ONLINE VISION-BASED ERROR DETECTION FOR FUSED FILAMENT FABRICATION

ABURAIA, A., HOLZGETHAN, C., AMBROS, C.,
STUJA, K. & KATALINIC, B.

Abstract: *Fused Filament Fabrication is a promising 3D printing technology, as it allows the creation of complex structures and prototypes in a relatively short time with low costs. Despite continuous development, the technology remains error-prone, often resulting in unusable end-products. Since a 3D printing job can take multiple hours, the unnoticed failure of a print does not only lead to wastage of materials but also to wastage of time and can even result in damaging the 3D printer. In this paper, an autonomous print monitoring system is proposed, which can detect the detachment of 3D printed objects during the printing process and terminate faulty printing jobs. The developed algorithm automatically starts at the launch of a printing job and regularly analyses the state of the print by comparing a camera-captured image to a visualization of the G-Code of the print at different stages in the printing process. The results show that the algorithm successfully detects and terminates printing jobs with part detachment at runtime and therefore reduces the waste of materials and time and the risk of printer damage.*

Keywords: *Error detection, Image processing, 3D printing, Fused Filament Fabrication*

Authors' data: Aburaia A[li]*, Holzgethan C[audia]*, Ambros C[lemens]*, Stuja K[emajl]*, Katalinic B[ranko]**, *University of Applied Sciences Technikum Wien, Höchstädtplatz 6, 1200 Wien, Austria **Vienna University of Technology, Getreidemarkt 9/311, 1040 Wien, aburaiaa@technikum-wien.at

This Publication has to be referred as: Aburaia, A[li]; Holzgethan, C[audia]; Ambros C[lemens]; Stuja K[emajl] & Katalinic B[ranko] (2021). Online Vision-based Error Detection for Fused Filament Fabrication, Chapter 16 in DAAAM International Scientific Book 2021, pp.193-212, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-31-0, ISSN 1726-9687, Vienna, Austria
DOI: 10.2507/daaam.scibook.2021.16

1. Introduction

Additive manufacturing, popularly known as 3D printing, is becoming increasingly popular in different industries and is moving from its initial function of prototyping into manufacturing production parts. Its ability to produce objects with complex geometric structures paves the way to be employed for a wide variety of industrial applications. One of the most known additive manufacturing technologies is Fused Filament Fabrication (FFF). Similar to typical additive manufacturing technologies, FFF also involves applying materials layer by layer to create three-dimensional (3D) objects. During this process, a thermoplastic filament is fed through a temperature-controlled extrusion head and is deposited onto a building platform. This computer-controlled process is repeated until the desired object is fully shaped.

Despite several advantages and wide applicability in different fields, FFF has lower reliability and part quality due to lower structural rigidity and adherence between layers. Compared to conventional manufacturing processes, e.g. injection molding, and other additive manufacturing processes, objects printed via FFF not only have different geometric and mechanical properties but also improvable surface textures. Quality differences are not only evident in comparison to other manufacturing processes, but also highly reflected in practical observation, as different qualities can often be expected from two identical 3D printers producing identical parts. Quality requirements increase even more when FFF is positioned in an automated industrial application, especially since production chains should not be interrupted and downtimes must be prevented. In general, higher reliability is required in order to further expand the practical usage of additive manufacturing technologies in automated production environments.

In this paper, we present an approach to implement an online vision-based error detection algorithm within automated Fused Filament Fabrication processes in order to achieve sufficient process capability and reliability.

2. State of the art

Fused Filament Fabrication is a promising 3D-printing technology, as it allows the creation of complex structures and prototypes in a relatively short time with low costs compared to other manufacturing techniques (Langeland, 2020, Baumann and Roller, 2016). Unfortunately, despite its popularity and constant development, the technology is still prone to failure (Petsiuk and Pearce, 2020, Lyngby et al., 2017). Common errors that can occur during FFF are warping, clogging of the nozzle, part detachment, shifts in layers, and volumetric deviations (Becker et al., 2020a). Even though some of these errors can be purely aesthetic, often they lead to the uselessness of the finished printing object. Since a 3D-printing job can take multiple hours, the unnoticed failure of a print does not only lead to wastage of materials but also to wastage of time and can even conclude in damaging the 3D printer (Langeland, 2020). Due to the long printing times, manual supervision is not practical. Therefore, many papers focus on developing autonomous monitoring technologies. Common approaches can be divided into vision-based and non-vision-based concepts.

