A blind 3D watermarking technique using spherical coordinates and skewness measure

Alaa E. Abdallah*, Emad E. Abdallah, Mohammad Bsoul, Ahmad Al-Khasawneh and Lina Amer

Faculty of Prince Al-Hussein Bin Abdullah II For Information Technology, Hashemite University, P.O. Box 150459, Zarqa 13115, Jordan
E-mail: aabdallah@hu.edu.jo
E-mail: emad@hu.edu.jo
E-mail: mbsoul@hu.edu.jo
E-mail: Akhasawneh@hu.edu.jo
E-mail: lina_it2008@yahoo.com
*Corresponding author

Abstract: Three-dimensional watermarking is very useful for copyright protection, verification, and indexing. In this paper we propose a robust computationally inexpensive 3D polygonal mesh watermarking methodology. The new algorithm is based on vertex distribution and skewness measure. The core idea behind our algorithm is to modify the skewness measure based on a predefined secret message. The proposed algorithm is slightly altering the skewness distribution of several intervals extracted from a 3D model. The main attractive features of this approach is the improved performance of the data embedding system, perceptual invisibility and it is resistant to a variety of the most common attacks.

Keywords: watermarking; three-dimensional models; perceptual invisibility; spherical coordinates; skew; 3D attacks.


Biographical notes: Alaa E. Abdallah is an Assistant Professor in the Department of Computer Science of Hashemite University since 2011. He obtained his BSc in Computer Science from Yarmouk University in 2000, MSc in Computer Science from University of Jordan in 2003, and PhD in Computer Science from Concordia University in 2008, Montreal-Canada. Prior to joining Hashemite University, he was a Network Researcher at consulting private company in Montreal (2008–2011). His research interest includes the routing protocols for ad hoc networks, parallel and distributed systems, and multimedia security.

Emad E. Abdallah is currently an Assistant Professor in the Department of Computer Information Systems and vice dean of Prince Al-Hussein bin Abdullah II Faculty of Information Technology at the Hashemite University (HU), Jordan. He received his PhD in Computer Science from Concordia University in 2008, where he worked on multimedia security, pattern recognition and 3D object recognition. He received his BS in Computer Science from Yarmouk University, Jordan, and MS in Computer Science from the University of Jordan in 2000 and 2004, respectively. Prior to joining HU, he was a Software Developer at SAP Labs Montreal. His current research interests include computer graphics, multimedia security, pattern recognition and computer networks.

Mohammad Bsoul is an Associate Professor in the Computer Science Department at the Hashemite University. He received his BSc in Computer Science from Jordan University of Science and Technology, Jordan, Master from University of Western Sydney, Australia, and PhD from Loughborough University, UK. His research interests include wireless sensor networks, grid computing, distributed systems and performance evaluation.

Ahmad Al-Khasawneh is an Associate Professor at Hashemite University and currently acting dean of Prince Al-Hussein bin Abdullah II Faculty of Information Technology. He holds a PhD of Information Systems, MS in Information Technology and Computer Engineering both from Newcastle University, Australia, and BS in Computer and Automatic Control Engineering, Jordan. He has more than 40 published refereed articles in scholarly international journals and proceedings of international conferences.
1 Introduction

The use of digital multimedia data such as movies, television broadcasts, 3D models and similar digital products has grown very rapidly over the last few years. Nowadays it is simpler for multimedia element owners to transfer data over the internet. Obviously, the data could be entirely carbon copy and rapidly redistributed. Thus digital copyright protection and fingerprinting of multimedia elements has been the centre of multimedia security research. Its importance is increasing rapidly in the field of computer graphics and multimedia communication because of the growing problem of the unauthorised duplication.

The problem of 3D mesh watermarking and hashing is fairly new area as weighed against two dimensional watermarking (Cox et al., 2001; Fotopoulou et al., 2000; Sverdlov et al., 2005; Abdallah et al., 2006). It has received less concentration partially because the technology that has been employed for the image and video analysis cannot be simply adapted to three dimensional models that can be characterised in numerous ways (voxels, NURBS, and polygonal meshes). Early algorithms on 3D watermarking (Benedens, 1999; Harte and Bors, 2002; Kwon et al., 2003) consist of hiding the secret data immediately by altering either the 3D mesh geometry or the topology of the triangles. These methods are usually simple and require low computational cost. However, they are not robust enough to different types of attacks. Recently, several watermarking algorithms for 3D mesh in the frequency domain have been proposed (Praun et al., 1999; Ohbuchi et al., 2001; Abdallah et al., 2008). Embedding in the frequency domain is mainly based on multi-resolution mesh analysis (spectral decomposition and wavelet transform). Frequency domain techniques show excellent resistant to the most common attacks.

