A High-Capacity Data Hiding Method for Polygonal Meshes*

Hao-tian Wu and Yiu-ming Cheung

Department of Computer Science,
Hong Kong Baptist University, Hong Kong, China

Abstract. This paper presents a high-capacity data hiding method for 3D polygonal meshes. By slightly modifying the distance from a vertex to its traversed neighbors based on quantization, a watermark (i.e., a string of binary numbers) can be embedded into a polygonal mesh during a mesh traversal process. The impact of embedding can be tuned by appropriately choosing the quantization step. The embedded data is robust against those content-preserving manipulations, such as rotation, uniformly scaling and translation, as well as mantissa truncation of vertex coordinate to a certain degree, but sensitive to malicious manipulations. Therefore, it can be used for authentication and content annotation of polygonal meshes. Compared with the previous work, the capacity of the proposed method is relatively high, tending to 1 bit/vertex. Besides to define the embedding primitive over a neighborhood so as to achieve resistance to substitution attacks, the security is also improved by making it hard to estimate the quantization step from the modified distances. A secret key is used to order the process of mesh traversal so that it is even harder to construct a counterfeit mesh with the same watermark. The numerical results show the efficacy of the proposed method.

1 Introduction

With the development of digital modeling and visualization techniques for 3D objects, 3D models have been widely created and used for geometry representation, such as the cultural heritage recording like Digital Michelangelo Project [1], CAD models, and structural data of biological macromolecules [2]. As more and more 3D models appear, polygonal meshes in particular, how to hide information within them [3] has received much attention for a variety of purposes, ranging from copyright enforcement (e.g. [9, 10]) to authentication (e.g. [4, 6]). In this paper, we only discuss fragile watermarking of polygonal meshes, which is contrast to robust watermarking for the fragility of the embedded watermark. Compared with digital images, video and audio streams, there exists no grid for meshes, i.e., each vertex in a mesh is connected with variable neighboring vertices at different distances. This flexibility of mesh data makes it an attractive cover object for data hiding.

* This work was supported by a Faculty Research Grant of Hong Kong Baptist University.

In the literature, quite a few watermarking methods (e.g.[4]-[18]) have been proposed to embed data into meshes. Depending on the applications, the requirements are different. For instance, one purpose of robust watermarking is to protect the copyright of digital works so that the embedded watermark is designed robust against outer processing while the original work can be used in the retrieval process [10]. In contrast, in fragile watermarking for authentication and integrity verification, the embedded data should be blindly retrieved and sensitive to illegal modifications [4], and high information rate is preferred. Nevertheless, there are some common requirements, such as security and fidelity. In [19], T. Kalker defined the security of robust watermarking as the inability of unauthorized users to remove, detect or change the watermark. A data hiding scheme is considered secure if there is little information leakage from the public domain. It should be assumed that the algorithms are publicly known and the attacker has sufficient computational capability so that some valuable information may be leaked from the observation of watermarked objects. Fidelity means that the embedded data is invisible (except the case that it is intentionally visible), i.e., the embedding process should not introduce noticeable distortion to the cover object. And it is often required that the introduced error can be numerically analyzed and bounded.

