Robust 3D Watermarking

Xavier Rolland-Névière

UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS

ÉCOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE

pour l'obtention du grade de

Docteur en Sciences

de l'Université de Nice-Sophia Antipolis

Mention : Automatique, traitement du signal et des images

présentée et soutenue par

Xavier ROLLAND-NEVIÈRE

Robust 3D Watermarking
Tatouage 3D robuste

Thèse dirigée par Pierre ALLIEZ
soutenue le 12 Novembre 2014

Jury :

M. Patrick BAS ..... Chargé de recherche CNRS, LAGIS, École Centrale Lille (*rapporteur*)
M. Guillaume LAVOUÉ ..... Maître de conférences, INSA Lyon (*rapporteur*)
M. Marc ANTONINI ..... Directeur de recherche CNRS, laboratoire I3S, UNSA-CNRS (*examinateur*)
M. Mauro BARNI ..... Associate Professor, University of Siena (*examinateur*)
M. François CAYRE ..... Maître de conférences, Grenoble-INP (*examinateur*)
M. William PUECH ..... Professeur des universités, LIRMM, Université Montpellier II (*examinateur*)
M. Pierre ALLIEZ ..... Directeur de recherche, Inria Sophia Antipolis - Méditerranée (*directeur de thèse*)
M. Gwenaël DOËRR ..... Ingénieur de recherche, Technicolor R&D France (*encadrant industriel*)

Abstract

3D models are valuable assets widely used in the industry and likely to face piracy issues. This dissertation deals with robust mesh watermarking that is used for traitor-tracing.

Following a review of state-of-the-art 3D watermarking systems, the robustness of several content adaptation transforms are benchmarked. An embedding domain robust against pose is investigated, with a thickness estimation based on a robust distance function to a point cloud constructed from some mesh diameters. A benchmark showcases the performance of this domain that provides a basis for robust watermarking in 3D animations.

For static meshes, modulating the radial distances is an efficient approach to watermarking. It has been formulated as a quadratic programming problem minimizing the geometric distortion while embedding the payload in the radial distances. This formulation is leveraged to create a robust watermarking framework, with the integration of the spread-transform, integral reference primitives, arbitrarily selected relocation directions and alternate metrics to minimize the distortion perceived. Benchmarking results showcase the benefits of these add-ons w.r.t the fidelity vs. robustness watermarking trade-off. The watermark security is then investigated with two obfuscation mechanisms and a series of attacks that highlight the remaining limitations. A resynchronization approach is finally integrated to deal with cropping attacks. The resynchronization embeds landmarks in a configuration that conveys synchronization information that will be lost after cropping. During the decoding, this information is blindly retrieved and significant robustness improvements are achieved.

Résumé

Les modèles 3D sont des contenus précieux très utilisés dans l'industrie, et donc la cible potentielle de piratages. Le tatouage robuste pour les maillages 3D apporte une réponse au problème du traçage de traître. Dans l'état de l'art du domaine, la couche d'adaptation du contenu en particulier est testée face à des attaques standards. Une approche robuste à la pose est alors étudiée. Elle utilise une estimation robuste de l'épaisseur, définie comme la distance à un nuage de points construits à partir de mesures du diamètre. Les performances expérimentales montrent qu'elle forme un point de départ prometteur pour le tatouage robuste de maillages 3D posés.

Pour les maillages statiques, la modulation des distances radiales est une approche efficace du tatouage. Elle a été formulée comme un problème d'optimisation quadratique sous contrainte, dont nous proposons plusieurs extensions : une transformée par étalement, des primitives de référence calculées de manière intégrale, des directions de déplacement arbitraires, et de nouvelles métriques pour minimiser la distorsion perçue par un utilisateur. Des expériences illustrent leurs bénéfices pour le compromis entre la robustesse et la fidélité du tatouage. La sécurité est analysée par l'intermédiaire de deux mécanismes de protection et par une série d'attaques et de contre-mesures. Un système de resynchronisation est intégré afin d'améliorer la résistance au rognage. Des points de recalage sont insérés dans une configuration spécifique qui porte les informations habituellement éliminées par l'attaque. Au décodage, elles sont récupérées de manière aveugle. Un gain significatif des performances est mesuré expérimentalement.

Acknowledgments

I would like to thank my advisors, Mr. Pierre Alliez and Mr. Gwenaël Doërr for their tremendous help in my research, and for being always available to discuss specific issues, even the minor ones. It has been three great years (three and a half counting my internship) at Technicolor, thanks to all my colleagues in the Security & Content Protection Labs. They have taught me a lot, even though they are working in very different fields of research, and I enjoyed our many – not always work related – talks.

I also thank my family, especially my parents, for their support throughout the years. While they may not understand everything I have been recently working on, they have always proposed to help. I am grateful for their advice and for the time they took proofreading parts of this dissertation, making sure my French summary is understandable, or at least without too many spelling or grammatical mistakes!

Thanks to all my friends and good luck to all of those that are still not done with their Ph.D. I am confident they will be as successful and happy as the others who just finished.

Contents

List of Tables

List of Figures

Notations

Vectors are denoted in bold $\mathbf{x}$, matrices in capital bold $\mathbf{X}$. Elements of a matrix $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ and a vector $\mathbf{x} \in \mathbb{R}^{n_1}$ are denoted $X_{i,j}$, $(i,j) \in [\![1, n_1]\!] \times [\![1, n_2]\!]$ and x_i, $i \in [\![1, n_1]\!]$.

In $\mathbb{R}^3$, $\mathbf{p}$ indifferently denotes the point or the vector. $\|.\|$ is the Euclidean norm. $|\mathcal{X}|$ (respectively $|\mathbf{x}|$) is the number of element in the set $\mathcal{X}$, (resp. the vector $\mathbf{x}$). A 'w' superscript indicates a watermarked variable.

Main Notations

Notations used throughout the dissertation.

Notation	Description
A	Total area of the surface mesh
$\mathcal{B}(\mathbf{g}, r)$	Ball centered at $\mathbf{g}$ and of radius r
$\mathbf{c}$	Watermark carrier
$\delta_{a,b}$	Kronecker delta between a and b
$\mathcal{E}, \mathcal{F}$	Set of mesh edges and facets
$\mathbf{I}_n$	Identity matrix of size n
$\mathbf{J}^{\mathbf{a}}_{\mathbf{b}}(\mathbf{a}_0)$	Jacobian matrix such that its entry $J_{(i,j)}(\mathbf{a}_0)$ is the first derivative at $\mathbf{a}_0$ of the ith component of $\mathbf{a}$ with regard to the jth variable in $\mathbf{b}$
$\mathbf{L}$	Laplacian matrix derived from a mesh
λ	Control parameter for trade-offs
$\mathcal{M}$	Mesh, and unless mentioned otherwise, a triangle surface mesh
$\mathbf{m}, \mathbf{M}$	Watermark payload with antipodal bits, in vector form and as a diagonal matrix
$\mathbf{n}_i, \mathbf{n}_f, \mathbf{n}_\mathbf{q}$	Normal vector to the mesh surface at the ith vertex location, on the facet f or at point $\mathbf{q}$
n_v, n_f	Number of vertices and facets in a mesh
n_b	Number of payload bits
$\mathcal{N}_1(v)$, $\mathcal{N}_1^{\mathcal{F}}(v)$	Sets of, respectively, vertices and facets, in the 1-ring neighborhood of vertex v
$\mathbf{p}$, $\mathbf{P}$	Location of a vertex and matrix of all vertex locations in a mesh
$\mathbf{q}$, $\mathbf{Q}$	Query point on the mesh surface, and set/matrix of all query points
$\mathcal{S}(\mathbf{g}, r)$	Sphere centered at $\mathbf{g}$ and of radius r
$(\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z)$	Basis vectors of the Cartesian coordinates system
v_i, $\mathcal{V}$	ith mesh vertex and set of mesh vertices
V	Volume of the 3D object bounded by the mesh

Thickness Estimation

Notations for Chapter 5.

Notation	Description
$a^q(f), a^q$	Local per-facet accuracy of the estimator, and global per-mesh accuracy, over q runs of the algorithm
b	Length of the space diagonal of the bounding box
δ_{SDF}	Thickness estimation resulting from the SDF procedure
η	Aperture closing per iteration
$g(f)$	Ground-truth estimation of the thickness at a facet f
I	Number of iteration for the diameter estimation
$I^q(f)$, I^q	Local per-facet instability of the estimator and global per-mesh instability, over q runs of the algorithm
k	Scale of the robust diameter estimate
n_s, $\bar{n_s}$	Number of samples in a mesh and normalized number of samples with regard to the mesh bounding box
ϕ	Opening angle of the cone in the diameter estimation
R	Number of rays cast in a diameter estimation cone
$R^q_{\mathcal{F},\mathcal{F}'}(f)$, $R^q_{\mathcal{F},\mathcal{F}'}$	Local per-facet error of the estimator and global per-mesh error, over q runs of the algorithm
τ	Threshold stopping the iterative cone closing
$t(\mathbf{q})$, $t_k(\mathbf{q})$	Thickness estimation at query point $\mathbf{q}$ (at a scale k when indicated)
w	Spatial window of the bilateral smoothing

Watermarking Radial Distances

Notations for Chapters 6 to 9.

Notation	Description
α, $\boldsymbol{\alpha}$	Watermark embedding strength (vector of α)
β	Watermark separation offset between bins of the histogram of radial distances
B_i	Bin in the histogram of radial distances associated to the ith radial distance (vertex)
$\delta\bar{\rho}_i^{\text{w}}$	ith unknown in the QP framework corresponding to the normalized relocation of the associated vertex in the radial direction
δr_i^{w}	ith unknown corresponding to the relocation of the associated vertex in the arbitrary direction $\mathbf{u}_i$
Δ	Histogram step
ϵ	Secret parameter in the watermarking system: relative offsets to obfuscate both ends of a histogram $(\epsilon_{\min}, \epsilon_{\max})$ or dither in QIM.
η	Secret seed to generate the secret parameters of a watermark system
$\mathbf{g}$	Position of the mesh center of mass
$\mathcal{G}$	Set of histogram bins associated to a payload bit
N_j	Number of samples in the jth bin of a histogram
$\mathcal{L}$	Set of landmark points
m, M	Minimum and maximum of the radial distances
n_B	Number of bins in the histogram of radial distances
$\mathbf{u}_i$	Relocation direction associated to the ith vertex
ϕ_i	Third spherical coordinate of the ith vertex, with regard to the mesh center of mass
$\boldsymbol{\Phi}$	Matrix for the spread-transform projection
$\boldsymbol{\Psi}$	Diagonal matrix of cosines between the radial unit vectors and the arbitrary relocation directions (scaled by the histogram step Δ)
ρ_i, $\bar{\rho}_i$, $\boldsymbol{\rho}_i$	Radial distance from the ith vertex position to the center of mass, normalized radial distance in $[0, 1)$, and unit radial vector from the center of mass to the vertex
$\boldsymbol{\rho}$	Vector of radial distances ρ_i from all the vertex positions to the mesh center of mass

$\mathbf{s}$, $\mathbf{s}(\eta)$	Spreading sequence, pseudo-randomly generated using the secret seed η
$\bar{\mathbf{t}}$	Vector of normalized watermark targets
θ_i	Second Spherical coordinate of the ith vertex, with regard to the mesh center of mass
ω	Cost function to represent the watermark fidelity
w	Nonnegative weights
$\mathbf{W}$	Matrix mapping the radial distances to the associated histogram bins

Chapter 1

Introduction

1.1 Context

Three-dimensional (3D) models have become ubiquitous in many industrial applications. In movie production, they have been replacing traditional two-dimensional (2D) graphics since the early eighties and the release of *Tron* (1982) by Walt Disney Pictures. Thanks to ever more powerful animation software products [Aut14] and motion capture systems, animations of 3D models are now routinely used not only in animated movies but also in live-action feature films and series as well. The quality and the level of details of 3D models make them more and more indistinguishable from real-life objects. We may for instance be unaware that in the large-scale battles of the *Lord of the Rings* trilogy, background combatants are 3D models animated through a complex artificial intelligence system [Reg14].

The rapid dissemination of 3D graphics processing units (GPU) in the mass market around the year 2000 has prompted the video game industry to drift away from 2D, and from pseudo-3D games (simulating 3D using projections of 2D graphics, also known as 2.5D), to fully capable 3D game engines, leading to an abundant use of 3D models. While these 3D assets may be generated by professional game development studios for internal use, new business models for 3D graphics have appeared when companies started producing and brokering this type of content [FC14]. In computational science and engineering, Computer-aided Design (CAD) routinely uses 3D solid modeling for numerical simulations, as it provides the means to cut down research and development costs.

In addition to their professional applications, 3D models are also playing a growing part in user-generated content. For instance, some modern game engines offer the possibility for custom content to be integrated. Creating 3D models for new characters or other game assets has thus grown into a popular activity, supported by dedicated tools [Ble14, Aut14, Epi14] and publications targeting the semi-professional market [Pub13].

The expected booming of 3D printing activities will further expand the importance of 3D models. The accessibility of 3D printers for everyday users is likely to impact consumption scenarios. Contrary to other types of multimedia content, 3D models will turn from cultural and artistic digital products into actual tangible consumer goods. Analysts forecast new trends based on downloading and printing models using on-line databases [WGL+13]. Because 3D models are versatile and will be increasingly available in the mass market, protecting their dissemination and their intellectual property has become a concern.

Aside from patents, trademarks or industrial design rights infringements that always appear with digital creations, copyright infringement is a crucial topic for the entertainment industries [Org08].

Films and music are notoriously pirated and at the center of a large-scale digital black market [Ahr06]. Copyright holders have then put a great deal of efforts into tracking the origins of illegal redistributions of their properties. As the complexity and the value of 3D assets increase, similar scenarios are expected to occur with the illicit transmission of copyrighted 3D models. However, because of the ever thinner frontier border between digital models and tangible goods in the real world, this issue is greatly amplified by its potential new impact on merchandising or licensing.

Consider a 3D model of the main character of the latest blockbuster. If this model can be illegally downloaded, anybody will be able to manufacture at home some custom goods that bear this character. This is going to affect the sales of, e.g., toys based on this character; if the illegal 3D model is a degraded version of the original, it may also impact the reputation of the copyright holder. For children's toys, a plurality of norms may also be violated by a counterfeit home-made product, leading to safety problems. Related issues have begun to emerge. In 2012, Games Workshop Limited issued a cease-and-desist notifications for a CAD model and a 3D print based on one of its miniature tank [New12]. In 2013, HBO sent a cease-and-desist against a 3D model for an iPod docking in the shape of the iron throne from the TV show *Game of Thrones* [Mac14]. In both cases, companies claimed copyright infringement took place. In this context, companies are facing the challenge of identifying if a model is an illegal reproduction of their work or where a counterfeit model originates from.

1.2 Digital Watermarking

Digital watermarking is a technical field that provides copyright owners with the means to protect their intellectual property rights. It is a central component of multimedia content protection architectures that complements traditional cryptography [CMB+07]. While cryptography aims at preventing an unauthorized user from accessing a content, watermarking addresses the issues that arise once an authorized user has been granted access, e.g., after the decryption, or if the encryption is broken.

In general, watermarking consists in modifying multimedia content in a robust and imperceptible way in order to hide a secret message. The embedded message, referred to as the watermark payload, can indeed serve as a forensic piece of evidence for 'traitor tracing' tasks. Alternatively, it may constitute a proof of ownership in case of litigations. In the former case, authorized users are only given access to a custom copy of valuable 3D assets. Each user would possess an imperceptibly different and unique version of the 3D models; users and copies of the original 3D assets thus being one-to-one mapped. If an illegal dissemination of the asset occurs, copyright owners can find the leak, as his or her identity is embedded in the pirated publicly-available content. In contrast, in proof of ownership use-cases, the watermark payload corresponds to the identity of the copyright owner. He or she can then successfully prove that a content is his or her own.

Digital watermarking has many other uses for security purposes, e.g., for content tampering detection, and it also has applications outside this scope, for instance in broadcast monitoring. Because of this plurality of applications, digital watermarking systems are usually adapted to meet the specific requirements of their intended use-cases.

1.3 Problem Statement

From a technical standpoint, all watermarking systems are akin to digital communication systems, where an emitter sends a signal to a receiver through a communication channel. In watermarking,

the embedder emits a signal, usually encoding a payload, which is carried by the copyrighted content, and then retrieved by the decoder. Both ends of this system are managed by the copyright holders, but the operations applied to the copyrighted content in-between are not controlled: an authorized user is able to arbitrarily modify his or her content. These modifications need to be handled by the watermarking systems, so that the decoder can still retrieve the payload. Watermarking is then characterized by its robustness.

Because increasing the size of the payload usually decreases the robustness, a balance between these two quantities needs to be set. In addition, most people will not accept that the watermarking impacts their everyday use of copyrighted content. The fidelity of the watermark, measuring the amount of change in the watermarked content, further constrains the system. In general, watermarking then faces a complex balance between robustness, fidelity and embedding-rate.

This dissertation focuses on watermarking for 3D models, abbreviated to '3D watermarking', in the context of traitor-tracing. The payload, representing the identity (ID) of a user, needs to be embedded in the model in a very robust and secure manner. Indeed, once it becomes public knowledge that watermarking techniques are being employed, people leaking 3D contents are likely to try and remove the incriminating messages so as to avoid prosecution. The embedding rate of the system then reaches at most a few dozens of bits, and the aforementioned routine watermarking trade-off focuses on the robustness. These systems are simply referred to as 'robust watermarking'.

In contrast, 'fragile' or 'high-capacity' 3D watermarking systems focus on increasing the embedding rate ('high-capacity') or on applications where the constraints on the robustness can be partially lifted, such as content tampering detection ('fragile'). A plurality of fragile or high-capacity 3D watermarking systems have been proposed instead of robust ones, because providing a high level of robustness in the 3D context yields several scientific and technical challenges.

1.4 Technical Challenges in Robust 3D Watermarking

Complex issues immediately arise from the very ways 3D models are digitally represented. The robustness of a watermarking system relies in part upon an agreement between the embedder and the decoder on the way they represent the 3D model. When the decoder does not have access to the original non-watermarked 3D object ('blind watermarking'), achieving such an agreement is actually tough. Nonetheless, providing the decoder with the original object ('non-blind watermarking') incurs several practical drawbacks. Issues regarding the representation of 3D models also impact the usefulness of the most successful signal processing tools for robust watermarking, such as the Fourier Transform or the Wavelet Transform. The extensions of these transforms for 3D models are indeed defined in a content-dependent manner. This makes it harder to handle any modification of the 3D model between embedding and decoding.

Watermarked 3D models can undergo a variety of possible modifications. Two types of modifications are very challenging to deal with: the cropping attack and the isometric deformation of the surface, a.k.a., the pose. Both types yield a synchronization problem for watermarking. With cropping, part of the model is deleted, and its value is usually reduced from the point of view of copyright holders. However, even mildly noticeable amounts of cropping may lead to large synchronization problems in 3D watermarking. Cropping thus remains a constant issue. Regarding the pose operations, they only occur when animating 3D models. Not all 3D watermarking systems are thus expected to be robust against pose, but it becomes a major concern in contexts where the watermarked 3D assets are parts of animations.

In traitor tracing, the ability for an unauthorized user to access and modify the watermark payload as he or she wishes may have serious legal consequences: the incriminating ID could

indeed be changed so as to point to another user. No one other than the copyright owner should thus be able to access the embedded ID. This constraint is referred to as a security issue. 3D watermarking research has often either overlooked this issue or used unsound validation methods. On the opposite, thorough theoretical analyses have been undertaken to assess the watermark security in other types of content.

As copyright holders may use their 3D assets to advertise their work, they expect robust watermarking systems to also preserve the visual appearance of the 3D models, i.e. to always achieve a given level of fidelity. There is however no definite way for measuring the perceptual impact of an embedding in 3D graphics. Perceptually-correlated distortion metrics are still being investigated. The few existing solutions all present some shortcomings. Their adoption by the watermarking community is limited, which has hindered research, as different watermarking systems are not aligned with regard to the same distortion metrics.

At last, many 3D automated operations (algorithms, procedures) have some requirements on their 3D inputs. These requirements are often not met in real-life, and 3D objects often need to be repaired before being processed, for instance by removing some defects. Most databases that were not created for research purposes thus cannot be straightforwardly used to perform large benchmarking campaigns. Unlike in audio and images, only small scale 3D watermarking benchmarks are feasible.

1.5 Outline

Chapter 2 provides a more technical introduction to the 3D watermarking domain and some key background notions on 3D objects processing and watermarking. The remaining chapters are then grouped in two parts.

Part one of this dissertation focuses on content adaptation transforms for 3D watermarking. The main state-of-the-art robust watermarking systems, classified according to their adaptation transforms, are reviewed in Chapter 3. A benchmark of some of the most common adaptation transforms is reported in Chapter 4. Finally, Chapter 5 investigates a novel extraction transform, based on the thickness of 3D objects, which exhibits promising properties against pose operations. Its performance is thoroughly tested.

Part two of this dissertation focuses on enhancing and extending a constraint optimization formulation for 3D watermarking, to create a modular and versatile framework for robust watermarking systems. Chapter 6 details several extensions to improve the robustness and the fidelity of the original watermarking formulation. These extensions are then experimentally benchmarked in Chapter 7. Chapter 8 takes a closer look at the security of the watermarking framework by describing a series of attacks and counter-measures. Finally, the specific issue of cropping is addressed in Chapter 9, with a novel resynchronization approach that is added to the framework.

Chapter 10 summarizes the main results presented in this dissertation. The original contributions of this work to the 3D watermarking field are emphasized, and stimulating directions for future research are listed.

1.6 List of publications

Our contribution has led to the following publications.

International Publications

- *Anti-Cropping Blind Resynchronization for 3D Watermarking.* Xavier ROLLAND-NEVIÈRE, Gwenaël DOËRR, Pierre ALLIEZ. Submission to ICASSP, 2015.
- *Security Analysis of Radial-based 3D Watermarking Systems.* Xavier ROLLAND-NEVIÈRE, Gwenaël DOËRR, Pierre ALLIEZ. Proceedings of the IEEE Workshop on Information Forensics and Security, 2014 [to appear].
- *Spread-Transform and Roughness-based Shaping to improve 3D Watermarking based on Quadratic Programming.* Xavier ROLLAND-NEVIÈRE, Gwenaël DOËRR, Pierre ALLIEZ. Proceedings of the IEEE International Conference on Image Processing, 2014 [to appear].
- *Triangle Surface Mesh Watermarking based on a Constrained Optimization Framework.* Xavier ROLLAND-NEVIÈRE, Gwenaël DOËRR, Pierre ALLIEZ. IEEE Transactions on Information Forensics and Security, vol. 9, no 9, September 2014, pp. 1491-1501.
- *Robust Diameter-based thickness estimation for 3D objects.* Xavier ROLLAND-NEVIÈRE, Gwenaël DOËRR, Pierre ALLIEZ. Graphical Models, vol. 75, no 6, November 2013, pp. 279-296.

Patents Applications

- *Generalized Quadratic Programming Framework for 3D Watermarking.* Xavier ROLLAND-NEVIÈRE, Gwenaël DOËRR, Pierre ALLIEZ.
- *Thickness-based 3D Watermarking.* Xavier ROLLAND-NEVIÈRE, Gwenaël DOËRR, Pierre ALLIEZ.
- *Method for introducing watermark feature points on a surface mesh.* Xavier ROLLAND-NEVIÈRE, Gwenaël DOËRR, Pierre ALLIEZ.
- *Method for using landmark points as a resynchronization mechanism for 3D watermarking.* Xavier ROLLAND-NEVIÈRE, Gwenaël DOËRR, Pierre ALLIEZ.

Chapter 2

Background Notions for 3D Watermarking

3D watermarking is a subfield of research whose foundations are built on both the digital watermarking and the geometry processing domains. Some necessary background information from these areas is reviewed in the first two sections of this chapter. In Section 2.3, the recent findings on the assessment of 3D mesh distortion are summarized. Research carried out in this domain is indeed especially relevant to 3D watermarking; the state-of-the-art review presented in Chapter 3 heavily relies on all the notions introduced next.

2.1 Triangle Mesh Processing

This section is dedicated to introducing some basic concepts relating to 3D objects. A few advanced topics for mesh processing, routinely used in the context of 3D watermarking, are eventually reviewed.

2.1.1 Triangle Mesh Definition

Representation of 3D Objects

Creating and processing the geometry of three-dimensional (3D) data is one of the main subfield of research in computer graphics. In this context, 3D data are 3D objects that can be represented in many ways, using voxels, point-clouds, splines, volumetric or polygonal meshes... Some of these representations focus on the description of the *surface boundary* of a 3D object, formally defined as "an orientable continuous 2D manifold embedded in $\mathbb{R}^3$" [BKP+10]. The *parametric* representation of a surface is a mapping $\mathbf{f}$ from $\Omega \subset \mathbb{R}^2$ to $\mathbf{f}(\Omega) \subset \mathbb{R}^3$.

The definition of a surface only allows for 3D objects to be non-degenerate 3D solids, i.e. both *watertight* and nowhere infinitely thin objects. Still, practical computer graphics applications are usually able to handle *surfaces with boundaries*. These correspond to surfaces with holes that can be filled, so as to turn them into proper orientable continuous 2D manifold.

Surface Mesh for 3D objects

Since surfaces are continuous, their digital representations are only discrete approximations, a.k.a. *samplings*. One of the most popular digital representation is a piecewise linear approximation in the form of a *polygon surface mesh*. Formally, a polygon surface mesh $\mathcal{M}$ is defined by its

geometry and connectivity. The latter is a graph structure. Its set of vertices and edges are respectively $\mathcal{V} = \{v_i,\, i \in [\![1, |\mathcal{V}|]\!]\}$, and $\mathcal{E} = \{e_{(i,j)},\, (i,j) \in [\![1, |\mathcal{V}|]\!]^2\}$. The geometry, also referred to as the 'embedding' of the 2D surface in $\mathbb{R}^3$, is defined by mapping a vertex v_i to a point $\mathbf{p}_i \in \mathbb{R}^3$. $\mathbf{P} \in \mathbb{R}^{3\times n_v}$, referred to as the 'vertex positions', denotes the matrix representing the concatenation of all vertex positions. It corresponds to approximation of the underlying surface boundary of the 3D object, and constitutes an *irregularly sampled* signal. Unless mentioned otherwise, n_v always denotes the number of vertices in mesh.

2-manifold surfaces are represented with 2-manifold polygonal surface meshes, which are characterized by the fact that: (i) they do not present any *self-intersection*, and (ii) all their edges are exactly shared by two faces of their graph (or at least one face for a 2-manifold with boundaries). An alternate characterization of a 2-manifold surface mesh is that the local neighborhood of all vertices is homeomorphic to a disk (or half a disk for a 2-manifold with boundaries). Data-structures have been developed for these meshes to minimize storage and optimize neighborhood searches and traversals of the mesh [FGK+98]. In this context, the set of polygon faces $\mathcal{F} = \{f_i,\, i \in [\![1, |\mathcal{F}|]\!]\}$ is often used instead of $\mathcal{E}$ to describe the connectivity information. n_f henceforth denotes $|\mathcal{F}|$.

Triangle Surface Mesh

A sub-case of polygonal surface meshes are triangle surface meshes, where all polygons are triangles. 3D watermarking mainly focuses on triangle surface meshes, which will subsequently be simply referred to as 'meshes'. The motivation for choosing triangles as primitives is that: (i) polygons can always be partitioned into triangles, and (ii) vertices in arbitrary polygon facets may neither be coplanar nor convex in $\mathbb{R}^3$. Note that most of the aforementioned efficient data-structures also handle some degenerate meshes, such as ones with non-manifold vertices incident to two distinct triangle fans (sets of connected triangles sharing one central vertex).

Neighborhood and Regularity

The *one-ring neighborhood* of a vertex v_i, also referred to as its star neighborhood, is the set $\mathcal{N}_1(v_i)$, formed by the vertices which are linked to v_i by a mesh edge, i.e. : $\{v_j \in \mathcal{V} \mid e_{(i,j)} \in \mathcal{E}\}$. The n-ring neighborhood of a vertex is then recursively defined from the 1 ring. This neighborhood definition is commonly used for its simplicity as a connectivity-based only quantity. A neighborhood search then reduces to a graph search in $\mathcal{E}$, and does not involve any computation on the geometric information in $\mathcal{P}$, which are only sampling approximations.

$|\mathcal{N}_1(v_i)|$ is the *valence* of v_i. Triangle meshes are labeled as *regular* when the valence of all the vertices is exactly six. When a mesh is only piecewise regular, in other words, almost everywhere regular, it is labeled as *semi-regular*. Otherwise, meshes are labeled as *irregular*.

The triangle facets in the one ring neighborhood of a vertex v_i, denoted by $\mathcal{N}_1^{\mathcal{F}}(v_i)$, are the facets of $\mathcal{F}$ in which v_i is a vertex.

Smooth Surface Mesh Representation

Smooth surfaces are characterized by: (i) their parametric representation maps $\mathbf{f}$ are C^k continuous ($k \geq 2$), and (ii) the partial derivatives of $\mathbf{f}$ do not vanish. Although the mesh geometry maps vertices to discrete points, a mesh surface is still continuous. But since it is only piecewise linearly continuous, most of the quantities that are defined on a *smooth* surface boundary of a 3D object cannot be straightforwardly extended to meshes. A first challenge in mesh processing is to approximate these quantities.

Moreover, in computer graphics, the surfaces represented via meshes are expected to be almost everywhere smooth, except in a finite number of locations called 'sharp features'. These are often found in mechanical objects, e.g. the *fandisk* mesh. Dealing with sharp features is another major challenge in geometry processing and in 3D watermarking.

Other Types of Information

Much additional information can be added to a mesh such as colors and normal directions for vertices, labels to create groups of faces, or texture maps, etc. This information enriches the visual appearance when rendering the mesh on screen. Because this information may be straightforwardly removed from a mesh file (for example, in an Object File Format (OFF), colors are stored in optional dedicated columns), robust 3D watermarking research generally does not take them into account.

All the meshes considered hereafter are solely defined through their vertex positions $\mathbf{P}$, and their vertices $\mathcal{V}$ linked to form the triangle facets $\mathcal{F}$. Table D.1 lists the experimental database of meshes that is mainly used in the following, as well as some of their specificities, such as defects, complexity or type.

To conclude this series of definition, meshes and surfaces described above are sometimes referred to as 'static', as opposed to the 'dynamic' ones that are used in 3D animations. In this dissertation, we only deal with statically defined objects.

2.1.2 Mesh Processing

Intrinsic vs. Extrinsic Quantities

The first fundamental form at a point $\mathbf{p}$ on a surface is defined as the dot product of two tangent vectors. It is canonically written as a two-by-two symmetric matrix whose coefficients are derived from the parameterization of the surface. It fully characterizes the metric properties of the surface, such as the area of a surface patch, or the geodesic distance between two points.

The first fundamental form is essential to distinguish between *intrinsic* and *extrinsic* quantities measured on a smooth surface. *Intrinsic* quantities are measured inside the surface. Intuitively, they could be computed by entities evolving on the surface, much like humans on the Earth. More formally, they are expressed solely in terms of the coefficient of the first fundamental form and do not depend on the embedding of the surface in $\mathbb{R}^3$. This is the case for the aforementioned surface area, or geodesic distances. In contrast, extrinsic quantities, such as the Euclidean distance, depend on the actual embedding.

Using intrinsic or extrinsic quantities directly impacts the properties of a watermarking system, such as its robustness (see Section 2.2.2) or its complexity.

Mesh Curvatures

One of the key notion in differential geometry is the *curvature* of a smooth surface. In $\mathbb{R}^2$, the curvature of a smooth curve intuitively measures how it deviates from a straight line, and is formally defined with the derivative of the tangent vector to the curve. For surfaces, given a tangent vector $\mathbf{t} \in \mathbb{R}^3$ to the surface at $\mathbf{p}$, the curvature $\kappa(\mathbf{p}, \mathbf{t})$ is the curvature of the curve defined by the intersection between the surface and the plane spanned by $(\mathbf{p}, \mathbf{n}, \mathbf{t})$, where $\mathbf{n} \in \mathbb{R}^3$ is the normal to the surface at $\mathbf{p}$ (see Figure 2-1.

The *principal curvatures* $(\kappa_{\min}(\mathbf{p}), \kappa_{\max}(\mathbf{p}))$ are the minimum and maximum values over the tangent directions of κ at $\mathbf{p}$; the *principal directions* are the tangent vectors associated to the principal curvatures. These specific curvatures, as well as the *mean curvature* $\kappa_{\text{mean}} = \frac{1}{2}(\kappa_{\min} + \kappa_{\max})$

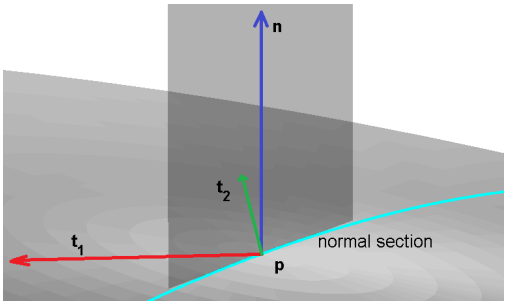


Figure 2-1: Local frame at $\mathbf{p}$ with normal section $(\mathbf{p}, \mathbf{n}, \mathbf{t})$, and the tangent plane spanned by $(\mathbf{t}_1, \mathbf{t}_2)$.

and the *Gaussian curvature*, $\kappa_G = \kappa_{\min}\kappa_{\max}$ are used to characterize and classify the local shape of smooth surfaces. While the mean curvature is extrinsic, the Gaussian curvature is intrinsic (Theorema Egregium).

Extending the computation of surface curvatures to meshes, which are only piecewise linear, has been an active field of research. Curvatures are extensively used in geometry processing. For instance, in remeshing applications[1], curvatures play a key part in defining non-uniform and locally adapted efficient sampling density [ACSD+03]. In 3D watermarking, the estimation of the principal curvatures has multiple applications, such as distortion metric (see Section 2.3) and synchronization mechanism definitions [AM05].

Two main strategies to compute the principal curvatures and the principal directions at a query point have been investigated. A first solution is to fit an analytic surface to a local patch around the point. For analytic surfaces with polynomial expressions, the curvatures can be written in closed-form depending on the polynomial parameters. This leads to an efficient curvature approximation, depending on the quality of the fitting [CP03]. A second category of approaches is built on the theory of normal cycles and its extension to polyhedral surfaces [CSM03]. In essence, the curvatures are computed as a weighted average of the signed dihedral angle of edges around $\mathbf{p}$. In Chapter 4, we investigate the use of both types of curvature estimators for robust watermarking purposes.

Rigid Mesh Alignment

Several mechanisms to canonically normalize a mesh have been proposed in the mesh processing literature. In 3D watermarking, a Principal Component Analysis (PCA)-based normalization mechanism [ZC01] and an Iterative Closest Point (ICP) [BM92]-based normalization mechanism have been widely adopted.

The ICP is inherently restricted to the case where the mesh to be normalized is associated with another mesh (for instance, in a non-blind watermarking system, as defined in Section 2.2.1). Its basic steps consists of (i) matching a series of points between an original and a query mesh, (ii) minimizing a Mean Square Error (MSE) cost function that models the mesh alteration with a rotation/translation, (iii) apply the estimated transform to the query mesh, and (iv) iterate the previous steps until some minimum cost threshold is reached.

The major advantage of PCA-based normalization over the ICP is that there is no need for a reference mesh (it can thus be used in blind watermarking, and the normalization is applied at both ends of the watermarking system, see Section 2.2.1). Its main steps are: (i) translating the mesh so that its center of mass is at the origin; (ii) scaling the mesh uniformly so that it is bounded within

[1] Remeshing consists in computing a second mesh from an initial mesh to achieve some quality requirement.

e.g. a unit-sphere[2]; (iii) performing a PCA of the mesh and aligning the principal directions with the coordinate axes; and (iv) selecting from the possible remaining mesh configurations the one in which some high order terms in the vertex positions are positive. For symmetric configurations, e.g. a spherical mesh, these last two steps may however be ill-defined, as the principal directions are ambiguous. The covariance matrix in the PCA is computed by summing over the mesh second degree terms, such as x_i^2, $x_i y_i \ldots$ The high order terms are the summation over the mesh of some third degree expressions in the vertex coordinates, such as x_i^3, y_i^3.

In watermarking, both the ICP and the PCA provide robustness against rigid transforms (and scaling).

Spectral Analysis

Frequency analysis (also called Fourier analysis) is a powerful and versatile family of tools for audio, image and video processing. Signal processing algorithms heavily rely on the Discrete Fourier Transform (DFT) or the Discrete Cosine Transform (DCT), and their efficient implementations with, e.g., the Fast Fourier Transform (FFT), to compute a spectral representation of a media. Their applications range from denoising to fingerprinting, watermarking, or compression. A large body of research has thus focused on extending these spectral analysis tools to meshes [ZVKD10].

In 1D, the fundamental transform to compute the spectrum of a digital signal in $\mathbb{R}^n$ is the DFT. It consists in projecting the signal onto a series of orthogonal and discretized basis functions, a.k.a. harmonics, which spans the spectral domain. These harmonics are the *eigenvectors of the 1D discrete Laplace Operator*, and correspond to a series of pairs of cosines and sines, sampled at the signal frequency. The frequencies of each harmonic pair is the square root of the associated eigenvalue of the discrete operator[3]. The projection results in two *series of scalar coefficients*, defined as the DFT of the signal, a.k.a. its spectral representation. To compute the spectral representation of a mesh, one needs to extend the definition of the discrete Laplace Operator to surface meshes embedded in $\mathbb{R}^3$.

This extension involves however a complex approximation problem, that was shown to be a 'no-free lunch' one [WMKG07]. It is theoretically impossible to define a Laplace Operator for meshes that match all essential properties of the continuous Laplace Operator in lower dimensions. Some of these properties are however instrumental for spectral analysis. Researchers have thus proposed a variety of Laplacian discretizations for meshes, which all present different benefits and limitations, depending on which ones of the antagonistic properties are preserved.

Notions	1D discrete signals	Mesh
Laplace Operator	Discrete 1D Laplace Operator	Laplacian Matrix $\mathbf{L}$
Harmonics	Discrete cosines/sines	Eigenvectors of $\mathbf{L}$ (bases)
Frequencies	Cosine/sines frequencies	(Square root) Eigenvalues of $\mathbf{L}$
Spectral Coefficients	Cosine/Sine amplitudes	3D projections of $\mathbf{P}$ on the bases

Table 2.1: Equivalences between the routine 1D spectral concepts and the spectral analysis for meshes.

Table 2.1 lists the analogous concepts between the 1D discrete spectral analysis and the mesh spectral analysis. Assuming that the Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n_v \times n_v}$ is a real symmetric positive

[2]This is therefore not a rigid mesh alignment operation.

[3]The operator is positive semi-definite and the multiplicity of all but the null eigenvalue (the DC component) is 2.

semi-definite, its eigenvectors $\mathbf{h}^k \in \mathbb{R}^{n_v}$ ($k \in [\![1, n_v]\!]$) are orthogonal and form the discretized basis functions for the spectral domain.

Their associated eigenvalues λ_k^2 are the squared frequencies of the mesh spectrum. The projection of the discrete geometry signal onto the kth eigenvector yields a spectral coefficient (triplet) associated to the frequency λ_k, denoted by X_k, Y_k, Z_k)[4]:

$$[X_k,\ Y_k,\ Z_k]^T = \mathbf{P}\mathbf{h}^k. \tag{2.1}$$

The amplitude of the mesh spectrum at the λ_k frequency is the magnitude of the spectral coefficient. Finally, reconstructing a mesh from its spectral representation amounts to projecting back the spectral coefficients, with:

$$\mathbf{P}^T = \sum_{k=1}^{n_v} \mathbf{h}^k [X_k,\ Y_k,\ Z_k] \tag{2.2}$$

For watermarking purposes, two discretizations have been mainly used: the combinatorial Laplacian and the Manifold Harmonics.

Combinatorial Laplacian The *combinatorial Laplacian* is only based on the mesh connectivity and has been introduced for compression purposes [KG00a]. It is equivalent to a uniform discretization of the continuous Laplacian for 2D surfaces. Because the connectivity represents a graph, some of the properties of this discretization have been studied in Spectral Graph Theory.

Formally, the Laplacian $\mathbf{L}$ is defined with:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{2.3}$$

where $\mathbf{A}$ is the adjacency matrix, whose entry $A_{(i,j)}$ is equal to 1 when v_i and v_j are connected by an edge, and null otherwise. $\mathbf{D}$ is a diagonal matrix whose entry $D_{(i,i)}$ is equal to the valence of v_i.

Since $\mathbf{L}$ is a real sparse positive semi-definite matrix, efficient algorithms can be used to extract some of its eigenvectors. Using the combinatorial Laplacian[5], researchers have analyzed the impact of quantizing the spectral coefficients in different spectrum ranges, e.g., low-frequencies vs. high-frequencies, and found that the Human Visual System (HVS) is less sensitive to perturbations in the lower frequencies [SCOT03]. In 3D watermarking, this observation has led to a thread of research (see Section 3.3).

The main limitation of this discretization is that none of the mesh geometry is taken into account. This is however considered to be one interesting property for a discretization of the Laplacian Operator [WMKG07].

Manifold Harmonics Manifold Harmonics [VL08] correspond to a more complex discretization approach. Applying either Discrete Exterior Calculus or Finite Element Method[6] to the definition of the operator in the continuous setting, i.e. the divergence of the gradient, leads to an approximation of the Laplacian as a non-symmetric matrix: $\mathbf{L} = -\mathbf{D}^{-1}\mathbf{Q}$. $\mathbf{Q} \in \mathbb{R}^{n_v \times n_v}$ is a matrix of cotangent

[4] This definition implicitly assumes a one-to-one correspondence between spectral coefficients and eigenvalues, which does not exist in the aforementioned 1D case, as all but one eigen subspace have dimension 2. This comes from the choice of boundary conditions: for the DFT, periodic boundaries are set; for the DCT, Neumann boundary conditions are used. In the latter cases, the multiplicity of the eigenvalues reduces to 1.

[5] More precisely, a full rank version thanks to the addition of as many constraints, i.e. rows, as the number of connected components of the mesh, since it can be shown that this number equals the order of the null eigenvalue of the Laplacian matrix.

[6] Both approaches yield the same results up to a sign difference.

weights (the stiffness matrix) and $\mathbf{D}$ is a diagonal matrix of triangle facet areas (the mass matrix). Their entries are:

$$Q_{i,j} = \frac{1}{2}\left(\cot(\beta_{i,j}) + \cot(\beta'_{i,j})\right), \tag{2.4}$$

$$Q_{i,i} = -\sum_{j=1}^{n_v} Q_{i,j}, \tag{2.5}$$

$$D_{i,i} = \frac{1}{3}\sum_{f \in \mathcal{N}_1^{\mathcal{F}}(v_i)} |f|. \tag{2.6}$$

$(\beta_{i,j}, \beta'_{i,j})$ are the two angles opposite the edge that connects v_i and v_j, $|f|$ is the area of facet f and $\mathcal{N}_1^{\mathcal{F}}(v_i)$ denotes the set of facets adjacent to v_i.

Because $\mathbf{L}$ is not symmetric, its eigenvectors are no longer orthogonal in $\mathbb{R}^{n_v}$. However, this issue can be addressed by rewriting the eigen-decomposition of $\mathbf{L}$ as a generalized eigenvalue problem:

$$-\mathbf{Q}\mathbf{h} = \lambda \mathbf{D}\mathbf{h}. \tag{2.7}$$

$(\lambda, \mathbf{h})$ is an eigen-pair of eigenvalue and eigenvector. Since (i) both $\mathbf{Q}$ and $\mathbf{D}$ are symmetric, and (ii) $\mathbf{D}$ is positive-definite, it follows that: (i) there still exists a basis of (generalized) eigenvectors $\mathbf{h}^k$ which span the spectral domain, (ii) the associated eigenvalues are real, and (iii) the eigenvectors are $\mathbf{D}$-orthogonal, e.g. $\mathbf{h}^1\mathbf{D}\mathbf{h}^2 = 0$.

In summary, spectral coefficients, in the manifold harmonics case, are defined with the following procedure. First, the generalized eigenvalue problem in Eq. (2.7) is solved. It provides the basis functions $\mathbf{h}^k$ and the associated frequencies that define the mesh spectral domain. To ensure that the basis is orthonormal with regard to the scalar product induced by $\mathbf{D}$, the basis functions are unitary normalized with:

$$\bar{\mathbf{h}}^k = \frac{1}{\|\mathbf{h}^k\|_D}\mathbf{h}^k = \frac{1}{\sqrt{(\mathbf{h}^k)^T \mathbf{D}\mathbf{h}^k}}\mathbf{h}^k. \tag{2.8}$$

Then, the geometry signal $\mathbf{P}$ is projected onto the basis using the modified scalar product, resulting in the triplet of spectral coefficients:

$$[X_k,\, Y_k,\, Z_k]^T = \mathbf{P}\mathbf{D}\bar{\mathbf{h}}^k. \tag{2.9}$$

The inverse transform is identical to the previously defined one:

$$\mathbf{P}^T = \sum_{k=1}^{n_v} \bar{\mathbf{h}}^k [X_k,\, Y_k,\, Z_k] \tag{2.10}$$

Discussion All the spectral decomposition tools result in a set of orthonormal basis vectors, a.k.a. the harmonics $\mathbf{h}^k$, that spans the spectral domain. Thanks to this property, practical applications, such as watermarking, compression or fingerprinting, do not need to fully perform the eigen-decomposition nor estimate all the eigenvectors. One usually only computes specific sub-bands of the spectrum, making this tool applicable to medium-sized meshes, with e.g. more than 10^6 vertices. For instance, as most of the energy of the geometric signal lies within the low-frequency part of the spectrum (eigenvectors associated with the smallest eigenvalues), only this sub-band is commonly extracted. Nevertheless, performing a limited eigen-decomposition of very large sparse matrices still presents practical challenges, especially on consumer-grade hardware.

A crucial theoretical drawback of all the spectral decomposition approaches is that the harmonics, on which the mesh geometry is projected, are derived from a discretized Laplacian that is itself content-dependent. This stems from the very nature of the mesh representation, as an irregular sampling of 2D data in a 3D space. In other words, unlike 1D or 2D, where the canonical cosines and sines are used, the basis functions are themselves content-dependent. This gives rise to a number of issues, especially in the watermarking context.

Multiresolution Analysis

Multiresolution analysis is a processing tool that consists in iteratively decomposing a signal into a coarse base approximation and a series of refining details that can be further decomposed. It was first introduced in the context of meshes [LDW94] to compute a level-of-detail hierarchy, i.e. multiple representations of the same input with increasing details. It has found applications in various domains [Gar99], such as multiresolution editing [ZSS97], progressive rendering, compression [EDD+95] and watermarking.

The initial approach is an extension of the wavelet transform for semi-regular meshes with the so-called 'lazy wavelet decomposition'. As depicted in Figure 2-2, the atomic decomposition operation transforms a group of four triangles into a coarse signal and a refinement signal. The former is a single triangle face that preserves three of the original vertices; the latter is a set of three prediction error vectors in $\mathbb{R}^3$, a.k.a. *the wavelet coefficients*, associated to the edges of the coarse triangle. Each one translates the mid-point of its associated edge to the position of the original vertex that has been removed. This subdivision and correction operation can be equivalently formulated in a series of filter banks. Applying the filters to a mesh creates a series of meshes ranging from detailed to coarse, and a series of associated 3D refinement details. The coarsest mesh is referred to as the *base mesh*.

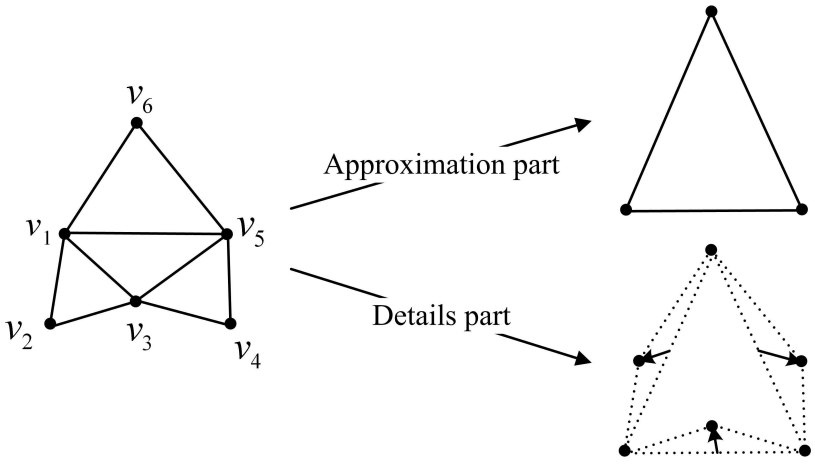


Figure 2-2: Lazy-wavelet decomposition on a 4-1 subdivision mesh. The four facets are decomposed into two parts: (i) vertices v_2, v_4 and v_6 form the coarse mesh, (ii) vertices v_1, v_3 and v_5 are labeled as details, and encoded with the *wavelet coefficients*, i.e. translation vectors from the midpoint of the edges of the coarse mesh.

Lazy wavelets are a particular instance of a broader class of decompositions labeled as 'lifting schemes' [Swe96], that uses linear interpolation instead of direct subsampling. In general, multiresolution analysis presents a low complexity, especially with regard to the existing spectral transforms, and can be applied to meshes with arbitrary topology. However, the need for semi-regularity is

in practice cumbersome. Researchers have explored two strategies to extend multiresolution to meshes with arbitrary connectivity.

A first solution is to define a remeshing procedure that automatically approximates an initial mesh with another that has subdivision connectivity [EDD+95]. This approach enables leveraging the large body of literature on remeshing and also keeps the efficient, but limited, multiresolution tool. However, the remeshing operation may be computationally costly.

Another solution is to directly extend the limited multi-resolution tool to meshes with arbitrary connectivity [VP04]. In this case, the atomic decomposition step is first modified by adapting the subdivision scheme to the local connectivity configuration. Instead of only allowing 4-1 subdivisions, codebooks with series of possible simplification operations are used. Second, the filter banks to compute the coarse and refined geometry information are modified to account for the changes in the subdivision procedure. On the one hand, this solution does not require a costly remeshing preprocess, and is fully capable of handling arbitrary connectivity. On the other hand, the filter bank analysis becomes more complex, and the multiresolution decomposition depends on additional parameters.

2.2 Notions of Watermarking

Digital watermarking is defined as "the practice of imperceptibly altering a Work to embed a message about that Work" [CMB+07]. A 'Work', also called 'content' is a multimedia host signal, which can be an image, a video, or a mesh.

2.2.1 Properties of Watermarking Systems

The basic functional model for watermarking, depicted in Figure 2-3, is made-up of three main components: the *embedder* and the *decoder* (also called the 'detector'), which are placed at both ends of the *communication channel*. Thanks to this representation, a watermarking system can be described with similar concepts as the ones used in communication systems.

A watermark embedder requires three inputs: a content, a *payload* and a secret key. It outputs the *watermarked media*. A detector requires at least two inputs: a content and a secret key. It outputs the payload (if any) estimated from the content.

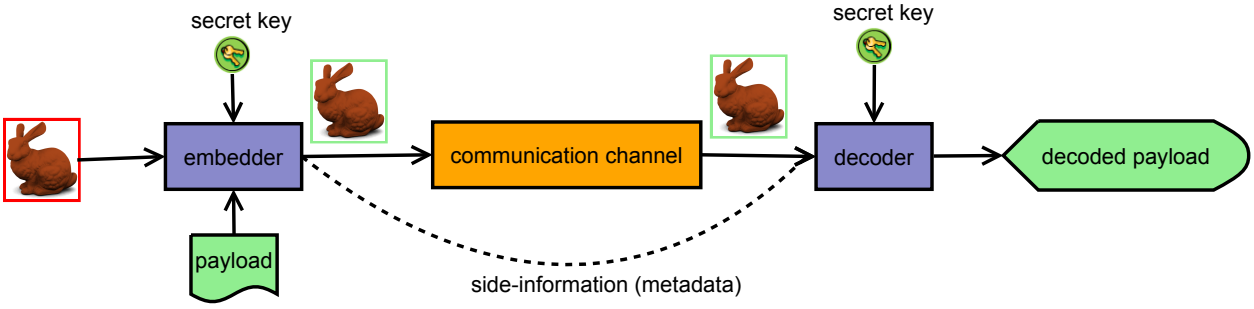


Figure 2-3: Generic watermarking system with its inputs and outputs

3D watermarking can be defined at a high-level as the subfield where the input content for the embedder is a 3D asset. In most cases, two main assumptions are added. First, the 3D asset input in the embedder is represented by a mesh, as defined in Section 2.1. This is motivated by the popularity and versatility of triangle surface mesh representations. Second, the decoder only processes 3D assets, and in most cases, meshes. This restricts the use-cases of 3D watermarking to

applications where 3D assets are not converted to 2D content within the communication channel. This would occur if a decoder were to extract the payload from a rendered image of a mesh. Such a scenario is considered to be out of scope of current watermarking research, and only a handful of attempts have been made to lift this second restriction [BD06].

Capacity

As in other communication systems, the watermark payload, a.k.a. the message, is always assumed to be a series of independent and identically distributed (i.i.d.) random antipodal bits, denoted by $\mathbf{m} \in \{-1, 1\}^{n_b}$). The payload size n_b depends on the target applications, and characterizes the *embedding rate* of a watermarking system, expressed in bits per mesh. The *capacity* of a communication channel is the upper-bound on the information rate that can be correctly conveyed through the channel. Following common practice in 3D watermarking, 'capacity', in this dissertation, refers to the embedding rate of the system.

For *robust* watermarking applications, such as traitor-tracing, common capacities range from a few dozen bits to a few hundred bits, with n_b often chosen among $\{16, 32, 64\}$. In 3D watermarking, some systems are referred to as 'high-capacity' watermark, as they focus on providing a maximum capacity, usually around 1 bit per vertex.

Robustness

The communication channel models all the transformations, a.k.a *attacks*, undergone by a watermarked content before being processed by the decoder. Section 4.2.1 details a list of attacks on meshes taken into account when designing robust 3D watermarking systems. Because of these alterations, the decoded payload $\hat{\mathbf{m}}$ may not be the same as the embedded one. In this context, the *robustness* of the system measures how close $\hat{\mathbf{m}}$ is from $\mathbf{m}$. The assessment metric is the Bit Error Rate (BER).

$$\mathrm{BER}(\mathbf{m}, \hat{\mathbf{m}}) = 1 - \frac{1}{n_b} \sum_{i=1}^{n_b} \delta_{(m_i, \hat{m}_i)}, \tag{2.11}$$

where $\delta_{(m_i, \hat{m}_i)}$ is the Kronecker delta.

Since the BER measures the ratio of erroneous estimated bits over the total number of transmitted bits, it is suitable for multi-bits watermarking. In the specific case where $n_b = 0$, also called 'zero-bit watermarking', the decoder outputs a binary decision on whether or not a content is watermarked. The Receiver Operating Characteristic (ROC) or the Area Under the Curve (AUC), designed to measure the performance of binary classifiers, are then used to assess the robustness. These metrics provide access to soft information; intuitively, they indicate the confidence of the decoder in its decision. In practice, zero-bit watermarking systems may be implemented using a non-null n_b, but the decoder only outputs a binary decision, indicating whether or not the input has been watermarked.

Blind watermarking

As depicted in Figure 2-3, watermarking systems may also rely on *metadata* (sometimes referred to as 'side information') being directly transmitted from the embedder to the decoder. For watermarking purposes, this content-dependent information is always assumed to be unaltered. In other words, the metadata output by the decoder, for some watermarked content, is identical to the one input to the decoder when decoding this watermarked content.

Watermarking systems relying on metadata are labeled as *non-blind*. On the one hand, integrating these systems in industrial work-flows is complex. All the metadata for all the watermarked content generated by the system have to be stored and made available to the decoder. Moreover, the issue of finding the correct metadata associated with a watermarked content inside a (large) database is challenging. On the other hand, since this last issue is considered to be out-of-scope of the watermarking systems, non-blind watermarking methods usually showcase better performance than *blind* methods in terms of robustness.

Nevertheless, as the size of the metadata is a key issue, watermarking systems are further partitioned into non-blind and *semi-blind* ones. In the former, the full content prior to watermarking is used by the decoder. In the latter, the size of the metadata is usually much smaller than the content.

Security

The secret key for the embedder and the decoder adds the *security* aspect into the design of a watermark system. Its common definition is the inability for an unauthorized user (usually referred to as the adversary, or the attacker) to gain access to the watermark communication channel (depicted in Figure 2-4) [Kal01]. Depending on the actual context, an adversary taking control of the channel may decode the payload, remove the payload, alter the payload so that e.g. it corresponds to another user, etc.

As in cryptography, the secret key is often taken as a pseudo-random value, from which some secret parameters of the system are derived. In contrast with metadata, these secret values do not depend on the content and are usually set at the implementation stage. Hence, protecting their secrecy is important. In studies focusing on the watermark security, the main attack contexts are Watermarked-Content Only Attack (WOA), in which the adversary only has access to contents watermarked using the same key, and Known-Message Attack (KMA), where the payloads associated to these watermarked contents are also available to the adversary.

Fidelity

A last key notion for watermarking is the *fidelity* (also called 'imperceptibility'), as underlined in the definition of digital watermarking. The fidelity measures the distortion between a watermarked and non-watermarked content, called the embedding distortion. The definition of a metric to benchmark the fidelity of 3D watermarking systems is still an active field of research. Its main challenges and findings are summarized in Section 2.3.

Conclusion

All these properties are conflicting and lead to trade-offs, such as the robustness vs. fidelity, which is benchmarked by measuring the evolution of the robustness of a system depending on a target embedding distortion. To better identify the intended applications of a 3D watermarking system, different labels are used.

'High capacity' indicates that this aspect of the system is maximized with regard to other properties, in particular, the robustness. 'Fragile' systems usually suggest a low robustness against complex attacks, and authors routinely indicate their usefulness for content authentication. In particular, these systems are designed to help identifying the specific alteration (location, magnitude...) that the content has undergone. In this dissertation, we mainly focus on robust watermarking.

2.2.2 Basic Components of a Watermarking System

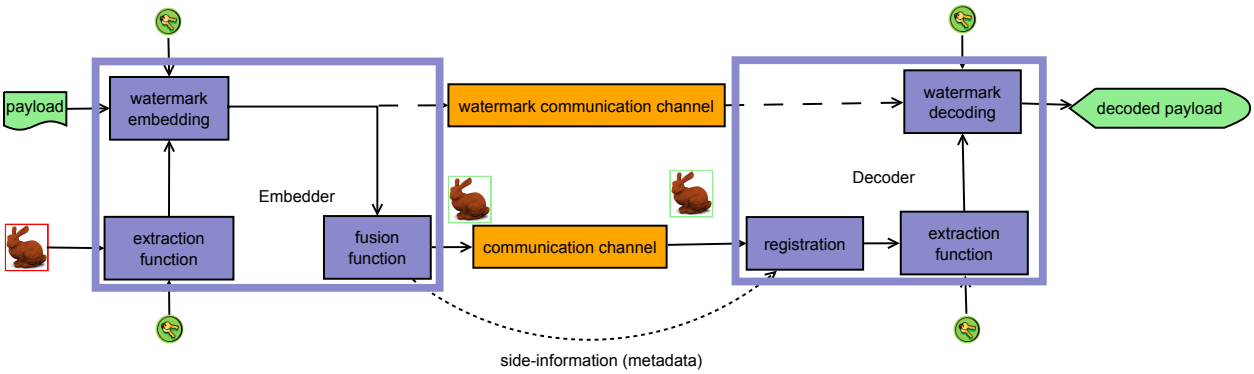


Figure 2-4: Basic components of a Watermarking System

In Figure 2-4, the embedder and the detector are decomposed into basic components. In general, the content adaption and the watermark embedding/detection components may be designed independently, and each one may reuse some existing elements. The decomposition then provides a means to efficiently organize watermarking systems in different categories, reviewed in Chapter 3.

Extraction Function

On the embedder side, the extraction function is a *content adaption transform* that maps the input mesh $\mathcal{M}$ to an element of the *embedding space*. In most cases, this element is a 1D signal $\mathbf{c}$ which is referred to as the *watermark carrier*. It is common practice in watermarking to further partition this function into an initial transform (for instance a Discrete Wavelet Transform (DWT), a FFT...), which is straightforwardly taken from the signal processing field (for instance, audio or image processing), and a subsequent post-process that usually grants additional robustness and specifically targets some watermark applications.

However, the maturity of the existing initial transforms in geometry processing (see Section 2.1.2 and Section 2.1.2) has not reached the same level as in audio or video. 3D watermarking therefore mainly focuses on designing efficient extraction functions, as most system can directly leverage on the properties of a transform. For instance, a rotation-invariant transform usually leads to a rotation-invariant watermarking system. The review of the state-of-the-art in 3D watermarking in Chapter 3 is driven by the properties of the extraction function. Some of these functions are then evaluated in Chapter 4.

Watermark Embedding and Decoding Functions

In the embedding domain, the watermark embedding component maps the watermark carrier to an element $\mathbf{c}^{\mathrm{w}}$ that also depends on both the payload and the secret key. $\mathbf{c}$, a.k.a the host signal, is often a vector. Two standard and well-studied approaches can be used for the embedding and decoding components: Spread Spectrum (SS) and Communication with Side Information.

Spread-Spectrum SS is a widely used technique in communication systems and in watermarking [IKLS97]. In its simplest formulation, the single-bit payload m_1 embedding is written:

$$\mathbf{c}^{\mathrm{w}} = \mathbf{c} + \alpha \mathbf{s}(\eta) m_1, \tag{2.12}$$

where $\alpha > 0$, a.k.a. the *embedding strength*, controls the magnitude of the distortion. $\mathbf{s}(\eta)$ is a pseudo-random spreading-sequence, generated from the secret key η, with null mean value and unit variance. In the decoder, the payload is estimated by projecting the carrier onto the spreading sequence:

$$\hat{m}_1 = \text{sign}\left(\mathbf{s}(\eta) \cdot \hat{\mathbf{c}}\right), \tag{2.13}$$

thanks to the independence of $\mathbf{c}$ and $\mathbf{s}(\eta)$.

SS embedding has been extensively studied in the watermarking literature. In particular, the performance of SS, in cases such as the Additive white Gaussian noise (AWGN) communication channel, can be derived in close-form. Different variations over this SS formulation have been proposed. Multi-bit embedding is achieved through multiplexing. For instance, in time-based (in 3D, spatial-based) multiplexing, $\mathbf{c}$ is partitioned into sequential sequences carrying the different bits. Following the Code Division Multiple Access (CDMA) approach for instance, Eq. (2.12) can be rewritten using a set of orthogonal spreading sequences, each one modulated with a different payload bit.

One major weakness of the SS formulation is that the original content, i.e. $\mathbf{c}$, is equivalent to noise in the watermark communication channel (communication system depicted in the upper-part of Figure 2-4), which is known as the *host interference*. To reduce its effects, the Improved Spread Spectrum (ISS) [MF03] variant adds the third term $-\lambda(\mathbf{c} \cdot \mathbf{s}(\eta))\mathbf{s}(\eta)$ in Eq. (2.12). The parameter λ controls how much of the interference is removed, e.g. at $\lambda = 1$, it becomes null.

For robust watermarking, the security of the system is a major concern. In SS, this property relies upon the secret spreading sequence $\mathbf{s}(\epsilon)$[7]. Researchers have demonstrated the theoretical weaknesses of plain SS [CFF05], and reported practical attacks based on Independent Component Analysis (ICA) in the WOA context. To improve the security, the so-called 'natural watermarking' variation of SS leverages both the symmetry of the (Normal) distribution of the vectors in Eq. (2.12) and the ISS approach. In short, the projections of the carrier onto the spreading sequence are only altered with a sign flip, so that the distribution of watermarked and natural contents are identical [BC07]. In this thread of research, the security becomes another element of the routine watermarking balance between robustness, capacity and fidelity.

Communication with Side-Information at the Transmitter Another remedy for the host interference issue found in plain SS is to use Communication with Side Information (at the embedder) strategies. Contrary to the SS embedding that combines two independent signals, i.e. the original carrier and the payload signal in Eq. (2.12), these strategies adapt the payload embedding to the content. The combination of the original content signal and the payload signal is enhanced by taking into account the former when embedding the latter, which corresponds to using 'Side Information' about the input content.

In Quantization Index Modulation (QIM) [CW99] embedding, a codebook $\mathcal{C}_r$, defined with a quantizer, is associated to the payload symbol $r \in [\![0, R-1]\!]$. The embedding then writes:

$$\mathbf{c}^{\text{w}} = \underset{\mathbf{q} \in \mathcal{C}_r}{\arg\min} \|\mathbf{c} - \mathbf{q}\|. \tag{2.14}$$

In other words, the watermark carrier is set to the nearest codeword in the codebook associated with the payload symbol r to be embedded. At decoding, the estimated symbol corresponds to the one associated to the codebook to which belongs the nearest codeword (distance measured between

[7] At the system level, security may also be provided by the extraction function, for instance if the adversary cannot identify the carrier.

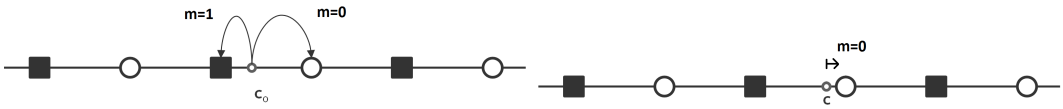
the carrier and the codeword), with:

$$\hat{r} = \underset{r \in [\![0, R-1]\!]}{\arg\min} \min_{\mathbf{q} \in \mathcal{C}_r} \|\hat{\mathbf{c}} - \mathbf{q}\| \tag{2.15}$$

One simple instantiation of this strategy is the Scalar Costa Scheme (SCS) [EBTG03], where the quantization is performed on the individual values of the carrier, i.e. a scalar quantization instead of a high dimensional one. The codebooks are:

$$\mathcal{C}_r = \left\{ k\Delta + r\frac{\Delta}{R} + \epsilon(\eta)\Delta,\ k \in \mathbb{Z} \right\}, \tag{2.16}$$

where Δ denotes a quantization step and $\epsilon(\eta)$ is a pseudo-random secret dither sequence to avoid publicly disclosing the codebooks. In Figure 2-5, the embedding and decoding functions are depicted for a binary payload ($R = 2$).



(a) Embedding (b) Decoding

Figure 2-5: Schematics for the Scalar Costa Scheme and a binary payload embedding 2-5(a) and decoding 2-5(b).

To control the distortion and widen the range of acceptable carrier values, distortion compensation is added. With α the embedding strength, the watermark carrier becomes a linear combination between the original signal and the codebook target value $\mathbf{c}^{\mathrm{w}}$ defined in Eq. (2.16), denoted $\mathbf{t}_{\mathcal{C}_r}$ next:

$$\mathbf{c}^{\mathrm{w}} = \mathbf{c} + \alpha(\mathbf{t}_{\mathcal{C}_r} - \mathbf{c}). \tag{2.17}$$

As for SS, the performance of SCS have been thoroughly examined. For example, its robustness vs. distortion trade-off in the AWGN case, or the influence of α on the security have been analytically studied [PFCnPG05]. Since Δ is a fixed parameter of the system, QIM is sensitive to a scaling of the signal, and solutions such as Rational Dither Modulation (RDM) [PGMBA05] have been proposed.

Fusion Function

The *fusion function* performs an inverse mapping to transform $(\mathcal{M}, \mathbf{c}^{\mathrm{w}})$ into $\mathcal{M}^{\mathrm{w}}$, a.k.a. the watermarked mesh.

The design of a fusion function is usually challenging. The more complex the extraction, the more difficult it is to define its inverse transform. In all but a few cases in 3D watermarking, an exact expression for the inverse cannot be found, which leads to the so-called watermarking *causality issue*. Since the watermarked content is an approximation, applying back the extraction function (in the decoder) does not always lead to $\mathcal{M}^{\mathrm{w}}$ being mapped to $\mathbf{c}^{\mathrm{w}}$ as expected. In practice, this problem occurs when the watermark carrier relies on some intermediary variables that are derived from the original content.

Two main mitigating solutions are used to address causality. The intermediary variables can be explicitly constrained to stay constant once the extraction is performed. Alternatively, the 3 components in the embedder are enclosed in an iterative procedure to ensure that the output $\mathcal{M}^{\mathrm{w}}$

is correctly mapped to $\mathbf{c}^{\mathrm{w}}$. In 1D or 2D watermarking, the causality issue is sometimes dismissed as a theoretical-only obstacle, since state-of-the-art systems undergo an exhaustive benchmarking procedure using large databases. However, there is currently no large database of meshes available for testing 3D watermarking systems. Therefore, when the causality issue does not appear in practical testing of a 3D watermarking method, it may simply be because the variety of the input data is not sufficiently wide.

Attacks

On the decoder side, the extraction function estimates the watermark carrier $\hat{\mathbf{c}}$, which is input to the watermark detection function that estimates the payload $\hat{\mathbf{m}}$. Because of the attacks[8] taking place in the communication channel, $\hat{\mathbf{c}}$ is a potentially altered version of $\mathbf{c}^{\mathrm{w}}$.

