Accepted for the *2004 International Conference on Cyberworlds, Nov. 18-20, 2004*
(Proceedings is to be published by the IEEE press.)

Watermarking a 3D shape model defined as a point set

Ryutarou Ohbuchi[1], Akio Mukaiyama[1], Shigeo Takahashi[2]
[1]Interdisciplinary Graduate School of Medicine and Engineering,
University of Yamanashi, 4-3-11 Takeda, Yamanashi-shi, Japan
[2]Graduate School of Arts and Sciences, University of Tokyo
ohbuchi@acm.org, mukaiyama.akio@lab.ntt.co.jp, takahashis@acm.org

Abstract

This paper discusses a method to watermark a 3D shape model defined as a set of unoriented points. Our approach is to perform frequency domain analysis of the shape of the 3D point set for watermarking. Our method applies the mesh-spectral analysis technique proposed by Zachi Karni, et al. to the point set model for frequency domain shape analysis. As the technique requires connectivity of vertices for the analysis, our method generates a non-manifold mesh from the point set and uses its connectivity for the analysis. The watermarks embedded by using the method can be detected, to some extent, after such attacks as cropping, (simulated) simplification, similarity transformation, and additive random noise applied on the watermarked point set model.

Keywords: digital watermarking, geometric modeling, mesh spectral analysis, computer graphics.

1. Introduction

Protection of intellectual property (IP) rights is an important requirement for a fair use of digital contents in the cyberworld. One of the tools to achieve IP right protection is the digital watermarking. While there have been many papers on digital watermarking audio, 2D image, and movie data [5], relatively small literature exists on digital watermarking of 3D models. To the authors' knowledge, all the digital watermarking algorithms for 3D shape model are for 3D mesh models.

Recently, unoriented and oriented *point set* (or *point cloud*) models of 3D shapes have gained the status of one of the mainstream 3D shape representations. Compared to a polygonal mesh representation, a point set representation has an advantage, among others, of being lightweight to store and transmit, due to its lack of connectivity information. With the proliferation of point set models, e.g., due to the broadcasting of point set models, the demand to protect IP rights for point set models would increase.

This paper presents a new algorithm for informed-detection, robust watermarking of (*unoriented*) *point-set 3D shape models*. The algorithm assumes, as its watermarking target, an unoriented point set 3D model that sampled surfaces of the underlying 3D shape with sufficient sampling density. An unoriented point has 3D position but no orientation. As an informed detection watermark, the extraction of the watermark requires the original model (the *reference model*) as well as the watermarked (and possibly attacked) model. As a robust watermark, the embedded information (watermark) is expected to survive, as far as possible, various attacks, e.g., cropping and additive random noise.

The algorithm proposed in this paper takes the same frequency-domain shape modification approach as the 3D mesh-watermarking algorithm by Ohbuchi et al [17]. The fundamental idea of Ohbuchi's algorithms is that a low-frequency distortion of a 3D shape is less noticeable to human beings when viewed than a high-frequency shape distortion. For example, slight elongation of the entire model much is less noticeable than the higher frequency shape changes such as ripples on the mesh surface. Similar idea is employed by Sorkine and others to reduce visible distortion in their shape compression work [19].)

In order to modify 3D shapes in their frequency domain, the mesh-watermarking algorithm by Ohbuchi et al [17] uses the *mesh spectral analysis* technique proposed by Karni, et al. [13]. As the mesh spectral analysis requires the connectivity of the vertices of the mesh, the watermarking algorithm proposed in this paper first generates a mesh model from the given point set.

[1] Mr. Akio Mukaiyama is currently working for the NTT Information Sharing Platform Laboratories. The work described in this paper has been performed while he was in the graduate school at the University of Yamanashi.

Generating a “good looking” 2D manifold mesh from a point set model is in general quite difficult. In our case, however, visual appearance is not an issue in meshing. The mesh does not have to be manifold, nor good looking. It just has to capture the 3D shape of the point set model enough so that the watermark can be embedded. We used a very simple 3D Euclidian neighbourhood-based meshing method, and the method worked well enough for watermarking according to our experiments. Once a mesh is created, mesh spectral analysis is performed, and the low-frequency component of the model is modified to embed watermark bit string into the mesh. By removing vertex connectivity from the watermarked mesh, a point set model with watermark is created.

1.1 Previous work

Digital watermarking add structures called watermarks to various target data objects so that information encoded in the watermark is added to the target data. The watermark must not interfere with the intended purposes of the target object (e.g., viewing for a 2D image data object) and the watermark should ideally be inseparable from the target object. Embedded watermarks can be used, in combination with the other technical and non-technical means (e.g., law) to enforce copyright, add comments, detect tampering, or to identify rightful purchasers of the data. Please refer to books, such as [5], for digital watermarking in general.

For 3D shape models, the polygonal mesh representation has been the targets for watermarking of the 3D shape models [14, 15, 16, 11, 2, 21, 18, 3, 21, 23, 17, 3]. These methods modify either vertex coordinate or vertex connectivity of the meshes so that the modifications do not interfere with the intended uses of the models, i.e., gaming experience.

