WATERMARKING GRAPHICAL OBJECTS

Thomas Harte and Adrian G. Bors

Department of Computer Science, University of York, York YO10 5DD, U.K.
E-mail: adrian.bors@cs.york.ac.uk

Abstract: Watermarking has been used for protecting the copyright information in various data such as audio signals, images and video. Copyright protection of graphical objects and models is important for protecting author rights in animation, multimedia, computer-aided design (CAD), virtual reality, medical imaging, etc. Watermarking techniques in 3D graphical objects are facing new challenges because of the specific nature of the information available in 3D graphics. In this paper we suggest a blind watermarking algorithm for 3D models and objects. A string of bits is embedded in the geometrical structure of the graphical object by changing the locations of certain vertices. The criterion employed for selecting the vertices ensures a minimal visibility of the distortion in the watermarked object. The proposed 3D graphical object algorithm is applied in various 3D graphical models including characters and industrial objects.

1. INTRODUCTION

In the case of copyright protection, watermarking is used for embedding a code in the data for protecting the ownership of that data. When used for authentication the watermark is designed to protect the data content from possible tampering and modifications. The watermark should not interfere with the intended purposes of the data. A general watermarking algorithm has two stages: watermark embedding and detection. Most of the research on watermarking has concentrated on watermarking audio data, still images, or video [1, 2]. While audio data consists of one-dimensional time varying signals, images are 2D mappings of digital data distributed on a rectangular lattice. When applied on still images, the watermark algorithms can be classified in those which are embedded in the space domain [3] or in a transform domain such as that of the discrete cosine transform (DCT) [4]. Meanwhile, graphical objects are described by a set of vertices, defined as 3D vectors, which are connected by line segments or by polygon surfaces, and a set of normals. Algorithms employed in still images cannot be used in a straight forward way for graphical objects due to the different nature of the data to be watermarked.

Watermarking 3D graphical objects or models is very important in many areas including cinematography, virtual reality as well as in computer aided design (CAD). Graphical characters or graphical layouts of technical objects are initially created by a specific author or designer. Afterwards they can be handled by a different person in a larger graphical context. In such a situation, the initial author or designer may want to maintain the copyright of the object or to authentify it. Authentication of graphical objects has been studied in [5] and [6]. In order to protect the copyright, the algorithm embeds in the object information related to the owner of the object, usually an ownership code. A copyright protection watermarking algorithm employing modifications in the histograms of 3D object surface normals was suggested by Benedens in [7]. Watermarking of 3D polygonal meshes in spectral domain has been shown to be robust under various attacks in [8]. However, these two approaches need the original object in the detection stage. Such algorithms can be considered only for private key watermarking systems.

In this paper we suggest an algorithm for watermarking 3D objects where we do not need the original object in the detection stage. In this sense the proposed watermarking algorithm can be considered as a public key technique. The watermark is locally embedded in two stages. In the first stage, a chain of vertices and their neighbourhoods are selected aiming to a minimal visibility of the modifications to be performed. The selected vertices are ordered according to a distance criterion. In the second stage, bounding volumes are modelled from the selected neighbourhoods and the corresponding vertices are moved inside the bounding volume when embedding a bit of one or outside for embedding a bit of zero. In the detection stage the embedded bits are retrieved by checking their relative location with respect to the bounding volumes. The sequence of retrieved bits is matched against the given watermark code.

The paper is organised as follows. The requirements for a 3D object watermarking system are outlined in Section 2. The procedure for selecting the locations where the watermark is embedded is described in Section 3 while the watermarking algorithm is presented in Section 4. Experimental results are provided in Section 5 while the conclusions are drawn in Section 6.

