Factors effecting real-time optical monitoring of fused filament 3D printing

Siranee Nuchitprasitchai¹ · Michael Roggemann¹ · Joshua M. Pearce^{1,2}

Received: 8 December 2016 / Accepted: 8 June 2017 / Published online: 17 June 2017
© Springer International Publishing AG 2017

Abstract This study analyzes a low-cost reliable real-time optical monitoring platform for fused filament fabrication-based open source 3D printing. An algorithm for reconstructing 3D images from overlapping 2D intensity measurements with relaxed camera positioning requirements is compared with a single-camera solution for single-side 3D printing monitoring. The algorithms are tested for different 3D object geometries and filament colors. The results showed that both of the algorithms with a single- and double-camera system were effective at detecting a clogged nozzle, incomplete project, or loss of filament for a wide range of 3D object geometries and filament colors. The combined approach was the most effective and achieves 100% detection rate for failures. The combined method analyzed here has a better detection rate and a lower cost compared to previous methods. In addition, this method is generalizable to a wide range of 3D printer geometries, which enables further deployment of desktop 3D printing as wasted print time and filament are reduced, thereby improving the economic advantages of distributed manufacturing.

Keywords Real-time monitoring · 3D printing · Optical monitoring · RepRap · Open hardware · Quality assurance

1 Introduction

As the Stratasys patent [1] expired on fused deposition modeling (FDM) in 2009, a more generalized form [fused filament fabrication (FFF)] enabled the self-replicating rapid prototyper (RepRap) 3D printer project [2–4] to develop and scale. The RepRap project was developed using open source hardware protocols [5] around the Arduino microcontroller [6, 7]. The expected rapid innovation in the open source community [8] succeeded, and dropped the cost of FFF 3D printers by several orders of magnitude [9], spawned dozens of 3D printer start-up companies, and brought 3D printing to prosumers at a rapid rate [10]. This change had helped to lower the cost of an open source RepRap style 3D printer, which makes it economically viable for average consumers to offset purchases with 3D printing [11]. RepRaps can reproduce more than half of their own components and can self-upgrade, which make them attractive for a wide range of applications including sustainable development and farming [12–14], education [15–19], rapid prototyping standard products [20, 21] to microfluidics [22, 23] small business manufacturing [24–27], as well as scientific tools [28–31]. However, these low-cost printers are still short of the reliability standards [32–35] that consumers are accustomed to with other consumer products. Some works have estimated a 20% failure rate for inexperienced 3D printer users on DIY machines [11]. This is primarily due to inherent challenges of FFF printing, which although far improved in the last several years [36] persist, including warping (the print peeling away from the print bed), elephant foot (base of model bugles near print bed) or other first-layer problems, lower region of model shrinks relative to the middle, skewed prints or shifted layers, layer misalignment (where a layer in z prints offset in x or

✉ Joshua M. Pearce
pearce@mtu.edu

¹ Department of Electrical and Computer Engineering,
Michigan Technological University, 1400 Townsend Drive,
Houghton, MI 49931-1295, USA

² Department of Materials Science and Engineering, Michigan
Technological University, Houghton, MI, USA

y above the previous layer), missing layers, deformation that causes cracks in tall objects, pillowing (where the top of a print has holes or bumps from non-uniform top surfaces), stringing (e.g., extraneous thin strands of polymer connect components on a plane of the model), under-extrusion (where a solid wall is not formed between subsequent extrusion passes), over-extrusion (where too much polymer is deposited), gaps in the top layers, visible lines in the bottom layers, scars on the top surface from extrusion problems, or no filament comes out of the nozzle [37]. These errors cost money and waste time as they reduce prosumer use due to frustration and reduce the environmental benefits of distributed manufacturing [38–42].

To build on previous work in order to develop a low-cost reliable real-time optimal monitoring platform for FFF-based 3D printing, this study uses an algorithm for reconstructing 3D images from overlapping 2D intensity measurements with relaxed camera positioning requirements. For single-side 3D printing monitoring, single- and double-camera solutions are compared for the following variables: six different 3D object geometries, five filament colors. The results are compared between the two-camera setups as well as the results of previously published techniques. The limitations of this approach are detailed and future work is described. The results are then discussed and conclusions are drawn in the context of furthering the adoption of desktop 3D printing for distributed manufacturing.

