16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

LOSSLESS DATA HIDING METHOD BASED ON MST AND TOPOLOGY CHANGES OF 3D TRIANGULAR MESH

P. Amat [a,b], W. Puech [a], S. Druon [a] and J.P. Pedeboy [b]

[a] LIRMM Laboratory, UMR CNRS 5506, University of Montpellier II
161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE
[b] Stratégies S.A., 41-43 rue de Villeneuve, Parc d'affaires SILIC - BP 80429
94583 RUNGIS CEDEX, FRANCE
philippe.amat@lirmm.fr, william.puech@lirmm.fr, sebastien.druon@lirmm.fr, jp.pedeboy@cadwin.com

ABSTRACT

For the last five years or so, several 3D data hiding algorithms have been developed. Few of them proposed methods to hide data in 3D objects without changing the position of vertices in the 3D space. In this paper we propose a new approach of data hiding in 3D objects based on the minimum spanning tree (MST). This method is lossless in the sense that the positions of the vertices are unchanged. The method is blind and does not depend on the ordering of the data in the files. Moreover the embedding capacity is significant and can be as high as 2.5% of the size of the 3D object.

1. INTRODUCTION

Digital data is expanding phenomenally on the Internet with each passing day, thus hindering fast transmission. In case of 3D objects, the problem of transmission is exacerbated by the issue of security that is vital for some applications, e.g., confidential transmission, video surveillance, military and medical applications. In this paper we present a new approach of data hiding in 3D objects without changing the position of vertices in the 3D space. This method is lossless in the sense that the positions of the vertices are unchanged.

3D data hiding algorithms can be classified as a function of the approach used to embed the data. The first class is the *Data file organization* whereby the information is encoded by modifying the organization of the data in the computer file associated with the 3D object. These methods preserves the original data model. Ichikawa *et al.* propose to modify the order of triangles within a list of triangles or the order of the triplet of vertices forming a given triangle [5]. Examples of other methods which employ the data file organization are given in [9].The second class is based on the *Topological features*. This class uses the topology or connectivity to embed the information in the 3D model. The geometry of the mesh (positions of the vertices) is not modified. Ohbuchi *et al.* applied the *Triangle Strip Peeling Symbol Sequence* to public watermarking scheme based on a topological embedding primitive. It employs, as its embedding primitive, an adjacency of a pair of triangles in a triangle strip, each of which encodes a bit of information. One-dimensional arrangement of embedding primitives is induced by the adjacency of triangles on the triangle strip. To recognize the triangle strip with watermark, the strip is peeled off from the original mesh [10]. Mao *et al.* add new triangles in the original triangles mesh and used the positions of the vertices of the new triangle to embed the watermark [8]. The third class is based on *Geometric data*. The algorithms of this class are based on the modifications performed on the geometric data of the 3D object. This class can be partitioned as a function of the application domain: the spatial or transformed domains. In the spatial domain, Benedens proposes an algorithm called *Vertex Flood* based on modifications of the radial distance of vertices from a pre-defined starting point of the mesh [2]. Ohbuchi *et al.* proposed two algorithms, the *Triangle Similarity Quadruple* (TSQ) and the *Tetrahedral Volume Ratio* (TVR), by embedding data in geometric domain. TSQ modifies ratios between triangle edge lengths or triangle height and base lengths. TVR uses the ratio between volume of an initial and modified tetrahedron. This volume is given by an edge and its two incident triangles. These ratios are slightly modified to embed the watermark. Examples of other methods which works in the spatial domain are given in [7, 3]. In transformed domains, Ohbuchi *et al.* proposed a spectral decomposition method [11]. The spectral domain is obtained after diagonalizing the combinatorial Laplacian matrix of the mesh. The watermarking algorithm embeds information after an eigen-decomposition of the patches of the mesh. Kanai *et al.* use the wavelet transform domain in their method [6] and the watermark bits are embedded by modifying the least significant bits of the wavelet coefficients. In the wavelet domain, Wang *et al.* proposed a hierarchical watermarking scheme wherein the cover mesh is re-meshed to generate a semi-regular mesh and multiple watermarks are inserted in this semi-regular mesh [12].

