

# Blind Watermarking for 3-D Printed Objects by Locally Modifying Layer Thickness

Arnaud Delmotte , Kenichiro Tanaka , *Member, IEEE*, Hiroyuki Kubo , Takuya Funatomi , *Member, IEEE*, and Yasuhiro Mukaigawa , *Member, IEEE*

**Abstract**—We propose a new blind watermarking algorithm for 3D printed objects that has applications in metadata embedding, robotic grasping, counterfeit prevention, and crime investigation. Our method can be used on fused deposition modeling (FDM) 3D printers and works by modifying the printed layer thickness on small patches of the surface of an object. These patches can be applied to multiple regions of the object, thereby making it resistant to various attacks such as cropping, local deformation, local surface degradation, or printing errors. The novelties of our method are the use of the thickness of printed layers as a one-dimensional carrier signal to embed data, the minimization of distortion by only modifying the layers locally, and one-shot detection using a common paper scanner. To correct encoding or decoding errors, our method combines multiple patches and uses a 2D parity check to estimate the error probability of each bit to obtain a higher correction rate than a naive majority vote. The parity bits included in the patches have a double purpose because, in addition to error detection, they are also used to identify the orientation of the patches. In our experiments, we successfully embedded a watermark into flat surfaces of 3D objects with various filament colors using a standard FDM 3D printer, extracted it using a common 2D paper scanner and evaluated the sensitivity to surface degradation and signal amplitude.

**Index Terms**—Blind watermarking, layer thickness, metadata, paper scanner, 3D printing.

## I. INTRODUCTION

THREE-DIMENSIONAL (3D) printing has become increasingly popular and accessible recently, and entry-level printers have become sufficiently cheap for consumer's budget and available in public libraries, schools, DIY centers, and makerspaces. However, this technology can also help criminals to commit crimes such as counterfeit production, theft by reproducing keys from pictures, printing TSA master keys, or printing untraceable weapons for violent crimes. These objects are often found at the crime scene but are difficult to trace [1].

Manuscript received June 14, 2019; revised October 27, 2019 and December 3, 2019; accepted December 9, 2019. Date of publication December 25, 2019; date of current version October 23, 2020. This work was supported by JSPS KAKEN under Grant JP17K19979. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiaochun Cao. (Corresponding author: Arnaud Delmotte.)

The authors are with the Optical Media Interface Laboratory, Nara Institute of Science and Technology, Nara 630-0192, Japan (e-mail: arnaud.delmotte.zr3@is.naist.jp; ktanaka@is.naist.jp; hkubo@is.naist.jp; funatomi@is.naist.jp; mukaigawa@is.naist.jp).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMM.2019.2962306

A practical solution to help investigations would be to automatically insert a watermark that contains the printer's serial ID and printing date, which requires cooperation from public printing services and printer manufacturers similar to what is done for two-dimensional (2D) paper printers [2]. This would not prevent criminals from printing objects using their own open-source 3D printer to stay untraceable, but it would make their task more difficult. Additionally, the watermark could be used to identify the owner when a stolen object is found by the police.

A watermark would also be useful for CAD applications to trace the batch ID or retrieve information about the object. Compared with artistic objects, the visibility of the watermark is less important, but there are stronger constraints regarding deformation to preserve the mechanical properties of objects. Cropping resistance is also desired so that information about a broken part can be retrieved so that a new part can be printed or the defective batch can be identified. For robotic grasping, identification and pose estimation for similar-looking objects are challenging problems that can be solved by embedding a watermark in the objects. In this context, it is preferable to extract the watermark from one face instead of performing a full scan to simplify the manipulation and reduce the time consumption.

A watermarking method is typically evaluated using three main characteristics: imperceptibility, robustness, and capacity. In the 3D printing context, similar to other media, imperceptibility means that modifications are not visible to the naked eye, but this definition is also extended to the preservation of the mechanical properties of the object in the CAD context. Robustness means that the watermark can resist various attacks, such as printer or scanner inaccuracies, deformation and surface degradation. Capacity refers to the number of bits of data that can be embedded in the object.

Developing a good watermarking method is challenging because it requires a trade-off between these three characteristics. For example, increasing capacity generally has a detrimental effect on imperceptibility or robustness, and vice versa. In addition to these characteristics, a fast extraction procedure is also often required, or the ability to extract the watermark without requiring the original non-watermarked object, which is called 'blind watermarking.' The requirements on these characteristics are application-specific and there is not a single method that works perfectly in every scenario.

For the previously described contexts, the robotic grasping context does not require high capacity, but a fast and robust decoding method. The CAD context requires robustness to

cropping attack and preservation of mechanical properties, but visibility to the human eye and decoding speed are generally less of a concern. The crime prevention context requires high imperceptibility to avoid criminals attempting to damage the watermark, and blind detection because we do not necessarily have access to the original model. All the described contexts also require a low false positive rate for detection. It is generally preferable to be unable to detect the watermark rather than return a false value, particularly for crime investigation.

Our method meets these requirements by locally modifying the layer thickness on small patches of the surface of an object, and applying it to multiple regions of the object for redundancy. Even if the object surface is locally degraded or cropped, we can still decode it if some patches are intact. Our method also provides low shape distortion, fast extraction from only one pose instead of a full scan, and blind detection. For flat surfaces, the watermark can be extracted using a standard 2D paper scanner and does not require any complicated or expensive equipment. Our method works on any fused deposition modeling (FDM) printer that provides access to the motor controls. This includes all printers with ‘G-code’ support, which is one of the most common file formats for 3D printing. We focused on FDM because it is the most frequently used 3D printing technology with a 67.7% share based on 3dHubs 2018-Q4 trends [3].

The novelties of this paper 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

The remainder of this paper is structured as follows: In Section II, we introduce related work and compare it with our method. In Section III, 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 IV, we describe how to extract the watermark from a printed object using a paper scanner. In Section V, we describe our error correction method and how multiple patches can be used to improve robustness to errors. In Section VI, we explain our experiments. Finally, in Section VII, we discuss our results and future work.

## II. RELATED WORK

Watermarking has been studied intensively for various media such as images [4], [5], audio [6], video [7] and 3D mesh [8], [9], but is still a relatively new field for 3D printing. The fields of 3D mesh watermarking and 3D printing watermarking are closely related, and sometimes methods work for both applications. Wang *et al.* [10] and Medimegh *et al.* [11] conducted a comprehensive survey of 3D mesh watermarking methods. Hou *et al.* [12] conducted a survey of 3D printing watermarking methods.

Most 2D paper printers include a watermark that contains information such as the printer ID and print time. Various methods can be used for this, and are generally not publicly documented

by the printer manufacturer, but the most famous technique consists of inserting a grid of yellow dots [13], [14], which are barely visible to the naked eye but easily retrievable from a scan or by illumination with blue light. The principal use is crime prevention by, for example, tracking leaked documents [2]. Our method can be regarded as a 3D print version of this method because it shares many similarities in terms of properties, use, and applications.

