ISSN 2278-3091
Volume 9, No.1, January – February 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse10912020.pdf
https://doi.org/10.30534/ijatcse/2020/10912020

# 3D Printing Watermarking Algorithm Based on 2D Slice Mean Distance

**Thang V. Tran[1], Giao N. Pham[2] (corresponding author), Binh A. Nguyen[3], Khoa D. Ta[4], and Phong H. Nguyen[5]**
[1]Institute of Engineering, Ho Chi Minh City University of Technology, Vietnam, tvthang74@gmail.com
[2]Dept. of Computing Fundamentals, FPT University, Hanoi, Vietnam, giaopn@fe.edu.vn
[3]ICT Department, FPT University, Hanoi, Vietnam, binhnase04865@fpt.edu.vn
[4]ICT Department, FPT University, Hanoi, Vietnam, khoatdhe130813@fpt.edu.vn
[5]Center of Machine Vision and Signal Analysis, University of Oulu, Finland, phong.nguyen@oulu.fi

## ABSTRACT

3D printing has recently been applied to many areas of life including education, healthcare, aerospace, automotive, industrial design and other domains in industry. Due to the fact that the benefit of 3D printing is significant, the copyright issue of 3D printing has been become essentially and importantly. To solve the copyright issue of 3D printing, we would like to propose a 3D printing watermarking algorithm based on the mean distance of 2D slices in this paper. In the proposed method, 3D printing model is sliced into a set of 2D slices, and each 2D slice is then embedded a bit of watermark data by changing the length of mean distance of that slice. Finally, the coordinates of points in each 2D slice will be altered according to the changed mean distance value of 2D slice which has been embedded watermark bit. Experimental results showed that the mean distance error between the watermarked 2D slices and the original 2D slices is very small. The bit error rate (BER) between the extracted watermark and the original watermark is also very low. So, we verify that the invisibility of the proposed method is effective, and the proposed method is also robust to geometric attacks as rotation, scaling, and translation. Comparing to the conventional works, the proposed method also provides a solution with better accuracy.

**Key words:** Additive Layer Manufacturing, 3D Printing Watermarking, 3D Slicing, Mean Distance, and Copyright for 3D Printing

## 1. INTRODUCTION

Three dimensional (3D) printing is a process of making 3D solid objects from a 3D printing model by cutting 3D printing model into 2D layers, and laying down successive layers of material [1]. 3D printing technology provides a solution to help us turn design ideas into reality. It allows everyone to become creators in many domains of life. So, 3D printing technology is a hot trend, properly exploited, able to completely change many things in life [2].

Due to the 3D printing process is very easy and low cost, many people could make 3D objects with this. Moreover, 3D printing can produce and replace many parts of industrial devices, medical devices thus it is used in many areas as scientific, medical, telecommunication, industrial and other domains [3]. But the production process of a 3D printing model is often consumed many resources and efforts while 3D printing models are easily copied and share without any permission from designers or providers. Therefore, manufacturer desires their products must be protected the copyright and ownership [4]. Consequently, a watermarking solution is essential to identify the ownership and copyright of 3D printing.

Due to the fact that users or sellers often search 3D printing models from free websites and they then print physical 3D objects, the aim of 3D printing watermarking is to embed watermark data into the files of 3D printing before the 3D printing process, and then verify the ownership or copyright of 3D printing files by extracting the embedded watermark data from the watermarked files. There are many watermarking methods proposed for 3D contents, 3D mesh and 3D animation. But previous methods are only suitable and useful for the copyright identification of virtual 3D contents. Because these method only extract watermark data from the watermarked 3D contents. Therefore, previous researches could not be applied to the copyright identification for 3D printing technology. For responding to the issues related to 3D printing copyright protection, in this paper we proposed a new watermarking algorithm for 3D printing based on the mean distance of 2D slices. The idea of the proposed algorithm is to embed watermark data into the mean distance of 2D slices. The proposed method is applied to the 2D slices of 3D printing file after the 3D slicing process. The proposed method did not used the geometric features of 3D model as shape, topology which are mentioned in previous method for 3D model watermarking in both spatial domain and frequency domain. It is difference between the proposed algorithm and the conventional works.

To understand the proposed method, our paper is organized as follow. In Sec. 2, the relation of 3D printing processing to the proposed method and previous watermarking techniques for 3D model will be explained. In Sec. 3, the proposed method will be presented in detail. Sec. 4 will describe the experimental results and the evaluation of the proposed method. Conclusion is explained in Sec. 5.

## 2. RELATED WORK

### 2.1 3D Model Watermarking

Since early 2000s, there are many proposed researches for 3D model watermarking [5]-[14]. The watermarking techniques for 3D models have been achieved in both the frequency and spatial domains. These methods are based on embedding watermark data into geometric features in the spatial domain as topology, shape or embedding watermark data into the spectrum features in the frequency domain as Discrete Cosine Transformation (DCT), Discrete Fourier Transformation (DFT) or Discrete Wavelet Transformation (DWT). Actually, the watermarking techniques for 3D models are only useful for the copyright identification of 3D content applications as 3D games and the applications of 3D graphic. This means the watermarking techniques for 3D model are only applied to the copyright protection of 3D contents in virtual environment. Consequently, they could not apply to 3D printing

### 2.2 3D Printing Based Watermarking

Up to present, in order to print physical 3D objects, 3D printers have to use 3D triangle meshes [15], [16]. 3D triangle meshes must be cut by a plane from bottom to top via the 3D slicing process before printing physical 3D objects. The 3D slicing process is executed by 3D slicer [17], [18]. The output of 3D slicing process is a set of 2D slices, and then 3D printer will print 3D objects from a set of 2D slices. Fig. 1 describes the general 3D printing process.

