IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 16, 2017, accepted October 21, 2017, date of publication October 30, 2017,
date of current version November 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2767072

A High-Capacity 3D Steganography Algorithm With Adjustable Distortion

NANNAN LI[1], JIANGBEI HU[1], RIMING SUN[2], SHENGFA WANG[3], AND ZHONGXUAN LUO[3]
[1]School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China
[2]School of Science, Dalian Jiaotong University, Dalian 116028, China
[3]DUT-RU International School of Information and Software Engineering and the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology, Dalian 116024, China

Corresponding author: Shengfa Wang (sfwang@dlut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772104, Grant 61432003, Grant 61720106005, and Grant 2017YFB1103700, in part by the Fundamental Research Funds for the Central Universities under Grant DUT17JC27 and Grant DUT2017TB02, and in part by the Startup Funds for Ph.D. Research under Grant 201601254.

ABSTRACT In this paper, we devise a novel steganography algorithm that has a high capacity while still retaining the ability of adjusting the embedding distortion. A shifting strategy is explored to embed the secret data into a given 3-D model effectively. In order to reduce the embedding distortion, we propose a truncated space of data instead of directly working on the critical geometric information from vertices of the cover model. The truncated space confines the distortion of each component of the stego-model within the space that means the embedding distortion could be controlled within a very low threshold. In theory, we can set the length of data truncation to adjust the embedding distortion below a specified level, at the cost of losing certain embedding capacity. Moreover, the embedding capacity is irrelevant to the shape of models, and the quality of the stego-model is mostly dependent on the length of truncation rather than the quantity of embedded secret data. The proposed 3-D steganography method has the capability to control the level of embedding distortion, and at the same time, has a high embedding capacity. Various experiments have demonstrated the flexibility and high performance of our new approach.

INDEX TERMS 3D steganography, high capacity, adjustable distortion.

I. INTRODUCTION

Steganography is an important research topic in computer communication and has drawn much attention from researchers for decades. With the rapid development of computing technologies, increasingly more data are being generated and acquired everyday, which in turn call for more powerful analytical tools to process the stored data for various applications. Steganography algorithms tend to require security, high capacity and low distortion when recovering hidden information.

Information hiding aims to conceal a secret message in other harmless media (also known as cover media), and only the legitimate receiver can extract the secret message with the secret key provided [1], [2]. Generally, information hiding can be roughly classified into two categories: steganography and watermarking. The former one attempts to embed as much information as possible into a cover signal, while the later one concentrates more on the robustness of the embedded information at the expense of embedding capacity. In contrast to the information hiding, steganalysis aims at detecting whether a given medium has hidden message in it or not, and if possible, tries to recover the hidden message and the cover signal. It can be used to measure the security performance of information hiding techniques, which should be imperceptible to both human vision systems (HVS) and more intelligent analysis. However, the information hiding and steganalysis for 3D models have received relatively less attention compared to image information hiding [3]–[8], in spite of the proliferation of 3D models which are fairly promising information carriers.

There are several good 3D watermarking methods [9]–[18], and most of them can be adapted for steganography. However, the embedding capacity is very low, since they are not designed for this purpose in the first place. One goal of the steganography algorithm is covert communication [19]–[32]. Therefore, the embedding capacity is its major concern. Most of the existing approaches and techniques investigate the tradeoff between embedding capacity and distortion. Nevertheless, robustness against similarity transformation attacks, such as translation, rotation, and uniform scaling, is still necessary, because such attacks are regarded as the common operations for 3D models. Therefore, higher capacity with lower distortion and robustness still deserves much more investigation in spite of the aforementioned research progresses.

The high embedding capacity and the low embedding distortion are the driving factors for us to propose a new and efficient steganography algorithm. In this paper, we utilize a truncated space of data with shifting strategy to constitute a high-capacity steganography algorithm while keeping the distortion within an adjustable threshold, which could be a very small value in theory. A shifting strategy is proposed to embed the secret data into the model by simply shifting the value in one interval into another one according to the given mapping. The truncated space is explored to further reduce the distortion below a given threshold by adjusting the length of truncation, which also governs the maximum embedding capacity. It should be noted that given a length of truncation, the embedding distortion of each vertex could be limited within the space, and the worst embedding distortion is only related to the length of truncation rather than the quantity of embedded information. It means that the embedding distortion does not grow with the increase of the quantity of embedded information below the maximum embedding. Fig. 1 illustrates the generic pipeline of our approach. A preprocessing of the 3D model is employed to generate a standard model suit for embedding. Then, a truncated space of data is constructed. In the truncated space, each component (such as x,y,and z) is decomposed into several equal-sized intervals in sequence. Then, the secret data can be embedded into the values of the components by shifting them among the intervals. Since the modification of each value is limited within the truncated space, the distortion of the stego-model could be very small. Moreover, we can set the length of truncation to adjust the distortion within a specified threshold. Since we preprocess models with normalization, including translation, rotation, and uniform-scale, the proposed algorithm can withstand the attacks of similarity transformation. The main contributions of this paper include:

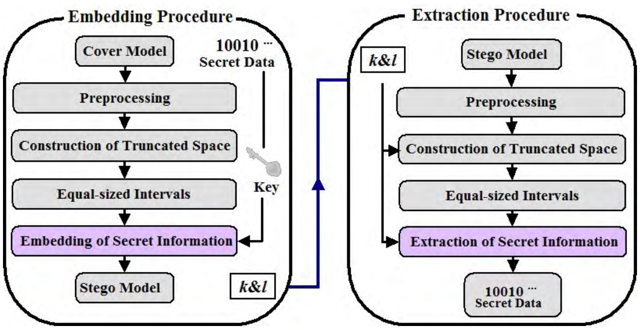


FIGURE 1. The pipeline of our steganography scheme.

- We propose a novel steganography algorithm based on a shifting strategy, which embeds the secret data into a cover model by simply shifting the values in one interval into another one. This strategy guarantees that the embedding distortion does not grow with the increase of the quantity of embedded secret data under the maximum embedding capacity. The proposed algorithm makes use of a simple function, and has high embedding capacity.
- We construct a truncated space of data instead of directly exploiting original vertices information of the cover model, and this process results in a property that the distortion in our steganography algorithm is adjustable by setting the length of truncation. To the best of our knowledge, our work is the first 3D steganography method that has the capability to control the level of embedding distortion while still retaining a high embedding capacity.