In the field of image watermarking, frequency domain techniques for watermarking is quite popular due to their advantages. For example, by modifying the spatial-frequency band human beings are not very sensitive to, a watermark embedded in an image can be made less visible. Or, by targeting the coarse shape features, the watermarks embedded are less susceptible to low-pass filtering or additive random noise. In addition, by doing so, watermarks become harder to remove since coarse shape features are often essential to the image.

Transformed domain techniques have found uses in the field of watermarking 3D shapes. A pioneering work by Satoshi Kanai [11] applied the wavelet-based 3D mesh shape analysis technique to the frequency-domain watermarking of polygonal mesh shapes. Their watermarks are resistant against affine transformation, partial resection, and random noise added to vertex coordinates. However, their method required the mesh to have 1-to-4 subdivision connectivity. Praun et al [18] reported an informed-detection, robust mesh-watermarking algorithm. It used the Hoppe’s progressive meshes [10] technique for the shape transformation. Significantly, they added the shape normalization and geometry-resampling step to the watermark extraction so that the watermarks can be extracted after mesh simplification and other operations that preserve shape but modify vertex connectivity. Yin et al. [23] reported an informed-detection, robust mesh-watermarking algorithm that also works in a transformed domain. It is based on a multiresolution decomposition of polygonal mesh shapes developed by Guskov et al [8]. Ohbuchi et al [17] employed the mesh spectral analysis technique proposed by Karni et al [13] for the frequency domain decomposition of 3D mesh shapes. Using geometry resampling during extraction the watermark withstood mesh simplification and other attacks that modify mesh connectivity. The method presented in this paper is based on Ohbuchi’s mesh-spectral analysis based watermarking method for meshes.

2. The Watermarking Algorithm

The *point-set watermarking* algorithm described in this paper takes the same frequency-domain approach to shape modification as the *mesh-watermarking* algorithm we have previously reported [17]. However, there is a significant difference; while a mesh model has connectivity information necessary for the mesh spectral analysis, a point-set model doesn’t. Our algorithm watermarks a point set model by converting the point set into a (non-manifold) mesh model before applying the mesh watermarking algorithm similar to Ohbuchi et al [17]. The watermark embedding and extraction processes can be summarized as below. Extraction of the watermark requires the watermarked (and possibly compromised) point set model, the original point set model and the information describing the clusters.

Embedding:

(1) Create disjoint set of clusters from the point set.

(2) Generate a (non-manifold) mesh for each cluster of the points. The clustering information is saved so that it can be in the extraction process.

(3) Apply mesh spectral analysis for each mesh.

(4) Modify the mesh spectral coefficients according to the information bit string to be embedded.

(5) Inverse transform the coefficients back into the vertex coordinate domain.

(6) Discard the vertex connectivity of the mesh to generate a watermarked point set model.

Extraction:

(1) Align the watermarked point set with the original point set so as to remove similarity transformation applied.

(2) Resample the geometry of the watermarked point set by using the original (reference) point set.

(3) Re-create the clusters on the reference point set. Transfer the clustering onto the resampled watermarked point set.

(4) Generate a mesh for each cluster.

(5) Perform the mesh spectral analysis on the corresponding clusters of the reference and watermarked point sets.

(6) Demodulate the signal and extract the embedded bit string by comparing the coefficients of the corresponding clusters.

Details of each step will be explained in the following.

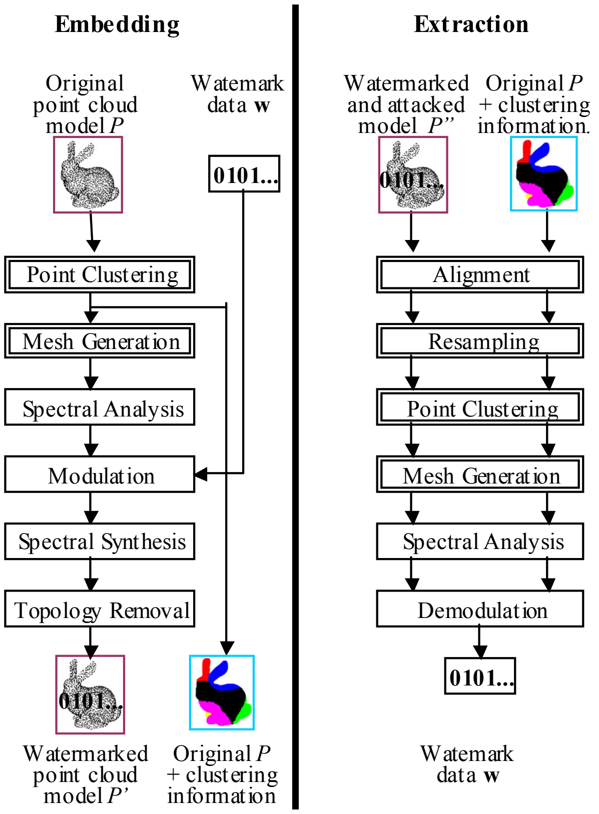


Figure 1. Overview of our 3D point set watermarking algorithm.

