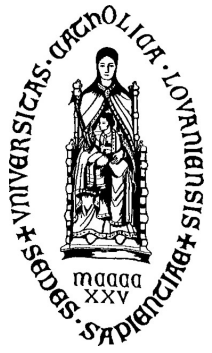


UNIVERSITÉ CATHOLIQUE DE LOUVAIN

ÉCOLE POLYTECHNIQUE DE LOUVAIN

Robust Feature-based 3D Mesh Segmentation and Visual Mask with Application to QIM 3D Watermarking

---

by

Mireia MONTAÑOLA SALES

A Thesis Submitted

In Partial Fulfillment of the Requirements for the Degree

MASTER INGÉNIEUR CIVIL ÉLECTRICIEN

Promoter:

Prof. Benoit MACQ

Academic year 2009-2010

Acknowledgements

It would not have been possible to write this master thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

I would like to extend my thanks to my advisor, Prof. Benoît Macq, for giving me the opportunity and the support to realize this master thesis.

My gratitude also goes to Joachim Giard, Patrice Rondão Alface and Rony Darazi for their support and guidance all through the duration of this thesis work.

I would like to thank my colleagues and friends at the Laboratory, specially Arnab Bhattacharya and Mathieu Van Wambeke, for the good and difficult times shared during the lasts months and for the kind atmosphere.

I am grateful to both University Catholique de Louvain (UCL) and Universitat Politècnica de Catalunya (UPC) for facilitating this Dual Cluster Master's degree program.

I also want to thank my friends from the Iberian Peninsula for their constant encouragement, David, Jonathan, Rubén, Carol, Elena and special mention to one of them who has been a very good friend and support, for sharing his experience, for listening to my complaints and for believing in me.

Finally, I dedicate this thesis to my family, my parents and my sister, for their unconditional support, care and encouragement to pursue my interests, even when the interests go beyond boundaries of language and geography.

ABSTRACT

The last decade has seen the emergence of 3D meshes in industrial, medical and entertainment applications. Many researches, from both the academic and the industrial sectors, have become aware of their intellectual property protection arising with their increasing use.

The context of this master thesis is related to the digital rights management (DRM) issues and more particularly to 3D digital watermarking which is a technical tool that by means of hiding secret information can offer copyright protection, content authentication, content tracking (fingerprinting), steganography (secret communication inside another media), content enrichment etc.

Up to now, 3D watermarking non-blind schemes have reached good levels in terms of robustness against a large set of attacks which 3D models can undergo (such as noise addition, decimation, re-ordering, remeshing, etc.). Unfortunately, so far blind 3D watermarking schemes do not present a good resistance to de-synchronization attacks (such as cropping or resampling).

This work focuses on improving the Spread Transform Dither Modulation (STDM) application on 3D watermarking, which is an extension of the Quantization Index Modulation (QIM), through both the use of the perceptual model presented in [DHM10], which presents good robustness against noising and smoothing attacks, and the the application of an algorithm which provides robustness against reordering and cropping attacks based on robust feature detection proposed in [RA06].

Similar to other watermarking techniques, imperceptibility constraint is very important for 3D objects watermarking. For this reason, this thesis also explores the perception of the distortions related to the watermark embed process as well as to the alterations produced by the attacks that a mesh can undergo.

Contents

List of Figures

Chapter 1

Introduction

1.1. Mesh watermarking and its applications

Nowadays the way people consume digital media has dramatically changed compared to the past. The amount of and the data volume related to multimedia content such as audio files, images and videos which are shared and transported over the internet is exponentially increasing. This content can be as well created by artists and professionals but also by users who more and more upload and share media files. This evolution poses several challenges such as bandwidth, storage and intellectual property rights (IPR).

The context of this master thesis is related to the IPR issues and more particularly to digital watermarking which is a technical tool that by means of hiding secret information can offer copyright protection, content authentication, content tracking (fingerprinting), steganography (secret communication inside another media), content enrichment etc. This technical solution has to be used in collaboration with other tools such as cryptography, digital right management (DRM) platforms and architectures, as well as a legal framework in order to be effective in practice.

The need for IPR solutions is very linked to the digital nature of current media content. In the past analog content was difficult to copy and reproduce (copying several times a video or audio tape lead to useless content after five iterations because of the content degradation). Now with digital content, copying can be processed an infinite number of times with a perfect fidelity. With the advent of peer to peer (P2P) file sharing, this causes huge benefit losses for artists, and the music and cinema industries since consumers tend to illegally download content instead of buying it. IPR tools are not only needed for protecting authorship rights. Another example of issues related to IPR is privacy protection. Indeed user-generated content uploaded on the internet can infringe rights of other people appearing on this content but ignoring it. Mobile recorded videos become a major threat for privacy compared to video surveillance cameras as none has control over them. A better regulation and better automatic media analysis tools could help detecting such undesirable and unauthorized distribution of individual's private life by allowing people to detect and track

whether they appear on the internet and how.

Among media files, 3D models are gaining attention since the last decade and the advent of 3D video games (with or without remote rendering), 3DTV (BlueRay3D, cinema movies e.g. Avatar, Dragons…), cultural heritage virtual visits in augmented or virtual reality as well as industrial design (CAD)… With respect to other medias (audio, image and video), signal processing tools are not as mature due to their manifold non-Euclidian nature and their graph-based non-uniform sampling. Widely accepted compression standards are still not available even though some first efforts have been proposed in MPEG4 for video object-based compression. In this context, it is not surprising to observe that state of the art watermarking tools for 3d meshes are not as advanced as the ones proposed for audio, image and video content.

1.2. Motivation, Objectives and Contributions

Given the exponential use of 3D models and their distribution, the need of a scenario aiming to protect their IPR against a wide variety of attacks becomes more acute. In order to increase such resistance a proposed scheme consisting in the combination of two 3D watermarking techniques is developed and evaluated.

This master thesis focuses on two state-of-the-art techniques which offer different and complementary advantages, respectively QIM-based 3d watermarking and feature point-based watermarking synchronization. The idea is to combine both in such a way that the new scheme would benefit from the advantages of both techniques and compensate for their respective fragilities.

Another point which is explored in this work is the perception of the distortions related to the embedding of the secret information needed by the watermarking algorithm as well as by the attacks that the mesh can undergo when a pirate wants to illegally infringe the IPR related to the watermarked content. These attacks tend to imperceptibly distort the content so that hidden data cannot be retrieved anymore. It is therefore important to develop tools in order to assess the imperceptibility of media content manipulations either by the insertion of the watermark or by the effect of well-known attacks so that the watermarking scheme can be optimized namely in terms of robustness and imperceptibility.

1.3. Outline

The master thesis text is organized as follows.

Chapter 1 shows a general overview of this master thesis. A brief introduction to the subject matter is provided, introducing the reader to the current situation of 3D watermarking systems and their evolution. Moreover, mandatory aspects such as motivation, goals and work done have

been introduced too.

Chapter 2 provides a review of 3D meshes and their specificities. It also presents the issues related to the measure of the perceived difference between the original and the distorted versions of the same 3D mesh. Two basic blocks define this chapter: the first one exposes general concepts of 3D triangle meshes, which are the object of the tested watermarking schemes; the second one is a revision of factors involved in the perception of 3D models, as well as the most used tools to evaluate its quality. Moreover, a more detailed description of the measure presented in [LDGDBE06], which we have used to evaluate our experimental results, is also provided.

Chapter 3 then introduces in detail the general background of digital watermarking from requirements and current systems to applications. A description of the watermarking system operation is also supplied, providing a general point of view about how the involved processes are expected to work.

A survey of the state-of-the-art 3D watermarking schemes is given in Chapter 4. A revision of the work done in terms of implementation and its limitations is presented. On the other hand, the specificities of 3D watermarking schemes, attacks and issues are discussed in the light of the context of this master thesis.

Chapter 5 is devoted to the analysis of the QIM-based watermarking scheme of Darazi et al. [DHM10], on which the scheme we propose is based. Experimental results of the algorithm and limitations are discussed.

Chapter 6 exposes and analyses improvements of this scheme by making use of a Visual Mask proposed in [DHM10]. 3D meshes features which have a direct implication on the visual perception are presented. The consequent perceptual-quality-oriented protocol is evaluated basing on state of the art perception metrics.

Feature point synchronization proposed by Rondao Alface et al. [RA06] is then detailed in Chapter 7. Improvements and its application to the Visual Mask-extended QIM algorithm are also presented and studied in this chapter. An overview of the scheme proposed is also provided and step by step deeply expounded.

Chapter 8 provides a practical overview and validation of the system operation. It illustrates the experimental results obtained during this master thesis in order to evaluate the capacities of the developed scheme.

Finally, Chapter 9 summarizes and concludes this thesis with a review of the current work. Conclusions are followed by possible future work paths on possible extensions and improvements of the contributions of this work.

Chapter 2

Background knowledge on 3-D Shape and Perception

2.1. Introduction

In this chapter we first present some basic concepts about 3D Polygonal meshes. Three dimensional data constitutes an actual emerging multimedia content. In this context, 3D models undergo through a wide variety of processing manipulations (compression, simplification, watermarking, etc.), which can introduce artifacts or degradations in the visual quality of the shape. We discuss in this chapter different tools to measure the loss of quality in 3D objects. Since this that is generally meant for human consumption, we discuss their subjective adaptation to the Human Visual System and subjective perception.

2.2. Triangle Meshes

A 3D *triangle mesh* consists of three combinational entities: vertices, edges and faces (triangles). The triangle is the basic geometric primitive for standard graphics rendering hardware and for many simulation algorithms. The connectivity of the mesh describes the way the vertices of the mesh are connected with edges and faces. It captures intrinsic topological properties of the mesh, containing all the information related to the genus [] and the existence and size of boundaries. The geometry of the mesh describes the actual positions of the vertices in the 3-dimensional Euclidian space R^3. Finally, there exist other attributes of the mesh, like color, normals or texture coordinates (see Fig. 2.1).

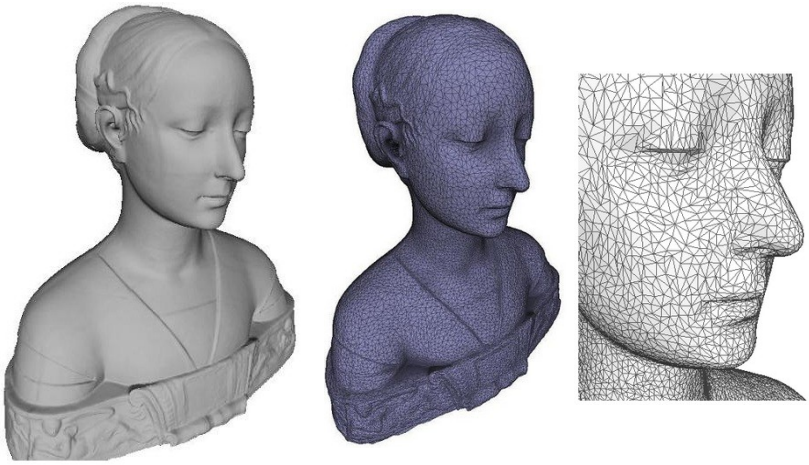


Figure 2.1. The Ippolita model. The close-up shows the details of a region of the triangulated surface.

2.3. Perception of a 3D shape

There are several factors which influence the visual perception of a shape, such as light sources, textures or the selected rendering technique. So far, the research community has been lacking of a widely used performance evaluation system for existing algorithms. Although it does not exist a standard distortion measurement, we present in this section some of the most common metrics used to evaluate distance measures in 3D meshes and their correlation with the perceived visual similarity.

2.3.1. Three-Dimensional Distance Metrics

Since 3D models are subjected to a wide variety of manipulations such as watermarking, compression or simplification, which can introduce degradations of the visual quality. These processes generally deal with visual information meant for human consumption. Hence the need of an efficient and accurate tool to measure the loss of shape quality perception or the visual difference between 3D objects become more acute. According to [RA06] these metrics can be classified as: Hausdorff distance, Volume based measure, Quadric Error Measure and Curvature based distance.

Hausdorff distance

Hausdorff distance is one of the simplest approaches to provide a mean square error (MSE)-like measurement for 3D models. It is based on a point to surface distance [CRS98].

The Hausdorff distance H(S,S') between two surfaces denoted S and S' is given by [ASCE02]:

$$H(S,S') = \max_{p \in S}(d(p,S')) = \max_{p \in S}(\min_{p' \in S'} \|p - p'\|_2) \quad \textbf{(2.1)}$$

, where $\|\cdot\|_2$ denotes the usual Euclidian norm, that is to say, the Euclidian distance between points p and p' in the 3D space.

This distance is in general not symmetrical, and thus the computation of a "one-sided" error can lead to significantly underestimated distance values. For this reason the symmetric Hausdorff distance $H_s(S,S')$, which provides a more accurate measurement of the error between two surfaces, is proposed and defined as follows:

$$H_s(S,S') = \max[d(S,S'), d(S',S)] \quad \textbf{(2.2)}$$

The metric provides results better correlated with the perception for quite similar meshes. However, iit does not match well the human visual perception as the measure grows larger. Moreover, the measure involves a high computational cost. Several optimizations have been developed for simplification applications [ASCE02, KT96, KLS96]. In a subjective opinion, Fig. 2.2 provides a visual example for the Venus model where noise addition and smoothing have been applied. Intuitively, the smoothed model appears visually less distorted than the noised one. We can see that the geometric mean distance does not reflect at all this subjective opinion (Hausdorff mean = 0.26 vs 0.25).

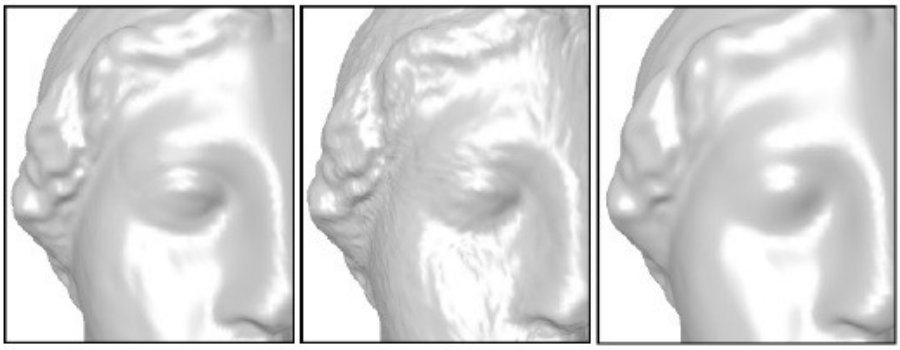


Figure 2.2. Samples of the corpus for the Venus model. From left to right, zoom on the original object, high noise on the whole object (Hausdorff mean = 0.26) and high smoothing (30 iterations using the smoothing filter from Taubin [T95]) on the whole object (Hausdorff mean = 0.25) (data courtesy of [LDBE06]).

RMSE, VNSR and Geometric Laplacian

Other 3D distances metrics used to compare compressed meshes which share the same connectivity are the VSNR (Vertex Signal to Noise Ratio, a.k.a. RMSE Root Mean Square Error), based on a mean point-to-point Euclidian distances, and the Geometric Laplacian (GLD) [KG00], which can be seen as the mean between point-to-point distances and a local smoothness evaluation.

The VSNR or RMSE distance between two surfaces M_1 and M_2 is defined as:

$$RSME = VSNR = \frac{1}{N}\sum_{i=0}^{N-1} d(a_i, b_i) \quad \textbf{(2.3)}$$

, where N refers to the number of vertices, which is the same in M_1 and M_2, d() represents the Euclidian distance in 3D space and a_i and b_i are the i^{st} points in M_1 and M_2. On the other hand, the GLD is defined as follows:

$$GLD(M_{1,}M_2) = \frac{1}{2N}\left(\sum_{i=0}^{N-1} \|a_i - b_i\| + \sum_{i=0}^{N-1} \|GL(a_i) - GL(b_i)\|\right) \quad \textbf{(2.4)}$$

, whith

$$GL(a_i) = a_i - \frac{\sum_{j\in N(a_i)} l_{ij}^{-1} a_j}{\sum_{j\in N(a_i)} l_{ij}^{-1}} \quad \textbf{(2.5)}$$

, where the neighborhood of ai is denoted by N(a_i) and l_{ij} refers to the Euclidian distance between a_i and a_j. GLD metric allows differentiating the addition of random noise in vertices of the mesh and poor reconstruction quality. Both measures increase in case of local disturbances detected by the geometric Laplacian. However, Fig. 2.3 shows that these three-dimensional distance metrics usually do not capture perceptual visual distortion. Image based-metrics [LT00] and perceptual models have been proposed to tackle this issue.

Figure 2.3. Perception of the Head model. From left to right, the Standford Head model, the same model after 430 iterations of Laplacian smoothing, and the same model after a noise addition of 0.17%. RMSE metric leads to the same score of approximately 0.0001 when comparing the original model with the smoothed and the noised one, while the last one is clearly perceptually more distorted. (data courtesy of [RA06])

Perceptual models

Up to now the visual perception of 3D meshes has not been deeply investigated. Nevertheless, a mathematical model of human visual acuity based in contrast and spatial sensitivity measures was proposed by Reddy in [R01]. Perceptually-guided rendering techniques have been also explored by Bolin and Meyer in [BM98]. In [FSPG97] Ferwerda et al. proposed a computational model of a visual masking based on psychophysical data which predicts how the presence of one visual pattern affects the detectability. The perceptual visual distortion is still tackled in [PCB05], where texture and wireframe resolution perceptions are evaluated for simplification purposes.

Finally, other two works to study the perception of artifacts caused by several 3D watermarking schemes have been proposed by Corsini et al. in [CGE05] and by Lavoué et al. in [LDGDBE06]. Corsini et al. also propose a metric based on surface roughness estimation and develop subjective evaluation experiments to test it. Lavoué et al. proposed a tool applicable to evaluate any kind of 3D mesh processing algorithms (simplification, compression, watermarking, etc.). The measure is based on the concept of structural similarity [WBSS04] and on curvature analysis (mean, standard deviation, covariance) on local windows of the meshes. Subjective experiments prove the robustness of the approach and its strong correlation with subjective ratings as compare to geometric metrics.

This last measure, following the framework of Wang et al., is based on the computation of statistics (mean, standard deviation and covariance of the curvature) on local windows of the meshes (Fig. 2.4). For each local window the mean curvature μ_x and standard deviation σ_x are defined as follows:

$$\mu_x = \frac{1}{n}\sum_{v_i \in x} C(v_i) \qquad \sigma_x = \sqrt{\frac{1}{n}\sum_{v_i \in x} (C(v_i) - \mu_x)^2} \tag{2.6}$$

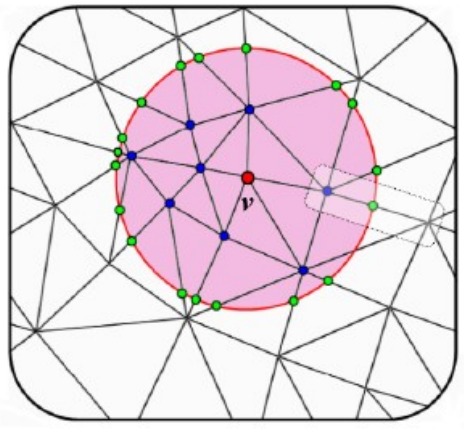


Figure 2.4. Example of local window computation.

The comparison between two corresponding local windows *x* and *y* from two meshes *X* and *Y* is done in by means of three functions: *L* (curvature comparison), *C* (contrast comparison) and *S*

(structure comparison), given by:

$$L(x,y)=\frac{\|\mu_x-\mu_y\|}{\max(\mu_x,\mu_y)} \qquad C(x,y)=\frac{\|\sigma_x-\sigma_y\|}{\max(\sigma_x,\sigma_y)} \qquad S(x,y)=\frac{\|\sigma_x\cdot\sigma_y-\sigma_{xy}\|}{\sigma_x\cdot\sigma_y} \tag{2.7}$$

, whith

$$\sigma_{xy}=\frac{\sigma_{xy}^2+\sigma_{xy}^y}{2} \qquad \sigma_{xy}^x=\frac{1}{n}\sum_{v_i\in x}(C(v_i)-\mu_x)(C(u_i)-\mu_y) \tag{2.8}$$

, where u_i is the vertex from the local window *y*, the closest (in Euclidian distance) to vertex v_i from window *x*, and σ_{xy}^x is the x-based covariance. The three comparison function *L*, *C* and *S* are combined in order to obtain a local distance measure *LMSDM* between two local windows x and y. The LMSDM distance measure is defined as follows:

$$LMSDM(x,y)=(\alpha\cdot L(x,y)^a+\beta\cdot C(x,y)^a+\gamma\cdot S(x,y)^a)^{\frac{1}{a}} \tag{2.9}$$

For our experiments the values of *a*, *α*, *β* and *Υ* have been chosen as *a*=3, *α*=0.4, *β*=0.4 and *Υ*=0.2 as proposed in [LDGDBE06], giving a lower weight to the structure function *S* since the disparity of the function is larger as compared to the other two.

Finally, the Mesh Structural Distortion Measure (*MSDM*) between two meshes *X* and *Y* is defined by:

$$MSDM(X,Y)=(\frac{1}{M}\sum_{j=1}^{M}LMSDM(x_j,y_j)^a)^{\frac{1}{a}} \tag{2.10}$$

, where *M* is the number of local windows in the meshes (i.e., the number of points of the mesh) and x_j and y_j are the local contents of the j^{th} 3D local window. MSDM value tends over 1 (theoretical limit) when the measured objects are visually very different and is equal to 0 for identical ones. Subjective experiments have been conducted proving the effectiveness and robustness of the *MSDM*, which provides a high correlation with subjective data.

2.4. Conclusion

This chapter has introduced the basic concepts of 3D triangle mesh and their visual perception, discussing about the most used 3D metrics used to evaluate the distortion and their limitations.

We have also presented the emerging research on perceptual metrics which integrates the Human Visual System to the distortion evaluation. The basic idea of the two mesh watermarking quality evaluation consists in prescribing the watermark payload and the watermark induced distortions. The perceptual measure presented by Lavoué (*MSDM*) is going to be used to evaluate the obtained results in next chapters. This perceptual-quality-oriented protocol requires that the MSDM distortion should be no more than 0.20 [WLDB08].

