# Vertex Shuffling : A Novel Information Hiding on 3D Model

Cong-Kai Lin*, Charlie Irawan Tan*, and Wen-Kai Tai*
* Department of Computer Science and Information Engineering,
National Dong Hwa University, Taiwan, R.O.C.
wktai@mail.ndhu.edu.tw

***Abstract*—This work presents an information hiding technique based on the vertex representation for 3D models. The vertex index of the sorted vertex sequence of a cover model (model without messages) is employed to hide messages sequentially, and to identify the correct arrangement of extracted message units. While embedding messages, a vertex is offset to $(j+1)th$ element in the output vertex sequence, which is saved as the stego model (model with messages), if the integer value of the message unit is $j$ (namely, convert a bit string into an integer). Some vertices are as message vertices and others as non-message and delimiter vertices to deal with collision which means that more than two vertices are offset to the same element. The hidden information is obtained by arranging all message vertices by the vertex order in the sorted vertex sequence. Experimental results show that the proposed approach achieves high capacity without any distortions, gains efficiency in real time, and is robust against rigid body transformations and uniform scaling.**

## I. Introduction

Steganography, the art of hiding one message in another, has been used over many centuries. Steganography ranges from simple hidden messages that can be deciphered by shifting each letter by a number of positions in the alphabet to watermarks that can be extracted from an image by a specific method, and has been applied widely in many topics related to information security. Steganography has even been used recently in 3D data, with similar purposes to those of 2D data. Whereas an image generally has hundreds of thousands of pixels, a typical mesh only has a few thousands vertices. Therefore, a small number of cover entities can be used. Additionally, hiding information in 3D data involves very complex and varied topological and geometric operations. Therefore, 3D data produces more complex problems than 2D data.

Inspired by Cheng and Wang [6], this work proposes an information hiding technique based on the vertex representation, not limited to the vertex state(0 or 1 state) in traversal order in [6], for 3D models with polygonal meshes or point sets. The representations of vertices in the vertex sets of a given cover model and of the stego model are exploited to embed and extract messages, respectively. A vertex sequence in ascending order of the three principal components of each vertex is obtained from the vertex set of a cover model by applying the PCA [10] to the vertices. The index of vertices in the ordered sequence is employed to embed information. Namely, embedding message is considered as a vertex order arrangement. The message to be embedded is partitioned into a sequence of message units. Each unit is a bit stream of the length the same as the integer value of logarithm of the quantity of vertex in the cover model. A vertex is mapped to $(j+1)$th element in the output vertex sequence, which is saved as the stego model, if the integer value of the message unit to be embedded is $j$. Some integers in the output vertex sequence may not be mapped because they do not correspond to any message units. Last some vertices in the sorted vertex set are selected as the non-message vertex to indicate the elements which are not mapped in the output vertex sequence. Some message units have the same integer value, which means that some vertices are mapped into the same element in the output vertex sequence. These vertices are colliding vertices. The first colliding vertex is employed as the delimiter vertex to annotate the output vertex sequence that vertices between the delimiter vertices are embedded in the same message.

The vertex in the stego model is read sequentially and a counter is maintained incrementally while extracting messages. A message unit with integer value $j$ is embedded in the current read message vertex if the counter is of value $j$. The counter is not incremented for colliding vertices. All distinct vertices in the stego model are sorted by their principal PCA components. All extracted message units are arranged in light of the vertex order in the sorted vertex sequence, and the hidden information is thus obtained.

The main contribution of this work is to improve the capacity of the steganogrpahic 3D models, thus enabling many applications that would previously have been impossible to apply due to the capacity limitation. For instance, large documents with many pictures can be embedded in for educational and communication purposes. Additionally, smaller models can be used to embed data for secure transmission. The proposed method preserves the robustness against subset of affine transformations (rigid body transformations and uniform scaling), similar to previous approaches [18][6][7]. However, the proposed method causes no distortion. The changes to the stego model are completely imperceptible, and hence not intelligible except to persons who are intended to see them. In summary, the proposed approach achieves high capacity without any distortions and gains efficiency in real-time and robustness against rigid body transformations and uniform scaling, as shown by the experimental results.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the proposed method. Experimental results are shown in Section 4. Conclusions are finally drawn in Section 5, along with recommendations for future research.

## II. Related Work

Steganography for 3D model was pioneered by Ohbuchi *et al.* [13], who introduced watermarking on 3D polygonal models. Several ideas of steganography methods have been proposed each with their own advantages and disadvantages. Many modifications and improvements to these methods have been developed. Some such methods, such as [13][3][1][12][18][6][20][23][2][22][4][21][19][8][7], work in the spatial domain, while others, including [11][14][17][16][15][9], work in the spectral domain.

Transforming the model to the spectral domain and applying the data embedding and extraction operations in this domain improve the robustness of the resultant stego model. Ohbuchi *et al.* [16][14][15] and Cotting *et al.* [9] hid data in the low frequency range of the model, making the model resistant to mesh simplification and random noises. With human interference, Cotting *et al.* [9] can even produce a stego model that can resist affine transformations. The processing time of these methods needed for embedding and extraction are relatively high, due to the domain transformation process. Furthermore, these methods have limited capacity. Therefore, these methods are more appropriate for data protection applications, such as watermarking, than for data hiding.

