Watermarking for Fused Deposition Modeling by Seam Placement

Felix W. Baumann[1,a] and Dieter Roller[1]

[1] *Institut für Rechnergestützte Ingenieursysteme University of Stuttgart Universitätsstr. 38 70569 Stuttgart, Germany*

Abstract. With the increased usage and application of 3D-printing or Additive Manufacturing (AM) the question arises of how content providers or creators can ensure their intellectual property on such model data. Similar to other digital media such object information that is represented in a number of file formats is easy to copy and reproduce lossless. This research contributes by a proposition of a watermarking schema for Fused Deposition Modeling (FDM) type 3D-printers. This system embeds information into the 3D-printed object without alterations to the structure or geometry by altering the entry points of each layer in a specific manner. With such a watermarking schema employed objects can be embedded with additional information such as a serial number or other traceable information.

1 Introduction

For commercial scenarios in 3D-printing where the creator of a digital model wants to ensure that objects 3D-printed are traceable to the customer it is essential to embed such information in a way that can't be removed easily. In the currently existing 3D-Model marketplaces, such as TurboSquid [1] try to prevent the Intellectual Property (IP) misuse by contractual obligations, others like i.materialise [2] and Shapeways [3] do not allow the user to download the acquired model but only the direct 3Dprinting through the service itself.

In case of limiting the use of a digital model by contract the unlicensed use of the digital model requires some form of identification or information within the model to prosecute the misconduct.

For future use cases where marketplaces for digital models allow the user to download the model it is also required to employ some form of tracing information when the service provider wants to ensure the legitimate usage of the digital models.

With this work a proposal for the addition of such information to digital models that are 3Dprinted is provided by a watermarking schema. This watermarking schema is intended for FDM 3Dprinting and tested in an experiment to possess the required features for identification. The watermarking schema does not require additional materials or components and does not alter the geometrical and structural composition and features of the object.

This work is an addition to Baumann et al. "Streaming Provisioning for Additive Manufacturing – Securing IP", which is submitted to the Journal of Smart and Sustainable Manufacturing Systems and is currently under review. Fragments from this work are used for this article. This work is structured as follows: In the next section (2) the existing and related literature is examined and discussed for the

[a] Corresponding author : felix.baumann@informatik.uni-stuttgart.de

applicability to the problem discussed within this work. In Sect.3 the conceptual background to the watermarking schema is presented. In Sect. 4 the watermarking schema is explained with Sect. 5 presenting information on the detection and decoding process required. The Sect. 6 presents the results of an experimental implementation of the watermarking schema. This work concludes with a brief summary in Sect. 7.

2 Related Work

Research on the security of digital models in AM is currently sparse.

This research is required to avoid problems [4] encountered in the video and music industry by the improvements in compression technology and the widespread use of broad-band Internet connections [5–7]. One distinction between the video and music sector and the AM sector is that with the latter, physical objects are the end-product and thus only duplication in previous steps can occur, as digital replication from analogue objects is non-error free [8].

Existing research on watermarking on 3D meshes is performed for more than 15 years. In Benedens [9], the author presents the required and recommended properties of 3D watermarking technologies as well as an overview of contemporary algorithms. The use case for this work is the Virtual Reality Modeling Language (VRML) data of the 3D Web, which is still in development. The author identifies Ohbuchi, Masuda, and Aono [10] as the first publication on 3D watermarking.

Research on the perceptual quality of such mesh-bashed watermarking algorithms is presented in the work of Gelasca et al. [11]. The authors develop their watermark quality comparison on the surface roughness which they identify as indicative of the quality of the object and the watermarking algorithm. Chou and Tseng [12] provide a review on 3D watermarking techniques with their work. In their work they identify and present six robust watermarking algorithms and six fragile watermarking methods.

Identification of objects not relying on surface watermarking or the integration of additional objects or material into an additively manufactured object is presented in the work of Okada et al. [13]. The authors describe a method to extract information on internal structural properties like cavities in objects by the utilisation of far-infrared light. A similar method is introduced by Willis and Wilson [14] which is called InfraStructs and relies on the detection of mixed-material detection using Terahertz radiation.

The book by Yu et al. [15] has a separate chapter dedicated to 3D model watermarking. The authors provide detailed information on the requirements for such systems, attack scenarios, restrictions and applications. In this work eight different classification schemes are presented. Despite providing a thorough work on 3D watermarking, the authors do not explicitly research the application in 3D-printing but only mention 3D-printing as a means to generate objects.