To get a scanned 3D triangle mesh of a 3D printed object, we have to pass the 3D scanning and reconstruction process. Due to the effect of noise in the 3D scanning process, the scanned 3D triangle mesh is not the same the original 3D triangle mesh. This is main reason leaded to the previous proposed watermarking methods for 3D model could not re-apply to 3D printing. But the overall shape of 3D triangle mesh is not changed after the 3D scanning and reconstruction process. Therefore, after the 3D slicing process, the 2D slices of the scanned 3D triangle mesh is similar the 2D slices of the original 3D triangle mesh. So, we used this feature to propose a 3D printing watermarking method based on embedding watermark data to 2D slices.

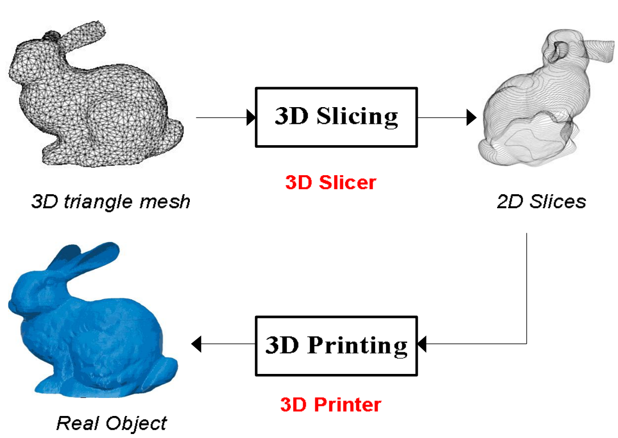



**Figure 1:** General 3D Printing Process

## 3. THE PROPOSED ALGORITHM

### 3.1 Overview

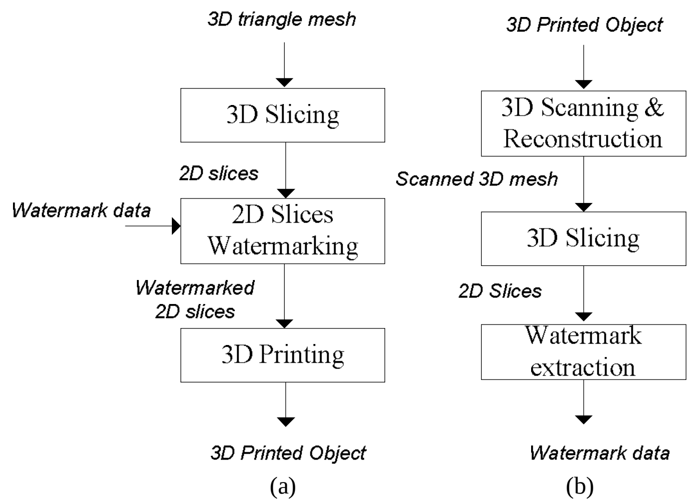



**Figure 2:** The proposed algorithm, (a) the process of watermark embedding, and (b) the process of watermark extraction.

The proposed algorithm is described in Fig. 2. First of all, 3D printing model is sliced into 2D slices from bottom to top by the 3D slicing process. 2D slices are the used to embed watermark data for obtaining the watermarked 2D slices. Each bit of watermark data is embed to each 2D slice by changing the mean distance length of that 2D slice on the reference of the minimum and maximum distances. After that, the coordinates of points in each 2D slice will be altered according to the changed mean distance length of 2D slice which has been embedded watermark bit. The watermarked 2D slices are then used in the 3D printing process to create 3D printed object as shown in Fig. 2a. The 3D printed object is used to scan and reconstruct the scanned 3D triangle mesh via the 3D scanning and reconstructing process. The scanned 3D triangle mesh will be sliced to obtain 2D slices. Finally, watermark data is extracted from the 2D slices of the scanned 3D triangle mesh as shown in Fig. 2b.

### 3.2 Watermark Embedding

Each 3D triangle mesh is cut from bottom to top by a 3D slicer to obtain a set of 2D slices. Each 2D slice contains the intersected points between 3D triangle mesh and the cutting plane. Thus each 2D slice contains one or many polygons. Each polygon contains the intersected points connected together. Fig. 3 shows the shape of 2D slices which is obtained from a bunny model. The watermark bit will be embedded to the mean distance length of each slice. The mean distance of slice is the average distance from the center of slice for its points. Fig. 4 show the distances of a slice of a bunny model from center to points.

Suppose, a 3D triangle mesh is cut into a set of 2D slices, $\mathbf{S} = \{S_i | i \in [1, \mathrm{N}]\}$ where **N** is the number of slices. Each 2D slice contained a set of the intersected points, $S_i = \{p_{ij} | j \in [1, |S_i|]\}$ where $|S_i|$ is the number of points in each 2D slice, and $p_{ij}$ is the $j^{th}$ point in the $i^{th}$ slice and is presented by a pair of coordinates $(x_{ij}, y_{ij})$. To calculate the mean distance length of each slice, we must calculate the center of each slice. Given $C_i(cx_i, cy_i)$ is the center of corresponding slice $S_i$, the coordinates of $C_i$ is the mean value of all coordinates as shown in Eq. (1) and Eq. (2)

$$cx_i = \frac{1}{|S_i|} \times \sum_{j=1}^{|S_i|} x_{ij} \quad (1)$$

$$cy_i = \frac{1}{|S_i|} \times \sum_{j=1}^{|S_i|} y_{ij} \quad (2)$$

Define $D_{mi}$, $d_{i,min}$ and $d_{i,max}$ is the mean distance of the slice $S_i$, the minimum distance and maximum distance in all distances from the center of the slice $S_i$ to points respectively.

