



POLITECNICO
MILANO 1863

DIPARTIMENTO DI MECCANICA



Direct texturing for additive manufacturing: software support and build tests

Antonio Armillotta

This is a post-peer-review, pre-copyedit version of an article published in Armillotta, A. (2019), "Direct texturing for additive manufacturing: software support and build tests", Rapid Prototyping Journal, Vol. 26 No. 5, pp. 881-894. The final authenticated version is available online at: <http://dx.doi.org/10.1108/RPJ-03-2019-0070>

This content is provided under [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/) license



Direct texturing for additive manufacturing: software support and build tests

Antonio Armillotta

Dipartimento di Meccanica, Politecnico di Milano

Via La Masa 1, 20156 Milano, Italy

E-mail: antonio.armillotta@polimi.it

Tel.: +39 02 23998296

ORCID: 0000-0003-2619-0669

Abstract

Purpose – The paper investigates on the feasibility of adding macro-textures to triangle meshes for additive manufacturing, focusing on possible time and quality issues in both software processing and part fabrication.

Design/methodology/approach – A demonstrative software tool was developed to apply user-selected textures to existing meshes. The computational procedure is a 3D extension of the solid texturing method used in computer graphics. The tool was tested for speed and quality of results, considering also the pre- and post-processing operations required. Some textured meshes were printed by different processes to test build speed and quality.

Findings – The tool can handle models with realistic complexity in acceptable computation times. Parts are built without difficulties or extra-costs achieving a good aesthetic yield of the texture.

Research limitations/implications – The tool cannot reproduce sample patterns but requires the development of a generation algorithm for each different type of texture. Mesh processing operations may take a long time when very fine textures are added to large parts.

Practical implications – Direct texturing can help obtain parts with aesthetic or functional textures without the need for surface post-treatments, which can be especially difficult and expensive for plastic parts.

Originality/value – The proposed method improves the uniformity and consistency of textures compared to existing approaches, and can support future systematic studies on the detail resolution of additive manufacturing processes.

Keywords

Additive manufacturing; mesh processing; texture; solid texturing; detail resolution.

This is a post-peer-review, pre-copyedit version of an article published in Rapid Prototyping Journal. The final authenticated version is available online at:

<https://doi.org/10.1108/RPJ-03-2019-0070>

1 Introduction

The high geometric complexity allowed by additive manufacturing (AM) techniques allows designers to add unique features to products. These include surface macro-textures, which are increasingly used for aesthetic and functional purposes (Evans and Bryan, 1999; Coblas *et al.*, 2015). As an alternative to post-processing treatments, AM parts can be directly textured by adding surface detail to the triangle meshes in STL format used as input for the build process (Fig. 1). This is becoming easier for several reasons. First, the ability to manage large sized digital models is supported by steadily increasing hardware capabilities. Secondly, AM machines have been greatly improving in accuracy and resolution, and can build parts with smaller geometric features than in the past.

Despite this progress, the use of textured models in AM is still limited. A first reason is related to the software support required for the treatment of digital models. Available tools are targeted to 3D rendering and animation, and consist of virtual clay modeling controls matching the workflow of graphic designers. They allow skilled users to get striking results, but require long interactive work sessions. This is a serious drawback in manufacturing applications, which need good-quality results in short times. For such purposes, a texturing tool should allow the user to select a texture, apply it to the model automatically, get an immediate feedback of the visual effect, and fine-tune the result by calibrating a set of geometric parameters. This is currently done by software tools for special applications (e.g. plaque or sign engraving), and are usually limited to flat or mildly curved surfaces; their application to generic 3D shapes may result into distorted textures with degraded visual effect.

Another difficulty is related to the manufacturing process. A faithful reproduction of textures on physical objects requires a process with good detail resolution. Fine textures call for high-resolution AM techniques, which are generally slow and expensive; their use is limited to small parts for both technical and economic reasons. Larger parts may be built by faster AM techniques provided that layer thickness and other build parameters are adequately reduced; again, an heavy burden is imposed to build time and cost.

To overcome the above difficulties, this study proposes a process chain for the direct texturing of AM parts. The work has the following objectives:

- develop a software tool for the texturing of STL models, which can streamline the task and improve the quality of generated textures;
- validate the method with different AM techniques to verify the actual response times and quality;
- assess the limits of the trade-off between texture resolution and object size considering the economic constraints of target applications.

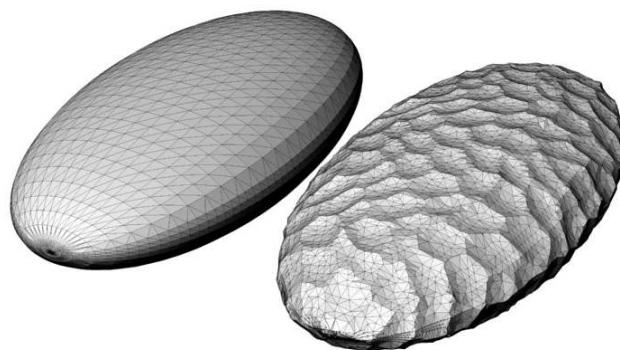


Fig. 1: Adding texture to a STL model

2 Background

The texturing of AM parts was first proposed for biomedical applications, since it has been recognized that surface patterns can improve the adhesion of orthopaedic implants to biological tissues. In (Sachs *et al.*,

1994), a ceramic mold for cast implants is built by a binder jetting process; the difficult-to-machine material and the mushroom-like pattern designed for the mold would have posed difficulties to alternative process solutions such as etching or, more recently, laser texturing. Similar tests are reported in (Curodeau *et al.*, 2000) with different patterns and implant geometries.

A wealth of new uses of direct texturing in AM has recently emerged in literature. Some of these exploit technical properties of textured surfaces. Reduced or enhanced coefficients of friction, as well as bio-inspired tribological behaviors, are achieved in (Hong *et al.*, 2017) on plastic parts, and in (Holovento *et al.*, 2018) on metal parts. In (Jafari *et al.*, 2019), textures created by high-resolution AM processes are shown to control the wettability of surfaces and approach a superhydrophobic behavior. Structural properties may also be enhanced by properly designed surface texture: (Ko *et al.*, 2019) show that Voronoi-like patterns increase the bending strength of plastic parts, while (Johnson *et al.*, 2018) fabricate prototypes of protective armor with bio-inspired patterns that improve resistance to blade penetration.

Other applications are aimed to controlling perceptual properties of AM products or prototypes. In (Mai *et al.*, 2016), grain patterns are created on plastic parts to imitate the look and feel of wood furniture. In (Ou *et al.*, 2016), hair-like patterns are created on plastic parts either for decorative purposes and to improve tactile sensing. In (van Rompay *et al.*, 2017), regular patterns on drinking cups are shown to improve the perceived quality of contained beverage. (Mantihal *et al.*, 2019) apply several types of texture on chocolate specimens to evaluate their influence on taste quality. (Spahiu *et al.*, 2017) show that the drape of garment can be improved by properly arranging 3D printed patterns of different sizes and shapes.

In most of the above studies, textures are applied to digital models through interactive modeling. Attempts have also been made to automate the texturing operations. This is done in different ways depending on the geometric complexity of the parts; the task is obviously easier when the target surface is topologically similar to the rectangular area where the texture is usually defined.

When the objects have simple primitive surfaces (planes, cylinders), the texture can be generated from analytic functions. In (Maekawa *et al.*, 2003), autocorrelated random patterns ($1/f$ fluctuations) are applied to STL models to get decorative effects on AM parts. In (Armillotta, 2006), simple patterns generated by mathematical equations are added to flat specimens in order to test the detail resolution of a specific AM process. Procedures based on mesh deformation are used in (Kanada, 2015) to create surface waviness in generative artwork, which might also be applied to macro-textures with lower scales and aspect ratios.

Partially automated texturing procedures have been proposed for moderately complex surfaces. In (Jee and Sachs, 2000), the texture is defined on a parametric surface patch which is then interactively mapped to a region of the triangle mesh. In (Zhou *et al.*, 2014), a pattern defined on a curve is replicated into a 3D model with interactively created geometry, which can be composed with other models to get complex manufacturable parts. In (Dumas *et al.*, 2015), an exemplar texture is replicated into a carved pattern on a skin of the part; the procedure projects texture images from randomly oriented planes around the part to a voxel structure on part surface, selecting the plane for each voxel so as to optimize the matching of the input texture. (Pintus *et al.*, 2010) and (Herholz *et al.*, 2017) propose shape enhancement procedures in order to improve the sharpness and visibility of given textures on 3D printed parts.

