

A Robust High Capacity Affine-Transformation-Invariant Scheme for Watermarking 3D Geometric Models

XIFENG GAO, Department of Computer Science, University of Houston
CAIMING ZHANG, YAN HUANG, School of Computer Science and Technology, Shandong University
and ZHIGANG DENG, Department of Computer Science, University of Houston

In this paper, we propose a novel, robust and high capacity watermarking method for 3D Meshes with arbitrary connectivities in spatial domain based on affine invariants. Given a 3D mesh model, a watermark is embedded as affine-invariant length ratios of one diagonal segment to the residing diagonal intersected by the other one in a coplanar convex quadrilateral. In the extraction process, a watermark is recovered by combining all the watermark pieces embedded in length ratios through majority voting. Extensive experimental results demonstrate the robustness, high computational efficiency, high capacity, and affine-transformation-invariant characteristics of the proposed approach.

Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*Hierarchy and geometric transformations*; K.6.5 [Management of Computing and Information Systems]: Security and Protection—*Authentication*

General Terms: Algorithms

Additional Key Words and Phrases: 3D model watermarking, affine transformation invariant, copyright protection, 3D model authentication, and high capacity

ACM Reference Format:

Gao, X., Zhang, C., Huang, Y., and Deng, Z. 2012. A Robust High Capacity Affine-Transformation-Invariant Scheme for Watermarking 3D Geometric Models. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1, Article 1 (January 2012), 20 pages. DOI = 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

1. INTRODUCTION

With the rapid development of 3D digital scanning technology and the fast growing popularity of video games, 3D geometric models have become the main exchanging form of 3D graphics data on the Internet. Therefore, how to efficiently protect the intellectual property rights and certificate the authentication of these 3D data (that is, ensure the received 3D models are from the proper senders, without

Emails: {xgao6|zdeng4}@uh.edu, {czhang|yan.h}@sdu.edu.cn.

This work is in part supported by the National Nature Science Foundation of China (61020106001, 60933008) and US National Science Foundation IIS-0914965.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2012 ACM 1551-6857/2012/01-ART1 \$10.00

DOI 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

uninformed manipulations in the process) have caught increasing attention from both academic researchers and industry practitioners in recent years.

In this paper, we propose a robust and high capacity method to watermark 3D Meshes with arbitrary topological connectivities. It can successfully extract the full watermark from 3D meshes even damaged by typical attacks, and extract a partial watermark if the models are severely damaged. Our approach is based on affine-invariants that are defined as the length ratios of two segments of a diagonal intersected by the other one in a coplanar convex quadrilateral. For each vertex in an inputted 3D mesh, we calculate its affine-invariant length ratios in a carefully selected coplanar quadrilateral constructed within a local neighborhood of vertices. By slightly changing these ratios while keeping the quadrilateral coplanar, we embed pieces of a watermark (chopped into pieces beforehand) into the model. Watermark pieces are indexed and embedded multiple times.

For the purpose of comparison, researchers generally evaluate the performance of a watermarking method in terms of the following aspects: (i) *Generality*. It is able to embed watermarks into different models with various geometry and connectivity characteristics. (ii) *Capacity*. It is able to embed watermarks that are large enough to meet application requirements. (iii) *Invisibility*. The embedded watermarks need to be imperceptible and a watermarked model does not have any obvious visual artifacts. (iv) *Robustness*. The watermarks can be still extracted from the watermarked models that undergo substantial distortion or damage. (v) *Efficiency*. It need to be computationally efficient with a low memory requirement. Based on these terminology definitions, our proposed watermarking method is generic, and achieves a provably outstanding trade-off among capacity, invisibility, robustness, and efficiency. Specifically, (1) in terms of *generality*, our approach can handle 3D mesh models with arbitrary topological connectivities due to the fact that a coplanar quadrilateral can always be obtained from a local neighborhood of vertices, regardless the complexity of the inputted 3D models. (2) In terms of *capacity*, our proposed method chops a watermark into pieces and the length of each piece can be adjusted according to application requirements. The longer each watermark piece is, the larger the embedding capacity will be. (3) In terms of *invisibility*, our proposed approach places watermark pieces at different places of length ratio such that a good trade-off between invisibility and robustness can be achieved. (4) In terms of *robustness against various attacks*, 3D models watermarked by our proposed method provide strong robustness against affine transformation, cropping, vertex reordering, and various local attacks, since i) the used length ratios are affine invariant; ii) watermark pieces are locally indexed; and iii) coplanar quadrilaterals are constructed independently. (5) In terms of *computational efficiency*, our proposed method is approximately linear in both the watermark embedding and the watermark extraction processes.

2. RELATED WORK

Digital watermarking techniques have been widely used in image, audio, video and multimedia computing. To date, a variety of approaches have also been proposed to watermark 3D geometric models [Wang et al. 2011]. In this section, we briefly review existing 3D watermarking methods by dividing them into transform domain methods and spatial domain methods.

Transform domain 3D watermarking: Transform domain methods embed a watermark by modifying the transformed coefficients of direct frequency analysis [Ohbuchi et al. 2002; Cayre et al. 2003; Luo and Bors 2008], multi-resolution analysis [Praun et al. 1999; Yin et al. 2001; Wang et al. 2008], or manifold harmonics analysis [Liu et al. 2008; Wang et al. 2009] of an inputted 3D model.

Based on Laplacian frequency analysis, researchers embed watermarks by additively modulating the low and medium frequency coefficients [Ohbuchi et al. 2002], by quantizing the low and medium frequency coefficients [Cayre et al. 2003], or by utilizing constraints embedded in sets of high frequency coefficients [Luo and Bors 2008]. However, for Laplacian analysis based watermarking methods, there

are two well-known limitations: i) computational cost increases rapidly with the increase of mesh complexity; ii) connectivity information is needed in the analysis procedure [Wang et al. 2008].

Using wavelet analysis, Wang et al. [2008] proposed a hierarchical watermarking framework to embed three watermarks for different purposes. Praun et al. [1999] extend the spread spectrum [Cox et al. 1997] to 3D models and apply the multi-resolution analysis method [Hoppe 1996] for non-blind digital watermarking. Based on the Burt-Adelson pyramid decomposition of multi-resolution analysis [Guskov et al. 1999], Yin et al. [2001] proposed a robust and non-blind watermarking algorithm.

Based on the manifold or spheroidal harmonics analysis, several recent blind spectral methods [Liu et al. 2008; Konstantinides et al. 2009; Wang et al. 2009] can resist many attacks except cropping attacks and affine transformations. By transforming an inputted mesh model into the spherical parameterization domain, Li et al. [2004] reported a watermarking method that can handle various attacks using informed detection. Lin et al. [2010] proposed a semi-blind robust watermarking method that can withstand many attacks including pose deformation.

Spatial domain 3D watermarking: When watermarks are embedded in the spatial domain, most watermarking algorithms change the geometry of an inputted model. The Tetrahedral Volume Ratio (TVR) method proposed by Ohbuchi et al. [1997] is the first watermarking algorithm that can resist affine transformations. However, TVR is designed for triangular meshes only and can not be used for non-manifold models. Later, Cho et al. [2007] present two blind watermarking algorithms by manipulating the distribution of vertex norms - shifting the mean value of the distribution and changing the variance, respectively. The two algorithms achieve robustness against many attacks by sacrificing embedding capacity. However, they are not applicable to small models and are vulnerable to cropping attacks and affine transformations.

In order to achieve robustness against affine transformations and simplifications, a combination of three methods, vertex flood algorithm (VFA), affine invariant embedding (AIE) and normal bin encoding (NBE), was proposed by Benedens and Busch [2000]. However, this combined method is not completely blind since some information produced by the original model must be provided in the extracting process. Researchers also developed principal component analysis (PCA) based methods [Zafeiriou et al. 2005; Kalivas et al. 2003] for 3D watermarking. Although the PCA-based methods are blind and can survive similar transformations, they are vulnerable to cropping attacks and non-uniform transformations due to their dependence on the centroid of an inputted model.

By perturbing the distance between local vertices to the center of the model, Yu et al. [2003] proposed a non-blind watermarking algorithm that can withstand signal processing attacks, but it is fragile to geometric transformations such as affine transformations. Lee and Kwon [2007] proposed a watermarking scheme which shows excellent robustness with limited capacity. However, their method can only be applied to large 3D models and the time complexity is very high.

Using a clustering algorithm, Agarwal and Prabhakaran [2007] proposed a robust, spatial and blind watermarking method for 3D point sampled geometry. Kuo et al. [2009] present a blind watermarking method that modifies vertices on the creases and corners of a triangular mesh based on the principle of moment-preserving [Cheng and Wu 2005]. However, this method can only be applicable to triangular meshes. Gao et al. [2010] proposed a high capacity and invisibility watermarking method based on similarity-transformation invariants constructed in four-points sets. This method can tune the watermark invisibility linearly, but it cannot resist file attacks, which narrows its range of applications.

Compared to the spatial domain watermarking methods, the transform domain watermarking methods usually achieve better robustness against signal processing attacks; however, they often require a higher computational cost and extra information in the watermark extraction process. This is not favorable to applications where speed and efficiency are expected or required (e.g., 3D gaming over the Internet). The spatial domain watermarking approach proposed in this paper can achieve a high

capacity, tunable invisibility, strong robustness (with respect to some typical attacks), and low computational cost for copyright protection of 3D models.