Chapter 3

Background knowledge on Digital Watermarking

3.1. Introduction

Since the digital original and copied information can be identical, the possibility of illicit reuse or manipulation of the data becomes an important issue. In this context, watermarking techniques can be applied to allow the protection of the ownership rights. Digital watermarking consists in a piece of information which is adhered to the data which it is intended to protect (the *cover* or *host* content), especially multimedia information, obtaining this way a watermarked data (the *stego* content). The Imperceptibility of the added mark by the legitimate end-user is a requirement for the watermarking system in most applications, as well as the robustness against attempts to remove or replace the mark.

Digital watermarking interest increase is most likely due to the increase in concern over copyright protection of content. Copyright protection of different types of data has experienced great advances since the publication of a seminal work by Tanaka et al. in 1990 [TNM90]. In this context, the watermark should be difficult to extract or remove from the watermarked content.

Cryptography [PGH99] provides also an effective solution to protect digital content. Nevertheless, the encryption does not provide any insurance about the legitimacy in the way that content is handled after the decryption, as oppose to digital watermarking, where the watermark is within the protected content and travels along with it undergoing the same transformations.

3.2. Requirements

Watermarking schemes must cope with four constraints: *capacity*, *robustness*, *security* and *imperceptibility*.

Robustness refers to the ability of the watermarking system to retrieve the watermark even in case that watermarked asset has been modified. These alterations of the content can be malicious, if they have been performed with the purpose to damage or remove the watermark, or non-intentional.

Robustness can be measured by applying correctness metrics to the extracted message as the bit error rate, the bit detection ratio and the correlation between both the extracted and the original bit strings.

The level of robustness of the hidden data will depend on each particular application. However, according to [BB04] we can consider four levels of robustness which cover most of the situations in practice.

Secure watermarking. The watermark should survive both non-malicious and malicious manipulations. The hidden data should be impossible to remove without damaging the quality of the host significantly. Concerning malicious manipulations we have to take into account that attacks are performed knowing the watermarking algorithm, thus proper strategies can be applied to remove the watermark. On the other hand, non-malicious manipulations include lossy compression, linear and non-linear filtering, cropping, editing, scaling, noise addition and other attacks belonging to a particular type of media. This robustness level deals mainly with copyright protection, ownership verification and other security-oriented applications.

Robust watermarking. The watermark should survive against non-malicious manipulations. The robustness level is in this case less demanding than secure watermarking. This constraint is suited for applications where the intention to remove or damage the watermark from the host content is not likely between the involved actors.

Semi-fragile watermarking. The watermark survives only a limited and specified set of manipulations. This is the case, for example, of data labelling for improved archival retrieval. In this application the watermark is needed to access the host data from an archive and can be discarded after the retrieval. In case that the host data has undergone some transformation, such as lossy compression, the watermark needs to be robust against them.

Fragile watermarking. The hidden data is lost or irremediably altered after any manipulation applied to the host content. The loss can be either global, when the whole watermark can no be retrieved, either local, when only part of the message is damaged. This level of robustness suits data authentication applications, where the hidden data loss evidences the existence of an attack. A well-designed fragile watermarking scheme is capable to locate or even identify the performed attack according to the retrieved watermark.

The *capacity* of a watermarking system refers to the amount of bits that the watermark is able to convey with more or less reliability. Capacity constrains will depend on the specific application. For instance, security-oriented applications where the robustness is a major concern often require hundreds of bits, while applications like captioning or labelling where the primary need is the embedding of a large number of bits will need a high-capacity watermarking scheme.

Imperceptibility requirement is often the most important and it refers to the quality of the watermarked asset. The embedding process of the system should ensure that the hidden data is imperceptible so the watermark does not degrade the quality of the content. Since watermarking can be applied to various types of data, the imperceptibility constraint will depend on the properties of the recipient. Imperceptibility relies on the imperfections of the human senses, thus their properties must be carefully studied. For instance, image and video watermarking will rely on the characteristics of the Human Visual System (HVS), while Human Auditory System (HAS) is studied in the case of audio watermarking.

In image and 3D watermarking it is often applied a perceptual mask which takes into account the characteristics of the HVS, in order to reduce the perception of the embedded message. Section X is going to analyse with more detail perceptual issues of 3D watermarking, based on the model proposed in [DHM10].

Security constraint is strongly application dependent. A secure watermarking scheme must be able to avoid non authorized people to detect or read the watermark.

Watermarking systems depend on the application at hand and a trade-off between capacity requirements, watermark imperceptibility, watermark robustness and security has to be found, due to the fact that these requirements are often antagonistic. High capacity is obtained at the expense of either robustness, imperceptibility or both. On the other hand, an improvement of the robustness of the watermarking system leads to a degradation of the quality of the asset, and vice versa.

3.3. Watermarking System Properties

This section introduces important concepts related to watermarking schemes. The host data, referred as the cover signal, is denoted by the symbol *A*. The embedding module may accept a secret key *K* as an additional input, which aim is to introduce some secrecy within the embedding step. The key *K* is used to parameterize the embedding process and make the recovery of the watermark impossible for non-authorized users (those who do not have access to *K*). The information message, i.e. the watermark code, is denoted by *b*.

The message *b* may not be embedded directly in the host signal. In this case, *b* undergoes through a transformation into a watermark signal *w*, which is more suitable for embedding. The watermark code *b* may be used to modulate a much longer spread-spectrum sequence, it may be transformed into a bipolar signal or it may be mapped into the relative position of pseudo-random signals in the case of position-encoded-watermarking. The message *b* may be also left as it is, coinciding with the

watermark signal *w*, being directly inserted within the host *A*.

In watermark embedding, the original content *A* undergoes through an extraction function *ε* and possibly a transform function *t*. The generated vector *x*, the message *w* and possibly the key *K* are the input of a watermarking function *wm*. An embedding function *ξ* takes the *x* vector and the output of the watermarking function *wm* and generates the watermarked data *y*. The inverse transform t^{-1} and the inverse of the extraction function ε^{-1} transform the watermarked data *y* into the watermarked content *B*. A general scheme illustrating this process is shown in Fig. 3.1.

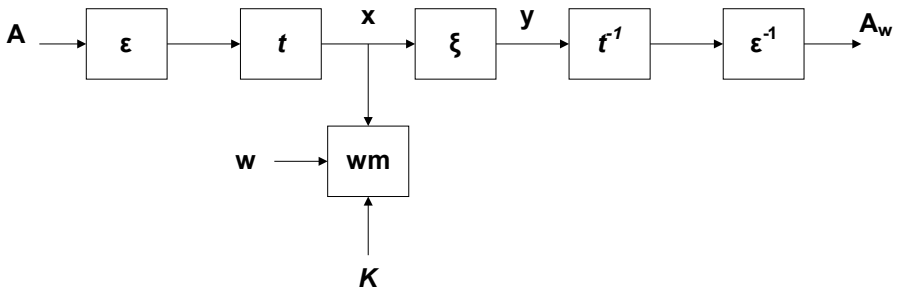


Figure 3.1. Scheme of the watermark embedding process.

After embedding, the watermarked asset A_w enters the channel and undergoes a series of manipulations. The output of the channel is denoted by A'_w.

The receiver part of the watermarking system can assume two different forms. The watermark detector reads A'_w and a watermark code *b'*, deciding if A'_w contains the watermark *b'* or not possibly with the requirement of the secret key *K* used in the embed process. In the case of a *non-blind* system, the detector will require the original asset A to compare it with the watermarked asset A'_w. In the case of *blind* detection, the detector does not need A to take its decision. This is the case of *detection* of the watermark, where it is possible to verify if a given code is present in the asset. Alternatively, in the case of *decoding* the watermark code *b'* is not know in advance and the aim of the receiver consists on extracting *b'* from A'_w. Fig. 3.2 shows and scheme of detection and decoding schemes. The use of detection or decoding in the receiver is relied to the specific application. For instance, in some applications related to authentication or stenography there is only a need of detection.

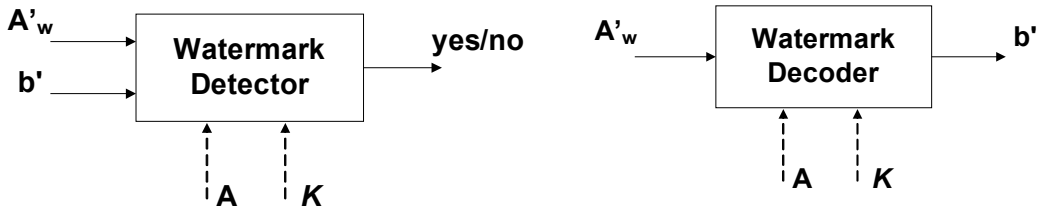


Figure 3.2. From left to right, watermark detection and watermark decoding schemes.

3.4. Watermarking applications

This section presents several applications that make use of watermarking. Different requirements in terms of robustness, imperceptibility, capacity and security for each specific application are also exposed. According to [RA06], watermarking applications can be classified in *copy protection*, *copyright protection*, *integrity protection*, *self-indexed contents*, *augmented contents*, *stenography* and *fingerprinting*.

Copy protection

The aim in copy protection applications [F02] is to embed in the cover content information needed to specify the use of it. For example, the information may describe whether the content can be copied or the number of copies which can be done. The most well-known example is the DVD [MKLTDH00]. Traded DVDs are encrypted and watermarked, whereas the burned personal DVDs are not. DVD players are then configured to read only encrypted and watermarked (commercial) DVDs and not encrypted and not watermarked (personal) DVDs. A watermarked but not encrypted DVD is detected as a pirate DVD and cannot be read. In this case, a simple detection of the watermark is needed.

Imperceptibility is the main requirement for these applications. An effective robustness is also required while security is not dealt and capacity can be limited to a few bits.

Copyright protection

As mentioned before, analogue copies lead to perceptual degradations in the cover content after several copies. This is not the case in the digital context. Copyright protection [AS03, WM01] intends to ensure the presence in the cover media of the identifier of the licenser.

Main requirements for this scenario are robustness and imperceptibility. The cover should resist to any combination of attacks until the cover media becomes too much degraded to be used. Security is not dealt and the capacity is normally constraint to 64 or 128 bits, which correspond to the memory needed to encode the serial number identifying the content buyer.

Integrity protection

Integrity protection [EG01, XA01] aims to ensure that the original content and the received one are conform. For example, press pictures sometimes undergo malicious alterations difficult to perceive but which change the semantic content. The goal of this application is to embed a fragile watermark which gets erased or damaged after any modification. The absence of the watermark means that the integrity of the received content is questionable.

In this applications, security and imperceptibility are the main requirements, and capacity is usually limited to a hundred of bits.

Self-indexed contents

The indexing applications give a semantical description of the content [L98], encoding the metadata in the header of a dedicated file format. When a format conversion is needed, the metadata may disappear if the targeted format does not take them into account. By means of embedding these metadata using a watermarking scheme able to resist to file format conversions the presence of them after the conversion can be ensured.

This kind of watermarking scheme is not intended to be secure.

Augmented contents

Augmented contents contain metadata enabling different levels of use. An example is the transmission of hidden depth maps in images, which allows everyone to watch these images, but only those who buy a dedicated decoder are allowed to watch the images in the augmented 3D.

Steganography

Steganography is the art to communicate a message secretly by hiding the communication in another message. It also complements cryptography, by transmitting discretely encrypted information.

In steganograpy the imperceptibility is required to make the stego-asset unsuspicious. Numerical undectability is also a main constraint. In this scenario, security is ensured by cryptography and robustness is not needed.

Fingerprinting

In this case the watermark identifies the buyer of a digital content. This mechanism deals with illegal copy of digital contents [KT03], and discourages people to make illegal copy of the watermarked asset. For instance, it is potentially possible to trace back the illegal copies to identify the pirates of the content.

Imperceptibility is here again one of the main constraints, while security and robustness are relatively important.

3.5. Conclusion

In this chapter we have discussed about the interest of Digital Watermarking. Requirements for such systems are presented as well as the compromise between them depending on the application.

Afterwards some basic concepts on digital watermarking systems have been presented. Next chapter presents an overview on 3D digital watermarking.

Applications which made use of watermarking have been presented, pointing out the different requirements in terms of robustness, imperceptibility, capacity and security needed for each one. It is consequently obvious that there exists no watermarking scheme which would be optimal for every application.

Chapter 4

Survey of 3D Objects Watermarking

4.1. Introduction

The use of 3D models has experienced a considerable increase during the last decade mainly due to its usefulness in industrial, medical an entertainment applications. This Section gives an introduction to the main requirements and linked applications of 3D watermarking systems. It also provides a comprehensive survey on 3D mesh watermarking, where the particular difficulties encountered are discussed as well as an analysis of the existing algorithms.

4.2. 3D watermarking requirements

4.2.1. Robustness

Robustness represents an important requirement in the context of certain applications such as the Intellectual Property Right (IPR) protection. It refers to the ability of the embedded watermark to resist against malicious or non-malicious manipulations of the content. Although it has not been proposed a standard classification for 3D watermarking manipulations, also known as attacks, the classification used by most authors of 3D watermarking papers is next shown [RA06]. Fig. 4.1. illustrates some of the presented attacks on the Bunny model.

Similarity and affine transforms. Similarity transforms such as *rotation*, *uniform scaling* and *translation* (RST) are considered as a common manipulations, thus minimum requirement for a 3D watermarking scheme [EFS07]. These operations are easily accessible for any attacker. They modify the mesh geometry by displacing the vertices from their original coordinates, and are commonly used to place a 3D model inside a scene. On the other hand, other general affine transforms such as *non-uniform scaling*, *shear* or *projective distortions* are

considered as a malicious attacks [B99a].

Noising and denoising. Noising attacks is a geometric operation often consisting on the addition of white gaussian noise on mesh vertices positions by adding them random displacement vectors. Denoising attack is usually performed by applying a *Laplacian smoothing*, which acts as a low-pass filter on the mesh, attenuating the roughness of the surface [T95].

Connectivity attacks. Those attacks modify the mesh adjacency information preserving at the same time the aspect of the mesh unaltered, that is to say, the geometry and topology of the mesh. *Re-triangulation* is one of the connectivity attacks performed by changing the connections between mesh vertices. *Vertices and/or faces re-ordering* changes the order of vertex or faces in the mesh. Notice that the order of a 3D mesh samples has no physical meaning [RA06].

Sampling attacks. These attacks modify the mesh geometry and connectivity, whereas the mesh topology remains unaltered. Amongst sampling attacks we find mesh *simplification*, which removes points and faces of the mesh, mesh *refinement*, which adds points and faces usually by subdivision, and *remeshing*, which changes the density and connectivity of points locally or globally [AUG06].

Topological attacks. They involve a change in the topological features of the shape. The most common amongst topological attacks is *cropping*, which refers to the disjunction of a part of the model. Cropping attacks degrade significantly the model but they can also preserve parts of the model that should indeed be protected (for example, the head of a statue). Other topological attacks include imperceptible cuts and hole filling.

Compression attacks. It does not exist a mesh compression standard [AG05], hence the robustness performance against compression is difficult to evaluate. The compression schemes for polygonal meshes are usually lossless for the connectivity and near-lossless for the geometry of the mesh. For this reason, the robustness against compression is commonly tested by the effect of the quantization of the vertex coordinates [U07].

Geometrical deformations. Amongst these attacks we find *bending* invariant signatures [EK03], mesh *editing*, mesh *morphing* and *local deformations*.

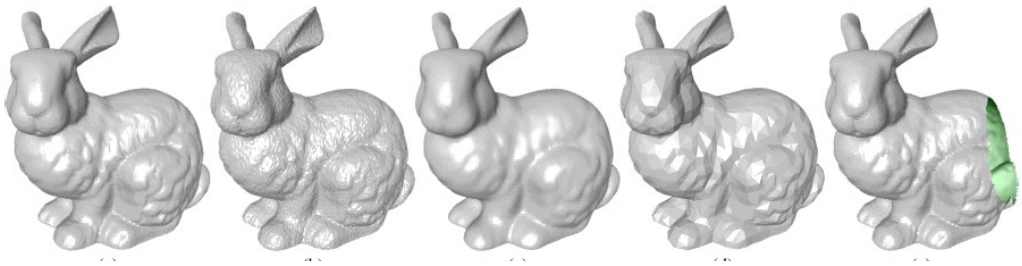


Figure 4.1. The Bunny model and four attacked versions. From left to right, the original mesh, the model after noise addition, the model after Laplacian smoothing, the same model after simplification, and finally the model after a cropping attack. (data courtesy of [WLDBH10])

4.2.2. Imperceptibility

The imperceptibility of the watermark is a crucial constraint for 3D watermarking systems. In most applications there is a need that the insertion of the watermark is not visually noticed. Many different meshes using different connectivities and densities can represent the same shape (Fig. 4.2). A robust watermark to protect the content should be properly retrieved on any version of the model representing the same content.

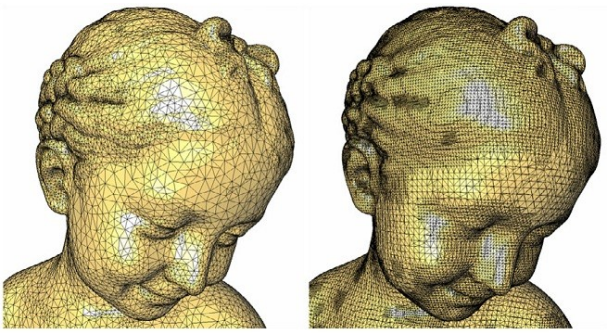


Figure 4.2. Two different triangle meshes representing the same content (Bimba model).

Nevertheless, the insertion of the watermark during the embedding process produces modifications in the host features. As said before, the imperceptibility requirement relies on the characteristics of the HVS, which has been deeply studied in order to improve the perception of the watermarked content in the case of image and video watermarking schemes [BBCP98, PZ98, WP99]. The imperceptibility is achieved through the visual masking process, which consists in properly detect those areas where the watermarking can be embedded without resulting perceptible.

Unfortunately, there is still no deep study of 3D objects features to mask the visual distortions. The realization of an effective visual mask for meshes is a real challenge since it depends on the particular rendering conditions used when visualizing the mesh, i.e., by the shading algorithm used [FDFH90], by the surface properties, by the textures [LPRM02], and so on [RA06, U07]. The possibility to interact with the object from users means another difficulty since the mesh can be observed from multiple viewpoints as compared as image and video watermarking.

Some of the metrics for benchmarking 3D watermarking schemes 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. However, this metrics do not provide a good estimation of the perceived visual distortion. For this reason, a perceptual model presented in [LDGDBE06] by Lavoué et al. which has been tested in subjective experiments is used in our experiments.

Section 6.1 of this survey presents a discussion on two of the properties of the 3D meshes, i.e. the curvature and the roughness, and their application in the creation of a visual mask which will be used in the QIM embedding algorithm [DHM10]. The same visual masking process will be applied later on Section 6.2, where we present another watermarking scheme which aims to improve the robustness against some attacks as compared to the scheme presented in [DHM10].

4.2.3. Capacity and Security

Whereas the security requirements for a 3D watermarking scheme do not differ significantly from the general requirements (Section 3.2), capacity constraints deserve some observations. Capacity is difficult to estimate and its measurement depends on the specific application. For authentication and data hiding purposes, since the mesh representation is the content intended to protect, the capacity refers to the number of points or faces of the mesh. For IPR protection purposes, the content to be protected is the shape of the mesh. In this case the capacity is thus related to the curvature variations of the surface [PKSRAA03, SPGKA06].

4.3. Applications of the 3D watermarking

This section briefly summarizes the application context of 3D watermarking as well as the requirements of such applications. According to the general consensus [RA06, NTTS01], 3D watermarking applications can be classified in three types which are related to different requirements of the watermarking scheme:

Intellectual Property Rights (IPR) protection applications. Copyright protection, fingerprinting, usage control and forensic are examples of this class of applications. They need robust watermarking schemes able to convey information about content ownership and IPR.

Content verification applications. Authentication and integrity checking belong to this class. These applications need a watermarking scheme able to detect if the content has undergone any alteration. In certain cases the scheme can determine the type of manipulation and its location.

Data Hiding applications. The aim of the watermark is based on conveying hidden information 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.

As previously said, the requirements of the watermarking system are very different depending on the application. They are often described in terms of capacity, robustness, imperceptibility and security.

4.4. State of the Art

Existing 3D mesh watermarking techniques can be classified in two main categories, depending on 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).

This state of the art review is mainly based on two references [WLDB08, RA06] and has been

updated to the latest publications in the field.

4.4.1. Spatial techniques

As explained before, the spatial description of a 3D mesh includes geometry and connectivity aspects. Most existing algorithms take the former as primitives, which show superiority in both robustness and imperceptibility compared to the latter.

Spatial Techniques Modifying the Geometry

Note that regardless of what the practical primitive is, all the techniques in this subsection are implemented by modifying the coordinates of involved vertices.

The algorithms that modify the vertices positions directly and individually are often fragile techniques. Yeo and Yeung [YY99] 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. At the extraction phase, they simply examine the validity of each vertex, and locate the possible attacks on the invalid vertices. In fact, this algorithm depends on a pre-established vertex order to avoid causality problem. Lin et al. [LLLL05] solved this defect and also proposed a more analytic and controllable modification scheme with a better attack localization capability. Cayre and Macq [CM03] proposed a high-capacity blind data-hiding algorithm for 3D triangular meshes. By choosing the projection of a vertex on its opposite edge in a triangle as the primitive (see Fig. 4.3), 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. Combining the above embedding scheme with an indexing mechanism [OMA97], which explicitly indicates the indexes of the embedded bits in the whole watermark sequence, Cayre et al. [CDMM04] devised an effective fragile watermark.

In Benedens's "Vertex Flood Algorithm (VFA)" [B99b], after grouping vertices according to their distances to the centre of a designated triangle, the range of the group interval is then divided into m = 2n subintervals, and all the group vertices distances to the chosen triangle centre are altered so that the new distances all fall into a certain subinterval that stands for the next n watermark bits.

Facets have several interesting measures for watermarking. Ohbuchi et al. [OMA97] 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).