2.1 Non-vision-based concepts

In the domain of non-vision-based 3D-print monitoring, concepts have been developed, that utilize signals such as electrical current (Tlegenov et al., 2019), vibration (Li et al., 2019, Zhao et al., 2020), or acoustics (Becker et al., 2020b) to detect a failed print.

Tlegenov et al. (2019) focused on monitoring the condition of the 3D-printer nozzle by examining the electric current of the extruding motors. The results showed that complete as well as partial clogging of the nozzle lead to higher values of the current. These influences can be detected by sensors and the print can be stopped before serious process failures occur.

Vibration data has been utilized in (Li et al., 2019) and (Zhao et al., 2020) to monitor printing processes and detect errors. The former uses the least square support vector machine to classify vibration data in normal and filament jam states. Furthermore, a neural network was employed to detect warpage and abnormal leakage of the filament (Li et al., 2019). (Zhao et al., 2020) on the other hand, used a random forest model to differentiate between normal and filament jam states.

Becker et al. (2020b) trained a Long Short-Term Memory model to classify and differentiate between multiple different relevant sounds that can occur during FDM. The goal of this concept was to detect errors that can occur due to the wrong calibration of the nozzle's height above the print-bed but also predict the state of the printer depending on the fan and printer noises.

2.2 Vision-based concepts

Vision-based concepts utilize camera systems to detect printing errors. A common approach is to compare the 3D-printed object to a simulation of what the object should look like at different stages in the printing process. Lyngby et al. (2017) compare a camera image of the actual print to a perspective projection of the CAD model. To counteract issues between the light, scene, and workpiece, the authors segment the images and compare their segmentation masks. Errors that could be detected with this method are workpiece warpage, extrusion failure, and the detachment of parts. A similar approach was developed in (Petsiuk and Pearce, 2020).

Here, the printed object is compared to a 3D representation of the G-Code at different stages of the print. Besides the external shape, also the internal structure of the layers is monitored to track errors and generate applicable printer actions. Langeland (2020) states that the successful print of the first layer is crucial for 3D printing, as it serves as the foundation for the rest of the print. Therefore, they developed an algorithm that analyses the first layer by comparing the number of contours in a camera-captured image to the number of contours in a simulation of the printed model. If the first layer was printed successfully, a second camera, which is placed in front of the printer, tracks the distance between the nozzle and the 3D-printed object to detect missing material flow.

Other approaches exist that are not based on comparing the 3D-printed object to a simulation. Baumann et al. (2016) use a camera to track the position of the object and the nozzle to detect missing material flow, deformed objects, and part detachment. In (Chen et al., 2019) a camera is used to capture top-down images of the print at different stages during the printing process.

The images are processed by applying image processing methods and classified into defective or non-defective groups. Hu moments are then used to extract information from the images. Okarma and Fastowicz (2016) focus on a non-reference assessment, which means that they do not compare the print to a reference image. Instead, the surface of the printed object is inspected by taking side view images. The authors assume that horizontal lines in the printed object correspond to good quality, whilst irregularities imply bad quality or defects.

3. Problem description

Several different errors can occur during Fused Filament Fabrication, such as warping, clogging of the nozzle, or low-quality surfaces. Errors considered in this paper are shifts in layers and the detachment of 3D-printed objects from the print-bed during the printing process, although the focus is put on the latter.

3.1 Detachment of 3D-printed objects

The detachment of 3D-printed objects can have different causes like temperature fluctuations in the printer environment, insufficient bed calibration, or vibrations and shock (Salentijn et al., 2017, Baumann and Roller, 2016).

