A Multi-carrier Information Hiding Algorithm Based on Layered Compression of 3D Point Cloud Model

Shuai Ren, Yuxiao Li(✉), Bo Li, Hao Gong, and Qiuyu Feng

School of Information Engineering, Chang'an University, Xi'an 710064, China
2022224083@chd.edu.cn

Abstract. Aiming at the problem that most of the information embedding carriers of existing information hiding technologies are two-dimensional images, and most of them are single carriers with limited embedding information capacity, a multi-carrier information hiding algorithm based on hierarchical compression of 3D point cloud model is proposed. Firstly, the 3D model is pre-standardized, the minimum bounding box of the model is generated, and the slice of the model is stratified. Secondly, the ratio of projected area between each layer region and the minimum bounding box is used as the weight of each layer region and the model feature vector, and the carrier set is classified by calculating the similarity of each model feature vector. Finally, each layer area is compressed according to the secret information to complete the secret information hiding. The experimental results show that, compared with the comparison algorithm, the proposed algorithm is robust while maintaining invisibility in the face of a single attack.

Keywords: Information hiding · Multi carrier · Carrier classification · Hierarchical compression · Feature extraction

1 Introduction

With the continuous development of 3D modeling technology, more and more application scenarios need to embed information in 3D models, such as copyright protection, digital watermarking, data hiding, etc. Multi-carrier 3D model information hiding algorithm is a technique that embeds secret information among multiple 3D models. Compared with traditional single-carrier information hiding technology, multi-carrier information hiding technology has higher security and robustness [6,15,19].

At present, the research of information hiding algorithm based on point cloud 3D model has made remarkable progress [3,8,18]. Luo et al. proposed a reversible information hiding algorithm based on a 3D point cloud model, which embedded data by creating a cluster of 8 adjacent vertices and using the high correlation between adjacent vertices [10]. Liu et al. proposed a 3D point cloud model watermarking algorithm based on ring distribution. In this algorithm, vertices whose

S. Goel and P. R. Nunes de Souza (Eds.): ICDF2C 2023, LNICST 570, pp. 132–145, 2024.
https://doi.org/10.1007/978-3-031-56580-9_8

average curvature is less than zero are selected as the feature vertices embedded in the watermark, and the remaining vertices are used to establish an invariant space. The 3D model is divided into several spheres and rings according to the number of watermark bits to resist geometric attacks. The purpose of information hiding is achieved by modifying the radial distance of feature vertices in different spheres and rings [9].

In order to improve the transmission efficiency of the model, it is usually necessary to compress the carrier. Kammerl et al. proposed a lossy compression algorithm for point cloud flow. The algorithm firstly performs spatial decomposition through octree data structure, then encodes structural differences of continuous point cloud models, and finally realizes point cloud compression by using spatial and temporal redundancy in point cloud data [5]. He et al. proposed a point cloud compression algorithm based on spherical projection, which carried out adaptive partitioning of the original point cloud, established a fitting sphere in each block, converted the 3D point cloud into a set of depth images through spherical coordinate transformation and spherical projection, and decomposed the depth images into occupancy images and attribute vectors for compression [4].

This chapter proposes a multi-carrier information hiding algorithm based on layered compression of 3D point cloud model, aiming at the limitations of capacity and security of single-carrier information hiding algorithm and point cloud compression algorithm. Firstly, the 3D model is pre-standardized, the minimum bounding box of the model is generated [1], and the slice of the model is stratified. Secondly, the ratio of the projected area between each layer region and the minimum bounding box is used as the weight of each layer region and the model feature vector, and the carrier set is classified by calculating the similarity of the feature vector of each model. Finally, the model is matched with the secret information by weighting index of each layer, and the secret information is hidden by compression of each layer.

