

# UNIVERSITY OF STRATHCLYDE

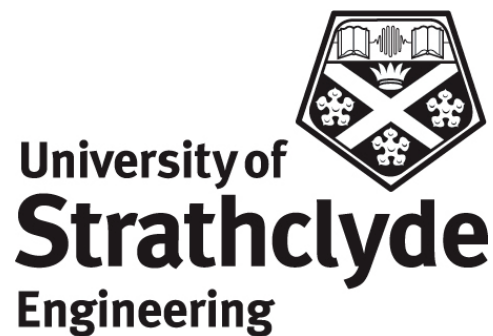
DEPARTMENT OF ELECTRONIC & ELECTRICAL ENGINEERING

## HYPERSPECTRAL IMAGING SYSTEM

By

SUVITHA KARTHICK

A thesis presented in fulfilment of the requirements for the degree of  
Master of Philosophy



2011

**This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.**

**The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.**

**Signed:**

**Date:**

## ACKNOWLEDGEMENTS

*"The mediocre teacher tells. The good teacher explains. The superior teacher demonstrates. The great teacher inspires". — William A. Ward*

I am thankful to my research supervisor **Prof. Stephen Marshall**, for all his support and encouragement during my research period. I am also very much grateful to him for having the faith in me and extending this research opportunity with Gilden Photonics Ltd.

*"The mind is not a vessel to be filled, but a fire to be kindled." — Plutarch*

I profusely thank **Dr. John R Gilchrist**, Gilden Photonics Ltd., for showing me a unique facet of work experience that combines both industrial and academic research with a business perspective. His tireless enthusiasm and support has always been a strong motivation factor behind many of my achievements so far.

*"The individual who says it is not possible should move out of the way of those doing it." — Tricia Cunningham*

I would be failing in my duty if I do not express my bountiful thanks to **Mr. Neil Cairns**, Gilden Photonics for the many hours he spent with me in guiding me to achieve the impossible.

I thank **Dr. Jinchang Ren**, University of Strathclyde and **Dr. Sivaram Shanmugam**, Glasgow Caledonian University for their inputs on my thesis.

I thank **Mr. Douglas Gourlay**, Gilden Photonics for all the brainstorming sessions and the constructive discussions and sometimes disagreements over design and development direction.

I thank my colleagues **Kevin, Liam, Lynn, Ross, Anna** and **Nic** at Gilden Photonics Ltd., for all their support and good fun and laughs that have made my working time so lively.

Lastly, I am thankful to **Mr. Bob Farrell** for allowing me to use some of his sample hyperspectral data in this thesis and his enthusiasm for the work.

## ABSTRACT

Hyperspectral imaging has emerged as a leading imaging technology in recent years with tremendous areas of research. This technology has become more commercial in the last decade and has a wide range of applications in the field of remote sensing, identification of tumours in biomedical, target detection in military, process monitoring, food processing and so on.

Hyperspectral cameras collect information about target objects over a wide range of narrow wavelength bands. They produce a set of images corresponding to each band of the whole electromagnetic spectrum. These set of images form the hyperspectral cube. Thus in a hyperspectral image, each pixel consists of a spectrum over a spectral region. Many features which do not appear within conventional imaging become visible in the hyperspectral domain providing more information about the target objects.

The enormous size of the multidimensional hyperspectral data is a challenging issue in the analysis and processing of information. As the application of hyperspectral imaging grows, there is a major need for a software system which could acquire and process the essential information in the hyperspectral data. The research focussed on developing a set of software tools with advanced visualisation techniques which could capture, render and process the hyperspectral data. The aim of the research was also to develop new techniques for faster hyperspectral data processing and image visualisation. Techniques for real time implementation of data acquisition and spectral data processing were developed to aid in overcoming the limitations of huge volume of data, processing time and complexity.

## LIST OF FIGURES

<i>Figure Number</i>	<i>Figure Title</i>	<i>Page Number</i>
1.1	<i>Evolution of HSI</i>	2
2.1	<i>Multispectral and Hyperspectral- Differences</i>	6
2.2	<i>Electromagnetic spectrum – Wavelength regions</i>	7
2.3	<i>Hypercube – Pictorial representation</i>	8
2.4.1	<i>Example of a greyscale image of a banknote</i>	8
2.4.2	<i>Example of an RGB image of a banknote</i>	9
2.4.3	<i>Example of an image of a banknote at 700nm</i>	9
2.4.4	<i>Example of an image of a banknote at 800nm</i>	9
2.5	<i>Hyperspectral image (from AISA Eagle sensor) – Spectral profile of water and vegetation.</i>	9
2.6	<i>Hyperspectral Imaging System – Schematic layout</i>	11
2.7	<i>Screenshot of Google Timeline representing the activity in HSI between 1960 - 2011</i>	12
2.8	<i>Word correction made in Declaration of Independence by Thomas Jefferson revealed by HSI. The word ‘subjects’ was over written with ‘citizens’. Image courtesy-Library of Congress</i>	15
3.1	<i>spectraSENS - Overview</i>	18
3.2	<i>Capturing spatial dimension over time</i>	18
3.3	<i>Capturing spectral dimension over time</i>	19
3.4	<i>Hyperspectral camera</i>	20
3.5	<i>Schematic representation of pushbroom imaging</i>	21
3.6	<i>Components of a HSI system</i>	21
3.7	<i>spectraSENS - Software architecture</i>	25
3.8	<i>spectraSENS – Interface to hardware</i>	25
3.9	<i>spectraSENS – Features</i>	26
4.1	<i>Motion control of scanner system</i>	28
4.2	<i>Linear stage</i>	28
4.3	<i>Rotational stage</i>	29
4.4	<i>Mirror scanner</i>	30

4.5	<i>Program flow</i>	31
4.6	<i>Screenshot of spectraSENS.ini file</i>	32
4.7	<i>Acquisition of one scan line</i>	34
5.1	<i>Working of the camera</i>	37
5.2	<i>Working of CCD camera</i>	38
5.3	<i>Frame transfer EMCCD sensor</i>	41
5.4	<i>Binning</i>	42
5.5	<i>Region of interest on CCD to be read out</i>	43
5.6	<i>Random track mode</i>	43
5.7	<i>Representation of Spectral calibration</i>	45
5.8	<i>Mercury lines at 436nm and 546nm in a frame</i>	46
5.9	<i>Screenshot of the calibration file</i>	47
5.10	<i>Representation of ENVI file format</i>	48
5.11	<i>Representation of the data storage formats</i>	50
5.12	<i>Screenshot of the ENVI header file</i>	51
6.1	<i>Functionalities of the Application layer</i>	52
6.2	<i>Stages in GUI development</i>	53
6.3	<i>spectraSENS – Screenshot of Graphical User Interface</i>	56
6.4	<i>Screenshot of the frame display</i>	58
6.5	<i>Screenshot of the greyscale palette file</i>	59
6.6	<i>Schematic diagram – Frame display</i>	60
6.7	<i>Palette control on the GUI</i>	60
6.8	<i>Screenshot of live acquisition display</i>	61
6.9	<i>Schematic diagram of the live acquisition display</i>	61
6.10	<i>Screenshot of waterfall display</i>	63
6.11	<i>Schematic diagram – waterfall display</i>	64
6.12	<i>Screenshot of spectral profile</i>	66
6.13	<i>Preview display</i>	67
6.14	<i>Screenshot of Hypercube tab control</i>	68
6.15	<i>Screenshot of a dark frame</i>	70
6.16	<i>Raw measurement of the spectral features of a leaf along with the</i>	70

	<i>measured white reference for the illumination source</i>	
6.17	<i>Screenshot of a white frame</i>	71
6.18	<i>Spectral profile of an apple – Before and after corrections</i>	72
6.19	<i>Screenshot of the corrections tab control</i>	73
7.1	<i>Mean reflectance spectra of fat and meat in the beef sample. The red plot represents fat and green plot represents meat</i>	77
7.2	<i>Mean reflectance spectra of bone, fat and meat in the lamb sample. The red plot represents fat, green represents meat and blue represents the bone region</i>	77
7.3	<i>Mean reflectance spectra of fat and meat in the pork sample. The red plot represents the fat and green plot represents meat</i>	78
7.4	<i>Mean reflectance spectra of fat in beef, lamb and pork samples.</i>	78
7.5	<i>Mean reflectance spectra of meat in beef, lamb and pork samples.</i>	79
7.6	<i>Beef sample : a)Image at 1200nm, b)PCA 1, c)PCA 2, d)PCA 3, e)PCA Colour composite</i>	80
7.7	<i>Lamb sample : a)Image at 1200nm, b)PCA 1, c)PCA 2, d)PCA 3, e)PCA Colour composite</i>	81
7.8	<i>Pork sample : a)Image at 1200nm, b)PCA 1, c)PCA 2, d)PCA 3, e)PCA Colour composite</i>	81
7.9	<i>Reflectance spectra of brown rust on wheat leaf (green plot) compared to that of non infected part of leaf (red plot)</i>	84
7.10	<i>Reflectance spectra of yellow rust on barley leaf (red plot) compared to that of non infected part of leaf (blue plot)</i>	84
7.11	<i>Reflectance spectra of Ryegrass seeds (red plot) and mustard seeds (green plot)</i>	85
7.12	<i>Wheat leaf infected with brown rust : a) Normal RGB image b) PCA colour composite image shows the locations of brown rust with higher contrast</i>	86
7.13	<i>Barley leaf infected with yellow rust: a) Normal RGB image b) PCA colour composite image shows the locations of yellow rust with higher contrast</i>	86
7.14	<i>Ryegrass and mustard seeds : a) Normal RGB image b) PCA colour composite image highlights the difference in colour between ryegrass and mustard seeds</i>	86
7.15	<i>RGB image of the forged cheque. The alterations could not be seen here.</i>	88

7.16	<i>False colour image of the cheque created with three different wavelength bands. This shows that the number '1' has been altered to number '4'</i>	89
7.17	<i>Reflectance spectra of the horizontal line (green plot) and vertical line (red plot) in the number '4'. The difference in spectra shows that different inks were used.</i>	89
7.18	<i>PCA colour composite image of forged cheque. This shows significant contrast differences between the two types of inks used in number '4'</i>	89
7.19	<i>Colour image of the Yuan banknote</i>	90
7.20	<i>Greyscale image of Yuan banknote at 800nm. Certain inks are transparent in this band.</i>	90
7.21	<i>False colour image of Yuan banknote created with images at three wavelength bands. This highlights different inks used in printing the document.</i>	90
7.22	<i>PCA colour composite image of Yuan banknote. This indicates significant contrast differences among the inks used in printing the document.</i>	91



## LIST OF TABLES

<b>Table Number</b>	<b>Table Title</b>	<b>Page Number</b>
2.1	<i>Differences - imaging modes</i>	5
2.2	<i>Regions of electromagnetic spectrum and wavelength range</i>	7
2.3	<i>Applications of HSI in different wavelength regions</i>	12
4.1	<i>Scanner API – Specifications</i>	30
4.2	<i>Motor – Specifications</i>	31
5.1	<i>Camera API functions</i>	39
5.2	<i>Camera specifications</i>	39
5.3	<i>ENVI header format</i>	49
5.4	<i>ENVI header – programmer’s perspective</i>	51
7.1	<i>GPL-N17E-30-XV60-USB HSI system specifications</i>	76
7.2	<i>Typical acquisition parameters</i>	76
7.3	<i>GPL-V8E-30-LRUSB HSI system specifications</i>	82
7.4	<i>Typical acquisition parameters</i>	82
7.5	<i>GPL-V8E-30-LRUSB HSI system specifications</i>	87
7.6	<i>Typical acquisition parameters</i>	87

## LIST OF ABBREVIATIONS AND SYMBOLS

### *Abbreviations*

$\lambda$

$\lambda_p$

$\mu\text{step}$

$\mu_s$

$\mu S_{rev}$

$\theta_s$

ADC

API

ARS

ASCII

AVIRIS

$B(i)$

$B(i, P_s(i))$

BIL

BIP

BMP

BSQ

$B_w$

CCD

CCFRA

C-H

$d$

EMCCD

ERTS

ETF

$f$

FPA

$fps$

### *Terms*

*Wavelength*

*Wavelength corresponding to pixel row number  $p$*

*Microstep*

*Microstep resolution*

*Microsteps per rev*

*Step angle*

*Analog to Digital Converter*

*Application Programming Interface*

*Agricultural Research Service*

*American Standard Code for Information Interchange*

*Airborne Visible/ Infrared Imaging Spectrometer*

*Blue value calculated for pixel  $P(i)$*

*Third value found in row number  $P_s(i)$  in the palette file*

*Band Interleaved by Line*

*Band Interleaved by Pixel*

*Bitmap*

*Band sequential*

*Binning factor*

*Charge Coupled Device*

*Campden & Chorleywood Food Research Association*

*Carbon-Hydrogen bond*

*Pitch*

*Electron Multiplying Charge Coupled Device*

*Earth Resources Technology Satellite*

*Electronically Tunable Filter*

*Focal length*

*Focal Plane Array*

*Frames per second*

$G(i)$	<i>Green value calculated for pixel <math>P(i)</math></i>
$G(i, P_s(i))$	<i>Second value found in row number <math>P_s(i)</math> in the palette file</i>
GigE	<i>Gigabit Ethernet</i>
GPL	<i>Gilden Photonics Ltd.</i>
GUI	<i>Graphical User Interface</i>
HSI	<i>Hyperspectral Imaging</i>
$I$	<i>Intensity</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
$I_{100}(\lambda)$	<i>Pixel value in white reference image</i>
$I_D(\lambda)$	<i>Pixel value in dark reference image</i>
$I_R(\lambda)$	<i>Pixel value in measured image</i>
JPL	<i>Jet Propulsion Laboratory</i>
LCTF	<i>Liquid Crystal Tunable filter</i>
LUT	<i>Look-up-table</i>
LWIR	<i>Long Wave Infrared</i>
MSI	<i>Multispectral Imaging</i>
MWIR	<i>Mid Wave Infrared</i>
NASA	<i>National Aeronautics and Space Administration</i>
NIR	<i>Near Infrared</i>
$N_p$	<i>Number of pixels in one scan line</i>
O-H	<i>Oxygen-Hydrogen bond</i>
$p$	<i>Pixel row number</i>
PCA	<i>Principal component analysis</i>
$P_{max}$	<i>Maximum pixel value in an image buffer</i>
$P_{min}$	<i>Minimum pixel value in an image buffer</i>
$P_s(i)$	<i>Scaled 8 bit pixel value</i>
$P_s(\lambda_B)$	<i>Scaled pixel value corresponding to the wavelength band assigned to the blue colour</i>
$P_s(\lambda_G)$	<i>Scaled pixel value corresponding to the wavelength band assigned to the green colour</i>
$P_s(\lambda_R)$	<i>Scaled pixel value corresponding to the wavelength band assigned to the red colour</i>
QHSI	<i>Quantitative Hyperspectral Imaging</i>
$R$	<i>Frame rate</i>

$R(\lambda)$	<i>Spectral reflectance</i>
$R(i)$	<i>Red value calculated for pixel <math>P(i)</math></i>
$R(i, P_s(i))$	<i>First value found in row number <math>P_s(i)</math> in the palette file</i>
RGB	<i>Red-Green-Blue</i>
ROI	<i>Region of Interest</i>
SAM	<i>Spectral Angle Mapper</i>
$S_{rev}$	<i>Steps per revolution</i>
SWIR	<i>Short Wave Infrared</i>
$t$	<i>Time</i>
TIFF	<i>Tagged Image File Format</i>
USB	<i>Universal Serial Bus</i>
USDA	<i>United States Department of Agriculture</i>
UV	<i>Ultraviolet</i>
V	<i>Volts</i>
VIS	<i>Visible</i>
VNIR	<i>Visible Near Infrared</i>
$W_i$	<i>Slit width or sensor width</i>
$W_i$	<i>Swath</i>
$W_p$	<i>Pixel width</i>

# CONTENTS

	<i>Acknowledgements</i>	I
	<i>Abstract</i>	II
	<i>List of Figures</i>	III
	<i>List of Tables</i>	VII
	<i>List of Abbreviations and Symbols</i>	VIII
	<i>Contents</i>	XI
<b>1</b>	<b>INTRODUCTION</b>	1
1.1	<i>Motivation and scope of research</i>	1
1.2	<i>Structure of the thesis</i>	3
<b>2</b>	<b>INVISIBLE WORLD MADE VISIBLE!</b>	5
2.1	<i>Hyperspectral imaging</i>	5
2.2	<i>Hyperspectral image acquisition</i>	10
2.3	<i>Applications of hyperspectral imaging</i>	11
2.3.1	<i>Remote sensing</i>	12
2.3.2	<i>Food monitoring</i>	13
2.3.3	<i>Conservation of art and history</i>	14
2.3.4	<i>Medical sciences</i>	15
<b>3</b>	<b>spectraSENS</b>	17
3.1	<i>Hardware components of spectraSENS</i>	18
3.1.1	<i>Data acquisition techniques</i>	18
3.1.1.1	<i>Whiskbroom technique</i>	19
3.1.1.2	<i>Electronically tunable filter technique</i>	19
3.1.1.3	<i>Pushbroom technique</i>	20
3.1.1.4	<i>Advantages of pushbroom technique</i>	22
3.1.1.5	<i>Design of a pushbroom hyperspectral imaging system</i>	22
3.2	<i>Software components of spectraSENS</i>	23
3.2.1	<i>Software design</i>	23
3.2.2	<i>Software architecture</i>	24
3.2.3	<i>Features of spectraSENS</i>	26
<b>4</b>	<b>SCANNER CONTROL MODULE</b>	27

4.1	<i>Scanner systems</i>	27
4.1.1	<i>Linear scanner system</i>	28
4.1.2	<i>Rotational scanner system</i>	28
4.1.3	<i>Mirror scanner system</i>	29
4.2	<i>Scanner API design</i>	30
4.3	<i>Programming motion control</i>	31
4.4	<i>Automatic calculation of scan speed</i>	34
5	<b>CAMERA CONTROL MODULE</b>	37
5.1	<i>Camera</i>	37
5.1.1	<i>Image readout mode</i>	40
5.2	<i>Configuring camera parameters</i>	41
5.3	<i>Spectral calibration</i>	44
5.4	<i>Configuring data acquisition process</i>	47
5.5	<i>Saving hyperspectral data as ENVI file</i>	48
5.5.1	<i>ENVI header format</i>	48
5.5.2	<i>ENVI - A programmer's perspective</i>	50
6	<b>APPLICATION DEVELOPMENT</b>	52
6.1	<i>Graphical User Interface design</i>	53
6.1.1	<i>Design</i>	54
6.1.2	<i>Prototype</i>	55
6.1.3	<i>Testing</i>	56
6.1.4	<i>Development</i>	56
6.2	<i>Data visualisation techniques</i>	57
6.2.1	<i>Frame display</i>	57
6.2.2	<i>Live acquisition display</i>	61
6.2.3	<i>Waterfall display</i>	62
6.2.4	<i>Preview display</i>	66
6.3	<i>Data recording</i>	67
6.4	<i>Data correction</i>	69
7	<b>APPLICATIONS OF <i>spectraSENS</i></b>	74
7.1	<i>Measurement procedure</i>	74

7.2	<i>Application I – Quality assessment of meat</i>	75
7.3	<i>Application II – Monitoring of vegetation samples</i>	82
7.4	<i>Application III – Counterfeit detection and ink analysis in documents</i>	87
8	<b>CONCLUSION AND FUTURE WORK</b>	92
	<b>REFERENCES</b>	94

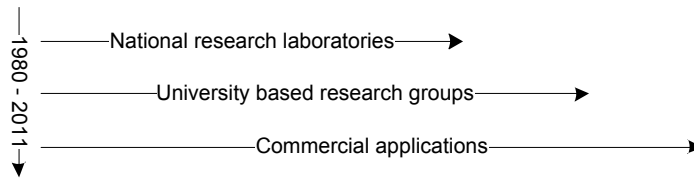
# 1. INTRODUCTION

Hyperspectral imaging (HSI) technology combines imaging with spectroscopy. The hyperspectral sensors split the light reflected by an object into different wavelength bands and record each band as an image. Thus both spatial and spectral information of an object of interest are acquired simultaneously. HSI can show information from a much wider spectrum, ultraviolet through visible and into the far infra red, at a greater resolution than the human eye can view the visible range. For example, HSI in the Shortwave Infra Red (SWIR) spectral region provides detailed information about the chemical composition of tablets, which is not possible using normal imaging. HSI provides significantly more information when compared to many of the other existing imaging techniques. However, this greater amount of information leads to larger dataset size and a much higher degree of complexity needed to interpret the data. Hyperspectral imaging systems are therefore multifaceted and involve simultaneous control of various components to collect the data and analyse it. So the challenge in this thesis is to implement a real time hyperspectral imaging system that will acquire and interpret data in a meaningful way.