Since the range of potential attacks on a mesh is very large, watermarking systems usually exhibit some robustness trade-offs against different attacks (in addition to the common robustness vs. imperceptiblity trade-off) [WLDB08a]. At the design stage, the likelihood that some attack occurs and its perceptual impact on a mesh are both assessed. In general, the less likely or the more perceptible an attack is, the less important it is for the system to achieve a high level of robustness against it. To simplify this approach, mesh attacks are usually grouped into general categories in which the resulting distortions share similar properties.

Similarity Transforms These are combinations of rotation, translation, and uniform scaling of the mesh (same scale in all directions). A common property of these attacks is that they are *content-preserving*, and naturally occur when using meshes. For this reason, most watermarking systems exhibit a complete invariance against these attacks, i.e. their effect on the watermark carrier is null.

Geometry-driven Alterations of the Surface These cover various types of attacks. General *affine transform* (of which similarity transforms are a particular case), include e.g. anisotropic scaling and projections. Noise addition procedures result in vertices being moved from their original position according to some random noise additive vector. Denoising procedures, such as *smoothing* and *fairing*, have an opposite effect on the mesh surface [BKP+10]. Finally, mesh compression attacks modify the vertex positions by, e.g., quantizing the individual coordinates, or by using the more involved spectral tools (see Section 2.1.2).

All these attacks usually correspond to valumetric attacks. They only affect the individual sample values in the watermark carrier signal $\tilde{\mathbf{c}}$, but not their ordering, and do not lead to interferences in-between carrier values. Robust watermarking systems are routinely benchmarked against these attacks.

Connectivity Attacks In essence, all attacks that do not modify the individual positions in $\mathbf{P}$ are connectivity attacks. The simplest one is the content-preserving 'vertex re-ordering' attack, in which only the indices $i \in [\![1, n_v]\!]$ of the vertices are changed (and the elements of $\mathcal{F}$ are modified accordingly) [Hop99]. Another possible attack is a re-triangulation, where pairs of adjacent triangle facets are modified by flipping their common edge [DHKL01].

Since watermarking systems are likely to face these attacks in real-life applications, they are also parts of standard benchmarking tests.

[8] This section does not deal with the so-called 'security' attacks, which aims at accessing the watermark communication channel.

Resampling Attacks Resampling procedures modify the geometry ($\mathbf{P}$ and $\mathcal{V}$), and usually also affect $\mathcal{F}$. The aim is to change the mesh representation without introducing too large of an alteration in the underlying surface. Common instances are refinement procedures (e.g., with facet subdivision) to increase the sampling density, simplification (removing some vertices, faces and edges), or full remeshing operations. Simplification is ubiquitous in mesh processing; the performance of watermarking systems against this type of attack is of particular interest for practical use-cases.

Topological Alterations This category covers a wide range of mesh operations. Contrary to the previous categories, the topology of the underlying surface represented with the mesh may be altered. The number of connected components may for instance change, or a new handle may be created. In the well-known cropping attack, some parts of the mesh are removed. In general, most of the topology-altering operations aim at fixing the defects produced within the 3D acquisition and processing pipeline, e.g. micro-holes, or duplicated features [Ju09]. The robustness of a watermarking system against these complex operations is usually considered a lesser issue.

Finally, the two following types of attacks have been overlooked so far for the most part of the 3D watermarking literature, as they are usually complex to benchmark and difficult to resist.

Mesh Conversions Converting a mesh to another type of 3D representation is an active research field. These conversions may at first appear out-of-scope of mesh watermarking, as only meshes are input to the watermark decoder. However, for real-life applications of 3D watermarking, e.g. print and scan attacks, while the input and output of the communication channel are meshes, multiple conversions are likely to occur. Because of their diversity and complexity, the robustness of a system against mesh conversions is nonetheless most of the time not benchmarked.

A notable exception is formed by the simple point cloud conversion, in which the connectivity of a mesh is removed. Not only is it easy to integrate point cloud attacks into a benchmark, but it also provides an effective way to determine how much a watermarking system depends on the mesh connectivity.

Surface Deformation Surface deformation is commonly used to create 3D animations or during modeling. For instance, a user may change the original pose of the mesh, e.g. by bending an arm mesh at the elbow. More complex applications involve e.g. morphing, where the original surface is continuously transformed into a different object.

Frameworks for surface deformation rely on (i) *handles*, i.e. specific elements of the mesh surface or within the 3D object, that drive the user-defined deformation, and (ii) a variational formulation to propagate the deformation inferred from the changes in the handle positions. The most popular instance is the skeleton-based *pose* of the mesh. Given a 'rigged mesh', i.e. a set of joints and bones attached to the vertices, a user may change their positions, thereby animating the mesh, much like a puppet. In general, the pose alteration corresponds to an (almost) isometric deformation of the surface: *intrinsic* quantities, such as geodesic distances, are left unmodified whereas *extrinsic* quantities, such as Euclidean distances, are changed.

At the end of Section 2.1.1, it was mentioned that only static meshes are used throughout this dissertation. In this case, surface deformation methods may still be applied to the static input mesh, and are considered as attacks. More complex deformations, such as an arm bending with the simulation of muscle contraction, or morphing, are currently out-of-scope of 3D watermarking research.

It is also common practice to simply differentiate between (i) content-preserving attacks that do not change the mesh surface, (ii) connectivity-preserving attacks that only alter the vertex positions $\mathcal{P}$, and (iii) connectivity-altering attacks, which alter both $\mathcal{V}$ and $\mathcal{F}$. In Chapter 4, numerous alterations are thus reviewed to form an experimental benchmark for robustness assessments.

Registration

The registration component of a watermarking system ensures that the embedder and the decoder are *synchronized*. Informally, correct synchronization is achieved when the implicit ordering and structure of the conveyed information stays the same at both ends of a communication system. For instance, in the AWGN communication channel, all the values in the estimated watermark carrier at the decoder side are one-to-one mapped to the elements of the original carrier output by the embedder. A registration is then unnecessary.

In contrast, if the communication channel introduces some form of delay (for instance a time-shift in an audio signal), the synchronization between the decoder and the embedder may be lost, as the ith estimated watermark carrier $\hat{c}_i$ corresponds to $c_{i-\tau}$, i.e. the delayed carrier. In this case, the registration component tries and realign the estimated carrier thanks to an estimation of τ. Specific means to estimate this delay in the 3D context are reviewed in Section 9.1. For more complex desynchronizing attacks in the communication channel, an inter-symbol interference can occur. The estimation of $\hat{c}_i$ is then impacted by multiple emitted carrier values, e.g., both c_i and c_{i-1} instead of the single c_i.

The consequences of a loss of synchronization are critical, and most watermarking systems have a dedicated registration mechanism. In non-blind watermarking, this registration often consists in realigning the input with regard to the metadata provided to the decoder. In blind watermarking systems, the registration is commonly referred to as a '*resynchronization*'. In most cases, the component is partitioned in two symmetric parts, as the resynchronization mechanism is applied in both the embedder and the decoder.

In Chapter 9, a dedicated resynchronization component for 3D watermarking is discussed.

2.3 Notions of 3D Watermarking Fidelity

The fidelity of the watermark embedding, a.k.a. the watermark 'visibility' or 'imperceptibility', is one of the main elements that characterize a watermarking system (see Section 2.2). It is measured with an embedding distortion metric, and is informally defined as "the perceptual similarity between the original and watermarked versions of the cover Work," [CMB+07] i.e. the mesh input to the embedder.

Watermark fidelity metrics are closely related to the ones used in lossy compression, as they all need to account for the specificities of the Human Visual System (HVS). These have been extensively studied for still image and video in the past decades, but they are not yet fully understood for 3D geometry [CLL+13]. As there is currently no universally-accepted way of measuring the fidelity of 3D watermarking systems, watermark benchmarks use different metrics which are difficult to compare. In the following, the most common ones are summarized.

2.3.1 Objective Metrics

Hausdorff Distance

Let $\mathcal{M}$ denote the original mesh and $\mathcal{M}'$ be its altered version. The Euclidean distance from a point $\mathbf{p} \in \mathcal{M}$ to $\mathcal{M}'$ is given by:

$$\mathrm{d}(\mathbf{p}, \mathcal{M}') = \min_{\mathbf{p}' \in \mathcal{M}'} \|\mathbf{p} - \mathbf{p}'\|. \tag{2.18}$$

The Hausdorff distance measures the similarity between two surface meshes based on:

$$\mathrm{d}_{\mathrm{Ha}}(\mathcal{M}, \mathcal{M}') = \max_{\mathbf{p} \in \mathcal{M}} \mathrm{d}(\mathbf{p}, \mathcal{M}'), \tag{2.19}$$

i.e. the greatest Euclidean distance, over all the elements of $\mathcal{M}$, to $\mathcal{M}'$. Contrary to a distance metric, Eq. (2.19) is asymmetric; in general, $\mathrm{d}_{\mathrm{Ha}}(\mathcal{M}, \mathcal{M}') \neq \mathrm{d}_{\mathrm{Ha}}(\mathcal{M}', \mathcal{M})$. These two quantities are sometimes referred to as the *forward* and *backward* distances. The symmetric Hausdorff distance is then defined as:

$$\mathrm{d}_{\mathrm{H}}(\mathcal{M}, \mathcal{M}') = \max\left(\mathrm{d}_{\mathrm{Ha}}(\mathcal{M}, \mathcal{M}'), \mathrm{d}_{\mathrm{Ha}}(\mathcal{M}', \mathcal{M})\right) \tag{2.20}$$

The Hausdorff distance is mathematically well-defined and presents important theoretical properties; but it is sensitive to small alterations that may not be perceived by a user, and, in all, does not correlate well with the HVS [LC10].

Root Mean Square Error

To address the sensitivity to small alterations, the Root Mean Square (RMS) is often used instead of the Hausdorff distance:

$$\mathrm{d}_{RMS}(\mathcal{M}, \mathcal{M}') = \sqrt{\frac{1}{A} \iint_{\mathbf{p} \in \mathcal{M}} \mathrm{d}(\mathbf{p}, \mathcal{M}')^2 \, \mathrm{d}M}, \tag{2.21}$$

where A is the area of the surface mesh. Since $\mathrm{d}_{RMS}(\mathcal{M}, \mathcal{M}')$ is also asymmetric, it is turned into a metric with the Maximum Root Mean Square (MRMS):

$$\mathrm{d}_{MRMS}(\mathcal{M}, \mathcal{M}') = \max\left(\mathrm{d}_{RMS}(\mathcal{M}, \mathcal{M}'), \mathrm{d}_{RMS}(\mathcal{M}', \mathcal{M})\right). \tag{2.22}$$

Both distances in Eq. (2.20) and Eq. (2.22) are usually discretized for practical purposes.

Researchers have proposed efficient tools to approximate these distances [CRS98, ASCE02]. They usually perform a preliminary resampling of the input surfaces, to decrease the dependency to the original sampling and connectivity. Intuitively, this step enables a meaningful comparison between e.g. two different mesh samplings $(\mathbf{P}, \mathcal{V}, \mathcal{F})$ and $(\mathbf{P}', \mathcal{V}', \mathcal{F}')$ of the same surface. The main drawback of this solution is its computational complexity. To speed-up the computation, the discretized RMS may be approximated with:

$$\mathrm{d}^{\star}_{RMS}(\mathcal{M}, \mathcal{M}') \approx \sqrt{\frac{1}{n_v} \sum_{i=1}^{n_v} \min_{\mathbf{p}' \in \mathbf{P}'} \|\mathbf{p}_i - \mathbf{p}'\|^2}. \tag{2.23}$$

In this case, no resampling is involved, and the distance is computed with a fast nearest-neighbor search in $\mathbb{R}^3$. In general $\mathrm{d}_{RMS}(\mathcal{M}, \mathcal{M}') \neq \mathrm{d}_{RMS}(\mathcal{M}', \mathcal{M})$. Eq. (2.23) can be used on differently

sampled meshes, although the result heavily depends on the sampling distributions. The MRMS definition in Eq. (2.22) is left unchanged.

When both input meshes share the same connectivity ($\mathcal{V} = \mathcal{V}'$ and $\mathcal{F} = \mathcal{F}'$), and the distortion is small, the neighbor search in Eq. (2.23) is unnecessary, and the RMS can be further simplified, thanks to the underlying one-to-one mapping between the vertex positions:

$$\mathrm{d}_{\mathrm{RMS}}(\mathcal{M}, \mathcal{M}') = \sqrt{\frac{1}{n_v} \sum_{i=1}^{n_v} \|\mathbf{p}_i - \mathbf{p}'_i\|^2}, \tag{2.24}$$

which is already symmetric. In Chapter 6, the quadratic programming formulation for 3D watermarking approximates the Square Error (SE) metric with this formula.

Quadric Error Metric

In the context of mesh simplification, the Quadric Error Metric (QEM) was initially proposed as a criterion to order a series of edge collapse operations, i.e. create an efficient priority queue containing pairs of vertices that will be merged, and find locally optimal vertex locations. The QEM between the initial vertex v and its modified version v', is the sum of squared distances from $\mathbf{p}'$ to the original neighboring planes around v (the supporting planes of the triangle facets in $\mathcal{N}_1^{\mathcal{F}}(v)$). Intuitively, it focuses on the amount of change in the normal direction with regard to the original surface, and overlooks a vertex displacement within the (local) tangent plane at $\mathbf{p}$. Formally:

$$\mathrm{d}^2_{\mathrm{QEM}}(v, v') = \sum_{f \in \mathcal{N}_1^{\mathcal{F}}(v)} (\mathbf{n}_f \cdot (\mathbf{p} - \mathbf{p}'))^2, \tag{2.25}$$

where $\mathbf{n}_f$ is the unit normal to facet f[9].

When both $\mathcal{M}$ and $\mathcal{M}'$ share the same connectivity (or when the mapping between vertices is trivial), the QEM can be defined at mesh level, by, e.g., averaging the contribution of the individual vertices:

$$\mathrm{d}^2_{\mathrm{QEM}}(\mathcal{M}, \mathcal{M}') = \frac{1}{n_v} \sum_{i=1}^{n_v} \mathrm{d}^2_{\mathrm{QEM}}(v_i, v'_i) \tag{2.26}$$

Although the two connectivities are identical, $\mathrm{d}^2_{\mathrm{QEM}}(\mathcal{M}, \mathcal{M}')$ is not symmetric (because of the $\mathbf{n}_f$ term in Eq. (2.25)). Following the approach indicated by Eq. (2.22), the maximum between $\mathrm{d}^2_{\mathrm{QEM}}(\mathcal{M}, \mathcal{M}')$ and $\mathrm{d}^2_{\mathrm{QEM}}(\mathcal{M}', \mathcal{M})$ may finally be used to define a proper distance.

Geometric Laplacian

Following the successful introduction of the spectral transform (see Section 2.1.2) for mesh compression, researchers have investigated the use of the Laplacian to also assess the compression distortion. Denote by $\mathbf{L}$ the Laplacian matrix computed from an original mesh, and $\mathbf{L}'$ its altered version. $\mathbf{P}_\delta$ and $\mathbf{P}'_\delta$ are the results of applying the Laplacian matrix to the vertex position: $\mathbf{P}_\delta = \mathbf{PL}$; they are related to the local mean curvature and normal [Tau95], and capture information about the surface

[9] This expression is equivalent to the original one [GH97], which rather focused on the computational efficiency of the QEM.

shape. The Laplacian-based metric is then defined with:

$$\mathrm{d_L}(\mathcal{M}, \mathcal{M}') = \sqrt{\sum_{\mathbf{p}_\delta \in \mathbf{P}_\delta} \|\mathbf{p}_\delta - \mathbf{p}'_\delta\|^2}, \tag{2.27}$$

which requires a mapping between the vertices of the two meshes[10]. In its initial form [KG00a], the Laplacian metric is (i) combined with the RMS in a linear trade-off controlled with $\lambda \in [0, 1]$, and (ii) based on a geometric discretization of the Laplacian (hence the name of the metric). However, as exploited in Chapter 6, the Laplacian-based metric can be used with other discretization e.g. the combinatorial one.

$\mathbf{p}_\delta$ is the difference in a vertex position after and before a Laplacian smoothing. Hence, $\mathbf{p}_\delta$ is indicative of the local roughness, which is instrumental to designing perceptually-correlated distortion metrics, and $\mathrm{d_L}$ is related to the change in the local roughness.

2.3.2 Perceptually Correlated Metrics

Two main threads of research have tackled the issue of perceptually-correlated metrics. In general, these works start by defining a local objective distortion metric that takes into account some properties of the HVS. The global distortion between meshes is computed by aggregating these local results. A perceptual benchmarking study is then undertaken to measure the correlation between the metric and the distortion perceived and rated by a pool of users. Eventually, a parameterized version of the metric is fitted to the user ratings thereby creating a quantitative perceptual metric suitable for a target application, such as watermarking.

Roughness-based Measures

A computationally efficient solution to estimate the local roughness is based on dihedral angles, i.e. the angle between the normals directions of adjacent facets [WHST01]. This approach has been extended to a multi-scale one, and integrated into a first metric that measures the distortion as the log-variation in the global mesh roughness [CGEB07]. Following the underlying principle behind the Geometric Laplacian based metric, researchers have also proposed estimating the roughness through the local variations of the vertex positions between an original mesh and its smoothed version. In this case, the distortion metric is still the log-variation in the global roughness [CGEB07].

A more elaborate estimation computes the roughness through principal curvatures [Lav09]. The local roughness is defined as the variation in the maximum curvature, caused by a mesh smoothing, and is averaged on a small geodesic surface patch around each vertex. This estimator represents a preliminary step towards the construction of the metrics described next.

Mesh Structural Distortion Measure

The Mesh Structural Distortion Measure (MSDM) is a metric based on statistics derived from the mesh curvatures [LDD+06], in an attempt to extend the structural similarity concept from images [WBSS04] to meshes. For two local mesh neighborhoods and the vertices within them, the MSDM is computed by (i) estimating the average, the standard deviation and the covariance of the maximum curvatures in each neighborhood, using the normal cycle estimator to compute the principal curvatures at a vertex (see Section 2.1.2), and (ii) comparing these statistics between both

[10] In the context of spectral-based compression, this mapping is trivial.

neighborhoods. In practice, geodesic neighborhoods around every vertex are used. The MSDM at mesh level is defined with the Minkowski norm of the local MSDM terms.

Because the MSDM is restricted to the comparison of meshes with the same connectivity, a revised version denoted by MSDM2 [Lav11] was presented. Its main differences consists in (i) constructing a mapping between the local neighborhoods in two meshes so that alterations of the connectivity are dealt with, and (ii) performing a multi-scale comparison using various sizes of local neighborhoods. While this metric achieves the best results in terms of correlation with user perceptions [CLL+13], its computational complexity is much larger than the original MSDM.

In short, all the metrics reviewed above measure the distortions in the mesh geometry only. However, meshes are associated to multiple additional components, e.g. texture information, and used within complex rendering pipelines. It is common knowledge in the 3D modeling community that these aspects greatly affects the way a mesh is perceived. The distortion impact at the rendering level, which is actually the one perceived by users, has mostly been overlooked [PCB05], and is the next main challenge in this field of research.

2.3.3 Using Distortion Metrics in 3D Watermarking

Assessing the watermark fidelity is central for the benchmarking of watermarking systems. While well-established metrics for the capacity and the robustness are available, namely the payload size and BER, the embedding distortion has to be quantitatively measured with one of the metrics reviewed, whose properties and sensitivity towards the different alterations are not similar.

Another use for distortion metric is to enable watermark systems to take into account the specificities of the HVS to reach high levels of imperceptibility. This approach is commonly referred to as *perceptual shaping*. One of the most common strategies is to take advantage of the so-called 'masking effect'. As users are more sensitive to alterations introduced in homogeneous regions than in highly textured ones, the watermark strength may be e.g. locally textured in rough regions.

In 3D watermarking, instances of perceptual shaping can be found in a wide range of works, e.g. [DHM10, KBT10, WLDB11] which are discussed next.

Chapter 3

State-of-the-Art in Robust 3D Watermarking

To avoid any confusion between the watermark embedding function and the mesh embedding in $\mathbb{R}^3$, the latter is henceforth referred to as the mesh 'geometry', which is common practice in the context of 3D watermarking [Wan09].

This survey of the state-of-the-art research in 3D watermarking focuses on robust systems for static meshes. After a short summary on geometry-preserving watermark in Section 3.1, the review of geometry-altering robust watermarking systems is structured around their extraction function: first at a coarse level, with the common taxonomy employed spatial (Section 3.2), spectral (Section 3.3) and multiresolution-based (Section 3.4) embedding domains [Wan09, Luo06, Yan13], then, through a finer structuring of the basic mechanisms partaking in the extraction procedure.

3.1 Geometry-preserving Watermark

Limited research has focused on designing geometry-preserving fusion functions. In the first study on 3D watermarking, a proposed watermark carrier is the density of faces [OMA97]. This carrier is altered to create meaningful patterns on the surface, visible in a wireframe representation. To complete the embedding, the fusion function locally applies subdivision operations. While this also increases the number of vertices and alters the vertex locations $\mathbf{P}$, the actual geometry of the mesh surface remains fixed, which supports the geometry-preserving labeling.

In another seminal algorithm, the faces in selected triangle strips are cropped out of the watermarked mesh [OMA98]. The extraction process uses a so-called Triangle Strip Peeling Symbol (TSPS) sequence to enumerate triangles. This results in a binary carrier signal[1], whose values are modulated through facet deletions. As opposed to the first study, the introduction of holes alters the geometry of the surface, albeit this is done in such a way that $\mathcal{V}$ and $\mathbf{P}$, referred to as the mesh geometry in 3D watermarking, are not modified.

On the one hand, geometry-preserving systems are theoretically immune to common signal processing operations on the mesh geometry, such as noise addition to the vertex positions or mesh smoothing. On the other hand, the visibility of the watermark is a major obstacle for robust watermarking. Even when this visibility issue is addressed [APDP10], there still remains a robustness problem. The mesh connectivity indeed exhibits a low resilience against mesh processing operations

[1]The carrier is one of the two possible remaining edges in a facet, once the third edge has been used to enter the facet in the traversal procedure.

such as simplification. Connectivity may even be fully lost when converting the mesh, for instance when performing a trivial conversion to point-based representation. In the two watermarking approaches above, the low robustness of the connectivity information is critical. Research in robust 3D watermarking instead mostly focused on finding stable extraction functions, often combined with geometry-altering-only fusion functions. Geometry-preserving strategies have been further explored for high-capacity watermarking or fragile-watermarking purposes, which are out-of-scope of this survey.

3.2 Spatial-domain 3D Watermarking

Watermarking systems where the extraction function is directly based on geometric properties of the vertex positions are referred to as spatial-domain approaches. Formally, the extraction function defines a vector field from the mesh surface to $\mathbb{R}^n$, a.k.a. the embedding domain. In most cases, the extraction reduces to a scalar field with $n = 1$. To allow for simpler fusion functions, the extraction is also restricted to the mesh vertices and this mapping is denoted by: $(\mathcal{V}, \mathbf{P}) \to \mathbb{R}$.

3.2.1 Watermark Carriers based on Local Geometric Properties

The watermark carrier corresponds to some local geometric properties, computed within small neighborhoods around vertices. The synchronization of the carrier relies on enumerating the mesh primitives (facets, vertices) along a canonical traversal procedure.

Synchronization based on Mesh Traversals

The Triangle Strip Peeling Symbol (TSPS) sequence forms the basis of the synchronization mechanism in watermarking proposed by Cayre et al. [CM03]. In this blind watermarking system, the triangle facets are enumerated in an order (mainly) derived from a secret key, and no longer from the payload, which is embedded in a geometric primitive instead of in the connectivity. The watermark carrier in a facet is the position of the projection of a vertex onto its opposite triangle edge. The embedding function is a modified Quantization Index Modulation (QIM), where the quantizer is non-uniform and restricts the carrier to lie within some target segment in $\mathbb{R}$. Because this system can embed around one bit per vertex, it is sometimes referred to as a 'high-capacity' watermark [Wan09].

This system has been improved to showcase the benefits of perceptual shaping in 3D watermarking [KBT10]. The non-uniform quantizer in the embedding function is modified so as to relax the fidelity constraint and allow for a greater robustness. The embedding strength is then locally adapted according to the surface roughness. The watermarking energy is reduced in smooth and increased in rough mesh regions. This approach improves performance against noise addition.

Nevertheless, the perceptual shaping protocol is complex and unreliable, as it is based on a series of simplified subjective experiments that also determine the distortion alignment in the benchmark. The actual embedding distortion is finally not measured.

Because the decoders also perform a mesh traversal, these systems are prone to synchronization issues. In particular, they exhibit low resilience to connectivity-altering attacks, such as remeshing.

Local Mesh Descriptors

To devise a synchronization procedure without any mesh traversal at decoding, some authors have explored conjointly embedding the payload bit and its index in the carrier signal [WH09]. Both

values are carried by different signals, whose samples are inherently linked. This approach was embodied using carriers based on integral invariants [PWHY09]. Given $r \in \mathbb{R}^+$, the first carrier at a vertex v is the area invariant: the surface area of the intersection between the mesh and the sphere of radius r centered at v. The second carrier is the volume invariant: the intersection between the ball centered at v with radius r, and the inside part of the 3D object. For watermarking purposes, the authors further proposed using an efficient approximation method to modify the invariant values. This system is referred to as 'semi-fragile', but when compared with other robust algorithms, the resilience against noise addition and cropping is promising.

Since the invariant computation depends on an entire surface patch, a causality issue arises, as altering one invariant modifies the nearby invariants. The authors tackle this problem with a suboptimal sphere packing procedure where non-overlapping (independent) surface patches are watermarked. However, this yields a chicken-or-egg problem: the decoding of the payload index, that ensures the synchronization, relies on the sphere packing, for which a synchronization strategy is in turn needed.

Discussion

Integral invariants are one among many categories of local mesh descriptors [HPPLG11]. Although descriptors have received a major interest, two problems have hindered their adoption as 3D watermark carriers.

The first relates to causality and synchronization. The spatial support to compute a descriptor is not discrete and the independence of the watermark carrier samples must be explicitly enforced. One may embed the payload in non-overlapping support patches, but the packing/selection mechanism must be exactly repeated at decoding. This further challenges the registration procedure, as recovering the non-overlapping patches faces issues akin to the ones resulting from mesh traversals.

The second main obstacle is that the definition of most descriptors is not easily invertible. Finding a fusion procedure that grants control over the carrier values while enabling reaching the targets set by the embedding function is usually challenging.

Chapter 9 describes an approach to overcome these obstacles with the integration of components of a local mesh descriptor within a watermarking framework, in order to enhance its robustness against the cropping attack.

3.2.2 Distribution of Euclidean Distances

Pioneering Work

To circumvent some of the aforementioned synchronization problems, a large body of 3D watermarking research focused on the distribution of the distances between a reference primitive and the vertex positions.

In the seminal blind Vertex Flood Algorithm (VFA), the center of mass $\mathbf{g}(f)$ of a carefully selected triangle facet f is set as a reference primitive [BB00]. The watermark carrier is made-up of the Euclidean distances between some vertex positions and this reference primitive: $c_i = \|\mathbf{g}(f)\mathbf{p}_i\|$. A histogram whose edges are spaced by a step $\Delta \in \mathbb{R}^+$ is populated with these distances. To embed a sequence of n_b payload bits, the carriers lying within one histogram bin are quantized according to a binning scheme with 2^{n_b} states. The fusion function reduces to straightforward relocation of the vertices along $(\mathbf{g}, \mathbf{p}_i)$.

There are two main shortcomings in this system. The preliminary selection of facet f, which is equivalent to the initialization of a mesh-traversal procedure creates a localization issue; the proposed application of this system is fragile watermarking. The capacity is also limited: the

payload size has dire negative impacts on the robustness. Nevertheless, the watermark carriers are intrinsically invariant to rigid transforms, independent of the connectivity, and the extraction mechanism is oblivious to vertex reordering in the mesh files.

In this initial research, the notion of letting the distribution of distances be the watermark carrier is not explicitly stated, and every distance is indeed an independent carrier sample. However, the decoding procedure advantageously relies on the average inside the histogram bins, instead of the decoding of individual carrier values. This has stimulated new research to define the carriers with quantities derived from the distribution of distances.

In a subsequent series of studies, three main changes have been introduced in the watermark carrier of the VFA [KTP03, ZTP05].

First, the payload is no longer embedded in the Euclidean distances, but in a prediction error signal, derived from the local variations of these distances in a small spatial neighborhood[2]. Second, multi-bit embedding is achieved by partitioning the vertices according to θ_i (angle in the spherical coordinates system) and by embedding one payload bit per region. Third, the reference primitive $\mathbf{g}(f)$ is replaced with the mesh center of mass $\mathbf{g}$. In other words, the Euclidean distances are the so-called radial distances ρ_i in the spherical coordinates system.

The center of mass and the radial distance based distribution are instrumental in the improvement of the robustness compared with the VFA. As confirmed in Chapter 4, ρ_i indeed exhibits high stability, making it a suitable watermark carrier. The localization issue in the VFA no longer occurs, and the robustness against connectivity alterations increases.

However, using a local neighborhood to compute the prediction error brings back some other problems that have been discussed for local-descriptor based carriers, e.g. it makes for a complex fusion function. Furthermore, because of the arbitrary mesh partitioning based on θ_i, a mesh rotation can desynchronize the carrier sequence. The synchronization is then ensured thanks to a preliminary mesh alignment. Both the embedding and the decoding starts by aligning the principal axis of the mesh Principal Component Analysis (PCA) with the z coordinates, which is a particular case of the normalization reviewed in Section 2.1.2 .

Watermarking of the Distribution of Radial Distances

Cho et al. have described a watermarking system that embeds the payload in the distribution of the distances between all vertices and the center of mass $\mathbf{g}$ of the mesh [CPJ07]. The watermark carrier is defined by the averages inside the n_b bins of the histogram of the distances ρ_i (referred to as the 'vertex norm'). Experimentally, the average values usually lie close to the middle of the bin. To embed a bit $m_j = +1$ (respectively $m_j = -1$), the average inside bin j is raised above (resp. lowered below) the middle of the bin. The fusion function that maps back the target average values into the watermarked mesh is referred to as the 'histogram mapping transform'. In essence, the transform corresponds to an iterative power function applied onto all ρ_i within the same bin: after each iteration, the exponent in the power function is modified so as to increase or decrease the resulting bin average. At decoding, the payload is extracted in a blind manner by comparing the average inside each bin to the middle of the bin.

In a variant, the authors propose using the variance inside a bin as the watermark carrier. The basic components of the system and their inner workings are preserved.

Contrary to previous work, neither a registration nor a localization procedures are needed, which greatly simplifies the synchronization between the embedder and the detector. This algorithm

[2]More specifically, in either one of the one-sided variances of a Normal distribution model fitted to the prediction error.

exhibits increased robustness and can embed around 75 bits of payload within meshes having a few dozen thousand vertices. As of today, this system is one of the most robust and computationally efficient [Luo06] and therefore serves as a reference *baseline* for several follow-up algorithms that deal with some of its remaining limitations.

One of such weaknesses is that the efficiency of the histogram mapping transform depends on the fact that ρ_i is uniformly distributed inside a histogram bin. Part of the causality issue is also not addressed within the embedder, as the stability of the mesh center of mass is overlooked. To explicitly address the causality issue and provide a more systematic fusion approach, the baseline algorithm has been incorporated within a Quadratic Programming (QP) framework [HRAM09]. In Chapter 6, we present a generalization of this formulation to address three limitations of this QP formulation.

Cropping Issue in Radial-distance-based 3D Watermarking

One important issue in the previous approach of Cho et al. is its sensitivity to cropping operations. This inherent weakness originates from the choice of the largest computation support for the reference primitive. On the one hand, using the center of mass of the whole mesh grants robustness against valumetric attacks on the vertex positions, such as noise addition or smoothing. On the other hand, it increases the sensitivity to synchronization attacks affecting the vertices, such as cropping.

To alleviate this issue, some authors have proposed to repeatedly embed the payload in different regions corresponding to some 'prong' neighborhoods [RAMC07]. Intuitively, prongs are the extremities of a 3D object and can be captured by computing the pairwise (between pairs of vertices) geodesic distances on the mesh surface. This approach is motivated by the observation that cropping is unlikely to affect all the regions where the payload is embedded. But this partitioning strategy presents two shortcomings.

First, restricting the watermark to prong neighborhoods reduces the number of samples populating the histogram, and diminishes the overall performance. It also amplifies the overlooked causality issue if the watermark embedding alters the center of mass. Empirically, this issue seldom appears in the baseline, because, in most meshes, the vertex relocations in one part are usually compensated by the vertex relocations in another part[3]. For smaller neighborhoods, this fortunate phenomenon is less likely to occur.

Second, geodesic distances are both computationally expensive and sensitive to valumetric attacks, compared with Euclidean distances (see Chapter 4). The prong detector in the system [VKS05] is therefore strongly limited both in terms of complexity and stability[4]. This has large negative repercussions on the global performance. In particular, when the prong neighborhoods are slightly misaligned (between embedding and decoding), the repercussion is similar as the one caused by applying a small cropping attack on the baseline algorithm, in which case the performance abruptly drops.

Integral Quantities in Radial-distance-based 3D Watermarking

Another weakness in the baseline is that $\mathbf{g}$ is computed as the average vertex position. This definition is sensitive to resampling operations, especially when lifting any uniformity constraint on the sampling. Integral definitions of the center of mass have therefore been advocated for

[3] due to frequent symmetries in 3D objects.

[4] In this context, the stability measures the ability of the detector to retrieve the same prongs at the same location after an attack.

instead. These definitions, based either on volume-weighted [ZC01], or surface-weighted [GAP08] vertex positions, are less sensitive to sampling defects and remeshing procedures. To improve the robustness of the baseline watermark carrier, the prong-based system [RAMC07] uses a surface-weighted center of mass, and, in the histogram, the radial distances are weighted by the surface area of the 1-ring neighborhood.

Although not presented in such a way, some authors also explored a fully integral variant of the baseline system using only moment-based quantities [WLDB11]. Given a volume in $\mathbb{R}^3$ and its indicator function $\mu(x, y, z)$, the moment m_{pqr} is defined as:

$$m_{pqr} = \iiint_{\mathbb{R}^3} x^p y^q z^r \mu(x, y, z) \, \mathrm{d}x\mathrm{d}y\mathrm{d}z. \tag{3.1}$$

The watermarking system alters 0-order moments (the volume) of different mesh patches. The mesh is first normalized with the previously described PCA procedure, but all the quantities involved are moment-based. The surface is then partitioned into patches. A series of patches is altered to embed the payload in their volume. The embedding function is Rational Dither Modulation (RDM): the quantization step of the nth patch volume depends on its volume and the volume of the patch $(n-1)$th. The fusion function has two steps. The first one is an iterative alteration procedure extending the histogram mapping function to continuous quantities, with an integrated perceptual shaping component. The second step addresses the causality issue by compensating for some of the embedding alterations using a series of unaltered patches, thereby ensuring the synchronization with the decoder.

Thanks to the integral quantities, this scheme exhibits a large robustness against valumetric attacks. Scale-invariance is achieved through the RDM. Finally, the authors show in a thorough benchmark that their synchronization mechanism is a very robust instantiation of the PCA-based normalization, which provides the rigid transform invariance.

Nonetheless, some issues remain regarding the synchronization. The partitioning is arbitrary and similar to e.g. the one in the prong-based system; but the payload is not repeated and the exact sequence of patches has to be recovered (especially since some of them do not carry watermark information). The synchronization problem hence becomes more critical than in the prong-based system. The setting of the quantization step and the embedding procedure are also quite complex and time-consuming, because it relies on a trial and error procedure combined with a binary search.

Fidelity Issues in Radial-distance-based 3D Watermarking

To improve the fidelity of the baseline, a thread of research proposed to add a post-processing optimization to minimize the embedding distortion [Luo06, LB13]. Once the target radial distances ρ_i^{w} have been determined in the baseline algorithm, there remains two independent and unexploited degrees of freedom for each vertex: the spherical coordinates (θ_i, ϕ_i). Instead of dismissing these degrees of freedom and apply the straightforward fusion function (the relocations along the radial directions), a solver can act upon them to minimize some carefully chosen distortion metric ω, under the constraint of preserving ρ_i^{w}.

Various metrics for ω have been investigated, based on e.g. the Mean Square Error (MSE) or the Quadric Error Metric (QEM). As emphasized in this work, the approach corresponds to a generic refinement post-process that enhances fidelity. It can thus be advantageously integrated into any watermarking system that is solely altering the vertex norms. Nevertheless, the two-stage strategy operates a non-joint optimization and there might be alternate solutions that yield better results.

Another research to increase the fidelity of the alteration of the vertex norms has explored the use of some local surface properties to leverage on the masking effect [DHM10]. The watermark carriers are the individual vertex norms ρ_i. The embedding function is based on the Spread Transform Dither Modulation (STDM) algorithm, and consists of: (i) partitioning the carriers into sequences, each one carrying a single bit; (ii) projecting the sequences onto random vectors, whose values have been perceptually modulated; and (iii) performing a QIM embedding. The perceptual modulation associated to ρ_i, controls the amount of quantization distortion, depends on both the roughness and the curvature at v_i.

On the one hand, this approach increases the security of the watermark, as the random vectors are secret. It also provides a generic approach to perceptual shaping, as the modulation may depend on other metrics than the proposed ones. On the other hand, this system is related to the ones based on local geometric properties (Section 3.2.1) and its sensitivity to synchronization attacks on the vertices is very high.

Still, since there is no mesh traversal procedure, and, unlike typical local descriptor, the computation support for ρ_i is essentially discrete, this approach is more relevant to the development of watermarking based on the distribution of the vertex norms. In Chapter 6, we present a watermarking system that alters the distribution of the vertex norms and leverage on the Spread Transform (ST) to minimize the embedding distortion and improve the watermarking security.

Table 3.1 provides an eagle-eye view of the state-of-the-art for distribution of Euclidean distances-based watermarking by summarizing the strengths and weaknesses of the different methods.

Algorithms	[CPJ07]	[LB13]	[HRAM09]	[WLDB11]	[RAMC07]
Integral quantities	-	+	-	++	+
Registration needed	no	no	no	yes	no
Imperceptibility	-	++	+	++	-
Causality addressed	-	-	++	+	-
Robustness vs. Cropping	-	-	-	-	+

Table 3.1: Strengths and weaknesses of the algorithms watermarking the distribution of the norms of the vertices.

3.2.3 Hybrid Systems

Some authors have proposed integrating the distribution-based research into a hybrid system in which the carriers are based on local geometric properties [YI10].

Instead of the vertex norms, the payload is embedded in the distribution of the magnitude of the Laplacian coordinate vectors. These vectors are computed with the product $\mathbf{P}_{\text{Laplacian}} = \mathbf{PL}$, where $\mathbf{L}$ is the combinatorial Laplacian (see Section 3.3). Intuitively, the Laplacian coordinate vectors are 3D 'detail' vectors that correspond to the difference between the position $\mathbf{p}_i$ of v_i, and the average vertex position in $\mathcal{N}_1(v_i)$.

The carrier is defined as the difference between the number of samples that fall within, e.g. bin i and $i+1$, in the histogram of the Laplacian coordinate norms. To switch bin, the target magnitude of a sample is set to the average of its target bin. The fusion function is a twofold inverse mapping: (i) the target magnitudes are mapped back to the 3D vectors $\mathbf{P}^{\text{w}}_{\text{Laplacian}}$ through radial-only relocation, then (ii) the actual 3D vertex positions are built with $\mathbf{P}^{\text{w}} = \mathbf{P}^{\text{w}}_{\text{Laplacian}}\mathbf{L}^{-1}$.

The key property of this approach is its robustness against pose attacks. Since the Laplacian coordinates norms are distances between neighbor vertices, they share similar properties with the intrinsic geodesic distances, namely their robustness against pose. Still, because of the connectivity-only definition for the Laplacian matrix, the robustness of this system against e.g. simplification (not reported) may be limited.

3.2.4 Distribution of Geodesic Distances

One major shortcoming of the vertex norm[5] is its instability against pose operations. To create pose-invariant watermarking systems, intrinsic quantities on the surface may be used, such as the geodesic distances. Recent work has then focused on adapting the systems based on Euclidean distances to deal with geodesic distances.

In a recent geodesic-based watermarking system, the mesh is first partitioned to repeatedly embed the payload [TLHK10]. This partitioning is a complex procedure that identifies pose-invariant mesh regions [KLT05]. Inside a region, vertices are further partitioned into n_b cells, according to their geodesic distance to the nearest boundary of the region. The carrier is the average distance inside a cell. The payload embedding is Spread Spectrum (SS)-based, and modulates the carrier with regard to some threshold. The fusion function iteratively relocates a single vertex alongside one edge of its 1-ring neighborhood, so as to increase/decrease its distance to the boundary. This system requires a very large number of pairwise geodesic distance computations and has a large computational complexity. In practice, it can only handle very small meshes.

Another watermarking system using geodesic distances [LB11] proposed to remove the partitioning and to strictly follow the fruitful baseline approach of Cho et al. [CPJ07] instead. The geodesic distances are computed with regard to a single predefined surface point, which alleviates the computational issue of estimating a myriad of pairwise distances between all vertices. The watermark carriers are the averages in the bins of the histogram of the geodesic distances between the vertices and this reference point. The embedding function still modulates the average with regard to a predefined threshold using SS approach. But the fusion is a procedure carefully designed to efficiently alter the geodesic distances approximated with the Fast Marching algorithm [PC06].

Although using geodesic distances may provide an elegant solution to achieve pose invariance, these distances have major shortcomings.

First, the computational cost is much greater than for Euclidean distances. Computing even a small subset of all pairwise geodesic distances is labor-intensive for medium-sized meshes, and almost impossible for large meshes (containing more than a few hundred thousand vertices) using consumer-grade hardware. Second, their robustness against valumetric attacks is poor (see Chapter 4). Third, they also suffer from a well-known sensitivity to topological alterations. For instance, small holes and gaps may have large effects on geodesic paths. Improving the robustness and the efficiency of the approximation of geodesic distances is still an open issue [CWW13].

For watermarking purposes, since geodesic distances are defined between pairs of points on the surface, a last issue involves the synchronization. Finding a reference canonical primitive at the embedding and decoding to compute the geodesic distances is challenging. For instance, in the second watermarking system, the computation of this reference point still uses a PCA-based normalization, which is not pose-invariant.

[5] In a more general way, for any Euclidean distance between points that are far apart on the mesh surface.

3.2.5 Distribution of Normals

A last series of approaches for 3D Watermarking in the spatial domain is relying upon the distribution of normals rather than vertex locations. In a seminal algorithm [Ben99], the normals of the facets are mapped onto the unit sphere and assigned a weight equal to their surface area. The watermark carriers are the averages of the normal vector[6] within clusters, defined through a discretization of the unit sphere. An improved version of this work proposed to weight the normals, so as to account for the spatial proximity of the facets, which was overlooked [LK07].

The main drawback of these systems is that the carrier is indirectly related to the vertex positions, which greatly increases the complexity of the fusion functions. Additionally, using the normal orientation always requires a normalization step to re-orient the mesh so that both the decoder and the embedder are synchronized.

3.3 Transform-domain 3D Watermarking

Following common watermarking approaches in 1D and 2D, transform-domain 3D watermarking uses spectral analysis to define the embedding domain. This strategy aims at benefiting from well-known properties of the spectrum of a signal; for instance, its alteration is readily propagated onto the mesh in a global manner. The two main instantiations of the Laplacian-matrix based spectral analysis, summarized in Section 2.1.2, have enjoyed the most popularity and laid the foundation for the systems reviewed in Sections 3.3.1 and 3.3.2. Very few attempts have also been made to define or employ alternate spectral analysis tools, as depicted in Section 3.3.3.

3.3.1 Laplacian-based Spectral Coefficients

The seminal work of Ohbuchi et al. [OMT02] introduces a non-blind 3D watermarking system where the payload is embedded in the mesh spectrum, computed with the combinatorial Laplacian of the mesh. More specifically, the watermark carriers are the individual components of the spectrum 3D coefficients (X_k, Y_k, Z_k), associated with the low and medium mesh frequencies. The payload is embedded in the carrier using SS. With the combinatorial Laplacian, the fusion function is exact; the inverse transform is well-defined and exactly written as in Eq. (2.2). Hence, there is no causality issue[7].

To speed up the computation of the eigenvectors of the Laplacian matrix, the authors initially partition the mesh into smaller surface patches. The payload is repeatedly inserted in each of these patches. As already observed for spatial domain approaches, this yields an effective mechanism with resilience to cropping attacks, but synchronization issues may arise during the patch registration. A much greater concern arises from the influence of the mesh connectivity on the spectral basis, which creates a large flaw against connectivity attacks. To avoid this pitfall, the system uses the *non-blind* Iterative Closest Point (ICP) registration, combined with a remeshing to recreate the original mesh connectivity on its attacked version. The partitioning and the non-blind registration have been adopted when extending the system to point cloud representations [OMT04], in which case a connectivity is first built, before plugging-in the original system.

The components of the spectral coefficients have been used as watermark carriers in a system targeting subdivision surfaces instead of meshes[8] [LDD07]. In this approach, an error correction component is added. The number of carriers is increased by allowing modifications in a wider range

[6] Contrary to distance-based systems, the carriers are not scalar but 3D vectors.
[7] This is obviously only valid for connectivity-preserving watermarking.
[8] Subdivision surfaces are defined through a coarse initial surface and a subdivision refinement rule.

of the spectrum. The SS embedding is replaced by an ad-hoc solution to adapt the magnitude of the modulation according to the spectrum coefficients. Finally, the synchronization is ensured with a complex non-blind state-of-the-art registration method.

In the first blind spectral-domain 3D watermarking system, the watermark carriers are the median of each spectral coefficient (X_k, Y_k, Z_k) [CAS+03]. The embedding modulates the median with regard to some threshold in-between the range defined by the minimum and maximum coefficient. The robustness of this solution is however very limited. To improve the robustness and reduce the computational complexity, a preliminary partitioning was introduced next [AM05]. Since this partitioning is content-driven (as opposed to the arbitrary ones of e.g. Ohbuchi et al. [OMT02]), the ICP registration is unnecessary, and a blind detection of the payload is performed. More precisely, the mesh umbilical points (points where the principal curvatures are equal) are first detected in a multi-scale fashion. Patches around these points are defined through a geodesic Delaunay triangulation. Each patch is then robustly remeshed before performing the spectral decomposition. The main robustness limitation stems from the instability of the umbilical points.

Dismissing the content-driven partitioning strategy, Luo and Bors have introduced the use of the PCA-based normalization, before performing a canonical mesh partitioning using a surface area criterion [LWBL09]. Three main changes are introduced: (i) the embedding is no longer repeated in different patches, as a single bit is rather embedded in each one; (ii) the watermarked spectral coefficients are in the medium and highest frequencies, whereas previous methods used the lower end of the frequency range; (iii) the watermark carrier is derived from the PCA of the spectral coefficients (3D vectors). In concrete terms, the ratio of the eigenvalues of the PCA is modulated. Because this modulation operates at the distribution level, instead of the individual samples, the authors report an improved robustness over previous systems.

In the context of video-games, watermarking the distribution of the magnitude of the spectral coefficients associated to the low frequencies has been investigated [TBSS13]. The main reported contribution consists in an evolutionary optimization procedure to define the watermarked magnitudes. Since the weakness against connectivity modifications is purposefully dismissed, and the eigen-decomposition needs to be performed on the whole mesh, the practical application of this approach is somewhat limited.

3.3.2 Manifold Harmonics Watermarking

Some authors have used manifold harmonics (see Section 2.1.2) instead of the combinatorial Laplacian. This discretization explicitly integrates some geometric information, which strengthens the link between the basis of the spectral domain and the surface.

In a first approach [LPG08], the watermark carrier is chosen as the magnitude of the spectral coefficients in the low frequency range, which is invariant to rigid transforms. The carrier signal is divided into frames of ten samples, and one bit is embedded by altering a single sample with regard to the frame average. Since the authors only investigate using a 5-bit payload, the practical applications are limited.

On one hand, as the basis of the spectral domain not only depends on the connectivity, but also on the geometry, the robustness against connectivity altering attacks is expected to improve. The Laplacian eigenvectors and the mesh spectrum should be more invariant to the actual sampling $\mathbf{P}$ and rather dependent on the actual 2D surface, which is informally referred to as a 'mesh-invariant' property.

On the other hand, this introduces a complex causality issue at the embedding. The payload embedding alters the geometry, effectively impacting the discretization of $\mathbf{Q}$ and $\mathbf{D}$ in Eq. (2.7).

The basis eigenvectors are changed and the decoder may be unable to access the same spectral domain as the embedder. To alleviate this issue, the payload is iteratively embedded until it can be correctly retrieved; the reported experiments suggest that this procedure converges after a few iterations.

This first approach has been extended by Wang et al. [WLBD09] with a Scalar Costa Scheme (SCS)-based embedding function. The quantization step is derived from a single spectral coefficient to achieve uniform-scaling invariance. To reduce the synchronization and causality issues, the payload is redundantly embedded in three sub-bands of the spectrum, and the altered magnitudes are separated by a minimum offset. While this system presents a large imperceptibility, its robustness and capacity are still limited compared with spatial methods, and the control over the causality issue is not yet fully understood [WLDB11].

3.3.3 Other Types of Harmonics

A small thread of research has explored alternate transforms to remedy some of the issues found in the Laplacian-based transforms. The most notable example relies upon the derivation of a new spectral analysis tool to compute the spectral coefficients [WK05]. This tool is based on a set of k radial basis functions with a wide (but still compact) support on the mesh surface[9]. Each basis function is defined through a randomly selected center vertex; its estimation at all vertex positions results in a pre-basis vector in $\mathbb{R}^{n_v}$. A Singular Value Decomposition (SVD) is applied to the matrix of pre-basis vectors to find the orthogonal basis of the spectral domain. In practice, this extraction is much faster than in the Laplacian case. This comes from the use of a small number of basis functions ($k \ll n_v$), but whose wide computation support still captures enough of the mesh geometry to efficiently represent the energy of the signal in a compact manner. In the embedding domain, each component of a spectral coefficient is modulated to embed a single payload bit.

While this approach lifts the complexity issue from the mesh spectral analysis, the decoding is necessarily non-blind, and involves a resampling procedure so as to recover the k center vertices of the basis functions. Finally, the evaluation of the radial basis functions uses Euclidean distances: this avoids some of the routine connectivity-dependency issues, but also makes for a non-intrinsic spectral domain.

Another attempt has looked into switching to spherical harmonics [KMD+09]. In short, spherical harmonics are used to decompose the 3D geometric signal onto a set of basis vectors, and the proposed system alters the spheroidal coefficients (coefficients associated with the spherical harmonics) with a multiplicative scheme that is advocated for its improved fidelity. To increase the robustness, a preliminary mesh smoothing is applied so that the watermark is not embedded in the high-frequency components of the signal, which are less robust. A canonical patch generation and surface sampling procedures ensure the synchronization of the system while limiting the computational complexity. While the authors showcase the robustness of a proposed non-blind decoding in terms of Equal Error Rate (EER), the actual Bit Error Rate (BER) results of a multi-bit embedding are not reported.

3.3.4 Discussion

Spectral transforms grant access to an embedding domain in which the mesh representation is compact (the energy of the signal is concentrated in the low frequencies), and where the type of embedding distortion (e.g. low-frequency distortions of the surface) is less perceptible [SCOT03].

[9] Most discretizations of the Laplacian can be described through basis functions with a 1 ring support, e.g. the umbrella operator for the combinatorial Laplacian.

While spectral-based watermarking exhibits high amount of robustness against valumetric attacks, they are very labor intensive; this is addressed through partitioning, which then creates synchronization problems.

The combinatorial Laplacian creates weaknesses against connectivity-altering distortions. These are fixed through non-blind remeshing procedures, hence limiting the practical applications of the watermark. A few blind remeshing procedures have been proposed, but their robustness is uncertain. Manifold harmonics can help increasing the resiliency to connectivity alterations, but their use is hampered by a causality issue that can only be addressed through limiting payload sizes (e.g. 16 bits), iterative embedding procedures, and a complex selection of only a few spectral coefficients as watermark carriers. Still, the fact that the spectrum is an intrinsic quantity (invariant to isometric deformations of the surface, such as pose) may prove to be instrumental for robust 3D watermarking of dynamic meshes. Finally, the research on alternate spectral transform (e.g. spherical harmonics) has had limited success, as trading robustness for a smaller computational cost often leads to unsatisfactory results for robust watermarking.

3.4 Multiresolution 3D Watermarking

Spectral transforms lack localization capabilities and also require a large computational power. In contrast, multiresolution analysis for meshes (see Section 3.4) provides an effective means to decompose a mesh into a coarse base surface and a series of refinement 3D details that can both be used as watermark carriers.

The first system based on the mesh wavelet decomposition proposed by Kanai et al. [KDK98] watermarks the ratio between the magnitude of a wavelet coefficient and the length of its associated edge. To improve fidelity, the embedding is limited to the magnitudes that are greater than a threshold, which only modifies rough regions. The decoding is non-blind.

In the progressive mesh representation [Hop96], the wavelet coefficients are essentially replaced by vertex split operations. Praun et al. [PHF99] presented a non-blind watermarking system in which the vertex splits with the largest magnitudes are selected as embedding locations. One notable aspect of this approach is that it uses a 3D instantiation of SS. Instead of modulating a 1D watermark carrier, the vertex positions in a compact neighborhood around the embedding locations are directly modulated with (i) a perceptually-pleasing 3D basis functions (hat, sombrero) taking the place of the spreading sequence, and (ii) a global direction of alteration, equivalent to a 3D embedding strength. The robustness reported against most routine attacks is high, but, as in the previous case, the system relies on a non-blind registration and resampling procedure that makes it not suitable for most practical applications.

Finally, using an extension of a Laplacian Pyramid scheme for the decomposition of triangle meshes [YPSZ01], another robust watermarking system has been proposed with a plain SS embedding of the payload in the vertex positions at the coarsest mesh level. The decoding is still non-blind, as the initial connectivity needs to be recovered, and the robustness against vertex re-ordering is then only partially achieved.

Subsequent research has focused on improving the wavelet-based non-blind system of Kanai et al.. With SS to modulate the magnitudes of the wavelet coefficients at a given level of resolution, a 0-bit blind watermarking system was described [UCB04]. Wang et al. have then replaced the SS with the SCS to embed multiple bits [WLDB08b] at the coarsest mesh level (base mesh). To achieve uniform-scale invariance, the quantization step depends on the average edge length in the base mesh. Thanks to the one-to-one mapping between the edges of the base mesh and the

wavelet coefficients, the edge length is explicitly employed as a synchronization signal, while the payload is only embedded in the wavelet coefficients. In concrete terms, edges are sorted according to their length to order the carrier sequence. This system outperforms most of the previous 3D watermarking methods in terms of robustness against valumetric attacks.

Taking an opposite approach, some authors have proposed to directly embed the watermark in the vertices of the base mesh instead of the wavelet coefficients [BBC+03]. The payload is embedded by modulating the radial distances of every vertex with a SS approach. This system is blind and its robustness stems from the resilience of the base mesh.

Because of the limitation of the original wavelet transform, all the previous systems are restricted to the semi-regular cases. To lift this restriction, a thread of research has explored switching to the extension of wavelets to irregular meshes [VP04]. Authors have presented a blind watermark algorithm that alters the magnitude of the wavelet coefficients through their histogram [KVJP05]. The embedding function is a straightforward SS instantiation. Since the irregular wavelet decomposition depends on the choice of an initial mesh facet, a connectivity-dependent synchronization step to systematically select the same location is needed. The robustness against connectivity-altering distortions, not reported in the publication, should thus be particularly small.

Dropping the histogram computation, and embedding the payload in the magnitude of individual coefficients through the SCS, authors have reported a better control over the distortion [PHOZ12]. But in the absence of any synchronization strategy, the system fails even against a vertex reordering[10].

Another strategy to allow for multiresolution watermarking of irregular meshes is to perform an initial resampling of the input mesh. This idea was first advocated for in a non-blind system based on preliminary spherical parameterization and spherical wavelet decomposition [JDBP04]. Because of (i) the sensitivity of the parameterization and resampling, and (ii) the limited robustness of the carrier, i.e. the vertices at a specific level of decomposition, the overall robustness is limited, as the watermark is not invariant to e.g. rigid transforms.

For mesh compression tasks, wavelet transforms have been met with a growing success, whereas the popularity of spectral-based compressions has diminished, in part because they lack the ability to adapt the level of details. A recent thread of research has then faced the challenge of joint watermarking and progressive compression.

Since there is still no compression standard, the reported approaches usually involve some state-of-the-art progressive compression technique, combined with an appropriate modification of one of the previously described watermarking methods [EsRT+13, BOZHP13]. Finally, in the multiresolution domain, a reversible 3D watermark for content authentication has been presented [LDLD11]. It is based on the method of Cho et al. [CPJ07] in the spatial domain, and it achieves a very large robustness against valumetric attacks. But the decoding is only semi-blind, as e.g. the position of the center of mass needs to be transmitted.

3.4.1 Discussion

Multiresolution methods have a moderately low computation overhead, especially with regard to spectral methods. Embedding the payload in a coarse mesh version provides a large robustness against valumetric attacks. It also seems to be a promising avenue of research for progressive mesh watermarking, where the payload has to be recoverable at any level of detail.

[10] Overlooking the synchronization problem while using individual vertices is an issue already found in spatial-based watermarking, with e.g. the proposed STDM-based approach [DHM10].

Nonetheless, whereas spectral domain watermarking may leverage the intrinsic discretization of the Laplacian (e.g. manifold harmonics) to resist remeshing, there is still no solution against connectivity attacks. Lifting the greatly limiting 4-1 subdivision constraint in the decomposition tools has also been actively investigated. But research in this domain, e.g. for compression purposes, often uses a preliminary remeshing, which may be inappropriate in the watermarking context. Hence, it is yet unclear how to define a flexible enough multiresolution 3D watermarking that will still achieve the same level of robustness as the state-of-the-art systems in the other domains.

3.5 Conclusion

In this review, summarized in Figure 3-1, the key benefits and drawbacks of the three main approaches to 3D watermarking have been illustrated with some state-of-the-art examples. In spatial approaches, a locally-defined carrier will resist cropping or pose attacks, but fail against valumetric ones; in any case, the synchronization mechanism is likely to be a complex issue. On the other hand, altering the distribution of geometric quantities often leads to a large robustness overall, but the control over the fidelity is reduced, because local adaptations of the watermarking mechanism are less easy. Moreover, cropping and pose attacks usually cannot be handled.

Spectral approaches are inherently global, i.e. the payload is automatically spread throughout the spatial domain. They have interesting properties in terms of robustness and distortions, thanks to the efficiency of the spectral representation. However, computational and causality issues arise when looking into connectivity-oblivious watermarking. Multiresolution approaches address both problems, but are mostly limited to semi-regular meshes and connectivity-preserving alterations.

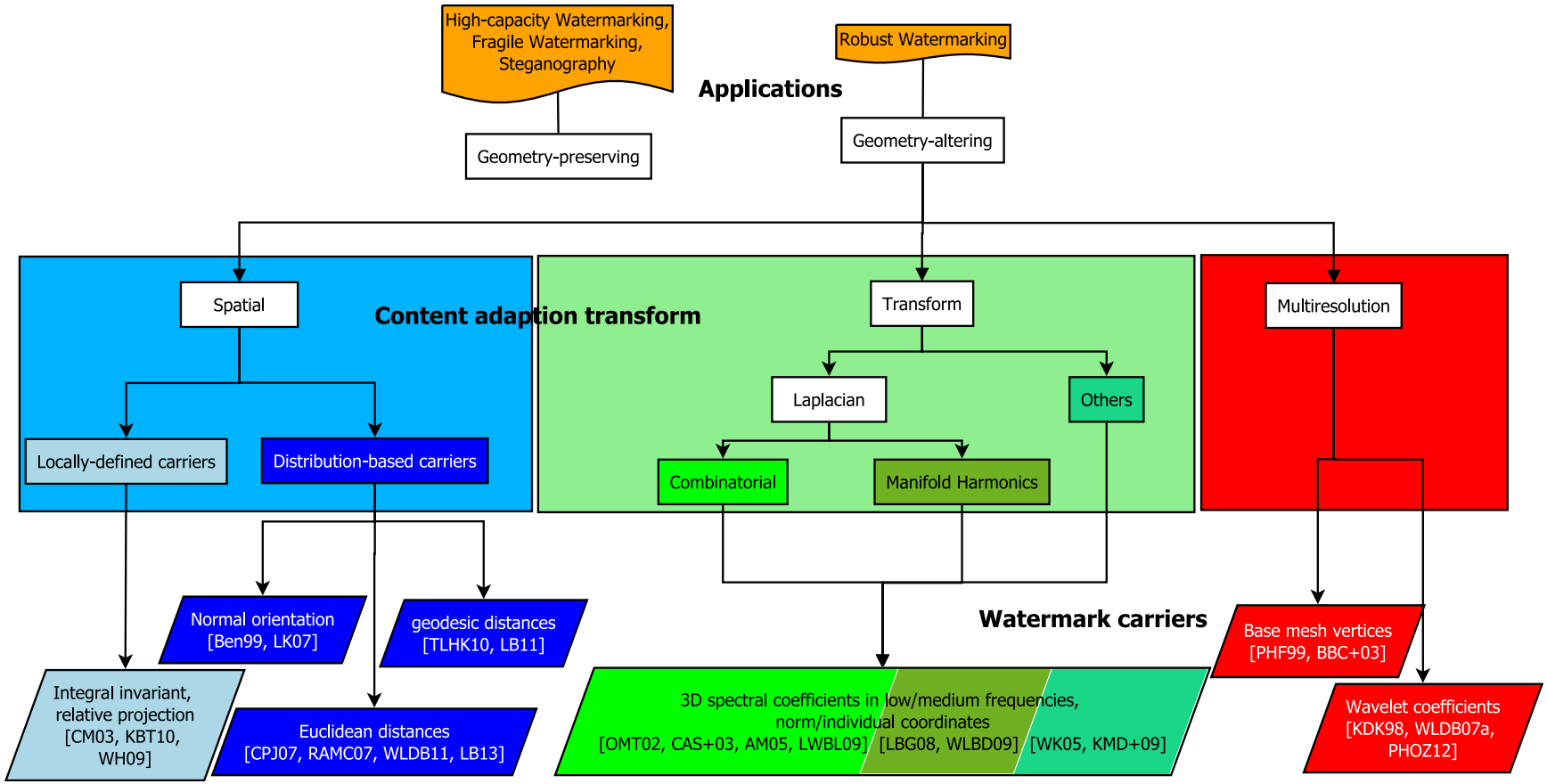


Figure 3-1: Summary of the state-of-the-art for 3D Watermarking

Chapter 4

Evaluation of 3D Watermarking Systems

4.1 Introduction

The performance of a watermarking system is commonly measured with three parameters. Its fidelity is assessed with one of the metrics presented in Section 2.3. Its capacity is given by the payload size n_b in bits. Its robustness against an attack is expressed with the Bit Error Rate (BER). The relation between these parameters can be depicted as a mapping from $\mathbb{R} \times \mathbb{N}$ to $\mathbb{R}$, where the two inputs are the imperceptibility and the capacity, and the output is the robustness. Comparing the performance of watermarking systems amounts to comparing their mapping, which is a complex task, especially when different attacks with various strengths are considered.

A popular benchmarking solution consists in selecting target levels of imperceptibility and capacity, and comparing the robustness. In practice, random payloads of $n_b = 32$ or $n_b = 64$ bits are embedded in multiple contents. The embedding strength is adjusted so that the fidelity reaches a target value. A wide range of attacks are then applied on the watermarked contents, before measuring the resulting BER. This protocol is equivalent to fixing the two inputs (ω, n_b), where ω denotes the distortion, and comparing $\mathrm{BER}(\omega, n_b)$ for different watermarking systems. This approach compares a single operating point $(c, n_b, \mathrm{BER}(c, n_b))$ between systems and can yield biased results. For instance, the results of the comparison may change when selecting a different target level of imperceptibility or a larger payload size.

An additional difficulty in the 3D case is that there is no universally accepted perceptually-correlated distortion metric; each behaves differently, depending on the type of distortions. For instance, while the Root Mean Square (RMS) is sensitive to alterations that preserve the tangent plane to the surface, the Quadric Error Metric (QEM) is not. Since the type of embedding distortion depends on the watermarking systems, e.g. ring-like alterations, high-frequency or low-frequency ripples on the surface..., selecting a metric introduces a bias in the benchmark.

This issue may be partially addressed by setting target upper-bounds for multiple metrics, in an attempt to limit the magnitude of the embedding distortions from different points of view. A 3D watermark benchmark protocol has for example been presented, in which the embedding distortion is calibrated through the Maximum Root Mean Square (MRMS) (objective geometric distortion) and the Mesh Structural Distortion Measure (MSDM) (more perceptually-correlated distortion) [WLD+10]. This approach creates a complex experimental setup, and setting target distortion values is cumbersome. Closed-form expressions with regard to the embedding strength cannot be found in most cases, and some manual tuning must be performed so that the distortion

stays within an acceptable range of the target value.

Finally, an important drawback of this protocol is that the BER aggregates information on the performance of all the layers in the watermarking system: the resynchronization scheme, the entire content adaptation layer, the watermark embedding strategy, and the effectiveness of the inverse mapping. For each layer, a variety of instances has been proposed in the literature (see Chapter 3). Since watermarking algorithms often differ from one to another in multiple aspects, a BER-based comparison is not always meaningful and does not clearly shed light on the specific aspects that make a system more robust than another.

This chapter presents the evaluation of 3D watermarking algorithms by focusing solely on the performance of the content adaptation transform itself. This addresses the previously underlined issues, as it avoids: (i) making a limited comparison based on a single operating point, (ii) finding an unbiased distortion metric, and (iii) aggregating the performance of all the layers in the system. This approach is thus a fruitful way to devise new research directions for 3D watermarking.

First, the generic benchmarking protocol is presented in Section 4.2. Section 4.3 deals with the robustness of various quantities and estimators on which watermark carriers are built. A comparative analysis of the performance of the most relevant carriers is given in Section 4.3.7. Conclusions from these results are drawn in Section 4.4

4.2 Experimental Setup

The following experimental setup forms a basis for the remainder of this dissertation. Given a (piece of a) content adaptation transform, its main steps are: (i) applying the transform on a database of $n_{\mathcal{M}} = 13$ meshes (see Table D.1), (ii) for each mesh in the database, generate a series of attacked versions (see Section 4.2.1), (iii) apply the transform on all the distorted versions and compare the results to the original ones (see Section 4.2.2).

The benchmark is implemented with MATLAB; a few algorithms (indicated below) are implemented in C++ with the CGAL library [CGA]. Multi-threading is used in e.g. sparse matrix decomposition. All experiments are carried on a quad-core PC clocked at 4.2 GHz with 16 GB of RAM.

4.2.1 Attacks

From the various possible 3D attacks reported in Section 2.2.2, we select the ones that are easily automatized and review their practical instantiation. Note that for every level of non-deterministic attacks, $n_T = 4$ versions of a distorted mesh are generated.

Content-preserving Attacks

Rigid transforms are implemented with a random rotation (three random Euler angles) and a random translation of the mesh. An *isotropic scaling* that applies a random scale factor to $\mathbf{P}$ is tested. Finally, a *vertex reordering* attack is implemented.

Many metrics, e.g. the MRMS, attribute a non-null distortion to some of these attacks. Nevertheless, they are considered to be perceptually content-preserving, as the intrinsic properties are preserved (or at most uniformly scaled). Hence, any robust watermarking scheme should be completely invariant to all these alterations. Indeed, even fragile watermarking systems are designed to be invariant to these attacks.

Connectivity-preserving Attacks

Noise addition modifies the vertex positions according to some random vectors in $\mathbb{R}^3$. In our implementation, each vertex is moved in a direction whose individual components (x, y, z) are uniformly drawn in $[-0.5, 0.5]$, and with a magnitude uniformly distributed in $[-dr, dr]$. d denotes half the space diagonal of the mesh bounding box, and r is the noise strength. This ensures that similar distortions are applied to differently scaled versions of the same models.

Authors have used many alternatives to generate the magnitude and the direction of the random noise: Normal distribution, the mesh bounding sphere instead of the bounding box, etc. In Chapter 5, we test the so-called 'normal noise' (i.e. the tangential component of the random direction is null), that depends on the input signal, whereas the noise addition is here *independent* from the mesh.

A *smoothing* operation filters out the high frequencies components of a signal. This attack is implemented by applying multiple iterations of a Laplacian-based smoothing to the input mesh. At every iteration, the position $\mathbf{p}_i$ is updated to $\mathbf{p}'_i$ with:

$$\mathbf{p}'_i = (1-\lambda)\mathbf{p}_i + \lambda \frac{1}{|\mathcal{N}_1(v_i)|} \sum_{k \in \mathcal{N}_1(v_i)} \mathbf{p}_k. \tag{4.1}$$

The deformation factor λ is set to 0.3. This smoothing operator exhibits shrinkage effects, but its complexity and storage space are small [VMM99].

