Spectral Watermarking for Parameterized Surfaces

Yang Liu, Balakrishnan Prabhakaran, Xiaohu Guo

***Abstract*—This paper presents a blind spectral 2-way watermarking framework for 3D models with parametric information. We introduce a spectral geometric watermarking technique based on Dirichlet Manifold Harmonic Transform to alter the geometric shape, while the spectral basis functions are computed from the parametric mesh as the analysis domain. This new geometric method embeds watermarks into small surface patches without introducing discontinuity across the patch boundary, while at the same time be robust against various spatial attacks. By manipulating part of the geometric shape on intermediate model instead of the original model, this method gains robustness against connectivity changing and cropping attacks. By combining the new geometric method with existing texture method into the 2-way watermarking framework, we can withstand various attacks applied to either geometric mesh or parametric information. Theoretical analysis and experiments show that this new geometric method is robust against the majority of attacks and the 2-way watermarking framework helps achieve better robustness.**

***Index Terms*—Dirichlet Manifold Harmonics, Spectral Watermarking, 3D Models, Parameterized Surfaces**

I. Introduction

Watermarking provides copyright protection and ownership assertion by embedding information into the host data [7,20]. A watermarking system consists of two modules: the embedding module that inserts the information into the data, and the extraction/detection module that checks whether a given piece of data hosts a watermark. The presence of the watermark verifies the copyright.

In applications such as computer animations and video games, most 3D models are used with texture to provide better visual result and performance with fewer vertices. Such models must be parameterized. Parameterization is the process of mapping a surface onto regions of the plane. In the case of triangular mesh surface, it refers to a correspondence between the discrete triangulated surface patch embedded in $\mathbb{R}^3$ and a homeomorphic planar mesh in $\mathbb{R}^2$ built through a piece-wise linear map. A surface coming with parametric information is referred as *Parameterized Surface*. Creating and editing parameterized 3D models are labor-intensive. Watermarking is a possible way to protect these assets.

Parameterized 3D models represented as triangular mesh surfaces contain both geometric and parametric information. As shown in figure 1, a parameterized 3D triangular mesh with texture consists of the following data:

1) **Geometric Mesh:** the set of triangles in 3D space, denoted as G.
2) **Texture Image:** the 2D image representing texture information, denoted as I. Note that 3D model with multiple texture images could be handled as several models with one texture image in each model.
3) **Parametric Mesh:** the set of 2D triangles, denoted as T. It is also referred as texture mesh. Note that it should be homeomorphic to G.
4) **Parametric Mapping:** correspondence between G and T, denoted as $\mathcal{M} : T \to G$.

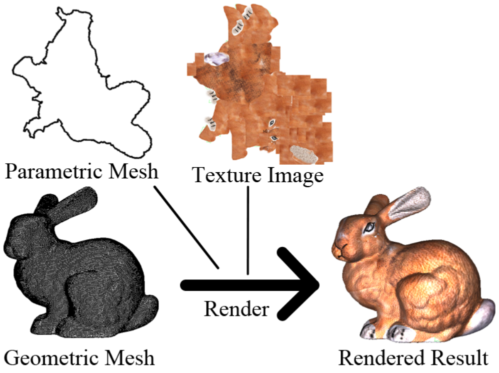


Fig. 1. A 3D triangular mesh with texture consists of geometric mesh G, texture image I, parametric mesh T (only the contour is visualized), and triangle mapping $\mathcal{M}$ (not visualized).

Related works on digital watermarking are listed in section II. The majority of existing watermarking methods for 3D models [25] embed the watermarks by altering the geometric mesh G, while some researchers propose to manipulate the texture image I with some image watermarking techniques. Those geometric watermarking methods can be roughly classified as spatial methods and spectral methods.

Spatial methods modify directly the geometric properties, in most cases the coordinates of vertices or connectivity. They are simple, easy to implement and efficient. They also tend to have larger embedding capacity. However, they are vulnerable to spatial attacks such as adding noise attack because these attacks change the positions of vertices or connectivity. Such method also tends to be vulnerable against connectivity changing attacks such as simplification and subdivision of the 3D model because they will have difficulty finding the corresponding vertices and their neighbors. They also tend to have more visual distortion because human eyes are very sensitive to the discontinuity such as the distortion introduced by modifying a single vertex.

Spectral methods, on the other hand, alter the geometry based on the spread-spectrum of 3D models. The triangular mesh has to be transformed into a spectral domain, then the coefficients corresponding to the perceptually salient basis functions will be modulated with watermarks. Thus the in-

Y. Liu, B. Prabhakaran and X. Guo are with the Department of Computer Science, University of Texas at Dallas, Richardson, TX, 75080 USA e-mail: {yxl072100, bprabhakaran, xguo}@utdallas.edu

EDICS: WAT-APPL, WAT-OTHA, WAT-OTHM

troduced distortion will be distributed globally, and they tend to have low visual distortion. Drawbacks of spectral methods include high computational cost of spectral transformation. In spectral analysis, the signal is represented as a function defined over certain domain, such as the 1D or 2D domain in Fourier analysis. These domains are referred as the *Analysis Domain* in this paper. One popular analysis domain for 3D models is the combinatorial graph consisting of vertices and edges of the mesh, and the eigenfunction of the graph Laplacian matrix can be used as basis function for spectral transformation [1]. Spectral watermarking methods tend to be robust against spatial attacks because distortion cause by local modification is distributed over the whole spectrum. However they are vulnerable against cropping and connectivity changing attacks since these attacks may disturb the analysis domain and lead to unstable basis functions.

To address these vulnerabilities of spectral method while taking its advantage, in this work we propose to manipulate the geometric shape in spectral manner using an intermediate model. Experiments show that this technique improves the robustness of spectral method against such attacks.

A. Motivation

The difficulties of watermarking 3D models include:

1) **Being blind:** Rotation, translation and uniform scaling attacks could easily change the geometric coordinates of vertices without changing the shape of the model. Registration is needed in many cases to help the watermarking method resist such kind of attacks. This requires information of the original model and makes it hard to build a blind method.
2) **Being robust against connectivity changing attacks:** 3D models could have different sampling and/or connectivity without changing their shapes. For spatial watermarking methods, such attacks could change the neighborhood connectivity of vertices and make it difficult to find the vertices holding the hidden information. For spectral watermarking methods, such attacks could disturb the analysis domain so that the computed basis functions become unstable.
3) **Being robust against pose changing attack:** It is commonly used in applications like skeleton-driven character animation and games. The shape of the model could be changed to give different poses of the character. However, the majority of local (small-scale) geometric details are purely undergoing isometric deformation. Many previous watermarking methods have not considered such kind of attacks.
4) **Being robust against cropping attack:** It removes a small part of the model. For spectral methods, cropping attack can change the analysis domain and lead to a totally different set of basis functions. For spatial methods, it will also make it difficult to find the correct vertices for hidden information.
5) **Being robust against adding noise attack:** It disturbs the shape of the model which makes it challenging for spatial methods.

By embedding watermarks in the spectral domain, existing spectral methods tend to be robust against spatial attack such as noise-adding attack. However, many of them require surface registration/alignment between the original and the given surfaces. That is, it is hard for them to be blind. They also tend to be vulnerable against connectivity changing and cropping attacks which may disturb the spectral analysis domain and lead to non-reproducible basis functions.

B. Our Earlier Work

To establish a blind watermarking method that is robust against connectivity changing and cropping attacks as well as noise-adding and local modification attacks, we conducted several research works [14, 15]. By using spectral analysis technique called Manifold Harmonics [24], we did achieve good robustness against local modification and noise-adding attacks [14]. Because 3D models are not defined over regular domain like the 2D rectangular domain of digital images, there is no canonical analysis domain for spectral analysis of 3D models. In the Manifold Harmonics method [14] the geometric mesh G is used as both signal and analysis domain. Because the watermark embedding process also disturbs the analysis domain, it suffers from unstable embedding process. This phenomena is also referred as “casualty” by Wang et al. [28]. Besides unstable embedding process, it also leads to very low capacity. To solve these problems, parameterization information was selected for embedding while the geometric shape itself was used as analysis domain in our later work [15]. The capacity was improved and the embedding process was reliable. But it is still fragile against cropping attack which may potentially change the analysis domain.

C. This Work

In this work, we present a novel 2-way watermarking framework including a new geometric watermarking method. The new geometric method embeds watermark by manipulating the geometric shape of G. The parametric mesh T is used as analysis domain to compute the basis functions to avoid instability in our earlier work [14]. The parametric mesh T is cut into multiple patches, in order to withstand potential cropping attack. Each patch is transformed by the Dirichlet Manifold Harmonic Transformation (D-MHT) [16] which will keep values on the open patch boundary intact no matter how we embed watermarks into the surface patch. This could avoid discontinuity on the surface and related visual distortion. This new geometric method works for both closed and open manifold surface as long as it has valid parameterization. It is robust against various attacks such as affine transformation attacks and pose changing attacks. With this new geometric method and our previous texture method that manipulates T [15], the new 2-way framework provides even better robustness. We perform experiments with more 3D models and attacks to show the robustness of the new method.

