Tomas Paulsen

Visual error detection on 3D printing

Bachelor's project in Automation Engineering
Supervisor: Ottar Osen, Saleh Abdel-Afou Alaliyat
December 2019

Bachelor's project

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

NTNU
Department of ICT and
Natural Sciences

Visual error detection on 3D printing

Bachelor Thesis

Tomas Paulsen

Supervisors
Ottar Osen
Saleh Abdel-Afou Alaliyat

Ålesund
Desember 16, 2019

Abstract

The process of 3D printing has become more available to the public, due to the increase of affordable consumer FDM printers on the market. This process is however not without errors. Most consumer printers do not have any way of detecting errors and will continue its instructions regardless of the physical result. Errors usually results in waste of materials, energy and time. The process of 3D printing objects can take anywhere from 10 minutes to several days. It is therefore tedious for the user to monitor the whole process to detect errors.

This paper presents the development of a system for error detection in 3D printing using computer vision. The system uses a raspberry pi, with camera to visually track and monitor the objects printed. This system utilizes image analysis to monitor in real time the process of 3D printing objects. The system detects and alerts errors of filament runout and vertical shift of the object.

The error detection compares consecutive images of each layer the printer builds. Key points in the object is tracked and compared with key points in previous images to determine if an error is present. Results from testing the system shows detection of common errors and handling of false detection.
The result of this paper is a system that uses computer vision to detect errors in 3D printing.

Norsk:
3D-skriving har blitt meir tilgjengeleg for forbrukarar på grunn av auke av rimelege FDM-skrivarar i marknaden. Prosessen med 3D-skriving er ikkje nødvendigvis utan feil. Dei fleste av forbrukarskrivarane har ikkje nokon måte å oppdaga feil på, og vil difor fortsette instruksjonane sine uavhengig av det som fysisk kjem ut av skrivaren. Feil resultera som regel i sløsing av material, straum og tid. Prosessen med å 3D-skrive objekt kan ta alt frå 10 minutt til fleire dagar. Der er difor lite hensiktsmessig for brukaren å måtte halde auge med heile prosessen for å oppdaga feil.
Denne oppgåva legg fram utviklinga av eit system for feildeteksjon i 3D-skriving ved bruken av maskinsyn. Systemet nyttar ein Raspberry Pi med kamera for å visuelt spore og overvaka objekt som blir skrive ut. Dette systemet nyttar bildeanalyse for å overvaka prosessen i

sanntid. Systemet klarar å oppdage og varsle feil dersom skrivaren går tom for materialar og forskyving av objektet.
Feiloppdaginga nyttar ei samanlikning av påfølgjande bilde av kvart lag skrivaren legg ned. Nøkkelpunkt i objektet blir spora og samanlikna med nøkkelpunkt i tidlegare bilde for å avgjera om det er ein feil i skrivinga. Resultata frå testing av systemet synar oppdaging av vanlege feil og handtering av falske feil.
Resultatet av denne oppgåva er eit system som nyttar maskinsyn for å oppdage feil i 3D-skriving.

Preface

This bachelor thesis is the result of a bachelor's project in Automation Engineering at the Norwegian University of Science and Technology. The motivation behind this thesis came from my own need for a system to detect errors in the 3D printing process. Therefor the project was self-developed but with good help from a few people I would like to acknowledge.

I would like to thank everyone who has supported and aided to this project, especially:

- Ingrid Kvammen Paulsen. For the essential support and love.
- Ottar L. Osen and Saleh Abdel-Afou Alaliyat. For guidance and supervision.
- Arne Styve and Girts Strazdins. For guidance in programming architecture and design.
- Anders Sætersmoen, Øivind Andre Hanken and Markus Gutvik Lyngstad. For supplying hardware and location for setup.

This project is open source and anyone who wants to further develop it are welcome to contact me. The full source code can be found at Github (Paulsen, 2019)

Tomas Paulsen

Ålesund, December 2019

Abbreviations

3D	Three dimensional
BLOB	Binary Large Object
FDM	Fused deposition modelling
FoV	Field of view
GCODE	Programming language used by 3D printers
GPIO	General Purpose Input/ Output
HSV	Hue Saturation Value
IDE	Integrated development environment
OOP	Object oriented programming
OS	Operating System
RAM	Random Access Memory
RGB	Red Green Blue
Rpi	Raspberry Pi
SLICER	Software to slice 3D model into layers
UX	User experience

Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background

The process of 3D printing has become more available to the public, due to the rise of affordable consumer printers on the market. 3D printing is a technology that can create physical objects from digital 3D models. It is an additive manufacturing process that has become consumer friendly and requires little knowledge to get started.

However, the process of printing out parts is not always without errors. An error in the printing process is any misplacement, deformation or flawed of the part that is not wanted in the result. If an error occurs, the printer will continue to complete the given set of instructions regardless. Because of this the printer will still output material on top of any fault or flawed part, and this will most likely be wasted and the part discarded.

The motivation for this project comes from my personal need for a system that can detect errors in the printing process. I started learning about 3D printers by building my own. I have used it to make prototypes, functional parts as well as decorative parts. It started out as a hobby but has proven valuable during my studies. I have many times experienced errors occurring during the process which renders the parts useless. Errors usually leads to waste of filament, time and energy.

The process of printing 3D parts is time-consuming. Printing parts can take up anything from 10min to several days depending on size and resolution of the print. This means that it is not practical for the user to monitor the whole process from start to finish. This is where the idea for this project comes from. If the process could be monitored by a system and detect if something went wrong, this could have an impact on the 3D printer for consumers market.

1.2 Problem formulation

Detecting errors in FDM 3D printing process, by using computer vision on live monitoring.

1.3 Project objectives

The main objective of this project is to develop a system that visually monitors the process of 3D printing to detect errors, and alerts if an error is detected.

1.4 Scope

The project is aimed for single colour consumer FDM 3D printers.

The system should be able to run on low-cost hardware like a microcontroller. The system shall have a user interface for configuring of the system, real-time monitoring and error detection.

The application should be made available as opensource.

1.5 Previous work

Straub (2015) writes a paper that considers the approach to this problem using a multi-camera setup. The image analysis used was a pixel-by-pixel differentiation with regards on the pixels RGB brightness values. In addition, a comparison of the final print to the in-progress was used. The result of this paper was a system able to detect filament runout (print not progressing). However, the paper mentions that utilizing greater colour filtering could add robustness in segmenting the object.

The second paper found, Vision based error detection for 3D printing processes (Baumann & Roller, 2016), has a slightly different approach. This paper defines five error classes, and they were able to detect three of them. They used HSV thresholding to segment the object from the background, and blob detection to track the object. In addition, they used visual markers to determine the area of interest for cropping the images. To detect errors, they also used a differentiation approach. They then calculated a differential image from three consecutive images, with a time interval of 3/25 seconds. Furthermore, they integrated the three consecutive frames to detect change in horizontal position of the object.
The paper identified obstacles in pixel errors from the camera and in change of lighting conditions.

The third paper found, In-line 3D print failure detection using computer vision (Lyngby, et al., 2017) builds on the approach of the first paper, but instead of comparing to the final print they sought to compare with a CAD model. They also used HSV thresholding to create a mask segmenting the object from the background, as used in the second paper. The result of the paper demonstrated that comparing the print to a rendered model of the object works in their rough prototype.

All three papers have done work towards solving the problem of detecting errors in 3D printing. Key aspects for these, that will be investigated in this thesis is, using HSV thresholding for segmentation, pixel-by-pixel differentiation and detect changes in horizontal position. In addition, this project will investigate detection of changes in vertical position.

1.6 Structure of the thesis

Chapter 1 gives an introduction and scope of the problem and presents key aspects of previous research on the problem.
Chapter 2 documents the theory behind the methods used in this project.
Chapter 3 contains documentation on the methods used to achieve the result.
Chapter 4 gives the result of the project
Chapter 5 Discusses the result and limitations of the project as well as the impact.
Chapter 6 Gives a conclusion of the project and discussion.

Chapter 2: Theory

2.1 3D Printing

3D printing is an additive manufacturing technology. It is the process of turning a digital model into a physical three-dimensional object. This is done by adding materials layer by layer to build an object (3D Hubs, u.d.). 3D printing can be done by many ways, however in this paper only FDM 3D printing is explored and developed for.

2.1.1 FDM

FDM 3D printing is rapidly growing on the consumer market. This technology was developed by Scott Cramp in 1988. It has become one of the most widely available processes of 3D printing and is mainly used for design verification and rapid prototyping (3D Hubs, u.d.).

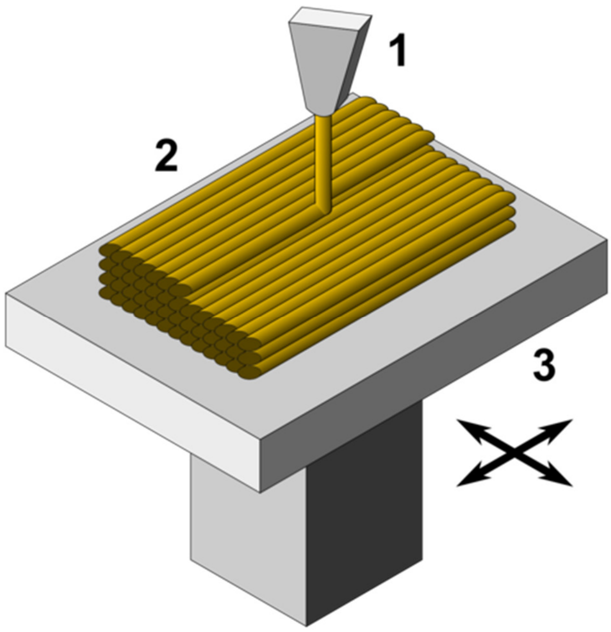


Figure 1 FDM printing diagram. From (Gringer, 2018)

Figure 1 shows the principal process of printing by FDM. The filament is melted and extruded through a nozzle in the print head (1). The print head is moved relative to the print bed (3), either by moving the print head, the print bed or a combination. The extruded filament is then cooled in the shape it has been laid down to form a layer (2). The process continues to create layers on top of each other to form a 3D model.

2.1.2 Errors

In consumer 3D printing errors may occur. Baumann et al. defines in their paper (2016) five error classes as there were no previous systematically research on error classification.

1. Detachment. This is an error due to the object detaching form the print-bed. This can happen if the distance between the print-bed and nozzle is not calibrated correctly, or due to vibration and rapid movement in the print bed.
   This error class includes warping, which is when the object is only partly detached. Warping is the result of fluctuating in temperature when the part is cooled. The mechanical properties of the filament determine how the material shrinks when cooled. If the part is cooled to quick or inconsistent warping can happen.

2. Missing material flow. This is an error due to the filament not extruding through the nozzle. This can happen if the nozzle is clogged, extruder not pushing the filament, or the filament has run out. When this happens, the printer continues the given instructions regardless, and the object will not grow.

3. Deformed object. This is an error when the object shape deviates from the CAD model. This can occur when printing bridges, with lower layer missing, between two points. This can also occur when the layer parameters are outside the edge of the previous parameter, as with steep overhang surfaces.

4. Surface errors. This is an error where the surface deviates from the CAD model. The surface texture is determined by the material property. This error can come from the object geometry not matching the chosen parameters, as with steep angles.

5. Deviation from model. This error is when none of the previously mentioned errors occurs, but the object deviates from the model in structure or size. This can occur without the object having any obvious flaws and the print is successful. Deviation can be the result of unsuitable slicing and printing parameters, leading to warping and shrinkage.

2.1.2.1 Human error

3D printing is subject to errors caused by the user. There are some key factors the user must be aware of when using a 3D printer.

- Object design. In order to print parts, the object should be designed with consideration of the 3D printing process. The process of adding materials layer-by-layer restricts the object shape and mechanical properties (Oropallo & Piegl, 2015).
- Slicing. The 3D printer needs slices of the model representing each layer. A slicing software is used to convert the digital model to slices. The software takes many parameters witch the user can adjust. Therefore, slicing can be a subject of human errors.

2.1.2.2 External influence

The process of 3D printing can be influenced by external factors.

- Environment. FDM printing is usually based on melting materials to extrude each layer. The layers are then cooled to form solid parts. The cooling process can be prone to external influence, which can result in wrong mechanical properties, warping and misplacement of the layer.
- Material properties. FDM printing can utilize many different materials. Some of these can be sensitive to moisture, temperature and cooling rate. Errors will most likely occur if the material properties are not considered.

2.2 Image analysis

The purpose of digital image analysis is to extract information from the images for data analysis (Easton, Jr, 2010). Image analysis is used in computer vision to make the computer able to process data from images.

2.2.1 Gaussian blur

Gaussian blur is a method used for smoothing images. This is a filter used to reduce noise, usually in pre-processing of images. This is done by convolving the image with a kernel. The amount of smoothing is determined by the size of the kernel.

2.2.2 Colour-space and conversion

Digital colour images are usually represented by RGB colour space as this is the way screens displays images. HSV colour space is a different way of representing colours in a human related way.

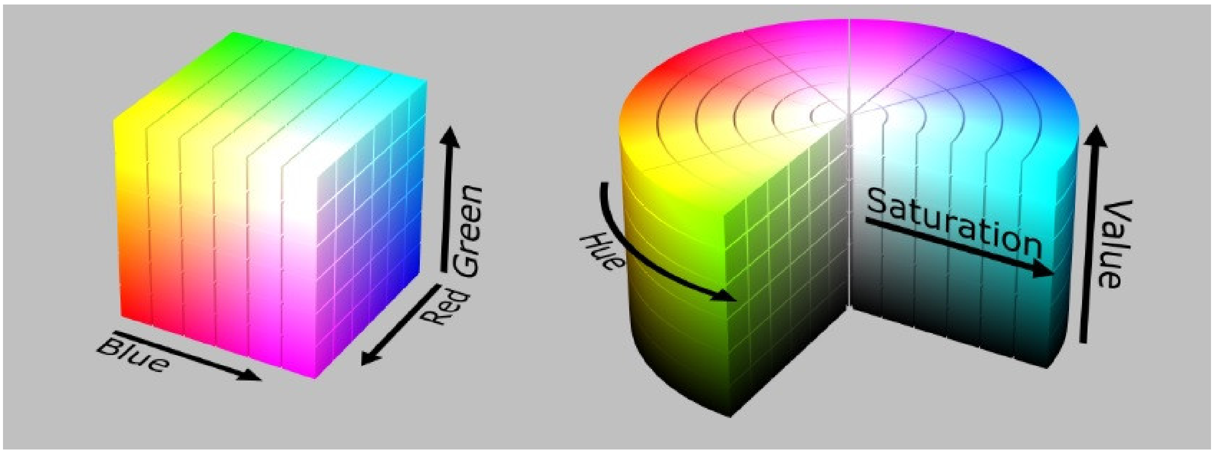


Figure 2 RGB and HSV colour space. From (SharkD, 2008)

Colour spaces can be represented as cubes shown in Figure 2. The images gathered by using OpenCV is defined in BGR colour space, which is the same as RGB but rearranged.

In BGR space each pixel is represented by a blue (B), green (G) and red (R), and in HSV space each pixel is represented by hue (H), saturation (S) and value (V) (OpenCV, 2019). The hue represents the colour. Saturation is a representation of the amount of white in the colour. Value is the brightness of the colour.
When threshold an image by colour range it is more intuitive for the user to define a range in the HSV colour space rather than in RGB values (Umbaugh, 2018, pp. 48, 306).

2.2.3 Thresholding

Thresholding is a way of creating a binary image from a colour, or grey-scale image by segmentation. An image can be mapped to a binary image by setting a threshold range for the colour values. The Values within the threshold range will be set to 1, and the values outside the range will be set to 0. This gives a binary image representing a mask of the threshold (Umbaugh, 2018, p. 93).

2.2.4 Morphological Operations

Morphological filtering is used in image analysis to segment objects of interest. This is done by two operations, dilation and erosion. Dilation is a way of expanding the object to fill holes. Erosion, the opposite, is a way of shrinking the objects to remove noise artefact. They can be used in combination to restore the object after filtering. This is called opening and closing. Opening is erosion followed by dilation, and closing is dilation followed by erosion (Umbaugh, 2018, p. 185).

2.2.5 Contour

A contour is the surrounding boundary of joining points that have similar properties. Contours are useful for getting information on the shape of objects (Sinha, u.d.).

2.2.6 Image comparing

Image comparing is the process of differencing images.

2.2.6.1 Pixel by pixel subtraction

Image differentiation by using pixel by pixel subtraction determines a difference image by subtracting each pixel in one image from the pixel of the same placement in another image. The resulting image contains the difference between the images. This method requires the images to be of the same size. This method is often used to detect motion in consecutive images (Umbaugh, 2018). One weakness to this method is if the camera moves between the images, this would indicate a difference.

2.2.6.2 Template matching

Template matching is the process of searching for an image (template) in another image. This is usually used to find smaller images within a larger image. The result of template matching two images is a probability measure for the template is within the image.

2.2.6.3 Object tracking

The process of object tracking is finding the position of key points of the object and detect change over time.

The printed layers should not move in the horizontal axis. By tracking the position of the outward edge across consecutive images it is possible to detect if the object has moved horizontally.

The printed part should increase in vertical axis for each layer printed. By tracking the position of the top edge of the part, it is possible to detect if the part grows or not.

2.3 Lighting

The most common way of lighting shiny surfaces, in order to reduce specular reflections, is by diffuse lighting. Diffusing the light is a technique for scattering the light in multiple directions and spreading the source of light across a larger surface.

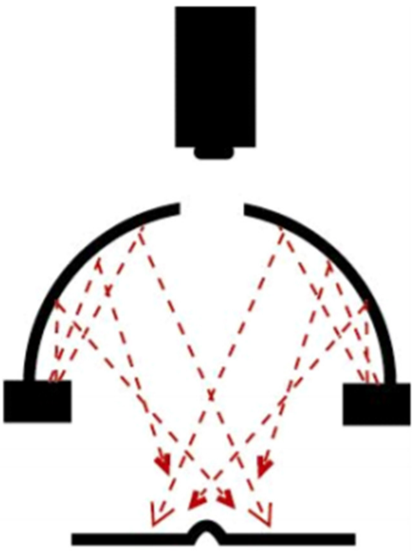


Figure 3 Dome diffuse lighting. From (Martin, 2013)

Dome diffuse, shown in Figure 3, is one of the most common ways of doing this and are very effective for lighting curved surfaces (Martin, 2013). This gives a smooth light evenly scattered across the scene.

2.4 Principles of Programming

2.4.1 OOP

Object oriented programming (OOP) is a programming paradigm. In the book “Objects First with Java” David J. Barnes and Michael Kölling states that OOP is used to model some part of the world (Barnes & Kölling, 2017). This book was part of the curriculum for the subject

Object-oriented programming (ID101912). This subject is the reason for using OOP in this project. OOP principles focuses on how to write good code.

2.4.2 Facade pattern design

A design pattern is used to structure large applications and defines how objects should interact in complex relationship (Barnes & Kölling, 2017).
The facade pattern is used to create a higher-level interface of the functionality of subsystems (Bosch, 1996). This makes implementation to user interfaces more general and provides a single integration

Chapter 3: Method

3.1 Preparations

The purpose of this project is to make a system that can monitor and detect error on the 3D printing process. This requires the system to run on hardware reserved for this purpose. Since this project focuses on consumer FDM printers this hardware had to be consumer friendly, and therefore the Raspberry microcontroller was chosen. The Raspberry environment is vastly supported online with many tutorials and good documentation free and open to the public.

3.2 Hardware and software

Hardware		
Name	Info	Usage
Ultimaker 2+	FDM 3D Printer	Printing parts for testing system
Raspberry Pi 4	Microcontroller	Main system
Pi Camera Module V2	Raspberry Pi Camera	Gathering of images for the system
LED strips	Consumer type all white LED	Lighting the parts for printing
LCD	Monitor	Main system display
Logitech C920	USB Camera	Testing and development on PC
Raspberry Pi 3b	Microcontroller	3D printer Remote control system
End stop button	Switch	Trigger to save frame from camera
TP-LINK	TL-WR841N	Deployment network for both Rpi

Table 1 Hardware used in the project

Software		
Name	Info	Usage
Python 3	V 3.5	Main system framework language
Pycharm	V 2019.2	IDE Development environment
Github	Git version control	Storing versions of the applications
OpenCV	Contrib V4.1.0.25	Computer vision library for image processing

NumPy	V 1.17.2	Scientific library for arithmetic operations
Tkinter	GUI library	Frontend GUI
PIL	V 6.2.0	Python Imaging Library for displaying images in GUI
Json	JavaScript Object Notation	Read/Write preset values from/to file
Rasbian	Raspbian Buster with desktop V September 2019	Operating system for main microcontroller
OctoPi	V 0.16.0	Operating system for remote microcontroller
OctoPrint	V 1.3.12	Running on octopi for controlling the 3D Printer
Octolapse	V 0.3.4	Octoprint plugin for controlling timelapse.
Autodesk Fusion 360	CAD/CAM tool	3D modelling parts
Ultimaker Cura	V 4.4	Slicing modelled parts for printing
MobaXterm	Home Edition V12.4	SSH client for remote terminal for the microcontrollers

Table 2 Software used in the project

3.3 Development environment

3.3.1 3D Printer

This project uses an Ultimaker 2+ for research and development of the system. It was also used to make physical parts needed in this project. The IIR institute at NTNU Ålesund had recently bought 4 of these, and this made the choose of printer easy as it could be reserved for this project entirely.

Figure 4 Ultimaker 2+

The Ultimaker, shown in Figure 4 is an FDM printer that uses a single extrusion nozzle to build 3D parts layer by layer. It is a robust consumer printer and delivers consistent results. However, it can be subject to errors due to environment influence and human errors from wrong usage.

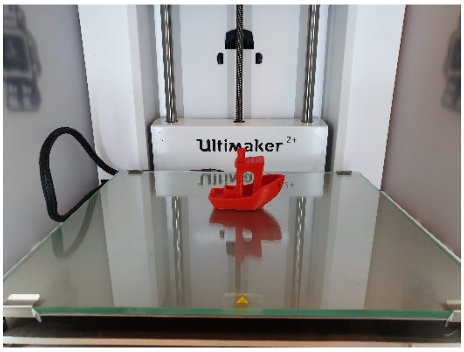


Figure 5 Unmasked print bed

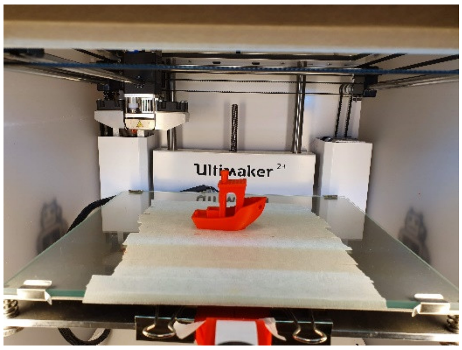


Figure 6 Masked print bed

Figure 5 shows that the print bed of the Ultimaker is made of reflective glass. This gave a reflection of the object that interfered in segmenting the object from the surroundings. Because of this some tape was applied to the print bed, as shown in Figure 6.

3.3.2 Microcontroller and camera

The development of this project started out using a Raspberry Pi 3B+. After initial work using multiple images and processing them in real time it was clear that this model had limitations in memory when processing large number of images. The model 3B+ has 1Gb of RAM. Therefore, a Raspberry model 4, which has 4Gb of RAM was acquired to this project. The Raspberry was setup with Raspbian Buster operating system.

The Raspberry Pi ecosystem contains an original camera module. This camera connects to the raspberry Pi with a ribbon cable. The camera was a natural choice due to the built-in library support, and the low cost of this module.

3.3.3 Camera stand

In order to get consistent images, the camera needed to be mounted relative to the print area.

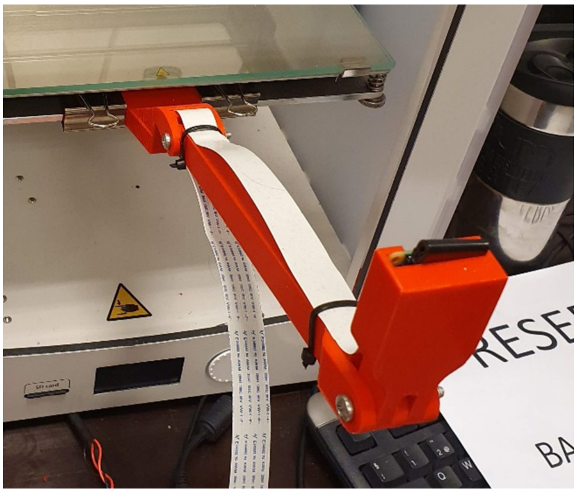


Figure 7 Camera mount attached to printer

A camera stand was made in order to mount the camera in good viewing distance to the printer bed, shown in Figure 7. The camera is then relatively stationary to the parts that will be printed.

