Watermarking of 3-D objects based on 2-D apparent contours

Bennour J.[a] and Dugelay J-L.[a]

[a]Institut Eurecom, 2229 route des Cretes, B.P.193 06904 Sophia-Antipolis, France

ABSTRACT

In this paper we describe a novel framework for watermarking 3-D objects via contour information. Instead of classical existing watermarking technologies dealing with 3-D objects that operate on the 3-D object itself to insert and extract the mark (3-D/3-D approach), the goal of our work is to retrieve information originally hidden in the apparent contour of the object, from 2D views of the object (3-D/2-D approach).

Keywords: : Watermarking, 3-D object, apparent contour.

1. INTRODUCTION

Ever more and more synthetic objects can be used in videos or images. Watermarking can then be useful for several purposes. In particular, we would like to check if the use of a given object is legal or not, to access additional information concerning that object (e.g for authentification or indexing), the owner (copyright), or even the purchaser (e.g for non-repudiation).

The already-existing watermarking algorithms allow the protection of 3-D objects in such a way that we need to access to 3-D suspicious watermarked data. However, based on our experience, it is usually more frequent to locate and recover suspicious 2-D images (i.e projection of the 3-D object) on Web pages for example than the 3-D files themselves. To the best of our knowledge, existing work dealing with watermarking 3-D objects does not enable users to extract the mark from represented views of 3-D protected objects.

The approach we present in this paper hides the mark in the apparent contour of the 3-D object. The mark can then be recovered from the represented views of the object in which watermarked contour information is present (see figure 1).

The paper is organized as follows: section 2 presents a short state of the art in watermarking of 3-D objects. Section 3 describes our new approach to watermark 3-D objects and discusses some obtained results. Finally, section 4 concludes and indicates future work.

2. STATE OF THE ART

There is a wide variety of watermarking techniques in the domains of audio, video and image data. Even if the number of watermarking techniques for 3-D objects (regarded as new objects in watermarking) has been continuously increasing since 1998, it remains marginal. In the multimedia field, 3-D objects are generally represented by meshes associated with a texture. Presentations with Nurbs, CSG (Constructive Solid Graph)

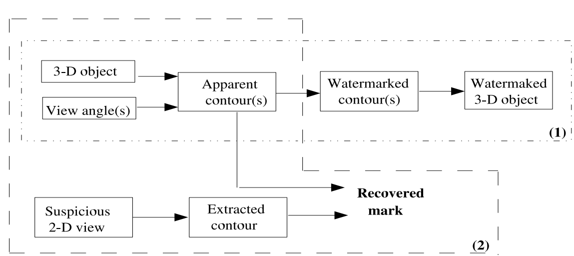

Figure 1. Global system: 1- Watermark insertion 2- Watermark detection.

trees and groups of dots are less widespread. Moreover, most of the previous works dealing with 3-D objects are based on slight modifications performed on meshes via geometric and/or topologic data of 3-D objects. [1] embeds information in a mesh described by a list of vertices and faces as well as by permutting the order of vertices of faces or by rotating them.[2] embeds information in the geometrical structure of the graphical object by changing the location of certain vertices. Other algorithms are based on normal distributions[3]-,[4] on normal sizes[5] or on geometrical embedding primitives that are invariant to certain class of geometrical transformations such as the ratio of the lengths between two segments of a straight line.[6] We can also cite watermarking algorithms[7]-[8]-[9] which hide the mark in 3-D objects presented by NURBS and[10] that dissimulates information in 3-D model represented by a CSG tree.
All these watermarking algorithms dealing with 3-D objects do not enable us to extract the mark from 2-D represented views of 3-D protected object. In this quite particular context, we can cite[11]-[12] which protect 3-D object usage through texture watermarking. We can then check if the represented object is protected by extracting the watermarked texture from the represented views of the object. This algorithm supposes that 3-D objects are realistic or at least rich in texture, whereas objects used in our approach could be with or without texture.

3. OUR APPROACH TO WATERMARK 3-D OBJECTS

This section is devoted to the description of our approach to watermark 3-D objects. We first explain the whole process and then we describe in detail the most important steps.

3.1. The whole process

Figure 2 shows a flowchart of our 3-D object watermarking scheme. Step 1-2-3 concern the watermark insertion while step 4-5-6 concern the watermark extraction.
Given a known 3-D object consisting of a geometric definition (represented by 3-D polygonal meshes) we can protect it by watermarking its apparent contour. Then we seek for extracting the mark from the represented views of the object.

