Yong-zhao ZHAN, Yan-ting LI, Xin-yu WANG, Yi QIAN, 2014. A blind watermarking algorithm for 3D mesh models based on vertex curvature. *Journal of Zhejiang University-SCIENCE C (Computers & Electronics),* **15**(5):351-362. [doi:10.1631/jzus.C1300306]

JZUS C (Comput & Electron)

A blind watermarking algorithm for 3D mesh models based on vertex curvature

Key words: Three-dimensional mesh model, Visual masking, Root mean square curvature, Blind watermarking, Attack resisting

Corresponding author: Yong-zhao ZHAN
E-mail: yzzhan@ujs.edu.cn

Motivation

- Design a robust blind watermarking algorithm for 3D mesh models based on vertex curvature, so as to maintain good robustness and improve visual masking for watermark embedding in 3D mesh models.

- Disadvantages of existing methods:
  - Existing methods might not satisfy the requirements for practical applications in watermark embedding and detection
  - Existing algorithms would not maintain good robustness to high degree attack
  - The visual masking for watermark embedding in 3D mesh models cannot be graciously improved

JZUS-C (Comput & Electron)

Features of our method

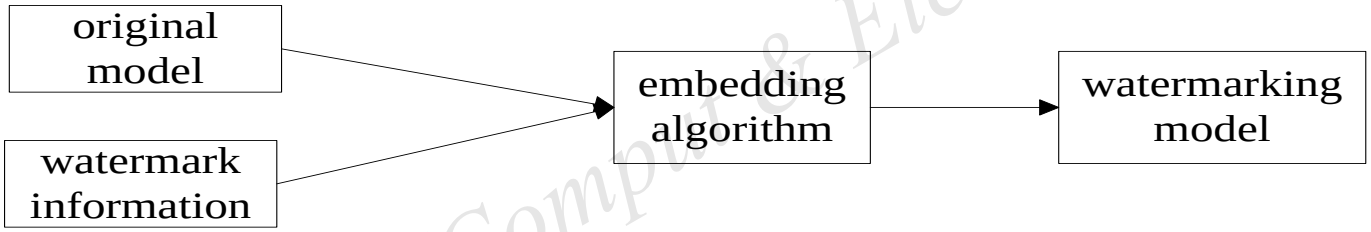
- Use root mean square curvature as the embedding primitive
- Maintain good robustness for vertex rearrangement, rotation, translating, uniform scaling, noise, smoothing, quantization, and simplification
- Improve the visual masking effect of watermark embedding
- Do not need to obtain reliable information about the original model

JZUSC (Comput & Electron)

Framework of our method

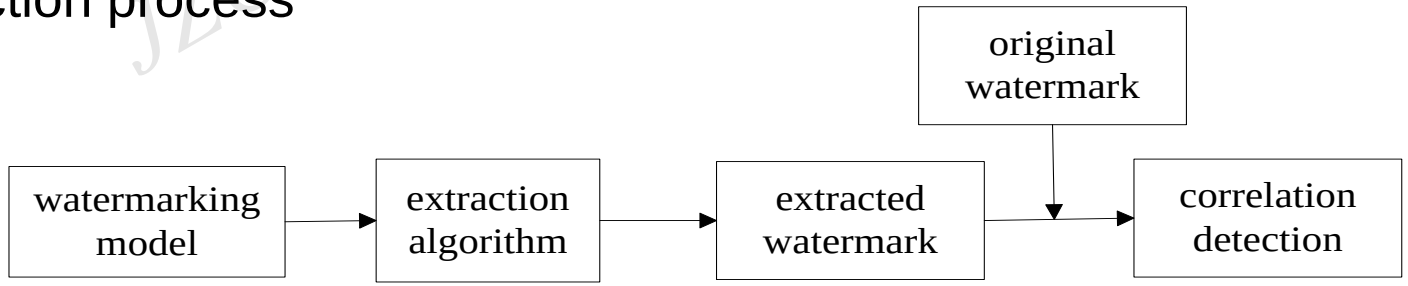
The blind watermarking algorithm based on vertex curvature can be divided into two processes:

1. Embedding process



The schematic diagram of the embedding process

2. Extraction process



The schematic diagram of the extraction process

JZUS C (Comput & Electron)

Framework of our method

1. Embedding algorithm

Step 1: Set the initial states of all vertices of the 3D mesh model valid.

Step 2: Calculate the root mean square curvature fluctuation values of all vertices.

Step 3: Sort the fluctuation values in ascending order.

Step 4: Divide the fluctuation values into bins.

Step 5: Normalize the fluctuation values in each bin and the value in every bin is in the range of [0, 1].

JZUSC (Comput & Electron)

Framework of our method

Step 6: Check the state of vertex and all the vertices in the local window of vertex and find the effective vertices.

Step 7: Modify the effective vertices in the bin in terms of the embedded watermark bit along the normal directions of the vertices until the sum of the fluctuation values of the vertices in the bin satisfies the condition.

Step 8: Set the states of these vertices and the vertices in their local windows invalid after modifying the effective vertices.

JZUSC (Comput & Electron)

Framework of our method

2. Extraction algorithm

Step 1: Calculate the mean fluctuation value of all vertices in the bin, similar to the embedding process.

Step 2: Extract the bits of watermark by comparing the mean fluctuation value of every bin with 0.5.

Step 3: Count the corresponding bits of the watermark sequence to form the final watermark.

JZUS C (Comput. & Electron)

Major results

We compared our algorithm with the algorithms of Cho *et al*. (2007) and Wang *et al*. (2011), which are concerned with visual masking and robustness. The experimental results for the simplification attack are as follows:

Model	Vertex reduction ratio	Our algorithm		Cho *et al*. (2007)		Wang *et al*. (2011)	
		Corr	MRMS	Corr	MRMS	Corr	MRMS
Bunny	70%	1.00	0.40	0.11	0.46	1.00	0.21
	90%	0.83	0.54	0.01	0.60	0.73	0.54
	95%	0.74	0.76	0.00	0.83	0.74	0.95
Dragon	70%	0.95	0.33	0.00	0.74	1.00	0.37
	90%	0.66	0.97	0.00	1.05	0.56	1.00
	95%	0.30	1.65	0.00	1.55	0.08	1.79
Horse	90%	1.00	0.10	0.00	1.51	1.00	0.13
	95%	1.00	0.08	0.00	1.52	0.96	0.24
	97.5%	0.78	0.12	0.00	1.55	0.87	0.43
Venus	90%	0.96	0.32	0.01	0.41	0.95	0.29
	95%	0.77	0.46	0.00	0.53	0.89	0.51
	97.5%	0.77	0.65	0.00	0.78	0.84	0.91

JZUS C (Comput & Electron)