II. RELATED WORK

Several steganography/watermarking schemes for 3D meshes have been proposed either in the spatial domain [27], [28], [33]–[36] or in a transformed domain [1], [22], [23], [26], [37]. Oftentimes, embedding in the spatial domain leads to a high capacity, but at the cost of distortion and robustness. In the general framework of steganography/watermarking, transformed domains have shown to offer a better robustness. Since we are interested in maximizing capacity, we plan to embed the information into the spatial domain. In order to reduce the distortion, we exploit a truncated space instead of directly manipulating on the model vertices.

In the spatial domain, Cayre *et al.* [34] proposed a substitutive procedure based steganography algorithm, and their algorithm could combat against similarity transformation and the vertex reordering attacks. But the embedding capacity and the inefficient triangle traversal are the main drawbacks. Then, Wang *et al.* [35] improved the triangle traversal using an advanced jump strategy, and increased the embedding capacity to three bits per vertex via a multi-level embedding procedure. Chao *et al.* [36] presented a multi-layer embedding to obtain a high embedding capacity, which is three times the number of embedded layers. However, the number of embedded layers is limited due to the rapidly increasing distortion. Yang *et al.* [27] computed an appropriate quantization level for the mesh vertices and replace the unused Least Significant Bits (LSB) with watermark bits. It achieved high capacity and low distortion, but the error increases significantly when the amount of embedded noise becomes large, and it could not withstand malicious attacks. Tsai [28] proposed an adaptive steganography algorithm by considering the accuracy of the complexity estimation and the embedding capacity. However, higher capacity with lower distortion and the robustness still deserves much more investigation in spite of the aforementioned research progresses.

In the transformed domain, Cheng and Wang [37] proposed a steganography method that combines both spatial domain and representation domain. In the spatial domain, they explored a multi-level embedding procedure to embed at least three bits into each vertex. In the representation domain, they used a representation rearrange procedure to embed six more bits into each vertex. Bogomjakov *et al.* [22] improved a permutation steganography method to change the ordering of vertices and polygons for data embedding, but the embedding capacity is less than one bit per vertex. Huang *et al.* [23] presented a more effective scheme with an embedding capacity closer to the optimal value under the same time complexity as that of Bogomjakov *et al.* algorithm [22]. Lin *et al.* [26] presented a distortion free steganography algorithm in representation domain using vertex representation orders, triangle representation orders and connectivity information. The algorithm could be combined with other spatial-based steganography methods to provide additional embedding capacity. However, all the existing transformation-based steganography algorithms offer a very limited embedding capacity.

III. SCHEME OF PROPOSED STEGANOGRAPHY

In this work, we propose a high-capacity steganography algorithm by exploiting a truncated space with a shifting strategy. The proposed algorithm consists of two separate procedures: the embedding procedure and the extraction procedure. Both procedures have four main steps: preprocessing, the construction of truncated space, equal-sized intervals division, and data embedding/extraction.

In the embedding procedure, a preprocessing of the 3D model is employed to generate a standard model suit for embedding. Then, a truncated space is constructed, and it will be used to embed the secret data later. The truncation of model data (called truncation-data) is further divided into several equal-sized intervals in an ascending order. The secret data can be embedded into the truncation-data by shifting the truncation-data in one interval to the corresponding positions in other intervals according to a given key mapping, which is called shifting strategy throughout this paper. This strategy limits the worst modification scale of truncation-data, and it guarantees the quality of the stego-model under the embedding capacity. Moreover, we can also set the length of truncation to adjust the distortion within a specified threshold. The extraction procedure is the inverse of the aforementioned procedure in a similar way. More details of the embedding phase will be presented in the later section.

A. PREPROCESSING

Given a 3D model $\boldsymbol{M}$, and V is its vertex set with three components (x, y, and z). We first preprocess the model $\boldsymbol{M}$ to obtain a standard model for embedding using similarity transformation, including the uniform scaling, rotation and translation.

Firstly, build a cuboid bounding box for the given model, and uniformly scale the model to make the length of the longest edge of the corresponding bounding box to be 1.

Secondly, alignment the bounding box using Principal Components Analysis (PCA) with the assist of the barycenter of the model (place the barycenter in the first octant), and obtain the corresponding model with the same rotation.

Finally, move the barycenter of the bounding box to the coordinate origin, and obtain the corresponding model with the same translation.

The preprocessing settles the impact of model diversity on algorithm stability. For convenience, we still use $\boldsymbol{M}$ to denote the cover model, and stego-model denotes the model after embedding.

B. CONSTRUCTION OF TRUNCATED SPACE

In this section, we construct a truncated space, then embed the secret data into the space. The vertex set V is a set in a real number space. We decompose the real number space R into two subspaces: the truncated space R_t and the residual space R_r. Specifically, the vertex set V of the cover model in R^3 could be represented as

$$V = F_s \times (V_t(l) + V_r(l)), \tag{1}$$

where F_s is a sign function that consists of positive or negative sign, $V_r(l)$ is the unsigned residual data in the residual space R_r^3, $V_t(l)$ is the unsigned truncation-data in the truncated space R_t^3, and l is the length of truncation from the decimal point. Concretely, the unsigned value (x_r, y_r, and z_r) in $V_r(l)$ retains the first l decimal places and assigns the other decimal places '0', and the unsigned value (x_t, y_t, and z_t) in $V_t(l)$ assigns '0' to the first l decimal places and retains the other decimal places unchanged. For example, we know that the absolute value of each component (such as x, y or z) of vertex v_i could be represented in form "$0.{*}{*}{*}\ldots{*}{*}{*}$", where $*$ is an integer between 0 and 9, and it contains 15 decimal places ($*$) in our experiment. Given a length of truncation l ($0 \leq l < 15$), data in $V_t(l)$ could be represented in form "$0.0\ldots0{*}{*}{*}$", where the first l decimal places are '0'. Data in $V_r(l)$ could be represented in form "$0.{*}{*}{*}0\ldots0$", where the first l decimal places are unchanged and the rest decimal places are '0'.