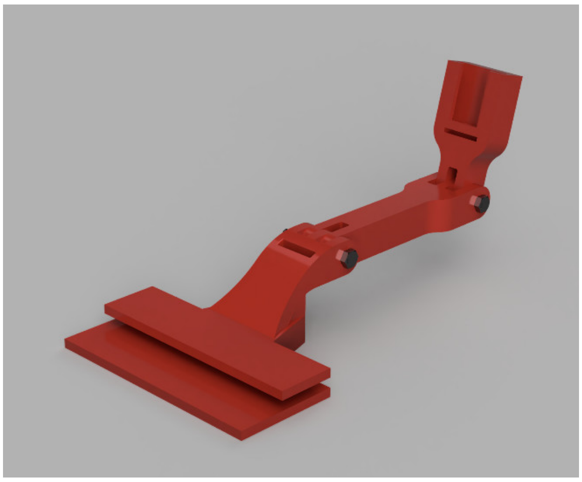


Figure 8 CAD render of camera mount

Figure 8 shows the parts that were designed in Autodesk Fusion 360. These were uploaded to Thingiverse #4010873 (Paulsen, 2019).

3.3.4 Lighting

The printer used in this project has some lighting of the print area. However, they were considered too dim for this project, and because of the exposed LED with no diffusion they gave a harsh light.

3D printed parts are usually made from highly reflective materials. Also, when working with computer vision in situations like this project, a fixed lighting is useful.

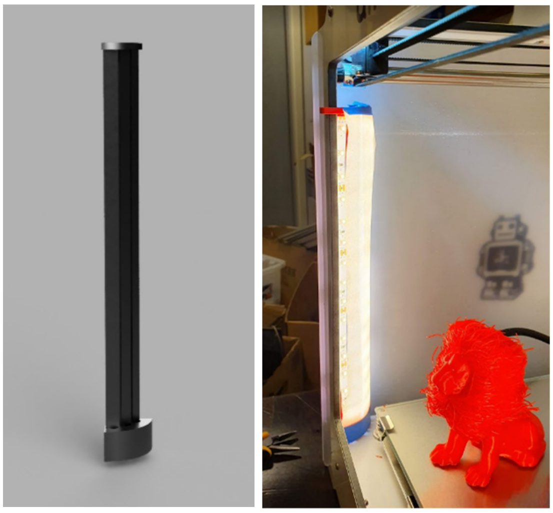


Figure 9 CAD model and mounted lighting fixture

This led to the making of two LED columns, shown in Figure 9. The parts were designed in Autodesk Fusion 360, printed on the Ultimaker and fitted with LED strips. They were placed in the front corners of the printer and were fitted with wax paper to diffuse the light. This gives a soft and consistent light that will minimize reflections.

3.3.5 Printer control

In order to gather consistent images where the printer nozzle is not covering the part, the printer needed to be controlled. This was done using a software called Octoprint, which runs on a raspberry pi. Octoprint overrides the Ultimakers own control and enables control in a web interface for remote access. The raspberry pi 3B+ initially intended for the system was reused for this.

Within this software there is a plugin called Octolapse. This is a plugin used for creating smooth timelapse video of the print, where the print head is moved to a corner for every image gathered. In other words, this plugin lets the user control what the printer does before starting the next layer of the print. This led to the opportunity of snapping images by having the print head push a physical switch.

3.3.6 Trigger button

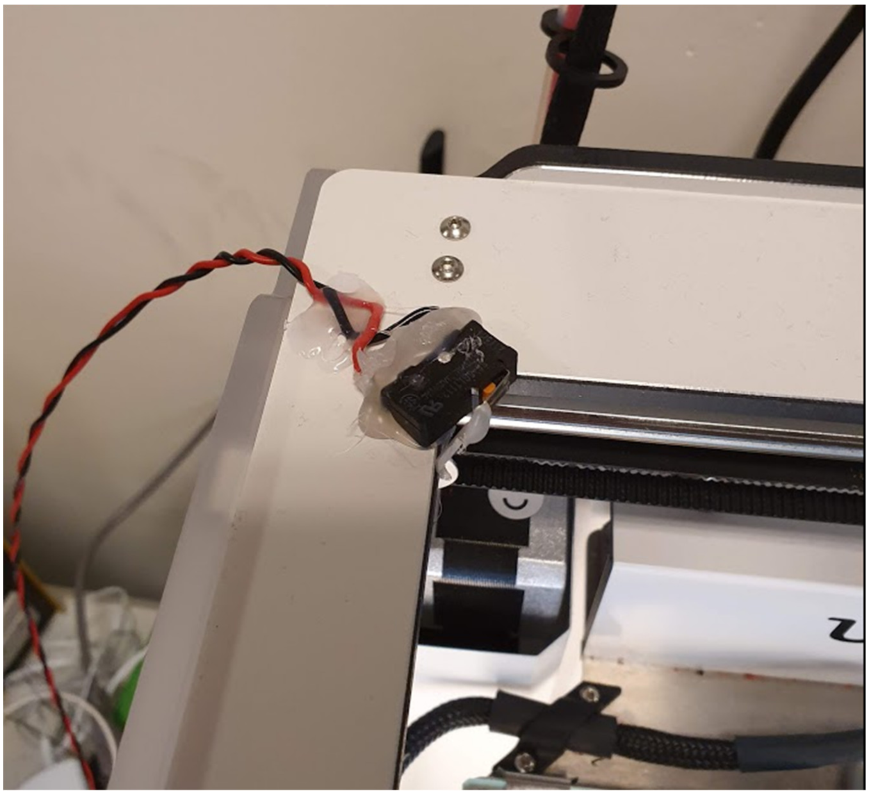


Figure 10 Trigger button mounted at printer corner

The physical switch, shown in Figure 10, is used to trigger the application to store a frame to the image register. The switch is mounted at a corner of the 3D printer, in such a way that the printer head can push it.

Initially the switch was connected by a pull-up method that uses the internal GPIO pins resistor of 1.8KΩ. The GPIO pin is connected to ground when the switch is closed. This was combined with a simple debounce and rising flag in software to detect a push to the switch. However, this solution was not as robust as needed. The switch triggered a few times without being pushed, when the print head was in the middle of the bed. It was theorized that this was due to electrical noise as the system is setup in the electro lab, or internal noise in the raspberry due to low pull up resistor. However, this interference was considered too noisy for this project.

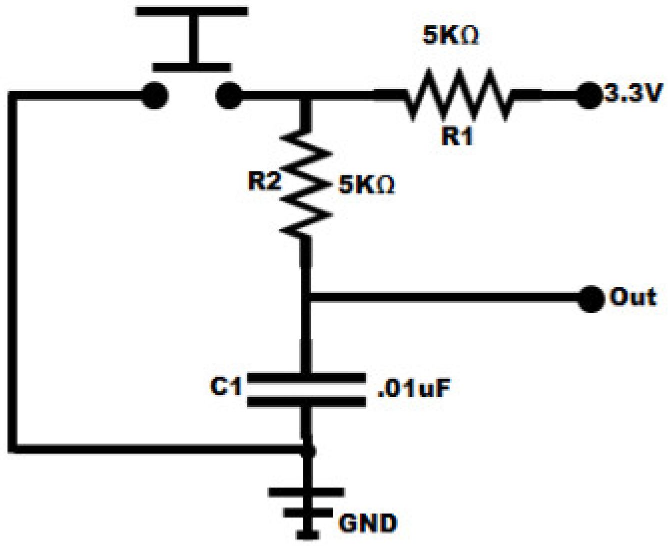


Figure 11 RC circuit for button. From (Hale, 2014)

This led to the decision of using a small RC filter circuit, shown in Figure 11. This solution is inspired by the project “Debouncing GPIO Input” by Ted B Hale (2014). This hardware solution is much more robust as it will shield against noise and debounce voltage spikes.

3.3.7 Deployment network

A network router for hosting a subnet was setup. This served the purpose of connecting both microcontrollers to a pc. The pc could then remote access the Rpi 4 over SSH, and upload and deploy the python files directly from the PyCharm IDE.

The network was also used to access the web interface of the Octoprint server, for remote controlling the printer.

3.4 Software development

3.4.1 Initial preparations

My previous experience in software development was limited to the curriculum for the subject Object-oriented programming (ID101912). In this subject Java was used, but the knowledge gained in OOP was general and not limited to Java.

Previous experience with computer vision and image analysis was done using Matlab. However, Matlab is specialized for data analysis and is not as suitable as other languages for developing on the Rpi.

The Raspberry environment supports both Java and Python programming languages, but due to the many available open source tutorials and native Raspberry GPIO support, Python was chosen. Python is free and open source and supports vast number of libraries.
The choice of using python to develop this application meant that the project had to spend a lot of time going over tutorials and trying out methods, in order to gain sufficient knowledge before building the application architecture.
The development started with gathering knowledge on image analysis using python. This was done using many online tutorials, especially Adrian Rosebrock`s community (2019). This site had several tutorials and articles for getting started with computer vision and OpenCV, and it was done using examples in python.

3.4.2 Virtual environment

A virtual python environment was created in the Rpi, to isolate the project from the OS. This was done to control the packages and resources in isolation. This meant that the libraries and packages used could not be altered outside of the environment.

In order to run the project from the Raspbian terminal, the virtual environment needs to be activated, then navigate to the project directory, before lastly running the python file. This was an unnecessary tedious process. Therefore, a bash script for automating this sequence of commands was made. This made launching the python files from the virtual environment much quicker.

Pycharm support remote deployment over SSH. This allowed the project to be developed on a pc before uploading to and running on the virtual python environment on the Rpi. However, running over SSH meant that any GUI or preview element could not be displayed. Therefore, this was only used to upload the files from the pc to the Rpi.

3.4.3 Testing and learning

The work continued with developing and testing small python scripts to deploy on Rpi. Like how to handle the Rpi camera and GPIO, capture frames to memory and writing image files, building time-lapse video from consecutive frames, colour thresholding and object segmentation. All these where small programs, preforming each task individually without the use of OOP. A sample of these can be found at Github (Paulsen, 2019). At this point the development had to start working towards a complete system implementing these.

A meeting with Arne Styve and Girts Strazdins was held regarding planning the software architecture. They gave good advice and recommended utilizing a facade design pattern to connect multiple subsystems, provide a simple interface to the client by hiding the complexity of the backend systems, and maintain responsibility-driven design. They also advised in developing by using a top-down approach, by identifying and defining what the system should provide to the user, not the other way around.

3.4.4 System initial spec

After the meeting with Arne and Girts, a minimum spec for the system was made, with regards to the user functionality. The initial spec:
The system shall have:

- GUI
- Live display of the camera feed
- Compare sequential images at given interval to detect changes
- Detect errors and alert if detected
- Configuration of HSV parameters for thresholding

3.4.5 Development

Developing with a top-down approach led to taking a step back from testing all small programs individually, and towards figuring out what the system should do for the user. This started the work on defining a user experience (UX).
UX design goals:

- User input of parameters should be intuitive.
- The system shall give the user information regarding status of the print.
- Displaying live camera to the user

3.4.6 GUI

In order to build the graphical user interface a library was needed. For this project the Tkinter package was chosen due to it being built-in to Python (Python Software Foundation, 2019) and is one of the most commonly used for creating GUI apps in python. The tkinter package builds on top of Tcl/Tk gui toolkit (Ross, 2019).

The GUI was developed with regards to the system spec and UX design goals. The main window holds buttons and sliders for the user to input the necessary values for the image processing. A window for displaying the live camera feed that constantly updates with the latest frame from the camera was implemented. In addition, two pages for displaying the latest and the second latest image from the image register was added.

3.4.7 Multithreading

In order to let the system preform multiple things at the same time, using multiple threads was needed. The camera instance needed a thread for continuous capturing frames. The trigger instance uses a thread for capturing a push of the button. The processor instance uses a thread for preforming image processing. And the GUI uses a thread to handle button events and refreshing frames displayed.

3.4.8 System spec revised

After some time in development the system spec was changed towards more specified demands.

The system shall have:

- Intuitive GUI for development and testing
- A window for displaying live camera feed
- Windows for displaying the last and second to last page stored in register
- Store images to a register when a physical button is pressed
- Configuration of HSV parameters for thresholding
- Save and store HSV parameters presets
- Buttons for applying morphological operations to the mask
- Detect errors and alert if detected

3.4.9 Software architecture

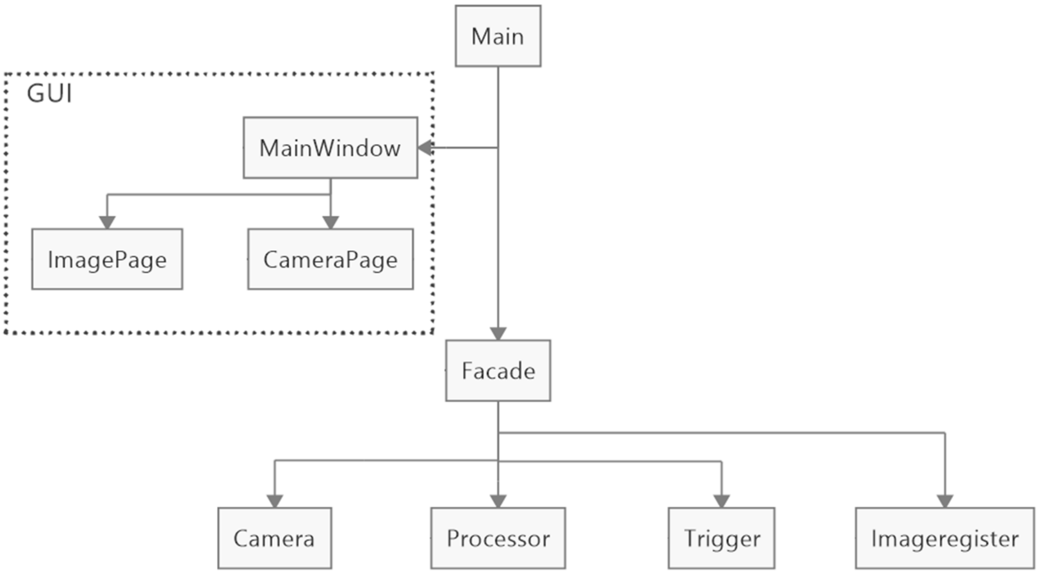


Figure 12 Class diagram

Figure 12 shows the class diagram of the application.

- Main. This is the entry point for starting the application. This class instantiates a MainWindow and a Facade object.
- MainWindow. This is the main window of the GUI. This instance handles the GUI elements for user input. This class instantiates the CameraPage and two versions of ImagePage on button press.
- CameraPage. This class displays and updates the current frame from the camera.
- ImagePage. This class displays and updates a frame from the Imageregister. One instance of this displays the latest image. Another instance of this displays the second to last image.
- Facade. This class connects functions of the subsystems (backend) to the GUI system (frontend).
- Camera. This class handles the pulling of frames from a physical camera.
- Processor. This class handles the image processing, and logic connected to data collecting. Currently this class also handles alerts.
- Trigger. This class listens for a push of a physical GPIO button.

- Imageregister. This class holds the images in a register.

3.4.10 Finalizing and clean-up

Towards the end of the project the code needed to be cleaned up. Methods that was not used was removed. Encapsulating the objects variables in private fields and utilizing getters- and setters- methods for these was implemented on every class of the code. Comments for the methods that was not self-explanatory, and where it was necessary to clarify.

3.5 Image processing

In order to detect errors by image processing multiple approaches was tested.

One approach was to use template matching. This used the second to last frame gathered as a template for matching in the current frame gathered. The images used in this approach was BGR. Tests conducted during development showed that the values gathered from this was inconsistent. This meant that the values could not be tested against a threshold value in order to detect errors. Because of this the template matching approach was abandoned.

Another approach was to use pixel-wise differentiation. This was done by subtracting the previous image from the current and calculate the difference in pixels. This gave a value of how much area of the frame had changed. This approach showed good results in testing against a threshold, when used on binary images.

The last approach explored was to use contouring to track extreme points (outer most point) of the object. The contours where gathered from binary images. This showed a much more reliable result in tests during development and was therefore used in the final algorithm.

The algorithms used in the final system are:

1. Gather image
2. Smoothing image
3. Convert colour space to HSV
4. Thresholding to binary
5. Morphological Operations
6. Contour and extreme points

7. Store binary image in register
8. If register har two or more images: Comparing consecutive images, pixel-wise and by extreme points position
9. Go to step 1 at next button press

3.5.1 OpenCV

OpenCV is an open source library of image analysis algorithms used for computer vision applications. This library was chosen for this project because it is open source and vastly supported online with tutorials and good documentations (OpenCV, 2019). The library covers most of the image analysis used in this project.

3.5.2 Gather images

Images are gathered in Camera instance. If Rpi is used the picamera modules capture_continuous method is used, while if PC is used the opencv modules VideoCapture is used. Both is handled by the Camera instance with returns the latest frame.

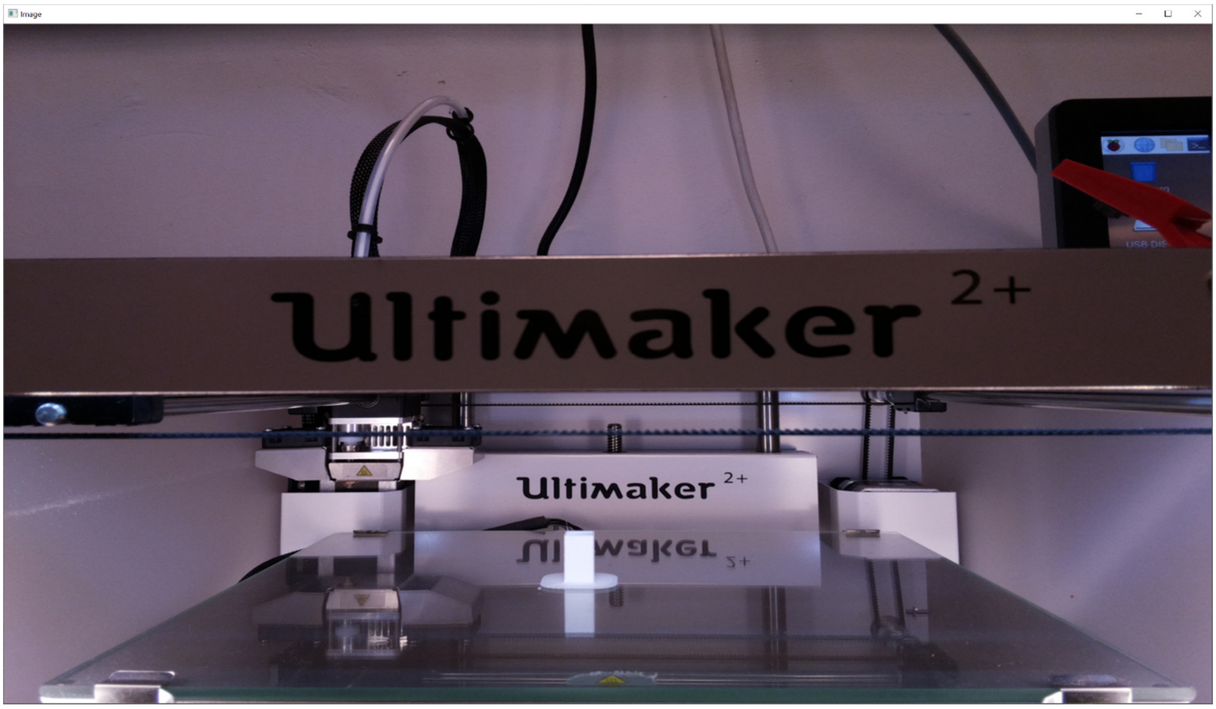


Figure 13 Image captured on picamera

Figure 13 displays and image captured by the picamera. Here the captured FoV of the print area is shown. The camera was placed in this manner to cover the whole print area.
Images is captured after each layer the printer has laid down. The images are then stored to a register in order to hold several images in memory.

3.5.2.1 Resolution

The Raspberry Pi Camera module supports capturing at multiple resolutions. To determine the physical resolution, images of a ruler was captured at different resolutions.

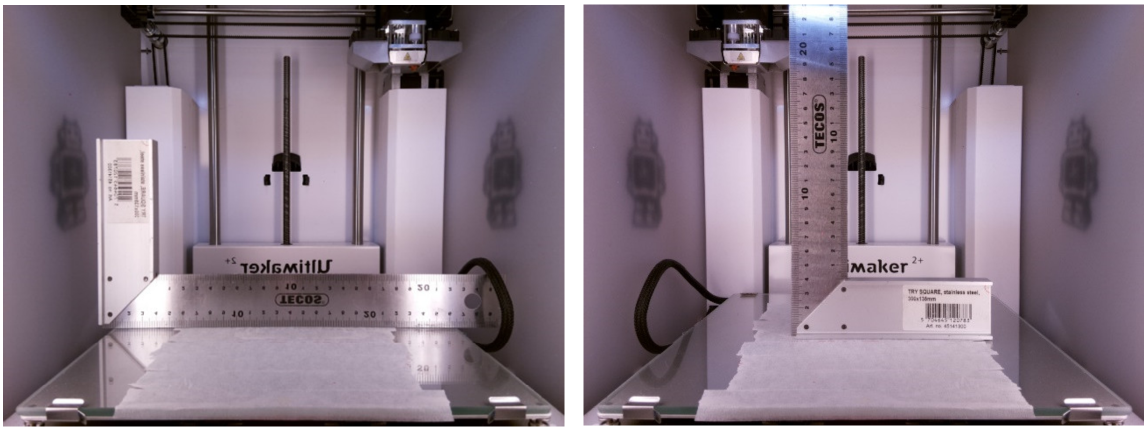


Figure 14 Image of ruler used to determine physical resolution

The images shown in Figure 14 was used to count the number of pixels per millimetre. This was done by counting pixels over 10/20 millimetre and average then to get relative accurate readings. The maximum resolution of the Rpi camera when capturing continually is limited to 2592x1944. In addition, resolutions of 1280x960 and 640x480 was measured.

Resolution	Vertical pixels per millimetre	Horizontal pixels per millimetre
640x480	1.5	1.5
1280x960	3	3.1
2592x1944	6.2	6.3

Table 3 Resolution accuracy

Table 3 show the difference in accuracy of the different resolutions. This measurement was taken at the centre of the printers build plate. In addition, images where the ruler is positioned at the back and front of the build plate was gathered. By measuring at the middle and back, the difference at 1280x960 resolution was 0,7 pixels per millimetre in vertical orientation. The resolution of 1280x960 yielded enough framerate to display live camera feed and maintain good accuracy, therefor this was used throughout the project.

3.5.3 Smoothing image

In order to remove noise from the gathered frame, the Processor instance smooths the image by using OpenCV's GaussianBlur. This method convolves a 5x5 kernel with the image to blur and smooth the image.

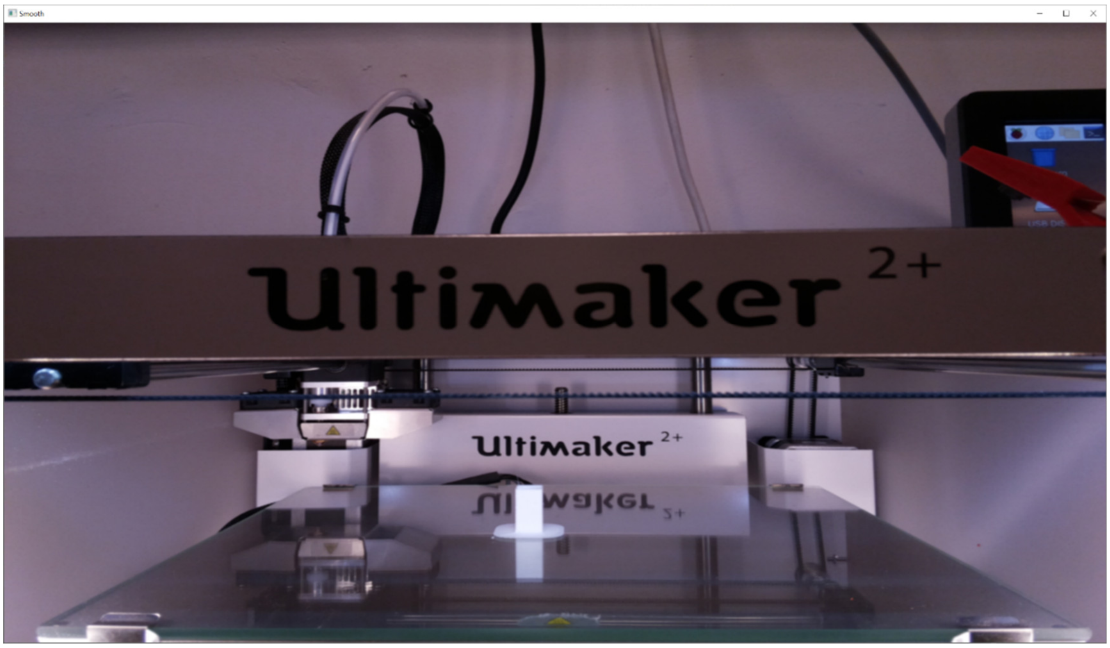


Figure 15 Smoothed image

The edges of the image get softer as shown in Figure 15. This is used to prepare the image for further processing.

3.5.4 Converting to HSV

The images are converted to HSV using OpenCV's cvtColor method. This maps the BGR values in the gathered frame to HSV.

3.5.5 Thresholding

The HSV images are converted to binary images by thresholding on user input HSV range using OpenCV's inRange method.

