BLIND INVISIBLE WATERMARKING FOR 3D MESHES WITH TEXTURES

Yang Liu, Balakrishnan Prabhakaran, and Xiaohu Guo

Department of Computer Science, University of Texas at Dallas

ABSTRACT

We propose to embed watermarks by modifying the texture mapping information of 3D models rather than modifying the geometry information or texture image as existing works do. We present a blind watermarking method based on spectral decomposition that incorporates the process of Texture Image Compensation (TIC) which ensures no visual distortion. We describe a Neighbor Couple Embedding (NCE) scheme that works on the Manifold Harmonics Transform (MHT) of the texture coordinate functions. Experiments show that this method is robust against common attacks such as adding noise attacks, uniform affine transformation attacks, local modification attacks and produces no visual distortion on the rendered 3D models. Our contributions include watermarking the texture mapping information with no visual distortion as well as a novel embedding method that is robust against various possible attacks.

***Index Terms*—** Triangular Mesh, Texture, Watermark

1. INTRODUCTION

Watermarking is a good way to protect the copyright of digital 3D models by embedding information into 3D model data [1, 2]. The presence of the watermark verifies the copyright.

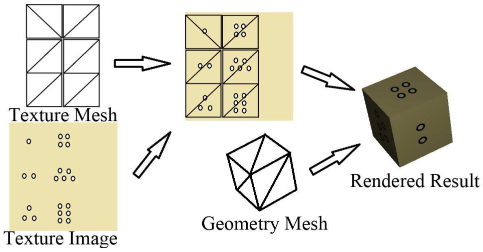

Fig. 1. A 3D triangular mesh with texture consists of geometric mesh G, texture image I, texture mesh T and triangle mapping M(not visualized).

In most applications 3D models come with textures. As shown in figure 1, a 3D triangular mesh with texture consists of following data:

1. **Geometric Mesh:** The set of triangles in 3D space, denoted as G.
2. **Texture Image:** 2D image representing texture information, denoted as I. Note that 3D model with multiple texture images could be handled as several models with one image each.
3. **Texture Mesh:** The set of 2D triangles, denoted as T.
4. **Triangle Mapping:** Mapping from 3D triangles to 2D triangles, denoted as $M : G \to T$.

The majority of existing watermarking methods [3] embed bits by modifying G, while some researchers [4, 5] propose to embed bits into I using existing image watermarking techniques. All these methods introduce visual distortion, more or less. This may be intolerable for some particular applications, e.g. computer-aided design (CAD). Also, watermarks in I are vulnerable to the attacks of simply replacing texture images.

To solve these issues, we propose to embed watermarks by modifying the texture mesh T in 2D, rather than G and I. Our watermarking method will introduce no visual distortion to the rendered 3D models, which is guaranteed by creating I' using the process of Texture Image Compensation (TIC). It is robust against most of the common attacks because it employed a spectral analysis tool called Manifold Harmonics (MH), and watermarks are embedded by modifying the geometric spectrum of T.

2. SPECTRAL WATERMARKING ON TEXTURE MAPPING

The algorithmic flow consists of the following steps:

1. *Preprocessing*: A vertex in G may be mapped to multiple vertices in T since the mapping is based on “triangles”. When any vertex in G is mapped to one vertex in T only, real-valued texture coordinate functions, $\mathbf{u}(G)$ and $\mathbf{v}(G)$, are well-defined on the vertices in G. The existence of $\mathbf{u}$ and $\mathbf{v}$ is required for Bit Embedding and Bit Extraction processes since they manipulate them directly. This is ensured by the preprocessing step.
2. *Bits Embedding*: The 3D model after preprocessing may consist of several disjoint surfaces which are referred as “patches” and handled separately. For each patch, $\mathbf{u}$ and $\mathbf{v}$ are transformed into the spectral domain

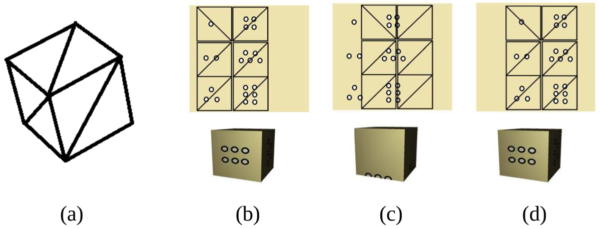
(a) (b) (c) (d)

Fig. 2. With (a) the geometric mesh G, (b) texture mesh T and texture image I of the dice model, we embed watermarks on T to get (c) the modified texture mesh T', then eliminate visual distortion by creating (d) a new texture image I' with Texture Image Compensation (TIC).

using Manifold Harmonics, and modified to embed the bit sequence k using the Neighbor Couple Embedding (NCE) scheme. Besides k, an integer parameter – Embedding Offset d_{off}, and a real parameter – Tolerance Factor f_t are required. They are also parts of the key, although not embedded into the model.