The set $V_t(l)$ will be used to embed the secret data, and the stego-model could be obtained as

$$\tilde{V} = F_s \times (\tilde{V}_t(l) + V_r(l)), \tag{2}$$

where $\tilde{V}$ is the vertex set of the stego-model, and $\tilde{V}_t(l)$ is the embedded vertex set in the truncated space.

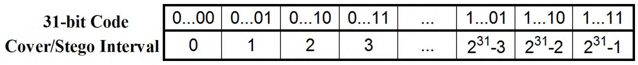
There are two main advantages of using the truncated space. First, the truncated space limits the embedding distortion of each vertex in the space itself, and the worst distortion only depends on the length l. This property allows us to adjust the length l to control the distortion within a specified threshold. Second, in the truncated space, we focus on the truncated values of each component, and it is not directly related to the original shape of cover models, therefore, it makes the performance of our method very stable. Generally, the longer the length l is, the smaller the distortion becomes. In theory, given a distortion threshold, we could adjust the length of truncation to make the worst distortion within the given threshold. It should be noted that the length l will affect the embedding capacity, which will be discussed in the section of experimental results.

C. EQUAL-SIZED INTERVALS DIVISION

After constructing the truncated space $V_t(l)$, three components (x_t, y_t, and z_t) will be handled separately, and the elements (values) of each component corresponding to the vertices. Take x_t-component for example, we divide $[min\{x_t\}, max\{x_t\}]$ into 2^k intervals of equal size in an ascending order, labeled as $\{x_{t_0}, x_{t_1}, x_{t_2}, \cdots, x_{t_{2^k-1}}\}$, where k is an integer that satisfies $0 \leq k < log_2^{10^{15-l}} + 1$ due to the digits limit of Matlab in our experiment. Each interval may contain several values correspond to the vertices. Here, in order to obtain the same intervals in the extraction processing, we ignore two extreme values ($min\{x_t\}$ and $max\{x_t\}$) and the values corresponding to the extreme values ($min\{x_r\}$ and $max\{x_r\}$) in the residual space, these values are not embeddable and unchanged during the embedding processing. We consider all the other values in these intervals as the embeddable ones, which will be used to embedded secret data later. Generally, there are $n - 4$ embeddable values in each component, and $k-$bit secret data can be embedded in each value, where n is the number of model vertices.

D. EMBEDDING PROCEDURE

In the embedding procedure, we process x_t-component, y_t-component and z_t-component in $V_t(l)$, respectively. Firstly, a secret key $Key(k)$ is constructed by the sender, which is used as the seed to generate a mapping between one value in the intervals and the $k-$bit code, where k is considered as the key information that will be sent to the receiver. Let us call this mapping **Keymapping** for short. The **shiftingstrategy** embeds some secret data ($k-$bit code) into a embeddable value by shifting the value in current interval to the corresponding position in another interval according to the Key mapping. When doing embedding, $k-$bit secret data can be hidden in each value, then each cover value in V_t^l will be replaced by a stego-value according to the Key mapping. In order to reduce the burden of transmission, we set the Key mapping between Decimal and $k-$bit Binary in a natural order (see Fig. 2). Also, the Key mapping could be a random correspondence between Decimal and $k-$bit Binary, which will be more secure at the cost of lager burden of transmission. However, the key mapping we used in this approach is secure enough, since there are over thousands other cases when try to obtain the embedded information without the key information k and the length of truncation l. Therefore, the extracted data will be meaningless unless the key information and the length of truncation are both correctly provided.



31-bit Code	0...00	0...01	0...10	0...11	...	1...01	1...10	1...11
Cover/Stego Interval	0	1	2	3	...	2^{31}-3	2^{31}-2	2^{31}-1

FIGURE 2. The Key mapping used in the approach ($k = 31$).

Given a secret message SM = "101010001110...", and it also can be represented in the $k-$bit-stream, such as k = 3, SM = "101 010 001 110....". Given a length of truncation l, we could obtain a corresponding truncated space $V_t(l)$, which limits the embedding distortion in the space. The detail of embedding procedure is listed in Algorithm 1. For example, for a embeddable value x in interval x_{t_0}, 31−bit data "111...1" can be embedded in x by shifting it to the corresponding position $\tilde{x}$ in interval $x_{t_{2^k-1}}$ according to the Key mapping in Fig. 2. In other words, when doing embedding, the cover value x can be embedded with 31− bit secret data just by replacing it with the stego-value $\tilde{x}$. In contrast with the previous steganography algorithms, our scheme is very simple, efficient, and has the adjustable embedding distortion. The embedded process uses only the Key mapping to shift the cover values to the positions in the mapping intervals (the cover value will be shifted by the distance between the current interval and the stego-interval in the direction of the stego-interval).

Algorithm 1 Embedding Procedure

```
Input  : A cover model $\boldsymbol{M}$, the length of truncation $l$
         and a secret message SM.
Output: A Stego-model $\tilde{\boldsymbol{M}}$.
   Step1. Preprocess the model $\boldsymbol{M}$ to obtain a standard
          model for embedding.
   Step2. Construct the truncated space $V_t(l)$.
   Step3. For $x_t-$component ($y$ and $z$ can be processed in
          the same way) in $V_t(l)$, divide
          $[min\{x_t\}, max\{x_t\}]$ into $2^k$ intervals of equal
          size, labeled as $\{x_{t_0}, x_{t_1}, x_{t_2}, \cdots, x_{t_{2^k-1}}\}$. Fix two
          values $min\{x_t\}$ and $max\{x_t\}$, and set the other
          values embeddable.
   Step4. For each value, if the value is embeddable, take
          out $k-$bit data from the secret message SM and
          hide the data into the value by shifting it to
          another interval according to the Key mapping.
   Step5. Repeat Step 3 and 4 until all three components
          are embedded, or all secret data in SM are
          hidden.
   Step6. Reconstruct the stego-model $\tilde{\boldsymbol{M}}$ using Eq. (2).
```

Moreover, the truncated space limits the embedding distortion in $V_t(l)$, in which the distortion is much less than it in the original model space. The modification of the embedded values is limited in space $V_t(l)$, therefore, the quality of stego-models will retain at a high level even in the worst case. The worst case can be obtained by shifting the value in the interval to the corresponding value in the farthest interval, and the overall worst case could be calculated when all the embeddable values are shifted to the farthest intervals. Therefore, given an interval division, the overall worst case is only related to the length of truncation l, and we could adjust the length l to make the distortion (worst case) in a specified range (the longer the length is, the smaller the distortion