Process-level approaches have been proposed to add simple textures to triangle meshes of any complexity. In (Kobayashi and Shirai, 2008), a periodic pattern is added to a CNC-machined part during toolpath generation; this is done by assembling spiral toolpaths generated on elementary shapes resulting from segmentation of the input pattern. At an earlier step in the AM workflow, (Yaman *et al.*, 2019) converts slicing contours into binary images, which are then modified through morphological operations corresponding to regular patterns. Adaptive toolpath generation methods proposed in (Jin *et al.*, 2011; Jin *et al.*, 2013) for models with extreme geometric details might also find applications on textured models.

Further progress in AM texturing methods would be achieved by developing an automatic procedure capable of treating a wider range of geometries. These should include triangle meshes of complex shape, and textures consisting of regular and random features for realistic visual effects. A similar objective has long been pursued in the field of computer graphics, where realistic models of natural and artificial objects are

generated for rendering and animation purposes. In that context, a texture is defined as color variation among surface points and does not involve real geometric detail as required in manufacturing applications. However, methods developed for 2D texturing can in principle be extended to 3D texturing.

Texture mapping, the basic approach to 2D texturing (Weinhaus and Devarajan, 1997), consists in picking or interpolating colors from a digital image. This gives high flexibility with a possible compromise on quality, as the sample image is a planar view of the texture on a limited area; when mapped on a freeform surface, the color pattern may warp and include undesired artefacts such as lack of consistency between adjacent surfaces or false patterns in random textures.

An extension of texture mapping, referred to as displacement mapping, has been the first achievement in the development of 3D texturing methods (Szirmay-Kalos and Umenhoffer, 2008). Rather than color variations, the method generates actual height variations on part surfaces. This allows highly undulated surfaces to be displayed with increased realism by revealing shadows and irregular silhouettes which would be missed by color-texture maps. Mesh points are displaced along local surface normals (Cook, 1984; Bidasaria, 1992), although different perturbation strategies have been proposed for particular types of textures (Pedersen, 1994; Jang and Han, 2013; Nießner and Loop, 2013). Displacements require a finer surface sampling than colors; to overcome computational difficulties in graphical applications, a finer patch can be superimposed to a coarse mesh at rendering time (Jung and Lee, 2015). Displacement mapping has the same quality issues of texture mapping and its use is restricted to simple part geometries; (Schroeder *et al.*, 2005) reduce distortion by generating height fields from purely stochastic models whose parameters may fit surface patterns of physical samples.

In 2D texturing, the shortcomings of plain texture mapping are avoided through an alternative method referred to as solid texturing (Dischler and Ghazanfarpour, 2001; Pietroni *et al.*, 2010). Colors are associated with the spatial positions of the points of an object regardless of the shape of the surfaces they belong to. This improves the consistency of the texture at edges and its uniformity on highly curved surfaces, thus allowing more realistic effects in graphical visualization. In the lack of a source image, a spatial color field has to be created around the object. In most cases, this is done by procedural methods, which generate the color field by an algorithm that is specifically developed for each texture. The approach has allowed to imitate a wide range of periodic and random textures (Ebert, 1998). In (Gardner, 1984), terrain and clouds are generated by summation and multiplication of sine waves. In (Peachey, 1985), wood and stone surfaces are imitated by virtually bombing the object with particles of various geometry. In (Perlin, 1985), the solid noise concept is introduced for the random perturbation of periodic functions, allowing the reproduction of textures with both deterministic and random components; improved formulations and implementations of solid noise are proposed in (Lewis, 1989; Lagae *et al.*, 2010). In (Cutler *et al.*, 2002), procedures are proposed for heterogeneous solid models made up of parallel layers possibly perturbed by noise.

In this work, textures are generated on STL models through a 3D extension of the solid texturing method. Such approach is aimed to developing a software tool that is simple yet comparable to existing ones in terms of quality and automation level. This is intended to provide a testbed for the evaluation of the technical and economic potential of direct texturing which is the core objective of the paper. However, the computational method is not proposed as a long-term solution to the problem, considering that more advanced approaches are being proposed in the field of computer graphics in order to further improve performance and flexibility. These methods are briefly cited in Tab. 1, which classifies them into two categories:

- 2D analytic methods, which start from exemplar images and turn them into spatial color fields enclosing the part (Wei *et al.*, 2009); most of them are extensions of prior methods for the synthesis of texture images (Haralick, 1979); their 3D extension has not been attempted yet;
- 3D geometric texturing methods, which are similar to CAD modeling and bear no direct relationship to 2D approaches; they have the potential to improve the reproduction of exemplar textures and solve difficult problems such as the visualization of soft materials (e.g. hair, fur); so far, however, complexity issues have discouraged their use in the manufacturing context.

Tab. 1: Advanced approaches to texturing

<i>Type</i>	<i>Method</i>	<i>References</i>
Analytic (2D)	Spectral analysis	Ghazanfarpour and Dischler, 1995
	Image pyramid transform	Heeger and Bergen, 1995
	Reaction diffusion	Turk, 1991
	Stereological methods	Jagnow <i>et al.</i> , 2004; Chiou and Yang, 2006; Du <i>et al.</i> , 2013; Shu <i>et al.</i> , 2014
	Grey-level aura matrices	Qin and Yang, 2007; Wei and Levoy, 2001
	Energy function	Kopf <i>et al.</i> , 2007; Chen and Wang, 2010
	Lazy synthesis	Dong <i>et al.</i> , 2008
	Morphing and warping	Pietroni <i>et al.</i> , 2007
	Lapped solid textures	Takayama <i>et al.</i> , 2008
	Texture evolution	Chen and Ip, 2004
Sketch-based texturing	Zhang <i>et al.</i> , 2011	
Geometric (3D)	Hypertexture	Perlin, 1989; Gilet and Dischler, 2009
	Texel mapping	Kajiya and Kay, 1989
	Cellular growth	Fleischer <i>et al.</i> , 1995
	Macrostructures	Dischler and Ghazanfarpour, 1999
	Image analogies	Bhat <i>et al.</i> , 2004
	Mesh quilting	Zhou <i>et al.</i> , 2006
	Height-and-tilt textures	Andersen <i>et al.</i> , 2009

3 Approach

This paper describes a method for the 3D texturing of triangle meshes, whose basic concepts were first presented in (Armillotta, 2004). The development of the method was driven by the attempt to provide software support with enhanced automation level, which could streamline the designer’s task while improving the quality of output meshes compared to displacement mapping. Over time, the increasing availability of mesh processing software for AM applications has allowed to integrate the method in a workflow that gives the user additional control on the final result.

The method is a 3D extension of the solid texturing method, as it applies displacements to mesh vertices according to their spatial position regardless of surface shape. This is different from displacement mapping, which starts from a height map defined on a reference area, and transforms it to fit the surface of the object. Mapping defects may arise due to the possible mismatch between the reference area, which is generally a flat rectangle, and the target surface. The latter can include flat or curved faces whose shapes are topologically different from the rectangle. If the texture includes regular features arranged in an orderly manner, their appearance may be warped. Furthermore, the need to map the texture separately on adjacent surface patches may create inconsistencies at the edges. These defects are avoided by solid texturing due to the following reasons:

- The height field is defined at all the points of the space that contains the object. Therefore there is no need to match points from a reference to a target, and the height features can preserve their uniform size and distribution as defined at spatial level.
- Points on opposite sides of an edge are displaced from the same spatial height field, rather than as a result of two mapping operations on separate patches. Texture features thus retain their consistency across the edge.

These advantages are illustrated in Fig. 2, where two textures including regular features (parallel marble veins, regular dots arranged in a close packing) are applied to the same object with flat triangular faces separated by sharp edges. Once the displacement fields have been defined in the space around the objects,

the features appear undistorted both in size and in distribution, and perfectly consistent across the edges. As will be shown in later examples, the inevitable warping on curved surfaces is also kept within visually acceptable limits.