2 Background

Several attempts have been made to improve the reliability of 3D printers using high-resolution imaging. However, the majority of this work has been based on high-cost, high-resolution laser-based 3D printing systems. Kleszczynski et al. [43] presented an overview of an error detection in an EOS INT M 270 Laser Beam Melting System with a monochrome CCD camera system, a tilt and shift lens to reduce perspective distortion, and an adjustable tube for changing height or reducing the distance between the lens and the object. Similarly, Jacobsmühlen et al. [44] successfully applied their images to inspect a powder bed AM process result on a microscopic scale for flaw detection, missing powder or low energy input, surface quality, and measurements of part geometries. Later, Jacobsmühlen et al. [45] showed that for providing high-resolution image-based measurements, calibration of perspective correction need to be done using their template matching approach based on the experiment setup from [44]. Kleszczynski et al. [46] presented two approaches to improve 3D printing process stability including (1) using a high-resolution imaging setup and (2) an enhanced version with a proximity sensor. In addition, several commercial systems

based on proprietary computer vision systems are also available for high-end printers. For example, the price for a Sintavia relies on Concept Laser for 3D process monitoring in real time for metal additive manufacturing system, which costs around US\$800,000 [47].

Relatively little work has investigated error detection in prosumer desktop FFF-based 3D printing. Many commercial FFF-based printers now incorporate some form of sensor to detect clogged nozzles or lack of filament. For example, re:3D, which sells the Gigabot brand of open source 3D printers, has a filament detector available as an add-on because their users have long (>10 h) print jobs. These approaches all target a single catastrophic failure mechanism. For full error detection, the work is based primarily on monitoring Makerbot branded derivatives [48–53] of the RepRap project. Baumann et al. met with limited success that did not support flat objects and those with similar material color with the printer using an open source software approach with OpenCV [54] and Python [55] to detect errors including detachment, missing material flow, and deformed object with a Playstation eye cam [50]. Hurd et al. successfully applied a mobile device to remotely monitor internal and external errors with Samsung Galaxy Tab 3 [48]. Ceruti et al. met with limited success using an open source software approach with Augmented Reality toolkit (AR) [56], Speeded Up Robust Features (SURF) algorithm [57], and The Random Sample Consensus (RANSAC) algorithm [58] to detect the differences between a reference 3D model (CAD) and the 3D printing model with a camera and Augmented Reality Wuzix glasses [51]. Faes et al. had nearly zero production failure in the z-direction to detect the deposited tracks and to determine the dimension of interest in a closed-loop feedback in an Extrusion-based 3D printing (E3DP) with a modular 2D laser triangulation scanner [49]. Straub successfully applied a visible light scanning with a multi-camera system and open source software approach with C# and Dot Net Framework [59] to detect dry printing when filament is not applied and premature job termination when project is not complete with Raspberry Pi [60], five Raspberry Pi cameras, and a visible light 3D scanning system [52]. Straub provide an overview on how to characterize an internal structures and covered surfaces defects of complex objects with a Raspberry Pi (full specifications available at [60]), a multi-camera system (five Raspberry Pi cameras), and a visible light 3D scanning system [53]. Other complex systems have been investigated [61]. Flexible plastic toys production line prototype systems with the integration of a 3D printer, industrial robot, and machine vision have been demonstrated in a laboratory environment [63]. Finally, Cummings et al. [63] presented some preliminary results with the detection and correction of filament in closed-loop control for 3D printing using ultrasonic signals with limited success.

3 Methods

This paper undertakes a detailed study of the use of an algorithm for reconstructing 3D images from overlapping 2D intensity measurements with relaxed camera positioning requirements [63]. For this paper, experiments were set up in two different ways: (1) using one camera to capture a 2D image from a single 3D printing model to create a 2D shape image, and (2) using two cameras to capture two 2D images from a single 3D printing model to perform a 3D reconstruction. A different algorithm is used for each experimental setup, but the same type of camera, printer, and tested objects are used. Due to the distance between the camera and the printer for the experiment setup, the field of view for both cameras can cover the printed area of 70 mm in width and 60 mm in height. To eliminate the shadow on the object scene, there should be sufficient light sources. Both experimental setups used the same 3D printer using a delta-style RepRap, PointGrey cameras, distance between the camera and the printer, distance between the light sources and the printer, blue printing base, and filament brand. The relation of geometry between the 3D printer and the camera system needs to be known for using camera calibration technique to calculate the intrinsic and extrinsic parameters for a specific camera setup. These parameters will be used to correct for lens distortion and to determine the location of the camera in the scene.

A low-cost (<US\$500 in parts) [16] open source delta-style polymer printing RepRap (MOST Delta) is used [64]. The MOST Delta is a RepRap [65] derived from the Rostock [66] printer as shown in Fig. 1 with a cylindrical build volume 270 mm in diameter and 250 mm in height and overall dimensions of 375 mm diameter and 620 mm in height. The cameras are set up on 1 side of the printer 580 mm from the outer edge as shown in Fig. 1. The cameras used in this study are two identical 1394a Firefly MVs, with an image size of 480-by-640 (height-by-width), pixel size of which is 6 μm with square pixels, and a focal length of 16.6 mm. The computer models chosen are a Tyrannosaurus rex skull, cube, twisted gear vase, rectangular prism, cylinder, and triangular prism are available [67] as shown in Fig. 2. The printing parameters used are layer height 0.2 mm, shell thickness 1 mm, unable retraction, bottom/top thickness 1 mm, fill density 20%, print speed 60 mm/s (except the skull model, which used 20 mm/s), printing temperature 180 $^{\circ}\text{C}$, diameter filament 1.94–1.98 mm, flow filament 100%, and nozzle size 0.5 mm. The PLA filament used in this experiment is Hatchbox 3D PLA with dimensional accuracy ± 0.05 mm on 1 kg spools, 1.75 mm diameter with red, pink, glow, black, and orange colors.