In contrast, methods that work on the spatial domain have better capacity but with weaker robustness. Based on TSPS method [13], Cayre and Macq [4] utilized the characteristic of a triangle; an edge represents the previous bit or the beginning of the information stream, and the other two edges represent the bit state of a bit of information. By projecting the top vertex to the base edge, the bit information contained in the triangle is employed to hide information. This method cannot guarantee to utilize all of the vertices in the mesh. Additionally, finding the triangle strip costs computation time. Wang and Cheng [18] used a similar method. However, instead of using the horizontal position of the top vertex, they also employed the vertical position and an angle between the triangle itself and another triangle formed by the base edge and the center of gravity. Furthermore, they reduced the computation time by a *kd*-tree when finding the triangle strip. Besides, Cho *et al.* [8] proposed another method to embed a watermark into 3D models by modifying the distribution of vertex norms instead of characteristic of triangles or vertices. [7] presented a new adaptive steganographic technique to increase the embedding capacity from three fixed bits per vertex to six bits at most per vertex. They proposed an adaptive minimum-distortion estimation procedure to achieve adaptability and preserve important shape features. Recently, Chao *et al.* [5] proposed a multilayered embedding approach that provides high capacity. Typically, methods that work on spatial domain will introduce some distortion to the cover model.

Most current methods work for 3D polygonal meshes. Several methods, such as those of Ohbuchi *et al.* [15], Wang and Wang [20] and Cotting *et al.* [9], utilize 3D models defined as point sets. Polygonal meshes provide fewer vertices than point set models, but have face information that can be used as the alternative medium. Higher vertex numbers allow a model to hide more information, but require more space and computing power to handle. Therefore, many applications use polygonal meshes rather than point set models. As 3D scanners become increasingly popular, the requirement that the model be defined as a point set should not be neglected. The proposed method works for both 3D polygonal meshes and point set models, since it is efficient, and it hides the message using the vertex order.

## III. Proposed Method

### A. Information Embedding

Given the vertex set of a cover model, all distinct vertices are first placed into the totally ordered set $V = \{v_0, v_1, ..., v_{n-1}\}$ in ascending order by their three principal components, after applying the principal component analysis (PCA) [10] to the set. Based on the PCA, the subset of affine transformations does not affect the vertex ordering (except for models including sphere, cylinder, and torus). Notably, any sorting method can be employed to sort the vertices.

The vertex order arrangement is considered as a mapping on a set $V$. Specifically, let $p : V \rightarrow I^+ = \{0, 1, 2, ..., n-1\}$ be a mapping function of a set $V$; that is, $p$ is a rearrangement of the vertices $v_0, v_1, ..., v_{n-1}$ according to the messages to be embedded. The message M to be embedded is converted into a binary representation and subdivided into a sequence of message units, $\{m_0, m_1, ..., m_{k-1}\}$, each of which is considered as a mapping $p$ for each vertex. Each message unit $m_i$ is of a length of $k$ bits, where $k = \lfloor \log_2 \|V\| \rfloor$ (the integer value of a message unit should not exceed the number of vertices of cover model) and $\|V\|$ is the number of vertex list. For example, $p : v_i \rightarrow m_j$ means that vertex $v_i$ is offset to the (INT($m_j$)+1)-th element in the rearranged vertex sequence $V'$ consists of $v'_0, v'_1, ..., v'_{n-1}$, where INT($m_j$) converts $m_j$ into integer. After mapping each vertex of cover model, the rearranged vertex sequence $V'$ is saved as the stego model.

The mapping $p$ may occur many to one mapping, called vertex collision, because of the same integer value of message units. For example, let $v_i < v_s < v_q$ be three colliding vertices, i.e., $p : v_i \rightarrow m_i, p : v_s \rightarrow m_s$, and $p : v_q \rightarrow m_q$ where $INT(m_i) = INT(m_s) = INT(m_q)$. Then, the first colliding vertex $v_i$ is introduced as an extra tag vertex to the sequence $\{v_i, v_s, v_q\}$, such that the new sequence $\{v_i, v_s, v_q, v_i\}$ is the output rearranged vertex sequence. The tag vertex $v_i$ is adopted as a delimiter vertex to annotate that $v_i$, $v_s$, and $v_q$ represent the same embedded message when extracting information from the stego model. Since the proposed mapping $p$ is many to one, many integers may not be mapped. To identify all integers that are not parts of the embedding message, another tag vertex is introduced, called non-message vertex and copied to all elements without mapping in the output vertex sequence. The last vertex of the sorted vertex set $V$ is chosen as the non-message vertex.

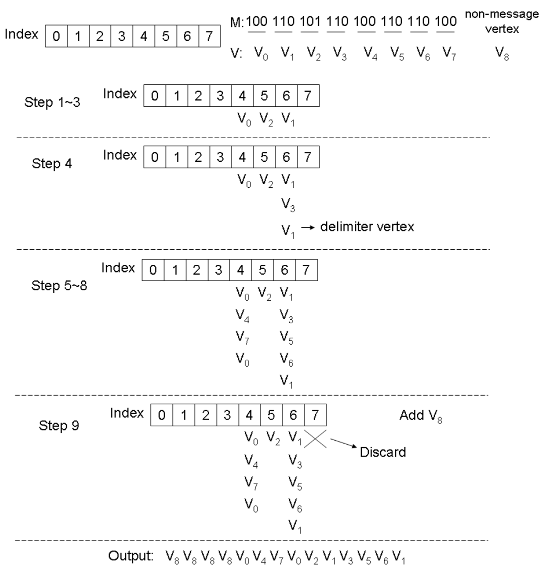


Fig. 1. The proposed information embedding method.

