensemble. The FLD ensemble consists of a set of base learners trained uniformly on a randomly selected feature subset of the whole training data. The dimensionality of the random subspace and the number of base learners are found by minimizing the out-of-bag (OOB) error, representing an estimate of the testing error calculated on bootstrap samples of the training set, [14]. Compared to the SVM classifier, the FLD ensemble can provide a comparable high accuracy, but with a relatively low computational cost. On the other hand, in the case of FLD ensemble, it is much easier to find the optimal tuning parameters. For more technical details of the FLD ensemble, we refer to the literature [11,20].

For each steganalyzer, we split the 354 pairs of cover-object and stego-object into 260 pairs, used for training, and 94 pairs for testing. We consider 30 different splits of the given 3D object database, into the training and testing data sets. Two different assessment measures are used: the first one is the median value of the detection errors which are the sum of false negatives (missed detections) and false positives (false alarms) from all 30 trials, while the other one is the median value of the area under the Receiver Operating Characteristic (ROC) curves of the detection results, evaluated over the 30 splits of the data into training and testing sets.

4.4. Statistical steganalysis study

In the following we test the FLD ensemble and QDA steganalyzers on 3D objects with information embedded by six different information hiding algorithms. Fig. 9 shows the detection errors for the six information hiding methods, SRW [49], MRS [10], VRS [10], WHC [42], WFR [42] and MLS [8], using the FLD ensemble classifier, trained with the six combinations of feature sets, formed as mentioned above. It can be seen from Fig. 9 that the LFS76 shows best performance among the six combinations for most of the cases. The improvement of the efficiency of LFS76, compared to YANG208, is quite evident for all the six embedding algorithms. The advantage of LFS76 over LFS52 is more obvious when detecting the watermarks embedded by SRW, MRS and VRS methods, than when detecting those embedded by WHC, WFR and MLS methods. This is because WHC, WFR and MLS methods do not produce changes in the spherical coordinate system, and consequently the feature sets VS12 and ES12 are less useful. When considering the feature sets VS12 and ES12, it appears that the combination of LFS52 and ES12 achieves better performance than that of LFS52 and VS12, which means that the ES12 features are more efficient than VS12.
We can observe from Fig. 9(a) that as the value of $K$ increases, the detection error for SRW [49] tends to increase as well. This happens because a larger $K$ will lead to fewer elements in each bin, and consequently less vertices have to be changed for embedding a single bit. From Fig. 9(b) and (c) we can observe that as the watermarking strength of MRS and VRS [10] increases, all steganalyzers provide better detection accuracy. This is due to the fact that more significant changes are produced in the 3D object surface, which is caused by watermarks that have stronger embedding parameters. For WHC [42] method, the detection error shown in Fig. 9(d) increases slightly when $\epsilon_{hc}$ ranges from 50 to 100, but remains stable afterwards. In Fig. 9(e), the detection error for WFR [42] method does not have obvious changes when the parameter $\Delta_y$ varies, which indicates that the parameter $\Delta_y$ does not influence significantly the embedding distortion of the object. It can be observed from Fig. 9(f) that the detection error for MLS [8] does not decline when the embedding capacity increases. The reason for this is that, according to the multi-layer embedding framework applied in [8], the distortions produced to the objects are well controlled during the embedding.

Fig. 10 shows the detection errors for the six information hiding methods, using the QDA classifier, trained with various feature sets. The trends of the detection errors for the six embedding algorithms depicted in Fig. 10 are similar to those shown in Fig. 9, but the performance of the QDA classifier is not as good as the FLD ensemble classifier in general.

In the following, we discuss the detectability of the watermarks embedded by the six information hiding algorithms by the steganalyzers trained with the LFS76 feature set. When $K = 128$, the payload of SRW is close to 64 bits, which is the payload of MRS and VRS as well in our experiments. From Fig. 9(a)–(c) we can see that SRW has a lower detectability than those of MRS and VRS, which was reported in [49] as well. Since the procedure and embedding domains of MRS and VRS are very similar, it is not surprising that their watermark detectability is close to each other. WHC and WFR also have quite similar watermark detectability ratios, but which are lower than those of the other four information hiding algorithms considered during the experiments. This is because LFS76 does not include the features from the wavelet domain, in which modifications of 3D objects are made by WHC and WFR. The detectability of changes embedded by MLS algorithm is moderate among the six information hiding methods, according to the Fig. 9, but the payload of MLS can be much higher than the other methods, equal to approximately 10 times the number of vertices in the object. If the payload of MLS information hiding algorithm is decreased to 64 bits per object, the stego-objects embedded by MLS would be much harder to detect.
4.5. Analysing the efficiency of features for steganalysis

