2010 International Conference on Signal Acquisition and Processing

Watermark Embedder Optimization for 3D Mesh Objects using Classification Based Approach

Rakhi C. Motwani, Mukesh C. Motwani, Bobby D. Bryant and Frederick C. Harris Jr.
University of Nevada, Reno USA

Akshata S. Agarwal
University of Pune, INDIA

Abstract

This paper presents a novel 3D mesh watermarking scheme that utilizes a support vector machine(SVM) based classifier for watermark insertion. Artificial intelligence(AI) based approaches have been employed by watermarking algorithms for various host mediums such as images, audio, and video. However, AI based techniques are yet to be explored by researchers in the 3D domain for watermark insertion and extraction processes. Contributing towards this end, the proposed approach employs a binary SVM to classify vertices as appropriate or inappropriate candidates for watermark insertion. The SVM is trained with feature vectors derived from the curvature estimates of a 1-ring neighborhood of vertices taken from normalized 3D meshes. A geometry-based non-blind approach is used by the watermarking algorithm. The robustness of proposed technique is evaluated experimentally by simulating attacks such as mesh smoothing, cropping and noise addition.

1. Introduction

The increased use of 3D objects in animation, virtual reality, computer aided design (CAD), architecture, archeology and scientific data visualization has attracted the research community to explore watermarking techniques for the copyright protection of 3D objects. Watermarking research during the past decade has mostly focused on text, images, audio and video. Machine learning techniques for embedding and extracting watermarks have been widely investigated for images [1], [2], [3], [4], [5], audio [6], [7] and video [8]. However, artificial intelligence techniques have yet to be explored for the domain of 3D graphics. The novelty of this paper lies in evaluating the use of support vector machine to identify optimal locations for embedding the watermark in the 3D mesh object.

SVM classifiers [9] are based on a class of hyperplanes i.e. a decision surface which characterizes the boundary between the classes of the data. A set of training data or feature vectors is used to create the SVM classifier which predicts the class label of future data sets that are not provided in the training set. The SVM tries to find the optimal hyperplane which gives the largest margin of separation between the classes. The optimal hyperplane is a weighted combination of a subset of the elements of the training data set known as support vectors. A linear decision surface is expressed by the equation:

$$w.x + b = 0 \tag{1}$$

where, x is the input vector, b is the bias and

$$w = \sum_{i=1}^{N} \alpha_i s_i y_i \tag{2}$$

where α is the weight, s is the support vector and y is the class label. A nonlinear decision surface makes use of a kernel function $K(.,.)$ that transforms the data to a higher dimensional space where the data set is separable. The kernel function must satisfy the Mercers condition [10] and be expressed as a dot product in some space. The nonlinear decision surface is expressed by the equation:

$$\sum_{i=1}^{N} \alpha_i y_i K(s_i, x) + b = 0 \tag{3}$$

The equation for the three basic kernels are:
Linear: $K(x_i, x_j) = x_i^T x_j$,
Polynomial: $K(x_i, x_j) = (x_i.x_j + c)^d$,
Radial basis: $K(x_i, x_j) = exp(-\frac{1}{2\rho} \|x_i - x_j\|^2)$

This rest of the paper is organized as follows. Section 2 gives a brief overview of literature that utilizes SVM for watermarking and presents various applications of SVM in the domain of 3D objects. Section 3 describes the proposed watermarking algorithm and the SVM training phase. Experimental results are presented in section 4. Finally some conclusions are drawn in section 5.

2. Literature Review

While SVM has not been used so far for watermarking of 3D objects, a review of literature reveals that SVM has been used for various application is 3D such as object recognition[11], face detection[12], face recognition[13], texture classification[14], head pose estimation[15], content-based database search[16], and shape processing[17]. This section explores some of this literature along with work that utilizes SVM for digital media watermarking. The literature

978-0-7695-3960-7/10 $26.00 © 2010 IEEE
DOI 10.1109/ICSAP.2010.83

is analyzed to determine what feature vectors are used for training, which kernel function is employed and what classification is achieved by the SVM.

The authors in [18] use a linear SVM for the extraction of watermark from digital images. The feature vectors are derived from the watermark embedded positions and consist of a reference watermark and the deviation of the watermarked pixel from the mean of the value of its surrounding pixels. The trained SVM is then used to determine the value of the pixel under consideration during the watermark extraction phase. In [19] the authors describe a frequency domain based image watermarking algorithm that uses a binary SVM to classify an image's texture characteristic into two classes. Their technique divides the training images into 8x8 blocks, calculates the texture value for each block and uses the derived texture values for training the SVM. The SVM classification model is then used to classify a given images 8x8 blocks to decide the watermark embedding intensity adaptively.