As there is no standard method for mesh compression, one of the most straightforward approach consists in *quantizing* the coordinates of the vertex positions on a three dimensional lattice [AG05]. Denoting by b_d the number of quantization bits, the step of the lattice along, e.g. the x-axis, is:

$$s_x = 2^{-b_d} \left(\max_i (\mathbf{p}_i \cdot \mathbf{u}_x) - \min_i \mathbf{p}_i \cdot \mathbf{u}_x \right). \tag{4.2}$$

More evolved compression methods, providing a better bitrate vs distortion trade-off, apply quantization on prediction error vectors in e.g. the spectral domain [SCOT03], and present a higher complexity. Furthermore, researchers have recently rather focused on *progressive compression*, that turns a mesh into a series of increasingly detailed meshes thanks to refinement operations, starting from a coarse approximation (see e.g. the wavelet transform for meshes in Section 2.1.2). Thus, compression attacks may be simulated by looking at some simplification and refinement procedures, which are not connectivity-preserving.

The three aforementioned attacks may create degeneracies and issues. For large noise addition, the mesh surface usually exhibits self-intersections and is no longer manifold. With quantization, and because of the shrinkage effect with the Laplacian smoothing, facets may become degenerate (triangle with null area) and vertices may be superimposed due to the limited precision of the Object File Format (OFF) representation and the quantization. These issues greatly impact some of the benchmarked content adaptation transforms, resulting in errors that prevent assessing their stability. They are therefore specifically recorded.

While the previous distortions correspond to valumetric attacks in the spatial domain, *pose* attacks are considered to be desynchronizing. They do not modify the connectivity but create nearly isometric distortions of the surface mesh. In the benchmark, 49 poses of an elephant mesh are used; the ground-truth mapping between the vertices in different poses being known.

Connectivity-altering Alterations

These attacks are challenging 3D watermarking, as they all correspond to synchronization attacks.

Cropping deletes a part of the mesh. In this benchmark, vertices are sorted according to some random direction before being deleted. This creates non-watertight meshes with boundaries.

Mesh *simplification* is a routine operation in the 3D modeling pipeline. Highly detailed meshes often need to be simplified to reach a level of details suitable for computer hardware with limited capabilities. This attack is implemented (with CGAL) by generating increasingly simplified versions of the original mesh, with their number of edges set to a ratio of the initial one [LT98]. To increase the complexity of a mesh, the *Loop subdivision* scheme [WW01] (with CGAL) is applied. For complexity reasons, a single iteration of the subdivision is performed.

Last, a *triangle soup* is generated from an input mesh by: (i) disconnecting all facets; (ii) scaling all edges lengths according to a specified ratio (thereby creating holes or overlaps); and (iii) adding a uniform random noise to all vertices. In this benchmark, the attack strength is only parameterized by the ratio used to alter the edge lengths. The strength of the uniform noise is set to 0.1% with respect to (half) the space diagonal of the bounding box.

4.2.2 Stability Metrics

The robustness of an extraction function against content-preserving alterations is in essence a binary question, whose answer is, in most cases, in theory demonstrable. Many content adaptation transforms are not robust against scaling, and watermarking systems often depend on either the resynchronization mechanism (through a bounding sphere/box normalization), or on the embedding function (using e.g. Rational Dither Modulation (RDM)). As the performance of these components is out-of-scope, only the theoretical robustness of the transform itself is reported, when available. In the few cases where there is no definite answer, experimental results are reported instead.

The benchmarked functions map an input mesh $\mathcal{M}$ to a quantity $\mathbf{c}$, which is hereafter referred to as a watermark carrier signal for simplicity[1]. Unless mentioned otherwise, $\mathbf{c}$ is assumed to be a vector. Given an altered mesh $\mathcal{M}'$ and the associated carrier $\mathbf{c}'$, the robustness of these functions can be measured with the stability, i.e. the variations between $\mathbf{c}$ and $\mathbf{c}'$. Let $\mathbf{e}$ be the vector of relative errors between both signals. Its elements are:

$$e_i = \left| \frac{c_i - c'_i}{c_i} \right| . \tag{4.3}$$

In the majority of spatial domain-based extraction, the carrier is defined at the vertex level, and $|\mathbf{c}| = |\mathbf{e}| = n_v$. Both the carrier and the error signals are scalar fields defined over the vertex positions $\mathbf{P}$. For connectivity-preserving attacks, Eq. (4.3) is applied at each vertex position. In case of a cropping attack, the estimation of $\mathbf{e}$ is restricted to the remaining vertices in the cropped mesh, as the mapping for these vertices is trivial, and: $|\mathbf{e}| = n'_v < n_v$. With the triangle soup, the subdivision and the simplification attacks, the estimation of $\mathbf{e}$ is performed for all the vertices in $\mathcal{M}'$ ($n'_v > n_v$). In the latter two cases, c_i in Eq. (4.3) is replaced with the carrier associated to the nearest vertex position in the original mesh: $\underset{k \in [\![1, n_v]\!]}{\arg\min} \|\mathbf{p}'_i \mathbf{p}_k\|$. For the triangle soup, the mapping between the altered vertices and the original ones is trivial and there is no need for a neighborhood search.

Carriers that do not induce a scalar field over the mesh vertices cannot be benchmarked with this protocol. This specific issue is dealt with in Section 4.3.6, that investigates the stability of the

[1] More specifically, a variety of watermark carriers are derived from the extracted quantity.

spectrum-based carriers.

To aggregate the individual results in $\mathbf{e}$, the instability for a given mesh is defined by the median of $\mathbf{e}$. For deterministic attacks, the reported instability at a given level of attack is the median of the instability at mesh levels (median over the 13 meshes). For the attacks in which $n_T = 4$ altered mesh versions are generated, the results at mesh level are averaged over the n_T trials, before computing the median instability.

4.3 Stability Results

This first round of experiments investigates the stability of various functions against fixed levels of attacks, listed in Table 4.1. The results are depicted with radar plots, where the sensitivity against an attack corresponds to the median instability over the different trials and meshes, as detailed above. In this representation, the scale is logarithmic and sensitivity values correspond to $\log_{10}(1+s)$, where s is the original value in percent. $s = 100\%$ is pictured with a black circle. The objective is (i) to provide an overview of the strengths and weaknesses of geometric primitives upon which watermark carriers are built, and (ii) to measure the influence of some key settings in the extraction, e.g., the type of estimator used to approximate a target quantity.

Attacks	Parameter values
Uniform Noise Addition	1% amplitude
Smoothing	20 iterations
Quantization	9 bits
Cropping	21% removal
Triangle Soup	40% length ratio
Simplification	90% edge removal
Subdivision	1 iteration

Table 4.1: Level of attack to assess the stability of some extraction functions.

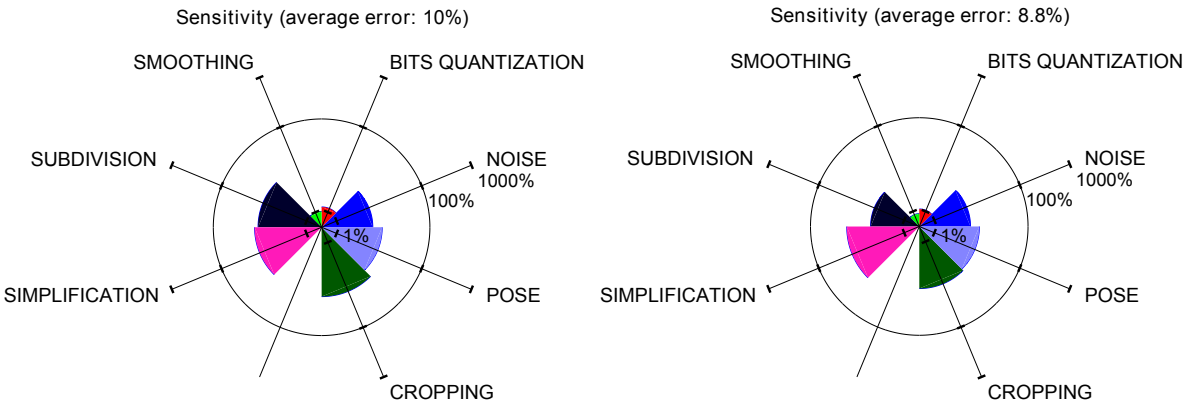
4.3.1 Surface Area Stability

To start this study, we look at the stability of a local surface patch around every point $\mathrm{a}(\mathbf{p}, \tau)$, defined as:

$$\mathrm{a}(\mathbf{p}, \tau) = |\mathcal{M} \cap \mathcal{S}(\mathbf{p}, \tau)|, \tag{4.4}$$

where $\tau > 0$ denotes the radius of the sphere $\mathcal{S}$ centered at $\mathbf{p}$. τ is the scale of the estimation. $|.|$ denotes the surface area. $\mathrm{a}(\mathbf{p}, \tau)$ is computed through CGAL, and computed on a triangle mesh through linear interpolation. τ is set as a ratio of the average distance between the center of mass and the vertices. Note that this function is different from the area invariant [PWHY09], which computes $|\mathcal{D} \cap \mathcal{S}(\mathbf{p}, \tau)|$, where $\mathcal{D}$ is the *inside domain* induced by the mesh, but which is much more complex to compute. Figure 4-1 depicts the aggregated performance results for the surface area stability.

The patch area is robust to rigid transforms and vertex reordering, but it is not stable in case of uniform scaling. For the triangle soup attack, which makes the estimation meaningless, the variations are not reported (blank quadrant in the lower left of the radar diagrams). From the two test settings $\tau = 2\%$ and $\tau = 3\%$, three conclusions are drawn. First, increasing the scale marginally increases the stability against the valumetric attacks. Second, it effectively improves



(a) $\tau = 2\%$ (b) $\tau = 3\%$

Figure 4-1: Aggregated stability of the local surface area for two neighborhood sizes.

the results against the subdivision operation, but it has the opposite effect on the simplification. Third, the stability against pose and cropping is low, and increasing τ yields a performance drop of about 6% against cropping.

This first observation is expected, as the area shows a significant sensitivity to high frequency perturbations of the surface, but the two others are counter-intuitive. Since τ is based on radial distances to the center of mass computed with an integral formulation, the simplification and subdivision do not change τ (see Section 4.3.2). For cropping and pose however, the radial distances and τ are modified. This illustrates a common pitfall in geometry processing: while some quantity may be e.g. pose invariant, it is still an open issue to select a consistent scale over all distortions. This clarifies the unexpected weakness against cropping and pose, but the results against simplification and subdivision are not yet fully understood.

4.3.2 Radial Distances

The Euclidean distance to the mesh center of mass $\mathbf{g}$ is a widely popular basis to design watermark carriers in the spatial domain (see Section 3.2.2). For compactness, this benchmark only deals with three variations in the definition of $\mathbf{g}$. First, $\mathbf{g}$ is taken as the average vertex position. Second, it is set as a surface-weighted average through:

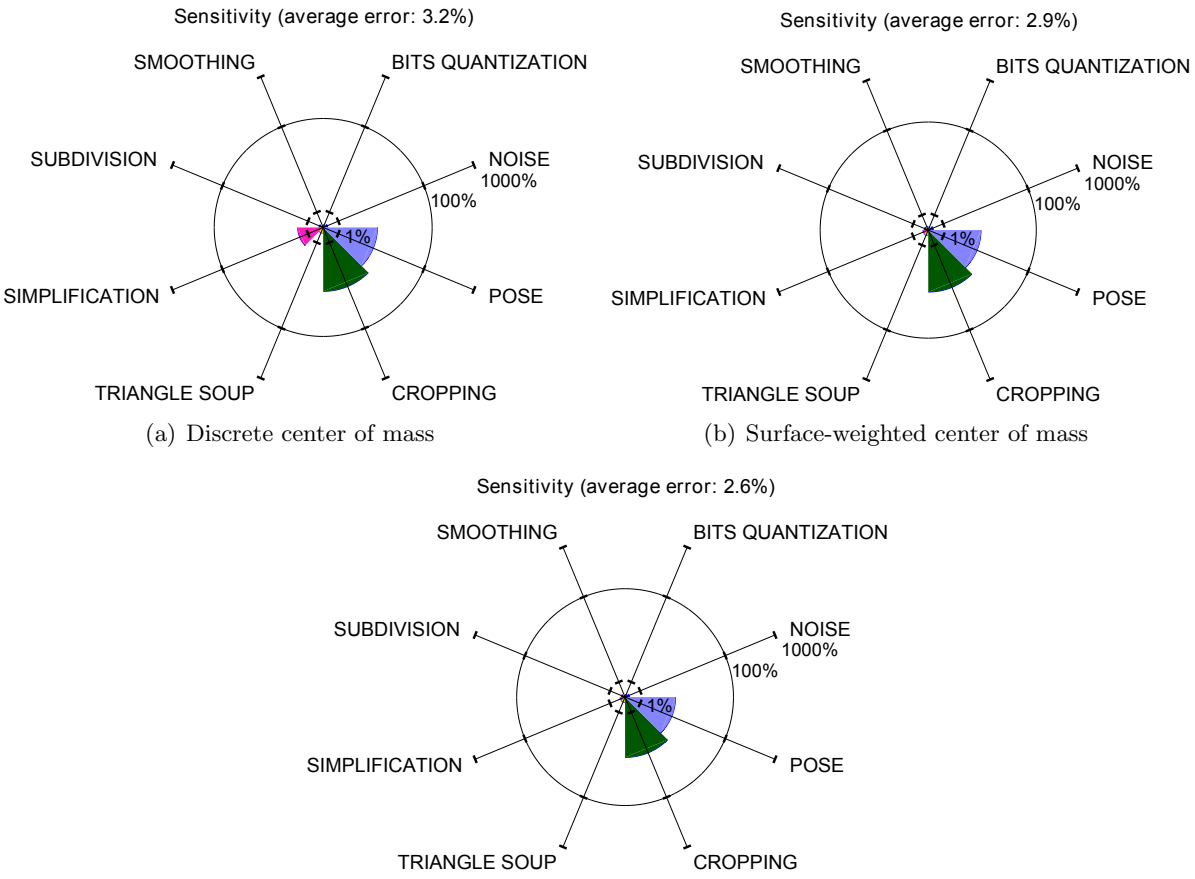
$$\mathbf{g} = \frac{1}{A} \sum_{f \in \mathcal{F}} \mathrm{a}(f)\mathbf{g}(f), \tag{4.5}$$

where $\mathbf{g}(f)$ is the center of mass of the triangle facet f and $\mathrm{a}(f)$ is its area. A is the total surface area of the mesh. Third, $\mathbf{g}$ is computed as a volume-weighted average [ZC01] with:

$$\mathbf{g} = \frac{1}{V} \sum_{f \in \mathcal{F}} \mathrm{v}(f, \mathbf{g})\mathbf{g}(f), \tag{4.6}$$

where $\mathrm{v}(f, \mathbf{O})$ is the signed volume of the tetrahedron whose vertices are formed by $\mathbf{g}$ and the three vertices of f. V is the volume of the 3D object. The stability of all the distances $\|\mathbf{g}\mathbf{p}_i\|$ is then assessed, and the results are depicted in Figure 4-2.

Distances to the discrete and surface-weighted barycenter are provably invariant to rigid trans-



(a) Discrete center of mass

(b) Surface-weighted center of mass

(c) Volume-weighted center of mass

Figure 4-2: Aggregated stability of the Euclidean distances to the center of mass for the discrete 4-2(a), surface-weighted 4-2(b), and volume-weighted 4-2(c) formulation to compute the center of mass.

forms, and these values are unaffected by vertex reordering. This is also correct when the volume-weighted barycenter is well-defined, i.e., when the watermarked mesh is watertight and without self intersections. In real-life however, 3D meshes, including the ones in our database, exhibit self-intersections and degeneracies. Experimentally, these degeneracies have no impact on the stability[2]. Similarly, distances depend linearly on the scaling factor, and in the case of the volume-weighted center of mass, this also holds true empirically.

Radial distances yield an overall very large stability against any valumetric attacks. Their only weakness is their sensitivity to cropping and pose, due to the change in the center of mass position. As expected, the discrete formulation for $\mathbf{g}$ is also unable to cope with mesh simplification; nevertheless, all the variations fare equally well against subdivision. This may be explained by the very mild distortion caused by this attack: as the subdivision is uniform over the mesh, the local vertex density is uniformly increased, and this connectivity alteration does not impact the center of mass.

Compared to the area invariant results, radial distances seem to be more stable against pose. Although Euclidean lengths are indeed inconsistent when isometric deformations of the embedded

[2]In MATLAB, the relative error is about 10^{-13}.

surface occur, real-life poses of humanoid or animal shapes usually have limited effects. For instance, in the series of meshes depicting a running elephant, $\mathbf{g}$ is overall fairly stable, and only the extremities are in motion.

4.3.3 Geodesic Distances

Geodesic distances are approximated with the Fast Marching algorithm [PC06]. Contrary to Euclidean distances, where the center of mass is a canonical geometric reference, there is no efficient and simple way to define a unique reference point on the surface. In the watermarking literature, reviewed in Section 3.2.4, a random vertex is used and the mesh is restricted to have a single connected component. This increases the burden on the synchronization mechanism. Since this issue is here out of scope, the geodesic distances are computed between a constant starting vertex and all the other vertices; their aggregated sensitivity is depicted on Figure 4-3.

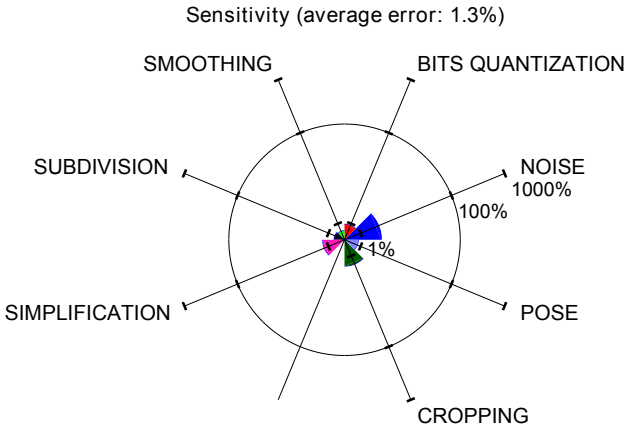


Figure 4-3: Aggregated stability of the geodesic distances from a single surface vertex to all the other vertices

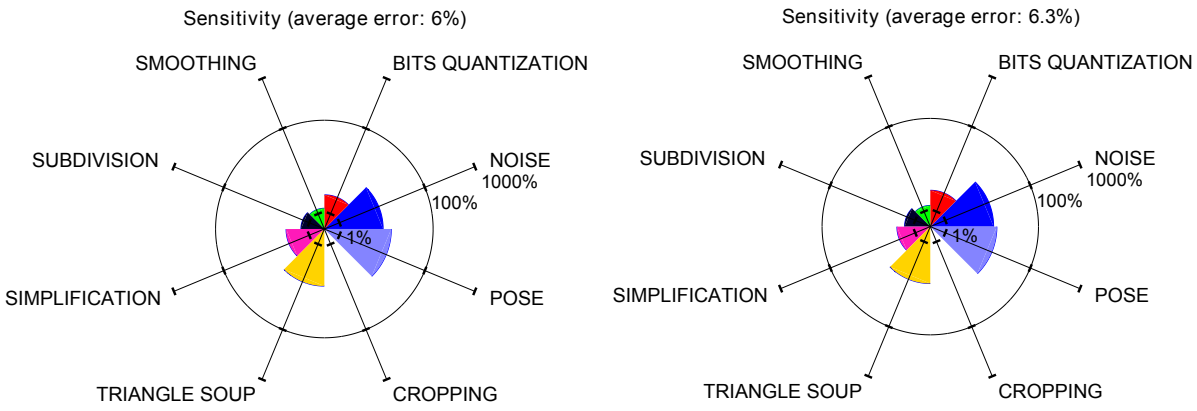
Geodesic distances are invariant to rigid transforms and vertex reordering, but not against uniform scaling. They are ill-defined in case of triangle soups; more generally, as they require a continuous surface representation (e.g. mesh, B-splines...), geodesic distances are unsuitable in applications where the connectivity of the mesh may be removed, for instance when performing a conversion to a point-based format.

Except for noise addition, geodesic distances are moderately less stable than the Euclidean distances against valumetric attacks. For noise additions, geodesic distances are much less stable. This last observation is expected, as high frequency variations on the surface greatly impact the length of the geodesic path[3]. Results against simplification also suggest a slight instability, but this may be due to the loss of the reference vertex during the simplification procedure. Finally, geodesic distances have a large consistency over different poses, and are stable against cropping. We also observed that the estimation becomes extremely inconsistent for large cropping attacks, or for carefully designed cuts, which modify the number of connected components of the mesh.

[3]Intuitively, the variation in the length of the geodesic path corresponds to the summation of the magnitude of the high frequency displacements over the path.

4.3.4 Normal Orientation

A few watermarking systems are based on the orientation of the mesh normals (see Section 3.2.5). In this benchmark, we assess the stability of the orientation by replacing c_i in Eq. (4.3) with a two-dimensional signal $(\theta, \phi)_i$, resulting from the conversion of the normal vector $\mathbf{n}_i$ to spherical coordinates. $\mathbf{n}_i$ is estimated at v_i with two well-known procedures: (i) the routine surface-weighted average, and (ii) the more computationally expensive angle-weighted average, of the normals in the facets adjacent to v_i [TW98]. The results are presented in Figure 4-4



(a) Area-weighted average

(b) Angle-weighted average

Figure 4-4: Aggregated stability of the vertex normal estimates using the area-weighted average 4-4(a) and the angle-weighted average 4-4(b).

Normal orientations are obviously modified by rotations of the mesh, but are unaffected by uniform scaling (and vertex reordering). Both types of estimations achieve very similar performances. The sensitivity to noise, quantization and pose is large. Subdivision and simplification however seem to have a lesser impact on the normal orientation. In all, normals showcase a larger stability than area invariants, but a much lower stability than geodesics and Euclidean distances.

In the soup attack, all facets are disconnected and the vertex normals are computed as the (perturbed) facet normals. This explains the inconsistency of the estimates. To improve its stability against connectivity disrupting attacks, normal estimators for point clouds could be used [MN03]. Since normals are ubiquitous in geometry processing, some research have also investigated in theory their instability under attacks [YI12]. However, for the real-life distortions tested in this benchmark, closed-form expressions for the stability are not available.

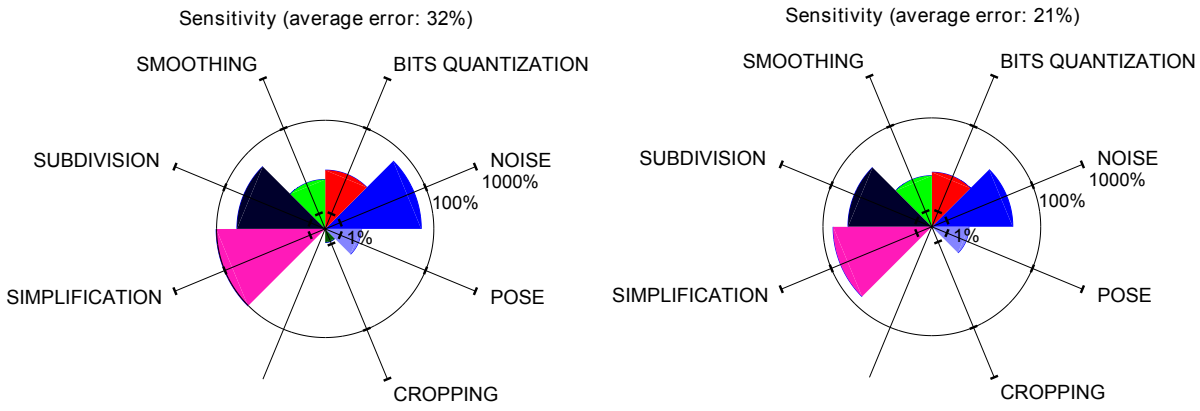
4.3.5 Principal Curvatures

Principal curvatures have not been directly used as watermark carriers, but rather as a synchronization signal [AM05]. Since a large body of research has been dedicated to their robust approximation on triangle meshes (see Section 2.1.2), the stability of the mean curvature is investigated to ascertain its usefulness in the watermarking context.

First, the normal cycle estimator [CSM03] is tested, using a MATLAB implementation [Pey11]. The basic principle of the normal cycle is to sum up a line density of tensor measured on edges of the neighborhood around a query vertex, which is here set to the 3 ring. Second, the estimation of the principal curvatures with a jet fitting [CP03] is tested (CGAL implementation [CP08]). This approach defines a local fitting basis where the mesh surface is approximated with a d-order bivariate

polynomial (the jet). It is then expressed in the Monge basis, and the coefficients correspond to the principal curvatures. d is set to 2, as higher order terms are not used, and the computation support is taken as the 3 ring neighborhood around the query vertex. While the fitting basis uses an integral formulation of the Principal Component Analysis (PCA) [GAP08], other computations are entirely sampling dependent, as they only use the individual vertex positions.

Both estimators output a series of pairs of values $(\kappa_{\min}(v), \kappa_{\max}(v))$ for all the mesh vertices, and the stability of $\kappa_{\text{mean}} = \frac{1}{2}(\kappa_{\min} + \kappa_{\max})$ is then measured. Figure 4-5 shows the aggregated stability of both estimators.



(a) Normal Cycle (b) Jet Fitting

Figure 4-5: Aggregated stability of the mean curvature, estimated with the Normal Cycle 4-5(a) and a Jet-Fitting 4-5(b), using a 3 ring neighborhood.

The mean curvature is invariant to rigid transforms and vertex reordering, but follows changes in the scaling ratio. The normal cycle estimator is only well-defined on continuous surface, and the triangle soup attack is not tested. In theory, the jet-fitting estimator could be extended to point-based representations and triangle soups. Since we do not switch to a different neighborhood definition than the ring-based one, the triangle soup is however also not tested.

While the jet fitting-based mean curvature estimate is the most stable of the two, its stability is rather small. In particular, mesh simplifications result in extremely large variations. Globally, the instability may be partially explained by the range of mean curvature values. κ_{mean} is not bounded and may be close to zero for e.g. saddle or flat points. The relative error may then reach arbitrarily large values. The only notable exceptions, where the estimators achieve a more satisfactory consistency, are the cropping attack and the pose distortion. Indeed, for quantities computed on a small support patch, changes only occur when the vertices in the 3-ring are impacted.

4.3.6 Spectral Carriers

A variety of watermark carriers are based on the combinatorial and cotangent discretizations of the Laplacian matrix (see Section 3.3). In most cases, the carrier is determined by the magnitude of the spectral coefficients in some frequency range (usually the lowest frequencies). Contrary to the previous functions, the spectral transform does not induce a scalar field in $\mathbb{R}^3$. To benchmark the stability of the magnitude-based signal, the first 50 3D spectral coefficients are computed. In Eq. (4.3), c_i is replaced by the magnitude of these coefficients, i.e.: $\sqrt{X_i^2 + Y_i^2 + Z_i^2}$ $(i \in [\![1, 50]\!])$. The procedure to further compute the aggregated sensitivity is unchanged.

The combinatorial Laplacian matrix is computed with Eq. (2.3) in Section 2.1.2. **P** is projected onto the 50 eigenvectors of **L** associated to the lowest non-null eigenvalues (the continuous component is discarded), and the stability of the magnitude of these projections is benchmarked.

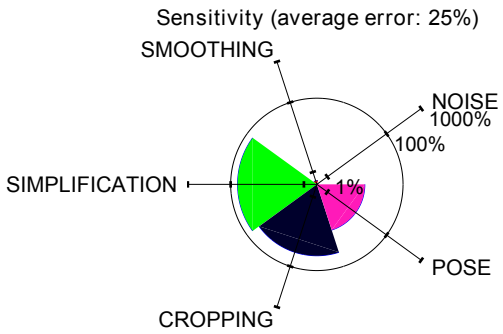


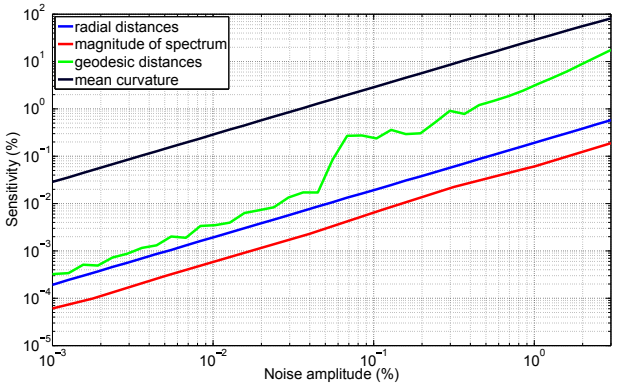
Figure 4-6: Aggregated stability of the magnitude of the spectral coefficients using a combinatorial Laplacian

The magnitude is invariant to rigid transforms and vertex reordering, but requires normalization against scaling. Because of the connectivity loss, the triangle soup attack is not tested. Moreover, the degeneracies that arise from large quantization of the coordinates of the vertex positions lead to numerical issues. Since the eigen decompositions then fails more often than not, this attack is also dropped. Most of the results reported in Figure 4-6 confirm the observations made by watermark researchers using spectral transforms: the stability against valumetric attacks such as noise and smoothing is high, but the change in connectivity due to simplification and cropping leads to large variations of the magnitude signal. Although the Laplacian matrix is unchanged by the pose attack, the magnitude of the coefficients is nonetheless impacted. A possible explanation for this unexpected result is that the ordering of the carrier values may be sensitive and change throughout the difference pose, e.g. two consecutive magnitudes switches.

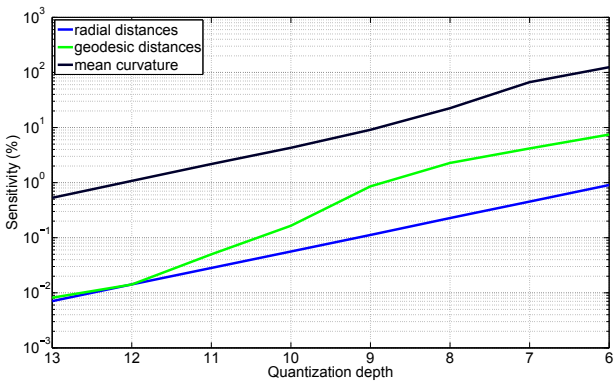
On addition to these results, the cotangent discretization was also briefly investigated. The stiffness matrix **Q** and the mass matrix **D** are built, before solving the generalized eigenproblem in Eq. (2.7) for the 50 first smallest non-null eigenvalues. Eq. (2.9) is applied to compute the spectral coefficients. However, the issues, listed below, that have been identified in the case of the combinatorial discretization become much more important for the cotangent formulation of the Laplacian.

One, the computation time is huge, especially compared with the previous functions. Solving the generalized eigenproblem with MATLAB based on a multithreaded version of LAPACK, takes on average 5 minutes for every mesh, compared with e.g. 30 seconds for the geodesics extraction. For this reason, the number of trials n_T has been reduced from 4 to 3. Two, the eigen decomposition often fails when estimating the 50 first eigenvalues. For large levels of attacks, these failures are due to the degeneracies in the mesh. As mentioned in the report on manifold harmonics [VL08], extracting the higher-end of the spectrum is much faster and reliable, but for watermarking purposes, high frequencies are unsuitable (robustness and fidelity issues) and a band-by-band computation may be needed to extract the low frequencies. Still, the extraction of the 50 first eigenvalues is usually not expected to be a challenge.

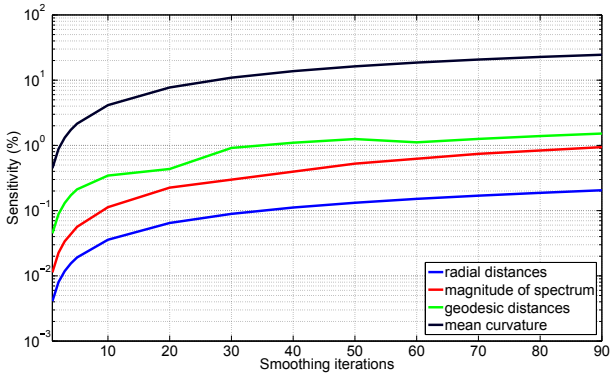
4.3.7 Evolution of the Stability against Increasing Levels of Attacks



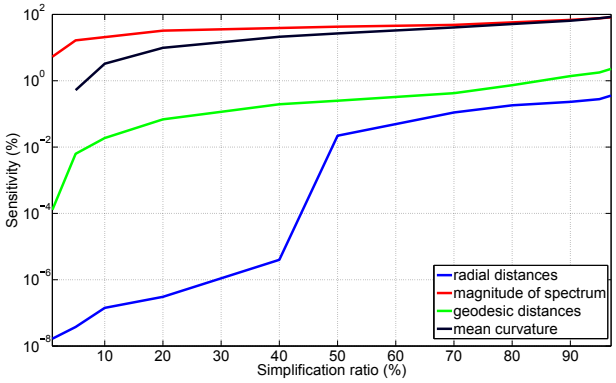
(a) Noise Addition



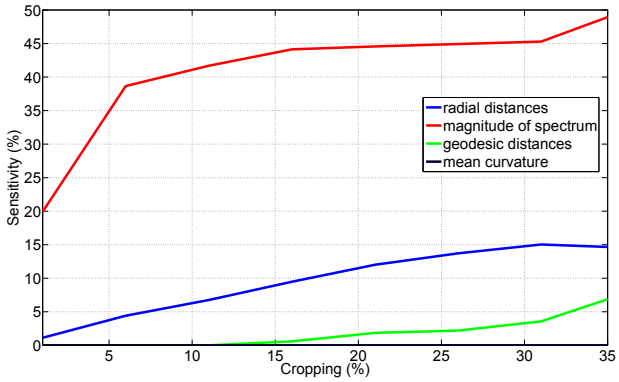
(b) Quantization



(c) Smoothing



(d) Simplification



(e) Cropping

Figure 4-7: Benchmark of the stability of geometric quantities, used to define a 3D watermark carrier, against multiple attacks

Based on the results presented above, the stability of the most stable quantities is benchmarked against increasing levels of attacks. The tested carriers are: (i) the radial distances based on the volume-weighted center of mass, the magnitude of the spectral coefficients computed with the combinatorial discretization of the Laplacian, (iii) the geodesic distances, and (iv) the mean

curvatures estimated with the jet-fitting approach. For every attack strengths in this second round of experiments, the settings are identical to the previous ones. The results are reported in Figure 4-7. The triangle soup is not depicted since only the Euclidean distance-based carriers are able to cope with this attack. The non-parameterized subdivision is omitted as well.

In summary, Euclidean distances (based on the volume-weighted definition of $\mathbf{g}$) and the magnitude of the spectral coefficients exhibit the highest stability against valumetric attacks. These quantities are indeed globally defined, i.e. they aggregate information on the whole mesh. Geodesic distances achieve slightly worse results for low levels of attacks, but for larger strengths, they show a significant drop in stability, especially at high noise levels. Because of numerical issues in the spectrum computation when meshes present degeneracies (empty facets, etc.), the results for the spectral carrier in the case of quantization are not reported.

Against the simplification attack, radial distances greatly outperform all the other carriers by several orders of magnitude. For very large simplification levels, geodesic and radial distances stay around 1% sensitivity, while the mean curvature and the spectral carriers are unstable. The latter cannot handle even slight alterations of the connectivity; but the former is in fact unchanged for simplification ratios below 5%, then skyrockets to large levels of instability above this threshold. Using the ring neighborhood in the estimation process is indeed sensitive to connectivity alterations. Finally, in case of the cropping attack, the mean curvatures are invariant (more exactly, the carrier values that can still be estimated after the cropping are identical to the original ones), while geodesic distances are greatly stable. The sensitivity of the radial distances grows almost linearly with the cropping ratio. Finally, the spectral carriers are significantly less robust than all the others, again because of the impact on the connectivity.

4.4 Conclusion

The extraction function is one of the core components of watermarking systems. The few extraction examples that have been reviewed correspond to the most widely used embodiments in state-of-the-art 3D watermarking. The stability assessments reported highlight the fundamental issues faced by these systems. For locally defined quantities, e.g. the surface area of a local patch, the robustness against valumetric attack is small. Their expected stability against desynchronizing attacks is often limited, because they commonly rely on a scale parameter to define the size of their support patch of computation, which is itself not robust to e.g. cropping or pose. Using ring neighborhoods instead of Euclidean or geodesic neighborhoods avoids this problem, but it has shortcomings when connectivity alterations are considered. This last issue was illustrated by the results of the mean curvature stability.

Normals and principal curvatures are often only involved in distortion assessment. Their sensitivity against many attacks indeed makes them altogether unsuitable for robust watermarking. Spectral quantities are interesting candidates as watermark carriers, but the computational cost is prohibitive. The eigen decomposition is also unreliable, especially for the cotangent-based discretization. The sensitivity of the combinatorial-based discretization against connectivity attacks is also a problem.

Geodesic and Euclidean distances achieve the largest stability. The former can be advantageously employed to resist cropping and pose, while the latter is more stable against valumetric attacks. Still, geodesic distances also face synchronization challenges that were not considered in this chapter. From the variety of evaluated carriers, radial distances exhibit most of the required properties for robust 3D watermarking.

To conclude, one notable limitation of this short benchmark is that it suffers from the common

modularity vs. efficiency problem. Although it provides a clear picture of the strengths and weaknesses of carriers, it does not measure synergies across the layers of the watermarking system and does not allow for joint optimization approaches.

Chapter 5

Pose-invariant embedding domain

5.1 Introduction

In 3D watermarking, a variety of embedding domains and watermark carriers has been considered. The review presented in the previous chapters shows that only very few of these domains exhibit an inherent robustness against pose, i.e., against an isometric alteration of the 2D surface approximated by the mesh. Geodesic distances are the primitives of choice to achieve pose invariance [TLHK10, LB11], as described in Section 3.2, while using the so-called Laplacian-coordinates [YI10] was also briefly explored. In this chapter, we investigate the definition of a new pose-invariant embedding domain, in an attempt to address the issues found in the aforementioned ones, such as the connectivity-dependency, the computational complexity, and the main weakness of geodesic approximations against e.g. noise addition. This new domain is based on the thickness of a 3D object.

Estimating the local thickness of complex 3D objects is a multi-faceted problem with a variety of other applications than watermarking. In computer graphics, algorithms such as mesh partitioning or curve skeleton extraction can successfully rely on a local thickness estimate such as the so-called Shape Diameter Function (SDF) [SSCO08]. The stability of this local thickness estimate is however still challenging.

5.1.1 Robustness against Operations

From our review of 3D attacks, a shape can be altered to, e.g., meet the limited computational capabilities of heterogeneous computer hardware, by matching a target level of detail. Assuming an input shape provided as a surface mesh, this goal commonly involves processing operations such as mesh simplification [Gar99]. In this context, any thickness estimate should ideally be consistent for all levels of detail. A shape can also be animated, involving complex distortions. We expect that articulated animations have only minor effects on the thickness overall, as changes only occur at the joints which are in general a small subset of a shape. In this context, a local thickness estimate should ideally be consistent across all poses of an animation. This property makes the local thickness estimate an interesting candidate to design new 3D watermarking primitives.

5.1.2 Robustness against Artifacts

When digitizing, the original physical shape is only known through sampling and approximation. A triangle mesh is an instance of such piecewise-linear approximation of a surface. In addition to the inherent uncertainty of any measurement device and imperfections of the acquisition process,

some imperfect algorithms along the geometry processing pipeline may produce a range of artifacts such as gaps, holes, non-manifold parts and triangle soups. While a thread of research has focused on repairing defect-laden data or removing artifacts, there is currently no definitive solution to such defects [BKP+10]. Moreover, some applications gather data from heterogeneous inputs and thus require the ability to convert between shape representations. These conversions also lead to artifacts such as handles or disconnected components. Ideally, a thickness estimation would provide results that are both robust and consistent for all these cases.

5.2 Related Work

One definition of the local thickness of a 3D shape is based on its Medial Axis Transform (MAT). The MAT was initially introduced to represent 2D shapes through the loci of maximally inscribed circles [Blu67]. In 3D, the medial axis is defined as the loci of centers of maximally inscribed spheres. The MAT of a 3D object is defined from the medial axis and the set of sphere radii, which defines a scalar field onto the medial axis. On the boundary of a smooth 3D object each point has a unique corresponding point on the medial axis, which is the center of the maximally inscribed sphere tangent to the boundary point. Through this correspondence one can map the radii of the spheres centered at the medial axis onto the surface, thus defining the local thickness for each boundary point (Figure 5-1) [Tag13]. Extracting the medial axis of a surface is complex however, as it is very sensitive to small variations of the surface [ABE09]. This issue is critical when dealing with defect-laden inputs, as small irregularities on a smooth surface may create large spikes on the medial axis. To alleviate this issue some robust variants taking advantage of the notion of scale have been proposed [GMPW09].

An approach to compute an intuitive and pose-invariant local thickness from surface meshes was explored with the SDF, based on statistics of local diameter estimates. Given a query boundary point $\mathbf{q}$, a single local diameter estimate is defined as the length of the segment joining $\mathbf{q}$ and the first intersection between the input mesh and a ray shot from $\mathbf{q}$ and aligned to a vector located inside an inward cone. The diameter does not rely on computing the medial axis in order to alleviate the aforementioned issue. It is experimentally shown that the SDF is stable with respect to articulated deformations and provides an effective means to consistently partition surface meshes over multiple poses. Its robustness with respect to noise and defect-laden inputs can however be improved. The original SDF has also been improved in terms of computational complexity by performing down-sampling followed by efficient interpolation [KGMS10]. This procedure improves the computational time of the original SDF, at the cost of a lower robustness for segmentation.

Extending the SDF approach, we propose a robust method to estimate the local thickness of a 3D object bounded by a surface mesh. Inspired by ideas introduced for robust medial axis extraction, we devise a scale-dependent estimation method. As contributions we (i) improve the accuracy of the original SDF, and (ii) provide several experimental pieces of evidence that illustrate the robustness of the proposed approach. These results also show benefits for robust shape segmentation. In the following the term 'thickness' relates to our robust scale-dependent diameter-based thickness, while 'mathematical thickness' relates to the radius of the maximal inscribed sphere (MAT).

5.3 Algorithm

The input to our algorithm is a surface triangle mesh $\mathcal{M}$ approximating the 3D object. $\mathcal{M}$ may contain defects such as noise or holes. The algorithm for computing the thickness of $\mathcal{M}$ comprises

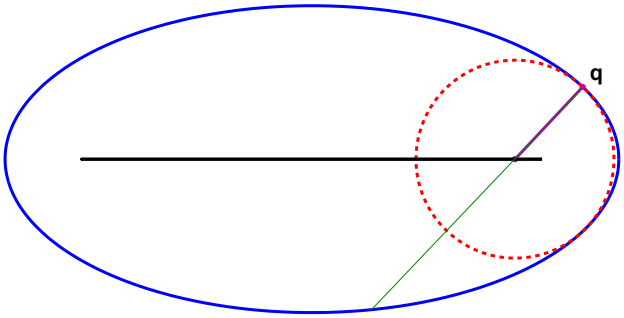


Figure 5-1: Thickness on an ellipse. The medial axis is depicted in black. The thickness (purple line) is defined as the radius of the maximal inscribed ball (red dotted line) associated to a boundary point **q**. The diameter is depicted in green. Notice that taking half the diameter value is different from the computing the thickness, as it yields a larger value.

two main steps. First, we compute a cloud $\mathbf{D} \in \mathbb{R}^{3\times n_s}$ of n_s 'half-diameter' points $\mathbf{d}_i$ (Section 5.3.1). Note that for simplicity, the matrix $\mathbf{D}$ also denotes the set of half diameter points. This step extends a curve skeleton extraction technique originally presented as a direct extension of the SDF computation [SSCO08]. Second, we define a robust scale-dependent thickness function t_k (defined for arbitrary query points on $\mathcal{M}$) using a noise- and outlier-robust distance function between each query point $\mathbf{q}_i$, and the half-diameter point cloud (Section 5.3.2).

Algorithm 1 provides a general overview of our method.

Algorithm 1 Overview of our thickness estimation algorithm.

procedure THICKNESS(Input mesh $\mathcal{M}$; sampling size n_s; series of boundary point queries $\mathbf{Q}$; scale parameter k)
    Random sampling of $\mathcal{M}$ with n_s points
    for all Sample points $\mathbf{s}_i$ **do**
        Probe mesh volume at sample $\mathbf{s}_i$
        Compute a local estimation of the diameter
        Create a half-diameter point $\mathbf{d}_i$ and add it to $\mathbf{D}$.
    end for
    for all Query points $\mathbf{q}_i$ in $\mathbf{Q}$ **do**
        Search appropriate k nearest neighbors $\mathbf{d}_i$ in $\mathbf{D}$.
        Compute robust distance function
        return scale-dependent thickness $t_k(\mathbf{q}_i)$
    end for
end procedure

5.3.1 Half-Diameter Points

The procedure for generating half-diameter points is summarized by Algorithm 2. The main input to the algorithm is a surface triangle mesh $\mathcal{M}$, which is first uniformly point sampled in order to generate a set of points $\mathbf{S}$. Section 5.4.1 provides implementation details on this sampling step.

Given a boundary sample point $\mathbf{s}_i$, the original SDF is computed by: (i) casting random rays inside an inward-oriented cone along the normal and computing their intersection with the input

surface $\mathcal{M}$, which defines a series of segments; (ii) measuring statistics δ based on a weighted average and the variance of the lengths of these segments; and (iii) smoothing the final thickness function over a small neighborhood and normalizing the output in log-space [SSCO08].

While this algorithm yields a sufficiently discriminative diameter estimate for mesh segmentation, we introduce several methodological differences to improve its accuracy and robustness, starting from the random re-sampling of $\mathcal{M}$.

SDF Ray-casting Strategy

Although effective when dealing with normal distributions of lengths, the outlier-robust statistics δ computed in the SDF often yields counter-intuitive estimations of the diameter. Consider the following dummy example: with two (infinite) parallel planes, the diameter can be estimated exactly and is equal to (twice) the mathematical thickness. Most importantly, this configuration does not involve scale-dependent quantities. Therefore, any thickness computation algorithm should provide an estimate as close as possible to half the diameter value. However, averaging multiple lengths inside a cone systematically overestimates the diameter. When dealing with mechanical parts and piecewise flat surfaces, this situation is quite common and δ always over-estimates the expected value. A more accurate value for δ would then be (half) the minimum length of the rays cast inside the cone.

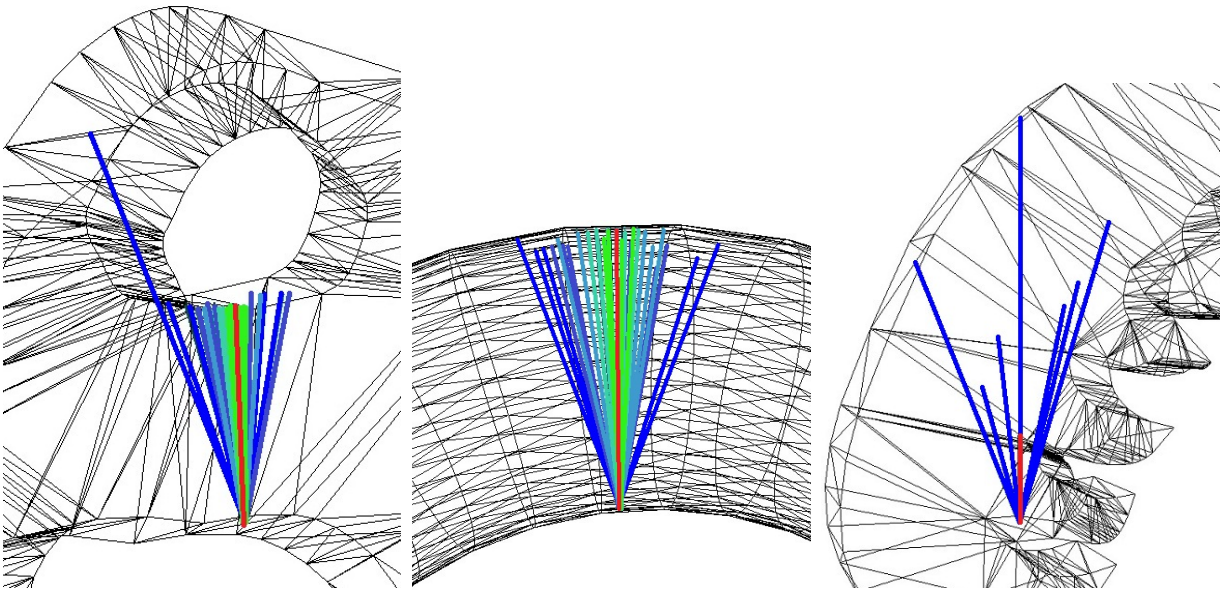
In the case of a tubular section with a circular cut, the mathematical thickness also coincides with half the length of a ray cast along the normal. This value is however not given by the minimum lengths of all possible random rays cast inside the cone. At the bifurcation of a Y-shape, the larger the opening of the cone, the more noise is added to the diameter estimation δ when taking an average ray length. In this case, notice that the outlier-robust strategy of the SDF, which is based on the median ray length, is inappropriate: the closest value to the mesh diameter (and the mathematical thickness) is given by a single minimal estimate among many noisy larger ones.

In all these examples, the larger the half-opening ϕ of the cone, the more δ differs from the mathematical thickness or from the mesh diameter. Nevertheless, a large opening angle is necessary to capture more information from the shape and to detect masking features such as protuberances or dents depicted by Figure 5-2(c). All possible configurations are not addressed when designing statistics to compute δ from the R ray lengths. However, in cases where the notion of scale is not needed, and when the mathematical thickness and the mesh diameter are identical, the accuracy and relevance of δ can be measured by its closeness to, e.g., the mathematical thickness chosen as 'ground-truth'.

Adaptive Ray Casting

To alleviate the dilemma between a large and a small opening angle of the cone to probe the local volume, we adopt an adaptive method when casting the rays inside the cone. Such a method provides a more conservative estimation of the diameter than the SDF. The aperture angle is initially set to a large value ϕ, and R rays are cast inside this cone. Let l_{min} be the minimum ray length. This ray casting procedure is iteratively repeated while decreasing the opening angle by a step η, yielding each time a new length l_{min}. At each step the stability of l_{min} is estimated by computing its absolute growth rate r (see Algorithm 2). If r is valued above a threshold τ, the process ends and δ is set to the previous l_{min}. Otherwise, δ is set to a final l_{min} after I iterations. In practice, we set $\phi = 25°$, $R = 5$, $\eta = 2°$, $\tau = 0.8$ and $I = 10$. Thus, the procedure either ends with a variation of l_{min} larger than 80%, or when the aperture of the cone reaches $\phi - I\eta = 5°$.

The results of this procedure are illustrated by Figure 5-2. In Figure 5-2(a), δ is eventually set



(a) Parallel planes (b) Torus (c) Dent

Figure 5-2: Illustrations of 3 specific configurations for the adaptive cone opening procedure. Rays cast inside the adaptive cones are shown with different colors. Blue rays correspond to the first iterations (larger cones), green rays to the last iterations (smaller cones). The (double) diameter estimation δ is depicted as a red segment along the normal. 5-2(a) and 5-2(b): the adaptive closing reaches a very small opening angle, which improves the estimation of the diameter, as the average length of the deep blue rays would not provide such an accurate estimate. 5-2(c): the algorithm stops after the first iteration, as a large variation in the minimum ray length is detected.

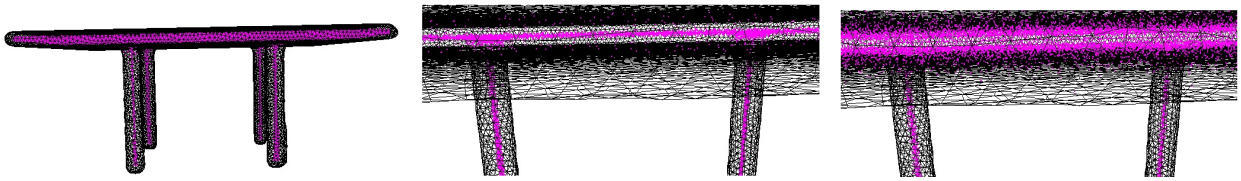
to half the length given by the ray cast along the normal. In Figure 5-2(b), the tubular section has a circular cut and our procedure ensures that the half-opening of the cone is very small when estimating the final $l_{\min}$, thus improving the accuracy of the estimate: with the previously described parameter settings, $r < \tau$ holds true until ϕ reaches its minimum value of 5°. Figure 5-2(c) shows that the adaptive closing algorithm stops when a large variation in the minimum ray length is detected. In this particular configuration, our procedure ensures that δ does not get too large.

Half-Diameter Points Construction

We create the half-diameter cloud, a point set approximating the middle of the shape using the diameter information δ (Figure 5-3). This step is similar to the one proposed for extracting a curve skeleton: the sample points are projected into the shape using their normal direction at half δ, thus creating the point set denoted by $\mathbf{D}$ [SSCO08].

5.3.2 Robust Thickness Estimation t_k

Another difference between the proposed thickness approximation and the SDF is that none of its post-processing operations (the bilateral smoothing and the normalization) are performed on δ. These operations were initially designed to counterbalance the variations due to the pose. Instead, our approach uses a variant of a robust distance function to compute a scale-dependent thickness t_k from the point cloud $\mathbf{D}$ [CCSM10]. Such an approach addresses the issue of robustness through



(a) table mesh (b) Adaptive cone (c) SDF

Figure 5-3: Half diameter cloud resulting from the SDF estimation and our procedure. 5-3(a): half-diameter points for a mesh of a table. 5-3(b): close-up on the points (purple), representing projections of samples at half their estimated δ value along the normal, using the adaptive-opening cone algorithm. This configuration corresponds to the parallel planes case. 5-3(c): same close-up, but points are the centers of facets projected at half their SDF value. In this case, two distinct parallel planes of points are created, due to the systematic overestimation of the actual diameter.

the specification of a single and intuitive parameter.

Defining the Thickness t_k

Given $k \in [\![1; n_s]\!]$, the thickness $t_k(\mathbf{q})$ is defined on the input surface as follows:

$$\forall \mathbf{q} \in \mathbf{Q},\ t_k(\mathbf{q}) = \sqrt{\frac{1}{k} \sum_{i \in [\![1,k]\!]} \|\mathbf{q}\mathbf{d}_i\|^2}, \tag{5.1}$$

where $\mathbf{d}_i$ denotes the i^{th} closest point to $\mathbf{q}$ in $\mathbf{D}$ that verifies the condition:

$$\mathcal{M} \cap [\mathbf{q}\mathbf{d}_i] = \{\mathbf{q}\}. \tag{5.2}$$

Eq. (5.2) ensures that half-diameter points and boundary query points are mutually visible. This is required to avoid the issue depicted by Figure 5-4.

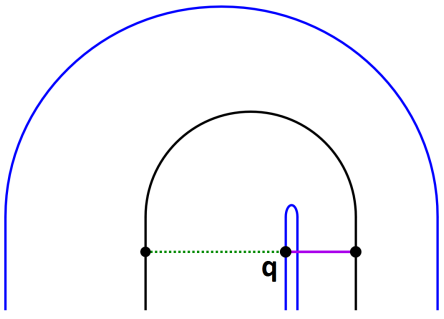


Figure 5-4: Visibility issue between a boundary point and the half-diameter point cloud. This dummy shape (blue) is formed by two parallel tubes joined by a circular connection. Notice that, at query point $\mathbf{q}$, directly using the nearest points in the point cloud (black) implies measuring the purple solid line, which yields an irrelevant thickness estimation. Instead, the visible points farther away (green dotted line) should be used to estimate t_k.

Algorithm 2 Half-diameter points computation.

procedure HALF-DIAMETER POINTS(Input mesh $\mathcal{M}$; sampling size n_s; cone aperture ϕ; rays R; iterations I; step η; variation threshold τ)
  for all facets in $\mathcal{F}$ **do**
    Generate (uniformly) a sampling set $\mathbf{S}$
    for all $\mathbf{s}_i \in \mathbf{S}$ **do**
      $r \leftarrow 0, j \leftarrow 0$
      Cast R random rays from $\mathbf{s}_i$ inside cone $(\mathbf{s}_i, -\mathbf{n}_{\mathbf{s}_i}, \phi)$ ▷ Cone defined by apex $\mathbf{s}_i$, direction as normal at $\mathbf{s}_i$, aperture denoted by ϕ.
      $l^B_{\min} \leftarrow$Compute minimum ray lengths
      while $j < I$ **do**
        $j \leftarrow j + 1, \phi \leftarrow \phi - \eta$
        Cast R rays from $\mathbf{s}_i$ inside cone $(\mathbf{s}_i, \mathbf{n}_{\mathbf{s}_i}, \phi)$
        $l_{\min} \leftarrow$Compute minimum ray lengths
        $r \leftarrow |l^B_{\min} - l_{\min}|/l^B_{\min}$
        if $r > \tau$ **then**
          break
        else
          $l^B_{\min} \leftarrow l_{\min}$
        end if
      end while
      Add projection of $\mathbf{s}_i$ along $-\mathbf{n}_{\mathbf{s}_i}$ at $\frac{1}{2} l^B_{\min}$ to $\mathbf{D}$
    end for
  end for
  return Set of half-diameter points $\mathbf{D} \in \mathbb{R}^{3 \times n_s}$.
end procedure

Computing t_k

We compute $t_k(\mathbf{q})$ by iteratively retrieving the next closest point $\mathbf{d}_i$ ($\mathbf{d}_i \in \mathbf{D}$) to $\mathbf{q}$. If Eq. (5.2) holds true for $\mathbf{d}_i$, we increment i and add $\|\mathbf{q}\mathbf{d}_i\|^2$ to $t_k(\mathbf{q})$. This procedures either ends when i reaches k or when all points in $\mathbf{D}$ have been queried. In Eq. (5.1), $\frac{1}{k}$ is replaced by $\frac{1}{\nu(\mathbf{q})}$, where $\nu(\mathbf{q})$ is the number of points in $\mathbf{D}$ which are visible from $\mathbf{q}$.

The main issue to compute t_k is related to the computational efficiency: if $k > \nu(\mathbf{q})$, computing the thickness involves useless queries to all points in $\mathbf{D}$. Although there is no way to exactly estimate $\nu(\mathbf{q})$ without testing all half-diameter points, we introduce two tests to speed up the computation: a preliminary check and an upper-bound to the neighbor search.

Before performing any other computation, we check:

$$\mathbf{q}\mathbf{d}_i \cdot \mathbf{n}_{\mathbf{q}} \leq 0,$$

where $\mathbf{n}_{\mathbf{q}}$ denotes the normal at $\mathbf{q}$ oriented outward. This condition is always verified by points visible from $\mathbf{q}$. Since this test is substantially faster than a ray-shape intersection query, it generally improves the efficiency of the process.

We also add an upper-bound $d_{\max}$ to the acceptable distance $\|\mathbf{q}\mathbf{d}_i\|$, thus limiting the neighbor search to a sphere of radius $d_{\max}$ centered at $\mathbf{q}$. When reaching a point $\mathbf{d}_i$ which does not verify Eq. (5.2), the computation of t_k terminates if $\|\mathbf{q}\mathbf{d}_i\| \geq d_{\max}$. Since the upper-bound is only applied

when the search returns unusable points, half-diameter points outside the radius $d_{\max}$ may still be used in $t_k(\mathbf{q})$. This heuristic is automatically dropped when the first point in $\mathbf{D}$ visible from $\mathbf{q}$ is farther than $d_{\max}$, so that t_k remains well-defined.

Empirically, $d_{\max}$ is set to $\frac{1}{5}$ of the largest diagonal of the bounding box. In practice, this heuristic roughly speeds up the computation by one order of magnitude. Algorithm 3 details the work flow of this algorithm, whose output t_k is scale-dependent.

Algorithm 3 k-nn based thickness computation.

procedure DIAMETERTOTHICKNESS(Input mesh $\mathcal{M}$; half-diameter points $\mathbf{D}$; boundary point query $\mathbf{q}$, scale parameter k, upper-bound $d_{\max}$)
  $t_k \leftarrow 0$, $j \leftarrow 0$
  while $j < k$ AND $\mathbf{D} \neq \emptyset$ **do**
    Pop next nearest neighbor $\mathbf{d}_i \in \mathbf{D}$ from $\mathbf{q}$
    if $\mathbf{n_q} \cdot \mathbf{qd}_i \leq 0$ **then** ▷ $\mathbf{n_q}$ is the normal at $\mathbf{q}$.
      if $\mathcal{S} \cap [\mathbf{qd}_i] = \{\mathbf{q}\}$ **then**
        $j \leftarrow j + 1$; update t_k
      else if $\|\mathbf{qd}_i\| \geq d_{\max}$ AND $j \neq 0$ **then**
        break
      end if
    end if
  end while
  return $t_k(\mathbf{q})$
end procedure

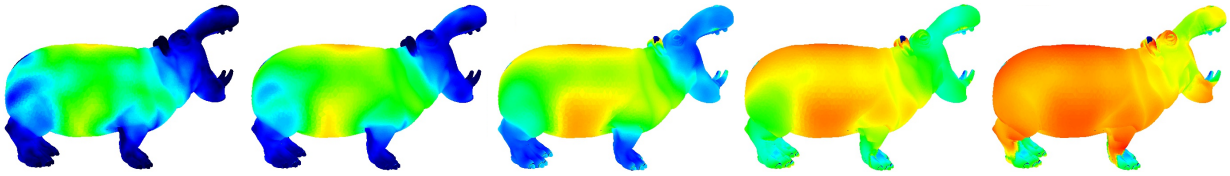
Properties of t_k

For a given size of the sampling set n_s, the parameter k provides the user with a means to trade robustness for discriminative capability of the thickness estimate. Large values of k increase the robustness through a lower sensitivity to outliers, but they also yield a low influence of small scale features since they are considered outliers compared with large scale features. In a nutshell, the scale of the thickness estimation is entirely controlled by the $\frac{k}{n_s}$ ratio.

Figure 5-5 illustrates this behavior for the *hippo* model and a fixed value of n_s. With a low k value, the importance of small scale features is enhanced for the toes and ears. This is indicated by the deep blue parts on the mesh (deep blue patches correspond to small values of t_k). With larger values of k, small scale features become less significant, and larger parts, such as the torso, become more important: deep blue parts on the mesh are no longer visible, as they have been replaced by a global red patch (larger thickness values). Notice that, in some mesh parts, t_k does not change when increasing k because of the visibility condition.

Since the thickness should be pose-invariant, very large values of k may also trigger issues: the probability that half-diameter points are not visible after a pose operation increases with their distance. In addition, the computation times directly depends on the parameters k and n_s.

Table 5.1 summarizes the main differences between our algorithm and the original SDF algorithm.



(a) $k = 5$ (b) $k = 10^2$ (c) $k = 10^3$ (d) $k = 5 \times 10^3$ (e) $k = 10^4$

Figure 5-5: Influence of $\frac{k}{n_s}$ on t_k. Variations in the local thickness t_k for different values of k in the *hippo* mesh for a fixed $n_s = 10^5$ value. The color map is the same for all meshes. For $k = 5$, 5-5(a), small thickness values are relatively important with respect to others. These are located on the legs, the toes, and the ears. Conversely, for $k = 5 \times 10^3$, 5-5(d), larger thickness values are more important, as small values have almost disappeared: the thickness estimated in the mid- and large scale features (e.g. legs, torso) has increased, indicated by the change from blue to green and red. Nevertheless, the visibility condition between half-diameter points and query points creates an upper-bound for t_k, especially noticeable at the ears. In these parts, t_k quickly becomes constant, and the model remains deep blue.

Parameter	SDF	t_k
Sampling	Facet centers	Random sampling
Cone	Large	Adaptive-opening
Diameter	Outlier-robust average	Minimum length
Postprocessing	Bilateral smoothing Normalization	Robust distance function

Table 5.1: Main stages of the thickness computation compared with the SDF.

5.4 Implementation Detail

This section provides details on our technical choices for implementing the thickness estimation t_k. Our algorithm is implemented in C++ with components from the CGAL library [CGA]. The ray casting and intersection queries use an AABB tree data structure. The robust distance function uses an incremental neighbor search based on a kD-tree. Both processes are multi-threaded through OpenMP. On most 3D objects of our database (with a few thousands vertices), the algorithm on average 30 seconds on a PC with two quad-core processors clocked at 2.93 GHz.

5.4.1 Algorithmic Choices

Surface Sampling

We observed that when dealing with anisotropic meshes or low-complexity meshes, using only the facet centers to estimate the diameter produces biased or incorrect results with high dependency to the input discretization. The first step of our thickness estimation algorithm is thus a dense re-sampling of the input surface mesh $\mathcal{M}$ to generate a set of n_s points samples denoted by $\mathbf{S}$. A mesh-independent solution consists in generating boundary points by casting random rays inside the bounding box of the object and computing intersection points with its surface. However, for shapes with very fine levels of detail this method requires a very dense sampling to avoid overlooking parts of the object. Our default random sampling method is thus based on uniform sampling of

each triangular facet, with a number of samples proportional to the contribution of the facet to the total area of the input surface mesh. In Section 5.5.2, we detail how we set the value for n_s for benchmarking, as the scale of the thickness estimation is controlled by the ratio $\frac{k}{n_s}$. Fixing n_s thus leads to a unique parameter for scale selection. In addition, when the set of point queries $\mathbf{Q}$ is known before thickness estimation, we add $\mathbf{Q}$ to $\mathbf{S}$.

Note that such a choice departs from the recent down-sampling approach [KGMS10] for computing SDF values. Our goal is to improve both the robustness and accuracy of the thickness estimation, while the down-sampling strategy aims at decreasing the computation time of the SDF. Still, our initial re-sampling strategy could also be used to reduce the number of points at which the diameter will be estimated, e.g. by setting n_s to a lower value than the number of facets.

Cone Sampling

In the original SDF approach, the diameter estimation δ is obtained through an outlier-robust weighted average of the lengths of rays cast inside a cone. The weights are designed to compensate for the bias in the casting of random rays: uniformly generating angles in $[0, \phi]$, where ϕ denotes the half opening angle of the cone, yields a non-uniform ray sampling of the cone.

In our implementation the random rays are uniformly generated inside the cone. Let $\mathbf{s}$ denote a point on the surface, $\mathbf{n_s}$ the normal vector at $\mathbf{s}$ (unitary, pointing outward), $(\mathbf{x_s}, \mathbf{y_s})$ two orthogonal unit vectors spanning the tangent plane to the surface at $\mathbf{s}$ and $\mathbf{r}$ the direction vector for the random ray. The cone is defined by its apex $\mathbf{s}$, its direction $-\mathbf{n_s}$ and its aperture ϕ. Let $\text{rnd}[a, b]$ denote a uniform random number generated within $[a, b]$. $\mathbf{r}$ is then defined as follows:

$$\mathbf{r} = -\mathbf{n_s} + \lambda(\cos(\theta)\mathbf{x_s} + \sin(\theta)\mathbf{y_s}), \tag{5.3}$$

with:

$$\lambda = \sqrt{\text{rnd}[0, \tan^2\phi]}, \qquad \theta = \text{rnd}[0, 2\pi[. \tag{5.4}$$

In addition, we always cast a ray in the direction of $-\mathbf{n_p}$.

Notice that by requiring that (i) the query points in $\mathbf{Q}$ at which t_k is to be estimated are used in the sampling procedure, (ii) the ray casting procedure always include the normal, (iii) δ is set to half the minimum ray length, and (iv) the projection of the samples is along the normal, we ensure that there is always at least one point $\mathbf{d}_i$ for which Eq. (5.2) is matched. The thickness $t_k(\mathbf{q})$ is thus defined for all points in $\mathbf{Q}$. Moreover, if $\mathcal{M}$ is a watertight surface mesh, all points in $\mathbf{D}$ are inside the mesh.

5.4.2 Complexity

The complexity of the algorithms mainly depends on the number of ray casting queries, which itself depends on the number of facets $|\mathcal{F}|$. For the SDF, the complexity is $O(R|\mathcal{F}|)$, R rays being cast at each facet center. For our method a worst case scenario has a complexity of $O(n_s(RI + |\mathbf{Q}|))$, where n denotes the number of sampling points on the surface, and I is the maximum number of iterations of the adaptive ray casting procedure. This corresponds to a configuration where: (i) the adaptive ray casting always reaches the smallest opening angles; (ii) for each query point in $\mathbf{Q}$, all the half diameter points in $\mathbf{D}$ have to be checked against Eq. (5.2). Conversely, the best case scenario involves $O(2n_sR + |\mathbf{Q}|k)$ ray casting queries: (i) the adaptive ray casting always ends after a single iteration; (ii) Eq. (5.2) holds true for all query points in $\mathbf{Q}$ and their k nearest half diameter

points. In our experiments we observe that our implementation of the original SDF approach runs between 2 and 5 times faster than t_k on most meshes. These results however greatly depend on the input mesh and the number of visibility checks performed (Section 5.3.2), which also depends on the parameter settings described below.

5.5 Experiments

In the following, we denote by δ_{SDF} the thickness computed using the original SDF method, with the following modifications: (i) we use the unbiased ray casting strategy described in Section 5.4.1, and therefore do not use any weight-based compensation mechanism; (ii) we do not perform the log-based normalization, as we benchmark the accuracy of the estimate and require the actual thickness measurements.

5.5.1 Setup

Database

The performance of δ_{SDF} and t_k are benchmarked on a database of 392 meshes. Most of them are watertight [Wat07], and the number of facets ranges from a few hundreds to around 100k. These meshes contain both articulated and non-articulated shapes, as well as mechanical parts, as partially shown by Figure 5-6.

