

Research Article

A Distortion-Free Data Hiding Scheme for Triangular Meshes Based on Recursive Subdivision

Yuan-Yu Tsai^{1,2}

¹Department of M-Commerce and Multimedia Applications, Asia University, Taichung 413, Taiwan

²Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 404, Taiwan

Correspondence should be addressed to Yuan-Yu Tsai; yytsai@asia.edu.tw

Received 24 November 2015; Revised 22 April 2016; Accepted 12 June 2016

Academic Editor: Balakrishnan Prabhakaran

Copyright © 2016 Yuan-Yu Tsai. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study adopts a triangle subdivision scheme to achieve reversible data embedding. The secret message is embedded into the newly added vertices. The topology of added vertex is constructed by connecting it with the vertices of located triangle. For further raising the total embedding capacity, a recursive subdivision mechanism, terminated by a given criterion, is employed. Finally, a principal component analysis can make the stego model against similarity transformation and vertex/triangle reordering attacks. Our proposed algorithm can provide a high and adjustable embedding capacity with reversibility. The experimental results demonstrate the feasibility of our proposed algorithm.

1. Introduction

Reversible data hiding algorithms [1–12] can recover the marked media to original one after the secret message is correctly extracted. The important media, such as those for medical or military applications, can apply such algorithms to carry the secret message. Therefore, researches on reversible data hiding have been springing up quickly in recent years.

Even though reversible image data hiding is a very hot research issue, algorithms for 3D models have not been intensively studied. Besides, most reversible data hiding algorithms for 3D models usually directly extended the ones for images to 3D models. For example, Jhou et al. [4] applied histogram modification on 3D models. They first calculate the gravity center of the input model and the embedding order for each vertex is then determined based on its distance to the center. Thereafter, the last d -digit of all coordinate values is used for histogram construction. The message is embedded into the coordinate values whose last d -digit is located at the peak point of the histogram. Chuang et al. [1] employed the Helmert transformation to raise the robustness against similarity transformation attacks. Their algorithm performs data embedding on the normalized distance between each vertex and the model center. Huang and Tsai [3] used the difference between each vertex and its neighboring ones

for histogram construction. Experimental results prove that their proposed algorithm can achieve higher embedding capacity, acceptable distortion, and greater robustness. Nevertheless, Wu and Dugelay [9] proposed a reversible data hiding algorithm based on difference expansion. They first predict a vertex position by calculating the center of its traversed neighbors. The secret message is then embedded by expanding the difference between the predicted and the original coordinate values. Zhu et al. [12] expand half aforementioned prediction difference to embed the watermark. Their proposed algorithm can provide lower distortion and controllable distortion. In contrast to above methods, Ji et al. [5] expand the distance ratio for data embedding to raise the robustness against noise attacks. Wu and Wang [10] first form a sequence of vertices by adopting a mesh traversal strategy. The secret message is embedded into the difference between every two adjacent vertices based on difference expansion and difference shifting.

Recently, Jung and Yoo [6] proposed the first interpolation-based reversible data hiding algorithm for images. Some improved algorithms [7, 8] are then proposed to improve its performance. The interpolation-based reversible data hiding algorithms first enlarged the input image to cover image based on their proposed interpolation scheme.

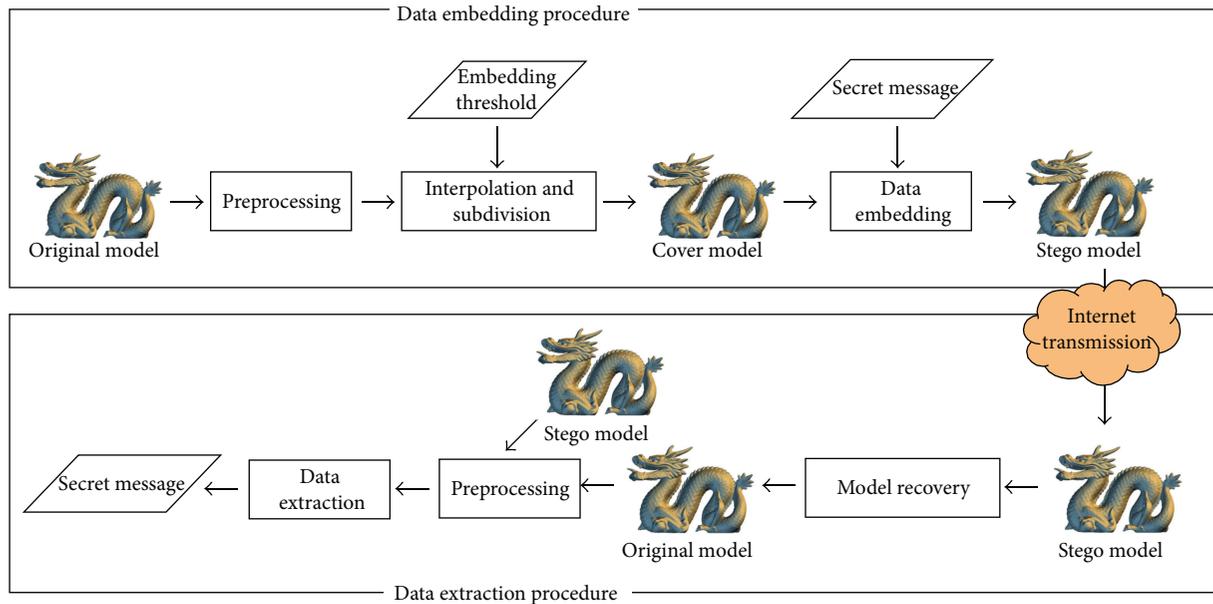


FIGURE 1: A flowchart of the proposed algorithm.

The secret message is embedded into the newly added interpolated pixels. In other words, more secret message can be embedded with larger magnifying factors. Because the pixels in the input image are not altered in the data embedding procedure, the input image can be recovered without any error. However, above proposed algorithm is currently applicable to grayscale images.

Similar interpolation concept for 3D models is so-called subdivision surface, representing a smooth surface via a recursive subdivision process on each given coarse polygonal face into smaller ones. Although some watermarking algorithms [13, 14] are operated on subdivision surface, they are either nonblind or irreversible or have low embedding capacity limited by the refinement process of subdivision surface.

This study extends the interpolation idea to triangular meshes. The algorithm first generates newly added vertices with the message embedded within one corresponding triangle. We then construct their topology with the vertices of located triangle. To raise the total embedding capacity, a recursive subdivision mechanism, terminated by a given criterion, is employed. Finally, a principal component analysis [15] makes our stego model against similarity transformation and reordering attacks.