3 Theory

In order to watermark a specific object 3D-printed with a FDM type 3D-printer without requiring additional or embedded material like a Radio-frequency identification (RFID) chip and in the moment of 3D-printing is possible to alter the geometrical structure of the object or alter the deposition strategy that can leave traces in the finished object. Alterations to the structure of an object are described in a number of publications on 3D mesh watermarking, see Yu et al. [15]. Structural alterations on the surface are prohibitive in cases that require the verbatim reproduction of the model data. As an alternative to these methods, internal structural changes for watermarking purposes are possible, as described in Willis and Wilson [14].

The approach proposed in this work is based on the alteration of the layer entry points of the model. In all 3D-printing methods that are based on slicing and that rely on the tracing of the structural elements of the objects, i.e., that are not manufacturing one layer in its totality, one point is selected as a starting point for this layer. The starting point can be located inside of the object or on its boundary. In either case, one point is the first point that is 3D-printed on the boundary of the layer of

the slice of the object. In case of an internal starting point, the strategy to 3D-print this layer can be altered without consequences to have a starting point at the boundary.
In case of FDM 3D-printing this starting point of a layer is travelled to by the printhead in a non-extruding move with retracted filament. By a miniscule increase of material extruded at this point, the starting point can be identified in the completed object. See Fig. 1 for a depiction of a visible stacking of starting points on an object 3D-printed with yellow acrylonitrile butadiene styrene (ABS) filament. In this figure, all the starting points for every layer are positioned at the same X- and Y-coordinates by the slicing software.

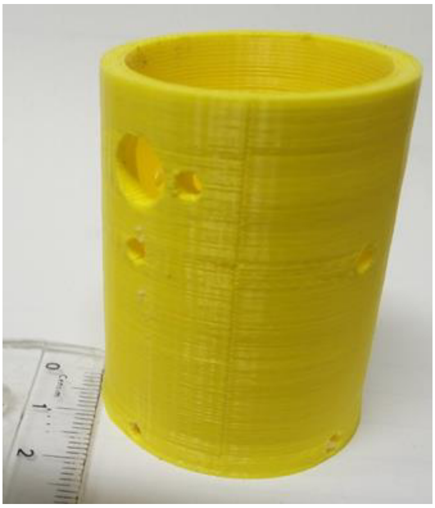


Figure 1. Object with Stacked and Visible Layer Start Points

As these starting points can be identified in the completed object and are free to place on the boundary of the object by the slicing software without alterations to the structure, geometry or topology, it is possible to encode information by controlled placement of these starting points and the consecutive altered layer deposition path.

4. Watermarking Schema

In the slicing software MiracleGrue [16] for Makerbot 3D-printers and Slic3r [17] the default setting for the starting or insertion point of a new layer is always at the same position for layers. This leads to a seam like structure that is visible on the 3D-printed object. By employing the following schema for the placement of the starting points it is possible to encode one of eight different symbols per layer.

In the proposed schema, the object's layers are viewed from above and are virtually centred around a centre point. The starting point is then located at any point of the layer's boundary that lies within the segment that reflects the value of the information that is to be encoded for this layer. See Fig. 2 for an illustration of the segmentation. This segmentation can be extended to employ more encoding capabilities by shrinking the areas of the separate segments. The segmentation can also be reduced by expanding the area of the separate segments down to two segments. In Fig. 2 the value 2 is encoded with this layer as the starting point, depicted by the green rectangle, is placed in the second segment. The segments are indicated on a circle with a dotted line and separated by dashed lines. The object's layer boundary or perimeter is indicated by the blue line.

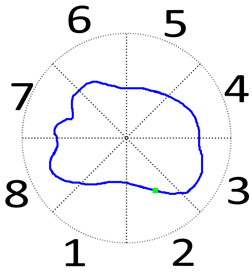


Figure 2. Watermarking Coding Schema with Starting Point Encoding Segment 2

The visibility of the entry point can be enhanced by either extruding a small amount of extra material at this location or by a miniscule displacement to generate a hole.

The path for the extrusion is to be adapted for the specific entry point and is under the responsibility of the utilised slicing software. Each slicing software employs a specific strategy for optimal path planning for each layer with respect to the layer below [18].

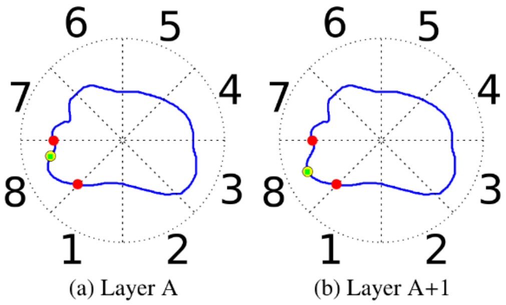
As a negative consequence of the application of this watermarking schema it is possible for the object's printing duration to be extended, as the selected starting point might be farthest from the last point of the layer below, thus increasing the travel time of the printhead.

