Doctoral Dissertation

Blind watermarking for 3D printed objects by applying small geometric modification on the surface

Arnaud Delmotte

March 12, 2020

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Arnaud Delmotte

Thesis Committee:

Professor Yasuhiro Mukaigawa	(Supervisor)
Professor Yuichi Hayashi	(Co-supervisor)
Associate Professor Takuya Funatomi	(Co-supervisor)
Assistant Professor Hiroyuki Kubo	(Co-supervisor)
Assistant Professor Kenichiro Tanaka	(Co-supervisor)

Blind watermarking for 3D printed objects by applying small geometric modification on the surface*

Arnaud Delmotte

Abstract

Watermarking for 3D printed objects is a relatively new field of research. It increasingly gained attention due to the democratization of 3D printing technology. Nowadays, anyone can own a reliable entry-level 3D printer for a few hundred dollars or get access to one in schools, public libraries, DIY centers, or makerspaces. This wide availability brought concerns about copyright infringements of physical objects that can be easily duplicated, and criminal usages such as reproducing keys from pictures, printing TSA keys, or even printing weapons. Similar to other media, watermarking technology provides ways to investigate these illegal usages.

This thesis focuses on blind watermarking for 3D printed objects by applying small geometric modification on the surface. Watermarking consists of covertly inserting a marker in a 3D printed object, and 'blind watermarking' is a subcategory in which the marker extraction process does not require any information from the original non-watermarked model. Blind watermarking is generally preferred over non-blind because the original model is not publicly available in many application scenarios. In our context, 'small geometric modification on the surface' means that the watermark is embedded by slightly modifying the geometry of the surface of the object, at an amplitude detectable by a machine but imperceptible for the human eye and with no effect on the function of the object.

For 3D printed objects, a watermark can be inserted either on the surface or under the surface. Subsurface methods have the advantage of not producing

*Doctoral Dissertation, Graduate School of Information Science,
Nara Institute of Science and Technology, March 12, 2020.

any surface distortion, but it requires specific equipment and procedure to scan the inside of the object and extract the marker. It becomes unusable in most copyright protection scenarios because a thief would only scan the surface of the object, removing the watermark in the process. For that reason, we focused on surface watermarking. We proposed two new methods.

Our first method uses a histogram of surface norms to encode the watermark bits and 3D moments for the alignment of the object. It can embed around 16-32 bits, works with any 3D printer and scanner as long as the reconstructed mesh is complete, and resists to a reprint from a scanned model.

Our second method uses the printing layer thickness as a 1D signal carrier. It works exclusively with Fused Deposition Modeling (FDM) or similar printing method. It can embed 64 bits, be decoded from a single shot using a consumer 2D document scanner, but it is thus limited to flat surfaces. The low cost of the decoding equipment allows this method to be easily deployed in real life.

The two proposed methods have different advantages, drawbacks, and application scenarios. The first method is more suited for copyright protection scenarios in which the input for decoding would be a standard 3D scan or a counterfeited object. The second method suits better metadata embedding scenarios in which high capacity and fast decoding are required, but resistance to reprint is not. Additionally, both methods can be used for crime investigation, identification of the owner of a stolen object, and potentially many other application scenarios. To avoid degrading the visual quality of the object, we paid attention to minimize the geometric distortions while keeping the watermark sufficiently robust to common printing artifacts. They have been successfully tested with real 3D print using a consumer 3D printer.

Keywords:

blind watermarking, 3D printing, 3D scanning, fused deposition modeling, document scanner

Contents

List of Figures

1 Introduction

3D printing became increasingly popular and accessible recently. Entry-level printers became affordable for the consumer budget and available in some public libraries, schools, DIY centers, makerspaces. It allows anyone to transform their ideas and concepts into prototypes or real products, reducing the cost to enter the market. But on a more pessimist view, it also brought a lot of potential bad usages of the technology. Some criminals use 3D printers to produce objects that help them to commit crimes [44] such as theft by reproducing the keys from pictures, printing fake credit card skimmer [38] or TSA master key [51], or violent crimes by printing untraceable weapons [66]. These objects, illustrated in Fig. 1.1 and Fig. 1.2, are often found on the crime scene but are difficult to trace, making the investigation difficult. Copyright protection is also a hot topic for 3D printing technology. For example, in 2016, a company called 'Just 3D print' got sued for selling printed objects using hundreds of models made by other creators without respecting their Creative Commons license limited to non-commercial applications [37]. What is considered as a copyright infringement depends on the law of the country, but in general, creative objects (*e.g.*, sculpture, figurine) are subject to copyright, whereas purely functional objects (*e.g.*, screw, nut) are not [78] [79] [1]. The international chamber of commerce forecasts that the value of trade in counterfeit and pirated goods could reach 991 Billion dollars by 2022 [55].

3D printing is not the only technology that brought problems when it became widely available. Digital camera allowed people to illegally record movies from the cinema. Document scanner allowed to scan books, mangas, and magazines. Audio recorders allowed to record from concert or duplicate the music from CDs. Internet simplified the diffusion of all these illegal copies, and it became relatively common for consumers to search for illegal copies instead of buying the original

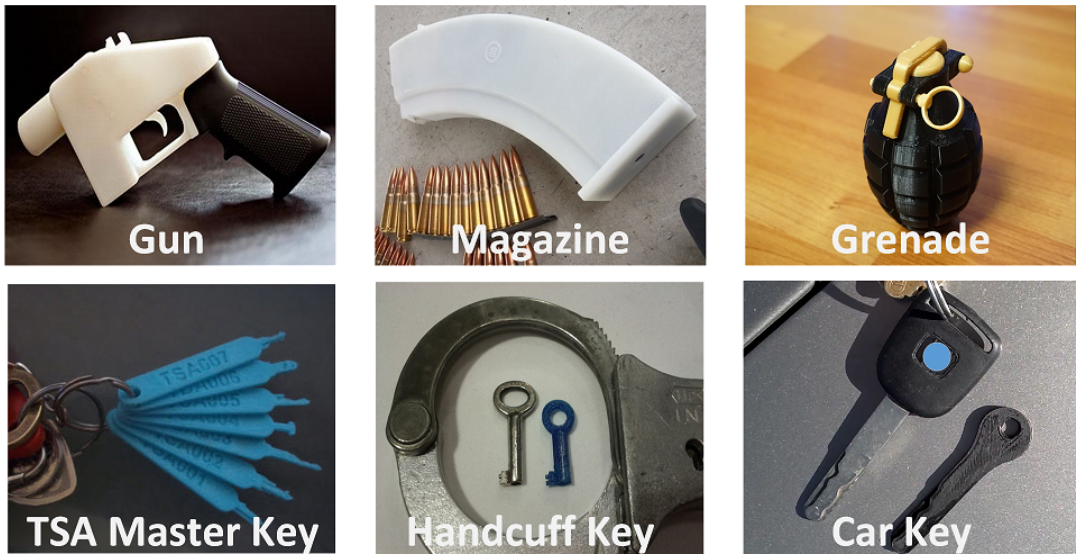


Figure 1.1: Example of 3D printed objects used for criminal activities (Image credit: [44]).

(a) Credit card skimmer

(b) Fake pin code keyboard

Figure 1.2: 3D printed credit card skimmer allowing criminals to duplicate credit card and obtain the code (Image credit: Brian Krebs [38] [44]).

product. To restrain these illegal behaviors, three main technologies emerged: Digital Right Management (DRM), Watermarking, and Fingerprinting. DRMs are an active way to prevent illegal copies of digital content. Only certified reader and writer software and hardware can read and write the content, but also frequently cause trouble to legit users using incompatible or outdated software or hardware. As a result, these users often turn to illegal alternatives after being frustrated by not succeeding to use the product they bought. While DRMs are an effective solution to prevent the digital content from being directly copied in their digital form, they generally lose their protection when they are converted into analog format. For example, a protected and undecipherable audio file becomes copiable if the audio is recorded from the speaker. This weakness is called an analog hole [65] and affect the media requiring to convert their content completely into analog signal such as image, audio, video, and 3D printed object. As exception, interactive media such as video-game, software, and 3D digital model, are less affected by the analog hole because the interactive part can not be recorded from the analog signal. Because DRMs are not applicable for 3D printing due to the analog nature of the 3D printed objects, we will not go further into detail about them in this thesis. Compared to DRM, watermarking technology is a more passive protection. It consists of covertly inserting a marker containing a few bits of data inside the media, allowing, for example, to identify the owner of the media, trace the leaker, or prove the authenticity of the product. In real-life, the insertion of the watermark is generally done by the creator of the media at the export of his work by selecting an option in the software, by using a post-processing tool, or automatically by the software or hardware. Ideally, the watermark should be imperceptible for the users to avoid degrading their experience and prevent the users from removing it easily. Because of the introduction of noise and the loss of the digital structure during the analog conversion, watermarking methods also frequently suffer from the analog hole, but some methods are specifically designed to resist it. Finally, fingerprinting is the most passive technology of the three. Nothing is voluntarily added to the media. It only tries to identify the source by analyzing the naturally occurring artifacts such as the mechanical imprecision on the motion of the head of a document printer, scanner, or 3D printer. It can also be used to uniquely identify copies of the same media

or identify a media after attacks such as compression, scaling, and cropping.

This thesis focuses on watermarking for 3D printed objects and, more specifically, on blind watermarking. 'Blind' means that it does not require any data from the original media to extract the marker. It is more easily deployable in a real situation because anyone can read the watermark and it does not require a trusted authority to extract it. It is also required for the scenarios in which the original model is not available. More specifically, we focused on methods applying a small geometric modification on the surface, that is, the watermark is embedded by slightly modifying the geometry of the surface of the object, at an amplitude detectable by a machine but imperceptible for the human eye and with no effect on the function of the object. In comparison to other approaches such as subsurface modification or ink-based, geometric approaches have the advantage that the embedded signal is preserved if the object is 3D scanned, providing that the resolution is high enough. Most of the copyright protection scenarios suppose that the person doing the scan ignores the existence of the watermark or at least would not follow a specific extraction procedure. By being decodable from standard high-resolution 3D scans, the geometric approaches can cover a broader range of scenarios. We proposed two new methods.

Our first method uses a histogram of surface norms to encode the watermark bits and 3D moments for the alignment of the object. It can embed around 16-32 bits, works with any 3D printer and scanner as long as the reconstructed mesh is complete, and resists to a reprint from a scanned model. This method is an extension of an existing mesh watermarking method, called vertex norm watermarking. The original method was not usable as-is, especially due to its sensibility to resampling, but we applied multiple modifications to make it usable and reliable in a 3D printing context.

Our second method uses the printing layer thickness as a 1D signal carrier. It works exclusively with Fused Deposition Modeling (FDM) or similar printing method. It can embed 64 bits, be decoded from a single shot using a consumer 2D document scanner, but it is thus limited to flat surfaces. The low cost of the decoding equipment allows this method to be easily deployed in real life.

The rest of this thesis is organized as follows. Section 1.1 describes a few real-life applications of the watermarking technology and Section 1.2 describes the current

or potential applications of 3D printing watermarking. Chapter 2 introduces the most commonly used 3D printing technologies. Chapter 3 introduces some theory about the properties and attack resistances required by the 3D printing watermarking methods. Chapter 4 describes the related work and evaluates their properties, attack resistances and potential applications. Chapter 5 describes our first method using a surface norm histogram and 3D moment alignment. Chapter 6 describes our second method using the thickness of the printed layers as 1D carrier signal. Finally, Chapter 7 discusses the results and concludes this thesis.

1.1 Real-life applications of watermarking and fingerprinting

Even if watermarking is not a very well-known technology by the public, it is largely used in some industries. In this section, we introduce some of the most largely deployed applications of these technologies.

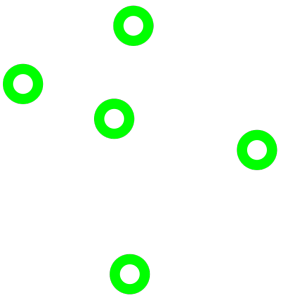
1.1.1 Anti-counterfeit currency protection

Numerous security features are used in the bills to prevent counterfeit production [47]. This includes using special paper, security thread, see-through number, multi-layer printing, color-changing ink, UV/infrared reaction, micro-text, hologram, and a lot of other protection methods. On top of these features that prove the authenticity by being extremely difficult to reproduce, some currency also includes a simple pattern called 'Omron rings' or 'EURion constellation' (Fig. 1.3), that most printers and scanners can detect. When the pattern is detected, the printer or scanner returns an error message and refuse to do the task [53].

1.1.2 Audio/video fingerprinting

Piracy affects a wide range of industries, but movie and music are two of the most affected ones. It is really common to find copyrighted music, series, or movies on public streaming websites. To prevent the illegal upload, the online platforms

(a) EURion Pattern



(b) EURion on a USD 20 bill

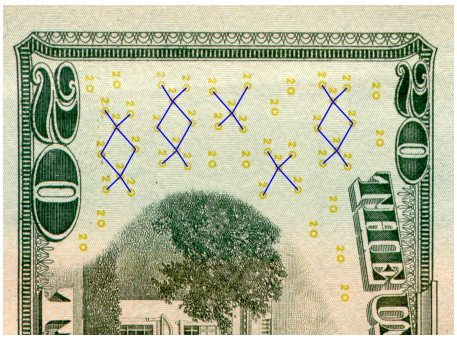


Figure 1.3: EURion pattern, composed of five 1 mm large circles with a specific disposition illustrated in (a), that can be found in some banknotes as anti-copy protection. (Image credit: wikipedia [80])

use methods to automatically identify segments of audio or videos and block the video before it is even published. These techniques are also used for any other illegal videos such as child pornography [49] [41].

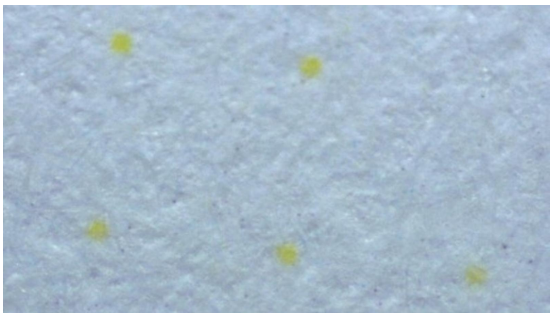
1.1.3 Traitor tracing

Including a different watermark in each copy of a media sold or provided under Non-Disclosure Agreement (NDA) allows tracing the source of a leak. This is also used in the cinema industry to know which cinema and what time was the source of a leak, and in most paper printers, which include a set of yellow dots encoding the printer serial ID and the time of print, that can be used for investigation [14] [21] [6].

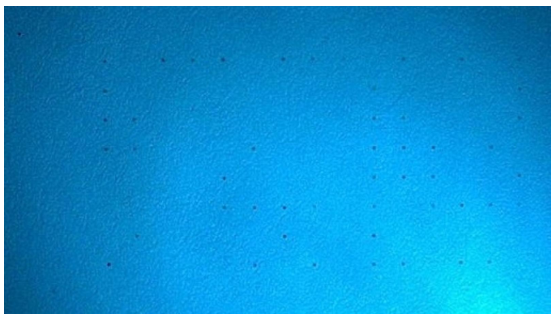
1.1.4 Stolen object identification

Tagging valuable objects helps the police to identify the owner when they find a stolen object [23]. Multiple companies such as Alpha-Dot in UK and DataDot in USA, sell microdots that contain a personal identification number (Fig. 1.5). These dots are around 1 mm in diameter, can be detected using UV-light and can be glued to valuables to protect them from theft. This is officially approved

(a) Magnified yellow dots from a Xerox printout



(b) Document illuminated with blue light



(c) Yellow dots decoded

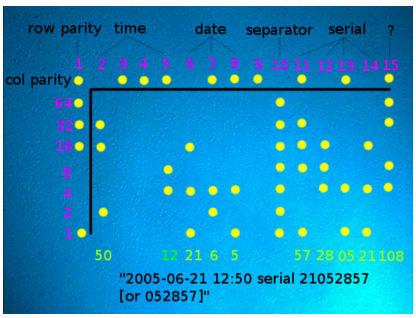


Figure 1.4: Yellow dots inserted automatically by most consumer document printers. They can be more easily detected when the paper is illuminated by blue light. (Image credit: Electronic Frontier Foundation/CC BY 3.0) [21] [6]

by the police, which has access to the detection equipment and the company's database.

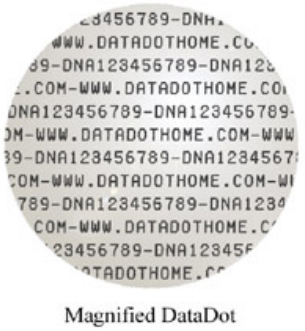


Figure 1.5: 1 mm microdot magnified (Image credit: DataDot USA)

1.2 Applications for 3D printing watermarking and fingerprinting

In this section, we present a non-exhaustive list of application scenarios for 3D printing watermarking and fingerprinting. We summarized the main scenarios proposed in our methods and the related works.

1.2.1 Copyright protection

Copyright protection for 3D printing is a large topic, we introduce here a few scenarios and protection methods.

Proof of ownership

To prevent a user from uploading a 3D scan of a copyrighted object illegally, the copyright holder could insert a watermark containing his ID and public key (Fig. 1.6). This could be used as proof of ownership and be used as legal proof in case of a lawsuit. It could also be automatically verified by the 3D model repository websites to prevent illegal upload.

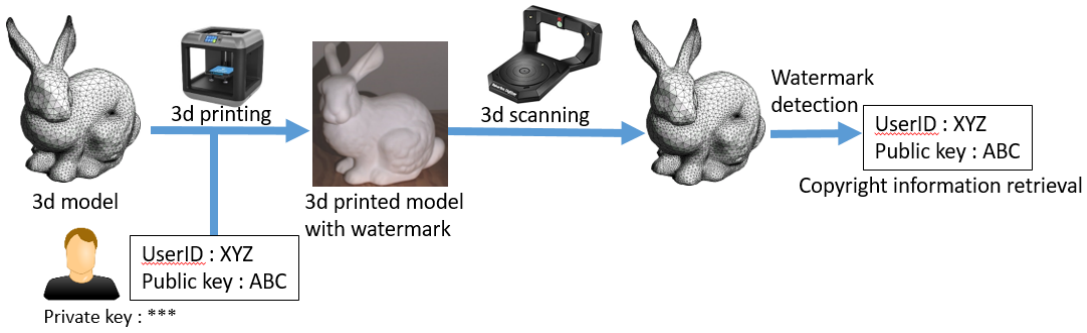


Figure 1.6: Proof of ownership : The owner ID and public key are inserted in the object and can be retrieved from a scan. The owner can then prove his ownership by using his private key.

No-copy flag

Similar to the currency anti-copy protection "EURion pattern" described in the previous chapter, creating an anti-copy watermark common for every 3D printer and scanner could be a simple solution to prevent illegal scan and duplication of the copyrighted 3D objects. At print time, the printer would give the choice to the manufacturer to insert the anti-copy flag in the object to prevent further scan or reprint from his clients (Fig. 1.7). Of course, it is a naive protection, it would only work if every 3D printer and scanner manufacturer agree on a common flag and it would not prevent malicious users from modifying the firmware to disable the protection or build their own system, but it would make their task harder and reduce the piracy from the non-advanced users.

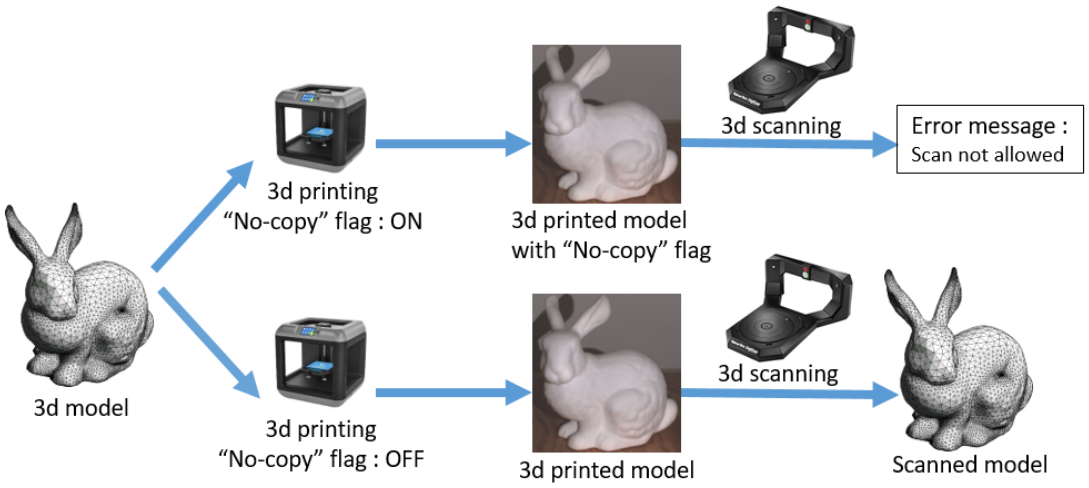


Figure 1.7: No-copy flag : An optional flag that indicates to the 3D scanners and printers that the model is copyrighted.

Traitor tracing

For contexts in which confidential objects are sold or provided under non-disclosure agreement (NDA) to clients, it is important to be able to identify the source of a leak when it happens. By embedding a unique identifier per object, it is possible to do something called traitor tracing (Fig. 1.8). If the confidential model is leaked, the leaker can be identified by the ID embedded in the leaked model.

It may be possible to use a fingerprint instead of a watermark, but it would be more difficult to be robust to scanning or duplication errors done by the leaker.

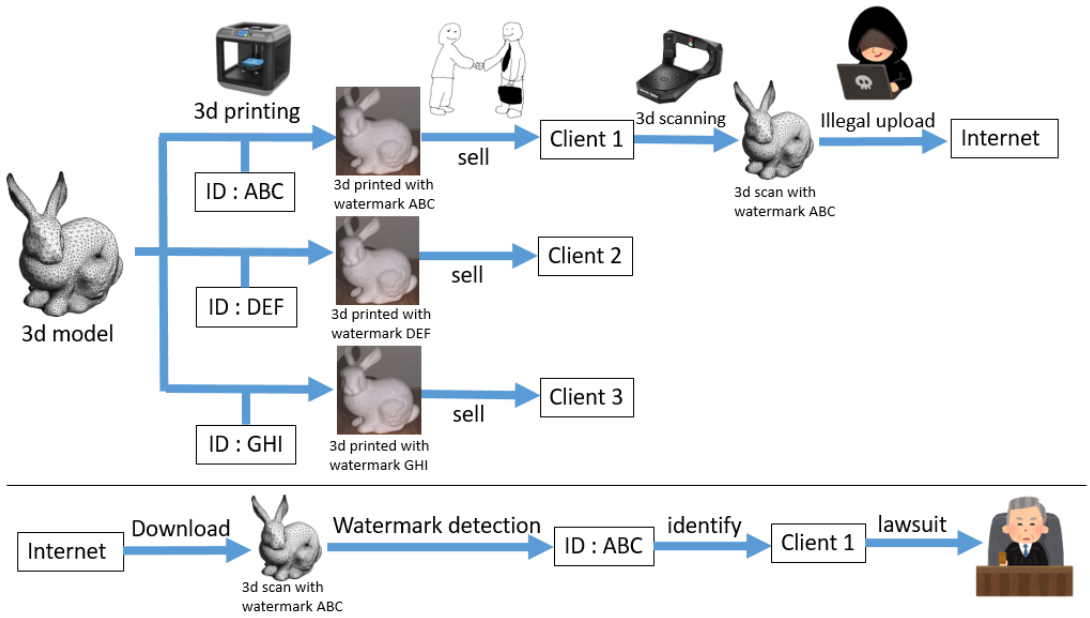


Figure 1.8: Traitor tracing : a unique watermark is inserted in each object before selling it. The leaker can be identified by extracting the watermark from the illegally uploaded model.

Proof of authenticity

For the user that wants to verify if the object he bought is an authentic or a counterfeit one, it is possible to provide proof of authenticity by inserting a unique watermark easy to read but difficult or impossible to reproduce. The manufacturer can maintain a database of the watermark of every object he produced and give access to the user to verify the authenticity of his object (Fig. 1.9). A naive protection can consist of a watermark difficult to reproduce to complicate the counterfeiter task. Methods embedding the watermark below the surface, using variable layer thickness, or infrared or UV ink, are a good example of this kind of protection because they can not be duplicated by a simple scan and print, but knowing the watermarking method used is enough to duplicate the watermark. Advanced scanning methods, such as destructive 3D scanning [52] or hy-

perspectral analysis of the type of plastic or ink used for printing [4], could allow the production of undetectable counterfeit without knowledge of the watermarking method used. For robust protection against counterfeit, physical unclonable function (PUF) is required. Fingerprinting is also usable in this context if it can uniquely identify each print, but it is more difficult to develop because it restricts to apply no voluntary modification the object.

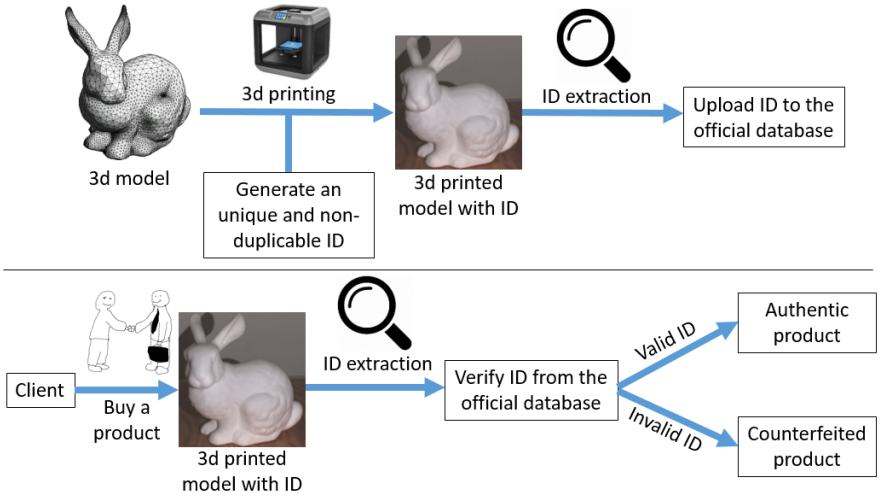


Figure 1.9: Proof of authenticity: A unique and non-duplicable ID is generated during the print, and then uploaded to the official database by the official manufacturer. When a client buy an object, he can verify the authenticity by searching the ID in the database.

1.2.2 Crime investigation

3D printed objects, such as those illustrated in Fig. 1.1 and Fig. 1.2, are used in various crime scenarios, as explained at the beginning of Chap. 1. These objects are often found in the crime scene but are difficult to trace because they do not contain any serial number or other identification features. If the 3D printers automatically inserted a watermark in the 3D printed objects, similar to the yellow dots in the document scanners, it would help the investigators to identify and catch the criminals. The application of this scenario depends on the country legislation and also the knowledge of the criminal about it. The task becomes

much more complicated if the criminal voluntarily tries to remove the watermark by sanding the surface, smooth it with chemicals or apply any other attack to degrade the watermark. For that reason, the watermark should be as robust and imperceptible as possible.

1.2.3 Stolen objects identification

As explained in Section 1.1.4, embedding an identification tag on the 3D object can help the police to identify the owner when a stolen object is found. The same scenario can, of course, be applied with a 3D printer. By embedding a watermark directly during the print, it is automatic, it does not have any additional cost and is more difficult to remove. Fingerprinting would also be usable in this context if robust enough.

