Inline monitoring of 3D concrete printing using computer vision

Rodrigo Rill-García, Eva Dokladalova, Petr Dokládal, Jean-François Caron, Romain Mesnil, Pierre Margerit, Malo Charrier

Inline monitoring of 3D concrete printing using computer vision

Rodrigo Rill-García [a,b,*], Eva Dokladalova [a], Petr Dokládal [c], Jean-François Caron [b], Romain Mesnil [b], Pierre Margerit [b,d], Malo Charrier [b]

[a] LIGM, Univ Gustave Eiffel, CNRS, F-77454 Marne-la-Vallée, France
[b] Navier Laboratory, École des Ponts ParisTech, Univ Gustave Eiffel, CNRS, F-77454 Marne-La-Vallée, France
[c] MINES Paris, PSL University, Center for Mathematical Morphology (CMM), F-77300 Fontainebleau, France
[d] Processes and Engineering in Mechanics and Materials laboratory (PIMM) ENSAM, CNRS, CNAM, HESAM Université, 151 Boulevard de l'Hôpital, F-75013 Paris, France

ARTICLE INFO

Keywords:
Automatic monitoring
3D concrete printing
Image processing
Deep learning

ABSTRACT

The detection of anomalies is at the basis of any 3D printing control. In this paper, we propose a methodology for detection of anomalies based on computer vision. This methodology is composed of three modules: (1) image acquisition, (2) interlayer line and layer segmentation and (3) characterization of the local geometry and texture of the layers and detection of anomalies. The image acquisition is performed with a camera fixed to the printing nozzle. The proposed layer segmentation method recognizes and locates the lines separating the printed layers (F-score = 91%). The third module – taking as input the segmentation and the original image – evaluates the geometry of the layers and the texture of the material. The results are used to detect geometry anomalies when the values are outside the expected range. The material texture is classified into four classes of quality (macro-averaged F-score = 94%). We present the results and show the suitability of our methodology for automatic detection and localization of anomalies on images acquired during a printing session.

1. Introduction

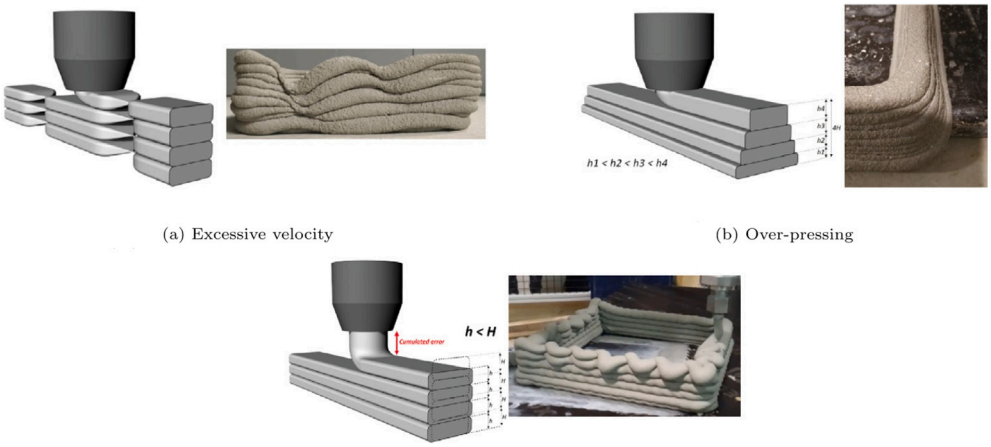
Construction based on additive manufacturing is now heading towards its maturity [1], with practical examples of 3D concrete printing (3DCP) at the building scale [2]. However, despite these high-profile applications, there is still work to be done on material development, construction strategies and the process control to make this technology reproducible and certifiable.

In principle, extrusion-based 3D printing is based on an established workflow: (1) design of the part, (2) definition of the print path and (3) extrusion of the material (printing) [3]. Nevertheless, many different process parameters influence the 3DCP extrusion e.g. the velocity of the extrusion nozzle, the distance between the nozzle and the surface of deposition, the dosing pump pressure, the extruded concrete's rheological properties, etc. For this reason, various monitoring methods have been proposed at different stages: before printing (determining the feasibility of the process), during printing (process feedback), and at post-production stages (post-printing processing) [4].

The primary objective of inline monitoring (during printing) is to allow the operators to carry out corrective actions in case of anomalous behavior. This principle can be extended to automatic control systems, as seen in the literature for manufacturing processes at different size scales. For example, in [5], a feedforward control system is tested for fused filament fabrication. In [6], a closed-loop control system is tested for 3DCP. In both cases, a vision system is used to measure the parameter of interest: the layer width, which can be controlled by adjusting the velocity of the extrusion nozzle.

Nonetheless, the layer width is not the only parameter that can be measured to be able to take corrective action and minimize defects. Some examples of geometrical deformations caused by process defects, that can be identified by inspecting the layers' thickness, are described in [7]:

- *Excessive velocity* appears when the velocity of the extrusion nozzle exceeds the extrusion rate, producing discontinuous layers also called longitudinal tearing (Fig. 1(a))
- *Over-pressing* occurs when the pressing force resulting from the extruded layer exceeds the strength of the penultimate layer, resulting in staggered layer patterns with a loss of control of layer width and thickness (Fig. 1(b))
- *Flow-out* occurs when the yield stress of the material is not sufficient to hold its own weight. This introduces distance between the



(a) Excessive velocity

(b) Over-pressing

(c) Flow-out

Fig. 1. Printing process defects and their influence in the local geometry of printed samples [7].

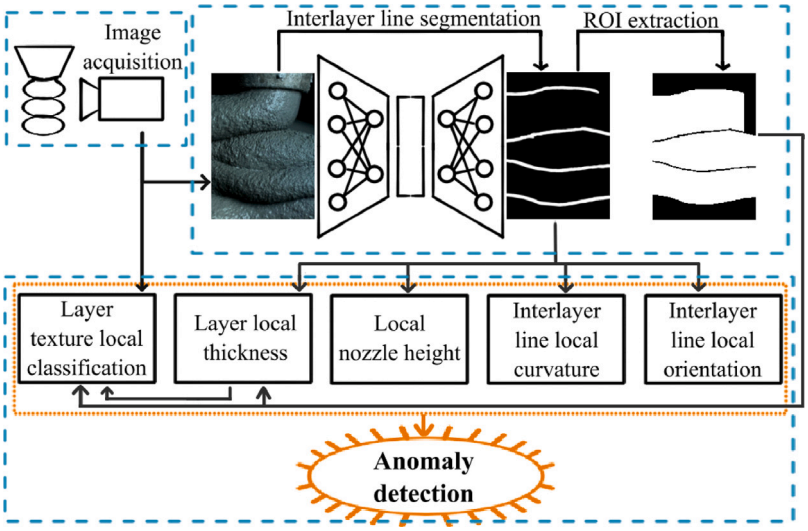


Fig. 2. Summary of the proposed methodology.

printed piece and the extrusion nozzle, producing a poor material deposition (e.g. coiling) (Fig. 1(c))

The measurement of the layers' thickness can be performed by vision systems too. Furthermore, vision systems are not limited to the monitoring of geometrical properties, allowing the analysis of the observed surfaces in terms of properties such as the texture. In this work, using a lateral view of the printed piece, we propose a methodology for inline 3D concrete printing monitoring using computer vision methods (see Fig. 2). Our contributions can be summarized as:

1. A method for automatic layer segmentation. We use a model based on Deep Learning (DL) to locate the interlayer lines in an image (i.e. the lines separating two printed layers), and we segment independent printed layers by using these lines as guide.
2. A monitoring method based on geometrical characterization of the layers. We use image processing to determine the local geometry, based on the interlayer lines, in terms of: orientation, curvature, layer thickness and distance to the nozzle. These measurements are used to detect and locate anomalies with respect to the expected values in the observed layers.
3. A monitoring method based on textural characterization of the layers. We use a machine learning approach to classify the extruded concrete, region-wise, based on textural properties. Textures are classified either as *good* or as one of *fluid*, *dry* and *tearing*. These classes, which are dependent on the water content of the mixture, are highly related to the layer's rheological properties.

The rest of this paper is organized as follows. In Section 2, we make a literature review of approaches used for inline 3DCP monitoring. In Section 3, we describe our experimental setup, focusing on the inline image acquisition. In Section 4, we discuss the properties of the images used in our experiments and the method used for interlayer line segmentation. The methods for layer geometry characterization and texture classification are described in Section 5. In Section 6, we show our results on experimental images acquired in a 3DCP laboratory. In Section 7, we discuss these results and close the paper by proposing future improvements.

2. Related literature

For inline monitoring of 3DCP, a lot of research has been published on rheological properties of the material. For example, an online yield strength measurement on an uni-axial flow [13], where the mass of concrete drops is measured, provides information about the yield

Table 1
Summary of methods based on optics for 3DCP inline monitoring.

Work	Sensor	Type of property	Measurements	Approximate cost[a]
[8]	1D Time-of-Flight	Geometrical	Layer to nozzle height (top view)	5 USD
[9]	Laser triangulation	Geometrical	Layer to nozzle height (top view)	6500 USD
[10]	RGB camera	Geometrical	Layer width (top view)	40 USD
[6]	Monochromatic camera	Geometrical	Layer width (top view)	349 USD
[11]	RGB camera	Geometrical	Layer orientation (lateral view)	40 USD
[12]	3D scanner	Geometrical	Printed piece cloud to 3D model cloud distance (lateral view)	472 USD
Ours	RGB camera	Geometrical, textural	Layer to nozzle height; layer thickness; interlayer line orientation and curvature; texture classification (lateral view)	57 USD

[a]Estimated with base on the sensor models and/or specifications reported in the respective cited works.

strength and heterogeneity of the material before the printing. The material properties can be controlled in real time using approaches such as a two-component strategy — with the addition of accelerator at the nozzle outlet [14]. This type of information is useful because insufficient stiffness or strength of fresh concrete can lead to an unacceptable cumulative error or even failure, as observed in [15]. On the other hand, other parameters may not make the printing process completely fail but would lead to the creation of visible defects leading to the rejection of the 3D printed part afterwards [16].

As a non-invasive approach to inline monitoring, methods based on specialized optical sensors have shown promising results to detect such defects. Most published research in this direction focuses on geometrical features. For example, a 1D Time-of-Flight distance sensor was mounted on the printing nozzle in [8]. The goal was to ensure that the printing nozzle is always at an adequate distance with respect to the last printed layer, so that the extruded material does not get unexpected deformations in its trajectory from the nozzle to the surface of deposition.

This method allows compensating the effect of non-planar print beds, and it forces the robot to extrude material at a proper distance even if lower layers are deformed by the weight of recent layers (flow-out). However, if such a flow-out deformation exist, the final result will be lower than the reference model, with excessively wide layers. This problem can be solved with approaches such as the one proposed by [9]. There, the printing method is shotcrete rather than layer extrusion; for printing monitoring, the authors use laser triangulation. Similarly to [8], the authors measure the distance from the nozzle to the last printed layer. In this work, however, the used printing method allows compensating the height of the printed piece if the bottom layers suffered deformation.