This rest of this paper is organized as follows: Section 2 details the proposed technique; Section 3 presents a discussion on the experimental results; and, finally, Section 4 offers a conclusion and directions for future research.

2. Proposed Method

This section details the proposed distortion-free data hiding algorithm for triangular meshes. A triangle subdivision approach is performed on triangle meshes and the

secret message is embedded into the newly added vertices. Because the original vertex is not altered during the entire data embedding procedure, the original triangle meshes can be recovered without any error. Further, a recursive subdivision is employed to raise the embedding capacity. A given criterion terminates the recursion, considering the fixed number precision for representing the stego model. For being robust against the similarity transformation and vertex/triangle reordering attacks, a principal component analysis is used to solve this issue. Figure 1 shows a flowchart of the proposed algorithm. The following sections describe the proposed algorithm.

2.1. Data Embedding Procedure. This section details the data embedding procedure. The procedure begins by preprocessing the topological information of the input triangular meshes. The model information can be also derived in this process. Each vertex and each triangle are then sorted with the help of the normal vector and the principal axis. The data embedding process generates three barycentric weights and two of them are with the secret message embedded. The final triangle subdivision process takes above data-embedded weights as input to produce a newly added vertex and the triangle is subdivided into three smaller ones. The embedding capacity can be further improved significantly by recursive subdivision. Finally, the stego model is generated after processing each triangle. For simplicity, this study illustrates the proposed technique using 2D triangles. However, this can be easily extended to a 3D space.

2.1.1. Preprocessing Process. The relative information about the original triangular meshes is obtained in this process. Assume the triangular meshes consist of N_V vertices and N_T triangles. A principal component analysis is performed

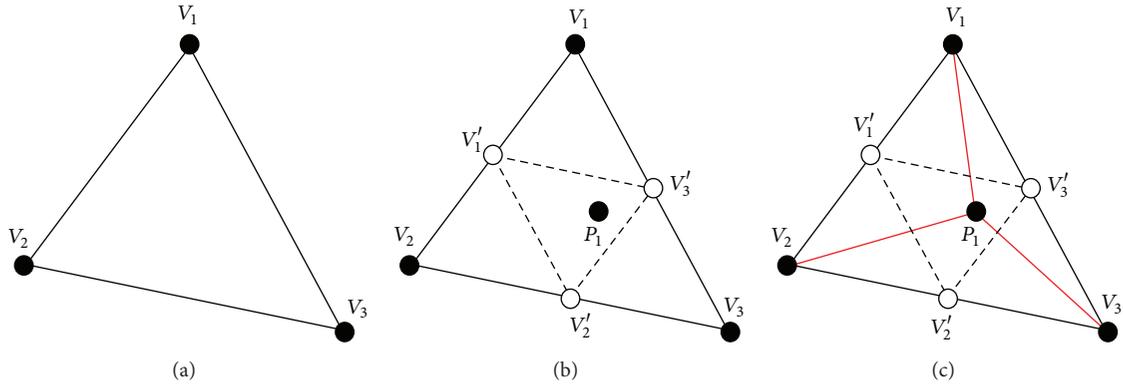


FIGURE 2: One example for data embedding process.

on the triangular meshes and one PCA coordinates system is generated. Obviously, the new coordinates system has a new origin, PCA_O , which is a gravity center of the triangular meshes, and three principal axes, called PCA_X , PCA_Y , and PCA_Z . Each vertex V_i is sorted according to the angle between the principal axis P_Z and the vector $\overrightarrow{V_i PCA_O}$. Each triangle T_i is also sorted according to the angle between the principal axis P_Z and the vector $\overrightarrow{C_{T_i} PCA_O}$, where C_{T_i} is the gravity center of the triangle T_i . The sorted result is the processing order for data embedding.

2.1.2. Data Embedding Process. According to the processing order for each triangle, the data embedding process is initialized. Assume a triangle T (see Figure 2(a)) is constructed by three vertices, V_1 , V_2 , and V_3 and two series of B -bit secret message B_1 and B_2 are prepared for embedding. Three weights can be generated using (1) and the sum of them equals one, where $SM_{10}^{B_i}$ is the decimal value converted from B -bit binary secret message. For avoiding the unbalanced subdivision for each triangle, three temporary vertices V'_1 , V'_2 , and V'_3 are first generated by (2) (see Figure 2(b)) and then sorted according to the angle between the principal axis P_Z and the vector $\overrightarrow{V'_i PCA_O}$. Assume that VS_1 , VS_2 , and VS_3 are the sorted results. A new point P_1 , within the located triangle T , is then obtained based on the concept of barycentric coordinates using (3). The vertex P_1 is now with $2B$ -bit secret message embedded (see Figure 2(b)). Consider

$$\begin{aligned} \alpha_1 &= \frac{SM_{10}^{B_1}}{2^B}, \\ \alpha_2 &= (1 - \alpha_1) \times \frac{SM_{10}^{B_2}}{2^B}, \\ \alpha_3 &= 1 - \alpha_1 - \alpha_2, \\ V'_{1r} &= \frac{(V_{1r} + V_{2r})}{2}, \\ V'_{2r} &= \frac{(V_{2r} + V_{3r})}{2}, \end{aligned} \quad (1)$$

$$V'_{3r} = \frac{(V_{3r} + V_{1r})}{2}, \quad \text{where } r = x, y, z, \quad (2)$$

$$P = \alpha_1 VS_1 + \alpha_2 VS_2 + \alpha_3 VS_3. \quad (3)$$

2.1.3. Triangle Subdivision Process. After the new point P_1 is generated, we then construct its topology with original vertices of located triangle T . The triangle $T_{V_1 V_2 V_3}$ is subdivided into three smaller triangles $T_{P_1 V_1 V_2}$, $T_{P_1 V_1 V_3}$, and $T_{P_1 V_2 V_3}$ (see Figure 2(c)). The first iteration for data embedding is now finished.

Following above rules, we can subdivide each smaller triangle again to raise the embedding capacity. However, considering the number precision of the stego model, the triangle cannot be subdivided infinitely. Therefore, some criteria can be adopted for terminating the recursive triangle subdivision, such as total embedding capacity, the area of subdivided triangle, or the number of subdivision iteration. In this study, we adopt the ratio R between the average triangle area AA of the triangular meshes and the area of subdivided triangle A_{ST} as the termination criterion. The triangle subdivision process is continued only when the area of subdivided triangle A_{ST} is larger than R times of the average area ($A_{ST} \geq R \times AA$).

