International Journal of Innovative Computing, Information and Control
Volume **9**, Number **3**, March **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 1321–1335

A HIGH-CAPACITY DISTORTION-FREE INFORMATION HIDING ALGORITHM FOR 3D POLYGON MODELS

Chao-Hung Lin[1], Min-Wen Chao[2], Jyun-Yuan Chen[1]
Cheng-Wei Yu[2] and Wei-Yen Hsu[3,*]

[1]Department of Geomatics
[2]Department of Computer Science and Information Engineering
National Cheng Kung University
No. 1, University Road, Tainan 701, Taiwan

[3]Department of Information Management
National Chung Cheng University
No. 168, University Road, Minhsiung Township, Chiayi County 62102, Taiwan
*Corresponding author: shenswy@gmail.com

Received December 2011; revised April 2012

Abstract. *A high-capacity distortion-free information hiding algorithm for 3D polygon models is presented. We introduce a novel embedding approach called representation permutation to embed messages in the representation domain. The proposed approach embeds messages by permuting/rearranging vertex representation orders, triangle representation orders, and connectivity information. In contrast to the general data hiding schemes that embed messages by slightly modifying the appearance of cover media, the proposed approach will not degrade any perceptual quality of the cover model, that is, without visual distortion. In addition, the proposed approach can hide $(\partial_p + \beta_p)\, n_v$ bits on a polygon model (n_v represents the number of vertices, and the variables ∂_p and β_p are greater than 2.1 and 3.7, respectively). Furthermore, the proposed approach can easily combine with other spatial-based and spectral-based information hiding schemes to provide additional hiding capacity. Experimental results show that the proposed approach is efficient and can provide greater hiding capacity than recent techniques, while complying with the basic security requirement for steganography.*

Keywords: Information hiding, Steganography, Representation domain, Information security

1. **Introduction.** Information hiding has recently become an important research topic and has drawn a lot of attention. Information hiding encompasses a broad range of applications in which the messages are embedded into covert media for different purposes. The main types of information hiding are watermarking and steganography. Both techniques are used to imperceptibly convey private information by embedding data into various digital media. Watermarking focuses on ownership authentication or content protection, whereas steganography hides messages so that no one, apart from the sender and intended receiver, suspects the existence of a message. Compared with the cryptography, the advantage of steganography is that messages do not attract attention to themselves. Thus, cryptography protects the contents of a message, whereas steganography protects both messages and communicating parties. Thus, steganography has been widely used in computer science applications, such as in the following examples: military agencies require unobtrusive communications; criminals place great value on unobtrusive communications, as law enforcement tries to detect and trace hidden messages; digital election and digital cash make use of anonymous communication [1]. Another example would be

modern printers that hide messages by adding tiny yellow dots to each page. These dots, although barely visible actually contain encoded printer serial numbers.

In general, secret information is hidden in another seemingly innocuous host to achieve covert communication in the network. To successfully conceal the existence of secret messages, the host medium is usually chosen in the manner of being nothing relation with the hidden information. With the development of 3D hardware, 3D computing or visualization has become much more efficient than ever. This leads to widespread use of 3D models in various applications such as digital archives, entertainment and game industries [2-5,38-41]. Thus, 3D models can serve as popular and innocuous-looking hosts for hiding other types of digital content. The triangle mesh is commonly used in 3D model representation and is supported by various graphic packages and libraries. A 3D model is represented by elementary elements that include vertex coordinates, vertex normals, texture coordinates, and connectivity information. This un-regular representation is highly different from typical sampling representation such as digital images or videos. Thus, the information hiding schemes for traditional cover media [6-18] are not suitable for 3D models. In this paper, we concentrate on exploiting the characteristic and representation of polygon models to provide a distortion-free and highcapacity steganography for covert communication.

In this study, the hidden data, called payloads, are embedded in the representation domain, which can avoid degrading the perceptual quality of host media. We propose a representation permutation approach to embedding messages by permuting/modifying the vertex/triangle representation orders and connectivity information. This proposed approach can efficiently obtain the desired vertex/triangle permutation from a given hidden message and the desired permutation index (hidden value) from a given vertex/triangle permutation. In addition, the hiding capacity can be increased from 9.5 bits to around 19 bits per vertex depending on the user-defined group size. Furthermore, we combine the proposed representation permutation (embedded in the representation domain) with the multi-layered embedding (embedded in the spatial domain) [19] to provide additional hiding capacity. Experimental results show increased hiding capacity and minimal distortion in our embedding scheme compared with recent approaches [19,20]. The remainder of this paper is organized as follows. Section 2 reviews previous work. Section 3 provides an overview of our information hiding scheme. Section 4 presents our algorithm including embedding and extraction processes. Section 5 illustrates the experimental results and offers certain discussions. Section 6 summarizes the proposed methods.

2. **Related Work.** Many steganography methods [19-22] and watermarking methods [23-29] have been proposed for 3D polygon models. The main purpose of watermarking is to robustly withstand various malicious attacks for ownership authentication or content protection. Instead of focusing on the aspect of robustness, the steganography algorithm considers capacity, reliability, and security to keep hidden information in the public. An elegant steganography algorithm should hide as many distinguishable secret messages as possible, while being secure enough to prevent attacks from enemies. These conflicting goals have rendered most watermarking schemes unsuitable for steganography.

