IMPERCEPTIBLE WATERMARKING SCHEME FOR 3D TRIANGULAR MESH

Lokendra Kumar Sharma[1] **and Dr. D. B. Ojha**[2]

[1]Research Scholar, Department of CSE, Mewar University, Chittorgarh, Rajasthan, India

[2]Director Research, Mewar University, Chittorgarh, Rajasthan, India

ABSTRACT

The proposed method for watermark insertion in 3D polygon mesh is based on its geometrical properties. In this method selected vertices (geometrical attribute) are shifted from its original position. The changes in position of vertices w.r.t. original position is considered as watermark. The selection of vertices for shifting from their original position is dependent on the number normalization property of the vertices. The Cartesian coordinate of vertices are converted into spherical coordinate. The vertices are shifted from its original position by displacing vertex normal distance (ρ) component of (ρ,θ,ϕ) spherical coordinate of vertices. The watermark information is uniformly distributed through out the surface of 3D mesh. The shifting of vertices are done to establish a tradeoff among payload, robustness, security and quality. The proposed scheme is evaluated against distortion attacks.

Keywords: 3D Mesh Objects, Hausdorff distance, RMS, Number Normalization

1. INTRODUCTION

In the growing era of internet and wide accessability of data items over internet, increase the misuse of these data items. These data can be text, image, video, 3D object or any thing available on internet. Now, it became a challenge to protect these data items from misuse. These data items can be protected by using Intellectual Property Right (IPR). Cryptography, digital signature [1], stenography or watermarking are different schemes used for IPR.

In the proposed work, we are focusing on security of 3D objects. Extensive market growth of 3D object in last decades focuses attention on security issues of these objects. 3D object are widely used in architecture design, machine design, cultural heritage and entertainment. 3D objects specialized data which can be viewed any any angle of object i.e we can see the object from front, back, side, top, lower or any other view. These 3D objects are basically categorized into two classes: Volume based or surface based objects. In the proposed work, we are considering 3D triangular mesh objects, special class of surface base objects.

3-D modeling is a process of developing a mathematical representation of any 3-D surface of object(either inanimate or living) via specialized software like CAD/CAM. The product is called a 3-D model or 3-D object. These 3D triangular mesh objects are categorized by these surface property while objects which are categorized by their volume are called volume based objects [2].Watermarking is an art of hiding secondary data on the primary data by maintaining perceived quality of primary data [3]. The watermarking scheme is designed to check the copy-right ownership (robust watermarking) or to verify the authentication (fragile watermarking) [4]. Watermarking techniques are being used for copy right protection and the integrity of digital contents.

The review of current 3-D watermarking techniques [5] motivates to develop a watermarking technique in spatial domain which should be robust, non-blind, imperceptible and secure. Different watermarking algorithms have been summarized [6] with different pros and cons in spatial and spectral domain. Spatial domain watermarking algorithms produce output of better quality [7] than watermarking algorithms of spectral domain. The non-blind nature of an algorithm makes it robust [5] [8] against different types of attacks. Usually, non-blind watermarking algorithms are used for authentication purpose [9].In 3D triangular mesh objects vertices and faces are geometrical and topological attributes respectively. We consider vertices for watermark embedding as watermark insertion using geometrical attributes are much robust as compare to the topological attributes [10].

2. WATERMARKING ALGORITHM

3D polygonal mesh is a collection of vertices and edges. The edges are connected to form the faces. Faces can triangular, quadrilateral or any polygon. we consider tringular unit of face as triangular face forming fine shape of polygon mesh objects. 3D mesh can be represented as G $\equiv$ (V,F) where V is a finite set of vertices and F is a finite set of faces [11]. Mathematically, it can be shown as V= $\{V_i \in R^3 \mid 1 \leq$ i $\leq$ N $\}$ where, N is number of vertices and F= $\{F_j \mid 1 \leq j \leq$ M$\}$ where M is total number of faces of the mesh. 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 indices of the mesh vertices. In 3D mesh edges and faces are topological data while vertices are the geometrical data. Modification in topological data does not change the shape of the object while any change in geometrical data do some changes in the shape of the 3D object [12]. The watermark insertion using topological properties is not resistant against different attacks like re-meshing, simplification etc. Therefore, geometrical attributes are considered for watermark embedding as it produces comparably more robust watermarking algorithm.

In the proposed watermarking of 3D triangular mesh, watermark is embedded by shifting the selected vertices from their original positions. The amount of repositioning of selected vertices are done such that less distortion is produced. The vertices are selected for watermark embedding based on number property of vertices.

A. Floating point representation of Number

The floating point representation of a number has two parts : unsigned mantissa and exponent. In floating point number representation a number is said to be normalized if most significant bit of mantissa is non-zero. In the proposed work, we use characteristics of floating point representation for identifying marked vertices for watermark embedding.

