Spectral Graph and Minimal Spanning Tree for 3D Polygonal Meshes Fingerprinting

Emad E. Abdallah, Faculty of Prince Al-Hussein bin Abdullah II for Information Technology, Hashemite University, Zarqa, Jordan

Ibrahim Al-Oqily, Faculty of Prince Al-Hussein bin Abdullah II for Information Technology, Hashemite University, Zarqa, Jordan

Alaa E. Abdallah, Faculty of Prince Al-Hussein bin Abdullah II for Information Technology, Hashemite University, Zarqa, Jordan

Ahmed F. Otoom, Faculty of Prince Al-Hussein bin Abdullah II for Information Technology, Hashemite University, Zarqa, Jordan

Ayoub Alsarhan, Faculty of Prince Al-Hussein bin Abdullah II for Information Technology, Hashemite University, Zarqa, Jordan

ABSTRACT

In this paper, the authors present a robust three-dimensional fingerprint algorithm for verification, indexing, and identification. The core idea behind our technique is to apply the eigen-decomposition to the mesh Laplacian matrix, and then compute minimum spanning trees (MST) of several concentrations of the mesh shape structure. The fixed size hash vector of a 3D mesh is defined in terms of the MST values and number of the initial patches. The extensive experimental results on several 3D meshes prove the uniqueness of the extracted hash vectors and the robustness of the proposed technique against the most common attacks including distortion-less attacks, compression, noise, smoothing, scaling, rotation as well as mixtures of these attacks.

Keywords: *3D Attacks, 3D Fingerprinting, 3D Hashing, Eigen-Decomposition, Robustness, Spectral Graph*

1. INTRODUCTION

The usage of digital multimedia data such as movies, television broadcasts, 3D models and similar digital products has grown rapidly over the last few years. Nowadays it is easier for multimedia element to be transmitted over the internet. Obviously, the data could be entirely photocopied or customized and retransmitted. Thus digital copyright protection and fingerprinting to ensure the authentication and the integrity of multimedia elements has long been

DOI: 10.4018/ijitwe.2014100104

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

the core of the digital data security research. Its importance is growing quickly due to the increasing problem of the illegal replication and the prohibited digital content modification.

The problem of 3D object hashing and watermarking is relatively new field as compared to 2D watermarking and hashing (Cox, Miller, & Bloom, 2002; Mihak & Venkatesan, 2001; Venkatesan, Koon, Jakubowski, & Moulin, 2000; Swaminathan, Mao, & Wu, 2006; Fridrich, 2000; Abdallah, Hamza, & Bhattacharya, 2006; Lin, Zsu, Oria, & Ng, 2001). It has received less attention partially because the technology that has been used for the 2D images and videos analysis cannot be easily adapted to 3D objects. Moreover, 3D meshes can be effortlessly altered by several graphics attacks without changing the general shape of the model.

Hash functions could assist in guaranteeing the authentication and the integrity of the multimedia element. The authenticity of the digital data can be verified by recalculating the fingerprint value from the underlying data and judging it against the attached fingerprint value (Fridrich, 2000; Abdallah et al., 2006). In addition, the multimedia hashes are used in content-based recovery from databases (Lin et al., 2001).

There has been significant research on image hashing for more than a decade. In Venkatesan et al. (2000) a robust image hashing is introduced. The algorithm employs arbitrary signal processing approaches for a nonreversible compression of images into random binary strings. Experiments show robustness against image changes due to compression, geometric distortions, and other attacks. Fourier transforms features and controlled randomization was used in (Fridrich, 2000) to generate a robust image hash. In (Mihak & Venkatesan, 2001) an image hashing prototype that employs geometric skins is proposed, it creates unpredictable randomized output. The technique has been modified for robust multimedia classification, identification and watermarking. Another image hashing technique is presented in (Monga & Evans, 2006), where several feature points is extracted for perceptual image hashing. The chosen feature points maintain the geometry and steady to perceptually irrelevant deformations. Statistical analysis is employed for image hashing and watermarking (Cannons & Moulin, 2004), where the hash is chosen from undisclosed division of the discrete cosine transforms of an image and the watermark is embedded using multiplicative technique.

Unlike images, 3D hashing and watermarking is much harder, given that a small change on the 3D mesh geometry would extremely damage the embedded watermark or change the mesh fingerprint (W. Berchtold & Steinebach, 2014). Early algorithms on 3D watermarking (Benedens, 1999) consist of embedding the watermark information directly by modifying either the 3D mesh geometry or the topology of the triangles. These methods are usually simple and require low computational cost. Nevertheless, the main problem of these techniques is the limited robustness against attacks. Lately, numerous watermarking techniques are proposed to embed in the frequency domain (Praun, Hoppe, & Finkelstein, 1999; Ohbuchi, Takahashi, Miyazawa, & Mukaiyama, 2001; Abdallah, Hamza, & Bhattacharya, 2009). The idea is based on spectral decomposition and wavelet transform. These methods show good resistance against attacks. In (Ohbuchi et al., 2001), a watermarking algorithms based on the mesh spectral matrix is proposed, where the watermark is embedded by altering the low frequency component of the spectral coefficients. More robust technique based on spectral domain is proposed in (Abdallah et al., 2009) and (Abdallah, Hamza, & Bhattacharya, 2008), the watermarking algorithm employs the eigen-decomposition and the nonnegative matrix factorization. The idea is to use the nonnegative matrix factorization to several blocks of the spectral matrix for a 3D object. In (Karni & Gotsman, 2000) the mesh Laplacian matrix was used to convert the 3D model into a reduced size representation. This was done by retaining the smallest eigen-values and associated eigen-vectors which contain the highest concentration of the shape information.

