IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 1, 2019, accepted December 13, 2019, date of publication December 17, 2019,
date of current version December 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2960455

Breaking Permutation-Based Mesh Steganography and Security Improvement

YIMIN WANG[1], LINGSHENG KONG[1], ZHENXING QIAN[2], GUORUI FENG[3], XINPENG ZHANG[2], AND JIANMIN ZHENG[4]

[1]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[2]School of Computer Science, Shanghai Institute of Intelligent Electronics and Systems, Fudan University, Shanghai 201203, China
[3]School of Communication and Information Engineering, Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China
[4]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798

Corresponding authors: Yimin Wang (y_wang@shu.edu.cn) and Zhenxing Qian (zxqian@fudan.edu.cn)

The work was supported by the National Natural Science Foundation of China under Grant 61402279, Grant U1736213, and Grant 61572308, and by the Singapore MOE Tier-2 Grants under 2016-T2-2-065 and 2017-T2-1-076.

ABSTRACT Permutation-based steganography in polygonal meshes can provide considerably large embedding capacities for hiding secret messages. However, corresponding steganalysis techniques against such methods have never been studied. This paper identifies the essential differences between naturally generated meshes and meshes produced by permutation-based steganography methods. It is found that the two types of meshes differ significantly in the distribution of topological distances between consecutive mesh elements. Therefore, by measuring the orderliness of the vertex list and the face list of meshes, we develop solutions for several mesh steganalysis problems. These solutions are effective, leading to high detection accuracy; and they are also universal, requiring no knowledge such as which steganography method is used and what data embedding rate is adopted for the detection mechanism to work. Moreover, this paper also presents a security-improved permutation-based mesh steganography approach, by taking advantage of the connectivity information of polygonal meshes and establishing a good trade-off between embedding capacity and undetectability. Without bringing global changes, our approach embeds secret messages into local neighborhoods on meshes. As a result, meshes generated by the proposed steganography approach tend to have natural structures that are unlikely to draw suspicions to steganalyzers.

INDEX TERMS Permutation-based steganography, steganalysis, polygonal meshes.

I. INTRODUCTION

Steganography and steganalysis are a pair of counterpart problems that have both received extensive attentions [1]. Steganography is the process of hiding secret information into host media in an unnoticeable way, while steganalysis is the process of detecting the presence of hidden information from given data by analysing its features and patterns. Steganography is a powerful tool for applications such as covert communication, and steganalysis is useful in areas such as internet security surveillance.

While early approaches of steganography and steganalysis mainly deal with images, videos, or audios [2], the use of 3D geometry as host media has been gradually brought to the attention in the past decade. As the latest generation of digital media, 3D geometry is now being frequently produced, used, and distributed due to the advances in computer graphics. Also, its flexible data structure potentially provides abundant spaces for hosting secret information.

Polygonal meshes, especially triangular meshes, are the most commonly used types of 3D geometry. In general, mesh steganography approaches fall into two major categories, the geometry-based approaches [3]–[5] and the permutation-based approaches [6]–[9]. The geometry-based approaches embed secret data into meshes by slightly perturbing the position of the vertices. Although such approaches try to keep geometrical distortions as low as possible, changes in the shape are inevitable after information hiding. On the

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

other hand, permutation-based approaches hide information into meshes by taking advantage of the redundancy in the mesh representation. Such approaches only alter the order of vertices and/or faces in the data storage, and do not cause any changes in the geometry.

Although considerable works have been done on mesh steganography, the problem of mesh steganalysis is relatively less investigated. As far as we realize, the recent approach reported in [10] is the only significant work on this topic. However, their steganalysis method is effective just against the geometry-based approaches, and is unfortunately not applicable for the permutation-based approaches. Therefore, one of the goals of this paper is to develop a steganalysis approach against permutation-based information hiding schemes of polygonal meshes. We define several distance measures that can be used to differentiate the naturally generated meshes from the ones that have been deliberately tampered. The proposed steganalysis method is capable of accurately identifying meshes that contain secret information, and is virtually effectively against all the current permutation-based approaches. Moreover, this paper presents a steganography method for safely embedding secret information into a host mesh. Instead of generating the new mesh arbitrarily with regard only to the data to be embedded, our approach carefully generate the new mesh in a similar way that a natural mesh is usually arranged. As a result, the mesh steganography approach is distortion-free, and also hardly detectable.

The rest of the paper is organized as follows. In section II, the state-of-the-art techniques related to mesh steganography and steganalysis are reviewed. In section III, the fundamental features of natural meshes and tampered meshes are studied. Based on that, a steganalysis approach is proposed in section IV. Next, a steganography approach is given in section V. Finally, the paper is concluded in section VI.

II. PRIOR WORK

This section gives an overview of the 3D steganography and steganalysis techniques, and provides necessary backgrounds for the methods to be described in this paper.

A. MESH STEGANOGRAPHY

Embedding capacity and geometrical distortion are the two most concerned indicators for the performance of mesh steganography approaches. Another well-established technique related to mesh steganography is mesh watermarking [11], [12], whose main purpose is to protect copyright of meshes by adding robust and unnoticeable signals into them. However, directly applying mesh watermarking techniques to mesh steganography usually results in quite low embedding capacity.

Therefore, several specialized approaches for mesh steganography have been developed. As mentioned above, these approaches can in general be classified into geometry-based and permutation-based.

1) GEOMETRY-BASED STEGANOGRAPHY

Essentially based on the classic idea of quantization index modulation [13], [14] or least significant bits (LSB) modification [15], the geometry-based approaches slightly perturb the positions of the vertices according to the data bits to be embedded. The problem was first investigated by Cayre and Macq [16]. Initially, the authors decide an ordered triangle list for the mesh by adopting the methods in [17], [18]. Then, a blind information hiding scheme is carried out by treating each triangle in the list as a two-state (i.e., 0 or 1) geometrical object and modifying its vertex position based on the data bit to be embedded. The embedding capacity for this approach is thus 1 bit per vertex. Later, Wang and Cheng [3] presented an improved method by introducing three independent vertex operations, sliding, extending, and rotating. The approach has definite upper bounds for geometrical distortions, and reaches an embedding capacity of 3 bits per vertex, one for each operation. Such steganography techniques were extended in [4] to point-sampled geometry. In the multi-layered data hiding scheme introduced by [5], a number of interleaved two-state intervals are constructed on the first principle axis. Each interval represents either the 0 or 1 state, and consists of change and un-change regions. Depending on the bit value to be embedded, the vertices either keep their original position or move to an appropriate change region. Moreover, by shifting the intervals back and forth, the data embedding process can be carried out repeatedly. Given a tolerance for face normal degeneration, [19] first calculates for each vertex a quantization level, and then overwrites the unused bits of the vertex coordinates with message bits.