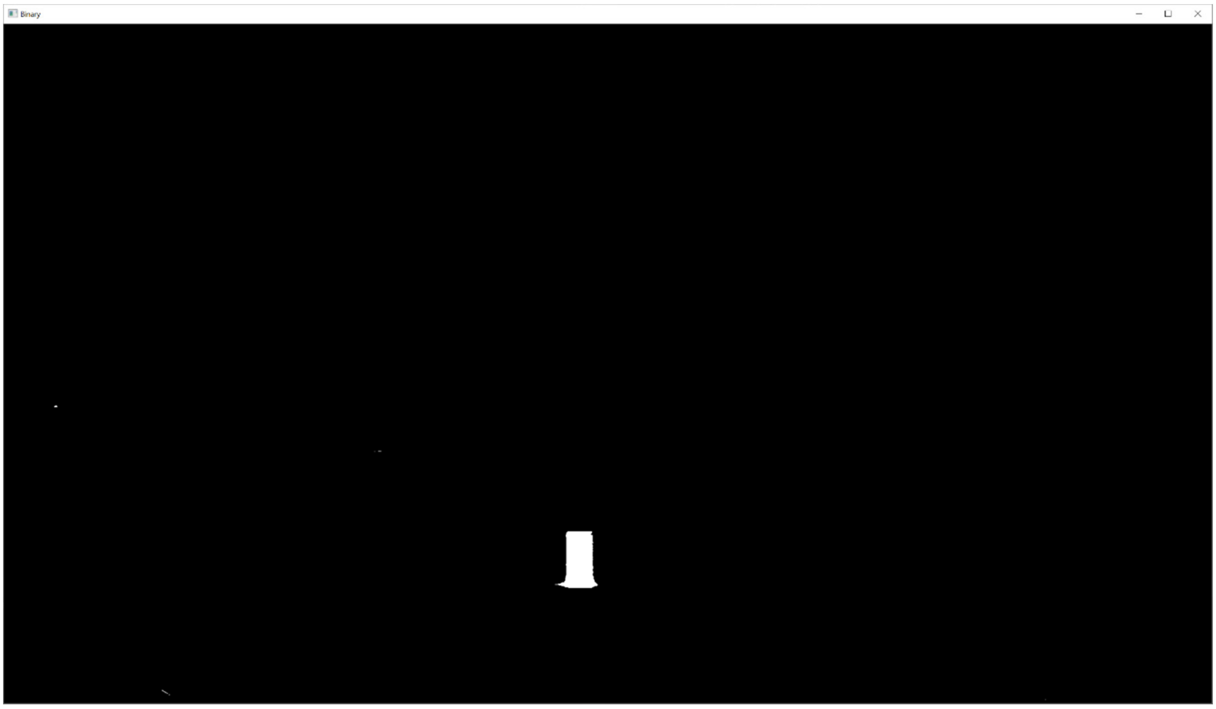


Figure 16 Binary image from the thresholding

The image is thresholded to mask the object from the background. Figure 16 show the binary mask of the segmented object, with the same size and scale as the input image shown in Figure 15.

A handling of three preset HSV value ranges was implemented in the GUI. The user can save the values for later usage in the presets.

3.5.6 Morphological Operations

The binary image is then processed by morphological operations if the used requests it. The user can then choose between erode, dilate, open and close. All these operations are used to clean the binary mask, in order to better separate the object from the environment.

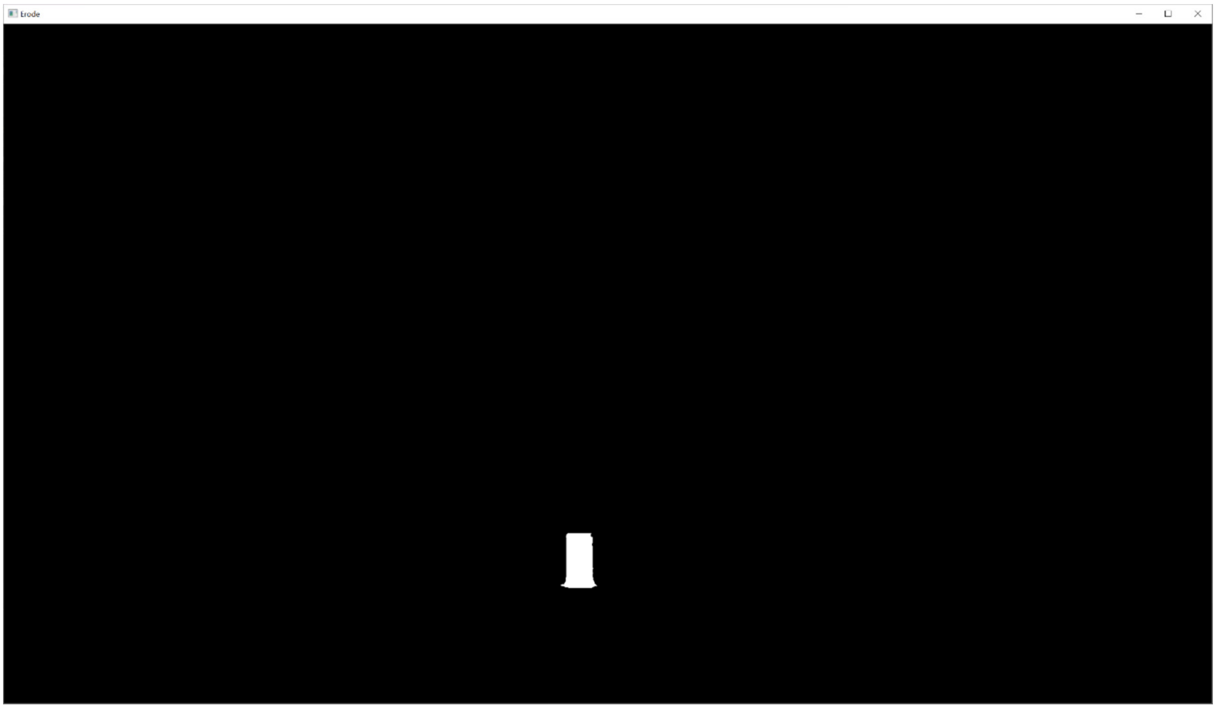


Figure 17 Binary mask after eroding

The image produced by thresholding, Figure 16, can contain some grainy noise that needs to be removed by morphological operations. By eroding the image, it is possible to remove these grains, as shown in Figure 17.

3.5.7 Contour and extreme points

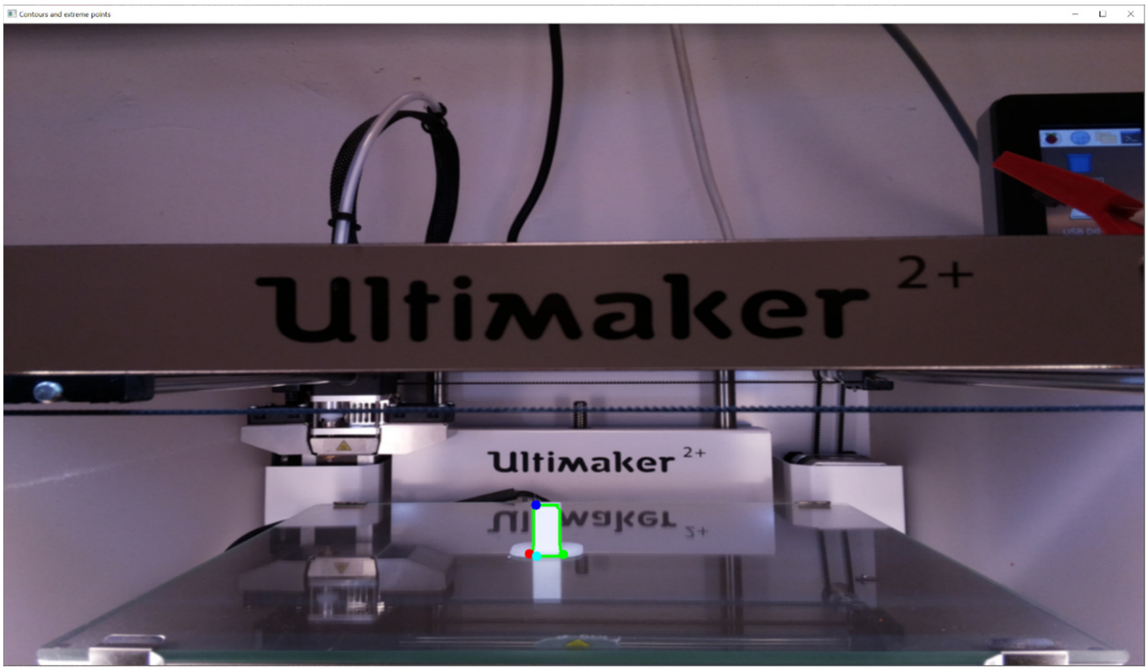


Figure 18 Contour and extreme points of the segmented object

By finding the contour of the object it is possible to find the extreme points of the boundary. These extreme points, shown in Figure 18 are used to track vertical position (highest point of the object) and horizontal position of the object. This is used when comparing images.

3.5.8 Comparing images

The current image is compared to a previously stored image from the register. The comparing explored in this project are:

- Pixel-wise differentiation. The current image is subtracted from the previous image, and the non-zero pixels of the resulting image is counted. This gives a differential value correlated to the motion between the two images. The method used are arithmetic operations using the NumPy module.
- Horizontal motion. By tracking the position of the extreme points on the left and right side of the object in one frame and compare to the same tracked point in the next frame. This gives a value representing the difference in position of the sideways movement of the object.
- Vertical motion. This is similar to the horizontal motion, but the tracked points are the top extreme points. This gives a value representing the difference in height of the object.

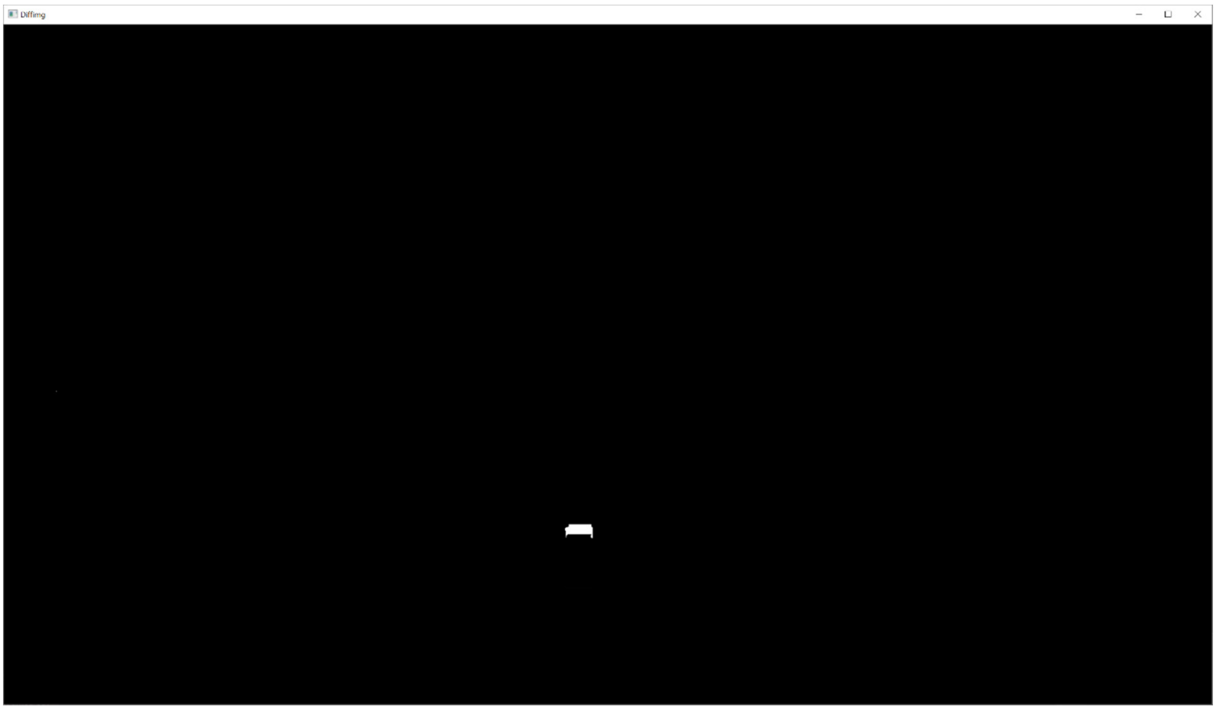


Figure 19 Differential image from pixel-wise subtraction on consecutive images

By comparing consecutive images of 640x480 resolution the difference of one layer was hardly noticeable by any of the methods, as this would be around 1.5 pixel. By using a resolution of 1280x960, one layer would be around 3 pixels. This led to a much more usable result in pixel-wise comparison and horizontal motion. However, the vertical motion still needed improvement as the value sometimes showed zero increase in height.
This led to the implementation of differentiating the current frame and the fourth last frame in the register, for detecting vertical motion. This gave a higher outcome that proved useful for the alert handling.

3.6 Alert handling

To handle alerts the values gathered from comparing images is compared to a threshold. If the values range outside the threshold, and alert is detected. During the development this sometimes triggered when an error was not present. And there by giving false alerts. This led to the implementation of a 2-stage error authentication.
If an error is detected in only one comparison and not the next the alert is not triggered. This is done by two errors flags that trigger the alert only if the system detects two consecutive errors.

The implemented threshold value for pixel-wise comparison can be input from the GUI. This was done because the value of pixel-wise comparison will wary according to the size of the object printed. By default, this threshold is set high to avoid false alerts, but the user can set this more appropriate by observing the values ranging during printing.

The threshold range for horizontal motion was set to +/- 3 in order to allow camera shake or otherwise interference.

The threshold for height difference was set to 0.

3.7 System tests

The system was constantly tested during the development. Parts where test-printed in three different colours, red, black and white filament. The black and white filament where challenging to threshold in order to get clean masks of the object, therefore they were put a side and red filament was used for the rest of the project.

Towards the end of the project a series of tests was conducted to document the result of the project. These tests consisted of monitoring several prints where the print completed without errors, where the print shifted in horizontal position and where the printer run out of filament during the print.

3.7.1 Prints

Three different 3D prints where tested in three different setups. The first setup was an unaltered environment, in order to print without errors and there by check if the system would trigger false alerts. The second setup was to retract the filament in the middle of the print, to test if the system detected filament runout. The third setup was to affect the print to no longer stick to the print bed, to test if the system detects horizontal shift.

3.7.2 Data

The data gathered from these tests:

- Timestamp
- Number of non-zeros pixels in differential image.
- Object height difference over last three layers
- Sideways movement of the object boundaries
- Status message

The data is written to a CSV file, in order to display and analyse in Microsoft Excel.

Chapter 4: Results

The results presented here are from tests conducted on the development environment of the project and have not been tested on different setup and other printers.

4.1 Application

This project has resulted in an application for detecting errors when monitoring the 3D printing process, which meets the spec mentioned in 3.4.8 System spec revised. The applications architecture allows for further development and can serve as a platform for future image analysis applications on both pc and Rpi.

The complete code for this system is uploaded to Github (Paulsen, 2019).

4.1.1 GUI

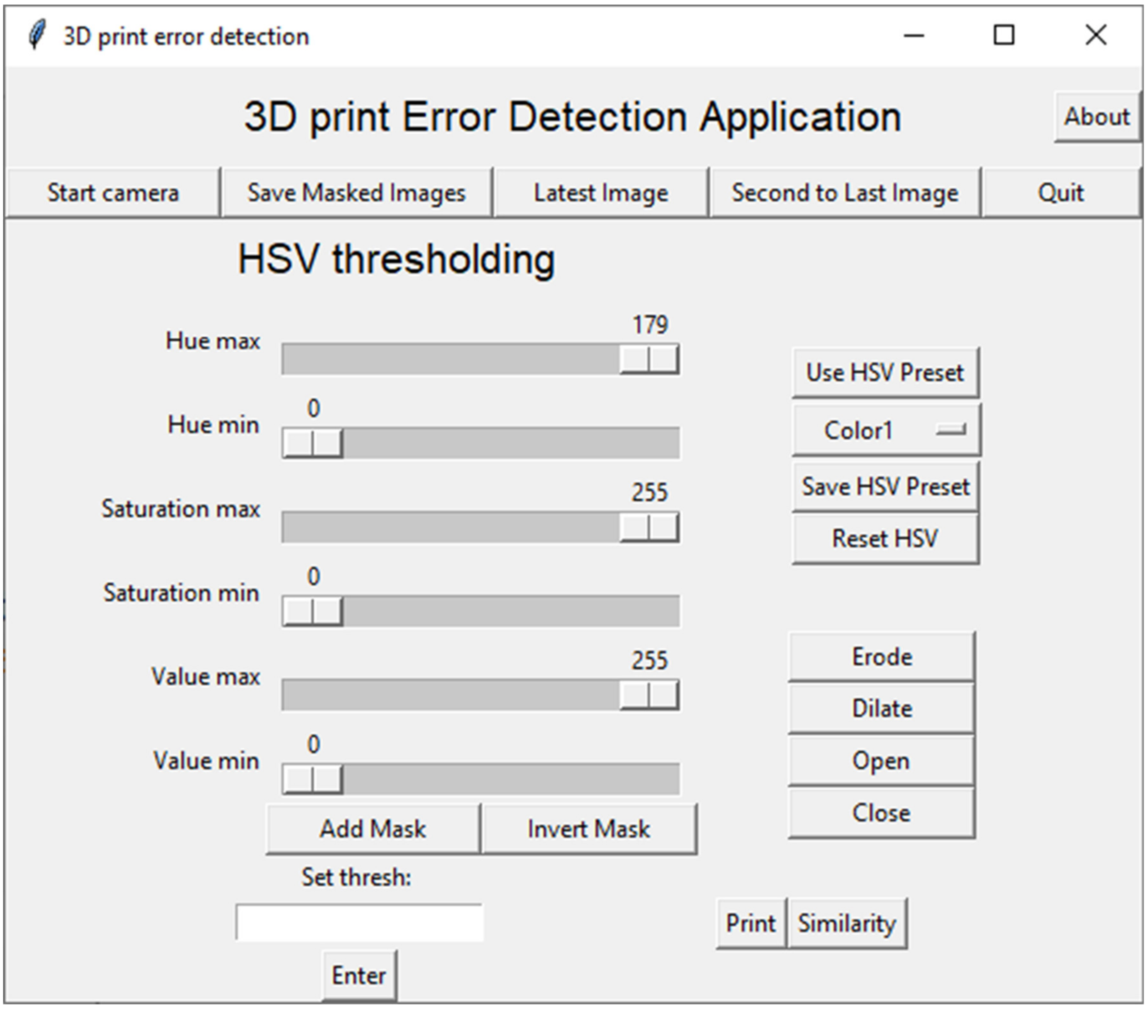


Figure 20 GUI application

The GUI, as shown in Figure 20, holds the necessary function the user needs to setup error monitoring.

The GUI also holds function needed in development. The “Print” and “Similarity” buttons is used to display numbers of frames in the register and display differences between the last frames.

4.1.2 Usage

The system assumes that the user monitors the first few layers put down, as errors regarding first layer bonding is not covered by the system.
In order to start the error detection, the first step is to start the camera page. This opens another GUI window containing live camera feed. The next step is adding the mask and set the threshold on the sliders. To find the correct threshold it is useful to put an object of the same colour as the filament the printer uses. If needed the next step is adding morphological operations to clean the binary image. The GUI holds buttons for eroding, dilating, opening and closing. The last step is to set the application to save masked images, before starting the print. Now the application waits for the physical button to be triggered, before storing images in the register. When the register has two or more images the pixel-wise differentiation and horizontal tracking starts. When the register has four images contain an object, the height differentiation starts.
Alerts registered is printed out in the console, as well as warnings, info and debug messages. In addition, the data tracked is logged to a csv file.

4.2 Printable hardware

The project has developed a camera mount to attach Rpi camera to an Ultimaker printer. In addition, a lighting fixture for LED strips vas developed. The parts can be found at Thingiverse #4010873 (Paulsen, 2019).

4.3 System tests

Data legend:

- Height diff. The number of vertical pixels over the last three frames
- Left diff. The variation between each frame, of the left extreme point tracked.
- Right diff. The variation between each frame, of the right extreme point tracked.

- Diff Normalized. The number of pixel difference between each frame. This number is represented as pixels * 1/100

The result of the system tests described in 3.7 System tests:

4.3.1 Error free prints

These tests were conducted without interference to the print.

4.3.1.1 Fast Cube

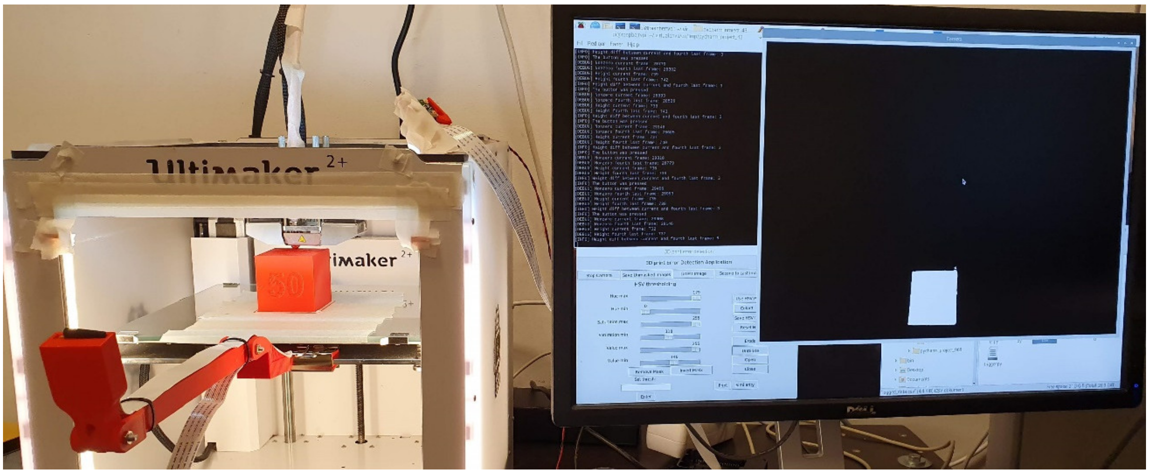


Figure 21 Fast Cube printed and system display

The object is a cube of 50x50x50mm printed complete without interference. Figure 21 Shows the setup of the test.

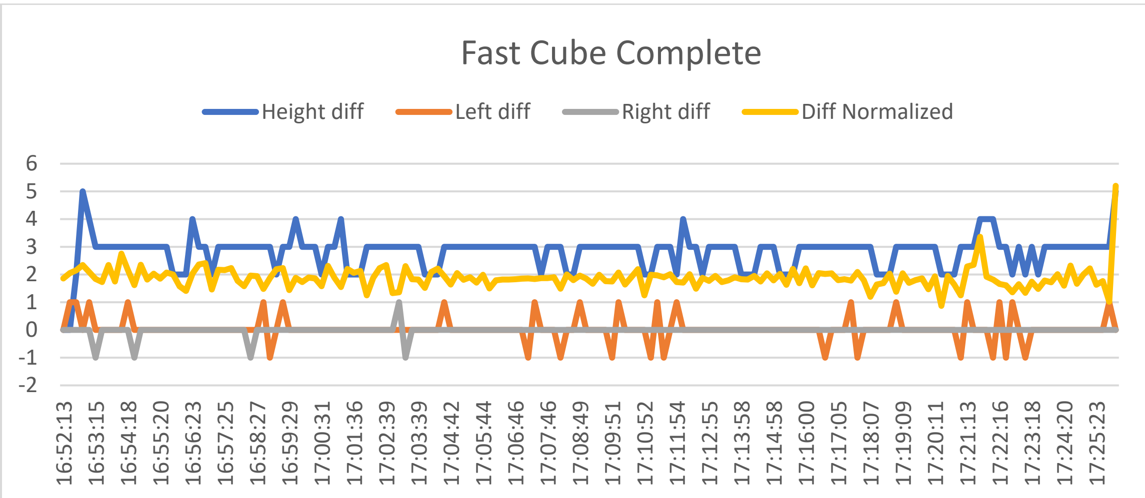


Figure 22 Fast Cube Complete Chart

The result of the test is shown in Figure 22. The part printed complete without any errors, and the system did not detect any errors.

4.3.1.2 Benchy

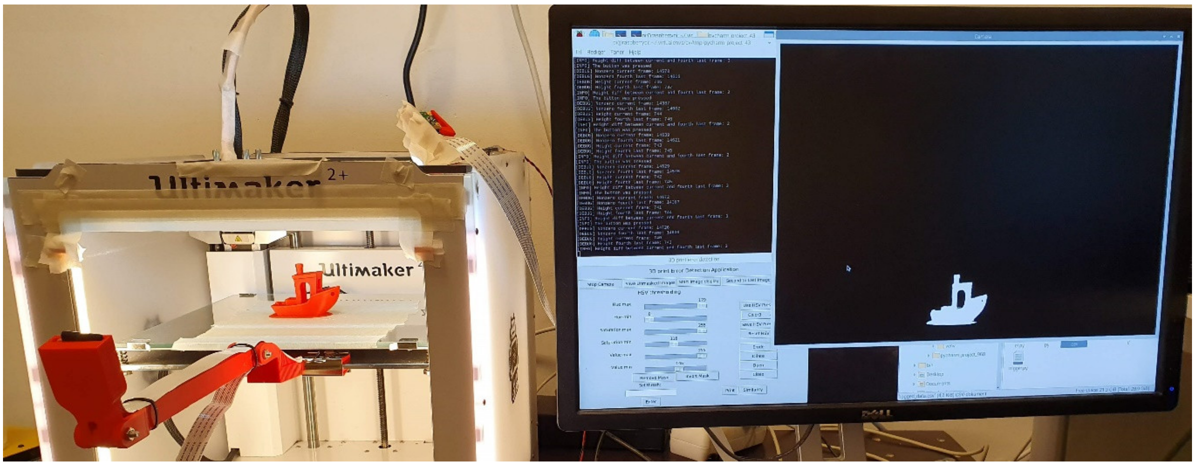


Figure 23 Benchy printed and system display

The object is a model of a boat called “Benchy”. Figure 23 shows the finished print and system display.

