ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **16**, Number 4, 2013, 287–303

A Secure Image Based Watermarking for 3D Polygon Mesh

Hitendra GARG, Suneeta AGARWAL

Department of Computer Science and Engineering
Motilal Nehru National Institute of Technology, Allahabad, India
E-mail: `hitendra.garg@gmail.com, suneeta@mnnit.ac.in`

Abstract. The proposed watermarking algorithm is based on geometrical properties of 3D mesh. The vertices of 3D mesh surface are categorized into flat, peak and deeper region. We first compare the perceivable distortion due to watermark insertion in the vertices of deep surfaces and the vertices belonging to either flat or peak surfaces. Our comparative analysis shows that insertion of watermark in deeper surface have less distortion in comparison to watermark insertion in flat and peak surfaces. In view of the fact, these vertices are categorized into three mutually exclusive groups. The first group (S_1) is composed by the vertices of deeper area. For obtaining the vertices of second group (S_2), the vertex normal is represented in IEEE-754 representation and n^{th} bit of mantissa is considered as '1'. The third set (S_3) contains the remaining vertices. The vertices from S_1 and S_2 are selected for watermark embedding according to the secure code, which is obtained from hash function applied over the secure image (I). The vertices of S_3 remain intact. The watermark embedding is done by re-positioning the selected vertices from their original positions according to the category they belong to. The robustness is assessed against different distortion and distortion-less attacks. The subjective (MoS) and objective (Hausdorff distance, RMS) assessment of watermarked object are also done to measure the extend of similarity between original object and watermarked object.

Key words. Original 3D mesh (O), Watermark 3D mesh (O^w), Mean Curvature, Hausdorff distance, RMS distance, Objective measurement, Subjective measurement, MD5.

1. Introduction

The extensive growth of internet and wide accessability of data items over internet are increasing the misuse of data. These data items may be text, audio, video or 3-Dimensional (3D) objects. Due to the misuse of data, security has become an important issue for their protection. 3D objects can be viewed from any angle like top-view, side-view, lower-view etc. 3D objects has large number of applications in architecture design, machine design, cultural heritage and entertainment. Thus, these 3D objects needed to be protected by using cryptography, digital signature[1], watermarking[2] etc. 3D objects are categorized in two groups volume based objects(voxels, constructive solid geometry) and surface based objects(implicit surface, parametric surfaces, polygon mesh). In our work, we are concerned with 3D polygon mesh only.

Watermarking is an art of hiding secondary data into primary data such that the perceivable quality of primary data remains at its acceptable level. The imperceptibility, robustness and payload are the foundation requirements for a good watermarking algorithms. Imperceptibility is the property that watermarked 3D object should look similar to the original one. The robustness refers to the sustainability of watermark against different types of attacks. The payload refers to the amount of information embedded into the original object. The designing of a watermark algorithm requires a trade-off among imperceptibility, robustness and payload.

The watermark algorithm proposed in this paper takes into consideration the geometrical properties of 3D mesh object for selecting vertices to embed the watermark. The selection of vertices is done considering the perceivable distortion, which is defined as the distortion observed by humans visual system. Since it cannot be mathematically modeled, hence cannot be automated. The subjective assessment has been in use to assess the perceivable distortion[3] due to various 3D processing operations (compression, smoothing, simplification and subdivision). Here we have used it to measure the distortion due to watermarking.

Watermark embedding in deeper surface has less perceivable distortion in comparison to watermark embedding in flat or peak surfaces. In the proposed algorithm, we exploit this observation for selection of vertices for watermark embedding. The vertices of deeper areas are less in number, therefore to enhance the robustness and payload, other vertices are also selected for watermark embedding. The vertices of deeper surface are watermarked with comparatively higher weight than the other selected vertices to maintain overall good perceivable quality.

The existing 3D processing like simplification, compression, watermarking etc are generally evaluated by Hausdorff distance, Root Mean Square Error (RMS), Geometric Laplacian measures (GL_1 and GL_2)[6], Mesh Structural Distortion Measure (MSDM)[7], Roughness-based Measures [3] etc but these measures are not correlated with the human vision. Figure 1 shows various processed Horse object with same RMS distance but varying visual quality.

The visual quality between two objects of 3D polygon mesh not only depends on objective measurement but also on subjective measures as observed by Human perception. In view of the above fact, we consider perceivable distortion for selection of vertices for watermark embedding.

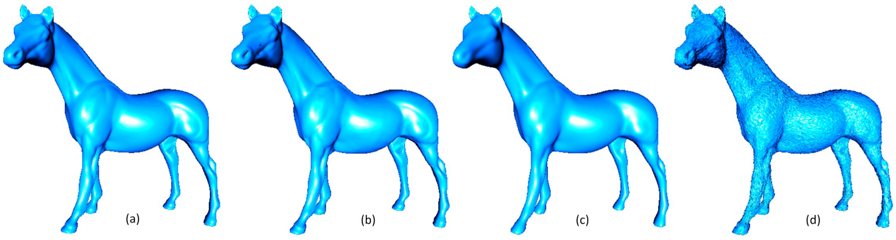


Fig. 1. Original and distorted versions of the Horse model, all 3-D mesh model are having RMS error (MRMS = 1.05×10^{-3}). (a) Original model, (b) watermarked 3-D object using [4], (c) Laplacian smoothing, (d) Gaussian noise addition (**Adapted from [5]**).

