Vision Sensing based Filament Clogging & Exhaustion Detection and Machine Interruption System using Serial Intrusion Model in Fused Deposition Modelling 3d Printers

K. Chandrakumar
Graduate Research Student
Bannari Amman Institute of Technology
Coimbatore, Tamilnadu, India - 638401
Email: chandrakumar3108@gmail.com

A. Naveensankar
PG Research Student
Bannari Amman Institute of Technology
Coimbatore, Tamilnadu, India - 638401
Email: naveensankar807@gmail.com

Over the past few years 3D printing, or Additive Manufacturing has been very prominent in the manufacturing industries. It has been widely adopted in enormous fields such as rapid prototyping and accelerated manufacturing, moulds for jewellery making, bioproduct engineering, tissue engineering and constructions. Inspite of all its leverage, it is still arduous in terms of time consumption and may take hours to few days for printing. Since the human presence is concerned for most of the errors whilst printing, in almost 90% of the printers clogging or exhaustion of filament due to various factors can lead to complete deterioration of the process. When the printing process is humongous, machine may run for several hours with clogged filament which leads to complete wastage of time, power and productivity. If we can able to automatically encounter the filament clogging as they occur in realtime and the process can be interrupted automatically at the immediate layer of printing so as to check the flow of filament from the nozzle to resume printing. This can save both the time and massive material wastage over printing. In this research work we implemented a camera module with series of Computer Vision techniques and image/video processing methods to distinguish the extruder with filament from the filament less extruder. Once the state of the extruder is analyzed printing process is interrupted (paused) using serial intrusion method and will be dynamically resumed if the filament flow is started anew. The results obtained using the proposed approach are impressive and further combination of this method with ambulatory camera module and explicit algorithms might be possible to deal with other prevailing printing errors in future.

1 Introduction

Technological advancement in the additive manufacturing era and the community development, keeps on spreading the wings of 3D printing technology to all engineering enthusiasts especially in Fused Deposition Modelling(FDM) method. This results in the development of industrial grade as well as DIY printers in the market. Today consumer grade 3D printers with the dimension of 50cm edge length are generally cubical. Due to consumer grade printer mechanics, the build area size may vary between 15*15*15 cm to 30*30*30 cm. Fused Deposition Modelling(FDM) is one of the most familiar consumer grade 3D printers available in the market. Based on this various technology in 3D printing or Additive manufacturing(AM) exists, from metal powder or ceramic positioned laser entering over laminated object modelling, adhesive based stereolithographic to thermoplastics.

FDM works on a simple mechanism and is cost efficient to manufacture as they dont require expensive and complex components. In these days, this is made possible due to the availability of colossal spare market. The foundation consists of 3 axes that are restrained by 3 stepper motors that are able to move countably based on command. Compared to stereolithographic, here we can reduce the granularity of object by affixed constellation which enables 3 degrees of latitude along this axis and besides from print-head to print bed.

For direct supervision of inhabitant, the printers are recommended to exploit rooms rather than workspace. The printing duration of FD machines can ranges upto 20 hours for large and complex objects alongside loaded filaments in the system. The main objective is to reduce the misprint as the filament flow is clogged or exhausted which consumes longer printing time without producing any effective object if left unchecked.

In order to pacify the problems user can place cameras [Video or web cams] in or at printer for remote supervision adequacy. Facilitate laser scanners, currents or thermography are other approaches of error detections which were previously in the development. In order to get the information on problems and errors the user has to watch the video constantly or in intervals and assess remotely on printing progress. This design is to support the user to detect the printing error by utilizing machine vision. The benefits of early detection of printer errors reduces material wastage and occupancy of printer resources. Constant surveillance is required for preventing completion of printing process and inappropriate or broken objects to reduce the time effort of the user.

To identify these failures video frames and template matching are presented in an in-line computer vision system which uses differential imaging, Normalized Root Mean Square and Structural Similarity Index Measure.

