2013 10th International Multi-Conference
on Systems, Signals & Devices (SSD)
Hammamet, Tunisia, March 18-21, 2013

Data embedding of 3D triangular mesh models using ordered ring facets

Naoufel Werghi
Department of Electrical and Computer Engineering
Khalifa University
Sharjah, UAE
Naoufel.Werghi@kustar.ac.ae

Nassima Medimegh, Sami Gazzah
SAGE, Advanced Systems in Electrical Engineering
National Engineering School of Sousse(ENISO),
University of Sousse, Tunisia
{Medimegh_Nassima,sami_gazzah}@yahoo.fr

***Abstract*—In this paper we propose a method for embedding binary data in triangular mesh models. Contrary to digital images and audio which benefit from the intrinsically ordered structure of the matrix and the array. 3D triangular mesh model lacks this capital property even though it can be encoded in an array date structure. Such a lack often complicates the different aspects of data embedding, like model traversal, data insertion, and synchronization. We address this problem with a mesh data representation which encodes the mesh data into a novel ordered structure, dubbed, the Ordered Rings Facets (ORF). This structure is composed of concentric rings in which the triangles are arranged in a circular fashion. This representation exhibits several interesting features that include a systematic traversing of the whole mesh model, simple mechanisms for avoiding the causality problem, and an efficient computation of the embedding distortion. Our method can be also adapted to different scenarios of data embedding, which includes stenography and fragile watermarking.**

I. Introduction

Data embedding refers to the process of inserting data into digital contents such as images, videos and 3D models. This process is referred to by different terms depending on the context, the scope and the aim of the data insertion. For example, for copyright protection, we refer to it by digital watermarking. Here the embedded data, called also the watermark, is meant to prove the ownership of the digital content and prevent its illegal use. Moreover, it is designed to resist and survive against malicious alterations (attacks). In such scenarios, the watermarking is labelled by the term robust. The term fragile watermarking is employed when the goal is to protect the integrity of the digital content from an unauthorized processing and to detect an eventual local or global manipulation. In Fragile watermarking data is embedded globally (e.g. across the whole model surface), whereas in robust watermarking data is often locally embedded. Data embedding falls under the stenography scope when the digital content is used as medium for carrying hidden information.

Within each of these applications, data embedding can be applied in the spatial or the spectral domain. In the former, the data is embedded by altering locally or globally the geometry or the topology of the model surface. Whereas the latter involves the modification of a certain components of a spectral transform coefficients. Data embedding can also be qualified to be blind or non-blind, depending on whether or not the original digital content is required to extract the embedded data.

With the recent advances of 3D shape acquisition and the CAD technology and the rapid evolution of network bandwidths, data embedding witnessed a new trend towards the use of 3D triangular mesh models. This fueled the need for new approaches and paradigms that can address the problematic aspects of embedding 3D models, and 3D triangular mesh models in particular.

A triangular mesh is a group of connected triangular facets that encode a given surface in terms of geometry and connectivity. The geometry defines the location of the triangular facets' vertices in the Euclidean space. Connectivity defines the sets of vertices that are connected to form the triangles or the facets of the mesh. Each triangular facet is referenced by an index value that points to the three vertices bounding the triangle. As has been mentioned by Wang et al [13], there is no simple robust intrinsic ordering the mesh elements, e.g. facets and the vertices, which often constitute the carrier of the embedded data. Some intuitive orders, such as the order of the vertices and facets in the mesh file, and the order of vertices obtained by ranking their projections on an axis of the objective Cartesian coordinate system, are easy to be altered.

To address this lack of order and its consequent issues in data embedding, we propose a novel paradigm, dubbed the Ordered Ring Facets. The paradigm has been firstly introduced in [1], [2] in the context of 3D facial surface analysis. In this paper we showcase the adaptability of this paradigm to data embedding for 3D mesh models and its distinguished features.

The rest of the paper will be organized as follows:Section II reviews part of the literature work. As our method deals with global data embedding we will restrict our review to fragile watermarking methods. We refer the reader to [3] for a more general and complete survey. Section III describes the ORF concept and the related algorithms. Section IV exposes the different aspects of our ORF-based data embedding method. Some experimental results are presented and discussed in Section V. Finally we draw some concluding remarks in Section VI.

