Robust watermarking for 3D objects

Zoran Rušinović[a], Željka Mihajlović[b]
[a] R&D Center, Ericsson - Nikola Tesla, Krapinska 45, 10000 Zagreb
[b]ZEMRIS, Faculty of Computing and Electrical Engineering, Unska 3, 10000 Zagreb
E-mail: zoran.rusinovic@ericsson.com, zeljka.mihajlovic@fer.hr

Abstract: Copyright protection of digital media is an important issue in the distribution of digital content. More and more interest is being taken in methods to protect the copyright of digital data and prevent illegal duplication of it. Digital watermarking technology allows users to embed specific information, identifying the owner, in the host data imperceptibly. After developing the theory and practical details of the Yu-Kwok's watermarking scheme, it is shown that this method will produce pathological case in some circumstances. Modification of the algorithm that will reduce this unwanted impact of the embedded watermark on the local geometry is proposed. The effectiveness of the proposed technique is then presented by the windows application for robust watermarking of 3D triangular mesh models that implements our modified algorithm.

I. INTRODUCTION

In the field of watermarking, until recently, most of the work was concentrated on audio, image and movie watermarking. However, spread of networks and digital multimedia materials, such as Web3D, MPEG4 as well as various 3D geometric CAD data, has prompted much attention to the watermarking techniques for 3D. Contrary to bitmap-based, data types in many fields are vector based, including 3D virtual environment represented by VRML scenes and 2D vector graphic that is part of PostScript and PDF documents.

As it can be seen, more and more 3D data is entering the World Wide Web. For that reason companies and copyright owners who present or sell their products in virtual environments are facing copyright-related problems. If an author produces 3D-based (catalogue) material and wishes them to be copyrighted against unauthorized use or distribution, there must be a method for determining the original author of the model for copyright laws to be enforced in the court of law. Conventional cryptographic system permit only authorized key-holders access to data, but once such data are decrypted there is no way to track its reproduction or transmission. For that reason an old technique known as watermarking has been adapted for use with digital data.

A. *Background*

Digital watermarking embeds imperceptible information, called a digital watermark, in digital content. Such an information is called a watermark code (or a watermark signal). Watermark code can be can be inserted in into a content in many ways, including spatial domain, frequency domain or using some statistical approach. Watermarks can be divided into robust and fragile watermarks, depending on the end applications. A robust watermark is used for ownership assertion, which means that such a watermark must be difficult (hopefully impossible) to remove by any innocent or malicious attack. On the other hand fragile watermarking is used for detecting any changes in host data. Such a watermark can become undetectable once a host data has been changed, or it can alternatively change in such a way that it in some way indicates part of the host data that has been changed. This work is concerned with robust watermarking only.

B. *3D oriented watermarking systems*

The design of watermarking schemes for 3D models is particularly complex for a few reasons. The most important are that only a low volume of data is available for watermark embedding (3D models usually consist of only few thousand vertices) and secondly there is no known transformation of a 3D object from a spatial domain to some other domain that would make an object robust to complex geometrical and topological transformations.

The first watermarking research on 3D models was presented by Ohbuchi et al [1]. They proposed several watermarking algorithms for 3D models: Triangle similarity quadruples (TSQ), Tetrahedral volume ratio (TVR) and mesh density pattern embedding algorithm. These techniques fall into spatial domain and include mesh altering, topology altering and pattern embedding method, respectively.

These algorithms are well suited for embedding public watermarks, but are not sufficiently robust for copyright protection, because they are vulnerable to most of the usual mesh operations (e.g. re-mashing, polygon simplification, noise addition).

In 1999. Praun et al. [2] proposed a robust mesh watermarking scheme that generalized spread spectrum technique to 3D surfaces. They firstly applied a mesh simplification based on edge collapses and then by analyzing approximations errors associated with collapses they identified those vertices that contain 'low frequency' components (i.e. visually the most important components). Those vertices are than in the original mesh displaced in normal/reverse direction (depending of the watermark bit value). The reason to choose 'low frequency' components is that they correspond to the visually most significant features of the model, and are therefore least affected by many types of attack.