## 1.1 MOTIVATION AND SCOPE OF RESEARCH

During the 1980's, National Aeronautics and Space Administration (NASA) employed hyperspectral imaging to produce remote sensing images of the earth's surface (<http://aviris.jpl.nasa.gov/>).





*Figure 1.1 Evolution of HSI*

Initially, the lack of processing power and the requirement of bulky and expensive hardware slowed the growth of HSI. During the 1990's, as computer processors became cheaper and more powerful, and with the development of sensitive detectors and faster data communication techniques – with a higher degree of interoperability, the once nascent technology attracted more researchers in various fields. In the last decade, the commercial sector has begun to embrace HSI.

HSI system contains various components such as lens optics, an imaging spectrometer, image acquisition devices, scanning devices, illumination system and data and signal processing hardware. It is imperative that all these various components are configured accurately and together for smooth and efficient operation of the system. Currently, HSI is not widely adopted, as it requires specialist knowledge of various fields including signal processing, image processing, optics and computing. The objective of this work was to produce a set of software tools, which will seamlessly link these various components by abstracting the complexity of the system so that a layman can utilise HSI for his/her own application.

Commercial applications demand real time hyperspectral imaging to achieve high throughput. The acquired data has to be presented to the system operator in a more meaningful way, which allows him/her to easily interpret the results. One aspect of this work has focused on implementing appropriate data reduction techniques to remap hyperspectral data into

smaller dimensions while still retaining the critical aspects of the acquired data for a particular application.

Real time imaging is achieved by appropriate design of the software and hardware architectures. The research has also focused on designing and implementing appropriate techniques for real-time data acquisition and visualisation. The main challenge here is to identify, extract and present essential information from the multidimensional dataset.

## **1.2 STRUCTURE OF THE THESIS**

The organisation of the thesis is as follows.

Chapter 2 explains the theory of HSI. Examples of hyperspectral images are presented. The benefits of HSI over other imaging technologies are discussed. This chapter also provides a brief description of various acquisition techniques adopted in HSI. A brief evolution and some applications of this technology are presented.

Chapter 3 explains the design of the system hardware and software. Firstly, the instrumentation involved in HSI is explained. The hyperspectral data acquisition technique adopted in this research is a pushbroom scanning technique. The theory and implementation of this technique is discussed. The design of the software architecture is explained.

Chapters 4, 5 and 6 explain the development phase of the software. Chapters 4 and 5 explain the development of control modules for two main components of a pushbroom HSI system, which includes the scanner and camera subsystem respectively. These chapters also present the novel ideas, which were incorporated to provide fully automated and efficient control of these components.

Chapter 6 describes the development of the application layer of the software, which integrates the hardware control modules for generating hyperspectral data. The application layer is an interface between the user and the hardware control modules. This chapter discusses the design of the Graphical User Interface. The techniques adopted for hyperspectral data visualisation and post processing are described.

Chapter 7 presents the results that were obtained by employing the software tools for various applications.

Chapter 8 provides the concluding remarks and a summary of this thesis, along with possible directions for future work.

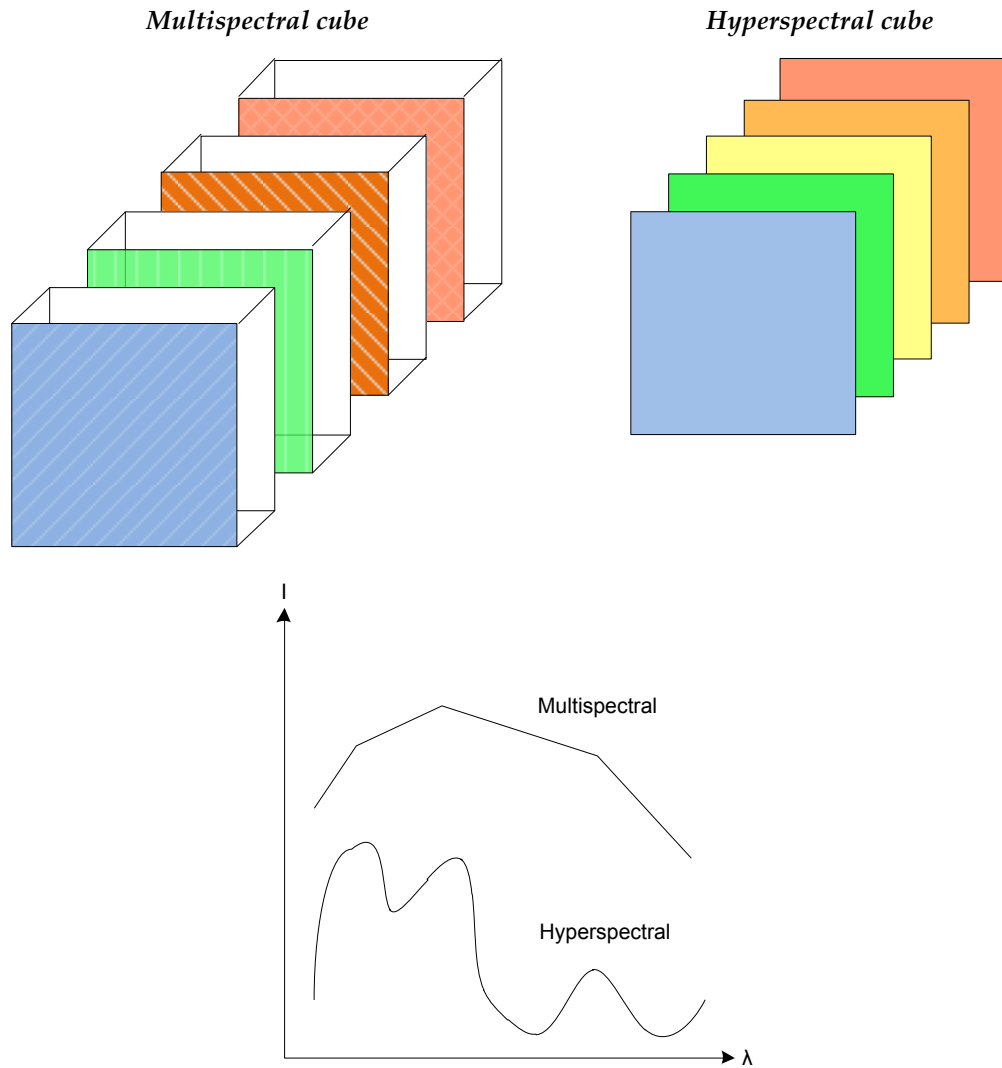
## 2. INVISIBLE WORLD MADE VISIBLE!

### 2.1 HYPERSPECTRAL IMAGING

Hyperspectral imaging (HSI) belongs to the class of spectral imaging. The framework of spectral imaging was laid by Issac Newton when he discovered in 1666, that white light is heterogeneous and is composed of primary colours. Spectral imaging is a combination of spectroscopy and imaging. The concept of spectral imaging led to the initial development of multispectral imaging (MSI) which further developed into hyperspectral imaging. MSI collects images in broad non contiguous wavelength bands. This technique collects images in typically no more than ten or so wavelength bands. HSI extends the concept of MSI and collects hundreds to thousands of narrow bands over a contiguous spectral range and thus provides dramatically more detailed information. Figure 2.1 illustrates the differences between HSI and MSI. Table 2.1 provides the main differences among the normal imaging, MSI and HSI modes.

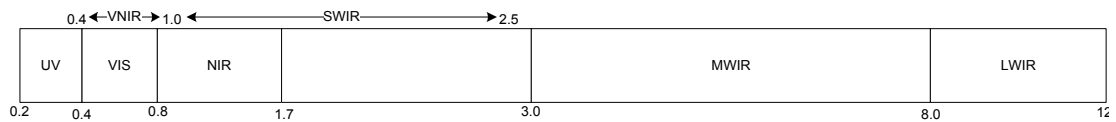
<b>MODE</b>	<b>NUMBER OF SPECTRAL BANDS</b>	<b>SPECTRAL RESOLUTION</b>	<b>CAPABILITY</b>
<b>IMAGING</b>	None	None	Image brightness
<b>MSI</b>	Few to tens	Broad	Detects solids and liquids
<b>HSI</b>	Hundreds to thousands	Narrow	Detects, identifies and quantifies solids and liquids

*Table 2.1 Differences - imaging modes*



*Figure 2.1 Multispectral and Hyperspectral- Differences*

Every object reflects or emits light from its surface. Human eyes can see only the visible light. This visible light occupies a tiny portion in the electromagnetic spectrum which lies between 400nm to 800nm. This is shown in Figure 2.2. The rest of the electromagnetic spectrum is invisible to the human eye. This invisible spectrum, ultraviolet or infrared, contains abundant information which helps us to understand the properties and nature of objects under study. Hyperspectral imaging reveals this invisible spectrum.

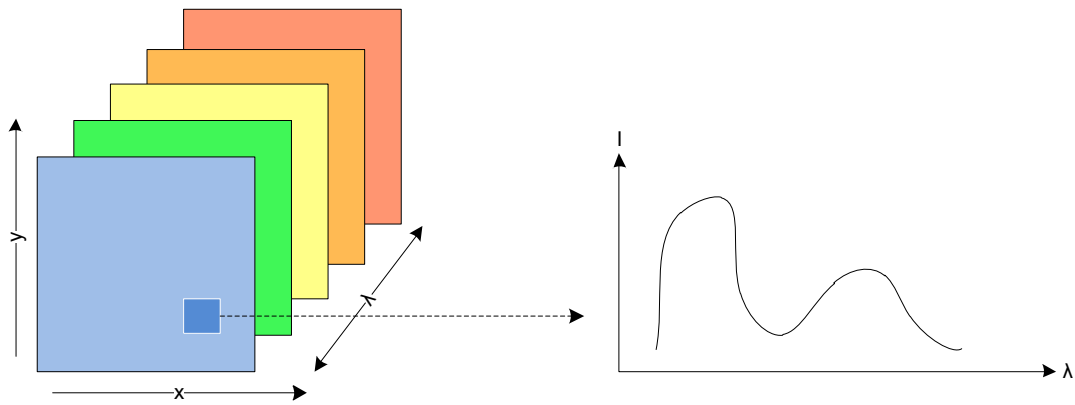


*Figure 2.2 Electromagnetic spectrum – Wavelength regions*

REGION OF ELECTROMAGNETIC SPECTRUM	WAVELENGTH RANGE (nm)
Ultraviolet (UV)	200 – 400
Visible (VIS)	380 – 800
Visible Near Infra red (VNIR)	400 – 1000
Near Infra red (NIR)	900 – 1700
Short Wave Infra red (SWIR)	970 – 2500
Mid Wave Infra red (MWIR)	3000 – 8000
Long Wave Infra red (LWIR)	8000 – 12000

*Table 2.2 Regions of electromagnetic spectrum and wavelength range*

Hyperspectral imaging (HSI) refers to collecting and analysing information of a target scene or object over a wide range of narrow wavelength bands. A hyperspectral image cube consists of a set of images layered on top of one another. Each image represents one particular wavelength band. This set of images form what is termed “the hypercube”. Thus in a hyperspectral image, each pixel consists of a spectrum over an appropriate spectral region which is represented in Figure 2.3. Every object has a unique characteristic across these different wavelength bands. This unique characteristic is referred to its ‘spectral signature’. For example, by comparing the spectral signature of the unknown target objects to that of known substances, the chemical composition of the object can be identified. The property of HSI, which looks beyond the visible spectrum, enables not only the identification of objects but also their classification and quantification to a high degree of accuracy.



*Figure 2.3 Hypercube – Pictorial representation*

A normal image can be represented as a two dimensional array of pixels or picture elements. It represents the horizontal and vertical spatial dimensions. A normal greyscale image represents one broad spectral band. An RGB or a colour image represents three broad wavelength bands for red, green and blue colours. A hyperspectral image consists of a four dimensional image cube where the third dimension represents wavelength. The structure of hypercube is shown in Figure 2.3. The x and y axis indicate the two spatial dimensions and  $\lambda$  represents the third dimension which is the wavelength. The fourth dimension is the intensity (I) at each wavelength position. Examples of a greyscale image, RGB image, image at 700nm and image at 800nm of a banknote is shown in figures 2.4.1, 2.4.2, 2.4.3, 2.4.4 respectively.



*Figure 2.4.1 Example of a greyscale image of a banknote*



Figure 2.4.2 Example of an RGB image of a banknote

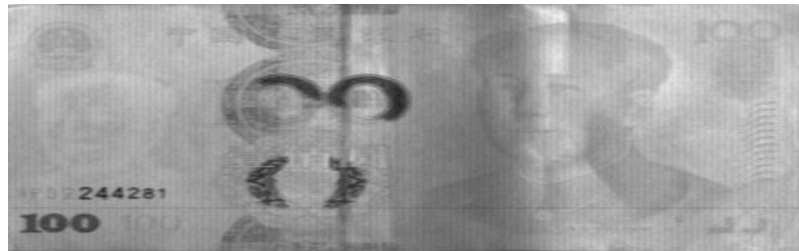


Figure 2.4.3 Example of an image of a banknote at 700nm

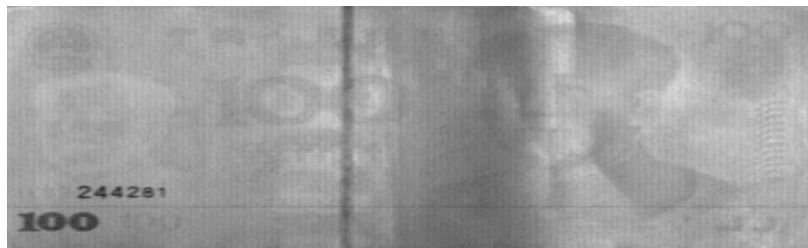


Figure 2.4.4 Example of an image of a banknote at 800nm

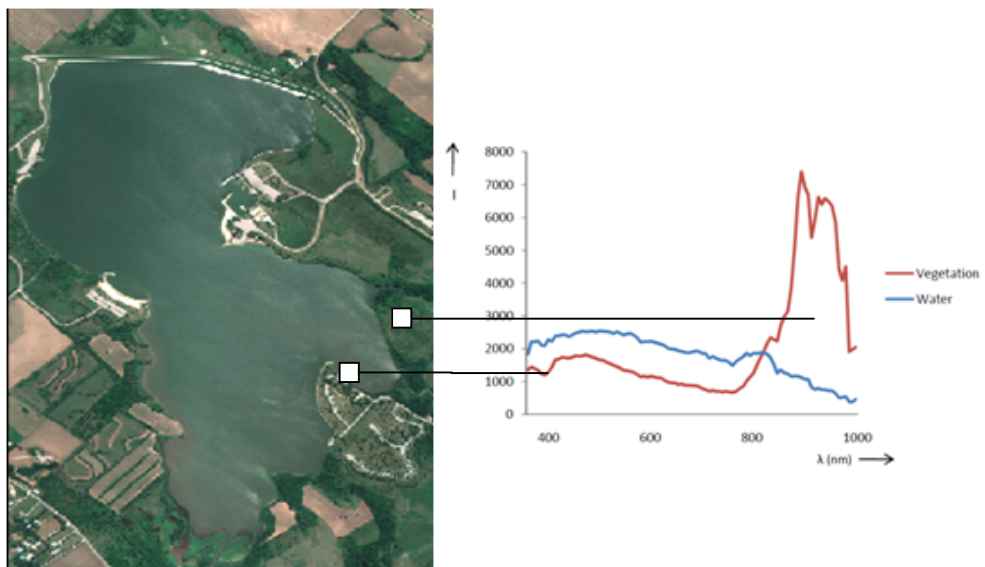


Figure 2.5 Hyperspectral image (from AISA Eagle sensor) – Spectral profile of water and vegetation.



