Investigation of Low Resolution Point Clouds for Illumination Correction in Pushbroom Hyperspectral Images

A thesis presented in partial fulfilment of the requirements

for the degree of

Master of Engineering
in
Mechatronics

at

Massey University, Turitea Campus
Palmerston North
New Zealand

William B. Haarhoff
February 2018

ABSTRACT

Global food demand is predicted to double between 2015 and 2050. Current agricultural production is unable to facilitate this growth. Consequently, plant breeding must be accelerated to breed improved cultivars that can meet this demand. While technologies such as genomics are suitable for accelerating plant breeding, phenotyping lags behind and is currently considered the bottleneck. Consequently, imaging and remote sensing technologies are being used to provide quantitative, reliable phenotype information. One such technology; hyperspectral imaging can provide physiological, biophysical, and biochemical phenotypic information. While hyperspectral imaging has reached a substantial level of maturity in aerial and satellite based remote sensing applications, it is still underdeveloped in the close-range lab-based phenotyping scenario. In particular is the effect of illumination and complex plant geometry which affects the measured signal and is even more pronounced in the close range hyperspectral imaging. Methods for correction of illumination/geometry effects developed for aerial, and satellite-based imaging are unsuitable for close range hyperspectral imaging. Recently there has been an interest in fusing hyperspectral images with point clouds captured by 3D imaging devices to provide more comprehensive high dimensional phenotype information. However, one study focusses on the possibility of using 3D geometry of the plant to correct for the effects of illumination in hyperspectral images. This study investigates the use of low resolution point clouds captured with low cost devices for use in illumination modelling and correction of hyperspectral images acquired in close range lab-based scenario.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor Ian Yule for providing me with this great opportunity to learn about the interesting topics of hyperspectral imaging, machine vision, and multiple view geometry and the unique experience of seeing New Zealand from the air while capturing aerial hyperspectral images over the north and south islands. I would also like to thank Ian for both providing me with a stipend and covering my course fees.

I would like to thank Gabor Kereszturi for the many invaluable discussions on all aspects of this project, and especially his excellent feedback and help with the editing of this thesis.

I would like to thank Gourab Sengupta for convincing me that completing a master's degree is a worthwhile endeavour, and for his input with regards to both experiments undertaken, and structure of the thesis.

I would like to thank Kate Saxton for helping to organise my supervisors and I and helping to quickly resolve any administrative issues.

Ian Thomas, Clive Bardell, Kerry Griffiths for their advice and help with manufacturing a precise calibration gauge which was used for assessing the calibration of the hyperspectral camera.

I would like to thank Jacques Johnston for our many interesting discussions regarding clear thinking, postgraduate life, and robotics.

Finally, thank you to my wife Lize for supporting me through the highs and lows of my master's degree.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API: Application Programming Interface

BRDF: Bidirectional Reflectance Distribution Function

CAD: Computer Aided Design

DAC: Data Acquisition Computer

DEM: Digital Elevation Model

DLT: Direct Linear Transform

DN: Digital Number

DSLR: Digital Single Lens Reflex (Camera)

FOV: Field of View

GIFOV: Ground Instantaneous Field Of View

HU: Hyperspectral Unmixing

IARR: Internal Apparent Reflectance Ratio

ICP: Iterative Closest Point

IR: InfraRed region of the electromagnetic spectrum

LiDAR: Light Detection And Ranging

NDVI: Normalised Difference Vegetation Index

NGS: Next Generation Sequencing

NIR: Near infra-red region of the electromagnetic spectrum 700-1000nm

PCL: Point Cloud Library

QTL: Quantitative Trait Linkage

RANSAC: RANdom SAmple Consensus

RGB: Red Green Blue

ROI: Region Of Interest

SAD: Sum of Absolute Differences

SFM: Structure From Motion

SIFT: Scale Invariant Feature Transform

SLAM: Simultaneous Localisation And Mapping

SNR: Signal to Noise Ratio

SWIR: Short Wave InfraRed region of the electromagnetic spectrum 1000-2500nm

VIS: VISible region of the electromagnetic spectrum 400-700nm

1 INTRODUCTION

1.1 CONTEXT AND BACKGROUND

Global agricultural crop demand is predicted to roughly double between 2005 and 2050 due to increasing populations, shifting diets, and increasing biofuel consumption(Ray et al. 2013, Godfray et al. 2010, Tilman et al. 2011). This is problematic since yield trends are either stagnating or not increasing at a rate that is sufficient to meet the required doubling of global crop production by 2050 (Ray et al. 2013, Ray et al. 2012) forming what has been commonly termed '*the yield gap'*. One strategy to meet this demand is agricultural expansion though clearing more land and growing more crops. However, this would result in ~1 billion ha land being cleared, CO_2 emissions reaching ~3Gty^{-1}, and N use would reach ~250Mty^{-1} by 2050 (Tilman et al. 2011). The preferred, and more sustainable approach is to attain higher yields on existing croplands (Tilman et al. 2011, Godfray et al. 2010). To close the yield gap in a sustainable manner requires that plant breeding, and ultimately yield gains are accelerated (Araus et al. 2008, Gerald, Frei, and Lübberstedt 2013, Yang et al. 2013). Plant breeding can be defined as 'the art and science of changing genetic architecture of plants for the benefit of mankind' (Panguluri and Kumar 2016). The goal of plant breeding is to maximise a given set of phenotypic traits, such as yield or drought resistance, meaning that plants are selected for on a phenotypic basis. (Panguluri and Kumar 2016) define a phenotype as: '*any measurable characteristic or trait of a plant and is a result of combination of genes expressing in the plant, environmental influence, and their interactions'.* To breed higher yielding plants, the complex web of gene- and environment interactions must be understood and linked to phenotype.

Phenotyping along with plant genotyping are the foundational techniques that are used in plant breeding to breed new cultivars with superior properties such as increased yield potential, and increased biotic and abiotic stress resistance (Cobb et al. 2013). It is phenotyping that is relied upon to quantitatively establish the link between the plant genetics and complex traits, such as yield potential, and drought resistance(Blum 2011, Campos et al. 2004). In the context of plant breeding, quantifying the genetic landscape with respect to these traits facilitates accelerated breeding of "better" cultivars(Gerald, Frei, and Lübberstedt 2013, Yang et al. 2013). While plant genotyping technologies have developed and matured significantly in recent years, those of plant phenotyping are sorely underdeveloped, being widely considered as the bottleneck to genetic analysis and plant breeding (Cobb et al. 2013). Consequently, phenotyping technologies need to be improved to facilitate an accelerated breeding of "better" cultivars, and meet the increased crop demands in a sustainable manner (Yang et al. 2013). Cobb et al. (2013) offer an apt comparison between genotyping and phenotyping, describing the former as mechanised and uniform across all organisms, while the latter is likened to a cottage industry that is species specific, labour intensive, and sensitive to the environment. Furthermore, Tardieu and Tuberosa (2010) describe the traditional phenotyping process as manual, subjective, inefficient, destructive, and error-prone. To make full use of the significant advances in genomics that has happened over

the last two decades (Cobb et al. 2013), the phenotyping paradigm has shifted to value techniques that have potential for high throughput (Li, Zhang, and Huang 2014, Kipp et al. 2014), and high quality phenotypic information. Consequently, 2D and 3D imaging techniques have gained significant importance within the phenotyping community (Li, Zhang, and Huang 2014), including hyperspectral imaging in close range plant phenotyping scenarios.

1.2 Problem Statement

Hyperspectral imaging is traditionally used in remote sensing applications such as geological mapping, agriculture, oceanography, ecology, law enforcement, military, study of ice and snow (Goetz 2009). Hyperspectral imaging also has huge potential for phenotyping applications as it can be used to observe biochemical, biophysical, and physiological plant traits in a non-destructive manner, and has potential for autonomous data collection (Sytar et al. 2017). Hyperspectral imaging has already been used in some phenotyping applications such as: characterization of plant pathogens (Kuska et al. 2015), estimation of leaf water content (Ge et al. 2016), and estimation of leaf chlorophyll content (Thorp et al. 2015).

As hyperspectral imaging has been primarily used in aerial and satellite platforms, techniques and methodologies found in the literature pertain mostly to dealing with phenomena (e.g. atmospheric effects and illumination changes) are minimized in a close range, controlled lab-based application. In close range applications complex plant geometry and illumination interactions can give rise to spurious reflectance signals, which cannot be corrected by common techniques including: flat field, empirical line corrections (Mishra et al. 2016, Behmann et al. 2016). Behmann et al. (2015) while investigating hyperspectral imaging of sugar beets found that the experiments suggested more than 60% of the spectral information depends on plant geometry. Correcting for the effects of complex plant geometry and illumination interactions is a very relevant and active area of research. Mishra et al. (2016) attempted to account for the effects of complex geometry and illumination for close range correction of hyperspectral data. However, this model required a reference spectrum to be chosen from within the scene ultimately making the results scene dependent, reducing the potential for comparison between phenotypic data sets. Behmann et al. (2016) proposed that a synergy between hyperspectral data and point clouds could be used to correct for complex geometry and illumination interactions. Furthermore, there has been a number of recent studies adopting or advocating the fusion of point clouds and hyperspectral data for improving insight into phenotyping problems (Aasen et al. 2015, Behmann et al. 2016, Busemeyer et al. 2013, Virlet et al. 2017).

1.3 Research Objectives & Questions

The main objective of this research is to investigate the correction of complex plant geometry and illumination interaction effects on hyperspectral images obtained from close range phenotyping. Furthermore, this research aims to investigate the suitability of point clouds and meshes obtained with low cost sensors for modelling the plant geometry for illumination correction. The main research question of this thesis: '*are point clouds and meshes obtained from low cost sensors a suitable tool for correcting complex illumination effects present in hyperspectral images within the application of close range phenotyping?*' This research question can be further sub divided into two sub questions:

- 'What is the accuracy of the models obtained by the low-cost 3D sensors and what is the effect of this accuracy on the illumination correction?'
- 'What methodology should be used for hyperspectral camera calibration in a close-range phenotyping scenario? What accuracy can be achieved from this calibration? And what effect will this have on the final illumination correction?

1.4 Thesis Outline

This thesis is divided into the following seven chapters: Introduction, Literature Review, Setup of the study, Methodology, Results, Discussion, Conclusion and Outlook. The thesis begins with the literature review which examines the literature pertaining to the requirements of hyperspectral imaging within the domain of close range phenotyping, the utility and methodologies of hyperspectral imaging, pushbroom camera models and their attainment, fusion of hyperspectral and point cloud data, and illumination correction methodologies within hyperspectral imaging. The study and Setup chapter briefly describes the experimental set up, equipment used, and the manufacture and verification of custom apparatus necessary for experiments, and the acquisition of data. The Methodology chapter is to elaborate on the procedure for a sub module of illumination correction followed by methodology for the assessment of the efficacy of that module. The procedures for illumination correction include pushbroom camera calibration, camera model decomposition, point cloud pre-processing, point cloud scale recovery, semi-automatic registration of the point cloud to world coordinate system, mesh attainment, ray tracing implementation, ray tracing support of multiple geometric primitives, geometric primitives and ray intersection tests, image radiometric correction, computing of Lambertian effects, computing of specular effects. Subsequently methodology is described for an efficacy assessment of non-linear effect removal, calibration stability, wavelength dependency of calibration, bias in angle estimation of planar surfaces in acquired point clouds, mesh angle deviation with textural frequency and radius, assessment of illumination predicted effects with actual illumination over cylinders and planes, comparison of corrections with ground truth reflectance spectra. The results chapter simply presents the findings of the experiments conducted as well as any relevant observations made that were not part of the

experiments conducted. The discussion chapter is concerned with the interpretation of the results and observation, with emphasis on any apparent disagreements and its cause. The conclusions and outlook chapter aim to answer the research questions outlined in the prior section, and provide recommendations pertaining to the extension of this research.

2 LITERATURE REVIEW

2.1 PHENOTYPING FOR THE GENOMICS ERA

2.1.1 Evolution of Plant Breeding and Phenotyping

Plant breeding is predicated upon genetic diversity, which forms the basis for all plant improvement (Cobb et al. 2013). The diversity and variation of traits means that some plants will exhibit superior performance regarding a given trait (i.e. biomass yield, drought resistance). Early plant breeding selected plants on a purely phenotypic basis with no insight into inheritance of genes. In the 20th century conventional methods aim to understand the inheritance of traits, and using replication and sophisticated experiments to select for better cultivars on a genetic basis (Cobb et al. 2013). This was immensely successful in the 20th century, yields were dramatically increased (Pérez-de-Castro et al. 2012), resulting in the 'green-revolution' which is attributed with preventing the starvation of over a billion people. Conventional methods use molecular techniques such as quantitative trait locus (QTL) to determine the likelihood that a region of DNA is linked with a phenotypic trait. For simple traits where only, a few loci were attributed to contribute to the trait, these regions would be sequenced and compared to a database of known genes to understand their contribution to the phenotype.

Recently, plant breeding has been faced with the ever-growing demand for increased yield. While yield has been optimised regarding a handful of loci that have had significant individual effect, the remaining effects are complex. Meaning the phenotype is a consequence of significant interactions between many loci, with weak or insignificant individual effect. (Houle, Govindaraju, and Omholt 2010) describes yield as a complex quantitative trait where the phenotype is a consequence of a complex web of interactions between multiple genes and the environment. Dissecting this complex web requires a paradigm shift from conventional plant breeding methods.

Genomics is the technology poised to deliver this paradigm shift (Pérez-de-Castro et al. 2012). Bringing with is a set of new methodologies and powerful tools, genomics will facilitate the direct empirical study of genotype with the phenotype. A popular tool is the DNA sequencing, often referred to as next-generation-sequencing (NGS). NGS allows the rapid sequencing of the entire genome thereby facilitating the study of all loci even if they have a weak individual effect. Coupling this accessible, state of the art technology with precise phenotypic information will allow the development of a genotype-phenotype map which will model the complex web of interactions. This will increase prediction accuracy of cultivar traits and contribute to increased efficiency and economy in the breeding process. Just as conventional breeding of the 20th century resulted in the green revolution, genomics is leading the 'greener' revolution which is expected to revolutionize plant breeding in the 21st century (Pérez-de-Castro et al. 2012).

Plant breeding in the genomics era requires both genotypic and phenotypic information to establish a genotype-phenotype map. Moreover, utilising the advances in genomics requires large volumes of highly

reliable genotypic, and phenotypic information. Unlike genomics, phenotyping is not able to deliver the high volumes of precise data required, thereby limiting the exploitation of genomics for plant breeding. Consequently, many trends have emerged in the literature relating what is required of phenotyping to be non-limiting in the genomics era, these include:

- Gathering, reliable, precise, and accurate, quantitative data (Araus et al. 2008, Cabrera-Bosquet et al. 2012).
- Data needs to be gathered in a high throughput manner (in order to achieve statistical significance huge volumes of plants must be scored relatively quickly) (Golzarian et al. 2011).
- Fusion of many data types e.g.: hyperspectral image, point cloud, environmental data (Houle, Govindaraju, and Omholt 2010).

In contrast to these requirements, conventional methods of phenotyping are qualitative, slow and labour intensive. One major project undertaken by phenotyping has been the study of yield related traits through large scale field-based corn breeding trials. White et al. (2012) reported a breeding project with 20,000 plots of 1mx4m which would take 27hrs to visually score, assuming the scorer could maintain a constant speed of 3Km/h without stopping. This task becomes near impossible to score in real time when physiological or morphological measurements are required (Zhang et al. 2016). Moreover, destructive practices are typically employed in nitrogen estimation or biomass measurements (Zhang et al. 2016) which are particularly problematic as the number of plants within each hybrid line are limited. Imaging technologies have been identified as meeting the requirements that conventional methodologies cannot and are considered to be paving the way forward for gathering phenotyping data that is quantitative, precise, accurate, in a high throughput manner. Liebisch et al. (2015) estimated that the scoring of the 20,000 plots described in (White et al. 2012) could be achieved in just 6 minutes using visible light, near infrared and infrared imaging technologies. Consequently, many imaging technologies are being adapted from their traditional use in other applications to phenotyping such as hyperspectral imaging being adapted from aerial imaging to high throughput, controlled environment phenotyping.

2.1.2 Imaging technologies for phenotyping

Imaging technologies have gained significant attention from the phenotyping community as they provide the means to quantify the plant phenotype in high resolution over a range of environments, and development stages. Many of which are suited to automation which will in turn facilitate phenotyping of a high throughput nature. Yang et al. (2013) argues that many imaging techniques have shown significant utility in remote sensing applications (especially regarding quantifying properties of vegetation), and it is the necessary next step for these remote sensing technologies to be absorbed into phenotyping platforms. While imaging technologies have been affirmed as the technology to relive the phenotyping bottleneck, they have yet to

mature. There is a need for standardisation of experimental protocols, sensor calibration(e.g.: radiometric, geometric corrections), and raw data processing routines before imaging technologies can fully be taken advantage of for phenotyping (Li, Zhang, and Huang 2014). Minervini, Scharr, and Tsaftaris (2015) argue that image analysis has become then new bottleneck in plant phenotyping, with the analysis of the imaging data being the weakest or even missing link. A range of 2D and 3D imaging solutions have been used for phenotyping some examples include: including Red-Green-Blue (RGB) visible light imaging, thermal imaging, fluorescence imaging, multispectral, hyperspectral, and a range of 3D acquisition technologies.

Visible light imaging utilises cameras sensitive to light in the 400nm to 700nm for the purposes of quantification of physical, structural, morphological, and volumetric traits (Li, Zhang, and Huang 2014), e.g. Golzarian et al. (2011) used visible light imaging to accurately infer shoot biomass in barley, while Arvidsson, Pérez-Rodríguez, and Mueller-Roeber (2011) developed an image processing pipeline for growth monitoring of *Arabidopsis Thaliana*. However, there are several limitations associate with visible light imaging in a field-based phenotyping context that limit the automation of the technique and thereby many of the benefits over conventional methods (speed, reduction in labour, etc.). Li, Zhang, and Huang (2014) report that the similar brightness and colour of the background and leaves, shadows present in images, and variable lighting conditions are limiting the ability to automate the postprocessing of images and extract phenotype parameters. Near infrared imaging can be used to measure phenotype parameters such as water content composition, and leaf area index. This technology has been successfully applied in rice, soybean, maize, barley and wheat (Li, Zhang, and Huang 2014). A major disadvantage of both visible light imaging and NIR cameras is that they are sensitive to broad regions of the spectrum, and information contained within specific wavebands is lost (Fahlgren, Gehan, and Baxter 2015). Fluorescence imaging plays an important role in quantifying the photosynthetic performance of a plant. Some examples include (Estrada et al. 2015) who used fluorescence imaging in blueberry breeding for genotype selection under drought conditions, while Ploschuk et al. (2014) used fluorescence imaging for assessment of chilling, and freezing induced stress of *Jatropha Curcas* in early vegetative stages. 3D imaging technologies are concerned with quantification of volumetric, or structural properties e.g. Hosoi, Nakabayashi, and Omasa (2011) used LiDAR to quantify the degree of leaf droop via estimation of leaf area and inclination angle, while Keightley and Bawden (2010) used LiDAR for 3D volumetric modelling of grapevine biomass.

2.1.3 Limitations of lab based hyperspectral imaging

Lab based hyperspectral imaging has been used within the context of phenotyping for detection of early plant stress responses (Behmann, Steinrücken, and Plümer 2014)), and proximal sensing of plant diseases (Mahlein 2016). However, within this context both Behmann et al. (2016) and Mishra et al. (2016) have noted that a major limitation of this technology is the degree to which .measured reflectance in images are influenced by complex illumination and plant geometry effects. Behmann et al. (2015) reports that experiments suggested that 60% of spectral information is affected by complex illumination and plant geometry interactions.

2.2 HYPERSPECTRAL IMAGING

2.2.1 Overview of hyperspectral imaging

Unlike visible light RGB cameras which are only sensitive to 3 bands (red, green, and blue) in the visible region of the spectrum, hyperspectral cameras are sensitive to hundreds of contiguous and narrow spectral bands typically covering the VIS - Visible (400-700nm), NIR – Near-Infrared (700-1000nm), and SWIR – Shortwave Infrared (1000-2500 nm) regions of the electromagnetic spectrum. The principle behind remote sensing in general is that electromagnetic radiation will interact with objects (e.g. absorption, reflectance and transmission), areas or phenomena in a manner that is dependent on their intrinsic properties. As different electromagnetic wavelengths will have unique interactions with the same object, measuring the reflected radiation allows the characterisation of a reflectance spectrum over a set of wavelengths. As defined by Nicodemus (1977) 'Reflection is the process by which electromagnetic flux (power), incident on a stationary surface or medium, leaves that surface or medium from the incident side without change in frequency; reflectance is the fraction of the incident flux that is reflected.' Hence, it is expressed as:

$$\rho(\lambda) = \frac{\varphi_r(\lambda)}{\varphi_i(\lambda)}$$

Where reflectance varies as a function of wavelength making it convenient to express this variation in the form of a reflectance profile, where reflectance is plotted against wavelength. Fig.2.1 shows an example of reflectance profiles for different minerals. Each material has a unique reflectance signature. The analysis of these reflectance signatures has led to the development of a range of earth observation applications including classification applications such as land cover classification, and geological mapping, and quantification and

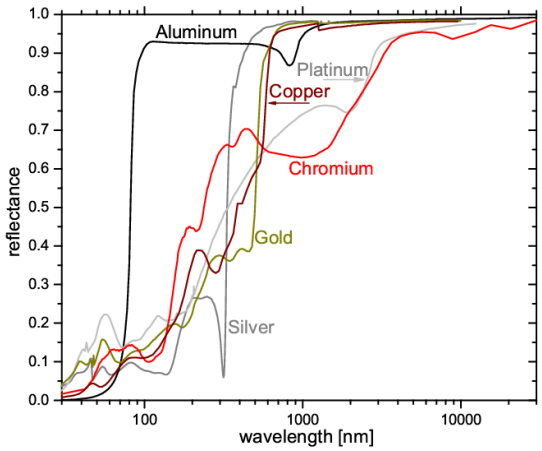

Figure 2.1 - Reflectance Profiles

Figure 2.2 Hyperspectral Data Cube (Ibraheem 2015)

prediction applications such as prediction of macro and micro nutrients in mixed pastures (Pullanagari, Kereszturi, and Yule 2016).

The raw data gathered by a hyperspectral sensor has three dimensions, two spatial directions and a third spectral dimension. Within hyperspectral, and remote sensing communities, this is often referred to as a data cube (Fig.2.2). It is helpful to conceptualise the data cube as a stack of 2D spatial photographs, where each pixel has a 3rd spectral dimension.

2.2.2 Hyperspectral imaging for vegetation

The spectral properties of vegetation are strongly determined by their biophysical, and biochemical attributes (Pullanagari, Kereszturi, and Yule 2016) some of which include moisture content, pigments (such as chlorophyll), live biomass, senesced biomass (Thenkabail and Lyon 2016). The high absorption of radiation in visible region of the spectrum is due to leaf pigments such as chlorophylls, with only a small green reflectance peak observed around 550nm (Knipling 1970). Photosynthesis relies on absorption of visible light by the chlorophyll, which is used to synthesize nutrients. Deviation from ideal chlorophyll absorption can be detected in the reflectance spectrum, and plant health/stress can be inferred. The NIR region of the spectrum is dominated by the cell internal structural features (Knipling 1970). The significant rise in reflectance values can be explained by lack of pigment and water absorption features in this region, and an increased amount of light scattering at cell-wall interfaces due to a change in the index of refraction (Knipling 1970). Finally, the SWIR region of the spectrum is dominated by absorbance's of C-O, O-H and N-H bonds. These organic bonds make up organic compounds such proteins, lignin and cellulose (Thenkabail and Lyon 2016).

2.2.3 The hyperspectral advantage and pixel unmixing

Unlike multispectral cameras, hyperspectral cameras commonly capture hundreds of contiguous wavebands, which can be used in material discrimination and abundance mapping through pixel unmixing. If the scattering between distinct endmembers or components of a pixel is negligible then the spectral mixing can be approximated by a linear model (Bioucas-Dias et al. 2012).

$$y_i = \sum_{j}^{p} p_{ij}\alpha_j + w_i$$

Where y_i is the observed spectra at band i, p_{ij} is the reflectance of band i for endmember j. α_j is the relative contribution of endmember j while w_i is additive noise peturbation. This can be reformulated in matrix form.

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{\alpha}$$

Where $\boldsymbol{M}$ is the n x p mixing matrix, whose columns are the endmembers, $\boldsymbol{y}$ is the n x1 mixed spectra column vector, and $\boldsymbol{\alpha}$ is the p x1 relative abundance cooefficent column vector. The goal of hyperspectral unmixing

(HU) is to solve the inverse problem and determine α_j, for all endmembers. Over the last 25 years there have been numerous published algorithms and models that are fundamentally based on the linear mixing models (Bioucas-Dias et al. 2012). Geometrical approaches are one of the common methods to determine the relative abundances. These consider the p number of wavebands as p-dimensional space that can be spanned by p-dimensional vectors. The linearly mixed vectors can be contained within a simplex set, or positive cone (Bioucas-Dias et al. 2012), facilitating a geometrical based computation of endmember abundance. Often the number of endmembers is much smaller than the number of wavebands available. A dimensionality reduction is often performed using techniques such as principle component analysis. This has the advantage of improving algorithm performance, complexity, and data storage (Bioucas-Dias et al. 2012).

2.2.4 Phenotyping requirements for hyperspectral imaging

In contrast to the development of hyperspectral imaging during the 1990's, the phenotyping community has reiterated the need for comprehensive, multidimensional data that can be "mined" to unravel the complex gene-environment interactions and develop a genotype phenotype map. Hyperspectral imaging has already shown significant promise, already being used in aerial (Aasen et al. 2015, Suomalainen et al. 2014), ground based (Busemeyer et al. 2013), and lab based (Behmann et al. 2016) platforms for phenotyping applications. There are several requirements for hyperspectral imaging to be useful to phenotyping, which are briefly outlined below.

2.2.4.1 Signal to noise ratio (SNR)

The inverse problem of HU is often difficult due to the often-strong spectral correlation which results in badly conditioned mixing matrices ultimately making HU estimates highly sensitive to noise (Bioucas-Dias et al. 2012). Consequently, HU requires a high signal to noise ratio (SNR).

2.2.4.2 Accurate, precise, reliable phenotypic data

As outlined in the phenotyping sections, the genomics era requires accurate, precise, reliable data. There are many actors that can compromise the resulting spectra, classification and relative abundance determinations. In the close-range-lab based-context, these are predominantly illumination-topography factors which range from simple spectral scaling (Lambertian) effects, to very complex view direction, wavelength dependent effects due to anisotropic surface reflectance properties. These effects and their mitigation or reduction will be discussed in more detail in subsequent sections.

2.3 ILLUMINATION FACTORS AND METHODOLOGIES FOR THEIR CORRECTION

The term reflectance as defined in section 2.2.1 referes to the ratio of all light reflected from a surface to all light incident on a surface. In in the context of vegetation and phenotyping, hyperspectral imaging is primarily concerned with only the ratio of light reflected diffusely (Lambertian reflectance) to all light incident on a surface. Illumination correction in this context can be defined as: methodologies or techniques for recovering the ratio of light reflected from a surface in a diffuse manner, from either apparent reflectance or radiance measurements.

2.3.1 Lambertian factors

In remote sensing of vegetation, the vegetation is often assumed to be a Lambertian reflector (i.e. reflectance does not vary with viewing angle). Lambertian effects deal primarily with the radiation incident to the vegetation. These effects are described by lamberts cosine law and inverse square law. Lamberts cosine law characterises the attenuation of irradiance at the vegetation surface by the cosine of the angle between the surface normal vector and the vector extending from the surface to the point light source.

Inverse square law describes the attenuation of irradiance at the surface due to the distance between the light source and surface. If the distance between the light source and the surface is increased by factor k then the irradiance is attenuated by factor k^2. Lambertian effects depend only on the distance inclination of a surface with respect to a light source and are wavelength independent. Consequently, spectrally identical surfaces at different inclinations and positions will have a measured reflectance profile that is scaled with respect to one another.

2.3.2 Specular reflections & BRDF

While useful, the assumption of perfect Lambertian reflector is not true for many materials. More likely they exhibit various combinations of Lambertian and non-Lambertian behaviour. A perfect Lambertian reflector will have a microscopically rough surface so that incident light is reflected isotopically (Nicodemus 1970).

In practice there are no ideal Lambertian or specular surfaces. Moreover, the specular reflection is not only in one direction but will exhibit often unique distributions of intensities around a direction of reflection that is equal to angle of reflection. In airborne and spaceborne remote sensing applications these effects have been empirically modelled with a Bidirectional Reflectance Distribution Function (BRDF). Current BRDF databases of materials would not be suitable for close range applications as their modelling does not correspond to a sub-leaf scale and requires leaf area index as input which is not applicable in close range phenotyping.

2.3.3 Overview of traditional correction methodologies

In remote sensing two main branches of reflectance correction techniques to minimise atmospheric and sensor and topographic effects. These can be broadly divided into empirical, and radiative transfer modelling. During the 1980s many empirical approaches were developed to correct for atmospheric effects, although some also account for topographic effects. Both absolute and relative empirical methods were developed including the logarithmic residuals approach by (Green and Craig 1985), flat field correction (Roberts, Yamaguchi, and Lyon 1986), empirical line approach (Conel et al. 1987), and the internal average relative reflectance (IARR) by (Kruse 1988). These empirical techniques are both computationally and theoretically simplified by their underlying assumptions. However, these assumptions also impose limitations on the accuracy, utility, and versatility of the techniques as detailed in sections 2.3.4.1, 2.3.4.2, 2.4.3.3, and 2.3.4.4.

Radiative transfer techniques model the propagation of light through the atmosphere accounting for scattering and absorption and determine the resultant radiance measured by the sensor. This approach can remove the modelled scattering and absorption effect by the atmosphere, providing absolute reflectance. Many radiative transfer codes have been proposed to solve the inverse problem. These can model atmospheric effects MODTRAN (Berk et al. 1998); vegetation interaction with illumination PROSPECT (Jacquemoud and Baret 1990), SAIL (Jacquemoud et al. 2009), PROSAIL (Jacquemoud et al. 2009). These techniques alter the parameters of the forward radiative transfer model until the results converge with the measured radiance. This is employed using genetic algorithms, non-linear least squares, look up tables and a variety of other techniques.

2.3.4 Review of empirical correction methodologies

2.3.4.1 Log residuals

The log residuals approach was developed by (Green and Craig 1985) for the classification and mapping of minerals using aerial hyperspectral images. The method aims to remove topographic effects (assumed to be spatially varying, spectrally constant) and atmospheric effects (assumed to be spectrally varying, spatially isotropic) which are assumed to be multiplicative. Topographically equalised spectra are computed by dividing each pixel radiance spectra by its geometric mean. The atmospheric effects are then removed by dividing the topographically equalised spectra by its spatial geometric mean. The resultant reflectance spectra show deviation (like a residual) from the mean absorption features. In practice, the geometric mean is computed from the arithmetic mean of logarithms, hence the term log-residuals. This method has great utility for identifying spectral absorption features, which can be useful for classification operations. However, the method cannot account for specular reflections, and the pseudo reflectance values will be relative to the image they were derived from making it difficult to compare between different scenes.

2.3.4.2 *Flat field correction*

The flat field correction algorithm is often used to remove atmospheric effects from the spectra in the scene. The basic principle is that user defined region of interest (ROI) known to have a 'spectrally neutral' (having no absorption features) reflectance profile will exhibit radiance spectra predominantly characterised by multiplicative atmospheric effects. Each pixel within the image is then divided by the average of the flat field spectra (spectra in the ROI) resulting in a reflectance spectrum that varies relative to the flat field. In this way, multiplicative atmospheric effects are accounted for (divided out) at the expense of absolute reflectance, and additive effects. This technique can also be used for correcting sensor effects such as vignetting, or non-uniform detector element gains. This technique is still useful and commonly used, however it doesn't provide an absolute reflectance measure doesn't support comparison of reflectance values between scenes, cannot account for topographic or specular effects.

2.3.4.3 *Internal Apparent Reflectance Ratio (IARR)*

The IARR developed by (Kruse 1988) is very similar to the flat field correction technique, except that the 'spectrally neutral' region of interest is the entire image. The assumption is that the scene is so heterogeneous that averaging of the scenes spectra will result in spectrally neutral radiance spectra, which is characterised by only the multiplicative atmospheric effects. Again, dividing every pixel in the scene relative to the average spectra will divide out atmospheric effects. This technique has mostly the same advantages and disadvantages as flat field correction. However, it is even more limited by its assumption that the scene is heterogeneous enough. This is a particularly bad assumption to make if the scene contains significant amounts of vegetation.

2.3.4.4 *Empirical flat line correction*

The empirical flat line is an absolute reflectance derived from field measurements using on-ground targets with known absolute reflectance. The empirical flat line technique is concerned with deriving a linear relationship between reflectance of a given waveband, and the digital number recorded by the detecting element.

$$\rho = c_{\mathrm{a}} DN + c_b$$

Like the flat field correction technique, the methodology can be used to correct for a specific effect, such as atmospheric effects in aerial imagery, or detector gains and offsets in lab-based push broom imagery. The assumptions are implicit in the application of the technique, and so they vary based on the application. However, the most common case is removal of atmospheric effects in aerial imagery. In this case, the atmospheric effects are assumed to be isotropic over the scene. Provided two or more references of known absolute reflectance are present in the scene, a relationship between DN and absolute reflectance can be derived for each band. In the case of lab-based push broom imaging, the empirical flat line technique can be used to correct for differing detector gains and offsets of detector elements, and for varying scene

illumination. In this case reference panels of known reflectance can be made to span the view of all detector elements. In this way, the empirical line technique can be applied on a detector element basis as opposed to just a waveband basis thereby accounting for the variation between detector elements.

