

A Novel Semi-blind-and-Semi-reversible Robust Watermarking Scheme for 3D Polygonal Models

Chao-Hung Lin · Min-Wen Chao · Chan-Yu Liang · Tong-Yee Lee

Received: date / Accepted: date

Abstract We introduce a novel semi-blind-and-semi-reversible robust watermarking scheme for three-dimensional (3D) polygonal models. The proposed approach embeds watermarks in the significant features of 3D models in a spread-spectrum manner. This novel scheme is robust against a wide variety of attacks including rotation, translation, scaling, noise addition, smoothing, mesh simplifications, vertex reordering, cropping, and even pose deformation of meshes. To the best of our knowledge, the existing approaches including blind, semi-blind, and non-blind detection schemes cannot withstand the attack of pose editing, which is a very common routine in 3D animation. In addition, the watermarked models can be semi-reversed (i.e., the peak signal-to-noise ratio (PSNR) of the recovered models is greater than 90dB in all experiments) in semi-blind detection scheme. Experimental results show that this novel approach has many significant advantages in terms of robustness and invisibility over other state-of-the-art approaches.

Keywords Watermarking · copyright protection

1 Introduction

In the last decade, digital watermarking has become a very active research area and has drawn a lot of atten-

tion in the fields of ownership protection and authentication [1]. Most efforts on watermarking have been concentrated on various media data types such as document, image, audio, and video. With the fast development of 3D hardware, 3D computing and visualization has become increasingly efficient. Furthermore, the universal popularity of 3D games has led to the widespread use of 3D models in various applications such as digital archiving, entertainment, Web3D, game industry, and mechanical engineering. Therefore, the watermarking of 3D models has gained increasing attention in recent years. Although watermarking algorithms dedicated to regularly sampled signals such as audio, image, and video are reaching maturity, it is still very challenging to extend these known algorithms to embed watermarks on 3D models that are usually not regularly sampled. The common purpose of robust watermarking is to hide a watermark in digital contents in an imperceptible way so that they can withstand various malicious attacks. Therefore, *robustness* and *invisibility* are the main requirements of a robust watermarking algorithm. In this paper, we aim to utilize the geometric characteristics of 3D models to provide a robust watermarking algorithm for ownership protection.

A watermarking technique that requires the original multimedia data to detect the watermark is called non-blind watermarking. On the other hand, a blind scheme does not require the original multimedia data to detect the watermark. Generally, in the literature, non-blind schemes are more robust in detecting watermarks or can withstand more malicious attacks than blind schemes. However, non-blind approaches require the original data to extract watermarks; therefore, the multimedia industry appears to prefer blind schemes due to their practicality. In this paper, we contribute a novel semi-blind robust watermarking scheme for 3D

C.-H. Lin

The Department of Geomatics, National Cheng-Kung University,
No. 1, Ta-Hsueh Road, Tainan 701, Taiwan, R.O.C.
E-mail: linhung@mail.ncku.edu.tw

M.-W. Chao, C.-Y. Liang and T.-Y. Lee

The Computer Graphics Group/Visual System Laboratory,
Department of Computer Science and Information Engineering,
National Cheng-Kung University, No. 1, Ta-Hsueh Road,
Tainan 701, Taiwan, R.O.C. E-mail: cvivians@hotmail.com,
tonylee@mail.ncku.edu.tw

polygonal models. Rather than require the original models, we only need a small amount of information to detect watermarks. Remarkably, in contrast to other blind, non-blind, and semi-blind schemes, our approach is robust against a wider variety of attacks including rotation, translation, scaling, noise addition, smoothing, mesh simplifications (a special case of mesh re-sampling), vertex reordering, cropping, and even pose deformation of meshes.

Pose editing (or called pose deformation) is a useful and common operation in 3D computer animation. Users (or enemies) may attack watermark-embedded models through editing their poses. A watermarking algorithm must be robust against this type of attack. However, to the best of our knowledge, the existing approaches [2–16] including blind, semi-blind, and non-blind detection schemes cannot withstand this type of attack because embedding positions are lost after the vertex coordinates are significantly modified. Our approach embeds a watermark in the significant features of the models, which are detected by the proposed similarity-invariant curvature estimation approach. As long as the significant features have not been severely damaged by malicious attacks, our approach has a chance to successfully extract the watermark. Our approach embeds the watermark by deforming the significant features with shape constraints and successfully leads to imperceptible watermark embedding. Moreover, our approach can semi-recover the original models by deforming the watermarked models back using only a little information relative to the original models ($m + 1$ floats, where m represents the number of bits in a watermark), i.e., the semi-blind scheme. Experimental results show that our watermarking scheme can withstand more types of attacks than all previous approaches [2–16].

The remainder of the paper is organized as follows. We review related works in Section 2. After briefly summarizing the watermarking scheme in Section 3, we describe it in detail in Section 4. Section 5 demonstrates and discusses the experimental results. Section 6 concludes the proposed approach.

2 Related Work

Watermarking approaches can be categorized into *robust watermarking* [2–15] and *fragile watermarking* [17–19] based on what objective the approaches want to achieve. For fragile watermarking, the main purpose is to detect slight changes for authenticating the integrity of digital content. In contrast, robust watermarking is designed to resist various attacks for copyright protection. In this paper, we concentrate on the robust watermarking issue for 3D models. In this section, we will re-

view the related watermarking work for 3D models represented by the polygon format which is the most-used digital representation of 3D models. As for other representations, the readers can refer to [20] for 3D Non-uniform Rational Basis Spline (NURBS) data, to [21] for 3D models with texture data, and to [22] for point data.