## 2.2 HYPERSPECTRAL IMAGE ACQUISITION

Hyperspectral image acquisition refers to the process of generating a hypercube of a target scene. This acquisition is hardware dependent. The main approaches used to collect hyperspectral data are as follows. These techniques are discussed in detail in Chapter 3 of this thesis,

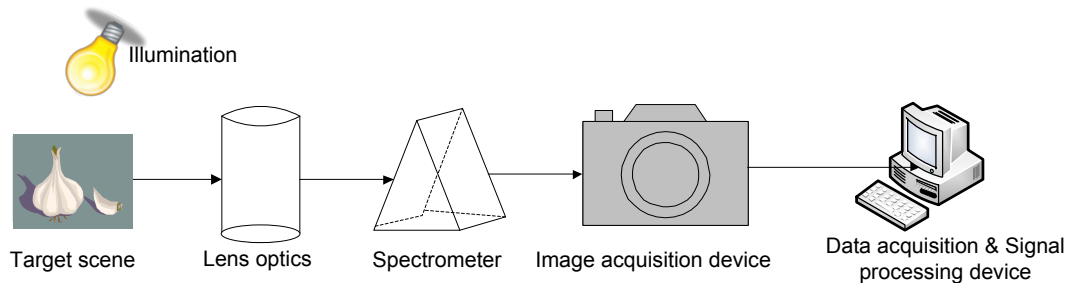
1. Pushbroom scanning technique
2. Whiskbroom scanning technique
3. Liquid Crystal Tuneable filters (LCTF)

These acquisition techniques employ devices such as lenses, optical components, filters, imaging sensors, scanning devices to capture images of the target at different wavelength bands. In all of the above acquisition techniques, proper illumination of the target scene is required. The light reflected by the target scene is focussed into the spectroscope. The main component of a HSI system is the spectroscope; the device that separates the light into its constituent spectral colours. This can be performed by any of the following components

1. Prism
2. Transmission grating
3. Reflective grating
4. Filter wheel
5. Linear variable filter
6. Tunable filter
7. Interferometer

The dispersed light is captured by a two-dimensional image acquisition device such as a Charge Coupled Device (CCD). The data captured by this device is then transferred to the host computer for further data processing and visualisation. A generic layout for a hyperspectral data acquisition

system is presented in Figure 2.6. The sensor type and optical arrangement of the hyperspectral system can vary depending on the region of the spectrum that is captured; say UV, VIS, NIR, SWIR or LWIR.

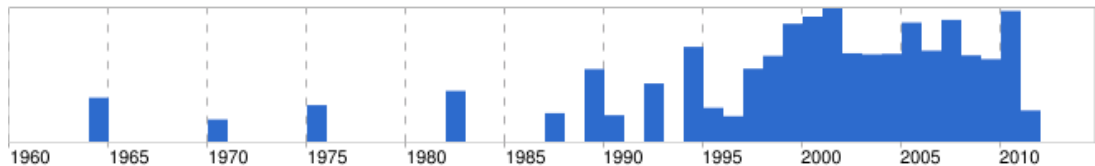


*Figure 2.6 Hyperspectral Imaging System – Schematic layout*

### **2.3 APPLICATIONS OF HYPERSPECTRAL IMAGING**

HSI finds itself being utilised in many different fields. The concept of hyperspectral imaging was first applied to remote sensing. The first multispectral imaging system was launched in 1972 aboard the LandSat 1 satellite which was formerly known as Earth Resources Technology Satellite (ERTS) [1]. This system collected remote sensing data of the Earth's surface over four spectral bands and a spatial resolution of 80m wide pixels. The spectral data acquired was used in applications such as agriculture, oceanography, forestry and environmental monitoring. The beginning of HSI was marked by the arrival of NASA/Jet Propulsion Laboratory (JPL) Airborne Visible/ Infrared Imaging Spectrometer (AVIRIS) in 1989 (<http://aviris.jpl.nasa.gov/>). AVIRIS collected images in 220 spectral bands over the range of 400 nm to 2500 nm. This system was capable of data rate of 16Mbps in 1994 and 20.4Mbps from 1995 to 2004. This system acquired 10 bit data from 1994, 12 bit from 1995 and 16 bit from 2005 onwards. AVIRIS was used to measure the Earth's surface and atmosphere for the study of climate change and for environmental monitoring. Following AVIRIS, many airborne

and ground based hyperspectral systems were produced. Since then, various researches have demonstrated the benefits of HSI technology in a wide range of applications such as medical, forensic, pharmaceutical sciences, mining, food monitoring and so on. A screenshot of Google timeline representing the amount of activity in HSI between 1960 and 2011 is shown in Figure 2.7.



*Figure 2.7 Screenshot of Google Timeline representing the activity in HSI between 1960 - 2011*

The applications of HSI depend on the region of the electromagnetic spectrum that is measured. Table 2.3 provides a list of the applications that are feasible in different wavelength regions.

WAVELENGTH REGION	APPLICATIONS
UV	Forensics, Semiconductors
VIS	Art, Colour measurement
VNIR	Fruit quality
NIR	Plastic sorting, Polymers, Food
SWIR	Pharmaceuticals, Chemical imaging, Minerals
LWIR	Minerals

*Table 2.3 Applications of HSI in different wavelength regions*

HSI is a non-destructive, non-contact technology which makes it ideal for a wide range of applications. Some examples of HSI application are discussed below.

### **2.3.1 REMOTE SENSING**

Hyperspectral remote sensing data has been used for environmental

monitoring, mineral mapping, target detection etc. HSI was used to detect oil spills in Chesapeake Bay [2]. Chesapeake Bay is highly vulnerable to oil spills which affect the wildlife and species. Air borne hyperspectral images were collected in 288 spectral bands in Visible and IR wavelengths. This data was used to predict the movement and spread of oil spill over time and its impact on wetlands and coastline. Classification techniques such as Spectral Angle Mapper (SAM) were employed to distinguish different types of oil such as crude and light oil. Conventional remote sensing techniques failed to identify minute concentrations of oil and also led to many false positive results. Recently, hyperspectral images of oil spill in the Gulf of Mexico, which occurred in April 2010, were acquired. This data has been analysed to map the oil slicks along the shoreline in order to assist the cleanup processes.

### ***2.3.2 FOOD MONITORING***

HSI has been widely used in food quality assessment and safety control. The non-destructive nature and rapid response of HSI aids the online quality control of food products. Researchers at Campden & Chorleywood Food Research Association (CCFRA) and Gilden Photonics Ltd. (GPL) developed a HSI system to determine the distribution of fat and moisture content in fried foods [3], distribution of fat in meat, freshness of fish and evaluation of baked goods. Conventional techniques such as NIR spectroscopy were able to provide only the average fat content in food. HSI facilitated the analysis of non-uniform distribution of fat in doughnut. A doughnut has been measured in NIR region. The fat and moisture shows separate characteristic absorbance bands in NIR region. The difference in spectral profiles was used to identify regions with high levels of fat content. A research team at the United States Department of Agriculture (USDA) - Agricultural Research Service (ARS)

used HSI to detect food borne pathogen *Campylobacter* in a solid media in 24 hours [4]. *Campylobacter* is a major cause of food borne illness in many countries. *Campylobacter* has a unique spectral signature in the VNIR region of the spectrum which enables its detection. Traditional techniques involved a long, time consuming process, over several weeks to detect this pathogen. HSI drastically reduced the detection time from several weeks to several hours. Classification algorithms such as band ratio thresholding, single band thresholding and spectral feature fitting were applied on the HSI data. This provided 100% accuracy in identifying this pathogen. The technique is now applied for detection of other pathogens such as salmonella etc.

### ***2.3.3 CONSERVATION OF ART AND HISTORY***

Quantitative Hyperspectral Imaging (QHSI) has been used to monitor and quantify ageing of historic documents [5]. The documents were measured in 70 spectral bands ranging from Near Ultraviolet to Infrared region (365nm – 1100nm). This data was used to quantify yellowing of paper over time. Traditionally this observation was done with the naked eye, which was inefficient and subjective. QHSI provided objective results with the help of pattern classification techniques. These results were used to determine the document's storage and exposition conditions. Such analysis helps in the development of appropriate conservation treatments.

HSI linked with humanities research has facilitated the preservation of cultural heritage objects. The Library of Congress, USA has been employing spectral imaging techniques to study ancient manuscripts, which reflects the history of United States [6]. HSI being a non-destructive, conservation- safe technique aids the analysis of fragile documents. HSI allows researchers to understand the creation techniques of the manuscripts, identify the origin of

substrate (paper etc.), characterise range of materials and reveal the previously unknown information about historic objects. The Library of Congress employed a Mega-Vision Equipoise Imaging system with a 39 Megapixel monochrome camera and conservation safe lighting. This system collects spectral data from 365 nm – 1050 nm. The analysis of this data has revealed hidden information in the manuscripts. For example, the analysis of The Declaration of Independence (1776) has revealed an interesting word correction made by Thomas Jefferson during his writing. This is illustrated in Figure 2.8.



*Figure 2.8 Word correction made in Declaration of Independence by Thomas Jefferson revealed by HSI. The word 'subjects' was over written with 'citizens'. Image courtesy-Library of Congress*

### **2.3.4 MEDICAL SCIENCES**

HSI has been applied in the medical field for the detection of tumours and skin cancers, to monitor healing of wounds and burns, for imaging of retina etc. Researchers at Heriot Watt University have developed a HSI system of human retina which enables early detection of retinal diseases such as glaucoma and diabetic retinopathy [7]. These diseases might lead to loss of

vision if not detected in early stages. This system has been able to detect variations in oxygenation of haemoglobin in retinal blood vessels. Linear spectral unmixing technique has been used to produce qualitative oximetric maps of the retina. This will aid in early detection of retinal diseases and to monitor the treatment process.

### 3. spectraSENS

spectraSENS is a visualisation and imaging system which is designed to facilitate easy interaction and control of a Hyperspectral Imaging (HSI) system. This was developed in collaboration between the University of Strathclyde and Gilden Photonics Ltd.

spectraSENS presents complex multidimensional hyperspectral data to the user, in a form that can be easily interpreted. spectraSENS also works in real time.

spectraSENS system has two aspects of components, namely,

1. Hardware components of spectraSENS

This covers the hardware aspects of the system for modelling a push broom hyperspectral imaging system

2. Software components of spectraSENS

This covers the programming of hardware components such as the operations of camera, stepper motor to control motion of the scanner; it also covers other aspects such as synchronisation of scanner motion with data acquisition and various functionalities associated with the hypercube; programming of graphical user interface for operation control.

These components are illustrated in Figure 3.1.



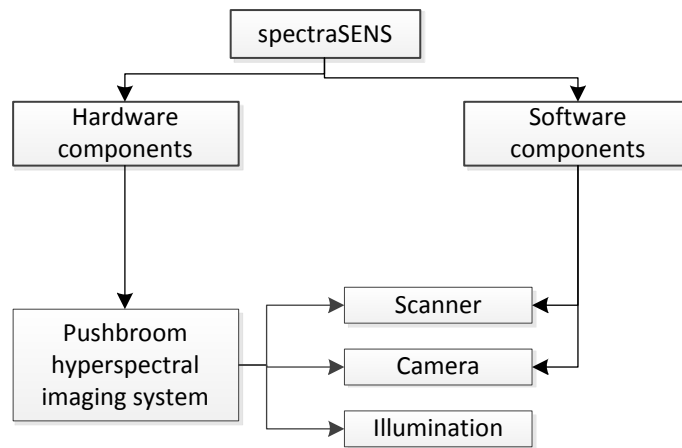


Figure 3.1 spectraSENS - Overview

### 3.1 HARDWARE COMPONENTS OF spectraSENS

Hardware systems such as whiskbroom, electronically tunable filters and pushbroom are used to acquire hyperspectral data.

#### 3.1.1 DATA ACQUISITION TECHNIQUES

There are two main approaches to building a hyperspectral image cube.

1. Capturing spatial dimension over time – Here, the spectral signature of one pixel to many thousand pixels in a spatial line (swath) across the target area is recorded at a time. The second spatial dimension is built over time by sequentially recording one swath after another. The whiskbroom and pushbroom acquisition techniques are examples for this approach.

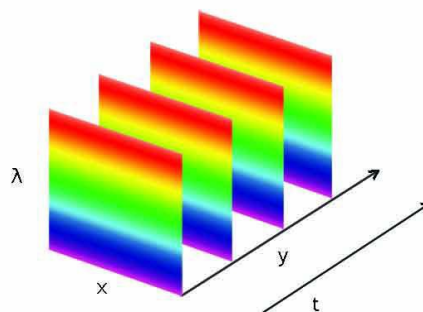
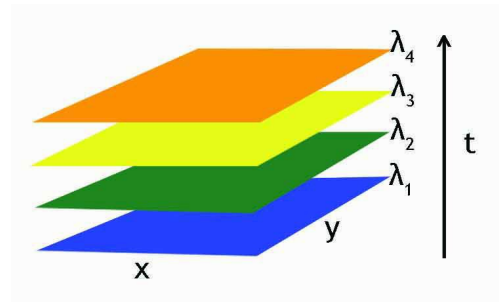


Figure 3.2 Capturing spatial dimension over time

2. Capturing spectral dimension over time – Here, the full spatial image of an object is recorded sequentially at each wavelength band. This approach is achieved by using electronically tunable filters (ETF) such as liquid crystal tunable filters (LCTF) combined with a charge coupled device (CCD).



*Figure 3.3 Capturing spectral dimension over time*

#### **3.1.1.1 Whiskbroom technique**

The whiskbroom technique collects hyperspectral data of one pixel at a time. The whiskbroom sensors employ rotating mirrors to scan across the target and reflect the light through the spectrometer to a linear array detector [8]. The disadvantage of this technique is that these mirrors cause spatial distortions in the image acquired. Also, this technique involves more moving parts which makes it more cumbersome and expensive to build. It is also very computationally complex to deconvolve the pixel data from the circulate whisk collection to form a 2D image.

#### **3.1.1.2 Electronically tunable filter technique**

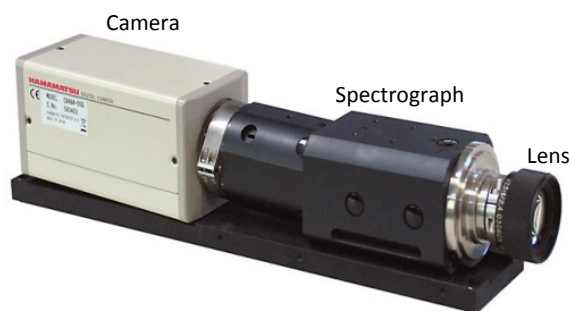
Electronically Tunable Filters (ETF) transmit radiation at one particular wavelength. This wavelength can be tuned to acquire images at a sequence of wavelengths [9]. The tunable filter approach has longer wavelength switching times in the order of 100 – 150 milliseconds. This is not a desirable feature. Also, the transmission throughput of these filters varies for different wavelengths. Hence the collected hyperspectral data has to be corrected for

these variations and spectral characteristics. In general, the pushbroom technique is more advantageous over other image acquisition techniques with an optical efficiency about 30 times higher than the tunable filter approaches.

### 3.1.1.3 Pushbroom technique

The pushbroom technique allows simultaneous acquisition of all spectral bands. A pushbroom hyperspectral camera is made up of the following components,

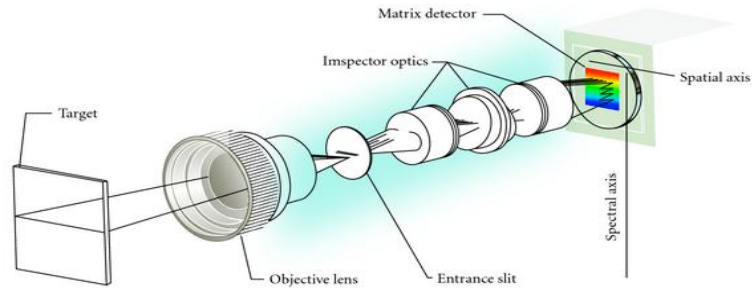
1. Camera (Two dimensional detector array)
2. Imaging Spectrograph
3. Objective lens



*Figure 3.4 Hyperspectral camera*

The fore optics, in a hyperspectral camera, image the light reflected from one line segment of the target sample onto the entrance slit in the spectrograph. This light is dispersed into different wavelengths by the dispersing element (prism / grating) in the spectrograph. This dispersed light is captured by the CCD sensor as a two dimensional image  $[X, \lambda]$  with  $X$  representing spatial position and  $\lambda$  represents spectral dimension. In order to capture the entire image, snapshots of every line across the target area have to be captured [8].

A schematic representation of pushbroom imaging is shown in Figure 3.5.



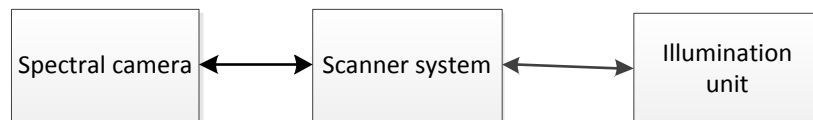
*Figure 3.5 Schematic representation of pushbroom imaging*

Moving one of the following components in a perpendicular direction to a measured line can capture the images of different lines on the target area,

1. Target object
2. Spectral camera
3. Scan mirror in front of the spectral camera

By synchronising the movement between camera and target object and the acquisition of the camera, one can capture different lines of the object and thus generate the hypercube  $[X, Y, \lambda]$  where  $Y$  represents the second spatial dimension. This technique works in a similar fashion to the pushing of a broom across the floor and hence the name. The movement between the camera and the target scene can be achieved by employing appropriate scanning systems. Thus the main components of a pushbroom hyperspectral system include

1. Spectral camera
2. Scanner system
3. Illumination unit



*Figure 3.6 Components of a HSI system*

#### **3.1.1.4 Advantages of pushbroom technique**

The advantages of pushbroom technique over other hyperspectral data acquisition techniques are

8. High ruggedness – Pushbroom systems consist of few moving parts compared to whiskbroom systems. Thus they are less complex, highly rugged and more stable.
9. Rapid HSI acquisition – The complete spectral information of a line is acquired simultaneously along with the spatial information. This facilitates high throughput and hence faster hypercube generation.
10. Reduced illumination load - Only one line across the target area needs to be illuminated at a time. This reduces the heat load on the sample drastically compared to other imaging techniques.
11. Ideal for online measurements with moving objects - The pushbroom hyperspectral system can be directly transferred from R&D labs to production line in industry with no changes to the system. For example, this can be applied for fruit sorting where the fruits are constantly moving over a conveyor belt.

#### **3.1.1.5 Design of a pushbroom hyperspectral imaging system**

A compact desktop hyperspectral imaging system has been designed and built. This system is a completely integrated pushbroom HSI system. This system is built with a high precision linear translation stage, Zolix KSA 111-200 S4N. The target object is scanned along one axis under the field of view of a hyperspectral camera. A stepper motor with intelligent control is connected to one end to drive the stage. This motor provides complete control over the positioning of the scanner and enables precise scanning.