No matter how changes are visually unobservable, the above geometry-based approaches always suffer from distortions. Increasing embedding capacity on one hand leads to more distortion on the other. More severely, these approaches tend to leave trails in statistical sense which would expose them to certain steganalytic systems.

2) PERMUTATION-BASED STEGANOGRAPHY

Many types of media data contains sets that have reorderable elements. Changing the order of these elements in the set does not affect the content of the data. Such a fact provides room for permutation-based steganography. For example, permutation-based steganography methods have been developed for GIF images by reordering colors in the palettes [20], [21].

For mesh models, permutation-based steganography can be carried out as follows. Denote a mesh as $M(\mathbf{P}, \mathbf{F})$, where $\mathbf{P} = \{P_1, P_2, \cdots, P_n\}$ is the set of vertices and $\mathbf{F} = \{F_1, F_2, \cdots, F_m\}$ is the set of faces. The elements in both $\mathbf{P}$ and $\mathbf{F}$ can be reordered without changing the mesh's geometry. Let T represent the secret message to be embedded. T consists of consecutive 0 and 1 and can be considered as a long integer. The basic idea is then to find an associated permutation π on $\mathbf{P}$ with regard to the value of T, in a recursive way. For each iteration i, the i-th element of the permutation is decided to be the (T/b_i)-th remaining element of a reference ordering of $\mathbf{P}$, where $b_i = (i-1)!$ is the factorial basis; also, T is updated by $T\%b_i$. By such a process, the information of T is conveyed by the permutation π. In order to facilitate the accurate recovery of T from π later, usually certain techniques on canonical traversal of meshes are used to decide a unique reference ordering of $\mathbf{P}$ [22]. By such an approach, the theoretical embedding capacity would be $\log_2(n!) = O(n\log(n))$ bits. Similarly, a permutation τ on $\mathbf{F}$ can also be calculated to embed more bits of T into the mesh. Because only the orders of the vertices and faces are changed during the process, the approach does not introduce any geometrical distortion. However, this rather straightforward approach can be extremely time-consuming, especially when a large-scale mesh is used as the cover media.

Therefore, a number of variations of the approach are proposed to find a reasonable tradeoff between the embedding capacity and the time complexity. In [6], the sets of vertices and faces are labelled with either state 0 or 1, and are reordered according to the bits in the secret message. Also, an extra scheme for shifting the order of the vertices within a face was proposed. Although these techniques give a total of 6 bits per vertex of capacity, [6] did not fully leverage the power of permutation. A method for true permutation-based steganography was investigated in [7]. Instead of doing a division between two long integers, the approach roughly uses the next $\lfloor\log_2(n-i+1)\rfloor$ bits in the secret message to pick the i-th element in the permutation π. Due to the simplification in computation, the approach is highly efficient, with only a loss of one bit per vertex for capacity than the optimal case. To further approach the optimal embedding capacity, improved works were presented in [8], [9], [23], where techniques such as building a special type of binary tree were adopted to find the next element in π.

In summary, all the above permutation-based approaches achieve considerable embedding capacities without introducing any geometrical distortion. Therefore, these methods can evade any steganalysis techniques designed against geometry-based approaches. However, the permutation-based approaches usually result in a complete rearrangement in both the vertex list and the face list, which still more or less leaves traces of deliberate modifications.

B. MESH STEGANALYSIS

A mesh steganalysis approach was recently presented in [10]. A calibrated mesh is first computed as the difference between the input mesh and its Laplacian smoothing. Next, a high dimensional feature vector is extracted for each calibrated mesh. Based on a training set consisting of a number of feature vectors, a supervised learning process is then carried out resulting in a steganalytic classifier. Experiments show that the approach is effective against several existing steganography algorithms, because all of these algorithms cause changes in the geometry of meshes to some extent and such changes are amplified in the corresponding calibrated mesh. However, just as other similar geometry-based steganalysis approaches [24]–[26], such a steganalytic classifier is not applicable for the permutation-based steganography algorithms where the geometry of the mesh is unchanged after secret data embedding, due to the nature of the approach. Besides, the straightforward universal classifier reported in [10] is not as accurate as the algorithm-specific ones, and it is somehow inconvenient that the steganalytic classifier might need to be trained over again every time it is applied to the detection of a new steganography algorithm.

III. MESH STORAGE STRUCTURE

Although a polygonal mesh can be stored in various formats, such as .obj, .m, .ply, etc, the underlying data structures of these formats are quite similar. A polygonal mesh $M(\mathbf{P}, \mathbf{F})$ is typically defined by a vertex list $\mathbf{P} = \{P_1, P_2, \cdots, P_n\}$ and a face list $\mathbf{F} = \{F_1, F_2, \cdots, F_m\}$, which are usually stored in data files in forms of arrays. Each vertex in the vertex list is further defined by the coordinates in the x, y, z dimensions, and each face in the face list defined by the index references of its containing vertices. Besides, because the edge information is actually implied in face lists, usually edge lists are not specifically stored in data files.

No matter how a mesh is generated, processed and stored, normally the arrangement of its vertex list and face list in the stored data file demonstrates certain patterns. Although no compulsory rules are specifically made, those consecutive vertices P_i and P_{i+1} $(i = 1, 2, \cdots, n-1)$ in $\mathbf{P}$ are likely to be topologically close on the mesh, i.e., only a few vertices lie on the shortest path from P_i to P_{i+1}. Denote $d(P_i, P_{i+1})$, or d_i for simplicity, as the length of the shortest path from P_i to P_{i+1}, where $d_i = n_i + 1$ with n_i being the number of the in-between vertices on that shortest path. In many cases, P_i and P_{i+1} are simply adjacent to each other (in other words, $d_i = 1$). Similar phenomena also exist for the face list $\mathbf{F}$, where consecutive elements F_i and F_{i+1} are quite likely to be topologically near each other.

For example, Fig. 1a shows a frequently used "Mannequin" model in a wireframe display. Fig. 1b and Fig. 1c show a local area of the mesh on which the 10 consecutive vertices $P_1, P_2, \cdots, P_{10}$ and faces $F_1, F_2, \cdots, F_{10}$ are marked, respectively. It can be seen that all these vertices and faces are very close to their corresponding previous and next counterparts in the vertex list and the face list. Moreover, the histogram of all the shortest paths of the consecutive vertices of the mesh is given in Fig. 2e. Among all these vertex pairs, 23.4% of them have the minimum shortest path that equals to 1, 59.3% of them have a shortest path that is no larger than 4, and only 12.2% of them have a shortest path that exceeds 10.

