JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 23, 1889-1900 (2007)

Reversible Data Hiding for Point-Sampled Geometry

PENG-CHENG WANG[1,2] AND CHUNG-MING WANG[1]
[1]*Institute of Computer Science*
National Chung Hsing University
Taichung, 402 Taiwan
[2]*Department of Information Management*
Hsiuping Institute of Technology
Taichung, 412 Taiwan

We present a novel reversible data hiding scheme for point-sampled geometry in the spatial domain. To the best of our knowledge, our scheme is the first in the literature for recovering the original point-sampled model using little amount of information. In contrast, other schemes generally need to store an extra large amount of bit size equal to the payload size for reversibility. Our scheme first employs principal component analysis (PCA) for the point-sampled model to produce three principal axes and to construct a PCA-coordinate system. We then translate the coordinates of the original points to the PCA-coordinate system in order to achieve robustness against translation, rotation, and uniform scaling operations. Second, we sort the points' coordinates for each axis to yield intervals which are the embedding positions. Finally, we utilize the left-shift operator to shift the bit of the state value of the interval left by c bits ($c \geq 1$) so that it creates c bits extra storage space for embedding the payload and we store the original state value for achieving reversibility. Experimental results show that our scheme can embed large amounts of data with insignificant visual distortion of the original model. The data capacity in bits achieves nearly $1.5c$ times the number of points in the models.

Keywords: point-sampled geometry, reversible, steganography, watermarking, PCA

1. INTRODUCTION

Most data hiding algorithms use digital multimedia data such as movies, music, images, and 3D models as cover media to embed hidden information, which are usually denoted as the payload [1]. Often, these algorithms distort the cover model in an irreversible way, *i.e*. they cannot exactly recover the original model. But there are many applications where any distortion of the cover model is intolerable and the acquisition of the cover model is too costly or too difficult to be performed repeatedly. There are also some applications where identical reproduction of the original data is required due to juridical, military, or medical demands. Even though reversible data hiding is an active research field and its application to 2D images has been studied, reversible data hiding for 3D models is under studied. This emphasizes the need to investigate reversible (lossless) data hiding algorithms for 3D models.

Point-sampled geometries are a versatile representation for geometric models [2]. An example of this has been the increased use of 3D scanning systems, which produce sampled models of 3D objects. Numerous point-based rendering systems have been developed as alternatives to 3D polygonal representations. Unfortunately, research in steganography has not kept pace with the advances of point-sampled geometries. Even

Received November 8, 2005; revised March 3 & May 8, 2006; accepted May 24, 2006.
Communicated by Ja-Ling Wu.

though some data hiding and watermarking schemes have been presented for 3D polygonal models [3-16], few have been proposed for point-sampled geometries so far [17-19]. Therefore, the need to provide data hiding schemes for point-sampled geometry is clear.

Generally speaking, data hiding in the spatial domain such as [4] increases capacity but leads to relatively poor robustness and visual quality. Steganography is a suitable solution for many applications, including covert communication and authentication. In watermarking, however, the transform domain offers better robustness but results in less payload size. For example, both [18] and [19] embedded 16-56 data bits into 3D models. A common and widely used application for watermarking is the ownership assertion. Since our main focus is for data hiding but not for watermarking, large payloads are our concern. For this reason, we embed the information into the spatial domain and assume light robustness such as similarity transformations including translation, rotation, and scaling. This robustness requirement corresponds to many common data hiding algorithms presented for 3D models, *e.g.*, the one presented by Cayre and Macq [4].

In general, reversible data hiding schemes [3, 4, 14, 17] need to store extra large amounts of data to recover the original 3D model and to embed a large payload into the cover model. Since a large payload degrades the visual quality, most techniques rely on lossless compression of the embedded data. Our algorithm utilizes the same PCA and sorting approach as the data hiding algorithm we previously presented [17]. However, now we present a reversible technique for 3D point-sampled model in this paper. In particular, we utilize a left-shift operator on the state values of the intervals to generate extra storage space (thus removing the need for lossless compression) for embedding the payload. We thus achieve reversibility without storing extra large amounts of data.

