Cognitive Computation (2022) 14:1172–1181
https://doi.org/10.1007/s12559-021-09919-5

Separable Reversible Data Hiding Based on Integer Mapping and MSB Prediction for Encrypted 3D Mesh Models

Na Xu[1] · Jin Tang[1] · Bin Luo[1] · Zhaoxia Yin[1]

Received: 30 September 2020 / Accepted: 17 July 2021 / Published online: 4 September 2021

Abstract
Reversible data hiding in encrypted domain (RDH-ED) technology can embed data into cover media without exposing the original content to third parties. In addition, the recipient can recover the cover media losslessly after extracting the embedded data. Image-based RDH-ED has been widely studied, but RDH-ED based on 3D meshes has obtained few research results due to the complex data structure and irregular geometric structure of 3D meshes. With the widespread application of 3D meshes, the research on 3D meshes has attracted extensive research from researchers in recent years. In this paper, we propose a reversible data hiding for encrypted 3D meshes based on integer mapping and most significant bit (MSB) prediction. The content owner divides all vertices into "embedded" sets and "reference" sets and then maps floating-point coordinates to integers. After calculating the MSB prediction error of the "embedded" sets, the encryption technology is performed. Then, additional data can be embedded through the MSB replacement strategy. According to different permissions, legal recipients can obtain the original meshes, the additional data or both of them by using the proposed separable method. Higher embedding capacity is achieved by adopting MSB embedding strategy, and perfect recovery of the original meshes is achieved by using ring prediction scheme. The experimental results show that the proposed method has greater embedding capacity compared with the state-of-the-art method.

Keywords Reversible data hiding · 3D mesh models · MSB prediction · Integer mapping · Privacy protection

Introduction

With the development of big data and cloud computing, the contradiction between privacy protection, big data and cloud computing has become increasingly prominent. How to realize the effective sharing and utilization of massive data [1] has become a hot research topic. In the past few decades, data hiding has been widely concerned by the research community. Data hiding embeds additional data by modifying the content of the cover medium, which are mainly divided into three categories: watermarking [2–4], steganography [5] and reversible data hiding (RDH) [6–10].

Reversible data hiding (RDH) can embed additional data in the cover media and recover the cover media losslessly after extracting the embedded data. The existing RDH methods are mainly based on three strategies: lossless compression [6], difference expansion [7, 8] and histogram shifting [9, 10]. In the lossless compression method, some features of the original image are extracted for lossless compression, and additional data are embedded in the reserved room. In order to improve the embedding capacity, RDH methods based on difference expansion and histogram shifting are proposed.

With the development of third-party cloud paradigms and privacy protection applications, the demand for privacy protection is growing. The combination of encryption and RDH plays a vital role in privacy protection. In order to store or share files securely using third-party services, content owner uses encryption method to convert the original content into unreadable ciphertext before transmission. The data hider then embeds the data in the ciphertext. At the same time, the recipient wants to recover the original content losslessly after decryption and data extraction. Such privacy protection schemes trigger RDH-ED to manage ciphertext data.

RDH-ED methods are mainly classified into two categories: vacating room after encryption (VRAE) [11–13] and

✉ Zhaoxia Yin
yinzhaoxia@ahu.edu.cn

[1] Anhui Province Key Laboratory of Multimodal Cognitive Computation, School of Computer Science and Technology, Hefei 230601, China

reserving room before encryption (RRBE) [14, 15]. Zhang et al. [11] first proposed the RDH-ED method. The data hider divides the encrypted images into non-overlapping blocks and embeds the data by flipping the three least significant bits (LSBs) of half of the pixels in each block. By using the spatial correlation of the images, the recipient designs a smoothness estimation function to estimate the texture complexity of each block for data extraction and images restoration. However, the quality of recovered images and the accuracy of data extraction are still not satisfactory. In order to separate data extraction and images recovery, Zhang et al. [12] proposed a separable RDH-ED method based on LSB compression. The recipient can restore the images without extracting the additional data, that is, the data can be extracted directly from the embedding space. Afterwards, Qian et al. [13] proposed that data hider can reserve room by compressing a series of selected bits obtained from encrypted images. The legitimate recipient uses the distributed source coding to correctly extract the additional data and restore the original image perfectly.

Compared with VRAE methods, RRBE methods have better performance in reducing data extraction errors and restoring original images. Ma et al. [14] first proposed to reserve room for data hiding before encryption. This method ensures that there are no errors in data extraction and image restoration. Recently, Puteaux et al. [15] proposed an RDH-ED method. The data hider uses MSB replacement to embed additional data. The recipient extracts additional data from the MSB plane of the encrypted image. After decryption, the recipient uses the correlation between adjacent pixels to reconstruct the original image through MSB prediction.

According to the state-of-the-art methods introduced above, image-based RDH has been extensively studied for many years, but these methods cannot be directly applied to other cover media, such as text, audio, video and 3D mesh. At present, 3D meshes have been applied in various fields. For example, in the medical field, 3D meshes are used to accurately describe organs. In the film industry, 3D meshes are used to represent characters, objects and scenes. Considering the commercial value, visual value and economic benefits, when the 3D meshes are distributed on the Internet, the producers or copyright owners inevitably face practical problems such as copyright protection and content authentication. Therefore, RDH based on 3D meshes is an important research topic. However, RDH research with 3D meshes as cover media is still in its infancy.

The existing RDH methods of 3D models are mainly divided into four domains: spatial domain, transform domain, compressed domain and encrypted domain. The spatial-domain-based RDH [16–20] embeds additional data into the 3D model by slightly modifying the vertex coordinates instead of modifying the connectivity data and has low complexity. The transform-domain-based RDH [21] embeds additional data in the transform coefficients of the model. The compressed-domain-based RDH [22, 23] uses vector quantization to compress the vertices of the 3D model and then embeds the data in the compressed model stream [24, 25]. In recent years, the encrypted-domain-based RDH [26, 27] has attracted the attention of research community.