2. REQUIREMENTS FOR 3D OBJECT WATERMARKING ALGORITHMS

The conditions and requirements imposed on watermarking algorithms when applied on various multimedia products such as audio, video or image processing have been defined in various studies such as those from [1, 2, 4]. Volumetric images resulted from computer tomography, MRI or by other processing means are represented by voxels located on a 3D lattice. Three-dimensional regions of interest are extracted from such images using segmentation and modelling. Geometrical objects are usually characterised by a set of vertices joined by line segments while the neighbouring surface orientations are defined by surface normals. A graphical object can represent the model of an object extracted from a volumetric image, can be designed using a CAD system or can be derived from an image using shape-from-shading techniques. In the case of a graphical object one has usually to deal with a low volume of data which carries mainly geometrical information.

Some of the 3D graphical object watermarking requirements are similar to those for audio or image watermarking. Such requirements include the non-perceptibility of changes brought to the watermarked model. Additionally, a system must ensure the presence of a watermark in a graphical object after performing certain transformations. Such transformations can be necessary for the usability of that data or can be performed by an attacker with the purpose of destroying the watermark. A graphical object watermark should be robust to two classes of possible object transformation algorithms: geometrical and topological. Geometrical transformations include rotation, scaling or combinations of local transforms. Possible topological transformations are: changing the order of vertices in the object description file, polygon simplification, mesh altering or cropping parts of the object. Other processing algorithms include object compression and encoding, smoothing algorithms, corruption with noise, etc.

In any watermarking algorithm we have to deal with a trade-off between the visibility of the watermark and its robustness. Most of the approaches suggested for watermarking graphical objects assume the knowledge of the original, un-watermarked object [7, 8]. Such algorithms can work in a private key watermarking system but in many applications they are not very useful. In the following we describe a blind watermarking algorithm in which no *a priori* knowledge about the original object is required in the detection stage.

3. SELECTING OBJECT REGIONS FOR WATERMARKING

The proposed algorithm embeds an watermark by changing the location of certain vertices. The embedding of the proposed 3D watermarking algorithm has two stages. In the first stage the algorithm finds the locations which are appropriate for embedding the watermark. Such locations are chosen based on a minimal perceptibility of the changes brought to the object.

We suggest a geometrical criterion for selecting the locations where to embed the watermark. If we consider $V_i \in \mathcal{O}$, a vertex whose 3D coordinates are given by the vector $\mathbf{V}_i$, from the 3D model $\mathcal{O}$, we define its neighbourhood as all the vertices connected to it :

$$\mathcal{N}(V_i) = \{ \forall V_j, j = 1, \ldots, N \mid |V_j V_i| > 0 \} \qquad (1)$$

where $|V_j V_i|$ denotes set cardinality and counts the number of points on the line segment joining the two vertices, while N denotes the total number of vertices from the neighbourhood $\mathcal{N}(V_i)$. The point V_i is not considered as being part of its own neighbourhood. We derive the ellipsoid which roughly models the neighbourhood of V_i. The centre of this ellipsoid is given by :

$$\mu_i = \frac{\sum_{j \in \mathcal{N}(V_i)} \mathbf{V}_j}{N} \qquad (2)$$

while the shape of the ellipsoid can be calculated as the second order moment (variance) of the sites composing the neighbourhood :

$$\mathbf{S}_i = K \frac{\sum_{j \in \mathcal{N}(V_i)} (\mathbf{V}_j - \mu_i)(\mathbf{V}_j - \mu_i)^T}{N} \qquad (3)$$

where K is a normalisation factor. Neighbourhoods for which would result a singular matrix $\mathbf{S}_i$ are excluded. We obtain a singular matrix $\mathbf{S}_i$ when all the vertices from the neighbourhood $\mathcal{N}(V_i)$ are located on the same plane. The ellipsoid which locally describes the vertices in the neighbourhood of V_i, forms a bounding volume which is modelled by :

$$(\mathbf{x} - \mu_i)^T \mathbf{S}_i^{-1} (\mathbf{x} - \mu_i) = 1 \qquad (4)$$

where $\mathbf{x}$ is a vector located inside the bounding volume described by $(\mu_i, \mathbf{S}_i)$.