Thus $D_{mi}$ is the mean distance of all distances from the center of slice to points as Fig. 4. The mean distance, minimum distance and maximum distance of slice is computed by equations (3), (4) and (5):

$$d_{i,min} = \mathrm{Min}\{d_{i,j} | j \in [1, |S_i|\} \quad (3)$$

$$d_{i,max} = \mathrm{Max}\{d_{i,j} | j \in [1, |S_i|\} \quad (4)$$

$$D_{mi} = \frac{1}{|S_i|} \times \sum_{j=1}^{|S_i|} d_{i,j} \quad (5)$$

With $d_{i,j} = \sqrt{(cx_i - x_{ij})^2 + (cy_i - y_{ij})^2}$ is the distance from the center of slice to $j^{th}$ point.

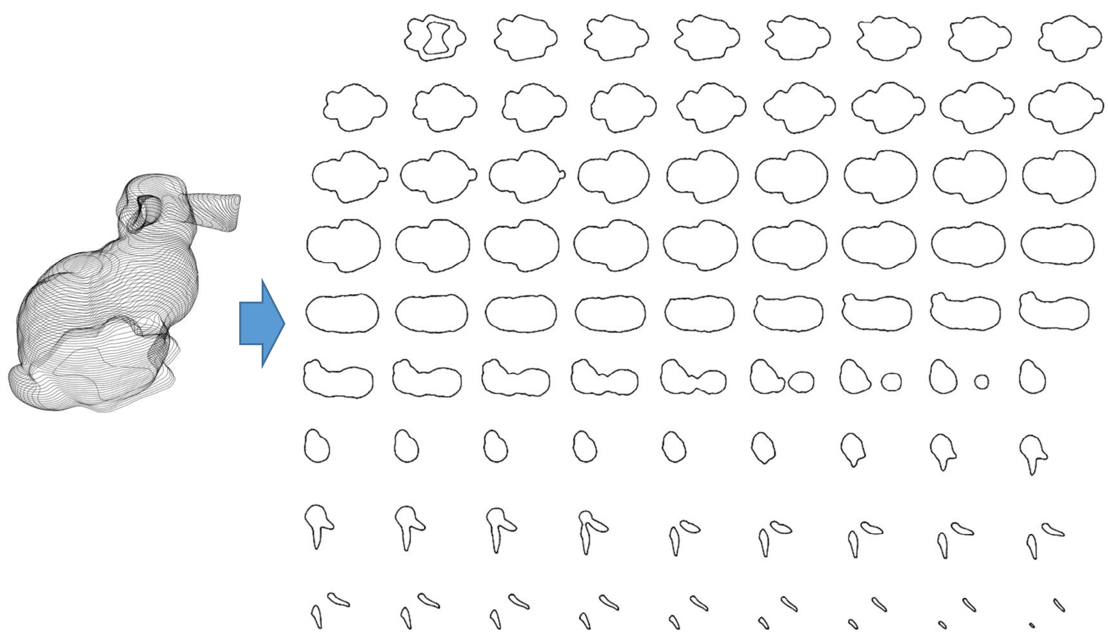


**Figure 3**: 2D slices of a 3D model in 3D space, and in 2D space

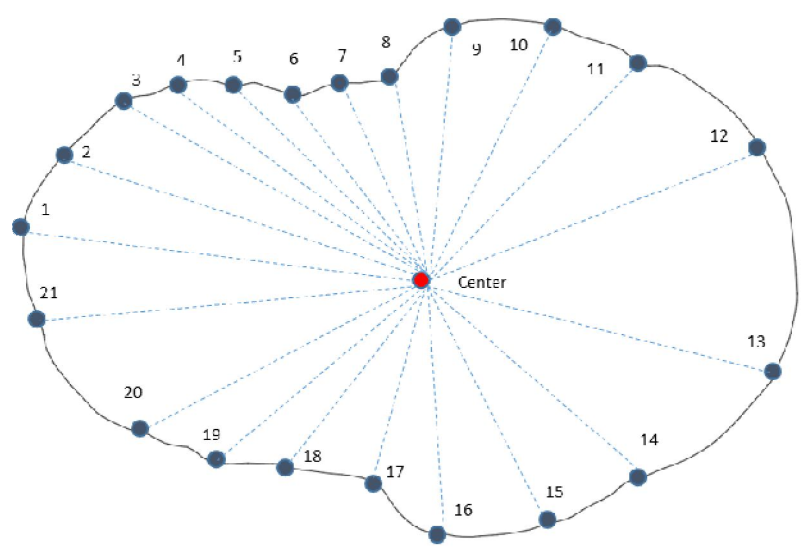



**Figure 4:** Distances from the center of a slice to points

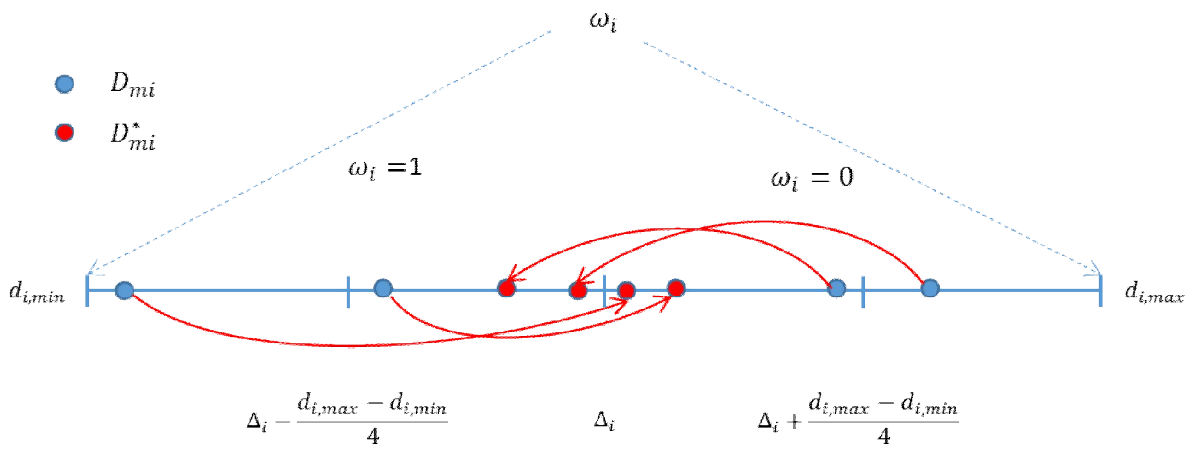



**Figure 5:** Embedding a watermark bit by changing the mean distance of slice

Next, we define $\Delta_i = (d_{i,min} + d_{i,max})/2$ is the average value of $d_{i,min}$ and $d_{i,max}$. Each watermark bit $\omega_i \in \{0,1\} (i \in [1, N])$ is embedded by changing the mean distance $D_{mi}$ to $D^*_{mi}$ on the reference of the average value $\Delta_i$ as Eq. (6). If $\omega_i = 0$, then $D_{mi}$ will be move to be less than $\Delta_i$. Otherwise, $D_{mi}$ will be move to be greater than $\Delta_i$:

$$D^*_{mi} = \begin{cases} D^*_{mi} > \Delta_i \; if \; \omega_i = 1 \\ D^*_{mi} < \Delta_i \; if \; \omega_i = 0 \end{cases} \tag{6}$$

To satisfy the above embedding condition, the watermarked mean distance $D^*_{mi}$ will be changed as follows:

If $\omega_i = 0$:

$$D^*_{mi} = \begin{cases} \Delta_i - \dfrac{D_{mi} - \Delta_i}{4} \quad \text{if } D_{mi} \in [\Delta_i, \Delta_i + \beta_i) \\ \Delta_i - \dfrac{d_{i,max} - D_{mi}}{4} \quad \text{if } D_{mi} \in [\Delta_i + \beta_i, d_{i,max}] \\ \text{No change if } D_{mi} < \Delta_i \end{cases} \tag{7}$$

If $\omega_i = 1$:

$$D^*_{mi} = \begin{cases} \Delta_i + \dfrac{D_{mi} - d_{i,min}}{4} \quad \text{if } D_{mi} \in [d_{i,min}, \Delta_i - \beta_i) \\ \Delta_i + \dfrac{d_{i,max} - D_{mi}}{4} \quad \text{if } D_{mi} \in [\Delta_i - \beta_i, \Delta_i] \\ \text{No change if } D_{mi} \geq \Delta_i \end{cases} \tag{8}$$

With $\beta_i = (d_{i,max} - d_{i,min})/4$. Fig. 5 describes the alteration of the mean distance of slice $D_{mi}$ according to the watermark bit $\omega_i$. The mean distance of slice $D_{mi}$ is represented by blue points. The watermarked mean distance $D^*_{mi}$ is represented by red points. When $\omega_i = 0$, $D_{mi}$ will be move to be less than $\Delta_i$ if it is equal or greater than $\Delta_i$. When $\omega_i = 1$, $D_{mi}$ will be move to be greater than $\Delta_i$ if it is less than $\Delta_i$.

The change rate $\alpha_i$ is calculated as described in Eq. (9) after embedding the watermark bit $\omega_i$ to the mean distance of the slice $S_i$:

$$\alpha_i = \frac{D^*_{mi}}{D_{mi}} \tag{9}$$

The change rate $\alpha_i$ is used to change all points of the slice $S_i$ as given in Equations (10) and (11).

$$x'_{ij} = \alpha_i . x_{ij} + cx_i(\alpha_i - 1) \tag{10}$$

$$y'_{ij} = \alpha_i . y_{ij} + cy_i(\alpha_i - 1) \tag{11}$$

$\forall j \in [1, |S_i|]$. Therein $(x'_{ij}, y'_{ij})$ are the coordinates of new point after the watermark data embedding process.

### 3.3 Watermark Extraction

The watermark extracting process is similar the watermark embedding process. From 2D slices, we calculate the center $C'_i(cx'_i, cy'_i)$ and mean distance $D'_{mi}$ of each slice as equations (12), (13) and (14):

$$cx'_i = \frac{1}{|S_i|} \times \sum_{j=1}^{|S_i|} x'_{ij} \tag{12}$$

$$cy'_i = \frac{1}{|S_i|} \times \sum_{j=1}^{|S_i|} y'_{ij} \tag{13}$$

$$D'_{mi} = \frac{1}{|S_i|} \times \sum_{j=1}^{|S_i|} d'_{i,j} \tag{14}$$

With $d'_{i,j} = \sqrt{(cx'_i - x'_{ij})^2 + (cy'_i - y'_{ij})^2}$.