Alpha-dot,<sup>1</sup> a UK company, developed microdots that contain a personal identification number. These dots are 1 mm in diameter and can be glued to valuables to protect them from theft. This is officially approved by the UK police, which has access to the company’s database and can identify the owner when it finds stolen valuables. Our method could be used for the same scenario, by directly embedding a code in a printed object without needing to buy and glue an identification tag, but our patches are currently bigger.

Wee *et al.* [15], [16] and Voris *et al.* [17] proposed inserting RFID tags inside objects. This is a simple solution, but requires additional work and cannot be performed automatically by a standard printer.

Yamazaki *et al.* [18] proposed a non-blind spread-spectrum watermarking method. They aligned the scanned model to the original mesh to extract the spectrum signal and extract the watermark.

Hou *et al.* [19] developed a non-blind watermark method using a circular shift structure. They used the original model to retrieve the base axis during the extraction process.

Hou *et al.* [20] published a blind watermarking method using spread-spectrum watermarking on a set of slices along the printing axis. At extraction time, the method retrieves the printing axis and slice positions by analyzing the layer artifacts. It can resist scanning and reprinting if the print axis is conserved, is robust to printing errors, and has resistance to a small amount of cropping, but the drawbacks are that it requires a full 3D scan, which is time-consuming, and the watermark capacity is low (1 bit).

Adobe [21] obtained a patent for a visible 3D barcode that can be added during printing, which is resistant to printing errors and is decoded from a standard picture taken, for example, using a smartphone. The main drawbacks are the deformation and visibility of the barcode on the object.

Zhang *et al.* [22] developed a method using a regular grid of small bumps on a flat surface to encode data. They proposed a deep convolutional neural network (CNN) for decoding using a standard RGB camera, and used a mix of real and synthetic rendered images to train the network.

Harrison *et al.* [23] proposed an acoustic 1D barcode made out of physical notches that produce an identifiable sound when swiped with e.g., a fingernail. This method only requires a single, inexpensive contact microphone attached to the surface of the object for the decoding.

HP<sup>2</sup> developed Multi Jet Fusion (MJF), which allows the precise control of ink deposition on a 3D object during printing.

<sup>1</sup>[Online]. Available: <https://www.alpha-dot.co.uk/>

<sup>2</sup>[Online]. Available: <https://www8.hp.com/us/en/printers/3d-printers.html>

They demonstrated watermarking to retrieve object metadata using slight color modifications and an invisible QR code that is revealed by UV light [24]. Similarly, Rize<sup>3</sup> patented Augmented Polymer Deposition (APD), which combines fused deposition modeling and ink-jetting, which allows the printing of a QR code on objects that can be decoded using any camera, such as a smartphone [25]. Both techniques are simple to use, do not deform the object, but require specific printers that are extremely expensive.

We developed a blind watermarking method [26] that is retrievable from a full 3D scan and resistant to reprinting but has stronger deformation and is not resistant to cropping or non-uniform deformation. The current application scenario is different from that in our previous paper, which assumed that we attempted the extraction from a standard 3D scan that could be created by someone else without following a precise procedure, whereas the present paper supposes that the user will follow our procedure, that is, the person that performs the scan is the person that is attempting to extract the watermark. Similarly, Yamamoto *et al.* [27] developed a blind watermarking method that resists molding and casting duplication. They estimated a pseudo axis of revolution, subdivided the mesh into slices along the axis, and modified the radius of slices to store the watermark.

Willis *et al.* [28] developed a method to embed tags below the surface and retrieve them using a terahertz camera. Similarly, Okada *et al.* [29] and Suzuki *et al.* [30], [31] used a thermal camera to retrieve the embedded tags, Suzuki *et al.* [32] used near-infrared light and a camera, and Li *et al.* [33] used a camera, projector, and polarizer. Their methods have similar uses and applications to ours and have a higher resistance to surface degradation, but impose constraints on the inner structure which may modify the mechanical properties, increase the extraction time, and be limited to a gently curved surface. They also have some limitations for FDM printing because it produces an irregular structure inside the object, and this structure makes extraction more difficult.

Li *et al.* [1] developed a method that allows a fingerprint to be extracted from a 3D print by analyzing the layers' bonding differences caused by mechanical component tolerances. It works on any printer and does not require the insertion of a watermark, but is more difficult to scale to a large number of printers because it requires an object to be printed and the fingerprint to be extracted for each printer. Additionally, it does not provide information on which user printed the model on a shared printer. Our method requires the printer to include the watermark, similar to the yellow dots for 2D printers, but can be easily scaled to a large number of printers. It is more suitable for public printer services, whereas PrinTracker is more suitable to provide evidence after the analysis of the printer of a suspect. Both methods can be used together to strengthen the evidence.

LayerCode [34] is the method most similar to our method. The watermark is encoded as a 1D barcode along the printing axis by being printed in two colors, with near-infrared ink, or with

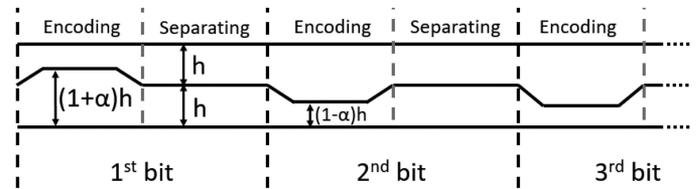


Fig. 1. Encoding layer pattern. The pattern corresponds to two layers with variable thickness. We can encode a 1 or 0 bit by increasing or decreasing the thickness of the bottom layer in the encoding region, respectively. The top layer thickness is adjusted such that the sum of the thickness of the two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

variable layer height. All these three methods have their own advantages and drawbacks. Bi-color printing is easy to implement and does not cause any deformation, but is completely visible. Near-infrared ink is invisible but requires a very specific printer that is not common. Variable layer height is a trade-off between the two other methods; it can be performed with a common FDM printer and is less visible than bi-color printing, but the surface is more degraded than that using the two other methods. Our method also uses layer thickness modification, but we apply it locally instead of modifying the thickness uniformly over the complete layer, which allows the storage of more data per layer. We also apply the method using a much lower amplitude than LayerCode, which preserves more of the surface quality.

We summarize the comparison between existing methods and our method in Table I.

### III. 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. 1 illustrates the layer thickness modifications.

In what follows, we explain our system in detail. The pattern is explained in Section III-A, the selection of regions to watermark is explained in Section III-B and the modification of the printer controls is explained in Section III-C.