In [20] a two-class SVM is used to locate the optimal embedding positions for the watermark in an audio signal. The feature vectors for training are the means of the absolute value of the coarse signal and the local maximal peaks of the detail signal in each wavelet subband of an audio segment. For higher average and peak values, the feature vector belongs to class 1 that represents optimal location for watermark insertion. For lower average and peak values the feature vector belongs to class 0 that is not suitable for watermark insertion. A radial basis kernel function is used for the SVM.

In [21] a video watermarking approach is proposed that uses SVM for watermark embedding and extraction. The watermark comprises of a training sequence and a digital signature. For training the embedding SVM, feature vectors are derived based on the difference of blue channels among a watermarked video frame and its neighbors. The trained SVM then embeds the watermark in the rest of frames. The detector takes frames and the training sequence from the watermark to train the extracting SVM and uses it to extract watermarks from the remaining frames. The final watermark is the average of these extracted watermarks.

The authors in [14] use a multi-class SVM for classification of 3D textures using a histogram model. Invariant features area extracted from images of textured surfaces which are transformed by 3D translation and rotation, by means of a non-linear histogram model. These invariant features are used as input feature vectors of the SVM in order to do classification in high-dimensional spaces.

The authors in [22] discuss 3D object classification using SVMs. The COIL image library of 3D objects is used for experiments and three different resolution levels of the images are used as feature vectors. A polynomial kernel of degree 2 is used for higher resolution level feature vectors and a linear kernel is used for low resolution representations. Binary SVMs are constructed for each pair of objects, where each SVM is trained as a classifier for one class against another class. A multi-class pattern recognition system is obtained by combining these binary SVMs. In order to classify test data, pair-wise competition between all the SVMs is performed and tennis game tournament strategy is used to classify a given test object.

3. Approach

The proposed approach employs SVM for the watermark embedding phase. Fig. 1 gives an overview of the watermarking system. SVM is a supervised learning algorithm and requires a training stage. The trained SVM is then used in the watermark embedder. Once the SVM is trained, feature vectors extracted from any 3D mesh model can be fed to the SVM classifier to decide which vertices are appropriate for watermark insertion. The watermark extractor retrieves the imperceptible watermark inserted during the insertion stage. Robustness of the watermark is evaluated by comparing the correlation between original watermark and attacked watermark using correlation.

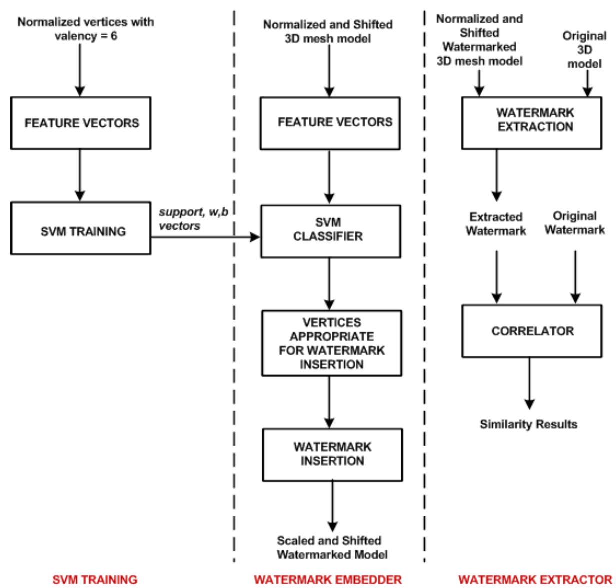

Figure 1. System Block Diagram

The choice of feature vectors to be classified is critical for the performance of SVM as a classifier. In our earlier work [23], we used angle variation between surface normals and the average normal corresponding to a vertex to determine the vertex smoothness measure [24]. This measure was then used to determine the amount of watermark to be added. The feature vector is a set of angles derived by computing the orientation of the surface normals to the average normal of the triangular faces that form a 1-ring neighborhood for a vertex, as shown in Fig. 2 and 3. This feature vector represents the curvature of the 1-ring vertex neighborhood. The length of the feature vector is equal to the valence of the vertex, which is the count of how many other vertices the vertex is connected to in the 3D model. Since feature vectors used for training of SVM have to be of fixed length, vertices with valence 6 are selected for feature extraction since most of the vertices in a 3D model have valence 6.