To minimize the effects of stacked starting points in consecutive layers for the encoding of the same information the starting point must not be placed at the same X- and Y-coordinates. This is achieved by randomisation of the starting point within the respective segments with thresholding around the previous layer's starting point and margins at the segments borders. In Fig. 3 this pattern is depicted. In this figure, the value for segment 8 is encoded in two consecutive layers, with the red dots on the boundary representing the margin at the segment borders that are unavailable for the selection of the starting point. The yellow circle with the black border around the green starting point represents the thresholding boundary around the starting point which is unavailable for the selection of the starting point in the consecutive layer.

In order to identify the order of the segments on the 3D-printed object the first three layers are to be identified with markings in segment 1. For error identification purposes an adapted ISBN10 checksum [19] is utilized. This checksum (see Eq. 1) is added to the information that is to be embedded in the object at every tenth place. In Fig. 1 the indication pi is for the ith place of the information.

$$p_{10} = ((\sum_{i=1}^{9} i * p_i)\ mod\ 7) + 1 \quad (1)$$

The watermarked information is to be repeatedly adjusted with markings in the segment 2 for two consecutive layers after the checksum is placed every tenth layer.



(a) Layer A (b) Layer A+1

Figure 3. Encoding of the Segment 8 in Two Consecutive Layers

The information to be embedded as a watermark is to be repeated for all layers of the object when the first iteration of the watermark is embedded. Consecutive iterations must not be identified with the initial three layer marking in segment 1.

If the information is not a multiple of 9, then the remaining layers to fill 9 layers for the checksum are to be identified by repetition of a marker in the segment of the last layer that contained information with an additional marking in the following segment. The layers indicated by this dual marking are not to be considered for the checksum.

As an example the information 352 662 631 334 476 1 is to be encoded in the following way. The object to be manufactured is 8.7 mm in height and the layer size is set to 0.3 mm. The information is stored in the layers starting with layer 1 at height 0.0 as follows: *1 1 1* 3 5 2 6 6 2 6 3 1 **7** *2 2* 3 3 4 4 7 6 1+2 1+2 **4** 3 5 2 6. In this listing, **bold digits** indicate the checksum digits, underlined digits indicate the missing information places and *italicised digits* indicate the placement markers.

5 Detection and Decoding

For the detection of the watermark at least four digital images of the object must be available. These images must cover all sides of the object and must be of sufficient resolution to detect separate layers and the starting points. Common layer heights of FDM 3D-printed objects range between 0.1 mm and 0.4 mm. A camera with 12 Megapixel resolution yields an image of 4200x2800 pixel (px) size. For an object of 10 cm height photographed in its entirety this results in approximately = 42 px per mm. As the common layer height is as described above, this means that one layer is approximately represented by 4 pxs.

The decoding can be performed manually by visual inspection or software supported. From digital images the orientation of the virtual segmentation can be deduced by the initial marking of the first three layers in segment 1. On identification of the first segment, the other segments can be identified from digital images as they are placed in a counter clockwise orientation.

In digital photo manipulation software, e.g., GIMP [20] is then possible to highlight or otherwise mark these entry points which can then be transcribed back to the original values. The repeated reposition markers after the checksum every tenth layer are further indicators that ease the decoding of the contained information.

6 Implementation

A prototype is implemented as a proof of concept by script that manually generates a tower with side lengths of 20 mm at variable heights. Information of 8 bits can be stored per layer with a selected layer height of 0.3 mm. The system is tested on a MakerBot Replicator 2Xwith ABS 1.75 mm Yellow, ABS filament from REC, Moscow, Russia.

The software to generate the G-code (G-Code) is a GNU Bash [21] script, available from the authors upon request.

See Fig. 4 for the result of an object created containing information embedded on its surface structure. In this figure the starting points are highlighted by red circles and the layers are indicated by blue lines. The numbers next to the indicative circles for the starting points indicate the respective layer. The figure depicts the front side with segments 1 (left-hand side) and 2 (right-hand side) visible. The information contained in this example is xxx1x2x1x2x1xxxxx (with x indicating values not visible from the figure, i.e., on other sides). The image was acquired using a Canon LiDE 210 [22] scanner at a resolution of 1200 dots per inch (dpi). Due to brevity reasons only one side is depicted. Future implementations will embed the watermarking schema into the Slic3r software.

