Research on watermark printing technology of STL model based on Menger curvature

Changhao Zhang¹, Hui Li**, Xianggang Chen¹ and Xiaolei Shi¹

¹School of Mechanical Engineering & Automation, Northeastern University, Shenyang, Liaoning, 110819, China
**E-mail: hhuli@mail.neu.edu.cn

Abstract. In order to solve the data protection problem of STL model, this paper proposes a 3D printing watermarking method based on Menger curvature and K-means clustering algorithm. By comparing the research contents of 3D mesh model watermarking algorithm and two 3D printing watermarking methods, the 3D printing watermarking method is applied based on Menger curvature and K-means clustering algorithm. The method flow, watermark embedding and watermark extraction are discussed. Finally, the experimental results show that this method can not only improve the invisibility of model watermark, but also be robust to geometric attacks such as rotation, scaling and translation.

1. Introduction

With the rapid development of core technologies such as cloud computing, the Internet of Things and computers, and the inevitable trend of social development, the 3D printing cloud service platform emerged as the times require. It is an online platform that meets user needs and provides diversified 3D printing industry services. However, the 3D model designed by the user in the 3D printing cloud platform is very likely to be plagiarized by others when it is transmitted over the network. In addition, after the production service provider receives the user-designed three-dimensional model, it may also be used or disseminated illegally. These situations are not conducive to the protection of creators’ intellectual property rights and seriously infringe on the personal interests of digital product owners. Therefore, the method of embedding the watermark in the model is suitable and necessary to protect the data protection of 3D printed digital goods [1].

The three-dimensional grid model watermarking technology based on the spatial domain and the transform domain extracts the watermark from the grid model embedded with the watermark. The watermarking algorithm in the spatial domain is to directly modify the geometric attributes or topological structure of the 3D model and embed the digital watermark. The earliest spatial domain watermarking algorithm is the Triangle Similarity Quadruple (TSQ) algorithm proposed by Ohbuchi et al [2] and the Tetrahedral Volume Ratio (TVR algorithm). The former uses the distance ratio on the triangle as embedding primitives. The latter uses the volume ratio of the tetrahedron formed by adjacent triangles as the embedding primitives. The design of these two algorithms is relatively simple, and they provide important reference value for the subsequent research of digital watermarking algorithms for 3D models. But the disadvantage of the algorithm lies in its weak robustness and can only resist attacks such as affine transformation, and is more sensitive to other common attacks.

Benedens et al [3] proposed a three-dimensional watermarking algorithm that adjusts the normal vector distribution of the grid surface in 1999. This method maps and groups the normal vectors, and achieves the purpose of embedding the watermark by finely adjusting the distribution of the normal vector.
vectors. Because these normal vectors can roughly describe the changing trend of the grid, this method is robust to grid simplification. Later, Benedens and others also proposed Affine Invariant Embedding (AIE) algorithm and Normal Bin Encoding (NBE) [4] algorithm. The former mainly uses affine invariants as embedding primitives to achieve robustness against affine transformation attacks such as translation, rotation and uniform scaling. The latter is mainly a further improvement of the author’s previous method of adjusting the normal vector, which makes it more robust to grid simplification.

The method used in this article is to scan the product firstly, then extract the watermark from the constructed 3D mesh model, and finally calculate the correlation with the original watermark, thereby verifying the source of the product model, and then verifying whether a product has violated the owner of the digital product copyright. Obviously, the objects of the two methods to extract the watermark are different. Although the latter will bring greater errors, in order to solve the defect that the 3D mesh model watermark is not suitable for the watermark of 3D printing products, and reduce the previous 3D printing watermark method risk, a 3D printing watermark method based on Menger curvature is proposed.