Comparisons with existing methods: To date there are two kinds of invariants employed in 3D watermarking research that are robust against affine transformations: TVR (Tetrahedral Volume Ratio) methods [Ohbuchi et al. 1998; Benedens 2000] and Nielson-Foley norm based methods [Benedens and Busch 2000; Wagner 2000]. The TVR methods embed watermark bits into ratios of volumes of tetrahedrons constructed by three “neighbored” triangles. They are restricted to orientable 2-manifold triangular mesh models. The TVR methods also depend on connectivity information of models such as consistent order of faces [Ohbuchi et al. 1998] or vertices [Benedens 2000]. Therefore, they are restricted to meshes entirely consisting of triangles; no planar polygons with more than three vertices can be processed. To use the Nielson-Foley norm to insert watermarks, Benedens and Busch [2000] quantize this norm while Wagner [2000] replaces medium-important bits of this norm. Both of them can deal with non-manifold models.

Our proposed approach is also affine-transformation-invariant. However, compared with the above TVR methods [Ohbuchi et al. 1998; Benedens 2000] and Nielson-Foley norm based methods [Benedens and Busch 2000; Wagner 2000], our approach has the following major distinctions:

- Generality.* Our approach can handle 3D mesh models with any topological connectivities including manifold or non-manifold. For the TVR methods [Ohbuchi et al. 1998; Benedens 2000], they are restricted to triangular meshes with orientable 2-manifolds. Although the Nielson-Foley norm based methods [Benedens and Busch 2000; Wagner 2000] can handle manifold and non-manifold mesh models, when the embedding primitive is a plane or nearly plane, the Nielson-Foley norm cannot be used for embedding watermarks.
- Capacity.* Our approach provides a higher embedding capacity than the above approaches [Ohbuchi et al. 1998; Benedens 2000; Benedens and Busch 2000; Wagner 2000].
- Robustness.* Besides affine transformations, our approach can handle file attacks such as vertex or face reordering, while both the TVR methods [Ohbuchi et al. 1998; Benedens 2000] and the Nielson-Foley norm based methods [Benedens and Busch 2000; Wagner 2000] cannot handle such attacks.

In addition, while our proposed method shares certain features with the existing approach described in [Gao et al. 2010], such as, both of them are geometric transformation invariants based (one is affine-transformation-invariants, the other one is similarity-transformation-invariants) and invariants are constructed in four points sets, there are major differences between them: (1) *Conceptual differences.* A similarity-transformation-invariant for a geometry shape means when this shape undergoes only rotation, translation, or uniform scaling, the similarity-transformation-invariant will not change. However, for an affine-transformation-invariant, not only rotation, translation, and uniform scaling, but also shearing, un-uniform scaling, and reflection, when they are added to the geometry shape, the affine-transformation-invariant still keeps the same. (2) *Construction differences.* In the proposed method, the affine-transformation-invariants are constructed by slightly moving one vertex of a four points set to make this set coplanar, and when the watermarking algorithm processed to this step, the geometry information of 3D meshes have been changed. While, in [Gao et al. 2010], none of the four vertices are changed, and the similarity-transformation-invariants are obtained from a virtual projected quadrilateral. In sum, the major differences between the two methods are the conception and construction of similarity-transformation-invariants and affine-transformation-invariants, which lead to the differences in watermarking process and those of experimental results.

3. AFFINE INVARIANT

An affine invariant is a geometric invariant defined as follows [Huttenlocher 1991]:

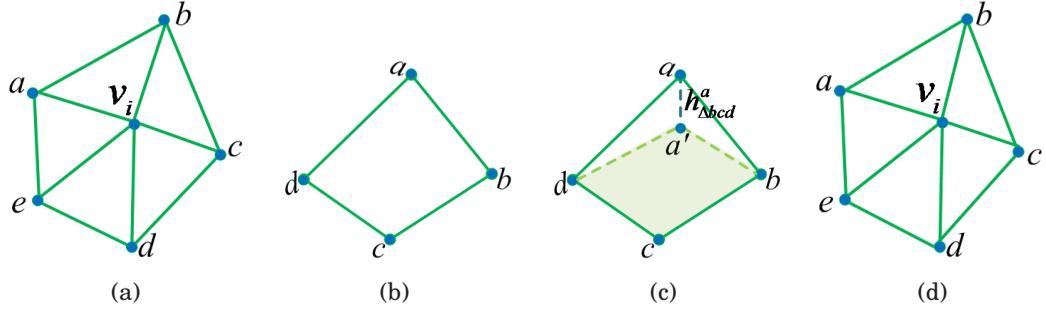


Fig. 1. An example of coplanar quadrilateral construction and embedding-cell selection. (a) Neighborhood of v_i . (b) A four-point set in $N(v_i)$. (c) a is projected to the plane Δbcd . (d) $N'(v_i)$ after disturbing c (a, b, c', e are now coplanar).

Definition 1: Given a space M and a transformation group G acting on that space, we say that $f(x)$ is an invariant of the geometric object $x \subset M$ under G if and only if $f(x) = f(g(x))$ for any $g \in G$.

Affine invariants are very favorable for watermarking since watermarks can be embedded into these invariants that are preserved under any affine transformations including uniform scaling, non-uniform scaling, rotation, shearing, reflection and so on. It's also well known that length ratios of three collinear points are affine invariants [Efimov 1980]. The length ratios of three collinear points can also be extracted from an approximately coplanar four-points set from a 3D point cloud in surface registration algorithm [Aiger et al. 2008]. Due to the above characteristics of length ratios, we embed watermarks in aforementioned length ratios extracted from a coplanar convex quadrilateral in our proposed method.

In the remainder of this section, we first describe the affine invariants defined in coplanar quadrilaterals and then detail the construction procedure of coplanar quadrilaterals in 3D models.

3.1 Affine Invariants of Coplanar Quadrilaterals

Let a, b, c, d be four coplanar points, where any three points are not collinear. The four points define a convex polygon in a three-dimensional space and the diagonals of ac and bd meet at point o . Let r_1 be the ratio of ao and ac (i.e., ao/ac) and r_2 be bo/bd . Then r_1 and r_2 are preserved under any affine transformations. Let $a'b'c'd'$ be the resultant quadrilateral after an affine transformation is applied to $abcd$, and $a'o'$ and $b'd'$ meet at point o' . Also, let $r'_1 = a'o'/a'c'$ and $r'_2 = b'o'/b'd'$. Then, the four points of a', b', c' and d' keep coplanar, that is, $r_1 = r'_1$ and $r_2 = r'_2$. Note that the geometry invariant defined in [Gao et al. 2010] comes from the projected ratios of a four-points-set. That's why the invariant ratios can not remain unchanged under non-uniform affine transformations.

3.2 Construction of Coplanar Quadrilaterals in 3D Models

A 3D model is composed of a set of vertices V and a set of edges E between these vertices. Let v_i denote the i^{th} vertex. We define the k -ring neighbors of v_i as $N_{v_i}^k$. The definition of $N_{v_i}^k$ is given by Eq. (1):

$$N_{v_i}^k = \begin{cases} v_i & k = 0, \\ N_{v_i}^{k-1} \cup \{v_j \mid \exists v_l \in N_{v_i}^{k-1} \cap (v_l, v_j) \in E\} & k > 0. \end{cases} \quad (1)$$

In the above equation, k determines the size of a local neighborhood. The larger k is, the larger $N_{v_i}^k$ will be. A larger $N_{v_i}^k$ provides more candidates for coplanar quadrilateral selection with a much higher computational cost which can be referred from the discussion in the second item of Section 5.1. However, from various experiments, we observe that a larger local neighborhood has little contribution to increase embedding capacity and has almost no benefit to improve watermark invisibility and

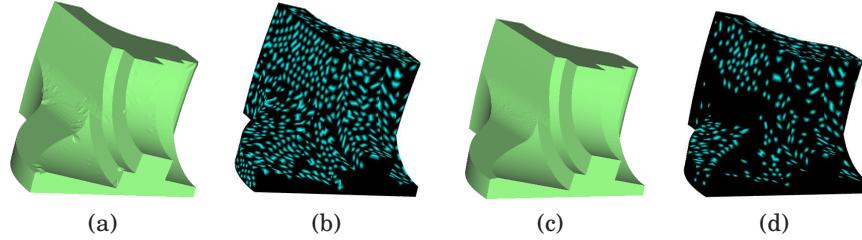


Fig. 2. The impact of λ on the visual effect of the Fandisk model (light blue dots represent embedding-cells obtained by changing the geometry of the original vertices). (a) Apparent indentations ($\lambda = 10^{-1}$). (b) More embedding-cells ($\lambda = 10^{-1}$). (c) No visible artifact ($\lambda = 10^{-6}$). (d) A reduced number of embedding-cells ($\lambda = 10^{-6}$).

robustness. Therefore, we constantly keep k to be 1 throughout this paper. In this case, our proposed method can achieve a time complexity $O(V)$ as further discussed in Section 5.1. For the convenience of description, we use $N(v_i)$ instead of $N_{v_i}^1$ in the rest of this writing.