The hyperspectral camera is mounted vertically above the scanner so that the field of view lies across the centre of the linear scanner. The hyperspectral

camera is built with a Specim V8E spectrograph connected to Andor LUCA-R camera. This spectral camera operates in the visible range 380 – 800 nm, and the resolution of the camera is 1004 \* 1002 pixels. The objective lens used was Schneider Xenoplan with a focal length of 23mm. A 150W halogen lamp is used to illuminate the target. The advantages of this system are

- High speed scanning and image acquisition
- High spatial precision
- Variable field of view
- Simple and compact

## **3.2 SOFTWARE COMPONENTS OF spectraSENS**

### **3.2.1 SOFTWARE DESIGN**

spectraSENS was designed with the following yardsticks namely,

- Usability with respect to the users; simple and user-friendly Graphical User Interface (GUI). This topic has been treated in detail in chapter 6.
- Maintaining the modularity of the framework, so that each component can be independently modified without affecting the rest of the associated components.
- spectraSENS functionalities can be extended to integrate different cameras, different spectrographs and different scanners. Providing extensibility for the existing application programming interface of spectraSENS has been kept at the core of the design.
- As spectraSENS is real time, memory management is vital. An effective memory management was achieved by keeping the entire application programming interfaces/dynamic link libraries unmanaged.
- Implementation using C++ language that provides strong object oriented features.

- Error handling and trapping are given due importance.

### **3.2.2 SOFTWARE ARCHITECTURE**

The software architecture refers to the structure of the software system that comprises various software components and the relationships among them. Edsger Dijkstra in 1968 and David Parnas in the early 1970s first introduced the concept of software architecture ([http://en.wikipedia.org/wiki/Software\\_architecture](http://en.wikipedia.org/wiki/Software_architecture)). The quality and stability of software is determined by its architecture.

The architecture of spectraSENS comprises of the following four modules,

1. Scanner control module:

This module provides an interface to the scanner hardware and is responsible for the motion control of the scanner system.

2. Camera control module:

This module provides an interface to the camera hardware. Camera manufacturers provide driver-level functions to communicate with their cameras. These functions will be packaged in different software languages. Camera control interface invokes these functionalities to configure the camera parameters.

3. Application layer:

The scanner control and camera control modules are integrated into a single functional layer called the application layer. This layer synchronises the scanner motion with the camera's acquisition to generate the hypercube.

4. User interface and visualisation module:

A GUI is an interface based on graphics that allows the user to interact with the application. The GUI takes input from the user to control the camera parameters, scan parameters and acquisition process with the help of graphical components like icons, menus and windows. The image data is displayed using the visualisation module. The visualisation module employs real time hyperspectral data visualisation techniques.

The software architecture of spectraSENS is illustrated in Figure 3.7.

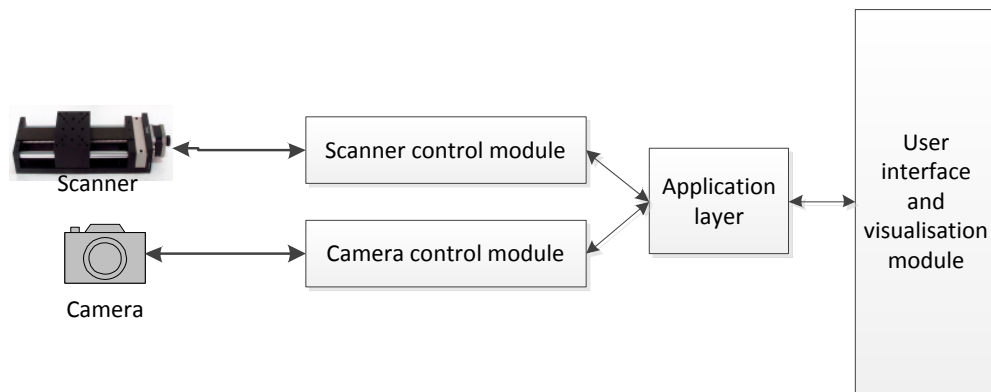


Figure 3.7 spectraSENS - Software architecture

The scanner control and camera control modules are implemented as Application Programming Interfaces (API). An API is an interface for a software program, which can be used, by other software programs in order to integrate with each other. The scanner and camera API provide a set of functions, which is used by the application layer in order to communicate with the scanner and camera respectively.

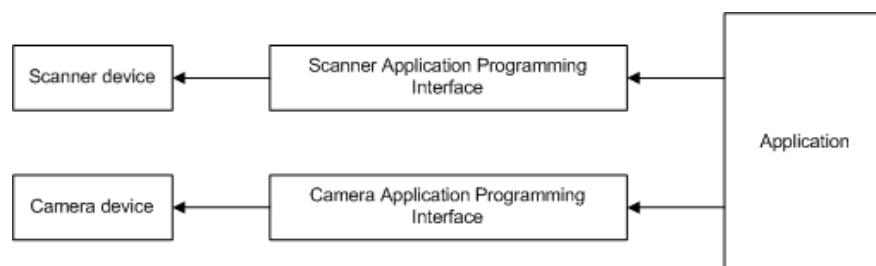


Figure 3.8 spectraSENS – Interface to hardware

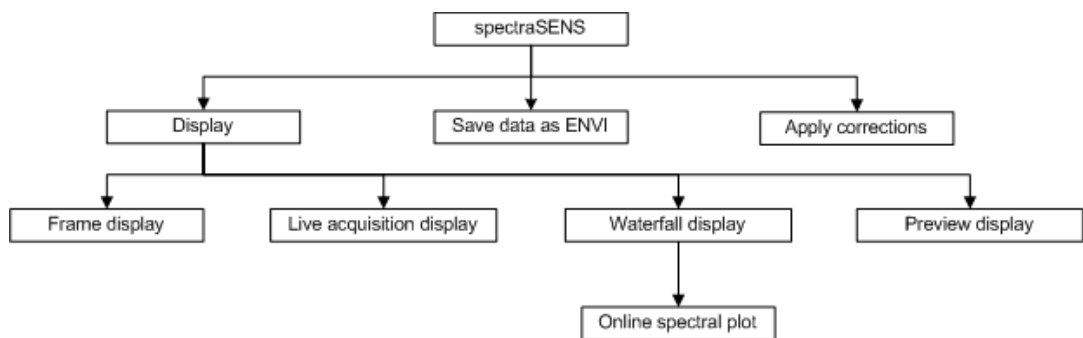


### 3.2.3 FEATURES OF *spectraSENS*

This section provides an overview of the features present in *spectraSENS*.

These features are discussed in detail in Chapter 6.

1. Generation of Hypercube
2. Hypercube display techniques
  - a. Frame display
  - b. Live acquisition display
  - c. Waterfall display with online spectral analysis
  - d. Preview display
3. Save hypercube in ENVI format
4. Apply corrections to hypercube



*Figure 3.9 spectraSENS - Features*

## **4. SCANNER CONTROL MODULE**

In the pushbroom HSI technique, one row of pixels across the target area is acquired at a time which is known as a swath and the full spectral information of each pixel in the swath is recorded simultaneously. Therefore, in order to construct a three dimensional hypercube, either the target or the camera is scanned along the track. In order to move the target or the camera, different types of high precision scanning devices are employed.

spectraSENS software is designed to provide an integrated control of the scanner system and image acquisition devices. The scanner control module deals with scanning operations. This module is programmed as an Application Programming Interface (API). The scanner API provides a set of functions which are used to control the motion of the scanning device.

### **4.1 SCANNER SYSTEMS**

The scanner systems are high precision devices which facilitate controlled movement of an object. In HSI, the scanner system is employed to move the target object over the field of view of the spectral camera. There are three types of commonly used scanner systems which include,

- Linear scanner system
- Rotational scanner system
- Mirror scanner system

The above three scanner systems are driven by a stepper motor with intelligent motor control. This device is integrated with a microstepping driver and a programmable motion controller. The speed of rotation of the motor, number of steps rotated and direction of rotation are programmable.

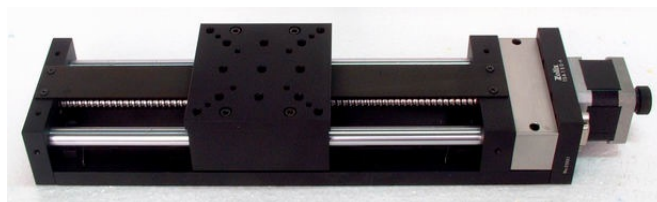
These parameters determine the speed, distance moved and direction of motion of the scanner. The motion control of the scanner system is illustrated in Figure 4.1.



*Figure 4.1 Motion control of scanner system*

#### **4.1.1 LINEAR SCANNER SYSTEM**

A linear scanner system or linear translation stage is used to move the target object or the hyperspectral camera along one linear axis. It consists of a moving platform and a fixed base. A stepper motor is connected to one end of the stage to drive the platform. For every revolution of the motor, the platform moves by a specific distance called the pitch. The direction of rotation of motor determines the direction of motion of the platform. The translation stage consists of limit switches on both ends and a zero datum sensor at one end. These switches and sensor restrict the motion of platform and protect the stage from any damage. A picture of linear stage is shown in Figure 4.2.



*Figure 4.2 Linear stage*

#### **4.1.2 ROTATIONAL SCANNER SYSTEM**

A rotational scanner system or rotational stage is used to rotate the target

object or the hyperspectral camera at a specified angle. These systems are mainly used to cover wide scene areas. The rotational stage consists of a moving platform, fixed base, zero datum sensor and a stepper motor. For every revolution of the motor, the stage rotates by a specific angle. The direction of rotation of the motor determines the direction of rotation of the stage. A picture of the rotational stage is shown in Figure 4.3.



*Figure 4.3 Rotational stage*

#### **4.1.3 MIRROR SCANNER SYSTEM**

A mirror scanner system consists of a mirror which is rotated by a stepper motor. The mirror is located in front of the lens of the hyperspectral camera. As the mirror moves in front of the lens, a three dimensional hypercube is constructed line by line. These systems are mainly used to capture wide scene areas or stationary objects. For every revolution of the motor, the mirror rotates by a specific angle. The direction of rotation of the motor determines the direction of rotation of the mirror. A picture of the mirror scanner system is shown in Figure 4.4.



*Figure 4.4 Mirror scanner*

## 4.2 SCANNER API DESIGN

The scanner API is designed to provide complete control over the motion of the scanner. The scanner API is designed to support the three types of scanner systems which include linear, rotation and mirror scanner systems. A functional review was done and the specification for the API was prepared. The functional specification for the scanner API is provided in table 4.1.

Scanner functions	Description
SE_InitialisePort	Automatically detect motor and establish communication with the motor
SE_MoveToStart	Move the scanner to the START position specified for the scan operation. This position is defined by using the zero datum sensor as the zero reference
SE_StartScan	Start the scanning process i.e. moving the scanner forward and backward by the scan distance and scan speed specified
SE_StopScan	Stop the motion of the scanner
SE_MoveSomeForward	Move forward by a specific distance
SE_MoveSomeBackward	Move backward by a specific distance
SE_CheckMotionStatus	Check whether the scanner is moving or stationary
SE_CheckMotionDirection	Check the direction of motion of the scanner

*Table 4.1 Scanner API - Specifications*

### 4.3 PROGRAMMING MOTION CONTROL

The precise motion of the scanner is controlled by programming the stepper motor. The specification of the motor being programmed is listed in Table 4.2. The motor is programmed to control

4.2. The motor is programmed to control

- Speed of the scanner
- Distance moved by the scanner
- Position of the scanner with the help of zero datum sensor and limit switches

Motor model	IMS MDrive 17 Plus (Motor + Driver)
Step angle	1.8°
Voltage range	+12 to +48V DC
Step range	20 µstep resolution upto 51200 µsteps per revolution
Programming language	MCode

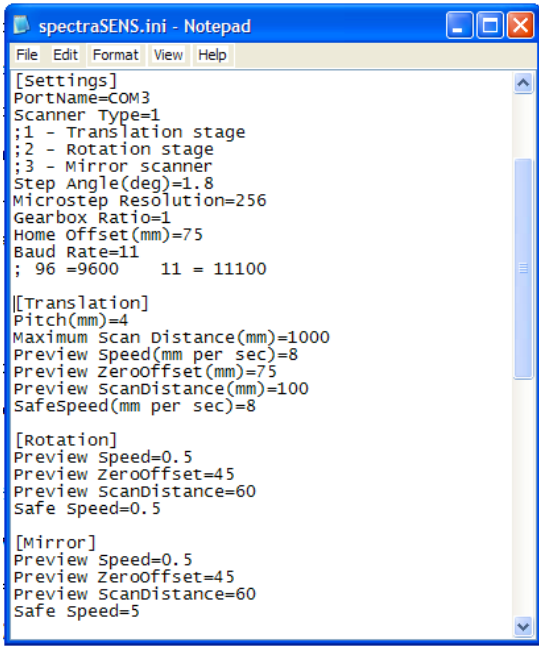
*Table 4.2 Motor – Specifications*

The motion controller of stepper motor, IMS MDrive was programmed using MCode programming language. The motor communicates over a serial RS 422/485 interface. MCode is the programming language for IMS MDrive devices which communicate over RS 422/485 interface. MCode programs were developed for each of the scanner API functionality [Table 4.1]. These MCode programs were tested on the motor using the IMS terminal, a programming interface which allows user to communicate and program MDrive devices. Once the MCode programs were tested, they were wrapped in individual C++ functions.



Figure 4.5 Program flow

The scan functions take user inputs for scan speed, zero offset distance and scan distance in millimetre (mm) units for linear stage or in degree (deg) units for rotary and mirror scanner systems. These values are converted in terms of motor step units. The unit conversion is performed according to the type of scanner system. The scanner parameters such as scanner type, microstep resolution of motor, step angle, pitch, baud rate and gear box ratio are configured in a configuration file, 'spectraSENS.ini'. The reason for storing these values in a configuration file is that these values are not frequently accessed by the user. A screenshot of spectraSENS INI file is provided in Figure 4.6.



```
spectraSENS.ini - Notepad
File Edit Format View Help
[Settings]
PortName=COM3
Scanner Type=1
;1 - Translation stage
;2 - Rotation stage
;3 - Mirror scanner
Step Angle(deg)=1.8
Microstep Resolution=256
Gearbox Ratio=1
Home Offset(mm)=75
Baud Rate=11
; 96 =9600 11 = 11100

[[Translation]
Pitch(mm)=4
Maximum Scan Distance(mm)=1000
Preview Speed(mm per sec)=8
Preview ZeroOffset(mm)=75
Preview ScanDistance(mm)=100
SafeSpeed(mm per sec)=8

[Rotation]
Preview Speed=0.5
Preview ZeroOffset=45
Preview ScanDistance=60
Safe Speed=0.5

[Mirror]
Preview Speed=0.5
Preview ZeroOffset=45
Preview ScanDistance=60
Safe Speed=5
```

Figure 4.6 Screenshot of spectraSENS.ini file

The procedure for converting scan speed, distance and zero offset from millimetre or degree units to motor step units is provided below.

Step angle ( $\theta_s$ ) = Amount of angle covered for each step of

the motor

$$\begin{aligned} \text{Steps per rev (S}_{\text{rev}}) &= \text{Number of steps for one revolution} \\ &= 360^\circ / \theta_s \end{aligned} \quad (4.1)$$

$$\text{Microstep resolution } (\mu_s) = \text{Number of microsteps for one whole step}$$

$$\begin{aligned} \text{Microsteps per rev ( } \mu\text{S}_{\text{rev}}) &= \text{Number of microsteps for one revolution} \\ &= S_{\text{rev}} * \mu_s \end{aligned} \quad (4.2)$$

$$\begin{aligned} \text{Pitch (d)} &= \text{Distance (in mm or degree) moved by the} \\ &\quad \text{scanner for one revolution of motor i.e. for} \\ &\quad \mu\text{S}_{\text{rev}} \text{ microsteps} \end{aligned}$$

Number of  $\mu$ steps rotated

$$\text{for 'x' mm or 'x' degree} = (x * \mu\text{S}_{\text{rev}}) / d \quad (4.3)$$

For example, for a scan distance of 100 mm, the number of motor steps is calculated as follows,

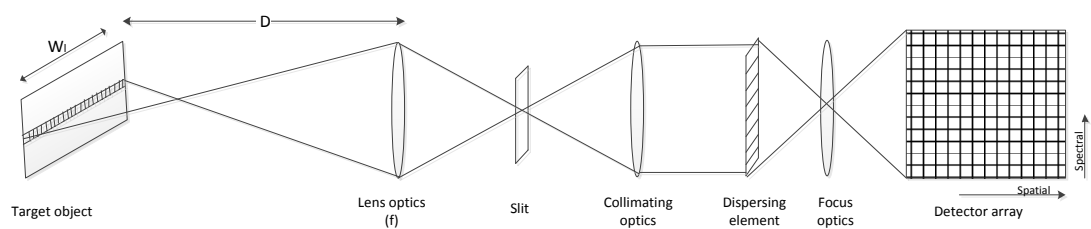
$$\begin{aligned} \text{Step angle of IMS MDrive 17 plus motor} &\quad \theta_s = 1.8^\circ \\ \text{Steps per revolution} &\quad S_{\text{rev}} = 360^\circ / \theta_s \\ &\quad = 360 / 1.8 = 200 \text{ steps} \\ \text{Microstep resolution specified} &\quad \mu_s = 256 \\ \text{Microsteps per revolution} &\quad \mu\text{S}_{\text{rev}} = S_{\text{rev}} * \mu_s \\ &\quad = 200 * 256 \\ &\quad = 51200 \\ \text{Pitch of linear scanner system, TSA 200 B} &\quad d = 4 \text{ mm} \\ \text{So, the number of microsteps rotated by} &\quad = (x * \mu\text{S}_{\text{rev}}) / d \\ \text{motor for 100mm} &\quad = (100 * 51200) / 4 \\ &\quad = 1280000 \text{ microsteps} \end{aligned}$$



The MCode programs are customised with these calculated values of scan speed, distance and zero offset in terms of motor steps units. A serial communication module is implemented in Microsoft Visual Studio C++ environment using Windows Application programming interface (API). This module establishes serial communication with the motor to download and execute the MCode programs.