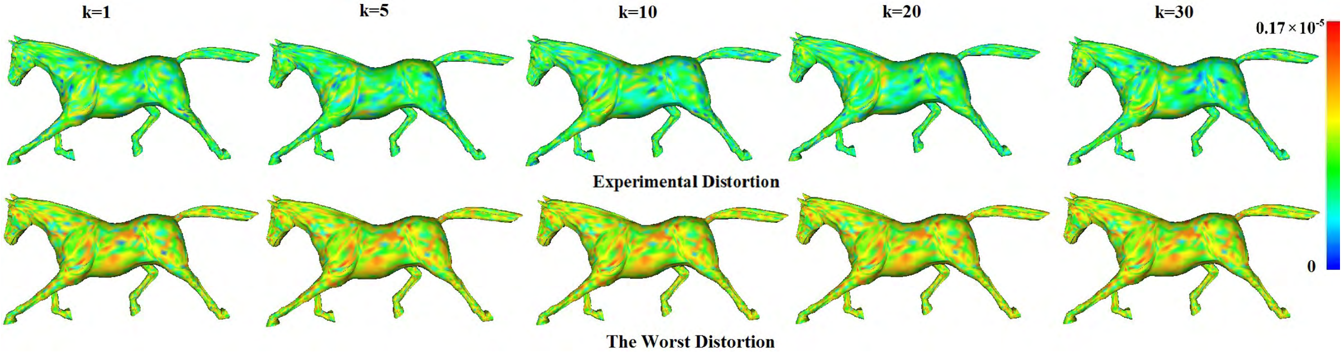


FIGURE 3. An illustration on the distortion of stego-models with different number of embedded data ($k = 1, 5, 10, 20, 30$). Both the experimental distortion (top) and the corresponding worst distortion (bottom) are shown using the same color bar (the largest distortion is 0.17×10^{-5}).

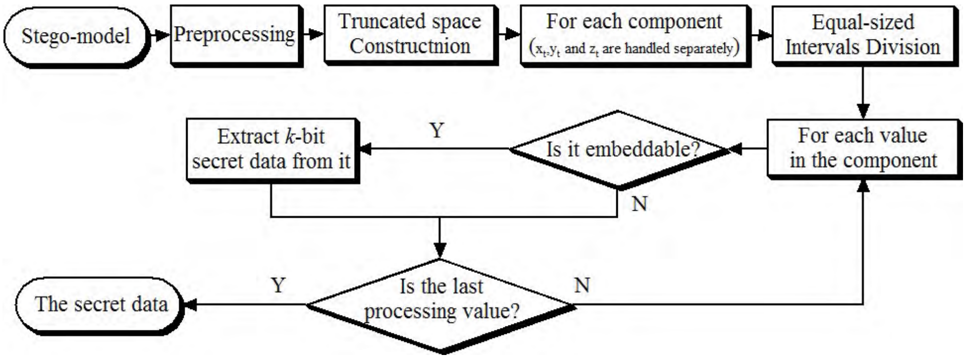

Stego-model
Preprocessing
Truncated space Constructnion
For each component (x_t,y_t and z_t are handled separately)
Equal-sized Intervals Division
For each value in the component
Is it embeddable?
Y
Extract k-bit secret data from it
N
Is the last processing value?
Y
N
The secret data

FIGURE 4. The flowchart of the secret extraction procedure.

becomes). However, the worst case never happens in all the experiments. Even though in the worst case, the quality of the stego-model still remains high, as it is within the range where the human eye can hardly see the distortion. Fig. 3 illustrates both the experimental distortion and the worst case one of the stego-models with different embedded information (measured by root-mean-square error). We can also see that the distortion does not grow with the increase of embedded information.

E. SECRET EXTRACTION

Fig. 4 illustrates the flowchart of the secret extraction procedure. In the secret extraction procedure, we first preprocess the stego-model $\tilde{\boldsymbol{M}}$ and construct the truncated space using the same way in embedding procedure. Since we fix both extreme values for each component, the bounding box is unchanged after embedding. We also have $min\{x_t\} = min\{\tilde{x_t}\}$ and $max\{x_t\} = max\{\tilde{x_t}\}$ (the same is true for y_t and z_t). Then, the equal intervals can be obtained using the received key information k that is used in the embedding procedure. For each component, we can recognize which interval a stego-value belongs to. After that, we find out all the stego-values and their corresponding stego-intervals. The $k-$bit secret data embedded in the stego-value can be extracted according to the Key mapping. Finally, all the secret message SM can be exactly extracted.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, various 3D models are selected to show the embedding capacity, the quality of stego-models in terms of the visualization and the fidelity measure. We want to declare that the proposed method can also be directly applied to point clouds and other representation of 3D models with point information. The models are represented in double precision (each component of vertices contains 15 decimal places) in our experimental environment. The binary secret bit stream SM is randomly generated using the library function randint() in the Matlab 7.6.0 library. The distortion is measured using peak signal-to-noise ratio (PSNR), which can be calculated using RMSE of the cover model and stego-model. The RMSE is defined as

$$\sqrt{\frac{1}{|V|}\sum_{i}^{|V|}\|v_i - \tilde{v}_i\|^2}, \tag{3}$$

where $|V|$ is the number of model vertices, v_i and $\tilde{v}_i$ are the vertices of cover model and stego-model respectively. Then the PSNR can be defined as

$$20\log_{10}(D_{max}/RMSE), \tag{4}$$

where D_{max} is set to be the diagonal distance of the bounding box of the cover model. The embedding capacity (EC) for a

TABLE 1. The correspondence between *l* and the maximum *k*.

l	0	1	2	3	4	5	**6**	7	8	9	10	11	12	13	14
$\max\{k\}$	50	47	44	41	37	34	**31**	27	24	21	17	14	11	7	4

TABLE 2. Relationship between the parameters (*l* and *k*) and the performances (PSNR and ER) on Horse model. Null means it is beyond the embedding capacity.

