

Académie de Montpellier
Université Montpellier II
Sciences et Techniques du Languedoc

MÉMOIRE DE STAGE DE MASTER M2

effectué au Laboratoire d'Informatique de Robotique
et de Micro-électronique de Montpellier

Spécialité : **Professionnelle et Recherche unifiée en
Informatique**

**Digital Watermarking For 3D Mesh
Models**

par **Maoyu DUAN**

Date de soutenance : **01/07/2008**

Sous la direction de **William.PUECH** et
Gérard.SUBSOL

Digital Watermarking For 3D Meshes Models

Maoyu DUAN

July 7, 2008

Contents

1	Introduction	4
2	Previous work	6
2.1	General digital watermarking	6
2.1.1	Objective and application	6
2.1.2	Principle of digital watermarking	6
2.1.3	Evaluation of Watermarking Algorithm	8
2.1.4	conclusion	9
2.2	Digital 3D watermarking	9
2.2.1	From watermarking to 3D mesh models	9
2.2.2	Applications of 3D watermarking	9
2.2.3	Requirements of 3D watermarking	10
2.3	Three-D mesh models	12
2.3.1	What is a mesh?	12
2.3.2	An example of 3D meshes	12
2.3.3	Classification of structures	12
2.3.4	Choosing the structure for watermarking	14
2.3.5	Conclusion	14
2.4	3D watermarking technique	15
2.4.1	The difficulty for 3D watermarking	16
2.4.2	Spatial Techniques	16
2.4.3	Transform techniques	18
3	The developed method for 3D watermarking	21
3.1	Algorithm of Cayre and Macq for 3D watermarking	21
3.1.1	Generation of a secret key	21
3.1.2	Embedding of information	22
3.2	Our Algorithm	25
3.2.1	Transforming the information from ASCII to binary	25
3.2.2	Embed the information to 3D meshes	25
3.2.3	Extract the information from 3D object	28
3.3	Experimental results	29
3.3.1	Capacity	29

3.3.2	Imperceptibility	29
3.3.3	Robustness	31
4	Conclusion and perspectives	42

Acknowledgement

Here, I would like to acknowledge William PUECH and Gerard SUBSOL for leading and teaching me in this training with their great patience.

I want to say thanks to Philippe AMAT and Nicolas TOURNIER for their helps on program and mathematics.

Chapter 1

Introduction

With the advent of digital media processing and content distribution, digital content has become widely available, and many digital products have been created. Through many of the existing tools and the Internet, people can obtain, duplicate, process, and distribute these media content relatively easily. However, these facilities are also exploited by pirates who use them illegally for their personal gains and violate the legal rights of the content providers. Because of possible copyright violation, potential authors are discouraged to publish their creative work in digital media.

Digital watermarking has been considered a potential efficient solution for copyright protection of various multimedia contents. This technique carefully hides some secret information in the cover content. The given secret information is called a digital watermark. Compared with traditional cryptography, digital watermarking technique is able to protect digital works after the transmission phase and the legal access. In addition to ownership protection, digital watermarking has also been used for copy control, authentication, conveying private information and so on. Further information on the applications of digital watermarking can found in [DD]. Due to its potentially wide applications in digital content protection, digital watermarking has received much attention in recent years and becomes a major focus in multimedia research.

Depending on the end application, watermarking techniques can fall into two broad categories: robust watermarking and fragile watermarking. In robust watermarking, the embedded watermark is undeletable unless the host data is rendered unusable and the watermark can resist any innocent and malicious attacks. Contrary to robust watermarking, the embedded data in fragile watermarking is sensitive to any changes of the host data. Usually, one hopes to construct a robust watermark, which is able to go through common malicious attacks, for copyright protection purpose. But

sometimes, the watermark is intentionally designed to be fragile for authentication applications.

This work is in the framework of ICAR team ¹of Lirmm. In this work, we present an adaptation of a 3D watermarking scheme that embed watermark information by modifying the local geometry of the model. In chapter 2, we will review framework of digital watermarking, then detail 3D digital watermarking from principal theories to concrete methods, as well as a classification of these methods. In chapter 3, our algorithm which is based on the algorithm of Cayre and Macq will be presented, we also test its some properties like robustness, capacity and imperceptibility. Lastly we conclude our work in chapter 4.

¹Image and Interaction (ICAR) team develops its activity in three domains involving scientific image and interaction: (1) coding and protection of images and videos, (2) image processing and computer vision, (3) 3D modeling, virtual and augmented reality.

Chapter 2

Previous work

In this chapter, at the beginning, we introduce digital watermarking in general. The objective, broad principle, protocol and evaluation of digital watermarking are referred to in section 2.1. In section 2.2, we detail different methods and attacks for digital 3D watermarking, as well as structure of 3D mesh model.

2.1 General digital watermarking

2.1.1 Objective and application

Over the last decennium, the digital rights management (DRM) problem of protecting data from theft and misuse has been addressed for many information types, including software code, digital images, videos, audio files and 3D graphical models. In addition to ownership protection, digital watermarking has also been used for copy control, authentication, conveying private information, verification of the integrity and so on. Further information on the applications of digital watermarking can be found in [DD].

2.1.2 Principle of digital watermarking

The information to be embedded is called a digital watermark, although in some contexts "digital watermark" means the difference between the watermarked signal and the cover signal. The signal where the watermark is to be embedded is called the host signal.

A watermarking system is usually divided into three distinct steps: embedding, attack and detection, as illustrated in Fig. 2.1. In embedding, an algorithm accepts the host and the data to be embedded and produces a watermarked signal.

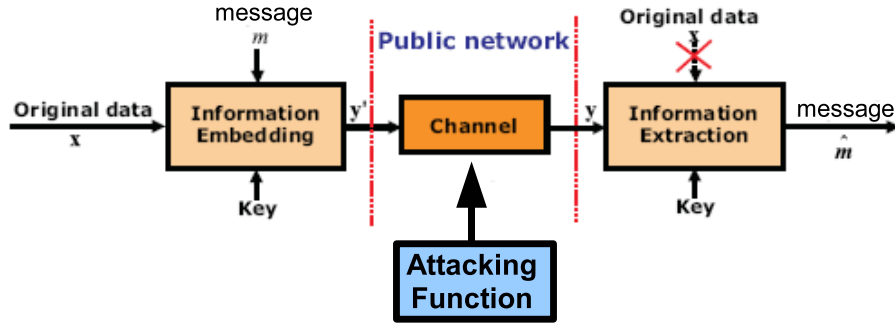


Figure 2.1: Generalized diagram of watermarking steps.

The watermarked signal is then transmitted or stored, usually transmitted to another person. If this person makes a modification, this is called an attack. While the modification may not be malicious, the term attack arises from copyright protection application, where pirates attempt to remove the digital watermark through modification. There are many possible modifications, for example, lossy compression of the data, cropping an image or video, or intentionally adding noise.

Detection (often called extraction) is an algorithm which is applied to the attacked signal to attempt to extract the watermark from it. If the signal was unmodified during transmission, then the watermark is still present and it can be extracted. In robust watermarking applications, the extraction algorithm should be able to correctly produce the watermark, even if the modifications were strong. In fragile watermarking, the extraction algorithm should fail if any change is made to the signal.

Protocol of digital watermarking

From a protocol point of view, we distinguish non-blind and blind watermarking schemes depending on whether the original digital work is needed at extraction, or not. Blind schemes are of special interest as they provide automatic retrieving of the payload (watermark) without any kind of assistance.

Attacking watermarking

Attacks on watermark play an important role for the suitable watermarking algorithms. General watermarking attacks can be classified in four categories following the analysis of Voloshynovskiy *et al.*[TVK⁺]: removal attacks, geometrical attacks, cryptographic attacks, and protocol attacks. But such a

classification has not been proposed yet for 3D watermarking attacks. See a detail about 3D watermarking attacks in section 2.2.3.

2.1.3 Evaluation of Watermarking Algorithm

The evaluation of digital watermarking schemes can provide detailed information for watermark designer or end users. Therefore, different evaluation strategies exist. So the broad constraints are as follows:

Capacity Capacity is a quantity of the information which can be embedded in the cover content (like audio, video, image or model 3D). The number of the bits which is embedded depends on the cover content or the objective of the application. In fact, between 16 and 64 bits are enough to protect copy right, but for hiding some specific data like a logo or a complete document, in that case, we need more space for embedding bits. Capacity of a watermarking algorithm is often measured according to the volume of the watermarking information and volume of the cover content.

Imperceptibility A watermark is called imperceptible if the original cover signal and the marked signal are perceptually indistinguishable. On the contrary, it is called perceptible. In general, in the watermarking domain, the distortion must be imperceptible, moreover, most watermarking algorithms are tested by HVS¹ for estimating the watermarking distortion. Moreover, PSNR (Peak Signal to Noise Ratio) and UIQI (Universal Image Quality Index) techniques were defined to measure for transparency and universal image quality, respectively. And Hausdorff distortion is used for the case of 3D Object. This measure will be detailed in chapter 3.

Robustness A watermark is called robust if it resists a designated class of transformations. On the contrary, it is called fragile. Robust watermarks are commonly used in copyright applications (to carry ownership or forensic information) and copy protection applications (to carry copy and access control information). Robustness is considered a broad element in the measure where a multimedia document can suffer some probable manipulation. Moreover, for evaluating the robustness of one watermarking algorithm, there are some existing protocols for image. We can cite Stimark [Sti], Checkmark [Chea], Optimark [Opt], and Certimark [Cer]. For the other kinds of cover contents (like video and object 3D), we have still some difficulties for this evaluation of robustness.

¹The Human Visual System Model, often referred to as the Human Visual System (HVS), is used by image processing, video processing and Computer vision experts to deal with biological and psychological processes that are not yet fully understood.

2.1.4 conclusion

In general it is easy to create robust watermarks or imperceptible or high-capacity watermarks, but the creation of robust, imperceptible and high-capacity watermarks has proven to be quite challenging. So nowadays, we are still in the case of compromise among robustness, capacity and imperceptibility as shown in Fig. 2.2.

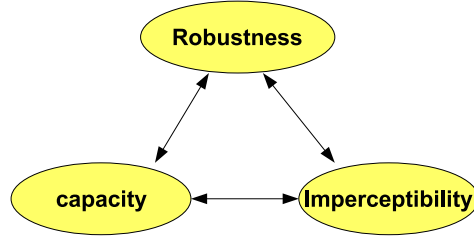


Figure 2.2: Compromise among capacity, imperceptibility and robustness

2.2 Digital 3D watermarking

2.2.1 From watermarking to 3D mesh models

Over the past few years, 3D hardware has become much more affordable than ever, allowing the widespread use of 3D meshes from CAD ²/CAM ³ industry into video games and other end-user applications. 3D meshes have become of great interest since they are widely used. And they are also a type of multimedia content, which in some cases has to be enhanced using watermarking techniques for copyright protection. However, due to the very particular specificities of 3D objects, 3D watermarking is far from the maturity of watermarking algorithms dedicated to regularly sampled signals such as audio, image and video watermarking. The need for secure communication of high value 3D virtual objects become very important, 3D watermarking activity is increasing in simulation, entertainment, industrial design and cultural heritage.

