WATERMARKING OF 3D MODELS FOR DATA HIDING

P. Daras [1], *D. Zarpalas* [2], *D. Tzovaras* [2] *and M. G. Strintzis* [1,2] Fellow, IEEE

[1]Information Processing Laboratory
Electrical and Computer Engineering Department
Aristotle University of Thessaloniki, Greece
540 06 Thessaloniki, Greece

[2]Informatics and Telematics Institute
1st Km Thermi-Panorama Road,
Thessaloniki 57001, Greece
Email:strintzi@eng.auth.gr

ABSTRACT

A novel method for the watermarking of 3D models is proposed, which is robust to geometric distortions such as rotation, translation and uniform scaling. After proper positioning and alignment of the 3D models, a watermark is embedded in the vertices of the model using a robust technique for modifying imperceptibly the location of a subset of the 3D model vertices. The watermark can be used as a link to an identifier for the 3D model and the entire system can be used for data hiding applications. One application is the use of watermark as a link to 3D model descriptors for content-based search and retrieval. Experimental results show the ability of the proposed method to aforementioned attacks. The proposed method is also robust against vertex reordering attack.

1. INTRODUCTION

Digital media, such as images, audio and video can be rapidly manipulated, reproduced and distributed over information networks. Due to potential copyright violation many authors are discouraged to publish and distribute their work digitally. Therefore, a lot of research has been carried out to protect the copyright of digital media; digital watermarking is one such protection technique. Further, watermarking techniques have long been used for multimedia annotation, with indexing and labelling information.

While much work has been done on watermarking of 2D multimedia content, only few methods have been proposed for the watermarking of 3D models. In [1], three watermarking algorithms for 3D models, were proposed: the triangle similarity quadruple embedding algorithm, the tetrahedral volume ratio embedding algorithm and the mesh density pattern embedding algorithm, which provide many useful insights into mesh watermarking. Benedens proposed several watermarking algorithms: the watermarking that is robust to mesh simplification and affine transform [2, 3, 4], high capacity watermarking [5] and a combined watermarking system [6]. Praun *et al.* [7] presented a technique for embedding secret watermarks using a spread spectrum technique. In [8] the proposed method is based on distributing the information over the entire model via vertices scrambling.

In this paper, a novel technique is proposed for 3D model watermarking for data hiding, which is robust against geometric attacks such as rotation, translation and uniform scaling. Prior to watermarking embedding a procedure that ensures invariance to the aforementioned attacks is used. The watermarking technique, which is based on the Cylindrical Integration Transform (CIT), is used in order to embed a watermark in the nodes of the 3D model. The most promising data hiding application that can be used with the proposed technique, is 3D content-based search and retrieval where the watermark can be used as a link to 3D model descriptors. The original 3D model is not necessary during the watermark detection. Finally, the proposed watermarking technique is robust against vertex reordering attack.

2. 3D MODEL PREPOSSESSING

A 3D model M is comprised of a set of vertices $\mathbf{V}$ and a set of connections between the vertices. Each vertex $\mathbf{v}_i$ has three coordinates in the cartesian space, $\mathbf{v}_i = \{x_i, y_i, z_i\}$. Before applying the proposed technique a canonical position and orientation is achieved for each 3D model. The steps are the following:

1. **Model Rotation and Translation.** Let $\boldsymbol{U}$ be the class of vectors for all pairs of vertices of the 3D model. The vector $\boldsymbol{u}_1$ is calculated, where: $|\boldsymbol{u}_1| = max\{|\boldsymbol{u}| : \boldsymbol{u} \in \boldsymbol{U}\}$. Further, the most distant vertex $\mathbf{v}_d$ from $\boldsymbol{u}_1$, and its projection $\mathbf{O}'$ to $\boldsymbol{u}_1$ are found. The point $\mathbf{O}'$ is the new origin of the model. The model is translated so that the new origin coincides with the old origin. That way translation invariance is accomplished. Finally, the model is rotated so that $\boldsymbol{u}_1$ coincides with the z axis and $\boldsymbol{u}_2$ coincides with the x axis. Rotation invariance follows.