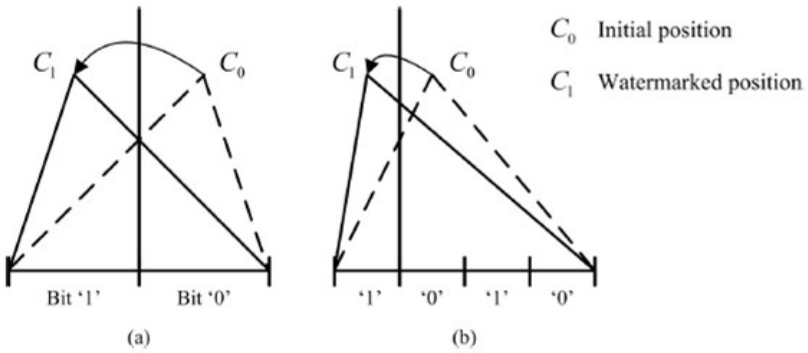


Figure 4.3. Watermarking primitive in the algorithm of Cayre and Macq [CM03], the projection is moved to the nearest correct interval: (a) opposite edge is divided in two intervals; (b) opposite edge is divided in four intervals. The inserted bits are both '1'.

Benedens [B99b] reported a blind algorithm in which the triangular facet height is quantized. By quantizing the distance of a facet to the mesh center, Wu and Chueng [WC05] gave a fragile but high-capacity scheme. In another Benedens's method [B99b], the Extended Gaussian Image (EGI) of a mesh is established by clustering facets according to their normal directions, then in each "bin" of the EGI, the average normal direction of the group of facets is modified to carry one watermark bit. Since these average normal directions approximately describe the mesh shape, this scheme is demonstrated to be relatively robust to simplification and remeshing.

Watermark embedding can be done in the spherical coordinate system, especially in the radius component. Since the radius (distance from the center of gravity to all points in a mesh) represents the shape of the mesh, its modification is supposed to be more robust than a single x_i, y_i, or z_i component modification. These are two main reasons for why numerous researchers chose to watermark in spherical coordinate system [CKPCJ05, ME04, ZTP04, DHM10].

Finally, resynchronization through spatial feature points of watermarks have been explored by Rondao Alface et al. with curvature-based features and protuberance features respectively [RAM05, RA06].

To summarize, the main drawback of the techniques that modify the geometry is the relatively weak robustness to both geometric and connectivity attacks. For blind schemes, the synchronization issue is really a difficult problem. However, these methods can have the advantages of high capacity and localization ability of malicious attacks.

Spatial Techniques Modifying the Connectivity

Actually, there are very few 3D meshes watermarking techniques based on connectivity modification. On the one hand, this kind of watermark is obviously fragile to connectivity attacks, and on the other hand, the introduced modification can be very easy to detect. Ohbuchi et al. [OMA97] presented two such algorithms.

In the first one, the local triangulation density is changed to insert a visible watermark. The second algorithm first cuts one band of triangular facets off the mesh, and then glues it to the mesh with just one edge. This facet band can be a meaningful pattern or be simply determined by a secret key. Both methods are visible and fragile, but the local distribution of the embedded watermark stops them from being a useful fragile watermark for integrity authentication.

4.4.2. Spectral Techniques

Most of the successful image watermarking algorithms are based on spectral analysis. A better imperceptibility can be gained thanks to a dilution effect of the inserted watermark bits in all the spatial/temporal and spectral parts of the carrier. A better robustness can also be achieved if the watermark is inserted in the low and median frequency parts. Unfortunately, for 3D meshes, there is currently no efficient and robust spectral analysis tool. Moreover, the lack of a natural parameterization makes spectral analysis even more difficult. As it can be seen in the following subsections, 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.

Spectral Techniques Based on Direct Frequency Analysis

Researchers have tried different types of basis functions for this direct frequency analysis. For Laplacian basis functions, a matrix of dimension N × N (N being the number of vertices) is constructed based on mesh connectivity. Then 3 * N spectral coefficients are calculated as the projections of the three coordinates vectors of all the vertices on the N ordered and normalized eigenvectors of this Laplacian matrix. Based on this analysis, Ohbuchi et al. [OMT02] proposed a nonblind method (additive modulation of the low and median frequency coefficients) while Cayre et al. [CRSMM03] gave a semi-blind one (quantization of the low and median frequency coefficients). There exist two serious problems with the Laplacian frequency analysis. The computation time increases rapidly with mesh complexity due to the diagonalization of the N ×N Laplacian matrix. Moreover, the analysis procedure depends on the mesh connectivity information. The first problem forced the authors to cut the original mesh into several patches possessing fewer vertices. To overcome the fragility to connectivity change, the authors proposed a pre-processing step of resampling at the extraction to recover exactly the same connectivity as the cover mesh.

Wu and Kobbelt [WK05] reported an algorithm that is based on radial basis functions. The construction of these basis functions is relative to the geometric information. This kind of analysis seems effective because it can give a good approximation of the original mesh with just a very limited number of basis functions. So calculation time can be greatly saved. In spite of this improvement, the algorithm remains sensible to various attacks, that’s why the authors still

proposed to do registration and resampling before the real extraction.

Although current 3D mesh spectral analysis tools are not efficient enough, they provide the opportunity to directly transplant the existing mature spectral watermarking techniques of digital images.

Spectral Techniques Based on Multiresolution Analysis

Multiresolution analysis 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 coarse mesh which represents the basic shape (low frequencies) and a set of details information at different resolution levels (median and high frequencies). These methods also permit realizing a synthesis process during which multiple representations with different complexities can be created.

The most interesting point of multiresolution analysis for watermarking is its flexibility. There are different available locations authorizing to meet different application demands. For example, insertion in the coarsest mesh ensures a good robustness, while embedding in the details parts provides an excellent capacity.

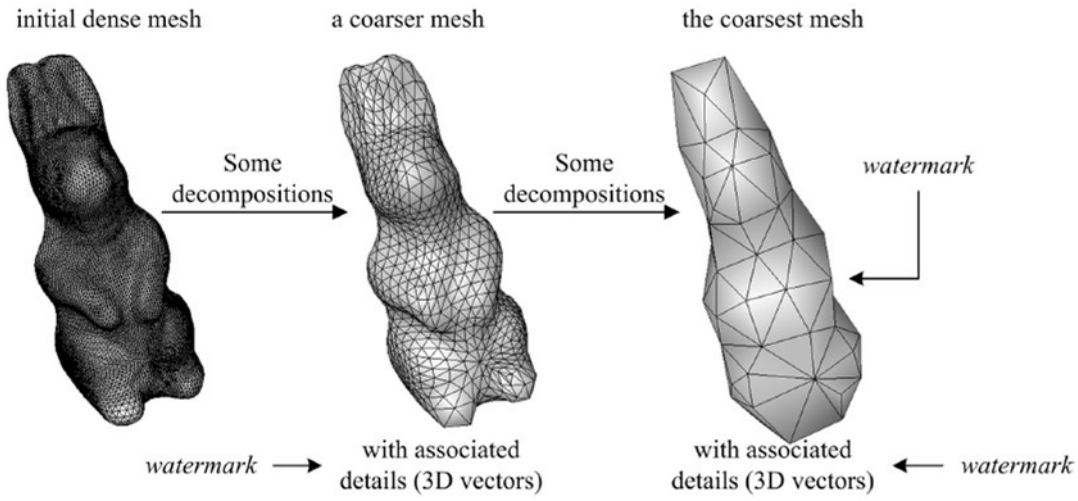


Figure 4.4. 3D watermarking techniques based on wavelet analysis

The insertion in low resolution can be both more robust and more imperceptible thanks to a dilution effect. The insertion in high resolution level may permit constructing some effective fragile watermarks with a precise localization ability of the attacks.

Wavelets are a common tool for such a multiresolution analysis. Fig. 4.4 shows the wavelet decomposition of a dense rabbit mesh, the watermark can be inserted either in the coarsest mesh, or in the wavelet coefficients at different levels. In fact, these wavelet coefficients are 3D vectors associated with each edge of the corresponding coarser mesh. Note that this kind of wavelet

analysis is applicable only on semi-regular triangular meshes. Based on this wavelet analysis, Kanai et al. [KDK98] 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. [UCB04] described a blind one-bit watermarking algorithm with the hypothesis of the statistical independence between the wavelet coefficients norms and the inserted watermark bit string.

Thanks to a remeshing step, the above analysis could be extended to irregular meshes. With this idea, Cho et al. [CLLP05] built a fragile watermark to accomplish authentication task in the wavelet domain. This remeshing step can also be done in spherical parameterized space. Jin et al. [JDBP04] used such a technique to insert a watermark into both the coarsest representation and the spherical wavelet coefficients of an irregular mesh.

Nonetheless, as the current direct spectral analysis tools, the available multiresolution analysis schemes have either connectivity restrictions or robustness deficiencies (especially to connectivity attacks). And for majority of these techniques, registration and resampling are recommended to ensure a sufficient robustness. But this inevitably makes the algorithms non-blind.

4.5. Conclusion

This chapter has presented important requirements of 3D watermarking systems and possible manipulations that 3D meshes can undergo.

Afterwards, a survey on the evolution of watermarking technology has been presented. 3D watermarking systems has still a wide set of challenging issues. The trade-off between capacity, robustness and imperceptibility supposes a classical problem in watermarking schemes. For instance, a high robust system with an important watermarking intensity risks to result in losses in terms of visual quality. On the other hand, whereas the robustness is improved along with the number of insertions of the watermark, the capacity gets unavoidably limited by this fact.

Data insertion in mesh elements such as point coordinates in order to resist RST transforms and vertes re-ordering has been solved in many ways. However, there is still work remaining in terms of security and capacity optimization.

Shape protection for forensic applications has been deeply explored. A large set of original techniques have been proposed based on signal processing extensions to irregularly sampled and manifold data. Their robustness tends to be satisfactory, even though there is still work remaining on the quality of the resynchronization with the original mesh.

In addition, there is still a need for a more careful analysis on how to modulate the watermark strength accordingly with the local perceived distortion. Assessing the watermarking capacity of a shape according to its curvature information (independently of its mesh representation) is another issue that deserves more research [RA06].

The construction of blind and robust algorithms has still to be improved. Blind detection or decoding is more suitable for copyright protection scenarios. Nevertheless, there is still no algorithm able to deal with cropping and resampling attacks. When referring to spatial techniques, they deal with a problem in the security of the synchronization in the search of a robust scheme. Using certain robust aspects of the mesh to be able to locate and index the watermarking primitives whereas establishing an ordering criterion for the indexation seems a good solution.

Chapter 5

QIM 3D Watermarking Scheme

5.1. Introduction

The 3D mesh watermarking method presented in [DHM10] proposes an algorithm to embed the watermark applying Spread Transform Dither Modulation (STDM). This method is an extension of Quantization Index Modulation (QIM). Besides the simplicity and the trade-off between high capacity and robustness provided by QIM methods, it is also resistant against requantization.

The use of STDM in the embedding provides a more effective Watermark to Noise Ratio (WNR) [SHW06] and its high robustness against noise addition and smoothing attacks is experimentally tested.

The modulation of the STDM is made through a perceptual model based on the curvature and the roughness of the mesh, which takes into account the perceptual properties of the human visual system (HVS), with the aim of achieving a good trade-off between robustness and imperceptibility. On the other hand, security property is also preserved. All these properties make this algorithm suitable for steganography or copyright and fingerprinting applications, and its application in augmented contents scenarios can be explored.

The watermarking takes place in the spherical coordinates system. For instance, in the embedding of the watermark, only the vertex norm of each point is modified, i.e., the distance of the points where the watermark is embedded towards the center of gravity of the 3D object. It is possible to deduce that this fact provides high resistance to the rotation and translation attacks.

This chapter is going to first introduce a brief explanation about QIM and STDM methods. After that, the main features of the proposed method in [DHM10] are presented. Finally, experimental results for noise addition and smooth attacks are shown.

5.2. STDM

QIM methods, introduced by Chen and Wornell [CW01], provide some attractive advantages for watermarking such as a better robustness against noise attacks and improved security features.

In [DHM10], the host data is quantized by means of scalar, uniform quantizers built using dither modulation (DM). The index of the quantizer is chosen taking into account the message we want to embed. The quantization function is defined as follows:

$$Q(x)=Q(x,\Delta)=[(\frac{x}{\Delta})]\Delta \quad \textbf{(5.1)}$$

, where Δ is the step size and [·] denotes a rounding operation. For a host signal x, the embedding function for hiding binary messages is:

$$Q_i(\boldsymbol{x})=Q_\Delta(\boldsymbol{x}-\boldsymbol{d_i})+\boldsymbol{d_i}, \quad \nabla i \epsilon \{0,1\} \quad \textbf{(5.2)}$$

, where

$$d_0=\frac{-\Delta}{4} \ , \ d_1=\frac{\Delta}{4} \quad \textbf{(5.3)}$$

In the equation above, $\boldsymbol{d_i}$ represents the dither sequence used to perturb the host signal before the quantization. Notice that Δ controls the quantity by which every element of the x signal is altered.

In the method proposed, a more robust way to implement QIM is used, i.e., STDM. Instead of applying the dither directly on the host signal, STDM first applies a random unitary transformation by projecting this signal onto a random vector, $\boldsymbol{p}$ [SHW06]. Due to these random projections, the noise will only affect in the $\boldsymbol{p}$ direction, so then STDM provides more robustness against noise addition attacks.

Considering the whole signal x with length N whose elements are the vertex norms for each point, we segment the signal into segments with the same length *n*. In every one of these segments we are going to embed one bit of the message. With the spread of each bit we are providing more robustness to the method.

After that, the quantization takes place using DM. The watermarked signal can be represented as:

$$\boldsymbol{y}=\boldsymbol{x}+(Q_\Delta(\boldsymbol{x}^T\boldsymbol{p}+d_i)-d_i-\boldsymbol{x}^T\boldsymbol{p})\,\boldsymbol{p} \quad \textbf{(5.4)}$$

$$\boldsymbol{y} = \boldsymbol{x} + (Q_m(\boldsymbol{x}^T \boldsymbol{p}) - \boldsymbol{x}^T \boldsymbol{p})\,\boldsymbol{p} \quad , \quad m \in \{0,1\} \tag{5.5}$$

As seen in the previous equations, the change takes place in the ***p*** direction. For our purpose, the detection is done without any original data or watermarked data (blind detection). Thus, in the detection the output signal ***z*** is projected onto vector ***p***, and the embedded message is determined as following:

$$m = arg\,min_{m \in \{0,1\}} \left| \boldsymbol{z}^T \boldsymbol{p} - (Q_\Delta(\boldsymbol{z}^T \boldsymbol{p} + d_m) - d_m) \right| \tag{5.6}$$

$$m = arg\,min_{m \in \{0,1\}} \left| \boldsymbol{z}^T \boldsymbol{p} - Q_m(\boldsymbol{z}^T \boldsymbol{p}) \right| \tag{5.7}$$

5.3. Experiments

In order to see how the STDM method to embedding the watermark proposed in [DHM10] some tests have been developed.

The implementation of the code embeds once the watermark message, distributing it between some points of the model without taking into account the perceptual mask which is presented in next chapter. The number of points of the 3D mesh is divided by the number of bits in the message to be embedded. The resultant number, *n*, corresponds to the size of the segments by which the whole vector of vertex norms is divided. Notice that some points may be left if the division is not integer. In the implemented code the size of the binary message is set up to 64 bits randomly generated.

The points used to embed the message are moved towards the center of gravity of the 3D mesh to the coordinates center (P(0,0,0)), where the watermark is inserted, and consequently the vertex norms *ρ* are modified. These points containing the watermark are then moved back towards the original center of gravity of the mesh.

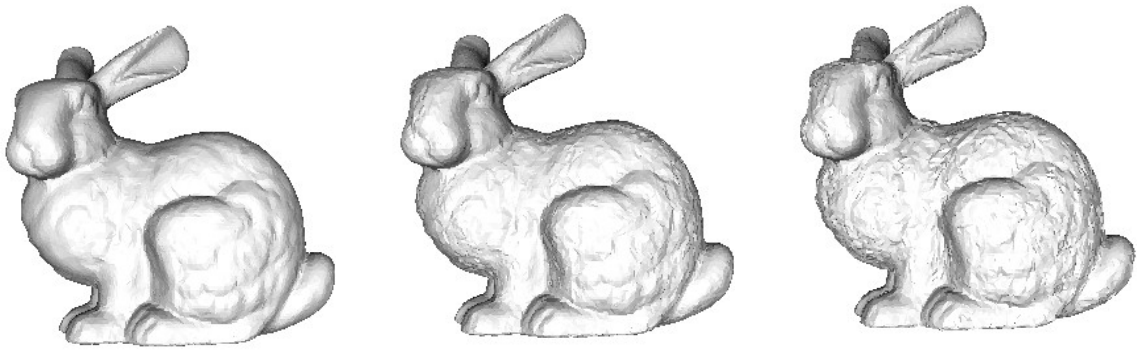


Figure 5.1. From left to right: Original 3D mesh of the bunny model; Watermarked 3D mesh of the bunny model (Δ=0.05); Watermarked 3D mesh of the bunny model (Δ=0.1).

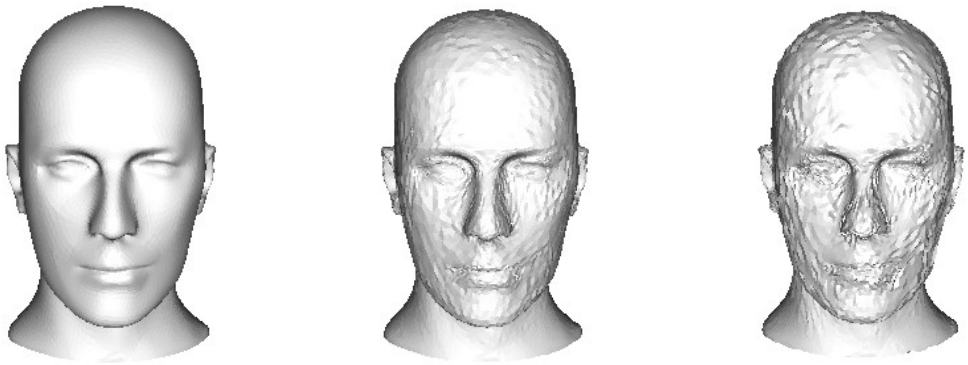


Figure 5.2. From left to right: Original 3D mesh of the head model; Watermarked 3D mesh of the head model (Δ=0.05); Watermarked 3D mesh of the head model (Δ=0.1).

We can observe in Fig. 5.1. and Fit. 5.2 the embedding of the watermark for the head and bunny models using different strength factors Δ. Obviously, the perception of the presence of the watermark in the 3D mesh grows along with the watermarking intensity Δ.

The center of gravity of the 3D object is calculated from the original model. In the implemented algorithm, this value is stored in the memory and is used later in the BER calculation to check the robustness in front of the attacks. However, the center of gravity of depart is a side information. For instance, the decoder can not have the information about this center of gravity, since this fact will suppose a security failure. The calculated BER after the attack should work with the center of gravity of the watermarked and maybe attacked object, which may differ from the original one after the insertion of the watermark or the attack alterations. It is important to underline that for this reason from some values of the strength factor Δ in the embedding the center of gravity will be different enough from the original one so that the recovery of the correct message will not be possible.

The embedding of the message without changing the center of gravity can be solved through an optimization problem with the position of it as a constraint [HRAM09]. This solution solves the problem for the extraction process after the embedding. However, in case of an attack the center of gravity will change anyway. This issue can be taken into account in the future work.

Although we can not ensure that this center of gravity will be the same of the original one, the method will work properly for some parameters, as we will see in the obtained results.

The STDM method presents strong robustness to the noise addition attack. A uniform white Gaussian noise with null mean is added to each point of the mesh. We observe in the tests that for some noise ratio (NR) values the BER changes from the optimal value 0 (the watermarked message is correctly decoded) to other worse values, depending on the defined Δ.

Fig. 5.3 shows the effect of this attack in the Bunny model mesh for different NR values and Δ=0.1. It also shows the obtained BER in each case. Both the message and the vector ***p*** are randomly generated. In order to filter the random effect, all tests have been repeated five times and the result shown corresponds to the average. On the other hand, attacks have been tested in the

Bunny model, which presents a proper model with flat, rough, plane and sharp areas.

In both cases the attack is clearly seen and it is obvious that the 3D mesh has been deformed. However, there exist some parameters by which the attack is not perceptually noticed and the watermark has to be properly decoded. For the case of Δ=0.1 the obtained BER is 0 for NR=0.009, where the noise addition is clearly observed. Hence the algorithm presents good robustness against a non obvious noise addition attack.

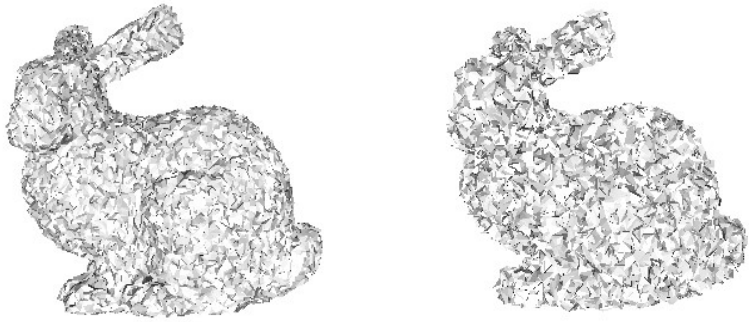


Figure 5.3. From left to right: Watermarked and attacked mesh for the bunny model. NR=0.009. BER=0; Watermarked and attacked mesh for the bunny model. NR=0.021. BER=2.5%.

QIM provides strong resistance to noise attack. Although the 3D mesh can be deformed, the watermark is still correctly read. The behaviour of the method becomes worse when the center of gravity of the attacked content becomes very different from the one of original model after the attack. For small changes of the center of gravity due to the attack the algorithm works properly, due to the average effects of the QIM algorithm.

The noising attack changes both the center of gravity of the 3D mesh and the position of the points between themselves. The resistance directly depends on the strength factor of the watermarking process, Δ. Fig. 5.4 shows a graph of the BER values obtained for several Δ and Noise Ratio values.