2.3.5 Review of methods accounting for geometry

2.3.5.1 Linear optical method

Mishra et al. (2016) proposed a methodology for correction of reflectance applied to the Phenovision close range hyperspectral phenotyping platform. While many correction methodologies have been successfully applied to remote sensing applications, the correction methodologies for close range hyperspectral imaging are currently underdeveloped. Airborne and spaceborne remote sensing methodologies use the sun as the source of illumination, in which case both the distance and angle between the sun and object surface being imaged is relatively constant over the scene. In the case of close range lab based hyperspectral imaging, there may be multiple light sources where the both the angle and distance between sources and surfaces will vary much more dramatically. Hence, Mishra et al. (2016) argue that leaf reflectance spectra are much more sensitive to plant geometry and distance and inclination towards the sensor and light source.

Mishra et al. (2016) propose a linear empirical model to account for the effects due to the geometry of the plant illumination sources and viewing angle as well as specular reflectance due to surface anisotropic reflectance.

$$\rho_{meas}(\lambda) = \beta\rho_{ref}(\lambda) + \alpha + n(\lambda)$$

$\rho_{meas}(\lambda)$ is the absolute reflectance spectrum obtained after the sensor has been radiometrically calibrated to a known reflectance standard after subtraction of a dark frame. The β term accounts for multiplicative effects due to illumination and plant geometry which can be described by inverse square law and lamberts cosine law $\beta \approx \cos(\theta)/d^2$. $\rho_{ref}(\lambda)$ refers to a reference reflectance spectrum that has similar spectral shape to $\rho_{meas}(\lambda)$. As specular reflectance is approximately constant with wavelength it can be modelled as an additive term (Vigneau et al. 2010) wavelength independent term α. The $n(\lambda)$ accounts for wavelength dependent sensor noise. The measured spectrum of an individual pixel is fitted to a reference spectrum through linear least squares yielding α, β and $n(\lambda)$. Using the fitted α, β terms, the measured spectrum can be corrected for distance, inclination, specular reflectance, and sensor noise by the following expression:

$$\rho_{corr}(\lambda) = \frac{\rho_{meas}(\lambda) - \alpha}{\beta}$$

This method has major disadvantages. Firstly, provided adequate reference spectra are chosen, the method will mask any changes which preserve the spectral shape i.e.: changes that shift the entire spectra up or down by the same constant over every waveband, or scale the spectra by the same constant over every waveband.

These changes in reflectance will be absorbed into the alpha and beta terms and won't be present in the final spectra. Secondly the choice of (plant)reference spectra is problematic as there is no way to ensure that the reference spectra chosen does not contain specular effects or is not abnormal. Thirdly the relative nature of the correction means that each images pseudo reflectance is only meaningful relative to the reference in the image from which it was derived limiting its use for comparison between images and therefore plants.

2.3.5.2 DEM for Citrus fruits

Moltó, Blasco, and Gómez-Sanchís (2010) used close range hyperspectral imaging for detection of defects in citrus fruit. The varying topography of the fruit meant the top of the fruit would be brightly illuminated while the edges would only be lit by ambient illumination. This variation in illumination masked the defects. Because the fruit has a smooth curved surface, Moltó, Blasco, and Gómez-Sanchís (2010) were able to generate a digital elevation model (DEM) of the fruit by fitting ellipses to the fruit based only on the hyperspectral image. The model was used to associate an inclination angle with every pixel of the fruit. A correction factor was calculated based on the inclination angle and applied across the image. This was able to significantly improve the measured reflectance when compared to traditional empirical techniques. Unfortunately, this technique is application specific, and can only be applied to objects with known to have smooth elliptical geometry, making it unsuitable for correction in images of plants.

2.3.5.3 Fusion with 3D geometry information

Roscher et al. (2016) propose a method for detection of disease symptoms using 3D hyperspectral plant models. The 3D hyperspectral point cloud is generated by fusing the hyperspectral image captured with a Pushbroom camera with the point cloud obtained by the laser scanner. They use a sparse representation which assumes that each pixel spectra can be reconstructed as a spare weighting of a few basis vectors contained within a dictionary. As opposed to most approaches which attempt to remove complex illumination effects such as BRDF, Roscher et al. (2016) builds these effects into the topographic dictionary, which is grouped by inclination angle obtained from the point clouds.

The fusion of point clouds and hyperspectral images is non-trivial as noted by Behmann et al. (2016). Relating the hyperspectral image to the world coordinate system requires a camera model to be obtained. The camera model encodes extrinsic parameters such as the rotation and translation of itself with respect to the world coordinate frame, and intrinsic parameters such as focal length and pixel aspect ratio. The point cloud needs to be captured, and then transformed into the world coordinate system so that it can be related to the hyperspectral image.

2.4 CAMERA MODELS

2.4.1 Note on Notation

A brief note on the mathematical notation used in this thesis. Non-scalar quantities (i.e.: vectors and matrices) are differentiated from non-scalar quantities by bold type face. Rotation and translation matrices $\boldsymbol{R_{AB}}$, $\boldsymbol{T_{AB}}$ define the rotation of one coordinate frame with respect to one another. Hence, the subscript indicates these coordinate frames. For example, $\boldsymbol{R_{AB}}$ is the rotation of coordinate frame $\boldsymbol{B}$ with respect to coordinate frame $\boldsymbol{A}$. $\boldsymbol{R_{BA}}$ is a different quantity and must not be confused with $\boldsymbol{R_{AB}}$.

2.4.2 Pinhole camera model

The most commonly known and used camera model is the one for a simple pinhole camera model. This model allows the specification of intrinsic and extrinsic camera parameters with a compact and efficient representation by a 3x4 matrix. The intrinsic matrix $\boldsymbol{K}$ containing parameters such as focal length, x, y offset can characterise the mapping of 3D points known in the camera 3D coordinate system to the 2D image coordinate system.

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \boldsymbol{K}\boldsymbol{x} = \begin{bmatrix} f & 0 & P_x \\ 0 & f & P_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

The u, and v pixel coordinates are extracted by normalising the vector with respect to factor w. Extending the model to characterise the mapping of 3D points in a world coordinate system to the image plane requires information about the position and orientation of the world with respect to the camera coordinate system. This mapping is expressed by the 3x1 transformation matrix $\boldsymbol{T_{cw}}$, and 3x3 rotation matrix $\boldsymbol{R_{cw}}$, which encode the translation and rotation of the world system with respect to the 3D camera coordinate system. A point in the world coordinate system $\boldsymbol{x_w}$ can be mapped into the camera coordinate system using the following transformation:

$$\boldsymbol{x_c} = \boldsymbol{R_{cw}}\boldsymbol{x_w} + \boldsymbol{T_{cw}}$$

This can be more compactly expressed using an extrinsic camera matrix to encode this transformation.

$$\boldsymbol{E} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{23} & r_{33} & t_z \end{bmatrix}$$

The camera matrix $\boldsymbol{M}$ can efficiently transform a point known in a generic coordinate system and map it to the image plane using. For clarity the entire camera matrix is shown below.

$$\boldsymbol{M} = \boldsymbol{KE}$$

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & P_x \\ 0 & f & P_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{23} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

2.4.3 Linear pushbroom model

Gupta and Hartley (1997) proposed the linear push broom model for computational simplicity while still providing accurate results comparable with more rigorous, mathematically intensive traditional models used by the remote sensing community. The model takes the same form (3x4 matrix) as the widely known and used pinhole camera model meaning concepts from pinhole camera models can be applied to pushbroom cameras as well, these are intrinsic calibration matrix $\boldsymbol{K}$, and rotation and translation matrices $\boldsymbol{R_{cw}}$, and $\boldsymbol{T_{cw}}$,. The main difference is the formation of intrinsic matrix $\boldsymbol{K}$. while the pinhole models encode a perspective projection in both u, and v pixel axes centred at the focal point, the pushbroom model encodes a perspective transformation in only the v direction. The u direction encodes an orthographic projection. Behmann et al. (2016) have already demonstrated the utility of this camera model for hyperspectral imaging of plants in a close-range phenotyping scenario. The linear pushbroom model uses a velocity vector $\boldsymbol{V}$, to characterise the relative movement of the projective centre of the pushbroom model with respect to movement between pixel lines in the u direction. The intrinsic matrix can be expressed as:

$$\boldsymbol{K} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & f & p_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/V_x & 0 & 0 \\ -V_y/V_x & 1 & 0 \\ -V_z/V_x & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{K} = \begin{bmatrix} 1/V_x & 0 & 0 \\ -k(fV_y + p_vV_z)/V_x & kf & kp_v \\ -kV_z/V_x & 0 & k \end{bmatrix}$$

As the u coordinate does not undergo a perspective projection the u coordinate is not scaled when transformed by camera matrix $\boldsymbol{M}$. Therefore, only element wv needs to be obtained by division by w.

$$\begin{bmatrix} u \\ wv \\ w \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

2.4.4 Model estimation

The most common and well-known method for camera model estimation is the direct linear transform (DLT). This method is widely applied for pinhole cameras. As the linear pushbroom model has the same form as the pinhole camera model, the DLT can be applied with little modification. The DLT treats the camera matrix $\boldsymbol{M}$ as consisting of unknown elements $m_{11} \cdots m_{34}$ which must be estimated to preserve the transformation between points known in the world and point in image coordinates.

$$\begin{bmatrix} u \\ wv \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

The underlying principle is that the relationship between image coordinates and world coordinates is a linear relationship and therefore can be solved simultaneously if enough image and world coordinates are known. However, any real system will have a non-zero error in error of pixel coordinates and or world coordinates which would lead to an inconsistent set of linear equations and therefore cannot be solved simultaneously. The DLT instead implements a linear least squares solution by solving the normal equations or inspecting the singular value decomposition of the equations. The linear relationships between the rows of $\boldsymbol{M}$ and $\boldsymbol{x}$:

$$u = \boldsymbol{m_1 x} = [m_{11} \quad m_{12} \quad m_{13} \quad m_{14}][x \quad y \quad z \quad 1]^T$$

$$wv = \boldsymbol{m_2 x} = [m_{21} \quad m_{22} \quad m_{23} \quad m_{24}]\,[x \quad y \quad z \quad 1]^T$$

$$wv = \boldsymbol{m_2 x} = [m_{31} \quad m_{32} \quad m_{33} \quad m_{34}]\,[x \quad y \quad z \quad 1]^T$$

$$wv = \boldsymbol{m_2 x}$$

$$w = \boldsymbol{m_3 x}$$

For N number of corresponding world and image coordinates, the set of simultaneous equations can be written as $\boldsymbol{Aq} = \boldsymbol{b}$, where $\boldsymbol{q}$ is the 12x1 vector containing the unknown elements of $\boldsymbol{M}$. $\boldsymbol{b}$ is the 2Nx1 vector containing the alternating u and v pixel coordinates of points in the image. Each column of the 2Nx12 matrix $\boldsymbol{A}$ can be conceptualised as vectors spanning 2xN dimensional space, then $\boldsymbol{q}$ can be conceptualised as the weighting of each of these vectors which have to be summed to produce vector $\boldsymbol{b}$**.** As the last two rows of $\boldsymbol{M}$ are free scaling and the scale factor can be recovered later, m_{34} can be set to be equal to one which simplifies the solution of v, and the formulation of $\boldsymbol{A}$.

$$\boldsymbol{Aq} = \boldsymbol{b}$$

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & y_1 & 1 & -v_1x_1 & -v_1y_1 & -v_1z_1 & 1 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & x_N & y_N & y_N & 1 & -v_Nx_N & -v_Ny_N & -v_Nz_N & 1 \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ \vdots \\ m_{34} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ \vdots \\ v_N \end{bmatrix}$$

Consider that column vectors of A span a hyperplane. If there were no error in the pixel and world coordinate points then the vector b would lie in this hyperplane, and it would be a simple matter to determine the linear combination of vectors ($\boldsymbol{q}$) that are summed to form $\boldsymbol{b}$. In this case $\boldsymbol{q}$ can be determined as:

$$\boldsymbol{q} = \boldsymbol{A^{-1}}\,\boldsymbol{b}$$

However, due to pixel error vector b will lie outside of the hyperplane spanned by $\boldsymbol{A}$. In this case the least squares solution can be obtained by projecting vector b onto the hyperplane. Hence $\boldsymbol{q}$ is solved for as:

$$\boldsymbol{q} = (\boldsymbol{A^T A})^{-1}\,\boldsymbol{A^T b}$$

One $\boldsymbol{q}$ has been attained, its elements can be written back into the camera matrix $\boldsymbol{M}$.

2.5 POINT CLOUD ACQUISITION FOR PHENOTYPING

Measuring the morphology of plants has been a common and important activity within plant phenotyping. Often this is employed on a two-dimensional basis, using 2D images to capture 2D measures of the plants morphology. Example include Arvidsson, Pérez-Rodríguez, and Mueller-Roeber (2011) who developed a high throughput phenotyping platforms which are able to measure leaf area of *Arabidopsis Thanalia*. Two dimensional methods suffer from occlusions, hence, 3D acquisition techniques such as LiDAR, use of depth cameras (Keightley and Bawden 2010, Chéné et al. 2012) . Moreover, there has been a trend of fusing of point clouds with hyperspectral images. Aasen et al. (2015) used a snapshot hyperspectral camera mounted on a quadcopter for capturing 3D models using the structure from motion principle. Roscher et al. (2016) used a high precision laser scanner for capturing point clouds which were fused with hyperspectral images and used for the early disease detection in plants.

2.5.1 Laser-based point cloud acquisition for phenotyping

Laser scanners are an active sensing technology. A beam of light/radiation is emitted, generating either a single point or a line which is projected on a surface, and observed by a camera. If the geometry of the emitter and camera are both known as well as their position in space, points on the object coincident with emitted laser light can be triangulated (Sabbadin 2016). Moving the laser over the object allows the geometry of the object to be triangulated and accumulated into a point cloud. Many different technologies have been used for acquisition of point clouds or meshes within the context of plant phenotyping. Hosoi, Nakabayashi, and Omasa (2011) used LiDAR to measure inclination angles of leaves, while Klose, Penlington, and Ruckelshausen (2009) performed a usability study on time of flight cameras for morphological measurements in robotic field based phenotyping. In the domain of close range phenotyping Laser triangulation scanners provide the greatest resolution and accuracy Paulus, Schumann, et al. (2014) demonstrated the use of the perceptron V5 scanning system for acquisition of barley organs on a sub-millimetre scale, and the measurement of the entire plant every 2-3 days for growth monitoring. With an accuracy of 45µm, and a resolution of 17µm, Paulus, Schumann, et al. (2014) noted that no other multi view optical method had demonstrated a quantitative investigation of more complex plant organs, making this the most precise multi view optical method reported in the phenotyping literature.

The utility of laser scanning for phenotyping has been realised on a commercial scale with the development of the PlantEye by Phenospex. The sensor was developed specifically for commercial high- throughput phenotyping with special emphasis on acquiring quantitative information on morphological parameters such as leaf area, leaf inclination, and digital biomass. It can achieve sub millimetre precision ~0.8mm in all directions at a working depth of 120cm. The PlantEye has successfully demonstrated utility for the phenotyping of plants in challenging environments with high temporal resolution (Kjaer and Ottosen 2015).

The main difference between the laser scanner and other techniques is that the correspondence problem is significantly simplified. As opposed to searching though many potentially ambiguous corresponding points as in photogrammetric techniques, only pixels illuminated by the laser line need to be searched. This improves both the speed and robustness of reconstruction, and quality of resultant point clouds when compared with photogrammetric techniques. The projected laser line removes the dependence on weak textural features of vegetation for solution of the pixel correspondence problem.

The major disadvantage of lasers scanning approaches, is that the hardware is very expensive. Wang et al. (2017) noted that one important trend required for phenotyping is the adoption of low cost 3D acquisition technologies that can be afforded by common users, and suggests that high cost techniques such as LiDAR and Laser scanning cannot satisfy this demand. Paulus, Behmann, et al. (2014) evaluated common lower cost technologies including the popular low cost DAVID laser scanner and Microsoft Kinect over a number scenarios, and found that they could produce comparable results with high cost laser scanners for a number of important phenotyping morphological and volumetric measurements. Paulus, Behmann, et al. (2014) concluded that lower cost, lower resolution technologies can be used in some phenotyping situations for a fraction of the cost of high precision laser scanning techniques.

2.5.2 Photogrammetry-based point cloud acquisition for phenotyping

Photogrammetric techniques are commonly reported in the literature for use in plant phenotyping. The most common of these techniques are calibrated stereo, and the structure from motion principle. For example Mizuno et al. (2007) used stereo vision for wilt detection in indoor environments; Yoon and Thai (2010), used stereo vision with NDVI index to characterise plant health in indoor conditions. Paproki et al. (2012) used structure from motion and a DSLR camera to generate meshed plant models, which were segmented into plant organs such as leaves and stems for the automatic measurement of morphological parameters.

With calibrated stereo vision, the baseline and orientation of both cameras are already known. The processing pipeline requires rectification (images resampled as if they were taken by cameras with co planar image planes), identification of corresponding pixels, and calculation of disparity images. The disparity image can then be converted into a point cloud by scaling of disparity values and projecting the pixels into the scene through the focal point of the camera at a radial distance equal to the scaled disparity value.

Structure from motion on the other hand does not have information about the position and orientation of the camera when the images were taken. Salient points in multiple images can be detected with SIFT features which are scale and rotation invariant. An initial pair of images which a high number of sift features are used to estimate the position of the first two cameras, following which more images and cameras are added to produce a low-density reconstruction and estimate the relative poses of the camera. This can be refined with a bundle adjustment. Once the relative poses of all cameras are known, the calibrated stereo algorithm can be applied for each pair of images building up a dense point cloud.

The correspondence problem refers to the problem of identifying corresponding points in both images. The underlying principle is that corresponding points in both images will have similar neighbourhoods of pixels as they view the scene from similar perspectives, meaning local textural patterns in the scene will be present in both images. Every stereo vision algorithm makes use of a matching cost function to determine the correspondence between two pixels. The most common algorithms calculate absolute differences, squared intensity differences, and normalised cross correlation. MATLAB implements this process through "block matching". For each point in the first image, the algorithm evaluates the similarity of potentially matching blocks in the second image. A similarity metric: sum of absolute differences (SAD) is calculated by aggregating the absolute difference between the two blocks. Blocks with a minimal SAD value are corresponding.

The main advantage of the stereo and structure from motion techniques is that they are relatively low-cost and can be employed with simple low-cost cameras. However, there are many disadvantages, these include being computationally demanding, have low acquisition rates, and performance is adversely affected by lack of surface texture and specular highlights. Moreover, the stereo matching process is poor for close range observation of leaf surfaces as the homogenous texture will produce pockets of missing depth information (Kazmi et al. 2014). Furthermore, while the calibrated stereo technique can produce real scale point clouds, structure from motion can only determine the point cloud up to an ambiguous scale factor. Scale information must be recovered by comparison of the cloud with objects with geometrical properties that are known in the world coordinate system.

2.5.3 Alternative approaches of point cloud acquisition for phenotyping

The Microsoft Kinect sensor was developed for the gaming industry to track body position and movement (Azzari, Goulden, and Rusu 2013). The Microsoft Kinect sensor operates on the structured light principle. A predefined pattern of dots is projected onto the scene by an infrared projector. An infra-red camera records the position of the dots on the scene which are then compared with factory calibrated reference pattern to produce a disparity image from which a point cloud can be extracted.

The Microsoft Kinect has gained popularity with the computer vision and robotics communities, often being used for applications such as simultaneous localisation and mapping of robots. However, there have been some studies that have used the Kinect for assessing the morphology of plants. These include: Azzari, Goulden, and Rusu (2013) who used the Microsoft Kinect for characterising the structure of vegetation and its canopy structure. Paulus, Behmann, et al. (2014) investigated the Microsoft Kinect as a low cost alternative to expensive laser scanners, and found that the Microsoft Kinect could produce comparable results for measuring the volume of vegetables, and measuring leaf area. Shah et al. (2016) mounted the Microsoft Kinect on a mobile robot platform for monitoring plants within a growth chamber, detecting and segmenting plants in a multi-plant environment.

The accuracy of the depth image is mainly dependent on the sensor to object distance (Mankoff and Russo 2013). The standard deviation ranges from a few millimetres to 4cm over the distance range of the sensor (Khoshelham and Elberink 2012), with a point resolution of approximately 0.9 to 7mm (Paulus, Behmann, et al. 2014). The main advantage of the Microsoft Kinect is that is can capture point clouds in real time 30Hz. While the main disadvantage of the Microsoft Kinect is its relatively low resolution. However, real time reconstruction algorithms such as Microsoft Kinect fusion can co-register the continual stream of point clouds into one cloud with superior detail. The vast data volume of point clouds allows the identification and removal of outliers, as well as the averaging and refinement of the registered views which decouples the random error and resolution from the sensor intrinsic properties.

3 STUDY SETUP AND METHODS

3.1 EXPERIMENTAL SETUP

Three groups of experiments are conducted within this thesis (detailed in subsequent chapters):

1. Experiments to assess the efficacy and suitability of geometric camera calibration.

2. Experiments to assess the suitability and potential of low resolution point clouds for illumination correction of hyperspectral images.

3. Experiments to assess efficacy of illumination correction of hyperspectral images by modelling illumination geometry with low resolution point clouds. These experiments require both hyperspectral images, and corresponding point clouds of the same scene.

To facilitate this range of experiments two experimental setups were used. The first setup which is detailed in section 3.1.1 is used for both experiments 1 and 3. Experiment 2 was conducted using an entirely different setup which is detailed in section 3.1.2.

3.1.1 Setup for Camera calibration and Illumination correction experiments

The major components of the setup (Fig. 3.1) included an AisaFENIX hyperspectral camera, ART scanner linear platform, 6 tungsten halogen lights attached to the linear frame, matt black matt used as background, white reference panel, custom manufactured calibration gauge, custom made planar and cylindrical reference objects, digital cameras, Microsoft Kinect, and AisaFENIX DAC.

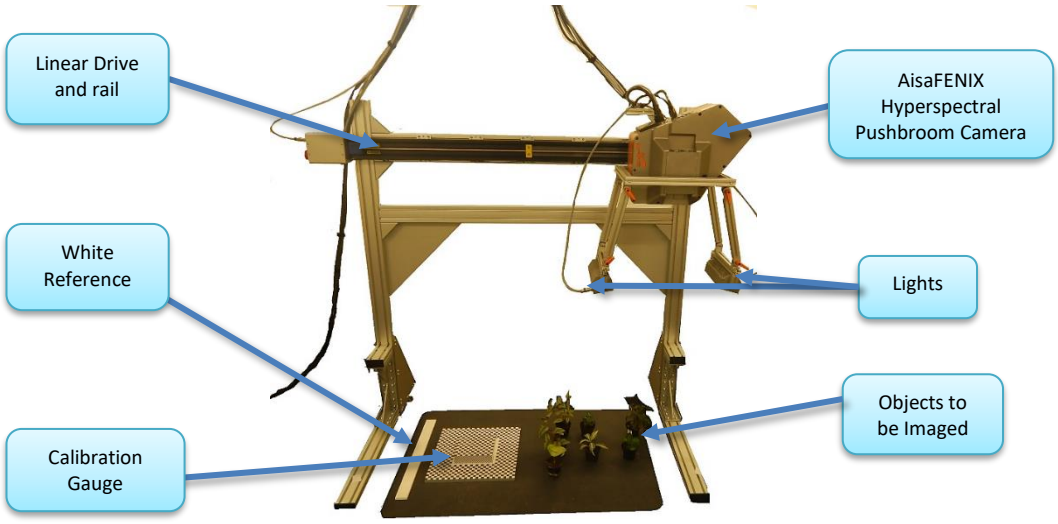

Figure 3.1 - Hyperspectral Camera Setup

3.1.2 Setup for point cloud assessment experiments

All objects to be imaged were placed at the centre of a circular table with a radius of approx. 0.5m. The perimeter of the table was marked with an equal division of 5.5 degrees. Structure from motion images could be captured with a tripod mounted camera, while the Microsoft Kinect captured point clouds were captured by unconstrained camera motion over the scene using SLAM reconstruct me software. The front two legs of the tripod when coincident with the perimeter of the table would keep the facing the centre of the table and keep the camera distance to the centre of the table relatively constant (Fig3.2).

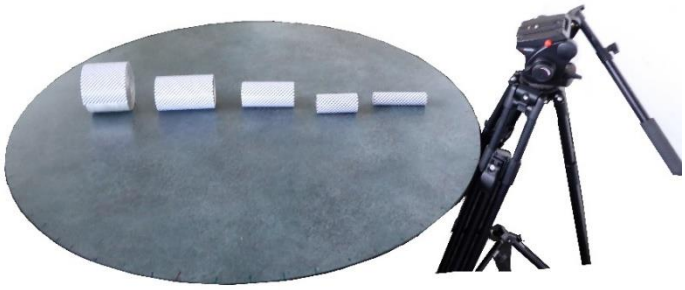

Figure 3.2 Point Cloud Assessment Experimental Setup

3.1.3 AisaFENIX, DAC, ART Scanner

The AisaFENIX Pushbroom hyperspectral camera acquires spectral data in the range of 380 – 2500nm, with a maximum spectral resolution of 620 bands. It uses both VIS/NIR Complimentary MOSfet (CMOS) and SWIR Mercury Cadmium Telluride (MCT) detectors which automatically provide VNIR and SWIR images that are co-aligned into a single data cube. The 32.3° FOV at approximately 1.3 m produces a ground swath width of ~750 mm. With a spatial resolution of 384 pixels, the Ground-Instantaneous-Field-Of-View (GIFOV) is approximately 1.9×1.9 mm per pixel.

The Data-Acquisition-Computer (DAC) is custom made by Specim for the control of their hyperspectral cameras. LUMO SCANNER software running on the DAC controls both the acquisition parameters of the AISA Fenix camera, and the motion of the ART Scanner linear platform.

The ART Scanner platform is an aluminium frame with linear actuator. The linear actuator can move the mounted camera and lights over the scene at a constant, user-defined velocity. Two sets of three 50W watt tungsten halogen bulbs were used to provide illumination for the scene. As they are attached to the linear stage their geometry remains fixed with respect to the camera frame while linear stage traverses the scene. A standard white reference panel supplied by Specim Ltd. is a diffuse reflector across 400-2500nm range. This is

placed in every image so that the hyperspectral images can be radiometrically calibrated, and reflectance can be determined.

3.1.4 Calibration gauge

The calibration gauge(Fig3.3) was custom made from precision machined aluminium blocks, with aligned checkerboard patterns printed on paper. The design is based on the reference gauge used by Behmann et al. (2016). The checkerboard patterns on the base and top blocks provide corner points that can be semi-automatically detected using the MATLAB Harris corners routine. The DLT calibration camera model estimation requires that the input coordinates be supplied over a range of (height) values to improve the quality of the model estimation. Hence, the base and top planes of the calibration gauge were manufactured to be 40.00 ± 0.02mm apart.

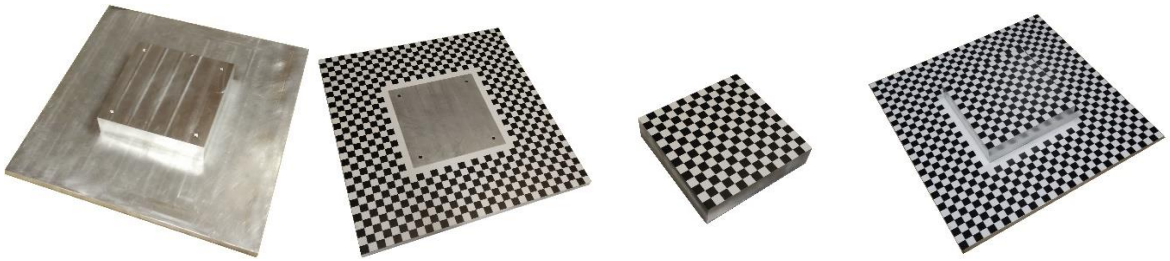

Figure 3.3 Calibration Gauge in Various Stages of Manufacture

While the aluminium components could be precisely manufactured to specification, and the checkerboard patterns could be printed with an accuracy of 1200dpi or approximately 0.02mm, gluing the checkerboard patterns onto the aluminium blocks had to be aligned by eye. To mitigate any misalignment by the gluing process the aluminium blocks were bolted together meaning that they could be adjusted until aligned and then finally bolted when the patterns are aligned.

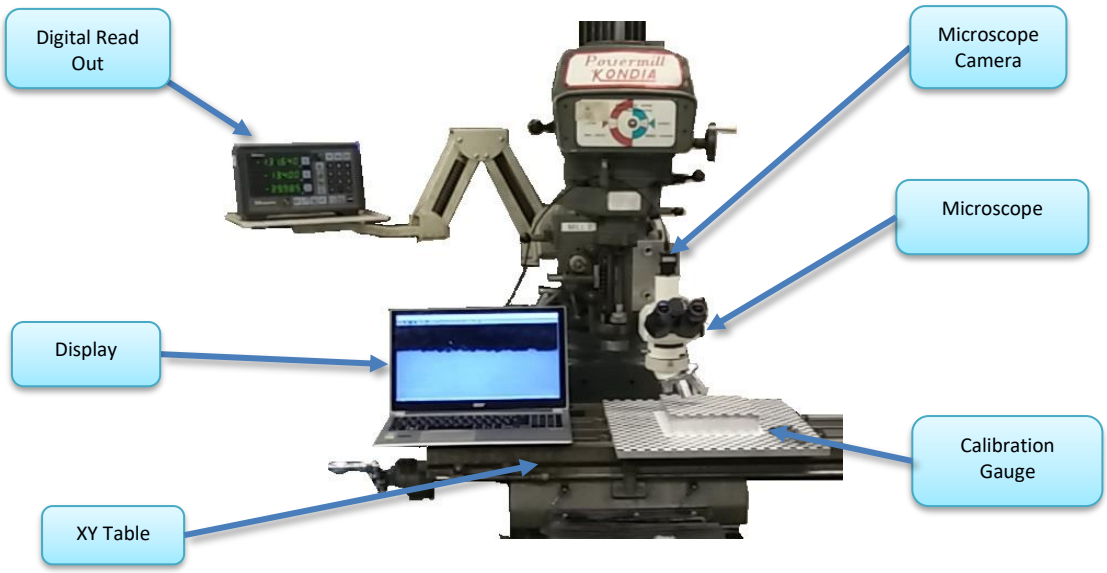

Figure 3.4-Verification of Accuracy of Calibration Gauge on Milling Machine XY Table

To validate the accuracy of the reference gauge, it was mounted to a milling machine XY table which allowed precise movement increments of 0.005mm over a 400mm range(Fig.3.4). A 10x microscope was mounted rigidly to the body of the milling machine, providing magnified images that were captured by a C mount IDS uEye camera, and displayed on a screen. The location of coordinates on both top and bottom planes were captured by moving the XY table so that the corner under measurement was centred with respect to the calibrated microscope. Analysis of the microscope images confirmed that this the corners were centred with an accuracy of better than 50µm. Arriving at a set of coordinates for both top and base planes in the coordinate system of the base plane allowed for estimation of worst case horizontal alignment error. This was verified to be less than half a pixel at 0.83 mm. The worst case alignment error (with respect to the base plane) would be observed in one of the outermost corners of the top plane. This can easily be found be determining the relative horizontal rotation and translation between the planes, and then computing the locations of the four top plane outermost corners, of which one will have the maximal misalignment.

3.1.5 Calibration objects

Calibration objects of known size and shape are used in both experiments set 2 (assessment of points clouds) and set 3 (assessment of illumination correction efficacy). In experiment set 2, point clouds captured by both SFM and the Microsoft Kinect technologies are evaluated against planes with known angles, and cylinders with known radius. Texture is also a variable of interest. A set of five wooden blocks were manufactured with plane angles of 0°, 11.5°, 21.5°, 31.3°, 44.6° (referred to as A0,A10,A20,A30,A45 respectively) with respect to the ground(Fig.3.5). These angles were computed from a set of height measurements between the top and bottom edges of the planes. There was maximum height error of 0.3 mm determined by taking repeat

measurements. The maximum angle error assumes that one edge is measured with an error of +0.3mm, while the other is -0.3mm. As the width of the regular block is 100mm the maximum angle error of the plane is 0.34° ($\tan^{-1}(0.3*2/100)$), and for A45 it is 0.17° ($\tan^{-1}$ ((100 + 0.3*2) /100) - 45°).