In fact, the vertices and faces of a naturally generated mesh are likely to be stored in good order, no matter how the mesh might be obtained (e.g., by 3D laser scanning, tessellation of a smooth surface, or using modeling softwares such as Autodesk 3ds Max, Blender, MeshLab, etc), and no matter what further geometric processing might be applied on the mesh (e.g., subdivision, re-mesh, simplification, or

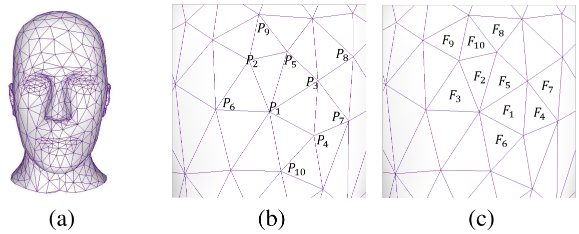


FIGURE 1. The Mannequin model.

shape manipulation). Besides, the underlying ways used to represent meshes, such as the half-edge and winged-edge data structures, also help to generate ordered vertex lists and face lists.

On the other hand, the meshes in which secret messages are embedded with permutation-based steganography methods usually lose such orderliness. Take the models shown in Fig. 2a to Fig. 2d as examples, their histograms of shortest paths between the consecutive vertices before steganography are shown in Fig. 2e to Fig. 2h, respectively. However, after applying the steganography method in [7] on these meshes, the corresponding histograms change to the ones shown in Fig. 2i to Fig. 2l. Now, the lengths of the shortest paths have increased significantly for the majority of the consecutive vertices pairs, and the histograms demonstrate a Gaussian-like distribution.

This is because all the permutation-based steganography methods effectively cause a shuffle of the vertex list and the face list. Based on the different methods and different secret messages, different permutations of the vertices and the faces are obtained. Nevertheless, the common thing is that the new vertex list and face list are often highly randomized. Such discrepancy between the naturally generated meshes and the tampered meshes can be exploited to develop steganalysis approaches.

IV. STEGANALYSIS AGAINST PERMUTATION-BASED MESH INFORMATION HIDING

This section describes a set of steganalysis approaches against permutation-based mesh information hiding methods. The approaches are designed to be universal, meaning that we do not have to be informed of which specific steganography is adopted. Depending on different hypotheses and settings, we identify three specific types of mesh steganalysis problems, and provide solutions respectively. Due to similarity between the analysis work on vertex lists and face lists, without loss of generality, the following discussion mainly focuses on the detection of anomalies in vertex lists.

A. THE STEGANALYSIS PROBLEM AGAINST FULL EMBEDDING

We first investigate the following steganalysis problem:

Problem 1 (Steganalysis Against Full Embedding): Assume a mesh can be normally generated or embedded with secret messages at its full capacity by using an unspecified permutation-based steganography method. Given a mesh $M(\mathbf{P}, \mathbf{F})$, decide whether it is a clean mesh (i.e., naturally generated) or a stego-mesh (i.e., tampered).

It is known that the vertex list is relatively orderly for a clean mesh, while it exhibits high randomness for a stego-mesh. Therefore we propose to use a distance term $D(\mathbf{P})$ to measure the orderliness of a vertex list $\mathbf{P}$:

$$D(\mathbf{P}) = \frac{\sum_{i=1}^{n-1} d(P_i, P_{i+1})}{n-1} \tag{1}$$

where $d(P_i, P_{i+1})$ is the shortest path between P_i and P_{i+1}. $D(\mathbf{P})$ is usually quite small for clean meshes, where consecutive vertices in $\mathbf{P}$ are close to each other. On the other hand, if embedded with secret messages, most consecutive vertices in $\mathbf{P}$ are no longer close, and the value of $D(\mathbf{P})$ can be quite large.

Since the steganography process can result in any random permutation, for a stego-mesh we in fact have:

$$D(\mathbf{P}) \approx d(u, v) \tag{2}$$

where u, v are two random vertices in $\mathbf{P}$. Assuming that the probability of any pair (u, v) being picked is even, the expectation of $d(u, v)$ (as well as $D(\mathbf{P})$) can be derived as:

$$E = \sum_{\forall u, v \in \mathbf{P}} \frac{2}{n(n-1)} d(u, v) \tag{3}$$

Let $\mathbf{F} = \{f_{ij}\}_{n \times n}$ be a matrix where $f_{ij} = d(P_i, P_j)$. $\mathbf{F}$ can be obtained by applying the Floyd–Warshall algorithm [27], [28] on the mesh. Thus, we further have

$$E = \frac{1}{n(n-1)} \mathbf{e}^T \mathbf{F} \mathbf{e} \tag{4}$$

where $\mathbf{e}$ is an $n \times 1$ column vector whose entries are all 1's.

An important observation according to the experiments on a large number of meshes is that, for a stego-mesh we have $D(\mathbf{P}) \approx E$; while for a clean mesh, the difference between $D(\mathbf{P})$ and E is large.

A few mesh models shown in Fig. 3 further clarifies this property. These models vary in size (tiny, medium, large), topology (low genus, high genus), and semantics (free-form models, CAD models). For each model, we calculate $D(\mathbf{P})$ for both the clean mesh and the 4 stego-meshes generated using different secret messages, and compare them with the expectation E calculated using Equation 4. As can be seen from the results given in Table 1, the $D(\mathbf{P})$ of a clean mesh is significantly different from its corresponding E, while for all stego-meshes the $D(\mathbf{P})$ values are very close to E.

Based on the above observation, our discrimination method is concise and effective. *If $D(\mathbf{P}) > \sigma E$, where σ is a positive quantity slightly smaller than 1, M is considered as containing hidden information in its vertex list; otherwise, M is considered as clean.* σ serves as a stabilizer to the method which helps to reduce the occurrences of possible false negatives.