The approach of data hiding presented in this article belongs to the *Topological data* category. This method has the constraint of not changing the position of vertices in the 3D space. A previous method has been presented in [1]. Even if the constraints are similar in the two methods, the selection of the embedding areas and the data embedding step are very different. Moreover, by using the MST, the synchronization is much better than the previous proposed method. In Section 2, we develop our new method based on the MST. Section 3 provides a set of simulation results and compares it with a previous method [1], while Section 4 concludes this paper.

2. THE PROPOSED METHOD BASED ON THE MST

In this section we detail a new method of data hiding in 3D objects. This method is based on a 3D model represented by a cloud of vertices, as illustrated in Figure 1, and a list of edges corresponding to the triangular mesh of the surface.

The main idea of this method is to find and to synchronize particular areas by scanning the minimum spanning tree (MST). These areas are used to embed the message. The data hiding relies on the modification of the topology of edges

Figure 1: Cloud of initial vertices (43039 vertices).

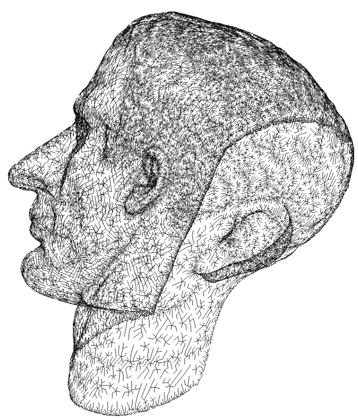


Figure 2: MST from the cloud of vertices Figure1.

in the chosen areas. These modifications have the effect of changing the structure of triangles built in these areas. This method has the advantage of not changing the position of the vertices of the 3D model. This invariant on the vertex positions allows us to carry out an embedding that is robust to the affine transformations such rotations, translations or zooms.

This data hiding method is based on the MST. An example of MST is illustrated in Figure 2. We call quadruple an area formed by two triangles of mesh having a common edge. This method is made up of three parts. The first part is the selection of the areas for the embedding of the data. These areas are chosen according to some constraints. The second part is the synchronization of the message with the 3D model. This second part depends on the MST. The third step is the embedding of the message. For that, the topology of the triangles of the chosen quadruple is changed. Figure 3 illustrates an overview of the proposed method.

2.1 Selection of the embedding areas

The first step of our algorithm is the building of the MST. The MST is based on the euclidean distances among all the vertices in the 3D object. For each 3D object, the MST is unique and undirected. For traversing the graph we need to choose a start node. This node of the MST is chosen by a secret key. This key is constructed in the coordinate system

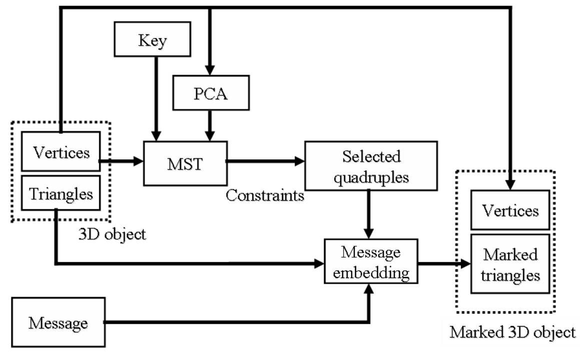


Figure 3: Overview of the proposed data hiding method based on the MST.

of the object acquired from a principal component analysis (PCA). After constructing the MST, we scan it for searching and selecting the quadruples to be used in the embedding step. A quadruple is a four-node subtree of the MST having either one node together with its parent and two children or one node along with three children. The quadruples will be selected as embedding zones if they comply to the constraints of coplanarity, convexity and overlapping.