2. **K-means clustering algorithm based on Menger curvature**

2.1. **Menger curvature algorithm**

The input of 3D printing is an STL file, which is a three-dimensional triangle mesh model. The three-dimensional triangle mesh model is composed of a series of triangular facets, and each facet includes three vertices and a normal vector, and each vertex is represented by three coordinates $x$, $y$, and $z$. The structure diagram of the small plane is shown in Figure 1. Menger curvature is the curvature of a three-phase n-dimensional Euclidean space. The Menger curvature of the facet is calculated by equation (1) as follows:

$$K_m = \frac{1}{R} = \frac{4A}{a \times b \times c}$$

$K_m$ is the Menger curvature, $A$ is the area of the facet, $R$ is the radius of the circumscribing circle of the facet, and $a$, $b$, and $c$ are the lengths of the three sides of the facet. According to equation (1), we can conclude that the Menger curvature of the facet depends on the radius of the circumscribing circle of the facet or the length of the three sides of the facet.

![Figure 1. Schematic diagram of the facet structure.](image)

Since the output of 3D printing is a physical 3D printed object, it is necessary to extract them from the scanned 3D triangular mesh model of the physical 3D printed object after the 3D scanning and reconstruction process when extracting the embedded watermark. Since the 3D scanning process is affected by noise, the vertex coordinates of each triangle in the 3D mesh model obtained by scanning are different from the vertex coordinates in the original 3D mesh model. However, the overall shape of the three-dimensional mesh model remains unchanged, and the area or length of the edge of the facet does not change, which causes the Menger curvature of the facet to remain unchanged or change little. Therefore, the Menger curvature of the facet is robust to geometric attacks such as rotation and translation. If the facet is rotated or translated, the area or length of the edge of the facet will not change.
2.2. *K-means clustering algorithm*

K-means clustering algorithm, as a centroid-based partitioning technology, has been widely accepted by people due to its simple and easy-to-understand principles, fast convergence speed, and high execution efficiency, and has become a common method in partition-based clustering models.

The basic principle of the algorithm is shown as follows. Firstly, the data set $X$ is known, and the total number of clusters $k$ is specified in advance, and $k$ objects are randomly selected from $X$ as the initial cluster centers. Then use the *Euclidean distance* as the criterion to allocate the data objects until all the data set objects are allocated. At this time, the entire data set is divided into $k$ sub-categories, each of which is called a cluster.

Then take the mean of all data observations in each cluster as the updated cluster centroid, and re-assign the data objects to the cluster with the closest distance according to the *Euclidean distance* criterion. The entire K-means clustering process requires multiple iterations until the distribution is stable, that is, the cluster formed in the last time is the same as the cluster formed in the previous time.

Evaluating the quality of a clustering result is often expressed by the intra-cluster variation $J$, as shown in equation (2).

$$ J = \sum_{k=1}^{k} \sum_{i=1}^{n} (x_i - u_k)^2 $$

Where $J$ refers to the intra-cluster variation, which is the sum of the squared errors between all data observations in the data set $X$ and the cluster centroid $u_k$ where the data point is located. The input of the K-means clustering algorithm is the data set $D$ and the number of clusters $k$, and the output is the clustering result of $k$ clusters. The basic flow chart of the K-means clustering algorithm is shown in Figure 2.

![Figure 2. Basic flowchart of K-means clustering algorithm.](image)

The K-means clustering algorithm is easy to implement without parameter estimation and training. And it is not sensitive to outliers (individual noise data does not have a great impact on the results), suitable for classifying rare events, and suitable for multi-classification problems.
However, the K-means clustering algorithm has a large amount of calculation when classifying the test sample, and the memory overhead is large, because for each text to be classified, the distance to all the known samples must be calculated to find its $K$ nearest neighbours. The current common solution is to edit the known sample points in advance, and remove the samples that have little effect on classification in advance.

The selection of $K$ value is uneven. The biggest disadvantage is that when the sample is unbalanced, such as the sample size of one class is very large, while the sample size of other classes is very small, it may lead to the $K$ neighbours of the sample account for the majority when a new sample is input. The algorithm only calculates the ‘nearest’ neighbour samples. If the number of samples of a certain type is large, then either such samples are not close to the target sample, or such samples are very close to the target sample. In any case, the quantity does not affect the results of the operation.