#### A. 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.

Fig. 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

<sup>3</sup>[Online]. Available: <https://www.rize3d.com>

TABLE I

COMPARISON WITH RELATED WORKS. SOME PARAMETERS SUCH AS THE CAPACITY ARE NOT ALWAYS CLEARLY STATED IN THE PAPER OR EVALUATED IN THE SAME MANNER. IN THIS TABLE, CROPPING RESISTANCE MEANS THAT THE OBJECT IS CROPPED AT RANDOM POSITIONS, NOT NECESSARILY THE TAG ITSELF. WE CLASSIFIED CROPPING RESISTANCE AS ‘++’ IF IT CAN RESIST TO THE CROPPING OF MORE THAN 50% OF THE OBJECT, PROVIDED ONE OF THE TAGS IS NOT DAMAGED; ‘+/-’ IF IT CAN RESIST SMALL CROPS OF AT LEAST 5–10%; AND ‘--’ IF IT DOES NOT RESIST SMALL CROPS. WE CLASSIFIED INVISIBILITY AS ‘++’ IF THE WATERMARK IS COMPLETELY INVISIBLE TO THE HUMAN EYE; ‘+’ IF SOME ARTIFACTS ARE VISIBLE IF WE LOOK AT THEM CLOSELY; ‘+/-’ IF SOME ARTIFACTS ARE VISIBLE AT FIRST GLANCE; AND ‘-’ IF THE WATERMARK IS CLEARLY PERCEPTIBLE

Method	Blind	Invisibility	Capacity (bits)	One-shot extraction	Irregular shape	Cropping resistance	Decoding equipment	Supported printers
Alpha-dot	✓	?	> 64	✓	✓	++	UV light	Not 3D printing
RFID tags [15] [16] [17]	✓	++	> 64	✓	✓	++	RFID tag reader	All, but requires the insertion of an RFID tag
HP [24], Rize [25]	✓	++	> 64	✓	No information	++	RGB camera	MJF, ADP
Adobe [21]	✓	--	> 64	✓	✗	++	RGB camera	All
Harrison <i>et al.</i> [23]	✓	--	8~24	✓	No evaluation	✓	Contact microphone	All
Yamazaki <i>et al.</i> [18]	✗	++	256	Full scan	✓	+/-	3D scanner	All
Hou <i>et al.</i> [19]	✗	++	24	Full scan	✓	++	3D scanner	All
Hou <i>et al.</i> [20]	✓	++	1	Full scan	✓	+/-	3D scanner	FDM
Zhang <i>et al.</i> [22]	✓	--	> 64	✓	✗	++	RGB camera	All
LayerCode [34] (Dual color)	✓	--	~ 24	✓	✓	++	RGB camera	Dual color printer
LayerCode [34] (near infrared ink)	✓	++	~ 12	✓	✓	++	RGB camera	Dual-resin printer (not common)
LayerCode [34] (variable layer height)	✓	+/-	~ 24	✓	✓	++	RGB camera	FDM
Infrastructs [28]	✓	++	> 64	Few shots Same pose	No evaluation	++	Terahertz camera	All, but with some limitations for FDM
AirCode [33]	✓	++	> 64	Few shots Same pose	✗	++	Projector + camera	Polyjet or similar (not FDM)
Okada <i>et al.</i> [29]	✓	++	> 25	Few shots Same pose	✗	++	Far-infrared camera	SLA or similar
Suzuki <i>et al.</i> [30] [31]	✓	++	> 64	Few shots Same pose	No evaluation	++	Far-infrared camera	All
Suzuki <i>et al.</i> [32]	✓	++	> 64	✓	No evaluation	++	Near-infrared camera	All, but only PLA filament evaluated
Delmotte <i>et al.</i> [26]	✓	+	8~16	Full scan	✓	--	3D scanner	All
Yamamoto <i>et al.</i> [27]	✓	++	~ 8	Full scan	✓	--	3D scanner	All
Our method	✓	+	> 64	✓	Future work	++	Paper scanner (for flat surface)	FDM

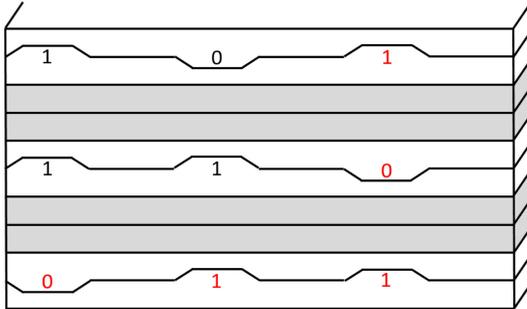


Fig. 2. Example of the watermark pattern. Encoding the value ‘1011’ plus the parity bits. The digits correspond to the encoded bits; those in red are the parity bits. The layers with a white background are the encoding layers and those in gray are the separating layers.

is adjusted to keep the sum of the thickness of the two layers constant.

Fig. 2 illustrates a  $2 \times 2$  watermark, in which the encoding layers are separated by  $M$  separating layers. Both the separating regions and separating layers 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  $(W + 1)$ .

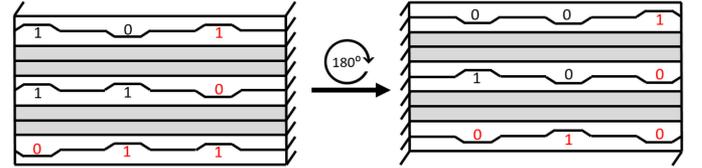


Fig. 3. If the pattern is wrongly oriented by a rotation of  $180^\circ$ , all the parity bits obtain the wrong value. This property allows the orientation at extraction time to be retrieved.

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  $180^\circ$  as illustrated in Fig. 3.

### B. 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

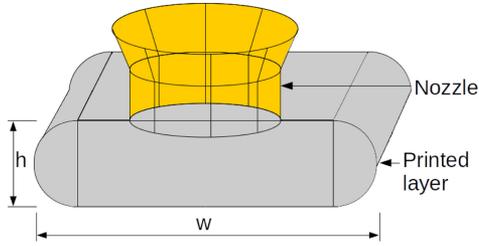


Fig. 4. Cross-section of the nozzle and printed layer.

they are stacked above each other, or 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.

### C. 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 [35], [36], but are complex to use. Therefore, we used the simplified model proposed by [37], [38].

The cross-sectional area of a layer is approximated by a rounded rectangle, as illustrated in Fig. 4, and can be calculated as

$$A_{layer} = h(w - h) + \pi \frac{h^2}{4}, \quad (1)$$

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

$$L_{filament} = \frac{A_{layer}L_{layer}}{\pi(\phi_{filament}/2)^2}, \quad (2)$$

where  $L_{filament}$  is the filament length,  $\phi_{filament}$  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. 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 8 mm/s for the watermarked regions and 60 mm/s for the other regions. There were still some 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 [39].

## IV. 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. 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(a), these parameters depend on the plastic color and printer model.