Figure 3.5-Calibration Blocks A45, A30, A20, A10, A0, with Texture T5

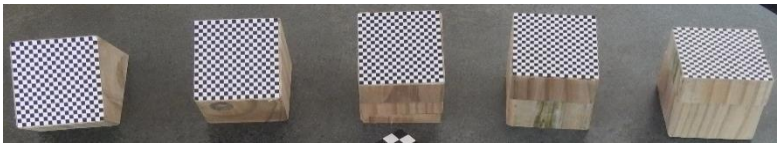

Figure 3.6-Calibration Blocks with Texture T4

The cylinder wrapped with the printed checkerboard patterns with spatial frequencies of 0.55,1.1,2.2,4.4, and 8.8 mm per square and are referred to as textures T1,T2,T3,T4,T5. These patterns can be interchanged between blocks. A T3 pattern was also printed onto matte brown paper which had a lower specular reflectance than white paper.

Cylinders were made from round aluminium bars with radii of 15,24,34,45,and 63 mm, which will be referred to as C1,C2,C3,C4,C5. These cylinders were also coated with checkerboard patterns (Fig.3.7).

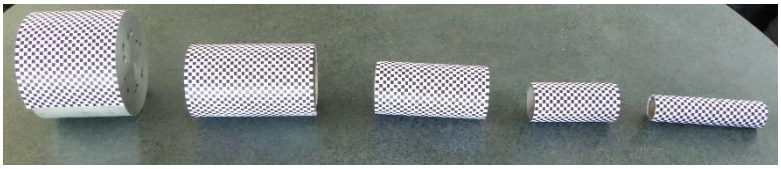

Figure 3.7 Cylinders C5, C4, C3, C2, C1 with Texture T4

The naming shorthand for referring to experiments involving calibration objects begins firstly with a designator for point cloud acquisition technology used; S for SFM, while K is used to indicate the Microsoft Kinect. Either an P or a C will be used to indicate the planar objects or cylindrical objects respectively. In the case of cylindrical objects, a number will immediately follow the C. In the case of a planar object the A at the end stands for angle and will be immediately followed by a number indicating which angled plane is referred to. For both the cylinder and plane objects, designator T indicates the textural frequency of the pattern used. For example:

SPT3A30 – refers to the experimental results acquired using SFM technology, for a planar object with texture of 2.2mm/square angled at 31.3° with respect to the ground plane. KC1T5 – refers to the cylinder of radius 15mm, imaged with the Microsoft Kinect, with textural frequency of 8.8mm per square. SP refers to all planes imaged by the SFM technique, while KCT4 refers to cylinders of all radii, with texture 4.4 mm/square, that are imaged by the Microsoft Kinect.

3.2 HYPERSPECTRAL IMAGE ACQUISITION

The hyperspectral images were acquired using LUMO software provided by Specim. The images were acquired with a frame rate of 20Hz, with spectral binning of 4 in the VNIR range and 2 in the SWIR range with an integration time of 48 ms in the VNIR range, and an integration time of 10 ms in the SWIR range. This results in a total of 448 contiguous spectral bands between 377.9 nm to 2497.3 nm. The linear velocity of the camera was set to approximately 40mms^{-1} to produce a pixel aspect ratio that was roughly 1:1. Closing the internal shutter of the camera allowed for the capture of a dark frame image for each image of the scene.

The list of images captured and used:

- 10 × Images of the reference gauge and white reference only, used for the camera calibration assessment
- 1 × Image of the reference planes, calibration gauge and white reference used for illumination correction assessment
- 1 × Image of the reference cylinder, calibration gauge, and white reference for illumination correction assessment.

3.3 Point Cloud Acquisition

Both the Microsoft Kinect and RGB camera were used to obtain pointclouds of the scenes. The list of point clouds is divided by the experiments. For the assessment of meshes and point clouds the following point clouds were acquired:

- 3 × repeats of T1,T2,T3,T4,T5 for planes A0 through A45 captured for SFM technology (15 point clouds from 1485 images)
- 5 × repeats of T1,T2,T3,T4,T5 for planes A0 though A45 for the Microsoft Kinect (25 raw point clouds)
- 3 ×repeats of T4 for cylinders C1 through C5 captured with the SFM technology(3 point clouds captured from 297 images)
- 5 × repeats of T4 for cylinder C1 through C5 captured with the Microsoft Kinect (5 point clouds)

For the illumination correction assessment experiments the point clouds were captured for the purposes of illumination correction. Based on the findings in the sections 5.1.2, 5.1.3, 5.3.2 and 5.3.3 the Microsoft Kinect wasn't used for any further illumination correction experiments. The following point clouds were captured for the illumination correction experiments:

- 1 × repeats of the setup capturing the position of the six halogen lights with respect to the reference gauge at the start of the camera motion (1 point cloud obtained from 291 images)
- 1 × repeats of T3 matte brown cylinders C1 through C5 with the reference gauge (1 point cloud 216 images)
- 1 × repeats of T3 matte brown planes A0 through A45 with reference gauge (1 point cloud from 139 images)

3.3.1 SFM Acquisition

Two different cameras were used to acquire structure-from-motion image dataset. For the illumination correction experiments images were acquired using a Panasonic DMC-TZ60 camera with image size of 4396x3672 pixels. The images of the planes, cylinders, and setup were acquired without a regular camera motion pattern. Due to the performance of the T3 texture as outlined in 5.3.2 and 5.3.3 an effort was made to decrease the camera distance from its target so that the texture observed would not be too fine for adequate reconstruction.

In the case of the mesh assessment experiments, images of size 4608x3456 pixels were acquired with Nikon Coolpix AW130 camera. Images were taken in a regular semi-circular pattern at 3 different heights from 33 equally spaced locations around the perimeter of the round table. The images were processed using the Visual structure from motion tool (Wu 2013, 2011), which first estimates the positions of the cameras based on SIFT feature matching, and then reconstructs a scale incorrect dense point cloud of the scene. The software took approximately 30 minutes for dense reconstruction on a HP Z220 desktop with i7 processor, and NVIDIA Quadro2000 GPU. Fig.3.8 and Fig.3.9 show the dense reconstructed point cloud and estimated camera positions for one of the plane experiment repeats. All SFM images were acquired under ambient lighting conditions.

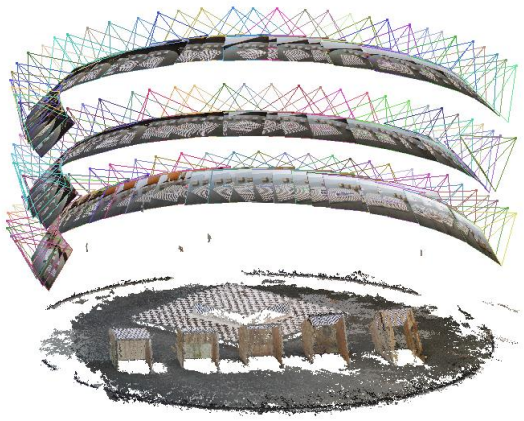

Figure 3.8 SFM Dense Reconstruction Top View

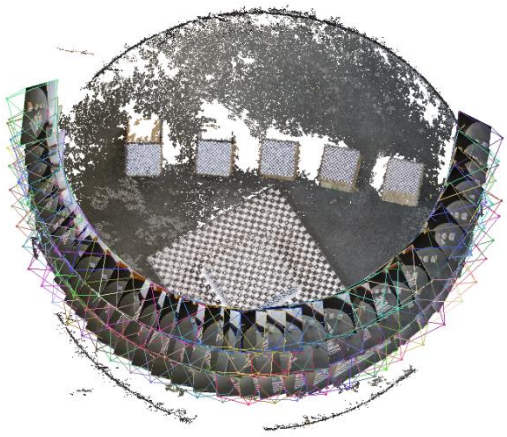

Figure 3.9 SFM Dense Reconstruction Back View

3.3.2 Microsoft Kinect point cloud acquisition

A free software package, ReconstructMe, used by (Paulus, Behmann, et al. 2014) performs a simultaneous localisation of mapping (SLAM) as the Microsoft Kinect is moved over the scene. Exploiting the 30Hz data rate of the Microsoft Kinect the unpublished algorithm can eliminate outliers and produce a dense, uniformly sampled, meshed point cloud. A Microsoft Kinect version 1414 which was shown by (DiFilippo and Jouaneh 2015) to provide more accurate measurements than other models was used in these experiments. The Microsoft Kinect was warmed up for over 60 minutes as recommended by (Chow et al. 2012) who showed that its accuracy is temperature dependent, and performed all data acquisition under ambient illumination conditions. The ReconstructMe software was configured to prioritise accuracy over acquisition speed. Due to lack of advanced acquisition setups such as automatic turntables, the Microsoft Kinect was moved over the scene manually. (Alnowami et al. 2012) showed that the Microsoft Kinect can achieve mm precision at a range of 0.8-1.5 m. This was consistent with the ReconstructMe software (Heindl et al. 2015), which only used points acquired within this or a similar range.

4 METHODOLOGY

4.1 PRE-PROCESSING

4.1.1 Hyperspectral image pre-processing

CaliGeoPro software was used to both remove bad pixels (due to malfunctioning detectors) within the raw hyperspectral images, subtract the dark frame to account for pixel offset, and values were converted from digital numbers to radiance using factory supplied calibration parameters. Subsequently, ENVI software was used to orient the images so the positive u-axis (horizontal axis) corresponds to the along track direction of the sensor.

4.1.2 Point cloud pre-processing

Both the Microsoft Kinect point clouds and structure from motion clouds were manually cleaned by removing any irrelevant objects present, such as the floor, or walls of the room. The point cloud captured by the structure from motion technique produced spurious clusters of points (misregistration artefacts) which were also removed. The point clouds generated by structure from motion were further meshed with a Poisson mesh using MeshLab software.

4.2 OVERVIEW

There are three main processes involved in the total illumination correction process (Fig4.1):

- *Geometric camera calibration* - this set of algorithms takes both hyperspectral image as input and user defined initial search regions. From these inputs the algorithms determine the pushbroom camera model (encoding the image formation geometry and relationship between the camera and world coordinate frames), and the geometric image distortion. It outputs both the pushbroom camera model, and a geometrically undistorted data cube.

- *Point cloud alignment routine* - which estimates a rough transformation between the input cloud and world coordinate system, then applies the ICP routine to finely align the point cloud/meshed model with the world coordinate frame.

- *Illumination correction routine* - which employs a ray tracing algorithm to determine the effects of illumination and geometry on the measured reflectance, and then correct for these, outputting an illumination corrected image. It requires a geometrically rectified hyperspectral data cube, pushbroom camera model, aligned meshed points cloud, and model of light source geometry as inputs.

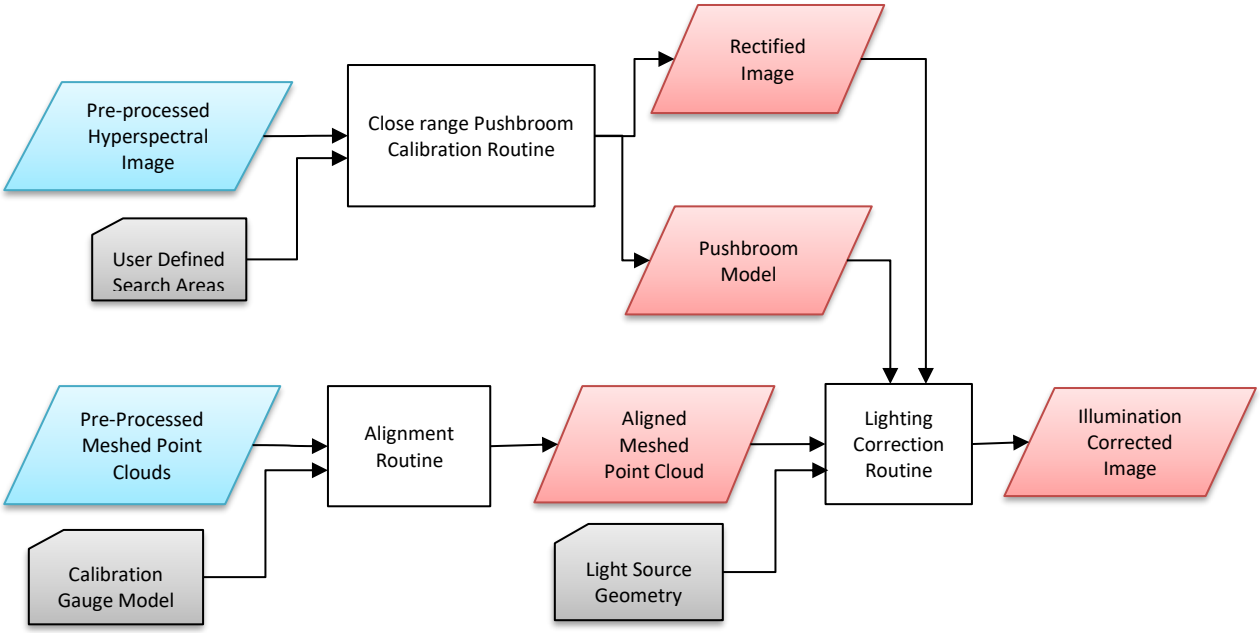

Figure 4.1 – Process for Illumination Correction

This chapter is structured so that each section outlining a module of illumination correction is immediately followed by a section detailing methodology for assessment of its efficacy.

4.3 PUSHBROOM MODEL ESTIMATION AND CAMERA CALIBRATION

The pushbroom model estimation and calibration routine uses the linear pushbroom model of Gupta and Hartley (1997) , and is based on the routine of Behmann et al. (2015) with minor deviations. The core of processing routine can be roughly broken down into three main sections: initial camera model estimation ,non-linear effect estimation and correction, camera model re-estimation and parameter retrieval.

4.3.1 Initial camera model estimation

As previously outlined in the literature review, the linear pushbroom model can be determined from at least eight image/world correspondences using a least squares estimation. In our implementation we use 12 correspondences for improved resilience to noise. The 12 coordinates known in the world frame are picked over the range of the reference gauge. The correspondences in the image coordinate system are determined semi-automatically. The user first defines the approximate location of the correspondences, which are further refined to sub pixel accuracy using a MATLAB implementation of the Harris corner detector. The Harris corner detector is popular within the computer vision community as it can robustly detect corner points in a manner that is scale, rotation, and illumination invariant (Derpanis 2004). While it can still detect corners over varying illumination conditions, the accuracy of the detector is affected by the across track vignetting effect (decrease in intensity towards the top and bottom of the image) present in the raw hyperspectral images, which ultimately caused noticeable errors in the camera model estimation (See Fig4.2). Vignetting is remedied by scaling each of the image rows so that the intensity values are homogenous across the white reference (Fig.4.3). The 12 correspondences are used to compute the 3x4 camera matrix $\boldsymbol{M}$.

4.3.2 Non-linear effect estimation and correction

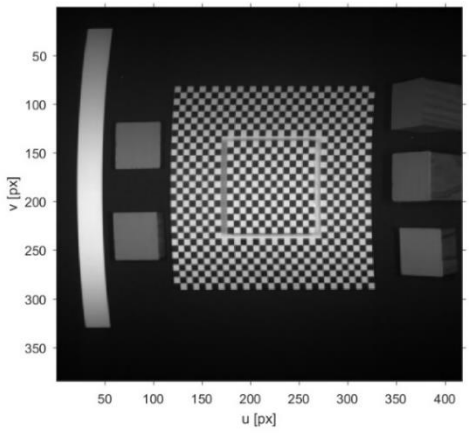

Figure 4.2 – Raw Hyperspectral Image (700nm)

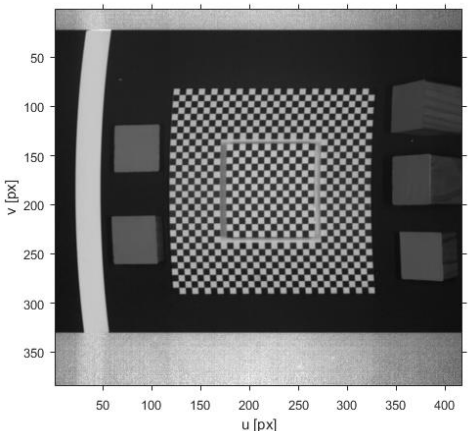

Figure 4.3- Removal of Vignetting by Row Based Scaling

The raw hyperspectral images (Fig.4.2) suffer from the effects of geometric distortions. Without correction for non-linear effects the linear mapping between world and image space $\boldsymbol{M}$, may encode an erroneous camera model. Common non-linear effects such radial or tangential distortions are typical in pinhole model cameras and have been thoroughly studied and characterised. In their review of pushbroom camera calibration routines, Behmann et al. (2016) concludes that in the context of close range phenotyping, a phenomenological approach that does not make assumptions about the nature of the geometric distortion were able to better model the non-linear effects and correct the images with a superior accuracy. The phenomenological approach of Behmann et al. (2016) firstly computes the error at each of the coordinate of the reference gauge over the base plane internal corners, following which an polynomial is fitted to model the error.

For computation of the error, all 672 internal corners in the image (Fig.4.6) are found using a semi-automatic method. The centre of search regions are computed by projecting the 672 corresponding world coordinates to the image plane using the initial camera model $\boldsymbol{M}$. In the case of very strong distortion effects, the search region may not coincide with its corresponding found Harris corner point. This is mitigated by a rough approximation of the effects from 3 user defined points, and laterally shifting the search regions so they are centred on the corresponding found Harris corner (Fig.4.5). When multiple Harris corner points are found within one search space, corners with lesser confidence are rejected.

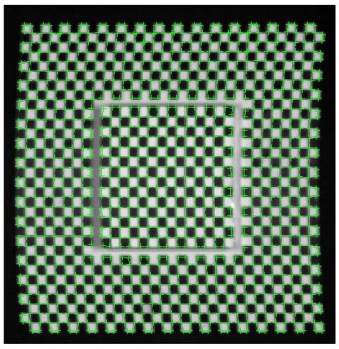

Figure 4.4 – All Found Harris Corners

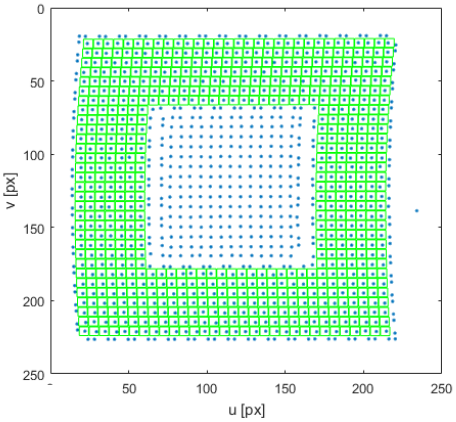

Figure 4.5 – Automatically Determined Search regions

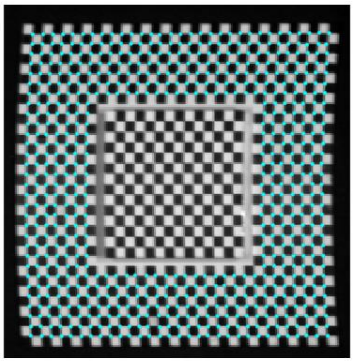

Figure 4.6 – Internal Base Plane Corners

Non-linear effects can be observed in the re-projection error $\boldsymbol{e}$ ie: the difference between a pixel location $\boldsymbol{x_{img}}$ and the estimated location based on the projection of the corresponding base plane coordinate $\boldsymbol{x_w}$ in the world 3D coordinate system by the projection matrix $\boldsymbol{M}$.

$$\boldsymbol{e} = \boldsymbol{x}_{\text{img}} - \boldsymbol{M}\boldsymbol{x_w}$$

Behmann et al. (2015) fitted two dimensional 3rd order Chebyshev polynomials $P_u(u, v), P_v(u, v)$ to the reprojection error, which model the error in the u and v directions as a function of the distorted images u, v coordinates. This is achieved through a least squares fitting.

$$P(u,v) = a_0 + a_{10}u + a_{01}v + a_{11}vu + a_{20}(2u^2 - 1) + a_{02}(2v^2 - 1) + a_{12}v(2u^2 - 1) + a_{21}(2v^2 - 1)u + a_{30}(4u^3 - 3u) + a_{03}(4v^3 - 3v)$$

An important note is the relevance of forward, and inverse mappings. The forward mapping allows the calculation of new image coordinates (u_{new}, v_{new}) as a function of the original image coordinates (u_{or}, v_{or}). Using polynomials $P_u(u_{or}, v_{or})$, and $P_v(u_{or}, v_{or})$, the forward mapping can be expressed as:

$$\begin{bmatrix} u_{new} \\ v_{new} \end{bmatrix} = \begin{bmatrix} P_u(u_{or}, v_{or}) \\ P_v(u_{or}, v_{or}) \end{bmatrix} + \begin{bmatrix} u_{or} \\ v_{or} \end{bmatrix}$$

While this is fine for computing the transformation of points from the original location to their new locations, it is not suitable for resampling an image as it may leave gaps in the resampled image. The solution to this problem is to use an inverse mapping which specifies the mapping between points in the new image, and original image as a function of (u_{new}, v_{new}). This mapping can be computed by defining the inverse error $\boldsymbol{e_{inv}}$ as:

$$\boldsymbol{e_{inv}} = \boldsymbol{M x_w} - \boldsymbol{x_{img}}$$

Fitting polynomials $P_{uinv}(u_{new}, v_{new}), P_{vinv}(u_{new}, v_{new})$ to the error allows for an inverse mapping where a point in the new images corresponding point in the original images is expressed as:

$$\begin{bmatrix} u_{or} \\ v_{or} \end{bmatrix} = \begin{bmatrix} P_u(u_{new}, v_{new}) \\ P_v(u_{new}, v_{new}) \end{bmatrix} + \begin{bmatrix} u_{new} \\ v_{new} \end{bmatrix}$$

Typical resampling of images involves interpolation of pixel values. In this case Behmann et al. (2015) recommend using a nearest neighbour assignment as opposed to bi-linear, or bi-cubic interpolation as it may result in implausible pixel spectra.

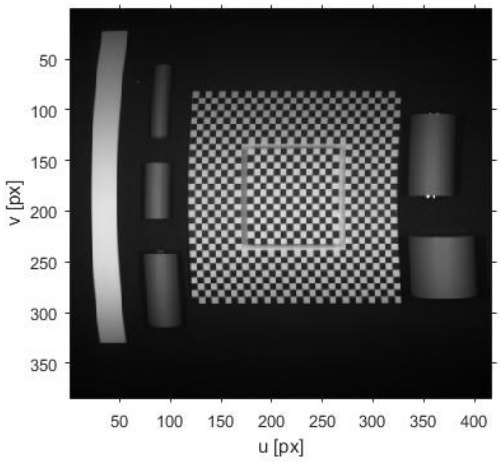

Figure 4.7 – Raw Hyperspectral Image (700nm)

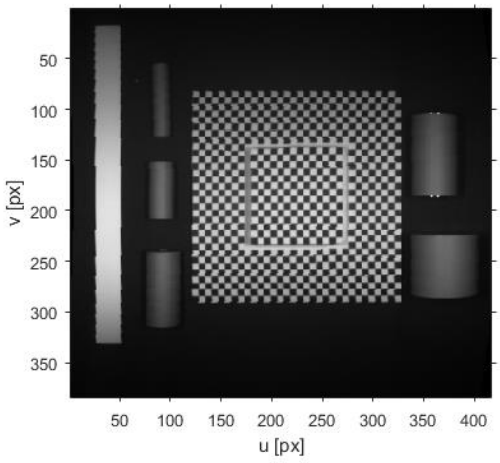

Figure 4.8 – Geometrically Rectified Image

Fig.4.8 and Fig.4.9 show the original distorted hyperspectral 700nm band, and its geometric undistotion using Chebyshev polynomial modelling of non-linear effects, and a resampling based on an inverse mapping.

4.3.3 Re-estimation of $\boldsymbol{M}$ and parameter retrieval by RQ decomposition

The initial camera model estimation is refined by undistorting the 12 original image coordinates using the forward mapping. These coordinates along with corresponding world coordinates are used to recompute the camera matrix $\boldsymbol{M}$.

The subsequent illumination correction stage requires access to both the extrinsic and intrinsic camera parameters. Parameter $\boldsymbol{T_{wc}}$ is the first to be extracted from matrix $\boldsymbol{M}$. In the literature review we formulated matrix $\boldsymbol{M} = \boldsymbol{K}[\boldsymbol{R_{cw}}|\boldsymbol{T_{cw}}]$. However, as $\boldsymbol{T_{cw}} = -\boldsymbol{R_{cw}}\boldsymbol{T_{wc}}$, we can reformulate into the more useful form $\boldsymbol{M} = \boldsymbol{K}[\boldsymbol{R_{cw}}|-\boldsymbol{R_{cw}}\boldsymbol{T_{wc}}]$. The pushbroom model now takes the form: $\boldsymbol{M} = [\boldsymbol{L}|-\boldsymbol{L}\boldsymbol{T_{wc}}]$, where $\boldsymbol{L}$ is the left hand 3x3 invertible block and $-\boldsymbol{L}\boldsymbol{T_{wc}}$ is the last 3x1 column. $\boldsymbol{T_{wc}}$ can be found by solving the linear equation $\boldsymbol{L}\boldsymbol{T_{wc}} = -\boldsymbol{c_4}$ where $\boldsymbol{c_4}$ is the last column of matrix $\boldsymbol{M}$ (Gupta and Hartley 1997).

$$\boldsymbol{T_{wc}} = -\boldsymbol{L^{-1}}\boldsymbol{c_4}$$

Extraction of rotation matrix $\boldsymbol{R_{cw}}$ and intrinsic parameter matrix $\boldsymbol{K}$ require a more complex RQ decomposition. As $\boldsymbol{K}\boldsymbol{R_{cw}} = \boldsymbol{L}$, and we know that matrix $\boldsymbol{K}$, has the form:

$$\boldsymbol{K} = \begin{bmatrix} * & 0 & 0 \\ * & * & * \\ * & 0 & * \end{bmatrix}$$

We can find a rotation matrix that multiplies $\boldsymbol{L}$ in such a way as to force the (1,2),(1,3),and (3,2) elements of their product to zero.

$$\begin{bmatrix} * & 0 & 0 \\ * & * & * \\ * & 0 & * \end{bmatrix} = \boldsymbol{L}\boldsymbol{R_{cw}^{T}} = \boldsymbol{L}\boldsymbol{R_{wc}}$$

This is achieved by determining three subsequent rotations by Givens matrices about the z, y, and x axes. [Hartely and Zisserman]. $\boldsymbol{R_{wc}} = \boldsymbol{Q_z}\boldsymbol{Q_y}\boldsymbol{Q_x}$

$$\boldsymbol{Q_z} = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ -\sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{Q_y} = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix}, \boldsymbol{Q_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix}$$

For the sake of clarity let us define the intermediate products of $\boldsymbol{L}$ and the givens rotation matrices as:

$$\boldsymbol{P} = \boldsymbol{L}\boldsymbol{Q_z},\ \boldsymbol{Q} = \boldsymbol{L}\boldsymbol{Q_z}\boldsymbol{Q_y}, \boldsymbol{K} = \boldsymbol{L}\boldsymbol{Q_z}\boldsymbol{Q_y}\boldsymbol{Q_x}$$

We firstly choose θ_z so that element (1,2) of $\boldsymbol{P}$ is eliminated. When $\boldsymbol{L}$ is multiplied through by $\boldsymbol{Q_z}$ element $P_{12} = 0 = L_{11}\cos(\theta_z) + L_{12}\sin(\theta_z)$. This can be solved for $\cos(\theta_z)$, and $\sin(\theta_z)$.

$$L_{11}\cos(\theta_z) + L_{12}\sin(\theta_z) = 0$$

$$L_{11}^2\cos^2(\theta_z) = L_{12}^2\sin^2(\theta_z)$$

$$L_{11}^2\cos^2(\theta_z) = L_{12}^2 - L_{12}^2\cos^2(\theta_z)$$

$$(L_{11}^2 + L_{12}^2)\cos^2(\theta_z) = L_{12}^2$$

$$\cos(\theta_z) = \frac{L_{12}^2}{\sqrt[2]{(L_{11}^2 + L_{12}^2)}}$$

The same solution can be obtained for $\sin(\theta_z)$. Care needs to be taken if computing θ_z directly. While the cosine of θ_z can be determined uniquely, the inverse operation can be ambiguous. Using the observation that elements L_{11} and L_{12} can be conceived as lengths of the two short sides of a right-angled triangle we can unambiguously compute θ_z directly as:

$$\theta_z = arctan2(L_{12}, L_{11})$$

θ_y, θ_x are computed in a similar manner, for the sake of brevity their solution will be presented directly.

$$\theta_y = arctan2(P_{13}, -P_{12})$$

$$\theta_x = arctan2(Q_{32}, -Q_{33})$$

Both $\boldsymbol{K}$ and $\boldsymbol{R_{wc}}$ are not uniquely determined. This can be solved by enforcing a positive focal length constraint as follows:

Firstly, we check to see if the focal length is positive by inspecting the quantity $\frac{K_{22}}{K_{33}}$. If it is negative and quantity K_{22} is also negative then we apply firstly a rotation of π about the y axis, followed by a rotation of π about the z axis. In if only the focal length is negative, and quantity K_{22} is positive, then we only apply the first rotation of π about the y axis (Gupta and Hartley 1997).

There is a further step of extracting the intrinsic parameters from matrix $\boldsymbol{K}$, however, this is relatively trivial and will not be detailed here.

4.3.4 Summary of Pushbroom Model Estimation

In summary, a single wavelength image of 700nm is used for estimation of the pushbroom model as it is relatively noise free and shows strong intensity distinction between checkerboard squares useful for automatic corner detections. The entire process was implemented in the form of MATLAB functions. The error free implementation was verified by artificially simulating a wide range of camera intrinsic and extrinsic parameters, projecting points to the image plane using the artificial model, then testing the ability of the algorithm to recover the initial parameters of the camera. Operations such as resampling the image, and even the entire set of 448 wavebands could take in the range of 30 minutes using an unoptimized MATLAB implementation. Consequently, the resampling algorithm was ported to a C/C++ code implementation which was generated semi automatically with the use of MATLAB C coder. Fig.4.9 shows a block diagram which provides a top-level view of the procedures implemented in the pushbroom model estimation and image unwarping algorithm.

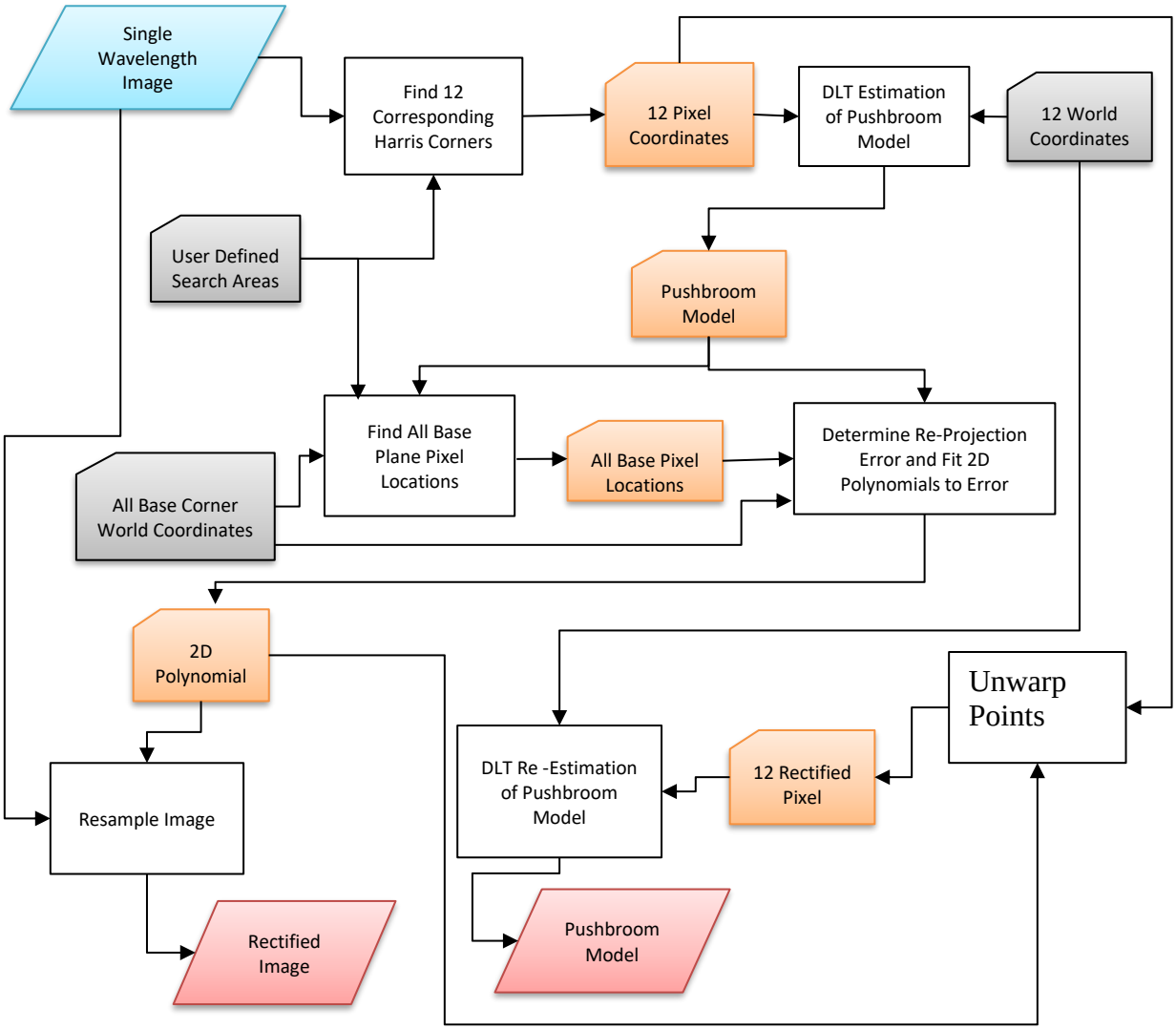

Figure 4.9 - Pushbroom Camera Model Estimation and Image Rectification

4.4 METHODOLOGY FOR THE ASSESSMENT OF PUSHBROOM CALIBRATION EFFICACY, IN TERMS OF REPROJECTION ERROR, STABILITY, WAVELENGTH DEPENDENCY

4.4.1 Error estimation

The illumination process proposed in here depends on characterising the illumination and viewing geometry associated with each pixel. Which in turn depends on the accuracy with which each pixel can be associated with its corresponding illumination and viewing geometry, and ultimately the error within the camera calibration process. The reprojection error can be quantified by measuring the deviation between actual pixel coordinates and those predicted by the model given a set of points known in the world. Both linear and non-linear models can be assessed in terms of efficacy for reducing total reprojection error. While Behmann et al. (2015) have already asserted that a phenomenological approach should be taken towards non-linear effects we are interested to compare this with an model that assumes the distortion varies in the same manner as a 1D radial lens distortion. This can be carried out by fitting one dimensional polynomial functions to the reprojection error. The polynomial corrections can be assessed by comparing reprojection errors.