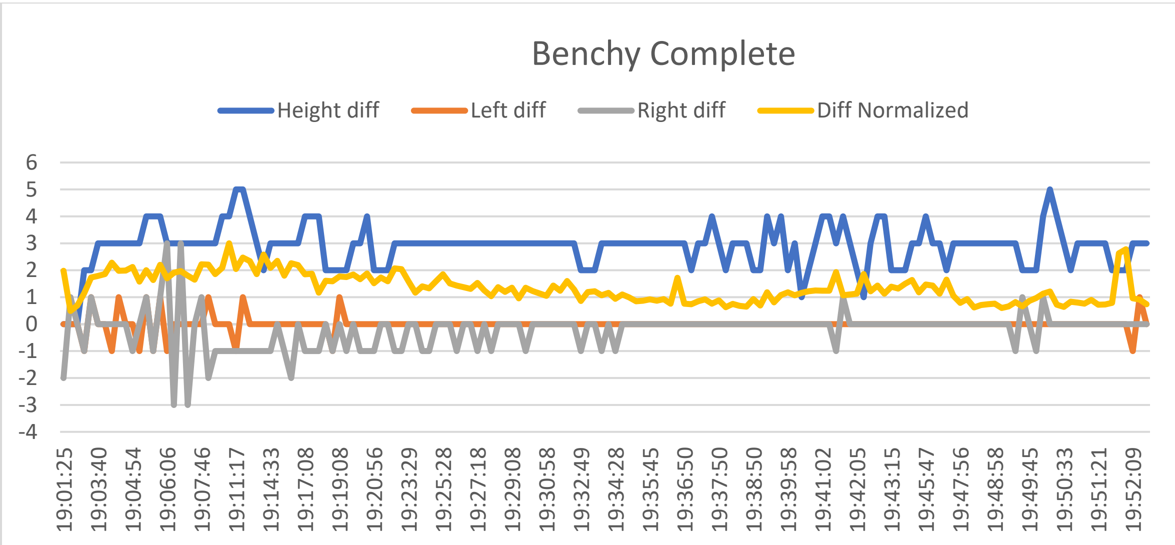


Figure 24 Benchy Complete Chart

The result of the test is shown in Figure 24Figure 22. The part printed complete without any errors. The system detected a warning “Part did not grow” at timestamp 19:02:59, however since this was not detected in the next frame the alert is not set. This show that the 2-stage alert handling mentioned in 3.6 Alert handling works as intended.

4.3.1.3 3 Cubes

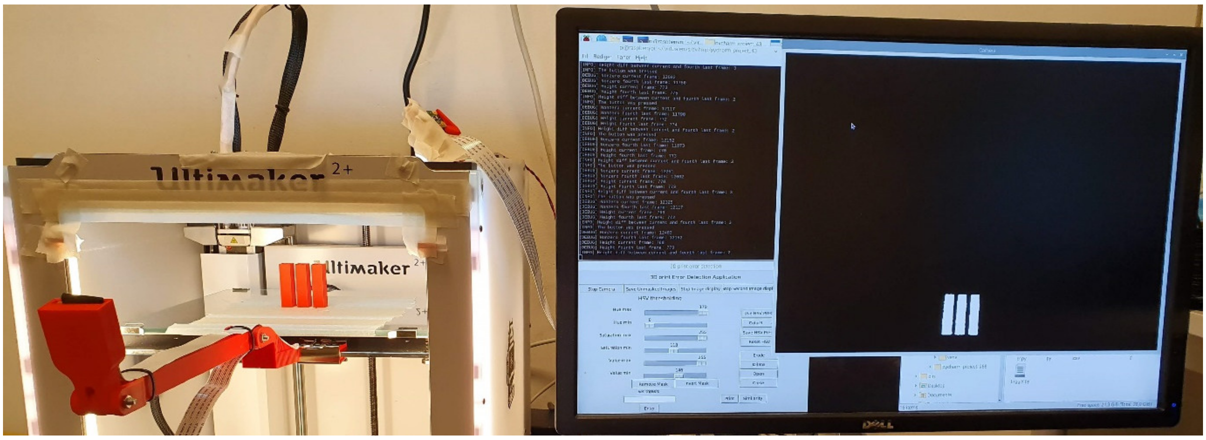


Figure 25 3 Cubes printed and system display

This is a print of three cubes as shown in Figure 25

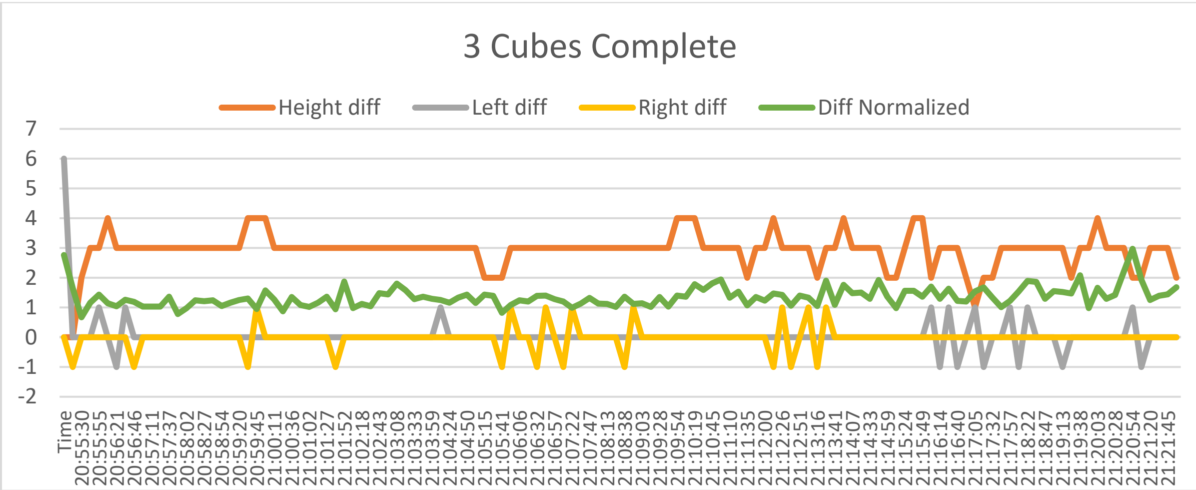


Figure 26 3 Cubes Complete Chart

The result of the test is shown in Figure 26. The print completed without any errors. The system detected a warning "Vertical shift, Left side" at the first comparison, however since this was not detected in the next frame the alert is not set.

4.3.2 Filament runout error

These tests were conducted with interference. After some time printing normally, the filament is retracted.

4.3.2.1 Fast Cube

Figure 27 Fast Cube filament runout error

The "Fast Cube" object is printed, and the filament is retracted to trigger a filament runout error, as shown in Figure 27

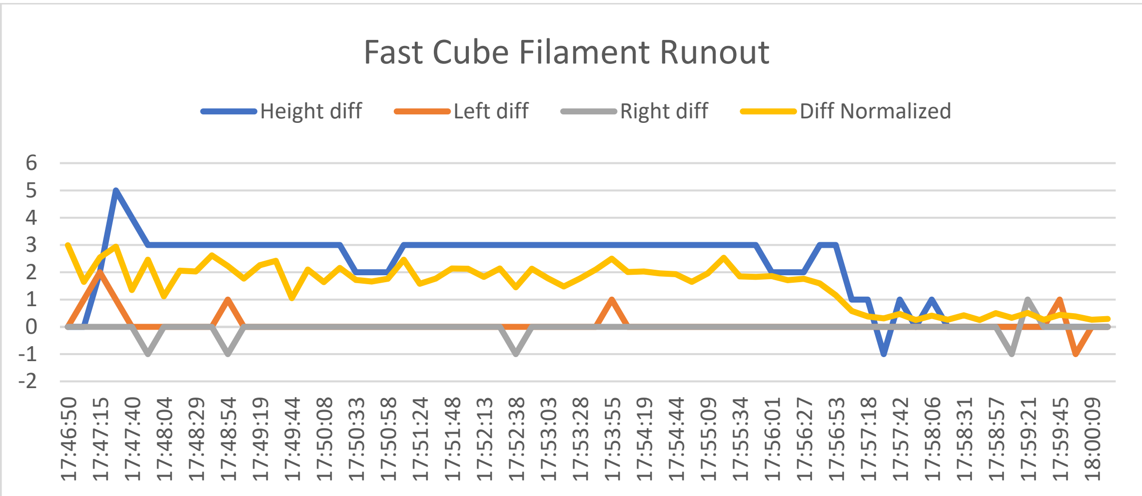


Figure 28 Fast Cube filament runout chart

The part printed until the filament was retracted at timestamp 17:56:53. This meant that the part did not grow after that point. The system detected a warning "Part did not grow" at 17:57:30, however the next frame did not detect a stop in height difference therefore the alert was not triggered.

Fast Cube Filament Runout

17:57:06	58	1	0	0	Normal	0,58
17:57:18	38	1	0	0	Normal	0,38
17:57:30	31	-1	0	0	Part did not grow.	0,31
17:57:42	47	1	0	0	Normal	0,47
17:57:54	25	0	0	0	Part did not grow.	0,25
17:58:06	41	1	0	0	Normal	0,41
17:58:18	26	0	0	0	Part did not grow.	0,26
17:58:31	42	0	0	0	Alert!!! ERROR detected. Check printer	0,42
17:58:44	25	0	0	0	Alert!!! ERROR detected. Check printer	0,25
17:58:57	50	0	0	0	Alert!!! ERROR detected. Check printer	0,5

Figure 29 Section of the data logged in Fast Cube filament runout test

Figure 29 shows the logged data after the filament was retracted. The warnings at 17:57:30 and 17:57:54 was followed by a detection in height difference, which show that the system needs a few layers of frames without growth before alerting.

4.3.2.2 Benchy

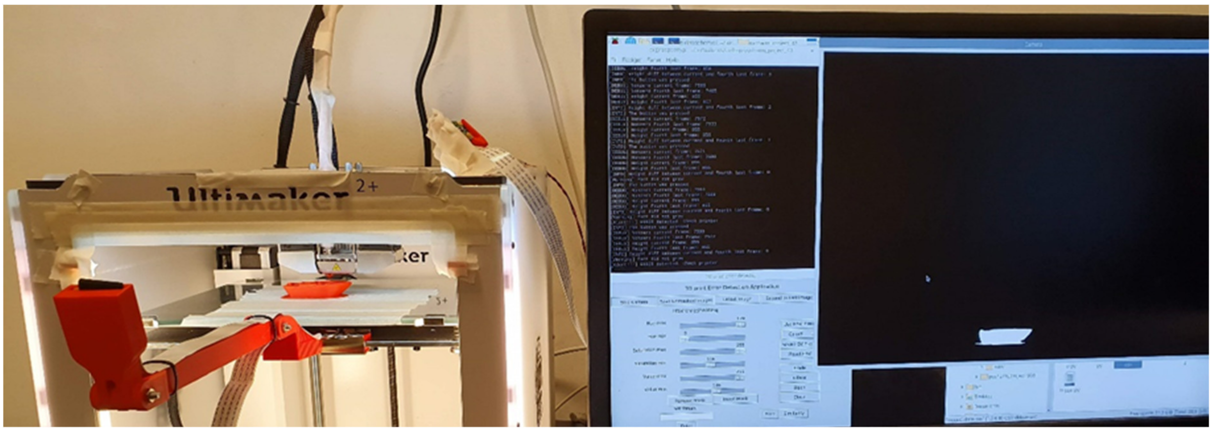


Figure 30 Benchy filament runout error

The "Benchy" object is printed, and the filament is retracted to trigger a filament runout error, as shown in Figure 30.

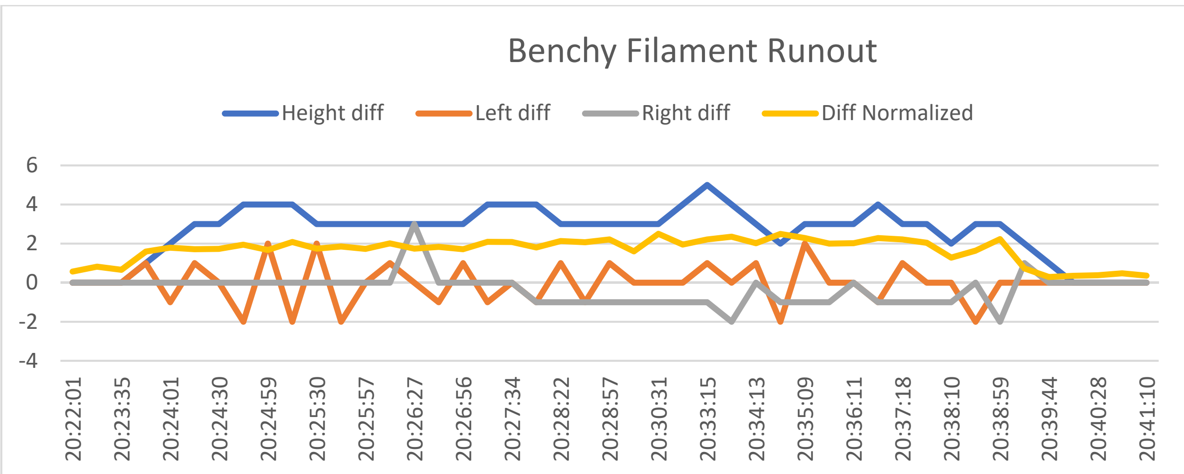


Figure 31 Benchy filament runout chart

The part printed until the filament was retracted at timestamp 20:38:59. This meant that the part did not grow after that point. The system detected a warning “Part did not grow” at 20:40:07, and the alert was set from the next comparison. This is shown in Figure 31 where the blue “Height diff” line hits the zero mark.

4.3.2.3 3 Cubes

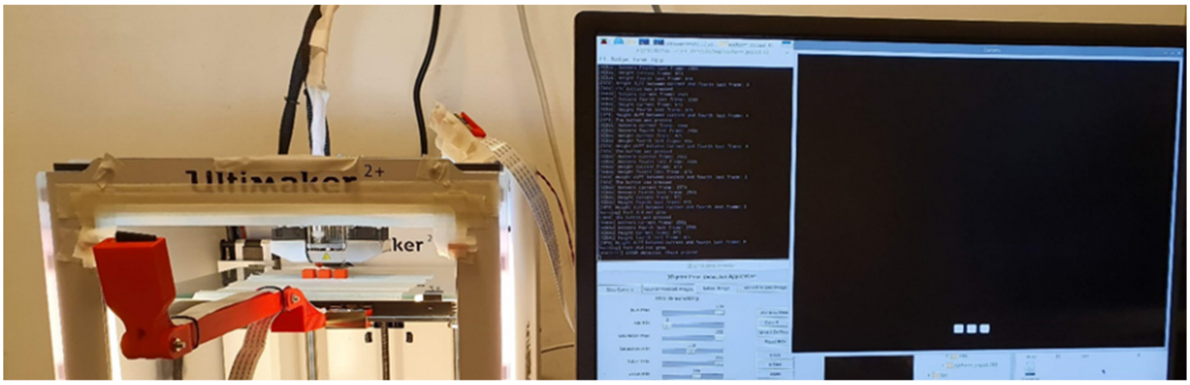


Figure 32 3 Cubes filament runout error

Figure 32 shows that the “3 Cubes” printed, until the filament was retracted.

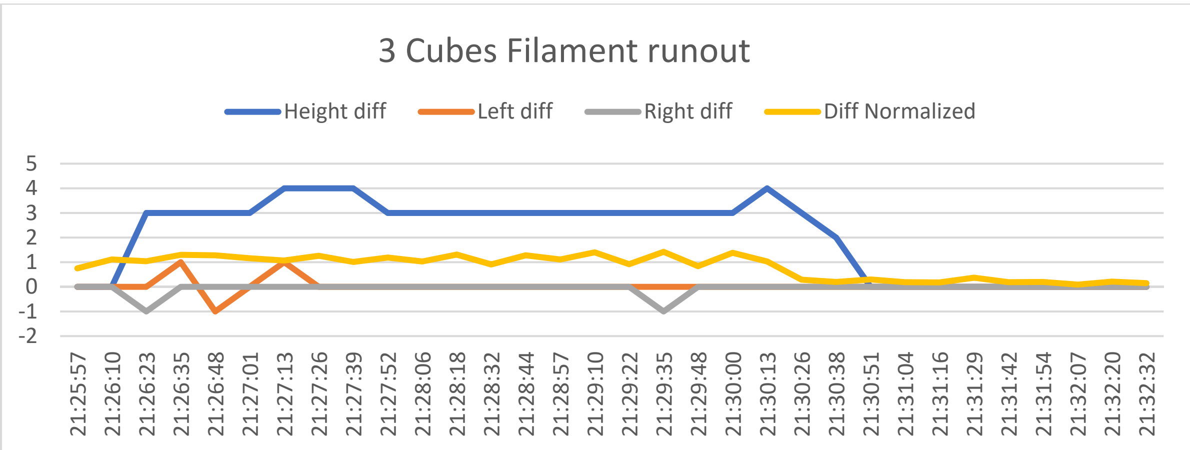


Figure 33 3 Cubes filament runout chart

The parts printed until the filament was retracted at timestamp 21:30:13. The part did not grow after that point. A warning was detected at 21:30:51, and an alert triggered from the next comparison, as shown in Figure 33.

4.3.3 Shifted part error

These tests were conducted with interference. After some time printing normally, the part is carefully loosened from the print bed. This was done by turning off the heated bed, and slightly lifting the edges of the print without shifting it. The shift of the object was caused by the print head.

4.3.3.1 Fast Cube

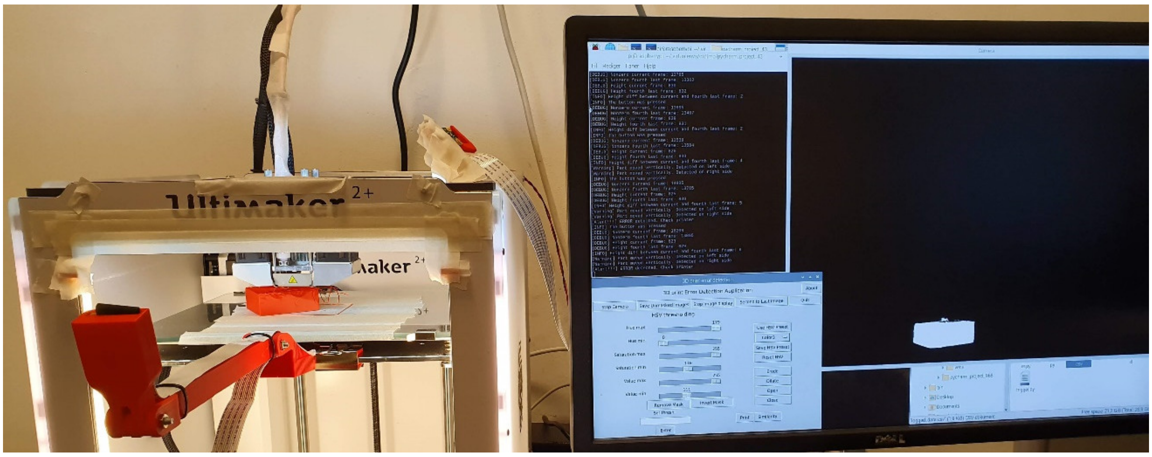


Figure 34 Fast Cube shifted error

Figure 34 shows the part shifted to the left during the test.

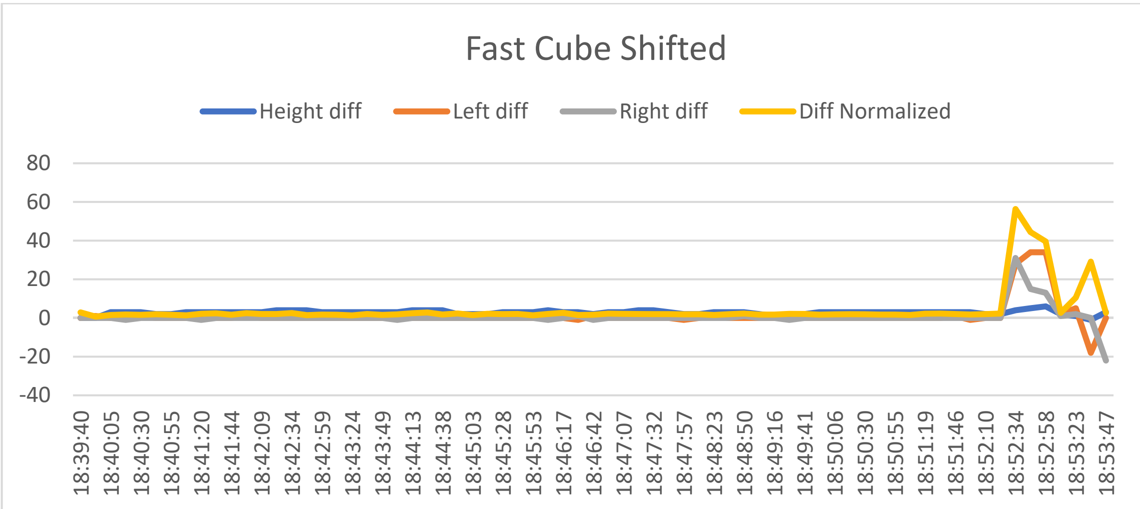


Figure 35 Fast Cube shifted chart

The part printed normally before being loosened at timestamp 18:52:22. A warning “Vertical shift, Left side. Vertical shift, Right side” was triggered at the next frame comparison, followed by the alert at timestamp 18:52:46. The chart in Figure 35 shows that the Left and Right diff spikes at the warning. The difference in pixel-wise comparison also spikes as the shifted part covers a larger area in the frame.

4.3.3.2 Benchy

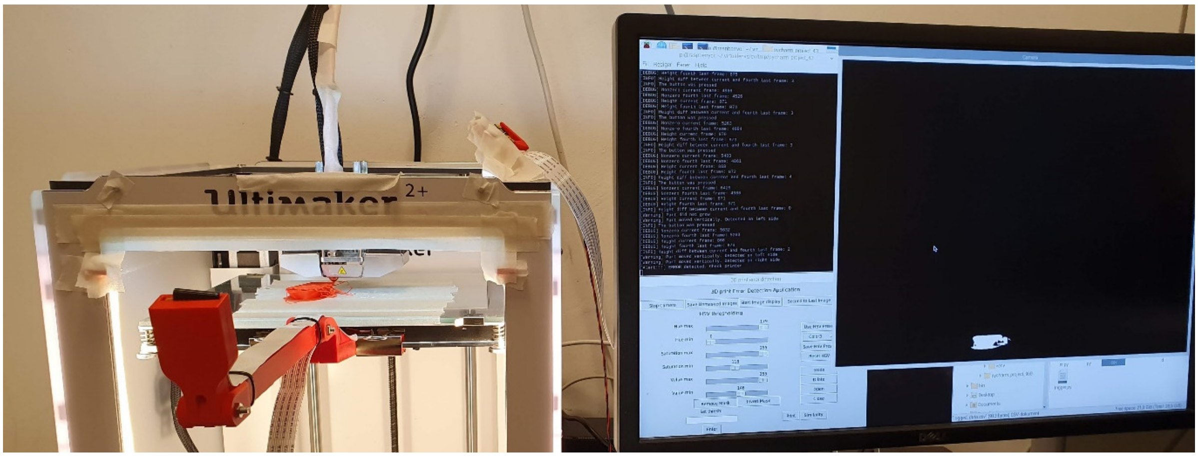


Figure 36 Benchy shifted error

Figure 36 show the “Benchy” object after shifting has occurred.