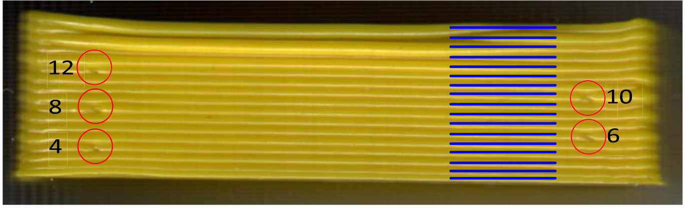


Figure 4. Experiment Result with Marked Starting Points

7 Summary

In this work, an approach to digitally ensure the traceability of a FDM manufactured object by embedding a watermark without the alteration of both the structure and geometry, is presented. The watermarking schema does not require additional materials or components and is readily available to any object that is built on an FDM type 3D-printer.

References

1. TurboSquid, 3d models for professionals :: Turbosquid, http://www.turbosquid.com
2. i.materialise, Online 3d printing service | i.materialise, https://i.materialise.com
3. Shapeways, Shapeways - 3d printing service and marketplace, https://www.shapeways.com
4. G. McDonald, C. Roberts, Journal of Product & Brand Management , **3**, 55 (1994)
5. S. Bhattacharjee, R.D. Gopal, G.L. Sanders, Communications of the ACM, **46**, 107 (2003)
6. M. Fetscherin, M. Schmid, Comparing the Usage of Digital Rights Management Systems in the Music, Film, and Print Industry, in Proceedings of the 5th International Conference on Electronic Commerce (ACM, New York, NY, USA, 2003), ICEC ’03, pp. 316–325, ISBN 158113-788-5, http://dx.doi.org/10.1145/948005.948047
7. R.D. Gopal, G.L. Sanders, S. Bhattacharjee, M. Agrawal, S.C. Wagner, Journal of Organizational Computing and Electronic Commerce, **14**, 89 (2004)
8. Q. Jiang, X. Feng, Y. Gong, L. Song, S. Ran, J. Cui, Computers and Geotechnics, **73**, 210 (2016)
9. O. Benedens, IEEE Computer Graphics and Applications, **19**, 46 (1999)
10. R. Ohbuchi, H. Masuda, M. Aono, Watermarking three-dimensional polygonal models, in MULTIMEDIA’97 Proceedings of the fifth ACM international conference on Multimedia (1997), pp. 261–272, ISBN 0-89791-991-2, http://dx.doi.org/10.1145/266180.266377
11. E.D. Gelasca, T. Ebrahimi, M. Corsini, M. Barni, Objective evaluation of the perceptual quality of 3D watermarking, in IEEE International Conference on Image Processing 2005 (2005), Vol. 1, pp. 1–4, ISSN 1522-4880, http://dx.doi.org/10.1109/ICIP.2005.1529732
12. C.M. Chou, D.C. Tseng, International Journal of Computer Science and Network Security, **7**, 328 (2007)
13. Okada, P. Silapasuphakornwong, M. Suzuki, H. Torii, Y. Takashima, K. Uehira, Nondestructively reading out information embedded inside real objects by using far-infrared light (2015), Vol. 9599, pp. 1–7, http://dx.doi.org/10.1117/12.2189486
14. K.D.D. Willis, A.D. Wilson, ACM Transactions on Graphics (TOG) - SIGGRAPH 2013 Conference, **32**, 1 (2013)
15. F. Yu, H. Luo, Z. Lu, P. Wang, 3D Model Watermarking (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010), pp. 305–370, ISBN 978-3-642-12651-2, http://dx.doi.org/10.1007/978-3-642-12651-2_5
16. Makerbot Industries, Miraclegrue (beta), http://makerbot.github.io/Miracle-Grue (2012)
17. Ranellucci, H.B. Andersen, N. Dandrimont, M. Hindness, P. Ledvina, Y. Sapir, M. Sheldrake, G. Hodgson, Slic3r – g-code generator for 3d printers, http://slic3r.org (2011)
18. F. Baumann, H. Bugdayci, J. Grunert, F. Keller, D. Roller, Computer-Aided Design and Applications, **13**, 14 (2016)
19. ISBN-Agentur für die Bundesrepublik Deutschland, Tech.rep. (2012), http://www.german-isbn.org/downloads/ISBN-Handbuch.pdf
20. T.G. Team, Gimp – gnu image manipulation program, https://www.gimp.org (1996)
21. Free Software Foundation, Bash – gnu project – free software foundation, https://www.gnu.org/software/bash (2007)
22. Canon Europe, Canon canoscan lide 210 – canoscan flatbed photo & document scanners – canon europe, http://www.canon-europe.com/scanners/flatbed-scanners/canoscan_lide_210 (2012)