4.4.2 Calibration stability

Calibration stability is the measure of how little the model varies between image acquisitions. An unstable camera model would require calibration and model re-estimation for each image capture. Moreover, the calibration is only valid within the image area containing the reference gauge where image resampling involves polynomial interpolation and not extrapolation. Meaning that objects need to be placed on top of the calibration gauge. With a stable calibration the gauge only needs to take up valuable space in one image and can thereafter be removed. The model stability can be assessed in terms of the standard deviations of intrinsic parameters f, V_x, V_y, V_z, P_v, as well as deviations of the roll, pitch, and yaw of the camera model (extracted from $\boldsymbol{R_{wc}}$), and standard deviation of its initial position $\boldsymbol{T_{wc}}$.

4.4.3 Wavelength dependent calibration error

Significant spatial misalignment between the images of differing wavebands means that a model generated from one waveband is inappropriate for use with another. Potential sources of wavelength dependent spatial misalignment include both keystone and smile distortions, and misalignment between the separate VNIR and SWIR detector elements of the AisaFENIX camera. This error can be investigated by quantifying the deviation of detected corner points between image wavebands (Behmann et al. 2016).

4.5 3D MODEL PROCESSING

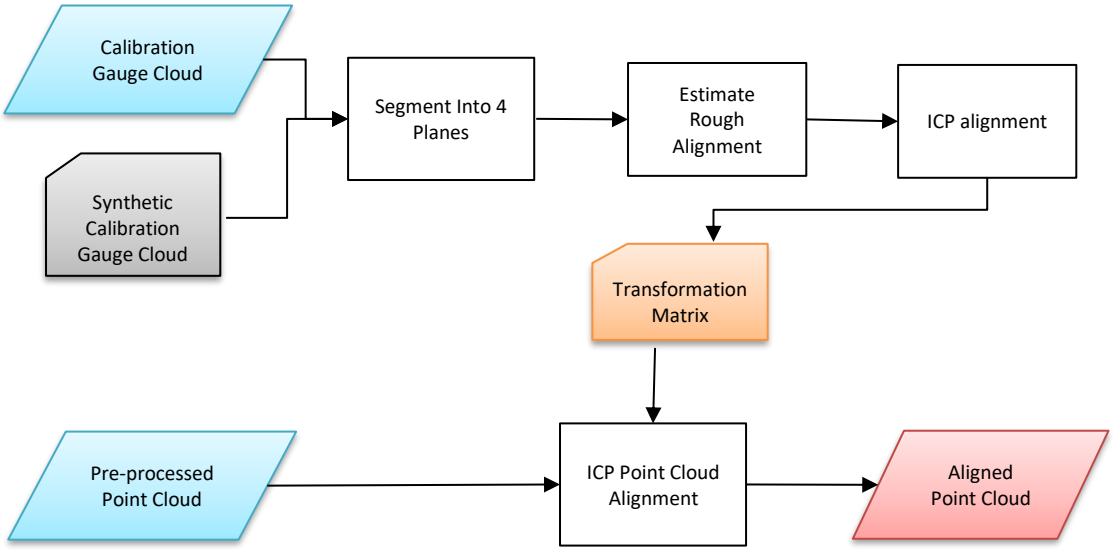

Figure 4.10 - Semi Automatic Point Cloud Alignment

4.5.1 SFM – pre-processing

Unlike the reconstruct me software pipeline which attains scale correct meshes using the Microsoft Kinect hardware, the structure from motion pipeline produces noisy point clouds that require cleaning, scaling, and meshing before they are suitable for illumination correction. Automated cleaning of a raw point cloud is far beyond the scope of this project, consequently the raw clouds are cleaned manually to remove erroneous point clusterings. The scaling factor is determined by comparing sizes of known objects in the cloud to their actual dimension and scaling the cloud to force the object to reassume its actual size.

4.5.1.1 Scale correction

Incorrect scaling of the point cloud would result in mis registration between parts of the image and the point cloud derived mesh. Keeping mis registration error due to scaling less than half a pixel (<1mm) over the ~1.2m travel of the hyperspectral camera requires a scaling percentage error ε_k of less than 0.08%. The scaling factor k is computed as the ratio between the actual distance l_{actual}, and the corresponding measured distance l_{meas} in the unscaled point cloud. l_{meas} can be found by taking the average of the multiple measurements of the distance, where P_a is a point at one end of the distance, and P_b is at the other end. Scaling factor k is computed as:

$$k = \frac{l_{actual}}{l_{meas}} = \frac{l_{actual}}{\frac{1}{N_{meas}} \sum_{i}^{i=N_{meas}} (P_{ai} - P_{bi})}$$

The propagation of error can be expressed in terms of manufacturing error of the object δ_{man}, its actual distance l_{actual}, the point error δ_{meas}, and the number of measurements of the distance N_{meas} that are averaged. The percentage error ε_k is given by:

$$\varepsilon_k = \sqrt[2]{\varepsilon_{meas}^2 + \varepsilon_{actual}^2}$$

$$\varepsilon_{actual} = \frac{\delta_{man}}{l_{actual}}$$

ε_{meas} the percentage error of the arithmetic means of l_{meas} is given by:

$$\varepsilon_{meas} = \frac{\delta l_{meas}}{|l_{meas}|} = \frac{\sqrt[2]{2\delta_{meas}^2 N_{meas}}}{N_{meas}} \frac{1}{l_{actual}}$$

This means that the ratio of manufacturing error to objects length must be at least equal or less than 0.08%. The top and bottom planes of the reference gauge are of 40.00±0.02mm apart, with a point cloud of the top plane (smaller of the two) containing in the order of 160,000 vertices, and a maximum deviation about the plane in the range of ±3mm. Scaling based on the distance between planes yields a scaling error: $\varepsilon_k = 0.06\%$ which is suitable for avoiding misalignment.

Erroneous point clusters small enough to survive manual cleaning can bias the measurement of l_{meas} , and offset the estimation of k. Outliers and inliers can be identified by robustly fitting planar models to the point clouds of both planes using a least squares fitting process embedded within a RANdom-SAmple-Consensus (RANSAC) algorithm. RANSAC will be discussed in greater detail in subsequent sections. Plane to plane distance can only be analytically determined for two parallel parametric models, which is not realistically dependable as models won't be error free and will not be parallel. In our implementation we compute l_{meas} by averaging the distance between inlier points on the top plane and their projection on the bottom plane. As the RANSAC algorithm requires a threshold distance to determine if a point is an outlier or not, the clouds were initially manually scaled to within 5% of their true scale so that, and this algorithm was used for accurate scaling.

4.5.1.2 Meshing

A surface can be defined as a "compact, connected, orientable, two-dimensional manifold" (Kazhdan and Hoppe 2013). The goal of surface reconstruction is to determine a surface that approximates an unknown surface given a partial sample (Hoppe et al. 1992). This problem manifests within a diverse range of scientific and engineering disciplines such as producing CAD models from laser scan data in reverse engineering, in medical studies biological objects are thinly sliced to form a stack of contours, from which three-dimensional

structures need to be recovered. In this case the domain of reconstructions is limited to those that use a three-dimensional point cloud as the partial sample of the surface.

Figure 4.11 - Point Cloud and Its Mesh (Tewari, Gotsman, and Gortler 2006)

Surface reconstruction algorithms can be classified based on the whether they approximate the data or interpolate. Interpolation based algorithms such as the ball pivot algorithm (Bernardini et al. 1999) want to 'connect the dots' while approximation based algorithms do not necessarily intersect all the points but rather aim to approximate the surface, making this method appropriate if the cloud is noisy (Tewari, Gotsman, and Gortler 2006). In this case we chose to use the commonplace screened surface Poisson reconstruction algorithm due to its resilience in the face of noisy data and mis registration artefacts (Kazhdan and Hoppe 2013). We used the algorithm implemented in the CGAL library through the MeshLab interface. The first three parameters: 'Octree Depth', 'Solver Divide', and 'Surface Offsetting' were set to their default values of 8,8 and 1 respectively. The parameter 'Samples per Node' however, can be adjusted to within the range of 1-5, for relatively noise free data while it can be increased to 15-20 to reduce the effects of noise in the reconstruction (Kazhdan n.d) After some tests with this parameter we found that a value of 50 reliably produced surfaces that appeared to be of high enough fidelity upon visual inspection.

4.5.2 Random Sample Consensus (RANSAC)

Least squares fitting of models is an incredibly useful technique and has demonstrated its power and versatility in scenarios where the ratio of outliers to inliers is very small. However, as the fraction of outliers increases, the results of model fitting rapidly decline to unusable. The random sample consensus algorithm proposed by Fischler and Bolles (1987) was designed for robust model estimation in the presence of high proportions of outliers. The following scenario can give an exposition of the inner workings of the RANSAC algorithm.

Consider that a set of data points has an inlier ratio of p,then the probaility of randomly selecting an inlier point from the data set is p. Considering that m is the minimal number of points required to estimate a given model (line, plane, sphere, etc.), then the probaility of picking m points that are inliers from the data set is p^m , and the probability of estimating an incorrect model is $1 - p^m$. However, if we were to iteratively pick m random points N times, the probability of estimating N wrong models is $(1 - p^m)^N$ which is a quantity whose value decreases as the number iterations increase. The probability of estimating a correct model at least once in N iterations is: $1 - (1 - p^m)^N$. Once a model has been estimated it is evaluated by determining the number of points which lie within some predetermined threshold distance from the model. If the ratio of points that fit within the model to number of points in the data set is greater than another predefined threshold, then the algorithm has found a correct model. The model is further refined by re-estimating the model from all the inlier points, and then algorithm terminates (Derpanis 2010).

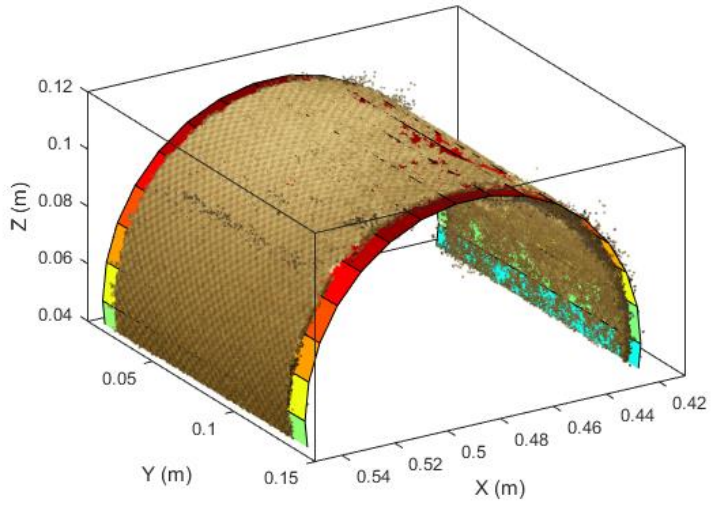

Figure 4.12- Cylinder Model Fitted to Point Cloud Using RANSAC Despite Outliers

There have since been numerous variants added to the family of RANSAC algorithms, with the intension of improving either accuracy, robustness, or speed of the algorithm (Choi, Kim, and Yu 1997). RANSAC algorithms are commonly used in the computer vision communities (in which they were developed) to robustly estimate camera models, fundamental matrices of stereo pairs, or in robotics communities for detection of geometric primitives such as planes, cylinders (Fig.4.12), spheres in indoor scenes. However, our application will involve

accurate measuring of angles and distances in both cylinder and planar models in which case we should prioritise accuracy. To this end we chose the MLESAC variant. The MLESAC estimator uses the same sampling strategy as the RANSAC algorithm, however, instead of using number of inliers as an indicator of the 'goodness' of the model, the MLESAC algorithm maximises the likelihood that all inlier points fit within the model (Torr and Zisserman 2000). The worst case performance of MLESAC is equivalent to RANSAC, however, its best case performance surpasses the accuracy of RANSAC (Torr and Zisserman 2000).

4.5.3 Model alignment and Iterative Closest Point (ICP)

The iterative closest point algorithm is widely used for determining the rigid body transformation between two sets of points usually in two or three-dimensional space, although higher dimensions are possible. Common applications include aligning fingerprint scans in fingerprint verification (Jain and Ross 2002), or estimating the pose of a robot in navigation scenarios (Milella and Siegwart 2006). It is used here as it can register a captured point cloud to the coordinate system of another already aligned cloud. It estimates the rigid body translation by minimising the point to point or point to plane distance between two point clouds. While the iterative closest point algorithm is able very accurately align two point clouds, it is only suitable for a final precise alignment, and performs poorly when presented with any two misaligned clouds. Its narrow convergence funnel requires that the two point clouds are pre-aligned.

In this case the point cloud was precisely registered to a fiducial, artificially generated model of the reference gauge after an initial manual pre-alignment. As the planes of the reference gauge were to be aligned to one another, we used an ICP variant that supported the minimisation of a point to plane metric, as this has been shown to produce higher quality results for point clouds of planes (Bergström and Edlund 2014).

4.6 ASSESSMENT OF MESH SUITABILITY FOR LIGHTING CORRECTION

The illumination correction requires accurate determination of illumination angle of incidence, and therefore requires accurate surface normal information. Inaccuracy in surface normal will translate to inaccuracy in the illumination correction. Therefore, this assessment aims to evaluate the efficacy of structure from motion and Microsoft Kinect SLAM point cloud acquisition methodologies to reproduce models appropriate for use in illumination correction of hyperspectral images. The two main concerns are susceptibility to misalignment with the world coordinate system, and surface accuracy, which translate into illumination correction of the incorrect pixel, and misevaluation of illumination attenuation effects respectively. Alignment and bias tests quantify the mis registration of the captured mesh with the real-world coordinate system, while the surface accuracy tests evaluate how far the digital surfaces deviate from reality.

4.6.1 Planar model surface angular bias

Illumination correction depends on the accuracy with which surface inclination is estimated by point cloud acquisition technique. For instance DiFilippo and Jouaneh (2015) has already shown that the Microsoft Kinect consistently underestimates the angle of planar surfaces by a few degrees, and requires a correction factor to improve its accuracy for applications such as volume estimation of objects. Here we are interested in comparing the structure from motion and Microsoft Kinect technologies for their accuracy in determining the angle of a planar surface and investigating how this estimation varies with both the angle of the surface and the spatial frequency of surface texture. Five experiments were run where covered the surface with printed checkerboard patterns with spatial frequencies of 0.55,1.1,2.2,4.4,8.8 millimetres per square (referred to as T1,T2,T3,T4,T5).

The planar angle bias was assessed by RANSAC fitting of planar models to each of the captured planes and evaluating the normal angle between the plane, and the planar model that was fitted to the table surface of the reference gauge. Manually this is computed as the inverse cosine of the dot product between the normal vectors of the two-plane's normals. These angles for both structure from motion, and the Microsoft Kinect clouds were compared against manual measurements which have a maximum angle error of 0.34 degrees (see section 3.1.5).

4.6.2 Planar model mesh angular deviation

We used the data from the same experimental procedure as outlined in section 4.6.1. However, instead of investigating the bias in angle measurements, we are interested in evaluating the angular deviation of the faces of a triangular mesh with respect to the plane that that they comprise. Which is evaluated for textures T1-T5. This is achieved by computing the normals of each of the triangular faces within the mesh and comparing it to the normal of the fitted plane.

4.6.3 Angular deviation with radius

The leaves and stems of plants will have varying local curvature and associated radii of curvature. The limited resolution and accuracy of point cloud acquisition techniques as well as unpublished meshing technique of the Microsoft Kinect will result in an error in the surface orientation of the 3D model, which will ultimately affect the illumination correction accuracy. Cylindrical objects were fabricated (with radii of 15,24,34,45,and 63 millimetres, referred to as C1,C2,C3,C4,C5), and their point clouds were acquired by both Microsoft Kinect and Structure from motion techniques. Cylindrical models were robustly fitted to the point clouds within a RANSAC algorithm. The cylinder 3D model surface was assessed for angular deviation by finding the normal of each triangle that comprised the cylindrical mesh and comparing this to the vector subtended from the centre of the triangle, to the principal axis of the cylinder under the constraint that the vector is orthogonal to the principal axis of the cylinder. Fig.4.13 below shows a triangular mesh (pink), to which a cylinder model is fitted. The red and corresponding green vectors are the normal vectors of the mesh and cylinder model respectively.

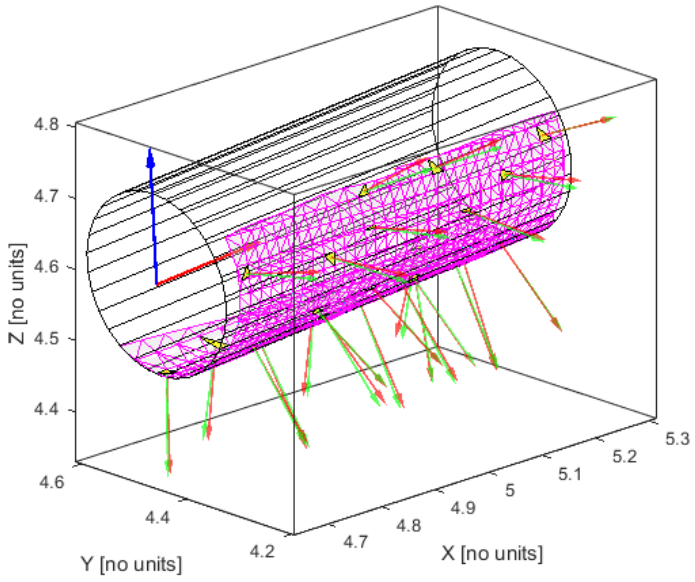

Figure 4.13- Normal Vectors of a Triangular Mesh and its Cylinder Model

4.7 ILLUMINATION CORRECTION

4.7.1 Benefits of Illumination correction

Both Behmann et al. (2016) and Mishra et al. (2016) provide useful advances upon current correction methodologies. However, while Behmann et al. (2016) is able to model the complex plant geometry, they do not directly correct illumination effects. While Mishra et al. (2016) is able to successfully handle specular effects without complex BRDF models, they are unable to adequately choose a reference reflectance spectrum to which all spectra are fitted against. Here a correction model that builds upon both the work of Behmann et al. (2016) and Mishra et al. (2016) is proposed.

The model aims to improve on the work of Mishra et al. (2016) by using geometrical information to determine which areas will not contain specular reflections, in which case their absolute reflectance can be measured, and then can be further used as reference spectra to which pixels containing specular reflectance can be corrected. Provided that the images and 3D models can be adequately aligned to a common coordinate system then it is expected that this method will allow: direct modelling of Lambertian effects, evaluation of likelihood of specular reflections, and will allow for choice of better reference spectrums when applying techniques such as proposed by Mishra et al. (2016).

4.7.2 Principle of illumination correction

The principle of the illumination correction is to account for Lambertian effects (inclination, distance) with Lambertian factor β then estimate the specular component α . The incident radiance L_{id} received by the surface of the object directly from a point light source can be expressed as:

$$L_{id} = \beta L_0 = \frac{\cos(\theta)}{\left(\frac{d}{d_0}\right)^2} L_0$$

The radiance received by the camera can be modelled as the sum of radiance from the halogen lights diffusely reflected by the surface ρL_{id}, the radiance received by the camera due to ambients light reflecting from the surface L_{ma}, and finally a specular component α.

$$L_m = \rho L_{id} + L_{ma} + \alpha$$

The gains, offsets, and internal optical system result in the radiance received by the camera being scaled and shifted into a digital number recorded by the detector element. This is a linear relationship expressed with gain c_1, and offset $c0$.

$$DN = c_1 L_m + c_0$$

These relationships can be combined to form the following relationship, which relates the digital number recorded to the diffuse reflectance.

$$\rho = \frac{1}{\beta L_0}\left[\frac{DN - c_0}{c_1} - L_{ma} - \alpha\right]$$

4.7.3 Implementation

The illumination correction here is implemented in two main steps: ray tracing to obtain information about the illumination geometry of each pixel, and secondly the determination of correction factors, which are used to correct the image. Ray tracing is a common technique popular in computer graphics applications for its realistic physical modelling of the propagation of light rays through a scene, and finally their formation of an image on the image plane of the camera. Propagation of light from the light source to a pixel on the image plane is called forward propagation. In practice it is more computationally efficient to use backpropagation, which involves tracing from the pixel to the light source, as there is no guarantee that an arbitrary ray from the light source will propagate and be visible to the camera. The ray tracing algorithm involves casting a ray into the scene from the camera, determining which surface is viewed by this pixel, determining which light sources illuminate this surface. The ray geometry can be used to calculate the β and α factors and correct the image.

4.7.4 Ray vector calculation

A ray can be defined by either two points, or a single point and direction vector. In this case the ray is defined by a single point in space, the centre of projection of the camera; and a direction vector, the vector pointing from the centre of projection to a pixel on the virtual image plane. While the camera coordinate system specifies the rotation and translation of the camera at the start of its motion, the vector $\boldsymbol{V}$ describes the linear translation of the camera centre with respect to this frame. Moreover, the components V_x, V_y, and $V_z$descibe the motion in metres between every pixel line along the u image axis. The centre of projection in the camera coordinate system is given as:

$$\boldsymbol{P_{cprj}} = u\widehat{\boldsymbol{V}} = u\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

As the camera is orthographic along the u-axis, the vector between the pixel (in the virtual plane) and the camera centre will have no deviation along the u-axis. The vector direction in the v-axis is the difference between the pixel coordinate v, and offset of the camera centre relative to the camera coordinate system. Finally, the difference between centre of projection in the z image axis and virtual image plane is the focal length f.

$$\boldsymbol{D_c} = \begin{bmatrix} 0 \\ v - P_v \\ f \end{bmatrix}$$

To be useful for calculation within a world coordinate system the ray must be transformed into the world coordinate system. The centre of projection, and direction are transformed as:

$$\boldsymbol{P_{wprj} = R_{wc}P_{cprj} + T_{wc}}$$

$$\boldsymbol{D_w = R_{wc}D_c}$$

4.7.5 Ray intersection

There are many techniques for determining the illumination of a surface as viewed by a pixel within the computer graphics literature. These can be generally broken into the computationally simple, and fast scanline rendering techniques; or the more computationally expensive, slower ray tracing or ray casting techniques common in high end photorealistic rendering applications. The advantage of the ray tracing algorithms is the underlying realistic light propagation model that allows for accurate modelling of shadows, and reflections. However, before illumination effects can be modelled, the intersection between the ray and an object in the scene must be determined. Calculation of both the intersection and surface geometry depend on the representation of the object. The ray tracer implemented here (in MATLAB) supports intersection between three primitive object models, cylinders, triangles, and boxes. Theory and implementation of intersection with these primitives is detailed in the subsequent sections.

4.7.5.1 Ray Cylinder Intersection

To determine an upper limit on the potential of illumination correction on the cylinder data set, the ray tracer needed to support ideal parametrically described cylindrical models (see section 4.8.1). The finite cylinder model (Fig.4.14) can be fully defined by two points $\boldsymbol{P_1, P_2}$, and its radius. Considering the following cylindrical model with end points defined $\boldsymbol{P_1, P_2}$, with $\Delta\boldsymbol{P} = \boldsymbol{P_2} - \boldsymbol{P_1}$, and vector q describing the displacement of a point on the surface from the principal axis, any point $\boldsymbol{P_s}$ on an infinite cylinder surface is given by:

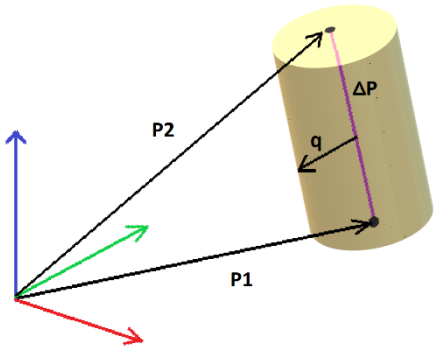

Figure 4.14 –Cylinder Line Intersection Model

$$\boldsymbol{P_s = P_1 + \propto \Delta P + q}$$

A ray intersecting this surface has the form: $\boldsymbol{P_0 + \mu d}$, where $\boldsymbol{P_0}$ is a point on the line, $\boldsymbol{d}$ is the normalised direction vector, and μ is the distance between $\boldsymbol{P_0}$, and the surface. Equating these two yields:

$$\boldsymbol{P_1 + \propto \Delta P + q = P_0 + \mu d}$$

$\propto, \mu$ and vector q, are unknown, and must be determined determine if there is an intersection, and the location at which it occurs. The first step is solving for $\propto$, μ, and $\boldsymbol{q}$, which must be followed by determining an

equation which is formulated in such a way as to contain only one unkown. As $\boldsymbol{q} \cdot \Delta \boldsymbol{P} = 0$, taking the dot product of both sides with $\Delta \boldsymbol{P}$ yields:

$$\propto (\Delta \boldsymbol{P} \cdot \Delta \boldsymbol{P}) = \boldsymbol{P_0} \cdot \Delta \boldsymbol{P} + \mu \boldsymbol{d} \cdot \Delta \boldsymbol{P} - \boldsymbol{P_1} \Delta \boldsymbol{P}$$

Solving for $\propto$:

$$\propto = \frac{\boldsymbol{P_0} \cdot \Delta \boldsymbol{P} + \mu \boldsymbol{d} \cdot \Delta \boldsymbol{P} - \boldsymbol{P_1} \Delta \boldsymbol{P}}{\Delta \boldsymbol{P} \cdot \Delta \boldsymbol{P}}$$

Solving for q, by substituting in:

$$\propto = \frac{\boldsymbol{P_0} + \mu \boldsymbol{d} - \boldsymbol{q} - \boldsymbol{P_1}}{\Delta \boldsymbol{P}}$$

$$\boldsymbol{q} = \boldsymbol{P_0} + \mu \boldsymbol{d} + \boldsymbol{P_1} - \left[\frac{\boldsymbol{P_0} \cdot \Delta \boldsymbol{P} + \mu \boldsymbol{d} \cdot \Delta \boldsymbol{P} - \boldsymbol{P_1} \Delta \boldsymbol{P}}{\Delta \boldsymbol{P} \cdot \Delta \boldsymbol{P}}\right] \Delta \boldsymbol{P}$$

Solving for μ, while eliminating all other unknowns:

As $\boldsymbol{q} \cdot \boldsymbol{q} = r^2$ where r is the cylinder radius

$$\left(\boldsymbol{P}_0 + \mu \boldsymbol{d} + \boldsymbol{P}_1 - \left[\frac{\boldsymbol{P_0} \cdot \Delta \boldsymbol{P} + \mu \boldsymbol{d} \cdot \Delta \boldsymbol{P} - \boldsymbol{P_1} \Delta \boldsymbol{P}}{\Delta \boldsymbol{P} \cdot \Delta \boldsymbol{P}}\right] \Delta \boldsymbol{P}\right)^2 - r^2 = 0$$

We can solve by treating μ as a quadratic and finding its roots.

$$\left(\mu \boldsymbol{d} - \mu \left(\frac{\boldsymbol{d} \cdot \Delta \boldsymbol{P}}{\Delta \boldsymbol{P} \cdot \Delta \boldsymbol{P}}\right) \Delta \boldsymbol{P} + \boldsymbol{P_0} - \boldsymbol{P_1} - \left[\frac{\boldsymbol{P_0} \cdot \Delta \boldsymbol{P} - \boldsymbol{P_1} \Delta \boldsymbol{P}}{\Delta \boldsymbol{P} \cdot \Delta \boldsymbol{P}}\right] \Delta \boldsymbol{P}\right)^2 - r^2 = 0$$

$$\left(\mu \left(\boldsymbol{d} - \frac{\boldsymbol{d} \cdot \Delta \boldsymbol{P}}{\Delta \boldsymbol{P} \cdot \Delta \boldsymbol{P}} \Delta \boldsymbol{P}\right) + \boldsymbol{P_0} - \boldsymbol{P_1} - \left[\frac{(\boldsymbol{P_0} - \boldsymbol{P_1}) \Delta \boldsymbol{P}}{\Delta \boldsymbol{P} \cdot \Delta \boldsymbol{P}}\right] \Delta P\right)^2 - r^2 = 0$$

Let: vectors $\boldsymbol{a} = \left(\boldsymbol{d} - \frac{\boldsymbol{d} \cdot \Delta \boldsymbol{P}}{\Delta \boldsymbol{P} \cdot \Delta \boldsymbol{P}} \Delta \boldsymbol{P}\right)$, and $\boldsymbol{b} = \boldsymbol{P_0} - \boldsymbol{P_1} - \frac{(\boldsymbol{P_0} - \boldsymbol{P_1}) \Delta \boldsymbol{P}}{\Delta \boldsymbol{P} \cdot \Delta \boldsymbol{P}} \Delta \boldsymbol{P}$

$$\boldsymbol{q} = \mu \boldsymbol{a} + \boldsymbol{b}$$

Denoting the x component of vector **a** as a_x

$$\boldsymbol{q} = \begin{bmatrix} \mu a_x + b_x \\ \mu a_y + b_y \\ \mu a_z + b_z \end{bmatrix}$$

$$\boldsymbol{q} \cdot \boldsymbol{q} - r^2 = 0$$

$$(\mu a_x + b_x)^2 + \left(\mu a_y + b_y\right)^2 + (\mu a_z + b_z)^2 - r^2 = 0$$

$$(\mu^2 a_x^2 + 2\mu a_x b_x + b_x^2) + (\mu^2 a_y^2 + 2\mu a_y b_y + b_y^2) + (\mu^2 a_z^2 + 2\mu a_z b_z + b_z^2) - r^2 = 0$$

$$\mu^2\left(a_x^2 + a_y^2 + a_z^2\right) + 2u\left(a_x b_x + a_y b_y + a_z b_z\right) + \left(b_x^2 + b_y^2 + b_z^2\right) - r^2 = 0$$

$$\mu^2 \boldsymbol{a}\cdot\boldsymbol{a} + 2\mu \boldsymbol{a}\cdot\boldsymbol{b} + \boldsymbol{b}\cdot\boldsymbol{b} - r^2 = 0$$

Let $A = \boldsymbol{a}\cdot\boldsymbol{a},\ B = \boldsymbol{a}\cdot\boldsymbol{b},\ C = \boldsymbol{b}\cdot\boldsymbol{b} - r^2$

$$\mu^2 A + 2\mu B + C = 0$$

$$\mu^2 + \frac{2\mu B}{A} + \frac{C}{A} = 0$$

$$u = -\frac{B}{A} \pm \sqrt{\frac{B^2}{A^2} - \frac{C}{A}}$$

There are three possible intersection possibilities, the ray intersects twice, once, or there is no intersection. This is corresponding to two roots, one root, or no roots which can be evaluated by checking the determinant. Once μ is obtained $\propto$,and $\boldsymbol{q}$ can be computed by substitution of μ. For finite cylinders there must be a further check to ensure that $0 \leq \propto \leq 1$, corresponding to an intersection of the surface between $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$.

4.7.5.2 Ray Triangle intersection

Representation of 3D geometrical objects of arbitrary accuracy and complexity can be achieved by representation as a mesh of small finite triangles. Triangles are preferable as their three points define a plane to which the fields of linear algebra and geometry offer many results that are attractive for computation (Wylie et al. 1967). While there are many variants of the ray tracing algorithm the version used was the efficient Möller and Trumbore (2005) algorithm, which was implemented in MATLAB.

There are four cases for the intersection between a ray and a triangle. These are:

- Triangle degeneracy where the three vertices of the triangle are collinear meaning the triangle has no surface to be intersected.
- The ray lies within the plane of the triangle meaning the surface cannot reflect light into the camera.
- The ray misses the triangle completely.
- The ray intersects the triangle.

For these cases, consider the triangle with three vertices $\boldsymbol{V_0}, \boldsymbol{V_1}, \boldsymbol{V_2}$, with sides $\boldsymbol{u} = \boldsymbol{V_1} - \boldsymbol{V_0}, \boldsymbol{v} = \boldsymbol{V_2} - \boldsymbol{V_0}$.