2. **Model Scaling.** In order to achieve scaling invari-

ance, the distance d_{max} between the new origin and $\mathbf{v}_d$, is calculated. Then, the model is scaled so that $d_{max} = 1$. At this point, scaling invariance is also accomplished.

The translated, rotated and scaled model is then placed into a bounding sphere with radius $R_a = d_{max}$.

3. THE CYLINDRICAL INTEGRATION TRANSFORM

Let $\boldsymbol{\eta}$ be the unit vector in $\Re^3$. The Cylindrical Integration Transform (CIT), is defined as the summation of the model's vertices $\mathbf{v}_i$ which lie inside the cylinders $CYL(\boldsymbol{\eta}_i)$ with radius T and axis the line $L(\boldsymbol{\eta}_i)$ (Figure 1). Each line (and cylinder) begins from the origin and ends at the points $\mathbf{P}_i$. The CIT is given by:

$$CIT(\boldsymbol{\eta}_i) = \sum_{\mathbf{v}_j \in CYL(\boldsymbol{\eta}_i)} \mathbf{v}_j \qquad (1)$$

where $i = 1, \ldots, N_{CYL}$, N_{CYL} is the total number of cylinders with orientation $\boldsymbol{\eta}_i$ and $j = 1, \ldots, J$, J the total number of points $\mathbf{v}_j$.

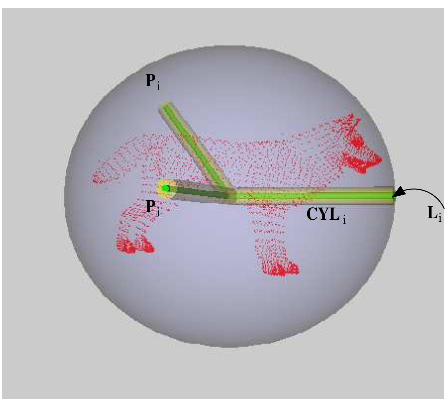


Fig. 1. The Cylindrical Integration Transform.

4. 3D MODEL WATERMARKING

Given a 3D model M, robustness to translation, rotation and scaling is achieved by applying the process described in Section 2 prior to watermark embedding and detection. Then, equation (1) is applied to the model's vertices producing the vector:

$$\boldsymbol{u}_{CIT} = [CIT(\boldsymbol{\eta}_i)], \qquad i = 1, \ldots, N_{CYL} \qquad (2)$$

The values $\boldsymbol{u}_{CIT}$ are sorted in descending order and the K, $(K \in Z)$ first, which carry the most important information of the model, are selected. In order to ensure that the vectors $\mathbf{u}_1, \mathbf{u}_2$ used in model's rotation and translation (Section 2, will not be altered after the watermark embedding, the cylinders which contain the vertices that form these vectors, are excluded from the watermarking process.

4.1. Watermark Embedding

The proposed watermarking method embeds K bits in the vertices of M. The procedure of embedding the watermark is shown in Figure 2.

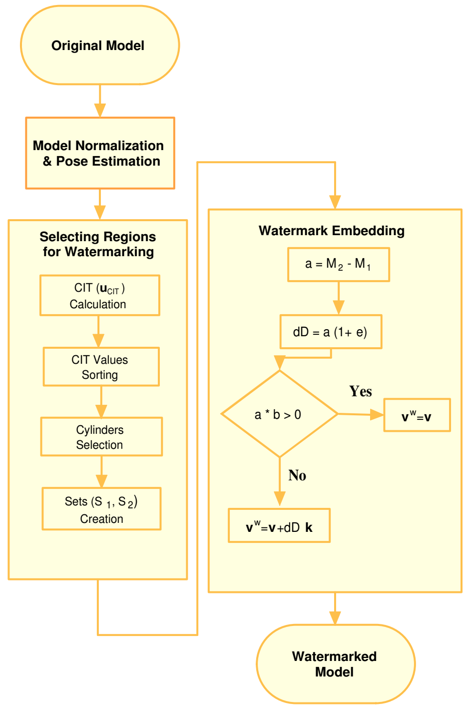


Fig. 2. Block diagram of the watermark embedding procedure

The watermark is inserted as follows:

