A New Reversible Watermarking of 3D Models Based on Ratio Expansion

Huamin Ji*, Xingqiang Yang*, Caiming Zhang*†,Xifeng Gao*
*School of Computer Science and Technology
Shandong University, Jinan, China, 250101
†School of Computer Science and Technology
Shandong Economic University, Jinan, China, 250014

***Abstract*—In this paper, a new reversible watermarking method based on ratio expansion for 3D mesh models is proposed. The distances between a vertex and its adjacent vertices are calculated respectively, each of which is then divided by the same reference value to form a series of ratios. Each ratio divided by 2^N is multiplied by 10^M and then the integer part of it is expanded to embed watermark information. The parameter M is adjusted to assure the invisibility. After translation, rotation and scaling, we could still extract the watermark and original model completely. With appropriate M and N, our algorithm is of good invisibility and high embedding capacity as well as robustness against some noises.**

I. Introduction

Reversible data embedding has recently attracted many researchers in the last few years for its wide application. Reversible watermarking is commonly used in images while it is not fully developed in 3D models. According to the location of embedded watermarks, there are algorithms respectively in spatial and frequency domains. Papers [1]–[9] introduce kinds of reversible watermarking algorithms in frequency domain. The invisibility of methods in frequency domain is very good, but it needs large calculations on coefficients resulting in high complexity.

Spatial reversible watermarking technique mainly composes histogram and difference expansion techniques. Hwang Jin Ha [10] brings up a reversible watermarking method based on histogram for images. The maximum is not changed after embedding so that no payload information is needed in extraction. It achieves high capacity but the PSNR is no more than 50 dB. Ni zhi Cheng et al. [11] calculate the image pixel values forming a histogram to embed information. Its disadvantage is that the minimum and the positions of relevant pixels must be saved in embedding process when the minimum value isn't 0. According to the algorithms in [11], Chung Kuo Liang et al. in [12] bring up a dynamic reversible watermarking method which refers to the data pairs of peak and valley, dynamically embed suitable watermark. The complexity of the method is very high and the capacity is still limited due to few peak-valley pairs in histogram of natural images. Histogram technique in 3D mesh modes is introduced in [13]. They calculate integer part of coordinates in 3D models to form a histogram whose maximum and minimum are found out and shift all data between them to save space for watermarks. It is not robust against any attacks because the watermark is embedded directly to the coordinates.

Difference expansion algorithm proposed by Jun Tian [14] has relatively high capacity and low distortion. He partitioned all the pixels into pairs each of which hides one bit. But the pixels being selected are limited because in gray-scale images the pixel value is between 0 and 255. The method brought up by Alattar in [15] increases embedding unit from one pair to three pixels, whose embedding rate is two bits per unit. Moreover, this technique needs less map information than that in [14]. however its PSNR is not high. Yong jian Hu in [16] puts forward a difference expansion method based on row and column directions which introduces the dynamic searching and selecting mechanism of expansible differences to construct two difference maps, therefore it sufficiently employs the relationship of adjacent pixels in two directions. Prediction-error method in 3D models is introduced in [17]. Some embedding units are determined which is composed of a vertex and its adjacent points. The difference between the vertex and the center of its adjacent points is expanded to embed watermarks. However the differences which are relatively large won't be selected to embed data. Literature [18] combines the advantages of both difference expansion and histogram and presents a new reversible watermarking method.

In this paper, we bring up a novel reversible watermarking method for 3D models based on the expansion of integer part of distance ratios. In contrast to the method in [17], we employ distance ratio as a feature which is not only robust against translation, rotation and scaling but also robust against some noises. The distances between vertices are represented by floating point number, and the representation range is not limited so that all the distance ratios can be expanded. The rest of this paper is organized as follows. The details of this algorithm are introduced in Section 2. Then the performance analysis and experimental results are presented in Section 3. Finally, the paper is concluded in Section 4.