Figure 1 illustrates an example illustration of the proposed information embedding method. Assume that the sorted vertex set $V$ of a cover model has nine vertices $v_0 < v_1 < v_2 < \cdots < v_8$, and the message $M = \{m_0, m_1, ..., m_7\}$, bit stream, is of length $K = (\|V\| - 1) \times k = 8 \times 3 = 24$. The integer value of the message unit $m_0$ to be embedded in vertex $v_0$ is $(4)_{10} = INT(100)_2$. Hence, $p : v_0 \to (100)$, and vertex $v_0$ is offset to the fifth element of the rearranged vertex sequence. Similarly, $v_1$ and $v_2$ are offset to the seventh and sixth elements, respectively. Notably, when mapping $v_3$, $p : v_3 \to (110)$, the first colliding vertex $v_1$ is chosen as the delimiter vertex to delimit $v_3$, so that the message to be embedded can be unambiguously extracted from them. Likewise, $v_4$ and $v_7$, $v_5$ and $v_6$, are colliding with $v_0$ and $v_1$, and delimited by $v_0$ and $v_1$, respectively. When all message units are embedded in the vertices, the final vertex $v_8$ of $V$ is the non-message vertex, and is copied to the first (0), second (1), third (2), and fourth (3) elements in the rearranged vertex sequence to indicate that these integers are not parts of the message. Finally, the output vertex sequence $V' = \{v_8, v_8, v_8, v_8, v_0, v_4, v_7, v_0, v_2, v_1, v_3, v_5, v_6, v_1\}$ is saved as the stego model.

Nevertheless, simply choosing the last vertex as the non-message vertex causes the large number of redundant vertices when the colliding vertices occur frequently. To reduce the number of redundant vertices, the last $k_{non}$ vertices are chosen to encode successive non-message vertices. For example, consider last three vertices $\{v_a, v_b, v_c\}$ that can be used to represent the number of successive non-message vertices ranging from 1 to $2^3 - 1 = 7$ using $1\,\{v_a\}$, $2\,\{v_b\}$, $3\,\{v_a, v_b\}$, $4\,\{v_c\}$, $5\,\{v_a, v_c\}$, $6\,\{v_b, v_c\}$, $7\,\{v_a, v_b, v_c\}$. More vertices to be chosen as non-message vertices means less vertices left to be embedded in messages. The number of chosen non-message vertices is suitably determined by the formula $k_{non} = \lfloor \log_2 \|V\| \rfloor$ because $k_{non}$ non-message vertices can encode $2^{k_{non}} - 1$ successive non-message vertices. If the number of successive non-message vertices is large, then only use a small number of chosen non-message vertices instead so the number of redundant vertices is much reduced.

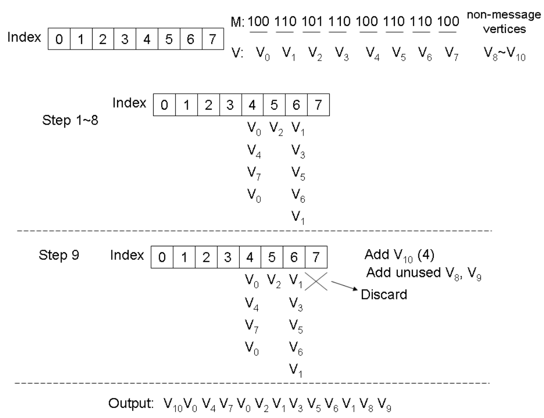


Fig. 2. Sample illustration of the proposed non-message vertex reducing method.

Figure 2 displays an illustration of the proposed non-message vertex reducing method. Assume that the sorted vertex set $V$ of a cover model has eleven vertices $v_0 < v_1 < v_2 < \cdots < v_{10}$, and that the message $M = \{m_0, m_1, ..., m_7\}$, bit stream, of length $K = (\|V\| - k_{non}) \times k = (11 - 3) \times 3 = 24$. The embedding procedure is the same as that of previous example above, except that the last three vertices $v_8$, $v_9$ and $v_{10}$ of $V$ can be chosen as the non-message vertices. The figure reveals that when all message units are embedded in the vertices, the vertex $v_{10}$ is chosen to represent the first four elements $(0 \sim 3)$ in the rearranged vertex sequence using the above rule, to indicate that these integers are not parts of the message. Additionally, the unused non-message vertices, $v_8$ and $v_9$, have to be added to the output sequence. Finally, the output vertex sequence $V' = \{v_{10}, v_0, v_4, v_7, v_0, v_2, v_1, v_3, v_5, v_6, v_1, v_8, v_9\}$ is saved as the stego model.

### B. Information Extraction

Based on representation domain in our method, the vertex sequence in the set $V'$ of a stego model conveys the embedded information of each message unit, and the vertex order in the sorted vertex set $V$ to $V'$ indicates the order of the message unit. All message units of the hidden information are then obtained by counting the message vertices in $V'$ and rearranged by the vertex order in $V$.

To arrange the extracted message units into the hidden information correctly, distinct vertices in $V'$ are sorted into a nondecreasing vertex sequence $V$ by PCA in the same way as in the embedding process. The message unit extracted from each message vertex is arranged in the same order as the order of its message vertex in the sorted sequence.