1. Each selected $\boldsymbol{u}_{CIT}(i), i = 1, \ldots, K$ corresponds to the cylinder $CYL(\boldsymbol{\eta}_i)$ and furthermore to a set of vertices lying into cylinder $CYL(\boldsymbol{\eta}_i)$. The length of the projection $l_{ij} = \mathbf{v}_{ij} \cdot \boldsymbol{\eta}_i$ of each vector $\mathbf{v}_{ij}$ onto the line $L(\boldsymbol{\eta}_i)$, is calculated. All vertices that lie on the same cylinder are sorted in descending order according to their l_{ij} and the first half of them form the set $S_{i1} = \{\mathbf{v}_{ij1}\}$, $j = 1, \ldots, N_{S1}$, while the remaining form the set $S_{i2} = \{\mathbf{v}_{ij2}\}$, $j = N_{S1} + 1, \ldots, N_{CYL\boldsymbol{\eta}_i}$. The sets S_{i1}, S_{i2}, which contains the same number of vertices, are chosen to be the sets for embedding in each of their vertices, a watermark.

2. A watermark $B = \{b_i, i = 1, \ldots, K\}$, $b_i \in \{-1, 1\}$ is embedded in each vertex of set S_{i1} or S_{i2} by modifying their location, according to the following: In each set, the distance D_{ij} between each vertex $\mathbf{v}_{ij} \in CYL(\boldsymbol{\eta}_i)$ and the axis $L(\boldsymbol{\eta}_i)$ is calculated. Then, the mean values of the D_{ij} for each set are found:

$$M_{i\beta} = \frac{1}{N_{S_i}} \sum_{j=1}^{N_{S_i}} D_{ij\beta} = \frac{1}{N_{S_i}} \sum_{j=1}^{N_{S_i}} \mathbf{v}_{ij\beta} \cdot \boldsymbol{\kappa}_i, \qquad (3)$$

where $i = 1, \ldots, K$, $\boldsymbol{\kappa}_i$ is unit perpendicular vector to $\boldsymbol{\eta}_i$ ($\boldsymbol{\kappa}_i \cdot \boldsymbol{\eta}_i = 0$) with direction to the line (from outside to inside), $\beta = \{1, 2\}$ denotes each set and N_{S_i} is the total number of vertices in set $S_{i\beta}$.

Then, the difference $a_i = M_{i2} - M_{i1}$ is calculated. If $a_i > 0$ the watermark is embedded in the vertices of the set S_{i2}, otherwise, in those of the set S_{i1} according to the formula:

$$\mathbf{v}_{ij}^W = \begin{cases} \mathbf{v}_{ij} & \text{if } a_i \cdot b_i > 0 \\ \mathbf{v}_{ij} + dD_i \boldsymbol{\kappa}_i & \text{if } a_i \cdot b_i < 0. \end{cases} \qquad (4)$$

where $\mathbf{v}_{ij}$ denotes the original vertex, $\mathbf{v}_{ij}^W$ denotes the watermarked vertex, and dD_i is the displacement:

$$dD_i = |a_i| \cdot (1 + e) \qquad (5)$$

where e is a positive sufficiently small number.

Easily can be proved that the term a_i^W, after the watermark embedding, has the same sign with b_i.

4.2. Watermark Detection

The block diagram of the watermark detection procedure is depicted in Figure 3. Let M^d be the model after geometric attacks. The watermark is detected as follows:

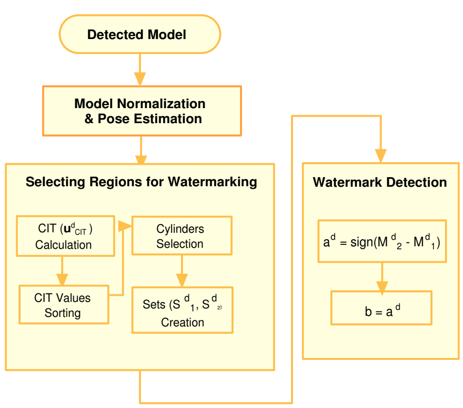


Fig. 3. Block diagram of the watermark detection procedure

1. The M^d is translated, rotated and scaled according to the procedure described in Section 2.

2. The CIT vector $\boldsymbol{u}_{CIT}^d$ of the model M^d is calculated and its values are sorted in a descending order. Then, the first K values are selected which correspond to specific cylinders $CYL(\boldsymbol{\eta}_i)$.