4.7.5.2.1 Case 1: Triangle degeneracy

Checking for triangle degeneracy is can be performed by taking the cross product between u, v which computes the normal of the triangle. In the case that the triangle is degenerate (all three vertices are collinear) $\boldsymbol{u} \times \boldsymbol{v} = [0 \quad 0 \quad 0]^T$, and the triangle should be ignored.

4.7.5.2.2 Case 2: Non-singular intersection with plane

This case arises when either the ray lies within the plane, or when the plane is behind the origin of the ray. The ray will lie within the plane if both its direction vector is orthogonal to the plane normal and if the plane ray origin lies within the plane. The plane normal is computed as:

$$\boldsymbol{n} = \boldsymbol{u} \times \boldsymbol{v}$$

The ray is parallel to the plane if the dot product between its normalised direction vector and the plane normal is zero ($b = 0$).

$$\boldsymbol{w_0} = \boldsymbol{P_0} - \boldsymbol{V_0}$$

$$b = \boldsymbol{n} \cdot \boldsymbol{d}$$

Checking if the ray origin lies within the plane is the same as checking that the vector between the ray origin and one of the triangle vertices is orthogonal to the plane normal $\boldsymbol{n}$, (a=0).

$$a = -\boldsymbol{n} \cdot \boldsymbol{w_0}$$

Finally, the last non-singular intersection case of the triangle lying behind the ray origin can be determined by finding the distance between the ray origin and the plane along the ray vector, if the distance is negative, the triangle lies behind the ray origin. Considering the plane equation, where $\boldsymbol{n}$ is the plane norm, $\boldsymbol{x_0}$ is a point on the plane, and d is the distance divided by length of $\boldsymbol{n}$ between the plane and the origin along $\boldsymbol{n}$.

$$\boldsymbol{n} \cdot \boldsymbol{x_0} = d$$

If the plane was shifted so that the triangle vertex $\boldsymbol{V_0}$ is coincident with the origin, distance $d = 0$, and the plane equation can be re-written as:

$$\boldsymbol{n} \cdot (\boldsymbol{x_0} - \boldsymbol{V_0}) = 0$$

Substituting a point on the ray for $\boldsymbol{x_0}$ yields:

$$\boldsymbol{n} \cdot (\boldsymbol{P_0} + \mu \boldsymbol{d} - \boldsymbol{V_0}) = 0$$

Solving for distance μ:

$$\mu = \frac{-\boldsymbol{n} \cdot (\boldsymbol{P_0} - \boldsymbol{V_0})}{\boldsymbol{n} \cdot \boldsymbol{d}} = \frac{-\boldsymbol{n} \cdot \boldsymbol{w_0}}{\boldsymbol{n} \cdot \boldsymbol{d}}$$

$$\mu = \frac{a}{b}$$

4.7.5.2.3 Cases 3 and 4: Verifying the point lies within the triangle

There is no guarantee that this point lies within the bounds of the triangle, despite its intersection with the plane. A triangular barycentric coordinate system makes it easier to determine if a point lies within a triangle, a point in a barycentric coordinate system can be expressed as:

$$P_x = (1 - s - t)V_0 + sV_1 + tV_2$$

Verification of the point lying within the triangle is done by checking that the conditions $s \geq 0, t \geq 0$, and $s + t \leq 1$ are met. Substituting in the ray equation gives the following formulation which can be solved to determine s, t,and μ.

$$\boldsymbol{P_0} + \mu\boldsymbol{d} = (1 - s - t)\boldsymbol{V_0} + s\boldsymbol{V_1} + t\boldsymbol{V_2}$$

$$[\boldsymbol{d}, \quad \boldsymbol{V_1} - \boldsymbol{V_0}, \quad \boldsymbol{V_2} - \boldsymbol{V_0}] \begin{bmatrix} \mu \\ s \\ t \end{bmatrix} = [\boldsymbol{P_0} - \boldsymbol{V_0}]$$

$$[\boldsymbol{d}, \quad \boldsymbol{u}, \quad \boldsymbol{v}] \begin{bmatrix} \mu \\ s \\ t \end{bmatrix} = [\boldsymbol{w_0}]$$

This system of equations can be solved using Cramer's rule, where 'det' stands for determinant.

$$\begin{bmatrix} \mu \\ s \\ t \end{bmatrix} = \frac{1}{\det([-\boldsymbol{d}, \quad \boldsymbol{u}, \quad \boldsymbol{v}])} \begin{bmatrix} \det([\boldsymbol{w_0} \quad \boldsymbol{u} \quad \boldsymbol{v}]) \\ \det([-\boldsymbol{d} \quad \boldsymbol{w_0} \quad \boldsymbol{v}]) \\ \det([-\boldsymbol{d} \quad \boldsymbol{u} \quad \boldsymbol{w_0}]) \end{bmatrix}$$

(Möller and Trumbore 2005)

4.7.5.3 Ray Box intersection and Space Partitioning

Implementation of the ray tracing algorithm on an image of 384x578 with a triangle mesh of 30,000 faces would require in the order of 10^9 primary intersection tests, let alone secondary intersection tests. This could easily take days to process one image. Real time rendering of large scenes has been studied intensely in the computer graphics literature (Chin and Feiner 1989). The common answer to this question has been the implementation of space partitioning. As opposed to traversing a list of every object/surface, only those occupying a smaller search space are checked for an intersection. There are multiple forms such as K-D trees(Shevtsov, Soupikov, and Kapustin 2007), quadtrees, hierarchical clustering(Garland, Willmott, and Heckbert 2001), octrees(Samet 1989), and Binary Space Partitioning (BSP).

The implementation for these techniques for triangular meshes is more complex than for a point cloud as faces lying on the partitioning hyperplane must be further subdivided adding additional complexity to the algorithm and the number of faces is often much larger than the original set. As the end goal was not to create a perfect ray tracer, an intuitively simple point cloud based octree space partitioning scheme was implemented.

4.7.5.3.1 Octree construction

An octree construction is performed by recursively dividing the volume containing all the vertices into eight equally sized subspaces. The recursive space division end when either the bins reach a minimal capacity of points, or the bin side length reaches a minimal threshold value, or the bin depth reaches a maximum value. Each bin is an axis aligned bounding box meaning each of its edges are parallel with the axes of the coordinate system. This means the box can be uniquely defined by two points.

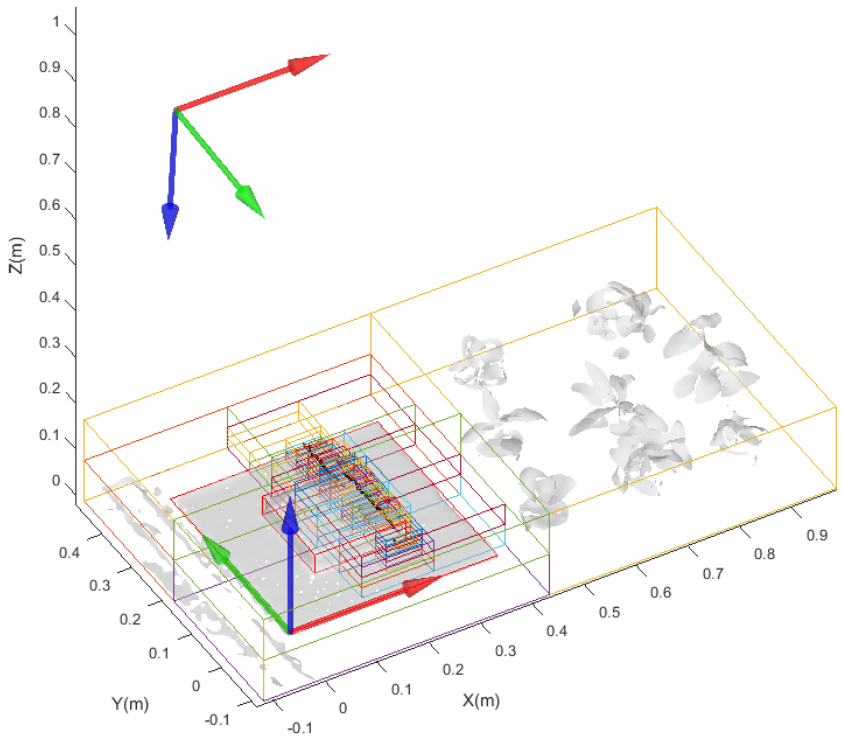

Figure 4.15- Partial Visualisation of Octree for a Mesh of Various Plants

4.7.5.3.2 Ray AABB intersection test

The correct search space(s) can be identified by performing an intersection test between the ray and the AABB that contain the search spaces. The two points (bounds) $\boldsymbol{B_0}, \boldsymbol{B_1}$ of the box correspond to the maximum and minimum x, y, z coordinates contained within the AABB.

The basic principle is to check a ray expressed as: $\boldsymbol{P_x = P_0 + \mu d}$ for intersection with the 6 planes that enclose the space and determine which of the intersections (if any) correspond to a ray-box intersection. The six planes are: $x = B_{0x}, x = B_{1x}, y = B_{0y}, y = B_{1y}, z = B_{0z}, z = B_{1z}$. The six intersections can be expressed in terms of μ, the distance along the ray from the origin of the ray yielding:

$$\mu_{0x} = \frac{B_{0x} - P_{0x}}{d_x}, \quad \mu_{1x} = \frac{B_{1x} - P_{0x}}{d_x}$$

$$\mu_{0y} = \frac{B_{0y} - P_{0y}}{d_y}, \quad \mu_{1x} = \frac{B_{1y} - P_{0y}}{d_y}$$

$$\mu_{0z} = \frac{B_{0z} - P_{0z}}{d_z}, \quad \mu_{1z} = \frac{B_{1z} - P_{0z}}{d_z}$$

Considering the diagram below of a 2D analogue, the intersection points parameterised as μ_{min}, μ_{max} correspond to the first and second intersection of the ray with the box. μ_{min} is the greatest μ value of the first two x, y plane intersections. μ_{max} is the smallest μ value of the last two x, y, plane intersections.

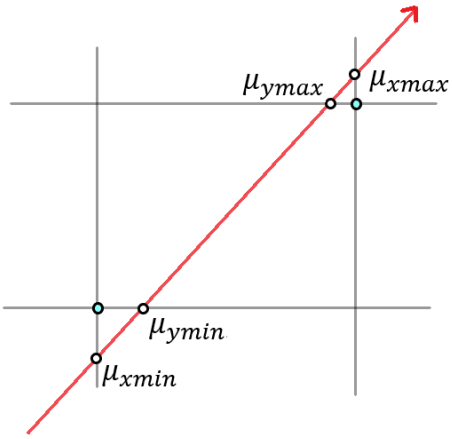

Figure 4.16 – 2D Ray Box Intersection

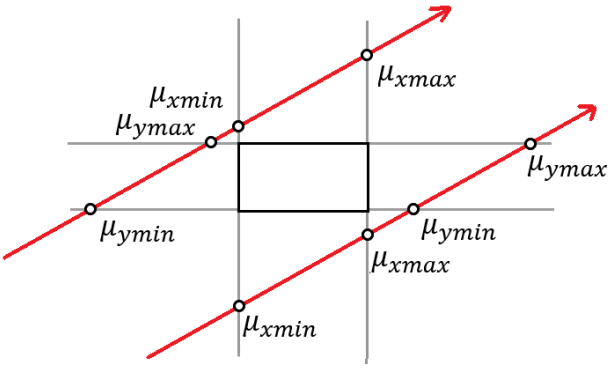

Figure 4.17 – 2D Ray Box Non-Intersection

This can be easily generalised into 3D:

$$\mu_{min} = \max(\mu_{xmin}, \mu_{ymin}, \mu_{zmin})$$
$$\mu_{max} = \min(\mu_{xmax}, \mu_{ymax}, \mu_{zmax})$$

To handle rays with any generic direction and origin $(\mu_{xmin}, \mu_{ymin}, \mu_{zmin})$ do not necessarily correspond to $(\mu_{0x}, \mu_{0y}, \mu_{0z})$. μ_{xmin} is the μ value corresponding to the closest of the two intersections with planes with constant x values ($\mu_{xmin} = \min(\mu_{0x}, \mu_{1x})$. Once μ_{min}, and μ_{max} have been found the intersection coordinates can be computed using the ray equation.

The case of no intersection corresponds to the ray intersecting twice with planes of one type (x, y, or z) before having intersected once with planes of the other two types. This can be identified when μ_{min} of one plane intersection is greater than the μ_{max} of either of the other two planes types.

4.7.6 Secondary Rays and Shadow Rays

Secondary rays are cast from a point of intersection in the scene to each light source (Fig 4.18). For each secondary ray if there are no triangles intersected between origin and light source, the illumination geometry is recorded. Otherwise if there is an intersection, the light source is occluded, and does not contribute to the illumination of the surface point viewed by the camera.

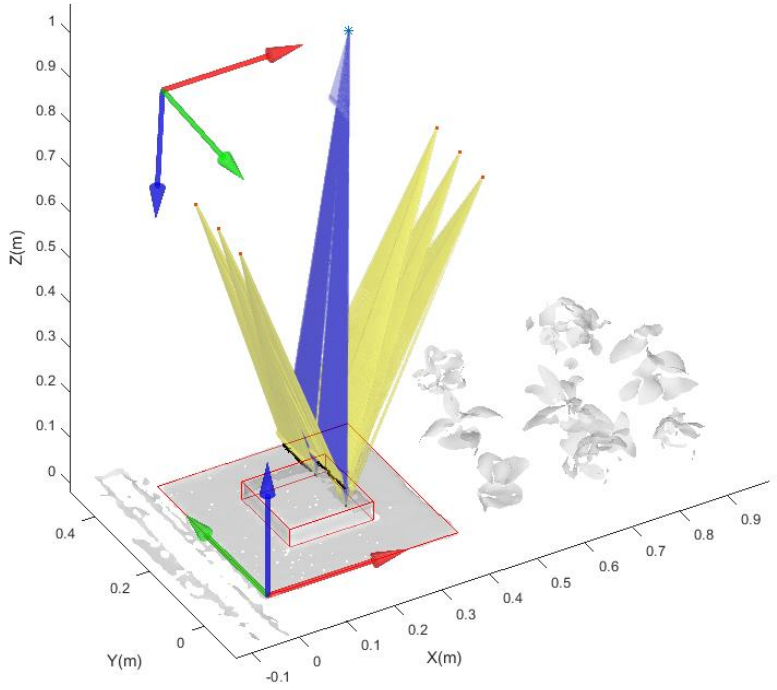

Figure 4.18 – Primary Rays (Blue), Secondary Rays (Yellow)

4.7.7 Illumination Correction

The image can be corrected in three stages. These are estimation of scaling factor *c1* which is computed using the white reference, the second stage is absolute reflectance correction in areas where there is little specular reflection, and the third stage is to correct parts of the image containing specular reflections.

Beta can be computed as the sum of all β_i for each illumination source visible to the intersection point on the surface.

$$\beta = \frac{1}{n}\sum_{i=1}^{i=n} \beta_i \;\; = \frac{1}{n}\sum_{i=1}^{i=n} \frac{\widehat{\boldsymbol{N}} \cdot \widehat{\boldsymbol{I_i}}}{\left(\frac{d_i}{d_{i0}}\right)^2}$$

Where $\widehat{\boldsymbol{N}}$ is the normal vector of the surface, $\widehat{\boldsymbol{I_i}}$ is the vector from the intersection point to point light source i, d_i is the distance between the intersection point and the light source, d_{i0} is the vertical distance between the light source on the first pixel line, n is the number of lightsources visible to the intersection point.

c_0 is the simplest factor to determine, as it is just the dark frame. c_1 accounts for the intrinsic sensor and optical system gains. This factor is typically determined using flat field correction with respect to a white reference, or normalisation in a close-range setting. In this case the factor c_1 will also account for spatial variation in illumination intensity which should be independently accounted for by lambertian factor β. This requires β to be known prior to determination of c_1. When calculating c_1 it is assumed that there are no specular reflections as the white reference is a near ideal lambertian reflector, therfore $\alpha = 0$. c_1 can be computed as:

$$c_1 = \frac{DN - c_0}{\rho\beta L_0} = \frac{DN - c_0}{\beta}$$

Where L_0 is assumed to be 1 for each point light source.

Once c_0, and c_1 have been determined, then it is possible to correct the rest of the image. Firstly, the areas likely to have little specular reflectance can be processed assuming that $\alpha = 0$. Areas likely to have specular effects require that α is calculed by fitting the raw spectra to a reference spectrum. Hence, there is a need to distinguish between the two situations. BRDF distribution functions commonly exhibit spectral lobes around a vector which has an azimuthal angle of 180° and zenith angle equal to that of the vector of an incident illumination ray. Here we propose that provided the viewing vector (from intersection point to centre of projection) lies outside the specular lobe we should be able to assume $\alpha = 0$. A specular region can be designated by a cone centred on the specular viewing vector. An angular threshold will define the narrowness of the cone. This threshold can be increased for more conservative estimates. The angle between the specular viewing vector $\hat{\mathrm{S}}_i$ for light source i, and viewing vector $\hat{V}$.

$$\varphi_i = |\cos^{-1}(\hat{\mathrm{S}}_i \cdot \hat{V})|$$

If the maximum of φ_i is greater than a threshold, then that pixel will be corrected using the second method.

The first method can be applied as:

$$\rho = \frac{DN - c_0}{c_1\beta L_0}$$

In the case that $\varphi_{max} > \varphi_{threshold}$, $\alpha \neq 0$. A reference spectrum is required to fit the measured spectrum to. As specular reflections can be assumed to be independent of wavelength, the difference between the measured reflectance and a reference spectrum from a very similar material can be expressed as. Alpha can

then be estimated through least squares linear regression that aims to minimise $n(\lambda)$, which is the wavelength dependent error.

$$\rho_{ref} = \rho_{meas} - \frac{\alpha}{\beta} - n(\lambda)$$

Once α, has been determined the pixel can be corrected as:

$$\rho = \frac{1}{\beta L_0}\left[\frac{DN - c_0}{c_1} - \alpha\right]$$

4.8 Assessment of Illumination Correction

4.8.1 Assessment of Illumination Correction on Cylindrical Surfaces

There exist methods in computer graphics such as antialiasing which cast multiple rays from one pixel and average the computed pixel colour as determined by each cast ray. As poor meshes can be averaged multiple times per pixel we are interested in determining an upper limit for the correction with respect to radius.

This can be implemented by obtaining a mesh and hyperspectral image of cylinders of various radius, applying the corrections (specular and Lambertian) and assessing them against a ground truth. To reduce the strength of specular effects the cylinders were wrapped in matte brown paper with a faint printed checkerboard pattern. As the textured patterns are on a similar scale to the GIFOV of the AisaFENIX, it is likely that most pixels will view mixed spectra. This is not a problem if the spectra from a region are averaged as the average reflectance will be a linear mixing of the spectra of alternately coloured checkerboard squares (Bioucas-Dias et al. 2012). The when averaging a single column of pixels, the worst SNR will be achieved when there is an odd number of whole checkerboard squares that are imaged.

Increasing the spatial frequency to T3 from T4 meant that for a cylinder of length 100mm the SNR along one pixel column would be 44:1 (approx. 44 × T3 squares +1 extra) as opposed to 22:1 (approx. 22 × T4 squares +1 extra). The 5 cylinders are placed as closely to the reference gauge as possible to minimise effects of polynomial extrapolation in the later hyperspectral image unwarping stage. The six tungsten halogen lights were warmed up for 40mins and allowed to stabilise before any imaging took place. Due to the unavailability of a dark room, the hyperspectral images were captured in minimal ambient lighting conditions. The images captured for SFM reconstruction we captured in ambient lighting conditions. The reflectance spectrum of the matte paper and black colouring of the paper were accurately determined using a hand-held spectrometer manufactured by ASD PANalitical by taking 8 spot measurements scattered over the paper and then averaging the measurements. The 800 spectral bands were then resampled to correspond to the 448 spectral bands of the AisaFENIX. Point clusters corresponding to the light filaments were averaged to determine all six locations of the lights modelled as point sources.

4.8.2 Assessment of Illumination Correction for Planar Surfaces

Similarly, to the cylinder illumination correction experiment this one aims to assess the upper limits of illumination correction on ideal planes, which also facilitates a comparison for illumination correction assessment between cylinders and planes.

The planes were covered with T3 patterned light brown paper, the same as the cylinders as a ground truth reading for plain white paper was difficult to obtain (showing measured reflectance values of greater than 1).

The 5 planes were arranged closely to the reference gauge to minimise any misalignment between the image and mesh due to extrapolation in the image resampling. The hyperspectral and RGB images were captured in the same manner as outlined in section 4.8.1. As earlier results showed high repeatability of camera calibration and camera movement, the same point locations of the lights captured in the previous cylinder experiment was used. After manual segmentation into planes and reference gauge, planar models were fitted using the PCL C++ API. The rigid transformation between the captured cloud and world coordinate system was determined through ICP fitting of the reference gauge cloud to a fiducial, artificially generated, aligned cloud. The planar models were transformed into the reference gauge coordinate system. The ray tracer only supports three primitives, not including a pure planar model. To compute beta and gamma values corresponding to the correct locations of the image, the planar model needed to be converted to a triangular mesh that delineated the outlines of the planes. This was achieved by finding the oriented bounding boxes of the segmented planes point cloud and projecting the four maximally divergent vertices onto the plane, thereby defining the four vertices of two coplanar triangles. The hyperspectral image processing followed the same procedure. Beta, Gamma, and c_1 values were determined for the hyperspectral image as outlined in the methodology chapter using the triangular mesh primitive without octree acceleration (as there were a total of 10 faces).

5 RESULTS

5.1 RAW DATA PRODUCTS

5.1.1 Raw Hyperspectral images

After the effects of bad pixels, and dark current are removed by pre-processing, as well as a relative scaling of wavebands to produce a relative radiance spectrum, we are left with a data cube with 384 pixels in the v direction, and a spectral range of 380nm to 2500nm acquired over 448 spectral bands. Below is presented images acquired in the 414nm (Fig.5.1), the 700nm (Fig.5.2), and the 2448nm (Fig.5.3) image bands. The bands at the extremities of the acquired spectrum suffer from the most noise, as can be seen in Fig.5.1 and Fig.32.

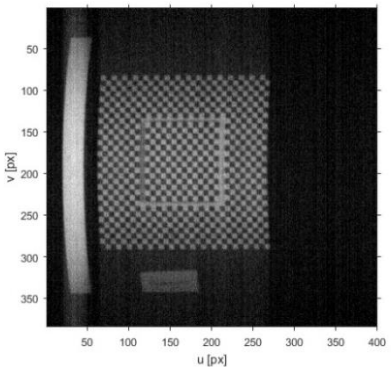

Figure 5.1 – Image Band 414nm

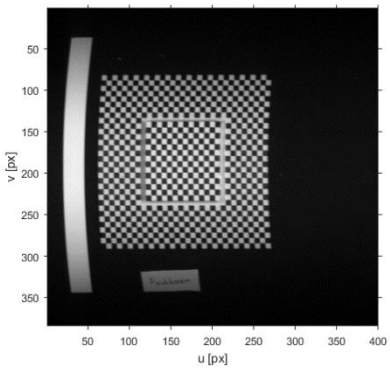

Figure 5.2 - Image Band 700nm

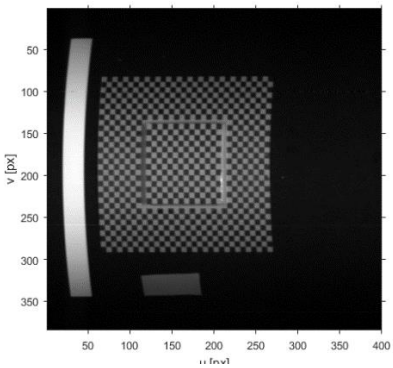

Figure 5.3 - Image Band 2448nm

Averaging the intensities of the white reference in the u-direction (in the 700nm band) and plotting a transect of these against the v direction shows a clear fall off of intensity across the white reaching an attenuation to 40% at the extremities of the white reference when compared to intensity in the middle (Fig.5.4). We cannot say that this is entirely due to intrinsic camera properties, as there may also be variation in the lighting pattern across the white reference. However, this is a significant effect that occurs without even any variation in topography.

Plotting the spectrum of a single pixel in the middle of the white reference yields the graph in Fig.5.5. The spectrum shows a spectral shape like that of black body radiation, which is expected from the incandescent tungsten halogen bulbs illuminating the scene.

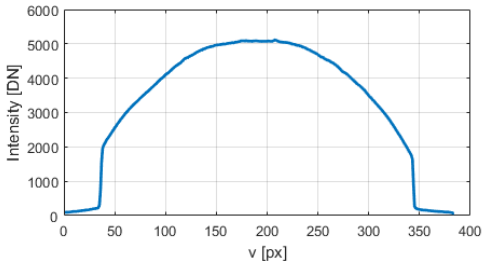

Figure 5.4 – v-axis Transect of White Reference

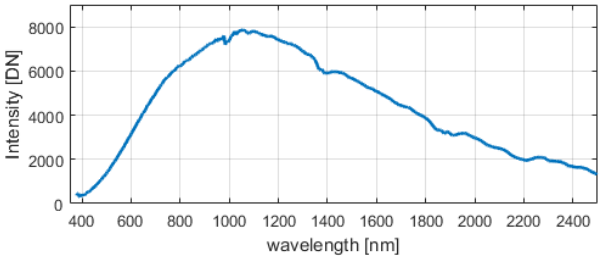

Figure 5.5 – Spectral Profile of White Reference

5.1.2 Raw Microsoft Kinect mesh

The ReconstructMe software uses a real-time SLAM algorithm to capture the scene and mesh it while simultaneously eliminating outliers. The point density is even but relatively poor at 0.2 vertices/mm^2. Flat surfaces are represented relatively well, while sharp edges are misrepresented showing significant amounts of smoothing (Fig.5.6). The raw mesh also shows significant misregistration between subsequent frames, this is most clearly seen at the white reference which appears to be erroneously fragmented into two almost separate objects (Fig.5.7).

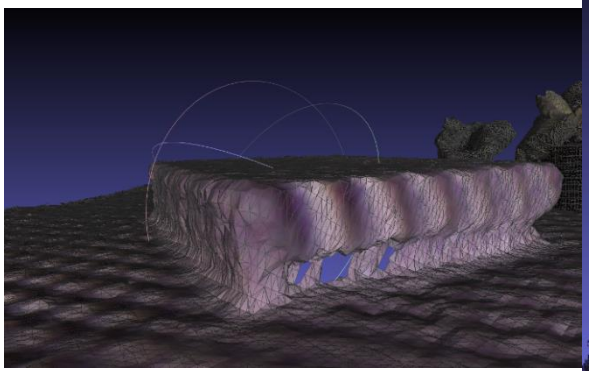

Figure 5.6 – Raw Microsoft Kinect Mesh

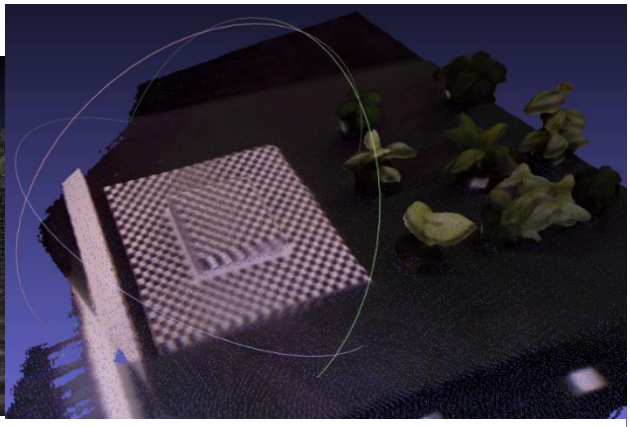

Figure 5.7 – Misalignment of White Reference

5.1.3 Raw SFM point cloud and simple Poisson mesh

The clouds captured by the structure from motion principle show uneven point density over the scene. The point density ranges between 0.6 to 0.8 vertices/mm^2.The carpet and smaller checkerboard squares show the greatest point density, while the larger checkerboard squares have no points at their centre, and the white reference shows the lowest point density corresponding to its lack of textural variation.

5.2 GEOMETRIC CALIBRATION EFFICACY

5.2.1 Correction of non-linear effects

1D and 2D polynomials are evaluated for non-linear effect removal, and the resulting camera model is compared with against manual measurements and intrinsic parameters listed in the data sheet.

5.2.1.1 Without non-linear effect removal

To evaluate the effects of non-linear effects on the camera calibration, the linear Pushbroom model was estimated accounting for only linear effects using the DLT procedure outlined in chapter 4. The camera matrix $\boldsymbol{M}$ was decomposed into extrinsic matrices $\boldsymbol{R_{wc}}$, $\boldsymbol{T_{wc}}$ and intrinsic matrix $\boldsymbol{K}$, from which the parameters f, p_v, V_x, V_y, V_z were extracted. The resultant camera model parameters are shown in Table 1. Matrix $\boldsymbol{T_{wc}}$ describes the position of the camera (in metres) at the start of its motion ($u = 0$)with respect to the reference gauge coordinate system. The roll, pitch, and yaw of the camera model was computed from $\boldsymbol{R_{wc}}$.

Table 1 – Pushbroom Model Extrinsic and Intrinsic Parameters

Extrinsic Parameters		**Intrinsic Parameters**	
$\boldsymbol{T_{wc}}\ [\boldsymbol{m}]$	$r, p, y\ [deg]$	$V_x, V_y, V_z\ [m]$	$f\ [px]$
0.1462	177.7	0.0019	516.4
0.2063	10.5	0.0000004	
0.9733	-0.02	-0.00003	

In terms of extrinsic properties: the positive x-coordinate of $\boldsymbol{T_{wc}}$ indicates that the camera started ahead of the reference gauge which certainly was not the case, the y-coordinate of $\boldsymbol{T_{wc}}$, 0.21 metres is in close agreement with reality as the camera was centred on the reference gauge at ~0.2 metres. However, the z-coordinate of 0.97 metres disagrees with the manually taken measurement which places the camera height at least higher than 1.2m. While the pitch angle of 10.5° disagrees with the observable close to nadir orientation, the 177.7° roll reflects the slight (observable) tilt of the art scanner setup. In terms of intrinsic properties, the focal length of 516.4 pixels, places the camera FOV at 40.8° contrary to the 32.3° reported in the camera manual. However, intrinsic direction vector $\boldsymbol{V}$ shows zero valued y, and z components as expected and an x component of close to 2mm per pixel line which agrees with the GIFOV of the camera at ~1.3m above the ground.

The error present in the linear model was estimated by projecting known 3D coordinates from the base plane of the calibration gauge into 2D image space by the estimated model $\boldsymbol{M}$, and determining the difference between these and actual locations of these points in the image: $e = x - \boldsymbol{M}\boldsymbol{x}$. The error in the residual quiver plot has been scaled by 4.5 to make the smallest displacements visible. A strong lateral trend is observable (Fig.5.8). The histogram plot (Fig.5.9) shows the distribution of absolute error magnitude, showing that it can reach up to 5 pixels, equivalent to approximately 10mm on the ground.

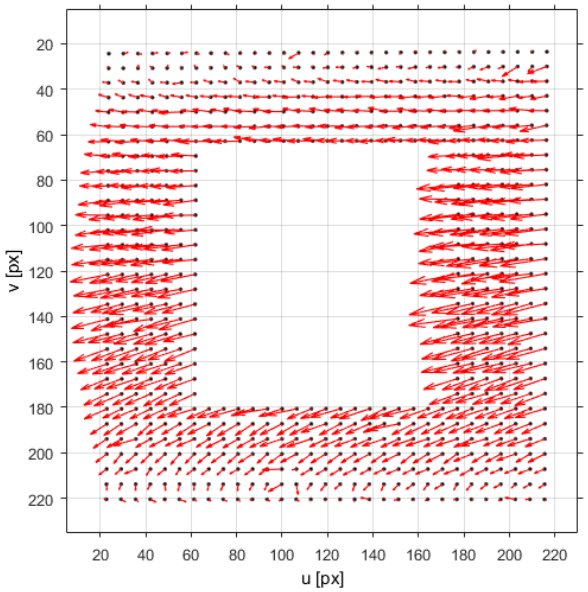

Figure 5.8- Quiver Plot of Residuals

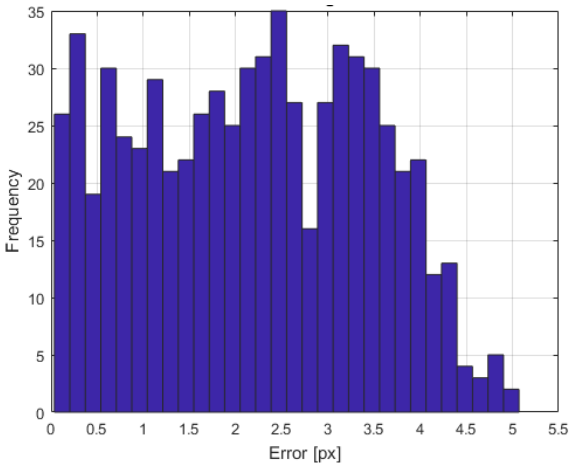

Figure 5.9 – Histogram of Residual Magnitude

5.2.1.2 1D polynomial non-linear effect removal

The residual error showed displacement in primarily the u-direction which varied strongly across the v-axis. To model and remove this error a one-dimensional cubic polynomial modelled the displacement error in the u direction as a function of position in the v direction was fitted to the residual error using least squares (Fig.39). This polynomial allowed both the unwarping of the image using an inverse mapping and the unwarping of the 12 points (using a forward mapping) used to estimate the linear pushbroom model. The linear pushbroom model was then re estimated from the 12 unwarped points using the DLT method outlined in Chapter 4. As before the camera model was decomposed into extrinsic and intrinsic matrices and relevant parameters were extracted for evaluation of the model and comparison of non-linear effect removal (Table 2).