Only a few fragile watermarking algorithms (e.g.[4]-[8]) have been proposed for authentication of polygonal meshes. The first fragile watermarking of 3D objects is addressed by Yeo and Yeung in [4] for authentication and integrity protection by using a set of lookup tables (LUTs). If two values generated from the positions of a vertex and its traversed neighboring vertices are identical to each other, the vertex is considered as valid. Otherwise, its position will be perturbed until the two values match. Since the data embedded in [4] is sensitive to *R*otation, uniformly *S*caling and *T*ranslation transformations (denoted as RST hereinafter), its applications may be limited. By adapting the work in [4], Lin et al. proposed a fragile watermarking method in [5] to detect malicious attacks. They improve the mapping from vertex positions to location indices so that the embedded watermark is resistant to incidental data processing, such as vertex reordering, but RST transformations are still not allowed. Moreover, Benedens and Busch proposed the algorithm called Vertex Flood Algorithm (VFA) in [6] for mesh authentication. Basically, their algorithm modifies the vertices so that their distances to the centroid of a designated triangle encode the watermark bits. In this way, a certain amount of vertex coordinate truncation caused by format conversions, as well as RST transformations, can be allowed. As for a triangle mesh, the security of VFA relies on the selection of the start triangle since the vertex position can be modified without changing the distance from it to the centroid of the start triangle. Later, Cayre and Macq presented a steganographic scheme [7] for triangle meshes by treating a triangle as a two-state geometrical object. By choosing an appropriate Macro Embedding Procedure (MEP) order, a watermark can be imperceptibly embedded with robustness against RST transformations. The upper bound of capacity has been given in [7], but the optimal mesh traversal to reach it has not been addressed yet. Alternatively, in

our previous work [8], a fragile watermark robust against RST transformations is embedded into polygonal meshes by quantizing the distances from the surface polygons to the mesh centroid. By choosing an appropriate quantization step, the embedded watermark can be made imperceptible and sensitive to illegal modifications. Although high information rate is required in fragile watermarking, the upper bound of capacity has not been reached in [8].

This paper presents a new data hiding method for polygonal meshes, in which the embedded data is designed to be robust against those content-preserving manipulations, such as RST manipulations and truncation of vertex coordinates to a certain degree, but sensitive to malicious manipulations. A new quantization method is employed to embed a watermark (i.e., a string of binary numbers) by slightly modifying the distance from a vertex to the centroid of its traversed neighbors. The impact of the embedding process, i.e., the difference between the original and watermarked meshes, can be tuned by choosing an appropriate quantization step. The capacity of the proposed method tends to 1 bit/vertex, which is higher than the former methods, such as 0.877 bit/vertex in [7]. It can be used for content annotation and authentication of polygonal meshes, or even secret message communication.

The rest of this paper is organized as follows. In the following section, the procedure of the data hiding method, including watermark embedding and retrieval, will be described in detail. The experimental results will be given and discussed in Section 3 by implementing the proposed method to authentication of polygonal meshes. Section 4 summarizes the paper and points out the future works.

2 A New Method to Hide Data within Polygonal Meshes

Polygonal meshes are considered as the common representation of 3D shapes and it's easy to convert other types of 3D models into them. Despite the appearance attributes associated with 3D models, such as color, transparency and texture, there are two parts of information contained in the mesh data, i.e. the mesh geometry and topology. The mesh geometry can be represented by the set of vertex positions $V = \{v_1, \cdots, v_m\}$, which defines the shape of the mesh in R^3 given m vertices in a mesh. The mesh topology, i.e., the connectivity between vertices, specifies the n vertices $\{v_k^1, \cdots, v_k^n\}$ in the k-th polygon, as described by IndexedFaceSet in VRML [20] format. The proposed method is performed on polygonal meshes, consisting of embedding and retrieval processes, detailed as follows.

2.1 Data Embedding

Given a string of binary numbers $W = (w_i)_{i=1}^N$ with the length N, the task of embedding is hide the value of each bit w_i into the mesh geometry. Since we aim to embed a watermark robust against RST transformations, the ratio between the distances in the cover mesh serves as a good candidate. In our method, the

distance from a vertex to the centroid of its traversed neighbors is chosen as the embedding primitive so that the upper bound of capacity can be reached. If we choose the distance from a vertex to the centroid of all its neighbors as the embedding primitive and modify the distance to embed a binary number by adjusting its position, the positions of its neighboring vertices cannot be changed any more to preserve the embedded value. As a result, the capacity will drop since most of the vertex positions cannot be modified to embed binary numbers. Therefore, only the traversed vertices of each vertex are chosen to generate the embedding primitive so that high information rate is achieved.