2.2.2 Applications of 3D watermarking

These applications can be classified following the general classes on digital media watermarking:

²Computer-aided design refers to the use of computer tools to assist engineers, architects and other design professionals in their design activities.

³Computer-aided manufacturing refers to the use of computer systems for the control of robotics and tools during the product manufacture.

Intellectual Property Rights (IPR) protection applications: This class of applications includes copyright protection, fingerprinting, usage control and forensic. For these applications, watermarking schemes are used to robustly convey information about content ownership and IPR.

Content verification applications: The goal of the watermarking scheme in this case is to indicate whether the content has undergone any alteration, in certain cases, to determine the type and location of such alteration. These applications are authentication and integrity checking.

Data Hiding applications: In this class of applications, watermark aim at conveying hidden information which is related or not to the content. Content-related information is mainly used for functionality enhancement purposes or for adding value to the content. Other kinds of hidden information are more related to steganography purposes.

2.2.3 Requirements of 3D watermarking

The requirements of these three classes of application contexts are very different. But they are often described in terms of capacity, robustness, imperceptibility, security and complexity.

Robustness and attacks

Robustness concerns the ability of the embedded watermark to resist against a given class of usual or malicious manipulations of the content. These manipulations are often called *attacks*. Here, we illustrate the relative wider variety of attacks a 3D watermarking scheme may undergo when compared with audio, image and video watermarking.

Similarity transforms: Rotation, uniform scaling and translation (RST) transforms are mesh geometry modifications which are considered as common mesh manipulations. They are often referred to as the minimal requirement for a 3D watermarking scheme.

Noising: Noising attacks are usually performed by white gaussian noise addition on vertex coordinates.

Connectivity attacks: Connectivity attacks modify the mesh adjacency information without modifying geometry. Among these attacks, vertex re-ordering is a manipulation which may desynchronize hidden data without any geometry or topology modification. Indeed, on the contrary of audio and image, the order of 3D mesh samples has no physical meaning.

Sampling attacks: Sampling attacks are here referred to as attacks which modify the mesh geometry and connectivity but leaving its shape topology unchanged. Sampling modifications include mesh simplification (lower resolution) mesh refinement (higher resolution) and remeshing (local or global point density and connectivity changes)

Topological attacks: Topological attacks are complex attacks which may change the topological features of the mesh shape (topological thus refers to shape topology). Cropping is the most well-known attack of this class.

Geometrical attacks: This kind of operation attacks the watermark by changing the local geometrical of a model. The geometrical transforms include mesh bending, mesh editing, mesh morph and local deformations.

Besides the above attacks there still exist many other forms of more complex attacks such as non-uniform scaling along arbitrary axes, projection, sampling after simplification and so on. However, these attacks usually degrade the visual quality and usability of the model so heavily that we can consider the resulting model is substantially different from the original model. Thus our watermarking algorithm is not designed to guard against such operations.

Imperceptibility

Evaluating whether two shapes are differently perceived when rendered on a 2D screen is a difficult yet necessary task to evaluate imperceptibility. Most metrics used for benchmarking 3D watermarking schemes have been developed in the field of mesh simplification. These are the Hausdorff distance, the Root Mean Square Error (RMSE) (a.k.a. Vertex Signal-to-Noise Ratio (VSNR)) and the Geometric Laplacian Distortion Metric. The Hausdorff distance is based on a point to surface distance, the RMSE and VSNR are based on mean point-to-point Euclidian distances and the Geometric Laplacian. However, these metrics give poor estimations of the perception of the mesh shape since the human eye is much more sensitive to perturbations of a surface smoothness by random additive noise than to the smoothing of an already smooth surface yet producing the same metric error. In conclusion, most watermarking schemes and attacks are limited to be imperceptible. However, there is still a lack of standard tools enabling to assess such imperceptibility.

Capacity

The capacity of 3D watermarking depend on the content aimed at by the application. such as an authentication or a data hiding applications, the

mesh representation is a content to protect. In that case, the capacity directly depends on the number of points or faces in the mesh. And for the applications related to IPR (Intellectual Property Rights) protection, the content is the shape approximated by the mesh. The watermarking capacity related to the shape certainly depends on the curvature variations of the surface and geometrical structure of the model.

security

The principles of 3D watermarking security are not so different from the general watermarking case. The quality of the secret key is a very important parameter for this property.

2.3 Three-D mesh models

Abstract The study of polygon meshes is a large sub-field of computer graphics and geometric modeling. Different representations of polygon meshes are used for different applications and goals. The variety of operations performed on meshes may include boolean logic, smoothing, simplification, and many others.

2.3.1 What is a mesh?

A mesh is a collection of polygonal facets targeting to constitute an appropriate approximation of a real 3D object. It possesses three different combinatorial elements: *vertices*, *edges* and *facets*. From another viewpoint, a mesh can also be completely described by two kinds of information. The *geometry* information gives essentially the positions (coordinates) of all its vertices, while the *connectivity* information provides the adjacency relations between the different elements.

2.3.2 An example of 3D meshes

As we can see in the Fig. 2.3, the facets usually consist of triangles, quadrilaterals or other simple convex polygons, since this simplifies rendering, but may also be composed of more general concave polygons, or polygons with holes. The degree of a facet is the number of its component edges, and the valence of a vertex is defined as the number of its incident edges.

2.3.3 Classification of structures

Polygon meshes may be represented in a variety of structures, using different methods to store the vertex, edge and face data. In general they include/

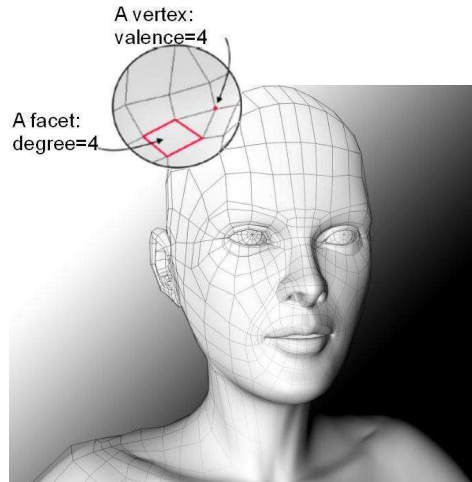


Figure 2.3: Example of 3D mesh with the valence of a vertex and the degree of a facet

Face-Vertex Meshes: A simple list of vertices, and a set of polygons that point to the vertices it uses.

Winged-Edge Meshes: - In which each edge points to two vertices, two faces, and the four (clockwise and counterclockwise) edges that touch it. Winged-Edge meshes allow constant time traversal of the surface, but with higher storage requirements.

Half-Edge Meshes: Similar to Winged-Edge meshes except that only half the edge traversal information is used.

Quad-Edge Meshes: A quad-edge mesh stores edges, half-edges, and vertices without any reference to polygons. The polygons are implicit in the representation, and may be found by traversing the structure. Memory requirements are similar to half-edge meshes.

Corner-Table: A corner-table stores vertices in a predefined table, such that traversing the table implicitly defines polygons. This is in essence the "triangle fan" used in hardware graphics rendering. The representation is more compact, and more efficient to retrieve polygons, but operations to change polygons are slow. Furthermore, Corner-Tables do not represent meshes completely. Multiple corner-tables (triangle fans) are needed to represent most meshes.

Vertex-Vertex Meshes: A Vertex-Vertex mesh represents only vertices, which point to other vertices. Both the edge and face information is implicit

in the representation. However, the simplicity of the representation allows for many efficient operations to be performed on meshes.

2.3.4 Choosing the structure for watermarking

The representations above each have particular advantages and drawbacks. The choice of the data structure is governed by the application, the performance required, size of the data, and the operations to be performed. For example, it's easier to deal with triangles than general polygons. For certain operations it is necessary to have a fast access to topological information such as edges or neighboring faces; this requires more complex structures such as the winged-edge representation. For hardware rendering, compact, simple structures are needed; thus the corner-table (triangle fan) is commonly incorporated into low-level rendering API such as DirectX and OpenGL.

In this training, the Face-Vertex Meshes with the faces which consist of the triangles, were used as its structure, moreover this representation is the most popular. Because a 3D model can be represented by an infinite number of triangle mesh representations without significantly altering its perceived quality. And it's easy to deal with triangles especially in computational geometry. Triangle meshes are a general representation of a 3D visual object that are very well suited to communications. Triangle offer many steganographic possibilities through modifications of their elementary features, vertices (geometry), and connectivity (topology). And it is possible that a structure of the quadrilateral or the polygon is transformed to a structure of the triangles. Let's see an example of a unit pyramid which is illustrated as Fig. 2.4. This pyramid may be represented by Fig. 2.5 which includes two tables of structure information. The right table is for a list of vertices, and the left one is a list of faces.

We can see, there is a quadrilateral in the last line of the faces list. The transformation from a structure of multi-shape to a problem of simplex triangle permit to simplify the computational geometry. So we can decompose the quadrilateral like Fig. 2.6 in the structure of the pyramid. In my training, this similar structure was used for watermarking.

2.3.5 Conclusion

Although there are many other 3D representations, 3D mesh has been the defacto standard of numerical representation of 3D objects thanks to its simplicity and usability. Furthermore, it is quite easy to convert other representations to 3D mesh, which is considered as a low-level but effective model. Up to now, 3D watermarking has mainly focused on triangle meshes

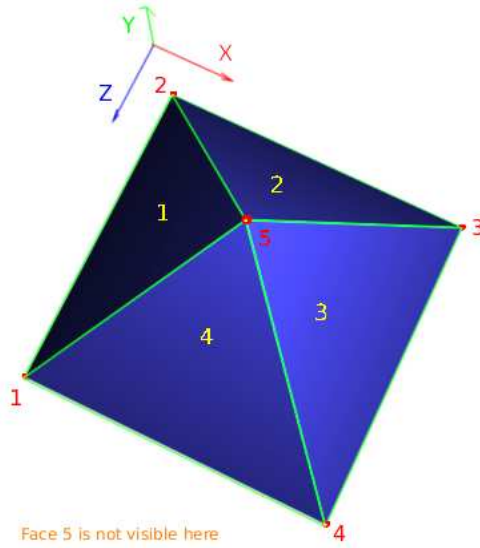


Figure 2.4: A 3D pyramid mesh. The mesh consists of 5 vertices, 8 edges and 5 faces which consist of 4 triangles and 1 quadrilateral.

Vertex index	X Coordinate	Y Coordinate	Z Coordinate	Face index	Type	Vertex 1	Vertex 2	Vertex 3	Vertex 4
1	-1	0	1	1	Triangle	1	2	5	
2	-1	0	-1	2	Triangle	2	3	5	
3	1	0	-1	3	Triangle	3	4	5	
4	1	0	1	4	Triangle	4	1	5	
5	0	1	0	5	Quadrilateral	1	2	3	4

Figure 2.5: The vertices list and the faces list

which are the most used for digital representations of the shape of a 3D model.