This paper is organized as follows: Section 2 surveys related works. Section 3 presents in detail the proposed algorithms. Section 4 shows the visual effect for several models and describes experimental results. Conclusions are described in the final section.

2. RELATED WORKS

The literature is quite limited in relation to reversible data hiding for 3D models. Some methods have been proposed by Cayre and Macq [3, 4], by Wang and Cheng [14], and by Wang and Wang [17]. Because these all use similar methods to achieve reversibility, we present here only one example of a reversible data hiding scheme [4]. After that, we survey some data hiding and watermarking schemes of 3D models in the spatial domain and in the transform domain.

The earliest research on reversible data hiding for 3D polygonal models was by Cayre and Macq [4]. There the authors describe a blind scheme in the spatial domain. The key idea is to consider a triangle as a two-state geometrical object, which means that the position of the orthogonal projection of the triangle summit on the bottom edge can be divided into a “0” or “1” state according to the bit to be hidden. This data hiding method first establishes a list of triangles of the model that will contain the payload. Each triangle that can be embedded with a bit is called an admissible triangle. Then, the vertex’s position at each admissible triangle is modified or not according to the embedding bits. In other words, when the embedding bit is the same as the state of the triangle, then

no modifications need to be processed. Otherwise, each vertex's position has to be shifted toward the other side across the symmetry axis. If their algorithm wants to recover the original model, it is necessary to store an extra bit for every bit of the payload, and its size is exactly the same as the payload size. The extra bit is set to 1 if the state of the triangle is changed and 0 otherwise. This scheme is robust against translation, rotation, and scaling operations. However, it is not robust against remeshing, mesh simplification, mesh altering, cropping, and other operations.

Benedens [5] proposed an algorithm that embeds private watermarks by altering normal distribution. It works in the spatial domain. This algorithm achieves robustness against randomization of points, mesh altering, and polygon simplification. However, the algorithm is not robust against cropping attacks, as the normal distribution is calculated for the entire model.

Kanai *et al.* [6] were the first to apply a transform-domain watermarking approach to a 3D polygonal model. Their algorithm is a non-blind scheme, which decomposes an original polygonal model by applying wavelet transform several times and then embeds watermarks into the wavelet coefficient vectors. This algorithm is robust against affine transformation, random noise added to vertex coordinates, and other attacks. Their method requires the mesh in a 3D polygonal model to have 4-to-1 subdivision connectivity.

Ohbuchi *et al.* [8, 9] proposed a non-blind algorithm that works in the transform domain. It embeds the watermarks into a 3D polygonal model by deforming the low-frequency components of the shape using mesh spectral analysis. This scheme achieves robustness against similarity transformation, cropping, random noise, mesh simplification, and other operations. The main drawback is that it requires much execution time for the eigenvalue decomposition.

Zafeiriou *et al.* [16] recently proposed a different method for the spatial domain. They present two novel approaches for blind 3D polygonal model watermarking applications. Before embedding and extraction, they first rotate and translate the 3D model so that its gravity center and its principal component axis coincide with the origin and the z-axis of the Cartesian coordinate system, respectively. To achieve robustness against uniform scaling, vertex deformations can only occur along the r coordinate of the corresponding (r, θ, φ) spherical coordinate system. In the first method, they embed a watermark into a set of vertices that correspond to specific angles θ. In the second method, they embed a watermark into a set of vertices that correspond to a range of angles in the θ domain in order to achieve robustness against mesh simplifications. The first method is robust against rotation, translation, and uniform scaling. The second is robust against both affine transformations and mesh simplification attacks. Unfortunately, they are not robust against more sophisticated attacks, such as cropping.

Cotting *et al.* [18] reported a non-blind watermarking approach for point-sampled geometry. It extends existing watermarking algorithms [8, 9] designed for 3D polygonal models to point-sampled geometry. Their approach embeds watermarks into the low-frequency components; it employs statistical methods based on correlation to analyze the extracted watermarks. This scheme is robust against low-pass filtering, resampling, affine transformations, cropping, additive random noise, and combinations of the above. Similarly, this algorithm also requires much execution time for the eigenvalue decomposition.