978-1-4673-6457-7/13/$31.00 ©2013 IEEE

II. Data embedding: State of the art

Yeung and Yeo [4], [5] pioneered fragile watermarking of3D models for verification purposes by extending a 2D image watermarking to 3D. They proposed the idea of moving the vertices to new positions so that each vertex would have the same value for two different and predefined hash functions. Attacks can then be revealed by the presence of vertices that do not comply with aforementioned condition. In this method the hash function requires a pre-defined order of the vertices within the 1-ring neighborhood, otherwise the scheme becomes vulnerable to the causality problem. Ohbuchi et al [6] method imbeds the data on a facet quadruples (a facet and its three adjacents) across the whole mesh. The quadruple facets must satisfy similarity conditions, dubbed, Triangle Similarity Quadruple (TSQ), that is used to recall them when the embedded information is retrieved. Each quadruple stores four symbols composed of marker, subscript and two information data. These are embedded in the dimensionless features of the triangles (e.g. edge ratios), modifying the vertices' positions. To avoid the causality problem the facet quadruples should not be connected to each other.

Lin et al [7] approached the causality problem by proposing a rearrangement of the pixels harmless to the embedded watermark and making the two hash functions depending only on of the position of the current vertex. Chou et al [8] proposed a watermarking mechanism in which one of the hash functions is dependent on the mean of the 1-ring vertex neighborhood. This mean is kept stable after watermarking by adjusting a vertex associated to the watermarked vertex.

High capacity steganographic methods, where the integrity of the hidden data is a requirement, can be also classified as fragile. In these methods too, vertices are altered to embed data bits. The larger the number of bits, the higher will be the capacity of the method. Cayre and Macq [9] proposed a two-stage blind method where, at first they select a candidate stripe of triangles, and then perform a bit embedding by projecting a triangle summit on the opposite edge segmented into two equal intervals. A facet is assigned the bit 0 or 1 depending on in which segment the projection occurred. The synchronization used some local (e.g. largest facet) or global (e.g. facet intersecting the largest principal axes) geometrical features. Boris [10] proposed a blind watermarking method that locally embeds a string of bits on a set of vertices selected and ordered based on a certain distortion visibility criterion. The vertices associated to 0 (respectively 1) are shifted outside (respectively inside) a bounding volume. He proposed two variants, in the first one, the bounding volume is an ellipsoid defined by the principal axes of the covariance matrix computed over the set 1-ring neighborhood, whereas in the second, bounded parallel planes are used. Here the vertex is moved along or opposite to the plane's normal depending on the bit value assigned to it.

A fragile method acting on the mesh connectivity, dubbed Triangle Strip Peeling Symbol Sequence, was also introduced by Ohbuchi et al in [6]. The method consists in cutting out a stripe from the mesh except the attaching edge that marks the start of the stripe. The stripe is formed by repeatedly appending adjacent facets through a path encoded in the message data. The stripe can be shaped as a meaningful pattern that becomes visible when the mesh undergoes global connectivity alteration. However, in this method the watermark cannot spread over the whole mesh, and therefore it cannot be employed for integrity authentication.

In the frequency space, geometrical wavelet-transform has been an attractive tool. Here the watermark is inserted by altering the wavelet-transform coefficients computed at each facet or by altering the facets at a given wavelet-transform resolution to equate a predefined function. Cho et al [12] followed the latter paradigm by embedding the watermark data in facets of the lower resolution of the wavelet transform. This method suffers, however, from the causality problem. The method of Wang et al [13] alters rather the module and the orientation of the one-level WT coefecients to keep a same watermark symbol across the whole facets. This scheme has been also extended to multi-resolution levels in [14].

III. The ORF reprsentation

The ORF representation is a structure in which triangular facets are arranged and ordered in a sequence of concentric rings that emanates from a root or seed facet. This representation has been inspired from the observation of the arrangement of triangular facets lying on a closed contour of edges (Figure 1.a). We classify these facets into two groups: 1) the $Fout$ facets, comprising facets having an edge on the contour and pointing outside the area delimited by the contour, and 2) the $Fgap$ facets, comprising facets having a vertex on the contour and that point inside the area delimited by the contour. We also notice that the $Fgap$ facets seem to fill the gaps between the $Fout$ facets. Together, these two groups form a kind of ring structure. From this ring, we can derive a new group of $Fout$ facets that are one-to-one adjacent with their $Fgap$ facets. These new $Fout$ facets will, in their turn, form the basis of the subsequent rings (Figure 1.b). By iterating this process, we obtain a group of concentric rings. These rings can be centered on a specific facet by setting the closed ring to the edges of that facet (Figure 1.c). Moreover, the sequence of facets across each rings can be ordered clock-wise or anti-clockwise (Figure 1.d)