Depending on the severity of the detachment, a continuation of the print might lead to a waste of materials and time and can even result in damage to the machine. Two cases can be differentiated when considering the consequences of part detachment: a) only a portion of the 3D-printed object detaches or b) the whole object detaches.

A - Partial detachment

Partial detachment refers to the situation when only a part of the 3D-printed object separates from the print-bed. This can result in different levels of the severeness, depending on the amount of filament that separated from the print-bed. The printing process does not have to be terminated in every case.

Fig. 1. Example of an irreparable partly detached printed object

Figure 1 shows an example of a 3D-printing job that is irreparable. In the image, a big portion of one-half of the 3D-printed object got detached from the print-bed. A continuation of the print would lead to a waste of materials and time since the finished product will not be useful. Additionally, the risk arises, that the whole object detaches from the print-bed.

B - Complete detachment

If the whole printing object detaches from the print-bed, severe consequences can occur. The detachment often leads to the object being dragged along by the printhead, whilst the nozzle still extrudes filament. Since the freshly extruded filament cannot adhere to the print-bed anymore, it instead gathers around the printhead, forming an ever-growing filament blob. Depending on the size of the object to be printed and the amount of time that passes until the error gets recognized, the filament blob can irreparably damage the nozzle and cables around the printhead.

Figure 2 shows two stages of the formation of a filament blob. In the left picture, only a small amount of filament got stuck on the printhead before the print got stopped. In the right image, more time passed until the error got detected by the user, and a bigger filament blob formed. In the case of the right image, the cables on the printhead and the nozzle could not be saved and had to be replaced, which led to unwanted downtime, additional costs, and the necessity to recalibrate the printer. Early detection of part detachment is crucial to prevent such consequences.

Fig. 2. Forming of a filament blob on the print-head (left: yellow filament, small blob, right: yellow filament, more severe blob)

3.2 Layer shifting

Another undesirable error that can occur during Fused Filament Fabrication is layer shifting. Layer shifting signifies that, due to impact, subsequent layers are printed with a horizontal offset to previous layers, resulting in an unusable finished product. An example is depicted in Figure 3.

Fig. 3. Example of layer shifting

Experiments showed that this error can occur if the print-bed and printhead get misaligned during the printing process. Possible causes can be movement of the print-bed due to impacts or de-calibration of the printhead due to collisions. In most cases layer shifting results in unusable 3D-printed objects and consequently material and time wastage.

4. Materials and methods

The autonomous print monitoring algorithm presented in this paper focuses on comparing the 3D-printed object to a visual representation of the G-Code at different stages during the printing process. In general, this is achieved by employing a camera module, which enables taking pictures of the 3D-printed object and the G-Code visualizer feature of OctoPrint (OctoPrint, 2021). The following section gives a detailed description of the hard- and software used and the functionality of the algorithm.

4.1 Experimental setup

The setup for this project is shown in Figure 4. It mainly consists of a FFF 3D printer, a camera module, and a Raspberry Pi. The 3D printer that is used is the Prusa i3 MK3S+. As a camera module, the Raspberry Pi Camera Module 2 was chosen. The camera contains an 8-megapixel sensor and offers an image resolution of 3280 x 2464 pixels. It is mounted centered above the print-bed, facing downward to obtain a top-down view of the 3D-printed object. To control the printer and process the images, a Raspberry Pi 4 is used.

Fig. 4. Experimental setup

4.2 Software

The communication between the 3D printer and the Raspberry Pi is achieved via the open-source software OctoPrint (OctoPrint, 2021). The software allows to obtain information about the print-job and printer state and to remote control the printer. Some key features of OctoPrint, that were used in this project, are the possibility of starting, pausing, and terminating printing jobs remotely, obtaining job information, and gaining a visual representation of the current print stage as computed from the G-Code of the print. For the processing of images, the open-source library OpenCV (OpenCV, 2021) has been chosen.

4.3 Print monitoring

An overview of the developed print monitoring algorithm is depicted in Figure 5. The algorithm checks if a print job is started by sending REST API requests to OctoPrint. As soon as a job was sent to the printer, the algorithm starts with the monitoring routine. During the print, the program continuously checks the number of the current layer. As soon as a layer checkpoint is reached, the print gets paused and the print-analysis is started. Layer checkpoints are placed at the start of the second, third and fifth layer and then at the start of every n^{th} layer.