In order to avoid the basic problem of 2-D/3-D alignment[13]-[14] we are faced with to align the suspicious 2-D views and the original 3-D object to extract the mark, we have used several 2-D views for which the set of projection parameters is assumed to be known. The viewing angles considered to get the represented views could be kept secret.

3.1.1. Watermark insertion

- **Step 1**: We fix k point of views (see subsection 3.2) and we extract each apparent contour of the 3-D object corresponding to these viewing angles (see subsection 3.3).
- **Step 2**: We watermark the extracted apparent contours (see subsection 3.4.1).
- **Step 3**: We transpose modifications on the 3-D object to get it watermarked.

3.1.2. Watermark detection

- **Step 4**: Considering the viewing angles we project the original object with respect to these angles and we extract the corresponding apparent contour (see subsection 3.3).
- **Step 5.1**: We align the two views : the first one representes the original 3-D object (among k views) and the second represents the suspicious 2-D view.
- **Step 5.2**: To extract the mark, we choose the most suitable view among the k represented ones. We call suitable view the view which is most similar to the suspicious one.
- **Step 6**: We detect the watermark's presence (see subsection 3.4.2).

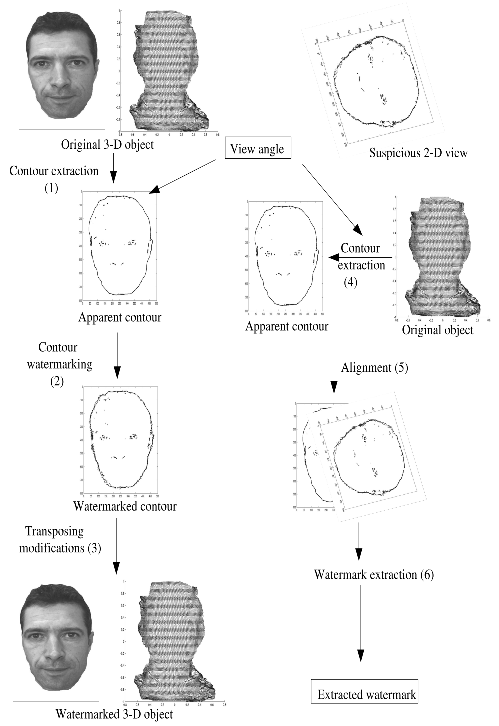

Figure 2. Principle.

Figure 3. Top view (a), intermediate view (b) and profile view (c).

3.2. Selection of the set of 2-D views

It goes without saying that the choice of caracteristic views has a great influence on the performance and limits of our watermarking approach.
For our first experiments we are interested in watermarking 3-D faces. We choose $k = 3$: top view, side view and intermediate one (figure 3). We estimate that these views are the most significant to identify a face.
The choice of represented views number k must satisfy two conflicting constraints:

1. The represented views must cover all the 3-D object.
2. We must define a limited number of represented views to insert the watermark in order to reduce time computing and to avoid conflicting overlaps between parts of the watermark signal.

For the future we will use an existing method in the literature to provide an optimal selection of 2-D views from a 3-D model. Early tests with the method developped in[15] are under construction. The selection of caracteristic views in this case is based on adaptive clustering algorithm and using statistical model distribution scores to select the optimal number and position of views.

3.3. Contour extraction

The usual method for computing 3-D object silhouette is to iterate over every mesh edge and check:

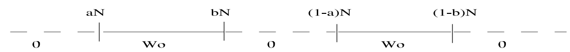

Figure 4. Watermark construction.

1. If the current edge is associated to only one face, in this case it is a contour.
2. If the current edge is associated to two faces F1 and F2, in this case:
   - We note $\overrightarrow{n1}$ and $\overrightarrow{n2}$ normal vectors of F1 and F2.
   - We note $\overrightarrow{v}$ a vector composed by camera position and one vertice of the current edge.

   If $(\overrightarrow{n1}.\overrightarrow{v}) \cdot (\overrightarrow{n2}.\overrightarrow{v}) < 0$ i.e $\overrightarrow{n1}$ and $\overrightarrow{n2}$ have opposite direction on camera axis
   Therefore F1 and F2 are oriented one face to the camera and the other one back to the camera therefore current edge is a contour .
   Else it is not an apparent contour edge.

3.4. Watermarking an apparent contour

In the previous subsection we have described how to extract apparent contour from 3-D objects represented via polygonal meshes (figure 2 - steps 1 and 4). In this subsection we describe the method used to watermark this apparent contour and we discuss the first obtained results concerning this module (figure 2 - steps 2 and 6).

Watermarking techniques are usually used for audio, video and image data while 3-D models and polygonal lines[16]-[17]-[18]) are relatively new covers. As far as we are concerned, we have chosen the algorithm described in the paper.[16] In addition to the fact that this algorithm is suitable for our approach, it is easy to implement. We summarize the main steps.