1.2.4 Object metadata

Embedding metadata such as the model name, the batch ID, print time, material used, or any other information can be useful in a wide range of applications. We provide a few examples:

Broken part replacement

For any mechanical device, it is common to have a part that breaks after using it for some time. Being able to extract an identification number from the broken part to order or print a replacement could be beneficial for the user.

Robotic grasping

Robotic grasping requires to recognize objects and their pose. Even if the object recognition methods made huge progress recently, it is still a challenging task. For that reason, QR code or similar tags are often used to help the robots for the detection and prevent any errors, but it degrades the visual appearance of the product for humans. Replacing the visible tags by invisible ones could be beneficial.

Smart factories

Smart factories are producing more and more customized parts in their automatized production lines, which requires a proper tracking of each individual part through the manufacturing and packaging process. It is currently done via barcodes or RFID chips [89] but could be extended to other methods.

Augmented reality (AR)

Augmented reality (AR) often uses tags similar to Fig. 1.10 for the interaction with physical objects. This kind of interactive objects is often used to make augmented toys for kids for a recreational or educational purpose [72, 91].

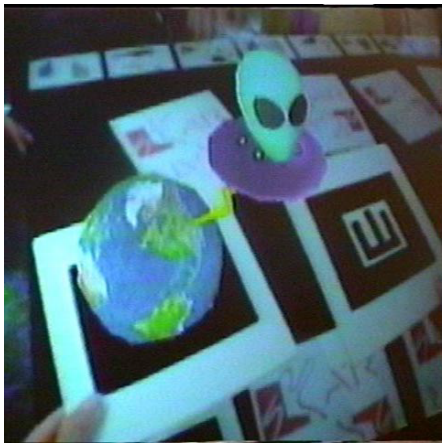


Figure 1.10: Example of AR tag [10].

2 Overview of the 3D printing technologies

3D printing technology, also referred to as additive manufacturing (AM), consists of a wide range of techniques allowing to build layer by layer an object in a large choice of materials. The watermarking methods do not necessarily work with every printing technology and sometimes depends on some properties of the printing process. It also affects the real-life deployment of the watermarking methods: only those compatible with the most common 3D printing technologies can be largely deployed in real life. In this chapter, we introduce the working principle, advantages, drawbacks, and market share of the main technologies used currently. The most commonly used technology is fused deposition modeling (FDM) or fused filament fabrication (FFF), with an overall 67.7% share based on 3dHubs 2018-Q4 trends [2] and almost 80% of the demand for online plastic 3D printing based on 3dHubs 2019-Q1 trends [3]. Its working principle is to build an object by deposing molten plastic layer by layer with a printing head, as illustrated in Fig. 2.1. It is a relatively old technique, it was patented in 1992, but the boom of the technology started around 2009 at the expiration of the patent, and pushed by the development of the open-source project "RepRap". The cost of FDM printers dramatically dropped, starting now at just a few hundred dollars, and the reliability and quality of these cheap printers increased over the years.

Other popular techniques for plastic printing include stereolithography (SLA), digital light processing (DLP), selective laser sintering (SLS), multi-jet fusion (MJF), material jetting (MJ, PolyJet, MultiJet). SLA and DLP use UV light to cure (solidify) a liquid photopolymer resin (Fig. 2.2). When a layer is finished, the building plate is moved and the next layer is processed. SLA uses a laser and can trace any shape at high precision, whereas DLP uses a projector and

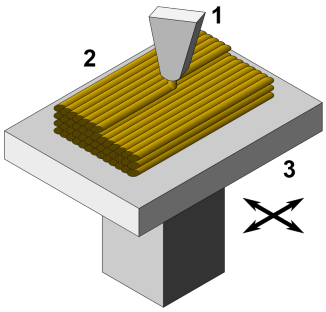


Figure 2.1: FDM/FFF printing principle.
(1) Printing head/extruder. (2) Layers of deposited plastic. (3) Moving printing bed.

is limited by the resolution of the projector used and tends to produce artifacts due to the pixelation. However, DLP has the advantage of being much faster than SLA because it can do one layer at a time instead of having to trace all the regions to cure. SLA and DLP combined are the second most used 3D printing technology, they reached a 15.5% share on 3dHubs 2018-Q4 trends [2]. Their printing quality is generally higher than FDM but the resins are more expensive, have a lower variety of material properties, and the printed objects have a lower mechanical strength. Another disadvantage is that the resins are generally toxic until cured and washed. It requires much more precaution for the utilization during the preparation and post-processing than FDM.

SLS also uses a laser but fuses a powder instead of curing a liquid, and requires a roller to apply a new layer of powder instead of just moving the plate (Fig. 2.3(a)). This technology is more oriented for the prosumers and the industry, with the cheapest printers around USD 6000, making it less accessible than FDM, SLA, and DLP. It has a similar quality than SLA, but a larger variety of materials, better mechanical strength, and the post-processing is easier and does not require the manipulation of toxic resin. MJF is also based on powder melting, but an inkjet head deposit drops of light-absorbing ink over the regions that need to be solidified. An infrared heater is then used to fuse the regions covered by the ink, a new layer of powder is deposed, and the process starts again for the next layer (Fig. 2.3(b)). This technology is totally oriented for the industry, with printers

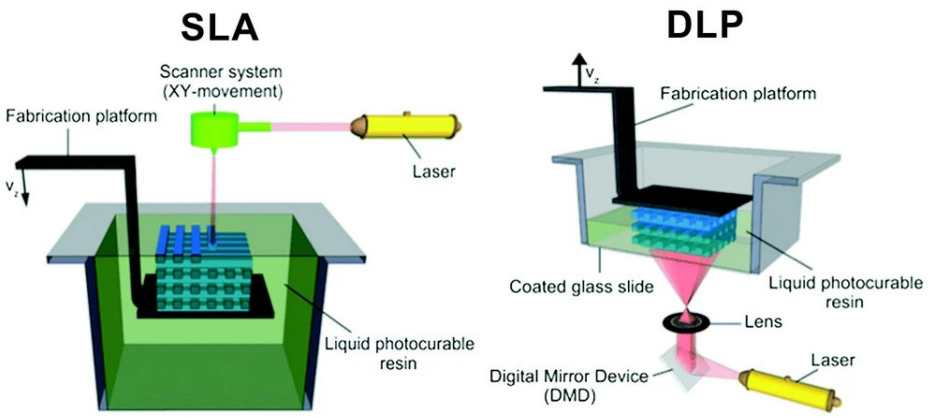


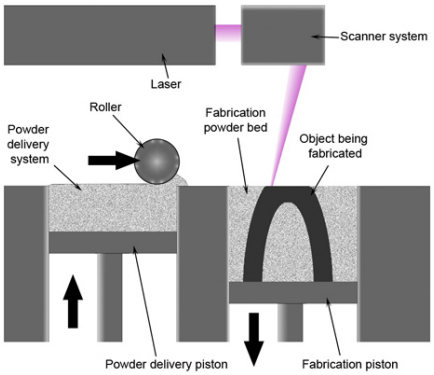
Figure 2.2: SLA and DLP printing principle. (Image credit: Royal Society of Chemistry)

costing over USD 100000. It allows a high-speed and high accuracy print, with simple post-processing, and precise control of the color, translucency, texture and material properties on each part of the object. Combined, SLS and MJF reached an 11.1% share on 3dHubs 2018-Q4 trends [2].

MJ, also known as PolyJet or MultiJet, is similar to SLA because it cures resin with UV light, but instead of storing a full tank of resin, the drops of resin are sprayed by the inkjet print heads. Getting rid of the tank of uncured resin makes it much less hazardous than SLA and DLP. MJ is also faster than SLA because it prints one line at a time, spraying simultaneously hundreds of tiny droplets along the X-axis(Fig. 2.4). It allows high-accuracy, multi-material, multi-color printing. The price of the printer is similar to SLS, starting around USD 6000, which brings it to the prosumer and industrial category. It reached a 3.3% share on 3dHubs 2018-Q4 trends [2].

A lot of other techniques exist, especially for other materials such as metal, but we will not enter too much into detail because they are much less commonly used.

(a) SLS



(b) MJF

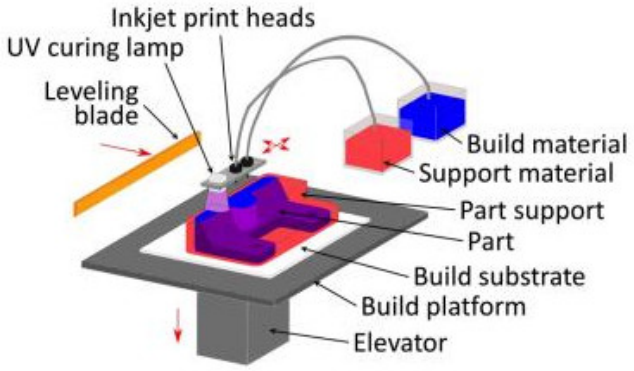


Figure 2.3: SLS and MJF printing principle. (Image credit: SLS: [18] CC BY-SA 3.0, MJF: [9])

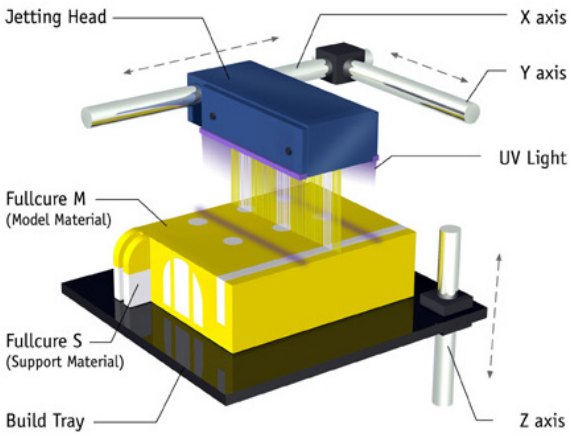


Figure 2.4: PolyJet printing principle. (Image credit: Objet Ltd (Stratasys))

3 Theory of 3D printing watermarking : properties and attacks

Watermarking for 3D printed objects is a broad subject. Each application scenario requires a specific set of properties of the watermarking methods and resistance to a set of attacks that may happen under the specific scenario. In this chapter, we made a list of the main properties and attacks, and evaluated which ones were required for each scenario. The terms defined in this chapter will be used again in the following chapters to address our methods and the related works.

3.1 Properties of 3D printing watermarking methods

In this section, we describe the properties that characterize a 3D printing watermarking method. We start by describing a few properties that are not specific to 3D printing watermarking but common for all watermarking fields :

- Blind/Semi-blind/Non-blind :
  Blind means that the method does not require any data from the original non-watermarked model. Non-blind, on the contrary, require the original model to extract the watermark. Semi-blind is in-between, it requires some information from the original model but not the full mesh.

- Robust/Fragile :
  The robustness of a watermark is always evaluated with some attacks. If a watermarking method can resist to a specific attack, it is called 'robust' to

this attack.
A fragile watermark, on the contrary, is designed to be destroyed at the slightest modification. It is generally used for tamper detection.

- Invisibility :
  Invisibility measures how perceivable the watermark is to the human eye. In general, it is expected that the watermark is barely visible or completely invisible to the human eye, but easily detectable by the machine.

- Capacity :
  Capacity indicates the quantity of data that can be embedded.

Watermarking methods always have to make a trade-off between robustness, invisibility, and capacity. Increasing one property is generally done at the detriment of the two others, as illustrated in Fig. 3.1. For example, it is possible to increase the robustness by increasing the redundancy and introducing error-correcting codes, but that reduces the capacity. It is also possible to increase the robustness by increasing the signal amplitude, but that decreases the invisibility.

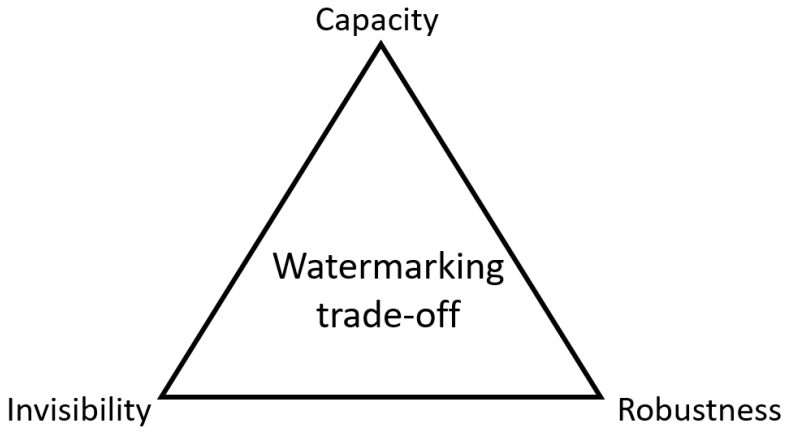


Figure 3.1: Trade-off between robustness, invisibility and capacity in watermarking methods

There are also some properties that are specific to the 3D printing watermarking field :

- One-shot extraction :
  One-shot extraction means that the watermark can be extracted by doing

only one capture from a single pose. It is important when the scenario requires a fast extraction or when the context does not allow to turn all around the object to take multiple captures.

- Extraction from standard 3D scan :
  It means that a complete 3D scan, done with any 3D scanner without specific constraint or procedure to follow, is enough to extract the watermark. It is important in most of the copyright protection scenarios in which the thief would not follow a specific procedure but instead just do a scan with his favorite method and only diffuse the resulting scan.

- Flat, curved or irregular surfaces :
  Some methods only support flat or slightly curved surfaces, while others are totally independent of the shape. Fig.3.2 illustrates the different types of surface. In this example, Fig. 3.2(a) is the most regular type of model, very common in CAD applications and the easiest to watermark. Figure 3.2(b) is intermediary, and Fig.3.2(c) is the most irregular type and most difficult to watermark.

(a) (b) (c)

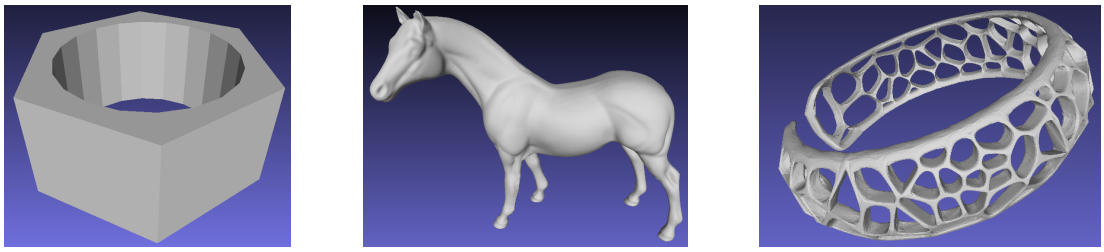


Figure 3.2: Example model containing different types of surface. (a) Flat and curved surfaces (b) Curved and irregular surfaces. (c) Irregular surfaces.

- Supported 3D printer technologies:
  Some methods are independent of the printing technology used, while others are specific to some technologies. The most common technologies have been introduced in Chapter 2.

- Extraction equipment :
  The extraction equipment describes what is required to extract the watermark. Methods with inexpensive and readily available equipment are generally preferred for real-life use, but it is really application-specific.

3.2 Common attacks on the watermark of 3D printed object

In this section, we describe the common attacks that occur in the application scenarios and could affect the robustness of the watermark.

- Translation, rotation :
  Common transformations that occur naturally due to the print-scan process and that all methods need to be resistant to.

- Uniform scaling :
  Modifying the size of the object is one of the most common transformations applied to the model before the print. It is also commonly applied to the scanned model.

- Resampling, remeshing :
  The printing process transforms a mesh into a physical object. All the digital information, such as the vertex positions and connectivity, are lost and replaced by new ones generated during the scanning process. Any method used for 3D printing requires to be either invariant to the sampling and meshing, or have a way to recover it after the print-scan.

- Non-uniform scaling :
  This attack is less common because it deforms the object. It can happen if the printer or scanner axes are poorly calibrated, if there are any plastic printing problems such as retraction or warping. This may also be a voluntary attack if a user knows that a model is watermarked and wants to get rid of it.

- Cropping :
  This attack happens when an object is broken after, for example, dropping

on the ground or receiving too high mechanical constraints. It can also happen if the 3D scan is incomplete due to some areas being difficult to scan or after post-processing by the user.

- Local distortion/printing artifacts :
  A good 3D printer should have low distortion, but it still happens sometimes especially with difficult parts that require a lot of overhang printing (Fig. 3.3).

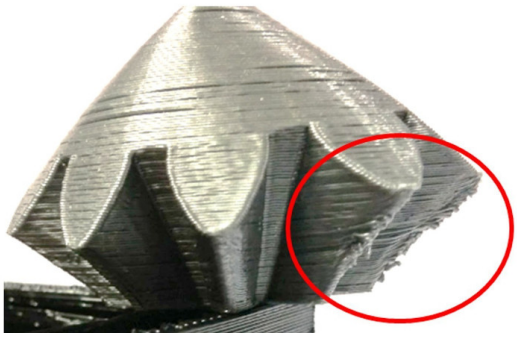


Figure 3.3: Overhang artifacts (Image credit: cults3d.com)

- Smoothing :
  Smoothing is a post-processing operation that aims to make the surface more beautiful. It can be damaging for some watermarking methods because it modifies the surface and removes rough details.

- Reprint :
  Reprint consists of 3D scanning the watermarked model and then print the scanned model. It also includes molding and casting the model. It typically happens when someone produces counterfeit objects. This is a relatively strong attack because it doubles the errors caused by the printing and scanning process.

3.3 Properties and attacks for each application scenario

We summarized the properties and attack resistances required by each scenario (Sec. 1.2) in Table. 3.1 and included the explanations for each property.

We observe that the required properties change quite a lot, depending on the scenario. Blind and invisibility are required or preferred in most of the scenarios, but some scenarios also require other properties such as higher capacity. The possibility to extract from a standard 3D scan is important when the only input available is a 3D scan done by the user as it is the case in the copyright protection scenarios, and the one-shot detection is important when the extraction speed is primordial. Proof of authenticity is a special case among the proposed scenarios because it requires a fragile watermark that should not be duplicable or tamperable. All the other scenarios require the opposite, with as much resistance as possible to the attacks.

To develop our two methods, we focused on producing blind methods with a good trade-off between invisibility, capacity, and robustness. These are the most common and most often required properties. The watermark should support at least 16 bits to fit most scenarios, be robust to the print-scan process, and be imperceptible to the human eye. On one of our methods, we additionally focused on one-shot extraction, whereas on the other, we focused on standard 3D scan extraction and reprint resistance. Our methods are analyzed and compared more in detail with the related work in Sec. 4.4.

Table 3.1: List of the properties and attack resistances required by each scenario.

		Properties						Attacks			
		Blind	Invisibility	Capacity	One-shot extraction	Standard 3D scan	Curved/irregular surface	Non-uniform scaling	Cropping	Smoothing	Reprint
Copyright protection	Proof of ownership	∼[1]	✓[7]	1 bit [9]	✗[11]	✓[13]	∼[15]	∼[16]	∼[16]	∼[16]	∼[16]
	Proof of authenticity	✗[2]	∼[8]	1 bit [9]	✗[11]	✗[14]	∼[15]	✗[16]	✗[16]	✗[17]	✗[19]
	Traitor tracing	∼[1]	✓[7]	8-32 bits [10]	✗[11]	✓[13]	∼[15]	∼[16]	∼[16]	∼[16]	∼[16]
	No-copy flag	✓[3]	✓[7]	1 bit [9]	✗[11]	✓[13]	∼[15]	∼[16]	∼[16]	∼[16]	∼[16]
	Crime investigation	✓[4]	✓[7]	32 bits [10]	✗[11]	✗[14]	∼[15]	∼[16]	∼[16]	∼[16]	∼[16]
	Stolen object	∼[5]	✓[7]	16-32 bits [10]	✗[11]	✗[14]	∼[15]	∼[16]	∼[16]	∼[16]	✗[20]
Metadata	Broken part replacement	✓[6]	∼[8]	8-16 bits [10]	✗[11]	✗[14]	∼[15]	✓[18]	✓[18]	∼[16]	✗[20]
	Robotic grasping	✓[6]	∼[8]	8 bits [10]	✓[12]	✗[14]	∼[15]	∼[16]	∼[16]	∼[16]	✗[20]
	Smart factory	✓[6]	∼[8]	>8 bits [10]	✓[12]	✗[14]	∼[15]	∼[16]	∼[16]	∼[16]	✗[20]
	Augmented reality	✓[6]	∼[8]	8 bits [10]	✓[12]	✗[14]	∼[15]	∼[16]	∼[16]	∼[16]	✗[20]

✓(green) means that the property is required, ∼ (orange) means that it is not mandatory but better to have, and ✗(red) means that it has no importance for the context.

Justifications :

1: The person extracting the watermark has access to the original model, but providing the proof requires to release the model or requires a trusted authority. It is easier if the detection is blind. 2: Accessing a database to verify the authenticity is not a problem in this scenario. 3: Requesting to query an online database for each print would not be practical, most printers are not connected to internet. 4: The original model is not necessarily public or available. 5: Not mandatory, just easier to avoid checking a database. 6: The goal of the watermark is to identify the object. 7: The user would try to remove it if visible. 8: Not mandatory, but better for visual quality. 9: Just requires a yes/no detection of the signature of the model creator / copyright holder. 10: Requires one ID number per user/object. 11: Detection speed is not really important in this scenario. 12: Detection speed is very important in this scenario. 13: The thief that illegally scanned the object will only upload a standard 3D scan of the surface, he would not follow any other procedure. 14: The person that wants to extract the watermark has physical access to the printed object, and can follow any procedure given to him. 15: Less restriction on the type of object supported is better. 16 : Not required, but robustness to more attacks is good. 17 : Proof of authenticity is a fragile watermark allowing to detect counterfeit or damaged product, it should not resist modifications. 18: Once damaged, the object will be deformed, but the watermark must resist. 19: The goal of proof of authority is to detect counterfeit. 20: No reason to reprint in this scenario.

4 Related works

Watermarking has been studied intensively for various media such as image [13, 62, 87, 90], audio [35, 54], video [61, 67] and 3D mesh [33, 36, 48, 74], but is still a relatively new field for 3D printing. Hou *et al.* [30] did a survey about the intellectual property protection issues and solutions in the 3D printing environment. We subdivided this chapter into 4 sections : In section 4.1, we list the related works for mesh watermarking and explain why these methods can not be directly transposed to 3D printing watermarking. In Sec. 4.2 and Sec. 4.3, we list the 3D printing watermarking embedding below the surface and on the surface, respectively. And finally, in Sec. 4.4, we compare the properties, attack resistances and applicability of each 3D printing watermarking method introduced in Sec. 4.2 and Sec. 4.3.

4.1 3D mesh watermarking

3D mesh and 3D printing watermarking are related fields, but transposing methods from mesh to 3D printing domain is not straightforward. We selected a few methods to introduce the different approaches used in the 3D mesh watermarking field :

Mesh density pattern embedding [58] modifies the local triangle density to generate patterns visible only in wireframe mode, as shown in Fig. 4.1. Triangle similarity quadruple (TSQ) [57, 58] embeds the watermark data into sets of four adjacent triangles (Fig. 4.2(a)). Within a set of adjacent triangles (Fig. 4.2(b)), pairs of dimensionless quantities such as $\{e_{14}/e_{24}, h_4/e_{12}\}$ are modified by moving slightly the vertices to embed the watermark.

Cayre *et al.* [12] proposed a high capacity blind watermarking method. They use the projection of a vertex on the opposite edge of the triangle to embed the

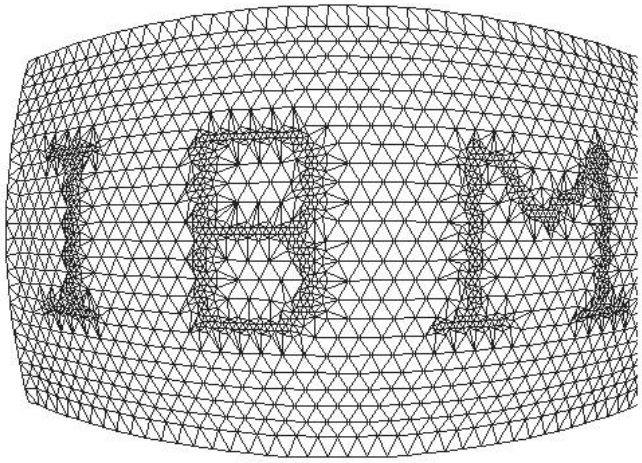


Figure 4.1: Mesh density pattern embedding [58] : the triangle density is modified to make a pattern visible in wireframe mode.

watermark, as illustrated in Fig. 4.3. They choose the first triangle based on a geometric criterion and then spread to the adjacent triangles by following a pseudo-random sequence controlled by a secret key.

Ohbuchi *et al.* [56] introduced a spectral watermarking method. They subdivide the mesh into a few patches and apply spread spectrum watermarking on the patches spectral coefficients. The spectral analysis is done by computing a Kirchoff matrix K of size $n \times n$:

$$K = D - A \tag{4.1}$$

where n is the number of vertices, D is a diagonal matrix of the valence of the vertices, and A is an adjacency matrix of the vertices. The Kirchoff matrix K is then decomposed into eigenvalues and eigenvectors and sorted by eigenvalue.

Vertex norm watermarking [15, 16] uses a histogram of vertex norms, that is, the distance between the vertex and the center of gravity. The histogram is subdivided into multiple bins that store one bit each. The bits are encoded by modifying the distribution within the bin, as illustrated in Fig. 4.4.

A more in-depth survey of the 3D mesh watermarking techniques can be found in Wang *et al.* [74], Medimegh *et al.* [48]. As shown in the two surveys, most mesh watermarking methods do not resist connectivity attacks such as remeshing, simplification, or subdivision of the mesh. In 3D printing watermarking, the

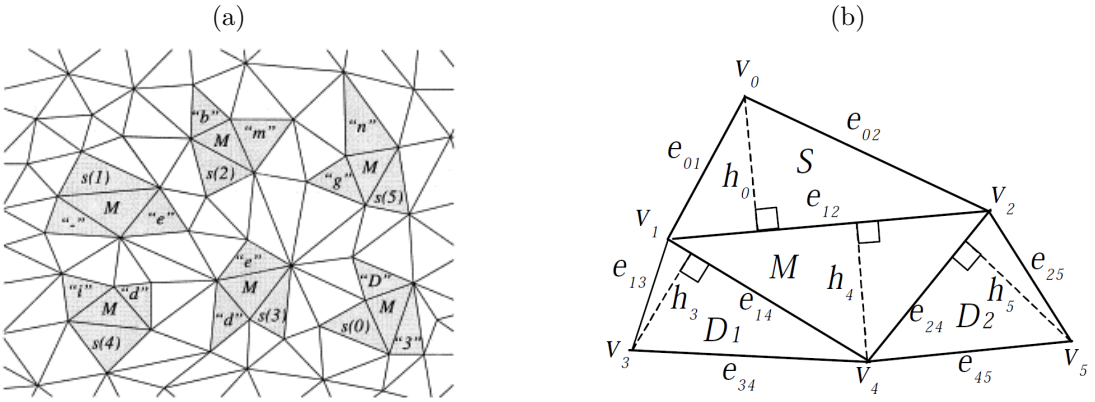


Figure 4.2: Triangle similarity quadruple (TSQ) [57, 58]. (a) six sets of four adjacent triangles used to encode the watermark. The sets do not share vertices to prevent interference. (b) an annotated set of four adjacent triangles.

printing and scanning process produces a complete resampling of the object, and thus most proposed methods are not applicable for this context. As explained in Wang *et al.* [74], some methods can become resistant to resampling by aligning the watermarked model with the original mesh and recovering the mesh structure at the extraction step, but that makes the methods non-blind, which prevents to use them in a lot of scenarios as explained in Chapter. 3. Yamazaki *et al.* [84] used such approach. They aligned the scanned object to the original mesh and used a spectral-domain watermarking method.