The value of n depends on the application and the geometry of the 3D-printed object. A small value of n , like $n = 5$, signifies a high frequency of print analyses. This is especially useful when the 3D-printed object has a small surface area and therefore a higher risk of detaching from the print-bed. For normal printing jobs, a value of about $n = 50$ proved to be sufficient. The high number of print analyses at the beginning of the print was chosen, as it could be observed, that part detachments have the highest chance of occurring at the first three to five layers.

Fig. 5. Print monitoring process

The analysis of the print is composed of the following steps:

- Pause the print
- Move the printhead to the side and place the print-bed centered underneath the camera
- Take a picture of the 3D-printed object on the print-bed and apply image processing
- Take a screenshot of the G-Code visualizer on the OctoPrint website and apply image processing
- Compare the two images and calculate the percentage of match
- Continue or terminate the print based on the value of the match

The instructions to pause, continue or terminate the print as well as the instructions to move the printhead and print-bed, are sent to the printer via REST API commands. Since the picture of the printed object and the screenshot of the visualized G-Code cannot be compared in their raw form, image processing must be applied to both. The image processing procedure and comparison of the images are described in more detail in the next two sections.

4.4 Image processing

Before the picture of the 3D-printed object and the G-Code image can be compared, image processing must be applied. The first step is to crop the images to remove unnecessary information. In the case of the camera picture, the image is cropped to only show the print-bed and remove the background. In the case of the OctoPrint screenshot, only the G-Code visualizer section is of interest.

Gaussian blur is applied to the camera image to remove noise from the raw pixel data. Next, the camera image is converted from the BGR to the HSV color space, and thresholds are applied to binarize the image. Following threshold values were applied in OpenCV to differentiate the yellow 3D-printed object from the print-bed:

• Hue:	4 to 40	(OpenCV range: 0-180)
• Saturation:	100 to 255	(OpenCV range: 0-255)
• Value:	90 to 255	(OpenCV range: 0-255)

These values were chosen to counteract different lighting conditions and overcome shadows, which occur on the inside of the printed object due to the infill. In the case of the G-Code image, the OctoPrint visualizer contains an unwanted grid. In order to remove it, the G-Code image is converted into a greyscale image, and thresholds are applied.

The following values were chosen:

- Lower limit: 185
- Upper limit: 255

To remove noise in the binarized images, the morphological operations “Open” and “Close” are used. This image processing procedure is exemplified in Figure 6.

Fig. 6. Image processing: 1a) original camera image, 1b) original G-Code image, 2a) & 2b) cropped images, 3a) & 3c) binarized images

4.5 Comparison of images

In order to compare the binarized images, they have to be cropped further and scaled to the same size. To achieve this, bounding boxes are applied to extract the printed object from the background and to define where to crop the images. Since the cropped images still vary in the number of pixels, the G-Code image is then resized to match the size of the camera image. This process can be seen in Figure 7.

The value of the match of the two images is calculated by comparing every pixel in the camera image with the corresponding pixel in the G-Code image via the OpenCV XOR command. The result is a new image, showing the differences between the actual print as captured by the camera and the supposed print as calculated from the G-Code.

Fig. 7. Cropped images: 1a) bounding box in binarized camera image, 1b) bounding box in binarized G-Code image, 2a) & 2b) cropped images

An example is depicted in Figure 8. Black pixels indicate a match of the camera and G-Code image, whilst white pixels show the differences. As can be observed in the figure, even though the 3D-printed object is printed without failure (see Figure 6 and Figure 7), there are still differences between the camera and G-Code image. These differences appear because the camera is not mounted perfectly parallel to the print-bed and therefore the resulting image suffers from lens distortions.