In our watermarking method, we employ the affine invariant ratios described in Section 3.1 to embed watermarks for 3D models. In order to obtain those length ratios, coplanar convex quadrilaterals (also known as embedding-cells) have to be constructed for vertices in an inputted model. Coplanar quadrilateral construction and embedding-cell selection are described as follow:

- (1) Scan $N(v_i)$ to look for coplanar four-point sets. If none, turn to step (3).
- (2) If there is only one set, we choose it as the embedding-cell for $N(v_i)$; if there is more than one set, we choose the one with longest diagonals in order to increase robustness against noise attacks. Exit the construction procedure.
- (3) When there are no coplanar four points in $N(v_i)$, we construct and select an embedding-cell as illustrated in Fig. 1:
 - (a) For each four-point set $\{a, b, c, d\}$ in $N(v_i)$, there are four approximately coplanar quadrilaterals like $\{a', b, c, d\}$ as shown in Fig. 1(c). Calculate the projection distances from a, b, c, d to the plane $\Delta bcd, \Delta acd, \Delta abd$ and Δabc , respectively. Let $h_{\Delta bcd}^a, h_{\Delta acd}^b, h_{\Delta abd}^c$ and $h_{\Delta abc}^d$ be the corresponding projection distances.
 - (b) Repeat the above calculations until all four-point sets of $N(v_i)$ have been checked. Let H_{v_i} be the set of all projection distances.
 - (c) Given H_{v_i} , we choose the minimum distance (i.e., $h_{\Delta abc}^c$). Then $\{a, b, c, e\}$ is the selected approximately coplanar quadrilateral. Let c' be the projection of c to the plane Δabc . Then the coplanar quadrilateral of $\{a, b, c', e\}$ is the selected embedding-cell for $N(v_i)$ and $N(v_i)$ is updated to $N'(v_i)$ after changing the geometry of c as shown in Fig. 1(d).

Given $N(v_i)$ of v_i , let n_i be the number of vertices in $N(v_i)$, then the total number of all four-point sets in $N(v_i)$ is $C_{n_i}^4$. Note that n_i should be no less than 4 in most cases. In our proposed method, we also set a threshold of λ to filter projection distances obtained in the construction process. Working as an antialiasing filter, λ filters out projection distances that are too large to create apparent visual artifacts. For example, in Fig. 2(a), when λ is set to be 10^{-1} , indentations are observed on the sharp edges of the Fandisk model; when λ is set to be 10^{-6} , modifications made to the geometry are hardly visible (Fig. 2(c)). This phenomenon is specially evident for models with sharp edges like the Fandisk model. However, as the value of λ gets smaller, the total number of embedding-cells significantly decreases (as shown in Fig. 2(d) and compared to Fig. 2(b)). With less embedding-cells, less copies of a watermark are embedded into the inputted model and the robustness of our proposed method is reduced accordingly. Therefore, we select the λ carefully in order to balance the visual effect and robustness of the watermarked model.

4. WATERMARKING PROCESS FOR A 3D MODEL

After the applied affine invariant ratios and their construction procedures (described above), we now describe the proposed watermarking method in detail. Basically, our proposed method consists of three steps: *watermark preprocessing*, *watermark embedding* and *watermark decoding*. Before a watermark is embedded, we firstly divide it into pieces and assign a unique index to each piece. A watermark piece and its associated index are embedded into the two length ratios defined in Section 3.1. When a watermark is extracted, the same length ratios are calculated, and the embedded watermark pieces and indices are retrieved from those ratios. A watermark is finally recovered by concatenating those watermark pieces according to their individual indices.

4.1 Watermark Preprocessing

The watermark is generated by a cryptographic hash function or other transformations with the input secret key. The secret key is selected by combination operations of algorithm parameters H_1 , H_2 , L_1 , L_2 , and λ with other additional information. Due to the fact that a watermark can always be represented by a string of 0s and 1s, we address watermark sequences composed of 0s and 1s only in this paper. Given the string of 0s and 1s, we convert it to an octal sequence. Then we divide the octal sequence into sequential groups, each of which is identified by a unique index. Let w_i be the i^{th} group and (i, w_i) be the i^{th} watermark piece. Let W denote the set of all the watermark pieces, then $W = \{(0, w_1), (1, w_2), \dots, (n-1, w_n)\}$, where $i = 0, 1, \dots, n-1$.

4.2 Watermark Embedding

In our proposed method, the watermark embedding procedure consists of two major steps: *embedding cell construction and selection* (for details, refer to Section 3.2), and *watermark piece embedding*. Watermark pieces are embedded into the length ratios by slightly changing their values while keeping the quadrilaterals coplanar. Details of length ratio modification are given as follow.

Given an embedding-cell $\{a, b, c, d\}$ and a watermark piece (i, w_i) , the length ratios to be used for watermark embedding are r_1, r_2 as defined in Section 3.1. Let L and H be the distances from the decimal point to the low notation and the high notation, respectively (as shown in Fig. 8). A watermark piece w_i and its individual index i are embedded into r_2 and r_1 by replacing their original decimal values from L to H with w_i and i , respectively. Note that, L and H may have different values with respect to r_1 and r_2 . How to optimally select L and H is further discussed in Section 5.3.

Given a convex coplanar quadrilateral $\{a, b, c, d\}$ as in Fig. 3(b), there are four candidate length ratios to be chosen for watermark embedding, namely, ao/ac , bo/bd , co/ac and do/bd . In order to identify and differentiate r_1 and r_2 without any prior knowledge of vertex order, r_1 and r_2 are chosen to satisfy the following relationship:

$$r_1 < r_2 < 0.5. \quad (2)$$

In case any of those four length ratios equals to 0.5, we move the corresponding vertex inwards a little bit, following the direction of the diagonal it resides on. An example is shown in Fig. 3(a). Given the length ratios of different diagonals, if their decimal notation right before L are the same, they are treated as equal (e.g., $ao/ac = bo/bd$). In this case, we subtract 1 from the $\{L-1\}^{th}$ decimal notation of either ao/ac or bo/bd such that they are no longer equal. After these adjustments, we can guarantee that there are two length ratios both of which are less than 0.5 and one is smaller than the other.

Let the smaller length ratio be r_1 and the other be r_2 . After a watermark piece is embedded, r_1 and r_2 are changed into r'_1 and r'_2 as illustrated in Fig. 3(c). Note that r'_1 and r'_2 still satisfy the relationship of $r'_1 < r'_2$. This is because the decimal notations before L of r'_1 and r'_2 are not affected by the embedding

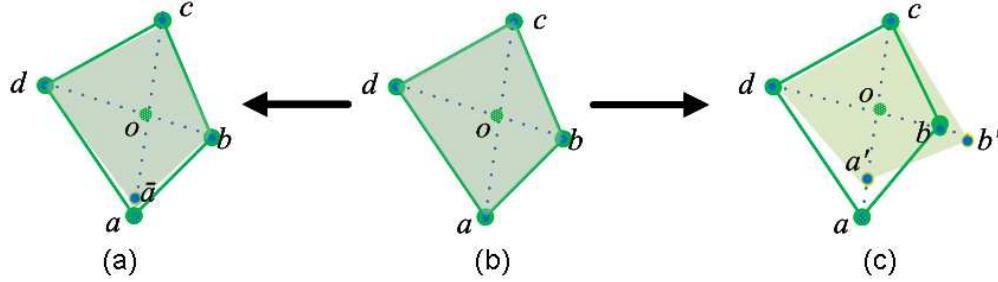


Fig. 3. Embed watermark pieces into length ratios. (a) r_1 and r_3 after a is adjusted. (b) The original embedding-cell. (c) The modified embedding-cell after the watermark piece (i, w_i) is embedded.

of a watermark piece. Since the changes of r_1 and r_2 are done by adjusting the coordinates of a and b along the direction of diagonals they reside on, the coordinate of o keeps constant, and the changes of r_1 and r_2 are mutual independent. Let a' and b' be the adjusted (new) vertices of a and b , respectively, then a' and b' are calculated as follows:

$$a' = (o - r'_1 \cdot c) / (1 - r'_1), \quad (3)$$

$$b' = (o - r'_2 \cdot d) / (1 - r'_2).$$

$\|a' - a\|$, the distance of between a' and a , is calculated as follows:

$$\|a' - a\| = \frac{\|r_1 - r'_1\|}{1 - r'_1} \|c - a\|, \quad (4)$$

where $\|c - a\|$ is the length of ac and $1 - r'_1 > 0.5$.

Given L and H , we have

$$\|a' - a\| \approx 2 \times P \times \|c - a\|, \quad (5)$$

where $P = (10^{-L}x_l + 10^{-L-1}x_{l+1} + \dots + 10^{-H+1}x_{h-1} + 10^{-H}x_h)$. $x_l, x_{l+1}, \dots, x_{h-1}, x_h$ are the corresponding decimal notations of $\|r_1 - r'_1\|$.

From Eq. (5), we can see that the invisibility of our proposed method mainly depends on the value of L . It is clear that as the value of L decreases, the distance of $\|a' - a\|$ increases. In other words, as watermark pieces are placed closer to the decimal point, more distortion can be observed in a watermarked model. This effect is demonstrated in Fig. 4, which shows two watermarked Fandisk models with $L = 2$ and $L = 3$, respectively.

As to the robustness to noise, the performance of our proposed method is affected by not only the value of L but also the value of H . Given the value of H , when the strength of attacking noise is above $10^{-H} \times \|c - a\|$, extracted watermarks start to show errors. In general, as H gets smaller, the robustness to noise increases. An in-depth discussion of L and H selection is presented in Section 5.3.

The pseudo code for embedding a watermark is described as follows:

- (1) Set the tags of all the vertices in V as “un-used” and initialize the set of embedding-cells Ω as \emptyset .
- (2) For each vertex v_i in V ,
 - (a) Construct its local neighborhood of $N(v_i)$. Note that if the tags of some vertices in $N(v_i)$ are “used”, we discard them from $N(v_i)$.
 - (b) Construct all coplanar quadrilaterals in $N(v_i)$ using the method described in Section 3.2.
 - (c) Select the embedding cell as described in Section 3.2 and set the tags of the four points in the embedding-cell as “used” and add this embedding-cell to Ω .