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Spectral watermarking is used again in (Liu, Prabhakaran, & Guo, 2012) for parametric models. Dirichlet Manifold Harmonic transform is used to modify the 3D model. The spectral bases are calculated to improve the robustness against distortion attacks. A statistical watermarking scheme for 3-D polygonal mesh models that modify the distribution of vertex norms via changing respectively the mean and the variance of each bin by histogram mapping function is presented in (Cho, Prost, & Jung, 2007). Through the simulations, they proved that the technique is robust against vertex reordering and simplification attacks. The main problem with this type of techniques is limited robustness with very small size models and flat regions.

3D mesh hashing using spectral graph is presented in (Ghaderpanah, Abbas, & Hamza, 2008; Tarmissi & Hamza, 2009). The idea depends on utilizing the spectral coefficient and the entropic spanning trees. It starts by partitioning the 3D triangle mesh into smaller parts, followed by applying the spectral decomposition to the Laplace Beltrami matrix of each part. The hash value is computed in terms of spectral coefficients and Tsallis entropy estimate. The results show robustness against several attacks. However, our observation shows weakness against mesh simplification attack. Other hashing method based on object feature vectors is proposed in (Wu & ming Cheung, 2006; Lee & Kwon, 2012).The distances from features objects groups are hashed with the highest surface area in a 3D model. The hash consists of a binary sequence extracted from several features values that are computed by combining group of values and a random key. The experiments show robustness against various perceptual geometrical and topological attacks (Lee & Kwon, 2012). A perfect multidimensional hash function with no hash collisions is proposed in (Lefebvre & Hoppe, 2006), the hash function generates one position to a small offset table. It protects spatial consistency and thus improve runtime locality. A robust hashing against content preserved attacks is proposed in (Lee, Lee, & Kwon, 2010). The algorithm projects the 3D mesh vertices to the shape coordinates, and then it slices the coordinates into rectangular blocks and calculates the concentration of each block. A hash vector is produced using the blocks shape concentrations.

Motivated by the good performance of the spectral watermarking techniques that alter the low frequency component of the spectral coefficients (Abdallah et al., 2009, 2008; Cho et al., 2007), we propose a robust fingerprinting approach by extracting the global and the local geometry of the 3D object into a hash vector which may be used for several protection purposes including authentication and integrity. To compute the fingerprint vector, the MST of a connected meshes is calculated for several compressed versions of the 3D model. Wide ranges of numerical tests are carried out to show the good performance of the proposed method and its robustness against attacks. The remainder of this paper is organized as follows. In section 2, we provide a brief background material about 3D model representation and the spectral mesh compression. In Section 3, we introduce the proposed approach and we describe in more details the fundamental steps of the fingerprint extraction algorithm. In Section 4, 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 5.

2. BACKGROUND

2.1. 3D Model Representation

In computer graphics, 3D objects are regularly represented as triangle meshes (Abdallah et al., 2009). A triangle mesh $\mathbb{M}$ is a triple $\mathbb{M} = \left(V, E, T\right)$, where $L = B\nabla B^T$ is the set of vertices, E is the set of edges, and T is the set of triangles. Each edge $e_{ij} = \left[v_i v_j\right]$ connects a pair of vertices $\left\{v_i, v_j\right\}$. Two distinct vertices $v_i, v_j \in V$ are adjacent if they are connected by an edge $e_{ij} \in E$. The neighborhood of a vertex v_i is the set vertices that are ad-

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

jacent to v_i. The set of vertices V may be written in matrix form as:

$$V = \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} \begin{pmatrix} v1_x & v1_y & v1_z \\ \vdots & \vdots & \vdots \\ vm_x & vm_y & vm_z \end{pmatrix}$$

2.2. Spectral Mesh Compact Representation

The 3D mesh geometry is represented as a linear combination of a few basis vectors (Karni & Gotsman, 2000).The main idea is to apply the eigen-decomposition to the mesh Laplacian matrix L, and then remove the high frequency eigenvectors in order to reduce the dimensionality of the new model. The Laplacian matrix $L = A - \nabla$, where A is the adjacency matrix between the vertices and ∇ is the $m \times m$ diagonal matrix whose (i,i) entry is an eigen value ∇_i. A major compression is obtained with a very small harm in the mesh quality. Obviously, the general shape is preserved since the low frequency basis functions contain the ideal concentration of the 3D model features. The eigen-decompression of the Laplacian matrix L isgivenby $L = B\nabla B^T$ where $B = \left(b_1, b_2, ..., b_n\right)$ is a perpendicular matrix whose columns b_i are the eigenvectors of L which we refer to as basis vectors, the diagonal matrix of eigen-values of L is sorted in increasing order (Abdallah, Hamza, & Bhattacharya, 2009). The generated mesh vertices form the eigenvectors and the diagonal matrix is given by:

$$V^T = C^T B^T = \sum_{i=1}^{n} c_i^T b_i^T$$

where C is the spectral matrix. To obtain a compressed version of the mesh vertices the $\sum_{i=1}^{t} c_i^T b_i^T$ should end with t where t is smaller than n (Abdallah, Hamza, & Bhattacharya, 2009; Karni & Gotsman, 2000).

3. PROPOSED APPROACH

In this section, we describe the main steps of the proposed fingerprinting algorithm. The main objective of our proposed approach may be described as extracting the 3D mesh fingerprint from both global and local geometry of the 3D shapes in a multi-resolution manner. In this case we generate a unique fingerprint and increase the robustness against several types of attacks. Figure 1 depicts the flow diagram of the fingerprint extraction process.