After that, we continue to find the minimum distance and maximum distance in all distances from the center of the slice $S_i$ to points as described in equations (3) and (4) for calculating $\Delta'_i$. The watermark bit $\omega_i$ will be then extracted by comparing the mean distance $D'_{mi}$ and $\Delta'_i$. If $D'_{mi}$ is greater or equal than $\Delta'_i$, $\omega_i$ will be 1. If $D'_{mi}$ is less than $\Delta'_i$, $\omega_i$ will be 0. The condition of the watermark extraction is described by Eq. (15).

$$\omega_i = \begin{cases} 1 \text{ if } D'_{mi} \geq \Delta'_i \\ 0 \text{ if } D'_{mi} < \Delta'_i \end{cases} \tag{15}$$

## 4. EXPERIMENTAL RESULTS

**Table 1:** Mean Distance Error

| **Name** | **# of facets** | **# of slices** | **Mean distance** (mm) |
|---|---|---|---|
| Cup | 28630 | 280 | $9.993756\times10^{-5}$ |
| Yoda | 49844 | 630 | $9.987180\times10^{-5}$ |
| Batman | 13566 | 730 | $9.962999\times10^{-5}$ |
| Bunny | 86632 | 830 | $9.987746\times10^{-5}$ |
| Armor | 975034 | 960 | $9.996013\times10^{-5}$ |
| Horse | 778532 | 1240 | $9.995868\times10^{-5}$ |

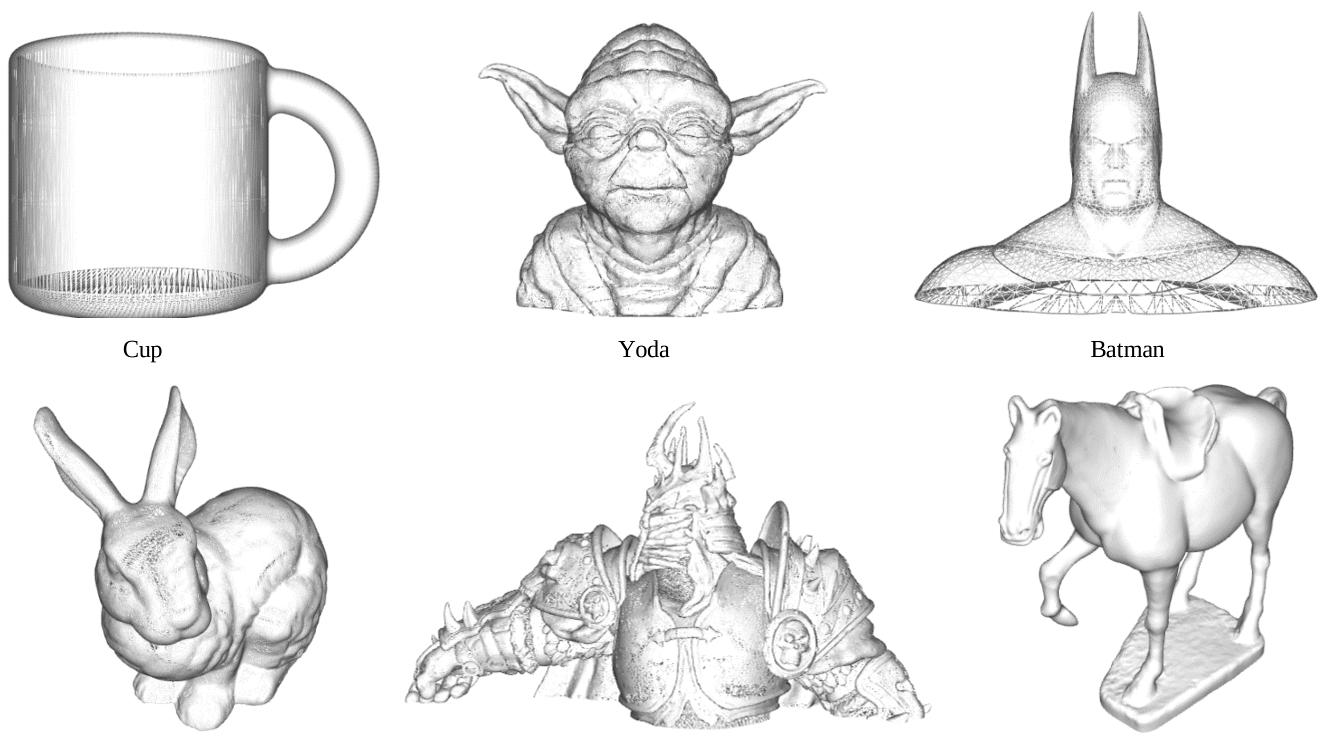
We used test models in Fig. 6 to experiment the proposed method. The format of 3D triangle meshes is STL file [15]. Tab. 1 shows the information of test models in detail. Each test 3D model is sliced into a set of 2D slices. The number of 2D slices of each 3D triangle mesh is dependent on both the thickness of each slice and the Z-axis height of that model. The thickness of slices is flexible and determined by users when they configure parameters in the 3D printing process. We defined the thickness of slice is 1 mm in our experiments. We evaluated the invisibility, robustness and performance of the proposed method for the evaluation the aspects of the proposed method.

### 4.1 Invisible Evaluation

We calculated the mean distance error $d^m(p,p')$ between the watermarked points $p'$ in the watermark slices and the original points $p$ in the original slices of a 3D triangle mesh as shown in Eq. (16) to evaluate the invisibility of the proposed method.

$$d^m(p,p') = \frac{1}{\mathrm{N}} \times \frac{1}{|S_i|} \sum_{i=1}^{\mathrm{N}} \sum_{j=1}^{|S_i|} ||p_{ij} - p'_{ij}|| \quad (16)$$