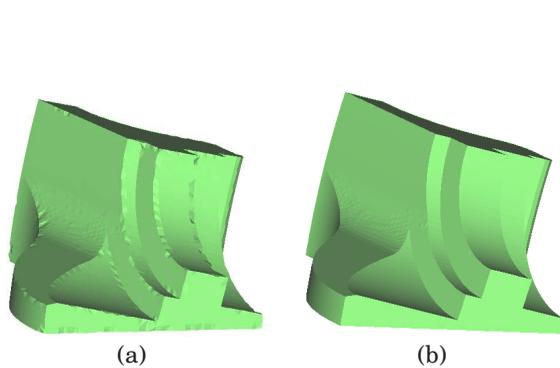


Fig. 4. the impact of L on the visual effect of the Fandisk model. (a) Visible artifacts are clearly observable when $L = 2$, (b) Visible artifacts are not so apparent when $L = 3$.

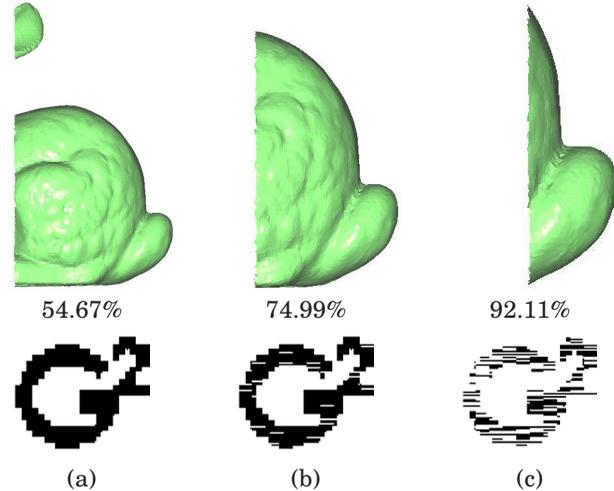


Fig. 5. watermarks extracted from the Bunny model at different cropping ratios

(3) Embed a watermark piece into every embedding-cell in Ω by employing the method described above.

Note that multiple copies of a watermark are embedded into a 3D model in our proposed method. The purpose is to enhance the robustness of a watermarked model against various attacks.

4.3 Watermark Decoding

Similar to the above watermark embedding procedure, the decoding process consists of the following steps:

- (1) Set the tags of all the vertices as “un-used” and initialize the set of watermark pieces W as \emptyset .
- (2) For each vertex v_i in V ,
 - (a) Construct its local neighborhood of $N(v_i)$. Same as the embedding procedure, if the tags of some vertices in $N(v_i)$ are “used”, we discard them from $N(v_i)$.
 - (b) Search for coplanar four-point set in $N(v_i)$. If there are more than one set in $N(v_i)$, we choose the one with the longest diagonals and set it as the embedding-cell.
 - (c) Set the tags of the four points in the embedding-cell as “used”.
 - (d) Extract a watermark piece with its corresponding index from the length ratios of the embedding-cell and add it to W . Note that the selected length ratios here should satisfy Equation (2). Otherwise, we simply regard this embedding-cell is severely damaged and discard it.
- (3) Given the complete set of W , if the watermark pieces in W can cover the entire watermark, go to step (5); otherwise, the watermark is distorted by some attacks, go to step (4) and extract the watermark pieces from the “un-used” vertices.
- (4) Traverse all the vertices tagged as “un-used” and find an approximately (up to some allowed tolerance τ) coplanar four-point set in $N(v_i)$ and label the four points as “used”. Then, extract the watermark piece and add it to W . Continue to traverse the next vertex until W can construct the entire watermark. If the attacks are very serious, the extracted watermark pieces may not be correct.

(5) Sort the extracted w_i according to i , and then translate the set of watermark pieces W into the original watermark by majority voting.

Finally, to identify the extracted watermark image, we transform it into the original watermark image which shows the information representing the copyright using anti-Arnold transformation and the secret key.

5. ALGORITHM ANALYSIS

5.1 Time Complexity

In this subsection, we conduct an asymptotic performance analysis on the computational cost of the proposed watermarking method. We focus on the cost associated with major algorithmic steps and the most expensive operations in each step. Consider a 3D model with V vertices, E edges and F faces, the most computationally expensive operations of the proposed method are the following:

—*Local neighborhood construction*: 1-ring neighborhood of all the vertices can be constructed after traversing each edge in an inputted model once. Considering Euler's formula of $V - E + F = 2$ for a closed manifold triangular mesh with a genus of zero, and $2E = 3F$, we have $E = 3V - 6$. Therefore, the computational cost incurred by the local neighborhood construction is $O(3V)$ in total.

—*Embedding-cells construction and selection*: In order to select the right embedding-cell for v_i , all four-point sets of the local neighborhood $N(v_i)$ have to be checked. Assuming the size of $N(v_i)$ is n_i , there are $C_{n_i}^4$ sets altogether. Given a four-point set, our proposed method carries out 4 projection operations to calculate the corresponding distances. Since an embedding-cell is associated with four points, the total number of embedding-cells is bounded by $O(V/4)$. Therefore, the total number of projection operations incurred by the embedding-cell construction is $O(C_{n_i}^4 \times V)$. In most cases, n_i is bounded by a constant. Therefore, the computational cost for the embedding-cell construction and selection is $O(KV)$ where K represents a constant value.

—*Watermark embedding and extraction*: Watermark embedding and extraction take place in each embedding-cell. Given an embedding-cell, we calculate the affine invariant length ratios, adjust them if necessary and embed/extract watermark pieces into/from them. It is clear that the total number of operations performed in each embedding-cell is also bounded by a constant. Therefore, the computational cost of watermark embedding and extraction is also $O(PV)$ where P is a constant.

Based on the above analysis, we conclude that the proposed watermarking method has a computational complexity of $O(3V) + O(KV) + O(PV) = O(V)$.

5.2 Embedding Capacity

Embedding capacity is the amount of data embedded for watermarking. However, in the proposed watermarking method, indices of watermark pieces are also embedded in affine invariant length ratio r_1 . Therefore, when the embedding capacity is considered, we only count the number of watermark pieces embedded in r_2 as the real watermarking data. The embedding capacity of the proposed method is calculated in Equation (6).

$$C = 3 \times (H - L + 1) \times \Omega \quad (6)$$

Here C denotes the pure embedding capacity and 3 is the number of bits used for translating three binary numbers into an octal number.

Based on this calculation, we can infer that the embedding capacity of our proposed method purely depends on the total number of embedding-cells (denoted by Ω) and the length of embedding positions

in affine invariant length ratios (denoted by $H - L + 1$). As described in Section 3.2, Ω is determined by λ , the antialiasing filter of projection distances. In our experiments, we set λ to be 10^{-3} for all the test models. The numbers of embedding-cells for four 3D models used in our experiments are: Fandisk, 1532; Dragon, 3533; Buddha, 5385; Bunny, 5712. Eq. (6) also indicates that as $H - L$ increases, C increases monotonically. However, through various robustness tests in our experiments, we observed that the positions of H and L affect the robustness to some attacks significantly. As shown in Eq. (5), we know that as L and $H - L$ increase, the robustness to noise attacks decreases. Therefore, H and L should be selected carefully according to specific application requirements.

5.3 Selection of H and L

When we choose the values of H and L for r_1 and r_2 in a coplanar quadrilateral, we should first make sure that the length of embedding positions for r_1 could describe all the watermark pieces. Here, setting H_1 and L_1 for r_1 , H_2 and L_2 for r_2 . Then $3 \times (H_2 - L_2 + 1)$ is the number of bits one ratio could afford and $10^{H_1 - L_1 + 1}$ is the largest number of watermark pieces the index can describe. Since r_1 and r_2 are at the same situation when attacks are added, the resistance to certain attacks of our approach would be decreased when L and $H - L$ increases, we simply let $L_1 = L_2$. We should try to make $H - L$ as small as possible and also let $H_1 - L_1$ and $H_2 - L_2$ as equal as possible. Then we have,

$$10^{H_1 - L_1} < \frac{L_W}{3 \times (H_2 - L_2 + 1)} < 10^{H_1 - L_1 + 1}, \quad (7)$$

where $H_1 - L_1 = H_2 - L_2$ and L_W denotes the number of bits in a watermark. For instance, let us embed a watermark with 576 bits, then we have $H_1 - L_1 = H_2 - L_2 = 2$ based on the above Eq. (7). If the value of L is determined, the left parameters will be settled. As stated above, the positions of L can affect the invisibility and robustness of our scheme, a trade-off need to be made between them for specific applications.

6. EXPERIMENTS AND ANALYSIS

We conducted a series of experiments to test invisibility and robustness of the proposed watermarking algorithm. As shown in Fig. 6(a)-(e), six models (*i.e.*, Fandisk, Dragon, Buddha, Bunny, Dinosaur, and Raptor) are used in our experiments.

In order to show the robustness of our approach visually, we employ an image shown in Fig. 6(f) as the watermark used in our experiments and it is presented in two forms: one with a resolution of 90×90 pixels (8100 Bits) and the other with 24×24 pixels (576 Bits). If not otherwise specified, the 90×90 pixels watermark image is used for the tests of invisibility and robustness to cropping, vertex reordering, and affine transformations (with the parameters set as $H = 4$ and $L=2$). The 24×24 pixels watermark image is used for the tests of robustness to noise, mesh simplification and deformation (parameter specifications are described in Section 6.2.4).