Figure 4.3: Cayre *et al.* [12]. One bit is encoded by moving the vertex C to C_0 or C_1 position such that its projection on the edge AB is in a "0" or "1" region. (a) two regions, more robust but bigger deformation required. (b) four regions, less robust but smaller deformation required.

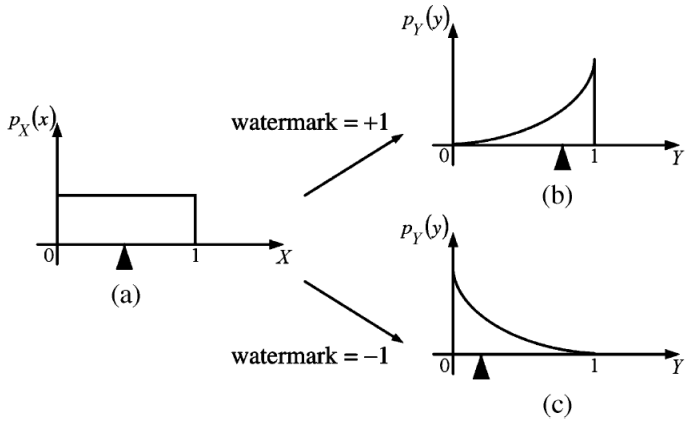


Figure 4.4: Vertex norm watermarking [15, 16]. (a) distribution within a bin before encoding the value (b) encoding a "1" bit by concentrating the distribution on the right side of the bin. (c) encoding a "0" bit by concentrating the distribution on the left side of the bin.

4.2 3D printing watermarking using subsurface modification

Modifying the inside of the object instead of the surface has the advantage of producing no distortion on the shape of the object but generally requires more complicated equipment and procedure for the extraction. Some patents [76] [77] [73] propose to add an RFID chip inside the object. This has the advantage of being cheap, reliable for the decoding, but it requires manual insertion during the print or advanced 3D printer able to insert the chip or print conductive materials [64].

Ivanova *et al.* [22,32] propose to embed quantum dots inside objects created by PolyJet material jetting. Quantum dots are nanoparticles that absorb ultraviolet light and emit light in the visible spectrum. Their stochastic arrangement at the microscale provides the randomness necessary to serve as the key element of a Physical Unclonable Function (PUF), that can be used as proof of authenticity of the object. Even if it is not exactly a watermark because we have no control over the value of the embedded signal, it can be used in a lot of similar applications to the other watermarking methods.

Acoustic Voxels [42] and Blowhole [71] propose to modify the inside structure of an object to insert acoustic cavities, as illustrated in Fig. 4.5. Acoustic Voxels allows to identify an object by the sound emitted by tapping with the hand on it while Blowhole allows to identify by the sound emitted by blowing in the hole. Acoustic Voxels and Blowhole can distinguish 16 and 9 sounds, respectively.

Many researchers proposed to insert a regular grid of cavities below the surface of the object and use computer vision methods to detect them through the surface. Li *et al.*proposed aircode, illustrated in Fig. 4.6, that uses a camera, projector, and polarizers to eliminate the global-component of the light and see through the surface to detect the cavities. Willis *et al.* [81] developed a similar method using a terahertz camera, Okada *et al.* [59] and Suzuki *et al.* [68, 70] used a thermal camera, and Suzuki *et al.* [69] used near-infrared light and camera.

(a) (b)

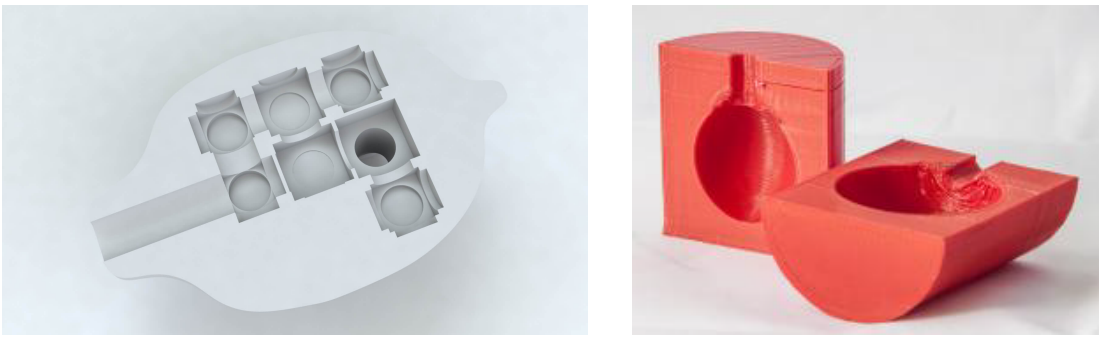


Figure 4.5: (a) Acoustic Voxels [42] inside cavities. (b) Blowhole [71] inside cavity

4.3 3D printing watermarking using surface modification

A straightforward approach consists of adding a visible tag on the surface. Adobe [39] patented a 3D barcode pattern, illustrated in Fig. 4.7(a). It is composed of a regular grid in which the surface of each cell is oriented in one of the four directions, allowing to embed two bits per cell. Zhang *et al.* [88] proposed a regular grid of bumps on flat surfaces (Fig. 4.7). They focused on extracting the watermark from a single image using an RGB camera and a convolutional neural network (CNN).

Another approach consists of attaching microdots at some points of the surface [23]. Microdots have already been introduced in Sec. 1.1 and are not exactly a 3D printing watermarking method because the dots are engraved with laser and glued on the object after print. They are an efficient method to tag an ID on any valuable object.

Harrison *et al.* [26] proposed to insert an acoustic barcode on the surface of the object. It consists of a structured pattern of physical notches, illustrated in Fig. 4.8. These notches produce a complex sound when swiped with a fingernail, marker, or other tools. The sound can be used to identify the object using a single inexpensive contact microphone.

HP [50],Rize [85] developed advanced 3D printers with ink-jetting capabilities that allow printing QR code either with visible or UV-reactive ink. Maia *et*

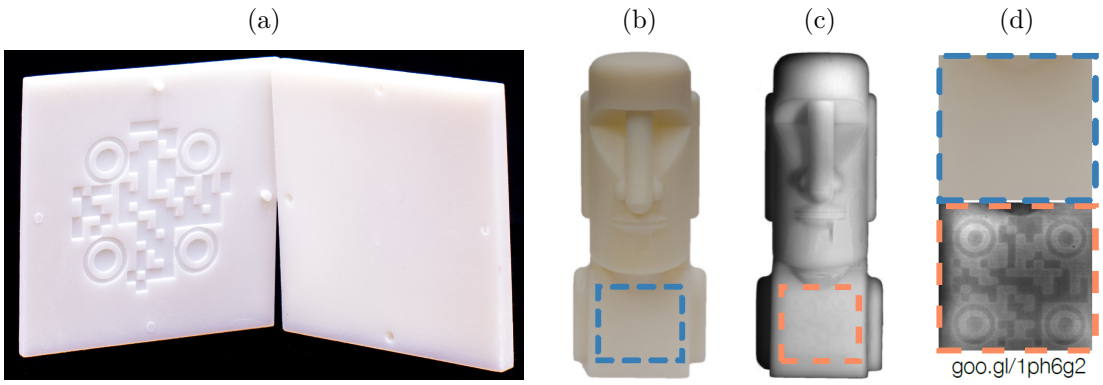


Figure 4.6: Aircode [43], air pocket behind the surface of the model. (a) A square piece cross-section. (b) The tag is invisible under standard lighting conditions. (c) Same object as (b) but after eliminating the global-component of the light. (d) Zoom on the selected regions from image (b) and (c), and encoded URL. (Model credit: "Moai" by gravity-isweak / CC BY 3.0 / modified from original)

*al.*proposed LayerCode [46], a 1D barcode to print across all the surface of the model. Compared to QR code that requires a planar surface, 1D barcode has the advantage of being usable on nearly any shape, as shown in Fig. 4.9. The barcode can be printed with dual-color, near-infrared ink, or variable layer height. For the variable layer height, instead of two different colors, two different layer thicknesses are used, as illustrated in Fig. 4.10. Similar to LayerCode, we also developed a method modifying the layer thickness (Chap. 6), but we modify it locally and continuously instead of applying the same thickness for a whole layer. We can embed more data per layer and also produce less distortion than LayerCode. Our decoding method uses a 2D document scanner, which is advantageous for real-life applications because the decoding equipment is low cost and readily available but limits the decoding to flat surfaces. It should be extendable to curved surfaces using a high-resolution depth camera.

Hou *et al.* [28] proposed a method that subdivides the mesh into slices along the printing axis and modifies the face normal distribution of each slice using a circular shift structure (Fig. 4.11(a)). Each slice uses the same pattern for the modification of the normal distribution but shifted by an angle. This angle is

(a) (b)

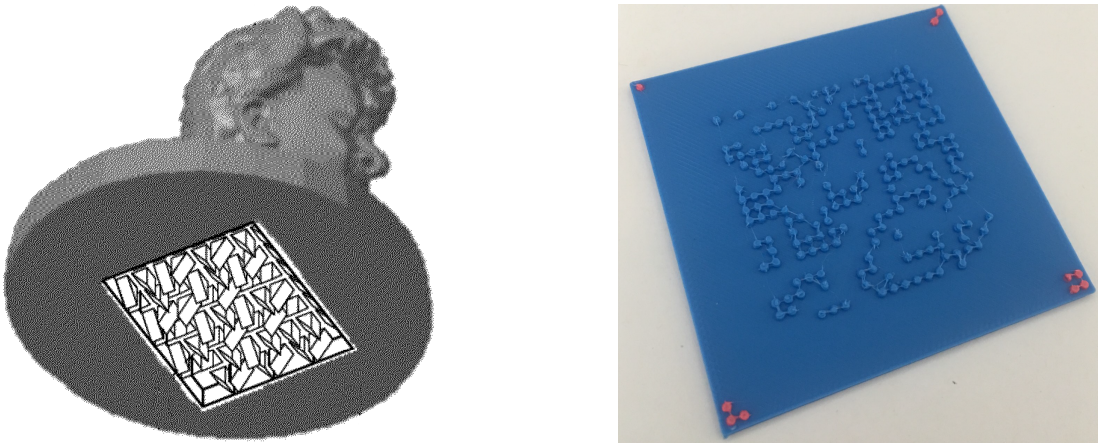


Figure 4.7: Visible geometric tags. (a) Adobe 3D barcode [39] (b) Zhang *et al.* [88] regular grid of bumps.

used to store the data of the watermark (Fig. 4.11(b)). The extraction procedure requires the base axis that is obtained by aligning with the original mesh, making the method non-blind. In a following research, Hou *et al.* [29] proposed to use the printing artifacts to retrieve the printing axis (Fig. 4.12), allowing a blind extraction. They also used spread-spectrum watermarking instead of the circular shift coding structure. Yamamoto *et al.* [83] proposed a method aligning the object differently. They used the pseudo rotation axis of the object. It corresponds to the axis of revolution or an approximation of it (Fig. 4.13) if the object is not a solid of revolution. We proposed a method (Chap. 5) using 3D moments for the alignment and a surface norm histogram to embed the watermark.

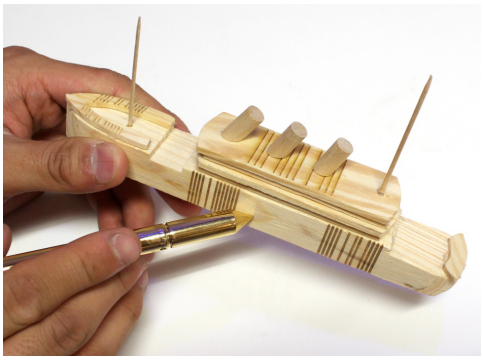


Figure 4.8: Acoustic barcode [26]

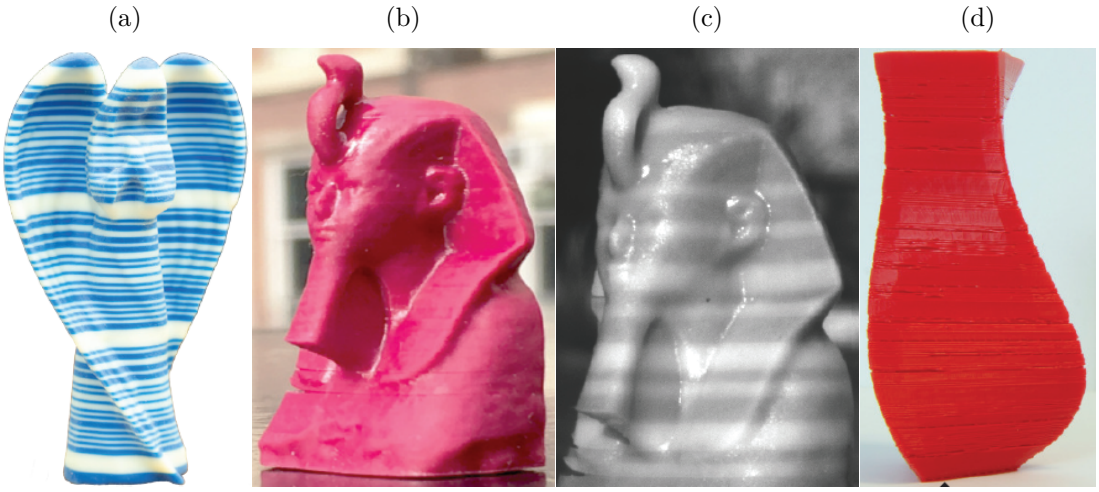


Figure 4.9: LayerCode [46]. (a) printed with two colors. (b) Printed with near-infrared ink, viewed in sunlight. (c) Same object as (b), but captured with a NIR filter in front of the camera. (d) Pattern encoded by changing the thickness of some layers.

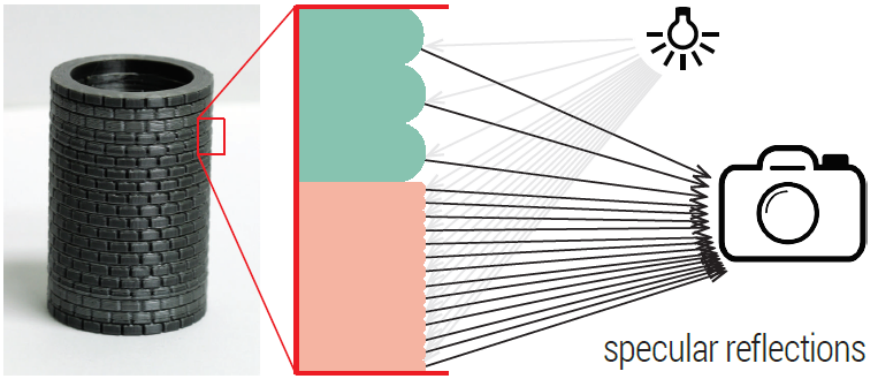


Figure 4.10: LayerCode [46]. Using two layer thicknesses to embed data during the print. The difference of specular reflection allows to decode it with a camera.

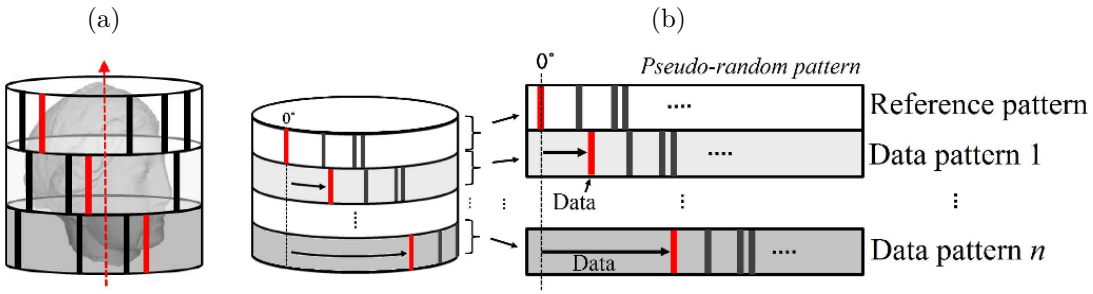


Figure 4.11: Hou *et al.* [28]. (a) The base axis (red arrow), the cylindrical slices and the normal histogram modification pattern. (b) The circular shift structure composed of rotating disks. The data is stored in the angle between the disks.

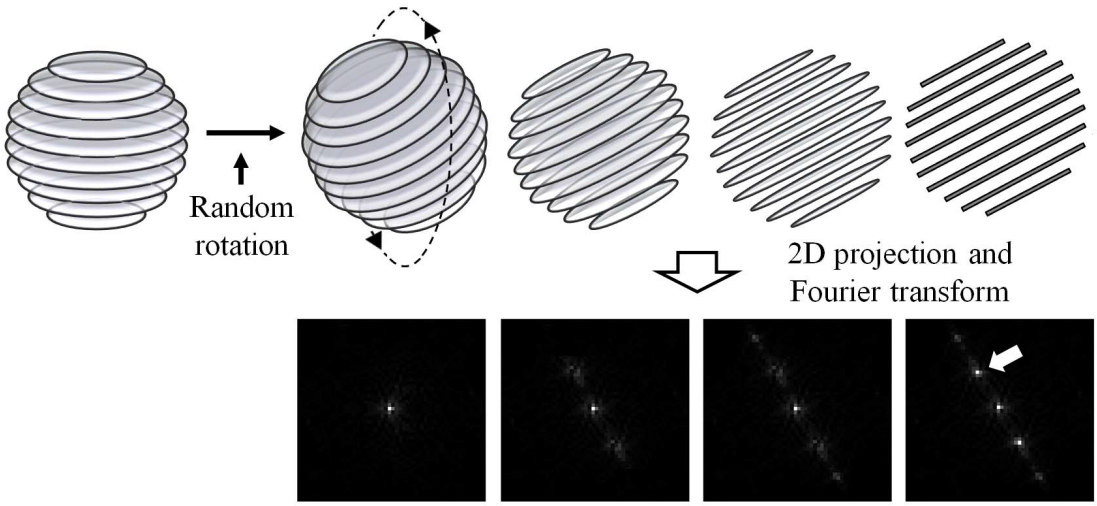


Figure 4.12: Hou *et al.* [29]. The printing axis can be retrieved by rotating the object until the layering artifacts from the front and back face of the object get aligned under an orthogonal projection. The optimal alignment is the rotation that provides the highest response on the Fourier transform, excluding the low frequency.

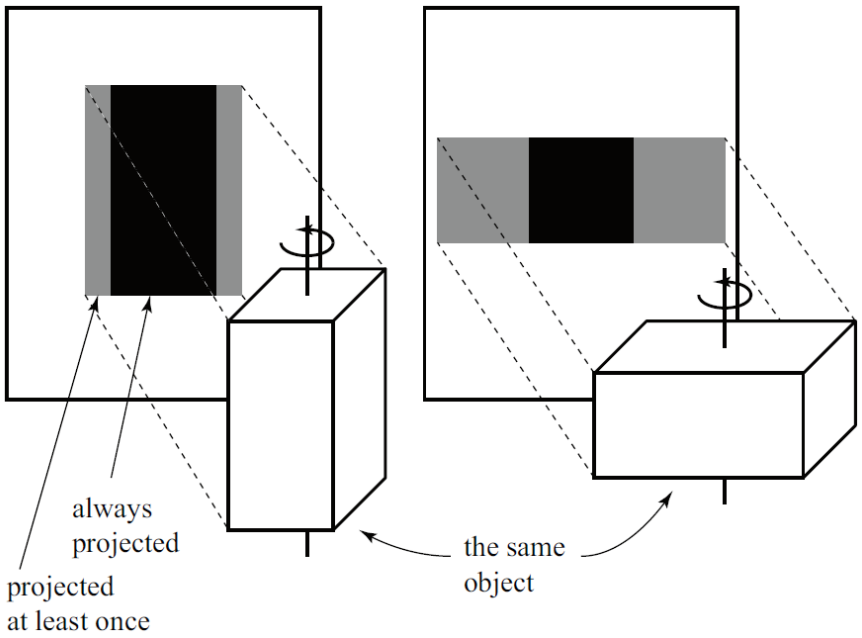


Figure 4.13: Yamamoto *et al.* [83]. The area of the orthogonal projection of a 3D object depends on its orientation. Depending on the axis of rotation, the projected area can have large change (right image), small change (left image), or not at all if the axis is an axis of revolution. The pseudo rotation axis is defined as the axis with the lowest ratio between the maximal projected area and minimal projected area.

4.4 Comparison of the properties, attack resistances and applicability of the existing methods

We summarized the properties and attack resistances for each method in Table. 4.1, and the equipment required for the embedding and extraction process with an estimation of its cost in Table. 4.2. Depending on the targeted market, the cost and the availability of the equipment may be a decisive factor for the adoption of the technology. Finally, we summarized the applicability of each method to the proposed scenarios in Table. 4.3. In the tables, the methods have been divided between subsurface and surface methods, as described in Sec. 4.2 and Sec. 4.3. For the surface methods, we refined into two categories: ink & paint and geometric. The ink & paint category includes the methods painting the surface of the object without modifying the geometry, whereas the geometric category includes the methods that only modify the geometry of the surface.

As shown in Table. 4.2, the subsurface methods are generally expensive for decoding, making it more difficult to deploy in real life. Ink-based methods generally require more expensive printer but are cheap for the decoding. Geometric approaches are cheap for the encoding but often require a 3D scanner that may be costly.

Among the proposed scenarios, 'proof of authenticity' is a little bit particular because the watermark should be impossible to duplicate, and any tampering on the model should be detectable instead of being robust to a wide range of attacks like the other scenarios. The only related methods we found applicable for this scenario were the ones [22,32] using quantum dots to produce a physically unclonable function (PUF), but the high cost of the equipment for both the embedding and extracting process makes it difficult to use in practice. RFID based methods could be an interesting alternative because some PUF implementations have been proposed for low-cost RFID chip [11,20], but recent vulnerabilities [7] have been discovered. Still, until a low-cost and proven robust method is found, the other proposed methods are a good compromise that would at least complicate the task for the counterfeiters, even if they offer no protection against an attacker that knows the algorithm used. Our layer thickness method fit in this category, it is

not duplicable by a standard scan and reprint, would complicate the task for a counterfeiter, but anyone knowing the method can reproduce it. On the contrary, our surface norm histogram method is not usable at all for this method because of the resistance to reprint.

Few methods are usable for the 'no-copy flag,' and our surface histogram method is one of them. This is due to the difficult constraints of this scenario: The requirement of being decodable from a standard 3D scan eliminates all the methods that are not purely based on the shape of the surface because the external shape is the only information we obtain from a standard 3D scanner. It should be blind because we do not have access to the original model at the decoding time, and it would be impracticable to search in a database for every scan or print. And finally, it should also be invisible to prevent the user to eliminate it easily, but not below the resolution of a 3D scanner. Our surface norm histogram method is well suited for this scenario because it combines all the requirements similarly to Yamamoto *et al.* [83]. Our layer thickness method would work only if the 3D scanner had enough resolution to see the layers, but our tests showed that it is generally not the case with current 3D scanners.

Most of the methods that aim to be used in copyright protection, crime investigation, and stolen object identification scenarios, require to be unknown by the criminal. This is called 'security by obscurity', and even if it is not a perfect way to protect, it is often effective because a majority of these criminals are not technical experts. It is similar to the yellow dots on printed documents: most consumers are unaware of it even though it has been publicly described in the press [6]. Nevertheless, it requires good invisibility to prevent the criminal to become aware of the protection.

Our layer thickness method has a many common points with LayerCode (variable layer height) [46]. Both methods modify the layer thickness, work with FDM printers, and can be decoded from inexpensive device, making them easily deployable for real applications. The main difference is that LayerCode modifies the thickness of complete layers and can only use integer multiple of the original layer thickness, whereas our method does it locally and has a variable thickness within a layer and can use much lower amplitude. This allows us to embed more data per surface area and reduce the distortions. Unfortunately, it also requires

higher resolution equipment for the decoding and is thus not decodable by most current 3D scanners. However, for flat surface, a standard 2D document scanner produces a high resolution with good contrast that allows us to decode it reliably.

Our surface norm histogram method is most similar to Yamazaki *et al.* [84], Hou *et al.* [28], Hou *et al.* [29], and Yamamoto *et al.* [83]. They are based on an imperceptible modification of the surface shape and take a standard 3D scan as input for the decoding. All of them can resist reprinting, but Hou *et al.* [29] requires that the print axis orientation is conserved during the reprint. Hou *et al.* [28] requires the original model or some features of it to retrieve the printing axis of the model, it is thus a semi-blind method.

Table 4.1: List of the properties and attack resistances of each method.

			Properties						Attacks				
			Blind	Invisibility[1]	Capacity	One-shot extraction	Standard 3D scan[2]	Curved/irregular surface	Non-uniform scaling	Cropping	Local distortion	Smoothing	Reprint
Subsurface		RFID [76] [77] [73]	✓	++	> 64 bits	✓	✗	✓	✓	✓	✓	✓	✗
		Quantum dots (PUF) [22, 32]	✓	+	N/A	✓	✗	✓	✓	✓	✓	✓	✗
		Acoustic [42, 71]	✓	-	~ 4 bits	✓	✗	✓	~	~	✓	✓	✗
		Subsurface cavities [43, 59, 68–70, 81]	✓	++	> 64 bits	~	✗	✗	~	~	✓	✓	✗
Surface	Ink & Paint	QR code with UV ink [50, 85]	✓	++	> 64 bits	✓	✗	✗	✓	✓	✓	?	✗
		LayerCode (dual color) [46]	✓	- -	~ 24 bits	✓	~	✓	✓	✓	✓	✓	✓
		LayerCode (infrared ink) [46]	✓	++	~ 12 bits	✓	✗	✓	✓	✓	✓	✓	✗
		Microdots [23]	✓	?	> 64 bits	✓	✗	✓	✓	✓	✓	✓	✗
	Geometric	Visible geometric tag [39, 88]	✓	- -	> 64 bits	✓	✓	✗	✓	✓	✓	~	✓
		Acoustic [26]	✓	- -	8-24 bits	✓	✓	✓	✓	✓	✓	~	✓
		LayerCode (variable layer height) [46]	✓	+/-	~ 24 bits	✓	✓	✓	✓	✓	✓	~	✗
		Spectral watermarking [84]	✗	+	256 bits	✗	✓	✓	?	~	✓	✓	✓
		Hou *et al.* [28]	✗	+	~ 24 bits	✗	✓	✓	?	✓	✓	✓	✓
		Hou *et al.* [29]	✓	+	1 bit	✗	✓	✓	?	~	✓	✓	~
		Yamamoto *et al.* [83]	✓	+	~ 8 bits	✗	✓	✓	✗	✗	✓	✓	✓
		Surface norm histogram (Chap. 5)	✓	+	16-32 bits	✗	✓	✓	✗	✗	✓	✓	✓
		Layer Thickness (Chap. 6)	✓	+	~ 64 bits	✓	~	~	✓	✓	✓	~	✗

✓ (green) means that the property or attack is supported by the method, ~ (orange) means that it is supported with some important limitations, and ✗ (red) means that it is not supported.

1: '++' means totally invisible for human-eye, '+' means that some small artifacts are visible if we look carefully, '+/-' means that some artifacts are presents and easily visible, '- -' means that the watermark is clearly visible. 2: It is considered true if the features encoding the watermark are preserved in a standard 3D scan, it does not require that the proposed extraction method uses a 3D scan as input.