The tag vertices for annotating the non-message vertices are the last $k_{non} = \lfloor \log_2 \|V\| \rfloor$ vertices in the sorted vertex set $V$. The vertices in $V'$ are processed one by one when extracting the message unit from them. A counter is maintained to

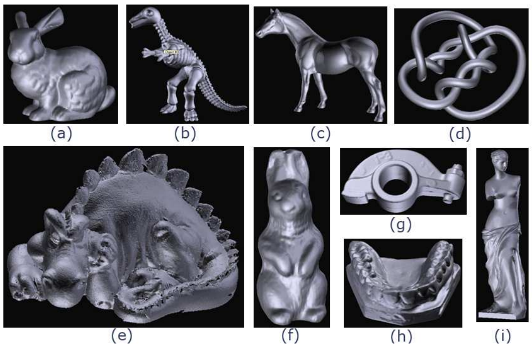


Fig. 3. Sample models used in our experiments. (a) bunny, (b) dinosaur, (c) horse, (d) knots, (e)Phlegmatic dragon, (f) rabbit, (g) rocker arm, (h) teeth, and (i) venus.

indicate the integer value of an extracted message unit for the message vertex. The counter increments by one whenever reading each message vertex but doesn't increment for colliding vertices except reading delimiter vertex. Moreover, the counter increments by a value that is implied by a set of non-message vertices.

For example, as illustrated in Figure 2, the vertex set $V' = \{v_{10}, v_0, v_4, v_7, v_0, v_2, v_1, v_3, v_5, v_6, v_1, v_8, v_9\}$ is read when extracting information from the stego model, yielding the sorted vertex set $V = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$. We have $k_{non} = \lfloor \log_2 \|V\| \rfloor = 3$ and $v_8$, $v_9$, and $v_{10}$ are three tag vertices for encoding successive non-message vertices. The first read vertex is $v_{10}$, a non-message vertex, so the counter increments by 4($2^2$). Next, $v_0$ is a message vertex and the counter increments by one so the integer value $4(100)$ (vertex index starts from 0) of a message unit is embedded in $v_0$. Although the next read vertex is $v_4$, the counter doesn't increments by one until $v_0$ is read again because a delimiter vertex $v_0$ appears afterward. Therefore, $v_0$, $v_4$ convey message units with the same integer value so does $v_7$. Similarly, $v_1$, $v_3$, $v_5$, and $v_6$ have the same hidden message unit. Finally, $v_8$ and $v_9$ are non-message vertices so the extracting process ends. All message units are arranged in terms of the sorted vertex order in $V$, and the correct message unit sequence is obtained as 4(100), 6(110), 5(101), 6(110), 4(100), 6(110), 6(110) and 4(100).

## IV. Experimental Results

The proposed method was implemented using the C++ programming language, and compared with recent related work. All experiments were performed on a PC with an Intel Pentium IV 2.4GHz processor and 512MB main memory.

In our experiments, the cover and stego models are obj file format. Similarly, obj-like file formats are also suitable for our method. Many types of messages such as image, logo, sound, video, and text, which are regarded as binary data can be embedded in the cover models. Sample visual models and messages are shown in Figure 3 and Figure 4 separately.

As shown in Table I, we use EG 2007 Phlegmatic dragon model with 336,017 vertices to evaluate the capacity of the proposed method and to estimate what types of messages are well suited for the proposed method. The message unit is 18 = $\lfloor \log_2 \|336,017\| \rfloor$ bits for all messages and the number of redundant vertices summing up non-message vertices and delimiter vertices. The capacity of bits per vertex vary from message to message, showing that the capacity of bits per vertex in the proposed method is message-dependent. The capacity of bits per vertex is high, and is close to the number of bits in the message unit for embedding logo and text messages. However, embedding other types of messages reveals that the capacity of bits per vertex is not as high as that of logo and text messages. Nonetheless, the proposed method has at least 12 bits per vertex. The distribution of the integer value of the message unit with small variation, like those of logo and text message types, means that the number of message units with same integer values does not spread out, and many of them have high multiplicity. Therefore, a small number of redundant vertices results in a high per-vertex capacity of both logo and text message types. As Figure 5, we illustrate that the concentrated distribution of message units results in a low redundancy ratio.

The redundant vertices do not change the geometry, topology or appearance of the 3D model, so any hidden messages in the model might not be discerned. However, the redundant vertices might increase the file size of the stego model, possibly breaking the imperceptibility of information hiding. The proposed solution to this problem is to provide an interaction mechanism to minimize the number of redundant vertices in the stego model but with acceptable high capactiy. Figure 6 presents the redundancy ratios and capacity of bits per vertex for each message type with respect to different lengths of the message unit. Different message lengths lead to different redundancy ratios and capacities of bits per vertex for different types of messages. In general, the smaller length of the message unit, the lower redundancy ratio but higher capacity of bits per vertex. Namely, with smaller length of the message unit, the message unit distribution is dense, making the delimiter vertex cost-effective. Each message type has its optimal length of the message unit. Namely, the length of the message unit can be interactively controlled in our method to minimize the redundancy ratio, and thus maintain a high imperceptibility without sacrificing the capacity of bits per vertex. Therefore, the proposed method is well suited for embedding messages with a concentrated distribution of message units like logos and texts, demostrating high capacity and low redundancy ratio. Table II presents the capacity statistics of the proposed method and that of Cheng and Wang [6] for 3D models with the number of vertices ranging from 14,007 to 116,604. The message to be embedded is a text file, and the message unit for the proposed method is shown in the last column. The table indicates that the proposed method has a higher capacity than the previous work. Although the proposed method introduces redundant vertices into the stego model, it still has a higher capacity of bits per vertex than Cheng and Wang's method. The capacity is fixed in Cheng and Wang's method, however, it