2 Basic Principle of Algorithm

2.1 Carrier Model Normalization Preprocessing

In order to avoid the influence of the position, direction and size of the 3D model on the feature extraction, the model needs to be pre-processed by translation, normalization and rotation. Suppose the model has n vertices, and the set of vertices $V = \{v_1, v_2, ..., v_n\}$. The specific steps are as follows:

Step 1: Generate the minimum bounding box of the model. Suppose that the coordinates of any vertex of the model $v_i (i \in 1, 2, ..., n)$ in 3D space are (x_i, y_i, z_i), the maximum boundary point v_M and the minimum boundary point v_m are found according to formula (1) and formula (2) respectively. Then, m boundary points v_b are found by formula (3).

$$v_M = (x_M, y_M, z_M) = (\max(x_i), \max(y_i), \max(z_i)) \tag{1}$$

$$v_m = (x_m, y_m, z_m) = (\min(x_i), \min(y_i), \min(z_i)) \tag{2}$$

$$v_b = \left\{ \begin{array}{l} (x_i, y_i, z_i) \mid \forall c_i = c_m \; or \; c_i = c_M, \\ for \; \mathrm{c} \in \{x, y, z\}, 1 \leq i \leq n \end{array} \right\} \tag{3}$$

By the distance between the maximum boundary point and the minimum boundary point, the minimum bounding box can be determined according to formula (4). Where L, W and H represent the length, width and height of the minimum bounding box respectively.

$$\|\mathrm{v}_M - v_m\|_2 = \sqrt{L^2 + W^2 + H^2} \tag{4}$$

Step 2: Calculate the center of mass O of the minimum bounding box. Formula (5) calculates the barycentric coordinates v_o of the minimum bounding box.

$$v_o = (x_o, y_o, z_o) = \frac{1}{2}(L, W, H) \tag{5}$$

Step 3: Translation model. The centroid of the model is translated to the origin of the coordinates, and the new vertices after the translation of the original vertices of the model are obtained according to formula (6).

$$v_i' = (x_i', y_i', z_i') = (x_i - x_o, y_i - y_o, z_i - z_o) \tag{6}$$

2.2 Classification Principle Based on Model Feature Similarity Calculation

In this paper, Euclidean distance is used to calculate the similarity between models, which can measure both the actual distance between two points in 3D space and the natural length of the vector [2]. Let the n-dimensional vectors constructed by two models A, B according to the same rule be $X = (x_1, x_2, ..., x_n)^T$ and $Y = (y_1, y_2, ..., y_n)^T$ respectively, then the Euclidean distance $d(X, Y)$ between the two vectors can be obtained by formula (7).

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{7}$$

The similarity threshold S_t is set, and the two models are classified according to the following rules: 1) If $d(X, Y) \leq S_t$, it indicates that the Euclidian distance between vector X and Y is relatively close, and the two models A and B are highly similar, so they are classified as the same carrier models; 2) If $d(X, Y) \leq S_t$, it means that the Euclidean distance between vector X and Y is far away, and the similarity between the two models A and B is low, so they are classified as heterogeneous carrier models.

2.3 Application of Magic Square Scrambling in Image Encryption Design Rules

A magic square is a square matrix filled with integers where the sum of every row, column, and diagonal is equal.

The idea of magic square scrambling is based on the idea of looking up tables [7,12,20]. In this process, "2" becomes the position of "1" and "3" becomes the position of "2", as shown in Fig. 1. The magic square scrambling of digital images can balance the scrambling effect and system overhead by reducing the magic square scrambling order or by scrambling image blocks.

2	9	4
7	5	3
6	1	8

(a) Original matrix

1	8	3
6	4	2
5	9	7

(b) 1st Magic Square

9	7	2
5	3	1
4	8	6

(c) 2nd Magic Square

Fig. 1. 3 × 3 magic square scrambling diagram

3 Multi-carrier Information Hiding Algorithm Based on Layered Compression of 3D Point Cloud Model

3.1 Multivector Classification

When a plane (clipping plane) is used to cut a 3D model, since the 3D model is mostly irregular, the cut plane outline is usually inconsistent. Using this property, different feature vectors are used to represent different models, and the models are classified by calculating the similarity between the models. The specific steps are as follows:

Step 1: Cut the plane slice.
Step 2: Determine the layer point cloud thickness. If the point cloud model is divided into m layers, the thickness δ of each layer of point clouds can be calculated by Eq. (8).

$$\delta = \frac{\max(z_i) - \min(z_i)}{m} \tag{8}$$

Step 3: Construct model feature vectors. The projected area S_i of slice contour was calculated by formula (9).

$$S_i = \iint_D dxdy, D = \{(x, y) \mid x_{m_i} \leq x \leq x_{M_i}, y_{m_i} \leq y \leq y_{M_i}\} \tag{9}$$

where, i_{m_x}, i_{m_y} represent the minimum coordinate value of the projected boundary point of the model contour obtained by any hierarchical operation, and i_{M_x}, i_{M_y} represent the maximum coordinate value of the boundary point. The projected area S_{max} of the enclosing box can be calculated by formula (10).

$$S_{\max} = L \times W \tag{10}$$

Construct the model feature vector F and define it as shown in formula (11).

$$F = (r_1, r_2, \ldots, r_i, \ldots, r_t) \tag{11}$$

where, the weight $r_i = s_i / s_{\max}\,(0 < r_i \leq 1)$, t is the number of slices.
Step 4: Model similarity calculation. Euclidean distance is used as an evaluation index for the similarity of the two models. As shown in formula (12), the smaller the calculated value, the higher the similarity of the two models.

$$Sim = \|F_1 - F_2\| \tag{12}$$

The similarity threshold $Sim_{(T)}$ is set [16], and the similar value Sim of the two models is compared: the model with $Sim \leq Sim_{(T)}$ is classified as the same, and the model with $Sim > Sim_{(T)}$ is classified as the different.

3.2 Feature Point Extraction

After slicing and layering the 3D point cloud model, the Meanshift clustering analysis method is used to extract the feature points of each layer region [11], and then the non-critical feature points are compressed for each region [14].

3.3 Information Hiding Rule

In this algorithm, according to the weight r_i of each hierarchical region, the region with greater weight is selected as the robust region, the region with less weight is selected as the vulnerable region, and other regions are selected as the information hiding region. The information hiding rules are designed as follows:

Rule 1: Hierarchical compression rule. The algorithm in this chapter realizes the compression of layered region by deleting some non-critical feature points of each layer of point cloud model. Assuming that the number of vertices in the original layering region is c and the number of points in the slice projection region after compression is c', then the compression ratio R is calculated as shown in formula (13):

$$R = \frac{c - c'}{c} \tag{13}$$

Rule 2: Heterogeneous model secret information embedding rule. If the 3D model carrier set is divided into class k, because each model has the same number of layers, then the secret information B is divided into k segments of the same length, denoted as B_1, B_2, ..., B_k. Let's say B_1, B_2, ..., B_n is arranged in order, the heterogeneous models are compressed according to the order of point cloud

compression rate from small to large, and the compression rate of similar models is the same. Different pieces of secret information are embedded in different types of carriers. Embed secret information from different fragments into different types of carriers, and merge the information from different fragments to obtain complete information $B = B_1 + B_2 + ... + B_n$.

Rule 3: Secret information embedding rules of the same class model. After the 3D model slices are layered, the projected area of each layer is calculated according to the order of weight from large to small, and the secret information is embedded in turn. The information is represented by whether the layered region of the model is compressed: if the secret information is 0, the layered region is not compressed; If the secret information is 1, the layer area is compressed.

3.4 Secret Message Embedding Steps

The overall information embedding process of the algorithm proposed in this paper is divided into the following six steps:

Step 1: Using the magic square scrambling method in Sect. 2, the secret image is scrambled to obtain the binary sequence and realize the preliminary encryption of the secret information.
Step 2: After normalizing the model, find out the maximum and minimum boundary points of the model, and generate the minimum bounding box of the 3D model.
Step 3: All the models in the carrier set are cut into the same number of layers, and the maximum and minimum boundary points and areas of each layer region are recorded, and entropy coding is carried out to embed them into the robust region.
Step 4: According to steps 2–3 in Sect. 3.1, the weights of each layered region are calculated, model feature vectors are constructed, and similarity is calculated to complete the classification of multiple models.
Step 5: According to "Rule 2", segment the secret information and select different compression rates in ascending order to compress the heterogeneous models.
Step 6: According to the preliminary encrypted information obtained in step 1, the layered areas that need to be modified in the model are compressed through "Rule 3" to complete information matching and embedding.