Table 4.2: List of the equipment required and cost of each method.

			Watermark embedding		Watermark extraction	
			Equipment	Cost[1]	Equipment	Cost[1]
Subsurface		RFID [76] [77] [73]	Any 3D printer RFID chip	$ $	RFID reader	$
		Quantum dots (PUF) [22, 32]	Polyjet printer Quantum dots	$$$ $	Florescent microscope	$$$
		Acoustic [42, 71]	Any 3D printer	$	Microphone	$
		Subsurface cavities:				
		Terahertz [81]	Any 3D printer, some limitations for FDM	$	Terahertz camera	$$$
		Thermal [59, 68, 70]	Any 3D printer	$	Thermal camera	$$
		Near infrared [69]	Any 3D printer	$	Near-infrared light and camera	$
		High-frequency pattern [43]	SLA, polyjet, or similar (not FDM)	$$	Projector and monochrome camera	$$
Surface	Ink & Paint	QR code with UV ink [50, 85]	Polyjet printer	$$$	UV light RGB camera	$ $
		LayerCode (dual color) [46]	Dual-color printer	$	RGB camera	$
		LayerCode (infrared ink) [46]	Dual-resin printer	$$	RGB camera	$
		Microdots [23]	Microdots (bought online or engraved with laser)	$	UV light RGB camera with macro lens	$ $
	Geometric	Visible geometric tag [39, 88]	Any 3D printer	$	RGB camera	$
		Acoustic [26]	Any 3D printer	$	Microphone	$
		LayerCode (variable layer height) [46]	FDM printer	$	RGB camera	$
		Spectral watermarking [84]	Any 3D printer	$	3D scanner	$$
		Hou *et al.* [28]	Any 3D printer	$	3D scanner	$$
		Hou *et al.* [29]	FDM 3D printer	$	3D scanner	$$
		Yamamoto *et al.* [83]	Any 3D printer	$	3D scanner	$$
		Surface norm histogram (Chap. 5)	Any 3D printer	$	3D scanner	$$
		Layer Thickness (Chap. 6)	FDM printer	$	document scanner	$

1: '$' (green) is between 0 and 1000$, '$$' (orange) is between 1000$ and 5000$, '$$$' (red) is above 5000$.

Table 4.3: List of the scenarios for each method.

			Copyright protection						Metadata			
			Proof of ownership	Proof of authenticity	Traitor tracing	No-copy flag	Crime investigation	Stolen object	Broken part replacement	Robotic grasping	Smart factory	Augmented reality
Subsurface		RFID [76] [77] [73]	✗	∼	✗	✗	✗	✓	✓	✓	✓	✓
		Quantum dot (PUF) [22, 32]	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
		Acoustic [42, 71]	✗	∼	✗	✗	✗	∼	∼	∼	✗	✓
		Subsurface cavities [43, 59, 68–70, 81]	✗	∼	✗	✗	✓	✓	∼	✓	✓	✓
Surface	Ink & Paint	QR code with UV ink [50, 85]	✗	∼	✗	✗	∼	✓	✓	✓	✓	✓
		LayerCode (dual color) [46]	∼	✗	∼	✗	✗	∼	✓	✓	✓	✓
		LayerCode (infrared ink) [46]	✗	∼	✗	✗	✓	✓	✓	✓	✓	✓
		Microdots [23]	✗	∼	✗	✗	✗	✓	✓	∼	✓	✓
	Geometric	Visible geometric tag [39] [88]	∼	✗	∼	✗	∼	∼	✓	✓	✓	✓
		Acoustic [26]	∼	✗	∼	✗	∼	✓	✓	∼	✓	✓
		LayerCode (variable layer height) [46]	∼	∼	∼	∼	✓	✓	✓	✓	✓	✓
		Spectral watermarking [84]	✓	✗	✓	✗	✗	∼	∼	✗	∼	✗
		Hou *et al.* [28]	✓	✗	✓	✗	✗	∼	✗	✗	∼	✗
		Hou *et al.* [29]	✓	✗	✓	∼	✓	✓	✗	✗	∼	✗
		Yamamoto *et al.* [83]	✓	✗	✓	✓	✓	✓	✗	✗	∼	✗
		Surface norm histogram (Chap. 5)	✓	✗	✓	✓	✓	✓	✗	✗	∼	✗
		Layer Thickness (Chap. 6)	∼	∼	∼	∼	✓	✓	✓	∼	✓	∼

✓ (green) means that the method is usable for the scenario, ∼ (orange) means that the method can be used for the scenario with some important limitations, and ✗ (red) means that the method can not be used for the scenario.

5 Blind watermarking for 3d printed objects using moment alignment and surface norm distribution

In this chapter, we present a method that embeds the watermark in the shape of the object using an histogram of surface norm for embedding the data and 3D moment based alignment to orient the model to a standard pose. The decoding can be done by a standard 3D scanner but requires a complete scan covering all the surface of the object. This method is based on the vertex norm watermarking technique [15, 16] that we modified to resist to the 3D printing and scanning process.

The contributions of our method are as follows.

- It works with any 3D printer and scanner as long as the scan is complete and good quality.
- It works with most object shapes.
- It can resist to a reprint from any orientation.

This method is good for the copyright protection, crime investigation and stolen object identification scenarios because the watermark can be retrieved from a standard 3D scan and resists to reprint. It means that even if a counterfeiter duplicate the object, the watermark is still retrievable in the duplicated objects. This method is more difficult to apply to the metadata embedding scenarios because they generally require a fast extraction process. Most 3D scanners are

slow to capture and merge all the faces of an object so the extraction of the watermark is time consuming.

This chapter is organized as follows: In section 5.1, we describe the principle of our algorithm. In section 5.2, we detail the implementation. In section 5.3, we describe our experiments. And finally, in section 5.4, we discuss our results and plans for future works.

5.1 Core ideas of the proposed method

The proposed method is an extension of our previous paper [19], which was based on the 3D mesh watermarking method from Cho *et al.* [15,16], that used a distribution of vertex norms. Cho *et al.*'s method has been developed for digital data with mesh structure, not for the print-scan scenario. It has a significant advantage due to its invariance to translation, rotation, rescaling, and vertex connectivity. However, it can also have problems when resampling is applied, which is the case in 3D print-scan. In this section, we first briefly explain the original method from Cho *et al.* [15,16] (Sec. 5.1.1), then explain our contributions to make it suitable for 3D print-scan. Section. 5.1.2 explains how we made our method resistant to resampling, Sec. 5.1.3 explains how to robustly estimate the center of the object and its minimal and maximal norms to prevent misalignment, Sec. 5.1.4 and Sec. 5.1.5 explain how to subdivide the surface into bins and combine them to obtain a good redundancy.

For convenience, we refer to the minimal and maximal norms as 'min-max norms' in the rest of this paper.

5.1.1 Watermarking using distribution of vertex norms [15, 16]

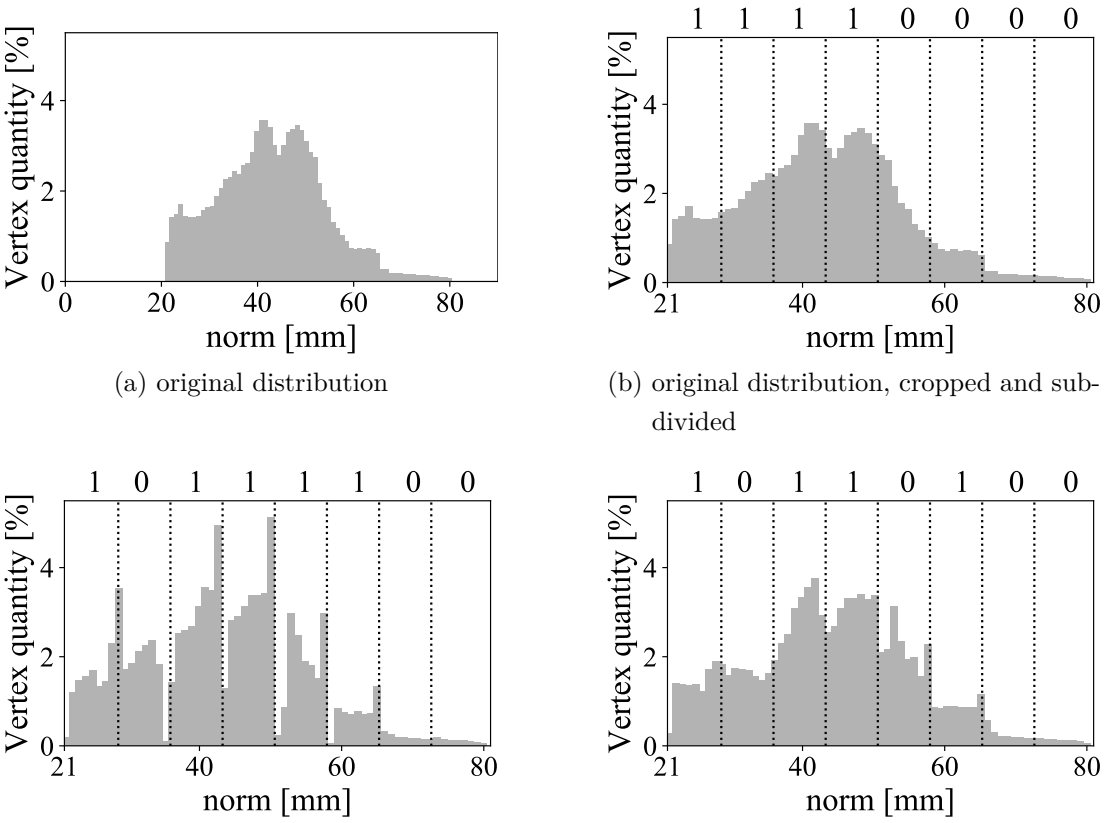
We describe here the mesh watermarking method that we extended in this paper. This method is based on a histogram of vertex norms and goes as follows. Each vertex V_i of the mesh can be represented in spherical coordinates $(\rho_i, \theta_i, \phi_i)$ where ρ_i is the norm of the vertex, that is, the distance between the vertex and the center of the object, and (θ_i, ϕ_i) are the direction angles. By taking all the vertex

norms ρ, a histogram of the norms distribution can be created, as shown in Fig. 5.1(a). It is then cropped to keep only the range between the minimum and maximum norms, and subdivided into bins, as shown in Fig. 5.1(b). The vertical lines represent the separations between the different bins. One bit is embedded inside each bin by modifying the norms distribution such that the mean of the distribution is respectively on the left or the right of the center of the bin, to encode a 0 or 1, as shown in Fig. 5.2. To modify the norms distribution, each vertex norm ρ_i is increased or decreased, without exceeding the bounds of its bin. The amplitude of the modification is kept low enough such that the shape deformation is imperceptible to the human eye, and the directions (θ_i, ϕ_i) stay unchanged. Figure 5.1(c) shows an example of histogram after embedding a watermark.

5.1.2 Increasing robustness against resampling

The original method uses the vertex norms to encode the watermark, which is usable for mesh watermarking but not for 3D printing. The print-scan process produces a complete resampling of the mesh and causes decoding errors because the vertex positions in which the watermark has been encoded are lost, and new vertices are generated. Using a uniform sampling with high vertex density could mitigate this effect because the newly generated vertices would be close to the original ones and in the same density, but the watermarking process does not preserve this uniformity, so it would still be a problem.

To prevent this problem, we compute the norms histogram continuously on all the surface of the object instead of a discrete set of vertices. It is a continuous approach instead of a discrete one and is thus not sensitive to resampling if the shape is preserved. In practice, to compute the average norm in each bin, each triangle is first subdivided such that each sub-triangle fits completely inside one bin. Then, we compute the average norm of the surface of all the triangles contained in each bin. An integration of the surface norm on a 3D triangle is difficult to do analytically, so instead, we approximate it numerically by the



(a) original distribution

(b) original distribution, cropped and subdivided

(c) watermarked distribution

(d) watermarked distribution after print simulation

Figure 5.1: Distribution of vertex norms from the Stanford bunny model, each vertical striped line represent the border of a bin. (a)the full histogram before watermarking. (b) histogram before watermarking, cropped to keep only the range of the the min-max norms, and subdivided into eight bins. (c) histogram after watermarking, watermark value = 10111100. (d) histogram after printing simulation.

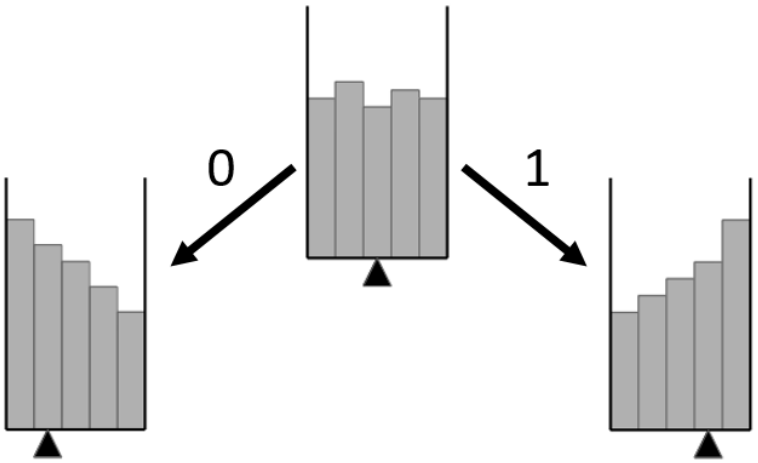


Figure 5.2: Encoding one bit of data by shifting the mean of the distribution to the left or right side of the bin. The black triangle cursors below each bin represent the position of the mean of the distribution.

average of the norm of the three vertices multiplied by the area of the triangle.

$$Surf(V_A, V_B, V_C) = \frac{||\overrightarrow{V_A V_B} \times \overrightarrow{V_A V_C}||}{2} \tag{5.1}$$

$$A = \sum_{(V_A,V_B,V_C)\in Bin} Surf(V_A, V_B, V_C) \tag{5.2}$$

$$\mu = \sum_{(V_A,V_B,V_C)\in Bin} \frac{||V_A||+||V_B||+||V_C||}{3} \frac{Surf(V_A, V_B, V_C)}{A} \tag{5.3}$$

Here, V_A,V_B,V_C are the three vertex of a triangle in the bin, *Surf* is the function that computes the area of a triangle, A is the total area of the bin, and μ is the average norm of the bin.

To obtain a high precision on the numerical approximation, each triangle must be subdivided until the difference between the norm of the center and the mean of the norms of the three vertices become negligible. This condition is expressed as

$$\left| \left|\left|\frac{V_A + V_B + V_C}{3}\right|\right| - \frac{||V_A|| + ||V_B|| + ||V_C||}{3} \right| \leq \epsilon_c, \tag{5.4}$$

where ϵ_c denotes the precision to reach. ϵ_c is expressed in the same unit as the vertices coordinates, which is in our case in millimeter. Section 5.3.1 evaluates the precision based on ϵ_c value.

5.1.3 Improving the robustness to misalignment

To extract the watermark without errors, we need to ensure that we do not have any misalignment during the decoding process. Any error on the center position would affect the surface norms estimation, and any error on the min-max norm values would shift the bins. Section 5.1.3 and Section 5.1.3 explain how we compute more robustly the center of the object and the min-max norms, respectively.

Center position estimation

As explained in [31], center estimation is a critical part of the method. If the center position is changed, the norms are changed too, and the watermark may not be extracted correctly. Because the mean of the vertices is not resistant to resampling, we use a moment-based center estimation that has been proven to be more resistant in the 3D printing context [29].

$$c = (\overline{x}, \overline{y}, \overline{z}) = \frac{(M_{100}, M_{010}, M_{001})}{M_{000}}, \tag{5.5}$$

$$M_{pqr} = \iiint x^p y^q z^r \tau(x, y, z) dx dy dz \tag{5.6}$$

$$\tau(x, y, z) = \begin{cases} 1, & \text{if } (x, y, z) \text{ is inside the mesh} \\ 0, & \text{otherwise} \end{cases}$$

where c is the estimated center, M_{pqr} denotes the p, q, r-th order volume moment of the mesh, and $\tau(x, y, z)$ is an indicator function. The 3D moments can be computed efficiently using the method described in [86].

The watermarking process slightly modifies the center of gravity, the min-max norms, and the orientation. To avoid any problem due to the modifications, we recompute these parameters at each iteration of the optimization process, which is explained later, until convergence.

Min-max norms quantization

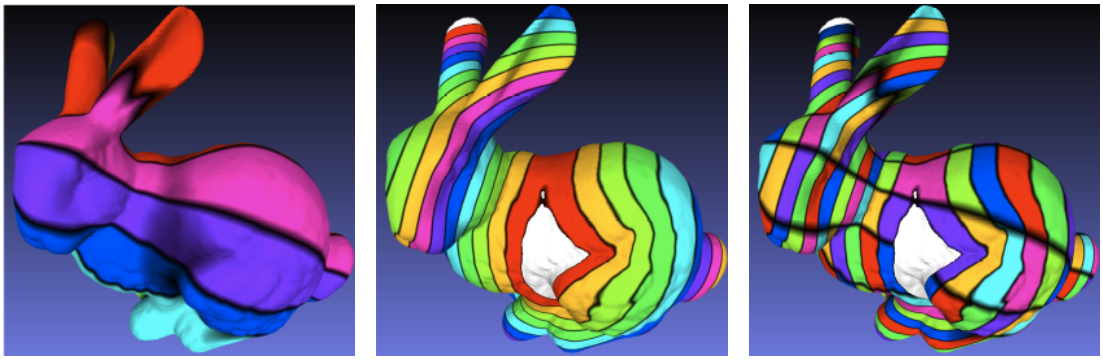
On the object surface, the regions that are within a few tenths of millimeters of the min-max norms are the most sensitive to errors. Any distortion can modify

the min-max norm values and result in a shift of the histogram. To prevent this problem, we quantize the min-max norms instead of using a continuous value. The shape is first slightly modified such that the min-max norms are an integer multiple of Q_{norm} during the watermarking process, with Q_{norm}, the quantization step. At extraction time, we round the measured min-max norms to the closest quantized value. It provides resistance to any min-max norm error strictly lower than $Q_{norm}/2$. In order to stay robust to uniform scaling, Q_{norm} is defined as a value proportional to the cube root of the volume of the object.

In this paper, the quantized min-max norms are written $\lfloor \mathcal{N} \rfloor$ and $\lceil \mathcal{N} \rceil$, respectively. $\Delta\mathcal{N}$ is equal to $\lceil \mathcal{N} \rceil - \lfloor \mathcal{N} \rfloor$.

5.1.4 Mesh subdivision into bins

The bin subdivision consists of selecting the portions of the surface used for each bin. This is a delicate task because any misalignment during the extraction process could strongly affect the decoded value. As illustrated in Fig. 5.3, we use two subdivision methods simultaneously to cut the mesh into slices along different directions. The angular subdivision, illustrated in Fig. 5.3(a), is the main extension to our previous paper. It allows to increase the number of bins without producing too thin slices that would get close to the precision limits of the 3D printer and 3D scanner.



(a) Angular subdivision (b) Radial subdivision (c) Angular + radial subdivision

Figure 5.3: Mesh subdivision into bins. (a) angular subdivision into eight bins. (b) radial subdivision into 16 bins + two border regions (in white). (c) combined angular and radial subdivision into 128 bins + two border regions (in white)

Radial subdivision

The radial subdivision is similar to the original vertex norms watermarking method [15, 16]: we compute the norm for each point on the surface with $\rho = \sqrt{(x-\overline{x})^2 + (y-\overline{y})^2 + (z-\overline{z})^2}$, where $(\overline{x}, \overline{y}, \overline{z})$ is the center of the object defined in Sec. 5.1.3.

We search for the min-max norms, apply the quantization described in Sec. 5.1.3, leave a margin $\delta_{\mathcal{N}}$ around the quantized min-max norms and then subdivide the remaining range into N_r slices of equal size. The size of a slice can thus be computed by :

$$S_r = \frac{(1 - 2\delta_{\mathcal{N}})\Delta\mathcal{N}}{N_r} \tag{5.7}$$

with S_r, the size of the radial slice. We generally use $\delta_{\mathcal{N}} = 0.05\Delta\mathcal{N}$ to prevent the min-max norms to be modified during the watermark embedding.

Angular subdivision

Compared to the radial subdivision, the angular subdivision method is not invariant to rotation, and we need to align to a standard orientation. We use the

moment PCA method [86] to align the principal axes to the base axes of the coordinate system. It is done by firstly centering the model with the moment center estimation method, as explained in Sec. 5.1.3. Then the principal axis can be obtained by applying principal component analysis (PCA) on the second-order moment matrix :

$$\begin{bmatrix} M_{200} & M_{110} & M_{101} \\ M_{110} & M_{020} & M_{011} \\ M_{101} & M_{011} & M_{002} \end{bmatrix} \tag{5.8}$$

where M_{pqr} denotes the p, q, r-th order moment of the mesh. To make the result unique, we need to ensure that the third principal axis is the cross product of the first and second PCA axes and that the 3^{rd} order moments M_{300} and M_{030} are positive [86] . The mesh is then rotated to align its principal axis to the world axis. Each vertex is transformed by :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} & \mathcal{A}_{13} \\ \mathcal{A}_{21} & \mathcal{A}_{22} & \mathcal{A}_{23} \\ \mathcal{A}_{31} & \mathcal{A}_{32} & \mathcal{A}_{33} \end{bmatrix} \begin{bmatrix} x - \overline{x} \\ y - \overline{y} \\ z - \overline{z} \end{bmatrix} \tag{5.9}$$

where (x', y', z') is the centered and reoriented vertex, $\mathcal{A}$ is the eigenvector matrix of the second-order moment matrix obtained by the PCA, (x, y, z) is the vertex before centering and reorienting, and $(\overline{x}, \overline{y}, \overline{z})$ is the center of the object described in Sec. 5.1.3.

Finally, we can apply the subdivision by computing for each vertex the angle $\Phi = \text{atan2}(z', y')$, and subdivide the mesh into bins of size $S_a = \frac{2\pi}{N_a}$ rad on the angle Φ, where N_a denotes the number of angular subdivisions.

The moment PCA alignment method does not work if the mesh has rotational symmetry around an axis. On this kind of mesh, we need to restrict N_a to 1 in our algorithm, meaning we do not apply angular subdivision and use only the radial subdivision.

5.1.5 Combination of multiple non-consecutive bins

To get a good resistance against local printing artifacts and scanning errors, we combine multiple bins for each watermark bit. These bins are spread over the

whole surface, avoiding the bins encoding the same bit to be concentrated on only one region of the object. The spreading is done via a fixed-function independent of the mesh shape and thus does not bring any problem at extraction time. To decide which bin contributes to encode a watermark bit, we use a two-dimensional lookup table of size $N_a \times N_r$, that contains a bit index and an invert flag for each bin, with N_a and N_r, the number of angular and radial bins, respectively. The bit index $binId_{i,j}$ defines which watermark bit is encoded in the bin (i, j), and $binInvert_{i,j}$ is a boolean flag that, if True, inverts the value encoded in the bin (i, j), i.e., it applies a NOT operator. The bit indices are chosen consecutively along the angular dimension, and we apply a constant shift along the radial dimension to avoid having two neighboring bins encoding the same bit. The invert flags are chosen by a pseudo-random sequence with a known seed so that we can retrieve the same lookup table at extraction time. More precisely, Algorithm 1 shows how *binId* and *binInvert* are used to combine the bins into the watermark signal, Algorithm 2 describes how we generate the lookup table, and Fig. 5.4 gives an example of a generated table.

The regular distribution of the bit index guarantees that the signal will be spread through the whole mesh and not concentrated in some regions. The invert flags reduce the sensitivity to angular misalignment because misaligned portions of the bins do not get a good correlation of their invert flags. They also help the embedding by decreasing the initial strength of the bins, as observed in Sec. 5.3.3.

The watermark bit values are computed by a weighted average of the bins. The weights are calculated such that a bin with a large surface gets more influence in the result than a bin with a small surface. However, we also avoid that a few bins with large surfaces get too much influence and make negligible the value from the other bins. Multiple weight functions are possible, and there is a room for improvement to find the optimal one, but we chose a relatively simple one that consists of an identity function of the surface area until it reaches the mean area $\overline{A}$, then multiply by 0.25 the excess:

$$weight(a, \overline{A}) = \begin{cases} a, & \text{if } a < \overline{A} \\ \frac{3\overline{A}+a}{4}, & \text{otherwise} \end{cases} \tag{5.10}$$

where a is the surface area of the bin, and $\overline{A}$ is the surface area of the whole object divided by the number of bins ($N_a N_r$).

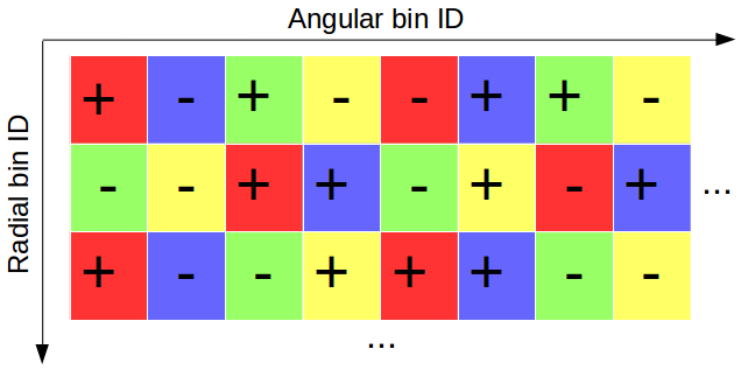


Figure 5.4: Bin lookup table example with four bits : each cell corresponds to one bin ID, each color corresponds to one bit of the watermark sequence, each "+" or "-" corresponds to the invert flag of the bin.

5.2 Watermark embedding and extraction algorithm implementation

In this section, we describe the algorithm used to embed and extract the watermark. Section 5.2.1 describes the extraction algorithm and Sec. 5.2.2 describes the embedding algorithm. The extraction algorithm is used inside the embedding algorithm as illustrated in Fig. 5.6 so we describe it first.