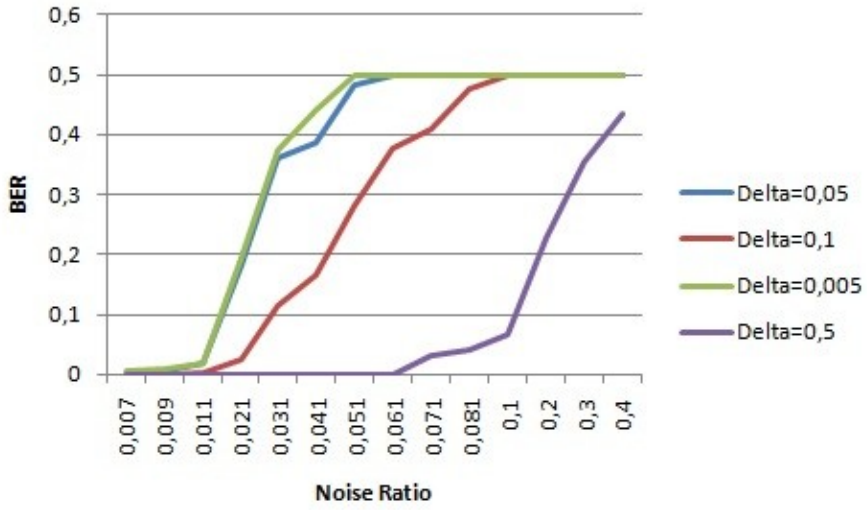


Figure 5.4. BER values obtained using different strength factor Δ and NR values.

It is important to underline that even if the resultant BER in decoding is not null, the watermark may have resisted and the watermark may not be retrieved due to a desynchronization caused by the noising attack.

In order to test the QIM robustness against the smoothing attack, a Laplacian smoothing is used in the implemented code to simulate this attack. Fig. 5.5 shows the effect of this attack in the Bunny model mesh for different smoothing levels and Δ=0.1.

As in the noising attack, the resulting BER is not the expected one from some values of the smoothing level for which the attack is already visible. Fig. 5.6 shows a graph of the BER values obtained for several values of Δ and smoothing levels.

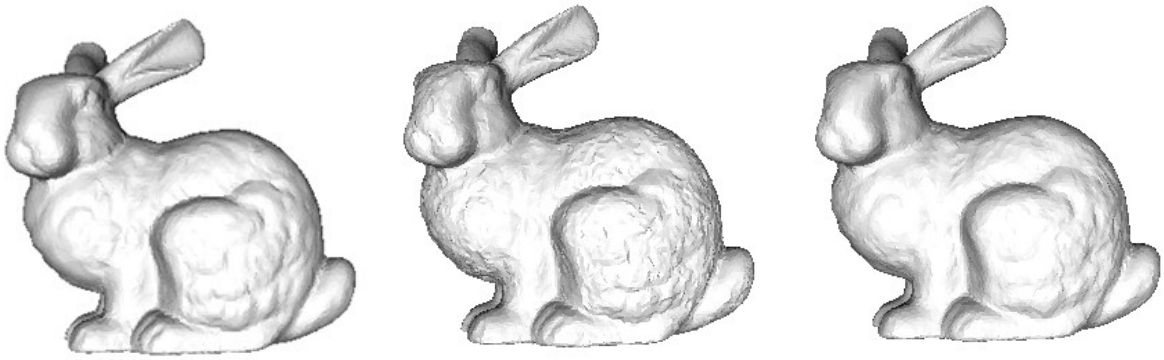


Figure 5.5. From left to right: Original mesh for the bunny model; Watermarked and attacked mesh for the bunny model. Smoothing level=0.02. BER=0; Watermarked and attacked mesh for the bunny model. Smoothing level=0.05. BER=29.7%.

Rotation, scaling and translation attacks are also tested in the code with good results. The construction of the scheme provides high robustness against them, since the watermark occurs in spherical coordinates system.

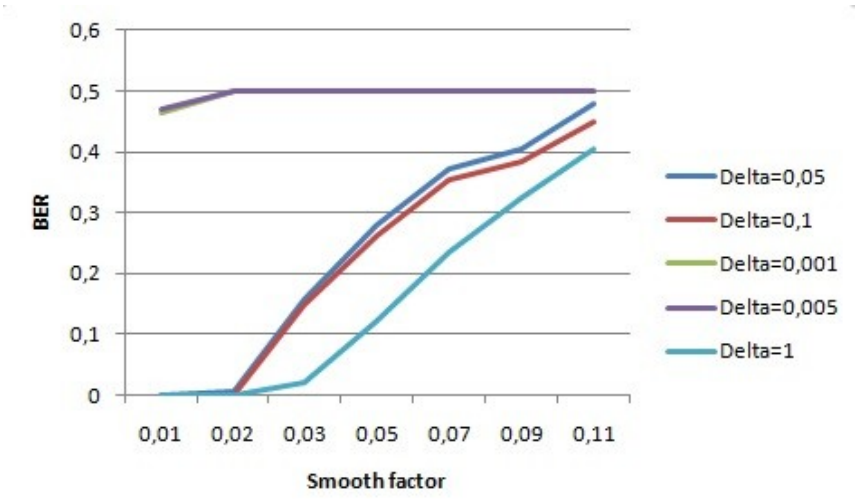


Figure 5.6. BER values obtained using different strength factor Δ values and smoothing levels.

QIM does not provide resistance to the cropping attack, since the center of gravity of the attacked mesh is significantly different from the original one. Moreover, the attack crops the segments of vertex norms $\boldsymbol{\rho}$, which are needed to retrieve the embedded message. The decoder requires as well the recovery of these segments in the same order established during the watermark embedding. Therefore, the scenario can not deal with re-ordering or resampling attacks (such as simplification or remeshing).

5.4. Conclusions

The method proposed is a robust 3D watermarking scheme for copyright protection. It uses STDM technique to embed the watermark in the 3D mesh, by dividing the vector of the vertex norms $\boldsymbol{\rho}$ (from the calculated points towards to the center of gravity) in segments of length n, spreading in this way each bit of the message in n points.

The algorithm resists rotation, scaling and translation attacks. If the center of gravity is provided as side information the algorithm also presents strong robustness to noising and smoothing attacks up to a point where the attack is clearly visible.

The scheme cannot deal with cropping, re-ordering or re-sampling attacks, either because it is not possible to retrieve the original center of gravity or the same ordered segments where the message has been inserted.

Next chapter introduces and evaluates the integration of a perceptual visual mask to the method presented in the present chapter.

Chapter 6

Perceptual Mask: Curvature and Roughness

6.1. Introduction

The need of imperceptibility is maybe the most important requirement in a watermarking system. In this chapter we present two qualities of the 3D models, i.e. the curvature and the roughness, which are going to be used to perform a visual mask that will improve the imperceptibility in the watermarked asset after the embedding process. As we will discuss later, the visual mask will also present advantages against noise and denoising attacks.

6.2. Curvature and Roughness

In a 3D surface *S*, a *normal curve* is defined as the intersection of *S* with a plane containing the normal at point *p* (see Fig. 6.1). The curvature of a normal curve is denoted as the sectional curvature. The minimal and maximal sectional curvatures are called the *principal curvatures* of *S* at *p*. The principal curvature directions (k_{max}, k_{min}) are the directions in the tangent plane for which the maximum and minimum are attained [RA06] (Fig. 6.1)

The product of the principal curvatures is called the *Gaussian curvature* (a.k.a. *intrinsic curvature*):

$$K_G = k_{max} \cdot k_{min} \quad \textbf{(6.1.)}$$

The sign of the Gaussian curvature indicates if principal curvatures have different signs (negative) or not. The average of the principal curvatures is called the *mean curvature* and it is defined as:

$$K_M = \frac{k_{max} + k_{min}}{2} \quad \textbf{(6.2.)}$$

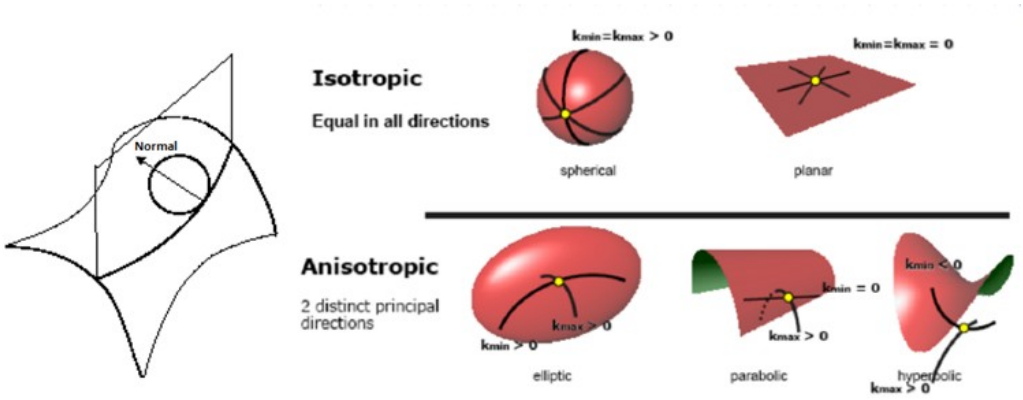


Figure 6.1. From left to right, curvature of a 3D curve at a point p, and principal curvatures for basic 3D shapes. (data courtesy of [RA06])

We say that S is a flat surface if K = 0. In order to implement the perceptual-quality-oriented model in the schemas evaluated in this Master thesis we have use a curvature which is an approximation of the mean curvature and it is calculated as follows:

$$Curvature(P_0) = \left| \sum_{P \epsilon V(P_0)} \frac{\overline{PP_0}\,\overline{N_{P0}}}{|\overline{PP_0}|} \right| \quad \textbf{(6.3)}$$

The roughness of a shape has also an impact in the perception of the model. A calculation on the roughness for each point of the mesh is developed in our scheme in order to improve the visual quality and is calculated as follows:

$$Rougness(P_0) = Variance_{P \epsilon V(P_0)}(|\overline{PP_c}|) \quad \textbf{(6.4.)}$$

, where $\overline{PP_0}$ is the vector from each point to its nearest neighbors, $\overline{NP_0}$ is the normal direction of the vertex P_0, $V(P_0)$ represents the set of the first M points with minor geodesic distances (see [GM09] for more information), and P_c is the center of gravity of the vertex P_0 and its M nearest neighbors (see [DHM10] for more information). Fig. 6.2 shows graphically the calculation of both parameters.

Fig. 6.2. From left to right: Calculation of curvature and roughness. (data courtesy of [DHM10])

6.3. Perceptual model

The perceptual model presented here is proposed in [DHM10]. The random vector ***p*** is modulated with a masking vector ***v***, which is calculated based on the curvature and roughness for each point. For instance, the amount by which every vertex norm will be modified in the watermarking process depends on the vector ***v***. This way, we are modulating ***p*** vector in the direction of least perceptual distortion.

$$[p'] = [v][p] \tag{6.5}$$

, where

$$\begin{bmatrix} v(1) \\ v(2) \\ . \\ . \\ v(n) \end{bmatrix} = \begin{bmatrix} Curvature(P_1) * Roughness(P_1) \\ Curvature(P_2) * Roughness(P_2) \\ . \\ . \\ Curvature(P_n) * Roughness(P_n) \end{bmatrix} \tag{6.6}$$

With this modulation, we are introducing the HVS perception while preserving the security, given a totally random vector ***p***. Notice that the dimension of ***v*** vector, *n*, corresponds to the size of the clusters by which the host signal is divided.

6.4. Experimental results

In this section some results on the operation of the integration on the watermarking QIM-based scheme presented in Chapter 5 of the perceptual model proposed are presented and evaluated using the MSDM measure previously introduced.

Fig. 6.3 illustrates the maps of the calculated curvature and roughness for the Head Dragon model, as well as the consequent mask values which will be used as weighting in the insertion process. The differences when using visual masking can be seen in Fig. 6.4 for the Head and David Head models.

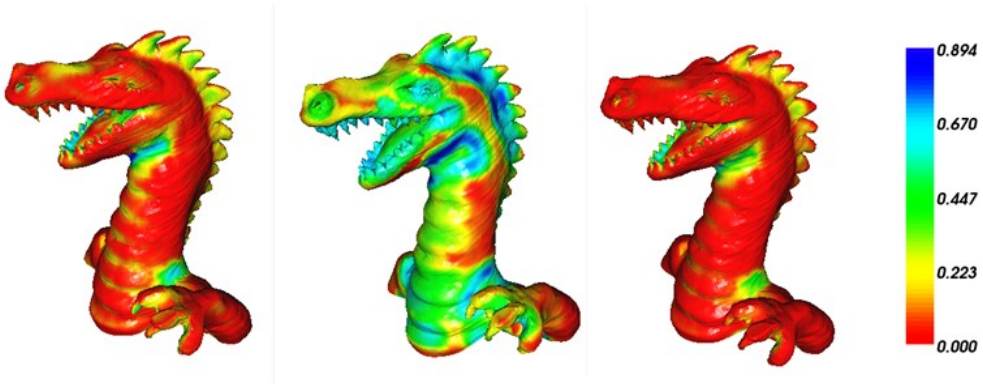


Figure 6.3. Visual masking mapping in the Head Dragon model. From left to right: the curvature mapping, the roughness mapping and the resultant masking.

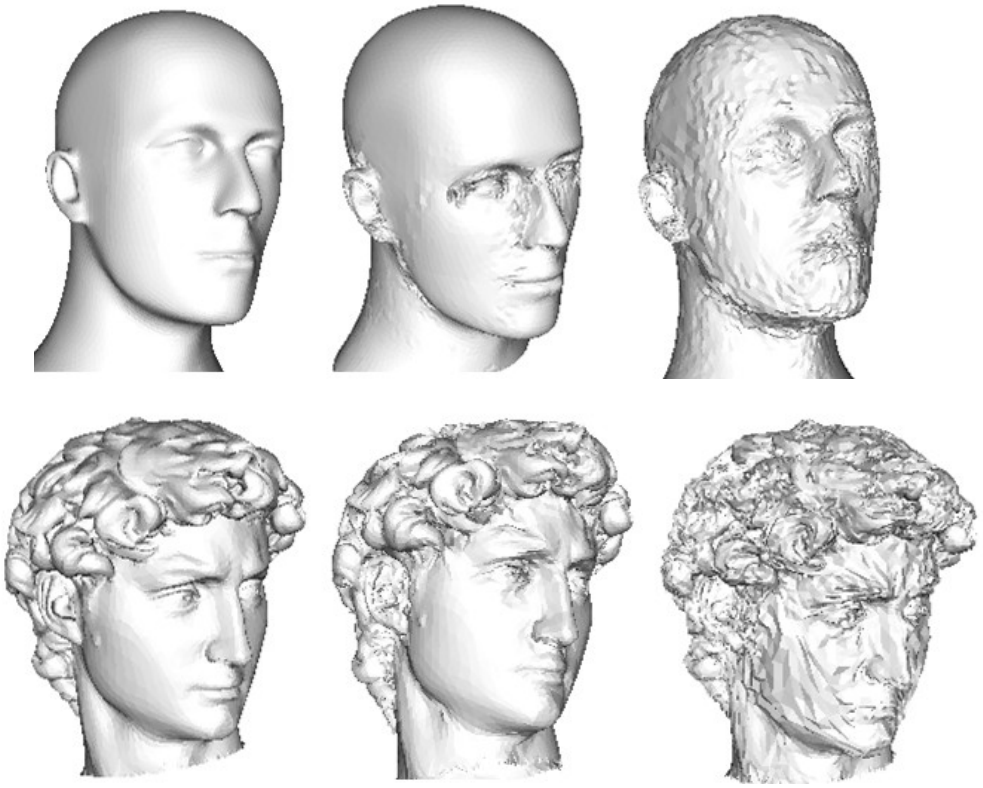


Figure 6.4. Visual masking operation. In the upper row, from left to right: original Head model, watermarked Head model when using the perceptual-quality-oriented model and watermarked Head model without using it. In the bottom row, the same layout for the David Head model. The values of the strength of the watermark (Δ=0.1 and Δ=0.3) are the same with and without the visual masking for each mesh. The watermarking strengths have been selected so that the distortions due to watermark are visible.

We have tested the imperceptibility of the proposed method for several meshes under different watermark strengths. We use the MSDM distance, previously presented in Section 2.3.1, to measure the difference between the original and the watermarked model. Fig. 6.5. gives the MSDM distances for the Bunny and Dinosaur models using different strengths factors.

Model	Watermark strength	MSDM with visual mask	MSDM without visual mask	Model	Watermark strength	MSDM with visual mask	MSDM without visual mask
Bunny	0,05	0,132722	0,178284	Dinosaur	0,05	0,157749	0,189489
	0,1	0,248411	0,293801		0,1	0,223013	0,404074
	0,2	0,407733	0,494154		0,2	0,490313	0,554856
	0,3	0,538557	0,55349		0,3	0,545953	0,615069
	0,4	0,596735	0,612512		0,4	0,592775	0,669915

Figure 6.5. MSDM comparison with and without visual masking for Bunny and Dinosaur meshes.

6.5. Conclusions and future work

In this chapter, basic notions on geometry of 3D meshes in terms of curvature and roughness have been introduced, as well as the perceptual model presented in [DHM10], which is going to be applied in following chapters in our scheme contribution.

The perceptual method presented in this chapter considers at the same level the implication in the visual quality of the curvature and roughness parameters, although their real effect has not been studied. A deeper survey on this topic can be developed for future works.

As it has been introduced in this Chapter, many different curvature measure exist and as a possible future work subjective tests could be elaborated in order to determine which of these curvatures is the most relevant for the human vision.

Moreover, the algorithm could also be tested when using the roughness calculation proposed by Lavoué in [L07]. In this paper Lavoué proposed a measure which provides the roughness value for each vertex of a given 3D mesh, as a local measure of geometric noise. This estimator depends on a scale parameter epsilon which determines the size (i.e. the frequency) of the details that have to be considered as noise and that can lead to a masking effect.

Chapter 7

Robust Feature Point Detection and Robust Mesh Segmentation for Robustness against Re-ordering and Cropping

7.1. Introduction

In this chapter we present a blind watermarking scheme which consists in robust feature point detection and robust mesh segmentation to be applied to the QIM scheme described in Chapter 5. The aim of the scheme relies on providing robustness against cropping and re-ordering attacks to the watermarking system.

For this purpose the coding system consists in detecting robust feature points around which we define neighborhoods where the watermark will be embedded. As described in the following sections, the feature point detection refers to the prominent points of the mesh described in [RA06]. Hollows detection based on the travel depth algorithm [GAGM09] is also introduced.

The construction of the patches will be carried out by performing a calculation of the neighborhood based on the n-rings progression (where n will define the extension of the patch). Those patches will also respect the calculated Voronoi regions.

The embedding process consists in the application of the QIM algorithm previously introduced, and the watermark is inserted in each created patch. The performance of the embedding process is tested with and without the use of the visual mask presented in Section 6.2.

In order to retrieve the watermark in the decoding process, the same algorithm to detect feature points is applied to the watermarked and possibly attacked model. Patches are reconstructed around detected features following the same algorithm than in the embedding process. To retrieve the message, the QIM extraction is applied to each patch.

7.2. Robust Prongs Detection

In this section we present how to detect robust prong features of a model. These points are defined as the local maximum of the protrusion function [RA06], and they do not depend on the curvature or the scale. The protrusion function performs an estimation of how much each point is perceived as protuberant extremities of a shape. As we will show later, prongs present a particular robustness against noising, denoising, re-sampling and cropping attacks.

Protrusion function

The protrusion value of each point depends on the sum of geodesic distances to all points of the shape starting from that point.

The protrusion μ of a point v on a surface S is defined by:

$$\mu = \int_{p \in S} g(v, p)^2 \, dS \qquad \textbf{(7.1)}$$

, where $g(v,p)$ is the geodesic distance from point v to point p on S. In the context of a 3D mesh representation M of the surface S, the function is discretized as follows:

$$\mu = \frac{1}{area(M)} \sum_{p_i \in M} g(v, p)^2 \, area(p_i) \qquad \textbf{(7.2)}$$

, whith

$$area(M) = \sum_{p_i \in M} area(p_i) \qquad \textbf{(7.3)}$$

, where $area(p_i)$ is the area of the geodesic neighborhood of point p_i and $area(M)$ is the sum of the area contributions of each point of the mesh M. The weighting areas can be chosen in different ways with similar results according to [RA06]. Three possible weighting areas are the total area of the 1-ring, the barycentric area and the Voronoi cell area. In our scheme we propose the use of the 1-ring area, which is divided by three since each triangle contributes to three points.

The geodesic distance between points p_i and p_j on a surface mesh M is the length of the shortest path on M linking both points. To calculate the geodesic distance we use the *Dijkstra* algorithm, for which the shortest path follows the edges of the mesh. The choice of the Dijkstra algorithm amongst others is based on the computation requirement of the protrusion of all points of a mesh ($O(n^3 \cdot n)$) and the fact that this algorithm offers better time complexity ($O(nlog(n))$) [RA06] as compared to other algorithms for interpolating on the faces of the mesh such as *fast marching algorithms* [KS98, PC03, PC05, SSG03] or *propagating windows algorithms* [SSKGH05]. Fig. 7.1 shows the calculation of the protrusion for several models.