3. Experimental realization

3.1. Method flow

The specific process of the 3D printing watermark method based on Menger curvature and K-means clustering algorithm is shown as follows. Firstly, extract the triangular faces from the 3D printing model (STL model) and calculate the Menger curvature of each facet. Then, the user defines or selects the number of facet classification groups, which is the value of the watermark key. The watermark key is used to encrypt the watermark information, which is secret and reused in the watermark embedding and extraction process. Take the value of Menger curvature of each facet as the data set and the number of facet classification groups as the number of clusters. These facets are grouped by K-means clustering algorithm. For each group of facets, calculate the average Menger curvature of the group of facets, and then change the value of the average Menger curvature by referring to a special value, thereby embedding the watermark bit into the average Menger curvature of the group of facets. Finally, a watermarked three-dimensional mesh model is generated according to the average Menger curvature with watermarked information. Input the watermarked 3D mesh model to the 3D printer.

![Flow chart of the proposed method.](image)

After the 3D printing process, the 3D printed physical object is scanned and reconstructed in 3D to obtain a three-dimensional mesh model firstly, and then the facets of the scanned three-dimensional mesh model are extracted to calculate its Menger curvature. Next, the data set based on the value of Menger curvature is also used to group the small planes of the scanned 3D mesh model with the number
of facet classification groups as the number of clusters. After the facet clustering step, the average Menger curvature of each group is calculated. The watermark information will be extracted from the average Menger curvature of each group by referring to the special value. The flowchart of the proposed method is shown in Figure 3.

3.2. Watermark embedding
Suppose a certain STL model $M$ contains a series of triangles, which is $M = \{F_i | i \in [1, |M|]\}$. $|M|$ is the number of triangles of the STL model $M$, and $F_i$ is the $i$-th triangle. Each face includes three vertices $F_i = \{V_j | j \in [1, 3]\}$ and a normal vector $n_i (nx_i, ny_i, nz_i)$. The Menger curvature $K_i$ of each face $F_i$ is calculated by its vertex and corresponding area, as shown in formula (3):

$$K_i = \frac{4A_i}{||V_{i1} - V_{i2}|| \times ||V_{i2} - V_{i3}|| \times ||V_{i3} - V_{i1}||}$$

Among them, $A_i$ is the area of the triangle and $V$ is the vertex coordinates. Divide the $|M|$ triangular faces of the STL model into $G$ groups, assuming that $m_g$ is the set of the group $g$ of triangular faces, then $M = \{m_g | g \in [1, |G|]\}$. Figure 4 shows the result of face clustering based on the Menger curvature of the rabbit triangle mesh. Figure (a) is the original rabbit triangle mesh while Figure (b) is the result of facet clustering based on Menger curvature. Facets in the same group have the same colour.

![Figure 4. Rabbit model original map and post-processing comparison chart.](image)

After the $|M|$ triangles are divided into $G$ groups, find the facet with the largest Menger curvature and the facet with the smallest Menger curvature in each group, and calculate the average Menger curvature of each group. Suppose that $K_{\max}^{m_g}$, $K_{\min}^{m_g}$, $K_{\text{mean}}^{m_g}$ are the minimum Menger curvature, the maximum Menger curvature, and the average Menger curvature in the group $m_g$, while the average Menger curvature refers to the average value of the Menger curvature of all triangles in the group $m_g$. The calculation is shown in equation (4), while $|m_g|$ is the number of triangles in the group $m_g$. The definition of $\Delta_{mg}$ is the average value of $K_{\max}^{m_g}$ and $K_{\min}^{m_g}$, as shown in formula (5). $\Delta_{mg}$ is the special value mentioned above, and its purpose is to change the average Menger curvature.

