# DEVELOPMENT OF AN ADDITIVE MANUFACTURING RE-COATER MONITORING SYSTEM FOR POWDER BED FUSION SYSTEMS

**Francois Du Rand**

**210124148**

**A dissertation submitted in fulfilment of the requirements for the degree**

**Magister Technologiae: Engineering: Electrical**

**in the Department of Electronic Engineering**


**Department: Electronic Engineering**

**Faculty of Engineering and Technology**

**Vaal University of Technology**

**Vanderbijlpark**

**Supervisor: Dr PJM van Tonder**

**Co-supervisor: Prof HCvZ Pienaar**

**Date: May 2018**

## Declaration

I, Francois Du Rand hereby declare that the following research information is solely my own work. This is submitted in fulfilment of the requirements for the Magister Technologiae: Engineering: Electrical to the Department of Electronic Engineering at the Vaal University of Technology, Vanderbijlpark. It has not been submitted before any assessment to any educational institution.

Francois Du Rand

Date

## Acknowledgements

I wish to express my thanks and gratitude to the following people and organisations without whose help I would not have been able to complete this dissertation successfully.

- Dr PJM van Tonder for all the help and encouragement during the research.
- Prof HCvZ Pienaar for help and guidance during this research study.
- Conrad Beukes for all his assistance on the Voxeljet VX500.
- Niquilene Coetzee for help with the referencing.
- The VUT Science Park for making all the machines available to conduct my experiments.
- The Telkom Centre of Excellence and VUT Science Park for the funding.
- The staff from the Telkom Centre of Excellence for their assistance with the general administration work.

## Dedication

I would like to dedicate this dissertation to my parents Johan and Charmain Du Rand, my sister Niquilene Coetzee and brother-in-law Johan Coetzee, for all their support and encouragement during my studies. All glory to God who gave me the strength and courage to complete this research study.

## Abstract

In the world of research and development, the ability to rapidly manufacture a prototype or part has become a significant part of the manufacturing process. This requirement has given rise to some unique manufacturing technologies. One of these technologies is Additive Manufacturing (AM), or also more commonly known as 3D printing. There are several AM technologies available and can be divided into three major AM categories namely: liquid, powder and solid sheet based. For this research study, the primary focus will be on powder-based technologies. Powder-based technologies make use of materials in powder form and use different fusion techniques to fuse the powder particles together. All the powder bed fusion technologies consist of the same basic components, namely a powder chamber, build chamber, re-coater and a powder fusion system. For each layer of the build, the re-coater applies a new layer of powder from the powder chamber to the build chamber, and then the specific type of powder fusion system will fuse the powder particles together. This process will then be repeated until the entire build has completed.

Currently, powder bed fusion AM platforms do not have re-coating quality feedback into the printing system. Thus, when errors or defects occur on the powder bed surface during the re-coating process, they can affect the structural integrity of the parts. Parts must then be reprinted, which becomes costly due to wasted raw materials, electricity and time. Raw material and sundry wastage was some of the key factors that reduces the overall efficiency of the identified AM technology. Due to the increased problem with wasted materials, the need arose to develop a re-coater monitoring system, which could be used to increase the overall efficiency of a powder-based system.

For the development of a re-coater monitoring system, a review of three different types of monitoring technologies such as computer vision, laser scanning and a time-of-flight camera was conducted. Based upon the relatively low cost, low computer resource requirements and high accuracy, computer vision was considered as the best suited technology for development of the monitoring system. To select the correct camera to capture images of the powder bed, the required specifications for the camera, lens and mounting position were determined mathematically. A software

program was then developed to autonomously detect re-coating errors on the captured image after each re-coating cycle using image processing techniques. Each of the captured powder bed images were divided into 16 equal sized quadrants, where each quadrant was processed individually. Each of the quadrants was examined using an edge detection algorithm to detect any changes in contrast that would indicate a defect or re-coating error. The probability of a possible re-coating error or defect was calculated for each quadrant and displayed as a percentage value.

The active re-coater monitoring system was also integrated into the Voxeljet VX500 to validate the system's operation. The system was used to monitor a total of seven build jobs on the Voxeljet VX500. However, the first three build jobs could not be successfully monitored as some parameters of the system had to be re-adjusted to ensure proper operation. The last four build jobs were monitored successfully and recorded results that proved that the active re-coater monitoring system could indeed detect defects and re-coating errors when they occurred.

# Table of Contents

## List of Figures

## List of Tables

# Glossary of abbreviations or symbols

| Symbol | Description | Symbol | Description |
|---|---|---|---|
| **3** | | **L** | |
| 3D | Three Dimensional | LED | Light Emitting Diode |
| **A** | | LS | Laser Sintering |
| AM | Additive Manufacturing | **M** | |
| AMP | Advanced Manufacturing Precinct | MIT | Massachusetts Institute of Technology |
| ARM | Advanced RISC Machine | **O** | |
| **B** | | OCR | Optical Character Recognition |
| BGR | Blue-Green-Red | OEM | Original equipment manufacturer |
| BIM | Building Information Modelling | OpenCV | Open Computer Vision |
| BJ | Binder jetting | OS | Operating System |
| **C** | | **P** | |
| CAD | Computer Assisted Drawing | PLA | Polylactic Acid |
| CCTV | Closed Circuit Television | **R** | |
| $CO_2$ | Carbon Dioxide | RGB | Red-Green-Blue |
| CSI | Camera Serial Interface | RISC | Reduced Instruction Set Computer |
| CSV | Comma Separated Value | ROI | Region-of-interest |
| CW | Continuous Wave | **S** | |
| **D** | | SGSTP | Southern Gauteng Science and Technology Park |
| DLP | Digital Light Processing | SLS | Selective laser sintering |
| DMLS | Direct metal laser sintering | SoC | System on a Chip |
| **E** | | SPI | Serial Peripheral Interface bus |
| EBM | Electron beam melting | STL | Stereolithography |
| EMI | Electromagnetic Interference | **T** | |
| **F** | | TLS | Terrestrial Laser Scanner |
| FDM | Fused deposition modelling | TOF | Time-of-flight |
| FOV | Field of View | **U** | |
| **G** | | USB | Universal Serial Bus |
| GUI | Graphical User Interface | USAF | United States Air Force |
| **H** | | **V** | |
| HDMI | High Definition Multimedia Interface | VUT | Vaal University of Technology |
| **I** | | **W** | |
| I2C | I-squared-C | WIFI | Wireless Fidelity |
| IDE | Integrated Development Environment | | |

# Chapter 1 Introduction

## 1.1 Preface

In the world of research and development, the ability to manufacture a prototype or final product has rapidly become a significant part of the manufacturing process. This requirement has given rise to some unique manufacturing technologies. One of these technologies is called Additive Manufacturing (AM), or also more commonly known as 3D printing. There are several types of AM technologies, as displayed in Figure 1 which can be divided into three major AM categories namely: liquid, powder and solid sheet based (Fram-Schwartz 2016).

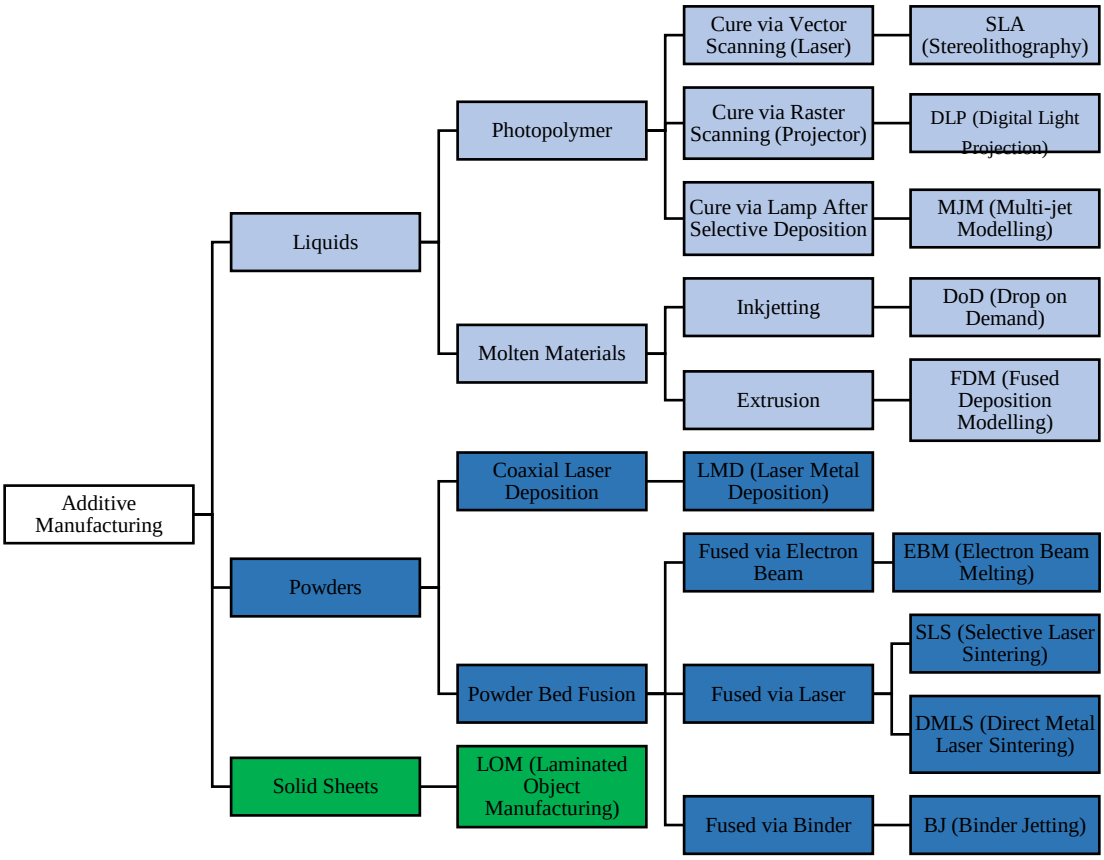



**Figure 1 Categories of additive manufacturing(Fram-Schwartz 2016)**

Liquid technologies make use of liquid materials that are cured using different types of light sources, by melting the material to create extrusions or spraying using inkjet technology. Powder technologies make use of materials in the form of powder where

different techniques could be used to fuse the powder particles together. Lastly, solid sheet technologies make use of sheets of material that are compressed or welded together as laminations to create the part geometry.

Under each category, several different sub-categories divide the main categories into smaller subcategories, depending on the base material used. Each of the subcategories is then broken down into the different types of technology platforms. Some of the technology platforms shown are old and becoming obsolete. Several of the older technology platforms have been replaced by more advanced and efficient methods.

### 1.1.1 VUT Southern Gauteng Science and Technology Park

The Vaal University of Technology Southern Gauteng Science and Technology Park (VUT SGSTP) was established in 2012 as a strategic technology hub in the Southern Gauteng region. The SGSTP is mandated by the Vaal University of Technology to develop products and foster enterprise development through technology transfer and innovation (Zhuwakinyu 2012). One of the flagship programs, being driven at the VUT SGSTP, includes the Advanced Manufacturing Precinct (AMP). The AMP specialises in the manufacturing of parts or prototypes for real-world applications using different AM technologies. The different AM technologies available at the VUT SGSTP include Fused Deposition Modelling (FDM), Laser Sintering (LS) and Binder Jetting (BJ). The manufacturing method of parts will depend on the part geometry, material and application.

Each one of these technology platforms is directed towards a specific industry sector. The BJ technology is focused primarily on the foundry sector. The technology allows the manufacturing/printing of moulds or patterns with very complex external and internal geometries without using traditional pattern or mould manufacturing methods. This offers the advantage of a near net shaped casting, with only the minimal amount of machining and finishing required (Mauchline 2016).

### 1.1.2 Powder bed fusion technologies

As indicated previously in Figure 1, powder bed fusion technologies can be divided into four different technology platforms, namely: Electron Beam Melting (EBM),

Selective Laser Sintering (SLS) or Laser Sintering (LS), Direct Metal Laser Sintering (DMLS) and BJ. Since most of the powder bed fusion technologies work on the same principle, the different platforms consist of the same fundamental components. The main components of the powder bed fusion technologies include a build platform, powder bed, powder feed platform and a re-coater or powder-levelling roller as illustrated in Figure 2.

The build and powder chambers have movable platforms, which actuates in the Z-axis (vertically). The powder chamber acts as a powder feeder, where the build chamber defines the building space. The particle fusion device can move in the X and Y axis. The re-coating process is when a thin layer of powder is spread over the build platform by the re-coater. Then, depending on the type of technology used, the computer-programmed head or beam moves/scans over the pre-programmed area to fuse the powder particles. This fusion of the powder particles creates the geometry of the part at the specific layer. The build platform will move down to re-coat the next layer of powder over the build platform. This manufacturing process will then be repeated until the part has been printed or sintered (Gonzalez *et al.* 2016).

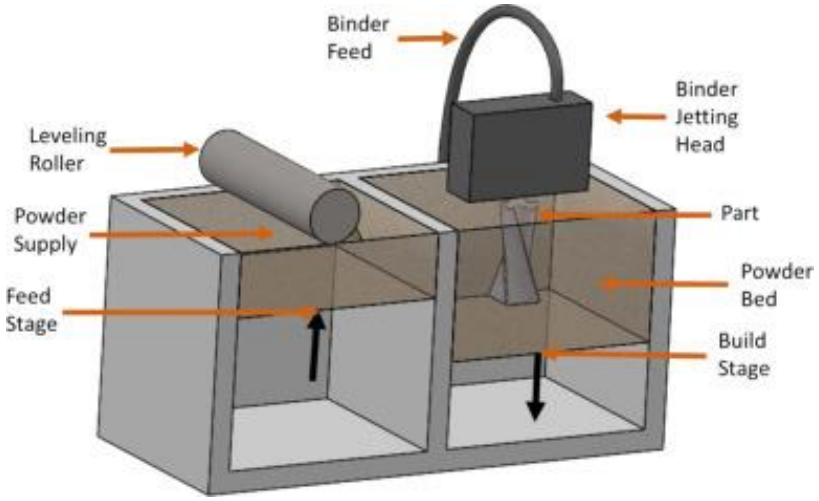



**Figure 2 Powder bed fusion machine basic parts (Gonzalez *et al.* 2016)**

Once the part geometry is fused together, the part is then removed from the powder bed and cleaned to remove the excess powder. Some of the powders are reusable; however, it will depend on the powder's material properties (Ardila *et al.* 2014). Since

the operation of the powder bed fusion technologies is very similar, only the different particle fusing methods will be discussed below.

- **Selective laser sintering and direct metal laser sintering**

LS, SLS and DMLS can be used to sinter parts from a variety of different polymer and metal alloy powders. Both SLS and DMLS technologies use high power lasers as an energy source to melt the powder particles together. The laser beam is computer guided using scanning optics as shown in Figure 3.

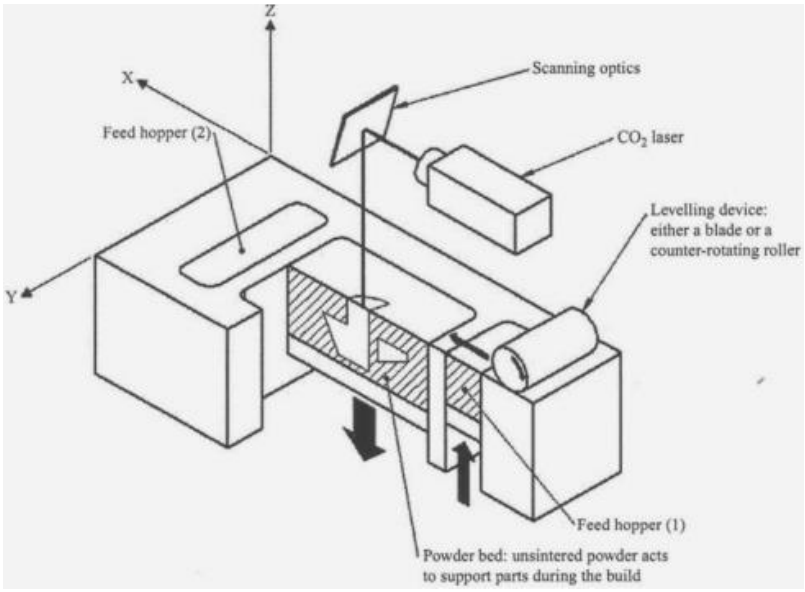



**Figure 3 SLS and DLMS printer (Cooke *et al.* 2011)**

Some manufacturers add a focusing lens to control the intensity of the laser beam. The laser source and laser intensities are dependent on the type of material used. The SLS/LS process uses a $CO_2$ laser to produce polymer parts, whereas the DMLS process makes use of a solid state, ytterbium type laser (Akande *et al.* 2016) (EOS GmbH 2017) DMLS is more suited to manufacture parts using metal or alloy powders whereas SLS is suited to manufacture parts from polymer and ceramic powders. The DMLS method also makes use of different types of inert gasses to prevent oxidation of the metals during the melting process. The majority of the SLS and LS machines use a nitrogen sintering atmosphere as different types of gasses can affect the density and quality of the sintered product (Asgharzadeh and Simchi 2005). Since the DMLS method does not require additional chemicals to assist in the melting process, it makes

it capable of producing 95% dense metal parts compared to the approximately 70% dense parts as produced using the SLS method (Custompartnet 2016a). Certain types of metal powders, however, require additives to improve the surface finish and sintering process.

- **Electron beam melting method**

The EBM technology is mostly used to manufacture metal and metal alloy parts. The EBM method is substantively like the SLS and DLMS, with the main difference being the heat source. The EBM technology uses an electron beam as its heat source and is carried out in a vacuum chamber (Everton *et al.* 2016). Illustrated in Figure 4 is the operation of the EBM process. The electron beam is computer controlled and guided using electromagnets, which moves the beam over the required areas.

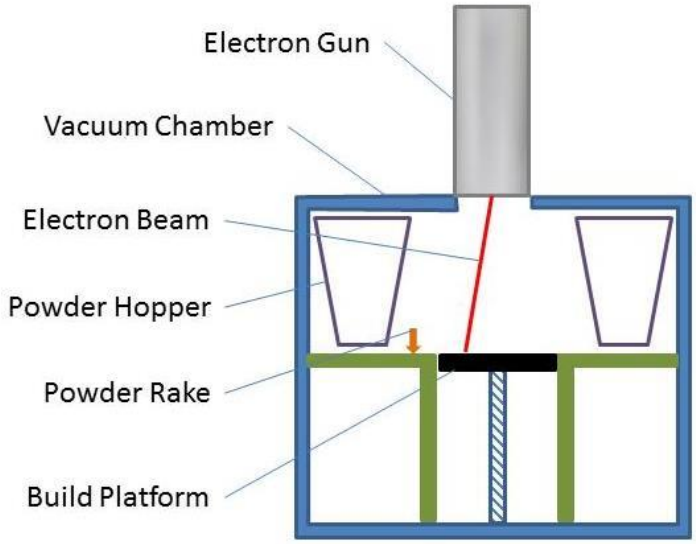



**Figure 4 Electron beam melting 3D printer (Dinwiddie *et al.* 2013)**

- **Binder jetting**

The Massachusetts Institute of Technology (MIT) developed the BJ technology in the 1990s. It was initially named the 3D printing process (3DP) but was later changed to binder jetting, since the term 3D printing is a more generic term used for AM processes (Sachs *et al.* 1992). Two very prominent companies in the AM sector, namely ExOne

and ZCorp have bought licensing rights to the BJ technology (Gibson *et al.* 2014). The primary components of the binder jetting technology are illustrated in Figure 5.

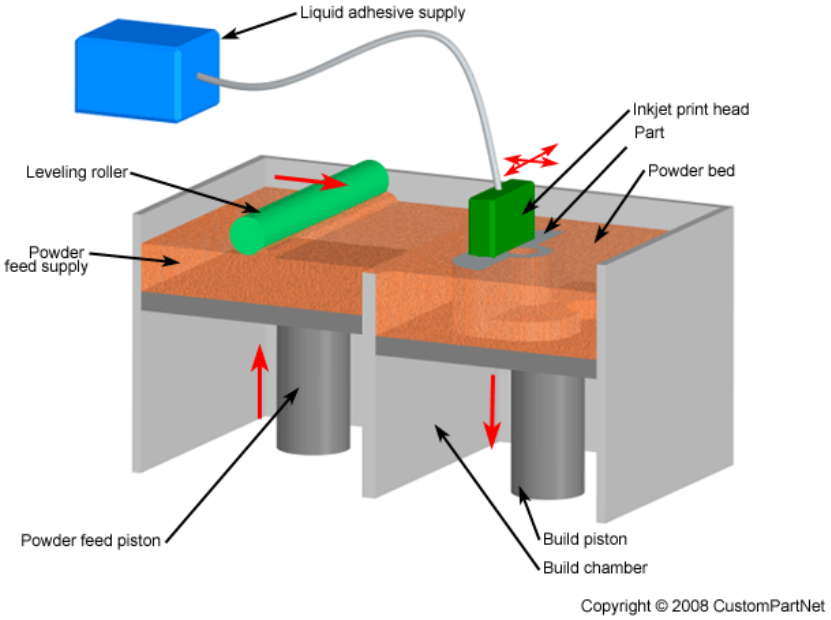



**Figure 5 Binder jetting printer (Custompartnet 2016b)**

The BJ process can manufacture parts using different types of powders such as polymers, plastics, ceramics, ferrous and non-ferrous metals as base materials. It uses a binder or adhesive solution to bond the powder particles together to create the desired shape or part. The BJ process uses inkjet technology to deposit the binder or adhesive solution to the required areas.

Once the printing process had been concluded, the part can then be removed and cleaned. Usually, the unused powder can be reused but must be sieved for solid particles. In some cases, fresh virgin powder needs to be mixed with the reused powder to ensure good particle bonding. The 3D printing or BJ process allows for fast printing speeds of up to four layers per minute (Custompartnet 2016b).

The binder jetting technology is primarily used for rapid prototyping applications, as only limited functional testing is possible on green (unprocessed) parts as the part strength is not as good as the other types of AM technologies directly from the machine. The parts can be infiltrated with sealants such as epoxies, wax or a silicone

resin to increase the strength of the part and to give it a good surface finish, if required (Fu *et al.* 2013).

## 1.2 The justification for this research

Currently, AM platforms do not have re-coating quality feedback into the printing system. It means that when re-coating errors or defects occur on the powder bed surface, there is no way to automatically determine if there are any anomalies present on the powder bed. These re-coating errors and defects can cause defects in the printed parts, as demonstrated in Figure 6. Often when parts have to be reprinted, it becomes costly as raw materials, electricity and time are wasted due to unusable parts.

**Figure 6 Effect of re-coating errors**

The VUT SGSTP discovered that the effects of re-coating errors could directly be linked to failed prints. When builds fail, they reduce the overall efficiency of the machine and increase the number of rejected parts due to imperfections on the part. Both sintering machines and binder jetting machines have been experiencing build failures due to re-coating errors.

A hypothesis is made that if these AM machines do have re-coating quality feedback into the system, the appropriate corrective measures can be taken on re-coating errors and defects as soon as they occur. Rectifying the re-coating errors could reduce the

number of failed prints that would increase the overall system efficiency. The following study will primarily focus on the Voxeljet VX500 technology.

## 1.3 Voxeljet VX500

The Voxeljet VX500 as displayed in Figure 7 was selected as the platform of choice to perform the experiments on for this research. The Voxeljet VX500 is a powder bed fusion AM machine that can prints parts using an acrylic type plastic called PMMA (polymethyl-methacrylate) (Voxeljet 2017). This type of AM technology uses a binder fluid or adhesive to fuse the plastic particles together to create the geometry of the part. Some of the bigger models in the Voxeljet series of machines can print parts using foundry sand to create moulds for making metal castings. The Voxeljet technology machines have a printing resolution of up to 600dpi and use a layer thickness of 150um for the PMMA plastic and a layer thickness of 300µm for the foundry sand (Voxeljet 2017). This technology is also more environmentally friendly as the residual PMMA plastic material can be re-used multiple times in the printing process.


**Figure 7 Voxeljet VX500 machine(Voxeljet 2017)**

The Voxeljet VX500 has a build envelope size of 500 x 400 x 300mm. Multiple parts could be stacked inside the build volume to optimise the building process.

## 1.4 Problem statement

Currently, powder-based AM technologies have no recoating quality feedback into the system, and this can be identified as one of the key aspects increasing the part rejection rate. Thus, the lack of an integrated recoating error detection system, to provide feedback to the system processor exists, and needs to be investigated.

## 1.5 Objectives

The main objective of the research is the development of a system that can detect re-coating errors during the printing process to increase the efficiency of powder-based AM technologies. The specific objectives of the following research project include:

- Identify the different re-coating errors and causes and evaluate existing types of active monitoring systems to identify re-coating errors.
- Design a prototype model of the selected active re-coater monitoring system.
- Build the prototype model of the active monitoring system and integrate it into a powder bed fusion system to gather print and error data.
- Evaluate the data captured from the integration of the active re-coater monitoring system to determine the efficiency thereof.

## 1.6 Research methodology

- An in-depth literature review will be done on the re-coating problems encountered in powder bed fusion technologies to narrow down the scope of the problem. This literature review will also help to identify previous research regarding the identified problem and the remaining shortfalls. The literature review will be conducted throughout the entire research process to keep up with any new developments in the field and any possible new research that may become known.
- Investigate and analyse powder bed fusion technologies to gather data on re-coating errors and their causes. This investigation and analysis can help to identify the exact causes of the recoating errors, and to visualise the effects of recoating errors.

- Gather information on possible methods that can be used to verify the quality of the powder bed surface autonomously.
- Evaluate the different methods that can be used to verify the quality of the powder bed surface. The effectiveness and cost-effectiveness of a method will be the most determining factors.
- Design an active re-coating monitoring system that can visually verify the quality of the powder bed.
- Build a prototype model of the system and integrate it into the Voxeljet VX500 to test the effectiveness of the system under diverse replicated conditions. The testing of the prototype model will help to identify any potential problems with the system and solve them to ensure the system works as expected.
- Test the re-coating monitoring system under production conditions to ensure that it can effectively detect defects during actual build conditions.

## 1.7 Delimitations

- Due to the closed source code and proprietary nature of the AM machines, the system could not be fully integrated into the Voxeljet VX500. Thus, the system was designed to give only a visual indication when errors were detected.
- Off-the-shelf lighting components were identified that conformed to the pre-defined requirements.
- Compiling and installation of the software packages did not form part of the discussion as there are manufacturer recommended walkthroughs and guides for these operations.
- The design methodology for the mounting brackets did not form part of the research as the mounting brackets were custom designed by a design team and used as off the shelf components.

## 1.8 Overview of dissertation

Chapter 2: In this chapter, a literature review was done on different types of active monitoring technologies that could be used for the monitoring of the powder recoating quality.

Chapter 3: This chapter focused on comparing the advantages and disadvantages of the different monitoring technologies, and the most suitable technology was selected for this research. The technical requirements and software specifications for the technology were also determined. A test rig was built to evaluate the small feature detection capabilities of the camera. The smallest features that can be detected by the camera was then determined using the 1951 USAF Resolution Chart.

Chapter 4: This Chapter focused on the integration of the active re-coating monitoring system into the Voxeljet VX500 machine. Once the system was integrated in the VX500, the image processing techniques could then be evaluated. The image processing program was then tested using a series of replicated defects on the powder bed to verify the program's capability to detect defects, as well as to determine the optimum parameters required by the various image processing algorithms being used. The system's capability to detect defects under production conditions was then validated using actual build jobs. A series of case studies were performed during actual builds on the VX500 to record data using the system under production conditions. These case studies were performed to validate the capability of the designed system to detect defects and recoating errors during an actual build. The data recorded during these studies were also used to determine whether the threshold value determined during the testing phase was high or low enough to detect defects during an actual build.

Chapter 5: This chapter contains all the conclusions of the research as well as the recommendations for future work.

## 1.9 Summary

This chapter highlights the background of the study, the literature around the different types of additive manufacturing, the basic operation of powder bed fusion technologies, and the different particle fusion methods. The justification of the study outlined the encountered problem in powder bed fusion technologies, as well as the causes and effects arising from this identified problem. From this, the problem statement was formulated based on the problems identified in the purpose of the study. Using the justification of the study and problem statement, the primary objective, as

well as the specific individual objectives, were identified. A research methodology was drawn up to explain how the researcher aims to achieve the different objectives. In conclusion, an overview of the conducted research was also given.

# Chapter 2 Active Monitoring Systems

## 2.1 Introduction

In this chapter, a literature review will be done on different types of active monitoring systems that could be used to monitor the condition of the powder bed surface autonomously. Some of the characteristics that need to be taken into consideration include scanning accuracy/resolution, hardware and software complexity, processing speed, the processing power required and the cost implications of each method. The three active monitoring technologies that will be investigated for their suitability for the autonomous surface inspection of the re-coated surface include, computer/machine vision, 3D laser scanning and a hybrid system of machine vision and laser mapping (TOF camera).

## 2.2 Computer vision

A widespread method used to identify objects or visual patterns is called image processing, or the more technically correct term is computer vision. The Access Science website defines computer vision as: "The technology concerned with computational understanding and use of the information present in visual images" (AccessScience 2015). Based on the definition, computer vision can broadly be described as the automatic analysis of images or videos by a computer or processor to develop some understanding of the world around it (Dawson-Howe 2014). The ability to use computer vision to describe scenes is demonstrated in Figure 8 by a collaborative project between Stanford University and Google (Li *et al.* 2009).

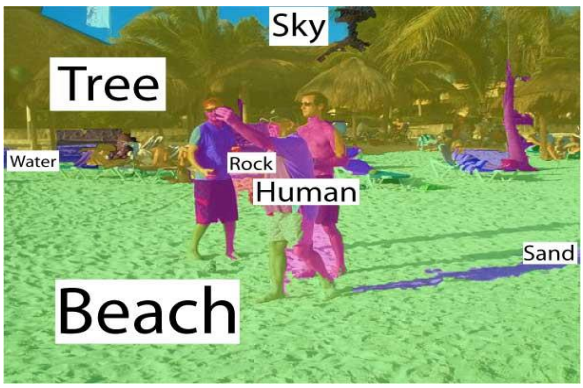



**Figure 8 Scene description (Karpathy and Li 2015)**

Computer vision can be used for a variety of applications and is not limited to facial recognition, number plate recognition, edge detection and pattern recognition. Computer vision systems and algorithms are often used in conjunction with artificial intelligence algorithms such as neural networks to detect patterns or specific objects captured inside the images. One of the applications where computer vision is widely being used is to perform quality control or process monitoring on assembly lines. The equipment required for computer vision is very simple, as the basics only require a computer and a camera such as a webcam. For more advanced applications or if small form factor equipment is required, there are specialized stand-alone processing units and cameras as demonstrated in Figure 9.

**Figure 9 Standalone processing unit and cameras**

There are a variety of cameras available that can be used for computer vision, such as colour, monochrome, infrared, thermal and line scan cameras. Most of these cameras can be equipped with a wide-angle or telephoto lens. The camera specification mainly depends on the application, as well as the environment the system will be required to operate in. The two most common types of cameras found in industry are area scan or line scan cameras, demonstrated in Figure 10.

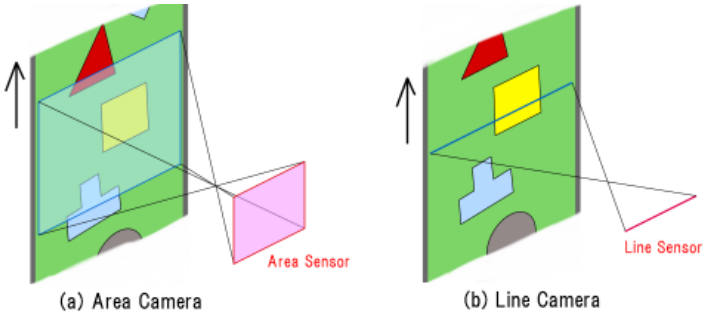



**Figure 10 Area scan versus line scan(Elm 2011)**

- **Area scan cameras**

An area scan camera captures an image of an area as demonstrated in Figure 10. This is the most popular type of camera in computer vision and for capturing videos and images for general photographic and videography purposes. The amount of lenses and accessories that are available for area scan cameras makes them very versatile. Most applications for computer vision require a very small resolution (640x480 pixels) as for example the PlayStation Eye camera that is used to track human body movement (Engadget 2016). A camera that has a higher resolution is required when very fine detail in the image needs to be examined, or if the subject or object to be imaged is at a distance. One of the aspects that needs to be considered is that an increase in the resolution will increase the required processing power to process and analyse the image.

- **Line scan cameras**

A line scan camera is a specialised type of camera that is used for very specific applications. This type of camera has one or two rows of pixels but can capture an image line by line in very high resolution and quality as demonstrated in Figure 10. Line scan cameras are mostly used for inspection of very large objects or when the object is moving. One of the applications of a line scan camera is to examine a roll of plastic or fabric for defects as demonstrated in Figure 11.

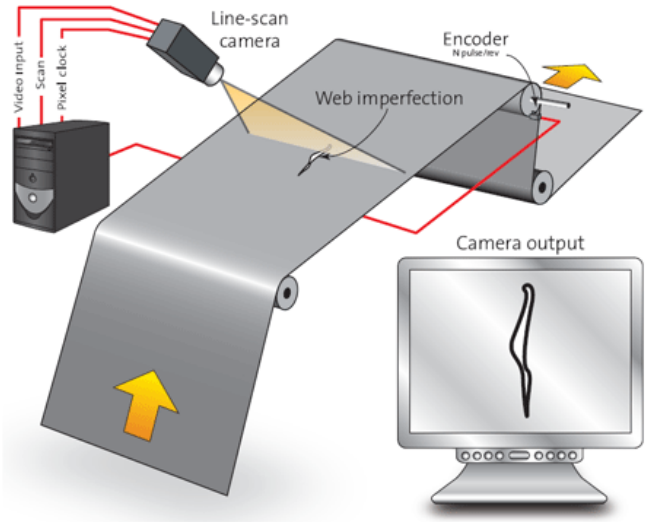



**Figure 11 Line Scan camera defect detection(Stemmer Imaging 2017)**

A disadvantage of line scan cameras is that multiple camera units need to be used, side by side, in order to cover a wider scan area. This is due to the limited FOV (field of view) and cost of line scan cameras. Another common use for line scan cameras are in photocopiers and scanners. They can also be used to perform OCR (Optical Character Recognition) from printed or written documents (Teledyne Dalsa 2014). The camera can also be triggered to capture images at specific intervals by an external encoder as demonstrated in Figure 11 or can be used in free run mode where it is continually capturing images.