2.4 3D watermarking technique

In this survey, we describe most well-known and recent contributions to 3D watermarking. Existing techniques concerning 3D meshes can be classified in two main categories, depending whether the watermark is embedded in the spatial domain (by modifying the geometry or the connectivity) or in the spectral domain (by modifying some kind of spectral-like coefficients).

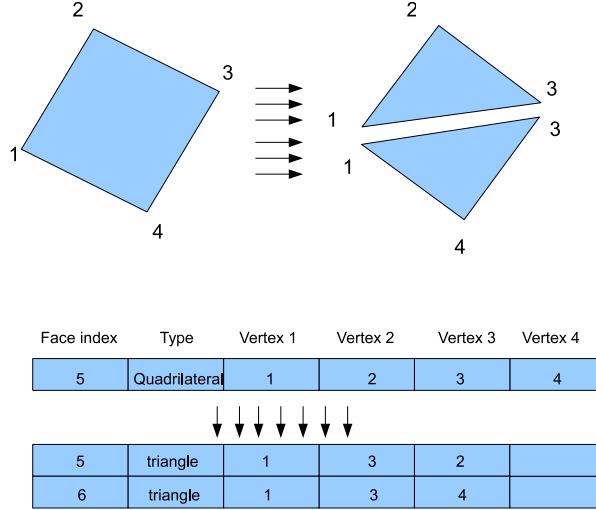


Figure 2.6: The transformation from a quadrilateral to a triangle

2.4.1 The difficulty for 3D watermarking

The 3D object appears as a list of elementary connected objects (polygons), which is very different from the usual regular sampling of pixels in photographic and video imagery. We can consider an image as a matrix, and each pixel as an element of this matrix. This means that all these pixels have an intrinsic order in the image, for example the order established by row or column scanning. On the contrary, there is no simple robust intrinsic ordering for mesh elements. Some intuitive orders, such as the order of the vertices and facets in the mesh file, and the order of vertices obtained by ranking their projections on an axis of the objective coordinate system, are easy to be altered. Global geometrical manipulations (like scaling, translation, rotation) are not easy to deal with when working on sampled meshes. In addition, because of their irregular sampling, we are still short of an efficient and effective spectral analysis tool for 3D meshes. This situation, makes difficult to put the "secure spread spectrum" watermarking schemes into practices.

2.4.2 Spatial Techniques

The 3D watermarking schemes which embed data in the spatial domain may be classified in two main categories: Connectivity-driven watermarking schemes and Geometry-driven watermarking schemes.

Schemes modifying the geometry

All the geometrical techniques of 3D watermarking are implemented by modifying the coordinates of vertices. But these algorithms that modify the positions of the vertices are usually fragile. Here we introduce some famous algorithms as follow: (All of these different methods are from [WDB08], except the methods of Mr AMAT)

Yeo and Yeung [YY] proposed such an algorithm that serves for mesh authentication. The basic idea is to search for a new position for each vertex where two predefined hash functions have an identical value, so as to make all vertices valid for authentication. In fact, this algorithm depends on a pre-established vertex order, which causes a causality problem.

Yu *et al.* [YIK] proposed another non-blind robust algorithms. Vertices are divided into N groups and in each of them is inserted one bit by modifying the length from its member vertices to the gravity center of the mesh. The modulation scheme is a simple additive method with an adaptive intensity obtained by a local geometrical analysis of the mesh. The extraction is also quite simple, since it is sufficient to regroup the vertices and inverse the additive insertion model. However, to ensure a good robustness, a pre-processing step of registration and resampling is necessary, which makes the algorithm non-blind.

In Benedens' "vertex flood algorithm" [Ben], first of all, all of the vertices were grouped, then the according to their distances to the center of a designated triangle, the range of the group interval is divided into $m = 2^n$ subintervals, and all the group vertices distances to the chosen triangle center are altered so that the new distances all fall into a certain subinterval that stands for the next n watermark bits.

Cayre and Macq [CM] proposed a high-capacity blind data-hiding algorithm for 3D triangular meshes. By choosing the orthogonal projection of a vertex on its opposite edge in a triangle as the primitive, the theoretical capacity can attain 1 bit per vertex. The synchronizing mechanism relies on the choice of the first triangle by a certain geometrical criterion, and a further spreading scheme that is piloted by a secret key.(In section 3, this algorithm will be explored in detail)

Bors [Bor] also reported a blind algorithm. The primitive is the relative position of a vertex to its 1-ring neighbors. A two-state space division is established, and the vertex is assumed to be moved into the correct subspace according to the next watermark bit.

Ohbuchi et al. [OMA] chose the ratio between the height of a triangle and its opposite edge length as primitive to construct a watermarking technique that is intrinsically invariant to similarity transformations (Triangle Similarity Quadruple (TSQ) algorithm).

Amat et al. [APDP08] present a new method of data hiding in 3D objects. This method is based on a 3D model represented by a cloud of vertices and a list of edges corresponding to the triangular mesh of the surface. The main idea of this method is to find and to synchronize particular areas that can be used to embed the message. The data hiding relies on the modification of the topology of edges in chosen areas.

Schemes modifying the connectivity

Actually, there are very few 3D meshes watermarking techniques based on connectivity modification. This kind of watermark is obviously fragile to connectivity attacks.

2.4.3 Transform techniques

Most of the successful image watermarking algorithms are done in transform domains. A better imperceptibility can be gained thanks to the "spread spectrum" principle. Unfortunately, for 3D meshes, the lack of a natural parameterization makes spectral analysis even more difficult. So almost all the existing tools have their limitations. Besides the algorithms that embed watermarks in the spectrum obtained by a *direct frequency analysis*, we also present here the class of algorithms that are based on *multiresolution analysis*. The basic idea behind both of them is the same: modification of some spectral-like coefficients.

Watermarking Based on Direct Frequency Analysis

In this field, researchers have tried different types of basic functions for this direct frequency analysis. Based on the Laplacian basis functions, Ohbuchi *et al.* [OT] proposed a non-blind method (additive modulation of the low and median frequency coefficients) while Cayre *et al.* [CAS⁺] gave a semi-blind one (quantization of the low and median frequency coefficients). Two problems exist in this basic function: (1) The computation time increases rapidly with increasing mesh complexity. (2) This analysis depends on the mesh connectivity information. Wu and Kobbelt [WK] reported an better algorithm that is based on radial basis functions.

Embed a watermark in the different resolutions

This technique is based on multiresolution analysis which is a useful tool to reach an acceptable trade-off between the mesh complexity and the capacity of the available resources. Such an analysis produces a rough mesh which represents the basic shape (low frequencies/resolution) and a set of details information at different resolution levels (median and high frequencies/resolution). In this way, the multiresolution allows to adapt the various application demands in the different available locations. The wavelets are a common tool for such a multiresolution analysis. Fig. 2.7 shows the wavelet decomposition of model feline. The watermark can be inserted either in the coarsest mesh, or in the wavelet coefficients at different levels. But this kind of wavelet analysis is applicable only on semi-regular triangular meshes⁴. Based on this wavelet analysis, Kanai *et al.* [KDK] proposed a non-blind algorithm that modifies the ratio between a wavelet coefficient norm and the length of its support edge, which is invariant to similarity transformations. Uccheddu *et al.* [UCB] described a blind one-bit watermarking algorithm with the hypothesis of the statistical independence. Thanks to a remeshing step, the above analysis could be extended to irregular meshes. With this idea, Cho *et al.* [CLLP] extended the algorithm in the wavelet domain. Other multiresolution analysis tools, such as the edge-collapse iterations technique [PHF] and the Burt-Adelson pyramid decomposition [YPZ], are employed to develop robust 3D mesh watermarking algorithms.

The watermarking in the low resolution Embedding data in low resolution can be both more robust and more distortion. But wavelet transformation from low resolution to original model can make the distortion more imperceptible thanks to a dilution effect.

The watermarking in the high resolution Embedding in the detail parts provides an excellent capacity. Embedding data in high resolution level may permit to construct some effective fragile watermarks with a precise localization ability of the attacks.

⁴In semi-regular triangle mesh, valence of most vertices is six

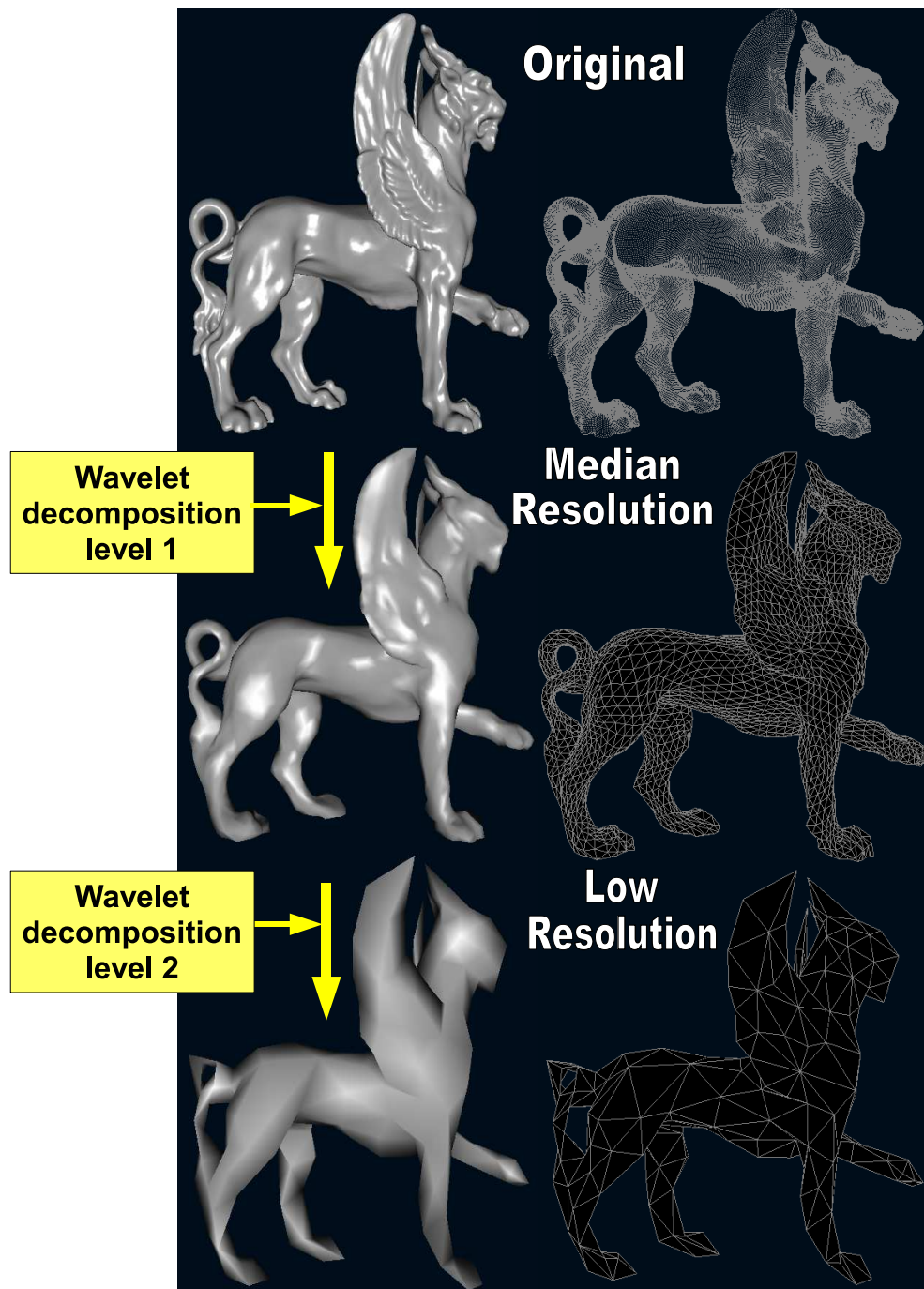


Figure 2.7: Wavelet analysis

Chapter 3

The developed method for 3D watermarking

As we presented in chapter 2, our method is based on the algorithm of Cayre and Macq. Because this is a blind scheme which is secure and fully automatic. In addition, a high-capacity 3D watermarking can be obtained, and his scheme is robust against translation, rotation, and scaling operation. In following section 3.1, we will detail the Algorithm of Cayre and Macq for 3D watermarking. In section 3.2, we present our Algorithm. Then we will discuss the experimental result that we have obtained in section 3.3. Moreover there is a small conclusion in section chapter.