3.1. Fingerprinting Extraction Algorithm

Let $\mathbb{M}$ be a 3D mesh of n vertices and $n \times n$ triangles, the 3D mesh fingerprint vector is extracted as follows:

- Compute the Laplacian matrix L of size $n \times n$;
- Compute the eigen-values and vectors for the Laplacian matrix L. The eigen-decompression of the matrix L is given by $L = B\nabla B^T$;
- Project the mesh vertices onto the eigenvectors to get the spectral coefficients matrix of the original 3D mesh $\mathbb{M}$; that is:

$$V^T = C^T B^T = \sum_{i=1}^{n} c_i^T b_i^T$$

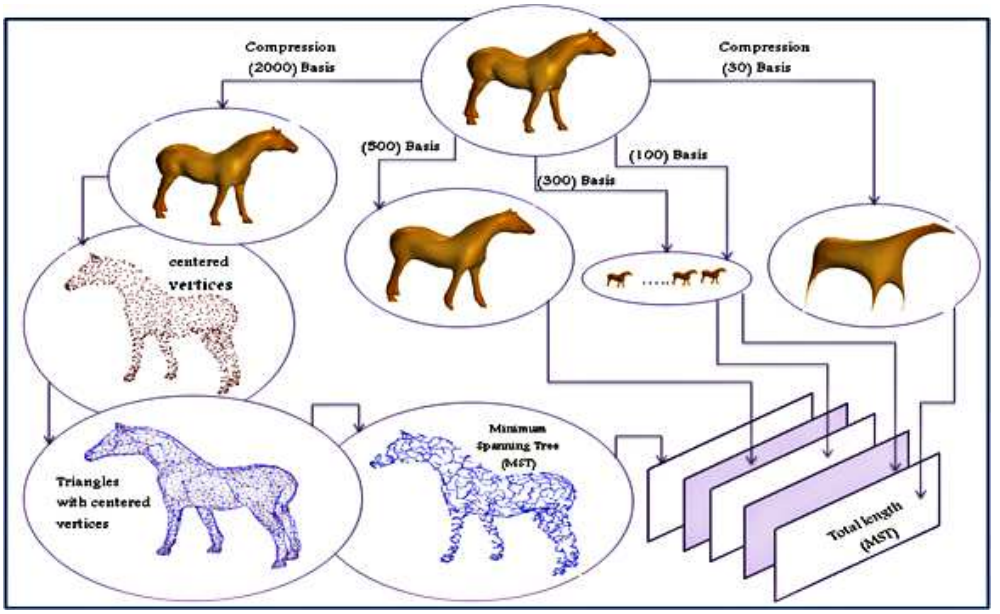
- Obtain several compressed versions of the mesh vertices using t_i, where t is smaller than n.

For each compressed mesh version $\mathbb{M}r$ of size $\left(m \times m\right)$, compute the MST as follows:

- Compute the centric point = average of all vertices:

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 1. Fingerprint extraction process



$$Cp = \left(cp_x, cp_y, cp_y\right) = \begin{pmatrix} x_1 & y_1 & zx_1 \\ \vdots & \vdots & \vdots \\ x_m & y_m & z_m \end{pmatrix} / m$$

where $\left(x_1, y_1, z_1\right)$ is the first vertex and m is the number of vertices in a compressed mesh $\mathbb{M}r$.

- Define the centered vertices:

$$Cv = \begin{pmatrix} x_1 & y_1 & zx_1 \\ \vdots & \vdots & \vdots \\ x_m & y_m & z_m \end{pmatrix} - Cp$$

- For each triangle t_i, compute the center of mass $\left(Ct_i\right)$ using the centered vertices Cv (see Figure 2):

$$Ct_i = \frac{\left(Cv_{tix}, Cv_{tiy}, Cv_{tiz}\right)}{3}$$

- Use the center of mass CT matrix to compute the total length of the MST:

$$CT = \begin{pmatrix} Cv_{t1x} & Cv_{t1y} & Cv_{t1xz} \\ \vdots & \vdots & \vdots \\ Cv_{tmx} & Cv_{tmy} & Cv_{tmz} \end{pmatrix}$$

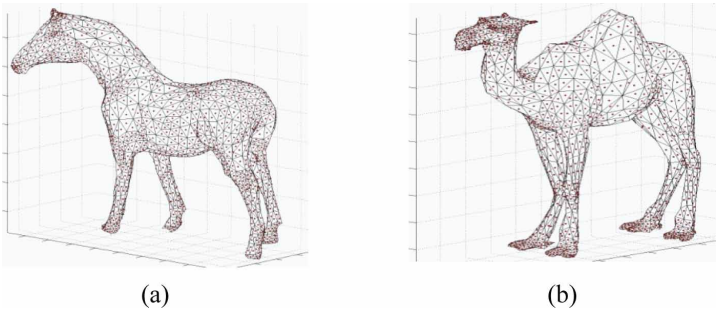
We assume that CT is a complete connected graph, and then the MST is computed using the well-known greedy algorithm in graph theory (Mukwembi, 2013).

- Compute the total length of the minimum spanning tree L_{Mr};
- The total length L_{Mr} of every compressed mesh (resolution) is used as an element of the hash vector.

Initially, the 3D mesh faces are first reduced or increased to 10000 faces. This initial step is important in order to have unified compression resolutions for all 3D models. Experimentally,