In Ref. [3], Yin et al. proposed a watermarking sceheme based on multiresolution edit. They adopted Guskov's multiresolution signal processing method for mashes and used his 3D non-uniform relaxation operator to construct a Burt-Adelson pyramid for mesh. After they displace vertices at the coarse level, the mesh pyramid is inverted. Yu et al. in Ref [4], presented a novel watermarking scheme that embeds the watermark via perturbing the

length of the vectors that extend form a vertex to the center of the model. Since this vectors possess global surface characteristics, the watermark embedded using this method are very robust.

For the issue of visual imperceptibility this algorithm for the first time presents a technique to compute the locally optimal watermarking strength that doesn't affect the visual quality of the model. Our work extends this algorithm.

Despite the fact that due to local strength adaptation this algorithm has very desirable features in respect to imperceptibility, we have discovered cases in which the embedding of the watermark, using the described algorithm, will produce significant alteration of the local geometry. In the next sections circumstances and events under which this problem occurs are described. It is the objective of this paper to propose a computational scheme which may be able to achieve better degree of control and thus reduces the impact of the embedded watermark on local geometry.

II. YU-KWOK'S ALGORITHM

We present the algorithm here in a simplified and condensed form to explain the way the transformation component work and to pinpoint some elements. For a full treatment of the algorithm we refer the reader to [4].

A. Watermark embedding

(1.) Watermark signal is given as a binary sequence W=($w_0, w_1,...,w_{n-1}$); $w_i \in \{-1,+1\}$.

(2.) Scramble the vertices of the original model according to the given permutation's key $V'=Permute(V, K)$. $V_o=(v_{o0}, v_{o1},...,v_{oL-1})$ denotes the vertices of the original model, L is the number of vertices, and $V'_o=(v'_{o0}, v'_{o1},..., v'_{oL-1})$ denotes the vertices of the permuted model.

(3.) Choose the first S*N vertices from V_o' and divide them into N sections $V'_{oi}=(v'_{oi0}, v'_{oi1}, v_{oi2},...,v_{oS-1})$, where $0 \le i < N$. Each section has S vertices.

(4.) Embed a watermark bit into each section $\vec{L}^{w}_{oij} = \vec{L}_{oij} + w_i M_{oij}(\alpha)\cdot\vec{U}_{oij}$, $0 \le i < N$, $0 \le j < S$. $\vec{L}_{oij}$ denotes the original vector, $\boldsymbol{M_{0ij}}$ is the parameter controlling the local watermarking strength and it is the function of global watermarking strength α. $\vec{U}_{oij}$ is the unit vector of the vector $\vec{L}_{oij}$.

(5.) Recover the original order of the watermarked vertices.

C. Watermark extracting

(1.) Perform the registration procedure if necessary.

(2.) Perform the re-sampling procedure if necessary.

(3.) Scramble and divide the vertices to get the N sections $V'_{oi}=(v'_{oi0}, v'_{oi1}, v_{oi2},...,v_{oiS-1})$ of the original model, and $V_{di}'=(v'_{di1}, v_{di2,....,}v_{diS-1})$ watermarked model.

(4.) Use the center of the original model as the center of the detected model. Compute the length's difference between the vectors of original model that link the vertices in each section to the center, and the vectors of the detected model that link the vertices in each section to the center: $D_{oij}=L^{d}_{oij} - L^{o}_{oij.}$

(5.) Weight the length difference, and calculate the sum of the length differences of each section: $D_{oi} = \sum_{j=0}^{S-1} W_{oij} D_{oij}$. W_{oij} is the weighting coefficient (weighting scheme is not important for the scope of this work, and it's not discussed here any further).

(6.) Extract the watermark sequence: $w^{d}_i=sign(D_{oi})$, $0 \le i < N$

(7.) Compute the correlation between the extracted watermark sequence and the embedded watermark sequence:

$$Cor(W^{d},W) = \frac{\sum_{i=0}^{N-1}(w_i^d - \overline{W}^d)\cdot(w_i - \overline{W})}{\sqrt{\sum_{i=0}^{N-1}(w_i^d - \overline{W}^d)^2}\cdot\sqrt{\sum_{i=0}^{N-1}(w_i - \overline{W})^2}} \quad (1)$$

D. Locally adaptive watermarking strength

In Figure 1, P denotes the vertex to be displaced from P to P' along P''.