In this section, we refer only to steganography methods for 3D polygon models. Most of the steganography methods for 3D polygon models are inspired by the well-known concept of quantization index modulation (QIM) proposed by Chen and Wornell [30]. The basic idea of QIM is to split the host media into two states, that is, state “0” and “1”. The elementary elements of host media are quantized to the nearest state region according to the embedded messages. In the work of Cayre and Macq [21], a blind information hiding scheme based on a substitution procedure in the spatial domain was proposed. They adopted a technique called triangle strip peeling sequence (TSPS) presented in [29] to determine the vertex embedding order for security considerations. Extending the concept of QIM to 3D polygon models, they represented a triangle as a two-state geometrical object, that is, state “0” or “1” by dividing one edge of the triangle by $2n$. Then, the position of the third vertex in this triangle was quantized/shifted to the nearest state region according to the bit to be hidden. As a result, this method can carry approximately one bit per vertex. Wang and Cheng [22] introduced a multi-level embedding procedure, which is an extension of the embedding scheme proposed in [21], with the aim of increasing the hiding capacity. They embedded messages on three embedding levels, namely, sliding, extending, and rotating, by slightly shifting/quantizing the vertex position to the nearest state region. Therefore, this approach can provide roughly three times the capacity of [21]. Furthermore, in [20], Cheng and Wang extended their previous work [22] further to provide additional hiding capacity. In addition to their previously proposed multi-level embedding procedure (with 3 bits/per vertex capacity), they presented a novel approach called representation rearrangement procedure to hide more data (about 6 bits/per vertex) without additional visual distortion. Their idea is to hide payload by rearranging vertex and triangle orders in the representation domain. Inspired by the concept of QIM, the vertex/triangle representation orders are divided into two different states. Payload is hidden according to the vertex/triangle traverse and representation orders. Recently, Chao et al. [19] presented an embedding approach of high capacity and limited distortion called multi-layered information hiding scheme. Payload is hidden in the spatial domain in multiple layers. This approach can provide close to $3n_{layers}n_v$ bits capacity for a polygon model (approximately 30 bits/per vertex), where n_{layers} and n_v represent the number of hiding layers and vertices, respectively. The above-mentioned steganography approaches are all based on the QIM embedding scheme. Unfortunately, these approaches provide insufficient capacity for polygon models even though the payload is embedded in multiple layers [19].

A novel data-embedding technique called representation permutation approach, which is inspired by the concept of embedding in the representation domain [20], is proposed in this paper. This approach can embed a large amount of data (at least 45 bits/per vertex when group size is greater than 10), resulting in a capacity that is higher than related approaches can provide. Unlike QIM-based steganography methods [19-22], the proposed embedding method hides messages by permuting vertex/triangle orders and connectivity information in the representation domain.

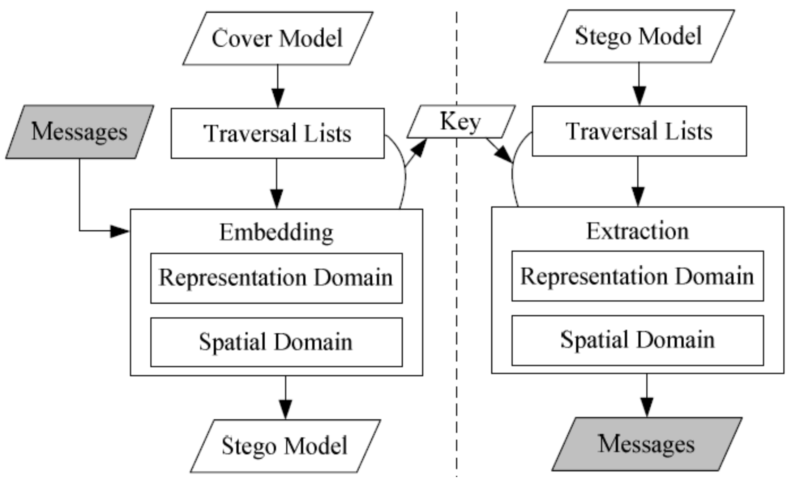


Figure 1. System workflow

3. **Overview of Steganographic System.** Figure 1 schematically illustrates the proposed information hiding scheme. This scheme consists of an ***embedding procedure*** and an ***extraction procedure***, each involving two main steps. First, vertex and triangle traverse lists are established. The traverse lists are used to determine the embedding order. Here, we propose a simple and efficient traversal approach based on breadth-first search (BFS) strategy and driven by a secret key. Second, the vertices and triangles in the traverse lists are uniformly partitioned into several independent groups. Payload is then embedded in each vertex/triangle group by permuting the vertex/triangle representation orders. In addition to embedding in the representation domain, messages are also embedded in the spatial domain. In our approach, the proposed representation permutation is combined with the multi-layered embedding approach [19]. In the extraction procedure, the embedding order can also be determined using the same secret key described in the first step. The payload can then be correctly extracted in that order.

4. **Proposed Stegonography Method.** In this section, we first describe the proposed traverse approach, followed by a description on the embedding and extraction approaches in the representation and spatial domains, respectively.

4.1. **Traversal sequence.** To determine the embedding order, most steganography approaches establish a vertex or triangle traverse list [20-22]. In [21], the TSPS technique is adopted to establish a vertex traverse list. However, this approach cannot visit all the available vertices. Thus the vertices of the cover model cannot be utilized fully. In [22], the authors improved the TSPS technique using a complex jumping strategy. To visit all triangles, the method jumps to an unvisited triangle and sets this triangle as the current triangle when the TSPS process cannot find any neighboring candidate triangles.

In this paper, a simple and efficient vertex/triangle traverse approach based on BFS strategy is presented. The traverse process starts with a user-selected initial triangle and a secret key. Under the BFS strategy, we traverse the vertices/triangles in a polygon model level by level. The traverse order for each depth level is decided by a sequence of binary bits, i.e., a secret key, denoted as $key : b_0 b_1 \cdots b_{n-1}$. The traverse order of the i-th level is determined by the (i mod n)-th bit value of the secret key, i.e., $b_{(i \bmod n)}$. If the bit value is "0", we traverse clockwise. If the bit value is "1", we traverse counterclockwise. To take security into more consideration, the start triangle is assigned in each level. First, an ordered index is assigned for each triangle f_i in a level, denoted as $index(f_i)$. In level k, the ordered index is determined by the bit value $key(k)$ and the last traversed triangle in the previous level, i.e., $(k-1)$-th level. The last traversed triangle is represented as a two-state geometric object, as shown in Figure 2. If the bit value $key(k)$ is "1", then the index order starts with the triangle adjacent to the right edge of the last traversed triangle and then traverses this level counterclockwise, as mentioned above. If the bit value $key(k)$ is "0", then the index order starts with the triangle adjacent to the left edge and then traverses clockwise. Once the triangle indexes are determined, the index of the start triangle is set to the remainder of the secret key divided by the number of triangles in a depth

