PERCEPTUAL HASHING FOR 3D MESH MODEL AUTHENTICATION

[1]EUNG-JOO LEE, [2]SUK-HWAN LEE, [3]TAEKOOKKIM, [4]KI-RYONG KWON

1Dept. of Communication Engineering, Tongmyong University, South Korea
2Dept. of Information Security, Tongmyong University, South Korea
3Dept. of IT Convergence and Application Engineering, Pukyong National University, South Korea
Email: ejlee@tu.ac.kr, sukhwanlee@gmail.com, skylee@tu.ac.kr, krkwon@pknu.ac.kr

Abstract- 3D content-based hashing has not been as widely used as compared to 2D content-based hashing in the case of multimedia content such as images and videos. In this study, we develop a robust 3D mesh–model hashing based on a heat kernel signature (HKS) that can describe a multi-scale shape curve and is robust against isometric modifications; we also discuss the robustness, uniqueness, security, and model space of the hash for 3D model hashing.

Index Terms- Perceptual 3D Model Hashing, 3D Hash Function, Heat Kernel Signature, Authentication

I. INTRODUCTION

With the rapid growth of the 3D content market and services, content-based hashing for the authentication and copy detection of 3D content has become a necessity. 3D content can be represented by various graphic models of polygon mesh, NURBS, or CAD. The conventional image and video hashing methods cannot be applied to vector data based on 3D graphic models.

3D content hashing has received less attention than have 3D watermarking [1] and 3D retrieval [2]. Among a few papers that have been presented to date, K. Tarmissia [3] presented the information-theoretic hashing of a 3D mesh using spectral graph theory and entropic spanning trees. Lee [4],[5] presented a robust 3D mesh hashing based on a key-dependent 3D surface feature, namely, the block shape feature combining the curvedness and the shape index. These schemes exhibit robustness against some attacks, including the uniqueness of the model and key and the security. But it cannot be robust against mesh simplification and tessellation of topologic attacks. Furthermore, they used the vertex distance [6] for 3D hierarchical object structure and used heat kernel signature [7] for shape feature.

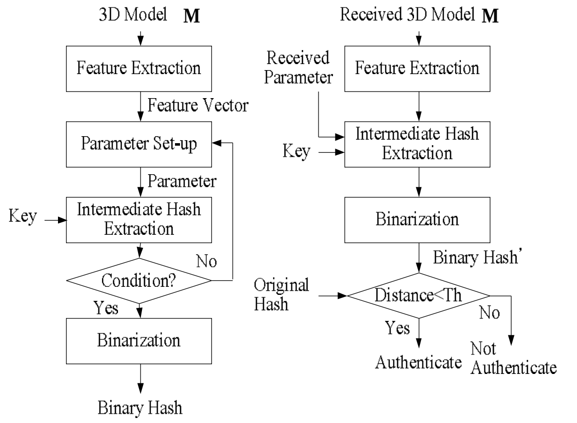
In this paper, we discuss the properties related to the robustness, uniqueness, security, and model space of 3D model hashing, and we then propose an HKS-based 3D model hashing dependent on key and parameter. The proposed hashing obtains a pair of HKS coefficients in scales of local time and global time; the coefficients are calculated by eigenvalues and eigenvectors of a mesh Laplace operator that is estimated discretely from the Laplace–Beltrami operator and that clusters HKS coefficients into 2D square cells with variable size. The binary hash is generated from the intermediate hash vector that is obtained by projecting feature values to random values. Feature values are defined by the weighted distance based on the n-order Butterworth function of a pair of HKS coefficients.We evaluated the robustness against various geometric attacks and topologic attacks using a 3D public editing tool, and we evaluated the uniqueness of models and keys and the model space by measuring the attack intensity in the available authentication range. These properties were evaluated by the normalized Hamming distance. Lastly, we evaluated the security by modeling the differential entropy of the intermediate hash according to the Swaminathan method [8]. Experimental results verified that, for all requirements, the proposed hashing has superior performance compared to conventional hashing.