Line scan cameras configured in free run mode could be used to simulate an area scan camera. However, the captured line images need to be stitched together to produce a complete image. This may produce blur or unwanted image defects as the camera captures the image line by line, and if an object moves in the same direction as the camera it may produce unwanted but humorous results. However, in the artistic photography market, these unwanted effects may be desirable as it can be used to create specific effects, thus giving it applications beyond just the industrial computer vision sector.

### 2.2.1 Computer vision case studies

In order to determine the suitability of the computer vision technologies, multiple case studies will be reviewed from literature. The following is a review of case studies of various applications of computer vision, as well as the methods and equipment used to achieve the desired objectives.

#### 2.2.1.1 Tile alignment quality control

In a study done by Lin and Fang, from the I-Shou University in Taiwan, the effectiveness of using computer vision to determine tile alignment as well as the grouting gaps of the installed tiles were tested (Lin and Fang 2013). To determine the proper alignment of the tiles as well as the grout width, an image was captured of the installed tiles using a normal point-and-shoot camera. The image was then processed and analysed by software to detect the corners of the tiles in the image. A baseline dataset was defined in software, with a minimum and maximum values for the width

of the grouting gaps. If the grouting gaps are not uniform around the tile, it can be deduced that the tile has been installed skew. The edges detected would be fed into the algorithm along with the baseline values and a score would be generated. This score is measured on an expert scale that ranges from 0 to 100, where 0 is a very poor installation, and 100 would be considered an excellent installation.

#### 2.2.1.2 Inspection of tiles for colour and structural defects

The second case study was done to aid in the quality inspection of ceramic tiles for colour and structural defects. Computer vision was used to detect any discolouration and texture defects, as well as small bumps, cracks, depressions and dirt that may be stuck on the tile (Boukouvalas *et al.* 1995). Traditionally, manual visual inspection of the ceramic tile is performed by an individual, but this slows down the production line as each tile must be inspected individually. This type of inspection is also not very objective as the inspection is subjective to the discretion of the person inspecting the tiles. When using computer vision for quality assurance of the tiles, the system needs to be fast enough to inspect tiles at a rate of approximately two tiles per second. The system then classifies the tiles according to two parameters, namely colour grading and defects. Depending on the number of defects, the tiles are then graded into three different categories.

#### 2.2.1.3 Non-destructive inspection of large calibre gun barrels

The third case study is based on the non-destructive inspection of large calibre gun barrels for numerous types of defects using image processing techniques (Shanmugamani *et al.* 2015). Surface images of the gun barrel's inside is captured and classified in a two-stage process using a variety of techniques. The surface defects were first segmented using extended maxima transform and then the size of the surface defects was calculated in metric units. Textural features are extracted from the image based on histogram and grey level co-occurrence matrix data, to select the best minimal features for the classification purposes. Once the segmentation process has been completed and the textural features have been identified, the surface defects can be classified using various classification algorithms. This classification process is necessary to be able classify the surface defect according to the cause of the defect.

Overall, the study aims to provide a framework from which to identify surface defects and classify them to help automate the inspection process.

#### 2.2.1.4 Re-coater blade wear inspection

The fourth case study is a study that is very much relevant to the identified research problem. The study was done to determine the effectiveness of computer vision to detect damage or wear on the re-coater blade and the effect it has on the powder deposition in a DMLS machine (Craeghs *et al.* 2011). The camera was used with a specific lighting setup, displayed in Figure 12, to highlight any irregularities in the powder bed.

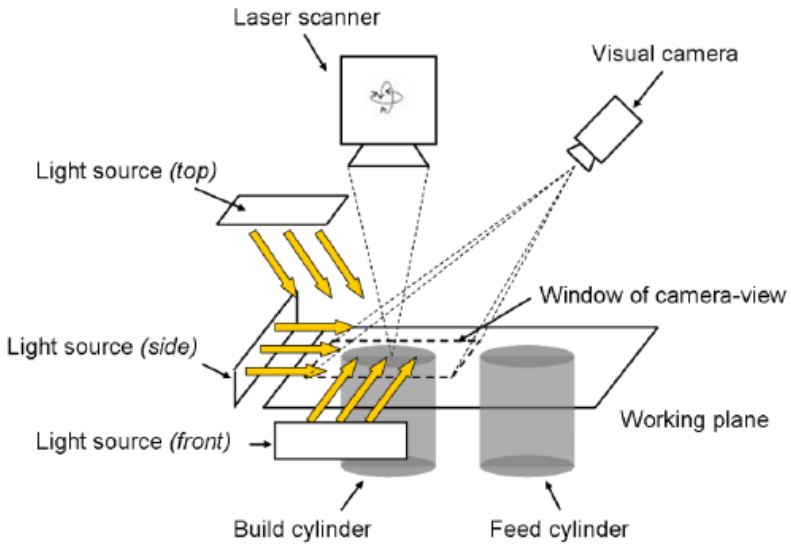



**Figure 12 Lighting setup for visual inspection of powder bed(Craeghs *et al.* 2011)**

To verify the integrity of the powder bed, a benchmark line profile grey value of the powder bed had to be calculated under normal conditions. Once the benchmark value has been established, the grey value of the captured images of the powder bed could be verified against the benchmark value. This was done after each re-coating operation until the part or parts have been completely sintered. If an irregularity occurred during the build process, in the form of a line or part warping, the grey value of the image will change, and depending on the set threshold of the grey value, the machine operator will then be notified.

#### 2.2.1.5 In situ surface monitoring of laser powder bed fusion systems

The following case study is also directly related to the identified research problem. A group of researchers from the University of North Carolina devised a method to monitor the condition of the powder bed of a metal powder-based laser AM machine using computer vision and fringe projection. The aim of the research was to develop a system that could monitor the condition of the powder bed after a new layer of powder has been deposited onto the build area. In addition, the system also inspected the area after it has been sintered to detect any defects such as spatter or warping of the part (Zhang *et al.* 2016). The warping of the sintered area or part can have a potentially detrimental effect on the re-coating process as the re-coater blade could scrape the part, causing damage to the re-coater blade as well as damaging the part itself. The equipment that was used in the system included a camera with a resolution of 4096 × 2160 pixels with a 50mm lens and a commercial DLP (Digital Light Processing) projector with a customized projection lens to project the fringe pattern onto the build area. The wavelength of the projector had to be calibrated, due to the angle of the projector, as demonstrated in Figure 13 to ensure the 3D topography of the powder bed is accurate.

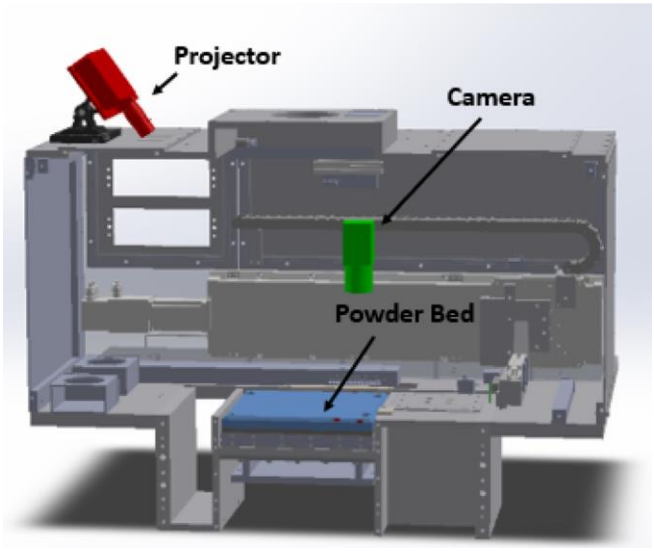



**Figure 13 Setup of the camera and DLP projector (Zhang *et al.* 2016)**

The system will capture an image of the powder bed while a fringe pattern is being projected onto the powder bed. Using a series of mathematical formulas, the surface

topography of the captured area was calculated using the projected fringe. This operation was performed before and after the laser beam has sintered the powder particles together as demonstrated in Figure 14.

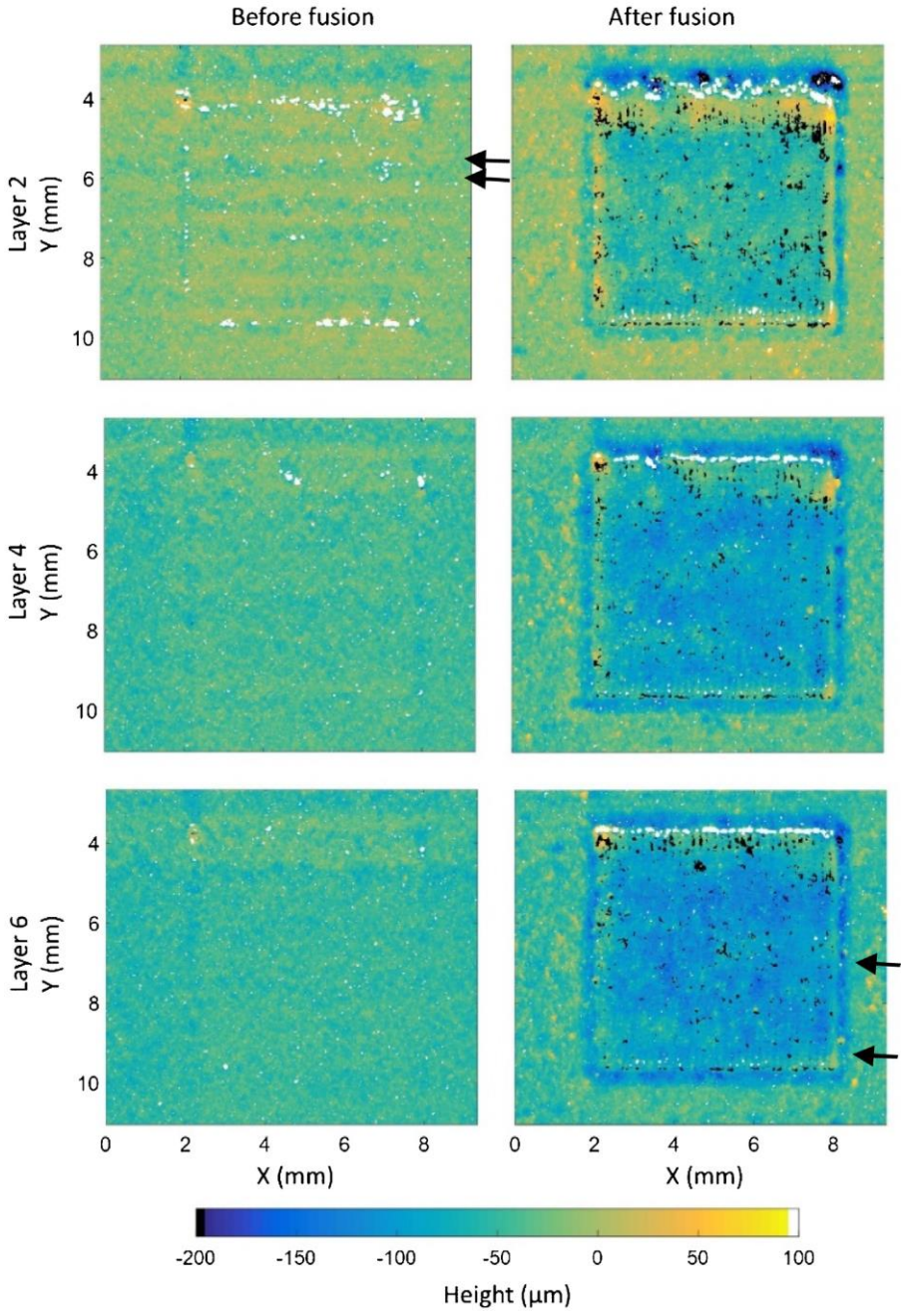



**Figure 14 Height maps of the powder bed before and after fusion(Zhang *et al.* 2016)**

As demonstrated in Figure 14, using the fringe projection methods, the variations on the powder bed can quite accurately be measured even down to a micro-meter scale.

The changes in colour indicates that the system could detect surface topography changes on the powder bed. The white dots on the image is a result of too much light and the black dots were a result of too little or no light, which causes and inaccurate distance measurement. However, this does not affect the overall effectiveness of the measurement as the standard deviation of the dataset was still within acceptable limits and usable for accurate measurement.

As could be seen from the various case studies, computer vision has a variety of potential applications and can be used for many purposes. The advantages and disadvantages of computer vision with reference to the inspection of parts/surfaces were identified as demonstrated in Table 1.

**Table 1 Advantages and disadvantages of computer vision**

| Advantages | Disadvantages |
|---|---|
| • Relatively low cost<br>• Changes are mostly implemented in software<br>• Large variety of sensor resolutions are available<br>• High scanning speed<br>• Very accurate | • High resolutions require more processing power<br>• Processing is time consuming<br>• Limited FOV<br>• Sensitive to lighting conditions |

From the case studies reviewed, information on how to identify defects present on a flat surface as well as the classification of defects from an image were reviewed. The last two case studies also looked at how to identify irregularities on a powder bed surface. Although the last two studies focused primarily on the metal AM machines, a lot of the principals used to detect defects on the powder bed is common across powder-based AM machines.

## 2.3 3D laser scanning

There are several different types of 3D laser scanners available on the market, ranging from small handheld scanners as demonstrated in Figure 15, and larger units that are used in area and architectural surveying. 3D laser scanners can also be used to reverse

engineer hard to obtain parts or parts that are at their end of life and are not being manufactured by the OEM or third-party manufacturers anymore.


**Figure 15 Handheld 3D scanner(Z-Corp 2009)**

Laser scanning is a technology that uses a line or a dot type laser to digitally create a point-cloud map of the object being scanned. A point-cloud map can be used to create a 3D model of the object or area being scanned (LaserDesign 2016). Laser scanning requires that the surface or part is clean and free from any visible defects such as dust or dirt as this can interfere with the scanning process and create noise or unwanted deformities in the scanned 3D model. The technology has an accuracy of 2-6mm at a distance of 50m for the devices used in building surveying, but have a much higher accuracy for the shorter range devices (Olsen *et al.* 2010). Laser scanning is not as accurate as the traditional contact-type scanner systems, but have much faster scanning speeds and provides a larger amount of measurement data (Wang and Feng 2014).

3D laser scanning methods can be classified into 3 main categories: time-of-flight, phase shift and laser triangulation. Although these methods of laser scanning are usually used independently, more than one method can be combined to create a hybrid system that is more effective and efficient than only one method being used in isolation (Ebrahim 2015).

- The time-of-flight method uses a laser to send a pulse of light to the object or surface being scanned and then uses a high precision timer to measure how long it takes the light pulse to travel from the laser emitter to the object and back to the photodetector, as demonstrated in Figure 16. The time it takes for the light pulse to complete a round trip can be used to determine the distance of the point relative

to the photodetector, as the speed of light is a constant. The accuracy of this method is dependent the accuracy of the timer.

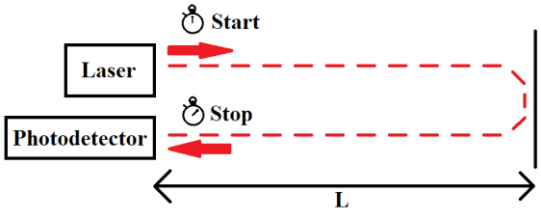



**Figure 16 3D laser scanner time-of-flight method operation**

- The phase shift method also consists of a laser emitter and a photodetector, similar to the time-of-flight scanner, but instead of measuring the light pulse flight duration, the phase shift method compares the phase change between the incoming and transmitted light pulses. A light pulse of a specific wavelength is transmitted and then the reflected light pulse is compared against the emitted light pulse wave to determine the phase shift between the two pulses. The phase shift is then used to calculate the distance. Figure 17 demonstrates the operation of the phase shift method.

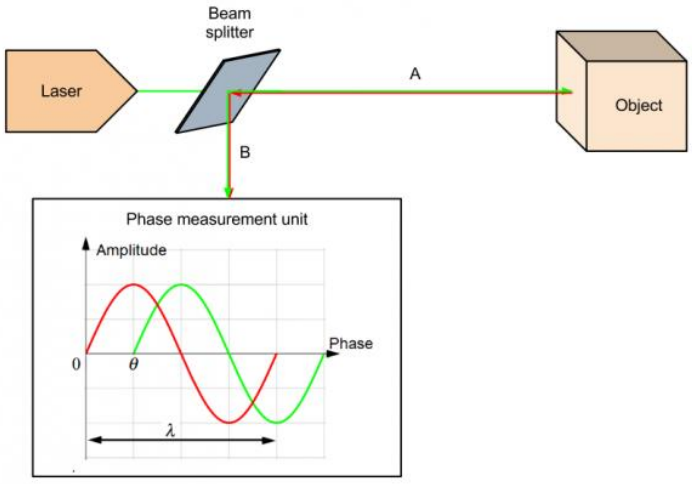



**Figure 17 3D laser scanner phase shift method of operation(Roboticlab 2013)**

- The triangulation method uses a laser beam and camera to determine the location of the laser beam on the target object. The camera is positioned at a predefined angle to the laser emitter and depending on how far the object is from the camera,

the laser dot or line will be projected at a different position on the camera's field of view. The system's operation is demonstrated in Figure 18.

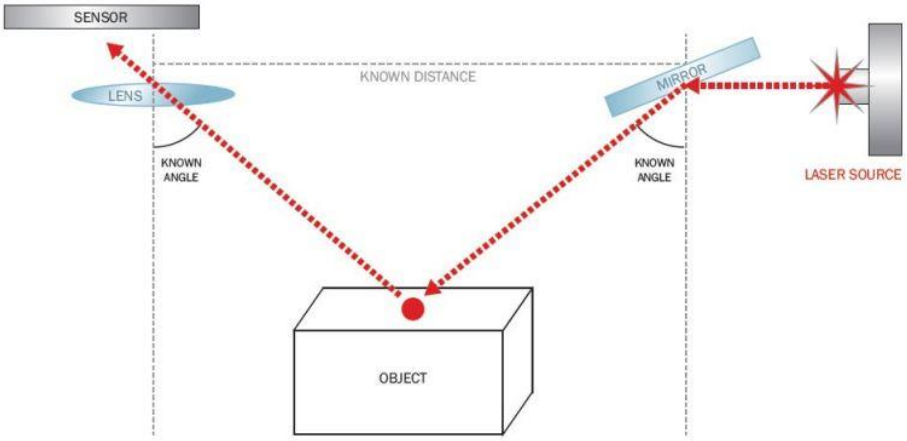



**Figure 18 3D laser scanner triangulation method of operation(Robinson and Hardin 2015)**

The distance between the scanner and the object can be calculated as the following parameters are known: the distance between the laser emitter and the camera, the angle of the laser emitter and the position of the laser dot/beam through the camera's field of view. The three laser scanning methods each have their strengths and weaknesses. The advantages and disadvantages of the three different methods are displayed in Table 2.

**Table 2 Laser scanning advantages and disadvantages**

| Technology Type | Advantages | Disadvantages |
|---|---|---|
| Time-of-flight | • Can operate over long distances | • Not very accurate (Ebrahim 2015)<br>• Very costly<br>• Low point density |
| Phase shift | • High point density<br>• Works well indoors | • Has a short range<br>• Very costly |
| Triangulation | • Is very accurate (FARO 2016b)<br>• Very compact<br>• High point density | • Has a very short range<br>• Very costly |

### 2.3.1 3D laser scanning case studies

To determine the suitability of using laser scanning technology in this research project, multiple case studies will be highlighted from literature. These case studies will highlight the applications where laser scanning was used as well as the methods and equipment used to achieve the desired objectives.

#### 2.3.1.1 Pre-cast concrete element inspection

In a study, done by (Wang *et al.* 2016), TLS (Terrestrial Laser Systems) were used to present a quality assurance technique of digitally measuring the dimensions of pre-cast concrete blocks or elements, which will be compared against a 3D model to determine any irregularities in the surface geometry. Traditional methods are very time consuming and at times inaccurate. Thus, an automated method to speed up and increase the accuracy of the quality assurance process and inspection was proposed. A BIM (Building Information Modelling) model, that conforms to the appropriate building code, is used to determine the dimensions of the element in its as-designed state. The TLS is then used to scan the pre-cast element to create a model of the element in its as-built state. Once it has been scanned, the as-built model is compared to the BIM as-designed model and based on the differences between the reference model and the scanned model, the part will be approved or rejected. This study also investigated two types of scanning methods, the first being the direct scanning of the element using the TLS, and the second being a mirror-aided scanning approach, with the aim to help reduce the incident angles in real scanning environments.

#### 2.3.1.2 Surface regularity/flatness quality control

The second study that will be reviewed is a proposed method to use TLS systems to automate the surface flatness quality control in the civil engineering and construction industry (Bosché and Guenet 2014). Developments in TLS systems and BIM has started offering great opportunities for increasing efficiency and completeness of dimensional control operations. However, one of the concerns that have prevented the widespread use of TLS systems for level measurement is that there are some concerns about the measurement accuracy it provides as well as the amount of time that is

required to manually process the data. Thus, the study presents a novel method that can be used to integrate TLS and BIM to assist in the automation of the processing of the TLS data. There were two standard flatness control techniques that were used, and the new novel method was also proposed and tested. Since these techniques will be of significance for this research, the techniques used in this case study will be discussed in detail in the next section.

#### 2.3.1.3 Surface flatness checking techniques

In the case study done on surface regularity/flatness quality control, there were two surface flatness checking techniques that were used to verify the flatness of a concrete floor. Both of these techniques are of some significance as they have the potential to help develop a solution to the identified research problem.

As mentioned in the case study the two methods that are commonly used to determine the global and local flatness of a poured concrete surface are the straight edge and the F-numbers methods (Bosché and Guenet 2014). There are other methods that could also be used such as the TR34 method and the Waviness Index method. However, they did not form part of the methods that were reviewed in the case study.

The straight edge method is a traditional method where a straight edge and a stainless-steel slip gauge is used to check the surface flatness and levelness on certain points on the floor. This is one of the most commonly used methods in the civil engineering industry to check the global and local flatness of poured concrete floors due to the simplicity of the method and basic construction equipment required. The global flatness refers to the overall flatness of the floor across the entire surface area, the local flatness refers to a smaller, more specific area such as a doorway or the corner of a room. To control the global flatness of the floor a straightedge of typically 2-3m long is used, and to control the local flatness of the floor a straightedge of 0.2-0.3m long is used. The standard deviation of the measurements is then compared against the tolerance values as specified in the building standards. There are some deficiencies when using this method to do manual measurements, namely: the difficulty of checking large floor areas, the difficulty of randomly sampling floors and the inability to reproduce accurate testing results. Based on the above-mentioned deficiencies,

alternative methods were developed to measure the flatness of poured concrete floors, one of these being F-numbers method. When using the straight edge method, there are three patterns presented in the case study that can be used to generate the straightedge measuring positions. These three patterns are Random, Grid-Square and Grid-Star. The random pattern created 100 straightedges at random positions on the scanned point cloud within certain boundary limits as shown in Figure 19.

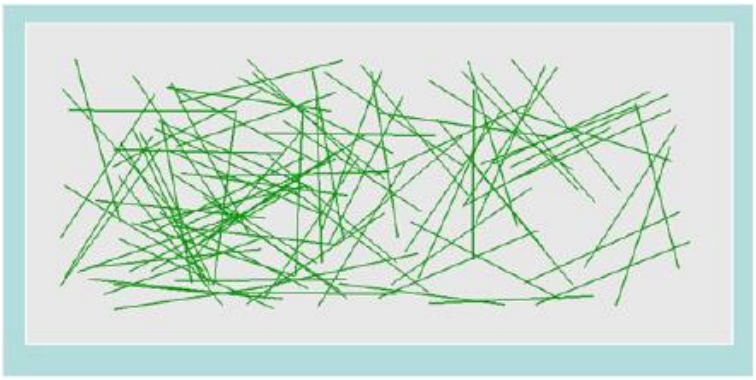


**Figure 19 Random generation of straightedges pattern(Bosché and Guenet 2014)**

The second pattern is the Grid-Square pattern. This pattern creates straight edges on the point cloud inside the boundary limits in the shape of a grid-square as demonstrated in Figure 20.

**Figure 20 Grid-square straightedges pattern(Bosché and Guenet 2014)**

The third pattern is the Grid-Star pattern. This pattern creates straight edges on the point cloud similarly to the Grid-Square pattern, but instead of creating straight edges that are at 90 degrees to each other, this pattern creates straightedges at specified

angles. E.g. in Figure 21 the straight edges are created at 10-degree angles from the origin point but still within the boundary.

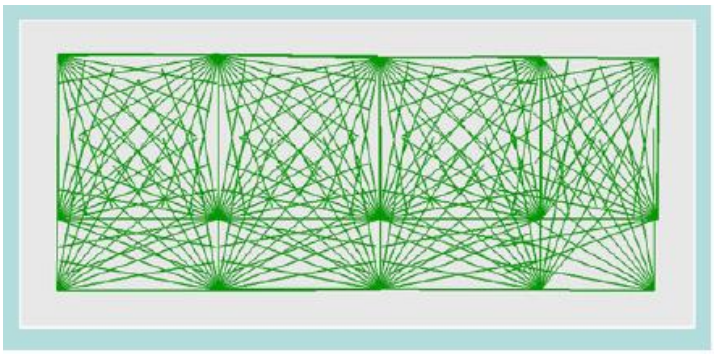


**Figure 21 Grid-star straightedges pattern with α = 10º(Bosché and Guenet 2014)**

Using these three patterns as demonstrated in Figure 19 to Figure 21 provided a very effective way of determining the levelness of the floor using the straight edge method as the results obtained using the laser scanner could be easily compared to the measurements that were made manually.

The second method used to verify the levelness of a poured concrete floor or slab is called the F-numbers method. This method is mathematically more complex but is easier to implement than the straightedge method and is also considered to be a more complete method. The tools that are best suited for this method includes optical levels, inclinometers or longitudinal differential floor profilometers. The F-numbers method is a mathematical method that summarizes a floor profile with two numbers. The first number is a statistically calculated number that considers the mean and standard deviations of measurements in 0.3m incremental curvatures, this estimates the floor's global flatness. The second number is also a statistically calculated number that also considers the mean and standard deviations of measurements in 3m elevation differences and using this value the floor's levelness can be estimated.

The F-numbers method of checking the levelness of a poured concrete floor or slab uses sampling lines that are defined in two orthogonal directions. The dimensions of the cast floor determine the orientation of the scan lines as demonstrated in Figure 22. The reason that the pattern varies is due to the dimensions of the slab. This method requires that the floor section to be measured must be of a minimum size. For instance,

the shortest side must be a length of at least 2.4m long and the area of the slab must be at least 30m$^2$. The parallel sampling lines must also be at least 1.2m apart from each other, and there must be a sampling point at least every 0.3m.

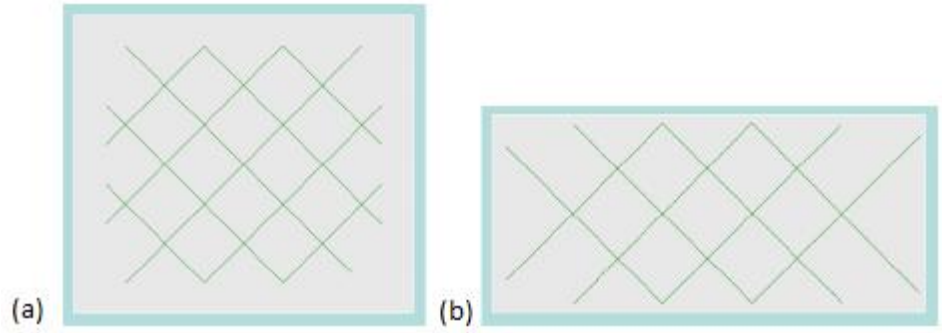



**Figure 22 F-numbers method sampling lines(Bosché and Guenet 2014)**

Some of the methods used in the article to check the levelness and surface flatness of a concrete floor can be modified and used to detect any irregularities in the powder bed, even if there are only slight deformations in the powder bed. It can also be useful to characterize and identify the defects in the powder bed as some defects may require immediate intervention, whereas other defects might be ignored as they won't affect the integrity of the part.

It is worth pointing out that the reviewed case studies primarily focussed on the construction industry and demonstrated that the scanning accuracy of the technology could be considered as high. It was found that accuracies of up to 0.015mm could be achieved at a measuring distance of two meters (FARO 2016b).

The advantages of using this technology are that it can scan a part or surface in very high resolutions and detail. It also has a very high scanning speed and can scan moving objects. However, some of the disadvantages include: when scanning a part, made of a highly reflective material or has a highly reflective surface finish, a lot of stray light rays are detected, and may produce multiple outlier data points in the scanned point cloud (Wang and Feng 2016). Another disadvantage is that the laser scanning equipment being used is expensive, as high-powered laser devices are being used. This also means that the proper safety procedures must be followed as laser devices can cause permanent eye damage.

## 2.4 Hybrid image/laser mapping

Hybrid image and laser mapping is essentially computer vision and 3D laser scanning combined to create a highly detailed 3-dimensional model of the object or area being scanned. The hybrid image/laser mapping method uses a special type of camera, called a time-of-flight (TOF) camera. A TOF camera produces both a colour or monochrome image and a depth image. Each pixel of the image is encoded with the distance of the specific point on the image being captured (Lefloch *et al.* 2013). A TOF camera combine the advantages of being able to scan parts using a laser scanner combined with the variety of options available with regards to computer vision processing and analysis techniques.

The 3D TOF camera works by illuminating the part or area/scene with a pulsed or modulated light source, as demonstrated in Figure 23.

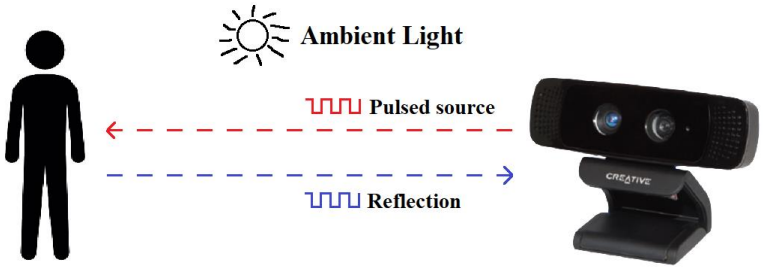



**Figure 23 TOF camera operation(Li 2014)**

The light source is usually a laser that operates in the near-infrared range of ~850nm, making it invisible to the human eye. There are two ways of measuring the time-of-flight of the laser light wave, the first being pulsed wave and the second being continuous wave (CW) modulation (Foix *et al.* 2011). Most TOF cameras available on the market make use of CW modulation. Pulsed wave modulation requires very fast electronic circuitry as a 1-millimetre accuracy requires an accurate pulse timing of 6.6 picoseconds. This presents a problem as it is nearly impossible with current silicon technology operating at room temperature to achieve this type of accuracy (Li 2014). When using the CW method, the laser light illuminates the scene at a specific frequency of light using a specific waveform and length. A photo sensor then detects the light reflected off the objects or parts. The photo sensor converts the light photons

into an electrical signal. Then the delay between the projected light and the reflected light is compared to each other as demonstrated in Figure 24.

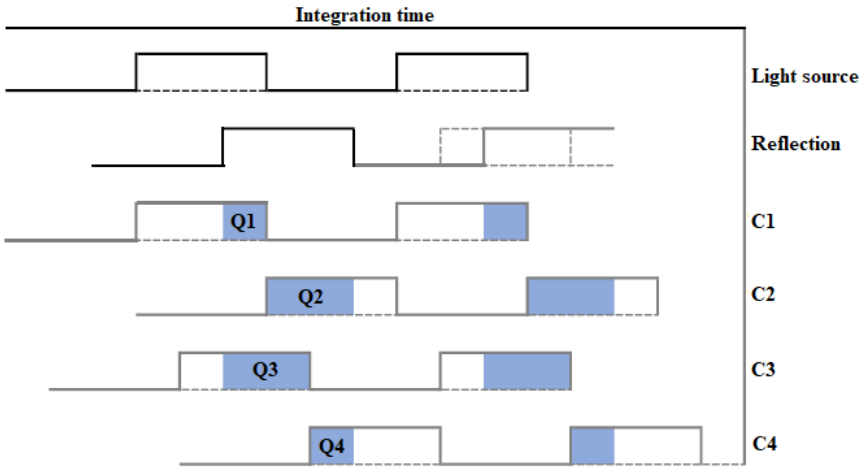



**Figure 24 CW time-of-flight modulation method operation(Li 2014)**

The CW modulation method takes four samples of the reflected light, each with a 90 degrees phase offset. Once the four samples have been collected, the delay between the illumination and the reflection is calculated, and from the delay between the waves, the corresponding distance can be calculated. This entire process is repeated for each pixel of the camera so that a corresponding depth map can be created using the laser and sensors.

TOF cameras are still considered to be a relatively new and emerging technology, however, its applications span from assembly line monitoring systems to the 3D modelling of parts or even buildings. It is not a mature technology such as some of the other image/camera-based depth measuring technologies or depth sensors, but it proves to be very useful in places where fast, but not as precise 3D environment data is needed. Some of these applications that do not require extreme precision are, obstacle avoidance, pose estimation and human body parts recognition, to name but a few (Foix *et al.* 2011).