3. As in step 1 of the embedding procedure, in each selected $CYL(\boldsymbol{\eta}_i)$ the two sets $S_{i1}^d = \{\mathbf{v}_{ij1}^d\}$,$S_{i2}^d = \{\mathbf{v}_{ij1}^d\}$ are found according to their $l_{ij}^d = \mathbf{v}_{ij}^d \cdot \boldsymbol{\eta}_i$. Obviously, the sets S_{i1}^d, S_{i2}^d are identical to the sets S_{i1}, S_{i2} because the projections of their vectors (vertices) onto the axis of the cylinder they belong to, do not change. Further, $M_{i1}^d = M_{i1}^W$ and $M_{i2}^d = M_{i2}^W$ and $a_i^d = a_i^W$.

4. Thus, the watermark sequence can be easily extracted using the formula:

$$a_i^d = sign(M_{i2}^d - M_{i1}^d), \qquad i = 1, \ldots, N_{P_i} \qquad (6)$$

and since $a_i^d (= a_i^W)$ and b_i have the same sign, the watermark sequence b_i is extracted.

This procedure is repeated for all watermark bits. The proposed technique can be efficiently combined with 3D model indexing where watermark detection can be used to retrieve the corresponding descriptor vector. The extracted watermark sequence of bits forms the identifier which links the 3D model to its descriptor vector. Every time a model is used as a query model, watermark detection is used so as to retrieve the corresponding descriptor vector, which can be further used in a matching algorithm.

5. EXPERIMENTAL RESULTS

The proposed 3D model watermarking technique for 3D model indexing was tested using 100 models from our database [9]:

1. *Geometric attacks and vertex reorder.* The geometric attacks that were tested are translation, rotation and uniform scaling. Due to the preprocessing steps applied to a model prior to embedding and detecting a watermark sequence, the percentage of correct extraction was 100% for $K = 16, 24, 32$ bits. Similarly, changing the order of the vertices, the percentage of correct extraction was, also, 100% for $K = 16, 24, 32$ bits, since the coordinates of the vertices remain unaltered after this type of attack. It should be noted that for $K = 32$ bits the total number of models is 10^{11}, which is a sufficiently large number of models for content-based search and retrieval applications.

2. *Imperceptibility.* In order to measure the imperceptibility of the embedded watermark the Signal-to-Noise ratio (SNR) was calculated for the 100 3D models. The following formula was used:

$$SNR = \frac{\sum_{i=1}^{N_M} x_i^2 + y_i^2 + z_i^2}{\sum_{i=1}^{N_M} (x_i - x_i^W)^2 + (y_i - y_i^W)^2 + (z_i - z_i^W)^2} \quad (7)$$

where N_M is the total number of M's vertices, x_i, y_i, z_i are the coordinates of the vertex $\mathbf{v}_i$ before the embedding of the watermark and x_i^W, y_i^W, z_i^W are the coordinates of the same vertex after the embedding of the watermark.

Figure 4 illustrates the results for this experiment. The original and the watermarked 3D models are shown in Figure 5 for various values of K. It is obvious that the embedded watermark is imperceptible.