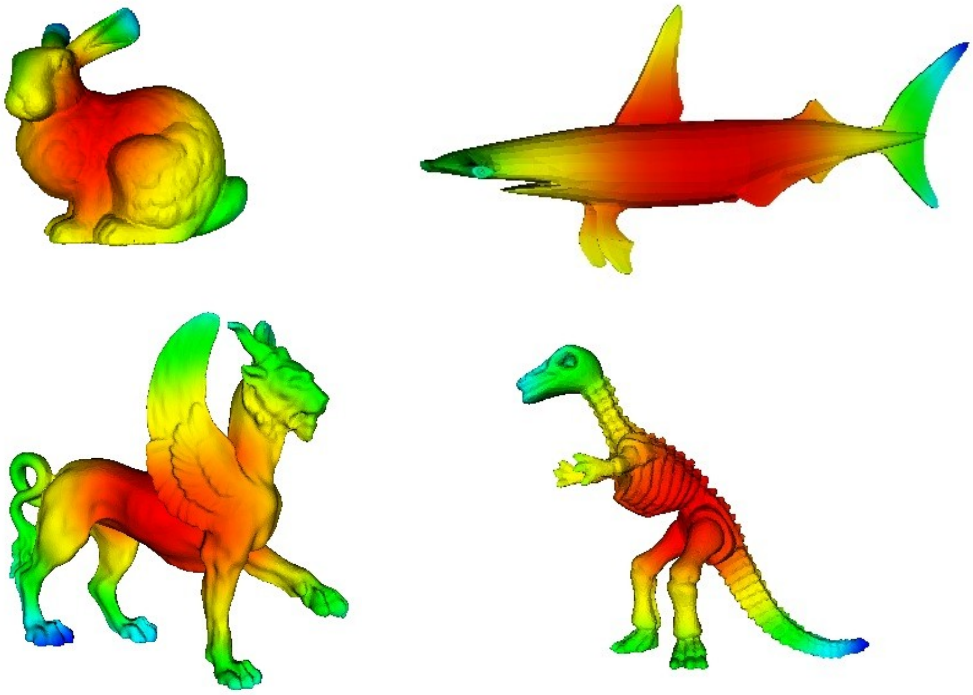


Figure 7.1. Illustration of the protrusion function on the Bunny, the Hammerhead, the Feline and the Dinnosaur models.

Local Maxima Detection

This section defines an algorithm to find out prominent feature points of a model through the maximization of the protrusion. Intuitively, prongs must reside on tips of prominent components of the model, and should be invariant to the pose of the model and to scaling.

Prongs detection can be applied to many applications such as metamorphosis [M00], deformation transfer [SP04], mesh retrieval [ZTS02], cross-parametrization [KS04] [PSS01][SAPH04], texture mapping [KSG03, ZMT05] and segmentation [ZMT05, ZH04].

The maximization of the protrusion function for detecting prongs has already been proposed in [RA06, KLT05, PKA03]. The first approach to carry out the detection consists on comparing the protrusion function of each point with points of its neighborhood (usually the 1-ring), resolving a

prong when its protrusion function is higher than the one of points contained in its neighborhood. However, this method leads to the detection of a too large number of prongs.

In [KLT05] Katz et al. proposed a variation of Multi-Dimensional Scaling (MDS) to find prominent feature points where the positions of the points depend on their protrusion function and their connectivity. MDS allows unfolding folded components of a model, thus the prominent feature points neither depend on the pose of the model nor on the scaling. An example is shown in Fig. 7.2. The goal of the performed scheme is based on the fact that it does not require prior knowledge regarding the number of feature points or user parameters. Nevertheless, MDS algorithm is very sensitive to re-sampling and cropping attacks. In [RA06] P. Rondão Alface proposes a prong detection algorithm which selects the *k* geodesically nearest neighbours of each point and compares its protrusion values. Two more constraints are applied in the performance of the detection: the detected prong should belong (or be very close to) the *convex hull* (a.k.a. *convex envelope*) of the mesh *M*, which is defined by the minimal volume convex shape containing *M*, and shape boundaries are rejected from the detection process since the algorithm aims to resist against cropping attacks (if the created boundary is far enough from the prong). This method offers resistance to changes in the pose of components of the mesh and uniform scaling, and it provides better results after re-sampling as compared to the 1-ring neighbourhood, the geodesic circle neighbourhood and the MDS algorithm.

Figure 7.2. Example of the MDS algorithm proposed in [KLT05]. On the left, segmentation of the monkey model. On the right, the monkey in MDS space (data courtesy of [KLT05]).

We propose a more general method which compares the protrusion of each point with the protrusion of the *k* geodesically nearest neighbours, as proposed in [RA06]. The local condition that a feature point should satisfy is defined as follows:

$$\forall \in N\, k(p), \mu(p) > \mu(q) \qquad \textbf{(7.4)}$$

The decision of the neighborhood extension for deciding whether a point is a maximum or not has a direct effect on the robustness of selected prongs, especially when we consider resistance to resampling and cropping attacks.

We can adjust the number of detected prongs by changing the value of the points contained in the neighborhood related to each point of the mesh. Moreover, we also discard those points which are on the borders of the mesh, since we want to resist to cropping attacks, as proposed in [RA06].

However, in order to perform a more general algorithm we do not restrict the prongs to be contained in the convex hull, since this fact can lead to some losses in the detection of the prongs. A proof is given in Fig. 7.3. Obviously the method is robust against re-ordering attacks and it also provides a good behaviour in case of re-sampling attacks due to the construction of the protrusion function. In exchange of this robustness, we have to specify the neighborhood to compute the calculation of the local maxima of the protrusion. Some examples of the protrusion over several meshes are illustrated in Fig. 7.4.

Computational cost

As said before, the protrusion function requires a high computational cost. From the formula of the protrusion (Equation 7.1) we can see a product between the complexity related to the number of basis points and the areas we take for approximating the integral by a sum and the complexity of the geodesic distance approximation. Since we compute the protrusion for all points of the mesh (*n* points), the complexity of the protrusion is $O(n^3 logn)$.

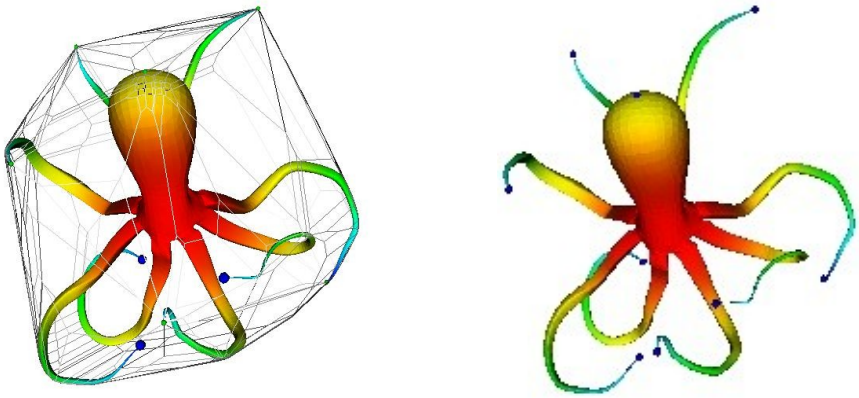


Figure 7.3. On the left, prongs detection scheme proposed in [RA06] on the Octopus model. The correctly detected prongs are in green. The prongs in blue are prongs that are not on the convex hull and thus cannot be detected with this method. On the right we see all prominent points in blue are properly detected with the more general method we propose.

In order to improve the computation requirement we use a decimation process. Decimation is an iterative process that at each step removes vertices from the mesh by minimizing the consecutive quality loss. When the number of vertices dramatically drops, it can be impossible to reproduce the correct topology of the mesh (number of holes or handles). The quality is then significantly damaged. The decimation we propose simplifies the mesh to *k* points with $k<<m$ in order to approximate the integral by a sum, decreasing the number of times (n·n) that we calculate the geodesic distances to $O(nlogn)$, obtaining a complexity of $O(kn^2 logn)$ *instead of* $O(n^3 logn)$. Another possibility would be add a simplification of the mesh to *m* points with $m<<n$ to compute the geodesic distances, obtaining a complexity proportional to $O(knmlogm)$. The decimation we propose allows keeping a good precision for the geodesic distances by following an approximation of the derivative for an equivalent of the Riemann sum.

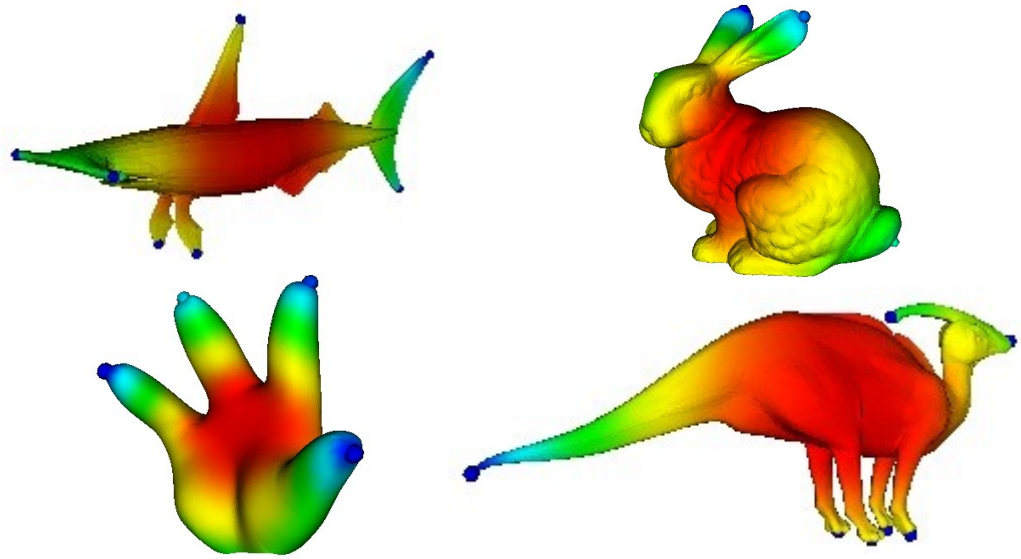


Figure 7.4. Illustration of the prongs detection on several models. On the top left, the Hammerhead model, on the top right, the Bunny model, on the bottom left the Hand Cartoon model and on the bottom right the Dilo model.

Some tests are done over the Hammerhead model with different decimation factors to test that the proper behaviour when using the decimation in the calculation of the protrusion. Results are given in Fig. 7.5.

Model	Decimation factor	Detected prongs ids							MISSING	NEIGHBOR
Hammerhead	0.01	553	883	935	1071	1283	1840	1988	0	-
	0.5	553	883	935	1071	1283	1840	1988	0	-
	0.7	553	883	935	1071	1283	1840	1988	0	-
	0.9	553	883	935	1071	1283	1840	1988	0	-
	0.99	553	883	935	1071	1283	1840	1988	0	-
	0.999	553	883	1071	1283	1533	1840	1988	2	1

Figure 7.5. Prongs detection in the Hammerhead model for different degrees of decimation. The results show that the scheme works properly for decimation factors from 0.01 and 0.99, that is to say, 99% and 1% of the mesh points left for the calculation of the protrusion, respectively. The column Missing resumes the number of prongs no properly detected as compared as to the prongs detected in the Bunny original model. The column Neighbour shows the number of missing prongs which are contained in the 1-ring neighbourhood of the missing prong.

Robustness of prongs

We perform some tests to evaluate the robustness of the feature points detection against watermarking noising attack. In order to prove the efficiency of the method, noise addition and smoothing attack are simulated on the watermarked object and the id of each prongs between original and attacked mesh is compared.

We first launch some tests with a decimation factor of 0.9 in the calculation of the protrusion, so 10% of the points are still left to calculate the function. From results obtained we observe that prongs present very good robustness against noise and smoothing attacks. The number of detected prongs and their id are maintained up to a very visible noise attacks. We reduce progressively the decimation factor and perform attacks to the mesh in order to check the proper detection of the prongs. Fig. 7.6 and Fig. 7.7 resume the prong detection (using a neighborhood equal to 12) results after the noise and smoothing attacks respectively with different strengths for a decimation factor of 0.992 (still 0.8% of the mesh points left to calculate the protrusion). It is important to underline that cases of missing prongs occur for strengths for with the attack is clearly visible. The high robustness can be explained by the fact that these points are detected by the maxima of an integral function.

7.3. Robust Local Neighborhood

Voronoi Mesh Segmentation

Once the prongs detection has been carried out we have to define a neighborhood related to each prong. These neighborhood areas are patches that do not have to entirely cover the shape. Since we want to avoid the overlap between different patches, we first define the geodesic Voronoi regions from the set of detected prongs. The Voronoi regions V_i of a set of seeds s_i in a shape are the set of points of the shape that are geodesically closer to their corresponding seed than to the other seeds. Some examples of the Voronoi region relied to a previous prong detection are shown in Fig. 7.8.

Model	Noise Ratio	Detected prongs ids					MISSING	NEIGHBOR
Bunny original	-	2825	7198	7329	8634	9388	-	-
Bunny watermarked with perceptual mask and attacked	0.001	2825	7198	7329	8634	9388	0	-
	0.003	2825	7198	7329	8634	9388	0	-
	0.005	2825	7198	7329	8926	9388	1	0
Bunny watermarked without perceptual mask and attacked	0.001	2825	7198	7329	8634	9388	0	-
	0.003	2825	7198	7329	8634	9388	0	-
	0.005	2825	7198	7329	9109	9388	1	0

Fig 7.6. Detected prongs in the Bunny model after the noising attack. The column Missing resumes the number of prongs no properly detected as compared as to the prongs detected in the Bunny original model. The column Neighbor shows the number of missing prongs which are contained in the 1-ring neighborhood of the missing prong.

Through adjustment of the neighborhood used to calculate the local maxima of the protrusion we can control the number of prongs detected, hence the amount of Voronoi regions and of patches where the embedding will take place. One example of the prongs detection and Voronoi regions with different values for the neighborhood parameter can be seen in Fig. 7.9. The increase in the

number of prongs will lead to an increase in the capacity up to a certain point, due to the fact that the later construction of patches is restricted by the boundaries of Voronoi regions. On the other hand, the watermarking will occur in more regions, but since in principle the patches would be smaller, there will be less points available in order to embed the watermark.

Model	Smoothing factor	Detected prongs ids					MISSING	NEIGHBOR
Bunny original	-	2825	7198	7329	8634	9388	-	-
Bunny watermarked with perceptual mask and attacked	0.1	2825	7198	7329	8634	9388	0	-
	0.3	2825	7198	7329	8634	9388	0	-
	0.5	2825	7198	7349	9388	13419	2	1
Bunny watermarked without perceptual mask and attacked	0.1	2825	7198	7329	8634	9388	0	-
	0.3	2825	7198	7329	8634	9388	0	-
	0.5	2825	6502	7198	7349	9388	2	0

Figure 7.7. Detected prongs in the Bunny model after the smoothing attack. The column Missing resumes the number of prongs no properly detected as compared as to the prongs detected in the Bunny original model. The column Neighbor shows the number of missing prongs which are contained in the 1-ring neighborhood of the missing prong.

Patches construction

In [RA06] P. Rondão Alface proposes a watermarking algorithm which uses a statistical approach for the watermark embedding (radius histogram quantization). This ensures that even though the sampling is different, if the shape is still similar, the histogram will still contain the watermark thanks to its statistical nature. The scheme constructs robust local neighbourhoods which offer good resistance to cropping if after the attack we are able to recover at least two nearest prongs.

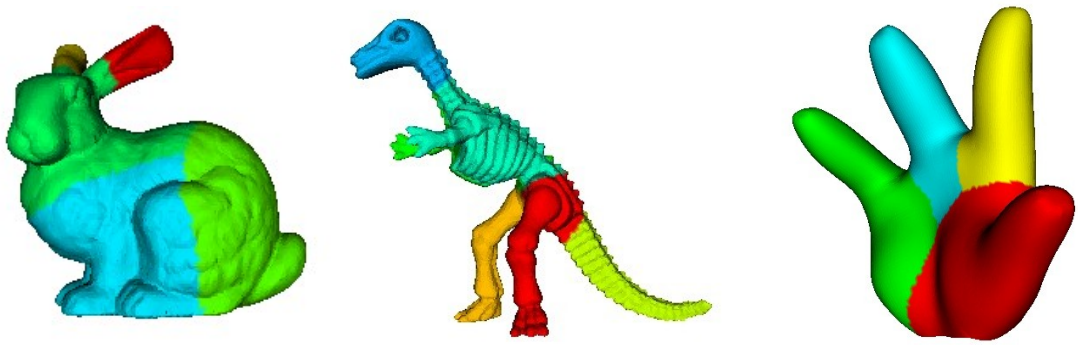


Figure 7.8. Illustration of the Voronoi mesh segmentation on several models. On the left, the Bunny model, on the middle, the Dinosaur model and on the right the Hand Cartoon model.

In the algorithm we propose patches are built in a similar way. The Voronoi expresses a similar condition that closest prong neighbours ensures no overlapping between patches and a proper recovery. Thus the proposed method still needs the closest prong or prongs to recover the correct

patch and watermark. The main improvement of the whole scheme is based on the fact that QIM can resist better to noise attack than the histogram-based method.

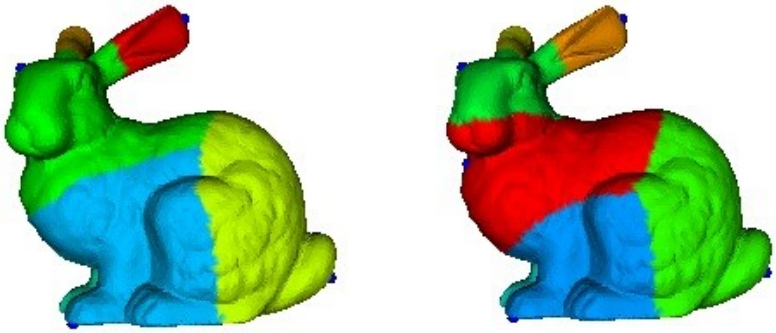


Figure 7.9. Detected prongs and subsequent Voronoi regions for the Bunny model with a neighborhood equal to 3 (3-ring), in the left image, and equal to 9 (9-ring), in the right image.

The patch related to each detected prominent point is constructed as a geodesic circle centered on the prong and contained inside the Voronoi region of that prong. We progressively calculate the n-rings starting from the prong up to a certain extension. One example of the construction of the patch by rings can be seen in Fig. 7.10 for the Head model. To make possible the embedding process, the chosen extension has to be obviously large enough in order to allow for finding out a neighbourhood which contains at least the same number of points as the number of bits of the watermark string. Since the patches are defined inside the Voronoi regions, no overlap is produced between them. An illustration of Voronoi regions for several shapes is shown in Fig. 7.11. The algorithm proposed here produces patches into visually meaningful components of the mesh. The main advantage of this method over others is its invariance to pose changes, which leads to a more robust watermarking.

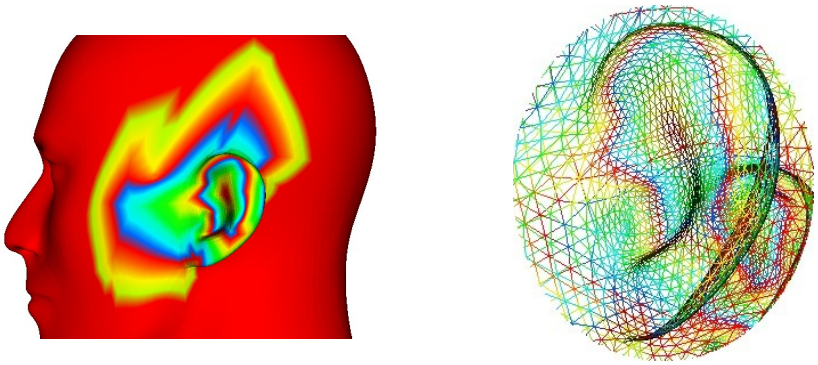


Figure 7.10. Patch construction following the geodesical progression by rings. The progression of colors corresponds to each detected neighbour ring from the prong. The progression stops when some of the points contained in the neighborhood reaches the Voronoi boundary. On the left, we can see the construction ring by ring of the patch around the prong detected in a close-up of the ear of the Head model. On the right, zoom on a region of the patch. Each ring of the progression has been displayed in a different color.

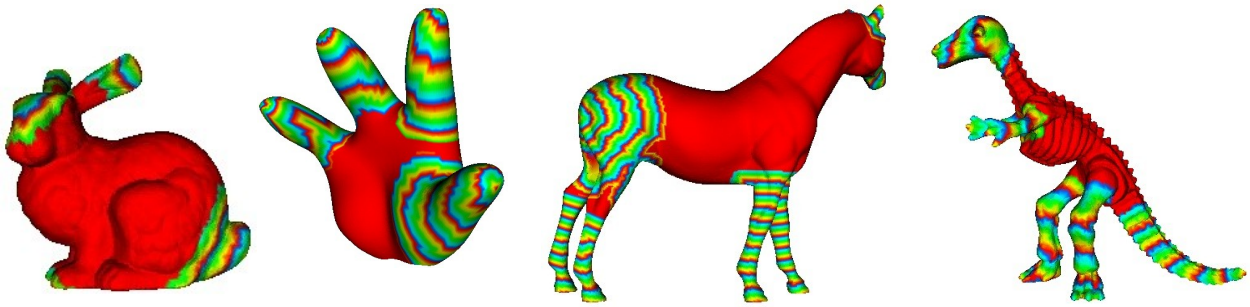


Figure 7.11. Constructed patches in different models. From left to right, patches detected in the Bunny model (6 patches), in the Hand Cartoon model (4 patches), in the Horse model (8 patches) and in the Dinosaur model (6 patches). We can observe that patches do not cover completely the surface of the shape, thus the capacity of the method proposed will depend on every model.

The extension of the patch can be chosen in different ways. One possibility is to define the number of points inside the patch as the integer part of n/k, where n refers to the number of points in the mesh and k is the number of detected prongs. Nevertheless, in case of cropping attacks n and k values change and thus the defined number of points inside a patch. Another possibility similar to the one proposed in [RA06] is to define the frontiers of the patches by stopping the progression when it riches the Voronoi boundary. In case of cropping we will still detect similar patches if we are able to locate the nearest prongs. Finally we can also combine this constraint with a limited number of points inside a patch to $n \cdot s$, where s is the number of bits in the message to embed. We will use this last option in order to continue the progression of the patch if the condition limited by the Voronoi frontiers does not provide a big enough number of points to perform the embedding (the embedding will take place at least once in each patch).

The more number of points we watermark in the mesh, the more difficult is to perform a cropping attack which erases the watermark and we will have more points to repeat the bits of the watermark (we increase the number of insertions). However, the increase in the robustness and capacity leads to losses in the quality perception of the mesh.

7.4. Travel Depth. Hollow Detection

In this section we present how to detect the hollow points of a shape. The algorithm, presented in [GAGM09], computes the shortest distance between the mesh surface and its convex hull without going through the mesh interior and assigns the corresponding value to each point of the mesh. Afterwards a local maximum detection takes place in order to detect hollows. The local maximum detection is developed by comparing the value of each point with a neighbourhood, which consists in a geodesical circle defined by a radius from the diagonal of the bounding box for every mesh. The travel depth algorithm and the later hollow detection are shown for some models in Fig. 7.12.

