社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－CG－125 (2)
2006／11／16

Watermarking 3D Mesh Models Using Affine Invariant Based on Local Shape Feature

Shuhui Hou,† Masaaki Iiyama,†† Koh Kakusho††† and Michihiko Minoh†††

In this paper, we propose a new watermarking method for 3D mesh models by using affine invariant based on local shape feature characterized by the mesh complexity, which is expected to be robust against four types of attacks:affine transformation, additive random noise of vertex, cropping and mesh simplification. On the other hand, by enforcing an interactive comparison between embedding capacity and resistance to attacks, our watermarking scheme embeds the copyright information into the featured shape as many as possible, which is corresponding to the featured shape that we intend to protect.

3Dモデル形状の局所的な特徴によるアフィン不変な電子透かし埋め込む手法

侯 書会† 飯山将晃††
角所 考††† 美濃導彦†††

形状の局所的な特徴（複雑さ等）に基づいて得られるアフィン不変量を用いて、３Ｄモデルに対する電子透かし埋め込み手法を提案する。提案手法はモデルに対するアフィン変換、ランダムノイズ、クロッピング、メッシュの簡単化の四つの攻撃に耐えることを目的としている。また、著作権を主張すべき特徴のある形状に対して多くの透かしを埋め込むことができ、ランダムノイズへの耐性も高いという利点がある。

1. Introduction

In the last decade, the commercial value of 3D models is becoming high with their being diffused widely in several applications such as entertainment industry(movies and video-games). In addition, such 3D models are usually those designers or experts spend lots of time and cost to make. So the need for efficient watermarking schemes for 3D models' copyright protection becomes more eminent.

Digital watermarking is a technique for embedding secret information called a watermark in various target object data. The watermark must not interfere with the intended purpose of the target object data such as viewing (Embedding Requirements-Imperceptible), and the watermark should be inseparable from the target object data (Embedding Requirements-Robust). Embedded watermarks can be used to secure copyright. To fulfill copyright protection, the amount of the watermark which can be embedded in the model is large enough to record the information needed for the application(Embedding Requirements-Capacity). In general, the three requirements are conflicting. For example, if one needs more robust embedding, the amount of data that can be embedded is reduced. The best trade-off depends on the application.

1.1 Related Works

We cope with 3D models represented by triangular mesh in this paper. When the watermarked model is distributed and appropriated on internet, the embedded watermark may be destroyed by intentional or unintentional interferences from the appropriator. Here, we call the intentional or uninten-

† Dept. of Intelligence Science and Technology, Graduate School of Informatices, Kyoto University
京都大学大学院 情報学研究科
†† Graduate School of Econonics, Kyoto University
京都大学経済学研究科
††† Academic Center for Computing and Media Studies, Kyoto University
京都大学学術情報メディアセンター

tional interferences as attacks. Below we introduce some attacks which we often run into.

- **Geometric Transformations**
  Geometric transformations such as parallel translation, rotation, uniform or unequal scaling are sometimes used in computer graphics to position a 3D model inside a scene.
- **Additive Random Noise**
  The appropriator may change vertex coordinates by adding random noise to them.
- **Cropping**
  Situation where the appropriator only takes a part of the model rather than the whole one is called cropping attack in this study.
- **Mesh Simplification**
  Mesh simplification means that the number of polygons is decreased with the shape kept unchangeable. In order to achieve adequate rendering speed, mesh simplification is often needed.

The related works are explained from the view point of their robustness against these attacks.

The previous works to watermark 3D models can be classified into two groups according to their operating fields. One group includes approaches 1)~3) which embed watermark directly in the spatial(non-transformed) domain, and the other group includes approaches 4)~7) which operate in the transformed(e.g., frequency, wavelet, basis function, etc.)domain.

Until now, there is no spatial domain approach which is robust against all four types of the attacks above. Despite previous robust 3D transformed domain watermarking approaches already exhibit good resistance and robustness, they are often either limited to specific mesh or far too slow to cope with nowadays large mesh due to the involved complicated numerical computations.

In this paper, we provide a novel watermarking method robust to all four types of attacks which operates in spatial domain. Here, our method is robust against affine transformation by using affine invariant, robust against cropping due to localized and repeatedly embedding and is robust against random noise and mesh simplification for embedding watermark code into set of vertices instead of vertex directly.

On the other hand, our watermarking scheme embeds the copyright information into the complex shape, displaying lots of small variations on the 3D model surface, as many as possible, which is corresponding to the featured shape that we intend to protect.

2. Overview

The whole watermarking scenario which consists of the watermark embedding and the watermark extraction procedure is roughly introduced below.

