Robust 3D Watermarking Using Vertex Smoothness Measure

Rakhi Motwani[1] and Frederick C. Harris, Jr.[1]
[1]Computer Science & Engineering Department, University of Nevada, Reno, NV, USA

Abstract - *Watermarking of 3D data has gathered renewed interest due to the explosion of graphic content and animated motion movies. This paper discusses non-blind watermarking of 3D models in the spatial domain. The proposed method estimates the local smoothness variation of the mesh to select vertices for inserting a watermark. Smoothness variation of the surface represented by the 1-ring neighborhood of each vertex is computed by the average angle difference between the surface normal and the average normal. Each vertex of the mesh is labeled in one of the bins corresponding to varying degrees of local smoothness variation from the low to the moderate to the highest variation. Vertices with the label of moderate local smoothness variation are then selected for the insertion of a random watermark. Simulation results prove that the inserted watermark is robust against cropping, affine operations and noise attacks, and at the same time imperceptibility is maintained.*

Keywords: imperceptible watermark, 3D graphics, content protection, local curvature estimation.

1 Introduction

Remarkable growth of digital media, such as the Internet, enables us to easily access, copy, modify and distribute digital content such as electronic documents, images, sounds and videos. In this scenario, there is a strong need for developing techniques for copyright protection of the original digital data thus preventing unauthorized duplication or tampering of it. 3D meshes are widely used in virtual reality, medical imaging, video games and Computer Aided Design (CAD). Watermarking of 3D meshes provides a solution to copyright infringement of 3D data.

Curvature estimation is an important task in 3D object description and recognition. Local bending of the surface is measured by curvatures. Surface curvature provides a unique viewpoint invariant description of local surface shape. Various algorithms on watermarking of three dimensional models have been proposed [1,2,3,10,11,12,13,14,15,16,17]. Published work incorporates spatial domain techniques that modify the geometry or connectivity and spectral domain techniques that use direct frequency analysis or multi resolution analysis. A brief explanation and classification of algorithms in spatial and spectral domain watermarking has been presented in [1]. In [2], attacking techniques have been described that can be roughly categorized into mesh-altering, topology-altering and visible pattern embedding methods. Alface [4] has done a thorough survey with classification and critical analysis of watermarking algorithms for 3D models. Benedens [5] has proposed selection of feature points on the 3D model and the comparison with the original model to determine whether the feature point has been moved inside or outside the surface along the normal. The method proposed in this paper takes inspiration from [5] by separating vertices into bins and using information about normals to faces. However, the proposed approach of calculating the bins and using information about normals is different and intuitive as well. A similar approach to use normals for computing curvature has been discussed in [6].

Gaussian and mean curvatures are the most commonly used measures for finding the curvature of a surface. A variety of curvature computing methods are discussed in [1]. However, these curvature measures capture the global characteristics of a surface. The authors propose to estimate local curvature variation to select areas for watermark insertion. The computed curvature estimate is local and relative to the geometry of the surface. A positive Gaussian curvature value means the surface is locally either a peak or a valley. A negative value means the surface locally has saddle points. A zero value means the surface is flat in at least one direction (i.e., both a plane and a cylinder have zero Gaussian curvature). Since the calculation of low and high curvature is relative to the model, the value of the curvature estimate is not significant. However, the variation in curvature is significant because it is used for selection of regions for embedding the watermark. Such regions are better qualified candidates for insertion of an imperceptible watermark as opposed to making selection based on globally computed Gaussian and mean curvature estimates.

2 Watermarking Algorithm

Fig. 1 outlines the various steps of the watermarking algorithm.