First, we sort three subdivided triangles and process each triangle sequentially. For example, in Figure 3(a), the triangle $T_{P_1 V_1 V_2}$ is assumed to be the next processing triangle. We then pick one additional point P_2 again using (1), (2), and (3) and then construct its topology with three vertices of located triangle. Three smaller triangles are then generated. The recursive triangle subdivision process is continued until the termination condition is reached. For example, in Figure 3(b), the triangle $T_{P_1 P_2 V_1}$ cannot be subdivided because its surface area termination condition is assumed to be reached. Otherwise, an insufficient number precision for representing the vertex coordinates of stego model may result in the failure of data extraction. Figure 3(c) shows the example subdivided result for the triangle $T_{P_1 P_2 V_2}$. After processing the residual

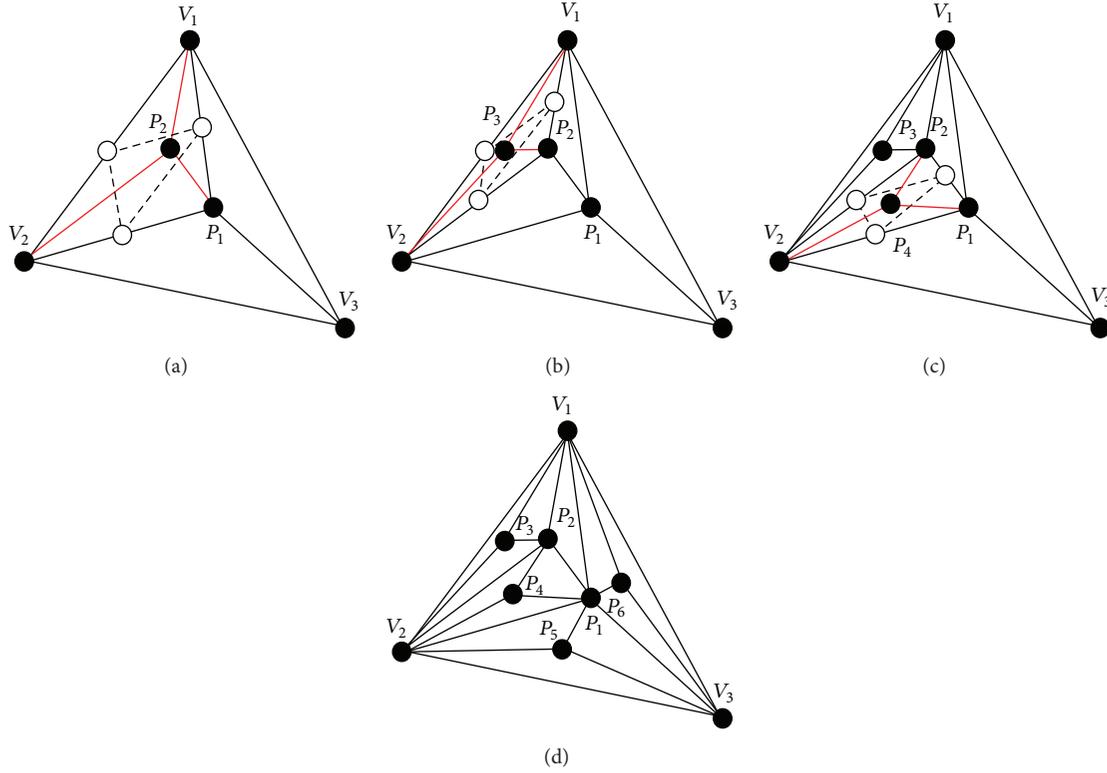


FIGURE 3: One example for triangle subdivision process.

triangles $T_{P_1V_2V_3}$ and $T_{P_1V_1V_3}$ recursively, one example stego model is shown in Figure 3(d).

2.2. Data Extraction Procedure. After the receiver obtained the stego model, the data extraction procedure is initialized to extract the secret message following the below steps. Because each added vertex is derived from the barycentric weights of planar triangle, the distance between the vertex and its located triangle equals 0. Therefore, we first remove all the added vertices to reconstruct the original triangular meshes and a principal component analysis is then performed to obtain the original PCA coordinates system. Thus, the data extraction order for each triangle T_i can be obtained with the help of the angle between the principal axis P_Z and the vector $\overrightarrow{C_{T_i}PCA_O}$.

Next, we adopt an iterative process to extract the secret message. It is obvious that the number of neighboring vertices for each added vertex is 3. Therefore, each iteration consists of four steps, including finding the added vertex, calculating its barycentric weights, extracting the secret message, and recovering the triangle. Above steps iteratively continue until all added vertices are removed. For example, in Figure 4(a), vertices $P_3, P_5,$ and P_6 are found in the first iteration because of its three neighboring vertices. Take vertex P_3 , for example, we can calculate its three barycentric weights. However, only two of them are with the secret message embedded. By checking the processing order of vertices $V_1, V_2,$ and P_2 , fortunately, we can obtain first two barycentric weights to extract secret message. Equation (4) shows the barycentric

weight calculation formula, where $VS_1, VS_2,$ and VS_3 are the sorted results of three vertices of the processing triangle and P' is the processing vertex for data extraction. Equation (5) is used to extract two series of B -bit secret message. Because the secret message is embedded according to the angle between the normal of the triangle and principal axis, we then first recover three smaller triangles into a bigger one by removing the vertex and its neighboring relationship. Two series of B -bit secret message are then attached to the recovered angle. The first iteration is finished and current triangular meshes are shown in Figure 4(b). In the next iteration, vertices P_2 and P_4 are processed, whereas two triangles are recovered (see Figure 4(c)). Finally, vertex P_1 is processed (see Figure 4(d)) and the original triangle is recovered. The data extraction procedure is finished. Consider

$$\alpha_1 = \frac{\Delta P'VS_2VS_3}{\Delta VS_1VS_2VS_3},$$

$$\alpha_2 = \frac{\Delta P'VS_1VS_3}{\Delta VS_1VS_2VS_3}, \quad (4)$$

$$\alpha_3 = \frac{\Delta P'VS_1VS_2}{\Delta VS_1VS_2VS_3},$$

$$SM_{10}^{B_1} = \lfloor \alpha_1 \times 2^B \rfloor,$$

$$SM_{10}^{B_2} = \left\lfloor \frac{(\alpha_2 \times 2^B)}{(1 - \alpha_1)} \right\rfloor. \quad (5)$$