3. *Reconstruction*: The modified patches and untouched patches are put together to produce the watermarked model as output.

4. *Texture Image Compensation*: The modified patches may produce a different rendering result from the original one, since the new texture mesh T' is different from the original T. By creating a new texture image I' correspondingly, we can eliminate any visual distortion when T' satisfies certain requirements. Figure 2 illustrates the idea.

5. *Bit Extraction*: Given patches after preprocessing, $\mathbf{u}$ and $\mathbf{v}$ are transformed into the spectral domain using Manifold Harmonics Transform. With d_{off} and f_t, a sequence of bits k' can be extracted from the spectral descriptors. By matching k' with k, the input model can be authenticated when the number of matching bits exceeds a specific threshold determined by the user.

2.1. Preprocessing

This stage is employed to ensure that $\mathbf{u}$ and $\mathbf{v}$ are well-defined on the geometric mesh G. We cut patches along the vertices in G that are mapped to multiple ones in T. Since the Bit Embedding/Extraction processes manipulate $\mathbf{u}$ and $\mathbf{v}$ directly, the one-to-one mapping between vertices in G and T is required for performing the spectral transformation of $\mathbf{u}$ and $\mathbf{v}$ functions. After the cutting, the disjoint 3D surfaces are referred as "patches", and are handled separately.

The patches that could be selected for bit embedding must satisfy the following requirements:

1. G is a 2-manifold.
2. G contains enough vertices. To be more specific, $|V| \geq d_{off} + 2|k|$ should hold.

Patches that do not satisfy the above assumptions could remain untouched, or they could be cut to meet requirement 1 and re-sampled to meet requirement 2.

2.2. Bit Embedding/Extraction

We employ a spectral analysis tool called Manifold Harmonics (MH) [6] to transform the texture coordinate functions $\mathbf{u}$ and $\mathbf{v}$ into the spectral domain. For any 2-manifold triangular mesh in 3D, a set of orthonormal Manifold Harmonic Basis (MHB) $\{H^j\}$, $(j = 1, \cdots, m)$ can be computed which is intrinsic to the 3D geometric shape of G. The spectral descriptor $[\tilde{u}_1, \tilde{u}_2, \cdots, \tilde{u}_m]^T$ of function $\mathbf{u}$ can be computed by the Manifold Harmonic Transform (MHT):

$$\tilde{u}_j = \mathrm{u}^T D H^j = \sum_{i=1}^{|G|} u_i D_{i,i} H_i^j, \tag{2.1}$$

where u denotes $[u_1, u_2, \ldots, u_{|G|}]^T$ which is the vector form of $\mathbf{u}$, and D is the mass matrix encoding the weight of each vertex. The descriptor of $\mathbf{v}$ can be computed similarly. The inverse MHT (IMHT) can be used to transform the spectral descriptor to the texture coordinate functions:

$$u_i = \sum_{j=1}^{m} \tilde{u}_j H_i^j. \tag{2.2}$$

We employ manifold harmonics in this work, because as a spectral transformation tool it can be computed on 2-manifold surfaces of arbitrary topology, and its basis (MHB) is an intrinsic property of the surface, independent of mesh resolution. Thus it provides better robustness to different types of attacks, especially uniform affine transformation, local modification, and noise-addition attacks.

Neighbor Couple Embedding (NCE):

Given the spectral descriptors $\{\tilde{u}_i\}$ and $\{\tilde{v}_i\}$ of the texture coordinate functions $\mathbf{u}$ and $\mathbf{v}$, NCE will use the integer parameter Embedding Offset $d_{off} > 0$ and float parameter Tolerance Factor $0 < f_t < 1$ to embed key bit sequence k.

To withstand rotational attack on texture mesh, our rotation-invariant spectrum is defined as $\{e_i = \sqrt{\tilde{u}_i^2 + \tilde{v}_i^2}\}$. It could be verified that $\{e_i\}$ is invariant to arbitrary rotation attacks on T and/or G. Low-frequency components satisfying $i < d_{off}$ will NOT be used for watermark embedding because they correspond to the large-scale features of T 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 $\{g_i = \{e_{d_{off}+2i}, e_{d_{off}+2i+1}\}\}, i = 0, \cdots, |k| - 1$.

It could be verified that $\{e_i\}$ will also experience uniform scaling under uniform scaling attacks on T and/or G. For

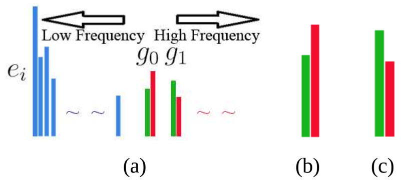

Fig. 3. NCE: (a) Dividing $\{e_i\}$ into small groups after omitting low-frequency components, (b) a group with bit 0 and (c) a group with bit 1.