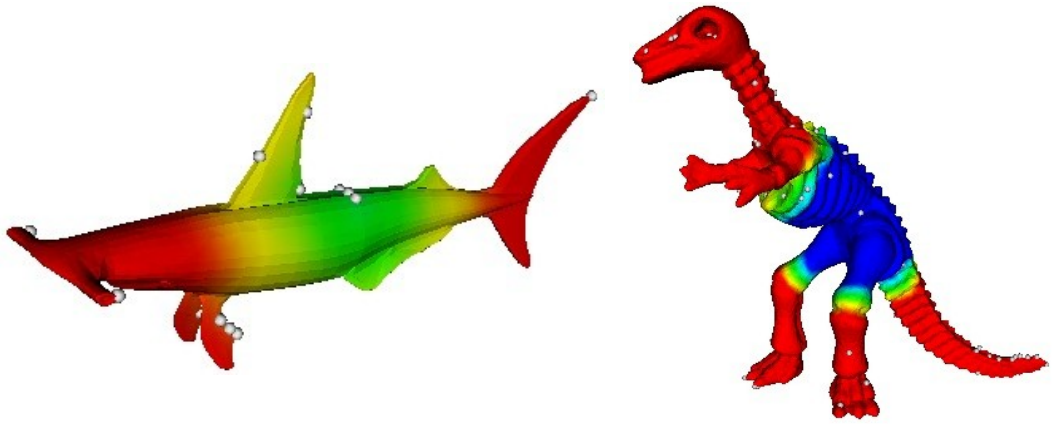


Figure 7.12. From left to right, travel depth and hollow detection for the Hammerhead model and for the Dinosaur model.

Some tests are done to test the robustness of the hollows detected. We observe that hollows are less robust against noise and smoothing attacks than the prongs. An example is illustrated in Fig 7.13. This fact is mainly due to the dependency of this algorithm on the convex hull, which depends on the *x, y, z* maximal coordinates (which is very sensitive to noise). From the tests performed we see that noise and smoothing attacks not only lead to a different hollows but also to a different number of hollows detected when using the same radius to define the geodesical circle. The number of points detected is higher in case of noising attack and lower in case of smoothing. Therefore in decoding after a noising attack there is no criterion to select detected points even if we would know the number of hollows in the embedding process. There is no way to know if a detected hollow corresponds to the correct one or it is geodesically close, which would lead to a construction of a similar patch.

7.5. Watermark Encoding and Decoding

Patches construction is done as an initial stage before embedding and extracting the watermark. Afterwards, each patch of the mesh is processed separately to embed the watermark by applying the QIM algorithm (Chapter 5). In the extracting process, the detection of prominent features and the later patch construction is done to retrieve the watermark. It is important to underline that by embedding the watermark several times robustness against local manipulations and cutting attacks is achieved.

Watermark embedding

The watermark is embedded by displacing the mesh vertices following the QIM algorithm. We can see in Fig. 7.14. an schematic of the whole watermark encoding system. The watermark is inserted

N times in each patch. As we increase N, the watermark robustness increases at the expense of making the insertion more perceptible.

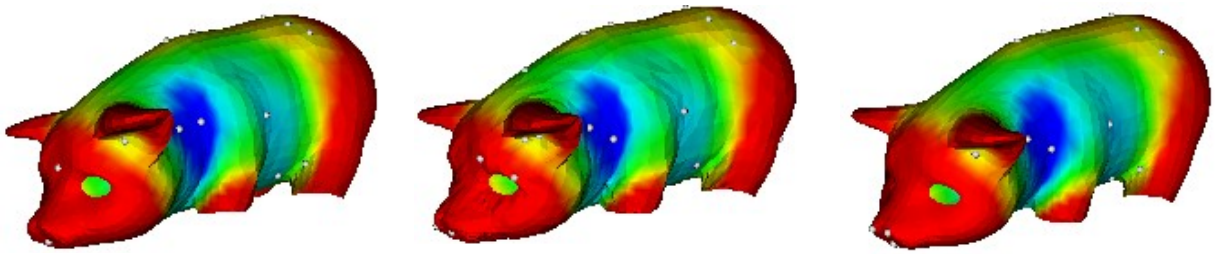


Figure 7.13. Travel depth and hollow detection of the Pig model. From left to right, the original model (25 detected hollows), the model in detection after a noise attack (30 hollows detected, NR=0.003), and the model in detection after a smoothing attack with just noticeable distortions (24 detected hollows, smoothing level of 0.03). The list of id's of hollows detected on the original model is not maintained in the attacked models. The detection has been performed by using the same radius to define the geodesical cercle in the local minimum detection process. The color corresponds to the travel depth distance for each point.

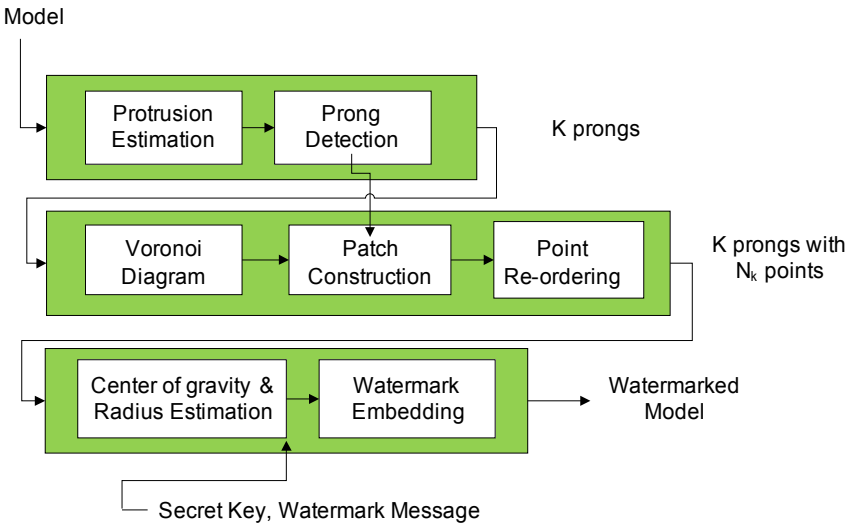


Figure 7.14. Proposed scheme of the feature points detection, patch construction and watermark embedding.

When a patch is watermarked, the perturbations caused by the watermark are embedded on the mesh by changing the distances of the vertex norms $\boldsymbol{\rho}$ of points in the patch towards the center of gravity of the model. An example of the watermark embedding for different watermark strengths is illustrated in Fig. 7.15.

Since the encoding occurs in spherical coordinates system, the algorithm resists to rotation and translation attacks. The scheme is also robust against re-ordering attacks. From the detected prong we assign id's to each point inside the patch. The assignation is done by ordering the points according to their geodesic distance towards the prong. We can have problems at the detection if the mesh is attacked with a manipulation which changes the position of the vertex, since the indexation of the points inside the patch following the geodesic distance criterion may change. For

the same reason, the system proposed is not resistant to re-sampling attacks, which can add or remove points in the mesh.

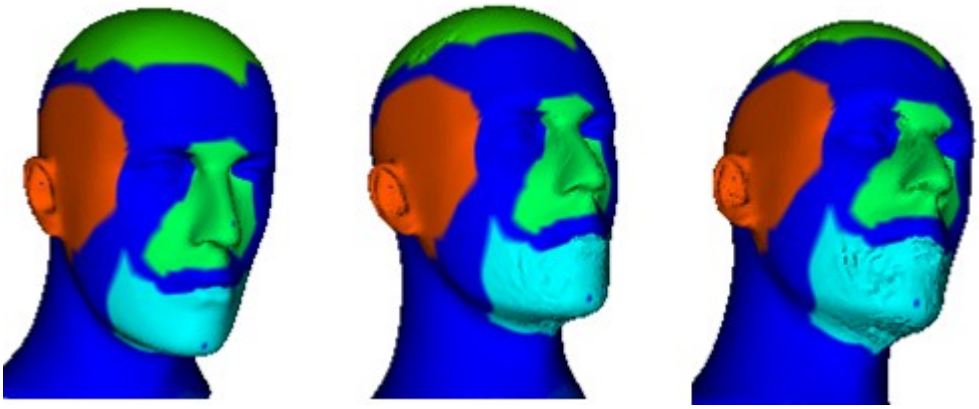


Figure 7.15. In the left, patches constructed in the original Head model. In the middle and in the right, local distortion introduced by the watermarking scheme in patches of the Head model for insertion forces of 0.02 and 0.07 respectively. The model in the middle respects the invisibility criterion.

Watermark extraction

As we previously pointed out, in detection we perform the detection of prongs and construct again their patches. Afterwards, watermark extraction is done patch by patch through the QIM extraction algorithm.

The correctness of the system is evaluated by regarding the BER results for every patch and calculating a weighted BER which takes into account the number of embeds of the watermark string in each patch. Figure 7.16 illustrates a scheme of the watermarking decoding system.

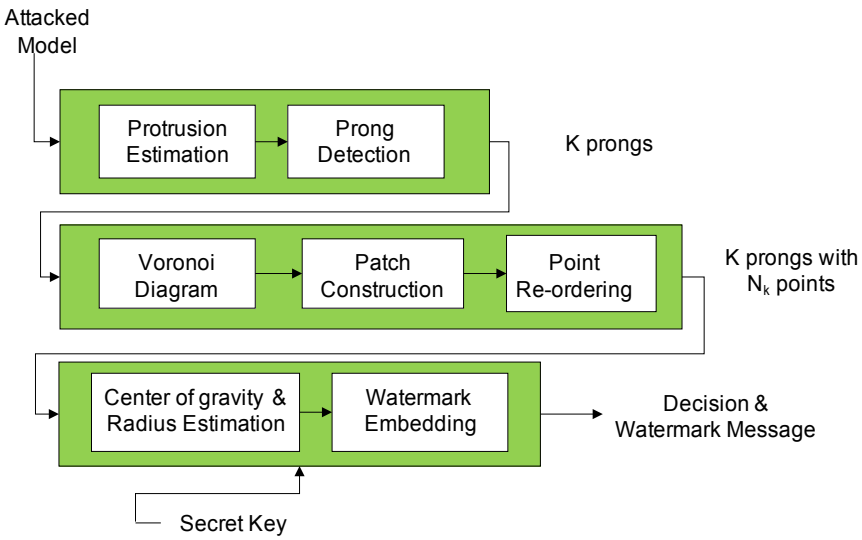


Figure 7.16. Proposed scheme of the feature points detection, patch construction and watermark retrieval in decoding.

7.6. Applying visual mask to the scheme

Visual masking previously presented (Chapter 6) is applied in this section to improve the imperceptibility of the watermark. The perceptual model is calculated from the original model in encoding. The embedding of the watermark is weighted with the visual mask in every point of each patch. We can see images of the difference when using the perceptual mask in Fig. 7.17 and Fig. 7.18.

Figure 7.17. The original model and watermarked model with and without visual masking for the David Head model when using the same watermarking strength (Δ=0.9). From left to right, original David Head model, watermarked David Head model using the perceptual mask and watermarked David Head model without perceptual masking. We can observe that the perception of the watermark is more noticeable when we do not apply the visual mask. The watermarking strength have been selected so that the distortions due to watermark are visible.

Figure 7.18. Same results than Fig. 7.17 for the Head model (Δ=0.2). As in Fig. 7.17, the difference when we do not apply the visual mask is noticeable. The watermarking strengths have been selected so that the distortions due to watermark are visible.

We have tested the imperceptibility of the proposed encoding scheme when using the perceptual model for several meshes under different watermark strengths. We use the MSDM distance, previously presented in Section 2.3.1, to measure the difference between the original and the watermarked model. From obtained results we observe that the MSDM distance between both meshes increases along with the watermark strength. Fig. 7.19 gives the MSDM distances for the Bunny and Dinosaur models using different strengths factors. Fig. 7.20 illustrates the progression of the MSDM distances with and without visual mask for Head and Head Dragon models.

Model	Watermark strength	MSDM with visual mask	MSDM without visual mask	Model	Watermark strength	MSDM with visual mask	MSDM without visual mask
Bunny	0,05	0,0647729	0,0893555	Dinosaur	0,05	0,0838624	0,0670266
	0,1	0,112522	0,136868		0,1	0,127431	0,130675
	0,2	0,169447	0,215327		0,2	0,186997	0,289273
	0,3	0,221071	0,254739		0,3	0,288427	0,401752
	0,4	0,253414	0,300318		0,4	0,353727	0,44441
	0,5	0,279136	0,336017		0,5	0,447371	0,482657
	0,6	0,310215	0,363313		0,6	0,44382	0,508213

Figure 7.19. MSDM comparison with and without visual mask for Bunny and Dinosaur meshes.

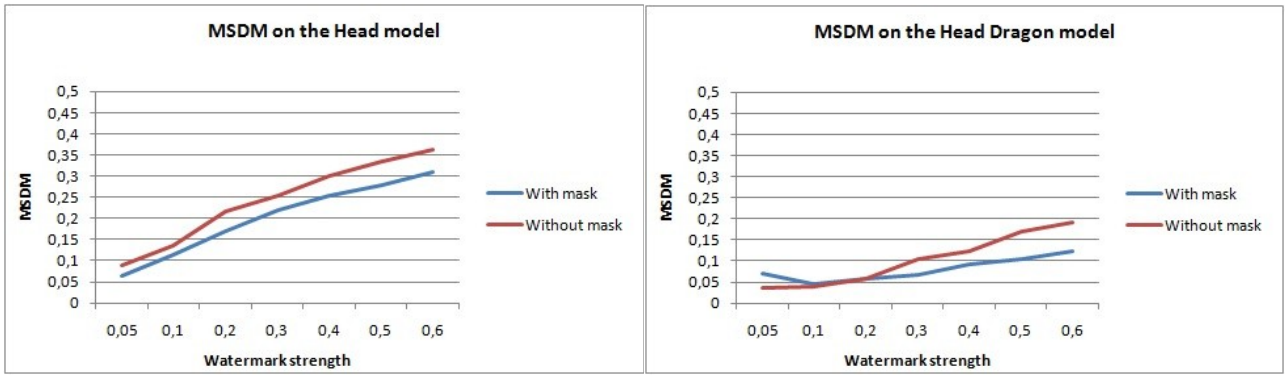


Figure 7.20. MSDM evolution when using different watermark strengths for the Head model, at the left, and the Head Dragon model, at the right.

7.7. Conclusions

This Chapter has presented a new approach for 3D models watermarking. The robustness of the selected feature points have been shown for detecting the same points of a 3D object in case of RST, re-ordering, noising and smoothing attacks. The integration of the QIM in the scheme provides its good properties to the watermark, in particular its robustness against noising and smoothing attacks. Selected feature points present also robustness against cropping attacks, if the specific point is far enough from the crop. This facts brings to the algorithm robustness against cropping up to a point, depending on the attacked mesh. This issue will be more deeply analysed in the next Chapter.

However, we have pointed out the fact that the construction of the QIM does not present robustness against re-sampling attacks. For instance, the encoding and the decoding processes are relied to specific points in the mesh. In case of simplification or remeshing we will not be able to detect the same patch where the watermark is embedded. In the next Chapter, some tests have been done to test the robustness of the proposed method.

Moreover, the increase on the robustness due to the insertion of the watermark on several patches of the mesh leads to a limitation in the capacity of the scheme. For instance, the capacity will be constrained by the size of the smaller patch we consider. In applications where the need of a high capacity system is required this problem could be solve by discarding those patches encountered which do not accomplish a minimum size specified.

Finally, the usefulness of the use of a perceptual-quality-oriented protocol has been evaluated when applying the visual mask presented in Section 6.2 to the patches watermark process.

Chapter 8

Experimental Results

8.1. Introduction

This Chapter presents and discusses the results obtained when testing the algorithm proposed in Chapter 7. The results of the prongs robustness have been already presented and proved in the precedent Chapter, as well as the perception of the scheme when applying the perceptual visual mask.

The results obtained when implementing the blind watermarking scheme are not as good as we expected in principle when applying noise addition and smoothing attacks. The reasons of this behaviour are next discussed.

In [DHM10], the algorithm presented by Darazi et al. uses all the asset to embed the watermark. However, the decoding depends on the recovery of the center of gravity from the original content to properly retrieve the message. In the scheme we propose the embed the watermark occurs in the patches constructed from robust detected prongs. As a matter of fact, the missing robustness against cropping of the shceme presented in [DHM10] is clearly improved. Unfortunately this solutions leads to a lower number of points containing the message (capacity losses) and thus to a lower number of embeddings. This fact has a direct implication in the robustness of the scheme.

In order to retrieve the watermark message, the decoder makes use of the center of gravity of each reconstructed patch in the receiver. Intuitively we can determine that the accuracy of those points decreases along with the number of points related.

As previously said the ordering for points inside the patch in encoding and decoding processes is assigned progressively on the increasing rings depending on the geodesical distance to each point of the ring towards the prong. This step provides robustness against re-ordering attacks to the system, as compared to the work presented in [DHM10], which depends on the order of the points inside the segments conveying the message. Nevertheless, the positions of the mesh points vary

when the watermarked asset undergoes alterations. Hence the geodesical distances in decoding may have changed after manipulations conducted in the attack (especially when adding noise).

For these reasons, we have implemented a watermarking scheme which uses as side information the center of gravity of constructed patches and we have tested its usability and robustness against noise addition, Laplacian smoothing, RST, re-ordering and cropping attacks. General schemes of the watermark embedding and decoding system are shown in Fig 8.1 and Fig 8.2, respectively.

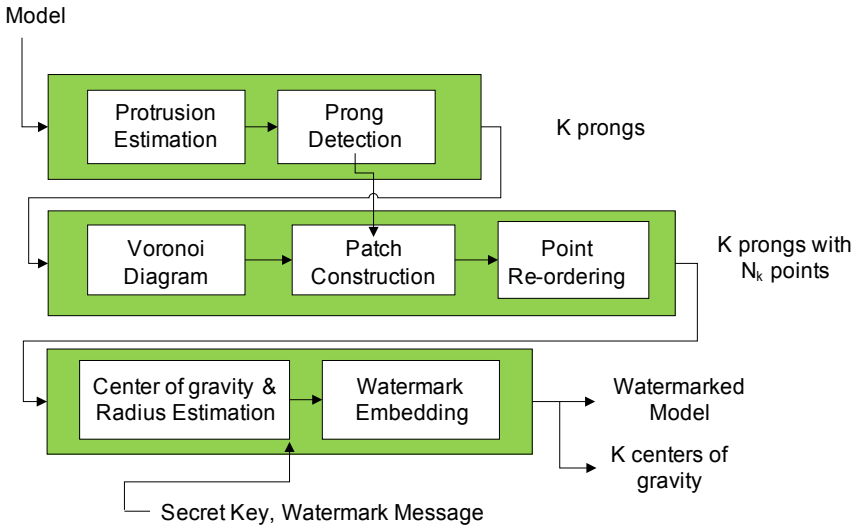


Figure 8.1. Scheme of the watermark embedding with side information.

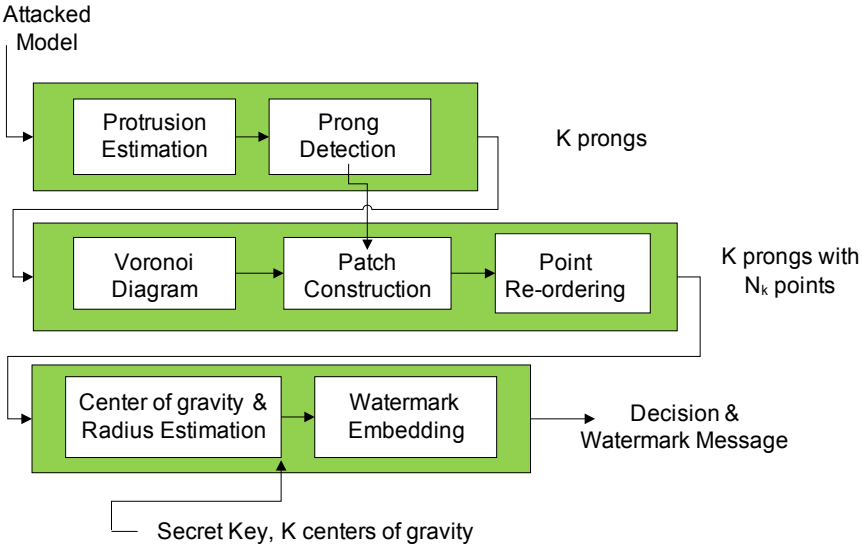


Figure 8.2. Scheme of the watermark decoding with side information.

Since the improvement when using the visual mask and the involving loss in capacity has been evaluated in Chapter 7 and Chapter 5, respectively, the tests have been developed using the

perceptual model. Different original models are illustrated in Fig. 8.3. In Fig. 8.4 we can see respective watermarked contents with minimal values for the watermark strength (the watermark is not visible).

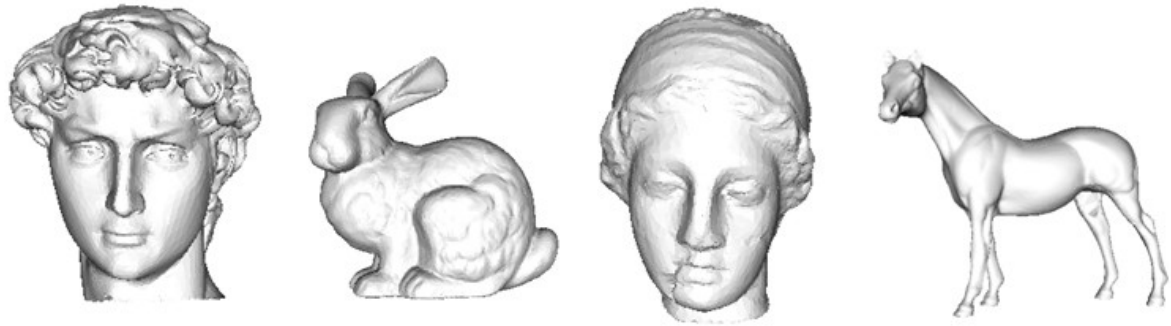


Figure 8.3. The original non-watermarked meshes. From left to right: David Head, Bunny, Venus and Horse.