Ohbuchi *et al*. [19] presented a non-blind watermarking algorithm for point-sampled geometry. The algorithm takes the same transform-domain shape modification approach as the watermarking algorithm presented in their previous paper [9]. Note that a polygonal model requires connectivity of vertices for the mesh spectral analysis, while a point-sampled model doesn't. Thus, the major difference between the two papers is that the algorithm in [19] first converts the point-sampled model into a (non-manifold) polygonal model. Then, it applies exactly the same watermarking approach shown in [9]. As a result, this scheme has the same robustness and downside as that in [9].

This brief summary of the limited available literature shows that current reversible algorithms need to store extra large amounts of data to recover the original model. In this paper, we present a reversible data hiding approach for point-sampled geometry using very little information.

3. THE PROPOSED TECHNIQUE

This section presents an algorithm for reversible data hiding. Our algorithm is designed to accept a point-sampled model (the cover model), a sequence of binary bits (referred to as the payload), and the secret key. Fig. 1 illustrates the outline of the proposed scheme. In the embedding process, a special case of quantization index modulation is extended to the point-sampled model. We use a secret key to generate a random sequence of legitimate intervals (define later), which are the embedding positions. Specifically, by modulating the positions of the legitimate points inside the legitimate intervals, a sequence of data bits is embedded into the model. To achieve reversibility, we need to record the original position prior to the modulation. We utilize left-shift operator (detailed later) on the legitimate points, which have finite and acceptable precision (six decimal digits), to record the position of the original points. We also employ a tolerance value to measure the reversibility of a data hiding scheme. In other words, a data hiding scheme is a reversible scheme when the difference between the cover and stego models is no greater than a given tolerance. The embedding algorithm embeds a payload according to the three steps below:

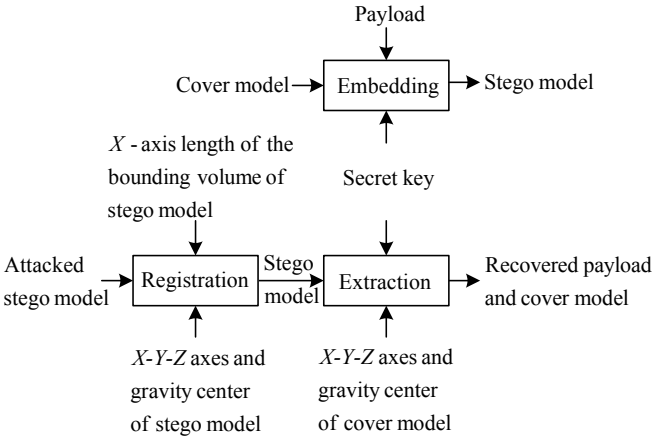

Fig. 1. Overview of our scheme: embedding and extraction.

(1) Constructing PCA axes: We adopt PCA [20] for the point-sampled model to produce three principal axes, constructing a PCA-coordinate system. We then translate the coordinates of the original points to the PCA-coordinate system. Obviously, this PCA-coordinate system has a new origin, which is a gravity center of the point-sampled model; it also has three basis vectors, which are the three principal axes, referred to as the *X*-, *Y*-, and *Z*-axis, respectively.
(2) Creating three sorting lists: After the coordinate translation, we sort points according to their coordinates associated to the PCA-coordinate system for each axis. This produces three lists containing sorted indices.
(3) Embedding data into lists: For simplicity, we use only the *X*-axis to illustrate the embedding process. Suppose $X_1, X_2, \ldots, X_m$ are *X*-coordinate values of sorted points $P_1, P_2, \ldots, P_m$, where m is the number of points in the point-sampled model, and each interval (which consists of two values, X_n and X_{n+2} ($1 \le n \le m-2$) on the *X*-axis) is considered an i-state object. Prior to data embedding, the i is set to two. Afterward, embedding c bits into each interval, the i is changed from two to 2^{c+1}. We define the state of the interval by the *X*-coordinate value X_{n+1} of the point P_{n+1}. We divide the interval X_nX_{n+2} into i equal parts. If X_{n+1} is located on the sth ($0 \le s \le i-1$) subinterval, then we consider the interval to be in s state, as shown in Fig. 2. Before embedding the payload, we first locate a legitimate interval of the *X*-axis according to the secret key and divide the interval into two equal parts, indicating the r ($r = 0$ or 1) state. To achieve exact reversibility, we utilize the left-shift operator to shift the bit of the state value r left by c bits. Thus, we can store the original state value r in the most significant bit (MSB) of the new state s, whose binary representation is (s_c, $s_{c\text{-}1}, \ldots, s_0$), and create c bits of free space to embed value d into this interval by