In [9], Praun et al. present a robust watermarking approach that extends the concept of spread spectrum [24] to 3D models. They identify the significant geometric differences between the simplified and the original models by using a multi-resolution analysis approach. Then each vertex in the identified areas is perturbed along the direction of its vertex normal. This algorithm is robust against similarity transformation, mesh smoothing, noise addition, and simplification attacks using *non-blind* detection, i.e., requires the original models to detect watermarks. Similar to [9], Date et al. [11], Yin et al. [10], Ohbuchi et al. [7,8], Ashourian et al. [2], and Benedens et al. [3,4] propose non-blind watermarking approaches. A watermark is embedded in the frequency domain, coarse mesh, mesh spectral domain, spherical domain, or vertex normals. However, all above algorithms proceed the time-consuming processes of model alignment and initial connectivity recovery with the original models to extract embedded watermarks. In this paper, these two processes are avoided using a *semi-blind* detection approach. We only require a small amount of information instead of the entire models.

In contrast to non-blind detection, blind detection schemes [5,6,12–14] only need to use a private key to detect watermarks. In general, blind detection is achieved by aligning the models with the principal axes generated by principal component analysis (PCA). However, these approaches cannot generally resist cropping and pose deformation attacks because such attacks can cause significant alteration to both the principal object axes and the mass center. Recently, Lee et al. [15] present an interesting semi-blind detection approach that requires storing the sampling density and some parameters to extract watermarks. They iteratively project the models onto two constrained convex sets and then embed watermarks by modifying the sample means of components in the convex sets. Compared with blind detection approaches [5,6,12–14], this approach can resist cropping attacks. However, this approach is time consuming (about 30 minutes) and cannot resist pose deformation attacks. In contrast, our approach can resist cropping and pose deformation attacks in addition to other general attacks. Furthermore, compared to [15], our watermark extraction is more efficient (about 30 seconds for a model with 50,000 vertices). In the

past, many efforts on watermarking have been concentrated on images. Based on image watermarking, an alternative approach is to embed watermarks on geometry images of 3D models [23]. However, the distortion problem of geometry images makes them difficult in handling 3D models with complex shapes.

3 System Overview

The proposed watermarking scheme consists of two separate procedures, the *embedding procedure* and the *detection procedure*. Both have two major steps: *significant patch determination* and *patch sorting*. The overview of the embedding procedure is described as follows. To consider perceptual invisibility, the watermark is embedded in significant patches, i.e., high-curvature areas, of the 3D models. We propose a curvature estimation approach to determine these embedding patches (Figure 1(a), Section 4.1). To determine the embedding order, we sort the embedding patches by geodesic distances that are insensitive to various aforementioned attacks (Figure 1(b), Section 4.2). Finally, each bit of watermark is embedded in one embedding patch by a feature-preserving deformation approach (Section 4.3). This step is very efficient because all patches are simultaneously embedded by solving a least-square minimization equation. Instead of the entire model, we only store the mean curvatures of m patches (m floats), embedding amplitude (1 float), and the watermark for watermark detection. In the detection procedure, the embedding positions and embedding order of the suspected models can be obtained in the same steps. The watermark is then extracted by comparing the embedding patch curvatures in the suspected model with the stored curvature information.

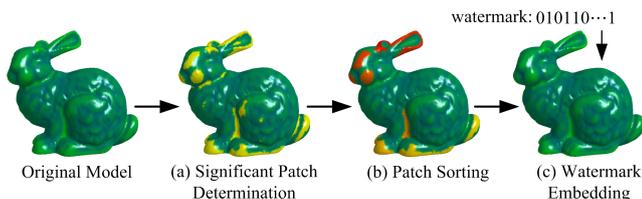


Fig. 1 The workflow of the proposed watermark embedding. (a) The original model; (b) the significant patches (visualized by yellow); (c) the sorting result of significant patches (the order is visualized by color starting from red to yellow); (d) the watermarking result.

4 Watermark Embedding

4.1 Significant Patch Determination

In the proposed approach, the watermark is embedded in the 3D models in a spread-spectrum manner. The

spread-spectrum technique is to transform the digital media to the frequency domain and perturb the coefficients of the most significant basis functions for embedding the watermark [24]. However, the polygonal models lack a natural approach for frequency-based decomposition. To apply the spread spectrum technique to the polygonal models, the significant patches are detected first and the vertices in each patch are then perturbed. In [9], the significant patches are determined by a multi-resolution analysis approach [25]. In their work, the original model is represented as a progressive mesh format consisting of a coarse base mesh and a sequence of refinement operations.

Each vertex in the base mesh corresponds to a significant patch in the original model. Therefore, the significant patches are determined by directly selecting the vertices in the base mesh with the largest geometric magnitudes between the base mesh and the original model. For a non-blind detection scheme [9], this patch determination approach is robust against various attacks since the original models are stored. However, for a blind or semi-blind detection scheme, this approach is sensitive to attacks of noise addition, mesh smoothing, and pose deformation since these attacks can potentially and significantly alter the selection order of collapsed edges, i.e., model simplification. The main challenge of embedding position determination in a blind or semi-blind detection scheme is that it must be insensitive to various malicious attacks. In this paper, a novel approach based on a similarity-invariant curvature estimation is proposed for determining embedding positions. This approach is described as follows.