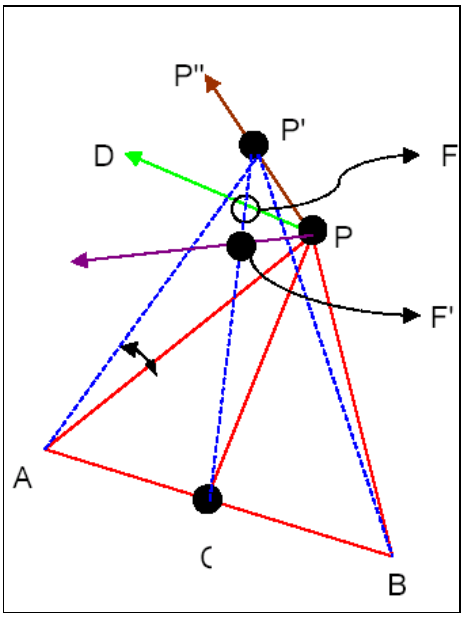

Figure 1. Computing the locally adaptive watermarking strength. The original triangle is shown with the full line, while the dotted triangle is made by displacement of the vertex P.

ΔPAB is one of the P' neighbor triangles. C is a point on line AB and $PC \perp AB$. Vector $\overrightarrow{PD}$ is the face normal of triangle ΔPAB. $\overrightarrow{PD}$ intersects triangle ΔP'AB in point F. F' is a point on line FC and PF' $\perp$ FC. Since PC $\perp$ AB and PD $\perp$ AB, PF' $\perp$ AB. Also since PF' $\perp$ AB and PF' $\perp$ FC, F' is the projection of P on ΔP'AB, which means PF' $\perp$ P'F'. In order to

make the embedded watermark as much as possible imperceptible, the displacement of vertex should small enough. This means that we want an angle $\angle$PCF' to be quite small. For this reason we can use $\angle$PCF' as the global strength coefficient of the embedded watermark. When given maximal $\angle$PCF' and the direction of displacement of, we can computer the maximal displacement of vertex P through calculating $\|PP'\|$ as follows (α denotes an angle $\angle$PCF') :

$$\overrightarrow{PF} = \left\|\overrightarrow{PC}\right\| \cdot tg(\alpha) \cdot \frac{\overrightarrow{PD}}{\left\|\overrightarrow{PD}\right\|} \tag{2}$$

$$\overrightarrow{FC} = \overrightarrow{PC} - \overrightarrow{PF} \tag{3}$$

$$\overrightarrow{FF'} = \frac{\overrightarrow{FP} \cdot \overrightarrow{FC}}{\left\|\overrightarrow{FC}\right\|} \cdot \frac{\overrightarrow{FC}}{\left\|\overrightarrow{FC}\right\|} \tag{4}$$

$$\overrightarrow{PF} = \overrightarrow{PF'} + \overrightarrow{FF'} \tag{5}$$

$$\left\|\overrightarrow{PP'}\right\| = \left\|\overrightarrow{PF'}\right\| \cdot \frac{\left\|\overrightarrow{PP''}\right\| \cdot \left\|\overrightarrow{PF'}\right\|}{\overrightarrow{PP''} \cdot \overrightarrow{PF'}} \tag{6}$$

From (2) and (3) we can see that the distance $\|PP'\|$ is a function of $\angle$PCF', or we can write $m=f(\angle \text{PCF'})$ where m=$\|PP'\|$.
As the local watermarking strength the minimum *m* from all P's neighbor triangles is chosen.

III. PROPOSED IMPROVEMENTS TO THE ALGORITHM

A. Identification of pathological cases

As already said, local adaptation of the watermark strength (i.e. of the displacement of the model's vertices) is achieved by limiting the angle that the old normal vector makes with the new normal vector, for all the triangles incident to the particular vertex. On the Figure 1 this angle is marked as $\angle$PCF' while in the computation of the maximal allowed displacement it is denoted as an angle α. After computing the allowed displacements for the particular vertex (in respect to all the incident triangles of the vertex in the input mesh), in such a way that no α angle exceeds the value of the global watermarking strength coefficient, only the minimum value for a displacement is chosen. The key fact to notice here is that the value as well as the direction of the maximal allowed displacement is calculated with respect to the vector $\overrightarrow{PP'}$ (which is collinear with the vector extended between the center of the 3D mash and the particular vertex). Watermarking bit x only determinates whether this displacement will be in the direction of this vector (for the bit value of 1) or in the opposite direction (for the bit value of 0).