Fig. 8. Computed image showing the differences between the camera image and the G-Code image

The overall match of the camera and G-Code image is computed by calculating the proportion of black or matching pixels in comparison to the overall number of pixels in the image, as described by equation (1).

$$match = \frac{\text{num matching pixels}}{\text{num all pixels}} \cdot 100 \quad (1)$$

In the case of Figure 8, the overall match amounts to 96.77%. This value is then compared with a threshold to decide if the print is successful or should be terminated.

The threshold depends on the application and desired accuracy. Experiments for this project setup showed that a threshold of 90% led to the best results when determining if the print should be aborted or continued.

5. Results

The robustness of the algorithm was examined by performing more than 50 3D-printing jobs of varying models on nine separate Prusa i3 MK3S+ 3D printers. Additional technical data and examples of the proposed algorithm are stated in the sections below.

5.1 Technical Data

Since the active print gets paused to perform the print analysis, the required time of the analysis is an important factor, as unnecessary delays of the printing are undesirable.

To determine the required analysis time, 20-time measurements were taken on three separate 3D printers and the mean values were calculated. The measurement was started as soon as the print got paused and stopped when the print-head was back at its original position. The results are summarised in Table 1 below.

printer	1	2	3
mean value of time	20.94 sec	21.04 sec	20.75 sec

Tab. 1. Mean values of required print analysis time on three separate printers

Thus, an average time requirement of 20.91 seconds per print analysis can be concluded. Compared to the typically long printing times of several hours this value proves satisfactory for this paper's project setup.

5.2 Part detachment

In order to determine the performance of the algorithm in detecting the detachment of 3D-printing parts, errors were provoked, by either printing objects with thin geometries that are likely to detach or loosening the printing parts manually. The respective results for detecting partial and complete detachments of printing objects are summarised below.

A - Partial Detachment

The algorithm was tested with 20 separate partial detached 3D-printed objects. The respective camera images and binarized G-Code images of three of the tests are exemplified in Figure 9.

Fig. 9. Examples of partial detachment (left: camera images, middle: binarized G-Code images, right: the match between the camera and G-Code image)

In the figure the first column shows the camera images of the printed object, the middle column shows the binarized G-Code image, and the last column shows the resulting differences between the camera and G-Code image. The first example concerns a printing job, where a big portion of the printed object separated from the print-bed and got deformed due to impact with the extruder. In the second example, the lower half of the printed object detached from the surface and got slightly moved by the extruder. In the case of the third example, only small sections of the printed object separated from the print-bed.

For this paper's project, this last example does not classify as a failed print, as the print may succeed in a usable end-product if it is continued. The resulting matches for these examples are listed in Table 2.

Example	Computed Match	State of the print	Correctly classified?
Example 1	47.91%	failed	Yes
Example 2	75.85%	failed	Yes
Example 3	91.48%	succeeded	Yes

Tab. 2. Exemplary matches of three partially detached print jobs

As can be observed from the print in example 3, a match of 91.48% was calculated. Since this percentage is higher than the defined threshold of 90%, the print was rightfully classified as successful and could be continued. In the other cases, a percentage lower than the threshold was computed and the prints were rightfully aborted. In total, a success rate of 85% could be obtained for the performed test prints.

B - Complete Detachment

Fig. 10. Examples of complete detachment (left: camera images, middle: binarized G-Code images, right: a match between the camera and G-Code image)

Complete detachment refers to the situation where the printed object entirely separates from the print-bed and gets dragged along by the printhead, which can conclude in damage to the printer. Similar to the section above, the algorithm was tested with 20 separate cases of this error. The images of three of these cases are depicted in Figure 10.

Example 1 shows a situation where the whole object detached from the print-bed and got dragged away by the printhead. In the second example, the left half of the print completely separated from the printing surface and got pulled away. The third example indicates a printing job where three of nine individual parts detached. The computed matches of these examples are summarised in Table 3. Since nothing can be found in the binarized camera image in cases like example 1, the match is automatically set to zero.