Similarly with image watermarking, we embed the watermark in those regions of the 3D object where it is less likely to be observed. Such regions are usually characterised by containing small segments joining the vertices. If we would modify the location of vertices ending large segments we may produce significant distortion in the graphical object. We calculate the distance from a vertex to its neighbourhood as :

$$D_i = \sum_{j \in \mathcal{N}(V_i)} \|\mathbf{V}_j \mathbf{V}_i\| \qquad (5)$$

where $\|\mathbf{V}_j \mathbf{V}_i\|$ represents the length of the line joining V_i and its neighbour V_j. We order all the vertices in the object according to their D_i and we consider a certain visibility threshold T. The vertices which have the distance to their neighbourhood smaller than a certain threshold T are considered for embedding the watermark :

$$D_i < T. \qquad (6)$$

The threshold T can be chosen based on a perceptibility criterion. Any vertex from the neighbourhood of a selected vertex is excluded from the set of possible candidates for embedding the watermark. In this case, the geometry of the bounding ellipsoid $\mathbf{S}_i$ remains unchanged even when we change $\mathbf{V}_i$. This is a very important property since we would like to recover the same neighbourhoods in the detection stage. Let us consider that we have a watermark code with B bits. All the vertices fulfilling (6) are split in sets of B vertices. Each set of vertices is ordered according to the distance between the centre of its ellipsoid and those of the neighbouring ellipsoids. Such a distance is measured as :

$$i = \arg\min_j \sum_{k=1, k \neq j}^{B-1} \|\mu_k - \mu_j\|^2 \qquad (7)$$

where we consider only the closest $B-1$ ellipsoids. Thus we associate a bounding ellipsoid to each bit from the watermark label. This procedure associates a label of B bits to a certain region from the graphical object.

4. THE WATERMARKING ALGORITHM

The watermark is embedded in a set of B neighbourhoods modelled as ellipsoids whose equations are given by (4). The number of times the watermark is repeatedly embedded in the object is given by the integer of the ratio between the total number of vertices that fulfil (6) and B. In order to increase the robustness to object cropping we embed the watermark locally, according to (7). For watermarking a specific bit we define two regions in the volume associated with a vertex V_i. A similar approach was employed in [4] for watermarking images, by defining linear and circular regions in the space of the DCT transform coefficients. The vertex to be marked is moved in the corresponding volume region according to its bit from the watermark label.

We suggest two approaches for embedding the code. In the first approach we extract two parallel planes from the geometry of a neighbourhood $\mathcal{N}(V_i)$. Let us consider known the normals $\mathbf{N}_i$ at the vertices from the neighbourhood $\mathcal{N}(V_i)$. In the case that these normals are not provided with the object, they can be derived using simple geometry from the knowledge of adjacent planes. The orientation of the two bounding planes is denoted by $\mathbf{Q}_i$ and it is given by averaging the orientations of all surface normals from that neighbourhood :

$$\mathbf{Q}_i = \frac{\sum_{j \in \mathcal{N}(V_i)} \mathbf{N}_j}{N} \tag{8}$$

The two planes are located at equal distance e_i on both sides of the centre μ_i, derived in (2). This distance is calculated as the local neighbourhood variance, projected along the direction of $\mathbf{Q}_i$:

$$e_i = \frac{\sum_{j \in \mathcal{N}(V_i)} [(\mathbf{V}_j - \mu_i) \cdot \mathbf{Q}_i]^2}{N} \tag{9}$$

where $|\cdot|$ denotes the scalar product. In the case when embedding a bit of one, we project the vertex $\mathbf{V}_i$ along the direction of $\mathbf{Q}_i$ inside the volume defined by the bounding planes such that :

$$|(\hat{\mathbf{V}}_i - \mu_i) \cdot \mathbf{Q}_i| < e_i \tag{10}$$

where $\hat{\mathbf{V}}_i$ is the position of the watermarked vertex. In the case when embedding a bit of zero, the vertex is projected outside the volume defined by the bounding planes. The projection defined by (10) ensures a minimal distortion in the graphical object.

