Protection of 3-D object usage through texture watermarking *

Jean-Luc Dugelay, Emmanuel Garcia, Caroline Mallauran
Institut Eurécom
2229 route des Crêtes, B.P. 193
06904 Sophia-Antipolis, France
dugelay@eurecom.fr, garciae@eurecom.fr, mallauran@eurecom.fr

ABSTRACT

In this paper, we describe a novel framework for watermarking 3-D objects via texture information. Instead of classical existing algorithms dealing with 3-D objects that operate on meshes in order to protect the object itself, the goal of our work is to retrieve information originally hidden in the texture image of the object, from *resulting images or videos* having used the 3-D synthetic object.
After developping a bit the theory and practical details of our new 3-D object watermarking scheme, we present preliminary results and make a comparison between the problem of recovering the watermark from a visualized textured 3-D object and that of recovering the watermark in an altered still image.

Introduction

More and more synthetic objects can be used in videos or images. Watermarking can be then useful for several purposes. In particular, viewers would like to check in videos if an object is synthetic or natural, to check if the use of a given object is legal or not, to access additional information concerning that object (e.g. copyright, date of creation, and so on.). This range of preoccupations will emerge with the increasing realism of synthetic objects, in particular realistic clones, and some upcoming standards such as MPEG-4 which will include possibilities to combine natural and synthetic information and allows users to easily manipulate data and make up virtual objects that look real.

Image watermarking is an emerging technique that allows to hide, in an invisible and robust manner, a message inside a digital audio-visual document. According to the targeted service, the message can contain some information about the owner (copyright), the picture itself (content authentication or indexing) or the buyer (non repudiation). It is then possible to recover the message at any time, even if the picture has been modified following one or several non destructive attacks (malicious or not) [1].

*This research is supported in part by the EU Certimark project [3]

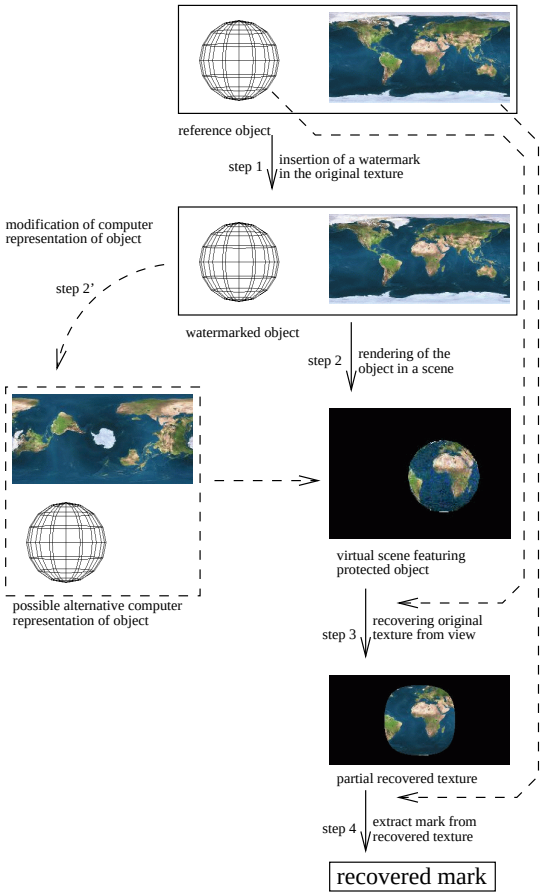


Figure 1: Principle

Originally mainly designed for the owner protection of still images, the range of both possible applications of watermarking technologies and possible covers have recently significantly increased [2].

To the best of our knowledge [4], all previous works dealing with 3-D objects are based on slight modifications performed on meshes via geometric or/and topologic data of 3-D objects. Typically, authors propose to modify either the 3-D coordinates of some points or the connectivity of triangles within a mesh. Interesting readers are invited to refer to the publications [5, 6, 7].

1 Texture watermarking of a 3-D object

1.1 General principle and hypothesis

Figure 1 shows the principle of our 3-D object watermarking scheme. Given a known 3-D object consisting of a geometric definition, a texture image and a texture mapping function, we protect the object by watermarking its texture (step 1) using a robust watermarking algorithm. This watermarked object can then be released for further representations in virtual scenes (step 2). We can then check that the represented object is protected by extracting the watermarked texture (step 3) from the represented views of the object, and by finally extracting the inserted watermark (step 4) from the recovered texture image.

Let us point out that this process does not depend on any modification of the internal representation of the released 3-D object (either geometry or texture) that may have been performed to try to erase the mark or to simplify the object or for any other purpose, as long the appearance of the rendered 3-D object remains the same (step 2').

One view of the 3-D object generally provides only a partial knowledge of the whole texture. So it is better, if possible, to recover partial texture images from several 2-D views of the 3-D object (e.g. face and profiles of the model of a human face) and then to merge them into a more complete texture image.

In order to reverse the tranformation undergone by the watermarked reference texture image until it is viewed, we need to know the projection matrix of the virtual camera and the reference 3-D object (geometry, texture, and texture mapping function)

Under those assumptions, our idea is not only feasible, but also resilient to any modification of the internal representation of the protected 3-D object as said above.