Although this method allows correcting the height of the piece, any anomaly in the width of the layer (over-pressing) is ignored. With respect to this problem, we can find alternatives like the one discussed in [17] based on computer vision.

2.1. Computer vision

3DCP monitoring can be performed with data retrieved from economic devices such as RGB cameras. For example, the work of [10] proposed a monitoring system fully based on computer vision, using a camera fixed to the extrusion nozzle. In that work, the image is first converted to grayscale and blurred with a Gaussian filter; the filtered image is binarized with the Otsu's method to segment the printed layer from the background. Finally, the authors map the width of the segmented layer from pixels to inches (with a top view). The patent number US 8944799 B2 [18] was generated based on this method that adopts computer vision techniques for contour crafting. Low extrusion rates are expected to create thinner layers, while wider layers are produced by high extrusion rates. Another similar work is presented in [6], where the borders of the printed layer are detected as edges. There, the width of the layers is associated with the velocity of the extrusion nozzle, rather than the extrusion rate.

Towards more robust vision systems, DL has gained a significant attention in recent years. For printed concrete monitoring, we can see the example of [11]. There, DL-based segmentation is performed to distinguish a contour-crafting-printed piece from background (similarly to [10]). In that work, the camera acquires images from an external, post-printing, lateral view rather than from an inline top view. This allows simultaneously monitoring the piece's height and the interlayer lines. Once the printed piece is isolated from the background in the captured images, a surface smoothing filter is used along with the Canny edge detector to identify the interlayer lines in the printed piece. Then, the Hough transform is used to estimate, region-wise, the orientation of these lines. Finally, a module for defect detection locates lines with unexpected orientations inside the printed piece.

For 3D geometry assessment, [12] compares the 3D model used for printing with a point cloud obtained (post-printing) by scanning the printed piece on a rotatory base. Using techniques from mathematical morphology, the authors calculate a distribution of cloud-to-cloud distances; this distribution allows quantifying global (all piece) and local (layer-wise) errors with respect to the model used to design the printing path.

As seen in the reviewed literature, optical sensors are useful for non-invasive 3DCP monitoring methods. However, research on these methods is still limited in the literature. Furthermore, research on methods based on computer vision is even more scarce. While geometrical assessment is a popular trend in methods using computer vision, monitoring of the surface has been ignored in 3DCP. In fact, surface analysis for inline monitoring of small-scale extrusion is an active topic of research [19]. Among the methods based on computer vision, the ones based on texture analysis for surface quality assessment can be extended for the monitoring of extruded concrete.

In this work, we propose to detect anomalies in both aspects – the geometry and the texture – with a single camera, during the printing process. Our methodology is compared in Table 1 with the different works based on optical sensors that were discussed in this

section. With the experiments presented in this work, we will show that: (1) an analysis of the interlayer lines allows measuring multiple geometrical parameters of the printed piece to locate anomalies, making the segmentation of these lines an important task; (2) the texture of the extruded layers allows locating abnormal regions when the mixture presents anomalies in the water content; (3) the analysis of the geometry and the texture can be performed from the input of a single RGB camera.

3. Experimental setup

For the results presented in this paper, we created a dataset from images recorded during a printing session run in the 3D printing laboratory of École des Ponts ParisTech. The cell is composed of a 6-axes industrial robot ABB IRB 6620,[1] using the xHEAD printing head developed by XTreeE[2] as extruder. It relies on a bi-component strategy, similar to the one described in [20], with an external feeding pump for the concrete, and a dosing pump for concrete and accelerator in the printing head. A mixing paddle guarantees the homogeneity of dispersion of accelerator in a mixing chamber. The material is the NAG3 concrete developed by Lafarge and tested in [21]. The nominal speed is 80 mm/s, with a nozzle diameter of 20 mm, while the pipe diameter is 30 mm. The typical flow rate is 1 L/min, but may slightly vary during the experiment.

The robot trajectories are constructed from a 3D model and a simple slicing algorithm in Grasshopper [22], which outputs a series of planes (targets). These targets are later converted into RobotStudio files through HAL [23], a general programming language for industrial robotics. Constraints of orientation were added in order to guarantee the perpendicularity between the camera frame and the deposited layer; the inverse kinematics is solved with HAL, which also indicates if a position is not accessible and allows either repositioning the printed piece in a more accessible region or changing its geometry in the 3D modeling environment.

The robot controller controls the position and speed of the robot, the flow rate of the dosing pump and the flow rate of accelerator. The printing parameters used during the session were dynamically varied to purposely create geometrical and textural imperfections (see Fig. 3).

Using this setup, the proposed monitoring methodology consists of 3 sequential modules (see blue dashed rectangles in Fig. 2):

1. Image acquisition
2. Interlayer line and layer segmentation
3. Local layer characterization and anomaly detection

4. Image acquisition and interlayer line segmentation

Several setups for image acquisition are possible. As illustrated in Fig. 4(a), we chose a camera fixed to the nozzle in order to have a fixed point of view with respect to the extrusion zone.

4.1. Image acquisition setup

To avoid blur in the acquired images, caused by regions of the printed piece outside the depth of field of the camera, the camera orientation should remain perpendicular with respect to the printed wall. This normally requires a seven axis in the robot that allows the extrusion nozzle to rotate around its longitudinal axis. This problem of orientation also exists when printing with non-circular nozzles, which are common place in 3DCP; although out of the scope of this paper, there are technical solutions to this issue [24].

(a) (b)

Fig. 3. (a) Experimental sample printed with purposely created defects. (b) Example of acceptable printing without purposely created defects.

Regarding image resolution, it should allow correctly capturing texture i.e. identifying the aggregates in the concrete. In the experiments for this article, the largest aggregate is roughly 0.3 mm. The desired camera resolution is therefore >3.3 px/mm. Due to eventual deformations or lack of control of width, small apertures are preferred in our applications: for a smaller aperture, the depth of field increases, allowing obtaining sharp images even if the distance between the observed concrete and the lens differs from the expected value.

Small aperture is usually used with long exposure times, which implies a trade-off on this parameter, because mitigation of motion blur requires short exposure times. Here, we are mostly in the case of linear motion blur. If we want the blur to be limited to 1 pixel, we get a constraint on exposure time. Using experimental data, we get an upper bound of $t_{exposure} \sim 1/3000$ s, which holds for a wide range of printed concretes, where typical aggregate size remains submillimetric and robot speed in the 100 mm/s range.

4.1.1. Properties of the acquired images

The distance of the lens was adjusted to contain approximately 4 layers per frame, with an expected layer thickness of 6 mm. To acquire images, a Raspberry Pi Camera *High Quality* Module featuring a SONY IMX477 CMOS sensor with a macro lens was fixed on the printing nozzle (more camera specifications available at [25]). This allowed having a fixed distance (~13 cm) between the lens and the extruded material. Additionally, we mounted a led lamp on the nozzle to allow a short, and constant, exposure time; this is illustrated in Fig. 4(a). The image resolution is 1280 × 960 pixels; therefore, our images contain ≈40 px/mm. The focus, exposure time and lens aperture were manually adjusted prior to printing the piece and fixed during extrusion.

To have uniformity in our final dataset, from the acquired images we chose a subset of pictures meeting 2 conditions: (1) no background is visible at the sides of the printed piece, (2) the center of the nozzle is coplanar with the center of the observed wall (as illustrated in Fig. 4(a)). We preserved 628 raw images in total; a schematic diagram of a typical frame with acceptable printing is shown in Fig. 4(b).

4.2. Interlayer line and layer segmentation

Once an image is acquired, the next step in our methodology is to segment its interlayer lines. On dry pieces, the detection of these

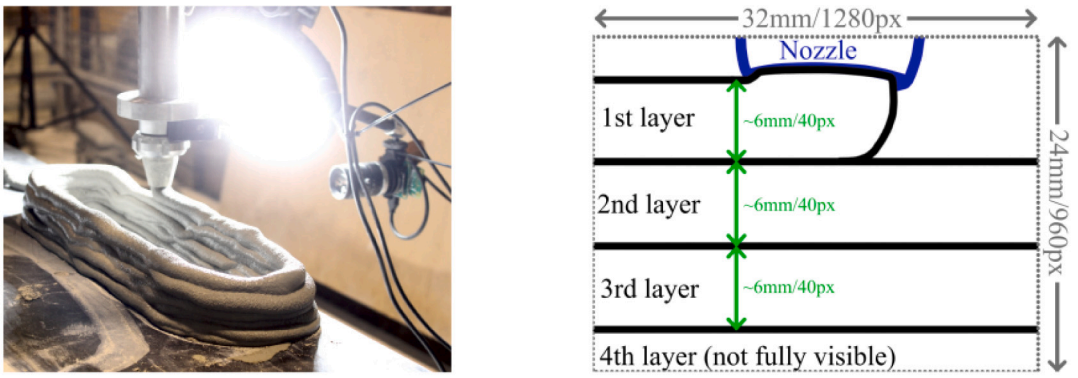


Fig. 4. The printing process with inline image acquisition.

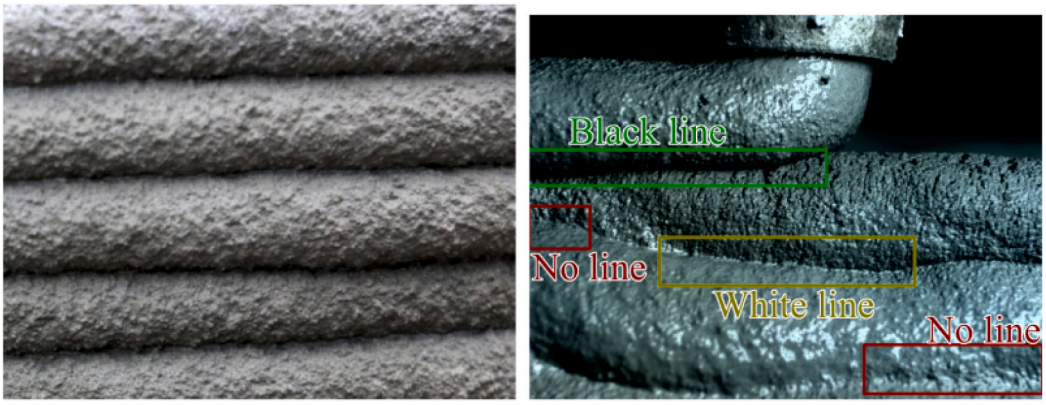


Fig. 5. In contrast with dry concrete, the interlayer lines in fresh concrete can appear black, white, or invisible, because of layer superposition and specular reflectance caused by the material's humidity.