II. New Algorithm

A. Data Selection

Most 3D models could be expressed by the triangle mesh, so the researches done in this paper are based on this kind of models.

978-1-4244-6516-3/10/$26.00 ©2010 IEEE

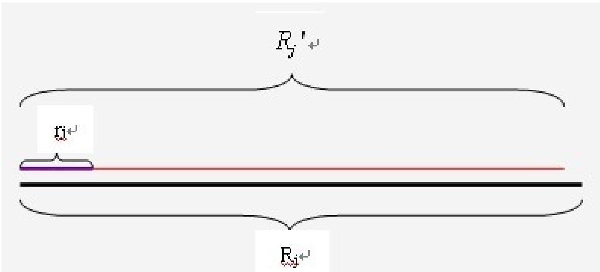


Fig. 1. The General Frame of Data Embedding

For the point v_i in triangle meshes, we denote the set of its adjacent vertices as $Nv(v_i)$ whose degree is K. The Euclidean distance between v_i and each of its adjacent vertices v_{ij} is denoted as Dis(i,j). The coordinates of v_i and v_{ij} are (x_i, y_i, z_i) and (x_{ij}, y_{ij}, z_{ij}) respectively, where $v_{ij} \in Nv(v_i)$.So the distances between v_i and its neighbors can be expressed as follows by Eq.(1)

$$Dis(i,j) = \sqrt{(x_i - x_{ij})^2 + (y_i - y_{ij})^2 + (z_i - z_{ij})^2} \quad (1)$$

According to this equation the distances between v_i and its neighbors are obtained. Select Dis(i,0) as denominator which divides respectively other distances Dis(i,j), where $j \in \{1, 2, ..., K-1\}$, then

$$R_j = \frac{Dis(i,j+1)}{Dis(i,0)}, j = 1, 2, ..., K-1 \quad (2)$$

Consequently the set of the final results is denoted as $U = \{R_1, R_2, ..., R_{K-1}\}$

B. Ratio Expansion

For every ratio R_j calculated in equation (2), a series of steps are done to embed watermarks, and the data is represented by floating point numbers. Whatever the form of watermark is (text, image or 3D model), they all can be transformed into binary strings. So we only deal with watermarks composed of 0 and 1. For the purpose of improving embedding capacity, every N binary bits are converted into a digital number which is in the set $\{0, 1, 2, ..., 2^N - 1\}$.

1) : At the beginning of embedding process, R_j is divided into 2^N pieces, and each piece is r_j, that is to say, $r_j = R_j/2^N$, and then r_j is cut into two parts,

$$r_{\text{int}} = \lfloor r_j \times 10^M \rfloor$$
$$r_{remain} = r_j \times 10^M - \lfloor r_j \times 10^M \rfloor$$

Parameter M represents that watermarks are embedded into the M digits after decimal point of r_j . The value of M determines the invisibility. The bigger of M, the better its invisibility is. Next watermarks are embedded into r_{int},

$$r'_{\text{int}} = 2^N \times r_{\text{int}} + w_i$$
$$R'_j = (r'_{\text{int}} + r_{remain})/10^M$$

Consequently, R_j is replaced by R'_j and $Dis'(i,j)$ is then calculated by the equation $Dis'(i,j) = R'_j \times Dis(i,0)$. Next, the adjacent vertex v_{ij} relevant to R'_j is shifted along the direction $\overrightarrow{v_i v_{ij}}$.

$$(x_{ij}, y_{ij}, z_{ij}) = (x_i, y_i, z_i) + Dis'(i,j) \times \frac{(x_{ij}-x_i, y_{ij}-y_i, z_{ij}-z_i)}{sqrt((x_i-x_{ij})^2+(y_i-y_{ij})^2+(z_i-z_{ij})^2)} \quad (3)$$

2) : In recovery, For R'_j calculated from the watermarked model, the integer part of $R'_j \times 10^M$ is selected to extract watermarks and restore original model.

$$r_{\text{int}} = \lfloor r_j \times 10^M \rfloor$$
$$r_{remain} = r_j \times 10^M - \lfloor r_j \times 10^M \rfloor$$

The restoration of original model is reverse to embedding process and the details are as follows,

$$r'_{\text{int}} = 2^N \times r_{\text{int}} + w_i$$
$$R'_j = (r'_{\text{int}} + r_{remain})/10^M$$

Until now, $R_{reverse} = R_j$, according to the equations (2) and (3) the original coordinates of each vertex are computed.