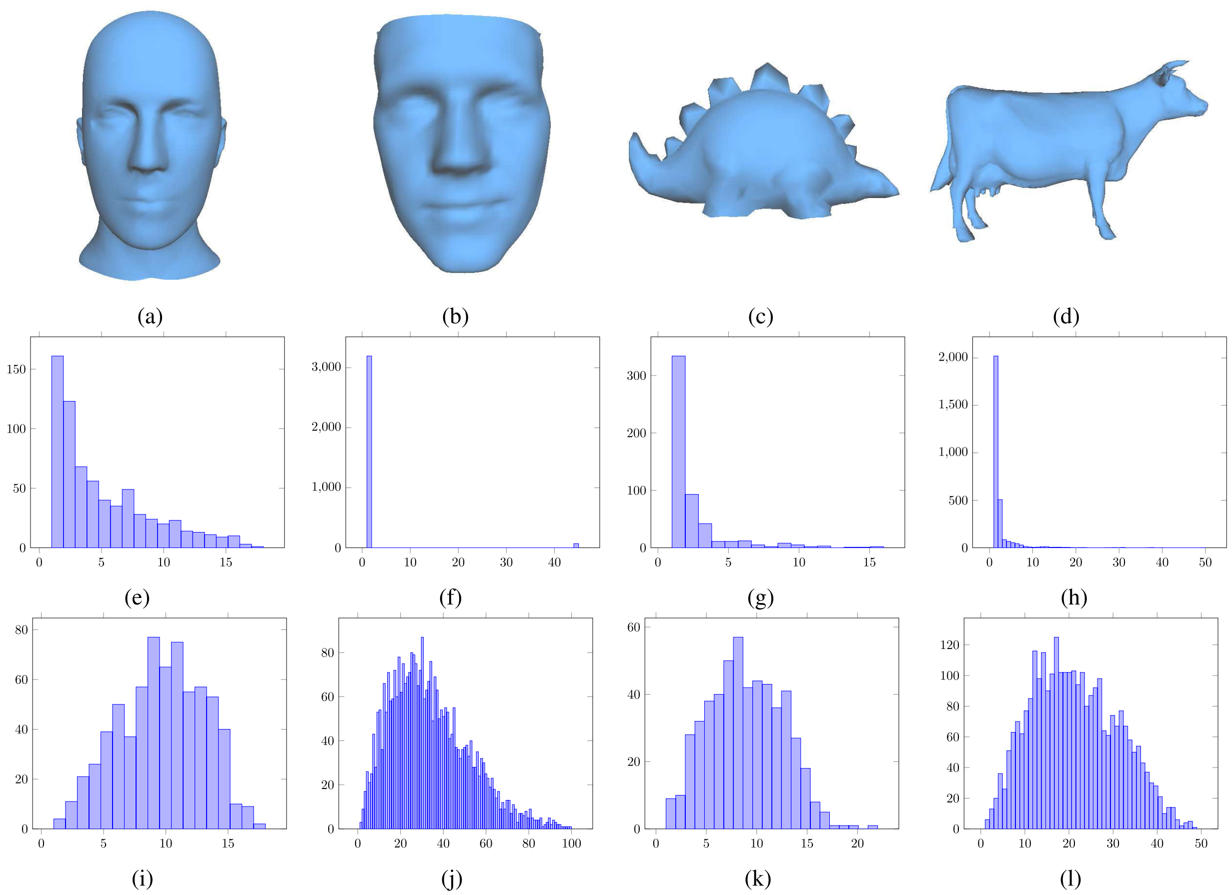


FIGURE 2. The histograms of shortest paths of several meshes before and after permutation-based steganography.

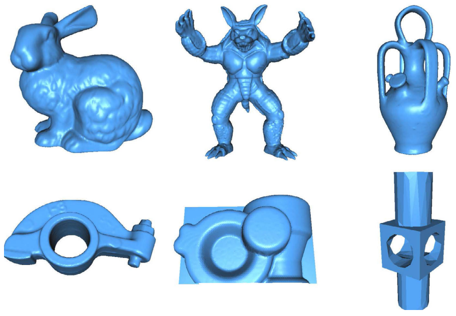


FIGURE 3. Example models of various types.

TABLE 1. The relationship between *E* and *D*(**P**) values for the clean and stego models in Fig. 3.

model		bunny	armadillo	botijo	rockerarm	stamp	mechpart
E		72.99	66.05	37.89	74.25	28.61	3.99
$D(\mathbf{P})$	clean	22.14	39.84	2.35	38.54	1.82	2.53
	stego 1	71.68	65.98	37.83	74.05	28.93	3.99
	stego 2	71.84	66.16	37.72	74.38	28.69	4.04
	stego 3	71.94	66.25	37.78	74.36	28.71	3.98
	stego 4	71.75	66.29	37.73	74.32	28.64	4.10

B. THE STEGANALYSIS PROBLEM AGAINST PARTIAL EMBEDDING

In the domain of image steganography and steganalysis, a common practice is to embed secret data into an image using only part of the image's embedding capacity to make the presence of the payload less detectable. Consequently, a good steganalysis method should be able to identify partially embedded images with a fair detection rate. Usually, the lower the embedding ratio, the harder can the steganography to be detected.

Here we are going to discuss similar cases in the mesh domain. First we need to extend the previous permutation-based steganography methods by allowing partial embedding, which has rarely been considered before. For a mesh with n vertices, the maximum embedding capacity is $\lfloor \log_2(n!) \rfloor$ bits in theory. Now, suppose we only embed $r\lfloor \log_2(n!) \rfloor$ bits into it, where $r \in (0, 1)$. Thus, only x vertices are involved in the steganography process, where x can be calculated in a numerical way:

$$\lfloor \log_2(x!) \rfloor >= r\lfloor \log_2(n!) \rfloor \tag{5}$$

Using a secret key agreed between the sender and the receiver, the x specific vertices are selected from the vertex list. These x vertices can then be used to carry secret messages by an appropriate reordering, adopting any previous steganography

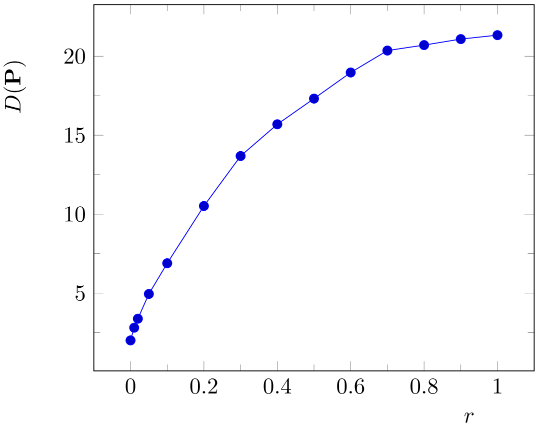


FIGURE 4. For a same model, larger embedding ratios r result in larger $D(\mathbf{P})$.

method. In the meantime, the positions of the other $n-x$ vertices in the vertex list are kept unchanged.

Next, the approach of breaking the permutation-based steganography with partial embedding is discussed.

Problem 2 (Steganalysis Against Fixed-Ratio Embedding): Assume a mesh can be normally generated, or embedded with secret messages by using an unspecified permutation-based steganography method at a known fixed embedding ratio r, where $r \in (0, 1)$. Given a mesh $M(\mathbf{P}, \mathbf{F})$, decide whether it is a clean mesh or a stego-mesh with the embedding ratio r.

Through a large amount of experiments, some nice properties for partially embedded meshes can be observed. First, if a mesh is embedded with payload of ratio r, we almost always have $D(\mathbf{P}) \approx E_r$, where E_r is a constant; while for a clean mesh, $D(\mathbf{P})$ is noticeably smaller than E_r. Besides, for two different ratios $a, b \in (0, 1)$, if $a < b$, then $E_a < E_b$. That is to say, the value of $D(\mathbf{P})$ increases monotonically with the embedding ratio r, as shown in Fig. 4. $D(\mathbf{P})$ increases quickly when the embedding ratio is in $[0.01, 0.5]$; after that, the speed of growth for $D(\mathbf{P})$ is gradually slowed down.