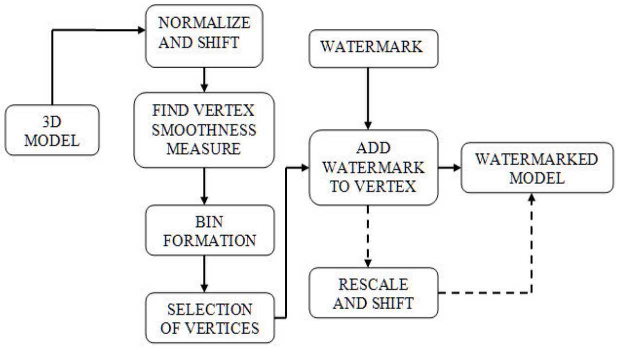

Fig.1. Watermarking Process

2.1 Normalizing and Shifting of 3D Model

At this initial stage, the 3D model's center of mass is determined and shifted to the origin of the rectangular co-ordinate system. The 3D model is then normalized by scaling the vertices coordinates to lie between -1 and +1 units. This step ensures that the watermark can withstand rotation and scaling transformations.

2.2 Finding Vertex Smoothness Measure

A mesh is a collection of polygonal facets, targeted to constitute an appropriate approximation of a real 3D object. It possesses three different combinatorial elements: vertices, edges and faces. The vertices represent a model's location and orientation in space, whereas edges connect the vertices to form faces, which in turn approximate the surface. Fig. 2 shows the wire frame model or mesh structure of a standard model. Fig. 4 demonstrates a simple mesh structure with vertices, edges, faces, face normals and shows the 1-ring neighborhood of a vertex in a model. All the vertices that a vertex under consideration is connected to, is called the 1-ring neighborhood of a vertex.

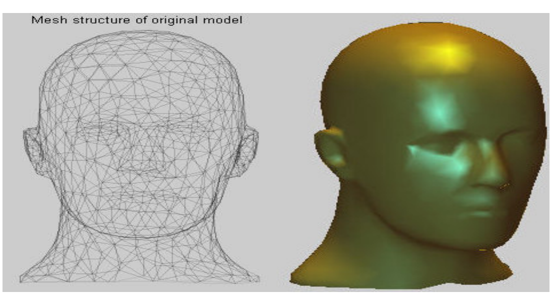

Fig.2. Mesh Structure of 'Mannequin' Model

The following steps are implemented to compute the local smoothness measure:

Step 1: Consider a vertex *v* from the mesh. Let *M* be the number of its adjacent faces. Find normals $\mathbf{N}_i$ to each face F_i which is formed by *v* and its neighboring vertices v_i as shown in Fig.3(a) and Fig.3(b).

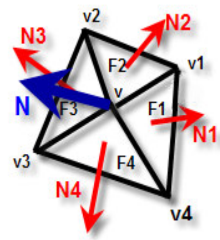

Fig.3(a) Average Normal

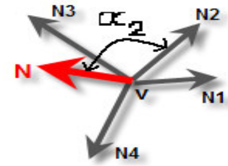

Fig.3(b) Normals Shifted to Pass Through Vertex

Step 2: Find the average resultant vector **N** of all the above normals passing through *v*.

$$\mathbf{N} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{N}_i \qquad \dots (1)$$

Step 3: Now compute angles α_i between each pair of N_i and N.

$$\alpha_i = \quad cos^{-1}\left(\frac{\mathbf{N}_i.\mathbf{N}}{|\mathbf{N}_i||\mathbf{N}|}\right) \qquad \dots (2)$$

Step 4: Compute the average of all these angles α_i which gives the local smoothness measure.

$$\alpha_{avg} = \frac{1}{M}\sum_{i=1}^{M}\alpha_i \qquad \dots (3)$$

Similarly, the algorithm is implemented at all the vertices in the mesh, to obtain local smoothness measure for the entire model. Thus, if the region around the considered vertex is flat, the angles α_i will be small in magnitude since the face normals will be almost parallel to the average normal. However, if the region represents a peak, the angle between the face normal and the average normal through the vertex, α_i will have a larger magnitude and so the smoothness measure's magnitude will be higher. Thus, this parameter α_i represents local geometry or shape of a surface or region. Fig.4 shows an angle α between the average resultant vector N passing through a vertex *v* and a normal to a face which has *v* as one of is vertices.