B. Preprocessing

Initially, the center of mass(C_m) of 3D mesh is determined. The center of mass(C_m) of the 3D mesh is evaluated as:

$$
\begin{aligned}
x_{cm} &= \frac{\sum_{i=1}^{n} x_i}{n} \\
y_{cm} &= \frac{\sum_{i=1}^{n} y_i}{n} \\
z_{cm} &= \frac{\sum_{i=1}^{n} z_i}{n}
\end{aligned} \tag{1}
$$

Here, x_{cm}, y_{cm} and z_{cm} represent the coordinates of center of mass (C_m).

The cartesian coordinates (x_i,y_i,z_i) of the vertices are converted into spherical coordinates (ρ_i,θ_i,ϕ_i) as:

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

$$\theta_i = \tan^{-1} \frac{y_i - y_{cm}}{x_i - x_{cm}} \tag{3}$$

$$\phi_i = \cos^{-1} \frac{z_i - z_{cm}}{\rho_i} \tag{4}$$

The first component ρ of spherical coordinates represents the normal distance of a vertex from the center of mass of 3D triangular mesh. During the watermark embedding only ρ's of selected vertices are modified to maintain minimum distortion. A watermarking algorithm generally consists of two modules named watermark embedding module and watermark authentication module.

C. Watermarking Embedding

The proposed non-blind watermarking algorithm in spatial domain directly inserts the information into the vertices. This scheme is geometry driven i.e. connectivity/topology of the vertices are not modified. The vertices are selected for watermark embedding according to the floating point representation property of vertex normal of selected vertices. The selected vertices of 3D mesh are displaced from their original positions by modifying ρ components of (ρ,θ,ϕ) spherical coordinates of vertices. The changes in positions of selected vertices of original mesh object is the watermark. 3D mesh is first preprocessed and center of mass (C_{cm}) of the object is calculated. The Cartesian coordinates of the vertex v_i =(x_i,y_i,z_i) are converted into spherical coordinates (ρ_i,θ_i,ϕ_i) as per equations - 2, 3, 4, where ρ_i represents the vertex normal distance of i^{th} vertex from center of mass. The vertices are selected in 3D mesh object uniformly throughout the surface. The selected vertices are shifted from their original positions by modifying the ρ with weight 'W' beyond limiting value. The limiting value of 'W' is the value beyond which 3D object gets distorted. The modified Cartesian coordinates of watermark object is evaluated from modied ρ_i to generate watermark object.

Algorithm 1 Watermarking in 3D triangular mesh using number normalization property

```
INPUT: 3D mesh object
OUTPUT: Watermarked 3D object
  Calculate the center of mass for the 3D polygon mesh ($O$)
  for each vertex do
    Calculate spherical coordinates ($\rho_i$,$\theta_i$,$\phi_i$) from cartesian coordinates of the vertex ($x_i$, $y_i$, $z_i$)
    Normalize $\rho_i$
    if $n^{th}$ bit of mantissa of $\rho_i$ is '1' then
      $\rho'_i = \rho_i + W$
    end if
    Re-calculate cartesian coordinates ($x'_i$,$y'_i$,$z'_i$) from spherical coordinates
    Reposition the vertices to obtain watermark triangular mesh object
  end for
```

D. Watermark Authentication

Watermarking algorithms can be either blind or non-blind in nature. When original 3D mesh is required along with watermark 3D mesh object for watermark extraction is said to non-blind algorithm while in blind algorithm only watermark object is sufficient for watermark extraction. The proposed algorithm is non-blind in nature as both original and watermark model is needed at the time of watermark authentication. At the time of authentication, watermark operation is performed on original object similar to the insertion step and modified vertices are sequentially compare with vertices of watermark object. The comparison reports number of vertices which is equal to the modified vertices in watermark object. It also reports region of tempering.

Algorithm 2 *Watermark authentication in 3D triangular mesh*

```
INPUT: Original 3D triangular mesh $O$, watermarked 3D object($O^w$), secret key
OUTPUT: Authentication result
  Load the original($O$) and watermarked ($O^w$) 3D triangular mesh object
  for each vertex of 3D triangular mesh ($O$) and watermark 3D object ($O^w$) do
    Convert cartesian coordinate of the vertices ($x_i$, $y_i$, $z_i$) into spherical coordinate
    Calculate the vertex normal distances of watermarked object w.r.t center of mass of original object
    Subtract the weight 'W' of selected vertices of watermark object
    Compare the vertices of original and watermarked object
    Comparison results authenticate the object
  end for
```