C. Process of Data Embedding

In embedding process, the length of watermark is embedded to extract watermarks exactly. Every ratio can be expanded, thus no other information needs to be saved in embedding. The steps are as follows,

Step 1 Set the flags of all vertices to 0 and the set V to null. Traverse each vertex of the model, if the flag of point v_i is 0 and all flags of its adjacent vertices are 0, v_i is added to V and the flags of v_i and its adjacent vertices are all set to 1;else continue traversing next vertex.

Step 2 For every vertex v_i in V, according to the method introduced in B.1, each distance from v_i to its adjacent vertex is computed. The standard value is the distance from v_i to its first adjacent vertex, which will respectively divide other distances, and the ratios will be saved.

Step 3 For the results obtained from step 2, we embed watermarks by the method introduced in B and the ratios are changed.

Step 4 According to the changed distance ratios, relevant points are shifted along certain direction, finally forming a watermarked model.

D. Extraction Process

The restoration of original model and extraction of watermarks is just reverse to the procedure of data embedding. Steps are as follows,

Step 1 Follow the same way as steps 1 and 2 did in embedding. The ratio R'_j is attained which is the distance between v_i and each of its adjacent vertex v_{ij}.

Step 2 By the extraction method introduced in B.2, each part of watermark is extracted and meanwhile relevant R_j is obtained.

Step 3 For each center point v_i using the equation $Dis(i,j) = R_j \times Dis(i,0)$, every distance value from v_i to its adjacent vertex v_{ij} is calculated, then the coordinates of all points are acquired using (3) to recover the original model.

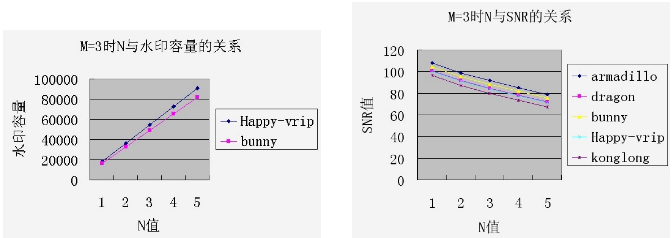
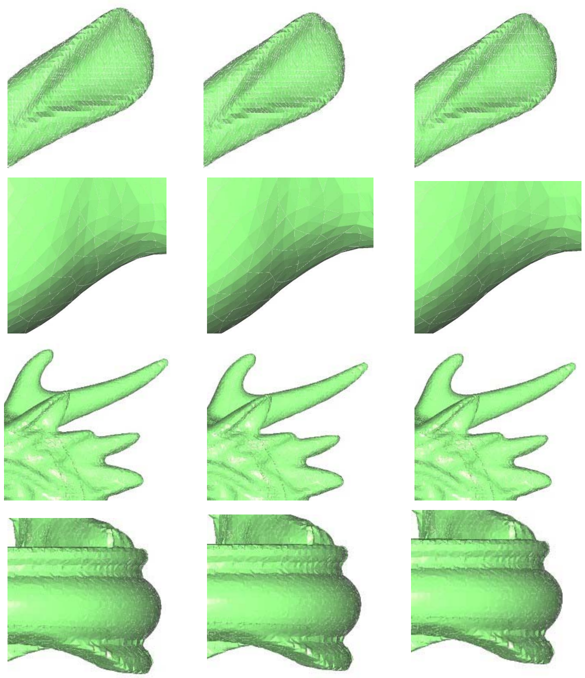