k PSNR l ER	1	5	10	15	20	25	30	35	40	45	50
0	13.95	13.91	13.92	13.89	13.89	13.93	13.87	13.92	13.94	13.92	13.90
	2.99	14.99	29.98	44.98	59.97	74.96	89.96	104.95	119.94	134.94	149.93
2	50.47	50.32	50.26	50.28	50.28	50.30	50.22	50.32	50.30	Null	Null
	2.99	14.99	29.98	44.98	59.97	74.96	89.96	104.95	119.94		
4	90.30	90.29	90.28	90.22	90.29	90.26	90.27	90.29	Null	Null	Null
	2.99	14.99	29.98	44.98	59.97	74.96	89.96	104.95			
6	130.42	130.26	130.28	130.25	130.29	130.28	130.35	Null	Null	Null	Null
	2.99	14.99	29.98	44.98	59.97	74.96	89.96				
8	170.55	170.47	170.32	170.31	170.29	Null	Null	Null	Null	Null	Null
	2.99	14.99	29.98	44.98	59.97						
10	210.30	210.11	210.26	210.29	Null	Null	Null	Null	Null	Null	Null
	2.99	14.99	29.98	44.98							
12	250.51	250.64	250.55	Null	Null	Null	Null	Null	Null	Null	Null
	2.99	14.99	29.98								
14	290.77	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null
	2.99										

model can also be measured using embedding rate (ER)

$$ER = \frac{EC}{|V|}, \quad EC = 3 \times k \times (|V| - 4), \tag{5}$$

where $0 \leq k < log_2^{10^{15-l}} + 1$ is the quantity of embedded information in each value in x_t, y_t and z_t component, respectively.

A. PARAMETERS AND PERFORMANCES

We first reveal the relationship between the parameters (k and l) and the performances (PSNR and ER) using in our approach. Table 2 shows the performances with selected l and k on Horse model. We could find that the length of truncation l affects the PSNR greatly, but the k has almost no effect on the PSNR. Table 1 lists the correspondence between l and the maximum k (corresponding to the maximum embedding capacity). Then, we explore the relationship in another point of view. Fig. 5 (a) shows that PSNR grows with the increase of l, at the cost of losing certain embedding capacity (Fig. 5 (b)). Fig. 5 also shows that the proposed method is not sensitive to the models. Therefore, the distortion (PSNR) of our steganography algorithm can be controlled by adjusting the length l to make the worst PSNR achieve the given threshold. Since the embedding distortion is limited within the truncated space, the best ER with tolerable distortion is our main consideration. In our experiments, we set $l = 6$ and $k = 31$ as the default value if there is no special declaration. The PSNR values in all our experiments are over 120 for the default parameters. To the best of our knowledge, it is the first steganography method that has the capability to adjust the embedding distortion while still retaining a considerable embedding capacity.

B. THE EMBEDDING CAPACITY AND DISTORTION ANALYSIS

Our approach has a very high embedding capacity while the incurred distortion is small, and could be adjusted within a given threshold. Even when examining multi-layer steganography schemes, we can notice that the numbers of layers in the relevant approaches [26], [36] are limited. The distortion increases significantly with the growth of embedded information in Yang *et al.*'s [27]. Fig. 6 shows the relationship between distortion (PSNR) and the quantity of embedded secret data in each vertex of Horse model. We can see that different from the LSB scheme in [27] and the multi-layer scheme in [36] (which is the same in [26]), the PSNR in our approach (even in the worst case) oscillates

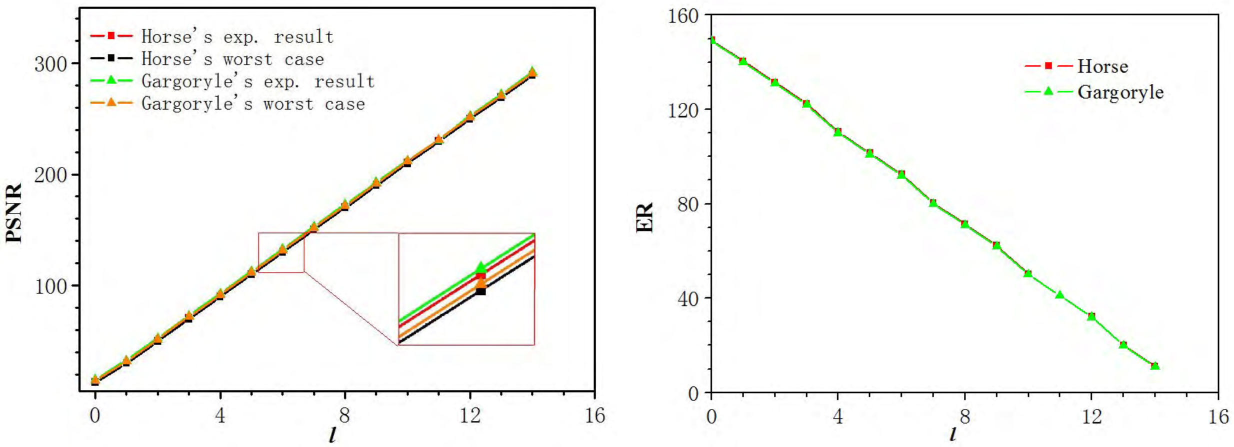


FIGURE 5. The relationship between *l* and the performances (PSNR and ER). PSNR grows with the increase of *l* (a) at the cost of certain ER (in the case of maximum embedding capacity with the corresponding *k*) (b).

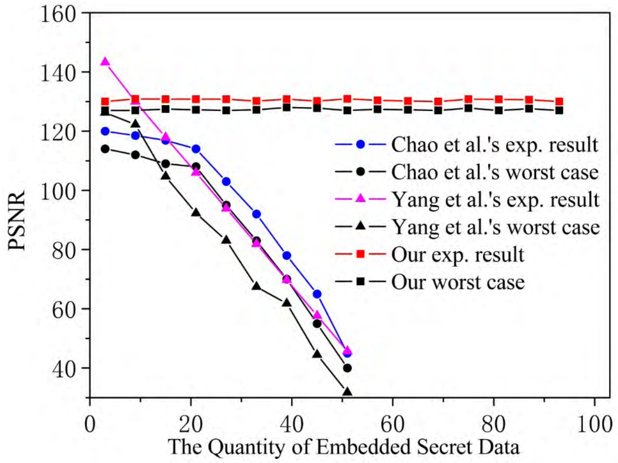


FIGURE 6. The comparison of our method (*l* = 6) with Chao *et al.*'s method [36] and Yang *et al.*'s [27] are shown in both the experiment results and the worst case.

weakly above a fairly high level. Because the modification of vertices is confined within the truncated space, the quality of the stego-model will always remain at a high level even approach the maximum embedding capacity (as listed in Table 3), and the distortion can be hardly detected by human eyes.