An application where TOF cameras can be used effectively is in the Closed-Circuit Television (CCTV) industry. A TOF camera can be used as part of a monitoring system for valuable items displayed or exhibited in a public space. This means that

less visible security personnel are required for guarding the item or even areas. This can also be used in R&D centres, prisons and even banks to prevent the tailgating of authorised personnel into restricted areas. What makes this type of surveillance more unique than traditional camera is that facial recognition can be performed not only on a 2-dimensional image, but can be performed in 3D, giving the surveillance or access control system a further level of verification. The company ICD Security Solutions also proposed to use TOF cameras to do people counting in very busy and crowded places like shopping centres, malls, subway stations and even airports (ICDSecurity 2012).

The advantages of TOF cameras is that they can combine a 2-dimensional image with a 3D model to a relatively high degree of precision and accuracy. The camera is also capable of scanning images at the same frame rate that the normal camera is capturing images at. One of the disadvantages of the TOF cameras is that the accuracy is currently only limited to a variance of about 1 cm (Basler Ace 2016). This is due to the fact that current silicon technology at room temperature cannot cope with timing a light pulse into the picosecond region (Li 2014).

## 2.5 Summary

In this chapter, three different technologies were reviewed to determine which technology would be best suited for designing an active re-coater monitoring system. The three technologies that were reviewed are computer vision, 3D laser scanning technology and the TOF camera. Several case studies for each of the technologies were also briefly reviewed to verify its suitability to monitor the powder bed.

In Chapter 3 these technologies will be compared to each other to determine the best suited technology for the active re-coater monitoring system. The chapter will also show the physical design of the active re-coater monitoring system.

# Chapter 3 Design of an Active Re-Coater Monitoring System

## 3.1 Introduction

In this chapter the three technologies discussed previously will be compared to each other to determine which technology is best suited for the active re-coater monitoring system. Once a suitable technology has been selected, it will be used as the foundation to develop an online monitoring system. The specific criteria that will be considered for the design of the active re-coater monitoring system will also be discussed as well as the software packages that will be used to develop the image processing program. A test model of the system will be built to benchmark the capability of the camera.

## 3.2 Comparison of technologies

The features, specifications, advantages and disadvantages of the three identified technologies addressed in the previous chapter, is demonstrated in Table 3. The three key features applicable to this research study includes the: measurement accuracy, cost of the proposed system, and the required computing power.

**Table 3 Technology features**

| Features | Technology type | | |
|---|---|---|---|
| | **Computer vision** | **3D laser scanning** | **TOF camera** |
| Measurement accuracy | High (less than 0.015mm (FARO 2016a)) | High (0.015mm (FARO 2016b)) | Low (1cm (Basler Ace 2016)) |
| Scanning speed | High (+30 fps) | High (16000 points/s) | Medium (+15fps) |
| Cost | Medium (R1000 – R15000) | High (R15000 – R300000) | Medium (R2500 – R30000) |
| Hardware complexity | High (Camera, Lights, DLP Projector, Processing Unit) | Medium (Laser Scanner, Processing Unit) | Low (TOF Camera, Processing Unit) |
| Software complexity | Medium | Medium | Low |
| Computing power required | Medium | High | Medium |

| Features | Technology type | | |
|---|---|---|---|
| | **Computer vision** | **3D laser scanning** | **TOF camera** |
| Advantages | • Very fine resolution.<br>• Very high accuracy.<br>• Requires very little equipment to set up a basic system.<br>• Capable of a very high scanning speed. | • Very fine resolution.<br>• Very high accuracy.<br>• Can scan moving objects. | • Can scan moving objects.<br>• Is not really affected by changing light conditions. |
| Disadvantages | • Can be affected by varying lighting conditions.<br>• Requires the surface or part to be stationary while the scan is in progress to prevent blur.<br>• Limited FOV (field of view) | • Produces ghost points when scanning highly reflective material.<br>• The laser scanning units are very expensive. | • It has an accuracy of 1cm under ideal conditions.<br>• Requires a high precision timing system. |

From the data shown in Table 3 it was determined that the computer vision and 3D laser scanning technologies could potentially be used to develop and active monitoring system for monitoring the powder bed on powder based AM technologies. The high accuracy and high scanning speed displayed by these technologies make them very well suited to develop an active re-coater monitoring system with. The accuracy of the TOF camera technology however, is not high enough for it to be effectively used in the design. Most powder bed fusion technology machines typically work on layer thicknesses from 150 microns for polymers and ceramic (Voxeljet 2017) and 300 microns for sand (Voxeljet 2015) depending on the manufacturer. This means that the active monitoring system must be able to detect imperfections on the powder bed of at least 300 microns and smaller. Since the TOF camera has a predicted accuracy of

+/- 1cm under ideal conditions, it will not be very well suited to the task as the accuracy is too low. By the time that the camera will have detected the defect on the powder bed under not so ideal conditions, the part integrity will be compromised already, and the part must be scrapped.

Considering the computer vision and laser scanning technologies, the accuracy of these two technologies are very similar to each other and is also accurate enough to be used as an active monitoring system. Since the accuracy is sufficient, the rest of the factors will be considered and discussed. The determining factors that will be investigated is the scanning and processing speed. The scanning and processing speed are very crucial factors as the active monitoring systems must be able to scan/capture the entire powder bed area after the re-coater has applied a new layer, but also before the printing head or laser/electron beam moves over the powder bed to bind the powder particles together. It is also crucial that the image or scanned model must be processed before the adhesive or laser is applied to the material. Since the two proposed technologies are different in their operation, they can't be directly compared to each other, but the key features can be compared. Both computer vision and the 3D laser scanning have a very high scanning rate, even though they vary in the way that the scan is done. However as highlighted in the literature study, the 3D laser scanner can scan and object or surface while moving, whereas using area scan cameras for computer vision, the image must be captured while the platform is stationary to prevent blurring. This gives 3D laser scanning an advantage over the computer vision as it can scan the powder bed while the re-coater is moving. However, if using line scan cameras for computer vision, it can be used in a similar fashion compared to the 3D laser scanner. The processing speed is also directly proportional to the amount of processing power that would be required to process the image or scanned 3D point cloud. It is also proportional to the type of processing that needs to be done on the captured image or 3D data. Both systems have the capability to perform on-device processing of the data, thus the computer is normally only required to view the image or 3D point cloud. However, a computer can be used to perform further analysis and processing of the captured data. The disadvantage of the 3D laser scanner in this regard

is that the point cloud must first be converted into readable formats, which requires a significant amount of computer resources and processing time.

The final factor to be considered is the costs associated with each of these technologies. The cost of the device and its required peripherals is a very important factor, as the chosen technology must be as cost-effective as possible. This factor does not affect the operation of the system but has a significant impact on the affordability of the system. If the system is not cost-effective, the return on investment will be very low and it would not be a viable option to implement the system in a production environment. The cost of any system consists of not only of the device itself but also must include all the peripherals that are required for the system to operate normally. Computer vision would require a camera, processing unit and lighting if needed. A laser scanner requires fewer peripherals than computer vision, but the price of the actual laser scanning unit is significantly more compared to computer vision as can be seen in Table 3.

Considering all the aspects mentioned above, the computer vision is most suited to develop the active monitoring system. The next step in the process is to determine all the component specifications that will be used with the computer vision technology.

## 3.2 Lighting for surface defects

The lighting is the most important aspect of computer vision, as without some form of illumination most computer vision technologies would be completely useless.

There are a few important things that should be considered when selecting the correct lighting. One of the first aspects that must be looked at is the application that the lighting will be used for. For this research, the purpose of the lighting will be the highlighting of surface defects on the PMMA surface. It should be higlighted that the PMMA powder has highly reflective characteristics due to the spherically shaped powder particles. Thus, the illumination over the platform should be uniform. Excessive blooming or lighting hot spots can cause important image features to be lost. The reverse is also true as a poorly illuminated image can also mask defects or in some cases make them completely undetectable. When the lighting is not uniform, it

causes problems with the signal to noise ratio and makes operations such as thresholding of the image very difficult (Edmund Optics 2017). This enforces that the correct types of illumination should be selected.

When selecting an illumination method for a particular application, the type and surface finish of the object must first be determined. For this application, a flat smooth surface, white in colour and made from highly reflective but grainy material, will be inspected. This indicates that a lighting method that can provide uniform illumination of the entire powder bed must be selected. However, the lighting must be able to highlight any surface defects, or at the least provide a detectable contrast difference between the defect that has occurred and the rest of the powder bed.

In an article written by Edmund Optics, a comparison of the different illumination methods are highlighted and is demonstrated in Table 4. From the data displayed in this table, it can be seen that the type of lighting required to highlight surface defects on a flat or nearly flat surface is a single source of directional light or a structured light projector.

**Table 4 Illumination methods comparison(Edmund Optics 2017)**

| Application requirement | Object under inspection | Suggested type of illumination |
|---|---|---|
| Reduction of specularity | Shiny object | Diffuse front, diffuse axial, polarizing |
| Even illumination of object | Any type of object | Diffuse front, diffuse axial, ring light |
| Highlight surface defects or topology | Nearly flat (2-D) object | Single directional, structured light |
| Highlight texture of object with shadows | Any type of object | Directional, structured light |
| Reduce shadows | Object with protrusions, 3-D object | Diffuse front, diffuse axial, ring light |
| Highlight defects within object | Transparent object | Darkfield |
| Silhouetting object | Any type of object | Backlighting |
| 3-D shape profiling of object | Object with protrusions, 3-D object | Structured light |

As suggested by the data in Table 4, the recommended type of light for this study is a form of directional or structured lighting. Directional lighting is lights that illuminate the surface only from one side or direction as displayed in Figure 25. When shining a directional light against an object, as shown in Figure 25, any light that cannot pass through the object will cause a shadow to be cast at the opposite side of the object. The length and size of the shadow is also dependent on the size of the object, as well as the angle that the light source is positioned at. Taking this into consideration, any defects that may occur on the bed will be contrasted from the rest of the powder bed by creating a shadow or highlight over the defective area. This makes it easier to identify the presence of the defect as the shadow will be considerably more visible than the defect itself. Directional lighting is not limited to spotlights only, as bar lights or fibre optic type lights can also be used for directional lighting.

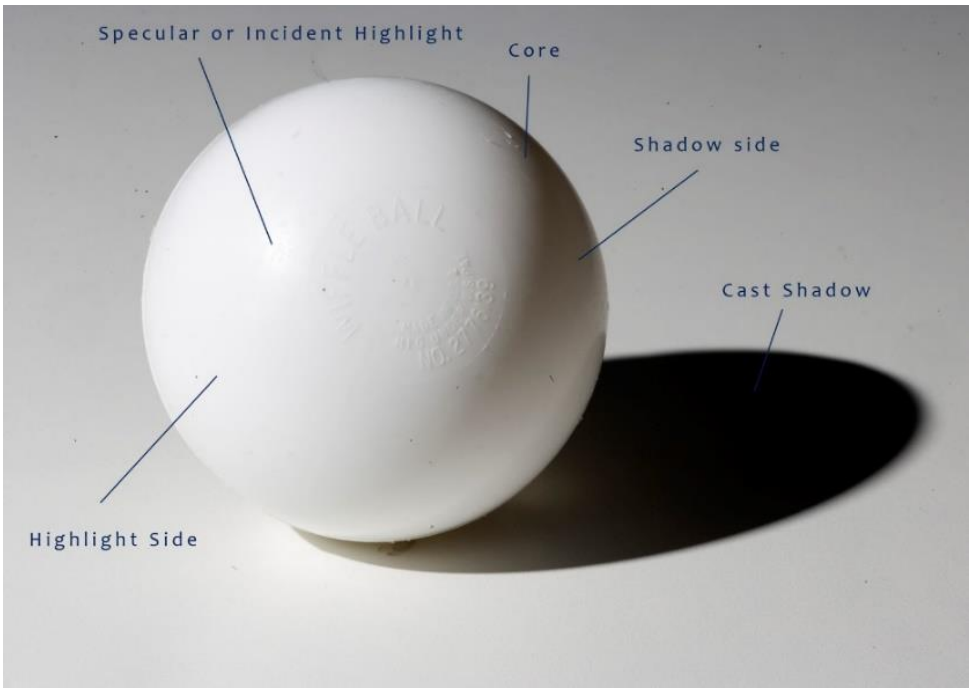



**Figure 25 Effects of directional lights on shadows(Callis 2010)**

For this specific application, because a flat surface is being examined, the angle that the light are mounted at is critical. The reason for this is that when a defect occurs, the angle of incidence must be low enough so as to not illuminate the inside of the defect itself. This would result in the fact that the camera might not detect the change on the powder bed.

In order to illuminate the powder bed uniformly, a bar light is required that is the same length as the powder bed. Since modification of the machine is extremely limited, the factory installed lighting luminaire will be used to illuminate the powder bed. The luminaire is the same length as the powder bed and is bright enough to illuminate the entire powder bed surface. It is essential that the light does not create hot spots on the surface of the powder bed, as this can affect later processing of the image.

The powder bed will be illuminated using the fluorescent light installed inside the hood lid as displayed in Figure 26.

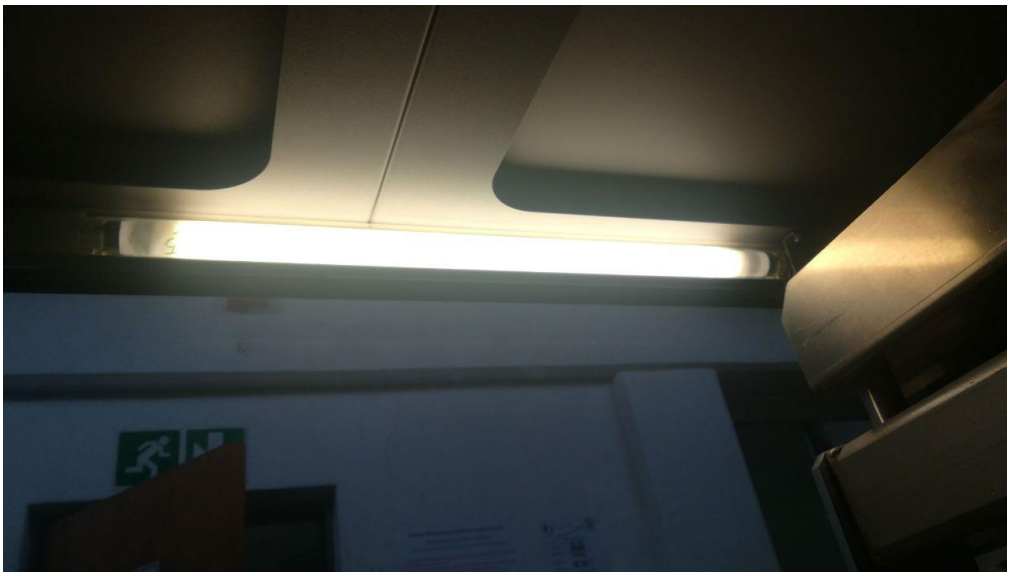


**Figure 26 Internal AM machine light**

The fluorescent light was a 20-watt fluorescent tube mounted at a 50-degree angle with reference to the surface of the powder bed.

It is necessary to determine the minimum shadow size that will be cast by a defect after a single layer. The size of this shadow is important as the camera specifications will be based upon this shadow size. The size of the shadow can be determined mathematically using basic trigonometry.

The diagram in Figure 27 illustrates visually how the size of the shadow will be calculated when a defect occurs after a single re-coating operation. This illustration assumes that the defect had occurred on the edge of the powder bed, as at this position the shadow that will be cast by the defect will be at its smallest.

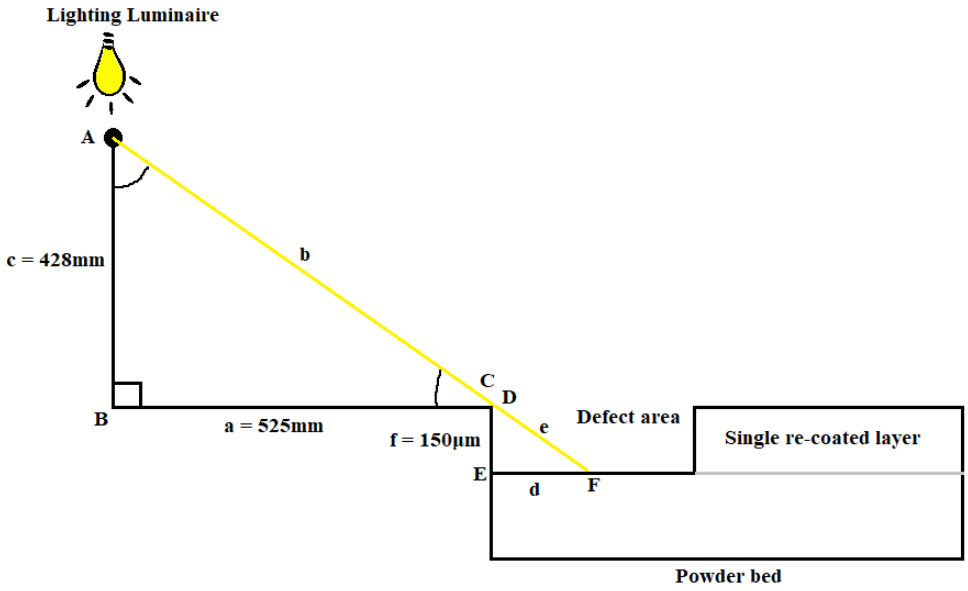



**Figure 27 Powder bed illumination**

The first step that is required is to determine the angle A at which the lighting luminaire is mounted in relation to the powder bed. Since the the height of the light above the powder bed as well as the distance of the luminaire from the side of the powder bed is known, the angle at which the powder bed is being illuminated can be calculated using trigonometry. Since the opposite and adjacent sides of the triangle is known, angle C can be calculated using the tangent function as shown in Eq. 1.

$$\tan\theta = \frac{c}{a} \qquad \textbf{Eq. 1}$$

Where: c ≡ Height of light above the powder bed (mm)

a ≡ Length of the light from the powder bed (mm)

Manipulation:

$$\theta = \tan^{-1}\frac{c}{a}$$

$$= \tan^{-1}\left(\frac{428}{525}\right)$$

$$\theta = 39.188°$$

Now that the angle at which the powder bed is being illuminated at is identified, the angle A at which the luminaire is mounted in relation to the powder bed, can be

calculated. Since the sum of all interior angles in a triangle is 180 degrees, the unknown angle $\theta_C$ can be calculated using the fomula shown in Eq. 2.

$$\theta_A + \theta_B + \theta_C = 180 \quad \textbf{Eq. 2}$$

Where: $\theta_A$ ≡ Angle of mounted luminaire (°)

$\theta_B$ ≡ Angle of light to powder bed (°)

$\theta_C$ ≡ Angle of illumination (°)

Manipulation:

$$\theta_A = 180 - \theta_B - \theta_C$$
$$= 180 - 90 - 39.188$$
$$\theta_A = 50.812°$$

Now that all the interior angles of triangle ΔABC is known, the length of the shadow that will be cast in triangle ΔDEF can be calculated. Looking at Figure 27, it can be seen that triangle ΔABC is similar to triangle ΔDEF. This means that since all the interior angles and the length of side f is known, the missing side d can be calculated using the Sine Rule as shown in Eq. 3.

$$\frac{d}{\sin\theta_D} = \frac{e}{\sin\theta_E} = \frac{f}{\sin\theta_F} \quad \textbf{Eq. 3}$$

Where: d ≡ Length of the shadow (μm)

$\theta_D$ ≡ Angle of Illumination (°)

f ≡ Depth of defect (μm)

$\theta_F$ ≡ Incoming light angle (°)

Manipulation:

$$d = \left(\frac{f}{\sin\theta_F}\right) \times \sin\theta_D$$
$$= \left(\frac{150}{\sin(39.188)}\right) \times \sin(50.812)$$
$$d = 184\mu m$$

Using the information calculated using Eq. 3, the minimum shadow size that must be detectable by the camera is measured at 184µm. This will be used as the smallest feature that the camera must be able to capture when calculating the specifications of the camera module. It is important to note that due to the width of the powder bed, the size of the shadow will increase on the further ends of the powder bed. This means that shadows wil be easier to detect on the further side of the powder bed than on the closest side.

A baseline reference image of the powder bed after a successful re-coating operation was captured to test the effectiveness of the light. The reference image as demonstrated in Figure 28 was captured using only the fluorescent light. When closely examining the image, it appears that the single light source sufficiently illuminates the entire powder bed. However, the powder bed is not as uniformly illuminated as would be expected. The illumination appears to diminish in intensity at the far side of the image, the reason for this is due to the distance between the lighting fixture to the far side of the powder bed.


**Figure 28 Illumination of the powder bed in the Voxeljet VX500**

However, the illumination at the far side of the powder bed did not drop to the point where the image was completely underexposed in some areas and thus the illumination was considered sufficient. Structured lighting was also one of the illumination methods that was suggested, however, this method of lighting is more suited to generating 3D models from the captured images as discussed in Chapter 2. This

method of scanning was determined to be unsuitable for the purposes of this research and will thus not be investigated.

## 3.3 Camera specifications

There are several factors that must be considered when selecting a camera for a computer vision application. Some of these factors include: the number of pixels, the focal length of the lens, field of view, minimum focusing distance, working distance, low light performance and sensor size. Not all these factors are as critical to the operation of the system, but they will influence the effectiveness of the system. The three most important parameters for this application are the sensor size, focal length combined with the field of view and the working distance of the camera. Each of these parameters will be discussed and calculated below.

#### 3.3.1.1 Required sensor size

The first variable that will be calculated is the minimum number of pixels required by the camera to capture very small features on the powder bed. This can be calculated mathematically using the formula in Eq. 4 (National Instruments 2014). The minimum number of pixels needed to cover the smallest feature can be any arbitrary whole number. However, it should be considered that if a too high pixel coverage value is chosen, it will drive up the total number of pixels that will be required by the camera. A higher pixel coverage value will increase the camera's capability to capture smaller details, but also increase the cost and required processing power.

$$Sr = Pc\left(\frac{FOV}{Sf}\right) \qquad \textbf{Eq. 4}$$

Where: Sr ≡ Sensor Resolution (pixels)
Pc ≡ Pixel Coverage (pixels)
FOV ≡ Field of View (mm)
Sf ≡ Smallest Feature (mm)

In the previous section, it was determined that the Voxeljet VX500 technology makes use of a layer thickness of 150µm. However, there is not much data available as to what the factors are that causes the recoating errors, thus the smallest defect width that may occur is very difficult to determine.

Therefore, the smallest feature that the camera must be able to capture was calculated using shadows cast by the defect as reference. This smallest feature value was calculated as 184µm. A pixel coverage value of one pixel was selected as this was deemed sufficient to cover the smallest assumed defect. From the previous section, it was also determined that the dimensions of the VX500's powder bed is 500mm wide and 400mm long. Thus, the FOV of the camera must be the same dimensions as the powder bed.

The number of horizontal and vertical pixels necessary for the camera to capture an image of the powder bed can be calculated using Eq. 4 and is demonstrated below:

$$\begin{aligned} \mathrm{Sr}_{\mathrm{horizontal}} &= \mathrm{Pc}\left(\frac{\mathrm{FOV}}{\mathrm{Sf}}\right) \\ &= 1\left(\frac{500}{0.184}\right) \\ &= 2717 \text{ pixels} \end{aligned} \qquad \textbf{Eq. 5}$$

$$\begin{aligned} Sr_{vertical} &= Pc\left(\frac{FOV}{Sf}\right) \\ &= 1\left(\frac{400}{0.184}\right) \\ &= 2174\ pixels \end{aligned} \qquad \textbf{Eq. 6}$$

From the calculations, it is evident that a minimum resolution of 2717 x 2174 pixel is needed. The calculated resolution needs to be compared against current commercial sensors to select the most compatible sensor. Table 5 shows a list of the standardised camera resolution sizes, in megapixels, as well as the pixel dimensions for each commercially available sensor. Using the calculated values from Eq. 5 and Eq. 6, the required camera sensor resolution was calculated as 5.91-Megapixels. Taking the list

of commercially available sensors into consideration, a 7-megapixel sensor was the closest specification to the calculated requirement. It is necessary to select a larger sensor because the number of vertical pixels that will be required is more than a 6-megapixel camera sensor can deliver.

**Table 5 Standard camera sensor sizes(Cohoon 2011)**

| Megapixels | Sensor resolution |
|---|---|
| 2 | 1600x1200 |
| 3 | 2048x1536 |
| 5 | 2592x1944 |
| 6 | 3032x2008 |
| 7 | 3072x2304 |
| 8 | 3264x2448 |
| 10 | 3648 x 2736 |

Although there are several other factors that can influence the extent of details that the camera can capture, the number of pixels when capturing the image is crucial to ensure that all the required details is captured. If the number of pixels is too small, a lot of minor detail can go missing (Edmund Optics 2011). The higher the number of pixels, combined with the correct lens choice, the smaller the object or amount of detail that can be captured in the image. However, a higher pixel count would result in a larger file size and will require additional computing power. This means that more storage is required to store the image, and more processing power is required to process and analyse the image, as discussed in the previous chapter. Therefore, a compromise must be reached as to what is useful and what would be practical.

When looking at the calculated camera sensor requirements, it has been determined that a camera with a sensor size of between 6 and 7 megapixels is required to enable imaging of the entire powder bed. The Raspberry Pi camera module met the required specifications and was selected. Although the Raspberry Pi camera has an 8-megapixel sensor, it was determined to be well suited to the application as it has a low profile, low cost and meets and exceeds the minimum megapixel requirements.

#### 3.3.1.2 Required Field of View

The next factor that should be considered is the focal length. The working distance of a camera is defined as the distance between the front of the camera lens to the surface of the object being examined (National Instruments 2014). The working distance influences the focal length of the lenses that can be used for a specific application. The formula to calculate the focal length is demonstrated in Eq. 7 and makes use of the vertical length of the camera sensor, as well as the length of the object being captured. (Czeranowsky 2016).

$$\mathrm{Fl} = \frac{\mathrm{Vess}}{\mathrm{Lpb}} \times \mathrm{Wd} \qquad \textbf{Eq. 7}$$

Where: Fl ≡ Focal Length (mm)
Vess ≡ Vertical sensor size (mm)
Lpb ≡ Length of Powder bed (mm)
Wd ≡ Working Distance (mm)

In this case, a fixed working distance must be selected as the system is ultimately being designed to be integrated into existing powder bed systems. Because there are a limited number of places where the camera can be located inside the machine, a suitable location must be chosen, and the focal length of the lens and the lighting must be adjusted to fit in with the mounting location. If the system is to be used in a different type or design of machine, these values need to be recalculated. This also gives the system a sense of modularity, as the various parts of the system can potentially be modified to suit different machines.

It was determined that the working distance for the camera will be 440mm. The vertical measurement of the camera sensor is required for the calculation of the lens focal length. The Raspberry Pi camera as displayed in Figure 29, has a sensor with a diagonal measurement of 4.6mm and a vertical measurement of 2.76mm (Raspberry Pi Foundation 2016a).

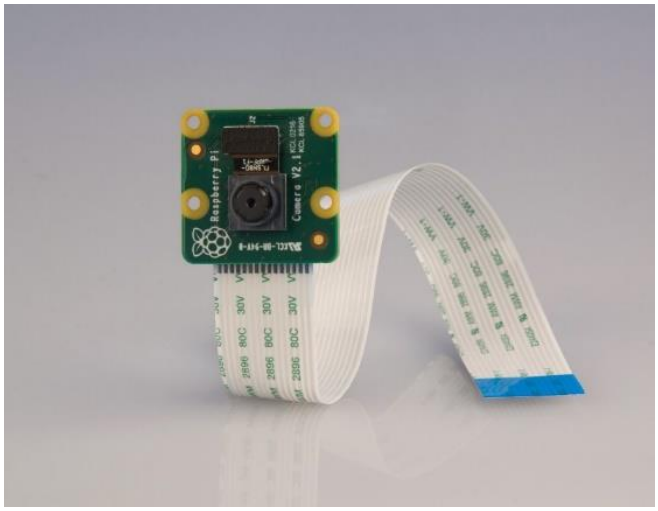


**Figure 29 Raspberry pi camera module(Raspberry Pi Foundation 2016a)**

Taking this into consideration, the required focal length can be determined as follows using Eq. 8:

$$\begin{aligned} \mathrm{Fl} &= \frac{\mathrm{Vess}}{\mathrm{Lpb}} \times \mathrm{Wd} \\ &= \frac{2.76}{400} \times 440 \\ &= 3.036 \text{ mm} \end{aligned} \qquad \textbf{Eq. 8}$$

The calculated focal length is not a standard lens size; thus, the closest available lens size should be selected. The Raspberry Pi camera module has a fixed 3.04mm lens attached. Thus, the stock factory lens can be used as it conforms to the calculated lens focal length requirements. The Raspberry Pi camera also provides an adjustable lens focus.

#### 3.3.1.3 Raspberry Pi camera interface

The Raspberry Pi camera module makes use of a Camera Serial Interface (CSI) for connectivity to the processing hardware. The Camera Serial Interface is supported by a number of ARM microprocessor devices, one of the more notable chips being the Broadcom BCM2837 that is based on the ARM Cortex A53 MPCORE processor (Raspberry Pi Foundation 2016b). This method of connectivity is very widely used in

the mobile device and automotive industry. Some of its fundamental features are that it delivers a high data transmission performance, uses low power and has a low EMI (Mipi Alliance 2017). Its most common use is for imaging and vision applications as the name suggests.

Due to recent advances in microcomputer technology, there are several compact computing devices that can be used to drive this specific camera module. One of these systems being the Raspberry Pi micro-computer as demonstrated in Figure 30.


**Figure 30 Raspberry pi 3 model B micro-computer(Raspberry Pi Foundation 2016b)**

The Raspberry Pi is a micro-computer that is also built around the ARM Cortex A53 microprocessor chip. It has a variety of connectivity options including I2C, SPI, USB, HDMI, CSI as well as hardwire Ethernet and wireless (WIFI/Bluetooth) network technologies. The added benefit of using this platform is that all the image processing tasks can be done on the Raspberry Pi micro-computer, which makes this an extremely cost effective and compact option. Since the hardware is very small in size, it can easily be installed inside the AM machine. The entire system can also be managed remotely as the micro-computer can be connected to an Ethernet network, which gives it the capability to monitor the condition of the machine without having to be physically present during machine operation.

### 3.3.2 System design

It was determined that a single camera could be used to capture images of the entire powder bed. The camera would have to be positioned so that a single captured image will cover the entire surface of the powder bed. The basic design concept of the active monitoring system is demonstrated in Figure 31 and Figure 32 where binder spraying head have been omitted for clarity.

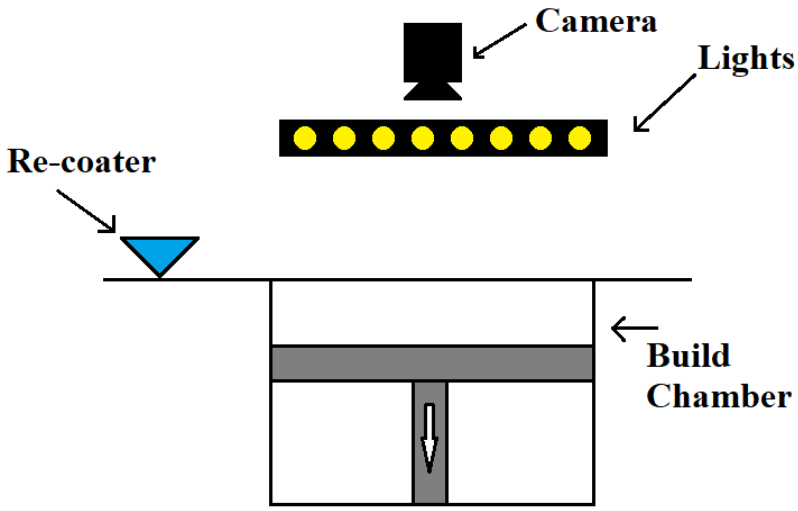



**Figure 31 Basic monitoring system design side view**

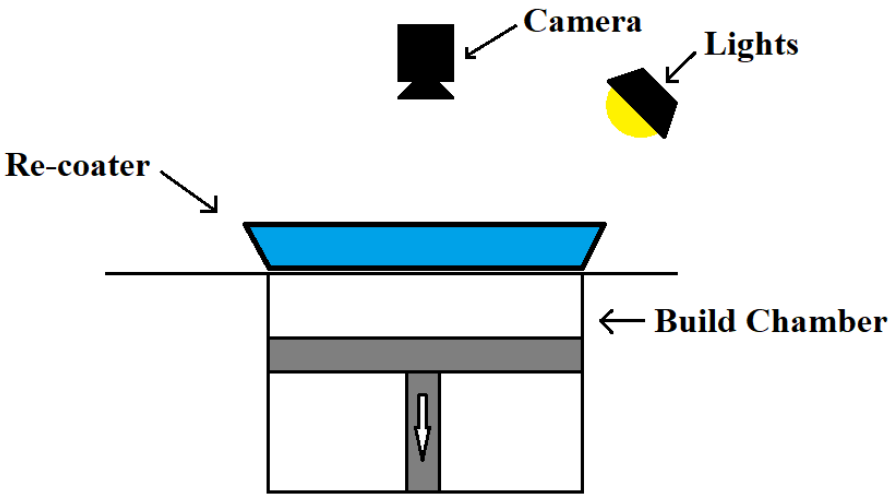



**Figure 32 Basic monitoring system design front view**

Since directional lighting will be used, the position of the lighting relative to the camera is critical to the success of the imaging system. Considering the machine

components in Figure 31, the re-coater only applies a new layer of powder in a single direction along the length of the powder bed. This means that the lighting would have to be placed perpendicular to the direction of recoating as shown in Figure 32, since the defects would occur in the direction of re-coating. However, the lights would only be placed on the one side of the powder bed to create the maximum contrasting effect on the defects. The position of the lights above the powder bed is also vital. The lights cannot be positioned too high above the powder bed, as this may illuminate the defects from the wrong angle, causing the camera not to detect them. This means that the camera would not be able to detect the difference in contrast between the powder bed and the defect that occurred.