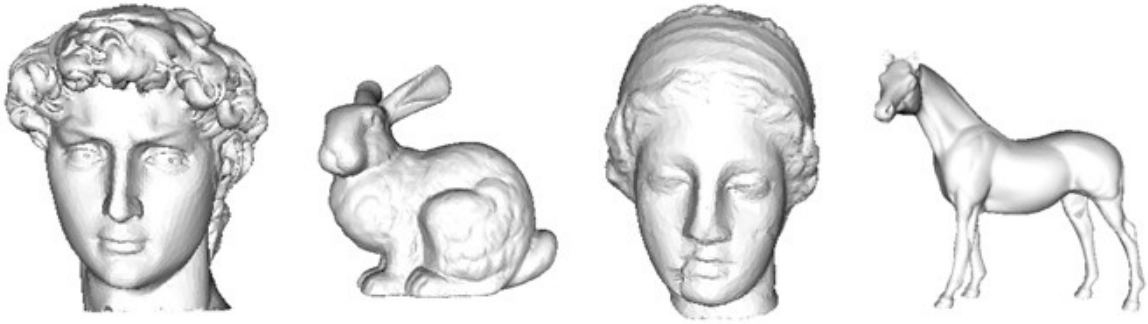


Figure 8.4. The watermarked meshes which follow the perceptible quality criterion. From left to right: David Head (Δ = 0.2), Bunny (Δ = 0.08), Venus (Δ = 0.08) and Horse (Δ = 0.08).

The Chapter is organized in the following way. Experimental results when applying a noising and a smoothing attack are presented in the first sections and compared to the results obtained when using the system presented by Darazi et al. in [DHM10]. In the following sections, resistance to RST attacks, re-ordering and cropping is discussed.

8.2. Robustness against noise addition

We perform some tests to test the robustness of the method we proposed against noise addition attack. In Fig. 8.5 the David Head model with different strengths of the attack is presented.

We run some tests using different parameters for the strength factor, which will directly have a repercussion in the visual perception, as well as the noise factor. Fig. 8.6 illustrates different obtained BER for some of the models. It is possible to observe that the behaviour in case of noise

addition depends on each mesh feature, likewise the perception of the attack is linked to different noise ratios depending on the shape features.

Figure 8.5. From left to right, original David Head model, the same model attacked with a Gaussian noise (NR = 0. 002) and the model attacked with a more strong noise (NR =0.004).

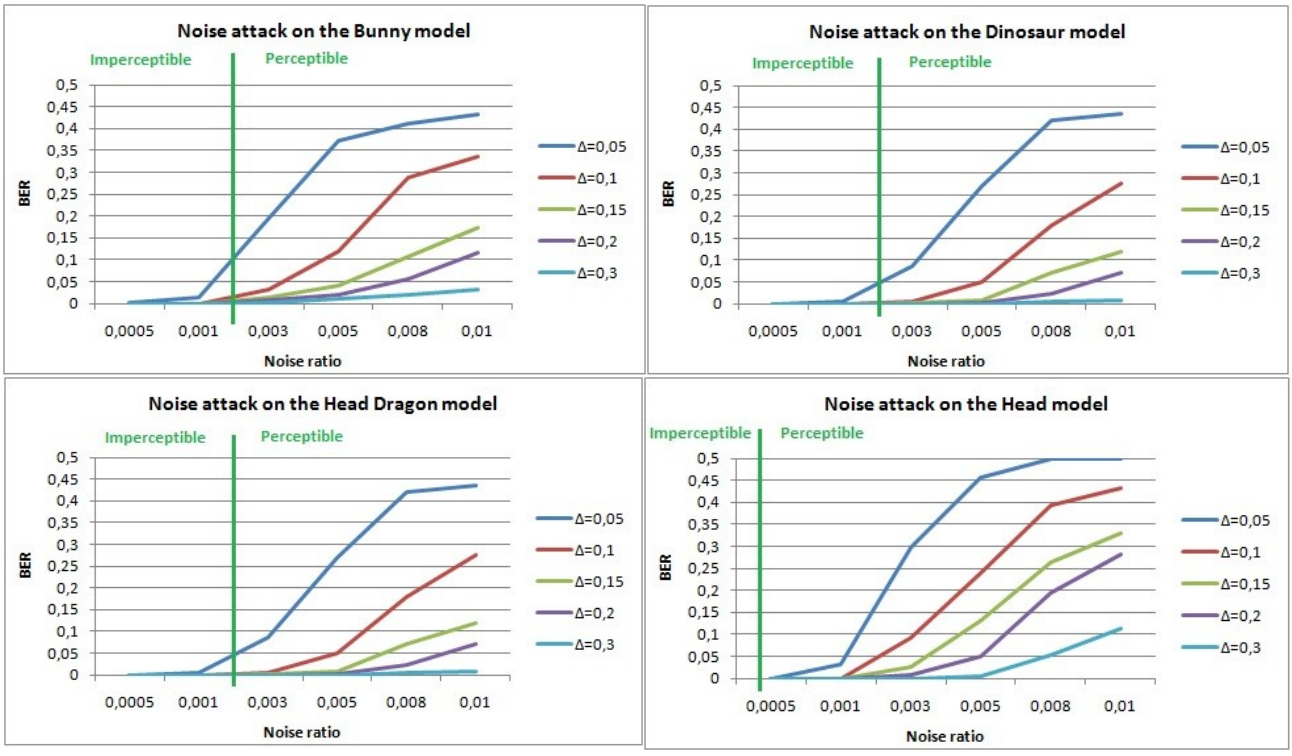


Figure 8.6. Obtained BER for the Bunny, Dinosaur, Head Dragon and Head models for different strengths of the watermark and different Noise Ratios (NR). The green line indicates the NR value from which the noising attack is visible.

As previously said, the strength of the attack and the watermark affect in a different way different meshes. Fig. 8.7 illustrates a comparison of obtained BER for different 3D meshes when using the same watermark strength for the insertion (Δ=0.1). From the results we conclude that those models where the watermark is visible for a lower value of Δ are at the same time visually affected from lower values of the noise attack.

From Fig. 8.6 we can see that results are quite good, since our main purpose is to be able to resist to visually perceptive attacks. Since the construction of the patches is constrained to be contained inside Voronoi regions, the number of insertions highly varies from one patch to another. For instance, in the Dinosaur model the watermark embedding occurs 32 times in the biggest patch, whereas the smallest one contains only 9 repetitions. For this reason, the calculated BER and the values represented are weighted depending on the influence of the specific patch. The retrieve of the watermark would be done by performing a maxima criterion over each bit of the message.

To expound a little bit more our results, we test the obtained BER for some models for different strengths attacks when using a watermark strength which is not perceptually visible. We observe that for these minimal Δ values the watermark system is able to resist against imperceptible attacks. Results are resumed in Fig. 8.8.

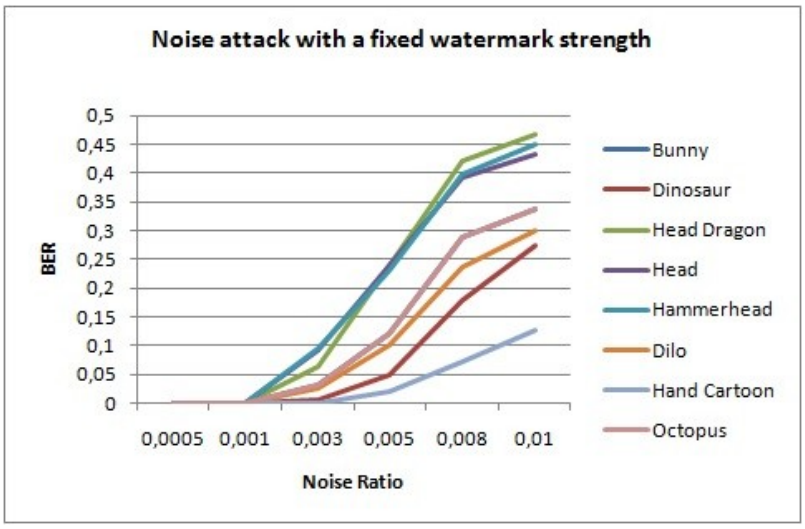


Figure 8.7. Obtained BER in several meshes for different strengths of the noising attack when using a fixed watermark strength (Δ=0.1).

Compared to the results obtained when using the model proposed in [DHM10], the robustness against this attack is much lower. The reasons are the use of less number of points to convey the message and thus the number of insertions on one hand, and the accuracy loss in the recalculation of the center of gravity of each patch as compared to the recalculation of the center of gravity of the whole model.

Moreover, the alteration of the vertex position due to the attack can lead to a different distribution of Voronoi regions along the mesh if the strength of the attack is enough high. These alterations can also lead to a difference in the id's assign process inside a patch, since the geodesical distances may change.

MODEL	Δ	MSDM	PRONGS	CAPACITY	NR	BER	PERCEPTION
	0.1	0.113	6	4032/14007	0.001	0	Imperceptible
					0.003	0.031	Perceptible
					0.005	0.120	Perceptible
					0.008	0.288	Perceptible
	0.1	0.127	6	8064/14070	0.001	0	Imperceptible
					0.003	0.004	Perceptible
					0.005	0.050	Perceptible
					0.008	0.180	Perceptible
	0.1	0.046	6	3328/19119	0.001	0	Imperceptible
					0.003	0.062	Perceptible
					0.005	0.241	Perceptible
	0.05	0.065	6	6400/11703	0.0005	0	Imperceptible
					0.001	0.035	Perceptible
					0.003	0.296	Perceptible

Figure 8.8. Obtained BER when applying noise addition to several meshes using a watermarking strength Δ for which the watermark is not visible. The watermark message length is of 64 bits. The MSDM measure shows the similarity between the original and the watermarked assets. The prongs detection has been done when using a Neighborhood = 12 (12-ring). The capacity column illustrates the ratio between the number number of points where the watermark has been embedded and the total number of points of the mesh. The NR values have been chosen to show the minimal case where the attack is not perceptible and cases where the system does not work properly. The perception column indicates whether the noise attack is visible or not.

8.3. Robustness against Laplacian Smoothing

In this section we present the results obtained when models undergo a Laplacian smoothing attack [T95]. We can see the output of this attack in illustrations shown in Fig. 8.9, where the smoothing is applied in the Bunny model with several strengths.

Figure 8.9. From left to right, original Bunny model, the same model attacked with a Laplacian smooth (smoothing level = 0. 05) and the model attacked with more iterations for the smoothing (smoothing level =0.1).

Tests are performed following the same way of the precedent chapter. Several smoothing levels are chosen to attack the mesh and evaluate the resultant BER of the system when using different

strengths for the watermark embedding. As we have seen the case when presenting the noising attack, the impact of the smoothing depends on the specific model in terms of both perception of the attack and correctness of the BER. These results are given in Fig 8.9 for the Bunny, Dinosaur, Head Dragon and Head model.

In the same way than in the noising attack, the strength of the attack has a different effect depending on the used model to perform a test. Fig. 8.10 illustrates the operation of the whole scheme for several 3D meshes when using a fixed value of the watermark strength (Δ=0.1). As we concluded in the precedent section, models where the watermark is visible for a lower value of Δ are at the same time visually affected from lower values of the noise attack.

From Fig. 8.9 we deduce that our scheme works properly if the smoothing attack is not visually perceived. To enter in more details, we run some simulations for strengths of the watermark where the fidelity constraint is fulfilled. The strengths of the attacks have been chosen in order to show cases in which the scheme does not work properly. Results are summarized in Fig. 8.11.

As we have already discussed in the previous section, the robustness against smoothing attack is lower as compared to the scheme proposed in [DHM10]. This fact can be explained in the same way than in the case of noising attack.

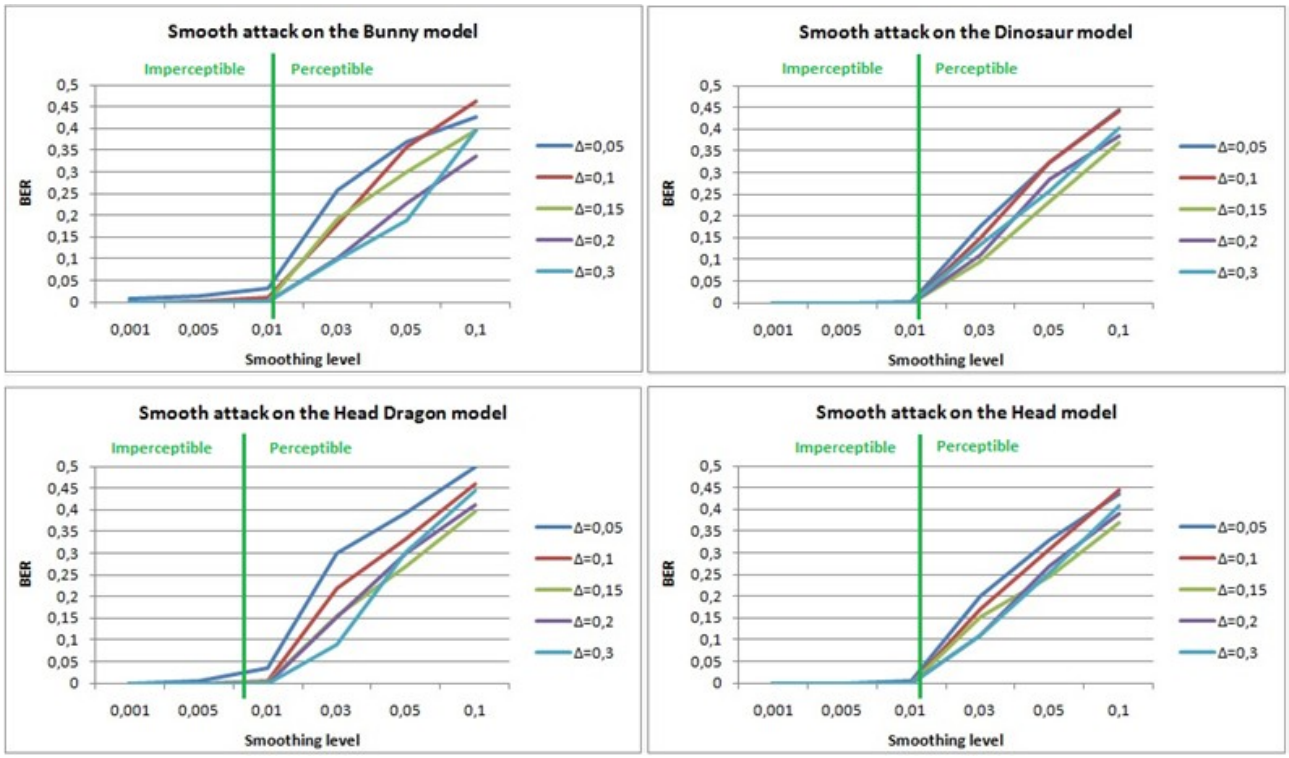


Figure 8.10. Obtained BER for the Bunny, Dinosaur, Head Dragon and Head models for different strengths of the watermark and different smoothing levels of the Laplacian Smoothing attack. The green line indicates the smoothing level value from which the noising attack is visible.

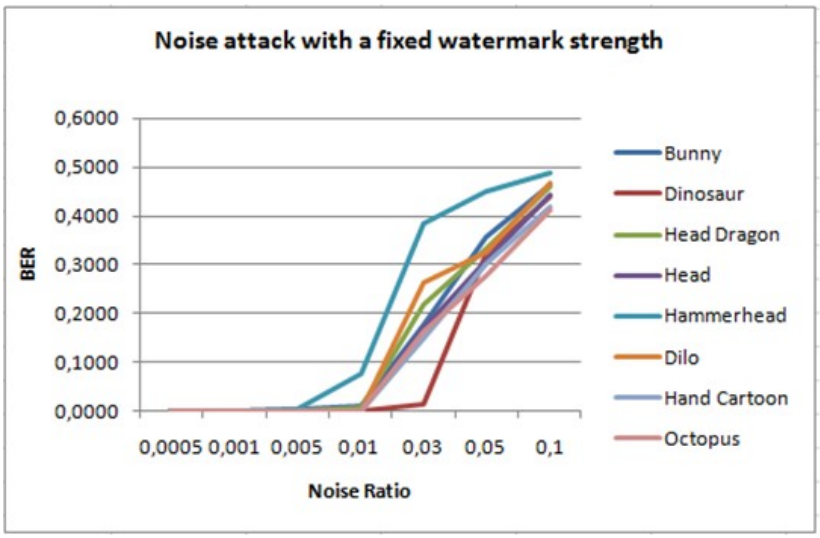


Figure 8.11. Obtained BER in several meshes for different strengths of the Laplacian smoothing attack when using a fixed watermark strength (Δ=0.1).

8.4. Robustness against RST, Reordering and Cropping attacks

As we have discussed in Chapter 7, the designed blind scheme presents high robustness to Rotation, Scaling and Translation attacks (RST). We have proven this fact by testing the developed algorithm under several values of these three attacks, resolving in a robust method against these geometry manipulations.

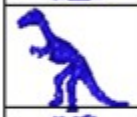
MODEL	Δ	MSDM	PRONGS	CAPACITY	Smooth	BER	PERCEPTION
	0.1	0.113	6	4032/14007	0.01	0.010	Imperceptible
					0.03	0.178	Perceptible
					0.05	0.357	Perceptible
	0.1	0.127	6	8064/14070	0.01	0	Imperceptible
					0.03	0.149	Perceptible
					0.05	0.323	Perceptible
	0.1	0.046	6	3328/19119	0.01	0.005	Imperceptible
					0.03	0.218	Perceptible
					0.05	0.334	Perceptible
	0.05	0.065	6	6400/11703	0.005	0.003	Imperceptible
					0.01	0.201	Perceptible
					0.03	0.327	Perceptible

Figure 8.12. Obtained BER when applying Laplacian smoothing to several meshes using a watermarking strength Δ for which the watermark is not visible. The watermark message length is of 64 bits. The MSDM measure shows the similarity between the original and the watermarked assets. The prongs detection has been done when using a Neighborhood = 12 (12-ring). The capacity column illustrates the ratio between the number number of points where the watermark has been embedded and the total number of points of the mesh. The smoothing level values have been chosen to show the minimal case where the attack is not perceptible and cases where the system does not work properly. The perception column indicates whether the smoothing attack is visible or not.

As it could be seen, keeping the positions of the centers of gravity as side information at the detection side can improve the robustness for noise dans smoothing. Nevertheless, this robustness is affected when implementing the side-information system. For instance, if a rotation, scaling or translation is achieved, the original positions of the centers of gravity become useless as they are de-synchronized.

One way to solve this issue is to also keep the position of the prongs as side information. At the decoding side, the prongs of the attacked mesh are computed and can be compared to the positions of the original ones. Simple algorithms can iteratively re-align the original prongs with the new ones such as PCA (Principal Component Analysis) or ICP (Iterative Closest Point algorithm) [SE09]. The position of the centers of gravity can then be re-synchronized thanks to the prongs. This keeps then the robustness to noise and smoothing as well as the robustness to rotation, scaling and translation.

Re-ordering attack changes the position of two adjacent-id points and reassigns properly the defined triangles in the VTK file, which is the format of the models we have used to test the operation of the scheme. Results obtained show that the implemented system provides good resistance to the attack, if the watermark strength Δ respects the position of the points in a measure enough which involves the same Voronoi regions distribution. The operation shows that for values of Δ for which the watermark is not visible the algorithm works in a proper way and the watermark is correctly retrieved.

The operation of the whole scheme under the cropping attacks has been also tested. We observe that the robustness offered in terms of detection of the prongs and neighborhoods is good if the cut is farther enough from the real prongs. As a matter of fact, a crop near enough to a prong would influence in the recalculation of the protrusion and thus in the local maxima detection output.

An example of this behaviour is shown in Fig. 8.12 and Fig. 8.13. In Fig. 8.12 we can see the protrusion and prong detection of the Head Dragon model, and the attacked model where the claw has been cropped. In this case, the recalculation of the protrusion function leads to a proper recovery of prongs and neighborhood. In detection the watermark retrieved from those two patches is the correct one (BER=0 for both patches).

Fig. 8.13 illustrates the same model for which the body of the dragon has cut and there is only the head remaining. We can observe in the figure that the performed cropping attacks the mesh in a place near to the prongs. The calculation in decoding of the protrusion leads to a correct detection of one of the prongs remaining, while the other one is nor properly recovered. In this case only one up to the two patches leads to a reading with BER=0.

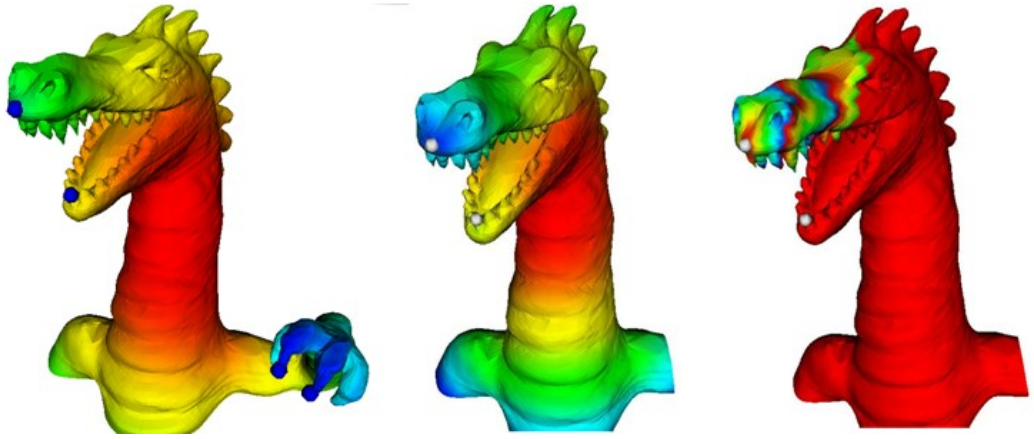


Figure 8.13. From left to right, the protrusion and prong detection on the original Dragon Head model; protrusion and prong detection on the the same model attacked with cropping; neighborhood of one of the prongs in the decoder. Both prongs of the head are properly detected. In the calculation of the protrusion a neighborhood of 15 (15-ring) has been specified.

