Multimedia Tools and Applications (2022) 81:8561–8573
https://doi.org/10.1007/s11042-022-11943-y

Coordinate quantization transformation CityGML watermarking

Ruichen Jin[1] · Jongweon Kim[2]

Received: 19 January 2020 / Revised: 5 February 2021 / Accepted: 3 January 2022
Published online: 3 February 2022

Abstract
As most of the contents are becoming digitized, robust and invisible copyright notation of visible digital contents is required. In this paper, we proposed a blind watermarking method for CityGML (City Geography Markup Language) model using a group coordinate quantization transformation (CQT) algorithm. Firstly, the Vertices of CityGML models were extracted from a text file. Secondly, the candidate coordinate of the vertex set was selected and it was quantized by a given space. Then, the quantized vertex set groups were sorted and selected. Finally, the watermark was inserted by altering their mean values. We embedded the watermark on buildings or objects on the map and the data load should be guaranteed and imperceptible should be maintained. The proposed method is blind and efficient for CityGML data copyright protection compared to the existing method.

Keywords CityGML watermarking · Quantization index modulation · Group quantization

1 Introduction

With the development of Virtual Reality (VR), computer games, and Augmented Reality (AR) technology, 3D models have been widely studied. Among them, the 3D city models, such as geographic information system (GIS), Computer-aided design (CAD), Building information models (BIM) and CityGML are well known [1]. Especially, the CityGML is a new and fashion open standardized data model and exchange format [8]. CityGML is an open standardized data model and exchange format for storing digital 3D models of cities and landscapes. Defines most common 3D features and objects found in cities and how to describe the relationships between them.

✉ Jongweon Kim
jwkim@smu.ac.kr

[1] HELIOSEN Co., Ltd., Seongnam-si, Gyeonggi-do 13487, Korea

[2] Department of Artificial Intelligence of Things, Sangmyung University, Seoul 03016, Korea

Due to the development of network technology and the spread of personal computers as a multimedia system, the problem of illegal distribution has emerged as a social issue. The CityGML as a Geography Markup Language (GML) application schema, is easy to be illegally distributed and illegally used without the permission of the copyrighters. Therefore, developing copyright protection methods for GML format files such as CityGML has big significance.

Digital watermarking is a classic and efficient way for copyright protection. Digital watermarking has been widely applied for copyright protection of multimedia such as images [22, 5], videos [6], audios [3], texts [17], even for GIS vector map [4], 3D meshes [15], and 3D printing [11]. The watermarking method covers the secret information — watermark to the original cover signal. The capacity of the watermark, the robustness, and the perceptibility, the security are the main factors needed to be considered [9]. Many watermarking and stereography methods have been studied and developed in the past decade. Research for robust and blind watermarking methods for covers such as image and video has formed a complete set of methodology. While, for 3D objects, the methods still have obvious defects, and the blind and robustness are the main problem needed to be solved. Ohbuchi [19] employed mesh-spectral analysis to modify mesh shapes in their transformed domain. The method is resilient against connectivity alterations and combined attacks. During the watermark extraction process, mesh alignment is needed. Song [21] proposed a method based on the watermarking of images from a virtual 3D scanner. The method using the principal component analysis (PCA) of the vertex points and modified the vertices of the model by using Discrete Cosine Transform (DCT). Zafeiriou [23] proposed method is robust against 3D rotation, translation, uniform scaling, and mesh simplifications. The watermark signal is covered by deforming vertices geometrically without altering the vertex topology. The method used the z-axis of the cartesian coordinates and spherical coordinated system. Cho [7] proposed method has been widely studied and developed, which using the distribution of the vertex norms. The mean value and variance are altered for casting watermark, the perceptibility of these methods also can be improved. Son [20] used the norm distribution method for watermark embedding, the improvement of the method is that it embedded the watermark into the salience region of the mesh. Li [18] also used the norm distribution method, however, the piecewise mapping function is combined used. Both the robustness and transparency have been improved.

The CityGML is formed as an XML file with some texture images. Theoretically, it can be watermarked into the text file, image file, and even geometric information. However, the CityGML has a big difference from the normal 3D meshes. Obviously, its geometric models don't have the normal topologies. In related research works, Kim [16] introduced several available watermarking methods. The zero-watermark [12] using the invariant feature of the model's vertex norm to generator the watermark, which is resistance geometric attacks and simplification. While the digital watermarking [13] by altering values of vertices to embed the watermark. The common of them is that vertex sets are partitioned into several regions. Hong [10] used the homograph to replace the text files' landmarks for watermarking, the method is robust to the simplification and crop attacks; however, the IDs of models' and polygons don't have any significance, which can be generated by a random generator and can be easily altered without changed any geometric information.