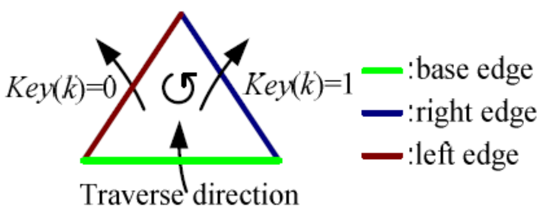


FIGURE 2. Illustration of the traverse direction

level. Take Figure 3 for example wherein the secret key is 0010 and the initial triangle is $f_f : (v_5, v_9, v_4)$. The edge represented by the first two vertices of the initial triangle is set as the base edge (edge $\overline{v_5 v_9}$ in this example). The base edge is used to determine the right and left edges in a triangle. The second bit value of the secret key is "0", and the last traversed triangle in the previous level is the initial triangle f_f. Therefore, the index order of the second level starts with the triangle f_b, and then traverses clockwise. As a result, the index order in the second level is $\{f_b, f_e, f_g\}$. Next, the value of the secret key is divided by the number of triangles in this level, i.e., $2/3 = 0.2$. We can deduce that the start triangle is f_g, i.e., $index(2)$, and the traverse order in this level is $\{f_g, f_b, f_e\}$. In the same way, the vertex traverse order is obtained. The vertex traverse order is established by directly following the triangle traverse order. Initially, the three vertex indexes in the initial triangle push into the vertex traverse list in the representation order ($\{v_5, v_9, v_4\}$ in this example). Then, following the triangle traverse order in the next level, the third vertex of the triangle is directly pushed to the vertex traverse list. Therefore, the vertex traverse order for the first level becomes $\{v_{10}, v_2, v_8\}$ in this example. We will visit all unvisited neighboring triangles/vertices until all the triangles/vertices of the cover model are visited. In this manner, the triangle and vertex traverse lists are obtained efficiently and securely. Figure 4 illustrates this traversal approach. The traverse order is visualized by color. The proposed traversal approach guarantees that all the triangles/vertices can be visited and that the security requirement for steganography is observed.

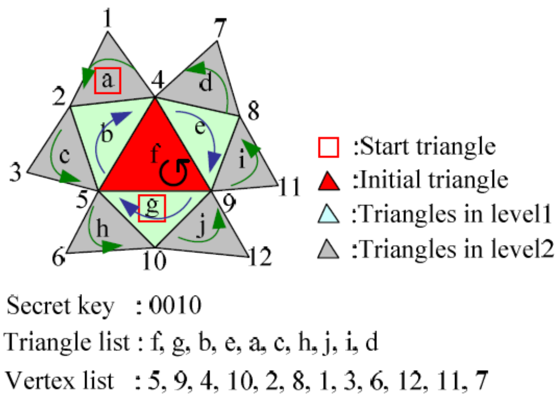


FIGURE 3. Example of vertex/triangle traversal

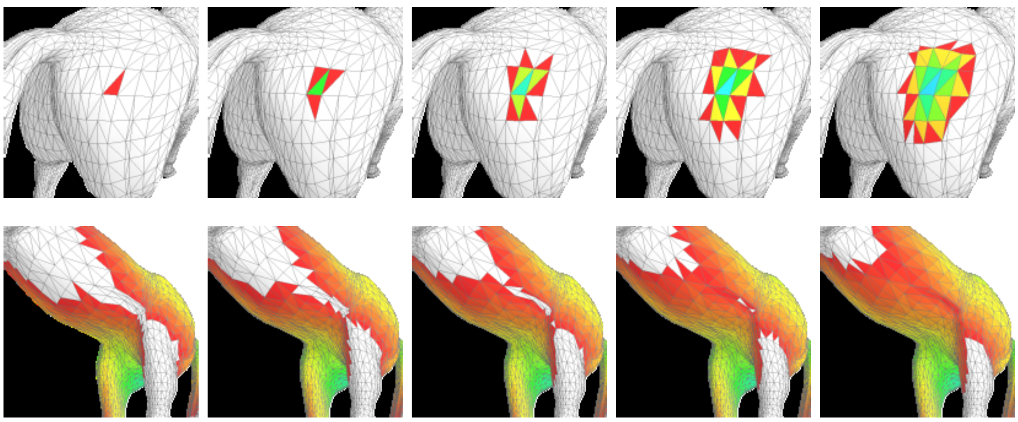


FIGURE 4. Traverse sequence generated by the proposed approach. The traverse order is visualized by color.

4.2. **Embedding/Extraction in the representation domain.** In general, 3D polygon models are represented by Cartesian coordinates of vertex (x, y, z) with a list of topology connectivity (i, j, k), where i, j, k represent the vertex indices of a triangle. Other attributes such as normal, color and texture coordinate can also be included in the representation. For clarity and conciseness, our description of the approach to embed payload is confined to the most important attributes: vertex coordinates and topology connectivity. The proposed embedding approach can be easily applied to the other attributes. The succeeding sections describe the proposed approach for embedding/extraction messages in vertex representation (Section 4.2.1) and topology connectivity (Section 4.2.2).

4.2.1. *Embedding/Extraction in vertex representation.* Once the vertex and triangle traverse lists are determined, the representation permutation strategy is adopted to hide messages in the representation domain. First, the vertex traverse list is uniformly divided into several distinct groups. Each group contains p vertices, denoted as $C = \{v_0, \cdots, v_{p-1}\}$. The basic idea is to hide messages by permuting the sequence $\{v_0, \cdots, v_{p-1}\}$ in each group. Each unique permutation represents a specified hidden value (bit string). For example in Figure 5 (the group size is 5 (from v_a to v_e)), the permutation "$v_a v_b v_d v_c v_e$" means the hidden message is 2 $(000010)_2$. We use the order of permutation as the hidden message. In this example, 6 bits can be hidden in each vertex group.