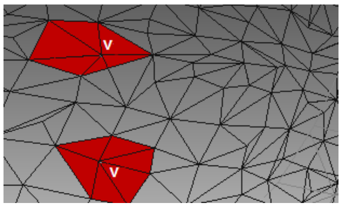

Figure 2. Example of 1-Ring Neighbourhood For Vertices of Valence 6

The steps for extracting feature vectors are given below. Step 1: Consider a vertex v with valence equal to 6 from the 3D mesh model. Let M be the number of its adjacent faces which is equal to 6. Find normals n_i to each face which is formed by v and its neighboring vertices v_i.

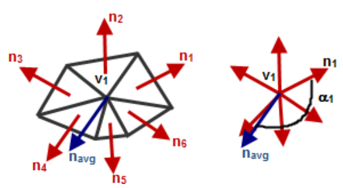

Figure 3. Surface Normals(in red) and Average Normal(in dark blue) For a 1-Ring Vertex Neighbourhood

Step 2: Find the average resultant vector n_{avg} of all the above normals passing through v.

$$n_{avg} = \frac{1}{M} \sum_{i=1}^{M} n_i \tag{4}$$

Step 3: Now compute angles between each pair of n_i and n_{avg}. Feature vector $F = [\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6]$

$$\alpha_i = \cos^{-1}\left(\frac{n_i.n_{avg}}{|n_i|.|n_{avg}|}\right) \tag{5}$$

3.1. SVM Training

During the training stage, feature vectors are fed to the SVM as input along with a class label corresponding to either 1 or -1 for each corresponding input feature vector. The class label for each feature vector is determined manually by a human operator. Random amount of information is added to a vertex and if the information added caused perceptible distortion, the vertex with valence 6 is labeled as -1. If the information added was imperceptible the vertex is labeled as 1. 100 sets of vertex rings with different geometrical structures are extracted from 7 normalized 3D objects and labeled appropriately deciding whether to insert the watermark or not. The labellings of output vectors is a manual process thereby transferring human intelligence to the classifier. Radial Basis Function (RBF) is used as the kernel function. The output of the training stage is the support vectors, the weight vector and the bias defining the optimal hyper plane for separation of the feature vectors in to class labels of 1 and -1.

3.2. Watermark Embedding

3D objects are normalized by scaling the size of the mesh to fit in a bounded cube with the vertices of the diagonal of the cube ranging from (-1,-1,-1) to (1,1,1) and the centre of mass of the vertices shifted to origin. 3D objects other than semiregular meshes can also be used, provided feature vectors are extracted for vertices with valence 6. The extracted feature vectors are then fed to the SVM classifier which was trained in the earlier stage. The SVM gives an output of -1(corresponding to the decision that the vertex should not be selected for watermarking) or output of 1(corresponding to the decision that the vertex should be selected for watermarking). The feature vectors extracted from the 3D model may not have been used in the training set. A random number sequence W(the watermark data) is added to the vertices which have SVM output of 1 according to the following equation:

$$v'(x, y, z) = v(x, y, z) + KW \tag{6}$$

where v' is the watermarked vertex, K is the scaling factor.

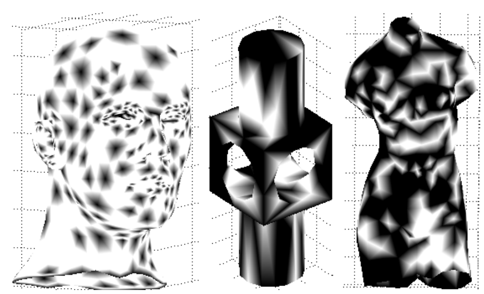

Figure 4. Watermark Locations Indicated By White Regions (Black regions denote vertices that are not modified by the watermarking algorithm)

The 3D model is then scaled back to its original size and the center shifted from origin to the original center. The watermark inserted in the 3D objects are randomly distributed throughout the model as shown in Fig. 4.

3.3. Watermark Extraction

To extract the watermark from a given watermarked or attacked 3D model, the model is normalized and scaled to the origin. Using the the original model, the difference in vertex coordinates is determined. A correlation measure as defined in Eq. 7 and Eq. 8 is used to get the degree of similarity between the extracted watermark and the watermark added during the embedding stage.

$$AmountofCorrelation = \frac{w + w' * correlation}{W} \quad (7)$$

$$correlation = \sum_{i=1}^{n} \frac{A(i).A'(i)}{|A(i)| * |A'(i)|} \quad (8)$$

where,
w = count of vertices not attacked in watermarked model,
w'= count of vertices attacked in watermarked model,
W = total number of vertices in the watermarked model,
$A(i)$ =$x(i) + y(i) + z(i)$, the x, y, z co-ordinates of the ith vertex in the attacked model, and
$A'(i) = x'(i) + y'(i) + z'(i)$, the x, y, z co-ordinates of the corresponding ith vertex in the watermarked model.