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:

$$\begin{cases} e'_{i,0} \leq e'_{i,1} \cdot (1 + f_t) & \text{when } k[i] = 0 \\ e'_{i,0} \geq e'_{i,1} \cdot (1 - f_t) & \text{when } k[i] = 1 \end{cases} \tag{2.3}$$

Float parameter f_t is used to balance robustness and distortion. Note that the visual distortion of the reconstructed 3D models will be eliminated by Texture Image Compensation described in section 2.3. The modified descriptors are computed as $\tilde{u}'_i = \tilde{u}_i \frac{e'_i}{e_i}$ and $\tilde{v}'_i = \tilde{v}_i \frac{e'_i}{e_i}$.

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

$$\begin{cases} k'[i] = 0 & \text{when } e_{i,0} < e_{i,1} \\ k'[i] = 1 & \text{when } e_{i,0} > e_{i,1} \end{cases} \tag{2.4}$$

2.3. Texture Image Compensation (TIC)

After watermark embedding, we get a modified texture mesh T' from $\mathbf{u}'$ and $\mathbf{v}'$. To eliminate possible visual distortion the Texture Image Compensation is proposed. The basic idea of TIC is creating a modified texture image I' so that it renders the same result with T' as the original model does. The only prerequisite is that T' contains no overlapping triangle. TIC consists of the following steps:

1. For each triangle $t' \in T'$, find the corresponding $t \in T$ as shown in figure 4.
2. Fill the pixels of I' in t' with colors from the pixels of I in t using linear interpolation.

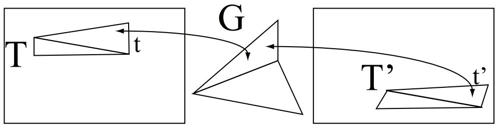

Fig. 4. TIC: mapping triangle $t' \in T'$ to $t \in T$ according to the triangle mapping M.

It could be verified that TIC eliminates all visual distortion, since most graphical rendering use linear interpolation during the rasterization process (see figure 5). Note that it can be implemented with OpenGL API and be accelerated with graphics hardware. This makes TIC very efficient. Users could embed information without worrying about any visual distortion, since it will be eliminated with TIC eventually.

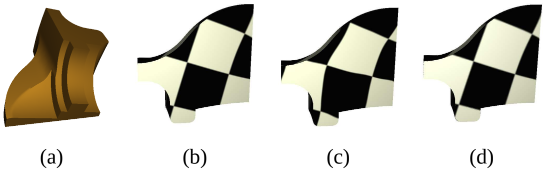

Fig. 5. Fan blade model (6,699 vertices): (a) Original model, (b) original model with checker board texture, (c) distortion introduced after embedding, (d) distortion eliminated by TIC.

3. EXPERIMENTAL RESULTS AND ANALYSIS

In the following experiments, a randomly-generated watermark of 100 bits is used with parameters $d_{off} = 10$, $f_t = 0.2$. All presented models provide about 50 correct bits before embedding.

3.1. Embedding Capacity

For a patch with n vertices, there are n components in the spectral descriptors $\{\tilde{u}\}$ and $\{\tilde{v}\}$. Since low frequency components will be omitted and each group of two components will hold one bit, we can embed $\lfloor \frac{n - d_{off}}{2} \rfloor$ bits in a patch with $|V| = n$. In other words, to embed bit sequence k, the patch needs to satisfy $|V| \geq d_{off} + 2|k|$.

3.2. Robustness Against Attacks

Affine Transformation Attack: This method is immune to uniform affine transformation attacks on T and/or G. That is, applying translation, rotation, uniform scaling, or combinations of them on T and/or G of the watermarked model will not erase any embedded bit due to the property of MH.

Shearing Attack: Although this method is not immune to shearing attacks on T, it is robust against moderate ones, as shown in figure 6.

Local Modification: As shown in figure 7, modifying texture coordinates of several vertices in T does not erase watermark bits easily.

Cropping Attack: This method can embed identical bits into different patches of the input model. For example, the mushroom model in figure 8 consists of 2 patches. k is embedded into those patches separately. As long as there is an uncut patch, the model could be authenticated.

Adding Noise Attack: Figure 9 shows the experimental result of Adding White Gaussian Noise(AWGN) attack on $\mathbf{u}$ and $\mathbf{v}$ with $SNR = 45db$. It shows that this method is robust against random noises.