The first step is to estimate the surface curvatures. Several excellent previous works generalize the curvature estimation in the differential geometry to the discrete polygon mesh [26–28]. In this paper, the approach presented by Alliez et al. [28] is extended to compute the mean curvatures of 3D models. Let $\kappa(v)$ represents the mean curvature of a vertex v and $NE(v)$ represents the set of edges in the neighborhood of a vertex v . A local curvature of an edge can simply be estimated as the angle between the two faces adjacent to this edge. Therefore, the curvature $\kappa(v)$ can be formulated as the integral of local curvature over the vertex neighborhood B (see Eq. 1 and Figure 2). The main drawback of this estimation is that the local curvature is sensitive to noise. To withstand the attack of noise addition, we de-noise the face normal by a smoothing filter F before computing the vertex curvature. In addition, to withstand the attack of scaling, we normalize the curvature. Therefore, the mean curvature of a vertex v is formulated as follows:

$$\kappa(v) = \frac{\sum_{e \in NE(v)} \|e \cap B\| \|F * n_{f_i} - F * n_{f_j}\|}{\sum_{e \in NE(v)} \|e \cap B\|},$$

$$F * n_{f_k} = \frac{\sum_{f_l \in NE(f_k)} (n_{f_l} \exp[-dist(f_l, v)^2 / \sigma^2])}{\sum_{f_l \in NE(f_k)} (\exp[-dist(f_l, v)^2 / \sigma^2])} \quad (1)$$

where n_{f_i} and n_{f_j} represent the normals of two adjacent faces of an edge e (see Figure 2). F represents a smooth Gaussian filter, and the symbol ‘*’ denotes convolution operation. $\|e \cap B\|$ is the length of edge e in the neighborhood B . In the smooth filter, f_k represents a face in the vertex neighborhood B , $NE(f_k)$ represents a face set of f_k ’s neighborhood, and σ represents the Gaussian’s standard deviation. The distance function $dist(f_l, v)$ returns the distance between the center of face f_l and vertex v .

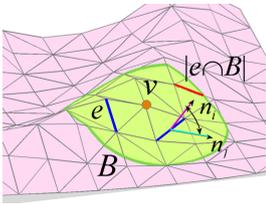


Fig. 2 An illustration of curvature calculation.

Once the surface mean curvatures are obtained, a region-growing strategy is adopted to determine the significant patches. Only the high-curvature vertices are selected as the growing areas (the top 30% high-curvature vertices in all experiments), and the vertices with a local maximal curvature are selected as the growing seeds. In the expansion step, for each seed, we simply find the maximal connected region, i.e., patch, in the growing areas. In other words, the high-curvature connected vertices are merged to become a patch for embedding watermark. Taking the robustness into account, the patches containing only a few vertices (less than 0.5% number of vertices in the model) are filtered out. The remaining patches are called significant patches and used for watermark embedding. Note that the number of extracted patches depends on the parameter setting of the smoothing kernel size, i.e., σ (set to 1.5% of the diagonal of the object bounding box in the experiments), and the threshold for the growing areas. It is well known in the field of data hiding that there is a tradeoff between embedding capacity and robustness. If more data need to be hidden in a model, then some patches in the small detailed features of the models would be selected for embedding, leading to weak robustness. It is because that the small detailed features are sensitive to the attacks of noise addition and mesh smoothing. In the application of ownership protection, robustness is more important than embedding

capacity. Therefore, we select a large size of smoothing kernel as well as large patches for watermark embedding.

4.2 Patch Sorting

To embed a bit string, i.e., watermark, to a model, we must determine the embedding order. The approach for determining embedding order must also be robust against aforementioned attacks. In the proposed approach, the embedding order is determined by the geodesic distances among the significant patches obtained in Section 4.1. First, within each patch, we select the vertex with maximal curvature as its representative vertex and compute the average curvature of the vertices in this patch as its representative value, called salient value. Among all significant patches, we call the patch with maximal salient value as the pivot patch. All patches are sorted according to the geodesic distance between the pivot patch and the other patches. We do not sort them using Euclidean distance since its distance can be significantly altered when the pose deformation is applied to 3D models. Figure 3 illustrates the order of patch sorting using geodesic distance; this order will be not changed by a pose deformation attack. However, obviously, if this sorting order is determined by Euclidean distance, it will be greatly altered (see red dashed-line paths in Figure 3). In our approach, there are two reasons to select the patch with maximal salient value as the pivot patch. First, the patch with the highest salient value implies that it is the most robust one in withstanding various malicious attacks. Second, it is generally the most significant patch in a model. If the most significant patch is severely damaged, for example, by cropping, the attacked model could become meaningless for people. In Figure 3, we show several examples of the significant patch determination and sorting.

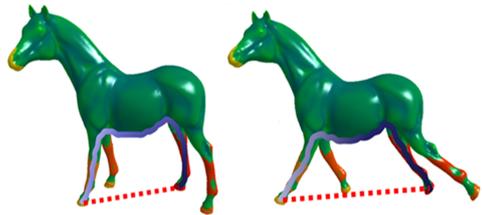


Fig. 3 An illustration of the patch sorting by geodesic distance. The thick blue lines show the paths between two patches found by the shortest geodesic distance. The red dashed lines show the shortest Euclidean distance between two patches and their values will be changed significantly after the attack.

