

 Open access • Journal Article • DOI:10.1016/J.IMAGE.2010.05.002

## Lossless 3D steganography based on MST and connectivity modification

— [Source link](#) 

Philippe Amat, William Puech, Sébastien Druon, Jean-Pierre Pedeboy

**Institutions:** University of Montpellier

**Published on:** 01 Jul 2010 - Signal Processing-image Communication (Elsevier)

**Topics:** Information hiding, Minimum spanning tree, Steganography, Lossless compression and Digital watermarking

Related papers:

- [Data hiding on 3-D triangle meshes](#)
- [Pattern based 3D image Steganography](#)
- [Review: Digital image steganography: Survey and analysis of current methods](#)
- [A High Capacity 3D Steganography Algorithm](#)
- [Distortion-Free Steganography for Polygonal Meshes](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/lossless-3d-steganography-based-on-mst-and-connectivity-q5pztrtx7e>



**HAL**  
open science

## Lossless 3D Steganography Based on MST and Connectivity Modification

Philippe Amat, William Puech, Sébastien Druon, Jean-Pierre Pedeboy

► **To cite this version:**

Philippe Amat, William Puech, Sébastien Druon, Jean-Pierre Pedeboy. Lossless 3D Steganography Based on MST and Connectivity Modification. *Signal Processing: Image Communication*, Elsevier, 2010, 25 (6), pp.400-412. lirmm-00600701

**HAL Id: lirmm-00600701**

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00600701>

Submitted on 15 Jun 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution| 4.0 International License

Provided for non-commercial research and education use.  
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Signal Processing: *Image Communication*journal homepage: [www.elsevier.com/locate/image](http://www.elsevier.com/locate/image)

# Lossless 3D steganography based on MST and connectivity modification

P. Amat<sup>a,b</sup>, W. Puech<sup>a,\*</sup>, S. Druon<sup>a</sup>, J.P. Pedeboy<sup>b</sup>

<sup>a</sup> LIRMM Laboratory, UMR CNRS 5506, University of Montpellier II 161, rue Ada, 34392 Montpellier Cedex 05, France

<sup>b</sup> Stratégies S.A., 41-43 rue de Villeneuve, Parc d'affaires SILIC—BP 80429, 94583 Rungis cedex, France

## ARTICLE INFO

### Article history:

Received 16 June 2008

Accepted 1 May 2010

### Keywords:

Data hiding

Steganography

Watermarking

3D triangular mesh

Quadruple

Graph scanning

## ABSTRACT

Data hiding has become increasingly important for many applications, like confidential transmission, video surveillance, military and medical applications. In this paper we present a new approach of 3D object data hiding without changing the position of vertices in the 3D space. The main idea of the proposed method is to find and to synchronize particular areas of the 3D objects used to embed the message. The embedding is carried out by changing the connectivity of edges in the selected areas composed of quadruples. The proposed approach of data hiding in 3D objects is based on minimum spanning tree (MST). This method is lossless in the sense that the positions of the vertices are remaining the same before and after embedding. Moreover the method is blind and does not depend of the order of the data in the files. This approach is very interesting when the 3D objects have been digitalized with high precision.

© 2010 Elsevier B.V. All rights reserved.

## 1. Introduction

With the evolution of computer networks, transmissions involving 3D downloadable objects are becoming commonplace. To be universally readable, these 3D objects must meet international standards even if we want to bring new features to these products. It is, therefore, important to follow standard formats<sup>1</sup> to manipulate these 3D objects. In our work, these new features relate not only to the protection of 3D objects during their transmission and archiving but also the enrichment of these objects with their meta-data without increasing the size of objects and synchronizing the meta-data in a hierarchical manner. The representation mostly used for 3D objects is the surface mesh. This representation, consisting of vertices, edges and faces, is widely supported by all the graphics rendering cards. Many data

hiding methods are based on this representation. Data hiding methods can be divided into two categories, spectral or spatial, depending on the domain used to embed the hidden data. Irrespective of the embedding domain, existing watermarking and steganography methods for 3D objects are usually based on the modification of vertices in the 3D mesh for embedding the data. For many applications, such as medical or industrial manufacturing, the 3D objects are acquired with great precision and accuracy and any change in the position of the vertices may well prove catastrophic. In this paper we propose a data hiding method wherein the embedding is not based on the modification of the position of vertices of the 3D objects. Few of the previous works propose to hide data in 3D objects without changing the position of vertices in the 3D space. The new approach presented in this paper proposes a method to hide data 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 practically as high as 0.128 bit/vertex. The method enables the embedding, in a 3D

\* Corresponding author.

E-mail address: [william.puech@lirmm.fr](mailto:william.puech@lirmm.fr) (W. Puech).

<sup>1</sup> Such as STL, PLY, X3D etc.

object, of meta-data like the author's name, the date of creation of the object, a 2D or 3D logo, color of the object or the texture of the object itself. It should be noted that the method resists geometric transformations such as rotation, translation and uniform scaling. The proposed method, based on a better selection of quadruples, is an improvement of a previous method presented in [3]. We compare our new proposed method with the previous one by showing the improvement in the embedding capacity.

The rest of this paper is divided into the following sections. In Section 2, we present a brief literature survey on different data hiding methods applied to 3D objects. Section 3 explains the method and in Section 4, we present and discuss the results of our method when applied to real 3D objects. Conclusion as well as the perspective are given Section 5.

## 2. Previous work

Depending on the type of application, watermarking methods can be classified according to certain mutually antagonistic constraints. These constraints include perceptual transparency, robustness and embedding capacity. This list is not exhaustive and in many applications, factors like security or time complexity should also be taken into account. Based on the embedding domain, data hiding can be carried out in either the spatial or in the transform domain. Another criterion can be the need (non-blind) or otherwise (blind) of the original 3D object. If only a few original parameters are needed for extracting the message, then the method can be termed as semi-blind. Totally blind methods may require the use of a secret key to retrieve the hidden message. In this case, the secret key is not only independent of the hidden content but also of the 3D object itself. A watermarking algorithm must be known to everyone, even if it uses a secret key, in order to respect the principle of Kerckhoffs [14].

In this paper, we propose to classify the 3D data hiding methods into two categories. The first of these categories is based on the scheduling of the object primitives or on the connectivity and topology of the 3D objects to be marked. In this first category, the positions of the vertices constituting the original 3D object are unchanged. The second category is based on the spatial displacement of the vertices of the 3D object or on the geometric deformations of other primitives constituting 3D objects, whether in the spatial or in the transform domain.

### 2.1. Methods based on primitives, connectivity or topology

This first category presents data hiding and steganography methods based on the scheduling of the object primitives or on the modification of connectivity or topological features of the 3D objects to be watermarked. In general, these changes are carried out in the spatial domain.

The embedding methods of this category are based on the organization of data with respect to the file comprising the 3D object. Ichikawa et al. [12] propose two watermarking methods which modify the organization of

data to embed a hidden message. The first method modifies the order of triangles in a list of triangles, while the second method modifies the order of triplets of vertices forming a triangle.

The *triangle strip peeling symbol sequence* (TSPS) watermarking algorithm of Ohbuchi et al. is based on a topological change [20]. To embed information in the triangle mesh, this algorithm proposes to detach a band of triangles of the original mesh. The first step of this algorithm is to choose, in the original mesh, a starting edge  $e$  of the initial triangle. From this edge of departure and the embedded message a strip of triangles is generated. The choice of the next triangle to be added to the band is based on the value of the bit to be embedded and the triangle scan order (clockwise or anticlockwise). This allows us to index all the edges to 1 or a 0 depending on their scan order in the triangle. In this manner the rest of the edges, and therefore the triangles generating the band, are chosen as a function of the message to be embedded. The next step of this algorithm is to detach this band of triangles by duplicating all the vertices of the band except those of the  $e$ . The embedded mesh is represented with a hole in the surface covered by the band of triangles containing the message. The message extraction is carried out by first searching the edge  $e$  and then scanning the associated band of triangles. The disadvantage of this method is that the characteristics of the object change with the introduction of the holes. Another disadvantage is that the file size increases with the addition of new vertices. Moreover, one can notice that this embedding is localized in the sense that the message embedded is not distributed across the mesh of the object, and it is easy to spot the duplicated vertices in the file and thus find the added message.