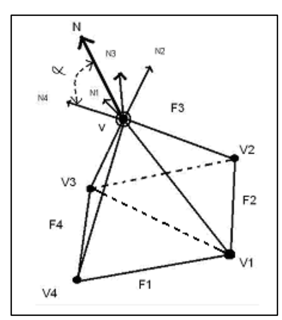

Fig.4. Normals of Faces Formed by Immediate Neighboring Vertices i.e. in the 1-Ring Neighborhood around a Vertex *v*

In computer graphics, Phong's illumination model computes the average normal for the vertex by interpolating the vertex normals across each face. On similar lines, it is intuitive to consider averaging the angle between each face normal with the average normal for the vertex under consideration, as shown in Equation (3). The illustration of this method is shown in Fig. 5. Since we are only considering the face normals in the 1-ring neighborhood of each vertex of the model, the measure is local. Local smoothness measure plots of some standard models (before watermarking), obtained by this method are displayed in the figure. The color bars on the right-hand side of the figures show the colors representing different smoothness measures.

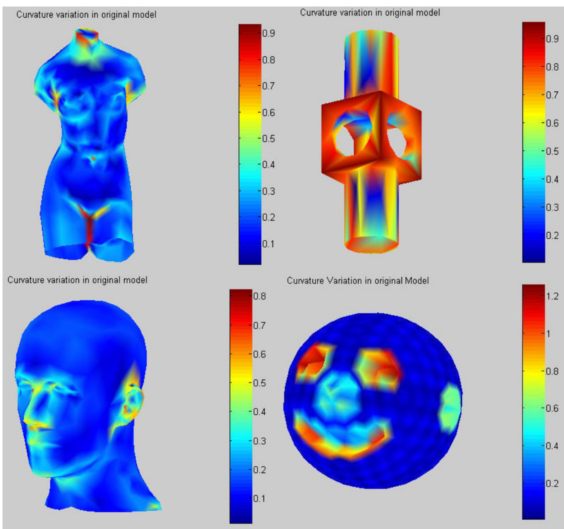

Fig.5. Curvature Variation in Original Models

2.3 Bin Formation

Based on the observed values of smoothness measure obtained for the vertex under consideration, we scale the degree of smoothness variation from 1 to 8. These scaled values are labeled into three bins. This scaling is done on the basis that the bins between 1 and 3 have a low smoothness measure, bins between 3 and 6 are have a moderate smoothness measure, and bins between 6 and 8 have a high smoothness measure. Thus, we can classify different regions of vertex smoothness measure in the model. This is in sharp contrast to other curvature methods such as Gaussian and Mean curvature. Fig. 6 shows the model and color bar indicating bins with pseudo colors (blue for the lowest variation and red for the highest variation). Toolbox graph [7] has been used to display the models in MATLAB.

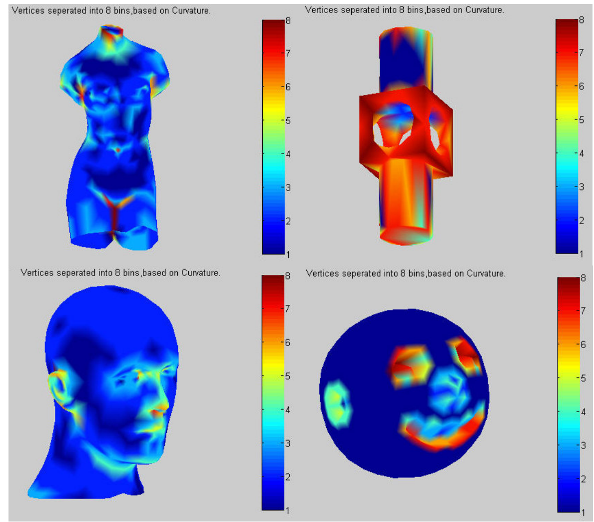

Fig.6. Bin Formation in Original Models

2.4 Selection of Vertices

Vertices lying in the regions which have moderate smoothness are selected to allow imperceptible distortions in the final watermarked model. High values of the smoothness measure represent very sharp changes such as peaks. Low values correspond to smooth or flat surfaces. Watermark insertion in these extreme high or low smoothness regions is perceptible due to response of the Human Visual System. Of the three bins, based on the observation of standard models, we selected the vertices lying in the moderate smoothness bins for insertion of the watermark. Fig. 7 shows these vertices (in dark red) in the models.