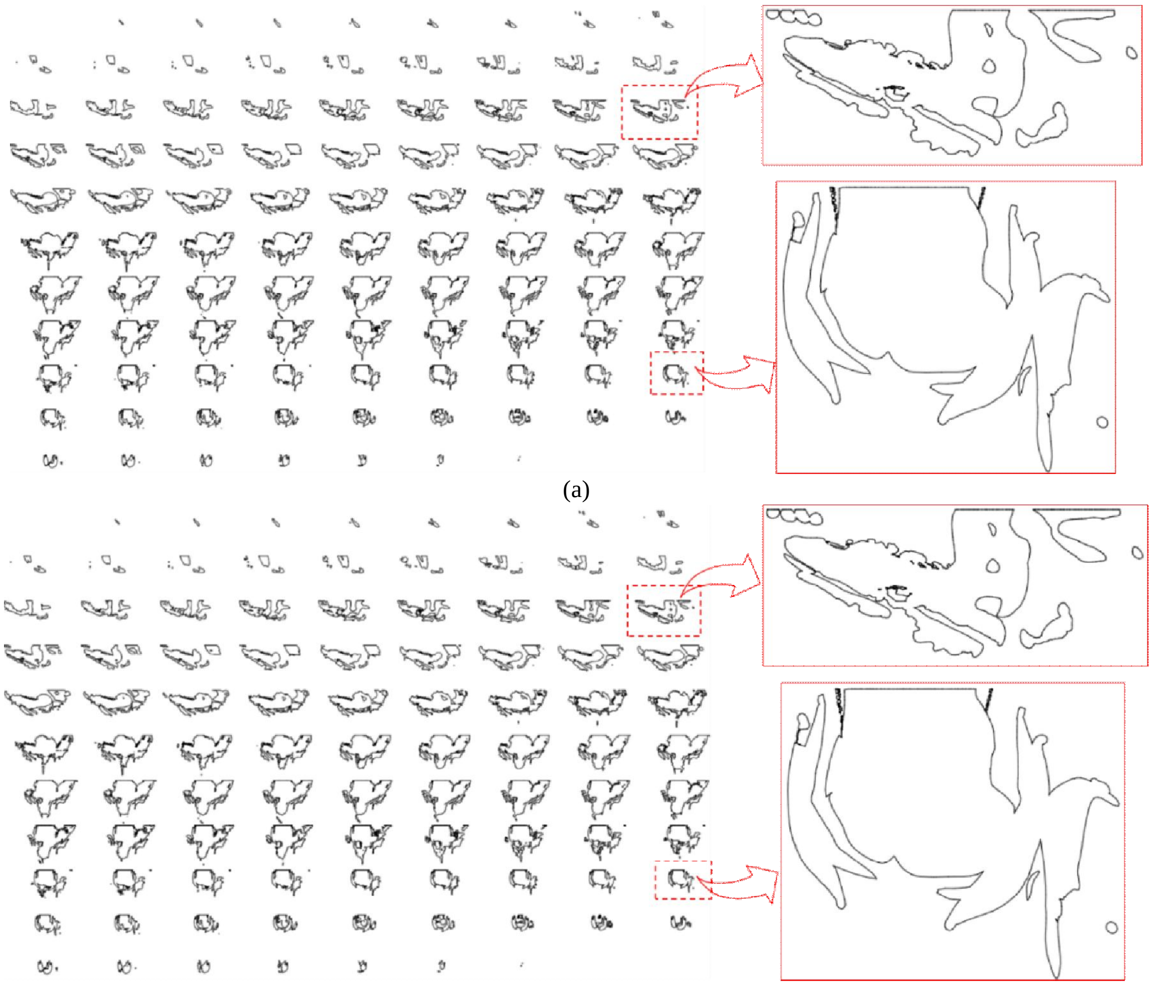
The mean distance error between the original 2D slices and the watermarked 2D slices of test 3D triangle meshes is shown in Tab. 1. From Eq. (16), we concluded that the mean distance error is dependent on the number of points in each 2D slice and the number of 2D slices in each model. The number of points in a slice is dependent on the number of facets in 3D triangle mesh, the thickness of slices and the height of each model. Based on Tab. 1, we concluded that the mean distance error between the original 2D slices and the watermarked 2D slices is very small. It is formed from $0.9963\times10^{-6}$ to $0.9993\times10^{-6}$ with test models in Tab. 1. This means the change rate between the watermarked 2D slices and the original 2D slices is very small. So, we concluded that the invisibility of the proposed method is very high. Fig. 7 shows an original 2D slice from a set of 2D slices and it's the watermarked 2D slice. We illustrate a set of 2D slices together a 2D slice for looking over minutely. From these figures, we verified that the perceptual difference of two sets of slices is very small.



Cup Yoda Batman

Bunny Amor Horse

**Figure 6:** Test models



(a)

(b)

**Figure 7:** (a) Original slices of Armor model, and (b) watermarked slices

### 4.2 Robustness Evaluation and Analysis

**Table 2:** Bit Error Rate

| **Name** | **Bit Error Rate** (%) | | |
|---|---|---|---|
| | 0.01% Noise | 0.05% Noise | 0.1% Noise |
| Cup | 0.000 | 0.0327 | 0.0535 |
| Yoda | 0.000 | 0.0339 | 0.0539 |
| Batman | 0.000 | 0.0382 | 0.0565 |
| Bunny | 0.020 | 0.0425 | 0.0602 |
| Armor | 0.036 | 0.0463 | 0.0625 |
| Horse | 0.041 | 0.0486 | 0.0683 |