2.2 Watermark Embedding

Clustering: The clustering step divides the point-set model into several subsets, or clusters, each of which becomes the target of watermarking. Clustering serves several purposes. First, it reduces computational costs for the mesh spectral analysis step. Second, by embedding multiple copies of the same message into the clusters, the watermark as a whole becomes more resistant against cropping, local deformation, and other attacks. Clustering also helps if the user wants to specify one of more specific regions in the shape model for watermarking. (For example, a user might wish to concentrate watermark on the ears of the rabbits.)

A cluster is generated as a set of points that are close to each other based on 3D Euclidian distance. To generate clusters, the user selects several *seed points* in the point set model to be the centers of the clusters. The algorithm starts from the seed points, and gradually expands the clusters by incrementally adding, for each cluster, a point closest to the cluster at the time. The expansion stops if a cluster touches a cluster (either itself or another cluster), or the number of point in the cluster exceeds the given threshold. The cluster size is a user specified parameter.

Figure 2(b) shows an example of five clusters, each consisting of about 4,000 points, generated from the “bunny4” point-set model (Figure 2(a)) containing 34,839 points.

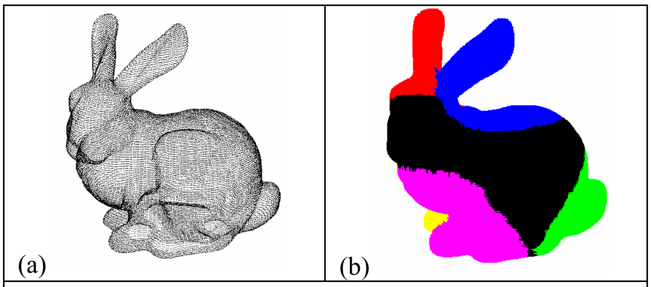


Figure 2. A point set model (34,839 points) (a). The non-black regions in (b) are the 5 clusters of size approximately 4,000 points each.

Meshing: Generating a visually appealing 2D manifold mesh from a point set model is not an easy task. In our case, however, the mesh just has to capture 3D shape of the point set enough for the watermarking. Our meshing algorithm is a very simple method based on 3D Euclidian neighbourhood. Given a point p of the point set model P, the meshing algorithm connects the p with all the points within the sphere of radius r centred at the point p (See Figure 3 (a)). The resulting mesh in general is not a 2D manifold mesh. If the surface of the shape is sampled densely enough, the resulting mesh M approximates the 2D manifold surface of the shape well. However, if the sampling is not dense enough, the resulting mesh may contain holes. Also, points sampled from separate surfaces may be "sewn-together" by the mesh, essentially producing a non-manifold "3D" mesh, as illustrated in Figure 3(b). For the purposes of viewing and NC milling, for example, such holes and sawn-through points are not desirable. However, such meshes can be used for shape analysis, and for digital watermarking, of a 3D shape model defined as a point set.

Figure 4 shows an example of shape approximation using the meshing algorithm described above. In this example, we generated the mesh from the point set model of a rabbit containing 1,633 points (a relatively coarse sampling). We analyzed the shape of the resulting mesh by using the mesh-spectral analysis technique of Karni et al [13], and reconstructed the shape by using subsets of the spectral coefficients. The shape approximation thus obtained closely resembled those obtained by using the mesh spectral analysis that used 2-manifold meshes of the same model containing the same vertices.

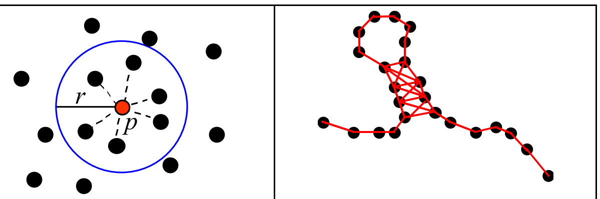
The neighbourhood radius *r* for the meshing is chosen so that the *average* number of connected points is equal to a given parameter N_n. In other words, N_n is the *average* of *degrees* (*valences*) of vertices of the generated patch of M_i for the cluster *i*. If N_n is too small, the mesh would have holes, and ultimately, at N_n=0, become a set of points. On the other hand, a large N_n will slow down the mesh spectral analysis as the the resulting mesh Laplacian matrix become denser with a wider diagonal band.

We compared the attack resistance of watermarks produced by various values of N_n. While the values $N_n = 12$, $N_n = 24$ and $N_n = 36$ produced similar results, smaller values of N_n resulted in loss of for their attack resistance. We thus use the empirically determined value of $N_n = 12$ in the experiments described in Section 4.