2.5 Insertion of Watermark

We insert a random sequence in the selected vertices. The difference between the watermarked vertices and the original vertices is the watermark. Thus, for each co-ordinate (x, y, z) of a vertex selected to be modified, we have:

$$v'(x, y, z) = v(x, y, z) + KW \qquad \dots (4)$$

where,
$v'(x, y, z)$ = Watermarked Vertex,
K = Scaling Factor,
W = Watermark Bits.

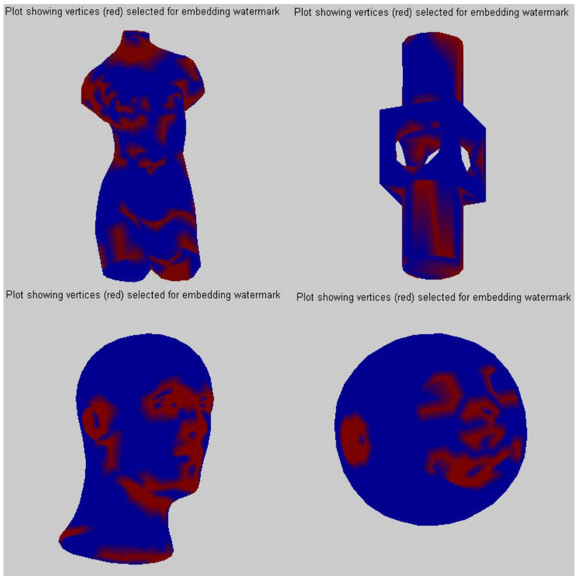

Fig.7. Vertices Selected for Watermarking (in red)

2.6 Rescaling and Shifting

Finally, the model is re-shifted to its initial location in space and the co-ordinates are also re-scaled. Thus, the watermark is inserted in the geometry of the model and this model can be distributed for use by others. The watermark inserted can be the logo of a company, the designer's identification, the user's signature or any other intellectual property. This watermarking method modifies only the locations of vertices, without changing the connectivity of vertices. Results of some of the watermarked models are shown in Figs. 8 and 9. As it can be seen from the figures, there is minimal perceptible distortion between the original model and the watermarked one. This proves that a watermark inserted in regions having moderate smoothness variation does not produce visible distortion in the model.

There is randomness in our process to ensure that even if the vertex smoothness measures of the watermarked model are calculated, extraction of the watermark is not guaranteed. This is a deterrent to brute force attacks to extract the watermark.

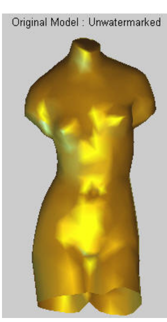

Fig.8(a). Original 'Venus' model

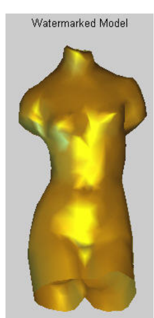

Fig.8(b). Watermarked 'Venus' model

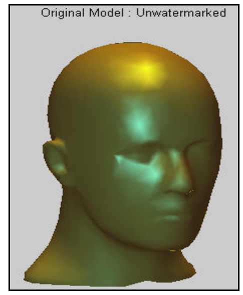

Fig.9(a). Original 'Mannequin' model

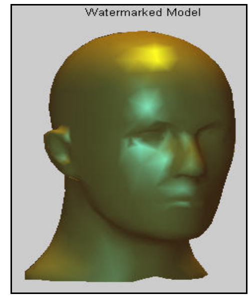

Fig.9(b). Watermarked 'Mannequin' model

3 Extraction of Watermark

Our watermarking technique requires the original model as well as the watermarked (and possibly degraded) model to do the extraction process. When a 3D model is attacked by vertex re-ordering, then in the absence of the original mesh, it is not possible to extract the watermark. This is why our method uses non-blind detection of the watermark.