From the above description it is safe to assume that for the vertices that are incident to more than one triangle, the embedded watermark will not be visible to the naked eye, since the distortion of those vertices will be limited by the above mentioned minimal displacements. This way even if there exists such a triangle, incident to the vertex, that allows visually detectable displacement of this vertex, it can be assumed that it's impact will be annihilated by some smaller displacement calculated for some other triangle incident to it.

However, using the described algorithm, pathologic case will occur in all those cases where no other incident triangle exists as well as in those cases where the other triangles may induce displacements no smaller than the first one. In the first case, when the vertex is incident to only one triangle, this will happen if the vector $\overrightarrow{PP''}$ is coplanar, or almost coplanar, with the plane in which this triangle lies. This way the vertex of interest for the watermark embedding will be significantly moved along the vector $\overrightarrow{PP''}$ before the angle α reaches the given value of global strength coefficient, thus making the displacement visually very detectable. This situation is shown in Figure 2.

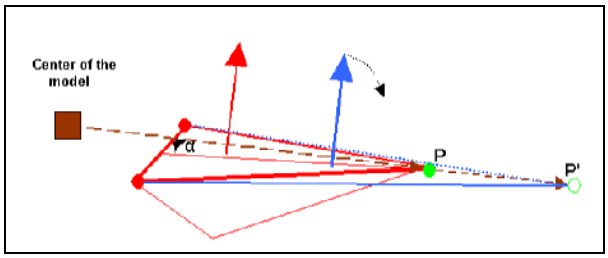

Figure 2. Pathologic displacement of the vertex P. The vertex P is incident only to those triangles. that are almost coplanar with the vector extended form the vertex to the center of the model. The vertex P is consequentially moved to point P', forming the new, significantly different, triangle.

The analogue holds true for the second case as well. In that case the just described problem is cloned to all the triangles incident to the vertex of interest, making it impossible to find a triangle that will reduce the displacement of the vertex to the visually imperceptible one. This situation is also shown in Figure 2.

B. Displacements restriction technique

After detecting those pathologic cases we next propose the solution that will nullify the local geometry changes in the cases mentioned, without virtually affecting the robustness of the embedded watermark or the watermarking scheme in whole.
The proposed solution is based on a calculation of relative displacements and reduction of those displacements that induce the biggest visual impact on 3D mesh. First the relative displacement is calculated as an absolute displacement divided by the length of the vector extended between the center of the model and the observed vertex.

$$\left|PP'\right|_{rel} = \frac{\left|\overrightarrow{PP'}\right|}{\left|\overrightarrow{CP}\right|} \tag{7}$$

Where C denotes the center of the model (please refer to the Figure 2).

By examining these values we have concluded that most of them have a tendency to cluster around some particular value, which is referred to as an estimator. Probably a best known one is the mean of the values $x_1 \ldots x_n$:

$$\bar{x} = \frac{1}{N} \cdot \sum_{j=1}^{N} x_j \qquad (8)$$

However, the mean is not the only available estimator of this value, nor is in this case the best one. Because of the infrequent occurring of the pathological displacements, distribution of the relative displacements has a strong central tendency and most of them are under a single peek. Computing the kurtosis of those distributions for many 3D models has validate this interpretation since all of the distributions were leptokurtic. As a quick reminder the kurtosis is a nondimensional quantity that measures the relative peakedness or flatness of a distribution relative to normal distribution. A distribution is said to be leptokurtic if it has positive kurtosis, platykurtic if it has negative kurtosis, while an in-between distribution is termed mesokurtic. The conventional definition of the kurtosis is:

$$Kurt(x_1 \ldots x_n) = \left\{ \frac{1}{N} \cdot \sum_{j=1}^{N} \left[\frac{x_j - \bar{x}}{\sigma} \right]^4 - 3 \right\} \qquad (9)$$