The method of Jiang et al. [26] uses scaling and quantization to map the vertex coordinates of the 3D mesh to integers. The data hider embeds additional data by flipping several LSBs of the encrypted coordinates. The recipient uses the smoothing measure function to realize data extraction and mesh restoration. At this time, data extraction and mesh restoration are inseparable. In [27], a two-layer RDH-ED method using a homomorphic Paillier cryptosystem is proposed for 3D mesh, which is more suitable for cloud data management. Due to the large ciphertext extension and high computational complexity of the Paillier cryptosystem, the method in [27] is not efficient in practice. For the sake of fairness, the proposed method is mainly compared with [26] , because both [26] and the proposed method are based on symmetric encryption.

In this paper, we propose a 3D mesh-based RDH-ED method based on integer mapping and MSB prediction, which not only improves the embedding capacity, but ensures that the method is separable. The main contributions of this paper are as follows:

(1) The MSB embedding strategy is adopted to achieve higher embedding capacity.
(2) By making full use of the correlation of adjacent vertices in the natural mesh, the recipient can recover the MSB of the "embedded" vertices through ring prediction, so as to achieve mesh lossless recovery.
(3) The proposed method can directly extract additional data from the encrypted mesh and guarantees the data

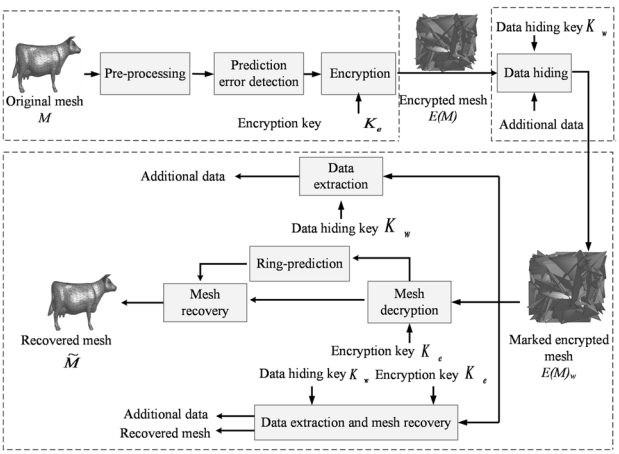


Fig. 1 Framework of the proposed method

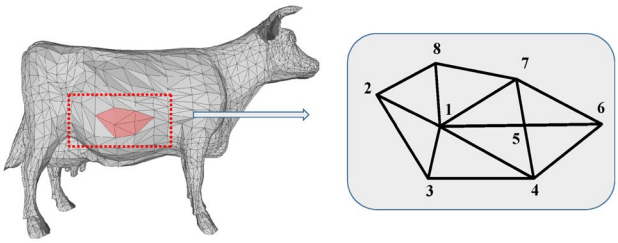


Fig. 2 Cow mesh

extraction is error-free and separable, which is of great significance for privacy protection.

The rest of this paper is organized as follows: Proposed Method introduces the pro-posed method. Experimental Results and Analysis presents the analysis of experimental results. Conclusion concludes this paper and describes the future work.

Proposed Method

The proposed method consists of three stages: 1) reserving room and encryption; 2) data hiding; 3) data extraction and mesh recovery. Figure 1 illustrates the framework of the proposed method. Pre-processing, prediction error detection, encryption, data hiding, data extraction and mesh recovery are elaborated in the following sub-sections.

Pre-processing

3D mesh models are represented in various file formats such as OFF, PLY, OBJ, etc. The 3D mesh is composed of vertices data and face data. Vertices data include coordinates data of vertices represented as $V = \{v_i \in \Re^3 | 1 \le i \le N\}$, where the vertex is represented as $v_i = (v_{i,x}, v_{i,y}, v_{i,z})$ and N is the number of vertices. Note that each coordinate $v_{i,j} < 1$ and $j \in \{x,\ y,\ z\}$. $F = (f_1, f_2, \ldots f_M)$ represent face sequence, where $f_i = (v_{i,x}, v_{i,y}, v_{i,z})$, M is the number of face. Figure 2 shows the local region of a "Cow" mesh, and Table 1 is its corresponding file format.

We can perform lossy compression of vertex coordinates according to the recommendation of [28]. According to the different precision m, the corresponding integer value is between -10^m and 10^m, where $m \in$ [1-33]. Normalizing floating point coordinates $v_{i,j}$ to integer coordinates $\bar{v}_{i,j}$ as

$$\bar{v}_{i,j} = \lfloor v_{i,j} \times 10^m \rfloor, \tag{1}$$

where i is the ith vertex, $j \in \{x, y, z\}$, $v_{i,j}$ is the original set of floating point vertices and $\bar{v}_{i,j}$ is the set of integer vertices. Recipient can convert the processed integer coordinates to floating point coordinates by Eq. (2).

$$\hat{v}_{i,j} = \bar{v}_{i,j} / 10^m. \tag{2}$$

The value of m corresponds to the bit-length l of integer coordinates as

$$l = \begin{cases} 8, & 1 \le m \le 2 \\ 16, & 3 \le m \le 4 \\ 32, & 5 \le m \le 9 \\ 64, & 10 \le m \le 33. \end{cases} \tag{3}$$

Prediction Error Detection

The "embedded" set s_e is used to embed additional data, and the "reference" set s_n is used to recover the mesh without modifying the vertices during the whole process. We traverse all the vertices contained in the face data in ascending order and assume that $F = (f_1, f_2 \ldots f_m)$ represents the face data sequence, where $f_i = (v_{i,x}, v_{i,y}, v_{i,z})$, M is the number of face data. Assuming that $f_n = (v_{n,x}, v_{n,y}, v_{n,z})$ is the next face sequence to be traversed, and both s_e and s_n are initially 0. If there is no vertex in f_n in s_e or s_n, we choose the first vertex in f_n to add $f_{n,x}$ to s_e and add $f_{n,y}$ and $f_{n,z}$ to s_n.