$$s = 2^c \times r + d \ (0 \le s \le 2^{c+1} - 1), \tag{1}$$

where $r = s_c$ and $d = (s_{c\text{-}1}, s_{c\text{-}2}, \ldots, s_0)$. Finally, the new *X*-coordinate value of the legitimate point P_{n+1} is computed as

$$X'_{n+1} = X_n + s \times \frac{X_{n+2} - X_n}{2^{c+1}} + \frac{\lambda}{2^c}, \tag{2}$$

where

$$\lambda = X_{n+1} - (1-r) \times X_n - r \times \frac{X_n + X_{n+2}}{2} \tag{3}$$

is the distance between X_{n+1} and the left border of the closest subinterval. In this way, visual quality is not much affected by the large payload size, because we limit the modified coordinate value within the interval range.

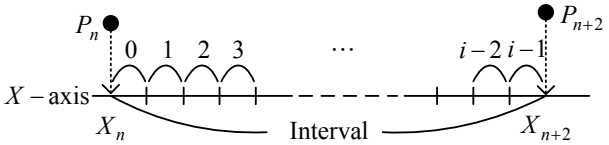

Fig. 2. Dividing the interval X_nX_{n+2} into i subintervals with i state values for every subinterval.

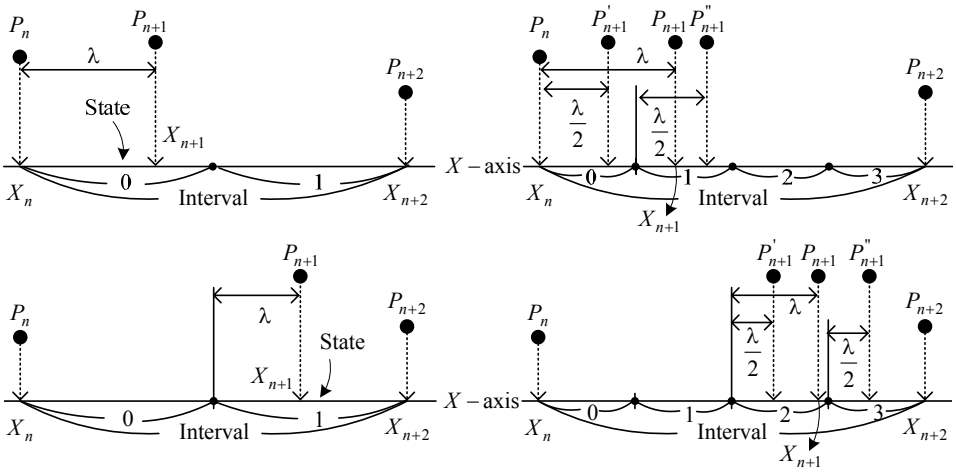

Fig. 3. An illustration of the states and positions of the points before (left) and after (right) the embedding.

Fig. 3 shows a simple example that embeds a payload with one bit (c = 1) into the interval. The left and right sides of the diagram represent the states and positions of the points before and after the embedding, respectively. In the top left figure, the original state r of the interval is in the 0 state. According to the definition, if we embed a bit 0 (d = 0) into the interval, the new state is s = 0 given by Eq. (1), and the new X-coordinate value is $X'_{n+1} = X_n + \frac{\lambda}{2}$ given by Eq. (2). In other words, the position of the point P_{n+1} is shifted toward P'_{n+1}. Assume that the embedding data is d = 1. Then the new state is s = 1 and the new X-coordinate value is $X'_{n+1} = X_n + \frac{X_{n+2} - X_n}{4} + \frac{\lambda}{2}$. Finally, the position of the point P_{n+1} is shifted toward P''_{n+1}. Similarly, in the bottom left figure, we set the interval to be state 1. After embedding one bit 0 or 1, the interval is changed into state 2 or 3 and the position is shifted toward P'_{n+1} or P''_{n+1}, respectively.