In the second approach we consider the ellipsoids defined by (4) as bounding volumes. The vertex V_i is projected along the direction of $V_i \mu_i$ inside the bounding ellipsoid for embedding a bit of one. In this case we have :

$$(\hat{\mathbf{V}}_i - \mu_i)^T \mathbf{S}_i^{-1} (\hat{\mathbf{V}}_i - \mu_i) < 1 \tag{11}$$

where $\hat{\mathbf{V}}_i \in \mu_i \mathbf{V}_i$. This procedure is exemplified in Figure 1 where the vertex $\mathbf{V}_i$ changes its location to $\hat{\mathbf{V}}_i$ in order to embed a bit of one. The neighbourhood of this vertex is shown in this figure together with its corresponding ellipsoid modelled by S_i. A bit of zero is embedded by moving the vertex V_i outside this ellipsoid. No change is performed if the vertex is already located according to the relationship of its corresponding bit. The algorithm enforces that geometrical changes produced by (10) or (11) do not modify the selection order of the vertices. After embedding B bits we proceed to embed the next B bits, preferably in a different part of the graphical object in order to maximise the robustness to model cropping.

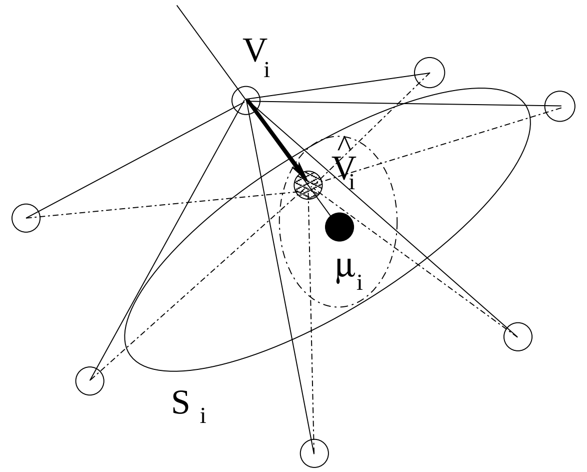


Fig. 1. Embedding a bit of one in the vertex V_i.

In the detection stage we select the chain of marked vertices and their neighbourhoods as it was explained in Section 3 for the embedding stage. For the chosen vertices and their neighbourhoods we verify the relationship (10) for the bounding volume delimited by planes, or (11) when we consider bounding ellipsoids for embedding code bits. A sequence of bits is retrieved from the graphical object. Eventually, a XOR operation between the code and the bits retrieved from the object decides if the watermark was embedded or not.

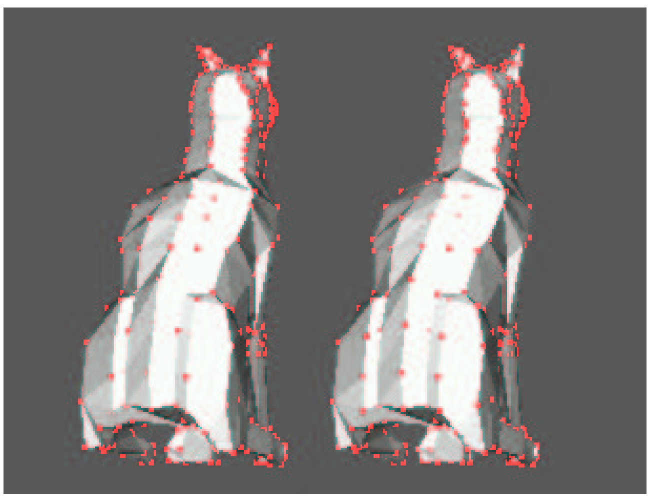
5. EXPERIMENTAL RESULTS

We have applied the proposed watermarking algorithm on several graphical objects representing graphical characters and industrial models. We present the results for applying the watermarking algorithm on two graphical objects. The first represents a dog and has 649 vertices and 1286 faces. Its frontal view is shown in the left part of Figure 2a while its back view is displayed in the left part of Figure 2b. The watermarked "dog" model using bounding planes for embedding the label according to (10) is shown in the right part of the views from Figure 2. In Figure 3 we display the graphical model of a screwdriver which has been watermarked using ellipsoids as bounding volumes (11). The original is shown on top and the watermarked object is located at the bottom of Figure 3. This graphical object has 2060 vertices and 4076 faces. There are 54 vertices where we can embed watermarks according to (6) for the "dog" object and 43 possible bit holder vertices for the "screwdriver" model. The length of the watermark was considered as $B = 20$ in both examples. We evaluate the mean square error (MSE) between the vertices composing the original and the watermarked object, respectively. MSE can be written as :