6.1 Watermark Invisibility

To demonstrate the watermark invisibility of our algorithm, we show in Fig. 7 the visual renderings of Fandisk, Dragon, and Bunny through the watermarking process, where Fig. 7(d) and Fig. 7(e) give the models after and before watermarking, respectively. Comparing the images in Fig. 7(d) and Fig. 7(e), we can hardly perceive any visual difference. Since the visual effect can be significantly influenced by changes in vertex normal of models, we adopt the PSNR mentioned in [Chao et al. 2009] to calculate the normal distortion between the original model and the watermarked model, which is illustrated in Table I. Note that in the experiments, we use $L = 3$ for CAD models like Fandisk in order to avoid the conspicuous modifications at sharp edges as illustrated in Fig. 4. In our experiments, we employ the

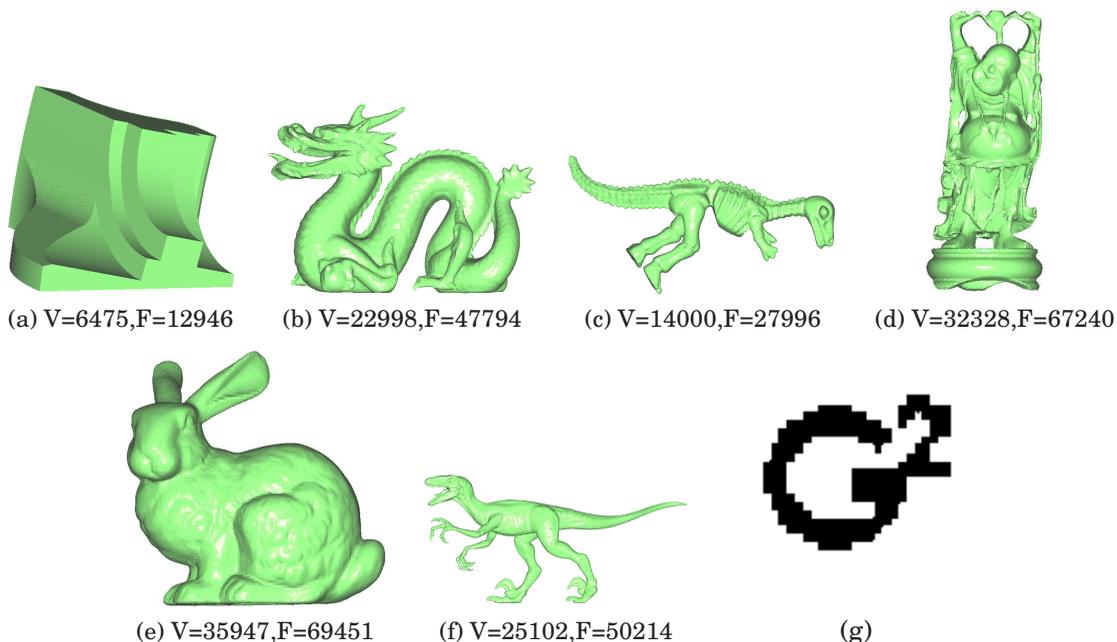


Fig. 6. Original models and the watermark image used in our experiments: (a) Fandisk, (b) Dragon, (c) Dinosaur, (d) Buddha, (e) Bunny, (f) Raptor, and (g) Watermark image. The Bunny, Dragon, and Buddha models are provided courtesy of the Stanford Computer Graphics Laboratory by the Stanford 3D Scanning Repository; the Raptor model is provided courtesy of SenSable Technologies by the AIM@SHAPE Shape Repository; the Fandisk model is provided courtesy of AIM@SHAPE by the AIM@SHAPE Shape Repository; the Dinosaur model is provided courtesy of Cyberware Inc.

commonly used signal-to-noise ratio (SNR) and the Hausdorff distance [Cignoni et al. 1998] as means of measuring the geometrical difference between the watermarked and original models.

As described in Section 4.2, the invisibility of our proposed method is mainly dependent on the embedding positions of affine invariant ratios. We perform a series of tests on the Bunny, Fandisk and Dragon models to observe the impact of L on the Hausdorff distance, and illustrate the results in Fig. 9. Fig. 9 shows that for a given L , as H increases, the Hausdorff distance increases accordingly. When H is larger than 5, this change is much less obvious. This phenomenon can be explained by Eq. (5), which shows the relationship between distance distortion and H and L . It's also shows that the values of Hausdorff distance have a large gap between the Fandisk and the other two models (Bunny and Dragon). This is mainly due to the fact that the average length of embedding-cell diagonals of the Fandisk model is larger than those of the Bunny and the Dragon models. Similarly, Fig. 9 shows that given $H = L+2$, as L increases, the Hausdorff distance decreases accordingly. Fig. 9 also illustrates that after L increases to 4, the values of Hausdorff distance almost keep constant. This can be explained by Eq. (5) too. Briefly speaking, the invisibility of our algorithm is affected by the distance distortion introduced in coplanar quadrilateral construction and watermark embedding.

Statistics of geometry distortions for the test models are given in Table I, where the pure embedding capacity for each model is listed in the “capacity” column. Corresponding visual effects of the watermarked Fandisk, Dragon, and Bunny models are listed in Fig. 7(d). Since we embed watermark pieces into the decimal notations of the affine invariant ratios and one ratio can afford a long sequence of

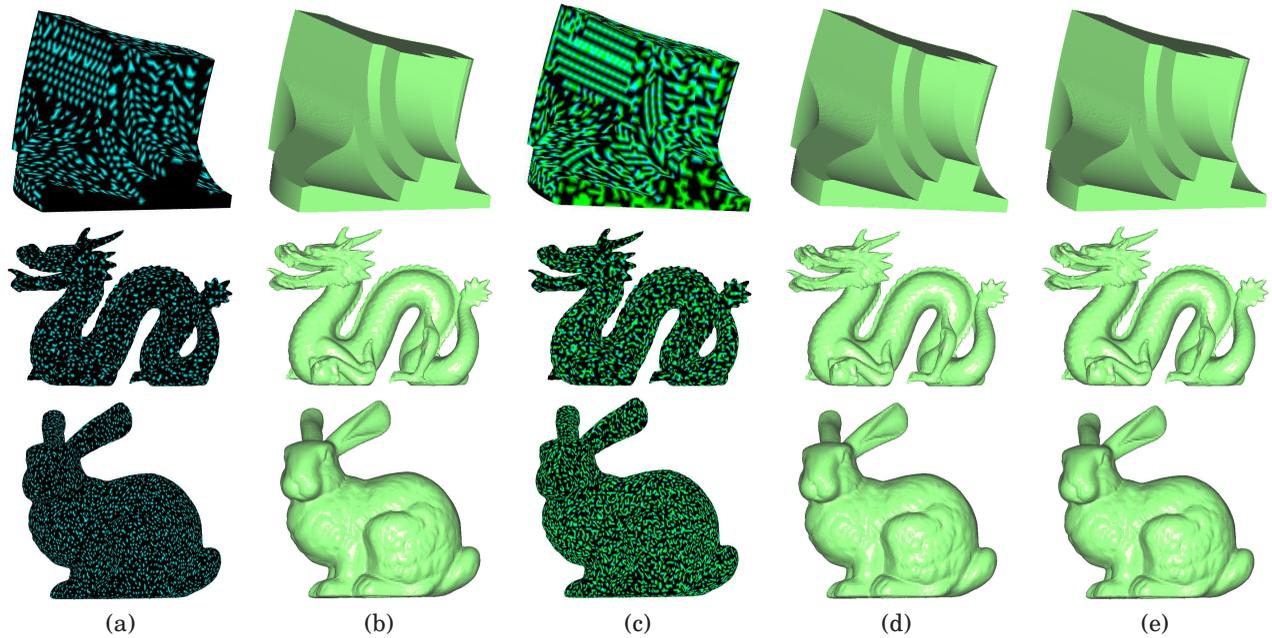


Fig. 7. Step-by-step illustration of the proposed watermarking method:(a) Models marked with vertices (blue dots) whose positions are changed in coplanar quadrilateral construction. (b) Models after coplanar quadrilateral construction. (c) Models marked with vertices (green dots) whose positions are adjusted when embedding a watermark piece. (d) Models embedded with a watermark. (e) The original models.

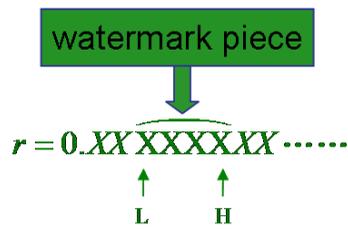


Fig. 8. Replace the decimal notations between L and H with a watermark piece

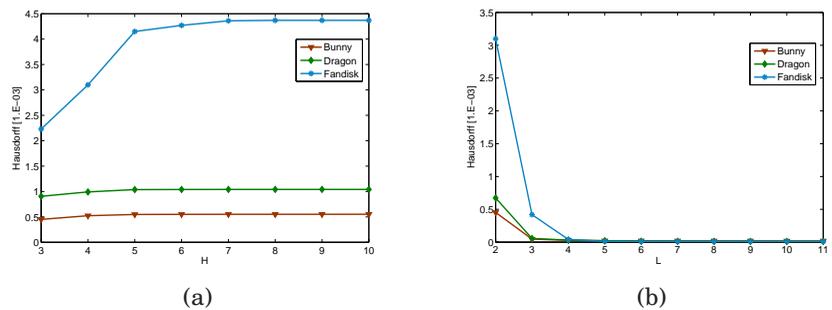


Fig. 9. (a)Geometry distortion comparison at different H (with L = 2), variation of the Hausdorff distance as H increases. (b) Geometry distortion comparison at different L (with H = L + 2), variation of the Hausdorff distance as L increases.