First, we embed repeatedly copyright information as a watermark into the original model which needs to be copyrighted. Next, we hide the original model and the watermark information in a safe place and publish the watermarked model on internet. When the appropriator illegally distributes the watermarked model or claims it to be his or her own after intentional or unintentional altering, in order to assert ownership we must extract embedded watermark from watermarked model(may be degraded). We explain our watermarking scheme according to the following steps:

(1) **Watermark Embedding**
  (1-1)Constructing Embedding-Primitive
  (1-2)Constructing Embedding-Field
  (1-3)Constructing Embedding-Invariant
  (1-4)Embedding Watermark

(2) **Watermark Extraction**

3. Watermark Embedding

As we know, the area where we should insist on copyright is generally the shape with certain feature, and the shape with certain feature is usually complex. Hence, we choose the complex areas to embed copyright information, where the perturbation derived from watermarking is not easily perceptible. Moreover, such areas often exhibit more robustness against random noise.

3.1 Constructing Embedding-Primitive

For a 3D triangular mesh model, the patches which share a common edge are defined as neighbor patches. Given a set of patches, the patch which shares a common edge with the patches of set but not belong to the set was defined as one neighbor

patch of set.

Our scheme embeds one bit of watermark information into a certain set of vertices derived from adjoining triangular patches. As unit of alteration, the set of vertices is called embedding-primitive, which is constructed as follows:

- step1: Initial patch is specified as a triangular patch with the maximum distance from its center to the center of model and is considered as the set of patches.
- step2: Calculate angles between neighbor patches of the set and the initial patch by their normal vectors, and add the neighbor patches into the set in ascending order of angles. When the number of triangular patches included in the set is greater than threshold $n(n \in N)$, keep on adding this neighbor patch into the set and update the set if the angle is less than threshold $\theta(\theta \in (0, \pi))$. Otherwise,stop adding and go to step3.
- step3:Pick up another initial patch from the rest of model. Repeat from step1 to step2 until the number of unused patches is smaller than n.
- step4:The set of vertices included in such set is considered as candidate for embedding-primitive. In order to ensure that a vertex is used only one time,we delete the common vertice from the latter set. Candidates with few vertices is unstable, and not used for embedding. The candidate for embedding-primitive which passed this check is considered as embedding-primitive.

The constructing of embedding-primitives $P_i(n, \theta)$ are illustrated on **Fig.1**(a). Then one bit of watermark information can be embedded into one primitive.

With this way, more embedding-primitives are constructed on the relatively complex shape where normal vectors of triangular patches change frequently or there are lots of small triangular patches.

In addition, it is clear that the embedding capacity varies with parameter θ and n. The smaller the value of n or θ is, the larger the embedding capacity is.

3.2 Constructing Embedding-Field

We call the area where watermark code can be embedded only one time as embedding-field. Since appropriators usually use the continuous and meaning part of model, it is ideal to choose adjoining primitives in space to form an embedding-field. After the embedding-fields are fixed, watermark code is repeatedly embedded in.

3.3 Constructing Embedding-Invariant

We adopt affine invariant to embed watermark information. As we know, ratio of the lengths of two segments of a straight line is invariant to affine transformation and parallel lines are still parallel lines under affine transformation. It is easily proved that ratio of the lengths of two segments of parallel lines is also invariant to affine transformation. Then we employ it in our embedding scheme.

As shown on Fig.1(b), where,

o: center of mass of the model.

o_p: center of mass of primitive P_i.

(x_l, y_l, z_l): size of bouding box of model,

$\overrightarrow{oo_p} = (x_p - x_0, y_p - y_0, z_p - z_0) = (d_x, d_y, d_z)$: vector from o to o_p.

Obviously, d_x/x_l, d_y/y_l and d_z/z_l are ratios of the lengths of two segments of parallel lines, which are invariant to affine transformation. Here, we call d_x/x_l, d_y/y_l and d_z/z_l components ratio.

3.4 Embedding Watermark

Our embedding scheme seeks to change the components ratio of each primitive greater(embedding bit 1) or less(embedding bit 0) than the initial value. These changes are performed by moving the vertices in a certain direction, which in turn influence the components ratio since the size of model may be changed by the moving of vertices.

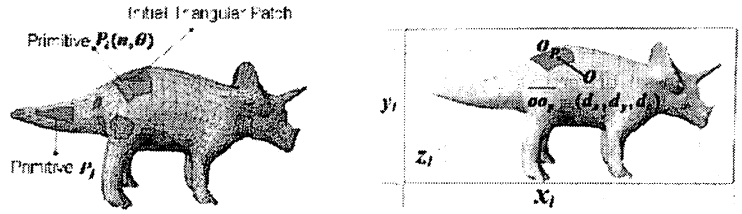
We modify components ratio according to the following formula:

$$\begin{cases} d_x/x_l \rightarrow r * d_x/x_l \\ d_y/y_l \rightarrow r * d_y/y_l \\ d_z/z_l \rightarrow r * d_z/z_l \end{cases} \quad (1)$$

