GEODESIC-BASED ROBUST BLIND WATERMARKING METHOD FOR THREE-DIMENSIONAL MESH ANIMATION BY USING MESH SEGMENTATION AND VERTEX TRAJECTORY

Jen-Sheng Tsai, Jen-Ting Hsiao, Win-Bin Huang, and Yau-Hwang Kuo

Center for Research of E-life Digital Technology
Department of Computer Science and Information Engineering
National Cheng Kung University, Taiwan

ABSTRACT

Motion capture (mocap) data normally used to produce a three-dimensional (3D) mesh animation is exploited for watermarking. Copyright protection of the 3D mesh animation obtained is not guaranteed since identical joints in the mocap data are not distinguished easily. A novel robust blind watermarking method is proposed for the copyright protection of 3D mesh animations in the paper. At the beginning, the shape diameter function (SDF) method segments one frame of a 3D animation into several meaningful parts. A watermark is then embedded into these parts by adjusting the geodesic distances. Vertex trajectories are further exploited to modify the geodesic distribution of residual frames to be the same as that of the watermarked frame. Our experimental result demonstrates that the proposed method is able to resist the noise addition, similarity transformation, quantization, cropping, simplification, smoothing, and resampling attacks.

Index Terms— Watermarking, Three-dimensional Mesh Animation, Geodesic Distance, Segmentation.

1. INTRODUCTION

With the ever-growing expansion of 3D mesh animations in computer graphics and video games industries, copyright protection becomes a major issue due to its commercial values and time-consuming efforts. Digital watermarking is an efficient solution for verifying the owner information. Many watermarking methods have been proposed for images, videos, and 3D mesh models [1]-[4], but there are only a few methods proposed for 3D mesh animations [5]-[10]. A pioneering watermarking method for motion capture (mocap) data, which is composed of a set of joints to produce a 3D mesh animation, was proposed by Kim *et al.* [5] based on multi-resolution analysis. In [6] and [7], discrete cosine transform (DCT) was used to conduct watermarks in the frequency domain of mocap data. Lately, Agarwal *et al.* [8] presented a blind method segmenting and clustering mocap data for watermarking. These methods show robustness against various attacks, such as noise addition, smoothing, cropping, and simplification.

However, once the watermarked mocap data is applied to generate a 3D mesh animation, some hiding information could be lost since identifying the same joints of mocap data obtained from the generated animation with the original ones is difficult [8]. The generated animation could be illegally used or distributed in various applications. Two watermarking methods, VRML-based keyframe animation [9] exploiting the interpolator nodes and skinning animation [10] modifying transformation matrices and skin weights, also have the issue. A non-blind watermark process using registration and resampling methods was proposed as the solution of this issue [6], but such a process would be inefficient in various applications because of the requirement of original mocap.

This paper therefore proposes a robust blind watermarking method for a 3D mesh animation, which normally consists of a discrete set of polygonal meshes. At first, one mesh frame of an animation is segmented into different parts by applying the shape diameter function (SDF) method [11]. Each part is divided into independent sets according to the distribution of geodesic distances from its vertices to the cutting boundary between parts. A copyright watermark is then embedded into these sets by adjusting the mean value of geodesic distances. Finally, vertex trajectories are exploited to modify the geodesic distribution of the remainder frames for constructing a watermarked animation. SDF method and vertex trajectories are useful to identify consistent meaningful parts over all mesh frames of a 3D animation. Besides, geodesic distance, which is invariant to translation, rotation, and insensitive to mesh simplification, guarantees the robustness of hidden watermarks. The benchmark of 3D mesh watermarks [13] and the resampling attack in the temporal domain are used to verify the performance of the proposed method. The experimental results show its ability to resist the noise addition, similarity transformation, quantization, cropping, simplification, smoothing, and resampling attacks.

This paper is organized as follows. Section 2 briefs SDF method. The proposed watermarking method is described in Section 3, and the experimental results are illustrated in Section 4. Finally, the concluding remarks and future work are given in Section 5.