Using our extraction algorithm, we proceed to data extraction with (1) an attacked stego model, (2) the secret key, (3) three principal axes plus the gravity center of both the cover model and the stego model, and (4) the X-axis length of the bounding volume for the stego model, as shown in Fig. 1. To counter the attack of similarity transformation on the stego model, we first perform model registration according to the two steps below:

(1) Compute the three principal axes and gravity center of the attacked stego model, and translate the coordinates of the attacked stego model to its PCA-coordinate system. We then translate the coordinates of the attacked stego model to the PCA-coordinate system of the stego model.
(2) Compute the X-axis length of the bounding volume of the attacked stego model. Utilizing the proportion of X-axis length of the attacked stego model to X-axis length of the stego model, we scale the attacked stego model so that the X-axis length of the scaled model is equal to the X-axis length of the stego model.

We then take the next three steps for each axis:

(1) Find a legitimate interval X_nX_{n+2} by interpreting the secret key.
(2) Calculate the state value s of the interval with a given 2^{c+1} subintervals according to the X-coordinate value X_{n+1}. The original state can be calculated as follows:

$$r = \left\lfloor \frac{s}{2^c} \right\rfloor, \tag{4}$$

where the symbol $\lfloor \bullet \rfloor$ is the floor function meaning "the greatest integer less than or equal to," and the embedded data can be extracted by the following equation:

$$d = s - r \times 2^c. \tag{5}$$

Finally, we can restore the original X-coordinate value by

$$X''_{n+1} = (1-r) \times X_n + r \times \frac{X_{n+2} + X_n}{2} + 2^c \times \lambda', \tag{6}$$

where

$$\lambda' = X_{n+1} - X_n - s \times \frac{X_{n+2} - X_n}{2^{c+1}}. \tag{7}$$

(3) Repeat steps (1) and (2) above for all the legitimate points P_{n+1} on a given stego model.

In this approach, we can embed c bits at each interval, resulting in a total of $mc/2$ bits for each axis. Clearly, the data capacity is $1.5mc$ bits for the X-Y-Z axes, where m is the number of points in the point-sampled model. On the other hand, to construct PCA axes requires the computation complexity of $O(m)$ time. Afterward, the cost of embedding the payload is $O(m \log m)$. The cost derives mainly from sorting points, according to the coordinate values of the X-Y-Z axes.

Finally, we establish evaluation metrics in terms of the imperceptibility, security, capacity, and complexity for the proposed method. Some proposals have been made for 3D polygonal models, mostly based on the Hausdorff distance [4] and Laplacian [21]. In the 3D point-sampled geometry, we employ a simple evaluation metric, the root mean square (RMS) ratio, to measure the 3D model distortion. The RMS is the average of the norm of the geometric distance between corresponding points in both cover model M^1 and stego model M^2 [21]. The RMS is expressed as follows:

$$||M^1 - M^2|| = \frac{1}{m} ||v^1 - v^2||, \tag{8}$$

where v is the point set of M. To normalize the RMS values for various point-sampled models, we utilize the RMS ratio to measure geometrical distortion by dividing the RMS value over the diagonal length of the bounding volume for a point-sampled model. Security is achieved by using a secret key to hide a payload in an undeterministic way. Since our main focus is capacity, we assume that no robustness is required, except for some

basic operations, such as translation, rotation, and uniform scaling. Finally, we estimate the computation complexity of our scheme by giving the execution time for various models.