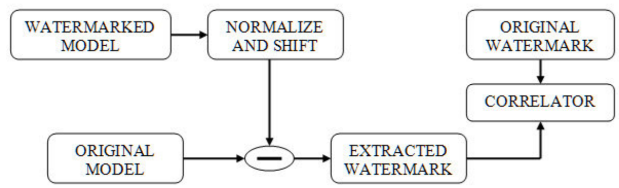

Fig.10. Watermark Extraction Process

The difference in magnitudes of corresponding vertices in the original un-watermarked model and the watermarked model will provide the watermark inserted. A correlation is performed between the original watermark and the extracted watermark, and the value of the correlation coefficient r gives us the extent of similarity between the embedded watermark x, and the recovered watermark y, both of size n.

$$r = \frac{\sum xy - \frac{(\sum x)(\sum y)}{n}}{\sqrt{\left(\sum x^2 - \frac{(\sum x)^2}{n}\right)\left(\sum y^2 - \frac{(\sum y)^2}{n}\right)}} \quad \text{... (5)}$$

Finding the correlation is a common method used to determine the extent of similarity between the original watermark, and the extracted watermark, as seen in [10, 11,12,13,14]. A correlation coefficient is a number between -1 and +1 which measures the degree to which two variables are linearly related. If there is perfect linear relationship with positive slope between the two variables, the correlation coefficient will be +1. If there is a perfect linear relationship with negative slope between the two variables, the correlation coefficient will be -1. A correlation coefficient of 0 indicates that there is no linear relationship between the variables. Multiplying by 100 gives us the percentage of correlation between the two watermarks. Percentage of correlation between the recovered watermark and original watermark is 100% in the absence of any attacks on the watermarked model.

4 Experiments and Results

An attack on a 3D model is an attempt to remove the watermark, but still retain enough of the model so that it can be used. 3D models are prone to operations like cropping, smoothing, noise addition, translation, rotation and scaling, which may destroy the watermark. This is not desired as the 3D model's ownership or copyright integrity inserted as a watermark may be destroyed as well. Thus, it is important that the watermark inserted should be robust enough to handle such attacks. To prove the efficiency of our method, typical attacks were simulated on the watermarked models. Table 1 gives the summary of tests and results for some models.

Table 1. Correlation Results for the Models for Vertex Smoothness Measure Method

Model ------>	Smiley	Mannequin	Venus
Total Vertices in Model	1026	688	711
Faces in Model	2048	1354	1396
Vertices Modified by Watermarking algorithm	94	275	195
Correlation after Uniform Scaling	100%	100%	100%
Correlation after Noise insertion	72.69%	79.67%	73.32 %
Correlation after Smoothing	79%	76%	62%
Correlation after cropping	87.1%	1005	85.01 %

4.1 Scaling, Translation, and Rotation

The implementation is completely invariant to uniform scaling and affine attacks. The change in these parameters does not affect the relative orientation of the normals at the vertices and thus the local smoothness measure for each vertex remains unchanged. Thus our algorithm gives 100% correlation between original and extracted watermarks.

4.2 Noise

This attack was simulated by adding normally distributed random numbers (with mean 0 and variance 0.3). Such an attack does affect the extracted watermark, but the correlation is still above the predetermined threshold 0.7. This threshold was found after attacking the watermarked model, and finding how much of the original watermark remains after the noise attack. Fig. 11 shows the result of noise attack on the 'Smiley' model.