Mao et al. have developed a method to introduce smaller triangles into those initial triangles of the original 3D object in which the information is embedded [19]. With this approach, each edge of an initial triangle is divided into two edges by the vertex of a small triangle inserted. The ratio between the lengths of the two new edges is used to achieve the embedding of a bit. To initiate the process, two sides of a triangle are selected through a pseudo-random number generator. After embedding all the three edges of a triangle, the adjacent triangle of the latest used edge, if it exists and if it has not already been embedded, is chosen for the continuation of the embedding process. The maximal embedding capacity of this method is 2 bits per triangle (0.667 bit/vertex). This method can have a strong embedding capacity since the division step of a triangle can be repeated several times. For extraction, the two edges of the initial triangle are found by using the key and the triangles are scanned as has been done in the embedding step. The major disadvantage of this method is that new vertices are added leading to an increase in the file size. In addition, the watermarked areas are visible because an embedded triangle generates four smaller in the triangles. Lastly, the adjacency requirement of triangles<sup>2</sup> is no longer respected along the watermarked area.

<sup>2</sup> Each edge of a mesh must have exactly two sides adjacent to an edge inside the mesh and one if it is a border of the mesh.

One can notice that the position of the vertices constituting the original 3D object remains unchanged in these methods. However, some methods propose to add additional vertices, and most of these methods are very fragile and do not resist the manipulation, like the reversal of two lines, of the file containing the original data. Our method presented in Section 3 belongs to this category but is robust to data file manipulation, spreads over the message in the whole object and does not add vertices.

## 2.2. Methods based on geometric modifications of the 3D object

This second family of watermarking is based on the change in the geometry of the 3D object. This may be in the form of change in the position of the vertices, or alteration of the normals or faces of the triangles. Unlike the first family of watermarking, which applies only in the spatial domain, this second category may be applied in both the spatial and transform domains.

In the spatial domain, Ohbuchi et al. propose different watermarking approaches, such as *tetrahedral volume ratio* (TVR) and *triangle similarity quadruple* (TSQ) [20]. The TVR uses the ratio of the volumes of two tetrahedrons. In this method, a reference tetrahedron is needed to compare the volume of two tetrahedra. The tetrahedron used is constituted by an edge and its two incident triangles. The scanning path of the connectivity graph is based on a spanning tree of vertices. The edges of the constructed tree are used to generate and reference tetrahedra and their order of scan. The *vertex flood* algorithm of Benedens [5] is based on the change in the allowable distance of the vertices at the center of mass of the object. In another method, Benedens (*triangle flood*) uses the connectivity and geometry information to generate a path of the mesh of the object [6]. In this method the message embedding is done by modifying the position of vertices to alter the height of the triangles. The method proposed by Bors uses local geometric configuration of vertices for selecting embedding areas [8]. Each area is represented by a vertex and all its adjacent triangles. Bors calculates an ellipsoidal neighborhood for this vertex. To encode a 0, the vertex is moved out of its neighborhood and to encode a 1 it is internally displaced. Cayre and Macq propose a technique whereby the embedding is carried out by projecting the apex of a triangle on its base and watching whether it belongs to the left or right half of the base [9]. The selection of the embedding areas is made by a unique path in the mesh with a secret key. The method of Wagner [27] seeks to embed the hidden data along the length of local Laplacian vectors. To that end, Wagner alters the local curvature of the object. The use of NURBS (non-uniform rational B-spline) was also explored for the watermarking of 3D objects. Benedens has proposed a method of 3D object watermarking using NURBS and changing weights and vectors of nodes in a manner that preserves the overall geometry [7]. The method of Lee et al. proposes to embed the watermark no more in the 3D object, directly, but in

the virtual images derived from it [17]. In this article, for affecting a watermarking, three virtual images are constructed by uniformly sampling the 3D object and recording the coordinates of vertices thus obtained. The method of Agarwal and Prabhakaran is a blind method, robust and resistant to cropping [2]. This approach is based on a method of clustering. The algorithm divides the vertices into clusters by the application of the closest neighbor heuristic. Within each cluster the vertices are organized in the form of a graph. In a cluster, vertices with at least to incident edges are called cluster heads. Vertices other than the cluster heads, i.e. the leaflets, are utilized for embedding. The distance between the cluster head and the leaflets, to be watermarked, is partitioned into intervals. The embedding is carried out by modifying this interval by decreasing the euclidean distance between the cluster head and the leaflet.

In the transform domain, Ohbuchi et al. propose a method that employs a Laplacian matrix of the 3D mesh in order to pass from the spatial domain to the frequency domain [21]. In the wavelet domain, the method of Kanai et al. uses a wavelet decomposition mesh and embed the information bit by changing the LSBs of certain wavelet coefficients [13]. Yin et al. too propose a wavelet domain algorithm [30]. The algorithm is based on a multiresolution decomposition of the 3D object mesh developed by Guskov et al. [11]. The method proposed by Praun et al. uses a Kernel space to embed information in the low frequency surface components [22]. Wang et al. offer a hierarchical watermarking method. Thanks to the wavelet decomposition, a watermark robust to geometric attacks and a high capacity watermark are embedded into different resolution levels of the mesh by modifying the values of the wavelet coefficients [29]. Rondaio-Alface et al. use and improve the robustness of very good methods of watermarking based on histograms and perform the robustness against cropping [26]. Bennour and Dugelay have proposed two embedding methods, one blind and the other non-blind, with the aim of embedding a 3D object such that the mark is always present, in one form or another, in the resulting 2D images [4]. Recently, two surveys have been developed in [25,28].

## 3. The proposed method

In this section we describe a new method of embedding data in 3D objects. Our method is based on a 3D model represented by cloud of vertices and a list of edges corresponding to a triangular mesh of the surface. The main idea of the method is to find and synchronize specific areas that can be used to embed the message. The embedding of data is based on the modification of the vertex connectivity in the selected areas. These modifications have implications for changing the structure of triangles in these areas. This approach of data hiding has the advantage of not changing the initial position of the vertices of the 3D model. This invariant position of the vertices enables us to make the embedding robust to affine transformations of such types like rotation,

translation or change of scale. The method is based on the concept of minimum spanning tree (MST).

Our approach consists of three steps. The first step is to construct a MST covering the vertices of the initial 3D model. The second step necessitates a scanning of the constructed MST to select and synchronize areas for embedding the data. The starting vertex of this step is selected on the basis of a secret key. From a principal component analysis (PCA) of the 3D object, the secret key allows us, in the coordinate system of the object, to locate the starting vertex of the MST. We employ the PCA in an attempt to obtain the three principal directions of the 3D object. To determinate them, the moments are calculated and we obtain coefficients of variance and covariance. This set of coefficients is represented by a  $3 \times 3$  covariance matrix and an analysis of this matrix gives three eigenvalues and three associated eigenvectors. These vectors denote directions providing maximal variance of the 3D object. Associated with the centroid of the 3D object, these directions form a frame of reference adapted to select the first vertex. As a result of the PCA, a reorganization of the file does not disturb the location of the starting vertex which is dependent, therefore, only on the secret key. The embedding areas must meet a number of requirements due to the connectivity of MST and the geometry of the areas. The third step involves embedding of the message itself. This is done by modifying the connectivity of triangles in the selected areas. Fig. 1 illustrates the overview of the method.

Section 3.1 is about the definition of a MST and its construction. We show, in Section 3.2, how to select embedding areas in order to synchronize them with the message. The data embedding process is described in Section 3.3. Section 3.4 depicts that the selected embedding areas are subject to geometrical constraints. Finally, Section 3.5 describes how to extract the message embedded in a watermarked 3D file.

### 3.1. Minimum spanning tree

#### 3.1.1. Definition

Let  $G$  be a graph consisting of a set of vertices and the connecting edges. An edge of  $G$  can be represented by  $(a,b)$ , where  $a$  and  $b$  are the corresponding vertices. Let  $A$  be a undirected connected graph without any cycle which

implies that there is no path linked to any vertex of  $A$  that would revert back to the same vertex. This graph has thus two characteristics:

- If an edge removed, it is no longer connected.
- If an edge is added, it forms a cycle.

For graph  $G$ , there is at least one tree spanning  $A$ . It is a partial graph which has the same vertices as  $G$  and that the overall relationship is included in the relationships in  $G$ .

The cost of graph  $G$  is the sum of all the weights of the edges of  $G$ . If the sum of weights of the edges of a tree  $A$  spanning  $G$  is the lowest among all the possible spanning trees of  $G$  then  $A$  is called the *minimum spanning tree* (MST) of  $G$ . In our method, the weights used to calculate the MST are the Euclidean distances between the vertices of the 3D object. The peculiarity of MST is that it is unique, although there is an ambiguity in the methods of construction when a vertex has two edges of the same weight. The method chooses the edge it encounters first in its walk over the vertices. If the scheduling of vertices in the file is changed then modifications in the MST may also be likely. To correct this ambiguity a calculation of distance between the two vertices of the two identical edges and the center of mass of the 3D object is utilized. In this way MST is always unique regardless of the reorganization of vertices in the file of the 3D object. It is this peculiarity of MST that we refer to Section 3.2 to synchronize the message with the 3D model. We also rely this feature to embed the data as presented in Section 3.3.