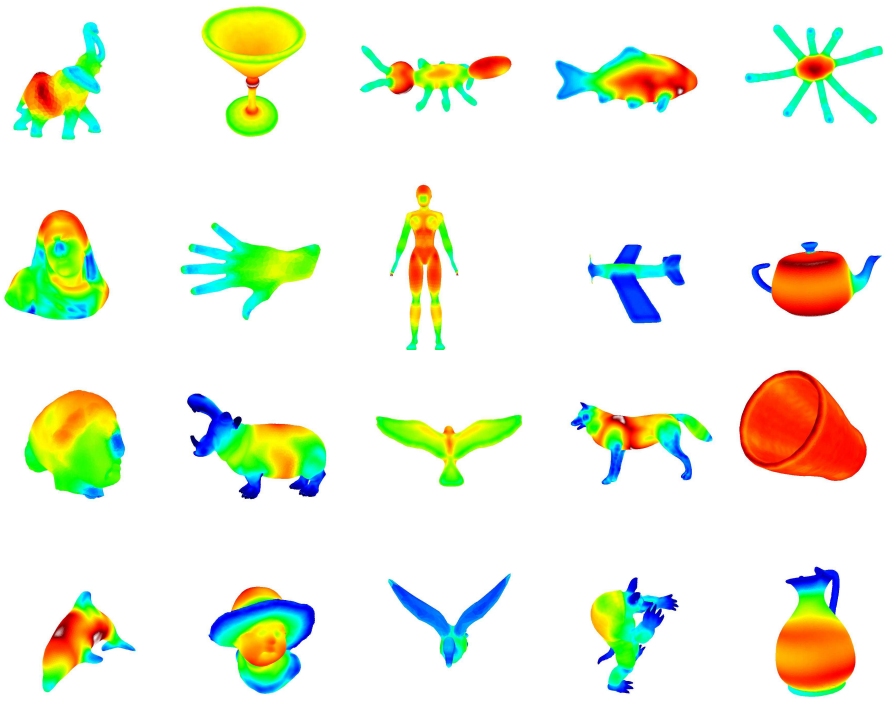


Figure 5-6: Subset of meshes in our database. Estimated thickness values are depicted with a color ramp ranging from blue to red. Note that the color map has not been normalized between meshes.

When benchmarking against specific distortions such as, e.g., addition of noise, a smaller subset of the database is used, described in Table 5.2. This subset contains 4 articulated meshes and 1

mechanical mesh, with large variance in the number of facets and feature sizes.

Mesh	# Facets	# Vertices	Properties
elephant	5.6×10^3	2.8×10^3	Thin features
U	168	86	Mechanical part
fish	14×10^3	6.8×10^3	Smooth surface
giraffe	27×10^3	9.2×10^3	Anisotropic
armadillo	50×10^3	24×10^3	Many small bumps

Table 5.2: Meshes used for benchmarking against distortions.

Performance Metrics

We benchmark the (i) instability, (ii) accuracy, and (iii) robustness of the estimation algorithms against distortions such as noise addition, simplification, etc. The performances are always benchmarked both locally for a facet and globally for the entire mesh.

Instability The instability measures the intrinsic[1] uncertainty of a thickness estimate, as both δ_{SDF} and t_k are based on a stochastic approach. In the following, $\mu^q(f)$ denotes the average thickness estimated at the center of facet f over q runs of each algorithm. For instance, denoting by $\delta^i_{\text{SDF}}(f)$ the estimated δ_{SDF} at trial i, $\mu^q(f) = \frac{1}{q}\sum_{i=1}^{q} \delta^i_{\text{SDF}}(f)$. In addition, $\sigma^q(f)$ denotes the standard deviation of the estimation.

1. The *(intrinsic) local instability* $I^q(f)$ is measured by the coefficient of variation of an estimate at a facet center, as follows:
$$I^q(f) = \frac{\sigma^q(f)}{\mu^q(f)}. \tag{5.5}$$

2. The *(intrinsic) global instability* I^q is defined as the average local instability over the surface mesh, as:
$$I^q = \frac{1}{n_f}\sum_{f \in \mathcal{F}} I^q(f). \tag{5.6}$$

Note that the instability is defined using several runs of an algorithm on the same mesh. We set $q = 4$ in all experiments.

Accuracy In general, the accuracy of a method is defined with regard to a so-called 'ground-truth'. Given a ground-truth defined per facet, the following metrics are defined:

1. The *local accuracy* is measured for a given mesh facet by computing the relative error between the averaged output of an algorithm and the ground-truth at the facet center. Denote by $g(f)$ the ground-truth at the center of facet f. We define the local accuracy $a^q(f)$ as:
$$a^q(f) = \frac{|\mu^q(f) - g(f)|}{g(f)}. \tag{5.7}$$

[1] 'intrinsic' here characterizes the fact that no alteration of the mesh is performed in-between measurements and the variations are only caused by the thickness estimation method.

2. The *global accuracy* on a mesh is then given by averaging the local accuracy over all facets.

$$a^q = \frac{1}{n_f} \sum_{f \in \mathcal{F}} a^q(f). \tag{5.8}$$

For the thickness computation, defining a ground-truth is a complex task. In the following, the mathematical thickness, which is defined at point $\mathbf{p}$ as the radius of the maximal ball associated with $\mathbf{p}$ (see Section 5.2), is selected. This choice requires computing the medial axis analytically, which is feasible for canonical shapes such as spheres, tori and infinite cylinders. However, for more complex shapes or noisy inputs, the notion of scale comes into play, and the mathematical thickness is no longer suitable as a ground-truth. Therefore, computing the accuracy with regard to the mathematical thickness would be irrelevant in most cases.

Robustness The robustness of a method against distortions which preserve the connectivity is evaluated through the average relative error. Denote by f a facet of mesh and f' its image in a distorted mesh (with facets $\mathcal{F}'$). f' is well-defined, since the distortion induces a one-to-one mapping between the two versions of the mesh.

1. The *local (per facet) error* is defined as:

$$R^q_{\mathcal{F},\mathcal{F}'}(f) = \frac{|\mu^q(f') - \mu^q(f)|}{\mu^q(f)}. \tag{5.9}$$

2. The *global error* of an algorithm is then measured by:

$$R^q_{\mathcal{F},\mathcal{F}'} = \frac{1}{n_f} \sum_{f \in \mathcal{F}} R^q_{\mathcal{F},\mathcal{F}'}(f). \tag{5.10}$$

Notice that the larger R^q, the lower the robustness. In other words, R^q can be seen as a measurement of the inconsistency of an algorithm for a given distortion.

For modifications that do not preserve the mesh connectivity, i.e., when the mapping between f and f' is lost through local mesh operators, depicting the thickness using identical color maps enables a visual comparison. A quantitative comparison is made using Eq. (5.9), where f' is chosen as the nearest facet to f in $\mathcal{F}'$. Notice that in this case $R^q_{\mathcal{F},\mathcal{F}'} \neq R^q_{\mathcal{F}',\mathcal{F}}$. Finally, another evaluation metric consists in comparing the normalized histograms of the thickness over $\mathcal{F}$ and $\mathcal{F}'$.

5.5.2 Comparison with the Shape Diameter Function

For comparison we define two comparable baseline parameter settings for δ_{SDF} and t_k. We then benchmark (i) the accuracy, (ii) the instability, and (iii) the robustness to pose for both algorithms. These last two criteria are already mentioned in the original SDF algorithm.

Parameters

Using a robust distance function instead of the bilateral smoothing of the SDF presents the immediate advantage of reducing the number of parameters for this part of the algorithm. However, setting the parameter k for computing $t_k(\mathbf{p})$ and setting the parameters of the bilateral filter are unrelated. Directly comparing the benefits and drawbacks of increasing k or the number of smoothing iterations is therefore not meaningful.

To compute δ_{SDF}, we start with a single diameter estimation for every facet center, and then apply i iterations of bilateral smoothing with a window of size w. In a regular 4–1 subdivision mesh, the size of the surface patch involved in this computation is $m = 6iw(iw+1)$, with m the number of facets. In practice, we choose $m = 72$ by setting $i = 3$ and $w = 1$, i.e. 3 iterations of a bilateral smoothing based on a 1-ring spatial neighborhood. Since the trade-off between robustness and accuracy for the estimation of local quantities depends on m, the same value is used for t_k when comparing results between algorithms.

Sampling Size Regarding t_k, we first set a normalized target sampling size $\bar{n}_s = 10^5$ in all experiments. For a given mesh $\mathcal{M}$, let A be the area of the surface, b the length of the space diagonal of the bounding box. To ensure that the sampling size is scale-invariant, the actual sampling size is defined as follows:

$$n_s = \bar{n}_s \frac{A}{b^2}. \tag{5.11}$$

k-nn Settings Half-diameter points being projections of uniformly sampled points, the area a of the surface boundary that is required to create k half-diameter points is defined as follows:

$$a = \frac{k}{d}, \tag{5.12}$$

where d denotes the sampling density, which is set to be uniform over the mesh.

Eq. (5.12) can then be rewritten using m, the average number of facets in area a as:

$$m\frac{A}{n_f} = \frac{k}{\frac{n_s}{A}} \implies k = \frac{n_s m}{n_f}. \tag{5.13}$$

As we use $m = 72$ and $\bar{n} = 10^5$ for our experiments, Eq. (5.11) and (5.13) show that k can be automatically set, its value ensuring a meaningful comparison between δ_{SDF} and t_k. In the following, we simply denote t_k as t.

Output Finally, we set $\mathbf{Q}$, the query points at which t is estimated, as all the facet centers. Therefore, for both δ_{SDF} and t, all parameters are automatically set, and a single value is eventually assigned to each facet.

Table 5.3 summarizes the baseline parameter settings.

Accuracy

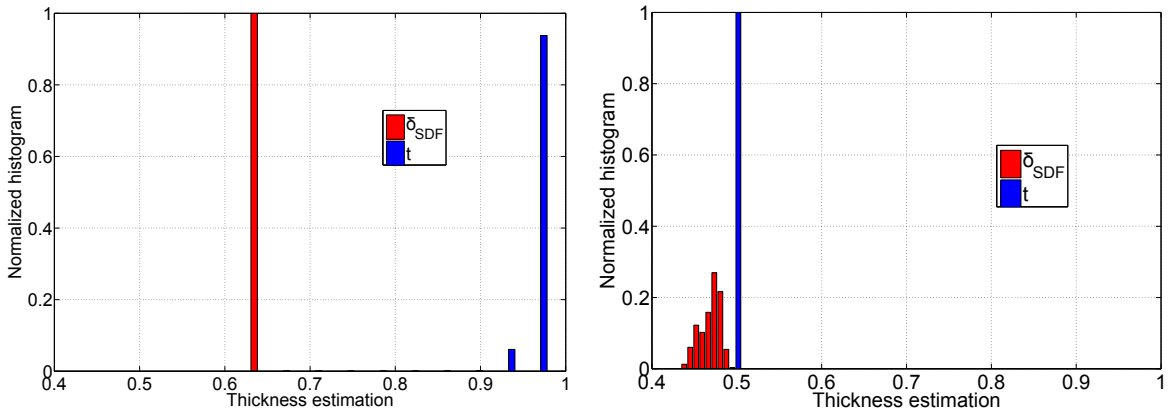
Sphere and Torus We compare the accuracy of t and δ_{SDF} for a unit sphere ($1,740$ facets), and a torus of minor radius 2.5 and major radius 1.5 ($3,200$ facets). Figure 5-7 depicts a normalized distribution of the per-facet values. Averaging multiple ray lengths in the case of a sphere yields a substantial underestimation of the actual radius: δ_{SDF} is around 0.65 all over the sphere instead of the expected unitary value. Among all rays cast inside a cone along the normal direction, only one yields the correct value (the actual radius), while the others yield smaller values. The global instability I^4 (i.e., over four iterations of the algorithm) of δ_{SDF} is around 2.5%, while it is about 1.6% for t.

For the torus, averaging ray lengths as in δ_{SDF} provides a better estimation of the half-section than for the radius of the sphere (theoretical value at 0.5), but leads to a slightly scattered distribution of δ_{SDF} around 0.48. The global instability I^4 is in this case around 3.0% against 0.03%

Parameter	δ_{SDF}	t
Sampling	-	$\bar{n} = 10^5$ (Eq. (5.11))
R (rays)	30	5
Cone opening	$\phi = 60°$	Adaptive opening (Section 5.3.1) $\phi = 25°$, $\eta = 2°$ $\tau = 0.8$, $I = 10$
Postprocessing	Bilateral filter window $w = 1$ iteration $i = 3$	Robust distance $m = 72$ (Eq. (5.13))
Output	Single estimate per facet	

Table 5.3: Baseline parameter setting to establish meaningful comparisons between the results of δ_{SDF} and t.

for t. Finally, note that the inaccuracy of the estimation process is also due to the sampling, since both meshes are only approximations of the unit sphere and torus.



(a) Unitary Sphere

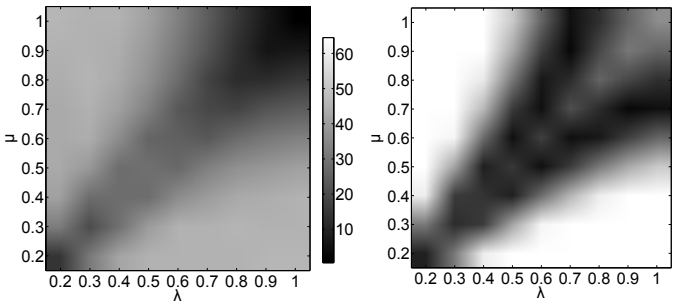
(b) Torus (major radius 2.5, minor radius 1.5)

Figure 5-7: Accuracy for a unit sphere 5-7(a) and a torus 5-7(b) measured with the distribution of δ_{SDF} and t. δ_{SDF} underestimates the radius of the sphere and exhibits a scattered distribution in the case of the torus.

Ellipsoids Both algorithms are evaluated on 81 ellipsoids (20k facets each), parameterized by their eccentricity. Their centers are at the origin, with two semi-axis (λ, μ) ranging from 2×10^{-1} to 1 (step: 10^{-1}) and the third semi-axis constantly set to $c = 1$. For each ellipsoid, the thickness t and δ_{SDF} are experimentally estimated as well as their local accuracy with regard to the mathematical thickness computed in closed form.

Figure 5-8 depicts in gray-scale the global error between each thickness estimation algorithm and the mathematical thickness. All values are normalized in order to depict the results with the same scale. On average, t is closer to the mathematical thickness, which is visually verified as the diagram 5-8(a) is darker than the diagram 5-8(b). However, when only looking at the diagonal from

$(\lambda, \mu) = (1, 1)$ to $(\lambda, \mu) = (0.2, 0.2)$, δ_{SDF} yields better results. These cases correspond to cigar-like ellipsoids, with one axis set to 1.0, and a circular cut ($\lambda = \mu$). In the upper-right part of the diagram, when the ellipsoids are very close to the unit sphere, t is more accurate than δ_{SDF}, which is consistent with the results shown by Figure 5-7. Finally, when transforming the unit sphere to a plate-like ellipsoid (λ or μ set to 1, i.e. only considering the first row or the first column of the diagrams), the accuracy of both algorithms drops abruptly, albeit less for t.



(a) t (b) δ_{SDF}

Figure 5-8: Benchmark of the global accuracy (average relative error) between the estimated thickness, i.e. δ_{SDF} and t, and the mathematical thickness for various ellipsoids with semi-axis $(\lambda, \mu, 1.0)$. The upper-right part of the diagram corresponds to sphere-like ellipsoids. On the lower-left, the ellipsoids have a cigar-like shape. On the upper-left and lower-right, the ellipsoids are plate-like shaped. Since both λ and μ have the same range, both diagrams are symmetrical.

Stability

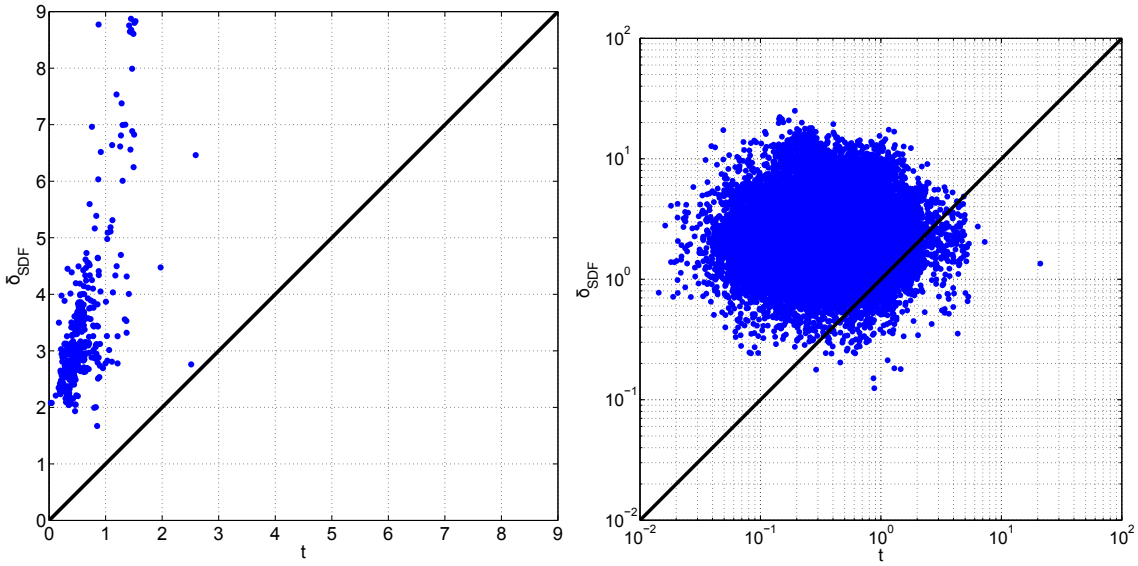
We measure the global instability I^4 of both thickness estimation methods on the entire mesh database. For δ_{SDF}, the instability is on average 3.4%. It is around 0.58% for t. Although these results confirm the stability of our thickness estimate, our computation time is about three times slower (around 30s for every mesh instead of 10s for δ_{SDF}). Figure 5-9 summarizes the results of the instability benchmark.

Figure 5-9(a) shows the global instability for every mesh in our database (392 points). For δ_{SDF}, the values are more scattered around the average 3.4% than for t, which only exhibits 3 outliers (above 2% global instability). These outliers correspond to mechanical parts: the point close to the diagonal represents the U shape, while the points in the upper-part of the diagram stand for bowls or plates. All the articulated meshes are located around the point $(0.5; 3)$.

Figure 5-9(b) shows the local instability of both methods for the *armadillo* mesh. The average instability is around 0.5% for t and 2.8% for δ_{SDF}. Both methods exhibit local outliers, but their ranges are slightly different. For t, some facets have an instability below 0.1%, and a single facet has an instability above 10%. For δ_{SDF}, all values are above 0.1%, and many above 10%. In other words, the local instability of t: (i) has a lower upper-bound than δ_{SDF}, with very few unstable estimates; (ii) is lower than the local instability of δ_{SDF} for most facets (the point cloud is mostly above the diagonal); (iii) has a few highly stable estimates.

Robustness to Pose

A series of poses of the *elephant* mesh are used to benchmark the robustness of δ_{SDF} and t. This mesh has 85k facets and contains some self-intersections at the ears. Figure 5-10 shows the local



(a) Global instability (%) for every mesh in our database.

(b) Local instability (%) for the *armadillo* mesh.

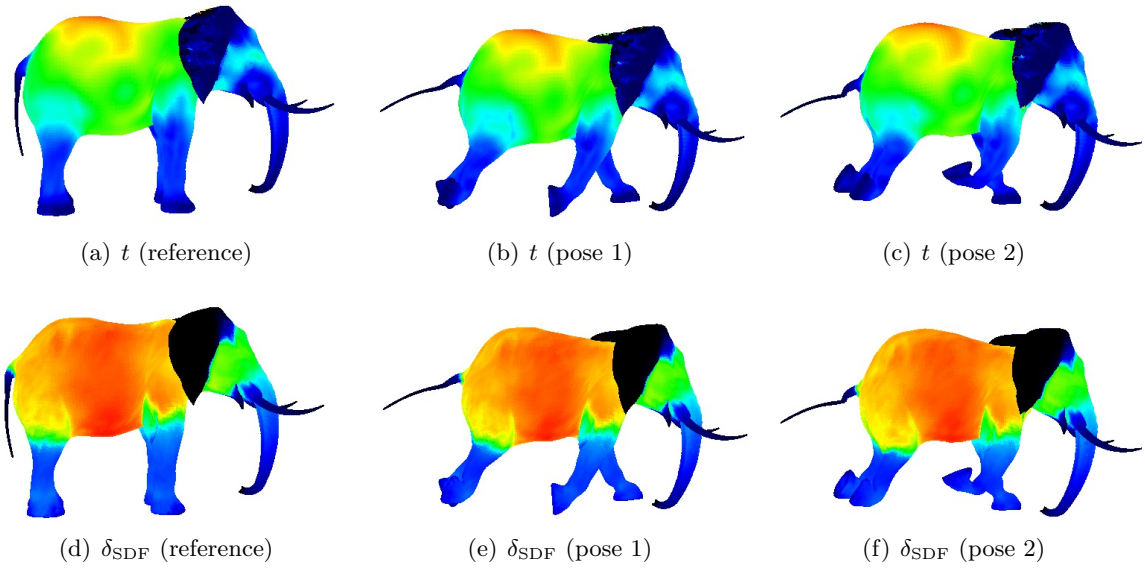
Figure 5-9: Global instability over the database and local instability for a single mesh, using δ_{SDF} and t. Points above the diagonal line $y = x$ represent facet centers for which δ_{SDF} is more unstable than t. 5-9(a): for a small number of meshes, the global instability of δ_{SDF} gets very large (around 7%), while it always stays below 3% for t. 5-9(b) the logarithmic scale denotes a large scattering, with the local instability ranging from 10^{-2}% to 10^{2}%.

thickness estimates on the reference mesh and two differently posed versions. This first experiment assesses the visual consistency of both algorithms over different poses.

A quantitative analysis of the robustness is depicted by Figure 5-11. Figure 5-11(a) first illustrates the global robustness of δ_{SDF} and t using the global error R^4 and all the various poses of the *elephant* mesh. Note that the reference values in Eq. (5.10) correspond to the ones computed on the reference pose (depicted on Figure 5-10(a) and Figure 5-10(d)). The error is upper-bounded by 16% for δ_{SDF} and by 11% for t. For almost all poses, t is more consistent than δ_{SDF} (points above $y = x$). Figure 5-10(b) and Figure 5-10(e) show the local thickness for the single pose where δ_{SDF} is more consistent than t.

Figure 5-11(b) illustrates the local robustness of both methods for one pose of the mesh. This pose corresponds to the point with coordinates $(10.73, 12.54)$ on the global error diagram (Figure 5-11(a)), i.e. one of the worst result for both methods. The actual estimated thickness is depicted by Figure 5-10(c) and Figure 5-10(f).

The local error for δ_{SDF} and t exhibits a large number of outlier values, i.e., facets for which the estimated thickness is highly modified, or conversely almost exactly identical between poses. The few facets with a large error (above 10^3%) could correspond to the joints of the model, at which the local thickness greatly varies between poses. However, these facets are not the same for δ_{SDF} and for t. Moreover, none of them are located at the joints, but only on the ears of the elephant. Furthermore, the largest local instability values (computed with Eq. (5.7)) are located in the same regions. This indicates a correlation between large local errors, large instability of the estimates, and the self-intersections of the mesh (which are located near the ears).



(a) t (reference) (b) t (pose 1) (c) t (pose 2)

(d) δ_{SDF} (reference) (e) δ_{SDF} (pose 1) (f) δ_{SDF} (pose 2)

Figure 5-10: Thickness estimation for different poses of an elephant mesh. The same range of colors is used in all cases. Both δ_{SDF} and t yield visually consistent results. Note that the estimated thickness values greatly differ between the two algorithms.

5.5.3 Benchmarking versus Distortions

This section presents the results on the robustness of t with regard to a variety of synthetic distortions, most of which are taken from the benchmark presented in Chapter 4. Without a direct comparison with δ_{SDF}, the parameter settings established in Section 5.5.2 is no longer needed. In particular, setting k as a function of the number of facets n_f is ill-suited for benchmarking against e.g. simplification. Similarly, making parameters dependent on the surface decreases the robustness against e.g. noise, as the parameters are altered over different noise magnitudes.

In the following, we simply change k so as to use a constant $\frac{k}{n}$ value. In other words, the scale at which the estimations are performed stays constant.

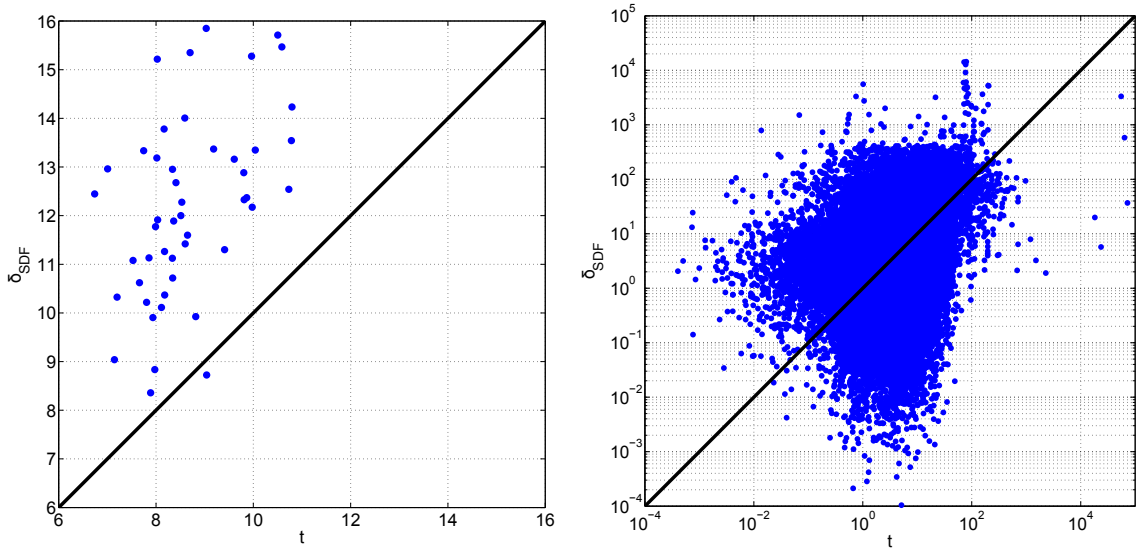
Affine Transformations

The most basic affine transformations in $\mathbb{R}^3$ consists in rigid transforms (rotations, translations), for which all the algorithms are in theory invariant. For a scaling operation with ratio α, we simply compute an estimate $\hat{\alpha} = \frac{1}{n_f} \sum_{f \in \mathcal{F}} \frac{\mu^m(f)}{\mu^m(f')}$ and compare it to α, using all the 392 meshes in our database. With α ranging in $[10^{-3}, 10^3]$, the relative error between the estimation $\hat{\alpha}$ and the ground-truth is always below 0.01%.

Uniform Geometric Noise

The local thickness is estimated for the 5 meshes in Table 5.2 after modifying the vertex coordinates by adding a uniform noise vector *along the normal*, a.k.a. a 'normal noise'. Denote by $\mathbf{n}$ the local unit normal. The noise vectors are locally generated uniformly within $[-\frac{s}{2}\mathbf{n}; \frac{s}{2}\mathbf{n}]$, with s the noise magnitude, corresponding to a ratio of the longest diagonal length of the bounding box.

Figure 5-12 shows the global error vs. noise levels. For articulated meshes and $s \leq 0.1\%$, t yields on average consistent results. Above $s = 0.5\%$, the thickness estimation exhibits low robustness,



(a) Global error (%) over a series of pose of an elephant mesh.

(b) Local error (%) for a single pose of an elephant mesh.

Figure 5-11: Robustness of t and δ_{SDF} against pose 5-11(a): global error in the thickness estimation on a series of poses of the *elephant* mesh. Every point represents a different pose, with its position w.r.t. the diagonal line indicating whether t or δ_{SDF} shows greater consistency. The local (per-facet) error of the pose represented by the point $(10.73, 12.54)$ is shown on Figure 5-11(b). Both algorithms present a similar behavior, with an average error around 10%, and large outliers: some values are above 10^3%.

as the global error quickly increases. Conversely, for the mechanical part, t shows a larger initial global error, but a slower decrease in performance with larger values of s. For $s \leq 0.1\%$ (around 2%), this is due to the high (intrinsic) global instability of the algorithm as shown in Section 5.5.2. For all the articulated meshes, this global instability stays around 0.5%, for all levels of noise.

Figure 5-13 presents the local error for the *giraffe* mesh. It shows that increasing s yields a global decrease in performance, as the local error increases for 90% of the facets. In particular, the large increase in the global error at $s = 0.5\%$ comes through a drop in robustness for all facets, and not for some specific parts of the mesh. Finally, Figure 5-13(b) illustrates the location of the local error at $s = 2\%$ on the distorted mesh. Small error values are located on large features (torso), while small features, such as ears, exhibit the largest errors. This is explained by the fact that the additive noise is generated without taking into account the feature size: small features are relatively more distorted than larger features. This phenomenon is magnified by the size of the space diagonal of the bounding box with regard to most of the mesh features.

Smoothing

The robustness of the thickness estimate is benchmarked against a common mesh smoothing method [Tau95]. We monitor the global error when increasing the number of smoothing iterations in Figure 5-14. We provide a close-up on the distribution of local error for the *armadillo* (Figure 5-15), as well as its location (Figure 5-15(b)).

For three out of four articulated models, t shows a large consistency (*elephant*, *armadillo* and *giraffe* curves in Figure 5-14) over smoothing. For the U mesh, t conversely yields much lower

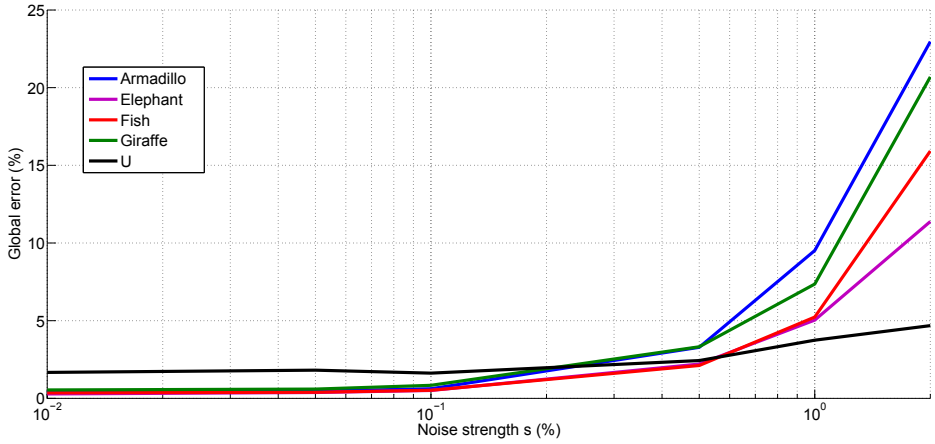


Figure 5-12: Global error between the thickness estimated on a series of 5 input meshes and their distorted version through normal noise addition. On all articulated models and for levels of noise $s \leq 0.1\%$ (s is defined w.r.t. the space diagonal of the bounding box), t yields a constant global error around 0.5%, indicating a large robustness. For $s > 0.1\%$, the global error increases rapidly, and t becomes less consistent. The U mesh (mechanical part) shows very different results: the error is systematically larger for small levels of noise (around 2%), but the drop in robustness for larger noise magnitudes is slower.

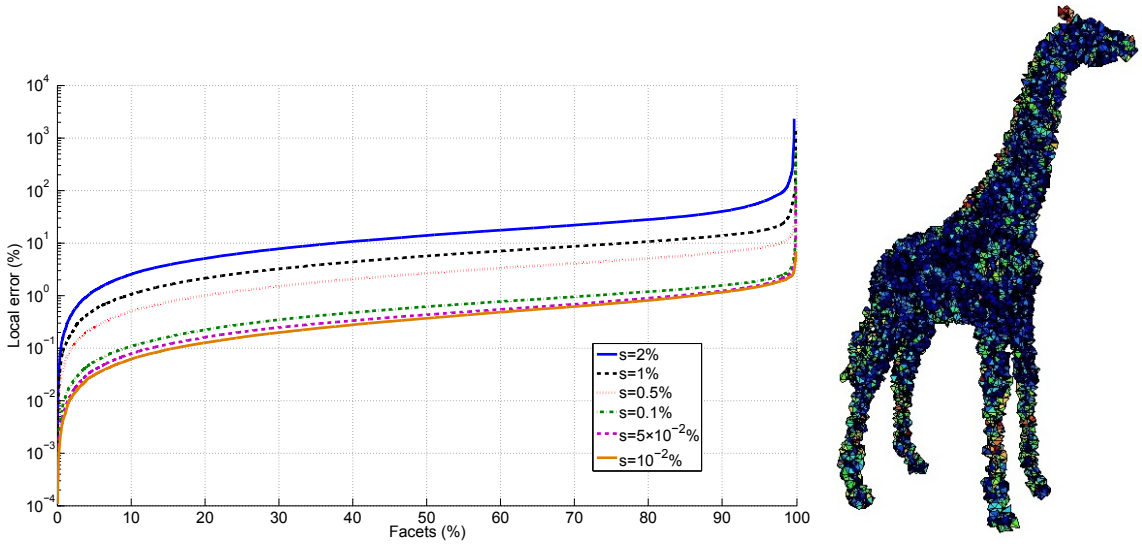
consistency, with a global error above 10% after a single smoothing iteration. Finally, the results for the *fish* mesh are significantly better than all the others, with a global error below 0.2% even after 20 smoothing iterations.

These results closely correlates to the distortion introduced by the smoothing iteration, estimated through the Root Mean Square (RMS) (from the distorted mesh to the original mesh) metric [CRS98]. For the U mesh, the RMS after one smoothing iteration is estimated around 4×10^{-2}, while it is only in the order of 10^{-4} for the articulated meshes. This is caused by the sharp features of the mechanical parts, which are heavily distorted by the smoothing process. Similarly, the RMS for the *fish* mesh stays very low (4×10^{-5} after one iteration and 4×10^{-4} after 20 iterations), as the original mesh does not present any sharp features.

Finally, the close-up on the distributions of the local error and the actual mesh of the *armadillo* (Figure 5-15(a) and Figure 5-15(b)) shows that t is not only robust against smoothing operations at a global level, but also at a local level: the range of the local error for 90% of the facets stays approximately within two order of magnitude, i.e. between 0.1% and 10%. The largest local error values are also correlated with the small bumps on the mesh and the small extremities, e.g. the fingers, which are heavily modified by the smoothing process.

Triangle Soup

Figure 5-16 reports the consistency results of t against this type of distortions (see Section 4.2.1). Notice that although the connectivity has changed, an obvious one-to-one mapping exists between facets in the original and in the modified mesh. Computing the global error R^4 with Eq. (5.10) is therefore still straightforward. The curve corresponding to the U mesh clearly presents inconsistencies, as the error decreases with the magnitude of the distortion. For all articulated meshes, t shows consistency until $r = 30\%$. An example for the *armadillo* mesh and $r = 40\%$ is given in Figure 5-17.



(a) Distribution of the local error for the *giraffe* mesh and different noise magnitudes.

(b) Local error on the giraffe mesh with 2% noise.

Figure 5-13: Close-up on the robustness against noise addition in the normal direction. 5-13(a): distribution of the local error for one of the articulated mesh (*giraffe*) and different values of s. Note that the per-facet values are sorted along the horizontal axis for each curve. All distributions are very similar: (i) for 80% of the facets (part of the curves between $x = 10\%$ and $x = 90\%$), the error stays within the same order of magnitude, e.g. between 1% and 10% for the largest noise magnitude; (ii) 10% of the facets exhibit outlier values (parts of the curves below $x = 5\%$ and above $x = 95\%$), with very low or very large errors. 5-13(b): local error for $s = 2\%$ depicted on the mesh. The upper-part of the local error distribution (red facets) corresponds to the small features, such as the ears. Using a scale-independent additive noise on this models creates very large distortions, as the giraffe exhibits a wide range of feature sizes (small for the tail, legs and ears, and large for torso), and the space diagonal of the bounding box is very large.

These figures show that t relies loosely on the mesh connectivity, since all facets have been disconnected. This provides robustness in the presence of small holes and cracks.

Simplification

Practical thickness estimation must provide consistent results for different levels of detail of the same mesh. On Figure 5-18, t is computed on the benchmarked meshes with levels of simplification ranging from 90% to 0.01% (indicating the percent of remaining edges with regard to the original mesh). For the U (resp. the *elephant*) mesh, with 252 (resp. $8,337$) edges, this simplification process quickly reaches too large a level of distortion for the model to be recognizable. The estimation of the global error becomes then meaningless, and the curves exhibit incoherences (decreased error with increased simplification). Moreover, computing a mapping between facets in the distorted and the original mesh based on their distance also create issues with the global error computation. For the *armadillo*, the *fish* and the *giraffe*, t shows robustness until a 5% simplification ratio.

Figure 5-19(a) and 5-19(b) display the actual thickness computed on the original Fish mesh and a simplified version ($5 \times 10^{-2}\%$ remaining edges), showing the consistency of t at a local level. Figure 5-19(c) presents the local error between these two versions: the errors are mostly located

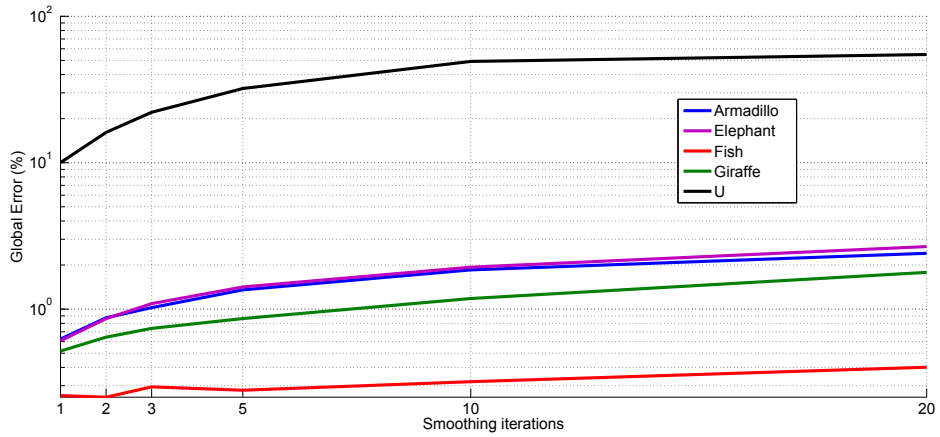


Figure 5-14: Average robustness of the thickness estimate against smoothing. We apply i iterations of the Taubin filter with parameters $(\lambda, \mu) = (0.5, -0.53)$ on 5 meshes. For the *elephant*, *giraffe* and *armadillo* mesh, the global error has the same evolution, ranging from 0.5% and reaching around 3% after 20 smoothing iterations. The *fish* mesh presents a much smaller global error than all the others. Finally, the global error is much larger for the only mechanical part in the benchmark database.

on the small features which are highly altered by the simplification process.

Remeshing

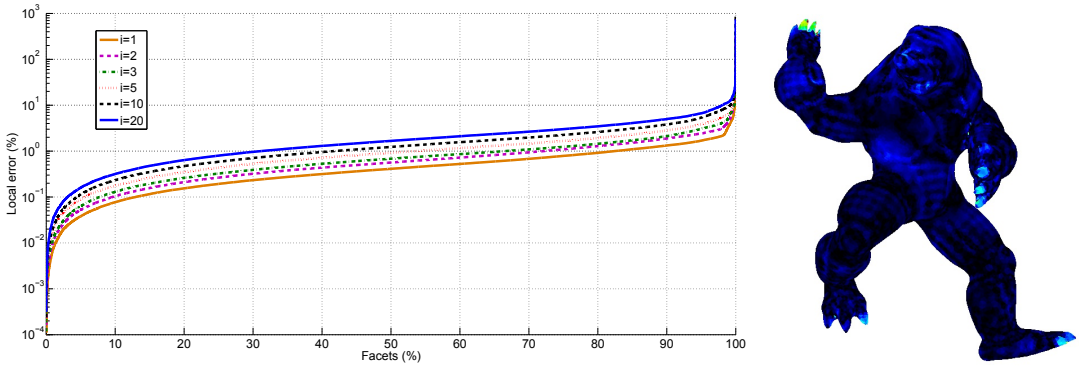
Finally, the robustness of t is benchmarked against a complete re-meshing process. We first estimate the local thickness on a watertight triangular mesh of a face with a hat. The mesh is then re-tessellated with a regular quadrangle mesh [VZ01], then transformed back into a triangular mesh by splitting facets. t is then estimated once more, and the local error is depicted on Figure 5-20. The global error is about 0.23%, with a maximum around 2%. This shows the consistency of t with regard to remeshing operations.

5.5.4 Segmentation

One of the main potential applications of local thickness estimation is to enable a robust and efficient mesh partitioning. This was originally achieved by using a soft clustering of the SDF values, followed by a graph-cut computation [SSCO08]. Note that the segmentation method itself also improves the robustness of the processing pipeline, as regrouping facets into patches has an averaging effect.

The benefits of using t for segmentation purposes are first illustrated by Figure 5-21. We applied the segmentation process on an Elephant mesh (89k facets) with 3% noise (Figure 5-21(a)). As a result, the segmentation based on δ_{SDF} (Figure 5-21(b)) is highly modified and creates 44 small segments. The segmentation relying on t (Figure 5-21(c)) provides a more intuitive partitioning of the mesh.

Regarding the robustness of mesh segmentation, Figure 5-22 shows the median relative error in the number of segments for the t-based and the δ_{SDF}-based segmentation when applying different types of distortions to the meshes in our database: (i) an increasing number of smoothing iterations (Figure 5-22(a)); (ii) an increasing number of edge simplifications (Figure 5-22(b)). These results show that even after a large number of smoothing iterations, e.g. 50, the number of segments



(a) Distribution of the local error for the *armadillo* and different number of smoothing iterations i.

(b) Local error on the *armadillo* after 20 smoothing iterations.

Figure 5-15: Close-up in the robustness of the thickness estimate against smoothing for the *armadillo* mesh. 5-15(a): distribution of the local error. The ratio of outliers is extremely low, but the part of the curves below $x = 2\%$ (resp. above $x = 98\%$), reaches significantly larger (resp. lower) values than the average, as highlighted by the logarithmic scale. On Figure 5-15(b), these values are displayed in the case $i = 20$: large errors (pale blue, green and red) in the thickness estimation mainly lie in the extremities, e.g. toes and fingers, which are more altered by the smoothing process. Note the pattern of average-valued local error (medium blue) all over the mesh. It corresponds to a pattern of small bumps on the original surface.

created by the t-based segmentation algorithm is still very close to the original ones, around less than 2% variation. In a similar manner, the segmentation based on t shows a large consistency, even after aggressive simplification of the input meshes.

5.6 Conclusion

We have introduced a robust thickness estimation method based on a shape diameter estimation. We modified the original SDF algorithm [SSCO08] by (i) introducing an adaptive scheme when sampling the local volume of a mesh; (ii) replacing the bilateral smoothing by a robust distance function to a cloud of half-diameter points, thus creating a scale-dependent robust thickness estimate over the mesh surface.

The rationale for changing the initial methodology was to increase (i) the accuracy with respect to a ground-truth defined through an exact medial axis extraction (for canonical shapes such as spheres or ellipsoids); (ii) the intrinsic instability of the stochastic process; and (iii) the robustness of the estimate against alterations of an original mesh. Our experiments confirm the accuracy of our thickness estimation, illustrated on some canonical examples for which an analytical ground-truth (the mathematical thickness) is both meaningful and well-defined. They also indicate a substantial improvement in the stability of the estimation process: with comparable parameter settings, the SDF-based thickness estimation shows on average an instability one order of magnitude larger than our method. We have also benchmarked our method against a wide range of modifications such as pose, noise addition, triangle soup, simplification and remeshing. The results show that for some distortions, our method exhibits an increased robustness with respect to the original strategy.

Our method has room for improvement: its application is in general limited to articulated shapes such as humanoids or animals. On mechanical parts, we have shown in Figure 5-3 that our

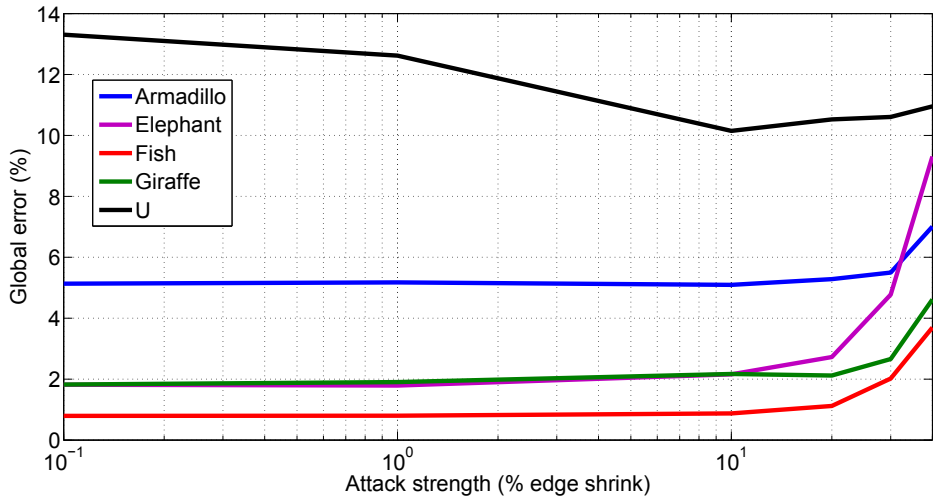


Figure 5-16: Average robustness of the thickness estimate against triangle soup. Facets are disconnected, then holes are created by shrinking edges with increasing ratios. Even with 30% shrink, the thickness estimate t only shows a 5% global error with regard to the original mesh for all articulated models. The curve corresponding to the mechanical part (U mesh) is however inconsistent, though it indicates a larger sensitivity towards cracks and holes.

adaptive strategy could be improved. Some experiments also indicate that the robustness of our method could be improved by finding a more appropriate outlier-removal strategy.

This chapter opens stimulating research directions in order to find a pose-invariant embedding domain for robust 3D watermarking based on the notion of thickness. While our findings are a promising basis to design new extraction components and robust content adaptation transforms for meshes, designing the reverse fusion function is still an open problem. In practice, building a watermarked mesh from a watermarked carrier based on the local diameter-based thickness estimate raises multiple issues. The carrier values are highly correlated and samples that are not close one the mesh surface (spatial correlation) may yet be interdependent, because of the ray casting. These interdependencies are non-deterministic, which makes it difficult to control the propagation of changes in the carrier values when altering only a handful of vertex locations. Finally, preliminary

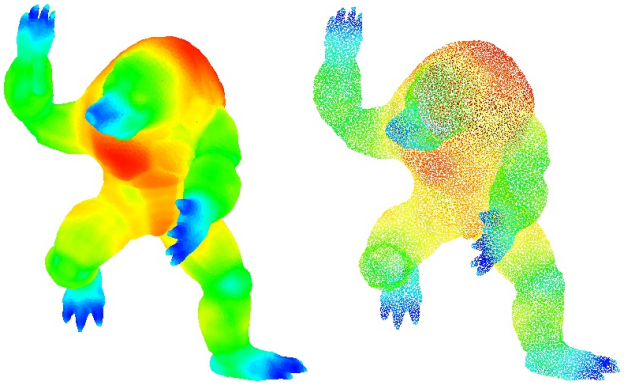


Figure 5-17: Estimated thickness on the original *armadillo* (left), and on a triangle soup version (right) with $r = 40\%$. No perceivable change appears on the mesh.

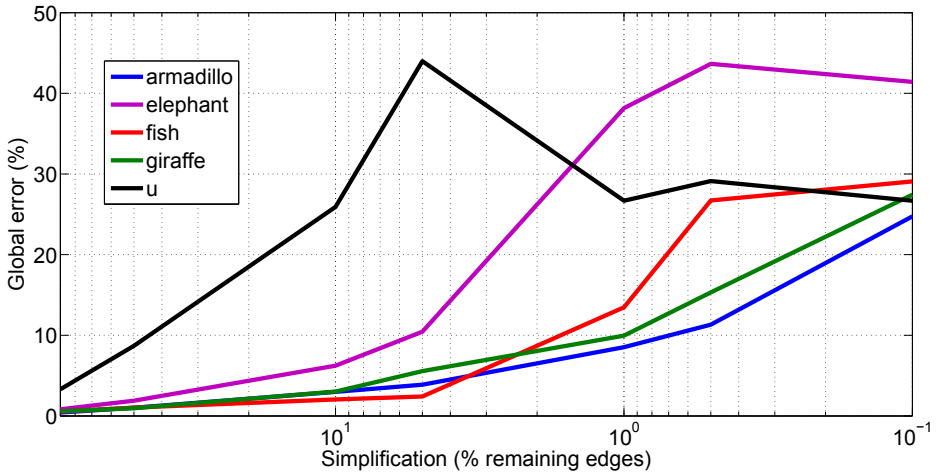
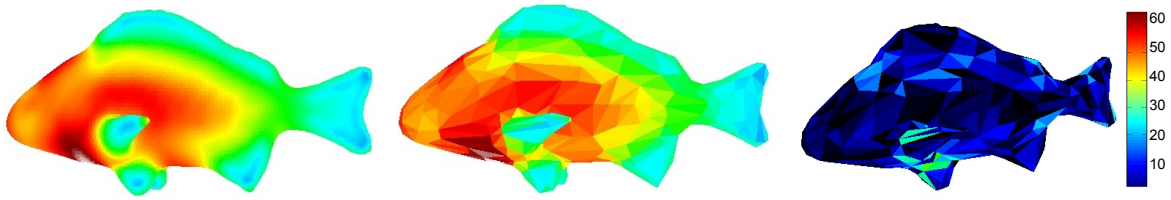


Figure 5-18: Global error for five series of meshes with increasing levels of simplification. For three models, t shows consistency until reaching a 5% simplification ratio. For the other two, the simplification quickly reaches a point where the estimation of the error becomes meaningless, as the models are visually unrecognizable.



(a) t (original *fish*)

(b) t (simplified *fish*, 5% remaining edges)

(c) Local error with 5% remaining edges

Figure 5-19: Local consistency of t for the *fish* model 5-19(a) and its simplified version 5-19(b). 5-19(c): quantitative measure of this consistency, as the local error is upper-bounded by 30%, mostly on the small features, which are heavily modified by the simplification process. On the other parts of the mesh, the local error remains below 10%.

experiments suggest that the robustness of thickness-based watermark carriers can still be improved to be on a par with, e.g., Euclidean distances to the center of mass, in case of noise additions. Hence, the second part of this dissertation will focus on using Euclidean distances to build a 3D watermarking system and will not rely on the thickness procedures introduced in this chapter.

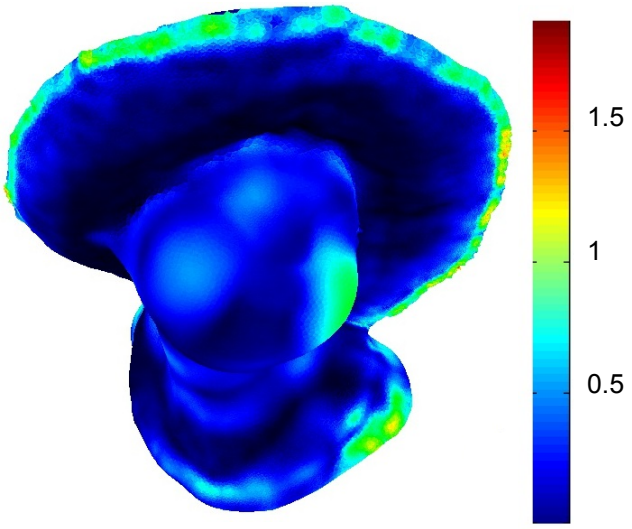
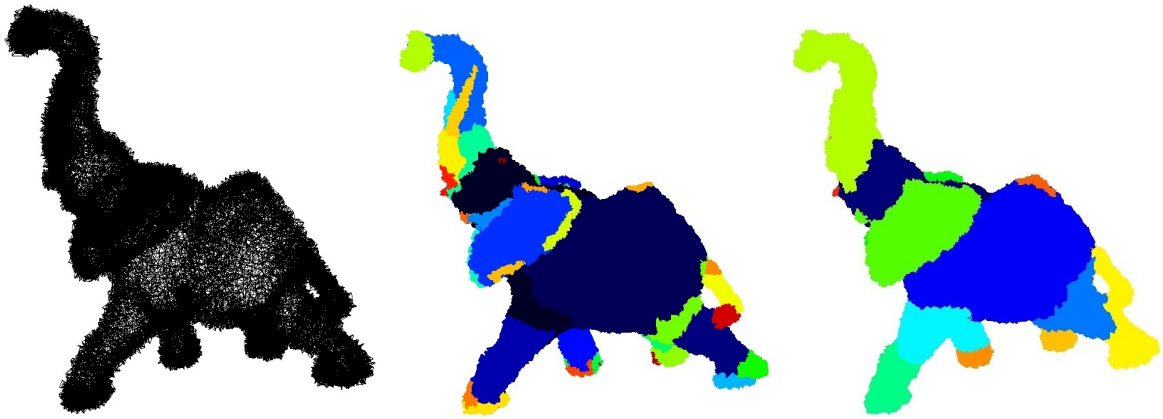


Figure 5-20: Robustness of the thickness estimate to remeshing operations. Local error (%) between t originally estimated on a mesh (face with hat) and a remeshed version. The local error stays below 2% all over the mesh.

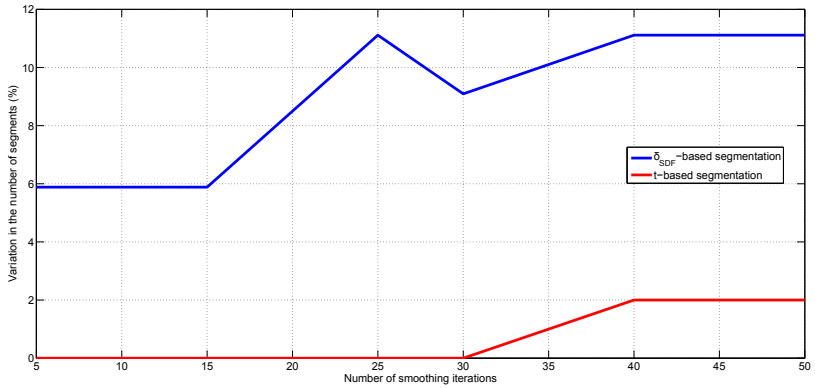


(a) Elephant mesh with uniform geometric noise

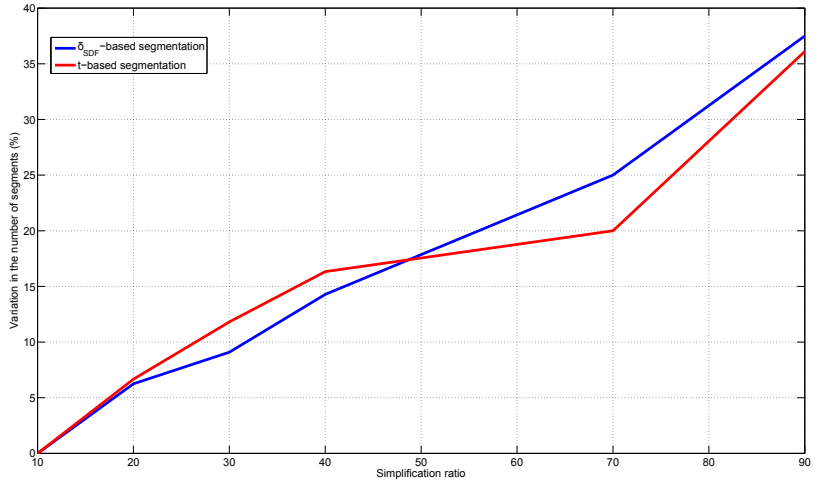
(b) δ_{SDF}-based segmentation

(c) t-based segmentation

Figure 5-21: Segmentation using soft clustering and k-way graph-cut [SSCO08] for a distorted *Elephant* mesh (Figure 5-21(a)) with 89k facets. We applied a random noise with 3% magnitude, as in Section 5.5.3. Figure 5-21(b) shows the results of the segmentation based on δ_{SDF} (44 distinct partitions), while Figure 5-21(c) shows the segmentation based on the thickness estimation t (8 distinct partitions). The latter yields more natural results. Notice that for t, the two tusks of the elephant are in the same segment as the trunk. In the original mesh, these parts are indeed merged together. The k-way graph cut partitioning is then inefficient to separate these features, as it cannot distinguish them using a topological criterion.



(a) Smoothing



(b) Simplification

Figure 5-22: Figure 5-22(a) monitors the average variations in the number of segments (output by the segmentation algorithm) over multiple large meshes in our database, when applying an increasing number of smoothing iterations. Figure 5-22(b) shows the average variation for consecutive simplifications of the input meshes (ratio expressed in % of remaining edges). For this type of distortion, the performances of both methods are very close.

Chapter 6

Optimization-based Framework for Spatial Watermarking

6.1 Introduction

Spatial-domain 3D watermarking approaches directly embed the payload in the coordinates of the mesh vertices without relying on an extension of e.g. the Fourier Transform or the Wavelet Transform for meshes. As presented in Section 3.2, a popular strategy is to watermark the Euclidean distances between the vertices $\mathcal{V}$ and a reference primitive. When this primitive is the mesh center of mass $\mathbf{g}$, these distance are usually referred to as the *vertex norms* and denoted ρ_i. In this chapter, we present the mathematical derivations for a series of enhancing components that extend a blind optimization-based watermarking framework. We first introduce the general mathematical formulation of the watermarking process in Section 6.2, then its state-of-the-art Quadratic Programming (QP) instantiation in Section 6.3. Sections 6.5, 6.6, 6.7 and 6.4 detail the four proposed enhancing components. Each one of those provides the means to improve the traditional robustness vs. imperceptibility watermarking trade-off. They also increase the flexibility of the watermarking system to handle some of the specific issues that arise in mesh processing. While this chapter only deals with the theoretical derivations of these components, the experimental benchmarking results showcasing their practical benefits are grouped in Chapter 7.

6.2 General Optimization Model

The description in this section is a general formulation of the state-of-the-art optimization framework instantiated using QP [HRAM09]. First, we briefly summarize the relevant quantities and notations for this chapter. Vectors are written in column layout by convention, and used indifferently with sets for conciseness.

Vertex $v_i \in \mathcal{V}$ of a triangle surface mesh is associated to the point $\mathbf{p}_i \in \mathbb{R}^3$, whose Cartesian coordinates are the triplet (x_i, y_i, z_i). Matrix $\mathbf{P} \in \mathbb{R}^{3\times n_v}$ contains the Cartesian coordinates of all vertices. The spherical coordinates of $\mathbf{p}_i$ with respect to the center of mass $\mathbf{g}$ of the mesh (also referred to as the mesh barycenter) are denoted by the triplet $(\rho_i, \theta_i, \phi_i)$. The unitary radial and normal vectors for v_i are $\boldsymbol{\rho}_i = \frac{\mathbf{gp}_i}{\|\mathbf{gp}_i\|}$ and $\mathbf{n}_i$, where $\|.\|$ is the Euclidean norm.

Define the histogram of radial distances $\boldsymbol{\rho} = \{\rho_i,\ i \in [\![1, n_v]\!]\}$ with n_B bins and edges evenly spaced by a step Δ. $(\rho^j_{\min}, \rho^j_{\max})$ denotes the boundaries of the jth bin; N_j is the number of

samples within it. All distances ρ_i are normalized in $[0, 1)$ using the affine transform:

$$\bar{\rho}_i = \frac{\rho_i - \rho_{\min}^{B_i}}{\Delta}, \tag{6.1}$$

where B_i denotes the index of the bin associated to the distance ρ_i. $\bar{\boldsymbol{\rho}} \in \mathbb{R}^{n_v}$ denotes the vector of all $\bar{\rho}_i$.

$\mathbf{m}$ is the vector formed by the n_b antipodal bits (in $\{-1, 1\}$) of the watermark payload. $\mathbf{M} \in \mathbb{R}^{n_b \times n_b}$ denotes the corresponding diagonal matrix and $\alpha \in \left(0, \frac{1}{2}\right)$ is the watermark embedding strength. A superscript '$^{\mathrm{w}}$' indicates a watermarked variable.

In the state-of-the-art framework, the watermark embedding corresponds to the minimization of a distortion metric, a.k.a. the fidelity criterion, subject to the constraints of (i) embedding $\mathbf{m}$ in the histogram of $\boldsymbol{\rho}$, a.k.a. the watermark carrier, and (ii) preserving causality.

6.2.1 Cost Function

The cost function ω corresponds to a mesh distortion metric. Minimizing ω is equivalent to minimizing the embedding distortion, with regard to this metric, and corresponds to the fidelity constraint in the watermarking system. To define an optimization model as general as possible, we set the cost function as the squared Euclidean norm of a scalar field $\mathbf{f}(.)$ which depends on: (i) the watermarked vertex positions, (ii) the initial vertex positions, and (iii) the connectivity of the triangular mesh, i.e., the facets.

$$\omega = \|\mathbf{f}(\mathbf{P}^{\mathrm{w}}, \mathbf{P}, \mathcal{F})\|^2 \tag{6.2}$$

Provided that the distortion does not change the mesh connectivity, this formulation encompasses most of the distortion metric definitions reviewed in Section 2.3.

6.2.2 Watermark Constraints

The payload $\mathbf{m}$ is embedded in the mesh by modulating the average value inside the bins of the histogram of $\boldsymbol{\rho}$. For simplicity, the number of bins of the histogram n_B and the payload length n_b are set equal in this section.

To embed the bit $m_j \in \{-1; +1\}$, the average value of $\boldsymbol{\rho}$ inside bin j is raised above the value $\mu^j + \Delta\alpha$ or lowered below $\mu^j - \Delta\alpha$. In the original approach, the radial distances $\boldsymbol{\rho}$ are assumed to be uniformly distributed and μ^j is therefore placed in the middle of the bin to minimize distortion. Moreover, the alteration of the bin averages relies on a power-like *histogram mapping function* which is computationally efficient [CPJ07]. Although our framework slightly differs, we use the same setting $\mu^j = \frac{1}{2}\left(\rho_{\max}^j + \rho_{\min}^j\right)$. The watermarking constraint for bin j can then be written:

$$\begin{cases} \dfrac{1}{N_j}\displaystyle\sum_{i=1}^{n_v} \rho_i^{\mathrm{w}} \delta_{j,B_i} > \rho_{\min}^{B_j} + \Delta\left(\frac{1}{2} + \alpha\right) & \text{if } m_j = 1 \\ \dfrac{1}{N_j}\displaystyle\sum_{i=1}^{n_v} \rho_i^{\mathrm{w}} \delta_{j,B_i} < \rho_{\min}^{B_j} + \Delta\left(\frac{1}{2} - \alpha\right) & \text{otherwise,} \end{cases} \tag{6.3}$$

where $\delta_{i,j}$ denotes the usual Kronecker delta.

Let $\mathbf{W} \in \mathbb{R}^{n_b \times n_v}$ denote the matrix whose coefficients $W_{j,i} = \frac{1}{N_j}\delta_{j,B_i}$ represent the mapping between the mesh vertices and the bins of the histogram, $\bar{\mathbf{t}} \in \mathbb{R}^{n_B}$ the vector with entries 0.5, and $\boldsymbol{\alpha} \in \mathbb{R}^{n_b}$ the vector with entries set to α. The watermark constraints in the general case then

reduce to a set of linear inequalities, written with the normalized variables:

$$\mathbf{M}\bar{\mathbf{t}} + \boldsymbol{\alpha} < \mathbf{M}\mathbf{W}\bar{\boldsymbol{\rho}}^{\mathrm{w}}. \tag{6.4}$$

6.2.3 Causality Constraints

Watermarking detection mainly assumes that the same histogram is reconstructed on the receiver side. The watermarking process should therefore preserve (i) the location of the center of mass, (ii) the mapping between vertices and bins of the histogram, and (iii) the histogram edges. These constraints can be expressed with the following set of equations:

$$\begin{aligned} \mathbf{g}^{\mathrm{w}} &= \mathbf{g}, && (6.5) \\ \forall i \in [\![1, n_v]\!],\ B_i^{\mathrm{w}} &= B_i, && (6.6) \end{aligned} \qquad \begin{aligned} \min \boldsymbol{\rho}^{\mathrm{w}} &= \min \boldsymbol{\rho} \\ \max \boldsymbol{\rho}^{\mathrm{w}} &= \max \boldsymbol{\rho}. \end{aligned} \tag{6.7}$$

6.3 Quadratic Programming Formulation

This general optimization problem has been solved using a QP formulation in the state-of-the-art [HRAM09]. The center of mass $\mathbf{g}$ is the average of all $\mathbf{p}_i$ and vertices are relocated along their radial directions. In this context, the optimization variables are the normalized radial displacements $\delta\bar{\rho}_i^{\mathrm{w}} = \bar{\rho}_i^{\mathrm{w}} - \bar{\rho}_i$ and the cost function is the Square Error (SE), computed as the sum of all squared displacements. As a result, the cost and constraint equations become respectively quadratic and linear and the optimization problem can be solved using efficient large-scale QP solvers [The13].

Let $\boldsymbol{\delta\bar{\rho}}^{\mathrm{w}} = [\delta\bar{\rho}_1^{\mathrm{w}}, \ldots, \delta\bar{\rho}_{n_v}^{\mathrm{w}}]^T$ denote the vector of normalized radial displacements. For the SE metric, the cost function in Eq. (6.2) becomes:

$$\omega = \|\boldsymbol{\delta\bar{\rho}}^{\mathrm{w}}\|^2, \tag{6.8}$$

up to a uniform scaling coefficient. Similarly, the watermark embedding constraints in Eq. (6.4) can be rewritten:

$$\mathbf{M}(\bar{\mathbf{t}} - \mathbf{W}\bar{\boldsymbol{\rho}}) + \boldsymbol{\alpha} < \mathbf{M}\mathbf{W}\boldsymbol{\delta\bar{\rho}}^{\mathrm{w}}. \tag{6.9}$$

The left hand-side of the equation corresponds to the difference between the initial bin averages and the target averages encoding the desired payload. The right hand-side accounts for the variations in the bin averages due to the relocation of the vertices with $\boldsymbol{\delta\bar{\rho}}^{\mathrm{w}}$.

Since $\mathbf{g}$ is the discrete center of mass, the barycenter stability constraint in Eq. (6.5) is equivalent to constraining that all radial displacements average to the null vector:

$$\sum_{i=1}^{n_v} \delta\bar{\rho}_i^{\mathrm{w}} \begin{pmatrix} \cos\theta_i \cos\phi_i \\ \sin\theta_i \cos\phi_i \\ \sin\phi_i \end{pmatrix} = \mathbf{0}. \tag{6.10}$$

This constraint can be advantageously rewritten using the Jacobian matrix of $\mathbf{p}_i = (x_i, y_i, z_i)$ with respect to $\bar{\rho}_i$:

$$\mathbf{J}_{\bar{\rho}_i}^{\mathbf{p}_i}(\bar{\rho}_i) = \Delta \begin{pmatrix} \cos\theta_i \cos\phi_i \\ \sin\theta_i \cos\phi_i \\ \sin\phi_i \end{pmatrix}. \tag{6.11}$$

By denoting $\mathbf{J}_{\bar{\boldsymbol{\rho}}}^{\mathbf{P}}(\bar{\boldsymbol{\rho}})$ the 3 by n_v matrix whose ith column is $\mathbf{J}_{\bar{\rho}_i}^{\mathbf{p}_i}(\bar{\rho}_i)$, Eq. (6.10) indeed becomes equivalent to:

$$\mathbf{J}_{\bar{\boldsymbol{\rho}}}^{\mathbf{P}}(\bar{\boldsymbol{\rho}})\boldsymbol{\delta\bar{\rho}}^{\mathrm{w}} = \mathbf{0}. \tag{6.12}$$

The stability of the histogram is guaranteed by two additional constraints. The first and last bins are not considered during embedding to preserve the extreme values of $\boldsymbol{\rho}$ (Eq. (6.7)). In practice, the number of bins of the histogram is simply set to $n_B = n_b + 2$ to yield a one-to-one mapping between the n_b watermark payload bits and the remaining bins. Then, to keep the vertex–bin mapping unaltered (Eq. (6.6)), the optimization variable $\boldsymbol{\delta\bar{\rho}}^{\mathrm{w}}$ is bounded using a bin separation offset $\beta \in \mathbb{R}^+$:

$$\forall i \in [\![1, n_v]\!],\ B_i \notin \{1, n_B\},\ \beta - \bar{\rho}_i \leq \delta\bar{\rho}_i^{\mathrm{w}} < 1 - \beta - \bar{\rho}_i. \tag{6.13}$$

Finally, the solution returned by the QP solver is used to build the watermarked mesh:

$$\forall i \in [\![1, n_v]\!],\ \mathbf{p}_i^{\mathrm{w}} = \mathbf{p}_i + \Delta\delta\bar{\rho}_i^{\mathrm{w}}\boldsymbol{\rho}_i. \tag{6.14}$$

On the receiver side, the payload extraction reduces to constructing the histogram of distances $\boldsymbol{\rho}$ and comparing the normalized average inside each bin to the value 0.5. In other words, denoting by $\bar{c}_j$ the normalized average inside bin j, the estimated bit is given by: $\hat{w}_j = \mathrm{sign}(\bar{c}_j - \frac{1}{2})$.