### A. 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 magnitude value of the Fourier transform, compute the corresponding angle, and reorient the image. This process is illustrated in Fig. 5, and is similar to the frequency analysis section from [20]. 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 0.2 mm layer at 1200 dpi yields  $0.1 \text{ mm} * 47.244 \text{ pixel/mm} = 4.72 \text{ pixels}$ . Fig. 6(b) shows the result of the Non-Maximum Suppression applied to Fig. 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 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.

### B. 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 layers 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. 8. Restricting the searching region for these edges makes the decoding process much more robust to noise.

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(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  $180^\circ$ , as illustrated in Fig. 3, and needs to be rotated. Instead of rotating

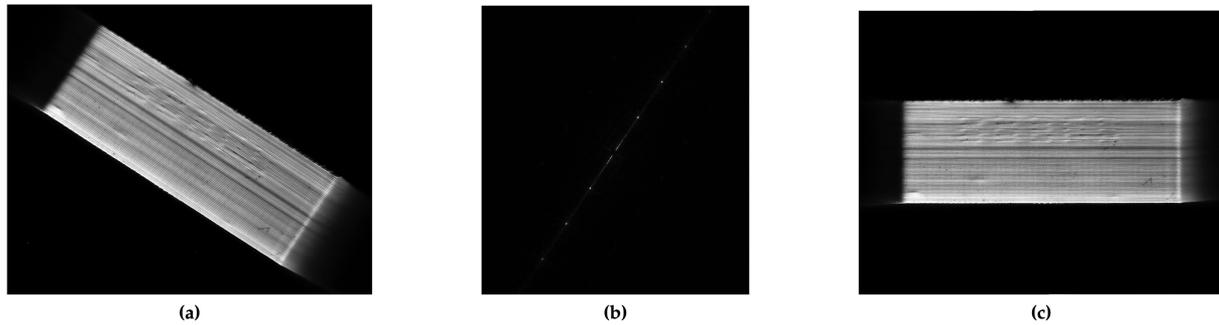


Fig. 5. (a) Image of one face of the object, obtained from the 2D paper scanner. (b) Magnitude of the Fourier transform of image (a) after a high-pass filter is used to remove the low frequency. The line formed by the peak values is perpendicular to the orientation of the layers. (c) Realigned image after extracting the angle from image (b).

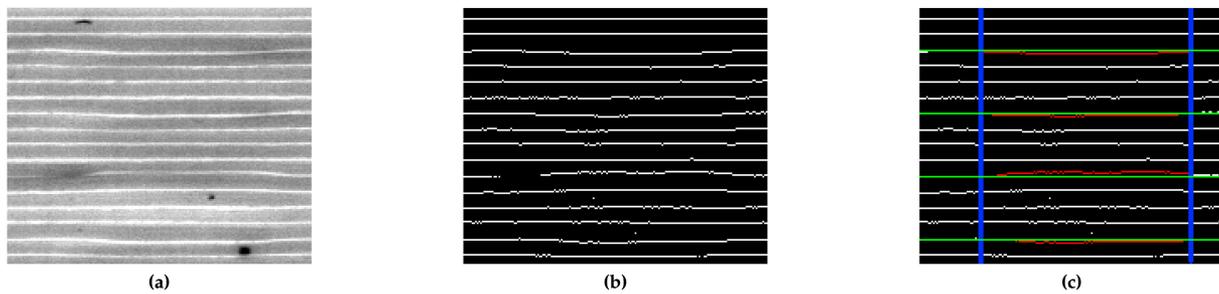


Fig. 6. (a) Zoom in on the scanned image after reorientation. The highlights allow each layer to be segmented. (b) Edges extracted from image (a). (c) Edges with annotation: the blue vertical lines indicate the start and end of the encoding region, the red horizontal curved lines indicate the encoding 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.

the image and restarting the extraction, it is possible to rotate the obtained watermark matrix by  $180^\circ$  and apply a ‘NOT’ operator on all the bits.

## V. ERROR CORRECTION

Each watermark patch contains 2D parity bits, also called two-dimensional parity check, or rectangular code in the literature [40], [41]. In addition to allowing the patch orientation to be determined, as explained in Section III-B, the main use of these 2D parity bits is error detection. As illustrated in Fig. 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. 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  $2 \times 2$  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. 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. 9(d) shows a 3-bit error wrongly detected as a 1-bit error. For a  $2 \times 2$  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. 10(a) shows the result with a majority vote without weighting and Fig. 10(b) shows the result with weighting. For the low error rates, we can observe an improvement, particularly for a

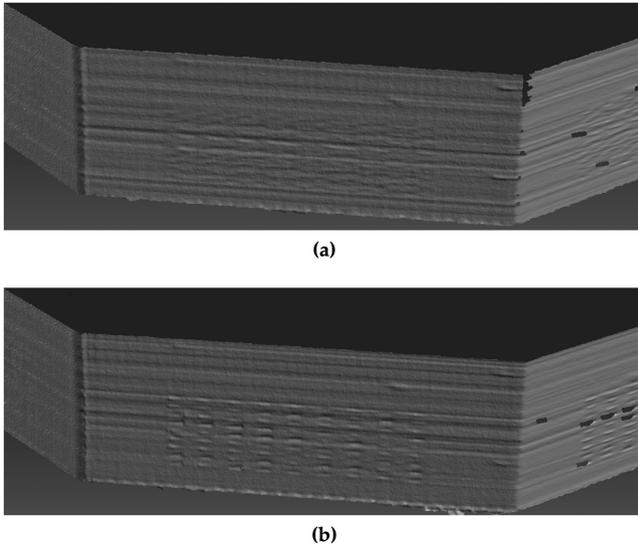


Fig. 7. 3D scan of a 64-bit watermarked object. Scanned using an ‘HP 3D structured light scanner pro S3’: (a) with reduced speed for the watermark region and an adjustment of the plastic extrusion to the layer thickness; and (b) with reduced speed for the watermark region, but without an adjustment of the plastic extrusion, thereby maintaining the original plastic flow even when the thickness is modified.

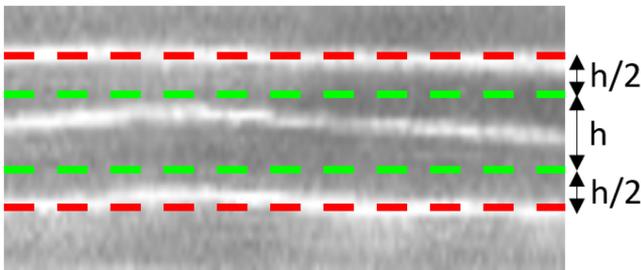


Fig. 8. Region in which to apply non-maximum suppression (in green). The top and bottom edges, in red, are straight lines fitted with RANSAC along the patch. 1D non-maximum suppression is applied in the range between the green lines to find the edge between the two encoding layers.