Fig. 7 shows the results of performances comparisons with the multi-layer steganography schemes (i.e., Chao *et al.*'s scheme [36] and Lin *et al.*'s scheme [26]) and the LSB in Yang *et al.*'s scheme [27], in terms of embedding rate and the PSNR for different stego-models. Different from the other steganography schemes, the distortion in our scheme is very stable. For a given length of truncation, the embedding rate ER has negligible effect on the distortion. We would like to mention that our scheme will produce less distortion if we set a larger length of truncation at the cost of losing certain capacity, but we still have the greatest embedding capacity at the same distortion case.

Moreover, we could also find that the performances have almost no effects associated with the quality of models due to the utility of the truncated space, take an example, the models with and without noise have almost the same embedding capacity and PSNR. As shown in Fig. 8, the performances are demonstrated in (a) Bunny (PSNR = 133.15, ER = 92.99), (b) Bunny with noise (PSNR = 133.14, ER = 92.99),

TABLE 3. Relationship between the embedding *k* and the PSNR on various models (*l* = 6).

Models (♯ vertices)	k							
	1	5	10	15	20	25	30	31
Horse(8431)	130.42	130.26	130.28	130.25	130.29	130.28	130.35	130.29
Venus(8268)	133.14	132.77	132.96	133.01	133.02	133.01	132.96	132.82
Elephant(24955)	129.09	128.77	128.90	128.58	128.83	128.96	128.83	128.80
Dinosaur(28287)	132.02	131.66	131.73	131.62	131.63	131.64	131.74	131.79
Armadillo(34594)	133.18	133.14	133.22	133.49	133.05	133.39	133.08	133.40
Bunny(35947)	133.67	133.24	133.12	133.38	133.11	133.14	133.15	133.15
Gargoyle(51335)	133.35	133.75	133.43	133.37	133.19	133.04	133.12	133.01
Hand(163657)	131.29	131.13	131.19	131.66	131.27	131.26	131.12	131.08
Dragon(4376455)	131.32	131.39	131.42	131.25	131.31	131.11	131.14	131.22

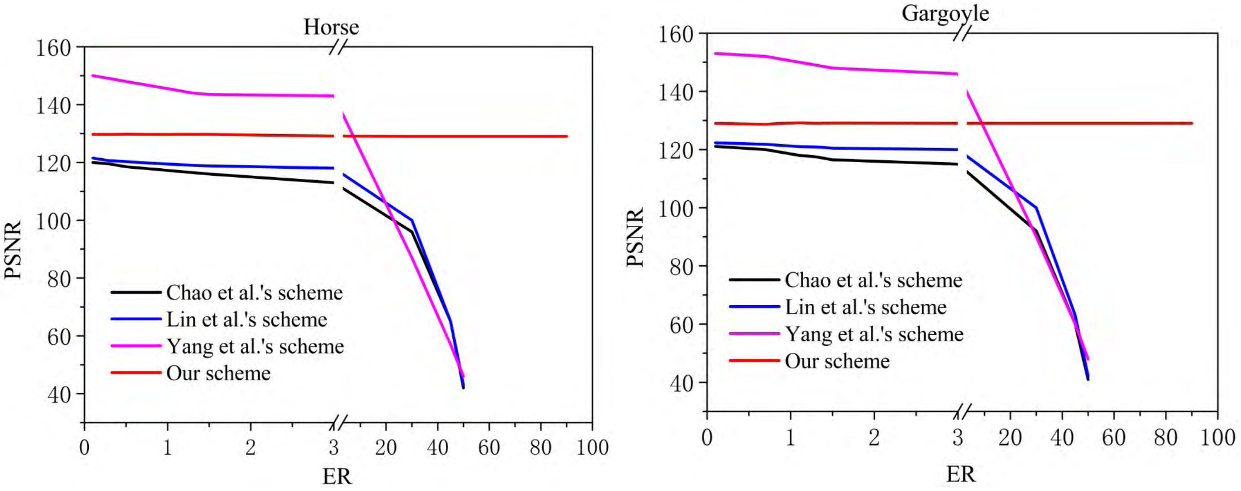


FIGURE 7. The comparisons with the steganography schemes: Chao *et al.*'s scheme [36], Lin *et al.*'s scheme [26] and Yang *et al.*'s [27], in terms of embedding rate and the PSNR for different models (Horse and Gargoyle, $l = 6$).

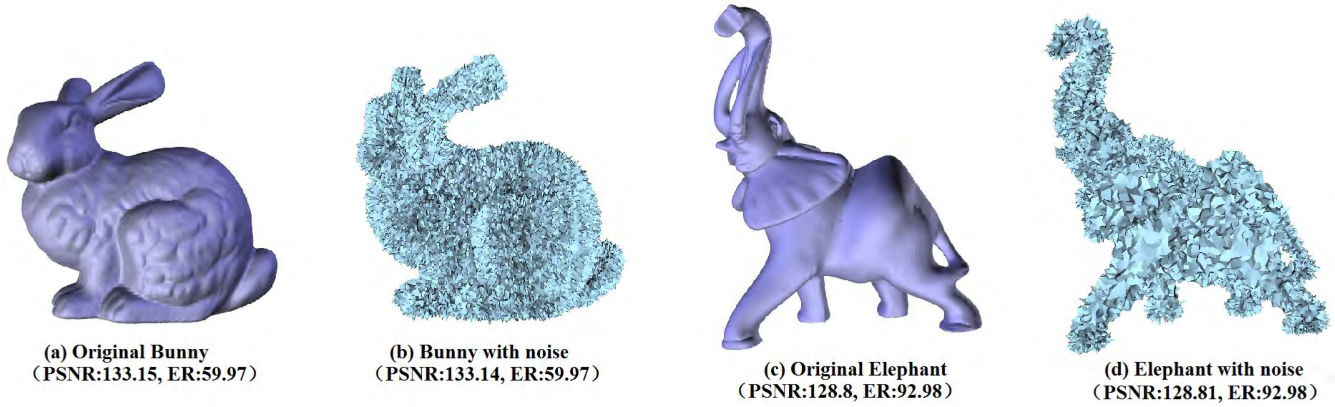


FIGURE 8. The embedding capacity of the proposed method is not affected by the quality of models. Both original models and the models with noise (100% mean edge length) are used to test the performances ($l = 6$, $k = 31$).