6.4 Spread-Transform Formulation

Spread Transform (ST) is routinely used in watermarking. It amounts to projecting the carrier signal onto a pseudo-random sequence $\mathbf{s} = [s_0 \dots s_{k-1}]^T$, i.e., to computing the inner product between the two sequences, prior to applying some embedding algorithm. The spreading sequence typically has zero-mean and unit norm with samples drawn from either a normal or uniform distribution. If the watermark modulation is binary, it is simply a regular spread-spectrum watermarking [IKLS97]. If the embedding mechanism instead relies on binning, it is the so-called Spread Transform Dither Modulation (STDM) [CW99]. While the latter has great potential in theory, its practical performances are closely related to the distribution of the carrier signal.

Prior work in 3D watermarking considered applying STDM directly to the radial distances of the vertices [DHM10]. In this case, the watermarking system is highly sensitive to the slightest changes in the ordering of the distances in $\boldsymbol{\rho}$. To avoid such instability, one may be tempted to apply STDM to the average radial distances considered in the QP framework to benefit from the stability inherited from the integration within the bins of the histogram. However, empirical observations revealed that the distribution of these values is highly concentrated around 0.5 [CPJ07], thereby making them ill-suited for binning schemes. For large quantization steps, a single bin is used; for small quantization steps, robustness performances are worse than for the baseline system. Still, it may be useful to keep the ST in an attempt to diversify the solution space of the QP framework. The system is then quite close to Spread Spectrum (SS) except that the individual displacement for each vertex is not given by a generic equation but is instead driven by the solver that optimizes the QP problem.

6.4.1 Framework Modification

Adding ST to the QP framework only modifies the watermark constraints in the embedding process; all the other constraints stay the same. First, the number of bins in the histogram is multiplied by the spreading factor k, i.e., $n_B = k.n_b$ (omitting the extreme bins at both ends), and the average normalized radial distances are partitioned into n_b consecutive carrier sequences with k values. Let $\tau = 0.5\sum_{i=0}^{k-1} s_i$ denote the projection of $\bar{\mathbf{t}}$ onto the spreading sequence. Projecting each carrier sequence onto $\mathbf{s}$ yields n_b values c_i and, depending on the bit m_j, c_j is either raised above $\tau + \alpha$ or lowered below $\tau - \alpha$. Formalizing these observations, the revised QP watermark constraint

becomes:

$$\mathbf{M\Phi}\left(\bar{\mathbf{t}} - \mathbf{W}\bar{\boldsymbol{\rho}}\right) + \boldsymbol{\alpha} < \mathbf{M\Phi W}\boldsymbol{\delta}\bar{\boldsymbol{\rho}}^{\mathrm{w}}, \tag{6.15}$$

where $\mathbf{\Phi}$ denotes a $n_b \times n_B$ matrix, whose rows contain k values of the spreading sequence $\mathbf{p}$, shifted by a multiple of k. For $k = 1$, $\mathbf{\Phi}$ is the identity matrix, and the constraint simplifies to Equation (6.9).

At decoding, the averages inside the n_B bins of the histogram are computed, the n_b sequences are projected back onto $\mathbf{p}$ and the resulting values are compared with τ.

6.5 Integral Centroids

The discrete center of mass used in the original approach is, by definition, neither robust to non-uniform nor anisotropic remeshing. This limitation calls for incorporating more integral formulations of the center of mass into the framework in an attempt to improve robustness. However, integral formulations involve non-linear and neighborhood-dependent weighting functions. As a result, Eq. (6.5) and (6.12) are no longer equivalent, and the mathematical model cannot be formulated as a QP problem.

6.5.1 Derivation of the Stability Constraint

Let $f \in \mathcal{F}$ be a mesh facet. Given a per-facet weight $w(f) \in \mathbb{R}^+$ and center $\mathbf{g}(f) \in \mathbb{R}^3$, an integral mesh center of mass is defined, in general, as a weighted sum over all facets of a 3D mesh[1]:

$$\mathbf{g} = \frac{1}{w_0} \sum_{f \in \mathcal{F}} w(f)\mathbf{g}(f), \tag{6.16}$$

where $w_0 = \sum_{f \in \mathcal{F}} w(f)$ is a normalization factor. The ith column of the Jacobian matrix $\mathbf{J}^{\mathbf{g}}_{\bar{\boldsymbol{\rho}}}(\bar{\boldsymbol{\rho}})$ is:

$$\mathbf{J}^{\mathbf{g}}_{\bar{\rho}_i}(\bar{\rho}_i) = \mathbf{J}^{\mathbf{g}}_{\mathbf{p}_i}(\mathbf{p}_i)\mathbf{J}^{\mathbf{p}_i}_{\bar{\rho}_i}(\bar{\rho}_i). \tag{6.17}$$

For the discrete barycenter, $\mathbf{J}^{\mathbf{g}}_{\mathbf{p}_i}(\mathbf{p}_i)$ simplifies to $\frac{1}{3}\mathbf{I}_3$, which subsequently leads to Eq. (6.12). In the general case, assuming that $\mathbf{g}(f)$ and $w(f)$ only depend on the vertices in the facet f, $\mathbf{J}^{\mathbf{g}}_{\mathbf{p}_i}(\mathbf{p}_i)$ is written:

$$\mathbf{J}^{\mathbf{g}}_{\mathbf{p}_i}(\mathbf{p}_i) = \frac{1}{w_0}\left(\sum_{f \in \mathcal{N}_1^{\mathcal{F}}(v_i)} [\mathbf{g}(f) - \mathbf{g}] \frac{\partial w(f)}{\partial \mathbf{p}_i}(\mathbf{p}_i) + \sum_{f \in \mathcal{N}_1^{\mathcal{F}}(v_i)} w(f) \frac{\partial \mathbf{g}(f)}{\partial \mathbf{p}_i}(\mathbf{p}_i) \right), \tag{6.18}$$

where $\mathcal{N}_1^{\mathcal{F}}(v_i)$ is the set of facets in the 1-ring neighborhood around v_i.

With a first-order development, Eq. (6.5) can be linearized:

$$\mathbf{J}^{\mathbf{g}}_{\bar{\boldsymbol{\rho}}}(\bar{\boldsymbol{\rho}})\boldsymbol{\delta}\bar{\boldsymbol{\rho}}^{\mathrm{w}} = \mathbf{0}, \tag{6.19}$$

where the i^{th} column of the matrix $\mathbf{J}^{\mathbf{g}}_{\bar{\boldsymbol{\rho}}}(\bar{\boldsymbol{\rho}})$ is given by:

$$\mathbf{J}^{\mathbf{g}}_{\bar{\rho}_i}(\bar{\rho}_i) = \left[\sum_{f \in \mathcal{N}_1^{\mathcal{F}}(v_i)} [\mathbf{g}(f) - \mathbf{g}] \frac{\partial w(f)}{\partial \mathbf{p}_i}(\mathbf{p}_i) + \sum_{f \in \mathcal{N}_1^{\mathcal{F}}(v_i)} w(f) \frac{\partial \mathbf{g}(f)}{\partial \mathbf{p}_i}(\mathbf{p}_i) \right] \mathbf{J}^{\mathbf{p}_i}_{\bar{\rho}_i}(\bar{\rho}_i). \tag{6.20}$$

[1] Alternate formulations use a sum over the vertices, for which similar equations than the following ones can be derived.

Eq. (6.19) provides a generalization of the center of mass stability constraint that is still linear in the variable $\boldsymbol{\delta\bar{\rho}}^{\mathrm{w}}$. It grants the flexibility to use more integral barycenter definitions without losing the benefits of the QP formulation, as all the other parts of the framework remain identical.

Surface-weighted Barycenter

Let $\left(\mathbf{p}_0^f, \mathbf{p}_1^f, \mathbf{p}_2^f\right)$ denote the vertex locations in facet f. The surface weights are defined by:

$$w(f) = \frac{1}{2}\left\|(\mathbf{p}_1^f - \mathbf{p}_0^f) \times (\mathbf{p}_2^f - \mathbf{p}_0^f)\right\|. \tag{6.21}$$

The facet center and its partial derivatives are defined by:

$$\mathbf{g}(f) = \frac{1}{3}\left(\mathbf{p}_0^f + \mathbf{p}_1^f + \mathbf{p}_2^f\right), \tag{6.22}$$

$$\frac{\partial \mathbf{g}(f)}{\partial \mathbf{p}_i^f}\left(\mathbf{p}_i^f\right) = \frac{1}{3}\mathbf{I}_3. \tag{6.23}$$

The gradient of the weights is thus defined by:

$$\frac{\partial w(f)}{\partial \mathbf{p}_i^f}\left(\mathbf{p}_i^f\right) = \frac{1}{2}\left[\left(\mathbf{p}_{i+2 \bmod 3}^f - \mathbf{p}_{i+1 \bmod 3}^f\right)^{\perp}\right]^T, \tag{6.24}$$

where $^{\perp}$ denotes a $\pi/2$ counter-clockwise rotation in the triangle plane.

Volume-weighted Barycenter

Let $\mathbf{o}$ represent an arbitrary reference point. For simplicity, $\mathbf{o}$ is chosen as the origin of the coordinate system. The facet f is associated to the tetrahedron $\left(\mathbf{o}, \mathbf{p}_0^f, \mathbf{p}_1^f, \mathbf{p}_2^f\right)$, and is assigned a weight $w(f)$ equating to its signed volume:

$$w(f) = \frac{1}{6}\det\left(\mathbf{p}_0^f, \mathbf{p}_1^f, \mathbf{p}_2^f\right). \tag{6.25}$$

The facet center and its partial derivatives are given by:

$$\mathbf{g}(f) = \frac{1}{4}\left(\mathbf{p}_0^f + \mathbf{p}_1^f + \mathbf{p}_2^f\right), \tag{6.26}$$

$$\frac{\partial \mathbf{g}(f)}{\partial \mathbf{p}_i^f}\left(\mathbf{p}_i^f\right) = \frac{1}{4}\mathbf{I}_3. \tag{6.27}$$

The gradient of the weights is thus given by:

$$\frac{\partial w(f)}{\partial \mathbf{p}_i^f}\left(\mathbf{p}_i^f\right) = \left[\mathbf{p}_{i+1 \bmod 3}^f \times \mathbf{p}_{i+2 \bmod 3}^f\right]^T. \tag{6.28}$$

6.6 Arbitrary Relocation Directions

In the original framework, the displacements of the vertices are restricted to the radial directions. When $\boldsymbol{\rho}_i \cdot \mathbf{n}_i \approx 1$, the watermark effectively alters the geometry of the surface by relocating vertices

along the normal direction. In contrast, when the radial direction lies within the tangent plane (i.e., when $\boldsymbol{\rho_i} \cdot \mathbf{n}_i \approx 0$), the embedding may be ineffective. For instance, in the case of coplanar vertices, points are only moved on the surface of the object and no geometric change is introduced. While such alteration may yield very low distortion according to most mesh distortion metrics, it also produces weak watermarks that could be easily erased, for instance after resampling.

In summary, the robustness versus imperceptibility trade-off is affected by the selected direction of alteration. To possibly leverage on this degree of freedom, and also provide greater flexibility, the optimization variables (modified during embedding) and the radial distances (carrying the watermark) are dissociated. This amounts to defining a vector field $\mathbf{u}_i$ that is used instead of $\boldsymbol{\rho}_i$ to relocate the vertices. The optimization variable δr_i^{w} accounts for the signed displacement of $\mathbf{p}_i$ along the preset directions $\mathbf{u}_i$.

6.6.1 Modifications to the QP Framework

This change of strategy translates to a number of modifications in the formulation of the watermarking process. The cost function is lightly modified; $\boldsymbol{\delta r}^{\mathrm{w}}$ being substituted to $\boldsymbol{\delta \bar{\rho}}^{\mathrm{w}}$ in Eq. (6.8). To build the watermarked mesh, Eq. (6.14) is updated to:

$$\forall i \in [\![1, n_v]\!],\ \mathbf{p}_i^{\mathrm{w}} = \mathbf{p}_i + \delta r_i^{\mathrm{w}} \mathbf{u}_i. \tag{6.29}$$

Using $\cos\psi_i = \boldsymbol{\rho}_i \cdot \mathbf{u}_i$, the linear expansion of the radial distance ρ_i^{w} is given by:

$$\rho_i^{\mathrm{w}} = \rho_i + \delta r_i^{\mathrm{w}} \cos\psi_i + \frac{1}{2\rho_i}(\delta r_i^{\mathrm{w}})^2 \sin^2\psi_i + o((\delta r_i^{\mathrm{w}})^2). \tag{6.30}$$

In the case of radial embedding ($\psi_i = 0$), the terms above the first order are null and the variables $\delta\bar{\rho}_i^{\mathrm{w}}$ and δr_i^{w} are equal (up to the bin scale Δ).

Without the ST described in Section 6.4, the watermark embedding constraints (Eq. (6.9)) are now given by:

$$\mathbf{M}(\bar{\mathbf{t}} - \mathbf{W}\bar{\boldsymbol{\rho}}) + \boldsymbol{\alpha} < \mathbf{M}\mathbf{W}\boldsymbol{\Psi}\boldsymbol{\delta r}^{\mathrm{w}}, \tag{6.31}$$

where $\boldsymbol{\Psi}$ denote the diagonal matrix of $\cos\psi_i$ scaled by Δ^{-1}. Intuitively, this matrix indicates how much the normalized relocation distortion is actually used to reach the watermarking target.

The previous watermark constraint in the ST component and the one indicated in Eq. (6.31) are compatible and are combined in a generalized watermark constraint:

$$\mathbf{M}\boldsymbol{\Phi}\,(\bar{\mathbf{t}} - \mathbf{W}\bar{\boldsymbol{\rho}}) + \boldsymbol{\alpha} < \mathbf{M}\boldsymbol{\Phi}\mathbf{W}\boldsymbol{\Psi}\boldsymbol{\delta\bar{\rho}}^{\mathrm{w}}. \tag{6.32}$$

The constraint on the barycenter is obtained through substituting $\mathbf{J}^{\mathbf{g}}_{\bar{\boldsymbol{\rho}}}(\bar{\boldsymbol{\rho}})$ for $\mathbf{J}^{\mathbf{g}}_{\boldsymbol{\delta r}}(\boldsymbol{\delta r})$ in Eq. (6.19), yielding:

$$\mathbf{J}^{\mathbf{g}}_{\boldsymbol{\delta r}}(\mathbf{0})\boldsymbol{\delta r}^{\mathrm{w}} = \mathbf{0}. \tag{6.33}$$

Applying the chain-rule, the ith column of $\mathbf{J}^{\mathbf{g}}_{\boldsymbol{\delta r}}(\mathbf{0})$ is $\mathbf{J}^{\mathbf{g}}_{\mathbf{p}_i}(\mathbf{p}_i)\mathbf{u}_i$, where the first term is given in Eq. (6.18).

Let us define:

$$\boldsymbol{\Gamma}^i = \frac{1}{\cos\psi_i}\begin{pmatrix}\Delta(1-\beta) + \rho_{\min}^{B_i} - \rho_i \\ \Delta\beta + \rho_{\min}^{B_i} - \rho_i\end{pmatrix} \tag{6.34}$$

Plugging the linearization in the histogram stability constraint (Eq. (6.13)) then yields:

$$\forall i \in [\![1, n_v]\!],\ B_i \notin \{1, n_B\}, \quad \min(\boldsymbol{\Gamma}_1^i, \boldsymbol{\Gamma}_2^i) \leq \delta r_i^{\mathrm{w}} \leq \max(\boldsymbol{\Gamma}_1^i, \boldsymbol{\Gamma}_2^i) \tag{6.35}$$

Geometrically, these linearized constraints approximate the spherical bin boundaries as tangent lines (cf. Fig. 6-1). The smaller $|\cos\psi_i|$, the larger the approximation error, as indicated by Eq. (6.30). Ill-defined numerical cases occur for $|\cos\psi_i| \approx 0$.

Besides, boundaries can be precomputed more accurately, since they correspond to intersections between spheres (bin boundaries) and lines (relocation directions) which are fixed throughout the optimization process. The mathematical derivations and the algorithm to compute accurate boundaries without ill-defined numerical cases are detailed next.

6.6.2 Boundary Constraints Derivation

Taking the square of Eq. (6.13) and using the equality:

$$(\rho_i^{\mathrm{w}})^2 = (\delta r_i^{\mathrm{w}})^2 + \rho_i^2 + 2\delta r_i^{\mathrm{w}} \rho_i \cos\psi_i,$$

the boundary constraints can be written as second-degree inequalities with respect to δr_i^{w}:

$$\delta r_i^{\mathrm{w}\,2} + 2\gamma_i \delta r_i^{\mathrm{w}} - L_i \;>\; 0, \tag{6.36}$$
$$\delta r_i^{\mathrm{w}\,2} + 2\gamma_i \delta r_i^{\mathrm{w}} - U_i \;<\; 0, \tag{6.37}$$

where $\gamma_i = \rho_i \cos\psi_i$, $L_i = \left(\rho_{\min}^{B_i} + \Delta\beta\right)^2 - \rho_i^2$, and $U_i = \left(\rho_{\max}^{B_i} - \Delta\beta\right)^2 - \rho_i^2$. For compactness, the offset β denotes the rescaled offset $\Delta\beta$ in this section.

A first degenerate case occurs when $\mathbf{p}_i$ is outside the sphere $\mathcal{S}_{\max}(\mathbf{g}, \rho_{\max}^{B_i} - \beta)$ and $\mathbf{u}_i \cap \mathcal{S}_{\max} = \emptyset$. Inequality (6.37) indeed has no solution (label C1 in Algorithm 1). In other words, this corresponds to using a bin separation offset β such that $\mathbf{p}_i$, which initially lies in the upper part of bin B_i, has to be relocated farther away from the upper bin boundary $\rho_{\max}^{B_i}$ to enforce the separation offset. But $\mathbf{p}_i$ can only be relocated along a direction which does not enable achieving this constraint (empty intersection). In practice, this case is handled by resetting $\mathbf{u}_i$ to $\boldsymbol{\rho}_i$.

Discarding this degenerate case, if $\mathbf{p}_i$ is outside the sphere $\mathcal{S}_{\min}(\mathbf{g}, \rho_{\min}^{B_i} + \beta)$ and $\mathbf{u}_i \cap \mathcal{S}_{\min} = \emptyset$ (for instance with direction $\mathbf{u}_2$ in Figure 6-1), the constraints reduce to Inequality (6.37) (label C2 in Algorithm 1) and thus become linear in δr_i^{w}.

In the other cases, the constraints correspond to the union of two disjoint segments (direction $\mathbf{u}_1$ in Figure 6-1). If $\mathbf{p}_i$ is already within one segment (label C3), the constraints are approximated with this single segment. If $\mathbf{p}_i$ is within the sphere $\mathcal{S}_{\min}$ (label C4), the constraints are approximated using the segment closest to $\mathbf{p}_i$. This case is symmetrical to the first degenerate case (label C1). However, there is always at least one intersection between the relocation direction and the lower bin boundary sphere, offset with β. Therefore, the boundaries are always well-defined.

The constraints on the histogram are thus the only ones which are not approximated when extending the QP framework to non-radial relocation directions.

6.6.3 Alteration Vector Fields

As mentioned earlier, using the normal directions as the alteration vector field ($\forall i \in [\![1, n_v]\!], \mathbf{u}_i = \mathbf{n}_i$), may provide additional robustness but is also likely to incur prohibitive embedding distortion. As a result, it may not provide a better trade-off between robustness and fidelity. The new flexibility with respect to the alteration vector field should be regarded as a means to perceptually shape the watermark through adjusting the 'direction' of the embedded watermark according to the local properties of the content. For instance, prior work clearly highlighted that alterations in rough areas of the mesh (a.k.a. textured regions) are significantly less noticeable than in smooth

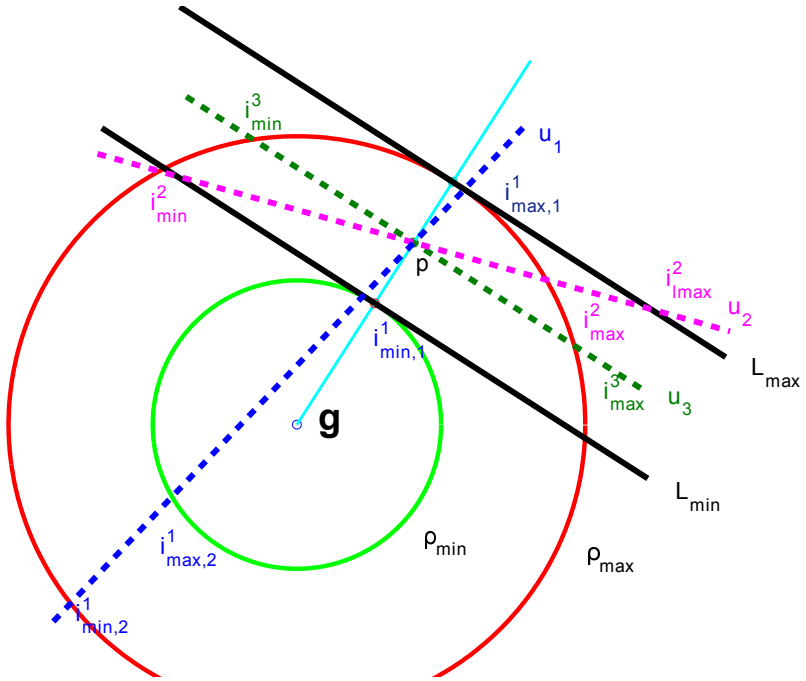


Figure 6-1: Simplified configurations to compute the boundary constraints. Using 2D projections and setting $\beta = 0$, three cases are shown for $\mathbf{p}$ with directions $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$. $\mathbf{p}$ is constrained to stay outside $\mathcal{S}_{\min}(\mathbf{g}, \rho_{\min})$ and inside $\mathcal{S}_{\max}(\mathbf{g}, \rho_{\max})$. Linearized constraints correspond to the lines $L_{\min}$ and $L_{\max}$, and boundaries are computed using their intersections with the direction of alteration. $\mathbf{u}_3$ is an ill-defined case ($|\cos\psi| = 0$). The smaller $|\cos\psi|$, the larger the approximation error ($\mathbf{i}^2_{\text{lmax}}$ vs. $\mathbf{i}^2_{\text{max}}$). Using the non-linearized constraints, two configurations are depicted: the boundary constraints can actually be linear ($\mathbf{u}_2$ and $\mathbf{u}_3$, C2 in Algorithm 1); otherwise $\mathbf{p}$ can be within two disjoint segments ($\mathbf{u}_1$) and the relocation is restricted to $[\mathbf{i}^1_{\text{min},1}, \mathbf{i}^1_{\text{max},1}]$ (C3 in Algorithm 1).

regions [CLL+13]. To leverage on this *masking effect*, the alteration vector field may favor radial alterations in rough areas while limiting displacements in the tangent plane, i.e. $\mathbf{u}_i = \boldsymbol{\rho}_i - (\boldsymbol{\rho}_i \cdot \mathbf{n}_i)\mathbf{n}_i$, in smooth regions in an attempt to mitigate distortion.

6.7 Perceptual Shaping

In the original QP framework, the embedding distortion is minimized with regard to the Square Error (SE) metric, which attributes the same weight to all alterations over the mesh. Despite its simplicity, this metric is known to be only mildly correlated with the distortion perceived by human observers [CLL+13]. In line with related work for other types of content, this limitation calls for incorporating perceptual metrics in the optimization framework. In contrast with the perceptual shaping mechanism presented earlier, the objective here is not to modify the direction of alteration but rather to adjust the magnitude of the displacement along the predefined direction according to some local properties. For instance, adapting the magnitude of the embedding alteration using the masking effect indeed can yield significant improvements in 3D watermarking [KBT10].

Various 3D metrics have been investigated and the ones showcasing the highest correlation with perceived distortion are based on multi-scale analysis of the roughness [CGEB07] or on the mesh curvatures [Lav11]. Unfortunately, these quantities are highly non-linear and cannot be readily

Algorithm 1 Boundary constraints approximation

procedure BOUNDARYCONSTRAINTS(radius ρ; upper-bound U; lower-bound L; projection γ)
  if $U + \gamma^2 < 0$ **then** ▷ C1
    return Boundaries $B = \emptyset$
  end if
  $(S_1, S_2) = \left(-\gamma - \sqrt{\gamma^2 + U}; -\gamma + \sqrt{\gamma^2 + U}\right)$
  if $L + \gamma^2 < 0$ **then** $B = [S_1, S_2]$ ▷ C2
  else
    $(S_3, S_4) = \left(-\gamma - \sqrt{\gamma^2 + L}; -\gamma + \sqrt{\gamma^2 + L}\right)$
    if $S_1 S_3 \leq 0$ **then** $B = [S_1, S_3]$ ▷ C3
    else if $S_4 S_2 \leq 0$ **then** $B = [S_4, S_2]$ ▷ idem
    else if $|S_3| < |S_4|$ **then** $B = [S_1, S_3]$ ▷ C4
    else $B = [S_4, S_2]$ ▷ idem
    end if
  end if
  return Boundaries B
end procedure

plugged into the QP framework. Still, a few existing metrics achieve better results than the SE metric and can be expressed as quadratic functions of the vertex displacements $\boldsymbol{\delta r}^{\mathrm{w}}$.

6.7.1 QEM-based Shaping

The Quadric Error Metric (QEM) [GH97] has been shown to improve the control over the embedding distortion in 3D watermarking [LB13]. In this case, the cost function becomes:

$$\omega = \lambda \underbrace{\|\boldsymbol{\delta r}^{\mathrm{w}}\|^2}_{\mathrm{SE}} + (1-\lambda) \underbrace{(\mathbf{Q}\boldsymbol{\delta r}^{\mathrm{w}})^T (\mathbf{Q}\boldsymbol{\delta r}^{\mathrm{w}})}_{\mathrm{QEM}}, \tag{6.38}$$

where $\lambda \in [0, 1]$ is a mixing parameter used to trade SE for QEM. The matrix $\mathbf{Q} \in \mathbb{R}^{n_v \times n_v}$ is diagonal and its i^{th} entry is the sum of the projections of the relocation direction $\mathbf{u}_i$ onto the normal $\mathbf{n}_f$ of the facets around vertex v_i, i.e., $\sum_{f \in \mathcal{N}_1^{\mathcal{F}}(v_i)} \mathbf{u}_i \cdot \mathbf{n}_f$. The motivation for the QEM to only take into account the distortion along the normal direction is that alterations in the tangent plane are less noticeable.

6.7.2 Laplacian-based Shaping

Alternatively, following a thread of research in mesh compression, the local roughness can be assimilated to the difference $\mathbf{d}_i$ between a vertex position and its position after smoothing using the Laplacian matrix $\mathbf{L}$ [KG00b]. Based on this rationale, it is possible to define a distortion metric that sums the squared magnitude of the difference in local roughness $\|\mathbf{d}_i^{\mathrm{w}} - \mathbf{d}_i\|^2$ over all vertices. Again, this Laplacian-based metric is usually combined with the SE through a blending using $\lambda \in [0, 1]$. Several discretizations of the Laplacian matrix have been proposed but only the ones based solely on the connectivity of the mesh (referred to as *combinatorial Laplacian*) can be integrated into the QP framework without further approximations. In this case, the cost function

is a quadratic function in the optimization variables, written as:

$$\omega = \lambda \|\boldsymbol{\delta r}^{\mathrm{w}}\|^2 + (1-\lambda) \sum_{i=1}^{n_v} \left\| \delta r_i^{\mathrm{w}} \mathbf{u}_i - \frac{1}{|\mathcal{N}_1(v_i)|} \sum_{v_j \in \mathcal{N}_1(v_i)} \delta r_j^{\mathrm{w}} \mathbf{u}_j \right\|^2 . \tag{6.39}$$

6.7.3 Roughness-driven Shaping

It is straightforward to directly scale the individual terms of the SE cost function by some weights $\mathbf{w} = \{w_i,\, i \in [\![1, n_v]\!]\}$ in an attempt to obtain a perceptually-driven weighted Square Error (wSE):

$$\mathrm{wSE} = \sum_{i=0}^{n_v-1} w_i \delta r_i^{\mathrm{w}2}. \tag{6.40}$$

More specifically, based on previous findings, it makes sense to tie these weights to the local roughness in order to harden the fidelity constraint in smooth areas of the object and, conversely, to relax it in rough regions. This has already been explored in a previous work for watermarking radial distances [DHM10]. In this Section, the local roughness χ_i is estimated at each vertex and is derived from statistics relating to the principal curvatures [Lav09]. Empirically, these roughness values have a distribution with few large outliers and small standard deviation and therefore need to be post-processed. First, outlying values are clipped to a minimal and maximal threshold, set to 5% of the lowest and largest values. Next, all values are (affine) mapped to obtain $\bar{\boldsymbol{\chi}} = \{\bar{\chi}_i,\, i \in [\![1, n_v]\!]\}$ in $[0, 1]$. Finally, the weights $\mathbf{w}$ are set to $\mathbf{1} - \bar{\boldsymbol{\chi}}$.

To ensure that vertices are not relocated to arbitrarily large distances, the QP cost function is still defined as a λ-driven linear trade-off between the SE and the wSE metric. In this case, ω simply writes:

$$\omega = \lambda \|\boldsymbol{\delta r}^{\mathrm{w}}\|^2 + (1-\lambda) \sum_{i=0}^{n_v-1} w_i \delta r_i^{\mathrm{w}2} = \sum_{i=1}^{n_v} (1 + (\lambda - 1)\bar{\chi}_i) \left(\delta r_i^{\mathrm{w}}\right)^2 . \tag{6.41}$$

6.8 Conclusion

In this chapter, we generalized a previous framework for 3D mesh watermarking, where the embedding process is formulated as a QP problem. More specifically, we described four different extensions to the baseline system: (i) the use of a state-of-the-art ST embedding function, (ii) the revision of the mathematical framework to support integral definitions of the center of mass of a mesh, (iii) the relaxation of the constraint on the direction of alteration to allow displacements deviating from the radial direction, and (iv) the integration of perceptual components in the cost function to better account for human perception during the minimization process. The resulting flexibility allows various combinations of the different components. Chapter 7 provides some practical implementation details and focuses on a thorough benchmarking campaign to investigate the added value of these modifications with respect to the common fidelity–robustness trade-off.

Chapter 7

Benchmarking of the Optimization-based Framework and its Extensions

7.1 Introduction

The previous chapter details the mathematical derivations of four extensions of a Quadratic Programming (QP) formulation [HRAM09] for 3D watermarking. In Section 7.2, we review some practical details regarding these extensions. Section 7.3 then presents the benchmarking protocol used to assess their performance, which is an extension of the one in Chapter 4.2 that only dealt with content adaptation transforms. Sections 7.4, 7.5, 7.7 and 7.6 then summarize the thorough benchmarking of the four components in terms of robustness and fidelity. We also investigate some specific impacts on the embedding distortions. Finally, Section 7.8 draws conclusions and proposes future research directions for the QP-based watermarking.

7.2 Implementation Details

The extensions of the QP framework have mainly been implemented with MATLAB [The13]. The Laplacian-based cost function (described in Section 6.7.2), the roughness estimator (basis estimator for the cost function in Section 6.7.3), and the perceptually-correlated metric used in the benchmark are implemented in C++. Figure D-2 in Section D.2 shows some snapshots of watermarked models.

The four new extensions can be partitioned into two groups, depending on whether or not they introduce first-order approximations in the QP formulation. For the Spread Transform (ST) embedding (Section 6.4) and the alternate cost functions that are expected to better account for the Human Visual System (HVS) (Section 6.7), no approximation is made. Conversely, the extension to using non-radial relocation directions (Section 6.6) and an integral formulation of the center of mass (Section 6.5) lead to approximations of most of the mathematical constraints. When using either one of these two components, the solution found by the solver is no longer exact, and the mathematical model and the practical implementation are no longer equivalent.

To guarantee watermark effectiveness, it is therefore necessary to perform the embedding procedure iteratively. After each embedding iteration, the payload is decoded and the Bit Error Rate (BER) is measured. This process continues until either the BER reaches 0% or the number of iterations reaches a maximal value, set to 10 by default. Still, our empirical observation during our experiments is that at most two iterations are needed to achieve embedding.

A similar problem already occurs in the state-of-the-art approach, as well as when using, e.g., the ST embedding extension. In theory the computed vertex positions $\mathbf{P}^{\mathrm{w}}$ is an exact solution. In practice however, when formatting the results in a file to transmit the watermarked asset, some quantization errors are introduced, due to the limited precision of floating-point numbers. For instance, in the Object File Format (OFF), numbers are commonly represented with only four of five decimal digits. This limited precision may lead to an imperfect decoding of the watermark in especially delicate cases, for instance when a payload bit depends on only a few vertex locations.

7.3 General Setup

Several variants of the proposed watermarking framework are evaluated and compared with the original QP approach. In a first series of experiments, we solely focus on the ST extension (Section 7.4), and each variant corresponds to a different spreading length k. The distortion alignment procedure to set some of the parameters of the framework is in this case fully detailed. The embedding distortion is indeed calibrated to guarantee a fair comparison between all the different variants and the baseline QP. Furthermore, the ST component mainly impacts the number of bins in the histogram, while the others do not. For this reason, the capacity of the watermarking system in the ST benchmark is set conservatively to $n_b = 16$. The next series of experiments benchmark the three remaining extensions and correspond to a combination of: (i) a center of mass, (ii) a direction of relocation, and (iii) a cost function.

In our experiments, we consider a database of thirteen 3D models (detailed in Table D.1 in Appendix), that provides a representative diversity of shapes. The benchmark attacks are similar to the ones used in Chapter 4. For each attack, the experimental procedure consists of: (i) generating six random payloads; (ii) watermarking all models once with each payload; (iii) creating ten attacked versions of the watermarked meshes (when an attack yields non-deterministic output); and (iv) detecting the payload from the resulting attacked meshes. For each variant in the framework, and for a given attack strength, the reported BER is the median value over the 780 detection trials resulting from this procedure.

7.4 Benchmark of the Spread-Transform Component

In this series of experiments, multiple spreading lengths are surveyed, including $k = 1$ (baseline), 2, 3, 4, 5, 6, 8, and 16. Using a fixed payload size of 16 bits means that the number of bins in the histogram of radial distances increases with k. For this reason, the payload size in this series of experiments is smaller than in the following series. Indeed, for $k = 16$, the histogram is made-up of 256 bins, which already corresponds to a high embedding rate for objects with a small number of vertices.

7.4.1 Embedding Distortion with ST

The symmetric Hausdorff distance is a popular objective geometric error metric in the watermarking community. However, its computation time is rather prohibitive and precludes large scale benchmarking campaigns. Since the Root Mean Square (RMS) error showcases similar correlation with the perceived distortion, we use it to assess the objective geometric embedding distortion and express its result as a percentage with respect to the length of the space diagonal of the bounding box. The Mesh Structural Distortion Measure (MSDM) [LDD+06] is reported to provide superior perceptually-correlated distortion estimations compared with other metrics [CLL+13]. Its

extension to a multi-scale approach yields even better results but at the cost of a large increase in complexity and computation time [Lav11]. In our experiments, we use the MSDM to assess the perceptual embedding distortion and its values are within $[0, 1]$. As advocated in an existing 3D watermark benchmark [WLD+10], simultaneously measuring both types of distortion enables a better assessment of the watermark system fidelity.

The fidelity is recorded for multiple embedding strengths $\alpha \in [0.001, 0.4]$, knowing that large α values are likely to create unsolvable QP problems in the baseline framework. The median results are reported in Figure 7-1.

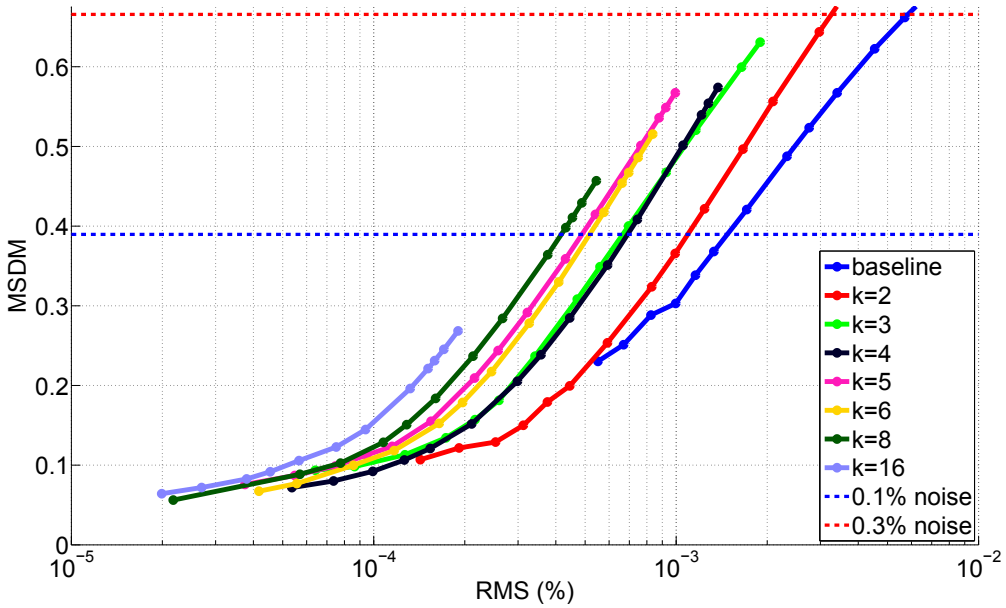


Figure 7-1: Median RMS vs. median MSDM using the ST extension and assessing the embedding distortion when the embedding strength α is varied.

Increasing the spreading length decreases the lower bound of achievable embedding distortions for both metrics. However, if a target RMS can be reached with a spreading length k, increasing this value yields larger perceptual distortion, as measured by the MSDM. This is due to the fact that increasing the spreading length translates into a larger number of bins in the histogram. As a result, the relocation energy is spread on a larger number of ring-like perturbations that are characteristic of the alteration of the distribution of radial distances [LB13]. Although these perturbations are slightly smaller in magnitude, they further trigger the MSDM as they correspond to noticeable high frequency ripples on the surface. Conversely, for a target MSDM value, increasing k lowers the RMS. Indeed, to maintain the MSDM while increasing the number of ring artifacts, it is necessary to greatly reduce their amplitude and thus to decrease the RMS. Overall, these observations corroborate that the MSDM is more sensitive to the ring-like embedding artifacts.

To place these first results into perspective, two dotted lines are added in Figure 7-1 to indicate the MSDM distortion introduced by uniform noise addition for two different levels of noise. The lower level (0.1% noise amplitude with respect to the size of the bounding box) is barely noticeable, while the larger one (0.3%) represents a perceptible alteration. This clearly highlights that ST is most useful to bring flexibility in the acceptable distortion range for the QP framework. Without ST, we would need to use very small embedding strengths to get in this region, which greatly impacts the robustness of the watermark system.

Although not reported on the previous figure, we also investigate the influence of the β parameter

that controls the size of the gap between consecutive bins (see Section 6.3). While using a non-null β marginally increases the RMS, it greatly affects the MSDM. In the state-of-the-art QP [HRAM09], β is set by default to 0.05 and the setting was established using a Hausdorff-only distortion calibration protocol. Because of its influence on the MSDM, we propose instead using $\beta = 0$.

In the following, in order to guarantee a fair comparison, the embedding strength α is adjusted for each spreading length k in order to obtain a MSDM distortion close to 0.37. Figure 7-1 indicates that at this level, the RMS is upper-bounded by around 0.1%.

7.4.2 Robustness with ST

By design, all QP variants are robust to rigid transforms, reordering of vertices, and uniform scaling, and we therefore investigate the impact of ST on the robustness after uniform noise addition. Figure 7-2 illustrates the recorded BER when increasing the attacking strength. There is no curve for $k = 16$ since this setup consistently produces MSDM distortion lower than the target value and therefore exhibits extremely poor robustness.

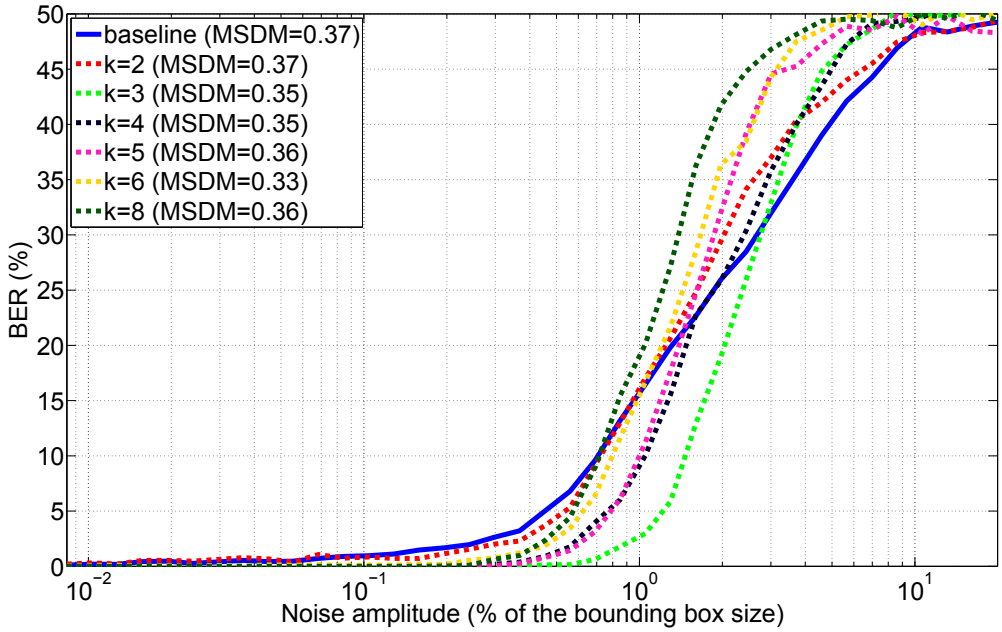


Figure 7-2: Average BER against noise attacks (amplitude as a ratio of the size of the bounding box) for different spreading lengths and a target MSDM of 0.37.

On average, using a longer spreading sequence decreases the BER with respect to the baseline QP framework for noise levels lower than a 1% cut-off threshold. The inverse phenomenon appears for stronger attacks: the BER rockets up more quickly to 50%. In other words, ST preserves the watermark transmission quality longer but collapses more drastically when the attack exceeds the capabilities of the system. The same trend is observed at various target MSDM values. As a rule of thumb, the larger the target MSDM, the longer the system survives and the sharper the transition regime is. To provide a better understanding, Figure 7-3 depicts the robustness vs. fidelity trade-off for 1% uniform noise addition.

The typical operating region for 3D watermarking systems is given by a MSDM in $[0.35, 0.4]$, to have the largest possible distortion that remains hardly noticeable. In this region, ST obviously manages to lower the BER for the same distortion level, and the spreading value $k = 3$ offers the

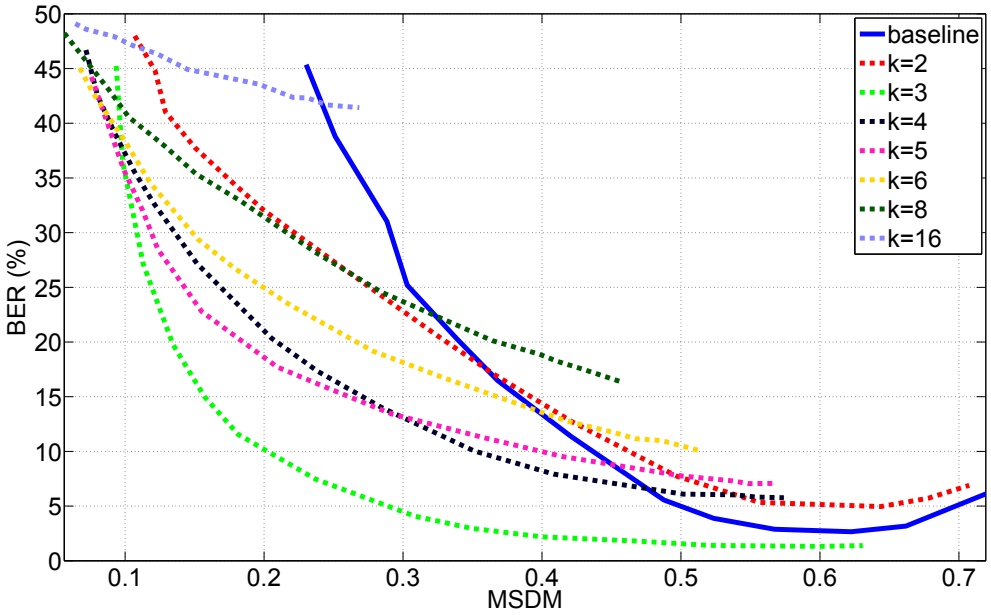


Figure 7-3: Average BER vs. perceptual embedding distortion for the ST extension of the QP framework at 1% uniform noise addition

best fidelity–robustness trade-off.

The same general trend is observed for other attacks, such as quantization or smoothing. At a given level of fidelity, ST provides a boost of robustness against lower levels of attack until a drop-off threshold where performances collapse faster than in the baseline QP 3D watermarking framework. For compactness, the exhaustive list of performance results for the ST component are not reported, and we move on to benchmarking the three other extensions to the QP framework.

7.5 Benchmark of the Integral Centers of Mass

Three different extensions of the QP framework are benchmarked, and, for simplicity, each one is denoted by three letters, as listed in Table 7.1. For instance, `DRS` is the baseline method.

Component	Variants
Center of mass	discrete (`D`), surface-weighted (`S`), volume-weighted (`V`)
Direction of relocation	radial (`R`), normal (`N`), roughness-adapted (`S`)
Cost function	SE (`S`), QEM (`Q`), Laplacian-based (`L`), roughness-based wMSE (`R`)

Table 7.1: Benchmarked variants of the QP framework and their designation.

Contrary to the previous extension, the payload size and the number of bins in the histogram are identical for all the three-letter variants, i.e. $n_b = n_B$[1]. The payload size is then arbitrarily set to $n_b = 64$ bits, which is a value suitable to traitor-tracing tasks and can be used for the low-complexity meshes in the database.

The embedding strength α is adjusted for each variant of the QP framework to calibrate distortion using the same approach than in Section 7.4.1. α is set so that both the RMS and the MSDM

[1]Except for the discarding of the first and last bins in the embedding.

are respectively lower or equal to 0.08% and 0.25, while β is still set to null. The first upper-bound is in practice never reached, as the MSDM is more sensitive to the watermark embedding distortions than the RMS.

In this section, the baseline method (`DRS`) is only compared with the variants resulting from using a surface (`SRS`) or a volume-weighted barycenter (`VRS`), without using the extension on the relocation directions nor the alternate cost functions. The computation time, averaged over all the meshes in the database, are respectively 9s, 11s and 9s for the embedding, and about 0.1s in all cases for the decoding. The optimization process takes around 80% of the overall embedding time.

The benchmarking results are summarized in Figure 7-4. The simplification attack is the one against which the integral formulations are expected to outperform the baseline method, as showcased by Figure 7-4(e). Figure 7-4(a) illustrates the performance against noise addition and suggests that `SRS` has lower robustness than the other methods for large attack levels. This observation reveals that surface-weighted quantities are more sensitive to noise attacks than discrete quantities (see Table 1 in [WLDB11]). The results against the quantization attack in Figure 7-4(b) confirm this fact. Against the smoothing, the triangle soup and the refinement attacks, all variants perform almost equally, as indicated in Figure 7-4(c) and 7-4(d), and Table 7.2. Finally, our experiments show that all methods fail against the cropping attack, i.e., the BER reaches 40% for a cropping ratio lower than 0.01%.

Variant	`DRS`	`SRS`	`VRS`	`VRQ`	`VRL`	`VRR`	`VNS`	`VSS`
BER (%)	18.4	17.9	17.2	20.0	12.9	16.1	28	18.5

Table 7.2: Median BER against Loop subdivision (1 iteration) for different variants of the QP framework.

The remaining extensions are next compared with `VRS`, since it achieves better results than the baseline against simplification attacks, and slightly better results in other cases. In other words, the center of mass is by default set to the volume-based one for all the next benchmarked variants.

7.6 Benchmark of the Perceptually-correlated Cost Functions

To evaluate the improvement resulting from altering the cost function, three variants labeled `VRQ`, `VRL` and `VRR` are tested. In `VRQ`, the cost function is based on the Quadric Error Metric (QEM), as indicated in Eq. (6.38). In `VRL`, the cost depends on the graph Laplacian, as indicated in Eq. (6.39). In `VRR`, the cost is based on the weighted Square Error (SE), as indicated in Eq. (6.41). In all three cases, the mixing parameter λ is set to 0.5. This ad-hoc value achieves marginally better performance. While the computational overhead to compute these cost functions is minimal (less than 0.02s increase in average), the optimization process exhibits a significant slow-down in the case of `VRL`, and a marginal slow-down for `VRQ`. The average computation times are indeed 39s and 12s respectively, compared with 9s for `VRS` and `VRR`.

Fig. 7-7 depicts the robustness of these variants against several attacks and Table 7.2 reports on the robustness against the subdivision attack. In summary, `VRL` (dotted green line, Laplacian-based minimization variant) and `VRR` (dotted purple line, weighted roughness-based square error metric) consistently outperform `VRS` (solid blue line, baseline minimization). The gain in robustness is substantial for the refinement attack. In contrast, the incorporation of the QEM metric in `VRQ` (dotted red line) seems to be counter-productive, with robustness performances slightly poorer than `VRS` in some cases. This may be due to the cost function being less aligned with the distortion metrics used for fidelity calibration.

Since the added value of the proposed extensions is not made clear in the robustness evaluation, the fidelity of some of the variants was further investigated. For instance, Figure 7-5 depicts the cumulative distribution function of the contribution of each vertex to the MSDM metric, for the two variants `VRS` and `VRL` in the *dragon* model. While both watermarked objects globally yield very similar MSDM measurements, they exhibit very different behaviors at a local level. Incorporating the Laplacian component in the cost function indeed appears to produce fewer high local MSDM values that may yield noticeable artifacts.

7.7 Benchmark of the Generalized Relocation Directions

The last round of experiments is dedicated to the use of alternate alteration vector fields. The MSDM used in the calibration process is highly sensitive to displacements along the normal and therefore yields an embedding strength α five times smaller than for other methods, e.g. 0.01 for `VNS` vs. 0.05 for `VRS`. In other words, the normal vector field does not offer a better fidelity–robustness trade-off.

To implement the variant `VSS`, and its roughness-dependent relocation vector field, we rely on the same local roughness estimate than in the perceptually-correlated metric described in Section 6.7.3. In practice, the same post-process is applied to discard outliers and then rescale all estimates in $[0, 1]$. The resulting scalar field is close to 1 in smooth area and almost null for regions with rich details.

The relocation directions in the smooth mesh parts are then forced to lie within the tangent plane. Smooth regions are identified through a threshold that corresponds to the smallest value between the eighth decile of the scalar field and 0.8. This accounts for objects such as the *fandisk* that have a local roughness estimate close to 1 in most places.

Since computing the local roughness takes on average 30s, it increases the embedding time to 40s, in average for `VSS`. Figure 7-7 shows that the robustness of this variant is very similar to `VRS`, and even slightly better for small and moderate attack levels in the noise addition or simplification cases. Figure 7-7 highlights that using the normal direction is not a good idea.

Figure 7-6 illustrates the impact of relaxing the constraint on the direction of alteration for the mechanical object *Fandisk*. This 3D model is characterized by the presence of large planar surfaces. As a result, when the embedding process is limited to the radial direction (`VRS`), fidelity collapses very quickly when increasing the embedding strength α. Moreover, typical ring-like artifacts are produced at the surface of the mesh. By forbidding displacements outside the tangent place in smooth areas (`VSS`), and in particular in planar regions, this ring effect disappears. However, the downside at same embedding strength is a loss of robustness, e.g., in case of remeshing attacks, since the watermark is not actually *burnt* into the geometry; the geometric information in smooth regions being unaltered by the watermark, although the embedder is conveying some payload information in these regions. Nonetheless, as reported earlier, both `VRS` and `VSS` exhibit comparable robustness against routine attacks when the embedding distortion is calibrated.

7.8 Conclusion

Several variants of the QP watermarking framework have been evaluated through an extensive benchmarking campaign. The reported experimental results demonstrate the added value of these modifications with respect to the traditional fidelity–robustness trade-off.

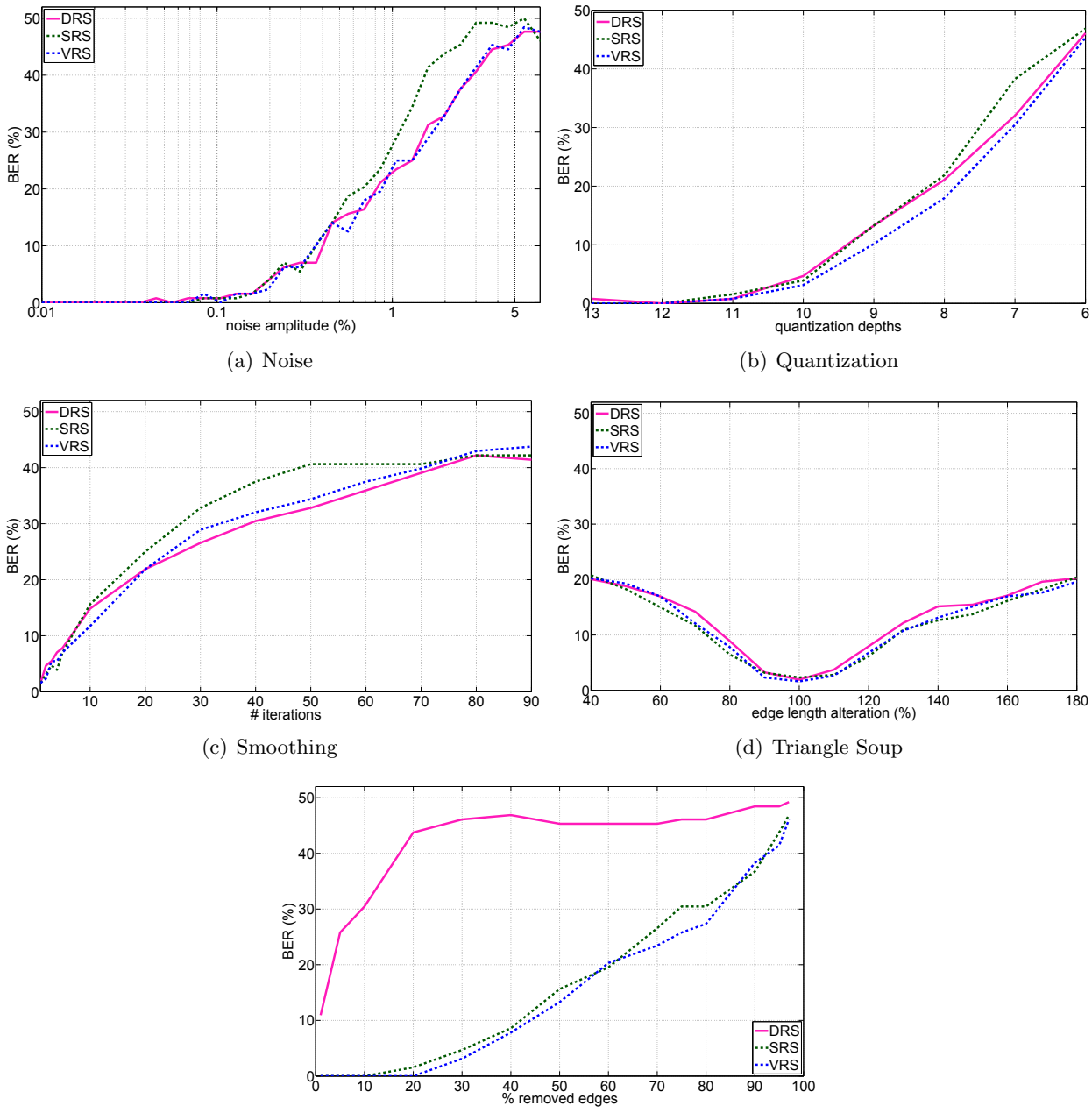
These new degrees of freedom also offer other promising research directions. For instance, the perceptual components incorporated into the cost function are constrained by the QP frame-

work: they have to be quadratic in the unknowns. Ideally, one would rather use well-established perceptually-correlated 3D distortion metrics. While our approach focused on approximating these perceptual components to fit them into the QP framework, alternate options exist. First, these extensions usually involve weights computed from the original mesh, and used throughout the optimization process; it may be advantageous to update these weights at each iteration of an instrumented solver. Second, it may be worth dropping the QP framework altogether and investigate whether alternate solvers could provide better results. Nevertheless, these options are currently limited to inputs with small or medium sizes [ESP08].

The degree of freedom related to the directions of alteration opens up a new line of research on its own. While we exemplified the potential benefit of this modification using a simple example, it comes with its own set of shortcomings and is definitely not optimal. This raises a fundamental question: what would be the alteration vector field that optimizes the watermarking fidelity–robustness trade-off? For instance, it may be tempting to consider the vector field corresponding to the direction of smallest distortion gradient. Nevertheless, regardless of the difficulty of deriving/computing such direction, it is likely to produce displacements close to the tangent plane and therefore less robust.

In future work, we will also investigate issues that have not been addressed, such as the robustness against cropping and isometric deformations, a.k.a. pose. 3D watermarking systems relying on the modulation of the radial distances share a weakness against these two attacks, due to the loss of vertices and the inability to recover the center of mass. A potential solution would be to extend the presented framework to support other watermarking carriers, such as the local thickness estimates described in Chapter 5 or the geodesic distances [LB11]. In Chapter 9, we explore another strategy, based on a locally-derived global synchronization for the center of mass, in an attempt to provide some robustness against cropping.

Another critical issue relates to the security evaluation of this family of watermarking schemes. The embedding process alters the distribution of radial distances and may introduce non-natural statistical features that could be exploited by an adversary. A handful of approaches have recently tackled this issue [LB13, Yan13] but they remain rather primitive in view of the maturity achieved for other types of content [CFF05, BF13]. To address this problem, the next chapter deals with the security of the watermarking system altering radial distances, especially focusing on the QP formulation and its ST extension. This last extension can potentially bring additional security into the 3D watermarking framework.



(a) Noise

(b) Quantization

(c) Smoothing

(d) Triangle Soup

(e) Simplification

Figure 7-4: Average robustness results. Median BER in five attack scenarios over the thirteen models in the database for three watermarking variants within the framework: the baseline one (`DRS`), and the ones based on the surface (`SRS`) and volume-weighted (`VRS`) extensions.

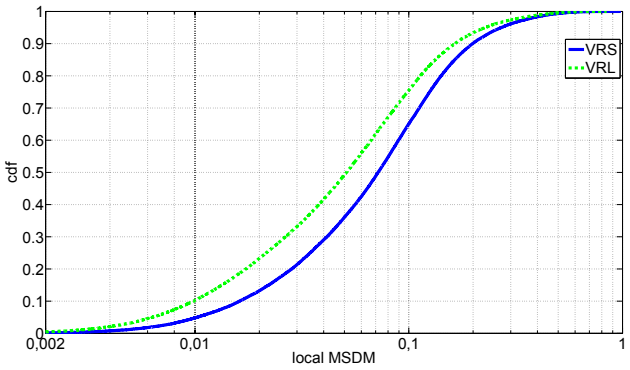


Figure 7-5: Cumulative distribution function of the local MSDM for the *dragon* mesh when using `VRS` and `VRL`.

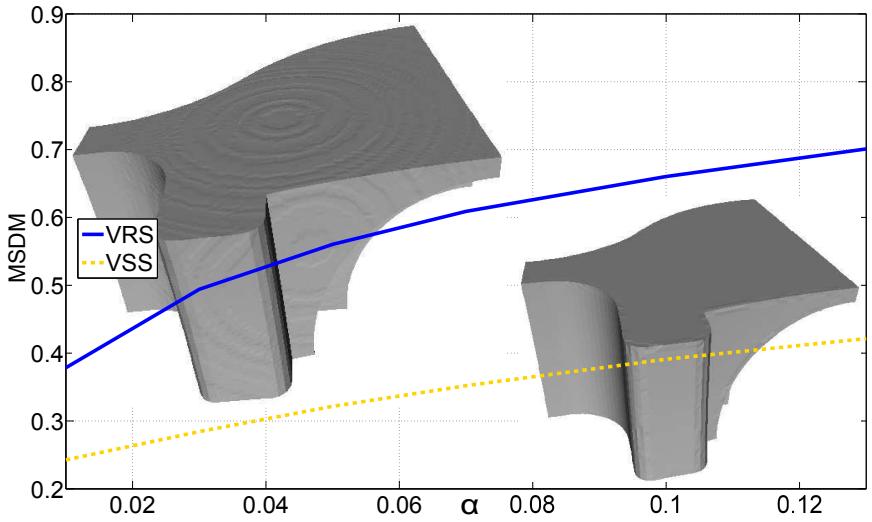
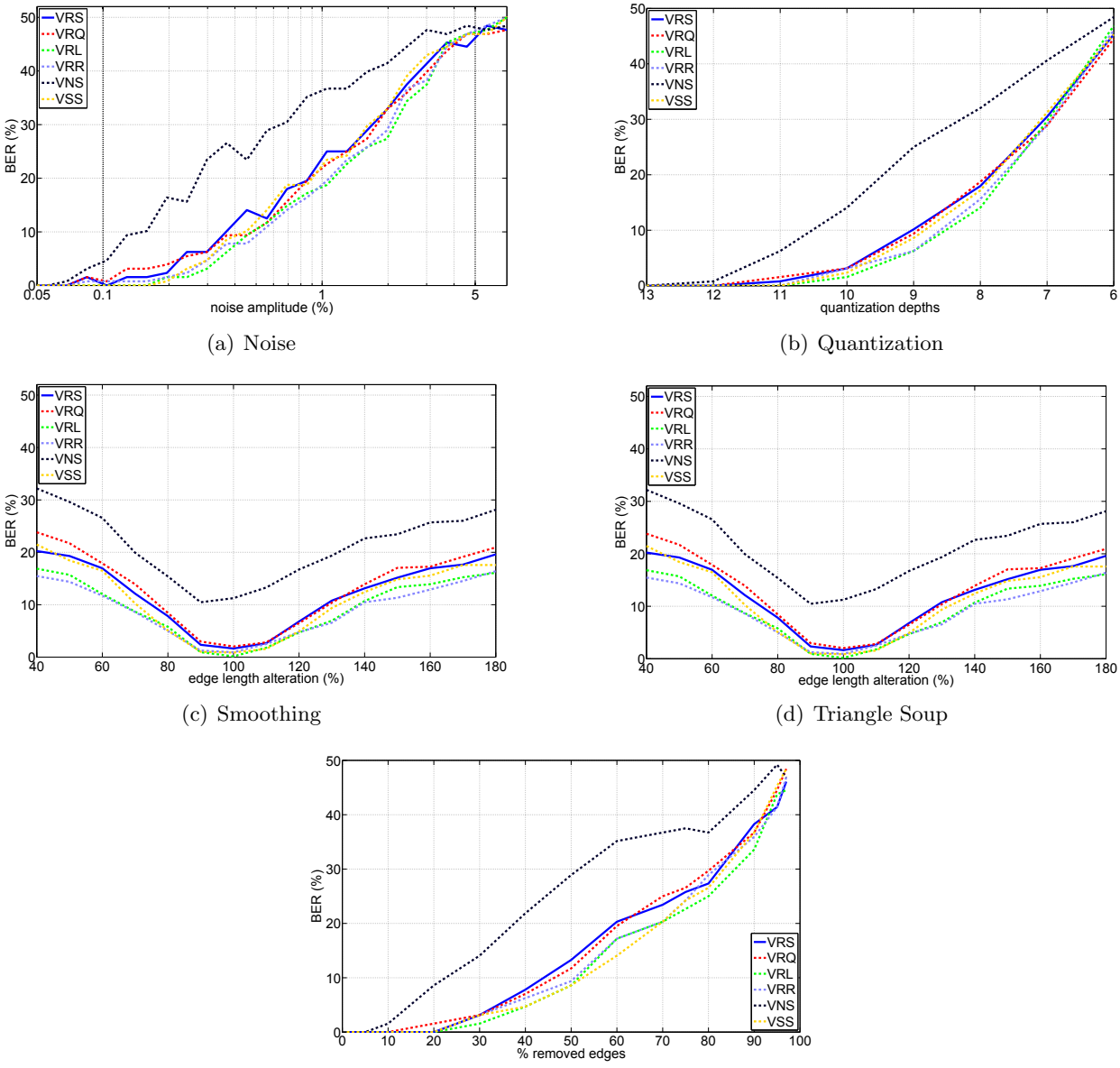


Figure 7-6: Embedding distortion for the *fandisk* when using `VRS` and `VSS` as a function of the embedding strength α. The mesh close-ups correspond to an embedding strength $\alpha = 0.05$.



(a) Noise

(b) Quantization

(c) Smoothing

(d) Triangle Soup

(e) Simplification

Figure 7-7: Average robustness results. Median BER for the thirteen models in the database for different attacks and several variants of the QP watermarking framework. Apart from using the volume-weighted barycenter (`VRS`), each variant differs from the baseline in a single aspect: (i) `VRQ` is based on a minimization with respect to the QEM; (ii) `VRL` is based on a minimization with respect to the Laplacian-based metric; (ii) `VRR` is based on a minimization with respect to the weighted square error metric bases on the local roughness; (iv) `VNS` uses relocation directions set to the mesh normal; and (v) `VSS` uses relocation directions taking into account the local roughness.

Chapter 8

Security Considerations

8.1 Introduction

Digital watermarking consists in modifying multimedia assets in a robust and imperceptible way to convey information. In traitor tracing scenarios, this auxiliary communications channel is exploited to embed information about the recipient of the content and thereby provides a forensic piece of evidence when illegal dissemination occurs. In such copyright protection applications, watermarking systems are required to provide appropriate levels of security to be usable in practice. Rephrased differently, an unauthorized user should not be able to access the watermarking communications channel in any way, e.g., read, erase, or write access [Kal01].

With the increasing use of 3D assets in the movie and video-game industry and the rapid development of 3D printing, copyright protection for 3D content is becoming more pressing and, consequently, new dedicated watermarking algorithms are needed. While a number of 3D watermarking techniques have been proposed in the past, security evaluation has been somewhat neglected in comparison to other types of content. For instance, it is common practice to rely on a 128-bit long secret key to derive pseudo-random parameters used in the watermarking algorithm and to invoke a cryptographic argument to argue that brute-force search would be computationally prohibitive [LB13, WLDB11]. However, recent works have clearly showcased that distinct keys could yield *equivalent parameters* that grants watermarking decoding capabilities [BF13]. In other words, watermarking security should not be reduced to the length of the secret key.

In this chapter, we analyze the security of the popular radial distance-based approach to 3D watermarking, using the Quadratic Programming (QP) instantiation that was previously investigated. We complement this generic framework with two conventional security mechanisms in Sections 8.2 and 8.3 and illustrate that such secret parameters could be reverse-engineered quite efficiently. Finally, our findings are summarized in Section 8.4 and potential means to improve the security of this 3D watermarking framework are suggested. The quantities and notations in this Chapter are the same as the ones introduced in Section 6.2.

8.2 Histogram Security

Without the use of the Spread Transform (ST) extension proposed in Section 6.4, there is currently no secret parameter in the watermarking system described in Chapter 6. In other words, an adversary can easily tap into and tamper with the watermarking channel. A simple, yet effective security strategy is to obfuscate the support used for watermarking. For instance, one could discard a pseudo-randomly determined portion of the distribution of radial distances $\boldsymbol{\rho}$ prior to

computing the histogram. To perform an attack, the adversary should then reverse-engineer this secret parameter in order to compute the same histogram as the watermarking system.

8.2.1 Symmetric Relative Offset

A first embodiment of this strategy is to remove a portion $\epsilon = \eta(\rho_{\max} - \rho_{\min})$ at both ends of the distribution [LB13], where $\eta \in \mathbb{R}^+$ is pseudo-randomly determined using a secret key. In other words, ϵ serves as a secret relative offset to ignore all samples in $[\rho_{\min}, \rho_{\min} + \epsilon) \cup [\rho_{\max} - \epsilon, \rho_{\max}]$ when deriving the watermark communication channel.

The natural distribution of radial distances in a bin of the histogram is close to uniform [CPJ07]. Prior to watermarking, most of the watermark carrier values, i.e., the normalized averages of the bins of the histogram of $\boldsymbol{\rho}$, are therefore close to 0.5. Since the embedding function either raises the watermark carrier above $0.5 + \alpha$ or lowers it below $0.5 - \alpha$, the distribution of the carriers notably differs for watermarked and non-watermarked content. Based on this a priori knowledge, an attacker can define a termination test for a brute force attack [LB13]. In this chapter, we use the value $\hat{\epsilon}$ that minimizes the deviation from the expected watermarked carrier values, e.g.:

$$\mathrm{d}(\tilde{\epsilon}) = \sum_{i=1}^{n_b} \min_{b \in \{-1,1\}} |\bar{c}_i(\tilde{\epsilon}) - 0.5 + b\alpha| \,, \tag{8.1}$$

where $\bar{c}_i(\tilde{\epsilon}) = \frac{1}{N_j(\tilde{\epsilon})} \sum_{k|B_k(\tilde{\epsilon})=i} \bar{\rho}_k(\tilde{\epsilon})$ denotes the normalized average in bin i for a test secret offset $\tilde{\epsilon}$.