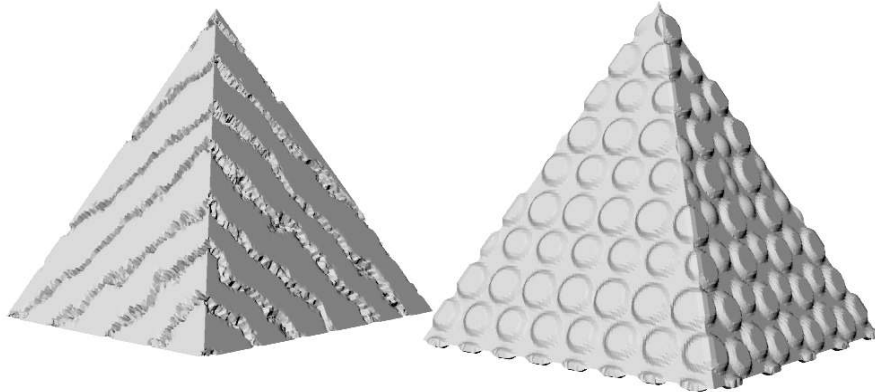


Fig. 2: Uniformity and consistency of vertex displacements

Another advantage of the solid texturing approach is that there is no need to arrange the vertices in regular grids as would be required by displacement mapping. This allows to use existing STL models where no access is provided to a source CAD model. However, three additional difficulties arise compared to 2D solid texturing:

- The triangle mesh must have high resolution to match the possibly fine detail of the texture; therefore, it is usually necessary to increase the resolution of an existing STL model. This is a straightforward task using available mesh processing tools, but it must be verified that the operation does not involve excessive delays and computational difficulties.
- Each vertex is displaced along the local surface normal; in a STL model, a vertex is shared by different triangles, and the normal vectors are generally different (especially across edges). By randomly choosing the normal of one of the triangles adjacent to a point, the displacement could cause further distortion of the texture. To avoid that, the displacement direction is calculated prior to displacement by averaging the normals to adjacent triangles, stored as explicit information in the STL model. The relational data of the mesh are saved as vertex lists for each triangle, in order to allow the generation of an output STL model. Again, the correctness and computational cost of the operation must be verified.
- The proposed approach is of the procedural type, i.e. a texture is translated into a computational procedure that creates an appropriate volumetric map of displacements. Although this need cannot be reasonably satisfied for all textures of practical interest, a texture catalogue has to be incrementally built for specific applications. Each procedure should allow to customize the design of the texture by setting its dimensions (e.g. width, height, spacing) and possible texture-dependent geometric attributes (e.g. orientation, randomness, coverage, directionality, turbulence). Procedures developed for 2D solid texturing may be extended for this purpose, but the actual results must be evaluated for visual quality and realism.

4 Computational procedures

The vertices of the input mesh have positions \mathbf{p} and normals \mathbf{n} in a coordinate system XYZ. The textured mesh is calculated by displacing each point by an appropriate distance d along the normal:

$$\mathbf{p}' = \mathbf{p} + d\mathbf{n}$$

According to the solid texturing approach, d depends only on the position of the point:

$$d = f(\mathbf{p}) = f(x, y, z)$$

Function f is evaluated by an algorithmic procedure designed for a specific texture type. In this work, a set of procedures has been designed by extending graphical algorithms for 2D solid texturing described in (Ferguson, 2014). The extension consists in replacing color levels with height levels defined in a proper scale, and in tuning appropriate modeling parameters in order to achieve the desired ranges of visual effects. This has been done for two main classes of procedures, which are described in more detail below.

4.1 Point location

Many textures are based on regular arrangements of elementary features. They can be generated by defining a spatial lattice of spheres filling the space around the part. Each point of the mesh is located with respect to the lattice by finding its distance to the center of the closest sphere. The displacement is then calculated as a function of distance, possibly modified by deterministic or random transformations.

As a sample application of point-location procedures, Fig. 3 shows some ‘scaly’ textures resembling the natural patterns found in fish or reptile skins. They are generated very similarly, with differences concerning the displacement direction (Figs. 3a-b), the size and height of the features (Fig. 3c), and the degree of waviness in the arrangement of features (Fig. 3d).

To generate such textures, the spheres used for point location are arranged as the vertices of a hexagonal close packed (HCP) lattice (Fig. 4a). The input feature size s determines the dimensions (c_x, c_y, c_z) of the HCP cell along the three coordinate axes, and the radius a of the spheres:

$$c_x = 2s, \quad c_y = c_z = \sqrt{3}s, \quad a = s$$

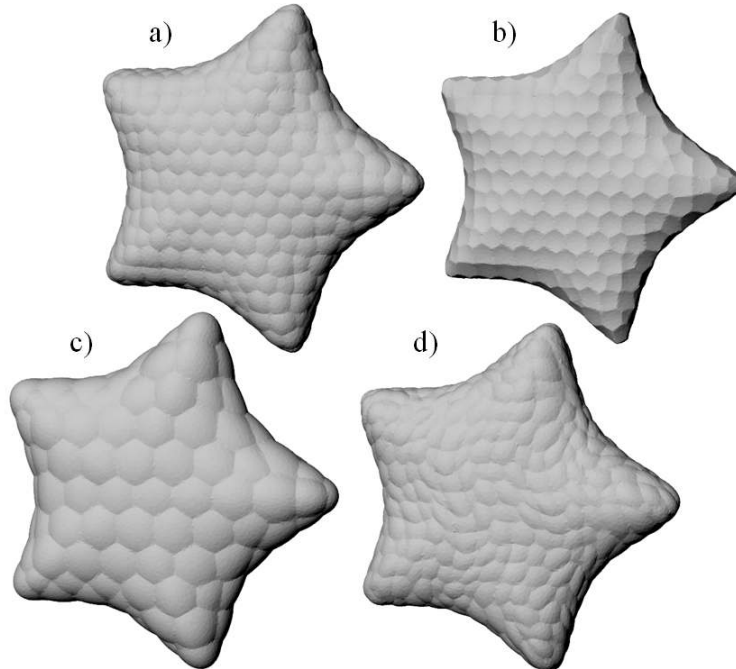


Fig. 3: Scaly textures: a) convex, b) concave, c) larger features, b) wavy pattern

A point with coordinates (x, y, z) is located by calculating the position (x_c, y_c, z_c) relative to the cell it belongs to:

$$(x_c, y_c, z_c) = (x \bmod c_x, y \bmod c_y, z \bmod c_z)$$

and the distance r from the nearest vertex of the lattice (Fig. 4b):

$$r = \sqrt{x_c^2 + y_c^2 + z_c^2}$$

If $r < a$, the displacement of the point is evaluated from the input feature height h by the following equation (Fig. 4c):

$$d = h \cdot \left(\sqrt{1 - r^2/a^2} - 1/2 \right)$$

To get concave features, the displacement d is changed in sign. If a given degree of waviness $w \in [0, 1]$ is specified, the position of the point is perturbed through the following periodic function:

$$(\delta x, \delta y, \delta z) = ws \left(z/s, 1.5 z/s, 2 y/s \right) \bmod 2\pi$$

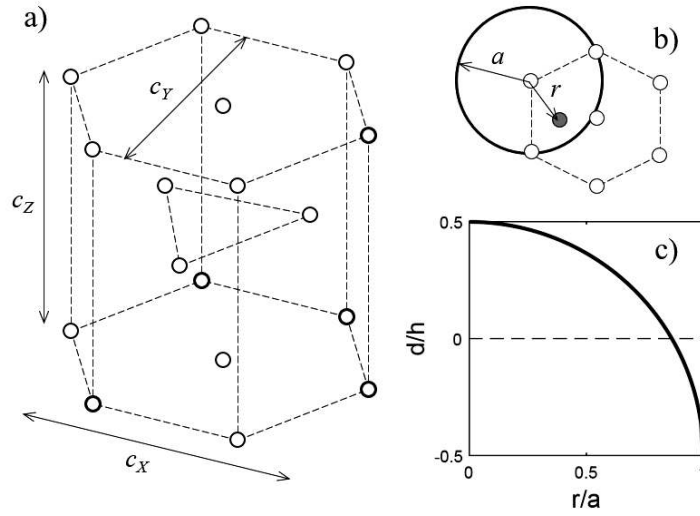


Fig. 4: Generation of a scaly texture: a) HCP lattice, b) point location, c) displacement

4.2 Solid functions

Some natural-looking textures have less structured appearance, with features at larger scales (or no features at all) and mostly random geometry. They can be generated by analytic functions, which include deterministic terms for feature generation along with stochastic models to add roughness or waviness. The functions used here are extensions of the stochastic solid functions used in graphics (Ferguson, 2014), which are designed so as to ensure autocorrelation, i.e. statistical dependence among values at adjacent points. Unlike other types of stochastic models proposed for texture generation, they only depend on point positions without any assigned ordering or prior knowledge of adjacency relationships.