The detailed process to embed a watermark $W = (w_i)_{i=1}^{N}$ is as follows: Initially, we use a secret key K as the seed of pseudo-random generator to permute the face indices F and vertex indices I, respectively. The process of mesh traversal is ordered by the permuted vertex indices I' and face indices F' as follows. Among those vertices in the polygon lastly indexed by F', the one first indexed by I' is traversed at first without adjusting its position since all of its neighboring vertices have not been traversed. Among the neighbors of the traversed vertices, the one first indexed by I' will always be subsequently traversed. Suppose there is m vertices in a polygonal mesh, there are $m-1$ embedding primitives because only the first traversed vertex has no traversed neighbor. For a newly traversed vertex v_i, N_i neighboring vertices have been traversed and denoted as $(v_i^j)_{j=1}^{N_i}$. Then the centroid of the traversed neighbors can be calculated by

$$v_{ic} = \frac{1}{N_i} \sum_{j=1}^{N_i} v_i^j. \tag{1}$$

The distance d_i from v_{ic} to v_i is chosen as the embedding primitive

$$d_i = \sqrt{(v_{icx} - v_{ix})^2 + (v_{icy} - v_{iy})^2 + (v_{icz} - v_{iz})^2}, \tag{2}$$

where $\{v_{icx}, v_{icy}, v_{icz}\}$ and $\{v_{ix}, v_{iy}, v_{iz}\}$ are the coordinates of v_{ic} and v_i in R^3, respectively. To embed a binary number w_i by slightly changing d_i with the quantization step Δ, its corresponding integer quotient Q_i and the remainder R_i should be calculated by

$$\begin{cases} Q_i = \lfloor d_i/\Delta \rfloor \\ R_i = d_i \% \Delta \end{cases}, \tag{3}$$

and d_i is modified by

$$d_i' = \begin{cases} d_i & \text{if } Q_i\%2 = w_i \\ d_i + 2 \times (\Delta - R_i) & \text{if } Q_i\%2 \neq w_i \ \&\ R_i \geq \frac{\Delta}{2} \\ d_i - 2 \times R_i & \text{if } Q_i\%2 \neq w_i \ \&\ R_i < \frac{\Delta}{2} \end{cases} \tag{4}$$

so that $\lfloor d_i'/\Delta \rfloor \% 2 = w_i$. The error introduce by Eq.(4), i.e., the difference between the modified distance d_j' and d_i, will not exceed the quantization step Δ so that the impact of embedding on the mesh content can be tuned with the quantization step Δ. To allow slight change of d_j', such as mantissa truncation

due to the limited precision, a margin around the quantization grid is required. So Eq.(4) is slightly deformed by adding a parameter $\epsilon \in (0, \frac{\Delta}{2})$ through

$$d_i' = \begin{cases} (Q_i + 1) \times \Delta - \epsilon & \text{if } Q_i\%2 = w_i \ \&\ \Delta - \epsilon < R_i \\ d_i & \text{if } Q_i\%2 = w_i \ \&\ \epsilon \le R_i \le \Delta - \epsilon \\ Q_i \times \Delta + \epsilon & \text{if } Q_i\%2 = w_i \ \&\ R_i < \epsilon \\ (Q_i + 1) \times \Delta + \epsilon & \text{if } Q_i\%2 \neq w_i \ \&\ \Delta - \epsilon < R_i \\ d_i + 2 \times (\Delta - R_i) & \text{if } Q_i\%2 \neq w_i \ \&\ \frac{\Delta}{2} \le R_i \le \Delta - \epsilon \\ d_i - 2 \times R_i & \text{if } Q_i\%2 \neq w_i \ \&\ \epsilon \le R_i < \frac{\Delta}{2} \\ Q_i \times \Delta - \epsilon & \text{if } Q_i\%2 \neq w_i \ \&\ R_i < \epsilon \end{cases} \tag{5}$$