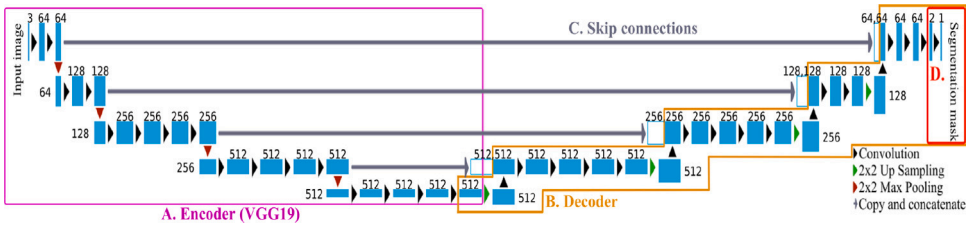


Fig. 6. Architecture of U-VGG19 as provided in [26].

lines can be based on finding dark edges (see Fig. 5(a)). However, fresh concrete exhibits specular reflectance as well as layer merging and superposition. Consequently, interlayer lines can take different aspects: black, bright or even not visible at all (see Fig. 5(b)). Furthermore, given the constraints imposed by inline image acquisition (as described in Section 4.1), the problem becomes particularly hard: loss of focus, unexpected motion blur, camera vibration, polluted air (presence of particles), etc.

To approach this problem, we trained a neural network originally used for crack segmentation: U-VGG19 [26]. It is a fully convolutional neural network inspired by U-net which uses VGG19 as backbone model; the architecture of this encoder–decoder network is illustrated in Fig. 6. By using this network, we get as output a binary mask with the location of the interlayer lines in the input image (as illustrated in Fig. 7).

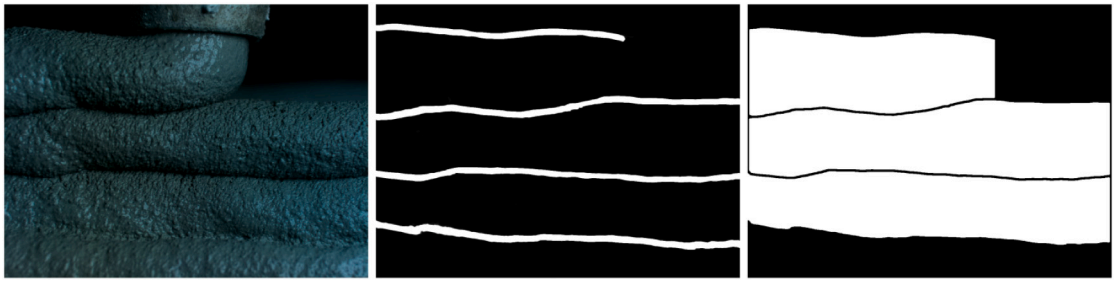
Given an input image I, let us denote by $S \subset \mathbb{Z}^2$ the interlayer lines segmented by the model. This segmentation S can be used to extract the layers delimited vertically between each two lines. Let us denote by $ROI \subset \mathbb{Z}^2$ this set of layers (see Fig. 7(c)).

5. Inline 3D concrete printing monitoring

Once the interlayer lines and the independent layers are segmented, the next step in our methodology is to perform a local layer characterization for anomaly detection.

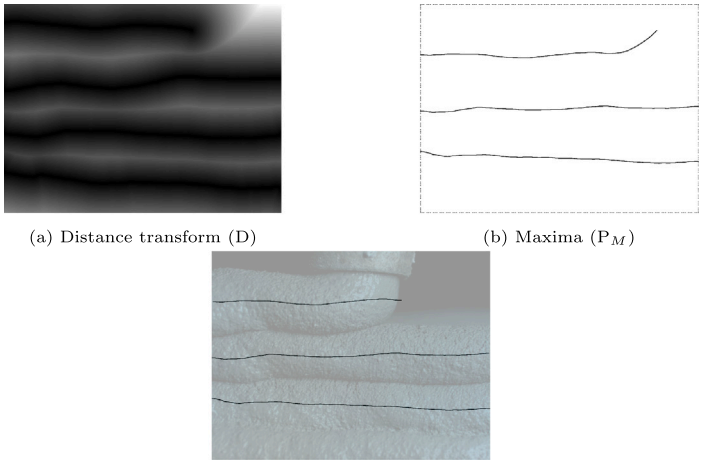
5.1. Geometrical characterization

The proposed geometrical characterization consists of local measures of the geometry of the printed layers and interlayer lines. These measures, and the posterior anomaly detection, are performed using the following methods.



(a) Image from the dataset (I) (b) Segmented interlayer lines (S) (c) Regions of interest (ROI)

Fig. 7. (a) One image from the collected dataset; the field of view was adjusted to contain mostly concrete regions including the last printed layer. (b) Interlayer lines segmented by U-VGG19. (c) Regions of interest in white; regions within a layer that are not contained between two interlayer lines are ignored because the layer is not seen entirely in the vertical axis.



(a) Distance transform (D) (b) Maxima (P_M)

(c) Layer centers

Fig. 8. Method proposed to measure the local thickness of the layers. (a) Distance transform of S_T. (b) Maxima from the distance transform in the vertical direction. (c) Local layer centers.

5.1.1. Local orientation of interlayer line

As discussed by [11], studying the orientation of the interlayer lines allows detecting deformations in the printed layers. To do this, we propose a method based on image filtering using rotating structuring elements [27].

First, we perform a thinning [28] to approximate the interlayer lines to 1-pixel-width lines. Let $S_T \subset \mathbb{Z}^2$ denote the thinned image, $R(d, l)$ a line oriented by d with length l, and D the set of possible orientations $d \in [-90, 90)$. The line length l is chosen to be 1/5 of the image width. This hyperparameter presents a trade off between the available line angles [27] and the locality of the measurement. The value (256 pixels) was selected to be similar to the expected layer thickness, but it is not critical for the method. Let $p \in \mathbb{Z}^2$ be a point in an image. Then, $F(p, d) = [S_T * R(d, l)](p)$ is the convolution of S_T with a line oriented in d. Finally, our local orientation map ϕ is defined as:

$$\phi(p) = \begin{cases} \arg\max_{d \in D} \mathrm{F}(p, d), & p \in S_T \\ \text{undefined}, & \text{otherwise} \end{cases} \tag{1}$$

5.1.2. Local curvature of interlayer line

Let $\hat{L}_n$ be a piece-wise approximation of the nth interlayer line in S_T, using m cubic splines [29]:

$$\hat{L}_n(t) = \begin{cases} s_n^1(t - t_0), & t_0 \le t < t_1 \\ \quad\vdots & \\ s_n^m(t - t_{m-1}), & t_{m-1} \le t < t_m \end{cases} \tag{2}$$

The knots are selected to split the interlayer line of interest into m fixed-length segments. This length is a hyperparameter, and it implies a trade-off between the accuracy and the smoothness of the spline approximation. The value was chosen to be 1/10 of the image width (128 pixels), but it is not critical for the method. The independent variable t corresponds to a parametric representation of the spline $s_n^i(t) = (x(t), y(t))$.

Since this smooth approximation is twice differentiable, the local curvature κ of $\hat{L}_n$ at point t can be computed as:

$$\kappa(t) = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}} \tag{3}$$

The primes refer to the derivatives with respect to t. The magnitude of $\kappa(t)$ is the inverse of the radius of the osculating circle touching the point (x, y) defined by $\hat{L}_n(t)$. The sign indicates the concavity.

5.1.3. Local thickness of layer

The local thickness of a layer is determined by the distance between its two interlayer lines. We calculate this thickness using the fast Euclidean Distance Transform (EDT) [30].

Recall, S_T is the image containing the 1-pixel-wide representation of the interlayer lines. The function $D : \mathbb{Z}^2 \to \mathbb{R}^+$ is the result of applying the EDT to S_T (see Fig. 8(a)). The local maxima of D indicate the centers of the printed layers (see Fig. 8(b–c)). Since we expect the layers to be nearly horizontal in the input picture, we estimate the locations of these local maxima along the vertical direction. Let $k_1 = (0, -1, 1)^T$ and $k_2 = (1, -1, 0)^T$ be vertical kernels. The convolution of the distance map D with these kernels allows detecting the maxima of D:

$$P_M = \{p \mid [\mathrm{sign}(D * k_1) * k_2](p) = 2\} \tag{4}$$

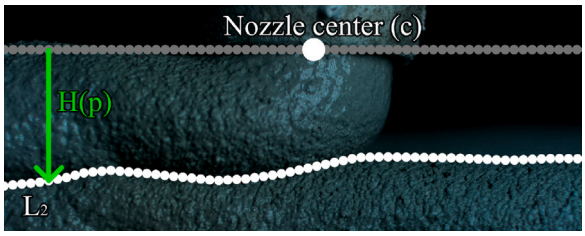


Fig. 9. Relative nozzle height measurement.

For the points located in the middle of two interlayer lines (the layer center), and inside ROI (see Fig. 7(c)), the layer thickness W is

$$W(p) = \begin{cases} 2D(p), & p \in (P_M \cap ROI) \\ \text{undefined}, & \text{otherwise} \end{cases} \quad (5)$$

The factor 2 allows obtaining the thickness (the double of the shortest distance from S_T to the centerline).

5.1.4. Relative nozzle height

Similarly to [8,9], we measure the relative height of the printing nozzle with respect to the surface of deposition. From the lateral point of view proposed in this paper, this height corresponds to the vertical distance from the nozzle to the line receiving the currently printed layer.

Since the camera is fixed to the robot, the center of the nozzle is at a constant position $c = (x_c, y_c)$ in the image. Let L_2, the second interlayer line from top to bottom in the image, be the set of points $p = (x_p, y_p)$ separating the currently printed layer from the layer of deposition (see Fig. 9). Then, the relative nozzle height is:

$$H(p) = \begin{cases} y_p - y_c, & p \in L_2 \\ \text{undefined}, & \text{otherwise} \end{cases} \quad (6)$$

With the vertical axis pointing downwards in an image, this height is positive as long as p is below the nozzle.

5.1.5. Geometrical anomalies

To summarize the information retrieved by Eqs. (1), (3), (5) and (6) (orientation, curvature, thickness and height, respectively), we obtain a histogram per measure. Under good printing conditions, the distributions generated from these histograms should be centered and concentrated inside a range of admissible values.

The principle is illustrated in Fig. 10. There, we show the interlayer segmentation maps from a frame with acceptable printing and one exhibiting signs of over-pressing (see Fig. 1 for reference). Additionally, we show the distributions of layer thickness measured pixel-wise by our method for each frame. The hypothesis is that a defect can be modeled as a deviation (Fig. 10(d)) from the distribution expected for a defect-less printing (Fig. 10(c)). Then it becomes straightforward to identify the position of these deviations and localize the corresponding portion of layer in the frame. Since the fixed pose of the camera with respect to the nozzle is known, and the pose of the nozzle at the time of acquiring the analyzed frame can be retrieved from the robot's control software, it is possible to calculate the physical localization of the anomalies in the printed piece.

In the next section, we present the characterization of the texture of the printed layers.

5.2. Texture characterization

The rheological properties of the extruded material are directly related to the composition of the mixture — which impacts the structural properties of the printed piece. In this work, we focus on the water content: lack of water can cause cold joints at the interface, but an excess of water makes the layers more prone to deformation or even collapse. The water content is one of the several factors affecting the texture of the extruded layers (see Fig. 11). When the texture exhibits abnormal properties, it becomes a good visual indicator of anomalies in the printing process.