Fig. 2. The relation between N ,SNR and embedding capacity when M is 3

models	Vertices	Faces	capacity	SNR		
				M=3	M=4	M=5
armadillo	172974	345944	188286	98.82	118.83	138.81
bunny	35947	69451	36304	91.58	111.567	131.66
dragon	100250	202520	101550	95.25	115.21	135.22
Happy-vrip	32328	67240	32756	91.17	111.17	131.14
konglong	11322	22640	11970	87.24	107.14	127.18

TABLE I
EXPERIMENTAL RESULTS OF MESH MODELS



(a):original models (b):watermarked models while M is 2 (c):watermarked models while M is 3

Fig. 3. original models and watermarked models

III. Results and Performance Analysis

The most commonly used algorithm metrics standard in 3D models is 3D signal-to-noise ratio (abbreviated as SNR). For a watermarked model, the bigger of its SNR, the better invisibility is. SNR is defined as follows: the set $V = \{v_1, v_2, ..., v_N\}$ represents N vertices in a model, and a vertex position v_i specifies the coordinates $\{v_{ix}, v_{iy}, v_{iz}\}$ in three-dimensional space R^3 for $i \in \{1, 2, ..., N\}$. After embedding watermarks into the mesh model, the vertex positions are changed to $G = \{g_1, g_2, ..., g_N\}$. Thus the 3D SNR of the watermarked model is calculated as follows:

$$\mathrm{SNR} = 10\log_{10}\frac{MS(V-\overline{v})}{MS(G-V)}$$

where $MS(.)$ stands for the mean square while $\overline{v} = (\overline{v}_x, \overline{v}_y, \overline{v}_z)$ is the mean of V, $V - \overline{v}$ represents signal and $G - V = G - \overline{v} - (V - \overline{v})$ is the noise. In addition,

$$MS(V-\overline{v}) = \frac{\sum_{i=1}^{N}(v_{ix}-\overline{v_x})^2+(v_{iy}-\overline{v_y})^2+(v_{iz}-\overline{v_z})^2}{N}$$
$$MS(G-V) = \frac{\sum_{i=1}^{N}(g_{ix}-v_{ix})^2+(g_{iy}-v_{iy})^2+(g_{iz}-v_{iz})^2}{N}$$

In this paper, we conduct our experiments using ”armadillo””bunny””dragon””konglong”and ”Happy-vrip”.

A. High Capacity and SNR

From Fig.2, with all other conditions being equal, the bigger N, the higher its capacity is. This is because N bits are converted into a digit before embedding and one digit can be embedded into a ratio. Besides, the SNR is smaller when N is growing bigger. Weighing the gains and losses, we select 2 as the value of N.

From table 1 we can see that each adjacent vertex can embed more than one bit in this method with high SNR. Because any of the distance ratios could embed watermarks, the capacity has nothing to do with M, only relative to the size and topological structure of models as well as the value of N. The embedding capacity is proportional to N and the number of vertices in a model because in a large model more distance ratios are selected to embed information. In addition, SNR is dependent on M. For the same model, the bigger M, the higher SNR is, indicating that the invisibility is better.

B. Invisibility

In Fig.3, the first column is original models, the second and third column are respectively watermarked models while M is 2 and 3. From the data above in Fig.3, we can see that no visual distortion happens after embedding. Therefore our method meets the demand of invisibility.

C. Robustness to Some Attacks

The proposed algorithm introduces distance ratios between vertices. Therefore, after translation, rotation and uniform scaling, watermarks and original model still can be extracted exactly and completely. In addition, this method embeds watermark into a part of distance ratio, which results in its robustness against some noises. Moreover, the ability to resist noises is in inverse proportion to M, but the invisibility is proportional to M. M and N are both set to 2 in the following experiments.