#### 4.4 AUTOMATIC CALCULATION OF SCAN SPEED

The speed at which the target is scanned determines the number of lines captured by the spectral camera to build a hyperspectral cube. The representation of one scan line is provided in Figure 4.7. If the target is scanned at too high or too low speed, the resulting image of the target will look compressed or stretched accordingly. Therefore, the target has to be scanned at an optimum speed so that the generated pixels are square in shape. A procedure to determine this optimum scan speed has been implemented.



*Figure 4.7 Acquisition of one scan line*

The optimum scan speed to form square pixels is calculated from the frame rate of the camera as follows,

The swath  $W_1$  (width of one scan line) covered by the camera is defined by the image width of the CCD sensor or slit width of the spectrograph  $W_1$  (in

mm), distance between target and lens  $D$  (in mm) and the focal length of lens  $f$  (in mm). The limit to the effective swath will be the smallest value between the CCD's image width and the slit width. The swath or line width (in mm) is given by

$$W_1 = (W_i * D) / f \quad (4.4)$$

The width of one target pixel  $W_p$  (in mm) is defined by the swath and the number of pixels in one scan line ( $N_p$ ). The value of  $N_p$  is determined by the number of active pixels ( $X$ ) of the sensor and the binning factor ( $B_w$ ),

$$N_p = X / B_w \quad (4.5)$$

$$W_p = W_1 / N_p \quad (4.6)$$

In order to form square pixels, the target has to be moved by the distance  $W_p$  in the time taken to capture one line (frame). Thus the scan speed is represented as,

$$\text{Scan speed (mm/s)} = W_p * R \quad (4.7)$$

where  $R$  is the detector frame rate in frames per second (fps).

As an example, for a hyperspectral camera, with a sensor width of 8mm, slit width 14mm, horizontal resolution ( $X$ ) of 1004 pixels, binning factor ( $B_w$ ) of 1, target distance ( $D$ ) of 280mm and focal length ( $f$ ) 23mm, let's calculate the optimum scan speed. Here, the limit to the effective swath is the sensor width ( $W_i$ ).

$$\begin{aligned} \text{Width of swath} \quad W_1 &= (W_i * D) / f \\ &= (8 * 280) / 23 \\ &= 97.4 \text{ mm} \end{aligned}$$

$$\begin{aligned} \text{Width of one ground pixel} \quad W_p &= W_1 / N_p = W_1 / (X / B_w) \\ &= 97.4 / (1004/1) \\ &= 0.979 \text{ mm} \end{aligned}$$

For a frame rate (R) of 12fps,

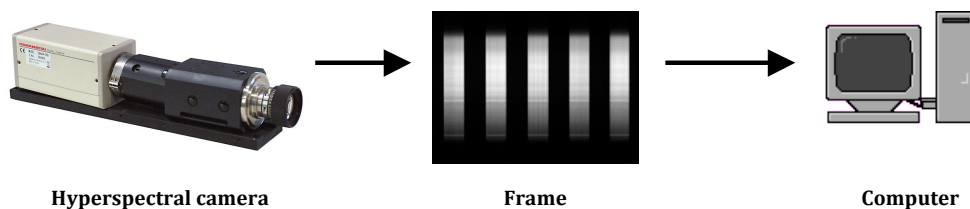
$$\begin{aligned}\text{Scan speed} &= W_p * R \\ &= 0.979 * 12 \\ &= 11.75 \text{ mm/sec}\end{aligned}$$

## 5. CAMERA CONTROL MODULE

The camera control module provides access to the main component of the hyperspectral system, the camera. This module is used to configure camera parameters and control the data acquisition process. This module has been programmed as an Application Programming Interface (API), which provides a set of camera control functions. The camera manufacturer defines the access functions to the camera.

### 5.1 CAMERA

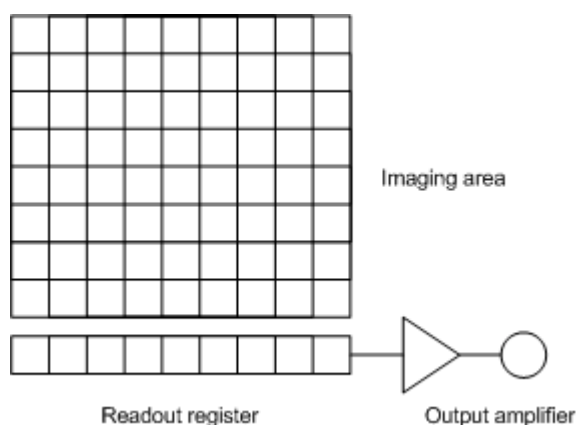
A pushbroom hyperspectral camera consists of a digital camera, imaging spectrograph and a front lens. The camera captures the light dispersed by spectrograph and transfers the digital image to the computer. This is represented in Figure 5.1.



*Figure 5.1 Working of the camera*

The basic building block of a camera is the Charge Coupled Device (CCD). A CCD is a two dimensional array of picture elements or pixels where each pixel is capable of detecting light. During acquisition, the CCD is exposed to light for a specified period of time and an image is captured. The electronic

charge accumulated on the CCD detector is then shifted vertically to the readout register at the bottom, one row at a time shown in Figure 5.2. This charge is shifted horizontally to the output amplifier, pixel by pixel. This amplified signal is fed to an analog to digital converter (ADC) to form a digital image.



*Figure 5.2 Working of CCD camera*

spectraSENS has been successfully integrated with four high resolution digital cameras. The camera control module has been designed to control any of the following cameras,

2. Andor Luca (R)
3. Hamamatsu ORCA C8484 – 05G
4. Kappa DX4 – 285 GigE
5. Xenics Xeva 1.7 – 320

Detailed specifications for these cameras are given in Table 5.2.

Table 5.1 provides a list of functions exported by the camera API.

Camera functions	Description
SE_InitialiseCamera	Automatically detect camera and establish communication with the camera

SE_GetDetectorSize	Get the resolution of CCD detector
SE_SetCameraForAcq	Set the acquisition parameters such as exposure, binning, region of interest, gain, acquisition mode etc. Return the resulting image dimensions, valid exposure and frame rate for the acquisition parameters set.
SE_StartAcquisition	Start capturing images
SE_AbortAcquisition	Stop capturing images
SE_GetStatus	Get the acquisition status; whether the camera is idle or acquiring images
SE_GetSingleFrame	Retrieve a single image acquired ; the image is returned in a 16 bit integer buffer
SE_GetNumberNewImages	Get the number of images acquired since acquisition began
SE_GetImages	Retrieve the images acquired in a 16 bit integer buffer
SE_AcquireENVIData	Acquire and save images to disk in ENVI file format
SE_SaveAsENVI	Create the ENVI header file
SE_ShutDownCamera	Shut down the camera
SE_GetLastError	Get the last error message

*Table 5.1 Camera API functions*

Feature	Andor Luca (R)	Hamamatsu ORCA C8484 - 05G	Kappa DX 4 – 285 GigE	Xenics Xeva 1.7 – 320
Sensor	EMCCD - Frame transfer	CCD – Frame transfer	CCD	InGaAs FPA
Number of pixels (H*V)	1004 * 1002	1344 * 1024	1392 * 1040	320 * 256
Pixel size (H*V)	8 $\mu$ m * 8 $\mu$ m	6.45 $\mu$ m * 6.45 $\mu$ m	6.45 $\mu$ m * 6.45 $\mu$ m	-
Image area (H*V)	8mm * 8mm	8.67mm * 6.60mm	8.93mm * 6.66mm	-
Frame rate (fps) (full frame)	12.4	8.9	15	100
Spectral sensitivity	400 -1000 nm	400 – 1000 nm	320 – 1100 nm	900 – 1700 nm
Digitization rate	14 bit	12 bit	12 bit	12 bit
Communication interface	USB 2.0	IEEE 1394	Gigabit Ethernet (GigE)	USB 2.0

*Table 5.2 Camera specifications*

### ***5.1.1 IMAGE READOUT MODE***

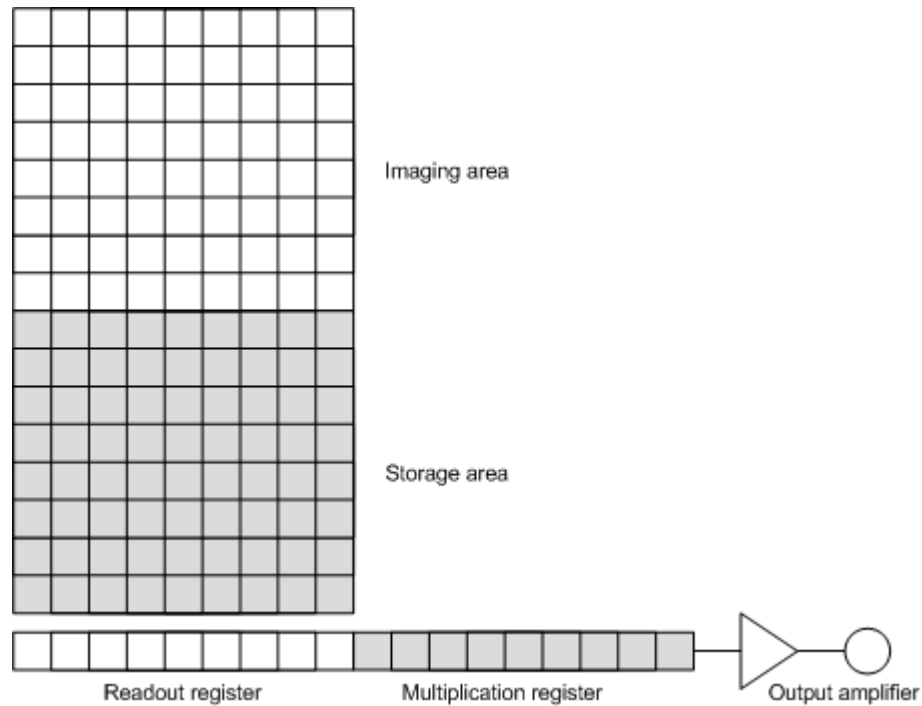
The Andor LUCA (R) camera consists of an Electron Multiplying Charge Coupled Device (EMCCD) frame transfer sensor shown in Figure 5.3. This sensor consists of a multiplication or gain register at the end of the normal read out register. The electrons that are transferred through this register get multiplied by the process of impact ionisation and generate secondary electrons. Thus the signal is amplified before being transferred to the output amplifier. As a result, the EMCCD technology facilitates

- Single photon detection sensitivity with full quantum efficiency which is useful especially under low illumination conditions
- High EM gain
- Low readout noise

The Andor LUCA and Hamamatsu ORCA 05G cameras consist of a frame transfer CCD. This type of CCD consists of two arrays, image array and storage array which are almost similar in size. The image array is exposed to light and the storage array is hidden from light i.e. it is covered with a layer of metal film. During exposure, the image array is exposed to light. The charge is then shifted quickly on to the storage array and is then read out. When the charge is read out from the storage array, the image array is exposed to light to capture the next image. The advantages of this CCD are

- No smearing of image
- Fast frame rates
- Less noise level
- No need of a mechanical shutter

These cameras can either be operated in normal CCD mode or in the frame transfer mode.



*Figure 5.3 Frame transfer EMCCD sensor*

## 5.2 CONFIGURING CAMERA PARAMETERS

Before starting an acquisition, the camera parameters such as exposure, binning etc. are set to the desired values. The camera control module is designed to provide access to these parameters. The function, *SE\_SetCameraForAcq* sets the camera parameters to the values specified by user. The camera parameters are explained in detail below,

### 12. Exposure

Exposure refers to the amount of time during which the CCD detector is exposed to the light. This can be changed to appropriate values depending on the illumination conditions. Each camera has its own permissible range of values for exposure. If the exposure is set to an invalid value, the camera automatically adjusts exposure to the nearest permissible value. This eliminates the need for the end user to have the technical knowledge to configure this parameter.



### 13. Binning

In a CCD camera, two or more pixels can be combined together horizontally or vertically before readout to form a super pixel. This process is called binning. By binning pixels together, the spatial and spectral resolutions of the image are reduced. The binning process has the following advantages,

- Better signal to noise ratio and sensitivity
- Reduced file size
- Increased read out speed

An example of 2 by 2 binning is shown in Figure 5.4.

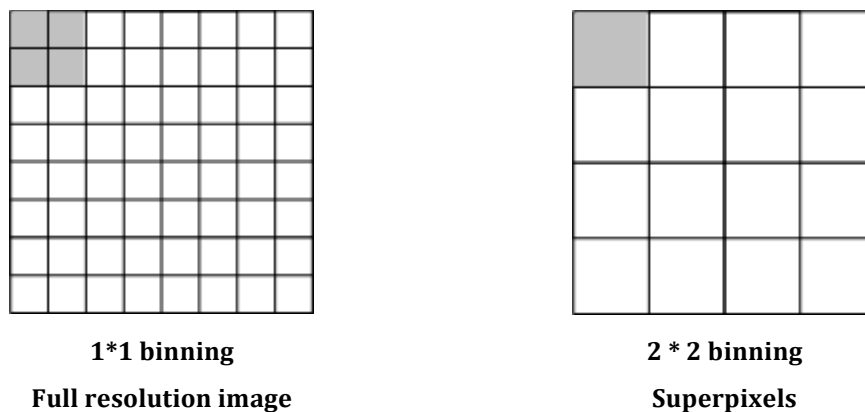


Figure 5.4 Binning

### 14. Region of Interest (ROI)

When reading out the image from the CCD sensor, a specific region on the sensor can be defined for the operation as shown in Figure 5.5. This feature allows the read out of desired spatial pixels and spectral pixels along the desired wavelength range. ROI has the following advantages

- Reduced file size

- Increased frame rate

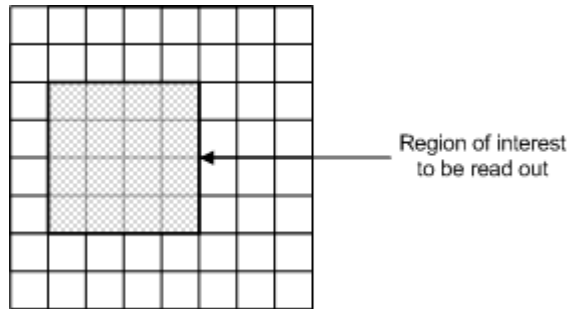


Figure 5.5 Region of interest on CCD to be read out

### 15. Read Mode

In the Andor LUCA camera, there are two modes in which the image can be read out from the CCD. They are as follows,

- Image: In image mode, the normal read out process which is explained in the previous section is carried out. The binning and ROI values for the image can be specified in this mode.
- Random Track: In random track mode, only the desired wavelength bands can be read out. In other words, this is a multi ROI mode represented in Figure 5.6. For example, the pixel rows corresponding to 400-500nm and 800nm can be read out.

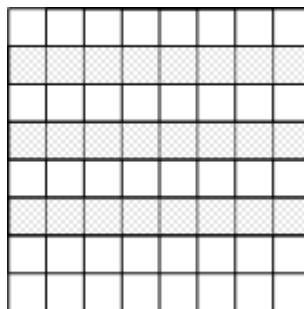


Figure 5.6 Random track mode

### 16. Acquisition mode

There are two modes in which the image can be acquired.

They include

- Snap: In this mode, the camera takes a single snapshot of the image. The frame display feature in spectraSENS uses the 'snap' acquisition mode.
- Continuous: In this mode, the camera captures images continuously until the acquisition is aborted. The waterfall display and live acquisition displays use the 'continuous' acquisition mode.

### **17. Gain**

Gain refers to the amount of amplification applied to the signal. The gain can be switched ON or OFF depending on the imaging conditions. Under low light imaging, a higher gain value may be required to obtain images of better quality. In the Andor LUCA camera, an additional EM gain can be applied to the signal.

### **18. Temperature**

In the Andor iXon, the temperature to which the sensor is cooled can be set. By cooling the sensor, the dark noise generated can be reduced.

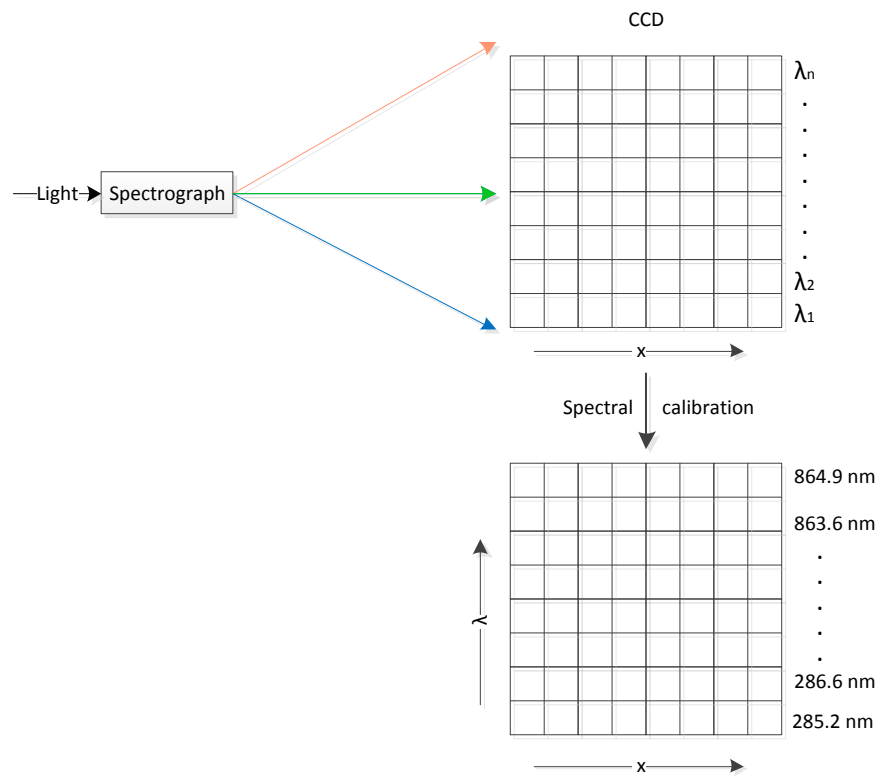
### **19. Frame rate**

Frame rate refers to the rate at which the frames are acquired from the camera. This value is specified in frames per second (fps). The frame rate depends on the exposure, binning, and ROI values that are set for acquisition. The function, *SE\_SetCameraForAcq* returns the frame rate for the acquisition parameters specified.

## **5.3 SPECTRAL CALIBRATION**

When the dispersed light from the spectrograph is projected onto the CCD,

each row of the detector corresponds to a particular wavelength band. Spectral calibration is the process of determining the wavelength values for individual channels in the detector array. This process is illustrated in Figure 5.7.