2. MESH SEGMENTATION

Mesh segmentation, which partitions a mesh model into multiple disjoint segments, has become a key process in computer graphics and animation techniques. SDF segmentation method proposed by Shapira *et al.* [11] is adopted in this paper since its meaningful segments are invariant to rigid-body transformations and in-sensitive to skeleton-based movements and articulated character deformations. According to the reference [12], SDF method is a good choice by considering the tradeoff between execution time and performance.

SDF method is briefed in the following three steps:

1) *Diameter Measurement*: The diameter of the volume in the neighborhood of a point is measured in this step. The centroid of each face of a target mesh model is given, and a cone is then applied to its inward-normal direction. Several rays inside the cone are sent to the other side of a mesh. The diameter is set as the weighted average of all rays lengths within one standard deviation from the median of all length.

2) *GMM-EM Clustering*: Multiple Gaussians to the histogram of all diameter values of a mesh is fit to Gaussian mixture model (GMM) by the expectation-maximization (EM) algorithm. Each face is assigned to one cluster according to the probability produced by GMM-EM.

3) *Cutting Boundary Refinement*: Boundary smoothness and concaveness between meaningful parts are refined by alpha expansion graph-cut algorithm which minimizes an energy function related to probability.

3. PROPOSED WATERMARKING METHOD

3.1. Watermark embedding procedure

We use a matrix A to represent an original 3D animation defined by

$$A = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N_F} \\ v_{21} & v_{22} & \cdots & v_{2N_F} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N_V 1} & v_{N_V 2} & \cdots & v_{N_V N_F} \end{pmatrix} = \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_{N_V} \end{pmatrix} \qquad (1)$$

where each v denotes the vertex coordinate and N_V is the number of vertices for each frame. The vertex trajectory is denoted as T and N_F is the number of frames in the animation. Assume face connectivity is constant because most animations are based on bones models, space warps, or simulation. Then, a watermark $W = \{w_i | 1 \le i \le N_w, i \in \mathbb{N}\}$ is a bi-polar sequence $\{+1, -1\}$ generated with a secret key by a pseudo-random generator. It is embedded into the 3D animation by the watermark embedding procedure shown in Fig. 1(a). The detailed procedure is explained as follows.

1) The SDF method introduced in Section 2 is used to segment the first mesh frame of the animation to several meaningful parts.

2) In each part of the frame, the average geodesic distance of each vertex v_{j1} to the vertices residing on the cutting boundary, denoted as d_{jB}, is calculated. According to these geodesic distances, the vertices inside the part are divided into N_w sets defined as

$$C_i = \left\{ v_{j1} \middle| (i-1) \cdot \left(\frac{d_{max}}{N_w}\right) < d_{jB} \le i \cdot \left(\frac{d_{max}}{N_w}\right) \right\} \qquad (2)$$

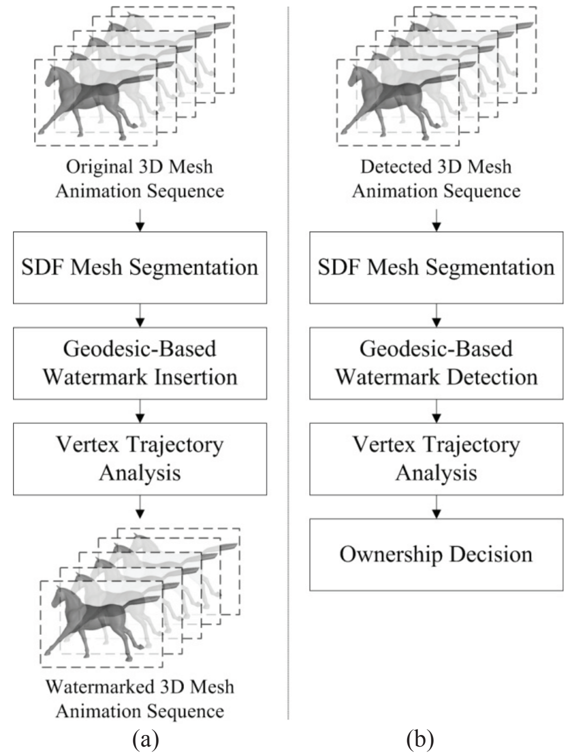


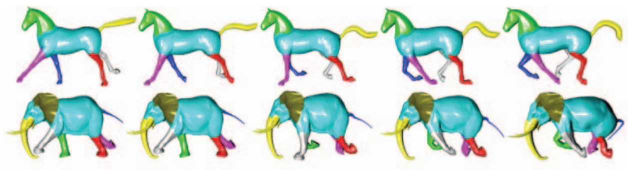
Fig. 1. Block diagrams of (a) watermark embedding procedure and (b) watermark detection procedure.