Table 2 – Pushbroom Model from 1D Non-Linear Effect Removal

Extrinsic Parameters		**Intrinsic Parameters**	
$\boldsymbol{T_{wc}}$ $[\boldsymbol{m}]$	r, p, y $[deg]$	V_x, V_y, V_z $[m]$	f $[px]$
-0.0165	177.7	0.0019	508.7
0.2097	1.03	-0.00000135	
0.975	0.42	0.00000192	

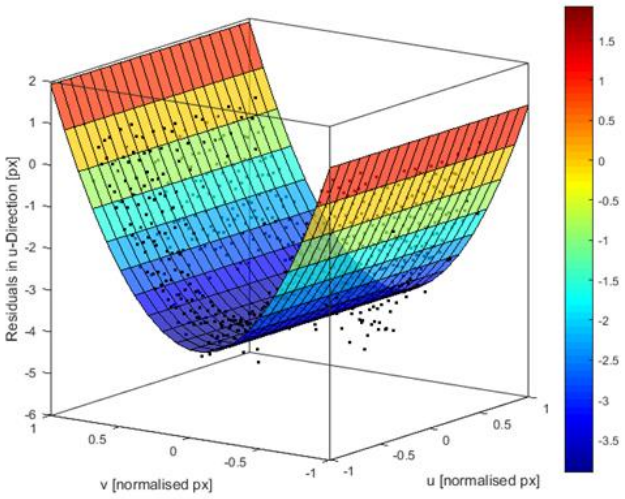

Figure 5.10- 1D 3rd Order Polynomial Fitted to Residual Error in v Direction

The extrinsic parameters are noticeably improved in comparison to no correction for non-linear effects. The starting position T_{wc} has a negative x-coordinate indicating agreement with the camera starting behind the reference gauge, the y-coordinate correctly indicates the camera centring on the reference gauge, however, the z-coordinate is still too small indicating the camera was much closer to the reference gauge than. In comparison to no correction, the pitch angle has reduced from 10.5° to 1.03° showing much closer agreement to reality. In terms of intrinsic parameters, the direction vector $\boldsymbol{V}$ is almost unchanged. The focal length f is 508.7 pixels, which would estimate that the cameras FOV is 41.4° which disagrees with the actual FOV of 32.3° (Specim 2013) .

After correction non-linear effect removal, the error still within the system was assessed by reprojecting corrected points to the image space by the pushbroom model and finding the difference between these and

actual recorded image coordinates. The reprojection error magnitude is reduced from a maximum of 5px to less than 1.4px (Fig.5.12). From the quiver plot (again error is scaled by 4.5 for viewing), while the error in the u-direction has been reduced, the error is mainly focussed in the v −direction (Fig.5.11.).

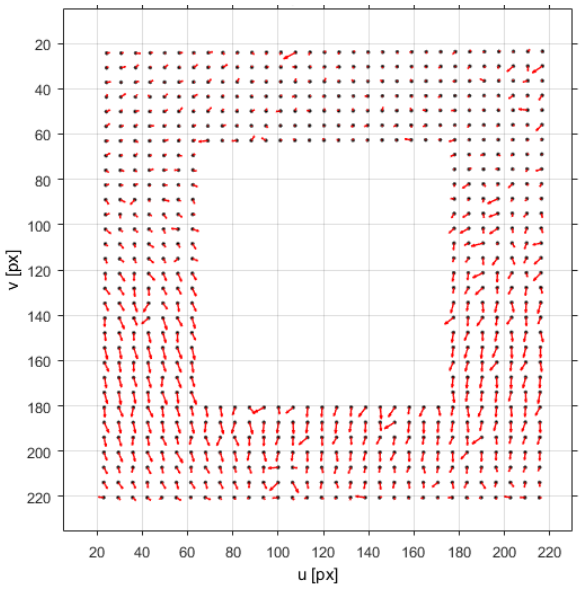

Figure 5.11 – Quiver Plot of Reprojection Error after 1D Non-Linear Error Removal

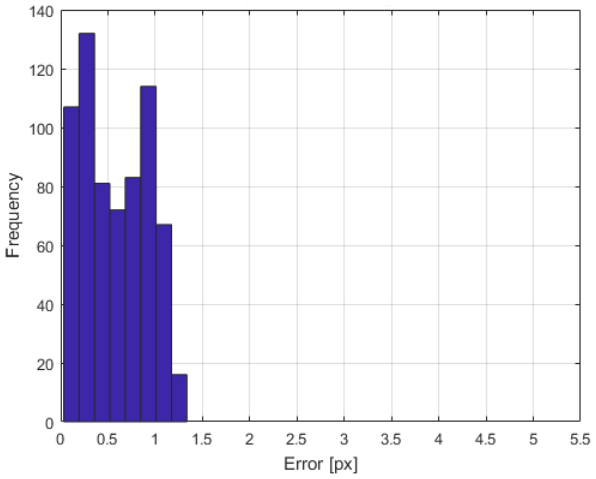

Figure 5.12 – Histogram of Reprojection Error Magnitude

5.2.1.3 2D polynomial non-linear effect removal

Unlike the 1D polynomial correction, 2D polynomial correction models the error in one of the dimensions as a function of position along both the u and v axes. As there is non-negligible non-linear error in both the u and v-directions two 2D polynomials were estimated to model the error in both u and v-directions (Fig.5.13). After this error was characterised both the image and originally detected 12 points were used to re-estimate the linear pushbroom model. Again, the 3x4 camera matrix $\boldsymbol{M}$ was decomposed into its constituent $\boldsymbol{R_{wc}}, T_{wc}, \boldsymbol{K}$, matrices using RQ decomposition.

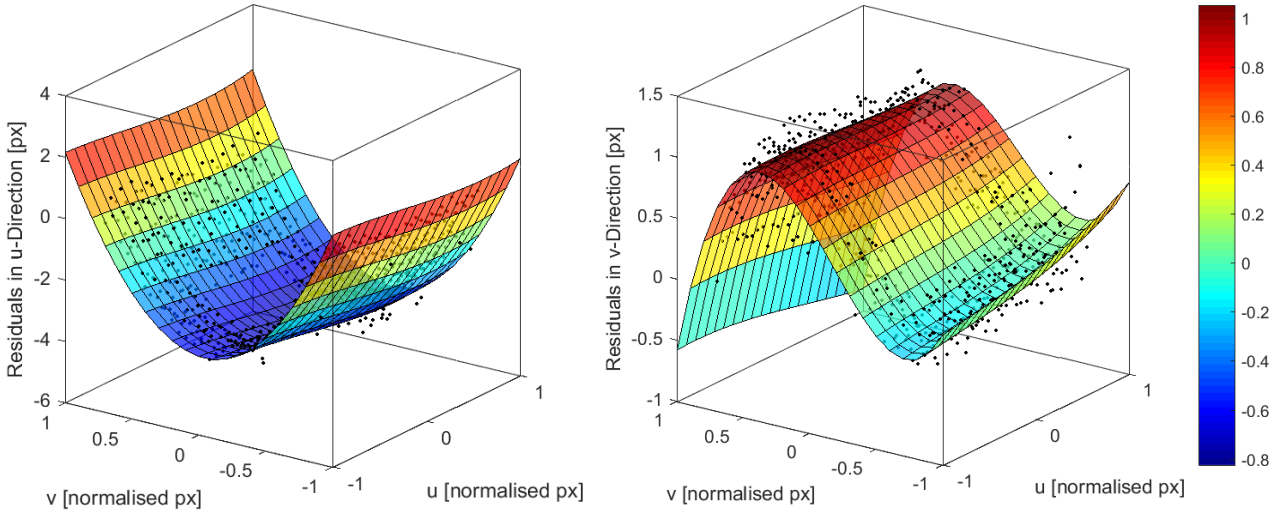

Figure 5.13 – 2D 3rd Order Polynomials Fitted to Residuals in both u and v directions

Both polynomial functions for error in u and v-directions show the most significant variation with respect to the v-direction as expected, however, there is still observable variation with the u-direction(Fig.5.13 left). The resulting camera model (Table 3) is in closer agreement with observation of the setup. While the x and y coordianates of $\boldsymbol{T_{wc}}$ are fairly like before, the z coordinate is now in agreement with manual measurements of the cameras height. In terms of orientation, both the pitch and yaw are less than half a degree from ideal values. The roll is less than 5 degrees from the ideal value, however, this still agrees with the slight tilt observable in the setup. $\boldsymbol{V}$ again, is unchanged, and the focal length is 662.0 pixels placing the FOV 32.35° which closely agrees with the 32.3° reported in the data sheet for the AisaFENIX (Specim 2013).

Table 3 – Pushbroom Model Parameters after 2D Polynomial Correction

Extrinsic Parameters		**Intrinsic Parameters**	
$\boldsymbol{T_{wc}}\ [\boldsymbol{m}]$	$r, p, y\ [deg]$	$V_x, V_y, V_z\ [m]$	$f\ [px]$
-0.0229	175.2	0.0019	662.07
0.2146	0.488	-0.0000023	
1.281	0.425	0.0000028	

After correcting for non-linear effects with 2D polynomials the reprojection error was computed. As can be seen in the residual magnitude histogram, the error reaches a maximum value of ~1.3 pixels while the median error is less than 0.3 pixels (Fig.5.15). The quiver plot indicates the both the reduced magnitude of the error and random distribution of error direction indicating that most of systematic error has been removed (Fig.5.14).

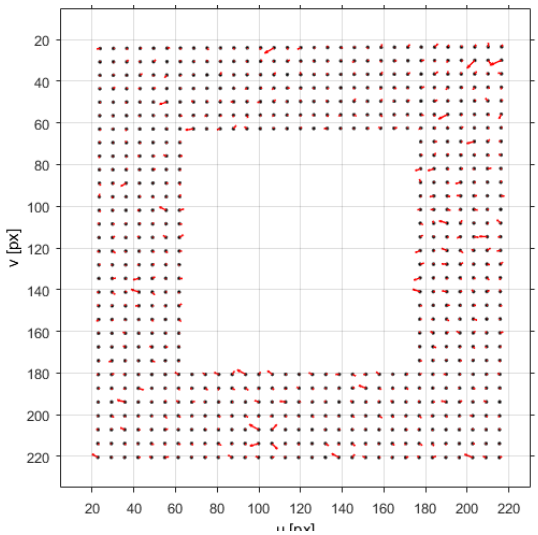

Figure 5.14 – Quiver Plot of Reprojection Error

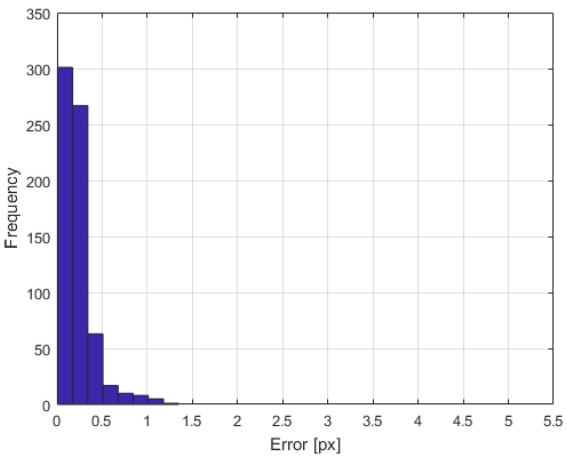

Figure 5.15 – Histogram of Reprojection Error Magnitude

5.2.2 Wavelength dependency of error

This section presents the results obtained from the methodology previously explained in section 4.4.3 on the assessment of wavelength dependent error. Pushbroom hyperspectral cameras are known to suffer wavelength dependent spatial distortions/aberrations such as keystone and smile distortion (Lawrence et al. 2003). To assess the wavelength dependency of the error, the locations of all 672 internal corner points on the base plane of the calibration gauge were found using the MATLAB Harris corner points detector. For each waveband all 672 coordinates were averaged in both u, and v-directions. The resulting plot (Fig.5.16) shows the deviation of the averaged coordinate of each waveband with respect to the average coordinate of all wavebands.

The most noticeable feature is the sharp change in average coordinate deviation between VIS/NIR and SWIR regions. This is attributable to the separate VIS/NIR and SWIR detector elements which are misaligned. The deviation of values within the VIS/NIR or SWIR regions of the spectrum are likely due to wavelength dependent

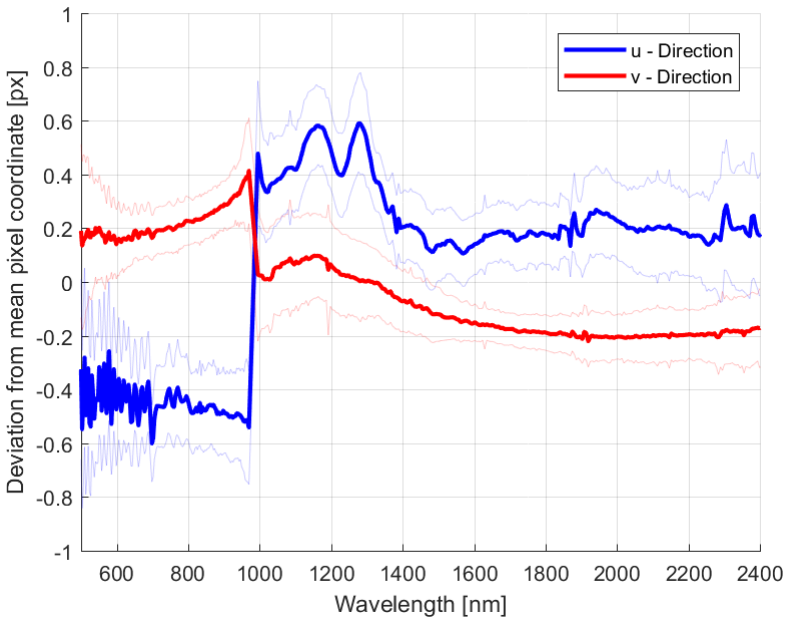

Figure 5.16 – Deviation from Mean Pixel Coordinate across all Wavebands

effects such as keystone and smiling distortion. The AisaFENIX data sheet (Specim 2013) reports keystone and smile effects to have a standard deviation of up to 0.2px. The VIS/NIR region in the v-direction shows a maximum variation range of 0.25px, while the SWIR region shows a deviation range of 0.3px which both indicate standard deviations below 0.25px. However, in the u-direction of the VISNIR region the deviation range is ~0.3px, while the deviation range of the SWIR region is ~0.5px.

5.2.3 Camera model stability

The calibration stability of the camera will have a bearing on multiple factors such as time between recalibrations and whether a calibration object will be required during the imaging process. The calibration stability is assessed by evaluating the variation in both intrinsic and extrinsic camera model parameters obtained over subsequent calibration procedures. Non-linear effects were removed with two 2D Chebyshev polynomials, upon which the camera model was computed, subsequently the extrinsic and intrinsic parameters were retrieved by decomposing the camera model using RQ decomposition. Table 4 presents the results obtained from application of methodology as outlined in section 4.4.2.

Table 4– Mean and Standard Deviation of Camera Model Parameters

Extrinsic Parameters				**Intrinsic Parameters**			
$\boldsymbol{T_{wc}}$ $[\boldsymbol{m}]$		r, p, y $[deg]$		V_x, V_y, V_z $[m]$		f $[px]$	
Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
-0.0216	0.0022	174.7	0.293	0.001945	0.00000266	662.07	4.578000
0.2161	0.0016	0.495	0.0896	0.0000193	0.00000309		
1.278	0.0084	0.353	0.0943	0.0000229	0.00000548		

Considering first the stability of parameter $\boldsymbol{T_{wc}}$, we observe that the standard deviations are in the orders of 8 millimetres or less for all x, y, z components. Roll pitch and yaw have deviations in the order of less than 0.3° which also seems reasonable given the small amount of vibration and rotation as the sensor travels on the rails. On the other hand, the repeatability of linear motion is better than 0.005mm in all x, y, z components. The average focal length estimates an average field of view of 32.5° with standard deviation of 0.2° making this consistent with the 32.35° reported in the AisaFENIX data sheet (Specim 2013).

5.3 Suitability of Point Clouds and Meshes for Illumination Correction

5.3.1 Evaluation of angle bias

Here the results of the planar angle bias assessment (as outlined in section 4.6.1) are presented (Table 4 and Table 5): The T1 texture corresponding to 0.55mm/square failed to reconstruct any sort of reasonable plane structure during the meshing step in SFM pre-processing and was omitted for both the Microsoft Kinect and SFM data sets due to the long manual pre-processing steps and checks required. However, the textures T2,T3, T4, and T5 corresponding to [1.1,2.2,4.4,8.8] mm/square could be reasonably reconstructed.

Table 4 – Angles of Planes as Measured by SFM and Microsoft Kinect Technologies.

Angles	**A0**		**A10**		**A20**		**A30**		**A45**	
Texture	SP	KP	SP	KP	SP	KP	SP	KP	SP	KP
T2	3.20	0.70	7.10	11.70	19.50	22.20	32.50	32.80	38.70	46.70
T3	2.90	1.50	8.70	10.60	18.50	21.10	29.40	32.80	44.60	47.10
T4	2.20	1.10	9.30	11.50	20.20	21.90	30.70	32.40	45.20	46.40
T5	1.50	0.80	10.70	12.00	21.30	22.70	31.70	35.20	46.40	48.30

Table 5 – Difference in Angles of Planes as Measured by SFM and Microsoft Kinect Technologies.

Angle	**A0**		**A10**		**A20**		**A30**		**A45**	
	SP	KP	SP	KP	SP	KP	SP	KP	SP	KP
T2	3.20	0.70	-4.40	0.20	-2.00	0.70	1.20	1.50	-5.90	2.10
T3	2.90	1.50	-2.80	-0.90	-3.00	-0.40	-1.90	1.50	0.00	2.50
T4	2.20	1.10	-2.20	0.00	-1.30	0.40	-0.60	1.10	0.60	1.80
T5	1.50	0.80	-0.80	0.50	-0.20	1.20	0.40	3.90	1.80	3.70

When evaluating performance with texture SFM shows the worst angle bias with texture T2, followed by T3, and performs the best on textures T4, and T5. The Kinect performs best on T4, and performs worse on all other textures. In terms of angle, the best results for the Kinect are acquired on planes A10, and A20 and generally performs worse on with deviation from these angles. On the other hand, the structure from motion approach performs best on angle A30 from which results appear to get progressively worse as the angle deviates. In terms of technology, both show a similar range of susceptibility to bias, however, when the Kinect sustained a better performance in terms of resilience to bias over A0 to A45 for T4 than was demonstrated by SFM for any texture.

5.3.2 Evaluation of mesh angular deviation for planes

Here are presented the results obtained by application of methodology as outlined in section 4.6.2. The strong smoothing effects at the edges of the mesh caused deviation from planar geometry. Consequently, the mesh was manually segmented to remove the edges before the planes panes were fitted, and before the angular deviation was quantified. Qualitatively the surface produced by the meshing of the SFM point clouds appears reasonably smooth, with little undulation in the region of the plane itself.

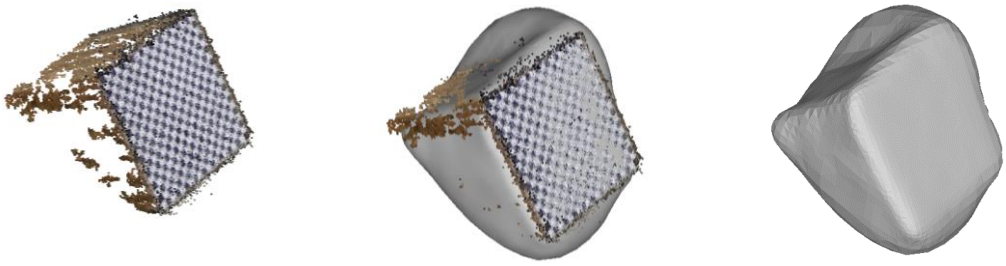

Figure 5.17 – Poisson Meshing of SFM point cloud

Texture T1 (0.55mm per square) produced point clouds that were empty or sparsely populated within the planar region, and consequently were not further processed. For each texture 15 meshes gathered over the five planes with three repeats were used to gather an aggregate measurement. Shown below is the normalised histograms for each of the textures (Fig.5.18 to 5.21).

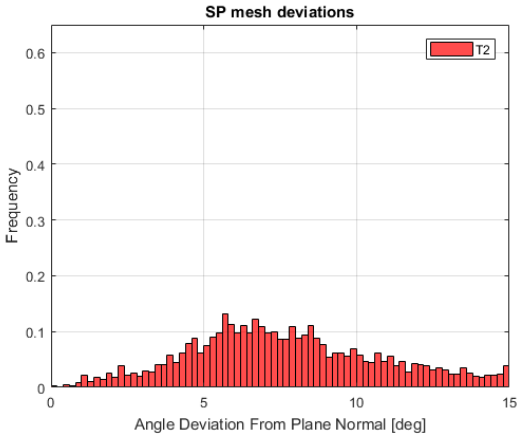

Figure 5.18 - Angular Deviation of SFM T2 Planar Mesh

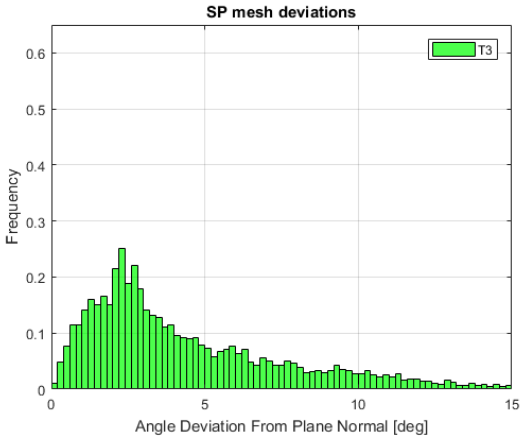

Figure 5.19 – Angular Deviation of SFM T3 Planar Mesh

The SFM distributions are right skewed, with T4 (Fig.5.20) and T5 (Fig.5.21) showing the best concentration of deviations below 5 and 10 degrees. T3 performs worse with a greater percentage of values above 5 and 10 degrees than T4,T5, while T2 (fig5.18) exhibits the worst performance with the flattest distribution that has a greater median than the other textures.

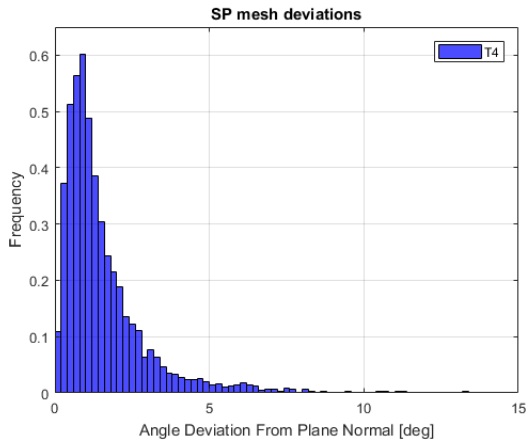

Figure 5.20 - Angular Deviation of SFM T4 Planar Mesh

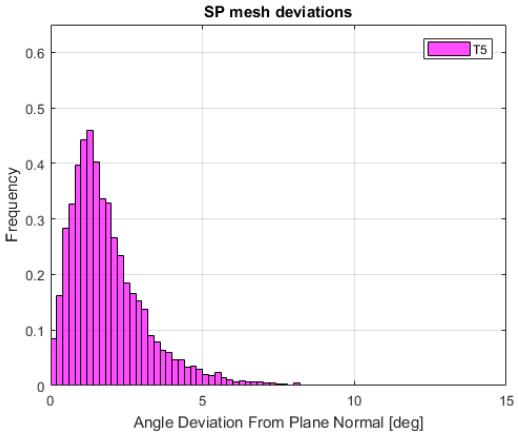

Figure 5.21 - Angular Deviation of SFM T5 Planar Mesh

The same process was applied for the Microsoft Kinect point clouds. As both acquisition and pre-processing Microsoft Kinect raw data is significantly less labour intensive, and the Microsoft Kinect produces a coarser mesh, five repeats were used instead of three. Qualitatively, the Microsoft Kinect mesh is coarser, and greater surface undulation is apparent.

Figure 5.22 – Mesh of Planar Surface Obtained with the Microsoft Kinect

The histograms are shown for textures T2 to T5 (Fig.5.23 to Fig.5.26). Like the SFM results. The histograms reveal right skewed distributions, with the distribution flattening as the spatial frequency of texture decreases. The histograms show distributions more favourable than SPT2 (Fig.5.18), however, all the Microsoft Kinect distributions show greater concentrations of deviations beyond 5 degrees, and 10 degrees than SPT4 (Fig.5.20) or SPT5 (Fig.5.21).

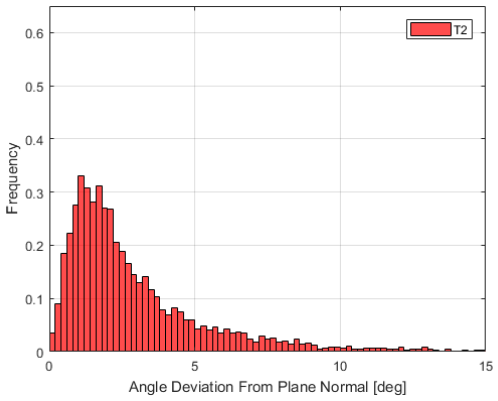

Figure 5.23- Angular Deviation of Microsoft Kinect T2 Planar Mesh

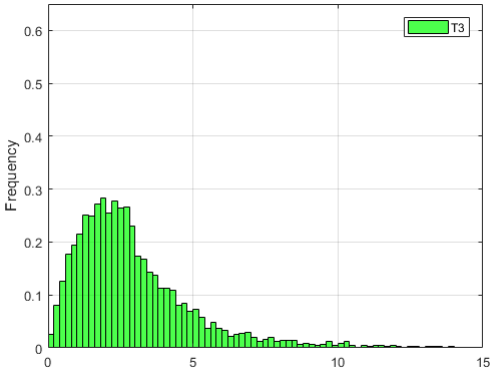

Figure 5.24 - Angular Deviation of Microsoft Kinect T3 Planar Mesh

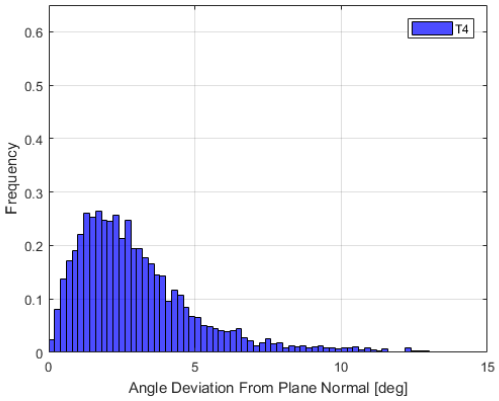

Figure 5.25 - Angular Deviation of Microsoft Kinect T4 Planar Mesh

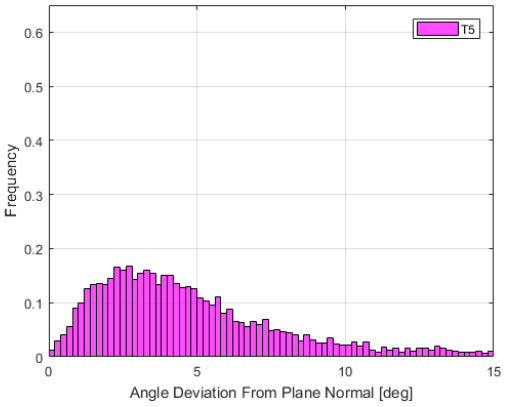

Figure 5.26- Angular Deviation of Microsoft Kinect T2 Planar Mesh

Table 6 – Statistical Description of the Angular Deviation of Planar Meshes

	KP					SP				
	Median	IQR	90th	95th	99th	Median	IQR	90th	95th	99th
T2	2.13	2.49	6.46	8.55	15.86	8.47	7.51	20.71	24.44	31.35
T3	2.47	2.18	5.45	7.00	11.30	3.46	4.32	10.30	12.93	25.43
T4	2.54	2.35	5.73	7.33	10.92	1.13	1.26	3.21	4.65	8.18
T5	4.14	4.00	10.02	13.18	18.64	1.57	1.44	3.52	4.41	6.46

The Microsoft Kinect performs worst on T5 (Fig.5.26), and comparably on T2 (Fig.5.23),T3 (Fig.5.24), and T4 (Fig.5.25) in terms of both spread and median. Structure from motion appears to be much more affected by the spatial frequency of texture, showing very poor results for the finer T2 texture and best results in terms of both median and spread for T4.

5.3.3 Evaluation of mesh angular deviation with radius

Here are presented the results acquired following the methodology as outlined in section 4.6.3. Section 5.3.2 demonstrated that texture T4 produced favourable results in terms of mesh deviation in meshed planes for both structure from motion and Microsoft Kinect technologies, hence texture T4 was used for the covering of the cylinders.

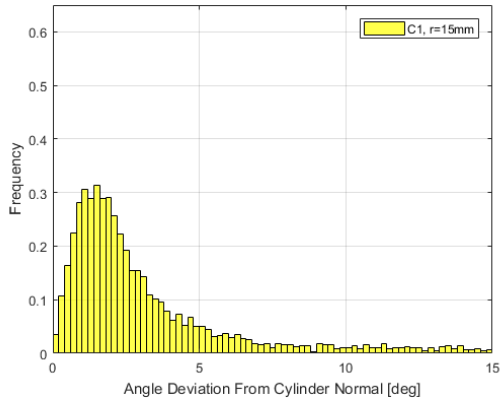

Figure 5.27- Angular Deviation SFM C1 Mesh

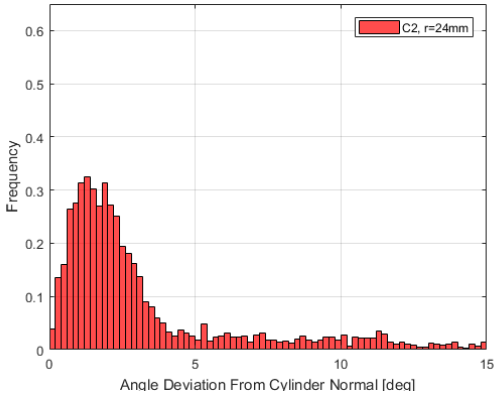

Figure 5.28 - Angular Deviation SFM C2 Mesh

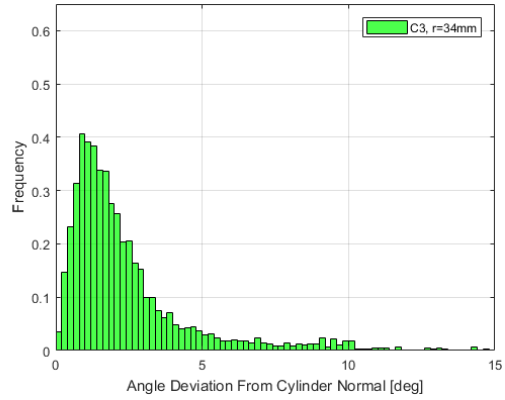

Figure 5.29- Angular Deviation SFM C3 Mesh

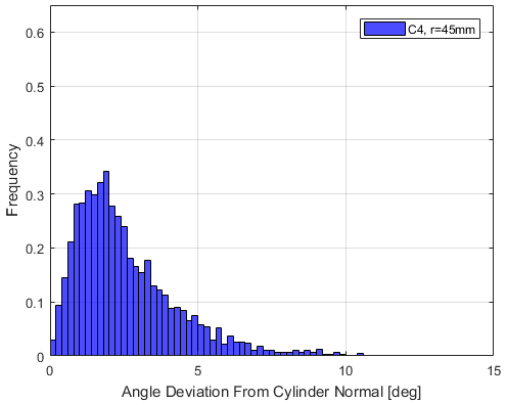

Figure 5.30 - Angular Deviation SFM C4 Mesh

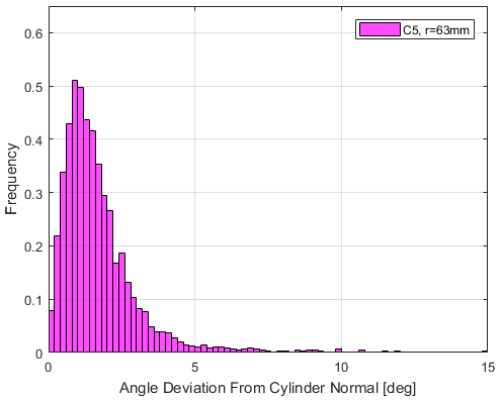

Figure 5.31 - Angular Deviation SFM C5 Mesh

Three repeats were acquired over C1 to C5 for SFM, and five for the Microsoft Kinect for the same reasons as outlined in section 5.3.2. The repeats for each cylinder were aggregated, and the normalised histograms are

shown for each. Cylinder C5 (Fig.5.31), shows the most favourable distribution for illumination correction with both the smallest median and interquartile ranges. While the median value tends to remain comparable for the other four cylinders, the spread (interquartile range, and percentiles) tend to increase as radius decreases.