For this reason in this case the mean is not very useful, since it will give only a compromise value between the two peaks. Instead as the estimator of the central value the median is used. The median of a distribution is estimated from a sample of values $x_1, \ldots, x_n$ by finding that value x_i which has equal numbers of values above and below it (in the case that N is even, it is conventional to estimate the median as the mean of the unique two central values). If the values x_j, j=1...N are sorted into order (either ascending or descending), then the formula for the median is:

$$x_{med} = \begin{cases} x_{(N+1)/2}, N \to odd \\ \frac{1}{2} \cdot \left(x_{N/2} + x_{(N/2+1)} \right), N \to even \end{cases} \qquad (10)$$

One may argue that calculating the median will significantly slow down the original watermarking scheme, since it may seems necessary to sort the relative distances and than apply (10), and that's a process of order N log N. However, that is not true, because the element xn+1/2 can be located in of order N operations (see [5]).

After characterizing a distribution's central value, we must next define variability around that value. As before, more than one measure can be used for that purpose and the most common one is variance. We use a more robust estimator of variability known as the mean absolute deviation, defined by:

$$ADev(x_1 \ldots x_n) = \frac{1}{N} \cdot \sum_{j=1}^{N} \left| x_j - x_{med} \right| \qquad (11)$$

After completing these two steps the Z-score for each displacement can be computed. The Z-score for a displacement, indicates how far that displacement deviates from its distribution's mean, expressed in units of the mean absolute deviation. The mathematics of the Z-score transformation are such that if every item in a distribution is converted to its Z-score, the transformed scores will necessarily have a mean of zero and a standard deviation of one [5].

$$Z = (data\ point - mean)/standard\ deviation \qquad (12)$$

Our algorithms then proceeds with work by selecting n displacements with top Z-scores, where the number of selected displacements can be given as a percentage of the whole set of displacements, or implicitly by threshold indicating the maximal Z-score of any displacement. After those displacements have been identified, they are round down to the relative displacement with the mean absolute deviation of zero. This way the watermarking capacity is saved, since the vertex of interest will nevertheless be displaced in the direction defined by the appropriate watermark's bit. At the same time the above described pathologic displacement of vertices is prevented thus saving the visual quality and usability of the model.

IV. EXPERIMENTAL RESULTS

In order to test this technique for robustness and imperceptibility we have created a Windows application that implements this modified watermarking algorithm. The application is written in C++ and uses win32 and DirectX 9.0 API.

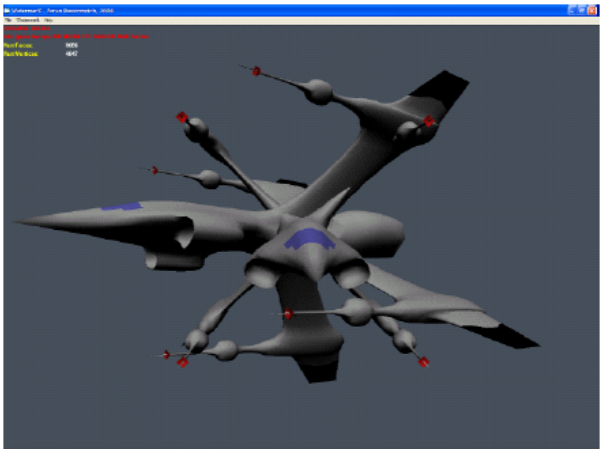

Figure 3. The interface of the watermarking application

Using this application we conducted experiments using two different models. Both models are standardly shipped with DirectX 9.0 SDK. The first model was chosen for it's ability to practically demonstrate such a topology that benefits from our modification while the second one was chosen in relation to the number of vertices and triangles in the model used in the Yu-Kwok's paper. With the second model (that consists of 4647 vertices and 5870 triangle faces) we have in particular tried to come up with a

triangle mesh that has similar number of vertices as the model used by Yu and Kwok for the sake of robustness comparison between the two algorithms.
In both cases we embedded a watermark of 50 bits with α=0.05.

As it can be seen on Figure 4b, embedding the watermark using the unmodified algorithm, results in a pathologic displacement of the vertex (encircled in red).