D. Watermarking Framework

The new method proposed in this paper, referred as the *geometric method*, uses T as analysis domain. So it is vulnerable against attacks that disturb the analysis domain such

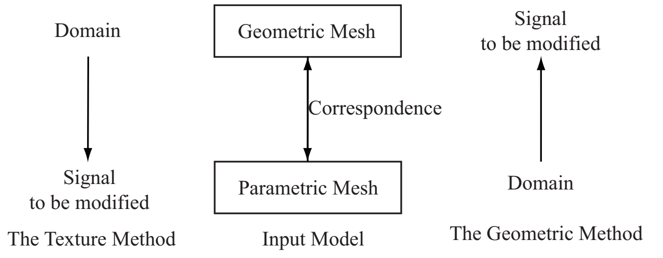


Fig. 2. Difference between the Geometric Method and the Texture Method: they switch the roles of domain and signal.

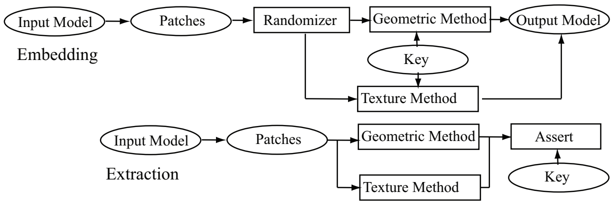


Fig. 3. The 2-way watermarking framework.

as non-uniform affine transformation attack on T. At the same time, our previous method [15], referred as the *texture method*, manipulates the parametric information while using geometric mesh G as analysis domain. It is vulnerable against attacks on G that disturbs its analysis domain. To overcome their vulnerabilities, we present a 2-way watermark framework based on these two methods.

One obvious difference between the new geometric method and the texture method is: the geometric method manipulates the shape as functions defined on T, while the texture method manipulates the parametric information as functions defined on G, as shown in figure 2.

The texture method chose the Manifold Harmonics Transform (MHT) [24] as spectral analysis tool. One drawback of manipulating data using MHT is the continuity across boundary is not preserved. The discontinuity across modified area boundary or patch boundary does not matter in the texture method – any visual distortion introduced by manipulating parametric information could be eliminated by Texture Image Compensation [15]. When modifying the geometry, any discontinuity across the boundary of modified area or patches will cause severe visual distortion. To avoid such phenomena, it is necessary to ensure that the boundary is intact. Unlike the existing texture method which chose MHT, the new geometric method chooses Dirichlet Manifold Harmonic Transform (D-MHT) [16] as analysis tool to satisfy this requirement.

These two methods share a common feature: they both handle patches of the input model separately. So in this paper, we present a 2-way framework based on these two methods. As shown in figure 3, each patch of the input model is processed by ONLY one of these 2 methods. It is recommended to assign the patch to these methods using a randomizer so the attacker could not predict which method we use. The same key bits (watermarks) are embedded into each patch.

In the extraction module, each patch of the model will be extracted using both methods. The extracted bits then are compared with the watermark. If any extracted sequence is similar enough with watermark sequence (i.e., more correct bits than threshold specified by the user), the ownership is claimed.

E. Contributions

The contributions of this paper include:

1) A 2-way blind watermarking framework for 3D models with parametric information. This framework combines the geometric watermarking method and the texture watermarking method together. In case of attacks, any survived method could verify the ownership. So this framework could help to provide better robustness.
2) A spectral geometric watermarking technique based on Dirichlet Manifold Harmonic Transform to alter the geometric shape, while the basis functions are computed using the parametric mesh T as analysis domain. This new geometric method embeds watermarks into small surface patches without introducing discontinuity across the patch boundary, while at the same time be robust against various spatial attacks including pose changing attack. By manipulating part of the shape on intermediate model instead of original model, this method is robust against connectivity changing and cropping attacks.
3) Experiments and theoretical analysis show that this new geometric method and the 2-way framework are robust against a variety of attacks applied to either geometric mesh or parametric information.

The rest of this paper is organized as following: Section II introduces some related works. In section III the new geometric watermarking method is explained in details. In section IV the texture method which is our previous work is introduced and compared with the new geometric method. In section V we analyzed the robustness and capacity of the new geometric watermarking method, and show that the 2-way watermarking framework could help to achieve better robustness. Finally we conclude this paper in section VI with potential future works.

II. Related Works

According to the applications, existing watermarking methods can be classified as: (1) content authentication and tamper proofing [29]; and (2) copyright protection. Yeo and Yeung's work [31] provides more information about these two categories.

Besides, watermarking methods can also be classified into another two categories according to the detection procedure:

- blind watermarking methods, and
- non-blind watermarking methods.

For blind watermarking methods, only the private key is needed for watermark extraction. There are many existing blind watermarking methods [2, 3, 8, 10, 11, 17, 18, 28, 32, 33]. For non-blind watermarking methods, additional information about the original object is needed for watermarking extraction [12, 30]. Although the information about the original object makes the watermarking extraction much easier, it limits the potential application as well.

Most existing methods for 3D models manipulate geometric information for embedding, while some methods manipulate texture image information [10]. Geometric watermarking methods for 3D meshes could be roughly classified as spatial methods [2, 3, 5, 8, 9, 11, 18, 32, 33] and spectral methods [1, 6, 12, 19, 30]. Spectral methods tend to have better robustness while spatial methods tend to have better capacity. The methods based on multi-resolution analysis tool like wavelet [23] are typically considered as spectral methods. According to [25], multi-resolution analysis approaches have either connectivity restrictions or robustness deficiencies, especially vulnerable to connectivity changing attacks such as re-meshing and mesh simplification.

Robustness is essential to any watermarking scenario. To make watermarking methods more robust against various attacks, Feng et al. [9] proposed to embed the watermark using two methods that do not disturb each other at the same time. This is different from our 2-way framework which handle different patches using different methods.

Our method employs spectral analysis tool for 3D models represented by triangular mesh. Zhang et al. [34] compared existing spectral analysis tools for 3D models. The spectral analysis tool used in this paper is called Dirichlet Manifold Harmonics Transform (D-MHT) [16], which was originally proposed for 3D mesh compression. It is closely related to Manifold Harmonics [24]. Both tools are based on eigenfunctions of Laplace-Beltrami Operator (LBO) [22] discretized using Finite Element Method [21].

III. Dirichlet Spectral Geometric Watermarking for Parameterized Surfaces

In this new geometric method, all input models are handled as 3 geometric coordinate functions $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ defined over parametric domain $T \subset \mathbb{R}^2$. Thus the problem comes to embedding information by modifying these 3 functions. It is assumed that all input models represent 2-manifold surfaces (with or without boundary) only. Here 2-manifold means that for each point on the surface there should be a small neighborhood that is homeomorphic to an open disk. Each input model is assumed to have only one connected parametric mesh T in the parametric space $\mathbb{R}^2$. T is assumed to be single connected, otherwise the input model could be handled as several models, each with single connected T. For parametric vertex $\forall vt_i \in T$, $X(vt_i) = x_i$ (Y and Z resp.). The coordinates on $\mathbb{R}^2$ are denoted as u, v.

The scheme of the new geometric method is shown in figure 4.

As a watermarking method, the new method needs to be blind because non-blind methods have only limited application. So it needs to locate the specific elements containing key information without any information of the original model. To achieve this goal, the step of *Preprocessing* is designed. This step also help this method to be immune to uniform affine transformation attacks on parametric mesh.

3D models are difficult to handle because the same geometry may be represented with very different sampling. Re-ordering attack and connectivity-changing attacks such as subdivision and simplification could change the model data with very minor distortion on the geometry shape. This makes them very difficult to address. To handle this problem, it is necessary to separate the geometric shape from the discrete triangular mesh that represents it. Unlike existing works, the new method handles the geometric shape without caring about the discrete representation by capturing and manipulating using an intermediate model. A special step *Cutting Analysis Domain and Re-sampling* is proposed to do this job.