Computer Vision(CV) is a scientific field that deals with how computers can gain high-level understanding from digital images or videos. This field is capable of performing Image classification, Object Detection, Object Tracking, Sematic Segmentation, Instance Segmentation, Image Reconstruction and various image processing methods such as additive, differential, template matching, edge detection, blob detection, etc. Current generation processors and GPUs supports computationally intensive and repetitive tasks of computer vision technology, thus making the system more viable for high speed operations in terafops. Various open source libraries are out there in web, comprises of set of classes, methods and functions written in many languages to perform Computer Vision in efficient way. Among them, Python seems to be more vibrant due to its large community contributors and huge library support which made us ends up with that in this project.

2 Literature Review

There are some techniques and studies to scale down the failure modes of 3D printers. Nuchitprasitchai et al to detect incomplete print and nozzle block, which performs the interpretation method to figure out the challenges. Garanger et al enforce closed-loop system control for AM process to stiffness objects like leaf spring. Delli and Chang designed binary 3D printing to check the quality at critical stages during printing process, once the critical stages are identified, then quality check operation based on computer vision to be performed. Fastowicz and Okarma developed the texture analysis with Haralick texture feature calculated from a Gray Level Co-occurrence Matrix (GLCM). Cummings et al developed with the help of ultrasonic to manipulate the temperature of print bed during printing process, to detect the closed loop frame work control and to rectify the failures to filament bonding. And finally Heterogeneous sensors are developed by Rao et al to analysis the surface roughness of framework. These techniques mainly focuses on detecting the challenges and provides a solution which are not automated. Manual interruption is required for all these figured problems. In the work made by Aleksei L entitled with computer vision based layer wise analysis deals about layer wise analysis by tracking printer error and generates the appropriate printer action. It uses the camera module to capture images of side view mapping and virtual top view concept to analyse the system with algorithm. That solution fails to detect the filament flow error which makes confusion to figure out the exact issue occuring without the manual intervention.

3 Model Architecture

There are various solutions coming up in association hardware level but they are not efficient in full filling the required conditions. In order to gain efficient solution, software integration is always recommended. This solution also reduces many additional hardwares which needs to be integrated other than a camera module. Inspite of various issues in FDM printers such as filament clogging, wrapping, over extrusion, under extrusion, printing offset, oozing, temperature issue, deformed infill, support fail, filament bridging etc., we have identified three prevalent errors such as filament exhaustion, filament clogging, under or over extrusion & filament oozing.

Common occurrences of these problems exits due to below stated causes, Filament exhaustion occurs as a result of human negligence where operator fails to load the required amount of filament spool. Effortlessly we can avoid this with the aid of slicing software which usually estimates the amount of filament required (mostly in grams) for that specific printing process prior to printing. Sometime this issue occurs due to mechanical failure of extruding setup where pushing head loosen out or the guide holder get removed from its position. Another rare case is grinding of filament by feeder mechanism roller instead of feeding it, which occurs due to the cool hot end. Due to cool hot end the filament clogs at nozzle tip and results in grinding effect. Anyway for all these problems simple Infrared sensor integration serves the purpose, which will monitor the filament flow throughout the printing process without any means of human intervention.

But the vision based detection always serves accurate and viable solution, thus the same camera unit can also be utlilized for various other error class detections. Other issues such as under and over extrusion arises when the flow rate fails to function appropriately. If the flow rate becames higher than the stipulated level at specific layer then filament flows more than required amount results in over extrusion. In contrary when the flow rate becames lower than the stipulated level at specific layer then filament flows less than the required amount which results in under extrusion. This ends in improper layers causing the models failing to serve the purpose. It is very hard to detect these issues and to provide autonomous solutions mechanically. Therefore we implemented Computer vision algorithms in camera module to address all these issues in one go.

3.1 Machine Setup

For this novel work of vision based sensing experimental DIY printer Ender 3 from the manufacturer Creality has been utlilized.The below figure shows the Ender machine and camera unit installed in it.

The camera unit used here is Microsoft LifeCam HD-3000 Web Camera HD 720P model with the following specification for ease functionality.