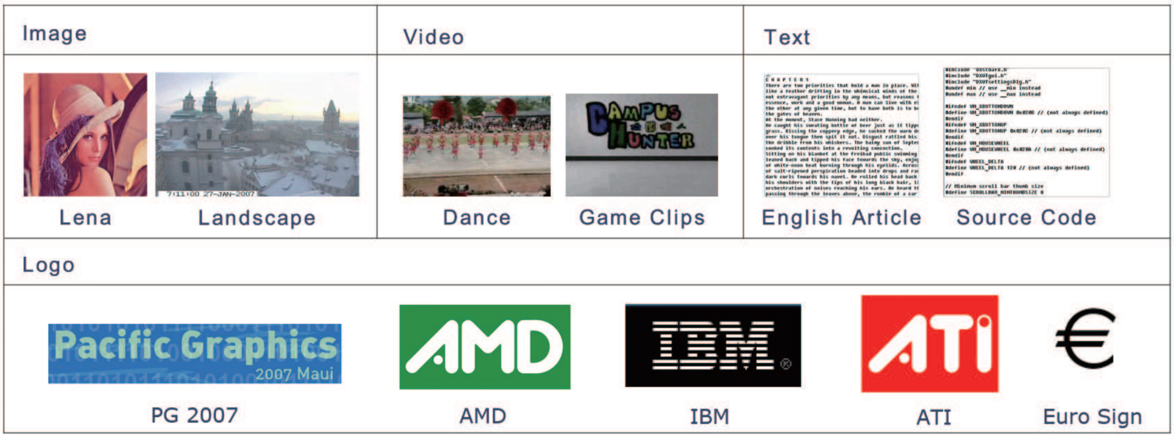


Fig. 4. Sample visual messages used in our experiments.

TABLE I
STATISTICS OF REDUNDANCY RATIO AND CAPACITY OF BITS PER VERTEX BY THE PROPOSED METHOD. THE CAPACITY IS MESSAGE-DEPENDENT. THE LOGO AND TEXT MESSAGE TYPES WITH A HIGH CONCENTRATED DISTRIBUTION OF THE MESSAGE UNIT PRESENT A LOW REDUNDANCY RATIO BUT HIGH CAPACITY OF BITS PER VERTEX.

| Type | Name | Non-message vertices | Delimiter vertices | Redundant vertices | Stego model | Redundancy ratio | Capacity bits per vertex |
|---|---|---|---|---|---|---|---|
| Image | Lena | 53,214 | 55,200 | 108,422 | 444,439 | 32.267% | 13.608 |
| | Landscape | 36,037 | 30,541 | 66,567 | 402,584 | 19.811% | 15.023 |
| Logo | PG 2007 | 23,877 | 18,288 | 42,153 | 378,170 | 12.545% | 15.993 |
| | AMD | 3,784 | 567 | 4,338 | 340,355 | 1.291% | 17.770 |
| | IBM | 3,619 | 688 | 4,294 | 340,311 | 1.278% | 17.772 |
| | ATI | 7,931 | 786 | 8,706 | 344,723 | 2.591% | 17.544 |
| | Euro Sign | 12,296 | 1,815 | 14,101 | 350,118 | 4.197% | 17.274 |
| Sound | Song | 68, 350 | 68,885 | 137,247 | 473,264 | 40.845% | 12.779 |
| | Choir | 64,883 | 70,558 | 135,435 | 471,452 | 40.306% | 12.828 |
| | Music | 66,746 | 76,714 | 143,454 | 479,471 | 42.692% | 12.614 |
| | Speech | 66,088 | 48,572 | 114,654 | 450,671 | 34.121% | 13.420 |
| Video | Dance | 73,720 | 47,952 | 121,667 | 457,684 | 36.209% | 13.214 |
| | Game Clips | 61,468 | 28,142 | 89,602 | 425,619 | 26.666% | 14.210 |
| Text | English Article | 13,734 | 6,906 | 20,645 | 356,662 | 6.144% | 16.957 |
| | Source Code | 25,146 | 13,370 | 38,522 | 374,539 | 11.464% | 16.148 |

increases with the number of vertices in the cover model in the proposed method. Each vertex has approximately $\lfloor \log_2 \|V\| \rfloor$ bits per vertex. The proposed method has the highest capacity than others when the number of vertices in a cover model is larger than $512(2^9)$. Table III presents the timing statistics of the proposed method and that of Cheng and Wang [6]. The computation time of both methods increases linearly with respect to the number of vertices in the testing models. The proposed embedding and extraction processes are fast because the proposed mechanism is simple and effective. The proposed method is an average of about 121 times the speed of Cheng and Wang's method.

Table IV compares the proposed approach with other recent steganographic approaches, such as Maret & Ebrahimi [12], Wang & Cheng [18], Cheng & Wang [6], and Cheng & Wang [7], in terms of capacity, type of extraction, robustness and domains on which methods based to embed messages. Both methods, Cheng & Wang [6] and ours, based on the representation domain have high capacity. Notably, the proposed method not only has a much higher capacity than the others, but also takes the shortest time.