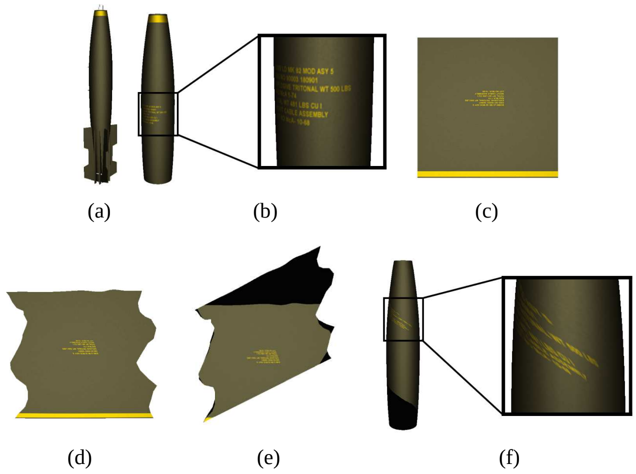

Fig. 6. Bomb model (6,426 vertices): (a) original model, (b) patch selected to be modified, (c) original texture mesh, (d) watermarked texture mesh with texture image compensation, (e) texture mesh after 30% shearing attack, (f) attacked model being rendered. 98 out of 100 embedded bits survived.

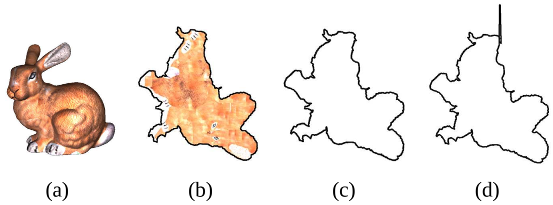

Fig. 7. Bunny model (35,190 vertices): (a) original model, (b) T with texture, (c) T', and (d) T' under the local modification attack of 2 vertices. 99 out of 100 embedded bits survived.

4. CONCLUSIONS

This paper proposes to embed watermarks by modifying texture mapping information rather than geometry information or texture image of 3D triangular models. It presents a new blind watermarking method that introduces no visual distortion with the help of Texture Image Compensation (TIC) technique. Neighbor Couple Embedding (NCE) is incorporated with Manifold Harmonics Transform (MHT) to provide robustness of the watermarks. Experiments show that this method is robust against a variety of attacks such as uniform affine transformation, shearing, local modification, cropping, and adding noise attacks.

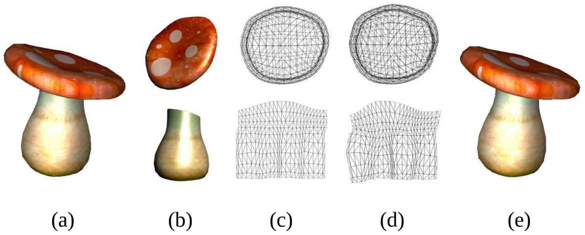

Fig. 8. Mushroom model consisting of 2 parts (596 vertices and 250 vertices): (a) original model, (b) 2 patches, (c) original texture mesh, (d) watermarked texture mesh, and (e) watermarked model being rendered.

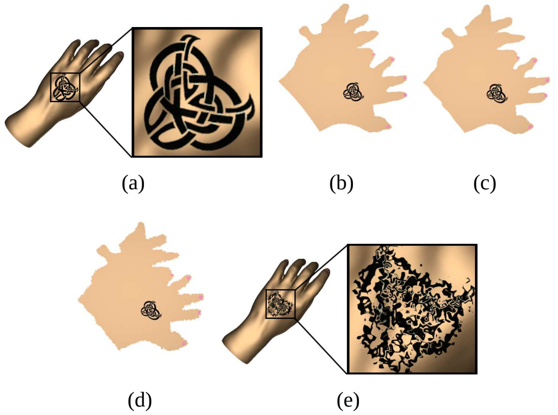

Fig. 9. Hand model (37,234 vertices): (a) original model, (b) original texture mesh with texture, (c) watermarked texture mesh with texture image compensation, (d) texture mesh under AWGN noise attack($SNR = 45db$), and (e) rendered model after attack. 98 out of 100 embedded bits survived despite the large visual distortion after attack.

5. REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Information hiding – A survey,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[2] I. Cox, M. Miller, J. Bloom, and M. Miller, *Digital Watermarking: Principles & Practice*, Morgan Kaufmann, 2001.

[3] K. Wang, G. Lavouè, F. Denis, and A. Baskurt, “Three-dimensional meshes watermarking: Review and attack-centric investigation,” in *Proceedings of the International Workshop on Information Hiding*, 2007, pp. 50–64.

[4] E. Garcia and J.L. Dugelay, “Texture-based watermarking of 3d video objects,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 8, pp. 853–866, Aug. 2003.

[5] Boon-Lock Yeo and Minerva M. Yeung, “Watermarking 3d objects for verification,” *IEEE Comput. Graph. Appl.*, vol. 19, no. 1, pp. 36–45, 1999.

[6] Bruno Vallet and Bruno Lévy, “Spectral geometry processing with manifold harmonics,” *Computer Graphics Forum (Proceedings of Eurographics)*, vol. 27, no. 2, pp. 251–260, 2008.