- We note $[x(n) \quad y(n)]$ the cartesian coordinates of each contour vertex.
- N the number of vertices.
- $z(n) = x(n) + iy(n)$, the complex signal.
- $Z(k) = \sum_{n=1}^{N} (z(n) \cdot \exp((-2\pi jkn/N))$ the Fourier transform representation of the signal z(n).
- W_0 a bivaluated +/-1 random sequence with zero mean and unit variance.

3.4.1. watermark embedding

The watermark is constructed as follows (see figure 4):

$$W(i) = \begin{cases} 0 & i < aN \;\; or \;\; bN < i < (1-b)N \\ & or \;\; (1-a)N < i. \\ W_0(i) & (1-b)N < i < (1-a)N \\ & or \;\; aN < i < bN. \end{cases}$$

The watermarked polygonal line is:

$$|Z'(k)| = |Z(k)| + p|Z(k)| \cdot W(k).$$

- N is the number of Fourier coefficients.
- a and b control the low and high frequency ranges that the watermark affects. $0 < a < b < 0.5$.
- p determines the watermark strength. p must be less than 1 to have $|Z'(k)|$ always positive.

The inverse Fourier transform of $Z'(k)$ produces the watermarked polygonal line L'.

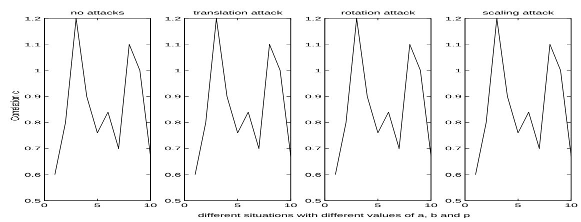

Figure 5. Watermarking detection: robustness to geometric attacks.

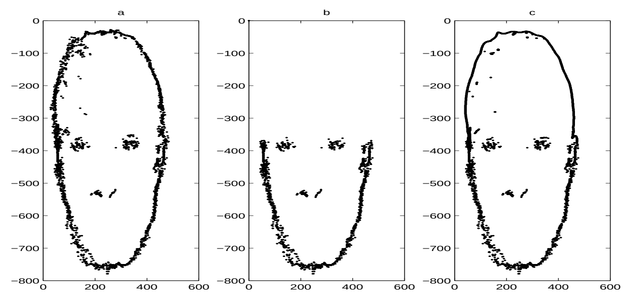

Figure 6. a- Contour with no attack, b- Contour with some deleted fragments, c- Contour with some replaced fragments.

3.4.2. Watermark detection

Let $|Z'(k)|$ be the Fourier descriptor of the watermarked line. The correlation c between W and $|Z'(k)|$ informs us about the watermark's presence.

$$c = \sum (W(k) \cdot |Z'(k)|)$$

Instead of c a normalized correlator $c' = c/mean(c)$ is used. The detection rule could be

- H_0: W watermarks L' if $c' > T$.
- H_1: W does not watermark L' if $c' < T$.

Some results of watermarking detection are shown on figure 5. The polygonal line used in the results shown here is the top view apparent contour (see figure 6 -a) with $N = 2389$ vertices. We have done many simulations with different values of a, b and p ($0 < a < b < 0.5$ and $p < 1$). We notice that in all cases the normalized correlator c' is more than 0.6 (the mean value of the normalized correlator c' is equal to 1 if W watermaks L). We also notice (see figure 5) that this algorithm is robust to geometrical transformations: translation (100 pixels on the x and y axis), rotation (rotation 30 degree) and scaling (scaling by factor 2). In all these cases, the correlator c' have been evaluated using 1000 distinct watermarks for each case. The value of the correlator c' when we have used a false watermark is $-0.2 < c' < 0.2$. In figure 5, we show only results obtained with the correct watermark.

As in some cases, suspicious views coincide partially with watermaked ones, we have tested the robustness of this algorithm in such a situation. We tested the effect of:

1. Deleting some watermarked contour fragments (see figure 6 -b).
2. Replacing some watermarked contour fragments with not watermarked ones (see figure 6 -c)

The obtained results are shown on figure 7. We notice that if we replace less than 500 vertices (among 2389 vertices) or if we delete less than 1200 vertices (among 2389 vertices) we are still able to detect watermark's presence.