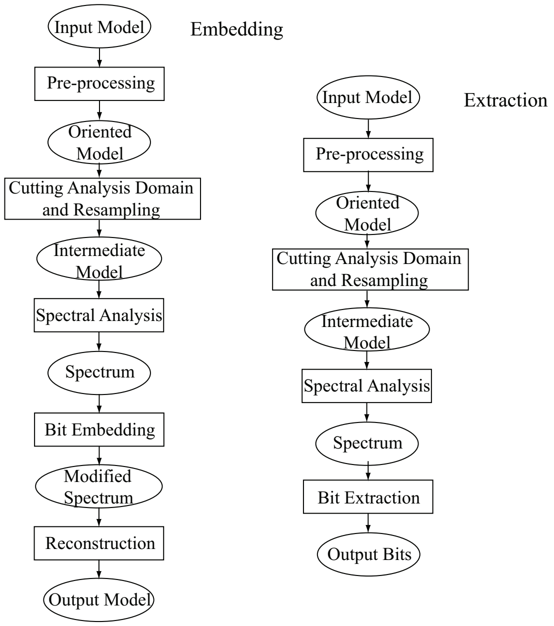


Fig. 4. The watermark embedding and extraction scheme of the geometric method.

Adding noise and smoothing are common attacks to remove watermark. Spatial watermarking methods tend to be fragile to them because they disturb the geometric information of every elements. Even when there is no intentional attacks, the embedded watermark still got challenges. Uniform affine transformation on the geometric shape is a common operation in real applications such as video games. In such cases, the geometric shape is intact. Pose changing is a little bit different. In these cases, the geometric shape of the 3D model undergoes large-scale deformation with details remain the same.

Besides being robust against attacks, being invisible is also essential for watermarking method. It does not make sense to destroy the data to protect it. For watermarking method of 3D models, minimizing visual distortion is necessary.

To address these requirements, we chose Dirichlet Manifold Harmonics Transform [16] in the step of *Spectral Analysis* to manipulate the geometric shape in spectral manner. It helps to distribute the perturbation of embedding over a large group of vertices rather than several neighboring elements. This helps to address spatial attacks and reduce visual distortion. Together with the step of *Bit Embedding/Extraction*, we make the new method immune to uniform affine transformation on the geometric shape, robust against spatial attacks such as adding noise and smoothing, and pose-changing attack.

Details about the steps shown in Figure 4 are presented in sub-sections below. This method could be adjusted by the user using several parameters:

1) Re-sampling density d_u, d_v in the step of Cutting Analysis Domain and Re-sampling. Increasing these parameters means more vertices in the intermediate model M_Q. This helps M_Q capture the geometric shape better for later processing. Drawback is more calculation cost.
2) Cutting Factor c_u, c_v in the step of Cutting Analysis Domain and Re-sampling. These parameters control the size of the regions to be manipulated. Larger factors mean larger area for each region. This leads to better capacity of each region and better robustness against spatial attacks such as adding noise. Drawback is being more vulnerable against cropping attack. Possible combinations of c_u, c_v are determined by the user beforehand and selected for each model. Successful extraction with any possible combination would claim ownership.
3) Embedding Offset d_{off} in the steps of Bit Embedding and Extraction. This parameter is used to avoid modification of large features on the surface. Larger d_{off} means lower distortion on large features. However this leads to less robustness against adding noise attack.
4) Embedding Factor f_t in the step of Bit Embedding. This parameter is used to balance between distortion and robustness against attacks such as adding noise. Larger f_t improves the robustness but introduces more distortion.

The parameters are part of the key for both embedding and extraction process except f_t.

A. Preprocessing

For watermarking method, it is necessary to locate the elements that contain the key information. Comparing with geometric mesh G which may have complicated topology, the parametric mesh T is guaranteed to be flat and non-overlapping. So we choose T as clue for this job.

In applications like computer animation, rigid transform on T is commonly used for various purposes. The step of Preprocessing is designed to make this method robust against such kind of attack on T. This is established by rotating T to a certain orientation at the beginning of both embedding and extraction process. In other words, T is always re-oriented no matter how the attacker rotates it. Effects of arbitrary rotation transformation attack applied on T is then eliminated. In our method, the longest distance between points in T is used as reference:

1) Find the pair $vt_m, vt'_m \in T$ so that $d(vt_m, vt'_m) = \max_{vt\in T, vt'\in T} d(vt, vt')$. Here $d(\cdot,\cdot)$ denotes the Euclidean distance. It could be verified that for such pair both vt_m and vt'_m must locate at boundary.
2) Find a rotation $\mathcal{R}: \mathbb{R}^2 \to \mathbb{R}^2$ such that the line connecting $\mathcal{R}(vt_m)$ and $\mathcal{R}(vt'_m)$ aligns with the v axis. Denote the parametric mesh after rotation as $T_R = \mathcal{R}(T)$.

As shown in figure 5, T is rotated to get a stable orientation. Denote the model after preprocessing as M_R. Please note that T always have open boundary when there is valid parameterization. This does not imply that G has open boundary.

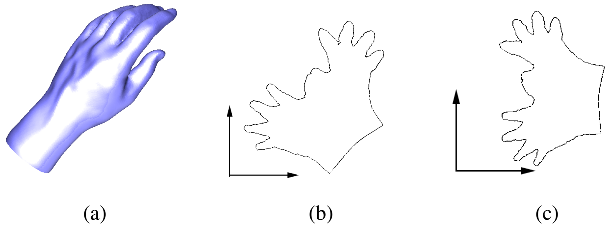


Fig. 5. Preprocessing: (a) the input model; (b) the parametric mesh T; and (c) the oriented parametric mesh T_R.

B. Cutting Analysis Domain and Re-sampling

The purpose of this step is to make the new method robust against connectivity changing attack and tolerate cropping attack. This is achieved by capturing the shape of the surface with an intermediate model which has generated regular connectivity so the following steps will not be aware of the original connectivity information neither the attacked connectivity information. As shown in figure 6, the parametric mesh T will be cut into several pieces for further processing, so that removing a small part of the surface will not destroy all areas containing the embedded information.

Please note that the operations performed in this step are used ONLY to modify the geometric shape. The original model M is NOT required to have any open boundary. M is NOT cut either. The watermarked model M_w always has identical connectivity and topology as M.

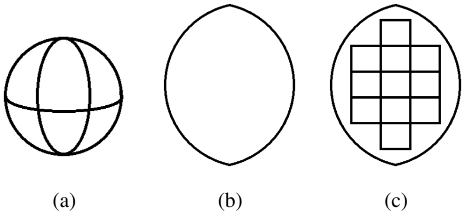


Fig. 6. Cutting analysis domain: (a) the geometric mesh G, (b) the parametric mesh T_R, and (c) cutting T and tiling of key-embedding regions.

There are several parameters the user could adjust: Cutting Factor $0 < c_u, c_v < 1$, and Re-sampling Density: $d_u, d_v \in \mathbb{N}$. Then the following operations are performed:

1) Building bounding box of T_R in parametric space $\mathbb{R}^2$. Denote the center of the bounding box as c_b, the width as w_b and height as h_b.
2) Building a rectangle area centered at c_b with width of $c_u w_b$ and height of $c_v h_b$. Then tile rectangle of the same size as shown in figure 6. Any rectangle region that completely falls inside T is selected for embedding.
3) Re-sampling each selected region Q using a regular grid with resolution of $d_u \times d_v$. Denote the new regular grid as $T_Q \subset \mathbb{R}^2$.
4) Building geometric coordinate functions X_Q, Y_Q and Z_Q for T_Q by linearly interpolating the original X, Y and Z. By combining them together with T_Q, the

intermediate 3D surface patch M_Q is created for each selected region.

C. Spectral Analysis

The purpose of this step is to analyze the shape of the intermediate model M_Q. As the result, a group of spectral descriptors are created for operations in the step of bit embedding and extraction. By manipulating the shape in the spectral domain, the distortion is distributed over all vertices of M_Q. This makes the visual distortion negligible. Due to the property of spectral analysis, disturbance of spatial attacks such as adding noise and smoothing will be distributed over all spectral descriptors as well. This helps to make this method robust against these attacks. Because modifying the shape of G_Q does not disturb the parametric mesh T_Q which is used as analysis domain, this method does NOT suffer instable embedding process problem as previous method [14].

In this step the shape of the intermediate model M_Q is represented as coordinate functions $\mathbf{X}_Q$, $\mathbf{Y}_Q$ and $\mathbf{Z}_Q$ defined over T_Q. There are requirements for the spectral analysis tool chosen in this step:

1) The chosen tool could be performed on 2-manifold surfaces with open boundary represented as triangular mesh. In our case, it is the parametric mesh T_Q.
2) The boundary value of coordinate function after embedding $\mathbf{X}_{Q,w}$ should have the same boundary value as $\mathbf{X}_Q$ ($\mathbf{Y}_Q$, $\mathbf{Y}_Q$ resp.).
   This is necessary because we are manipulating the shape of selected region Q using M_Q. If $M_{Q,w}$ does not have identical boundary coordinates as M_Q, there may be discontinuity along the boundary between the modified region and other regions on the watermarked model M_w. This may cause severe visual distortion which is not acceptable.