4. EXPERIMENTAL RESULTS

We implemented the proposed technique using C++ programming language. We also performed experiments to validate the feasibility of our algorithms. Results were collected on a personal computer with a Pentium IV 2.8 GHz processor and 512 MB memory. For simplicity, we only hid one bit per interval in the experiment even though c bits could be embedded into each interval. Clearly, the value c depends on the machine's degree of precision and the distribution of the points in each axis. In addition, we used six decimal digits, such as 0.091278, for coordinate values of the points in the point-sampled models. After extraction, the original cover model could be restored with the tolerance

$$T = 2 \times 10^{-6} \geq ||p^1 - p^2||, \tag{9}$$

where p^1 and p^2 are the corresponding points in the cover model and recovered model, respectively.

Table 1. Model size, embedding data capacity, RMS ratio, and execution times for PCA, embedding, and extraction.

Cover	Number of points (model size)	Data capacity (bits)	RMS ratio	PCA execution time (seconds)	Embedding execution time (seconds)	Extraction execution time (seconds)
Dinosaur	56,194	83,976	2.99×10^{-6}	0.047	0.156	0.032
Horse	48,485	72,456	3.78×10^{-6}	0.047	0.094	0.032
Bunny	35,947	53,862	5.26×10^{-6}	0.032	0.078	0.031
Venus	33,591	50,310	5.26×10^{-6}	0.032	0.078	0.016

Table 1 contains, for various point-sampled models, the data capacities, RMS ratios, and execution times for PCA, embedding, and extraction. As expected, the data capacity is nearly 1.5 times the number of points when we hide one bit per interval. The small RMS ratios indicate insignificant position changes during the data hiding scheme. The embedding times are longer than the extraction times, because we had to sort points to construct intervals and their corresponding points. We detected no errors when extracting the payloads.

From the security point of view, finding the sequence of intervals is a challenge for an attacker. We believe that extracting the embedded data without the key is virtually impossible. This is because the problem is NP-hard with respect to the number of intervals in the point-sampled model. On the other hand, the payload can be encrypted in advance by some cryptographic algorithms such as DES or RSA. In this way, the payload will not be exposed even though it is extracted. Therefore, it is clear that our scheme is secure in the cryptographic sense.