3.1 Algorithm of Cayre and Macq for 3D watermarking

Algorithm of Cayre and Macq is based on a substitutive blind procedure in the spatial domain. From a geometrical point of view, one could see this scheme as a QIM¹ [Cheb] scheme extended on a discrete partition of a physical measurement. The basic idea behind the TSPS² algorithm is the insertion of bits while moving on the mesh. The key idea is to consider a triangle as a two-states geometrical object and choose the projection of a vertex on its opposite edge in a triangle as the primitive. Then we discuss our algorithm in section 3.2.

3.1.1 Generation of a secret key

The first step: a list of triangles of the mesh that will be watermarked is established. This operation is driven by a secret key.

¹quantization index modulation

²triangle strip peeling sequence

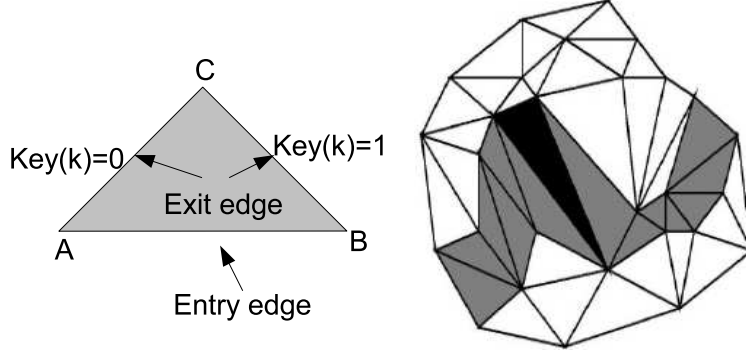


Figure 3.1: Generation of the triangle list

In the method of Cayre and Macq, they keep the basic TSPS idea. The list of triangles is established according to the scheme as illustrated in Fig. 3.1. Once a starting triangle is determined, the next triangle in the list is either the first³ or the second one⁴ in clockwise order, depending on the bit value of the key. The length of the key must be as long as the list of admissible triangles required watermarking bits. The key may also be the seed of a Pseudo-Random Number generator (PRNG).

3.1.2 Embedding of information

In the second step, the watermarking bits are embedded in each triangle which is in the list built in the first step.

Each triangle is considered as a two-state object, the state of the triangle is defined by the position of the orthogonal projection of the vertex C on the entry edge AB. Edge AB is divided into two subsets S_0 and S_1 by a symmetry axis of AB. If this position of the projection $P(C) \in S_0$, then we consider that the triangle is in a "0" state; otherwise, $P(C) \in S_1$, and the triangle is in a "1" state. The states are invariant to an affine transform of the meshes. To set the triangle in the i ($i=0$ or 1) state, there are two cases:

- a) $P(C) \in S_i$: C's position must not be modified.
- b) $P(C) \notin S_i$: C has to be shifted toward C' to make $P(C') \in S_i$.

The two keys for changing vertex position from C to C':

- a) $|C - C'|$ has to be small enough to avoid visual distortion of the mesh.
- b) The $C \rightarrow C'$ mapping is a symmetry across the closest axis orthogonal to AB that intersects sub-interval belonging to S_i .

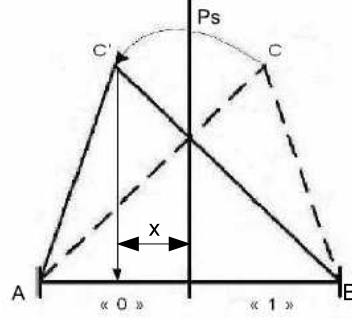


Figure 3.2: First-order (macro embedding procedure)MEP [OM] ($n=1$). Two geometrical configurations. Opposite edge AB is divided in two intervals

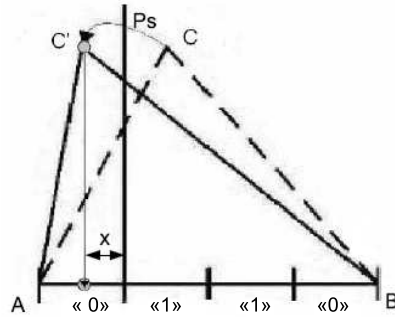


Figure 3.3: second-order MEP ($n=2$). Opposite edge AB is divided in four intervals. Geometrical distortion is decreased

See also an example from Fig. 3.2 and Fig. 3.3, the technique of Cayre and Macq for 3D watermarking (The projection is moved to the nearest correct interval, the inserted bits are both "0").

With this method, the triangle can be geometrically modified or not, depending on the difference of the bit value to be hidden and the initial state of the triangle.

Partition on the edge AB A graphical summary of the partition is shown in Fig. 3.4. The aim of this partition is to extend the QIM concept to 3D triangle meshes [CW].

Projection of vertex C on the Quantized Base In this paragraph, the planes/axis (Ps) are defined, which are all orthogonal to AB, (See the Fig. 3.2 and Fig. 3.3). All these axes are placed on the center of every

³its new entry edge is AC

⁴its new entry edge is BC

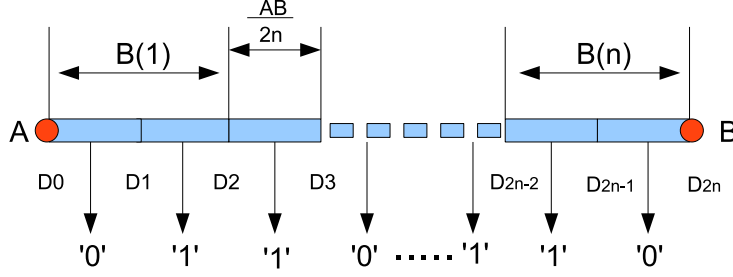


Figure 3.4: Decomposition of the entry edge AB into two interleaved subsets S_0 and S_1 , with the $2n$ binary values for every D_k, D_{k+1}

$B(k)$, $1 \leq k \leq n$, then all the distances X_k from vertex C to all the (P_s) are calculated, and let $X = \min(X_k)$, thus X 's boundaries $0 \leq X \leq |AB|/n$. The relationship between X and $B(k)$ is straightforward, as the projection of C selects both a $B(k)$ and a interval $[D_p, D_{p+1}]$ of $B(k)$, $2k-2 \leq p \leq 2k-1$. X represents the minimal amount of geometrical distortion to be introduced for changing the state from C to C' . Thus n represents the parameter of the smoothness in this algorithm. As n increases, X decreases, moreover the amount of distortion to be caused gets smaller. On the other hand, the number of frontiers between domains increases, that can cause bit retrieval errors due to the limited machine precision.

Modify the position of the vertex C in the triangle Keeping this frame in mind, the MEP⁵ is defined to be the symmetry of C with respect to (P_s) . If the triangle is in the correct state, nothing will be moved, so no geometrical distortion is introduced; otherwise, only a small reversible shift is performed, the summit C' is always moved on the beeline which parallels AB and cross the summit C , and the moving distance equal to $2 \times X$.

advantage and shortcoming Theoretically, this method can decrease distortion of watermarking to minimum. Actually, we must face much difficulty to realize this method on program. It is also limited by complexity and runtime. Another problem is the generation of secret key. This algorithm depend on its secret key so much, if there exist an error in the secret key, it will bring us a great difficulty in extraction step. We must think about how a secret key can lead a right list of triangle, namely if there is no triangle appear two times in the list, and so on. Thanks to all of these reasons, we decide to research another schema for our watermarking.

⁵Macro embedding procedure

3.2 Our Algorithm

As we described in chapter 2, the digital 3D watermarking consists three step :

- a) The embedding part.
- b) The extraction part.
- c) The analyze part.

Here, we remain the basic idea of Cayre and Macq and also make some changes according to our practical situation.

3.2.1 Transforming the information from ASCII to binary

Actually, our program for watermarking is just fit for insertion of the binary information, but this type of the information is not understandable for a normal human, so the transformation of the information from ASCII⁶ to binary is necessary, as the first step. And the inverse process will be used in the part of the extracting watermark from the watermarked 3D model.

In general, the ASCII information means these 128 characters defined in the norm ASCII. See a given example in Fig. 3.5. Every character occupy 8 bits, and each bit can be embedded in only one triangle. But in our method, about 15 percent of the triangles can be embedded a bit in the 3D mesh.

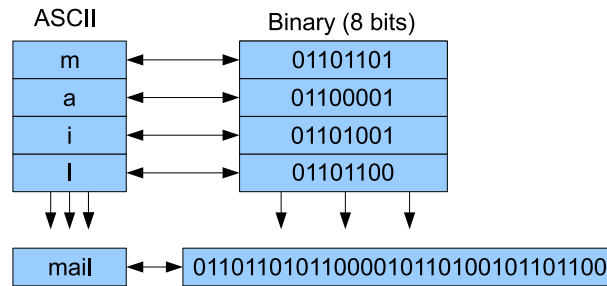


Figure 3.5: An example of coding

3.2.2 Embed the information to 3D meshes

Once, we finished the coding as a pretreatment, and the procedure of the insertion begin.

⁶American Standard Code for Information Interchange

Establishing the triangle for watermarking

In the step, setting a secret key and establishing the triangle are finished. But we use one way which is different from the method of Cayre and Macq for simplifying the computing process.

First of all, we need to input a file of 3D object with an extension (.stl), which includes all the information of triangle and point as a data base, all the triangles and the vertices are ranked from the first to the last one, respectively. The constraints for establishing the triangle are:

- Each triangle can be used only one time for watermarking.
- Once a triangle was used(in that case, there is at most one vertex was modified), the three vertices of this triangle can not be used for the next watermarking process.
- The embedded bits must be less than or equal to the capacity ⁷of the current mesh.