To realize the robustness and blind watermarking, in this paper, the authors proposed a blind digital watermark method for the CityGML model by using the group quantization method. A watermark bit is embedded by altering all the vertices in the same group.

The remainder of this paper is organized as follows: Section 2 reviews the related works; Section 3 presents the watermark embedding and extraction procedures; Section 4 presents and analyzes experimental results; Section 5 concludes the paper and proposes future working directions.

2 Related works

The proposed watermark method is based on group quantization. The watermark was embedded into the vertex information of the CityGML models. The groups were segmented based on the histogram algorithm.

2.1 CQT algorithm

Quantization is the digital process of mapping input values from a large set (often a continuous set) to output values in a (countable) smaller set, often with a finite number of elements. Rounding and truncation are typical examples of quantization processes [14].

A CQT-based technique embeds information by quantizing the original sample values. Typical CQT is accomplished by modulating a signal with the embedded information. Quantization is then performed using the associated quantizer.

2.2 Histogram grouping

In the paper, the watermark bits are embedded into each vertices group. The number of the groups decided the number of bits that can be watermarked. Therefore, the number of groups that available for watermarking should be large enough. However, if the space of each group is too small, the number of vertices in each group will not be enough for watermarking, even the number of each group is enough, some of the watermark bits may be inserted some times.

As shown in Fig. 1, the image (a)~(d) shows the histogram counts of each group of a CityGML object coordinate values. The object has 1299 vertices. The smallest space at the image (a) has the most groups with smaller counts in each group. The image (d) with the highest space has the least groups, and the first group has more than 180 vertices. At this condition, no more than 40 watermark bits can be embedded. It is obvious that the groups are decreased with the space increased.

First of all, all the watermark bits should be ensured to be embedded, therefore, the space value will be set from little to higher stage by step.

In the experiment, the space value was tested for finding an experience value.

3 Proposed methods

In CityGML, objects such as buildings often have difficulty in inserting watermark information because there are many polygons that need to maintain flatness, such as walls or roofs. Therefore, the strength of the watermark embedding should be strictly restricted. Unlike the 3D mesh, the CityGML objects have multiple polygons, not only

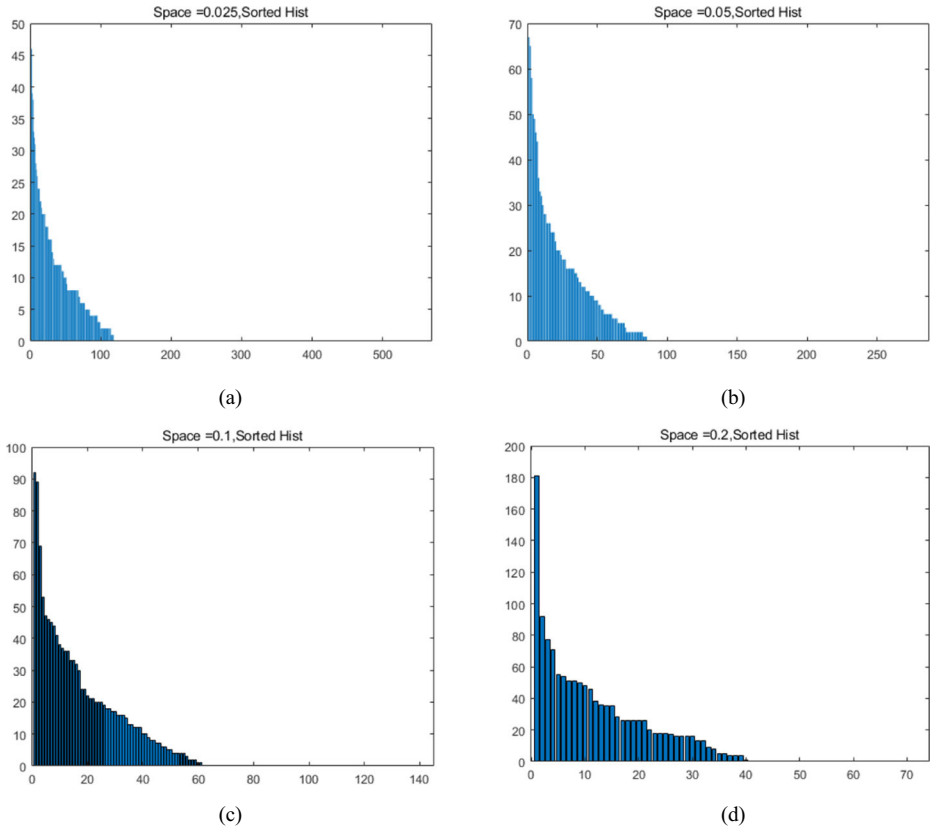

Fig. 1 Histogram Plot using different spaces. (**a**) Space = 0.025. (**b**) Space = 0.05. (**c**) Space = 0.1. (**d**) Space = 0.2

limited in triangular. Therefore, the polygons information is hard to be applied for watermarking.