The algorithm for constructing the ORF rings is as follows:

Algorithm ORF_Rings
Rings ← ConcentricRings(Fin_root, Fout_root)
Rings ← [ ]; Fgap ←Fin_root ; Fout ← Fout_root
For i = 1:NumberOfRings
(Ring, NewFout, NewFgap) ← GetRing(Fout, Fgap)
Append Ring to Rings
Fout ← NewFout
Fgap ← NewFgap
End For
End ORF_Rings

The algorithm *ORF_Rings* has a computational complexity

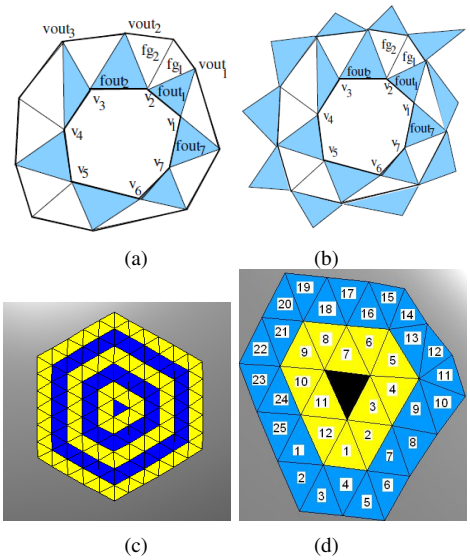

Fig. 1. a: $Fout$ facets (dark) on the contour E_7 : $(v_1, v_2, ., v_7)$. The $Fgap$ facets (clear) bridge the gap between pairs of consecutive $Fout$ facets. b: Extraction of the new $Fout$ facets. Notice that the new $Fout$ facets are one-to-one adjacent to the $Fgap$ facets. c: An example of a 5-ring ORF. d: facets in each ORF ring can be arranged clockwise or anti-clockwise.

of $O(n)$ where n is the number of facets in the rings. The function $GetRing$ extracts the sequences of $Fgap$ facets across the pairs of consecutive $Fout$ facets, constructs the new ring and derives the $Fout$ facets for the subsequent ring. The circular arrangement within one ring implicitly produces a spiral-wise ordering of the facets across the concentric rings.

The algorithm of $GetRing$ is as follows:

Procedure GetRing

(Ring,NewFout,NewFgap) ← GetRing(Fout,Fgap)

NewFout ← [ ] ; NewFgap ← [ ]

For each pair $(fout_i, fout_{(i+1)\%n})$, i = 1..n

Append $fout_i$ to Ring

$(Fgap_i, NewFout_i)$ ← Bridge $(fout_i; fin_i, fout_{(i+1)\%n})$

Append $Fgap_i$ to Ring

Append $Fgap_i$ to NewFgap

Append $NewFout_i$ to NewFout

End for

End GetRing

The function *Bridge* extracts a circle-wise ordered sequence of $Fgap$ adjacent facets and bridges the gap between a pair of consecutive $Fout$ facets. Its algorithm is as follows:

IV. Data embedding

In this section we will describe the data embedding process in 3D mesh model employing the ORF structure and we will discuss its features and properties with respect to different data embedding scenarios. In the rest of the paper will refer to the embedded data by the term ”payload” employed in stenography context.

A. Selection of the root facet

Usually, The choice of the first triangle is based on either local properties, e.g. the largest/smallest facet areas or global properties, e.g. the facet intersecting the major axis of the object principal axes. The method we propose exploits the ORF structure to set a secure starting facet that can be identified using a secret key, which we refer to by key 0. The key is a specific sequence of bits embedded in the 2nd and the 4th ring of the ORF. Therefore any facet can serve as a first key, provided it is marked by the secret key with which it can be identified. This secret key will be then part of the embedded data.