Mesh-Spectral Analysis: The mesh spectral analysis step transforms the shape of the mesh M_i created from a cluster in the point set *P* into its "frequency" domain. Given a set of *n* points from a cluster having connectivity, we generate the $n \times n$ mesh Laplacian matrix $\mathbf{K} = \mathbf{D} - \mathbf{A}$. Here, $\mathbf{D}$ is a diagonal matrix whose diagonal element $\mathbf{D}_{ii} = d_i$ is a *degree* (or *valence*) of the vertex *i*, and $\mathbf{A}$ is an adjacency matrix of the polygonal mesh whose elements a_{ij} are defined as below;

$$a_{ij} = \begin{cases} 1, & \text{if vertices } i \text{ and } j \text{ are adjacent;} \\ 0, & \text{otherwise.} \end{cases} \qquad (1)$$

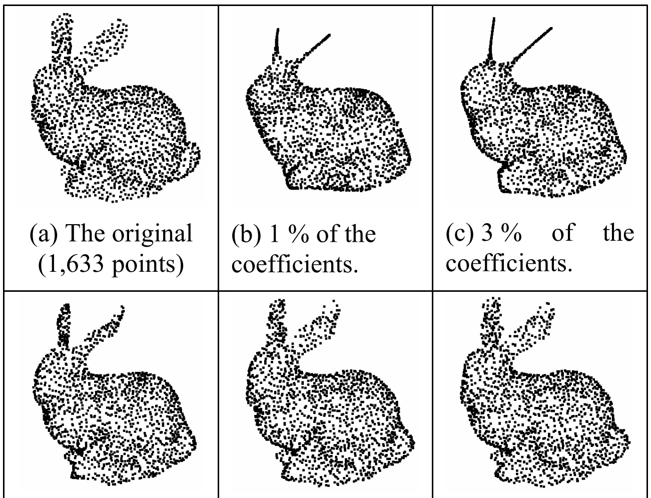
The eigenvalue decomposition of the matrix $\mathbf{K}$ produces *n* eigenvalues λ_i $(1 \le i \le n)$ and *n* eigenvectors $\mathbf{w}_i$ $(1 \le i \le n)$. By normalizing $\mathbf{w}_i$, we obtain the mesh spectral basis $\mathbf{e}_i = \mathbf{w}_i / \|\mathbf{w}_i\|$ $(1 \le i \le n)$. A set of mesh spectral coefficients $\mathbf{r}_i = (r_{s,i}, r_{t,i}, r_{u,i})$ $(1 \le i \le n)$ can be computed by projecting, *x*, *y*, and *z* components of the vertex coordinate $\mathbf{v}_i = (x_i, y_i, z_i)$ $(1 \le i \le n)$ onto the $\mathbf{e}_i$. The *s*, *t*, and *u* are the spectral axes corresponding to the *x*, *y*, and *z* of the spatial axes. We used a fast eigenvalue decomposition algorithm called *Arnoldi's* method implemented in the *Meschach* library [20] for the eigenvalue decomposition.



(a) The meshing algorithm connects with the point *p* all the points within the radius *r* from the point *p*.

(b) Using our meshing algorithm, points sampled on multiple surfaces may be "sewn-through".

Figure 3. The meshing algorithm based on 3D Euclidian proximity.



(a) The original (1,633 points)

(b) 1 % of the coefficients.

(c) 3 % of the coefficients.

(e) 5 % of the coefficients.

(e) 10 % of the coefficients.

(f) 15 % of the coefficients.

Figure 4. The shape of the "bunny0" model (a) consisting of 1,633 points is approximated by using 1% (b), 3 % (c), 5 % (d), 10 % (e), and 15 % (f) of its spectral coefficients.

Modulation: Our algorithm embeds watermark by modulating amplitude of the mesh spectral coefficients for the patch of mesh M_i Each spectral axis *s*, *t*, and *u*, has a mesh spectral coefficient vector of size *n*. To modify the coefficients, our watermarking algorithm employs the method used by Hartung et al in [9]. Assume that the data to be embedded into a mesh is an *m*-dimensional bit vector $\mathbf{a} = (a_1, a_2, ..., a_m)$, in which each bit takes the values $\{0,1\}$. Each bit a_j is duplicated by the *chip rate c* to produce a watermark symbol vector $\mathbf{b} = (b_1, b_2, \ldots b_{mc})$, $b_i \in \{0,1\}$ of length $m \cdot c \le n$;

$$b_i = a_j, \quad j \cdot c \le i < (j+1) \cdot c \tag{2}$$

Such duplication of the symbol vector increases the noise immunity of the watermark by averaging the signal during the watermark demodulation. The duplicated bit vector $\mathbf{b}_i$ is converted to another vector $\mathbf{b}' = (b'_1, b'_2, \ldots b'_{mc})$ $b'_i \in \{-1,1\}$ by the following simple mapping;

$$b'_i = \begin{cases} -1, & \text{if } b_i = 0; \\ 1, & \text{if } b_i = 1. \end{cases} \tag{3}$$

Let us now consider modulating spectral coefficients of one of the spectral axis s . Let $r_{s,i}$ be the i-th spectral coefficient prior to watermarking corresponding to the spectral axis s , $p_i \in \{-1,1\}$ be the *pseudo-random number sequence* (PRNS) generated from a known watermark-key k_w , and α $(\alpha > 0)$ be the modulation amplitude. Watermarked i-th spectral coefficient $\hat{r}_{s,i}$ is computed by the following formula;

$$\hat{r}_{s,i} = r_{s,i} + b_i \cdot p_i \cdot \alpha \tag{4}$$

The modulation amplitude α is computed by $\alpha = \phi \cdot \beta$ where ϕ is the maximum length of the axis-aligned bounding box (AABB) of the model, and β is the modulation amplitude ratio. The extraction requires the same watermark-key, which is a seed for the PRNS used for the embedding. Depending on the application of the watermark, the key may be made public. Or, the key may be delivered securely, for example, by using a public-key cryptography. Performing the similar modulation to the coefficients of t and u spectral axes produces a set of watermarked set of spectral coefficients $\hat{\mathbf{r}}_i = (\hat{r}_{s,i}, \hat{r}_{t,i}, \hat{r}_{u,i})$.