5.2.1 Watermark extraction algorithm

As illustrated in Fig. 5.5, the extraction algorithm consists of computing the histogram of surface norms for each bin, then combine the bins to compute the mean for each bit. Each bit is set to 1 if its histogram mean is greater than 0.5, or 0 otherwise. More concretely, we begin by centering the model with the moment method described in Sec. 5.1.3, and doing the rotation alignment described in Sec. 5.1.4. Next, we subdivide each triangle such that the three vertices are fully included in a single bin and until it matches the criterion from Eq. 5.4 in Sec. 5.1.2. We then define a function that remaps a norm X to the range [0,1], with 0 and 1 corresponding to the minimum and maximum radial boundary of the bin that contains it, respectively :

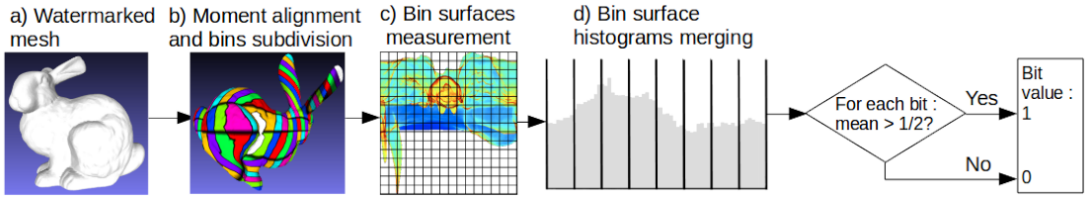


Figure 5.5: Overview of the watermark extraction algorithm. (a) the mesh from which we want to extract the watermark. (b) the mesh after 3D moment-based alignment, and bins subdivision. bins with the same color are assigned to the same watermark bit. (c) surface distribution is computed inside each bin, with white for no surface, and from blue to red for low quantity to high quantity (d) histogram formed by the combination of the bins assigned to the different bits. Each column corresponds to one bit, whose value is 1 if the distribution is more concentrated on the right side of the bin, or 0 otherwise.

$$f_j(X) = \frac{X - (\lfloor \mathcal{N} \rfloor + \delta_{\mathcal{N}} \Delta\mathcal{N} + jS_r)}{S_r} \tag{5.11}$$

where $\lfloor \mathcal{N} \rfloor$ is the quantized minimal norm (Sec. 5.1.3), $\delta_{\mathcal{N}}$ is the margin around min-max norm (Sec. 5.1.4), $\Delta\mathcal{N}$ is the difference between the minimum and maximum norm (Sec. 5.1.3), j is the index of the radial slice, and S_r is the size of a radial slice (Sec. 5.1.4). By using the equations 5.2 and 5.3 from Sec. 5.1.2 combined with the remapping function (Eq. 5.11), we compute the total area $A_{i,j}$ and mean norm $\mu_{i,j}$ for each bin (i, j) :

$$\begin{aligned} A_{i,j} &= \sum_{(V_A,V_B,V_C)\in T_{i,j}} Surf(V_A, V_B, V_C), \\ \mu_{i,j} &= \sum_{(V_A,V_B,V_C)\in T_{i,j}} f_j\left(\frac{||V_A||+||V_B||+||V_C||}{3}\right) \frac{Surf(V_A, V_B, V_C)}{A_{i,j}}, \end{aligned} \tag{5.12}$$

where V_A, V_B, V_C are the coordinates of a vertex, $T_{i,j}$ is the list of triangles included in the bin (i, j).

The bin values are then combined to produce the vector μ' of size N_b, the number of bits of the watermark signal :

$$\mu' = binMerge(\mu, A) \tag{5.13}$$

with *binMerge* corresponding to Algorithm 1, explained in Sec. 5.1.5. The parameters *binId* and *binInvert* used in Algorithm 1 are generated by Algorithm 2.

We finally extract the watermark bits by thresholding :

$$w'_k = (\mu'_k > 0.5) \tag{5.14}$$

Algorithm 1: Bin merging algorithm

input : The number of angular bins N_a, radial bins N_r and bits N_b. The mean table μ, area table A, bit index table $binId_{i,j}$ and bin invert table $binInvert_{i,j}$ of size $N_a \times N_r$

output: The mean array μ' and area array A' of size N_b

$\mu' := [0, ..., 0]$;
$T' := [0, ..., 0]$;
$A' := [0, ..., 0]$;
$\overline{A} := average(A)$;
for *i in [0,*N_a*]* **do**
  for *j in [0,*N_r*]* **do**
    $k = binId_{i,j}$;
    $W = weight(A_{i,j}, \overline{A})$; `// Eq. 5.10`
    if $binInvert_{i,j}$ **then**
      $T'_k \mathrel{+}= (1 - \mu_{i,j}) * W$
    else
      $T'_k \mathrel{+}= \mu_{i,j} * W$
    end
    $A'_k \mathrel{+}= W$;
  end
end
$\mu' = T'/A'$

5.2.2 Watermark embedding algorithm

The embedding algorithm is illustrated in Fig. 5.6 and described in Algorithm 3. It is an iterative process that applies gradient descent on each vertex until all the watermark bits are encoded with enough strength.

Algorithm 2: bin-bit assignation algorithm

input : The number of angular bins N_a, radial bins N_r and bits N_b.

output: The bit index table $binId_{i,j}$ and bin invert table $binInvert_{i,j}$ of size $N_a \times N_r$

```
srand(1) ; // Seed for the random function
```

Set $currentBit := 0$;

for *j in [0,*N_r*]* **do**

  for *i in [0,*N_a*]* **do**

    $binId_{i,j} := currentBit$;

    $binInvert_{i,j} := (rand()\%2 = 1)$;

    $currentBit := (currentBit + 1)\%N_b$;

  end

  if $((j + 1) * N_a)\%N_b = 0$ **then**

    $currentBit := (currentBit + N_b/4 + 1)\%N_b$

  end

end

Before starting the optimization, the mesh is first resampled to get a relatively uniform and well-adapted vertex density. If the mesh has too large triangles, especially larger than the bin size, it is not possible to modify precisely the surface norm histogram and is then difficult to embed a watermark. On the contrary, if the meshing is too detailed, the optimization process can produce a non-smooth surface and have a convergence problem. In practice, we apply mesh simplification with tools such as MeshLab [17] if the mesh is too high resolution, or triangle subdivision if the resolution is too low. For each edge of the mesh, we split it into two equal parts if the edge length is bigger than the threshold ϵ_e or if the difference of the norm of the two vertices is bigger than the threshold ϵ_n.

Then we start the iterative process of the gradient descent. At the start of each iteration, we recompute the center of gravity, orientation, and min-max norms (Sec. 5.1.3, 5.1.3, 5.1.4).

The cost function for the gradient descent is defined by:

$$F = \sum_{k=0}^{N_b} \max(0, \mathcal{S}_{enc} - \mathcal{S}_k)^2 \tag{5.15}$$

where F is the cost function, $\mathcal{S}_{enc}$ is the encoding strength, and $\mathcal{S}_k$ is the current encoded strength of k^{th} bit defined as follows:

$$\mathcal{S}_k = \begin{cases} 2(\mu'_k - 0.5), & \text{if } w_k = 1 \\ 2(0.5 - \mu'_k), & \text{if } w_k = 0 \end{cases} \tag{5.16}$$

Note that outside the training process or performance evaluation, the ground-truth value w_k is not available. In that case, we approximate the strength by replacing w_k by w'_k in Eq. 5.16, which results in an absolute value of the strength.

Finally, the min strength $\lfloor \mathcal{S} \rfloor$ and mean strength $\overline{\mathcal{S}}$ of the watermark signal, used for performance evaluation, are defined by :

$$\lfloor \mathcal{S} \rfloor = \min_k \mathcal{S}_k \tag{5.17}$$

$$\overline{\mathcal{S}} = \frac{1}{N_b} \sum_{k=0}^{N_b} \mathcal{S}_k \tag{5.18}$$

The strength metrics are used during the training process and for performance evaluation but can also be used in real situation to estimate the confidence of the result. Having a high value on both $\lfloor \mathcal{S} \rfloor$ and $\overline{\mathcal{S}}$ provides good confidence in the correctness of the result.

To choose the best encoding strength $\mathcal{S}_{enc}$ and sampling parameters ϵ_n and ϵ_e, we evaluate multiple sets of values for these parameters, encode the watermark for each set and keep the one that has the highest value for the minimal strength $\lfloor \mathcal{S} \rfloor$. We can use the maximum root mean square error (MRMS) and mesh structural distortion measure (MSDM) metrics to define additional deformation and visibility thresholds [40]. The encoding strength and sampling parameters are not required for decoding, and can thus be freely chosen during the encoding process. Typical values were $\mathcal{S}_{enc} = 5\%$, $\epsilon_e = \Delta\mathcal{N}/\{32, 64\}$ and $\epsilon_n = \Delta\mathcal{N}/\{64, 128\}$.

5.3 Experiments

To evaluate the performances of our method, we tested the influence of our parameters and the resistance to multiple attacks. We evaluated some parameters

Algorithm 3: Watermark embedding algorithm

input : the list of vertex V and triangles of the mesh to watermark, the N_b bits watermark sequence w, the optimization rate λ, and all the other watermark parameters used in watermark extraction and embedding algorithm

output: the watermarked mesh obtained by the optimization process

Resample the mesh (Sec. 5.2.2);

Set N_V, the number of vertex V;

Compute $\overrightarrow{N}_v$, the normal for each vertex v, obtained by averaging the normal of the triangles incident to the vertex v ;

foreach *iter in range(0,nbIter)* **do**

  Center, reorient the mesh (Sec. 5.1.3 and 5.1.4) and apply the same rotation to each normal $\overrightarrow{N}_v$;

  Compute $\lfloor \mathcal{N} \rfloor$ and $\lceil \mathcal{N} \rceil$ (Sec. 5.1.4);

  Compute $\mu_{i,j}$ and $A_{i,j}$ for each bin ; `// Eq. 5.12`

  Set $\mu' = binMerge(\mu, A)$; `// Eq. 5.13`

  Set $\Delta\mu := []$;

  foreach k *in range(0,*N_b*)* **do**

    if $w_k = 1$ **then**

      $\Delta\mu_k := \max(0, \mathcal{S}_{enc} - 2(\mu'_k - 0.5))$

    else

      $\Delta\mu_k := -\max(0, \mathcal{S}_{enc} - 2(0.5 - \mu'_k))$

    end

  end

  Set ΔV, a $N_V \times N_b$ matrix, in which $\Delta V_{v,k}$ represent the gradient of μ'_k when moving the vertex v along its normal $\overrightarrow{N}_v$;

  foreach v *in* V **do**

    $V_v := V_v - \lambda(\Delta V_{v,:} \cdot \Delta\mu)\overrightarrow{N}_v$;

  end

end

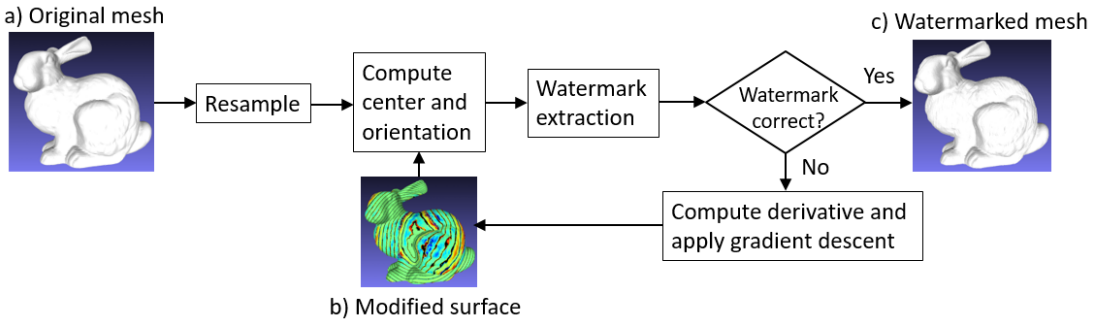


Figure 5.6: Overview of the watermark embedding algorithm. (a) The mesh before watermarking (b) The shape modifications, with colors from red to blue when the surface is moved on the direction of the normal or on the opposite direction, respectively. Green when there is little to no modification. (c) The resulting mesh after watermarking process.

specific to our algorithm, such as a misalignment of the center of gravity, the rotation alignment, and the min-max norms. Because our target is 3D printing, we evaluated with 3D print simulation and with a few real printed objects. Except when specified differently, we used the ten meshes from the dataset proposed in [75], illustrated in Fig. 5.7, with five different watermark values for each, and reported the average result.
We used two metrics for performance evaluation :

- Success rate, which is an all-or-nothing metric. Each evaluation provides a score of 100% if all the bits are correct, or 0% if there is at least one bit error.
- Bit-error rate (BER), which corresponds to the ratio of wrongly decoded bits.

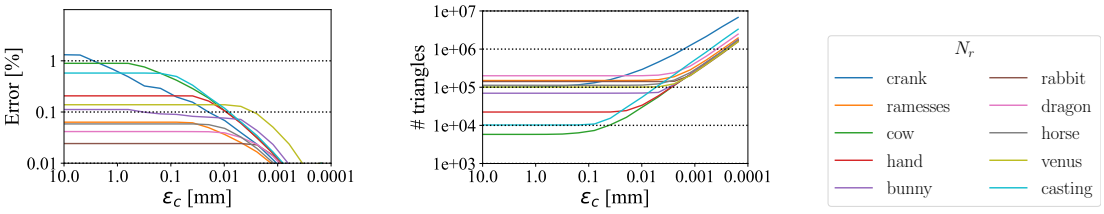
When not specified differently, we evaluated our method on the database using $N_b = \{8, 16, 32\}$, $N_a = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$, $N_r = \{16, 32, 48, 64, 128\}$, $\delta_{\mathcal{N}} = 5\%$, $Q_{norm} = 0.2/\sqrt[3]{86}$, $\epsilon_e = \Delta\mathcal{N}/\{32, 64\}$, $\epsilon_n = \Delta\mathcal{N}/\{64, 128\}$, $\epsilon_c = 0.01$ mm, $\mathcal{S}_{enc} = 5\%$. The quantization step Q_{norm} is proportional to the cube root of the volume of the object. For our default printing volume 86 cm^3, it corresponds to a 2 mm step.

(a) Rabbit (b) Hand (c) Venus (d) Crank

(e) Bunny (f) Dragon (g) Casting

(h) Cow (i) Horse (j) Ramesses

Figure 5.7: Dataset used for our experiments [75]



(a) Histogram error (b) Number of triangles

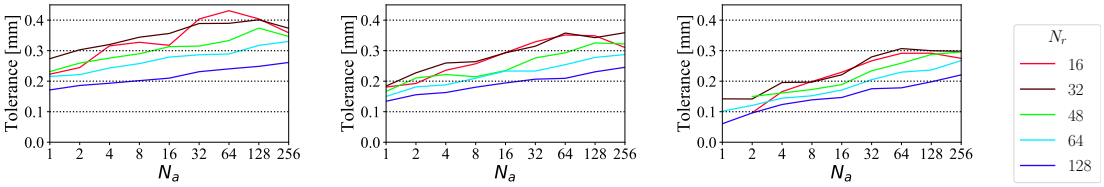
Figure 5.8: Influence of the subdivision parameter ϵ_c on the triangles numerical integration error and on the total number of triangles.

5.3.1 Numerical integration of the surface norm

As explained in Sec. 5.1.2, the integration on the triangles is done numerically instead of analytically, and the precision depends on the parameter ϵ_c. A too high ϵ_c would bring errors on the histogram calculation, but a too low would make it computationally expensive. This subdivision can be done triangle-by-triangle during the integration and does not require to store the full mesh at higher resolution, it does not bring any memory consumption problem. The two graphs of Fig. 5.8 show the effects of the subdivision with various ϵ_c values. Figure 5.8(a) shows the mean histogram error, which is expressed in the same units as the encoding strength $\mathcal{S}_{enc}$. Figure 5.8(b) shows the resulting number of triangles to process. We observe that ϵ_c around 0.01 mm produces a negligible error if compared to an encoding strength of about 5%, and also keeps the number of subdivisions reasonable.

5.3.2 Resistance to misalignment

Compared to local errors made by small printing artifacts that only affect their corresponding bins, misalignment affects all the bins. Three types of misalignment can occur in our method: on the center position, on the min-max norms, and on the orientation. In this section, we encode a watermark in the database models and then decode it with increasing shift values applied to the center position, the min-max norms, or the orientation angle. In the following subsections, we refer the tolerance to misalignment as the highest shift value we can apply to a watermarked model without getting errors in the decoded value.



(a) $N_b = 8$ (b) $N_b = 16$ (c) $N_b = 32$

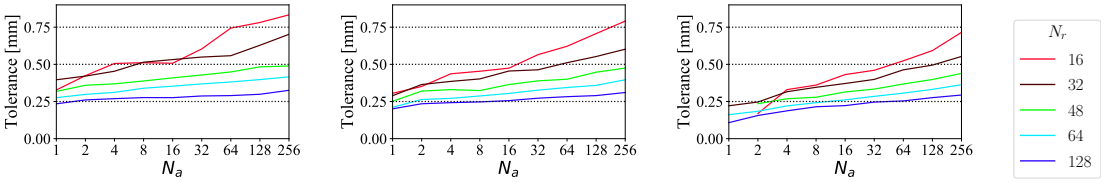
Figure 5.9: Mean center position error tolerance. Vertical axis is the mean misalignment distance of the center before getting an error. Object volume : 86 cm^3

Center misalignment

An error in the center position results in an error in the measurement of all the surface norms. This is the most sensitive parameter in our method. Errors on the center position can mainly be caused by non-uniform scaling that may happen in a poorly calibrated printer, by warping effects due to bad control of the temperature during the print and cooling, or by cropping if a part of the object breaks. It can also happen during the scanning process if the captor is poorly calibrated or if there is an error during the fusion of multiple scans. We evaluated the center misalignment tolerance using 20 random translation axes on which we applied a shift. The result was considered wrong if any of the 20 shifts produced an error. The results are shown in Fig. 5.9. We observe that the center misalignment tolerance decreases when N_r increases, because the radial slices become smaller, and the influence of the misalignment is inversely proportional to the radial size of the bins. We also observe that increasing N_a while keeping the other parameters constant, improves the misalignment tolerance, probably because it increases the redundancy.

Min-Max norms error

The radial subdivision requires to find the min-max norms, and subdivide the range in a certain number of bins. Getting an error on the min-max norm values would shift all the bin positions and produce errors on the decoded bin values. It can happen if we get printing or scanning artifacts around the min-max norm



(a) $N_b = 8$ (b) $N_b = 16$ (c) $N_b = 32$

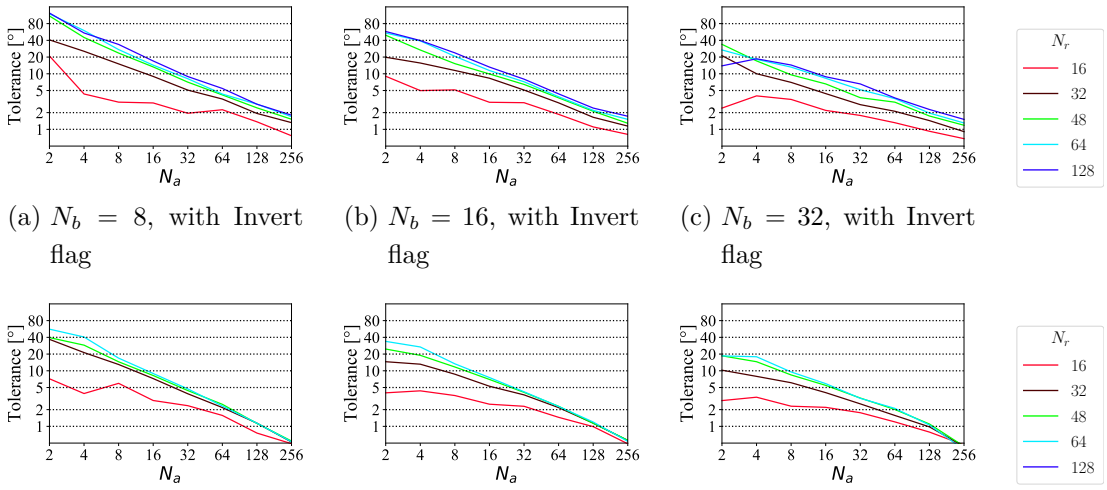
Figure 5.10: Average of the min-max norms error tolerance without quantization. Object volume : 86 cm^3.

regions. We evaluated both with the min-max norms quantization (Sec. 5.1.3) and without. Figure 5.10 shows the average min-max norms error tolerance without quantization. The graphs are similar to the center misalignment tolerance (Sec. 5.3.2) and same conclusions apply.

When we applied min-max norm quantization with our default parameters $Q_{norm} = 0.2/\sqrt[3]{86}$ and the object volume is 86 cm^3, which results in a step of 2 mm, we got a constant tolerance just below 1 mm.

Rotation misalignment

The introduction of angular subdivision in the method removed the invariance to rotation and brought the risk of errors in case of misalignment. Theoretically, the error introduced in the bin value will be proportional to the angular alignment error divided by S_a, meaning that the smaller the angular slices are, the more sensitive we get to error by rotation misalignment. To evaluate the angular tolerance to error, we selected eight rotation axes, including three base axes and five random axes, then increasingly chose misalignment angle and decoded the watermark for each angle and rotation axes. For each mesh and watermark value, we kept the biggest angle which produced no decoding error on any of the eight rotation axes. Figure 5.11 shows the average angle misalignment tolerance. We observe that increasing the number of angular bins makes the method more sensitive to angle error, but also that increasing the number of radial bins reduces the sensitivity to angle error until 128 bins and that we get higher tolerance with the invert flag than without. This can be explained by the increase of redundancy and the reduction of the correlation of wrongly aligned bins (Sec. 5.1.5).



(a) $N_b = 8$, with Invert flag

(b) $N_b = 16$, with Invert flag

(c) $N_b = 32$, with Invert flag

(e) $N_b = 8$, without Invert flag

(f) $N_b = 16$, without Invert flag

(g) $N_b = 32$, without Invert flag

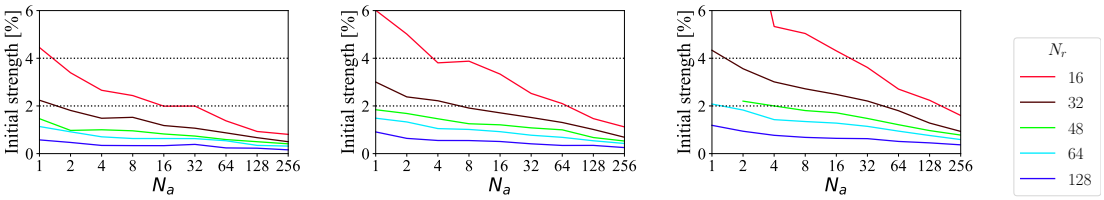
Figure 5.11: Mean rotation alignment error tolerance (higher is better). The graphs without Invert flag do not have curve for $N_r = 128$ because there was convergence problem when using this value or higher.

5.3.3 Initial bin value

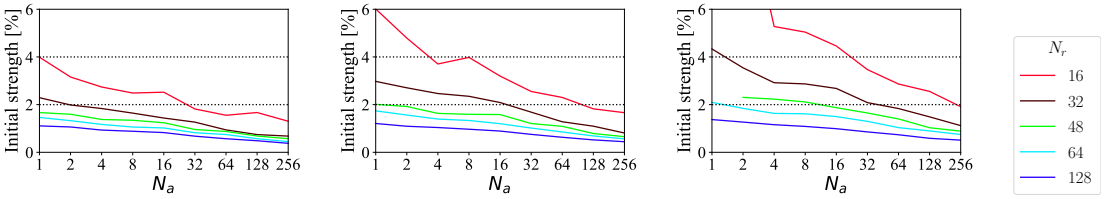
Before applying the watermarking optimization process, each bin already has an initial value. When this initial value is opposite to the one we need to encode, the watermarking optimization process needs to compensate for this value plus the additional encoding strength. A low initial strength makes the optimization process easier and reduces the surface distortion produced by the watermarking process. In this experiment, we evaluated the influence of the invert flag and the number of radial and angular bins on the average initial strength of the bins (Fig. 5.12). We observe that the average initial strength decreases when the number of radial or angular bins increases, and is also slightly lower with the invert flag than without.

5.3.4 Visibility evaluation

One of the goals of watermarking is to embed data in the model without being visible to the human eye. We evaluated it with the same metrics as the 3D digital



(a) $N_b = 8$, with Invert flag (b) $N_b = 16$, with Invert flag (c) $N_b = 32$, with Invert flag



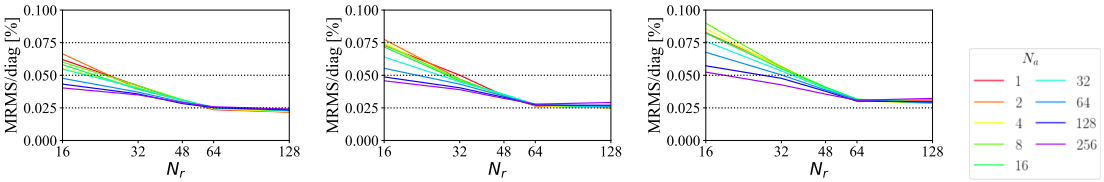
(e) $N_b = 8$, without Invert flag (f) $N_b = 16$, without Invert flag (g) $N_b = 32$, without Invert flag

Figure 5.12: Average initial strength of the bins (lower is better).

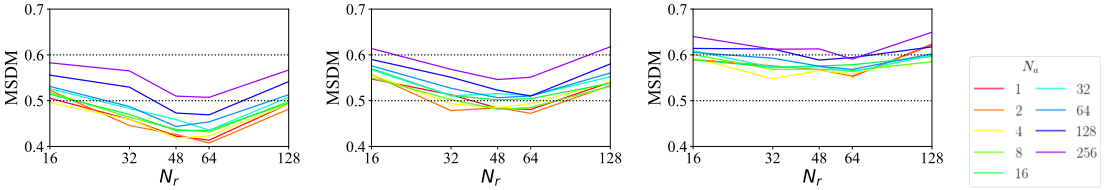
watermarking benchmark [75]: a geometrical metric called maximum root mean square error (MRMS) and a perceptual metric called mesh structural distortion measure (MSDM) [40]. Figure 5.13 shows the influence of the parameters on the MRMS and MSDM metrics. We observe that MRMS is decreasing when the number of radial bins increases until 64 bins then stays nearly constant. If the number of radial bins is low ($N_r < 48$), increasing the number of angular bins also decreases MRMS. MSDM increases when the number of angular bins increases, especially for value above 32 bins. MSDM produces a U-shaped curve for the number of radial bins, with an optimal value of around 48-64 bins. The difference between the two metrics can be explained by the fact that MRMS is only influenced by the amplitude of the distortions, whereas MSDM is also influenced by the frequency because high-frequency distortions are more visible to the human eye than lower ones.

5.3.5 3D print simulation

Printing and scanning 3D objects is a time-consuming process, it limits the number of experiments that can be done. Therefore, we developed a printing simula-



(a) MRMS/diag, $N_b = 8$ (b) MRMS/diag, $N_b = 16$ (c) MRMS/diag, $N_b = 32$



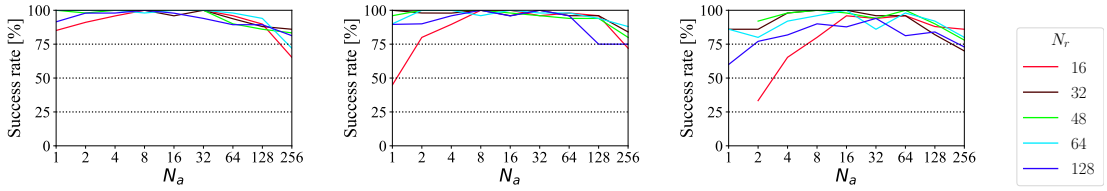
(e) MSDM, $N_b = 8$ (f) MSDM, $N_b = 16$ (g) MSDM, $N_b = 32$

Figure 5.13: Surface degradation evaluation using MRMS/diag and MSDM metrics (lower is better), with $\mathcal{S}_{enc} = 0.05$

tion process that takes the 'gcode' from the printer slicing software, simulates the print using the flow math proposed in [24, 27], generates a 2D Truncated Signed Distance Function (TSDF) for each layer, fills the holes inside the object, and generates the corresponding mesh with the layering artifacts using the Marching Cube Algorithm. This process simulates the quantization of the shape, the rounded borders of the layers, and the resampling of the mesh, which are the main artifacts produced by a real print. We used 0.3 mm layer height, which corresponds to the coarsest resolution generally used by consumer 3D printers. The result in Fig. 5.14 shows the influence of the number of radial and angular bins on the success rate. As expected, the performance decreases when the watermark length N_b increases because it reduces the redundancy. Our best range seems to be around [32,64] for N_r, and around [8,64] for N_a.