Example	Computed Match	State of the print	Correctly classified?
Image 1	0.00%	failed	Yes
Image 2	64.65%	failed	Yes
Image 3	78.66%	failed	Yes

Tab. 3. Exemplary matches of three completely detached print jobs

In general, the error computed match of completely detached objects is lower than the one of partially detached objects. Thus, it is easier for the algorithm to detect this failure. For the 20 test cases, an overall success rate of 95% could be computed. However, there is a possibility that the algorithm does not detect detached objects if their size is too small in comparison to the overall size of the printed object.

5.3 Layer shifting

Even though the focus of the proposed algorithm is put on detecting detached 3D-printed objects during the printing process, the detection of layer shifts was tested additionally. Since this type of error did not occur naturally during the testing period, it was manually simulated by changing the print-beds position during the printing process and consequently misaligning the printed objects with the extruder. In total, ten tests of layer shifting were carried out. Figure 11 shows three examples of this error. In examples 1 and 2 the print-bed was moved by about half a centimeter in both directions. In example 3, the print bed was additionally rotated slightly. Table 4 shows the respective results.

Example	Computed Match	State of the print	Correctly classified?
Image 1	82.39%	failed	Yes
Image 2	92.62%	failed	No
Image 3	85.86%	failed	Yes

Tab. 4. Exemplary matches of three cases of layer shifting

Fig. 11. Examples of layer shifting (left: camera images, middle: binarized G-Code images, right: a match between the camera and G-Code image)

As can be seen in Figure 11 and Table 4, the algorithm has difficulties with detecting the discrepancies in layer shifting. This can be explained by the fact that the G-Code image is resized to match the number of pixels in the camera image.

As stated in section 4.5, this is necessary for the pixel-wise comparison of the images. However, because of the resizing, only small differences remain between the images. The success rate of detecting layer shifting during the test phase amounted to 60%.

5.4. Limitations

The experiments showed that the algorithm can fulfill its requirement of detecting and aborting prints with detached objects and therefore reducing the waste of time and materials and the risk of printer damage. However, the algorithm still comes with limitations that are mainly caused by image processing. Some of these limitations will be considered in detail in this section.

A - Lighting

Correct lighting plays a key factor in image processing as it considerably affects the colors in the image. Even though, different lighting conditions could be compensated by choosing wide ranges for the thresholding of the HSV values (see Section 4.4), some forms of lighting are still troublesome.

Figure 12 shows an image that was taken during the print monitoring process, where the light of a nearby window shined directly on the print bed. Consequently, sections of the printed object appeared almost completely white in the camera image.

Fig. 12. Example of poor lighting, the object in the camera image appears almost completely white

Unfortunately, the HSV thresholds cannot be extended to include complete white, as then the risk of detecting reflections of the print-bed arises. The binarized camera and G-Code images as well as the computed match are depicted in Figure 13.

Fig. 13. Poor lighting: comparison of the camera and G-Code image – match: 74.15%.

As can be seen, the section that was directly illuminated by the light is not detected by the image processing. Therefore, only a match of 74.15% can be computed. In this case, since the percentage is lower than the 90% threshold, the algorithm would mistakenly abort the print, even though no actual failure is present.

B - Filament Colour

The color of the filament is another factor that can dampen the success of image processing. Experiments with different filament colors showed, that the color grey is not suited for the setup of this project, as it is too similar to the color of the print-bed.

Fig. 14. Image processing with grey filament

Even after trying different HSV thresholds no better results than depicted in Figure 14 could be achieved. Therefore, the algorithm is limited to be used with filament colors that differentiate enough from the color of the used print-bed.

6. Conclusion and outlook

This paper proposed an algorithm that can detect part detachment by comparing the 3D-printed object with a visual representation of its G-Code at different stages during the print. To do so, the print gets paused at defined layer checkpoints. During the pause, a camera captures a top-down view of the printed object and a screenshot is taken of the G-Code visualizer in the OctoPrint web interface. After applying image processing operations, the overall match between the camera captured image and the G-Code image is calculated. If the match exceeds a pre-defined threshold value, the print is classified as successful, and the printer continues printing. If not, the print gets automatically aborted.

The conducted experiments showed that partial as well as complete detachment of the printed object can be successfully detected. Detecting shifts between the layers however proves to be difficult. Extreme lighting conditions and filament colors that are too similar to the print-bed can have negative effects on the proposed algorithm.