B. ORF bit embedding

The data embedding process uniformly spread the payload across the 3D mesh model by following the traversal path implicitly defined in the ORF rings. Indeed, the facets within the ORF rings are arranged in a kind of a spiral path that spans the whole model starting at the first facet. This ordering aspect allows data embedding in different ways depending on the nature of the application. For example, for model integrity verification, the payload can be inserted at equally spaced and ordered locations within this path. Integrity checking can then be easily and efficiently performed by parsing the path, retrieving and checking the payload at each specific location. For stenography applications, where the 3D mesh model is used to hide a secret data, the payload can also be spread across the whole path. Moreover, the ordered structures of the ORF allows different encrypting scenarios. In this paper, we will showcase a data embedding process within this latter scope.

Figure 2 depicts the different stages of the embedding process, illustrated over a sphere mesh model. First the ORF rings are extracted from the mesh model then they are sampled (sampling parameter τ). The sampling aims to avoid having adjacent rings. Afterwards the rings' indexes are scrambled. This ring sampling and scrambling compose the the first layer of encryption. Their related parameters (e.g τ and the scrambling code are stored in the encryption key labeled key 1. Afterwards a second layer of encryption is applied using a circular shifting of the facets in each ring followed by a sampling of order ρ. Each ring will have its own shifting value. The groups of shift values and the sampling parameter ρ constitute the third encryption key(key 2), to be used, together with the key 0 and key 1, for retrieving the embedded data. Assuming a reasonably tessellated mesh, the set of facets that come out from stage 2 are uniformly spread over the mesh model surface. In the last stage the payload is embedded in this set at the cadence of of one bit per facet. Here we employed the embedding technique of Cayre and Macq [9] (Figure 3), but other bit embedding techniques can be used as well. In Cayre and Macq method, basically, one vertex is projected into its opposite edge, which is divided in two equal intervals, coded 1 and 0. If the projection falls in the interval that matches the embedded bit (for instance, it falls in the interval 1 and the bit to be embedded is 1), then the vertex is kept unchanged.

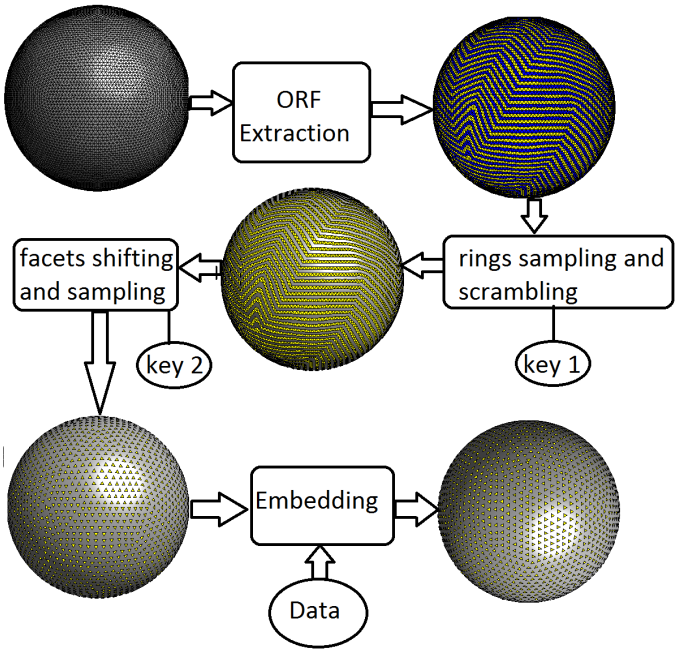

Fig. 2. : Data embedding process: After setting the root facet location, the ORF rings are extracted, sampled (sampling parameter τ), then their order is scrambled. The parameters of this first stage (e.g. τ and the scrambling code are stored in key 1). In the next stage, facets in each ring are shifted then sampled (sampling parameter ρ). Parameters of these stage are stored in key 2. The outcome of these stages are a sequence of facets spread over the whole mesh, and which locations can be retrieved using key 0, key 1 and key 2. In the last stage the payload embedded in that sequence of facets.

Otherwise it is shifted collinearly with that edge, and within the facet's plane, to a new location for which the projection meets the good match.

The sampling performed in stages 1 and 2, aim to ensure isolation between the tampered, that carry the payload, so that they cannot share an edge or a vertex. The appropriate values of τ and ρ depend on the regularity of the mesh. For a uniform mesh, $\tau = 2$ and $\rho = 3$ can guarantee compliance with the aforementioned condition. The uniform sphere mesh model depicted in Figure 2 is an example of this case. When the mesh exhibits some irregularities, larger values might be needed.