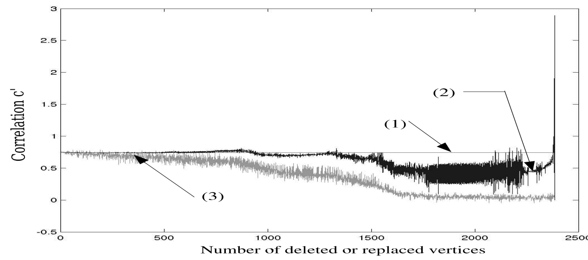

Figure 7. Results of attacks: deleting and replacing contour fragments. 1- Watermarked contour with no attacks. 2- : Watermarked contour with some deleted fragments. 3- : Watermarked contour with some replaced fragments.

4. CONCLUSION AND PERSPECTIVES

We have presented in this article a new scheme to watermark 3-D objects which is based on inserting the watermark in the apparent contour and extracting it from the 3-D object's represented views. This is an important approach since, as mentioned earlier, it is usually more frequent to locate and recover suspicious represented views of 3-D objects than the 3-D files themselves.

Several steps of our approach are still under construction and are not fully designed and validated, e.g. the automatic selection of caracteristic 2-D views, the exact way for the retriever to select the closest view point, the evaluation of the traditional compromise: capacity, robustness and visibility in our case (3-D/2-D approach) and so on.

By the time of the conference and camera ready paper delay, we plan to make progress on previously cited points as well as to conduct more experiments on diverse and numerous objects to better evaluate the performances and limits of our approach.

Long term works will also concentrate on 3-D object blind watermarking: the mark would be extracted from the 3-D object's represented views without knowing a having any idea about the set of parameters used for the projection.

REFERENCES

1. S. Ichikawa, H. Chiyama, and K. Akabane, "Redundancy in 3D polygon models and its application to digital signature," *Journal of WSCG* **10**(1), pp. 225–232, 2002.
2. T. Harte and A. G. Bors, "Watermarking 3D models," in *IEEE International Conference on Image Processing*, (Rochester, NY, USA), Sept. 2002.
3. O. Benedens, "Watermarking of 3D polygon based models with robustness against mesh simplification," in *SPIE Security and Watermarking of Multimedia Content*, pp. 329–340, 1999.
4. O. Benedens, "Geometry-based watermarking of 3D models," *IEEE Computer Graphics and Applications* **19**(1), pp. 46–55, 1999.
5. M. G. Wagner, "Robust watermarking of polygonal meshes," in *Geometric Modeling and Processing*, (Hong Kong, China), Apr. 2000.
6. R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models," in *ACM Multimedia*, pp. 261–272, (Seattle, Washington), Nov. 1997.
7. R. Ohbuchi, H. Masuda, and M. Aono, "A shape-preserving data embedding algorithm for NURBS curves and surfaces," in *Computer Graphics International*, pp. 170–177, June 1999.
8. F. P. M. Mitrea, T. Zaharia, "Spread spectrum robust watermarking for nurbs surfaces," *WSEAS Transactions on Communications* **3**, pp. 734–740, April 2004.
9. R. Ohbuchi and H. Masuda, "Managing cad data as a multimedia data type using digital watermarking."
10. C. Fornaro and A. Sanna, "Public key watermarking for authentication of CSG models," *Computer Aided Design* **32**(12), pp. 727–735, 2000.

11. E. Garcia and J.-L. Dugelay, "Watermark recovery from 2D views of a 3D video object," in *SPIE Electronic Imaging*, (Santa Clara, California), Jan. 2003.
12. J.-L. Dugelay, E. Garcia, and C. Mallauran, "Protection of 3D object usage through texture watermarking," in *European Signal Processing Conference (EUSIPCO)*, (Toulouse, France), Sept. 2002.
13. R. Basri and W. Jacobs, "Projective alignement with regions," in *IEEE Transactions on pattern analysis and machine intelligence*, 2001.
14. J. Feldmar, N. Ayache, and F. Betting, "3d-2d projective registration of free-form curves and surfaces," in *Computer Vision and Image Understanding*, 1997.
15. T. F. Ansary, D. Mohamed, and J. philippe Vandeborre, "3d model retrieval based on adaptive views clustering," in *3rd International Conference on Advances in Pattern Recognition (ICAPR)*, 2005.
16. V. Solachidis and I. Pitas, "Watermarking polygonal lines using fourier descriptors," in *IEEE Computer Society*, 2004.
17. H. Sonnet, T. Isenberg, J. Dittmann, and T. Strothotte, "Illustration watermarks for vector graphics," in *IEEE Pacific Conference on Computer Graphics and Applications*, 2003.
18. A. Giannoula, N. Nikolaidis, and I. Pitas, "Watermarking of sets of polygonal lines using fusion techniques," in *IEEE C2002*, 2002.