Detection of faults in 3D prints

project plan

Cameron drain

[c.drain.15@abdn.ac.uk](mailto:c.drain.15@abdn.ac.uk)

Department of Computing Science,

University of Aberdeen, Aberdeen AB24 3UE, UK

**Introduction**

in this day and age we are currently getting to the point we need to produce more and more unique one off products or build prototypes as we are beginning to design faster and faster. One of the many tools are used to help with this rapid prototyping is 3D printing sadly will 3D printing especially in home grade printers are not the most reliable machines and frequently result in field parts resulting in significant waste in both material and time. My plan is to create a system that will help alleviate this problem.

**Goals**

the main goal is to create a system that is capable of detecting when a 3D print has developed a mistake. There exist some other systems that are designed to detect mistakes in 3d prints but most offer a very crude approach of simply being able to detect failure passes. I would like to create a system is capable of identifying a large range of mistake that can occurred and self rectify the problem (time permitting).

**Methodology**

the basic idea for detecting a mistake in a 3D print will to be to use a series of devices to scan a 3D print and create a virtualisation of what is currently being produced by the printer this will then be compared to the intended design. To achieve this I will be using xbox360 kinects which have the ability to act as not only a camera but also return a point cloud. For the software side I’m not entirely sure what I’ll be using at this stage as I am still comparing the various options available to me.

For the self rectification part this will be achieved by modifying the special control signals that are sent to the printer during printing.

**Resources required**

- 3D printer

-Kinect

-laptop

-Linux or Windows

**risk assessment**

the main risk here would currently be there at the centres I currently am using are unable to provide sufficient accuracy to perform any level of detection. If this occurs I would have to switch to a different form of scanning the model potentially looking into using a multi-camera setup. Or use 2 virtual objects and add mistakes in software

**Timetable**

21/01-4/2 Kinect scan 3 phases setup

4/1-11/2 comparing scan to model

11/2-18/2 write up

25/02-4/3 Kinect scan 1 phases setup

4/3-11/3 make model from g code and compare to scan

11/3-18/3 write up

18/3-20/3 Draft of the Results section.

20/3-22/3 Draft of the Discussion section

22/3 24/3 -Draft of the Conclusion section

24/3-26/3 Draft of Abstract and Introduction

26/3-19/4 Additional Goals

19/4-26/4 & Writeup Period

3/5 hand in