3.1 Watermark embedding

In the paper, only the vertices information is used to simplify the process. The steps of the watermark embedding are listed as follow:

Step 1: Parse the CityGML text file, and extract the vertex information. Vertex set is represented as $V = \{V_i \mid V_i = (V_{ix}, V_{iy}, V_{iz}), i = 1, 2, \ldots N\}$. N is the number of the unique vertex. V_{ix}, V_{iy}, and V_{iz} represent the value of the *i*th vertex in the X, Y, and Z coordinate.

Step 2: Select the appropriate coordinate of the vertex set. Since distribution of the values in the X, Y, and Z coordinates axis are different. Here chooses the coordinate that has a wide range and nearly uniform distribution as possible. At the default condition, the Z coordinate axis is selected. So, the selected vertex set for watermarking is $V_z = \{V_{iz} \mid i = 1, 2, \ldots N\}$.

Step 3: The given space of the bin range is Sp. The quantized vertex set is $QV_z = [V_z/Sp]*Sp$. Here, [] is the rounding operation, and * is the multiple operations.

Step 4: Partitioning the quantized vertex set and select the partitions that available for watermarking.
Step 5: Sorting the vertex groups by descending order of the number of vertices in each group.
Step 6: Embedding the watermark of binary sequences $Wm = \{W_{mi} \mid W_{mi} \in \{1,0\}, i = 1, 2, \ldots M\}$ by altering the values in each vertex group. M is the length of the watermark. All the values of the same group are added or reduced by a given strength value delta Δ according to the watermark value.

3.2 Watermark extraction

The foremost processes of watermark extraction are similar to the watermark embedding process. The last process is the reverse process of the vertex altering process. The extraction process is easy to operate. The original watermark information is not needed. The mean value of each group is calculated and compared to a given threshold. Here, the threshold is the center value of each group.

The flowchart of the proposed watermarking embedding and extraction method is shown in Fig. 2(a) and (b), respectively.

Step 1: Parse the CityGML text file, and extract the vertex information.
Step 2: Select the appropriate coordinate of the vertex set. Since distribution of the values in X, Y, and Z coordinates axis are different. Here chooses the coordinate that has a wide range and nearly uniform distribution as possible. At the default condition, the Z coordinate axis is selected.
Step 3: The given space of the bin range is Sp. The quantized vertex set is $QV_z = [V_z/Sp]*Sp$.
Step 4: Partitioning the quantized vertex set and select the partitions that available for watermarking.
Step 5: Sorting the vertex groups by descending order of the number of vertices in each group.
Step 6: Choose segments and quantize the vertices. Group quantization segments vertices into similar ranges of values and quantizes the values of vertices composed of groups into Δ based on the median of the segment. Calculate the mean of each segment then compares it to the center value of the segment. If the mean is greater equal than the center value the watermark is 1 and in the opposite case, it is 0.

4 Experimental results and analysis

4.1 Evaluation methods

In the experiment, the authors use Vertex Signal-to-noise Ratio (VSNR) as the evaluation of transparency. Where Nv is the number of all unique vertex, (x_i, y_i, z_i) and (x_i^*, y_i^*, z_i^*) are the original and watermarked Cartesian coordinates of the vertices.

$$VSNR = 10*log_{10}(SNR) = \frac{\sum_{i=1}^{Nv}\left(x_i^2+y_i^2+z_i^2\right)}{\sum_{i=1}^{Nv}\left(\left(x_i^*-x_i\right)^2+\left(y_i^*-y_i\right)^2+\left(z_i^*-z_i\right)^2\right)}$$