In Ohbuchi et al. (2001) a watermarking scheme based on the mesh spectral matrix has been proposed. The watermark is embedded by modifying the spectral coefficients. In Abdallah et al. (2008), the authors use the spectral eigen-decomposition and nonnegative matrix factorisation. The idea is to apply the NMF to small blocks of the spectral coefficient matrix of the 3D polygonal meshes. Then modify the singular values of the weight matrix. A statistical watermarking scheme for 3D polygonal mesh models that modify the distribution of vertex norms via changing respectively the mean and the variance of the histogram mapping function is presented in Cho et al. (2007). Through experiments and simulations, they proved that the technique is robust against vertex reordering and simplification attacks. The main problem with this technique is that, the result will drop significantly with small size models. In Karni and Gotsman (2000) the mesh Laplacian matrix was used to encode the 3D shape into a more compact representation. This was done by retaining the smallest eigen-values and associated eigenvectors which contain the highest concentration of the shape information. The spectral compression is used (Abdallah et al., 2009), where the 3D model is partitioned into smaller parts and the spectral compression is applied to the sub-meshes. The watermark is then spread over the low frequency spectral coefficients of the compressed meshes. The altered spectra with some other less important basis functions are used to get uncompressed watermarked 3D mesh.

A hashing scheme for 3D models using spectral graph and entropic spanning trees is presented in Tarmissi and Ben Hamza (2009). The idea is to partition a 3D model, and then apply eigen-decomposition to the Laplace-Beltrami matrix of each sub-model. The hash value is computed using spectral coefficients and Tsallis entropy estimate. The results show robustness against several attacks. However, our observation shows weakness against mesh simplification attack. Other hashing scheme based on object features is proposed in Lee and Kwon. (2012). The hashing groups the distances from feature objects with the highest surface area in a 3D model. It generates a binary hash through the binarisation of feature values that are calculated by the combinations of group values and a random key. The experiments show robustness against various perceptual geometrical and topological attacks.

Motivated by the good performance of the oblivious watermarking techniques for 3D polygonal models that modify the distribution of vertex norms, we propose a robust blind watermarking approach using the 3D mesh Skewness Measure of the spherical coordinates. Extensive numerical experiments are performed to demonstrate the much improved performance of the proposed method.

The remainder of this paper is organised as follows. In Section 2, we briefly review some background material and introduce the proposed approach and describe in detail the watermarking algorithm. In Section 3, we present some experimental results, and we show the robustness of our method against the most common attacks. Finally, we conclude and point out future directions in Section 4.
2 Proposed approach

In this section, we briefly review some background material and describe the main steps of the proposed 3D watermarking methodology.

2.1 3D objects

3D objects are usually represented as triangle meshes. A triangle mesh \( \mathbb{M} \) is a triple \( \mathbb{M} = (V, E, T) \), where \( V = \{v_1, \ldots, v_m\} \) is the set of vertices, \( E = \{e_{ij}\} \) is the set of edges, and \( T = \{t_1, \ldots, t_n\} \) is the set of triangles. Each edge \( e_{ij} = [v_i, v_j] \) connects a pair of vertices \( \{v_i, v_j\} \). The sets \( V \) and \( T \) may be written in matrix form as:

\[
V = \begin{pmatrix}
v_1 \\ v_2 \\ \vdots \\ v_m
\end{pmatrix} = \begin{pmatrix}
v_{1x} & v_{1y} & v_{1z} \\ v_{2x} & v_{2y} & v_{2z} \\ \vdots & \vdots & \vdots \\ v_{mx} & v_{my} & v_{mz}
\end{pmatrix},
\]

\[
T = \begin{pmatrix}
t_1 \\ t_2 \\ \vdots \\ t_t
\end{pmatrix} = \begin{pmatrix}
t_{11} & t_{1y} & t_{1z} \\ t_{21} & t_{2y} & t_{2z} \\ \vdots & \vdots & \vdots \\ t_{t1} & t_{tj} & t_{tk}
\end{pmatrix}.
\]