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. 10(c) and we can observe that it effectively obtained the best results of the two methods. As shown in Fig. 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

0	1	1	0	1	0		1
1	0	0	1	1	1		0
1	1	0	0	1	0		1
0	0	1	1	1	1		0
1	1	0	0	1	1		0
1	0	0	1	1	0		1
0	1	0	1	0	1		1

(a)

0	1	1	0	1	0		1
1	0	0	1	1	1		0
1	1	0	0	1	0		1
0	<span style="border: 1px solid black;">1</span>	1	1	1	1		0
1	1	0	0	1	1		0
1	0	0	1	1	0		1
0	1	0	1	0	1		1

(b)

0	1	1	0	1	0		1
1	0	0	<span style="border: 1px solid black;">0</span>	1	1		0
1	1	0	0	1	0		1
0	<span style="border: 1px solid black;">1</span>	1	1	1	1		0
1	1	0	0	1	1		0
1	0	0	1	1	0		1
0	1	0	1	0	1		1

(c)

0	1	1	0	1	0		1
1	<span style="border: 1px solid black;">1</span>	0	<span style="border: 1px solid black;">0</span>	1	1		0
1	1	0	0	1	0		1
0	<span style="border: 1px solid black;">1</span>	1	<span style="border: 1px solid black;">0</span>	1	1		0
1	1	0	0	1	1		0
1	0	0	1	1	0		1
0	1	0	1	0	1		1

(d)

0	1	1	0	1	0		1
1	<span style="border: 1px solid black;">1</span>	0	0	1	1		0
1	1	0	0	1	0		1
0	<span style="border: 1px solid black;">1</span>	1	<span style="border: 1px solid black;">0</span>	1	1		0
1	1	0	0	1	1		0
1	0	0	1	1	0		1
0	1	0	1	0	1		1

(e)

0	1	1	0	1	0		1
1	<span style="border: 1px solid black;">1</span>	0	<span style="border: 1px solid black;">0</span>	1	1		0
1	1	0	0	1	0		1
0	<span style="border: 1px solid black;">1</span>	1	1	<span style="border: 1px solid black;">0</span>	1		0
1	1	0	0	1	1		0
1	0	0	1	1	0		1
0	1	0	1	0	1		1

(f)

Fig. 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.

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. 12 shows the success rate, and Fig. 13 shows the false-positive rate. The maximum false-positive rate was achieved when the error was a  $2 \times 2$  burst error with a single patch, but was much smaller for other configurations.

## VI. 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’ [42] and the object was printed.

We evaluated our method on two CAD models, illustrated in Fig. 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

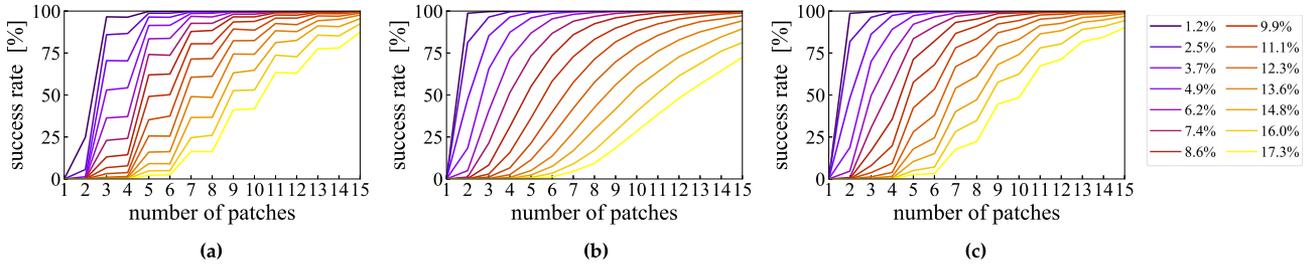


Fig. 10. Success rate of the error correction methods depending on the number of patches used and the initial error rate, simulated on a 64-bit watermark with a uniform error distribution: (a) majority vote without weighting (b) majority vote with weighting; and (c) combination of (a) and (b).

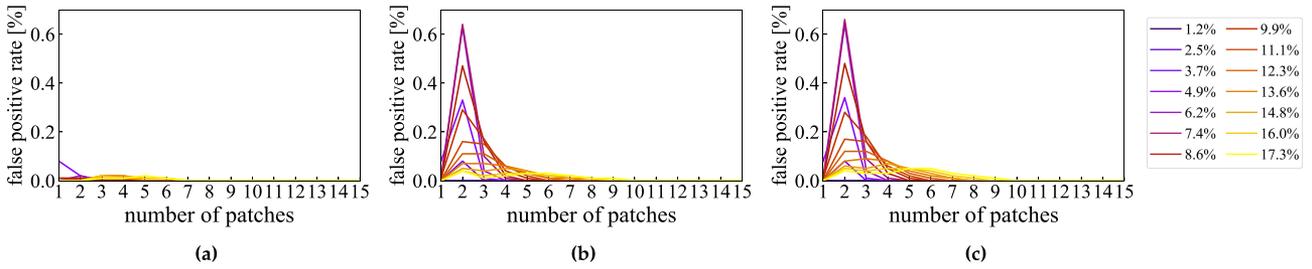


Fig. 11. False-positive rate of the error correction methods depending on the number of patches used and the initial error rate, simulated on 64-bit watermark with a uniform error distribution: (a) majority vote without weighting (b) majority vote with weighting; and (c) combination of (a) and (b).

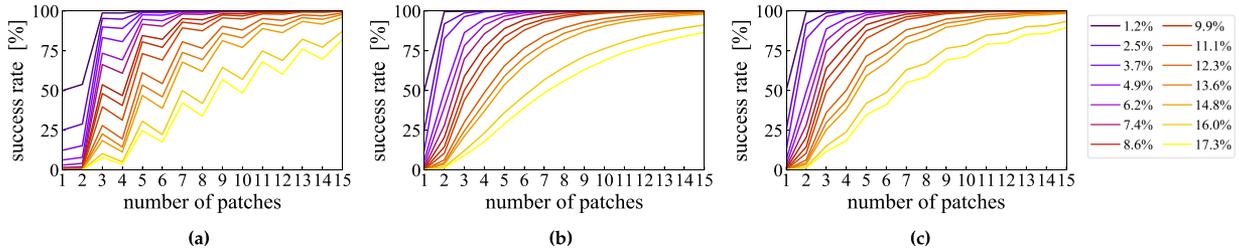


Fig. 12. Success rate of the error correction methods depending on the number of patches used and the burst error region size, simulated on 64-bit watermark: (a) majority vote without weighting (b) majority vote with weighting; and (c) combination of (a) and (b).