The paper is structured as: Section 2 – summary of literature survey; Section 3 – explains watermark algorithm; Section 4 and Section 5 describe experimental results and conclusion of the proposed work respectively.

2. Background

Watermarking of 3D object is different from the watermarking scheme of image, video or audio. Watermarking algorithms of 3D objects can be broadly categorized based on 1. Data file Organization, where watermark is inserted by modifying the organization of data associated with file of 3D object 2. Topological data, where connectivity of polygon mesh is used for watermark insertion maintaining the position of the vertices. The geometry of the mesh i.e. positions of the vertices are not modified during the embedding of the watermark. The watermark is embedded by modifying on the edges the mesh. 3. Geometrical data, where the watermark is inserted by slight modifications performed on the geometric data of the 3D object[8].
The above categorized watermarking algorithms of 3D objects can be implemented either in spatial domain or transform domain. In the spatial domain watermarking, watermark information is embedded in the geometrical data(vertices) and connectivity data(edges) of the mesh directly. In spatial domain, changes due to the watermark insertion can directly visually affect the visual quality. In transform domain, the watermark data is inserted without directly modifying the coordinates of the vertices, but its characteristics are modified. The transform domain watermarking of 3D objects are the extension of regularly signal processing of 3D mesh by spectral decomposition, spherical wavelet transform or wavelet transform etc. The transform domain are comparatively successful for conventional data type (1D or 2D) but they do not attain the similar results in 3D. The important reason is due to the irregular nature of sampled 3D data and their connectivity.

In the first work of 3D watermarking, Ohbuchi [9] suggested four watermarking algorithms named Triangle Similarity Quadruple (TSQ), Tetrahedral Volume Ratio (TVR), Triangle Strip Peeling Sequence (TSPS) and Macro Density Pattern (MDP). These schemes provide a basic foundation of almost all algorithms based on topological data so far.

The watermarking algorithms based on topological data are comparatively less robust than geometrical data based watermarking algorithm [2] due to different attacks like mesh simplification, vertex recording etc. The watermarking algorithm based on geometrical are more robust but degrade the visual quality of watermark object. The degradation in visual quality of 3D object w.r.t. original object is due to the repositioning of vertices from their original positions.

In existing watermarking algorithms [2][10][11][12] the selection of vertices are done regardless of perceptual distortion. We here propose an algorithm which processes the 3D mesh to identify the regions which reflect less perceptual distortion even by embedding watermark of higher strength. The subjective and objective measures for accessing the visual quality of 3D object are to quantify the differences between distorted 3-D object w.r.t. the cover object[13]. The non-blind nature of the proposed algorithm makes it more robust than the other blind one[14][2], where both cover and watermark objects are required at the time of watermark authentication.

3. Watermarking algorithm

In the proposed watermarking algorithm, watermark is embedded by re-positioning some selected vertices. It is done by displacing ρ component of (ρ,θ,ϕ) spherical coordinate of vertices. The re-positioning of selected vertices are done such that change in geometrical shape of 3D object causes less distortion. The watermarking of 3D polygon mesh object is done in the following steps:

3.1. Preprocessing

3D polygon mesh does not have implicit connectivity and order of vertices. So, the preprocessing is required such as registration and re-sampling[15]. Initially, the center of gravity(Cg) of 3D polygon mesh is determined and shifted to the origin of the rectangular co- ordinate system. 3D mesh object is then normalized by scaling the vertices coordinates to lie between -1 and $+1$ units. This ensures the algorithm to be robust against distortion-less attack[15].

3.2. Selection of vertices for watermark embedding

A 3D mesh is represented by G $\equiv$ (V,F) where V is a finite set of vertices and F is a finite set of faces[16]. Mathematically, it has been shown as V= $\{V_i \in R^3 \mid 1 \leq$ i $\leq$ N $\}$ where N is no. of vertices and F= $\{F_j \mid 1 \leq$ j $\leq$ M$\}$ where M is total no. of faces in the mesh[8]. There is one another simplices called edge, formed by connecting two vertices and is defined as edge E=$\{\{$i,j$\}|$i,j $\in I$,$\{$i,j$\}\}$ is an edge$\}$, where I represents the index of the mesh vertices [8].

In view of the distortion and robustness, the vertices are categorized into three mutually exclusive groups S_1, S_2 and S_3. The first group (S_1) contains the vertices of deeper surface obtained by using mean curvature method suggested by M. Mayer et al. [17]. The second group (S_2) is obtained by considering n^{th} bit of mantissa in IEEE-754 floating point representation of vertex normal being '1'. The third group

(S_3) contains the remaining vertices. We create ten bins of S_1 by equally partitioning into size m and move vertices from S_1 to S_3 having mean curvature value less than zero and greater than $m/2$. This process ensures that S_1 contains vertices of only high deeper surface. The surface curvature gives a unique view point description of the local shape [17]. A secure image (I) is used as watermark primitive. The cryptographic hash function (H_f(I)) is applied on this secure image to obtain the hash code of 256 bits ($p_1, p_2 ...... p_{256}$). Thus, hash code is partitioned into two parts ($W1$ and $W2$) which are used to select vertices in S_2 and S_1 respectively for watermark embedding. The selection of vertices for watermark embedding in S_1 and S_2 depends on the secure code while vertices of S_3 remains intact.

The vertices with mean curvature value greater than zero, zero and less than zero represents vertices of peak, flat and deeper surfaces respectively.