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

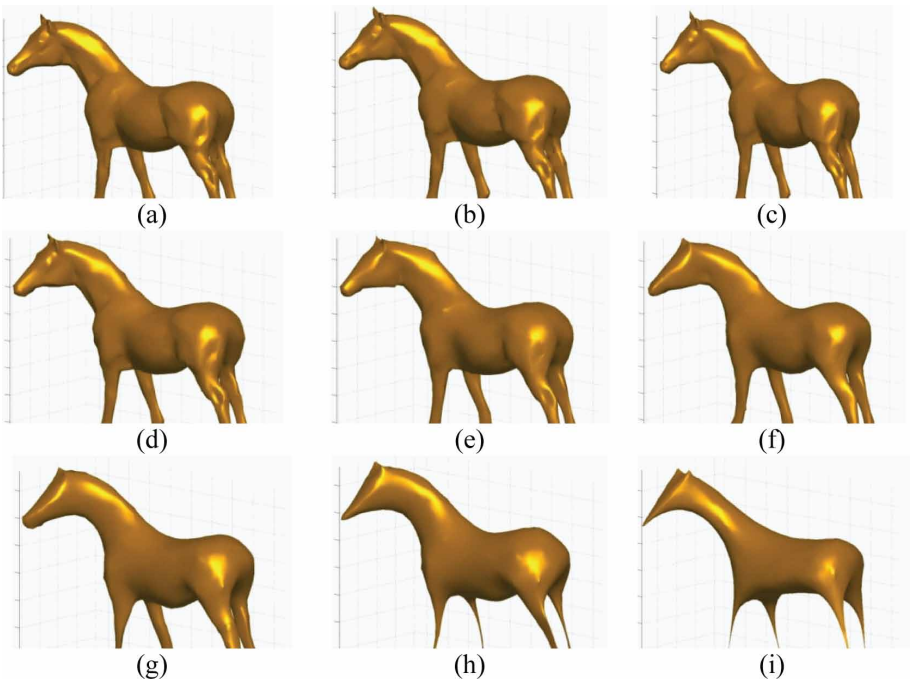
Figure 2. 3D models triangles with their centric vertices: (a) Horse model with 2000 batches, (b) Camel model with 2000 batch



(a) (b)

we used 8 different resolutions for each 3D model (Karni & Gotsman, 2000). In the first resolution we used 1/2 of the original basis functions, and then 1/4 of the basis vectors are used. In the last resolution, only 1/256 of the original basis functions are employed to extract one element to be added to the hash vector. Figure 3 illustrate the horse 3D mesh in 8 different resolutions and Figure 4 depict the MST for the horse models shown in Figure 3, as well as the corresponding total length of the MST (hash values).

Figure 3. 3D horse model under 8 different resolutions: (a) Horse with 10000 batches, (b) Spectral compression with 2000 basis functions (BF), (c) 1000 BF, (d) 500 BF, (e) 125 BF, (g) 65 BF, (h) 30 BF, (i) 10 BF



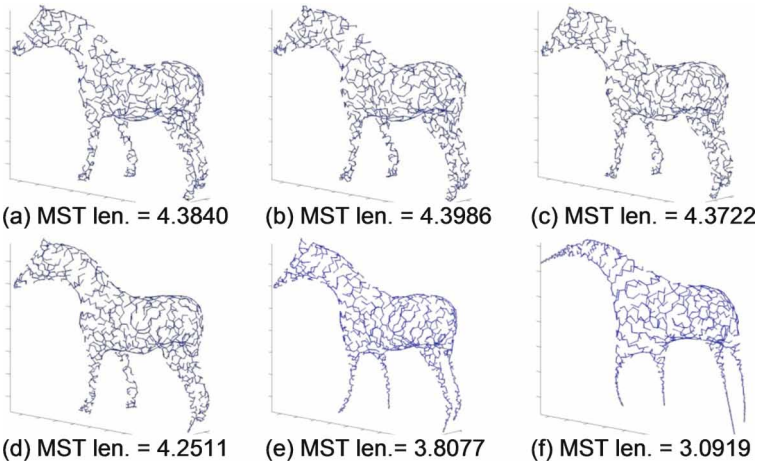
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 4.The MST of the horse 3D model under 6 different resolutions: (a) Spectral compression with 1000 BF and MST length, (b) 500 BF, (c) 125 BF, (d) 65 BF, (e) 30 BF, (f) 10 BF



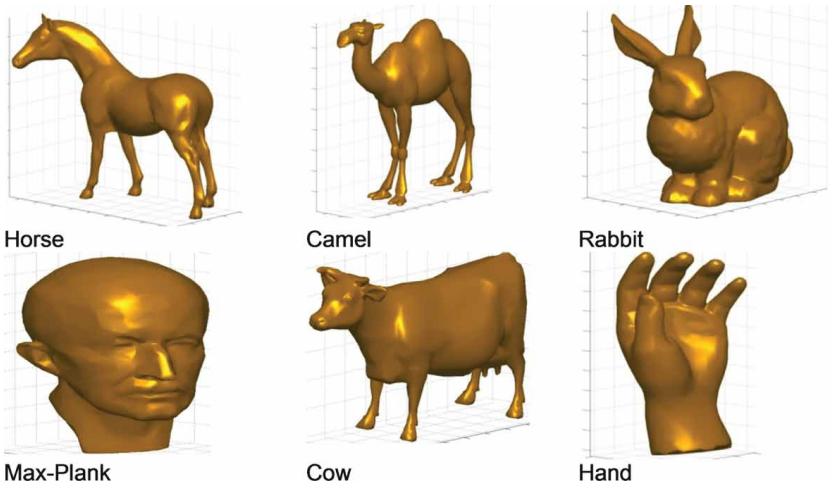
4. EXPERIMENTAL RESULTS

Our experiments were performed using a variety of 3D models represented as triangle meshes as shown in Figure 5. We conducted experiments to test the uniqueness of the hash vectors and the robustness against numerous 3D attacks.

4.1. Robustness against 3D Attacks

Robustness is an important factor to test the effectiveness of a fingerprinting algorithm. It is essential issue that we need to accomplish when proposing a scheme for verification, indexing, and identification. Attacks are usually

Figure 5. 3D models used for experimentation



Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

operations to make the fingerprint undetectable (Voloshynovskiy, Pereira, Pun, Eggers, & Su, 2001). We verified the robustness of the proposed algorithm with different 3D models (see Figure 5) against various attacks including mesh transformation, mesh simplification, mesh smoothing, additive random noise, and mesh compression. In the experiments we show the attacked models, MST, and the correlation coefficient between the original fingerprint and the generated one from the attacked model (Matching Performance). A threshold is experimentally chosen to reduce presenting falsely fingerprint in the attacked model and falling to detect the right fingerprint from the model. If the correlation coefficient is larger than 0.8, then the fingerprint is exist, hence, we can draw several conclusions for verification, indexing, and identification applications.