In our method, the texture of the printed piece is analyzed locally, layer by layer, in small windows as shown in Fig. 12. The height of the windows is adapted as to cover the entire thickness of either layer, excluding adjacent layers. The windows have a fixed width of 200 pixels (~5 mm) and are adjacent each to another. The value of this hyperparameter (chosen to have near-square windows) presents a trade off between the available area to analyze per window and the locality of the analysis. Each window is then independently classified, either as good or one anomalous class.

5.2.1. Pre-processing

Before extracting the windows, we level the image's grayscale intensities to preserve textural information while reducing the effects of lightning and shadows. Given the input image I, first we obtain a local-mean-intensity image using a Gaussian filter with $\sigma = 40$. By subtracting the filtered image from I, we obtain a new image with an intensity distribution centered around 0. We further subtract the minimum to avoid negative values. The result of this whole process is illustrated in Fig. 13.

5.2.2. Texture features

Each window is analyzed individually to provide a label using a machine learning approach. Our feature extraction is based on two visual descriptors: gray-level co-occurrence matrices (GLCMs) and local binary patterns (LBP). These descriptors are obtained from the maximum rectangular box contained within concrete pixels in the window. Let each of these boxes be called a T_{box}.

Before feature extraction, each T_{box} is quantized to q discrete values. To avoid that outlier intensity values have an undesired influence, each T_{box} is clipped to values in the range $mean(T_{box}) \pm 3.1 \cdot std(T_{box})$ before quantization. Let the quantized version of the image, in the range [0, q-1], be called a T^q_{box}.

GLCMs. Lets define a GLCM [31] as:

$$GLCM^I_{\Delta x,\Delta y}(i,j) = \sum_{x=1}^{w}\sum_{y=1}^{h} \begin{cases} 1, & \text{if I(x, y)=i and} \\ & \text{I(x + } \Delta\text{x, y + } \Delta\text{y)=j} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Here, h and w are the height and the width of an analyzed image I, respectively; Δx and Δy are a horizontal and a vertical offset, respectively. $I(x, y)$ returns the intensity value of the pixel at the position x, y in I.

GLCMs are distributions of co-occurring pixel values at a given scale (the chosen offset). From preliminary experiments, $q = 12$ was chosen for GLCM calculation. With 12 intensity levels in a T^q_{box}, per each offset we obtain a square 12×12 matrix (see Fig. 14). Per each GLCM, we obtain scalar features in terms of statistical properties. With $P_{i,j}$ the value of the normalized GLCM at the position i, j (i.e. the probability of the intensity pair (i, j)), the features are obtained based on 5 properties:

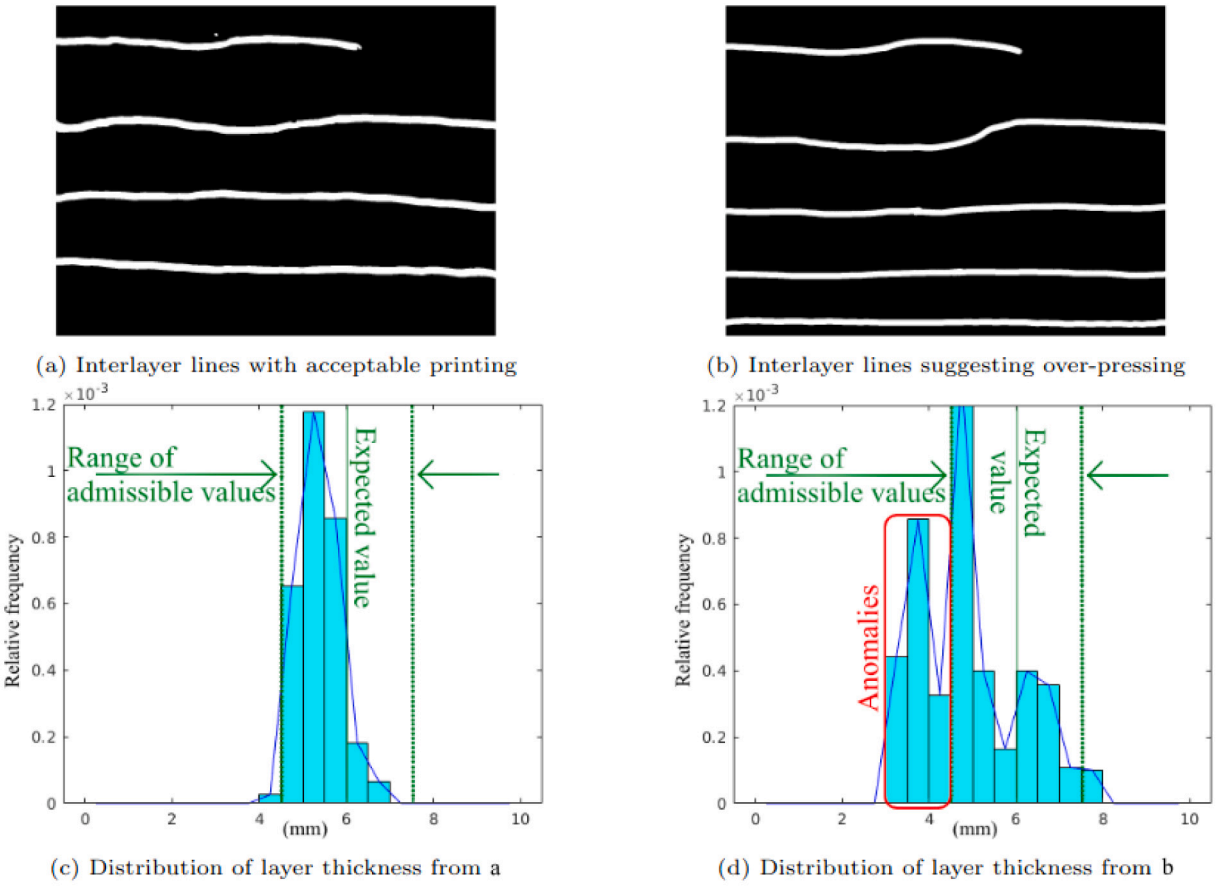


Fig. 10. Distributions of layer thickness, measured pixel-wise. (c) In a defect-free process, the distribution of any measured parameter is contained in an expected interval. (d) When printing defects occur, a deviation from this interval is observed.

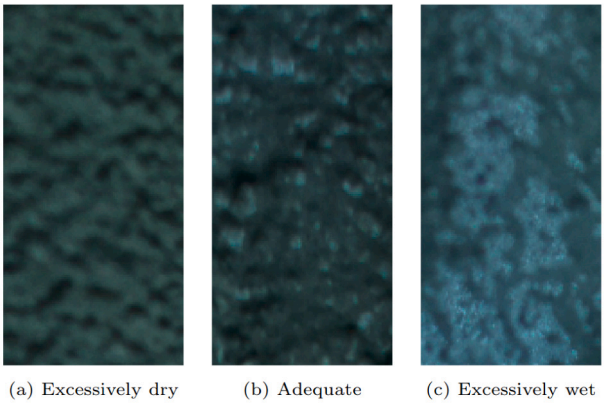


Fig. 11. Texture of concrete with different water contents.

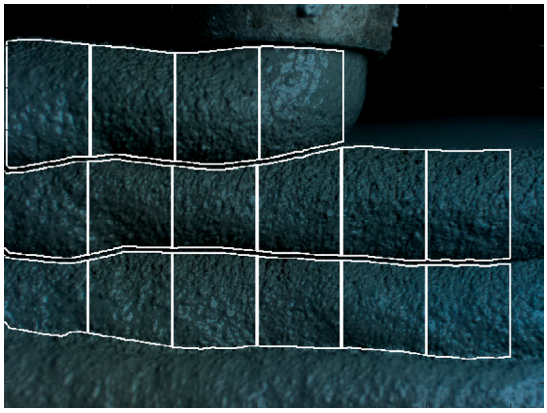


Fig. 12. Windows used to analyze the textures of Fig. 7(a). The width is fixed (200 px ≈ 5 mm) and the height is adapted to contain the entire thickness of the layer.

Contrast: the expected squared difference of intensities in the GLCM. It is defined as:

$$\sum_{i,j=0}^{levels-1} P_{i,j} \cdot (i-j)^2 \tag{8}$$

Dissimilarity: the expected absolute difference of intensities in the GLCM. It is defined as:

$$\sum_{i,j=0}^{levels-1} P_{i,j} \cdot |i-j| \tag{9}$$

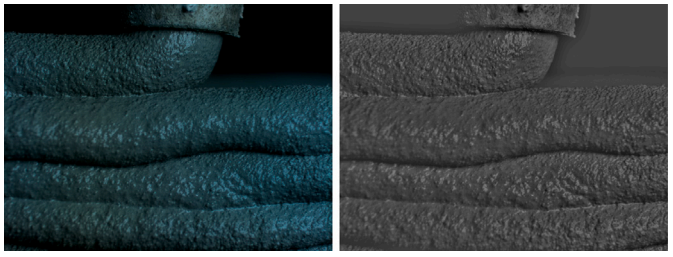


Fig. 13. Raw input (left) and leveled image (right) for texture analysis.

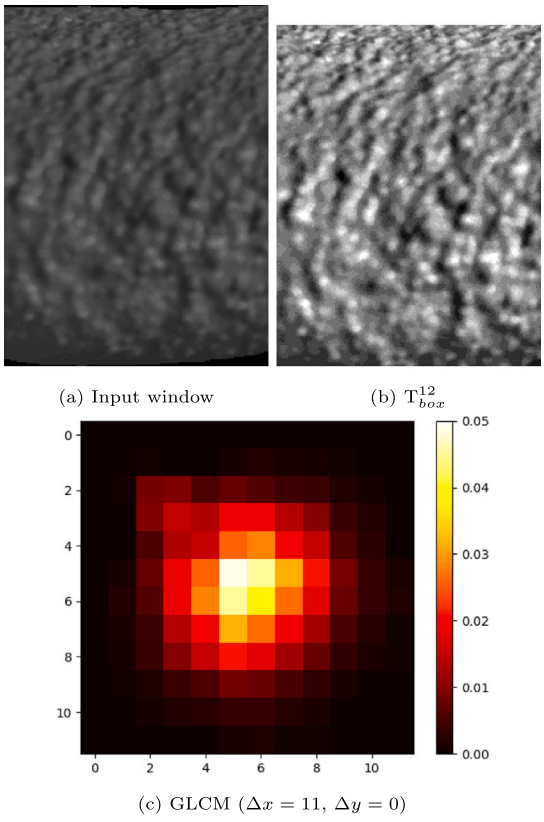


Fig. 14. Example of extracted GLCM.

Homogeneity: a measure of the closeness of the distribution of elements in the GLCM to its diagonal. It is defined as:

$$\sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2} \cdot |i-j| \tag{10}$$

Energy: the square root of the sum of squared elements in the GLCM. It is defined as:

$$\sqrt{\sum_{i,j=0}^{levels-1} P_{i,j}^2} \tag{11}$$

Correlation: a measure of how correlated are the pixels to their neighbors with the given offset. It is defined as:

$$\sum_{i,j=0}^{levels-1} P_{i,j} \cdot \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^2 \cdot \sigma_j^2}} \tag{12}$$

Due to the printing process, the texture properties are anisotropic. Because of this, we extract GLCMs with offsets in the horizontal and vertical directions. Per each direction, we use offset distances from 1 to 50 with step 1. The total number of features obtained per T^q_{box} is 500 (2 directions × 50 distances × 5 statistical properties).