(c) Elephant (PSNR = 128.80, ER = 92.98), and (d) Elephant with noise (PSNR = 128.81, ER = 92.98), respectively.

C. COMPARISON AND DISCUSSION

We also compare our approach with five most related steganography approaches [34], [37], [36], [27], [26] in Table 4. We can see that our steganography approach provides a larger embedding capacity even than the multi-layer methods. Generally, the embedding capacity is over $90|V|$ with the default parameters ($l = 6, k = 31$), it could be larger when we set a shorter length of truncation, and the extreme capacity can approach $150|V|$ when we set $l = 0$. Moreover, the distortion (PSNR) in our method does not grow (decrease) with the increase of the quantity of embedded data, and it will be stable above a fairly low (high) threshold that can also be adjusted. The distortion in all the former steganography approaches grow with the increase of embedded quantity, and will be insufferable finally (see Fig. 6 and Fig. 7). Since we utilize a bounding box that is unchanged during the embedding processing to preprocess the models in both embedding and extraction procedures, our method can withstand similarity transform attacks, such as translation, rotation, and uniform scaling. Our steganography scheme is relying heavily on the segments of interval $[min\{x_t\}, max\{x_t\}]$ (y_t and z_t components are the same), which means that if the two ends of the component $min\{x_t\}$ and $max\{x_t\}$ can be properly found, we still have a chance to recover the information of the part that is not moved out of their corresponding intervals. Therefore, our approach might also suffer from some attacks in a rather minor way, such as vertex reordering, local smoothing and local noise.

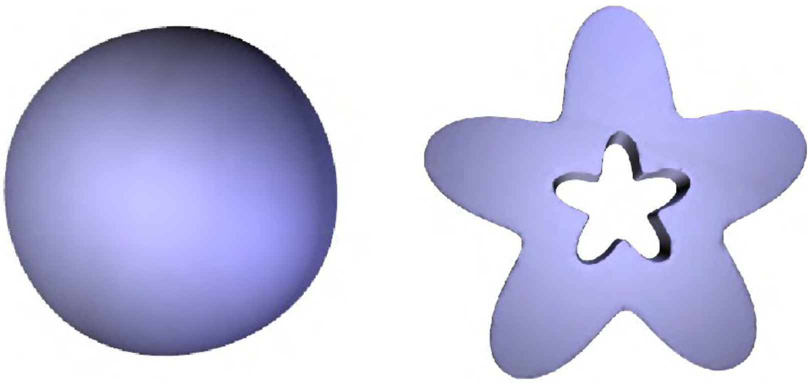
D. LIMITATIONS

Since we use the PCA to align the model in the preprocessing, just like many steganography approaches based on PCA [26], [36], failing to extract the embedding information because the PCA can not find the right alignment for models with the multi-axial symmetry. (see Fig. 9). In our future work, we improve our preprocess to make our method more robust. Also, we would like to extend approach to construct robust watermarking that can withstand more attacks.

TABLE 4. The comparison between our method ($l = 6, k = 31$) and the methods in [26], [27], [34], [36], and [37].

	[34]	[37]	[36]	[26]	[27]	Our method
Capacity (bits)	$\sim \lvert V\rvert$	$9\lvert V\rvert$	$3n_{layer}\lvert V\rvert$, $n_{layer} < 23$	$\alpha + 3n_{layer}\lvert V\rvert$, $n_{layer} < 23$	$\sim 70\lvert V\rvert$	$\sim 90\lvert V\rvert$
Distortion	incremental	incremental	incremental	incremental	adjustable	adjustable
Domain	spatial	spatial & Representation	spatial	spatial & Representation	spatial	spatial
Extraction	blind	blind	blind	blind	blind	blind
Noise	X	X	X	X	X	X
Smoothing	X	X	X	X	X	X
Vertex reordering	$\triangle$	X	$\triangle$	$\triangle$	$\triangle$	$\triangle$
Similarity transform	$\triangle$	$\triangle$	$\triangle$	$\triangle$	$\triangle$	$\surd$

[1] Note that n_{layer} represents the number of layers, α is constant in [26], normal degradation tolerances $\epsilon = 10$ in [27], symbols $'X'$, $'\surd'$ and $'\triangle'$ indicate that the approach can not withstand, can withstand, and can weakly withstand attacks, respectively.


(a)

(b)

FIGURE 9. Cases the proposed method failed to handle: the models with multi-axial symmetry models, such as (a) sphere and (b) star.

V. CONCLUSION

In this paper, we have proposed a novel steganography algorithm that utilizes a shifting strategy and a truncated space. The proposed method has the ability of adjusting the embedding distortion while retaining a high embedding capacity. Our method offers several salient improvements over the existing schemes: (1) The capacity of our steganography algorithm is higher than existing methods; (2) The embedding distortion could be adjusted within a specified threshold at the cost of certain embedding capacity; (3) The performances of the proposed method is stable (w.r.t. the shape of cover models) and robust (w.r.t. withstand similarity attacks); and (4) The steganography algorithm makes use of a simple function, and can be directly applied to point clouds and other representation of 3D models with point information. Our comprehensive experiments and extensive comparisons with other state-of-the-art methods have demonstrated that our method has high embedding capacity, retains an adjustable embedding distortion, affords low time complexity, and ensures adequate security.

REFERENCES

[1] 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.

[2] S. Katzenbeisser and F. A. Petitcolas, Eds., *Information Hiding Techniques for Steganography and Digital Watermarking*, 1st ed. Norwood, MA, USA: Artech House, 2000.

[3] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color, and gray-scale images," *IEEE Multimedia*, vol. 8, no. 4, pp. 22–28, Oct./Dec. 2001.

[4] W.-N. Lie and T. C.-I. Lin, "A feature-based classification technique for blind image steganalysis," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1007–1020, Dec. 2005.

[5] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[6] Y. Wang and P. Moulin, "Optimized feature extraction for learning-based image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 1, pp. 31–45, Mar. 2007.