In this way, a list of triangles of the mesh that can be watermarked is established automatically in the process of the watermarking. In this list, there exist no common point in all the triangles. This operation is driven by searching the new triangles with the three constraints in the ranking of the triangles. But there is only one list for each mesh object. The first 16 triangles of this list are reserved to record the quantity of the information to watermark. Although, usually the constraints reduce the capacity of the watermarking, they help us to extract the information accurately from the watermarked 3D models. So it is considered a very important role in our algorithm.

At the same time, a secret key can be generated by random selecting the triangles in that list of triangle without triangle repetition. The process of the watermarking can be driven by this secret key with security. The size of the secret key must equal to the number of the watermarking bits. (But in our training, using the secret key is overleaped)

So without the secret key, the binary information will be embedded into the triangles according to one order of this list of triangle until all the bits are inserted into the mesh object. And in our method, the capacity is depend on:

- the amount of the triangles,

⁷The capacity of the current mesh equal to the number of the triangle which can be watermarked with the constraints a) and b) in our method.

- the amount of the vertices,
- the geometrical structure of precessed 3D object.

Partition of the Edge AB

In this section, we simplify the partition and quantization of the algorithm of Cayre and Macq. We still consider a triangle as a two-states object, the state of the triangle is always defined by the position of the orthogonal projection of the vertex C on the AB edge. For keeping an equilibration between the robustness and the distortion, we decide to divide the AB edge by 4 partitions and extend their partition of the AB edge to the whole axis, in this way, the projections of C can be always handled, even though it falls out of the AB edge. The vertex A is defined as an origin in this axis (The vertex B is always put on the right side of vertex A). The binary values are given as we see in Fig. 3.6. The alternation numbering "...0,1,0,1,0" and "0,1,0,1,0..." are suitable for decreasing distortion of watermarking.

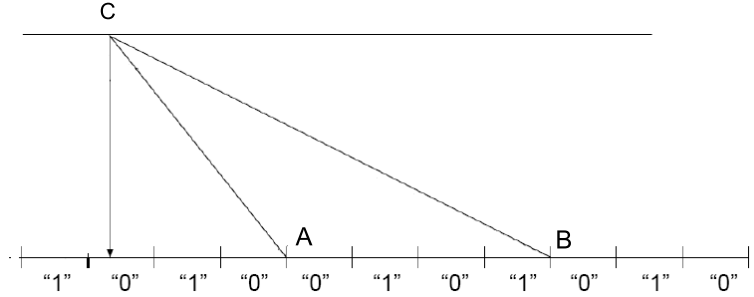


Figure 3.6: Quantized partition with the alternation numbering

Projection of the vertex C on the quantized base

The projection of C on the axis which defined by AB will select a partition which present the initial value of the triangle. This value of the triangle will be regarded as default bit for watermarking the binary information. If the triangle is in the correct state (the default value of the triangle equals to the current bit which will be embedded), we must not move the position of the vertex C for changing its projection, so there is no geometrical distortion; otherwise, we must change the position of C to adapt the current embedding bit.

Modify the position of the vertex C for changing the initial value of the triangle

Keeping this frame in mind, according to value of current embedding bit , we let the vertex C' to be always moved on the beeline which parallels AB and cross the summit C. The destination of movement: when its projection is in the center of the next partition on the right side, if the original projection of vertex C is on the right side of point A. Otherwise , it will be in the center of the next partition on left side. This movement is for changing the initial value of current triangle. And the moving distance or distortion depend on the actual projection of summit C, ($0.5 * L < distance < 2 * L$, where L is the length of one partition in current triangle.) Fig. 3.7 describes the different movement for dealing with the different triangles. In general, there is four basic type of the movement.

Algorithm for embedding watermark

In Fig. 3.8 the pseudo-code of insertion algorithm is shown.

3.2.3 Extract the information from 3D object

The extraction of the embedded information is an easy process, if we have finished information embedding. Because we will use the same method for establishing the list of the triangles ,the same partition of the AB edge and the same method for giving a default value to each triangle of this list in the extraction process. Furthermore there is no step for moving the summits of the triangle. We just need retrieve the default value of useful triangle in the list.

The first step, input a watermarked 3D model, then we profit the default value of first 16 triangles of the list for computing the quantity of the watermarked bits. These first 16 bits are embedded in the embedding process for storing the quantity of the information to watermark.

The second step, extract the default value stating from the seventeenth triangle in the list according to the quantity of the watermarked bits.

The third step, we collect all of these bits one by one, and transform these binary information to ASCII form.

algorithm for retrieving the watermark

In Fig. 3.9 the pseudo-code of extraction algorithm is shown.

3.3 Experimental results

After the experiment, we have some results to show you. We have tested our algorithm on different 3D objects. The execution time varies along with the complexity of the input mesh. The complexity of different meshes are illustrated in Fig. 3.10. Because the capacity of "felineC" which is a model in low resolution is 69 bits. For adapting the three different model, we set only 5 watermarking characters as well as 40 bits, and additive 16 bits for storing the quantity of the watermark, the final sum is 56 bits. The runtime of the insertion procedure is 0.5 ,11 and 18 seconds for the models "felineC", "forme" and "Kara", respectively , Then, at the seconde time, we inserted 556 bits, and the runtime is 13 and 18 seconds for the models "forme" and "kara", respectively. At the third time, 2128 bits were inserted as a watermark, the runtime is 54 and 36 seconds for "forme" and "kara", respectively. All of these tests are based on a computer with a 3.0 GHz Intel Pentium mobile processor and 512 MB RAM. Thus we can conclude that our watermarking runtime is related to not only the number of the points, but also the structure of the model and watermarking quantity.

3.3.1 Capacity

Our algorithm belong to the data hiding applications, so in this case, the maximin capacity is directly lies on the structure of the mesh, the number of points or faces in the mesh. The Fig. 3.10 presents the different results about capacity, where $Embeddingfactor = \frac{payload}{vertices}$. Our algorithm is easy to manipulate, but in general, we have less capacity for the watermarks than algorithm of Cayre and Macq, namely our embedding factor is lower.

3.3.2 Imperceptibility

Examples for distortion In general, the SVH is very sensitive to perturbations of a surface smoothness by random additive noise. An example is illustrated in Fig. 3.11, there exist a obvious distortion in the watermarked model, but this distortion can not be avoided, and it is acceptable. Actually, once the surface is added on this watermarked mesh, this type of distortion is invisible.

The method for reducing the distortion of watermarking

Searching the minimal distance X In the Algorithm of Cayre and Macq, X represents the minimal amount of geometrical distortion by changing the state of summit C, see the detail in chapter 3.1. This method can also be used in our training, but due to limited time, we used a simpler method which was referred in section 3.2. So, to test its imperceptibility, we will do this work in the future.

Increasing the number of the partition on the AB edge can reduce the distortion, but decrease robustness. An example is illustrated in Fig. 3.12. If the partition is more frequent, on the one hand the summit C will be moved with smaller distance for modifying the value(state) of the triangle, on the other hand while the mesh suffer some noise in the channel of digital communications, the distortion is introduced, namely some positions of the points were changed, and the default value of the triangle will be changed more easily in the smaller partition.

Dealing with big triangle is a good idea for decreasing the distortion. In the mesh, there exist usually very different size of triangle, if we use the same method to deal with it, sometimes, some strong distortion will be introduced. As we see in Fig. 3.13, the original model is watermarked by two different ways. Certainly, we have two different result. The watermarked model(1) is generated without dealing with big triangles. The watermarked model(2) is generated with dealing with big triangles. In the model(1), there is a so strong distortion, why? Let's analyze broad principle.

A introduced distortion comparison from the big triangle and normal triangle is shown in Fig. 3.14. Two red triangles are a normal triangle and a grand triangle, respectively. If we watermark one bit to each of two triangles, the vertex C of the big triangle is moved much farther than the normal triangle's, so it introduces much more distortion. Base on this viewpoint, we will detail two methods for dealing with big triangle, as follows

Method (a): Give up to watermark big triangle.

Actually, once we input 3D model into our program, the program distributes automatically a name(A or B or C) to each vertex of every triangle. And we always project the summit C on AB edge for following calculation. So here, triangle which has the longer AB edge is defined as a big triangle. Since intensity of distortion depends on length of AB edge not its volume or something else. For finding big triangle, we calculate an average length of AB edge (L) in mesh. If ($AB > 2 * L$), where AB present the length of AB edge of current triangle, we define this triangle as a big triangle, and delete it from watermarking triangle list. This is a effective method for decreasing distortion. However, deleting big triangle from the list will decrease capacity of mesh.

Method (b): define the shortest edge as AB edge in triangle.

When we input 3D model into our program, three points of triangle are random named as A, B, C. In this method, we must rename three points of triangle, namely the shortest edge is named AB in triangle. In this way, we can limit the distortion and remain capacity. For example, it is fit for

the Model form.stl. In Fig. 3.14, in the big red triangle, if the shortest edge is named AB edge, in fact its watermarking distortion intensity is like a normal triangle. But there is always some exceptions, for instance, we can not treat with a big equilateral triangle by this method. In addition, thanks to rename three point of triangle, we will meet a difficulty in extraction step.

The distortion can be decreased but can not be avoided in the 3D watermarking process, moreover the quality of watermarking techniques lies on the distortion. Our aim is embedding the watermark, but making it invisible. In this case, the watermarked mesh will no longer be evaluated only by the human visual system. There are several proposals to address the issue of measuring distortion for 3D watermarking. The simplest is the Hausdorff distance, which is based on an infinite norm. Another proposal was made by Karni and Gotsman[KG] to handle meshes with different topologies. Because in our training, we do not change the topology, we use the Hausdorff distance to evaluate the distortion caused by the insertion process.

So we use "metro" which is a software to measure the distances(include the Hausdorff distance) between the original models and the cover models.

3.3.3 Robustness

The robustness of 3D object watermarking is very different from 2D media. In the general framework of 3D watermarking, embedding in the spatial domain naturally leads to a relatively poor robustness, whereas capacity increases. Transform(spectral) domains have shown to offer a better robustness. Our algorithm is defined in the spatial domain. Thanks to the limitation of the time of training, so we just tested robustness of our algorithm in the simulation of the noise.

Some proposed methods for increasing the robustness in this training

The alternating numbering like ' 0, 1 ' and ' 1, 0 ' of the B(k) is most fit for the robust watermarking, see Fig. 3.4 in detail.

The number of frontiers between the partition increases, it can cause the bit retrieval error due to the limited machine precision, at the same time, it can bring a fragile watermark, which can not face to noise. So we only considered the cases when the number of intervals is 2 or 4. See also Fig. 3.2 and Fig. 3.3.