2.2 Skewness measure

One of the essential tasks in any statistical analysis is to describe the variability of a dataset. A more advance data feature that can be helpful for several application is the skewness measure. Skewness is used to determine how symmetric or not symmetric a dataset. A distribution is symmetric if it appears identical to the left and right of the core point (Hippel and Paul, 2005). The skewness value can indicate a perfectly symmetric dataset and the values most of the values are to the left of the mean. A zero skew illustrates that the data are distributed more to the right of the mean than to the right. Moreover, most of the values are to the right of the mean. A positive skew specify that the data are distributed more to the right than to the left and most of the values are to the left of the mean. A zero value indicates a perfectly symmetric dataset and the values are equally distributed on both sides of the mean (Kendall and Stuart, 2010). Figure 1 depicts a left-skewed distribution, right-skewed distribution and a symmetric distribution. The skewness measure \( \gamma \) of a dataset is \( Y_1, Y_2, \ldots, Y_n \) is:

\[
\gamma = \frac{\sum_{i=1}^{n} (Y_i - \bar{Y})^3}{(N - 1)S^3}
\]

where \( \bar{Y} \) is the mean, \( S \) is the standard deviation, and \( N \) is the number of data points. The skewness for a normal distribution is zero [13], and any symmetric data should have skewness near zero.

2.3 Proposed watermarking algorithm

In this section, we explain the core idea of the proposed watermark embedding and extraction algorithms. One of the most attractive features of the proposed methodology is the blindness of the watermark extraction. It is a more challenging task than non-blind techniques where we do not have the advantage of using the original model in the extraction process. Thus no information about the original model is used. This is significant for testing robustness against attacks where no need for a perfect realignment between the original and the watermarked models. Having this advantage, a wider range of applications can benefit from the blind watermarking techniques. Furthermore, web based detection is offered and anyone can perform detection.

2.3.1 Watermark embedding process

We propose a watermarking method that embeds a watermark into the 3D model by modifying the vertices skewness. The modification is done according to a predefined watermark bits to be embedded. The algorithmic steps of the proposed 3D watermarking approach may be summarised as follows:

- Extract the 3D model vertices and faces. Figure 2 shows the armadillo 3D model (181002 vertices and 162000 faces) with its triangles.
- Extract the spherical coordinates \( \rho_i, \theta_i, \phi_i \) from the 3D model vertices. A spherical coordinate is a point in a three-dimensional space (see Figure 3) where the position of this point is specified by: \( \rho_i \) is the ith radial distance which represents the distance between each vertex and the centre \( (O) \), \( \phi_i \) is the angle between the...
z-axis and the vector from the origin to the point (ranges from 0 to 180 degrees), and \( \theta_i \) is the angle between the \( x \)-axis and the projection of the point onto the \( XY \)-plane (ranges from 0 to 360 degrees). The mean value of all vertices of a 3D model is calculated \((x_g, y_g, z_g)\) then the spherical coordinates is computed using the following equations:

\[
\begin{align*}
\rho_i &= \sqrt{(x_i - x_g)^2 + (y_i - y_g)^2 + (z_i - z_g)^2} \\
\theta_i &= \tan^{-1}\left(\frac{(y_i - y_g)}{(x_i - x_g)}\right) \\
\phi_i &= \cos\left(\frac{z_i - z_g}{\rho_i}\right).
\end{align*}
\]

1. Divide the radial distances \( \rho_i \) into separate intervals with equal range according to their values. As every interval should accommodate one watermark bit, make sure that the number of intervals is greater than or equal to number of the secret message bits (watermark). The radial distances \( \rho_i \) are assigned to their intervals based on the difference between highest and the lowest value of the radial distances as follows:

\[
\rho_{\text{min}} + \frac{\text{Diff}}{N} \times n_i \leq \rho_i \leq \rho_{\text{min}} + \frac{\text{Diff}}{N} \times (n_i + 1)
\]

where \( N \) is number of intervals, \( \text{Diff} = \rho_{\text{max}} - \rho_{\text{min}} \), \( \rho_{\text{max}} \) is the value of the maximum radial distance of the whole model, \( \rho_{\text{min}} \) is the value of the minimum radial distance of the whole model, \( \rho_i \) must fall between the first part (lowest bound value) and second part (highest bound value) of the equation.

2. Adjust the values of radial distances to different scale. This could be done by normalising the radial distances of every interval to have them between \([0, 1]\). Then, compute the skewness measure \( \gamma \) of every single interval using the normalised radial distances in the interval.