Unlike watermarking approaches, in which the watermark should remain recognizable after attacks, steganography usually concerns high capacity, security and blind extraction, but not robustness. The major goal of this work is to maximize the capacity. Experimental results show that the proposed approach has a higher capacity than previous steganographic methods. Upon robustness, the proposed approach can only resist trivial attacks, such as rigid body transformations and uniform scaling since any non-rigid or non-uniform scale would hurt the appearance of the 3D model, and therefore we can assume this kind of transformations will not be used.

As described in Section III, based on the representation, the spatial information such as the position of vertices, face and normal remains unchanged, but the sequence of vertices is rearranged. Therefore, the cover model and stego model are the same without any distortion, but the stego model includes redundant vertices, namely the non-message and delimiter vertices. The number of redundant vertices depends on the distribution of the message unit of the message to be

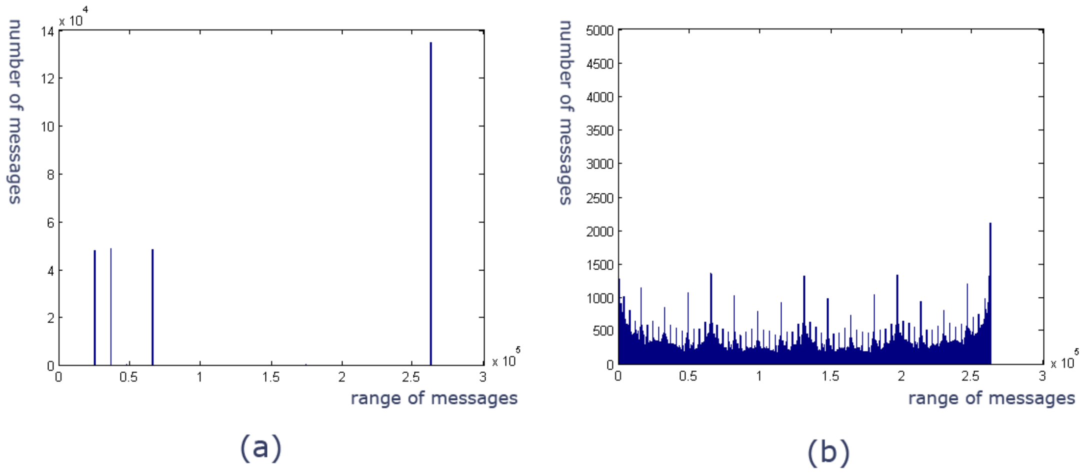


Fig. 5. The message distributions of two cases which represent the best case (a) and the worse case (b). (a)The distribution of IBM logo with message unit:18 bits. (b)The distribution of music with message unit:18 bits.

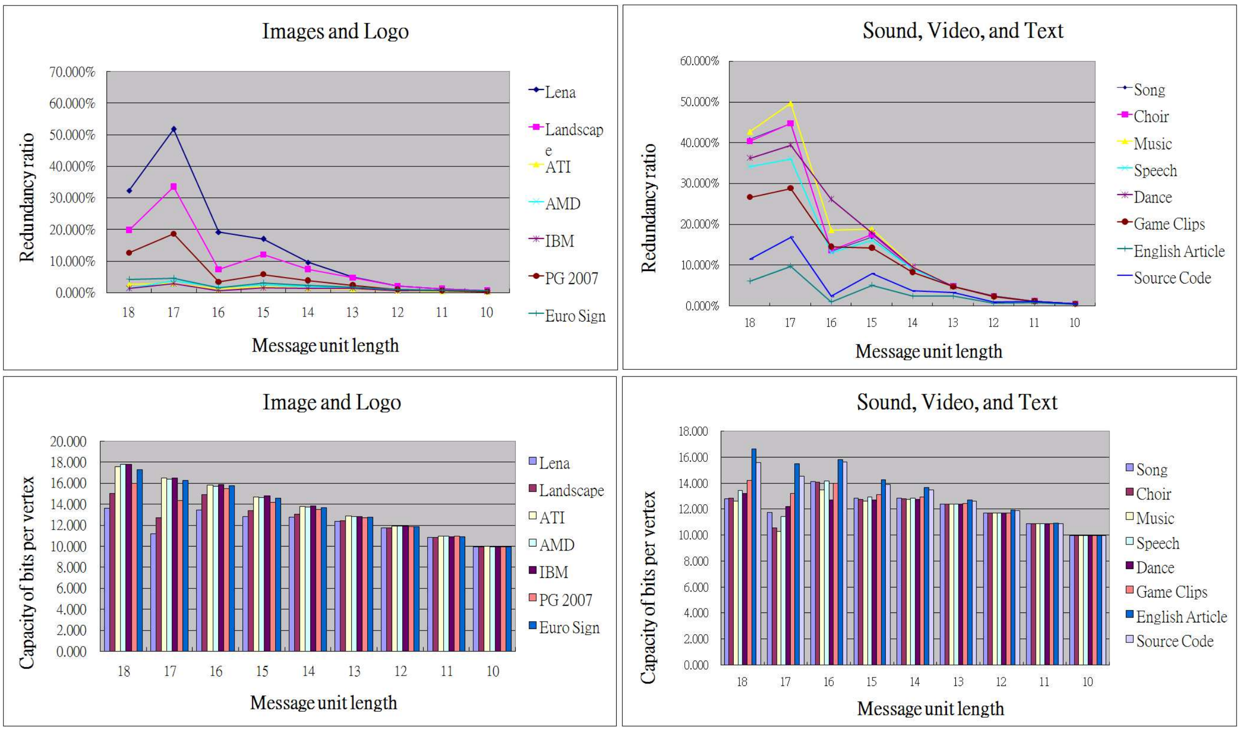


Fig. 6. The statistics of redundancy ratios and capacity of bits per vertex for testing message types with respect to different lengths of the message unit. To keep a high imperceptibility, one can select the optimal message unit length for a message type to minimize the redundancy ratio but maximize the capacity.

