An Efficient and Distortion-controllable Information Hiding Algorithm for 3D Polygonal Models with Adaptation

Yuan-Yu Tsai, Wen-Ching Huang, and Bo-Feng Peng
(Corresponding author: Yuan-Yu Tsai)

Department of Applied Informatics and Multimedia, Asia University,
No. 500, Lioufeng Rd., Wufeng Dist., Taichung City 41354, Taiwan
(Email: yytsai@asia.edu.tw)

(Received Apr. 18, 2013; revised and accepted Oct. 6, 2013)

Abstract

We present an efficient information hiding algorithm for polygonal models. The decision to referencing neighbors for each embeddable vertex is based on a modified breadth first search, starting from the initial polygon determining by principal component analysis. The surface complexity is then estimated by the distance between the embedding vertex and the center of its referencing neighbors. Different amounts of secret messages are adaptively embedded according to the surface properties of each vertex. A constant threshold is employed to control the maximum embedding capacity for each vertex and decrease the model distortion simultaneously. The experimental results show the proposed algorithm is efficient and can provide higher robustness, higher embedding capacity, and lower model distortion than previous work, with acceptable estimation accuracy. The proposed technique is feasible in 3D adaptive information hiding.

Keywords: Adaptation, breadth first search, controllable distortion, information hiding, polygonal models

1 Introduction

3D information hiding algorithms [1, 5, 9] hide the secret message in a cover model to produce a stego model which is undetectable to all expect the legitimate receiver. According to different operation domains for the cover models, different embedding manners are used for data embedding. Generally, the algorithms in the spatial domain [6] are efficient and suitable for the covert communication; whereas the algorithms in the transform domains [10] are of higher robustness and appropriate for copyright protection.

Adaptive information hiding algorithms [2, 3, 8] embed different amounts of secret message into the embedding vertex according to its surface complexity. The main reason is that a vertex located on a rough surface can generally tolerate more positional changes than one on a smooth surface. Thus, the majority of the secret message is embedded into rougher regions to avoid causing visible model distortion on smoother regions.

However, only two information hiding algorithms consider adaptation for the purpose of covert communication. Cheng and Wang [3] proposed an adaptive algorithm utilizing the correlation between neighboring polygons to estimate the amount of message for the embedding vertex. The algorithm first employs a contagious diffusion scheme to efficiently traverse each mesh. Thereafter, an adaptive minimum-distortion estimation procedure is employed to embed the secret message into extending, sliding, and rotating levels of the embedding vertex. However, their proposed algorithm only uses two of the other vertices within the same polygon containing the embedding vertex. This may lead to inaccurate estimation results. To raise the estimation accuracy for surface complexity, Tsai [8] introduced a vertex decimation process to determine the referencing neighbors for each embedding vertex. A quantization index modulation concept is then employed to embed different amounts of secret message into each embedding vertex according its surface complexity. Although the estimation accuracy can be significantly raised from 34.11% to 65.51%, on average, the time complexity for the vertex decimation process is higher and the performance of the algorithm is seriously affected.

In this study, a modified breadth first search (BFS) scheme is developed to improve the performance of determining the referencing neighbors for each embedding vertex. The proposed algorithm can efficiently resolve the referencing neighbors but slightly decrease the estimation accuracy. The proposed algorithm introduces a constant threshold *CT* to control the maximum embedding capacity of each vertex and to lower model distortion. Finally, the proposed algorithm can be robust against similarity transformation and vertex reordering attacks.

This rest of this study is organized as follows; Section 2 illustrates the proposed scheme; Section 3 presents a discussion of the experimental results; and finally, Section 4 offers a conclusion of this work.

2 The Proposed Algorithm

This section illustrates the proposed algorithm, including the data embedding and data extraction procedure. The flowchart of the proposed algorithm is shown in Figure 1.