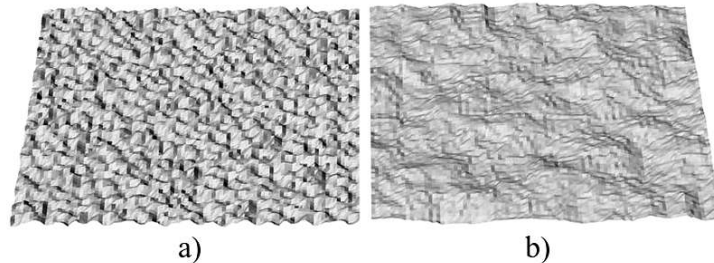


Fig. 5: Examples of turbulence functions: a) T_1 , b) T_4

A solid function is generally built upon the basic turbulence function $T_k(\mathbf{p})$, which maps points \mathbf{p} in the XY plane to autocorrelated random numbers. These are calculated by adding k low-pass filtered white noise components with increasing frequency and decreasing amplitude (Fig. 5). Operations on the turbulence

function include amplification, uniform and anisotropic scaling, translations and rotations, thresholding at upper and lower bounds, and superposition to deterministic displacements.

As an example, solid functions can be used to generate ‘wood grain’ textures (Fig. 6a) with several attributes related to the scale and position of the features (Fig. 6b), the spatial orientation of the pattern (Fig. 6c), and the deformation of the features into directional and random shapes (Fig. 6d).

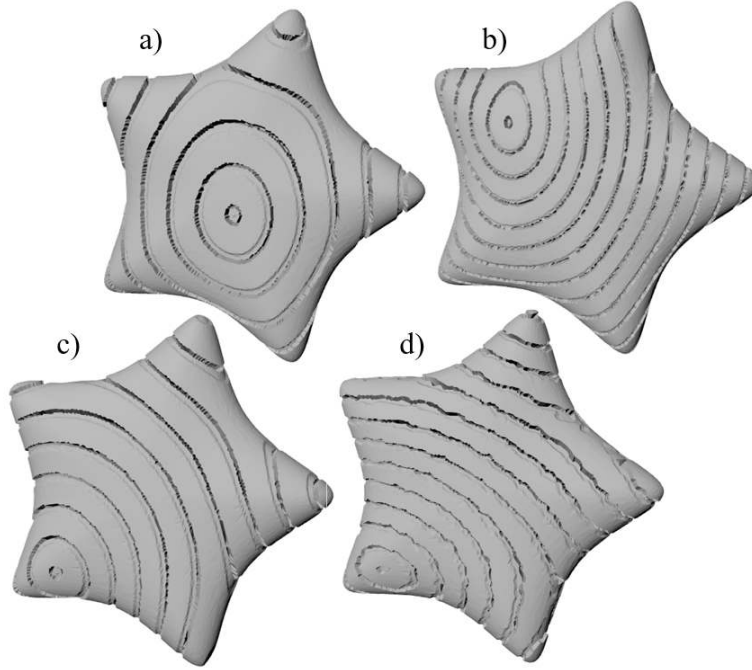


Fig. 6: Wood grain textures: a)-b) different scales and positions, c) orientation, d) directionality and randomness

The above textures include concentric grooves whose input dimensions are the width w , the depth h , and the spacing s . They are generated by the following displacement function:

$$d = -h \cdot f(r, w, s)$$

where $r = (x^2 + y^2)^{1/2}$ is the distance of a point from the Z-axis of the coordinate frame, and f is a trapezoidal waveform with period s and nonzero values for intervals of length w (Fig. 7a). Before the displacement is applied, point coordinates are transformed according to some additional input parameters. These include the rotation angles α , β and $\gamma \in [0, 90^\circ]$ about axes X, Y and Z, and a translational distance $\delta \in [0, 5s]$ along each of the three axes, which are used for the following rigid-body transformation (Fig. 7b):

$$(\mathbf{p}')^T = \text{rot}(X, \alpha) \cdot \text{rot}(Y, \beta) \cdot \text{rot}(Z, \gamma) \cdot \mathbf{p}^T + [\delta, \delta, \delta]^T$$

where $\text{rot}(\dots)$ are 3×3 rotation matrices on column vectors. Then an anisotropy parameter $\rho \in [0, 1]$ is used for scaling along the X-axis (Fig. 7c):

$$(\mathbf{p}'')^T = [(1 + \rho/2), 1, 1] \cdot (\mathbf{p}')^T$$

Lastly, a turbulence parameter $\tau \in [0, 1]$ applies random translations to the points (Fig. 7d):

$$(\mathbf{p}''')^T = \tau \cdot [T_1(\mathbf{p} + \boldsymbol{\pi}_x), T_1(\mathbf{p} + \boldsymbol{\pi}_y), T_1(\mathbf{p} + \boldsymbol{\pi}_z)] \cdot (\mathbf{p}'')^T$$

where three different random values are obtained for the three axes by evaluating the turbulence function T_1 (interpolated white noise) on each point after perturbing it with different translation vectors $\boldsymbol{\pi}_x$, $\boldsymbol{\pi}_y$, $\boldsymbol{\pi}_z$.

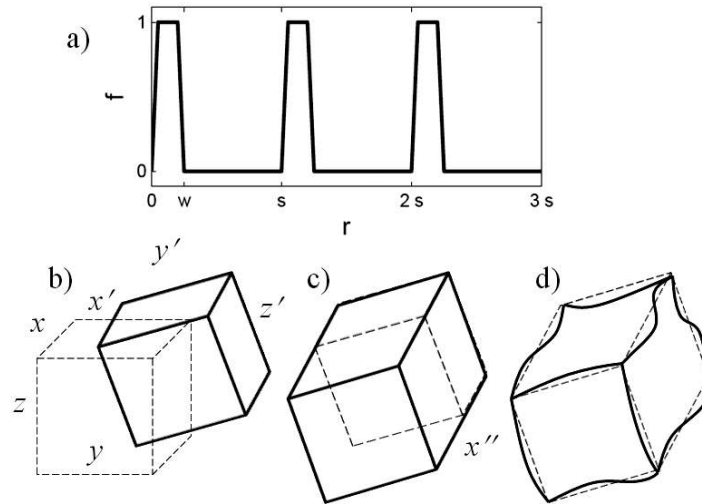


Fig. 7: Generation of a wood grain texture: a) displacement, b) translations and rotations, c) anisotropic scaling, d) random scaling

5 Workflow

The 3D solid texturing method has been implemented into a software tool for proof-of-concept and testing purposes. The use of the tool requires some pre-processing on the input mesh as well as post-processing on the output mesh. These operations are carried out by one of the available mesh processing software tools, such as Autodesk Meshmixer that was used in this work. The flowchart in Fig. 8 outlines the steps of the workflow.

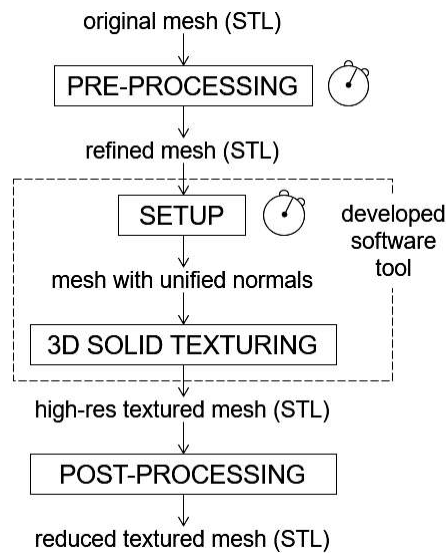


Fig. 8: Texturing workflow

The original STL file is likely to have lower resolution than needed for texturing, with no access allowed to the raw geometric data (e.g. a CAD model) in order to change meshing parameters (Fig. 9a). Therefore, the mesh has to be refined by either triangle subdivision or full remeshing (Fig. 9b). In either option, the maximum size of the triangles should be set to a proper value related to the layer thickness of the AM process (typically 0.05-0.25 mm). Remeshing allows a more uniform triangle size, thus keeping file size to a minimum for a given spacing of points; as a lower bound, a triangle mesh on a regular point grid with spacing l (in mm) would have a binary STL file size around $(0.01 / l^2)$ MB/cm².