where,

$$\begin{cases} r > 1 & w_i = 1 \\ 0 < r < 1 & w_i = 0 \end{cases} \quad (2)$$

w_i is a certain watermark code. Here, we practice it



(a)Embedding-Primitive Constructing (b)Embedding-Invariant Constructing

Fig.1 Watermark Embedding

by moving the vertices of primitive instead of moving the center of mass of the whole model. In other words, embedding code 1, we move all the vertices of primitive toward the outside(in a direction $\overrightarrow{oo_p}$) for getting greater components ratio. While embedding code 0, we move all the vertices of primitive toward the inside (in a direction $\overrightarrow{o_po}$)for getting less components ratio(without considering its sign of plus or minus).

Now we discuss the updating of vertices after watermarking. Assume

$v_1, v_2, \cdots, v_p$: vertices of primitive P_i,

$o_p = (x_p, y_p, z_p) = (1/p)\sum_{i=1}^{p} v_i$: center of mass of primitiveP_i,

$v_1^{'}, v_2^{'}, \cdots, v_p^{'}$: vertices of $P_i^{'}$ which is watermarked primitive P_i,

$o_p^{'} = (x_p^{'}, y_p^{'}, z_p^{'}) = (1/p)\sum_{i=1}^{p} v_i^{'}$: center of mass of primitive $P_i^{'}$.

From (1) and $\overrightarrow{oo_p} = (x_p - x_0, y_p - y_0, z_p - z_0) = (d_x, d_y, d_z)$, the center of mass of watermarked primitive $P_i^{'}$ is easily calculated by $o_p^{'} = r * (o_p - o) + o$. Then the shift of the center of mass of primitive is

$\Delta o = o_p^{'} - o_p = (1 - r) * (o - o_p)$.

By $o_p^{'} = (1/p) * \sum_{i=1}^{p} v_i^{'}$ and $o_p = (1/p) * \sum_{i=1}^{p} v_i$, the vertex v_i will be moved to $v_i^{'}$ according to

$$v_i^{'} = v_i + \Delta o = v_i + (1 - r) * (o - o_p) \qquad (3)$$

From (3), we know the variations of vertex coordinates resulted from watermarking process depend on $|1 - r|$. It is also clear that the visual quality of watermarked model is tightly related to the value of $|1 - r|$.

Here, we give some discussion about why our scheme is robust against random noise and mesh simplification.

- Random Noise

  That our embedding scheme is robust against the additive random noise of vertex coordinates consists in:

  - The additive random noise of vertex coordinates is averaged when the center of mass of primitive is calculated, which reduce the influence resulted from noise.
  - Theoretically, the embedded watermark code can be extracted correctly only if the moved distance of vertices resulted from watermarking process is greater than the moved distance resulted from random noise, which provides the permitted range of random noise.

- Mesh Simplification

  It is obvious that components ratio depends on the average of vertex coordinates of primitive instead of number of vertices or connectivity of the mesh. So our embedding scheme exhibits robustness against mesh alteration .

4. Watermark Extraction

As mentioned earlier, the modification of components ratio and the movement of vertices affect each other. To cope with this, when the attacked model appears to be in the different location or different size from watermarked model, we first bring it back to the same location and size as watermarked model. We align attacked model to watermarked model like this:first, the attacked model is rotated so that its principal axes of inertia coincide with principal axes of inertia of watermarked model. Next, the attacked model is adjusted in order to get the same orientation as the watermarked model. Then scale the attacked model to the same

size of the watermarked model based on bounding box. Last, the attacked model is transferred to match the center of mass of the watermarked model. Such operation is iterated automatically until that the attacked model appears to be in the same location and size as the watermarked model.

For the attacked model which has already been aligned, we calculate the distance between the vertices of attacked model and the ones of original model. Assign the IDs of vertices on the original model to the closest vertices on the attacked model. Then we can get the primitives which contain vertices with the same IDs as the embedding-primitive. Last,calculate the distance from the center of mass of such primitive and to center of mass of original model. Compare it with corresponding distance on the original model. Extract code 1 when the former is greater than the latter, or extract code 0 contrarily.

Finally, we should note:one can try to skip the alignment steps if the object appears to be in the same location , orientation, and scale.

5. Experiments and Results

Here we present some of the results that we obtained while testing the validity of our watermarking sheme. The triceratops model showed on **Fig.2**(a) was used as an example of original mesh model which had 2833 vertices 5661 triangular patches.

5.1 Overall Performance

We evaluated the performance of the watermarking method, which include embedding capacity, perceptual invisibility, and robustness against the four types of attacks.