When looking at the side view of the basic design, the positioning of the re-coater monitoring system components can clearly be seen in relation to the basic operating components of the machine. Although the binder spraying head is omitted from the basic design in Figure 31, the camera must be mounted in such a way that the additional components does not interfere with the normal operation of the machine. However, the camera will still be mounted on the calculated working distance so that it would not be necessary to re-calculate the camera lens focal length.

Since the machine has a re-coater as well as the adhesive spraying head that moves over the powder bed during the normal course of operation, there are only specific intervals between these operations at which images of the powder bed must be captured. Thus, a triggering mechanism is required when the re-coater has applied a new layer of powder to the powder bed. The easiest method to trigger the camera system would be to install a limit switch, at the home position. When the re-coater reaches the home position, it will close the contact and trigger the camera to capture the required image and process it before the next re-coating operation.

The hardware will be able to take clear images of the powder bed, because for that split second the entire machine would be relatively motionless, ensuring that blur-free images can be captured. Ensuring that the images are clear is critical to the successful operation of the system, as finer details may be lost if the images are distorted or if unnecessary components of the machine are captured in the image.

The Raspberry Pi micro-computer primarily makes use of a GNU-Linux based operating system that has been compiled specifically for the ARM SoC hardware. The GNU-Linux distribution of choice for the Raspberry Pi platform is the Raspbian OS. Raspbian OS inherently supports all the features of the Raspberry Pi hardware (Raspberry Pi Foundation 2017). The latest version of the distro, at the time that this research was conducted, was Raspbian Jessie with the PIXEL GUI environment. Included in this distro is several software packages, including the Python development environment and runtime, Java, Wolfram Mathematica and various other software development and scientific research packages. Additional software can also be installed onto the OS via the standard installation procedures, but custom written or specialized software packages and libraries can be compiled and installed on the OS for more specialized applications. A software package that will be applicable to this research is OpenCV library. The OpenCV library is an open source computer vision and machine learning library that can be used in conjunction with several software programming packages and languages to develop computer vision applications, with emphasis on real-time applications. The library consists in excess of 2500 optimised algorithms and consists of classic and modern state-of-art algorithms (OpenCV 2017). This makes the library of great use to this research application due to its large base of algorithms, as well as the ability to run on the Raspberry Pi micro-computer.

#### 3.3.2.1 Open CV

The OpenCV library natively supports the Python programming language. However, there are additional add-on packages that are required by the OpenCV library when used in Python. Some of these add-ons include a mathematical function plotting library called Matplotlib, Numpy array tools and the Picamera library so that the Raspberry Pi camera can be controlled by the Python application.

It is commonly known in that images are stored as 3-dimensional arrays, which stores the pixel values of each primary colour, namely red, green and blue (RGB). However, when the OpenCV library was created in the year 2000, the accepted colour format for storing images was blue, green and red (BGR) (Mallick 2015). Even though this colour format has changed over recent years, a very large portion of the library was built upon

this foundation and has never been changed. Therefore, images need to be converted to the correct colour format, before it is imported into the OpenCV library.

There are a number of image processing techniques that will be used in conjunction with one another to detect surface defects.

- **Colour to greyscale conversion**

The first technique that will be discussed is the conversion of a BGR colour image to a single channel grey scale image as demonstrated in Figure 33. There are many image processing techniques that can only process a single colour channel at a time. Some of these techniques are algorithms such as thresholding and edge detection. Therefore, it is often necessary to convert colour images to a greyscale format.

**Figure 33 Converting colour image to greyscale(Abbas 2013)**

The cvColor method will be used to convert a colour image from the BGR format to greyscale. The cv::cvtColor method requires four parameters, and does not return any value as the output array is part of the method's parameters.

The constructor function for the cvtColor method is displayed as follows:

void cv::cvtColor(src, dst, code, dstCn=0)

Where:

Input array ≡ Source image (src)

Output array ≡ Destination image (dst)

Code ≡ Colour space conversion code (code)

dstCn ≡ number of channels in destination image (dstCn)

The four parameters that can be passed into the method consists of the input array which stores the original image to be converted, the second parameter is the output array where the converted image will be stored. The third parameter is the code for the type conversion that must be performed. Although OpenCV has several conversion codes that can be used, the one that is important for the purposes of this research is COLOR_BGR2GRAY. Although the code is an integer value, the name of the conversion that is needed can be passed to the method and the correct conversion code will be determined from the lookup table built into the method. The fourth parameter is the number of channels in the destination image, however, when this parameter is set to zero, the number of channels will automatically be determined from the source and code value (Bradski 2017).

- **Histogram equalization**

The second technique that will be discussed is the equalization of the image's histogram. A histogram is used to indicate the distribution of the pixel intensities. To properly explain this, the histogram and image in Figure 34 will be used as an example.

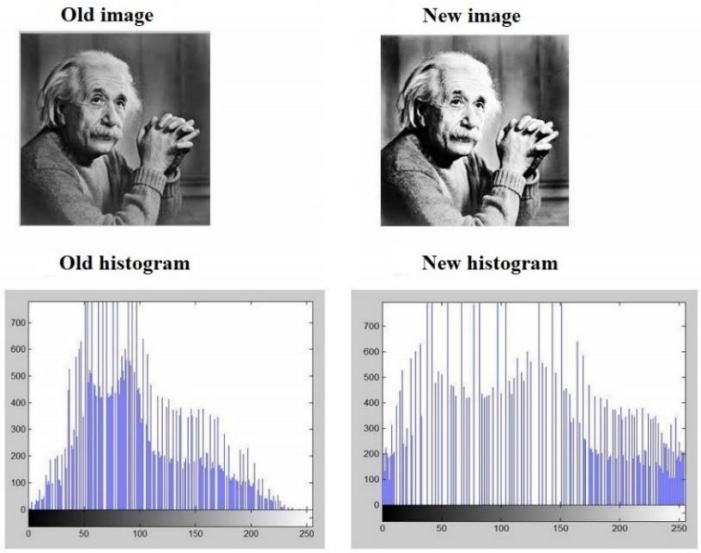



**Figure 34 Histogram equalization (Tutorials Point 2014)**

When capturing an image, the lighting conditions under which the image is captured may not always be ideal and may result in an image that is over or underexposed resulting in a loss of contrast. Histogram equalization is often used as an attempt to

increase the contrast in an image. As can be seen from the original image's histogram that there are many pixels clustered around the middle range of pixel intensities. This is not an ideal situation as it is more desirable to equally spread the pixel intensities over the entire spectrum. The histogram equalization stretches out the clustered range of intensities and try to spread the intensities more equally across the entire spectrum. Once the equalization has been applied to the original image, the histogram clearly shows a more spread out pattern of pixel intensities as demonstrated in Figure 34. On the new image it can be seen that the contrast has been increased when compared to the original image. Although the image may seem slightly over exposed after histogram equalization, the edges on the image are much more defined and clearer. However, in some cases, the contrast may not be increased, but the equalization process may decrease the contrast. Therefore, the effects of the equalization must be examined visually after being applied, to ensure that the equalization had the desired effect on the image. Although it may look like the image is over exposed, the equalization process makes it easier for certain types of image processing algorithms to process the image, as the higher contrast on the image highlights a lot more edges on the images than what would normally be visible. The effect of histogram equalization is a lot clearer on colour images than greyscale since greyscale images tend to look over exposed when equalized.

To equalize the histograms of the required images, the OpenCV method of cv::equalizeHist will be used. This method uses two parameters, namely the source image and the destination image. The constructor function for the cv::equalizeHist method is as follows:

void cv::equalizeHist (src, dst)

Where:

Input array ≡ Source image (src)

Output array ≡ Destination image (dst)

The first parameter of the method is the source image that needs to be processed. This input array can only be an 8-bit single channel image, which means the image must

already have been converted to greyscale before processing (Bradski 2017). The second parameter is the output array where the equalized image will be stored. This output image will also be an 8-bit single channel image, the same type as the source image.

- **Absolute difference of images**

The third technique that will be used in the processing of the images is the calculation of the absolute difference between two images. When trying to identify the differences between two images as demonstrated in Figure 35, subtracting the two images from each other can highlight the differences between the two.

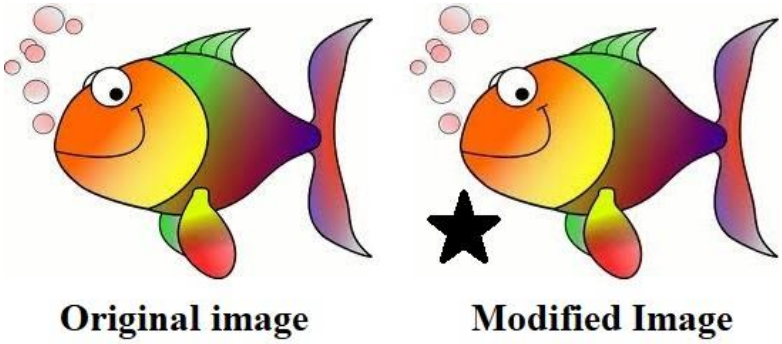



**Figure 35 Original and modified images to be subtracted**

However, for this method to be successful, both images must be similar, so that only the differences between the two images are highlighted. Otherwise the results will not have the desired effect. It is also important that the two images must have the same size, i.e. the same amount of vertical and horizontal pixels. The image as displayed in Figure 36 shows the difference between the original and modified image after the two images have been subtracted from each other.

**Figure 36 Difference between the original and modified image**

It is worth noting that since the pixels in an image can only be a positive number, the absolute difference between the two pixels would have to be calculated, otherwise the pixel will be given the value zero as it is not possible for a pixel to have a negative value because any value below zero is considered out of range.

The method that will be used to calculate the absolute difference between the images using the OpenCV library is the cv::absdiff method. This method requires three parameters, namely the original image, the second image that will be subtracted from the original image, and the output array where the resultant image will be stored. The constructor function for the method cv::absdiff includes:

void cv::absdiff (src1, src2, dst)

Where:

Input array ≡ Source image 1 (src1)

Input array ≡ Source image 2 (src2)

Output array ≡ Destination image (dst)

It is necessary to create an empty array similar to the two images being subtracted from each other. This empty array is essential as the resultant image must be stored in a separate array for further processing.

- **Gaussian blur**

The fourth technique that will be used to process the image is the application of Gaussian blur to the image. The purpose of applying blur to an image is an attempt to reduce the amount of noise present in the image, but to still retain the key features.

Excessive noise in an image can have a detrimental effect on any further processing attempts. If any further processing such as thresholding or edge detection operations is performed on an image with large amounts of noise present, the algorithms would perform very poorly due to the amount of noise present.

Therefore, before any subsequent processing is performed on the image, the noise present on the image would have to be reduced or eliminated if possible.

The image as displayed in Figure 37 contains excessive amounts of high frequency noise and would require smoothing before further processing can be applied to the image.

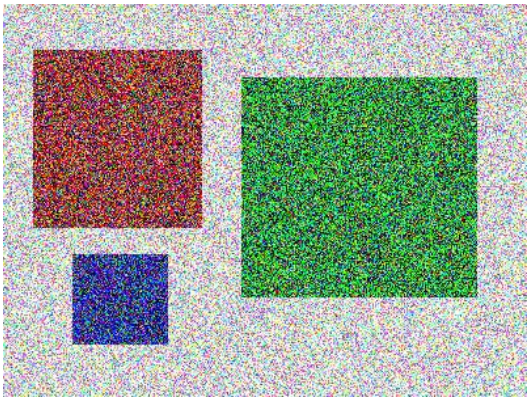


**Figure 37 Noisy source image**

The Gaussian blur filter is a 2-dimensional convolution operator that's primary purpose is to blur images to remove small detail and noise from an image. The degree of smoothing that is applied to the image by the filter is determined by the standard deviation of the Gaussian window. The Gaussian window uses a pre-set block size of pixels that determines how the filter is applied to the image. The Gaussian window outputs a weighted average of each pixel neighbouring the middle pixel, with the middle pixel having the highest weight value, and the surrounding pixels having a lesser weighted value (Fisher 2003). A Gaussian filter smooths and preserves more detail than a similar sized mean filter. The result of the image in Figure 37 after a Gaussian blur filter was applied is shown in Figure 38.

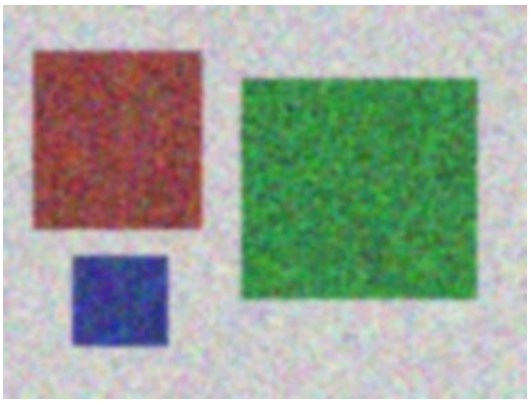


**Figure 38 Gaussian blur applied to noisy image**

When examining the processed image in Figure 38, it can clearly be seen that the amount of noise present in the image has significantly been reduced without a great loss of detail to the features depicted in the image. Once the filter has been applied to the image, the image can then be further processed using thresholding or edge detection algorithms with a much higher success rate.

The method that will be used to apply the Gaussian Blur to the image inside the OpenCV library is the cv::GaussianBlur method. This method requires six parameters, namely the input and output images, the Gaussian Kernel size, the Gaussian sigma values for the X and Y direction, and the last parameter is the border type. The constructor function for the cv::GaussianBlur method includes:

void cv::GaussianBlur (src, dst, size, sigmaX, sigmaY, borderType)

Where:

| | |
|---|---|
| Input array | ≡ Source image (src) |
| Output array | ≡ Destination image (dst) |
| Kernel size | ≡ Gaussian window kernel size (ksize) |
| Sigma X | ≡ Gaussian kernel standard deviation in X direction (sigmaX) |
| Sigma Y | ≡ Gaussian kernel standard deviation in Y direction (sigmaY) |
| Border type | ≡ Pixel extrapolation method (borderType) |

The Gaussian Blur filter can be used on images that have any number of channels, but each channel will be processed individually. This means that an image does not need to be a greyscale image before the filter can be applied. The Gaussian kernel size can be specified, but the value specified must be positive and odd. If the kernel size is not specified, a value of zero can be entered and the kernel size will be calculated from the sigma values.

The values for sigma X and Y can also be specified, but if the value of sigma Y is zero, the value for Y will be set to the same value as Sigma X. However, if both Sigma's are zero, the Sigma values will be calculated from the kernel width and height value. For the best control of the results and to prevent incompatibilities with future developments, it is recommended to specify both the kernel and Sigma values.

- **Canny edge detection**

The fifth and last technique that will be discussed is probably the most import process of the monitoring system. This process is called the canny edge detection algorithm. Canny edge detection is a very popular edge detection algorithm that was developed in 1986 by John F. Canny (Rosebrock 2017). It is a multi-stage algorithm that can detect edges with noise that is suppressed at the same time. The algorithm can be broken down into 5 steps. Each of the steps will be discussed briefly to explain its operation.

1. The first step is to smooth the image using a Gaussian blur filter as discussed previously to reduce the amount of high-frequency noise.
2. The second step is to calculate the intensity gradient of the image using a gradient operator such as the Sobel kernel.
3. The third step is then to apply a non-maximum suppression to remove any false responses to the edge detection.
4. The fourth step is to apply thresholding to the image using an upper and lower value on the gradient values.
5. The fifth and final step in the algorithm is to track the detected edges using hysteresis and suppressing weak edges on the assumption that edges are long lines. The purpose of the fourth and fifth step together is to distinguish what are indeed edges, and what is not.

The image on the left in Figure 39 displays an image before and after the Canny edge detection algorithm has been applied.


**Figure 39 Original image and edge detected image**

As can be seen from the image in Figure 39, the edge detection successfully detects most of the strong edges present. There are some details that may still be missing, however by adjusting the various parameters in the algorithm, an acceptable result can be achieved. The resolution and lighting quality also play a significant role in the amount of detail that can be identified using this algorithm.

The method that will be used for canny edge detection inside the OpenCV library is the cv::canny method. Although there are two methods for this single function, only the standard method will be used as the overloaded method is not applicable to this specific application.

The standard cv::canny method takes six parameters, however only four of these parameters are critical, as the last two parameters can be calculated automatically if required. The first four parameters that must be supplied for the method to perform its function is the input array for the source image, the output array where the processed image will be stored, and the upper and lower threshold values for the hysteresis procedure. The two optional parameters are the aperture size for the Sobel operator, and then the flag for the L2 gradient to use the normal or the more accurate function to calculate the image gradient magnitude (Bradski 2017). The constructor function for the cv::canny method includes:

void cv::Canny(Input Array image, Output Array edges, double threshold1, double threshold2, int aperturesize = 3, bool L2gradient = false)

Where:

| | |
|---|---|
| Input array | ≡ Source mage |
| Output array | ≡ Destination image |
| Threshold1 | ≡ High threshold value |
| Threshold2 | ≡ Low threshold value |
| Aperture size | ≡ Aperture size for Sobel operator |
| L2 gradient | ≡ L2 gradient flag |

When using Canny Edge detection, there is often a problem that is encountered, namely how to determine the two threshold values. In a study done by Adrian

Rosebrock, a method was proposed to determine the optimal threshold values automatically using basic statistical methods (Rosebrock 2015). First, the median of all the pixel intensities in the images is calculated. Next, a value of sigma is determined from which the upper and lower threshold values will be calculated. The smaller the value of sigma that is selected, the tighter the threshold values will be, and the larger the value of sigma, the wider the threshold values will be. The value of sigma can be any value between 0 and 1. Once these threshold values have been determined, the canny edge detection algorithm can be applied to the image with relative ease.

#### 3.3.2.2 Graphical user interface

For the software to be more user-friendly, a Graphical User Interface (GUI) had to be included as an interface between the user and the hardware. The GUI was designed to be as simple as possible to ensure that the monitoring system can be operated with minimal difficulty. The GUI, designed for the monitoring system, is shown in Figure 40.

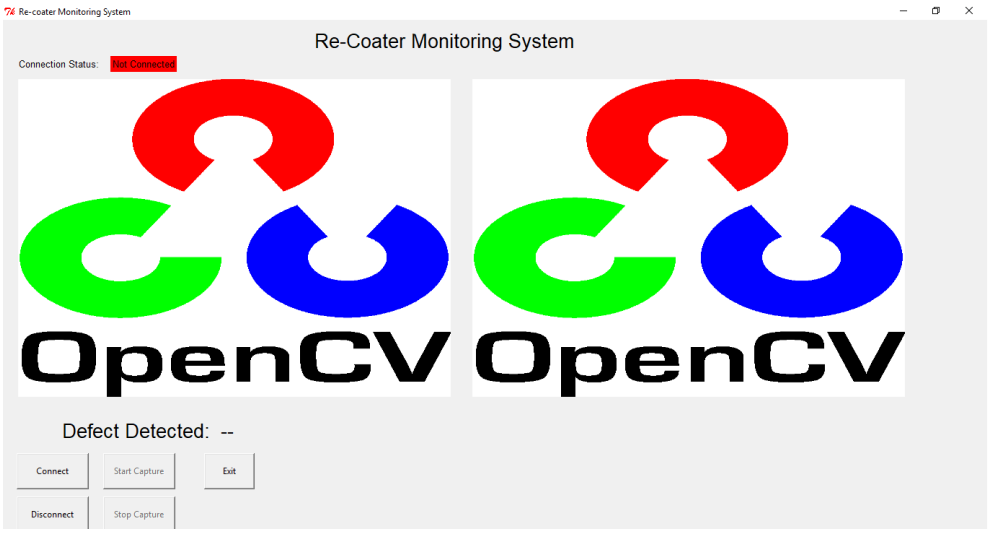



**Figure 40 Re-coater monitoring system GUI layout**

As demonstrated in Figure 40, the program consists of two display windows. The left-hand side window displays the original captured image and the window on the right-hand side displays the image that has been completely processed. These two windows update every time that the re-coater has finished recoating and a new image is captured. There are also four buttons on the interface that are used to initialize the

camera and activate or deactivate the system. The connect and disconnect buttons are used to initialize the camera. The camera needs to be initialized before the system can be activated. Once the connect button has been enabled, the start capture and stop capture buttons are enabled. When the start capture button has been pressed, the program will capture and process the images whenever the limit switch is pressed. Once the build process has been completed, the stop capture button can be pressed to de-activate the system, and this will close the data file that stores all the processing data. The disconnect button will disconnect the camera connection and release all the resources allocated to it. Finally, the exit button can only be clicked once all the capturing has been stopped and all resources controlling the camera module have been released. This will prevent the program from being closed prematurely, preventing allocated resources from being released for other tasks. The GUI was written using the Tkinter GUI library that is included as part of the Python programming language. Although the library has limited functions compared to some of the more advanced libraries available for Python, for the purposes of this research the library had all the necessary basic functions required to develop a basic GUI.

#### 3.3.2.3 De-bouncing methods

When using mechanical switches with electronic devices, there is a phenomenon that occurs called switch contact bouncing. When a mechanical switch contact is opened or closed, the contact bounces between the on and off position for a few microseconds before settling in the desired position. This causes problems when the switch is connected to a microcontroller, as a microcontroller can execute a single instruction in a few nanoseconds. This means that each time that the switch contacts bounces between the on and off position, the microcontroller will detect it as a deliberate switching operation. This bouncing of the switch contacts will cause multiple instances of the interrupt routine to be called in short succession, which is undesirable. Since the proposed active re-coater monitoring system will be using a mechanical limit switch, this unintended switching operation will cause the system to capture more than one image in rapid succession. This can result in images being captured of the powder bed when the re-coater is not in the proper position, causing a false positive to be

recorded as a defect. The oscilloscope readout displayed in Figure 41 displays the switch contact bouncing phenomenon as it occurs.

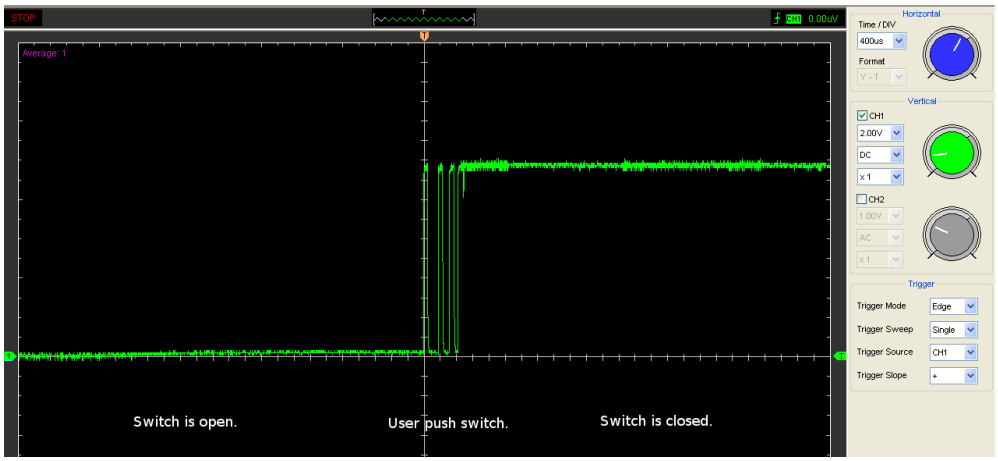



**Figure 41 Switch bouncing output(Christoffersen 2015)**

There is a few hardware and software-based methods that can be used to minimise or remove the bouncing effect. The first and most common method to minimise the effect of the switch bouncing is to connect a capacitor and resistor between the switch's contacts as shown in the circuit diagram in Figure 42.

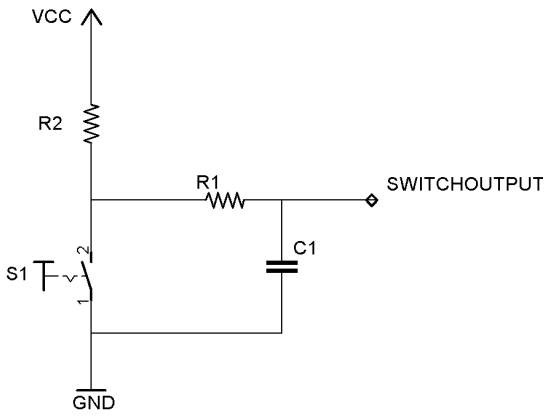



**Figure 42 Hardware switch de-bouncing**

The capacitor builds up a charge, and once the switch contact is closed, the capacitor quickly discharges, which in turn will smooth out the signal being provided to the microcontroller. This circuit can eliminate the effect of the switch contact bouncing. An alternative hardware de-bouncing method is to use a Schmitt trigger. This is a much more expensive method of de-bouncing a switch and is therefore not a very commonly

used method. With microcontrollers becoming increasingly more cost effective to use in electronic circuits, software de-bouncing of switch contacts is becoming increasingly more popular due to less additional hardware requirements. There are a lot of libraries available for different microcontrollers that already include this function, and as a result, can easily be implemented into the main program. A common way that programmers use to eliminate the effect of switch contact bouncing is to include a delay of 50-100 milliseconds to ensure the program ignores any subsequent bounces. However, this is not considered very good programming practice as the delay function is a blocking function, which means the microcontroller won't be able to process any other tasks during this time. The preferred method of handling switch contact bouncing in software is to write a small function that checks when the switch bounces for the first time, and then if any more bounces are detected during a specific time they are simply ignored. The flow diagram in Figure 43 demostrates the flow of the program to determine if the switch contact has stopped bouncing.

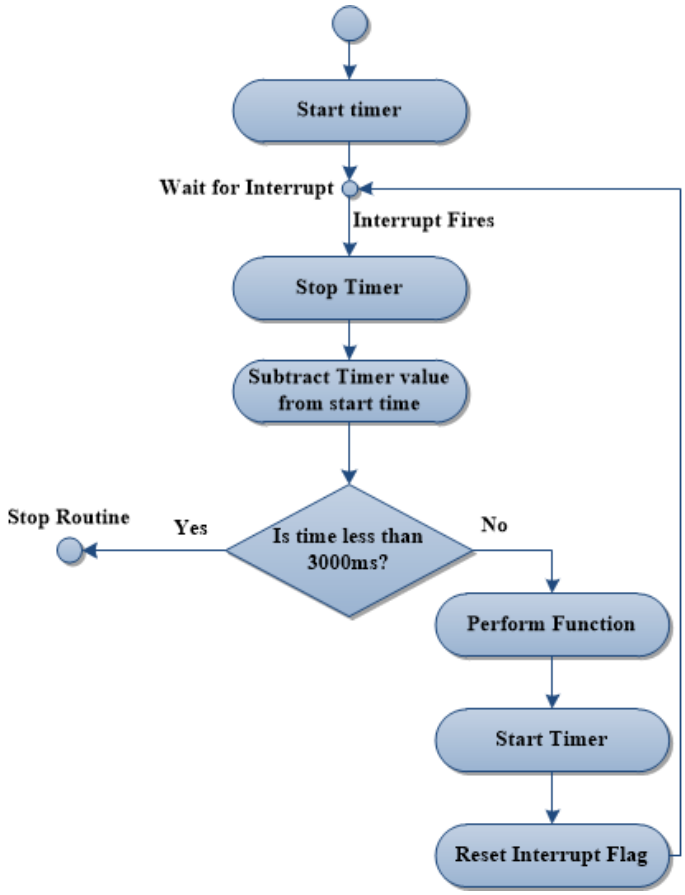



**Figure 43 Software switch de-bouncing**

## 3.4 Active re-coater monitoring system test rig

To test the working of the active monitoring system, a testing rig was constructed based upon the basic design principles as discussed previously in this chapter.

### 3.4.1.1 Camera benchmarking

For the sake of simplicity, the structural design of the initial testing rig was built using Lego construction blocks as the design can easily be changed. This was identified as a more cost-effective method of building temporary structures for research applications. The test rig that was used to perform benchmarking tests on the camera setup is displayed in Figure 44.

**Figure 44 Camera test rig**

The objective of the tests is to determine whether the theoretical calculations for both the smallest defect size and the focal length calculation does indeed translate into practical applications. To determine the capability of the camera setup, a 1951 USAF resolution chart will be used.

The resolution chart, as demonstrated in Figure 45, consists of groups of bars arranged in descending thickness size.

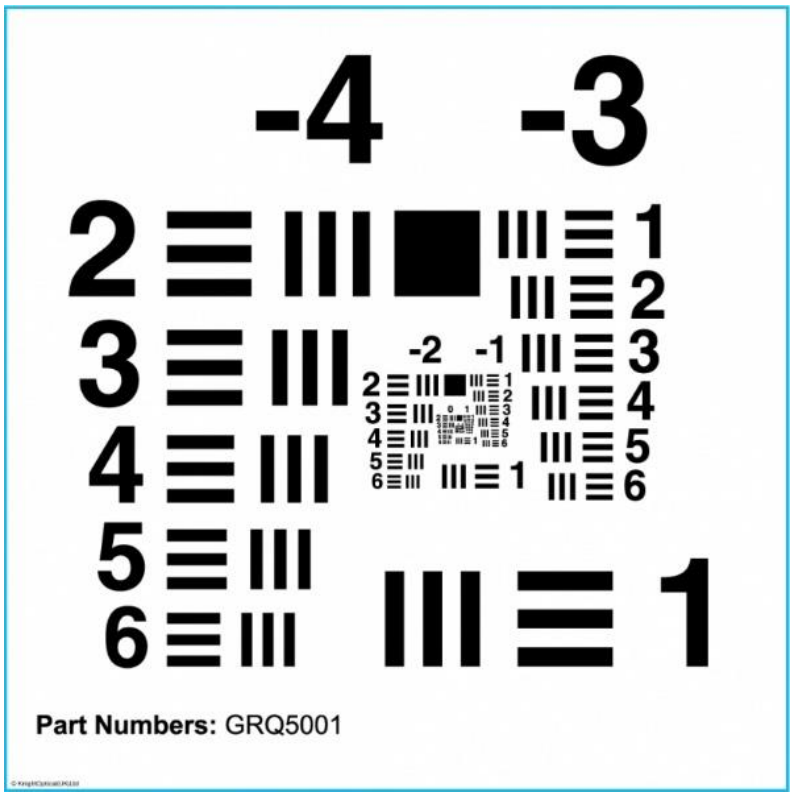



**Figure 45 1951 USAF resolution chart(Knights Optical 2014)**

Each bar on the chart is called an interval, and each grouping of horizontal and vertical intervals is called an element. These elements are given a number and assigned to a specific group that is also numbered. There are 6 elements per group, and the groups are numbered from -4 to 9. The width of the intervals measures from a maximum of 2 mm down to the smallest bars that measure 0.00078 mm (Pyser-SGI Ltd 2010).

The chart was placed at the same working distance that the camera will be mounted at inside the machine. The captured image of the chart will then be analysed to determine the smallest horizontal and vertical intervals that can still be clearly identified at the specified working distance. The width of the intervals is specified by Table 6. Some columns of the interval list have been omitted for clarity.

The full table with all the interval values can be seen in Annexure A for further reference.

**Table 6 USAF 1951 resolution chart, width of bars in mm(Pyser-SGI Ltd 2010)**

| Element number | Group number | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **-2** | **-1** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **1** | 2.00 | 1.00 | 0.500 | 0.250 | 0.125 | 0.063 | 0.031 | 0.016 | 0.0078 | 0.0039 | 0.0020 |
| **2** | 1.79 | 0.89 | 0.446 | 0.223 | 0.111 | 0.056 | 0.028 | 0.014 | 0.0070 | 0.0035 | 0.0017 |
| **3** | 1.59 | 0.79 | 0.396 | 0.198 | 0.099 | 0.050 | 0.025 | 0.012 | 0.0062 | 0.0031 | 0.0016 |
| **4** | 1.42 | 0.71 | 0.355 | 0.177 | 0.088 | 0.044 | 0.022 | 0.011 | 0.0055 | 0.0028 | 0.0014 |
| **5** | 1.26 | 0.63 | 0.315 | 0.158 | 0.079 | 0.039 | 0.020 | 0.0098 | 0.0049 | 0.0025 | 0.0012 |
| **6** | 1.12 | 0.56 | 0.281 | 0.140 | 0.070 | 0.035 | 0.018 | 0.0088 | 0.0035 | 0.0022 | 0.0011 |

Using the constructed test bench as displayed in Figure 44, a series of images of the resolution chart was captured and subsequently examined. All the images in the experiment were captured at a working distance of 440mm as this was determined in the previous section to be the working distance at which the camera will be mounted at inside the machine. The first image that was captured is displayed in Figure 46.

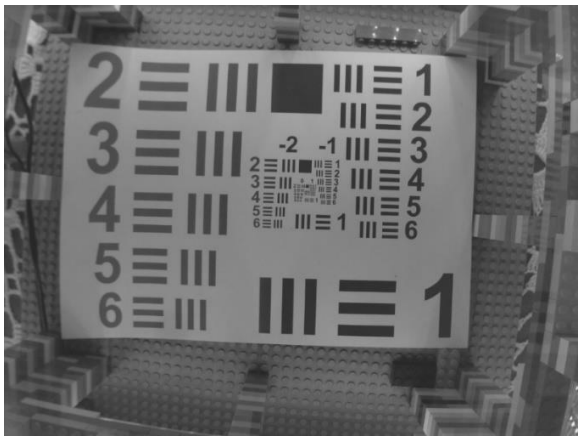


**Figure 46 FOV with 1951 USAF resolution chart**

The area of interest of the image in this experiment is the centre of the resolution chart. As shown in Figure 46 most of the larger features are clearly visible. To determine the

smallest identifiable feature, the centre of the image will be cropped to determine which of the smallest groups can be identified. The image displayed in Figure 47 shows the cropped part of the image to display the smaller intervals.