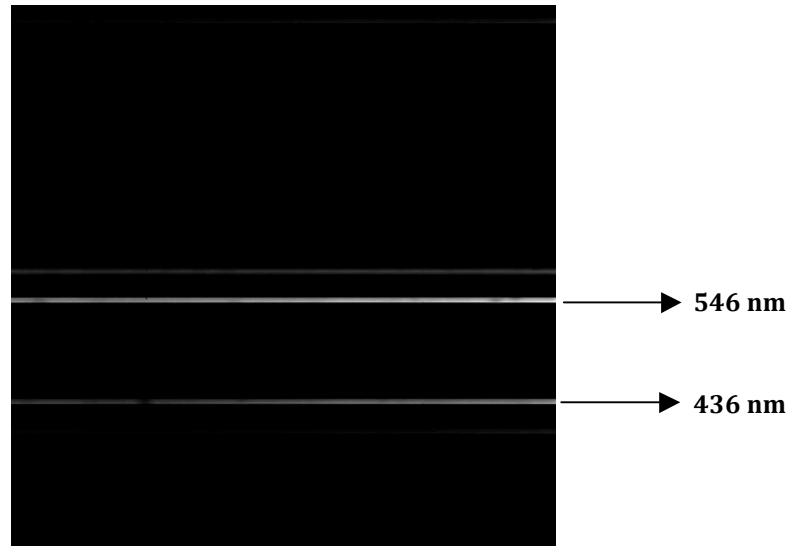


*Figure 5.7 Representation of Spectral calibration*

Spectral calibration is performed by using Mercury (Neon) – Hg (Ne) pencil type lamp. This lamp produces a series of narrow and intense lines at extremely well defined wavelength positions. This lamp when placed in front of the hyperspectral camera causes these spectral lines to appear as bright lines across the image plane. This is shown in Figure 5.8. With the pixel location and wavelength value corresponding to these lines, the wavelength value corresponding to every row of the detector array is determined by using the linear equation,

$$\lambda_p = mp + c \quad (5.1)$$

where  $\lambda_p$  represents the wavelength value corresponding to the pixel row number,  $p$  of the detector array. The values of  $m$  and  $c$  are calculated from the pixel location of the known spectral emission wavelengths of the mercury lamp such as 436 nm and 546 nm.



*Figure 5.8 Mercury lines at 436nm and 546nm in a frame*

The wavelength values calculated for every row of the detector are stored in a file called 'Calibration.txt'. A screenshot of the calibration file is shown in Figure 5.9. The camera control module uses these values while configuring the acquisition parameters such as ROI, read mode etc. For example, the function `SE_SetCameraForAcq` receives the desired wavelength range of interest (400nm – 500nm) and converts the wavelength to corresponding pixel numbers (245 – 356) and at the same time configures the ROI to be used by the detector.

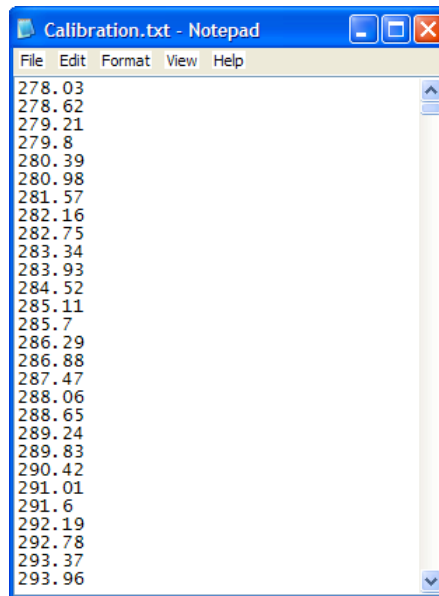


Figure 5.9 Screenshot of the calibration file

#### 5.4 CONFIGURING THE DATA ACQUISITION PROCESS

In the camera control module, certain functions provide control over the data acquisition process. These functions are used by spectraSENS to start and abort the acquisition. Once the acquisition is started, the camera acquires data depending on the configured acquisition mode.

In snap mode, once the acquisition is started, the camera captures a single image and stores it in an internal memory. The *SE\_GetStatus* function is configured to monitor the status of the acquisition process. This function returns the information on whether the camera is idle or capturing data. The *SE\_GetSingleFrame* function retrieves the acquired image from the memory. This function returns the pixel values of the image in a buffer which are then processed and displayed.

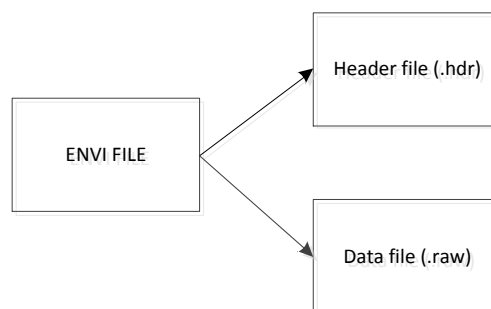
In continuous mode, once the acquisition is started, the camera continues to acquire images until the acquisition is aborted. The acquired images are stored temporarily in an internal circular buffer. Once the maximum memory limit of the buffer is reached, these images will be overwritten by new

images unless they have been retrieved. The functions *SE\_GetNumberNewImages* and *SE\_GetImages* are responsible for retrieving these images from the memory during the acquisition process. By using these functions, all the acquired images can be obtained without any loss of frames.

## 5.5 SAVING THE HYPERSPECTRAL DATA AS ENVI FILE

The camera control module exposes functions which will acquire and save hyperspectral data in ENVI format. ENVI is a software application which is designed to provide comprehensive post-processing analysis of hyperspectral data. This software, marketed by ITT Visual Information Solutions, provides advanced image processing and analysis tools.

An ENVI file format is a combination of two files: a flat binary file (.raw) and an ASCII header file (.hdr). The ENVI file format is represented in Figure 5.10. The binary file consists of the pixel values of the hyperspectral images. The header file contains information about how the hyperspectral data is stored in the raw file.



*Figure 5.10 Representation of ENVI file format*

### 5.5.1 ENVI HEADER FORMAT

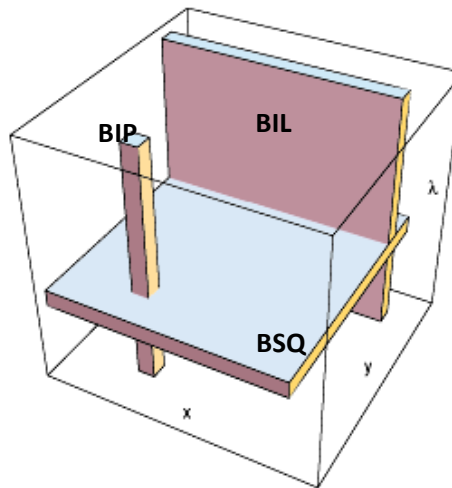
ENVI software uses the information stored in the ENVI header file to read the binary image file which contains the hyperspectral data. The header file

contains keywords which specify essential information about the raw file. The description of main keywords in an ENVI header file is provided in Table 5.3. The header file should start with the text 'ENVI' to indicate it as a native file header. A screenshot of an ENVI header file is shown in Figure 5.12.

Field	Description
Samples	Number of pixels (columns) along spatial dimension per image line
Bands	Number of pixels (rows) along spectral dimension per image line
Lines	Number of images (lines on the target) stored in the binary file
File type	File type defined by ENVI for certain data format. This can include ENVI specific files such as metafiles, spectral libraries etc. or data-specific formats such as TIFF, BMP etc.
Data type	Data representation of a pixel, e.g. 1=8 bit byte, 2=16 bit signed integer, 3=32 bit signed integer, 4=32 bit floating point
Interleave	The sequence in which images are stored in the file, BIL, BSQ or BIP format. BIL – Band Interleaved by Line; First line of first band followed by first line of second band and so on BSQ – Band sequential ; each line of data is followed by the next line in the same band BIP – Band Interleaved by Pixel; first pixel of all bands in sequential order followed by the second pixel for all bands and so on. The data format is represented in Figure 5.11
Wavelength units	String indicating units for the wavelength values stored in the file
Binning	Horizontal and vertical binning values used for image acquisition
Wavelength	List of wavelength values corresponding to each band in an image

**Table 5.3** ENVI header format





*Figure 5.11 Representation of the data storage formats, BIL, BIP and BSQ. Image courtesy- <http://www.optics.arizona.edu/detlab/>*

### 5.5.2 ENVI - A PROGRAMMER'S PERSPECTIVE

The camera control module provides the following two functions which are configured to save the hyperspectral data in ENVI format.

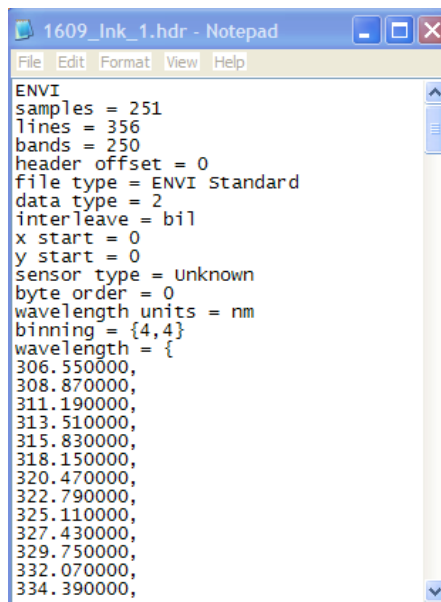
4. SE\_AcquireENVIData
5. SE\_SaveAsENVI

The function, *SE\_AcquireENVIData* acquires the images in continuous mode. It creates a binary file with the file name and directory path specified by the user and saves the images to the binary file until the acquisition is aborted. The function, *SE\_SaveAsENVI* gets the information about the hyperspectral data saved in the binary file and creates the ASCII header file. The values for keywords stored in the header file are provided in Table 5.4.

Field	Description
Samples	The X dimension of the image. This is calculated from the X resolution of the camera, and acquisition parameters such as horizontal binning and ROI.
Bands	The Y dimension of the image. This is calculated from the Y resolution of the camera considering the acquisition parameters such as vertical binning and ROI.

Lines	Number of images saved to the binary file
File type	This value is set to 'ENVI standard' as the data is stored in standard ENVI format
Data type	The pixel values are stored as 16 bit signed integers. So this field is set to value '2'.
Interleave	In the pushbroom technique, one image corresponds to spectral image of one spatial line on the target. The binary file contains the spectral information of one line followed by the spectral information of second line and so on. So this field corresponds to the BIL file format.
Wavelength units	The wavelength values are stored in nanometre units. This value is set to 'nm'
Binning	The horizontal and vertical binning values used for image acquisition
Wavelength	The wavelength values are obtained from the spectral camera calibration file, 'Calibration.txt'. The wavelength values and range are calculated from the binning and ROI parameters used for acquisition.

*Table 5.4 ENVI header – programmer's perspective*



```

1609_Ink_1.hdr - Notepad
File Edit Format View Help
ENVI
samples = 251
lines = 356
bands = 250
header offset = 0
file type = ENVI standard
data type = 2
interleave = bil
x start = 0
y start = 0
sensor type = Unknown
byte order = 0
wavelength units = nm
binning = {4,4}
wavelength = {
306.550000,
308.870000,
311.190000,
313.510000,
315.830000,
318.150000,
320.470000,
322.790000,
325.110000,
327.430000,
329.750000,
332.070000,
334.390000,

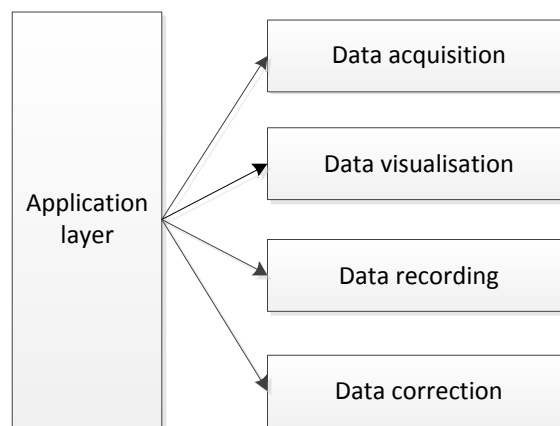
```

*Figure 5.12 Screenshot of the ENVI header file*

## 6. APPLICATION DEVELOPMENT

The Application development chapter discusses in detail the design and implementation of various components associated with spectraSENS. This chapter brings out the effort of how two different research domains, namely, software development and hyperspectral imaging have been interlaced together to form the spectraSENS system.

The application layer of spectraSENS coordinates the scanner control, hyperspectral data acquisition, data display and data recording functionalities. This layer acts as an interface between the Graphical User Interface (GUI) and the two hardware modules namely camera control and scanner control [Figure 3.8].



*Figure 6.1 Functionalities of the Application layer*

The application interface has been designed in such a manner that it provides various functions such as data acquisition, data correction, and qualitative and quantitative analysis in real-time. The functionalities of the application layer is represented in Figure 6.1.

This chapter discusses the following topics in detail,

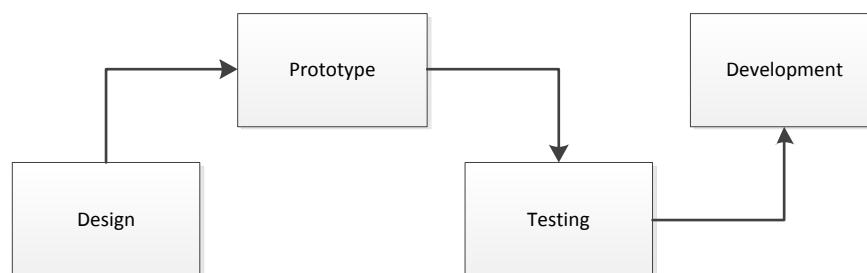
1. Graphical user interface design,
2. Visualisation techniques,
3. Data recording techniques,
4. Data correction techniques.

## 6.1 GRAPHICAL USER INTERFACE DESIGN

The GUI defines the human and computer interaction. In 1977, Xerox STAR was the world's first GUI designed by researchers at Xerox Palo Alto Research Centre. The success or failure of a product in the market place often depends on the GUI reception rather than the actual functionalities.

The usability of software is reflected by its GUI design. Lots of importance was given to the design of the GUI for spectraSENS. A good GUI design makes the application more likable and easy to use. The development cycle of the GUI consists of four steps, illustrated in Figure 6.2.

1. Design
2. Prototype
3. Testing
4. Development



*Figure 6.2 Stages in GUI development*

### **6.1.1 DESIGN**

A good design of the GUI results in reduced training and support time. The design focussed on the requirements and expectations of the end users. The main goals involved in the design of the GUI for spectraSENS are

#### **1. Simplicity**

spectraSENS is designed for end-users with no specialist knowledge in imaging and computing. They range from farmers, fruit sorters to academic researchers. As such, the user interface has been kept as simple and as easy to use. The GUI clearly shows the available functionalities of the software to the user by grouping the relevant functions.

#### **2. Efficiency**

The GUI provides optimum controls to all the functionalities of the software. The number of user inputs has been kept minimal. This allows the users to perform minimum configurations during the time of installation or configuration. Hence they are able to run the software with minimal action.

#### **3. Feedback**

The GUI provides a prompt response for every action of the user e.g. clicking a button, entering a value etc., enhancing the human computer communication. The GUI provides live status updates for user's action. For example, if an error occurs during a scanning task, an error message is presented to the user indicating the likely reason of what has gone wrong.

#### **4. Adaptability**

A global vision has been adopted during the design of the GUI. A GUI that has been created at one geographical location may have to be

adapted for different geographical locations. Such a strategy helps minimise the future development costs and increases the possibility to explore new markets with less investment. The GUI is adaptable for different user's needs such as differences in terms of computer monitor resolutions, operating systems and user languages.

#### 5. **User friendly**

All the dialogues and controls in the GUI are self-descriptive.

#### **6.1.2 PROTOTYPE**

The word prototype is derived from the Greek word, "prototypon" meaning primitive form. Development of a prototype is the second stage involved in the development cycle of a GUI. The functionalities of the software were reviewed before implementing the prototype.

The Gestalt Principle is utilised to design the prototype. This principle is not exploited in the visual sense but in a functional sense. Similar functionalities are grouped together within tabs. The GUI for spectraSENS consists of four sections

1. Menu bar
2. Tool bar
3. Control panel
4. Console

The control panel in turn is divided into five different sections. Each of these sections is represented as a separate 'Tab control'. These sections include

1. Camera
2. Scanner
3. Hypercube
4. Corrections

## 5. Information

Prototyping resulted in the alpha version of spectraSENS.

### 6.1.3 TESTING

Usability testing was performed with the help from volunteers. The parameters, which we used in testing, have already been described in 6.1.1. Feedback obtained from the volunteers was fed into the future development process.

### 6.1.4 DEVELOPMENT

The GUI was developed using Microsoft Visual Studio C#. A screenshot of the GUI is presented in Figure 6.3.