$$K_{\text{mean}}^{m_g} = \frac{\sum K_i \in m_g}{|m_g|}$$

$$\Delta_{mg} = \frac{K_{\max}^{m_g} + K_{\min}^{m_g}}{2}$$
The watermark information is a user-defined binary code whose length is the number of packets. It is composed of a series of watermark bits, $\omega_g \in \{0,1\}\{g \in \left[1,|G|\right]\}$. On the basis of referring to the special value $\Delta_{mg}$ of each group, the average Menger curvature value of the group is changed by the value of the embedded watermark bit $\omega_g$. This shows that the $K_{mean}^{m_g}$ will be converted to $K_{mean}^{m_g^*}$ smaller than $\Delta_{mg}$ when $\omega_g=0$. If $\omega_g=1$, $K_{mean}^{m_g}$ will be converted to $K_{mean}^{m_g^*}$ larger than $\Delta_{mg}$. The expression formula of $K_{mean}^{m_g^*}$ is shown in formula (5). In order to meet the above embedding conditions, the average Merger curvature $K_{mean}^{m_g^*}$ of the watermark must be corrected, as shown in equation (6), equation (7) and equation (8).

\[
K_{mean}^{m_g^*} = \begin{cases} 
K_{mean}^{m_g^*} > \Delta_{mg} & \text{if } \omega_g = 1 \\
K_{mean}^{m_g^*} < \Delta_{mg} & \text{if } \omega_g = 0 
\end{cases}
\]

(6)

if $\omega_g = 1$, $K_{mean}^{m_g^*} = \begin{cases} 
\Delta_{mg} - \frac{K_{mean}^{m_g^*} - K_{mean}^{m_g}}{2} & \text{if } K_{mean}^{m_g} < \Delta_{mg} \\
K_{mean}^{m_g^*} & \text{if } K_{mean}^{m_g} > \Delta_{mg} 
\end{cases}
\]

if $\omega_g = 0$, $K_{mean}^{m_g^*} = \begin{cases} 
\Delta_{mg} - \frac{K_{mean}^{m_g^*} - K_{max}^{m_g}}{4} & \text{if } K_{mean}^{m_g} > \Delta_{mg} \\
K_{mean}^{m_g} & \text{if } K_{mean}^{m_g} < \Delta_{mg} 
\end{cases}
\]

(7)

Figure 5. Embedding the watermark bit by changing the average Menger curvature.

Figure 5 shows the process of changing the Menger curvature $K_{mean}^{m_g}$ of the group $m_g$ to $K_{mean}^{m_g^*}$ according to the watermark position $\omega_g$. Menger curvature $K_{mean}^{m_g}$ is represented by blue dots, and watermarked Menger curvature $K_{mean}^{m_g^*}$ is represented by red dots. When $\omega_g=0$, if $K_{mean}^{m_g}$ is greater than or equal to $\Delta_{mg}$, then $K_{mean}^{m_g}$ will become smaller than $\omega_{mg}$. When $\omega_g=1$, if $K_{mean}^{m_g}$ is smaller than $\Delta_{mg}$, then $K_{mean}^{m_g}$ will become greater than $\Delta_{mg}$. 
After the watermark bit $g_\omega$ is embedded into the average Menger curvature of the group $m_g$, the change rate $\alpha_g$ is calculated by the average Menger curvature $K_{\text{mean}}^{m_g}$ after adding the watermark and the reference value $\Delta_{mg}$, as shown in equation (9).

$$\alpha_g = \frac{K_{\text{mean}}^{m_g} - \Delta_{mg}}{\Delta_{mg}}$$  \hspace{1cm} (9)

$$v'_j = \alpha_g \times v_j + (\alpha_g - 1) \times v_j \forall j \in [1,3]$$  \hspace{1cm} (10)

According to the change rate $\alpha_g$ of the average Menger curvature of each group $m_g$ of watermarked models, a watermarked three-dimensional triangle mesh model $M'^\omega$ is generated. Firstly, select a facet whose Menger curvature value is closest to the average Menger curvature $K_{\text{mean}}^{m_g}$ of the watermarked facet in each group $m_g$, and then correct the rate of change $\alpha_g$ according to the actually selected facet.

