Honors Thesis Proposal

Adam Slattum

April 14, 2017

Presented to Dr. John Bowers, Ph.D.
Department of Computer Science
James Madison University

1 Purpose and Objectives

Automated detection of three-dimensional (3D) printer failures has become an interesting topic for researchers since the use of the 3D printers has become wide-spread. The use of 3D printers for prototyping is a popular practice among many companies and individuals for proof of concept testing of their ideas and designs. The purpose of this honors thesis is to develop a method for automated detection of 3D printer failures. 3D printing errors often snowball into catastrophic errors, requiring the current print to be discarded, the cause of the error to be diagnosed and fixed, and a new print to be started from the beginning. Furthermore, a printing error that occurs in the second hour of a thirty hour print might only be discovered twenty-eight hours later, resulting in lots of wasted material at best, and damage to equipment at worst. Catching such errors early would save both time and material.

The field of 3D printing came into existence in the late 1980s, when 3D printing technology was referred to as Rapid Prototyping technology. 1986 marks the year when the first patent was issued for a stereo-lithographic apparatus (SLA) by Charles Hull [5]. He went on to co-found 3D Systems Corporation which is one of the largest and most prolific organizations operating in the 3D printing sector today. Though the field has its roots in the 1980s, it was not until 2007 that the 3D printing market saw its first 3D printer available for purchase under 10,000 dollars [5]. More importantly, 2007 is the year that started the open source 3D printing movement. In January 2009, the BfB RapMan 3D printer, the first commercially available 3D printer, went on sale to the public [4]. In 2012, as the market began to diverge and there were advances in capabilities as well as applications of 3D printers, an increase in awareness and interest flooded the market. Today, 3D printing has amassed a diversified market, expanding the usefulness of 3D printing to aerospace, architecture, automotive, electronics, defense, dentistry, medicine, and many other areas [8].

The most recent error detection system that has been developed is by an Australian Engineer by the name of Chris Barr. The technology he uses is only for detecting overheating stepper drivers, belt slips, and accidental filament buildup, not actual errors with prints due to layer shifting or the print not looking how it is supposed to [1]. While this is beneficial for hardware malfunctions and issues with the components of a 3D printer, it is not focused on errors with the print itself.

This project is motivated by the lack of software to detect errors in 3D prints and the impact that this software will have on several industries utilizing 3D printers. By combining aspects of computer vision and computational geometry, this project seeks to better the overall experience of working with 3D printers and minimize cost and time metrics due to failures that occur during the printing process.

2 Tentative Outline

There has been little done in the development of a software application that can detect if a 3D printed object looks right as it prints and/or detects errors on the fly by looking at the object itself. Furthermore, starting from scratch will be a challenge; however, in this project, we will incrementally develop a system for automated detection of print errors resulting in incorrect prints. This contribution to 3D printing should help pave the way for further exploration, research, and application of 3D printing.

In this project, we will develop a software package with the following features:

- Implement and deploy an image processing algorithm for collection of pictures suitable to be passed into a mesh generation algorithm.
- Implement a mesh generation algorithm to create point clouds of 3D printed objects.
- Create an image processing algorithm to compare a correct 3D printed object's mesh to an incorrect 3D printed object's mesh for error detection and parameter selection.
- Develop a real-time script that can compare meshes while the 3D object is being printed.

Determining what constitutes an error and what does not will be difficult as it depends on the print and how much of margin of error the user is willing to accept. To accomplish this and make the error detection customizable, I plan to integrate a user selected error tolerance which the user can select the tolerance they are willing to accept. This might be challenging; however, it could make the problem of determining an error simpler. Lastly, the end goal for this project is the creation of a software application that compares a mesh of a 3D printed object with the expected ground truth 3D model to determine if an error has occurred during the printing phase to either correct or stop the print before wasting time and resources.

3 Methodology and Timeline

Remaining accountable and having visible deliverables are vital when constructing a creative thesis. Therefore, I plan to approach this project in a structured way using a modified Scrum software development methodology. Scrum is an incremental and iterative software development approach where one conducts 'sprints', which are two to three week long cycles that focus on implementing a particular portion of the project. At the end of each sprint, a working prototype is shown for criticism. Scrum will compliment the development of 3D error detection software quite well because demonstrations happen frequently and there is an opportunity to reflect and improve the process along the way.