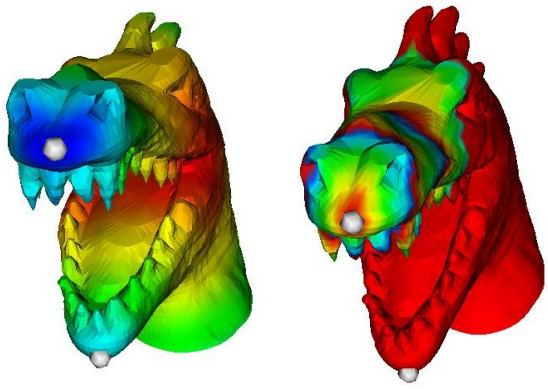


Figure 8.14. Cropping attack by a vertical plane on the Dragon Head model. The image on the left shows the protrusion and local maxima detection performed in the decoder. Only one of the two detected prongs is properly calculated. The image on the right illustrates the neighborhood of the correct prong, which leads to a correct reading of the watermark (BER=0).

8.5. Conclusions

In this chapter we have presented the operation of the contribution of this Master thesis. The method presented in Chapter 7 has been tested progressively to evaluate its fragilities. When performing a blind detection we can see that the recalculation of the center of gravity of patches is not properly recovered and the consequent watermark is not correctly retrieved.

For this reason a scheme using side information has been presented. In such case the integration of QIM and robust feature detection constitutes an effective system. The knowledge in the decoder of these centers of gravity and the position of prongs calculated in the insertion process

resolves in a robust scheme which can protect the watermark against noising, smoothing, RST, reordering and cropping attacks, in cases of interests (when the attack can not be perceived).

The proposed feature detection method in decoding is also robust against local deformations and cropping, which are very difficult to handle for watermarking methods, mainly due to the insertion of the message from different starting points that are correctly recovered, even in the case of a blind scheme (see Section 7.2).

The detailed results are still preliminary and the whole scheme needs some improvements and optimizations to improve its robustness.

A possible improvement consists in defining a minimum size for patches in terms of number of points which can convey the watermark. As a matter of fact, the obtained BER during the tests showed a strong correlation between the number of insertions of the message and the correctness of the BER. The accuracy of the recalculation of the center of gravity in those patches is also lower, since the algorithm is performed from less points. Moreover, small patches restrict the capacity of the whole scenario.

Chapter 9

Conclusions and Future Work

9.1. Introduction

In this master thesis, a new 3d watermarking scheme has been proposed by combining two state-of-the-art techniques which turn to be very complementary and extending them to the usage of a perceptual visual mask. The advantages and disadvantages of each part of the scheme have been discussed in detail and experimentally tested. The developed scheme requires more enhancements in order to provide a proper behaviour to future feasible scenarios.

The present chapter outlines the most relevant aspects than can be completed in future revisions of the current work. It starts with the description of the achieved improvements of the integration of both techniques, and continues with a discussion about some problems found in the operation of the system. Finally, a brief description of the future work is provided.

Although the designed and tested scenario fits some of the initial expectations, a few problems have appeared. Some of them have been previously commented, but deserve being mentioned together with solving proposals. After observing the operation of the designed and implemented algorithm, it is possible to point out a set of possible lines of work and feasible improvements.

Those proposals to solve the problems that have appeared and to improve the operation and possibilities of the proposed scheme aim to improve the current implementation by achieving a more robust application in future versions.

9.1. Conclusions

Broadly speaking, the extension has been successful at combining QIM and prong features for re-synchronization together with a visual mask. Compared to QIM, the new scheme is able to resist to

more attacks i.e. re-ordering and cropping but due to the fact that the number of points present in patches is naturally smaller than in the case of the whole mesh embedding, the robustness against noise and smoothing attacks is slightly reduced. When compared to feature-based re-synchronization, the robustness to noise and smoothing is better with the usage of the visual mask but side information is necessary for optimal results.

Globally the new scheme is therefore able to resist to more attacks at the cost of a slightly reduced robustness to noise and smoothing. Side information clearly improves the robustness scheme compared to totally blind detection because center of gravity based methods usually are fragile against any modification in the position of this center of gravity. Storing center of gravities together with prongs and some encoding parameters leads to optimal robustness and necessitates less information than informed detection where the whole original unwatermarked (thus unprotected) mesh is necessary at the detection side.

9.3. Future work

Future work concern many parts of the overall scheme. First of all, the curvature and roughness different existing measures could be deeper studied and subjective tests could be elaborated in order to determine which of them are more relevant for the human vision. Moreover, the visual mask could be extended to other features of the content of a mesh rather than only roughness and curvature. Secondly, the choice of other quantizers could be tested in the QIM algorithm particularly when embedding small patches where there are not enough points for repeating the embedded bits.

Another aspect that could be studied is how to extend this algorithm to the robustness against decimation and subdivision (a.k.a. re-sampling) attacks. The current problem is that radius-based QIM as such is not robust to re-sampling while prongs and histogram-based methods can provide such robustness. Maybe the use of QIM on the coefficients of another transform could provide this robustness when combined with prongs. The links between sampling and the robustness of each method or step of the algorithm seem to be interesting to explore.

Moreover, the scheme could maybe be combined with multiple watermark techniques so that the entirety of the mesh is protected rather than only the prong-based patches. Finally, the usage of side information and its security could be studied in more details.

Bibliography

[AG05] P. Alliez and C. Gotsman, "Recent advances in compression of 3D meshes," in Advances in Multiresolution for Geometric Modelling, pp. 3-26, 2005.

[AS03] A. Adelsbach and A. Sadeghi. Advanced techniques for dispute resolving and authorship proofs on digital works. In Proc. of SPIE, Security and Watermarking of Multimedia Contents V, San Jose, CA, USA, pages 677–688, January 2003.

[ASCE02] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. Mesh: Measuring error between surfaces using the hausdorff distance. In Proc. of the IEEE International Conference on Multimedia and Expo 2002 (ICME), volume I, pages 705–708, 2002.

[AUG06] P. Alliez, G. Ucelli, C. Gotsman, and M. Attene. Shape Analysis and Structuring, De Floriani, L., Spagnuolo M. (eds.), chapter Recent Advances in Remeshing of Surfaces. Springer-Verlag, 2006.

[B99a] O. Benedens. Geometry-based watermarking of 3d models. IEEE Computer Graphics and Applications, 19(1):46–55, January 1999.

[B99b] Benedens, O.: Two High Capacity Methods for Embedding PublicWatermarks into 3D Polygonal Models. In: Proceedings of the Multimedia and Security Workshop at ACM Multimedia, pp. 95–99 (1999).

[BB04] Barni M., Bartolini F.: Watermarking Systems Engineering: Enabling Digital Assets Security and other Applications. Marcel Dekker Inc., 2004.

[BBCP98] F. Bartolini, M. Barni, V. Cappellini, and A. Piva, \Mask building for per ceptually hiding frequency embedded watermarks," in Proceedings of the 5th IEEE International Conference on Image Processing, ICIP98, vol. I, Chicago, IL, USA, Oct. 1998, pp. 450-454.

[BM98] M.R. Bolin and G.W. Meyer. A perceptually based adaptive sampling algorithm. In Proc. of SIGGRAPH'98, pages 299–309, 1998.

[CDSM04] Cayre, F., Devillers, O., Schmitt, F., Maitre, H.:Watermarking 3D Triangle Meshes for Authentication and Integrity, Research Report of INRIA, p. 29 (2004).

[CGE05] M. Corsini, E.D. Gelasca, and T. Ebrahimi. A multi-scale roughness metric for 3-d watermarking quality assessment. In Workshop on Image Analysis for Multimedia Interactive Services 2005, April 13-15, Montreux, Switzerland, 2005.

[CKPCJ05] Cho, J.W., Kim, M.S., Prost, R., Chung, H.Y., Jung, H.Y.: Robust Watermarking on Polygonal Meshes Using Distribution of Vertex Norms. In: Proceedings of the International Workshop on Digital Watermarking, pp. 283–293 (2005).

[CLLP05] Cho, W.H., Lee, M.E., Lim, H., Park, S.Y.: Watermarking Technique for Authentication of 3-D Polygonal Meshes. In: Proceedings of the International Workshop on Digital Watermarking, pp. 259–270 (2005).

[CM03] Cayre, F., Macq, B.: Data Hiding on 3-D Triangle Meshes. IEEE Transactions on Signal Processing 51(4), 939–949 (2003).

[CRS98] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error onsimplified surfaces. Computer Graphics Forum, 17(2):167–174, 1998.

[CRSMM03] Cayre, F., Alface, P.R., Schmitt, F., Macq, B., Maitre, H.: Application of Spectral Decomposition to Compression and Watermarking of 3D Triangle Mesh Geometry. Signal Processing 18(4), 309–319 (2003).

[CW01] B. Chen and G. W. Wornell. Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. IEEE *Trans. On Information Theory*, Vol. 47, No. 4, p.p. 1423-1443, May 2001.

[DHM10] R. Darazi, R. Hu and B. Macq. Applying Spread Transform Dither Modulation for 3D- Mesh Watermarking by using Perceptual Models. Communications and Remote Sensing laboratory, Université Catholique de Louvain, 2010.

[EG01] J. Eggers and B. Girod. Blind watermarking applied to image authentication. In Proc. of

International Conference on Audio, Speech and Signal Processing, Salt-Lake City, USA, May 2001.

[EFS07] Mahsa Eshraghi , Faramarz F. Samavati, 3D watermarking robust to accessible attacks, Proceedings of the First International Conference on Immersive Telecommunications, October 10-12, 2007, Bussolengo, Verona, Italy.

[EK03] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. IEEE Transactions on PAMI, 25(10):1285–1295, 2003.

[F02] T. Furon. Use of watermarking techniques for copy protection. PhD thesis, Ecole Nationale des Télécommunications, Rennes, 2002.

[FDFH90] J. Foley, A. van Dam, S. Feinera, and J. Hughes. Computer Graphics. Principle and Practice. Addison Wesley, Reading, MA., 1990.

[FSPG97] J.A. Ferwerda, P. Shirley, S.N. Pattanaik, and D.P. Greenberg. A model of visual masking for computer graphics. In Proc. of SIGGRAPH' 97, pages 143–152, 1997.

[GAGM09] Joachim Giard, Patrice Rondao Alface, Jean-Luc Gala, Benoît Macq, "Fast Surface-Based Travel Depth Estimation Algorithm for Macromolecule Surface Shape Description," IEEE/ACM Transactions on Computational Biology and Bioinformatics, 18 May. 2009. IEEE computer Society Digital Library. IEEE Computer Society.

[GM09] J. Giard and B. Macq. Fast geodesic distance approximation using mesh decimation and front propagation, in Proceedings of the SPIE, IS&T/SPIE Electronic Imaging 2009, San Jose, CA, 2009, pp. 72450B-72450B.

[GECB05] Gelasca, E.D., Ebrahimi, T., Corsini, M., Barni, M.: Objective Evaluation of the Perceptual Quality of 3D Watermarking. In: Proceedings of the International Conference on Image Processing, vol. 1, pp. 241–244 (2005).

[HRAM09] Roland Hu, Patrice Rondao-Alface, Benoit M. Macq: Constrained optimization of 3D polygonal mesh watermarking by quadratic programming. ICASSP 2009: 1501-1504.

[IRS03] I. Ivrissimtzis, C. Rôssl, H.P. Seidel. Tree-based Data Structures for Triangle Mesh Connectivity Endocing. Max-Planck-Institut für Informatik (2003).

[JDBP04] Jin, J.Q., Dai, M.Y., Bao, H.J., Peng, Q.S.: Watermarking on 3D Mesh Based on SphericalWavelet Transform. Journal of Zhejiang University: Science 5(3), 251–258 (2004).

[KDK98] Kanai, S., Date, H., Kishinami, T.: Digital Watermarking for 3D Polygons Using Multiresolution Wavelet Decomposition. In: Proceedings of the International Workshop on Geometric Modeling: Fundamentals and Applications, pp. 296–307 (1998).

[KG00] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In Computer Graphics (Proc. of SIGGRAPH'00, pages 279–286, 2000.

[KLS96] R. Klein, G. Liebich, and W. Strasser. Mesh reduction with error control. In Proc. of IEEE Visualization, pages 311–318, 1996.

[KLT05] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. The Visual Computer, 21(8-10):649–658, 2005.

[KS98] R. Kimmel and J.A. Sethian. Computing geodesic paths on manifolds. Proc. of National Academy of Science, 95(15):8431–8435, 1998.

[KS04] Kraevoy, V., Sheffer, A.: Cross-parameterization and compatible remeshing of 3D models. ACM Trans. Graph. 23(3), 861–869 (2004).

[KSG03] Kraevoy, V., Sheffer, A., Gotsman, C.: Matchmaker: constructing constrained texture maps. ACM Trans. Graph. 22(3), 326–333 (2003).

[KT96] A. D. Kalvin and R. H. Taylor. Superfaces: Polygonal mesh simplification with bounded error. Computer Graphics and Applications, IEEE, 16(3Ã˘a):64–77, May 1996.

[KT03] M. Kuribayashi and H. Tanaka. A watermarking scheme applicable for fingerprinting protocol. In LNCS, editor, Proc. of Int. Workshop on Digital Watermarking, IWDW'03, Springer-Verlag, pages 532–543, 2003.

[L98] J.-P. Linnartz. Method and system for transferring content information and supplemental information relating thereto. Patent WO 98/33325, 1998.

[L07] Lavoué, G., A Roughness Measure for 3D Mesh Visual Masking, ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization, Tübingen, Germany, July 2007.

[LDBE06] Lavoué, G., Gelasca, E.D., Dupont, F., Baskurt, A., Ebrahimi, T.: Perceptually Driven 3D Distance Metrics with Application to Watermarking. In: Proceedings of the SPIE Applications of Digital Image Processing, p. 63120 (2006).

[LDGDBE06] LAVOUé, G., DRELIE GELASCA, E., DUPONT, F., BASKURT, A., AND EBRAHIMI, T. 2006. Perceptually driven 3D distance metrics with application to watermarking. In SPIE Applications of Digital Image Processing XXIX. Vol. 6312.

[LLLL05] Lin, H.S., Liao, H.M., Lu, C., Lin, J.: Fragile Watermarking for Authenticating 3-D Polygonal Meshes. IEEE Transactions on Multimedia 7(6), 997–1006 (2005).

[LPRM02] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. In ACM SIGGRAPH Conference Proceedings, pages 362–371, 2002.

[LT00] P. Lindstrom and G. Turk. Image-driven simplification. ACM Transactions on Graphics, 19(3):204–241, 2000.

[M00] Alexa, M.: Merging polyhedral shapes with scattered features. Visual Comput. 16, 26–37 (2000).

[MKLTDH00] M. Maes, T. Kalker, J.-P. Linnartz, J. Talstra, G. Depovere, and J. Haitsma. Digital watermarking for dvd video copy protection. Signal Processing Magazine, 17(5), September 2000.

[ME04] Maret, Y., Ebrahimi, T.: Data Hiding on 3D Polygonal Meshes. In: Proceedings of the Multimedia and Security Workshop, pp. 68–74 (2004)

[NTTS01] A. Nikolaidis, S. Tsekeridou, A. Tefas, and V. Solachidis. Asurvey on watermarking application scenarios and related attacks. In Proc. Of International Conference of Image Processing, volume 3, pages 991–994, 2001.

[OMA97] Ohbuchi, R., Masuda, H., Aono, M.: Watermarking Three-Dimensional Polygonal Models. In: Proceedings of the ACM International Multimedia Conference and Exhibition, pp. 261–272 (1997).

[OMT02] Ohbuchi, R., Mukaiyama, A., Takahashi, S.: A Frequency-Domain Approach to Watermarking 3D Shapes. Computer Graphics Forum 21(3), 373–382 (2002).

[PC03] G. Peyré and L. Cohen. Geodesic re-meshing and parameterization using front propagation. In Proc. of VLSM'03, pages 33–40, 2003.

[PC05] G. Peyré and L. Cohen. Geodesic computations for fast and accurate surface remeshing and parameterization. Progress in nonlinear differential equations and their applications, 63:157–171, 2005.

[PCB05] Y. Pan, I. Cheng, and A. Basu. Quality metric for approximating subjective evaluation of 3-d objects. IEEE Trans. on Multimedia, 7(2):269–279, 2005.

[PGH99] Perez-Gonzalez, F. & Hernandez, J. R., 1999: A Tutorial in Digital Watermarking. Unpublished.

[PKA03] D.L. Page, A. Koschan, and M.A. Abidi. Perception-based 3d triangle mesh segmentation using fast marching watersheds. In Proc. Of the International Conference on Computer Vision and Pattern Recognition, volume 2, pages 27–32, June 2003.

[PKSRAA03] D.L. Page, A. Koschan, S. Sukumar, B. Roui-Abidi, and M. Abidi. Shape analysis algorithm based on information theory. In Proceedings of the International Conference on Image Processing, volume 1, pages 229–232, 2003.

[PSS01] Praun, E., Sweldens, W., Schröder, P.: Consistent mesh parameterizations. In: SIGGRAPH '01, pp. 179–184. ACM Press, New York (2001).

[PZ98] C. I. Podilchuk and W. Zeng. Image-adaptive watermarking using visual models, IEEE Journal on Selected Areas in Communications, vol. 16, pp. 525{539, Apr. 1998.

[R01] M. Reddy. Perceptually optimized 3d graphics. IEEE Computer Graphics and Applications, 21(5):68–75, 2001.

[RA06] P. Alface. Perception and Re-Synchronization Issues for the Watermarking of 3D Shapes. PhD thesis, Universite catholique de Louvain(UCL), Belgium, 2006.

[RAM05] Rondao Alface, P., Macq, B.: Feature-Based Watermarking of 3D Objects Towards Robustness against Remeshing and De-synchronization. In: Proceedings of the SPIEIS and T Electronic Imaging, vol. 5681, pp. 400–408 (2005).

[SAPH04] Schreiner, J., Asirvatham, A., Praun, E., Hoppe, H.: Inter-surface mapping. ACM Trans. Graph. 23(3), 870–877 (2004)

[SE09] David C. Schneider and Peter Eisert. "Fast Nonrigid Mesh Registration with a Data-Driven Deformation Prior". In ICCV Workshop on Non- Rigid Shape Analysis and Deformable Image Alignment, 2009.

[SHW06] Swaminathan, A., He, S., Wu, M.: Exploring QIM based anti-collusion fingerprinting for multimedia. In: Proc. SPIE Conf. Security, Watermarking, and Steganography. San Jose, CA (2006).

[SP04] Sumner, R., Popovic, J.: Deformation transfer for triangle meshes. ACM Trans. Graph. 23(3), 399–405 (2004).

[SPGKA06] S. R. Sukumar, D. Page, A. Gribok, A. Koschan, and M. A. Abidi. Shape measure for identifying perceptually informative parts of 3D objects. In Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT ’06), pages 679– 686, 2006.

[SSG03] O. Sifri, A. Sheffer, and C. Gotsman. Geodesic-based surface remeshing. In Proc. 12th Intnl. Meshing Roundtable, pages 189–199, 2003.

[SSKGH05] Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S. J., and Hoppe, H. 2005. Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.* 24, 3 (Jul. 2005), 553-560.

[T95] G. Taubin. A signal processing approach to fair surface design. In Proc. of SIGGRAPH’95, pages 351–358, 1995.

[TNM90] K. Tanaka, Y. Nakamura, and K. Matsui, Embedding secret information into a dithered multi-level image, in Proc. 1990 IEEE Military Communications Conference, pp. 216–220, 1990.

[U07] F. Uccheddu. Robust and imperceptible watermarking of 3D meshes. Ph.D. thesis, University of Florence, Italy.

[UCB04] Uccheddu, F., Corsini, M., Barni, M.: Wavelet-Based Blind Watermarking of 3D Models. In: Proceedings of the Multimedia and Security Workshop, pp. 143–154 (2004).

[WBSS04] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13(4), pp. 1–14, 2004.

[WC05] Wu, H.T., Chueng, Y.M.: A Fragile Watermarking Scheme for 3D Meshes. In: Proceedings of the Multimedia and Security Workshop, pp. 117–124 (2005).

[WK05] Wu, J., Kobbelt, L.: Efficient Spectral Watermarking of Large Meshes with Orthogonal Basis Functions. Visual Computer 21(8-10), 848–857 (2005).

[WLDB08] K. Want, G. Lavoué, F. Denis, A. Baskurt. "A comprehensive survey on three-dimensional mesh watermarking", IEEE Trans. On Multimedia, vol. 10, no. 8, pp. 1513-1527, 2008.

[WLDBH10] K. Wang, G. Lavoué, F. Denis, A. Baskurt, X. He, A benchmark for 3D mesh watermarking, in: Proceedings of the IEEE International Conferenceon Shape Modeling and Applications, 2010, (to appear).

[WM01] P.W. Wong and N. Memon. Secret and public key image watermarking schemes for images authentication and ownership verification. IEEE Trans. on Image Processing, 10(10):1593–1601, 2001.

[WP99] R. B. Wolfgang, C. I. Podilchuk, a d E. J. D. III, Perceptual watermarks for digital images and video, in Security and Watermarking of Multimedia Contents, P. W. Wong and E. J. D. III, Eds., vol. 3657, no. 1. San Jose, CA, USA: SPIE, 1999, pp. 40-51.

[XA01] L. Xie and G.R. Arce. A class of authentication digital watermarking for secure multimedia communication. IEEE Trans. on Image Processing, 10(11):1754–1764, 2001.

[YY99] Yeo, B., Yeung, M.M.: Watermarking 3D Objects for Verification. IEEE Computer Graphics and Applications 19(1), 36–45 (2005).

[ZTS02] Zuckerberger, E., Tal, A., Shlafman, S.: Polyhedral surface decomposition with applications. Comput. Graph. 26(5), 733–743 (2002).

[ZMT05] Zhang, E., Mischaikow, K., Turk, G.: Feature-based surface parameterization and texture mapping. ACM Trans. Graph. 24(1), 1–27 (2005).

[ZH04] Zhou, Y., Huang, Z.: Decomposing polygon meshes by means of critical points. In: MMM, pp.

187–195 (2004).

[ZTP04] Zafeiriou, S., Tefas, A., Pitas, I.: A Blind Robust Watermarking Scheme for Copyright Protection of 3D Mesh Models. In: Proceedings of the International Conference on Image Processing, vol. 3, pp. 1569-1572 (2004).