In watermarking, the security against brute-force attacks depends both on the number of to-be-tested parameters $\tilde{\epsilon}$, and on the probability for the adversary to pick an equivalent parameter, which allows for a reliable decoding of a content that is watermarked with ϵ [CDF06, BF13]. In the QP framework, this probability is not easily tractable as ϵ controls multiple aspects of the watermark carrier computation. For instance, in contrast with Quantization Index Modulation (QIM) where the quantization step-size is set, both the positions and the width of the bins of the histogram (defined Chapter 6) here depend on the secret parameter.

8.2.2 Experimental Results

A series of 200 brute-force attacks is conducted on the database of 12 meshes (see Table D.1 in Appendix), with a number of vertices ranging from 10^4 to 10^5. Since the attack scenario is a Watermarked-Content Only Attack (WOA), all the estimated quantities are watermarked and the superscript $^{\mathrm{w}}$ is dropped. For each attack, a seed η is generated in $(0, 0.1]$ and a randomly selected mesh is watermarked with a random 32-bits payload. The brute force attack is then performed by sampling $(0, 0.1(\rho_{\max} - \rho_{\min})]$. In all trials, the sampling step is set to twice the average of $\nabla\boldsymbol{\rho}^\star$, $\nabla\boldsymbol{\rho}^\star$ being the difference between two consecutive values of the radial distances sorted in ascending order, denoted $\boldsymbol{\rho}^\star$.

Computing all $\mathrm{d}(\tilde{\epsilon})$ values takes on average 13 seconds with a PC clocked at 2.5 GHz. Using $\hat{\epsilon} = \arg\min \mathrm{d}(\tilde{\epsilon})$ as the secret parameter in the decoder leads to an average Bit Error Rate (BER) of about 0.4%. Close-up results for a single trial are depicted in Fig. 8-1. The offset yielding the minimum value of d(.) (solid blue line) coincides with the ground truth ϵ (position marked with the green dotted line) and the corresponding BER (solid green line) is null. Since d(.) presents multiple local minima, using optimization methods to straightforwardly minimize the function may fail.

While $\hat{\epsilon}$ only provides a coarse estimation of ϵ, it is sufficient to fully read the payload in most cases. Still, one can additionally perform a standard minimization of d(.), over a limited interval

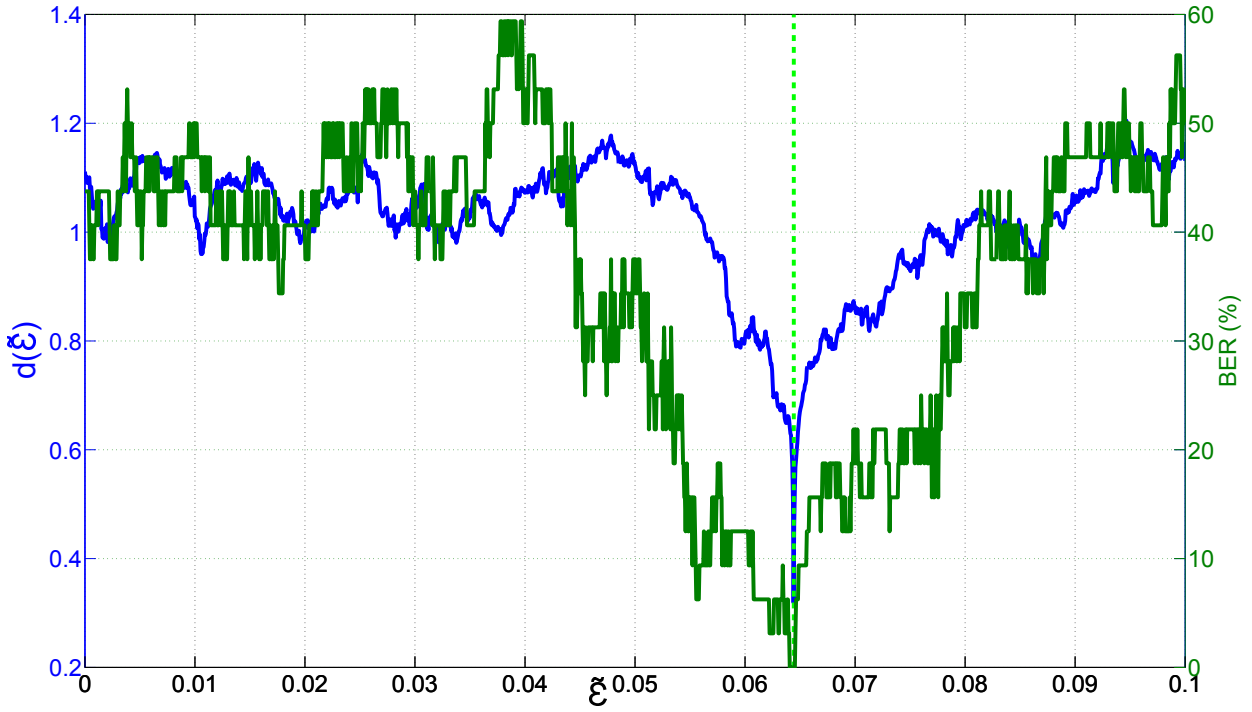


Figure 8-1: Brute-force attack against ϵ on the *Bunny* mesh. Values of $\mathrm{d}(\tilde{\epsilon})$ are depicted in blue; in green the BER resulting from decoding the watermark using $\tilde{\epsilon}$ is indicated. Note that $\mathrm{d}(\epsilon)$ is not null, but around 0.2, since the definition of the function d(.) does not take into account the fact that the watermarking constraints are inequality constraints only.

around this coarse estimate, to refine it. This second step takes on average 0.05 seconds and reduces the BER to 0.2%.

8.2.3 Asymmetric Relative Offsets

When the same offset ϵ is added at both ends of the distribution of $\boldsymbol{\rho}$, the complexity of the brute-force attack is linear. To make the brute force attack more difficult, one could use different offsets to ignore all samples in $[\rho_{\min}, \rho_{\min} + \epsilon_{\min}) \cup [\rho_{\max} - \epsilon_{\max}, \rho_{\max}]$ instead. The complexity then becomes quadratic, as Δ is a function of both $\epsilon_{\min}$ and $\epsilon_{\max}$[1] and, based on our previous results, a brute-force attack would take about 14 hours. While it remains manageable, we will exploit hereafter another flaw of the QP framework to defeat this security component.

The watermarking process essentially aims at biasing the natural distribution of the average of the radial distances in each bin and thereby produces noticeable alterations on the distribution of radial distances. For a given bin and a $+1$ payload bit (respectively a -1 bit), the solver adds (resp. removes) an offset to the smallest (resp. the largest) $\bar{\rho}_i$, depending on β (see Eq. (6.13)). This creates small gaps in the distribution of $\boldsymbol{\rho}$, with intervals where no sample may be found. For $\beta = 0$ or when consecutive bits are identical, this phenomenon still occurs, although with smaller gaps. Such tell-tale artifacts can be exploited to estimate the secret offsets $(\epsilon_{\min}, \epsilon_{\max})$. For instance, an adversary could (i) identify the values of $\nabla\boldsymbol{\rho}^\star$ likely to correspond to histogram edges; (ii) compute and iteratively refine an approximation of $\Delta(\epsilon_{\min}, \epsilon_{\max})$ from these gaps; and (iii) estimate $\epsilon_{\min}$.

[1] Equation (8.1) can be expanded and written as a rational function in $\epsilon_{\max}$ and $\epsilon_{\min}$, or, equivalently, in $\epsilon_{\min}$ and $\Delta(\epsilon_{\min}, \epsilon_{\max})$.

Identifying histogram edges

Histogram edges correspond to large values in $\nabla\boldsymbol{\rho}^\star$, but large gaps are also naturally present at both ends of the distribution of $\boldsymbol{\rho}$. To alleviate this problem, the analysis is restricted to the interval $[\rho_{\min} + 0.1\Delta(0,0), \rho_{\max} - 0.1\Delta(0,0)]$. While it does discard samples lying within the first and last bins of the histogram when no secret offset is used, it can be shown that the probability of also discarding a histogram edge is negligible for the parameter values typically used in our study, e.g., $n_b = 32$ and $(\eta_{\min}, \eta_{\max}) \in (0, 0.1]^2$. The attack then starts by keeping the $n_b + 1$ largest values of $\nabla\boldsymbol{\rho}^\star$ computed in this interval.

Some of the identified gaps may not correspond to histogram edges, which could critically hamper the estimation of Δ. In order to eliminate false positives, each gap $g_i = \nabla\rho^\star_{\pi(i)}$, where $\pi(.)$ is an index mapping function that sorts gaps according to their corresponding radial distance i.e., $\rho_{\pi(i+1)} > \rho_{\pi(i)}$, is assigned a confidence score $w_i = w_i^a.w_i^l \in [0, 1]$ and gaps having a score smaller than 0.4 are discarded. The first component w_i^a accounts for the fact that larger gaps are more likely to correspond to edges. Denoting $\bar{g}_8$ the 8th decile of the selected gaps, it is defined as $w_i^a = (g_i - \min g_j)(\bar{g}_8 - \min g_j)^{-1}$ for gaps $g_i \leq \bar{g}_8$ and set to 1 otherwise. The second component w_i^l accounts for an empirical observation that suggests that the likelihood of false positives increases with the radial distance $\rho_{\pi(i)}$. It is defined as $w_i^l = (\max \rho_{\pi(j)} - \rho_{\pi(i)})(\max \rho_{\pi(j)} - \text{mean } \rho_{\pi(j)})^{-1}$ for gaps having $\rho_{\pi(i)} \geq \text{mean } \rho_{\pi(j)}$ and set to 1 otherwise. Figure 8-2 illustrates this outlier rejection procedure on the *bunny* mesh.

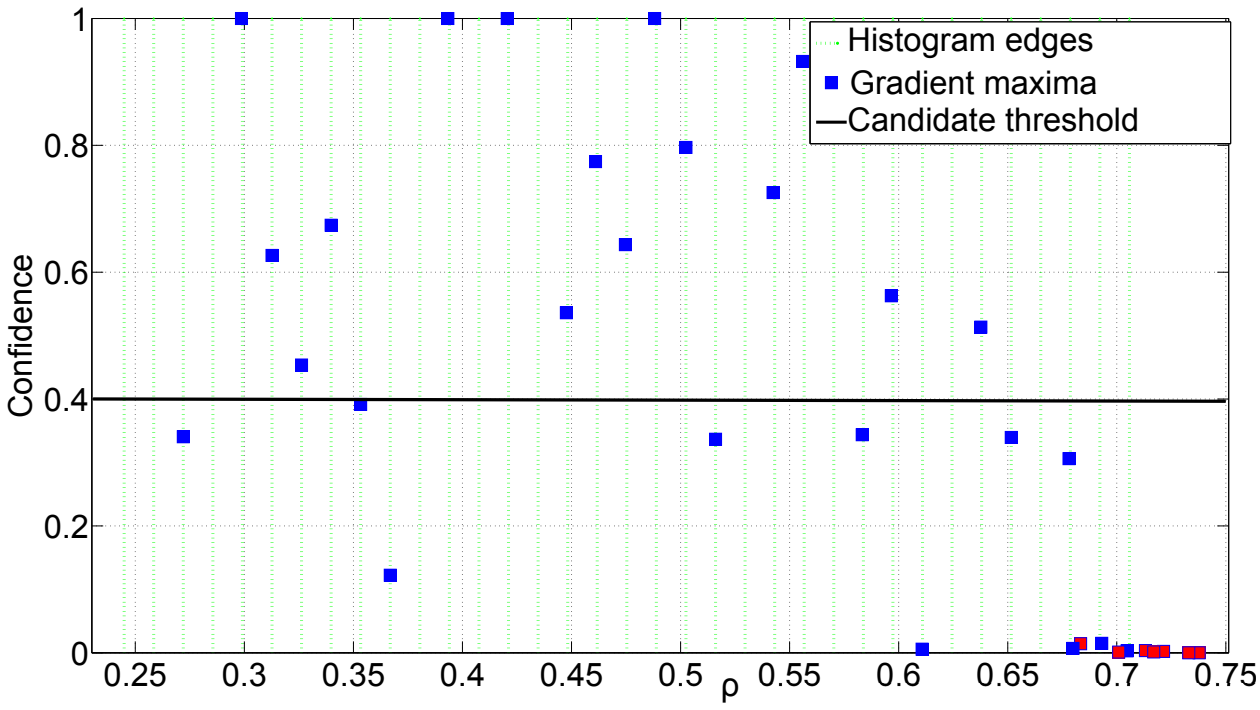


Figure 8-2: Illustration on the *bunny* mesh of the removal of the largest values in $\nabla\boldsymbol{\rho}^\star$ that do not correspond to histogram edges (false positives). In green, the ground truth histogram edges are evenly spaced along the x-axis corresponding to ρ values. Markers correspond to the largest gaps g_i (x-axis) and their associated confidence w_i (y-axis). Blue markers are true positives (aligned with the ground truth edges). False positives in red are all located in the upper-part of the distribution and correctly rejected with the selected threshold, although the miss rate of this process is large.

Since the location of the edge associated to a gap g_i is in an interval $[\rho^\star_{\pi(i)}, \rho^\star_{\pi(i)+1}]$, the distance between two gaps $\tilde{\Delta}_{i,j}$ $(j > i)$ can be defined as the distance between the centers of these intervals. Intuitively, when two successive gaps g_i and g_{i+1} are true positives, the distance between them is approximately equal to some multiple $k\Delta$ $(k \in \mathbb{N})$. To further reduce the number of false positives, $\tilde{\Delta}_{i,i+1}$ is compared with the lower bound $\Delta_{\min}$ obtained when $\eta_{\min} = \eta_{\max} = 0.1$. When $\tilde{\Delta}_{i,i+1}$ is too small, the gap with the lowest confidence is removed. This procedure is repeated until all $\tilde{\Delta}_{i,i+1}$ are larger than $\Delta_{\min}$. Although this greedy scheme is suboptimal, it ensures that multiple gaps clustered around one histogram edge are removed. This first part of the attack results in a set of n_g candidate gaps, which are still denoted g_i for simplicity.

Estimation of Δ

The initial estimation $\hat{\Delta}_0$ is based on the minimum distance between consecutive candidate gaps $\Delta_{\inf} = \min_i \tilde{\Delta}_{i,i+1}$. This minimum distance should be close to $k\Delta$ for some multiplier k but nothing guarantees that this multiplier is actually equal to one. Based on the definition of Δ, one can show that $k\Delta < \Delta(0,0)$ is equivalent to $\eta_{\min} + \eta_{\max} > \frac{k-1}{k}$. With secret parameters $\eta_{\min}$ and $\eta_{\max}$ lower than 0.1, this inequality only holds for $k = 1$. As a result, if $\Delta_{\inf} < \Delta(0,0)$, $\hat{\Delta}_0$ is set to $\Delta_{\inf}$. Otherwise, $\hat{\Delta}_0$ is set to $\hat{k}_0^{-1}\Delta_{\inf}$, where $\hat{k}_0$ is the smallest integer such that $\hat{k}_0^{-1}\Delta_{\inf} < \Delta(0,0)$.

This first estimation is then iteratively refined by considering increasingly distant gaps. More specifically, at the tth iteration, a series of lower bounds $\mathcal{L}_t$ and upper bounds $\mathcal{U}_t$ on Δ are approximated from $\tilde{\Delta}^{\max}_{i,i+j} = \rho_{\pi(i)} - \rho_{\pi(j)+1}$ and $\tilde{\Delta}^{\min}_{i,i+j} = \rho_{\pi(i)+1} - \rho_{\pi(j)}$, where $1 \leq i < n_g - t$ and $1 \leq j \leq t$. There indeed exists an integer $k_{i,i+j}$, such that: $\tilde{\Delta}^{\min}_{i,i+j} \leq k_{i,i+j}\Delta \leq \tilde{\Delta}^{\max}_{i,i+j}$. Using $\hat{\Delta}_{t-1}$ instead of Δ, the lower and upper bounds are approximated as:

$$\mathcal{L}_t = \left\{ \tilde{\Delta}^{\min}_{i,i+j} \Big/ \hat{k}^{\min}_{i,i+j} \right\}_{\substack{1 \leq i < n_g - t \\ 1 \leq j \leq t}}, \hat{k}^{\min}_{i,i+j} = \left\lceil \frac{\tilde{\Delta}^{\min}_{i,i+j}}{\hat{\Delta}_{t-1}} \right\rceil \tag{8.2}$$

$$\mathcal{U}_t = \left\{ \tilde{\Delta}^{\max}_{i,i+j} \Big/ \hat{k}^{\max}_{i,i+j} \right\}_{\substack{1 \leq i < n_g - t \\ 1 \leq j \leq t}}, \hat{k}^{\max}_{i,i+j} = \left\lfloor \frac{\tilde{\Delta}^{\max}_{i,i+j}}{\hat{\Delta}_{t-1}} \right\rfloor . \tag{8.3}$$

The estimate $\hat{\Delta}_t$ is then set to the average between $\max(\mathcal{L}_t)$ and $\min(\mathcal{U}_t)$, before proceeding to the next iteration. The rationale for not directly computing all the bounds from all the pairs of distances based on $\hat{\Delta}_0$ is that the initial approximation error results in incorrectly estimated multiple factors $\hat{k}_{i,i+j}$ for distant gaps and thus invalidates the use of these distant gaps to obtain a better estimation. Fig. 8-3 illustrates the sets $\mathcal{U}_{n_g}$ and $\mathcal{L}_{n_g}$ resulting from one experiment, as well as the ground-truth Δ. In this case, the iterative procedure provides an accurate estimate of Δ.

This procedure however fails when the accuracy of the candidate gap locations drops. If $\hat{k}^{\max}_{i,i+j}$ differs from $\hat{k}^{\min}_{i,i+j}$, one of the two values can be adjusted by 1 before the next iteration. For very inaccurate gap locations, this straightforward correction becomes insufficient, but lower and upper bounds on Δ are then intertwined, which can be detected. The correction is then not applied.

Estimation of $\epsilon_{\min}$

In the last step of the attack, $\epsilon_{\min}$ is indirectly estimated with the solution of an optimization problem minimizing $\mathrm{d}(\epsilon_{\min}, \epsilon_{\max})$. Thanks to our estimate $\hat{\Delta}_{n_g}$, the optimization problem can be simplified to a single variable one. Moreover, for efficiency, and because of the multiple local minima, the search space is limited to $[\rho_{\pi(i_M)}, \rho_{\pi(i_M)+1}]$, where i_M is the index of the candidate gap

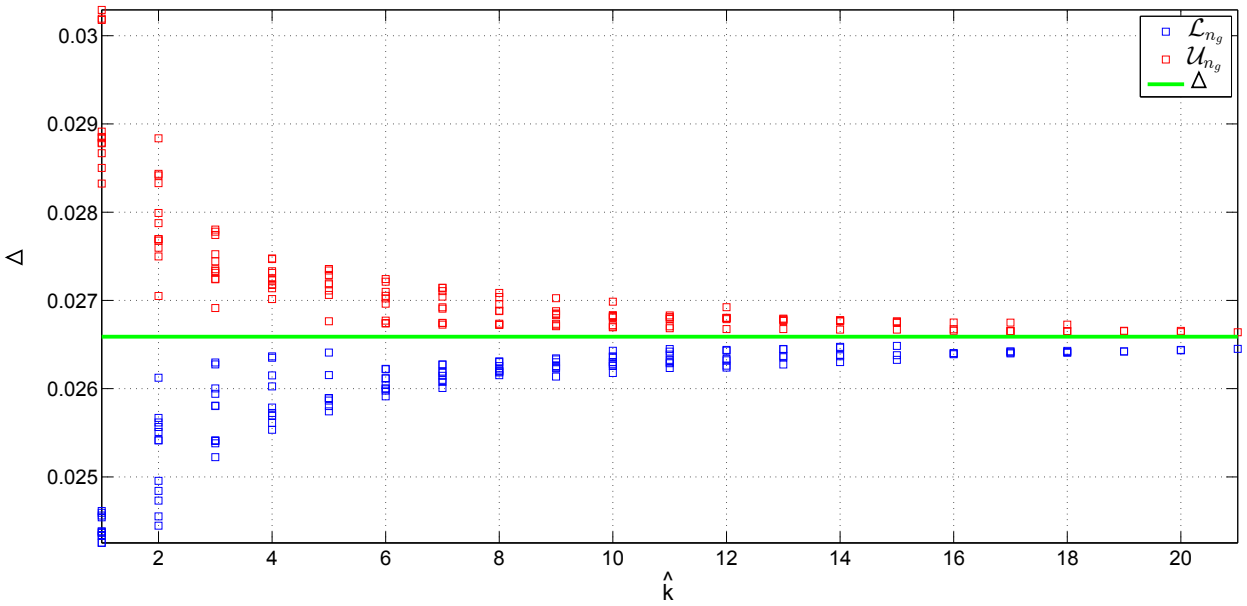


Figure 8-3: Lower bounds $\mathcal{L}_{n_g}$ and $\mathcal{U}_{n_g}$ on Δ inferred from the multiple candidate gaps g_i and the locations of the corresponding histogram edges defined with the intervals $[\rho_{\pi(i)}, \rho_{\pi(i)+1}]$.

with the highest confidence. Based on the resulting minimum and $\hat{\Delta}_{n_g}$, a list of potential histogram edges is built. The histogram edges are a subset of these candidate edges. To decode the payload, this subset is chosen as the one with $n_B + 1$ consecutive elements, for which d(.) is minimum.

8.2.4 Experimental Results

This attack is tested using the same protocol as in Section 8.2.2. In each of the 200 trials, the secret parameters are generated so that $(\eta_{\min}, \eta_{\max}) \in (0, 0.1]^2$. Fig. 8-4 reports the averaged relative error in the estimates of the secret parameters for decreasing values of the gap controlling parameter β (as defined in Equation (6.13) in Chapter 6[2]). These measurements clearly showcase the benefit of the proposed refinement process, with more than one order of magnitude of accuracy gain between $\hat{\Delta}_0$ and $\hat{\Delta}_{n_g}$. As anticipated, the smaller β is, the less pronounced are the gaps and the slightly less accurate is the estimation of Δ and $\epsilon_{\min}$. Nevertheless, even for $\beta = 0$, gaps are still created and form evidence to find the histogram edges.

Using this approach, an attacker can successfully access the bins of the histogram which lie in the middle part of the distribution. However, the last part of the attack, i.e., the selection of the subset of edges on which d(.) is minimum, is the one where most of the errors occur. At this point, the success of the attack can be measured through the BER, which is 10.45% at $\beta = 0.05$ and 13.82% at $\beta = 0$.

Looking at the BER between the embedded payload and the estimated payload with a shift of 1 bin, the BER decreases in both cases to 4.74% and 10.3% respectively. The explanation for these second results is that, when β is non-null, decoding errors mainly occur in the last part of the attack. Edges are correctly reconstructed, but the subset of bins that is identified as carrying the payload is misaligned, e.g., the carrier is taken as starting at the ith bin instead of the $(i+1)$th bin. In practice, the adversary is then granted access to all but one watermark carrier values. When β

[2] β controls the bounds on the relocation unknowns and provides the means to achieve 'dead-zones' around histogram edges, which limits the risk that vertices switch bins because of an attack.

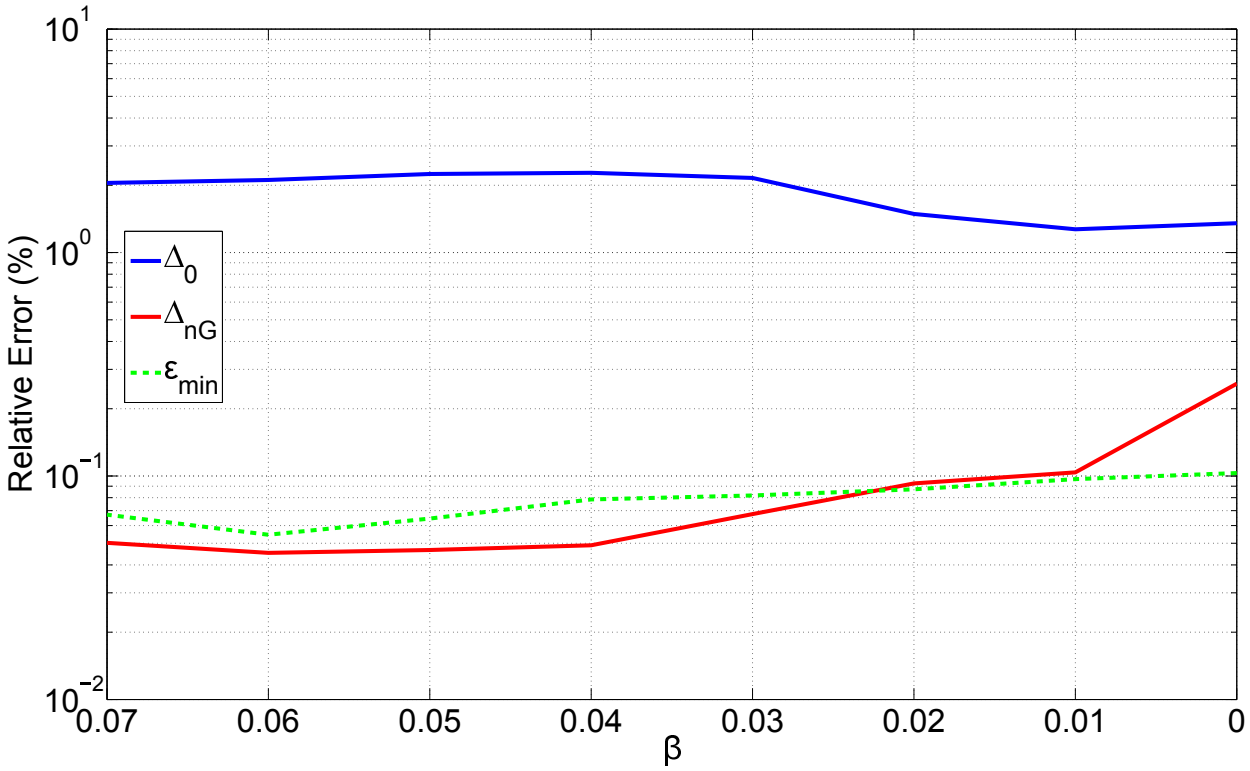


Figure 8-4: Relative error for the approximations $\hat{\epsilon}$ and $\hat{\Delta}_{n_g}$ of the secret parameters, averaged over 200 independent trials (the attack only needs one watermarked observation), for decreasing values of β, the gap controlling parameter. The stability of the accuracy with respect to β suggests that very few candidate gaps are required for the attack to succeed. Indeed, the confidence w_i diminishes with β, thus decreasing the number of available candidate gaps.

is null however, the BER allowing for the misalignment of a single bin is only slightly smaller than the routine BER. In this case, decoding errors are caused by a complete failure of the attack, as the locations of all the histogram edges are incorrect. The non-null BER are indeed close to 50%.

8.2.5 Countermeasures

To improve the security of the watermarking system, the cost function in the QP framework can be altered to mitigate the embedding distortion in the histogram of $\boldsymbol{\rho}$. In general, this represents a challenge which is very similar to finding a more appropriate cost function than the squared error metric to lessen the perceptual impact of watermark embedding (see Section 6.7). Still, a straightforward solution consists in setting a cost function which increases the cost of relocating vertices that are close to bin boundaries. For instance, we tested the following cost function:

$$\omega = \sum_{i=1}^{n_v} \left(1 + \gamma(2\bar{\rho}_i - 0.5)^{\nu}\right) \delta\bar{\rho}_i^{\mathrm{w}2}, \tag{8.4}$$

where the parameters (γ, ν) controls the relocation penalty. Furthermore, the gap controlling parameter β is set to 0.

In Fig. 8-5, ν is set to 12, and the influence of γ on the gap magnitude, measured at the

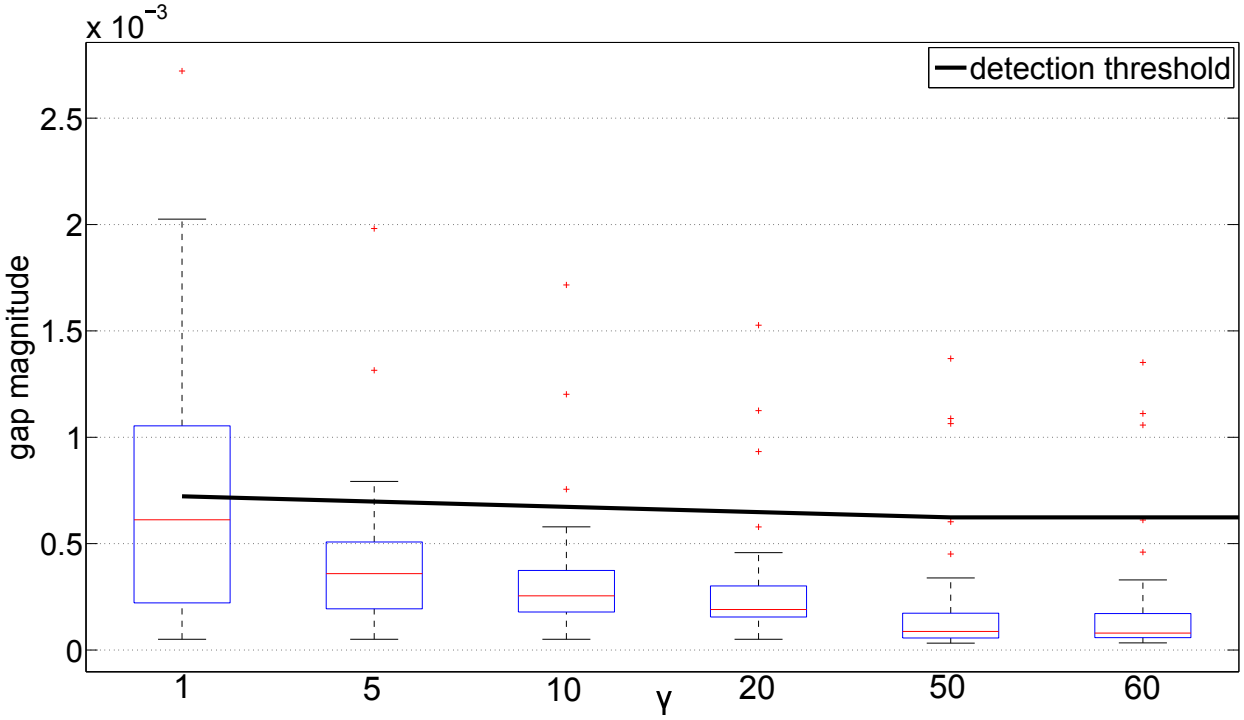


Figure 8-5: Boxplot of the gap magnitudes at the ground-truth histogram edge locations, depending on the penalty parameter γ defined in Eq. (8.4). Results correspond to the *Bunny* mesh and a 32-bits payload. The threshold to recover the n_b+1 largest gaps (black) quickly becomes inappropriate, as all but 4 edges are missed. For $\gamma \geq 10$, the weighting process of the candidate gaps also becomes inefficient and the attack is thwarted.

ground-truth location of the histogram edges, is depicted. The detection threshold corresponds to the selection of the n_b+1 largest gaps in $\nabla\boldsymbol{\rho}^\star$. For $\gamma > 1$, the number of edges recovered drops from 17 to about 4. Lowering the detection threshold greatly increases the number of false positives. For instance, at $\gamma = 10$, lowering the threshold to 4×10^{-4}, induces a detection process with 8% precision. On a side note, for $\gamma = 10$, the embedding distortion, assessed with the perceptually-correlated Mesh Structural Distortion Measure (MSDM), is actually slightly smaller than when using the baseline squared error cost function. This suggests that the proposed security-oriented counter-measure could provide some additional benefits to the system with respect to the traditional distortion vs. robustness trade-off.

8.3 Security with a Spread-Transform

Another typical security mechanism in watermarking is to rely on random projections, through the ST, to obfuscate the watermarking subspace. The induced theoretical changes on the baseline QP framework have been presented in Section 6.4, while the practical consequences of using the ST have been investigated in Section 7.4. In the following, the $k = \frac{n_B}{n_b}$ values of the spreading sequence $\mathbf{s}$ associated with payload bit i are denoted $\mathbf{s}_i$. For simplicity, the previous offset-based security component is not used in this section, i.e.: $\eta_{\min} = \eta_{\max} = 0$.

8.3.1 Estimating the Spreading Sequences

For the ith payload bit, the embedder relocates the watermarking carriers in $\mathbb{R}^k$ so that their projection onto $\mathbf{s}_i$ is at a distance greater than α from $\bar{\mathbf{t}}\cdot\mathbf{s}_i$, where $\bar{\mathbf{t}}$ is the vector whose k components are 0.5. Geometrically, this watermarking process translates in a dead zone in $\mathbb{R}^k$ defined by two hyperplanes characterized by the normal vector $\mathbf{s}_i$ and the offsets $\bar{\mathbf{t}}\cdot\mathbf{s}_i \pm \alpha$. The objective of the attacker is then to exploit this statistical bias to estimate the secret spreading sequence $\mathbf{s}_i$ based on the observation of several objects watermarked with the same key. Since the watermarking carriers are naturally distributed around 0.5, the watermarking process is indeed expected to produce two well-separated clouds of points in $\mathbb{R}^k$, depending on the sign of the payload bit (see Fig. 8-7). In the WOA context, $\mathbf{s}_i$ can be estimated as the first principal direction of the Principal Component Analysis (PCA) of n_o observations of the watermarked carriers [CFF05]. In the Known-Message Attack (KMA) context, finding $\mathbf{s}_i$ amounts to estimating the best discriminating hyperplane to partition the two point clouds. This is equivalent to a supervised binary classification problem and Fisher's Linear Discriminant (FLD) can be used instead.

Attacks are experimented with spreading lengths k in $[\![2, 5]\!]$. 32-bits random payloads are embedded in n_o randomly selected meshes among the 12 in the database. The attack consists in: (i) computing the $n_B.n_o$ watermarked carriers; (ii) performing either a PCA or a FLD on each one of the n_b sub-signals to obtain an estimation $\hat{\mathbf{s}}_i$ of the spreading sequence associated to each bit; and (iii) assessing the quality of the estimation with $\frac{1}{n_b}\sum_{i=1}^{n_b} |\mathbf{s}_i \cdot \hat{\boldsymbol{s}}_i|$. The absolute value in the performances metric is due to the ambiguity on the spreading sequence sign that cannot be lifted in the WOA setup.

Fig. 8-6 shows that very few objects are required for the estimation process to become stable. While the estimated spreading sequence $\hat{\mathbf{s}}_i$ are always very close to the ground truth in KMA, the accuracy of the estimation greatly diminishes in WOA when increasing the spreading length k, even when using larger numbers of observations (e.g. 700, not reported). This discrepancy between the FLD and the PCA stems from a particular configuration of the watermarked carriers illustrated in Fig. 8-7. Although the carriers are clustered according to the bit sign, the variation in-between clusters is smaller than the one inside each cluster. As a result, the first component of the PCA, which captures the direction of largest variation, is drawn away from the ground truth whereas the FLD can successfully cope with the situation thanks to the payload labels attached to the observations.

The spreading length k has a large influence on the robustness of the embedded watermark and small values (e.g., $k = 3$) have been reported to yield better performances in Section 7.4.2. The results of the attack suggest that an adversary, with access to a few dozens of watermarked meshes, can obtain a good estimate of the secret spreading sequence $\mathbf{s}$, which can then be exploited to alter the payload bits with limited distortion for instance.

8.3.2 Accommodating for Shuffling

To prevent an adversary from straightforwardly estimating $\mathbf{s}$, a possible countermeasure is to apply a pseudo-random permutation to the watermark carriers prior to applying the ST component. Without knowing the secret permutation, the attacker can no longer partition the cover into independent chunks to estimate the $\mathbf{s}_i$ individually. As a result, the PCA attack has to be performed in $\mathbb{R}^{n_B}$ (rather than $\mathbb{R}^k$) and therefore requires significantly more observations to succeed. Still, in the WOA context, an attacker may rely on the mutual information between carriers to group the histogram bins assigned to the same payload bit prior to estimating the spreading sequences for each one of these n_b groups.

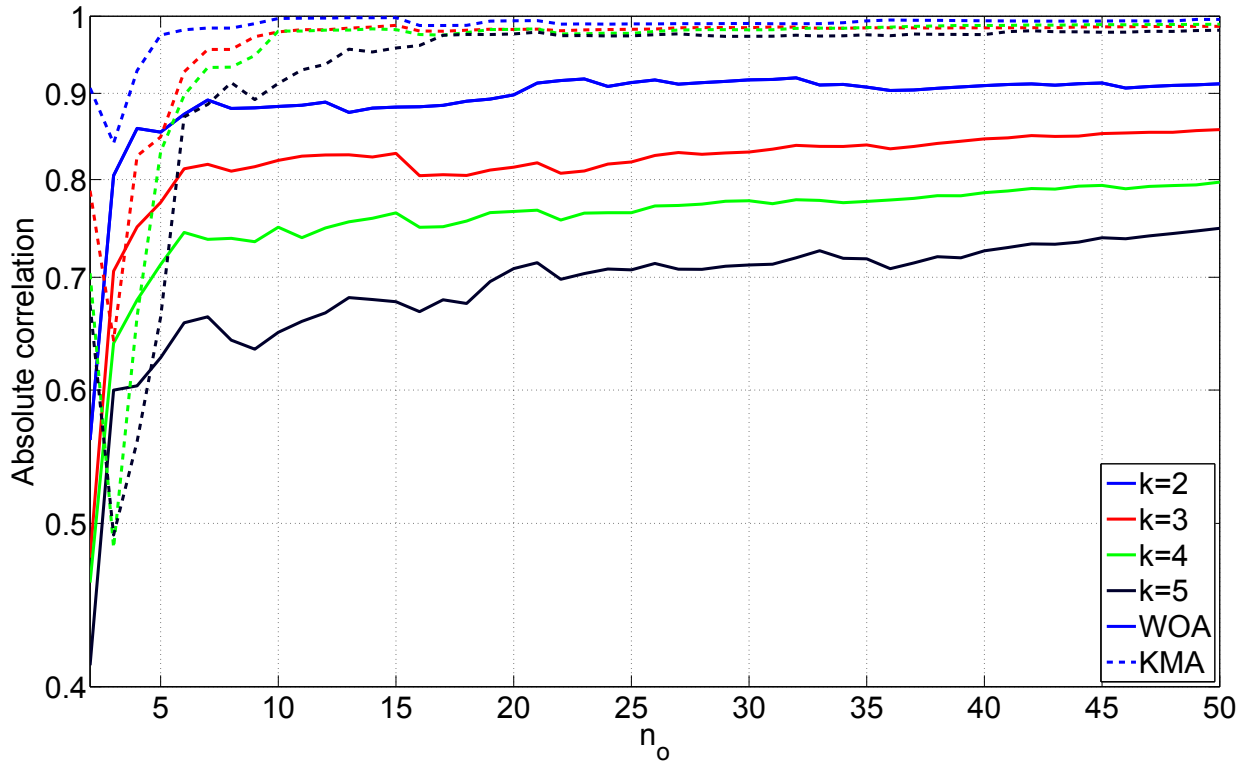


Figure 8-6: Average absolute correlation between the secret spreading sequence $\mathbf{s}$ and the estimation $\hat{\mathbf{s}}$, based on an increasing number n_o of watermarked objects. Each watermarked object has been randomly selected and watermarked with a 32-bits payload, with spreading lengths $k \in [\![2, 5]\!]$. For larger k, the accuracy of the estimation drops in the WOA context (PCA-based estimation), even with numerous observations.

Equipped with a number of 3D meshes, watermarked using the same key, the attacker can compute the mutual information between each pair of bin average $c_j \in [\![1, n_B]\!]$ and record the result in a symmetric matrix $\mathbf{\Omega} \in \mathbb{R}^{n_B \times n_B}$. In theory, the mutual information for histogram bins associated to the same payload bit should be significantly larger than for unrelated bins. The optimal solution of this group assignment process is generally complex but approximated solutions can be obtained following a greedy assignment procedure. The mutual information values in the lower triangular part of $\mathbf{\Omega}$ are scanned in descending order and the groups are updated sequentially based on the associated row and column indices using the following rules:

1. `create` – if none of the indices has been previously assigned, and less than n_b groups have been created, both are added inside a new group;
2. `add` – if either one of the indices is already in a group containing less than k elements and the other index is not yet assigned, it is assigned to the unfilled group;
3. `merge` – if both indices are already assigned to different groups, both groups are merged as long as the resulting group has at most k elements;
4. `skip` – in any other case, the couple of indices is dismissed.

Note that the couples that are then lost (because e.g., in case (i), n_b groups have already been created) can still be indirectly recovered when the indices appear in other correct associations.

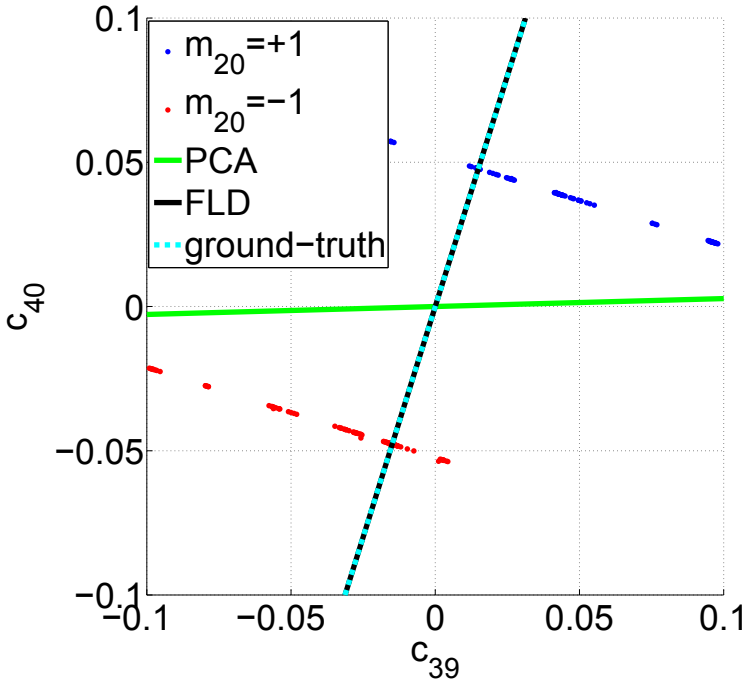


Figure 8-7: Watermark carriers (c_{39}, c_{40}) associated with the bit m_{20} when $k = 2$. $n_o = 200$ observations are depicted. Carriers are clustered, according to the bit sign, on one side of two hyperplanes. The ground-truth spreading $\mathbf{s}_{20}$ is accurately estimated with the FLD as the direction best partitioning both clusters. The PCA however provides a very poor estimate, biased by the larger variations within both clusters.

This procedure outputs a list of $\mathcal{G}_i$ groups ($i \in [\![1, n_b]\!]$), each one with k elements, and each element is only present one in the whole list.

The second part of the attack is the same as Section 8.3.1, except that the watermark carriers are grouped according to the estimated $\mathcal{G}_i$, instead of being grouped by sequences of k consecutive values. The estimation $\hat{\mathbf{s}} \in \mathbb{R}^{n_B}$ of the spreading sequence can then be used to alter the payload.

8.3.3 Attack Performances

In our experiments, we discretized the probability distribution of $\bar{c}_j$ with a 10^{-3} precision to compute the mutual information. This setting has been found to empirically provide a good trade-off between the performances and the computation time of the attack. To first assess the accuracy of the proposed group assignment process, we compute, for each pair $(i, j) \in [\![1, n_b]\!]^2$, the ratio $J_{(i,j)}$ of elements in group $\mathcal{G}_i$ that are associated with payload bit j in the ground truth:

$$J_{(i,j)} = \frac{1}{k} \sum_{B \in \mathcal{G}_i} \delta_{(\sigma(B),j)}, \tag{8.5}$$

where $\sigma(.)$ is the key-seeded mapping function between histogram bin index and payload bit index. Intuitively, the better the group assignment is, the closer $\mathbf{J} = \{J_{(i,j)}\}$ gets to a matrix with exactly one non-null entry (equal to 1) in each column. It is then possible to define the following entropy-

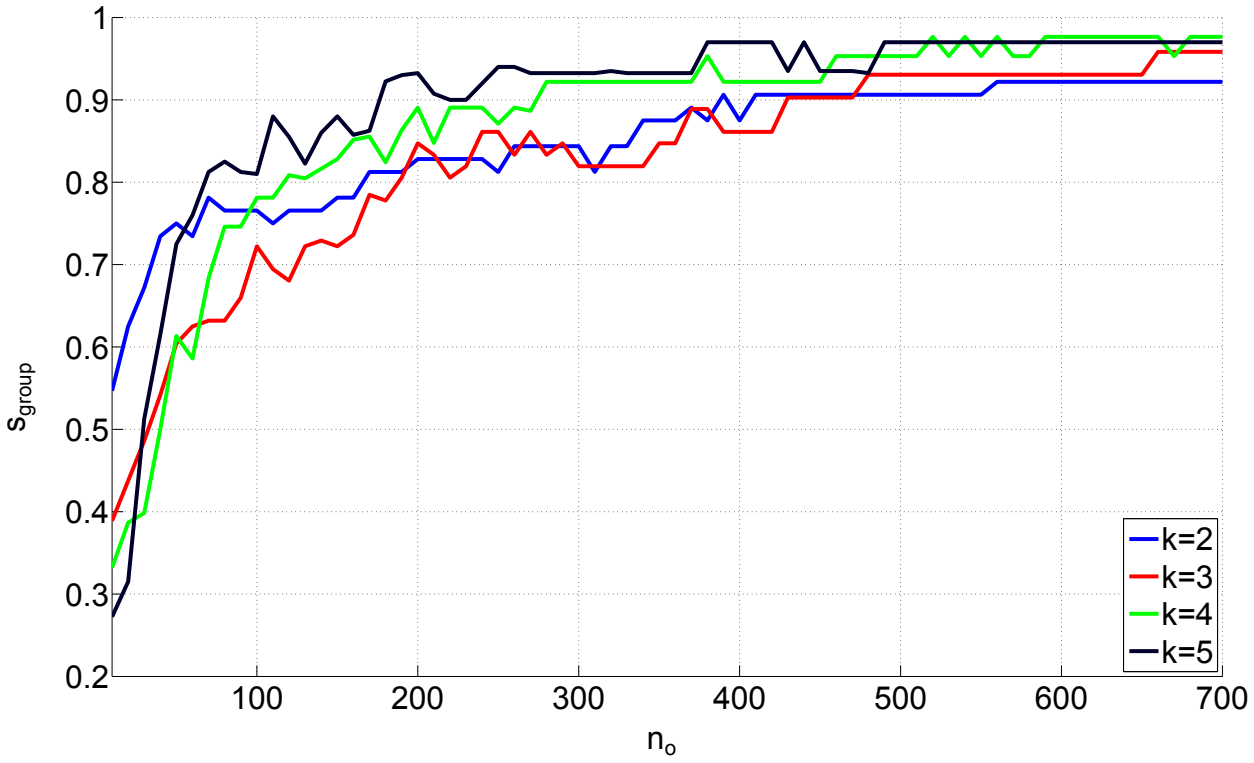


Figure 8-8: Performance of the mutual-information-based estimations of the secret shuffling. $s_{\text{group}} \in [0,1]$ is defined in Equation (8.6). The n_o observations correspond to randomly watermarked meshes with 32-bits payloads.

inspired performance metric:

$$s_{\text{group}} = \frac{1}{n_b}\sum_{i=1}^{n_b}\left(1+\frac{1}{\log(k)}\sum_{j=1}^{n_b} J_{(i,j)}\log\left(J_{(i,j)}\right)\right) \in [0,1]. \tag{8.6}$$

The score s_{group} is equal to 1 when each group $\mathcal{G}_i$ is composed of bins associated with the same payload bit, whereas when $s_{\text{group}} = 0$, all these bins are associated with different payload bits. Fig. 8-8 illustrates the evolution of s_{group} for an increasing number of observations n_o. Regardless of the spreading length, a few hundred watermarked objects are enough to obtain a very accurate group assignment of the histogram bins.

The second metric s_{spread} evaluates the performance of the PCA to estimate the different spreading sequences $\mathbf{s}_i$. When the group assignment is not fully correct, there exist some groups $\mathcal{G}_i$ for which the estimated spreading sequence $\hat{\mathbf{s}}_{\mathcal{G}_i}$ is related to several sequences of the ground truth. To capture all scattered watermark energy, the following correlation score is computed for all pairs $(i,j) \in [\![1, n_b]\!]^2$:

$$C_{(i,j)} = \frac{1}{k+1-\sum_{B\in\mathcal{G}_i}\delta_{(\sigma(B),j)}}\left|\sum_{B\in\mathcal{G}_i} s_B\hat{s}_B\delta_{(\sigma(B),j)}\right|. \tag{8.7}$$

In other words, a partial correlation is computed for the components that are shared between the estimated sequence and the ground truth, and the result is weighted to account for the interfering normalization that is part of the PCA. Intuitively, the more the group $\mathcal{G}_i$ contains elements

assigned to the same ground truth bits, the closer $C_{(i,j)}$ gets to the conventional linear correlation. Conversely, when the group assignment is incorrect, the correlation of the subsequence is penalized accordingly. The metric s_{spread} then simply amounts to computing the average of the sum of the columns of $\mathbf{C}$[3]

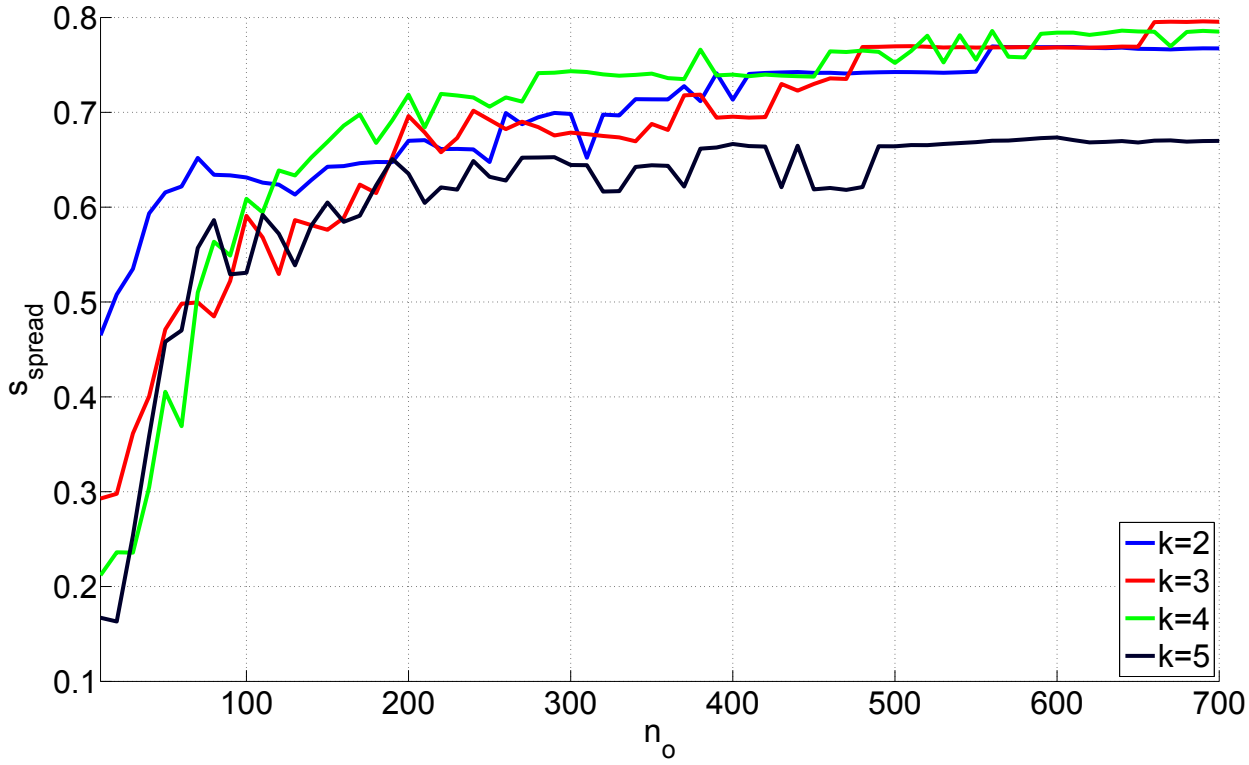


Figure 8-9: Performance of the spreading estimations based on the estimated shuffling. Observed meshes have been watermarked with 32-bits payloads.

Fig. 8-9 depicts the evolution of s_{spread} when an increasing number of watermarked meshes are observed. For $n_o < 200$, the estimated group assignment is inaccurate and the partial correlation scores are penalized. Moreover, since several bits interplay in the same group, the observations are not bimodal and the PCA is actually no longer pertinent. When the group assignment becomes correct i.e., around 200 objects according to Fig. 8-7, most of the partial correlation scores simplify to the ones computed in Section 8.3.1 and the effect of the attack is drastically improved. Over 600 observations, performances seem to saturate and no longer improve.

8.4 Conclusion

In this chapter, we exemplified the current limitations of 3D watermarking with respect to security using a state-of-the-art system that modulates the radial distances to embed the watermark payload

[3]At this point, it is worth explaining why the metrics defined in Eq. (8.6) and (8.7) are replacing the previously used absolute correlation to measure the performance of the attack. The spreadings sequences are indeed estimated without their sign; when projecting the estimate $\hat{\mathbf{s}}$ onto the ground-truth $\mathbf{s}$, the absolute value is taken. When group assignments are incorrect, summing individual absolute values resulting from the projections of different spreading sequences would greatly overestimate the performance of the attack. On the other hand, dismissing any group $\mathcal{G}_i$ where at least one element is incorrect would greatly underestimate the performance.

in a 3D object [HRAM09]. In contrast with previous works which investigated exhaustive search and key exhaustion, we focused on estimation techniques to showcase how secret parameters of the algorithm could be recovered by an attacker. More specifically, we showed that it is possible to exploit statistical artifacts introduced by design by the watermarking system to reverse-engineer the secret construction of a histogram. Additionally, we demonstrated that it is possible to estimate the pseudo-random sequences used to obfuscate the watermarking subspace even if a shuffle has been applied to the cover prior to watermark embedding. Of course, combining all the security mechanisms mentioned in this chapter definitely makes the task of the attacker much more difficult, but not impossible.

These findings clearly call for new designs, especially if the 3D watermarking system is envisioned to be deployed for copyright protection applications. It is unclear at this stage if the whole framework should be revised or if it could be fixed with slight adjustments. For instance, we briefly suggested that the attack against histogram estimation could be somewhat mitigated by introducing weights at the edges. Another potential fix is to disrupt the periodicity that the attacker is exploiting to recover Δ and to use pseudo-randomly shifted bins instead, in a manner similar to what has been done in steganography [SSM07]. Additional investigations are however required to validate that such a modification does not impair robustness performances. In future work, we also plan to refine our attacks. For instance, the Independent Component Analysis (ICA) offers, in theory, the potential to perform the last attack in a single step. Nevertheless, our preliminary investigations in this direction proved unsuccessful, for reasons yet to be explained.

Chapter 9

Resynchronization Approach against Cropping

9.1 Introduction

9.1.1 Review of the state-of-the-art

This chapter explores a mitigation technique to alleviate the sensitivity of the Quadratic Programming (QP) watermarking system (Chapter 6) against mesh cropping.

In blind watermarking, the techniques against desynchronizing attacks, such as cropping, that disrupt the internal convention between the embedder and the decoder (see the registration component in Section 2.2.2) usually fall within one of three categories: (i) using pilot sequences, (ii) defining an invariant carrier, and (iii) applying implicit resynchronization [CMB+07, Chapter 9]. In the first category, a known sequence of symbols, a.k.a. a resynchronization pattern, is embedded alongside the payload, either with interleaving techniques, or by conveying it through a distinct communication channel[1]. At decoding, the registration is enhanced by comparing the expected pattern with the recovered one, so as to appropriately deal with the desynchronizing attack.

In the second category, the watermark carrier is derived from a quantity that is invariant to the desynchronizing attack. As cropping attacks remove an arbitrarily large amount of information, finding a fully invariant quantity is in theory out of reach. Nevertheless, some quantities still exhibit very high resilience to cropping. Most often, the spatial support patch to compute these quantities is small enough so that cropping attacks are likely not to impact them. For instance, in the systems presented in Section 3.2.1, the watermark carriers are defined on local neighborhoods around the vertices. Unless these small patches are cropped, the carrier values are preserved. Locally defined watermark carriers are however often less robust than global ones (see Chapter 3) [WH09].

In the third category, the mesh features, e.g., corners or sharp edges, themselves form the resynchronization pattern, as opposed to the pilot sequence category, where the resynchronization is explicitly set through the specific symbol sequence. In 3D watermarking, implicit resynchronization mechanisms to protect a system against cropping attacks have been used to guide a mesh partitioning. The payload is repeatedly embedded and decoded in every partition. Some of these partitions are expected to be fully recovered after a cropping, as the mesh features are often cropping invariant (mesh features are indeed one example of the aforementioned cropping-invariant quantities). Locally defined carriers are then advantageously replaced with global watermark carriers, such as spectral coefficients, that are independently computed in each partition [OMT02].

[1] in which case the watermark and pilot sequence symbols may not correspond to the same primitive.

In all, watermarking systems relying on implicit resynchronization and payload repetition through partitioning exhibit some robustness against cropping, as well as a robustness against valumetric attacks that is close to the performance of state-of-the-art systems that do not provide resynchronization in case of cropping. For instance, researchers have used umbilical points (points where principal curvatures are equal) to canonically partition the mesh [AM05]. These features are cropping-invariant, and experiments suggest that the partitions are also robust against this attack. Nevertheless, the robustness of umbilical points is an issue and, on spherical or nearly isotropic surface patches, multiple closely located umbilical points would be detected by the estimator on which the watermarking system relies[2]. Furthermore, cropped-out features may impact the partitioning overall; the set of decoded partitions greatly differing from the set of partitions at the embedder side.

These issues were addressed in a subsequent work [RAMC07], where the watermark is instead embedded in the neighborhood of mesh prongs, i.e. feature points that are local maxima of the geodesic distances between all pairs of vertices. Intuitively, these points are located at the protrusions of the mesh and they are more robust against, e.g., noise addition. Finally, the prong-based partitioning and the QP baseline [HRAM09] have been recently combined to leverage both of their strengths [HXYD14]. Nonetheless, these approaches still exhibit many shortcomings.

First, altering the histogram of Euclidean distances in a limited mesh region reduces the robustness of this carrier against valumetric attacks [RAMC07, HXYD14]. The performance might yet be boosted by the payload repetition, but optimizing this complex trade-off has not been researched. Moreover, as the spatial support patch to compute the carrier depends on the features, the watermark robustness is limited by the stability of the feature detection. The watermark designer also cannot control the number and location of the feature points. In all, implicit resynchronization trades some robustness against valumetric attacks for robustness against cropping attacks, in a way that is content-dependent and that cannot be parameterized [RAMC07]. Second, the distance between neighboring features serves as reference in the neighborhood or partitioning construction procedures. When features are lost, the synchronization is jeopardized. Indeed, defining cropping-invariant partitioning methods for meshes is an actively researched open problem. Third, adversaries also have access to the features, which creates security issues. By removing or altering features, one may efficiently disrupt the communication channel, thus tampering with the watermark.

9.1.2 Overview of the Resynchronization Mechanism

In this chapter, we propose a novel synchronization strategy devised to resist cropping attacks with the previously described QP framework. It consists of (i) introducing a specific configuration of secret landmark reference points on the surface during the embedding, and (ii) blindly retrieving the configuration of these landmarks at decoding to recover the synchronization information it conveys. Figure 9-1 depicts the complete watermarking system with the resynchronization components added to the QP framework. As depicted, the last part of the resynchronization is further decomposed into two steps: the inserted landmarks are detected, and, based on their configuration, the decoded geometric information is used in the QP decoder.

The proposed procedure is akin to using a synchronization pattern given by the landmark configuration, but unlike standard pilot sequences, this configuration is content-dependent and transmits geometric metadata on the original watermarked mesh. The resynchronization approach can thus

[2] In a nutshell, the umbilical point detector is parameterized by a detection scale. Pairs of umbilical points could be merged or created between the embedder and the decoder, depending on the effects of, e.g., a valumetric attack on the mesh.

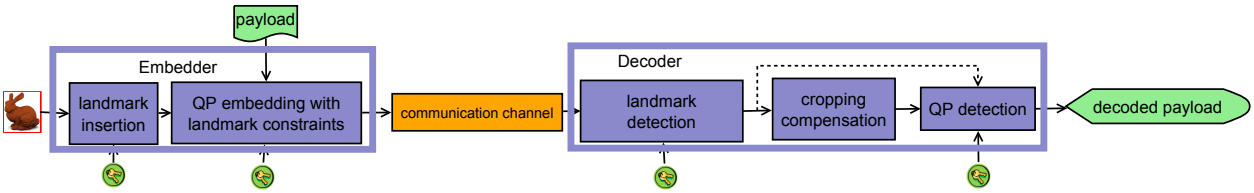


Figure 9-1: Modified QP-based watermarking system with a cropping-resilient resynchronization component.

be seen as a distinct, secondary watermarking system, whose robustness focuses on cropping, and whose capacity is measured by the number of geometric primitives that are sent to the decoder.

Section 9.2 details the landmark characterization, and explains how landmarks are introduced at arbitrary locations in a mesh and recovered in a blind manner. Section 9.3 describes the landmark-based resynchronization approach for the QP framework. Benchmarking results between the baseline system and the enhanced QP watermarking system are reported in Section 9.4. To conclude, future perspectives for this synchronization approach are summarized in Section 9.5.

9.2 Landmark Points Generation

9.2.1 Landmark Definition

Given a field on a surface mesh $\mathbf{f} : \mathcal{M} \to \mathbb{R}^n$, a landmark point $\mathbf{p}$ is generally defined as a point such that $\mathbf{f}(\mathbf{p})$, a.k.a. its 'signature', is in a predefined subspace $\Gamma \subset \mathbb{R}^n$. The set of landmark points is denoted by $\mathcal{L}$. Feature points can be viewed as a specific case of landmarks, where $\mathbf{f}$ is the 'shape descriptor'. When the descriptor is a scalar field, Γ usually writes $[\tau, \infty[$, where τ is the 'detection threshold'. Features identify salient and characteristic points, such as fingertips or sharp corners, and address a variety of applications, such as segmentation [KLT05, VKS05], watermarking [AM05] or registration [PWHY09, SOG09]. Moreover, the compact description of the neighboring patch provided by the signatures allows for efficient shape retrieval techniques [HPPLG11].

The resynchronization approach proposed in this chapter relies upon the creation of landmark points. Since introducing new features would not only be complex, but also raise security and imperceptibility issues, landmarks are defined following a communication with side information technique [EBTG03]. They are characterized as points whose signature is close to an element $\mathbf{t} = [t_1 \dots t_n]^T$ of a lattice $\mathcal{T} \subset \mathbb{R}^n$. t_i is defined with Quantization Index Modulation (QIM) [CW99]: $t_i = k\Delta_i + \epsilon_i(\eta)$ $(k \in \mathbb{Z})$, where $\Delta_i > 0$ denotes a quantization step, and $\epsilon_i(\eta)$ is a secret shifting offset, a.k.a. the 'dither'. $\mathcal{T}$ is commonly referred to as a 'quantization grid'. Let $\mathbf{t}(\mathbf{p})$ be the element of $\mathcal{T}$ closest to $\mathbf{f}(\mathbf{p})$. $t_i(\mathbf{p})$ is computed as:

$$\forall i \in [\![1, n]\!],\ t_i(\mathbf{p}) = \left\lfloor \frac{f_i(\mathbf{p}) - \eta_i(\epsilon)}{\Delta_i} + \frac{1}{2} \right\rfloor \Delta_i + \epsilon_i(\eta). \tag{9.1}$$

In contrast with QIM approaches, no payload is defined in Eq. (9.1).

$\mathcal{L}$ is formally characterized by:

$$\mathcal{L} = \left\{ \mathbf{p} \in \mathcal{M},\ \sqrt{\sum_{i=1}^{n} \big(f_i(\mathbf{p}) - t_i(\mathbf{p})\big)^2} < \bar{\alpha}_{\mathcal{L}} \right\}, \tag{9.2}$$

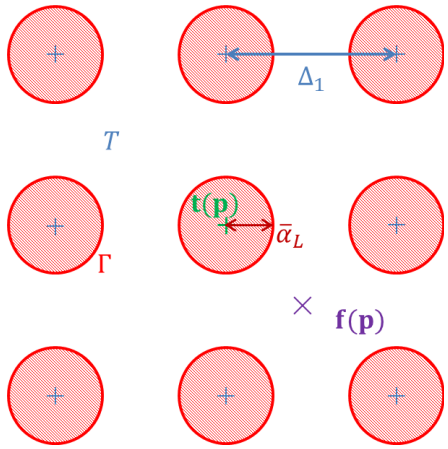


Figure 9-2: Quantization grid $\mathcal{T}$ that determines landmarks in a 2D signature space. For $\mathbf{p}$ to be a landmark, its signature $\mathbf{f}(\mathbf{p})$ has to be within a distance $\bar{\alpha}_{\mathcal{L}}$ of $\mathbf{t}(\mathbf{p})$.

where $\bar{\alpha}_{\mathcal{L}} > 0$ is a threshold distance. From a geometric perspective, Γ is equivalently defined as a union of balls centered at all $\mathbf{t}$ in $\mathcal{T}$, with radius $\bar{\alpha}_{\mathcal{L}}$. Eq. (9.2) induces a notable difference from the distortion-compensation QIM as landmarks are not defined through a cover of $\mathbb{R}^n$, unless the balls overlap. Figure 9-2 illustrates these notions in $\mathbb{R}^2$.

If $\mathbf{f}$ is continuous, $\mathcal{M}$ is connected and $\mathcal{L} \neq \emptyset$, there exists an infinite number of landmarks. In other words, under the first two assumptions, if a landmark point is found, then there exists an entire neighborhood of landmarks around it. This conflicts with the resynchronization approach, which relies upon the ability to accurately locate all landmarks. To overcome this obstacle, landmarks are restricted to the mesh vertices $\mathcal{V}$. In the following, $\mathbf{p}$ is replaced with v.

Landmarks differ from feature points as their signature $\mathbf{f}(v)$ is not extreme according to some metric, but instead approximately aligned with the quantization grid $\mathcal{T}$. The main advantage of landmarks over features is that they are less perceptible and easier to generate. The drawback is that their stability is smaller.

The next obstacle to the creation of landmarks lies within the selection of $\mathbf{f}$ and the definition of the surface patch to compute the signature $\mathbf{f}(v)$. For most applications, the issue of inverting $\mathbf{f}$ does not exist. The creation of landmarks is however a watermarking problem. One needs to find a procedure to modify the geometry so as to adjust the signature of v within the positive detection subspace $\mathcal{B}(\mathbf{t}(v), \bar{\alpha}_{\mathcal{L}})$, a.k.a. the ball of radius $\bar{\alpha}_{\mathcal{L}}$ and centered at $\mathbf{t}(v)$. When the spatial support patch to compute the signature is large, for instance if $\mathbf{f}(v)$ depends on vertices that are far away on the surface, it becomes very challenging to find a fusion function. Moreover, altering large parts of the mesh to adjust a single signature leads to complex interdependency issues. This explains why only limited research has looked into watermarking the signature and the location of feature points.

Among the multiplicity of mesh descriptors, we consider the first component in the extension of the Harris detector to triangle meshes [SB11]. In short, this descriptor identifies feature points by: (i) fitting a paraboloid model to a local surface patch, (ii) deriving a matrix of robust parameters from the fitting coefficients, and (iii) computing a detection score based on the eigenvalues of this matrix. The detector has a small computational complexity and interdependency issues are limited thanks to the small size of the surface patch, i.e. a small local neighborhood around the vertex.

The signature of v at $\mathbf{p}$ is computed with three steps. First, the vertex positions in the neighborhood of v are represented in a local frame, centered at $\mathbf{p}$, and whose axes are given by a Principal

Component Analysis (PCA) (the z-axis is consistently oriented outward the mesh). Second, a parametric model of a paraboloid surface $z(x, y) = ax^2 + bxy + cy^2 + dx + ey + f$ is fitted using linear least squares, which requires at least 6 points. Third, the signature is taken as:

$$\mathbf{f}(v) = \left(\frac{a^2 + 2b^2}{(\nu|a + c|)^2}, \frac{c^2 + 2b^2}{(\nu|a + c|)^2} \right), \tag{9.3}$$

which is two-dimensional ($n = 2$). ν is a normalization factor, empirically set to 5×10^{-2}. The other coefficients of the parametric model are discarded.

This signature is invariant to rigid transform and uniform scaling. It is also cropping oblivious, provided that the attack does not impact the local neighborhood of v. The robustness to other connectivity distortions, such as subdivision, simplification or remeshing, might benefit from using an integral formulation of the PCA [GAP08], as well as using a Euclidean [SB11] or even a geodesic neighborhood for the local surface patch. Nevertheless, this would make for a more complex fusion function. Neighborhoods defined through Euclidean or geodesic distances additionally yield some issues regarding their size, as this parameter is often altered by cropping attacks (see the instability results against pose and cropping in Section 4.3.1).