Mesh-Spectral Synthesis: Multiplying the eigenvectors $\mathbf{e}_i$ with the watermarked spectral coefficients $\hat{\mathbf{r}}_i = (\hat{r}_{s,i}, \hat{r}_{t,i}, \hat{r}_{u,i})$ and summing over i produce vertex coordinates of the watermarked mesh M' .

Point Set Generation: The watermarked point set P' is produced by removing the vertex connectivity from the watermarked mesh M' .

2.2 Watermark Extraction

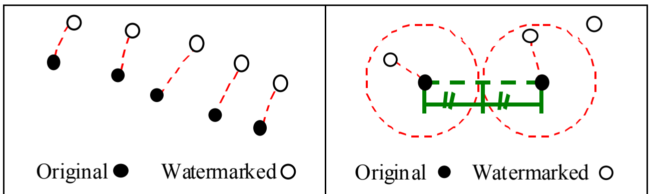
As an informed-detection watermark, the extraction requires the *reference-point-set-model*, i.e., an original model without the watermark, as well as the watermarked, and possibly degraded, point-set model.

Point Set Registration: Extraction of watermark requires accurate registration of the watermarked point set model with the original reference point set model (without watermark). Our algorithm tries to register a watermarked point set model transformed by using a similarity transformation. Without accurate registration, the resampling step and thus the watermark extraction fails.

Our registration algorithm is still a rudimentary one. It essentially employs the covariance matrix computed from the point set, which is similar to the method used by Gottschalk and others [6]. If the model is cropped, we create a cluster of points on the watermarked and reference point sets based on the manually specified seed points. Then, the same covariance analysis is applied to the clusters for their registration.

Point Set Resampling: We assume, as attacks, operations that alter the set of points of the model, e.g., simplification. To make the watermark robust against these attacks, we resample the geometry of the watermarked point set model P' . We take the reference point set P , and add geometry of the watermarked (and possibly simplified) point set P' via geometry resampling to produce a resampled point set P'' that has the same size as P .

To resample geometry of P' by P , we basically replace the coordinate of a point $p \in P$ with the coordinate of a point $p' \in P'$ closest, in 3D Euclidian distance, to p . We set a threshold to the neighbourhood size, so as not to replace the coordinate of p with that of p' which is far from p . The resampling steps are as follows (See Figure 5a).



(a) Replace the coordinate values of the reference (original) point set by their counterparts in the watermarked point set.

(b) Find corresponding points in the watermarked set p'_i of the reference points p_j .

Figure 5. Our resampling algorithm for the point sets.

(1) Using points in the reference point set P as vertices, generate a mesh M_c for resampling. Vertex coordinates of M_c are the same as the corresponding points in P . M_c is used to efficiently traverse points in P .

(2) Find a point $p'_i \in P'$ such that the 3D Euclidian distance from a point $p_j \in P$ is the smallest among all the points in $p' \in P'$.

(3) If the distance between p_j and p'_i is below the threshold, replace the coordinate values of the p_j with those of the p'_i .

(4) Perform the steps above for all the point p_j in P .

The distance threshold d_r is set to the 1/2 of the minimum distance from the point p_j to all the point in the reference point set P. We tested several values of d_r for their attack resistance, and the value 1/2 performed the best.

Point Set Clustering: The same clustering is applied to the resampled point set P'' and to the reference point set P. Clustering recreates exactly the same clusters as those used for the embedding on P by using the saved information on the seed points and sizes of the clusters.

Mesh Generation: Meshes are generated of the resampled point set P'' and to the reference point set P. The same meshing algorithm as for the embedding is first applied to the reference point set P, producing the reference mesh M. Then, the mesh connectivity is directly transferred to the P'' to create the watermarked mesh M''. As the point-to-point correspondences have been established during the resampling, the transfer is straightforward.

Mesh-Spectral Analysis: Mesh-spectral analysis is applied to each of the clusters to produce the spectral coefficients $\mathbf{r}_i = \left(r_{x,i}, r_{y,i}, r_{z,i}\right)$ of M and the spectral coefficients $\hat{\mathbf{r}}_i = \left(\hat{r}_{x,i}, \hat{r}_{y,i}, \hat{r}_{z,i}\right)$ of M''. As the meshes M and M'' have the same number of vertices and the connectivity, actual eigenvalue decomposition need to be perfomed only on the clusters of mesh M, whose results can be applied to the watermarked M''.