4. Experimental Results

The algorithm has been implemented in *Matlab* using the *Least Squares SVM* Toolbox. This section presents experiments on robustness of the watermark and evaluates the imperceptibility of the embedded watermark. Fig. 5 shows the various attacks performed on the watermarked 3D models. Hausdorff distance and Vertex Signal-to-Noise Ratio (VSNR) quantify the visual differences between the original and watermarked objects and are very low as shown in Table 1 indicating very good imperceptibility of the watermark.

Model Name	Number of Vertices	Number of Modified Vertices	VSNR (dB)	Hausdorff Distance
Mannequin	1681	1230	119.25	0.002038
Mechanical	175	61	107.57	0.004738
Venus	711	229	128.75	0.001307

Table 1. Comparison Of Original Model With Watermarked Model

The following attacks are simulated on the watermarked 3D objects. The correlation coefficient between the attacked watermark and the original one is shown in Table 2. A threshold value of 0.7 is used to determine survival of the watermark.
1) Translation, Rotation and Scaling: The algorithm is completely resistant to uniform scaling and affine attacks since the model is normalized prior to watermark insertion. Any uniform rotation, translation and scaling operations on the

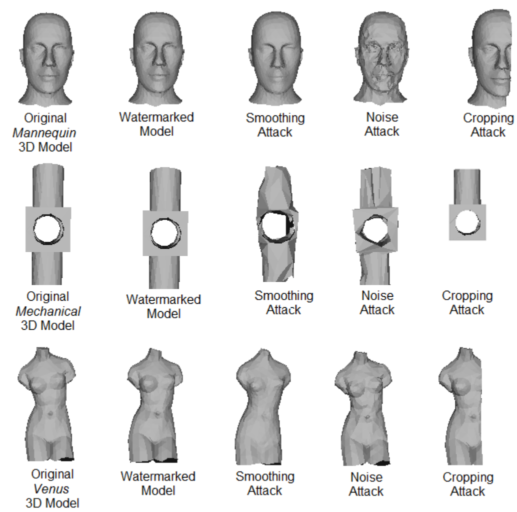

Figure 5. Attacks on Watermarked 3D objects

watermarked models gives correlation coefficient of 1.0.
2) Noise on geometry: Addition of Gaussian noise consists of adding random noise with mean 0 and variance 0.01 to vertex coordinates (see Fig. 5). A noise level of 100% i.e. all vertices are modified by additive noise, destroys the watermark yielding lower than 0.7 correlation values. However, the watermark survived noise levels of 50% or less.
3) Mesh smoothing: Laplacian smoothing when applied to the watermarked model smooths the sharp edges in the model by applying a low pass gradient filter to the vertices. The watermark does not survive mesh smoothing operations.
4) Cropping: Various experiments that crop sections of the 3D model verified that the watermarked can be recovered even when a part of the mesh is removed, as shown in Fig. 5.

3D Model	Correlation Value		
Name	Smoothing	Cropping	Noise
Mannequin	0.5407 (Laplacian Smoothing 1 time)	0.9546 (627 vertices cropped)	0.5874 (Gaussian Noise added to 1681 vertices)
Mechanical	0.4976 (Laplacian Smoothing 1 time)	0.7571 (48 vertices cropped)	0.6798 (Gaussian Noise added to 175 vertices)
Venus	0.6327 (Laplacian Smoothing 1 time)	0.9694 (193 vertices cropped)	0.633 (Gaussian Noise added to 711 vertices)

Table 2. Correlation Results For Various Attacks

5. Conclusions

This paper has investigated and illustrated the potential of support vector machines for embedding a watermark in 3D mesh objects. The performance of the algorithm is heavily dependent on the quality of training feature vectors and the size of the training set. Experimental results show good performance in terms of imperceptibility and robustness for the watermarking algorithm. Future work will include experimenting the use of various kernel function for the SVM and deriving feature vectors of different lengths for vertices with different valence. SVM will also be evaluated in future work for not only selecting the vertices to be watermark but also to determine how much watermark to add to the selected vertex. Using SVM in the watermark detection phase is an additional task to explore.