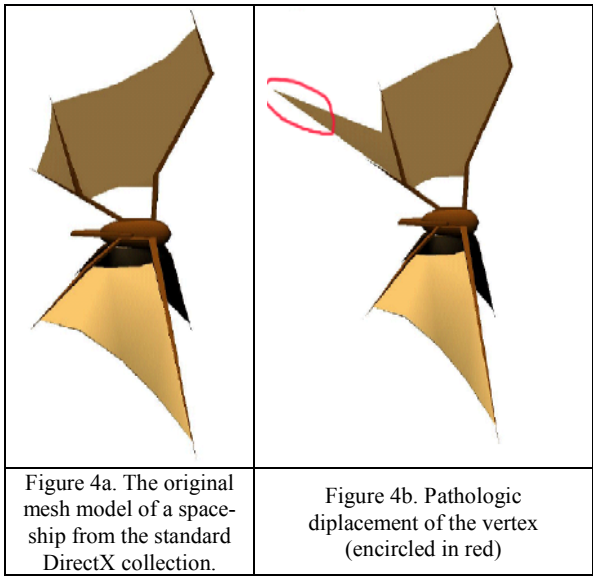

Figure 4a. The original mesh model of a space-ship from the standard DirectX collection.

Figure 4b. Pathologic diplacement of the vertex (encircled in red)

On the Figure 4c the same watermark is embedded using the modified algorithm. Visually comparing these two figures we can see that that in our case the embedded watermark is imperceptible (the original model is shown in the Figure 4a).

As for the robustness testing we have used the usual set of attacks.

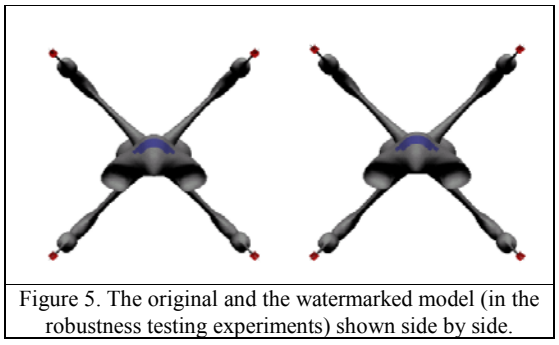

Figure 5. The original and the watermarked model (in the robustness testing experiments) shown side by side.

For simplification attack we firstly conducted the progressive mesh conversion. Afterwards, we reduced 50%, 60%, 70%, 80% and 90% of the vertices of the ship model.

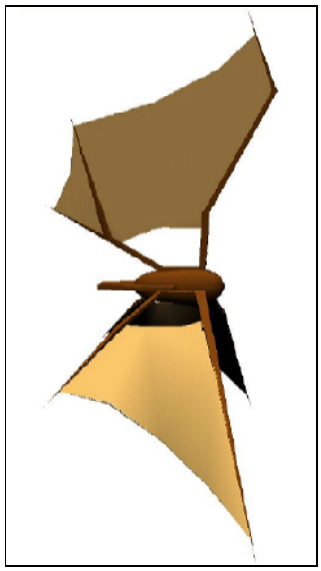

Figure 4c. The same model watermarked using the modified algorithm

Figures 6a. and 6b. show models with 80% and 90% of vertices removed respectively.

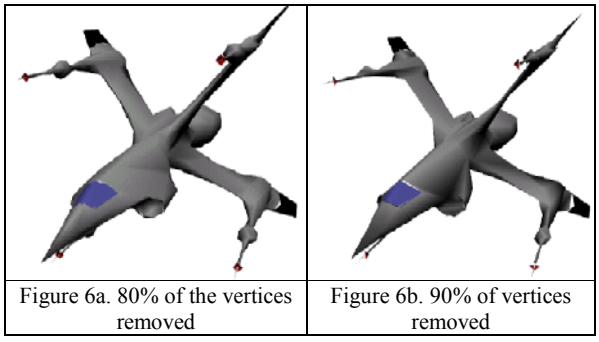

Figure 6a. 80% of the vertices removed

Figure 6b. 90% of vertices removed

The experimental results shown in Table 1. were same in case were we used our modified algorithm, as well as in case were the original algorithm was used, indicating that robustness of the watermarking scheme was not deteriorated by our modifications.