4.3 Watermark Embedding

A 3D polygonal model can be described as a pair (K, V) , where K is a simplicial complex representing the con-

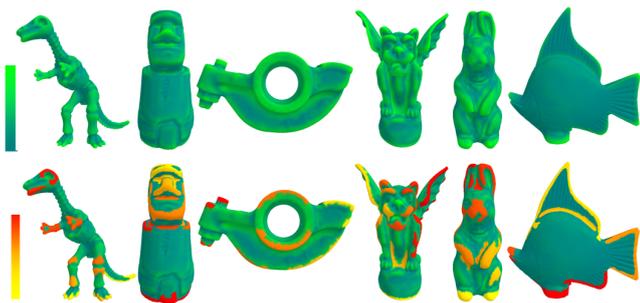


Fig. 4 Top: the mean curvatures encoded by colors ranging from dark green (low curvature) to light green (high curvature); Bottom: the extracted significant patches and their sorting results (the order is also represented by colors starting from red (the most important patch) to yellow (the least important patch)).

nectivity of vertices, edges, and faces; V is the vertex positions in R^3 . The differential coordinate δ_i of vertex v_i is defined as follows:

$$\delta_i = \sum_{(i,j) \in K} w_{ij}(v_i - v_j), \quad (2)$$

where w_{ij} is the weights for approximating the continuous Laplace operator. Here, we adopt the cotangent weights [27].

We embed a watermark $\{m_i\}_{i=1}^m$ containing m bits by deforming the significant patches along the differential coordinates. Specifically, it is achieved by multiplying the differential coordinates δ_i of vertices v_i in the significant patches with the defined scale factor h (the embedding amplitude, h is set to 0.1 in all experiments) and fixing the differential coordinates of the vertices in the other regions. That is,

$$\delta'_i = \begin{cases} \delta_i + hm_k \delta_i, & \text{if } v_i \text{ belongs to } P_k \\ \delta_i, & \text{otherwise} \end{cases} \quad (3)$$

where P_k is the k^{th} significant patch.

To correctly extract the pivot patch (a patch with the largest salient value), we do not minify the differential coordinates of vertices (minifying differential coordinates will lower the salient value). Therefore, the differential coordinates of vertices are either magnified or unchanged in the Eq. 3. The effect of simply magnifying only the differential coordinates is similar to directly enlarging the local shapes. It potentially results in a larger distortion of the embedded models and therefore less robust embeddings. To solve this problem, we add edge constraints to enforce the length and direction of the edges in the original models on the deformed models, i.e., watermarked meshes. Taking into account the aforementioned constraints and the vertices in other non-embedded regions, the watermark embedding is formulated as follows:

$$\arg \min_{V'} \left(\sum_{v_i \in V} \|\delta_i - \delta'_i\|^2 + \alpha \sum_{(i,j) \in K} \|e_{ij} - (v'_j - v'_i)\|^2 + \beta \sum_{v_i \in U} \|v_i - v'_i\|^2 \right) \quad (4)$$

where U represents the set of fixed vertices (the vertices not in the significant patches or the vertices in the significant patch P_k and their embedded bit is 0, i.e., $m_k = 0$; $e_{ij} = (v_j - v_i)$; α and β are the weighting factors for the edge constraints and vertex constraints, respectively. To fix the vertices whose embedding bit is 0, we give a larger weight for β and a smaller weight for α to enforce the vertex constraints α is set to 0.1 and β is set to 1.0 in all experiments).

In practice, Eq. 4 is solved by an over-determined linear system $Ax = b$ (see Eq. 5) in an iterative manner [29]. The minimization solving is very efficient since the system matrix A is sparse, and all entities are fixed (they contain the Laplacian matrix L_{ij} , edge-constraint matrix E_{ij} , and vertex-constraint matrix C_{ij}). Therefore, the factorization of matrix $A^T A$ can be pre-computed and therefore there is only a back-substitution required for each iteration.

$$\begin{bmatrix} L_{ij} \\ E_{ij} \\ C_{ij} \end{bmatrix} [V'] = \begin{bmatrix} \delta'_{ij} \\ k_{ij} \\ v_{ij} \end{bmatrix}, \quad (5)$$

where $L_{ij} = \begin{cases} 1, & \text{if } i = j \\ -w_{ij}, & \text{if } (i, j) \in K \\ 0, & \text{otherwise.} \end{cases}$, $C_{ij} = \begin{cases} \beta, & \text{if } v_i \in U \\ 0, & \text{otherwise.} \end{cases}$

and $E_{ij} = \begin{cases} E_{ii} = -\alpha, E_{ij} = \alpha, & \text{if } (i, j) \in K \\ 0, & \text{otherwise.} \end{cases}$

Under the same embedding amplitude, i.e., when the vertex offsets induced by the watermark embedding is identical, the embedding results generated by our approach are better in terms of invisibility than those generated by the approach [9]. It is because that the approach [9] is to directly enlarge the models in normal directions (as shown in Figure 5(a)). In contrast, we preserve the shape feature when the vertex differential coordinates are enlarged (as shown in Figure 5(b)).

4.4 Watermark Extraction

To extract a m -bit watermark, we need 1) the salient value S of each significant patch (i.e., m floats for m patches) and 2) the embedding amplitude h (1 float). Therefore, we only need to store these $(m + 1)$ float data instead of the entire original model. The extraction process is similar to the embedding process. The

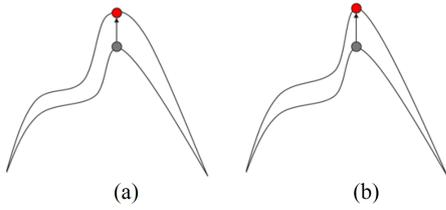


Fig. 5 An illustration of the comparison between (a) perturbing vertices along the direction of normal and (b) perturbing vertices by deforming the local shapes with constraints (Eq. 5).

significant patches are extracted from the suspected model first and then the extracted patches are sorted by geodesic distances. By considering the difference between the salient value S_i^* of the i^{th} patch (i.e., sorted by geodesic distances) in the suspect models and the corresponding S_i in the stored data, we can extract the watermark m_i^* . Specifically, our watermarking extraction can be formulated as follows:

$$m_i^* = \begin{cases} 1, & \text{if } 0.5h \leq \frac{|S_i^* - S_i|}{S_i} < 1.5h \\ 0, & \text{if } \frac{|S_i^* - S_i|}{S_i} < 0.5h \\ \text{false}, & \text{otherwise} \end{cases} \quad (6)$$

where h is the embedding amplitude, and "false" means extracting nothing.