References

[1] Y.-G. Fu, R.-M. Shen, L.-P. Shen, and X.-S. Lei, "Reliable information hiding based on support vector machine," *Informatica*, vol. 16, no. 3, pp. 333–346, 2005.

[2] C.-Y. Chang and S.-J. Su, "A neural-network-based robust watermarking scheme," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, Oct. 2005, pp. 2482–2487.

[3] F. Meng, H. Peng, Z. Pei, and J. Wang, "A novel blind image watermarking scheme based on support vector machine in DCT domain," in *International Conference on Computational Intelligence and Security*, vol. 2, Dec. 2008, pp. 16–20.

[4] X.-Y. Wang, H.-Y. Yang, and C.-Y. Cui, "An SVM-based robust digital image watermarking against desynchronization attacks," *Signal Process.*, vol. 88, no. 9, pp. 2193–2205, 2008.

[5] S.-H. Yen and C.-J. Wang, "SVM based watermarking technique," *Tamkang Journal of Science and Engineering*, vol. 9, no. 2, pp. 141–150, 2006.

[6] X.-Y. Wang, P.-P. Niu, and W. Qi, "A new adaptive digital audio watermarking based on support vector machine," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 735–749, November 2008.

[7] S. Kirbiz, Y. Yaslan, and B. Gunsel, "Robust audio watermark decoding by nonlinear classification," in *European Signal Processing Conference*, Sept. 2005.

[8] C.-J. Wang, S.-H. Yen, and P. S. Wang, "A multimedia watermarking technique based on SVMs," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 8, pp. 1487–1511, 2008.

[9] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods)*. CUP, 2000.

[10] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[11] M. Pontil and A. Verri, "Support vector machines for 3d object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 637–646, Jun 1998.

[12] M. Mousavi, K. Faez, and A. Asghari, "Three dimensional face recognition using svm classifier," in *IEEE/ACIS International Conference on Computer and Information Science*, May 2008, pp. 208–213.

[13] J. Huang, V. Blanz, and B. Heisele, "Face recognition with support vector machines and 3d head models," in *Pattern Recognition with Support Vector Machines, First International Workshop, SVM 2002*, 2002, pp. 334–341.

[14] L. Meng, F. Ping, and S. Shenghe, "3D texture classification using 3D texture histogram model and SVM," in *International Conference on Electronic Measurement and Instruments*, July 2007, pp. 2–940–2–943.

[15] M. Krinidis, N. Nikolaidis, and I. Pitas, "3D head pose estimation using support vector machines and physics-based deformable surfaces," in *International Symposium on Signal Processing and Its Applications*, Feb. 2007, pp. 1–4.

[16] M. Elad, A. Tal, and S. Ar, "Directed search in a 3D objects database using svm," HP Labs www.hpl.hp.com/techreports/2000/HPL-2000-20R1.pdf.

[17] F. Steinke, B. Schölkopf, and V. Blanz, "Support vector machines for 3D shape processing. Eurographics," in *Computer Graphics forum*, vol. 24, no. 3, 2005.

[18] D. Lakshmi, R. Ganesh, S. Marni, R. Prakash, and P. Arulmozhivarman, "SVM based effective watermarking scheme for embedding binary logo and audio signals in images," in *IEEE TENCON*, Nov. 2008, pp. 1–5.

[19] F. Yang and L. Li, "An adaptive, SVM-based watermarking in frequency domain," in *International Conference on Wavelet Analysis and Pattern Recognition*, vol. 2, Aug. 2008, pp. 465–469.

[20] X.-Y. Wang, P.-P. Niu, and W. Qi, "A new adaptive digital audio watermarking based on support vector machine," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 735–749, 2008.

[21] S.-H. Yen, H.-W. Chang, C.-J. Wang, P. Wang, and M.-C. Chang, "A scene-based video watermarking technique using svms," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, Dec. 2008, pp. 1–4.

[22] D. Roobaert and M. V. Hulle, "View-based 3D object recognition with support vector machines," in *IEEE International Workshop on Neural Networks for Signal Processing*, 1999, pp. 77–84.

[23] R. Motwani and F. Harris, "Robust 3D watermarking using vertex smoothness measure," in *International Conference on Image Processing, Computer Vision, and Pattern Recognition*, July 2009.

[24] P. Flynn and A. Jain, "On reliable curvature estimation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings CVPR,*, Jun 1989, pp. 110–116.