where d_{max} is the biggest value of all d_{jB} inside the part and $1 \le i \le N_w$. The set C_i contains the vertices whose average geodesic distances belong to the ith range. Note that the one with minimum capacity is selected if there are multiple cutting boundaries in the part.

3) Each element in the watermark sequence W is inserted into a set from $i = 1$ to $i = N_w$. For the ith set, we calculate the average value μ_d of d_{jB}, and check if μ_d is bigger or smaller than a threshold K_i defined by

$$K_i = \begin{cases} (i + 0.5 + \Delta) \cdot \left(\frac{d_{max}}{N_w}\right), & \text{if } w_i = +1 \\ (i + 0.5 - \Delta) \cdot \left(\frac{d_{max}}{N_w}\right), & \text{if } w_i = -1 \end{cases} \qquad (3)$$

where $\Delta \in [0, 0.5)$ is used to control the embedding strength and set empirically. If $\mu_d > K_i$ and $w_i = +1$, output the watermarked set C_i^w. Otherwise, increase μ_d by refining the position of a vertex in the set and go back to check its value. Here, the refinement is performed by randomly selecting a vertex in the set, and then adjusting its position along the edge connecting the vertex to its one-ring neighbor vertex with the longest average geodesic distance to the cutting boundary. The adjustment quantity is an empirical value and d_{max} should be intact. Similarly, if $\mu_d < K_i$ and $w_i = -1$, output the watermarked set C_i^w. Otherwise, decrease μ_d by refining the position of a vertex in the set and then go back to check its value. Also, we select randomly vertex in the set, and adjust its position along the edge connecting the vertex to its one-ring



(a) (b) (c) (d) (e)

Fig. 2. Two frames in the horse-gallop and elephant-gallop animations with different colors in their meaningful parts (a), and their following frames segmented according to vertex trajectories (b)-(e).

Table 1. Two distortion measures, Hausdorff distance and MRMS, for three animations watermarked, Horse-Gallop, Elephant-Gallop, and Camel-Gallop.

Animation	#Vertex	#Face	#Frame	Hausdorff Distance	MRMS
Horse-Gallop	8431	16843	48	0.007246	0.000451
Elephant-Gallop	42321	84638	48	0.006062	0.000335
Camel-Gallop	21887	43814	48	0.002688	0.000153

neighbor vertex with the shortest average geodesic distance to the cutting boundary.

4) According to the vertex trajectories, the meaningful parts of remainder frames for watermarking are identified from boundary information of the results segmented in step 1. If a vertex in the mesh frame resides in the cutting boundary, its corresponding vertices in other frames following the trajectory can be found and denoted as cutting boundary. Two results of the meaningful parts of five frames in horse-gallop and the elephant-gallop animation are shown in Fig. 2. The difference of the geodesic distances between the coordinate of a vertex and its new location adjusted for watermark insertion in step 3 is then calculated as δ_j. Along the trajectories, the positions of the vertices in other frames are also modified based on δ_j and w_i. If $w_i = +1$, the position of a vertex is adjusted with δ_j quantity along the edge connecting it to the vertex which belongs to its one-ring neighbor vertices and has the longest average geodesic distance to the cutting boundary. Oppositely, if $w_i = -1$, it is modified with δ_j quantity along the edge connecting it to the vertex which belongs to one-ring neighbor vertices and has the shortest average geodesic distance to the cutting boundary.