Since partial embedding greatly complicates the situation, the effort to derive a closed form expression for E_r is not trivial. Instead, we obtain E_r by construction. No matter the given mesh $M(\mathbf{P}, \mathbf{F})$ is clean or not, we build a new vertex list $\mathbf{P_0}$ by rearranging $\mathbf{P}$ so that any consecutive vertices in $\mathbf{P_0}$ are made as near to each other as possible. $M(\mathbf{P_0}, \mathbf{F})$ can be regarded as a clean mesh. Next, by adding payload of ratio r into the mesh, we obtain a stego-mesh $M(\mathbf{P_r}, \mathbf{F})$ with a designated embedding ratio. $D(\mathbf{P_r})$ can be served as a good estimation for E_r. Finally, the discrimination strategy is just the same: *If* $D(\mathbf{P}) > \sigma D(\mathbf{P_r})$*, where* σ *is a positive quantity slightly smaller than* 1*,* M *is considered as containing hidden information with embedding capacity of ratio r; otherwise,* M *is considered as clean.*

Finally, we look into the steganalysis problem in a more general form:

Problem 3 (Steganalysis Against Arbitrary-Ratio Embedding): Assume a mesh can be normally generated, or be embedded with secret messages by using an unspecified permutation-based steganography method. Also assume that the embedding ratio a is an arbitrary number in $(0, 1]$. Given a mesh $M(\mathbf{P}, \mathbf{F})$, decide whether it is a clean mesh or a stego-mesh.

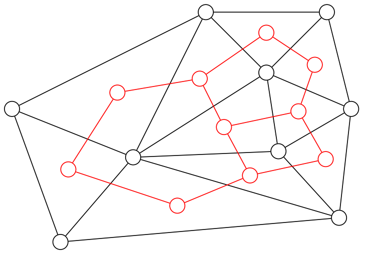


FIGURE 5. A local region of a mesh (in black) and its dual graph (in red).

It is hard to draw a line that perfectly separates stego-meshes from clean ones. It is possible that some normally generated meshes have disorderly vertex list, and thus have relatively large $D(\mathbf{P})$, while some meshes are encoded with secret messages at very low embedding ratio, and thus still have relatively small $D(\mathbf{P})$. As a result, false alarms on clean meshes and misses on stego-meshes are usually inevitable.

Therefore, our solution is to set a reasonable threshold, e.g., $D(\mathbf{P_{0.15}})$, the measure for 15% embedding. *Meshes whose* $D(\mathbf{P})$ *are larger than* $D(\mathbf{P_{0.15}})$ *are identified as stego-meshes; otherwise, consider them as clean.*

C. THE CASE FOR THE FACE LISTS

At last, similar methodology can be adopted to detect the presence of secret messages hidden in face lists. Define another distance term $D(\mathbf{F})$ to measure the orderliness of the face list $\mathbf{F}$:

$$D(\mathbf{F}) = \frac{\sum_{i=1}^{m-1} d(F_i, F_{i+1})}{m-1} \tag{6}$$

where $d(F_i, F_{i+1})$ is the shortest distance between F_i and F_{i+1}. To calculate $d(F_i, F_{i+1})$, we need to construct a dual graph G of the mesh M. In the dual graph, each face in M now becomes a vertex, and there is an edge between the two vertices in G if the two corresponding faces in M are adjacent. Refer to Fig. 5 for an illustration. After that, we find the two vertices in G that are a dual of F_i and F_{i+1}, and let $d(F_i, F_{i+1})$ equal the shortest path between these two vertices on the graph G.

Once the measure $D(\mathbf{F})$ is established, a method similar to the above can be used to carry out steganalysis on face lists.

D. EXAMPLES AND DISCUSSIONS

Now, we examine effectiveness of the proposed steganalysis approaches through a number of experiments. In total 412 models are used for validation, integrating several popular model repositories such as the Princeton Mesh Benchmark [29], MIT CSAIL database, the Stanford 3D Scanning Repository, Caltech Mesh Compendium, and some others frequently used models. Models in the dataset range from tiny to large sizes and differ in geometry and topology, ensuring diversity of the testing samples.

TABLE 2. Summary of results for Experiment 1.

True Positive 239	**False Positive** 1
False Negative 1	**True Negative** 259
Sensitivity $= TP/(TP+FN)$ $= 99.58\%$	**Specificity** $== TN/(FP+TN)$ $= 99.62\%$
Accuracy $= (TP+TN)/(TP+FN+FP+TN) = 99.6\%$	

It should be mentioned that, although most experiments and examples in this work are based on triangular meshes, the proposed steganalysis approaches actually make no assumptions on the type of meshes being processed. Thus, there is no technical barrier in applying the proposed approaches to general polygonal meshes.

We verify effectiveness of the proposed steganalysis approach by testing it against several major permutation-based steganography methods, including the ones reported in [6]–[9], [23]. Since our approach is designed to be universal, we do not separate the stego-models generated by different methods for the purpose of detection. Also, in all experiments, the parameter σ is set to 0.98.

1) EXPERIMENT 1

The dataset for Experiment 1 contains 500 models, in which 260 are clean, and 240 are stego-models. Each stego-model is generated by applying a random steganography method listed above on a clean model, with full embedding capacity. The goal is to use our steganalysis approach to correctly identify clean and stego meshes in the dataset. We first detect the presence of hidden information in the vertex list of the models. For each model, we calculate its current $D(\mathbf{P})$ and the expectation E of $D(\mathbf{P})$ for a random stego-mesh. Then, we compare these two terms to decide the attribute of the mesh. The testing results are summarized in Table 2. Sensitivity, specificity, and accuracy are calculated from true positive (TP), false positive (FP), true negative (TN), and false negative (FN), where sensitivity $= TP/(TP+FN)$, specificity $= TN/(FP+TN)$, and accuracy $= (TP+TN)/(TP+FN+FP+TN)$. As can be seen, the proposed approach is quite effective against different steganography methods and has achieved nearly perfect detection performance. Among all the 500 testing samples, only one false alarm and one miss have occurred. Compared to the steganalysis approach described in [10], although the proposed approach does not target at the detection of the same type of stego-meshes, it is still surprising that the proposed approach achieves a much higher accuracy than [10] (99.6% to 80%).

We have also examined how the approach performs in the situations where secret payloads are added to the face list of models. Again, we use the same dataset of 500 models for testing, and as expected the result is satisfactory (refer to Table 3).

TABLE 3. Steganalysis on the face list of meshes.