- Coplanarity constraint: this constraint of coplanarity of the quadruples is based on the fact that any change in the topology of the two triangles forming one quadruple also changes the angle formed between these triangles. Consequently, using on a non-coplanar quadruple for embedding affects the surface and therefore the visual aspect of the 3D model. In an ideal situation, the message should be embedded only in quadruples which are strictly coplanar. However, the number of quadruples complying strictly to this coplanar criterion is very limited. To increase the embedding capacity, we introduce in our approach a tolerance threshold, S_c, on the coplanarity of the chosen quadruples.
- Convexity constraint: this second constraint imposes that the quadruples used for the data hiding must be convex. For this constraint, we assume that the selected quadruples already obey the coplanarity constraint. The measurement of this constraint is a combination of the two following procedures:
  – Vectors $V_{i,i+1 \bmod 4}$ between vertices P_i et P_{i+1} of the quadruples are calculated for $i \in \{1,...,4\}$. Angles α_i between two successive vectors are then calculated.

    $$Q_{1234} \text{ is selected only if } \alpha_i < 180^\circ \text{ for i } \in \{1,...,4\}. \quad (1)$$

    This allows eliminating all quadruples having obtuse angles or three collinear vertices.
  – The second measure prevents us from having quadruples too close to a triangular form when three out of four vertices are aligned. In that case the data hiding would reveal too disproportional triangles. A threshold of tolerance S_t is chosen so that the proportion of both triangles of one quadruple respects a certain value.
- Overlapping constraint: the third constraint concerns the overlapping of the quadruples. From the quadruples selected with the two previous constraints, there may re-

main quadruples with common edges. The quadruples should have at most one common edge and in that case the quadruples are selected for data hiding. On the contrary if quadruples have more than one common edge, the modification of topology of one of these quadruples during the embedding step, may disturb the second. To avoid this problem, a choice should be made among all the overlapped quadruples to select only one of these. This choice is made in order to keep the quadruple which will affect less the 3D model. Thus, the most coplanar quadruple will be chosen to embed the data. A more detailed presentation of the three constraints could be founded in [1].

2.2 Synchronization of selected quadruples

The synchronization depends on a secret key. This key generates index of the 3D point to be used to start the MST algorithm. This key is constructed in the coordinate system of the object acquired from a principal component analysis (PCA). PCA enables us to make our method robust to transformations such as rotations, translations or zooms. To synchronize the message we scan the MST with the key vertex being the starting point. The position of these points on the axis will give the embedding order of the bits of the message on the quadruples. Figure 4 illustrates a synchronization of three quadruples on a part of the MST.

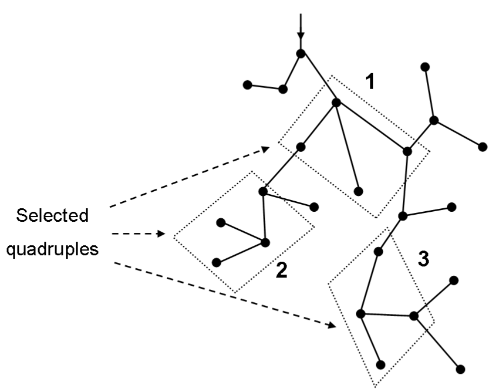
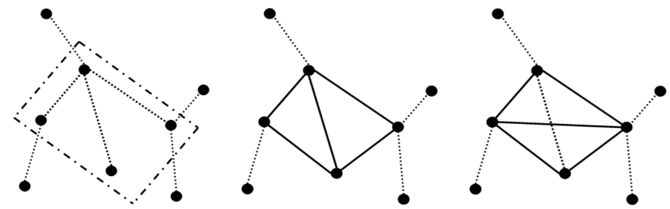


Figure 4: Synchronization of the quadruples.

2.3 Data embedding step

As for synchronization, the data embedding step is also based on the MST. Code 0 is allocated if the initial MST edge corresponds to the common edge of the two triangles forming the quadruple and code 1 is allocated if the initial MST edge does not correspond to the common edge of the quadruple. To change the value of hidden bit it is enough to change the common edge of the two triangles forming the quadruple.