Fig.11. Noise added to "Smiley" model

4.3 Smoothing

Smoothing has a considerable effect on the watermarked model. By smoothing, large transitions in surface levels are minimized by shifting or removal of some vertices. This resulted in degradation of the watermark. Fig. 12 shows the effect of HC smoothing on 'Venus' model. The HC smoothing algorithm is discussed in detail in [8].

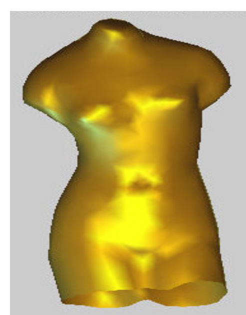

Fig. 12. HC Smoothed 'Venus'

4.4 Cropping

Cropping refers to removal or chopping of a part or parts of a model. The amount of watermark destroyed depends upon the extent of cropping. This necessitates adequate presence of the watermark in various regions. Fig.13 shows the results after cropping of the models. The technique is robust against cropping; a high value of correlation is obtained between the original watermark and the extracted one. Since no watermark was inserted in the forehead region of the 'Mannequin' model, no information was lost when the forehead was cropped, thus giving a 100%

correlation between the original watermark and extracted watermark.

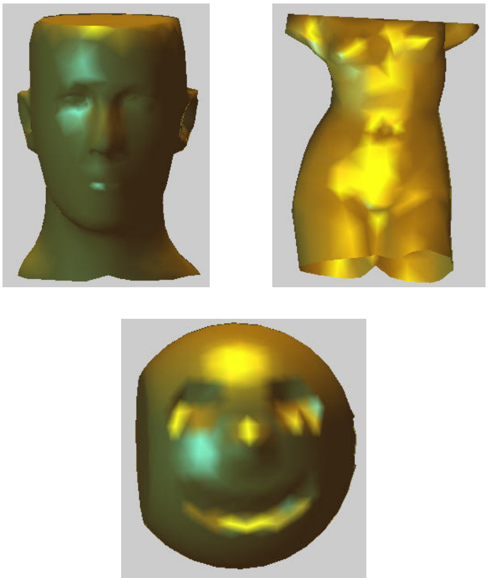

Fig.13. Cropped Models

4.5 Attacks to Destroy the Watermark

Assuming that an attacker knows what our watermark insertion criteria is, we tried to modify the model to destroy the inserted watermark. However, since random sequences were inserted into the selected vertices, we were not successful with our experiment since it resulted in a distorted model. Fig. 14 below shows the results of such modifications on some of the models.

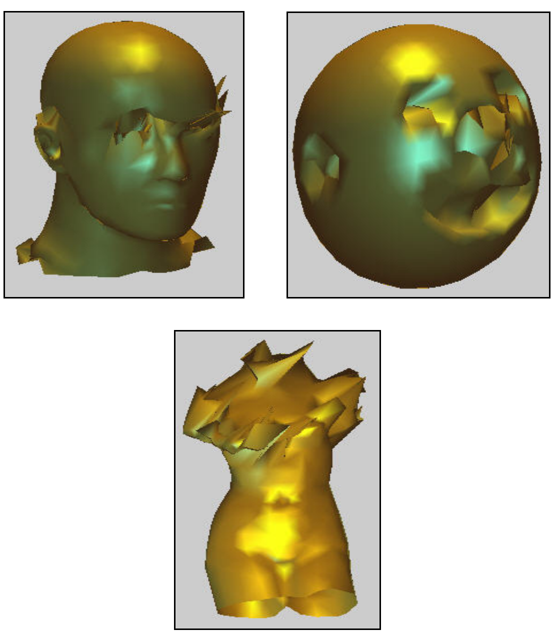

Fig.14. Attacks to Randomly Destroy the Watermark

5 Conclusions

In this paper, a non-blind spatial domain method has been proposed for watermarking of 3D models. The proposed method selects vertices based on local smoothness variation of a model, and gives good results (visual and analytical) for various types of surfaces (flat, curved, uneven, etc.) present on 3D models. The watermarking algorithm limits the perceptible distortion of the model by modifying vertices which have moderate local smoothness measures. The system is robust against various possible attacks and very easy to implement as well. This spatial watermarking method can be further adapted to utilize spectral domain information to improve robustness. For future work, the system could be extended to perform local shape analysis.