For simplicity, the local neighborhood is taken as the r-ring neighborhood. Since both elements of the signature measure similar quantities, respectively in the x and y directions, the quantization steps Δ_i in Eq. (9.1) are set to the same constant Δ.

9.2.2 Creating New Landmarks

To turn a vertex v into a landmark, a state-of-the-art active-set method (`fmincon` in MATLAB [The13]) is used to relocate the vertex positions in the ring neighborhood $\mathcal{N}_r(v)$. This generic optimization method allows for non-linear constraints, which are set to the following function:

$$\mathbf{c} = \begin{cases} \mathbf{0} & \text{if } \|\mathbf{f}(v) - \mathbf{t}(v)\| < \bar{\alpha}_{\mathcal{L}} \\ (|f_1(v) - t_1(v)|, |f_2(v) - t_2(v)|)^T & \text{otherwise.} \end{cases} \tag{9.4}$$

The solver minimizes the distortion ω, set to the Square Error (SE):

$$\omega(\mathbf{P}') = \sum_{v_i \in \mathcal{N}_r(v)} \|\mathbf{p}_i - \mathbf{p}'_i\|^2, \tag{9.5}$$

where $\mathbf{P}'$ are new vertex positions, under the constraint that $\mathbf{c}$ is the null vector. The SE metric ensures that small distortions are introduced, but it does not preserve some properties of the mesh, such as its manifoldness, or the smoothness of the underlying surface. Perceptually-correlated metrics [CLL+13] may help achieving this goal. Nonetheless, their principal advantage is to leverage masking effects at mesh levels by, e.g., locally adapting the embedding distortion to the geometry. As they do not explicitly enforce the aforementioned properties, and as the alteration is restricted to a limited ring neighborhood, only marginal benefits may come from using more evolved ω (e.g. using Eq. (6.39) or Eq. (6.41) from Section 6.7).

Since the computation of $\mathbf{f}$ involves a PCA, one may be tempted to simplify the procedure above, and only perform the minimization on the coordinates expressed in the local coordinate frame. The results could then be mapped back into the canonical referential. Such a split is routinely used when estimating quantities through, e.g., least-squares minimization [CP08]. When altering the surface however, this yields a causality issue. Empirical results confirm that the PCA before and after the alteration is indeed inconsistent.

For the one ring neighborhood ($r = 1$), the number of vertices in the fitting procedure is small, and two problems are more likely to occur: (i) less than 6 vertices, required for the least-squares fitting, are available, or (ii) a specific vertex configuration leads to an ill-defined model, e.g., when symmetries are present. To mitigate these problems, r is set to 2. Multiple landmarks can thus be independently introduced, provided that their 2-ring neighborhoods do not overlap.

In the following, $\bar{\alpha}_{\mathcal{L}}$ is defined as $(1 - \alpha_{\mathcal{L}})\Delta/2$, where $\alpha_{\mathcal{L}}$ is a distortion compensation parameter taken in $(0, 1)$. For larger $\alpha_{\mathcal{L}}$, landmark signatures get closer to the quantization grid, and creating landmarks yields larger alterations.

9.2.3 Blind Recovery of Landmarks

A key property of landmarks is the ability for the decoder to detect them in a blind manner. Given a mesh in which landmarks are to be recovered, the signature associated with each vertex is estimated. Each estimation is assigned a score $\tilde{s}(v) \leq 1$ with $\tilde{s}(v) = 1 - 2d(v)\Delta^{-1}$, where $d(v)$ is the distance between the signature and the nearest codeword in $\mathcal{T}$. Based on $\tilde{s}(v)$, determining whether v corresponds to a landmark or not is a binary classification problem. Note that in this configuration, and without any attack, the detection threshold τ to retrieve all possible landmarks is: $\tau = \alpha_{\mathcal{L}}$.

Two-Dimensional Signature

The ability of the decoder to recover landmarks is experimentally measured on the *bunny* mesh for increasing values of $\alpha_{\mathcal{L}}$ within $[0.75, 0.98]$. In each trial, 50 random vertices with non-overlapping 2-ring neighborhoods are turned into landmarks following the creation procedure. Δ is set to 1. The results of the blind detection are reported in Figure 9-3(a) and depicted with Receiver Operating Characteristic (ROC) curves.

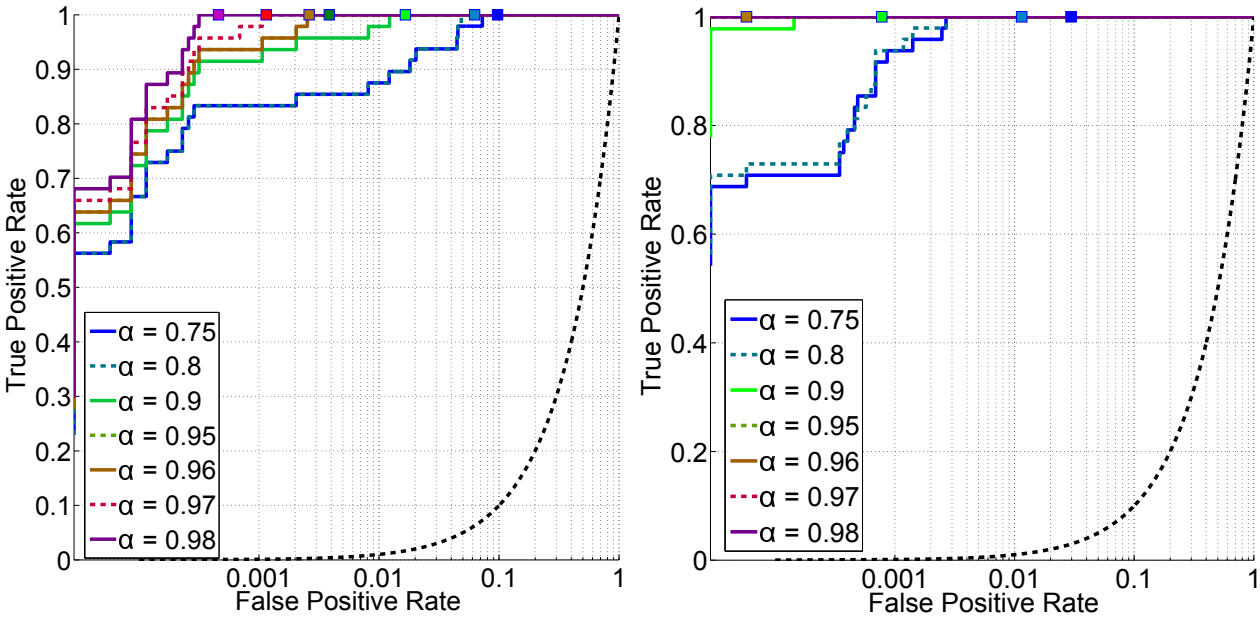
The True Positive Rate (TPR) corresponds to the ratio of recovered landmarks over their total number (50), depending on the detection threshold τ. Since there are $n_v \approx 3.5 \times 10^4$ vertices in the mesh, the prior distributions are greatly unbalanced ($50 \ll n_v$), and the False Positive Rate (FPR) is depicted in log-scale. The success of the landmark retrieval is measured for low FPR. When the FPR reaches 10^{-3}, the precision[3] of the classification is at most 40%, i.e., more than half of the detected landmarks are incorrect. As indicated by the relative positions of the operating point at $\tau = \alpha_{\mathcal{L}}$ (depicted by squared markers) for the different curves, $\alpha_{\mathcal{L}} = 0.98$ achieves the most promising results. Still, the performance for very small FPR is low. To address this problem, a refined approach is explored next.

Nested Signatures

Intuitively, the larger the size n of the signature, i.e., the more dimensions in the quantization grid, the lower the FPR. It is indeed less likely for a larger number of model parameters to be within the detection ball $\mathcal{B}(\mathbf{t}(v), \bar{\alpha}_{\mathcal{L}})$. Instead of changing the two-dimensional signature, we apply a matryoshka principle to increase the number of dimensions from $n = 2$ to $n = 4$ by nesting independent signatures at the same location.

Landmarks are defined using nested ring neighborhoods, in which the fitting parameters are sequentially modified. The nested landmark creation starts by applying the insertion procedure for the 2D signature in the 2-ring neighborhood. A second set of model parameters is then estimated using the 3-ring neighborhood. The positions of the vertices in the 3-ring but not in the 2-ring are

[3]The precision of a binary classifier is the ratio of true positives over the total number of positive outcomes.



(a) $r = 2$ (b) $r = \{2, 3\}$

Figure 9-3: Performance of the blind detection of landmarks on the *bunny* mesh, measured through ROC curves using increasing landmark strengths $\alpha_{\mathcal{L}}$. In 9-3(a), landmarks are defined by quantizing a 2D signature fitted on the 2-ring neighborhood. In 9-3(b), this procedure is applied consecutively to the 2 and 3 rings to create nested modified neighborhoods. The dotted black curve represents the theoretical performance of a random classifier. Operating points at $\tau = \alpha_{\mathcal{L}}$ are indicated with squares.

finally altered to turn the second signature, derived from this second set of model parameters, into a landmark signature. This last restriction prevents any interference issue. In practice, the second alteration procedure, applied to the 3-ring, is almost identical to the original insertion. The only difference is that the solver has limited degrees of freedom, as the coordinates of the vertices in the 1 and 2 rings are no longer unknowns.

Apart from the fitting and the alteration neighborhoods, the two consecutive modifications share the same parameters, and are set equal to the ones previously used. Intuitively, the matryoshka solution enables reaching a larger signature space by increasing the size of the support patch to compute $\mathbf{f}$ (spatial extension of the parametric model) instead of n (degree of the parametric model). The performance of this approach is depicted by Figure 9-3(b). In this context, the score $\tilde{s}(v)$ is taken as:

$$\tilde{s}(v) = \frac{1}{\sqrt{2}} \sqrt{\tilde{s}_2(v)^2 + \tilde{s}_3(v)^2}, \tag{9.6}$$

where $\tilde{s}_k(v)$ is the previously defined score for the k-ring neighborhood ($k \in \{2, 3\}$). The decision threshold τ is then identically set to $\alpha_{\mathcal{L}}$. For $\alpha_{\mathcal{L}} \geq 0.8$, the largest benefits are achieved, as the TPR is boosted for very small FPR.

Discussion

The drawbacks of larger quantization strengths $\alpha_{\mathcal{L}}$ are a smaller fidelity and the increased probability for the solver not to find a solution to the minimization problem. Regarding the latter issue,

2 failures occurs at $\alpha_{\mathcal{L}} = 0.75$ and 3 at $\alpha_{\mathcal{L}} = 0.98$ (over the 50 initially selected landmarks) for the initial 2-ring neighborhood alteration. For the nested approach, approximately 5 failures always occur. Regarding the first issue, global distortion metrics are uninformative, as the alterations are sparse on the surface. We then use the Mesh Structural Distortion Measure (MSDM) [LDD+06] without aggregating the local distortions over all the vertices. Experimentally, the average local MSDM at the landmarks marginally increases when varying $\alpha_{\mathcal{L}}$. It is around 10^{-9} (negligible) for the 2-ring alteration, while it stays below 10^{-4} for the nested procedures.

To conclude these observations on the first round of experiments, using the nested landmark creation, $\alpha_{\mathcal{L}} = 0.98$ and a detection threshold $\tau = \alpha_{\mathcal{L}}$ provides the means for a blind decoding that recovers all landmarks and no false positives (as indicated by the lack of a purple operating point on Figure 9-3(b)).

If needed, two mechanisms could be added to further improve the precision of the detection. One, the procedure to create nested multiple signatures can be repeated. In this case, the paraboloid fitting should be modified to reduce the embedding distortion, as it becomes prohibitive for larger neighborhoods. Two, one could altogether prevent false positives from being emitted, using the embedder to move the signatures of unwanted landmarks outside the detection regions.

9.3 Resynchronization based on Landmarks

In the QP framework, the key synchronization information are the position of the center of mass $\mathbf{g}$, and the upper and lower bounds on the radial distances, a.k.a. $m = \min(\boldsymbol{\rho})$ and $M = \max(\boldsymbol{\rho})$. When one of these quantities is altered by an attack, the decoder cannot recover the carrier, whose elements interfere with each another.

A major obstacle for 3D watermarking is that vertex locations cannot be expressed in a canonical way in a content-independent referential. We could indeed independently transmit $\mathbf{g}$, m and M from the embedder to the decoder, thereby making the decoding semi-blind. For real-life traitor-tracing applications, the memory footprint of this meta-data is sufficiently small. In case of cropping, the decoder would use the transmitted $\mathbf{g}$ and disregard its own estimation $\hat{\mathbf{g}}$, a.k.a. the center of mass of the received cropped mesh. However, the radial distances computed with regard to $\mathbf{g}$ would be erroneous in case of rigid transforms, which would replace cropping as desynchronizing attacks for the watermarking system. Semi-blind decoding is in itself ineffective, as it assumes the decoder can guess which attack has occurred. Moreover, for combinations of attacks, such as cropping followed by rigid transform, the semi-blind decoder cannot recover the payload as both $\mathbf{g}$ (transmitted) and $\hat{\mathbf{g}}$ (cropped and translated) yield erroneous payloads.

The resynchronization approach proposed in this chapter embeds the critical synchronization information within the mesh using a pattern created with landmark points. The payload is then routinely embedded using the QP framework. This synchronization mechanism amounts to creating a new geometric structure (the synchronization pattern) on the mesh, from which $\mathbf{g}$, m and M can be derived in a manner that is robust to cropping. In the decoder, one can estimate e.g., the center of mass, from either the whole mesh or the synchronization pattern.

First, we only show how the center of mass $\mathbf{g}$ can be transmitted by arranging secret landmarks onto a secret sphere (Section 9.3.1). We then explain how to exploit the *quality* of the identified landmarks to automatically switch between alternate resynchronization strategies depending on the attacking context (Section 9.3.2). Indeed, the estimate $\hat{\mathbf{g}}$ is a global quantity that shows a large stability against valumetric attacks. On the other hand, the landmark-based estimate, denoted by $\hat{\mathbf{g}}_{\hat{\mathcal{L}}}$, is only robust to cropping. To decode the payload and possibly resynchronization the carrier, only one of two estimates is automatically used, depending on its quality.

Finally, several secret landmarks configurations are combined to retrieve all the critical resynchronization information $\mathbf{g}$, m and M on the receiver side (Section 9.3.3).

9.3.1 Embedder Side for Resynchronizing g

The synchronization pattern created on the mesh $\mathcal{M}$ is the intersection $\mathcal{I}(r)$ between $\mathcal{M}$ and the sphere $\mathcal{S}(\mathbf{g}, r)$, which is of zero measure in general. Landmarks are introduced in $\mathcal{I}(r)$, so that the decoder may recover $\mathbf{g}$ as the center of the sphere.

For watertight meshes and a radius r in $[m, M$, $\mathcal{I}(r)$ is non empty set, but unlikely to contain any vertex. Since landmarks are only defined on vertices, $\mathcal{I}(r)$ is replaced with the intersection set $\mathcal{I}(r, \epsilon_{\mathcal{S}})$:

$$\mathcal{I}(r, \epsilon_{\mathcal{S}}) = \{v_i \in \mathcal{V} \mid \mathbf{p}_i \in \{\mathcal{M} \cap \mathcal{B}(\mathbf{g}, r - \frac{1}{2}\epsilon_{\mathcal{S}})\} \setminus \{\mathcal{M} \cap \mathcal{S}(\mathbf{g}, r - \frac{1}{2}\epsilon_{\mathcal{S}}\}, \tag{9.7}$$

where $\epsilon_{\mathcal{S}} > 0$ controls the size of the intersection.

The elements of $\mathcal{I}(r, \epsilon_{\mathcal{S}})$ are referred to as 'intersection vertices'. It is likely that they are located next to one another, and that their ring neighborhoods overlap. Because creating several landmarks requires disjoint support patches, the set $\mathcal{L}$ of landmarks is a subset of $\mathcal{I}(r, \epsilon_{\mathcal{S}})$. The two constraints on the construction of $\mathcal{L}$ are: (i) $\mathbf{g}$ is well-defined as the center of landmarks, i.e., they are not coplanar; and (ii) the synchronization pattern has at least n_L elements. This corresponds to a sphere packing problem on the mesh surface, which is solved by the following greedy procedure, as summarized in Algorithm 4.

Creating landmarks

The size of the set $\mathcal{I}(r, \epsilon_{\mathcal{S}})$ is estimated for multiple radii that sample $[m, M]$ with a step $\epsilon_{\mathcal{S}}$. This amounts to computing the histogram of radial distances with bins spaced by $\epsilon_{\mathcal{S}}$. $\epsilon_{\mathcal{S}}$ is set to 0.1% of $M - m$. The construction of $\mathcal{L}$ is then sequentially tried for each intersection set, ordered according to their size.

In each test of an intersection set, the number of collisions $n_c(v) \geq 0$ between an intersection vertex and all the others intersection vertices is:

$$n_c(v) = |\{v_j \in \mathcal{I}(r, \epsilon_{\mathcal{S}}) \setminus v \mid \mathcal{N}_3(v) \cap \mathcal{N}_3(v_j) \neq \emptyset\}|.$$

$\mathcal{L}$ is iteratively filled, starting from the vertices with the smallest n_c. If an intersection vertex has a collision with any element already in $\mathcal{L}$ it is discarded. When $|\mathcal{L}|$ equals n_L, the non-coplanar constraint is checked. If it is violated, or if $|\mathcal{L}|$ cannot reach n_L, the next intersection set is tested. Adding first the intersection vertices for which n_c is null experimentally scatters landmarks all over the mesh. This decreases the likelihood that a cropping attack impacts all of them.

Before applying the nested landmark creation procedure, all vertices in $\mathcal{L}$ are projected onto $\mathcal{S}(\mathbf{g}, r)$ to ensure that $\mathbf{g}$ coincides with the center of the spherical synchronization pattern. During the alteration of the local neighborhoods, the landmarks are then constrained to stay on $\mathcal{S}(\mathbf{g}, r)$. As explained in Section 9.2.3, the optimization procedure to create landmarks through nested signatures does not always succeed. $\mathcal{L}$ is thus usually filled with more than the minimum n_L elements. If the creation does not succeed for at least n_L vertices, $\mathcal{L}$ is modified by replacing the candidates leading to failures with new ones from $\mathcal{I}(r, \epsilon_{\mathcal{S}})$, while maintaining the non-overlapping constraints. If this attempt also fails, the next intersection set is tested. Between each test, all the potentially introduced alterations on the mesh surface are removed. For simplicity, $\mathcal{L}$ henceforth denotes the set of the n_L successfully created landmarks that have been moved into $\mathcal{I}(r)$. $\mathcal{L}$ is depicted in Figures 9-5(a) and 9-5(b) for a toy 2D example.

Algorithm 4 Main steps in the embedding of the spherical pattern of landmarks in the mesh.

procedure SPHERE PATTERN EMBEDDING(Input mesh $\mathcal{M}$; $\epsilon_{\mathcal{S}}$ intersection size; $\boldsymbol{\rho}$ radial distances; n_L number of landmarks.)
  Let $\mathcal{I}^{\star}(\epsilon_{\mathcal{S}})$ be the list of all $\mathcal{I}(r, \epsilon_{\mathcal{S}})$, for r in $[\min(\boldsymbol{\rho}), \max(\boldsymbol{\rho})]$ sampled by $\epsilon_{\mathcal{S}}$, sorted according to their number of elements (descending order).
  while $\mathcal{I}^{\star}(\epsilon_{\mathcal{S}}) \neq \emptyset$ **do**
    Compute the number of collisions $n_c(v)$ for all $v \in \mathcal{I}(r, \epsilon_{\mathcal{S}})$ (head of $\mathcal{I}^{\star}(\epsilon_{\mathcal{S}})$).
    Iteratively fill $\mathcal{L}$ with n_L vertices in $\mathcal{I}(r, \epsilon_{\mathcal{S}})$ sorted from lowest to largest n_c, avoiding any overlap in the local neighborhoods.
    Perform the landmark spherical pattern creation.
    if $|\mathcal{L}| > n_L$ **then**
      break
    else
      Discard vertices for which the landmark creation failed and add remaining vertices from $\mathcal{I}(r, \epsilon_{\mathcal{S}})$ avoiding any overlap in the local neighborhoods.
      Perform the landmark spherical pattern creation.
      if $|\mathcal{L}| > n_L$ **then**
        break
      else
        Pop $\mathcal{I}(r, \epsilon_{\mathcal{S}})$ from $\mathcal{I}^{\star}$, reset $\mathcal{L}$ to $\emptyset$ and remove all alterations of $\mathcal{M}$.
      end if
    end if
  end while
  return Mesh with the embedded spherical pattern of landmarks.
end procedure

Payload Embedding

The embedder finally performs the payload embedding with the QP framework. The locations of the landmark vertices as well as their 3-ring neighborhood are unchanged, thanks to a simple modification of the constraints in Eq. (6.13). Figure 9-5(c) depicts this step of the resynchronization approach. These synchronization vertices only represent a small part of $\mathcal{V}$. In the mesh database in Table D.1, the ratio of fixed vertices varies from 0.5% to 4.8%. The robustness of the watermark carrier in the QP framework is therefore not reduced. The stability in the position of the center of mass before and after the synchronization also does not need to be explicitly enforced; the causality issue being negligible.

Security Considerations

This chapter does not present a thorough security assessment, and the proposed algorithms as well as the parameter settings are set to achieve the best detection performance without taking into account the security.

For instance, landmarks are defined with a lattice-based watermarking approach that relies on secret shifting offsets for security. In the Known-Message Attack (KMA) context, an adversary may use multiple observations to statistically infer the secrets [PFCnPG05]. Nevertheless, the small number of landmarks compared with the overall number of vertices may prove beneficial in this regard. Intuitively, the impact of the landmark creation on the distribution of the signatures in $\mathbb{R}^{2\times 2}$ is small. In practical attacks on dirty paper watermarking [PFPGCn06], mixing a large

number of non-watermarked observations with a few watermarked ones still yields large obstacles for the adversary.

Moreover, because the previous procedures are deterministic, they would need to be modified when the security is an issue. The order in which the intersection sets are tested should for instance be shuffled.

9.3.2 Decoder Side for Resynchronizing g

At decoding, the resynchronization approach summarizes as follows. First, landmark vertices are blindly retrieved, as described in Section 9.2.3. This amounts to estimating the scores $\tilde{s}(v_i)$ defined in Eq. (9.6) for all the mesh vertices, and setting the threshold τ to detect the landmarks $\hat{\mathcal{L}}$. Second, the decoder estimates $\hat{\mathbf{g}}_{\hat{\mathcal{L}}}$, a.k.a. the center of the sphere on which the detected landmarks are located. The decoder then switches between $\hat{\mathbf{g}}_{\hat{\mathcal{L}}}$ and $\hat{\mathbf{g}}$ when needed, since (i) both have been aligned in the embedder, and (ii) the former is robust against cropping while the latter is robust against valumetric attacks. In the following, τ is conservatively set as in Section 9.2.3 to $\alpha_{\mathcal{L}} = 0.98\%$; the detection region does not allow for the landmark signatures to be moved outside the embedding region.

Dealing with False Landmarks

In general, some of the detected landmarks may be false positives and some correct landmarks may be missed (false negatives): $\mathcal{L} \neq \hat{\mathcal{L}}$. Estimating $\hat{\mathbf{g}}_{\hat{\mathcal{L}}}$ through least-squares sphere fitting is unreliable, as this category of estimator is sensitive to outliers. False positives in $\hat{\mathcal{L}}$ are indeed randomly located on the mesh surface. To alleviate this first issue, a RANdom SAmple Consensus (RANSAC) approach is used [FB81]. Its parameters are: the number of trials $n_t = 250$, the minimum number of landmarks to initially estimate the sphere model (4 landmarks), a fitting threshold τ_{RANSAC} (set by default to 10^{-3}) to select the landmarks that are close enough to the initial fit, and the minimum number of landmarks that are close enough so that the model is to be considered as relevant. This last parameter is set to 75% of $|\hat{\mathcal{L}}|$. Finally, the RANSAC model is taken as the least-squares sphere fitting, and the weight associated to an element of $\hat{\mathcal{L}}$ is its distance to the fitted sphere model. This procedure outputs either $\hat{\mathbf{g}}_{\hat{\mathcal{L}}}$ or fails to find any suitable model from the input detected landmarks. False negatives in the decoding are less of an issue: as long as at least 4 landmarks are correctly retrieved, the sphere fitting is well-defined. The blind retrieval of landmarks, the rejection of false positives through the RANSAC and the correct retrieval of the original center of mass are illustrated in Figure 9-5(g) for a toy 2D example. Figure 9-4 depicts the actual result of the resynchronization approach on a cropped *bunny*.

Automatic Resynchronization Switch

Most existing approaches that are robust against cropping have limited performance against valumetric attacks: they are often hindered by the low stability of the primitives relied upon by the synchronization, such as umbilical points [AM05] or prongs [RAMC07]. In other words, these systems exhibit an implicit trade-off in robustness between synchronization and valumetric attacks[4],

[4]This observation corresponds to systems with globally defined watermark carriers and relying upon a resynchronization mechanism against cropping. For systems with locally-defined watermark carriers, the robustness of the carrier against valumetric attacks is intrinsically small, thereby also achieving a similar balance between the two types of attacks.

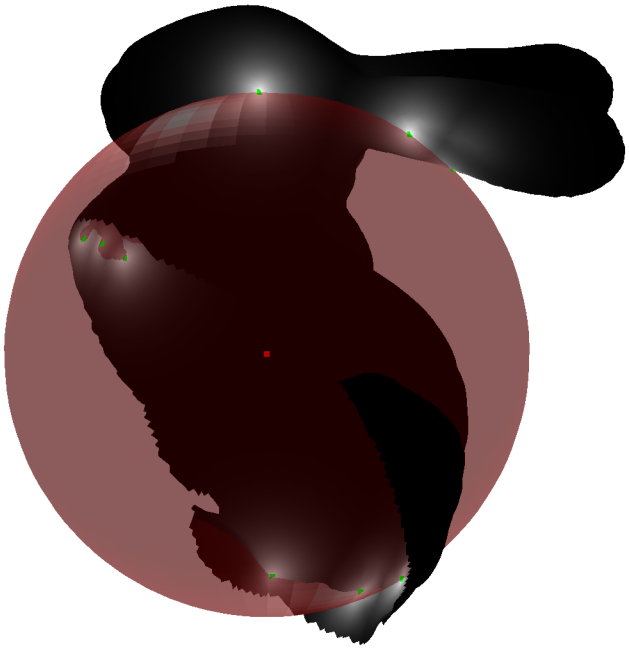


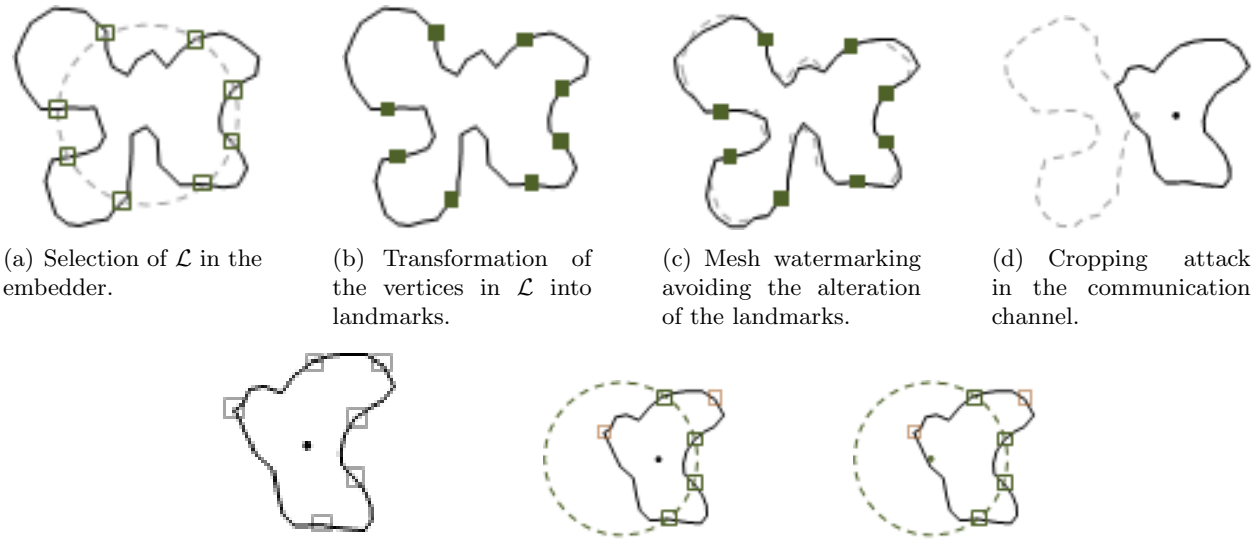
Figure 9-4: Blind recovery of the center of mass (red dot) of the *bunny*. The landmarks inserted at embedding (indicated by white spots on the mesh) are used to fit a sphere whose center coincides by construction with the original center of mass, which cannot be estimated from the cropped mesh (40% cropping ratio).

and this balance is delicate to control. In contrast, the landmark resynchronization decoding provides an explicit means to control this trade-off through τ and the routine sensitivity vs. specificity of the landmark detection.

In case of, e.g., a noise addition, the signatures of the landmarks in $\mathcal{L}$ are altered, and may be outside the embedding regions. To recover the original landmarks in $\hat{\mathcal{L}}$, the detection threshold τ needs to be smaller than $\alpha_{\mathcal{L}}$, thereby increasing the FPR. Moreover, the stability of the landmark center is smaller than the stability of the mesh center of mass: $\|\mathbf{g} - \hat{\mathbf{g}}\| \ll \|\mathbf{g}_{\mathcal{L}} - \hat{\mathbf{g}}_{\mathcal{L}}\|$, since $n_L \ll n_v$. In this situation, using $\hat{\mathbf{g}}_{\hat{\mathcal{L}}}$ instead of $\hat{\mathbf{g}}$ may therefore significantly hamper the performance. Conversely, in case of a cropping, $\hat{\mathbf{g}}_{\hat{\mathcal{L}}}$ may be advantageously used instead.

Setting $\tau = \alpha_{\mathcal{L}}$ trades sensitivity for specificity in the landmark detection, and ensures that the performance against valumetric attacks are not impacted by the resynchronization. When this type of attack occurs, all $\tilde{s}$ are lower than τ, and the decoder bypasses the resynchronization mechanism. The QP decoding uses the standard estimation $\hat{\mathbf{g}}$. When at least 4 scores are above τ and the RANSAC converges, the QP decoder uses $\hat{\mathbf{g}}_{\hat{L}}$, as it is likely to be the center of true landmarks, and its position is close to $\mathbf{g}_L$ (dismissing rigid transforms). A cropping attack will then be thwarted.

In summary, the decoding resynchronization identifies landmark and use their center instead of the mesh center of mass, depending on the confidence threshold τ. τ grants the decoder the ability to distinguish between types of attacks, so that the robustness performance against valumetric attacks is not limited by the resynchronization procedure. This overcomes the limitations of, e.g., a straightforward semi-blind decoding relying on the transmission of $\mathbf{g}$.



(a) Selection of $\mathcal{L}$ in the embedder.

(b) Transformation of the vertices in $\mathcal{L}$ into landmarks.

(c) Mesh watermarking avoiding the alteration of the landmarks.

(d) Cropping attack in the communication channel.

(e) At decoding, blind landmark detection.

(f) False positives rejection by RANSAC

(g) Estimation of the initial center of mass used in the QP framework

Figure 9-5: Steps of the spherical resynchronization approach using a spherical pattern of landmark points. 9-5(a): a set of vertices close to $\mathcal{S}(\mathbf{g}, r)$ are selected to create landmarks. r is computed through the iterative procedure described in Section 9.3.1. 9-5(b): the vertices are (all successfully) turned into landmarks, according to the nested landmark creation procedure. 9-5(c): the QP framework to watermark the mesh is applied. The introduced landmarks and their neighborhoods are however not relocated. 9-5(d): in the communication channel, parts of the objects are cropped and the location of the center of mass is strongly affected. 9-5(e): in the decoder, all the potential landmark vertices are blindly recovered. 9-5(f): false positives (red squares) are detected by relying on the assumption that landmarks should be on a sphere and using a RANSAC approach to fit a least-squares spherical model. The output of the RANSAC is a sphere whose center coincides with the center of mass of the original mesh, prior to cropping 9-5(g).

9.3.3 Conveying Additional Resynchronization Information

To transmit the remaining critical information for resynchronization, namely the bounds of the histogram m and M, a straightforward approach consists in using two distinct spherical synchronization patterns $\mathcal{S}(\mathbf{g}, r_1)$ and $\mathcal{S}(\mathbf{g}, r_2)$ and to exploit the two radii r_1 and r_2 to encode m and M. First, two distinct types of landmarks need to be defined to correctly associate landmarks with their resynchronization patterns. This is achieved by relying on two distinct lattices $\mathcal{T}_1$ and $\mathcal{T}_2$ in a manner similar to binary QIM [CW99]. Next, the radii are set as: $[r_1\, r_2]^T = \mathbf{L}[m\, M]^T$, where $\mathbf{L}$ is a full-rank mixing matrix. In this study, the first row of $\mathbf{L}$ is set to $[0.5\, 0.5]$ and the second is set to $[0.6\, 0.4]$. The sequential constructions of the set of candidate landmarks $\mathcal{L}_1$ and $\mathcal{L}_2$ are then modified to guarantee that there is no neighborhood overlap between both sets of vertices. On the receiver side, the information derived from the spheres estimated using RANSAC ($\hat{\mathbf{g}}_{\hat{\mathcal{L}}}$, $\hat{m}_{\hat{\mathcal{L}}}$, and $\hat{M}_{\hat{\mathcal{L}}}$, where $\hat{\mathcal{L}} = \hat{\mathcal{L}}_1 \cup \hat{\mathcal{L}}_2$) is used for watermark decoding rather than the quantities directly derived from the mesh. When one of the two resynchronization patterns cannot be recovered, the center of the only remaining sphere $\hat{\mathbf{g}}_{\hat{\mathcal{L}}}$ is used, and $\hat{M}$ and $\hat{m}$ are directly computed from the mesh.

9.4 Benchmarking of the Watermarking System

The performance of the QP watermarking framework with and without the resynchronization approach is benchmarked following the same protocol as in Chapter 7. Four different variants of the baseline system are studied:

1. without any resynchronization;
2. with a single resynchronization pattern encoding $\mathbf{g}$;
3. with two resynchronization patterns encoding $\mathbf{g}$, m, and M;
4. with the ground truth resynchronization information, a.k.a. semi-blind system.

The last variant provides a theoretical upper bound on the performances that could be achieved. First, the embedding strength α[5] of all systems are adjusted so that have equal fidelity. The MSDM is upper-bounded by 0.15 and the Root Mean Square (RMS) is set to 0.08% (this second upper-bound is in practice never reached). In all cases, the same fidelity is for identical α, which confirms the marginal perceptual impact of the landmark creation, and the insignificance of constraining a small ratio of the vertices during the QP embedding. The meshes in Table D.1 are then randomly watermarked 5 times, then attacked 7 times (for non-deterministic attacks) before being input to the decoder. Figure 9-6 depicts the performance of both systems against noise addition, quantization, smoothing, cropping, and triangle soup attacks.

In line with previously reported benchmarking results, the baseline system is extremely sensitive to cropping due to the loss of critical information ($\mathbf{g}$, m, and M). In contrast, the solid green curve using ground truth side information indicates how much gain could be achieved with an efficient resynchronization module. The BER could remain below 5% even for strong cropping attacks. Adding a single resynchronization pattern to recover the original center of mass already provides significant performances improvement. As a matter of fact, the BER actually alternates between 0% and 50% depending on whether the bounds of the histogram m and M are recovered or not. The stronger the cropping attack is, the more likely it is that these bounds are lost. As a result, incorporating the second resynchronization pattern to convey m and M further improves robustness. The dotted red curve remains very close to the lower bound provided by the ground truth, until it rockets above 10% deletion. For such strong cropping attacks, many landmarks are lost. As a result, the resynchronization approach is less likely to recover both patterns and the bounds $\hat{m}$ and $\hat{M}$ used for watermark decoding are then corrupted. This explains why the curves for the two variants with resynchronization patterns nearly coincide for large cropping ratios.

For all the other attacks, the performance of the baseline QP and the performance of the QP enhanced with the resynchronization component are similar. This validates that setting $\tau = \alpha_{\mathcal{L}}$ effectively prevents the synchronization module from jeopardizing the established large robustness of the baseline watermarking system against valumetric-only attacks. As depicted on Figure 7-4, the baseline QP is not robust against simplification and refinement attacks. This is also the case for the landmarks (because of, e.g., their neighborhood definition). Thus, when using a different center of mass definition, as proposed in Chapter 6, the resynchronization component would also be bypassed in the decoder.

In addition to these promising results, the resynchronization approach handles cropping as well as combinations of rigid transforms and cropping. However, if one were to combine valumetric and synchronization attacks, the decoder would systematically drop the resynchronization steps in

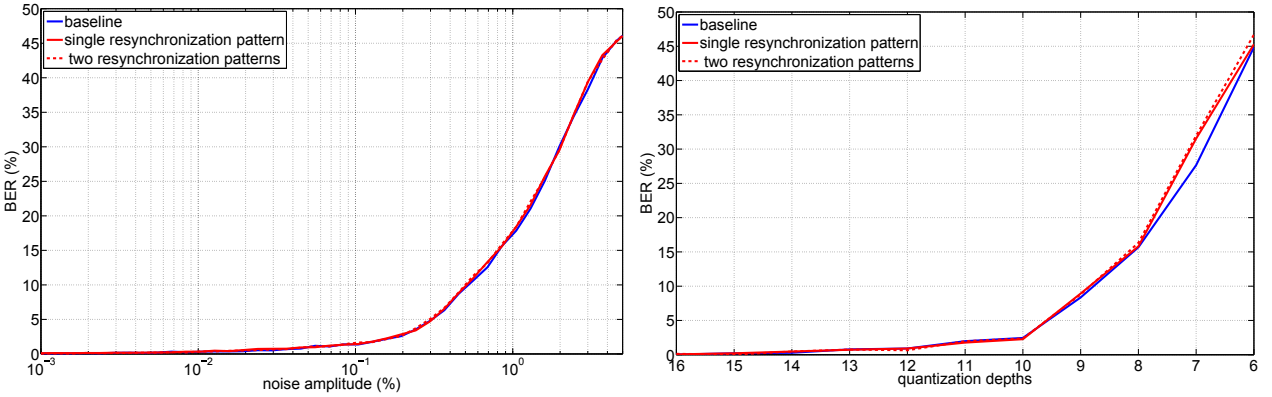
[5]Recall that α controls the amount by which the average radial distances are altered.

an attempt to benefit from the robustness of the center of mass derived from a larger number of vertices. This is the main limitation of the proposed watermarking system.

9.5 Conclusion and Future Perspectives

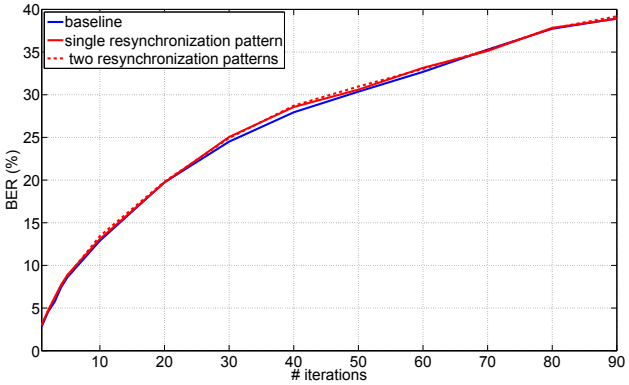
The resynchronization approach for the QP framework relies upon a few landmark vertices introduced by the embedder in a spherical configuration. The decoder blindly identifies this configuration and computes its center. It can then resynchronize the QP watermark carrier, a.k.a. the radial distances, so as to deal with a cropping attack. One of the major differences with other resynchronization methods is the ability for the decoder to recognize when the resynchronization is beneficial, thanks to the sensitivity vs. specificity trade-off of the landmark detection. In the reported experiments, the chosen settings lead to improving the robustness against cropping, without reducing the performance against other attacks. The proposed resynchronization approach thus defeats cropping attacks, possibly combined with rigid transforms and uniform scaling, or even shot noise that does not impact too many landmarks. However, when the cropping attack is combined with a valumetric attack e.g. noise addition, the performances of the decoder rapidly collapse.

In future work, it may be worth investigating alternate signature definitions to cope with such cases and thereby address real-life scenarios, e.g. print-and-scan attacks of 3D objects. Further research regarding the landmark definition may also prove fruitful. The approach taken in this chapter amounts to a uniform quantization of the signature $\mathbf{f}(v)$. When $\|\mathbf{f}(v)\|$ is close to null, a non-uniform quantization may be preferable, so as to avoid ill-defined numerical computations. A number of variants of the proposed synchronization patterns could also be investigated: the spherical configuration may be replaced with another primitive, such as an ellipsoid. Transmitting additional geometric parameters could then be used against anisotropic scaling attacks. Finally, the security of the resynchronization component should be thoroughly investigated.

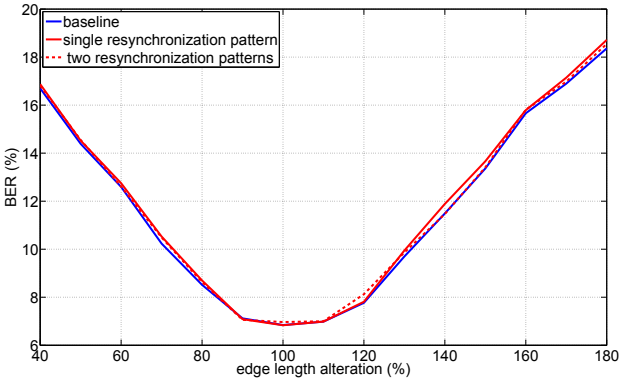


(a) Noise

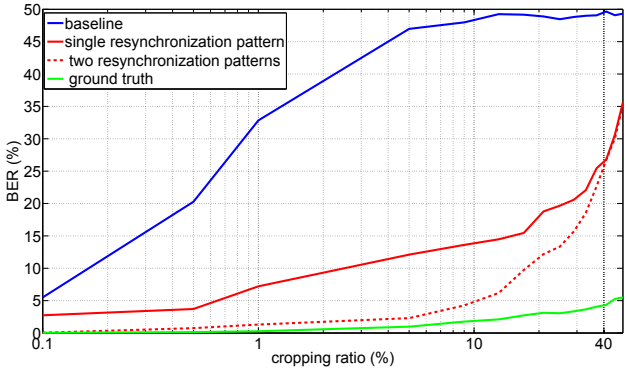
(b) Quantization



(c) Smoothing



(d) Triangle Soup



(e) Cropping

Figure 9-6: Benchmark of the robustness of the QP framework with and without the synchronization mechanism. With the parameter settings presented in Section 9.3.2, the synchronization marginally interferes with the robustness, except against the cropping attack, in which case the robustness improves 9-6(e): the solid red curve (single resynchronization pattern, i.e. only $\mathbf{g}$ is resynchronized) indicates an average 30% boost of the BER. The dotted red curve (two resynchronization patterns, i.e. all the critical information are transmitted) is at first even closer to the green curve, that depicts the performance of a semi-blind decoding, which provides an upper-bound on the performance of any resynchronization approach, as $\mathbf{g}$, $\max(\boldsymbol{\rho})$ and $\min(\boldsymbol{\rho})$ are the ground-truth ones.

Chapter 10

Conclusion

The research presented in this dissertation focused on blind and robust 3D watermarking to protect copyrighted triangle surface meshes in traitor-tracing use-cases. The review of the state-of-the-art and the benchmark of the content adaptation transforms first highlighted the current limitations and remaining challenges in this field of research.

Representing 3D assets through surface meshes embedded in $\mathbb{R}^3$ hinders the use of the popular spectral and multiresolution embedding domains for watermarking. As the basis of these domains are content-dependent, the watermarking systems are then sensitive to connectivity alterations, and the synchronization in blind approaches is often lost. Existing mitigating solutions, such as manifold harmonics or remeshing prior to wavelet decomposition, lead to further complex causality and synchronization issues.

In the spatial domain, intrinsic geometric quantities or locally-defined geometric quantities can in theory resist cropping or pose attacks. In practice, these quantities come with their own set of synchronization shortcomings. They are also more sensitive to valumetric attacks. In contrast, watermark carriers based on globally-defined geometric quantities exhibit larger robustness against valumetric attacks. Thanks to integral quantities, they also exhibit a moderate level of robustness against some connectivity-altering attacks, but their resilience against cropping attacks is usually not sufficient for practical use-cases.

10.1 Contributions

The previous chapters have detailed several approaches to overcome the limitations above.

Chapter 5 introduced a robust thickness estimation procedure, amenable to creating an embedding domain where the synchronization issues resulting from a pose are mitigated. Starting from an existing diameter-based thickness approximation, our estimation provides greater accuracy and robustness over the state-of-the-art Shape Diameter Function (SDF). Thickness estimates are promising as novel watermark carrier to resist pose. They indeed form an unexplored approach to robust 3D watermarking in the context of animation, which exhibits higher performances than the aforementioned locally-defined geometric quantities or intrinsic geometric quantities. Nonetheless, the complexity of the fusion function is currently prohibitive, as the correlation between thickness estimates at different locations is difficult to predict.

Chapters 6 and 7 focused on extending state-of-the-art Quadratic Programming (QP)-based optimization 3D watermarking system relying upon the radial distances as watermark carriers. The main purpose of the initial approach was to explicitly address the causality issue. The research presented in this dissertation then further extended this approach to create a versatile

and modular watermarking framework, in which the robustness and the fidelity can be controlled with greater precision. Multiple instantiations of systems within this framework were thoroughly benchmarked, and significant experimental and theoretical improvements were showcased by using integral quantities, adding some degrees of freedom to the optimization, and better fidelity metrics in the minimization.

More specifically, the robustness against some connectivity-altering distortions, such as simplification, was greatly increased by the volume-weighted definition of the center of mass. Leveraging the Mesh Structural Distortion Measure (MSDM) instead of the Root Mean Square (RMS) to calibrate the fidelity of the different systems in the experimental benchmark, the benefits of the other extensions in terms of fidelity were highlighted. For mechanical parts, the reported improvements are especially significant: unlike most radial-distance base approaches, preserving smooth parts of the mesh can be seamlessly enforced.

Another benefit of the presented extensions is the possibility to modify the embedding function, and switching to a Spread Transform (ST) formulation was studied. Aside from the advantages in terms of fidelity vs. robustness trade-off, it also offers the possibility to increase the security of a watermarking system. Chapter 8 introduced a systematic security assessment of the proposed radial distance-based watermarking framework, starting from the only existing secret offset-based mechanism in the 3D watermarking literature for radial distances. A series of experimental attacks and counter-strategies were presented, leveraging the flexibility of the framework to further add new security mechanisms into 3D watermarking.

Finally, Chapter 9 tackled the resynchronization issue against cropping attacks in the QP framework by adding a new resynchronization component. Secret landmark points are inserted into the mesh with a specific configuration such as, e.g., a sphere, prior to using the QP embedder. Landmarks are blindly retrieved in the decoder, and their configuration conveys the synchronization information that the QP decoder needs to handle the cropping attack. Experimental results showcased the clear-cut benefits of this effective resynchronization mechanism, whose integration into the framework is straightforward. This research hence overcomes one of the main limitations of 3D watermarking without relying on the routine partitioning and repeated embedding strategy.

10.2 Follow-up Research

The study on the robust estimation of the thickness for 3D objects has shed light on the remaining difficulties for a thickness-based embedding domain. Finding an effective fusion function indeed becomes the main challenge, as the robustness against pose becomes less of an issue. One solution may be to consider other thickness estimation methods than the ones related to the Shape Diameter Function. Research in this direction would involve an extensive comparison of the pros and cons of the plurality of robust Medial Axis Transform (MAT) approaches.

The powerful QP-based framework to watermark radial-distances can be further complemented with other extensions. For instance, even more evolved fidelity metrics could be integrated. The final goal of this research direction is to enable a direct minimization of a complex perceptually-correlated distortion assessment metric, which are non-quadratic. Even more integral definitions for the watermark carriers could also be investigated using, e.g., volume-weighted histograms. Carriers could even be not limited to the vertex positions, but extended to facet centers or arbitrary locations on the surface to reduce the sampling-dependency of watermarking approaches.

Regarding the security, combining all the counter-measures presented in Chapter 8 is sufficient to prevent an adversary from using the tested attacks. Nevertheless, it may not be sufficient for real-life traitor-tracing applications. Besides, while the results focused on only attacking the QP

framework, most of the existing radial-distance based 3D watermarking systems suffer from similar security flaws. Adapting some of the attacks to these systems is a stimulating research direction for improved 3D watermark security.

The resynchronization component introduced in Chapter 9 shows great potential against cropping attacks. For instance, using more complex configuration such as ellipsoids would allow for larger amounts of synchronization information to be efficiently conveyed to the decoder. Most radial-distance based systems could also leverage this new component. Landmarks may finally not be restricted to the signature derived from coefficients of a paraboloid, and they could be defined through more robust feature descriptors.

10.3 Long-term Perspectives

Three main threads of research need to be investigated so that 3D watermarking is made really practical for real-world applications.

First, as progressive compression techniques are now part of many 3D distribution systems, watermarking is expected to handle decoding at different levels of details. Second, one of the most common purposes for using meshes in the entertainment industry is to create animations, whereas 3D watermarking mostly only allows for static meshes. Existing approaches to watermarking 3D animations have considered the content as a 3D signal varying through time: the motions of the vertices are modified instead of the mesh itself. These approaches are not robust against different animations of the same mesh, and 3D watermarking based on pose-invariant quantities such as thickness or geodesic distances is thus direly needed.

Third, because of the incoming print & scan attack on 3D meshes, watermarking systems will have to deal with severe conversion attacks, e.g. from surface-based to volumetric representations, before being integrated into real-world 3D creation and distribution environments. One of the main obstacle to this thread of research lies into the benchmark. In contrast to videos, where small size benchmarking campaigns using camcorders can be performed, undertaking a print & scan benchmark is currently too labor-intensive and expensive.

Appendix A

Introduction

Contexte

Les modèles tridimensionnels (3D) sont omniprésents dans de nombreuses applications industrielles. Dans la production cinématographique, ils ont commencé à remplacer les dessins informatiques bidimensionnels (2D) traditionnels depuis le début des années quatre-vingt et la sortie de *Tron* (1982) par les studios Walt Disney. Grâce à des logiciels toujours plus puissants [Aut14] et à de nouveaux systèmes de capture de mouvements, les animations à partir de modèles 3D sont à présent couramment utilisées non seulement dans des films d'animation, mais aussi dans des contenus principalement constitués de prises de vue réelles. La qualité et la précision des détails des modèles 3D les rendent de plus en plus indistinguables des objets réels. Nous n'avons par exemple pas nécessairement conscience que les combattants à l'arrière plan des vastes batailles du *Seigneur des Anneaux* sont des modèles 3D, animés par une intelligence artificielle complexe [Reg14].

L'arrivée massive de processeurs graphiques 3D sur le marché grand public autour de 2000 a incité l'industrie du jeu vidéo à remplacer les moteurs 2D et pseudo-3D (qui simulent la 3D à partir d'une projection 2D, aussi appelée 2.5D) par des moteurs entièrement 3D. L'utilisation des modèles 3D s'est alors fortement accrue. Bien qu'ils soient souvent créés par les studios de jeu vidéo professionnels pour un usage interne, de nouveaux types de distribution sont apparus lorsque des entreprises ont commencé à commercialiser directement leurs productions 3D [FC14]. En sciences numériques et en ingénierie, la conception assistée par ordinateur utilise activement la modélisation 3D dans les simulations numériques, car elle permet de réduire les coûts de recherche et de développement.

Les modèles 3D n'ont pas seulement un rôle grandissant dans le cadre des applications professionnelles, mais également dans celui des contenus générés par les particuliers. Par exemple, certains moteurs de jeu récents offrent la possibilité d'intégrer des contenus personnalisés. La création de modèles 3D représentant des personnages ou d'autres éléments d'un jeu vidéo s'est popularisée, soutenue par des outils dédiés [Ble14, Aut14, Epi14] et des revues spécialisées [Pub13] à l'intention des semi-professionnels.

Dans un avenir proche, l'importance des modèles 3D va être amplifiée par l'expansion des activités d'impression 3D. La mise en vente d'imprimantes 3D pour le grand public va modifier les modes de consommation. Contrairement aux autres types de contenus multimédias, les modèles 3D passeront alors du statut de produits artistiques et culturels numériques à celui de biens de consommation concrets et tangibles. Les prédictions des analystes prévoient de nouveaux usages fondés sur l'impression de modèles téléchargés depuis des bases de données en ligne [WGL+13]. En raison de cette diversité d'applications et de cette accessibilité croissante, la protection de

la propriété intellectuelle et le contrôle de la diffusion des modèles 3D constituent des enjeux importants.

Les créations numériques font toujours face à des violations de brevets ou de marques, mais l'industrie du divertissement porte une attention toute particulière à la protection du droit d'auteur [Org08]. Les films et la musique sont piratés de façon notoire et se trouvent au centre d'un vaste marché noir numérique [Ahr06]. Les ayant-droits ont donc lourdement investi pour retrouver les sources de redistribution illégale de leurs contenus. Alors que la complexité et la valeur des contenus 3D augmentent, des scénarios semblables sont anticipés, et les objets 3D protégés par le droit d'auteur seront eux aussi illégalement diffusés. Cependant, vu l'amincissement de la frontière entre les modèles numériques et les biens de consommation dans le monde réel, l'impact de ce problème s'élargit, entre autres, à la vente de produits dérivés.

Si le modèle 3D du personnage principal de la toute dernière grosse production cinématographique peut être téléchargé illégalement, n'importe qui pourra manufacturer chez soi un produit avec ce personnage pour insigne. Ceci affecte les ventes de produits dérivés, et la réputation de l'auteur, lorsque la copie pirate est une version dégradée de l'original. Les jouets pour enfant par exemple respectent une multitude de normes, ce qui ne sera pas forcément le cas d'un produit manufacturé par un amateur : des problèmes de sécurité peuvent apparaître. Des affaires de ce type commencent à émerger. En 2012, Games Workshop Limited a envoyé une lettre de cessation pour un modèle numérique et une impression 3D inspirés d'une de leur miniature de tank [New12]. En 2013, HBO en a envoyé pour un support pour iPod reproduisant la forme du trône de fer de la série *Game of Thrones* [Mac14]. Dans ces deux cas, les entreprises ont déclaré qu'il s'agissait d'une violation de leur droit d'auteur. Elles ont dû ainsi accomplir deux tâches complexes : vérifier si un modèle est une reproduction illégale, et retrouver sa source de diffusion.

Tatouage numérique

Le tatouage numérique est un domaine technique qui fournit aux ayant-droits des moyens de défendre leur propriété intellectuelle. C'est un élément central des systèmes de protection des contenus multimédias, et qui complète la cryptographie [CMB+07]. Cette dernière vise à prévenir un accès illégitime au contenu, tandis que le tatouage répond aux problèmes qui surviennent lorsque les utilisateurs autorisés accèdent au contenu, par exemple après le déchiffrement, ou lorsque le chiffrement est cassé.

En général, le tatouage modifie un contenu multimédia pour insérer un message secret d'une manière robuste et imperceptible. Ce message de tatouage peut servir de preuve pour retrouver les auteurs de fuite, ou "traçage de traître". Il peut aussi être une marque de propriété en cas de contestation. Dans le premier cas, les utilisateurs autorisés ont seulement accès à une copie personnalisée d'un modèle 3D coûteux. Chaque usager possède alors une version unique aux variations imperceptibles ; les usagers et les copies sont associés un à un. Si une diffusion illégale survient, les ayant-droits peuvent en retrouver la source, puisque son identité est insérée dans le contenu piraté publiquement accessible. Inversement, lorsque le message de tatouage correspond à l'identité de l'ayant-droit, celui-ci peut démontrer qu'un contenu lui appartient dans le cas de litiges autour de la propriété.

Le tatouage numérique a d'autres applications pour la sécurité, comme la détection de falsification, ou plus généralement pour le suivi de diffusion. Conçus en vue d'applications très diverses, les systèmes de tatouage sont adaptés pour répondre à des exigences spécifiques.

Problématique

Du point de vue scientifique, un système de tatouage est une forme de système de communication, dans lequel un émetteur envoie un signal au récepteur à travers un canal de communication. En tatouage, l'inserteur introduit un signal, qui encode le plus souvent un message, dans un contenu protégé par le droit d'auteur, qui est apporté au décodeur. Le propriétaire des droits contrôle uniquement ces deux composants, mais pas les opérations appliquées au contenu lors de la transmission : un utilisateur qui y accède peut modifier ce contenu de façon arbitraire. Pour que le décodeur puisse récupérer le message, ces modifications doivent être gérées par le tatouage, qui est alors caractérisé par sa robustesse.

Augmenter la taille du message conduit souvent à une baisse de la robustesse, et un équilibre entre ces deux quantités doit donc être recherché. De plus, la plupart des utilisateurs n'acceptent pas un tatouage dégradant les contenus. La fidélité du tatouage, qui mesure quantitativement l'altération du contenu tatoué, ajoute une nouvelle contrainte au système. En général, le tatouage répond donc à un compromis complexe entre la robustesse, la fidélité et la quantité d'informations transmises.

Cette dissertation traite du tatouage pour les modèles 3D, abrégé par "tatouage 3D", dans le contexte du traçage de traître. Le message, représentant l'identité de l'utilisateur, doit être inséré dans le modèle d'une façon particulièrement robuste. Une fois que l'existence du tatouage est connue, les auteurs de fuite vont essayer de retirer du contenu leur identifiant incriminant, afin d'éviter d'être poursuivis. La quantité d'information utilisée par le système atteint quelques dizaines de bits, et le compromis précédemment évoqué est donc orienté vers la robustesse. Ces tatouages sont donc plus simplement appelés "robustes".

À l'opposé, les tatouages 3D "fragiles" ou de "haute capacité" se concentrent sur la quantité d'information insérée pour les seconds ou, pour les premiers, sur des applications qui requièrent des contraintes de robustesse moins fortes, comme la détection de falsification. Si un grand nombre de systèmes fragiles ou de haute capacité ont été développés plutôt que des systèmes robustes, c'est que fournir un haut niveau de robustesse dans le contexte de la 3D présente plusieurs problèmes scientifiques et techniques.

Enjeux scientifiques du tatouage 3D robuste

La représentation numérique des modèles 3D est elle-même à l'origine de problèmes complexes. La robustesse du tatouage s'appuie entre autres sur un accord entre l'inserteur et le décodeur quant à la représentation du contenu. Lorsque le décodeur n'a pas connaissance du modèle initial non tatoué (tatouage aveugle), cet accord est difficilement réalisable. D'un autre côté, fournir un original au décodeur présente des inconvénients pratiques. Ces problèmes de représentation amoindrissent également l'utilité des outils de traitement du signal les plus courants dans le cadre du tatouage robuste, comme la transformée de Fourier ou la transformée en ondelettes. Leurs extensions pour les modèles 3D sont en effet elles-mêmes dépendantes du contenu. Gérer des modifications entre l'inserteur et le décodeur est alors complexe.

Un modèle 3D tatoué peut subir des altérations de types très variés. Deux d'entre eux représentent un important enjeu technique et pratique : le rognage et la déformation isométrique de la surface, plus connue sous le nom de "pose". Ces deux cas créent des problèmes de synchronisation du tatouage. Pour le rognage, une partie du modèle est supprimée, ce qui peut réduire la valeur de l'objet aux yeux des ayant-droits. Même des suppressions faiblement perceptibles peuvent suffire à désynchroniser un système de tatouage, et la résistance au rognage constitue un obstacle impor-

tant en règle générale. La pose, elle, concerne seulement les animations 3D. Les seuls systèmes de tatouage, pour lesquels la robustesse à la pose est primordiale, sont ceux qui traitent des contenus destinés à être animés.

La possibilité pour un utilisateur non autorisé de modifier de façcon arbitraire, à son gré, le message du tatouage, peut avoir des conséquences judiciaires graves. L'identifiant pourrait alors être modifié afin d'accuser un innocent. A l'exception de l'ayant-droit, personne ne doit avoir accès aux identifiants insérés dans les contenus. C'est à cette contrainte spécifique que correspond le problème de la sécurité du tatouage. Les recherches en tatouage 3D ont souvent négligé cet aspect du système ou utilisé des approches de validation peu fiables. Pour d'autres types de contenus multimédias, au contraire, des études théoriques minutieuses ont été menées.

Les propriétaires de modèles 3D, soucieux de préserver la qualité visuelle de leurs objets, exigent des systèmes de tatouage robuste et garantissant aussi un niveau certain de fidélité. Pour mesurer la distorsion résultant de l'insertion du message, telle qu'elle est perçue par un utilisateur, il n'existe pas de méthode sûre. Des études sont en cours pour définir des métriques dont les résultats soient corrélés avec les perceptions humaines ; seules existent quelques solutions partielles. Leur adoption par la communauté du tatouage est faible, ce qui ralentit les recherches, car les différents systèmes proposés ne sont pas calibrés suivant les mêmes métriques de distorsion.

Enfin, la plupart des opérations (algorithmes, procédures informatiques) requièrent que les objets 3D auxquels elles s'appliquent vérifient plusieurs propriétés contraignantes. Dans la pratique, celles-ci ne sont pas souvent respectées par les modèles 3D. Il faut les réparer avant de les exploiter, par exemple en retirant certains défauts. La plupart des collections d'objets qui n'ont pas été rassemblées dans le cadre de travaux de recherches ne sont donc pas directement utilisables pour des campagnes de tests. À l'inverse de ce qui se passe pour le tatouage audio ou vidéo, seules sont menées des campagnes de faible envergure.

Annonce de plan

Le chapitre 2, introduction technique au tatouage 3D, présente les pré-requis en traitement des objets 3D et en tatouage. Les chapitres suivants constituent deux parties.

Dans la première partie, on s'intéresse à la couche d'adaptation du contenu pour le tatouage 3D. L'état de l'art des systèmes de tatouage 3D robuste, classés suivant leur stratégie d'adaptation, est passé en revue dans le chapitre 3. Une étude expérimentale des couches d'adaptation les plus utilisées est présentée au chapitre 4. Enfin, le chapitre 5 propose une nouvelle fonction d'adaptation, fondée sur l'épaisseur d'un objet 3D, qui possède des propriétés intéressantes face à la pose. Les performances de cette fonction y sont minutieusement testées.

La seconde partie traite de l'extension et de la généralisation d'une formulation du tatouage 3D comme un problème d'optimisation sous contraintes, pour créer un cadre souple et modulaire pour le tatouage robuste. Le chapitre 6 développe de multiples extensions pour améliorer la robustesse et la fidélité de la formulation originale. Leurs performances pratiques sont mesurées et comparées au chapitre 7. Le chapitre 8 propose une étude approfondie de la sécurité de ce cadre de tatouage, par le biais d'une série d'attaques et de contre-mesures. Enfin, le problème des attaques par rognage est analysé au chapitre 9, et une nouvelle méthode de resynchronisation est ajoutée au système de tatouage.

Le chapitre 10 conclut la dissertation en récapitulant les principaux résultats obtenus. Il suggère des améliorations possibles à court terme et résume les pistes ouvertes pour les futures recherches en tatouage 3D.

Appendix B

Résumé

Pré-requis pour le tatouage tri-dimensionnel

Représentation d'un objet tri-dimensionnel

Il existe de nombreuses formes de représentation pour les objets tridimensionnels (3D). L'une des plus courantes est la représentation surfacique, dans laquelle un objet est défini par sa surface bidimensionnelle (2D). Mathématiquement, la surface est une variété possiblement à bords orientée, de dimension deux, plongée dans un espace de dimension trois. Cette définition est particulièrement adaptée à la description d'objets solides. Cette représentation mathématique trouve son approximation numérique dans les maillages surfaciques polygonaux. Le maillage surfacique polygonal est constitué d'une composante géométrique, correspondant à une série de points $\mathbf{p}_i$ dans $\mathbb{R}^3$, et d'une connectivité, correspondant à un graphe dont les sommets v_i sont associés aux points $\mathbf{p}_i$ et reliés par des arêtes formant des facettes polygonales. Un cas particulier, le maillage triangulaire, dont toutes les facettes sont des triangles, est le plus couramment employé.

Un maillage forme une représentation linéaire par morceaux d'une surface. Elle est donc continue mais généralement non lisse, ce qui constitue un frein pour l'estimation de certaines quantités. Afin d'apporter des informations qui améliorent le rendu de l'objet 3D, il est possible de compléter le maillage par de nombreuses informations comme la texture ou les directions normales aux sommets.

Traitement de la géométrie pour maillages

Le traitement des maillages est un domaine de recherche actif. Ces travaux sont mis à contribution lors de la conception des systèmes de tatouage 3D. Ainsi d'intenses recherches ont-elles été menées sur l'estimation des courbures principales, qui pose problème, avant que celle-ci puisse être utilisée en tatouage 3D.

L'extension de l'analyse spectrale constitue un autre exemple de l'utilisation pour le tatouage 3D des recherches en traitement informatique de la géométrie. La représentation fréquentielle d'un signal constitue l'un des principaux outils pour le traitement du son ou de l'image. Cette analyse se fonde sur une discrétisation de l'opérateur Laplacien pour les variétés 2D. Dans le cas des maillages, qui correspondent à un échantillonnage irrégulier du signal géométrique, cette discrétisation est particulièrement complexe. Le Laplacien combinatoire, seulement lié à la connectivité du maillage, est la solution la plus couramment employée en tatouage. Une seconde solution plus complexe, mais davantage liée à la géométrie, intitulée "manifold harmonics" a également reçu une attention croissante. Dans tous les cas, ces discrétisations aboutissent à la définition d'un domaine spectral dont la base dépend du contenu, ce qui n'est pas le cas pour les images ou le son. Cette propriété

ainsi que la complexité de la transformée spectrale 3D sont particulièrement problématiques pour le tatouage.

L'analyse multi-résolution, s'appuyant par exemple sur la transformée en ondelettes, est un autre outil majeur pour le traitement du signal. Plusieurs travaux ont proposé des extensions qui permettent de représenter un maillage semi-régulier sous la forme d'une série de maillages, dont chacun atteint un niveau de détails graduellement décroissant. La contrainte de semi-régularité sur le maillage est particulièrement délicate pour le tatouage, et plusieurs solutions pour contourner cette difficulté ont été étudiées, mais chacune présente des inconvénients.

Propriétés d'un système de tatouage

Le tatouage numérique est défini comme "une méthode d'altération imperceptible d'un contenu, maillage entre autres, pour insérer un message à propos de ce contenu." À la manière d'un système de communication, un système de tatouage est constitué de trois composants : (i) un émetteur, appelé aussi "inserteur", auquel sont fournis le *contenu* et le *message* à insérer, (ii) un récepteur, appelé "décodeur", qui extrait d'un contenu le message inséré, et (iii) un canal de communication entre les deux premiers composants. Le tatouage 3D est un sous-domaine du tatouage numérique dans lequel l'inserteur et le décodeur traitent des maillages surfaciques triangulaires, plus simplement appelés dans ce qui suit "maillages".

Quatre propriétés caractérisent un système de tatouage : la capacité, la robustesse, la sécurité, la fidélité.
La *capacité* d'un système de tatouage est mesurée par la taille du message binaire $\mathbf{m} \in \{-1, 1\}^{n_b}$ inséré dans le maillage. Dans le cadre du tatouage robuste, n_b est habituellement compris entre 16 et 64 bits.
Lors du passage dans le canal de communication, le contenu et le message peuvent être altérés par des attaques. La *robustesse* du système est mesurée par le taux d'erreur binaire, c'est-à-dire le ratio entre le nombre de bits du message correctement décodés et la taille du message initialement émis.

Dans le cas du décodage dit "aveugle", seul est fourni au décodeur le maillage tatoué, éventuellement altéré, duquel le message doit être extrait. Mais dans le cas du tatouage dit "non aveugle", le décodeur dispose en plus du maillage original dans lequel le message avait été inséré. Ce dernier cas permet l'utilisation de techniques de recalage qui améliorent la robustesse du système de tatouage.
La *sécurité* du système de tatouage consiste à prévenir tout accès non-autorisé au canal de tatouage, tel que la lecture ou la modification du message par un utilisateur malveillant. Elle est assurée au moyen d'une clé secrète η.
Enfin, la *fidélité* du système de tatouage mesure la distorsion, perçue par un utilisateur, résultant de l'insertion du message dans le contenu.

La capacité, la robustesse et la fidélité d'un système sont des propriétés antagonistes qui donnent lieu à différents compromis selon les applications du système de tatouage. Les travaux de recherche présentés ici s'intéressent plus particulièrement au *tatouage robuste* qui nécessite un niveau important de robustesse et de sécurité.

Composants élémentaires d'un système de tatouage

Les composants élémentaires de l'inserteur et du décodeur ont fait chacun l'objet d'une analyse puisque chacun a un rôle propre (voir Figure **??**). Le premier de ces composants, appelé la *fonction d'extraction*, est appliquée en entrée de l'inserteur et du décodeur. Il s'agit d'une couche d'adaptation qui transforme le maillage initial en un signal dans le *domaine d'insertion*. Ce signal, appelé *signal porteur du tatouage*, est en général un vecteur, noté $\mathbf{c}$ au niveau de l'inserteur. La plupart

des recherches en tatouage 3D se focalisent sur l'élaboration d'une fonction d'extraction qui soit robuste aux attaques survenant dans le canal de transmission.