True Positive 240	**False Positive** 0
False Negative 0	**True Negative** 260
Sensitivity $= TP/(TP+FN)$ $= 100\%$	**Specificity** $== TN/(FP+TN)$ $= 100\%$
Accuracy $= (TP+TN)/(TP+FN+FP+TN) = 100\%$	

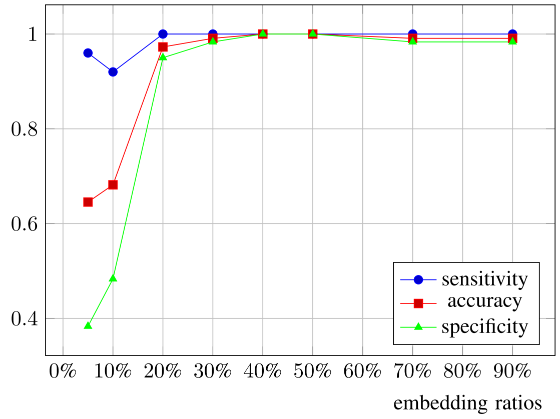


FIGURE 6. Summary of results for Experiment 2.

2) EXPERIMENT 2

Next, we apply the steganalysis approach on the datasets containing clean meshes and partially embedded meshes. The experiment is carried out for several times, each time with a fixed embedding ratio for stego-meshes. The tested embedding ratios range from 90% to 5%. The results are summarized in Fig. 6.

It can be seen that when the embedding ratio is larger than 20%, the approach performs well, with accuracy, sensitivity and specificity all close to 1. However, when the embedding rate is lowered to 10% and 5%, although sensitivity is still high, specificity drops considerably. Therefore the overall accuracy becomes poorer. This is due to the fact that the vertex list and the face list of clean meshes are sometimes not necessarily perfectly well-ordered, if not in a completely random order. Therefore, when compared with $E_{0.1}$ or $E_{0.05}$, steganalysis may fail.

3) EXPERIMENT 3

Finally, we mix all types of meshes into a single dataset and evaluate the performance of our approach. The dataset includes 450 stego-meshes of various embedding ratios (50 meshes each for 5%, 10%, $\cdots$, 100%), as well as 400 clean meshes. For each mesh in the dataset, we compute its $D(\mathbf{P})$ and compare it with the corresponding $E_{0.15}$. The result is presented in Table 4. Both sensitivity and specificity are at a satisfactory level, showing that the proposed approach is effective and universal against all existing permutation-based mesh steganographic methods with arbitrary embedding ratios. However, some clean meshes are mistakenly

TABLE 4. Summary of results for Experiment 3.

True Positive 405	**False Positive** 85
False Negative 45	**True Negative** 315
Sensitivity $= TP/(TP+FN)$ $= 90\%$	**Specificity** $== TN/(FP+TN)$ $= 78.75\%$
Accuracy $= (TP+TN)/(TP+FN+FP+TN) = 84.71\%$	

judged as stego, because their original structures are not orderly enough. On the other hand, some stego-meshes, mainly with low embedding rate such as 10% and 5%, are mistakenly judged as clean. In general, false alarm and miss are a tradeoff, and it is always a challenge to achieve both high sensitivity and specificity.

V. SECURITY-IMPROVED PERMUTATION-BASED STEGANOGRAPHY

Previous permutation-based steganography methods make full use of the encoding power of the vertex list (and face list). Depending solely on the value of the message bits, any of the remaining unpicked $(n - i + 1)$ vertices in the original list is allowed to be selected as the i-th element in the new vertex list. Although such a practice helps in achieving relatively large embedding capacities, it results in a mesh of large $D(\mathbf{P})$ and thus leads to a major issue of exposing the stego-meshes to the above steganalysis technique.

In this section, we present an improved permutation-based mesh steganography method. Our goal is to generate stego-meshes that have a similar storage structure as normal meshes and thus do not look suspicious. We attempt to find a trade-off between the embedding capacity and security. Instead of forming the new vertex list and face list arbitrarily, we apply certain constraints to make sure that the currently picked element is to some extent related to the previously picked one.

As an example, we encode the vertex list $\mathbf{P} = \{P_1, P_2, \cdots, P_n\}$. First, a vertex reference ordering $\mathbf{R} = \{R_1, R_2, \cdots, R_n\}$ of the original vertex list is obtained using techniques such as the one presented in [22]. By fixing the position of the first vertex in $\mathbf{R}$ (as well as the first face in the face reference ordering) when encoding, the same ordering can be precisely recovered during decoding. We then seek an encoded vertex list $\mathbf{Q} = \{Q_1, Q_2, \cdots, Q_n\}$ based on the reference ordering $\mathbf{R}$ and the secret message T.

An essential part of our approach is the process of finding Q_{i+1}, given that $Q_1, Q_2, \cdots, Q_i$ are already found. We examine the 1-ring neighborhood of Q_i, as illustrated in Fig. 7. We use this local neighborhood to carry the next few bits in T. It is possible that some vertices in this neighborhood have already been picked as Q_j $(1 \leq j < i)$, while others not picked yet. Of all the unpicked vertices, find the vertex that appears first in $\mathbf{R}$, and assign it with an alias v_1. Then, all other unpicked vertices are denoted as $v_2, v_3, \cdots, v_k$, in a clockwise order. Based on the value represented by the next $\lfloor \log_2(k) \rfloor$ bits in T, the corresponding vertex in $v_1, v_2, \cdots, v_k$ is picked as Q_{i+1}.

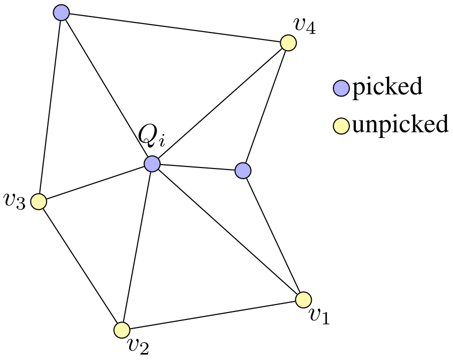


FIGURE 7. The 1-ring neighborhood of Q_i.

In case the 1-ring neighborhood of Q_i does not contain any unpicked vertex, we simply extend the searching range to the 2-ring neighborhood of Q_i. By carrying out the above process repeatedly, a new vertex list $\mathbf{Q}$ is obtained. The whole steganography approach is summarized in Fig. 8. By such an approach, it can be guaranteed that in the vertex list of stego-meshes, most pairs of adjacent vertices are near each other topologically like naturally generated meshes.

The process of encoding the face list is similar except that, to find the next element in the new face list, we now examine the 1-ring neighborhood of the previously picked face. The 1-ring neighborhood of a face can be defined as a set consisting of all the faces that share a common vertex or edge with that face, as illustrated in Fig. 9.

Finally, extraction of the secret messages from a stego-mesh is trivial. From a vertex list or a face list, by reconstructing the reference orderings and sequentially observing the neighborhood information of each element in the given list, the embedded secret information can be precisely retrieved.

A. EXAMPLES AND DISCUSSIONS

Now we perform testings on effectiveness of the proposed permutation-based steganography.