3. The embedding is done by hiding the first bit of the watermark into the first interval, and hiding the second bit in the second interval and so on. The embedding is done in two ways summarised in Algorithm 1. The algorithm presents our proposed watermark embedding technique. If the bit to be hidden is 0 then the skewness value of the selected interval is modified by subtracting a constant from all the radial distances \( p_i \) that fall in the interval recursively until the skewness value is less than zero. On the hand, if the skewness value of that interval is less than zero, the interval is kept with no changes. Part 2 of the algorithm is used to handle a watermark bit of value = 1. The skewness value of the selected interval is changed by adding a constant for all the distances \( p_i \) that fall in the interval recursively until the skewness value is greater than zero. If the skewness value of that interval is greater than zero, the interval is kept with no changes.

4. Inverse the normalisation in step 4 to its normal scale. Then, retrieve the cartesian coordinates from the spherical coordinates for all vertices using:

\[
\begin{align*}
 x_i &= p_i \cos\theta_i \sin\phi_i \\
y_i &= p_i \sin\theta_i \sin\phi_i \\
z_i &= p_i \cos\theta_i.
\end{align*}
\]

Figure 2 shows the armadillo 3D model with its vertices (see online version for colours)

Figure 3 Spherical coordinates

Figure 4 illustrates the armadillo 3D model before and after hiding the secret message (see online version for colours)
2.3.2 Watermark extraction process

We proposed a blind watermarking scheme. The original un-watermarked model is not needed for extracting the embedded watermark. Algorithm 2 shows in details the extraction process. If the correlation coefficient between the initial watermark vector and the extracted vector is greater than a predefined threshold, then the watermark is present.

3 Evaluation

Extensive experiments were performed using several 3D models. The conducted experiments are used to investigate the watermark robustness against attacks. Robustness is the most significant attribute that we need to judge when designing a watermarking system for copyright protection or indexing. In our experiments we used a random watermark vector of
32 bits. The watermark is embedded several times in the model as much the investigated 3D mesh intervals are able to accommodate the watermark bits. A 3D model with 70 intervals would provide accommodation to 32 bits watermark twice.

To evaluate the robustness, we applied different attacks to several models embedded by modifying the skewness values of the normalised radial distances. Attacks are not always stand for eliminating the embedded watermark (Voloshynovskiy et al., 2001); it may consist of several processes to make the watermark untraceable.

Figure 5 The armadillo 3D mesh with different attacks (see online version for colours)

Figure 6 The correlation coefficient between the original watermark and the average extracted watermarks for attacks shown in Figure 4. The line with darker colour represents the correlation coefficient between the original watermark and the average

In order to increase the reliability of the presented results, each experiment is repeated 10 times independently and the average values are reported. For each of the attacks, we display the attacked 3D model and the average correlation coefficient between the original watermark and the average
Table 1 The robustness of the proposed method against attacks for six different 3D models (see online version for colours)