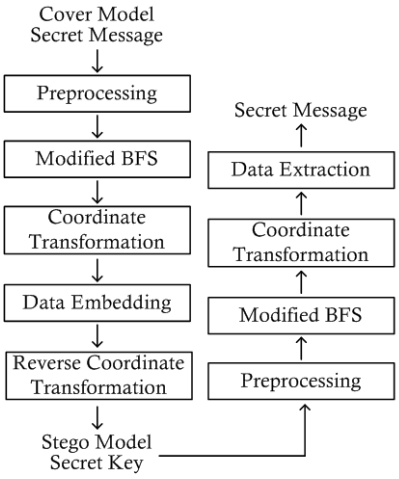

Figure 1: The flowchart of the proposed algorithm

2.1 The Data Embedding Procedure

The data embedding procedure begins by preprocessing the topological information of the input model. To efficiently derive the referencing neighbors for each embedding vertex in the modified BFS process, a vertex neighboring table records the indices of the actual neighbors based on the polygonal information appearing in the model file. Furthermore, the model information, including the diagonal length of the bounding volume, the number of vertices, and the number of faces, can be also derived in this process.

The breadth first search is a search method that begins from the root node of a graph and explores all its neighboring nodes. For each neighboring node, the algorithm explores unexplored neighboring nodes iteratively. The algorithm stops after all nodes have been traversed. All neighboring nodes derived by expanding a node are added to a queue with a first-in-first-out (FIFO) property. However, the breadth first search may not only have a unique search result for a graph. The search results depend on how the user chooses the neighboring nodes of each node. In the proposed technique, the search order is based on the sequence appearing in the vertex neighboring table for each corresponding vertex. This vertex neighboring table is robust against similarity transformation and vertex ordering attacks. Thus, the same search order can be derived in the data extraction procedure.

The modified BFS process determines the referencing neighbors for complexity estimation and data embedding. Each vertex can have three different statuses "NS," "SS," and "S," representing the corresponding vertex as being unsearched, semi-searched, and searched. Each vertex is initialized as "NS." The initial vertex for performing this process is resolved by the first index of the polygon intersected by the principal axis and regarded as the root node for performing the BFS algorithm. In the first iteration, the process enqueues the initial vertex, dequeues it, and then enqueues its unsearched neighbors. Note that, the order of the enqueued neighbors is not arbitrary, but based on the sequence appearing in the vertex neighboring table for each corresponding vertex. When the vertex is enqueued, its status is modified to "SS," whereas the status is modified to "S" when the corresponding vertex is dequeued. In the second iteration, the process dequeues one vertex from the queue and euqueues its unsearched neighbors. The order of the enqueued neighbors is still based on the sequence appearing in the vertex neighboring table. This iteration is repeated until the queue is empty. The algorithm then proceeds until all vertices are searched. In each iteration, the neighbors with the status "NS" or "SS" are included in the referencing list for each dequeued vertex.

To be robust against similarity transformation attacks, the coordinate transformation process collects the vertex whose referencing list is empty. Thereafter, principal component analysis [7] is performed again on these vertices to produce a coordinate system. All vertices of the 3D cover model are then transformed from the Cartesian coordinate system to the new one. Because the coordinate values of these vertices are never modified, the constructed coordinate system can be used for model registration under rotation and translation attacks. This process also derives the diagonal length *DL* of the bounding volume of the vertex whose referencing list is empty. This diagonal length is the key point making the proposed algorithm robust against scaling attacks.

For each embeddable vertex, the data embedding process acquires the referencing neighbors from the referencing list. Next, the embedding capacity is proportional to the distance *d* between the embedding vertex and the center of its referencing neighbors [8]. The quantization index modulation concept is then employed for data embedding with the embedding threshold *ET*. To control the maximum embedding capacity for each vertex and lower model distortion, a constant threshold σ_{EC} is introduced in Equation (1). Thus, the maximum distortion for each embedding capacity can be reduced from $2^{EC_1} \times ET$ to $2^{\sigma_{EC}} \times ET$. Equation (2) shows the derivation for the data-embedded distance *d'*. When the calculated embedding capacity EC_1 is smaller than σ_{EC}, secret message SM_2 with length EC_1 is extracted from the stream of secret message with binary form and SM_{10} is the decimal value from a binary-to-decimal format transformation of SM_2. Figure 2 shows a graphical representation of this type of data embedding. However, for a calculated embedding capacity EC_1 larger than σ_{EC}, the final embedding capacity is limited to $EC_2 = \sigma_{EC}$ (see Equation (2)). Figure 3 is a graphical representation of the data embedding process with limited embedding capacity. To lower model distortion, we first calculate the proper embedding interval