so that $d_i'\%\Delta \in (\epsilon, \Delta - \epsilon)$. As a result, the change of d_i' within $(-\epsilon, \epsilon)$ can be allowed without changing the embedded value w_i. An appropriate value should be assigned to ϵ without disclosing the quantization step Δ. If we choose the value of ϵ in proportional to Δ, $\frac{\Delta}{6}$ for instance, the allowed range can be adjusted by appropriately choosing the quantization step Δ. Consequently, the resulting d_i' is used to adjust the position of v_i by

$$v_i' = v_{ic} + (v_i - v_{ic}) \times \frac{d_i'}{d_i}, \tag{6}$$

where v_i' is the adjusted vertex position. At each iteration, to embed one bit value, the position of the newly traversed vertex is adjusted to modulate the distance from it to the centroid of its traversed neighbors. So the number of the embedded bits is equal to the number of the adjusted vertices. Given m vertices in the cover mesh, there will be $m - 1$ bit values embedded after the position of the last traversed vertex is adjusted so that the watermarked mesh is generated. After that, the position of mesh centroid is calculated by

$$v_c = \frac{1}{m} \sum_{i=1}^{m} v_i', \tag{7}$$

and the distance from the last traversed vertex v_l to the mesh centroid is calculated by

$$D = \sqrt{(v_{lx} - v_{cx})^2 + (v_{ly} - v_{cy})^2 + (v_{lz} - v_{cz})^2}. \tag{8}$$

The ratio R between D and Δ is obtained by

$$R = D/\Delta, \tag{9}$$

which will be used in the retrieval process to calculate the quantization step Δ.

2.2 Message Retrieval

To retrieve the embedded data from the watermarked mesh, the quantization step Δ used in watermark embedding is required. To obtain Δ, the distance D from the last traversed vertex to the mesh centroid is required, besides the

parameter R. Since the mesh traversal is ordered by the permuted vertex indices I' and face indices F', the secret key K is required to generate them. Therefore, the secret key K and the parameter R are used as the inputs of the retrieval process, besides the watermarked mesh.

The detailed process of watermark retrieval is as follows: At first, the vertex indices I and face indices F in the watermarked mesh are permuted by using K as the seed of pseudo-random generator to generate I' and F', respectively. By performing the mesh traversal, the distance from a vertex to the centroid of its traversed neighbors can be calculated by using Eq.(1) and Eq.(2). If the watermarked mesh is intact, the obtained distances are those that have been modified in the embedding process, i.e., $\{d'_1, d'_2, \ldots, d'_{m-1}\}$, given m vertices in the watermarked mesh. With the distance D from the last traversed vertex v_l to the mesh centroid calculated by Eq.(7) and the provided parameter R, the quantization step Δ is obtained by

$$\Delta = D/R. \tag{10}$$

With the obtained Δ, the bit value w'_i is extracted by

$$w'_i = \lfloor d'_i/\Delta \rfloor \% 2. \tag{11}$$

The whole message string $W' = (w'_i)_{i=1}^{m-1}$ will be retrieved after the last bit is extracted from the last traversed vertex.

2.3 The Properties of The Embedded Data

Since the ratio between any two distances in a polygonal mesh is invariant to RST transformations, while the quantization step used in the retrieval process is proportional to the distance from the last traversed vertex to the mesh centroid, the ratio between the distance from a vertex to the centroid of its traversed neighbors and the quantization step remains the same after RST transformations, as well as the embedded watermark. After topological modifications that change the neighboring information between vertices, the mesh traversal in the retrieval process will be different from that in the embedding process so that the embedded watermark cannot be correctly retrieved. Therefore, the embedded data is sensitive to the modifications made to the connectivity between vertices.