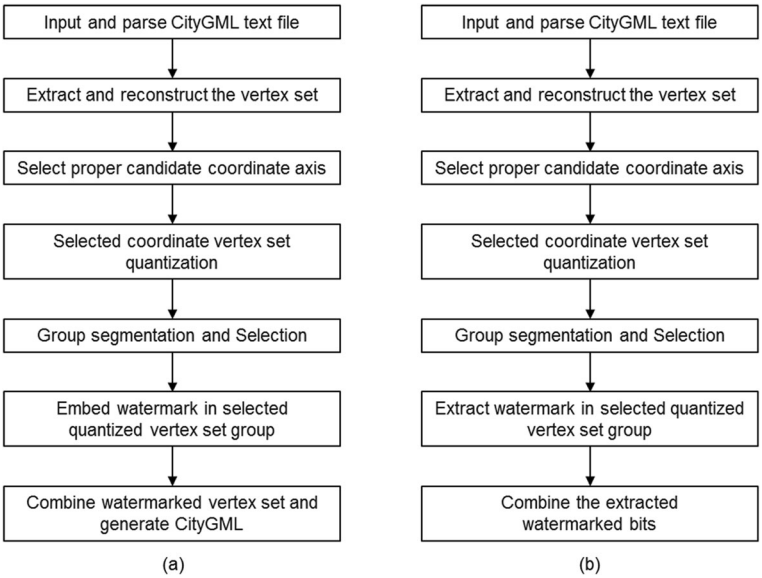

Fig. 2 Flowchart of watermark embedding (**a**) and watermark extraction (**b**)

The Bit Error Rate (BER) is used to estimate the robustness of the method. Where *Wm'* and *Wm* are the extracted watermark and original watermark, respectively. Where *Nw* is the length of the watermark and $\oplus$ is the bit-wise XOR function.

$$BER = (W_m \oplus W'_m)/N_w$$

4.2 Test CityGML models

The CityGML contains various objects those can be found in cities, such as terrain, buildings, railways, tunnels, bridges, water bodies, vegetation, city furniture, generic city objects, and city object groups. Here, the authors select some objects that at different Level of detail (LOD) with the different number of vertices for an experiment. The information of each object is listed as below (Fig. 3):

4.3 Test normal 3D models

To compare the proposed method with the existing method, the authors also take some famous 3D models for experiments. The models can be downloaded from the website [2]. Those meshes all have more than 2000 vertices, the 'Horse' and the 'Venus' models even have more than 100000 vertices. At this condition, the watermark can be completely embedded (Fig. 4).

4.4 Attack experiment

In general, there are kinds of routine attacks on a watermarked 3D models: file attack, geometry attack, and connectivity attack [2].

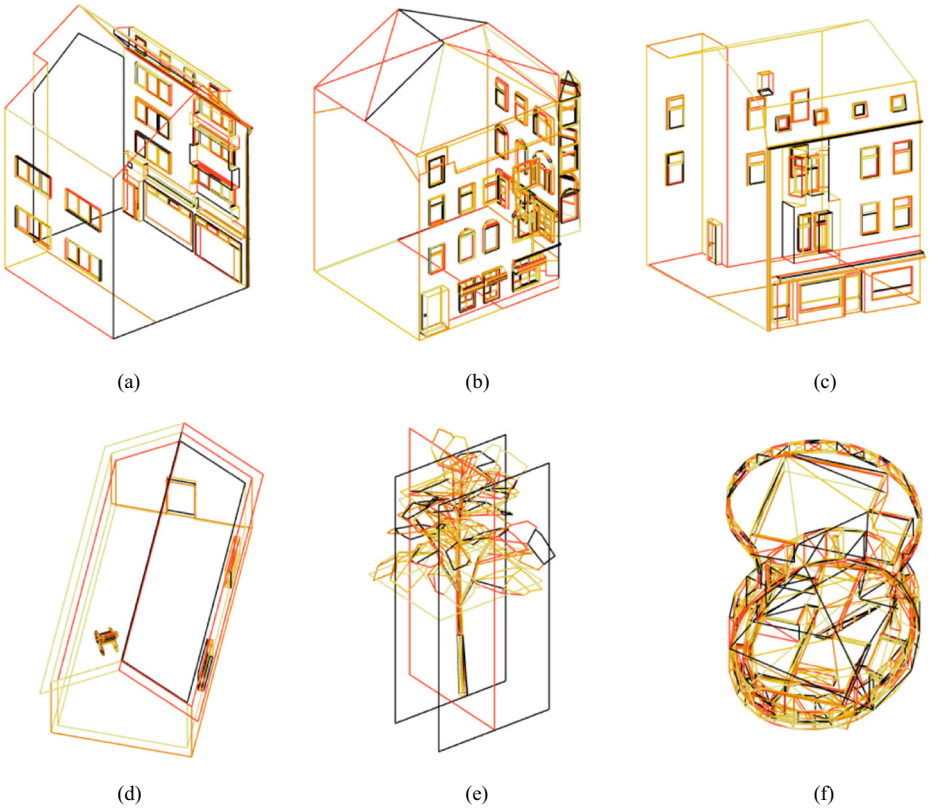

Fig. 3 Samples of test CityGML objects. (**a**) BD1-LOD3-1299 vertices. (**b**) BD2-LOD3-2138 vertices. (**c**) BD3-LOD3-1083 vertices. (**d**) BD4-LOD4-1413 vertices. (**e**) SVO1-LOD2-562 vertices. (**f**) BD5-LOD2-597 vertices