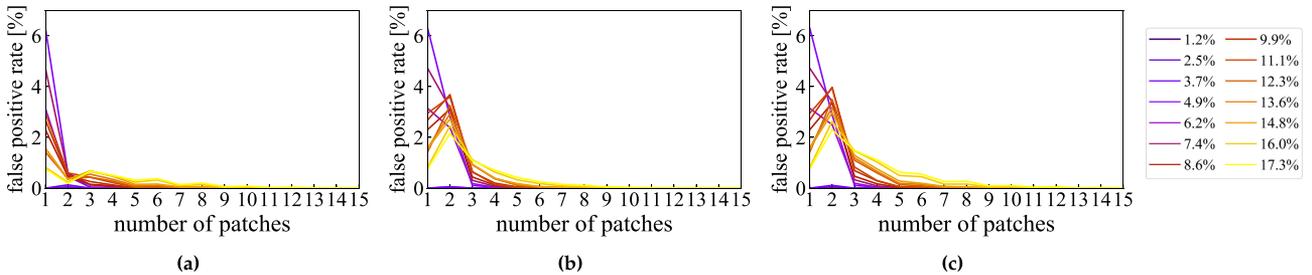


Fig. 13. False-positive rate of the error correction methods depending on the number of patches used and the burst error region size, simulated on 64 bits watermark: (a) majority vote without weighting (b) majority vote with weighting; and (c) combination of (a) and (b).

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  $6.8 \times 25$  mm, with  $bit_{width} = 2.77$  mm.

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

TABLE II  
SCANNER PARAMETERS USED FOR EACH PLA FILAMENT COLOR

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							

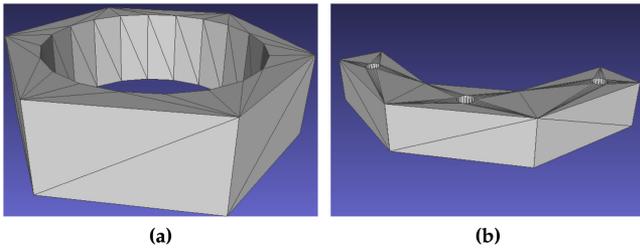


Fig. 14. CAD models used in our experiments. (a) 'hexagonal' model and (b) 'corner connector' model.

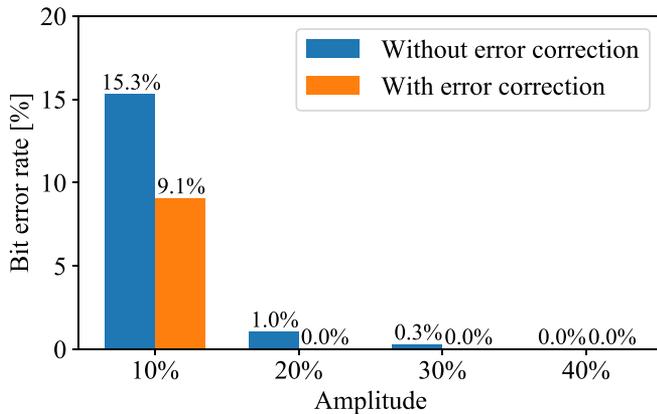


Fig. 15. Bit error rate at multiple signal amplitudes. Tested on the 'hexagonal' model. A total of 18 patches per amplitude value, with 64 bits of data and 17 parity bits per patch. The bit error rate was computed using the 81 bits of each patch, which yielded a total of 1458 bits per amplitude value. For the error correction, three patches from each face were merged. Starting from 20%, the error correction method recovered all the errors.

software 'XSane' on Linux, which gave us more control than the default Windows software. We used the parameters in Table II.

#### A. Robustness to Variation of the Signal Amplitude

First, we evaluated the robustness of our method to multiple signal amplitudes  $\alpha$  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. 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.



Fig. 16. Objects used for color robustness evaluation. Each object was printed with a PLA filament, 0.2 mm layer thickness, and signal amplitude  $\alpha = 40\%$ .

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.

#### B. Robustness to the Printing and Scanning Process With Various Filament Colors

After choosing the value for the amplitude  $\alpha$ , 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. 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 list the parameters used in our experiments in Table II. As we observed in Fig. 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  $7.3 \times 10^{-5}$ , or  $9.3 \times 10^{-5}$  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

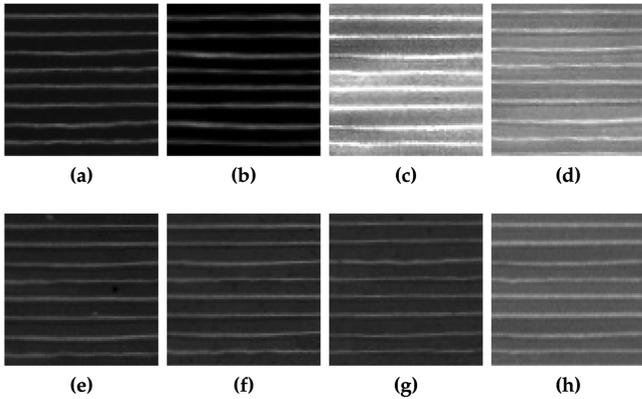


Fig. 17. Zoom in on the scanned images with multiple filament colors, using parameters from Table II: (a) black PLA (b) gray PLA (c) white PLA (d) yellow PLA (e) red PLA (f) green PLA (g) blue PLA (h) orange PLA.

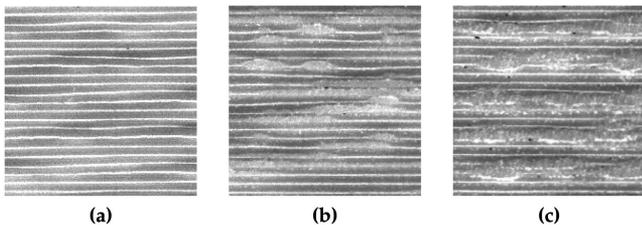


Fig. 18. Scanned layers after manually sanding the surface. (a) No sanding. (b) 400-grit sand paper, medium pressure for 30 seconds; and (c) 240-grit sand paper, medium pressure for 30 seconds.

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. 17. Once the scanning parameters for a filament have been chosen, they can be used for any object printed using this filament.

### C. 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. Fig. 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. 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

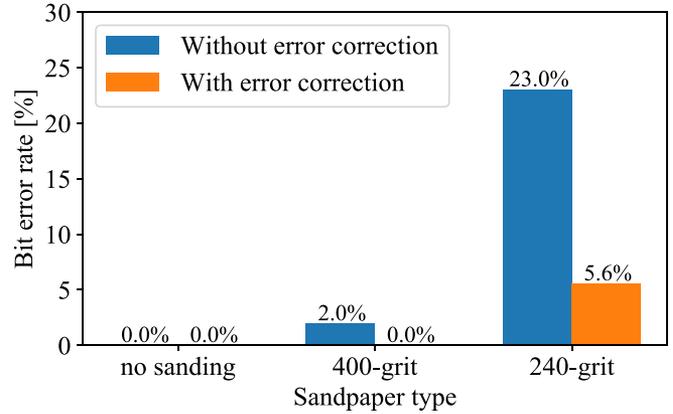


Fig. 19. Bit error rate after manually sanding the surface. Tested on the ‘hexagonal’ model. A total of 18 patches per grit-size, with 64 bits of data and 17 parity bits per patch. The bit error rate was computed using the 81 bits of each patch, which yields a total of 1458 bits per grit size. Each face was sanded for 30 seconds using medium pressure with 400-grit and 240-grit sand paper.