#### 3.1.2. Construction of the MST

There are many algorithms for MST construction, the most popular being the *Prim's algorithm* [23] and *Kruskal's algorithm* [16].

- *Prim's algorithm* gradually constructs a subgraph  $A$  of  $G$  by choosing the edge of minimum weight locally at each step, hoping it to be a part of the global solution [23]. Initially,  $A$  is empty and a vertex  $s$  of  $G$  is chosen as a starting vertex.  $A$  is thus composed of only  $s$  at the start. Let  $E$  be the set of vertices connected to  $s$  in  $A$  and let  $F$  is the set of remaining vertices. Initially  $E$  is empty, and  $F$  contains all the vertices of  $G$ . At each step, an edge is added to  $A$  provided that this edge has minimum weight among all the edges having a vertex  $x$  in  $F$  and the other in  $E$ . This must not form a cycle. Thus vertex  $x$  comes out of the  $F$  and is included in  $E$ . The algorithm is stopped when  $F$  is rendered vacant. Fig. 2b presents a result of Prim's algorithm when applied to the example given in Fig. 2a.
- *Kruskal's algorithm* gradually constructs a subgraph  $A$  of  $G$  by first sorting all the edges depending on their weight and then selecting those which do not form cycle in  $A$  [16]. For this, the vertices are chosen one at a time and sorted in ascending order according to the weight of the set of those edges that link a vertex  $s$  to others. For each sorted edge  $(s, s')$  check whether the list of neighbors of  $s$  in  $A$  is different from those of  $s'$  in  $A$ . If no vertex neighbor is common, then the edge  $(s, s')$  is added to  $A$ .

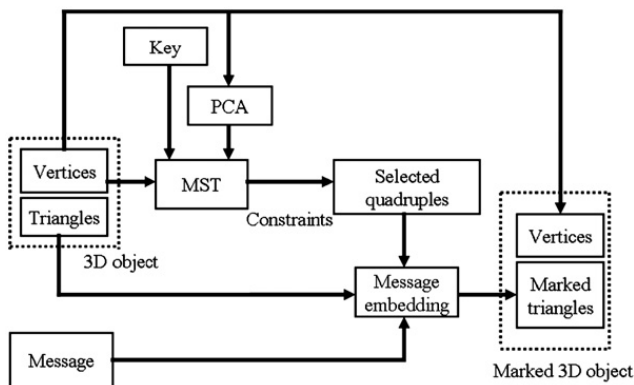


Fig. 1. Overview of the method of MST-based embedding.

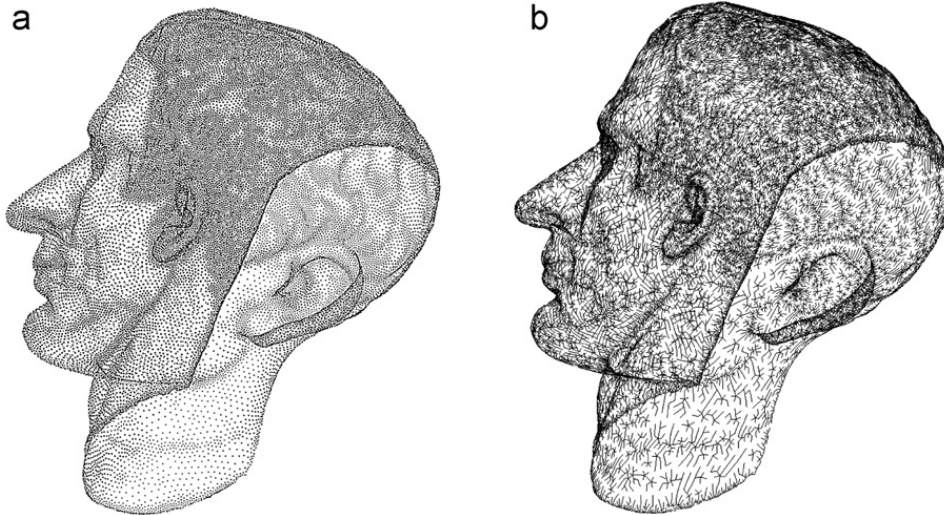


Fig. 2. (a) Cloud of initial vertices (43,039 vertices) and (b) calculation of the MST with Prim's algorithm.

### 3.2. Selection and synchronization of embedding areas

Synchronization in the embedding process is achieved by identifying the specific connectivities present in the MST. Once MST is built, two phases are required to synchronize the message to be embedded. The first phase is to define the manner in which MST is to be scanned. Indeed, the tree is unique, but the way of scanning depends on the vertex to start with. In our approach this vertex is selected by the use of a secret key that brings us security of the message to be embedded into the 3D model. One can say that the number of ways to read the MST is equal to the number of nodes in the 3D model. Thus the greater the number of nodes, higher is the system security.<sup>3</sup> Once the scanning pattern of MST is defined, specific areas are identified for the perspective embedding of data. Such areas correspond to edges of the MST belonging to particular areas of the 3D mesh as presented in Fig. 3. These areas correspond to a specific edge of the MST which have two triangles with specific configurations. There are two configurations allowing to select the areas used for the embedding step. The first configuration is represented by two triangles with a common edge corresponding to the edge of MST. The second configuration is represented by two triangles with a common edge and the two vertices which do not belong to this common edge correspond to two nodes of the current edge of MST. These quadruples, composed of two triangles and synchronized with two nodes of the MST, therefore, represent the areas where the message bits are to be embedded. Fig. 3 illustrates the selection and synchronization of three quadruples which can be used to embed data. Fig. 3a presents the construction of the MST starting from the node A. Fig. 3b illustrates the corresponding triangles we can find in the file of the 3D object. From the MST and these triangles, we can detect and synchronize three quadruples illustrated in Fig. 3c. For example, the first quadruple is based on the

edge number 2 and the two triangles using this edge such common edge.

The embedding method presented in Section 3.3 directly utilizes these detected and synchronized areas. However, as we will explain in Section 3.4, some part of the quadruples cannot be used for embedding. In fact, some areas do not meet certain constraints imposed by the data hiding method and others cannot be used for reasons of not being imperceptible. The MST, due to its construction, will allow us to resist the affine transformations such as rotation, translation or scale variation. This is made possible through our embedding technique which does not alter the position of the vertices in the 3D object. The number of selected areas depends not only on the number of vertices in the 3D object and threshold values for the various constraints but also on the overall mesh of the object. We show, in Section 4, how many areas are selected, at an average, based on all these parameters.

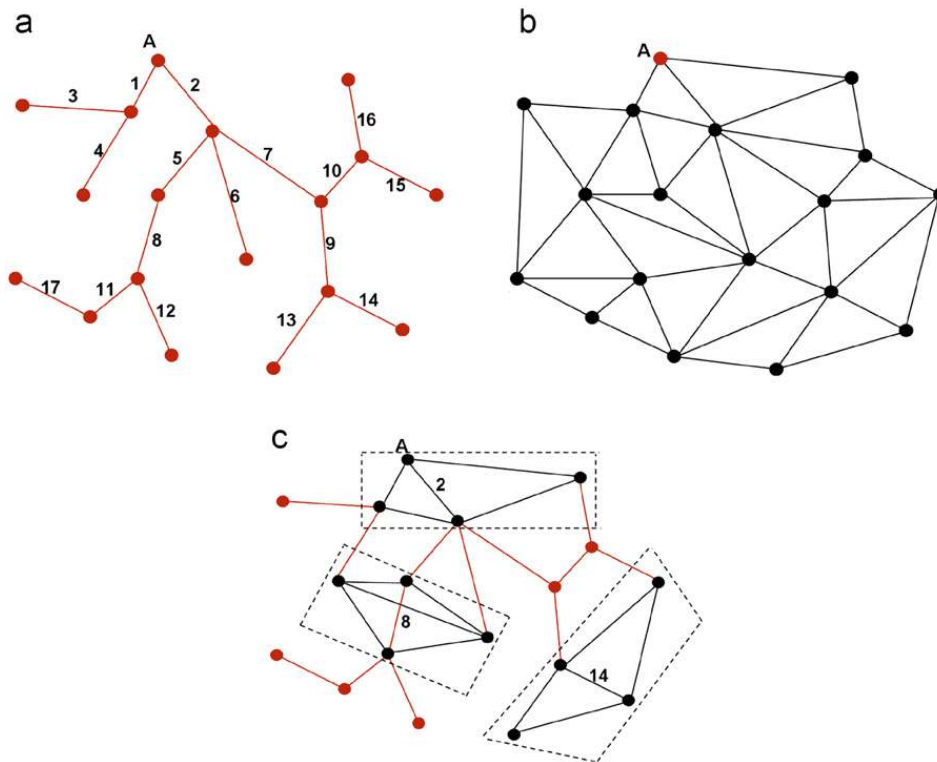
### 3.3. Data hiding step

This section details the embedding step. In Section 3.2 we presented how to locate embedding areas in the MST. These areas are based on edges of the MST and need triangles with common edge. These areas are therefore quadruples. To add one bit of the message in a selected quadruple we look at the edge of the two included triangles in the quadruple.