Figure 5 shows an example to hide a 0 bit and a 1 bit in a quadruple. Figure 5.a shows a part of MST with a selected quadruple. The Figure 5.b shows the same quadruple after embedding a 0 bit; the common edge of the two triangles corresponds to the MST edge. Embedding a 1 bit in the quadruple can be done by reversing the common edge of the triangles as illustrated in Figure 5.c.



(a) (b) (c)

Figure 5: a) A part of MST, b) Quadruple hiding one bit to 0, c) Quadruple hiding one bit to 1.

3. SIMULATION RESULTS

In this section we have applied and analyzed the proposed method of data hiding on more than a dozen of various 3D objects. Table 1 shows the characteristics of the analyzed 3D objects. Figure 6 illustrates the triangular mesh of Julius which includes 85145 triangles. In Figures 7 and 8 we can see all the selected quadruples for data hiding with a coplanarity threshold $S_c = 30°$ and $S_c = 10°$, respectively. Figures 9 and 10 represent the mesh after embedding for the two thresholds.

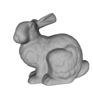
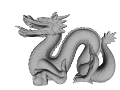
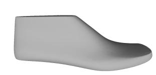
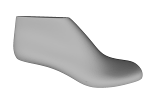
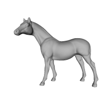
3D Objects	bunny	dragon
Number of vertices	453/1887/8146	5203/22982
Number of triangles	948/3851/16301	11102/47794
		
happy	man's shoes	Chinese dragon
7102/32316	12606	20184
15536/67240	25208	40364
		
woman's shoes	horse	Julius
31088	43039	48485
62168	85145	96966
		

Table 1: Characteristics of the 3D objects.

To illustrate the effects of embedding in quadruples, we applied a zoom on a part of mesh. Figure 11.a shows quadruples before embedding and Figure 11.b shows the same after embedding. If we consider that the selected quadruples are convex and coplanar as represented in Figure 11, then the change of topology performed during the embedding of this quadruple will not going to change the surface of the 3D model.

Three criteria are to be taken into account in the analysis of results. The first criteria is the embedding capacity (in bits per vertex). In Figure 12 we can see the embedding capacity of our method according to the 3D models used with different thresholds. Globally, the embedding capacity slightly

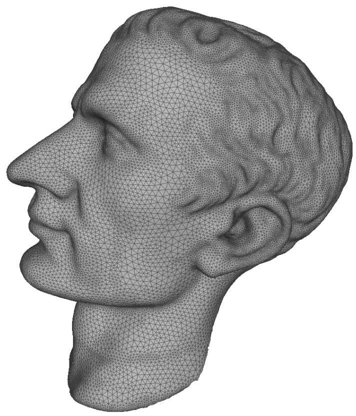


Figure 6: Triangulated mesh (85145 triangles).

Figure 7: Selected quadruples with a threshold $S_c = 30°$ (3974 selected quadruples).

Figure 8: Selected quadruples with a threshold $S_c = 10°$ (3852 selected quadruples).

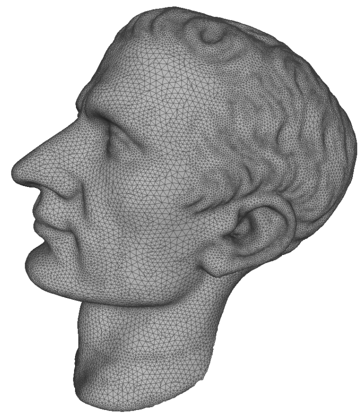


Figure 9: Marked triangulated mesh with $S_c = 30°$(3974 quadruples marked).