4 Theoretical Analysis and Experimental Comparison of Algorithm Performance

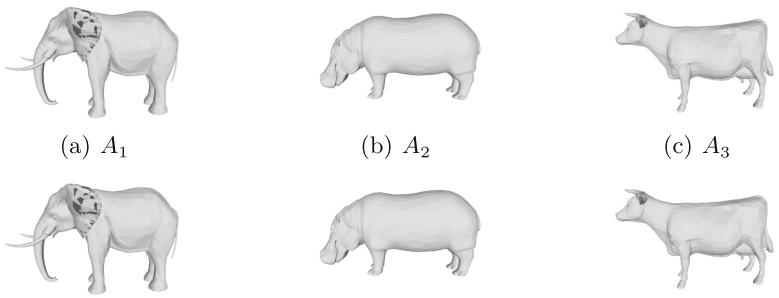
4.1 Comparison Between Simulation Algorithm and Experiment

The experimental environment of the algorithm in this paper is MatlabR2018, pycharm2022 and MeshLab2020. The GA algorithm proposed in literature and SS algorithm proposed in literature are used as the comparison algorithm of the experiment [13,17].

The layering number of the initial 3D point cloud model was set to 2^{17}, and the 3D model in the model carrier library was sliced. According to the result data, the similarity threshold was selected as 0.36 in this experiment to complete the classification of the carrier model. Three kinds of models are selected, and three models from each class are selected for information hiding experiment.

The Invisibility Experiment. The algorithm in this chapter comprehensively considers HVS features and Hausdorff distance to evaluate the invisibility of dense carrier models.

HVS Characteristics. Since the embedded information of the algorithm in this chapter is realized by compressing the non-critical feature points of each layer, the model is slightly modified. As shown in Fig. 2, A_1–A_3 is the original carrier, and A'_1–A'_3 is the dense carrier, which is no different from the original model in the eyes of the human visual system, satisfying the imperceptibility of the human visual system.



(a) A_1 (b) A_2 (c) A_3

(d) A'_1 (e) A'_2 (f) A'_3

Fig. 2. Original carrier and dense carrier

Hausdorff Distance The Hausdorff distance is a method used to measure the distance between two sets, defined as follows: for two sets of points P and Q, the Hausdorff distance from P to Q is shown in the formula (14):

$$H(P,Q) = \max\{h(P,Q), h(Q,P)\} \tag{14}$$

Among them, $h(P,Q) = \max(p \in P)\min(q \in Q)\|p-q\|$, $h(Q,P) = \max(q \in Q)\min(p \in P)\|q-p\|$.

In a 3D model, a model can be represented as a set of points, and the Hausdorff distance refers to the longest distance between the two sets of points, that is, the maximum distance from one point to another. If an attacker modifies a part of the model in such a way that it is far removed from the Hausdorff of the original model, then the invisibility of the model is affected. By calculating the

Hausdorff distance between the algorithm in this chapter and the comparison algorithm, experimental results are obtained as shown in Fig. 3.