Table 1.
Results of simplification attacks

Vertices left (%)	50%	40%	30%	20%	10%
Verrtices left (number)	2322	1859	1395	930	465
Correlation	1.0	1.0	0.85	0.96	0.7

We have also tested the watermarking robustness using other usual attacks such as noise vector adding to each vertex and cropping of the model in four different cases. The noise was added by displacing the each vertex by 0.1%, 0.2%, 0.3%, 0.4% and 0.5% of the length of the

longest vector extended from a vertex to the center of the model.

Figure 7a. adding a noise vector of 0.4%

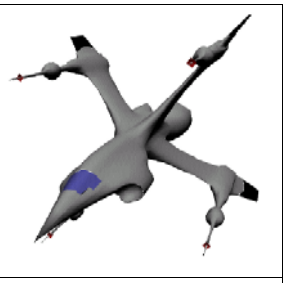

Figure 7b. adding a noise vector of 0.5%

The cropping attacks included removing 25%, 30%, 45% and 60% of the triangles in the watermarked space-ship model.

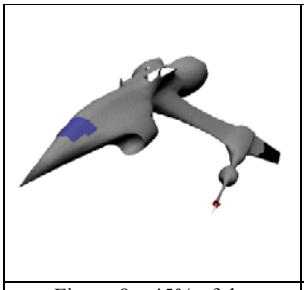

Figure 8a. 45% of the triangles removed

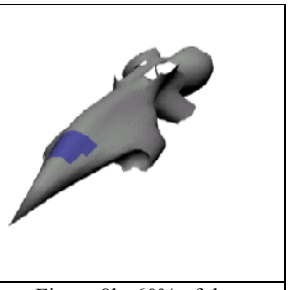

Figure 8b. 60% of the triangles removed

These results again demonstrated that the modified algorithm's robustness suffered no deterioration since correlation values were same as in the results obtained by an unmodified algorithm. These results are shown in Table 3 and Table 4, respectively.

Table 3.
Results of noise attack

Noise	0.1%	0.2%	0.3%	0.4%	0.5%
Correlation	1.0	1.0	1.0	1.0	1.0

Table 4.
Results of cropping attacks

Triangles left (%)	85	70	55	40
Triangles left (number)	7400	6114	4857	3562
Correlation	1.0	1.0	1.0	1.0

V. CONCLUSION

In this paper we presented the modified method for robust watermarking of 3D models. We have analyzed Yu-Kwok's algorithm and detected cases where the embedded watermark can alter local geometry, making the watermarked object unusable. This work extends the original watermarking scheme in a way to make the embedded watermark visually imperceptible in all cases. To make this possible we have developed the technique called displacement restriction that does not require any extensive computation time and at the same time makes the watermarking scheme no less robust. Furthermore, we have implemented this technique in the windows application for robust watermarking of 3D models. Experiments showed that with this approach we were able to accomplish our objectives.

REFERENCES

[1] Ryutarou Ohbuchi, Hiroshi Masuda, and Masaki Aono, "Embedding data in 3D Models", Proceedings of the *European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services* (*IDMS*) *'97,* pp. 261-272, 1997

[2] E. Praun, H. Hoppe, A. Finkelstein. "Robust mesh watermarking". *ACM SIGGRAPH 1999*, pp. 69-76.

[3] Kangkang Yin, Zhigeng Pan, Jiaoying Shi, David Zhang, "Robust mesh watermarking based on multiresolution processing", *Computers & Graphics 25*, (2001), Elsevier Science Ltd., pp. 409–420

[4] Zhiqiang Yu, Horace H. S. Ip and L. F. Kwok, "A robust watermarking scheme for 3D triangular mesh models". *Pattern Recognition*, vol. 36, pp. 2603-2614, 2003

[5] William H. Press, Brian P. Flannery, Saul A. Teukolsk William T. Vetterling, "Numerical recepies in C", Cambridge University Press; 2 edition, 1992.

[6] Burt P., Adelson EH., "Laplacian pyramid as a compact image code", *IEEE Transactions on Communications 1983*, (04.1983), pp. 532–540.

[7] Benedens, O. "Geometry-based watermarking of 3d models"**.** *IEEE Computer Graphics and Applications*, 1999, pp. 46–55.

[8] Mauro Barni, Franco Bartolini, "Watermarking Systems Engineering"*, Marcel Dekker Inc*, 2004, ISBN: 0-8247-4806-9