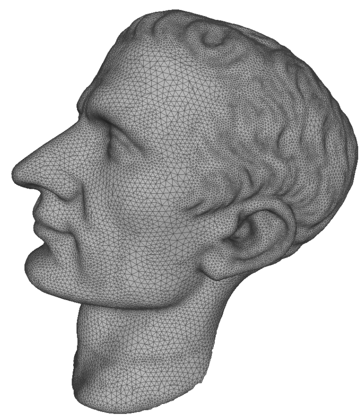
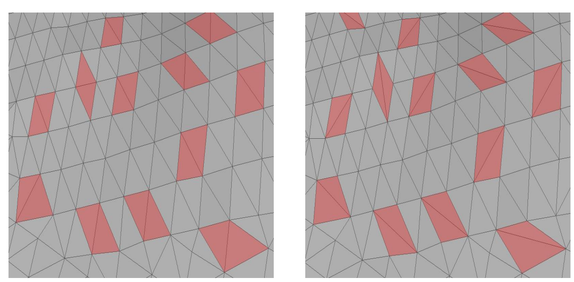


Figure 10: Marked triangulated mesh with $S_c = 10°$ (3852 quadruples marked).



(a) (b)

Figure 11: a) Mesh before the embedding, b) Mesh after the embedding.

increases with the size of the 3D objects. When we increased the coplanarity threshold this difference of capacity between the 3D objects decreased. The increase in the threshold allows us to increase the embedding capacity, as shown in Figure 13 for the average of all the 3D objects. The second criterion of analysis serves for estimating the error made on the surface according to the threshold of coplanarity. Figure 14 shows the maximum Hausdorff error made on the surface according to the chosen threshold of coplanarity for all the 3D models. The Hausdorff error calculation is made with the algorithm proposed by the Metro software [4]. The errors are small because we did not modify vertex positions. The error slightly increases with the coplanarity threshold. The third criterion is the robustness attack analysis. The main aim of our method is to hide data and not to be robust to attacks of

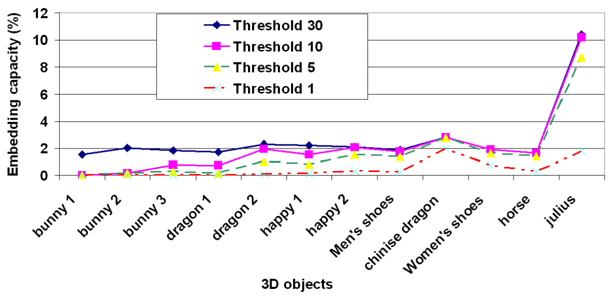


Figure 12: Number of selected quadruples in function of the number of vertices of the 3D model.

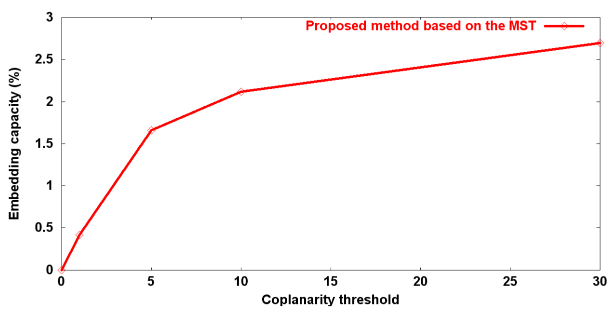


Figure 13: Embedding capacity in function of coplanarity threshold (for the average of the 12 analyzed objects).

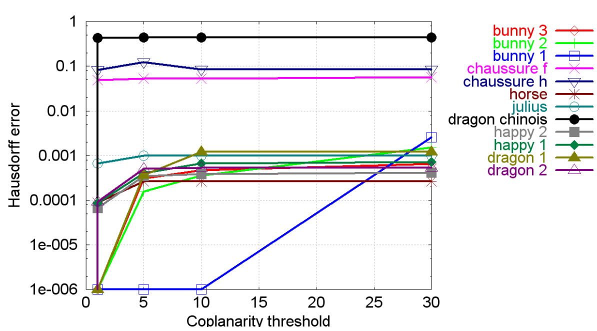


Figure 14: Maximum Hausdorff error in function of coplanarity threshold.