II. PROPOSED PERCEPTUAL 3D MESH MODEL HASHING

This paper presents 3D model–based hashing using HKS distribution. The proposed hash generation, as shown in figure 1(a), consists of the shape feature extraction, the parameter setting, the intermediate hash generation, and the binarization for generating the final binary hash. Among these steps, the parameter setting and the intermediate hash generation are performed iteratively until the intermediate hash value is the target value that satisfies the conditions for robustness, uniqueness, and model space. The model authentication by hash, as shown in figure 1(b), performs the authentication based on the Hamming distance between an original hash and a hash generated in the transmitted 3D model using the transmitted parameter and the stored key.

A. Cell HKS Extraction

The heat kernel signature at a vertex $\boldsymbol{v}_i \in \mathbf{V}$ is $HKS(\boldsymbol{v}_i, t) = k_t(\boldsymbol{v}_i, \boldsymbol{v}_i) = \sum_{j=1}^{n} e^{-\lambda_j t}\phi_j^2(\boldsymbol{v}_i)$, which is obtained from eigenvalues λ_j and eigenvectors ϕ_j of the Laplace–Beltrami operator. We rescale all vertices so that the surface area of the 3D model is 100, and we calculate eigenvalues and eigenvectors from a discrete Laplace operator $L_K^h f(\boldsymbol{v})$ on a mesh surface that is presented by Belkin [9] according to the same method of J. Sun [10]. The discrete Laplace operator $L_K^h f(\boldsymbol{v})$ is defined as



(a) (b)

Fig. 1. The process of (a) 3D model hash generation and (b) 3D model authentication based hash.

$$L_K^h f(\boldsymbol{v}_i) = \frac{1}{4\pi h_i^2} \sum_{M_v} \sum_{j=1}^{N_{M_v}} w_{ij}\left(f(\boldsymbol{v}_i) - f(\boldsymbol{v}_{ij})\right) \quad (1)$$

where $w_{ij} = \frac{S(M_v)}{N_{M_v}} exp\left(-\frac{\|v_i - v_j\|}{4h_i}\right)$.

$M_v$is the valence mesh of $\boldsymbol{v}$ and N_{M_v} is the number of vertices in M_v. The sphere surface area of the local valence meshes of $\boldsymbol{v}$ is $4\pi h^2$. We calculate eigenvalues and eigenvectors in the case of n=300 and divide the time duration of the log-scale, $[ln(t_{min}), ln(t_{max})]$ ($t_{min} = e^{ln10}, t_{max} = e^{100ln10}$), 150 times into uniform intervals $\Delta t = (ln(t_{max}) - ln(t_{min}))/150$, then we calculate $HKS(\boldsymbol{v}_i, t)$ at each time $t_i = t_{min} + (i-1)e^{\Delta t}$ ($i \in [1{,}150]$). The proposed hashing divides the time scale into a local scale ($ln(t_{min}) \leq ln(t) \leq ln(t_{min} + 75)e^{\Delta t}$) and a global scale ($ln(t_{min} + 75)e^{\Delta t} < ln(t) \leq (ln(t_{max})$) and uses average HKS coefficients in local and global scales as an HKS vector $hks(\boldsymbol{v}_i) = (x_i, y_i)$ at a vertex $\boldsymbol{v}_i$.

For feature extraction, we firstly calculate the square cell vector using local and global HKS coefficients. Thus, we define 2D X,Y axes as local and global HKS coefficients, respectively, and we select a square cell by extending the 1D bin center points of each axis to two dimensions. The size of each cell can be determined by the 1D bin center points of the X,Y axes that are allocated to this cell, and the length of the final hash is determined by the size of each cell. The center points of each cell are obtained from the minimum 1D bin size and simple k-means clustering. The minimum 1D bin size based on an estimated L^2 risk function is calculated from the mean $\bar{m}$ and biased variance σ^2 of a histogram with bin-width h. We calculate the minimum 1D bin size Δx for all local HKS coefficients X = $\{x_i | i \in N_V\}$ and the minimum 1D bin size Δy for all global HKS coefficients Y = $\{y_i | i \in N_V\}$ then we determine the number N of 1D bins that are the same in the X, Y axes.

$$N = a\min\left(\left\lfloor\frac{max(x_i) - min(x_i)}{\Delta x}\right\rfloor, \left\lfloor\frac{max(y_i) - min(y_i)}{\Delta y}\right\rfloor\right) + b \quad (4)$$

Therefore, the number of square cells of the X,Y axes is $N \times N$, which is the bit length of the final hash. Since the number of square cells is different for 3D models, the hash uniqueness will be improved. However, this variable cell number effects the distribution of the cell vector, the bit length of the hash, and the hash robustness. We limit the available range of the 1D bin number to within [10,19] by setting two variables a, b to 1/10 and, 10, which makes the number of square cells and the hash bit length [100,361].