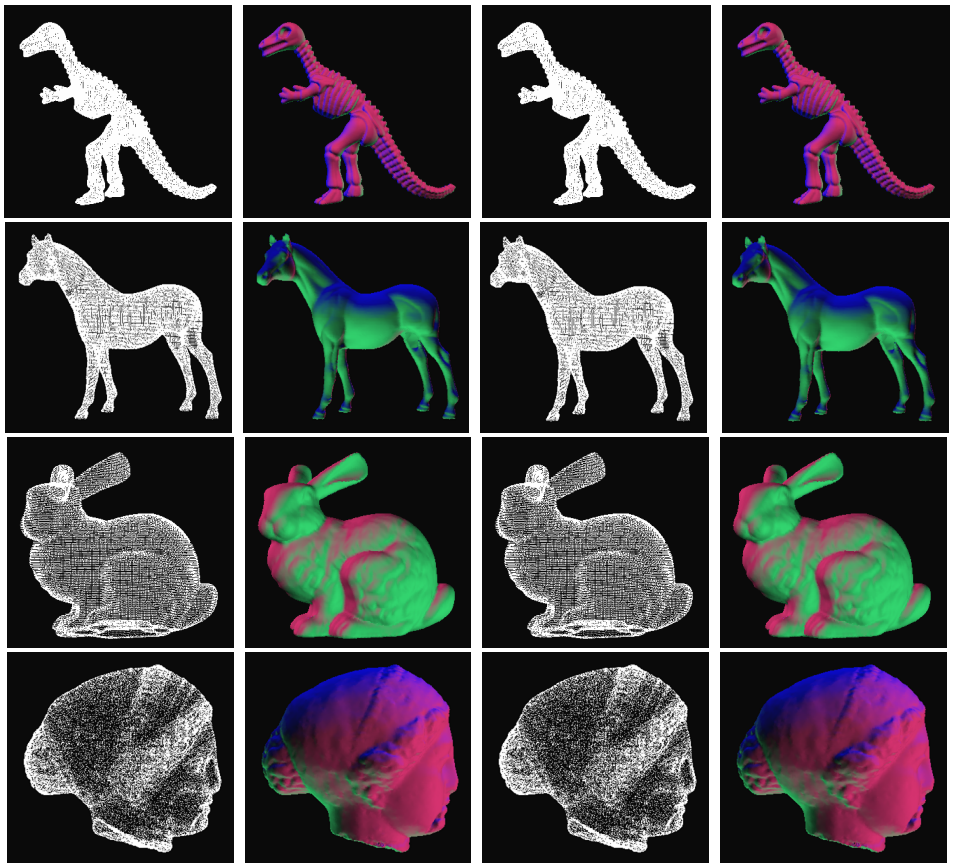

Fig. 4. Cover models (middle-left) and stego models (middle-right) for **Dinosaur**, **Horse**, **Bunny**, and **Venus**.

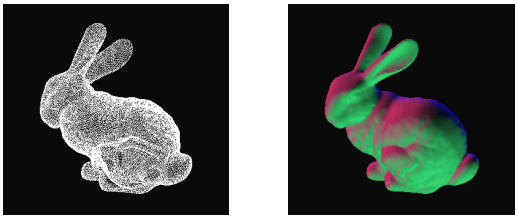

Fig. 5. Rotation and down scaling attacks for **Bunny**.

Fig. 4 demonstrates a point and shading display for the cover (middle-left) and the stego (middle-right) models, and it compares the visual effects between the cover and stego models. Note that no visual distortion can be perceived between the cover and stego models. In addition, we did not embed any payloads at the legitimate points inside an interval when the distance between legitimate points and neighboring points in the same interval was smaller than a given threshold. This avoided possible extraction errors due to machine imprecision.

Fig. 5 shows the stego model with the similarity transformation attack. No extraction error occurred after the model registration. However, if a stego model is under some attacks, *e.g.*, by keeping fewer decimal digits or by down-sizing causing significant precision reduction, an extraction error will occur.

5. CONCLUSIONS

Several reversible data hiding schemes are available in the spatial domain for 3D models. Our scheme is the first one to restore the original 3D point-sampled model without storing extra large amounts of data, which is generally the same as the payload size. This paper presented a high capacity, high quality, reversible data hiding scheme using PCA. Our scheme is robust against translation, rotation, and uniform scaling operations; and it is simple to implement. By utilizing the left-shift operator on the state of the interval, reversibility is achieved. Furthermore, it generates extra storage space to embed large amounts of data. Therefore, compression is not necessary for our scheme. Of course employing compression can increase the hiding capacity. In terms of data capacity in bits, our method achieves nearly 1.5c times the number of points in the cover models. To address the issue of imperceptibility, we used a metric, RMS ratio, to measure distortion. This and the visual appearance of images showed insignificant distortion for the stego models. Security was also achieved by using a key to embed the data. We believe that extracting the data without the key is virtually impossible. In addition, we estimated the complexity of our scheme by giving execution times for various models. Finally, we demonstrated the feasibility of our algorithms for a number of point-sampled models. We believe our scheme is appropriate for most point-sampled models.

REFERENCES

1. S. Katzenbeisser and F. A. P. Petitcolas, ed., *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, London, 2000.
2. L. Kobbelt and M. Botsch, "A survey of point-based techniques in computer graphics," *Computers & Graphics*, Vol. 28, 2004, pp. 801-814.
3. F. Cayre, O. Devillers, F. Schmitt, and H. Maître, "Watermarking 3D triangle meshes for authentication and integrity," Research Report No. RR-5223, Inria, 2004.
4. F. Cayre and B. Macq, "Data hiding on 3-D triangle meshes," *IEEE Transactions on Signal Processing*, Vol. 51, 2003, pp. 939-949.
5. O. Benedens, "Geometry-based watermarking of 3D models," *IEEE Computer Graphics and Applications*, Vol. 19, 1999, pp. 46-55.
6. S. Kanai, H. Date, and T. Kishinami, "Digital watermarking for 3D polygons using multiresolution wavelet decomposition," in *Proceedings of the 6th IFIP WG 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications* (GEO-6), 1998, pp. 296-307.
7. R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models," in *Proceedings of ACM Multimedia*, 1997, pp. 261-272.
8. R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama, "Watermarking 3D polygonal meshes in the mesh spectral domain," in *Proceedings of Graphics Inter-*