Algorithm 1 *Watermark insertion by identifying the vertices using geometrical properties*

```
Require: 3D mesh object (.obj file format) and one secure image.
1: Preprocess the 3D mesh object by re-scaling the vertices between [−1,1] and calculate the center of gravity of 3D mesh object. Also transform the vertices from cartesian coordinate to spherical coordinate.
2: Calculate the mean curvature of 3-D polygon mesh as suggested by [17] to obtain mean curvature values of 3-D mesh object.
3: Divide the vertices in three mutually exclusive groups $S_1$, $S_2$ and $S_3$ based on mean curvature value and IEEE754 floating point representation of number as follows.
4: Group $S_1$ contains all the vertices having mean curvature value less than zero.
5: Group $S_2$ is created by using IEEE-754 floating point representation of vertex normals of vertices having mean curvature value greater than and equal to zero. Group $S_2$ contains the vertices having '1' at $5^{th}$ position in its mantissa part of vertex normal.
6: All the remaining vertices belong to group third $S_3$.
7: Create ten bins each of size $m$ by equal partitioning the vertices of $S_1$.
8: Move vertices from $S_1$ to $S_3$ having value greater than m/2.
9: Arrange vertices of $S_1$ according to curvature value in ascending order.
10: Arrange vertices of $S_2$ w.r.t. positions in the file.
11: Cryptographic hash function(MD5) is applied on one secure image to get the hash code of 256 bits.
12: Divide 256 bit hash code into two groups W1,W2. Constitute W1 by taking initial 10 to 50 bits from the hash code. Remaining bits will form W2.
13: Bit sequence of W1 and W2 are mapped to $S_2$ and $S_1$ repeatedly. The vertices with corresponding hash code bit being '1' are selected for watermark embedding.
14: Watermarking is done by modifying the vertex normal distance of selected vertices. Vertices of $S_1$ are re-positioned by higher value 'W' than the vertices of $S_2$ from their original position.
15: Vertices are re-arranged according to their index values.
16: Modified spherical coordinates are converted back into cartesian coordinates to obtain watermark object.
```

3.3. Watermark insertion

The proposed non-blind algorithm in spatial domain directly inserts the watermark information into the vertices. The proposed algorithm is geometry-driven i.e. connectivity/topology of the vertices is not modified. The watermark is embedded by repositioning the selected vertices according to the groups they belong. The repositioning of selected vertices are performed by increasing the vertex normal distance without modifying the connectivity of the vertices. The amount of increment or decrement is done so as to minimize the distortion due to the repositioning of vertices. The vertices of S_1 are assigned comparatively higher weights than the vertices of S_2. The vertices of the S_3 are kept intact to maintain the perceptual visual quality and robustness. The watermark is embedded by increasing only ρ value of selected vertices while θ and ϕ remain constant using equation 1:

$$\rho' = \rho + W, \tag{1}$$

where weight 'W' is the amount to reposition of the selected vertices from their original positions. The value of 'W' for S_1 is more than its value for S_2.

Firstly, cartesian coordinates of the vertex v$_i$ =(x_i,y_i,z_i) of the original mesh is converted into the spherical coordinates (ρ_i,θ_i,ϕ_i) where:

$$\rho_i = \sqrt{(x_i - x_{cg})^2 + (y_i - y_{cg})^2 + (z_i - z_{cg})^2}, \tag{2}$$

$$\theta = \tan^{-1} \frac{y_i - y_{cg}}{x_i - x_{cg}}, \tag{3}$$

$$\phi = \cos^{-1} \frac{z - z_{cg}}{\rho_i}. \tag{4}$$

Now, vertices (v_i' =(x_i',y_i',z_i')) of watermark object are obtained as :

$$\begin{aligned} x_i' &= \rho_i' * cos(\theta) * sin(\phi) + x_{cg}, \\ y_i' &= \rho_i' * sin(\theta) * sin(\phi) + y_{cg}, \\ z_i' &= \rho_i' * cos(\phi) + z_{cg}. \end{aligned} \tag{5}$$

Here (x_{cg} , y_{cg} ,z_{cg}) is the center of gravity and ρ is the normal distance of a vertex from the center of gravity. The watermark is inserted by modifying the value the vertex normal. So, modified vertices are then re-shifted to their original positions and re-scaled to obtain a watermarked object. The changes in 3D object w.r.t. the original object is the watermark.

3.4. Watermark Authentication

The proposed watermarking algorithm is non-blind in nature as it requires cover object as well as watermarked object at the time of authentication. The watermark information can simply be obtained by subtracting the vertex normals of cover object from the respective vertex normals of the watermarked object. The subtraction results in sequence of bit pattern after replacement of non-zero value by '1'. This obtained bit pattern is compared with secure hash code. The same bit pattern has been embedded redundantly in 3D mesh depending on the number of segments of the 3D mesh. The bit pattern obtained can be matched from any of the segments. The change in sequence of bits in any segment reports tampering. Actually, non-blind watermarking is used for authentication purpose. The detail description of authentication process is described in Algorithm 2.