TABLE II
CAPACITY STATISTICS AND COMPARISON OF THE PROPOSED METHOD AND THAT OF CHENG AND WANG [6] FOR TESTING MODELS.

| Cover model | Vertices | Embedded messages(bits) [6] | Ours | Redundant vertices Ours | Bits per vertex [6] | Ours | Message unit bits |
|---|---|---|---|---|---|---|---|
| teeth | 116,604 | 1,049,417 | 1,865,408 | 5,513 | 9.00 | 15.28 | 16 |
| rabbit | 67,038 | 603,328 | 1,072,352 | 4,455 | 9.00 | 15.00 | 16 |
| dinosaur | 56,194 | 505,727 | 842,685 | 15,525 | 9.00 | 11.75 | 15 |
| horse | 48,485 | 436,346 | 727,050 | 14,811 | 9.00 | 11.49 | 15 |
| rocker arm | 40,177 | 361,584 | 602,430 | 14,027 | 9.00 | 11.11 | 15 |
| knots | 23,232 | 209,081 | 325,052 | 6,393 | 9.00 | 10.97 | 14 |
| venus | 19,847 | 187,771 | 276,374 | 6,167 | 9.46 | 10.62 | 14 |
| bunny | 14,007 | 125,584 | 181,922 | 5,457 | 8.97 | 9.35 | 13 |

TABLE III

A TIMING COMPARISON OF OUR METHOD AND THAT OF CHENG AND WANG [6] WITH RESPECT TO TESTING MODELS.

| Cover model | Vertices | Triangles | Timing (ms) [6] | Timing (ms) Ours | Speed up | Extraction time (ms) |
|---|---|---|---|---|---|---|
| teeth | 116,604 | 233,204 | 7,151 | 71 | 100.51 | 3 |
| rabbit | 67,038 | 134,074 | 4,591 | 41 | 112.85 | 1 |
| dinosaur | 56,194 | 112,384 | 3,962 | 32 | 124.93 | 1 |
| horse | 48,485 | 96,966 | 3,258 | 28 | 117.18 | 1 |
| rocker arm | 40,177 | 80,354 | 2,581 | 21 | 121.09 | 1 |
| knots | 23,232 | 46,464 | 1,366 | 12 | 111.38 | $<< 1$ |
| venus | 19,847 | 43,357 | 1,256 | 11 | 118.08 | $<< 1$ |
| bunny | 14,007 | 27,826 | 991 | 7 | 150.96 | $<< 1$ |

TABLE IV

COMPARISONS OF RECENT RELATED STEGANOGRAPHIC METHODS. NOTABLY, THE CAPACITY IS DENOTED AS AVERAGE BITS PER VERTEX, AND THE ROBUSTNESS CONCERNS TRANSFORMATION.

| | [12] | [18] | [6] | [7] | Ours |
|---|---|---|---|---|---|
| capacity | 0.5 | 3 | 9 | $3 \sim 6$ | $> 9$ |
| extraction | blind | blind | blind | blind | blind |
| robustness | affine | affine | affine | affine | subset of affine |
| domain | transform | spatial | spatial& representation | spatial | representation |

embedded.

## V. CONCLUSIONS AND FUTURE WORK

This work presents an efficient and high-capacity technique of embedding information without distortions, based on exploiting the representation information of 3D models. While embedding information, the message unit is embedded in the message vertex by offsetting the vertex to $(j+1)th$ element in the output sequence, where $j$ denotes the integer value of the message unit. Considering a stego model, the embedded message unit is extracted by the counter that is maintained when reading vertices from the model. The hidden information is then obtained by arranging the message unit correctly according to the vertex order in the sorted vertex sequence.

The inherent nature of representation domain based information hiding approaches means that the proposed method does not distort the cover model. The proposed method is robust against subset of affine transformations such as uniform scaling, translation and rotation. The capacity of bits per vertex is as high as the value of logarithm of the number of vertices of a cover model. The capacity of the proposed approach is comparable with that of other high-capacity techniques when the cover model has more than 512 vertices, and the highest among all previous works. The proposed idea of embedding messages using the vertex order rearrangement is simple and easy to implement, making it a very efficient, real-time method. The proposed technique is well suited for applications such as content annotation and authentication of 3D models, for which high-capacity and low distortion are the key requirements. The proposed method could also be adopted for secret communication applications. Moreover, our method can keep a high imperceptibility by interactively choosing the length of message unit yet with high capacity.