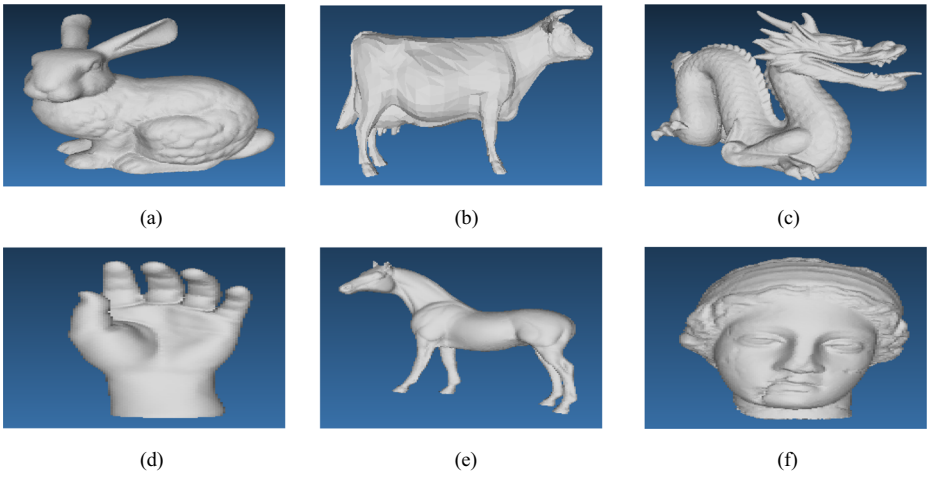

Fig. 4 Samples of test normal 3D meshes. (**a**) Bunny-34,835 vertices. (**b**) Cow-2904 vertices. (**c**) Dragon-50,000 vertices. (**d**) Hand-36,619 vertices. (**e**) Horse -112,642 vertices. (**f**) Venus -100,759 vertices

For file attacks such as changing format may change the geometric shape of the CityGML, for example, converting the CityGML file to STL (stereolithography) file will lose some information, because all the polygons are changed into triangular. However, it has no effect on the normal 3D models. In the paper, this condition is not considered.

For geometry attack, only the vertex coordinates are modified while the mesh connectivity is kept unchanged.

- Vertex Reordering Attack.

The vertices of CityGML models do not have an implicit order, therefore, vertex reordering is needed to keep the watermark embedding and extraction consistent.

- RST attacks.

The common type of geometric attack operations such as rotation, translation, scaling, and cropping also can be applied to CityGML objects. The distortion due to geometric attack is clearly visible. The vertices positions will be modified by rotation, translation, and scaling.

- Noise addition attack.

To evaluate the resistance to noise attack, normally distributed random noise was added to each vertex. Here, the noise rate represents the noise amplitude as a fraction of the mean vertex norm of the object. Each noise rate was conducted three times using different random seeds.

In a connectivity attack, the mesh connectivity information is changed, at the same time, the vertices coordinate may also be changed.

- Simplification attack.

In practical applications, the watermark is often embedded in the original complex model, and then the mesh is simplified so as to adapt to the capacity of the available resources.

- Cropping attack.

The vertices numbers will be reduced by cropping. The watermark segmentation or object segmentation methods can against it. Embedding the watermark sub-blocks into different objects is easier and better than embedding the integral watermark into object sub-blocks. Due to the complex geometric structure, the CityGML objects are difficult to be segmented into several sub-blocks. Here, the cropping attack is only focused on each object.

4.5 Experiment results and analysis

- Without attack.

Here, the authors employed the watermarking method based on group vertex norm distribution from Cho [9].

to conduct the comparison experiments. Without any attacks, the results of BER and VSNR values by using two methods are listed in Table 1 for CityGML models, the length of the watermark is 32.

The table shows that the proposed method can extract the 32 bits watermark without error under the test environment. The model SVO1 has a big difference from other models of VSNR value because the coordinates values of the model SVO1 are in a small range so that even a little change will bring a high difference to the original cover model.