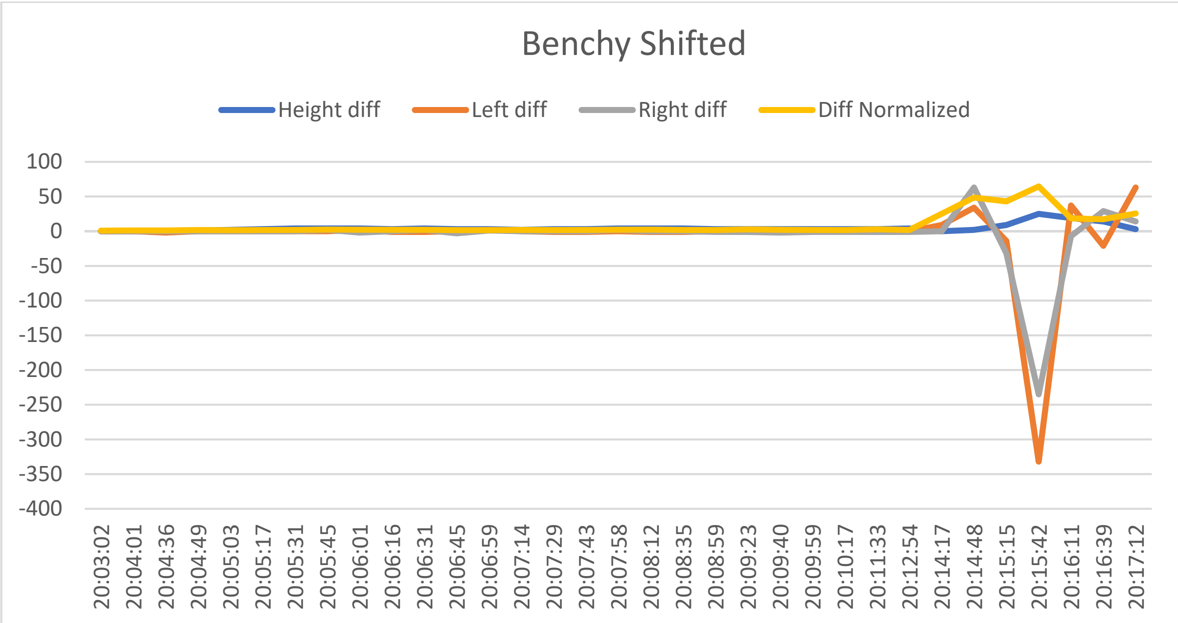


Figure 37 Benchy shifted chart

The part printed normally until 20:12:54, where it was loosened. The next comparison detected a warning “Part did not grow. Vertical shift, Left side”, because the part had shifted. Figure 37 show that left and right diff spikes at 20:14:17, this indicates a shift to the left. And at 20:15:15 the values spike the other way, indicating a shift to the right. This was because the part was further shifted by the print head at the following layers.

4.3.3.3 3 Cubes

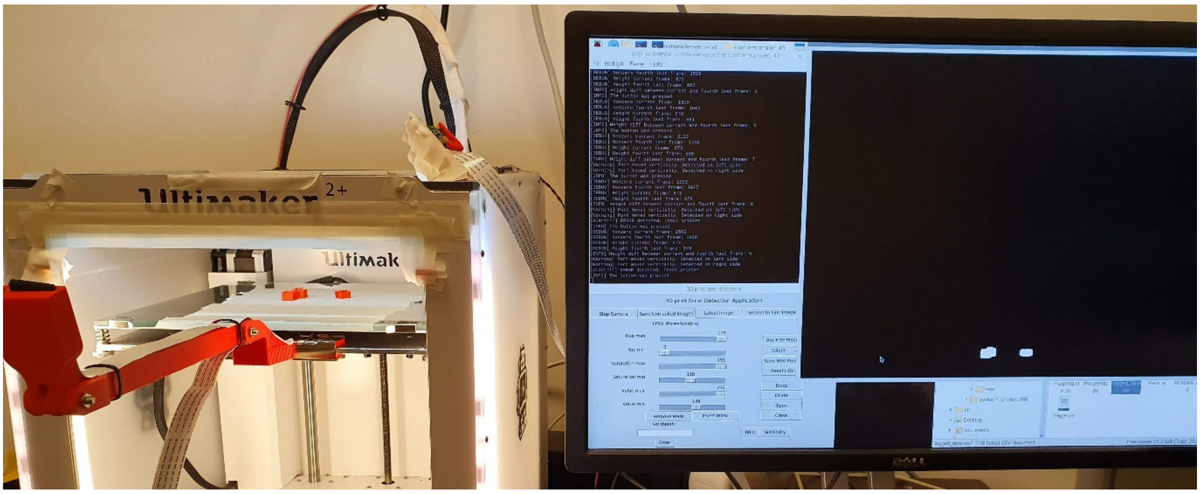


Figure 38 3 Cubes shifted error

Figure 38 shows the 3 Cubes print after it has shifted. The print consists of three objects, and only two of these where loosened, to check if the system would detect the error even if one object remained normal.

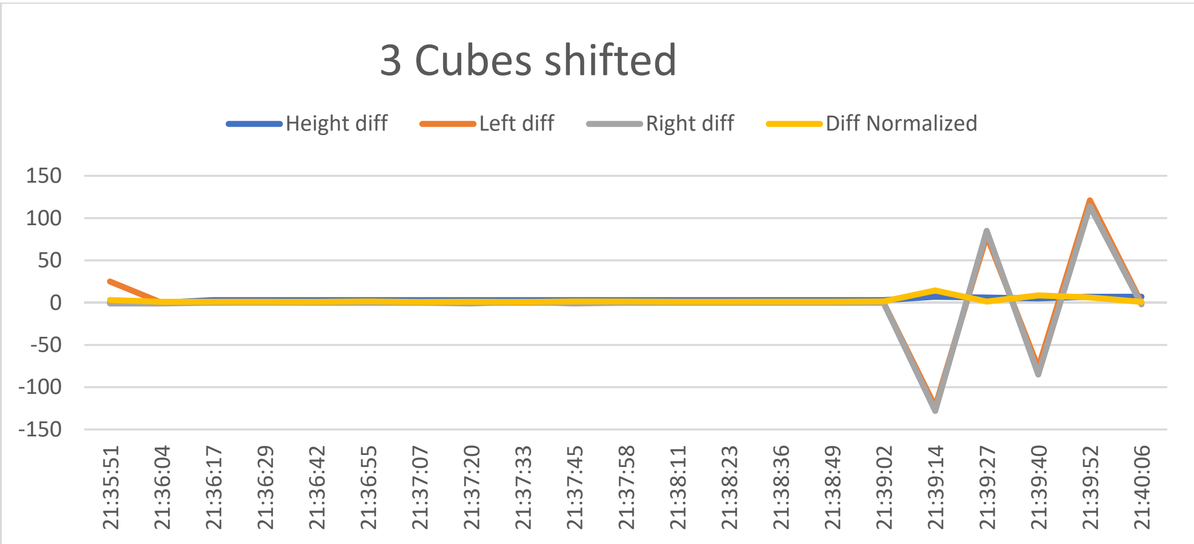


Figure 39 3 Cubes shifted chart

The two objects were loosened at timestamp 21:39:02. A warning “Vertical shift, Left side. Vertical shift, Right side” was detected in the next comparison, followed by an alert. Figure 39 shows the spikes of where the shift occurred. In addition, the first comparison detected a warning at 21:35:51, however this did not trigger an alert as it was not an error.

4.4 Accuracy

The measurement of an object at the middle of the print bed shows that the camera frame was 3 pixels per millimetre in vertical and 3.1 pixels per millimetre in horizontal direction, for a resolution of 1280x960, as documented in "3.5.2 Gather images".

The layer height of the print also affects the accuracy. The most common layer height for this printer is between 0.3mm and 0.15mm. This means that one layer of 0.3mm is approximately 1 pixel high in the images gathered. However, by averaging the height difference gathered in the tests the result is ~2.87 pixels in the difference over three frames. Giving less than 1 pixel at 0.3mm layer height. Which means that the resolution chosen cannot detect one layer alone.

Chapter 5: Discussion

5.1 Project

The application developed by this project is structured in such a way that it can be used for developing other image analysis applications, as this was defined in the scope of the project. This on one hand, required that the project spent some time in achieving this structure, which could have been used for researching and testing other methods for detecting errors in 3D printing. On the other hand, the result from using and testing the system show that the methods explored in this project is useful in detecting errors.

5.2 Key findings

The development process found that template matching did not yield usable results and was therefore left out of the final application. However, the development process found that using pixel-wise differentiating can be used, and the tracking of extreme points in the contours of the object proved very useful for achieving the results.

The result of printing without errors, as documented in 4.3.1 Error free prints indicates that the system does not trigger false positive warnings. However, the tests conducted does not cover all possible situations, and further testing is advised.

The result of printing with errors, as documented in 4.3.2 Filament runout error and 4.3.3 Shifted part error shows that the system is able to detect errors of class 1 (Detachment) and class 2 (Missing material flow) from 2.1.2 Errors. However, the other three error classes are not covered by this system.

The accuracy obtained from the tests, documented in 4.4 Accuracy show that differentiating the last frame and the fourth last frame is necessary to detect a high enough change in height at a resolution of 1280x960. This however leads to a delay from the error occurring to the system detects it. The delay is acceptable for this system.

The resolution (1280x960) used for the system resulted in a less than one pixel covered by a layer height of 0.3mm. Furthermore, if using a lower layer height than this, the differentiating over the last and the fourth last frame would not be enough for the detection. However, the camera supports a resolution of 2592x1944, which would result in around 2 pixels for a

0.3mm layer height. Further development should consider this and using a larger distance between the frames in the differentiation.

The 2-stage alert handling described in 3.6 Alert handling, proved useful when printing the error free test 4.3.1.2 Benchy. This test showed that single alerts can occur in the system, however due to the alert handling being implemented the system did not detect this as an error.

5.3 Limitations

One limitation with this approach is the usage of a single camera. This can only detect error that are visible to the camera. Any error that occurs behind another object or otherwise not visible to the angel of the camera, would not be detected. However, most errors will be visible from the camera.

Another limitation is the need to mount the camera to the print bed. This can be a problem as some printers is enclosed or of such formfactor that mounting to the bed physically is not possible.

The use of HSV threshold limits the system to using colour that stand out from the surroundings. In addition, this is a limitation when using black, white or shaded materials as mentioned in 3.7 System tests.

By using hardware with relative low processing power, the project had to limit the resolution of the gathered images in order to maintain a live feed of acceptable frame rate. This led to a limitation in accuracy, which in turn led to the need for differencing between the current image and the fourth last image.

The system also has a limitation regarding the need for being on, in order to hold the images in the register. This is because frames are held in memory and not written to the disk. This means that if the system needs to restart during a print, all previous captured frames would be disregarded.

5.4 Recommendations

The use of a raspberry pi has as mentioned limitations, therefore the use of more powerful hardware should be addressed. The application is designed in way that it could run from a pc. Another way of overcoming the limitations of the raspberry pi is to capture images at multiple resolutions. By separating the images used for processing and the images used for displaying the live camera feed, they could be captured at different resolutions. This however was not implemented in this project.

5.5 Impact

The result of the tests done show that the system can be used for error detection. However, as the result shows, the system can alert warnings even if an error is not present. This means that one should not fully trust the system to monitor and detect, as the possibility of false positive can occur. However, these false warnings were captured by the 2-stage alert handling, but further tests should be conducted to verify more.

The impact of this project can be looked at as a good start for further development. The project has developed a solution that is available open source, as determined by 1.4 Scope. This means that anyone of interest can use or continue the work on the application.

Chapter 6: Conclusion

The main objective of this project was to develop a system that visually monitors the process of 3D printing to detect errors, and alerts if an error is detected. This project has achieved this objective by developing an approach to detecting errors in the 3D printing process. This approach used methods that compares consecutive frames and the latest frame to the fourth latest frame in order to detect changes. The methods that were found useful was pixel-wise differentiating, tracking of extreme points in contours of the objects and a 2-stage alert handling to avoid false alerts. The result from these methods show that error detection in 3D printing can be done by computer vision.

The system developed by the project has some limitations regarding visibility from the use of a single camera, the need to mount the camera relative to the print area, the need for using filament colours that stands out from the surroundings, accuracy from the resolution needed to maintain a live feed, and lastly the need for uptime to hold frames in memory. However, the results from the test preformed in this project show that the system can have an impact towards further development.

6.1 Further work

6.1.1 Optimizing

This project recommends further work in optimizing the system in order to use higher resolution and achieve more accuracy.

6.1.2 Rendered model

Implementing comparing captured images with images of the rendered model can increase accuracy, make the system more robust and could lead to detection of other error classes than 1 and 2.

6.1.3 Octoprint plugin

The system could be very useful to many if implemented to Octoprint as a plugin. Octoprint allows control over the printer which means that it can stop the printer if an error occurs. Many people have already an Octoprint server monitoring their printers. With this system implemented the server could send and alert and snapshot of the print if an error is detected, or fully automate the process of stopping the printer when an error is detected.

Bibliography

3D Hubs, n.d. *What is 3D printing? The definitive guide.* s.l.:s.n.

Barnes, D. J. & Kölling, M., 2017. *Objects First with Java.* Sixth edition ed. Essex: Pearson Education Limited.

Baumann, F. W. & Roller, D., 2016. Vision based error detection for 3D printing processes. *MATEC Web of Conferences,* p. 8.

Bosch, J., 1996. *Design Patterns as Language Constructs.* Ronneby, Sweden: University of Karlskrona/Ronneby.

Easton, Jr, R. L., 2010. *Fundamentals of Digital Image Processing.* s.l.:s.n.

Gringer, 2018. *Wikimedia.* [Online]
Available at: https://commons.wikimedia.org/wiki/File:FDM_printing_diagram.png
[Accessed 10 12 2019].

Hale, T. B., 2014. *The Raspberry Pi Hobbyist.* [Online]
Available at: http://raspberrypihobbyist.blogspot.com/2014/11/debouncing-gpio-input.html
[Accessed 05 11 2019].

Lyngby, R. et al., 2017. *In-line 3D print failure detection using computer vision.* Leuven, Belgium: s.n.

Martin, D., 2013. *A Practical Guide to Machine Vision Lighting.* [Online]
Available at: https://www.advancedillumination.com/wp-content/uploads/2018/10/A-Practical-Guide-to-Machine-Vision-Lighting-v.-4-Generic.pdf
[Accessed 10 12 2019].

OpenCV, 2019. *Documentation.* 4.1.1 ed. s.l.:s.n.

Oropallo, W. & Piegl, L. A., 2015. Ten challenges in 3D printing. *Engineering with Computers*.

Paulsen, T., 2019. *devproj.* s.l.:s.n.

Paulsen, T., 2019. *errordetector,* s.l.: s.n.

Paulsen, T., 2019. *Ultimaker Raspberry Pi Camera bed mount.* s.l.:Thingiverse.

Python Software Foundation, 2019. *Python documentation 25.1. tkinter — Python interface to Tcl/Tk.* [Online]
Available at: https://docs.python.org/3.5/library/tkinter.html
[Accessed 01 12 2019].

Rosebrock, A., 2019. *PyImageSearch.* [Online]
Available at: https://www.pyimagesearch.com/
[Accessed 16 09 2019].
Ross, C., 2019. *The Python Wiki.* [Online]
Available at: https://wiki.python.org/moin/TkInter
[Accessed 05 12 2019].
SharkD, 2008. *Wikimedia.* [Online]
Available at:
https://commons.wikimedia.org/wiki/File:Color_solid_comparison_hsl_hsv_rgb_cone_sphere_cube_cylinder.png
[Accessed 10 12 2019].
Sinha, S., n.d. *Geeksforgeeks.* [Online]
Available at: https://www.geeksforgeeks.org/find-and-draw-contours-using-opencv-python/
[Accessed 12 12 2019].
Straub, J., 2015. *Initial Work on the Characterization of Additive Manufacturinga (3D Printing) Using Software Image Analysis,* Fargo, ND, United States: North Dakota State University.
Umbaugh, S. E., 2018. *Digital Image Processing and Analysis Applications with MATLAB and CVIPtools.* Third Edition ed. Edwardsville, United States: Tyalor & Francis Group: CRC Press.

Appendix

A: Pre-project report
B: Data from tests

A: Pre-project report

FORPROSJEKT - RAPPORT
FOR BACHELOROPPGAVE

NTNU
Kunnskap for en bedre verden

TITTEL:
Feildeteksjon i 3D printing ved bruk av kamerasyn

KANDIDATNUMMER(E):			
DATO:	EMNEKODE: **IE303612**	EMNE: **Bacheloroppgave**	DOKUMENT TILGANG: - Open
STUDIUM: **AUTOMATISERINGSTEKNIKK**		ANT SIDER/VEDLEGG: /	BIBL. NR: - Ikke i bruk -

OPPDRAGSGIVER(E)/VEILEDER(E):
Ottar L. Osen, Saleh Abdel-Afou Alaliyat

OPPGAVE/SAMMENDRAG:

3D-printing er ein teknologi som stadig blir meir allment tilgjengeleg. Kostnad og brukarvenlegheit har ført til at det ikkje berre er industri og bedrifter som nyttar seg av teknologien, då det no har blitt meir vanleg med skrivebords 3d-printera også i heimen.

FDM 3D-printing har framleis mykje problem med å skrive ut modellar utan feil eller problem. Feil der modellen lausnar frå plata, vrir og bøyer seg, hektar seg i printerdysa, lagforskyving er eksemplar på problem som kan oppstå.
Denne oppgåva går ut på å detektere om ein 3D-printer, undervegs i prosessen, gjer feil slik at det er ynskeleg å avbryte prosessen. For å oppdage feil skal det nyttast kamera og kontinuerleg bildeanalyse når 3D-printeren er aktiv. Oppgåva er sjølvvald av studenten.

Postadresse
Høgskolen i Ålesund
N-6025 Ålesund
Norway

Besøksadresse
Larsgårdsvegen 2
Internett
www.hials.no

Telefon
70 16 12 00
Epostadresse
postmottak@hials.no

Telefax
70 16 13 00

Bankkonto
7694 05 00636
Foretaksregisteret
NO 971 572 140

Denne oppgaven er en eksamensbesvarelse utført av student(er) ved NTNU i Ålesund.

INNHOLD

1 INNLEDNING

3D-printing er ein teknologi som stadig blir meir allment tilgjengeleg. Kostnad og brukarvenlegheit harført til at det ikkje berre er industri og bedrifter som nyttar seg av teknologien, då det no har blitt meir vanleg med skrivebords 3d-printera også i heimen.

FDM 3D-printing har framleis mykje problem med å skrive ut modellar utan feil eller problem. Feil der modellen lausnar frå plata, vrir og bøyer seg, hektar seg i printerdysa, lagforskyving er eksemplar på problem som kan oppstå.
Denne oppgåva går ut på å detektere om ein 3D-printer, undervegs i prosessen, gjer feil slik at det er ynskeleg å avbryte prosessen. For å oppdage feil skal det nyttast kamera og kontinuerleg bildeanalyse når 3D-printeren er aktiv. Oppgåva er sjølvvald av studenten.

2 BEGREPER

FDM - Fused Deposition Modeling er ein 3D-printing teknikk som smeltar materiale gjennom ei dyse, for så å legge lag for lag av ein modell.

3 PROSJEKTORGANISASJON

Prosjektet er organisert slik at studenten styrar arbeidet sjølv, men med tett oppfylging frå rettleiarane.

4 PROSJEKTBESKRIVELSE

4.1 Problemstilling - målsetting - hensikt

Prosjekter har som hensikt å oppdage dersom det kjem ein feil i løpet av prosessen med å skrive ut ein fysisk modell frå ein digital modell (3D-printing).

Resultatmål: Eit system av bildeanalysemetodar som oppdagar når feil oppstår i 3D-printing, men også kunne luke ut metodar som ikkje fungera like godt eller optimalt og dokumentere dette.

Prosessmål: Auka kompetanse innan bildeanalyse. Fyrstehandserfaring med gjennomføring av utviklingsprosjekt.

4.2 Krav til løsning eller prosjektresultat – spesifikasjon

Prosjektet har som utgangspunkt å detektere feilprint. Der større feil er først prioritert. Kravet er då at løysninga skal detektere og varsle om feilen. Det skal også etterstrevast at løysninga er moduler for vidare implementering.
Prosjektet skal dokumentere metodar som fungera og ikkje fungera.

4.3 Planlagt framgangsmåte(r) for utviklingsarbeidet – metode(r)

Prosjektet tek utgangspunkt i denne forprosjektrapporten som styringsdokument. Aktivitetar er planlagt i eit gantt-diagram, og det skal nyttast personleg kanban (Trello) for fortløpande sporing av aktivitetar gjennom prosjektet. Periodisk internevaluering vil oppdatere desse.

Då dette er eit prosjekt beståande av ein person er det utfordrande å nytte systemutviklingsmetode, då desse er meint for team-arbeid. Likevel er det verktøy i desse metodane som kan nyttast til dette prosjektet. Sprints og retrospectives er verktøy i Agile utviklingsmetoden som vil vere gode verktøy for gjennomføring av prosjektet. Retrospectives gjer at det vil vere eit iterativt utviklingsarbeid der ein undervegs i utviklinga får innsyn i dei spesifikke oppgåvene som må utførast for å nå prosjektmål. I dette prosjektet skal internevalueringa og møte med rettleiarar fungere som retrospective.

4.4 Informasjonsinnsamling – utført og planlagt

Nokre kjelde som kan vere nyttige:

- Forskingsartikkel som såg på liknande problem. Dette vil nok vere mi mest nyttige kjelde for prosjektet.
https://www.researchgate.net/publication/303505965_Vision_based_error_detection_for_3D_printing_processes

- Forskingsartikkel, mykje likt med den fyrste.
https://www.researchgate.net/publication/331864321_In-line_3D_print_failure_detection_using_computer_vision

- Bachelor prosjekt som såg på tracking av posisjon på extruder.
https://cyber.felk.cvut.cz/theses/papers/498.pdf

- Eit firma som har ei løysning for å detektere nokre feil. Men dei har ikkje publisert noko rundt det, og har no endra retning til meir industrielle printera. Truleg lite nyttig kjelde.
https://www.youtube.com/watch?time_continue=74&v=HFIzTGGnaLg

Det meste av informasjonsinnsamling i prosjektet vil vere data frå prototyping, utvikling og testing undervegs i prosjektet. Likevel er det nyttig å sjå på kjeldene som her er oppført, særs dei to forskingsartikkelane.

4.5 Vurdering – analyse av risiko

MATRISE FOR RISIKOVURDERINGER ved NTNU

KONSEKVENS	Svært alvorlig	E1	E2	E3	E4	E5
KONSEKVENS	Alvorlig	D1	D2	D3	D4	D5
KONSEKVENS	Moderat	C1	C2	C3	C4	C5
KONSEKVENS	Liten	B1	B2	B3	B4	B5
KONSEKVENS	Svært liten	A1	A2	A3	A4	A5
		Svært liten	Liten	Middels	Stor	Svært stor
		SANNSYNLIGHET				

Prinsipp over akseptkriterium. Forklaring av fargene som er brukt i risikomatrisen.

Farge		Beskrivelse
Rød		Uakseptabel risiko. Tiltak skal gjennomføres for å redusere risikoen.
Gul		Vurderingsområde. Tiltak skal vurderes.
Grønn		Akseptabel risiko. Tiltak kan vurderes ut fra andre hensyn.

Tid – Det vil vere avgjerande for prosjektet at det blir lagt ned nok tid i prosjektet. Dersom det skulle oppstå utfordringar som sjukdom og liknande, må tiltak innførast. Det er helder ikkje klart i planlegginga kor mykje tid som er naudsynt for å gjennomføre måla, då dette vil vere eit utviklingsprosjekt som vil avdekke kva som er naudsynt for å nå måla.
Risiko: D2
Tiltak: Dersom det skulle oppstå problem med tid skal måla revurderast. Oppgåve som E1 Brukargrensesnitt og F1 Implementering vil då prioriterast bort.

4.6 Hovedaktiviteter i videre arbeid

Nr	Hovedaktivitet	Kostnad	Tid/omfang
A1	Lage testmiljø		2
B1	Datainnsamling		2
C1	Proof-of-concept		3
D1	Utvikling		3
E1	Brukargrensesnitt		2
F1	Implementering		2
G1	Rapportering		?

4.7 Framdriftsplan – styring av prosjektet

4.7.1 Hovedplan

Prosjektet skal gjennomføre planlagde aktivitetar som kjem fram av gantt-diagrammet. Aktivitetane skal sporast i personleg-kanban. Desse verktøya vil kontinuerleg bli oppdaterte av internkontrollen.

4.7.2 Styringshjelpemidler

- Trello
- GanttProject
- Sourcetree
- Forprosjektrapport
- Evalueringsrapport
- Referat rettleiingsmøte

4.7.1 Utviklingshjelpemidler

- PyCharm /m SSH for deploying
- Python /m nødvendige bibliotek
- Sourcetree og Bitbucket for versjonskontroll
- Rasbian OS (Raspberri Pi)
- Autodesk Fusion 360
- Slic3r og Cura

4.7.2 Intern kontroll – evaluering

Internkontroll av framdrifta i prosjektet skal utførast ved å sette av tid til å gå igjennom heilheita av prosjektet med omsyn på prosjektmåla. Det skal skrivast ein evalueringsrapport minst kvar 14 dag, oftare om behov.
Denne skal innehalde:

- Evaluering av framdrifta, arbeidsmetode og eventuelle forbetringar.
- Aktivitetar som er fullført og ikkje fullført til planlagt tid.
- Aktivitetar som fell ut av prosjektet.
- Nye aktivitetar som blir avdekka i prosjektet
- Oppdatert gantt diagram

5 DOKUMENTASJON

5.1 Rapporter og tekniske dokumenter

- Bacheloroppgåve som dokumentera prosjektet.
- Evalueringsrapport som dokumentera framdrift.