5.3.6 Real print-scan

To evaluate our method in real condition, we did a few real print-scan experiments using the dataset objects with 8 to 32 bits watermark. We removed the model 'crank' because our 3D scanner was not able to produce a complete scan due to



(a) $N_b = 8$ (b) $N_b = 16$ (c) $N_b = 32$

Figure 5.14: Average success rate of the watermark decoding process after printing simulation. A decoding is considered successful when all the extracted bits are correct.

the number of holes and occluded surfaces in the model. We used an 'original prusa I3 MK3S' with white PLA to print and an 'HP 3D structured light scanner pro S3' to scan. Because the print-scan is very time consuming, we only did one evaluation per model and watermark size, which gives a total of 27 objects. Based on the simulation results about misalignment resistance, visibility, and print simulation, we chose the parameters $N_r = 48$ and $N_a = 16$ for the real prints. A subset of our printed objects is illustrated in Fig. 5.15, and the bit error rates are reported in Fig 5.16. To measure the surface degradation caused by the print-scan process, we aligned the scanned meshes using iterative closest point (ICP) and computed the MRMS metric divided by the diagonal of the object. We observe that all our tested models except 'casting' succeeded up to 16 bits, and the error rate is still low at 32 bits. We also observe that a majority of our experiments produced a surface degradation (MRMS/diag) between 0.06% and 0.2%, which is higher than the $0.025 \sim 0.05\%$ degradation caused by the watermarking process (Sec. 5.3.4, Fig. 5.13). The model 'casting' is the one that produced the highest error rate in our evaluation. After doing a few more experiments with this model, we discovered that the scanning process generally produces high distortion because of the lack of features to align and fuse the multiple shots. By being really careful during the scan, we succeeded to decode it without error, but the process is much more difficult than for the other models. A higher-quality scanner would probably have less problems.

We also did a few more experiments to verify that our method also works with ABS instead of PLA and that it could resist to a reprint, *i.e.*, printing the model

Figure 5.15: Database models printed with 8 bits watermark

obtained after scanning a watermarked object, producing the double quantity of printing and scanning errors. We printed 'bunny' and 'venus' with 16 bits watermark using ABS and got no decoding error. We did a reprint of 'bunny' and 'venus' with 8 and 16 bits. We got a 1-bit error on 'bunny' with 16 bits, no error on the three other prints.

5.4 Discussion and Future Work

We presented a new blind watermarking method for 3d printed objects with low visibility, resisting to the printing and scanning process, and independent of the printing or scanning technology.

We solved the limitations that prevented to use the original vertex watermarking method in 3d printing context. We made it resistant to resampling by computing the histogram from all the surface instead of a discrete set of vertices and by computing the center of gravity with a moment-based method invariant to resampling. We introduced the angular subdivision to prevent the bins to become too thin in the radial direction, making them less sensible to local distortion and

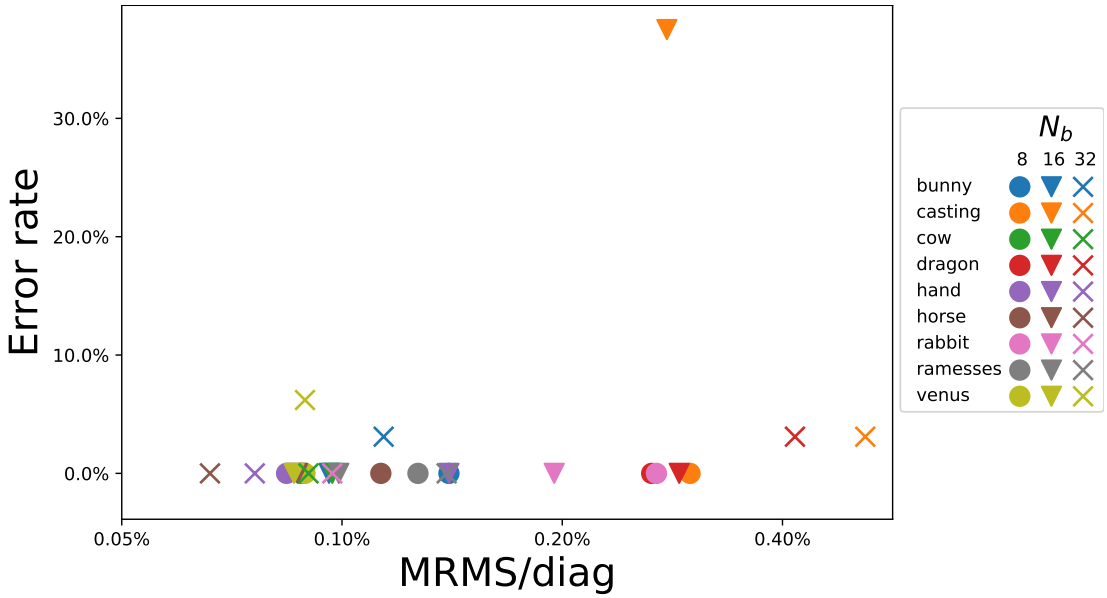


Figure 5.16: Decoding error rate for real print experiments. The horizontal axis represents the degradation caused by the print-scan process, measured by MRMS divided by the diagonal of the object.

misalignment. And we introduced a quantization for the minimal and maximal norms to prevent local printing artifacts to misalign the histogram.

We demonstrated that that our method could be applied successfully both in simulation and with real print-scan to multiple objects from a public database with watermark length up to 32 bits. We also demonstrated that a watermarked object could be scanned and reprinted, and the watermark could still be extracted from the reprinted model.

Our main limitation is the low tolerance for center position misalignment. With our default parameters ($N_r = 48$, $N_a = 16$) chosen for real print evaluation, the average tolerance is about 0.3mm for $N_b = 8$ and 0.2mm for $N_b = 16$ and $N_b = 32$, which is larger than most 3d printers resolution * but not by a big margin. For future works, we will try to find a way to resynchronize it via known markers or similar method to get higher tolerance.

When deployed for real life application, a different set of parameters than the

*Standard FDM printers have a precision around 0.1mm, and other technologies such as SLA are even higher precision

proposed one can be chosen for the algorithm, but it should be unique for a given application. Once it has been chosen, it should be the same for every object produced because blind watermarking prevents the extraction process to use any data from the model including non-unique parameters used for the encoding. If multiple actors want to use the same method, they should agree on the same parameter set or ensure that their application will be independent from each other. A more advanced method could choose the parameters based on some metrics than can be measured on the object itself. For example, 'Objects with a majority of flat surfaces' could have different parameters than 'Objects with rotational symmetry', as long as the rule to fix the parameters is strictly defined before the evaluation and the values can be obtained by only analyzing the printed object.

About the visibility evaluation, we could reach the MRMS threshold but not the MSDM threshold from the mesh watermarking benchmark. But the MSDM metric and especially the chosen threshold have been designed for mesh and not 3d printing context. We mainly used these metrics to influence our choice of optimal parameters, making a trade-off between resistance and visibility. In practice, the watermark had low visibility on our printed objects except on flat surfaces such as the sides of 'Ramesses' and the horizontal faces of 'Casting' and 'Crank' where the artifacts were more easy to see.

Finally, for the scenarios in which the original model is available at extraction time and that do not require the watermarking to be blind, refining the alignment with the original model could solve the misalignment problems that occur with the current algorithm and increase the robustness.

6 Blind watermarking for 3d printed objects by locally modifying layer thickness

In this chapter, we present a method that embeds a watermark by locally modifying the layer thickness of a 3D object printed with a FDM printer, and uses a common document scanner to extract it in a single shot. By only locally increasing or decreasing the thickness of the layers by a small amount, the modification is nearly invisible to human eye and does not deteriorate the function of mechanical parts. The extraction method is currently limited to flat surface due to the usage of a 2D document scanner for the extraction, but could be extended to curved surface by using a high-resolution 3D scanner.

The novelties of our method are as follows.

- the thickness of a printed layer is used as a one-dimensional (1D) carrier signal to embed data;
- distortion is minimized by only modifying the layer thickness locally; and
- the watermark is extracted in a single shot using a common paper scanner

This method is well suited for the crime investigation, stolen object identification and metadata embedding scenarios because the watermark has a good capacity, robustness, invisibility and extracted speed. On contrary, it is not really suited for copyright protection scenario because the watermark is embedded at too small scale compared to the resolution of most 3D scanners. If a counterfeiter scan the object and upload or reproduce it, the watermark would be destroyed during the process. However, it may become usable in the future if the resolution of the 3D scanners increases.

This chapter is organized as follows: In Section 6.1, we describe how to select the regions to watermark, how to encode the watermark bits into the printed layers, and how to control the printer to perform this task. In Section 6.2, we describe how to extract the watermark from a printed object using a paper scanner. In Section 6.3, we describe our error correction method and how multiple patches can be used to improve robustness to errors. In Section 6.4, we explain our experiments. Finally, in Section 6.5, we discuss our results and future work.

6.1 Watermark Embedding

To embed the watermark, we chose to locally modify the layer thickness because it allows the global curvature of the surface to be conserved while a pattern is embedded at high frequency. We performed modifications along the tangent of the surface instead of the normal, and thus caused less deformation to the shape. The layer thickness is a feature of the print that is typically constant and has low noise, which allowed us to obtain a high signal-to-noise ratio while maintaining low visibility. Fig. 6.1 illustrates the layer thickness modifications.

In what follows, we explain our system in detail. The pattern is explained in Section. 6.1.1, the selection of regions to watermark is explained in Section. 6.1.2 and the modification of the printer controls is explained in Section. 6.1.3.

6.1.1 Watermark pattern

To embed an N bit watermark, we first reshape the signal into an $H \times W$ matrix, where $HW = N$, and add a column and row of parity bits for error detection, which yields an $(H + 1) \times (W + 1)$ matrix.

Figure 6.1 illustrates the encoding of one row of the matrix by locally modifying the thickness of the two encoding layers. The layers are divided into equally sized encoding and separating regions for each bit. In the encoding regions of each bit, the thickness of the bottom layer is multiplied by $(1 + \alpha)$ or $(1 - \alpha)$ to encode a 1 or 0 bit, respectively. The top layer thickness is adjusted to keep the sum of the thickness of the two layers constant.

Figure 6.2 illustrates a 2×2 watermark, in which the encoding layers are separated by M separating layers. Both the separating regions and separating layers

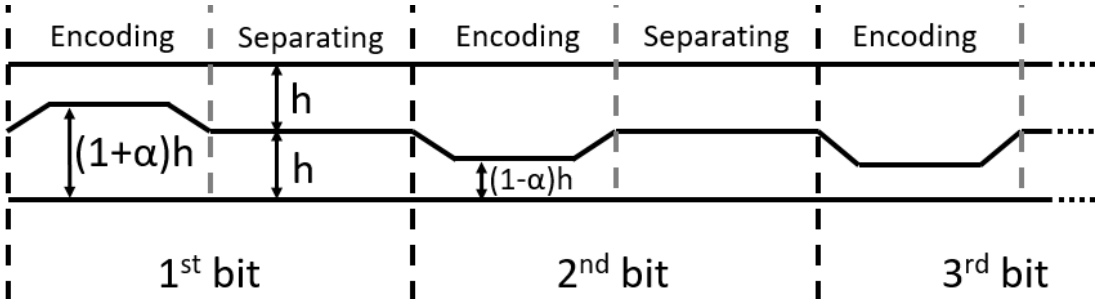


Figure 6.1: Encoding layer pattern. The pattern corresponds to two layers with variable thickness. We can encode a 1 or 0 bit by increasing or decreasing the thickness of the bottom layer in the encoding region, respectively. The top layer thickness is adjusted such that the sum of the thickness of the two layers remains equal to $2h$. The separating region is always of thickness h.

are used to simplify the detection and correlation at extraction time and reduce deformation by performing smooth transitions. To embed a watermark patch, the required number of layers is $2(H+1)$ encoding layers and HM separating layers, which is a total of $H(M+2)+2$ layers. The width of the pattern is equal to the width per bit bit_{width} multiplied by $(W+1)$.

In practice, we used $\alpha = 0.4$, $M = 2$ and $bit_{width} = 2.78$ mm. H and W must be even numbers to allow the parity check to detect whether all the bits have been inverted, which occurs when the pattern is rotated by 180° as illustrated in Fig. 6.3.

6.1.2 Watermark region selection

To find the candidate regions in which to embed the watermark, we analyze the sliced mesh produced by the printer software and find the surface regions composed of $H(M+2)+2$ layers with printed traces parallel to each other and whose width is $(W+1)bit_{width}$. Even if it is possible to embed the watermark in any sufficiently large surface, flat surfaces and smooth curved surfaces are preferred.

When multiple watermark patches are inserted close to each other, this requires a separation of at least $(M+1)$ layers if they are stacked above each other, or

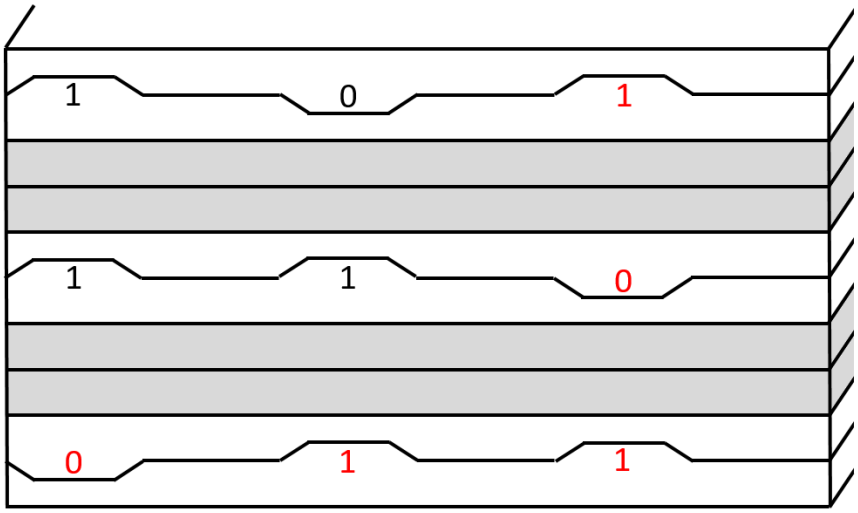


Figure 6.2: Example of the watermark pattern Encoding the value '1011' plus the parity bits. The digits correspond to the encoded bits; those in red are the parity bits. The layers with a white background are the encoding layers and those in gray are the separating layers.

at least bit_{width} if they are positioned side by side, to allow the detection of the patch boundaries. If possible, we also avoid inserting patches too close to the bottom layer or close to the border of the object because printing distortion and scanning errors tend to be more important there.

6.1.3 Printer control

Modifying the layer thickness requires the adjustment of the extruded plastic volume to keep the layer width constant. High-precision models have been developed [60] [82], but are complex to use. Therefore, we used the simplified model proposed by [27] [24].

The cross-sectional area of a layer is approximated by a rounded rectangle, as illustrated in Fig. 6.4, and can be calculated as

$$A_{layer} = h(w - h) + \pi \frac{h^2}{4}, \tag{6.1}$$

where w is the layer width and h is the layer height. We obtain the volume of plastic by multiplying the cross-sectional area by the length of the layer, and

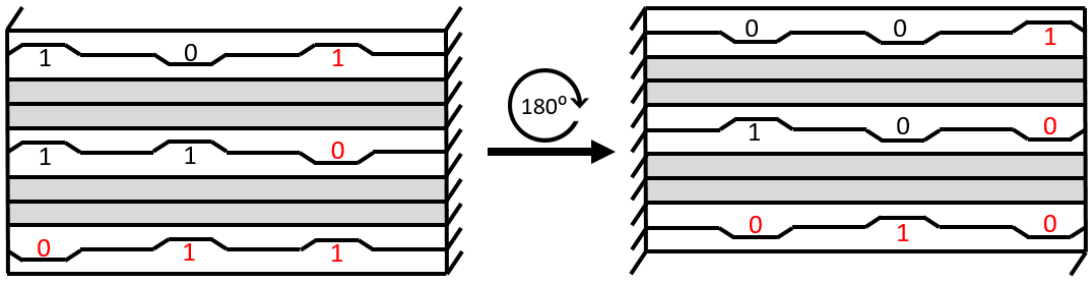


Figure 6.3: If the pattern is wrongly oriented by a rotation of 180°, all the parity bits obtain the wrong value. This property allows the orientation at extraction time to be retrieved.

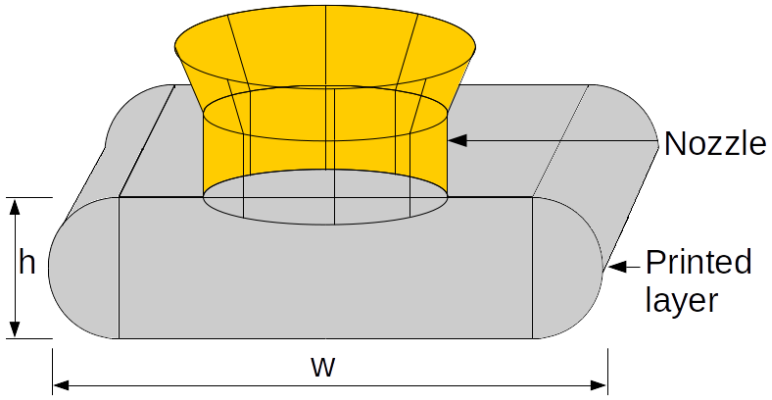


Figure 6.4: Cross-section of the nozzle and printed layer.

obtain the filament length by dividing the volume of plastic by the filament cross-sectional area:

$$L_{filament} = \frac{A_{layer} L_{layer}}{\pi(\varnothing_{filament}/2)^2}, \tag{6.2}$$

where $L_{filament}$ is the filament length, $\varnothing_{filament}$ is the filament diameter and L_{layer} is the length of the layer.

Adjusting the plastic extrusion is important because it reduces deformation on the surface, as shown in Fig.6.7. For the patches in which we modify the layer thickness, printing at high speed produces more artifacts; hence, we reduce the speed for these regions. In practice, we used approximately 8mm/s for the watermarked regions and 60mm/s for the other regions. There were still some

(a) (b) (c)

Figure 6.5: (a) Image of one face of the object, obtained from the 2D paper scanner. (b) Magnitude of the Fourier transform of image (a) after a high-pass filter is used to remove the low frequency. The line formed by the peak values is perpendicular to the orientation of the layers. (c) Realigned image after extracting the angle from image (b).

small deformations because of the approximations in our model and because the plastic flow could not change instantaneously because of nonlinearity in the liquefier, as explained in [8].

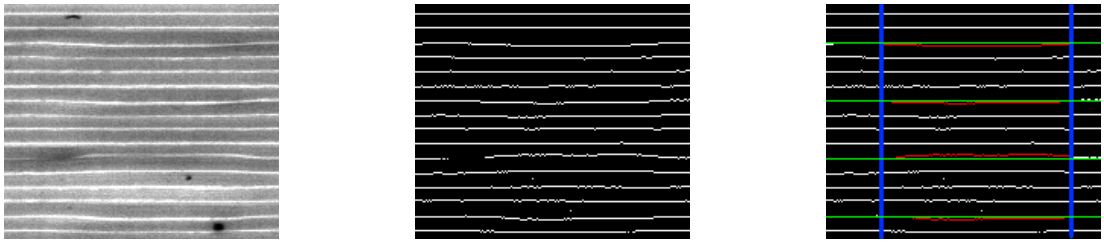
6.2 Watermark Extraction

Watermark extraction can be performed using multiple approaches. All that is required is to segment the different layers and correlate them with the encoding pattern to extract the data. The borders of each layer can be detected because of their rounded shape, which reflects light non-uniformly. This rounded shape is caused by the fluid dynamics of melted plastic, as illustrated in Fig. 6.4.

For flat surfaces, we can simply use a low-cost 2D paper scanner to extract the watermark. We need to adjust the gamma, brightness, and contrast to obtain a picture in which the layer edges are visible as shown in Fig. 6.6(a), these parameters depend on the plastic color and printer model.

6.2.1 Watermark localization

After obtaining the scanned image, we align the image so that the layer edges become horizontal using the Fourier transform. Specifically, we detect the peak



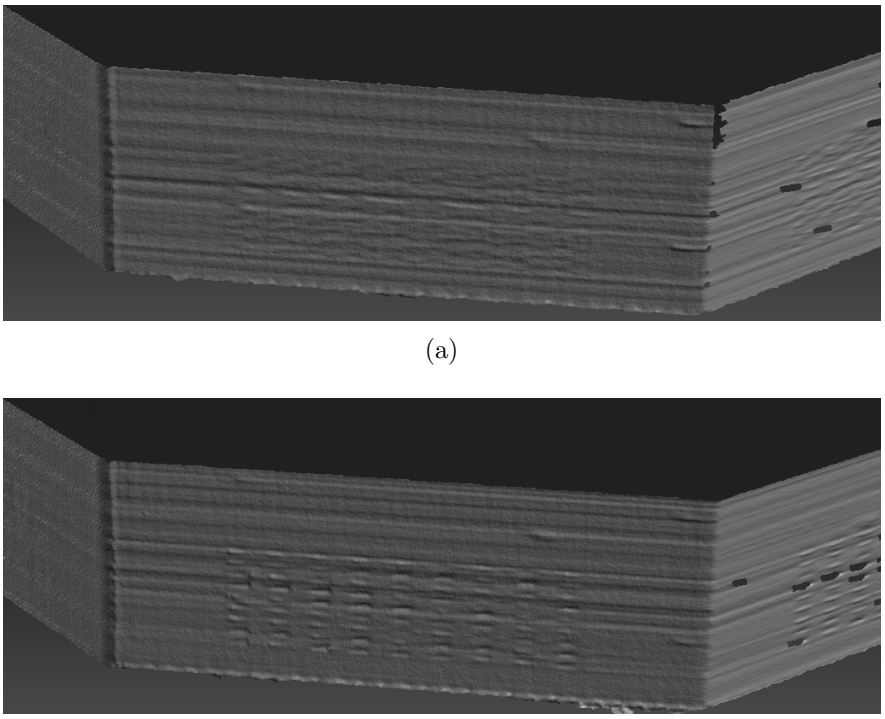
(a) (b) (c)

Figure 6.6: (a) Zoom in on the scanned image after reorientation. The highlights allow each layer to be segmented. (b) Edges extracted from image (a). (c) Edges with annotation: the blue vertical lines indicate the start and end of the encoding region, the red horizontal curved lines indicate the encoding region, and the green horizontal lines indicate the middle between the two other edges. We decode a 1 or 0 if the red curve is above or below the green line, respectively. From top to bottom, the extracted bits are 0,0,1,0.

magnitude value of the Fourier transform, compute the corresponding angle, and reorient the image. This process is illustrated in Fig. 6.5, and is similar to the frequency analysis section from [29]. If the layer thickness is known, then it is possible to restrict the search range for the peak magnitude and therefore be more resistant to potential errors. Otherwise, we calculate the thickness on the point with the highest magnitude in the Fourier transform.

Then we extract the highlighted lines that separate each layer using 1D non-maximum suppression on the columns of the image, with a neighborhood of a half-layer thickness. For example, a 0.2mm layer at 1200 dpi yields 0.1 mm * 47.244 pixel/mm = 4.72 pixels. Fig. 6.6(b) shows the result of the Non-Maximum Suppression applied to Fig. 6.6(a).

The edges of the separating layers can be easily detected because they are horizontal straight lines up to small distortions and the regularity of the pattern makes the distance between consecutive separating layers constant. The encoding layers can be distinguished from the separating layers because their encoding regions have a different thickness to their separating regions. The regularity of the pattern allows the watermark location to be determined precisely, even in the



(a)

(b)

Figure 6.7: 3D scan of a 64-bit watermarked object Scanned using an 'HP 3D structured light scanner pro S3': (a) with reduced speed for the watermark region and an adjustment of the plastic extrusion to the layer thickness; and (b) with reduced speed for the watermark region, but without an adjustment of the plastic extrusion, thereby maintaining the original plastic flow even when the thickness is modified.

presence of noise.

If multiple patches are detected on top of each other or side by side, then their alignment can be used as an additional constraint to improve the robustness of localization.

6.2.2 Watermark decoding

Once the exact location of the watermark patches has been determined, we recompute the edges more precisely using the constraints from the pattern and its location. We first recompute robustly the edges between the encoding lay-

ers and separating layers using RANSAC line detection with additional distance constraints on the neighboring parallel edges. Then, we recompute the edges between the encoding layers using 1D non-maximum suppression only in the range between the top half of the bottom encoding layer and the bottom half of the top encoding layer, as illustrated in Fig. 6.8. Restricting the searching region for these edges makes the decoding process much more robust to noise.

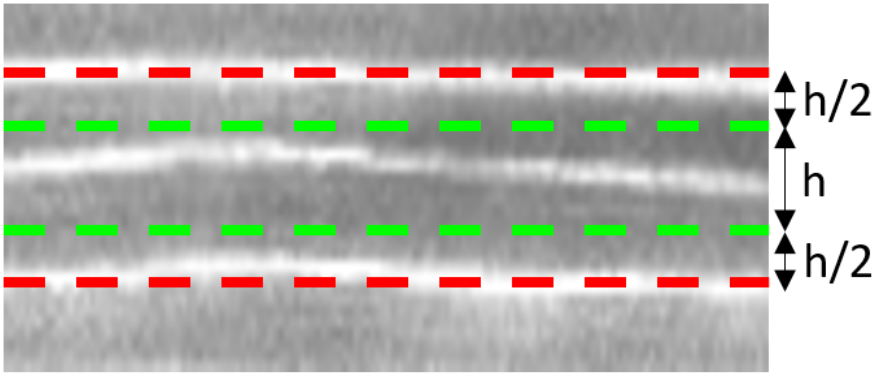


Figure 6.8: Region in which to apply non-maximum suppression (in green). The top and bottom edges, in red, are straight lines fitted with RANSAC along the patch. 1D non-maximum suppression is applied in the range between the green lines to find the edge between the two encoding layers.

After each edges has been robustly estimated, we can compute the value of each bit by measuring the average thickness of the layers in the encoding regions, as shown in Fig. 6.6(c),and verify the parity bits. If the majority of parity bits have the wrong value, this means that the image is probably wrongly oriented by 180° , as illustrated in Fig. 6.3, and needs to be rotated. Instead of rotating the image and restarting the extraction, it is possible to rotate the obtained watermark matrix by 180° and apply a 'NOT' operator on all the bits.

6.3 Error Correction

Each watermark patch contains 2D parity bits, also called two-dimensional parity check, or rectangular code in the literature [25] [5]. In addition to allowing the patch orientation to be determined, as explained in Section 6.1.2, the main use of these 2D parity bits is error detection.As illustrated in Fig. 6.9, they guarantee detection when up to 3 bits have the wrong value, and can often detect above 3-bits error. The parity checks detect each column or row that has an odd number of errors, but cannot detect them when there is an even number of errors. If a patch has an even number of errors in each column and row, similar to Fig. 6.9(e), it would be mistakenly considered as valid and produce a false positive. The probability of a false positive is low, and the easiest configuration for which this occurs is a 2×2 burst error, that is, a contiguous region that is undecodable because of local surface degradation. However, even in that case, if the 4 bits have random values, the risk of being undetected would be only 1/16.