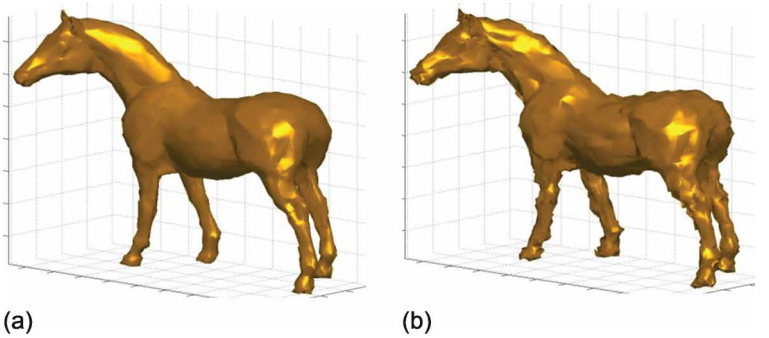
Testing the robustness of the proposed fingerprinting scheme against random noise is evaluated by adding a random vector to each vertex in the 3D model. See Figure 6 (a, b) for noisy horse model with Gaussian random noise $\left(\sigma^2 = 0.25\%\right)$ and $\left(\sigma^2 = 0.5\%\right)$ respectively. The correlation coefficients between the original fingerprint and the attacked one are reported for experiments.

Mesh filtering is very common attack that is used to destroy or to make the fingerprint undetectable. The Laplacian filter (Vollmer, Mencl, & Mller, 1999) is employed to test our algorithm against smoothing attack. Laplacian filter alter the location of each mesh vertex to the centric point of its neighboring vertices. Clearly the low spectral components are most affected by the Lablacian filter. The new scheme is robust against smoothing attack due to the way we extract the fingerprint from different resolutions of the 3D mesh. Figure 7 (a, b) depicts the horse model attacked by 7 and 10 smoothing iterations respectively. As can be seen, the mesh is significantly smoothed but the fingerprint is still perfectly detectable.

Testing the robustness against geometric transformations is shown in Figure 8 (a, d, c, and d). Transformation is the easiest attacks that might be employed to make the fingerprint undetectable. Geometric attack is defined by a set of operations that performed over the targeted 3D mesh. A rotation attack for example is applied by spinning the 3D model around x, y or z- axis by a predefined angle. MST length of the 3D mesh will not change by applying rotation and translation attacks making the proposed scheme robust against these attacks. Scaling attack is very risky for fingerprinting schemes that use the MST where the distances between the centric vertices changes significantly. However our algorithm shows good robustness for most of the models (see Table 1) against scaling attack.

3D models compression (Karni & Gotsman, 2000; Voloshynovskiy, Pereira, Pun, Eggers, & Su, 2001) has lately turn into one

Figure 6. Robustness against Gaussian random noise: (a) Noisy horse model with noise standard deviation $\left(\sigma^2 = 0.25\%\right)$*, (b) Noisy horse model with noise standard deviation* $\left(\sigma^2 = 0.5\%\right)$



Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 7. Robustness against Laplacian smoothing: (a) Zoom-into the horse model with Laplacian smooth attack $\left(5 iterations\right)$*, (b) Zoom-into the horse model with Laplacian smooth attack* $\left(10 iterations\right)$

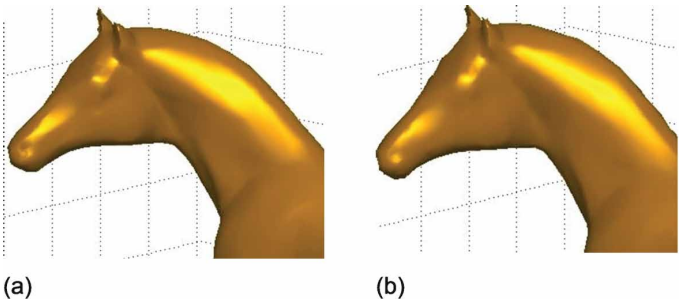
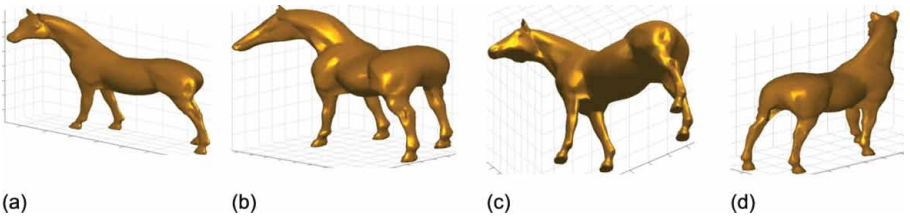


Figure 8. Robustness against geometric transformations: (a) Horse model with scaling attack ($Y-axis$ *), y values of all vertices multiplied by 2, (b) Horse model with scaling attack (* $X-axis$ *), x values of all vertices multiplied by 2, (c) Horse model with rotation attack around (* $Z-axis$ *), (d) Horse model with rotation attack around (* $Y-axis$ *)*



of the most efficient attacks, because the new compression algorithms achieve a very significant compression ratio with very little loss in the mesh features. We assessed the robustness of our scheme against several compression attacks (Karni & Gotsman, 2000; Jovanova, Preda, & Preteux, 2009). The proposed scheme shows an outstanding robustness against compression. It can be explained as our fingerprint is extracted using several compressed version of the 3D polygonal mesh, making compression attack ineffective and harmless to our fingerprints. Figure 3 depicts the compressed horse model constructed with several basis functions from the original mesh.