3.1 Single-camera setup

To detect an error from a single-camera setup, after simulating a 2D shape image (CameraImage) of the 3D object, it is compared to the 2D shape model (STLImage). To create a STLImage, a rendered 3D model in OpenSCAD is saved into stl file (STLImage), and then all data from stl file are plotted in x , y , z axes using stlTools [68] to display the shape of the rendered 3D model, which can be observed from different viewpoints. The position of the viewer for plotting the model needs to be set specify as the position of camera viewpoint while taking an image. Thus, in the right position of the viewer, the shape of the STLImage is saved as PNG image type on xz -plane. The CameraImage is created after capturing a 2D image from the 3D printing model. The background is then removed and rendered white. Distortion is removed from the image by intrinsic parameters from camera calibration [69] following the details in the method. Next a region of interest (ROI) is calculated from the image by converting the color image into a gray-scale image, and then converting it into binary image. The object area in the binary image is converted to be white in color used as the ROI; otherwise, it is converted to be black. The sizes of the object in STLImage and CameraImage are defined by edge detection, and then the object size ratio between these two images can be found for rescaling. After rescaling, edge detection is applied again to find the min and max positions of the object in both images for rectification. After rectification, any errors in the process are detected by subtracting the simulated 3D object image from the actual image. If the difference of subtraction is greater than 5%, there is an error; otherwise, there is no error flagged (see Fig. 3).

3.2 Two-camera setup

To detect an error from the two-camera setup between a 3D printed object and a 3D reconstruction from two cameras, the following process is used (Figs. 4, 5). First, the background is removed and rendered white from the images taken from two cameras (LeftImage and RightImage). Distortion and the ROI are calculated as above.

However, in the two-camera case, there is another problem as points in 3D space must be matched between the two images. To resolve this problem, the Scale Invariant Feature Transform (SIFT) [70] and the Random Sample Consensus (RANSAC) [58] models are applied for rescaling and rectification. The algorithm for doing this has been described previously [63].

Next, the error detection is obtained by comparing the 3D printed object and 3D reconstruction image. If the difference between the two is more than 5%, there is an error identified; otherwise there is no action taken to stop the print.

Fig. 1 MOST Delta printer with optical monitoring experimental setup

Fig. 2 Rendering of STL models for testing: **a** Tyrannosaurus rex skull, **b** cube, **c** twisted gear vase, **d** rectangular prism, **e** cylinder, and **f** triangular prism

3.3 Validation

The dimensions of the 3D printed objects are measured with a digital caliper (± 0.05 mm). A 3D reconstruction of

the object is created from two images and the object size is calculated. Next, the sizes of both objects are compared to calculate size difference, an error of the reconstruction. For validation of this approach, six different test objects with

Fig. 3 Error detection for one-camera model flowchart

different color filaments are printed including (a) Tyrannosaurus rex skull (pink), (b) cube (black), (c) twisted gear vase (red), (d) rectangular prism (red), (e) cylinder (glow), and (f) triangular prism (orange).

4 Results

The validation print images are shown in Fig. 6. They are printed in order to detect missing material flow when the supply of filament is cut during a 3D print.

The error detection from one camera was tested with different geometries (Tyrannosaurus rex skull, cube, twisted gear vase, rectangular prism, cylinder, and triangular prism) with different filament colors (pink, black, red, glow, and orange) because different colors give both different 3D printing results and can represent different

Fig. 4 Error detection for two-camera model part 1 flowchart: 1 checking 3D object calculation and 2 plotting stl object

Fig. 5 Error detection for two-camera model part 2 flowchart

Fig. 6 Original left and right image for different geometries with *different colors*: **a** Tyrannosaurus rex skull (pink), **b** cube (black), **c** twisted gear vase (red), **d** rectangular prism (red), **e** cylinder (glow), and **f** triangular prism (orange) (color figure online)

challenges for image processing. The error detection system is tested with two different conditions: first, when the 3D printer finishes complete printing, and second when the 3D printer fails and a print is incomplete. Printing is tested with different geometries as shown in Fig. 7 and Table 1. Please note that the camera and OpenSCAD images are silhouettes and not full images. Table 1 shows that the shape errors are between 0.984 and 2.987%. This error is acceptable because the error of shape difference is less than 5%.