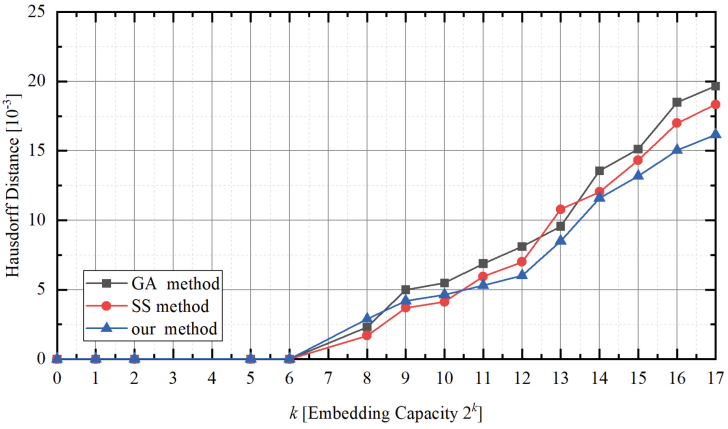


Fig. 3. Comparison of invisibility experiments based on Hausdorff distance

As can be seen from Fig. 3, when the embedding quantity index $k \geq 11$, the Hausdorff distance of the algorithm in this chapter is obviously always smaller than that of the comparison algorithm. When $k = 17$, the Hausdorff distance of the algorithm in this chapter is 16.15×10^{-3}, and the Hausdorff distance of the GA algorithm and SS algorithm is 19.67×10^{-3} and 18.34×10^{-3}, respectively. Compared with the comparison algorithm, the Hausdorff distance of the algorithm in this chapter is reduced by 17.90% and 11.94% respectively, indicating that the algorithm in this chapter can still ensure good invisibility when the embedding capacity is large.

Robustness Experiment. Correlation (Corr) refers to the correlation between the extracted secret information and the original secret information after the watermark extraction algorithm operates the model. The Corr value can measure the robustness of watermark information. The larger the Corr value is, the closer the extracted secret information is to the original embedded secret information, that is, the more robust the density-containing model is.

Single Attack. Taking the densely loaded model A_1' as an example, a series of single attack comparison experiments such as cutting, uniform simplification, random noise and Laplacian smoothing are carried out.

The schematic diagram of different degrees of shear is performed on the dense model A_1'. The experimental results are shown in Fig. 4.

As can be seen from Fig. 4, when only a small part of the model is cut, the Corr value of each algorithm has a small difference. When the cut rate reaches

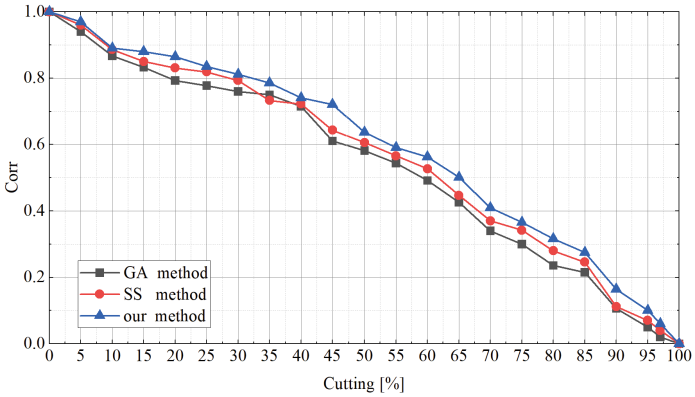


Fig. 4. Comparison of shear attack experiments

45%, the Corr value of the algorithm in this chapter is 0.721, and the Corr value of GA algorithm and SS algorithm is 0.611 and 0.643, respectively. Compared with the comparison algorithm, the performance of the algorithm in this chapter is improved by 18.00% and 12.13% respectively, indicating that the algorithm in this chapter can resist the shear attack robustly.

Schematic diagram of different degrees of simplification for the dense model A_1'. The experimental results are shown in Fig. 5.