$$MSE = \frac{1}{M} \sum_{i=1}^{M} \|\mathbf{V}_i - \hat{\mathbf{V}}_i\|^2 \tag{12}$$

where M is the total number of vertices composing the model and the Euclidean distance is calculated in 3D. The MSE for the "dog" object is $4.98 * 10^{-3}$ while for the "screwdriver" is $3.51 * 10^{-5}$. Hardly any differences can be observed between the watermarked and the original graphical objects. The watermark was successfully recovered from these images. The watermark can be retrieved after scaling, rotation or any combination of global geometrical transformations. It is easier to find several watermark bit-holder vertices in the structure of a graphical object which contains many vertices.

(a) Front part of a graphical model representing a dog



(b) Back part of a graphical model representing a dog

Fig. 2. The graphical model from left is the original while that from right represents the watermarked object.

6. CONCLUSIONS

In this paper we have presented a public digital watermark algorithm for 3D graphical objects. The proposed watermarking algorithm does not require the original object in the detection stage. This makes the proposed approach suitable for a public key 3D object watermarking system. A set of vertices and their neighbourhoods are selected and ordered. The embedding stage consists of changing the vertex locations according to the geometry of their neighbourhoods. Two different bounding volumes are considered for embedding the watermark label: using separation planes and ellipsoids. A bit of one is embedded by moving the vertex inside the bounding volume, while a bit of zero is embedded by moving the vertex outside the bounding volume, ensuring also a minimal distortion in the resulting watermarked model. The watermark algorithm has been applied to both graphical characters and to 3D models of mechanical objects. A potential area of application for the proposed algorithm is for watermarking 3D models extracted from volumetric images.

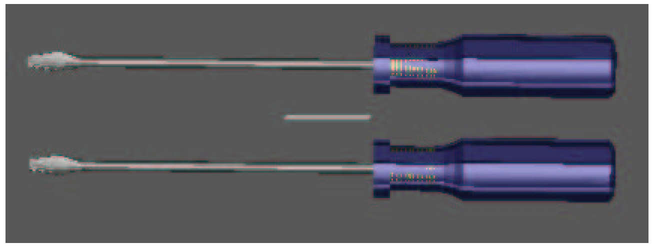


Fig. 3. Graphical model of a screwdriver where top represents the original and bottom the watermarked object.

REFERENCES

[1] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. of the IEEE*, vol. 87, no. 7, pp. 1079–1107, July 1999.

[2] G. Voyatzis and I. Pitas, "The use of watermarks in the protection of digital multimedia products," *Proc. of the IEEE*, vol. 87, no. 7, pp. 1197–1207, July 1999.

[3] C. D. Coltman and A. G. Bors, "Hierarchical watermarking depending on local constraints," in *Proc. of IEEE Inter. Conf. on Image Proc., vol. III*, Thessaloniki, Greece, 7-10 Oct 2001, pp. 1011–1014.

[4] A. G. Bors and I. Pitas, "Image watermarking using block site selection and DCT domain constraints," *Optics Express*, vol. 3, no. 12, pp. 512–522, 1998.

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

[6] C. Fornaro and A. Sanna, "Private key watermarking for authentication of CSG models," *Computer-Aided Design*, vol. 32, pp. 727–735, 2000.

[7] O. Benedens, "Geometry based watermarking of 3D models," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 46–55, 1999.

[8] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama, "Watermarking 3D polygonal meshes in the mesh spectral domain," in *Proc. of Graphics Interface*, Ottawa, Canada, June 2001, pp. 9–17.