Table 1 File format for Fig. 2

Index of vertex	x-axis	y-axis	z-axis	coordinates	Index of face	Elements in each face
1	$v_{1,x}$	$v_{1,y}$	$v_{1,z}$	(0.180757, 0.034214, 0.193897)	1	(1, 2, 8)
2	$v_{2,x}$	$v_{2,y}$	$v_{2,z}$	(0.118210, 0.059086, 0.189724)	2	(7, 8, 1)
3	$v_{3,x}$	$v_{3,y}$	$v_{3,z}$	(0.092150, 0.029539, 0.197267)	3	(5, 7, 1)
4	$v_{4,x}$	$v_{4,y}$	$v_{4,z}$	(0.137215, 0.043615, 0.201492)	4	(1, 5, 4)
5	$v_{5,x}$	$v_{5,y}$	$v_{5,z}$	(0.136288, 0.065522, 0.187564)	5	(3, 4, 1)
6	$v_{6,x}$	$v_{6,y}$	$v_{6,z}$	(0.160892, 0.015154, 0.200969)	6	(2, 1, 3)
…	…	…	…	…	…	…
20	$v_{20,x}$	$v_{20,y}$	$v_{20,z}$	(0.082017, 0.026986, 0.253443)	20	(20, 19, 133)
21	$v_{21,x}$	$v_{21,y}$	$v_{21,z}$	(0.026661, 0.037672, 0.246828)	21	(21, 20, 343)
…	…	…	…	…	…	…

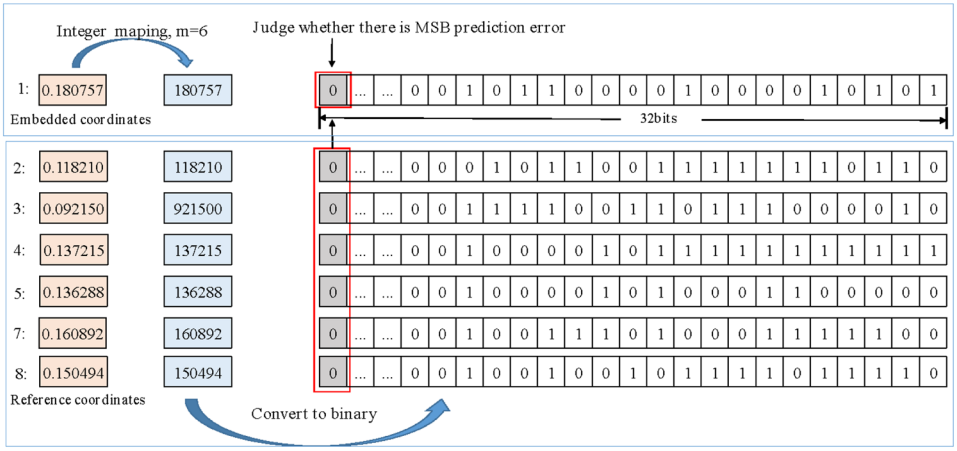


Fig. 3 An example of prediction error detection test on cow mesh

As shown in Fig. 3, the MSB of the x-coordinate of the "embedded" vertex numbered 1 is 0. The sender counts the number of occurrences of 0 or 1 in the MSB of the coordinates of the "reference" vertices numbered 2, 3, 4, 5, 7 and 8. If the number of 0 is greater than or equal to the number of 1, the MSB of the vertex coordinate numbered 1 is predicted to be 0. Vertex 1 is called a vertex in the embedding set that does not have prediction errors. Otherwise, the vertex index information is recorded as auxiliary information. Finally, the sender sends the auxiliary information together with the original mesh to the data hider.

Encryption

After the vertex coordinates are pre-processed, the sender uses Eq. (4) convert integer coordinates to binary.

$$b_{i,j,u} = \lfloor \bar{v}_{i,j}/2^u \rfloor \mod 2, \qquad u = 0, 1 \ldots bitlen - 1, \tag{4}$$

where $\lfloor . \rfloor$ is a floor function and $1 \leq i \leq N$ and $j j \in \{x, y, z\}$ the bitlen of the coordinate can be obtained by Eq. (3).

The sender uses the stream cipher function to generate pseudo-random bits $c_{i,j,u}$ and encrypts the original 3D mesh bit stream $b_{i,j,u}$ to obtain the encrypted binary $e_{i,j,u}$.

$$e_{i,j,u} = b_{i,j,u} \oplus c_{i,j,u}, \tag{5}$$

where $\oplus$ stands for exclusive OR.

The sender can get the encrypted mesh using Eq. (6)

$$E_{i,j} = \sum_{u=0}^{bitlen-1} e_{i,j,u} \times 10^m, \tag{6}$$

where $E_{i,j}$ is the integral value of coordinates.

Data Embedding

To prevent additional data from being detected, the data hiding key Kw is used to encrypt the to-be-inserted data. The sender first calculates s_e and then embeds the data in the vertices in s_e where there is no prediction error. The MSB of the x, y and z coordinate values of each vertex is replaced with 1 bit. With Eq. (7), each vertex in s_e is embedded with 3 bits.

$$v_{i,j}'' = w \times 2^{bitlen-1} + v_{i,j}' \mod 2^{bitlen-2}, \tag{7}$$

where w is additional data, $v_{i,j}' \in C$ is the vertex after pre-processing and encryption, $v_{i,j}''$ is the vertex of marked encrypted mesh.