LBP. This descriptor consists of assigning a n-bit binary number to each pixel depending on its n neighbors with radius R [32]. Following the neighbors along the hypothetical circle of radius R, each bit is assigned a 0 if the center pixel's intensity value is greater than the corresponding neighbor's value. Otherwise, the bit is assigned a 1. The transformed image contains, per pixel, the decimal equivalent of the binary number calculated according to their neighbors. We use 8 neighbors, therefore the transformed images contain values in the range [0, 255].

From preliminary experiments, $q = 64$ was chosen for quantization to extract LBP (see Fig. 15). After getting the LBP transform of a T^q_{box}, we calculate the histogram of intensities of the transformed image to extract a fixed-size vector per T^q_{box}. To reduce the dimensionality of this vector, the histograms are calculated using 8 bins. These histograms are normalized so that their magnitudes are not influenced by the size of the T^q_{box}.

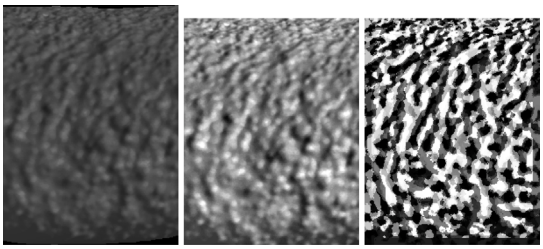


Fig. 15. Example of extracted LBP.

Similarly to GLCMs, we use different radii to extract information at different scales. These radii are from 1 to 31 with step 10. The total number of features per T^q_{box} is 32 (4 radii × 8 bins).

5.2.3. Classification

With the extracted textural features, we classify each T^q_{box} into 1 of 4 classes as illustrated in Fig. 16: excessively fluid, good quality, excessively dry, and tearing.

The *fluid* class is characterized by a smooth surface. Since the fluidness is associated with high water content, this class is also characterized by a high specular reflectance. The *good* class corresponds to the desired material properties. The amount of present water is adequate, producing the appearance of more visible grains at the surface with higher homogeneity and a lower reflectance. The *dry* class is characterized by a rougher texture, caused by an increased amount of visible grains with bigger size. Since the amount of water is reduced, the material exhibits very low to no reflectance. However, this class is also determined by the absence of cracks. The *tearing* class is the one characterized by the appearance of cracks. It is also identified by the lack of reflectance and often by a rougher texture than the dry class.

From the images acquired during the printing process, we extracted and labeled a total of 111 texture windows. Each of these windows is labeled by the agreement of two annotators. The distribution of classes is: 24 fluid, 27 good, 24 dry and 36 tearing. The base dataset is composed by the 532 features calculated per each of these images (GLCM + LBP features), and their corresponding class labels. To avoid that some features dominate the classifier because of their magnitude, we standardize each one of them during training.

The model used to learn from this domain is a small convolutional neural network (CNN) with one convolutional input layer and one fully-connected output layer (see Fig. 17). Because of the feature extraction implementation, adjacent elements in the feature vector represent the same property but with different offsets (see Fig. 18). Since these features have a high likelihood of being correlated, we use the convolution to capture meaningful information from the neighborhood. The kernel size is 6, with a ReLu activation to obtain non-linear outputs. To reduce the likelihood of overfitting, we use dropout regularization with 30% probability. The output layer has 4 neurons and uses a softmax activation so that $\sum_{c \in classes} P(c|\text{features}) = 1$.

5.2.4. Textural anomalies

Similarly to the approach used for geometrical anomaly detection, we can summarize an image in terms of the histogram of window labels. In this case, the categorical distribution should be centered and concentrated in the good class. High occurrence of other classes indicates some anomaly in the printing process.

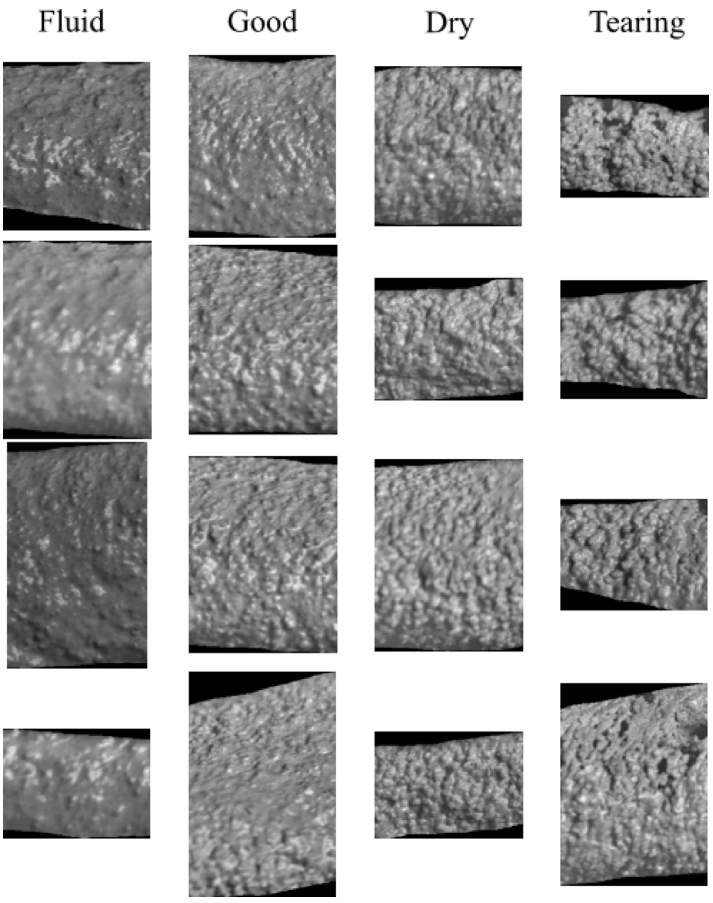


Fig. 16. Examples of extracted windows for the four proposed classes (with improved contrast for visualization).

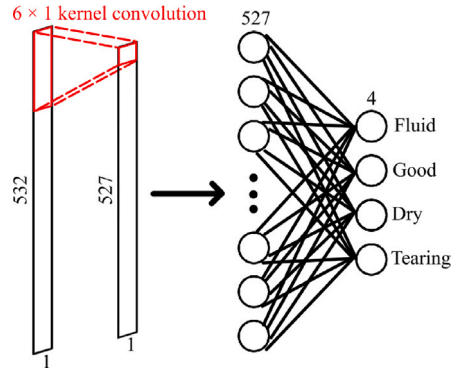


Fig. 17. CNN for texture classification.

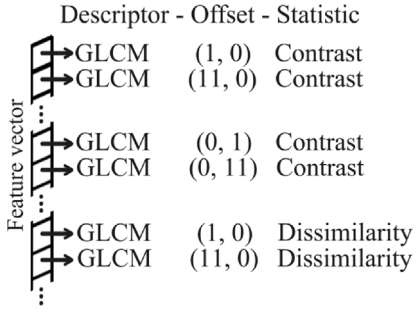


Fig. 18. Feature vector for texture classification. Adjacent elements represent the same property but with different offsets.

6. Experiment results

6.1. Segmentation

Once an image from the printing process is acquired, the next step is to segment the interlayer lines to obtain the binary segmentation maps.

We used 128 manually annotated images for training and 32 for testing. The labeling was done in Krita 4.4.5 using a circular brush with a diameter of 20 pixels (see Fig. 19 for an example of manual segmentation). During training, we provided 256 × 256 image crops with a batch size of 4. We implemented U-VGG19 using Tensorflow 2.1.0. Similarly to [26], we initialized the weights of the encoder with those of the convolutional layers of VGG19 (pre-trained on ImageNet). We used the same optimizer and loss function i.e. Adam and a weighted combination of binary cross-entropy loss and dice score loss. The model is trained during 200 epochs, with an early stop if the testing loss did not improve for 20 consecutive epochs. The learning rate was decreased by a factor of 2 on testing loss plateau with 5 epochs tolerance; the initial learning rate was 10^{-4}.

We evaluate the performance of U-VGG19 on the test images in terms of precision, recall and F-score. We calculate each score per image and we report the average over all the test images. There is a possibility of offsets between prediction and annotation caused by the inaccuracy of the annotators at the moment of locating the center of the interlayer lines. To address this, we use a tolerance margin similarly to works such as [33,34]. The chosen tolerance is 2 pixels, meaning that predicted positive pixels no more than 2 pixels away from a positive pixel in the manual annotation are considered true positives. This represents approximately 0.5 mm in real life. The scores are shown in Table 2.

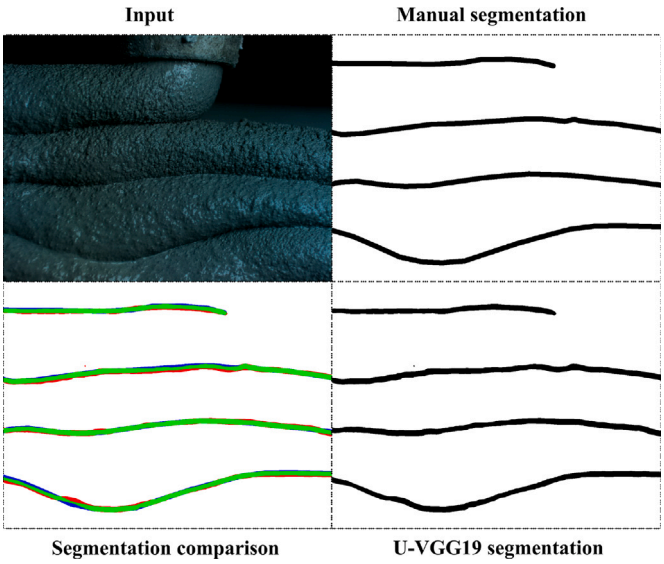
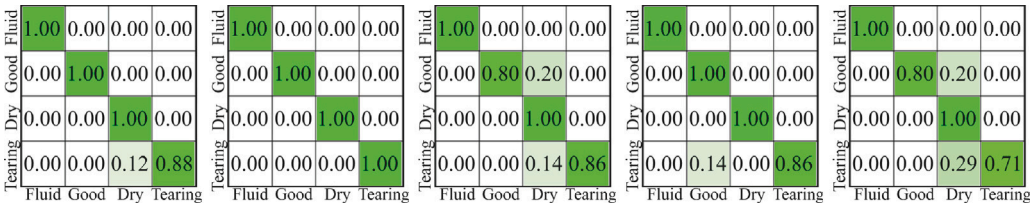


Fig. 19. Comparison between manual and predicted segmentation. The color code is: (Green) True positives, (Red) False positives, (Blue) False negatives. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Fold 1 (b) Fold 2 (c) Fold 3 (d) Fold 4 (e) Fold 5

Fig. 20. Confusion matrices of the 5 testing folds for texture classification. The rows, corresponding to the true labels, are normalized.