To satisfy the requirements, we choose a spectral analysis tool called Dirichlet Manifold Harmonics Transform (D-MHT) [16] to transform $\mathbf{X}_Q$, $\mathbf{Y}_Q$ and $\mathbf{Z}_Q$ of M_Q for further processing.

On the 2-manifold surface represented as triangular mesh (in our case, the parametric mesh T), the discrete Laplacian L could be constructed as a matrix. Dirichlet Manifold Harmonic Basis (D-MHB) $\{H^j\}$, $(j = 1, \cdots, m)$ [16] can be computed as the Dirichlet eigenvectors of L which is intrinsic to the 3D geometric shape of the surface. D-MHB could be used to analyze any scalar function defined over the manifold. In this method T_Q is considered as a flat 2-manifold surface embedded in $\mathbb{R}^3$ and used as the analysis domain to calculate D-MHB. $\mathbf{X}_Q$, $\mathbf{Y}_Q$ and $\mathbf{Z}_Q$ are then considered as scalar functions defined over T_Q. The spectral descriptor $[\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_m]^T$ of function $\mathbf{X}_Q$ can be computed by the Dirichlet Manifold Harmonic Transform (D-MHT) [16]:

$$\mathtt{x}_d = \mathtt{x} - \mathtt{x}_h, \tag{1}$$

$$\tilde{x}_j = \mathtt{x}_d^T D H^j = \sum_{i=1}^{|G|} x_i D_{i,i} H_i^j, \tag{2}$$

where $\mathtt{x}$ denotes $[x_1, x_2, \ldots, x_{|G|}]^T$ which is the vector form of $\mathbf{X}_Q$, $\mathtt{x}_h$ is the harmonic function having the same boundary value as $\mathtt{x}$, and D is the “mass” matrix encoding the weight of each vertex. Here $\mathtt{x}_h$ being harmonic function means $L\mathtt{x}_h(v_i) = 0$ for non-boundary vertex v_i and $\mathtt{x}_h(v_i) = \mathtt{x}(v_i)$ for boundary vertex v_i. $\mathtt{x}_h$ is also referred as the *harmonic component* of $\mathtt{x}$. $\mathtt{x}_d$ is referred as the *Dirichlet component*. The descriptor of $\mathbf{Y}_Q$ and $\mathbf{Z}_Q$ can be computed in the same way. The inverse D-MHT (I-DMHT) can be used to transform the spectral descriptors back to the geometric coordinate functions:

$$x_i = \sum_{j=1}^{m} \tilde{x}_j H_i^j + x_{h,i}. \tag{3}$$

D. Bit Embedding/Extraction

Our bit embedding/extraction strategy is designed to make this method robust against the rotation attack on G, translation attack on G and uniform scaling attack on both T and G. In case of these attacks, the spectral descriptors of the shape also undergo certain kinds of transformation. But the watermark information will not be erased. Theoretical analysis about this is presented in section V-C.

In this step the watermark key bits are embedded by manipulating spectral descriptors of geometric coordinate functions using **Neighbor Couple Embedding (NCE)** [14]. Given the spectral descriptors $\{\tilde{x}_i\}$, $\{\tilde{y}_i\}$, and $\{\tilde{z}_i\}$ of the geometric coordinate functions $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$, NCE needs the integer parameter of Embedding Offset $d_{off} > 0$ and the float parameter of Tolerance Factor $0 < f_t < 1$ to embed key bit sequence k.

Our rotation-invariant spectrum is defined as $\{e_i = \sqrt{\tilde{x}_i^2 + \tilde{y}_i^2 + \tilde{z}_i^2}\}$. In section V-C we show that $\{e_i\}$ is invariant to arbitrary rotation attacks on G. Low-frequency components satisfying $i < d_{off}$ will NOT be used for watermark embedding because they correspond to the large-scale features of G and may introduce more severe distortions if modified. Middle-frequency components satisfying $d_{off} \leq i < d_{off} + 2|k|$ are selected and divided into $|k|$ groups of 2 adjacent components $\{g_i = \{e_{d_{off}+2i}, e_{d_{off}+2i+1}\}\}, i = 0, \cdots, |k| - 1$.

In section V-C we show that $\{e_i\}$ will also experience uniform scaling under uniform scaling attacks on T and/or G. For better robustness against such kind of attacks, the bits of watermarks are embedded by modifying the members of each group $g_i = \{e_{i,0}, e_{i,1}\}$ such that:

$$\begin{cases} e'_{i,0} \leq e'_{i,1} \cdot (1 + f_t) & \text{when } k[i] = 0 \\ e'_{i,0} \geq e'_{i,1} \cdot (1 - f_t) & \text{when } k[i] = 1 \end{cases} \tag{4}$$

The float parameter f_t is used to balance robustness and distortion. Note that the distortion could not be arbitrary small. That is, $f_t = 0$ also introduces distortion. The modified spectral descriptors are computed as $\tilde{x}'_i = \tilde{x}_i \frac{e'_i}{e_i}$, $\tilde{y}'_i = \tilde{y}_i \frac{e'_i}{e_i}$ and $\tilde{z}'_i = \tilde{z}_i \frac{e'_i}{e_i}$.

The watermark extraction process computes $\{e_i\}$ and gets $\{g_i\}$ similarly. Bits are extracted as:

$$\begin{cases} k'[i] = 0 & \text{when } e_{i,0} < e_{i,1} \\ k'[i] = 1 & \text{when } e_{i,0} > e_{i,1} \end{cases} \tag{5}$$

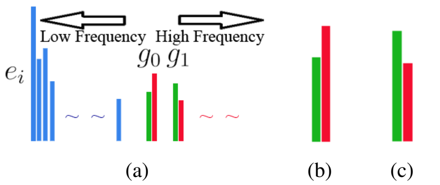


Fig. 7. NCE: (a) dividing $\{e_i\}$ into small groups after omitting low-frequency components; (b) a group with bit 0; and (c) a group with bit 1.

E. Reconstruction

This step intends to build the watermarked model M_w according to the modified spectral descriptors got from the previous step of bit embedding. Here only the shapes of selected regions of the input model M are modified accordingly. The watermarked model M_w has identical connectivity and parametric information as M. Please note that M and M_w are NOT required to have open boundary.

The reconstruction contains following operations:

1) Re-create modified coordinate functions $X_{Q,w}$, $Y_{Q,w}$ and $Z_{Q,w}$ for the intermediate model M_Q using I-DMHT. Denote the watermarked intermediate model as $M_{Q,w}$.
2) Build watermarked coordinate functions X_w, Y_w, and Z_w on M_R by linearly interpolating $X_{Q,w}$, $Y_{Q,w}$, and $Z_{Q,w}$, using the overlaying relationship between T_R and T_Q.
3) Build watermarked mesh M_w using the original model M and X_w, Y_w, and Z_w.

IV. The Texture Method

The texture method here refers to our previous work [15]. It is also a spectral watermarking method. The primary idea is, on the contrary of the new geometric method, to manipulate parametric coordinate functions U and V defined over G.

The texture method employs Manifold Harmonics Transform (MHT) as spectral analysis tool. Both MHT and D-MHT employ eigenfunctions of discretized Laplacian operator L as basis functions. The difference is MHT employs Neumann boundary condition while D-MHT employs Dirichlet boundary condition.

The advantage of D-MHT compared with MHT is it always guarantees the accuracy of the boundary value. This property could help to avoid discontinuity across analysis domain boundary and related visual distortion in case of embedding watermarks by manipulating only part of the shape. In the texture method, keeping continuity near boundary is not important because a step called Texture Image Compensation [15] is used to eliminate possible visual distortion by altering I according to the modified parametric information. Thus both D-MHT and MHT can work for the texture method. Because D-MHT requires more computational resource while making no difference to the watermarking result, we still use MHT (with Texture Image Compensation) for the texture method.

The advantage of the texture method includes robustness against various attacks on T and having no visual distortion. Because the texture method employs G as the analysis domain directly, it is vulnerable against local modification attack on G and cropping attack that disturb G. By combining the new geometric method with existing texture method into the 2-way watermarking framework shown in figure 3, we can withstand various attacks applied to either geometric mesh G or parametric mesh T. Please note that in the framework each patch of the input model is ONLY embedded using one of these 2 methods.

V. Evaluations

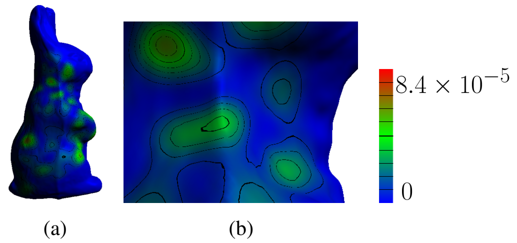


Fig. 8. (a) the rabbit model rendered with distortion (Hausdorff Distance) of the geometric method; (b) close look at the same model. The diagonal length of the bounding box is 1.82.