Table 1 Comparison experiment results under watermark length 32

		BD1	BD2	BD3	BD4	SVO1	BD5
Proposed	VSNR (dB)	164.54	168.40	164.49	183.62	55.23	118.22
	BER	0	0	0	0	0	0
Cho [9]	VSNR (dB)	164.24	167.37	163.39	178.57	56.82	118.73
	BER	0	0	0	6.25	0	0

To evaluate the performance of the proposed method on normal models, the authors also using test normal models for comparison experiments. In the experiment, making the VSNR value of the proposed method as close as possible to the VSNR value of the Cho method(Table 2).

The results listed in the above table show that the proposed method can completely and correctly extract the watermark from normal models at the same time with higher perceptibility.

- Vertex reordering attack

Since the quantized vertices have been sorted and the watermark was embedded into each group, so that the reordering attack has no effect on the proposed method. Consequently, the proposed method can resist the vertex reordering attack.

- RST attacks.

Apply scaling transform: Scaling up z-axis values by multiplying scaling factor 2. For Cho's method on BD1, the BER value is 0.47. So that it cannot resist the ununiform scaling. Then scaling up all the axis at the same factor, the watermark can be correctly extracted. Because the method has a normalized process, therefore, it is resisting uniform scaling attacks. For the proposed method, the watermark was embedded into a selected axis coordinate. As long as the transformation occurs on the axis coordinate that has embedded the watermark, there is no difference between the effects of a uniform scaling attack and a non-uniform scaling attack. Whatever, the proposed method is not robust to the scaling attacks.

Apply rotation transform: Due to Cho's method using the vertex norm for watermarking, it is invariant to the rotation attacks. Suppose the watermark was embedded into the z-axis, create an affine transform object that defines a 45-degree rotation along the z-axis. On this condition, the watermark can be correctly extracted, since the watermarked values have not been changed. For other situations, only if the values of the z-axis have been changed, the watermark cannot be correctly extracted.

Apply translation transform: The vertex norm is the distance to the center of the model so that it is invariant to the translation attack. While the proposed method has a process to

Table 2 Comparison experiment results under watermark length 64

		Bunny	Cow	Dragon	Hand	Horse	Venus
Proposed	VSNR (dB)	60.15	62.48	54.82	53.43	57.10	59.67
	BER	0	0	0	0	0	0
Cho [9]	VSNR (dB)	52.93	61.23	43.05	47.05	56.97	57.98
	BER	0	0	0	0	0	0

segment the groups, which used the maximum value and minimum value of the coordinate, therefore, the proposed method is robust to the translation attack.

- Noise addition attack.

Applying the noise addition attack, the vertices set were changed. Here, 10 different noise rates from 0.001 to 0.019 by step 0.002 were used to test the performance of noise resistance. While conducting an experiment on the test set of CityGML models, the results under various noise factors are shown in Fig. 5.

While conducting an experiment on the test set of normal models, the results under various noise factors are shown in Fig. 6.

From Figs. 5 and 6, it can be concluded that the proposed method has better performance than Cho's method under noise addition attacks. Especially for CityGML models, the differences between them are obvious. The cow model has higher BER values than other models. The number of vertices of it is the key induction factor. The VSNR values of the normal model are much less than the CityGML model, since the number of vertices in normal models is larger than CityGML's, the sum value of VSNR.

- **Simplification attack**.

Here, the authors using the point cloud down sample algorithm to simulate the simplification attack on CityGML models. The parameter 'gridstep' is used to indirectly control the number of sampled points. With higher gridstep, the number of the down sample point cloud is lower. In the experiments, for both the proposed method and Cho's method, the damage from the simplification attack to the watermark is fatal.