The incomplete project has been tested with different geometries between the camera image and SCAD image in different conditions as shown in Fig. 13 and Table 2. Table 2 shows that the shape errors are greater than 5%. When the nozzle is clogged, or an incomplete project is caused by filament running out that affects the 3D printing shapes, then they are smaller than the SCAD models. The one exception in this case is the triangle model that is less than 5% between shape image (150 layers) and SCAD Image (200 layers) because the top of triangle has a small area (see Fig. 8).

The error detection for the complete project from two-camera setup was tested with different geometries (Tyrannosaurus rex skull, cube, twisted gear vase, rectangular prism, cylinder, and triangular prism) with different filament colors (pink, black, red, glow, and orange) because different colors give us different 3D printing results. A manual caliper is used to measure the width and

the height of the real object size in millimeters (the 3D model printing) as shown in Fig. 9. The width and the height of the 3D reconstruction is calculated after pointing those points on the image to get the X, Y, Z positions manually. The percentages of the error measurements for the complete project for each geometry with different colors for the width and the height are calculated after the difference in the width and height are found in millimeter. The 3D reconstruction for different geometries are shown in Figs. 9, 10, 11, 12, 13, and 14 and the percentages of errors are between 0.07 and 3.94 that there are acceptable because the error of size difference is less than 5% as shown in Table 3.

5 Discussion

The experiments demonstrate that both the single- and two-camera setups can be used to detect any form of catastrophic FFF 3D printing error such as clogged filament. This differentiates it from other more simple sensor-based methods that can detect only one error. In addition, it opens up the possibility to do far more detailed error detection (e.g., inaccurate layers) and even for future advances to correct these errors automatically during printing.

Table 3 summarizes the percentage of error for complete printing between single- and two-camera setups. The size error percentage of two cameras is less than the shape

Fig. 7 One-camera setup with different geometries: **a** Tyrannosaurus rex skull (pink), **b** cube (black), **c** twisted gear vase (red), **d** rectangular prism (red), **e** cylinder (glow), and **f** triangular prism (orange)

error percentage of single camera. However, the calculation time of two cameras setup is greater than that of the single camera setup as the two-camera setup provided the width and height error. There are more error details for the double-camera setup than the single camera that provided only the total shape error.

The error detection system works as designed for both the single- and two-camera setups. To detect an error more

accurately, the perspective view of STLImage needs to be set as the actual perspective view between cameras and the 3D printing object. It should be noted that a printed 3D object usually has a small error when compared to its designed 3D model because of the FFF process that impacts error detection calculation. These experiments show that the shape error detection can determine when the printing has failed because the 3D printed objects are

Fig. 7 continued

Table 1 One camera: error measurements for each geometries

Object	Skull		Cube		Twisted gear vase		Rectangle		Cylinder		Triangle	
Color	Pink		BlackPink		Red		Red		Glow		Orange	
Size (mm)	W	H	W	H	W	H	W	H	W	H	W	H
SCAD model	28.68	60.55	30.00	30.00	43.82	38.03	10.00	50.00	28.10	51.10	50.23	43.50
Shape error (%)	2.98		1.34		2.20		0.98		2.58		2.06	
Calculation time (s)	6.64		6.90		7.07		7.16		9.03		6.56	

W width, H height

Table 2 One camera: example for error measurements when the printings fail in different layer heights

Layer heights		Error (%)					
Shape image	SCAD image	Skull	Cube	Vase	Rectangle	Cylinder	Triangle
50	100	18.68	15.87	12.71	19.36	15.06	13.76
100	150	13.82	9.97	11.97	11.54	12.19	7.60
150	200	12.95		6.78	10.04	10.13	3.39

smaller than the SCAD models and the error percentage is greater than 5%. However, the error detection system will detect an error better than either process alone, when the single- and two-camera setups are combined to detect error together. While the 3D printer is printing, the single-camera system detects a shape error every N layers because the computation time is less than 10 s for the whole object. If a shape error is greater than 5%, it will report to the user and pause printing. If there is no shape error, the two-camera

system will start to detect a size error. If a size error is greater than 5%, it will report to the user and pause printing. If there is no size error, the 3D printer will continue. This combined method provides both the size and shape error detection with required accuracy in reasonable times for FFF printing.

Overall, the combination of the two methods (single and double camera) was found to be the most effective. The use of cameras can be less expensive than other methods used

Fig. 8 One-camera setup: error detection for different geometries between camera and SCAD image. **a** Skull model between 250 layers and full model. **b** Twisted gear vase model between 150 layers and full model. **c** Cube model between 150 layers and full model.