In our implementation, we took bit sequence which was generated randomly as watermark code. An example of results for the triceratops model is shown on Fig.2(c), 552 bits of watermark code were embedded for the embedding methods as described above, with parameter $n = 6$, $\theta = \pi/6$ and $|1 - r| = 1/1000$.

5.2 Capacity

Our experiment data showed that more watermark codes can be embedded into the area where the shape is relatively complex(Fig.2(b), one color represents one primitive where one bit is embedded). We also tested that the smaller the value of n or θ is, the larger the embedding capacity is. On the other hand, the larger the value of n or θ is, the more robust the resistance to random noise and mesh simplification is. It is easy to attain optimum solution between capacity and robustness against random noise and mesh simplification by modifying n and θ.

5.3 Imperceptiblity

The appearance of watermarked model can hardly be distinguished from the appearance of the original model while $|1 - r| \in (0, 1/100)$. Here, we can improve the robustness against random noise by adopting greater value of $|1-r|$ where the shape is complex and smaller value of $|1 - r|$ where the shape is relatively simple.

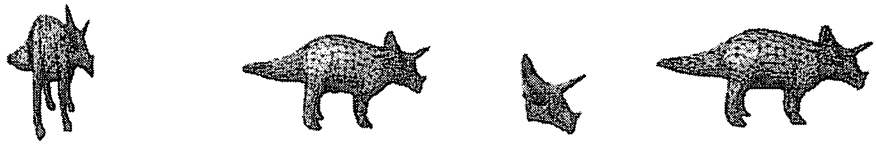
5.4 Robustness Against Attacks

We tested the robustness of our algorithm under several attacks:

- **Affine Transformation**
  No extraction error occurred while model was affine transformed arbitrarily(**Fig.3**(a)).
- **Additive Random Noise of Vertex**
  We added random noise of amplitude of 1.4 % of the maximum of the axis-aligned bounding box (Fig.3(b)). Despite that the distortion of shape is percetible, the watermark could be extracted without loss.
- **Cropping**
  We simulate an attacker to crop the continuous and meaning part of model, for example, the head (Fig.3(c))or other parts, and then test the robustness against cropping. The result is that we can extract the information without loss due to the cropped area is complex and large enough to contain one embedding-field or more.
- **Mesh Simplification**
  We simplified watermarked model triceratops by using MeshToSS(Kanai[8]), and watermark is still remained until the model is simplified to (90 %)(2833 → 2552)of original model vertex(Fig.3(d)).
  We should point out, when selecting the embedding areas, we did not take into account

(a) Original Model (b)Embedding Primitive (c) Watermarked model

Fig.2



(a) Affine Transformation (b)Additive Random Noise (c) cropping (d)Mesh Simplification

Fig.3 Robustness Against Attacks

if they possess more robustness against simplification than other areas. Moreover, we extracted information directly from simplified model without resampling process. Hence, the results revealed our embedding scheme holds potential robustness against mesh simplification.

6. Summary and Future Work

In this paper, we have presented a novel watermarking scheme which is robust against four types of attacks and we can change embedding capacity flexibly by enforcing an interactive comparison between capacity and resistance to attacks. Comparing to related works, our scheme showed robustness against a wider range of attacks and exhibited flexible and practical advantages.

As a future work, we plan to focus on the align process to offer better resistance when models have been cropped since a robust and accurate registration is crucial in watermark extraction while manual interaction is needed.

参 考 文 献

1) Ohbuchi, R. , Masuda, H. , and Aono, M: *Watermarking Three-Dimensional Polygonal Models*, Proc. ACM Multimedia'97, Seattle, Washington USA, pp. 261-272(1997).
2) Benedens, O: *Geometry-Based Watermarking of 3D Models*, IEEE CG and A, pp. 46-55(1999).
3) Adrian G. Bros: *Watermarking Mesh-Based Representations of 3-D Objects Using Local Moments*, IEEE Transaction on Image Processing, Vol. 15, No. 3(2006).
4) Emil, P. , Hugues, H. , and Adam, F. :*Robust Mesh Watermarking*, Proc. SIGGRAPH'99 , pp. 49-56(1999)
5) Kanai, S. , Date, H. , and Kishinami, T. : *Digital Watermarking for 3D Polygons Using Multiresolution Wavelet Decomposition*, Proc. of Sixth IFIP WG 5. 2 CEO-6, pp. 296-307(1998).
6) Ohbuchi, R. , Mukaiyama, A. , and Takahashi, S. : *A Frequency-Domain Approach to Watermarking 3D Shapes*, Computer Graphics Forum, 21(3), pp. 373-382(2002).
7) Wu, J. H. , Leif, Kobbelt: *Efficient Spectral Watermarking of Large Meshes with Orthogonal Basis Functions*, The Visual Computer, Vol. 21(8-10), pp. 848-857(2005)
8) Kanai, K. :*MeshToSS*, Version 1. 0. 1.