6 PLANLAGTE MØTER OG RAPPORTER

6.1 Møter med styringsgruppen

- Møte med rettleiarar kvar 14dag. Dokumenterast i eit referat.
- Internkontroll minst kvar 14 dag, eller oftare ved behov. Dokumenterast i ein evalueringsrapport

7 PLANLAGT AVVIKSBEHANDLING

- Internkontrollen og møte med rettleiar vil avdekke om framdrift ikkje går som planlagt.
- Planlagt prosedyre er at det skal gjerast tiltak på aktivitetsplanen der ein må prioritere kva aktivitetar som må felle bort frå prosjektet for å fullføre.
- Oppdatering av gantt diagram
- Dokumentasjon på avvik og/eller større endringar i prosjektet.

8 UTSTYRSBEHOV/FORUTSETNINGER FOR GJENNOMFØRING

- Det er behov for å reservere ein Ultimaker 3D printer, samt ein raspberry pi med kameramodul. Dette blir ståande på tunglabben gjennom prosjektet.

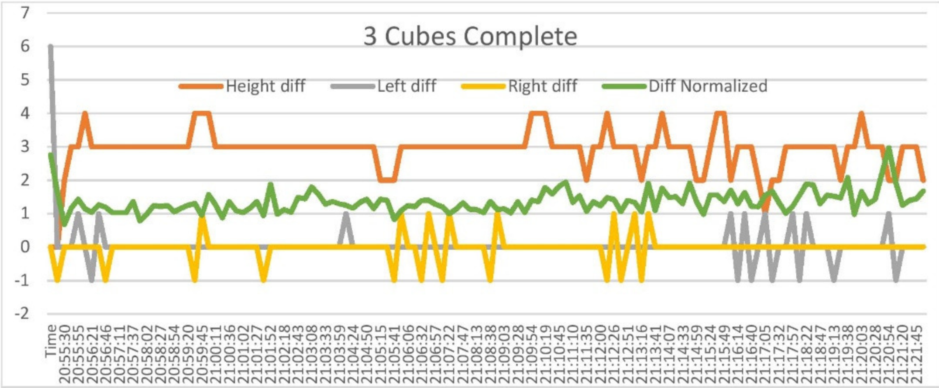
VEDLEGG

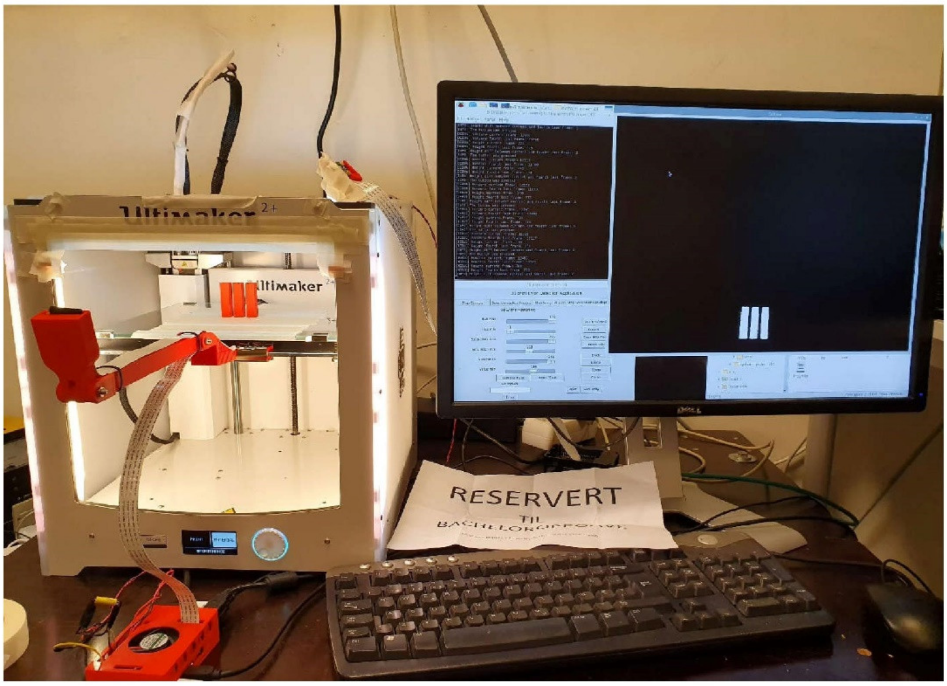
1 Trello – Personleg kanban *(trello_v1.pdf)*

2 GanttProject *(gantt_v1.pdf)*

B: Data from tests

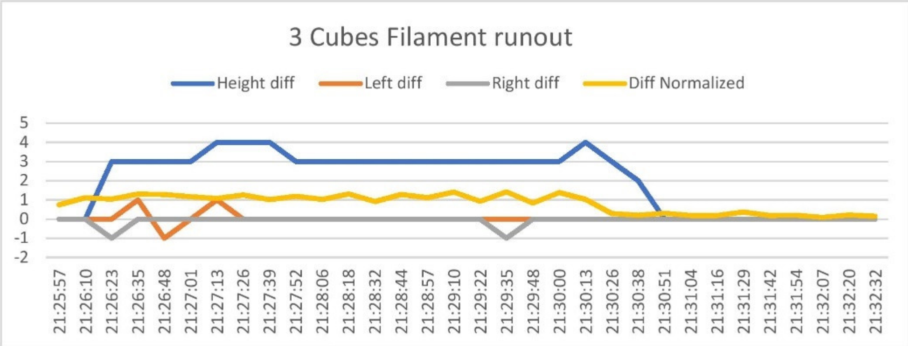
3 Cubes Charts

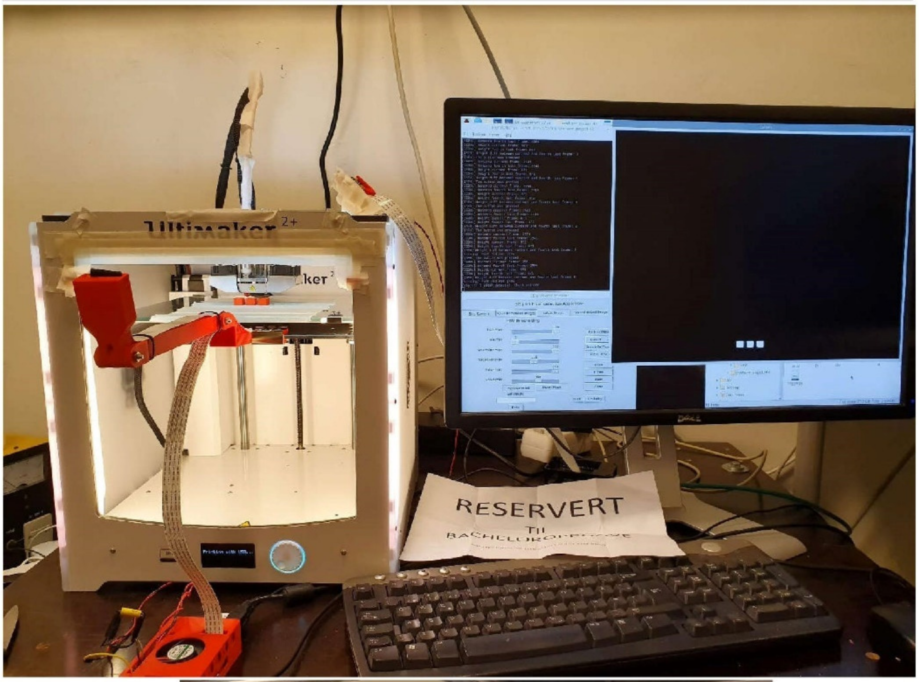


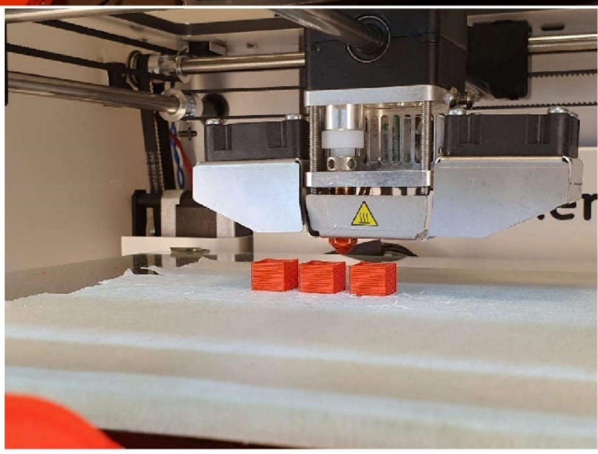


B: Data from tests

3 Cubes Charts

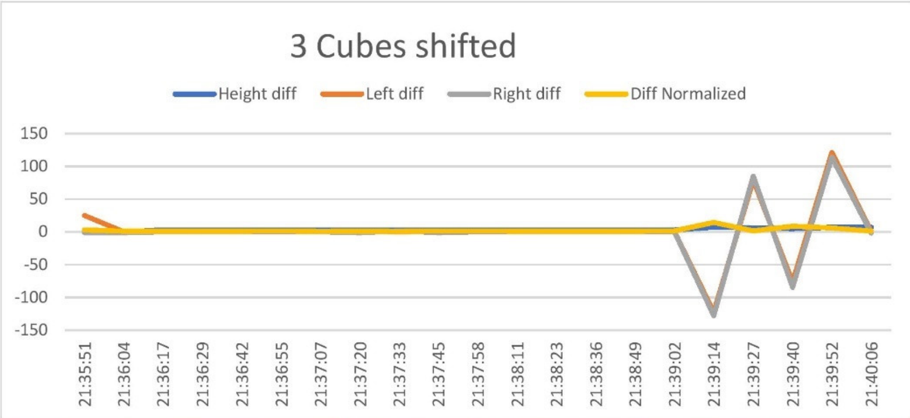


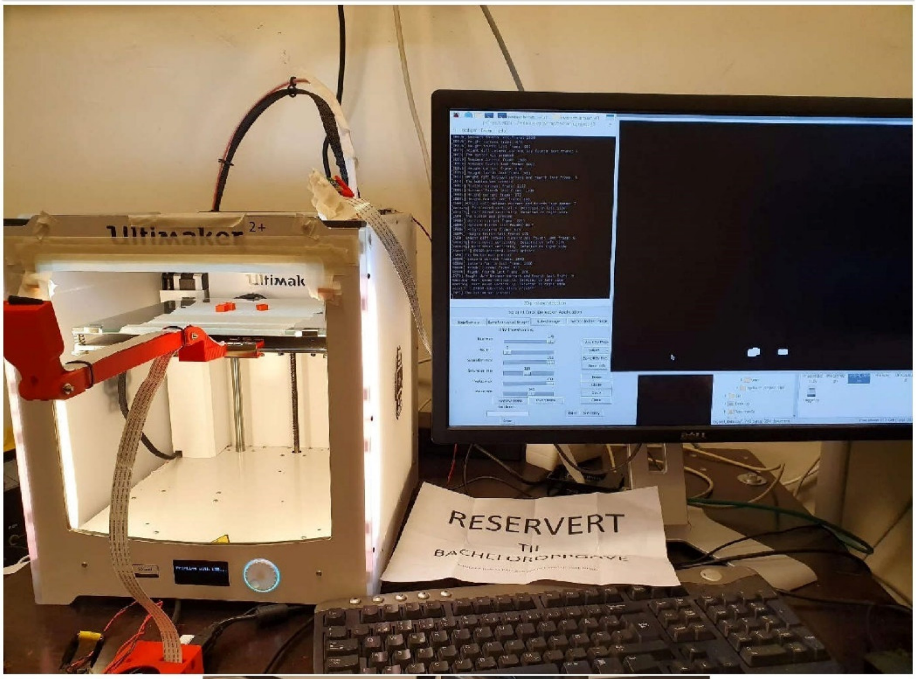


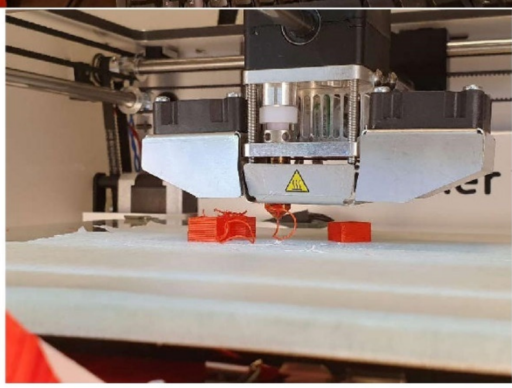


B: Data from tests

3 Cubes Charts







3 Cubes Complete

Time	Diff	Height diff	Left diff	Right diff	Status	Diff Normalized
20:55:17	276	0	6	0	Vertical shift, Left side.	2,76
20:55:30	165	0	0	-1	Normal	1,65
20:55:43	67	2	0	0	Normal	0,67
20:55:55	116	3	0	0	Normal	1,16
20:56:08	143	3	1	0	Normal	1,43
20:56:21	114	4	0	0	Normal	1,14
20:56:33	104	3	-1	0	Normal	1,04
20:56:46	126	3	1	0	Normal	1,26
20:56:59	119	3	0	-1	Normal	1,19
20:57:11	103	3	0	0	Normal	1,03
20:57:24	103	3	0	0	Normal	1,03
20:57:37	103	3	0	0	Normal	1,03
20:57:49	136	3	0	0	Normal	1,36
20:58:02	78	3	0	0	Normal	0,78
20:58:14	98	3	0	0	Normal	0,98
20:58:27	124	3	0	0	Normal	1,24
20:58:40	121	3	0	0	Normal	1,21
20:58:54	124	3	0	0	Normal	1,24
20:59:06	105	3	0	0	Normal	1,05
20:59:20	116	3	0	0	Normal	1,16
20:59:33	125	3	0	0	Normal	1,25
20:59:45	130	4	0	-1	Normal	1,3
20:59:58	95	4	0	1	Normal	0,95
21:00:11	157	4	0	0	Normal	1,57
21:00:23	127	3	0	0	Normal	1,27
21:00:36	87	3	0	0	Normal	0,87
21:00:48	135	3	0	0	Normal	1,35
21:01:02	108	3	0	0	Normal	1,08
21:01:14	102	3	0	0	Normal	1,02
21:01:27	116	3	0	0	Normal	1,16
21:01:40	136	3	0	0	Normal	1,36
21:01:52	94	3	0	-1	Normal	0,94
21:02:05	187	3	0	0	Normal	1,87
21:02:18	98	3	0	0	Normal	0,98
21:02:30	112	3	0	0	Normal	1,12
21:02:43	104	3	0	0	Normal	1,04
21:02:56	148	3	0	0	Normal	1,48
21:03:08	144	3	0	0	Normal	1,44
21:03:21	180	3	0	0	Normal	1,8
21:03:33	158	3	0	0	Normal	1,58
21:03:46	128	3	0	0	Normal	1,28
21:03:59	136	3	0	0	Normal	1,36
21:04:12	129	3	0	0	Normal	1,29
21:04:24	125	3	1	0	Normal	1,25
21:04:37	116	3	0	0	Normal	1,16
21:04:50	133	3	0	0	Normal	1,33
21:05:02	143	3	0	0	Normal	1,43
21:05:15	115	3	0	0	Normal	1,15
21:05:28	143	2	0	0	Normal	1,43

3 Cubes Complete

21:05:41	139	2	0	0	Normal	1,39
21:05:53	82	2	0	-1	Normal	0,82
21:06:06	108	3	0	1	Normal	1,08
21:06:19	124	3	0	0	Normal	1,24
21:06:32	120	3	0	0	Normal	1,2
21:06:44	139	3	0	-1	Normal	1,39
21:06:57	140	3	0	1	Normal	1,4
21:07:09	128	3	0	0	Normal	1,28
21:07:22	120	3	0	-1	Normal	1,2
21:07:35	99	3	0	1	Normal	0,99
21:07:47	113	3	0	0	Normal	1,13
21:08:00	132	3	0	0	Normal	1,32
21:08:13	113	3	0	0	Normal	1,13
21:08:25	112	3	0	0	Normal	1,12
21:08:38	102	3	0	0	Normal	1,02
21:08:51	136	3	0	-1	Normal	1,36
21:09:03	112	3	0	1	Normal	1,12
21:09:16	114	3	0	0	Normal	1,14
21:09:28	102	3	0	0	Normal	1,02
21:09:41	135	3	0	0	Normal	1,35
21:09:54	103	3	0	0	Normal	1,03
21:10:07	140	4	0	0	Normal	1,4
21:10:19	136	4	0	0	Normal	1,36
21:10:32	178	4	0	0	Normal	1,78
21:10:45	159	3	0	0	Normal	1,59
21:10:57	181	3	0	0	Normal	1,81
21:11:10	194	3	0	0	Normal	1,94
21:11:22	133	3	0	0	Normal	1,33
21:11:35	153	3	0	0	Normal	1,53
21:11:48	107	2	0	0	Normal	1,07
21:12:00	135	3	0	0	Normal	1,35
21:12:13	123	3	0	0	Normal	1,23
21:12:26	147	4	0	-1	Normal	1,47
21:12:38	142	3	0	1	Normal	1,42
21:12:51	106	3	0	-1	Normal	1,06
21:13:04	140	3	0	0	Normal	1,4
21:13:16	133	3	0	1	Normal	1,33
21:13:29	105	2	0	-1	Normal	1,05
21:13:41	190	3	0	1	Normal	1,9
21:13:55	109	3	0	0	Normal	1,09
21:14:07	176	4	0	0	Normal	1,76
21:14:20	148	3	0	0	Normal	1,48
21:14:33	150	3	0	0	Normal	1,5
21:14:46	129	3	0	0	Normal	1,29
21:14:59	192	3	0	0	Normal	1,92
21:15:11	136	2	0	0	Normal	1,36
21:15:24	98	2	0	0	Normal	0,98
21:15:36	156	3	0	0	Normal	1,56
21:15:49	156	4	0	0	Normal	1,56
21:16:02	136	4	0	0	Normal	1,36

3 Cubes Complete

21:16:14	170	2	1	0	Normal	1,7
21:16:27	129	3	-1	0	Normal	1,29
21:16:40	163	3	1	0	Normal	1,63
21:16:52	122	3	-1	0	Normal	1,22
21:17:05	120	2	0	0	Normal	1,2
21:17:18	154	1	1	0	Normal	1,54
21:17:32	167	2	-1	0	Normal	1,67
21:17:44	133	2	0	0	Normal	1,33
21:17:57	101	3	0	0	Normal	1,01
21:18:09	122	3	1	0	Normal	1,22
21:18:22	154	3	-1	0	Normal	1,54
21:18:35	189	3	1	0	Normal	1,89
21:18:47	186	3	0	0	Normal	1,86
21:19:00	129	3	0	0	Normal	1,29
21:19:13	155	3	0	0	Normal	1,55
21:19:25	152	3	-1	0	Normal	1,52
21:19:38	147	2	0	0	Normal	1,47
21:19:51	208	3	0	0	Normal	2,08
21:20:03	98	3	0	0	Normal	0,98
21:20:16	166	4	0	0	Normal	1,66
21:20:28	129	3	0	0	Normal	1,29
21:20:41	142	3	0	0	Normal	1,42
21:20:54	221	3	0	0	Normal	2,21
21:21:07	297	2	1	0	Normal	2,97
21:21:20	193	2	-1	0	Normal	1,93
21:21:33	125	3	0	0	Normal	1,25
21:21:45	139	3	0	0	Normal	1,39
21:21:58	144	3	0	0	Normal	1,44
21:22:11	168	2	0	0	Normal	1,68

3 Cubes Filament Runout

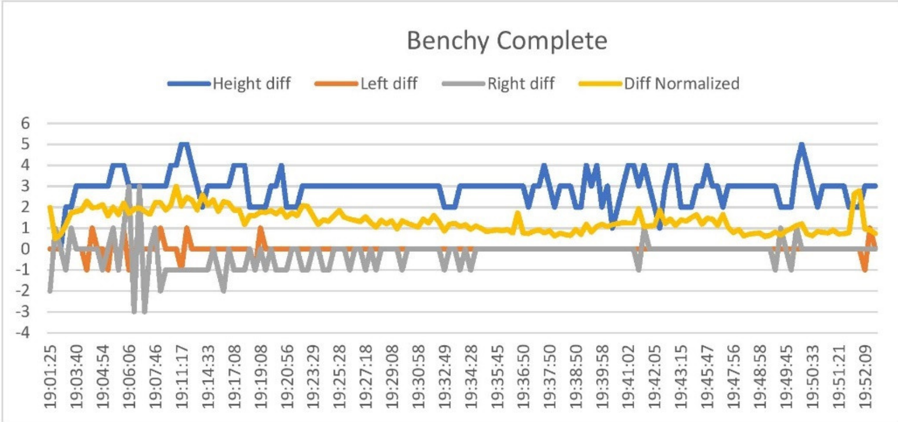
Time	Diff	Height	Left dif	Right d	Status	Diff Normalized
21:25:57	75	0	0	0	Normal	0,75
21:26:10	111	0	0	0	Normal	1,11
21:26:23	104	3	0	-1	Normal	1,04
21:26:35	130	3	1	0	Normal	1,3
21:26:48	128	3	-1	0	Normal	1,28
21:27:01	116	3	0	0	Normal	1,16
21:27:13	107	4	1	0	Normal	1,07
21:27:26	126	4	0	0	Normal	1,26
21:27:39	101	4	0	0	Normal	1,01
21:27:52	119	3	0	0	Normal	1,19
21:28:06	103	3	0	0	Normal	1,03
21:28:18	131	3	0	0	Normal	1,31
21:28:32	91	3	0	0	Normal	0,91
21:28:44	128	3	0	0	Normal	1,28
21:28:57	111	3	0	0	Normal	1,11
21:29:10	140	3	0	0	Normal	1,4
21:29:22	92	3	0	0	Normal	0,92
21:29:35	142	3	0	-1	Normal	1,42
21:29:48	84	3	0	0	Normal	0,84
21:30:00	138	3	0	0	Normal	1,38
21:30:13	103	4	0	0	Normal	1,03
21:30:26	29	3	0	0	Normal	0,29
21:30:38	20	2	0	0	Normal	0,2
21:30:51	30	0	0	0	Part did not grow.	0,3
21:31:04	19	0	0	0	Alert!!! ERROR detected. Check printer	0,19
21:31:16	18	0	0	0	Alert!!! ERROR detected. Check printer	0,18
21:31:29	37	0	0	0	Alert!!! ERROR detected. Check printer	0,37
21:31:42	19	0	0	0	Alert!!! ERROR detected. Check printer	0,19
21:31:54	20	0	0	0	Alert!!! ERROR detected. Check printer	0,2
21:32:07	9	0	0	0	Alert!!! ERROR detected. Check printer	0,09
21:32:20	21	0	0	0	Alert!!! ERROR detected. Check printer	0,21
21:32:32	15	0	0	0	Alert!!! ERROR detected. Check printer	0,15

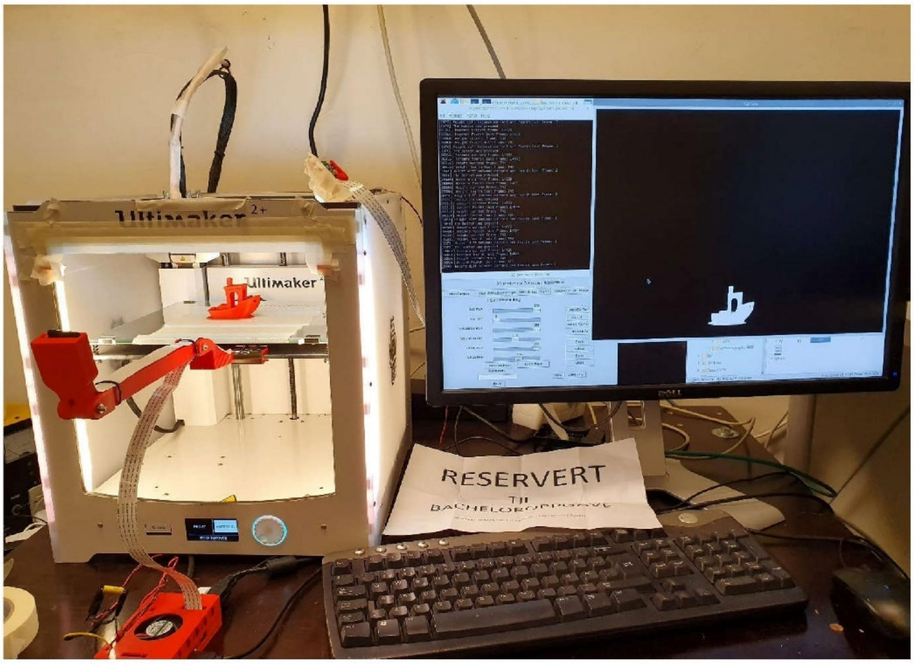
3 Cubes Shifted

Time	Diff	Heig	Left (	Righ	Status	Diff Normalized
21:35:51	315	0	25	-1	Vertical shift, Left side.	3,15
21:36:04	107	0	0	-1	Normal	1,07
21:36:17	108	3	0	0	Normal	1,08
21:36:29	108	3	1	0	Normal	1,08
21:36:42	104	3	0	0	Normal	1,04
21:36:55	146	3	0	0	Normal	1,46
21:37:07	87	3	0	0	Normal	0,87
21:37:20	109	3	0	-1	Normal	1,09
21:37:33	85	3	0	1	Normal	0,85
21:37:45	146	3	0	-1	Normal	1,46
21:37:58	136	3	0	0	Normal	1,36
21:38:11	116	3	0	0	Normal	1,16
21:38:23	113	3	0	0	Normal	1,13
21:38:36	118	3	0	0	Normal	1,18
21:38:49	125	3	0	0	Normal	1,25
21:39:02	145	3	0	0	Normal	1,45
21:39:14	1442	7	-123	-128	Vertical shift, Left side. Vertical shift, Right side.	14,42
21:39:27	135	6	79	85	Alert!!! ERROR detected. Check printer	1,35
21:39:40	861	5	-77	-85	Alert!!! ERROR detected. Check printer	8,61
21:39:52	613	7	121	114	Alert!!! ERROR detected. Check printer	6,13
21:40:06	96	7	0	-2	Alert!!! ERROR detected. Check printer	0,96