C. Data extraction

The data extraction process is virtually similar to the embedding process. The starting facet should be firstly detected. Here the key 0 is used to inspect the four rings around the candidate facet. Afterwards, the rest of the ORF are extracted then browsed in the order and at the sampling rate encoded in Key 1. Then the facets in each ring are parsed according the parameters encoded in key 2.

D. Causality problem

Our method offers control mechanism allowing data embedding free from the causality problem. Such a problem happens when an anterior tampered facet (e.g. already carried a bit) is to be affected by a posterior one (e.g a newly tampered). This might corrupt the content of the former facet. Usually

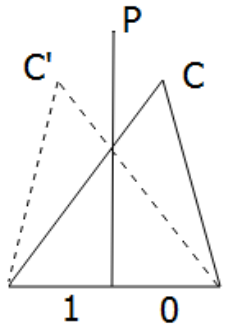

Fig. 3. : Bit embedding method of Cayre and Macq [9]. In this example, a bit 1 is embedded. The vertex C is projected on its opposite edge, divided into equal segments labels '0' and '1'. If the vertex falls on a segment which label does not math the data bit (which is the case in this example). The vertex is moved, collinearly to the opposite edge, within the facet's plane and symmetrically to the plane P so that the project of its new location C' falls on the segment that meets the correct match.

this problem occurs between neighbouring facets. The size of the neighbourhood depends on the embedding technique and on how many facets surrounding the tampered one can be affected. This problem is addressed by inserting a flag data in or around the tampered facets so that these will not embedded again when revisited during the mesh traversing. In our method, a proper setting of the sampling parameters τ and ρ ensures sufficient spaces between the tampered facets preventing any mutual alteration, avoiding therefore the need for flagging and checking procedures. In the technique of Cayre and Macq [9], which we have used, only facets sharing the displaced vertex of the tampered facet are affected. Therefore, for a regular mesh, setting the sampling parameters τ and ρ to 2 and 3, respectively, any triangles connected to the tampered facet, either by an edge or a vertex, cannot be the subsequent tampered one. However, depending on the state of the mesh, τ and ρ might need to be increased.

E. Capacity

The total number of facets N can be expressed by $N = \sum_{i=1}^{M} m_i$, where M is the number of ORF rings and m_i is the number of facets in the i^{th} ring. Considering the sampling parameters τ and ρ. The number of facets can be expressed by $F = \sum_{k=1}^{M/\tau} m_{\tau k}/\rho$. The number of tampered facets can be evaluated in average to $N/(\tau \times \rho)$ bits. Assuming a uniform mesh in which the facets have close area and edge values, $\tau =$ and ρ can be set to 2 and 3, respectively, the number of embedded facets can reach $N/6$.

F. Security

Accessing the payload requires addressing three challenges in our algorithm, namely, finding the root facet, finding the ring sampling rate τ and the correct combination of ring order, and finally finding the facet sampling rate ρ and the facet shifting value at each ring. These three cascading stages of encryption give our scheme a high level of security. Exhaustive search is virtually out of reach, indeed the number of encryption

combination in our scheme is evaluated to

$$N \times M! \times n_\tau \times \prod_{k=1}^{M/\tau} m_{\tau k} \times n_\rho \quad (1)$$

Where N is the number of facets in the mesh, M is the number of rings, n_τ and n_ρ are the numbers of possible values of the sampling parameters τ and ρ, and m_i is the number of facets in the i^{th} ring. Such a number makes the extraction of the payload without the encryption code nearly impossible.

G. Mesh distortion

For mesh models with a single topology, Haussdorff distance has been adopted as an objective metric for evaluating the mesh distortion inferred by the data embedding. The computation of the Haussdorff distance is time demanding because of its quadratic complexity ($O(n^2)$). In our method, the computation of the mesh distortion is brought down to a linear complexity $O(n)$. In effect, the linear traversal path embedded in the ORF structure allows a one-to-one mapping between the original model and the embedded model facets. Based on this mapping, we can simply express the mesh distortion with the following formula

$$\sum_{i=1}^{N} \frac{\sum_{j=1}^{3} min_{1\leq k\leq 3}(u_i^j - v_i^k)}{3\bar{u}_i}, \quad (2)$$

where u and v form the pair of corresponding facet edges in the original and embedded model respectively, $\bar{u}$ is the mean of facet edge in the original model, and N is the number of facets

H. Integrity checking

As the embedding is uniformly spread over the whole model, our method can be used for checking the integrity of the mesh model by comparing the extracted payload (formatted as a string of bits),and matching it with its reference counterpart. A mismatch indicates that the model has been altered. The indexes of the mismatches in the string of bits can be easily mapped to the facets' indexes, and thus used to trace the locations of the attacked areas. Moreover, the global mesh alteration can be evaluated using the following simple formula

$$\sum_{i=1}^{M} \delta_i, \quad \delta_i = \begin{cases} 0 & \text{if } \Omega_i = \Upsilon_i \\ 1 & \text{otherwise} \end{cases}$$

where Ω and Υ are the original and the retrieved strings and M is their length.