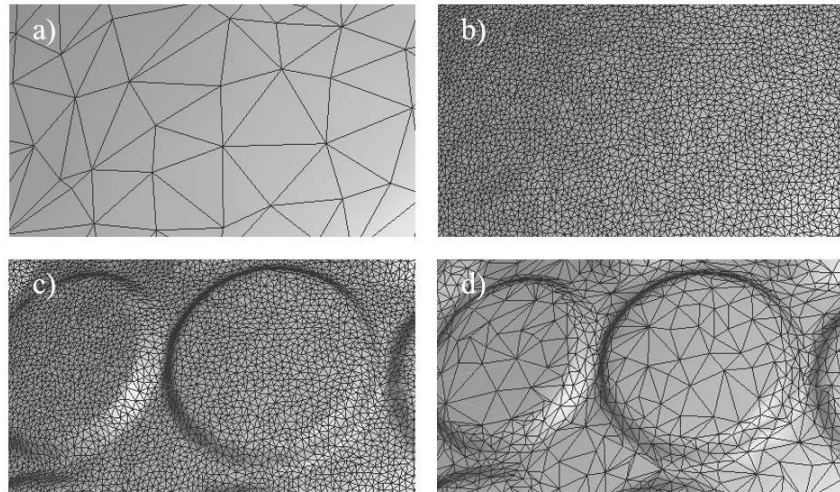


Fig. 9: Mesh processing steps: a) original, b) refined, c) displaced, d) reduced

The refined STL file is processed by the developed software tool, which prepares the mesh by calculating the average normal for each point. To avoid repeating the setup of the mesh, the vertex list is stored in a separate file and automatically recalled for later uses. Then the mesh is interactively textured through dialog templates (Fig. 10) where users can select texture type and parameters and preview the result. The textured mesh is eventually saved into an output STL file (Fig. 9c).

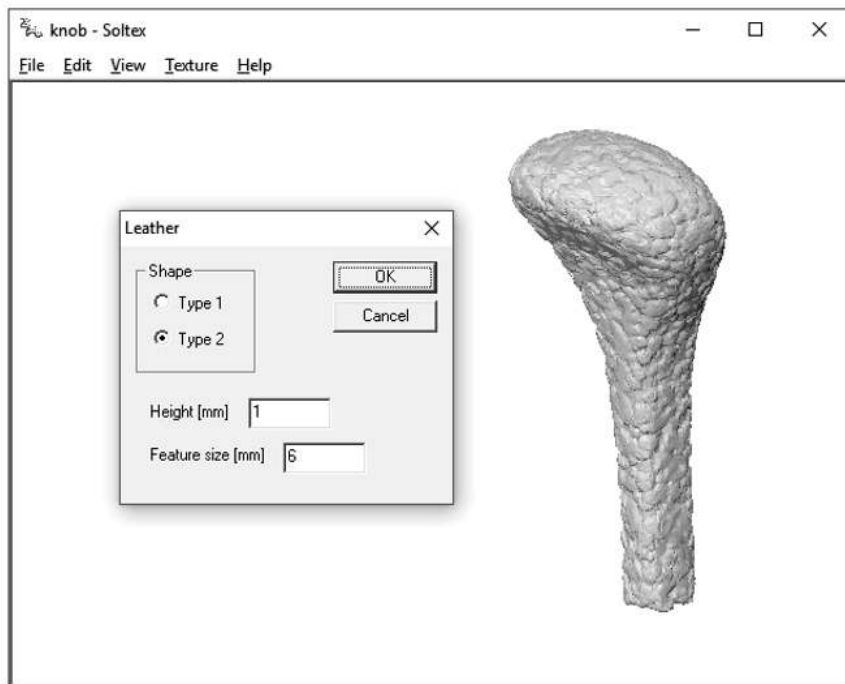


Fig. 10: User interface of the texturing tool

To avoid difficulties in the manufacturing phase, mesh processing software is again used to reduce the triangle count with a tolerance on point coordinates (Fig. 9d). This operation usually allows file size reductions of 50-80% without visible alteration of mesh appearance. The final STL file is ready for the AM process without any need for repair.

As highlighted in Fig. 8, only few seconds of computation time are required by the texturing and the post-processing steps, while the first two steps of the workflow may take considerably longer for meshes with large storage size. Fig. 11 shows the results of speed tests made on a low-end PC on file sizes from 2 to 350 MB. The files were obtained by remeshing a 20-mm cube with 1.5k triangles (70 kB). Pre-processing and

mesh setup times grow respectively with powers of about 1.3 and 2 with the size of the STL file; the actual computational complexity of the mesh setup operation is probably lower, and a better performance could be obtained with a more efficient software implementation. For file sizes within 100 MB, the sequence of the two operations does not exceed 2 hours of calculation. Processing larger files may require high-performance computing resources.

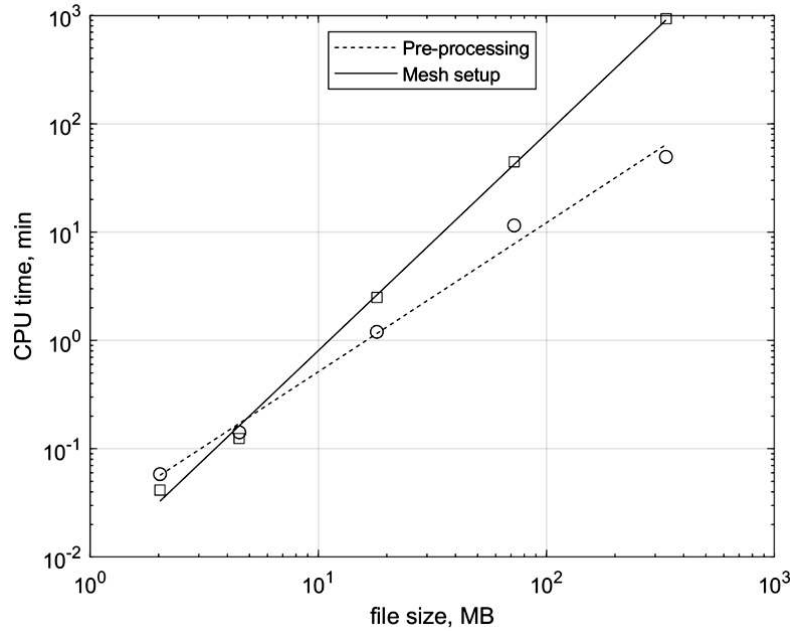


Fig. 11: Computation time for pre-processing and mesh setup

6 Results

The proposed solution for direct texturing has been tested against the constraints posed by AM processes. These include the possible difficulties when dealing with complex digital models, and the limits in detail resolution due to layer-by-layer fabrication. The tests were intended to provide suggestions on how to achieve a good balance among the detail size of the texture, the resolution of the mesh, and the layer thickness of the chosen process.

Coarse textures are likely to be applied to relatively large parts to be printed with fast process settings. In these cases, mesh resolution and process parameters need to be fine-tuned in order to balance geometric accuracy and manufacturing cost. In one test case, a textured model was fabricated using two processes with different resolutions, both using thermoplastic materials: Fused Filament Fabrication (FFF, extrusion of a filament through a heated nozzle) and Selective Laser Sintering (SLS, melting and bonding of a powder bed by a laser beam).

A model with a size of $170 \times 85 \times 30$ mm (Fig. 12a) was remeshed at a resolution of 0.25 mm into a 40-MB STL file. The texture added to the model (Fig. 12b) consists of flat depressions with random diameters of 5.0 ± 1.5 mm, and was generated by a point location procedure. The textured mesh was eventually reduced to the 35% of its triangle count.

Fig. 13a shows an FFF print of the textured model in polylactic acid (PLA). The part was manufactured on a BlueTek 3D Strato Pro 350 printer with a layer thickness of 0.2 mm, a 10% infill, and a 2 mm skin thickness. The build time was about 8 hours, not significantly longer than would have been required for an untextured model. The texture is reproduced faithfully enough despite a minor smoothing of feature edges. At a closer look, horizontal surfaces appear moderately aliased by the staircase effect, albeit not so much as to jeopardize the aesthetic yield of the texture.