Demodulation: The signal q_j is computed by the following formula;

$$\begin{aligned} q_j &= \frac{1}{3} \sum_{l \in \{x,y,z\}} \sum_{i=j \cdot c}^{(j+1) \cdot c-1} (\hat{r}_{l,i} - r_{l,i}) \cdot h_i \\ &= \frac{1}{3} \sum_{l \in \{x,y,z\}} \sum_{i=j \cdot c}^{(j+1) \cdot c-1} b_i' \cdot \alpha \cdot h_i^2 . \end{aligned} \tag{5}$$

If the PRNS h_i is synchronized among the embedding and the extraction and the attacks can be ignored, $q_j = c \cdot \alpha \cdot b_i'$ holds. Here, the q_j ranges $\{-\alpha c, \alpha c\}$, and the α is always positive. Thus, the embedded watermark bits $\mathbf{w} = \left(w_1, w_2, ..., w_m\right)$ can be computed by checking for the sign of q_j, as in $w_j = sign(q_j)$.

3. Experiments and Results

We implemented the algorithm described in Section 2 using C++ on Linux operating system. We evaluated (1) the shape change due to watermarking, and (2) the attack resistance of the watermark.

As test data, we used densely sampled 3D mesh models that are created by 3D range scanning. By removing the connectivity, we created point set models. As we do not have had a tool for point set data manipulation, such as the point set 3D model simplification tool, we simulated the simplification of point set model via mesh simplification algorithms. That is, to produce point set models having smaller number of sample points, we (1) applied a mesh resampling algorithm [12] or mesh simplification algorithm to the given densely sampled mesh, and then (2) threw out the vertex connectivity. In the following, we call such a simplification of point set models via mesh simplification “simulated simplification”.

3.1 Attack Resiliency

We tested the resiliency of the watermark against attacks listed below. For the experiment, a watermark of 32 bits is embedded by using the amplitude ratio $\beta = 0.002$ the duplication factor $c = 10$, and the number of neighbourhood points for the mesh generation $N_n = 12$. The cluster used is the one illustrated in Figure 3(b). No visually observable distortion occurred due to the watermarking.

- **Similarity transformation:** The model was rotated about x axis by 10 degree, then about y axis by 30 degree, and down-scaled by the factor 0.6 (Figure 6(a)). No extraction error occurred as a result of this attack. On some models that have symmetry, such as a sphere or a cylinder, watermark is lost after the transformation, since registration can’t be performed on such symmetrical models.
- **Additive noise:** We added random noise of amplitude 0.5% of the maximum of the axis-aligned bounding box (AABB). Despite the noticeable distortion of the shape (Figure 6(b)), the watermark could be extracted without loss. The watermark is lost (at least partially) when we increased the noise amplitude to 0.8% of the AABB.
- **Point removal:** We reduced the number of points from 34,839 down to 4,839 without loss of the watermark (Figure 6(c)). The same point removal combined with similarity transformation destroyed the watermark. The loss was due to registration failure.
- **Cropping:** Watermark could be extracted without loss after certain (not arbitrary) cropping (Figure 6(d)). This is because the same watermark is embedded five times (i.e., the number of the clusters) into the model, and that at least one of the clusters survived the cropping.
- **Simulated resampling:** The watermark survived the simulated resampling (Figure 6(e)). The simulated resampling reduced the number of points from 34,839 down to 10,000 using the *MeshToss* [12]. When the number of points is further reduced down to 5,000, the watermark was destroyed.

Figure 8 shows the result of experiment that evaluated the relationship of additive noise amplitude, watermark amplitude factor β , symbol chip rate *c*, and the extraction bit error rate. The watermark noise amplitude is given relative to the largest axis of the AABB, as in the case of the watermark embedding amplitude factor β . In this experiment, we used a model horse2 having 8,485 points shown in Figure 7, and embedded 32 bit watermark without clustering the model. The graph plotted the numbers that are the averages of 20 experimental runs using the same parameter setting.