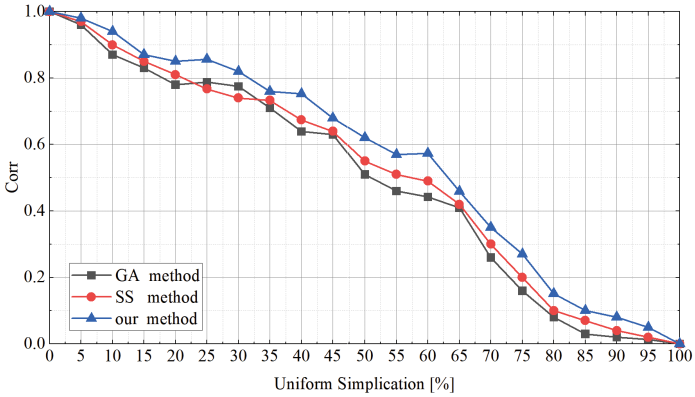


Fig. 5. Comparison of uniform simplified attack experiments

As can be seen from Fig. 5, the Corr value of the algorithm in this chapter is basically higher than that of the comparison algorithm, and when the simplification rate reaches 40%, the Corr value of the algorithm in this chapter is 0.753, and the Corr value of the GA algorithm and SS algorithm is 0.639 and 0.674, respectively. The performance of the algorithm in this chapter is improved by 17.84% and 11.72% respectively. Since the algorithm in this chapter embeds secret information by compressing each layer, even if the simplification rate is high, as long as some non-critical points in each layer are removed, the model will not cause large visual deformation, and the secret information can still be extracted more completely.

Noise is randomly added to the dense model A_1', and the experimental results are shown in Fig. 6.

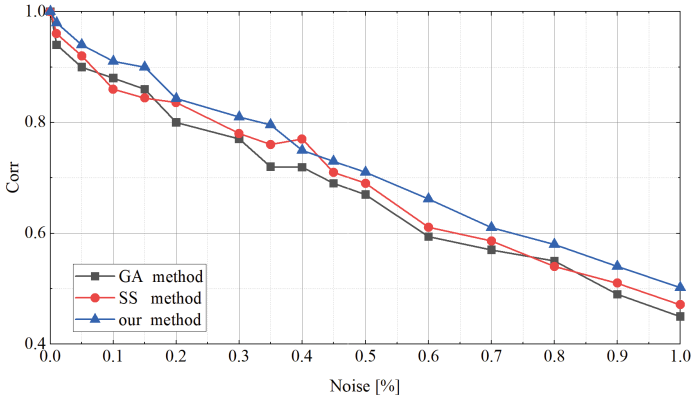


Fig. 6. Comparison of random noise attack experiments

As can be seen from Fig. 6, the Corr value of the three algorithms decreases with the increase of noise amplitude, but the overall Corr value of the algorithm in this chapter is greater than that of the comparison algorithm. When the noise amplitude reaches 0.6%, the Corr value of the algorithm in this chapter is 0.662, and the Corr value of the GA algorithm and the SS algorithm is 0.594 and 0.611, respectively. Compared with the comparison algorithm, the performance of the algorithm in this chapter is improved by 11.45% and 8.35% respectively, indicating that the algorithm in this chapter has strong robustness in resisting noise attacks.

Laplace smoothing of different iterations was performed on the dense model A_1', and the experimental results are shown in Fig. 7.

As can be seen from Fig. 7, the three algorithms all show high robustness when dealing with Laplacian smoothing attacks, and the Corr value of the algorithm in this chapter is slightly higher than that of the comparison algorithm. After 20 iterations, the Corr value of the algorithm in this chapter is 0.945, the Corr value of the GA algorithm and the SS algorithm are 0.889 and 0.919,