d Rectangle model between 150 layers and 200 layers. **e** Cylinder model between 150 layers and full model. **f** Triangle model between 100 layers and full model

to determine the accuracy of a 3D print such as a laser scanning or sensor [49]. Using the single-camera method, the computation time (6.9 s for 9 cm²) is faster than both subtraction (fastest is 10 s for 6.25 cm²) and the searching (fastest is 12 s for 6.25 cm²) algorithm developed by Hurd et al. [48]. There are other methods to stop catastrophic failures. For example, Barker developed a system that works for delta-style RepRaps, which stops a print when electrical connections are broken if any of the linking rods

are thrown [71]. In addition, to the increase in complexity for the 3D printing system, this is also not generalizable to other 3D printers that do not have magnetic bearings (e.g., most Cartesian-based printers). Early work has tried to determine ways to use relatively expensive ultrasonic sensors to detect errors with promise, but unreliable results [63]. This method (100% detection) can detect an error better than vision-based error detection for 3D printing processes when missing material flow (80% detection)

Fig. 8 continued

[50]. When the square model is tested printing every ten layers when the layer height is 0.2 mm, the shape errors are greater than 5% when the nozzle is clogged, or an incomplete project. Using the single-camera method can detect an error at 2 mm in height which is smaller than 5 mm [51].

Other solutions to 3D print failure provided in the RepRap community have had video monitor of printing [72], but the user has to stop the print manually if the user detects an error through continuous human surveillance.

This obviously undermines one of the primary benefits of bespoke automated fabrication with 3D printers because of the necessary human involvement. The system described here overcomes that issue to allow for automatic error detection with no human oversight. However, the algorithm here still has two fundamental limitations. First, the finite (several seconds) of commutation time (as summarized in Table 4) does not allow every layer to be monitored in real time for small printed objects as the print speed is faster than the analysis time. For larger more

Fig. 9 Tyrannosaurus rex skull (pink): **a** width measurement and **b** height measurement (color figure online)

Fig. 10 Twisted gear vase (red): **a** width measurement and **b** height measurement (color figure online)

Fig. 11 Cube (*black*): **a** width measurement and **b** height measurement

Fig. 12 Cylinder (*glow*): **a** width measurement and **b** height measurement (color figure online)

Fig. 13 Triangle (orange): **a** width measurement and **b** height measurement (color figure online)

complex prints, this is less of an issue and as the results have shown here sampling a printed object after several layers is adequate for catastrophic failures although it does not enable real-time automatic error detection (and the potential for real-time error correction). To get to that goal, the computation time would need to be reduced. This may be possible by streamlining the computation and removing it from the Matlab environment. Doing the latter will also overcome one of the other primary challenges to the use of this method in the distributed manufacturing community, specifically, although the algorithms provided here are open source [73]. They currently are run in the Matlab environment which costs \$2150 [74]. This is not that expensive for research or in higher end 3D printer applications, but represents a barrier to deployment in the low-cost prosumer printers used for distributed manufacturing, which generally cost in total \$2500 or less (the RepRap used in this study was \$500 in parts).

In addition, to overcoming these limitations there are several other areas of future research. First, this system would be improved if it was applied to all sides of the printing object. For future research, this error detection system will be implemented and extended from the basic approach into 360° around FFF-based 3D printing. It will improve the object detection capability as there is better understanding for the scene geometry and

therefore for object detection in the depth dimension. Furthermore, to reduce the cost for adding the error detection system to FFF-based 3D printing, low-cost web cameras will be applied in this system. Using low-cost optics will need to be vetted for its effects on the performance of the system and the algorithms presented here.

6 Conclusions

This paper described a low-cost reliable real-time monitoring platform for FFF-based 3D printing based on a single- and two-camera system for a single side. The results showed that both of the algorithms with a single- and double-camera system were effective at detecting a clogged nozzle, loss of filament, or an incomplete project for a wide range of 3D object geometries and filament colors. The error calculation was determined from the difference in shape between STLImage and CameraImage, or the different sizes between STLImage and the 3D reconstruction. Printing was stopped when these errors exceeded 5%. The validity of this approach using experiments shows that the error detection system is capable of a 100% detection rate for failure detection. The combined method analyzed here has a better detection rate and a lower cost to previous

Fig. 14 Rectangle (red): **a** width measurement and **b** height measurement (color figure online)

Table 3 Error measurements for complete project for each geometries with different colors

Object		Skull		Cube		Twisted gear vase		Rectangle		Cylinder		Triangle	
Color		Pink		Black		Red		Red		Glow		Orange	
Size (mm)	Axes	W	H	W	H	W	H	W	H	W	H	W	H
Error	Real object	28.68	60.55	29.86	30.23	44.01	38.50	9.59	50.14	28.65	49.48	50.73	43.58
	3D reconstruction	28.70	61.1	30.39	30.00	44.35	38.19	9.52	51.2	29.78	50.98	51	44
	$\pm$	+0.02	+0.55	+0.53	−0.22	+0.34	−0.30	−0.07	+1.06	+1.13	+1.5	+0.27	+0.42
	%	0.07	0.90	1.80	0.75	0.77	0.79	0.70	2.11	3.94	3.03	0.54	0.96
Calculation time (s)		73.88		45.66		67.05		38.73		58.78		51.56	