Prior Work (Spring 2017): Familiarize myself with OpenCV which is open source computer vision software and experiment with different open sources packages for computer vision.

- *Week of January 24, 2017:* Familiarized myself with the project and determine a starting point.
- *Week of February 3, 2017:* Downloaded OpenSfM, which is an open source structure for motion library for creating meshes from 3D pictures.
- *Week of February 12, 2017:* Revisited OpenSfM to use different parameters for more accurate meshes, explored epipolar geometry and its applications, and considered the possibility of rendering images.
- *Week of February 17, 2017:* Familiarized myself with BoofCV, a real-time computer vision library for robotics. Also, I considered more effective methods for meshing to include image segmentation, 3D stereo clouds, and a 3D scanner in cooperation with the Physics Department at James Madison University.
- *Week of February 26, 2017:* Familarzied myself with the ASUS Xtion Pro Live Stereo RGB camera and OpenNI, which is open source middleware for communicating with the Xtion Pro Live.
- *Week of March 12, 2017 and March 19, 2017:* Decided to use the Xtion Pro Live within ROS (Robot Operating System) and visualize point clouds in rviz
- *Week of March 23, 2017:* Succeeded in seeing point cloud visualizations within ROS and began some Python scripting with OpenCV compiled with OpenNI support

Future Timeline (Fall 2017): Determine a camera to use capturing close images, high resolution images, implementation of image processing software to mesh the correct images of a 3D object and the incorrect image of a 3D object, and create a software application to compare the two.

- *Week 1 through Week 2:* Acquire high resolution stereo camera for capturing images with little stereoscopic deviation.
- *Week 2 through Week 8:* Implement image processing algorithm from current open source libraries to create a 3D reconstruction pipeline for mesh/point cloud comparison of 3D printed objects.
- *Week 8 through Week 15:* Create of mesh/point cloud comparison algorithm to compare an error free 3D printed object and one with errors to detect that a failure has occurred.

Future Timeline (Spring 2018): Wrap up the completed mesh comparison algorithm into a software application that can be used in the field, finalize thesis on experience and project as well as steps that can be taken to enhance the current algorithm.

- *Week 1 through Week 4:* Package the necessary algorithms and software into an application that is useful.
- *Week 4 through March 15, 2018:* Develop thesis paper to reflect on my experience, the progress made the research, and next steps.
- *By March 15, 2018:* Finalize thesis paper and send out to readers and faculty advisor for criticism.
- *By April 10, 2018:* Submit final project and thesis to the honors college for consideration.
- *Late April:* Honors Symposium presentation or conference presentation.

References

[1] Barr, C. (2016, April 05). 3D Printer Error Detection. Retrieved April 02, 2017, from `http://chrisbarrbuilds.com/2016/04/3d-printer-error-detection/`.

[2] Felix Baumann, Dieter Roller, Vision based error detection for 3D printing Processes, MATEC Web of Conferences 59 06003 (2016), DOI: 10.1051/matecconf/20165906003.

[3] Moreno, R., Newcombe, R. A., Strasdat, H., Kelly, P., and Davison, A. J. (n.d.). SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. Retrieved April 2, 2017, from `http://www.doc.ic.ac.uk/~ajd/Publications/salas-moreno_etal_cvpr2013.pdf`.

[4] Parametric Modeling with Autodesk Fusion 360, Randy Shih, SDC Publications 2017.

[5] The Free Beginner's Guide - History. (n.d.). Retrieved April 02, 2017, from `https://3dprintingindustry.com/3d-printing-basics-free-beginners-guide/history/`.

[6] Tomasi, C., Zhang, J., and Redkey, D. (1995, July 2). Experiments With a Real-Time Structure-From-Motion System. Retrieved April 2, 2017, from `https://users.cs.duke.edu/~tomasi/papers/tomasi/tomasiIser95.pdf`.

[7] Tu, Z., Chen, X., Yuille, A. L., and Zhu, S. (2003, June 23). Image Parsing: Unifying Segmentation, Detection, and Recognition. Retrieved April 2, 2017, from `http://www.stat.ucla.edu/~sczhu/papers/reprint_IJCV_parsing.pdf`.

[8] 3D Printing Industry Examples and Case Studies. (2017). Retrieved April 02, 2017, from `http://www.javelin-tech.com/3d-printer/industry/`.