The RANSAC cylinder model fitting failed to generate acceptable fitted cylinder models for the C1 and C2 Microsoft Kinect cylinder meshes and so were excluded to acquire fair, comparable results. The following normalised histograms for C3 to C5 are shown below.

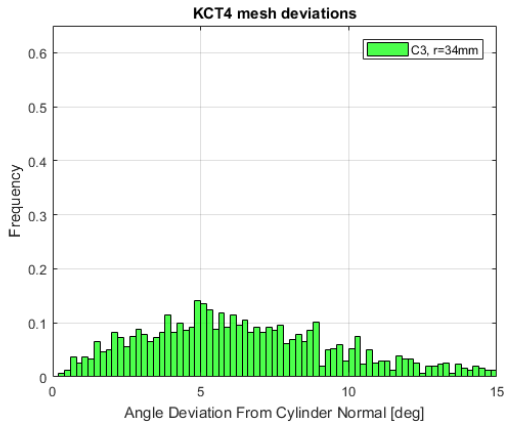

Figure 5.32 - Angular Deviation Microsoft Kinect C3 Mesh

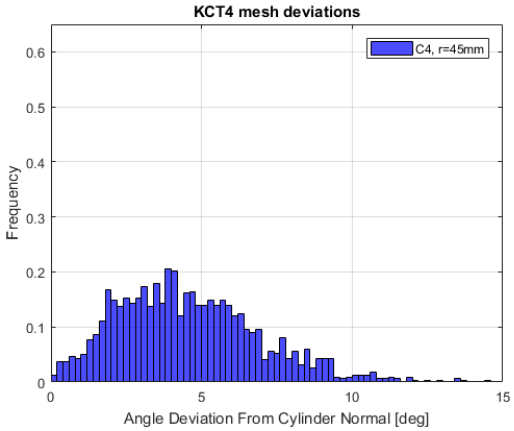

Figure 5.33 - Angular Deviation Microsoft Kinect C4 Mesh

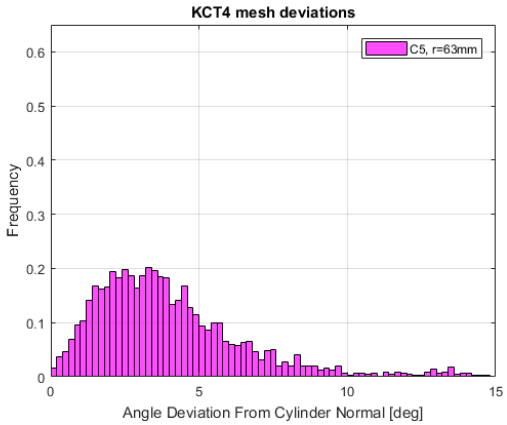

Figure 5.34 - Angular Deviation Microsoft Kinect C5 Mesh

Both the medians and the interquartile ranges tend to increase as cylinder radius decreases(Fig.5.32 to Fig.5.34). The SFM and Poisson meshing performs consistently better than the Microsoft Kinect meshes with much lower median deviation, lower interquartile range and consistently less deviation for equivalent percentiles (Table.7).

Table 7- Statistical Description of the Angular Deviation of Cylindrical Meshes

	KCT4					SCT4				
	Median	IQR	90th	95th	99th	Median	IQR	90th	95th	99th
C1	NA	NA	NA	NA	NA	2.14	2.55	7.74	11.61	16.57
C2	NA	NA	NA	NA	NA	2.07	2.31	9.30	11.53	15.23
C3	6.66	5.93	17.37	23.06	33.20	1.74	1.81	5.25	8.33	16.15
C4	4.29	3.21	7.78	8.95	11.40	2.13	2.04	4.91	6.01	8.74
C5	3.59	3.18	8.20	12.92	26.83	1.40	1.26	3.18	4.19	8.96

5.4 Efficacy of Illumination Correction

5.4.1 Evaluation of Cylinder illumination correction

5.4.1.1 Results of illumination correction

After an initial scaling of the point clouds to within 5% of the correct scale, scaling algorithm described in section 4.5.1.1 was applied to refine the scaling accuracy. A manual verification of the scaling process found that the scaling had produced an error of ~4%. To proceed with the experiment the point clouds were rescaled by manually taking 10 distance measurements of known distances over the cloud, averaging, and then manually computing a scaling factor. The error of the manually scaled cloud was verified to be <1%. The following meshing, ICP alignment and cylinder fitting processes were executed as planned. The Microsoft Kinect cylinders C1 and C2 reconstructed poorly and fitted cylinder models were not representative of the mesh geometry. Consequently, these were excluded from further assessment.

Presented below are images in the waveband of 700nm from various stages in the illumination correction processing routine. The raw radiance image obtained after bad pixel removal and dark frame subtraction (Fig.5.35). To the right is the geometrically rectified image obtained by modelling the non-linear effects with 2D 3rd order Chebyshev polynomials (Fig.5.36).

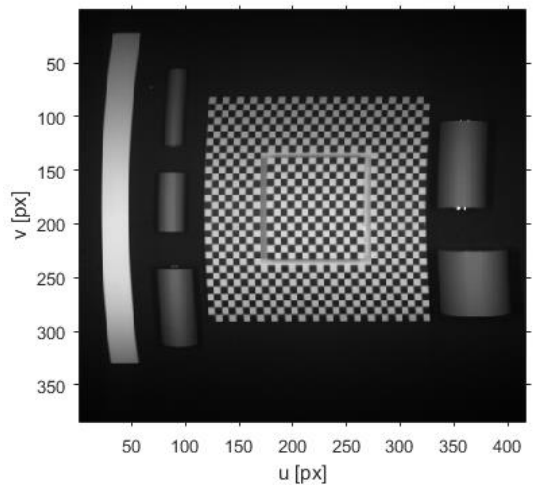

Figure 5.35 – Raw Hyperspectral Image (700nm)

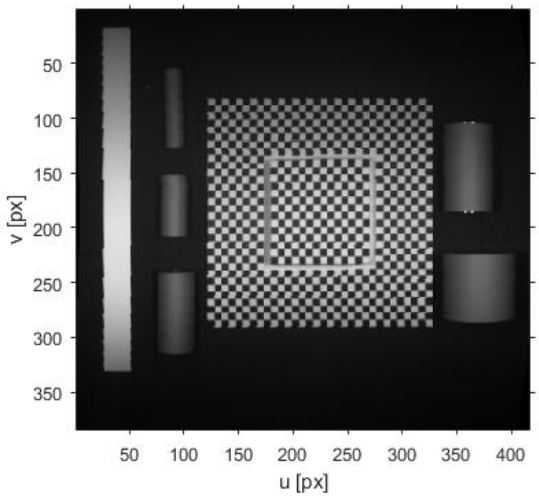

Figure 5.36 – Geometrically Rectified Image (700nm)

The vignetting and detector gain effects are estimated in factor c_1 for all detector elements over all wavebands. Correcting the image by dividing each row by its corresponding c_1 factor (Fig.5.37).

In Fig.5.38 is the predicted Lambertian illumination factor β for each pixel, based purely on the ideal cylinder models and modelled position of the six halogen lights. Qualitatively the predicted illumination shows general agreement with the illumination pattern over the cylinders. The main qualitative difference is the presence of two vertical highlights (especially evident on C4) in the real scene.

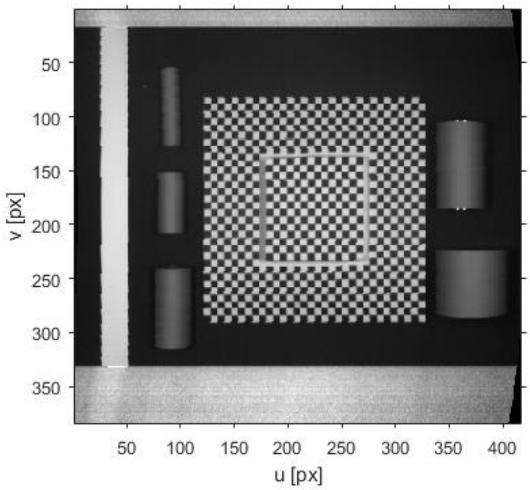

Figure 5.37 – Removal of Vignetting (700nm)

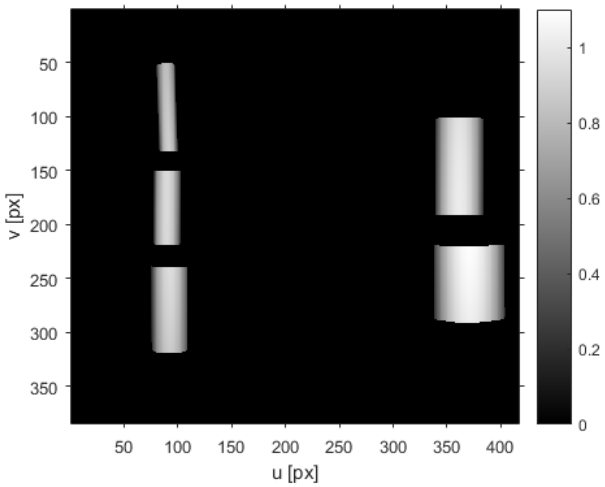

Figure 5.38 – Predicted Illumination factor β (Lambertian Only)

Fig.5.39 shows the predicted γ values for each pixel in the image based on the mesh and modelled illumination. It is expected that the closer to zero the γ, the more likely there is an occurance of specular reflection in the image. The pattern of low gamma angles in Fig.5.39 corresponds well with the observed specular highlights in Fig.5.36. The final Lambertian corrected image shows a much more homogenous pixel values in the region of the cylinders with an exception of overcorrected values at the cylinder edges.

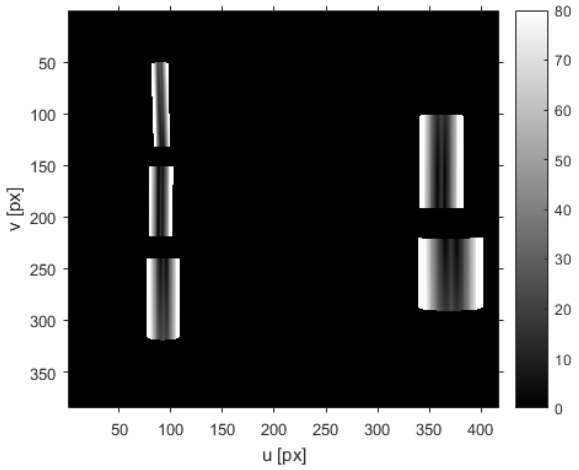

Figure 5.39 - Predicted Factor γ

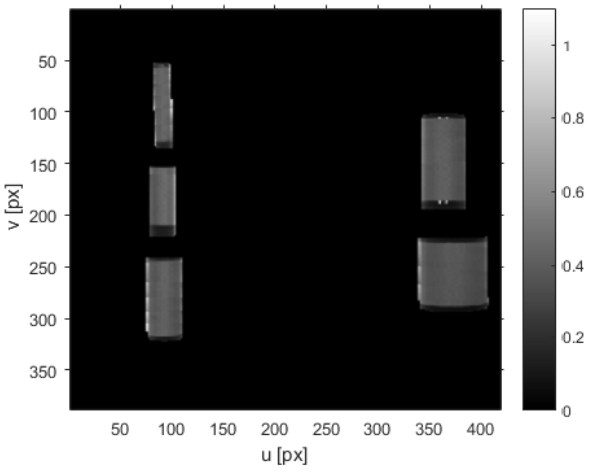

Figure 5.40 – Lambertian Corrected Image (700nm)

5.4.1.2 *Illumination correction over the cylinder profile*

The following graphs show the average values at the 700nm band over a horizontal transect of each of the cylinders. Shown are the Lambertian correction factor (green), the Lambertian corrected reflectance (blue), the original uncorrected reflectance (red) and the ground truth reflectance from hand held spectroscopy (pink). The Lambertian corrected reflectance value is relatively flattened out when compared to the original uncorrected reflectance. Moreover, the middle of the corrected reflectance curve has slightly greater values than to either side. This is consistent with the a purely Lambertian correction that does not account for specular factors.

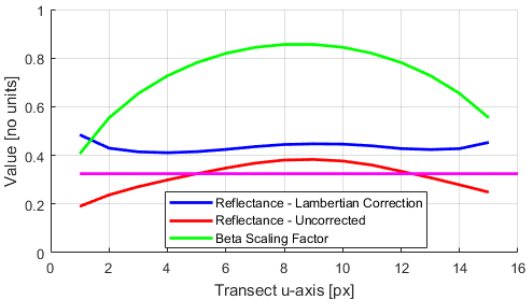

Figure 5.41 – Horizontal Transect of C1 at 700nm

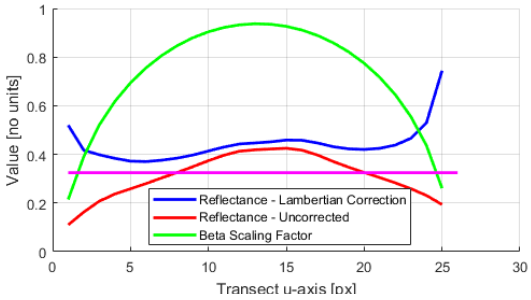

Figure 5.42 – Horizontal Transect of C2 at 700nm

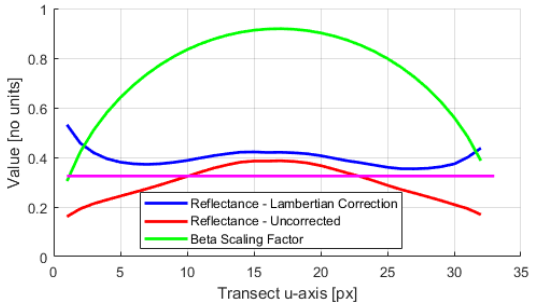

Figure 5.43 – Horizontal Transect of C3 at 700nm

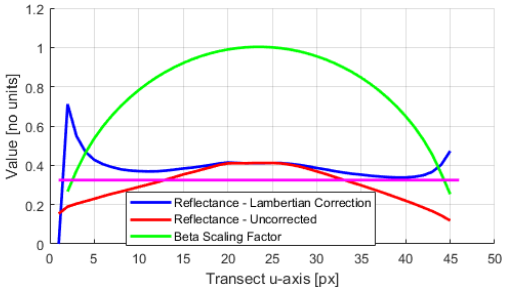

Figure 5.44 – Horizontal Transect of C4 at 700nm

More evident in the transects of C1 (Fig.5.41) and C2 (Fig.5.42) is the horizontal misalignment between the peaks of the predicted Lambertian factor and the uncorrected reflectance which suggests misalignment between the cylindrical models and the image. The unexpectedly large values at the horizontal extremities indicate that the correction underestimates the illumination at the edge of the cylinders. The corrected reflectance value also deviates from the ground truth value, while the uncorrected reflectance curve is centred on the ground truth value.

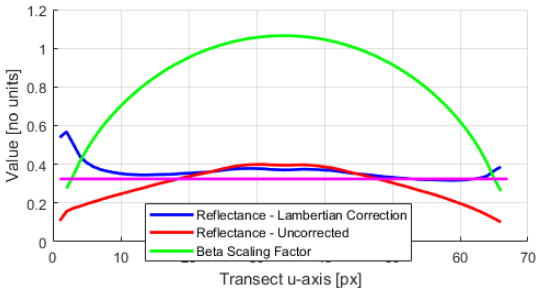

Figure 5.45 – Horizontal Transect of C5 at 700nm

5.4.1.3 Corrected illumination spectra for each cylinder

Fig.5.46 shows the uncorrected spectral of C1 through C5 plotted over 400nm to 2500nm, which is compared to the ground truth spectra of the paper (pink). This first graph in Fig.5.46 shows that the average spectra of C1 through C5 have a much smaller spread than the ground truth. This is expected as most of the cylinder would expect to have lower measured reflectance than the ground truth values, yielding lower reflectance, and lower average spectral spread. With the Lambertian correction the 1400+nm SWIR region shows much greater consistency between cylinders and the ground truth, while the VNIR region shows less consistency with each other and with respect to the ground truth (Fig.5.47). However, the spectral spread of both the VNIR and SWIR regions is more consistent with the ground truth.

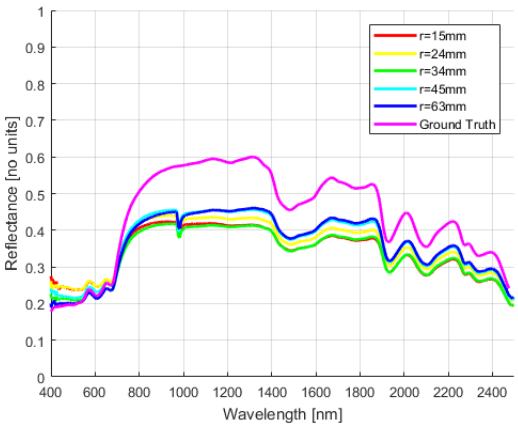

Figure 5.46– Uncorrected Reflectance Cylinders

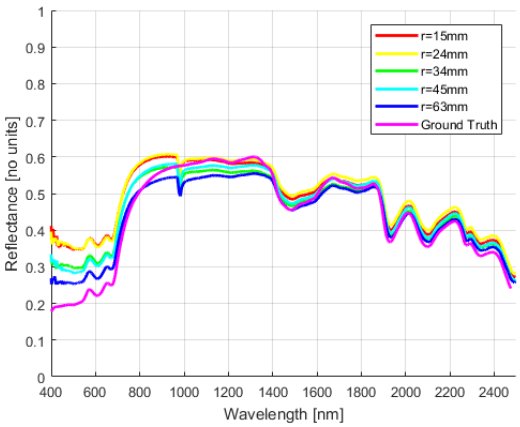

Figure 5.47 – Lambertian Corrected Reflectance of Cylinders

Specular correction based on estimating a common reflectance shift, yield spectra between the cylinders which are much more consistent with each other (Fig.5.48). There is a difference in spectral shape between all the cylinder spectra and the ground truth spectra especially in the below 1400nm region. Fig.5.49 compares different thresholds applied on the gamma value to determine if a specular correction should be applied. The resultant spectra for all cylinders was averaged for each γ threshold value applied. Interestingly there is very little difference over the range of γ thresholds.

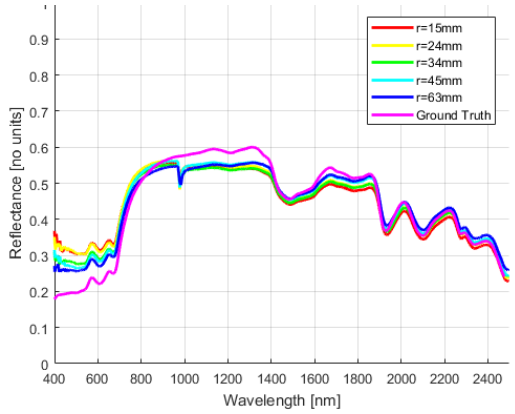

Figure 5.48 – Specular Corrected Reflectance (γ Thresh = 180°)

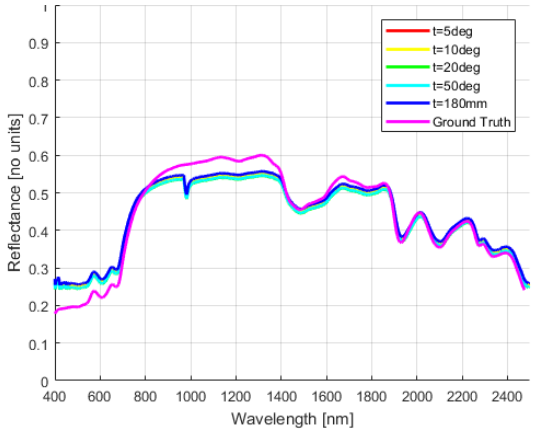

Figure 5.49 – Specular Correction for Differing γ Thresholds

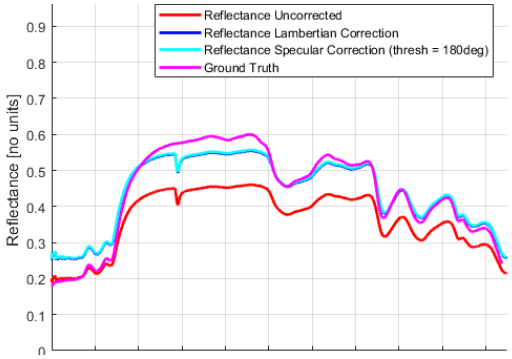

Figure 5.50 – Comparison of Specular and Lambertian Corrections

Averages of all values were taken over all the cylinders. In red is the average of uncorrected spectral values across all cylinders, in blue is the average value of all Lambertian corrections over all cylinders, and in cyan is the average of all specular corrections which estimates a specular correction for every pixel (Fig 5.50). The Lambertian and specular corrections are approximately identical but show a clear improvement over the uncorrected spectra in terms of approximating the ground truth.

5.4.2 Evaluation of Plane illumination correction

5.4.2.1 Results of illumination correction

Fig.5.51 shows the raw hyperspectral image of the 700nm band, obtained after pre-processing steps of bad pixel removal, and dark frame subtraction. Shown in Fig.5.52 is the result of geometric rectification, and removal of multiplicative effects explained by factor c_1. On the left-hand side from the bottom to top are planes A10 and A0. On the right-hand side from bottom to top of the image are planes A20, A30, A45.

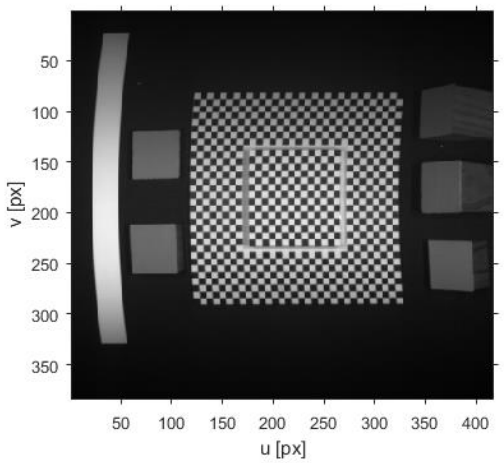

Figure 5.51- Raw Hyperspectral Image (700nm)

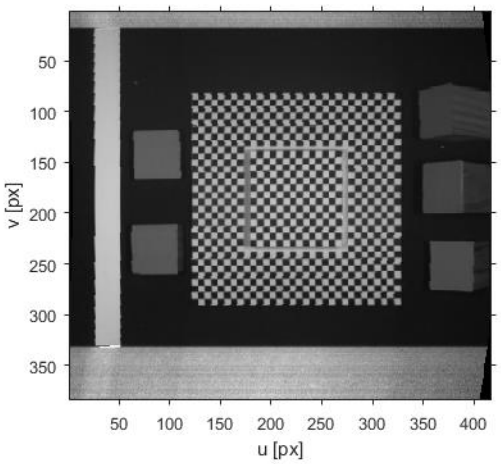

Figure 5.52- Undistorted with Vignetting Removal

Below is shown the predicted beta (Fig.5.53) and gamma (Fig.5.54) factors for the image based on ray tracing. Beta values decrease as the angle of inclination increases. Gamma values decrease as inclination angle decreases. A10 shows the lowest gamma value, which would be expected of A0. However, this may be explained by the small height variation between A0 and A10.

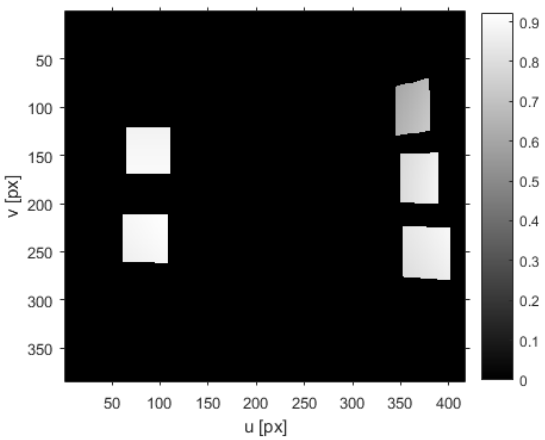

Figure 5.53- Predicted Factor γ

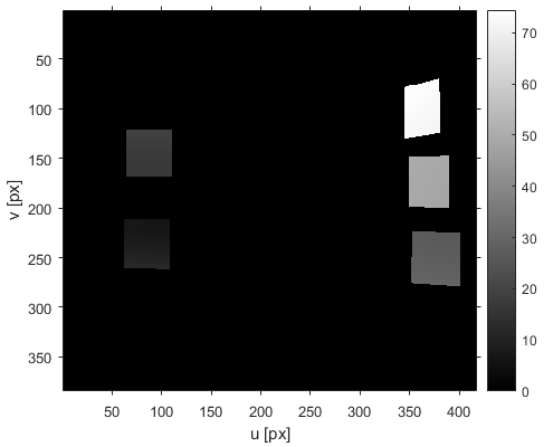

Figure 5.54- Predicted Lambertian Factor β

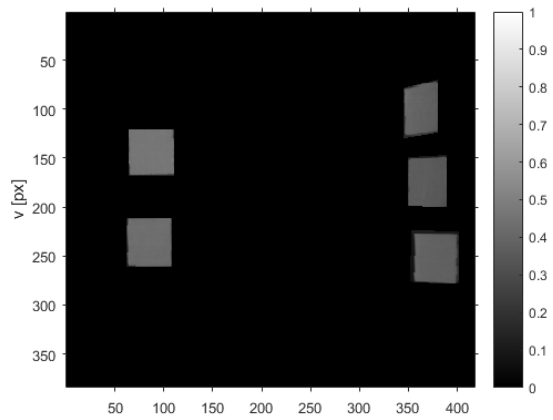

Figure 5.55- Lambertian Corrected Reflectance (700nm)

The Lambertian correction (Fig.5.55) appears to be reasonable, with all planes showing close to homogenous reflectance. Interestingly, unlike the cylinder Lambertian correction there are no over corrected regions (bright patches at the edges).

5.4.2.2 *Corrected illumination spectra for each plane*

As expected, as the inclination angle increases from A0 through to A45 the entire spectra appears to have less spread, while maintaining the same shape (Fig.5.56). Interestingly planes A0 and A10 show a shift in reflectance of ~10% in the 400-700nm range when compared to both the ground truth and the A20 to A45 uncorrected spectra (Fig.5.56). Lambertian corrections (Fig.5.57) show A30 through A45 regaining a spread much closer to the ground truth spectra, while A0 and A10 increase beyond ground truth values over all wavebands.

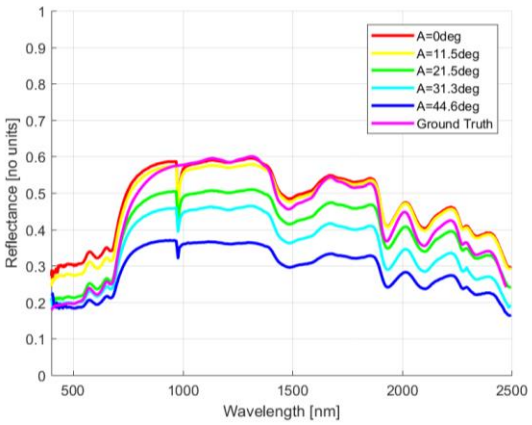

Figure 5.56 – Uncorrected Reflectance for Planes

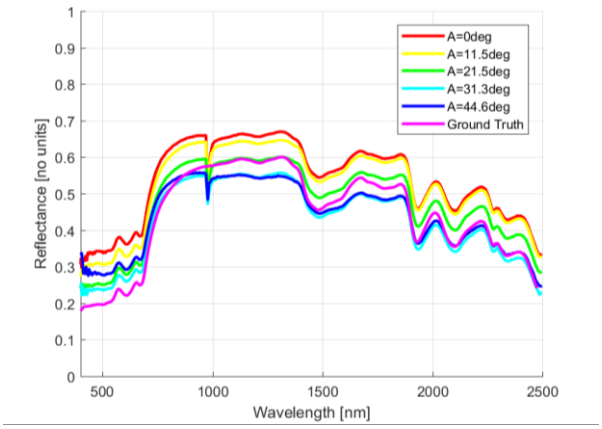

Figure 5.57 – Lambertian Corrected Reflectance for Planes

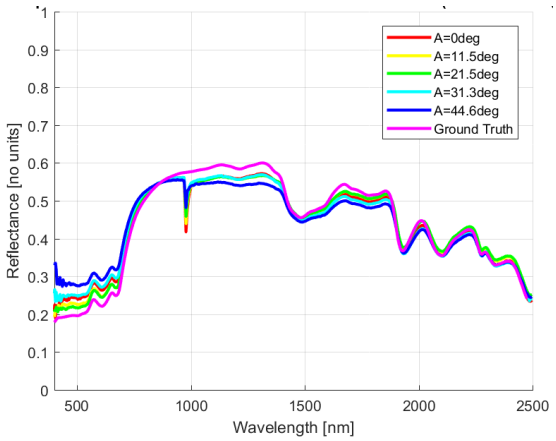

Figure 5.58 – Specular Corrected Reflectance (γ Thresh = 180°)

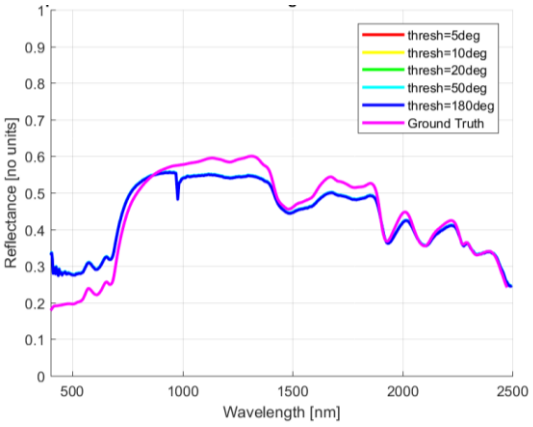

Figure 5.59 – Specular Correction for Differing γ Thresholds

Specular corrections show all planes A0 through A45 closely resembling the ground truth spectra (Fig.5.58). Interestingly the shape of the planes spectra is much more like that of the ground truth than the cylinder spectra. Like the cylinder experiments, changing the threshold of the correction has little impact on the correction (Fig.5.59). Lambertian correction shows significant improvement over the uncorrected spectra. And specular correction (Fig.5.60) shows significant improvement over Lambertian corrections.

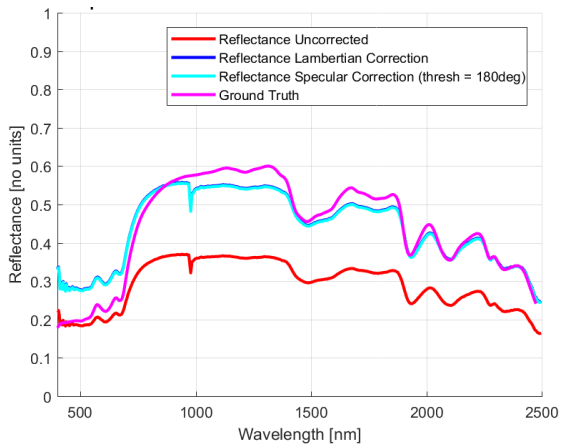

Figure 5.60 – Specular and Lambertian Corrections for Planes

6 DISCUSSION

6.1 GEOMETRIC CALIBRATION EFFICACY

6.1.1 Non-linear effect removal

Considering the plots of 1D polynomials(Fig.39) and fitted 2D Chebyshev polynomials (Fig.42) in section 5.2.1.2 and 5.2.1.3. Residual error that changes as a function of u-direction indicates that this error related to the camera motion. On the other hand, residual error that changes as a function of v-direction indicates that the error is related to intrinsic properties of the camera itself.

Both the Chebyshev polynomials that model error in the u and v directions show noticeable changes as a function of both u and v directions (Fig.42). This indicates that the non-linear effects are due to both inconsistencies in the linear motion of the camera along the rail, and intrinsic sensor properties. Both the u and v-residuals show an approximately parabolic shape with respect to the v direction, more so for the u-residuals, while the v-residuals include a small inflection in curvature. Apart from this inflection in curvature, these effects suggest the presence of barrel/pincushion distortion such as that found in a typical perspective camera. Both u and v-residuals shows much less variation in shape as a function of the u-direction. Suggesting that the motion of the camera is relatively consistent and does not undergo any large, nor sudden translations or rotations. However, while the error as a function of camera motion can be considered relatively small compared to error due to camera intrinsic it is certainly non-negligible.

While the 1D polynomial correction (Fig.39) was less effective than 2D correction. Only error in the u-direction was accounted for and neglected to include v-error in terms of a barrel/pincushion distortion effect. Due the error as a function of camera motion is relatively small, it is expected that fitting of 1D polynomials to both u and v-residuals as a function of only the v-direction would yield an error correction, and accurate camera model estimation comparable to that of the 2D Chebyshev polynomials in section 5.2.1.3.

6.1.2 Calibration Stability

As outlined in section 4.4.2 high calibration stability is a desirable property as it means that the camera does not have to be re calibrated after each image acquisition, reducing downtime, and increasing throughput. There are also implications for both correction of Lambertian effects, and misregistration of pixels to the world coordinate system.