To evaluate the robustness of the proposed method we calculate the bit error rate (BER) between the extracted watermark and the original watermark when the watermarked slices are attacked by rotation, scaling, translation and random noises. Firstly we performed geometric attacks as rotation, scaling and translation. Rotation and translation only change the spatial location of 3D triangle mesh. Scaling changes the size of 3D triangle mesh. With rotation attack, it rotates 3D triangle mesh by an angle, thus we only align 3D triangle mesh and slice the 3D triangle mesh along the Z axis for obtaining 2D slices. Therefore, rotation did not affect to the result of watermark extraction. With translation attack, it only changes the position of 3D triangle meshes in 3D space. It did not change the shape and the volume of 3D triangle mesh. Thus, when we slice it by the cutting plane from bottom to top, the shape of slices and the number of slices will be not changed. This means translation does not affected to the result of watermark extraction. With scaling attack, it increases or reduces the size of 3D triangle mesh but it does not change the center of 3D triangle mesh. To re-scale, we find the highest vertex and the lowest vertex on the original 3D triangle mesh. We then compute the distance between these vertices. With the scaled 3D triangle mesh, we also perform similar. Finally, we compare distances to find scale-ratio for the re-scaling process. Here, we could concluded that the proposed algorithm is robust to geometric attacks. Secondly, we performed the noise attacks to 3D printing. We used visual 3D printer and added random noises. We experimented with random noises 0.01 %, 0.05 % and 0.1 % respectively.

We compute BER by comparing the original watermark data with the extracted watermark data:

$$BER = \frac{Extracted\ watermark}{Original\ watermark} \times 100\% \tag{17}$$

Tab. 2 shows the BER between the extracted watermark data and the original watermark data. In experiment with "Cup" model, the number of 2D slices is small, thus the BER is also small with differential noise ratios (from 0.000 to 0.0535). In experiment with "Horse" model, the number of 2D slices is bigger, thus the BER is also high (from 0.041 to 0.0683). Consequently, with noise ratio is small and the number of 2D slices small, the BER is zero (see Cup, Yoda and Batman models with 0.01 % noise). But with the ratio increase of noise or the number of 2D slices, the BER also increases (see Bunny, Armor and Horse models with 0.1 % noise). Based on Tab. 2, we conclude that the BER is increased according to the number of 2D slices and noise. So, it is dependent on both the number of slice and noise ratio. If both the number of slices and noise are small, the BER will be low and otherwise. From Tab. 2, we verify that the BER of the proposed method is very low. So, it can be responsive to the copyright identification of 3D printing. Fig. 8 shows the mean BER of test models according to the ratios of random noise.

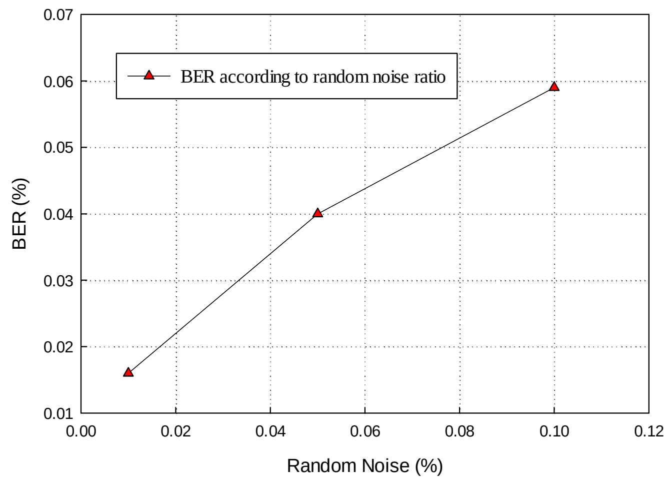



**Figure 8:** BER according to the random noise

### 4.3 Performance Evaluation

**Table 3**: Effect of Geometric Attacks to Methods

| Geometric Attacks | Affected | | | | | |
|---|---|---|---|---|---|---|
| | Our method | Rolland's method | Mona's Method | Liu's Method | Hou's Method | Wang's Method |
| Rotation | No | Yes | Yes | Yes | Yes | Yes |
| Scaling | No | Yes | Yes | Yes | Yes | Yes |
| Translation | No | Yes | Yes | Yes | No | Yes |