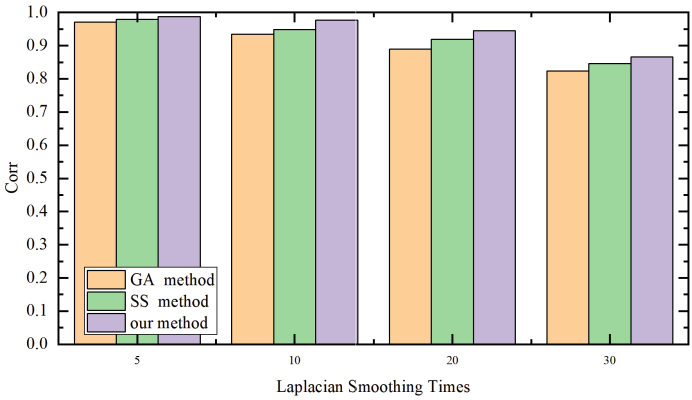
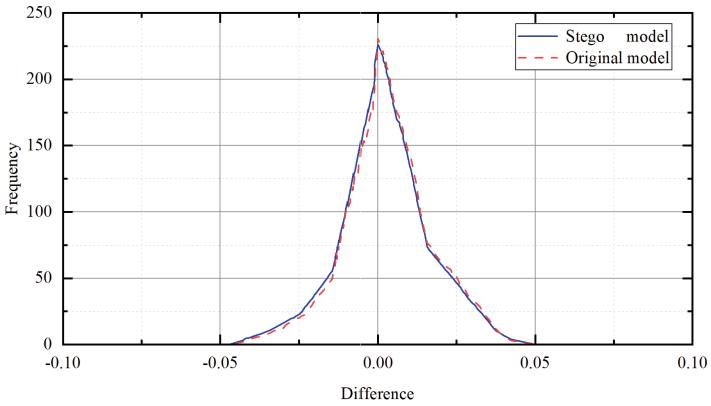


Fig. 7. Comparison of Laplace smoothing attack experiments

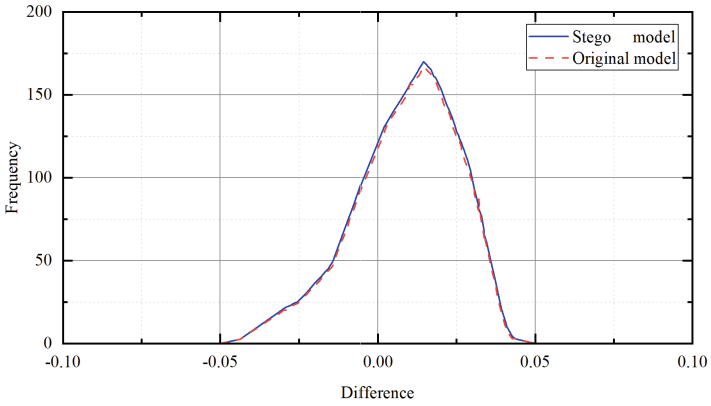
respectively. Compared with the comparison algorithm, the performance of the algorithm in this chapter is improved by 6.30% and 2.83% respectively, indicating that the algorithm in this chapter has strong robustness against the Laplacian smoothing attack.

Resistance to Analytical Experiments. Aiming at the general steganography algorithms, this paper presents steganography experiments based on Laplace smoothing statistics. The solid and dashed lines in Fig. 8 represent the absolute difference of the X component of vertex coordinates of the dense model A_1', the original model A_1 and its corresponding first-order Laplacian smooth model, and the distance difference between the vertex and the origin, respectively.

As can be seen from Fig. 8, the difference between the change curves of each feature of the densified model and the original model is very small, indicating that the statistical characteristics of the Laplace transform coefficient of the densified model are very similar to those of the original model, which makes it difficult for the Laplace statistics to distinguish the difference between the two, indicating that the algorithm in this chapter is difficult to steganographic analysis.



(a) Absolute difference of vertex X component



(b) Distance difference between vertex and origin

Fig. 8. Experimental results of steganalysis

5 Conclusion

This paper presents a multi-carrier information hiding algorithm based on 3D point cloud model with layered compression. The minimum bounding box of the model was generated and the model was stratified. The ratio of projected area between each layer region and the minimum bounding box was marked as the weight of each layer region, which was used as the model feature vector. The carrier set was classified by calculating the similarity of various models, and the weights of each layer were sorted at the same time. In this way, the information in the correct order can be extracted according to the weights in the secret information extraction stage, and the non-critical point compression of some hierarchical regions can be carried out according to the secret information

to ensure the invisibility of the algorithm. The experimental results show that the proposed algorithm can effectively resist many kinds of attacks and has some ability to resist steganography.

Acknowledgements. This work has been supported by the National Natural Science Foundation of China (No. 62372062), and the Fundamental Research Funds for the Central Universities, CHD (No. 300102240208).