3.2. Watermark detection procedure

The block diagram of the watermarking detection procedure is shown in Fig. 1(b). It is blind since the original animation is not needed, and its detailed procedure is described as follows:

1) The same as the first step in the embedding procedure.
2) The same as the second step in the embedding procedure.
3) The watermark $W' = \{w'_i | 1 \le i \le N_w, i \in \mathbb{N}\}$ is blindly detected from the sets of parts in the mesh frame. For the ith set, we calculate the average value μ'_d and detect the watermark element by

$$w'_i = \begin{cases} +1, & \text{if } \mu'_d > K'_i \\ -1, & \text{if } \mu'_d < K'_i \end{cases} \tag{4}$$

where $K'_i = (i + 0.5) \cdot \left(\frac{d'_{max}}{N_w}\right)$, and d'_{max} is the biggest value among the average geodesic distances of all vertices in the part to the cutting boundary.

4) Similar to the fourth step in the embedding procedure, the meaningful parts are identified by the vertex trajectories and the segmentation in the step 1. The hidden watermark in each part is then detected according to the process in the step 3.

5) In order to verify the ownership information, the bit error rate (BER) between the detected sequence W' and original sequence W from each extracted part is calculated. If the BER is smaller than a defined threshold, it is denoted that a watermark exists in a part.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

Three 3D mesh animations, Horse-Gallop, Elephant-Gallop, and Camel-Gallop listed in Table 1 are used in our experiments. A watermark generated with $N_w = 16$ is embedded into these animations imperceptibly. The embedding strengths Δ and the partition level in SDF method are set as 0.05 and 2, respectively. The distortion metrics, Hausdorff distance and maximum root mean square error (MRMS), are applied for measuring the difference between the original frame and its watermarked version. Table 1 shows the average Hausdorff distance and the average MRMS of all frames in the mesh animations after watermarking.

In order to evaluate the robustness of our proposed method, the benchmark for 3D mesh watermarking [13], which consists of the geometric attacks and connectivity attacks, is adopted. Geometric attacks would modify the vertex coordinates and keep mesh connectivity unchanged in the meantime, including noise addition, quantization, similar transformations, and smoothing. Connectivity attacks change the coordinates of a vertex or the adjacency relations between vertices, such as simplification and cropping. Moreover, two resampling attacks, first down-sampling the mesh animations with 1/2 and 1/4 rate and then up-sampling them by a linear interpolation, are applied. The watermark with the smallest bit errors is detected among the meaningful parts of the frame attacked. The average BER (BER_{avg}) and the maximum BER (BER_{max}) of the watermarks detected from all frames in the attacked animation are calculated. It is noted that SDF method should be applied to the deformed frames in the animation while the information of the vertex trajectories is lost under connectivity attacks. Table 2 shows the detection results of the three test animations after different types of attacks. Obviously, the low BER_{avg} values in the table indicate these watermarks detected from the animations are highly correlated. In the worst cases, the BER_{max} values in the mesh animations, Horse-Gallop, Elephant-Gallop, and Camel-Gallop, are 0.1875, 0.1250, and 0.0625, respectively. As a result, the proposed method guarantees robustness against resampling attacks and some attacks in the benchmark.

Table 2. The BER results of the watermarked animations, Horse-Gallop, Elephant-Gallop, and Camel-Gallop, come through various attacks. BER_{avg}, and BER_{max} are the averaged BER and the maximum BER among all frames in the animation, respectively.

Attacks	**Horse-Gallop**		**Elephant-Gallop**		**Camel-Gallop**	
	BER_{avg}	BER_{max}	BER_{avg}	BER_{max}	BER_{avg}	BER_{max}
Noise Addition 0.05%	0.0443	0.1250	0.0026	0.0625	0.0013	0.0625
Noise Addition 0.10%	0.0443	0.1250	0.0026	0.0625	0.0013	0.0625
Quantization 2^{10}	0.0469	0.1875	0.0026	0.0625	0.0052	0.0625
Quantization 2^{11}	0.0469	0.1875	0.0026	0.0625	0.0052	0.0625
Similarity Transformation	0.0391	0.1875	0	0	0	0
Simplification 10%	0.0391	0.1875	0.0091	0.0625	0	0
Simplification 30%	0.0677	0.1875	0.0339	0.1250	0.0104	0.0625
Smoothing 5-itr	0.0469	0.1250	0.0104	0.0625	0	0
Smoothing 10-itr	0.0469	0.1250	0.0273	0.0625	0	0
Cropping 10%	0	0	0	0	0	0
Cropping 30%	0	0	0	0	0	0
Resampling 1/4	0.0534	0.1875	0	0	0	0
Resampling 1/2	0.0508	0.1875	0	0	0	0