1) EXPERIMENT 4

Using various types of media as the secret messages, e.g., texts, images, audios, and videos, we apply the proposed steganography approach on a number of mesh models. Some results are provided in Table 5, including basic information of the models, expectation E of $D(\mathbf{P})$ after a normal steganography operation, embedding capacity of the proposed approach (in bits/vertex), and $D(\mathbf{P})$ value after data embedding. The approach works well for both small-sized and large-sized meshes. As can be seen, there is a significant difference between E and $D(\mathbf{P})$ for all the models, which is an important factor that prevent the stego-meshes produced by our approach being detected. Unlike previous approaches whose embedding capacity is related to the size of the mesh

```
Input: A mesh $M(\mathbf{P},\mathbf{F})$, with the vertex list
    $\mathbf{P} = \{P_1, P_2, \cdots, P_n\}$ and the face list
    $\mathbf{F} = \{F_1, F_2, \cdots, F_m\}$; a binary message string $T$.
Output: A new vertex list $\mathbf{Q} = \{Q_1, Q_2, \cdots, Q_n\}$, which
    is a reordering of $\mathbf{P}$ with regard to $T$.
    Calculate a reference ordering $\mathbf{R} = \{R_1, R_2, \cdots, R_n\}$;
    Mark all the vertices as unpicked;
    Let $Q_1 = R_1$;
    Mark $Q_1$ as picked;
    $i = 1$;
    repeat
        if $Q_i$'s 1-ring neighborhood contains unpicked vertices then
            Find all the unpicked vertices in $Q_i$'s 1-ring
            neighborhood, and form a temporary set $V$;
        else if $Q_i$'s 2-ring neighborhood contains unpicked vertices then
            Find all the unpicked vertices in $Q_i$'s 2-ring
            neighborhood, and form a temporary set $V$;
        else
            Find all the unpicked vertices in $\mathbf{R}$, and form a
            temporary set $\mathbf{V}$;
        end if
        Let the number of vertices in $\mathbf{V}$ be $k$;
        if $\mathbf{V}$ consists of $Q_i$'s 1-ring or 2-ring neighborhood then
            Among all the vertices in $\mathbf{V}$, assign the one that
            appear first in $\mathbf{R}$ with an alias $v_1$;
            Assign the other vertices in $\mathbf{V}$ with aliases
            $v_2, v_3, \cdots, v_k$ respectively, in a clockwise order;
        else
            Assign all the vertices in $\mathbf{V}$ with aliases
            $v_1, v_2, \cdots, v_k$, according to the order in $\mathbf{R}$;
        end if
        Get the next top $\lfloor \log_2(k) \rfloor$ bits in $T$, and denote the
        value that these bits represent as $val$;
        Let $Q_{i+1} = v_{val+1}$ and mark $Q_{i+1}$ as picked;
        $i$ ++;
    until $i == n$
```

FIGURE 8. Encoding the vertex list based on the secret message.

(numbers of vertices and faces), capacity of the proposed method is independent of the mesh size.

Finally, by following a reverse process, all the embedded data in the meshes can be correctly recovered.

2) EXPERIMENT 5

In order to increase embedding capacity, one alternative is to increase the searching range for the next vertex/face from the 1-ring neighborhood to the 2-ring neighborhood of the current vertex/face. From the results given in Table 6, it can be observed that embedding capacities for all examples are boosted by roughly 78% to 87%. In the meantime, increases in $D(\mathbf{P})$ are slight.

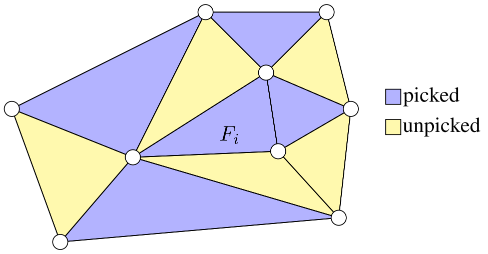


FIGURE 9. The 1-ring neighborhood of F_i.

TABLE 5. Summary of results for Experiment 4.

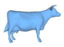
name		face	stegosaurus	cow	botijo
model					
#vertices		3266	533	2904	14859
#faces		6300	1028	5804	29734
E		34.27	8.96	21.44	37.89
text	capacity	5.04	4.98	5.21	5.20
	$D(\mathbf{P})$	2.06	1.65	1.62	1.68
image	capacity	5.01	5.06	5.21	5.20
	$D(\mathbf{P})$	1.82	1.72	1.64	1.67
audio	capacity	4.96	5.07	5.19	5.20
	$D(\mathbf{P})$	1.34	1.76	1.50	1.71
video	capacity	5.00	5.04	5.20	5.20
	$D(\mathbf{P})$	1.61	1.70	1.57	1.71

TABLE 6. Summary of results for Experiment 5.

name	face	stegosaurus	cow	botijo
#vertices	3266	533	2904	14859
#faces	6300	1028	5804	29734
capacity	8.97	9.30	9.36	9.49
$D(\mathbf{P})$	2.10	1.99	2.02	2.14

3) EXPERIMENT 6

We add the stego-meshes generated by the security-improved steganography approach into the dataset used in the steganalysis Experiment 3, and run the steganalysis testing over again. Just as expected, all these models are tagged as "clean".

VI. CONCLUSION

In this paper, we proposed a technique for breaking the permutation-based mesh steganography, which can be used to detect the presence of either full embedding or partial embedding. The approach is universal, and does not require any prior knowledge about which specific steganography method might be used, or what the embedding ratio might be. Unlike many other classification methods, our approach does not require any training process, making it useful when training data are unavailable or insufficient. Last but not least, the approach is effective as shown by a number of experiments.

Also, we presented an improved approach towards secure permutation-based steganography. The approach takes advantage of the connectivity information of meshes, and is capable of hosting secret messages in meshes while keeping mesh files in ordinary structures. This makes the generated stego-meshes hard to be detected.

On the other hand, this work still leaves some room for further investigation. First, the theoretical foundation of the steganalysis approach could be enhanced. We have made some observations in this paper such as the fact that $D(\mathbf{P})$ of a tampered mesh almost always equals its expectation. The underlying theory accounting for such observation is worth exploration. As a matter of fact, one can never rule out the situations where some clean meshes are just awfully organized. These meshes may have relatively large $D(\mathbf{P})$, and it is sometime not easy to differentiate them from a stego-mesh with a 5% embedding rate. To better solve this problem, more studies on the structure of normally generated meshes should be carried out to find their most distinct features.

Finally, to obtain higher embedding capacity, we need to develop more advanced steganography strategies by further exploiting the geometrical and topological characteristics of meshes.

REFERENCES

[1] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[2] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding-A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[3] C.-M. Wang and Y.-M. Cheng, "An efficient information hiding algorithm for polygon models," *Comput. Graph.Forum*, vol. 24, no. 3, pp. 591–600, Sep. 2005.

[4] C.-M. Wang and P.-C. Wang, "Steganography on point-sampled geometry," *Comput. Graph.*, vol. 30, no. 2, pp. 244–254, Apr. 2006.