As for the mantissa truncation of vertex coordinate, which is stored as a single-precision floating-point number, if the truncation error is distributed within $(-T, T)$, then the errors introduced to the coordinates of the mesh centroid in Eq.(7) and the centroid of a vertex's neighboring vertices in Eq.(1) are also distributed within $(-T, T)$. The error introduced to d'_i in Eq.(2) and D in Eq.(8) will be both distributed within $(-2\sqrt{3}T, 2\sqrt{3}T)$. Based on Eq.(10), we know the error introduced to Δ is within $(-\frac{2\sqrt{3}T}{R}, \frac{2\sqrt{3}T}{R})$ so that Eq.(11) can be rewritten as

$$w'_i = \lfloor \frac{d'_i + \delta d}{\Delta + \delta_1} \rfloor \% 2, \tag{12}$$

where δd and δ_1 are the change of d'_i and Δ caused by the truncation, respectively. It can be seen the integer quotient $\lfloor \frac{d'_i+\delta d}{\Delta+\delta_1} \rfloor$ will be different from $\lfloor \frac{d'_i}{\Delta} \rfloor$ if $d'_i\%\Delta + \delta d - \lfloor d'_i/\Delta \rfloor \times \delta_1 \notin (0, \Delta)$. If Eq.(5) is used in the embedding process, $d'_i\%\Delta$ will be distributed within $(\epsilon, \Delta - \epsilon)$. As a result, the retrieved bit value w'_i in Eq.(12) will be identical to w_i, i.e. $\lfloor d'_i/\Delta \rfloor\%2$, if $|\delta d - \lfloor d'_i/\Delta \rfloor \times \delta_1| < \epsilon$. Since $\delta d \in (-2\sqrt{3}T, 2\sqrt{3}T)$ and $\delta_1 \in (-\frac{2\sqrt{3}T}{R}, \frac{2\sqrt{3}T}{R})$, the truncation of vertex coordinates is allowed if

$$T < \frac{\epsilon}{2\sqrt{3}(1 + \frac{\lfloor d'_M/\Delta \rfloor}{R})}, \tag{13}$$

where d'_M is the greatest one among all the modified distance $\{d'_1, d'_2, \ldots, d'_{m-1}\}$. On the other side, truncation of vertex coordinates can be allowed by appropriately choosing the quantization step Δ if $\lfloor d'_M/\Delta \rfloor < (\frac{\epsilon}{2\sqrt{3}T} - 1)R$, or $\Delta > \frac{d'_M}{(\frac{\epsilon}{2\sqrt{3}T}-1)R}$ since $\epsilon > 2\sqrt{3}T$ as indicated by Eq.(13). If the parameter ϵ in Eq.(5) is assigned proportional to the quantization step Δ (we take $\frac{\Delta}{6}$ for instance), the value of Δ should be chosen so that $\Delta > \frac{d'_M}{(\frac{\Delta}{12\sqrt{3}T}-1)R}$, i.e.,

$$\Delta > 6\sqrt{3}T + \sqrt{108T^2 + \frac{12\sqrt{3}Td'_M}{R}}, \tag{14}$$

where the value of d'_M and R are obtained from the watermarked mesh. Otherwise, the embedded value will probably be altered.

For the geometrical modifications that take place on part of the vertices, we take for instance the case that one vertex is modified. The distance d'_i from the modified vertex to its traversed neighbors will be changed by the modification as denoted by $d'_i + \delta d_i$ with δd_i as the change. Suppose the quantization step Δ obtained from Eq.(10) is unchanged, the integer quotient $\lfloor \frac{d'_i+\delta d_i}{\Delta} \rfloor$ will be possibly changed if $|\delta d_i| > \epsilon$ given $d'_i\%\Delta \in (\epsilon, \Delta - \epsilon)$. For the untraversed neighbors of the modified vertex, i.e., those vertices regarding the modified vertex as their traversed neighbor, the distances from them to their traversed neighbors will also be changed by the modification so that the chance to detect the modification is increased. In summary, if one vertex is modified outside the allowed range, the data embedded by adjusting the positions of itself and its untraversed neighbors will probably be altered.

3 Experimental Results

We performed the proposed method on several mesh models as listed in Table 1, where the capacity of each mesh model is also given. Suppose the precision interval of vertex coordinates is $(-T, T)$, an appropriate quantization step Δ should be chosen as in Eq.(14) if $\frac{\Delta}{6}$ has been assigned to the parameter ϵ in Eq.(5). The runtime of the embedding and retrieval processes for the "teapot" model were only 0.750 and 0.875 seconds in a 2.66G Pentium 4 PC with 512MB RAM, while those for the "horse" model were 40.844 and 44.438 seconds, respectively.