W width, H height

Table 4 Error measurements for complete printing each tested geometry (W: width, H: height) of two cameras (size error) and single camera (shape error)

Object		Skull		Cube		Twisted gear vase		Rectangle		Cylinder		Triangle	
Color		Pink		Black		Red		Red		Glow		Orange	
Size (mm)	Axes	W	H	W	H	W	H	W	H	W	H	W	H
SCAD model		55.13	59.90	30	30	43.82	38.03	10	50	48.45	49	50.23	43.5
Size error (%)		0.07	0.9	1.8	0.75	0.77	0.79	0.7	2.11	3.94	3.03	0.54	0.96
Calculation time (s)		73.88		45.66		67.05		38.73		58.78		51.56	
Shape error (%)		2.98		1.34		2.20		0.98		2.58		2.06	
Calculation time (s)		6.64		6.90		7.07		7.16		9.03		6.56	

methods. In addition, this method is generalizable to a wide range of FFF 3D printer geometries, which enables further adoption of desktop 3D printing for distributed manufacturing as wasted print time and filament are reduced.

References

- Crump SS, Stratasys Inc. (1992) Apparatus and method for creating three-dimensional objects. US Patent 5,121,329
- Sells E, Smith Z, Bailard S, Bowyer A, Olliver V (2010) RepRap: the replicating rapid prototyper: maximizing customizability by breeding the means of production. In: Piller FT, Tseng MM (eds) Handbook of research in mass customization and personalization: strategies and concepts, vol 1. World Scientific, New Jersey, NJ, pp 568–580
- Jones R, Haufe P, Sells E, Iravani P, Olliver V, Palmer C, Bowyer A (2011) RepRap—the replicating rapid prototype. Robotica 29:177–191. doi:10.1017/S026357471000069X
- Bowyer A (2014) 3D printing and humanity's first imperfect replicator. 3D Print Add Manuf 1(1):4–5. doi:10.1089/3dp.2013.0003
- Gibb A, Abadie S (2014) Building open source hardware: DIY manufacturing for hackers and makers. Pearson Education, Upper Saddle River, NJ
- Banzi M, Shiloh M (2014) Getting started with Arduino: the open source electronics prototyping platform. Maker Media Inc, Sebastopol, CA
- Arduino <https://www.arduino.cc/>. Accessed 10 Nov 2016
- Raymond E (1999) The cathedral and the bazaar. Knowl Technol Policy 12(3):23–49
- Rundle G (2014) A revolution in the making. Simon and Schuster, South Melbourne, Vic
- Wohlers T (2016) Wohlers report 2016. Wohlers Associates Inc, Fort Collins, CO
- Wittbrodt BT, Glover AG, Laureto J, Anzalone GC, Oppliger D, Irwin JL, Pearce JM (2013) Life-cycle economic analysis of distributed manufacturing with open-source 3-D printers. Mechatronics 23(6):713–726
- Pearce JM, Morris Blair C, Laciak KJ, Andrews R, Nosrat A, Zelenika-Zovko I (2010) 3-D printing of open source appropriate technologies for self-directed sustainable development. J Sustain Dev 3(4):17–29
- Fox S (2010) After the factory [Manufacturing renewal]. Eng Technol 5(8):59–61
- Pearce JM (2015) Applications of open source 3-D printing on small farms. Org Farm 1(1):19–35. doi:10.12924/of2015.01010019
- Kentzer J, Koch B, Thiim M, Jones RW, Villumsen E (2011) May. An open source hardware-based mechatronics project: the replicating rapid 3-D printer. In: Mechatronics (ICOM), 2011 4th international conference on IEEE, pp 1–8
- Irwin JL, Oppliger DE, Pearce JM, Anzalone G (2015) Evaluation of RepRap 3D printer workshops in K-12 STEM. In: 122nd ASEE 122nd ASEE Conf. proceedings, paper ID#12036
- Gonzalez-Gomez J, Valero-Gomez A, Prieto-Moreno A, Abderahim M (2012) A new open source 3d-printable mobile robotic platform for education. In: Rückert U, Joaquin S, Felix W (eds) Advances in autonomous mini robots. Springer, Berlin Heidelberg, pp 49–62
- Grujović N, Radović M, Kanjevac V, Borota J, Grujović G, Divac D (2011) 3D printing technology in education