In this section, we use both theoretical analysis and experimental results to show that the new geometric method and the 2-way framework are robust against various attacks. As shown in figure 8, the distortion introduced by the geometric method is minor. Distortion data is shown in Table IV. We choose Hausdorff distance, root mean square error (RMS) and mesh structural distortion measure (MSDM) [27] as distortion metrics. For verification purpose, only the embedding area Q located at the center of T is selected for watermark embedding in the following experiments because all possible regions are handled in the same way. The new method is implemented using Python programming language and Matlab. All experiments are performed on a Windows desktop PC with 2.66Ghz CPU and 4GB memory. It takes about 3 minutes for both embedding and extraction. Most time is spent on creating basis functions for spectral analysis because solving an eigen problem is necessary. Other operations cost only seconds.

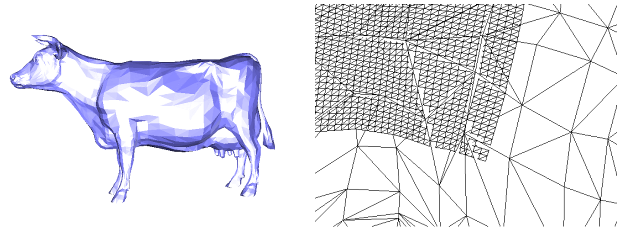
Using theoretical analysis, we show that the new geometric method is immune to re-ordering and similarity attack. As suggested by Wang et al. [27], we perform adding noise, smoothing, quantization, subdivision, simplification and cropping attacks on various models. Attack data is shown in Table V. "N/A" in Table V means the process of Cutting Analysis Domain and Sampling failed to find a valid region Q for extraction. This is caused by cropping attack. Different "cases" stand for different random-generated attack configuration. "Rounds" stand for iteratively performed attacks. Note that the bunny model is handled as bunny patch 1 and bunny patch 2 because it has 2 separate parametric mesh. Most models listed in Table V are closed surfaces without boundary. Besides, we also perform pose changing attack which is common operation in real applications. The cow model has the worst performance because it does not have enough degree of freedom as explained in section V-A.

A. Embedding Capacity Analysis

In this method, we choose to embed each key bit by manipulating one pair of adjacent spectral descriptors. The reason is that it could provide better robustness against noise attack on G comparing with embedding more bits into fewer descriptors, as disturbance of noise on spectrum tends to be spreading. As a result, more spectral descriptors are needed.

It is easy to see that to embed $|k|$ bits, $d_{off} + 2|k|$ MHB $\{H^i\}$ are needed. That is, the intermediate model M_Q must have at least $d_{off} + 2|k|$ non-boundary vertices. The embedding capacity could be increased by simply increasing the re-sampling density d_u, d_v. Here we recommend using d_u, d_v large enough so that the edges of M_Q is shorter than the shortest edge of M within the patch Q. This is to make sure that M_Q captures the shape of patch Q sufficiently.

But increasing d_u, d_v could NOT increase the capacity unlimitedly. In our method the original connectivity of the 3D model is not modified. So the cardinality of vertex set of M_w will be the same as that of M. The shape of the selected region Q could not retain very tiny detail no matter how much detail $M_{Q,w}$ could have. In other words, Q may not have enough degrees of freedom (DOF) to carry much information. Figure 9 shows such a case of the Cow model. The Cow model has too few vertices and can not have very detailed shape. That's why it has the worst performance as shown in table V. We tried different parameters to embed watermarks into the rabbit model shown in figure 5. Table I shows the influence of the parameters c_u and c_v – higher values means larger selected region Q and more vertices to manipulate. As we can see from the table, more bits get lost when the selected region gets smaller. This may be improved by increasing the size of selected region to contain more vertices to manipulate. Drawbacks include making the watermark less robust against cropping attacks, since larger embedding regions are more likely to be affected and lead to less number of available regions for embedding.



(a) (b)

Fig. 9. The Cow Model: (a): the Cow Model (3,097 vertices); (b): the Cow Model and the intermediate model enlarged. The surface can not have detailed shape due to limited number of vertices.

TABLE I
INFLUENCE OF c_u, c_v ON THE RABBIT MODEL (100 BITS EMBEDDED)

c_u, c_v	Bits Extracted Successfully	Affected Vertices in Q
0.05	70	266
0.10	98	1034
0.15	100	2339
0.25	100	23945

We also tried to embed more key bits without enlarging Q on the Rabbit model. As shown in table II, more and more bits are lost as $|k|$ increases. The reason of this phenomena is that embedding more bits means modifying more high-frequency spectral descriptors of the geometric coordinate functions. That means there will be more high-frequency manipulation on the shape of $M_{Q,w}$. Recall that in the reconstruction step the shape of $M_{Q,w}$ is applied back to the original model by linear interpolation. During the extraction process the shape is interpolated again to capture the shape of selected region Q. These operations of linear interpolation always introduce noises. Such noises tend to be local, and contribute to the high-frequency part of the shape. So the bit error rate may increase when more bits are embedded by modifying high-frequency spectral descriptors. Using larger parameter for bit embedding factor f_t could improve. As shown in table III, larger f_t does help to lower the bit error rate, with barely more distortion. The visual difference between the watermarked models is not distinguishable for human eyes. Note the capacity of a model is determined by the number of vertices. When there are enough vertices, in our experiments all bits embedded with a proper parameter of f_t could be extracted correctly without attack. We recommend using f_t around 0.2. Note that the distortion is restricted inside the selected region Q because modification is applied to the shape of Q only.

TABLE II
INFLUENCE OF $|k|$ ON THE RABBIT MODEL

$\|k\|$	Bits Extracted Successfully
100	100
200	190
300	278

TABLE III
INFLUENCE OF f_t ON THE RABBIT MODEL (100 BITS EMBEDDED)

f_t	RMS	MSDS	Hausdorff Distance	Bounding Box Diagonal
0.05	0.00005	0.034757	0.00096	1.823454
0.10	0.000054	0.035702	0.001041	
0.15	0.000058	0.037280	0.001135	
0.25	0.000069	0.041420	0.001357	

Based on our experiments, we would recommend that the number of embedded key bits should be less than 1/10 of the number of affected vertices in region Q.

B. Re-ordering Attack

Because the step of Re-sampling could eliminate any possible effect of re-ordering attack on the intermediate model M_Q, this method is naturally immune to such kind of attacks.

C. Invariance Under Similarity Attack

Here we show the effect of uniform affine transformation attack and why this method is immune to them by theoretical analysis.

Denote the geometric coordinate functions to be analyzed as $\mathtt{x}$, $\mathtt{y}$, and $\mathtt{z}$ in column-vector form. Suppose there are n internal vertices and m boundary vertices for the texture patch under consideration. Then $\mathtt{x}$, $\mathtt{y}$ and $\mathtt{z}$ are all $(n + m)$-vectors.

Rotation on G: Suppose there is a pure rotation transformation applied on G. Denote the transformation matrix as $\mathcal{R}$, which is a 3×3 real matrix with $\det(\mathcal{R}) = 1$ and

TABLE IV
DISTORTION INTRODUCED (100 BITS EMBEDDED)

	f_t	c_u, c_v	Bounding Box Diagonal	Hausdorff Distance	RMS	MSDM
Bunny Patch 1	0.25	0.18	2.127146	0.002877	0.000258	0.081628
Bunny Patch 1	0.25	0.18	1.989403	0.004499	0.000446	0.082736
Cow	0.6	0.42	2.033784	0.0018612	0.002057	0.171333
Dragon	0.35	0.2	2.364309	0.006543	0.000318	0.118555
Hand	0.25	0.3	2.157906	0.002922	0.000303	0.059712
Rabbit	0.25	0.45	1.824128	0.008309	0.0001147	0.123487

$\mathcal{R}\mathcal{R}^T = \mathcal{R}^T\mathcal{R} = I$. It's easy to see that the column-vectors of coordinate functions $\mathtt{x}$, $\mathtt{y}$, and $\mathtt{z}$ undergo the same rotation as well. For coordinate (x, y, z), the coordinate after rotation is $\begin{bmatrix} x' & y' & z' \end{bmatrix}^T = \mathcal{R} \begin{bmatrix} x & y & z \end{bmatrix}^T$.

In D-MHT, geometric coordinate functions are decomposed into harmonic component and Dirichlet component. That is, $\begin{bmatrix} \mathtt{x} & \mathtt{y} & \mathtt{z} \end{bmatrix} = \begin{bmatrix} \mathtt{x}_h & \mathtt{y}_h & \mathtt{z}_h \end{bmatrix} + \begin{bmatrix} \mathtt{x}_d & \mathtt{y}_d & \mathtt{z}_d \end{bmatrix}$ holds.