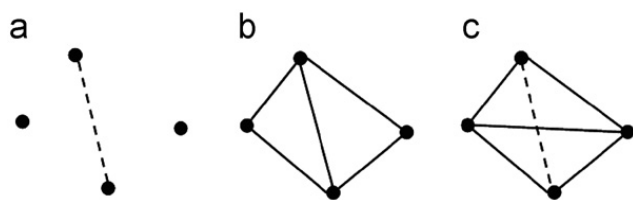
An example of areas used for embedding data is presented Fig. 4. Embedding of a 0-bit is realized by constructing the common edge of the two triangles on the edge of the original MST as presented in Fig. 4b. In contrast the embedding of a 1-bit, the common edge of the two triangles is constructed so as not to be superimposed over the initial edge of the MST as presented in Fig. 4c. In Fig. 5, from the selected quadruples illustrated in Fig. 3, we show the embedding of the message "100". Note that the change takes place in the mesh connectivity and not in that of the MST. This method of embedding permits not to change the position

<sup>3</sup> The key can also serve as a seed for a PRNG to randomly scatter the message in the 3D object.





**Fig. 3.** Example of areas forming quadruples selected to embed data: (a) construction of the MST starting from A, (b) the corresponding triangles in this area and (c) selection and synchronization of three quadruples.



**Fig. 4.** (a) A selected quadruple based on an edge of the MST, (b) embedding of a 0-bit and (c) embedding of a 1-bit.

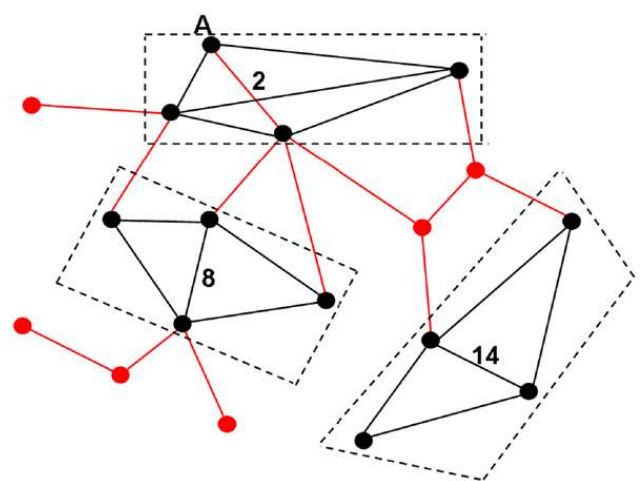
of the vertices of the 3D model. This is an important point for the synchronization and allows us to rebuild the same MST during the extraction of the message. However, this method of embedding imposes certain constraints on the quadruples to be chosen. Modification in the mesh connectivity can lead not only to errors in it but also to visual defects in the surface. We explain in Section 3.4 why these constraints are to be respected and why some quadruples cannot be used to embed the data.

### 3.4. Constraints on the selection of the embedding areas

In this section we present the constraints applied to quadruples resulting from the synchronization step. In fact, some of these quadruples cannot be used for the data embedding. These constraints are concerned with coplanarity, convexity and overlapping.

#### 3.4.1. Coplanarity constraint

The first constraint is the coplanarity of the quadruples. The underlying fact is that any change in



**Fig. 5.** Data hiding of the message “100”.

the connectivity of the two triangles forming one quadruple also changes the angle formed between these two triangles. Consequently, the data hiding on a non-coplanar quadruple 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 threshold of tolerance,  $S_c$ , on the coplanarity of the chosen quadruples. This threshold allows us to reach a compromise between the embedding capacity and the quality of the 3D model. The higher the threshold, higher

will be the embedding capacity at the expense of visual rendering. Conversely, lower threshold will lead to lower embedding capacity and hence smaller visual degradation of the 3D object. Through the threshold  $S_c$ , adjustment is thus possible between the embedding capacity and the quality of the visual rendering of the 3D model.

Let the quadruple  $Q_{1234}$  is formed by two triangles  $T_{123}$  and  $T_{234}$ . First of all, the normal  $N_1$  to plane  $P_{123}$  and the normal  $N_2$  to plane  $P_{234}$ , as well as angle  $A_1$  between the  $N_1$  and  $N_2$  is calculated. In the second step, the common edge of the two triangles in the quadruple is modified to simulate an inverse embedding. The new angle  $A_2$  between the normals of the triangles is calculated. Fig. 6 illustrates the calculation of the angles  $A_1$  and  $A_2$ . The quadruple will be retained only if both the angles are smaller than the already fixed threshold  $S_c$ :

$$Q_{1234} \text{ is retained if and only if } \max(A_1, A_2) < S_c. \quad (1)$$

### 3.4.2. Convexity constraint

The second constraint imposes that the quadruples used for the data hiding are convex. For this constraint, we assume that the selected quadruples already obey the coplanarity constraint. Fig. 4 illustrates a convex quadruple where a 0-bit and a 1-bit are embedded as shown in Fig. 4b and c, respectively. In this case, despite the change in triangle connectivity, the quadruple responds to the same characteristics and covers the same geometric space for the two embedding cases. Fig. 7a illustrates a case of non-convex quadruple formed by the planes  $P_{123}$  and  $P_{234}$ . We note that the modification of the triangle connectivity when embedding a bit modifies the surface covered by the previous quadruple. This modification introduces an error on the surface of the quadruple. This error is represented by the surface  $P_{134}$  in Fig. 7b. It is to prevent this error on the surface that constraint of convexity was introduced.

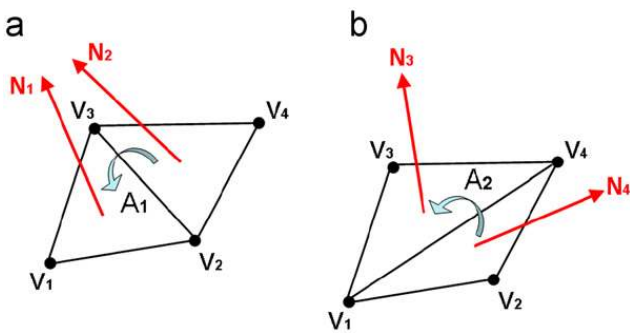


Fig. 6. Calculation of the angles  $A_1$  and  $A_2$  between two triangles forming a quadruple.

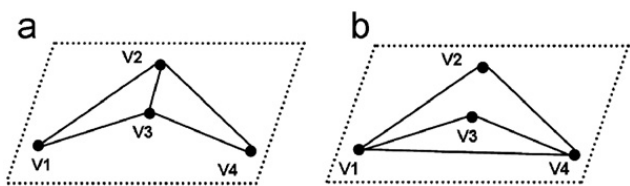


Fig. 7. (a) Triangles  $T_{123}$  and  $T_{234}$  before embedding and (b) triangles  $T_{124}$  and  $T_{134}$  after embedding.

The measurement of this constraint is a combination of the two following procedures:

- Vectors between vertices  $V_i$  et  $V_{(i \bmod 4) + 1}$  of the quadruples are calculated for  $i \in \{1, \dots, 4\}$ . Angles  $\alpha_i$  between two successive vectors are then calculated.

$$Q_{1234} \text{ is selected only if } \alpha_i < 180^\circ \text{ for } i \in \{1, \dots, 4\}. \quad (2)$$

This allows to eliminate 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 make 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. To this end we calculate the length of edges  $\overline{V_{23}}$  and  $\overline{V_{14}}$  and the distance  $D$  between the centers of these two edges. A quadruple is thus retained if and only if:

$$D \leq S_t \times \overline{V_{14}} \text{ and } D \leq S_t \times \overline{V_{23}}. \quad (3)$$

For this constraint, compared to the previous method presented in [3], we have a better formalization of this second measure. Thus, with this new constraint we can keep more quadruples than with the previous method.