After the data embedding stage, the sender obtains encrypted mesh with additional data, namely E(M)w. Figure 4 shows the embedding process on vertex 1 of the "embedded" set. Assuming that the x-axis coordinate value is 0.180757, when m = 6, after pre-processing, it is mapped to an integer 180757. The data hider directly replaces the MSB of vertex 1 with additional data 1 by bit replacement strategy. After completing the above steps, the data embedding process is completed.

Data Extraction and Mesh Recovery

The recipient uses the data hiding key Kw to extract additional data and uses the encryption key Ke to restore the original mesh respectively. Since the proposed method is

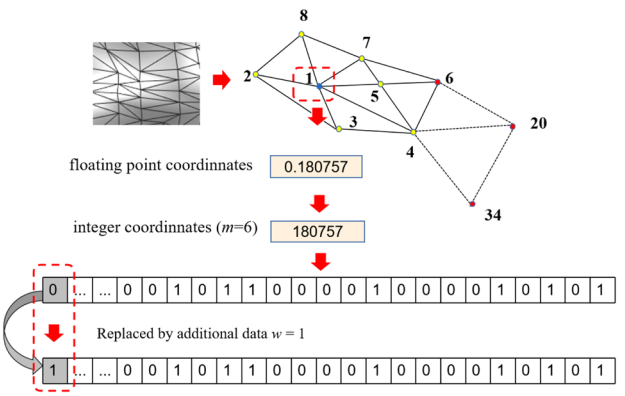
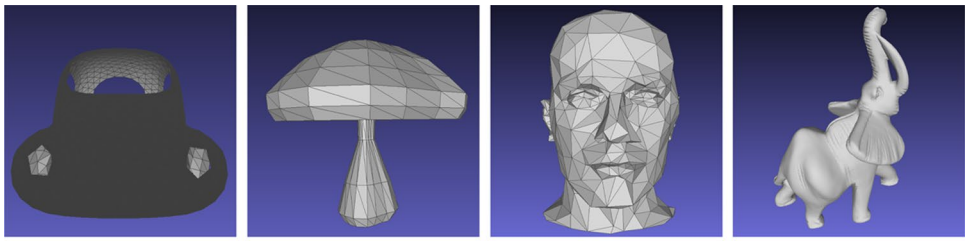


Fig. 4 An example of data embedding on cow mesh



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

Fig. 5 Test meshes: (**a**) Beetle, (**b**) Mushroom, (**c**) Mannequin, (**d**) Elephant

separable, there are the following three situations according to the different situations:

Case 1: With only the data hiding key Kw, the recipient can extract the MSB from the vertex coordinates of s_e without prediction error and then obtain the corresponding plaintext additional data.

$$w = v_{i,j}''/2^{bitlen-1}, \tag{8}$$

where $v_{i,j}'' \in C$ is vertex of the marked encrypted mesh.

Case 2: With only the encryption key Ke, the recipient can recover E(M)w to get M. M is recovered in two steps : mesh decryption and MSB prediction recovery.

The pseudorandom bits $c_{i,j,u}$ are generated by the encryption key Ke and used to perform xor function with $e_{i,j,u}''$ to decrypt the marked encrypted mesh E(M)w.

$$b_{i,j,u}'' = e_{i,j,u}'' \oplus c_{i,j,u}, \tag{9}$$

where $e_{i,j,u}''$ is the binary stream of the marked encrypted mesh, $b_{i,j,u}''$ is the binary stream of the decrypted mesh with additional data and u=0, 1...bitlen-1.

After decryption, the vertex coordinates of the s_n set is restored. In the data embedding stage, the MSB of the coordinates of the vertices embedded in the set is replaced by additional data. Therefore, the recipient uses the spatial correlation of the original mesh, and the MSB of the "embedded" set vertices is predicted by the MSB of the surrounding adjacent vertices. The method of using adjacent reference coordinates to predict the embedded vertex coordinates is called ring prediction. The recipient can obtain a high-quality restoration mesh by using ring prediction.

For example, the coordinate values of adjacent vertices 2, 3, 4, 5, 7 and 8 have been restored correctly after decryption. Based on their MSB values, the coordinate value of the vertex number 1 is predicted to be 0 or 1. When predicting the MSB of $v_{1,x}$, we count the MSB of the x coordinate of vertex index numbers 2, 3, 4, 5, 7 and 8. If the number of occurrences of MSB 0 is greater than or equal to the number of occurrences of 1, the MSB of $v_{1,x}$ is expected to be 0, otherwise it is 1.

Case 3: With the data hiding key Kw and encryption key Ke at the same time, the recipient can extract additional data and restore the original 3D mesh perfectly. Note that data extraction step needs to be performed before mesh restoration.

Experimental Results and Analysis

In this section, the reversibility and embedding capacity of the improved method are analyzed, and the results are compared with the state-of-the-art method [26]. We perform extensive experiments in MATLAB R2018b under windows 10. As shown in Fig. 5, there are four standard test meshes: Beetle, Mushroom, Mannequin, Elephant. Two datasets: meshes with OFF format from The Princeton Shape Retrieval and Analysis Group[1] and those in OBJ format from The Stanford 3D Scanning Repository[2] are used to test performance. The key indicator is the embedding capacity. In Embedding Capacity, we analyze the embedding capacity of the proposed method. In Geometric and Visual Quality, for the distortion of the original mesh caused by the data hider, the Hausdorff distance and the signal-to-noise ratio (SNR) are used to evaluate the reversibility. In Performance Comparison, the performance comparison of the proposed method and the state-of-the-art method [26] is given. The additional data embedded is a randomly generated 0/1 sequence.