Table 2
Interlayer line segmentation testing scores.

F-score (%)	Precision (%)	Recall (%)
90.63	92.14	89.22

The average precision and the average recall have very similar values (around 90%). Therefore, U-VGG19 exhibits a balanced ability to detect interlayer lines while discriminating false positives. An example of automatic interlayer line segmentation using U-VGG19 is illustrated in Fig. 19. With an average F-score of 90.63%, we use the predictions of U-VGG19 to perform the characterization methods proposed in this paper.

6.2. Texture classification

In this multi-class context, we evaluate the proposed method with 5-folds stratified cross-validation i.e. each fold has the same proportion of observations with a given class. To further reduce the likelihood of overfitting, per iteration we perform another stratified split to divide the training folds into a training and a validation split; the size of the validation split is half the size of the test split. Finally, we perform data augmentation on the images from the resulting training split, obtaining 16 additional images per texture window. The transformations are rotations (with angles −4, −3, −2, 2, 3, 4 degrees), zooms (with scale factors 0.97, 0.98, 0.99, 1.01, 1.02, 1.03) and illumination rescales and shifts defined by $\alpha \cdot \text{intensity} + \beta$ (with pairs α/β: 0.9/0, 1.1/0, 1.0/−50, 1.0/50).

Table 3
F-scores (%) of texture classification using 5-folds stratified cross-validation.

Class	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Class macro-average
Fluid	100	100	100	100	100	100
Good	100	100	88.9	90.9	88.9	93.7
Dry	90.9	100	83.3	100	76.9	90.2
Tearing	93.3	100	92.3	92.3	83.3	92.2
Class macro-average	96.1	100	91.1	95.8	87.3	

Per iteration, the approximate number of samples per split is 1326 training, 11 validation and 22 test. Validation and testing are performed on real images only, while the training split does not contain modified versions of the images in the validation and test splits. The approximate class distribution is 22% fluid, 24% good, 22% dry and 32% tearing. We train the CNN with a binary cross-entropy loss using the Adam optimizer with default parameters in Tensorflow 2.1.0. We train for a maximum of 2000 epochs with batch size 32, with an early stop if the loss in the validation split did not improve for 100 consecutive epochs.

We evaluate the trained models in terms of the macro-averaged F-score in the test splits; the macro F-score gives equal importance to all the classes. Table 3 shows the macro F-score per fold and per class; the single F-scores per fold and class are shown too.

Among the four classes, fluid exhibited the best scores (100% in all folds). On the other side, the most difficult class was dry, with a 90.2% average over the 5 folds. The good and tearing classes have slightly better averages: 93.7 and 92.2%, respectively. From an analysis of the resulting confusion matrices per fold (see Fig. 20), we discovered that the main source of these errors was classifying good and tearing images

as dry. Nonetheless, this kind of confusion is understandable since the dry class is, physically, a transitory state between good printing and tearing. Furthermore, these errors are infrequent, as denoted by the average 94.1% macro-averaged F-score over the 5 folds (average of the last row).

With good models for interlayer line segmentation and texture classification, in the next section we show the results of using these models along with the proposed methods for characterization and anomaly detection.

6.3. Layer characterization and anomaly detection

In this section, we show two study cases to characterize, based on the proposed measurements: interlayer line orientation and curvature; nozzle height; and layer thickness and texture. Besides the measurement of orientation, expressed in degrees, all the other geometrical measurements are based on pixels as metric unit. Pixels are converted to millimeters by taking 40 px/mm in our setup (Section 4.1.1). The locally measured values are plotted using color maps (see Fig. 21). Additionally, we show the distributions resulting from these measurements (see Fig. 22). The ranges of admissible values, delimited with green dotted lines in the distributions, are user-defined; deviations outside those ranges are considered anomalies, and they are plotted in blue if below the minimum threshold and in red if above the maximum one.

For orientation, we expect near-to-horizontal layers; the user-defined range is (−10, 10) degrees. For curvature, we expect near-to-straight layers; the range is (−0.05, 0.05) mm^{-1}. The expected layer thickness is 6 mm; the range is (4.5, 7.5) mm. The nozzle should keep a height close to the expected layer thickness; the range is (5, 7) mm. Regarding texture, any class but good is considered as anomaly; the texture classes are associated to a color code similar to the color maps used for geometrical measurements: fluid/blue, good/green, dry/yellow, tearing/red. Additionally to the class, we show the probability of the prediction according to the classifier.

The first study case, analyzed in Figs. 21 and 22, portrays an overall acceptable printing. As observed in Fig. 21(b–e), there are almost no geometrical anomalies (except a few high curvature segments); consequently, the distributions obtained from these plots are contained inside the ranges of admissible values, as seen in Fig. 22(b–e).

The second study case, analyzed in Figs. 23 and 24, depicts a scenario with severe anomalies. As observed in Fig. 24(b–e), the distributions of the geometrical measurements are very different from the ones in Fig. 22: in all the distributions, there are high density regions outside the range of admissible values. The most extreme case is present in Fig. 24(e), where the distribution is concentrated outside the admissible range. This behavior is likely to cause a coiling effect on the material deposition; in fact, as depicted in Figs. 23(c) and 24(c), we see high curvatures that can be directly related to this phenomenon.

Furthermore, with respect to the texture analysis, we see that the second layer from top to bottom exhibits many regions with tearing. This is the extreme case of texture anomalies, since it implies the appearance of cracks that can jeopardize the structural safety of the printed piece.

As shown by this study case, as well as the first one with overall acceptable printing, the proposed methodology is able to provide an inline characterization of the process based on visual inspection of the last printed layers. With this characterization, the proposed methodology detects and locates anomalies in the extruded layers.

7. Conclusions and future work

In the present article, we proposed a methodology for computer-vision based, inline monitoring of 3D concrete printing. This methodology consists of three sequential modules: (1) image acquisition, (2) interlayer line and layer segmentation, and (3) local layer characterization and anomaly detection.

With the presented experiments using the aforementioned methodology, we show that:

- The use of Deep Learning allows an inline segmentation of the interlayer lines separating adjacent extruded layers. The proposed model obtains an F-score of 91%. We demonstrate that locating these lines in the analyzed images allows a further segmentation of the independent layers.
- An analysis of the segmented interlayer lines allows measuring multiple geometrical properties of the printed piece simultaneously. We demonstrate that the measurement of these properties allows detecting and locating anomalies in the printed piece.
- An analysis of the texture of the independent layers allows detecting when the extruded material exhibits anomalies in the water content. We suggest 4 classes for this analysis, and the model proposed to classify the textures obtains a macro-averaged F-score of 94%.

Our experiments show that the geometrical and the textural analyses can be performed simultaneously from the input of a single RGB camera. With this, we extend the catalogue of possible geometrical measurements using computer vision with respect to the current proposals in the literature. Additionally, we are the first to propose an analysis based on texture for 3D concrete printing, inspired by the approaches already used in small-scale, extruded-based additive manufacturing.

In the two presented study cases, we show that our methodology provides the operator with the exact position and nature of the detected anomalies. If the results are recorded, they can serve as a quality report after the printing process is finished. This report can serve as a proof of acceptability of the printed piece. Inversely, it can also serve, when severe anomalies are detected to reject the piece — and even before the end the printing to avoid material and time waste. Finally, the nature and severity of the detected anomalies can be reported to the operator to make corrective adjustments.

A future research could extend this monitoring towards a closed-loop control. The nature and the severity of the anomalies could be used as feedback to an automated decision system issuing a corrective action on the printing system. It could stem from a rule-based expert system, vector-to-action dictionaries, supervised machine learning models, etc.

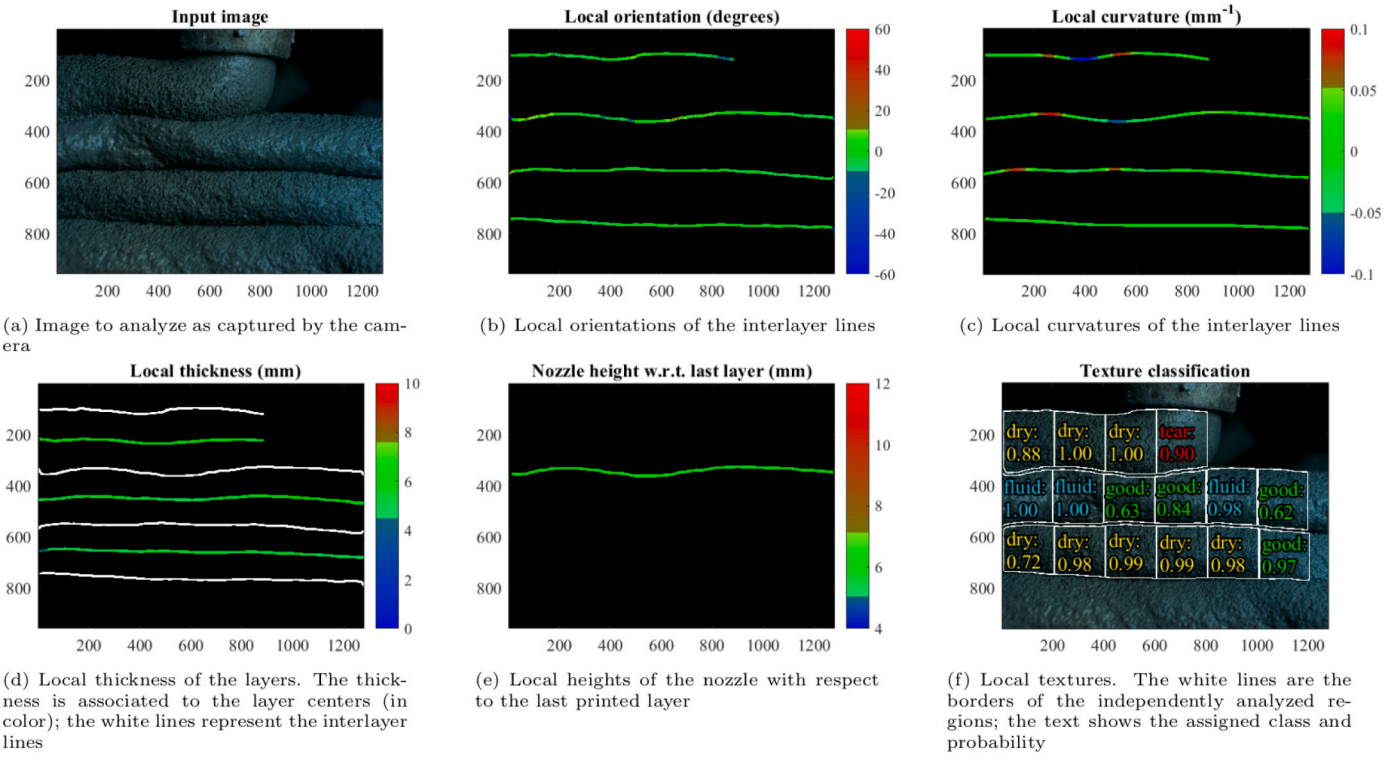
The images and annotations used for the results presented in this paper are available as a public dataset named I3DCP: https://github.com/Sutadasuto/I3DCP. The code to reproduce our results is available at https://github.com/Sutadasuto/3dcp_cv_monitoring.