Table 1: The model information, the embedding capacity, and the model distortion for our proposed algorithn

Model Name	N_V	N_F	DL_{BV}	DL_{V^N}	N_{V^N}	Capacity	BitsPV	BitsPEV	NHD
Armadillo	172974	345944	228.80	226.51	440	2784917	16.10	16.14	0.1142%
Brain	294012	588032	10.03	9.84	572	3494908	11.89	11.91	0.1789%
Cow	46433	92864	30.49	29.04	49	644982	13.89	13.91	0.1075%
Golfball	122882	245760	1.73	1.73	1219	1181227	9.61	9.71	0.0543%
Lucy	262909	525814	1918.29	1913.97	4050	4858167	18.48	18.77	0.1608%
Maxplanck	49132	98260	697.49	624.28	55	922368	18.77	18.79	0.1490%
Dragon	437645	871414	26.69	26.62	10422	5321934	12.16	12.46	0.1260%
Hand	327323	654666	8.41	8.28	1555	3501159	10.70	10.75	0.0789%

m based on the σ_{EC} and SM_{10} is finally embedded into the distance between the embedding vertex and the center of its referencing neighbors. Obviously, SM_{10} is a decimal value from a binary-to-decimal format transformation of SM_2 with length σ_{EC}.

$$EC = \begin{cases} EC_1 = \log_2\left(d/ET\right) \\ EC_2 = \min\left(\left\lfloor \log_2\left(d/ET\right)\right\rfloor, \sigma_{EC}\right) \end{cases} \quad (1)$$

$$d' = \begin{cases} \left(2^{EC_1} + \mathrm{SM}_{10}\right) \times ET \\ \left(\left\lfloor \left(d/ET - 2^{EC_1}\right) / 2^{\sigma_{EC}} \right\rfloor \times 2^{\sigma_{EC}} + 2^{EC_1} + \mathrm{SM}_{10}\right) \times ET \end{cases} \quad (2)$$

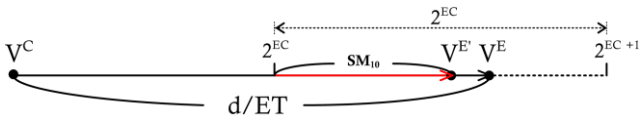

Figure 2: A graphical representation of the data embedding

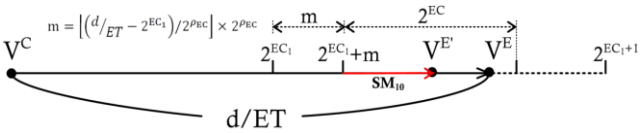

Figure 3: A graphical representation of the data embedding

After the data-embedding process is complete, the algorithm transforms all the vertices from the PCA coordinate system back to the Cartesian system. This produces a stego model that is then delivered to the receiver. To ensure robustness against scaling attacks, a secret key is then calculated by dividing the embedding threshold ET by the diagonal length DL of the bounding volume of the vertices with no referencing neighbors.

2.2 The Data Extraction Procedure

During the data extraction procedure, the following processes are performed sequentially. First, the preprocessing process deals with the topological information of the input model. A vertex neighboring table is then constructed. As mentioned before, the construction of this table is based on the topological property of each vertex. However, the topological information is never modified in the data embedding procedure. Therefore, this table can be correctly reconstructed according to the same processes in the data embedding procedure. Second, the modified BFS process iteratively determines the referencing neighbors for each embedding vertex of the stego model. Third, we collect all the vertices whose referencing list is an empty set. Thereafter, a principal component analysis is performed on the above vertices to produce a coordinate system and transform all vertices of the stego model from the Cartesian coordinate system to the new one. Fourth, the embedding threshold can be also derived in this process based on the secret key and the diagonal length DL of the bounding volume of the vertices with no referencing neighbors. Finally, the data-embedded distance d' can be calculated. The secret message with a binary format can be easily derived based on the embedding capacity and decimal-to-binary format transformation of SM_{10} derived from Equation (3).

$$SM_{10} = \begin{cases} \left\lfloor d'/ET \right\rfloor - 2^{EC_1} \\ \left\lfloor d'/ET \right\rfloor - 2^{EC_1} - \left\lfloor \left(d'/ET - 2^{EC_1}\right) / 2^{\sigma_{EC}} \right\rfloor \times 2^{\sigma_{EC}} \end{cases} \quad (3)$$