Random noise is added to all the three coordinates of vertices. Assuming that the original coordinates of a vertex v_i are (x_i, y_i, z_i), the coordinates after noises are (x'_i, y'_i, z'_i) and $(\Delta x_i, \Delta y_i, \Delta z_i)$ is a variable randomly distributed in the section [-a, a], where a is the noise intensity. Then $(x'_i, y'_i, z'_i) = (x_i + \Delta x_i, y_i + \Delta y_i, z_i + \Delta z_i)$.

Fig. 4. original models and extracted watermarks after 100% noises

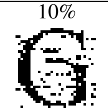
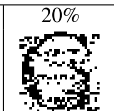
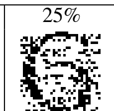
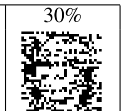
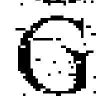
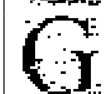
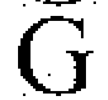
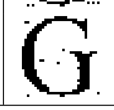
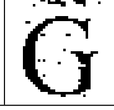
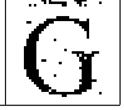
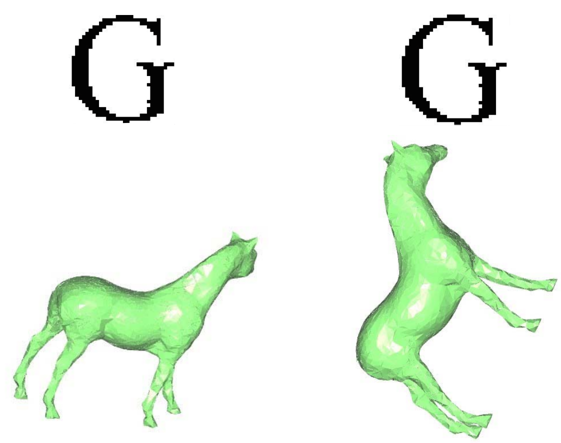
	10%	20%	25%	30%
1.0×10^{-3}				
1.0×10^{-4}				
1.0×10^{-5}				

TABLE II
THE EXTRACTED WATERMARK AFTER ADDING DIFFERENT NOISES IN HORSE



(a): original horse model and watermark

(b):model and extracted watermark after translation, rotation and scaling

Fig. 5. experiments on attacks

Models	Vertices	Faces	Capacity		SNR	
			Our's	Wu's	Our's	Wu's
bunny	35947	69451	108912	104499	85.77	35.71
dragon	100250	202520	304650	300600	89.38	39.48
Happy-vrip	32328	67240	98268	96942	85.31	35.63
konglong	11322	22640	35910	33963	81.50	32.14

TABLE III
RESULTS COMPARED WITH WU'S METHOD

In the following experimental results, we test on horse model with 2367 vertices and 4730 faces. In Fig.5, we can conclude that after translation,rotation and uniform scaling, watermark can still be extracted completely. So our method is robust against translation,rotation and scaling. In Fig.4, the experiments prove that if the noise intensity is 1.0×10^{-5} and 100% noises is added to models, the watermarks could still be extracted (Fig.4).

D. Compared with Other Method

In this part we evaluate reversible watermarking method mainly from two aspects, the capacity and SNR. High capacity and big SNR are expected so that more watermarks can be embedded and less distortion in model is caused. When M is 4 and N is 6, the results are compared with those in paper [17].

From table 3, we can see that when M equals 4 and N is 6, the capacity in our method is a little higher than that of Haotian Wu's algorithm. The SNR in our algorithm is more than 80 dB. However it is no more than 40 dB in paper [17].

IV. CONCLUSION

In this paper, we propose a reversible watermarking algorithm with high capacity, low distortion and robustness against translation, rotation as well as uniform scaling. Distances between vertices are dealt with and then expanded to embed watermarks. Each distance ratio can embed watermarks without any conditions. Moreover in extraction no payload information is needed, truly completing high capacity. The slight changes of distance ratios do not affect the coordinates very much so the embedding can cause no visual distortions if

M and N are all suitable. Also the algorithm is robust against some noises. This method only needs computing distances between vertices with relatively simple calculation and low time complexity. But the selection of vertices needs improving which is our next work.