Mesh Simplification (Gonzalez, Gumbau, Chover, Ramos, & Quiros, 2009) may also be used to transform a 3D model to a new mesh with smaller number of faces, edges and vertices. The simplification procedure is regularly restricted by a set of quality criteria that can protect specific properties of the original model as much as possible (Gotsman, Gumhold, & Kobbelt, 1998). Classical properties include the geometric distance and the visual appearance. Simplification algorithms work iteratively by removing vertex and edge at a time. Obviously reducing number vertices could destroy or make the fingerprint undetectable. See Figure 9 for simplified horse model. The mesh is simplified down to 70% of its original vertices and faces. Our proposed method is robust against the simplification attack because of the pre-process of reducing the number of faces before extracting the fingerprint.

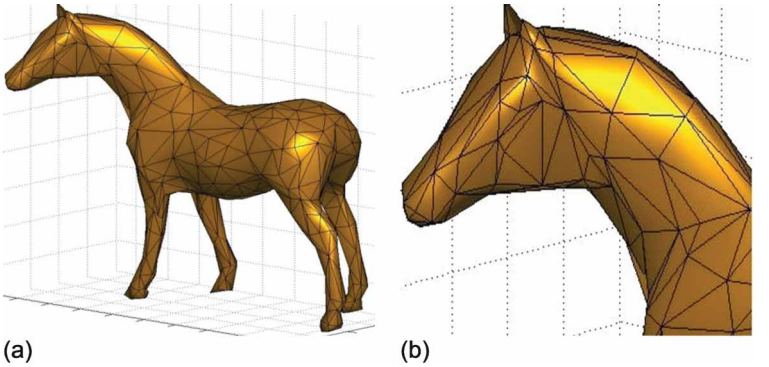
The robustness of our fingerprinting algorithm is evaluated using a combination of the previous attacks. Figure 10 illustrate the 3D

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Table 1. Normalized correlation coefficient between the original fingerprint and the generated one from the attacked model

Attacks	3D Models					
	Horse	**Camel**	**Rabbit**	**M.Plank**	**Cow**	**Hand**
Gaussian random noise $\left(\sigma^2 = 0.25\%\right)$	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Gaussian random noise $\left(\sigma^2 = 0.50\%\right)$	1.0000	1.0000	0.9999	1.0000	0.9998	0.9999
Laplacian smoothing $\left(5 iterations\right)$	0.9976	0.9998	0.9710	1.0000	0.9996	1.0000
Laplacian smoothing $\left(10 iterations\right)$	0.9894	0.9990	0.9710	0.9998	0.9997	0.9997
Transformations-scaling $\left(X - axis\right)$	0.8989	0.9999	0.7517	1.0000	1.0000	1.0000
Transformations-rotation $\left(Z - axis\right)$	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Simplification to 70%	0.9868	0.9995	0.9333	0.9998	0.9992	0.9996
Compression attack 1000 face	0.9865	0.9898	0.9658	0.9998	0.9997	0.9998
Noise $\left(\sigma^2 = 0.25\%\right)$ and simp. (75%)	0.9907	0.9987	0.9271	0.9999	0.9998	0.9997
Smoothing (8 iter.) and Scaling $\left(Y - axis\right)$	0.9152	0.9997	0.7371	0.9996	1.0000	0.9996
Noise $\left(\sigma^2 = 0.25\%\right)$, smoothing (8 iter.), and rotation $\left(Z - axis\right)$	0.9864	0.9902	0.9658	0.9976	0.9894	0.9865

Figure 9. Robustness against simplification attack, (a) Horse model simplified to 70% of its original vertices and faces, (b) Zoom-into the simplified horse head



Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

horse model with multiple attacks. Figure 10 (a) shows the 3D horse attacked by adding additive random noise of ($\sigma^2 = 0.25\%$) and then simplified down to (75%) of its original vertices and faces. Figure 10 (b) depicts the model with Laplacian filter (8 iterations), then transformation attack (rotation and scaling on ($Y - axis$)) is applied to the smoothed model. In both situations the proposed scheme detects the fingerprint faultlessly. The correlation coefficients for all attacks are shown in Table 1. Clearly, the good performance is in fact consistent with a variety of 3D models used for experimentation (see Table 1).

4.2. Uniqueness of the Fingerprints Vectors

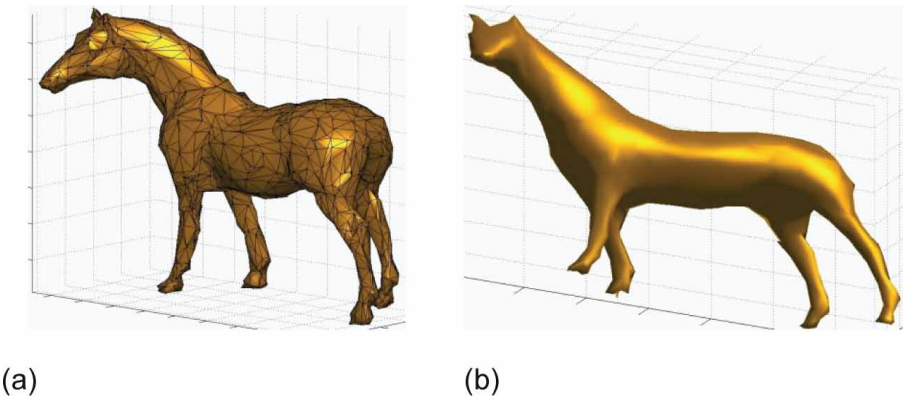
One essential issue we need to consider when designing a fingerprinting algorithm is uniqueness. The fingerprint of each model should be exclusive. Two different models with the same exact fingerprint is prohibited for several applications. Our new proposed algorithm generates a fixed size fingerprint for each 3D mesh regardless of number of vertices or the size of the model. We tested the uniqueness of the fingerprint vector by computing the correlation coefficients between the 3D model under investigation and other fingerprints for different models.