Table I. SNR, PSNR, and Hausdorff Distances for Test Models

Model	Capacity[bit]	Hausdorff[10^{-3}]	SNR[db]	PSNR[db]
Fandisk	13788	0.421	61.24	67.60
Dragon	31797	0.922	51.59	64.67
Buddha	48465	0.737	54.50	66.43
Bunny	51408	0.548	51.39	70.02

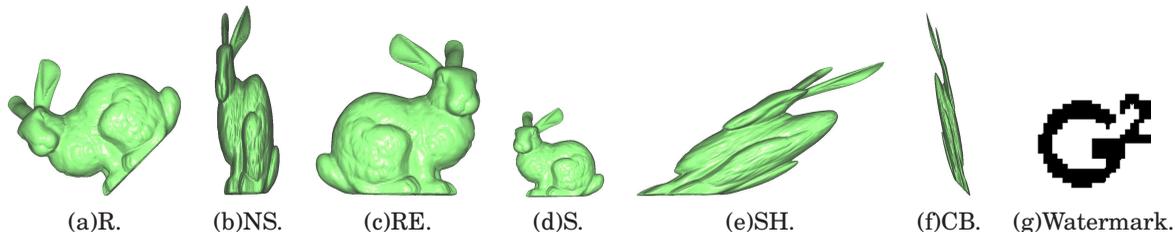


Fig. 10. The embedded watermark can be fully extracted from affine-transformed watermarked models.

watermark bits, compared to other methods in the literature, our proposed algorithm achieves a good watermark invisibility while retaining a high capacity as shown in Table I.

6.2 Robustness

As stated in Section 4.2, an embedded watermark can be fully extracted from unattacked models with our proposed method. To demonstrate the robustness of the proposed method, we use various attacks including affine transformation, cropping, vertex reordering, noise, mesh simplification, deformation, quantization, and smoothing. The metric of BER (Bit Error Rates) ([Kuo et al. 2009; Wang and Hu 2009]), is used to evaluate the robustness.

6.2.1 Robustness against Cropping. Since watermark pieces are embedded into mutual independent embedding-cells and we embed multiple copies of the same watermark into the model, the watermark can survive much more serious cropping attacks than other existing watermarking methods. In our experiments, the cropping ratio is measured by counting the percentage of cut vertices with respect to the total number of vertices in the original model. A few cropped Bunny models are shown in Fig. 5 which demonstrates that, even when the cropping ratio goes as high as 92.11%, our algorithm can still extract a clearly distinguishable image of the embedded watermark.

6.2.2 Robustness against Vertex Reordering. We iteratively perform the vertex reordering attack for 50 times with various seeds from a random number generator. According to our experiments, the embedded watermark is fully extracted from all the watermarked models. This can be explained by the following facts: (1) In the watermark embedding process, we encode both the index and its corresponding watermark piece into ratios of an embedding-cell, which ensures that we can find the correct order of the watermark pieces. (2) In the watermark extracting process, coplanar quadrilaterals are designated once embedding-cells are chosen, and the embedding-cell selection is completely independent to the order of vertices in a local neighborhood.

6.2.3 Robustness against Affine Transformations. Experimental tests of the robustness against affine transformations of the proposed method are shown in Fig. 10. In these tests, the parameters for various affine transformations are set as the following: (a) R: rotation by 45° around X-axis and Y-axis, respectively; (b) NS: scaling along X-axis, Y-axis and Z-axis with a factor 0.5, 1.5 and 2.0, respectively; (c) RE: reflection relative to the YOZ plane; (d) S: scaling along X-axis, Y-axis and Z-axis with a factor 0.5, respectively; (e) SH: shearing along X-axis. The shear level is $s = 1.5$ and the shear formula is $X' = X + s * Y$; and (f) CB: combination of the above five transformations in turn. Since the

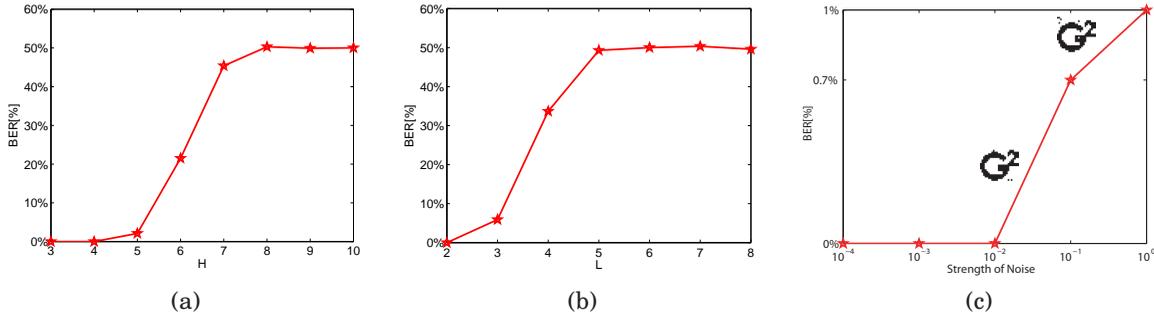


Fig. 11. The robustness against noise is related to the position of H and L (the strength of noise is 10^{-5}): (a) The variation of BERs at different H (with $L = 2$). (b) The variation of BERs at different L (with $H = L + 1$). The BERs for different strengths of noise when 50% vertices are attacked - (c).

Table II. BERs of the Extracted Watermark from Different Models at Different Noise Ratios and Simplification Rates

Model	Noise	SNR _n	BER _N	CR _N	Sim.	BER _S	CR _S
Fandisk	10 ⁻⁵	53.90	12.62%	67%	10%	0.00%	31%
	10 ⁻⁴	51.01	30.11%	17%	20%	0.00%	32%
	10 ⁻³	48.45	49.03%	3%	30%	4.03%	34%
Dragon	10 ⁻⁵	50.86	0.00%	81%	10%	0.00%	30%
	10 ⁻⁴	48.32	16.31%	24%	20%	0.00%	21%
	10 ⁻³	45.81	39.77%	3%	30%	17.00%	10%
Buddha	10 ⁻⁵	52.18	0.00%	64%	10%	0.00%	39%
	10 ⁻⁴	48.91	15.48%	18%	20%	0.00%	31%
	10 ⁻³	45.76	38.25%	2%	30%	3.40%	24%
Bunny	10 ⁻⁵	52.44	0.00%	70%	10%	0.00%	36%
	10 ⁻⁴	50.32	13.46%	30%	20%	0.00%	29%
	10 ⁻³	46.79	38.37%	1%	30%	3.51%	20%
					40%	19.23%	7%

“Sim.” is an abbreviation for “Simplification”; “CR_N” means the ratio of the detected embedding-cells right match the original cells of the unattacked model to the number of all detected embedding-cells during the detection process of the attacked model by noise. Likewise, “CR_S” denotes for the damaged model by simplification.

proposed algorithm employs affine invariants, the extracted watermark is preserved under any affine transformations involving translation, rotation, uniform scaling, non-uniform scaling, shear transformations, reflection, and their combinations.

6.2.4 Robustness against Noise. To evaluate the resistance to noise attack, we add a random number uniformly distributed in the interval of $[-a, a]$ to each vertex coordinate, where a is the strength of noise. We carry out a series of tests on the Bunny model with $a = 10^{-5}$ to observe the relationship between the robustness against noise and the embedding positions H and L . As shown in Fig. 11(a), the robustness against noise decreases as H increases. When H increases to 6, there is a dramatic decrease in the performance. When H equals to 7, the robustness against noise is very weak, and the extracted watermark turns out to be a random binary string with a BER around 50%. In Fig. 11(b), we keep the distance between H and L constant and observe that the robustness against noise decreases as L increases.

According to Fig. 11, we should keep the values of H and L as small as possible in order to achieve strong robustness against noise. In our tests, we set $H = 3$ for the Dragon, Buddha and Bunny models and $H = 4$ for the Fandisk model. Experimental results for the four test models under three different noise strengths are given in Table II. From the table we can see that our method is fairly robust to noise attacks under a strength of 10^{-4} , but the quality of the embedded watermark starts to devastate for a noise strength of 10^{-3} or higher. Simultaneously, the ratio of the correct embedding-cells detected to the whole decreases sharply. One point should be clear that although some embedding-cells detected correctly, the information extracted from them could be wrong. This is because decimal values indicated by L in affine invariant length ratios are affected by vertex displacement caused by additive noise. This means that the proposed approach is slightly sensitive to noise attack. Note that, the performance for the Fandisk model is not as good as that for the other models, since H is set to be 4 for the Fandisk model, which is more sensitive to noise.

6.2.5 Robustness against Local Attacks. To evaluate the robustness against local attacks, we use an example of the robustness against noise when it is applied to 50% of the watermarked model, and show the results for the Bunny model in (c) of Fig. 11. The coordinate axis represent the same as Table. V. Two extracted watermark images are illustrated in (c) of Fig. 11 when the noise strengths are 10^{-1} and 10^0 , respectively. The results show that the extracted watermark is still close to the original even when the noise strength is as high as 10^0 . The robustness against local attacks of our algorithm is very strong for the reason that watermark pieces are embedded in neighborhoods that are mutually independent. Therefore, damages applied locally in a small range will not affect the recovery of the embedded watermark which is performed throughout the inputted model.

6.2.6 Robustness against Simplification Attacks. Using the QSlim algorithm [Garland and Heckbert 1997], we evaluated the robustness against simplification attacks. Table II shows the results for the four models under four different simplification rates. Thanks to the repetition of the watermark embedded throughout the model, our approach demonstrates the robustness against simplification when the ratio under 30% (i.e., sufficient unaltered vertices in the simplified model). If the simplification factor is larger than 30%, our approach can weakly withstand this attack. The explanation for noise attacks can be used for this case.