The discrete Laplacian operator L is a $(n+m) \times (n+m)$ real matrix. Denote it as

$$L = \begin{bmatrix} L_{n,n+m} \\ L_{m,n+m} \end{bmatrix} \tag{6}$$

where $L_{n,n+m}$ is a $n \times (n+m)$ sub-matrix and $L_{m,n+m}$ is a $m \times (n+m)$ sub-matrix of L. Then we have $L_{n,n+m} \begin{bmatrix} \mathtt{x}_h, \mathtt{x}_h, \mathtt{x}_h \end{bmatrix} = 0_{n,3}$.

By applying the rotation $\mathcal{R}$ on the harmonic components we get $\begin{bmatrix} \mathtt{x}'_h & \mathtt{y}'_h & \mathtt{z}'_h \end{bmatrix} = (\mathcal{R} \begin{bmatrix} \mathtt{x} & \mathtt{y} & \mathtt{z} \end{bmatrix}^T)^T = \begin{bmatrix} \mathtt{x}_h & \mathtt{y}_h & \mathtt{z}_h \end{bmatrix} \mathcal{R}^T$. It is obvious that $L_{n,n+m} \begin{bmatrix} \mathtt{x}'_h & \mathtt{y}'_h & \mathtt{z}'_h \end{bmatrix} = 0_{n,3}$ holds.

Thus we know that $\mathtt{x}'_h$, $\mathtt{y}'_h$, and $\mathtt{z}'_h$ are harmonic components of $\mathtt{x}'$, $\mathtt{y}'$, and $\mathtt{z}'$. That is, harmonic components of rotated geometric coordinate functions are just rotated harmonic components. Therefore, $\mathtt{x}'_d$, $\mathtt{y}'_d$, and $\mathtt{z}'_d$ defined as $(\mathtt{x}'_d, \mathtt{y}'_d, \mathtt{z}'_d)^T = \mathcal{R}(\mathtt{x}_d, \mathtt{y}_d, \mathtt{z}_d)^T$ are the Dirichlet component of $\mathtt{x}'$, $\mathtt{y}'$, and $\mathtt{z}'$.

With the definition of D-MHT, it is easy to see that $\{\tilde{x'}_i\}$, $\{\tilde{y'}_i\}$ and $\{\tilde{z'}_i\}$ defined as

$$\begin{bmatrix} \tilde{x'}_1, \dots, \tilde{x'}_n \\ \tilde{y'}_1, \dots, \tilde{y'}_n \\ \tilde{z'}_1, \dots, \tilde{z'}_n \end{bmatrix} = \mathcal{R} \begin{bmatrix} \tilde{x}_1, \dots, \tilde{x}_n \\ \tilde{y}_1, \dots, \tilde{y}_n \\ \tilde{z}_1, \dots, \tilde{z}_n \end{bmatrix} \tag{7}$$

are the D-MHT spectral descriptors of $\mathtt{x}'_d$, $\mathtt{y}'_d$, and $\mathtt{z}'_d$. That is, in the case of rotation, D-MHT spectral descriptors undergoes the same rotation as well. It is easy to see that $e'_i = \sqrt{(\tilde{x}'_i)^2 + (\tilde{y}'_i)^2 + (\tilde{z}'_i)^2} = e_i$. That is, $\{e_i\}$ is rotation-invariant.

Translation on G: A translation on G can be denoted as $\mathrm{Trans}(x, y, z) = (x + t_x, y + t_y, z + t_z)$ where $[t_x, t_y, t_z] \in \mathbb{R}^3$ is the translation vector. It is easy to verify that $\mathtt{x}'_h = \mathtt{x}_h + t_x$ and similar equations hold for $\mathtt{y}'_h$ and $\mathtt{z}'_h$. Thus we know $\mathtt{x}'_d = \mathtt{x}_d$ ($\mathtt{y}$ and $\mathtt{z}$ resp.). So we have $e'_i = e_i$. That is, $\{e_i\}$ is translation-invariant.

Uniform Scaling on G: A pure uniform scaling transformation on G could be denoted as $\mathrm{Scale}(x, y, z) = (cx, cy, cz)$ where $c \in \mathbb{R}$ is the scaling factor. It could be verified that $\mathtt{x}'_h = c\mathtt{x}_h$ and $\mathtt{x}'_d = c\mathtt{x}_d$ hold (resp. $\mathtt{y}$ and $\mathtt{z}$). So we have $\tilde{x}'_i = c\tilde{x}_i$ ($\tilde{y}$ and $\tilde{z}$ resp.). That is, spectral descriptors of the geometric coordinate functions undergo the same scaling transformation. So we have $e'_i = |c|e_i$.

Uniform Scaling on T: In case of uniform scaling on T, it is easy to see that T_Q undergoes uniform scaling as well. Consider the discrete Laplacian operator L which could be decomposed as $L = D^{-1}Q$. According to the definition of D and Q we can see that after the uniform scaling, $Q' = Q$ and $D' = c^2 D$ where $c \in \mathbb{R}$ is the scaling factor. Thus $H'^i = |c|H^i$ for $\forall i$ holds. Then we know $\tilde{x}' = |c|\tilde{x}$ ($\tilde{y}, \tilde{z}$ resp.).

Rotation and Translation on T: In case of rotation on T, due to the preprocessing the same T_Q is ensured. In case of translation on T, T_Q undergoes translation. In both cases, according to the definition, the discrete Laplacian operator L keeps unchanged. That is, D-MHB $\{H^i\}$ is invariant of rotation and translation on T. So are $\{e_i\}$, $\{\tilde{x}_i\}$, $\{\tilde{y}_i\}$, $\{\tilde{z}_i\}$ as well.

We have shown that the new geometric method is robust against uniform affine transformation attacks on both T and G using theoretical analysis. For other kinds of attacks, there is no such theoretical result. So we conduct experiments to verify the robustness against those attacks. We also conduct experiments to show that the proposed 2-way watermarking framework could help to achieve better robustness.

D. Cropping Attack

The experiments of cropping attack are performed using tool provided by Wang *et al* [27]. The parts to be removed are decided randomly. To better evaluate the performance, we perform the attack several times (different cases) with the same cropping percentage. Results are shown in Table V. "N/A" means the step of Cutting Analysis Domain and Re-sampling failed to find a valid region Q for extraction. Different "cases" stand for different configuration of cropping. Figure 10 shows one example of attacked model in which all 100 key bits survived. As described earlier, the shape of the intermediate model M_Q is retrieved using orientation process and bounding box of parametric mesh T. When the orientation process, parametric bounding box and embedding area Q are not disturbed, all embedded key information could be extracted successfully, otherwise they will be lost. The new geometric method is not fragile against cropping attack, though it is relatively weak.

E. Smoothing Attack, Quantization Attack and Noise Attack

These attacks share one common characteristics: they all disturb geometric coordinates of vertices and leave connectivity information untouched. The experiments are performed using the tool provided by Wang *et al* [27]. Results are shown in Table V. "Rounds" of smoothing attack means for how many times the model is smoothed iteratively. Different

TABLE V
SURVIVING BITS UNDER VARIOUS ATTACKS (100 BITS EMBEDDED)

	Intensity	Case#	Bunny Patch 0 18,974 Vertices	Bunny Patch 1 16,578 Vertices	Cow 3,097 Vertices	Dragon 50,582 Vertices	Hand 37,230 Vertices	Rabbit 71,004 Vertices
Noise Attack	0.0005	1	100	100	81	100	100	100
		2	99	100	83	99	100	100
		3	98	99	82	100	100	100
	0.001	1	97	100	78	100	99	100
		2	94	98	80	100	99	100
		3	96	98	82	100	99	100
	0.002	1	86	96	80	100	91	100
		2	92	92	77	95	98	100
		3	90	95	84	97	97	100
	0.003	1	91	90	79	92	91	100
		2	85	90	80	88	94	100
		3	91	82	80	92	94	100
	0.004	1	81	83	79	88	92	100
		2	83	80	81	90	84	100
		3	87	86	73	93	82	99
	Rounds		Bunny Patch 0	Bunny Patch 1	Cow	Dragon	Hand	Rabbit
Smooth Attack	1		100	99	79	100	100	100
	2		98	99	71	96	100	100
	3		93	96	67	89	100	100
	4		91	95	64	86	100	100
	5		89	93	62	82	100	100
	Bits		Bunny Patch 0	Bunny Patch 1	Cow	Dragon	Hand	Rabbit
Quantization Attack	11		100	99	82	100	100	100
	10		94	96	82	99	97	100
	9		90	87	84	97	94	100
	8		76	83	81	87	93	99
	7		74	66	74	74	74	96
	Rounds		Bunny Patch 0	Bunny Patch 1	Cow	Dragon	Hand	Rabbit
Subdivision Attack	1		90	98	63	93	100	100
	2		89	96	60	86	100	100
	Percentage		Bunny Patch 0	Bunny Patch 1	Cow	Dragon	Hand	Rabbit
Simplification Attack	5%		100	100	82	100	98	100
	10%		100	97	81	100	99	100
	15%		98	95	81	100	95	100
	20%		93	88	82	100	90	100
	25%		91	87	80	100	88	100
	30%		88	88	79	100	87	100
	Percentage	Case#	Bunny Patch 0	Bunny Patch 1	Cow	Dragon	Hand	Rabbit
Cropping Attack	1%	1	100	100	82	48	53	61
		2	N/A	100	N/A	100	100	100
		3	100	100	N/A	100	100	100
	2%	1	100	100	N/A	55	100	100
		2	100	100	N/A	47	45	91
		3	100	100	N/A	100	100	100