- environment. In: 34th international conference on production engineering, pp 29–30
19. Schelly C, Anzalone G, Wijnen B, Pearce JM (2015) Open-source 3-D printing technologies for education: bringing additive manufacturing to the classroom. *J Vis Lang Comput* 28:226–237
 20. Campbell I, Bourell D, Gibson I (2012) Additive manufacturing: rapid prototyping comes of age. *Rapid Prototyp J* 18(4):255–258
 21. Gibson I, Rosen D, Stucker B (2014) Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing. Springer, New York, NY
 22. O'Neill PF, Azouz AB, Vazquez M, Liu J, Marczak S, Slouka Z, Chang HC, Diamond D, Brabazon D (2014) Advances in three-dimensional rapid prototyping of microfluidic devices for biological applications. *Biomicrofluidics* 8(5):052112
 23. Pearce JM, Anzalone NC, Heldt CL (2016) Open-source wax RepRap 3-D printer for rapid prototyping paper-based microfluidics. *J Lab Autom* 21(4):510–516
 24. Rimock M (2015) An introduction to the intellectual property law implications of 3D Printing. *Can J Law Technol* 13(1):1–32
 25. Laplume A, Anzalone GC, Pearce JM (2016) Open-source, self-replicating 3-D printer factory for small-business manufacturing. *Int J Adv Manuf Technol* 85(1):633–642. doi:[10.1007/s00170-015-7970-9](https://doi.org/10.1007/s00170-015-7970-9)
 26. Tech RP, Ferdinand JP, Dopfer M (2016) Open source hardware startups and their communities. In: Ferdinand JP, Petschow U, Dickel S (eds) *The decentralized and networked future of value creation*. Springer, Switzerland, pp 129–145
 27. Troxler P, van Woensel C (2016) How will society adopt 3D printing? In: van den Berg B, van der Hof S, Kosta E (eds) *3D printing*. TMC Asser Press, Netherlands, pp 183–212
 28. Pearce JM (2012) Building research equipment with free, open-source hardware. *Science* 337(6100):1303–1304. doi:[10.1126/science.1228183](https://doi.org/10.1126/science.1228183)
 29. Pearce JM (2014) Open-source lab: how to build your own hardware and reduce research costs. Elsevier, Waltham, MA
 30. Baden T, Chagas AM, Gage G, Marzullo T, Prieto-Godino LL, Euler T (2015) Open labware: 3-D printing your own lab equipment. *PLOS Biol.* doi:[10.1371/journal.pbio.1002086](https://doi.org/10.1371/journal.pbio.1002086)
 31. Coakley M, Hurt DE (2016) 3D printing in the laboratory maximize time and funds with customized and open-source labware. *J Lab Autom*:2211068216649578
 32. Klodowski A, Eskelinen H, Semken S (2015) Leakage-proof nozzle design for RepRap community 3D printer. *Robotica* 33(04):721–746
 33. Mercuri R, Meredith K (2014) An educational venture into 3D printing. In: *Integrated STEM education conference (ISEC), 2014 IEEE, IEEE*, pp 1–6
 34. Chonga S, Chiub HL, Liaob YC, Hungc ST, Pand GT (2015) Cradle to Cradle® design for 3D printing. *Chem Eng* 45:1669–1674
 35. Moilanen J, Vadén T (2013) 3D printing community and emerging practices of peer production. *First Monday* 18(8). <https://firstmonday.org/ojs/index.php/fm/article/view/4271/3738>. Accessed 13 June 2017
 36. Frauenfelder M (2013) Make: ultimate guide to 3D printing 2014. Maker Media. Inc., O'Reilly, Sepaspol
 37. Alastair J (2016) presented 16 common 3D Printing Problems and Solutions publishing all3dpweb. <https://all3dp.com/common-3d-printing-problems-and-their-solutions/>. Accessed 10 Nov 2016
 38. Kreiger M, Pearce JM (2013) Environmental life cycle analysis of distributed three-dimensional printing and conventional manufacturing of polymer products. *ACS Sustain Chem Eng* 1(12):1511–1519
 39. Vera J (2010) Promoting tools that integrate LCA into the product design process: a case study in Ontario. <http://wbooth.mcmaster.ca/epp/publications/student/JoseVeraInquiry.pdf>. Accessed 13 June 2017
 40. Kreiger M, Pearce JM (2013) Environmental impacts of distributed manufacturing from 3-D printing of polymer components and products. In: *MRS proceedings*, vol 1492. Cambridge University Press, Cambridge, pp 85–90
 41. Kostakis V, Roos A, Bauwens M (2016) Towards a political ecology of the digital economy: socio-environmental implications of two competing value models. *Environ Innov Soc Trans* 18:82–100
 42. Bonvoisin J (2016) Implications of open source design for sustainability. In: Setchi R, Howlett R, Liu Y, Theobald P (eds) *Sustainable design and manufacturing 2016*. Springer International Publishing, pp 49–59
 43. Kleszczynski S, zur Jacobsmühlen J, Sehrt JT, Witt G (2012) Error detection in laser beam melting systems by high resolution imaging. In: *Proceedings of the solid freeform fabrication symposium*
 44. zur Jacobsmühlen J, Kleszczynski S, Schneider D, Witt G (2013) High resolution imaging for inspection of laser beam melting systems. In: *2013 IEEE international instrumentation and measurement technology conference (I2MTC)*. IEEE
 45. zur Jacobsmühlen J, Kleszczynski S, Witt G, Merhof D (2014) Robustness analysis of imaging system for inspection of laser beam melting systems. In: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE
 46. Kleszczynski S, zur Jacobsmühlen J, Reinartz B, Sehrt JT, Witt G, Merhof D (2014) Improving process stability of laser beam melting systems. In: *Proceedings of the Fraunhofer direct digital manufacturing conference*
 47. Concept laser. <http://www.conceptlaserinc.com/>. Accessed 10 Nov 2016
 48. Hurd S, Carmen C, Jules W (2015) Quality assurance in additive manufacturing through mobile computing. In: Sigg S, Nurmi P, Salim F (eds) *International conference on mobile computing, applications, and services*. Springer International Publishing
 49. Faes M, Abbeloos W, Vogeler F, Valkenaers H, Coppens K, Ferraris E (2014) Process monitoring of extrusion based 3D printing via laser scanning. In: *PMI 2014 conference proceedings*, vol 6, pp 363–367
 50. Baumann F, Dieter R (2016) Vision based error detection for 3D printing processes. In: *MATEC web of conferences*, vol 59. EDP Sciences
 51. Ceruti A, Alfredo L, Tiziano B (2016) Augmented vision and interactive monitoring in 3D printing process. *IJDeM*:1–11
 52. Straub J (2015) Initial work on the characterization of additive manufacturing (3D printing) using software image analysis. *Machines* 3(2):55–71
 53. Straub J (2016) Characterization of internal geometry/overed surface defects with a visible light sensing system. In: *SPIE Commercial + Scientific Sensing and Imaging*. International Society for Optics and Photonics
 54. Opencv. <http://opencv.org/>. Accessed 10 Nov 2016
 55. Python software foundation [US]. <https://www.python.org/>. Accessed 10 Nov 2016
 56. ARTOOLKIT. <https://artoolkit.org/>. Accessed 10 Nov 2016
 57. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (SURF). *Comput Vis Image Underst* 10(3):346–359
 58. Fischler Martin A, Bolles Robert C (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM* 24(6):381–395
 59. Microsoft. <https://msdn.microsoft.com/>. Accessed 13 Nov 2016
 60. RaspberryPi. <https://www.raspberrypi.org/>. Accessed 10 Nov 2016