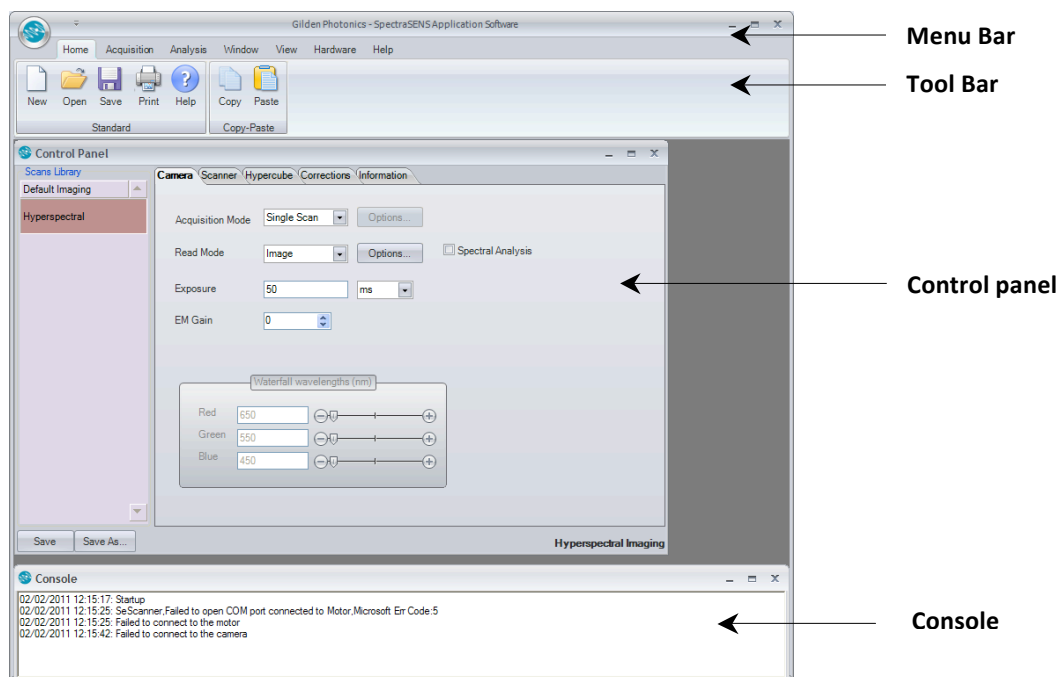


Figure 6.3 spectraSENS – Screenshot of Graphical User Interface

## **6.2 DATA VISUALISATION TECHNIQUES**

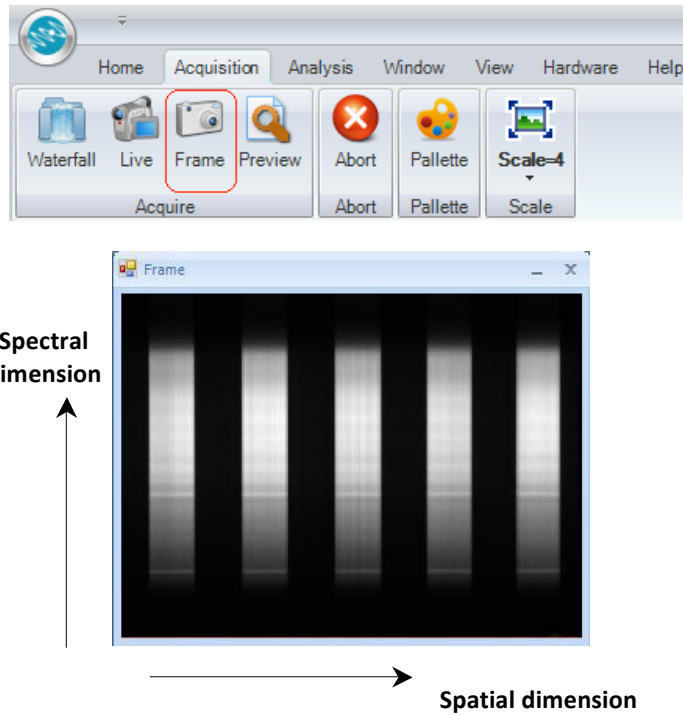
Data visualisation refers to visually presenting the acquired data for human interpretation. The goal of data visualisation is to provide the users with a better understanding and insight into the data. The visualisation of data to users is essential to make informed decisions and interpretations. The visualisation of the hyperspectral data cube allows users to analyse the data in real time and to determine the optimum acquisition parameters such as exposure, scan parameters etc. Hyperspectral images contain far more spectral or colour information than a normal colour image. Therefore, visualising this multi-dimensional image on a standard colour monitor presents many challenges.

The visualisation techniques adopted in spectraSENS are designed to provide useful and interpretable information to the users. These techniques allow the users to extract significant information quickly and easily. These techniques are capable of fast rendering and are therefore compatible with real-time interactivity. The display techniques adopted in spectraSENS are discussed below.

### **6.2.1 FRAME DISPLAY**

Frame display presents a single frame captured by the hyperspectral camera. A frame refers to hyperspectral image of one scan line of the object. In a frame, each row of pixels corresponds to one particular wavelength band. The captured frame is displayed on to the display window in the GUI. In frame display, the horizontal axis represents the spatial dimension and vertical axis represents the spectral dimension. Figure 6.4 shows a screenshot of the frame display.





*Figure 6.4 Screenshot of the frame display*

The user pre-configures the camera parameters before acquiring the frame. The pixels of the acquired frame are stored in a buffer. These values can range from 8 bit to 32 bit each depending on the digitisation rate of the camera used. To display these pixels in a colour monitor, the pixels are scaled using a pre-defined display look-up-table (LUT). The LUT, also called palette is used to determine the colour format (Red-Green-Blue values) for a given intensity value with which the image is displayed. A screenshot of a greyscale palette is shown in Figure 6.5.

A list of predefined palettes is used for image display. These palettes contain 256 values of colours ranging from 0 - 255. The pixel values in the image buffer are scaled between 0 and 255. These scaled values are then converted to colour values using the palette and then are displayed on the frame display window. The formula used to determine the colour of a pixel in the image buffer using a palette is provided in equation 6.1.

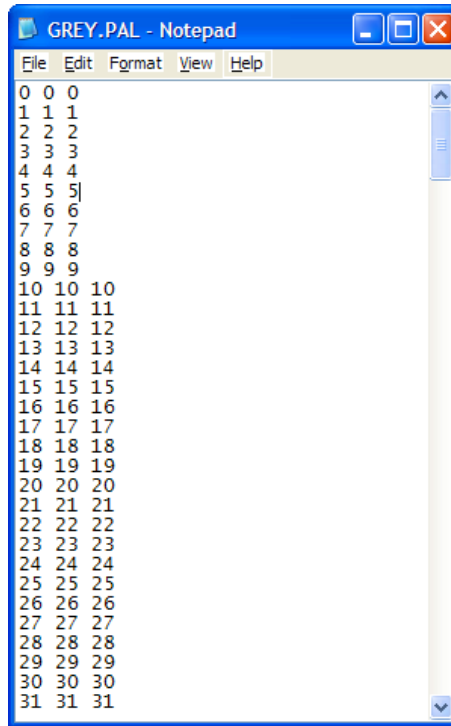


Figure 6.5 Screenshot of the greyscale palette file

For a given pixel  $P(i)$  in the image buffer,

$$P_s(i) = \left[ \frac{255}{(P_{\max} - P_{\min})} \right] * [P(i) - P_{\min}] \quad (6.1)$$

where,

$P_s(i)$  : Scaled 8 bit pixel value

$P_{\max}$  : Maximum pixel value in an image buffer

$P_{\min}$  : Minimum pixel value in an image buffer

Now, the three values found in the row number  $P_s(i)$  in the palette file are assigned as red, green and blue values for the pixel  $P(i)$ .

$$\{ R(i), G(i), B(i) \} = \{ R(i, P_s(i)), G(i, P_s(i)), B(i, P_s(i)) \} \quad (6.2)$$

where,

$R(i), G(i), B(i)$  : Red, green and blue values calculated for pixel  $P(i)$  respectively

- $R(i, P_s(i))$  : First value found in row number  $P_s(i)$  in the palette file
- $G(i, P_s(i))$  : Second value found in row number  $P_s(i)$  in the palette file
- $B(i, P_s(i))$  : Third value found in row number  $P_s(i)$  in the palette file

The schematic diagram of the frame display is provided in Figure 6.6.

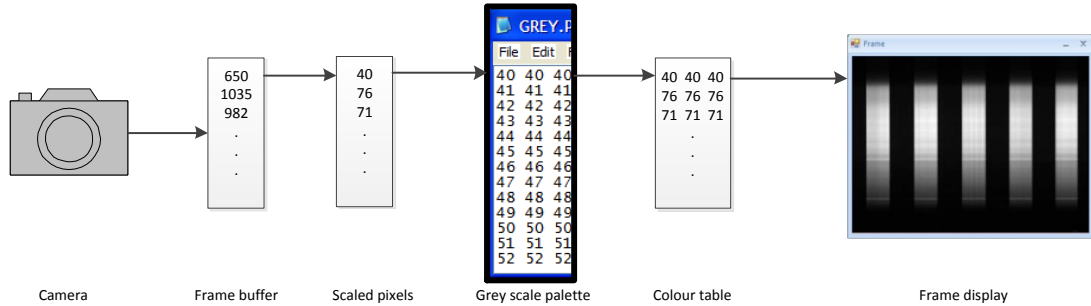


Figure 6.6 Schematic diagram – Frame display

In the frame display, the acquired data is scaled using a greyscale palette as default. The palette for the frame display can be chosen according to the user’s requirement. By clicking the palette button in the toolbar, the display will be updated with a series of optional LUTs such as red, green, blue and false colour modes.

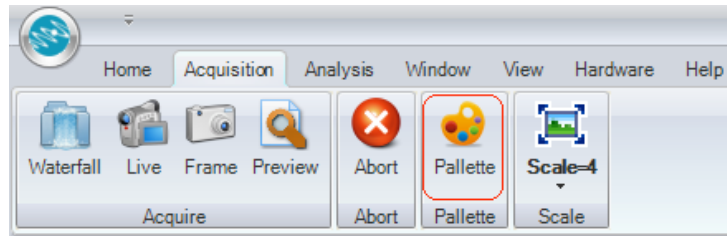


Figure 6.7 Palette control on the GUI

### 6.2.2 LIVE ACQUISITION DISPLAY

The live acquisition display presents the acquired frames as a live video. When the live acquisition display is triggered, the acquisition parameters such as exposure, binning etc. are set in the associated hardware by the camera and scanner control modules. Once the parameters are set, the frames

are captured from the camera with the object being scanned simultaneously. Figure 6.8 provides a screenshot of the live acquisition display.

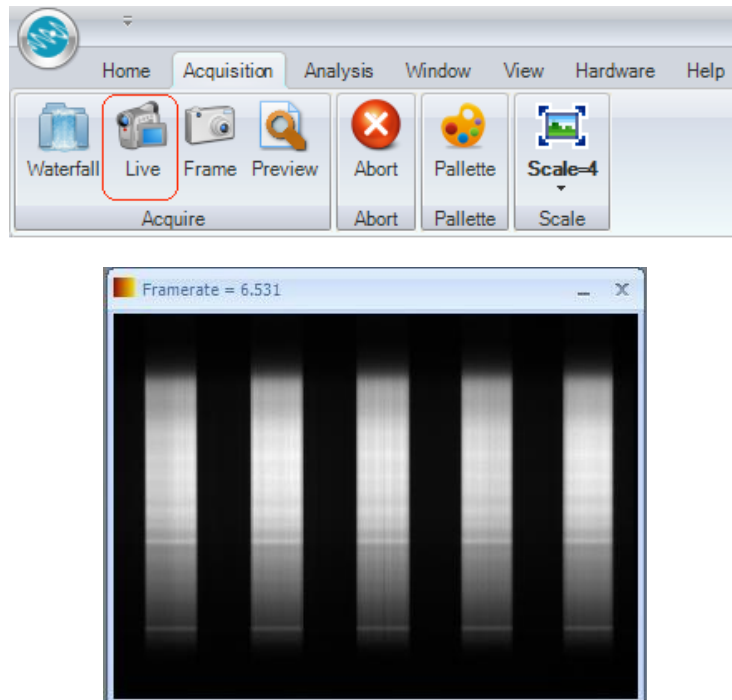


Figure 6.8 Screenshot of live acquisition display

This display function has been programmed in a similar manner to the frame display except that the screen is constantly updated whenever a new frame is available. The acquired frames are scaled using a greyscale palette by default and displayed. Clicking the 'Palette' button on the GUI can change the display palette. The schematic diagram of the live acquisition display is provided in Figure 6.9.

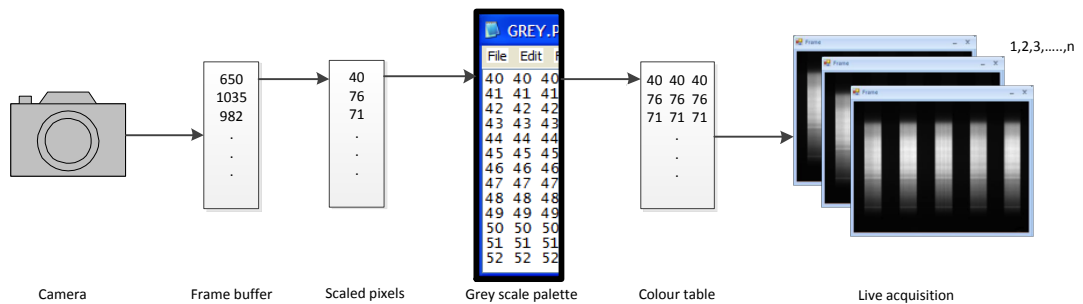


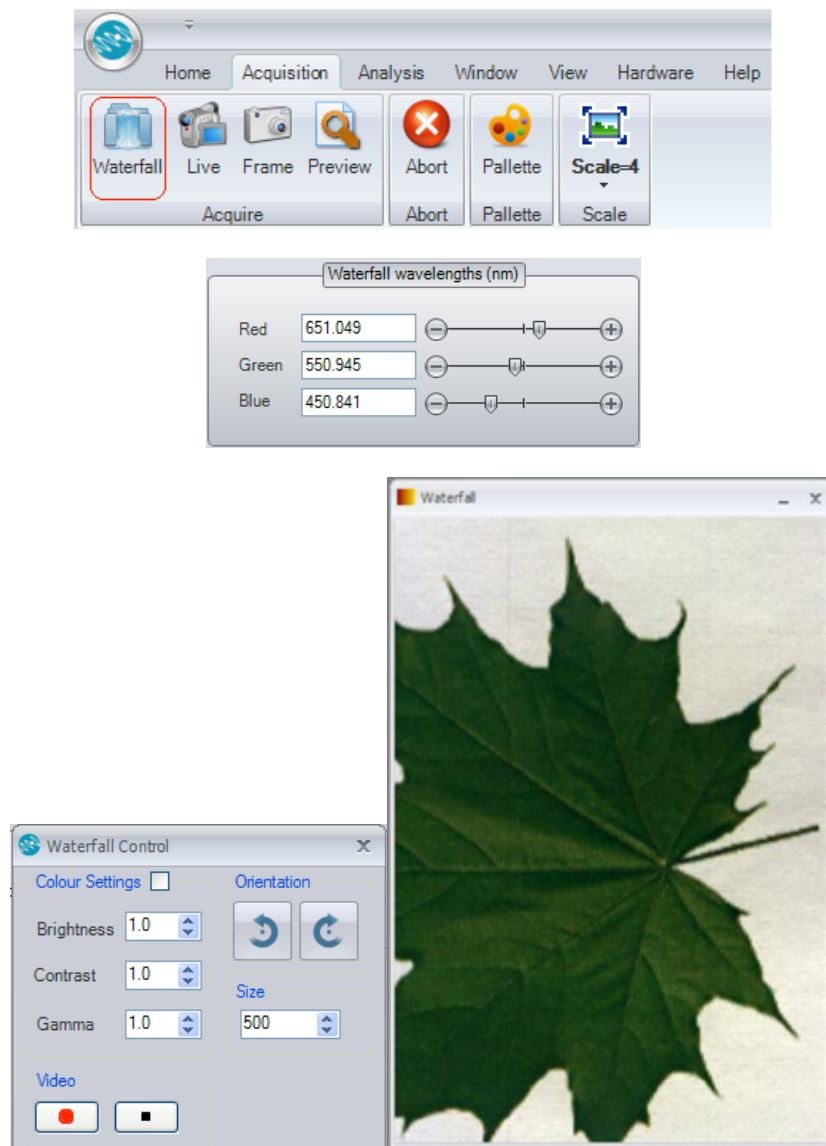
Figure 6.9 Schematic diagram of the live acquisition display

The live acquisition display allows the user to view the frames acquired in real time. This display is useful for aligning and focussing the hyperspectral camera.

### **6.2.3 WATERFALL DISPLAY**

The waterfall display presents the hyperspectral data as a false colour image with the spatial dimension built over time, as the scene is scanned. Assigning three user specified wavelength bands for Red, Green and Blue (RGB) colour components creates the false colour image of the scene. Different combination of the wavelength bands and colour channels can highlight various spectral features in the image during this display. The waterfall display features a continuously updated scrolling video of the hyperspectral data in real-time. This display resembles the flow of a waterfall and hence the name. A screenshot of the waterfall display, waterfall control on GUI are provided in Figure 6.10.

The user pre-configures the camera and scanner parameters. When the waterfall display is triggered, the acquisition parameters such as exposure, binning etc. are set in the associated hardware by the camera and scanner control modules. The frames are captured from the camera when the object is scanned simultaneously. The pixels of the acquired frame can be 8 bit to 32 bits each depending upon the camera type. Three rows of pixels are selected from each frame depending on the red, green, blue wavelength values specified by the user. These rows are mapped from the wavelength values using the calibration file [Section 5.3]. These pixel values are scaled between 0 and 255 using the formula (6.3). The scaled values are then assigned for red, green and blue colour values for displaying the image.



*Figure 6.10 Screenshot of waterfall display*

For a given pixel  $P(i)$  in the image buffer,

$$P_s(i) = \left[ \frac{255}{(P_{\max} - P_{\min})} \right] * [P(i) - P_{\min}] \quad (6.3)$$

where,

$P_s(i)$  : Scaled 8 bit pixel value

$P_{\max}$  : Maximum pixel value in an image buffer

$P_{\min}$  : Minimum pixel value in an image buffer

The red, green, blue values for each pixel in the waterfall display are represented as below,

$$\{ R(i), G(i), B(i) \} = \{ P_s(\lambda_R), P_s(\lambda_G), P_s(\lambda_B) \} \quad (6.4)$$

where,

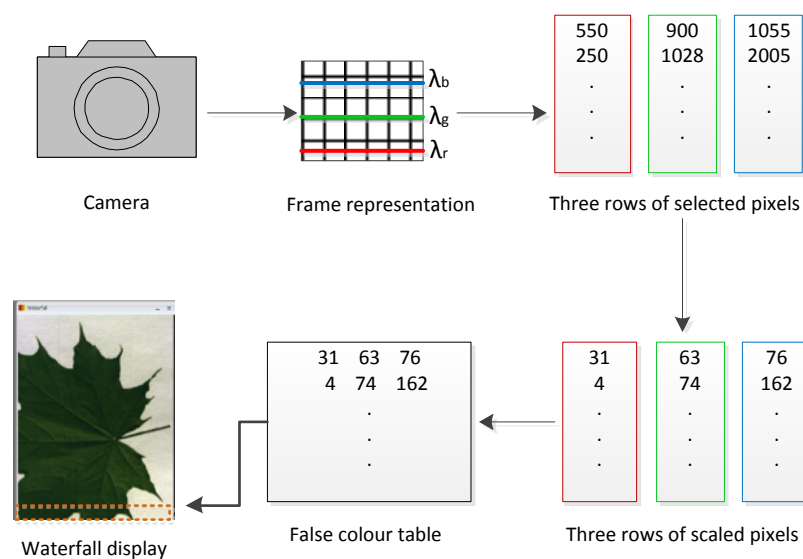
$R(i), G(i), B(i)$  : Red, green and blue values for a pixel in the waterfall display

$P_s(\lambda_R)$  : Scaled pixel value corresponding to the wavelength band assigned to the red colour

$P_s(\lambda_G)$  : Scaled pixel value corresponding to the wavelength band assigned to the green colour

$P_s(\lambda_B)$  : Scaled pixel value corresponding to the wavelength band assigned to the blue colour

Thus three rows of a frame are processed to create a false colour image of a single line for display. As more frames are captured, the waterfall display builds and then is constantly updated line by line and scrolls down the window. The schematic diagram of the working of waterfall display is given in Figure 6.11.