Connectivity Technology	Wired
Dimensions (w x d x h)	4.5 cm x 3.9 cm x 10.9 cm

Table 1. Camera Specifications

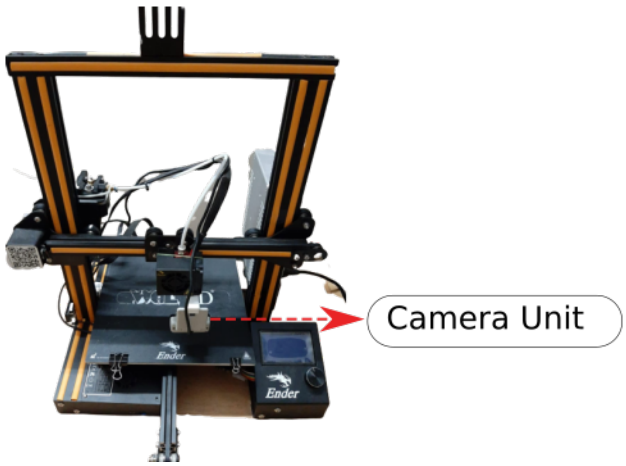


Fig. 1. Ender machine with Camera unit

The camera unit is installed parallell to the extruder nozzle end and kept aligned in such a way that it focuses extrusion along with bed. 3d printed holder is used to serve the purpose of holding the camera intact in the same position.

Along with this setup steel brush has also been fitted with bed platform which supports the nozzle to get blur free. Fitting enabled by clamp mounting to heated bed and the coordinates are noted. Open source firmware Marlin has been used in this work which supports dual extrusion also. GCode setting for machine. While pausing we need the machine to be settled at the home position and either waits for manual intervention or can automatically loads filament from the next spool. Gcode is nothing like the machine code which has been used to control the machine. In this cause the following operations needs to be performed based on results and action to be performed. The system should able to perform following operations,

1. Pausing
2. Resuming
3. Custom Gcode exec
4. Nozzle cleaning
5. Extruder Reloading
6. Return to Printing

4 Vision Sensing Detection

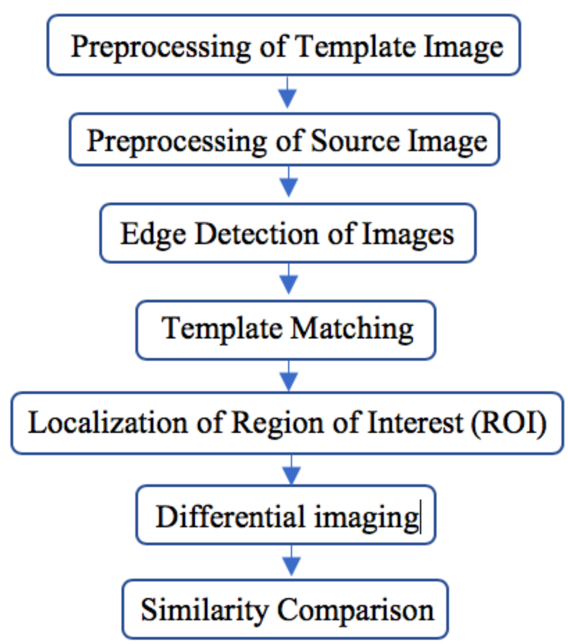
With the intention of detecting filament clogging or exhaustion in the immediate printing layer defined in [2.1 Errors] this work utlizes the adaptive template matching algorithm utingto locate the Region of Interest (ROI). In our case, ROI is the extruder nozzle that has to be located in the image frame segmented from the live feed of the webcamera adhered to the extruder. In this work we use Microsoft LifeCam HD3000 Web Camera since it is considerably low in cost and has focus-free lens with High Definition video streaming quality. The cost effective camera module with low installation time and cost can facilitate the users of such mediocre printers to implement this system effectively. The adaptive Framerate system is availed using the imutils library in python which provides series of convenience functions to make translation, rotation, resizing, skeletonization and much more. This adaptive framerate system can adjust the frames per second capturing according to the relative speed of the extruder.