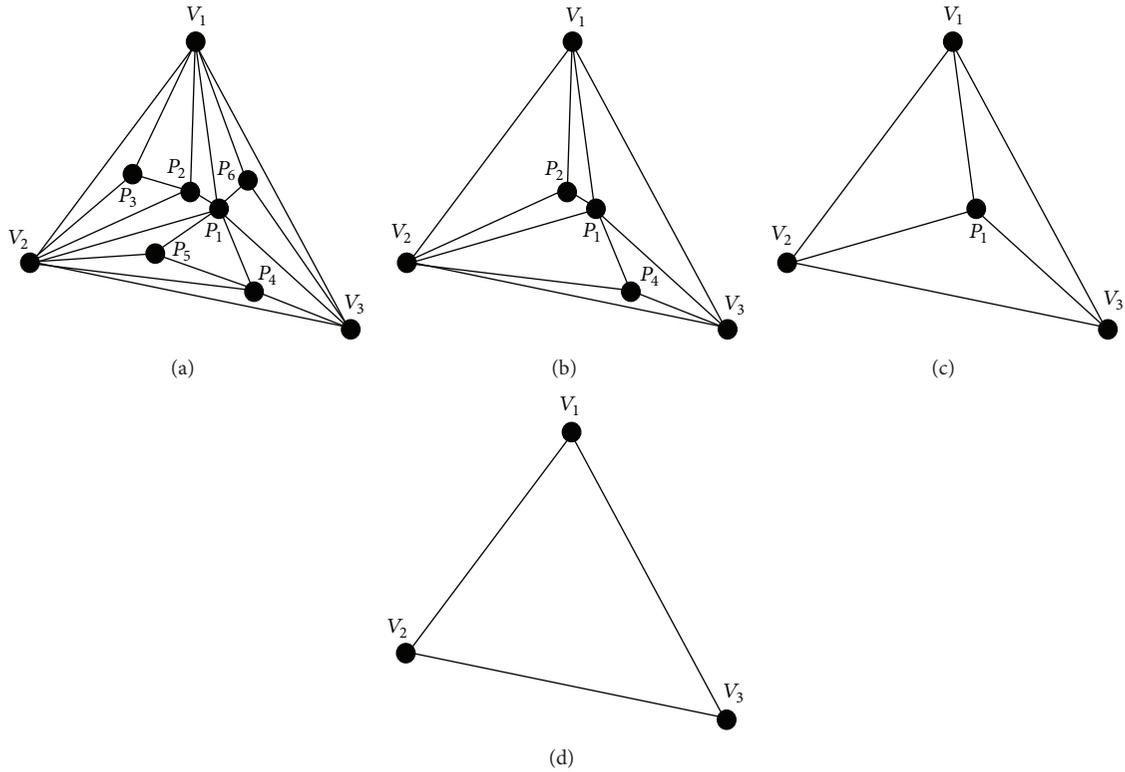


FIGURE 4: One example for data extraction process.

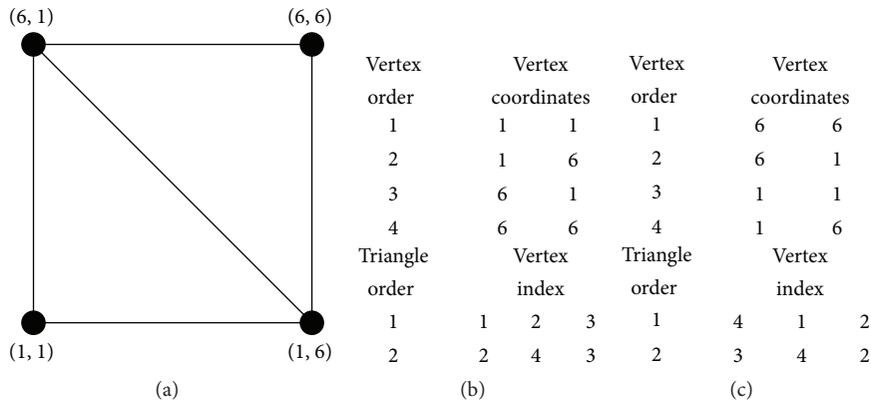


FIGURE 5: Examples of vertex reordering attacks.

2.3. Robustness Assessment. This section discusses the robustness of the proposed method. In this study, a principal component analysis is adopted to determine the processing order for each vertex and each triangle. In the beginning of the data extraction procedure, each added vertex can be removed because it is located on the triangle constructed by its neighboring vertices. The original cover model can be reconstructed and the same PCA coordinates system is obtained. Because the PCA coordinates system is robust against the rotation and translation operations, our stego model can resist above two attacks. Further, the barycentric weights are invariant before and after scaling operations; the same weights can be obtained.

Reordering attacks [3] modify the vertex/triangle representation orders in the model file but do not affect the appearance of 3D models. Figure 5 gives a 2D example of reordering attacks both on vertices and on triangles. The model in Figure 5(a) consists of four vertices and two triangles. Two examples for representing the model are shown in Figures 5(b) and 5(c). One of them can be the reordering attack of the other one. Although their vertex and triangle orders are different, the visual effects are the same. When performing our proposed data embedding and extraction procedure, each vertex and each triangle are sorted according to the angle between the normal vector and principal axis. The reconstructed extraction order is the same as the embedding

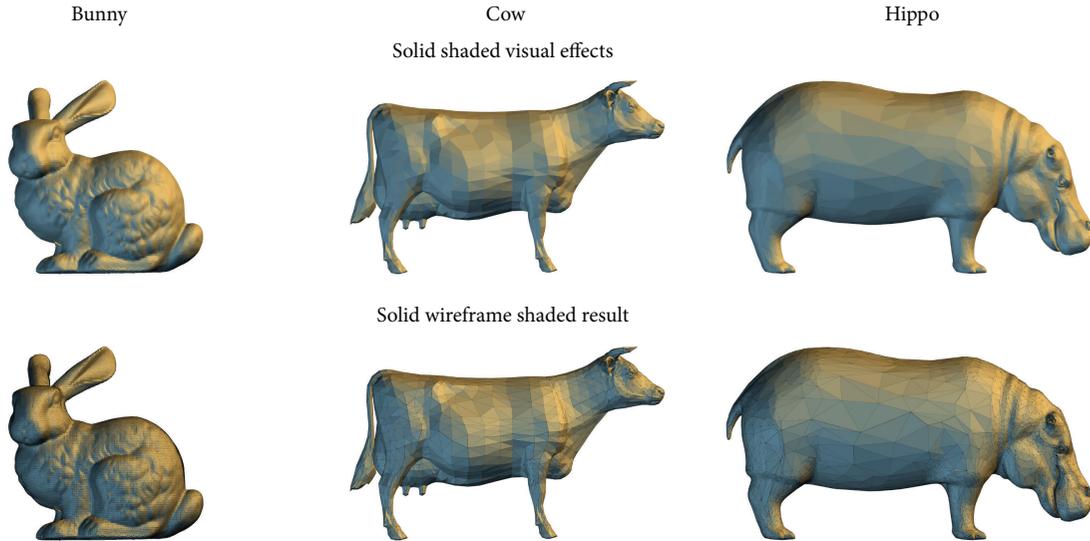


FIGURE 6: The visual effects for three triangular meshes.

order. Thus, our stego model can be robust against reordering attacks.