Algorithm 2 *Watermark authentication process*

```
Require: Cover 3D object $O$, watermark 3D object($O^w$), original secure image $I$.
1: Calculate secure hash code from secure image after applying cryptographic hash function (MD5) on it.
2: Divide the hash code of 256 bits into two parts W1,W2 same as insertion process.
3: Calculate the mean curvature at each vertex of both $O$ and $O^w$ 3-D polygon meshes as suggested by [17].
4: Divide all the vertices of $O$ and $O^w$ in three groups mutually exclusive groups $S_1$, $S_2$ and $S_3$ same as insertion process.
5: Arrange vertices of $S_1$ and $S_2$ according to insertion process. Vertices of $S_1$ are arranged according to asscending order of curvature value while vertices of $S_2$ are arranged according to an ascending order of the index value.
6: Subtract the vertex normals of $O$ from the vertex normal of $O^w$ respectively according to the arrangements of sets.
7: Obtained result is compared with the secure hash code.
8: Comparison results authenticate the object (if sequence of bits match with the hash code) otherwise report tempering.
```

4. Metrics used for Evaluation

We have run our algorithm on six different 3D polygon mesh objects : Armadillo(30995 vertices and 61986 faces), duck (2108 vertices and 4202 faces), Bunny (34834 vertices and 69451 faces), Cow (2800 vertices and 5596 faces), Dragon (35000 vertices and 70216 faces) and Nefertiti (654 vertices and 1252 faces) as shown in Fig. 2. Different secure images of varying sizes produce different secure codes. These secure codes are taken as watermark primitive to decide which vertices are to be selected for watermark embedding. The cryptographic hash function (MD5) produces 32 words of 8 bits i.e. 256 bit hash code. Each generated hash code is unique which is dependent on the secure image I. The hash code obtained are partitioned into two parts($W1$ and $W2$) which are mapped to S_2 and S_1 group repeatedly for selecting vertices for watermark embedding. The number of '1's in a hash code decides the number of selected vertices and their positions for watermark embedding. In the proposed work image lena.jpg of size 128 × 128 is taken as secure image which consists of 119 '1's in the hash code obtained.

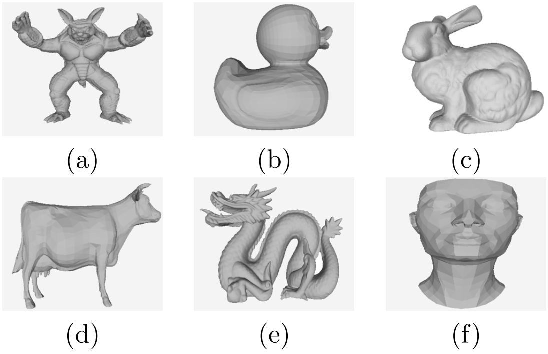


Fig. 2. Original 3-D mesh objects (a) Armadillo, (b)Duck, (c) Bunny, (d) Cow, (e) Dragon (simplified), (f) Nefertiti.

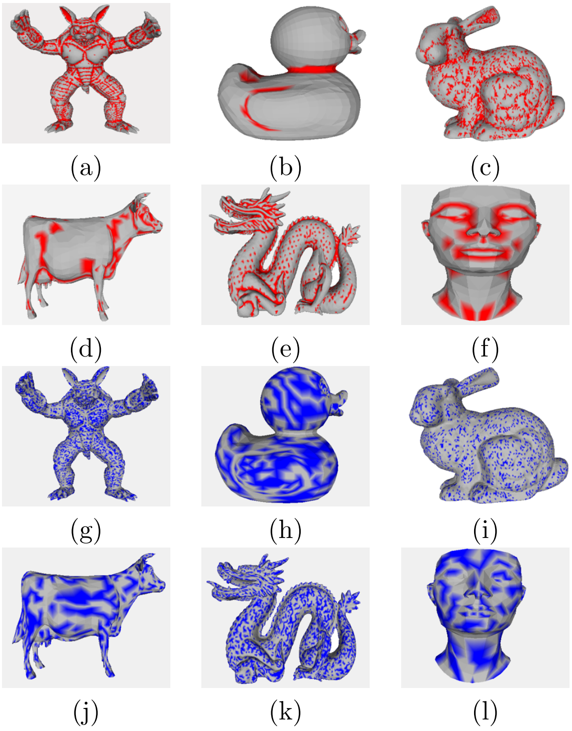


Fig. 3. In (a)-(f) Red area shows Mean Curvature area, while in (g)-(l) Blue area shows uniform distribution of watermark.

Table 1 shows distortion in watermark object w.r.t. cover objects in terms of Hausdorff distance, RMS, Mean Observation Score(MoS) and also % change in surface area of watermarked object w.r.t. original object. The results of Table 1 also represent the values of correlation factor, which reports the % of watermarked vertices authenticated. The algorithm is able to retrieve 100% watermark information when there is no attack. The vertices are repositioned from their original position below this threshold value. All the six different 3D polygon mesh objects are watermarked with strength 3.5% for S_1 and 3.0% for S_2, which specifies that all selected vertices of S_1 and S_2 are repositioned from their original position by value(weight) equal to the strength.

In the proposed scheme perceivable visual quality is the main concern besides robustness and security. The distortion due to watermarking is assessed through objective measurements along with subjective measurement[3].

Here, the distortion due to watermark embedding is measured in terms of objective measurement (Hausdorff Distance, Root Mean Square error) and subjective measurement (MoS).

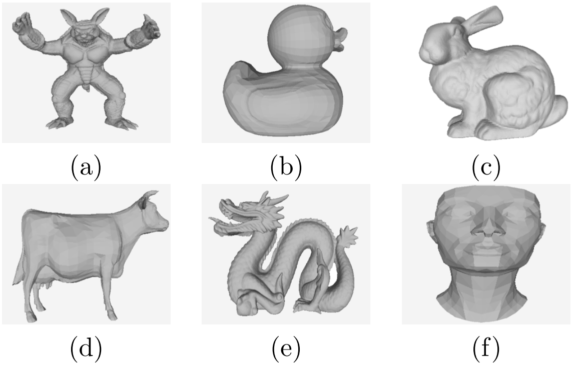


Fig. 4. Watermark 3-D mesh Objects: (a) Armadillo, (b) Duck, (c) Bunny, (d) Cow, (e) Dragon (simplified), (f) Nefertiti.