In order to further improve the detection of completely detached objects, a similar approach as in (Langeland, 2020) could be attempted in the future, where the number of contours or objects in the images are compared. Furthermore, vibration sensors could be added to the print-bed and printhead to detect impacts that might result in layer shifting. To counteract poor lighting an active illumination source should be employed as well as a protection against external light disruptions.

7. References

- Baumann, F., Roller, D., 2016. Vision-based error detection for 3D printing processes. In: 2016 International Conference on Frontiers of Sensors Technologies, 12-14 March, Hong Kong. Hong Kong, China: MATEC Web of Conferences, pp. 1-7.
- Becker, P., Spielbauer, N., Roennau, A., Dillmann, R., 2020a. Real-Time In-Situ Process Error Detection in Additive Manufacturing. In: 2020 Fourth IEEE International Conference on Robotic Computing (IRC), 9-11 November 2020, Taichung. Taichung, Taiwan: IEEE, pp. 426-427.
- Becker, P., Roth, C., Roennau, A., Dillmann, R., 2020b. Acoustic Anomaly Detection in Additive Manufacturing with Long Short-Term Memory Neural Networks. In: 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), 16-21 April 2020, Bangkok. Bangkok, Thailand: IEEE, pp. 921-926.
- Chen, W., Ho, J., Mustapha, K., Chai, T., 2019. A Vision-Based System for Anomaly Detection and Classification in Additive Manufacturing. In: 2019 IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies (CSUDET), 7-9 November 2019, Penang. Penang, Malaysia: IEEE, pp. 87-92.
- Langeland, A., 2020. Automatic Error Detection in 3D Printing using Computer Vision. [Master Thesis] Bergen: Western Norway University of Applied Sciences, Master's program: Software Engineering.

- Li Y., Zhao, W., Li, Q., Wang, T., Wang, G., 2019. In-Situ Monitoring and Diagnosing for Fused Filament Fabrication Process Based on Vibration Sensors. *Sensors*. 19(11), pp. 1-18.
- Lyngby, R., Wilm, J., Eiriksson, E., Nielsen, J., Jensen, J., Aanaes, H., Pederson, D., 2017. In-line 3D print failure detection using computer vision. In: euspen and ASPE Special Interest Group Meeting: Additive Manufacturing, 10-11 October 2017, Leuven. Leuven, Belgium: European Society for Precision Engineering and Nanotechnology (euspen), pp. 1-4.
- OctoPrint, [Online] Available at: <<https://octoprint.org/>> [Accessed on 14 November 2021]
- Okarma, K., Fastowicz, J., 2016. No-reference quality assessment of 3D prints based on the GLCM analysis. In: 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR), 29 August – 1 September, 2016, Miedzyzdroje. Miedzyzdroje, Poland: IEEE, pp. 788-793.
- OpenCV, [Online] Available at: <<https://opencv.org/>> [Accessed on 14 November 2021]
- Petsiuk, A., Pearce, J., 2020. Open Source Computer Vision-based Layer-wise 3D Printing Analysis. *Additive Manufacturing*. 2020(36), pp. 1-17.
- Salentijn, G., Oomen, P., Grajewski, M., Verpoorte, E., 2017. Fused Deposition Modeling 3D Printing for (Bio)analytical Device Fabrication: Procedures, Materials, and Applications. *Analytical Chemistry*. 89(13), pp. 7.053-7.061.
- Tlegenov, Y., Feng Lu, W., Hong, G., 2019. A dynamic model for current-based nozzle condition monitoring in fused deposition modeling. *Progress in Additive Manufacturing*. 4(3), pp. 211-223.
- Zhao, Y., Li, Y., Wang, W., Wang, G., 2020. In-Situ Monitoring of Additive Manufacturing Process Based on Vibration Data. In: 2020 4th Annual International Conference on Data Science and Business Analytics (ICDSBA), 5-6 September 2020, Changsha. Changsha, China: IEEE, pp. 155-158.