

# 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. In spite 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 especially in Fused Deposition Modelling(FDM) method out to all engineering

enthusiasts. This end in development of industrial grade as well DIY printers out 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 familiar consumer grade 3D Printers. Based on this various technology for 3D printings 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 cost efficient to manufacture as they don't require expensive and complex components which is also due to availability of huge spare market nowadays. The foundation consists to 3 axes that are restrained by 3 stepper motor 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.

For direct supervision of inhabitant, the printers are recommended to exploit rooms rather than workspace. FDM machine printing duration can be up to 20 hours ample for large and complex objects alongside of loaded filaments in the system. The main objective is to reduce the misprint as they drain material and soak the printing time for longer time without producing a useful object and consuming the running time as well.

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 detection errors which experienced previously in the development. In order to get the information on problems and errors the user to watch the video constantly or in interval that helps to receive information on problems and assess remotely on printing progress. The design is to support the user to detect the printer error by utiliz-

ing machine vision. The benefits of early detection of printer errors reduces material waste and occupancy of printer resources. In need of constant surveillance for preventing completion of printing process, inappropriate or broken object to reduce the time effort of the user.

To identify these failures modes video frames and BLOB detection are presented in an in-line computer vision system which uses differential imaging between consecutive.

Computer Vision(CV) is the computer based methods of processing the signal, in this case image are the resource. The method is capable of performing all sort of processing like adding images, subtracting, comparing, validating all the possibilities via mathematical formulas. Current generation processor and GPU supports a lot for this computer vision technology hence making the system more viable for high speed operation. Various open source library are out there in market consisting of set of Code written in programming languages which has many mathematical formulas as library to preform. Among the interpreted language, python seems to be more flexible in our case ending up with development language for 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. Perform 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 sensor are developed by Rao et al to analysis the surface roughness of framework. These various techniques mainly focuses on detecting the challenges and provide 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 unit support to take images of side view mapping and virtual top view concept to analysis the system with algorithm. That solution fails to detect the filament error which make confusion to figure the exact problem recurring the manual intervention

### 2.1 Second-Level Heading

The next level of heading is also boldface with upper and lower case letters. The heading is flushed left with the left margin. The spacing to the next heading is two line spaces.

#### 2.1.1 Third-Level Heading.

The third-level of heading follows the style of the second-level heading.

## 3 Use of SI Units

An ASME paper should use SI units. When preference is given to SI units, the U.S. customary units may be given in parentheses or omitted. When U.S. customary units are given preference, the SI equivalent *shall* be provided in parentheses or in a supplementary table.

## 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 utilizes the adaptive template matching algorithm to 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 webcam-camera 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:

1. Preprocessing of Template Image
2. Preprocessing of Source Image
3. Edge Detection of Images
4. Template Matching
5. Localization of Region of Interest (ROI)
6. Differential imaging
7. Similarity Comparison

#### 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.

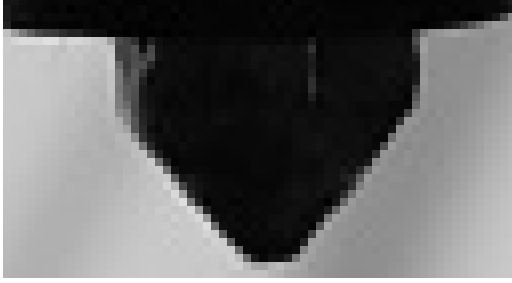


Fig. 1. Template image

#### 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.

#### 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 undesirable textures, both images are smoothened with Gaussian Filter where

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

where Gaussian filter  $G_{\sigma}$  is,

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

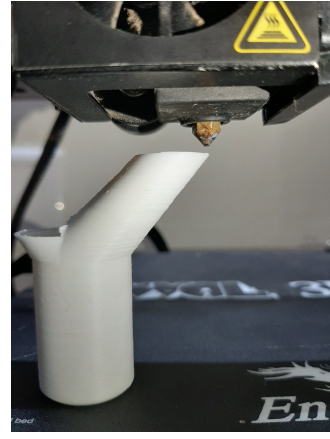


Fig. 2. Source image without Filament flow

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  $\theta$  are calculated as follows:

$$|G| = \sqrt{g_m^2(m,n) + g_n^2(m,n)} \quad (2)$$

$$\theta(m,n) = \tan^{-1}[g_n(m,n)/g_m(m,n)]$$

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

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

4. The output of previous step is again Thresholded by two different thresholds  $\tau_1$  and  $\tau_2$  (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 until reaching another edge segment in  $T_2$ .

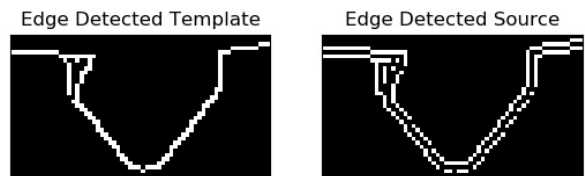


Fig. 3. Edge detected

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.