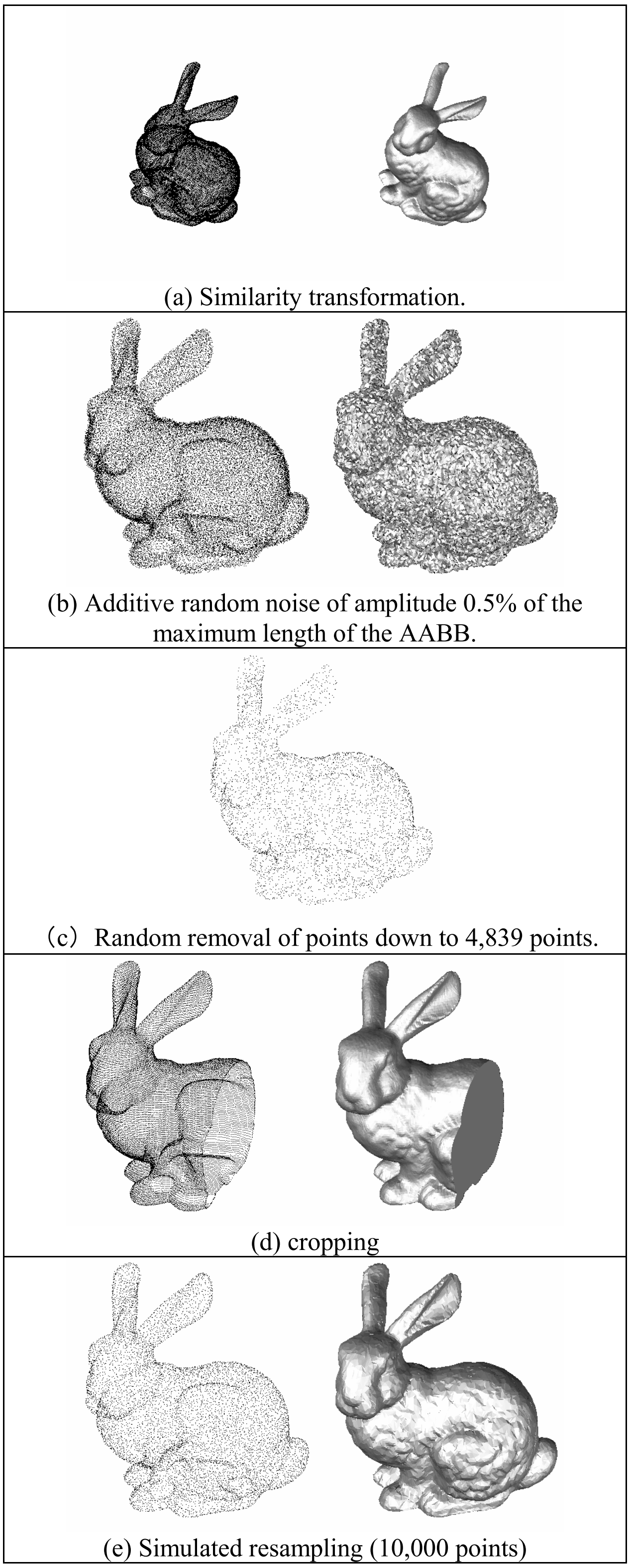
As shown in Figure 8, bit error rate increased gradually as the noise amplitude factor increases. As expected, error rate decreased (or, the noise resiliency increased) as the parameters *c* and/or β is increased. Note that these figures vary depending on the model used for the experiment.

As the experiments show, the watermark method described in this paper has certain resiliency against similarity transformation, additive random noise, cropping, point removal, and simulated resampling. The watermark can be destroyed if an attack that combines similarity transformation with either point removal or resampling. This is due to the failure of the registration step, followed by the failure in resampling.

4. Summary and conclusion

In this paper, we proposed and evaluated a watermarking algorithm for a 3D shape defined as an unoriented point set. The algorithm proposed in this paper essentially applied the frequency domain mesh watermarking algorithm of Ohbuchi, et al [17] after the point set model is converted into a mesh model. Our meshing algorithm is a simple, 3D Euclidian proximity based method that on occasion produces non-manifold meshes. Such a simple meshing algorithm may not be acceptable for viewing and other purposes, but is sufficient for analyzing 3D shape for watermarking. Experiments showed that the watermark produced by using our proposed method is resistant, to a certain extent, against similarity transformation, additive random noise, cropping, simulated resampling, and point removal. The watermark is destroyed fairly easily, however, if either simulated resampling or point removal is followed by similarity transformation. After such an attack, the registration step, and then the resampling step, failed.

Improvement of point-set model registration technique tops the list of our future work. A robust and accurate registration is crucial in achieving a watermark that withstands attacks which combine geometrical transformations with resampling or simplification. A possible approach is to perform better estimation of surfaces from the point set data, for example, using the algorithm described by Adamson, et al. [1].



(a) Similarity transformation.

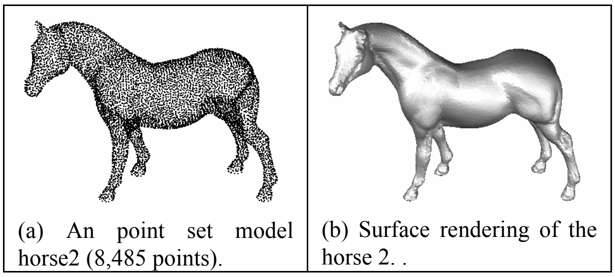
(b) Additive random noise of amplitude 0.5% of the maximum length of the AABB.

(c) Random removal of points down to 4,839 points.

(d) cropping

(e) Simulated resampling (10,000 points)

Figure 6. Attacks.



(a) An point set model horse2 (8,485 points).

(b) Surface rendering of the horse 2. .

Figure 7. The point-set model horse2 (8,485 points) used for the experiment of Figure 8.