Acknowledgment

This work is supported by the National Nature Science Foundation of China(60933008,60703081) and Key Science and Technology Project of Shandong Province (No. 2007gg10001004, 2009GG1001013).

References

[1] Y. Bian and S. Martin, "Reversible image watermarking by histogram modification for integer dct coefficients," *In IEEE 6th Workshop on Multimedia Signal Processing*, no. 11, pp. 143–146, September 2004.

[2] G. Xuan and J. Zhu, "Distortionless data hiding based on integer wavelet transform," *In Electronics Letters*, vol. 38, no. 7, pp. 1646–1648, December 2002.

[3] G. Xuan and C. Yang, "Reversible data hiding using integer wavelet transform and companding technique," *In 3rd International Workshop on Digital Watermarking*, vol. 3304, no. 12, pp. 115–124, April 2005.

[4] J. Stach and A. M.Alattar, "A high-capacity, invertible, data-hiding algorithm using a generalized, reversible, integer transform," *In Security, Steganography, and Watermaking of Multimedia Contents VI*, vol. 5306, no. 3, pp. 386–396, January 2004.

[5] A. M.Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *In IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, August 2004.

[6] Z. Sun and L. Zhe-Ming, "Reversible data hiding for 3d meshes in the pvq-compressed domain," *In International Conference on Intelligent Information Hiding and Multimedia Signal Processing,*, no. 14, pp. 591–596, December 2006.

[7] M. U. Celik and G. Sharma, "Lossless generalized-lsb data embedding," *In IEEE Transactions On Image Processing*, vol. 14, no. 19, pp. 253–266, February 2005.

[8] V. L. Arno, V. der Veen Michiel, and F. Bruekers, "Reversible image watermarking," *In IEEE International Conference on Image Processing*, vol. 2, no. 6, pp. 731–734, September 2003.

[9] W. Shaowei, Z. Yao, and P. Jeng-Shyang, "A novel reversible watermarking based on an integer transform," *In International Conference on Image Processing*, vol. 3, no. 9, pp. III241–III244, September 2007.

[10] H. J. Ha, K. J. Weon, and C. HonqUk, "A reversible watermarking based on histogram shifting," *In 5th International Workshop on Digital Watermarking*, vol. 4283LNCS, no. 11, pp. 348–361, November 2006.

[11] N. Zhicheng and S. Yun-Qing, "Reversible data hiding," *In IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 20, pp. 354–361, March 2006.

[12] C. Kuo-Liang, H. Yong-Huai, and Y. Wei-Ning, "Capacity maximization for reversible data hiding based on dynamic programming," *In Applied Mathematics and Computation*, vol. 208, no. 25, pp. 284–292, February 2009.

[13] J. Chang-Yun1, P. Jeng-Shyang, and C. Ding, "Reversible data hiding base on histogram shift for 3d vertex," *In 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, vol. 1, no. 5, pp. 365–368, November 2007.

[14] T. Jun, "Reversible data embedding using a difference expansion," *In IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, August 2003.

[15] A. M., "Reversible watermark using difference expansion of triplets," *In IEEE International Conference on Image Processing*, vol. 1, no. 9, pp. 501–504, September 2003.

[16] Y. Hu, H.-K. Lee, and K. Chen, "Difference expansion based reversible data hiding using two embedding directions," *In IEEE Transactions on Multimedia*, vol. 10, no. 15, pp. 1500–1512, December 2008.

[17] H. W.-L. Dugelay, "Reversible watermarking of 3d mesh models by prediction-error expansion," *In IEEE 10th Workshop on Multimedia Signal Processing*, no. 20, pp. 797–802, October 2008.

[18] H. Hsiang-Cheh and F. Wai-Chi, "Reversible data hiding using histogram-based difference expansion," *In IEEE International Symposium on Circuits and Systems*, no. 7, pp. 1661–1664, May 2009.