Although a few redundant vertices have been introduced into the stego model, and may not arouse suspicion when the model size is large, these vertices increase the model size. The imperceptibility of the embedded information is an important requirement in steganography. Minimizing the number of extra vertices decreases the likelihood of detecting a hidden message. To reduce the number of redundant vertices and improve the imperceptibility, future work will be to study a mechanism using the message unit with non-static length for the message vertex, thus significantly reducing the chance of vertex collision. Also, another novel vertex arrangement method is worth exploring to increase the capacity.

## REFERENCES


[1] Aspert, N., Drelie, E., Maret, Y., Ebrahimi, T.: Steganography for three-dimensional polygonal meshes. In: Proceedings of SPIE, vol. 4790, pp. 211–219 (2002)

[2] Benedens, O.: Geometry-based watermarking of 3d models. IEEE Comput. Graph. Appl. **19**(1), 46–55 (1999). DOI http://dx.doi.org/10.1109/38.736468

[3] Cayre, F., Devillers, O., Schmitt, F., Maître, H.: Watermarking 3d triangle meshes for authentication and integrity. In: INRIA Research Report RR-5223 (2004)

[4] Cayre, F., Macq, B.: Data hiding on 3-d triangle meshes. IEEE Transactions on Signal Processing **51**(4), 939–949 (2003)

[5] Chao, M.W., Lin, C.H., Yu, C.W., Lee, T.Y.: A high capacity 3d steganography algorithm. IEEE Transactions on Visualization and Computer Graphics **15**(2), 274–284 (2009). DOI http://dx.doi.org/10.1109/TVCG.2008.94

[6] Cheng, Y.M., Wang, C.M.: A high-capacity steganographic approach for 3d polygonal meshes. Vis. Comput. **22**(9), 845–855 (2006). DOI http://dx.doi.org/10.1007/s00371-006-0069-4

[7] Cheng, Y.M., Wang, C.M.: An adaptive steganographic algorithm for 3d polygonal meshes. Visual Comput. **23**(9-11), 721–732 (2007). DOI http://dx.doi.org/10.1007/s00371-007-0147-2

[8] Cho, J.W., Prost, R., Jung, H.Y.: An oblivious watermarking for 3-d polygonal meshes using distribution of vertex norms. IEEE Transactions on Signal Processing **55**(1), 142–155 (2007)

[9] Cotting, D., Weyrich, T., Pauly, M., Gross, M.: Robust watermarking of point-sampled geometry. In: SMI'04: International Conference on Shape Modeling and Applications 2004, pp. 233–242 (2004)

[10] Jolliffe, I.T.: Principal Component Analysis. Springer-Verlag, New York (1986)

[11] Kanai, S., Date, H., Kishinami, T.: Digital watermarking for 3d polygons using multiresolution wavelet decomposition. In: Proceedings of Sixth IFIP WG 5.2 GEO-6, pp. 296–307 (1998)

[12] Maret, Y., Ebrahimi, T.: Data hiding on 3d polygonal meshes. In: MM&Sec '04: Proceedings of the 2004 workshop on Multimedia and security, pp. 68–74. ACM Press, New York, NY, USA (2004). DOI http://doi.acm.org/10.1145/1022431.1022445
[13] Ohbuchi, R., Masuda, H., Aono, M.: Watermaking three-dimensional polygonal models. In: MULTIMEDIA '97: Proceedings of the fifth ACM international conference on Multimedia, pp. 261–272. ACM Press, New York, NY, USA (1997). DOI http://doi.acm.org/10.1145/266180.266377
[14] Ohbuchi, R., Mukaiyama, A., Takahashi, S.: Surfaces, shapes, and compression: A frequency-domain approach to watermarking 3d shapes. Computer Graphics Forum **21**(3), 373–382 (2002)
[15] Ohbuchi, R., Mukaiyama, A., Takahashi, S.: Watermarking a 3d shape model defined as a point set. In: CW'04: Third International Conference on Cyberworlds 2004, pp. 392–399 (2004)
[16] Ohbuchi, R., Takahashi, S., Miyazawa, T., Mukaiyama, A.: Watermarking 3d polygonal meshes in the mesh spectral domain. In: GRIN'01: No description on Graphics interface 2001, pp. 9–17. Canadian Information Processing Society, Toronto, Ont., Canada, Canada (2001)
[17] Wagner, M.G.: Robust watermarking of polygonal meshes. In: Proceedings of Geometric Modeling and Processing, pp. 201–208 (2000)
[18] Wang, C.M., Cheng, Y.M.: An efficient information hiding algorithm for polygon models. Computer Graphics Forum **24**(3), 591–600 (2005)
[19] Wang, C.M., Wang, P.C.: Data hiding approach for point-sampled geometry. IEICE Transactions on Communications **E88-B**(1), 190–194 (2005)
[20] Wang, C.M., Wang, P.C.: Steganography on point-sampled geometry. Computers & Graphics **30**(2), 244–254 (2006)
[21] Wu, H.T., ming Cheung, Y.: A reversible data hiding approach to mesh authentication pp. 774–777 (2005). DOI http://dx.doi.org/10.1109/WI.2005.11
[22] Yu, Z., Ip, H.H.S., Kwok, L.F.: A robust watermarking scheme for 3d triangular mesh models. Pattern Recognition **36**(11), 2603–2614 (2003)
[23] Zafeiriou, S., Tefas, A., Pitas, I.: Blind robust watermarking schemes for copyright protection of 3d mesh objects. IEEE Transactions on Visualization and Computer Graphics **11**(5), 596–607 (2005)