[7] Z.-H. Wang, C.-F. Lee, and C.-Y. Chang, "Histogram-shifting-imitated reversible data hiding," *J. Syst. Softw.*, vol. 86, no. 2, pp. 315–323, Feb. 2013.

[8] P. Tsai, Y.-C. Hu, and H.-L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Process.*, vol. 89, no. 6, pp. 1129–1143, 2009.

[9] E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," in *Proc. SIGGRAPH*, 1999, pp. 49–56.

[10] 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.

[11] K. Yin, Z. Pan, J. Shi, and D. Zhang, "Robust mesh watermarking based on multiresolution processing," *Comput. Graph.*, vol. 25, no. 3, pp. 409–420, Jun. 2001.

[12] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, "A frequency-domain approach to watermarking 3D shapes," *Comput. Graph. Forum*, vol. 21, no. 3, pp. 373–382, Sep. 2002.

[13] S. Zafeiriou, A. Tefas, and I. Pitas, "Blind robust watermarking schemes for copyright protection of 3D mesh objects," *IEEE Trans. Vis. Comput. Graphics*, vol. 11, no. 5, pp. 596–607, Sep./Oct. 2005.

[14] V. R. Doncel, N. Nikolaidis, and I. Pitas, "An optimal detector structure for the Fourier descriptors domain watermarking of 2D vector graphics," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 5, pp. 851–863, Sep./Oct. 2007.

[15] Y. Yang, X. Sun, H. Yang, C. T. Li, and R. Xiao, "A contrast-sensitive reversible visible image watermarking technique," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 656–667, May 2009.

[16] Y. Yang and I. Ivrissimtzis, "Polygonal mesh watermarking using Laplacian coordinates," *Comput. Graph. Forum*, vol. 29, no. 5, pp. 1585–1593, Jul. 2010.

[17] R. Motwani, M. Motwani, and F. Harris, Jr., "An intelligent learning approach for information hiding in 3D multimedia," in *Proc. Int. Conf. Future Netw.*, Jan. 2010, pp. 447–451.

[18] K. Wang, G. Lavoué, F. Denis, A. Baskurt, and X. He, "A benchmark for 3D mesh watermarking," in *Proc. Shape Modeling Int. Conf.*, Jun. 2010, pp. 231–235.

[19] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE Security Privacy*, vol. 1, no. 3, pp. 32–44, May/Jun. 2003.

[20] P. Sallee, "Model-based methods for steganography and steganalysis," *Int. J. Image Graph.*, vol. 5, no. 1, pp. 167–189, 2005.

[21] H.-T. Wu and Y.-M. Cheung, "A reversible data hiding approach to mesh authentication," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, Sep. 2005, pp. 774–777.

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

[23] H. Huang, B. Liao, and J. Pan, "Special issue on information hiding and multimedia signal processing," *Int. J. Innov. Comput., Inf. Control*, vol. 6, no. 3, pp. 1207–1208, 2010.

[24] M. Luo and A. G. Bors, "Surface-preserving robust watermarking of 3-D shapes," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2813–2826, Oct. 2011.

[25] M.-T. Li, N.-C. Huang, and C.-M. Wang, "A novel high capacity 3D steganographic algorithm," *Int. J. Innov. Comput., Inf. Control*, vol. 7, no. 3, pp. 1055–1074, 2011.

[26] C.-H. Lin, M.-W. Chao, J.-Y. Chen, C.-W. Yu, and W.-Y. Hsu, "A high-capacity distortion-free information hiding algorithm for 3D polygon models," *Int. J. Innov. Comput., Inf. Control*, vol. 9, no. 3, pp. 1321–1335, Mar. 2013.

[27] Y. Yang, N. Peyerimhoff, and I. Ivrissimtzis, "Linear correlations between spatial and normal noise in triangle meshes," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 1, pp. 45–55, Jan. 2013.

[28] Y.-Y. Tsai, "An adaptive steganographic algorithm for 3D polygonal models using vertex decimation," *Multimedia Tools Appl.*, vol. 69, no. 3, pp. 859–876, Apr. 2014.

[29] Y. Yang and I. Ivrissimtzis, "Mesh discriminative features for 3D steganalysis," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 10, no. 3, pp. 27:1–27:13, Apr. 2014.

[30] H. Kaveh and M.-S. Moin, "A high-capacity and low-distortion 3D polygonal mesh steganography using surfacelet transform," *Secur. Commun. Netw.*, vol. 8, no. 2, pp. 159–167, Jan. 2015.

[31] Y.-Y. Tsai, "An efficient 3D information hiding algorithm based on sampling concepts," *Multimedia Tools Appl.*, vol. 75, no. 13, pp. 7891–7907, Jul. 2016.

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

[33] 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.

[34] 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.

[35] 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.

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

[37] 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, Sep. 2006.

NANNAN LI received the B.S. degree in computational mathematics from the Dalian University of Technology in 2010, where she is currently pursuing the Ph.D. degree with the School of Mathematical Sciences. Her research interests include computer graphics, differential geometry analysis, and machine learning.

JIANGBEI HU received the B.S. degree in applied mathematics from the Dalian University of Technology in 2016, where he is currently pursuing the Ph.D. degree with the School of Mathematical Sciences. His research interests include computer graphics, 3-D printing, and mesh processing.

RIMING SUN received the B.S. degree in computational mathematics from the Changchun University of Technology, the M.S. degree in applied mathematics from Jilin University, and the Ph.D. degree in computational mathematics from the Dalian University of Technology. She is currently a Lecturer with the College of Science, Dalian Jiaotong University. Her research topics include restoration of distorted document images, multiscale analysis, and compression methods of graphic and image.

SHENGFA WANG received the B.S. and Ph.D. degrees in computational mathematics from the Dalian University of Technology. He is currently an Associate Professor with the DUT-RU International School of Information and Software Engineering and the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology. His research interests include computer graphics, diffusion geometry, and differential geometry processing and analysis.

ZHONGXUAN LUO received the B.S. degree from Jilin University, and the M.S. and Ph.D. degrees from the Dalian University of Technology, all in computational mathematics. He is currently a Full Professor with the School of Mathematical Sciences and the Dean of the School of Software, Dalian University of Technology. His research interests are on computational geometry, computer graphics and image, and computer-aided geometric design.