Assuming that in the group $m_g$, we find that the Merger curvature of the facet $f_i$ is the facet closest to the average Menger curvature $K_{\text{mean}}^{m_g}$ of the watermarked model, and the facet $f'_i$ in the watermarked three-dimensional angular grid $M'^\omega$ will be transformed into facet $f'_i$. Each remaining group of facets closest to the average Menger curvature $K_{\text{mean}}^{m_g}$ of the watermarked model is converted in turn. The conversion formula is shown in formula (10), while $v_j'$ are the three vertices of the facet $f'_i$.

### 3.3. Watermark extraction

The watermark extraction process is similar to the embedding process. Firstly, the surface is extracted from the scanned three-dimensional triangle mesh $M$ to calculate the Menger surface curvature. Then, they are classified according to the clustering algorithm based on Menger curvature and K-means. The watermark key is reused in the clustering process. For each group, find the maximum Menger curvature $K_{\text{max}}^{m_i'}$ and the minimum Menger curvature $K_{\text{min}}^{m_i'}$, and calculate the average Menger curvature $K_{\text{mean}}^{m_i'}$ according to equation (4). $\Delta_{mg} = \frac{1}{2}(K_{\text{min}}^{m_i'} + K_{\text{max}}^{m_i'})$ is the average of $K_{\text{min}}^{m_i'}$ and $K_{\text{max}}^{m_i'}$. Finally, the watermark bit $g_\omega$ can be extracted by comparing the average Menger curvature $K_{\text{mean}}^{m_i'}$ with the average value $\Delta_{mg}'$, as described in formula (11).

$$\omega_g = \begin{cases} 1 & \text{if } K_{\text{mean}}^{m_i'} \geq \Delta_{mg}' \\ 0 & \text{if } K_{\text{mean}}^{m_i'} \leq \Delta_{mg}' \end{cases}$$  \hspace{1cm} (11)

### 4. Experimental results and analysis

We experimented with the proposed method by using the test model and print model shown in Figure 6. The format of the three-dimensional triangle mesh is the STL file format. The K-means algorithm is used to divide facet clustering into several groups, and the watermark key is defined by the user to determine the number of groups. Each model will correspond to a watermark key. These watermark keys are stored in the database and will be queried during watermark extraction, as shown in Figure 3. The number of groups must always be less than half of the number of facets. In order to meet the above conditions, in the experiment, we determine the number of watermark keys $G$ according to the number of facets $|M|$, as shown in formula (12).
\[ G = \text{Integerpart} \left( \frac{|M|}{2^S} \right) \]  

(12)

\( S \) is the number digits of \(|M|\), for example, if \(|M|=2425\), then \( S = 4 \). In order to evaluate the proposed method, the information protection performance of watermark embedding is evaluated.

![Figure 6. Test model and print model.](image)

The watermark information is embedded into the test model according to the number of groups, where the length of the watermark information is equal to the number of groups corresponding to the model. In order to evaluate the invisibility of the proposed method, the average distance error \( d^n(\nu, \nu') \) between the original 3D triangle mesh and the watermarked 3D triangle mesh is calculated. The average distance error \( d^n(\nu, \nu') \) is calculated by equation (13). Among them, \( \nu \) and \( \nu' \) are respectively the vertices of the original 3D triangle mesh and the vertices of the watermarked 3D triangle mesh.
Table 1 shows the calculation result of the average distance error between the watermarked three-dimensional triangular mesh of the test model and the original three-dimensional triangular mesh in Figure 6. The test model in Figure 6 is divided into 19 to 277 groups according to the number of facets, and the average distance error is calculated to be $4.0952 \times 10^{-6}$ to $3.1274 \times 10^{-6}$. This proves that the difference between the watermarked 3D triangular mesh and the original 3D triangular mesh is very small. Therefore, the invisibility of this method is proved to be high.

\[
d^m(\nu, \nu') = \frac{1}{3|M|} \sum_{i=1}^{n} \sum_{j=1}^{j} \|v_{ij} - v_{ij}'\| \tag{13}
\]

According to equation (13), the average distance error depends on the number of watermarked vertices and the number of vertices, while the number of vertices of the watermarked model depends on the number of groups. Therefore, it is concluded that the average distance error depends on the number of groups. From Table 1, it is concluded that the average distance error decreases as the number of groups increases. Figure 7 shows the relationship between the number of groups and the average distance error.
5. Conclusion
In this paper, we introduce a 3D printing watermarking method based on Menger facet curvature and K-means clustering algorithm. Experiments on the proposed method were carried out by using XYZ Printing 3D printer and XYZ 3D scanner. Experimental results show that this method can effectively prevent geometric attacks such as rotation, translation, and scaling.