I. Robustness

Being dedicated to stenography applications, the method is robust to affine and similarity transforms, namely, translation, rotation, uniform scaling, and affine transforms. It is also resistant to vertices and facets reordering. It is not robust against geometrical or topological transformation such as cropping, simplification, and mesh resembling. These kind of attacks corrupt the message content.

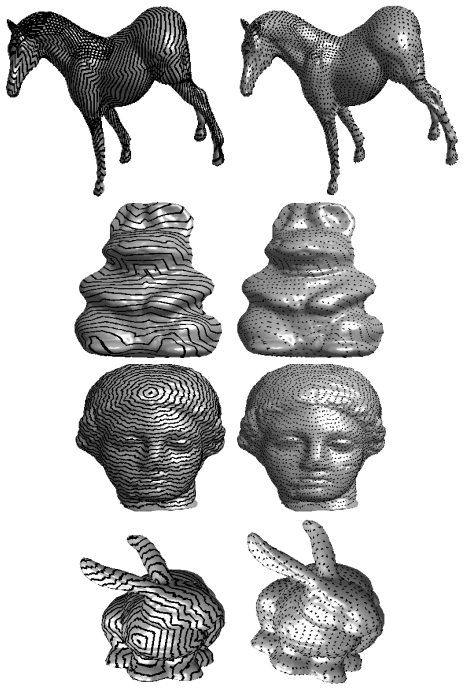

Fig. 4. ORF rings and the tampered facets displayed on the models

Model	facets	vertices	τ	ρ	capacity	distortion	time (s)
bunny	69666	34835	6	8	3201	0.90	6.85
rabbit	141312	70658	4	7	5121	1.65	8.89
venus	201514	100759	3	9	9016	0.56	13.13
horse	225080	112 642	3	5	15845	3.35	13.48

TABLE I
RESULTS OBTAINED FOR THE TESTED MODELS.

V. EXPERIMENTS

We applied on several mesh models collected from the 3D mesh watermarking benchmark [15]. The models are bunny, horse, rabbit, and venus. We used a PC-based 2.93 GHz, 8 GB RAM. The implementation was performed under Matlab. Table 1 depicts the list of the models, their corresponding sampling parameters, and the related performance measures. The embedding capacity represents the number of bits embedded in the model, which is also equal to the number of tampered triangles, since the embedding technique inserts one bit per triangle. The distortion is computed using equation 2. The different models show a capacity in the range of 7%-14%. However this could be further increased by inspecting further each ORF ring to locate other free facets. The distortion shows quite low values. It seems inversely proportional with the sampling parameters. This is expected as the number of tampered facets increases as the sampling decreases. The processing time seems to increase at a fair rate because of the linear complexity of our method. This aspect is confirmed in another experiment, performed with the horse model, in which we computed the embedding time for increasing sizes of the payload (Fig. 5). The growth rate of the processing time confirms clearly the linear complexity property of our method.

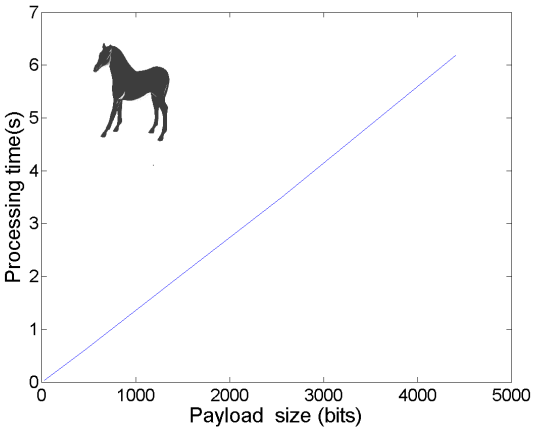

Fig. 5. Processing time evolution for embedding 4408 bits in the horse model.