CRediT authorship contribution statement

Rodrigo Rill-García: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Eva Dokladalova:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Petr Dokládal:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Jean-François Caron:** Funding acquisition, Resources, Supervision. **Romain Mesnil:** Conceptualization, Methodology, Resources, Writing – review & editing. **Pierre Margerit:** Conceptualization, Methodology, Resources, Software, Validation. **Malo Charrier:** Conceptualization, Methodology, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



(a) Image to analyze as captured by the camera

(b) Local orientations of the interlayer lines

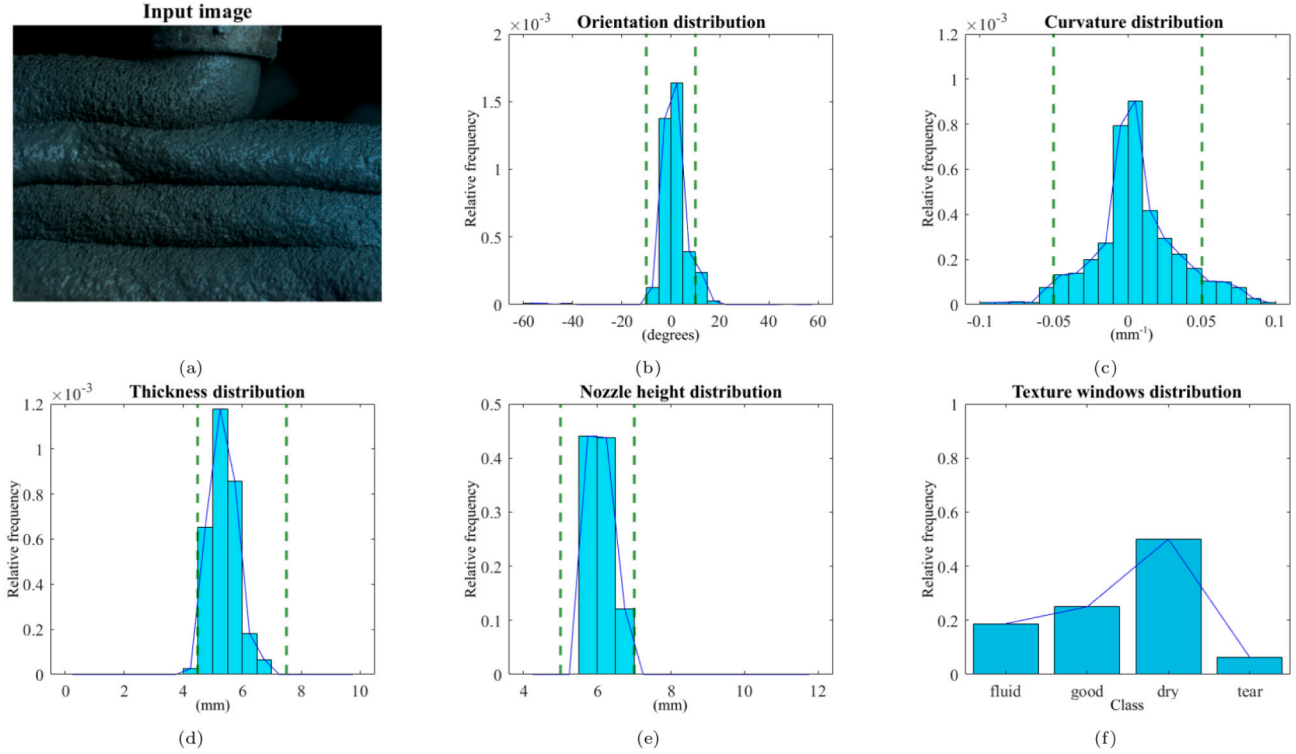
(c) Local curvatures of the interlayer lines

(d) Local thickness of the layers. The thickness is associated to the layer centers (in color); the white lines represent the interlayer lines

(e) Local heights of the nozzle with respect to the last printed layer

(f) Local textures. The white lines are the borders of the independently analyzed regions; the text shows the assigned class and probability

Fig. 21. Plots from the first study case. The black pixels in the plots b–e correspond to undefined values in the 2D space. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a)

(b)

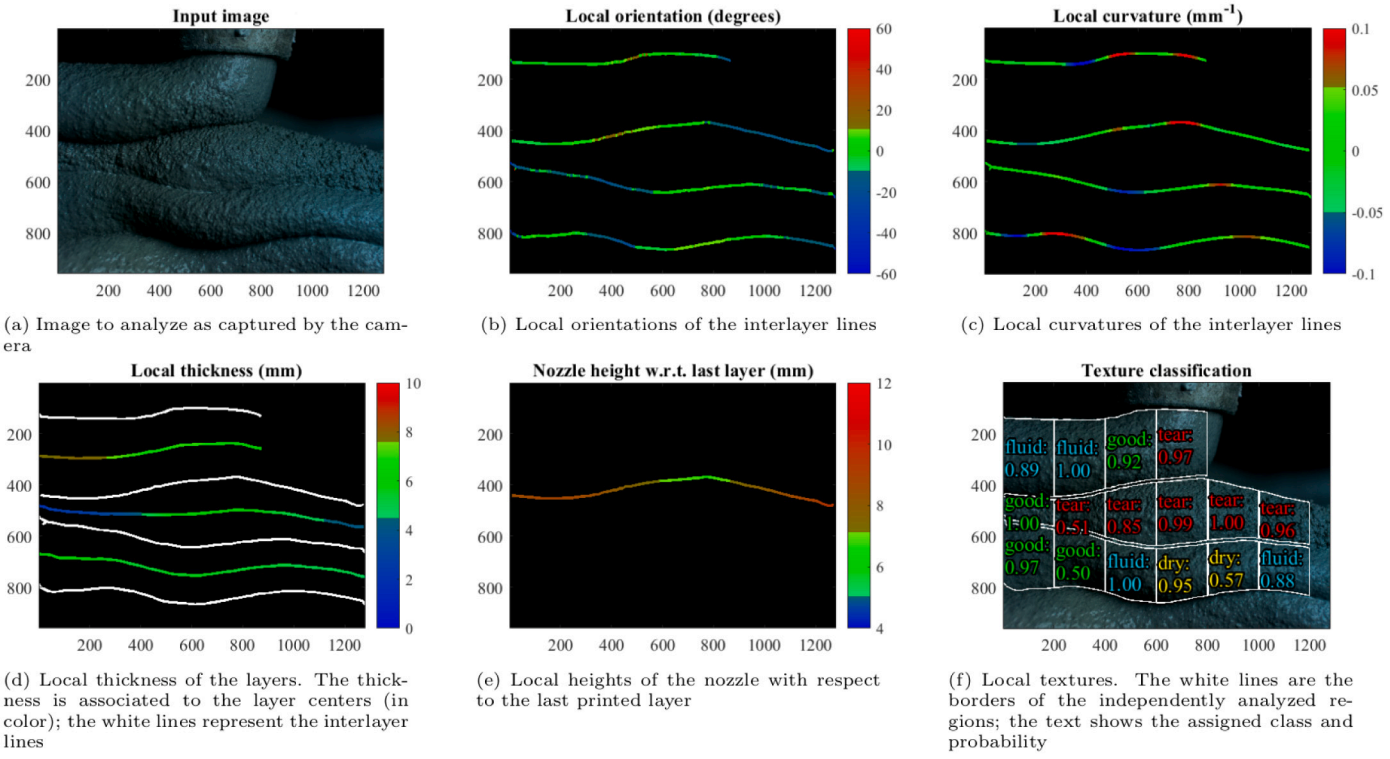
(c)

(d)

(e)

(f)

Fig. 22. Distributions from the first study case. The doted lines represent the range of admissible values; regions with values outside these ranges are detected as anomalies in Fig. 21.



(a) Image to analyze as captured by the camera

(b) Local orientations of the interlayer lines

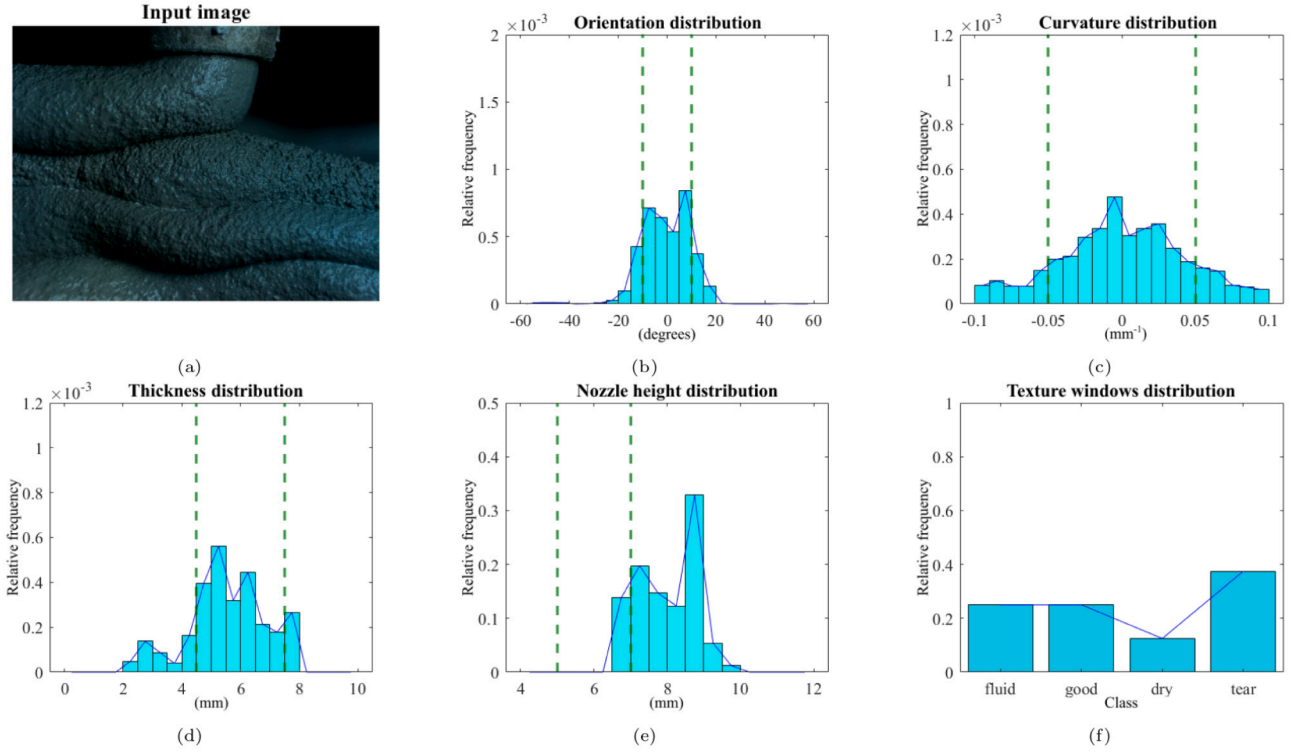
(c) Local curvatures of the interlayer lines

(d) Local thickness of the layers. The thickness is associated to the layer centers (in color); the white lines represent the interlayer lines

(e) Local heights of the nozzle with respect to the last printed layer

(f) Local textures. The white lines are the borders of the independently analyzed regions; the text shows the assigned class and probability

Fig. 23. Plots from the second study case. The black pixels in the plots b–e correspond to undefined values in the 2D space. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 24. Distributions from the second study case. The doted lines represent the range of admissible values; regions with values outside these ranges are detected as anomalies in Fig. 23.