4.1. Hausdorff Distance

Hausdorff distance between two sets of points is defined as the maximum distance of a set to the nearest point in the other set[18]. Hausdorff distance estimates the extent to which each point of a object set lies near some point of another object set and vice versa[19]. This distance is used to estimate the degree of resemblance between two objects that are superimposed on one another. The objective is to minimize the Hausdorff distance to reduce the degree of mismatch between cover object(O) and watermarked object(O^w) [20], [18].

Let e(p,O) represent the distance of a point p in 3D space from the 3D object O as [21]:

$$e(p, O) = \min_{v_i^O \in O} \{d(p, v_i^O)\}, \tag{6}$$

where $d(p, v_i^O)$ is the Euclidian distance between v_i^O, i^{th} vertex of object O and p. Then the Hausdorff distance between two 3D objects O and O^w is:

$$H_a(O, O^w) = \max_{v_i^O \in O} \{e(v_i^O, O^w\}. \tag{7}$$

This distance is not symmetrical *i.e.* $H_a(O,O^w) \neq H_b(O,O^w)$. $H_a(O,O^w)$ and $H_b(O,O^w)$ are referred as forward and backward distance respectively[21].

The symmetrical Hausdorff distance can be defined as:

$$H_d(O,O^w) = \max(H_a(O,O^w), H_a(O^w,O)). \tag{8}$$

The Symmetrical Hausdorff distance reports more accurate measurement of the error between two surfaces as computation of a "one-sided" error can lead to significantly underestimated distance value[18].

4.2. Root Mean Square Error (RMS)

The Root Mean Square error is based on the correspondence between each pair of vertices of the objects to compare, thus it is limited to the comparison between two meshes sharing the same topology [21], [18]. The root mean square error is evaluated as :

$$d_{rms}(O,O^w) = \sqrt{\sum_{i=1}^{n} ||v_i^O - v_i^{O^w}||^2}, \tag{9}$$

where n is number of vertices of mesh and v_i^O is a vertex of O corresponding to the vertex $v_i^{O^w}$ of O^w.

4.3. Subjective Assessment

Actually, there is no Human Visual System (HVS) or automated tool available for assessing the perceivable visual quality of 3D objects. There is no specified range of quality parameter to decide the quality of 3D objects due to discrete nature of vertices of different 3D mesh objects. It is also observed that some time although objective parameter shows good results but object is not of good perceivable visual quality. Therefore, the subjective assessment is equally important as objective assessment for assessing the quality of watermarked objects. The subjective assessment is performed under standard conditions suggested by Corsini et al.[3] for perceptual metrics obtained for the quality assessment of watermarked 3D mesh objects. The watermarked 3D object is manually compared w.r.t. original object besides objective measurement[22]. In subjective measurement Mean Observation Score (MoS) is measured for certain view dependent 2D images rendered from 3D object as well as view-independent error matrix obtained from watermarked 3D mesh object[22]. In the subjective assessment, watermark objects are manually compared with original object. The comparison between both the objects are performed by taking different objects from different angles.

Inspection of tested objects are performed manually by a selected group of students. These students are trained about the 3D objects, processing operations, structure, noise and distortion. They are instructed to assign score to the pair of objects between 0-10. Score 0 and 10 represent lowest and highest scores given to each subject by human observer. Mean Observation Score (MoS) is the result of group of human observers for different pair of test objects.

$$MoS_i = \frac{1}{n}\sum_{j=1}^{n} m_{ij}. \tag{10}$$

MoS_i is mean observation score of i^{th} object by n different test observers. m_{ij} is score in the range [0-10] given by j^{th} observer to i^{th} test object[21]. MoS as shown in Table 1 shows that watermark objects are of good visual quality w.r.t. original object when there is no attack on watermarked objects.

5. Result Analysis

The main aim of the proposed work is the authentication of original watermark object. 3D polygon meshes are tested against various distortion and distortion-less attacks. 3D object are authenticated by correlation factor, which is equivalent to Bit Error Rate (BER) or detection ratio. The BER is the ratio between the number of bits that are correctly detected and the total number of bits embedded. The correlation value lies between [0, 1]. The correlation value '1' signifies that 100% of the watermark information is extracted.

The watermark inserted is invariant to translation, rotation and uniform scaling (RST) as ratio of the distance between center of gravity and vertex remain the same.

$$\rho_i' = \rho_i + W. \tag{11}$$

After scaling by factor t vertex normal becomes $\rho_i'' = (\rho_i' * t)$ or

$$\rho_i'' = (\rho_i + W) * t, \tag{12}$$

where ρ_i, ρ_i' and ρ_i'' represent the vertex normal distances of cover object, watermark object and attacked watermark object respectively.

The uniform scaling reflects the constant ratio of normal distance of vertices of distorted and watermarked object, which is computable. Thus, the watermark information is not degraded by uniform scaling. Similarly, normal distance of the vertices from center of gravity remains same in transformation and rotation attacks preserving the shape of 3D mesh object. In RST, getting original watermarked object is time consuming process. Using the property of constant ratio or distance of each vertex from center of gravity of 3D mesh, all the vertices are again rearranged taking help from cover object and watermark object. In RST attack, we are able to retrieve 100% of watermark information.