1.2 Practical implementation

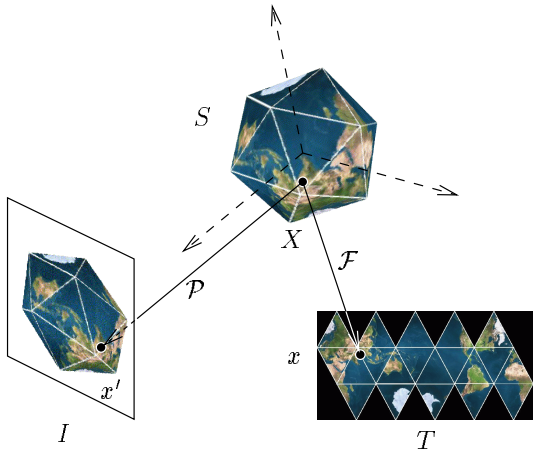


Figure 2: Texture mapping and viewing of a 3-D object

As shown in figure 2,

- let S be the set of 3-D points of the surface of the 3-D object,
- let I be an image of the rendered 3-D object,
- let T be the texture image to recover,
- let $\mathcal{F} : S \rightarrow T$ be the reference texture mapping function,
- let $\mathcal{P}$ be the projection from the system of coordinates of the reference 3-D object onto the image of the virtual camera.

The main problem is to accurately compute the projection matrix $\mathcal{P}$ of the virtual camera. In practice we have defined a VRML scene containing our test object at a known location and with a known orientation. So we knew the extrinsic parameters of the projection. But the visualization software we used did not allow us to know the intrinsic parameters of the virtual camera. So we placed a square of known size in the scene, parallel to the image plane, and picked the coordinates of its corners in the image to compute those parameters.

To recover the part of texture T visible in image I we

1. consider each pixel x of T
2. compute $X = \mathcal{F}^{-1}(x)$ (if it exists)
3. compute the pixel $x' = \mathcal{P}(X)$ of the image I where X is projected
4. check that X is actually visible at pixel x' of the image I
5. set the color of x to be that of x' in case X is visible in I at pixel x'

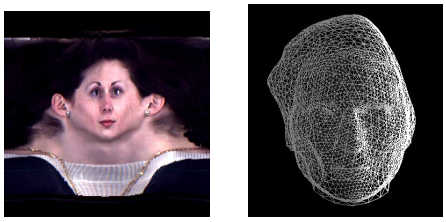
The fourth step is required because several points of the 3-D object could project on the same pixel in the image but (at least if we assume the object is opaque) only one would be seen. For this we used a Z-buffer technique associated with the observed image.

Since x' can have non-integer coordinates, it is not obvious what its color is. What we have currently done is rounding its coordinates to the closest integers and taking the color of the resulting pixel. We did not use any interpolation. It is not yet obvious that using interpolation would improve the recovered texture, at least for the specific purpose of extracting the watermark.

2 Preliminary results

2.1 Numerical results

For our experiments we have used the 3-D model of a human face (figure 3). We have inserted a 2-bit mark in the texture (figure 3a) using watermarking algorithms previously developped within the context of robust watermarks for still images [11]. The relatively small size of



a b

Figure 3: Reference model

a b c d e f

Figure 4: Recovering texture from various views

the mark allowed for important local and global duplication when hidden in the image, and hence for better robustness. We have then rendered the watermarked textured 3-D model in various known locations and orientations in the 3-D space of the virtual camera, whose internal parameters have been estimated using the image of a square as mentionned earlier.

Figures 4a, 4c and 4e show 3 rendered views (front, left and right view respectively) of the 3-D model and the corresponding recovered textures (figures 4b, 4d and 4f respectively) using the scheme described in the previous section.

Finally figure 5a shows the texture resulting from the merging of those 3 partial recovered textures, and figure 5b shows the merging of 3 other similar views (front, left and right) but where the model was placed much farther from the viewer (and thus appeared smaller in the image).

In three experiments, each involving three views of the model at a close enough distance like in figure 4 we have been able to recover the 2-bit mark inserted in the reference texture (figure 3a) from the recovered texture image (figure 5a). However, in a fourth experiment involving similar views but with a model approximately twice as far as in the first experiments, the 2-bit mark was not recovered from the recovered texture (5b).

a b

Figure 5: Merging recovered textures

2.2 Comments

Apart from the numerical results presented above in terms of hidden bits, we can make an interesting comparison between the alteration undergone by the original texture image in our texture recovery process, and well-known watermarking attacks in case of still images.

Those attacks, whose purpose are to slightly modify a watermaked image so as to make it impossible to recover the hidden mark, include but are not limited to:

- cropping of the image
- photometric attack (altering colors)
- geometric attack (locally warping the image)
- rotation, stretching, flipping of the image
- low level filtering (e.g. blurring)

On figure 6 we show an original texture (left), the texture recovered after 3-D visualization and the recovery error with respect to the original texture (top), and the texture altered by a watermark attack software (stirmark) along with the difference image (bottom).