4.1 Software Requirements

1. We use OpenCV(Open Source Computer Vision Library) in Version 4.3.0 for accelerated Computer Vision functionalities and the opencv-python API for python language support and this provides enormous Core Operations in image processing, Feature Detection, Image Reconstruction and Video Analysis.
2. We use Scikit-Image (skimage) in Version 0.16.1 to utilize versatile set of image processing routines such as image comparison and similarity check using NRMSE, SSIM, etc.

4.2 Workflow Methodology

Our proposed algorithm follows series of steps and methodologies as follows:



4.2.1 Preprocessing of Template Image

1. Selection of Template image (T) by capturing nozzle which is a patch to be compared with the source image with the fixed dimension of 432 x 239 x 3 pixels.
2. Resizing the Template image (T) to 15 percent of its original dimension which is 65 x 36 x 3 pixels to make the sliding window smaller for accurate matching.
3. Conversion of Resized RGB template image (3 color channels) to Grayscale/Greyscale image (TG) (1 color channel) to reduce the dimensionality ramification and rapid processing.

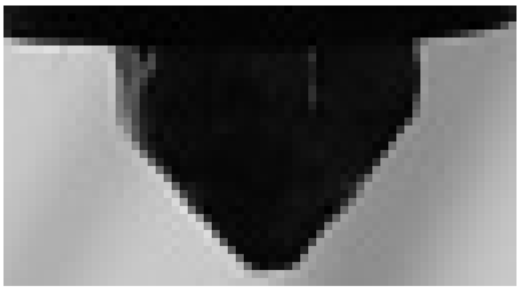


Fig. 2. Template image

4.2.2 Preprocessing of Source Image

1. Every fifth frame of the Video stream is grabbed as Source Image (S) which we expect to find a match to the template image with the fixed dimension of 4032 x 3024 x 3 pixels.
2. Resizing the Source image (S) to 15 percent of its original dimension which is 605 x 454 x 3 pixels to make the processing time agile.
3. Conversion of Resized RGB source image (3 color channels) to Grayscale/Greyscale image (SG) (1 color channel) to reduce the dimensionality ramification and rapid processing.
4. Resizing the SG image according to the scale and will keep track of the ratio of resizing to sustain the shape of TG image to SG image.

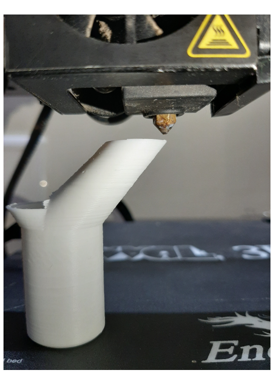


Fig. 3. Source image without Filament flow

4.2.3 Edge Detection of Images

The resized TG grayscaled and SG grayscaled images are then enforced edge detection. In our work we use Canny edge Detector which is a multi-step algorithm that can detect edges with noise suppression.

1. To reduce noise and undesireable textures, both images are smoothened with Gaussian Filter where

$$g(m,n) = G_\sigma(m,n) * f(m,n) \tag{1}$$

where Gaussian filter G_σ is,

$$G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{m^2+n^2}{2\sigma^2}\right)$$

2. This step detects the edge intensity and direction by calculating the gradient of the images using Sobel operator. The magnitude G and the slope θ are calculated as follows:

$$\begin{gathered} |G| = \sqrt{g_m^2(m,n) + g_n^2(m,n)} \\ \theta(m,n) = tan^{-1}[g_n(m,n)/g_m(m,n)] \end{gathered} \tag{2}$$

3. Threshold M_T is calculated for both the source and template image as follows:

$$M_T(m,n) = \begin{cases} M(m,n) & \text{if M(m,n) > T} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

4. The output of previous step is again Thresholded by two different thresholds τ_1 and τ_2 (where $\tau_1 < \tau_2$) to obtain two binary images T_1 and T_2.
5. Continuous edges are formed using Link edge segments in T_2. To do so, trace each segment in T_2 to its end and then search its neighbors in T_1 to find any edge segment in T_1 to bridge the gap untill reaching another edge segment in T_2.