Watermark analysis. The watermark analysis is simply achieved by comparing the inserted and extracted watermarks bit-by-bit. However, when the watermarked models are attacked by cropping, some significant patches could be cropped. This leads to an unsuccessful watermark matching. A bit-shifting approach is used here if the suspected models had been cropped. This approach simply shifts the mismatched bit to the right one in order to test if the following bits in the watermark are matched better, as shown in Figure 6. The third and sixth bits are mismatched, and the watermark is shifted right twice.

Embedding order	: 1 2 3 4 5 6 7 8 9 10
Inserted watermark	: 1 0 1 0 1 1 0 0 1 1
Extracted watermark	: 1 0 0 1 0 0 1 1 0 1
1 st bit shifting	: 1 0 x 0 1 0 0 1 1 0 1
2 nd bit shifting	: 1 0 x 0 1 x 0 0 1 1 0 1

Fig. 6 An example of bit shifting.

Since our watermark is embedded in high-curvature regions, the smoothing attack could significantly alter the curvature, and therefore the embedded watermark could potentially be destroyed. To strengthen the robustness of our approach in withstanding the smooth-

ing attack, we approximately align the salient values S_i and S_i^* by using the following:

$$S_i' = S_i^* + \text{offset}, i = 1, \dots, m$$

$$\text{offset} = \frac{1}{m} \sum_{i=1}^m (S_i - S_i^*) \quad (7)$$

Note that the processes of bit shifting and salient value alignment could increase the probability of Type II error [30], i.e., the error of not rejecting a false bit. To solve this problem, we perform this alignment only on the condition that $S_i > S_i^*$ for all i , and perform the bit shifting on the condition that the bit error rate (BER), i.e., (the number of false bits / the number of correct bits)*100%, is significantly reduced (20% in all experiments) after shifting the watermark starting from a false bit.

Model recovery. The proposed approach can semi-recover the original models after extracting the watermarks. It is achieved by deforming the watermarked model back. We divide the vertex differential coordinates δ^* in the significant patches P^* by $(1 + hm^*)$, and fix the differential coordinates of vertices in the other regions. That is,

$$\delta_i' = \begin{cases} \delta_i^* / (1 + hm_k^*), & \text{if } v_i^* \text{ belongs to } P_k^* \\ \delta_i^*, & \text{otherwise} \end{cases} \quad (8)$$

where P_k^* is the k^{th} significant patch.

Then a recovered model is obtained by solving Eq. 5. Table 1 shows the PSNR rates of the watermarked and recovered models. PSNR is calculated via the root mean squared error (RMSE) between the original model and the evaluated model (watermarked or recovered models). The RMSE is defined as $\sqrt{\frac{1}{|V|} \sum_i^{|V|} \|v_i - v_i'\|^2}$, and the PSNR is defined as $20 \log_{10}(D_{max} / \sqrt{MSE})$, where D_{max} represents the diagonal distance of the bounding box of the original model. In Table 1, the PSNR statistics are all above 70dB for the watermarked models (the range of PSNR is $[0, \infty]$ and the acceptable values for 3D modeling are considered to be about 60dB to 70dB). It implies that the alteration of the watermarked models is imperceptible with respect to the human visual system. After deforming the watermarked models back, the PSNR rates of the recovered models increase about 20dB. In other words, the recovered models are almost equivalent to the original models. Note that it is very difficult (or impossible) to completely recover the watermarked models because the 3D models are represented by finite precision floating points. Therefore, there are some truncation errors in any floating operation.

Table 1 A statistics of model distortions in the watermarked and recovered models.

						
Embedded models (dB)	73.25	76.60	74.39	87.24	87.69	81.46
Recovered models (dB)	93.72	93.50	97.74	108.11	110.67	104.18

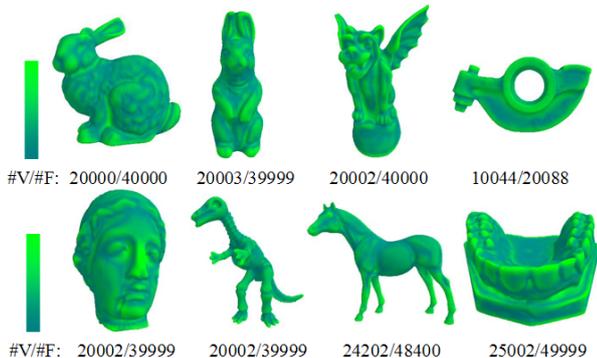


Fig. 7 Various test models. The #V and #F represent the number of vertices and faces, respectively. The mean curvatures are encoded by colors ranging from dark green (low curvature) to light green (high curvature).

5 Experimental Results

To validate the feasibility of the proposed approach, various 3D models are selected in the experiments, as shown in Figure 7. The proposed watermarking system is based on the significant patch determination and patch sorting. Therefore, we start this section with the experiments of testing if they are robust against various attacks including noise addition, smoothing, cropping, simplification (a special case of mesh re-sampling), and pose deformation (see Figures 8 – 10). The embedding order is visualized by colors starting from red to yellow. We can see that the significant patches and embedding orders can be accurately obtained even though the models are altered by these attacks. In contrast to the previous blind detection approaches [5, 6, 12–14] that will fail under the attacks of cropping and pose deformation (see Figure 11), the proposed approach can still resist these two types of attacks (see Figures 8 – 10 (d) and Figure 9 (c)) since the surface curvature is only slightly altered.