In order to investigate the contribution of different categories of features from the set LFS76 to the steganalysis, we use the relevance between the feature vectors and the class label in order to assess each feature’s efficiency. The measurement of the relevance is addressed by using the Pearson correlation coefficient,

$$\rho(x_i, y) = \frac{cov(x_i, y)}{\sigma_x \sigma_y}$$

(19)

where $x_i$ is the $i$th feature of a given feature set, $x = \{x_i|i = 1, 2, \ldots, N\}$, and $N$ is the dimensionality of the input feature, $y$ is the class label indicating whether the class corresponds to a cover-object or a stego-object, $cov$ represents the covariance and $\sigma_x$ is the standard deviation of $x_i$. The Pearson correlation coefficient is well known as a measure of the linear dependence between two variables [16]. Then we set $|\rho(x_i, y)|$ for assessing the relevance, where $|\rho(x_i, y)| = 1$ indicates a highly linear relationship between the feature and the class label, corresponding to a better discriminant ability of that feature.

The analysis is conducted on the features extracted from the 354 cover-objects used above and for the six sets of corresponding stego-objects which are produced by the six watermarking and steganographic algorithms, SRW, MRS, VRS, WHC, WFR and MLS, respectively. We set the parameter $K$ in SRW algorithm from [49] as 128. For the two watermarking methods, MRS and VRS, from [10], in order to find a balance between the watermarking strength and its detectability, we set the watermarking strength as 0.04 and embed a payload of 64 bits. For WHC and WFR methods, we set the parameters as $\epsilon_w = 100$ and $\Lambda_0 = \pi / 6$, which are the same as the settings in [42]. In the case when using MLS method from [8], we consider ten layers of embedding.

We split the features from the set LFS76 into 10 categories according to their representations of the local shape geometry: 1, the vertex coordinates in the Cartesian coordinate system ($\phi_1, \phi_2, \phi_7$); 2, the vertex norm in the Cartesian coordinate system ($\phi_4$); 3, the vertex coordinates in the Laplacian coordinate system ($\phi_5, \phi_6, \phi_8$); 4, the vertex norm in Laplacian coordinate system ($\phi_3$); 5, the face normal ($\phi_{10}$); 6, the dihedral angle ($\phi_9$); 7, the vertex normal ($\phi_{11}$); 8, the curvature ($\phi_{12}$ and $\phi_{13}$); 9, the vertex coordinates in the spherical coordinate system ($\phi_{14}, \phi_{15}$ and $\phi_{16}$); 10, the edge length in the spherical coordinate system ($\phi_{17}, \phi_{18}$ and $\phi_{19}$).

The relevance for all features from LFS76, is calculated according to Eq. (19), and the averaged relevances of the features in each category are shown in Fig. 11. From Fig. 11 we can observe that the new proposed features, represented by labels 7, 8 and 10, have relatively high relevance to the class label. More specifically, in Fig. 11 (a), the features characterizing the local curvature (label 8) achieve the highest relevance. The relevance of the proposed ES12 feature, represented by label 10, is higher than that of the proposed VS12 represented by label 9. This implies that the efficiency of ES12 is higher than that of VS12, which is also reflected in the results shown in the Figs. 9 and 10. Comparing the formulation of VS12 and ES12, it is noted that two adjacent vertices are taken into account when extracting the ES12 features. However, the vertices in the object are considered individually in the case of VS12. So ES12 is better able of capturing the distortion in the local region caused by the embedding modifications than VS12. However, the VS12 probably detects hidden information in the 3D object, which ES12 cannot identify, resulting in better performance of LFS76 than the combination of LFS52 and ES12 as shown in Figs. 9 and 10. Meanwhile, in Fig. 11, the vertex norm feature (label 7) and the face normal feature (label 5) have a similar level of relevance, which is because the vertex norm is dependent on face normal in Eq. (9).

It is interesting that the relevance of the dihedral angle feature (label 6) shows a high relevance to the class label in the cases of MRS, VRS and MLS, but shows much lower relevance when the stego-objects are generated by SRW, WHC and WFR methods. This happens because almost all vertices from a mesh are only slightly changed by MRS, VRS and MLS methods, while such changes are scattered among the vertices in the case of SRW method from [49], as it can be observed from Fig. 7(b) (g) and (l). Similarly, when considering WHC and WFR, the modifications are made after only one wavelet decomposition, so only half of the vertices are likely to be modified, preserving the dihedral angles to some extent. It is noticed that the features representing the vertex coordinates and the norm in the Laplacian coordinate system (labels 3 and 4) have much higher relevance than those representing the vertex coordinates and the norm in the Cartesian coordinate system (labels 1 and 2). This is because, according to [46], the Laplacian coordinates of a vertex are calculated from the position of the vertex and its adjacent vertices, which capture the geometrical information of a larger region than the Cartesian coordinates for each vertex.