References

1. Barequet, G., Har-Peled, S.: Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. J. Algorithms **38**(1), 91–109 (2001)
2. Fazel, M., Hindi, H., Boyd, S.P.: Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In: Proceedings of the 2003 American Control Conference, vol. 3, pp. 2156–2162. IEEE (2003)
3. Feng, X.: A watermarking for 3D point cloud model using distance normalization modulation. In: 2015 4th International Conference on Computer Science and Network Technology (ICCSNT), vol. 1, pp. 1449–1452. IEEE (2015)
4. He, Y., Li, G., Shao, Y., Wang, J., Chen, Y., Liu, S.: A point cloud compression framework via spherical projection. In: 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP), pp. 62–65. IEEE (2020)
5. Kammerl, J., Blodow, N., Rusu, R.B., Gedikli, S., Beetz, M., Steinbach, E.: Real-time compression of point cloud streams. In: 2012 IEEE International Conference on Robotics and Automation, pp. 778–785. IEEE (2012)
6. Ker, A.D.: Batch steganography and pooled steganalysis. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437, pp. 265–281. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74124-4_18
7. Lin, K.T.: Hybrid encoding method by assembling the magic-matrix scrambling method and the binary encoding method in image hiding. Opt. Commun. **284**(7), 1778–1784 (2011)
8. Liu, J., Yang, Y., Ma, D., He, W., Wang, Y.: A novel watermarking algorithm for three-dimensional point-cloud models based on vertex curvature. Int. J. Distrib. Sens. Netw. **15**(1) (2019)
9. Liu, J., Yang, Y., Ma, D., Wang, Y., Pan, Z.: A watermarking algorithm for 3D point cloud models using ring distribution. Trans. Edutain. XIV 56–68 (2018)
10. Luo, H., Pan, J.S., Lu, Z.M., Huang, H.C.: Reversible data hiding for 3D point cloud model. In: 2006 International Conference on Intelligent Information Hiding and Multimedia, pp. 487–490. IEEE (2006)
11. Paris, S., Durand, F.: A topological approach to hierarchical segmentation using mean shift. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
12. Rani, N., Mishra, V., Sharma, S.R.: Image encryption model based on novel magic square with differential encoding and chaotic map. Nonlinear Dyn. **111**(3), 2869–2893 (2023)
13. Ren, S., Xu, J., Zhang, Q., Shi, L., Lei, X., Dan, Z.: Information hiding algorithm based on spherical segmentation of 3D model. In: Proceedings of the 4th International Conference on Computer Science and Application Engineering, pp. 1–7 (2020)

14. Simpson, G.: Mechanics of non-critical fold-thrust belts based on finite element models. Tectonophysics **499**(1–4), 142–155 (2011)
15. Tian, Z., Gao, Z.: Multi-carrier steganography algorithm based on executable program. In: Sun, X., Zhang, X., Xia, Z., Bertino, E. (eds.) ICAIS 2022. LNCS, vol. 13340, pp. 363–372. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06791-4_29
16. Wang, L., Zhang, Y., Feng, J.: On the Euclidean distance of images. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8), 1334–1339 (2005)
17. Wang, X., Zhan, Y.: A digital watermarking algorithm for constructing vertex distribution features in 3D models. J. Comput. Aided Design Graph. **26**(2), 272–279 (2014)
18. Zhang, C., Hu, L., Hao, L., Peng, S.: Research on information encryption and hiding technology of 3D point cloud data model. In: 2020 International Conference on Computer Science and Management Technology (ICCSMT), pp. 54–58. IEEE (2020)
19. Zhang, X., Qian, Z., Li, S.: Prospects of information hiding research. J. Appl. Sci. **34**(5), 475–489 (2016)
20. Zhang, Y., Xu, P., Xiang, L.: Research of image encryption algorithm based on chaotic magic square. In: Jin, D., Lin, S. (eds.) Advances in Electronic Commerce, Web Application and Communication. Advances in Intelligent and Soft Computing, vol. 149, pp. 103–109. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28658-2_16