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.

## VII. 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 [43] 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,

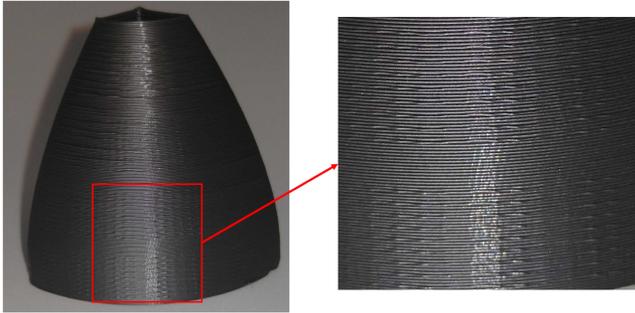


Fig. 20. Watermark encoded on a curved surface printed with gray PLA.

Hadamard, and Reed–Muller codes [43] 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. 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.

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 [44] and be used for educating children [45], [46]. For smart manufacturing, it could replace RFID tags for individual part identification during the complete manufacturing process [47].

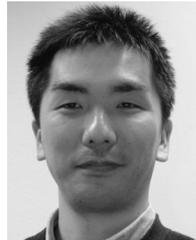
## REFERENCES

- [1] Z. Li *et al.*, “Printracker: Fingerprinting 3d printers using commodity scanners,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2018, pp. 1306–1323.
- [2] C. Baraniuk, “Why printers add secret tracking dots,” (2017). Accessed: Mar. 17, 2019. [Online] Available: <http://www.bbc.com/future/story/20170607-why-printers-add-secret-tracking-dots>
- [3] 3dHubs, “Online manufacturing trends q4-2018,” (2018). Accessed: May 27, 2019. [Online] Available: <https://www.3dhubs.com/trends>
- [4] X. Zhu, J. Ding, H. Dong, K. Hu, and X. Zhang, “Normalized correlation-based quantization modulation for robust watermarking,” *IEEE Trans. Multimedia*, vol. 16, no. 7, pp. 1888–1904, Nov. 2014.
- [5] W. Zhang, H. Wang, D. Hou, and N. Yu, “Reversible data hiding in encrypted images by reversible image transformation,” *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1469–1479, Aug. 2016.
- [6] R. Kazemi, F. Pérez-González, M. A. Akhac, and F. Behnia, “Data hiding robust to mobile communication vocoders,” *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2345–2357, Dec. 2016.
- [7] T. Stütz, F. Atrousseau, and A. Uhl, “Non-blind structure-preserving substitution watermarking of h. 264/cavlc inter-frames,” *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1337–1349, Aug. 2014.
- [8] J. M. Konstantinides, A. Mademlis, P. Daras, P. A. Mitkas, and M. G. Strintzis, “Blind robust 3-d mesh watermarking based on oblate spheroidal harmonics,” *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 23–38, Jan. 2009.
- [9] R. Jiang, H. Zhou, W. Zhang, and N. Yu, “Reversible data hiding in encrypted three-dimensional mesh models,” *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 55–67, Jan. 2018.
- [10] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, “A comprehensive survey on three-dimensional mesh watermarking,” *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1513–1527, Dec. 2008.
- [11] N. Medimegh, S. Belaid, and N. Werghi, “A survey of the 3d triangular mesh watermarking techniques,” *Int. J. Multimedia*, vol. 1, no. 1, pp. 1–8, 2015.
- [12] J.-U. Hou, D. Kim, W.-H. Ahn, and H.-K. Lee, “Copyright protections of digital content in the age of 3d printer: Emerging issues and survey,” *IEEE Access*, vol. 6, pp. 44 082–44 093, 2018.
- [13] P.-J. Chiang *et al.*, “Printer and scanner forensics,” *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 72–83, Mar. 2009.
- [14] EFF, “Investigating machine identification code technology in color laser printers,” (2005). Accessed: Mar. 17, 2019. [Online] Available: <http://www.eff.org/wp/investigating-machine-identification-code-technology-color-laser-printers>
- [15] J. Y. S. Wee, C. I. Byatte, A. D. G. Rhoades, and D. L. McNeight, “Objets de vertu,” Dec. 10, 2015, U.S. Patent App. 14/485,880.
- [16] J. Y. S. Wee, C. I. Byatte, A. D. G. Rhoades, and D. L. McNeight, “Product authentication,” Sep. 3, 2015, U.S. Patent App. 14/250,533.
- [17] J. Voris, B. F. Christen, J. Alted, and D. W. Crawford, “Three dimensional (3d) printed objects with embedded identification (id) elements,” May 23, 2017, U.S. Patent 9,656,428.
- [18] S. Yamazaki, S. Kagami, and M. Mochimaru, “Extracting watermark from 3d prints,” in *Proc. IEEE 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 4576–4581.
- [19] J.-U. Hou, D.-G. Kim, S. Choi, and H.-K. Lee, “3d print-scan resilient watermarking using a histogram-based circular shift coding structure,” in *Proc. 3rd ACM Workshop Inf. Hiding Multimedia Secur.*, 2015, pp. 115–121.
- [20] J.-U. Hou, D.-G. Kim, and H.-K. Lee, “Blind 3d mesh watermarking for 3d printed model by analyzing layering artifact,” *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 11, pp. 2712–2725, Nov. 2017.
- [21] A. Kumar, N. P. Goel, and M. Hemani, “Method and apparatus for storing and retrieving data embedded into the surface of a 3d printed object,” Jul. 26 2016, U.S. Patent 9,400,910.
- [22] X. Zhang, Q. Wang, and I. Ivrisimtzi, “Single image watermark retrieval from 3d printed surfaces via convolutional neural networks,” in *Proc. Comput. Graph. Visual Comput.*, 2018, pp. 117–120.
- [23] C. Harrison, R. Xiao, and S. Hudson, “Acoustic barcodes: Passive, durable and inexpensive notched identification tags,” in *Proc. 25th Annu. ACM Symp. User Interface Softw. Technol.*, 2012, pp. 563–568.
- [24] M. Molitch-Hou, “The future of hps multi jet fusion 3d printing,” (2016) Accessed: May 21, 2019. [Online] Available: <https://www.engineering.com/3DPrinting/3DPrintingArticles/ArticleID/12298/The-Future-of-HPs-Multi-Jet-Fusion-3D-Printing.aspx>
- [25] B. Yusuf, “Digitally augmented additive manufacturing parts from rize,” (2018). Accessed: Apr. 24, 2019. [Online] Available: <https://all3dp.com/digitally-augmented-additive-manufacturing-parts-from-rize>