However, unlike watermarking algorithms with higher robustness against some intentional attacks, the proposed data embedding algorithm can provide more embedding capacity but with less robustness. Noise attacks cause some random disturbances on the vertex coordinates of 3D models. Even a noise attack for each vertex with small intensity can affect the integrity of extracted secret message. Some vertices with secret message embedded may be classified as ones without message embedded. However, if only part of vertices in the stego model is under the noise attack, we still have a chance to extract partial secret message. Finally, lossy compression attacks quantize the vertex coordinates, introduce new vertices, remove original vertices, or modify the topological relationship between vertices. The attacks seriously affect the generation of principal axis and the extraction order for each vertex is different from its embedding order. The extraction of the secret message from the attacked model fails.

3. Experimental Results

This section presents the experimental results obtained from three 3D triangular meshes, called “bunny,” “cow,” and “hippo.” Table 1 shows the mesh information, including the number of vertices, N_V , the number of triangles, N_T , and model size (represented by the diagonal length DL_{BV} of the bounding volume), of each mesh. We also show the file size for each test model, stored in PLY file format, for later comparing the one of each stego model. Microsoft Visual C++ programming language was used to implement the proposed algorithm on a personal computer with an Intel® Core™ i7-6700K processor and 16 GB memory. Figure 6 shows the solid and solid wireframe shaded results for each cover triangular mesh. The embedded secret message was a

TABLE 1: Information about three triangular meshes.

Model name	N_V	N_T	DL_{BV}	File size
Bunny	34834	69541	25.02	2366 KB
Cow	2903	5804	12.71	179 KB
Hippo	23105	46202	60.00	1572 KB

randomly generated 0/1 bit string. The parameter B is set as 10.

This section first presents the visual effects for three stego models. We also show the close-up views for each model for a better comparison. Second, this section also presents the embedding capacity for three triangular meshes with different embedding thresholds R . Thereafter, the triangle area distribution for each mesh is shown to know the position of the embedding threshold. This section next presents the robustness assessment for our stego models. Finally, a comparison of this algorithm with current outstanding non-reversible/reversible algorithms demonstrates the feasibility of the proposed method.

Figure 7 shows the visual effects for three triangular meshes with secret message embedded. From Figures 6 and 7, the cover and stego models with different number of vertices and triangles can have the same rendering results. The reason is that the entire data embedding procedure makes all the added vertices and triangles being located on the original triangle. No influence is caused for visual effects. Figure 8 shows the close-up views of different meshes with different embedding thresholds, enabling a better comparison. Obviously, a close-up view for the wireframe shaded result of each model is quite different especially for small embedding threshold.

Table 2 shows the embedding capacity, number of processed triangles, and final number of vertices/triangles for

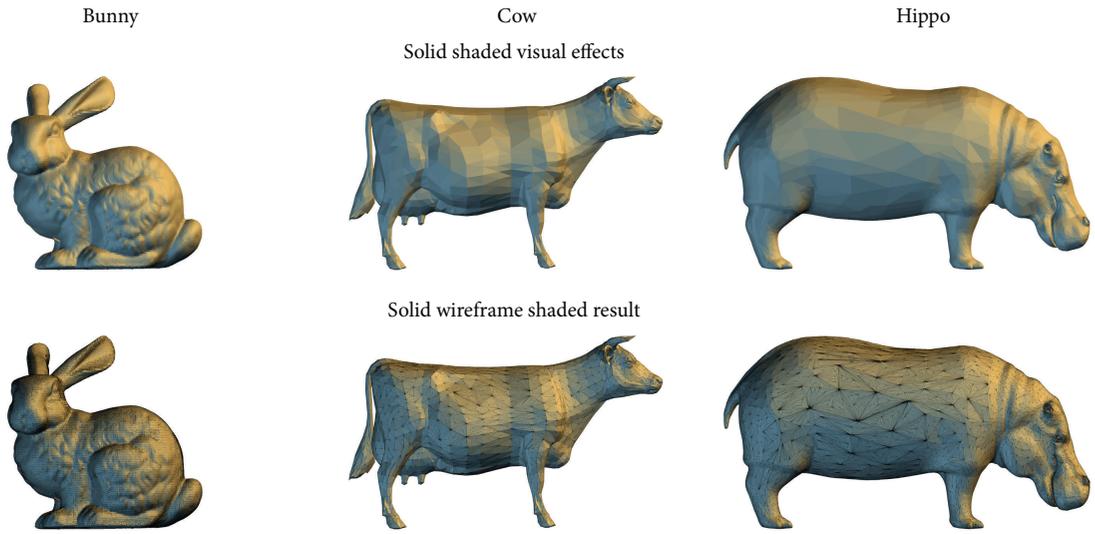


FIGURE 7: The visual effects for three triangular meshes with secret message embedded.