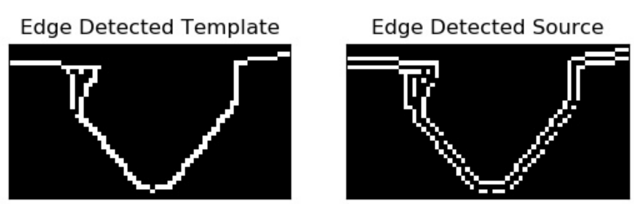


Fig. 4. Edge detected

If the source image at any frame is smaller than the template image, the iteration will stop and the preceding image with edge detection is passed for template matching. The edge detected Source image and Template image is denoted as S_E and T_E

4.2.4 Template Matching

The edge detected source and template images from the previous step encounters template matching algorithm. We have to compare the template image against the source image by sliding it through specific window size to determine the matching area i.e. extruder nozzle in our case. Sliding the patch one pixel at a time (left to right, up to down). At each location, a metric is calculated to represent how similar the patch is to that particular area of the source image. For each location of T over I, we store the metric in the result matrix (R). Each location (x,y) in R contains the match metric. We used Template Matching Correlation Coefficient (TM_CCOEFF) method. This method is used to a) make the template and image zero mean and b) make the dark parts of the image negative values and the bright parts of the image positive values.

This means that when bright parts of the template and image overlap we'll get a positive value in the dot product, as well as when dark parts overlap with dark parts (-ve value * -ve value gives +ve value). That means you get a +ve score for both bright parts matching and dark parts matching.

When you have dark on template (-ve) and bright on image (+ve) you get a -ve value. And when you have bright on template (+ve) and dark on image (-ve) you also get a -ve value. This means you get a negative score on mismatches.

$$R(x,y) = \sum_{x',y'} (T'(x',y') \cdot I(x+x',y+y')) \tag{4}$$

where

$$T'(x',y') = T(x',y') - 1/(w \cdot h) \cdot \textstyle\sum_{x'',y''} T(x'',y'') \tag{5}$$

$$I'(x+x',y+y') = I(x+x',y+y') - 1/(w \cdot h) \cdot \sum_{x'',y''} I(x+x'',y+y'') \tag{6}$$

Normalize the output of the TM_CCOEFF and localize the location with higher matching probability from the matrix R usually retrieved by minMaxLoc method in OpenCV class.

4.2.5 Localization of Region Of Interest (ROI)

To localize the coordinates of our region of interest we use the highest matching probability coordinates (x_p, y_p) from the previous step and the resizing ratio (r) to bound the ROI. The bounding box is formed by,

$$(X_s,Y_s) = (x_p * r, y_p * r)$$
$$(X_e,Y_e) = ((x_p + t_w) * r, (y_p + t_h) * r)$$

where

$X_s \Rightarrow$ starting value for x-coordinate
$Y_s \Rightarrow$ starting value for y-coordinate
$X_e \Rightarrow$ ending value for x-coordinate
$Y_e \Rightarrow$ ending value for y-coordinate
$t_w \Rightarrow$ width of the edge detected template image
$t_h \Rightarrow$ height of the edge detected template image

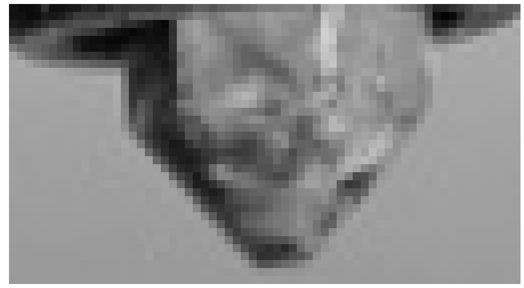


Fig. 5. ROI image

Using the starting and ending coordinates derived, ROI is constructed in single channel of dimension 64 x 35 pixels in addition to the template image of similar dimension.