VI. Conclusion

In the paper, we have showcased a new paradigm for embedding data in 3D triangular mesh model, based on a structured and ordered arrangement of the triangular facets. Compared to other global embedding methods, our method is distinguished by an automatic traversal of the mesh model, an immunity mechanism against causality problem, a linear complexity of the embedding process, and a high encrypting power. Our paradigm is robust against translation, rotation, scaling, vertex and facets reordering. In addition to stenography, our paradigm can be easily adapted to fragile watermarking and authentication. However it is not meant for robust watermarking.

The embedding capacity is one of the aspects that deserves some improvements. At the current state, our paradigm ensures an embedding automatically free of the causality problem, but seemingly at the expense of the embedding capacity. We plan to further investigate this issue so that we can reach a better compromise. One technique which we are currently studying is replacing the fixed-rate sampling at the ring level, with a kind of adaptative sampling whereby the sampling step is adjusted according to the facet's neighbourhood in the ring. Finally our approach is not qualified to deal with the genius 2 models, exhibiting holes and gaps.

References

[1] N. Werghi, M. Rahayem, J. Kjellander, "An ordered topological representation of 3D triangular mesh facial surface: Concept and applications", EURASIP Journal on Advances in Signal Processing 2012, pp.144

[2] N. Werghi, M. K. Naqbi, "A novel surface pattern for 3D facial surface encoding and alignment", Proc. IEEE SMC 2011, pp.902-908.

[3] K. Wang, G. Lavoue, F. Denis, A. Baskurt, "A Comprehensive Survey on Three-Dimensional Mesh Watermarking", IEEE Transactions on Multimedia, vol.10, no.8, pp. 1513-1527.

[4] M. M. Yeung, B.L. Yeo, Fragile watermarking of three dimensional objects, Proc. 1998 IEEE Int. Conf. Image Processing, ICIP98, vol. 2, pp. 442-446, 1998.

[5] B. Yeo and M. M. Yeung, "Watermarking 3D objects for verification," IEEE Computer Graphics and Applications, Vol. 19, no. 1, pp. 36-45, Jan.-Feb. 1999.

[6] R. Ohbuchi, H. Masuda, and M. Aono, "Data embedding algorithms for geometrical and non-geometrical targets in three-dimensional polygonal models", Computer Communications archive vol. 21, no. 15, pp. 1344-1354, October, 1998.

[7] H. S. Lin, H. M. Liao, C. Lu, and J. Lin, "Fragile watermarking for authenticating 3-D polygonal meshes," IEEE Transactions on Multimedia, vol. 7, no. 6, pp. 997-1006, Dec. 2005.

[8] C. M. Chou and D. C. Tseng, "A public fragile watermarking scheme for 3D model authentication," Computer-Aided Design, vol. 38, no. 11, pp. 1154-1165, Novembre, 2006.

[9] F. Cayre and B. Macq, "Data hiding on 3-D triangle meshes," IEEE Transactions on Signal Processing, vol. 51, no. 4, pp. 939-949, April, 2003.

[10] A. G. Bors, "Watermarking mesh-based representations of 3-D objects using local moments," IEEE Transactions on Image Processing, vol. 15, no. 3, pp. 687-701, Mar. 2006.

[11] Y. M. Cheung and C. M. Wang, "A high-capacity steganographic approach for 3D polygonal meshes," The Visual Computer,vol. 22, no. 9-11, pp. 845-855, Sep. 2006.

[12] W. H. Cho, M. E. Lee, H. Lim, and S. Y. Park, Watermarking technique for authentication of 3-D polygonal meshes, Proc. of the International Workshop on Digital Watermarking05, 2005, pp. 259270.

[13] K. Wang, G. Lavoue, F. Denis, and A. Baskurt, A fragile watermarking scheme for authentication of semi-regular meshes, in Proc. of the Eurographics Short Papers08, 2008.

[14] K. Wang, G. Lavoue, F. Denis, and A. Baskurt, Hierarchical blind watermarking of 3D triangular meshes, in Proc. of the IEEE International Conference on Multimedia and Expo07, 2007, pp. 12351238.

[15] K. Wang and G. Lavoue, F. Denis, A. Baskurt, X. He, "A benchmark for 3D mesh watermarking", Proc. of the IEEE International Conference on Shape Modeling and Applications, 2010, pp.231-235