Embedding Capacity

The embedding rate (ER) is measured by the number of bits per vertex (bpv), which is the ratio of the number of embedded bits to the number of vertices in the mesh.

In fact, in the clear areas, MSB predictions are easier than LSB predictions. In this paper, the MSB of each coordinate axis is replaced with 1-bit additional data, and as a result, 3 bits are embedded per vertex. We test the embedding rate of the proposed method on four standard test meshes. The embedding rate of the proposed method on Beetle, Mushroom, Mannequin and Elephant is 0.98 bpv, 1.34 bpv, 0.95 bpv and 1.02 bpv, respectively.

[1] http://shape.cs.princeton.edu/benchmark/index.cgi

[2] http://graphics.stanford.edu/data/3Dscanrep/

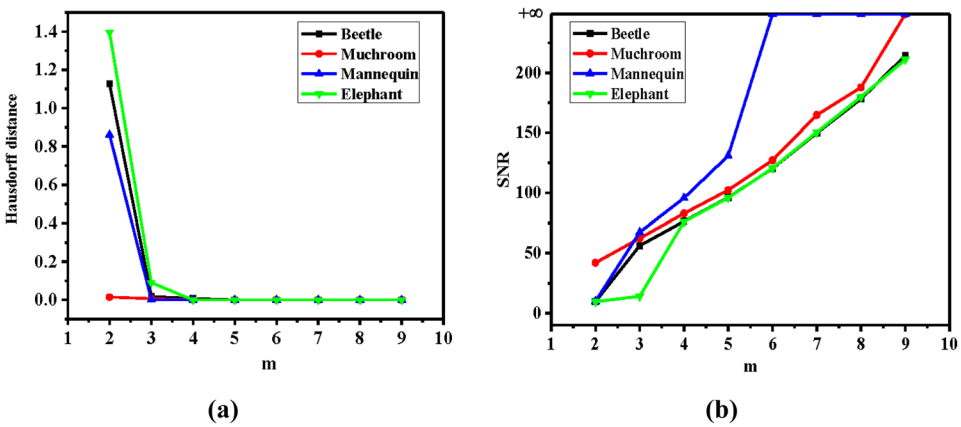


Fig. 6 Results of test meshes on different m: (**a**) Hausdorff distance, (**b**) SNR

Geometric and Visual Quality

Hausdorff distance and signal-to-noise ratio (SNR) are used to measure the geometric distortion of the mesh. Hausdorff distance measures the similarity between two sets of points by calculating the distance between two sets of points. Assuming there are two sets A=(a_1,a_2...a_p) and B=(b_1,b_2...b_q), the Hausdorff distance between two sets of points is defined as:

$$H(A,B) = max(h(A,B), h(B,A)), \tag{10}$$

$$h(A,B) = max(a \in A) min(b \in B) \parallel a - b \parallel, \tag{11}$$

$$h(B,A) = max(b \in B) min(a \in A) \parallel b - a \parallel, \tag{12}$$

where $\parallel . \parallel$ is the distance between point a of set A and point b of set B (such as L2), p and q are the number of elements in the set.

Signal-to-noise ratio (SNR) is defined as: SNR=

$$10 \times \lg \frac{\sum_{i=1}^{N}[(v_{i,x} - \bar{v}_x)^2 + (v_{i,y} - \bar{v}_y)^2 + (v_{i,z} - \bar{v}_z)^2]}{\sum_{i=1}^{N}[(g_{i,x} - \bar{v}_x)^2 + (g_{i,y} - \bar{v}_y)^2 + (g_{i,z} - \bar{v}_z)^2]}, \tag{13}$$

where $\bar{v}_x$, $\bar{v}_y$, $\bar{v}_z$ are the averages of the mesh coordinates, $v_{i,x}$, $v_{i,y}$, $v_{i,z}$ are the original coordinates, $g_{i,x}$, $g_{i,y}$, $g_{i,z}$ are the modified mesh coordinates, N is the number of vertices.

The value of m is a trade-off between the quality of the recovered mesh and the computational overhead of the process.

As shown in Fig. 6(a), when 2≤m≤4, the Hausdorff distance gradually decreases, and when m≥4, the Hausdorff distance steadily approaches 0. The results show that as the accuracy m increases, the similarity of the point set between the recovered mesh and the original mesh increases. As shown in Fig. 6(b), SNR shows an upward trend as m increases. Thus, as m increases, the Hausdorff distance decreases, while the SNR increases. This indicates that the quality of the recovered mesh is increasing.

Figure 7 shows the visual effect of the original mesh at different stages of the proposed method when m = 4, including original mesh, encrypted mesh, marked encrypted mesh and recovered mesh.The difference between the original mesh and the recovered mesh is invisible to the naked eye, which means that the proposed method does not introduce perceptual distortion.