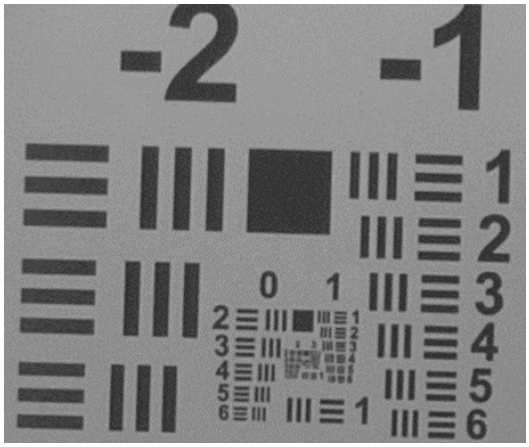



**Figure 47 Cropped image of the resolution chart centre**

As demonstrated in Figure 47 the smallest intervals that can still clearly be identified are the intervals from element 3 inside group 1. The intervals in this element have a line thickness of 198µm, which means that the smallest feature that can theoretically be clearly identified by the camera is a feature of 198µm and bigger. Although the smallest calculated shadow was 184µm and the smallest feature that is detectable by the camera is 198µm, the performance of the camera was considered acceptable as there is only a 6µm difference between the calculated and physically verified features. Although the data on this captured image does not necessarily translate into the camera's capability to detect all the defects that may occur, it does demonstrate the camera's capability to detect very small details at the specified working distance of 440mm. The reason that the results from the resolution chart do not necessarily translate to the defect detection capability of the camera, is because there is a much clearer contrast difference between the black and white intervals on the resolution than what a defect on the powder bed would have. This is where the lighting method's performance becomes critical as it will need to create shadows with as large a contrast difference as is possible. Because there are not a lot of data available about defects that occur in powder-based AM technologies, it is very difficult to determine the actual sizes of defects that may occur without an initial gathering of data about defects.

Therefore, further testing will be conducted on the Voxeljet VX500 using the selected camera to determine if the camera can detect replicated defects on the powder bed.

#### 3.4.1.2 Defect detection process

Some of the techniques that will be used in the defect detection process were drawn from existing methods used by autonomous driving cars and line following flying drones. One of the requirements of the process is an image of the powder bed under its ideal state. This image is key, as one of the first steps in the process is to compare all the captured images to the original image. The flow diagram, shown in Figure 48, displays the overall process that will be followed to analyse the images for any potential defects. Since the diagram is not clear enough to be read in the text, it will be included as Annexure B. The complete source code for the program is attached in Annexure C.

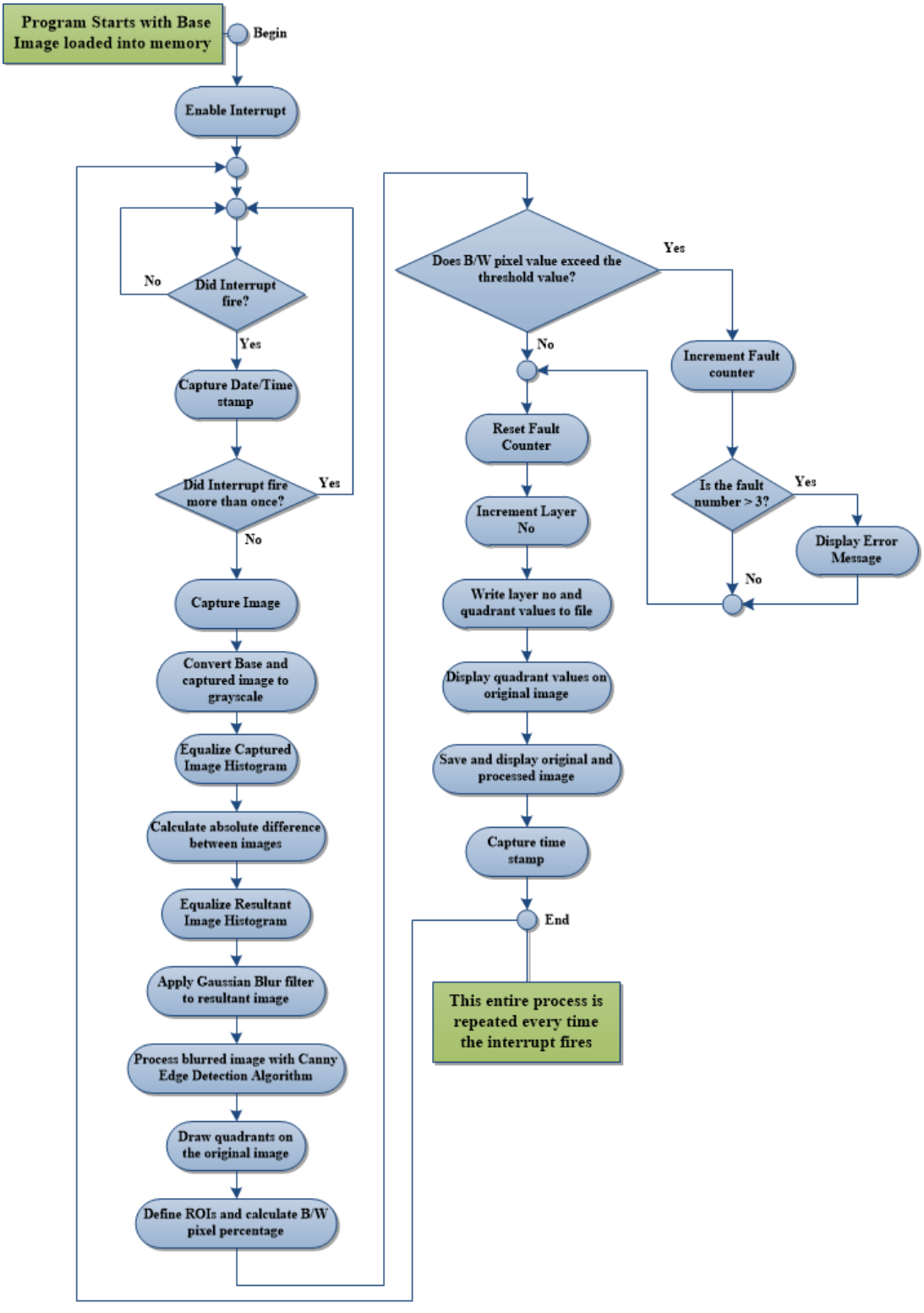



**Figure 48 Flow diagram of image processing program**

In order to visually explain the steps that is followed by the image processing program in Figure 48, the images for each step of the image processing program is displayed in Figure 49.

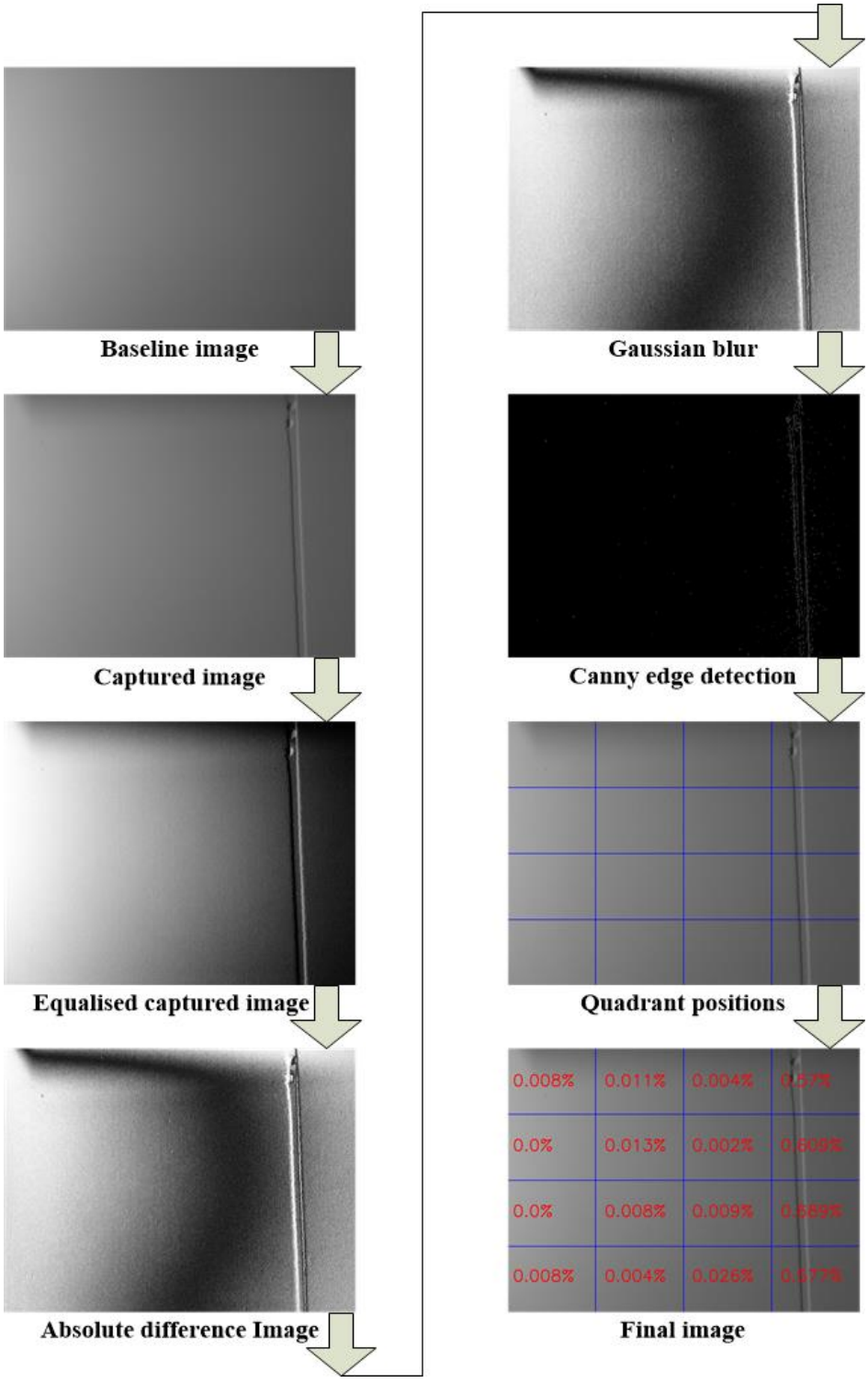



**Figure 49 Flow diagram of images**

1. The baseline image as displayed in Figure 49 is loaded into the program so that all captured images can be referenced to it. The baseline image is then converted to greyscale as displayed.
2. An image of the powder bed demonstrates the powder bed with a replicated re-coating error. This image is also converted to greyscale before any further processing is done.
3. The histogram of the captured image will be equalized to increase the contrast of the image and any possible defects present on the powder bed as can be seen in step 3.
4. An empty array will be created with the same dimensions as the original image. This array will be used for storing the absolute difference image between the original and captured image as demonstrated in step 4.
5. A Gaussian blur filter is then applied to suppress some of the high frequency noise present in the image. The image after applying the Gaussian blur filter is shown in step 5. Since the filter is not too aggressive, the visual changes to the images is not clearly visible, but a large amount of high frequency noise have been supressed.
6. The canny edge detection algorithm was used to identify the edges of any possible defects present on the image as shown in step 6. When looking at the image on step 5 in Figure 49, there is a large reverse C shaped shadow on the image. This shadow does not affect the performance of the canny edge detection algorithm, as it searches for a more definite contrast difference between neighbouring pixels before it is identified as an edge. This is due to the upper and lower threshold levels being provided to the canny edge detection algorithm.
7. The image is then divided into smaller quadrants to identify more accurately the position of defects present on the powder bed. For this purpose, the image has been divided into 16 equal sized blocks in a 4 x 4 matrix as demonstrated in step 7. Each of the quadrants will be processed individually with a specified region of interest algorithm. Since the canny edge detection algorithm delivers a binary image, meaning it will only consist of white or black pixels, the white pixels indicate the edges of the defect. The specified region of interest algorithm then determines the ratio of white pixels in comparison to black pixels and this value is expressed as a percentage.

8. The percentage values of each quadrant are overlaid onto the image for visual verification purposes as displayed in step 8. The layer number, percentage values, as well as the date and time that the image was captured is also stored as a CSV file for further analysis. This image analysis process will be repeated for each image that is captured. The initial captured image, as well as the final processed image of each layer will also be stored for post build analysis purposes.

In the last portion of the program, all the percentage values recorded for each quadrant of the image is compared to a threshold value that determines whether the feature that was detected by the image processing program is a defect or not. This threshold value will be determined by first establishing a baseline percentage values for a defect by monitoring actual builds using the active re-coater monitoring system. If a percentage value exceeded the threshold value, a counter would be incremented. Once the counter exceeded a total count of 3, a visual alert would be displayed on the software GUI indicating that a recurring defect had occurred on the powder bed.

## 3.5 Summary

In this chapter the specifications, features, advantages and disadvantages of the selected 3 active monitoring technologies was reviewed. After comparing all the different features of the different technologies, computer vision technology was selected as the most suitable technology to develop an active re-coater monitoring system. The specifications for the computer vision camera was also calculated so that a suitable camera could be selected that will satisfy all the requirements. The Raspberry Pi camera module was selected as it matched all the calculated criteria.

Since the OpenCV library was selected for integration into the image processing program, the various algorithms that is used to process the captured images was discussed. The triggering mechanism and adjoined circuitry was also briefly discussed.

The initial testing of the camera hardware and software was also executed. The first tests that were conducted were to determine the smallest feature that can be clearly identified by the Raspberry Pi camera module. This was done using a 1951 USAF

resolution chart placed at the same working distance that the camera will be mounted at, inside the Voxeljet VX500. The benchmarking tests revealed that the Raspberry Pi camera module can capture small features of up to 198µm in width. This proved that the camera does have the capability of capturing very fine details that are very close to the calculated requirements of 184µm. Once the benchmarking of the camera module was completed, the various steps that will be followed by the image processing program was also reviewed.

In Chapter 4 the proposed active re-coater monitoring system will be integrated into the Voxeljet VX500, where different types of defects will be replicated, and the effectiveness of the active re-coater monitoring system to detect defects under production conditions will be determined.

# Chapter 4 Experimentation and Validation

## 4.1 Introduction

The primary discussion of this chapter would be to review the results that will be collected after the proposed active re-coater monitoring system has been integrated into the Voxeljet VX500. The FOV of the camera will first be verified to ensure that the physical FOV of the camera matches the calculated values. Then several types of defects will be replicated on the powder bed surface to ensure that the image processing techniques can detect the various types of defects under production conditions. These tests will also be used to select the optimum parameters for the various image processing functions. Once satisfactory results had been recorded with the initial tests on the Voxeljet VX500, a case study will be conducted using an actual build job. The purpose of this case study is to validate the threshold value for the black to white pixel percentage value that determines when a defect or re-coating error has occurred somewhere on the powder bed, as well as to validate the overall effectiveness of the system's capability to detect defects.

## 4.2 VX500 system integration

The active re-coater monitoring was installed into the Voxeljet VX500 to test the system under actual operating conditions. There were several hardware modifications that had to be made to the original machine to be able to install the system into the Voxeljet VX500. The identified mounting location of the camera was directly above the powder bed as demonstrated in Figure 50.


**Figure 50 Voxeljet VX500 hood lid**

This position for the cameras also allows the cabling that is required to be easily routed to the outside of the machine for the control of the camera. The clearance between the binder spraying head and the hood lid of the machine is only 11mm. This means that the camera with its bracket had to be slimmer than this clearance gap to ensure that the camera does not become entangled with the binder head. The clearance gap is demonstrated in the image in Figure 51.

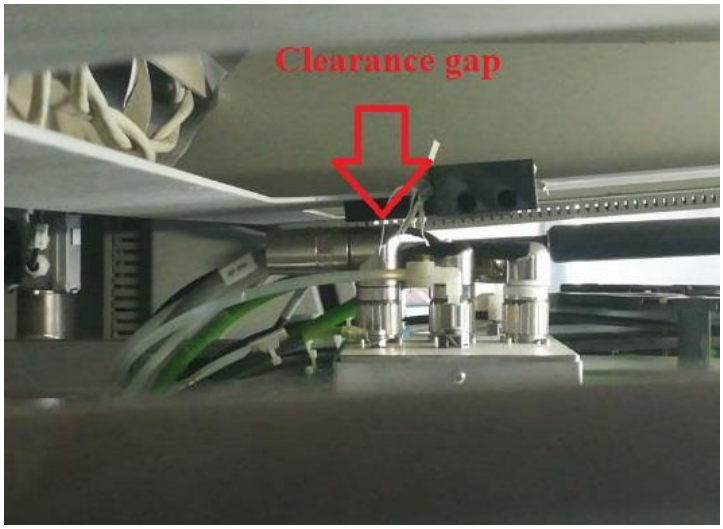



**Figure 51 Clearance between binder head and hood lid**

Because of this clearance limitation, a special housing with a mounting bracket had to be designed for the camera unit to ensure that the camera is not easily damaged and is firmly affixed in its position. The camera housing and its brackets were manufactured using an FDM type 3D printer using PLA. Once the brackets were printed, the camera was mounted into place using the housing and brackets as displayed in Figure 52.

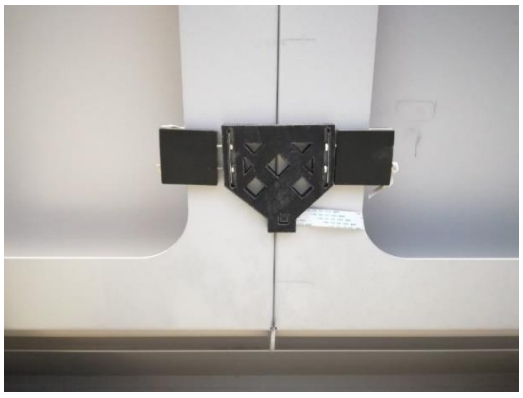


**Figure 52 Mounted camera housing and brackets**

The Raspberry Pi micro-computer unit was mounted outside of the machine. The limit switch for the triggering of the camera also required a special bracket that was designed to be attached to the side of the aluminium extrusions that houses the Y-axis rails.

The bracket was also 3D printed using PLA. The image in Figure 53 displays the mounting bracket with the limit switch attached and attached to the side of the aluminium extrusion.

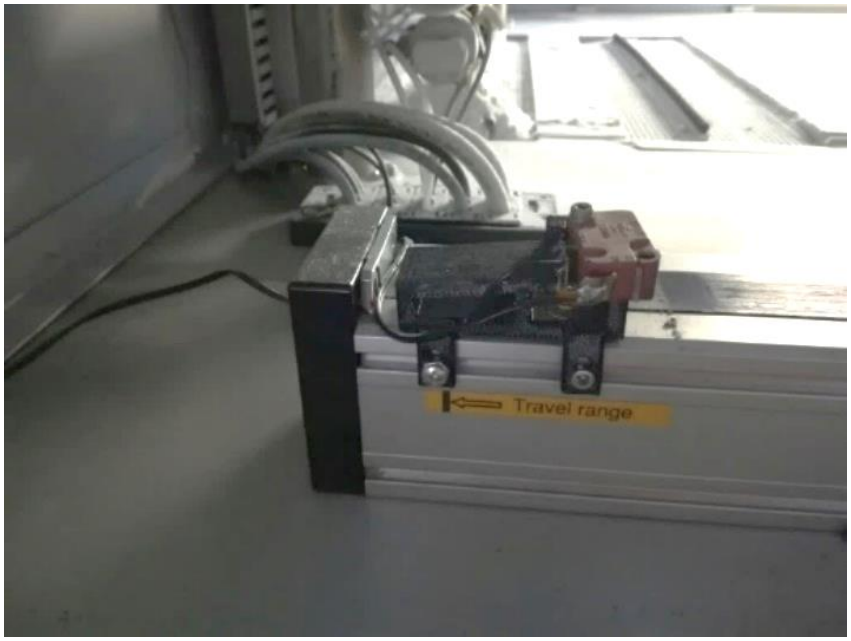



**Figure 53 Mounted limit switch bracket**

## 4.3 System setup

After the system has been installed in the Voxeljet VX500, the Raspberry Pi camera's FOV had to be tested. Although the FOV of the camera lens was calculated in Chapter 3, it is still necessary to verify the FOV after installation as the focal length and FOV of the lens does not always match the actual performance specified by the manufacturer.

To verify the FOV, an image was captured with a ruler placed horizontally and vertically across the powder bed.

The image in Figure 54 displays the FOV size of the camera with the ruler measuring the vertical axis.

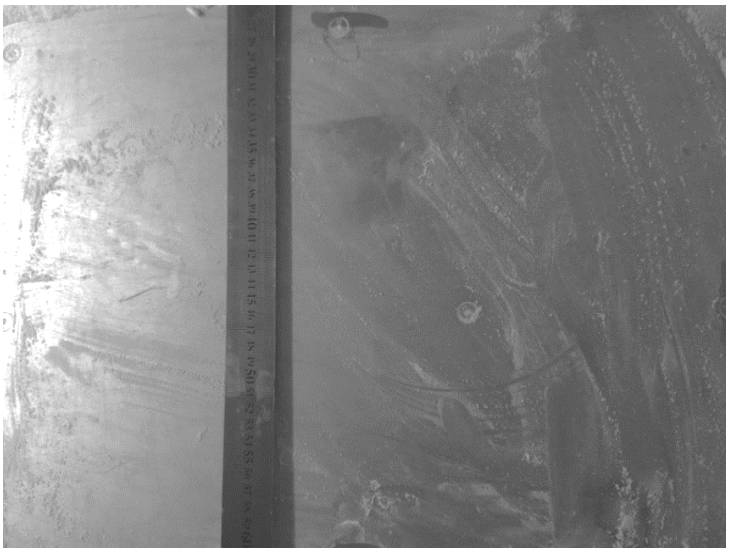


**Figure 54 Raspberry Pi camera vertical FOV**

Although all the markings on the ruler do not appear very clear on the in-text image, if the image were to be viewed at full screen size the markings are very clear. When viewing the image at full screen size the vertical FOV was measured at 350 mm. An image was also captured with a ruler placed on the horizontal axis as displayed in Figure 55.

**Figure 55 Raspberry Pi camera horizontal FOV**

The measurement as taken from the second image captured when viewed at full screen size of the ruler measures the horizontal FOV at 455 mm. Thus, it can be concluded that the FOV of the Raspberry Pi camera module at a working distance of 440mm will be 455mm x 350mm. However, when comparing these values to the actual powder bed size, the FOV of the camera module is smaller than the size of the powder bed. With the camera being positioned to be in the centre of the powder bed, a border of 22.5mm across the width and 25mm across the length of the bed could not be captured. The recommended stacking rule is not to place the part to be built closer than 5 mm from the side of the build box to prevent binder fluid from being sprayed onto the side of the build box, potentially causing damage to parts when removing the build (Shabanga 2018). If binder fluid is sprayed onto the build box, it will cause the powder to stick to the inside of the build box and require cleaning after each build using special chemicals. This means that only a small area of between 18.5 – 20mm of the powder bed will not be covered by the camera module. Thus, the small area not covered by the camera was not considered to be a problem.

### 4.3.1 VX500 simulated recoating errors results

The replicated re-coating errors and defects that will be discussed, was used to ensure that the camera is in the correct position, that the lighting was able to provide adequate illumination and to test the effectiveness of the image processing program to detect defects. The re-coating errors and defects that was replicated is similar to those that occur during actual builds. These tests also provided the opportunity to gather real-world data on the overall effectiveness of the system and on the formation of defects and their possible causes.For the initial tests of the active re-coater monitoring system, these initial parameters were selected for the Gaussian blur filter:

- **Gaussian Blur Filter**
    - Standard Deviation = 11 x 11-pixel matrix
    - Kernel size = 2

If these selected parameters do not produce the desired results, the parameters will be adjusted accordingly.

- **Powder clumps**

The first re-coating error that will be simulated is when a piece of debris or a powder clump falls onto the powder bed. This type of error occurs when powder or debris that is stuck to the outside of the re-coater vibrates loose and falls onto the surface of the powder bed. This image was captured during an actual build, as the outline of the part geometry can be seen on the image. This did not have an effect at all on the system's ability to identify defects and did not cause false positives. The original captured image of the powder bed is displayed in Figure 56.

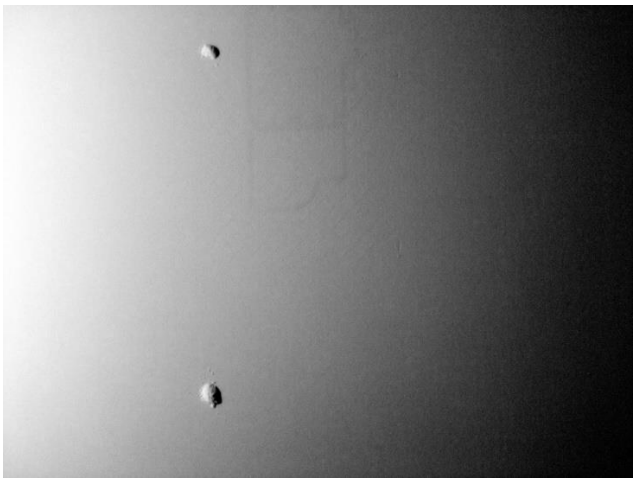


**Figure 56 Original image of powder clumps**

The second image in Figure 57 displays the output of the program after the image has been processed.

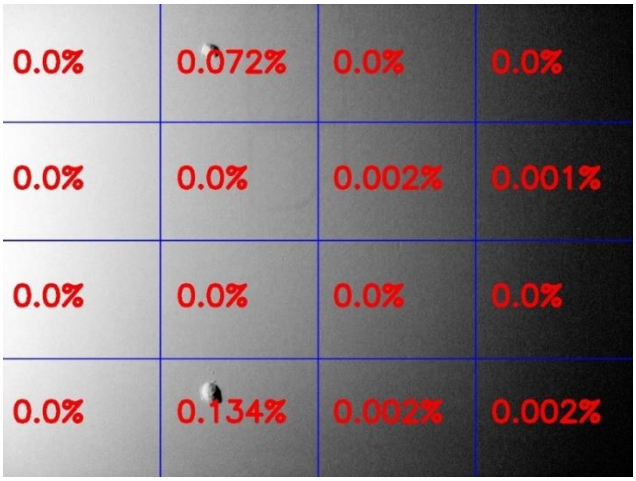



**Figure 57 Processed powder clumps image**

Observing at the image in Figure 57, the two powder clumps that fell onto the powder bed occurred on the top and bottom of the image. Examining the two quadrants that contain the defects, it could be seen that both the quadrants recorded a significant increase in values, indicating that a feature had been detected inside these quadrants.

For this image the initially defined parameter kernel size was selected. Since the values recorded indicated a significant change when compared to the surrounding quadrants, the parameter kernel size was left at its default setting of 2.

- **Clogged re-coater**

The next type of re-coating error that was simulated is a clogged re-coater. This re-coating error can be caused by debris stuck in the re-coater, or powder that has a very low flowability. This type of re-coating causes a line to form on the powder bed, as demonstrated in Figure 58. It is worth noting that this defect was formed after a single layer, thus the defect depth would be 150µm.

**Figure 58 Single re-coating error on the powder bed**

When examining the image in Figure 58, the re-coating error was replicated on the far side of the powder bed to test if the illumination was still bright enough to create a detectable contrast difference between the re-coating error and the rest of the powder bed.

The image in Figure 59 displays the output image of the re-coater monitoring system. When analysing the output image, it is evident that the monitoring system could detect the re-coating error that had occurred on the powder bed.

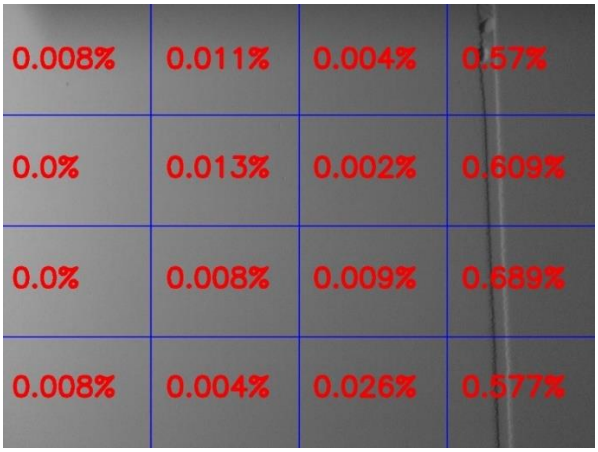



**Figure 59 Processed re-coating error image**

When looking at the black to white percentage values, a significant increase in the values can be seen on all the quadrants covering the defect that had occurred on the surface.

Also, since the values recorded was substantially larger than the values recorded on the quadrants surrounding the defect, the initially selected kernel size values of 2 were considered optimal. This means that the system could clearly differentiate between the area where the re-coating occurred and the areas surrounding the re-coating error.

- **Shallow line defect**

The next defect that was investigated is a minor defect. This type of defect usually caused by debris that is stuck to the re-coater blade and gets dragged across the powder bed surface. Sometimes impurities are present in the powder, and then the foreign particles are dragged across the surface of the powder bed. However, since a lot of these smaller types of defects are very shallow, they are considered non-critical defects. Such a type of minor defect is displayed on the image in Figure 60. Since this defect was captured during a build, the part geometry outline is visible on the image, however, this did not influence the detection capability of the system and did not trigger a false positive.

**Figure 60 Shallow line defect across the powder bed**

These types of defects are usually only one layer deep and can usually be covered by subsequent recoating operations. However, often they may also be the indicator that a much bigger defect or re-coating error is forming. The image in Figure 61 displays the processed image of the shallow line defect.


**Figure 61 Processed shallow line defect image**

When examining Figure 61, the recoating error runs vertically across the image, and have been detected by four quadrants. It is also clear that the percentage values recorded for these quadrants are higher than the recorded values for the surrounding quadrants, indicating that the recoating error has successfully been detected by the system. For this image, the initially selected kernel size value of 2 were used and no

further adjustments to this value was deemed necessary as there was a clear enough difference between the recorded values.

- **Re-coater short feeding**

The last type of re-coating error that was replicated is a re-coater short feeding error. During the process of performing tests of the Voxeljet VX500, this was a condition that occurred due to a low powder flowability. The captured image in Figure 62 displays a very good example of the powder bed during re-coater short feeding.

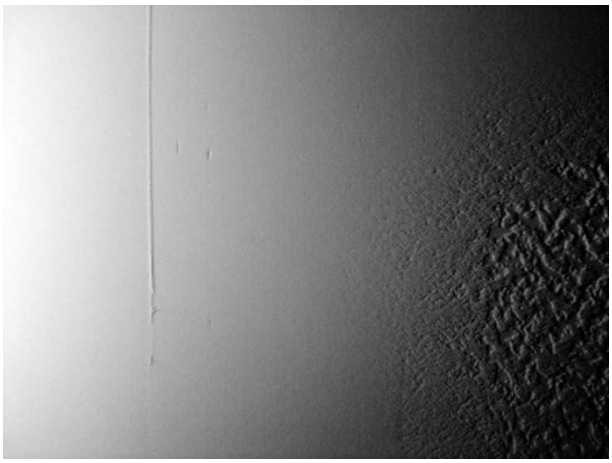


**Figure 62 Re-coater underfeeding**

There are two types of re-coaters error present Figure 62, the first and most obvious being the short feeding on the right-hand side, and the smaller mark and line on the left-hand side. Figure 63 shows the processed image of the re-coater underfeeding.

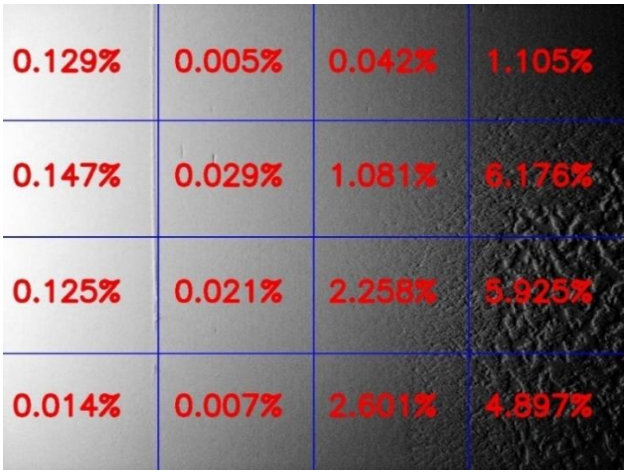
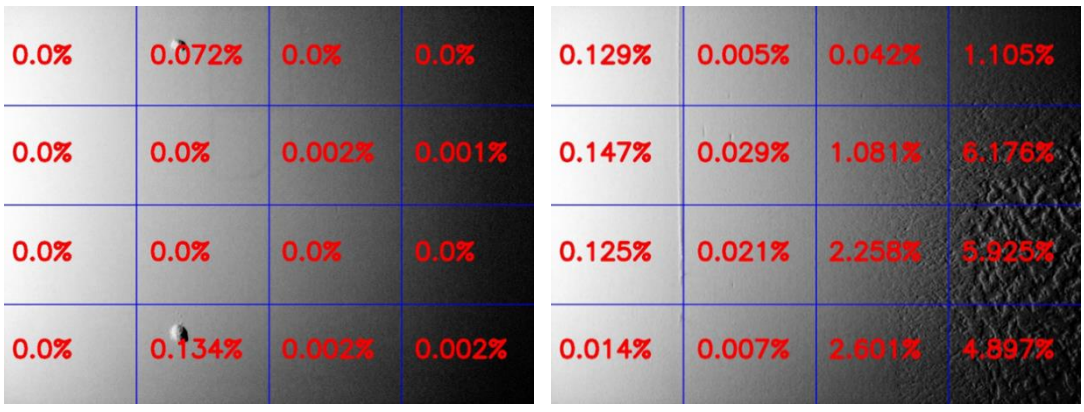



**Figure 63 Processed re-coater underfeeding image**

When examining the processed image, all the defects present on the image was successfully detected. There are a few small defects inside the second column of quadrants that did not record a very high value. Although there was an increase in values, the difference between these recorded values and the surrounding quadrants was not as great as with the previous defect. However, the increase in recorded values still indicated the system's capability to record these smaller types of defects. The black to white pixel percentage values on the right-hand side of the image was significantly higher than the rest of the powder bed, the main reason for this being the number of edges detected due to the rough pattern caused by the re-coater underfeeding. These higher values are not considered a problem, as this gives an indication of the severity and the size of the defect on the powder bed.