Figure 11 (a) illustrate the correlation coefficient between the horse fingerprint and 5 other fingerprints for 5 different models. Clearly, the correlation coefficient between the fingerprints for the two horse models is perfect (See the right bar on Figure 11 (a)). However, the correlation between the horse model and other 3D models is never exceeding (0.6). Therefore, the proposed fingerprinting algorithm shows an excellent capability to achieve uniqueness by producing completely different fingerprints for the tested 3D models. Figure 11 (b, c, d, e, and f) illustrate the correlation coefficient between the camel, rabbit, Max blanck, cow, and hand 3D models respectively.

5. CONCLUSION

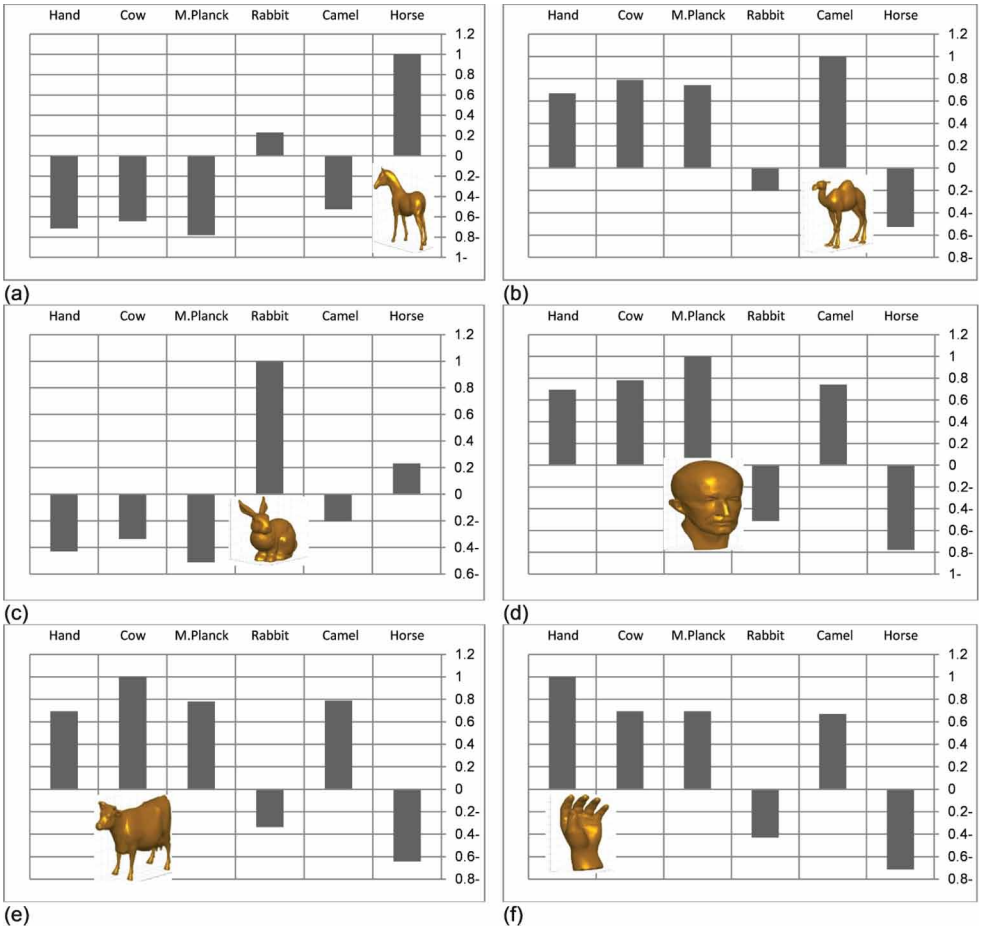
In this paper, we proposed a robust identification and authentication fingerprinting algorithm. It can capture the global and the local geometry of several compressed versions of the 3D model.

Figure 10. Robustness against multiple attacks: (a) Horse model attacked by additive random noise of $\left(\sigma^2 = 0.25\%\right)$ *and simplification (75%), (b) Horse model attacked by Laplacian filter (8 iterations) and two level of transformations*



Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 11. Uniqueness of the generated fingerprint vectors: (a) Correlation coefficient between the horse fingerprint and 5 other fingerprints, (b) Camel, (c) Rabbit, (d) MaxPlanck, (e) Cow, (f) Hand



The minimum spanning tree of the spectral mesh compression is reported as a one element of the fingerprinting vector. The main attractive property of the proposed algorithm is uniqueness and robustness against attacks. The performance of the proposed method was evaluated through extensive experiments that clearly showed a perfect resiliency against transformations, simplification, additive random noise, and mesh compression attacks. For future work, we plan to examine the relationship between the number of spectral vectors used in the compression process, size of the fingerprint, and MST length to further improve the robustness against attacks.

REFERENCES

Abdallah, E. E., Hamza, A. B., & Bhattacharya, P. (2006). A robust block-based image watermarking scheme using fast hadamard transform and singular value decomposition. In *Icpr (3)* (pp. 673–676). IEEE Computer Society. doi:10.1109/ICPR.2006.167

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Abdallah, E. E., Hamza, A. B., & Bhattacharya, P. (2008). Robust 3d watermarking technique using eigendecomposition and nonnegative matrix factorization. In A. C. Campilho & M. S. Kamel (Eds.), *Iciar* (Vol. 5112, pp. 253–262). Springer. doi:10.1007/978-3-540-69812-8_25

Abdallah, E. E., Hamza, A. B., & Bhattacharya, P. (2009). Watermarking 3d models using spectral mesh compression. Signal. *Image and Video Processing*, *3*(4), 375–389. doi:10.1007/s11760-008-0079-y

Benedens, O. (1999). Geometry-based watermarking of 3d models. *IEEE Computer Graphics and Applications*, *19*(1), 46–55. doi:10.1109/38.736468

Berchtold, W., Schfer, M. R., & Steinebach, M. (2014). Robust hashing for 3d models. In *Proc. spie 9028, media watermarking, security, and forensics*. SPIE.