The two bin center points $\boldsymbol{u}_X, \boldsymbol{u}_Y$ of X,Y can be extended to a 2D square cell $\mathbf{B} = \{B_{ij} = \Delta x_i \times \Delta y_i | i, j \in [1, N]\}$. The set of X–Y pairs included in any cell can be defined as $\mathbf{G} = \{G_{ij} | i, j \in [1, N]\}$, where $G_{ij} = \{hks(\boldsymbol{v}_k) = (x_k, y_k) \in B_{ij} | k \in [1, N_{ij}]\}$.It is known that the probability that any HKS coefficient is included in a cell G_{ij} is N_{ij}/N_V. The first hash parameter- that is, the bin center points $\mathbf{u} = (\boldsymbol{u}_X, \boldsymbol{u}_Y)$ of X,Y- is generated differently in each 3D model and also in the same model because the parameter is sensitive to the initial points. Therefore, this parameter achieves the improvement of the uniqueness and security.

B. Hash Generation

The proposed hashing projects feature values of cells onto random key patterns and it calculates the intermediate hash so that the projected values will reach the target robustness, and it generates the final binary hash through the binarization of the intermediate hash.

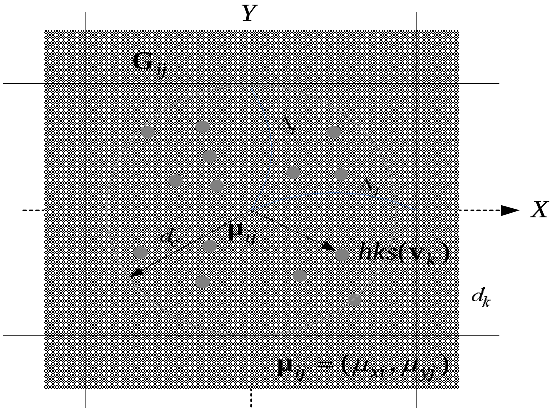


Fig. 2. Center points $\boldsymbol{u}_{ij} = (\boldsymbol{u}_{Xi}, \boldsymbol{u}_{Yj})$ of a cell $\boldsymbol{B}_{ij}$ and the weighting function based on the distance.

Firstly, we calculate the cell value b_{ij} from the magnitude of the HKS coefficients $hks(\boldsymbol{v}_k) = (x_k, y_k)$with a weight $w_k$that is obtained from the n-order Butterworth function, according to the distance $d_k = \|hks(\boldsymbol{v}_k) - \boldsymbol{u}_{ij}\|$ between the centre point $\boldsymbol{u}_{ij} = (u_{Xi}, u_{Yj})$ of a cell B_{ij} and the HKS coefficients $hks(\boldsymbol{v}_k)$.

$$b_{ij} = \sum_{k=1}^{N_V} \frac{a_{ij}}{1 + (d_k/dc_{ij})^n} \frac{|hks(\boldsymbol{v}_k)|}{|\boldsymbol{u}_{ij}|} \quad (5)$$

where $|hks(\boldsymbol{v}_k)| = \sqrt{{x_k}^2 + {y_k}^2}$, $|\boldsymbol{u}_{ij}| = \sqrt{{u_{Xi}}^2 + {u_{Yj}}^2}$

dc_{ij} is the cut-off frequency distance in a cell, which is the minimum distance between a cell and the left and right cells.