Parity bits allow the detection and correction of single errors as shown in Fig. 6.9(b). However, at extraction time, the exact number of errors is unknown, and the parity check alone cannot guarantee that the error is unique and that the result is correct. For example, Fig. 6.9(d) shows a 3-bit error wrongly detected as a 1-bit error. For a 2×2 burst error, the probability of obtaining a false positive using this method is 5/16, which is high. Thus, instead, we classify the patch as invalid instead of taking the risk of obtaining a wrong value.

When multiple patches are extracted, it is possible to combine them to correct the errors. For each bit of the watermark, we can combine their values from multiple patches and apply a majority vote, which corrects the errors if the different patches have different error bit positions. When a patch has a low error rate, we can obtain high confidence on a bit value if both the column and row parity are correct, and, inversely, low confidence if both parities are wrong. By weighting the majority vote based on confidence, we can improve the correction rate. In our method, for each bit inside a patch, we assign a weight of 1 if both the column and row parity are correct, 0.5 if one of the parities is incorrect, and 0.25 if no parity is correct.

We simulated the success rate of our error correction methods on a 64-bit watermark with a uniformly distributed error at different rates and for different

numbers of patches. Fig. 6.10(a) shows the result with a majority vote without weighting and Fig. 6.10(b) shows the result with weighting. For the low error rates, we can observe an improvement, particularly for a small number of patches; however, for high error rates, we can also observe lower performance than without weighting. This occurs because the probability of undetected error lines, that is, the even number of errors in a line, increases with the error rate and makes weighting unreliable. To obtain the best results, we combine the weighted and non-weighted method: we apply the weighted method first, verify whether the solution is acceptable using the parity bits, and if not, apply the non-weighted method. The results of this combined method are shown in Fig. 6.10(c) and we can observe that it effectively obtained the best results of the two methods. As shown in Fig. 6.11, the false-positive rate, that is, where the result is wrong but all the parity checks are correct, is relatively low: below 0.7%. The false positive rates of the weighted and combined methods are slightly higher than that of the non-weighted method, but still sufficiently low to be acceptable. It occurs because undetected error lines are assigned higher weights, which increases the risk of an even number of errors on each line, particularly when there are only two patches. We ran a simulation with a burst error, thereby replacing a continuous portion of the watermark by random values. The modified region position was taken randomly for each patch and the shape was as close as possible to a square depending on the surface that was modified. Fig. 6.12 shows the success rate, and Fig. 6.13 shows the false-positive rate. The maximum false-positive rate was achieved when the error was a 2×2 burst error with a single patch, but was much smaller for other configurations.

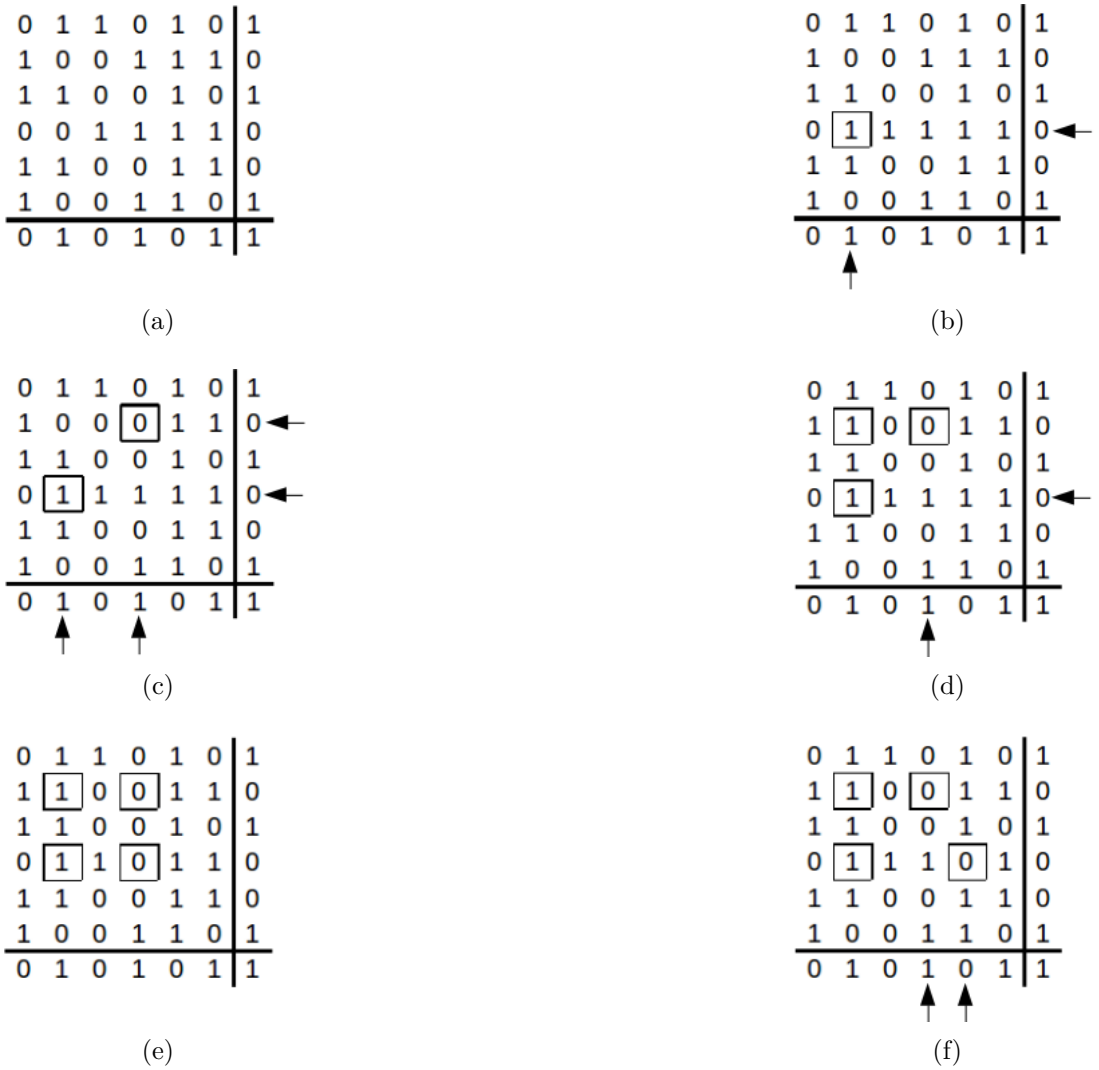
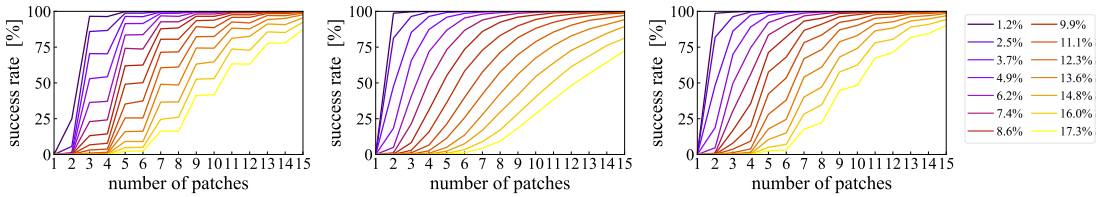
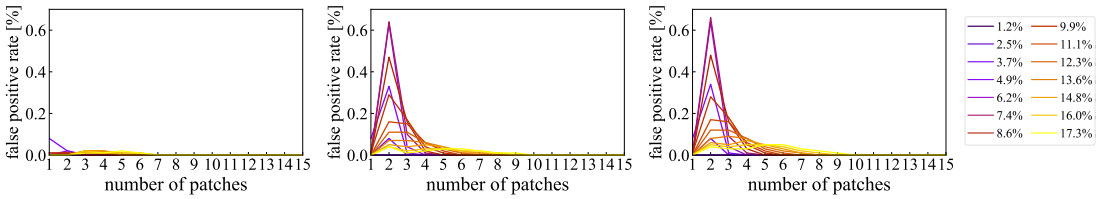


Figure 6.9: Example of a 2D parity check. For each table, the last column and row separated by a line contain the parity bits. Squares indicate the error bits and arrows indicate the columns or rows with parity errors. (a) Original signal, no errors; (b) 1-bit error, and its column and row have parity check errors; (c) 2-bit error, and parity errors detected; (d) 3-bit error, and only one column and row are detected; (e) 4-bit error, but no detection because the errors are aligned and conserve the parity; and (f) 4-bit error, and a different disposition than in (e), which makes two of the error columns detectable.



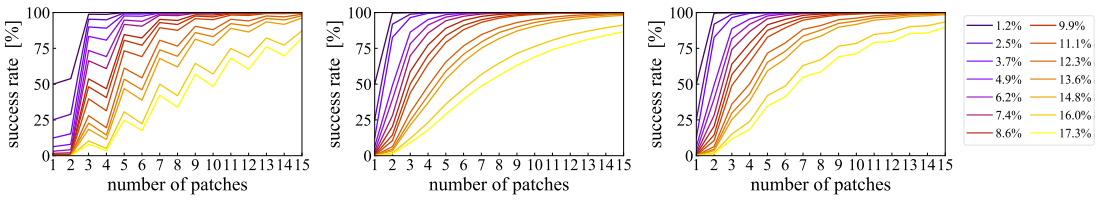
(a) (b) (c)

Figure 6.10: Success rate of the error correction methods depending on the number of patches used and the initial error rate, simulated on a 64-bit watermark with a uniform error distribution: (a) majority vote without weighting (b) majority vote with weighting; and (c) combination of (a) and (b).



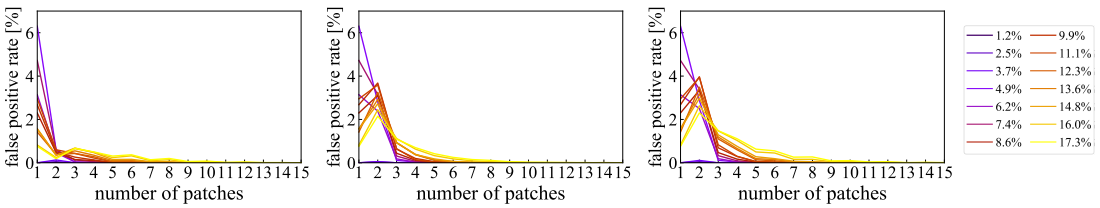
(a) (b) (c)

Figure 6.11: False-positive rate of the error correction methods depending on the number of patches used and the initial error rate, simulated on 64-bit watermark with a uniform error distribution: (a) majority vote without weighting (b) majority vote with weighting; and (c) combination of (a) and (b).



(a) (b) (c)

Figure 6.12: Success rate of the error correction methods depending on the number of patches used and the burst error region size, simulated on 64-bit watermark: (a) majority vote without weighting (b) majority vote with weighting; and (c) combination of (a) and (b).



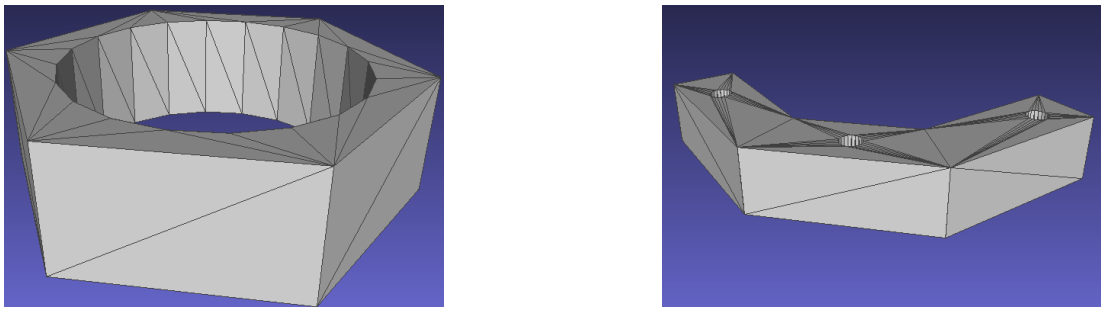
(a) (b) (c)

Figure 6.13: False-positive rate of the error correction methods depending on the number of patches used and the burst error region size, simulated on 64 bits watermark: (a) majority vote without weighting (b) majority vote with weighting; and (c) combination of (a) and (b).

6.4 Experiments

To embed a watermark on the surface of an object, we first generated a list of printer motor commands, that is, the 'G-code' or similar format, using printer software and recommended parameters. If the software included an option to generate variable layer thickness, this was disabled, at least for the regions in which we embedded a watermark patch. The watermark was then embedded by modifying the layer thickness of the patches in the 'G-Code' [63] and the object was printed.

We evaluated our method on two CAD models, illustrated in Fig. 6.14, which are referred to as the 'hexagonal' model and 'corner connector' model. In our experiments, we inserted three watermark patches on each of the six lateral faces of the 'hexagonal' model, and one watermark patch on each of the large lateral faces of the 'corner connector' model. Each watermark patch contained 64 bits of data and 17 parity bits. The surface used for one patch was $6.8 \times 25mm$, with $bit_{width} = 2.77mm$.



(a) (b)

Figure 6.14: CAD models used in our experiments (a) 'hexagonal' model and (b) 'corner connector' model.

For printing, we used an 'Original Prusa i3 MK3S,' with the G-code generated using 'PrusaSlicer' with the variable layer thickness option disabled, and applied our method to embed a watermark in the G-code. For scanning, we used a 'Canon PIXUS MG3630,' and controlled it using the open-source software 'XSane' on Linux, which gave us more control than the default Windows software. We used

Filament color	White	Yellow	Gray	Black	Red	Green	Blue	Orange
Scanning color mode	Grayscale							
Resolution	1200dpi							
Gamma	0.3	0.6	1.0	1.5	1.1			
Brightness	−30.0	30.0	30.0	100.0	50.0			
Contrast	100.0							

Table 6.1: Scanner parameters used for each PLA filament color.

the parameters in Table 6.1.

6.4.1 Robustness to variation of the signal amplitude

First, we evaluated the robustness of our method to multiple signal amplitudes α to find the value that produced the best result. We used 10%, 20%, 30%, and 40% amplitudes and printed one white 'hexagonal' object with 18 watermark patches for each amplitude value. The average bit error rates are shown in Fig. 6.15. We observe that above the 10% ratio, the error rate decreased significantly. For the 20% and 30% amplitudes, the error rate was below 1%, which was completely corrected by the error correction method. Because 40% did not produce any error in this evaluation, we used this value for subsequent experiments. The 20% and 30% amplitudes are interesting candidates to even further reduce distortion while maintaining a low error rate for contexts in which invisibility is a priority.

6.4.2 Robustness to the printing and scanning process with various filament colors

After choosing the value for the amplitude α, we evaluated the influence of the filament color to the extraction process. We printed the two CAD models using eight filament colors, which resulted in the 16 objects shown in Fig. 6.16, with 168 watermark patches in total. Each filament color required its own set of parameters for the scanner to obtain an image with clearly distinguishable layer edges. We

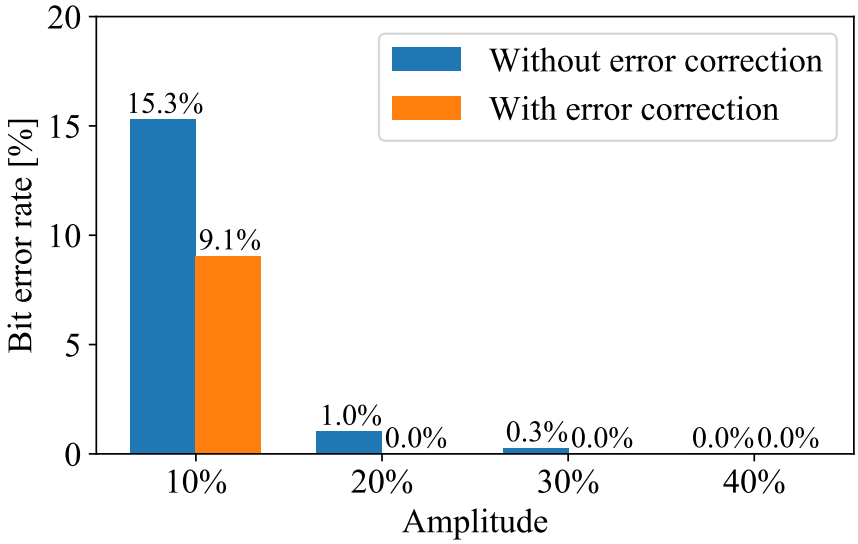


Figure 6.15: Bit error rate at multiple signal amplitudes Tested on the 'hexagonal' model. A total of 18 patches per amplitude value, with 64 bits of data and 17 parity bits per patch. The bit error rate was computed using the 81 bits of each patch, which yielded a total of 1458 bits per amplitude value. For the error correction, three patches from each face were merged. Starting from 20%, the error correction method recovered all the errors.

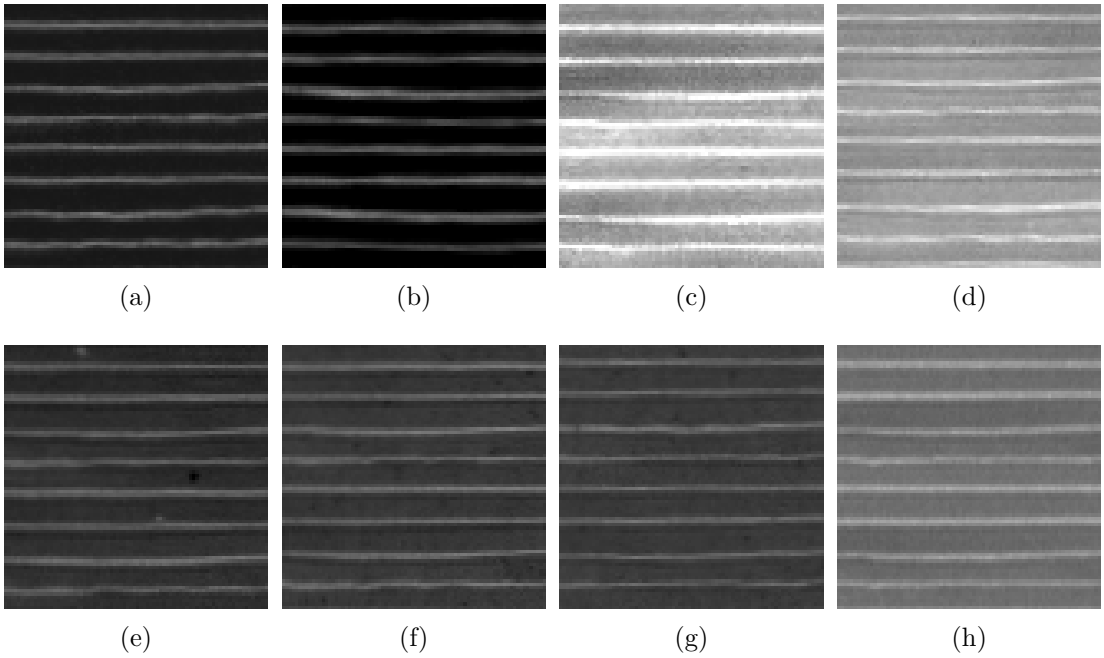
list the parameters used in our experiments in Table 6.1. As we observed in Fig. 6.17, the layer edges could be easily distinguished in the resulting image for every filament color. Among the 168 watermark patches, with 64 bits of data and 17 parity bits per patch, we obtained only a single bit of error when the method was used without error correction, which resulted in a bit error rate of 7.3×10^{-5}, or 9.3×10^{-5} if considering only the data bits. Zero errors remained when the method was used with error correction. This result shows that our method is robust to printing and scanning errors, and not sensitive to the variation of the filament color, provided the scanning parameters are adjusted appropriately. The robustness to color change can be explained by the fact that our detection method only depends on specularity, which is mainly independent of the color but depends on the material. New filament colors can be easily added by simply increasing or

decreasing the gamma and brightness parameters until the layer edges become easily distinguishable without saturation similar to Fig. 6.17. Once the scanning parameters for a filament have been chosen, they can be used for any object printed using this filament.

6.4.3 Robustness to surface degradation

In most real-life scenarios, the watermark would not be read immediately after printing, but after using the object for some time. The object would be manipulated and the surface could be damaged. To evaluate the resistance of our method to such damage, we sanded the surface of our objects and measured the error. In our experiment, we used the 'hexagonal' model printed with white PLA and signal amplitude $\alpha = 40\%$, and each face was manually sanded for 30 seconds with medium pressure using 400-grit and 240-grit sandpaper. Figure 6.18 shows the appearance of the surface after sanding. Even if the exact damage applied to the surface is difficult to quantify precisely, we can observe in Fig. 6.19 that the error rate increased with damage applied to the surface. The 400-grit experiment shows that our method was relatively well resistant to that amount of degradation and that our error correction method was effective in this scenario. The 240-grit experiment showed that even with important surface degradation, a large portion of the signal was still retrievable and error correction was still useful even if it was not able to completely correct all the errors.

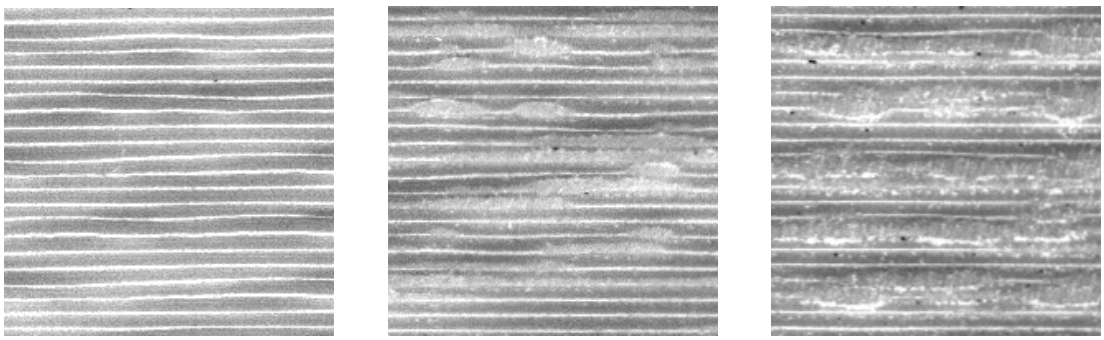
Figure 6.16: Objects used for color robustness evaluation Each object was printed with a PLA filament, 0.2mm layer thickness, and signal amplitude $\alpha = 40\%$.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6.17: Zoom in on the scanned images with multiple filament colors, using parameters from Table 6.1: (a) black PLA (b) gray PLA (c) white PLA (d) yellow PLA (e) red PLA (f) green PLA (g) blue PLA (h) orange PLA.



(a) (b) (c)

Figure 6.18: Scanned layers after manually sanding the surface (a) no sanding. (b) 400-grit sand paper, medium pressure for 30 seconds; and (c) 240-grit sand paper, medium pressure for 30 seconds.

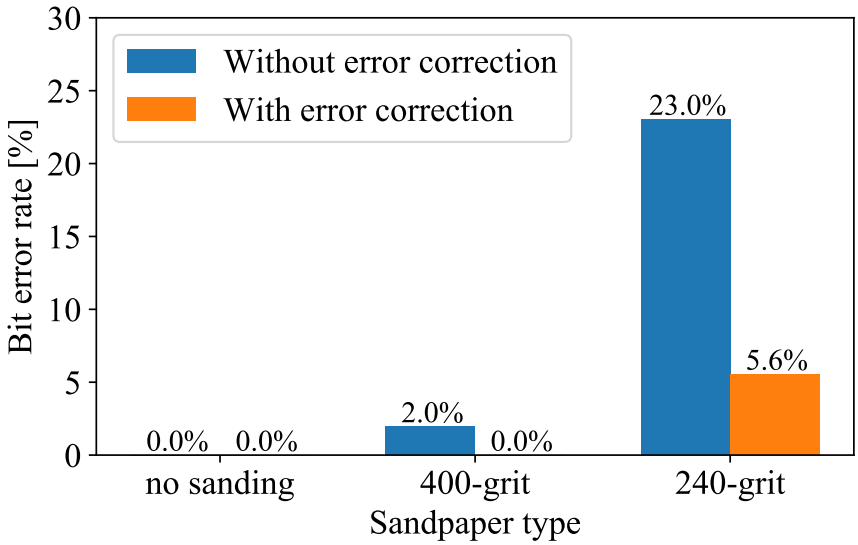


Figure 6.19: Bit error rate after manually sanding the surface. Tested on the 'hexagonal' model. A total of 18 patches per grit-size, with 64 bits of data and 17 parity bits per patch. The bit error rate was computed using the 81 bits of each patch, which yields a total of 1458 bits per grit size. Each face was sanded for 30 seconds using medium pressure with 400-grit and 240-grit sand paper.

6.5 Discussion and Future Work

Our method has numerous advantages because of its low shape deformation and low visibility while providing high data density, and allowing high redundancy and resistance to attacks such as cropping and degradation. For objects that have flat surfaces, which are common for CAD applications, our method can be applied with low cost and relatively standard equipment, which is convenient for real-life application. On a single patch, the extraction error rate is low and the method becomes extremely robust when we combine multiple patches to find and correct errors. The parity checks allow us to obtain a low false positive rate, which guarantees the correctness of the result. The error correction method provides a good trade-off between redundancy and patch size. Instead of including all redundancy inside one patch, similar to a QR code, we only included parity bits for orientation and error detection and created the redundancy by producing multiple identical patches. This made an individual patch smaller, which made it easier for us to find areas to embed the watermark on the surface of the object and reduce the risk of error caused by local surface degradation. Using multiple identical patches allowed us to adjust the redundancy to the available surface. We can add as many patches as can cover the available surface, and the decoding process can use all the available patches from one or multiple scans without additional constraints.

If required, other error-correcting methods can be used with our watermarking technique. Techniques such as the Reed-Solomon code [45] used in the QR code have a strong error correction capability for burst errors, but a 3D printer generally produces more random errors than burst errors, so it does not seem useful in our context. Techniques such as Hamming, Hadamard, and Reed–Muller codes [45] have a strong error correction capability for random errors, but increase the quantity of data required to insert in a patch, which makes patches larger and more difficult to embed.

In future work, we will focus on creating new extraction methods, improving the surface quality, and improving error correction performance.

For the extraction methods, our goal is to automatically adapt the scanning parameters to handle any colors and materials, including multicolored objects, to extract the watermark without contact with the surface, and to support curved

surfaces to make the method applicable to any object. Our encoding method is already usable with curved surfaces, and we printed the object shown in Fig.6.20 to test it. The distortion was similar to that of flat surfaces, but we observed that the artifacts were less visible because their specular reflections were not oriented in the same direction. This makes it possible for future work to further improve the imperceptibility of patches by embedding them in irregular surfaces. However, the remaining challenge with curved surfaces is the extraction of the watermark. Most commercial depth cameras, including our 'HP 3D structured light scanner pro S3', do not have sufficient resolution to distinguish and segment different layers reliably. If we succeed in segmenting the layers and recovering their 3D paths, it will be possible to parametrize from 3D to 2D and apply the extraction method to retrieve the watermark value.