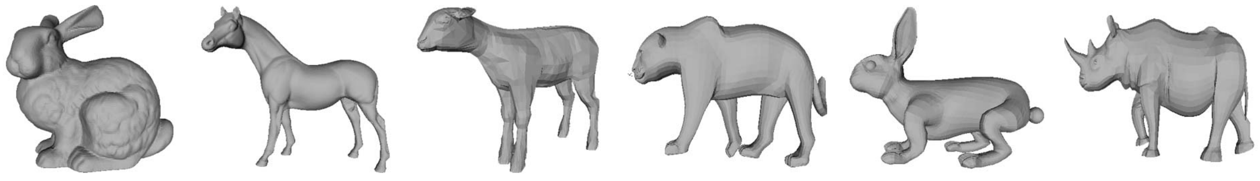

(a) Bunny (b) Horse (c) Lamb (d) Leopard (e) Rabbit (f) Rhino

Fig. 1: Original 3-D mesh objects

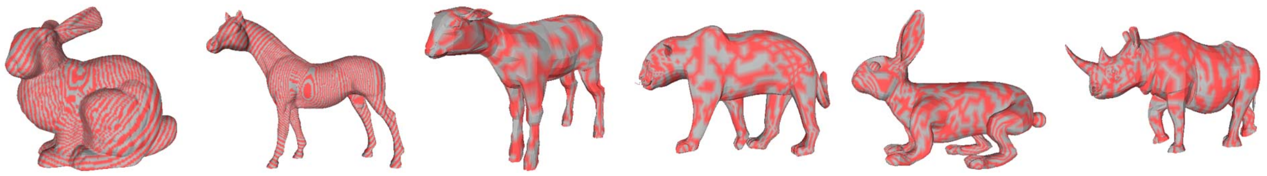
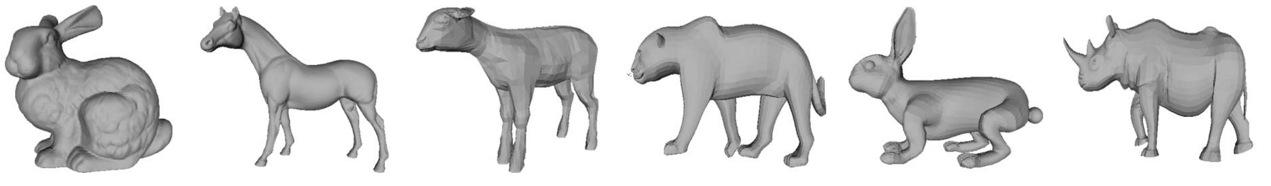

(a) Bunny (b) Horse (c) Lamb (d) Leopard (e) Rabbit (f) Rhino

Fig. 2: Selected vertices of 3-D mesh objects indicated in red color



(a) Bunny (b) Horse (c) Lamb (d) Leopard (e) Rabbit (f) Rhino

Fig. 3: Watermarked 3D mesh objects

3. Result Analysis

The watermarking algorithm is performed on different 3D mesh objects Bunny ($v = 34834, f = 69451$), Horse($v = 48484, f = 96964$), Lamb($v = 2995, f = 3960$), Leopard($v = 7302, f = 11508$), Rabbit($v = 14411, f = 14976$), Rhino ($v = 6459, f = 9374$) as shown in Figure-1. These object are taken from www.archive3d.net for experimental purpose. We have selected these 3D mesh objects due to variation in parameters (number of vertices, structure, surface roughness etc) of 3D mesh objects.

In the proposed watermarking algorithm, 50% vertices are selected for watermark insertion approximately. The distortion is measured between original mesh and watermark mesh. Hausdorff distance, Root Mean Square Error (RMS), surface roughness are basic parameters for measuring the distortion. Hausdorff distance, Root Mean Square Error (RMS) are measured by Metro Tool [13]

In the watermark mesh object, watermark information can be completely retrieved. Correlation factor 100% signifies that 100% of watermark vertices are identified correctly. Similarly, correlation factor is measured against different attacked watermark mesh objects. Robustness of an algorithm is directly proportional to the correlation factor. We have shifted the selected vertices from their original position by some fixed amount (W) and the amount of shifting of vertices must be less than the limiting value. The vertices of 3D mesh objects are shifted to 10%, 20% and 25% from their original value. The distortion increases by increasing the weight.

Bunny 3D mesh object has rough surface, Horse has smooth surface, Lamb has bumpy surfaces, Rabbit and Rhino has sharp pointed edges. These selected 3D mesh objects also varies in number of vertices and faces. Hausdorff distance and RMS are the parameters for evaluating the distortion introduced due to watermark insertion as shown in Table-I. The correlation factor is also determine on watermark model without any attack.

A. 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 [14]. Hausdorff distance estimates the extent to which each point of a object set lies near some point of another object set and vice versa [15]. This distance is used to estimate the degree of resemblance between two objects that are superimposed on one another. Informally, two sets of points are close if each point of either set is close to some point of the other set. The objective is to minimize the hausdorff distance to reduce the degree of mismatch between cover object(O) and watermarked object(O^w) [14] [16].
Let e(p,O) represent the distance of a point p in 3-D space from the 3-D object O as [6]:

$$e(p, O) = \min_{v_i^O \in O} \{d(p, v_i^O)\} \tag{5}$$

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

$$H_a(O, O^w) = \max_{v_i^O \in O} \{e(v_i^O, O^w\} \tag{6}$$

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 [6].
The symmetrical Hausdorff distance can be defined as:

$$H_d(O, O^w) = \max(H_a(O, O^w), H_a(O^w, O)) \tag{7}$$

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 [14].