### 3.4.3. Overlapping constraint

The third constraint concerns the overlapping of the quadruples. To avoid the recovery of quadruples when a quadruple is selected for embedding then the four nodes of the MST are marked and will no longer be used to search another quadruple. Finally we retain only the quadruples satisfying the presented constraints. Stability is also a strong constraint for data hiding. In fact, the connectivity change of triangles for the data hiding creates new triangles and therefore new quadruples. These new quadruples may well be susceptible to the constraints imposed and thus create new candidates for embedding which can interfere with the data hiding process. To avoid this problem, when all the neighbors of a node are covered even if no quadruple is selected for embedding, then the node is marked as no longer utilizable to search another quadruple. In the previous method presented in [3] we did not mark the nodes, consequently in order to extract the correct hidden message we eliminated too many quadruples. The embedding method presented in Section 3.3 is then applied only to the selected quadruples. The constraints restrict the maximum number of quadruples we can watermark to the theoretical limit of  $N/4$ , where  $N$  is the number of vertices of the 3D object. The maximum theoretical embedding capacity equals then to 0.25 bit/vertex.

### 3.5. Message extraction

To extract the hidden message, the first step is to rebuild the MST from the vertices of the model. We thus run a PCA on the vertices to determine the starting vertex

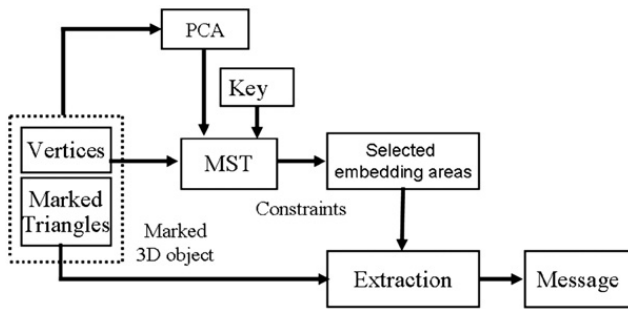


Fig. 8. Overview of the extraction method.

of the MST using the secret key. Since our embedding method does not change the position of the vertices, the rebuilt MST is the same as was for the embedding step. The second step is to locate the embedding areas, using the same constraints as had been for embedding. Once these two steps are performed, then extraction of the message from these areas is easy. For this we compare the position of the common edge of the two triangles forming a quadruple with the current edge of the MST. If the common edge forms a part of the MST then the extracted bit is a 0, otherwise it is a 1 as shown in Fig. 4. Note that the reconstruction of MST requires the use of a secret key to protect hidden information. Fig. 8 illustrates the process of extraction of the message. The MST of the object is unique, but the quadruples selected for embedding depend among them on the manner of scanning the MST. If  $N$  is the number of vertices in the 3D object, the complexity of this method is  $N$ . To increase the complexity of the method, the key can be used to scramble the order of bits in the message. The complexity will then be  $N \times 2^n$ , with  $n \leq N$  being the number of quadruples used for the embedding of the hidden data.

#### 4. Experimental results

In this section we apply and analyze our method to 12 3D objects having a number of vertices and different characteristics. Table 1 shows the characteristics of the 3D objects used.

Fig. 9 represents the mesh triangulation of the 3D object called *Julius* composed of 43039 vertices and 85145 triangles. In this section, to illustrate the results of quadruple search, we rely on this model. Fig. 10a–d illustrate respectively the results of the quadruples search with our method for a threshold  $S_c=1^\circ$ ,  $S_c=5^\circ$ ,  $S_c=10^\circ$  and  $S_c=30^\circ$ . With  $S_c=1^\circ$ , 777 quadruples are selected for the data embedding, while 2921, 4261 and 5517 quadruples are selected with respectively  $S_c=5^\circ$ ,  $S_c=10^\circ$  and  $S_c=30^\circ$ .

In order to illustrate the results we obtained, we compare the original 3D object illustrated Fig. 9 with the watermarked 3D objects illustrated Fig. 11a–d. We notice for these images of the watermarked 3D objects that when the threshold of coplanarity  $S_c$  is high the quadruples that had been reversed to achieve the embedding can become visible. The maximal embedding

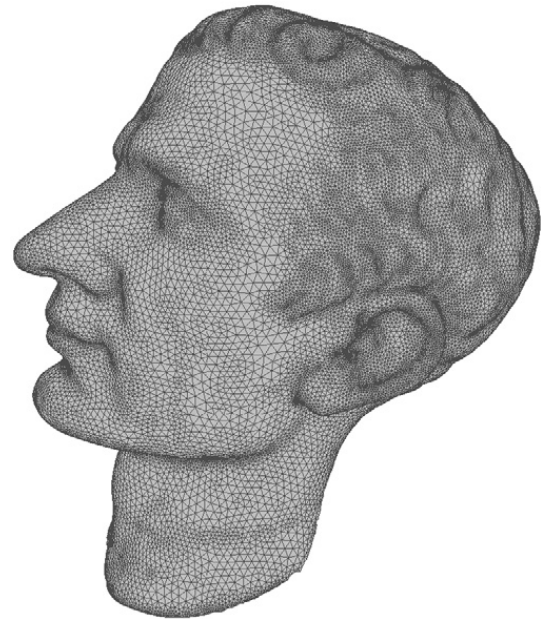


Fig. 9. Triangular mesh (43,039 vertices and 85,145 triangles).

capacity is observed for the object *Julius*, with  $S_c=30^\circ$  the embedding capacity equals to 0.128 bit/vertex. On the other hand when one lowers the threshold of coplanarity these quadruples are less visible. For the given 3D object with the coplanarity threshold of  $S_c=1^\circ$  one can note that it is not visually possible to see the difference with the original 3D object.









To illustrate the result of embedding conducted on a mesh, we present in Fig. 12 magnified a portion of the mesh. Fig. 12a shows a mesh before embedding, whereas Fig. 12b shows the same mesh wherein the set of the common edges of the selected quadruples for embedding have been reversed. If one considers that the selected quadruples used to embed bits of the message are convex and coplanar, then the change in connectivity made at the watermarking of these quadruples will not change the surface of the 3D model. The criteria to be taken into account in the analysis of results are the embedding capacity, analysis of the error on the surface generated by the embedding of hidden message and the resistance of our method to noise due to transformation.

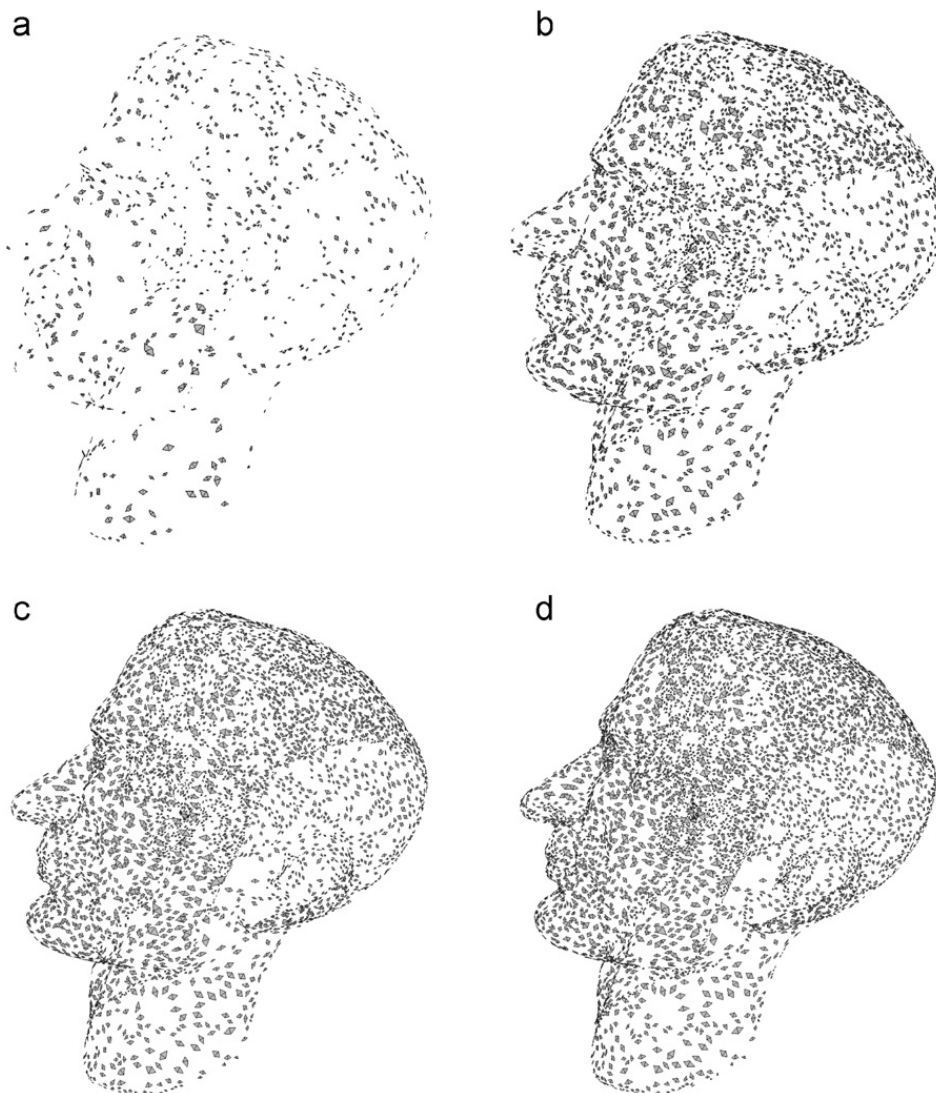
The first criterion regarding the embedding capacity, calculated in bit per vertex, allows us to evaluate the quality of our method. As illustrated in Fig. 13, the embedding capacities change for the 3D models presented in Table 1 for various values of the coplanarity threshold,  $S_c$ . The  $x$ -axis represents the type of the 3D object, from the smallest to the largest, and the  $y$ -axis represents the embedding capacity in bit/vertex. For various resolutions of the same object we note that the embedding capacity increases with the object size. If the coplanarity threshold  $S_c$  is increased, the difference in the embedding capacities in the objects no longer hold. Fig. 14 also illustrates this criterion wherein the  $x$ -axis represents the threshold of coplanarity  $S_c$  used for the quadruples search and the  $y$ -axis represents the mean embedding capacity calculated for the set of 3D objects given in Table 1.