61. Hu F, Li L, Liu Y, Yan D (2016) Enhancement of agility in small-lot production environment using 3D printer, industrial robot and machine vision. *Int J Simul Sys Sci Technol* doi:10.5013/IJSSST.a.17.43.32. <http://ijssst.info/Vol-17/No-43/paper32.pdf>. Accessed 13 June 2017
62. Cummings I, Hillstrom E, Newton R, Flynn E, Wachtor A (2016) In-process ultrasonic inspection of additive manufactured parts. *Topics in modal analysis & testing*, vol 10. Springer, Berlin, pp 235–247
63. Nuchitprasitchai S, Roggemann MC, Havens TC (2016) Algorithm for reconstructing three dimensional images from overlapping two dimensional intensity measurements with relaxed camera positioning requirements to reconstruct 3D image. *IJMER* 6(9):69–81
64. Appropedia. Delta Build Overview: MOST publishing appropedia web. http://www.appropedia.org/Delta_Build_Overview: MOST. Accessed 13 June 2016
65. RepRap. <http://reprap.org/>. Accessed 13 June 2016
66. Johann. Rostock <http://reprap.org/wiki/Rostock>. Accessed 5 Nov 2016
67. Nuchitprasitchai 3-D models. <https://osf.io/6rfky/>. 2016-11-17
68. Pau M. (2015) presented stlTools publishing mathworkswb. <https://www.mathworks.com/matlabcentral/fileexchange/51200-stltools>. Accessed 1 Oct 2016
69. Camera Calibration publishing mathworkswb. <https://www.mathworks.com/help/releases/R2013b/vision/ug/find-camera-parameters-with-the-camera-calibrator.html> Accessed 15 Oct 2016
70. Lowe DG (1999) Object recognition from local scale-invariant features. In: *Computer vision, 1999. In: The proceedings of the seventh IEEE international conference on*, vol 2, pp 1150–1157
71. Barker B. Thrown Rod Halt Mod. Published on Appropedia.org http://www.appropedia.org/Thrown_Rod_Halt_Mod. Accessed 5 Nov 2016
72. David G (2016) presented adding a Raspberry Pi case and a camera to your LulzBot Mini publishing kupoos Web. <http://www.kupoos.com/video/q7oqOPzCHYE/adding-a-raspberry-pi-case-and-a-camera-to-your-lulzbot-mini/>. Accessed 20 Nov 2016
73. Real time optical monitoring of fused filament 3-D Printing. Published on the Open Science Framework. <https://osf.io/hwdzm/> Accessed 8 Dec 2016
74. Mathworks, pricing and licensing. <https://www.mathworks.com/pricing-licensing/index.html?intendeduse=comm>. Accessed 8 Dec 2016