The texture recovered after 3-D visualization can be seen as having undergone a crop attack (all the neck is

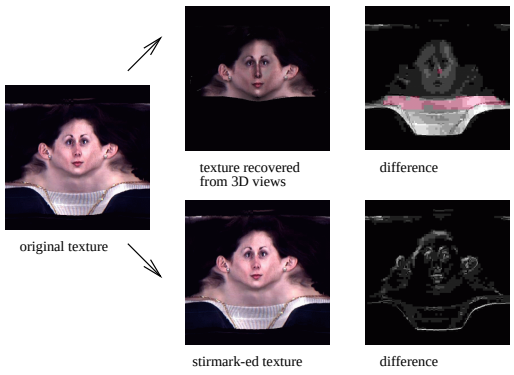


Figure 6: Comparison with stirmark

missing), a photometric attack (the colors are significantly altered due to synthetic lighting) and a filtering attack (subsampling due to the distance at which the texture is observed).

The texture altered by stirmark has undergone mainly a geometric attack (areas around top of head, mouth, eyes and ears are slightly warped as can be seen in the difference image) and possibly any additionnal combination of the attacks mentionned above.

All in all the alteration induced by 3-D visualization of a texture image (texture mapping followed by projection on the image plane) are not more severe than those induced by the use of a watermark attack software like stirmark. Even if the crop may be important, the local deformations are not as important as when stirmark is used.

Yet those alteration are quite severe and the use of a robust watermarking scheme is mandatory. For our experiments we have used a watermarking algorithm developped at Eurecom Institute [8, 9, 10] whose features include robustness to stirmark.

Concluding remarks

The idea developped in this paper is to protect the use of a 3-D object, instead of a computer representation of the object itself, by watermarking its texture in a given reference representation.

We have shown this idea to be possible under certain conditions: we must know the reference 3-D object (geometry, texture and texture mapping function), which means we recover the watermark in a non-blind mode, we must know the position and orientation of the object with respect to the virtual camera that was used to render it, and we must know the intrinsic parameters of this camera.

We have been able to recover a 2-bit mark from the texture recovered from three views of a 3-D face model, when this model was not observed from too far, even though the recovered texture can be a heavily altered version of the reference texture.

Yet this does not impair the principle of our method since we were able to recover a 2-bit mark when the distance of the observed object was fair. Our scheme is even resilient to any modification of the internal representation of the object (using a different texture mapping, or slightly modifying the 3-D mesh) as long as the rendering of the object remains the same. The main limitation is the assumption of known intrinsic and extrinsic virtual camera parameters.

Future work could be dedicated to modeling and quantifying the alteration of the texture due to the rendering process and the possible modification of the internal representation of the 3-D object, so as to better understand how to adapt the watermaking scheme to this particular application.

But the main concern, for our scheme to allow the protection of the use of a 3-D object in an unknown environment, remains to be able to recover the complete projection matrix (from the system of coordinates of the object to that of the image of the camera) from the observed images and from the knowledge of the reference 3-D object only.

References

[1] Katzenbeisser (S.), Petitcolas (F. A.P.), *Information Hiding - Techniques for Steganography and Digital Watermarking*, Artech House, Boston-London, 2000.

[2] *Special Session on Watermarking for Industrial Applications*, 2001 IEEE Fourth Workshop on Multimedia Signal Processing, October 3-5, Cannes.

[3] European Project - IST-1999-10987, *CERTIMARK - Certification for watermarking technique*, `http://www.certimark.org`.

[4] C. Mallauran, *Internship Report on 3-D Video Objects Watermarking*, Eurécom/ESSI-UNSA, September 2001.

[5] Olivier Benedens, *Watermarking of 3D polygon based models with robustness against mesh simplification*, Proceedings of SPIE: Security and Watermarking of Multimedia Contents, SPIE, pp. 329-340, 1999.

[6] Yutarou Ohbuchi, Hiroshi Masuda, Masaki Aono, *Geometrical and Non-Geometrical Targets for Data Embedding in Three-Dimensional Polygonal Models*, Computer communications, Elsevier, August 1998.

[7] E.Praun, H.Hoppe, A.Finkelstein, *Robust Mesh Watermarking*, ACM Siggraph 99 Conference Proceedings, Los Angeles, California, August 1999.

[8] J.-L. Dugelay, *Method for hiding binary data in a digital image*, Pending patent PCT/FR99/00485 (EURECOM 09-PCT), March 1999.

[9] J.-L. Dugelay & S. Roche, *Process for marking a multimedia document, such an image, by generating a mark*, pending patent EP 99480075.3 (EURECOM 11/12 EP), July 1999.

[10] J.-L. Dugelay & C. Rey, *Method of Marking a multimedia document having improved robustness*, pending patent EUP 99480075.3 (EURECOM 14 EP), May 2001.

[11] J.-L. Dugelay & C. Rey, *Image Watermarking for Owner and Content Authentication*, ACM Multimedia, Los Angeles, California, US, November, 2000.