The position of the six tungsten halogen lights is fixed to the position of the camera. Meaning error in the camera pose with respect to the previous calibration also results in error of actual illumination geometry with respect to the modelled geometry. Table 4. shows a translation standard deviation of 8.4mm in the z-

direction, and rotational standard deviation of 0.2° about the x-axis. Considering the camera is approximately 1.3m above the ground, this corresponds to a Lambertian error of 1.3% due to translation stability, and 0.0006% due to rotational stability.

$$1.3\% = 1 - \frac{(1.3 + 0.0084)^2}{1.3^2}$$

$$0.0006\% = 1 - \cos(0.2°)$$

Concerning misregistration effects, the x and y translational standard deviation are 2.2mm and 1.6mm respectively. This corresponds to a standard deviation of greater than 0.5 pixels for both u and v pixel directions. At a worst case standard deviation of 2.2mm which is approximately 1.16 pixels (1.16=2.2/1.9), meaning that only 33.3% of image acquisitions will be registered to within ±0.5 pixels (assuming gaussian distribution) which is exceptionally poor. This means that this setup will need to be recalibrated for each image acquisition. However, the AisaFENIX hyperspectral camera is designed for aerial applications.

6.1.3 Wavelength dependency of error

Deviations within the VIS/NIR and SWIR regions individually are in range of the smile and keystone specifications in the datasheet (Specim 2013). However, the differences in mean pixel coordinate between VIS/NIR and SWIR regions for the u-direction indicates that a difference in u-value of up to 0.7 pixels is not uncommon. This implication here is that there is no guarantee that VIS/NIR and SWIR pixels correspond to the same point in the scene. It may be possible to correct for this effect by determining a unique calibration for the VIS/NIR and SWIR regions.

6.2 Suitability of Point Clouds and Meshes for Illumination Correction

6.2.1 Evaluation of angle bias results

Planes imaged with the Microsoft Kinect performed best with regards to angle bias for A10 and A20 and worse as the angle deviated either side of these. During image acquisition the Microsoft Kinect had an altitude angle in the range of approximately 10 to 30 degrees with respect to the table. Planes with an inclination angle within this range have normal more aligned with the Kinect optical axis. This agrees with the findings of DiFilippo and Jouaneh (2015) who observed that the Microsoft Kinect plane angle estimation becomes increasingly poor with plane normal deviation from the Microsoft Kinect's optical axis. However, DiFilippo and Jouaneh (2015) also observed that the Microsoft Kinect underestimated these angles further from alignment with its optical axis.

To completely agree with DiFilippo and Jouaneh (2015), the angle deviation should be negative for A45, and positive for A0, however, this is not the case. While DiFillipo performed experiments on the raw Microsoft Kinect data, these experiments make use of ReconstructMe software which uses an unpublished algorithm.

The SFM technology also shows worse angle bias as plane angle deviates towards A0 and A45.

In terms of texture, the Microsoft Kinect is known to perform well even on texturally homogenous surfaces as it only requires that the texture being imaged to reflect the IR structured light pattern. While the performance is relatively constant over T2, T3 and T4, its poor performance on T5 can be due the increased size of lower reflectance black tiles, meaning only a fraction of the IR pattern is reflected. On the other hand, the SFM performance varies strongly with texture, increasing as the textural frequency decreases. Determining correspondences between points in two images for dense reconstruction requires finding the two candidate correspondences with the most similar pixel neighbourhoods. In the case of high textural frequency, the correspondences will become more ambiguous as many points will share similar neighbour hoods. However, in the case of low textural frequency there will be less ambiguity, which should lead to a better reconstruction.

6.2.2 Evaluation of plane mesh angular deviation results

Similarly, to the plane bias results, the Microsoft Kinect shows similar plane deviation distributions for T2,T3, and T4, but much greater mesh undulation for T5. This can be explained by the increased size of low reflectance black tiles, meaning less of the Kinect IR projected pattern was visible to the Microsoft Kinect. It is expected that this should not be a problem in phenotyping scenarios as plants will have consistently better reflectance in the NIR wavebands.

On the other hand, SFM shows best performance at texture T4 which decreases as both the textural frequency increases to T3 or decreases to T5. This effect is related to solving the correspondence problem for dense reconstruction where the textural pattern is constant over a large range of pixels. When the textural frequency is too high, there is ambiguity with block matching as there are no distinct features to unambiguously determine a correspondence. However, in the case that the spatial frequency is too low the problem can also be ambiguous as the sparse areas may have too little information to unambiguously determine a correspondence. Incorrect correspondences will result in poorer quality point cloud, which in turn will result in a poorer quality mesh.

6.2.3 Evaluation of cylinder mesh angular deviation results

Both SFM and Kinect technologies showed increasingly worse mesh angle deviation with decrease in cylinder radius. There are many factors that will contribute to the fidelity of the final mesh, these will include sensor resolution, amount of surface area visible to the sensor, averaging of points in the meshing stage, sensor depth error. However, one likely explanation is the exacerbation of sensor depth error as cylinder radius decreases. The two-dimensional analogue of fitting a triangular plane to a cylindrical surface is fitting a line to a circle. Consider one point of the line has been correctly found p_A, (Fig.6.1) the sensor then tries to find the second point p_B. However, some small lateral error dy in the sensor measurement results in the point p_e being sampled instead of point p_B (Fig.6.1). The sensor records the depth coordinate of p_e with the lateral coordinate of p_B therby introducing error. Any small lateral error dy in sensor measurement will result in a small depth error dz. As can be seen in Fig.6.1 and Fig.6.2, if the sensor resolution is constant, and there is some small error dy, then the depth error increases as the cylinder radius decreases.

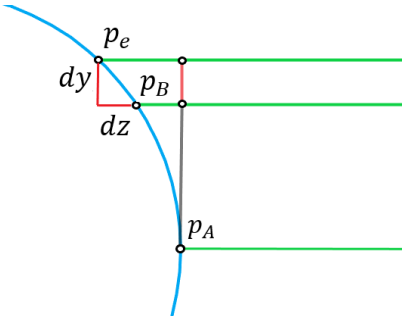

Figure 6.1 – Exacerbation of Depth Error (large radius)

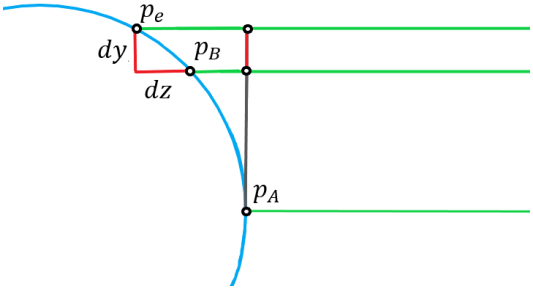

Figure 6.2 – Exacerbation of Depth Error (small radius)

6.3 Efficacy of Illumination Correction

6.3.1 Evaluation of cylindrical illumination correction

One possible explanation for the overcorrection of illumination at the cylinder edges is the assumption of negligible ambient illumination (Fig.5.43). Consider the illumination model:

$$\rho = \frac{1}{\beta L_0}\left[\frac{DN - c_0}{c_1} - L_{ma} - \alpha\right]$$

Under the assumption of no specular reflectance (which is reasonable at the cylinder edges) and no ambient illumination, the α and L_{ma} terms are set to zero, the true Lambertian reflectance can be expressed as:

$$\rho_{actual} = \frac{1}{\beta L_0}\left[\frac{DN - c_0}{c_1}\right] = \frac{1}{\beta L_0} L$$

where L is the radiance. The presence of unaccounted for ambient illumination can be expressed as:

$$\rho = \frac{1}{\beta L_0}[L + L_{ma}] = \rho_{actual} + \frac{L_{ma}}{\beta L_0}$$

As the predicted Lambertian factor β decreases the deviation from actual lambertian reflectance ρ_{actual} increases hyperbolically. This is exactly what happens at the edges of the cylinders, where the lambertian factor β decreases towards the cylinders edges. This could be mitigated by imaging within a dark room or capturing an ambient frame as well as a dark frame.

6.3.2 Evaluation of planes illumination correction

Surprisingly the Lambertian correction of planes is much worse than the cylinders, with corrected reflectance varying by about 0.1 between the planes (Fig.5.57). However, the planes least susceptible to specular effects (Fig.5.57) are A20, A30 and A45 which show corrected reflectance closer to the ground truth, while those more susceptible (having gamma values in the range of 20 degrees or less) have Lambertian corrected reflectance consistently greater than the ground truth.

Interestingly both the corrected spectra of the planes (Fig.5.57) and cylinders (Fig.5.47) had spectral shapes the clearly differed from each other and from the ground truth spectra. This can be explained by the different materials behind the paper patterns of the cylinders and planes. The planes were made from wooden blocks, and the cylinders were metal.

7 CONCLUSION AND OUTLOOK

In conclusion, this thesis has explored camera calibration, point cloud acquisition, alignment, and illumination correction methodologies using low resolution point clouds. While no hyperspectral images of plants were corrected, this study was still able to answer the research questions outlined section 1.3.

With regards to camera calibration, this research has confirmed that very reasonable camera models and calibrations can be obtained using the linear pushbroom camera model proposed by Gupta and Hartley (1997). The camera movement along the linear rail causes significant non-linear geometric image distortions, in addition to the already present barrel/pincushion distortion. Meaning camera calibrations should model non-linear error as a function of the u as well as v-direction. Camera calibration stability is important to avoid both misregistration, and error in Lambertian reflectance corrections. The AisaFENIX is not suitable for spatially precise lab based hyperspectral imaging due to poor calibration stability, large wavelength dependent error due to VIS/NIR and SWIR detector misalignment, and large reprojection error (1.4pixels) after image undistortion.

This research has evaluated the performance of the Microsoft Kinect and SFM technologies with respect to textural frequency and cylindrical radius of curvature and plane angle. Both technologies show bias in plane angle estimation which imposes a limit to their accuracy in illumination correction of hyperspectral images. The Microsoft Kinect performed very poorly with small radius, indicating that it may not be useful for use on objects with very small radii of curvature. The experiments conducted here showed that SFM produced highly noisy data with many misregistration artefacts. SFM showed a significant decrease in performance when the textural frequency of the object was close to and beyond the camera resolution. While this research has demonstrated successful illumination correction of hyperspectral images based on meshes obtained by these technologies, both technologies have significant performance and processing drawbacks that indicate that they shouldn't be used for illumination correction in any practical applications.

8 BIBLIOGRAPHY

Aasen, Helge, Andreas Burkart, Andreas Bolten, and Georg Bareth. 2015. "Generating 3D hyperspectral information with lightweight UAV snapshot cameras for vegetation monitoring: From camera calibration to quality assurance." *ISPRS Journal of Photogrammetry and Remote Sensing* 108:245-259. doi: https://doi.org/10.1016/j.isprsjprs.2015.08.002.

Alnowami, M., B. Alnwaimi, F. Tahavori, M. Copland, and K. Wells. 2012. "A quantitative assessment of using the Kinect for Xbox 360 for respiratory surface motion tracking." SPIE Medical Imaging.

Araus, José Luis, Gustavo A Slafer, Conxita Royo, and M Dolores Serret. 2008. "Breeding for yield potential and stress adaptation in cereals." *Critical Reviews in Plant Science* 27 (6):377-412.

Arvidsson, Samuel, Paulino Pérez-Rodríguez, and Bernd Mueller-Roeber. 2011. "A growth phenotyping pipeline for Arabidopsis thaliana integrating image analysis and rosette area modeling for robust quantification of genotype effects." *New Phytologist* 191 (3):895-907.

Azzari, George, Michael L Goulden, and Radu B Rusu. 2013. "Rapid characterization of vegetation structure with a microsoft kinect sensor." *Sensors* 13 (2):2384-2398.

Behmann, Jan, Anne-Katrin Mahlein, Stefan Paulus, Jan Dupuis, Heiner Kuhlmann, Erich-Christian Oerke, and Lutz Plümer. 2016. "Generation and application of hyperspectral 3D plant models: methods and challenges." *Machine Vision and Applications* 27 (5):611-624.

Behmann, Jan, Anne-Katrin Mahlein, Stefan Paulus, Heiner Kuhlmann, Erich-Christian Oerke, and Lutz Plümer. 2015. "Calibration of hyperspectral close-range pushbroom cameras for plant phenotyping." *ISPRS Journal of Photogrammetry and Remote Sensing* 106:172-182.

Behmann, Jan, Jörg Steinrücken, and Lutz Plümer. 2014. "Detection of early plant stress responses in hyperspectral images." *ISPRS Journal of Photogrammetry and Remote Sensing* 93:98-111. doi: https://doi.org/10.1016/j.isprsjprs.2014.03.016.

Bergström, Per, and Ove Edlund. 2014. "Robust registration of point sets using iteratively reweighted least squares." *Computational Optimization and Applications* 58 (3):543-561.

Berk, A., L. S. Bernstein, G. P. Anderson, P. K. Acharya, D. C. Robertson, J. H. Chetwynd, and S. M. Adler-Golden. 1998. "MODTRAN Cloud and Multiple Scattering Upgrades with Application to AVIRIS." *Remote Sensing of Environment* 65 (3):367-375. doi: http://dx.doi.org/10.1016/S0034-4257(98)00045-5.

Bernardini, F., J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. 1999. "The ball-pivoting algorithm for surface reconstruction." *IEEE Transactions on Visualization and Computer Graphics* 5 (4):349-359. doi: 10.1109/2945.817351.

Bioucas-Dias, J. M., A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. 2012. "Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5 (2):354-379. doi: 10.1109/JSTARS.2012.2194696.

Blum, Abraham. 2011. "Drought resistance and its improvement." In *Plant Breeding for Water-Limited Environments*, 53-152. Springer.

Busemeyer, Lucas, Daniel Mentrup, Kim Möller, Erik Wunder, Katharina Alheit, Volker Hahn, Hans Peter Maurer, Jochen C Reif, Tobias Würschum, and Joachim Müller. 2013. "Breedvision—A multi-sensor platform for non-destructive field-based phenotyping in plant breeding." *Sensors* 13 (3):2830-2847.

Cabrera-Bosquet, Llorenç, José Crossa, Jarislav von Zitzewitz, Marıa Dolors Serret, and José Luis Araus. 2012. "High-throughput Phenotyping and Genomic Selection: The Frontiers of Crop Breeding ConvergeF." *Journal of Integrative Plant Biology* 54 (5):312-320.

Campos, H, M Cooper, JE Habben, GO Edmeades, and JR Schussler. 2004. "Improving drought tolerance in maize: a view from industry." *Field crops research* 90 (1):19-34.

Chéné, Yann, David Rousseau, Philippe Lucidarme, Jessica Bertheloot, Valérie Caffier, Philippe Morel, Étienne Belin, and François Chapeau-Blondeau. 2012. "On the use of depth camera for 3D phenotyping of entire plants." *Computers and Electronics in Agriculture* 82:122-127.

Chin, Norman, and Steven Feiner. 1989. "Near real-time shadow generation using BSP trees." *ACM SIGGRAPH Computer Graphics* 23 (3):99-106.

Choi, Sunglok, Taemin Kim, and Wonpil Yu. 1997. "Performance evaluation of RANSAC family." *Journal of Computer Vision* 24 (3):271-300.

Chow, JC, KD Ang, DD Lichti, and WF Teskey. 2012. "Performance Analysis of a Low-Cost Triangulation-Based 3d Camera: Microsoft Kinect System." *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*:175-180.

Cobb, Joshua N., Genevieve DeClerck, Anthony Greenberg, Randy Clark, and Susan McCouch. 2013. "Next-generation phenotyping: requirements and strategies for enhancing our understanding of genotype–phenotype relationships and its relevance to crop improvement." *Theoretical and Applied Genetics* 126 (4):867-887. doi: 10.1007/s00122-013-2066-0.

Conel, James E, Robert O Green, Gregg Vane, Carol J Bruegge, Ronald E Alley, and Brian J Curtiss. 1987. "AIS-2 radiometry and a comparison of methods for the recovery of ground reflectance."

Derpanis, Konstantinos G. 2004. "The harris corner detector." *York University*.

Derpanis, Konstantinos G. 2010. "Overview of the RANSAC Algorithm." *Image Rochester NY* 4 (1):2-3.

DiFilippo, Nicholas M, and Musa K Jouaneh. 2015. "Characterization of different Microsoft Kinect sensor models." *IEEE Sensors Journal* 15 (8):4554-4564.

Estrada, Félix, Alejandro Escobar, Sebastián Romero-Bravo, Jaime González-Talice, Carlos Poblete-Echeverría, Peter D. S. Caligari, and Gustavo A. Lobos. 2015. "Fluorescence phenotyping in blueberry breeding for genotype selection under drought conditions, with or without heat stress." *Scientia Horticulturae* 181:147-161. doi: https://doi.org/10.1016/j.scienta.2014.11.004.

Fahlgren, Noah, Malia A Gehan, and Ivan Baxter. 2015. "Lights, camera, action: high-throughput plant phenotyping is ready for a close-up." *Current opinion in plant biology* 24:93-99.

Fischler, Martin A, and Robert C Bolles. 1987. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." In *Readings in computer vision*, 726-740. Elsevier.

Garland, Michael, Andrew Willmott, and Paul S Heckbert. 2001. "Hierarchical face clustering on polygonal surfaces." Proceedings of the 2001 symposium on Interactive 3D graphics.

Ge, Yufeng, Geng Bai, Vincent Stoerger, and James C. Schnable. 2016. "Temporal dynamics of maize plant growth, water use, and leaf water content using automated high throughput RGB and hyperspectral imaging." *Computers and Electronics in Agriculture* 127:625-632. doi: https://doi.org/10.1016/j.compag.2016.07.028.

Gerald, N, Ursula K Frei, and Thomas Lübberstedt. 2013. "Accelerating plant breeding." *Trends in plant science* 18 (12):667-672.

Godfray, H Charles J, John R Beddington, Ian R Crute, Lawrence Haddad, David Lawrence, James F Muir, Jules Pretty, Sherman Robinson, Sandy M Thomas, and Camilla Toulmin. 2010. "Food security: the challenge of feeding 9 billion people." *science* 327 (5967):812-818.

Goetz, Alexander FH. 2009. "Three decades of hyperspectral remote sensing of the Earth: A personal view." *Remote Sensing of Environment* 113:S5-S16.

Golzarian, Mahmood R, Ross A Frick, Karthika Rajendran, Bettina Berger, Stuart Roy, Mark Tester, and Desmond S Lun. 2011. "Accurate inference of shoot biomass from high-throughput images of cereal plants." *Plant Methods* 7 (1):2.

Green, AA, and MD Craig. 1985. "Analysis of aircraft spectrometer data with logarithmic residuals."

Gupta, Rajiv, and Richard I Hartley. 1997. "Linear pushbroom cameras." *IEEE Transactions on pattern analysis and machine intelligence* 19 (9):963-975.

Heindl, Bauer, Harald BAUER, Martin ANKERL, and Andreas PICHLER. 2015. "ReconstructMe SDK: a C API for Real-time 3D Scanning." 6th International Conference and Exhibition on 3D Body Scanning Technologies.

Hoppe, Hugues, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. 1992. *Surface reconstruction from unorganized points*. Vol. 26: ACM.

Hosoi, Fumiki, Kazushige Nakabayashi, and Kenji Omasa. 2011. "3-D modeling of tomato canopies using a high-resolution portable scanning lidar for extracting structural information." *Sensors* 11 (2):2166-2174.

Houle, David, Diddahally R Govindaraju, and Stig Omholt. 2010. "Phenomics: the next challenge." *nATuRe RevIeWS| Genetics* 11:855.

Ibraheem, Issa. 2015. "Maximum Likelihood and Spectral Angle Mapper and K-means algorithms used to detection of Melanoma." *American Journal of Biomedical and Life Sciences* 3 (2-3):8-15.

Jacquemoud, S., and F. Baret. 1990. "PROSPECT: A model of leaf optical properties spectra." *Remote Sensing of Environment* 34 (2):75-91. doi: http://dx.doi.org/10.1016/0034-4257(90)90100-Z.

Jacquemoud, Stéphane, Wout Verhoef, Frédéric Baret, Cédric Bacour, Pablo J. Zarco-Tejada, Gregory P. Asner, Christophe François, and Susan L. Ustin. 2009. "PROSPECT+SAIL models: A review of use for vegetation characterization." *Remote Sensing of Environment* 113:S56-S66. doi: http://dx.doi.org/10.1016/j.rse.2008.01.026.

Jain, Anil, and Arun Ross. 2002. "Fingerprint mosaicking." Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on.

Kazhdan, Michael. n.d. "Screened Poisson Surface Reconstruction (Version 5)."

Kazhdan, Michael, and Hugues Hoppe. 2013. "Screened poisson surface reconstruction." *ACM Transactions on Graphics (ToG)* 32 (3):29.

Kazmi, Wajahat, Sergi Foix, Guillem Alenyà, and Hans Jørgen Andersen. 2014. "Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison." *ISPRS journal of photogrammetry and remote sensing* 88:128-146.

Keightley, Keir E, and Gerald W Bawden. 2010. "3D volumetric modeling of grapevine biomass using Tripod LiDAR." *Computers and Electronics in Agriculture* 74 (2):305-312.

Khoshelham, Kourosh, and Sander Oude Elberink. 2012. "Accuracy and resolution of kinect depth data for indoor mapping applications." *Sensors* 12 (2):1437-1454.

Kipp, Sebastian, Bodo Mistele, Peter Baresel, and Urs Schmidhalter. 2014. "High-throughput phenotyping early plant vigour of winter wheat." *European Journal of Agronomy* 52, Part B:271-278. doi: https://doi.org/10.1016/j.eja.2013.08.009.

Kjaer, Katrine, and Carl-Otto Ottosen. 2015. "3D Laser Triangulation for Plant Phenotyping in Challenging Environments." *Sensors* 15 (6):13533.

Klose, Ralph, Jaime Penlington, and Arno Ruckelshausen. 2009. "Usability study of 3D time-of-flight cameras for automatic plant phenotyping." *Bornimer Agrartechnische Berichte* 69 (93-105):12.

Knipling, Edward B. 1970. "Physical and physiological basis for the reflectance of visible and near-infrared radiation from vegetation." *Remote sensing of environment* 1 (3):155-159.

Kruse, Fred A. 1988. "Use of airborne imaging spectrometer data to map minerals associated with hydrothermally altered rocks in the northern grapevine mountains, Nevada, and California." *Remote Sensing of Environment* 24 (1):31-51.

Kuska, Matheus, Mirwaes Wahabzada, Marlene Leucker, Heinz-Wilhelm Dehne, Kristian Kersting, Erich-Christian Oerke, Ulrike Steiner, and Anne-Katrin Mahlein. 2015. "Hyperspectral phenotyping on the microscopic scale: towards automated characterization of plant-pathogen interactions." *Plant Methods* 11 (1):28.

Lawrence, Kurt C, Bosoon Park, William R Windham, and Chengye Mao. 2003. "Calibration of a pushbroom hyperspectral imaging system for agricultural inspection." *Transactions of the ASAE* 46 (2):513.

Li, Lei, Qin Zhang, and Danfeng Huang. 2014. "A Review of Imaging Techniques for Plant Phenotyping." *Sensors* 14 (11):20078.

Liebisch, Frank, Norbert Kirchgessner, David Schneider, Achim Walter, and Andreas Hund. 2015. "Remote, aerial phenotyping of maize traits with a mobile multi-sensor approach." *Plant Methods* 11 (1):9. doi: 10.1186/s13007-015-0048-8.

Mahlein, Anne-Katrin. 2016. "Plant disease detection by imaging sensors–parallels and specific demands for precision agriculture and plant phenotyping." *Plant Disease* 100 (2):241-251.

Mankoff, Kenneth David, and Tess Alethea Russo. 2013. "The Kinect: A low-cost, high-resolution, short-range 3D camera." *Earth Surface Processes and Landforms* 38 (9):926-936.

Milella, Annalisa, and Roland Siegwart. 2006. "Stereo-based ego-motion estimation using pixel tracking and iterative closest point." Computer Vision Systems, 2006 ICVS'06. IEEE International Conference on.

Minervini, Massimo, Hanno Scharr, and Sotirios A Tsaftaris. 2015. "Image analysis: the new bottleneck in plant phenotyping [applications corner]." *IEEE Signal Processing Magazine* 32 (4):126-131.

Mishra, Puneet, Mohd Shahrimie, Stien Mertens, Stijn Dhondt, Nathalie Wuyts, and Paul Scheunders. 2016. "Close range hyperspectral imaging for plant phenotyping."

Mizuno, Shinji, Keiichi Noda, Nobuo Ezaki, Hotaka Takizawa, and Shinji Yamamoto. 2007. "Detection of wilt by analyzing color and stereo vision data of plant." International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications.

Möller, Tomas, and Ben Trumbore. 2005. "Fast, minimum storage ray/triangle intersection." ACM SIGGRAPH 2005 Courses.

Moltó, Enrique, José Blasco, and Juan Gómez-Sanchís. 2010. "CHAPTER 10 - Analysis of Hyperspectral Images of Citrus Fruits A2 - Sun, Da-Wen." In *Hyperspectral Imaging for Food Quality Analysis and Control*, 321-348. San Diego: Academic Press.

Nicodemus, Fred E. 1970. "Reflectance nomenclature and directional reflectance and emissivity." *Applied Optics* 9 (6):1474-1475.

Nicodemus, Fred Edwin. 1977. *Geometrical considerations and nomenclature for reflectance*. Vol. 160: US Department of Commerce, National Bureau of Standards.

Panguluri, Siva Kumar, and Are Ashok Kumar. 2016. *Phenotyping for Plant Breeding*: Springer.

Paproki, Anthony, Xavier Sirault, Scott Berry, Robert Furbank, and Jurgen Fripp. 2012. "A novel mesh processing based technique for 3D plant analysis." *BMC plant biology* 12 (1):63.

Paulus, Stefan, Jan Behmann, Anne-Katrin Mahlein, Lutz Plümer, and Heiner Kuhlmann. 2014. "Low-cost 3D systems: suitable tools for plant phenotyping." *Sensors* 14 (2):3001-3018.

Paulus, Stefan, Henrik Schumann, Heiner Kuhlmann, and Jens Léon. 2014. "High-precision laser scanning system for capturing 3D plant architecture and analysing growth of cereal plants." *Biosystems Engineering* 121:1-11.

Pérez-de-Castro, A. M., S. Vilanova, J. Cañizares, L. Pascual, J. M. Blanca, M. J. Díez, J. Prohens, and B. Picó. 2012. "Application of Genomic Tools in Plant Breeding." *Current Genomics* 13 (3):179-195. doi: 10.2174/138920212800543084.

Ploschuk, E. L., L. A. Bado, M. Salinas, D. F. Wassner, L. B. Windauer, and P. Insausti. 2014. "Photosynthesis and fluorescence responses of Jatropha curcas to chilling and freezing stress during early vegetative stages." *Environmental and Experimental Botany* 102:18-26. doi: https://doi.org/10.1016/j.envexpbot.2014.02.005.

Pullanagari, RR, Gábor Kereszturi, and IJ Yule. 2016. "Mapping of macro and micro nutrients of mixed pastures using airborne AisaFENIX hyperspectral imagery." *ISPRS Journal of Photogrammetry and Remote Sensing* 117:1-10.

Ray, Deepak K, Nathaniel D Mueller, Paul C West, and Jonathan A Foley. 2013. "Yield trends are insufficient to double global crop production by 2050." *PloS one* 8 (6):e66428.

Ray, Deepak K, Navin Ramankutty, Nathaniel D Mueller, Paul C West, and Jonathan A Foley. 2012. "Recent patterns of crop yield growth and stagnation." *Nature communications* 3:1293.

Roberts, DA, Y Yamaguchi, and RJP Lyon. 1986. "Comparison of various techniques for calibration of AIS data." *NASA STI/Recon Technical Report N* 87.

Roscher, Ribana, Jan Behmann, Anne-Katrin Mahlein, Jan Dupuis, Heiner Kuhlmann, and Lutz Plümer. 2016. "DETECTION OF DISEASE SYMPTOMS ON HYPERSPECTRAL 3D PLANT MODELS." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 3 (7).

Sabbadin, Manuele. 2016. "Interaction and rendering with harvested 3D data." PhD Thesis, Department of Computer Science, University of Pisa, Pisa, Italy.

Samet, Hanan. 1989. "Implementing ray tracing with octrees and neighbor finding." *Computers & Graphics* 13 (4):445-460.

Shah, D., L. Tang, Jingyao Gai, and R. Putta-Venkata. 2016. "Development of a Mobile Robotic Phenotyping System for Growth Chamber-based Studies of Genotype x Environment Interactions." *IFAC-PapersOnLine* 49 (16):248-253. doi: https://doi.org/10.1016/j.ifacol.2016.10.046.

Shevtsov, Maxim, Alexei Soupikov, and Alexander Kapustin. 2007. "Highly Parallel Fast KD-tree Construction for Interactive Ray Tracing of Dynamic Scenes." Computer Graphics Forum.

Specim. 2013. "AisaFENIX VNIR/SWIR Hyperspectral Imager User Manual Ver. 1.7."

Suomalainen, Juha, Niels Anders, Shahzad Iqbal, Gerbert Roerink, Jappe Franke, Philip Wenting, Dirk Hünniger, Harm Bartholomeus, Rolf Becker, and Lammert Kooistra. 2014. "A lightweight hyperspectral mapping system and photogrammetric processing chain for unmanned aerial vehicles." *Remote Sensing* 6 (11):11013-11030.

Sytar, Oksana, Marian Brestic, Marek Zivcak, Katarina Olsovska, Marek Kovar, Hongbo Shao, and Xiaolan He. 2017. "Applying hyperspectral imaging to explore natural plant diversity towards improving salt stress tolerance." *Science of The Total Environment* 578:90-99. doi: https://doi.org/10.1016/j.scitotenv.2016.08.014.

Tardieu, François, and Roberto Tuberosa. 2010. "Dissection and modelling of abiotic stress tolerance in plants." *Current Opinion in Plant Biology* 13 (2):206-212. doi: https://doi.org/10.1016/j.pbi.2009.12.012.

Tewari, Geetika, Craig Gotsman, and Steven J Gortler. 2006. "Meshing genus-1 point clouds using discrete one-forms." *Computers & Graphics* 30 (6):917-926.

Thenkabail, Prasad S, and John G Lyon. 2016. *Hyperspectral remote sensing of vegetation*: CRC Press.

Thorp, K. R., M. A. Gore, P. Andrade-Sanchez, A. E. Carmo-Silva, S. M. Welch, J. W. White, and A. N. French. 2015. "Proximal hyperspectral sensing and data analysis approaches for field-based plant phenomics." *Computers and Electronics in Agriculture* 118:225-236. doi: https://doi.org/10.1016/j.compag.2015.09.005.

Tilman, David, Christian Balzer, Jason Hill, and Belinda L Befort. 2011. "Global food demand and the sustainable intensification of agriculture." *Proceedings of the National Academy of Sciences* 108 (50):20260-20264.

Torr, P. H. S., and A. Zisserman. 2000. "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry." *Computer Vision and Image Understanding* 78 (1):138-156. doi: https://doi.org/10.1006/cviu.1999.0832.

Vigneau, N., G. Rabatel, P. Roumet, and M. Ecarnot. 2010. "Calibration of a chemometrical model from field hyperspectral close-range images: Taking into account leaf inclination and multiple reflection effects." 2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 14-16 June 2010.

Virlet, Nicolas, Kasra Sabermanesh, Pouria Sadeghi-Tehran, and Malcolm J Hawkesford. 2017. "Field Scanalyzer: An automated robotic field phenotyping platform for detailed crop monitoring." *Functional Plant Biology* 44 (1):143-153.

Wang, Huanhuan, Yi Lin, Zeliang Wang, Yunjun Yao, Yuhu Zhang, and Ling Wu. 2017. "Validation of a low-cost 2D laser scanner in development of a more-affordable mobile terrestrial proximal sensing system for 3D plant structure phenotyping in indoor environment." *Computers and Electronics in Agriculture* 140:180-189. doi: https://doi.org/10.1016/j.compag.2017.06.002.

White, Jeffrey W, Pedro Andrade-Sanchez, Michael A Gore, Kevin F Bronson, Terry A Coffelt, Matthew M Conley, Kenneth A Feldmann, Andrew N French, John T Heun, and Douglas J Hunsaker. 2012. "Field-based phenomics for plant genetics research." *Field Crops Research* 133:101-112.

Wu, Changchang. 2011. "VisualSFM: A visual structure from motion system."

Wu, Changchang. 2013. "Towards linear-time incremental structure from motion." 3D Vision-3DV 2013, 2013 International Conference on.

Wylie, Chris, Gordon Romney, David Evans, and Alan Erdahl. 1967. "Half-tone perspective drawings by computer." Proceedings of the November 14-16, 1967, fall joint computer conference.

Yang, Wanneng, Lingfeng Duan, Guoxing Chen, Lizhong Xiong, and Qian Liu. 2013. "Plant phenomics and high-throughput phenotyping: accelerating rice functional genomics using

multidisciplinary technologies." *Current Opinion in Plant Biology* 16 (2):180-187. doi: http://dx.doi.org/10.1016/j.pbi.2013.03.005.

Yoon, Seung-Chul, and CN Thai. 2010. "Stereo spectral imaging system for plant health characterization." In *Technological Developments in Networking, Education and Automation*, 181-186. Springer.

Zhang, Chongyuan, Honghong Gao, Jianfeng Zhou, Asaph Cousins, Michael O. Pumphrey, and Sindhuja Sankaran. 2016. "3D Robotic System Development for High-throughput Crop Phenotyping." *IFAC-PapersOnLine* 49 (16):242-247. doi: https://doi.org/10.1016/j.ifacol.2016.10.045.