<table>
<thead>
<tr>
<th>3D model</th>
<th>Attack</th>
<th>Average correlation</th>
<th>Model details</th>
</tr>
</thead>
<tbody>
<tr>
<td>AG noise $\sigma^2 = 0.01$</td>
<td>0.9379</td>
<td>Bunny</td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.05$</td>
<td>0.881917</td>
<td>81002 vertices</td>
<td></td>
</tr>
<tr>
<td>MU noise $\sigma^2 = 0.01$</td>
<td>1.000</td>
<td>162000 faces</td>
<td></td>
</tr>
<tr>
<td>Simplification 0.7</td>
<td>0.881917</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaling on z-axis</td>
<td>0.93887</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise $\sigma^2 = 0.01 +$ simplification 0.8</td>
<td>0.873015</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.01$</td>
<td>1.000</td>
<td>Max Planck</td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.05$</td>
<td>0.81263</td>
<td>58958 vertices</td>
<td></td>
</tr>
<tr>
<td>MU noise $\sigma^2 = 0.01$</td>
<td>0.881917</td>
<td>117912 faces</td>
<td></td>
</tr>
<tr>
<td>Simplification 0.7</td>
<td>0.812637</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaling on z-axis</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise $\sigma^2 = 0.01 +$ simplification 0.8</td>
<td>0.93887</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.01$</td>
<td>0.82841</td>
<td>Hand</td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.05$</td>
<td>0.65079</td>
<td>87139 vertices</td>
<td></td>
</tr>
<tr>
<td>MU noise $\sigma^2 = 0.01$</td>
<td>0.873015</td>
<td>174278 faces</td>
<td></td>
</tr>
<tr>
<td>Simplification 0.7</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaling on z-axis</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise $\sigma^2 = 0.01 +$ simplification 0.8</td>
<td>0.93792</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.01$</td>
<td>0.9379</td>
<td>Hippo</td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.05$</td>
<td>0.881917</td>
<td>52674 vertices</td>
<td></td>
</tr>
<tr>
<td>MU noise $\sigma^2 = 0.01$</td>
<td>0.93887</td>
<td>105340 faces</td>
<td></td>
</tr>
<tr>
<td>Simplification 0.7</td>
<td>0.87381</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaling on z-axis</td>
<td>0.88191</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise $\sigma^2 = 0.01 +$ simplification 0.8</td>
<td>0.80965</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.01$</td>
<td>0.9388</td>
<td>Horse</td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.05$</td>
<td>0.82841</td>
<td>59549 vertices</td>
<td></td>
</tr>
<tr>
<td>MU noise $\sigma^2 = 0.01$</td>
<td>0.87301</td>
<td>119094 faces</td>
<td></td>
</tr>
<tr>
<td>Simplification 0.7</td>
<td>0.820651</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaling on z-axis</td>
<td>0.881917</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise $\sigma^2 = 0.01 +$ simplification 0.8</td>
<td>0.81263</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.01$</td>
<td>0.88191</td>
<td>Lucy</td>
<td></td>
</tr>
<tr>
<td>AG noise $\sigma^2 = 0.05$</td>
<td>0.65079</td>
<td>50002 vertices</td>
<td></td>
</tr>
<tr>
<td>MU noise $\sigma^2 = 0.01$</td>
<td>0.881917</td>
<td>100000 faces</td>
<td></td>
</tr>
<tr>
<td>Simplification 0.7</td>
<td>0.80965</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaling on z-axis</td>
<td>0.82841</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise $\sigma^2 = 0.01 +$ simplification 0.8</td>
<td>0.74603</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

extracted watermarks. Figure 5 shows the armadillo 3D mesh after scaling, rotation, smoothing, mesh simplification, mesh compression, and several noise attacks including additive gaussian (AG), multiplicative uniform (MU), additive uniform (AU), and multiplicative gaussian (MG). Combinations of the previous attacks are used on the tested models to test the resistance of the watermark against more than one attack.

Figure 6 depicts the correlation coefficient between the original watermark and the average extracted watermarks for the attacks shown in Figure 5. Moreover, Figure 6 display the correlation coefficient for another 49 arbitrarily produced watermarks. The correlation coefficient between the original watermark and the average extracted watermarks is located at the middle position on the x-axis and plotted with darker colour. Experimentally the correlation coefficient of 0.7 is chosen as a threshold to identify whether the watermark is present or it got destroyed by the attack. The threshold is used to decrease false-positive alarm and false-negative alarm. The extracted secret messages from the attacked models shown in Figure 5 are used to evaluate the robustness of the proposed technique. The correlation results between the original embedded messages and the average extracted messages is above 0.7 for all attacks. Hence, the proposed algorithm was able to recover the embedded watermark for most common attacks.

This good performance is in fact consistent with all the models employed in our experimentation. Table 1 show more experiments with different models. We extracted the watermark from several attacked models and cross-correlated
with the original watermarks. The results clearly indicate the high robustness of the proposed algorithm against commonly used attacks in the 3D domain.

4 Conclusion

In this paper, we presented a blind robust and imperceptible watermarking algorithm for 3D models. The watermark is embedded by slightly modifying the skewness measure based on the watermark bits. Embedding the watermark with no significant change to any of the 3D model vertices is not only increasing the robustness against attacks but also increasing the embedding imperceptibility. The main striking feature of this scheme is the blindness of extracting the watermark that opens a wide range of applications. Furthermore, the new proposed algorithm is simple, flexible in data embedding capacity and requires low computational cost. Extensive experimental results show a great resiliency against the most common attacks including the geometric transformations, adaptive random noise, mesh smoothing, and combinations of these attacks. For future work, we plan adopt the skewness measure for 3D mesh hashing to ensure the authentication and the integrity of the 3D models.

References