Table 1. The mesh models used in the experiments

Model	Meshes	Vertices	Polygons	Capacity(bits)
fish	1	742	1408	741
teapot	5	1631	3080	1626
dog	48	7616	13176	7568
wolf	90	8176	13992	8086
horse	31	10316	18359	10285

3.1 Distortion of the Cover Mesh

In the experiments, the impact of the embedding process can be tuned by the quantization step Δ used. From Eq.(6), it can be seen that the adjustment of each vertex position is within the sphere with its original position as the centroid while Δ as the radius, since the change of the distance from a vertex to its traversed neighbors is bounded by $(-\Delta, \Delta)$. Upon the fact that the mesh topology has not been changed, the distance from the adjusted vertex to its former position is used to represent the distortion of the mesh content. In the experiments, if 0.01 was assigned to Δ, the greatest error (i.e., the greatest distance among all the adjusted vertices) never exceeded 0.01, while the greatest error was below 0.001 if 0.001 had been assigned to Δ, as shown in Fig. 1. The pictures rendered from the mesh models "teapot" and "horse" before and after the embedding process are shown in Fig. 2.

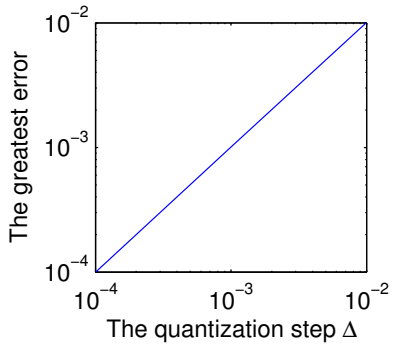

Fig. 1. The greatest error increases with the quantization step.

3.2 Capacity

The proposed method is applicable to 3D polygonal meshes with arbitrary connectivity. Given m vertices in the cover mesh, the capacity of our method will be $m - 1$ bits, tending to 1 bit/vertex when m is sufficiently large. If a mesh

(a) The original mesh model "teapot"

(b) The "teapot" model with 1626 bits embedded

(c) The original mesh model "horse"

(d) The "horse" model with 10285 bits embedded

Fig. 2. 1626 and 10285 bits in total are hidden within the mesh model "teapot" and "horse", respectively, by choosing $1/10,000$ of the greatest distance D_m from a vertex to the mesh centroid as the quantization step Δ and $\frac{\Delta}{6}$ as the parameter ϵ.

model consists of l separate meshes as in Table 1, the capacity will be $m-l$ bits since the first indexed vertex within each mesh is traversed without adjusting its position.

3.3 Security

The security of the proposed method relies on the secrecy of the key K, as well as the parameter R, which is used to calculate the quantization step Δ in the retrieval process. Given there are m vertices and p polygons in a polygonal mesh, the permutation of the vertex indices is $m!$. Without the secret key K, the mesh traversal must be performed $pm!$ times to guarantee the embedded data can be correctly retrieved, given the accurate quantization step Δ. To make it hard to estimate the quantization step Δ from the set of modified distances, the parameter ϵ used in Eq.(5) should be assigned with a relatively small value, $\frac{\Delta}{6}$ for instance. Moreover, we define the embedding primitive over the neighborhood

Table 2. By assigning $1/100,000$ of the greatest distance from a vertex to the mesh centroid to the quantization step Δ and $\frac{\Delta}{6}$ to the parameter ϵ, the NC values are calculated from the extracted bit values and the original ones after the watermarked mesh have been processed by the following manipulations, respectively.