Watermarking on the low resolution object is a great idea, but the procedure of wavelet decomposition is very complex from the original model to low resolution model and inverse. During my training, we have tried to find a software to execute this decomposition, but we had no chance. Moreover, the watermarking on the low resolution object is also limited by the capacity constraint. Since in our implementation, the capacity is related to the number of the triangle and vertex on the meshes, however the low resolution model is a simplification with few primitive.

The simulation of noise for testing the robustness

Robust watermarking should provide the reliable communication of a message in a 3D model content. Various intentional and unintentional attacks constitute watermarking channel. And the noise is a broad unintentional attacks in the transportation channel, so we did this simulation of noising attack.

For simulating this process of testing, we add the white noise to our watermarked 3D model. This kind of operations attacks the watermark by adding white noise into the coordinates of vertices. So in our implementation, the subprogram ("bruiteur") effects this simulation of noise according to the original size⁸ of the 3D model. About this subprogram, we input a watermarked 3D model, the output are some models with the different noise levels⁹. Actually, we change the three coordinate x, y, z for every point, this noising distance is present by $L * P * R$, where L is the length of the diagonal of model. P present a percentage which we need according to the length of the diagonal, so $P \in (0, 1)$ and the noise intensity depends on this parameter P. R is one random number of $\{-1, -0.9, \dots -0.1, 0, 0.1, \dots, 1\}$, and its function is making the simulation closer to the real channel. Now, we can test the robustness in the different noise levels.

Here, there are some results for the model "Kara" and "felineC" about this test. As we presented, the capacities are 2645 bits and 69 bits for "Kara" and "felineC", respectively. In Fig. 3.15, the value of P(noise intensity) is set in $[0.00005, 0.1]$ according to the our practical state in this test.

"Kara" as a model original, so we decided to insert 552 bits to this model namely "duanmaoyu@gmail.com, gerard.subsol@lirmm.fr, william.puech@lirmm.fr". The noise intensity P is starting from 0.00005, and error rate has already existed in this case. This percentage begin to increase with an augmentation of noise intensity until $P = 0.0005$. Then the the error rate will be between 40 and 60 percent.

⁸Usually, we use the length of diagonal of 3D model for measuring size

⁹This level depends on the percentage of the size of the model.

"FelineC" is a model on low resolution, so according to its capacity 69 bits, we decide to set 48 bits as a watermark, as well as "Wpuech". We can see in the seconde curve, when the noise intensity P is on the interval $[0.00005, 0.0003]$, there is non error bits. This result prove that the low resolution model felineC is more robust against the noising attack. And in the interval $[0.0004, 0.01]$, error percentage begin to raising and arrive to 50 percent. Then it is similar to the Kara's curve, so nothing special.

According to raise the noise intensity(based on the diagonal of model), the error rate increases from 0 to round about 50. Here, there exist a problem about natural probability. Because we always have 50 percent chance to receive a correct bit in a binary case, once this error rate is raised to round about 50, the it will stop raising, even though we continue to augment the rate of the noise. That's a reason why finally, error rate will rest round about 50 percent with increasing noise intensity.

In the process of the extraction ,when the noise is strong enough, it can change some default values of the triangle. once the default values of the first 16 triangles are changed by the noise, it can cause a error in this process, since these values of first 16 bits store a quantity of the watermark. If this figure(the quantity of the watermark) is not accurate, sometimes the extracted information will be completely false. So to deal with this problem is also our task in the future.

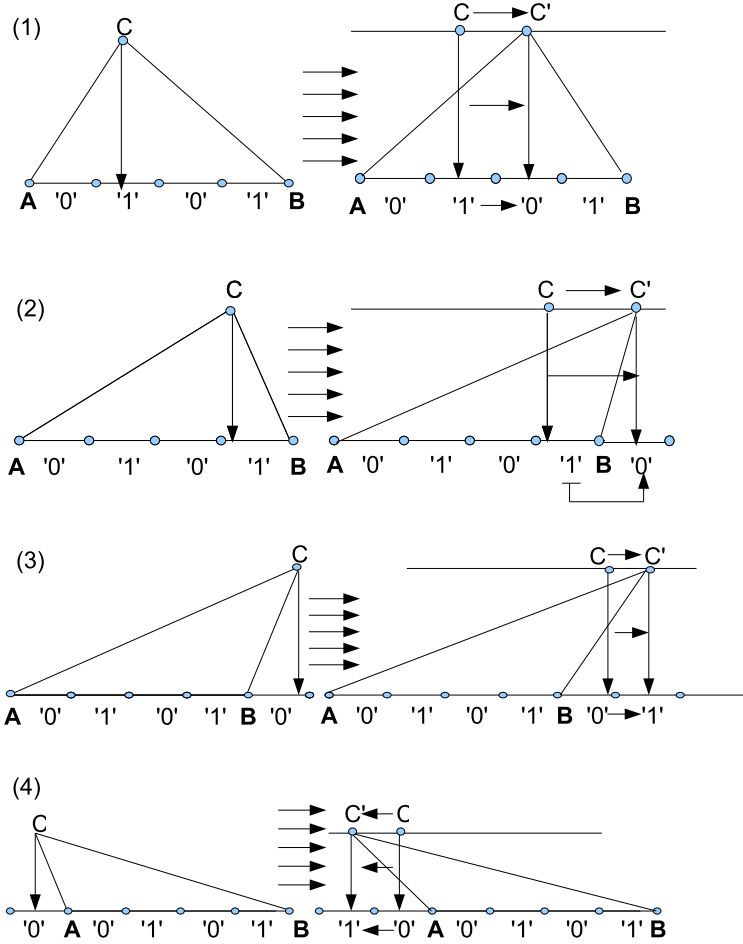


Figure 3.7: In the triangle 1 and 2, the original projections of the vertex C fall into the edge AB. In the triangle 3 and 4, their original projections of the vertex C is outside of the edge AB. We change the default value of the triangles 1 and 2 from '1' to '0', and change the default value of the triangles 3 and 4 from '0' to '1'. The vertices C of the triangle 1, 2 and 3 were moved to next partition on the right side, the vertices C in the triangle 4 was moved to next partition on the left side.

Algorithm: high capacity and blind watermarking method

Insertion step

Input: 1. a 3D triangle mesh model (extension: stl)
2. a watermark file which include all the watermarking information (extension: txt)

Output: 1. a watermarked model (extension: stl)
2. a file including the watermarking information in the binary form (extension: txt)
3. a number of the triangles AND a number of the vetices
4. capacity of current model
5. a quantity of watermarking bits
6. a runtime

Main part:

1. Transform ASCII watermarking information to binary format AND calculate the quantity (Q) of the watermark.
2. Establish a list of triangle for storing watermarks AND calculate the capacity (Ca) of the current 3D model.
3. The watermarking step :
 - if ($Q < Ca$)
 - {
 - a. Distribute a default value (V_j) to each triangle in this list according to projection of summit C on the edge AB. ($0 < j < n$)
 - b. for each bit (W_i) of binary watermarking information ($0 < i < m, m < n$)
 - {
 - Compare W_i with V_i ,
 - If ($W_i = V_i$)
 - Do nothing.
 - Else
 - Change the position of the summit C for modifying the default value into W_i .
4. Return a modified 3D model.

Figure 3.8: Watermarking algorithm - insertion

Algorithm: high capacity and blind watermarking method

Extraction step

Input: 1. a watermarked 3D model (extension: stl)

Output: 1. a file including the watermarking information in binary form (extension: txt)
 2. a file including the watermarking information in ASCII format (extension: txt)
 3. a quantity of watermarking bits
 4. a runtime

Main part:

1. Establish a list of triangle for extracting watermarks.
 2. Compute the quantity of the watermarking
 3. Distribute a default value (0 or 1) to each triangle in this list according to projection of summit C on the edge AB according to the quantity of the watermarking.
 4. Transformer these binary values of triangles into ASCII format and return it.
-

Figure 3.9: Watermarking algorithm - Extraction

Results (performed on 3 models)				
Cover content	triangles	Vertices	payload	Embedding factor bit/vertex
felineC.stl	504	250	69 bits	0,276
forme.stl	16128	8020	2645 bits	0,3298
Kara.stl	25208	12606	3190 bits	0,253

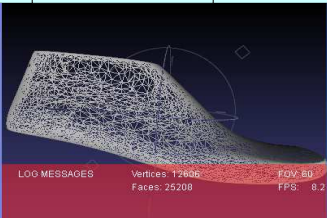
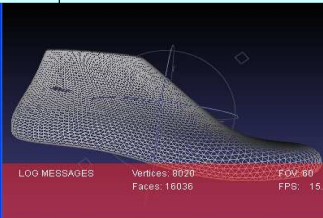
 LOG MESSAGES Vertices: 250 Faces: 504 FOV: 60 FPS: 7.9 felineC.stl	 LOG MESSAGES Vertices: 12606 Faces: 25208 FOV: 60 FPS: 8.2 Kara.stl	 LOG MESSAGES Vertices: 8020 Faces: 16036 FOV: 60 FPS: 15 forme.stl
--	---	---

Figure 3.10: Capacities for watermarking three different models without using the secret key.