Permutation						Hidden value
v_a	v_b	v_c	v_d	v_e	:	0
v_a	v_b	v_c	v_e	v_d	:	1
v_a	v_b	v_d	v_c	v_e	:	2
		.				.
		.				.
		.				.
v_e	v_d	v_c	v_b	v_a	:	119

FIGURE 5. The vertex permutations and their corresponding hidden values

One simple way to realize this embedding/extraction approach is to construct a mapping table that records all possible permutations and their corresponding orders (hidden values). We can then embed/extract messages using the values in this table. Unfortunately, the search time and memory storage requirement for this table will be very huge when the group size p is very large. To solve this problem, we present a more efficient approach to search through the permutation sequences and orders. In the embedding process, given a message m, we want to embed this message into a vertex group/sequence $\{v_i\}_{i=0}^{p-1}$, that is, to find the corresponding permutation, where the suffix i represents the order of vertex v_i in the sequence. If the vertex v_0 is fixed in front of a permutation, there are $(p-1)!$ possible orderings in this permutation. If the vertices $v_0, \cdots, v_i$ are fixed in front of a permutation, there are $(p-i-1)!$ possible orderings in this permutation. Therefore, in this case, we call $(p-i-1)!$ the permutation coefficient of vertex v_i. To find the ordering (vertex sequence) for a given embedded message m, we divide m by the permutation coefficient of the first vertex in the sequence v_0, that is, $m/(p-1)! = q \ldots m'$, where q is the integer quotient, and m' is the remainder. The obtained quotient q represents the vertex order in the sequence that lies in the first position of the permutation. In other words, the vertex v_q is the first element in the permutation. Next, remove this vertex from the sequence and then reorder the sequence, that is $\{v_i\}_{i=0}^{p-2}$. In the same way, we divide the remainder m' by the permutation coefficient of the first vertex v_0 in the sequence, i.e., $(p-2)!$, to obtain the next element of the permutation. In this manner, we only use p divisions to find the corresponding permutation for a given embedded message. The following is a simple example where $m = 4$ and $p = 5$. After five steps, the permutation "$v_a v_b v_e v_c v_d$" can be obtained for the embedded message $m = 4$.

$m = 4\ (0000100)_2$

Step 1:

0	1	2	3	4	(sequence index)
v_a	v_b	v_c	v_d	v_e	

$4/4! = 0 \ldots 4 : v_a$

Step 2:

–	0	1	2	3	(sequence index)
v_a	v_b	v_c	v_d	v_e	

$4/3! = 0 \ldots 4 : v_a v_b$

Step 3:

–	–	0	1	2	(sequence index)
v_a	v_b	v_c	v_d	v_e	

$4/2! = 2 \ldots 0 : v_a v_b v_e$

Step 4:

–	–	0	1	–	(sequence index)
v_a	v_b	v_c	v_d	v_e	

$0/1! = 0 \ldots 0 : v_a v_b v_e v_c$

Step 5:

–	–	–	0	–	(sequence index)
v_a	v_b	v_c	v_d	v_e	

$0/0! = 0 \ldots 0 : v_a v_b v_e v_c v_d$

The goal is to extract the embedded message m from a permutation ρ, that is, $\left\{v_{\rho(i)}\right\}_{i=0}^{p-1}$, where $\rho(i)$ represents the i-th element in the permutation ρ. The extraction approach is similar to the embedding approach. First, the permutation coefficient of the first vertex $v_{\rho(0)}$ in the sequence is multiplied by the sequence order of vertex $v_{\rho(0)}$. Then, the vertex $v_{\rho(0)}$ is removed from the sequence, and the vertices in the sequence are reordered. This process is repeated until the sequence is empty. The message extraction is formulated in Equation (1). This process is also very efficient as only five multiplications and five additions are required to extract the message. The following is a simple example of extracting a message from permutation "$v_d v_a v_e v_c v_b$." After five steps, the extracted message is 104 (110100_2) for the permutation "$v_d v_a v_e v_c v_b$."

$$m = \sum_{i=0}^{p-1} (p-i-1)! \times \rho(i) \tag{1}$$

Permutation: $v_d v_a v_e v_c v_b$

Step 1:

0	1	2	3	4	(sequence index)
v_a	v_b	v_c	v_d	v_e	

$v_d : 4! \times 3$

Step 2:

0	1	2	–	3	(sequence index)
v_a	v_b	v_c	v_d	v_e	

$v_d v_a : 4! \times 4 + 3! \times 0$

Step 3:

–	0	1	–	2	(sequence index)
v_a	v_b	v_c	v_d	v_e	

$v_d v_a v_e : 4! \times 4 + 3! \times 0 + 2! \times 2$

Step 4: $\boxed{\begin{array}{llllll} - & 0 & 1 & - & - & \text{(sequence index)} \\ v_a & v_b & v_c & v_d & v_e & \end{array}}$

$v_d v_a v_e v_c : 4! \times 4 + 3! \times 0 + 2! \times 3 + 1! \times 1$

Step 5: $\boxed{\begin{array}{llllll} - & 0 & - & - & - & \text{(sequence index)} \\ v_a & v_b & v_c & v_d & v_e & \end{array}}$

$v_d v_a v_e v_c v_b : 4! \times 4 + 3! \times 0 + 2! \times 3 + 1! \times 2 + 0! \times 0\,(104 : 1101000_2)$

For a sequence with p vertices, there are $p!$ possible permutations. Therefore, k bits can be embedded in this sequence, where $2^k \leq p! < 2^{k+1}$. The hiding capacity of embedding in the vertex representation α_p is k/p bits per vertex. As a result, the approach provides a total hiding capacity of $\alpha_p n_v$ bits, where n_v represents the number of vertices.