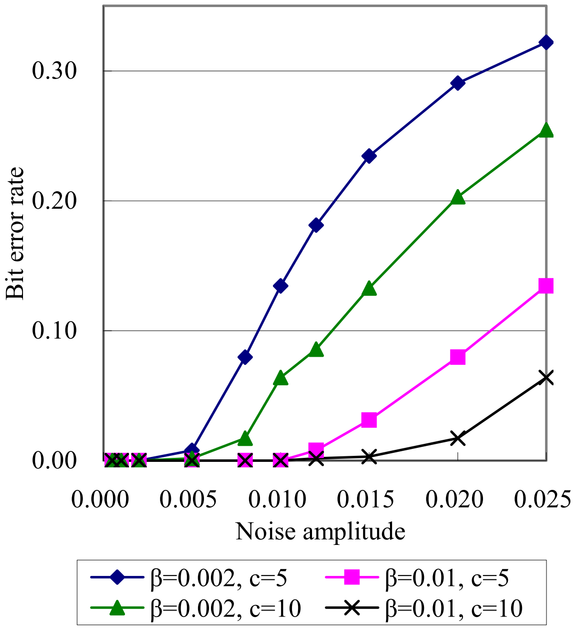


Figure 8. Bit error rate due to additive random noise as a function of the watermark amplitude, chip rate, and noise amplitude.

Acknowledgements

This research has been funded, in part, by the grants from the *Ministry of Education, Culture, Sports, Sciences, and Technology of Japan* (No. 12680432), *Okawa Foundation for Information and Telecommunications*, and the *Artificial Intelligence Research Promotion Foundation*.

References

[1] Anders Adamson, Marc Alexa, Approximating and Intersecting Surfaces from Points, Proc. *EG Symposium on Geometry Processing* 2003, 245-254 (2003)

[2] O. Benedens, Geometry-Based Watermarking of 3D Models, *IEEE CG&A*, 46-55, Jan./Feb., 1999.

[3] O. Benedens, C. Busch, Towards Blind Detection of Robust Watermarks in Polygonal Models, *Computer Graphics Forum*, Volume **19**(3), 2000 (Proc. EUROGRAPHICS 2000)

[4] F. Cayre, P. Rondao-Alface, F. Schmitt, B. Macq, H. Maitre, Application of spectral decomposition to compression and watermarking of 3D triangle mesh geometry, *Signal Processing : Image Communications*, **18**(4), 309--319, 2003.

[5] I. J. Cox, M. L. Miller, J. A. Bloom, *Digital Watermarking*, Morgan Kaufman Publishers, 2002.

[6] Gottschalk, S., Lin, M.C., Manocha, D., OBBTree: A Hierarchical Structure for Rapid Interference Detection, Proc. SIGGRAPH '96, 171-180, 1996.

[7] Golub, G. H., Van Loan, C. F., *Matrix Computations*, Third Edition, Johns Hopkins University Press, 1996.

[8] I. Guskov, W. Sweldens, P. Shröder, Multiresolution signal processing for meshes, Proc. *SIGGRAPH '99*, 49-56, 1999.

[9] F. Hartung, P. Eisert, and B. Girod, Digital Watermarking of MPEG-4 Facial Animation Parameters, *Computer and Graphics*, **22**(4), 425-435, 1998.

[10] H. Hoppe, Progressive Meshes, Proc. ACM SIGGRAPH '96, 99-108 (1996).

[11] S. Kanai, H. Date, and T. Kishinami, Digital Watermarking for 3D Polygons using Multiresolution Wavelet Decomposition, Proc. *Sixth IFIP WG 5.2 GEO-6*, 296-307, Tokyo, Japan, December 1998.

[12] T. Kanai, *MeshToSS* Version 1.0.1, http://grahics.sfc.keio.ac.jp/MeshToSS/indexE.html.

[13] Zachi Karni, Craig Gotsman, Spectral Compression of Mesh Geometry, Proc. SIGGRAPH 2000, 279-286, 2000.

[14] R. Ohbuchi, H. Masuda, and M. Aono, Watermark-ing Three-Dimensional Polygonal Models, Proc. *ACM Multimedia '97*, 261-272, 1997.

[15] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications, 551-560, *IEEE JSAC*, 1998.

[16] R. Ohbuchi, H. Masuda, and M. Aono, Geometrical and Non-geometrical Targets for Data Embedding in Three-Dimensional Polygonal Models, *Computer Communications,* **21**, 1344-1354, 1998.

[17] Ryutarou Ohbuchi, Akio Mukaiyama, Shigeo Takahashi, A Frequency-Domain Approach to Watermarking 3D Shapes, *Computer Graphics Forum,* **21**(3), 373-382, (2002).

[18] Emil Praun, Hugues Hoppe, Adam Finkelstein, Robust Mesh Watermarking, Proc. *SIGGRAPH '99*, 49-56, 1999.

[19] O. Sorkine, D. Cohen-Or, S. Toledo, High-Pass Quantization for Mesh Encoding, *Eurographics Symposium on Geometry Processing*, 42-51, (2003).

[20] David E. Stewart, Zbigniew Leyk, Meschach Library Ver. 1.2b, http://www.netlib.org/c/meschach

[21] M. G. Wagner, Robust Watermarking of Polygonal Meshes, Proc. *Geometric Modeling & Processing 2000*, 201-208, Hong Kong, April 10-12, 2000.

[22] B-L. Yeo and M. M. Yeung, Watermarking 3D Objects for Verification, *IEEE CG&A*, 36-45, Jan./Feb. 1999.

[23] Kangkang Yin, Zhigeng Pan, Jiaoying Shi, David Zhang, Robust mesh watermarking based on multiresolution processing, *Computers & Graphics*, **25**, 409-420 (2001)