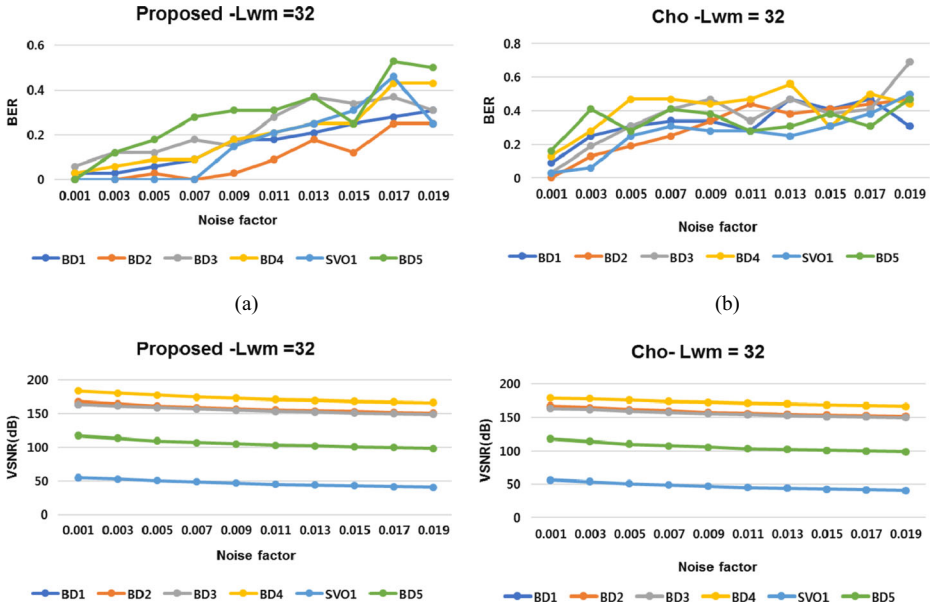

Fig. 5 Comparison results of BER and VSNR on test CityGML models. (**a**) BER of proposed method on CityGML. (**b**) BER of Cho [9] on CityGML. (**c**) VSNR of our method on CityGML. (**d**) VSNR of Cho [9] on CityGML

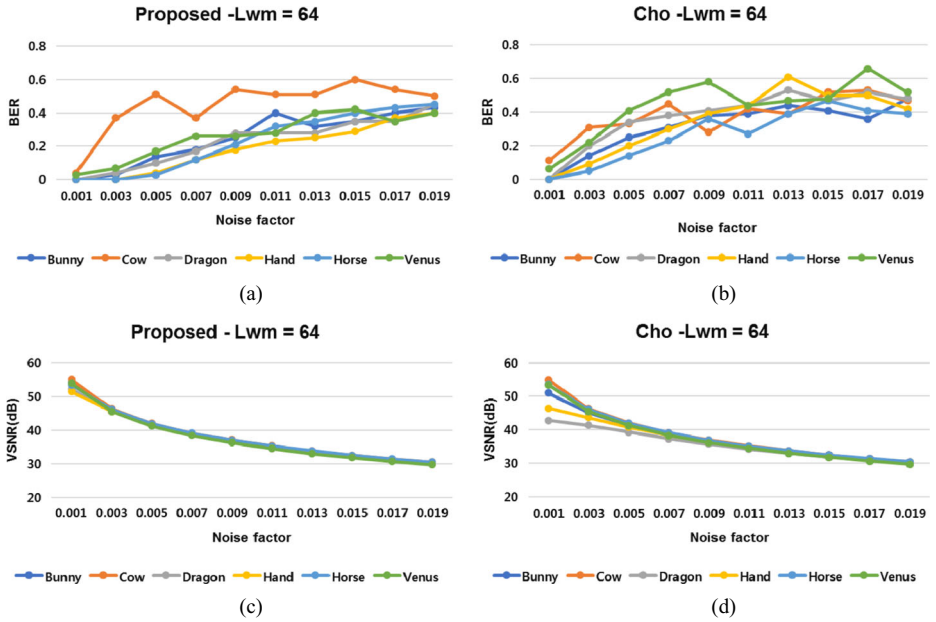

Fig. 6 Comparison results of BER and VSNR on test normal models. (**a**) BER of proposed method on normal models. (**b**) BER of Cho [9] on normal models. (**c**) VSNR of our method on normal models. (**d**) VSNR of Cho [9] on normal models

- **Cropping attack**.

Firstly, cropping the vertices set at the z-axis, delete vertices which values along the z-axis are higher than a threshold. On this condition, the length of the vertices set is reduced. There has a situation that the cropped vertices did not embed the watermark, therefore, the watermark can be correctly extracted. However, it is an extremely idealized situation. Under the situation, the proposed method is more robust against crop attack than Cho's method. Because Cho's method is highly relied on the total coordinates' values, while the proposed method using the group segment and embed the watermark literately. Taking BD1 as an example, cropping the vertices those z-axis values higher than 80% of the maximum z-axis value, the proposed method can correctly be extracted the watermark, while the BER of Cho's method is 0.28. When cropping vertices which a z-axis value higher than 60% of the maximum a-axis value, the BER of the proposed method is 0.28, while the BER of Cho's method is 0.68.

5 Contribution and conclusion

Watermark embedding of coordinated buildings is more difficult than general 3D object models. 3D object models tend to have even the density of vertexes to represent the surface. Therefore, watermark embedding in 3D object models is excellent in invisibility. However, in a building, the distribution of vertices is large and even in the roof or curved part, whereas the wall or floor has fewer and dense vertices. In this case, the slope of the building is very obvious even with small changes.