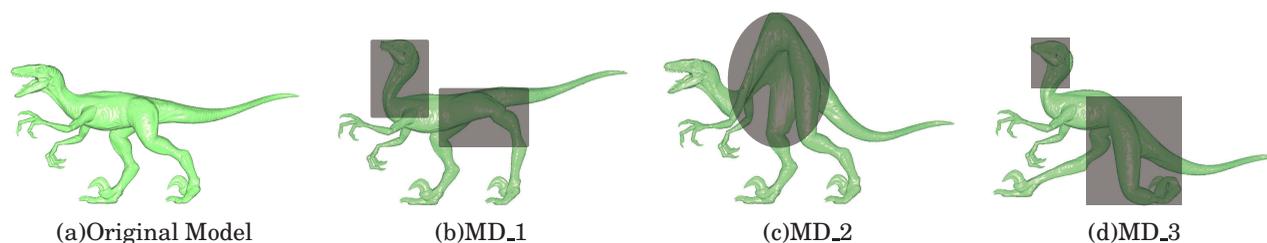


Fig. 12. The Raptor models suffered from the mesh deformation attacks.

6.2.7 Robustness against Mesh Deformation. In our test for robustness against mesh deformation, the deformation algorithm proposed by Au et al. [2006] is adopted to deform two models, namely, Dinosaur and Raptor. For Dinosaur, we just set the feet as its constraint vertices, which renders the watermark pieces may not be able to retrieve the whole watermark as examined in Table III. While we set five constrain places on the Raptor to limit the changes in local parts of the model (see Fig. 12), our approach can obtain super robustness as shown in Table III. In short, our approach can resist

mesh deformation as long as there are unaltered regions or there are retained coplanar quadrilaterals during the mesh deformation.

Table III. BERs of the Extracted Watermark from Different Models after Random Mesh Deformations

Model	Deform.	BER_D	CR_D
Dino-saur	MD_1	0.00%	21%
	MD_2	0.00%	22%
	MD_3	2.18%	12%
Raptor	MD_1	0.00%	18%
	MD_2	0.00%	19%
	MD_3	0.00%	16%

“MD_1”, “MD_2”, and “MD_3” represent for three random model deformations, respectively.

Table IV. BERs of the Extracted Watermark from Models at Different Quantization Rates and Smoothing Strengths

Model	Quan.	BER_Q	CR_Q	Smoothing	BER_S	CR_S
Dragon	16	14.08%	60%	10^{-4}	0.00%	30%
	15	25.14%	14%	10^{-3}	13.37%	17%
	14	39.77%	2.6%	10^{-2}	35.62%	14%
Bunny	16	10.38%	51%	10^{-4}	0.00%	32%
	15	18.64%	10%	10^{-3}	14.13%	15%
	14	37.82%	2.5%	10^{-2}	34.54%	2.1%

6.2.8 Robustness against Quantization and Smoothing. To evaluate the robustness against uniform quantization attacks, three different quantization rates are applied to Dragon and Bunny models. Each coordinate of vertices is represented with 16, 15, and 14 bits. As shown in Table IV, our approach can be fairly resistant up to 15 bit quantization. Similar to the case of noise, the proposed method has weaker robustness as the quantization step size increases. Table IV also shows the performance of the watermarking scheme after smoothing attacks [Taubin 1995]. Three different smoothing strengths are applied to the vertices (only one iteration). The statistics shows that our approach is robust until the smoothing strength is greater than 10^{-2} . In sum, as long as the coordinates of vertices which are members of coplanar quadrilaterals, modified by the quantization or smoothing attacks, do not threaten the positions of H and L in the affine invariant ratios, our watermarking scheme is robust against such attacks. When the disturbance incurred by the attacks seriously affects the digits among H and L of the ratios, the resistance to quantization and smoothing would be lost.

6.3 Comparison with Existing Watermarking Methods

Table V shows the theoretical comparison of our approach with the most related approaches [Cho et al. 2007; Zafeiriou et al. 2005; Praun et al. 1999; Cayre et al. 2004; Lee and Kwon 2007; Wang et al. 2009; Konstantinides et al. 2009; Lin et al. 2010; Yu et al. 2003; Chao et al. 2009; Bogomjakov et al. 2008] including the watermarking schemes and the high capacity steganography methods. “Limited” listed in the “Model Generality” row indicates that the application of the corresponding approach has limitations on 3D models to be watermarked (i.e. manifold models only). None of these watermarking schemes [Cho et al. 2007; Zafeiriou et al. 2005; Praun et al. 1999; Cayre et al. 2004; Lee and Kwon 2007; Wang et al. 2009; Konstantinides et al. 2009; Lin et al. 2010; Yu et al. 2003] can withstand affine transformations since the watermark extraction will fail when the watermarked model changed in a non-uniform or shear way. Also, most of them are not robust against mesh deformation, cropping and serious local attacks (all the imaginable attacks) since these attacks cause severe alterations to the principal object axis, the mass center or the intrinsic shape of the 3D model. Compared to their robustness performances against these attacks, our approach achieves competitively robustness, since the watermark is embedded into the affine-transformation invariants that are locally constructed and repetitively inserted into all the vertices of the 3D model. With respect to the capacity metric, our approach also achieves a significantly high capacity that is comparable to that of the steganography approaches [Chao et al. 2009; Bogomjakov et al. 2008].

Table V. Comparisons between our approach and existing watermarking approaches

Evaluation	[Cho et al. 2007]	[Zafei. et al. 2005]	[Praun et al. 1999]	[Lee & Kwon 2007]	[Wang et al. 2009]	[Konst. et al. 2009]	[Lin et al. 2010]	[Yu et al. 2003]	Our method	[Cayre et al. 2004]	[Chao et al. 2009]	[Bogo. et al. 2008]
Domain	S.	S.	F.	S.	F.	F.	F.	S.	S.	S.	S.	S.
Decoding scheme	I	I	III	II	I	I	II	III	II	I	I	III
Model Generality	L.	L.	G.	L.	L.	G.	L.	L.	G.	L.	L.	G.
Domain	S.	S.	F.	S.	F.	F.	F.	S.	S.	S.	S.	S.
Decoding scheme	I	I	III	II	I	I	II	III	II	I	I	III
Model Generality	L.	L.	G.	L.	L.	G.	L.	L.	G.	L.	L.	G.
Flexibility(Ca.&In.)	×	×	×	×	×	√	×	△	√	√	√	√
Capacity	⊥	⊥	⊥	⊥	⊥		⊥	⊥	⊥	⊥	⊥	⊥
Similarity-Transformation	√	√	√	△	△	√	√	△	√	△	△	√
Affine-Transformation	×	×	×	×	×	×	×	×	√	×	×	√
Mesh Deformation	×	×	×	×	×	×	△	×	△	×	×	√
Noise	√	△	△	√	√	√	△	√	△	×	×	√
Local-Attacks	△	△	△	△	△	△	△	△	√	×	×	×
Quantization	√	△	△	√	△	√	△	√	△	×	×	√
Smoothing	√	△	△	△	√	√	△	√	△	×	×	√
Simplification	√	△	△	√	√	√	△	√	△	×	×	×
Cropping	×	×	×	△	×	×	△	√	√	×	×	×
File attacks	√	√	√	√	×	√	√	√	√	×	×	×

“S.”, “F.”, “I”, “II”, “III”, “L.”, “G.”, “⊥”, “|”, “△”, and “Ca. & In. ” are the abbreviations of “Spatial”, “Frequency”, “Blind”, “Semi-blind”, “Non-blind”, “Limited”, “General”, “Small”, “Medium”, “Large” and “Capacity&Invisibility”, respectively; symbols “×”, “△”, and “√” indicate that this approach cannot withstand, can weakly withstand, or can strongly withstand attacks, respectively.

6.4 Security

Finally, we briefly discuss the security issue in this section. Based on the classification of attacks to watermarking system introduced in [Perez-Freire et al. 2006], we consider attacks based on watermark key estimation as the watermarking security issue concerned. As stated in Section 4.1, a parameter-dependent key is used as a controller to the embedding and detecting strengths to generate the watermark. This can protect the watermarking channel in a certain degree. Further considering the security of the proposed method, we can improve the security level by doing follows: Randomness in choosing embedding-cells from the constructed coplanar quadrilaterals in the embedding procedure based on a secret key to the encoded embedding-cells. This means that the embedding-cells can be randomly chosen to embed a watermark piece or not. The secret key, which can be expressed as a string of 0 and 1, is corresponding to the set of all embedding-cells. For example, we can only encode those embedding-cells corresponding to 1s. This means that they can be randomly chosen to embed a watermark piece or not. In this way, it needs a search space of $2^{\Omega_{\mathcal{L}}}$, where $\Omega_{\mathcal{L}}$ is the number of encoded embedding-cells, for the adversary to removal, detection (estimation), and modification the embedded watermark.

To cope with collusion attack, due to the topological complexity and irregularity of 3D meshes, the extraction phase of a 3D watermarking scheme should be able to resist vertex reordering attacks. To prevent a team of colluders from getting a correct averaged model from multiple watermarked models, thanks to the robustness against vertex reordering of our approach, a trivial way is to scramble the order of vertices of the mesh model without introducing any geometry changes before the watermark embedding procedure of our proposed method. However, colluders can re-order vertices or register different versions of their obtained models. In this case, we should resort to more complex operations, such as, pre-warping of 3D models. We could use the warping functions proposed in [Francesca Ucheddu and Barni 2008] to transform vertices before watermarking them so as to interfere with the potential re-ordering procedure or registration procedure.