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

where

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

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

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,

$$\begin{aligned} (X_s, Y_s) &= (x_p * r, y_p * r) \\ (X_e, Y_e) &= ((x_p + t_w) * r, (y_p + t_h) * r) \end{aligned}$$

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

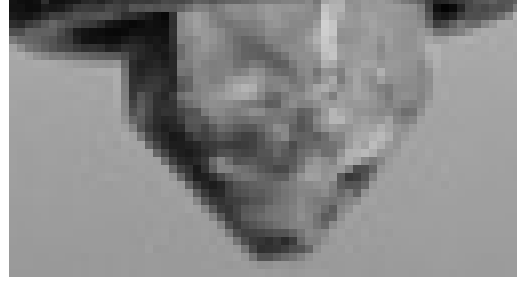


Fig. 4. ROI image

Using the starting and ending coordinates derived, ROI is constructed in single channel of dimension  $64 \times 35$  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.

$$D_i = \text{saturnate}(|T_i - S_i|) \quad (7)$$

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.

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

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

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 arbitrarily high NRMSE values. In other hand SSIM ranges between -1 and +1. A score of 1 meant very similar and vice versa.

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

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

where,

0  $\Rightarrow$  Filament

1  $\Rightarrow$  No Filament

## 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.

Fig xx shows the CAD model and corresponding converted .stl file.

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).

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.

Fig xxx shows CAD model being sliced

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 sumup 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 1. Event Object and State

This M thread maintains the transmission of Gcode and keeps track of the machine's response for the the Gcode received. 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 Citing References

The ASME reference format is defined in the authors kit provided by the ASME. The format is:

*Text Citation.* Within the text, references should be cited in numerical order according to their order of appearance. The numbered reference citation should be enclosed in brackets.

The references must appear in the paper in the order that they were cited. In addition, multiple citations (3 or more in the same brackets) must appear as a "[1-3]". A complete definition of the ASME reference format can be found in the ASME manual [?].

The bibliography style required by the ASME is unsorted with entries appearing in the order in which the citations appear. If that were the only specification, the standard `BIBTEX` `unsorted` bibliography style could be used. Unfortunately, the bibliography style required by the ASME has additional requirements (last name followed by first name, periodical volume in boldface, periodical number inside parentheses, etc.) that are not part of the `unsorted` style. Therefore, to get ASME bibliography formatting, you must use the `asmems4.bst` bibliography style file with `BIBTEX`. This file is not part of the standard `BibTeX` distribution so you'll need to place the file someplace where `LaTeX` can find it (one possibility is in the same location as the file being typeset).

With `LaTeX/BIBTEX`, `LaTeX` uses the citation format set by the class file and writes the citation information into the `.aux` file associated with the `LaTeX` source. `BIBTEX` reads the `.aux` file and matches the citations to the entries in the bibliographic data base file specified in the `LaTeX` source file by the `\bibliography` command. `BIBTEX` then writes the bibliography in accordance with the rules in the bibliography `.bst` style file to a `.bbl` file which `LaTeX` merges with the source text. A good description of the use of `BIBTEX` can be found in [?, ?] (see how two references are handled?). The following is an example of how three or more references [?, ?, ?] show up using the `asmems4.bst` bibliography style file in conjunction with the `asme2ej.cls` class file. Here are some more [?, ?, ?, ?, ?, ?, ?, ?, ?] which can be used to describe almost any sort of reference.

### 7 Conclusions

The only way to ensure that your figures are presented in the ASME Journal of Mechanical Design in the way you feel is appropriate and meets the requirement for quality presentation is for you to prepare a double column version of the paper in a form similar to that used by the Journal.

This gives you the opportunity to ensure that the figures are sized appropriately, in particular that the labels are readable and match the size of the text in the journal, and that the line weights and resolutions have no pixilation or rasterization. Poor quality figures are immediately obvious on the printed page, and this detracts from the perceived quality of the journal.

I am pleased to provide advice on how to improve any figure, but this effort must start with a two-column version of the manuscript. Thank you in advance for your patience with this effort, it will ensure quality presentation of your research contributions.

## 8 Discussions

This template is not yet ASME journal paper format compliant at this point. More specifically, the following features are not ASME format compliant.

1. The format for the title, author, and abstract in the cover page.
2. The font for title should be 24 pt Helvetica bold.

If you can help to fix these problems, please send us an updated template. If you know there is any other non-compliant item, please let us know. We will add it to the above list. With your help, we shall make this template compliant to the ASME journal paper format.

## Acknowledgements

ASME Technical Publications provided the format specifications for the Journal of Mechanical Design, though they are not easy to reproduce. It is their commitment to ensuring quality figures in every issue of JMD that motivates this effort to have authors review the presentation of their figures.

Thanks go to D. E. Knuth and L. Lamport for developing the wonderful word processing software packages  $\text{\TeX}$  and  $\text{\LaTeX}$ . We would like to thank Ken Sprott, Kirk van Katwyk, and Matt Campbell for fixing bugs in the ASME style file `asme2ej.cls`, and Geoff Shiflett for creating ASME bibliography style file `asmems4.bst`.

## 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>

## Appendix A: Head of First Appendix

Avoid Appendices if possible.

## Appendix B: Head of Second Appendix

### Subsection head in appendix

The equation counter is not reset in an appendix and the numbers will follow one continual sequence from the beginning of the article to the very end as shown in the following example.

$$a = b + c. \quad (10)$$