3 Experimental Results

This section presents the experimental results obtained from eight 3D common polygonal models: “Armadillo,” “Brain,” “Cow,” “Golfball,” “Lucy,” “Maxplanck,” “Dragon,” and “Hand.” The proposed algorithm was implemented in Microsoft Visual C++ programming language on a personal computer with an Intel Core i7 2.67 GHz processor and 3 GB of memory. Table 1 shows the model information, including the number of vertices N_V, the number of faces N_F, and model size (represented by the diagonal length DL_{BV} of the bounding volume) of each model. Figure 4 shows the visual effect of each cover polygonal model. The embedded secret message is a 0/1 bit string randomly generated. The distortion between the cover model and the stego model was measured using normalized Hausdorff distance (NHD) [4], which is derived from dividing the Hausdorff distance by DL_{BV}. The number precision of the cover model and the stego model

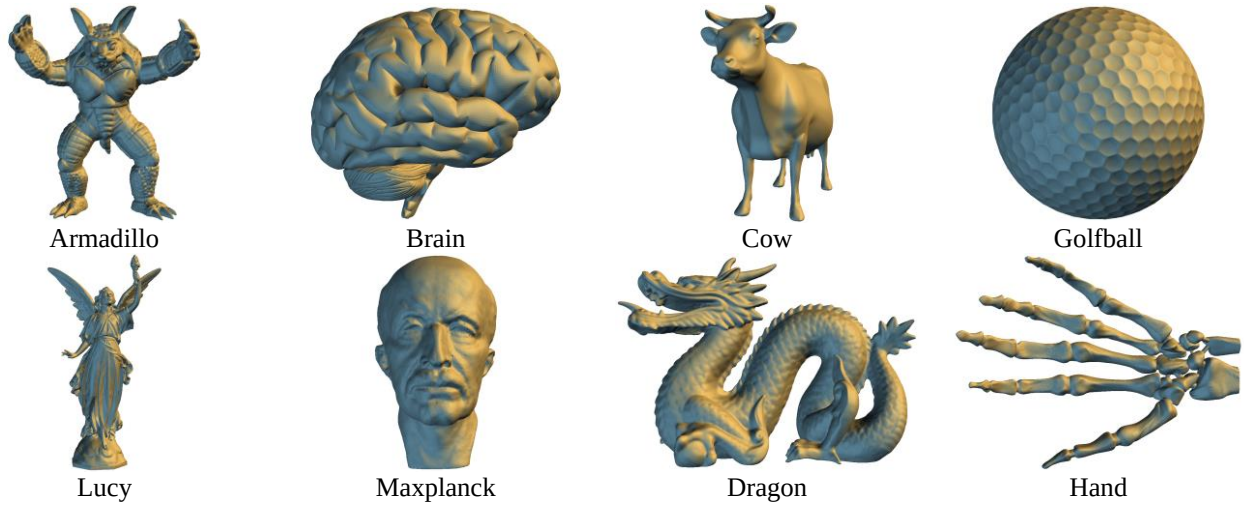

Figure 4: The visual effects of the cover models

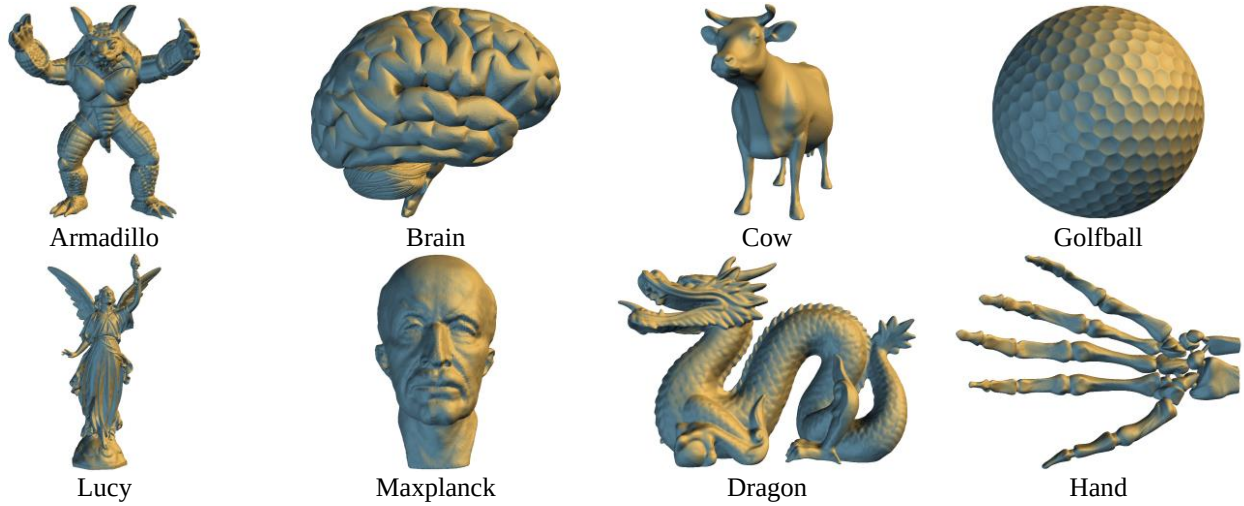

Figure 5: The visual effects of the stego model with an embedded secret message

are both precise to six decimal places. The experimental results show there is no error in the extracted secret message under the similarity transformation and vertex reordering attacks.

This section first presents the experimental results of the proposed algorithm, including its embedding capacity and the visual distortion of the different input models. Second, this section also presents the embedding capacity and model distortion of the Armadillo and Lucy models under different constant thresholds. Finally, this section compares the proposed algorithm with existing adaptive information hiding algorithms to demonstrate the feasibility of the proposed method.