7. CONCLUSIONS AND DISCUSSION

In this paper, we propose a novel, robust, affine-transformation-invariant watermarking approach that embeds/extracts watermarks semi-blindly for 3D mesh models with arbitrary topological connectivities. The proposed method embeds a watermark into an inputted model's geometry by imperceptibly modifying its affine invariant length ratios. It not only provides good watermark invisibility, but also is robust against various attacks including cropping, reordering, affine transformations, noise, simplification, and mesh deformation. Other distinctions of our method include that the computational complexity is as low as linear ($O(V)$), and that the embedding capacity can be tuned to satisfy various watermarking application requirements. Due to its simplicity and efficiency, our proposed method is suitable for general copyright protection of 3D models.

There are several directions to extend the current approach. First, in our current implementation, embedding-cells are used without any classification. We could improve the robustness against global noise by choosing embedding-cells with longer diagonals. Other interesting directions of future exploration include watermarking 3D models of continuous LODs (Levels of Details) and watermarking gigantic models.

REFERENCES

- AGARWAL, P. AND PRABHAKARAN, B. 2007. Robust blind watermarking mechanism for point sampled geometry. In *MM&Sec '07: Proceedings of the 9th workshop on Multimedia & security*. 175–186.
- AIGER, D., MITRA, N. J., AND COHEN-OR, D. 2008. 4-points congruent sets for robust surface registration. In *Siggraph '08*.
- AU, O. K.-C., TAI, C.-L., LIU, L., AND FU, H. 2006. Dual laplacian editing for meshes. *IEEE Transactions on Visualization and Computer Graphics* 12, 3, 386–395.
- BENEDENS, O. 2000. Affine invariant watermarks for 3D polygonal and nurbs based models. In *Proceedings of the Third International Workshop on Information Security*. ISW '00. Springer-Verlag, London, UK, 15–29.
- BENEDENS, O. AND BUSCH, C. 2000. Towards blind detection of robust watermarks in polygonal models. *Computer Graphics Forum* 19, 3, C199–C208.
- BOGOMJAKOV, A., GOTSMAN, C., AND ISENBURG, M. 2008. Distortion-free steganography for polygonal meshes. *Computer Graphics Forum* 27, 2, 637–642.
- CAYRE, F., DEVILLERS, O., SCHMITT, F., AND MAÎTRE, H. 2004. Watermarking 3D triangle meshed for authentication and integrity. INRIA research report RR-5223. June.
- CAYRE, F., RONDAO-ALFACE, P., SCHMITT, F., MACQ, B., AND MATÎRE, H. 2003. Application of spectral decomposition to compression and watermarking of 3D triangle mesh geometry. *Signal Processing: Image Communication* 18, 4, 309–319.
- CHAO, M.-W., LIN, C.-H., YU, C.-W., AND LEE, T.-Y. 2009. A high capacity 3D steganography algorithm. *IEEE Transactions on Visualization and Computer Graphics* 15, 2, 274–284.
- CHENG, S.-C. AND WU, T.-L. 2005. Subpixel edge detection of color images by principal axis analysis and moment-preserving principle. *Pattern Recognition* 38, 4, 527–537.
- CHO, J.-W., PROST, R., AND JUNG, H.-Y. 2007. An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. *IEEE Trans. Signal Processing* 55, 1, 142–155.
- CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. 1998. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum* 17, 2, 167–174.
- COX, I. J., KILIAN, J., LEIGHTON, F. T., AND SHAMMOON, T. 1997. Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Processing* 6, 12, 1673–1687.
- EFIMOV, N. V. 1980. *Higher Geometry*. Mir Publishers, Moscow.
- FRANCESCA UCCHEDDU, C.-C. J. K. AND BARNI, M. 2008. Anticollusion watermarking of 3d meshes by prewarping. In *Proc. SPIE 6819, 68190S*.
- GAO, X., ZHANG, C., WEI, Y., AND LI, W. 2010. A highly adaptable capacity and invisibility 3D watermarking based on four-points sets. In *MM&Sec '10: Proceedings of the 12th workshop on Multimedia & security*. 137–146.
- GARLAND, M. AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 209–216.
- GUSKOV, I., SWELDENS, W., AND SCHRÖDER, P. 1999. Multiresolution signal processing for meshes. In *Proc. ACM SIGGRAPH '99*. 325–334.

- HOPPE, H. 1996. progressive mesh. In *Proc. ACM SIGGRAPH96*. 99–108.
- HUTTENLOCHER, D. P. 1991. Fast affine point matching: An output-sensitive method. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR '91)*. 263–268.
- KALIVAS, A., TEFAS, A., AND LOANNIS PITAS. 2003. Watermarking of 3D models using principal component analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Vol. 5. 676–679.
- KONSTANTINIDES, J. M., MADEMLIS, A., DARAS, P., MITKAS, P. A., AND STRINTZIS, M. G. 2009. Blind robust 3-D mesh watermarking based on oblate spheroidal harmonics. *IEEE Trans. Multimedia* 11, 1, 23–38.
- KUO, C.-T., CHENG, S.-C., WU, D.-C., AND CHANG, C.-C. 2009. A blind robust watermarking scheme for 3D triangular mesh models using 3d edge vertex detection. *Asian Journal of Health and Information Sciences* 4, 1, 36–63.
- LEE, S.-H. AND KWON, K.-R. 2007. A watermarking for 3D mesh using the patch cegis. *Digital Signal Processing: A Review Journal* 17, 2, 396–413.
- LI, L., ZHANG, D., PAN, Z., SHI, J., ZHOU, K., AND YE, K. 2004. Watermarking 3D mesh by spherical parameterization. *Computers & Graphics* 28, 6, 981–989.
- LIN, C.-H., MIN-WEN-CHAO, LIANG, C.-Y., AND LEE, T.-Y. 2010. A novel semi-blind-and-semi-reversible robust watermarking scheme for 3D polygonal models. *Visual Computer*, 1–11.
- LIU, Y., PRABHAKARAN, B., AND GUO, X. 2008. A robust spectral approach for blind watermarking of manifold surfaces. In *MM&Sec '08: Proceedings of the 10th ACM workshop on Multimedia and security*. 43–52.
- LUO, M. AND BORS, A. G. 2008. Principal component analysis of spectral coefficients for mesh watermarking. In *Proceedings of the International Conference on Image Processing*. 441–444.
- OHBUCHI, R., MASUDA, H., AND AONO, M. 1997. Watermarking three-dimensional polygonal models. In *Proceedings of the Fifth ACM International Conference on Multimedia (Multimedia '97)*. 261–272.
- OHBUCHI, R., MASUDA, H., AND AONO, M. 1998. Watermarking three-dimensional polygonal models through geometric and topological modifications. *IEEE J. Selected Areas in Comm.* 16, 4, 551–560.
- OHBUCHI, R., MUKAIYAMA, A., AND TAKAHASHI, S. 2002. A frequency-domain approach to watermarking 3D shapes. *Computer Graphics Forum* 21, 3, 373–382.
- PEREZ-FREIRE, L., COMESANA, P., TRONCOSO-PASTORIZA, J. R., AND PEREZ-GONZALEZ, F. 2006. Watermarking security: a survey. In *LNCS Transactions on Data Hiding and Multimedia Security*.
- PRAUN, E., HOPPE, H., AND FINKELSTEIN, A. 1999. Robust mesh watermarking. In *Proc. ACM SIGGRAPH '99*. 325–334.
- TAUBIN, G. 1995. A signal processing approach to fair surface design. In *Siggraph '95 Proceedings*. 351–358.
- WAGNER, M. G. 2000. Robust watermarking of polygonal meshes. In *Proceedings of the Geometric Modeling and Processing 2000*. GMP '00. IEEE Computer Society, Washington, DC, USA, 201–.
- WANG, K., LAVOUE, G., DENIS, F., AND BASKURT, A. 2008. A comprehensive survey on three dimensional mesh watermarking. *IEEE Trans. Multimedia* 10, 8, 1513–1527.
- WANG, K., LAVOUE, G., DENIS, F., AND BASKURT, A. 2008. Hierarchical watermarking of semiregular meshes based on wavelet transform. *IEEE Transactions on Information Forensics and Security* 3, 4, 620–634.
- WANG, K., LAVOUE, G., DENIS, F., AND BASKURT, A. 2011. Robust and blind mesh watermarking based on volume moments. *Computer & Graphics* 35, 1, 1–19.
- WANG, K., LUO, M., BORS, A. G., AND DENIS, F. 2009. Blind and robust mesh watermarking using manifold harmonics. In *Proceedings of the IEEE International Conference on Image Processing*. 3657–3660.
- WANG, Y.-P. AND HU, S.-M. 2009. A new watermarking method for 3D models based on integral invariants. *IEEE Trans. Visualization and Computer Graphics* 15, 2, 285–294.
- YIN, K., PAN, Z., SHI, J., AND ZHANG, D. 2001. Robust mesh watermarking based on multiresolution processing. *Computers & Graphics* 25, 3, 409–420.
- YU, Z., IP, H. H. S., AND KWOK, L. F. 2003. A robust watermarking scheme for 3D triangular mesh models. *Pattern Recognition* 36, 11, 2603–2614.
- ZAFEIRIOU, S., TEFAS, A., AND LOANNIS PITAS. 2005. Blind robust watermarking schemes for copyright protection of 3D mesh objects. *IEEE Trans. Visualization and Computer Graphics* 11, 5, 596–607.

Received August 2011; revised November 2011; accepted January 2012