In this paper, we proposed a watermarking method for CityGML model using a group coordinate quantization transformation (CQT) algorithm. This inserts the watermark inconspicuous through the fine movement of coordinates. The histogram is used to figure out the movement of the points, which shows that it is robust against various attacks. Compare with the 3D model, the CityGML has the different positional characteristics of the vertices. Unlike 3D models, CityGML has a large number of vertices located on the same plane. Because of these characteristics, the group coordinate quantization transformation method is suitable for the watermarking method of the cityGML model. With the proposed CityGML watermark embedding method, we overcame the specialized problem of the building by inserting sufficient data load and making it difficult to discern after embedding the watermark with the eyes.

References

1. 3D city models. https://en.wikipedia.org/wiki/3D_city_models. Accessed 13 Jan 2022
2. A benchmark for 3D mesh watermarking. https://projet.liris.cnrs.fr/meshben/. Accessed 13 Jan 2022
3. Bassia P, Pitas I, Nikolaidis N (2001) Robust audio watermarking in the time domain. IEEE Trans Multimed 3(2):232–241
4. Chang HJ, Jang BJ, Lee SH, Park SS, Kwon KR (2009) 3D GIS vector map watermarking using geometric distribution. In: 2009 IEEE International Conference on Multimedia and Expo. IEEE, pp 1014-1017
5. Chen L, Zhao J (2020) A robust blind watermarking algorithm for depth-image-based rendering 3D images. Signal Process Image Commun 87:115935
6. Cheng Q, Huang TS (2000) Blind digital watermarking for images and videos and performance analysis. In: 2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast-Changing World of Multimedia (Cat. No. 00TH8532, vol 1). IEEE, pp 389-392
7. Cho JW, Prost R, Jung HY (2006) An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. IEEE Trans Signal Process 55(1):142–155
8. CityGML. https://en.wikipedia.org/wiki/CityGML. Accessed 13 Jan 2022
9. Cox IJ (2008) Digital watermarking and steganography. Morgan Kaufmann, Burlington
10. Hong DJ, Kim JW (2019) CityGML Watermarking Using Homographs. Information and Communication Technology and Digital Convergence Business (ICIDB-2019)
11. Hou JU, Kim DG, Lee HK (2017) Blind 3D mesh watermarking for 3D printed model by analyzing layering artifact. IEEE Trans Inf Forensics Secur 12(11):2712–2725
12. Jiang DY, Kim JW (2018) Zero watermarking scheme for CityGML. J Theor Appl Inf Technol 96(22): 7455–7463
13. Jiang DY, Kim JW (2018) A blind watermarking method for CityGML. In: Proceedings of 12th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, Madrid, Spain, 17 – 20 July (2018), 229-236
14. Jin RC, Kim JW (2019) Digital watermarking using group quantization for CityGML Objects. In: 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, pp 718-720
15. Khalil OH, Elhadad A, Ghareeb A (2020) A blind proposed 3D mesh watermarking technique for copyright protection. Imaging Sci J 68(2):90–99
16. Kim JW (2017) Digital watermarking model for CityGML. J MacroTrends in Technol and Innov 5(2):1-11. https://macrojournals.com/yahoo_site_admin/assets/docs/1TI17Ki.225555.pdf. Accessed 13 Jan 2022
17. Kim YW, Moon KA, Oh IS (2003) A text watermarking algorithm based on word classification and inter-word space statistics. In: ICDAR, pp 775–779
18. Li S, Ni R, Zhao Y (2017) A 3D mesh watermarking based on improved vertex grouping and piecewise mapping function. J Inf Hiding Multimed Signal Process 8(1):97e108
19. Ohbuchi R, Mukaiyama A, Takahashi S (2002) A frequency-domain approach to watermarking 3D shapes. Computer Graphics Forum, vol 21. Blackwell Publishing, Inc., Oxford, pp 373–382
20. Son J, Kim D, Choi HY, Jang HU, Choi S (2017) Perceptual 3D watermarking using mesh saliency. In International Conference on Information Science and Applications. Springer, Singapore, pp 315-322
21. Song HS, Cho NI, Kim JW (2002) Robust watermarking of 3D mesh models. 2002 IEEE Workshop on Multimedia Signal Processing. IEEE, pp 332–335

22. Van Schyndel RG, Tirkel AZ, Osborne CF (1994) A digital watermark. In Proceedings of 1st International Conference on Image Processing, vol 2. IEEE, pp 86-90
23. Zafeiriou S, Tefas A, Pitas I (2005) Blind robust watermarking schemes for copyright protection of 3D mesh objects. IEEE Trans Vis Comput Graph 11(5):596–607