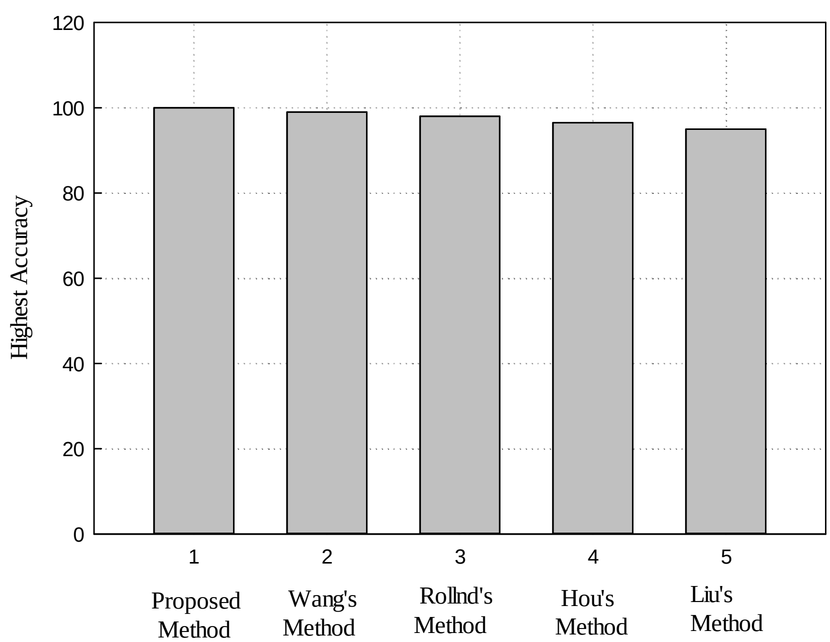



**Figure 9:** The highest accuracy of methods

To compare the performance of the proposed algorithm to the conventional works of 3D model watermarking recently, we compare the robustness of methods with geometric attacks and compare the highest achieved accuracy between methods. Tab. 3 shows geometric attacks affected to methods. Based on Tab. 3, the robustness of our method is better than other methods. In Liu's method [10], he presented that his method could extract the embedded watermark when less than 5% of meshes are simplified. This means the max accuracy of his method is 95%. In Rolland's method [11], he used 13 models with noised 0.1%, 1% and 5% respectively in his experiments. The max accuracy of Rolland's method is 98%. In Hou's method [13], four 3D triangle meshes are used in experiments and the average bit error rate (BER) is approximate 4%. Thus, the average accuracy of Hou's method is 96%. In Wang's method [14], he achieved the max accuracy is 98.5 %. In our method, the maximum accuracy of the extracted watermark is 100%. This means the accuracy of our method is better than the accuracy of previous researches for 3D model watermarking. Fig. 9 the accuracy of our method comparing to the conventional works of 3D model watermarking.

## 5. CONCLUSION

We proposed a watermarking algorithm for 3D printing in this paper. It is based on watermarking for the 2D slices of 3D triangle mesh. The proposed algorithm is invisible and robust to geometric attacks as rotation, translation and scaling. Experimental results verified that the BER of the proposed method is low BER. Next time, we will improve and experiment the proposed algorithm with other attacks to evaluate the proposed algorithm for full aspects. Moreover, we will experiment with some real 3D printers and apply to copyright protection system. We will consider to apply the proposed algorithm to case applications.

## ACKNOWLEDGEMENT

This work is supported by the FPT University, Hanoi, Vietnam; and Ho Chi Minh University of Technology, Ho Chi Minh city, Vietnam.

## REFERENCES

1. 3D Systems Circle Rock Hill. **White paper: How 3D Printing works, The Vision, Innovation and Technologies behind Inkjet 3D Printing**, Jan., 2012.
2. T. A. Chandra, M. Patel, and P. K. Singh. **Study of 3D Printing and its Application**, *International Journal for Research in Advanced Computer Science & Engineering*, Vol. 2, pp. 9-12, March 2016.
3. Learn how 3D Printing is useful everywhere. https://www.sculpteo.com/en/applications/ (accessed on 10 Nov. 2019).
4. S. Ira and S. Parker. **Copyright Issues in 3D Printing**, in *Proc. of the International Technology Law Conference*, 2014, pp. 1–14.
5. H. Y. Amir, S. A. Mohd, and G. Alireza. **A Novel Color Image Watermarking Method based on Digital Wavelet Transform and Hungarian Algorithms**, *International Journal of Advanced Trends in Computer Science and Engineering,* Vol. 8. No. 2, pp. 154-164, March 2019. https://doi.org/10.30534/ijatcse/2019/09822019
6. D. Kalyan, S. Aromita, and K. B. Samir. **A Robust Visual Cryptography Scheme for Imperceptible Distribution of Secret Message without Pixel Expansion**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 6, No. 4, pp. 62–70, July 2017.
7. K. H. Raviya1, D. V. Vyas, and M. K. **Ashish. SVD Based Performance Improvement in Hiding a Message Behind an Image**, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No. 2, March 2019. https://doi.org/10.30534/ijatcse/2019/12822019
8. X. H. Tan. **A 3D Model Asymmetric Watermarking Algorithm Based on Optimization Statistics**, *Journal of Theoretical and Applied Information Technology*, Vol. 51, No. 2, pp. 175-181, May 2013.
9. Q. Hu and Z. Lang. **The Study of 3D Digital Watermarking Algorithm Which is based on a Set of Complete System of Legendre Orthogonal Function**, *The Open Automation and Control Systems Journal*, Vol. 6, pp. 1710–1716, June, 2014. https://doi.org/10.2174/1874444301406011710
10. J. Liu, Y. Wang, W. He and Y. Li. **A New Watermarking Method of 3D Mesh Model**, *Indonesian Journal of Electrical Engineering*, Vol. 12, No. 2, pp. 1610–1617, Feb. 2014.
11. X. Rolland, G. Do and A. Pierre. **Triangle Surface Mesh Watermarking based on a Constrained Optimization Framework**, *IEEE Trans. on Information Forensics and Security*, Vol. 9, No. 9, pp. 1491–1501, Sep. 2014. https://doi.org/10.1109/TIFS.2014.2336376
12. X. Feng, Y. Liu and L. Fang. **Digital Watermark of 3D CAD Product Model**, *International Journal of Security and Its Applications*, Vol. 9, No. 9, pp. 305–320, Sep. 2015.
13. J. U. Hou, D. G. Kim, S. H. Choi and H. K. Lee. **3D Print-Scan Resilient Watermarking Using a Histogram-Based Circular Shift Coding Structure**, in *Proc. of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, 2015, pp. 115–121.
14. Y. Wang, L. Jing, Y. Yang, D. Ma, and R. Liu. **3D model watermarking algorithm robust to geometric attacks**, *IET Image Processing*, Vol. 11, pp. 822-832, Oct. 2017. https://doi.org/10.1049/iet-ipr.2016.0927
15. STL format in 3D printing. https://all3dp.com/what-is-stl-file-format-extension-3d-printing/, (accessed on 10 Nov. 2019).
16. The Virtual Reality Modeling Language. http://www.cacr.caltech.edu/~slombey/asci/vrml/, (accessed on 10 Nov. 2019).
17. 3D Slicer. https://www.slicer.org/ (accessed on 10 Nov. 2019).
18. KIS Slicer. http://www.kisslicer.com/ (accessed on 10 Nov. 2019)