We can point out that the embedding capacity increases with the coplanarity threshold  $S_c$ . In Fig. 14 we also compare the embedding capacities of our proposed

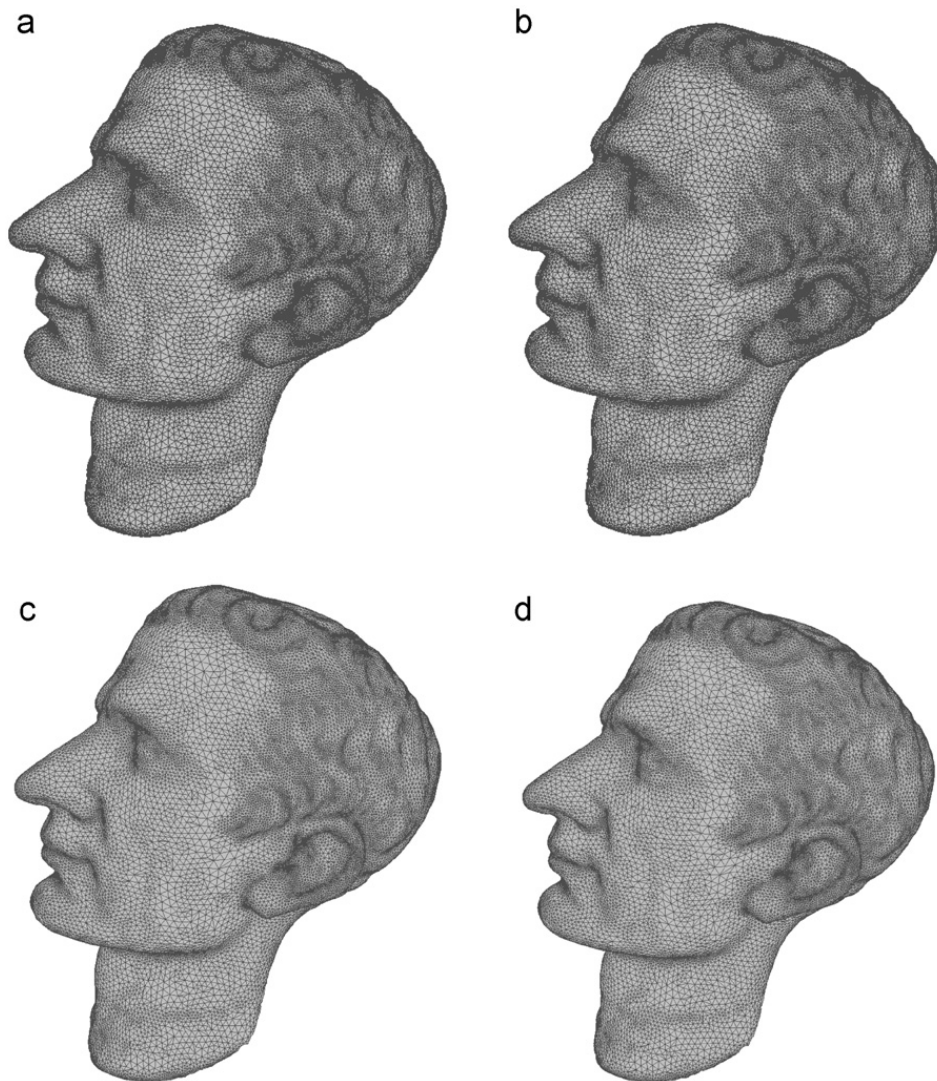
method with a previous one [3]. With this new method, we can observe that we doubled the embedding capacity irrespective of the coplanarity threshold.

**Table 1**  
Characteristics of the 3D objects used.

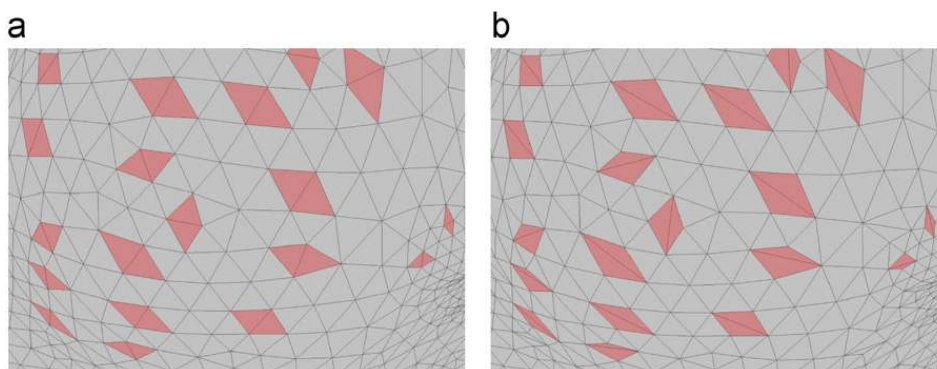
3D Object	bunny	dragon	happy	male shoe
Number of vertices	453/1887/8146	5203/22,982	7102/32,316	12,606
Number of triangles	948/3851/16,301	11,102/47,794	15,536/67,240	25,208
				
3D Object	Chinese dragon	lady shoe	Julius	horse
Number of vertices	20,184	31,088	43,039	48,485
Number of triangles	40,364	62,168	85,145	96,966
				



**Fig. 10.** Quadruple selection for the (43,039 vertices and 85,145 triangles): (a) with a threshold  $S_c=1^\circ$ , 777 quadruples are selected, (b) with a threshold  $S_c=5^\circ$ , 2921 quadruples are selected, (c) with a threshold  $S_c=10^\circ$ , 4261 quadruples are selected and (d) with a threshold  $S_c=30^\circ$ , 5517 quadruples are selected.



**Fig. 11.** Watermarked 3D Object, *Julius* (43,039 vertices and 85,145 triangles): (a) coplanarity threshold  $S_c=1^\circ$  (777 bits embedded), (b) coplanarity threshold  $S_c=5^\circ$  (2921 bits embedded), (c) coplanarity threshold  $S_c=10^\circ$  (4261 bits embedded) and (d) coplanarity threshold  $S_c=30^\circ$  (5517 bits embedded).



**Fig. 12.** (a) Zoom in a area of the mesh before embedding, (b) zoom in the same area of the mesh after a reversal of all the common edges of the triangles of the selected quadruples.

The second analysis criterion consists of assessing the error made on the surface depending on the threshold of coplanarity  $S_c$  and therefore the number of embedded bits. As stated previously, increasing the threshold can increase the embedding capacity but this is done at the

expense of the visual quality of the surface. To evaluate the error in the watermarked surface relative to the surface of the original object, a calculation is performed to quantify the errors made at the embedding of a bit in a quadruple. The standard method for the calculation of the

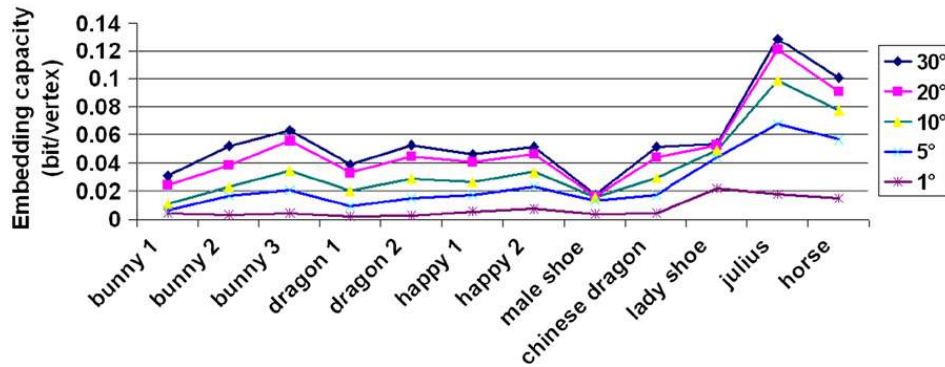


Fig. 13. Embedding capacity of the proposed method as a function of the 3D objects and the threshold  $S_c$ .