Note that the representation order of the initial triangle and the group size p in the embedding process must be preserved, as it serves as basis for the extraction process. The initial triangle, it will not be placed in a group and permuted its representation order. To preserve and secure this initial triangle, it is moved to a specified position in the triangle representation after the triangle representation orders are rearranged (embedding messages). This specific position of the initial triangle is calculated as the remainder after dividing the secret key with the number of triangles. As for the group size p, a user-selected private integer is incorporated into a secret key. After receiving the stego model and the secret key, clients can extract the group size p from the secret key and locate the initial triangle using the stego model and secret key.

4.2.2. *Embedding/Extraction in topology connectivity information.* The approach of embedding/extracting messages in vertex representation can be applied to triangle representation. Permutation is extended to the connectivity information which is represented by three vertex indexes (i, j, k). To maintain the consistency of triangle normal, these three vertex indexes must be consistently arranged in clockwise or counterclockwise order. Therefore, three permutations are available for connectivity, that is, (i, j, k), (k, i, j), and (j, k, i). We specify the connectivity states 0, 1, and 2 for these three permutations, respectively. Combining these three permutations with the triangle representation orders will lead to more distinguishable permutations and increase hiding capacity in the triangle representation. For a triangle sequence containing p elements and three states $\left\{f_i^k\right\}_{i=0;k=0}^{p-1;2}$, there are $p! \times 3^p$ total possible permutations. All permutations and the corresponding orders (the hidden messages) of a triangle sequence with five elements are listed in Figure 6.

Permutation						Hidden value
f_a^0	f_b^0	f_c^0	f_d^0	f_e^0	:	0
f_a^0	f_b^0	f_c^0	f_d^0	f_e^1	:	1
f_a^0	f_b^0	f_c^0	f_d^0	f_e^2	:	2
f_a^0	f_b^0	f_c^0	f_d^1	f_e^0	:	3
		.				.
		.				.
		.				.
f_e^2	f_d^2	f_c^2	f_b^2	f_a^2	:	29159

FIGURE 6. The triangle permutations and their corresponding hidden values. The suffix symbols $a \ldots e$ represent the triangle order from a to e in the triangle sequence.

A triangle permutation ρ is represented by two sub-permutation functions ρ_α and ρ_β, where ρ_α represents the triangle index in the sequence and ρ_β represents the triangle stats. The embedding/extraction approaches presented in Section 4.2.1 can be used here. Only the permutation coefficients must be redefined. If the first triangle in the sequence f_0 is fixed in front of a permutation, then there are $(p-1)! \times 3^p$ possible orderings in the permutation. If the triangles $f_0, \cdots, f_i$ are fixed in front of a permutation, there are $(p-i-1)! \times 3^{(p-i)}$ possible orderings in this permutation. Therefore, the permutation coefficient for f_i is defined as $(p-i-1)! \times 3^{(p-i)}$. When the first triangle is fixed in front of a permutation, the number of possible orderings from $\{f_0^0, \cdots\}$ to $\{f_0^1, \cdots\}$ in the permutation is $(p-1)! \times 3^{(p-1)}$. When the triangles $f_0, \cdots, f_i$ are fixed in front of a permutation, the number of possible orderings from $\{f_0, \cdots, f_i^0, \cdots\}$ to $\{f_0, \cdots, f_i^1, \cdots\}$ in the permutation is $(p-i-1)! \times 3^{(p-i-1)}$. Therefore, the permutation coefficient for f_i^k is defined as $(p-i-1)! \times 3^{(p-i-1)}$. In the embedding process, the message m is divided by the permutation coefficients to obtain the corresponding permutation. In the extraction process, the permutation coefficients are multiplied by the sequence orders to obtain the message m. The message extraction is formulated as:

$$
\begin{aligned}
m &= \sum_{i=0}^{p-1} (p-i-1)! \times 3^{(p-i)} \times \rho_\alpha(i) + (p-i-1)! \times 3^{(p-i-1)} \times \rho_\beta(i) \\
&= \sum_{i=0}^{p-1} (p-i-1)! \times 3^{(p-i-1)} \times (3\rho_\alpha(i) + \rho_\beta(i))
\end{aligned} \tag{2}
$$

For a sequence with p triangles, the proposed approach can carry y bits, where $2^y \leq p! \times 3^p < 2^{y+1}$. Therefore, the hiding capacity for embedding in the triangle representation β_p is y/p bits per triangle. As a result, the approach can provide $\beta_p n_f$ bits of hiding capacity in the triangle representation, where n_f represents the number of vertices.

Note that the vertex indexes will change during vertex permutation. Therefore, in the embedding process, the messages are embedded using the vertex permutation approach. The vertex indexes in the face representation (connectivity information) are then modified according to the new vertex indexes obtained after vertex permutation. Then, the succeeding messages are embedded using the triangle permutation approach.

4.3. **Embedding/Extraction in spatial domain.** The proposed representation permutation approach embeds messages in the representation domain. Only the vertex/triangle representations and the connectivity information are modified; vertex coordinates (appearance of the mode) are preserved. In other words, the cover and stego models remain the same after messages are embedded in the representation domain. Therefore, the proposed distortion-free embedding approach can be successfully combined with other embedding approaches, which embed data in spatial domain or transform domain. Our embedding scheme first embeds messages in the representation domain, and the succeeding messages are then embedded in a different domain. In this paper, the multi-layered embedding approach [19] is adopted to embed messages in the spatial domain. Initially, three end vertices $V_aV_bV_c$ are selected from the principal axes of the cover models; then the line segment $\overline{V_aV_b}$ is aligned with the basis axis in the Cartesian coordinate system. Next, inspired by the concept of QIM, the line segment $\overline{V_aV_b}$ is uniformly divided into two state region subsets in an interleaved manner (e.g., 010101...). The state regions are denoted as R_0 and R_1, as shown in Figure 7. To recognize the hidden bit value in state region, each state region is divided further into two sub-regions, namely, the ***change region*** and the ***un-change region***. The purpose of dividing state regions is to find an empty region, i.e., the change region, among all state regions in order to proportionally move

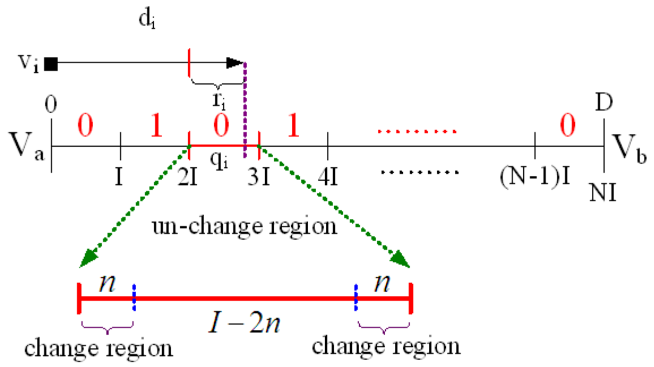


FIGURE 7. Illustration of state region q_i and its change region and un-change region. In this example, $q_i \in R_0$.