- [26] A. Delmotte, K. Tanaka, H. Kubo, T. Funatomi, and Y. Mukaigawa, "Blind watermarking for 3-d printed objects using surface norm distribution," in *Proc. IEEE Joint 7th Int. Conf. Inform., Electron. Vision 2nd Int. Conf. Imag., Vision Pattern Recognit.* 2018, pp. 282–288.
- [27] H. Yamamoto and K. Sano, "A watermarking method for embedding into the external shapes of objects," in *Proc. IEEE Int. Symp. Inf. Theory Appl.*, 2018, pp. 321–325.
- [28] K. D. Willis and A. D. Wilson, "Infrastructs: Fabricating information inside physical objects for imaging in the terahertz region," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 138.
- [29] A. Okada *et al.*, "Non-destructively reading out information embedded inside real objects by using far-infrared light," in *Proc. Appl. Digit. Image Process. XXXVIII*, vol. 9599. Int. Soc. Opt. Photonics, 2015, paper 95992V.
- [30] M. Suzuki *et al.*, "Technique for protecting copyrights of digital data for 3-d printing, and its application to low infill density objects," in *Proc. 8th Int. Conf. Adv. Multimedia*, 2016, pp. 56–59.
- [31] M. Suzuki *et al.*, "Embedding information into objects fabricated with 3-d printers by forming fine cavities inside them," *Electron. Imag.*, vol. 2017, no. 7, pp. 6–9, 2017.
- [32] M. Suzuki, T. Matumoto, Y. Takashima, H. Torii, and K. Uehira, "Information hiding inside 3-d printed objects by forming high reflectance projections," in *Proc. Int. Conf. Video Image Process.*, 2017, pp. 146–150.
- [33] D. Li, A. S. Nair, S. K. Nayar, and C. Zheng, "Aircode: Unobtrusive physical tags for digital fabrication," in *Proc. 30th Annu. ACM Symp. User Interface Softw. Technol.*, 2017, pp. 449–460.
- [34] H. T. Maia, D. Li, Y. Yang, and C. Zheng, "Layercode: Optical barcodes for 3d printed shapes," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 112:1–112:14, Jul. 2019. [Online]. Available: <http://doi.acm.org/10.1145/3306346.3322960>
- [35] A. Pandey and S. K. Pradhan, "Investigations into complete liquefier dynamics and optimization of process parameters for fused deposition modeling," *Mater. Today: Proc.*, vol. 5, no. 5, pp. 12 940–12 955, 2018.
- [36] H. Xia, J. Lu, S. Dabiri, and G. Tryggvason, "Fully resolved numerical simulations of fused deposition modeling. Part I: fluid flow," *Rapid Prototyping J.*, vol. 24, no. 2, pp. 463–476, 2018.
- [37] G. Hodgson, A. Ranellucci, and J. Moe, "Slic3r manual: Flow math," 2011. [Online]. Available: <http://manual.slic3r.org/advanced/flow-math>
- [38] G. P. Greeff and M. Schilling, "Closed loop control of slippage during filament transport in molten material extrusion," *Additive Manuf.*, vol. 14, pp. 31–38, 2017.
- [39] A. Bellini, S. Guceri, and M. Bertoldi, "Liquefier dynamics in fused deposition," *J. Manuf. Sci. Eng.*, vol. 126, no. 2, pp. 237–246, 2004.
- [40] R. W. Hamming, *Coding and Information Theory*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1986.
- [41] R. Banerjee, S. Naskar, and S. S. Sarma, "A closer look into the rectangular codes: Where the parity check predominates," *J. Global Res. Comput. Sci.*, vol. 3, no. 3, pp. 17–23, 2012.
- [42] Reprap, "Reprap g-code," Accessed: Jun. 11, 2019. [Online]. Available: <https://reprap.org/wiki/G-code>
- [43] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. New York, NY, USA: Elsevier, 1977, vol. 16.
- [44] A. Katiyar, K. Kalra, and C. Garg, "Marker based augmented reality," *Adv. Comput. Sci. Inf. Technol.*, vol. 2, no. 5, pp. 441–445, 2015.
- [45] Y. Zhu and S. J. Wang, "A tangible augmented reality toy kit: Interactive solution for early childhood education," in *Proc. Interactivity, Game Creation, Des., Learn., Innov.* Springer, 2016, pp. 12–19.
- [46] M. Tentori, L. Escobedo, and G. Balderas, "A smart environment for children with autism," *IEEE Pervasive Comput.*, vol. 14, no. 2, pp. 42–50, Apr.–Jun. 2015.
- [47] L. Zhekun, R. Gadh, and B. Prabhu, "Applications of rfid technology and smart parts in manufacturing," in *Proc. DETC*, 2004, vol. 4, pp. 1–7.



**Arnaud Delmotte** received the M.S. degree in civil engineering in informatics from the University of Louvain, Belgium, in 2013. He is currently working toward the Ph.D. degree in computer science division, Nara Institute of Science and Technology, Ikoma, Japan. His research interests include watermarking, 3D printing, and computer vision.



**Kenichiro Tanaka** received the M.S. and Ph.D. degrees in CS from Osaka University, Suita, Japan, in 2014 and 2017, respectively. In 2017, he joined Nara Institute of Science and Technology, Ikoma, Japan, as an Assistant Professor. His research interests include computer vision and computational photography, imaging, and illumination. He is a member of CVF.



**Hiroyuki Kubo** received the M.S. and Ph.D. degrees from Waseda University, Tokyo, Japan, in 2008 and 2012, respectively. He has been an Assistant Professor with Nara Institute of Science and Technology, Ikoma, Japan, since 2014. His research interests include computer graphics and computer animation. He is a member of ACM.



**Takuya Funatomi** received the B.S. degree in engineering from the Graduate School of Informatics, Kyoto University, Kyoto, Japan, in 2002, the M.S. and Ph.D. degrees in informatics from the Graduate School of Informatics, Kyoto University, in 2004 and 2007, respectively. He has been an Associate Professor with Information Science Department, Nara Institute of Science and Technology, Ikoma, Japan, since 2015. He was an Assistant Professor with Kyoto University from 2007 to 2015, and a visiting Assistant Professor with Stanford University, Stanford,

CA, USA, in 2014. His research interests include computer vision, computer graphics, and pattern recognition.



**Yasuhiro Mukaigawa** received the M.E. and Ph.D. degrees from the University of Tsukuba, Tsukuba, Japan, in 1994 and 1997, respectively. He became a Research Associate with Okayama University, Okayama, Japan, in 1997; an Assistant Professor with University of Tsukuba, Tsukuba, Japan, in 2003; an Associate Professor with Osaka University, Suita, Japan, in 2004; and a Professor with the Nara Institute of Science and Technology, Ikoma, Japan, in 2014. His current research interests include photometric analysis and computational photography.