The non-blind nature of the proposed work make the algorithm robust and secure against vertex reordering attack. The preprocessing steps make the algorithm robust against distortion and distortion-less attacks.

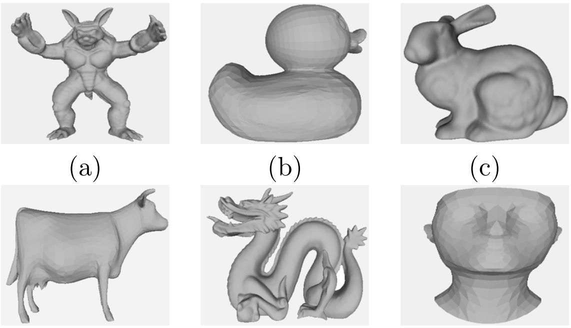
Cropping is the removal of any part/parts of a object. Cropping destroys the watermark to some extend but the remaining portion of the 3D mesh authenticates the validity/presence of the watermark inserted.

Table 1. Evaluation of robustness of 3D watermark mesh without attack

Model	$H_s(O,O^w)$	$d_{rms}(O,O^w)$	MoS	correlation	%Change in Area
Armadillo	0.5116	0.36×10^{-1}	9.876	1.0	0.1474
Duck	0.2672	0.47×10^{-1}	9.823	1.0	0.0738
Bunny	0.4924	0.32×10^{-4}	9.746	1.0	0.1198
Cow	0.0057	0.16×10^{-2}	9.653	1.0	0.0326
Dragon	0.0017	0.35×10^{-4}	9.725	1.0	0.1217
Nefertiti	0.0026	0.37×10^{-3}	9.362	1.0	0.0367

In Table 1, the Hausdorff distance and RMS error report distortion between original and watermark objects. There is no specific range of parameter to decide the affect of distortion on watermark objects. Therefore, these objects are manually accessed by subjective measurement in terms of MoS. MoS in Table 1 indicates good quality of watermarked object w.r.t. original object.

Smoothing attack minimizes the surface levels by removing some vertices as shown in Fig. 5. Table 2 represents the performance of the proposed watermarking algorithm because of smoothing attack in terms of correlation value. The smoothing over the watermark object is applied using Taubin Smoothing in MeshLab open source[23]. The results shown in Table 2 explain that the proposed algorithm is robust against smoothing attack as watermark can be retrieved. Cow object shows sensitivity towards smoothing attack as only 38% of watermark vertices can be matched in attacked watermarked object as shown in Table 2. While in case of Bunny object, we are able to get 68% of watermarked information after smoothing attack. Table 2 shows distortion between cover object(O) and attacked watermark object($O^{w'}$) in term of hausdorff distance and Root Mean Square error.



(a) (b) (c)

(d) (e) (f)

Fig. 5. Smoothing Attack on watermark 3-D mesh objects.

Subdivision attack enhances the surface smoothness by increasing the density of the vertices on the surface iteratively as shown in Fig. 6. Table 3 shows the performance after subdivision attack in terms of numerical distortion(Hausdorff distance, RMS) and correlation value. The result shows the good degree of robustness in terms of correlation factor. The numerical distortion(Hausdorff distance, RMS) is also measured between cover object and attacked watermark object as shown in Table 3. It is observed that 96%, 98%, 93%, 85%, 89% and 78% of watermark information is authenticated from Armadillo, Duck, Bunny, Cow, Dragon and Nefertiti respectively after applying subdivision attack. The Nefertiti is more sensitive as compared to other objects against subdivision attack while in Armadillo, Duck and Bunny more than 90% of watermark is authenticated.

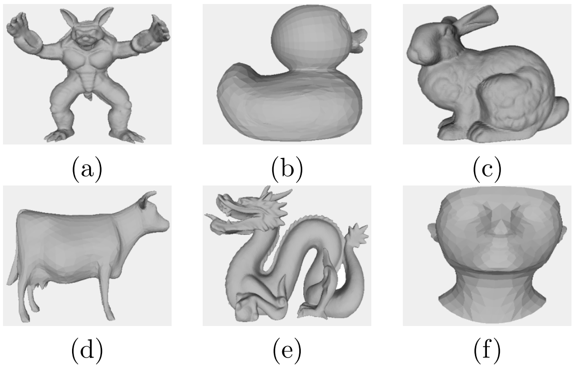


Fig. 6. Subdivision Attack on watermark 3-D mesh objects.

Simplification is a process of removing some vertices and faces maintaining the shape of the object as shown in Fig. 7. Table 4 shows the robustness against simplification attack. Objects are simplified using Mesh Lab open source[23]. The watermark embedded in curvature area is found to be more resistant against simplification attack. In the proposed method, one set of vertices for watermark embedded belongs to curvature region of the surface, which helps to retrieve the watermark vertices even after simplification attack. The watermark information could be authenticated even after simplifying the watermarked objects up to 90%. The high density (payload) of watermark enables to determine the watermark vertices even after simplification up to 90%.