In the following, we increase gradually the feature set used for training the steganalyzer, from YANG40 to LFS52, by adding either VS12 or ES12 to LFS52, and eventually to the LFS76 feature set, and then compare with YANG208 feature set. YANG40 includes the features represented by labels 1–6 in Fig. 11. Features represented by labels 1–8 form LFS52, while labels 1–10 correspond to LFS76. VS12 and ES12 are represented by labels 9 and 10, respectively. We employed the FLD ensemble and QDA as the machine learning algorithms to train the steganalyzers when the information was hidden into 3D objects by the six information hiding methods mentioned above. The performance of the feature sets are evaluated by the area under the ROC curves of the corresponding steganalyzers. A larger area under the ROC curve means that the classifier has a better detection accuracy.

Tables 1 and 2 provide the median values of the area under the ROC curves for the steganalytic methods when using six combinations of feature sets for 30 independent splits of the training/testing set. It can be seen from Table 1 that the areas under the ROC curves of the steganalyzers increase with the addition of new features, such as the vertex normal and the curvature features, to the YANG40 feature set. After adding VS12 and ES12 to LFS52 feature set, the LFS76 feature set
Fig. 11. The relevance between the features and the class label, for cover-objects (o) or stego-objects (1), where the stego objects are generated by the six information hiding methods, SRW, MRS, VRS, WHC, WFR and MLS, respectively. The meaning of the category labels are: 1, the vertex coordinates in Cartesian coordinate system; 2, the vertex norm in Cartesian coordinate system; 3, the vertex coordinates in Laplacian coordinate system; 4, the vertex norm in Laplacian coordinate system; 5, the face normal; 6, the dihedral angle; 7, the vertex normal; 8, the curvature; 9, the vertex coordinates in spherical coordinates; 10, the edge length in spherical coordinate system.

Table 1
Median values of the area under the ROC curves for the steganalysis results of the six information hiding algorithms when using the FLD ensemble classifier. The best results are shown in bold.

<table>
<thead>
<tr>
<th>Feature sets</th>
<th>Information hiding methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>YANG208</td>
<td>0.8781</td>
</tr>
<tr>
<td>YANG40</td>
<td>0.7782</td>
</tr>
<tr>
<td>LFS52</td>
<td>0.8621</td>
</tr>
<tr>
<td>LFS52+VS12</td>
<td>0.8617</td>
</tr>
<tr>
<td>LFS52+ES12</td>
<td>0.9064</td>
</tr>
<tr>
<td>LFS76</td>
<td>0.9032</td>
</tr>
</tbody>
</table>

Table 2
Median values of the area under the ROC curves for the steganalysis results of the six information hiding algorithms when using the QDA classifier. The best results are shown in bold.

<table>
<thead>
<tr>
<th>Feature sets</th>
<th>Information hiding methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>YANG208</td>
<td>0.8035</td>
</tr>
<tr>
<td>YANG40</td>
<td>0.7485</td>
</tr>
<tr>
<td>LFS52</td>
<td>0.8228</td>
</tr>
<tr>
<td>LFS52+VS12</td>
<td>0.8351</td>
</tr>
<tr>
<td>LFS52+ES12</td>
<td>0.8702</td>
</tr>
<tr>
<td>LFS76</td>
<td>0.8871</td>
</tr>
</tbody>
</table>
achieves the best performance in most cases. For SRW and MLS, the combination of LFS52 and ES12 and that of LFS52 and VS12 give the best performance, respectively, which also justifies the importance of VS12 and ES12 features. But in general, the combination of LFS52 and ES12 has a better performance than that of LFS52 and VS12, indicating the higher efficiency of ES12 when compared to that of VS12, consistently reflected in the results provided in Fig. 11. The upward trend in the area under the ROC curves along with the addition of new features can be identified in the results provided in Table 2 as well. According to these results, the FLD ensemble classifier provides better results than the QDA, used in [47].

5. Conclusion

The task of a 3D steganalyzer is very challenging because it has to find very small differences between stego-objects and cover-objects. In this research study, we propose to use the statistics of some new shape features as inputs for 3D steganalyzers. We analyze various combinations of local features used for 3D steganalysis by evaluating their relevance to the class label and by testing their performance in the experiments. The first four statistical moments in various 3D feature sets are used for training steganalyzers by two machine learning methods, namely, the Quadratic Discriminant Analysis (QDA) and the Fisher Linear Discriminant (FLD) ensemble. After training, these steganalyzers are used for differentiating the stego-objects from the cover-objects. The experimental results show that the proposed 3D feature set provides the best results for the steganalysis of six 3D information hiding algorithms. Since the detection errors for the wavelet-based embedding algorithms, such as WHC and WFR, are higher than those for the other embedding algorithms, in future studies we will aim to improve the detection performance for WHC and WFR by using characteristic 3D wavelet features.

Acknowledgements

The authors would like to thank the anonymous editor and reviewers for their constructive comments. The first author acknowledges the scholarship received from Zhengzhou Institute of Information Science and Technology.

References