B: Data from tests

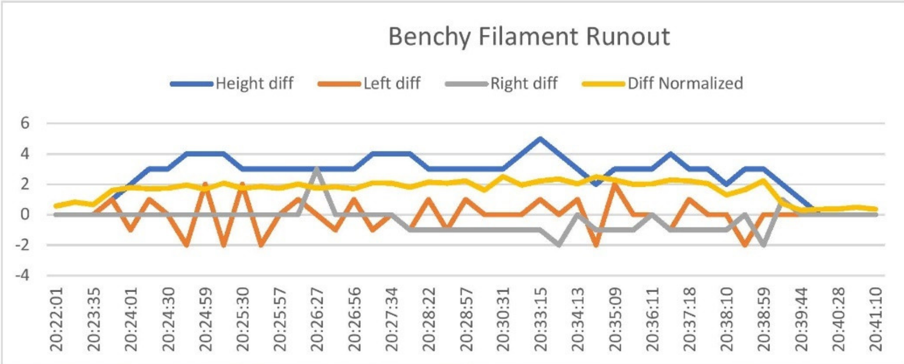
Benchy Charts

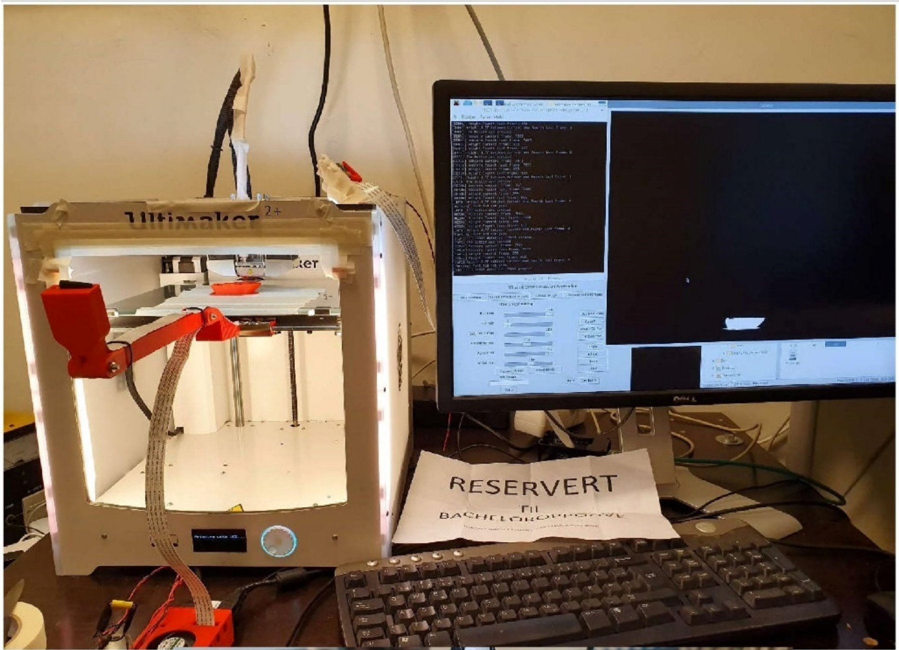


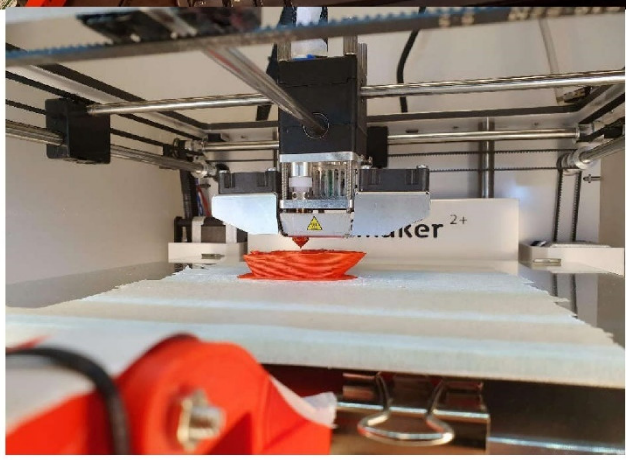


B: Data from tests

Benchy Charts

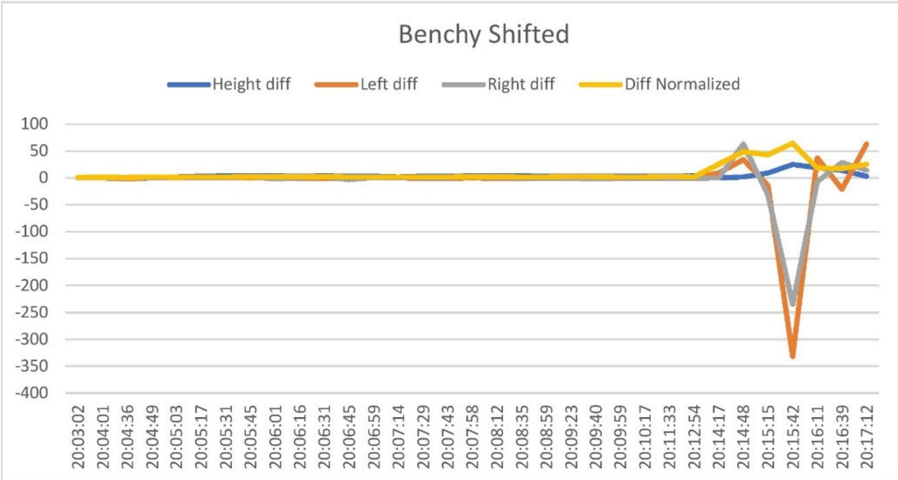


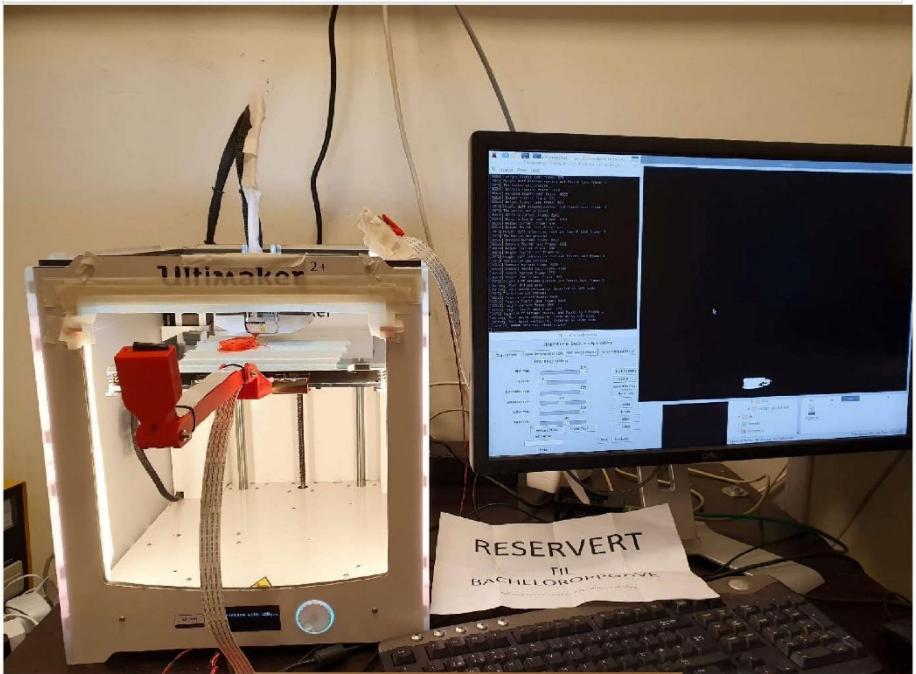


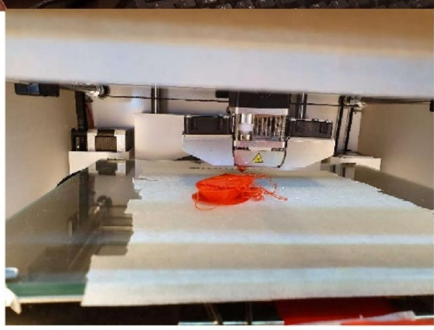


B: Data from tests

Benchy Charts







Page 11 of 28

Benchy Complete

Time	Diff	Height diff	Left diff	Right diff	Status	Diff Norma
19:01:25	198	0	0	-2	Normal	1,98
19:02:24	49	0	0	1	Normal	0,49
19:02:59	69	0	0	0	Part did not grow.	0,69
19:03:12	115	2	-1	-1	Normal	1,15
19:03:26	173	2	1	1	Normal	1,73
19:03:40	179	3	0	0	Normal	1,79
19:03:54	186	3	0	0	Normal	1,86
19:04:08	228	3	-1	0	Normal	2,28
19:04:23	198	3	1	0	Normal	1,98
19:04:39	199	3	0	0	Normal	1,99
19:04:54	212	3	0	-1	Normal	2,12
19:05:07	158	3	-1	0	Normal	1,58
19:05:22	200	4	1	1	Normal	2
19:05:36	164	4	-1	-1	Normal	1,64
19:05:52	220	4	1	1	Normal	2,2
19:06:06	170	3	-1	3	Normal	1,7
19:06:20	191	3	0	-3	Normal	1,91
19:06:35	197	3	0	3	Normal	1,97
19:06:58	180	3	0	-3	Normal	1,8
19:07:22	165	3	0	0	Normal	1,65
19:07:46	222	3	0	1	Normal	2,22
19:08:03	221	3	1	-2	Normal	2,21
19:08:22	186	3	0	-1	Normal	1,86
19:08:39	209	4	0	-1	Normal	2,09
19:09:56	300	4	0	-1	Normal	3
19:11:17	205	5	-1	-1	Normal	2,05
19:12:39	247	5	1	-1	Normal	2,47
19:13:11	234	4	0	-1	Normal	2,34
19:13:37	186	3	0	-1	Normal	1,86
19:14:05	258	2	0	-1	Normal	2,58
19:14:33	209	3	0	-1	Normal	2,09
19:15:02	235	3	0	0	Normal	2,35
19:15:35	180	3	0	-1	Normal	1,8
19:16:09	226	3	0	-2	Normal	2,26
19:16:42	219	3	0	0	Normal	2,19
19:17:08	185	4	0	-1	Normal	1,85
19:17:34	187	4	0	-1	Normal	1,87
19:17:59	117	4	0	-1	Normal	1,17
19:18:23	161	2	0	0	Normal	1,61
19:18:45	159	2	-1	-1	Normal	1,59
19:19:08	177	2	1	0	Normal	1,77
19:19:31	175	2	0	-1	Normal	1,75
19:19:52	184	3	0	0	Normal	1,84
19:20:13	167	3	0	-1	Normal	1,67
19:20:34	188	4	0	-1	Normal	1,88
19:20:56	152	2	0	-1	Normal	1,52
19:21:23	173	2	0	0	Normal	1,73
19:21:51	159	2	0	0	Normal	1,59
19:22:24	207	3	0	-1	Normal	2,07

Benchy Complete

19:22:56	204	3	0	-1	Normal	2,04
19:23:29	159	3	0	0	Normal	1,59
19:23:53	117	3	0	0	Normal	1,17
19:24:17	140	3	0	-1	Normal	1,4
19:24:41	133	3	0	-1	Normal	1,33
19:25:05	159	3	0	0	Normal	1,59
19:25:28	185	3	0	0	Normal	1,85
19:25:50	151	3	0	0	Normal	1,51
19:26:13	143	3	0	-1	Normal	1,43
19:26:35	137	3	0	0	Normal	1,37
19:26:57	131	3	0	0	Normal	1,31
19:27:18	153	3	0	-1	Normal	1,53
19:27:39	125	3	0	0	Normal	1,25
19:28:00	104	3	0	-1	Normal	1,04
19:28:20	137	3	0	0	Normal	1,37
19:28:44	120	3	0	0	Normal	1,2
19:29:08	135	3	0	0	Normal	1,35
19:29:32	96	3	0	0	Normal	0,96
19:29:54	135	3	0	-1	Normal	1,35
19:30:16	123	3	0	0	Normal	1,23
19:30:37	114	3	0	0	Normal	1,14
19:30:58	105	3	0	0	Normal	1,05
19:31:20	143	3	0	0	Normal	1,43
19:31:41	124	3	0	0	Normal	1,24
19:32:03	160	3	0	0	Normal	1,6
19:32:28	129	3	0	0	Normal	1,29
19:32:49	86	2	0	-1	Normal	0,86
19:33:09	119	2	0	0	Normal	1,19
19:33:29	122	2	0	0	Normal	1,22
19:33:49	107	3	0	-1	Normal	1,07
19:34:09	116	3	0	0	Normal	1,16
19:34:28	94	3	0	-1	Normal	0,94
19:34:45	110	3	0	0	Normal	1,1
19:35:00	99	3	0	0	Normal	0,99
19:35:15	85	3	0	0	Normal	0,85
19:35:30	87	3	0	0	Normal	0,87
19:35:45	92	3	0	0	Normal	0,92
19:35:59	87	3	0	0	Normal	0,87
19:36:13	93	3	0	0	Normal	0,93
19:36:26	76	3	0	0	Normal	0,76
19:36:38	172	3	0	0	Normal	1,72
19:36:50	76	3	0	0	Normal	0,76
19:37:01	74	2	0	0	Normal	0,74
19:37:14	85	3	0	0	Normal	0,85
19:37:26	92	3	0	0	Normal	0,92
19:37:38	76	4	0	0	Normal	0,76
19:37:50	88	3	0	0	Normal	0,88
19:38:02	63	2	0	0	Normal	0,63
19:38:14	75	3	0	0	Normal	0,75
19:38:26	68	3	0	0	Normal	0,68

Benchy Complete

19:38:39	65	3	0	0	Normal	0,65
19:38:50	93	2	0	0	Normal	0,93
19:39:02	69	2	0	0	Normal	0,69
19:39:15	118	4	0	0	Normal	1,18
19:39:29	81	3	0	0	Normal	0,81
19:39:44	108	4	0	0	Normal	1,08
19:39:58	118	2	0	0	Normal	1,18
19:40:12	107	3	0	0	Normal	1,07
19:40:26	116	1	0	0	Normal	1,16
19:40:38	122	2	0	0	Normal	1,22
19:40:51	125	3	0	0	Normal	1,25
19:41:02	124	4	0	0	Normal	1,24
19:41:14	124	4	0	0	Normal	1,24
19:41:25	193	3	0	-1	Normal	1,93
19:41:38	107	4	0	1	Normal	1,07
19:41:52	110	3	0	0	Normal	1,1
19:42:05	113	2	0	0	Normal	1,13
19:42:19	186	1	0	0	Normal	1,86
19:42:34	122	3	0	0	Normal	1,22
19:42:48	143	4	0	0	Normal	1,43
19:43:02	113	4	0	0	Normal	1,13
19:43:15	139	2	0	0	Normal	1,39
19:43:52	132	2	0	0	Normal	1,32
19:44:28	150	2	0	0	Normal	1,5
19:45:06	164	3	0	0	Normal	1,64
19:45:26	118	3	0	0	Normal	1,18
19:45:47	147	4	0	0	Normal	1,47
19:46:11	143	3	0	0	Normal	1,43
19:46:36	113	3	0	0	Normal	1,13
19:47:02	165	2	0	0	Normal	1,65
19:47:31	105	3	0	0	Normal	1,05
19:47:56	79	3	0	0	Normal	0,79
19:48:16	93	3	0	0	Normal	0,93
19:48:29	62	3	0	0	Normal	0,62
19:48:39	71	3	0	0	Normal	0,71
19:48:48	74	3	0	0	Normal	0,74
19:48:58	76	3	0	0	Normal	0,76
19:49:07	60	3	0	0	Normal	0,6
19:49:18	66	3	0	0	Normal	0,66
19:49:26	82	3	0	-1	Normal	0,82
19:49:36	66	2	0	1	Normal	0,66
19:49:45	86	2	0	0	Normal	0,86
19:49:55	97	2	0	-1	Normal	0,97
19:50:04	113	4	0	1	Normal	1,13
19:50:14	121	5	0	0	Normal	1,21
19:50:23	72	4	0	0	Normal	0,72
19:50:33	64	3	0	0	Normal	0,64
19:50:43	83	2	0	0	Normal	0,83
19:50:52	80	3	0	0	Normal	0,8
19:51:02	76	3	0	0	Normal	0,76

Benchy Complete

19:51:12	90	3	0	0	Normal	0,9
19:51:21	72	3	0	0	Normal	0,72
19:51:31	73	3	0	0	Normal	0,73
19:51:40	79	2	0	0	Normal	0,79
19:51:50	263	2	0	0	Normal	2,63
19:52:00	278	2	0	0	Normal	2,78
19:52:09	96	3	-1	0	Normal	0,96
19:52:19	91	3	1	0	Normal	0,91
19:52:29	74	3	0	0	Normal	0,74

B: Data from tests

Benchy Filament Runout

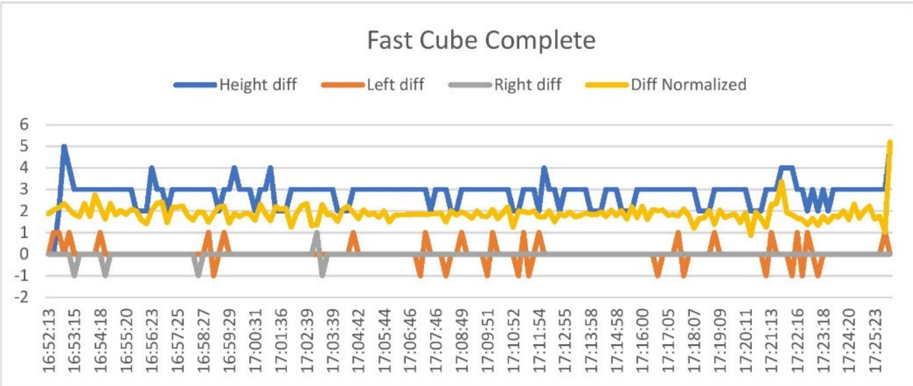
Time	Diff	Height dif	Left diff	Right c	Status	Diff Normal
20:22:01	57	0	0	0	Normal	0,57
20:23:00	82	0	0	0	Normal	0,82
20:23:35	66	0	0	0	Part did not grow.	0,66
20:23:48	159	1	1	0	Normal	1,59
20:24:01	179	2	-1	0	Normal	1,79
20:24:16	171	3	1	0	Normal	1,71
20:24:30	173	3	0	0	Normal	1,73
20:24:43	194	4	-2	0	Normal	1,94
20:24:59	167	4	2	0	Normal	1,67
20:25:15	208	4	-2	0	Normal	2,08
20:25:30	174	3	2	0	Normal	1,74
20:25:43	185	3	-2	0	Normal	1,85
20:25:57	173	3	0	0	Normal	1,73
20:26:12	201	3	1	0	Normal	2,01
20:26:27	174	3	0	3	Normal	1,74
20:26:42	183	3	-1	0	Normal	1,83
20:26:56	171	3	1	0	Normal	1,71
20:27:11	209	4	-1	0	Normal	2,09
20:27:34	208	4	0	0	Normal	2,08
20:27:58	180	4	-1	-1	Normal	1,8
20:28:22	213	3	1	-1	Normal	2,13
20:28:39	207	3	-1	-1	Normal	2,07
20:28:57	221	3	1	-1	Normal	2,21
20:29:15	160	3	0	-1	Normal	1,6
20:30:31	250	3	0	-1	Normal	2,5
20:31:52	195	4	0	-1	Normal	1,95
20:33:15	221	5	1	-1	Normal	2,21
20:33:47	235	4	0	-2	Normal	2,35
20:34:13	202	3	1	0	Normal	2,02
20:34:41	249	2	-2	-1	Normal	2,49
20:35:09	228	3	2	-1	Normal	2,28
20:35:38	200	3	0	-1	Normal	2
20:36:11	202	3	0	0	Normal	2,02
20:36:45	228	4	-1	-1	Normal	2,28
20:37:18	221	3	1	-1	Normal	2,21
20:37:45	204	3	0	-1	Normal	2,04
20:38:10	128	2	0	-1	Normal	1,28
20:38:35	164	3	-2	0	Normal	1,64
20:38:59	223	3	0	-2	Normal	2,23
20:39:21	73	2	0	1	Normal	0,73
20:39:44	29	1	0	0	Normal	0,29
20:40:07	35	0	0	0	Part did not grow.	0,35
20:40:28	38	0	0	0	Alert!!! ERROR detected. Check printer	0,38
20:40:48	48	0	0	0	Alert!!! ERROR detected. Check printer	0,48
20:41:10	36	0	0	0	Alert!!! ERROR detected. Check printer	0,36

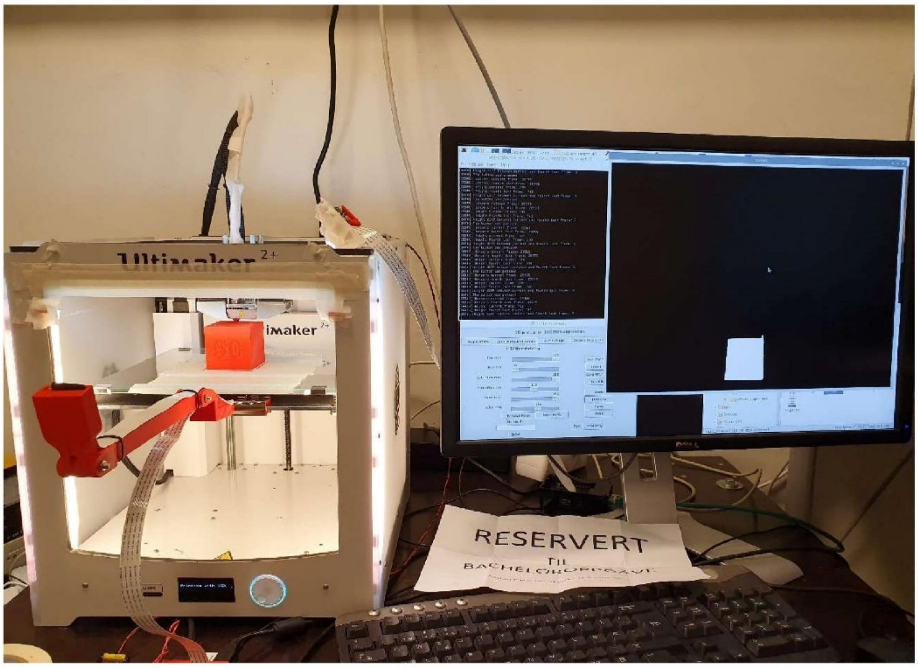
Benchy Shifted

Time	Diff	Height	Left dif	Right d	Status	Diff Norma
20:03:02	78	0	0	0	Normal	0,78
20:04:01	119	0	0	0	Normal	1,19
20:04:36	111	0	-2	0	Part did not grow.	1,11
20:04:49	177	1	0	0	Normal	1,77
20:05:03	152	2	0	0	Normal	1,52
20:05:17	173	3	0	0	Normal	1,73
20:05:31	188	4	0	0	Normal	1,88
20:05:45	244	4	0	2	Normal	2,44
20:06:01	214	4	2	-2	Normal	2,14
20:06:16	211	3	-1	0	Normal	2,11
20:06:31	208	4	-1	2	Normal	2,08
20:06:45	165	3	0	-3	Normal	1,65
20:06:59	165	3	2	1	Normal	1,65
20:07:14	179	2	0	0	Normal	1,79
20:07:29	167	3	-1	0	Normal	1,67
20:07:43	183	3	-1	0	Normal	1,83
20:07:58	213	4	0	2	Normal	2,13
20:08:12	225	4	-1	0	Normal	2,25
20:08:35	183	4	-1	0	Normal	1,83
20:08:59	169	3	0	-1	Normal	1,69
20:09:23	265	3	2	-1	Normal	2,65
20:09:40	206	3	0	-2	Normal	2,06
20:09:59	183	3	0	-1	Normal	1,83
20:10:17	169	3	0	-1	Normal	1,69
20:11:33	277	3	0	-1	Normal	2,77
20:12:54	184	4	0	-1	Normal	1,84
20:14:17	2518	0	9	0	Part did not grow. Vertical shift, Left side.	25,18
20:14:48	4849	2	34	63	Alert!!! ERROR detected. Check printer	48,49
20:15:15	4299	9	-14	-32	Alert!!! ERROR detected. Check printer	42,99
20:15:42	6452	25	-332	-235	Alert!!! ERROR detected. Check printer	64,52
20:16:11	1849	19	37	-7	Alert!!! ERROR detected. Check printer	18,49
20:16:39	1729	14	-21	29	Alert!!! ERROR detected. Check printer	17,29
20:17:12	2558	3	63	14	Alert!!! ERROR detected. Check printer	25,58