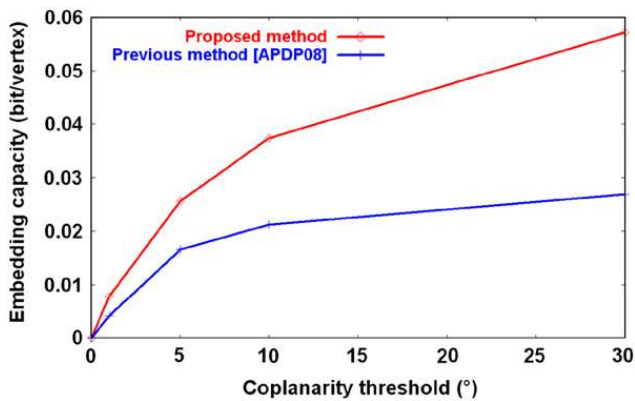


Fig. 14. Comparison of the average of the embedding capacity as a function of the threshold  $S_c$  between a previous method [3] and the proposed method.

error is the Hausdorff's geometric distance which uses the calculation of the Euclidean distance between two vertices on a mesh. There are many methods for calculating the error between two surfaces. Alliez et al. utilize volumetric information to achieve the calculation of the error between two 3D objects [1]. Kim et al. calculate the distance based on the curves [15]. Rondao, De Craene and Macq use a metric based on the estimate of the energy distortion [24]. Lavoué et al. measure the distortion between two 3D objects using the concept of structural similarity [18]. To measure the error with Hausdorff's geometric distance, we used in this work the algorithm proposed by the software Metro [10].

Fig. 15 shows the surface error values as a function of the threshold of coplanarity  $S_c$ . This error calculation has been applied to 3D models. Depending on the threshold  $S_c$ , Fig. 15 illustrates errors obtained with the method of Hausdorff. In all the cases we note that the error increases with the coplanarity threshold. We can also remark that the error are not the same as a function of the watermarked 3D object. Indeed, the error is larger for three particular objects which are *lady shoe*, *male shoe* and *Chinese dragon*. For the two shoes, these distortions are due to the original surface of these objects composed of triangles very disorganized compared to the other 3D objects presented in Table 1. The object *Chinese dragon* is the more complex object we have tested because a lot of

high curvatures are near flat areas which have been watermarked.

The third criterion allows us to evaluate the robustness of the proposed method to noise. Fig. 16 represents the robustness of our method against the noise added to the vertices in the 3D object. The  $x$ -axis represents the magnitude of added noise and  $y$ -axis the percentage of correct quadruples detected. One can observe that with a low amplitude noise, 100% of the quadruples are selected. However, with an increase in the amplitude of the noise, the number of correct quadruples detected declines fairly rapidly throughout except for one object which seem to be more robust to noise. This object called *bunny 1* is the smallest object we have tested and its embedding capacity is very small. All the other objects with a larger embedding capacity have been faster disturbed by an adding of noise. We can notice that object the more complex, *Chinese dragon*, is the less robust to noise of all the objects we have tested.

In order to analyze the reliability of the proposed method, even if we have a large complexity for the extraction due to the use of a secret key, we have done a receiver operating characteristic (ROC) analysis. The ROC curves presented in Fig. 17 illustrate the sensibility and the specificity of our method by calculating the false positives and false negatives during the extraction step. We have watermarked the object *Julius* with  $S_c=15^\circ$ , in this case 4816 bits are hidden. For several values of the noise, we changed the threshold value  $S_c$  between  $1^\circ$  and  $30^\circ$  to extract the message. We note that when you change the value of  $S_c$ , then false positives and false negatives appear. This is due to the fact that when we change the value of  $S_c$ , the constraints, described Section 3.4, particularly the overlapping constraint, do not select the same quadruples. This analysis shows that the proposed method is very sensitive to noise.

We have compared our method with previous methods presented in Section 2. We have embedded the same message of 5517 bits in the object *Julius* with the method of Mao et al. [19]. With the method of Mao et al., the watermarked object is no more robust than with our method and the distortion measured with the Hausdorff's geometric distance is larger than with our method. Moreover, the size of the watermarked 3D object has increased. We have also compared our method with the

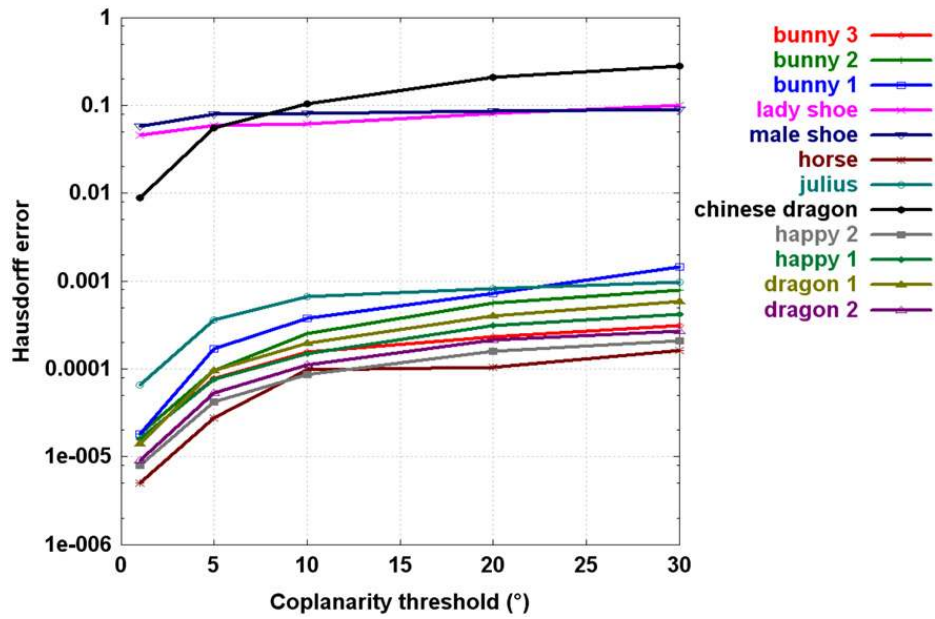


Fig. 15. Error as a function of  $S_c$  calculated by Hausdorff's Method.

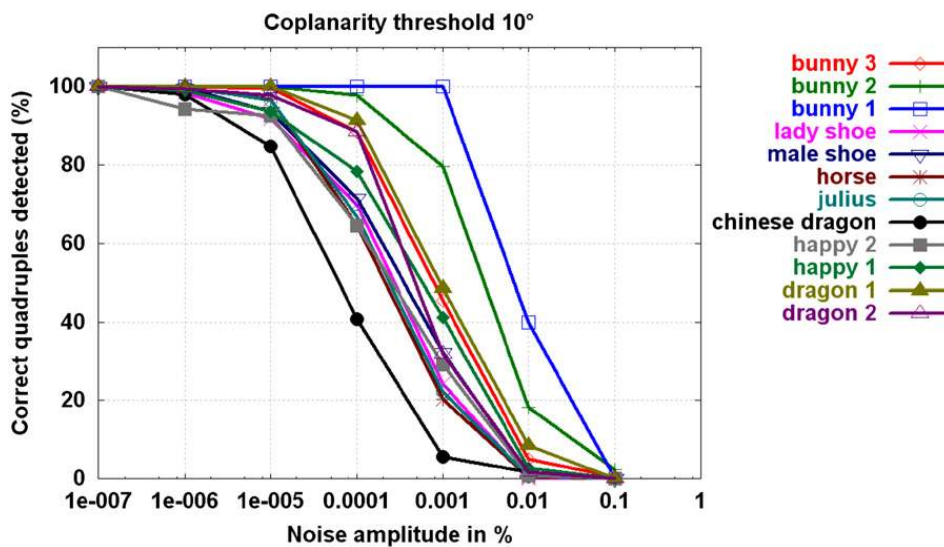


Fig. 16. Effect of noise on the detection of the quadruples.