Data availability

Links to Github repositories containing our data and source code, respectively, are provided in the conclusions of the article.

Acknowledgments

This work has been supported by the project DiXite. Initiated in 2018, DiXite (Digital Construction Site) is a project of the I-SITE FUTURE, a French initiative to answer the challenges of sustainable city. This research project was financially supported by I-SITE FUTURE, France (ANR-16-IDEX-0003).

References

[1] F. Barbosa, J. Woetzel, J. Mischke, Reinventing Construction: A Route of Higher Productivity, Tech. rep, McKinsey Global Institute, 2017.

[2] All3DP Pro, 8 biggest companies building 3D printed houses, 2021, All3DP Pro Available at https://all3dp.com/2/best-companies-building-3d-printed-houses/.

[3] I. Hager, A. Golonka, R. Putanowicz, 3D printing of buildings and building components as the future of sustainable construction? Procedia Eng. 151 (2016) 292–299, http://dx.doi.org/10.1016/j.proeng.2016.07.357, URL https://www.sciencedirect.com/science/article/pii/S1877705816317453.

[4] R. Buswell, P. Kinnell, J. Xu, N. Hack, H. Kloft, M. Maboudi, M. Gerke, P. Massin, G. Grasser, R. Wolfs, F. Bos, Inspection methods for 3D concrete printing, in: F.P. Bos, S.S. Lucas, R.J. Wolfs, T.A. Salet (Eds.), Second RILEM International Conference on Concrete and Digital Fabrication, in: RILEM Bookseries, Springer International Publishing, Cham, 2020, pp. 790–803, http://dx.doi.org/10.1007/978-3-030-49916-7_78.

[5] P. Wu, K.S. Ramani, C.E. Okwudire, Accurate linear and nonlinear model-based feedforward deposition control for material extrusion additive manufacturing, Addit. Manuf. 48 (2021) 102389, http://dx.doi.org/10.1016/j.addma.2021.102389, URL https://www.sciencedirect.com/science/article/pii/S2214860421005431.

[6] E. Shojaei Barjuei, E. Courteille, D. Rangeard, F. Marie, A. Perrot, Real-time vision-based control of industrial manipulators for layer-width setting in concrete 3D printing applications, Adv. Ind. Manuf. Eng. 5 (2022) 100094, http://dx.doi.org/10.1016/j.aime.2022.100094, URL https://www.sciencedirect.com/science/article/pii/S2666912922000228.

[7] P. Carneau, R. Mesnil, O. Baverel, N. Roussel, Layer pressing in concrete extrusion-based 3D-printing: Experiments and analysis, Cem. Concr. Res. (2022) 106741, http://dx.doi.org/10.1016/j.cemconres.2022.106741, URL https://www.sciencedirect.com/science/article/pii/S0008884622000321.

[8] R.J.M. Wolfs, F.P. Bos, E.C.F. van Strien, T.A.M. Salet, A real-time height measurement and feedback system for 3D concrete printing, in: D. Hordijk, M. Luković (Eds.), High Tech Concrete: where Technology and Engineering Meet, Springer International Publishing, Cham, 2018, pp. 2474–2483, http://dx.doi.org/10.1007/978-3-319-59471-2_282, URL http://link.springer.com/10.1007/978-3-319-59471-2_282.

[9] H. Lindemann, R. Gerbers, S. Ibrahim, F. Dietrich, E. Herrmann, K. Dröder, A. Raatz, H. Kloft, Development of a shotcrete 3D-printing (SC3Dp) technology for additive manufacturing of reinforced freeform concrete structures, in: T. Wangler, R.J. Flatt (Eds.), First RILEM International Conference on Concrete and Digital Fabrication – Digital Concrete 2018, Vol. 19, in: RILEM Bookseries, Springer International Publishing, Cham, 2019, pp. 287–298, http://dx.doi.org/10.1007/978-3-319-99519-9_27, URL http://link.springer.com/10.1007/978-3-319-99519-9_27.

[10] A. Kazemian, X. Yuan, O. Davtalab, B. Khoshnevis, Computer vision for real-time extrusion quality monitoring and control in robotic construction, Autom. Constr. 101 (2019) 92–98, http://dx.doi.org/10.1016/j.autcon.2019.01.022, URL https://linkinghub.elsevier.com/retrieve/pii/S0926580518307751.

[11] O. Davtalab, A. Kazemian, X. Yuan, B. Khoshnevis, Automated inspection in robotic additive manufacturing using deep learning for layer deformation detection, J. Intell. Manuf. (2020) http://dx.doi.org/10.1007/s10845-020-01684-w.

[12] S.A.O. Nair, G. Sant, N. Neithalath, Mathematical morphology-based point cloud analysis techniques for geometry assessment of 3D printed concrete elements, Addit. Manuf. (2021) 102499, URL https://www.sciencedirect.com/science/article/pii/S2214860421006461.

[13] N. Ducoulombier, R. Mesnil, P. Carneau, L. Demont, H. Bessaies-Bey, J.-F. Caron, N. Roussel, The "Slugs-test" for extrusion-based additive manufacturing: Protocol, analysis and practical limits, Cem. Concr. Compos. 121 (2021) 104074, http://dx.doi.org/10.1016/j.cemconcomp.2021.104074, URL https://www.sciencedirect.com/science/article/pii/S0958946521001438.

[14] C. Gosselin, R. Duballet, P. Roux, N. Gaudillière, J. Dirrenberger, P. Morel, Large-scale 3D printing of ultra-high performance concrete – a new processing route for architects and builders, Mater. Des. 100 (2016) 102–109, http://dx.doi.org/10.1016/j.matdes.2016.03.097, URL https://www.sciencedirect.com/science/article/pii/S0264127516303811.

[15] J. Archez, S. Maitenaz, L. Demont, M. Charrier, R. Mesnil, N. Texier-Mandoki, X. Bourbon, S. Rossignol, J.-F. Caron, Strategy to shape, on a half-meter scale, a geopolymer composite structure by additive manufacturing, Open Ceram. 5 (2021) 100071.

[16] R. Buswell, W. Leal de Silva, S. Jones, J. Dirrenberger, 3D printing using concrete extrusion: A roadmap for research, Cem. Concr. Res. 112 (2018) 37–49, http://dx.doi.org/10.1016/j.cemconres.2018.05.006, URL https://linkinghub.elsevier.com/retrieve/pii/S0008884617311924.

[17] A. Kazemian, B. Khoshnevis, Real-time extrusion quality monitoring techniques for construction 3D printing, Constr. Build. Mater. 303 (2021) 124520, http://dx.doi.org/10.1016/j.conbuildmat.2021.124520, URL https://www.sciencedirect.com/science/article/pii/S0950061821022765.

[18] B. Khoshnevis, Techniques for sensing material flow rate in automated extrusion, 2013, URL https://patents.google.com/patent/US8568121/en?oq=Techniquesor+ensingateriallow+atenutomatedxtrusion.

[19] A. Oleff, B. Küster, M. Stonis, L. Overmeyer, Process monitoring for material extrusion additive manufacturing: a state-of-the-art review, Prog. Addit. Manuf. (2021) 1–26.

[20] C. Gosselin, R. Duballet, P. Roux, N. Gaudillière, J. Dirrenberger, P. Morel, Large-scale 3D printing of ultra-high performance concrete – a new processing route for architects and builders, Mater. Des. 100 (2016) 102–109, http://dx.doi.org/10.1016/j.matdes.2016.03.097, URL https://www.sciencedirect.com/science/article/pii/S0264127516303811.

[21] V. Esnault, A. Labyad, M. Chantin, F. Toussaint, Experience in online modification of rheology and strength acquisition of 3D printable mortars, in: RILEM International Conference on Concrete and Digital Fabrication, Springer, 2018, pp. 24–38.

[22] D. Rutten, Grasshopper, Robert McNeel and Associates, 2010.

[23] T. Schwartz, HAL: Extension of a visual programming language to support teaching and research on robotics applied to construction, in: Rob| Arch 2012, 2013, pp. 92–101, http://dx.doi.org/10.1007/978-3-7091-1465-0_8.

[24] Z. Liu, M. Li, Y.W.D. Tay, Y. Weng, T.N. Wong, M.J. Tan, Rotation nozzle and numerical simulation of mass distribution at corners in 3D cementitious material printing, Addit. Manuf. 34 (2020) 101190.

[25] R.P. Ltd, Buy a Raspberry Pi High Quality Camera. URL https://www.raspberrypi.com/products/raspberry-pi-high-quality-camera/.

[26] R. Rill-García, E. Dokladalova, P. Dokládal, Pixel-accurate road crack detection in presence of inaccurate annotations, Neurocomputing 480 (2022) 1–13, http://dx.doi.org/10.1016/j.neucom.2022.01.051, URL https://www.sciencedirect.com/science/article/pii/S0925231222000728.

[27] P. Soille, Morphological operators with discrete line segments, in: International Conference on Discrete Geometry for Computer Imagery, Springer, 2000, pp. 78–98.

[28] Morphological operations on binary images (r2020b), The MathWorks, Inc, URL https://fr.mathworks.com/help/releases/R2020b/images/ref/bwmorph.html.

[29] C. De Boor, C. De Boor, A Practical Guide To Splines, Vol. 27, springer-verlag New York, 1978.

[30] C.R. Maurer, R. Qi, V. Raghavan, A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2) (2003) 265–270.

[31] R.M. Haralick, K. Shanmugam, I.H. Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. (6) (1973) 610–621.

[32] T. Ojala, M. Pietikainen, D. Harwood, Performance evaluation of texture measures with classification based on Kullback discrimination of distributions, in: Proceedings of 12th International Conference on Pattern Recognition, Vol. 1, 1994, pp. 582–585 vol.1, http://dx.doi.org/10.1109/ICPR.1994.576366.

[33] R. Amhaz, S. Chambon, J. Idier, V. Baltazart, Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection, IEEE Trans. Intell. Transp. Syst. 17 (10) (2016) 2718–2729, http://dx.doi.org/10.1109/TITS.2015.2477675, Conference Name: IEEE Transactions on Intelligent Transportation Systems.

[34] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic road crack detection using random structured forests, IEEE Trans. Intell. Transp. Syst. 17 (12) (2016) 3434–3445, http://dx.doi.org/10.1109/TITS.2016.2552248, Conference Name: IEEE Transactions on Intelligent Transportation Systems.