B: Data from tests

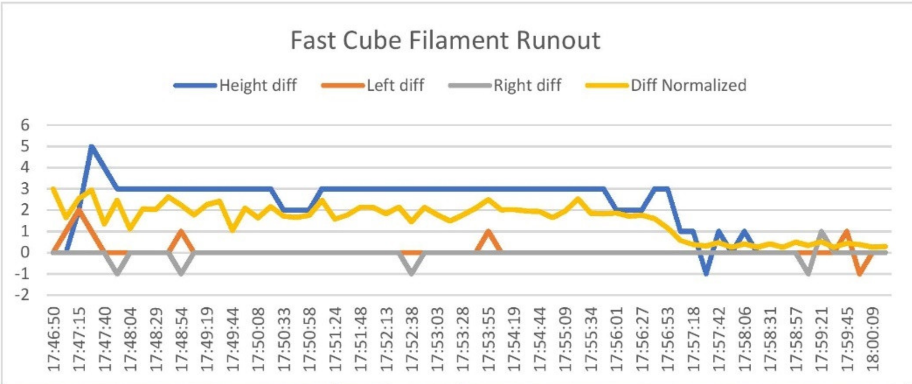
Fast Cube Charts

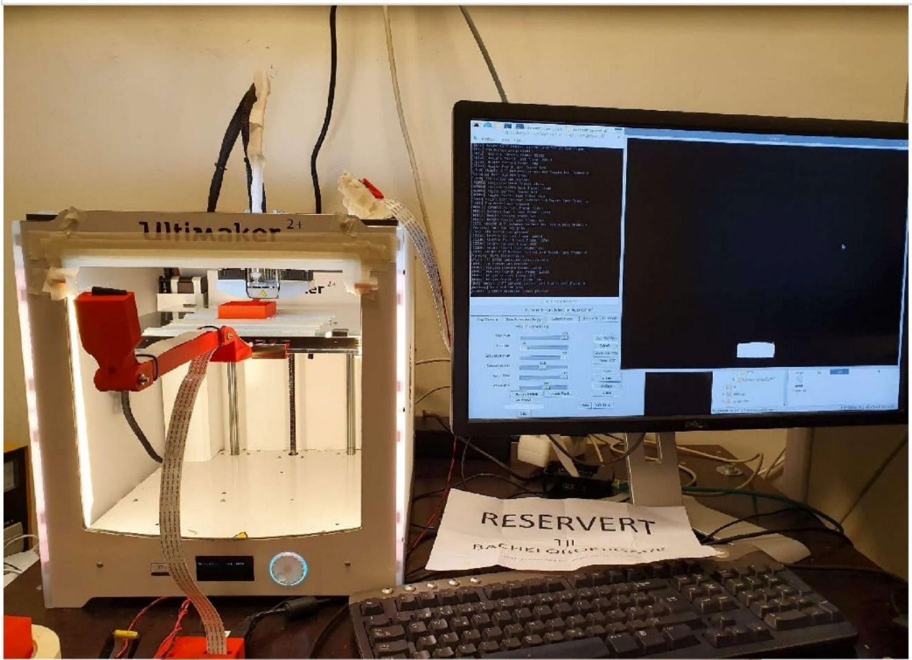


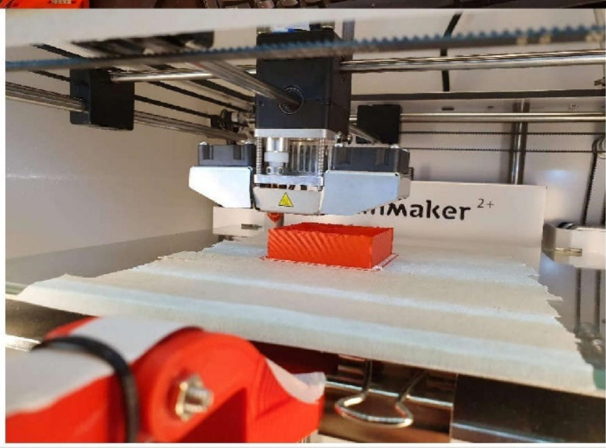


B: Data from tests

Fast Cube Charts

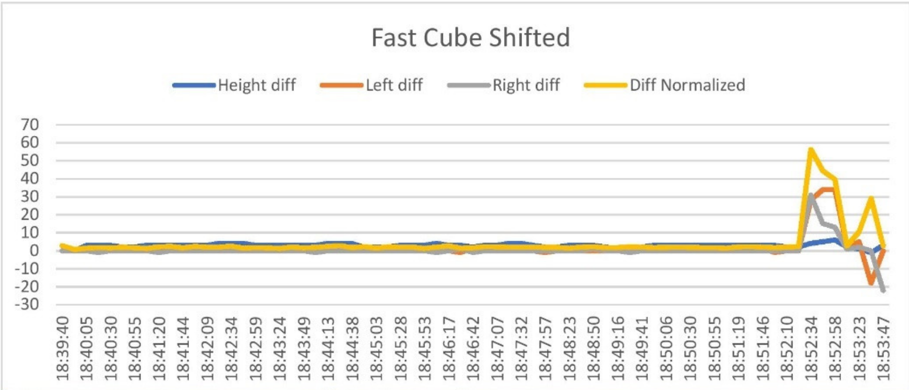


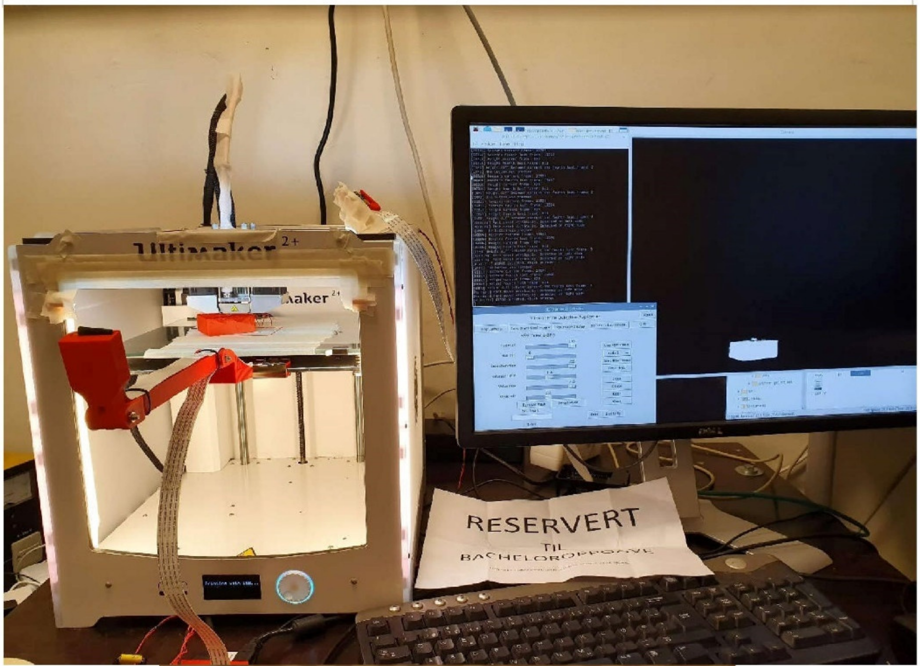


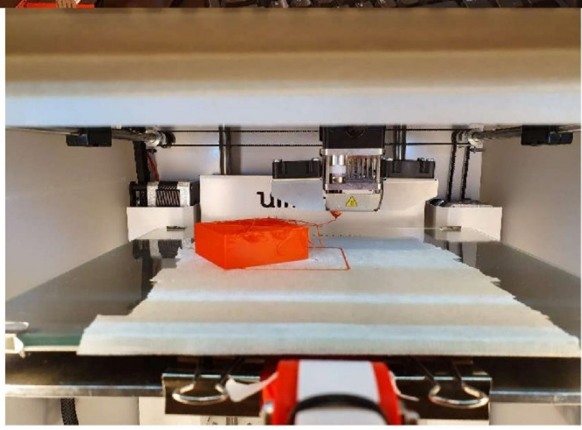


B: Data from tests

Fast Cube Charts







B: Data from tests

Fast Cube Complete

Time	Diff	Height diff	Left diff	Right diff	Status	Diff Norma
16:52:13	186	0	0	0	Normal	1,86
16:52:26	205	0	1	0	Normal	2,05
16:52:38	215	2	1	0	Normal	2,15
16:52:50	234	5	0	0	Normal	2,34
16:53:03	209	4	1	0	Normal	2,09
16:53:15	184	3	0	-1	Normal	1,84
16:53:28	173	3	0	0	Normal	1,73
16:53:40	234	3	0	0	Normal	2,34
16:53:53	175	3	0	0	Normal	1,75
16:54:05	275	3	0	0	Normal	2,75
16:54:18	216	3	1	0	Normal	2,16
16:54:30	162	3	0	-1	Normal	1,62
16:54:42	235	3	0	0	Normal	2,35
16:54:55	182	3	0	0	Normal	1,82
16:55:08	203	3	0	0	Normal	2,03
16:55:20	185	3	0	0	Normal	1,85
16:55:33	207	3	0	0	Normal	2,07
16:55:45	200	2	0	0	Normal	2
16:55:58	158	2	0	0	Normal	1,58
16:56:10	141	2	0	0	Normal	1,41
16:56:23	203	4	0	0	Normal	2,03
16:56:35	236	3	0	0	Normal	2,36
16:56:48	241	3	0	0	Normal	2,41
16:57:00	146	2	0	0	Normal	1,46
16:57:12	217	3	0	0	Normal	2,17
16:57:25	216	3	0	0	Normal	2,16
16:57:37	223	3	0	0	Normal	2,23
16:57:50	177	3	0	0	Normal	1,77
16:58:02	158	3	0	0	Normal	1,58
16:58:14	196	3	0	-1	Normal	1,96
16:58:27	194	3	0	0	Normal	1,94
16:58:39	148	3	1	0	Normal	1,48
16:58:52	187	3	-1	0	Normal	1,87
16:59:04	219	2	0	0	Normal	2,19
16:59:16	223	3	1	0	Normal	2,23
16:59:29	144	3	0	0	Normal	1,44
16:59:41	188	4	0	0	Normal	1,88
16:59:53	173	3	0	0	Normal	1,73
17:00:06	189	3	0	0	Normal	1,89
17:00:19	186	3	0	0	Normal	1,86
17:00:31	158	2	0	0	Normal	1,58
17:00:43	231	3	0	0	Normal	2,31
17:00:56	188	3	0	0	Normal	1,88
17:01:10	155	4	0	0	Normal	1,55
17:01:23	220	2	0	0	Normal	2,2
17:01:36	206	2	0	0	Normal	2,06
17:01:49	213	2	0	0	Normal	2,13
17:02:01	125	3	0	0	Normal	1,25
17:02:14	188	3	0	0	Normal	1,88

Fast Cube Complete

17:02:27	223	3	0	0	Normal	2,23
17:02:39	234	3	0	0	Normal	2,34
17:02:51	133	3	0	0	Normal	1,33
17:03:03	136	3	0	1	Normal	1,36
17:03:15	230	3	0	-1	Normal	2,3
17:03:27	183	3	0	0	Normal	1,83
17:03:39	182	3	0	0	Normal	1,82
17:03:52	152	2	0	0	Normal	1,52
17:04:05	209	2	0	0	Normal	2,09
17:04:18	221	2	0	0	Normal	2,21
17:04:30	194	3	1	0	Normal	1,94
17:04:42	164	3	0	0	Normal	1,64
17:04:55	205	3	0	0	Normal	2,05
17:05:07	181	3	0	0	Normal	1,81
17:05:19	190	3	0	0	Normal	1,9
17:05:31	171	3	0	0	Normal	1,71
17:05:44	199	3	0	0	Normal	1,99
17:05:56	150	3	0	0	Normal	1,5
17:06:08	179	3	0	0	Normal	1,79
17:06:22	182	3	0	0	Normal	1,82
17:06:34	182	3	0	0	Normal	1,82
17:06:46	183	3	0	0	Normal	1,83
17:06:58	185	3	0	0	Normal	1,85
17:07:10	186	3	-1	0	Normal	1,86
17:07:22	184	3	1	0	Normal	1,84
17:07:34	187	2	0	0	Normal	1,87
17:07:46	187	3	0	0	Normal	1,87
17:07:59	190	3	0	0	Normal	1,9
17:08:11	149	3	-1	0	Normal	1,49
17:08:24	200	2	0	0	Normal	2
17:08:37	180	2	0	0	Normal	1,8
17:08:49	196	3	1	0	Normal	1,96
17:09:02	185	3	0	0	Normal	1,85
17:09:14	167	3	0	0	Normal	1,67
17:09:26	199	3	0	0	Normal	1,99
17:09:38	176	3	0	0	Normal	1,76
17:09:51	175	3	-1	0	Normal	1,75
17:10:03	207	3	1	0	Normal	2,07
17:10:15	164	3	0	0	Normal	1,64
17:10:27	191	3	0	0	Normal	1,91
17:10:40	219	3	0	0	Normal	2,19
17:10:52	125	2	0	0	Normal	1,25
17:11:04	199	2	-1	0	Normal	1,99
17:11:16	197	3	1	0	Normal	1,97
17:11:29	190	3	-1	0	Normal	1,9
17:11:42	201	3	0	0	Normal	2,01
17:11:54	173	2	1	0	Normal	1,73
17:12:06	171	4	0	0	Normal	1,71
17:12:19	201	3	0	0	Normal	2,01
17:12:30	149	3	0	0	Normal	1,49

Fast Cube Complete

17:12:43	188	2	0	0	Normal	1,88
17:12:55	177	3	0	0	Normal	1,77
17:13:07	195	3	0	0	Normal	1,95
17:13:20	173	3	0	0	Normal	1,73
17:13:33	179	3	0	0	Normal	1,79
17:13:45	190	3	0	0	Normal	1,9
17:13:58	183	2	0	0	Normal	1,83
17:14:10	182	2	0	0	Normal	1,82
17:14:22	195	2	0	0	Normal	1,95
17:14:34	175	3	0	0	Normal	1,75
17:14:46	204	3	0	0	Normal	2,04
17:14:58	179	3	0	0	Normal	1,79
17:15:10	202	2	0	0	Normal	2,02
17:15:22	163	2	0	0	Normal	1,63
17:15:35	220	2	0	0	Normal	2,2
17:15:47	169	3	0	0	Normal	1,69
17:16:00	222	3	0	0	Normal	2,22
17:16:14	161	3	0	0	Normal	1,61
17:16:27	205	3	0	0	Normal	2,05
17:16:40	202	3	-1	0	Normal	2,02
17:16:52	204	3	0	0	Normal	2,04
17:17:05	180	3	0	0	Normal	1,8
17:17:17	183	3	0	0	Normal	1,83
17:17:30	178	3	1	0	Normal	1,78
17:17:42	209	3	-1	0	Normal	2,09
17:17:54	179	3	0	0	Normal	1,79
17:18:07	120	3	0	0	Normal	1,2
17:18:19	164	2	0	0	Normal	1,64
17:18:32	169	2	0	0	Normal	1,69
17:18:44	203	2	0	0	Normal	2,03
17:18:56	138	3	1	0	Normal	1,38
17:19:09	204	3	0	0	Normal	2,04
17:19:21	170	3	0	0	Normal	1,7
17:19:34	180	3	0	0	Normal	1,8
17:19:46	186	3	0	0	Normal	1,86
17:19:58	147	3	0	0	Normal	1,47
17:20:11	193	3	0	0	Normal	1,93
17:20:23	87	2	0	0	Normal	0,87
17:20:36	195	2	0	0	Normal	1,95
17:20:48	164	2	0	0	Normal	1,64
17:21:01	125	3	-1	0	Normal	1,25
17:21:13	229	3	1	0	Normal	2,29
17:21:27	235	3	0	0	Normal	2,35
17:21:39	336	4	0	0	Normal	3,36
17:21:52	192	4	0	0	Normal	1,92
17:22:04	182	4	-1	0	Normal	1,82
17:22:16	166	3	1	0	Normal	1,66
17:22:29	161	3	-1	0	Normal	1,61
17:22:41	136	2	1	0	Normal	1,36
17:22:53	165	3	0	0	Normal	1,65

B: Data from tests

Fast Cube Complete

17:23:06	134	2	-1	0	Normal	1,34
17:23:18	174	3	0	0	Normal	1,74
17:23:31	148	2	0	0	Normal	1,48
17:23:43	178	3	0	0	Normal	1,78
17:23:55	172	3	0	0	Normal	1,72
17:24:08	201	3	0	0	Normal	2,01
17:24:20	160	3	0	0	Normal	1,6
17:24:33	232	3	0	0	Normal	2,32
17:24:45	167	3	0	0	Normal	1,67
17:24:58	200	3	0	0	Normal	2
17:25:10	222	3	0	0	Normal	2,22
17:25:23	163	3	0	0	Normal	1,63
17:25:35	176	3	0	0	Normal	1,76
17:25:48	102	3	1	0	Normal	1,02
17:29:53	520	5	0	0	Normal	5,2

Fast Cube Filament Runout

Time	Diff	Height c	Left di	Right diff	Status	Diff Norma
17:46:50	299	0	0	0	Normal	2,99
17:47:02	165	0	1	0	Normal	1,65
17:47:15	254	2	2	0	Normal	2,54
17:47:27	294	5	1	0	Normal	2,94
17:47:40	135	4	0	0	Normal	1,35
17:47:52	246	3	0	-1	Normal	2,46
17:48:04	112	3	0	0	Normal	1,12
17:48:17	206	3	0	0	Normal	2,06
17:48:29	203	3	0	0	Normal	2,03
17:48:42	262	3	0	0	Normal	2,62
17:48:54	223	3	1	-1	Normal	2,23
17:49:07	177	3	0	0	Normal	1,77
17:49:19	226	3	0	0	Normal	2,26
17:49:31	242	3	0	0	Normal	2,42
17:49:44	105	3	0	0	Normal	1,05
17:49:56	210	3	0	0	Normal	2,1
17:50:08	164	3	0	0	Normal	1,64
17:50:21	216	3	0	0	Normal	2,16
17:50:33	172	2	0	0	Normal	1,72
17:50:46	166	2	0	0	Normal	1,66
17:50:58	176	2	0	0	Normal	1,76
17:51:11	246	3	0	0	Normal	2,46
17:51:24	158	3	0	0	Normal	1,58
17:51:36	177	3	0	0	Normal	1,77
17:51:48	214	3	0	0	Normal	2,14
17:52:01	213	3	0	0	Normal	2,13
17:52:13	183	3	0	0	Normal	1,83
17:52:26	214	3	0	0	Normal	2,14
17:52:38	145	3	0	-1	Normal	1,45
17:52:51	213	3	0	0	Normal	2,13
17:53:03	178	3	0	0	Normal	1,78
17:53:16	148	3	0	0	Normal	1,48
17:53:28	177	3	0	0	Normal	1,77
17:53:42	210	3	0	0	Normal	2,1
17:53:55	250	3	1	0	Normal	2,5
17:54:07	201	3	0	0	Normal	2,01
17:54:19	203	3	0	0	Normal	2,03
17:54:32	196	3	0	0	Normal	1,96
17:54:44	193	3	0	0	Normal	1,93
17:54:56	165	3	0	0	Normal	1,65
17:55:09	196	3	0	0	Normal	1,96
17:55:21	253	3	0	0	Normal	2,53
17:55:34	185	3	0	0	Normal	1,85
17:55:48	183	3	0	0	Normal	1,83
17:56:01	186	2	0	0	Normal	1,86
17:56:15	171	2	0	0	Normal	1,71
17:56:27	176	2	0	0	Normal	1,76
17:56:40	159	3	0	0	Normal	1,59
17:56:53	116	3	0	0	Normal	1,16

Fast Cube Filament Runout

17:57:06	58	1	0	0	Normal	0,58
17:57:18	38	1	0	0	Normal	0,38
17:57:30	31	-1	0	0	Part did not grow.	0,31
17:57:42	47	1	0	0	Normal	0,47
17:57:54	25	0	0	0	Part did not grow.	0,25
17:58:06	41	1	0	0	Normal	0,41
17:58:18	26	0	0	0	Part did not grow.	0,26
17:58:31	42	0	0	0	Alert!!! ERROR detected. Check printer	0,42
17:58:44	25	0	0	0	Alert!!! ERROR detected. Check printer	0,25
17:58:57	50	0	0	0	Alert!!! ERROR detected. Check printer	0,5
17:59:09	33	0	0	-1	Alert!!! ERROR detected. Check printer	0,33
17:59:21	51	0	0	1	Alert!!! ERROR detected. Check printer	0,51
17:59:33	26	0	0	0	Alert!!! ERROR detected. Check printer	0,26
17:59:45	44	0	1	0	Alert!!! ERROR detected. Check printer	0,44
17:59:57	38	0	-1	0	Alert!!! ERROR detected. Check printer	0,38
18:00:09	26	0	0	0	Alert!!! ERROR detected. Check printer	0,26
18:00:22	29	0	0	0	Alert!!! ERROR detected. Check printer	0,29

Fast Cube Shifted

Time	Diff	Heigl	Left di	Right	Status	Diff Norma
18:39:40	287	0	0	0	Normal	2,87
18:39:53	68	0	1	0	Normal	0,68
18:40:05	155	3	0	0	Normal	1,55
18:40:17	180	3	1	-1	Normal	1,8
18:40:30	164	3	1	0	Normal	1,64
18:40:42	190	2	0	0	Normal	1,9
18:40:55	168	2	0	0	Normal	1,68
18:41:07	138	3	0	0	Normal	1,38
18:41:20	215	3	0	-1	Normal	2,15
18:41:32	240	3	0	0	Normal	2,4
18:41:44	165	3	0	0	Normal	1,65
18:41:57	250	3	1	0	Normal	2,5
18:42:09	206	3	0	0	Normal	2,06
18:42:22	202	4	0	0	Normal	2,02
18:42:34	252	4	0	0	Normal	2,52
18:42:47	148	4	0	0	Normal	1,48
18:42:59	176	3	0	0	Normal	1,76
18:43:11	165	3	0	0	Normal	1,65
18:43:24	132	3	0	0	Normal	1,32
18:43:36	206	3	0	0	Normal	2,06
18:43:49	157	3	0	0	Normal	1,57
18:44:01	186	3	0	-1	Normal	1,86
18:44:13	247	4	0	0	Normal	2,47
18:44:26	282	4	0	0	Normal	2,82
18:44:38	178	4	0	0	Normal	1,78
18:44:50	233	2	0	0	Normal	2,33
18:45:03	152	2	0	0	Normal	1,52
18:45:15	218	2	0	0	Normal	2,18
18:45:28	199	3	0	0	Normal	1,99
18:45:40	209	3	0	0	Normal	2,09
18:45:53	137	3	0	0	Normal	1,37
18:46:05	207	4	1	-1	Normal	2,07
18:46:17	267	3	0	0	Normal	2,67
18:46:30	154	3	-1	1	Normal	1,54
18:46:42	168	2	1	-1	Normal	1,68
18:46:55	226	3	0	0	Normal	2,26
18:47:07	214	3	0	0	Normal	2,14
18:47:20	202	4	0	0	Normal	2,02
18:47:32	201	4	0	0	Normal	2,01
18:47:45	199	3	0	0	Normal	1,99
18:47:57	187	2	-1	0	Normal	1,87
18:48:09	193	2	0	0	Normal	1,93
18:48:23	153	3	1	0	Normal	1,53
18:48:37	197	3	0	0	Normal	1,97
18:48:50	229	3	0	1	Normal	2,29
18:49:03	175	2	0	0	Normal	1,75
18:49:16	175	1	0	0	Normal	1,75
18:49:28	212	1	0	-1	Normal	2,12
18:49:41	208	2	0	0	Normal	2,08

Page 27 of 28

Fast Cube Shifted

18:49:54	172	3	0	0	Normal	1,72
18:50:06	189	3	0	0	Normal	1,89
18:50:18	197	3	0	0	Normal	1,97
18:50:30	192	3	0	0	Normal	1,92
18:50:43	175	3	0	0	Normal	1,75
18:50:55	175	3	0	0	Normal	1,75
18:51:07	155	3	0	0	Normal	1,55
18:51:19	219	3	0	0	Normal	2,19
18:51:33	229	3	0	0	Normal	2,29
18:51:46	205	3	1	0	Normal	2,05
18:51:58	179	3	-1	0	Normal	1,79
18:52:10	199	2	0	0	Normal	1,99
18:52:22	228	2	0	0	Normal	2,28
18:52:34	5632	4	28	31	Vertical shift, Left side. Vertical shift, Right side.	56,32
18:52:46	4440	5	34	15	Alert!!! ERROR detected. Check printer	44,4
18:52:58	3956	6	34	13	Alert!!! ERROR detected. Check printer	39,56
18:53:11	267	2	2	1	Alert!!! ERROR detected. Check printer	2,67
18:53:23	1051	1	5	2	Alert!!! ERROR detected. Check printer	10,51
18:53:35	2912	-1	-18	0	Alert!!! ERROR detected. Check printer	29,12
18:53:47	266	3	0	-22	Alert!!! ERROR detected. Check printer	2,66

NTNU
Norwegian University of
Science and Technology