Performance Comparison

The data hiding method in [26] flips the LSBs of each vertex to embed 1 bit data. Due to the spatial correlation of the mesh, the original mesh local region is much smoother than the modified mesh local region, so the recipient uses a smoothness estimation function to estimate the fluctuation

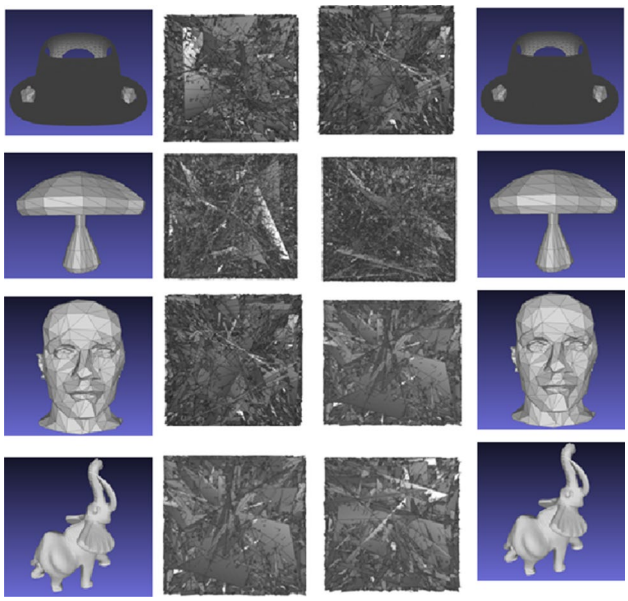


Fig. 7 Illustrative examples showing the appearance of the mesh of different stages when m = 4. From left to right is the original mesh, encrypted mesh, marked encrypted mesh and recovered mesh

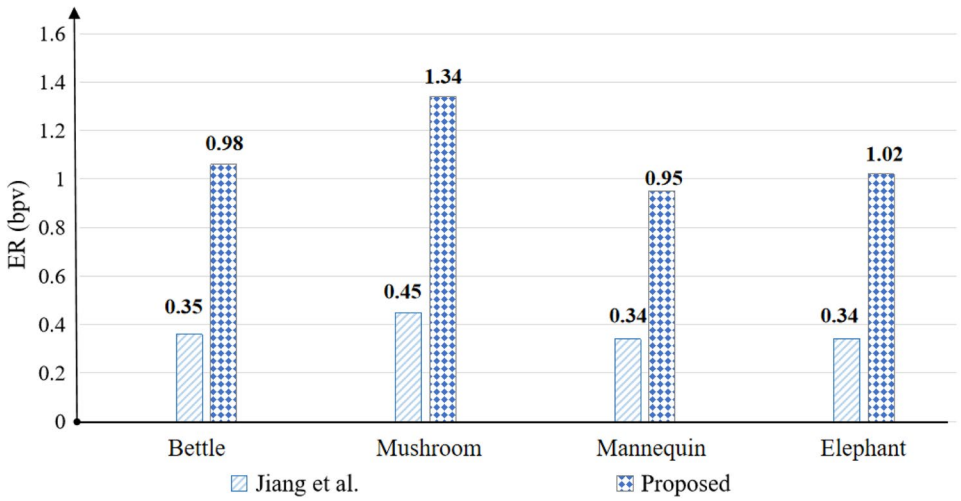


Fig. 8 Comparison of embedding rate between the proposed method and Jiang et al.'s method

of each local region for data extraction and image restoration. As shown in Fig. 8, the embedding rate of Jiang et al.'s method [26] on Beetle, Mushroom, Mannequin and Elephant is 0.35 bpv, 0.45 bpv, 0.34 bpv and 0.34 bpv, respectively. The embedding rate of the proposed method is 0.98 bpv and 1.34 bpv, 0.95 bpv and 1.02 bpv. The experimental results show that the proposed method improves the embedding capacity compared with the method of Jiang et al. [26].

We tested the performance of embedding rate on the Princeton Shape Retrieval and Analysis Group dataset to reduce the impact of randomly selecting test meshes. As shown in Table 2, the embedding rate of Jiang et al. is 0.35 bpv, while the average embedding rate of the proposed method is 1.02 bpv. Thus, the proposed method has significant advantages in embedding rate compared with the method proposed by Jiang et al. [26].

The Hausdorff distance and signal-to-noise ratio (SNR) are used to measure the geometric distortion of the meshes. Lower Hausdorff distance values and higher SNR values indicate that the quality of the recovered mesh is better. Figure 9 shows the experimental results of the Hausdorff distance and SNR on four test meshes. Taking the Beetle as an example, the ER of the proposed method is 0.98 bpv, while the method of Jiang et al. [26] is 0.35 bpv. When m = 4, the Hausdorff distance of this method is 0.008 (10^{-3}), while the method of Jiang et al. is 0.990 (10^{-3}). The SNR of this method is 76.37, while the method of Jiang et al. is 43.06. Thus, the proposed method not only obtains a higher embedding rate, but also has good performance in obtaining high-quality recovered meshes compared with the method [26].

Table 2 Average embedding rate comparison with Jiang et al.'s method on datasets

Datasets	Methods	Average ER
The Princeton Shape Retrieval and Analysis Group	Jiang et al. [26]	0.35bpv
	Proposed	1.02bpv

Feature Comparison

The method of Jiang et al. [26] uses a smoothing estimation function to calculate the local smoothness of the embedded data and the local smoothness of the unmodified part. Since data extraction and model restoration are performed at the same time, this method is inseparable. When applied to the cloud management, it means that if the cloud administrator wants to extract the additional data embedded in the embedding stage, it must have the decryption key to decrypt the marked encrypted mesh first, which may expose sensitive information of the content owner. This method is suitable when the content owner fully trusts the third-party platform, and there are certain application limitations. The proposed method can directly extract the additional data embedded in the cloud from the ciphertext and does not involve the decryption of the mesh. Therefore, as shown in Table 3, the proposed method is separable and more suitable for cloud management.

In fact, the data extraction error rate of the method in [26] on the Beetle mesh is 36.78%, and that of Mushroom, Mannequin and Elephant are 32.08%, 45.21% and 4.94%, respectively.