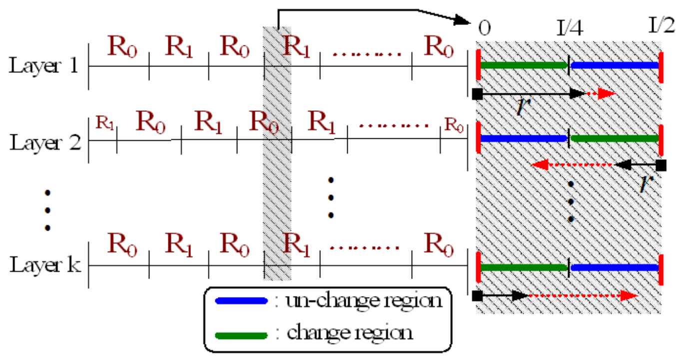


FIGURE 8. Illustration of the multi-layered embedding scheme. On the right side, a close look at the shaded interval of a state region is shown.

the mismatched vertices ($v_i \in R_k$, but the hidden bit value is not equal to k, $k = \{0, 1\}$) to the empty region. In this way, the original vertex position can be reversed when the hidden bit value is extracted.

The approach presented above is a singlelayered embedding scheme, which can easily be extended to the multi-layered version by directly adding more layers Similar to the single-layered version, two state region subsets are arranged in an interleaved manner for each layer in the multi-layered embedding scheme. However, a slight modification is required: the state regions in the even layers are shifted to the right by $I/2$ as shown in Figure 8. This causes vertex to swing, the moving direction of a vertex is bidirectional. Vertices are swinging back and forth in the half interval of a state region when embedding payloads on them. In this manner, additional data can be hidden on each vertex in multiple layers without enlarging the degree of distortion of the cover model. The lower bound distortion is limited to $I/2$.

This approach can embed data on x, y, z-components of a vertex coordinate. Every vertex in each layer can hide 3 bits of information. Therefore, the theoretical upper-bounded capacity of this embedding scheme is $3n_{layers}n_v$ bits, where n_{layers} represents the number of embedding layers.

4.4. **Hiding capacity analysis.** The proposed embedding scheme embeds messages in both the representation and the spatial domains. In the representation domain, the proposed approach can provide $(\alpha_p n_v + \beta_p n_f)$ bits of hiding capacity. For a two-manifold polygon mesh, we can deduce the number of triangles to be approximately twice the number of vertices. As a result, the proposed representation permutation approach can carry

Table 1. Hiding capacity in different group sizes

p	α_p	β_p	Hiding Capacity (bits/per vertex)
10	2.100	3.700	9.500
20	3.050	4.600	12.250
30	3.567	5.167	13.900
40	3.975	5.550	15.075
50	4.280	5.860	16.000
60	4.533	6.117	16.767
70	4.743	6.329	17.400
80	4.800	6.513	17.825
90	5.089	6.678	18.444
100	5.240	6.830	18.900

approximately $(\alpha_p + 2\beta_p)\, n_v$ bits in the representation domain. The hiding capacity is mainly dependent on group size p, as shown in Table 1. With a group size of 10, parameters α_p and β_p will be 2.1 and 3.7, respectively, indicating that each vertex can carry 9.5 bits. When group size is set to 100, the hiding capacity can reach 18.9 bits per vertex. In the spatial domain, the multi-layered embedding approach can hide about $3n_{layers}n_v$ bits on a polygon model. As a result, our approach can provide up to $(\alpha_p + 2\beta_p + 3n_{layers})\, n_v$ bits of hiding capacity in total.

5. **Experimental Results and Discussions.** Various 3D polygon models (shown in Table 2) are selected to test the performance of the proposed steganography approach. In the experiments, the commonly used root mean squared error (RMSE) and peak signal-to-noise ratio (PSNR) are adopted to measure the distortion. As the connectivity of the cover and stego models are identical, RMSE is defined as $\sqrt{\frac{1}{n_v}\sum_{i}^{n_v}\|v_i - v_i'\|^2}$, and PSNR is defined as $20\log_{10}\left(D_{\max}/\sqrt{MSE}\right)$, where vertices v_i and v_i' are the corresponding vertices in the cover and stego models, respectively; $D_{\max}$ represents the diagonal distance of the bounding box of the cover model. The group size p is set to 50 in all experiments. As expected, embedded messages in the representation domain do not degrade any perceptual quality of the cover models (RMSE is always 0.0). Embedded messages in the spatial domain only cause very small distortion (PSNR are all above 82 dB for the test models), as the number of hiding layers range from 7 to 13. Embedding in both the representation and spatial domains can provide a very high hiding capacity (at least 42 bits/per vertex in these test models) with little or no visual distortion.

Table 3 shows a detailed comparison of the proposed approach and three other steganography approaches [19-22] in spatial domain and representation domain. In the spatial domain, the hiding capacity of the proposed approach is about n_{layers} times as many as that of the approaches presented in [20,22], as our approach embeds messages in a multi-layer manner. In the representation domain, the capacity of our approach is much higher than that in [20] (about 2.5 times higher when the group size p is set to 50). Furthermore, the proposed embedding approach can successfully combine the representation domain and the spatial domain to provide a large hiding capacity ($(\alpha_p+2\beta_p+3n_{layers})$ bits /per vertex). Compared with previous 3D steganography methods [19-22], our approach significantly improves hiding capacity.