The experimental results of XYZ Printing Pro 3D printer and 3D scanner also prove that the accuracy of this method is moderate and higher than the first two methods in the 3D printing field. Therefore, the proposed method can be applied to 3D printing watermarks. We can also provide a correction method during or after the 3D scanning process to improve the accuracy of the proposed method.

Acknowledgments
This paper is supported by National Key Research and Development Program of China (2017YFB1104200).

References
[1] Ohbuchi R, Masuda H, Aono M. (1997) Watermarking three-dimensional polygonal models. 16(4):261-272.
[2] Yang Xianyi, Niu Xinxin. (2006) Digital Watermarking Theory and Technology. Higher Education Press, 1-25.
[3] Wang K, Lavoue G, Denis F, et al. (2008) A Comprehensive Survey on Three-Dimensional Mesh Watermarking. IEEE Transactions on Multimedia, 10(8):1513-1527.
[4] Makkol N M, Khoo B E. (2013) Robust blind image watermarking scheme based on Redundant Discrete Wavelet Transform and Singular Value Decomposition. AEUE - International Journal of Electronics and Communications, 67(2):102-112.
[5] Ye Dengpan. (2005) Research on some algorithms of image authentication and video digital watermarking. Nanjing University of Science and Technology, Nanjing.
[6] Ohbuchi R, Masuda H, Aono M. (1998) Watermarking three-dimensional polygonal models through geometric and topological modifications. Selected Areas in Communications IEEE Journal on, 16(4):551-560.
[7] Benedens, Oliver. (1999) Geometry-Based Watermarking of 3D Models. Computer Graphics & Applications IEEE, 19(1):46-55.
[8] Benedens O, Busch C. (2000) Towards Blind Detection of Robust Watermarks in Polygonal Models. Computer Graphics Forum. 2000:199 – 208.
[9] Yu, Z. Ip H H S, Kwok L F. (2003) A robust watermarking scheme for 3D triangular mesh models. Pattern Recognition, 36(11):2603-2614.
[10] Hu Min, Xie Ying, Xu Liangfeng, et al. (2008) Adaptive 3D model watermarking algorithm based on geometric features. Journal of Computer Aided Design and Graphics, 20(3):390-394.
[11] Cho J W, Prost R, Jung H Y. (2007) An Oblivious Watermarking for 3-D Polygonal Meshes Using Distribution of Vertex Norms. IEEE Press.
[12] Liu Jing. (2017) Research on Digital Watermarking Algorithm of 3D Model Against Geometric Attack. Xi'an University of Technology, Xi'an.
[13] Wang Y P, Hu S M. (2009) A New Watermarking Method for 3D Models Based on Integral Invariants. IEEE Transactions on Visualization & Computer Graphics, 15(2):285.
[14] Kim K, Barni M, Tan H Z. (2010) Roughness-adaptive 3-D watermarking based on masking effect of surface roughness. IEEE Press.
[15] Sayahi I, Elkefi A, Amar C B. (2016) Blind watermarking algorithm based on spiral scanning method and error-correcting codes. Multimedia Tools & Applications, 76(15):1-24.
[16] Molaei A M, Ebrahimimnezhad H, Sedagahi M H. (2016) Robust and Blind 3D Mesh Watermarking in Spatial Domain Based on Faces Categorization and Sorting. 3d Research, 7(2):1-18.
[17] Soliman M M, Hassanien A E, Onsi H M. (2015) A Robust 3D Mesh Watermarking Approach Using Genetic Algorithms. 323:731-741.