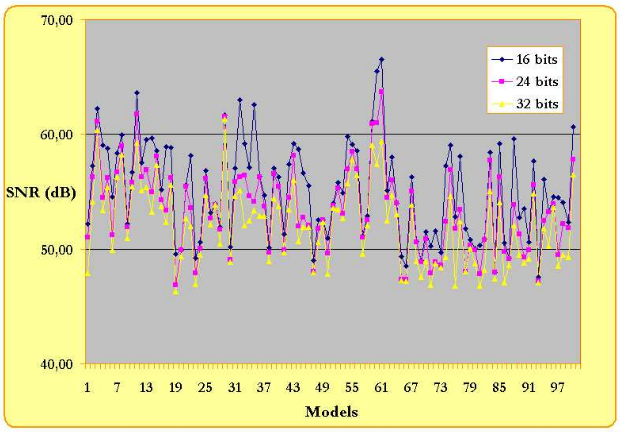
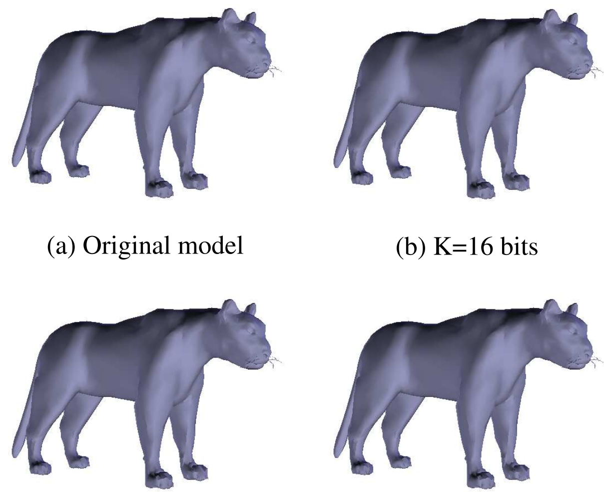


Fig. 4. SNR measure of 3D models.

3. *Watermark extraction time.* Additionally, the proposed method was tested in terms of time needed for the detection of the watermark sequence. The results show that 0.02 seconds are needed, on average for the detection of the watermark sequence. Thus, the proposed scheme is very appropriate for use as an efficient tool for web-based, real-time application for 3D model content-based search and retrieval.

6. CONCLUSIONS

A novel technique for 3D model watermarking for indexing, was presented. The watermarking method is (a) robust to geometric attacks such as translation, rotation and uniform scaling, (b) robust to points reordering attack and (c) imperceptible. Finally, the extraction of the watermark sequence is very fast and accurate and can be effectively used in real-time 3D model content-based search and retrieval applications.



(a) Original model (b) K=16 bits

(c) K=24 bits (d) K=32 bits

Fig. 5. Visual results of the watermarking procedure.

7. REFERENCES

[1] R. Ohbucci, H. Masuda and M. Aono, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE Journal on Selected Areas in Communications*, 16(4) pp. 551-560, 1998.

[2] O. Benedens, "Affine Invariant Watermarks for 3D Polygonal and NURBS Based Models," *Springer Information Security, Third International Workshop*, vol. 1975, pp. 15-29, Australia, Dec. 2000.

[3] O. Benedens, "Geometry-based Watermarking of 3D Models," *IEEE Computer Graphics and Applications*, vol. 19, pp. 46-55, Jan.-Feb. 1999.

[4] O. Benedens, "Watermarking of 3D Polygon Based Models with Robustness against Mesh Simplification," *In Proc. of SPIE: Security and Watermarking of Multimedia Contents*, vol. 3657, pp. 329-340, 1999.

[5] O. Benedens, "Two High Capacity Methods for Embedding Public Watermarks into 3D Polygonal Models," *In Proc. of the Multimedia and Security Workshop at ACM Multimedia 99*, pp. 95-99, Orlando, Florida, 1999.

[6] O. Benedens, "Towards Blind Detection of Robust Watermarks in Polygonal Models," *Eurographics 2000*, vol. 19, No. 3 2000.

[7] E. Praun, H. Hoppe and A. Finkelstein "Robust mesh watermarking," *In Proc. of SIGGRAPH 99*, pp. 69-76, 1999.

[8] Yu Zhi-qiang, Horace H. S. Ip, and L. F. Kowk "Robust Watermarking of 3D Polygonal Models based on Vertice Scrambling," *In Proc. of Computer Graphics International CGI'03*, 2003.

[9] The ITI database, http://3d-search.iti.gr/