$$dc_{ij} = \min\left(\left\|\frac{\boldsymbol{u}_{i+1j}-\boldsymbol{u}_{ij}}{2}\right\|, \left\|\frac{\boldsymbol{u}_{ij+1}-\boldsymbol{u}_{ij}}{2}\right\|\right) \tag{6}$$

a_{ij} is the amplitude of the n-order Butterworth weighting for bringing the intermediate hash to the robustness target, and we use it as the hash parameter. From the n-order Butterworth weighting, as shown in figure 2, the HKS coefficients have different weights, whether they are in the cut-off frequency region or not. The feature value of each cell f_{ij} is

$$f_{ij} = \sum_{i=1}^{N} b_{ij} + \sum_{j=1}^{N} b_{ij}. \tag{7}$$

As mentioned above, the amplitude of a cell must be adjusted for the intermediate hash to reach the robustness target. We define the amplitude to be $a_{ij} = a_{ij}^{(0)}\alpha_{ij}$, that is, a multiplication of $a_{ij}^{(0)}$ of normal distribution $N(1.0.1.0)$ and an adjustable variable α_{ij}. After $\alpha_{ij} = 1$, the feature value f_{ij} can be rewritten as

$$f_{ij} = a_{ij}^{(0)}\left(\sum_{i=1}^{N}\left(\sum_{k=1}^{N_V}\frac{1}{1+\left(\frac{d_k}{dc_{ij}}\right)^n}\frac{|hks(\boldsymbol{v}_k)|}{|\boldsymbol{u}_{ij}|}\right)+\sum_{j=1}^{N}\left(\sum_{k=1}^{N_V}\frac{1}{1+\left(\frac{d_k}{dc_{ij}}\right)^n}\frac{|hks(\boldsymbol{v}_k)|}{|\boldsymbol{u}_{ij}|}\right)\right) = a_{ij}^{(0)}\hat{f}_{ij} \tag{8}$$

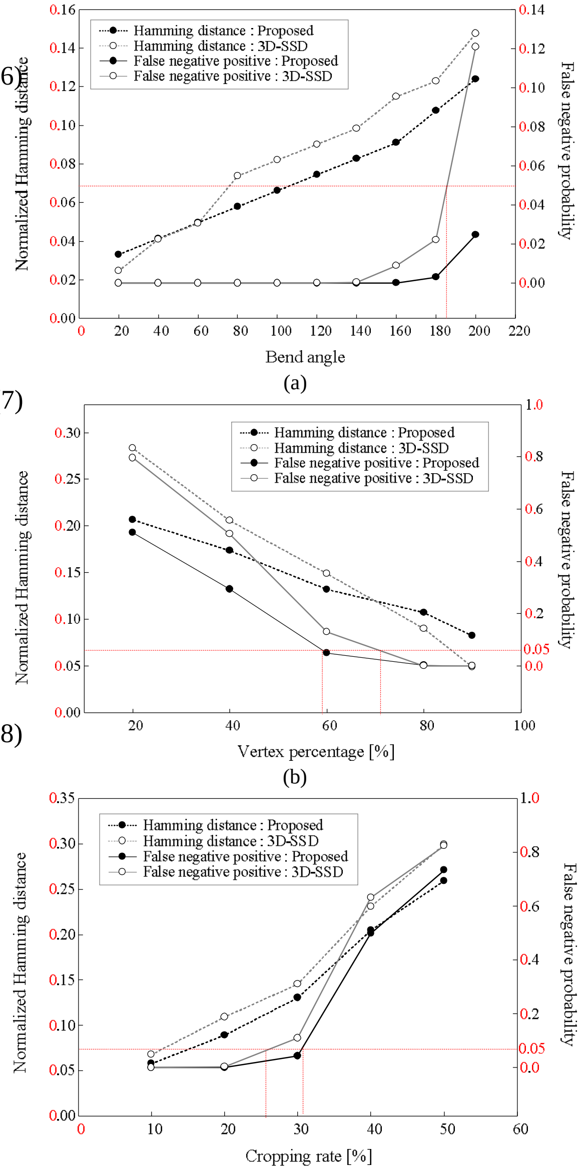
The intermediate hash $\mathbf{H}_I = \{hi_{ij} | i, j \in [1, N]\}$ is calculated by projecting the feature vector $\mathbf{F} = \{f_{ij} | i, j \in [1, N]\}$ onto the random vector $\mathbf{R} = \{r_{ij} | i, j \in [1, N]\}$. The intermediate hash $\mathbf{H}_I$ will be the same as the target values $\mathbf{T}$ for the cell amplitudes $\mathbf{A}$**,** which are the second hash parameter. The proposed hashing permutes $\mathbf{H}_I$ using a 2D permutation key $\mathbf{P}$ to improve the security and model space and it generates the final binary hash $\mathbf{h}_{M,\Theta;K}$ by thresholding.