Table 1 shows the results of the embedding capacity and the model distortion under the given embedding threshold $ET = 3\times10^{-6}$. For the maximum embedding capacity, no restriction is performed on the parameter σ_{EC}. The embedding capacity can achieve 9.61 to 18.77 bits per vertex and 9.71 to 18.79 bits per effective vertex, which is the vertex with the embedded secret message. The difference between the value of BitsPV and BitsPEV is small because only 0.1 to 2.4 percent of the total number of vertices cannot have the secret message embedded. The model distortion is acceptable, ranging only from 0.05% to 0.18% of the value DL_{BV} . Figure 5 shows the shading effects with insignificant distortion of the stego models.

This section presents the embedding capacity and model distortion of the Armadillo and Lucy models under different constant thresholds (Table 2). Other test models have similar results. When the constant threshold is smaller, indicating each vertex can only convey a reduced amount of secret message, the total embedding capacity is apparently lower and the model distortion is less. With an increasing constant threshold, both the embedding capacity and the model distortion are increased. From Table 2, our proposed algorithm is superior to previous algorithms with higher embedding capacity and lower model distortion for

the Armadillo model and the Lucy model under the threshold value 15 and 17 separately.

Table 2: The capacity and distortion comparison under different embedding thresholds

Model \ *CT*		5	10	15	17	∞	[8]
Armadillo	Capacity	862670	1725331	2581163	2784917	2784917	2220480
	NHD	0.0001%	0.0024%	0.0666%	0.1406%	0.1142%	0.0963%
Lucy	Capacity	1294295	2588590	3882815	4395901	4858167	4101995
	NHD	0.0000%	0.0003%	0.0091%	0.0337%	0.1608%	0.0555%

Table 3: A comparison of the adaptive methods of [3], [8], and the current method

Algorithm	[3]	[8]	Current Method
Capacity	3.00~6.00 bpv	7.76~18.00 bpv	9.71~18.79 bpv
Robustness	Similarity Transformation	Similarity Transformation	Similarity Transformation Vertex Reordering
Complexity	$O(N_F)$	$O(N_V^2)$	$O(N_E)$
Referencing Ratio	34.11%	64.51%	49.67%