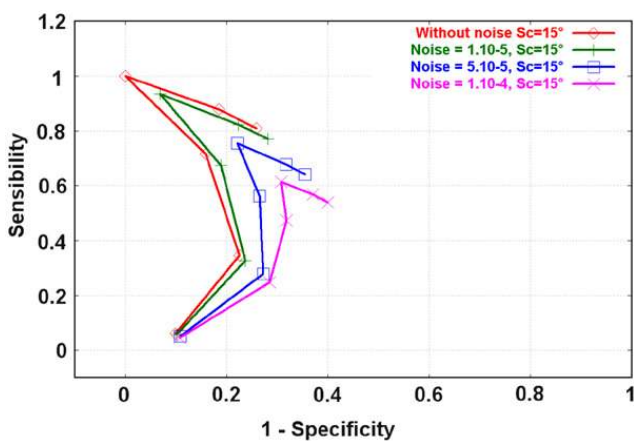


Fig. 17.

method presented by Cayre and Macq [9]. For the same message, the method of Cayre and Macq is more robust to noise, but the distortion is a hundred times larger than with our method.

### 5. Conclusion and perspective

In this paper we presented a new 3D steganography method. The peculiarity of the method lies in the fact that there is no change in the position of the vertices in the 3D object. In this sense the approach is lossless. For specific applications, like industrial, medical, military or cultural heritage applications, the fact to not move the vertices is very important. The proposed method enables the embedding, in a 3D object, of meta-data like the author's name, the date of creation of the object, a 2D or 3D logo,

color of the object or the texture of the object itself. For example, if a texture is embedded in a 3D object, when someone receives the watermarked 3D object, if he does not have the algorithm and the corresponding key he can visualize the 3D object with a standard software but only if he has algorithm and the key he will be able to see the 3D object with texture. This method, based on MST scanning, synchronizes the embedding areas with the 3D model. Contrary to the previous methods belonging to this category, our method spreads the message all over the 3D object and does not add any new vertices. Moreover, our approach is robust to data file manipulation. The originality of the proposed method is to synchronize the hidden message with quadruples selected from the MST. This method is robust to attacks like rotation, translation and scaling but very sensitive to noise. The method is blind and the embedding capacity as a function of the threshold of coplanarity  $S_c$  allows us to hide large message. Indeed, the embedding capacity, on the average, is close to 0.06 bit/vertex and can reach 0.128 bit/vertex for particular objects. In perspective we are planning to develop a new method robust to noise based on the presented one. A new idea may be to change positions of vertices during the MST scanning to select quadruples. But in that eventually, the method will not be lossless and the application domains cannot be the same.

## References

- [1] P. Alliez, N. Laurent, H. Sanson, F. Schmitt, Mesh approximation using a volume-based metric, in: Proceedings of the Seventh Pacific Conference on Computer Graphics and Applications (PG '99), Washington, DC, USA, 1999, pp. 292–301.
- [2] P. Agarwal, B. Prabhakaran, Robust blind watermarking mechanism for point sampled geometry, in: Proceedings of the Ninth Workshop on Multimedia & Security (MM&Sec '07), Dallas, Texas, USA, 2007, pp. 175–186.
- [3] P. Amat, W. Puech, S. Druon, J.-P. Pedebay, Lossless data hiding method based on MST and topology changes of 3D triangular mesh, in: 16th European Signal Processing Conference on (EUSIPCO 2008), Lausanne, Switzerland, August 2008.
- [4] J. Bennour, J.L. Dugelay, Watermarking of 3D objects based on 2D apparent contours, in: Proceedings of 18th Annual Symposium on Electronic Imaging, vol. 6072, San Jose, USA, January 2006.
- [5] O. Benedens, Geometry-based watermarking of 3D models, IEEE Computer Graphics and Applications 19 (1) (1999) 46–55.
- [6] O. Benedens, Two high capacity methods for embedding public watermarks into 3d polygonal models, in: Proceedings of Multimedia and Security-Workshop at ACM Multimedia, 1999, pp. 95–99.
- [7] O. Benedens, Affine invariant watermarks for 3D polygonal and NURBS based models, in: Proceedings of Third International Workshop on Information Security (ISW '00), London, UK, 2000, pp. 15–29.
- [8] A.G. Bors, Watermarking mesh-based representations of 3-D objects using local moments, IEEE Transaction on Image Processing 15 (3) (2006) 687–701.
- [9] F. Cayre, B. Macq, Data hiding on 3D triangle meshes, IEEE Transactions on Signal Processing 51 (4) (2003) 939–949.
- [10] P. Cignoni, C. Rocchini, R. Scopigno, METRO: measuring error on simplified surfaces, Computer Graphics Forum 17 (2) (1998) 167–174.
- [11] I. Guskov, W. Sweldens, P. Schröder, Multiresolution signal processing for meshes, in: Proceedings of 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99), New York, NY, USA, 1999, pp. 325–334.
- [12] S. Ichikawa, H. Chiyama, K. Akabane, Redundancy in 3D polygon models and its application to digital signature, in: Proceedings of 10th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG '02), 2002, pp. 225–232.
- [13] S. Kanai, H. Date, T. Kishinami, Digital watermarking for 3D polygons using multiresolution wavelet decomposition, in: Proceedings of Sixth International Workshop Geometric Modeling: Fundamentals and Applications, December 1998, pp. 296–307.
- [14] A. Kerckhoffs, La cryptographie militaire, Journal des sciences militaires 9 (1883) 5–38.
- [15] S.J. Kim, S.K. Kim, C.H. Kim, Discrete differential error metric for surface simplification, in: Proceedings of 10th Pacific Conference on Computer Graphics and Applications (PG '02), Washington, DC, USA, 2002, p. 276.
- [16] J.B. Kruskal, On the shortest spanning tree of a graph and the traveling salesman problem, Proceedings of American Mathematical Society 7 (1956) 48–50.
- [17] J.J. Lee, N.I. Cho, J.W. Kim, Watermarking for 3D NURBS Graphic Data, in: Proceedings of IEEE Workshop on Multimedia Signal Processing, 2002, pp. 304–307.
- [18] G. Lavoué, E. Drelie Gelasca, F. Dupont, A. Baskurt, T. Ebrahimi, Perceptually driven 3D distance metrics with application to watermarking, in: Proceedings of SPIE Applications of Digital Image Processing, XXIX, San Diego, August 2006.
- [19] X. Mao, M. Shiba, A. Imamiya, Watermarking 3D Geometric Models Through Triangle Subdivision, in: P. W. Wong, E. J. Delp (Eds.), Proceedings of SPIE Security and Watermarking of Multimedia Contents III, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, vol. 4314, August 2001, pp. 253–260.
- [20] R. Ohbuchi, H. Masuda, M. Aono, Watermarking three-dimensional polygonal models, in: Proceedings of Fifth ACM International Conference on Multimedia (MULTIMEDIA '97), New York, NY, USA, 1997, pp. 261–272.
- [21] R. Ohbuchi, A. Mukaiyama, S. Takahashi, A frequency-domain approach to watermarking 3D shapes, Computer Graphics Forum 21 (3) (2002) 373–382.
- [22] E. Praun, H. Hoppe, A. Finkelstein, Robust mesh watermarking, in: A. Rockwood (Ed.), Proceedings of 26th Annual Conference on Computer Graphics and Interactive Techniques (Siggraph '99), Los Angeles, 1999, pp. 49–56.
- [23] R.C. Prim, Shortest connection networks and some generalisations, Bell System Technical Journal 36 (1957) 1389–1401.
- [24] P. Rondao-Alface, B. Macq, Shape quality measurement for 3D watermarking schemes, in: security, steganography, and watermarking of multimedia contents VIII, vol. 6072, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, 2006, pp. 622–634.
- [25] P. Rondao-Alface, B. Macq, From 3D mesh data hiding to 3D shape blind and robust watermarking: a survey, LNCS Transactions on Data Hiding and Multimedia Security 2 (2007) 99–115.
- [26] P. Rondao-Alface, B. Macq, F. Cayre, Blind and robust watermarking of 3D models: how to withstand the cropping attack? in: Proceedings of IEEE International Conference on Image Processing (ICIP07), San Antonio, Texas, USA, 2007, pp. 465–468.
- [27] M.G. Wagner, Robust watermarking of polygonal meshes, in: Proceedings of Geometric Modeling and Processing (GMP '00), Washington, DC, USA, 2000, pp. 201–208.
- [28] K. Wang, G. Lavoué, F. Denis, A. Baskurt, Three-dimensional meshes watermarking: review and attack-centric investigation, in: Ninth International Workshop on Information Hiding, Saint Malo, France, 2007, pp. 50–64.
- [29] K. Wang, G. Lavoué, F. Denis, A. Baskurt, Hierarchical blind watermarking of 3D triangular meshes, in: Proceedings of IEEE International Conference on Multimedia & Expo, Beijing, China, July 2007, pp. 1235–1238.
- [30] K. Yin, Z. Pan, J. Shi, D. Zhang, Robust mesh watermarking based on multiresolution processing, Computers and Graphics 25 (June 2001) 409–420.