## 4.4 Threshold value

After conducting the initial tests using the replicated defects and re-coating errors, a defect threshold value must be determined. This threshold value will be used to determine the lowest black to white pixel percentage that must be recorded before a captured feature is considered a defect. Thus, a suitable method had to be used to determine a suitable threshold value. When examining the captured images of the replicated defects, quadrants with the smallest values that still had visible defects will be considered. The two images with the smallest recorded values are displayed in Figure 64.




(a) (b)

**Figure 64 Defects with smallest values**

When examining the smallest defect in in Figure 64b, the recorded value for this small defect is 0.029%. However, in the quadrant below this recorded defect, a value of 0.021% was recorded although this quadrant contained no visible defect. Thus, this recorded value was not considered reliable as the values are too close to each other. The second value that will be considered as a threshold value is from the defect in the quadrant of the second column and the second row of Figure 64a. In this quadrant the defect is clearly present on the image, and the value recorded for this quadrant shows a clear difference between the value of the quadrant with the defect present and the surrounding quadrants. When comparing the value recorded by this quadrant to some of the other replicated defects, the value appeared to be closely in line with other defects. Thus, for the purposes of the further experiments, this value of 0.072% will be used as the threshold value. If during the process of the case study it is determined that this selected threshold value is not sufficient, a new value can be selected.

## 4.5 Case study

In order to assess the performance of the active re-coater monitoring system, a series of case studies was conducted to verify the performance of the system. A total of 4 case studies was conducted on the VX500 using the active re-coater monitoring system. However, not all the results will be discussed, but the results can be made available upon request. The build job that was monitored for this specific case study consisted of several small and large parts and consisted of 1946 layers. The estimated time to print the entire build was 16 hours. For the purposes of the case study, two images of the powder bed for each layer of the STL was stored. The first image being the originally captured image, and the second being the processed image. Both images were stored using the current date and time stamp as the image file name. A separate text file was also stored that recorded all the data in CSV format. The data stored inside the file consisted of the layer number, date and time stamp and the black to white percentage values of each of the 16 quadrants. All of this data was recorded for each layer of the build. During the build process, numerous re-coating errors and defects had occurred on the powder bed. Although this was unfortunate, it provided the ideal conditions to determine the effectiveness of the system. Once the build process had finished, the data from CSV file as well as both the processed and unprocessed image

were analysed and reviewed. The first step that was performed was to plot all the data as graphs to graphically determine whether there were possible defects or recoating errors. Once a possible defect was identified, the specific image was retrieved, and the severity of the defect was analysed. All the images captured for the build was manually reviewed to validate the results recorded and ensure that no defects went through undetected. Figure 65 demonstrates the stored data of the black to white pixel percentage values. Since the graph is not clear enough in the text to analyse, a full-size graph will be included as Annexure D.

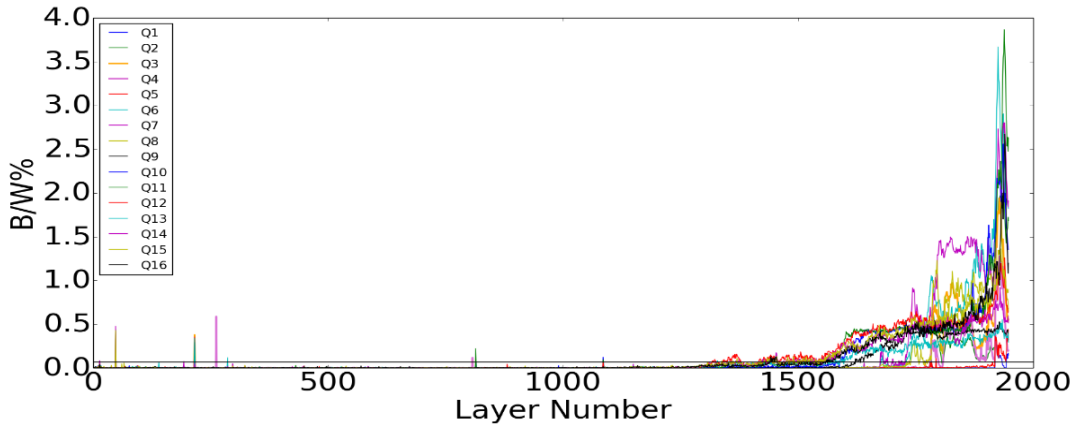



**Figure 65 Graph of black to white pixels percentage values**

The legend shown in Figure 65 represents the sixteen quadrants marked from Q1 – Q16. The position of each quadrant on the powder bed is displayed in Figure 66.

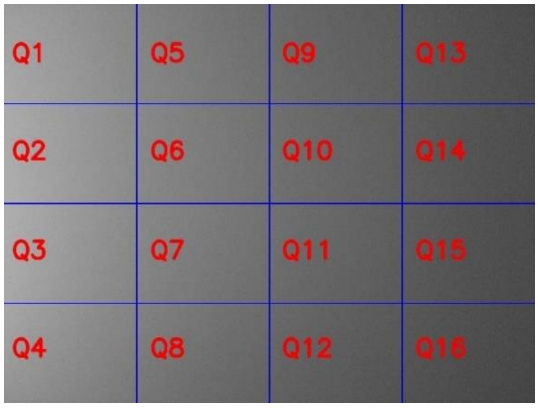



**Figure 66 Quadrant numbering**

Each of the different quadrants on the image were given a different colour on the graph to make it easier to match a peak on the graph with a specific quadrant.

As discussed previously, the error detection threshold value was set to 0.072%. However, as discussed, if after analysis this value is determined to be too low or too high, an appropriate recommendation will be made. To make an analysis of the data in the graph easier to interpret, the maximum value on the Y-axis was set to 0.072% as seen in Figure 67. After all the data has been reviewed from the CSV file, each of the images captured will also be briefly analysed to ensure that no defect or re-coating error may have been skipped. Considering the graph in Figure 67, all the peaks that exceeded the 0.072% threshold have been highlighted with red arrows. These points on the graph was matched to their individual photograph(s) to evaluate the defects or re-coating errors that may have occurred, causing a spike in the black to white percentage values.

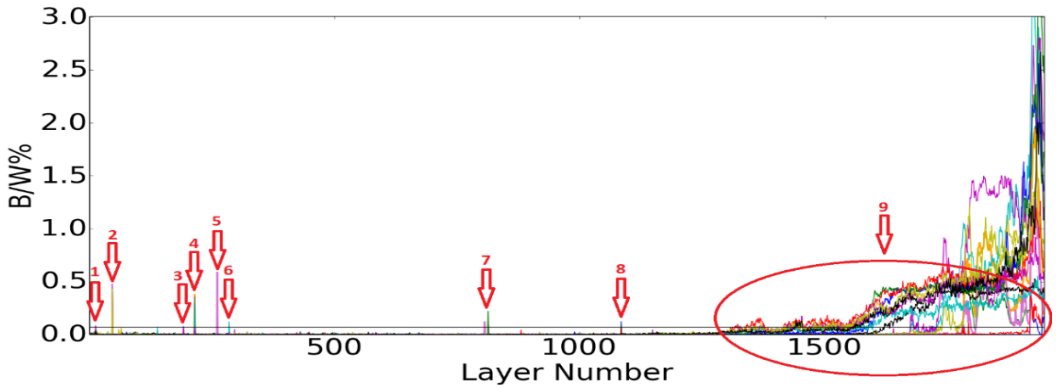



**Figure 67 Black to white pixel percentage graph with spikes indicated**

The first spike on the graph that will be examined is the red arrow marked as number 1. When zooming into the spot on the graph, the spike had occurred on layer no. 14 as displayed in Figure 68.

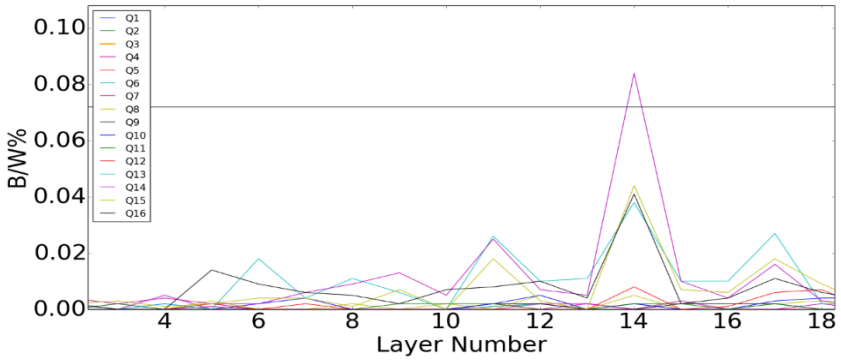



**Figure 68 Graph of peak no. 1**

When examining the legend on the left-hand side of the graph, it can be seen that the defect had occurred in quadrant 14. What can also be seen is that several other quadrants also recorded a change in black to white percentage values. However, most of the other quadrants did not record values higher than the threshold. In order to determine what had occurred on the powder bed, it is necessary to review the image that was captured for layer no. 14. The processed image for layer no. 14 is displayed in Figure 69.

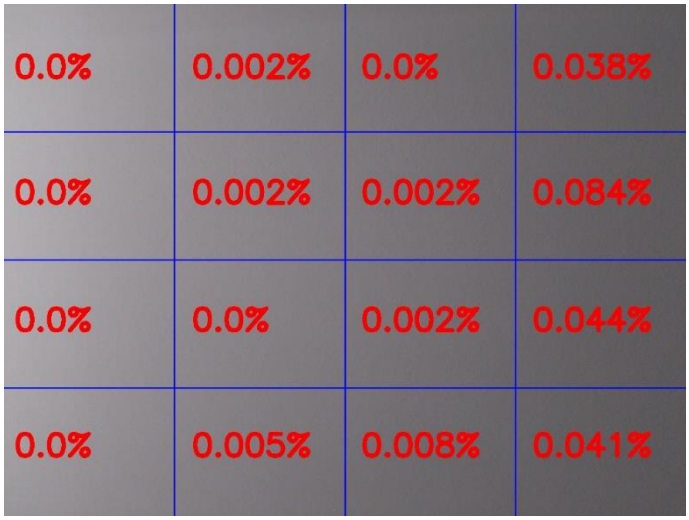



**Figure 69 Processed image for layer no. 14**

When examining the image in Figure 69, it is not clear as to what had caused the sharp increase in percentage values. When viewing the image at 100%, the only issue that is slightly visible is that the surface re-coating quality is slightly rougher than normal. This is not an issue as this is not considered a defect because excess powder may have been spilled on the powder bed by the re-coater and this often happens when the re-coater is overfilled as the machine refills the re-coater hopper. Therefore, this recorded defect will not be considered as a valid defect due to the system registering a false positive.

The second spike on the graph that will be examined occurred on layer no. 48 as displayed in Figure 70. When the portion of the graph has been zoomed in, it is possible to see whether the spike was only a single occurrence or was spread over multiple layers.

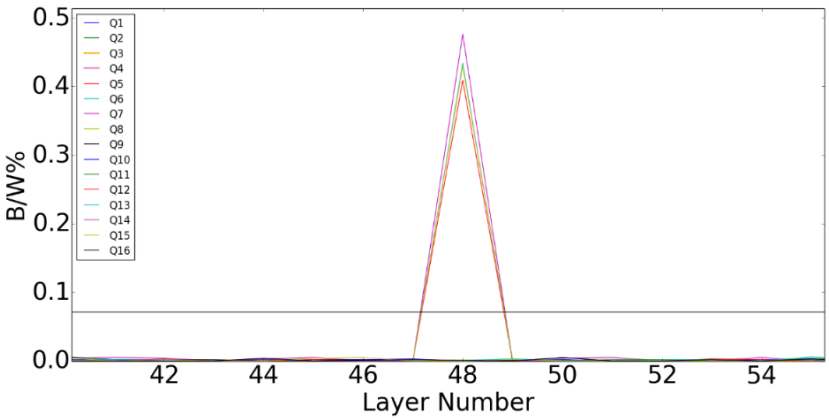



**Figure 70 Graph of peak no. 2**

Examining Figure 70, the layer at which the defect had occurred was identified as layer number 48. The graph showed that multiple quadrants had registered a change in the black to white pixel percentage value. It was determined that the defect had occurred in the direction of re-coating since the colours of the graph correspond to quadrants 5,6,7 and 8. The processed image that was captured for layer number 48 is displayed in Figure 71.

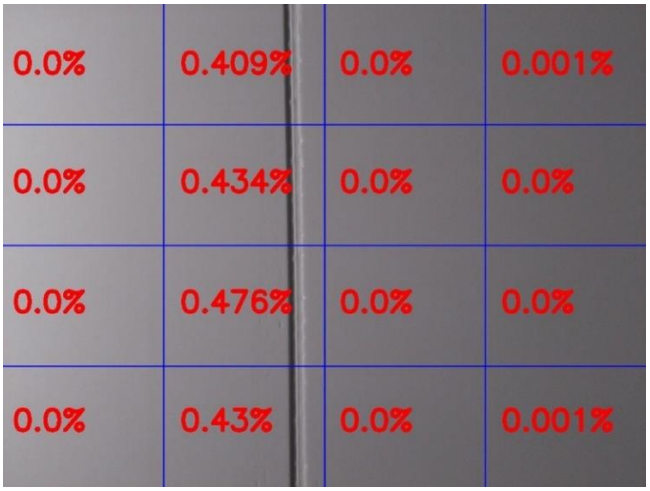



**Figure 71 Processed image for layer no. 48**

Considering the image in Figure 71, it is very clear that a drastic re-coating error had occurred on the powder bed. Also, when examining the image, the quadrants in which the re-coating error had occurred, namely quadrants 5,6,7 and 8, corresponds to the recorded data on the graph in Figure 70. Looking at the values recorded for the black to white pixel percentage, the selected value of 2 for the kernel size of the Gaussian

blur filter appeared to be optimal, as the defect was detected by the system and recorded a black to white percentage value of higher than 0.072%. When examining the re-coating error in Figure 71, it appears that the defect is quite deep, much deeper than a single layer. However, no errors were detected on layers no. 47 and 49 as shown in Figure 72.

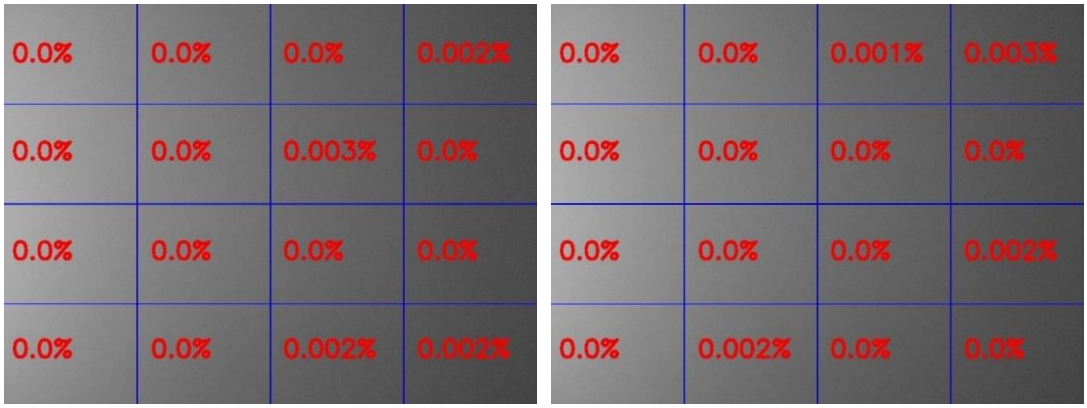



**Figure 72 Images for layer no. 47 and 49**

The only conclusion that can be drawn from these images is that a piece of debris had been dragged across the powder bed. The subsequent re-coating operation then did successfully repair the defect created without damage to the build.

The third spike on the graph that will be investigated occurred at layer no. 193. The image in Figure 73 displays the recorded data for layer no. 193.

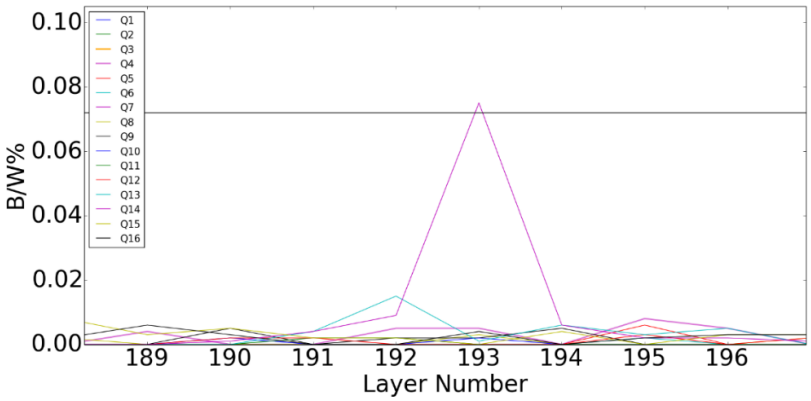
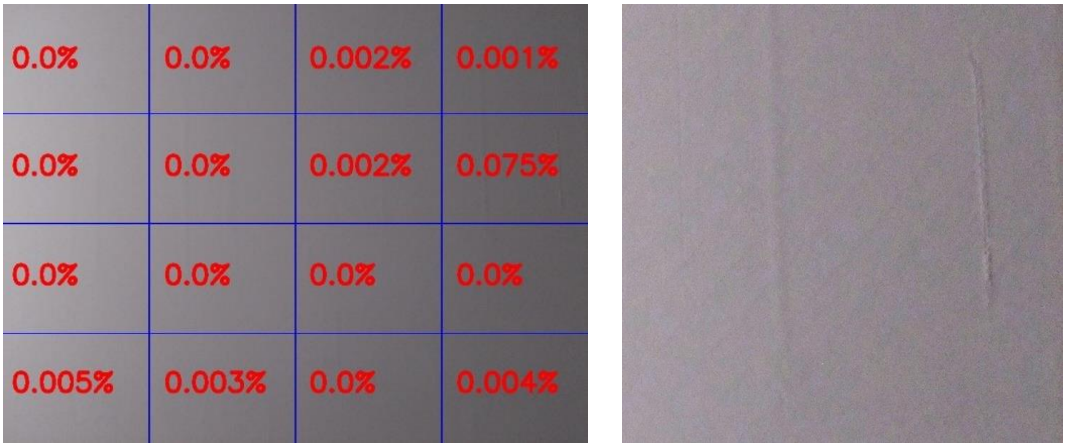



**Figure 73 Graph for peak no. 3**

Upon examination of the graph it can be seen that only quadrant 14 had recorded a change. What can also be seen is that no other quadrants registered a major change of

even lower than the threshold. The processed image that was captured for layer no. 193 is displayed in Figure 74a. When looking at the processed image, a very small defect can be identified that had occurred on quadrant 14.




(a) (b)

**Figure 74 Processed images for layer no. 193**

This small defect appears to be caused by a small clump of powder that fell onto the powder bed and was partially flattened by the re-coater during the re-coating operation as can be seen in Figure 74b. This type of defect is considered as a non-critical defect, as it is indeed a defect, but do not compromise the integrity of the powder bed. When examining the images for the subsequent layers, the defect appears to be repaired by the subsequent re-coating operations. However, these images will not be displayed due to space constraints. The fourth spike on the graph that has been highlighted occurred at layer number 216. The image in Figure 75 displays section of the graph where the second spike had occurred.

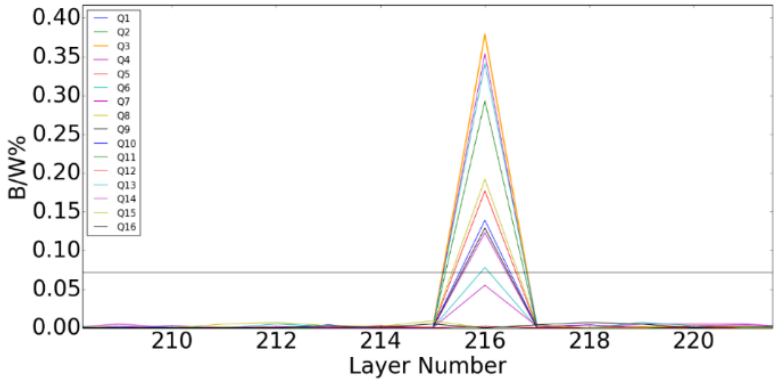



**Figure 75 Graph of peak no. 4**

When examining the spike in the percentage value on the graph, several quadrants had recorded the change in value. A total of 10 quadrants exceeded the 0.072% threshold. This would mean that a relatively large defect or re-coating error had occurred on the powder bed. The image that was captured and processed at layer 216 is displayed in Figure 76.

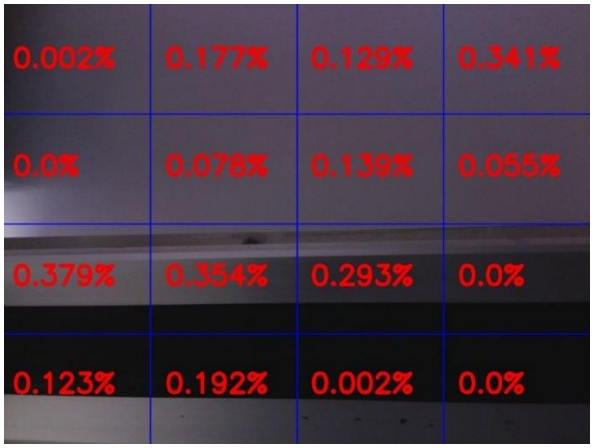



**Figure 76 Processed image for layer no. 216**

Examining Figure 76, the reason for the sudden spike in the black to white pixel percentages is very clear. The re-coater arm was captured during the re-coating cycle. This might have been caused by a glitch that had triggered the camera to capture an image during the re-coating cycle. A possible reason for this occurrence could be static electricity, causing the trigger pin to float to unknown voltages. This kind of problem often happens due to the length of the wire from the limit switch to the Raspberry Pi micro-computer board. In the case of the Voxeljet VX500, the length of the wire exceeds 2 meters in length. Although all the necessary precautions have been taken to prevent this from happening, it is possible to have ghost triggers due to the length of the wire. Thus, a future improvement to the system would be to re-design the triggering mechanism to be more robust to interference.

The fifth spike on the graph that has been highlighted occurred at layer number 262. The image in Figure 77 shows the zoomed-in section of the graph where the third spike in the percentage value had occurred. When comparing the colours of the two peaks to the graph legend on the left-hand side, it was identified that the defect or re-coating error had been detected in quadrants 3 and 4. This once again indicates that the defect

had occurred in the direction of re-coating, indicating a possible re-coating error. However, the value recorded for quadrant 4 is larger than the value recorded for quadrant 3. This means that the assumption could be made that quadrant 4 detected the larger portion of the defect.

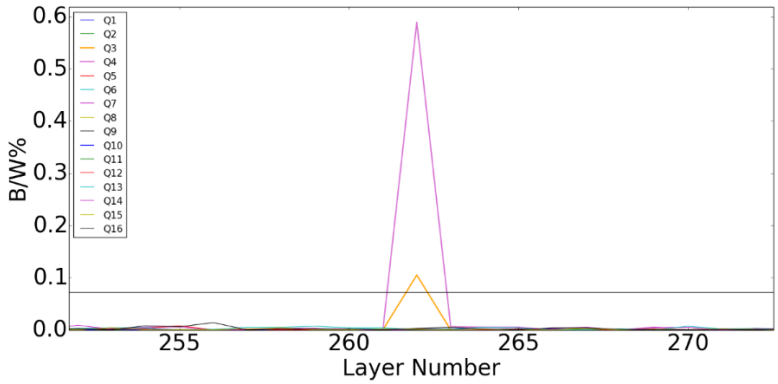



**Figure 77 Graph of peak no. 5**

To verify the assumption made from the review of the plotted data, it is necessary to review the image captured of layer 262. The image that was captured and processed of layer 262 is displayed in Figure 78.

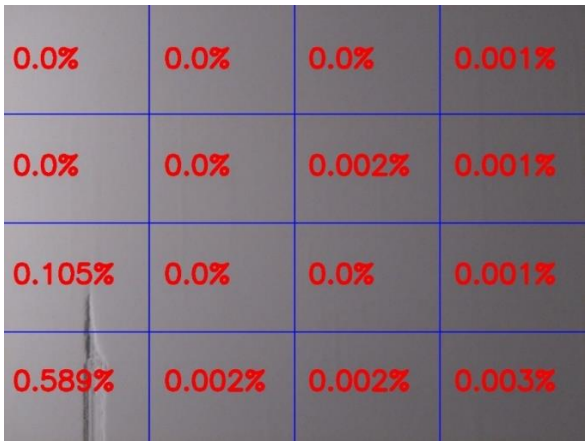



**Figure 78 Processed image for layer no. 262**

As can be seen in the image in Figure 78, a re-coating error had occurred on the bottom left-hand corner of the powder bed. When looking at the quadrant, the re-coating error that had occurred was detected in quadrants 3 and 4. This confirms that the spike that had occurred on the graph in Figure 77 was indeed a defect. When examining the quadrants that had detected the re-coating error, the black to white pixel percentage

value that had been recorded was higher than the preselected threshold value of 0.072%. It was verified that the re-coating error only occurred on a single layer as no visible defects were visible on layers 261 and 263 as shown in Figure 79.

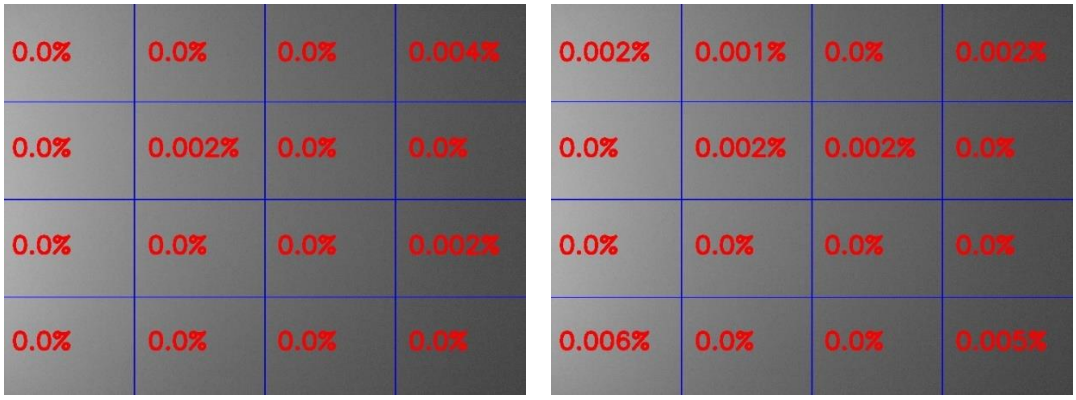



**Figure 79 Processed images for layers no. 261 and 263**

Thus, the only conclusion that can be made from these images is that a clump of powder or debris got stuck in the re-coater, causing the defect. Even for quadrant 3, which contained only a small portion of the defect, the percentage value recorded was still higher than the threshold value. The sixth spike that will be investigated had occurred on layer no. 286. The recorded data for this layer is displayed in Figure 80. When consulting the legend on the left-hand side of the graph, it can be seen that the defect had been recorded in quadrant 13.

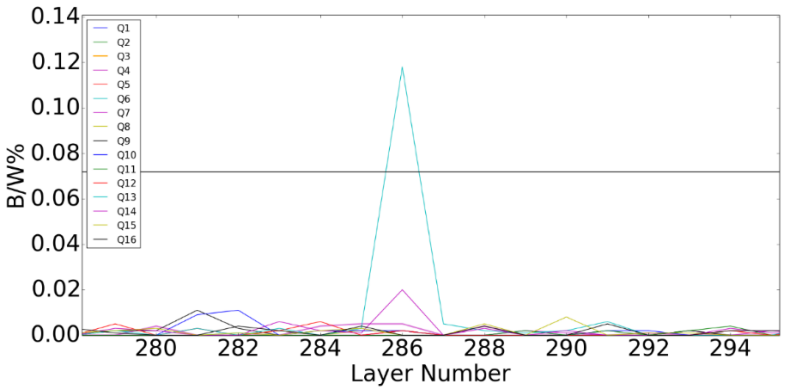



**Figure 80 Graph of peak no. 6**

When looking at the graph, a change in black to white percentage values had been recorded across quadrants 13 and 14. However, only quadrant 13 had recorded a change in value higher than the threshold value. This means that the defect had

occurred in the direction of re-coating and that a larger portion of the defect was contained in quadrant 13 than quadrant 14. To verify this assumption, it is necessary to review the image that was captured for layer no. 286 as displayed in Figure 81.

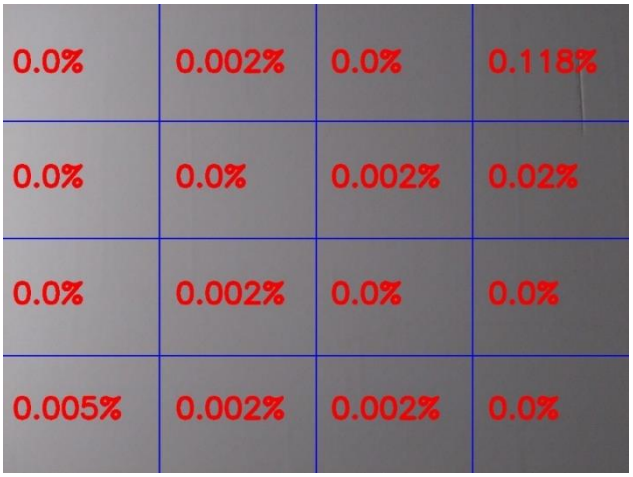



**Figure 81 Processed image for layer no. 286**

When examining the processed image for layer no. 286, the assumption that the defect had occurred across quadrants 13 and 14 is confirmed. The assumption that a larger portion of the defect is contained in quadrant 13, and only a very small portion of the defect is contained in quadrant 14. This type of defect could also have been caused by a clump of powder or debris that was stuck in the re-coater and caused this defect to have formed. This defect was considered as a non-critical defect as the defect was repaired by a subsequent re-coating operation as displayed in Figure 82.

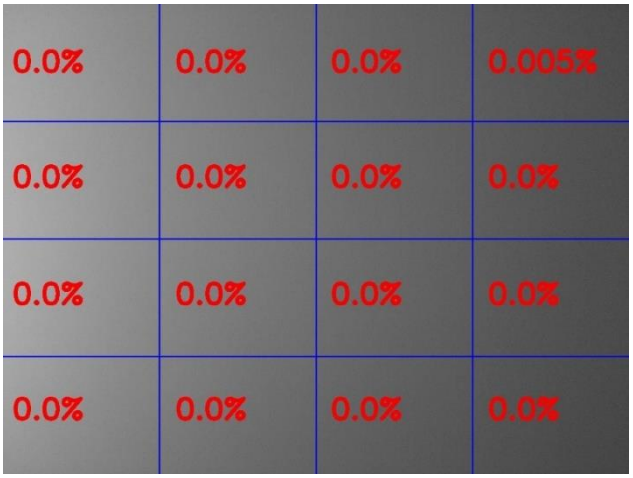



**Figure 82 Processed image for layer no. 287**

The seventh spike that has been highlighted occurred on layers 807 and 814 as demonstrated in Figure 83. There were two spikes on the graph that had occurred close to each other. However, they are only single spikes, indicating that a defect had occurred only on a single layer.

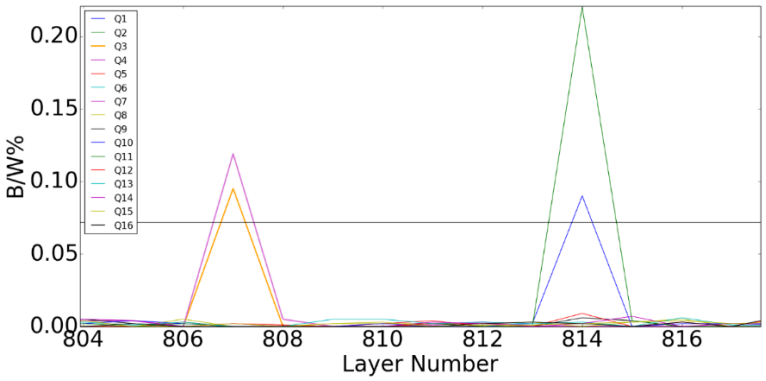



**Figure 83 Graph of peak no. 7**

The first spike at layer no 807 shows that a defect had occurred in quadrants 3 and 4 and the second spike on the graph had occurred at layer 814 in quadrants 10 and 11. When examining the graph in Figure 83, the defects have been covered over with subsequent re-coating operations. This means that the defects were singular defects that may have been caused by some obstruction in the re-coater or debris that fell onto the powder bed. To verify what had indeed happened on the powder bed, it is important to first review the images captured of the powder bed for the detected layers. The processed image displayed in Figure 84 shows the image captured of layer no 807.

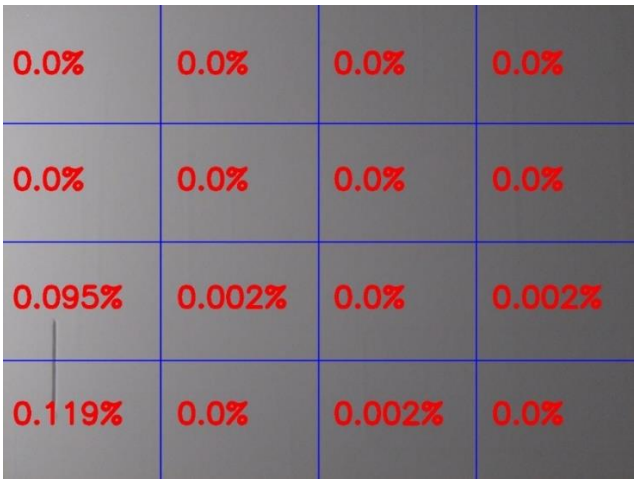



**Figure 84 Processed image for layer no. 807**

When examining the processed image captured for layer no. 807, a narrow defect has occurred on the powder bed surface, and that the defect had occurred in the direction of re-coating. As was identified from the first spike in the graph in Figure 83, the defect had occurred in quadrants 3 and 4.