6 References

[1] P. Flynn and A. Jain: "On Reliable Curvature Estimation," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 110-116, 1989.

[2] K. Wang, G. Lavou´e, F. Denis, and A. Baskurt: "Three-Dimensional Meshes Watermarking: Review and Attack-Centric Investigation," *Conference on Information Hiding 2007*, 2007.

[3] O. Masuda and N. Aono: "Watermarking Three-Dimensional Polygonal Models," *ACM Multimedia 97, 1997, ACM Press*, New York, pp. 261-272, 1997.

[4] P. R. Alface: "Perception and Re-Synchronization Issues for the Watermarking of 3D Shapes," *Universite catholique de Louvain(UCL),* http://edoc.bib.ucl.ac.be:81 /ETD-db/collection/available/BelnUcetd-10192006-155635/, 2006.

[5] O. Benedens: "Robust Watermarking and Affine Registration of 3D Meshes," *Proc. of 5th International Workshop on Information Hiding*, Netherlands, October 7-9, pp. 177-195, 2002.

[6] R. Alface, P. Macq and B. Benoit: "Feature Based Watermarking of 3D Objects: Toward Robustness against Remeshing and De-Synchronization," *Proceedings of the SPIE*, Volume 5681, pp. 400-408, 2005.

[7] G. Peyré: "A Toolbox to Perform Computations on Graph," *URL:http://www.ceremade.dauphine.fr/~peyre/,* Toolbox_Graph.

[8] J. Vollmer, R. Mencl, and H. Muller: "Improved Laplacian Smoothing of Noisy Surface Meshes," *Computer Graphics Forum*, 18(3):131-138, 1999.

[9] MeshLab: "An open source, portable, and extensible system for the processing and editing of unstructured 3D triangular meshes," *URL: http://meshlab.sourceforge.net/* .

[10] M. Barni, F. Bartolini, V. Cappellini, M. Corsini, and A. Garzelli : "Digital watermarking of 3D meshes," *SPIE proceedings series, SPIE*, Bellingham WA, 2004.

[11] H. Wu and Y. Cheung: "A Reversible Data Hiding Approach to Mesh Authentication," *Proceedings of The International Conference on Web Intelligence IEEE/WIC/ACM*, pp. 774 – 777, 2005.

[12] J. Cho, R. Prost, and H. Jung: "An Oblivious Watermarking for 3-D Polygonal Meshes Using Distribution of Vertex Norms," *IEEE Transactions on Signal Processing*, Volume 55, Issue 1, pp. 142 – 155, Jan. 2007.

[13] N. Yi-qiang; L. Bo, and Z. Hong-bin: "A Blind Watermarking of 3D Triangular Meshes Using Geometry Image," *Computer Graphics, Imaging and Visualisation, 2007. CGIV '07,* pp. 335 – 340, 14-17 Aug. 2007.

[14] F. Uccheddu, M. Corsini, and M. Barni : "Wavelet-Based Blind Watermarking of 3D Models," *Proceedings of the ACM Multimedia and Security workshop*, pp.143-154, Magdeburg, Germany, Sept. 2004.

[15] M. Mitrea: "Toward Robust Spread Spectrum Watermarking of 3D Data," *Rapport de recherché,* N04007, May 2004.

[16] R. Ohbuchi, A. Mukaiyama, and S. Takahashi : "A Frequency-Domain Approach to Watermarking 3D Shapes", *Computer Graphics Forum*, Vol. 21, No. 3, 2002.

[17] M. Ashourian and R. Enteshary: "A new masking method for spatial domain watermarking of three-dimensional triangle meshes," *TENCON 2003 Conference on Convergent Technologies for Asia-Pacific Region*, pp. 428 - 431 Vol.1, 15-17 Oct. 2003.