$$h_{ij} = \begin{cases} 1, \text{if} h_{I,ij} > Th \\ 0, \text{otherwise} \end{cases}, \forall i, j \in [1, N] \tag{9}$$

Therefore, given a model $\mathbf{P}$ and a key $\mathbf{K} = (\mathbf{R}, \mathbf{P})$, the final hash $\mathbf{h}_{M,\Theta;K}$ is generated by the parameter $\Theta(\mathbf{M}, \mathbf{K}) = (\mathbf{u}, \mathbf{A})$ that satisfies the hash requirements for robustness, uniqueness, security, and model space.

III. EXPERIMENTAL RESULTS

We converted 1,000 models provided by "*Princeton Shape Benchmark*" to VRML data for test models and we rescaled all of the test models to be in the same bounding box for the same experiment condition. We made a comparison of the performance of the robustness, uniqueness, model space, and security between the proposed hashing and 3D-SSD based hashing [4],[5].



(a)

(b)

(c)

Fig. 3. Normalized Hamming distances and authentication error probabilities of the proposed hashing and 3D-SSD hashing to (a) Bending, (b) Mesh downsampling, (c) Cropping.

C. Robustness Evaluation

For the robustness evaluation, we generated 100 hashes in each model and extracted hashes $\mathbf{h}'$ in models that were attacked by geometric and topologic modifications in 3ds-max. We analyzed the authentication error probability P_e of the attacked models $\mathbf{M}'$ using the normalized Hamming distance $D(\mathbf{h}_{M,\Theta;K}, \mathbf{h}_{M',\Theta;K})$.

$$P_e = 1 - \Pr\left[D\left(\mathbf{h}_{M,\Theta;K}, \mathbf{h}_{M',\Theta;K}\right) < 0.2\right] \tag{10}$$

Figure 3 provides the normalized Hamming distance $D(\mathbf{h}_{M,\Theta;K}, \mathbf{h}_{M',\Theta;K})$ and the authentication error probability P_e in all of the attack tests. Hereafter, we refer to the normalized Hamming distance as D shortly. The above experimental results verified that the proposed hashing has more robustness than does 3D-SSD hashing.

D. Uniqueness Evaluation

To evaluate the model-different key uniqueness, we calculated normalised Hamming distances between hashes generated by different keys in 1,000 models and analysed the unique probability $\Pr[D(\mathbf{h}_{M_k,\Theta_k;K_k}, \mathbf{h}_{M_l,\Theta_l;K_l}) > 0.3]$, the undecided probability $\Pr[0.2 < D(\mathbf{h}_{M_k,\Theta_k;K_k}, \mathbf{h}_{M_l,\Theta_l;K_l}) \leq 0.3]$, and the non-unique probability $\Pr[D(\mathbf{h}_{M_k,\Theta_k;K_k}, \mathbf{h}_{M_l,\Theta_l;K_l}) \leq 0.2]$. Table 1 lists three probabilities of the model-different key uniqueness. From this table, we know that the unique probability of the proposed hashing is very high (0.998) and is 4.06% higher than that of 3D-SSD hashing; also, the sum of the undecided probability and the non-unique probability is very low (0.00125).

E. Security Evaluation

We modeled the differential entropy $H(h)$ of the intermediate hash values based on random values, then we evaluated them for the proposed hashing and 3D-SSD hashing. The intermediate hash value hi_{ij} of the proposed hashing is $hi_{ij} = \sum_{k=1}^{N} a_{ik} w_{ik} r_{kj}$, which is the sum of the feature value $a_{ik} w_{ik}$ and the random value r_{kj}. The random value r_{kj} exhibits a Gaussian distribution of $N(m_r, \sigma_r^2)$. The differential entropy $H(h)$ of $p_{hi}(x)$ is

$$H(h) = \frac{1}{2} log(2\pi e \times (m_A^2 \sigma_r^2 + m_r^2 \sigma_A^2) \sum_{k=1}^{N} w_{ik}^2). \quad (11)$$