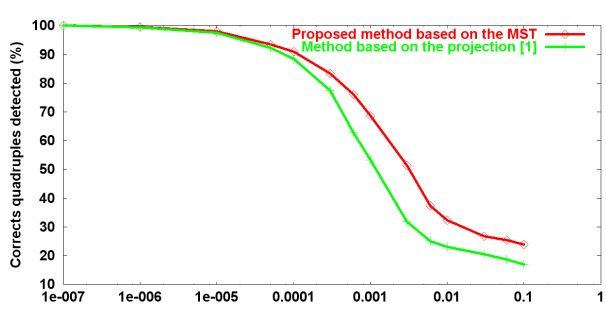


Figure 15: Correct quadruples detected with a threshold $S_c = 10°$ (for the average of the 12 analyzed ob-jets).

the 3D objects. Figure 15 shows resistance of our method to noise addition for the average of the 12 analyzed ob-jets. In Figure 15 one can also observe that this new method is more robust to noise than the previous method [1].

4. CONCLUSION

In this paper we presented a new approach of data hiding in 3D objects without changing the position of vertices in the 3D space. This method is lossless in the sense that the positions of the vertices are unchanged. By using the MST and a secret key it is then possible to synchronize quadruples and to embed data in the selected quadruples. In the simulation results section we applied our method to various 3D objects. As a function of the threshold S_c we showed that the embedding capacity is significant and can reach 2.5% of the size of the 3D objects. The main advantage of this new method is its stability during the extraction of the hidden data. Compared to the previous method [1], this new approach reduces the computational effort and the required computation time. Moreover this new method is more robust than the previous one. In the continuation of this work it would be worthwhile to analyze and to compare the complexity of these two methods.

REFERENCES

[1] P. Amat, W. Puech, S. Druon, and J-P. Pedeboy. A Data Hiding Method Based on the Topology Change of a 3D Triangular Mesh. In *15th European Signal Processing Conference (EUSIPCO 2007)*, pages 2524–2528, Poznan, Poland, 2007.

[2] O. Benedens. Geometry-Based Watermarking of 3D Models. *IEEE Comput. Graph. Appl.*, 19(1):46–55, 1999.

[3] A. G. Bors. Watermarking Mesh-Based Representations of 3-D Objects Using Local Moments. *IEEE Trans. on Image Processing*, 15(3):687–701, march 2006.

[4] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.

[5] S. Ichikawa, H. Chiyama, and K. Akabane. Redundancy in 3D Polygon Models and Its Application to Digital Signature. In *WSCG*, pages 225–232, 2002.

[6] S. Kanai, H. Date, and T. Kishinami. Digital Watermarking for 3D Polygons using Multiresolution Wavelet Decomposition. *Proc. int. Workshop Gemetric Modeling : Fondamentals and Applications*, pages 296–307, Decembre 1998.

[7] B. Macq and F. Cayre. Data Hiding on 3D Triangle Meshes. *IEEE Transactions on Signal Processing*, 51(4):939–949, 2003.

[8] X. Mao, M. Shiba, and A. Imamiya. Watermarking 3D geometric models through triangle subdivision. In *Proc. SPIE Vol. 4314, p. 253-260, Security and Watermarking of Multimedia Contents III*, volume 4314, pages 253–260, August 2001.

[9] R. Ohbuchi and H. Masuda. Managing CAD Data as a Multimedia Data Type Using Digital Watermarking. In *Proc. 4th Workshop on Knowledge Intensive CAD to Knowledge Intensive Engineering*, pages 103–116, Deventer, The Netherlands, 2001.

[10] R. Ohbuchi, H. Masuda, and M. Aono. Watermaking Three-Dimensional Polygonal Models. In *MULTIMEDIA '97: 5th ACM international conference on Multimedia*, pages 261–272, NY, USA, 1997.

[11] R. Ohbuchi, A. Mukaiyama, and S. Takahashi. A Frequency-Domain Approach to Watermarking 3D Shapes. *Computer Graphics Forum*, 21(3):373–382, 2002.

[12] K. Wang, G. Lavoué, F. Denis, and A. Baskurt. Hierarchical Blind Watermarking of 3D Triangular Meshes. In *Proc. of the IEEE International Conference on Multimedia & Expo*, pages 1235–1238, July 2007.