To demonstrate the robustness of our watermarking approach, a variety of malicious attacks including noise addition, smoothing, cropping, vertex reordering, simplification, and pose deformation are tested. The experimental statistics are shown in Tables 2 – 3. To fairly evaluate our method, the configuration of all parameters in the watermark embedding algorithm is identical



Fig. 8 (a) The original model; (b) noise addition, (c) smoothing and (d) cropping attacks.



Fig. 9 (a) The original Model; (b) noise addition, (c) pose deformation and (d) cropping attacks.

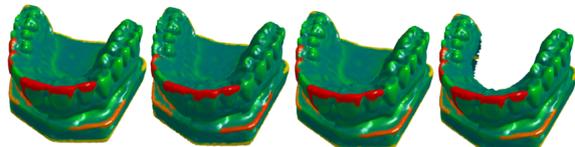


Fig. 10 (a) The original Model; (b) noise addition, (c) simplification and (d) cropping attacks.

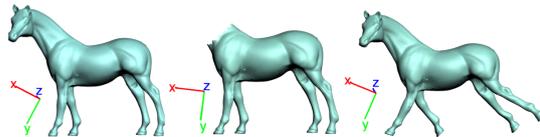


Fig. 11 Determining the embedding positions by PCA. Left: original model; Middle: cropping attack; Right: pose deformation attack. In addition, we show the PCA axes for each case.

in all experiments (the size of smoothing kernel σ is set to 1.5% of the diagonal of the bounding box; the growing regions: the top 30% vertices; embedding amplitude $h = 0.1$; the weights for edge and vertex constraints: $\alpha = 0.1$, $\beta = 1.0$; the size of watermark: 12 bits). BER is used to evaluate the robustness. Various magnitudes of attacks are also tested in these experiments. In the 1st group of Table 2 (noise addition attack), various noise magnitudes are tested. The '%' represents the noise magnitude as a fraction of the diagonal distance of the bounding box. The noise effects on the 3D models are shown in Figure 12. The statistics show that our approach is slightly sensitive to the noise addition when the noise magnitude is greater than 0.08%. This is because the magnitude of the noise attack is greater than the watermarking magnitude, i.e., the model deformation (determined by the embedding magnitude h , see Eq. 3). In the 2nd–4th groups of Table 2 (the cropping (see Figure 13), pose deformation (see Figure 14) and vertex reordering attacks), the statistics show that our approach can withstand these attacks. In

Table 2 A statistics of robustness estimation (using BER). 1st group: the noise addition attack (Gaussian noise). The '%' in this group represents the noise magnitude (variance) as a fraction of the diagonal distance of the bounding box. 2nd group: the cropping attacks (Crop.). Two arbitrary cropping attacks (Crop.1 and Crop.2) are tested, and the removed bits of watermark do not be included in the estimation of BER. 3rd group: the pose deformation attack (P.D.). Two arbitrary pose deformations (P.D.1 and P.D.2) are tested here. The notation '-' represents no experiments. 4th group: the vertex reordering attack (V.R.). 'N' represents a failure in the process of significant patch extraction.

Attacks								
Noise Addition								
0.02%	0%	0%	0%	0%	0%	0%	0%	0%
0.04%	0%	0%	0%	8%	16%	16%	8%	16%
0.06%	0%	0%	8%	16%	16%	16%	16%	16%
0.08%	0%	N	16%	16%	33%	16%	25%	N
0.10%	8%	N	16%	25%	33%	66%	33%	N
0.12%	25%	N	33%	25%	42%	N	N	N
Crop.1	0%	0%	0%	0%	0%	0%	0%	0%
Crop.2	0%	0%	0%	0%	0%	0%	0%	0%
P.D.1	0%	-	-	0%	0%	-	-	-
P.D.2	0%	-	-	0%	10%	-	-	-
V.R.	0%	0%	0%	0%	0%	0%	0%	0%

Table 3 Statistics of robustness estimation (using BER). 1st group: the smoothing attack. The '%' in this group represents the smoothing strength as a fraction of the differential coordinates. 2nd group: the simplification attacks (Crop.). The '%' in this group represents that the percentage of the number of vertices in the original models is simplified.

Attacks							
Smoothing							
5%	0%	0%	0%	0%	0%	0%	0%
10%	0%	0%	0%	16%	0%	0%	0%
15%	0%	0%	0%	16%	0%	0%	0%
20%	8%	0%	0%	16%	0%	0%	0%
25%	8%	0%	0%	16%	0%	0%	0%
30%	16%	0%	0%	25%	0%	0%	0%
35%	16%	0%	0%	25%	0%	0%	0%
40%	16%	16%	0%	N	0%	N	N
Simplification							
5%	0%	0%	0%	0%	0%	0%	0%
10%	0%	0%	16%	0%	8%	25%	0%
15%	0%	0%	16%	16%	8%	33%	0%
20%	16%	0%	16%	16%	8%	N	0%
25%	42%	16%	N	25%	8%	N	0%

the 1st group of Table 3 (the smoothing attack), the smoothing filter [32] with various smoothing strengths is applied to the vertex coordinates (only one iteration). The '%' represents the smoothing strength as a fraction of the differential coordinates. The statistics show that our approach is robust until the smoothing strength is greater than 40%. In the 2nd group of Table 3 (the sim-

plification attack), the QSlim algorithm [34] is adopted. The statistics show that our approach weakly withstand this attack when the size of the simplified data is greater than 20%. Note that there is a tradeoff between robustness and invisibility in a watermarking system. If a larger embedding amplitude is used to embed the watermark, a more robust but less invisible watermarking is obtained. Therefore, our approach will be more robust against these malicious attacks if a larger deformation, i.e., h , is applied to the 3D models.