The differential entropy values of each bin number of the proposed hashing and of 3D-SSD hashing are shown in Figure 4. We calculated the differential entropy $H(h)$ of each cell or block of 1,000 models and presented the maximum, minimum and average $H(h)$ in Figure 4. This figure shows that $H(h)$ of the proposed hashing is 9.47 to 15.22 in 10 bin numbers and it increases by the bin number but $H(h)$ of 3D-SSD hashing is 6.92 to 9.47. On the average, the differential entropy of the proposed hashing is 5.34 to 14.42 higher than that of 3D-SSD hashing.

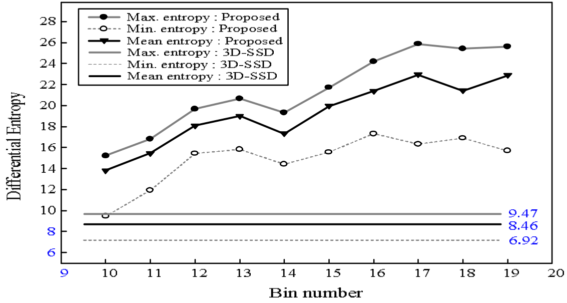


Fig. 4. Differential entropy of the proposed hashing and 3D-SSD hashing.

CONCLUSIONS

This paper presented HKS-based 3D model hashing dependent on a key and parameter for 3D model authentication. Similar to 2D image hashing, 3D model hashing must satisfy the requirements for robustness, uniqueness, and security, though some of the requirements exhibit a trade-off relationship. For satisfying the above four requirements, we design a key-parameter–dependent hash function instead of a key-dependent hash function. From the experimental results, we confirmed that the proposed hashing has higher robustness, wider model space, higher unique probability, and higher differential entropy than does 3D-SSD hashing.

ACKNOWLEDGEMENT

This work was supported by 2013 Brain Busan (BB21) Project and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-NRF-2014R1A1A4A01006663, and NRF-NRF-2016R1D1A3B03931003).

REFERENCES

[1] K. Wang, G. Lavoue, F. Denis, A. Baskurt, “A Comprehensive Survey on Three-Dimensional Mesh Watermarking,”IEEE Transactions on Multimedia, Vol. 10, Issue 8, pp. 1513, Dec. 2008.

[2] B. Bustos, D. A. Keim, D. Saupe and T. Schreck, "Content-Based 3D Object Retrieval," IEEE Computer Graphics and Applications, Vol. 27, Issue 4, pp. 22-27, July-Aug. 2007.

[3] K. Tarmissia and A. B. Hamza, “Information-theoretic hashing of 3D objects using spectral graph theory,”Expert Systems with Applications, Vol. 36, Issue 5, pp. 9409-9414, July 2009.

[4] S.-H. Lee, E.-J. Lee, and K.-R. Kwon, “Robust 3D mesh hashing based on shape features,” IEEE International Conference on Multimedia and Exp, pp. 1040-1043, July 2010.

[5] S.-H. Lee, K.-R. Kwon, and W.-J. Hwang, “Perceptual 3D Model Hashing Using Key-dependent Shape Feature,” Multimedia Tools and Applications, Vol. 73, No. 3, pp. 1723-1755, 2014.

[6] S.-H. Lee, K.-R. Kwon, “Robust 3D mesh model hashing based on feature object,” Digital Signal Processing, Vol. 22, No. 5, pp. 744-759 2012.

[7] S.-H. Lee, K.-R. Kwon, W.-J. Hwang, and V. Chandrasekar, “Key-dependent 3D model hashing for authentication using heat kernel signature,” Digital Signal Processing, Vol. 23, No. 5, pp. 1505-1522 2013.

[8] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," IEEE Trans. on Information Forensics and Security, Vol. 1, Issue 2, pp. 215-230, June 2006.

[9] M. Belkin, J. Sun, and J. Wang, “Discrete Laplace operator on meshed surfaces,” In Proceedings of SOCG, pp. 278-287, 2008.

[10] J. Sun, M. Ovsjanikov and L. Guibas, "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion," Eurographics Symposium on Geometry Processing, Vol. 28, No. 5, 2009.