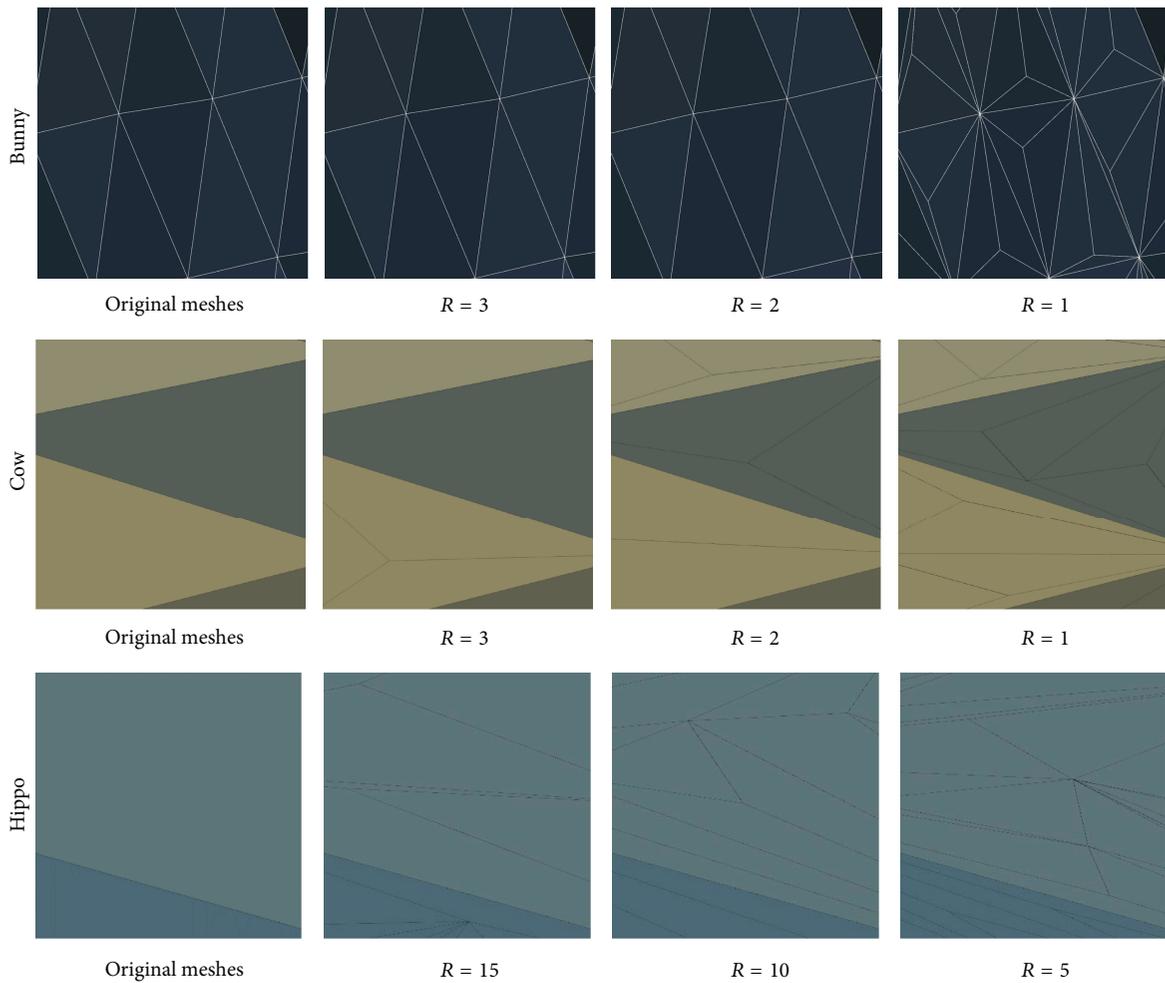


FIGURE 8: The close-up views for three triangular meshes with secret message embedded.

TABLE 2: The embedding capacity, number of processed triangles, and the number of vertices and triangles in the stego model.

Model name	Bunny			Cow			Hippo		
AA	0.008226			0.018753			0.065556		
R	1	2	3	1	2	3	5	10	15
Embedding capacity	525180	12760	1460	64820	19500	7960	88880	34680	19660
Number of processed triangles	25706	604	72	1842	753	342	1340	677	456
Vertex number, N'_V	61093	35472	34907	6144	3878	3301	27549	24839	24088
Triangle number, N'_T	121969	70727	69597	12286	7754	6600	55090	49670	48168
File size	4179 KB	2410 KB	2371 KB	383 KB	241 KB	204 KB	1882 KB	1693 KB	1641 KB

each test model. With the increase of the embedding threshold, the number of triangles which can be processed is less and the embedding capacity is decreased. Because each added vertex can be with $2B$ -bit secret message embedded, the total embedding capacity equals $2B \times (N'_V - N_V)$. Further, the file size of each stego model is increased because of storing the coordinate values of newly added points and their topological relationship. Figure 9 shows the histogram of the area distribution of each triangle for different models. We also indicate the position of the embedding threshold R to show the located region used for data embedding. For each example in Figure 9, each triangle located at the gray region cannot be with message embedded because the embedding criterion is not reached. With the increasing of the embedding threshold, the number of embeddable triangles decreases and less secret message can be embedded. Only the triangle with an area located at the blue region can perform data embedding when the embedding threshold R is set as 3 for the bunny/cow models and is set as 15 for the hippo model. More triangles can be processed with message embedded with smaller threshold. For example, when the embedding threshold R is set as 2 for the bunny/cow models and is set as 10 for the hippo model, the triangles with an area located within the green region can be additionally used for data embedding.

This section next presents the robustness assessment for our proposed algorithm. As mentioned above, each added vertex is derived from the barycentric weights of planar triangle; the distance between the vertex and its located triangle equals 0. Therefore, we can reconstruct the original model by removing all the added vertices. The same PCA coordinates system is then obtained and the extraction order for each vertex is determined with no error. Further, to test the robustness of our scheme, we randomly rotate, translate, and uniformly scale each 3D stego model. Figure 10 shows the visual effects after similarity transformation attacks. The experimental results show that no error is in the extracted secret message. However, the number precision of the stego model must be increased properly under downscaling attack. For reordering attacks, vertices and triangles in the model file were reordered randomly for several times according to the scenario illustrated in Section 2.3. The embedded secret message can be extracted without any error. That is, the proposed scheme is robust against reordering attacks. For the experiment of our proposed method under noise attacks, we add random noise within the interval $[-\sigma, \sigma]$ to all the three coordinates of p percentage of vertices in

TABLE 3: The ratio of the vertices for correct message extraction after noise attacks with different parameters.

σ	p				
	10	20	30	40	50
1.0×10^{-4}	46.84%	19.73%	7.42%	2.70%	0.91%
5.0×10^{-5}	49.85%	23.04%	10.04%	4.13%	1.81%
1.0×10^{-5}	69.19%	48.49%	33.31%	22.75%	15.72%
5.0×10^{-6}	84.48%	71.96%	60.12%	48.98%	41.00%
1.0×10^{-6}	99.65%	99.42%	99.22%	99.05%	98.89%

the model and record the ratio R of the vertices for correct message extraction, where σ is the amplitude of noise. Table 3 shows the experimental results of bunny model under above scenarios. Other test models have similar results. As the value of parameter p increases, the ratio of the vertices for correct message extraction is apparently lower. Further, when the value of parameter σ is larger, indicating that each vertex can be with larger displacement for its coordinates, the ratio of the vertices for correct message extraction is also lower. As a result, the proposed algorithm has weak resistance to noise attacks.