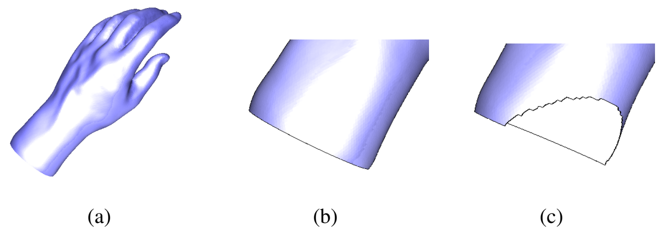
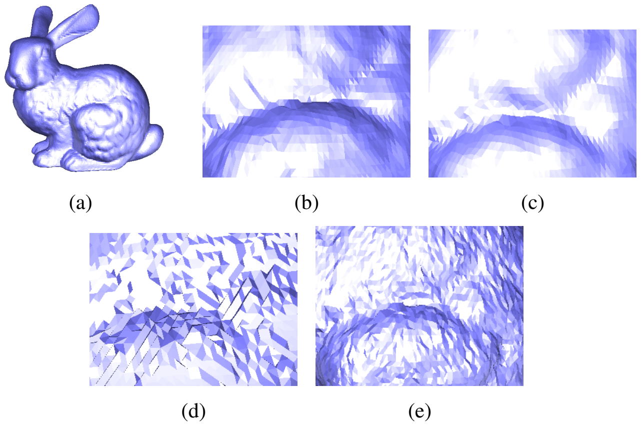

(a) (b) (c)

Fig. 10. Cropping Attack: (a) the hand model; (b) enlarged hand model; and (c) the hand model after 2% cropping attack.


(a) (b) (c)

(d) (e)

Fig. 11. Smooth Attack, Quantization Attack and Noise Attack: (a) the bunny model; (b) the enlarged bunny model; (c) the bunny model after 5 rounds of smooth attack; (d) the bunny model after 7 bits of quantization attack; and (e) the bunny model after 0.04 intensity of noise attack.

“cases” of adding noise attack stand for different random-generated noises. Figure 11 shows the result of smoothing attack on the bunny model. The new geometric method is very robust against smoothing attack, quantization attack and noise attack. This is due to the fact that modification of the shape is distributed over large surface area in a spectral manner.

F. Subdivision Attack and Simplification Attack

Both subdivision attack and simplification attack change the connectivity of the model. Subdivision attack is performed using the Catmull-Clark method [4]. Simplification attack is performed using the Lindstrom-Turk method [13]. In both cases the parametric information is generated using linear interpolation based on edges of the texture patch. To ensure the integrity of the parametric information, edges corresponding to parametric mesh boundary are not removable in case of simplification. The results are shown in Table V, "Rounds" of subdivision attack means for how many times the model is modified iteratively. Figure 12 shows the result of subdivision attack on the dragon model.

This new method is very robust against such kind of connectivity changing attacks, especially when the model contains more vertices. This is because the new method manipulates the shape of the surface rather than the connectivity.

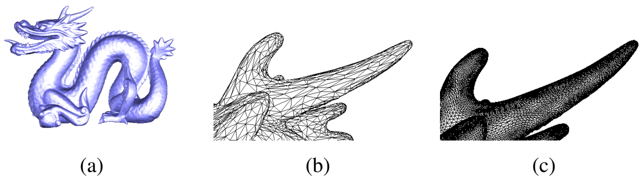


Fig. 12. Subdivision Attack: (a) the dragon model; (b) the enlarged dragon model; and (c) the enlarged model after 2 rounds of subdivision attack.

G. Pose Changing Attack

Pose changing is commonly used in applications like skeleton-driven character animation. The shape of the model could be changed driven by the underlying skeleton structure to give different poses of the character. However, the majority of local geometric details are purely undergoing isometric deformation while the shape undergoes large scale deformation. This experiment shows that our watermarking method can withstand such attacks. The original Armadillo model used here has 7 separate parts. The watermark embedding parameters we used are $d_u = d_v = 200$, $c_u = c_v = 0.25$, $d_{off} = 20$ and $f_t = 0.1$. The extraction result is shown in Table VI and the corresponding poses are shown in Figure 13.

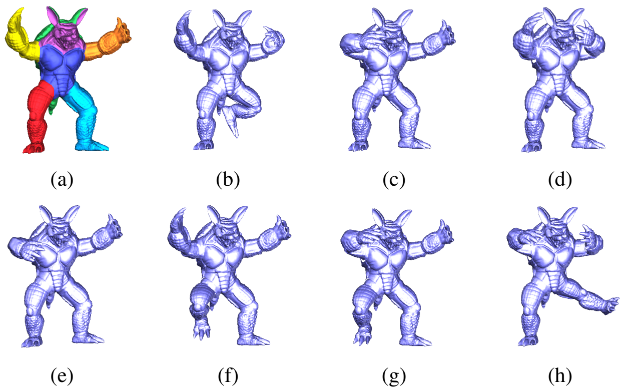


Fig. 13. Pose changing attack: (a): the Armadillo model (7 patches) in rest position; (b)-(h): different poses of the model.

H. Comparison

We compare the geometric method with Wang *et al*'s work [26]. The "N/A" in table VII means the data is not provided by Wang *et al*'s work [26]. Our speculation is that the BER (Bit Error Rate) of 30% Simplification Attack for Wang *et al*'s method should be 0 since the BER of 70% Simplification Attack is 0. As table VII shows, our geometric method provides better capacity while providing comparable robustness against noise attack and quantization attack, low distortion and relatively less robustness against simplification attack. Note that our geometric method provides more capacity with models containing more vertices while Wang *et al*'s method provides the same capacity of 45 to 75 bits.

TABLE VI
SURVIVING BITS UNDER POSE CHANGING ATTACK (100 BITS EMBEDDED PER PATCH)

Pose	Patch #1	#2	#3	# 4	# 5	#6	# 7
pose 2	95	99	99	79	99	78	87
pose 3	95	100	79	100	99	100	87
pose 4	95	100	89	76	99	100	87
pose 5	95	100	80	100	99	100	87
pose 6	95	100	99	100	82	100	87
pose 7	95	100	85	100	87	100	87
pose 8	95	100	85	83	99	87	87

VI. Conclusions

In this work a blind spectral 2-way watermarking framework is presented for 3D models with parametric information. A new spectral geometric technique based on D-MHT is presented to manipulate the geometric shape using parametric mesh T as analysis domain. We embed watermarks into small surface patches without introducing discontinuity across patch boundaries. We manipulate part of the shape on intermediate model M_Q instead of the original mesh M, which helps the new method to be robust against connectivity changing and cropping attacks. Based on this new geometric method and the existing texture method, a 2-way watermarking frame work is presented to achieve better robustness to withstand various attacks applied to either geometric mesh or parametric information. With theoretical analysis and experiments we show that the new geometric method and 2-way framework are robust against a variety of attacks such as uniform affine transformation, quantization, and adding noise attacks.