Table 3 Feature comparison between the proposed method and Jiang et al. [26]

Methods	Features		
	Separable	Error-free in data extraction	Error-free in mesh recovery
Jiang et al. [26]	×	×	×
Proposed	√	√	√

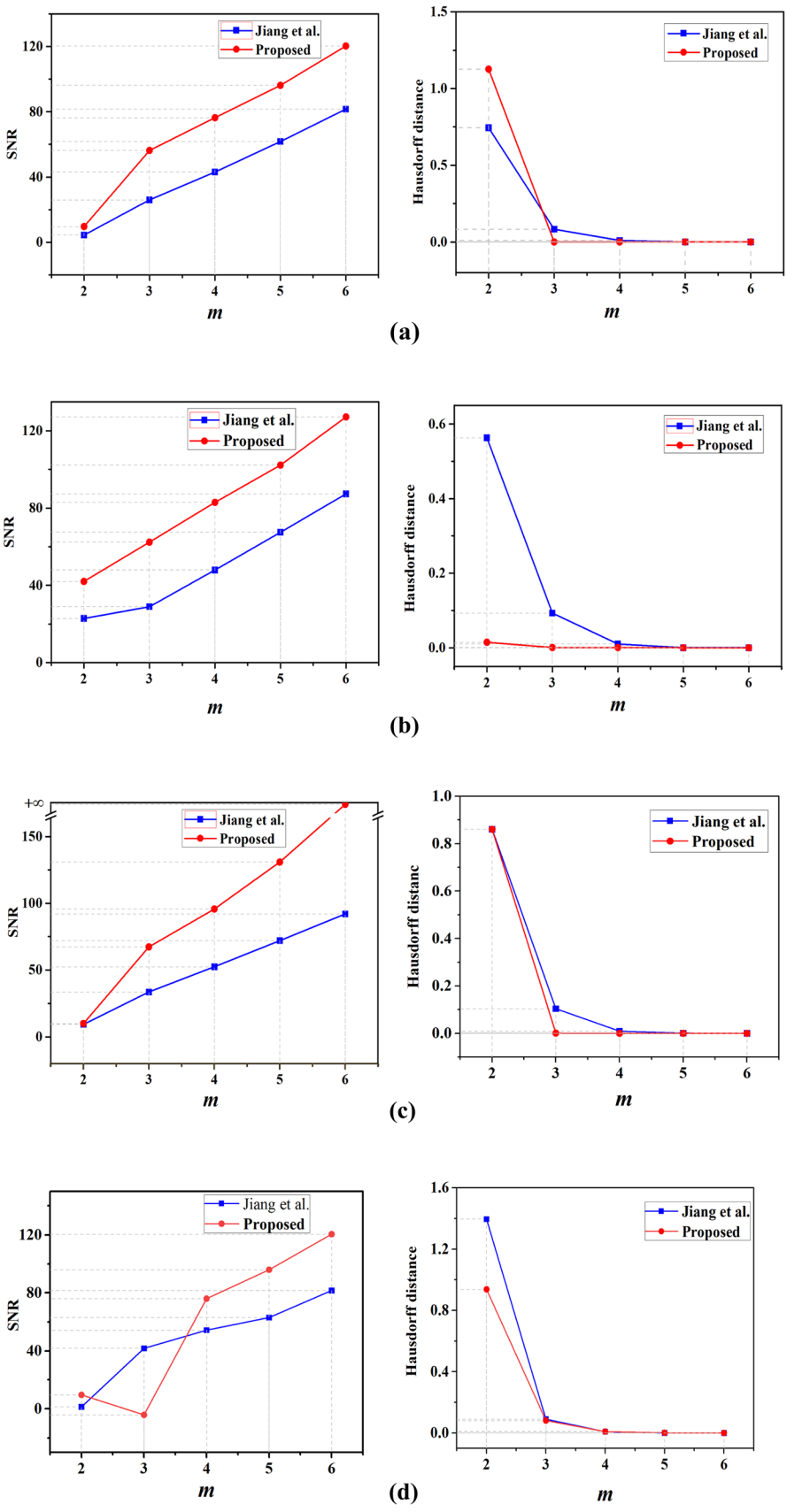


Fig. 9 Comparison results of Hausdorff distance and SNR on test meshes: (**a**) Beetle, (**b**) Mushroom, (**c**) Mannequin, (**d**) Elephant

Table 4 Performance of reversible data hiding on dense meshes

Meshes	Number of vertices	Number of faces	ER (bpv)	Hausdorff distance (10^{-5})	SNR	Error-free in data extraction
Dragon	871414	437645	1.04	0.73	143.23	√
Armadillo	172974	345944	1.06	0.08	161.14	√
Happyvrip	543652	1087716	1.07	0.18	143.56	√

Larger data extraction error rate indicates that it is possible to transmit inaccurate additional data and cause invalid communication. In contrast, our method can directly and correctly extract additional information from the ciphertext domain to achieve effective communication. This method [26] is not achieve reversibility. The proposed method controls the degree of distortion by adjusting parameter m values and combines ring prediction to obtain perfect restoration mesh, that is, reversibility is achieved.

Performance Analysis on Dense Meshes

In practical applications, models between different formats are often formatted. The proposed method is designed for the mesh in .OFF format. But after modifying the model reading function, the proposed method can directly use the mesh in the .PLY format as a carrier. In order to verify the effectiveness of the method, we performed experiments on The Stanford 3D Scanning Repository data set. Three dense meshes in .PLY format are randomly selected from the data set to show performance. The embedding rate and distortion performance of Table 4 show that the proposed method also achieves a higher embedding rate on dense meshes. Thus, experiments on dense meshes show the applicability and effectiveness of the proposed method to dense meshes.