However, the smaller portion of the defect that spread into quadrant 3 was still detected. As with the previous defect, this defect could have been caused by a clump of powder or debris that was stuck in the re-coater. The second processed image that was captured for layer no. 814 is displayed in Figure 85.

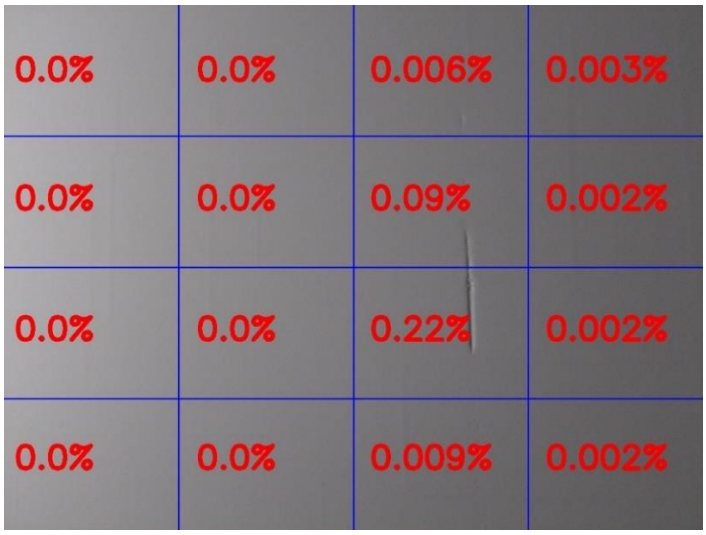



**Figure 85 Processed image for layer no. 814**

When examining the processed image in Figure 85, it can be seen that a defect occurred in quadrants 10 and 11. When looking at the graph in Figure 83, the second spike on the graph coincides with the quadrants that display the defect in the image in Figure 85. When looking at the percentage values of quadrants 10 and 11, quadrant 10 which covers the smaller part of the defect did not record as high a percentage value as quadrant 11. However, for quadrant 11 that covers the larger portion of the defect, a percentage value that is much greater than the threshold value was recorded.

The possible cause for this defect could also have been a clump of powder or debris stuck in the re-coater. However, because both of these defects had been repaired by the subsequent re-coating operations, these types of defects are considered non-critical defects.

The eighth highlighted spike on the graph in Figure 67 occurred on layer no 1085. The spike that had occurred on the graph has been magnified in Figure 86.

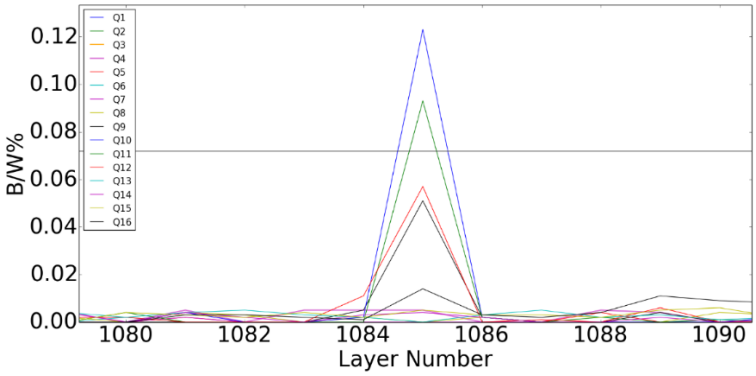



**Figure 86 Graph of peak no. 8**

When looking at the graph in Figure 86, there are four quadrants that had recorded a spike in the black to white pixel percentage values. After consulting the legend, the four different colours of the spikes in the graphs were identified as quadrants 9, 10, 11 and 12. This once again means that the defect that occurred on the powder bed was a form of a re-coating error, as the defect had occurred in the direction of re-coating. To confirm the nature of the re-coating error, it is necessary to consult the image that was captured by the camera for this layer on the powder bed. The processed image captured for layer no. 1085 is displayed in Figure 87.

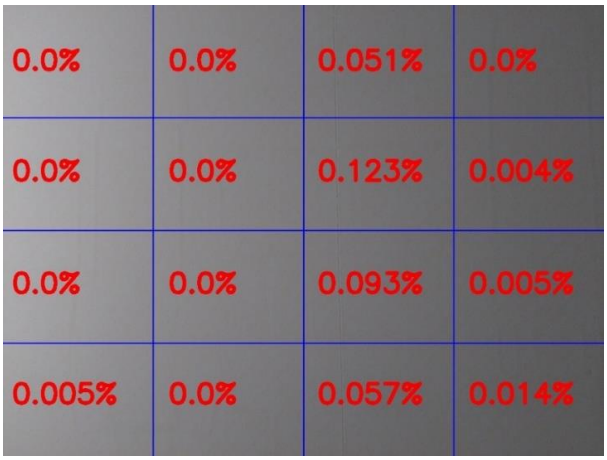



**Figure 87 Processed image for layer no. 1085**

When examining the processed image in Figure 87, a very small defect can be seen that spans the entire length of the powder bed. As was seen from the graph in Figure

86, quadrants 9, 10, 11 and 12 recorded the defect. This can be confirmed when looking at the processed image in Figure 87.

However, it is very clear that the defect that had occurred that is a very fine and shallow. Although only quadrants 10 and 11 recorded values higher than the threshold value, quadrants 9 and 12 still recorded a definite increase in percentage value. Still, the system proved its ability to detect the defect, even though it was not very clear or well-defined contrast wise.

The last spike on the graph in Figure 67 that will be looked at is the large area of the graph highlighted as no 6. The highlighted section of the graph as displayed in Figure 88 has been isolated for easier analysis.

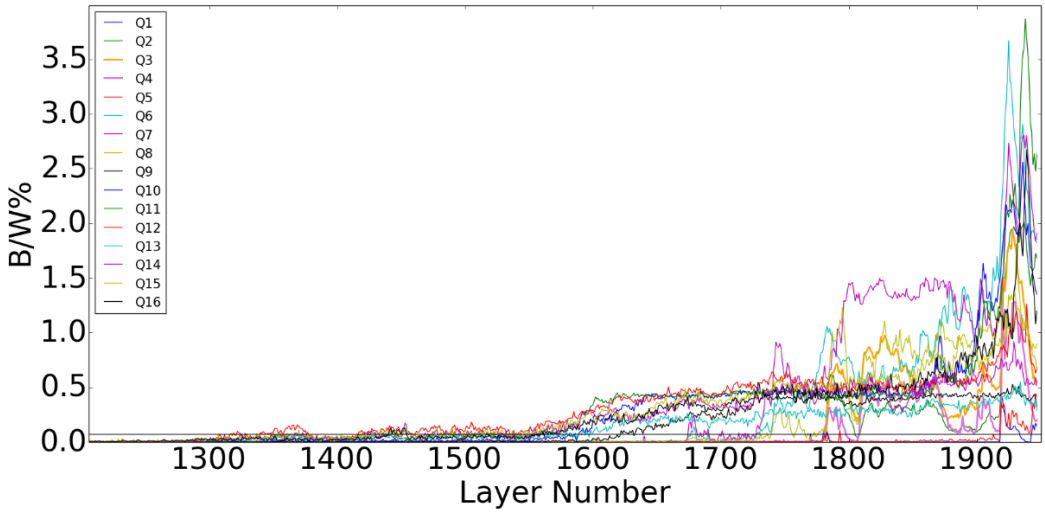



**Figure 88 Graph for the highlighted area no. 6**

When looking at the portion of the graph displayed in Figure 88, there is a definite trend that can be identified across all the different quadrants. From layer no. 1300, a definite upward trend of the black to white pixel percentage value can be seen. The trend slightly declined at layer no. 1380, and then shortly after layer no. 1400, the graph starts trending upwards, with the graph growing exponentially in the last approximately 100 layers of the build. When looking at the graph, a re-coating error occurring on the powder bed surface, and as the build progressed, the condition of the powder bed deteriorated with each subsequent re-coating operation. To determine what had happened on the powder bed, all the images that was captured, starting from

layer no. 1300, had to be reviewed until the last image of the build. Since it would be impractical to display all the captured images for these layers in this document, the images for the entire build will be included on external media together with this document. In Figure 89, several images were selected from the last 600 layers of the build to evaluate and showcase the problem. Looking at the first two images of layers no. 1400 and 1500, the defects on the powder bed appear to be minor and very small. However, when looking at the graph, the defect did initially not exceed the threshold value, which means that the system did not record the change on the powder bed as a defect. This could in part be linked to the fact that the defect did not create a big enough contrast difference for the system to detect. An image after every 100 re-coating cycles is shown to demonstrate intensification of the defects due to the re-coating error. Finally, when looking at layers 1800 and 1900, the re-coater starts showing signs of shortfeeding. This problem could possibly be linked to powder with a low flowability. The following images is displayed without the overlaid data so that the defects can be clearly seen on the image as the defects gradually increase in size and severity.

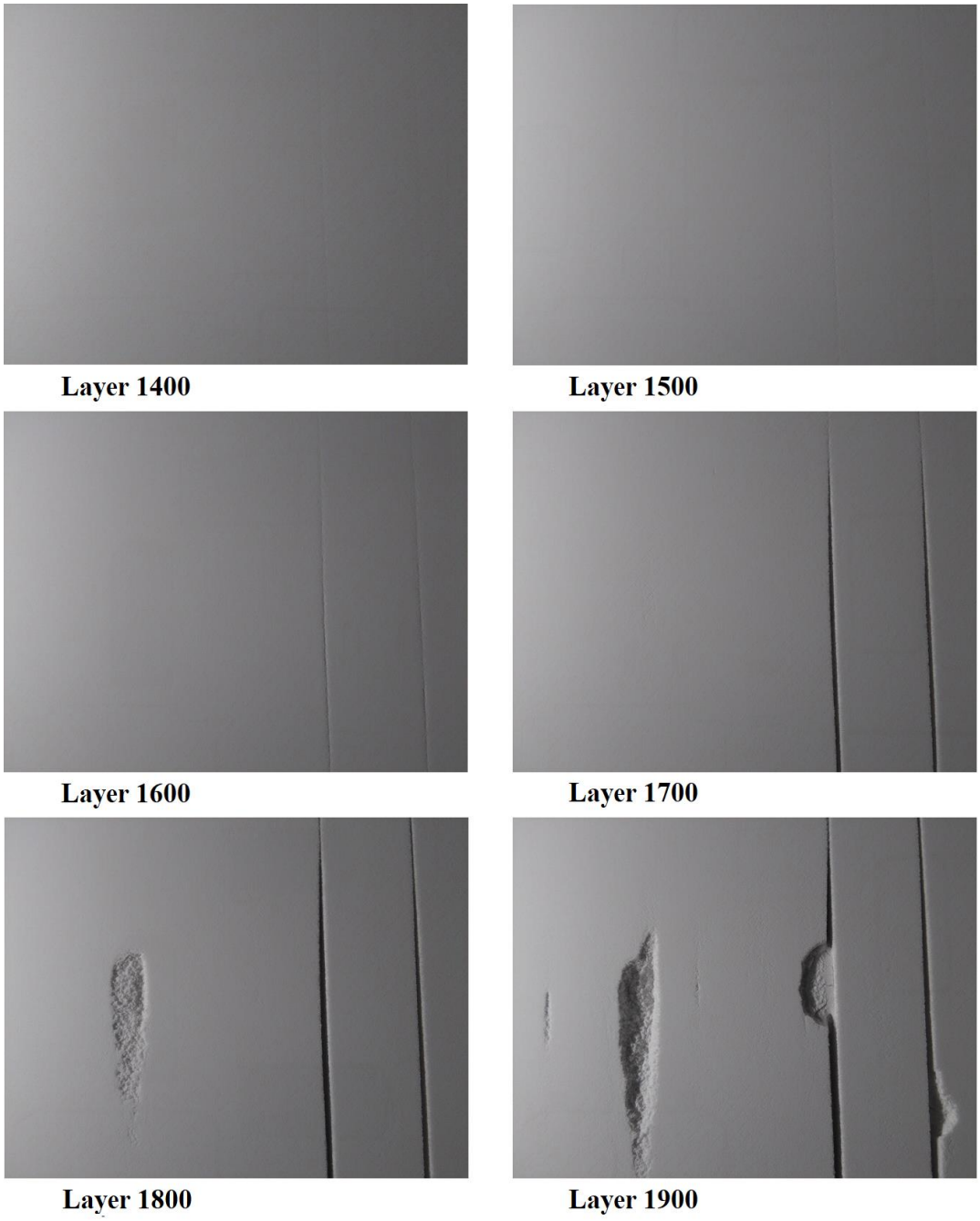



**Figure 89 Powder bed deterioration**

The damage to the part that is being built has already occurred since the re-coating error has not been cleared after more than 600 layers. This demonstrates why the re-coating monitoring system was developed, as errors like these can become very costly when they occur. When examining the final layer of the build as displayed in Figure

90, it is very clear that the surface of the powder bed has deteriorated to the point where the some areas of the powder bed have been severely short fed by the re-coater.

**Figure 90 Last layer of the build**

When referring to the graph in Figure 88, the black to white pixel percentage values have started to increase almost exponentially due to the re-coating errors growing in number and size on the powder bed. When referring to the processed image of the final layer of the build as shown in Figure 91, it could be seen that all the defects currently present on the powder bed had successfully been detected.

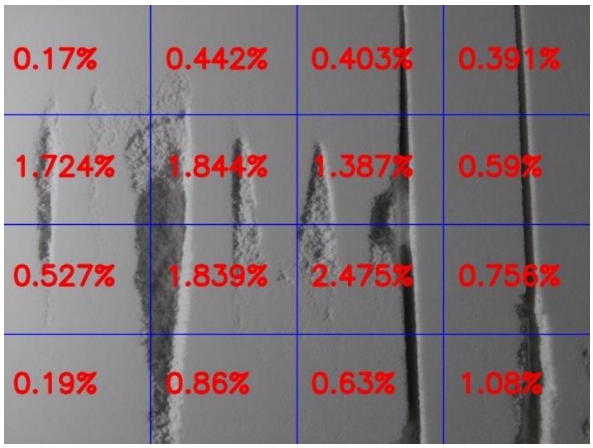



**Figure 91 Processed image of the final layer**

During the build that was used for this case study, several re-coating problems and re-coating errors had occurred which resulted in the scrapping of the entire build. However, it did provide the opportunity to test the designed re-coater monitoring system under a variety of conditions that occurs in the machine during normal

operation. As referred to in the beginning of this section, one of the purposes of the case study was to determine whether the selected parameters of the Gaussian blur filter kernel size could produce large enough black to white pixel percentage values when features were detected on the powder bed. As discussed in the previous section, a threshold value was also selected based on the results achieved with the replicated defects. However, both of these values needed to be verified by testing it during production. As could be seen throughout this case study, most of the defects that occurred on the powder bed were successfully detected. This gave a very good indication that the selected parameters were sufficient for the purpose of detecting defects. The system recorded one initial false positive result, but since this was a single occurrence in the entire build, this data was treated as a statistical anomaly as adjustment of the threshold value could not have prevented this from occurring.

Lastly, upon final review of all the images that was captured by the system for all the layers, several minor defects were discovered. When looking at the recorded data for these layers, it was discovered that the system had recorded an increase in the black to white pixel percentage values for these defects, however the values were below the threshold and as a result where not considered critical. A few of these defects are displayed in Figure 92.

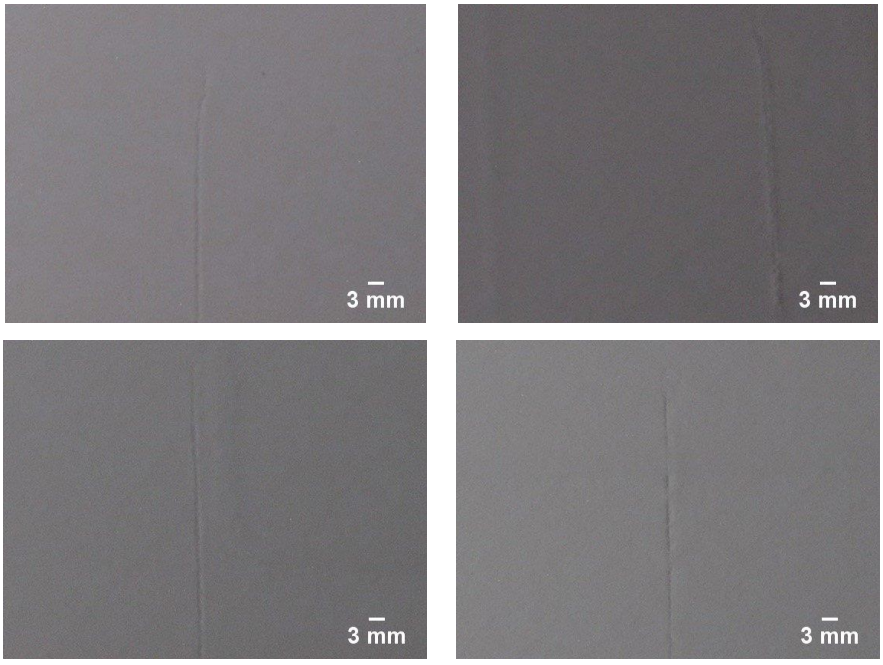



**Figure 92 Minor defects**

Upon closer inspection of these defects, it was discovered that several of these defects where extremely fine defects and in some cases, were not even visible unless the image was examined at 100%. Yet the system still did record an increase in value for the quadrants where the defects were located in. This proves that the system has the capability to detect even very small minor defects, as some of these defects were barely 1.38mm wide.

Overall, it be concluded that the system could successfully identify defects and re-coating errors when they occur on the powder bed during an actual build.

## 4.6 Summary

In this Chapter, the proposed active re-coater monitoring system was integrated into the Voxeljet VX500. The system was then tested using replicated defects, to verify the selected parameters of the various image processing functions, and also to determine if the system could indeed detect re-coating errors and defects. The Gaussian Blur kernel size was selected at a value of 2 and using this value, the active re-coater monitoring system successfully managed to detect all the replicated defects, including the re-coater short feeding defect.

Once the initial testing was completed, a threshold value that determines whether a detected feature is a defect or not was determined. This threshold was selected at a value of 0.072%. This threshold value was key in the testing of the system to detect defects on the powder bed during production conditions.

A case study of a build job was then conducted to test and determine if the selected parameters for the software program and threshold value produced acceptable results under production conditions. The case study was also used to determine if the system can indeed detect re-coating errors and defects during the actual build process.

Three other case studies were also conducted on different builds to validate the overall effectiveness of the system. Although some of the builds did not complete successfully, as was the case with the case study discussed in the previous section, they did provide valuable data as to the different types of defects as well as the possible

reasons for the defects. All the data that was collected has been combined in Table 7 for easier examination.

**Table 7 Case study data**

| | Case study 2 | Case study 3 | Case study 4 |
|---|---|---|---|
| **Defects @ 0.072%** | 3 | 6 | 36 |
| **Kernel size** | 2 | 2 | 2 |
| **No of layers** | 1032 | 993 | 1300 |
| **Build successful** | Yes | Yes | Yes |
| **Types of defects** | • Shallow lines<br>• Debris on powder bed | • Deep lines<br>• Shallow lines | • Powder clumps<br>• Shallow lines<br>• Underfeeding |

In conclusion, during the case studies, it was discovered that the designed active re-coater monitoring could successfully detect re-coating errors and defects as they occurred on the powder bed. Although one of the case studies had numerous defects that were detected, most of these defects were not critical and did not affect the success of the build. A possible reason for the larger number of defects present in this build job could have been due to low powder quality, thus causing lower quality re-coating to be performed.

In the next Chapter the conclusion of the research, recommendations for the study, as well as possible future research will also be discussed.

# Chapter 5 Conclusion and Recommendations

## 5.1 Introduction

The aim of this research was the development and building of an active re-coater monitoring system to detect re-coating errors during the printing process. In this chapter the intended objectives that were achieved will be discussed, as well as possible recommendations for future research.

## 5.2 Conclusions

### 5.2.1 Design of the active re-coater monitoring system

In order to develop a re-coater monitoring system, research had to be conducted in technologies that could be used to monitor the condition of the powder bed in a powder bed fusion AM machine. After comparing the advantages and disadvantages of the different types of technologies, computer vision was chosen as the most suitable technology.

The literature then covered topics such as sensor size, the focal length of the lens, camera working distance and the required lighting methods, illumination angles and fixtures. The specifications for the required computer vision camera was calculated mathematically at 7-megapixels, with a focal length of 3.036mm. The Raspberry Pi camera module was identified as a suitable camera module. The Raspberry Pi 3 Model B micro-computer was selected as the processing/computing platform for all image processing. In order to process the images captured by the camera, an image processing program was developed using the Python programming language and IDE. The OpenCV library was also integrated into the program to perform the image processing. Some of the image processing functions that was used to analyse the images includes Gaussian blur filters, Histogram equalization and Canny edge detection.

### 5.2.2 System verification

A test rig was built for benchmarking the capabilities of the camera. This was done to determine the smallest feature that could be identified using the selected camera

module. This test was conducted using the 1951 USAF resolution chart that was positioned at the same working distance that the camera would be required to operate on when installed inside the Voxeljet VX500 machine. During the testing phase, it was determined that the smallest feature that could still be clearly identified by the Raspberry Pi camera module was 198µm. Although the minimum feature that must be detectable by the camera was calculated at 184µm, the 14µm deficit was considered acceptable as there are several factors that influenced this value such as the exact working height, as well as the exact focal length of the camera.

### 5.2.3 Active Monitoring System Validation

The proposed active re-coater monitoring system was integrated into the Voxeljet VX500 machine for the testing of the image processing procedures. The purpose of these tests was to determine if the proposed image processing program can indeed detect defects and re-coating errors on the powder bed of an actual machine. A crucial part of these tests was also to optimise the software parameters that were used for the processing of the image. Several types of defects and re-coating errors were replicated to mimic real-world scenarios. The data recorded for each of the replicated scenarios was used to optimise the parameters for the different image processing functions to ensure that the defects were correctly detected. The image was split up into 16 quadrants, and each quadrant was processed for defects individually using a variety of image processing algorithms such as histogram equalization, Gaussian Blur filtering and Canny Edge detection. A value of 2 for the Gaussian Blur filter kernel size was determined as optimum for detecting the simulated defects. The black-to-white pixel threshold value that determines whether a defect had occurred on the powder bed was determined at a value of 0.072%. During this testing phase it was also determined that the active re-coater monitoring system could detect simulated re-coating errors and defects when they occurred on the powder bed.

Once the active re-coater monitoring system produced reliable results under the replicated error conditions, the in-production testing could commence. The active re-coater monitoring system was used to monitor a series of build jobs to determine the overall effectiveness of the system. All the build jobs were evaluated in detail to

determine the system's capability to detect defects as they occur, however, due to time and space constraints only the results from one build was discussed as part of this document. The recorded data was plotted into a graph, and significant spikes on the graphs were investigated and discussed. Once all the areas of interest on the graphs were discussed, some conclusions about the overall of effectiveness of the system could be reached. The active re-coater monitoring system displayed the ability to successfully detect a variety of re-coating errors and defects as they occurred in real-time. However, there was a re-coater underfeeding condition that started forming due to low powder flowability that was only detected once the defect became more severe. When the images and data for this section of the build was analysed, it could clearly be seen that the defect started out very small. Although the percentage values did record changes on the powder bed, they were not identified as defects due to the value being less than the threshold value of 0.072%. However, the case study did reveal that the active re-coater monitoring system could indeed detect various types of re-coating errors and defects under production conditions and was able to achieve most of the set-out objectives during the research study.

## 5.3 Recommendations

Further research should be conducted into:

- How machines with larger powder beds such as the Voxeljet VX1000 can be imaged using multiple cameras, as the powder bed is too large to be captured using a single camera due to the machine's internal construction.
- Adjustment of the camera FOV to image the entire powder bed surface.
- How to use different image processing techniques to identify defects on the powder bed, as well as to classify defects and re-coating errors that may occur on the powder bed.
- Implementing the feedback from the active re-coater monitoring system into the machine's control system. This would allow the machine to rectify the problems autonomously, or alternatively stop the build to prevent further loss of raw materials.

- The use of more powerful image processing hardware should be investigated as certain image processing function can quickly overwhelm the Raspberry Pi micro-computer’s available resources.

## References


ABBAS, G. 2013. *How to Build OpenCV 2.4.6 with GPU Module in Windows*. (Online). Available: http://www.programmerfish.com/how-to-build-opencv-2-4-6-with-gpu-module-in-windows/#.Wi6TETdx3IU. (Accessed: 05 November 2017).

ACCESSSCIENCE. 2015. *Computer Vision* (Online). Available: http://0-www.accessscience.com.library.vut.ac.za/search?q=computer+vision. (Accessed: 15 January 2017).

AKANDE, S.O., DALGARNO, K.W., MUNGUIA, J. and PALLARI, J. 2016. Assessment of tests for use in process and quality control systems for selective laser sintering of polyamide powders. *Journal of Materials Processing Technology,* 229:549-61.

ARDILA, L., GARCIANDIA, F., GONZÁLEZ-DÍAZ, J., ÁLVAREZ, P., ECHEVERRIA, A., PETITE, M., DEFFLEY, R. and OCHOA, J. 2014. Effect of IN718 Recycled Powder Reuse on Properties of Parts Manufactured by Means of Selective Laser Melting. *Physics Procedia,* 56:99-107.

ASGHARZADEH, H. and SIMCHI, A. 2005. Effect of sintering atmosphere and carbon content on the densification and microstructure of laser-sintered M2 high-speed steel powder. *Materials Science and Engineering,* 403(1-2):290-98.

BASLER ACE. 2016. *Accuracy of Basler TOF Camera.* (Online). Basler AG. Available: http://www.baslerweb.com/en/products/cameras/3d-cameras/time-of-flight-camera/tof640-20gm_850nm (Accessed: 26-07-2016).

BOSCHÉ, F. and GUENET, E. 2014. Automating surface flatness control using terrestrial laser scanning and building information models. *Automation in Construction,* 44:212-26.

BOUKOUVALAS, C., KITTLER, J., MARIK, R., MIRMEHDI, M. and PETROU, M. 1995. Ceramic tile inspection for colour and structural defects. *Proceedings of AMPT95* 1:390-99.

BRADSKI, G. 2017. *OpenCV: OpenCV modules*. (Online). Available: https://docs.opencv.org/3.3.1/index.html. (Accessed: 13 December 2017).

CALLIS, C. 2010. *Lesson 3 Light Falling on an Sphere*. (Online). Available: http://fall2010lightingclass.blogspot.co.za/2010/09/lesson-2-light-falling-on-sphere.html. (Accessed: 14 December 2017).

CHRISTOFFERSEN, J. 2015. *Switch Bounce and How to Deal with It*. (Online). Available: https://www.allaboutcircuits.com/technical-articles/switch-bounce-how-to-deal-with-it/. (Accessed: 18 December 2017).

COHOON, C. 2011. *How big can I print my photo on canvas?* (Online). Available: http://www.canvaspress.com/focal-point/article/2011/05/24/how-big-can-i-print-my-photo-on-canvas/. (Accessed: 22 May 2017).

COOKE, W., TOMLINSON, R.A., BURGUETE, R., JOHNS, D. and VANARD, G. 2011. Anisotropy, homogeneity and ageing in an SLS polymer. *Rapid Prototyping Journal*, 17(4):269-79.

CRAEGHS, T., CLIJSTERS, S., YASA, E. and KRUTH, J.-P. 2011. Online quality control of selective laser melting. *In 22nd Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference, SFF 2011, August 8, 2011 - August 10, 2011*. Austin, TX, United states. University of Texas at Austin (freeform), pp. 212-26.

CUSTOMPARTNET. 2016a. *DMLS - Direct Metal Laser Sintering*. (Online). Available: http://www.custompartnet.com/wu/direct-metal-laser-sintering. (Accessed: 20 May 2016).

CUSTOMPARTNET. 2016b. *Rapid Prototyping - Three Dimensional Printing*. (Online). Available: http://www.custompartnet.com/wu/3d-printing. (Accessed: 24 April 2016).

CZERANOWSKY, C. 2016. *What Lenses Are Available and How Do You Select the Right Lens for a Camera?* (Online). Ahrensburg, Germany: Basler AG. Available: https://www.baslerweb.com/en/support/downloads/document-downloads/#type=whitepapers;language=all (Accessed: 16 August 2017).

DAWSON-HOWE, K. 2014. *A Practical Introduction to Computer Vision With OpenCV*. 1 ed. West Sussex: John Wiley & Sons.

DINWIDDIE, R.B., DEHOFF, R.R., LLOYD, P.D., LOWE, L.E. and ULRICH, J.B. 2013. Thermographic in-situ process monitoring of the electron beam melting technology used in additive manufacturing. *In Thermosense: Thermal Infrared Applications XXXV, April 30, 2013 - May 1, 2013*. Baltimore, MD, United states. SPIE, pp. The Society of Photo-Optical Instrumentation Engineers (SPIE).

EBRAHIM, M.A.-B. 2015. 3D Laser Scanners' Techniques Overview. *International Journal of Science and Research*, 4(10):323-31.

EDMUND OPTICS. 21/03/2011. *EO Imaging Lab 1.6: Resolution In Depth* (Video online). Available: https://www.youtube.com/watch?v=iD0kzxU0fvY. (Accessed: 18 May 2017)

EDMUND OPTICS. 2017. *Choose the Correct Illumination*. (Online). Available: https://www.edmundoptics.com/resources/application-notes/illumination/choose-the-correct-illumination/. (Accessed: 29 May 2017).

ELM, C. 2011. *Area Camera and Line Camera*. (Online). Available: http://elm-chan.org/works/lcam/report.html. (Accessed: 20 June 2017).

ENGADGET. 2016. *Sony PlayStation Eye* (Online). Available: https://www.engadget.com/products/sony/playstation/eye/specs/. (Accessed: 27 October 2016).

EOS GMBH. 2017. *Laser sintering system EOSINT M 280 for the production of tooling inserts, prototype parts and end products directly in metal.* (Online). Munich: EOS GmbH. Available: https://www.eos.info/systems_solutions/metal/systems_equipment/eosint_m280 (Accessed: 24 July 2018).

EVERTON, S.K., HIRSCH, M., STRAVROULAKIS, P., LEACH, R.K. and CLARE, A.T. 2016. Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing. *Materials and Design*, 95:431-45.

FARO. 2016a. *FARO Cobalt Array Imager*. (Online). Available: http://www.faro.com/products/metrology/faro-cobalt-array-imager/overview. (Accessed: 26 December 2016).

FARO. 2016b. *FARO Laser Tracker*. (Online). Available: http://www.faro.com/products/metrology/faro-laser-tracker/overview#main. (Accessed: 07 December 2016).

FISHER, R., PERKINS, S., WALKER, A., WOLFART, E. 2003. *Hypermedia Image Processing Reference.* (Online). West Sussex: John Wiley & Sons. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/copyrght.htm (Accessed: 06 June 2017).

FOIX, S., ALENYA, G. and TORRAS, C. 2011. Lock-in time-of-flight (ToF) cameras: a survey. *IEEE Sensors Journal*, 11(9):1917-26.

FRAM-SCHWARTZ, N. 2016. *Types of additive manufacturing*. (Online). Available: http://noahfs.com/#3DP. (Accessed: 17 March 2017).

FU, Z., SCHLIER, L., TRAVITZKY, N. and GREIL, P. 2013. Three-dimensional printing of SiSiC lattice truss structures. *Materials Science and Engineering A,* 560:851-56.

GIBSON, I., ROSEN, D. and STUCKER, B. 2014. *Additive Manufacturing Technologies.* 2 ed. New York: Springer-Verlag.

GONZALEZ, J.A., MIRELES, J., LIN, Y. and WICKER, R.B. 2016. Characterization of ceramic components fabricated using binder jetting additive manufacturing technology. *Ceramics International,* 42(9):10559-64.

ICDSECURITY. 2012. *Security in 3D: a closer look at "Time of Flight" technology*. (Online). Available: www.icdsecurity.com/2013/08/30/security-in-3d-a-closer-look-at-time-of-flight-technology/. (Accessed: 13 December 2016).

KARPATHY, A. and LI, F.-F. 2015. Deep visual-semantic alignments for generating image descriptions. *In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, June 7, 2015 - June 12, 2015*. Boston, MA, United states. IEEE Computer Society, pp. 3128-37.

KNIGHTS OPTICAL. 2014. *Custom resolution charts*. (Online). Available: https://www.knightoptical.com/custom/?category=/optical-components/uvvisnir-optics/resolution-charts/. (Accessed: 05 May 2018).

LASERDESIGN. 2016. *What is 3D scanning?* (Online). Available: http://www.laserdesign.com/what-is-3d-scanning. (Accessed: 14 October 2016).

LEFLOCH, D., NAIR, R., LENZEN, F., SCHÄFER, H., STREETER, L., CREE, M.J., KOCH, R. and KOLB, A. 2013. *Technical Foundation and Calibration Methods for Time-of-Flight Cameras*. *In* Grzegorzek, M., Theobalt, C., Koch, R. & Kolb, A., eds. *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications: Dagstuhl 2012 Seminar on Time-of-Flight Imaging and GCPR 2013 Workshop on Imaging New Modalities.* Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 3-24.

LI, L.-J., SOCHER, R. and LI, F.-F. 2009. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework.

LI, L. 2014. *Time-of-Flight Camera – An Introduction.* (Online). Dallas: Texas Instruments. Available: http://www.ti.com/lit/wp/sloa190b/sloa190b.pdf (Accessed: 30 August 2017).

LIN, K.-L. and FANG, J.-L. 2013. Applications of computer vision on tile alignment inspection. *Automation in Construction,* 35:562-67.

MALLICK, S. 2015. *Why does OpenCV use BGR color format ?* (Online). Available: https://www.learnopencv.com/why-does-opencv-use-bgr-color-format/. (Accessed: 11 December 2017).

MAUCHLINE, D. 2016. VUT supports SA foundry industry through additive manufacturing technology. *VUT E-News*, 08 July 2016: 3.