Finally, this section compares the proposed algorithm with other existing outstanding algorithms in terms of embedding capacity, robustness, processed domain, and reversibility (Table 4). From the aspect of embedding capacity, Huang et al. introduced the permutation steganography for data embedding; each vertex can be with $\log_2 n!/n$ -bit message embedded in the representation domain. However, the original orders for each vertex and triangle cannot be recovered after the message is embedded. Huang and Tsai proposed a reversible data hiding scheme for 3D polygonal models based on histogram shifting. The embedding capacity is unsatisfactorily only from 0.09 to 0.51 bit per vertex. Ji et al. adopted the difference expansion to achieve reversible data embedding with the embedding capacity of 1.01 to 1.09 bit per vertex. Our proposed algorithm derives the newly added points for data embedding; each can be with $2B$ -bit secret message embedded. From the aspect of robustness, four algorithms are all robust against similarity transformation (ST) attacks. Huang et al.'s proposed algorithm can withstand similarity transformation attacks since the representation order of vertices and polygons is not modified after translation, scaling, and rotation attacks. The distance ratio in Ji et al.'s proposed method is invariant after the

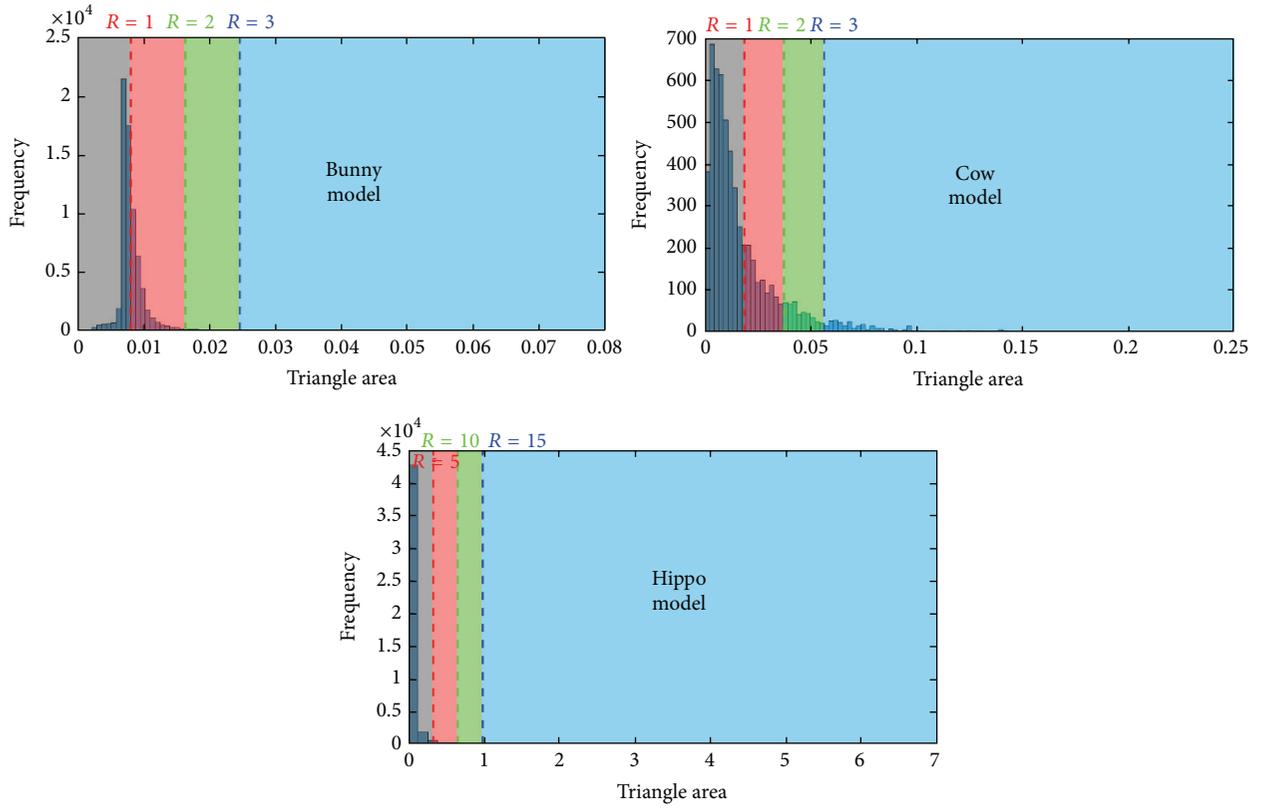


FIGURE 9: The histogram of the area distribution of each triangle for different models and the position of the embedding threshold R .