5. CONCLUSIONS

In this paper, a geodesic-based blind watermarking method for 3D mesh animations is proposed by using SDF segmentation and referring vertex trajectories. SDF segmentation and vertex trajectories are helpful to identify the consistent meaningful parts in different mesh frames in watermark embedding and detection procedures. The distribution of geodesic distances is able to provide a kind of robustness for hiding watermark. As a result, the robustness of the proposed method depends on the meaningful parts and the distribution of the geodesic distances. The experimental results are also confirmed that the proposed method can resist the attacks, such as noise addition, similarity transformation, quantization, cropping, simplification, smoothing, and resampling attacks. However, the proposed method is generally more suitable for rigid-body animations than soft-body animations. The segmentation results may not be consistent in the mesh frames of some soft-body animations. In the future, we would like to improve this inconsistency while applying to soft-body animations. Moreover, we will also study the issue of embedding the content-dependent watermark to increase the security of 3D animation watermarking.

ACKNOWLEDGMENT

The mesh animations used in this paper were made available by Robert Sumner and Jovan Popovic from the Computer Graphics Group at MIT. The authors would also like to thanks Lior Shapira, Ariel Shamir, Daniel Cohen-Or to provide the executables for SDF segmentation.

REFERENCES

[1] S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Norwood, MA: Artech House, 2000.

[2] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, San Francisco, CA: Morgan Kaufmann, 2001.

[3] P. Rondao-Alface and B. Macq, "From 3D Mesh Data Hiding to 3D Shape Blind and Robust Watermarking: A Survey," *LNCS Trans. Data Hiding and Multimedia Security*, vol. 4499, pp. 91-115, June 2007.

[4] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "A Comprehensive Survey on Three-Dimensional Mesh Watermarking," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1513-1527, Dec. 2008.

[5] T. H. Kim, J. Lee, and S. Y. Shin, "Robust Motion Watermarking Based on Multiresolution Analysis," *Computer Graphics Forum*, vol. 19, no. 3, pp. 189-198, Sept. 2000.

[6] S. Yamazaki, "Watermarking Motion Data," In *Proc. Pacific Rim Workshop on Digital Steganograph*, pp. 177-185, Nov. 2004.

[7] X. Wu, L. Ma, Z. Dong, and L. Revéret, "Robust watermarking motion data with DL-STDM," *Computer Graphics*, vol. 32, no. 3, pp. 320-329, June 2008.

[8] P. Agarwal and B. Prabhakaran, "Blind Robust Watermarking of 3D Motion Data," *ACM Trans. Multimedia Computing*, *Communications*, *Applications*, vol. 6, no. 1, pp. 1-32, Feb. 2010.

[9] S. H. Lee and K. R. Kwon, "3D Keyframe Animation Watermarking Based on Orientation Interpolator," *IEICE Trans Information System*, vol. E90-D, no. 11, pp. 1751-1761, Nov. 2007.

[10] R. Motwani, A. Ambardekar, M. Motwani, and F. Harris, "Robust Watermarking of 3D Skinning Mesh Animations," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1753-1756, 2008.

[11] L. Shapira, A. Shamir, and D. Cohen-Or, "Consistent Mesh Partitioning and Skeletonisation Using the Shape Diameter Function," *Visual Computer*, vol. 24, no. 4, pp. 249-259, March 2008.

[12] X. Chen, A. Golovinskiy, and T. Funkhouser, "A Benchmark for 3D Mesh Segmentation," *ACM Trans. Graphics*, vol. 28, no. 3, pp. 73:1-73:12, Aug. 2009.

[13] K. Wang, G. Lavoué, F. Denis, A. Baskurt, and X. He, "A Benchmark for 3D Mesh Watermarking," *IEEE International Conference on Shape Modeling and Applications,* pp. 231-235, June 2010.