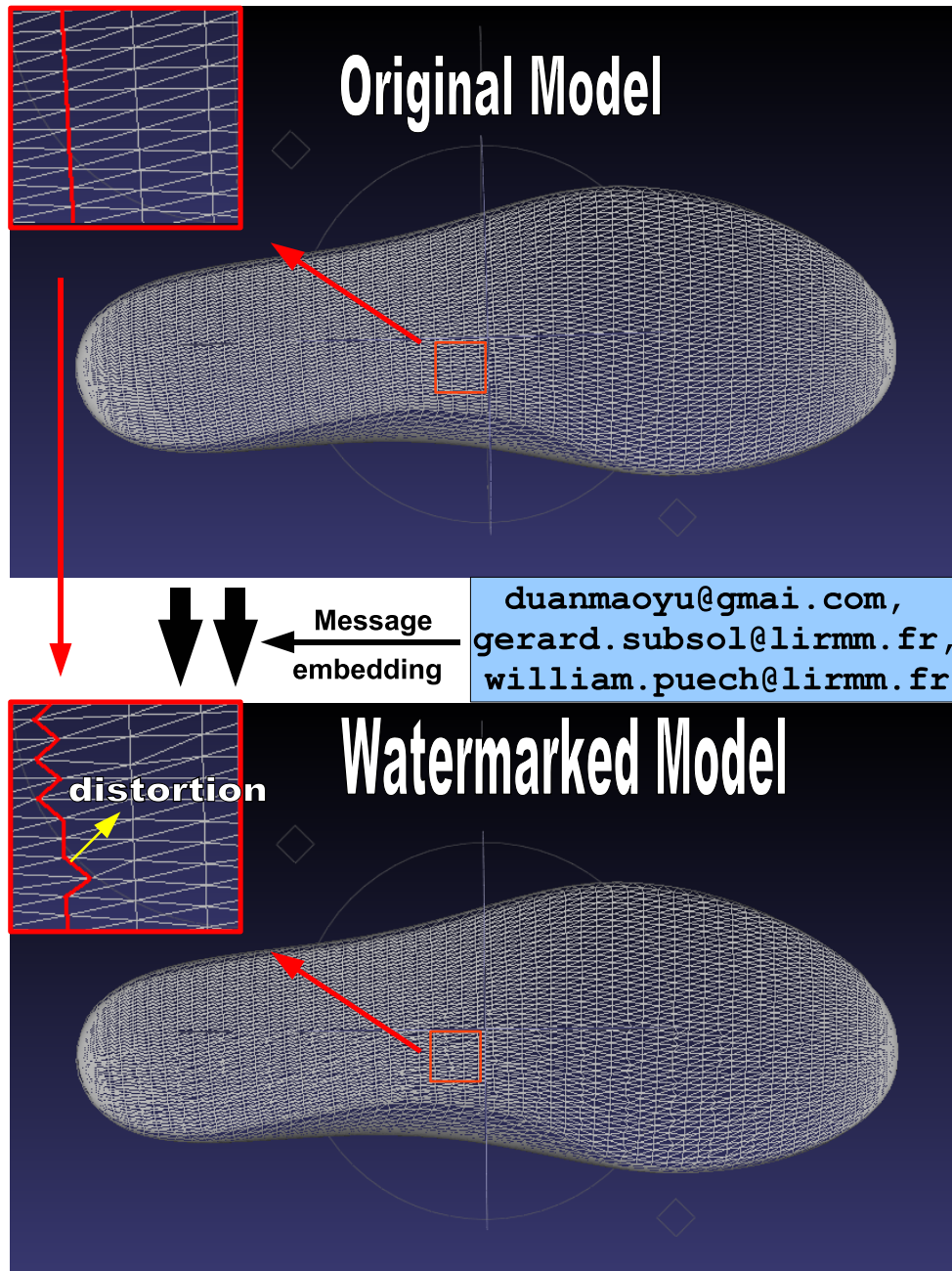


Figure 3.11: A distortion is generated in 3D watermarking process, a message of 552 bits is embedded in model "forme.stl". The Hausdorff distance is 0.118536 between the original model and the watermarked model.

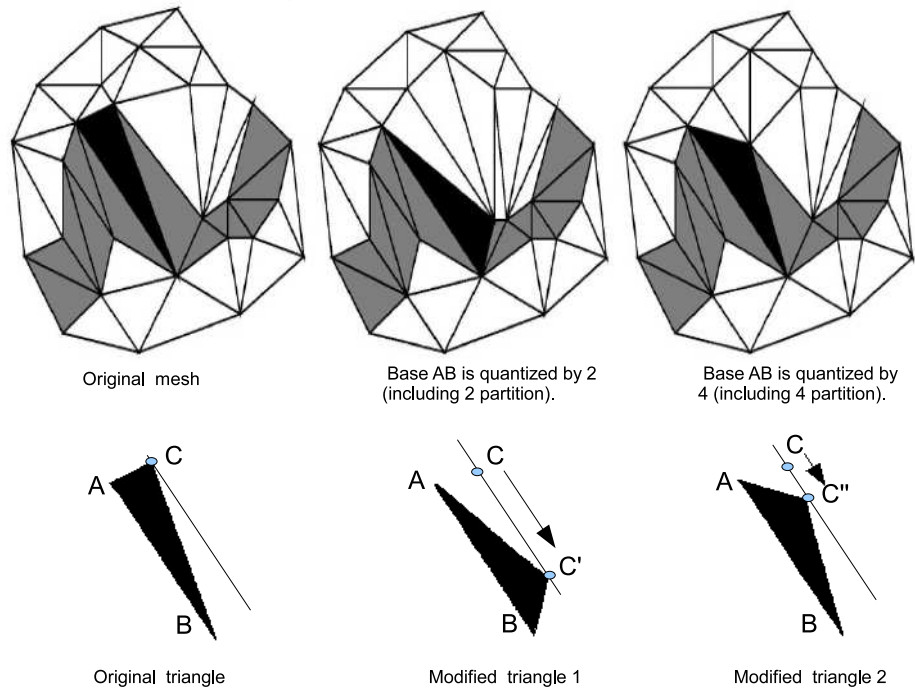


Figure 3.12: In this figure, we can see that the different distortion of the mesh is introduced by the different amount of the partition. The distance between C and C' is larger than between C and C'' , so the modified *triangle1* contain bigger distortion than the modified *triangle2*.

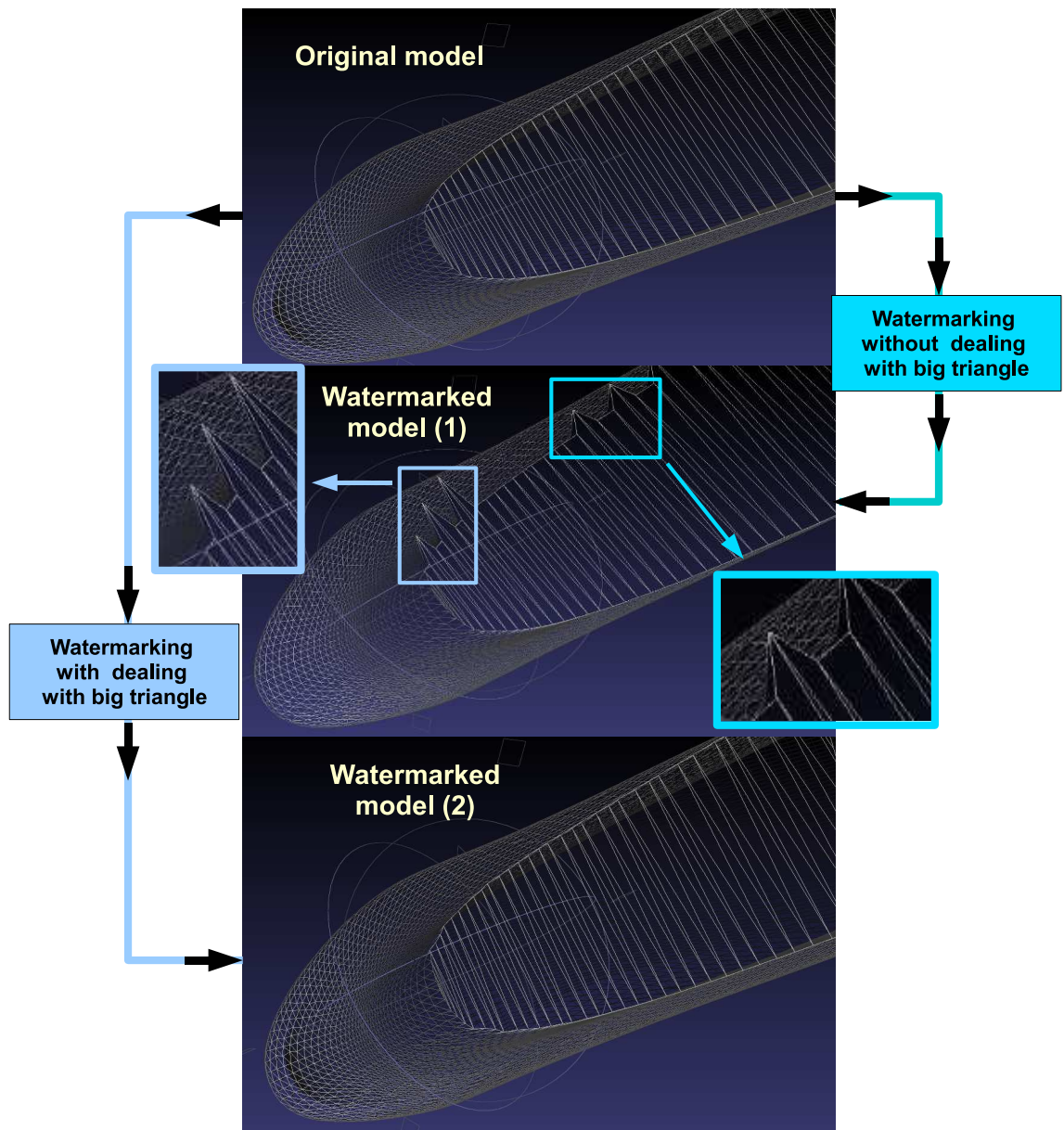


Figure 3.13: Strong distortion is avoided by dealing with big triangle. Exemplar model: forme.stl

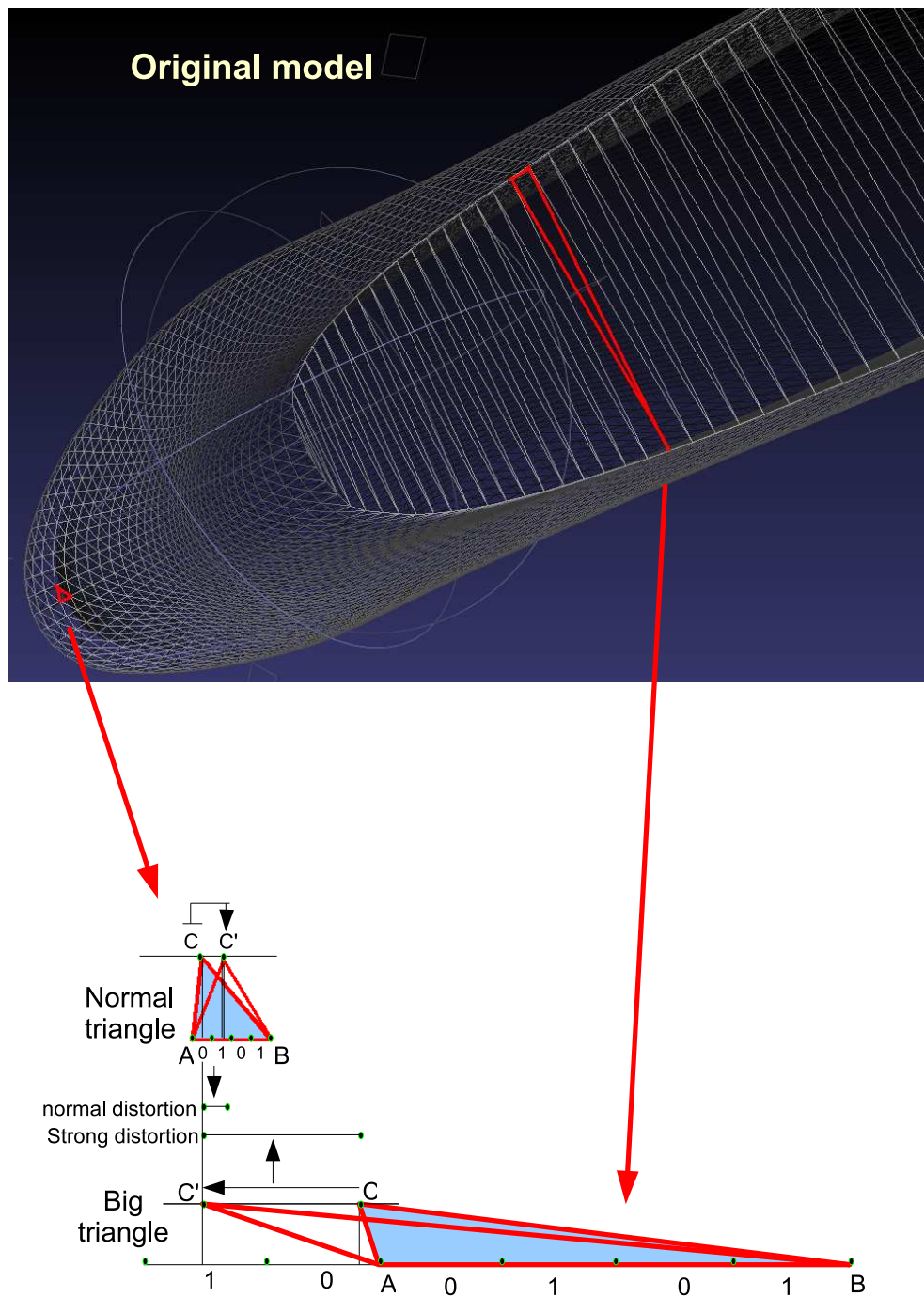
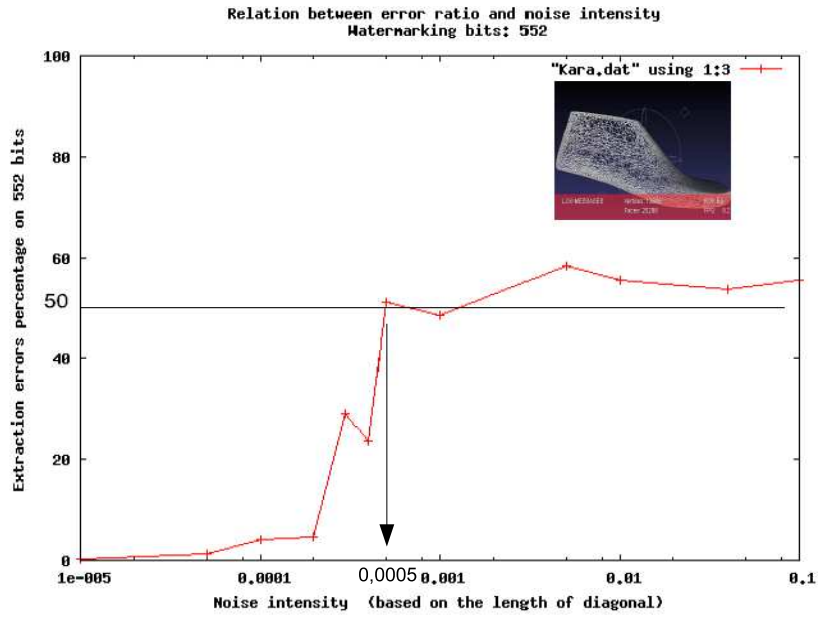