Finally, this section compares the proposed algorithm with existing adaptive information hiding algorithms to demonstrate the feasibility of the proposed method. Table 3 shows that the proposed technique can convey the maximum embedding capacity within the current adaptive algorithms. The robustness is also superior to other two algorithms because of the vertex reordering attack. For the complexity comparison, Cheng and Wang's algorithm employs a new contagious diffusion technique with the time complexity $O(N_F)$ to generate a traversal path for each polygon; whereas our previous work introduces a vertex decimation process with the time complexity $O(N_V^2)$ for determining the referencing neighbors for each embedding vertex. The proposed algorithm adopts a modified breadth first search scheme that explores all edges of the polygonal model to determine the referencing neighbors. Thus, the complexity is only $O(N_E)$, whether N_E is the number of edges in the cover model. However, the proposed technique lowers the number of the referencing neighbors for each embedding vertex at approximately half of the actual neighbors. Despite this, the referencing ratio is still superior to that of Cheng and Wang's algorithm.

4 Conclusions

This study proposes an adaptive information hiding algorithm for polygonal models. The main point of this algorithm is to use a modified BFS method to efficiently derive the referencing neighbors for each embeddable vertex. Thereafter, the surface complexity of the embedding vertex is then estimated by the distance from the center of the referencing neighbors. Different amounts of secret messages are embedded according to the surface properties of each vertex. To decrease the model distortion caused by a large embedding capacity, a constant threshold is employed to control the maximum embedding capacity for each vertex. The proposed algorithm can provide a higher embedding capacity, higher robustness, and a lower model distortion under acceptable estimation accuracy. Most importantly, the performance for determining the referencing neighbors of each embeddable vertex can be significantly improved. With the help of experimental results, this study demonstrates the feasibility of this technique for 3D adaptive information hiding.

Acknowledgments

We thank Dr. Timothy Williams for his assistance in improving the clarity of this article. The authors also thank the anonymous reviewers for their constructive comments. This work was supported by the National Science Council under the grant numbers NSC 101-2221-E-468-026 and NSC 102-2221-E-468-025.

References

[1] M. W. Chao, C. H. Lin, C. W. Yu, and T. Y. Lee, "A high capacity 3D steganography algorithm," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 274-284, 2009.

[2] T. Y. Chen, M. S. Hwang, and J. K. Jan, "Adaptive authentication schemes for 3D mesh models," *International Journal of Innovative Computing, Information and Control*, vol. 5, pp. 4561-4572, 2009.

[3] Y. M. Cheng and C. M. Wang, "An adaptive steganographic algorithm for 3D polygonal meshes," *The Visual Computer*, vol. 23, pp. 721-732, 2007.

[4] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, pp. 167-174, 1998.

[5] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, Second Edition, Morgan Kaufmann, Burlington, 2008.

[6] L. C. Huang, L. Y. Tseng, and M. S. Hwang, "The study of data hiding in medical images," *IJNS* 14, pp. 301-309, Springer-Verlag, 2012.

[7] A. C. Rencher, *Methods of Multivariate Analysis*, Second Edition, Wiley, New York, 2002.

[8] Y. Y. Tsai, "An adaptive steganographic algorithm for 3D polygonal models using vertex decimation," *Multimedia Tools and Applications*, in press, DOI: 10.1007/s11042-012-1135-8.

[9] C. M. Wang and Y. M. Cheng, "An efficient information hiding algorithm for polygon models," *Computer Graphics Forum*, vol. 24, pp. 591-600, 2005.

[10] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "A comprehensive survey on three-dimensional mesh watermarking," *IEEE Transactions on Multimedia*, vol. 10, pp. 1513-1527, 2008.

Yuan-Yu Tsai received a B.S. degree in Department of Computer Science and Information Engineering from National Central University, Taiwan, in 2000, and his Ph.D. degree from the Institute of Computer Science at National Chung Hsing University, Taiwan, in 2006. He is currently an assistant professor at the Department of Applied Informatics and Multimedia, Asia University, Taichung, Taiwan. His research interests include computer graphics and information hiding algorithms for three-dimensional models and images. He is a member of the ACM and the IEEE Computer Society.

Wen-Ching Huang is a third-year student at the Department of Applied Informatics and Multimedia, Asia University, Taichung, Taiwan. Her research interests include information hiding algorithms for three-dimensional models and images.

Bo-Feng Peng is currently a Master's student. He received his B.S. degree in Applied Informatics and Multimedia from Asia University, Taiwan, in 2012. His research interests include information hiding algorithms for three-dimensional models and images.