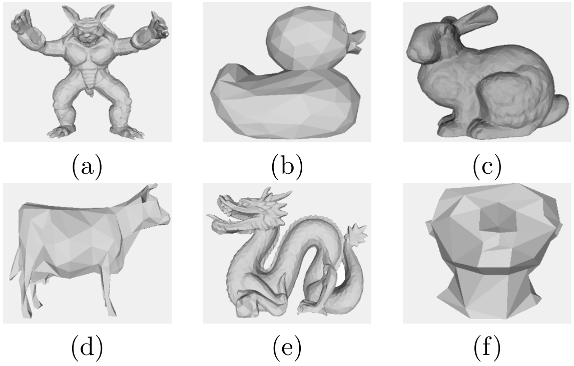


Fig. 7. Simplification Attack on watermark 3-D mesh objects.

Tables 1–4 show the correlation factor and visual quality of watermark object w.r.t. original object. Correlation factor reports the robustness of the algorithm.

The visual quality is simply the degree of similarity between original object and watermark object. Table 1 shows that 100% of watermark can be authenticated when there is no attack applied on watermark 3D objects. These results also report that distortion between original and watermark object is acceptable. As such there is no range of parameter (Hausdorff distance, RMS error) to decide the degree of distortion but it varies from object to object. The final watermark object is accessed manually by subjective analysis. MoS in Table 1 informs the good degree of similarity between original and watermark object. Tables 2–4 describe that distortion and robustness against smoothing, subdivision and simplification. The results from these tables shows the watermark objects are sensitive against these attacks in terms of robustness and distortion. These attacks destroy the geometry/ shape of the object to some extend therefore these are also called distortion attack, even then we are able to authenticate watermark information. Hausdorff distance and RMS error of Tables 2–4 are compared with Table 1, it informs that distortion increases when distortion attacks are applied over watermark objects.

In the present watermarking algorithms [2], [10], [11], [12] for 3D polygon mesh objects, selection of vertices for watermark embedding are independent on the geometrical properties/ shape of 3D mesh object *i.e.* perceivable distortion is not considered during the selection of vertices for watermark embedding. In our detailed analysis, we have compared our proposed method with Cho et al. [10] methods and found that watermarked object obtained by Cho et al. [10] produces staircase like distortion on the surface of 3D mesh object. While, our watermarked objects are free from such distortion. MoS shown in Table 1 validate the results. Our proposed method, also produces less Root Mean Square (RMS) error than the method suggested by Cho et al.[10]. The proposed method is also robust against different distortion (simplification, subdivision, smoothing) and distortion-less (Rotation, Uniform Scaling, Transformation) attacks.

Table 2. Robustness measure against smoothing attack

Model	$H_s(O,O^{w'})$	$d_{rms}(O,O^{w'})$	correlation value
Armadillo	0.659	0.90×10^{-1}	0.57
Duck	0.928	0.69×10^{0}	0.56
Bunny	0.876	0.21×10^{-4}	0.68
Cow	0.249	0.21×10^{-1}	0.38
Dragon	0.002	0.98×10^{-4}	0.46
Nefertiti	0.040	0.78×10^{-2}	0.47

Table 3. Robustness measure against subdivision attack

Model	Original	Modified	$H_s(O,O^{w'})$	$d_{rms}(O,O^{w'})$	correlation
Armadillo	30995	185960	0.5136	0.35×10^{-1}	0.96
Duck	2108	12638	0.7381	0.47×10^{0}	0.98
Bunny	34834	139118	0.5526	0.21×10^{-4}	0.93
Cow	2800	16790	0.0081	0.16×10^{-2}	0.85
Dragon	35000	210540	0.0017	0.35×10^{-4}	0.89
Nefertiti	654	3850	0.0028	0.36×10^{-3}	0.78

Table 4. Robustness measure against simplification attack

Model	simplify	$H_s(O,O^w)$	$d_{rms}(O,O^{w'})$	correlation
Armadillo	30.00%	0.512	0.37×10^{-1}	0.91
	50.01%	0.641	0.71×10^{-1}	0.71
	70.01%	0.691	0.91×10^{-1}	0.53
	90.02%	0.727	0.12×10^{0}	0.32
Duck	30.03%	0.367	0.51×10^{0}	0.91
	50.05%	0.353	0.61×10^{0}	0.68
	70.07%	0.364	0.61×10^{0}	0.53
	90.09%	0.723	0.83×10^{1}	0.38
Bunny	30.08%	0.335	0.79×10^{-4}	0.96
	50.14%	0.344	0.89×10^{-4}	0.83
	70.06%	0.373	1.019×10^{-4}	0.64
	90.03%	0.392	3.12×10^{-4}	0.47
Cow	30.00%	0.019	0.26×10^{-2}	0.71
	50.00%	0.024	0.33×10^{-2}	0.51
	70.00%	0.027	0.46×10^{-2}	0.46
	90.00%	0.301	0.29×10^{-1}	0.33
Dragon	30.00%	0.002	0.40×10^{-3}	0.91
	50.00%	0.002	0.66×10^{-3}	0.81
	70.00%	0.002	0.09×10^{-2}	0.56
	90.00%	0.003	0.14×10^{-2}	0.43
Nefertiti	30.28%	0.004	0.48×10^{-3}	0.77
	50.46%	0.005	0.91×10^{-3}	0.54
	70.64%	0.051	0.98×10^{-2}	0.41
	90.82%	0.067	0.12×10^{-1}	0.29

6. Conclusion

We have proposed a non-blind, secure and robust watermarking algorithm in spatial domain. The proposed algorithm is based on geometrical properties of 3D mesh object. The 3D mesh is then processed to identify those areas which reflect less perceivable distortion after watermark embedding. The watermark is embedded in the selected vertices based on hash code obtained from secure image and cryptographic hash function. In the proposed algorithm repetitive insertions of watermark information is done in different segments of 3D object. The watermark is embedded by re-positioning the selected vertices from their original positions. The secure image, strength of watermark (amount of re-position) applied over vertices are kept as secret keys.