Table 2. Embedding results of various 3D models ("E.M." is the abbreviation of "Embedded Messages," and "p.v." is the abbreviation of "per vertex".)

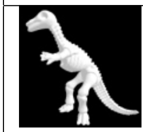
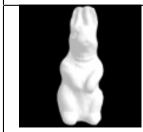
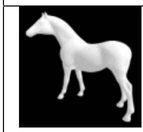
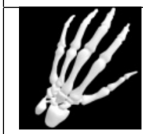
Model	#Vertices	#Faces	Representation Domain			Spatial Domain			Total (bits/p.v.)
			P	RMSE	E.M. (bits)	n_{layers}	E.M. (bits)	PSNR	
	56194	112384	50	0.0	898693	10	1685730	91.87	45.99
	67053	134075	50	0.0	1072507	7	1407756	107.19	44.13
	2502	5000	50	0.0	39707	12	89964	84.13	51.86
	34834	69664	50	0.0	557093	9	940464	100.57	42.99
	1058	2116	50	0.0	15838	13	41184	82.56	53.89
	116604	233204	50	0.0	1865600	10	3498060	91.30	45.99
	10044	20088	50	0.0	160293	11	331353	89.25	48.94

Table 3. A comparison between the proposed method and the methods presented in [19-22]

	Spatial Domain					Representation Domain		In Total	
	[21]	[22]	[20]	[19]	Our method	[20]	Our method	[20]	Our method
Capacity (bit/p.v.)	~ 1	3	3	$3n_{layers}$	$3n_{layers}$	6	$\alpha_p + 2\beta_p$	9	$\alpha_p + 2\beta_p$ $+3n_{layers}$
Extraction Scheme	blind	blind	blind	blind	blind	blind	blind	blind	Blind

6. **Conclusion and Future Work.** A high-capacity steganography scheme for 3D polygon models is presented. Unlike other QIM-based approaches [19-22], our approach embeds messages by permuting the vertex representation order and triangle representation

orders with connectivity information in a distortion-free manner. In addition to embedding in the representation domain, we also combine the multi-layered embedding approach to embed messages in the spatial domain. Our approach can provide high hiding capacity with little or no visual distortion. To the best of our knowledge, our approach can hide higher bit rates/vertex (i.e., 42 to 53 bits) compared with similar methods in steganography for 3D polygon models. In the near future, we plan to extend the proposed embedding scheme to the data of point clouds computer-aided design (CAD) models, and electroencephalography [31-37,42].

Acknowledgments. We would like to thank the anonymous reviewers for their valuable comments and suggestions. This paper was supported in part by the National Science Council, Taiwan (contracts NSC 100-2119-M-006-008, NSC 100-2119-M-006-025, NSC 101-2320-B-038-001 and NSC101-2221-E-038-004), and was supported in part by the Central Geological Survey, Ministry of Economic Affairs, Taiwan.

REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, Information hiding – A survey, *Proc. of the IEEE*, pp.1062-1078, 1999.

[2] I-C. Yeh, C.-H. Lin, O. Sorkine and T.-Y. Lee, Template-based 3D model fitting using dual-domain relaxation, *IEEE Transactions on Visualization and Computer Graphics*, vol.17, no.8, pp.1178-1190, 2011.

[3] M.-W. Chao, C.-H. Lin, C.-C. Chang and T.-Y. Lee, A graph-based shape matching scheme for 3D articulated objects, *Computer Animation and Virtual Worlds*, vol.22, pp.295-305, 2011.

[4] M.-W. Chao, C.-H. Lin, J. Assa and T.-Y. Lee, 3D motion retrieval from hand-drawn sketch, *IEEE Transactions on Visualization and Computer Graphics*, 2012.

[5] T.-Y. Lee, C.-H. Lin, Y.-S. Wang and T.-G. Chen, Animation key-frame extraction and simplification using deformation analysis, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.18, pp.478-486, 2008.

[6] C.-C. Chang, T.-C. Lu, Y.-F. Chang and C.-T. Lee, Reversible data hiding schemes for deoxyribonucleic acid (DNA) medium, *International Journal of Innovative Computing, Information and Control*, vol.3, no.5, pp.1145-1160, 2007.

[7] H.-C. Huang, B.-Y. Liao and J.-S. Pan, Special issue on information hiding and multimedia signal processing, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.1795-1795, 2009.

[8] C.-C. Lai, H.-C. Huang and C.-C. Tsai, A digital watermarking scheme based on singular value decomposition and micro-genetic algorithm, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.1867-1873, 2009.

[9] Z.-M. Lu and X.-W. Liao, Counterfeiting attacks on two robust watermarking schemes, *International Journal of Innovative Computing, Information and Control*, vol.2, no.4, pp.841-848, 2006.

[10] T.-H. Chen, T.-H. Hung, G. Horng and C.-M. Chang, Multiple watermarking based on visual secret sharing, *International Journal of Innovative Computing, Information and Control*, vol.4, no.11, pp.3005-3026, 2008.

[11] L.-D. Li, B.-L. Guo and J.-S. Pan, Robust image watermarking using feature based local invariant regions, *International Journal of Innovative Computing, Information and Control*, vol.4, no.8, pp.1977-1196, 2008.

[12] C.-C. Chang, C.-C. Lin and Y.-S. Hu, An SVD oriented watermark embedding scheme with high qualities for the restored images, *International Journal of Innovative Computing, Information and Control*, vol.3, no.3, pp.609-620, 2007.

[13] D.-J. Wang, T.-Y. Chen, T.-H. Chen and S.-W. Lin, A novel multipurpose image watermarking scheme based on block-relation, *International Journal of Innovative Computing, Information and Control*, vol.3, no.5, pp.1161-1172, 2007.