Conclusion

In this paper, we propose a RRBE separable RDH-ED method for encrypted 3D mesh models based on integer mapping and MSB prediction. The proposed method not only achieves feasibility, but also emphasizes the balance between capacity and distortion. The data hider achieves larger embedding capacity through the MSB embedding strategy. The recipient gets higher quality recovered mesh using ring prediction. At the same time, data extraction and mesh restoration in the proposed method are separable and error free. Experiments show that our method has larger embedding capacity and higher quality recovered mesh compared with the state-of-the-art methods. Since the selection of "embedded" sets is limited by the connectivity of the mesh, the embedding capacity of the proposed method is not very ideal. Designing a more effective method to select the "embedded" set to improve the embedding capacity is a problem to be solved in future work.

Funding Information This research work is partly supported by National Natural Science Foundation of China (61872003, 61860206004).

Declarations

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed Consent Informed consent was not required as no human or animals were involved.

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Dashtipour K, Gogate M, Cambria E, et al. A novel context-aware multimodal framework for Persian sentiment analysis[J]. Neurocomputing. 2021.
2. Ren Jinchang, Zhao Huimin. Cognitive Computation of Compreesed Sensing for Watermark Signal Measurement[J]. Cogn Comput. 2016;8:246–60.
3. Behrouz BH, Taherinia AH, et al. An Effective Semi-fragile Watermarking Method for Image Authentication Based on Lifting Wavelet Transform and Feed-Forward Neural Network[J]. Cogn Comput. 2020;12(2):863–90.
4. Gao X, Deng C, Li X, et al. Local Feature Based Geometric-Resistant Image Information Hiding[J]. Cogn Comput. 2010;2(2):68–77.
5. Sachnev V, Savitha R, Suresh S, et al. A Cognitive Ensemble of Extreme Learning Machines for Steganalysis Based on Risk-Sensitive Hinge Loss Function[J]. Cogn Comput. 2015;7(1):103–10.
6. Celik MU, Sharma G, Tekalp AM, Saber E. Lossless generalized-lsb data embedding. IEEE Trans Image Process. 2005;14(2):253-266.
7. Tian J. Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol. 2003;13(8):890–6.
8. Yongjian H, Lee HK, Chen K, Li J. Difference expansion based reversible data hiding using two embedding directions. IEEE Trans Multimedia. 2008;10(8):1500–12.
9. Li X, Zhang W, Gui X, Yang B. Efficient reversible data hiding based on multiple histograms modification. IEEE Trans Inf Forensics Secur. 2015;10(9):2016–27.
10. Wang J, Ni J, Zhang X, Shi Y. Rate and distortion optimization for reversible data hiding using multiple histogram shifting. IEEE Trans Cybern. 2016;47(2):315–26.

11. Zhang X. Reversible data hiding in encrypted image. Signal Process Lett IEEE. 2011;18(4):255–8.
12. Zhang X. Separable reversible data hiding in encrypted image. IEEE Trans Inf Forensics Secur. 2011;7(2):826–32.
13. Qian Z, Zhang X. Reversible data hiding in encrypted images with distributed source encoding. IEEE Trans Circuits Syst Video Technol. 2015;26(4):636–46.
14. Ma K, Zhang W, Zhao X, Nenghai Y, Li F. Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans Inf Forensics Secur. 2013;8(3):553–62.
15. Puteaux Pauline, Puech William. An efficient msb prediction-based method for high-capacity reversible data hiding in encrypted images. IEEE Trans Inf Forensics Secur. 2018;13(7):1670–81.
16. Wu H, Cheung Y. A reversible data hiding approach to mesh authentication. In IEEE/WIC/ACM International Conference on Web Intelligence, Compiegne, France. 2005.
17. Chuang CH, Cheng CW, Yen ZY. Reversible data hiding with affine invariance for 3d models. In IET International Conference on Frontier Computing Theory. 2010. p 77–81.
18. Tsai YY. A distortion-free data hiding scheme for triangular meshes based on recursive subdivision. Adv Multimed. 2016;2016:1–10.
19. Jiang R, Zhang W, Hou D, Wang H, Nenghai Y. Reversible data hiding for 3d mesh models with three-dimensional prediction-error histogram modification. Multimed Tools Appl. 2018;77(5):5263–80.
20. Zhang Q, Song X, Tao W, Fu C. Reversible data hiding for 3d mesh models with hybrid prediction and multilayer strategy. Multimed Tools Appl. 2018;1–17.
21. Luo H, Lu ZM, Pan JS. A reversible data hiding scheme for 3d point cloud model. In IEEE International Symposium on Signal Processing and Information Technology, Vancouver, BC, Canada. 2006. p 863–67.
22. Sun Z, Lu ZM, Li Z. Reversible data hiding for 3d meshes in the pvq-compressed domain. In International Conference on Intelligent Information Hiding and Multimedia. Vancouver, BC, Canada. 2006. p 593–596.
23. Lee H, Dikici Ca, Lavoue G, Dupont Fl. Joint reversible watermarking and progressive compression of 3d meshes. Vis Comput. 2011;27(6–8):781–92.
24. Li L, Zhu L, Zakharchenko V, et al. Advanced 3D motion prediction for video based dynamic point cloud compression. IEEE Trans Image Process. 2019;29(99):289–302.
25. Li L, Li Z, Liu S, Li H. Rate control for video-based point cloud compression. IEEE Trans Image Process. 2020;(99):1–1.
26. Jiang R, Zhou H, Zhang W, Nenghai Y. Reversible data hiding in encrypted three-dimensional mesh models. IEEE Trans Multimed. 2018;20(1):55–67.
27. Shah M, Zhang W, Honggang H, Zhou H, Mahmood T. Homomorphic encryption-based reversible data hiding for 3d mesh models. Arab J Sci Eng. 2018;43(12):8145–57.
28. Deering M. Geometry compression. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. New York, United States. 1995. p 13–20.