Fig. 12 The models suffering from the noise attack.

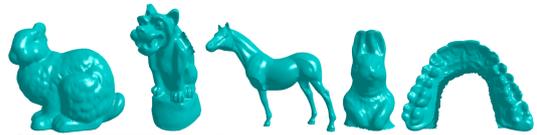


Fig. 13 The models suffered from the cropping attack.

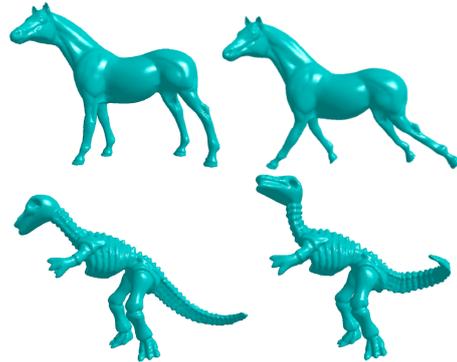


Fig. 14 Left: The original models; Right: the models suffered from the pose deformation attack.

Table 4 shows a theoretical comparison between the proposed approach and the related watermarking approaches [3, 6, 8, 9, 12, 13, 15] including the blind, semi-blind, and non-blind detection schemes. In the non-blind detection schemes [8, 9], the approaches cannot withstand the pose deformation attack since the watermark detection will fail in the processes of model alignment and original connectivity restoration. In addition, these two approaches require the original model to extract watermarks. In contrast, our approach can resist the attack of pose deformation while only requiring a small amount of information to extract watermarks. In the blind detection schemes [6, 12, 13], the approaches cannot resist the cropping and pose deformation attacks since these attacks will cause severe alteration to both the principal object axis and the mass center. In our

Table 4 The comparisons between our approach and the related approaches. Here, the symbols ' χ ,' ' Δ ,' and ' \surd ' indicate that the approach cannot withstand, can withstand, or can absolutely withstand attacks (i.e., BER=0%), respectively. (1), (2), and (3) indicate the blind, semi-blind, and non-blind detection schemes, respectively.

Detection Scheme	Our approach (2)	[15] (2)	[13] (1)	[12] (1)	[6] (1)	[3] (1)	[9] (3)	[8] (3)
(Attacks)								
Simplification	Δ	Δ	Δ	χ	χ	Δ	Δ	Δ
Cropping	Δ	Δ	χ	χ	χ	χ	Δ	Δ
Noise	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
Smoothing	Δ	Δ	Δ	Δ	χ	Δ	Δ	Δ
Similarity transform	\surd	\surd	\surd	\surd	\surd	\surd	\surd	\surd
Vertex reordering	\surd	\surd	\surd	\surd	\surd	\surd	\surd	\surd
Pose Deformation	Δ	χ	χ	χ	χ	χ	χ	χ

approach, the embedding positions are determined by the proposed similarity-invariant curvature estimation approach instead of the PCA approach. The problems mentioned above can be avoided, and thus the cropping and pose deformation attacks can be resisted. In the semi-blind detection scheme [15], the approach cannot also resist the pose deformation attack, while the embedding or extraction process is time consuming (about 30 minutes). In contrast, our approach can resist this attack, and the algorithm is efficient (about 30 seconds for a model with 50,000 vertices).

6 Conclusions, Limitations and Future Work

We propose a novel semi-blind, semi-revisable robust watermarking scheme for 3D polygon models. The experimental results show that our approach is robust against a wide variety of attacks, including similarity transformation, noise addition, smoothing, cropping, vertex reordering, simplification, and even pose deformation. In addition, our approach has the ability to semi-recover the original models. Currently, the proposed approach has the following two limitations. Our approach is not suitable to handle the models that have smooth shapes as well as few protrusive patches, such as sphere and knot, since we select protrusive patches, i.e., high curvature patches, to embed watermarks. The other limitation is that our approach cannot resist the attacks of non-uniform scaling, shearing, and even free-form deformation since these attacks could cause severe alteration to the surface curvature. In addition, they are usually considered as intentional degradations of the mesh shape. While in this paper we focus on 3D mesh, some directions for future work are to provide watermarking approaches for 3D models with skeletons [33], deformable meshes [31], and deformable volume data [35], since these data have gained increasing attention in some popular applications.

Acknowledgements We thank the anonymous reviewers for their insightful comments that help us improve the paper. This work was supported by the National Science Council of Taiwan, R.O.C. under contracts No. NSC 98-2221-E-006-179, NSC-97-2628-E-006-125-MY3 and NSC- 96-2628-E-006-200-MY3, and also supported by the Landmark Project of National Cheng Kung University, Taiwan under contract No. B0008 and C0038.