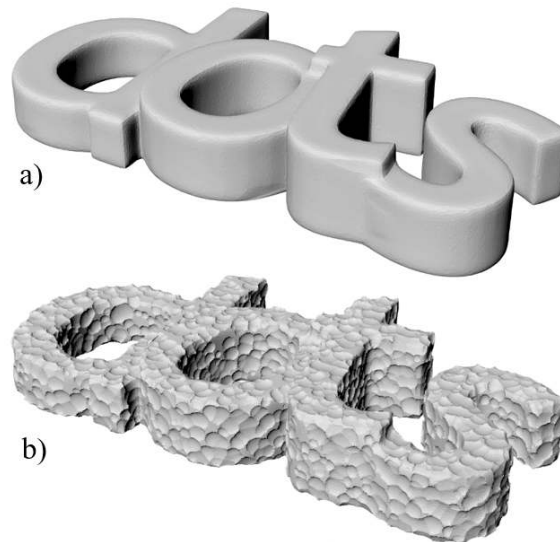


Fig. 12: Coarse texture on a large part: a) original mesh, b) textured mesh

The STL model was edited for the requirements of the SLS process by hollowing the volume at a thickness of 3 mm and adding some draining holes on the bottom plane. Fig. 13b shows the SLS print of the model in polyamide 12 (PA12), made on an EOS Formiga P100 system with a layer thickness of 0.1 mm. The build was part of a batch, where the share of build time for the model can be estimated in about 1 hour. The details are reproduced crisply on the whole surface of the part, and the artefacts resulting from the powder bed fusion process (grainy surfaces, layer marks) are on a much smaller scale than texture features.

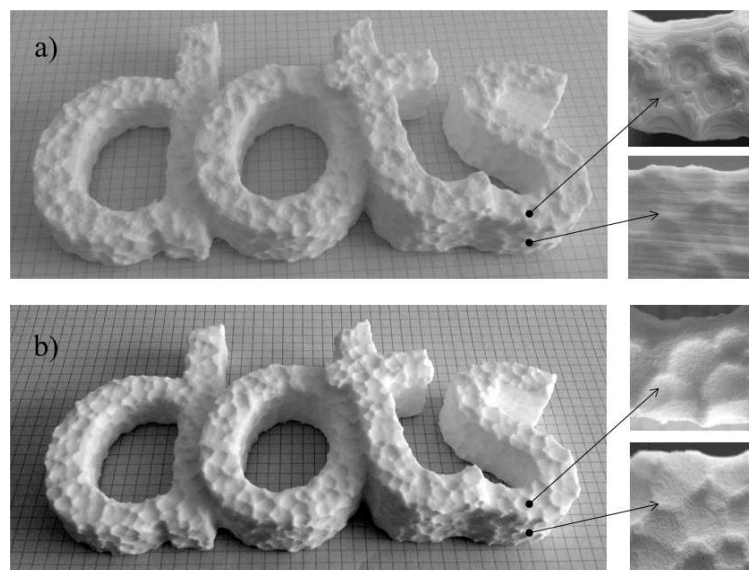


Fig. 13: Coarse texture on a large part: a) FFF sample part; b) SLS sample part

Finer textures are more suitable to small parts, which can be economically manufactured by higher-resolution processes. Such chance was tested in a second test case on the Polyjet process, a material jetting technique where a liquid photopolymer is deposited by a multi-nozzle printhead and cured by a UV lamp; the final layer thickness is obtained after a slight machining pass after curing.

A model with a size of $40 \times 20 \times 30$ mm (Fig. 14a) was remeshed at a resolution of 0.10 mm (Fig. 14b) into a 17-MB STL file. Two different textures were then added to the mesh. The first one (Fig. 14c) is generated by a point-location procedure and includes uniformly spaced raised dots with 4 mm diameter and 0.5 mm height. The second one (Fig. 14d) is generated by solid functions and includes grooves imitating marble veins with 4 mm spacing and 0.5 mm depth, with a moderate turbulence and an orientation interactively

chosen by two rotation parameters (pitch and yaw). Both meshes were eventually reduced to 20% of their triangle count.

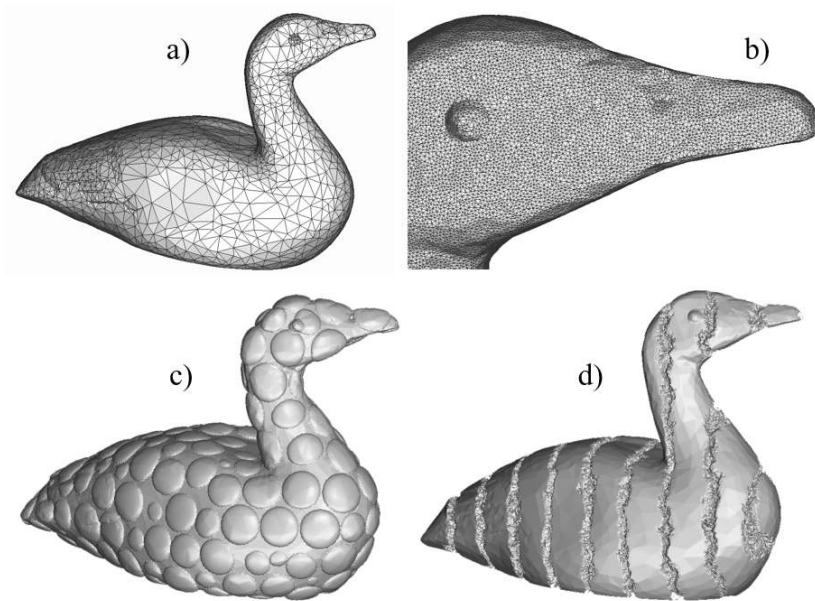


Fig. 14: Fine texture on a small part: a) original mesh, b) refined mesh; textured meshes with c) regular dots, and d) marble veins

Figs. 15a and 15b show the parts built from the two textured meshes on an Objet Eden 330 system in Vero White Plus resin with layer thickness of 0.016 mm. The two parts were made in the same build run in 2.5 hours, without any increase of estimated build time compared to the untextured model. The quality of surface details is fully acceptable all around the parts even in magnified details; thanks to the properties of the wax-based support material, no damaging or loss of resolution occurs on the overhang surfaces.

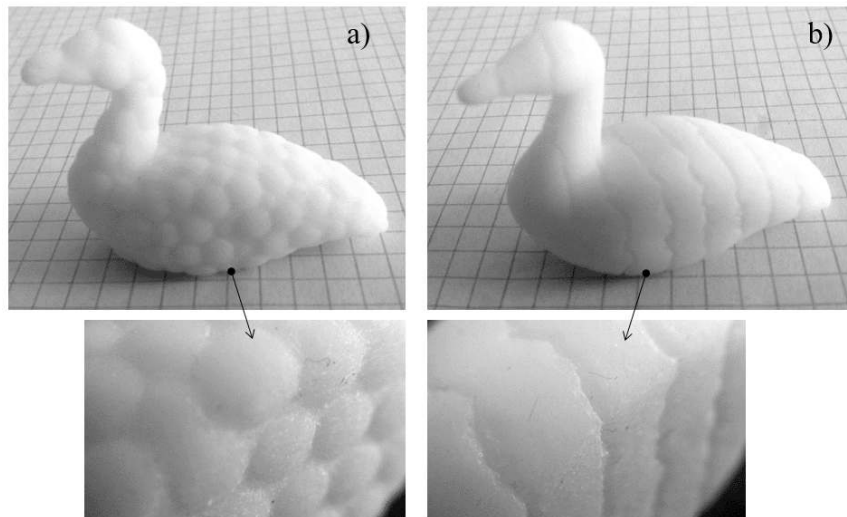


Fig. 15: Fine texture on a small part: Polyjet parts with a) regular dots, and b) marble veins

The above results suggest that the limits of the proposed process chain are more related to computational issues than to constraints of the build process. This consideration seems to be confirmed by the diagram in Fig. 16, which illustrates the trade-off between geometric size (expressed as the surface area of a mesh) and detail size (expressed as the maximum triangle size). The contour plot shows the STL file size as a function of the two variables, under the assumption that the mesh is a regular grid of triangles; as observed in the test cases, such condition is fairly well approximated by the remeshing operation. The 300-MB limit corresponds

to the most severe conditions tested in this work, while the 100-MB limit corresponds to those situations where the workflow is fairly rapid even with low-performance computing hardware. Within these limits, a texture requiring resolutions of 0.2-0.3 mm (as in the case in Fig. 12) can be applied to objects with area of 1500-2000 cm², which are likely to be printed on AM systems with medium resolution (0.1-mm layer thickness) and build size of about 500 mm. Under the same conditions, a finer texture requiring a 0.1-mm resolution (as in the case of Fig. 14) can be applied to objects with area of 200-300 cm², in the upper range of the build size of high resolution AM systems (layer thickness of 0.02-0.05 mm).