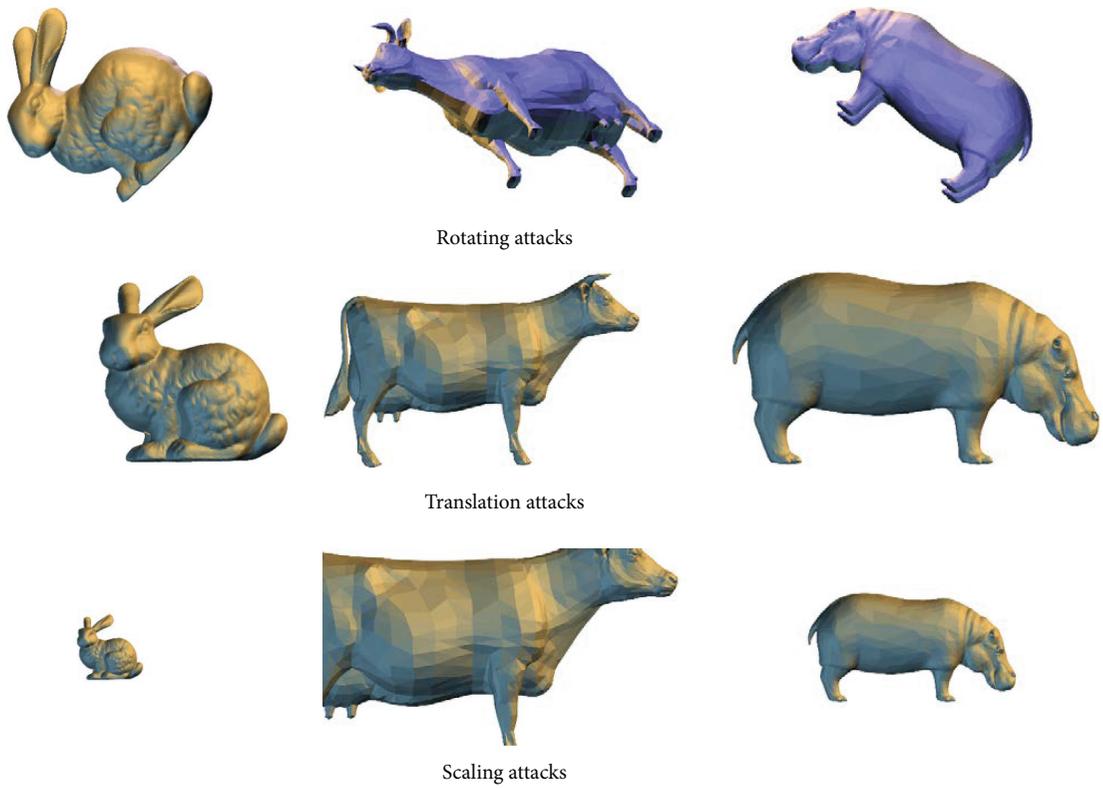


FIGURE 10: The visual effects of stego models after similarity transformation attacks.

TABLE 4: A comparison of the methods of Huang et al. [16], Huang and Tsai [3], and Ji et al. [5] and the current method.

Algorithm	[16]	[3]	[5]	Our approach
Capacity (bpv)	$\approx \log_2 n!/n$	0.09~0.51	1.01~1.09	2B
Robustness	ST	ST, reordering	ST, noising	ST, reordering
Domain	Representation	Spatial	Spatial	Spatial
Reversibility	No	Yes	Yes	Yes

similarity transformation attacks. Noising attack can be also resisted in their proposed algorithm. The remaining two algorithms employ principal component analysis to make their algorithms robust against similarity transformation attacks. Our proposed algorithm can additionally be robust against reordering attack, because each processing vertex and each triangle are determined by the principal axis.

4. Conclusions and Future Work

This study proposes a novel reversible data hiding algorithm for 3D triangular meshes based on recursive subdivision. The proposed algorithm can provide reversibility, high embedding capacity, no visual distortion, and the greater robustness. For each triangle, a threshold is employed to recursively subdivide the triangle and the secret message is embedded into the added vertex. A principal component analysis can help the stego model against similarity transformation and vertex reordering attacks. The experimental results in this study demonstrate the feasibility of the proposed technique for reversible data hiding.

However, in the future, we hope to integrate the concept of subdivision surface to generate high-quality smooth meshes. Further, a novel systematic way to determine the appropriate threshold R is also an interesting problem deserving investigation. Finally, improvements on raising the robustness against cropping or noising attacks may be considered.

Competing Interests

The author declares that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work was supported by Ministry of Science and Technology of Taiwan under Grant nos. MOST 103-2221-E-468-021 and MOST 104-2218-E-468-001.

References

- [1] C.-H. Chuang, C.-W. Cheng, and Z.-Y. Yen, "Reversible data hiding with affine invariance for 3D models," in *Proceedings of the IET International Conference on Frontier Computing. Theory, Technologies and Applications*, pp. 77–81, Taichung, Taiwan, August 2010.
- [2] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, Morgan Kaufmann, Burlington, Mass, USA, 2nd edition, 2008.
- [3] Y.-H. Huang and Y.-Y. Tsai, "A reversible data hiding scheme for 3D polygonal models based on histogram shifting with high embedding capacity," *3D Research*, vol. 6, no. 2, article 20, pp. 1–12, 2015.
- [4] C.-Y. Jhou, J.-S. Pan, and D. Chou, "Reversible data hiding base on histogram shift for 3D vertex," in *Proceedings of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '07)*, vol. 1, pp. 365–370, IEEE, Kaohsiung, Taiwan, November 2007.
- [5] H. Ji, X. Yang, C. Zhang, and X. Gao, "A new reversible watermarking of 3D models based on ratio expansion," in *Proceedings of the 3rd International Congress on Image and Signal Processing (CISP '10)*, pp. 3899–3903, Yantai, China, October 2010.
- [6] K.-H. Jung and K.-Y. Yoo, "Data hiding method using image interpolation," *Computer Standards & Interfaces*, vol. 31, no. 2, pp. 465–470, 2009.
- [7] C.-F. Lee and Y.-L. Huang, "An efficient image interpolation increasing payload in reversible data hiding," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6712–6719, 2012.
- [8] Y.-Y. Tsai, J.-T. Chen, Y.-C. Kuo, and C.-S. Chan, "A generalized image interpolation-based reversible data hiding scheme with high embedding capacity and image quality," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 9, pp. 3286–3301, 2014.
- [9] H.-T. Wu and J.-L. Dugelay, "Reversible watermarking of 3D mesh models by prediction-error expansion," in *Proceedings of the IEEE 10th Workshop on Multimedia Signal Processing (MMSP '08)*, pp. 797–802, Cairns, Australia, October 2008.
- [10] D. Wu and G. Wang, "A reversible watermarking scheme for 3D meshes," in *Active Media Technology: 5th International Conference, AMT 2009, Beijing, China, October 22–24, 2009. Proceedings*, vol. 5820 of *Lecture Notes in Computer Science*, pp. 513–521, Springer, Berlin, Germany, 2009.
- [11] C.-Y. Yang, "Reversible data hiding using two marked images based on adaptive coefficient-shifting algorithm," *Advances in Multimedia*, vol. 2012, Article ID 473896, 9 pages, 2012.
- [12] A. Zhu, C. Zhang, X. Yang, and X. Gao, "An efficient 3D information hiding algorithm based on sampling concepts," in *Proceedings of the 3rd International Congress on Image and Signal Processing (CISP '10)*, vol. 3, pp. 1171–1175, Yantai, China, October 2010.
- [13] X. Feng, "A novel watermarking for 3D subdivision mesh," *International Journal of Advancements in Computing Technology*, vol. 4, no. 15, pp. 66–73, 2012.
- [14] G. Lavoué, F. Denis, and F. Dupont, "Subdivision surface watermarking," *Computers and Graphics*, vol. 31, no. 3, pp. 480–492, 2007.
- [15] A. C. Rencher, *Methods of Multivariate Analysis*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2002.
- [16] N.-C. Huang, M.-T. Li, and C.-M. Wang, "Toward optimal embedding capacity for permutation steganography," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 802–805, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