Meshes	RST transformations	Moving two vertices oppositely	Modifying one vertex position	Reducing one face	Truncating five LSBs	Truncating six LSBs
fish	1.0000	0.9932	0.9959	0.9757	1.0000	0.9838
teapot	1.0000	0.9963	0.9987	0.7915	1.0000	0.9907
dog	1.0000	0.9980	0.9984	0.5776	0.9988	0.9912
wolf	1.0000	0.9993	0.9997	0.6070	0.9991	0.9881
horse	1.0000	0.9997	0.9999	0.5402	0.9994	0.9860

of a vertex so that resistance to substitution attacks is achieved, which makes it even harder to construct a counterfeit mesh with the same watermark.

3.4 Authentication of Polygonal Meshes

We try to apply the proposed method to authentication of polygonal meshes. To detect the illegal modifications made to the watermarked mesh and estimate its strength, the retrieved watermark $W' = (w'_i)_{i=1}^N$ is compared with the original one $W = (w_i)_{i=1}^N$ by defining a numerical value NC over them

$$NC = \frac{1}{N}\sum_{i=1}^{N} I(w'_i, w_i), \tag{15}$$

with

$$I(w'_i, w_i) = \begin{cases} 1 \text{ if } w'_i = w_i \\ 0 \text{ otherwise} \end{cases}. \tag{16}$$

The value of NC is expected to be less than 1 if the mesh content has been illegally modified.

The watermarked mesh model went through RST transformations, changing the positions of two vertices oppositely (respectively by adding the vectors $\{2\Delta, 2\Delta, 2\Delta\}$ and $\{-2\Delta, -2\Delta, -2\Delta\}$), modifying one vertex position by adding the vector $\{3\Delta, 3\Delta, 3\Delta\}$, reducing one face from the mesh, and truncating the least significant bits (LSB) of each vertex coordinate, respectively. By retrieving the embedded bit values from the processed mesh models and comparing them with the original ones by using Eq.(15), the resulting values of NC listed in Table 2 indicated that the embedded data was robust against RST transformations and truncation of vertex coordinates to a certain degree, but sensitive to other modifications. It should be noted the allowed range of coordinate truncation could also be adjusted with the quantization step Δ. If $1/10,000$ of D_m, which is defined as the greatest distance from a vertex to the mesh centroid, was assigned to Δ, truncating of 8 least significant bits (LSB) of vertex coordinate was allowed for the "teapot" model. While $1/100,000$ of D_m was assigned

instead, only 5 LSBs of vertex coordinate could be truncated without changing the embedded data.

From the obtained NC values, it can be seen the illegal modifications made to the watermarked mesh can be classified into severe and slight ones. Topological and severe geometrical modifications may lead the retrieved watermark to be dramatically different from the original one, while those geometrical modifications that have little impact on the quantization step Δ are possible to be localized by comparing the extracted values with the original ones. For a vertex where the two values do not match, its position or those of its previously traversed neighbors might have been changed. Normally, the number of the previously traversed neighbors of a vertex is very limited so that this type of modification can be localized. In our experiments, the watermarked mesh model "teapot" in Fig. 2(b) was tampered by modifying one vertex on its handle and the tampered mesh model is shown in Fig. 3(a). The illegal modification is detected by comparing the extracted watermark with the original one so as to find the region where the two values do not match, as shown in Fig. 3(b).

(a) The tampered mesh model "teapot"

(b) The mesh model with the tampered region detected

Fig. 3. The mesh model in Fig. 2(b) is tampered by modifying one vertex and the tampered region has been localized.

4 Concluding Remarks and Future Works

A high-capacity data hiding method has been proposed for polygonal meshes by choosing the distance from a vertex to the centroid of its traversed neighbors as the embedding primitive. A new quantization method has been employed to embed a watermark by slightly modifying the embedding primitives. It is hard to estimate the quantization step from the modified primitives, while slight change of them can be allowed to a certain degree by reserving a margin around the quantization grid. The embedded data is robust against those content-preserving manipulations, such as RST transformations and truncation of vertex coordinates to a certain degree, but sensitive to malicious manipulations. The impact of embedding on the mesh content can be tuned by choosing an appropriate quantization step. In the future, we will further investigate on: (1) the information of the embedded data leaked from the watermarked mesh, if any; and (2) the attacks to the proposed method for authentication and secret message communication.