[5] M. W. Chao, C. H. Lin, C. W. Yu, and T. Y. Lee, "A high capacity 3D steganography algorithm," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 2, pp. 274–284, Mar. 2009.

[6] Y.-M. Cheng and C.-M. Wang, "A high-capacity steganographic approach for 3d polygonal meshes," *Vis. Comput.*, vol. 22, nos. 9–11, pp. 845–855, Aug. 2006.

[7] A. Bogomjakov, C. Gotsman, and M. Isenburg, "Distortion-free steganography for polygonal meshes," *Comput. Graph. Forum*, vol. 27, no. 2, pp. 637–642, 2008.

[8] N. C. Huang, M. T. Li, and C. M. Wang, "Toward optimal embedding capacity for permutation steganography," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 802–805, Sep. 2009.

[9] S.-C. Tu and W.-K. Tai, "A high-capacity data-hiding approach for polygonal meshes using maximum expected level tree," *Comput. Graph.*, vol. 36, no. 6, pp. 767–775, Oct. 2012.

[10] Y. Yang and I. Ivrissimtzis, "Mesh discriminative features for 3D steganalysis," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 3, p. 27, Apr. 2014.

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

[12] K. Wang, G. Lavoue, F. Denis, and A. Baskurt, "Hierarchical watermarking of semiregular meshes based on wavelet transform," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 620–634, Dec. 2008.

[13] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[14] F. Perez-Gonzalez and F. Balado, "Quantized projection data hiding," in *Proc. Int. Conf. Image Process.*, vol. 2, Sep. 2002, pp. II–II.

[15] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding: Steganography and Watermarking-Attacks and Countermeasures* (Advances in Information Security), vol. 1. Boston, MA, USA: Springer, 2001.

[16] F. Cayre and B. Macq, "Data hiding on 3-D triangle meshes," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 939–949, Apr. 2003.

[17] R. Ohbuchi, H. Masuda, and M. Aono, "Watermaking three-dimensional polygonal models," in *Proc. 5th ACM Int. Conf. Multimedia*, New York, NY, USA, 1997, pp. 261–272.

[18] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 551–560, May 1998.

[19] Y. Yang, N. Peyerimhoff, and I. Ivrissimtzis, "Linear Correlations between Spatial and Normal Noise in Triangle Meshes," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 1, pp. 45–55, Jan. 2013.

[20] *Stegano Gif Palette Order*. Accessed: Nov. 1, 2019. [Online]. Available: http://users.skynet.be/glu/sgpo.htm

[21] *GIF Colourmap Steganography*. Accessed: Nov. 1, 2019. [Online]. Available: http://www.darkside.com.au/gifshuffle/

[22] J. Rossignac, "Edgebreaker: Connectivity compression for triangle meshes," *IEEE Trans. Vis. Comput. Graph.*, vol. 5, no. 1, pp. 47–61, Jan. 1999.

[23] S.-C. Tu, W.-K. Tai, M. Isenburg, and C.-C. Chang, "An improved data hiding approach for polygon meshes," *Vis. Comput.*, vol. 26, no. 9, pp. 1177–1181, Sep. 2010.

[24] Z. Li and A. G. Bors, "3D mesh steganalysis using local shape features," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2144–2148.

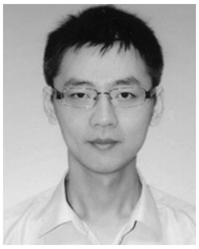
[25] D. Kim, H. U. Jang, H. Y. Choi, J. Son, I. J. Yu, and H. K. Lee, "Improved 3D mesh steganalysis using homogeneous kernel map," in *Information Science and Applications* (Lecture Notes in Electrical Engineering), vol. 424, K. Kim and N. Joukov, Eds. Singapore: Springer, 2017.

[26] Y. Yang, R. Pintus, H. Rushmeier, and I. Ivrissimtzis, "A 3D steganalytic algorithm and steganalysis-resistant watermarking," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 2, pp. 1002–1013, Feb. 2017.

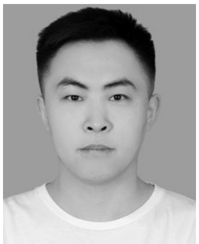
[27] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, p. 345, Jun. 1962.

[28] S. Warshall, "A theorem on Boolean matrices," *J. ACM*, vol. 9, no. 1, pp. 11–12, 1962.

[29] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Trans. Graph.*, vol. 28, no. 3, p. 73, Aug. 2009.



YIMIN WANG received the B.S. degree from Fudan University and the Ph.D. degree from Nanyang Technological University. He is currently with the School of Computer Engineering and Science, Shanghai University. His research interests include computer graphics, visualization, and mixed reality.



LINGSHENG KONG is currently pursuing the degree with the School of Computer Engineering and Science, Shanghai University. His research interests include computer graphics and large-scale imaging computing.

ZHENXING QIAN received the B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), in 2003 and 2007, respectively. He has been a Faculty Member of Shanghai University, since 2019. He is currently a Faculty Member of the School of Computer Science, Fudan University, where he is a Professor. He has published over 100 peer-reviewed articles on journals and conferences. His research interests include information hiding and multimedia security.

GUORUI FENG received the B.S. and M.S. degrees in computational mathematics from Jilin University, China, in 1998 and 2001, respectively, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, China, in 2005. From January 2006 to December 2006, he was an Assistant Professor with East China Normal University, China. In 2007, he was a Research Fellow of Nanyang Technological University, Singapore. He is currently with the School of Communication and Information Engineering, Shanghai University, China. His current research interests include image processing, image analysis, and computational intelligence.

XINPENG ZHANG received the B.S. degree in computational mathematics from Jilin University, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he was a Faculty Member of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. He is also a Faculty Member of the School of Computer Science, Fudan University. He was a Visiting Scholar with the State University of New York at Binghamton, from 2010 to 2011, and an Experienced Researcher with Konstanz University. He was supported by the Alexander von Humboldt Foundation, from 2011 to 2012. His research interests include multimedia security, image processing, and digital forensics. He has published over 200 articles in his research areas. He was an Associate Editor of the IEEE Transactions on Information Forensics and Security, from 2014 to 2017.

JIANMIN ZHENG received the B.S. and Ph.D. degrees from Zhejiang University, China. He is an Associate Professor with the School of Computer Science and Engineering and the Co-Director of the Institute for Media Innovation, Nanyang Technological University, Singapore. His research interests include computer graphics, geometric modeling, 3-D scanning and imaging, 3-D printing, visualization, and AR/VR. He is currently a member of the Executive Committee of Asia Association for Computer Graphics and Interactive Technology. He has been serving as a program committee member for many international conferences in the areas of computer graphics and geometric modeling. He is an associate editor of several international journals.