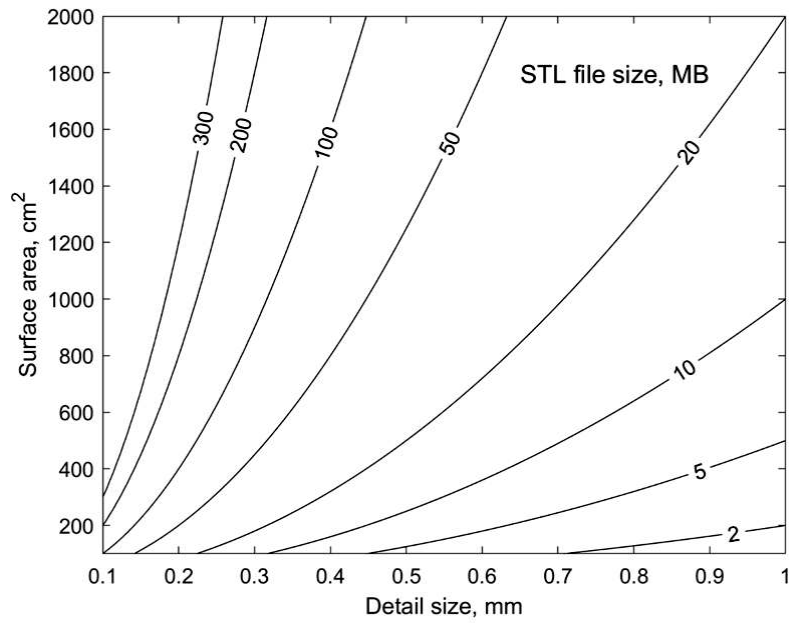


Fig. 16: Trade-off between model size and mesh detail

7 Conclusions

The paper has dealt with the feasibility of adding surface texture to digital models for AM applications. The aim of the work was twofold: develop a software tool able to effectively assist the designer's task, and verify the quality achievable in the manufacturing process.

On the computational side, the proposed procedural approach led to the development of a software tool where the texture is interactively chosen in a set of available types, then defined through context-dependent dimensional and shape parameters, and eventually generated on the surface of a triangle mesh in STL format. The generation method is only partially new, as it is based on the 3D extension of the solid texturing method available in literature for graphical applications. It has been verified that the method overcomes the distortion and inconsistency issues of classical displacement mapping; its limitation, however, is the need to develop a specific generation algorithm for each new texture. The workflow proved to be smooth and capable of obtaining good quality, print-ready models. The very fine mesh required came out to be manageable with remeshing operations using third-party software; these required reasonable times even on models of size comparable to the working space of AM systems. The setup operation that calculates the displacement directions at the points of the mesh appears as the main bottleneck of the process chain when handling very complex models; however, the reported tests suggest that the issue seems to be limited to special cases where fine textures are to be applied to large objects. This difficulty could be overcome with a more efficient software implementation, which will also try to integrate the entire workflow into a single tool. To improve the flexibility of the direct texturing procedure, future developments will be based on more advanced geometric texturing algorithms, which may be able to generate textures from sample images or 3D height fields.

On the process side, the contribution of the work includes some build tests of textured models using different AM processes. The tests were planned so as to address the issues expected when texture features are small relative to the part. Beside requiring meshes with a larger number of triangles, very fine textures require high-resolution processes, which are usually slower or more expensive and therefore less suitable for large parts. The tests involved two possible trade-offs between texture detail and part size. Textures with height around 1 mm are faithfully reproduced on models within 200 mm size without excessive increase of the manufacturing cost: this was verified both with the SLS process, which has moderate build speed with a minimal compromise on detail resolution, and with the FFF process, which is slower and rougher but allows the use of highly affordable 3D printers. Finer textures (0.5 mm high) are obtained at comparable costs on parts with size around 50 mm using the Polyjet process, which gives superior detail resolution at the expense of a lower build speed. At present, process constraints seem to be less burdensome than computational issues in determining the application limits of direct texturing. Future studies will extend the tests to a wide range of AM processes in the search of suitable solutions to further extend the allowable range of texture detail and object size.

References

- Andersen, V. *et al.* (2009), "Height and tilt geometric texture". *Proceedings International Symposium on Visual Computing*, pp. 656-677.
- Armillotta, A. (2004), "Procedural texturing of triangle meshes for rapid prototyping", SME Technical Paper TP04PUB124, Society of Manufacturing Engineers.
- Armillotta, A. (2006), "Assessment of surface quality on textured FDM prototypes", *Rapid Prototyping Journal*, Vol. 12 No. 1, pp. 35-41.
- Bhat, P. *et al.* (2004), "Geometric texture synthesis by example", *Proceedings Eurographics Symposium on Geometry Processing*, pp. 41-44.
- Bidasaria, H.B. (1992), "Defining and rendering of textured objects through the use of exponential functions", *CVGIP: Graphical Models and Image Processing*, Vol. 54 No. 2, pp. 97-102.
- Chen, Y. and Ip, H.H.S. (2004), "Texture evolution: 3D texture synthesis from single 2D growable texture pattern", *The Visual Computer*, Vol. 20, pp. 650-664.
- Chen, J. and Wang, B. (2010) "High quality solid texture synthesis using position and index histogram matching", *The Visual Computer*, Vol. 26, pp. 253-262.
- Chiou, J.W. and Yang, C.K. (2006), "A new algorithm for solid texture synthesis", *Lecture Notes in Computer Science*, 4292, pp. 780-789.
- Coblas, D.G. *et al.* (2015), "Manufacturing textured surfaces: state of the art and recent developments", *Proceedings IMechE Part J: J. of Engineering Tribology*, Vol. 229 No. 1, pp. 3-29.
- Cook, R.L. (1984), "Shade trees", *Computer Graphics*, Vol. 18 No. 3, pp. 223-231.
- Curodeau, A. *et al.* (2000), "Design and fabrication of cast orthopedic implants with freeform surface textures from 3-D printed ceramic shell", *Journal of Biomedical Materials Research*, Vol. 53, pp. 525-535.
- Cutler, B. *et al.* (2002), "A procedural approach to authoring solid models" *ACM Transactions on Graphics*, Vol. 21 No. 3, pp. 302-311.
- Dischler, J.M. and Ghazanfarpour, D. (1999), "Interactive image-based modeling of macrostructured textures", *IEEE Computer Graphics and Applications*, Vol. 19 No. 1, pp. 66-74.
- Dischler, J.M. and Ghazanfarpour, D. (2001), "A survey of 3D texturing", *Computer & Graphics*, Vol. 25, pp. 135-151.
- Dong, Y. *et al.* (2008), "Lazy solid texture synthesis", *Proceedings Eurographics Symposium on Rendering*, Vol. 27 No. 4, pp. 1165-1174.
- Du, S.P. *et al.* (2013), "Semiregular solid texturing from 2D image exemplars", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 19 No. 3, pp. 460-469.
- Dumas, J. *et al.* (2015), "By-example synthesis of structurally sound patterns", *ACM Transactions on Graphics*, Vol. 34 No. 4, #137.
- Ebert, D.S., ed. (1998), *Texturing and modeling: a procedural approach*, Academic Press, San Diego, CA.
- Evans, C.J. and Bryan, J.B. (1999), "Structured, textured or engineered surfaces", *Annals of the CIRP*, Vol. 48 No. 2, pp. 541-556.