La *fonction d'insertion*, au niveau de l'inserteur, associe au message de tatouage et au signal porteur, un élément $\mathbf{c}^w$ dans le domaine d'insertion. Deux principales familles de fonctions d'insertion existent : l'*étalement de spectre*,et la *communication informée à l'émission*. Dans le premier cas, une séquence pseudo-aléatoire, pondérée par le message de tatouage antipodal, module le signal porteur à l'émission ; le décodeur récupère le message en corrélant le porteur reçu avec la séquence pseudo-aléatoire. Dans le second cas, pour limiter les problèmes d'interférences entre le porteur et le message, le signal $\mathbf{c}$ est modifié selon un partitionnement pré-établi du domaine d'insertion. Chacune des partitions est associée à un message et est habituellement définie par un quantificateur.
La *fonction de fusion* réalise le plongement inverse du porteur tatoué $\mathbf{c}^w$ pour obtenir le maillage en sortie de l'inserteur.
Enfin, le module de *resynchronisation* réaligne le maillage attaqué dans le décodeur.

Dans un système de tatouage 3D, les attaques dans le canal de transmission prennent des formes variées. Il peut s'agir de simples similarités, d'altérations de la géométrie par ajout de bruit ou par lissage, de modifications de la connectivité, de ré-échantillonnage du signal géométrique par un remaillage, d'altérations de la topologie, voire de changement de représentations ou de déformations de la surface, par exemple un changement de pose.

Fidélité du tatouage 3D

La fidélité d'un système de tatouage est mesurée par une métrique de distorsion, appliquée entre l'entrée et la sortie de l'inserteur, et dont les variations coïncident avec la distorsion perçue par un utilisateur. Contrairement à d'autres signaux, la définition d'une métrique de distorsion pour le tatouage 3D reste un problème non résolu.

La distance de Hausdorff et l'erreur quadratique moyenne, bien que d'un usage encore courant, ne sont que très faiblement corrélées avec la perception de la distorsion par un utilisateur. Des métriques liées au Laplacien ou au déplacement des sommets selon les directions normales présentent des résultats sensiblement plus appropriés pour le tatouage. Les métriques spécifiquement conçues pour comparer des systèmes de tatouage 3D, telles que la MSDM, mesurent en général une information de rugosité pour tenir compte d'effets de masquage. Bien que complexes à calculer, elles sont néanmoins parfois aussi utilisables pour optimiser un système de tatouage en améliorant la répartition de la distorsion d'insertion sur le signal porteur.

État de l'art du tatouage 3D

Dans le cadre du tatouage 3D robuste, peu de recherches se sont appuyées sur des fonctions d'extraction et de fusion préservant la géométrie et altérant uniquement la connectivité d'un maillage. Ces approches sont en effet souvent limitées par une fidélité faible et par un manque de robustesse face au remaillage. La plupart des travaux se sont donc orientés vers des fonctions d'extraction modifiant la géométrie. Ils sont regroupés en trois principales approches.

Tatouage dans le domaine spatial

Dans le domaine spatial, la fonction d'extraction utilise directement la position des sommets du maillage pour définir le signal porteur du tatouage. Une première famille de systèmes repose sur des quantités géométrique locales, par exemple lorsque chaque valeur du porteur est entièrement

définie par la forme de la surface au voisinage d'un sommet. Néanmoins, cette stratégie présente des problèmes de synchronisation et de robustesse.

Une seconde famille de systèmes extrait des informations sur la distribution des distances euclidiennes entre le centre de gravité du maillage et les sommets, communément appelées "distances radiales". La plupart de ces systèmes reposent sur une altération de la moyenne de chaque classe de l'histogramme des distances radiales. Ce type d'approche présente une importante robustesse, en particulier contre les attaques valumétriques. De nombreuses recherches ont alors été menées pour utiliser des quantités géométriques encore plus robustes, pour assurer une robustesse au rognage, ou pour améliorer la fidélité du système de tatouage.

Une dernière famille s'est intéressée à l'extraction de distributions liées à d'autres quantités géométriques, par exemple les distances géodésiques. Ces distributions sont souvent moins robustes que dans le cas précédent, mais possèdent des propriétés intéressantes face à certaines attaques, telles que le rognages.

Tatouage dans le domaine transformé

La majorité des fonctions d'extraction dans le domaine transformé utilise une extension de la transformée de Fourier pour les maillages surfaciques, qui nécessite une discrétisation de l'opérateur Laplacien. L'extension la plus fréquemment employée, fondée sur une discrétisation combinatoire, aboutit à un signal de tatouage correspondant aux coefficients spectraux associés aux basses fréquences de la surface. Celui-ci est robuste aux attaques valumétriques, mais sensible aux altérations de la connectivité. Une seconde extension, intégrant des informations géométriques sur la surface, a été employée dans un nombre restreint de systèmes. Elle offre une plus grande robustesse aux attaques modifiant la connectivité, mais la modification du maillage (fonction de fusion) devient complexe et souffre d'un problème qualifié de "causalité".

Tatouage multi-résolution

Dans ce type d'approche, la fonction d'extraction du système calcule les coefficients d'ondelettes associés à une version du maillage avec peu de détails. Les approches multi-résolutions présentent deux problèmes : elles imposent une contrainte sur la connectivité initiale du maillage, et elles offrent une faible robustesse aux altérations de la connectivité dans le canal de communication. Des techniques pour contourner le premier problème ont pu être développées, mais, dans le cadre du tatouage robuste aveugle, la sensibilité du système aux attaques sur la connectivité devient encore plus forte.

Conclusion

La robustesse des approches de type transformée ou multi-résolution rencontre de nombreux problèmes face aux attaques modifiant la connectivité. Ces approches reposent sur des composants qui restent des sujets d'études ouverts pour le traitement des maillages. Les approches spatiales, fondées sur des distributions de quantités géométriques, présentent une robustesse moindre face aux attaques valumétriques, mais certaines sont capables de résister à d'importants changements de connectivité. D'un point de vue théorique, elles sont cependant limitées face à la pose ou le rognage.

Évaluation des systèmes de tatouage

Les performances d'un système de tatouage dépendent d'un compromis entre la robustesse, la fidélité et la capacité. Elles sont généralement mesurées par la robustesse pour une fidélité et une capacité données. Cette mesure correspond à un unique point de fonctionnement, et ne rend pas compte de l'ensemble des performances possibles. De plus, cette mesure nécessite la sélection d'une métrique de distorsion. Dans le cadre du tatouage 3D, cette tâche est complexe, et introduit souvent un biais en faveur d'un système. Afin d'éviter ces problèmes, les performances ont été mesurées sur les seules fonctions d'extraction plutôt qu'à l'échelle d'un système complet.

Dans le protocole expérimental proposé, plusieurs types d'attaques sont appliqués aux treize maillages d'une base de données. La robustesse face à l'ajout de bruit, au lissage, au rognage ou à la pose est ainsi testée. Les performances d'une fonction d'extraction correspondent à la stabilité du signal porteur du tatouage après de telles attaques. Cette stabilité est d'abord mesurée localement sur chaque sommet, puis agrégée à l'échelle d'un maillage, et enfin intégrée à une statistique globale sur l'ensemble de la base.

Les premières simulations montrent l'instabilité de l'aire de la surface au voisinage d'un sommet. En particulier, l'aire dépend d'un paramètre global sur le maillage, et n'est donc pas robuste à la pose ou au rognage. À l'inverse, les distance radiales, entre le centre de gravité et les sommets, présentent une très large stabilité, excepté face à la pose et au rognage. Les distances géodésiques, calculées entre tous les sommets du maillage et un sommet de référence, sont modérément stables face à toutes les attaques testées. Les courbures principales présentent une très forte instabilité face aux attaques valumétriques, mais résistent au rognage. A l'inverse, la norme des coefficients spectraux est robuste aux attaques valumétriques, mais très instable face au rognage ou à la simplification. Par ailleurs, l'estimation pratique de ces coefficients est difficile.

Les performances mesurées pour des niveaux d'attaque croissants confirment ces observations. En conclusion, les distances radiales offrent dans la plupart des cas de meilleurs résultats que les autres porteurs, mais elles sont inexploitables dans le cas du rognage et de la pose. Seules les distances géodésiques ont la capacité à résister face aux attaques. Elles s'accompagnent néanmoins de plusieurs inconvénients, qui n'apparaissent pas au travers de l'analyse des fonctions d'extraction.

Domaine d'insertion robuste face à la pose

Lorsqu'un objet est animé suivant différentes poses créées à partir d'un squelette, l'épaisseur locale est une quantité stable, et seules ses valeurs au niveau des articulations sont parfois altérées. L'épaisseur locale constitue donc un candidat prometteur pour définir un nouveau domaine d'insertion du tatouage. Un signal porteur doit également être robuste aux autres attaques évoquées précédemment (bruit, lissage). Pour l'épaisseur, la robustesse aux imperfections liées à la numérisation d'un objet, par exemple des micro-trous sur la surface, présente un enjeu supplémentaire car l'intérieur et l'extérieur du modèle ne sont plus clairement définis.

L'épaisseur locale est définie mathématiquement par la transformée de l'axe médian, qui est particulièrement sensible à des défauts sur la surface. Elle n'est donc par appropriée dans le cadre du tatouage robuste. Une approche plus intuitive consiste à définir l'épaisseur suivant le diamètre (SDF). Cette solution a été utilisée avec succès pour la segmentation de maillage. L'estimateur d'épaisseur présente cependant des limites en terme de robustesse, de stabilité et de précision. Pour cette raison, un nouvel estimateur robuste de l'épaisseur locale est étudié.

La méthode proposée débute par un ré-échantillonnement de la surface, avant de construire un nuage de points associés au demi-diamètre estimé pour chaque échantillon. L'épaisseur locale est

alors définie à partir d'une distance robuste au nuage de points.

Le demi-diamètre pour un point d'échantillon sur la surface est calculé en mesurant la longueur de rayons lancés vers l'intérieur du modèle 3D. Ces rayons sont générés aléatoirement dans un cône dont l'ouverture est itérativement adaptée pour identifier des parties saillantes. Le nuage de points est ensuite construit en projetant à l'intérieur de l'objet les échantillons suivant leur direction normale, à une distance égale à leur demi-diamètre. L'épaisseur en un point de requête est définie par la racine de la moyenne des distances carrées aux k points du nuage les plus proches, et visibles depuis le point de requête. Cette dernière condition permet de tenir compte d'éventuels obstacles entre la requête et le nuage. Le paramètre k contrôle l'échelle de l'estimation de l'épaisseur locale : l'influence des détails du maillage diminue lorsque k augmente.

Les performances de la méthode proposée, notée t, sont comparées aux performances de la méthode originale (SDF). Leurs paramètres sont d'abord ajustés afin d'assurer une comparaison équitable, c'est-à-dire à une échelle égale. Leur précision est ensuite mesurée sur des tores, des sphères et des ellipses. Pour ces formes géométriques simples, l'épaisseur définie par l'axe médian est calculée analytiquement et sert de référence pour la métrique de précision. Les résultats expérimentaux indiquent que t est plus précise que la SDF.

Les deux méthodes testées reposent sur des processus stochastiques ; des mesures répétées sur des données identiques ne sont donc pas égales. La stabilité correspond aux variations entre plusieurs estimations d'épaisseur locale sur un même objet. Dans l'ensemble, la SDF est plus instable que t. Enfin, la robustesse de l'estimation d'épaisseur face à des perturbations d'un maillage est mesurée à partir d'une série d'attaques, qui incluent l'addition de bruit, le lissage, la soupe de triangle, la simplification, le remaillage et la pose. Les résultats obtenus montrent que t présente un niveau de robustesse élevé et des atouts pour la segmentation robuste de maillage.

Trois obstacles majeurs doivent néanmoins être surmontés pour construire un domaine d'insertion du tatouage fondé sur l'épaisseur. Premièrement, l'utilisation du diamètre est pertinente pour des formes animales ou humanoïdes, mais ce n'est pas le cas pour certaines pièces mécaniques, par exemple un cube, pour lesquels la notions de squelette n'est pas intuitive. L'estimation d'épaisseur fournit alors des résultats difficilement exploitables. Deuxièmement, la robustesse de t aux attaques valumétriques est toujours inférieure à celle des distances radiales. Troisièmement, la définition de la fonction de fusion, qui fait correspondre le maillage avec le signal d'épaisseur tatoué, demeure un problème à résoudre.

Optimisation sous contrainte pour le tatouage 3D dans le domaine spatial

Dérivations des extensions de l'optimisation sous contrainte

Dans le domaine spatial, le tatouage de l'histogramme des distances radiales, entre le centre de masse et les sommets, notées ρ_i, a été formulé comme un problème d'optimisation sous contraintes. Tatouer signifie déplacer les sommets du maillage de telle sorte que la moyenne dans chaque classe de l'histogramme de ρ_i atteigne une valeur cible, tout en minimisant la distorsion due à ces déplacements. Mathématiquement, la première partie du problème correspond à une contrainte de tatouage, tandis que la minimisation correspond à une fonction de coût représentant la fidélité du système. Pour répondre aux contraintes de causalité, les déplacements doivent également assurer la stabilité du centre de masse, ainsi que l'association entre les sommets et les classes de l'histogramme de ρ_i. Au décodage, tous les distances radiales $\hat{\rho}_i$ sont d'abord estimées, puis, à partir de la moyenne dans chacune des classes de leur histogramme, le message inséré peut être récupéré de manière aveugle.

Dans l'état de l'art, les variables de l'optimisation, notées $\delta\rho_i$, représentent le déplacement unidimensionnel des sommets suivant l'axe qui les relie au centre de masse. La métrique de fidélité est l'erreur quadratique, c'est-à-dire la somme, sur tous les sommets, du carré de $\delta\rho_i$. La contrainte de causalité sur le centre de masse se traduit par une triple égalité linéaire suivant les variables, qui impose à la somme des déplacements d'être nulle dans toutes les directions. La causalité sur l'histogramme se réduit à borner chaque $\delta\rho_i$ dans un intervalle pré-calculé. La contrainte de tatouage s'exprime par une combinaison linéaire des $\delta\rho_i$. Plus précisément, la moyenne dans chacune des classes de l'histogramme des distances radiales est associée à un bit du message de tatouage, et modulée autour d'une constante normalisée pour indiquer un symbole -1 ou $+1$. Cela correspond à une série d'inégalités linéaires suivant les variables. L'insertion du tatouage peut ainsi être formulée comme un problème d'optimisation quadratique, abrégé par "tatouage QP", pour lequel des méthodes de résolutions efficaces existent.

Transformée par étalement

La transformée par étalement consiste à répartir chaque bit du message de tatouage sur une séquence pseudo-aléatoire secrète. Dans le tatouage QP, un bit du message n'est alors plus directement lié à une contrainte sur la moyenne d'une seule classe de l'histogramme, mais à une contrainte sur la projection de la moyenne d'une suite de classes sur la séquence porteuse. Le nombre de classes, initialement égal au nombre de bits, est alors multiplié par la taille de la séquence d'étalement. La transformée par étalement est intégrée en modifiant uniquement les inégalités associées à la contrainte de tatouage. En pratique, cette modification se réduit à la multiplication des contraintes par une matrice creuse contenant les coefficients de la séquence pseudo-aléatoire.

Formulation intégrale du centre de masse

La robustesse du tatouage QP face à la simplification est faible. Cela est principalement lié à l'utilisation d'un centre de masse calculé comme la moyenne de la position des sommets. Cette définition permet d'exprimer la contrainte de stabilité qui lui est associée comme une combinaison linéaire des variables de déplacements, mais elle est inadaptée pour des surfaces échantillonnées de manière non-uniforme, ou face à des attaques altérant la connectivité. Les définitions plus intégrales du centre de masse, par exemple utilisant une pondération des sommets par un volume ou une aire, sont plus robustes face à ce type d'altérations, mais elles ne sont pas linéaires. Pour intégrer ces définitions plus intégrales au tatouage QP, la contrainte de stabilité du centre de masse est linéarisée.

En pratique, cette linéarisation aboutit à la généralisation de l'équation de stabilité initiale en utilisant une nouvelle matrice creuse de dérivées partielles. Deux exemples de cette matrice sont présentés : le premier pour le centre de masse utilisant une pondération de type surfacique, le second pour le centre de masse pondéré par un volume.

Directions de déplacement arbitrairement pré-définies

Le tatouage QP restreint les déplacements des sommets aux directions radiales. Dans certaines configurations, ceci peut aboutir à un simple déplacement des sommets dans le plan tangent, ce qui diminue fortement la robustesse du tatouage. Afin d'ajouter de nouveaux degrés de liberté dans la minimisation, les variables d'optimisation associées aux déplacements, notées δr_i, sont dissociées des déplacements radiaux $\delta\rho_i$. Chaque sommet est alors déplacé suivant une direction $\mathbf{u}_i$ prédéfinie, qui n'est plus nécessairement la direction radiale. Le signal porteur du tatouage demeure néanmoins la moyenne des distances radiales.

Toutes les contraintes du tatouage QP sont modifiées pour traduire l'impact des déplacements des sommets, dans les directions $\mathbf{u}_i$, sur les distances radiales porteuses du message. Cette extension requiert seulement de linéariser une partie des contraintes dans le tatouage QP, en multipliant certains termes par une nouvelle matrice creuse de dérivées partielles. Les contraintes de causalité de l'histogramme sont en effet exprimables sans approximations, puisque les intersections entre les frontières de l'histogramme et les nouvelles directions d'altérations $\mathbf{u}_i$ peuvent être pré-calculées.

Deux nouvelles stratégies de tatouage sont proposées : le déplacement des sommets suivant leur normales, ou l'utilisation d'une direction incluse dans le plan tangent pour certaines zones de l'objet dans lesquelles toute modification suivant la normale serait particulièrement visible.

Amélioration de la métrique de distorsion

L'erreur quadratique, utilisée pour optimiser la fidélité du tatouage QP, mesure une distance dans l'espace tridimensionnel du maillage qui n'est que faiblement corrélée avec les distorsions perçues par un utilisateur. Aussi la fidélité du tatouage QP pourrait-elle bénéficier de l'intégration de métriques subjectives dans la fonction de coût de la minimisation, afin de mieux tenir compte du ressenti d'un utilisateur. Cependant, la plupart des métriques appropriées pour mesurer la distorsion perçue ne peuvent pas être exprimées sous forme quadratique.

Trois exceptions notables existent et permettent de résoudre ce problème : l'erreur quadrique, qui capture la distorsion introduite dans la direction normale, la distorsion mesurée à partir du Laplacien combinatoire associée au maillage, et la pondération de l'erreur quadratique suivant des informations sur la rugosité locale (wSE). Cette dernière solution s'appuie sur l'un des facteurs prépondérants dans la définition des métriques subjectives : dans les zones présentant une forte rugosité, les déplacements des sommets sont peu perceptibles, à l'inverse, dans les zones lisses, les modifications de la surface sont très visibles. Ces métriques sont chacune combinées avec l'erreur quadratique. Trois nouvelles fonctions de coût possibles sont donc définies, pour améliorer les performances du tatouage QP.

Analyse expérimentale des extensions

Les performances des extensions du tatouage QP sont testées sur une base de treize maillages, et comparées aux performances du système initial. Un message aléatoire est d'abord inséré à plusieurs reprises dans chacun des contenus en utilisant l'une des extensions. Les maillages générés sont ensuite attaqués, avant de mesurer le taux d'erreur binaire au décodage. Pour assurer une comparaison équitable, chacune des extensions est calibrée en alignant la distorsion introduite au moyen d'un paramètre α qui contrôle la force d'insertion du message dans le tatouage QP.

Les expériences menées indiquent que la transformée par étalement permet d'améliorer la flexibilité du compromis entre la distorsion et la robustesse dans le tatouage QP. Ainsi, la fidélité peut être accrue de manière conséquente, et atteindre un niveau en pratique inaccessible avec le système initial. Certaines tailles d'étalement permettent également d'accroître la robustesse face à de faibles attaques, mais augmentent la sensibilité face aux attaques plus fortes.

L'extension du tatouage aux centres de masses pondérés par des quantités intégrales présente un intérêt majeur dans le cas de la simplification, et le taux d'erreur binaire est fortement réduit. La robustesse de la pondération surfacique est néanmoins plus faible que celle du système de référence face aux attaques valumétriques. Les performances du centre de masse pondéré par le volume sont en revanche systématiquement supérieures à celles de la méthode originale.

L'extension s'appuyant sur l'une des trois nouvelles fonctions de coût permet d'améliorer la fidélité. Pour la métrique fondée sur le Laplacien et l'erreur quadratique pondérée, les performances

par rapport au système initial sont en moyenne légèrement améliorées. Pour l'erreur quadrique, les résultats sont équivalents à ceux du système QP de référence.

Le bilan du relâchement de la contrainte sur les directions d'altération est plus mitigé. Imposer un déplacement suivant les directions normales introduit une distorsion importante, qui est compensée par une force d'insertion minimale lors de la calibration suivant la fidélité. La robustesse est alors fortement réduite. Ce choix ne présente pas d'avantage certain par rapport au système de référence. En revanche, l'utilisation d'une direction de déplacement incluse dans le plan tangent pour les parties les plus lisses de l'objet est globalement bénéfique. Dans le cas de pièces mécaniques présentant de nombreuses parties planes, le système de référence introduit des vaguelettes particulièrement visibles. En adaptant les directions d'altération dans ces régions, on observe que l'extension proposée diminue fortement la visibilité du tatouage.

En résumé, les différentes extensions pour le tatouage QP permettent de répondre à plusieurs faiblesses de l'approche initiale en terme de robustesse et de fidélité. En exceptant le calcul de la rugosité, ces nouvelles possibilités affectent peu la durée d'encodage, et n'affectent pas du tout celle du décodage. Pour compléter ces extensions, deux autres aspects limitatifs pour le tatouage sont ensuite étudiés.

Sécurité du tatouage des distances radiales

Tous les paramètres du tatouage QP sont publics. La sécurité du système n'est donc pas assurée car un utilisateur peut accéder au canal de communication et modifier ou supprimer le message. Pour surmonter cet obstacle majeur à une mise en œuvre dans le contexte du traçage de traître, plusieurs mécanismes introduisant dans le tatouage des paramètres dérivés d'une clé secrète sont étudiés.

Protection de l'histogramme des distances radiales

La protection de l'histogramme consiste à générer un paramètre secret ϵ pour réduire l'intervalle de la distribution sur lequel l'histogramme est défini. Au lieu de prendre en compte l'ensemble de la distribution des distances radiales, les valeurs les plus faibles et les plus élevées ne sont pas utilisées. Sans ϵ, un utilisateur ne peut pas calculer l'histogramme sur lequel s'appuie le système de tatouage. La sécurité de ce mécanisme proposé pour le tatouage 3D n'a cependant pas été attentivement examinée et elle ne résiste pas à une attaque par recherche exhaustive. Bien que l'espace de recherche du paramètre secret soit grand, une estimation même approximative de ϵ suffit à accéder au canal de tatouage. Comme le tatouage altère la statistique naturelle de la distribution des distances radiales, l'adversaire peut vérifier, pour chaque clé testée, si la distribution est tatouée, et ainsi valider son estimation de la clé.

Une méthode efficace pour accroître l'espace de recherche et rendre l'attaque exhaustive plus coûteuse, consiste à retirer les extrémités de la distribution de façon asymétrique en remplaçant ϵ par deux paramètres $\epsilon_{\max}$ et $\epsilon_{\min}$. Pour contourner cette protection, un attaquant peut néanmoins s'appuyer sur la distorsion caractéristique du tatouage introduite dans la distribution des distances radiales. Il est en effet possible de détecter certaines frontières de l'histogramme, puis de créer un estimateur capable de reconstruire l'ensemble des classes. Plusieurs expériences illustrent l'efficacité de cette attaque, que l'on peut fortement réduire cependant par une modification judicieuse de la fonction de coût dans le tatouage QP.

Protection par étalement

La transformée par étalement constitue un paramètre secret du système de tatouage. La direction de projection, associée à une suite de classes de l'histogramme des distances radiales, est pseudo-aléatoire. Sans la clé secrète, l'attaquant ne peut pas projeter le signal porteur du tatouage et récupérer le message.

Cependant, si l'adversaire accumule des observations du signal tatoué, il peut estimer statistiquement la direction de projection par une analyse en composante principale. Une étude expérimentale montre qu'un attaquant peut accéder au canal de tatouage à partir d'une dizaine d'objets tatoués. Comme le signe de chaque bit est indéterminé, cet accès est néanmoins limité, et le message peut être modifié, mais non pas lu.

Dans le cas où l'adversaire a également accès au message de tatouage associé à chacune des observations, l'analyse en composantes principales peut être remplacée par une analyse discriminante linéaire. Les performances de l'attaque sont alors décuplées, et l'attaquant peut accéder sans limite au canal de tatouage car l'indétermination sur le signe de chaque bit est levée.

Protection par permutation

La faiblesse de la protection par étalement réside dans la possibilité d'effectuer une attaque statistique, dans un espace de faible dimension égale à la taille de la séquence d'étalement associée à un bit, plutôt que dans un espace d'une grande dimension égale au nombre de classes de l'histogramme. Ceci est dû au caractère public de l'association entre les groupes de classes de l'histogramme et les groupes de valeurs dans la séquence d'étalement (ces valeurs, elles, sont secrètes). Un mécanisme de sécurité supplémentaire consiste à modifier par une permutation pseudo-aléatoire cette association. L'adversaire ne peut plus appliquer l'approche statistique précédente, puisqu'il ne peut plus découper le signal de tatouage observé en sous-séquences consécutives de faible dimension, avant de les attaquer de façon indépendante.

Cependant, l'information mutuelle entre les moyennes des classes de l'histogramme, estimées à partir de plusieurs observations d'objets tatoués, peut être utilisée pour retrouver les groupes de classes associés à chaque bit du message. Un exemple concret d'attaque est présenté en utilisant une approche gloutonne pour identifier la permutation aléatoire secrète et la séquence d'étalement. Les performances de l'attaque sont par ailleurs complexes à mesurer, ce qui nécessite l'utilisation de métrique *ad hoc*. Même si les performances de cette attaque sont réduites par rapport au cas précédent, car les estimations de l'adversaire sont toujours en partie erronées avec plusieurs centaines d'objets, elle permet toutefois d'accéder à une grande partie de l'information protégée.

Chacun des mécanismes de protection proposé présente donc des failles, mais celles-ci peuvent être réduites, et la combinaison de ces mécanismes les rend efficaces.

Resynchronisation face au rognage

Le tatouage QP et la plupart des systèmes aveugles dans le domaine spatial présentent une faible robustesse face au rognage. Pour résister à ce type d'attaque, des mécanismes de "resynchronisation implicite" ont été proposés. Le maillage est d'abord segmenté de manière canonique en fonction de ses points d'intérêt, puis le message est entièrement inséré dans chacun des segments. Il suffit qu'un seul soit préservé par le rognage pour décoder correctement le tatouage. Cette approche présente néanmoins des inconvénients : la segmentation peut être elle-même affectée par le rognage, et les points d'intérêts sont difficiles à localiser de façon robuste et précise. Les performances de ces

systèmes de tatouage présentent alors un compromis implicite entre la robustesse aux attaques valumétriques (bruit, lissage, etc.) et aux attaques de désynchronisation (rognage).

Le composant de resynchronisation proposé s'inscrit dans une démarche novatrice, qui consiste à procurer au décodeur QP la position du centre de masse du maillage initial à partir d'une estimation sur la version attaquée. Cette solution s'attache également à garantir que les performances face aux attaques valumétriques ne soient pas diminuées par la resynchronisation.

Points de recalage

La resynchronisation proposée repose sur l'insertion de points de recalage dans le maillage à des positions arbitrairement définies, puis à leur récupération de manière aveugle. Un point de recalage est un sommet au voisinage duquel la forme du maillage appartient à une classe prédéfinie. En pratique, cette classe "cible" correspond à l'ensemble des paraboloïdes dont les paramètres sont alignés sur une grille de quantification secrète.

Pour créer un nouveau point de recalage, la surface est localement altérée par l'intermédiaire d'une optimisation sous contrainte, qui minimise la distorsion introduite par le déplacement des sommets. L'objectif est de faire en sorte que le voisinage du point soit proche d'un paraboloïde de la classe cible. Pour récupérer les points de recalage, le voisinage de chacun des sommets du maillage est modélisé par un paraboloïde dont les paramètres sont estimés. Le score s_i associé à un sommet caractérise l'alignement de ces paramètres par rapport à la grille de quantification. Les scores élevés correspondent aux points de recalage.

Intuitivement, l'insertion et la détection des points de recalage définissent un système de tatouage dont le signal porteur est formé par les paramètres du paraboloïde, et qui utilise une technique de communication informée à l'émission.

Composant de resynchronisation

Avant d'appliquer le tatouage QP, le composant de resynchronisation introduit des points de recalage localisés à l'intersection entre le maillage et une sphère dont e centre coïncide avec le centre de masse du maillage. Ces points sont également choisis de telle sorte que leur voisinage soit disjoints. Au cours du tatouage QP, une contrainte supplémentaire garantit que les points de recalage et leur voisinage ne sont pas modifiés.

Au décodage, la resynchronisation détecte de manière aveugle la position des points de recalage. En s'appuyant sur l'hypothèse que ces points sont situés sur une sphère, les faux positifs peuvent être efficacement éliminés par une approche RANSAC. Cette procédure permet d'estimer la position du centre de la sphère, qui coïncide avec le centre de masse de l'objet initial. En cas de rognage, le décodeur QP utilise le centre de la sphère sur laquelle les points de recalage sont placés, pour calculer les distances radiales.

Face à des attaques valumétriques, une telle solution devient inefficace car le centre des points de recalage est plus instable que le centre de masse. Le composant de resynchronisation utilise donc le score de confiance s_i pour vérifier la fiabilité de la position des points de recalage. Lorsque s_i est faible, le décodeur QP n'utilise pas le composant de resynchronisation, et s'appuie sur l'estimation du centre de masse du maillage attaqué.

Les performances de la resynchronisation sont mesurées de la même manière que les précédentes extensions du tatouage QP. La principale limite de cette approche réside dans lincapacité du système à gérer la combinaison d'un rognage avec des attaques valumétriques. En revanche les résultats obtenus montrent que les performances face au rognage sont fortement améliorées, tandis que la robustesse face aux autres attaques est entièrement préservée.

Appendix C

Conclusion

Les recherches présentées dans cette dissertation se concentrent sur le tatouage 3D robuste et aveugle pour protéger les droits d'auteur associés aux maillages surfaciques triangulaires, dans le contexte du traçage de traître. L'aperçu proposé de l'état de l'art, et la comparaison des transformées d'adaptation du contenu ont d'abord illustré les limites et les enjeux de ce domaine.

Les objets 3D sont représentés par des maillages surfaciques plongés dans $\mathbb{R}^3$. Ceci constitue un problème pour les domaines d'insertion du tatouage de type spectral et multirésolution, qui sont par ailleurs très appréciés. Utiliser ces domaines rend le tatouage sensible aux altérations de la connectivité du maillage, car leurs bases dépendent du contenu. Des problèmes de synchronisation apparaissent alors pour les méthodes aveugles. Les stratégies pour surmonter ces problèmes, comme les "manifold harmonics" ou un remaillage précédant la décomposition en ondelettes, introduisent à leur tour de nouveaux obstacles en termes de causalité et de synchronisation.

Dans le domaine spatial, les quantités géométriques intrinsèques ou définies sur un domaine spatial très localisé peuvent théoriquement résister à la pose et au rognage. En pratique, elles ont cependant leurs limites propres en terme de synchronisation. Elles sont également plus sensibles aux attaques valumétriques, à l'inverse des quantités géométriques calculées de façon globale sur le maillage. Lorsqu'elles sont définies de manière intégrale, ces quantités globales sont également robustes aux attaques altérant la connectivité. Leur performance face au rognage est en revanche généralement insuffisant pour une mise en œuvre pratique.

Contributions

Les chapitres qui précédent ont détaillé plusieurs approches pour surmonter les difficultés évoquées.

Le chapitre 5 a introduit un estimateur robuste de l'épaisseur, qui permet de créer un espace de tatouage dans lequel les problèmes de synchronisation liés à la pose sont réduits. Notre estimateur est fondé sur une approximation de l'épaisseur par le diamètre qui a été proposée dans la littérature. Il présente une précision et une robustesse supérieures à l'état de l'art et à la "Shape Diameter Function (SDF)". L'épaisseur est une quantité prometteuse pour définir un signal porteur du tatouage robuste face à la pose. Les performances de cette approche nouvelle pour le tatouage 3D dans le contexte d'animations sont supérieures à celles mesurées pour les quantités intrinsèques ou définies sur un domaine restreint. Néanmoins, la complexité de la fonction de fusion est actuellement prohibitive, car la corrélation des mesures d'épaisseur estimées à différents endroits est difficile à prévoir.

Les chapitres 6 et 7 se sont concentrés sur l'extension d'une approche du tatouage 3D formulé comme un problème d'optimisation quadratique sous contrainte (QP). Le signal porteur du

tatouage est alors constitué par les distances radiales. Le principal objectif de l'approche initiale était de résoudre le problème de causalité du tatouage. Les recherches présentées ont étendu cette solution pour créer un cadre de tatouage modulaire, dans lequel la robustesse et la fidélité peuvent être contrôlées avec une plus grande précision. Plusieurs systèmes définis dans ce cadre ont été minutieusement comparés. Des bénéfices théoriques et pratiques notables ont été identifiés, grâce à l'utilisation de quantités intégrales, l'ajout de degrés de liberté dans l'optimisation, et l'amélioration des métriques de fidélité minimisées.

Plus précisément, la robustesse face à certaines altérations de la connectivité, telles que la simplification, est fortement augmentée dans le cas d'une définition du centre de gravité pondérée par le volume. En utilisant la "Mesh Structural Distortion Measure (MSDM)" à la place de l'erreur quadratique moyenne pour calibrer la comparaison des systèmes en alignant leur fidélité, les bénéfices des autres extensions sur l'imperceptibilité du tatouage sont significatifs. Pour des pièces mécaniques, les améliorations sont particulièrement importantes : contrairement à la plupart des systèmes utilisant les distances radiales, les parties lisses du maillage peuvent être aisément préservées.

Ces extensions offrent également la possibilité de modifier la fonction d'insertion, illustrée par l'utilisation et l'étude d'une transformée par étalement. En plus de ses avantages pour le compromis robustesse–fidélité, elle permet d'augmenter la sécurité du système de tatouage. Le chapitre 8 a introduit une analyse systématique de la sécurité de notre cadre de tatouage, en commençant par son seul mécanisme de sécurité connu, qui ajoute un paramètre de décalage secret. Une série d'attaques et de contre-mesures a été présentée, en s'appuyant sur la flexibilité du cadre de tatouage pour introduire de nouveaux mécanismes de sécurité.

Enfin, le chapitre 9 traite du problème de la resynchronisation face aux attaques par rognage dans notre cadre de tatouage QP, et un nouveau composant est proposé. Avant l'insertion du tatouage, des points de recalage secrets sont générés sur le maillage suivant une configuration précise (par exemple une sphère). Ces points de recalage sont récupérés par le décodeur de manière aveugle, et leur configuration contient les informations de resynchronisation nécessaires pour le décodage dans le cas d'un rognage. Les résultats expérimentaux illustrent les bénéfices et l'efficacité de cette méthode de resynchronisation. Son intégration dans le cadre du QP est par ailleurs simple. Ces recherches permettent ainsi de surmonter l'une des principales difficultés pour le tatouage 3D sans faire intervenir la stratégie, souvent défendue, d'un partitionnement et d'une répétition du message.

Perspectives à court terme

L'analyse de l'estimation robuste de l'épaisseur pour les objets 3D a mis en évidence les difficultés restantes pour définir un domaine d'insertion fondé sur l'épaisseur. C'est la définition de la fonction de fusion qui constitue actuellement le problème majeur, et non plus la robustesse à la pose. Une solution serait de considérer d'autres estimateurs de l'épaisseur que ceux liés au diamètre. Des recherches en ce sens nécessiteraient une comparaison minutieuse des avantages et des inconvénients des nombreuses approches proposant une transformée de l'axe médian robuste.

Le cadre de tatouage pour les distances radiales peut être encore complété par d'autres extensions. Des métriques de fidélité encore plus évoluées pourraient par exemple être utilisées. L'objectif de ces nouvelles recherches seraient d'aboutir à une minimisation directe d'une métrique liée à la distorsion perçue ; une telle métrique n'étant pas quadratique. Des définitions plus intégrales du signal porteur du tatouage pourraient aussi être employées, par exemple avec un histogramme pondéré par le volume, avec des définitions ne se restreignant pas aux sommets mais en utilisant, par exemple, les centres de facettes ou des positions arbitraires sur le maillage. Cela pourrait limiter

davantage la dépendance du tatouage à l'échantillonnage de la surface.

En ce qui concerne la sécurité, combiner toutes les contre-mesures présentées au chapitre 8 permet d'empêcher un adversaire d'utiliser les attaques décrites. Cependant, cela n'est peut-être pas suffisant pour de véritables applications du traçage de traître. Par ailleurs, cette étude s'est focalisée sur le cadre de tatouage proposé, mais la plupart des systèmes de tatouage fondés sur les distances radiales présentent des failles similaires. Adapter ces attaques pour ces systèmes est une piste de recherche prometteuse pour améliorer la sécurité du tatouage 3D.

Le composant de resynchronisation introduit au chapitre 9 offre des perspectives intéressantes pour résister au rognage. Par exemple, l'utilisation de configurations plus complexes, comme des ellipsoïdes, permettrait de transmettre encore plus d'informations au décodeur. La plupart des systèmes utilisant les distances radiales pourraient bénéficier de ces recherches. Enfin, les points de recalage pourraient ne pas être restreints à une signature dérivée des coefficients d'un paraboloïde, mais liés à des descripteurs de forme plus complexes.

Perspectives à long terme

Trois principaux sujets de recherche restent à explorer pour que le tatouage 3D devienne utilisable dans des applications concrètes.

Premièrement, les techniques de compression progressive font maintenant partie des systèmes de distribution 3D, et les méthodes de tatouage devraient être capables de décoder le message à différents niveaux de détails. Deuxièmement, l'une des principales utilisations des maillages dans l'industrie du divertissement est la création d'animations, alors que le tatouage 3D s'intéresse principalement aux maillages statiques qui ne sont pas animés. Les méthodes de tatouage pour des animations 3D considèrent le contenu comme un signal 3D variant au cours du temps : au lieu du maillage, ce sont les mouvements des sommets qui sont tatoués. Ces approches ne sont pas robustes à un changement d'animation, et un tatouage 3D fondé sur des quantités stables face à la pose, telles que l'épaisseur ou les distances géodésiques, est donc nécessaire.

Troisièmement, en raison de l'arrivée d'attaques de type impression et numérisation pour les maillages 3D, les systèmes de tatouage devront gérer des conversions complexes, par exemple le passage d'une représentation surfacique à une représentation volumique, avant d'être intégrés dans des environnements de création et de distribution. L'un des principaux freins dans cette direction est la capacité à mener des campagnes de tests. Contrairement à la vidéo, pour laquelle de petites campagnes peuvent être menées avec une caméra, effectuer des tests de type impression et numérisation en 3D est actuellement trop coûteux.

Appendix D

Database

D.1 Original Meshes

Model	n_v	n_f	Model	n_v	n_f
Armadillo	24,473	48,942	*Hand*	36,619	72,958
Bunny	34,835	69,666	*Head*	15,941	31,620
Caesar	27,726	55,448	*Hippo*	49,057	98,110
David	23,889	47,280	*Horse*	112,642	225,280
Dragon	50,000	100,000	*Rabbit*	70,658	141,312
Elephant	24,955	49,918	*Venus*	100,759	201,514
Fandisk	25,894	51,784			

Table D.1: Database of 3D models used for benchmarking and their complexity.

D.2 Watermarked Meshes

Figure D-1: Some of the meshes in the database.

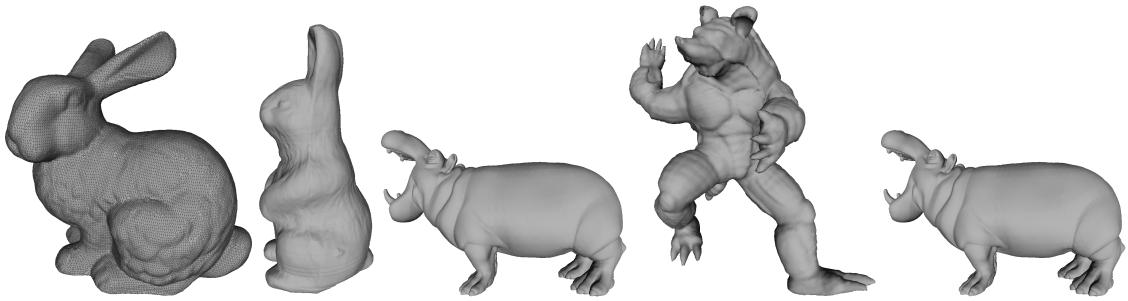


Figure D-2: Examples of watermarked meshes.

Index

Acronyms

AUC Area Under the Curve.

AWGN Additive white Gaussian noise.

BER Bit Error Rate.

CAD Computer-aided Design.

CDMA Code Division Multiple Access.

DCT Discrete Cosine Transform.

DFT Discrete Fourier Transform.

DWT Discrete Wavelet Transform.

EER Equal Error Rate.

FFT Fast Fourier Transform.

FLD Fisher's Linear Discriminant.

FPR False Positive Rate.

HVS Human Visual System.

ICA Independent Component Analysis.

ICP Iterative Closest Point.

ISS Improved Spread Spectrum.

KMA Known-Message Attack.

MAT Medial Axis Transform.

MRMS Maximum Root Mean Square.

MSDM Mesh Structural Distortion Measure.

MSE Mean Square Error.

OFF Object File Format.

PCA Principal Component Analysis.

QEM Quadric Error Metric.

QIM Quantization Index Modulation.

QP Quadratic Programming.

RANSAC RANdom SAmple Consensus.

RDM Rational Dither Modulation.

RMS Root Mean Square.

ROC Receiver Operating Characteristic.

SCS Scalar Costa Scheme.

SDF Shape Diameter Function.

SE Square Error.

SS Spread Spectrum.

ST Spread Transform.

STDM Spread Transform Dither Modulation.

SVD Singular Value Decomposition.

TPR True Positive Rate.

TSPS Triangle Strip Peeling Symbol.

VFA Vertex Flood Algorithm.

WOA Watermarked-Content Only Attack.

wSE weighted Square Error.

Bibliography

[ABE09] Dominique Attali, Jean-Daniel Boissonnat, and Herbert Edelsbrunner. Stability and computation of medial axes, a state-of-the-art report. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, pages 109–125. Springer, 2009.

[ACSD+03] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. Anisotropic polygonal remeshing. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 485–493, 2003.

[AG05] Pierre Alliez and Craig Gotsman. Recent Advances in Compression of 3D Meshes. In Neil A. Dodgson, Michael S. Floater, and Malcolm A. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 3–26. Springer Berlin Heidelberg, 2005.

[Ahr06] Frank Ahrens. U.S. Joins Industry in Piracy War. *The Washington Post*, June 2006. http://www.washingtonpost.com/wp-dyn/content/article/2006/06/14/AR2006061402071.html, Accessed: 2014-08-06.

[AM05] Patrice Rondao Alface and Benoit Macq. Blind watermarking of 3d meshes using robust feature points detection. In *IEEE International Conference on Image Processing*, 2005.

[APDP10] P. Amat, William Puech, Sébastien Druon, and Jean-Pierre Pedeboy. Lossless 3d steganography based on mst and connectivity modification. *Signal Processing: Image Communication*, 25(6):400–412, 2010.

[ASCE02] Nicolas Aspert, Diego Santa-Cruz, and Touradj Ebrahimi. MESH: measuring errors between surfaces using the Hausdorff distance. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, volume 1, pages 705–708, August 2002.

[Aut14] Autodesk. Autodesk maya, 2014. http://www.autodesk.com/products/maya/overview, Accessed: 2014-08-06.

[BB00] Oliver Benedens and Christoph Busch. Towards blind detection of robust watermarks in polygonal models. In *Proceedings of the Computer Graphics Forum*, 2000.

[BBC+03] M. Barni, F. Bartolini, V. Cappellini, M. Corsini, and A. Garzelli. Digital watermarking of 3d meshes. In *Proceedings SPIE*, volume 5208, August 2003.

[BC07] Patrick Bas and François Cayre. Natural Watermarking: A Secure Spread Spectrum Technique for WOA. In Jan L. Camenisch, Christian S. Collberg, Neil F. Johnson, and Phil Sallee, editors, *Information Hiding*, volume 4437 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2007.

[BD06] Jihane Bennour and Jean-Luc Dugelay. Protection of 3D object through silhouette watermarking. In *ICASSP 2006, 31st International Conference on Acoustics, Speech, and Signal Processing, May 14-19, 2006, Toulouse, France*, May 2006.

[Ben99] Oliver Benedens. Geometry-based watermarking of 3d models. *IEEE Comput. Graph. Appl.*, 19:46–55, January 1999.

[BF13] Patrick Bas and Teddy Furon. A new measure of watermarking security: The effective key length. *IEEE Transactions on Information Forensics and Security*, 8(8):1306–1317, August 2013.

[BKP+10] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Levy. *Polygon Mesh Processing*. AK Peters, 2010.

[Ble14] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2014. http://www.blender.org, Accessed: 2014-08-06.

[Blu67] Harry Blum. A Transformation for Extracting New Descriptors of Shape. In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.

[BM92] Paul J. Besl and Neil D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[BOZHP13] Ines Bouzidi, Azza Ouled Zaid, Meha Hachani, and William Puech. Joint Watermarking and Progressive Geometric Compression of 3D Meshes. In *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*, pages 209–214, 2013.

[CAS+03] François Cayre, Patrice Rondao Alface, Francis Schmitt, Benoit Macq, and Henri Maïtre. Application of spectral decomposition to compression and watermarking of 3D triangle mesh geometry. In *Signal Processing: Image Communication*, volume 18, April 2003.

[CCSM10] Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric Inference for Measures based on Distance Functions. Research report RR-6930, INRIA, 2010.

[CDF06] Ingemar J. Cox, Gwenaël Doërr, and Teddy Furon. Watermarking is not Cryptography. In *Proceedings of the International Workshop on Digital Watermarking*, volume 4283 of *Lecture Notes in Computer Science*, pages 1–15, November 2006.

[CFF05] François Cayre, Caroline Fontaine, and Teddy Furon. Watermarking security: Theory and practice. *IEEE Transactions on Signal Processing*, 53(10):3976–3987, October 2005.

[CGA] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[CGEB07] Massimiliano Corsini, Elisa Drelie Gelasca, Touradj Ebrahimi, and Mauro Barni. Watermarked 3-D mesh quality assessment. *IEEE Transactions on Multimedia*, 9(2):247–256, February 2007.

[CLL+13] Massimiliano Corsini, Mohamed-Chaker Larabi, Guillaume Lavoué, Oldrich Petrík, Libor Vása, and Kai Wang. Perceptual metrics for static and dynamic triangle meshes. *Computer Graphics Forum*, 32(1):101–125, February 2013.

[CM03] François Cayre and Benoit Macq. Data hiding on 3-d triangle meshes. *IEEE Transactions on Signal Processing*, 51:939–949, April 2003.

[CMB+07] Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers Inc., 2nd edition, 2007.

[CP03] Frédéric Cazals and Marc Pouget. Estimating Differential Quantities Using Polynomial Fitting of Osculating Jets. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, pages 177–187, 2003.

[CP08] Frédéric Cazals and Marc Pouget. Algorithm 889: Jet_Fitting_3:—A Generic C++ Package for Estimating the Differential Properties on Sampled Surfaces via Polynomial Fitting. *ACM Transactions on Mathematical Software*, 35(3):24:1–24:20, October 2008.

[CPJ07] Jae-Won Cho, Rémy Prost, and Ho-Youl Jung. An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. *IEEE Transactions on Signal Processing*, 55(1):142–155, January 2007.

[CRS98] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, June 1998.

[CSM03] David Cohen-Steiner and Jean-Marie Morvan. Restricted Delaunay Triangulations and Normal Cycle. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, SCG '03, pages 312–321, 2003.

[CW99] Brian Chen and Gregory W. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47:1423–1443, 1999.

[CWW13] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in Heat: A New Approach to Computing Distance Based on Heat Flow. *ACM Transactions on Graphics*, 32(5), October 2013.

[DHKL01] Nira Dyn, Kai Hormann, Sun-Jeong Kim, and David Levin. Optimizing 3d triangulations using discrete curvature analysis. In Tom Lyche and Larry L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces: Oslo 2000*, Innovations in Applied Mathematics, pages 135–146. Vanderbilt University Press, 2001.

[DHM10] Rony Darazi, Roland Hu, and Benoit Macq. Applying spread transform dither modulation for 3D-mesh watermarking by using perceptual models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1742–1745, March 2010.

[EBTG03] J.J. Eggers, R. Bauml, R. Tzschoppe, and B. Girod. Scalar Costa scheme for information embedding. *Signal Processing, IEEE Transactions on*, 51(4):1003–1019, April 2003.

[EDD+95] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution Analysis of Arbitrary Meshes. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 173–182, 1995.

[Epi14] Epic Games. Unreal Engine 3 - Unreal Development Kit, 2014. https://www.unrealengine.com/products/udk/, Accessed: 2014-08-06.

[ESP08] Michael Eigensatz, Robert W. Sumner, and Mark Pauly. Curvature-Domain Shape Processing. *Computer Graphics Forum*, 27(2):241–250, 2008.

[EsRT+13] Samir Abou El-seoud, Nadine Abu Rumman, Islam A. T. F. Tajeddin, Khalaf F. Khatatneh, and Christain Gutl. Robust Digital Watermarking for Compressed 3D Models based on Polygonal Representation. *International Journal of Computer Applications*, 61(4):1–14, January 2013.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[FC14] Dan Farr and Chris Creek. Digital Art Zone (DAZ) 3D, August 2014. http://www.daz3d.com/new-releases/, Accessed: 2014-08-06.

[FGK+98] Andreas Fabri, Geert-Jan Giezeman, Lutz Kettner, Stefan Schirra, and Sven Schönherr. On the Design of CGAL, the Computational Geometry Algorithms Library. Technical Report RR-3407, INRIA, April 1998.

[GAP08] Ankit Gupta, Pierre Alliez, and Sylvain Pion. Principal component analysis in CGAL. Technical Report 6642, INRIA Sophia-Antipolis, September 2008.

[Gar99] M. Garland. Multiresolution modeling: Survey & future opportunities. In *EUROGRAPHICS '99 – State of the Art Reports*, pages 111–131, 1999.

[GH97] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 209–216, August 1997.

[GMPW09] Joachim Giesen, Balint Miklos, Mark Pauly, and Camille Wormser. The scale axis transform. In *Proceedings of the 25th annual Symposium on Computational Geometry*, pages 106–115. ACM, 2009.

[Hop96] Hugues Hoppe. Progressive Meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 99–108, 1996.

[Hop99] Hugues Hoppe. Optimization of mesh locality for transparent vertex caching. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 269–276, 1999.

[HPPLG11] Paul Heider, Alain Pierre-Pierre, Ruosi Li, and Cindy Grimm. Local Shape Descriptors, a Survey and Evaluation. In *Proceedings of the 4th Eurographics Conference on 3D Object Retrieval*, pages 49–56, 2011.

[HRAM09] Roland Hu, Patrice Rondao-Alface, and Benoit Macq. Constrained optimisation of 3D polygonal mesh watermarking by quadratic programming. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1501–1504, April 2009.

[HXYD14] Roland Hu, Li Xie, Huimin Yu, and Baocang Ding. Applying 3D Polygonal Mesh Watermarking for Transmission Security Protection through Sensor Networks. *Mathematical Problems in Engineering*, 2014.

[IKLS97] Cox Ingemar, Joe Kilian, Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, December 1997.

[JDBP04] Jian-Qiu Jin, Min-Ya Dai, Hu-Jun Bao, and Qun-Sheng Peng. Watermarking on 3D mesh based on spherical wavelet transform. *Journal of Zhejiang University SCIENCE*, 5(3):251–258, 2004.

[Ju09] Tao Ju. Fixing Geometric Errors on Polygonal Models: A Survey. *Journal of Computer Science and Technology*, 24(1):19–29, 2009.

[Kal01] Ton Kalker. Considerations on watermarking security. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, pages 201–206, October 2001.

[KBT10] Kwangtaek Kim, Mauro Barni, and Hong Z. Tan. Roughness-adaptive 3-D watermarking based on masking effect of surface roughness. *IEEE Transactions on Information Forensics and Security*, 5(4):721–733, December 2010.

[KDK98] S. Kanai, H. Date, and T. Kishinami. Digital watermarking for 3d polygons using multiresolution wavelet decomposition. In *Proc. Sixth IFIP WG 5.2 GEO-6*, pages 296–307, 1998.

[KG00a] Zachi Karni and Craig Gotsman. Spectral Compression of Mesh Geometry. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 279–286, 2000.

[KG00b] Zachi Karni and Craig Gotsman. Spectral compression of mesh geometry. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 279–286, July 2000.

[KGMS10] Maurizio Kovacic, Fabio Guggeri, Stefano Marras, and Riccardo Scateni. Fast approximation of the shape diameter function. In *Proceedings Workshop on Computer Graphics, Computer Vision and Mathematics (GraVisMa) 2010*, volume 5, 2010.

[KLT05] Sagi Katz, George Leifman, and Ayellet Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21:649–658, 2005.

[KMD+09] John M. Konstantinides, Athanasios Mademlis, Petros Daras, Pericles A. Mitkas, and Michael G. Strintzis. Blind robust 3-D mesh watermarking based on oblate spheroidal harmonics. *IEEE Transactions on Multimedia*, 11(1):23–38, January 2009.

[KTP03] Andreas Kalivas, Anastasios Tefas, and Ioannis Pitas. Watermarking of 3d models using principal component analysis. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 676–679, April 2003.

[KVJP05] M.-S. Kim, S. Valette, H.-Y. Jung, and R. Prost. Watermarking of 3d irregular meshes based on wavelet multiresolution analysis. In *IWDW'05*, pages 313–324. Springer, 2005.

[Lav09] Guillaume Lavoué. A Local Roughness Measure for 3D Meshes and Its Application to Visual Masking. *ACM Transactions on Applied Perception*, 5(4):21:1–21:23, February 2009.

[Lav11] Guillaume Lavoué. A multiscale metric for 3D mesh visual quality assessment. *Computer Graphics Forum*, 30(5):1427–1437, August 2011.

[LB11] Ming Luo and Adrian G. Bors. Surface-preserving robust watermarking of 3-D shapes. *IEEE Transactions on Image Processing*, 20(10):2813–2826, October 2011.

[LB13] Ming Luo and Adrian G. Bors. Optimized 3D watermarking for minimal surface distortion. *IEEE Transactions on Image Processing*, 22:1822–1835, May 2013.

[LC10] Guillaume Lavoué and Massimiliano Corsini. A comparison of perceptually-based metrics for objective evaluation of geometry processing. *IEEE Transactions on Multimedia*, 12(7):636–649, November 2010.

[LDD+06] Guillaume Lavoué, Elisa Drelie Gelasca, Florent Dupont, Atilla Baskurt, and Touradj Ebrahimi. Perceptually-driven 3D distance metrics with application to watermarking. In *Applications of Digital Image Processing XXIX*, Proceedings of SPIE, August 2006.

[LDD07] Guillaume Lavoué, Florence Denis, and Florent Dupont. Subdivision surface watermarking. *Computers & Graphics*, 31(3):480–492, 2007.

[LDLD11] Ho Lee, Cagatay Dikici, Guillaume Lavoué, and Florent Dupont. Joint Reversible Watermarking and Progressive Compression of 3D Meshes. *The Visual Computer*, 27:781–792, June 2011.

[LDW94] Michael Lounsbery, Tony DeRose, and Joe Warren. *Multiresolution Analysis for Surfaces of Arbitrary Topological Type*. PhD thesis, University of Washington, January 1994.

[LK07] Suk-Hwan Lee and Ki-Ryong Kwon. A Watermarking for 3D Mesh Using the Patch CEGIs. *Digital Signal Processing*, 17(2):396–413, March 2007.

[LPG08] Y. Liu, B. Prabhakaran, and X. Guo. A robust spectral approach for blind watermarking of manifold surfaces. In *Proceedings of the 10th ACM workshop on Multimedia and security*, pages 43–52. ACM, 2008.

[LT98] Peter Lindstrom and Greg Turk. Fast and Memory Efficient Polygonal Simplification. In *Proceedings of the Conference on Visualization '98*, pages 279–286. IEEE Computer Society Press, 1998.

[Luo06] Ming Luo. *Robust and Blind 3D Watermarking*. PhD thesis, University of York, UK, May 2006.

[LWBL09] Ming Luo, Kai Wang, Adrian G. Bors, and Guillaume Lavoué. Local patch blind spectral watermarking method for 3d graphics. In *Proceedings of the 8th International Workshop on Digital Watermarking*, 2009.

[Mac14] Laura MacFarlane. Canada: The Legal Perils Of 3D Printing, February 2014. http://www.mondaq.com/canada/x/292362/Patent/The+Legal+Perils+of+3D+Printing, Accessed: 2014-08-06.

[MF03] Henrique S. Malvar and Dinei A. F. Florêncio. Improved Spread Spectrum: A New Modulation Technique for Robust Watermarking. *IEEE Transactions on Signal Processing*, 51(4):898–905, April 2003.

[MN03] Niloy J. Mitra and An Nguyen. Estimating Surface Normals in Noisy Point Cloud Data. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, SCG '03, pages 322–328, 2003.

[New12] John Newman. 3D Printing, IP and Industry: Opening Shots, June 2012. http://www.rapidreadytech.com/2012/06/3d-printing-ip-and-industry-opening-shots/, Accessed: 2014-08-06.

[OMA97] Ryutarou Ohbuchi, Hiroshi Masuda, and Masaki Aono. Watermarking Three-Dimensional Polygonal Models. In *Proceedings of the fifth ACM internation conference on Multimedia*, pages 261–272. ACM Press, 1997.

[OMA98] Ryutarou Ohbuchi, Hiroshi Masuda, and Masaki Aono. Data embedding algorithms for geometrical and non-geometrical targets in three-dimensional polygonal models. *Computer Communications, Vol.21, Issue*, 15:1344–1354, 1998.

[OMT02] Ryutarou Ohbuchi, Akio Mukaiyama, and Shigeo Takahashi. A frequency-domain approach to watermarking 3d shapes. *Computer Graphics Forum*, 21:373–382, 2002.

[OMT04] Ryutarou Ohbuchi, Akio Mukaiyama, and Shigeo Takahashi. Watermarking a 3D Shape Model Defined as a Point Set. In *Proceedings of the IEEE International Conference on Cyberworlds*, pages 392–399, 2004.

[Org08] Organisation for Economic Co-operation and Development. *The Economic Impact of Counterfeiting and Piracy*. OECD, June 2008.

[PC06] Gabriel Peyré and Laurent D. Cohen. Geodesic Remeshing Using Front Propagation. *International Journal of Computer Vision*, 69(1):145–156, August 2006.

[PCB05] Yixin Pan, Irene Cheng, and A. Basu. Quality Metric for Approximating Subjective Evaluation of 3-D Objects. *IEEE Transaction on Multimedia*, 7(2):269–279, April 2005.

[Pey11] Gabriel Peyré. The Numerical Tours of Signal Processing - Advanced Computational Signal and Image Processing. *IEEE Computing in Science and Engineering*, 13(4):94–97, 2011.

[PFCnPG05] Luis Pérez-Freire, Pedro Comesaña, and Fernando Pérez-Gonzalez. Information-Theoretic Analysis of Security in Side-Informed Data Hiding. In Mauro Barni, Jordi Herrera-Joancomart, Stefan Katzenbeisser, and Fernando Pérez-González, editors,

Information Hiding, volume 3727 of *Lecture Notes in Computer Science*, pages 131–145. Springer Berlin Heidelberg, 2005.

[PFPGCn06] Luis Pérez-Freire, Fernando Pérez-González, and Pedro Comesaña. Secret dither estimation in lattice-quantization data hiding: a set membership approach. In *Proceedings of the SPIE conference on Security, Steganography, and Watermarking of Multimedia Contents*, February 2006.

[PGMBA05] Fernando Pérez-Gonzalez, C. Mosquera, Mauro Barni, and A Abrardo. Rational dither modulation: a high-rate data-hiding method invariant to gain attacks. *IEEE Transactions on Signal Processing*, 53(10):3960–3975, October 2005.

[PHF99] E. Praun, H. Hoppe, and A. Finkelstein. Robust mesh watermarking. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 49–56. ACM Press/Addison-Wesley Publishing Co., 1999.

[PHOZ12] William Puech, Meha Hachani, and Azza Ouled Zaid. Robust Mesh Data Hiding Based on Irregular Wavelet Transform. In *EUSIPCO'12: 20th European Signal Processing Conference*, pages 1742–1746, August 2012.

[Pub13] Imagine Publishing. *The 3D Art & Design Book Volume 2*. Imagine Publishing Ltd, 2013.

[PWHY09] Helmut Pottmann, J. Wallner, Q.-X. Huang, and Y.-L. Yang. Integral invariants for robust geometry processing. *Computer Aided Geometric Design*, 26:37–60, January 2009.

[RAMC07] Patrice Rondao Alface, Benoit Macq, and François Cayre. Blind and Robust Watermarking of 3D Models: How to Withstand the Cropping Attack? In *Proceedings of the IEEE International Conference on Image Processing, 2007*, volume 5, pages 465–468, September 2007.

[Reg14] Stephen Regelous. MASSIVE (Multiple Agent Simulation System in Virtual Environment), 2014. http://www.massivesoftware.com/index.html, Accessed: 2014-08-06.

[SB11] Ivan Sipiran and Benjamin Bustos. Harris 3D: A Robust Extension of the Harris Operator for Interest Point Detection on 3D Meshes. *The Visual Computer*, 27(11):963–976, November 2011.

[SCOT03] Olga Sorkine, Daniel Cohen-Or, and Sivan Toledo. High-pass Quantization for Mesh Encoding. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, pages 42–51, 2003.

[SOG09] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, pages 1383–1392. Eurographics Association, 2009.

[SSCO08] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Visual Computer*, 24(4):249–259, 2008.

[SSM07] Kaushal Solanki, Anindya Sarkar, and B. S. Manjunath. YASS: Yet another steganographic scheme that resists blind steganalysis. In *Proceedings of Information Hiding*, volume 4567 of *Lecture Notes in Computer Science*, pages 16–31, June 2007.

[Swe96] Wim Sweldens. The Lifting Scheme: A Custom-Design Construction of Biorthogonal Wavelets. *Applied and Computational Harmonic Analysis*, 3(2):186–200, 1996.

[Tag13] Andrea Tagliasacchi. Skeletal representations and applications. *CoRR*, abs/1301.6809, 2013.

[Tau95] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 351–358, 1995.

[TBSS13] Daniel Trick, Waldemar Berchtold, Marcel Schäfer, and Martin Steinebach. 3d watermarking in the context of video games. In *Proceedings of the IEEE 15th International Workshop on Multimedia Signal Processing, 2013*, pages 418–423, September 2013.

[The13] The MathWorks Inc. Optimization toolbox user's guide, 2013. http://www.mathworks.com/access/helpdesk/help/toolbox/optim/.

[TLHK10] J.-S. Tsai, M.-C. Liao, W.-B. Huang, and Y.-H. Kuo. Geodesic distance-based pose-invariant blind watermarking algorithm for three-dimensional triangular mesh model. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 209 –212, September 2010.

[TW98] Grit Thürmer and Charles A. Wüthrich. Computing Vertex Normals from Polygonal Facets. *Journal of Graphics Tools*, 3(1):43–46, March 1998.

[UCB04] F. Uccheddu, M. Corsini, and M. Barni. Wavelet-based blind watermarking of 3d models. In *Proceedings of the 2004 workshop on Multimedia and security*, pages 143–154. ACM, 2004.

[VKS05] Sébastien Valette, Ioannis Kompatsiaris, and Michael G. Strintzis. A polygonal mesh partitioning algorithm based on protrusion conquest for perceptual 3d shape description. In *Workshop towards Semantic Virtual Environments*, pages 68–76, March 2005.

[VL08] Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum*, pages 251–260, 2008.

[VMM99] Jürgen Vollmer, Robert Mencl, and Heinrich Müller. Improved laplacian smoothing of noisy surface meshes. In *Computer Graphics Forum*, pages 131–138, 1999.

[VP04] Sébastien Valette and Rémy Prost. Wavelet-based multiresolution analysis of irregular surface meshes. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):113–122, March 2004.

[VZ01] Luiz Velho and Denis Zorin. 4-8 subdivision. *Computer-Aided Geometric Design*, 18(5):397–427, 2001.

[Wan09] Kai Wang. *Quantization-Based Blind Watermarking of Three-Dimensional Meshes*. PhD thesis, INSA Lyon, 2009.

[Wat07] Watertight models, 2007. http://watertight.ge.imati.cnr.it/, Accessed: 2014-08-06.

[WBSS04] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Imgage Processing*, 13(4):600–612, April 2004.

[WGL+13] B.T. Wittbrodt, A.G. Glover, J. Laureto, G.C. Anzalone, D. Oppliger, J.L. Irwin, and J.M. Pearce. Life-cycle economic analysis of distributed manufacturing with open-source 3-D printers. *Mechatronics*, 23(6):713 – 726, 2013.

[WH09] Y.-P. Wang and S.-M. Hu. A new watermarking method for 3d models based on integral invariants. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):285–294, March 2009.

[WHST01] Jian-Hua Wu, Shi-Min Hu, Jia-Guang Sun, and Chiew-Lan Tai. An Effective Feature-Preserving Mesh Simplification Scheme Based on Face Constriction. In *Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, pages 12–21, October 2001.

[WK05] Jianhua Wu and Leif Kobbelt. Efficient spectral watermarking of large meshes with orthogonal basis functions. In *In The Visual Computers (Pacific Graphics 2005 Proceedings*, pages 8–10, 2005.

[WLBD09] Kai Wang, Ming Luo, Adrian G. Bors, and Florence Denis. Blind and robust mesh watermarking using manifold harmonics. In *Proceedings of the IEEE International Conference on Image Processing*, pages 3657–3660, November 2009.

[WLD+10] Kai Wang, Guillaume Lavoué, Florence Denis, Atilla Baskurt, and Xiyan He. A Benchmark for 3D Mesh Watermarking. In *Proceedings of the 2010 Shape Modeling International Conference*, pages 231–235. IEEE Computer Society, 2010.

[WLDB08a] Kai Wang, Guillaume Lavoué, Florence Denis, and Atilla Baskurt. A comprehensive survey on three-dimensional mesh watermarking. *IEEE Transactions on Multimedia*, 10(8):1513–1527, December 2008.

[WLDB08b] Kai Wang, Guillaume Lavoué, Florence Denis, and Atilla Baskurt. Hierarchical Watermarking of Semiregular Meshes Based on Wavelet Transform. *IEEE Transactions on Information Forensics and Security*, 3(4):620–634, December 2008.

[WLDB11] Kai Wang, Guillaume Lavoué, Florence Denis, and Atilla Baskurt. Robust and blind mesh watermarking based on volume moments. *Computer Graphics*, 35(1):1–19, February 2011.

[WMKG07] Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun. Discrete Laplace Operators: No Free Lunch. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 33–37, 2007.

[WW01] Joe Warren and Henrik Weimer. *Subdivision Methods for Geometric Design: A Constructive Approach*. Morgan Kaufmann, November 2001.

[Yan13] Ying Yang. *Information Analysis for Steganography and Steganalysis in 3D Polygonal Meshes*. PhD thesis, Durham University, UK, October 2013.

[YI10] Ying Yang and Ioannis P. Ivrissimtzis. Polygonal mesh watermarking using Laplacian coordinates. In *Eurographics Symposium on Geometry Processing 2010*, volume 29, 2010.

[YI12] Ying Yang and Ioannis Ivrissimtzis. A Logistic Model for the Degradation of Triangle Mesh Normals. In *Proceedings of the 7th International Conference on Curves and Surfaces*, pages 697–710, 2012.

[YPSZ01] K. Yin, Z. Pan, J. Shi, and D. Zhang. Robust mesh watermarking based on multiresolution processing. *Computer & Graphics*, 25:409–420, June 2001.

[ZC01] Cha Zhang and Tsuhan Chen. Efficient feature extraction for 2D/3D objects in mesh representation. In *Proceedings of the IEEE International Conference on Image Processing*, volume III, pages 935–938, October 2001.

[ZSS97] Denis Zorin, Peter Schröder, and Wim Sweldens. Interactive Multiresolution Mesh Editing. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 259–268, 1997.

[ZTP05] Stefanos Zafeiriou, Anastasios Tefas, and Ioannis Pitas. Blind robust watermarking schemes for copyright protection of 3D mesh objects. *IEEE Transactions on Visualization and Computer Graphics*, 11(5):596–607, September 2005.

[ZVKD10] Hao Zhang, O. Van Kaick, and Richard Dyer. Spectral mesh processing. *Computer Graphics Forum*, 29(6):1865–1894, 2010.