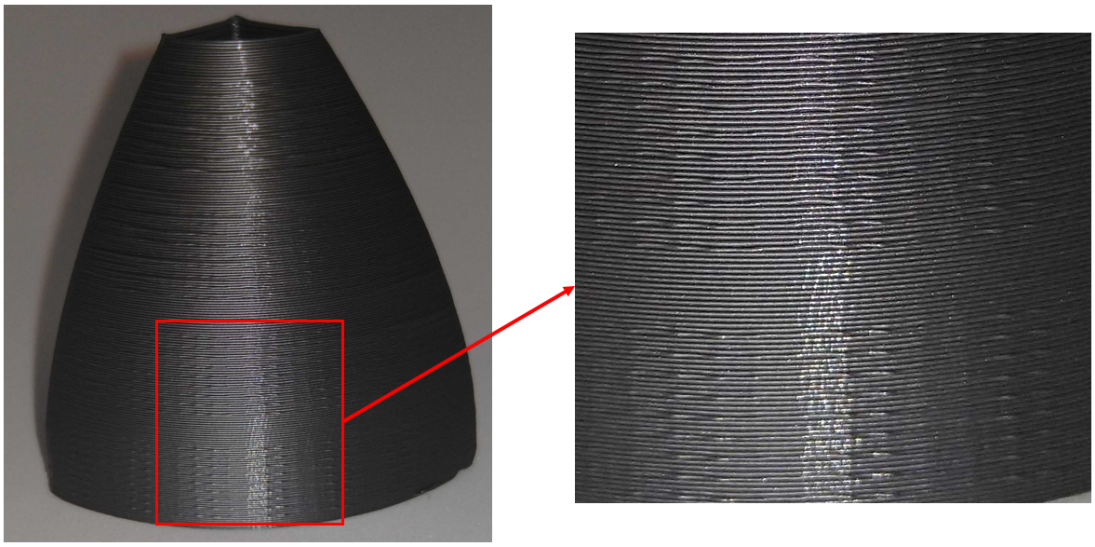


Figure 6.20: Watermark encoded on a curved surface printed with gray PLA.

Regarding surface quality, if we succeed in modeling the printing process more precisely, it may be possible to compensate for the remaining artifacts on watermark patches and reduce the visibility of watermark patches.

Regarding error correction, a good improvement would be to obtain a confidence score for each bit from the extraction method based on how clear or noisy the signal is, and use this score for weighting when combining multiple patches. Another improvement would be to add an interleaving method to prevent false

positives from the burst error.

Reliably detecting and decoding the watermark without contact would allow us to extend our method to other 3D applications, such as augmented reality (AR) and smart manufacturing. For AR, it could augment objects by replacing traditional markers [34] and be used for educating children [91] [72]. For smart manufacturing, it could replace RFID tags for individual part identification during the complete manufacturing process [89].

7 Conclusion

Blind watermarking for 3D printed objects is still a relatively new and challenging topic. The resampling, caused by the print-scan process, makes most of the method developed for blind mesh watermarking not usable for the 3D printing context. Therefore, new methods had to be developed specifically for this case. Some methods used a purely geometric approach, whereas others took advantage of the physical structure of the object (*e.g.*, ink, subsurface structure, material properties, layering). All the methods have to do a trade-off between multiple properties such as the invisibility, capacity, robustness, decoding procedure, decoding speed, and equipment required. The importance of each property is strongly dependent on the application scenario requirements, and it is preferable to fit strongly the requirements of a limited set of applications than meeting moderately a broader set.

Compared to watermarking in the digital domain, the distortions occurring during a print-scan are more difficult to analyze because they occur more randomly. Real-print experiments are also really time-consuming, with an average print taking a few hours, limiting the number of evaluations that can be done. For that reason, we introduced a new evaluation method using a printing simulation that reproduces the layering artifacts of FDM printing. This simulation is much faster than a real print and allowed us to evaluate much more extensively the parameters of our method before doing the real prints, sparing time and money. We focused on FDM for the simulation because it is the most commonly used 3D printing technology, and it is also the one that produces the strongest artifacts, which makes it great to evaluate the robustness of a method.

We proposed two blind watermarking methods for 3D printing objects, and both methods apply a small geometric modification on the surface of the object to encode the watermark. They have different advantages and drawbacks, as well

as different application scenarios, making them complementary. They have a set of default parameters, including the number of bits that can be embedded and the signal amplitude. These parameters have been chosen after evaluation to get a good trade-off between capacity, invisibility, and robustness, but it is naturally possible to use different values based on the requirements of each application.

Our first method uses the surface norm distribution to encode the watermark. It is one of the rare methods to be simultaneously blind, low visibility, detectable from a standard 3D scanner, and resistant to reprint. It can mainly be used for copyright protection and crime investigation scenarios. The method has been mainly developed and evaluated with artistic and visibility constraints, but could easily be used for functional parts. Because the data is spread all along the surface, we could add some constraints on the functional regions of the object. A lower signal amplitude would be applied to these regions, and a higher amplitude would be applied to the other non-constrained regions. It would allow to ensure the function is preserved.

Our second method uses the thickness of the printed layers as a 1D signal carrier. It has a high data density and low deformation. The low deformation is both an advantage and drawback in this method: It is good for invisibility and can still be extracted reliably with a document scanner, but it is too low amplitude to be detectable by most current 3D scanners and thus currently difficult to use in the copyright protection scenarios. Future improvement of the 3D scanning resolution could make it usable for these scenarios. It is well suited for metadata embedding because of high-capacity and fast decoding. It is currently more suited for functional parts than artistic parts because a document scanner can only extract the watermarks from flat surfaces, but it could be extended to any shape in the future using a high-resolution 3D scanner.

The two methods have a different approach to resist the degradation caused by printing artifacts, manipulation damage, and aging, resulting in lower surface quality and non-uniform deformations. By spreading the data on all the surface and using a weighted average to compute the result, the surface histogram method can resist local surface degradation but is sensitive to non-uniform scaling or any deformation that would significantly displace the center of gravity. On the contrary, the layer thickness method is not sensitive to non-uniform scaling.

It embeds multiple small watermark patches on the surface, and as long as at least one of these patches remain in good condition, the watermark is retrievable. The damages from aging depend on the type of plastic used for printing. For most plastics, as long as the objects are conserved indoor at room temperature, they should not suffer much damage by aging even after months or years. Common 3D printing plastics such as ABS and PETG are often used for mechanical applications and conserve their properties for a long time. However, a more aggressive environment with higher temperature, UV exposition or humidity, could damage the object and affect the watermark. For the temperature, it depends on the glass-transition temperature of the plastic, at which it starts to become viscous instead of solid. For PLA, this temperature is around 60°C and can be reached easily inside a car during summer, for example. ABS is more resistant, with a glass-transition temperature around 105°C, but would still be affected by UV light and humidity over time. Some advanced plastics such as PEEK would probably resist better to these difficult conditions, but require a more advanced printer and is also much more expensive.

For specific context, multiple techniques can be used simultaneously to improve the robustness in various situations. For example, a subsurface method would probably not interfere with a method modifying slightly the geometry of the surface or a method using invisible ink. An obvious watermark such as a barcode could also be used as a lure to make a criminal believe that he successfully removed the marker while a second imperceptible marker would still allow to trace him. Copyright protection scenarios could also benefit from having simultaneously an unclonable watermark for proof of authenticity and a robust watermark for proof of ownership, traitor tracing or 'no-copy' flag. Concerning our two methods, even though they are both geometric methods, they are based on different encoding principle that does not interfere with each other and could be combined with ink or subsurface methods.

Acknowledgements

First of all, I would like to thank Professor Yasuhiro Mukaigawa, for supporting my candidature to the MEXT scholarship, for welcoming me into his lab "Optical Media Interface", and for providing me advice and guidance through all my PhD. I would like to thank Professor Yuichi Hayashi for being on my thesis committee and for his constructive comments. I would like to thank Professor Takuya Funatomi, Professor Hiroyuki Kubo, and Professor Kenichiro Tanaka for being on my thesis committee and for the numerous advices and helpful comments they provided me during my stay in the lab. I would like to thank all the students of the lab for being welcoming and supportive despite the language barrier.

I would like to thank the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) that allowed me to pursue my PhD in Japan under the MEXT scholarship. I would like to thank the Nara Institute of Science and Technology for accepting me as PhD student, providing a lot of help for my life as foreigner in Japan. I would like to thank all the teachers and volunteer tutors that helped me to progress in Japanese language and communicate better in my daily life in Japan.

A part of this work is supported by JSPS KAKEN JP17K19979.

I'd like to thank Professor Christophe De Vleeschouwer that was my supervisor during my master thesis and during my 2 years as research assistant at UCLouvain in Belgium before coming to Japan. He taught me to computer vision and the knowledge I acquired with him has been useful during my phD.

Finally, I would like to thank all my previous teachers for helping me to obtain the knowledge and skills required for doing research, my girlfriend for all her love and support, and my family and friends for their support and encouragements during all my life.

References

[1] Overview of 3d printing & intellectual property law, Sep 2016.

[2] 3dHubs. Online manufacturing trends q4-2018. [Online] Available: `https://www.3dhubs.com/trends`, 2018. Accessed: 2019-12-18.

[3] 3dHubs. Online manufacturing trends q1-2019. [Online] Available: `https://www.3dhubs.com/trends`, 2019. Accessed: 2019-12-18.

[4] José Manuel Amigo, Hamid Babamoradi, and Saioa Elcoroaristizabal. Hyperspectral image analysis. a tutorial. *Analytica Chimica Acta*, 896:34–51, 2015.

[5] Romi Banerjee, Saptarshi Naskar, and Samar Sen Sarma. A closer look into the rectangular codes: Where the parity check predominates. *Journal of Global Research in Computer Science*, 3(3):17–23, 2012.

[6] Chris Baraniuk. Why printers add secret tracking dots. [Online] Available: `http://www.bbc.com/future/story/20170607-why-printers-add-secret-tracking-dots`, June 2017. Accessed: 2019-03-17.

[7] Georg T Becker. The gap between promise and reality: On the insecurity of xor arbiter pufs. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 535–555. Springer, 2015.

[8] Anna Bellini, Selcuk Guceri, and Maurizio Bertoldi. Liquefier dynamics in fused deposition. *Journal of Manufacturing Science and Engineering*, 126(2):237–246, 2004.

[9] Nirveek Bhattacharjee, Arturo Urrios, Shawn Kang, and Albert Folch. The upcoming 3d-printing revolution in microfluidics. *Lab on a Chip*, 16(10):1720–1742, 2016.

[10] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. Collaboration with tangible augmented reality interfaces. In *HCI international*, volume 1, pages 5–10, 2001.

[11] Leonid Bolotnyy and Gabriel Robins. Physically unclonable function-based security and privacy in rfid systems. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07)*, pages 211–220. IEEE, 2007.

[12] François Cayre and Benoit Macq. Data hiding on 3-d triangle meshes. *IEEE Transactions on signal Processing*, 51(4):939–949, 2003.

[13] Namita Chandrakar, Jaspal Bagga, et al. Performance comparison of digital image watermarking techniques: a survey. *International Journal of Computer Applications Technology and Research*, 2(2):126–130, 2013.

[14] Pei-Ju Chiang, Nitin Khanna, Aravind K Mikkilineni, Maria V Ortiz Segovia, Sungjoo Suh, Jan P Allebach, George T-C Chiu, and Edward J Delp. Printer and scanner forensics. *IEEE Signal Processing Magazine*, 26(2):72–83, 2009.

[15] Jae-Won Cho, Min-Su Kim, Rémy Prost, Hyun-Yeol Chung, and Ho-Youl Jung. Robust watermarking on polygonal meshes using distribution of vertex norms. In *International Workshop on Digital Watermarking*, pages 283–293. Springer, 2004.

[16] Jae-Won Cho, Rmy 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, 2007.

[17] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. Meshlab: an open-source mesh processing tool. In *Eurographics Italian chapter conference*, volume 2008, pages 129–136, 2008.

[18] Wikimedia Commons. File:selective laser melting system schematic.jpg — wikimedia commons, the free media repository, 2019. [Online; accessed 18-December-2019].

[19] Arnaud Delmotte, Kenichiro Tanaka, Hiroyuki Kubo, Takuya Funatomi, and Yasuhiro Mukaigawa. Blind watermarking for 3-d printed objects using surface norm distribution. In *2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 282–288. IEEE, 2018.

[20] Srinivas Devadas, Edward Suh, Sid Paral, Richard Sowell, Tom Ziola, and Vivek Khandelwal. Design and implementation of puf-based" unclonable" rfid ics for anti-counterfeiting and security applications. In *2008 IEEE international conference on RFID*, pages 58–64. IEEE, 2008.

[21] EFF. Investigating machine identification code technology in color laser printers. [Online] Available: `http://www.eff.org/wp/investigating-machine-identification-code-technology-color-laser-printers`, July 2005. Accessed: 2019-03-17.

[22] Amelia McDow Elliott. *The Effects of Quantum Dot Nanoparticles on Polyjet Direct 3D Printing Process*. PhD thesis, Virginia Tech, 2014.

[23] James Gooch, Barbara Daniel, Vincenzo Abbate, and Nunzianda Frascione. Taggant materials in forensic science: A review. *TrAC Trends in Analytical Chemistry*, 83:49–54, 2016.

[24] Gabriel Pieter Greeff and Meinhard Schilling. Closed loop control of slippage during filament transport in molten material extrusion. *Additive Manufacturing*, 14:31–38, 2017.

[25] R. W. Hamming. *Coding and Information Theory*. Prentice Hall, U.S.A, 2 edition, 1986.

[26] Chris Harrison, Robert Xiao, and Scott Hudson. Acoustic barcodes: passive, durable and inexpensive notched identification tags. In *Proceedings of the*

25th annual ACM symposium on User interface software and technology, pages 563–568. ACM, 2012.

[27] G Hodgson, A Ranellucci, and J Moe. Slic3r manual: Flow math. [Online] `http://manual.slic3r.org/advanced/flow-math`, 2011.

[28] Jong-Uk Hou, Do-Gon Kim, Sunghee Choi, and Heung-Kyu Lee. 3d print-scan resilient watermarking using a histogram-based circular shift coding structure. In *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pages 115–121. ACM, 2015.

[29] Jong-Uk Hou, Do-Gon Kim, and Heung-Kyu Lee. Blind 3d mesh watermarking for 3d printed model by analyzing layering artifact. *IEEE Transactions on Information Forensics and Security*, 12(11):2712–2725, 2017.

[30] Jong-Uk Hou, Dongkyu Kim, Won-Hyuk Ahn, and Heung-Kyu Lee. Copyright protections of digital content in the age of 3d printer: Emerging issues and survey. *IEEE Access*, 6:44082–44093, 2018.

[31] Roland Hu, Patrice Rondao-Alface, and Benoit Macq. Constrained optimisation of 3d polygonal mesh watermarking by quadratic programming. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1501–1504. IEEE, 2009.

[32] O Ivanova, A Elliott, T Campbell, and CB Williams. Unclonable security features for additive manufacturing. *Additive Manufacturing*, 1:24–31, 2014.

[33] Ruiqi Jiang, Hang Zhou, Weiming Zhang, and Nenghai Yu. Reversible data hiding in encrypted three-dimensional mesh models. *IEEE Transactions on Multimedia*, 20(1):55–67, 2017.

[34] Anuroop Katiyar, Karan Kalra, and Chetan Garg. Marker based augmented reality. *Advances in Computer Science and Information Technology (ACSIT)*, 2(5):441–445, 2015.

[35] Reza Kazemi, Fernando Pérez-González, Mohammad Ali Akhaee, and Fereydoon Behnia. Data hiding robust to mobile communication vocoders. *IEEE Transactions on Multimedia*, 18(12):2345–2357, 2016.

[36] 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, 2008.

[37] Tyler Koslow. 3d printing on trial: Just 3d print brings new lawsuit against 3dr holdings, Jan 2018.

[38] Brian Krebs. Krebs on security : all about skimmers, 2010.

[39] Abhishek Kumar, Naveen Prakash Goel, and Mayur Hemani. Method and apparatus for storing and retrieving data embedded into the surface of a 3d printed object, July 26 2016. US Patent 9,400,910.

[40] 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*, volume 6312, page 63120L. International Society for Optics and Photonics, 2006.

[41] Herwig Lejsek, Ársæll Þ Jóhannsson, Friðrik H Ásmundsson, Björn Þ Jónsson, Kristleifur Daðason, and Laurent Amsaleg. Videntifier™ forensic: a new law enforcement service for automatic identification of illegal video material. In *Proceedings of the First ACM workshop on Multimedia in forensics*, pages 19–24. ACM, 2009.

[42] Dingzeyu Li, David IW Levin, Wojciech Matusik, and Changxi Zheng. Acoustic voxels: computational optimization of modular acoustic filters. *ACM Transactions on Graphics (TOG)*, 35(4):88, 2016.

[43] Dingzeyu Li, Avinash S Nair, Shree K Nayar, and Changxi Zheng. Aircode: Unobtrusive physical tags for digital fabrication. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 449–460. ACM, 2017.

[44] Zhengxiong Li, Aditya Singh Rathore, Chen Song, Sheng Wei, Yanzhi Wang, and Wenyao Xu. Printracker: fingerprinting 3d printers using commodity scanners. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1306–1323. ACM, 2018.

[45] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*, volume 16. Elsevier, 1977.

[46] Henrique Teles Maia, Dingzeyu Li, Yuan Yang, and Changxi Zheng. Layercode: Optical barcodes for 3d printed shapes. *ACM Trans. Graph.*, 38(4):112:1–112:14, July 2019.

[47] Manisha Mann, S Shukla, and Shruti Gupta. A comparative study on security features of banknotes of various countries. *Int. J. Multidiscip. Res. Dev*, 2:83–91, 2015.

[48] Nassima Medimegh, Samir Belaid, and Naoufel Werghi. A survey of the 3d triangular mesh watermarking techniques. *International Journal of Multimedia*, 1(1), 2015.

[49] Dominic Milano. Content control: Digital watermarking and fingerprinting. *White Paper, Rhozet, a business unit of Harmonic Inc., http://www. rhozet. com/whitepapers/Fingerprinting—Watermarking. pdf, Last accessed May*, 30, 2012.

[50] Michael Molitch-Hou. The future of hp's multi jet fusion 3d printing. [Online] Available: `https://www.engineering.com/3DPrinting/3DPrintingArticles/ArticleID/12298/The-Future-of-HPs-Multi-Jet-Fusion-3D-Printing.aspx`, June 2016. Accessed: 2019-05-21.

[51] Mariella Moon. Anyone can now print out all tsa master keys, Jul 2019.

[52] Stefanie Mueller, Martin Fritzsche, Jan Kossmann, Maximilian Schneider, Jonathan Striebel, and Patrick Baudisch. Scotty: Relocating physical objects across distances using destructive scanning, encryption, and 3d printing. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 233–240. ACM, 2015.

[53] Javier Nieves, Igor Ruiz-Agundez, and Pablo G Bringas. Recognizing banknote patterns for protecting economic transactions. In *2010 Workshops on Database and Expert Systems Applications*, pages 247–249. IEEE, 2010.

[54] Masoud Nosrati, Ronak Karimi, and Mehdi Hariri. Audio steganography: a survey on recent approaches. *world applied programming*, 2(3):202–205, 2012.

[55] International Chamber of Commerce. The economic impacts of counterfeiting and piracy, 2016.

[56] 1 Ryutarou Ohbuchi, 1 Akio Mukaiyama, and 2 Shigeo Takahashi. A frequency-domain approach to watermarking 3d shapes. In *Computer Graphics Forum*, volume 21, pages 373–382. Wiley Online Library, 2002.

[57] Ryutarou Ohbuchi, Hiroshi Masuda, and Masaki Aono. Data embedding algorithms for geometrical and non-geometrical targets in three-dimensional polygonal models. *Computer Communications*, 21(15):1344–1354, 1998.

[58] Ryutarou Ohbuchi, Hiroshi Masuda, and Masaki Aono. Watermarking three-dimensional polygonal models through geometric and topological modifications. *IEEE Journal on selected areas in communications*, 16(4):551–560, 1998.

[59] Ayumi Okada, Piyarat Silapasuphakornwong, Masahiro Suzuki, Hideyuki Torii, Youichi Takashima, and Kazutake Uehira. Non-destructively reading out information embedded inside real objects by using far-infrared light. In *Applications of Digital Image Processing XXXVIII*, volume 9599, page 95992V. International Society for Optics and Photonics, 2015.

[60] Ashutosh Pandey and Sharad K Pradhan. Investigations into complete liquefier dynamics and optimization of process parameters for fused deposition modeling. *Materials Today: Proceedings*, 5(5):12940–12955, 2018.

[61] Swati Patel, Anilkumar Katharotiya, and Mahesh Goyani. A survey on digital video watermarking. *International Journal Comp. Tech. Appl*, 2(6):3015–3018, 2011.

[62] Vidyasagar M Potdar, Song Han, and Elizabeth Chang. A survey of digital image watermarking techniques. In *INDIN'05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, pages 709–716. IEEE, 2005.

[63] Reprap. Reprap g-code. [Online] Available: `https://reprap.org/wiki/G-code`. Accessed: 2019-06-11.

[64] M Rizwan, MWA Khan, H He, J Virkki, L Sydänheimo, and L Ukkonen. Flexible and stretchable 3d printed passive uhf rfid tag. *Electronics Letters*, 53(15):1054–1056, 2017.

[65] Douglas C Sicker, Paul Ohm, and Shannon Gunaji. The analog hole and the price of music: An empirical study. *J. on Telecomm. & High Tech. L.*, 5:573, 2006.

[66] Stilgherrian. Fear of downloadable guns becoming a reality, Jul 2017.

[67] Thomas Stütz, Florent Autrusseau, and Andreas Uhl. Non-blind structure-preserving substitution watermarking of h. 264/cavlc inter-frames. *IEEE Transactions on Multimedia*, 16(5):1337–1349, 2014.

[68] Masahiro Suzuki, Pailin Dechrueng, Soravit Techavichian, Piyarat Silapasuphakornwong, Hideyuki Torii, and Kazutake Uehira. Embedding information into objects fabricated with 3-d printers by forming fine cavities inside them. *Electronic Imaging*, 2017(7):6–9, 2017.

[69] Masahiro Suzuki, Tomohisa Matumoto, Youichi Takashima, Hideyuki Torii, and Kazutake Uehira. Information hiding inside 3-d printed objects by forming high reflectance projections. In *Proceedings of the International Conference on Video and Image Processing*, pages 146–150. ACM, 2017.

[70] Masahiro Suzuki, Piyarat Silapasuphakornwong, Y Takashima, H Torii, H Unno, and K Uehira. Technique for protecting copyrights of digital data for 3-d printing, and its application to low infill density objects. In *The Eighth International Conferences on Advances in Multimedia*, pages 56–59, 2016.

[71] Carlos Tejada, Osamu Fujimoto, Zhiyuan Li, and Daniel Ashbrook. Blowhole: Blowing-activated tags for interactive 3d-printed models. In *Proc. Graphics Interface*, pages 131–137, 2018.

[72] Monica Tentori, Lizbeth Escobedo, and Gabriela Balderas. A smart environment for children with autism. *IEEE Pervasive Computing*, 14(2):42–50, 2015.

[73] Jeffrey Voris, Benjamin Foster Christen, Jorge Alted, and David W Crawford. Three dimensional (3d) printed objects with embedded identification (id) elements, May 23 2017. US Patent 9,656,428.

[74] 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, 2008.

[75] Kai Wang, Guillaume Lavoué, Florence Denis, Atilla Baskurt, and Xiyan He. A benchmark for 3d mesh watermarking. In *2010 Shape Modeling International Conference*, pages 231–235. IEEE, 2010.

[76] Joseph Ying Sen Wee, Christopher Ian Byatte, Anthony David George Rhoades, and David Leslie McNeight. Objets de vertu, December 10 2015. US Patent App. 14/485,880.

[77] Joseph Ying Sen Wee, Christopher Ian Byatte, Anthony David George Rhoades, and David Leslie McNeight. Product authentication, September 3 2015. US Patent App. 14/250,533.

[78] Michael Weinberg. *It will be Awesome if They don't Screw it up: 3D Printing, Intellectual Property, and the Fight Over the Next Great Disruptive Technology*. Institute For Emerging Innovation, Public Knowledge., 2010.

[79] Michael Weinberg. *What's the Deal with Coypright and 3d Printing?*. Institute For Emerging Innovation, Public Knowledge, 2013.

[80] Wikipedia contributors. Eurion constellation — Wikipedia, the free encyclopedia, 2019. [Online; accessed 30-November-2019].

[81] Karl DD Willis and Andrew D Wilson. Infrastructs: fabricating information inside physical objects for imaging in the terahertz region. *ACM Transactions on Graphics (TOG)*, 32(4):138, 2013.

[82] Huanxiong Xia, Jiacai Lu, Sadegh Dabiri, and Gretar Tryggvason. Fully resolved numerical simulations of fused deposition modeling. part i: fluid flow. *Rapid Prototyping Journal*, 24(2):463–476, 2018.

[83] Hiroshi Yamamoto and Kazuki Sano. A watermarking method for embedding into the external shapes of objects. In *2018 International Symposium on Information Theory and Its Applications (ISITA)*, pages 321–325. IEEE, 2018.

[84] Shuntaro Yamazaki, Satoshi Kagami, and Masaaki Mochimaru. Extracting watermark from 3d prints. In *2014 22nd International Conference on Pattern Recognition*, pages 4576–4581. IEEE, 2014.

[85] Bulent Yusuf. Digitally augmented additive manufacturing parts from rize. [Online] Available: `https://all3dp.com/digitally-augmented-additive-manufacturing-parts-from-rize`, April 2018. Accessed: 2019-04-24.

[86] Cha Zhang and Tsuhan Chen. Efficient feature extraction for 2d/3d objects in mesh representation. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 935–938. IEEE, 2001.

[87] Weiming Zhang, Hui Wang, Dongdong Hou, and Nenghai Yu. Reversible data hiding in encrypted images by reversible image transformation. *IEEE Transactions on multimedia*, 18(8):1469–1479, 2016.

[88] Xin Zhang, Qian Wang, and Ioannis Ivrissimtzis. Single image watermark retrieval from 3d printed surfaces via convolutional neural networks. 09 2018.

[89] Li Zhekun, Rajit Gadh, and BS Prabhu. Applications of rfid technology and smart parts in manufacturing. In *Proceedings of DETC*, volume 4, pages 1–7. Citeseer, 2004.

[90] Xinshan Zhu, Jie Ding, Honghui Dong, Kongfa Hu, and Xiaobin Zhang. Normalized correlation-based quantization modulation for robust watermarking. *IEEE Transactions on Multimedia*, 16(7):1888–1904, 2014.

[91] Yujie Zhu and Stephen Jia Wang. A tangible augmented reality toy kit: Interactive solution for early childhood education. In *Interactivity, Game Creation, Design, Learning, and Innovation*, pages 12–19. Springer, 2016.

Publication List

Journal Paper

1. Arnaud Delmotte, Kenichiro Tanaka, Hiroyuki Kubo, Takuya Funatomi and Yasuhiro Mukaigawa, Blind watermarking for 3d printed objects by locally modifying layer thickness, IEEE Transaction On Multimedia (ACCEPTED, pre-published)

2. Arnaud Delmotte, Kenichiro Tanaka, Hiroyuki Kubo, Takuya Funatomi and Yasuhiro Mukaigawa, Blind 3d printing watermarking using moment alignment and surface norms distribution, IEEE Transaction On Multimedia (submitted, under review)

International conference (Reviewed)

1. Arnaud Delmotte, Kenichiro Tanaka, Hiroyuki Kubo, Takuya Funatomi and Yasuhiro Mukaigawa, Blind Watermarking for 3-D Printed Objects using Surface Norm Distribution, 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR). IEEE, 2018. p. 282-288.

Domestic conference (Reviewed)

1. Arnaud Delmotte, Kenichiro Tanaka, Hiroyuki Kubo, Takuya Funatomi and Yasuhiro Mukaigawa, Blind watermarking for 3d printed objects by modifying layer thickness, 第22回 画像の認識・理解シンポジウム (MIRU2019), 大阪府大阪市，2019 年7 月