MIPI ALLIANCE. 2017. *Evolving CSI-2 Specification.* (Online). Piscataway: Mipi Alliance. Available: https://www.mipi.org/specifications/csi-2 (Accessed: 06 December 2017).

NATIONAL INSTRUMENTS. 2014. *Calculating Camera Sensor Resolution and Lens Focal Length*. (Online). Available: http://digital.ni.com/public.nsf/allkb/1BD65CB07933DE0186258087006FEBEA. (Accessed: 21 May 2017).

OLSEN, M.J., KUESTER, F., CHANG, B.J. and HUTCHINSON, T.C. 2010. Terrestrial Laser Scanning-Based Structural Damage Assessment. *Journal of Computing in Civil Engineering*, 24(3):264-72.

OPENCV. 2017. *About OpenCV*. (Online). Available: https://opencv.org/about.html. (Accessed: 11 December 2017).

PYSER-SGI LTD. 2010. *RESOLUTION CHARTS AND GRATINGS.* (Online). Kent: Pyser-SGI Ltd. Available: https://www.molenaar-optics.nl/graticules_bestanden/Resolution%20Charts.pdf (Accessed: 07 July 2017).

RASPBERRY PI FOUNDATION. 2016a. *Camera Module*. (Online). Available: https://github.com/raspberrypi/documentation/blob/master/hardware/camera/README.md. (Accessed: 01 June 2017).

RASPBERRY PI FOUNDATION. 2016b. *Raspberry Pi 3 Model B.* (Online). Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. (Accessed: 01 June 2017).

RASPBERRY PI FOUNDATION. 2017. *Raspbian*. (Online). Available: https://www.raspberrypi.org/downloads/raspbian/. (Accessed: 11 June 2017).

ROBINSON, G. and HARDIN, M. 2015. An Overview of Modern Metrology to Support Military Testing. *DSIAC*, 2(3):36.

ROBOTICLAB. 2013. *Lidar*. (Online). Available: http://home.roboticlab.eu/en/examples/sensor/lidar. (Accessed: 11 June 2017).

ROSEBROCK, A. 2015. *Zero-parameter, automatic Canny edge detection with Python and OpenCV*. (Online). Available: https://www.pyimagesearch.com/2015/04/06/zero-parameter-automatic-canny-edge-detection-with-python-and-opencv/. (Accessed: 19 March 2018).

ROSEBROCK, A. 2017. *Practical Python and OpenCV + Case Studies.* 3rd ed. Baltimore: PyImageSearch.

SACHS, E., CIMA, M., WILLIAMS, P., BRANCAZIO, D. and CORNIE, J. 1992. Three dimensional printing: Rapid Tooling and prototypes directly from a CAD model. *Journal of engineering for industry*, 114(4):481-88.

SHABANGA, Y. (vonnytricia@gmail.com). 2018. *RE: Re: Build Box Parameters.* Email received: 27 August 2018.

SHANMUGAMANI, R., SADIQUE, M. and RAMAMOORTHY, B. 2015. Detection and classification of surface defects of gun barrels using computer vision and machine learning. *Measurement: Journal of the International Measurement Confederation*, 60:222-30.

STEMMER IMAGING. 2017. *Schematic structure of a line-scan camera application with potential error locations indicated.* (Online). Available: http://www.stemmer-imaging.co.uk/en/technical-tips/line-scan-cameras/#. (Accessed: 13 July 2017).

TELEDYNE DALSA. 2014. *Understanding Line Scan Camera Applications.* (Online). Waterloo: Teledyne Dalsa. Available: http://leadwise.mediadroit.com/files/292072069_TD.LineScanApp_whitepaper.v4.pdf (Accessed: 06 July 2017).

TUTORIALS POINT. 2014. *Histogram Equalization*. (Online). Available: http://www.tutorialspoint.com/dip/Histogram_Equalization.htm. (Accessed: 05 May 2018).

VOXELJET. 2015. *VX1000 - The universal 3D printer*. (Online). Friedberg: Voxeljet Services. Available: http://www.voxeljet.de/en/systems/3d-druckervx10000/ (Accessed: 17 June 2017).

VOXELJET. 2017. *VX500 Brochure.* (Online). Friedberg: Voxeljet Services. Available: http://www.voxeljet.com/3d-drucksysteme/vx1000/ (Accessed: 25 June 2016).

WANG, Q., KIM, M.-K., CHENG, J.C.P. and SOHN, H. 2016. Automated quality assessment of precast concrete elements with geometry irregularities using terrestrial laser scanning. *Automation in Construction*, 68:170-82.

WANG, Y. and FENG, H.-Y. 2014. Modeling outlier formation in scanning reflective surfaces using a laser stripe scanner. *Measurement: Journal of the International Measurement Confederation,* 57:108-21.

WANG, Y. and FENG, H.-Y. 2016. Effects of scanning orientation on outlier formation in 3D laser scanning of reflective surfaces. *Optics and Lasers in Engineering,* 81:35-45.

Z-CORP. 2009. *Portable 3D Laser Scanner from Z Corporation*. (Online). Available: http://www.makepartsfast.com/portable-3d-laser-scanner-from-z-corporation/. (Accessed: 14 July 2017).

ZHANG, B., ZIEGERT, J., FARAHI, F. and DAVIES, A. 2016. In situ surface topography of laser powder bed fusion using fringe projection. *Additive Manufacturing,* 12(A):100-07.

ZHUWAKINYU, M. 2012. New Vaal Technology Park aims to support entrepreneurs. *Engineering News.*, 15 June 2012: 1.

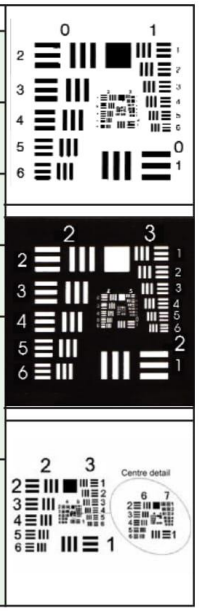
# Annexure A:

## USAF 1951 Test Chart

USAF resolution charts are recognised the world over as a universal standard for testing the vertical and horizontal resolution of imaging systems. Each element on the chart comprises three vertical bars and three horizontal bars, and the detail on these slides is as fine as 0.78microns (644 line-pairs per mm). The resolution of the imaging system is normally specified as the Group and Element of the finest bars that can be clearly defined - See further information on back page of brochure.

The six versions manufactured by Pyser are as follows:

| Pattern | Description | Order Code |
|---|---|---|
| R70 | USAF Test Chart, positive image. Group 0, element 1 to Group 7, element 6. B270 glass, size 50mm x 50mm | 06B01096 |
| R71 | USAF Test Chart, positive image. Group -2, element 1 to Group 7, element 6. B270 glass, size 75mm x 75mm | 06B01097 |
| R75P | USAF Resolution Chart, positive version. Group 0, element 1 to group 9, element 3. Soda lime glass size 50mm x 50mm | 06B01102 |
| R75N | USAF Resolution Chart, negative version. Group 0, element 1 to group 9, element 3. Soda lime glass size 50mm x 50mm | 06B01103 |
| PS75P | USAF Resolution Chart, positive version. Group 2, element 1 to Group 9, element 3. Soda lime glass mounted in stainless steel microscope slide, with engraved serial number for traceability, 76mm x 25mm. Supplied in polished wood case. | 05B01090 |
| PS75N | USAF Resolution Chart, negative version. Group 2, element 1 to Group 9, element 3. Soda lime glass mounted in stainless steel microscope slide with engraved serial number for traceability, 76mm x 25mm. Supplied in polished wood case | 05B01091 |




We are also able to make custom versions of the USAF Test Chart which are often required for special applications, and can supply them with Internationally traceable certificates of calibration.

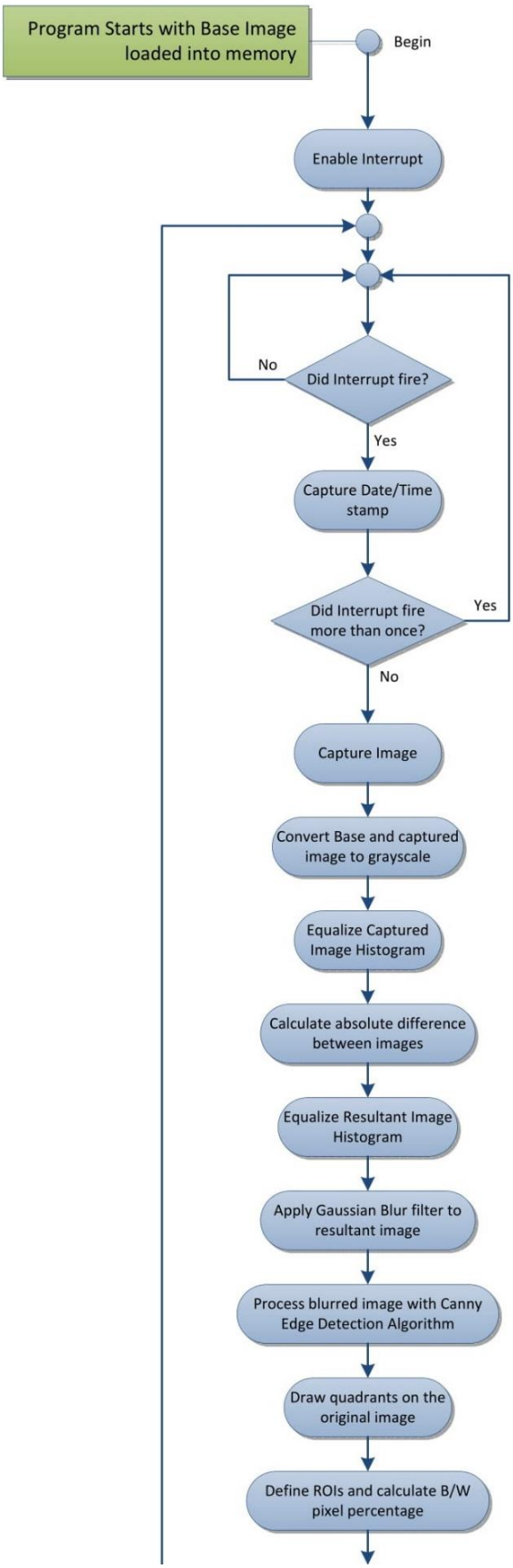
**USAF Resolution Chart, Number of Lines per mm**

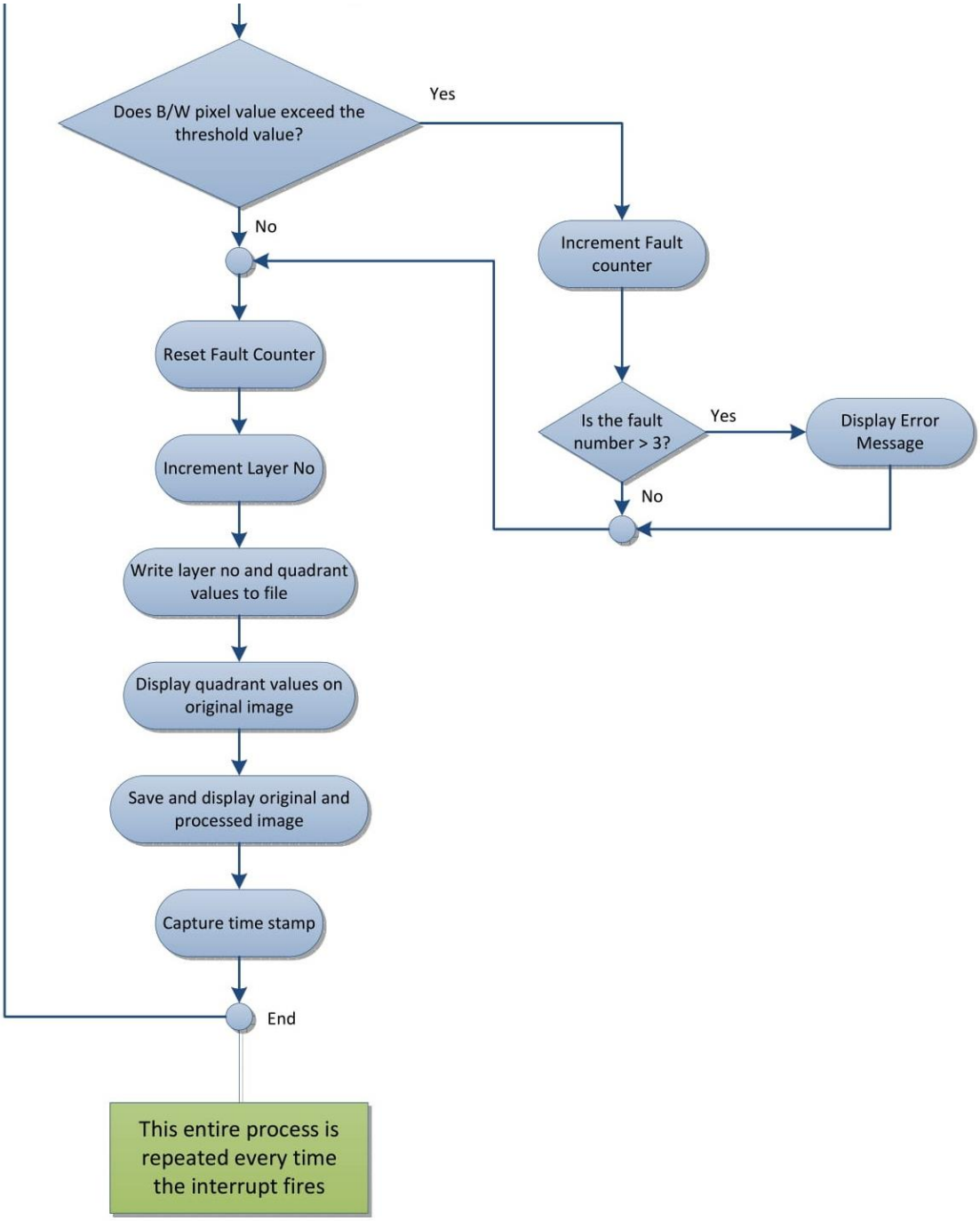
| *Element Number* | *Group Number* | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 0.250 | 0.500 | 1.00 | 2.00 | 4.00 | 8.00 | 16.00 | 32.00 | 64.0 | 128 | 256 | 512 |
| 2 | 0.280 | 0.561 | 1.12 | 2.24 | 4.49 | 8.89 | 17.95 | 36.00 | 71.8 | 144 | 288 | 576 |
| 3 | 0.315 | 0.630 | 1.26 | 2.52 | 5.04 | 10.1 | 20.16 | 40.30 | 80.6 | 161 | 322 | 644 |
| 4 | 0.353 | 0.707 | 1.41 | 2.83 | 5.66 | 11.3 | 22.62 | 45.30 | 90.5 | 181 | 362 | |
| 5 | 0.397 | 0.793 | 1.59 | 3.17 | 6.35 | 12.7 | 25.39 | 50.80 | 102.0 | 203 | 406 | |
| 6 | 0.445 | 0.891 | 1.78 | 3.56 | 7.13 | 14.3 | 28.51 | 57.00 | 114.0 | 228 | 456 | |

**USAF Resolution Chart, Width of Bars in mm**

| *Element Number* | *Group Number* | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 2.00 | 1.00 | 0.500 | 0.250 | 0.125 | 0.063 | 0.031 | 0.016 | 0.0078 | 0.0039 | 0.0020 | 0.00098 |
| 2 | 1.79 | 0.89 | 0.446 | 0.223 | 0.111 | 0.056 | 0.028 | 0.014 | 0.0070 | 0.0035 | 0.0017 | 0.00087 |
| 3 | 1.59 | 0.79 | 0.396 | 0.198 | 0.099 | 0.050 | 0.025 | 0.012 | 0.0062 | 0.0031 | 0.0016 | 0.00078 |
| 4 | 1.42 | 0.71 | 0.355 | 0.177 | 0.088 | 0.044 | 0.022 | 0.011 | 0.0055 | 0.0028 | 0.0014 | |
| 5 | 1.26 | 0.63 | 0.315 | 0.158 | 0.079 | 0.039 | 0.020 | 0.0098 | 0.0049 | 0.0025 | 0.0012 | |
| 6 | 1.12 | 0.56 | 0.281 | 0.140 | 0.070 | 0.035 | 0.018 | 0.0088 | 0.0035 | 0.0022 | 0.0011 | |

## Annexure B:


Program Starts with Base Image loaded into memory
Begin
Enable Interrupt
No
Did Interrupt fire?
Yes
Capture Date/Time stamp
Did Interrupt fire more than once?
Yes
No
Capture Image
Convert Base and captured image to grayscale
Equalize Captured Image Histogram
Calculate absolute difference between images
Equalize Resultant Image Histogram
Apply Gaussian Blur filter to resultant image
Process blurred image with Canny Edge Detection Algorithm
Draw quadrants on the original image
Define ROIs and calculate B/W pixel percentage


Does B/W pixel value exceed the threshold value?
Yes
No
Increment Fault counter
Reset Fault Counter
Is the fault number > 3?
Yes
Display Error Message
No
Increment Layer No
Write layer no and quadrant values to file
Display quadrant values on original image
Save and display original and processed image
Capture time stamp
End
This entire process is repeated every time the interrupt fires

## Annexure C:

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
from decimal import Decimal
import numpy
import os
import RPi.GPIO as GPIO
import PIL.Image
import PIL.ImageTk
from Tkinter import *

#Setup the GPIO Connections for the Limit Switches
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down = GPIO.PUD_UP)

start_time = time.time()

layerVal = 0
faultVal = 0
timestr = time.strftime("%Y%m%d-%H%M%S")
file = open("Job_"+timestr+".txt","w")
img1 = cv2.imread('/home/pi/Raspberry Pi Control/VXoriginalLED1.jpg')

#Initialize Camera model
print("Intializing Camera")
camera = PiCamera()
camera.resolution = (3264,2448)
print("Camera Resolution set to 3264 x 2448 pixels")
rawCapture = PiRGBArray(camera)
time.sleep(0.1) #Give the Camera time to stablize

#Create the GUI Window
top = Tk()
top.geometry("1280x720")
top.wm_title("Re-coater Monitoring System")
```

```
#Import Processed Image
procImg = cv2.imread('/home/pi/OpenCV_Logo.png')
resPImg = cv2.resize(procImg, (600,440))
imgP = cv2.cvtColor(resPImg, cv2.COLOR_RGB2BGR)
imP = PIL.Image.fromarray(imgP)
imgtkP = PIL.ImageTk.PhotoImage(imP)

#Function Definitions
def changeLabel():
    lblConnectionStatus.config(bg="green")
    connectionStatus.set("Connected")
    btnExit.configure(state = DISABLED)
    btnStartCapture.configure(state = NORMAL)
    btnStopCapture.configure(state = NORMAL)

def auto_canny(image, sigma=0.33):
    # compute the median of the single channel pixel intensities
    v = numpy.median(image)
    # apply automatic Canny edge detection using the computed median
    lower = int(max(0, (1.0 - sigma) * v))
    upper = int(min(255, (1.0 + sigma) * v))
    edged = cv2.Canny(image, lower, upper)
    # return the edged image
    return edged

def pROI(x1,y1,x2,y2,image):
    roi = image[x1:y1,x2:y2]
    whiteval = cv2.countNonZero(roi)
    noPixels = (Decimal(whiteval)/Decimal(1088*816))*100
    roundPix = round(noPixels, 3)
    val = str(roundPix)
    return val

def captureImage(self):
    global start_time
    global layerVal
    global faultVal
    global img1
```

```
    time.sleep(0.01)
    if GPIO.input(23) != GPIO.HIGH:
        print("False Trigger")
        return
    else:
        #Check the trigger time
        ti = round(time.time()-start_time)
        #Check if the trigger was fired more than once in a 3 second window
        if ti > 3:
            #Capture the image to an array
            camera.capture(rawCapture, 'bgr')
            rawCapture.truncate(0)
            image = rawCapture.array
            print("Image Captured")

            #Convert the baseline image to greyscale and equalize the histogram
            gray1 = cv2.cvtColor(img1,cv2.COLOR_BGR2GRAY)
            cv2.equalizeHist(gray1,gray1)
            #Convert the captured image to greyscale for further processing and equalize the histogram
            img2 = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            cv2.equalizeHist(img2,img2)
            #Create empty numpy array to store the processed images in
            res = numpy.zeros((2448,3264), numpy.uint8)
            #Calculate the absolute difference between the captured image and the baseline image and equalize
the histogram again
            cv2.absdiff(gray1,img2,res)
            cv2.equalizeHist(res,res)
            #Apply a Gaussian blur filter to the image to smooth out the worst noise
            blur2 = cv2.GaussianBlur(res,(11,11),2)
            #Apply the canny edge detection algorithm to the blurred image
            edges = auto_canny(blur2)

            #colImg = cv2.cvtColor(img2,cv2.COLOR_GRAY2BGR)
            #Draw the lines to indicate the 16 quadrants
            origImg = image
            cv2.line(origImg,(0,612),(3264,612),(255,0,0),5)
            cv2.line(origImg,(0,1224),(3264,1224),(255,0,0),5)
            cv2.line(origImg,(0,1836),(3264,1836),(255,0,0),5)
```

```
        cv2.line(origImg,(816,0),(816,2448),(255,0,0),5)
        cv2.line(origImg,(1632,0),(1632,2448),(255,0,0),5)
        cv2.line(origImg,(2448,0),(2448,2448),(255,0,0),5)
        #Defining the different regions of interest (ROI)
        #First Column
        rval1 = pROI(0,612,0,816,edges)
        rval2 = pROI(613,1224,0,816,edges)
        rval3 = pROI(1225,1836,0,816,edges)
        rval4 = pROI(1837,2448,0,816,edges)
        #Second Column
        rval5 = pROI(0,612,817,1632,edges)
        rval6 = pROI(613,1224,817,1632,edges)
        rval7 = pROI(1225,1836,817,1632,edges)
        rval8 = pROI(1837,2448,817,1632,edges)
        #Third Column
        rval9 = pROI(0,612,1633,2448,edges)
        rval10 = pROI(613,1224,1633,2448,edges)
        rval11 = pROI(1225,1836,1633,2448,edges)
        rval12 = pROI(1837,2448,1633,2448,edges)
        #Fourth Column
        rval13 = pROI(0,612,2449,3264,edges)
        rval14 = pROI(613,1224,2449,3264,edges)
        rval15 = pROI(1225,1836,2449,3264,edges)
        rval16 = pROI(1837,2448,2449,3264,edges)

        refVal = float(0.6)

        if(float(rval1)>refVal or float(rval2)>refVal or float(rval3)>refVal or float(rval4)>refVal or
float(rval5)>refVal
           or float(rval6)>refVal or float(rval7)>refVal or float(rval8)>refVal or float(rval9)>refVal or
float(rval10)>refVal
           or float(rval11)>refVal or float(rval12)>refVal or float(rval13)>refVal or float(rval14)>refVal or
float(rval15)>refVal or float(rval16)>refVal):
            #increment per time a fault is detected
            faultVal = faultVal + 1
            if(faultVal >= 3):
               #Display that an error has been detected
               print("Something Happened")
```

```
            var.set("Defect Detected")

        else:
            #Reset if fault has been cleared
            print("All Good")
            var.set("No Defects")
            faultVal = 0



        #Retrieve current date and time
        timestr = time.strftime("%Y%m%d-%H:%M:%S")
        #Increment the layer number value
        layerVal += 1
        #Write the Layer Number, time stamp and white pixel percentage of each quadrant to a text file

file.write(str(layerVal)+","+timestr+","+rval1+","+rval2+","+rval3+","+rval4+","+rval5+","+rval6+","+rv
al7+","+rval8

+","+rval8+","+rval9+","+rval10+","+rval11+","+rval12+","+rval13+","+rval14+","+rval15+","+rval16+"
\n")
        #Set the font of the text to be written
        font = cv2.FONT_HERSHEY_SIMPLEX
        #Display each quadrant's white pixel percentage
        #First Column
        cv2.putText(origImg,rval1+'%',(50,350), font, 6,(0,0,255),10)
        cv2.putText(origImg,rval2+'%',(50,950), font, 6,(0,0,255),10)
        cv2.putText(origImg,rval3+'%',(50,1560), font, 6,(0,0,255),10)
        cv2.putText(origImg,rval4+'%',(50,2160), font, 6,(0,0,255),10)
        #Second Column
        cv2.putText(origImg,rval5+'%',(900,350), font, 6,(0,0,255),10)
        cv2.putText(origImg,rval6+'%',(900,950), font, 6,(0,0,255),10)
        cv2.putText(origImg,rval7+'%',(900,1560), font, 6,(0,0,255),10)
        cv2.putText(origImg,rval8+'%',(900,2160), font, 6,(0,0,255),10)
        #Third Column
        cv2.putText(origImg,rval9+'%',(1710,350), font, 6,(0,0,255),10)
        cv2.putText(origImg,rval10+'%',(1710,950), font, 6,(0,0,255),10)
        cv2.putText(origImg,rval11+'%',(1710,1560), font, 6,(0,0,255),10)
        cv2.putText(origImg,rval12+'%',(1710,2160), font, 6,(0,0,255),10)
```

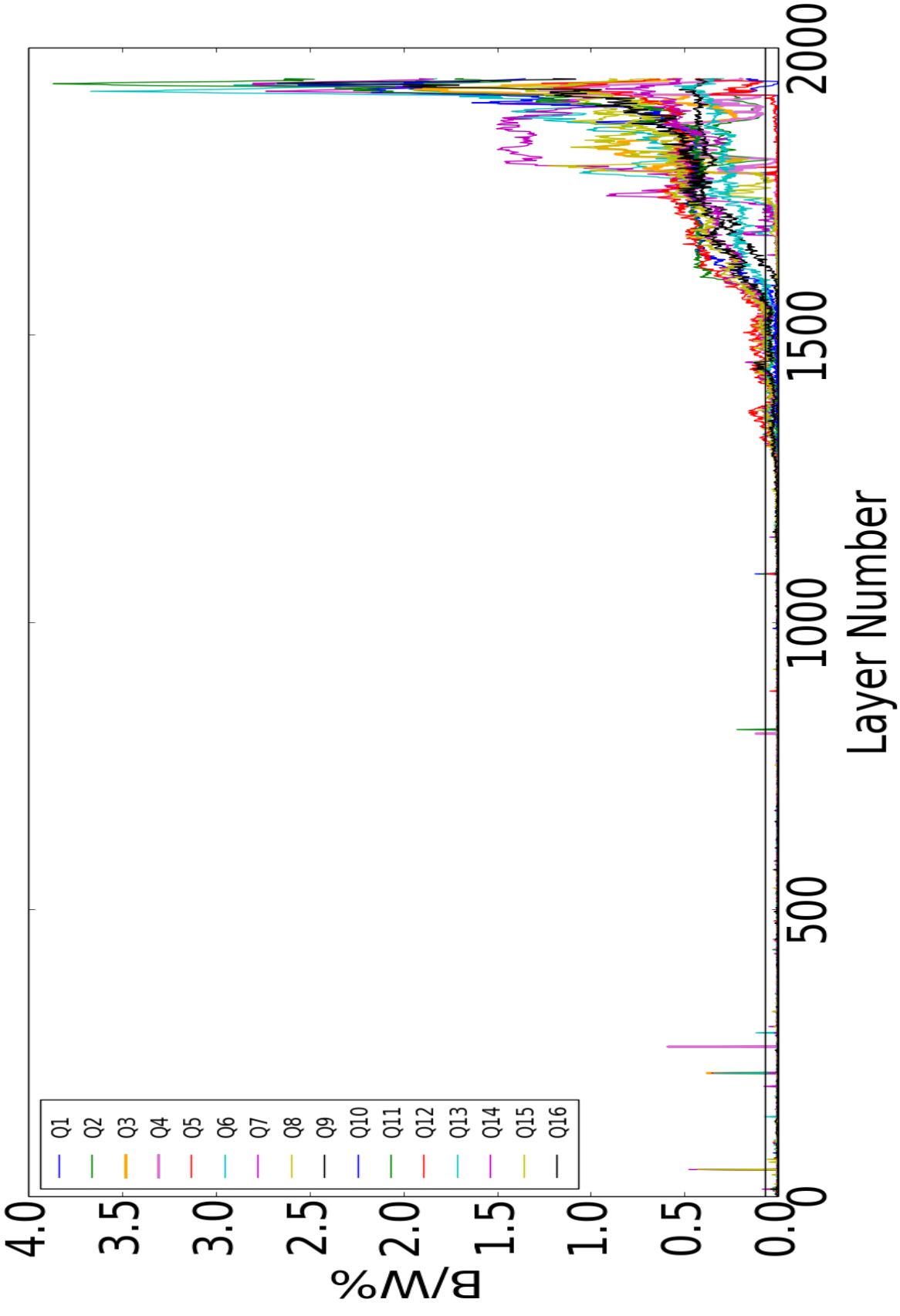
```
            #Fourth Column
            cv2.putText(origImg,rval13+'%',(2530,350), font, 6,(0,0,255),10)
            cv2.putText(origImg,rval14+'%',(2530,950), font, 6,(0,0,255),10)
            cv2.putText(origImg,rval15+'%',(2530,1560), font, 6,(0,0,255),10)
            cv2.putText(origImg,rval16+'%',(2530,2160), font, 6,(0,0,255),10)
            #Save Images of processed and unprocessed image
            cv2.imwrite(timestr + '_lines.jpg',origImg)
            cv2.imwrite(timestr + '.jpg',img2)
            #Resize the original and processed image
            image = cv2.resize(img2, (600,440))
            origImg = cv2.resize(origImg, (600,440))
            #Display the original and processed image
            updateCapImage(image)
            updateProcImage(origImg)
            lines = []
            #Retrieve new time stamp
            start_time = time.time()
            print("Image Processed")
            return
        else:
            #Do Nothing
            return

def disconnect():
    lblConnectionStatus.config(bg="red")
    connectionStatus.set("Not Connected")
    print("Camera Unit Disconnected")
    updateCapImage(imgP)
    updateProcImage(imgP)
    btnExit.configure(state = NORMAL)
    btnStartCapture.configure(state = DISABLED)
    btnStopCapture.configure(state = DISABLED)

def exitProg():
    print("Bye bye")
    print("Closing Monitoring System")
    print("Program Done")
    camera.close()
```

```
    GPIO.cleanup()
    top.destroy()

def startCapture():
    layerVal = 0
    var.set("Job Loaded")
    btnStartCapture.configure(state = DISABLED)
    btnStopCapture.configure(state = NORMAL)
    #Setup File for data storage
    file.write("Recoater Monitoring System \n")
    file.write("\n")
    file.write("Layer Number,Image
Name,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11,Q12,Q13,Q14,Q15,Q16\n")
    #Interrupt Setting
    GPIO.add_event_detect(23, GPIO.FALLING, callback=captureImage, bouncetime=5000)

def stopCapture():
    layerVal = 0
    btnStartCapture.configure(state = NORMAL)
    btnStopCapture.configure(state = DISABLED)
    file.close()
    GPIO.remove_event_detect(23)
    print("Event Removed")

def updateProcImage(srcImage):
    img = cv2.resize(srcImage, (600,440))       #Resizing Image
    im = PIL.Image.fromarray(img)
    imgtk = PIL.ImageTk.PhotoImage(im)
    lblProcessed.configure(image=imgtk)
    lblProcessed.image = imgtk

def updateCapImage(srcImage):
    img = cv2.resize(srcImage, (600,440))       #Resizing Image
    im = PIL.Image.fromarray(img)
    imgtk = PIL.ImageTk.PhotoImage(im)
    lblCamera.configure(image=imgtk)
    lblCamera.image = imgtk
```

```
#Button Declarations
btnStart = Button(top, text = "Connect", command = changeLabel)
btnStop =  Button(top, text = "Disconnect", command = disconnect)
btnExit = Button(top, text = "Exit", command = exitProg)
btnStartCapture = Button(top, text = "Start Capture", command = startCapture, state = DISABLED)
btnStopCapture = Button(top, text = "Stop Capture", command = stopCapture, state = DISABLED)

#String Variables
connectionHead = StringVar()
connectionStatus = StringVar()
heading = StringVar()
var = StringVar()
wLabel = StringVar()

#Image Declarations
lblCamera = Label(top, image = imgtkP)
lblCamera.image = imgtkP
lblProcessed = Label(top, image = imgtkP)
lblProcessed.image = imgtkP

#Label Declarations
lblConnectionHead = Label(top, textvariable = connectionHead, font = ("Helvetica",10))
lblConnectionStatus = Label(top, bg = "red", textvariable = connectionStatus, font = ("Helvetica",10))
lblHeading = Label(top, textvariable = heading, font = ("Helvetica",20))
lblWhiteVal = Label(top, textvariable = var, font = ("Helvetica",20))
lblWhite = Label(top, textvariable = wLabel, font = ("Helvetica",20))

#Setting Labels
connectionHead.set("Connection Status: ")
connectionStatus.set("Not Connected")
heading.set("Re-Coater Monitoring System")
var.set("Job Unloaded")
wLabel.set("Job Status: ")

#Placing of Buttons
btnStart.place(height = 50, width = 100,x = 20,y = 600)
btnStop.place(height = 50, width = 100,x = 20,y = 660)
btnExit.place(height = 50, width = 70,x = 280,y = 600)
```

```
btnStartCapture.place(height = 50, width = 100,x = 140,y = 600)
btnStopCapture.place(height = 50, width = 100,x = 140,y = 660)

#Placing of Labels
lblConnectionHead.place(x = 20, y = 50)
lblCamera.place(x = 20, y = 80)
lblProcessed.place(x = 650, y = 80)
lblConnectionStatus.place(x = 150, y = 50)
lblHeading.place(x = 430, y = 10)
lblWhiteVal.place(x = 450, y = 550)
lblWhite.place(x = 80, y = 550)

#Start of main loop
top.mainloop()
```

**Annexure D:**


B/W%
Layer Number
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
500
1000
1500
2000
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q10
Q11
Q12
Q13
Q14
Q15
Q16