Cannons, J., & Moulin, P. (2004). Design and statistical analysis of a hash-aided image watermarking system. *IEEE Transactions on Image Processing*, *13*(10), 1393–1408. doi:10.1109/TIP.2004.834660 PMID:15462148

Cho, J.-W., Prost, R., & Jung, H.-Y. (2007). An oblivious watermarking for 3-d polygonal meshes using distribution of vertex norms. *IEEE Transactions on Signal Processing*, *55*(1), 142–155. doi:10.1109/TSP.2006.882111

Cox, I., Miller, M. L., & Bloom, J. A. (2002). *Digital watermarking*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Fridrich, J. (2000). Robust hash functions for digital watermarking. In *Itcc* (pp. 178–183). IEEE Computer Society.

Ghaderpanah, M., Abbas, A., & Hamza, A. B. (2008). Entropic hashing of 3d objects using laplace-beltrami operator. In *Icip* (pp. 3104–3107). IEEE. doi:10.1109/ICIP.2008.4712452

Gonzalez, C., Gumbau, J., Chover, M., Ramos, F., & Quiros, R. (2009). User-assisted simplification method for triangle meshes preserving boundaries. *Computer Aided Design*, *41*(12), 1095–1106. doi:10.1016/j.cad.2009.09.007

Gotsman, C., Gumhold, S., & Kobbelt, L. (1998). *Simplification and compression of 3d meshes. In the european summer school on principles of multiresolution in geometric modelling* (pp. 319–361). Springer.

Jovanova, B., Preda, M., & Preteux, F. (2009). Mpeg-4 part 25: A graphics compression framework for xml-based scene graph formats. *Image Commun.*, *24*(1-2), 101–114.

Karni, Z., & Gotsman, C. (2000).Spectral compression of mesh geometry. In Eurocg (p. 27-30).

Lee, S. H., & Kwon, K.-R. (2012). Robust 3d mesh model hashing based on feature object. *Digital Signal Processing*, *22*(5), 744–759. doi:10.1016/j.dsp.2012.04.015

Lee, S. H., Lee, E.-J., & Kwon, K.-R. (2010). Robust 3d mesh hashing based on shape features. In Icme (p. 1040-1043).IEEE.

Lefebvre, S., & Hoppe, H. (2006). Perfect spatial hashing. *ACM Transactions on Graphics*, *25*(3), 579–588. doi:10.1145/1141911.1141926

Lin, S., Zsu, M. T., Oria, V., & Ng, R. T. (2001). An extendible hash for multi-precision similarity querying of image databases. In P. M. G. Apers, P. Atzeni, S. Ceri, S. Paraboschi, K. Ramamohanarao, & R. T. Snodgrass (Eds.), *Vldb* (pp. 221–230). Morgan Kaufmann.

Liu, Y., Prabhakaran, B., & Guo, X. (2012). Spectral watermarking for parameterized surfaces. *IEEE Transactions on Information Forensics and Security*, *7*(5), 1459–1471. doi:10.1109/TIFS.2012.2204251

Mihak, M. K., & Venkatesan, R. (2001). New iterative geometric methods for robust perceptual image hashing. In T. Sander (Ed.), *Digital rights management workshop* (Vol. 2320, pp. 13–21). Springer.

Monga, V., & Evans, B. L. (2006). Perceptual image hashing via feature points: Performance evaluation and tradeo_s. *IEEE Transactions on Image Processing*, *15*(11), 3452–3465. doi:10.1109/TIP.2006.881948 PMID:17076404

Mukwembi, S. (2013). On spanning cycles, paths and trees. *Discrete Applied Mathematics*, *161*(13-14), 2217–2222. doi:10.1016/j.dam.2013.03.018

Ohbuchi, R., Takahashi, S., Miyazawa, T., & Mukaiyama, A. (2001).Watermarking 3d polygonal meshes in the mesh spectral domain. In Graphics interface (p. 9-18).

Praun, E., Hoppe, H., & Finkelstein, A. (1999). Robust mesh watermarking. In Siggraph (p. 49-56).

Swaminathan, A., Mao, Y., & Wu, M. (2006). Robust and secure image hashing. *IEEE Transactions on Information Forensics and Security*, *1*(2), 215–230. doi:10.1109/TIFS.2006.873601

Tarmissi, K., & Hamza, A. B. (2009). Information-theoretic hashing of 3d objects using spectral graph theory. *Expert Systems with Applications*, *36*(5), 9409–9414. doi:10.1016/j.eswa.2008.12.062

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Venkatesan, R., Koon, S.-M., Jakubowski, M. H., & Moulin, P. (2000).Robust image hashing. In Icip (p. 664-666).

Vollmer, J., Mencl, R., & Mller, H. (1999). Improved laplacian smoothing of noisy surface meshes. In Computer graphics forum (pp. 131-138). doi:10.1111/1467-8659.00334

Voloshynovskiy, S., Pereira, S., Pun, T., Eggers, J. J., & Su, J. K. (2001). Attacks on digital watermarks: Classification, estimation-based attacks, and benchmarks. *IEEE Communications Magazine*, *39*(8), 118–126. doi:10.1109/35.940053

Wu, H.-T., & Ming Cheung, Y. (2006). Public authentication of 3d mesh models. In *Web intelligence* (pp. 940-948). IEEE Computer Society.

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.