face, 2001, pp. 9-17.
9. R. Ohbuchi, A. Mukaiyama, and S. Takahashi, "A frequency-domain approach to watermarking 3D shapes," *Computer Graphics Forum*, Vol. 21, 2002, pp. 373-382.
10. M. G. Wagner, "Robust watermarking of polygonal meshes," in *Proceedings of Geometric Modeling and Processing*, 2000, pp. 201-208.
11. N. Aspert, E. Drelie, Y. Maret, and T. Ebrahimi, "Steganography for three-dimensional polygonal meshes," in *Proceedings of SPIE 47th Annual Meeting*, 2002, pp. 705-708.
12. Y. Maret and T. Ebrahimi, "Data hiding on 3D polygonal meshes," in *Proceedings of Multimedia and Security Workshop*, 2004, pp. 68-74.
13. F. Uccheddu, M. Corsini, and M. Barni, "Wavelet-based blind watermarking of 3D models," in *Proceedings of Multimedia and Security Workshop*, 2004, pp. 143-154.
14. C. M. Wang and Y. M. Cheng, "An efficient information hiding algorithm for polygon models," *Computer Graphics Forum*, Vol. 24, 2005, pp. 591-600.
15. J. Wu and L. Kobbelt, "Efficient spectral watermarking of large meshes with orthogonal basis functions," in *Proceedings of Pacific Graphics*, 2005, pp. 848-857.
16. S. Zafeiriou, A. Tefas, and I. Pitas, "Blind robust watermarking schemes for copyright protection of 3D mesh objects," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 11, 2005, pp. 596-607.
17. C. M. Wang and P. C. Wang, "Data hiding approach for point-sampled geometry," *IEICE Transactions on Communications*, Vol. E88-B, 2005, pp. 190-194.
18. D. Cotting, T. Weyrich, M. Pauly, and M. Gross, "Robust watermarking of point-sampled geometry," in *Proceedings of International Conference on Shape Modeling and Applications*, 2004, pp. 233-242.
19. R. Ohbuchi, A. Mukaiyama, and S. Takahashi, "Watermarking a 3D shape model defined as a point set," in *Proceedings of International Conference on Cyberworlds*, 2004, pp. 392-399.
20. A. C. Rencher, *Methods of Multivariate Analysis*, 2nd ed., Wiley, New York, 2002.
21. Z. Karni and C. Gotsman, "Spectral compression of mesh geometry," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 2000, pp. 279-286.

Peng-Cheng Wang (王鵬程) received the B.S. degree in Computer Science from Tunghai University, Taichung, Taiwan, in 1987, the M.S. degree in Computer Science and Information Engineering from National Chung Cheng University, Chiayi, Taiwan, in 1992, and the Ph.D. degree in Computer Science, National Chung Hsing University, Taichung, Taiwan, in 2006. Now, he is currently an associate professor at the Department of Information Management, Hsiuping Institute of Technology, Taiwan. His research work focuses on computer graphics and data hiding.

Chung-Ming Wang (王宗銘) received the B.S. degree in Applied Mathematics from the National Chung Hsing University, Taiwan, in 1984, and the Ph.D. degree in Computer Science from the University of Leeds, Leeds, U.K., in 1992. From August 1986 to August 1988, he was a system analyst in the President Enterprise Cooperation and the PAL Company. He is currently a professor at the Institute of Computer Science, National Chung Hsing University, Taiwan. His research interests include computer graphics, color science, virtual reality, multimedia systems, and three-dimensional watermarking and steganography. Dr. Wang has won three Dragon Thesis Awards, funded by the Acer Computers, and the Outstanding Paper Awards from the Computer Society of the Republic of China. He is a member of ACM, IEEE Computer Society and Eurographics.