4.2.6 Differential imaging

Absolute Difference of template image T_i and ROI image R_i generates a differential image. It yields the temporal difference between source and the template.

$$D_i = saturate(|T_i - S_i|) \tag{7}$$

This value of D_i will be considerably low when the filament flow is in the appropriate level. When the filament flow is clogged or exhausted the D_i value will increase abruptly to peak value. This helps in distinguishing whether the filament flow is "good" or "bad" to a meagre level.

4.2.7 Similarity Comparison

To gain more precise results, in addition to differential imaging we calculate Normalized Root Mean Squared Error (NRMSE) and Structural Similarity Index Measure (SSIM) available in skimage module.

1. **NRMSE:** NRMSE is exact deviation between the original and observed images. If the NRMSE value tends to be less compared to the fixed threshold of 0.558 then the template and ROI images are relatively same which concludes that the flow in the nozzle is filament less.

$$I_{NRMSE} = \frac{\sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{n}}}{X_{obs,max} - X_{obs,min}} \tag{8}$$

$$Extruder_{status} = \begin{cases} 1 \text{ if } I_{NRMSE} \leq 0.558 \\ 0 \text{ otherwise} \end{cases}$$

where,
$1 \Rightarrow$ No Filament
$0 \Rightarrow$ Filament

2. **SSIM:** SSIM is a perceptual metric that quantifies image quality degradation caused by processing and finds the similarity of provided images. SSIM will look for similarities within pixels i.e. similar pixel density values. SSIM addresses the main drawback of NRMSE which is difficulty in standardizing arbitarily high NRMSE values. In other hand SSIM ranges between -1 and +1. A score of 1 meant very similar and vice versa.

$$I_{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c1)(\sigma_x^2 + \sigma_y^2 + c2)} \tag{9}$$

$$Extruder_{status} = \begin{cases} 0 \text{ if } I_{SSIM} \leq 0 \\ 1 \text{ if } I_{SSIM} > 0 \end{cases}$$

where,
$0 \Rightarrow$ Filament
$1 \Rightarrow$ No Filament

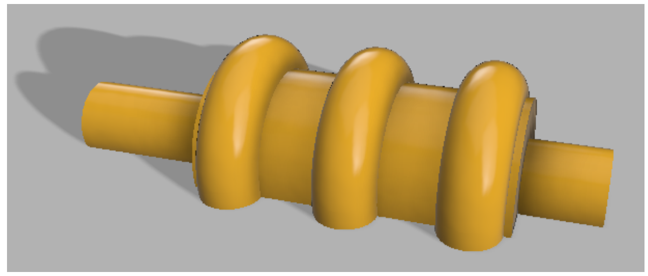


Fig. 6. CAD model

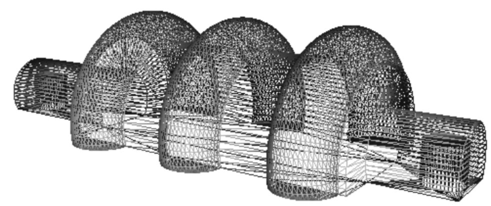


Fig. 7. STL Model

5 Machine Interruption System

The second phase of this work is the machine interruption system (pause) subsequent to the status of the extruder procured from the first phase (Vision sensing detection) for efficient printing. To accomplish this we have written certain conditions which is considered as Serial Intrusion algorithm to perform specific defined states of printing process such as pause, resume and stop amid printing. Serial intrusion algorithm overrides the Gcode sender script which is sending commands asynchronously along the tty port, to pause the printing process if the extruder status is 1 (No Filament). This work also facilitates sending of explicit Gcode during this state further to check for the filament flow anew with a trance time of 5 seconds. If the status of extruder changes back to 0 (Filament), the printing process is resumed at the exact layer and position without human interference.

5.1 CAD Modelling and Slicing Process

CAD Model being designed in any viable designing software is being converted to stl (Standard Triangle Language) format since it is the standard input model format in almost all 3D printers.

This file conversion depends upon various criteria such as layer height, filament diameter, machine work volume, travel speed, extrusion speed, and much more will be considered. Few open source software even allow slicer to auto create support when the angle of inclination is below 45 degree (defined as ideal condition). The following figures shows the CAD model and corresponding converted .stl file.