There are also some limitations that will motivate our future research. The new geometric method interpolates the geometric coordinate functions linearly to capture the shape for the intermediate model M_Q, and then to put the modified shape of $M_{Q,w}$ to the modified model M_w. These two steps of linear interpolation introduce noise always, especially when there are sharp features in the 3D model. The way we manipulate the spectral descriptors may introduce more distortion than expected. The new geometric method is fragile against attacks that disturb the bounding box of the texture patch T because they disturb the process of finding embedding region Q. It requires valid parameterization and parametric mesh patch with enough vertices to manipulate. CAD models tend to have multiple parametric patches, because it is hard to create parameterization with single parametric patch for them. These small patches contain limited number of vertices though the geometric mesh has enough vertices. They are not suitable for embedding. That's why we do not use any model of mechanical part in our experiments. In addition, geometric distortion introduced by watermarking is not acceptable for CAD models. These make the new geometric method not suitable for CAD models. In our future work, we would like to reduce the noise by using interpolation techniques with higher order

TABLE VII
COMPARISON WITH METHOD OF WANG *et al* [26]

	Wang *et al*'s Method [26]		Geometric Method			
	Bunny Model	Dragon Model	Bunny Model (2 patches)	Dragon Model	Hand Model	Rabbit Model
d_{MSDM}	0.19	0.20	0.08	0.118555	0.0597	0.12
d_{RMS}	0.0008	0.0002	0.000258/0.000446	0.000318	0.0003	0.0001147
#Bits	67	49	100 + 100	100		
Attack ↓	Bit Error Rate (BER) ↓					
Noise 0.10%	0.01	0.01	0.028	0	0.01	0
Noise 0.30%	0.07	0.12	0.118	0.093	0.07	0
Noise 0.05%	0.11	0.19	0.2	0.15	0.15	0
Smoothing 10 Rounds ($\lambda = 0.03$)	0.13	0.08	0.06	0.10	0	0
Smoothing 30 Rounds ($\lambda = 0.03$)	0.19	0.24	0.20	0.30	0	0
Smoothing 50 Rounds ($\lambda = 0.03$)	0.37	0.41	0.28	0.32	0.03	0
Quantization 9	0.04	0.02	0.115	0.003	0.006	0
Quantization 8	0.04	0.18	0.205	0.13	0.07	0.01
Quantization 7	0.15	0.39	0.3	0.26	0.26	0.04
Subdivision	0.09	0.01	0.06	0.07	0	0
Simplification 30%	N/A	N/A	0.12	0	0.13	0
Simplification 70%	0	0	0.35	0.23	0.33	0.07

continuity; preserve sharp features by introducing the feature lines into the analysis domain cutting and re-sampling [16]; improve technique of locating embedding region Q; and better technique of manipulating spectral descriptors that introduces less distortion.

ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant No. 1012975 and No. 1149737. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Emad E. Abdallah, A. Ben Hamza, and Prabir Bhattacharya, *Spectral graph-theoretic approach to 3D mesh watermarking*, Proceedings of Graphics Interface, 2007, pp. 327–334.

[2] Patrice Rondao Alface, Benoit Macq, and Francois Cayre, *Blind and robust watermarking of 3D models: How to withstand the cropping attack?*, Proceedings of 2007 IEEE International Conference on Image Processing (ICIP), 2007, pp. 465–468.

[3] Adrian Gheorghe Bors, *Watermarking 3D shapes using local moments*, Proceedings of 2004 IEEE International Conference on Image Processing (ICIP), 2004, pp. 729 –732 Vol.2.

[4] E. Catmull and J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes*, Computer-Aided Design **10** (November 1978), no. 6, 350–355.

[5] Hung-Kuang Chen and Yung-Hung Chen, *Progressive watermarking on 3D meshes*, Proceedings of 2010 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2010, pp. 1–7.

[6] Daniel Cotting, Tim Weyrich, Mark Pauly, and Markus Gross, *Robust watermarking of point-sampled geometry*, Proceedings of the Shape Modeling International, 2004, pp. 233–242.

[7] Ingemar Cox, Matthew Miller, and Jeffrey Bloom, *Digital watermarking: Principles & Practice*, Morgan Kaufmann, 2001.

[8] P. Daras, D. Zarpalas, D. Tzovaras, and M.G. Strintzis, *Watermarking of 3D models for data hiding*, Proceedings of 2004 IEEE International Conference on Image Processing (ICIP), 2004, pp. 47–50.

[9] Xiaoqing Feng, Li li, Zhigen Pan, Shusen Sun, and Daxing Zhang, *A robust double watermarking 3D mesh model based on feature*, Proceedings of 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP), 2008August, pp. 1109–1112.

[10] Emmanuel Garcia and Jean-Luc Dugelay, *Texture-based watermarking of 3D video objects*, IEEE Transactions on Circuits and Systems for Video Technology **13** (2003), no. 8, 853–866.

[11] Thomas Harte and Adrian Bors, *Watermarking 3D models*, Proceedings of 2002 IEEE International Conference on Image Processing (ICIP), 2002, pp. 661–664.

[12] Li Li, David Zhang, Zhigeng Pan, Jiaoying Shi, Kun Zhou, and Kai Ye, *Watermarking 3D mesh by spherical parameterization*, Computers & Graphics **28** (2004), no. 6, 981–989.

[13] Peter Lindstrom and Greg Turk, *Fast and memory efficient polygonal simplification*, IEEE visualization, 1998, pp. 279–286.

[14] Yang Liu, Balakrishnan Prabhakaran, and Xiaohu Guo, *A robust spectral approach for blind watermarking of manifold surfaces*, Proceedings of the 10th ACM Workshop on Multimedia and Security, 2008, pp. 43–52.

[15] ———, *Blind invisible watermarking for 3D meshes with textures*, Proceedings of 2010 IEEE International Conference on Image Processing (ICIP), 2010, pp. 3689 –3692.

[16] ———, *Dirichlet harmonic shape compression with feature preservation for parameterized surfaces*, Computer Graphics Forum **29** (2010), no. 7, 2039–2048.

[17] Ming Luo, Kai Wang, Adrian Bors, and Guillaume Lavoué, *Local patch blind spectral watermarking method for 3D graphics*, Proceedings of Digital Watermarking, 2009, pp. 211–226.

[18] Shinichi Nakazawa, Sho Kasahara, and Shigeo Takahashi, *A visually enhanced approach to watermarking 3D models*, Proceedings of 2010 sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010, pp. 110 –113.

[19] Ryutaro Ohbuchi, Akio Mukaiyama, and Shigeo Takahashi, *A frequency-domain approach to watermarking 3D shapes*, Computer Graphics Forum **21** (2002), no. 3, 373–382.

[20] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn, *Information hiding – A survey*, Proceedings of the IEEE **87** (1999), no. 7, 1062–1078.

[21] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke, *Laplace-Beltrami spectra as "shape-DNA" of surfaces and solids*, Computer-Aided Design **38** (2006), no. 4, 342–366.

[22] Steven Rosenberg, *The Laplacian on a Riemannian manifold: An introduction to analysis on manifolds*, Cambridge University Press, 1997.

[23] F. Uccheddu, M. Corsini, and M. Barni, *Wavelet-based blind watermarking of 3D models*, Proceedings of the ACM Workshop on Multimedia and Security, 2004, pp. 143–154.

[24] Bruno Vallet and Bruno Lévy, *Spectral geometry processing with manifold harmonics*, Computer Graphics Forum **27** (2008), no. 2, 251–260.

[25] Kai Wang, Guillaume Lavouè, Florence Denis, and Atilla Baskurt, *Three-dimensional meshes watermarking: Review and attack-centric investigation*, Proceedings of the International Workshop on Information Hiding, 2007, pp. 50–64.

[26] Kai Wang, Guillaume Lavoué, Florence Denis, and Atilla Baskurt, *Robust and blind mesh watermarking based on volume moments*, Computers & Graphics **35** (2011), no. 1, 1–19.

[27] Kai Wang, Guillaume Lavoué, Florence Denis, Atilla Baskurt, and Xiyan He, *A benchmark for 3D mesh watermarking*, Shape modeling international, June 2010, pp. 231–235 (en).

[28] Kai Wang, Ming Luo, Adrian Bors, and Florence Denis, *Blind and robust mesh watermarking using manifold harmonics*, Proceedings of

2009 IEEE International Conference on Image Processing (ICIP), 2009, pp. 3657–3660.

[29] Yu-Ping Wang and Shi-Min Hu, *A new watermarking method for 3D models based on integral invariants*, IEEE Transactions on Visualization and Computer Graphics **15** (2009), no. 2, 285–294.

[30] Jianhua Wu and Leif Kobbelt, *Efficient spectral watermarking of large meshes with orthogonal basis functions*, The Visual Computer **21** (2005), no. 8–10, 848–857.

[31] Boon-Lock Yeo and Minerva M. Yeung, *Watermarking 3D objects for verification*, IEEE Computer Graphics Application **19** (1999), no. 1, 36–45.

[32] Stefanos Zafeiriou, Anastasios Tefas, and Ioannis Pitas, *A blind robust watermarking scheme for copyright protection of 3D mesh models*, Proceedings of 2004 IEEE International Conference on Image Processing (ICIP), 2004, pp. 1569–1572.

[33] ______, *Blind robust watermarking schemes for copyright protection of 3D mesh objects*, IEEE Transactions on Visualization and Computer Graphics **11** (2005), no. 5, 596–607.

[34] Hao Zhang, Oliver van Kaick, and Ramsay Dyer, *Spectral mesh processing*, Computer Graphics Forum **29** (2009), no. 6, 1865–1894.