Figure 3.14: Different distortion is introduced by different triangle. Exemplar model: forme.stl

(1) Model Kara (original)



(2) Model felineC (in lowest resolution)

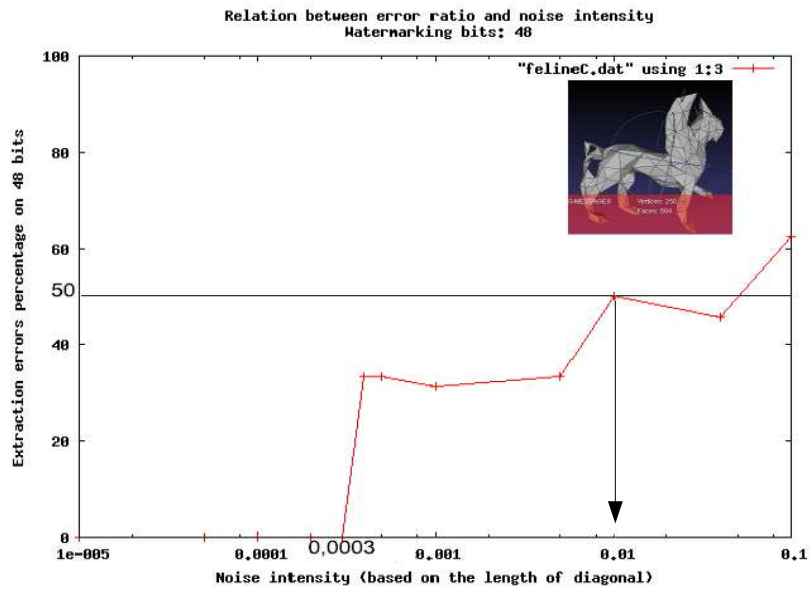


Figure 3.15: Comparison of the robustness on error percentage

Chapter 4

Conclusion and perspectives

In this training, we have reported a new, easy, universal and blind method for 3D triangle mesh watermarking. Normally it is fit for all the irregular triangle mesh models. It can be used with a secret key or without a secret key according to a requirement of user. It also can be used for watermarking of low resolution model or high resolution model with a wavelet analysis. Watermarked 3D model may be robust or high capacity, the capacity and robustness in our method depend on the number of the triangles and vertices as well as structure of current mesh. Imperceptibility is kept very well by dealing with big triangle. In the extracting step, error rate is 0 percent without any noising attack.

At the beginning of the training, hierarchical 3D watermarking is considered as a basic method for our training, finally we didn't find a suitable solution for wavelet decomposition with the limitation of the time, so it was given up. But i would like to develop this method in the further work .

In Algorithm of Cayre and Macq, X represents the minimal amount of geometrical distortion by changing the state of summit C , see the detail in chapter 3.1.2. This method can also be used in our training, but due to limited time, we will test its distortion in the future.

In the extraction step, the first 16 bits that store the watermarking quantity are very important for us, so we must make it robust for noising attack. In this case we can make a mixing partition in a watermarking process, namely, we divide the each triangle edge AB in 2 intervals(partitions) for embedding and extracting the first 16 bits, and we divide the edge AB in 4 intervals for embedding and extracting the other bits. In this ways, the first 16 bits(watermarking quantity) will be extracted more accurately, thus extracting process can be better ensure, and the more distortion is introduced

just in the 16 triangles. So in the further work we will also add this method in our program for testing.

Bibliography

- [APDP08] P. Amat, W. Puech, S. Druon, and J. P. Pedeboy. Data Hiding Method Based on MST and the Topology Change of a 3D Triangular Mesh. *Journal Signal Processing: Image Communication*, 2008.
- [Ben] O. Benedens. Two High Capacity Methods for Embedding Public Watermarks into 3D Polygonal Models. Proceedings of the Multimedia and Security-Workshop at ACM Multimedia, 1999, pp. 95-99.
- [Bor] A.G. Bors. Watermarking Mesh-Based Representations of 3-D Objects Using Local Moments. *IEEE Transactions on Image Processing*, vol. 15, no. 3, 2006, pp. 687-701.
- [CAS⁺] F. Cayre, P.R. Alface, F. Schmitt, B. Macq, and H. Maitre. Application of spectral decomposition to compression and watermarking of 3d triangle mesh geometry. *Signal Processing*, vol. 18, no. 4, 2003, pp. 309-319.
- [Cer] Certimark. www.certimark.org/.
- [Chea] Checkmark. www.watermarking.unige.ch/checkmark/.
- [Cheb] G. W. Chen, B. and Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. submitted to *IEEE Trans. on Information Theory*, 1999.
- [CLLP] W.H. Cho, M.E. Lee, H. Lim, and S.Y. Park. Watermarking technique for authentication of 3-d polygonal meshes. Proceedings of International Workshop on Digital Watermarking 2005, pp. 259-270.
- [CM] F. Cayre and B. Macq. Data Hiding on 3-D Triangle Meshe. *IEEE Transactions on Signal Processing*, vol. 51, no. 4, 2003, pp. 939-949.

- [CW] B. Chen and G.W. Wornell. Quantization Index Modulation: A class of provably good methods for Digital Watermarking and Information Embedding of Multimedia. *IEEE Trans. on Information Theory*, vol. 47, NO. 4, May 2001.
- [DD] K. Deepa and H. Dimitrios. Digital watermarking for telltale tamper proofing and authentication. *Proc. IEEE* vol.87, no.7, pp.1167-1180, July 1999.
- [KDK] S. Kanai, H. Date, and T. Kishinami. Digital watermarking for 3d polygons using multiresolution wavelet decomposition. *Proceedings of IFIP WG*, 1998, pp. 296-307.
- [KG] Z. Karni and C. Gorsman. Spectral compression of mesh geometry. In *Proc. SIGGRAPH*, 2000, pp. 279-286.
- [OM] R. Ohbuchi and M. Masuda, H. and Aono. Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications. *IEEE J. Sel. Areas Commun*, vol.16, no.4, pp.551-560, May 1998.
- [OMA] R. Ohbuchi, H. Masuda, and M. Aono. Data Embedding Algorithms for Geometrical and Non-geometrical Targets in Three-Dimensional Polygonal Models. *Computer Communications*, vol. 21, no. 15, 1998, pp. 1344-1354.
- [Opt] Optimark. www.poseidon.csd.auth.gr/optimark/.
- [OT] A. Ohbuchi, R. Mukaiyama and S. Takahashi. A frequency-domain approach to watermarking 3d shapes. *Computer Graphics Forum*, vol. 21, no. 3, 2002, pp.373-382.
- [PHF] E. Praun, H. Hoppe, and A. Finkelstein. Robust mesh watermarking. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 1999, pp. 49-56.
- [Sti] Stirmark. www.watermarkingworld.org/.
- [TVK⁺] E. Topak, S. Voloshynovskiy, O. Koval, M. K. Mihcak, and T. Pun. Security Analysis of Robust Data-Hiding with Geometrically Structured Codebooks. *Proceedings of SPIE EI: Security, Steganography, and Watermarking of Multimedia Contents VII*, San Jose, CA, January 2005.
- [UCB] F. Ucheddu, M. Corsini, and M. Barni. Wavelet-based blind watermarking of 3d models. *Proceedings of the Multimedia and Security Workshop*, 2004, pp. 143-154.

- [WDB08] G. Wang, K. Lavoue, F. Denis, and A. Baskurt. Three-Dimensional Meshes Watermarking: Review and Attack-Centric Investigation. *Information Hiding*, pages 50–64, 2008.
- [WK] J. Wu and L. Kobbelt. Efficient spectral watermarking of large meshes with orthogonal basis functions. *Visual Computer*, vol. 21, no. 8-10, 2005, pp. 848-857.
- [YIK] Z. Yu, H.H.S. Ip, and L.F. Kwok. A Robust Watermarking Scheme for 3D Triangular Mesh Models. *Pattern Recognition*, vol. 36, no. 11, 2003, pp. 2603-2614.
- [YPZ] K. Yin, J. Pan, Z. Shi, and D. Zhang. Robust mesh watermarking based on multiresolution processing. *Computers and Graphics*, vol. 25, no. 3, 2001, pp. 409-420.
- [YY] B. Yeo and M.M. Yeung. Watermarking 3D Objects for Verification. *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 36-45 1999.