- Ferguson, R.S. (2014), *Practical algorithms for 3D computer graphics*, 2nd ed., CRC Press, Boca Raton, FL.
- Fleischer, K.W. *et al.* (1995), "Cellular texture generation", *Proceedings ACM SIGGRAPH Conference*, pp. 239-248
- Gardner, G.Y. (1984), "Simulation of natural scenes using textured quadric surfaces", *Computer Graphics*, Vol. 18 No. 3, pp. 11-20.
- Ghazanfarpour, D. and Dischler, J.M. (1995), "Spectral analysis for automatic 3-D texture generation", *Computer & Graphics*, Vol. 19 No. 3, pp. 413-422.
- Gilet, G. and Dischler, J.M. (2009), "A framework for interactive hypertexture modelling", *Computer Graphics Forum*, Vol. 28 No. 8, pp. 2229-2243.
- Haralick, R.M. (1979), "Statistical and structural approaches to texture", *Proceedings IEEE*, Vol. 67 No. 5, pp. 786-804.
- Heeger, D.J. and Bergen, J.R. (1995), "Pyramid-based texture analysis/synthesis", *Computer Graphics*, Vol. 29, pp. 229-238.
- Herholz, P. *et al.* (2017), "Unsharp masking geometry improves 3D prints", *Computer & Graphics*, Vol. 66, pp. 135-142.
- Holovenko, Y. *et al.* (2018), "Friction studies of metal surfaces with various 3D printed patterns tested in dry sliding conditions", *Proceedings IMechE Part J: Journal of Engineering Tribology*, Vol. 232 No. 1, pp. 43-53.
- Hong, Y. *et al.* (2017), "Friction and wear of textured surfaces produced by 3D printing", *Science China: Technological Sciences*, Vol. 60 No. 9, pp. 1400-1406.
- Jafari, R. *et al.* (2019), "Recent progress and challenges with 3D printing of patterned hydrophobic and superhydrophobic surfaces", *International Journal of Advanced Manufacturing Technology*, Vol. 103, pp. 1225-1238.
- Jagnow, R. *et al.* (2004), "Stereological techniques for solid textures", *ACM Transactions on Graphics*, Vol. 23 No. 3, pp. 329-335.
- Jang, H. and Han, J.H. (2013), "GPU-optimized indirect scalar displacement mapping", *Computer-Aided Design*, Vol. 45, pp. 517-522.
- Jee, H. and Sachs, E. (2000), "Surface macro-texture design for rapid prototyping", *Rapid Prototyping Journal*, Vol. 6 No. 1, pp. 50-59.
- Jin, G.Q. *et al.* (2011), "Adaptive tool-path generation of rapid prototyping for complex product models", *Journal of Manufacturing Systems*, Vol. 30, pp. 154-164.
- Jin, G.Q. *et al.* (2013), "A hybrid and adaptive tool-path generation approach of rapid prototyping and manufacturing for biomedical models", *Computers in Industry*, Vol. 64, pp. 336-349.
- Johnson, A.A. *et al.* (2018), "The design and assessment of bio-inspired additive manufactured stab-resistant armour", *Virtual and Physical Prototyping*, Vol. 13 No. 2, pp. 49-57.
- Jung, S. and Lee, C.H. (2015), "Displacement mapping with an augmented patch mesh", *IEICE Transactions on Information and Systems*, E98-D, pp. 741-744.
- Kajiya, J. and Kay, T. (1989), "Rendering fur with three dimensional textures", *Computer Graphics*, Vol. 23 No. 3, pp. 271-298.
- Kanada, Y. (2016), "3D printing of generative art using the assembly and deformation of direction-specified parts", *Rapid Prototyping Journal*, Vol. 22 No. 4, pp. 636-644.
- Ko, K. *et al.* (2019), "Bio-inspired bimaterial composites patterned using three-dimensional printing", *Composites Part B*, Vol. 165, pp. 594-603.
- Kobayashi, Y. and Shirai, K. (2008), "Multi-axis milling from micro-texturing", *International Journal of Precision Engineering and Manufacturing*, Vol. 9 No. 1, pp. 34-38.
- Kopf, J. *et al.* (2007), "Solid texture synthesis from 2D exemplars", *ACM Transactions on Graphics*, Vol. 26 No. 3, #2.
- Lagae, A. *et al.* (2010), "A survey of procedural noise functions", *Computer Graphics Forum*, Vol. 29 No. 8, pp. 2579-2600.
- Lewis, J.P. (1989), "Algorithms for solid noise synthesis", *Computer Graphics*, Vol. 23 No. 3, pp. 263-270.
- Maekawa, K. *et al.* (2003), "Laser fabrication of affective 3D objects with 1/f fluctuation", *JSME International Journal Series A*, Vol. 46 No. 3, pp. 460-466.
- Mai, J. *et al.* (2016), "Towards a new techno-aesthetic paradigm: experiments with pattern, texture and colour in 3D-printed wood-plastic composites", *Proceedings Solid Freeform Fabrication Symposium*, pp. 1868-1877.
- Mantihal, S. *et al.* (2019), "Texture-modified 3D printed dark chocolate: sensory evaluation and consumer perception study", *Journal of Texture Studies*, pp. 1-14.

- Nießner, M. and Loop, C. (2013), "Analytic displacement mapping using hardware tessellation", *ACM Transactions on Graphics*, Vol. 32 No. 3, #26.
- Ou, J. *et al.* (2016), "Cillia: 3D printed micro-pillar structures for surface texture, actuation and sensing", *Proceedings CHI Conference on Human Factors in Computing Systems*, San Jose CA, pp. 5753-5764.
- Peachey, D.R. (1985), "Solid texturing of complex surfaces", *Proceedings ACM SIGGRAPH Conference*, Vol. 19 No. 3, pp. 279-286.
- Pedersen, H.K. (1994), "Displacement mapping using flow fields", *Proceedings ACM SIGGRAPH Conference*, Vol. 28, pp. 279-286.
- Perlin, K. (1985), "An image synthesizer", *Proceedings ACM SIGGRAPH Conference*, Vol. 19 No. 3, pp. 287-296.
- Perlin, K. (1989), "Hypertextures", *Computer Graphics*, Vol. 23 No. 3, pp. 253-262.
- Pietroni, N. *et al.* (2007), "Texturing internal surfaces from a few cross sections", *Computer Graphics Forum*, Vol. 26 No. 3, pp. 637-644.
- Pietroni, N. *et al.* (2010), "Solid-texture synthesis: a survey", *IEEE Computer Graphics and Applications*, Vol. 30 No. 4, pp. 74-89.
- Pintus, R. *et al.* (2010), "Shape enhancement for rapid prototyping", *The Visual Computer*, Vol. 26, pp. 831-840.
- Qin, X. and Yang, Y.H. (2007), "Aura 3D Textures," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13 No. 2, pp. 379-389.
- Sachs, E. *et al.* (1994), "Surface texture by 3D printing", *Proceedings Solid Freeform Fabrication Symposium*, pp. 56-64.
- Schroeder, C.A. *et al.* (2005), "Stochastic microgeometry for displacement mapping", *Proceedings IEEE International Conference on Shape Modeling and Applications*, pp. 164-173.
- Shu, Y. *et al.* (2014), "Efficient texture synthesis of aggregate solid material", *The Visual Computer*, Vol. 30, pp. 877-887.
- Spahiu, T. *et al.* (2017), "Varying fabric drape by 3D-imprinted patterns for garment design", *IOP Conference Series: Materials Science and Engineering*, Vol. 254, 172023.
- Szirmay-Kalos, L. and Umenhoffer, T. (2008), "Displacement mapping on the GPU: state of the art", *Computer Graphics Forum*, Vol. 27 No. 6, pp. 1567-1592.
- Takayama K. *et al.* (2008), "Lapped solid textures: filling a model with anisotropic textures," *ACM Transactions on Graphics*, Vol. 27 No. 3, pp. 1-9.
- Turk, G. (1991), "Generating textures on arbitrary surfaces using reaction-diffusion", *Computer Graphics*, Vol. 25 No. 4, pp. 289-298.
- van Rompay, T.J.L. *et al.* (2017), "See me, feel me: effects of 3D-printed surface patterns on beverage evaluation", *Food Quality and Preference*, Vol. 62, pp. 332-339.
- Weinhaus, F.M. and Devarajan, V. (1997), "Texture mapping 3D models of real-world scenes", *ACM Computing Surveys*, Vol. 29 No. 4, pp. 325-365.
- Wei, L.Y. and Levoy, M. (2001), "Texture synthesis over arbitrary manifold surfaces", *Proceedings ACM SIGGRAPH Conference*, pp. 355-360.
- Wei, L.Y. *et al.* (2009), "State of the art in example-based texture synthesis", *Eurographics 2009 State of the Art Report*, EG-STAR, pp. 93-117.
- Yaman, U. *et al.* (2019), "Generation of patterned indentations for additive manufacturing technologies", *IISE Transactions*, Vol. 51 No. 2, pp. 209-217.
- Zhang, G.X. *et al.* (2011), "Sketch guided solid texturing", *Graphical Models*, Vol. 73, pp. 59-73.
- Zhou, K. *et al.* (2006), "Mesh quilting for geometric texture synthesis", *Proceedings ACM SIGGRAPH Conference*, pp. 690-697.
- Zhou, S. *et al.* (2014), "Topology-constrained synthesis of vector patterns", *ACM Transactions on Graphics*, Vol. 33 No. 6, #215.