Acknowledgment

The authors would like to sincerely thank the three anonymous reviewers for their valuable comments and insightful suggestions.

References

1. M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues," *Proc. ACM SIGGRAPH*, pp. 131-144, 2000.
2. The protein data bank, http://www.rcsb.org/pdb/.
3. F.A.P. Petitcolas, R.J. Anderson and M.G. Kuhn, "Information hiding-A survey," *Proc. of the IEEE*, vol. 87(7), pp. 1062-1078, July, 1999.
4. B. L. Yeo and M. M. Yeung, "Watermarking 3-D objects for verification," *IEEE Comput. Graph. Applicat.*, pp. 36-45, Jan./Feb. 1999.
5. H. Y. S. Lin, H. Y. M. Liao, C. S. Lu and J. C. Lin, "Fragile Watermarking for Authenticating 3-D Polygonal Meshes," *IEEE Transaction on Multimedia*, Vol. 7, No. 6, pp. 997-1006, 2005.
6. O. Benedens and C. Busch, "Toward blind detection of robust watermarks in polygonal models," *Proc. EUROGRAPHICS Comput. Graph. Forum*, vol. 19, pp. C199-C208, 2000.
7. F. Cayre and B. Macq, "Data hiding on 3-D triangle meshes," *IEEE Trans. Signal Processing*, vol. 51, pp. 939-949 (4), 2003.
8. H. T. Wu and Y. M. Cheung, "A Fragile Watermarking Scheme for 3D Meshes," *Proc. ACM Multimedia & Security Workshop*, pp. 117-123, New York, 2005.
9. O. Benedens, "Geometry-based watermarking of 3-D models," *IEEE Comput. Graph., Special Issue on Image Security*, pp. 46-55, Jan./Feb. 1999.
10. E. Praun, H. Hoppe and A. Finkelstein, "Robust mesh watermarking," *Proc. ACM SIGGRAPH*, pp. 69-76, 1999.

11. R. Ohbuchi, H. Masuda and M. Aono, "Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 551-560, Apr. 1998.
12. R. Ohbuchi, S. Takahashi, T. Miyasawa and A. Mukaiyama, "Watermarking 3-D polygonal meshes in the mesh spectral domain," *Proc. Graphics Interface*, pp. 9-17, Ottawa, ON, Canada, June 2001.
13. H. Date, S. Kanai and T. Kishinami, "Digital watermarking for 3-D polygonal model based on wavelet transform," *Proc. ASME Des. Eng. Techn. Conf.*, Sept. 12-15, 1999.
14. M. G. Wagner, "Robust Watermarking of Polygonal Meshes," *Proc. Geometric Modeling & Processing 2000*, pp. 201-208, Hong Kong, April, 2000.
15. K. Yin, Z. Pan, J. Shi and D. Zhang, "Robust mesh watermarking based on multiresolution processing," *Computers & Graphics*, vol. 25, pp. 409-420, 2001.
16. A. Kalivas, A. Tefas and I. Pitas. "Watermarking of 3D Models using Principal Component Analysis," *Proc. ICASSP*, vol. 5, pp.676-679, 2003.
17. Y. Maret and T. Ebrahimi, "Data Hiding on 3D Polygonal Meshes," *Proc. ACM Multimedia & Security Workshop*, pp. 68-74, Magdeburg, Germany, 2004.
18. F. Uccheddu, M. Corsini, and M. Barni, "Wavelet-based blind watermarking of 3d models," *Proc. ACM Multimedia & Security Workshop*, pp. 143-154, Magdeburg, Germany, 2004.
19. T. Kalker, "Considerations on watermarking security," *IEEE Int. Workshop on Multimedia Signal Processing*, pp. 201-206, Cannes, France, October, 2001
20. The Web3D Consortium, http://www.vrml.org/