References

- Podilchuk CI, Delp EJ, Digital watermarking: Algorithms and applications, *IEEE Signal Processing Magazine*, 18(4):33–46 (2001)
- Ashourian M, Enteshari R, Jeon J, Digital watermarking of three-dimensional polygonal models in the spherical coordinate system, In: *CGI '04: Proceedings of the Computer Graphics International*, IEEE Computer Society, Washington, DC, USA, pp 590–593 (2004)
- Benedens O and Busch C, Toward Blind Detection of Robust Watermarking in Polygonal Models, In: *Computer Graphics Forum*, 19:199–208 (2000)
- Benedens O, Geometry-based watermarking of 3d models, *IEEE Comput Graph Appl* 19(1):46–55 (1999)
- Kalivas A, Tefas A, Pitas I, Watermarking of 3d models using principal component analysis, In: *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo*, IEEE Computer Society, Washington, DC, USA, pp 637–640 (2003)
- Ni Yq, Liu B, Zhang Hb, A blind watermarking of 3d triangular meshes using geometry image, In: *CGIV '07: Proceedings of the Computer Graphics, Imaging and Visualisation*, IEEE Computer Society, Washington, DC, USA, pp 335–340 (2007)
- Ohbuchi R, Takahashi S, Miyazawa T, Mukaiyama A, Watermarking 3d polygonal meshes in the mesh spectral domain, In: *GRIN'01: No description on Graphics interface 2001*, Canadian Information Processing Society, Toronto, Ont., Canada, pp 9–17 (2001)
- Ohbuchi R, Mukaiyama A, Takahashi S, A frequency-domain approach to watermarking 3d shapes. *Computer Graphics Forum* 21:373–382 (2002)
- Praun E, Hoppe H, Finkelstein A, Robust mesh watermarking. In: *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp 49–56 (1999)
- Robust mesh watermarking based on multiresolution processing
- Kanai S, Date H, Kishinami T, Date SKH, Digital watermarking for 3d polygons using multiresolution wavelet decomposition, In: *Proc. Sixth IFIP WG 5.2 GEO-6*, pp 296–307 (1998)

12. Uccheddu F, Corsini M, Barni M, Wavelet-based blind watermarking of 3d models, In: Proceedings of the 2004 workshop on Multimedia and security, ACM, New York, USA, pp 143–154 (2004)
13. Zafeiriou S, Tefas A, Pitas I, Blind robust watermarking schemes for copyright protection of 3d mesh objects, IEEE Transactions on Visualization and Computer Graphics 11(5):596–607 (2005)
14. Song HS, Cho NI, Robust Watermarking of 3D Polygonal Meshes, IEICE Transactions on Information and Systems, E91-D(5) (2008)
15. Lee SH, Kwon KR, Mesh Watermarking based projection onto two convex sets, Multimedia Systems 13:323–330 (2008)
16. Chao MW, Lin Ch, Yu CW, Lee TY, A high capacity 3d steganography algorithm, IEEE Transactions on Visualization and Computer Graphics 15(2):274–284 (2009)
17. Yeo BL, Yeung MM, Watermarking 3d objects for verification, IEEE Comput Graph Appl 19(1):36–45 (1999)
18. Chou CM, Tseng DC, A public fragile watermarking scheme for 3d model authentication, Comput Aided Des 38(11):1154–1165 (2006)
19. Lin HY, Liao HY, Lu CS, Lin JC, Fragile Watermarking for Authenticating 3-D Polygonal Meshes, IEEE Trans, on Multimedia, 7(6):997–1006 (2005)
20. Lee JJ, Cho NI, Lee SU, Watermarking algorithms for 3d nurbs graphic data, EURASIP J Appl Signal Process 2004:2142–2152 (2004)
21. Garcia E, luc Dugelay J, Member S, Texture-based watermarking of 3-d video objects, IEEE Transactions on Circuits and Systems for Video Technology 13:853–866 (2003)
22. Cotting D, Weyrich T, Pauly M, Gross M, Robust watermarking of point-sampled geometry, In: SMI '04: Proceedings of the Shape Modeling International 2004, IEEE Computer Society, Washington, DC, USA, pp 233–242 (2004)
23. Yao CY, Lee TY, Adaptive geometry image, IEEE Transactions on Visualization and Computer Graphics 14(4):948–960 (2008)
24. Cox IJ, Member S, Kilian J, Leighton FT, Shamoon T, Secure spread spectrum watermarking for multimedia, IEEE Transactions on Image Processing 6:1673–1687 (1997)
25. Hoppe H, Progressive meshes, (1996)
26. Taubin G, Estimating the tensor of curvature of a surface from a polyhedral approximation, In: ICCV '95: Proceedings of the Fifth International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA, pp 902 (1995)
27. Meyer M, Desbrun M, Schroder P, Barr AH, Discrete differential-geometry operators for triangulated 2-manifolds, (2002)
28. Alliez P, Cohen-Steiner D, Devillers O, Lévy B, Desbrun M, Anisotropic polygonal remeshing, In: SIGGRAPH '95: Proceedings, pp 485–493 (2003)
29. Sorkine O, Cohen-Or D, Lipman Y, Alexa M, Rössl C, Seidel HP, Laplacian surface editing, In: SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, pp 175–184 (2004)
30. Allchin D, Error Types. In: Perspectives on Science '01: MIT Press Perspectives on Science, MIT Press, pp 38–58 (2001)
31. Lee TY, Yao CY, Chu HK, Tai MJ, Chen CC, Generating genus-n-to-m mesh morphing using spherical parameterization: Research articles, Comput Animat Virtual Worlds 17(3-4):433–443 (2006)
32. Taubin G, A signal processing approach to fair surface design, In: SIGGRAPH '95: Proceedings, pp 351–358 (1995)
33. Wang YS, Lee TY, Curve-skeleton extraction using iterative least squares optimization, IEEE Trans. TVCG, 14(4):926–936 (2008)
34. Garland M, Heckbert PS, Surface simplification using quadric error metrics, In: SIGGRAPH '97: Proceedings, pp 209–216 (1997)
35. Lee TY, Lin YC, Sun YN, Lin LW, Fast Feature-based Metamorphosis and Operator Design, In: Computer Graphics, pp 15–22 (1998)