Numerous open source Slicer softwares such as CURA, Slic3r, Craftware etc. are available widely. Among them this work rely on CURA Slice Engine considering the versatility and the community support. Commencing the system requires a software to feed the machine parameters and certain functionalities including the machines serial responses such as machine motion coordinates, bed temperature and other coherent data both at the transmission and the receiving end. In this system we are using the RepetierHost platform as a exhibiter. From the CAD model various algorithmic approaches are available to slice the model. Usually slicing performs two operations (i.e.) detecting the contour and generating the infill pattern among the contour. There are various infill patterns such as grid, honeycomb, star, diagonal, etc to serve various purposes.The below figure shows CAD model being sliced by CURA slicer.

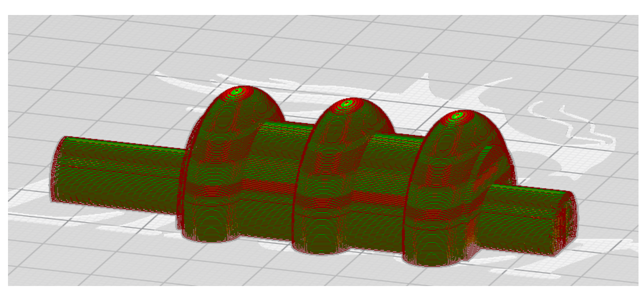


Fig. 8. Sliced model

The workflow for slicing mostly depends on the path tracing algorithms which commonly separates 3D design into 2D layers and stacking one over another. The corresponding flowchart is shown below in fig xx.

5.2 Firmware Setup

Firmware is the one which connects the hardware and software thereby facilitating as a bridge. The firmware plays the major role in controlling the functionality of 3D printers from converting GCode to motor signals, controlling heat ends and heat bed, operating LCD display and much more machine akin functionalities. Abundant firmwares are available which corresponds to the control board in the machine. Various firmware like Marlin, Repetier firmware, Reprap, Smoothieware are available as open source. For this proposed work Repetier host has been adopted since the Repetier host and Repetier firmware works well with each other. Once GCode is received, the defined 8 or 16 or 32 bit processor starts processing the code and adds it in the queue inside the sump to feed the printer. This will convert the Gcode into electric pulses which actuates the motor. The Stepper motors in the printer could able to count the rotations performed with the minimum step angle.

$$\text{Resolution} = \frac{\text{number of steps}}{\text{number of revolutions of the rotor}}$$

The above stated relation is used to figure the resolution of the motor, smaller the step angle more the resolution. In this case the angle is 1.8 degrees which has 200 steps per resolution which is nominal. Then comes the mechanical structure which defines the movement and transfer of rotational motion into linear motion in the motors. Various transfer methods can be achieved using different setups such as belt drive, screw drive, etc. In this work, we use 2mm pulley teeth belt. Simple mechanical calculation of the motor pulley diameter, and belt dimension gives the number of rotations required to move the quantified distance. All these calculations are performed by the control board and these criteria must be preloaded in the firmware.

5.3 Serial Intrusion Algorithm

This algorithm is designed in such a way that the microcontroller can able to handle multiple process at the same time using the concept of multithreading. A thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system. This algorithm runs two threads - Main thread (M) and Worker thread (W) parallelly with Event Handling. An Event is an object that can be accessed from all the threads and allows very basic communication between them. Main thread sends the Gcode through tty port asynchronously to the printer. The below table shows the event object value and its corresponding states:

Event Object	Result State
0	Stop
1	Print
2	Pause
3	Resume
4	Manual GCode Sender

Table 2. Event Object and State

This M thread maintains the transmission of Gcode and keeps track of the machine's response for the the Gcode recieved. Worker thread starts capturing the live feed of the extruder whenever M thread is active and process the first phase of the system. Everytime status of the extruder changes to 1 (NO FILAMENT) the W thread changes the value of event object to 2 which corresponds to pause state. Thus the M thread interrupts the serial communication and stores the code number being sent in a temporary memory to resume later. After the trance time of 5 seconds, W thread again captures the extruder to check for filament flow. If the status of extruder still remains unchanged (being 1), then the machine waits for the human response to resume printing, else (changes to 0) by default the event object changes to 3 which corresponds to resume state. Thus the M thread grabs the code number from the temporary memory and search for the same in the Gcode file to resume printing from where it is being interrupted. This saves the time and material wastage in humongous amount. In addition to this, during the pause state we can manually send Gcode to perform specific actions if required by giving event object as 4 which corresponds to manual gcode sender state. The printing process can also be completely stopped by providing event object value as 0 which corresponds to break state.

6 Results and Discussions

The following figures shows the histogram comparison between the template and filament flowing images obtained from the phase one.

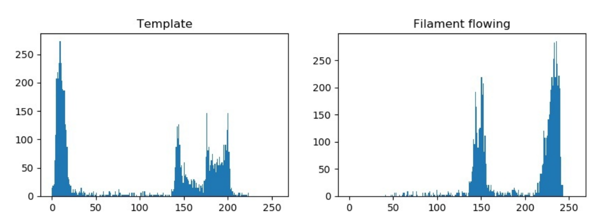


Fig. 9. Histogram Comparison

Thus if the Vision detection system detects the appropriate filament flow from the extruder, then the printing process is uninterrupted. Every fifth frame will undergo the same detection process for efficient results which yields the obligatory solution. The below mentioned figure shows the histogram comparison between the template and filamentless images.

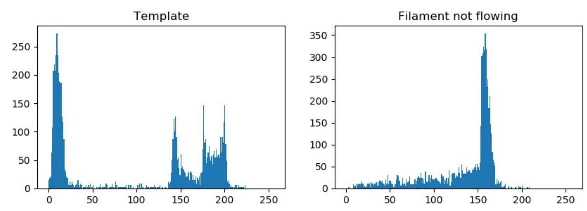


Fig. 10. Histogram Comparison

Here if the detection system detects, that there is no flow of filament from the extruder, it immediately overrides the serial transmission of Gcode to the printer and pause the printing process. Then the extruder goes to defined coordinates for secondary filament flow check. In secondary check the extruder motor will run again and the detection system verifies the filament flow. If the response is unchanged then the filament is exhausted so machine will remain in the pause state till human command, whether to pause or to resume once new filament roll is loaded or abort the printing process considering few erroneous layers. This functionality is achieved through the storage and retrieval of the GCode sequence. In case if the pause response is received at the mid of active GCode line ”X” then the system will accomplish that line ”X” before pausing the printing process. Therefore the codes are sent in a sequential fashion and cannot be shunned unless power failure in the system. Additionally filament oozed during secondary filament check can also be eliminated using a motile brush setup adhered to the machine. This ensures that there is no trace of blur exists in the printed object.

7 Conclusion

The above novel method of approach provides the solution to figure filament exhaustion and clogging while printing using advanced Computer vision approaches. The camera unit adhered to the printer is not only capable of providing solution for the above stated problem but also addresses various other issues such as over extrusion, under extrusion, filament misalignment and can stream the live feed to web which facilitates the user to regularly monitor the system. This concept works perfectly in Marlin Firmware with few functional changes in the firmware endorses to operate across other firmwares as well. Considering the market growth of 3D printers, these solutions will make the technology more reliable, efficient and provides inevitable standards for Additive Manufacturing itself. Not only for Fused Deposition Modelling but this prototype can be employed in other Additive manufacturing technologies. Ultimately this solution reduces humongous material wastage and saves immense amount of time during filament exhaustion or clogging error occurs in the system.

References

[1] Thabiso Peter Mpofu, Cephas Mawere, Macdonald Mukosera (2014). The Impact and application of 3D printing Technology. *International Journal of Science and Research, 3(6), 2148-2152.* Available: `https://home.liebertpub.com/publications/3d-printing-and-additive-manufacturing/621/overview`