The proposed watermarking scheme is also robust against various distortion and distortion-less attacks. In case of distortion attacks like smoothing, simplification, subdivision; watermark is distorted to some extend. The major watermarking information is extracted from the curvature region of attacked watermark object. The execution time of watermarking scheme depends on the total number of numbers of vertices and numbers of vertices selected for watermark embedding of 3D mesh.

Acknowledgment. The Author would like to thank Stanford 3D Scanning Repository for providing 3D Model for experimental purpose. We would also like to thank P. Cignoni et al. for providing Metro V4.06 tool [24] for measuring objective quality parameter.

References

[1] Ichikawa S., Chiyama H., Akabane K., *Redundancy in 3D polygon models and its application to digital signature*, Journal of WSCG, **10**(1):225–232, 2002.

[2] Wang K., Lavoué G., Denis F., Baskurt A., *A comprehensive survey on three-dimensional mesh watermarking*, IEEE Transactions on Multimedia, **10**(8):1513–1527, 2008.

[3] Corsini M., Gelasca E. D., Ebrahimi T., Barni M., *Watermarked 3-D mesh quality assessment*, IEEE Transactions on Multimedia, **9**(2):247–256, 2007.

[4] Wang K., Lavoué G., Denis F., Baskurt A., *Robust and blind mesh watermarking based on volume moments*, Computers & Graphics, **35**(1):1–19, 2011.

[5] Corsini M., Larabi M.-C., Lavoué G., Petrík O., Váša L., Wang K., *Perceptual metrics for static and dynamic triangle meshes*, Eurographics 2012-State of the Art Reports, pp. 135–157, The Euro-graphics Association, 2012.

[6] Karni Z., Gotsman C., *Spectral compression of mesh geometry*, Proceedings of the 27^{th} annual conference on Computer graphics and interactive techniques, pp. 279–286, 2000.

[7] Lavoué G., Gelasca E. D., Dupont F., Baskurt A., Ebrahimi T., *Perceptually driven 3d distance metrics with application to watermarking*, *Optics & Photonics*, pp. 63120L–63120L. International Society for Optics and Photonics, 2006.

[8] Dugelay J.-L., Baskurt A., Daoudi M., *3D Object Processing: Compression, Indexing and Watermarking*, Wiley, 2008.

[9] Ohbuchi R., Masuda H., Aono M., *Watermarking three-dimensional polygonal models through geometric and topological modifications*, IEEE Journal on Selected Areas in Communications, **16**(4):551–560, 1998.

[10] Cho J.-W., Prost R., Jung H.-Y., *An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms*, IEEE Transactions on Signal Processing, **55**(1):142–155, 2007.

[11] Wang W.-B., Zheng G.-Q., Yong J.-H., Gu H.-J., *A numerically stable fragile watermarking scheme for authenticating 3D models*, Comput. Aided Des., **40**(5):634–645, May 2008.

[12] Motwani R., Harris F., *Robust 3D watermarking using vertex smoothness measure*, Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition, 2009.

[13] Maret Y., Ebrahimi T., *Data hiding on 3D polygonal meshes*, Proceedings of the 2004 Workshop on Multimedia and Security, pp. 68–74, ACM, 2004. 2004.

[14] Alface P. R., Macq B., *Blind watermarking of 3D meshes using robust feature points detection*, IEEE International Conference on Image Processing, 2005, ICIP 2005, vol. **1**, pp. 682–693, 2005.

[15] Johnson A. E., Kang S. B., *Registration and integration of textured 3D data*, Image and vision computing, **17**(2):135–147, 1999.

[16] Turk G., *Re-tiling polygonal surfaces*, ACM SIGGRAPH Computer Graphics, vol. **26**, pp. 55–64, ACM, 1992.

[17] Meyer M., Desbrun M., Schröder P., Barr A. H., *Discrete differential-geometry operators for triangulated 2-manifolds*, Visualization and mathematics, **3**(7):34–57, 2002.

[18] Aspert N., Santa-Cruz D., Ebrahimi T., *Mesh: Measuring errors between surfaces using the Hausdorff distance*, Proceedings of 2002 IEEE International Conference on Multimedia and Expo, ICME'02, vol. **1**, pp. 705–708, 2002.

[19] Huttenlocher D. P., Klanderman G. A., Rucklidge W. J., *Comparing images using the Hausdorff distance*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **15**(9):850–863, 1993.

[20] Wu W., *Quantized Gromov-Hausdorff distance*, Journal of Functional Analysis, **238**(1):58–98, 2006.

[21] Lavoué G., Corsini M., *A comparison of perceptually-based metrics for objective evaluation of geometry processing*, IEEE Transactions on Multimedia, **12**(7):636–649, 2010.

[22] Bulbul A., Capin T., Lavoué G., Preda M., *Assessing visual quality of 3D polygonal models*, Signal Processing Magazine, IEEE, 28(6):80– 90, 2011.

[23] Cignoni P., Corsini M., Ranzuglia G., *Meshlab: an open-source 3D mesh processing system*, Ercim News, **73**:45–46, 2008.

[24] Cignoni P., Rocchini C., Scopigno R., *Metro: measuring error on simplified surfaces*, Computer Graphics Forum, vol. **17**, pp. 167–174, Wiley Online Library, 2001.