[14] S. D. Lin, Y. Kuo and M. Yao, An image watermarking scheme with tamper detection and recovery, *International Journal of Innovative Computing, Information and Control*, vol.3, no.6(A), pp.1379-1387, 2007.

[15] C.-C. Chen and D.-S. Kao, DCT-based zero replacement reversible image watermarking approach, *International Journal of Innovative Computing, Information and Control*, vol.4, no.11, pp.3027-3036, 2008.
[16] A. Ahmad and A. M. Mirza, Genetic programming based perceptual shaping of a digital watermark in wavelet domain, *ICIC Express Letters*, vol.4, no.5(B), pp.1893-1898, 2010.
[17] X. Luo, F. Liu and P. Lu, A LSB steganography approach against pixels sample pairs steganalysis, *International Journal of Innovative Computing, Information and Control*, vol.3, no.3, pp.575-588, 2007.
[18] X. Luo, D. Wang, W. Hu and F. Liu, Blind detection for image steganography: A system framework and implementation, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.433-442, 2009.
[19] 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.
[20] Y. M. Cheng and C. M. Wang, A high-capacity steganographic approach for 3D polygonal meshes, *The Visual Computer*, vol.22, pp.845-855, 2006.
[21] F. Cayre and B. Macq, Data hiding on 3-D triangle meshes, *IEEE Trans. Signal Processing*, vol.51, no.4, pp.939-949, 2003.
[22] C. M. Wang and Y. M. Cheng, An efficient information hiding algorithm for polygon models, *EUROGRAPHICS*, vol.24, no.3, pp.591-600, 2005.
[23] W. Y. Hsu, Wavelet-coherence features for motor imagery EEG analysis posterior to EOG noise elimination, *International Journal of Innovative Computing, Information and Control*, vol.9, no.1, pp.465-475, 2013.
[24] W. Y. Hsu, C. C. Lin, M. S. Ju and Y. N. Sun, Wavelet-based fractal features with active segment selection: Application to single-trial EEG data, *Journal of Neuroscience Methods*, vol.163, no.1, pp.145-160, 2007.
[25] W. Y. Hsu, W. F. P. Poon and Y. N. Sun, Automatic seamless mosaicing of microscopic images: Enhancing appearance with color degradation compensation and wavelet-based blending, *Journal of Microscopy-Oxford*, vol.231, no.3, pp.408-418, 2008.
[26] C. H. Lin, M. W. Chao, C. Y. Liang and T. Y. Lee, A novel semi-blind-and-semi-reversible robust watermarking scheme for 3D polygonal models, *The Visual Computer*, vol.26, pp.1101-1111, 2010.
[27] W. Y. Hsu and Y. N. Sun, EEG-based motor imagery analysis using weighted wavelet transform features, *Journal of Neuroscience Methods*, vol.167, no.2, pp.310-318, 2009.
[28] W. Y. Hsu, EEG-based motor imagery classification using neuro-fuzzy prediction and wavelet fractal features, *Journal of Neuroscience Methods*, vol.189, no.2, pp.295-302, 2010.
[29] R. Ohbuchi, H. Mazuda and M. Aono, Watermarking three-dimensional polygonal models, *Proc. of ACM Multimedia*, pp.261-272, 1997.
[30] B. Chen and G. W. Wornell, Quantization index modulation: A class of provably good methods for digital watermarking and information embedding, *IEEE Trans. Inf. Theory*, vol.47, no.4, pp.1423-1443, 2001.
[31] W. Y. Hsu, Continuous EEG signal analysis for asynchronous BCI application, *International Journal of Neural Systems*, vol.21, no.4, pp.335-350, 2011.
[32] W. Y. Hsu, Application of competitive hopfield neural network to brain-computer interface systems, *International Journal of Neural Systems*, vol.22, no.1, pp.51-62, 2012.
[33] W. Y. Hsu, Analytic differential approach for robust registration of rat brain histological images, *Microscopy Research and Technique*, vol.74, no.6, pp.523-530, 2011.
[34] W. Y. Hsu, EEG-based motor imagery classification using enhanced active segment selection and adaptive classifier, *Computers in Biology and Medicine*, vol.41, no.8, pp.633-639, 2011.
[35] W. Y. Hsu, Y. C. Li, C. Y. Hsu, C. T. Liu and H. W. Chiu, Application of multiscale amplitude modulation features and FCM clustering to brain-computer interface, *Clinical EEG and Neuroscience*, vol.43, no.1, pp.32-38, 2012.
[36] W. Y. Hsu, Enhanced active segment selection for single-trial EEG classification, *Clinical EEG and Neuroscience*, vol.43, no.2, pp.87-96, 2012.
[37] C.-H. Lin, P.-H. Tsai, K.-H. Lai and J.-Y. Chen, Cloud removal from multitemporal satellite images using information cloning, *IEEE Transactions on Geoscience and Remote Sensing*, 2012.
[38] Y. S. Wang, M. W. Chao, C. C. Yi and C. H. Lin, Cubist style rendering for 3D polygonal models, *Journal of Information Science and Engineering*, vol.27, no.6, pp.1885-1899, 2011.
[39] I. C. Yeh, C. H. Lin, H. J. Chien and T. Y. Lee, Efficient camera path planning algorithm for human motion overview, *Computer Animation and Virtual Worlds*, vol.22, no.2-3, pp.239-250, 2011.

[40] H.-K. Chu and C.-H. Lin, Example-based deformation transfer for 3D polygon models, *Journal of Information Science and Engineering*, vol.26, no.2, pp.379-391, 2010.

[41] Y.-S. Wang, C.-H. Lin and T.-Y. Lee, Interactive model decomposition using protrusive graph, *International Journal of Innovative Computing, Information and Control*, vol.4, no.8, pp.1887-1896, 2008.

[42] W.-Y. Hsu, C.-H. Lin, H.-J. Hsu, P.-H. Chen and I-R. Chen, Wavelet-based envelope features with automatic EOG artifact removal: Application to single-trial EEG data, *Expert Systems with Applications*, vol.39, no.3, pp.2743-2749, 2012.