*Figure 6.11 Schematic diagram – waterfall display*

The main advantages of the waterfall display technique are

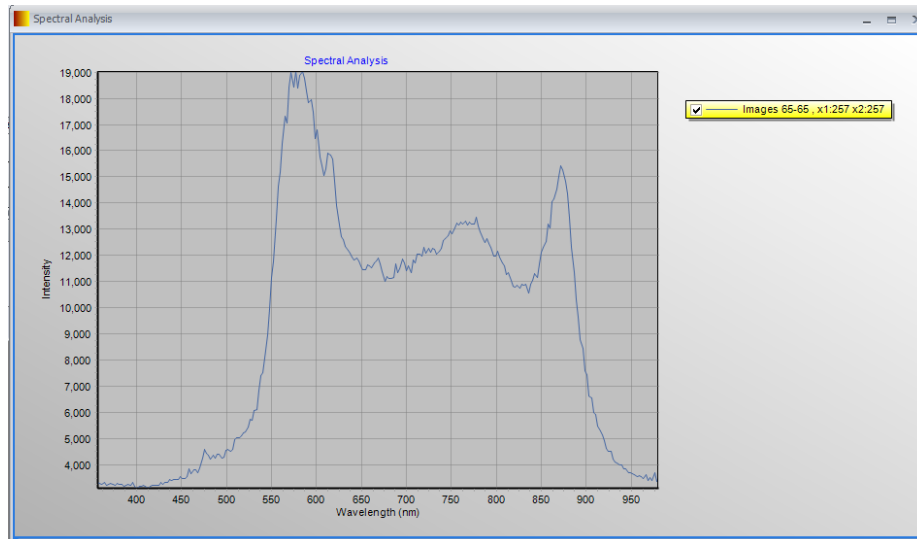
1. Only three rows of pixels in a frame (three wavelength bands) are processed to generate the false colour image. This reduces the amount of data handled by the processor and hence saves the processing power.
2. A fast image rendering is achieved due to the reduced time for data handling.
3. The user can select any choice of colour - wavelength combinations for displaying the image for effective visualisation.

Uniquely the waterfall display can be

1. Rotated and resized
2. Adjusted for brightness, contrast and gamma
3. Saved as a video file

The spectral profile of a single pixel or an average of pixels can be obtained from the waterfall display. This is performed by pausing the display and then clicking on any point or drawing an area on the display window. This feature allows the user to analyse the spectral profiles of the features of interest in real time. A screenshot of spectral profile display is provided in Figure 6.12.



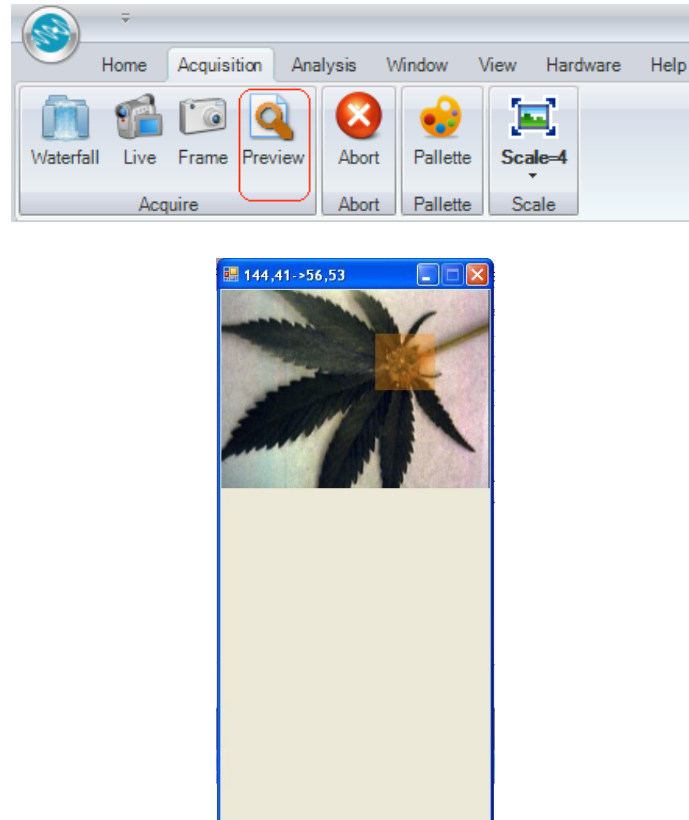


*Figure 6.12 Screenshot of spectral profile*

#### **6.2.4 PREVIEW DISPLAY**

The preview display provides a quick preview of the scene that is being scanned. This allows the user to select a particular region of interest (ROI) on the preview and perform actual scanning only on the ROI from the next scans. The display technique used for the preview display is similar to that of the waterfall display except that in the preview display, the wavelength values of images used to create false colour is predefined. These predefined values are set to 650nm, 550nm and 450nm for red, green and blue colours respectively. These values have been found to provide a good false colour picture of the scene when using a VIS or VNIR system.

The user can use any input device (mouse, pen etc) to draw a ROI on the preview display window. Depending on the area selected, the sub image parameters of the camera, scan distance and zero offset values are automatically calculated. From the next scan onwards, the ROI selected within the image alone will be acquired. This feature allows users to easily configure the camera and scan parameters to capture the desired region of the image. A screenshot of the preview display is provided in Figure 6.13.



*Figure 6.13 Preview display*

### **6.3 DATA RECORDING**

Acquiring and storing hyperspectral data cube to memory storage for post processing is called data recording. The hypercube is saved as a raw binary data in ENVI compatible format [Section 5.5]. The camera control module provides functionalities to acquire and store data in the ENVI format [Section 5.5]. The application layer uses functions from the scanner and camera interfaces to provide data recording. Synchronisation has to be achieved between the data acquisition from the camera and the motion of the scanner in order to generate the hypercube of a target object. This process has been automated, requiring no configuration from the end user. Thus spectraSENS provides a fully automated control on data recording.

A screenshot of the hypercube tab control on the GUI which provides the data recording controls is shown in Figure 6.14.

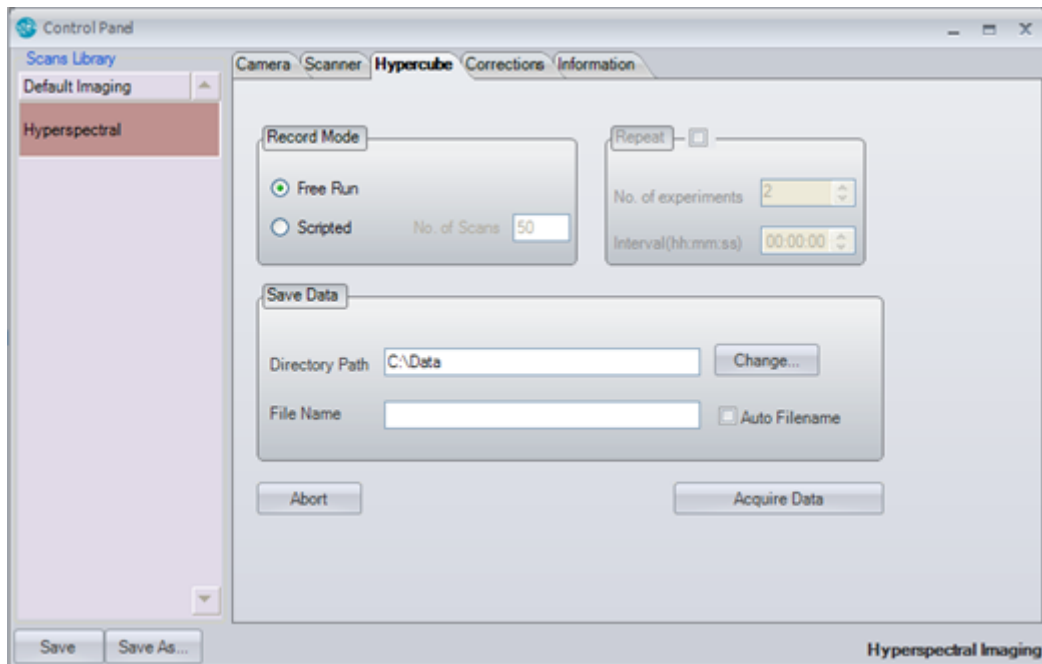


Figure 6.14 Screenshot of Hypercube tab control

Three data recording modes have been implemented. They are as follows,

### 1. Free run mode

In free run mode, the hyperspectral data is recorded continuously until aborted manually. As the scanner moves forward and backward, the data captured from each scan is recorded as a separate file. Two individual threads in real time handle acquisition and storing of data.

### 2. Scripted mode

In scripted mode, the user needs to pre-configure the number of scans required. This mode records the number of scans specified by the user and automatically aborts the acquisition and scanning process once completed.

### 3. Scripted repeat mode

In scripted repeat mode, the user will pre-configure the number of scans

required and the interval between two consecutive scans. This mode allows automatic recording of data without the need for the presence of the user during the actual scanning process. This feature is particularly useful for applications where temporal changes have to be measured. For example, during the measurement of dissolution of drugs in a solvent, this option can be set to record data every few minutes or so.

#### **6.4 DATA CORRECTION**

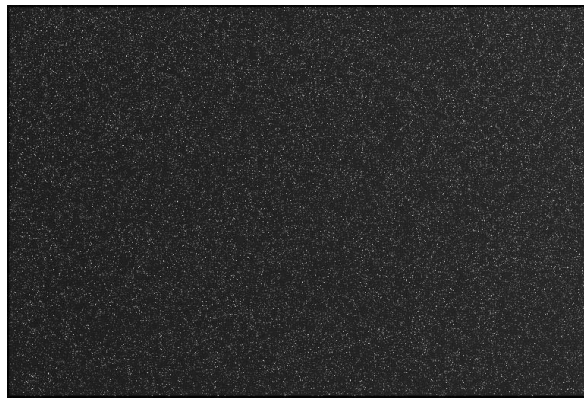
While measuring the hyperspectral data, there is heterogeneity in various factors such as detector response, illumination and optics. These factors have to be compensated in order to obtain a uniformly normalised data. This process is called spectral correction. Spectral correction involves performing pixel-by-pixel reflectance calibration for all wavelengths.

The major factor affecting a detector's response is dark current. Dark current is one of the main sources of noise in image detectors. Dark current refers to the small leakage current that flows in a CCD even in the absence of light. This results in a pixel offset in the image captured. This offset varies from pixel-to-pixel and depends on the temperature of the detector. This dark noise can be removed by 'dark frame subtraction' i.e. by measuring a dark frame and subtracting it from the actual measured frame. A screenshot of a dark frame is shown in Figure 6.15. The dark response can be determined by capturing a frame with either the,

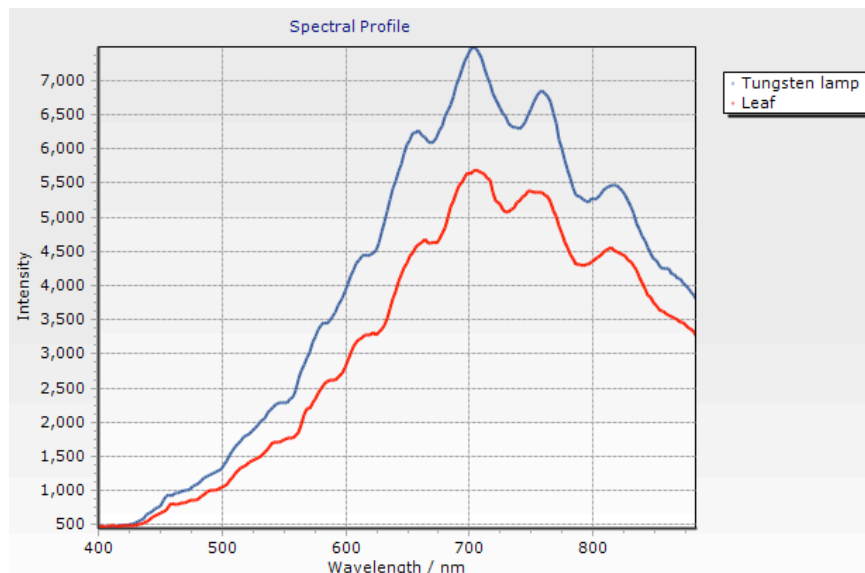
- Lens completely covered with its cap, or
- Shutter of the camera completely closed.

The second major factor affecting the hyperspectral image measurement is the spectral response of the source illumination. Hyperspectral sensors record the amount of light reflected by the target object (spectral radiance) at

different wavelength bands. The spectral radiance is influenced by external factors such as spectral variations in the source illumination, atmospheric absorption effects, optics etc. These factors distort the actual or “true” spectral features of the target object. An example of how the spectral features of a leaf are influenced by that of the source illumination is shown in Figure 6.16. Converting the measured spectral radiance into spectral reflectance compensates for these factors. Spectral reflectance is the ratio of intensity of reflected light to the intensity of incident light on target at each wavelength band.

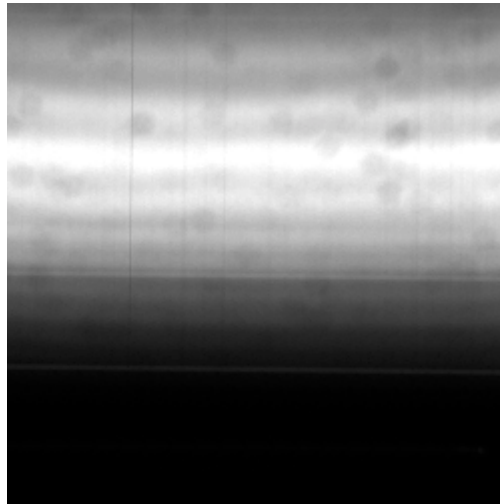


*Figure 6.15 Screenshot of a dark frame*



*Figure 6.16 Raw measurement of the spectral features of a leaf along with the measured white reference for the illumination source*

The incident light can be measured directly or by measuring a reference material whose reflectance is known: “the white reference”. The most common reference material is a certified gradient Spectralon reflection panel. The spectralon panel provides 99% reflectance. A screenshot of a white frame is shown in Figure 6.17.



*Figure 6.17 Screenshot of a white frame*

The effective spectral reflectance of a pixel at a given wavelength is calculated using the following equation

$$\text{Spectral Reflectance} = (\text{Reflected pixel} - \text{Dark pixel}) / (\text{White pixel} - \text{Dark pixel}) \quad (6.5)$$

This can be represented as,

$$R(\lambda) = \left[ \frac{I_R(\lambda) - I_D(\lambda)}{(I_{100}(\lambda) - I_D(\lambda))} \right] \quad (6.6)$$

where,

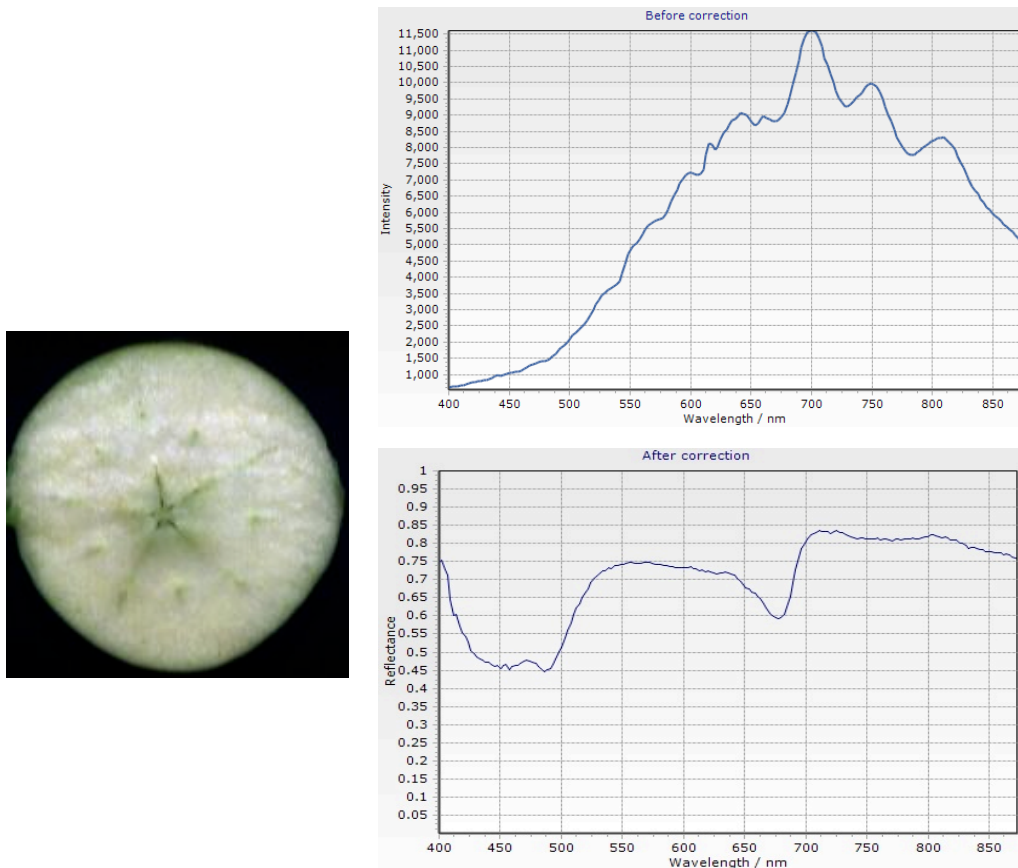
$R(\lambda)$  – Spectral reflectance

$I_R(\lambda)$  – Measured image

$I_D(\lambda)$  – Dark frame

$I_{100}(\lambda)$  – White reference

An example showing the spectral profiles of an apple before and after corrections is shown in Figure 6.18.



*Figure 6.18 Spectral profile of an apple – Before and after corrections*

The screenshot of the correction tab in spectraSENS is shown in Figure 6.19. The dark reference and white reference are calculated by averaging over 500 frames. These reference data are saved in raw data format. The measurements of dark reference and white reference must be performed with the identical exposure and acquisition settings used for the hypercube acquisition.

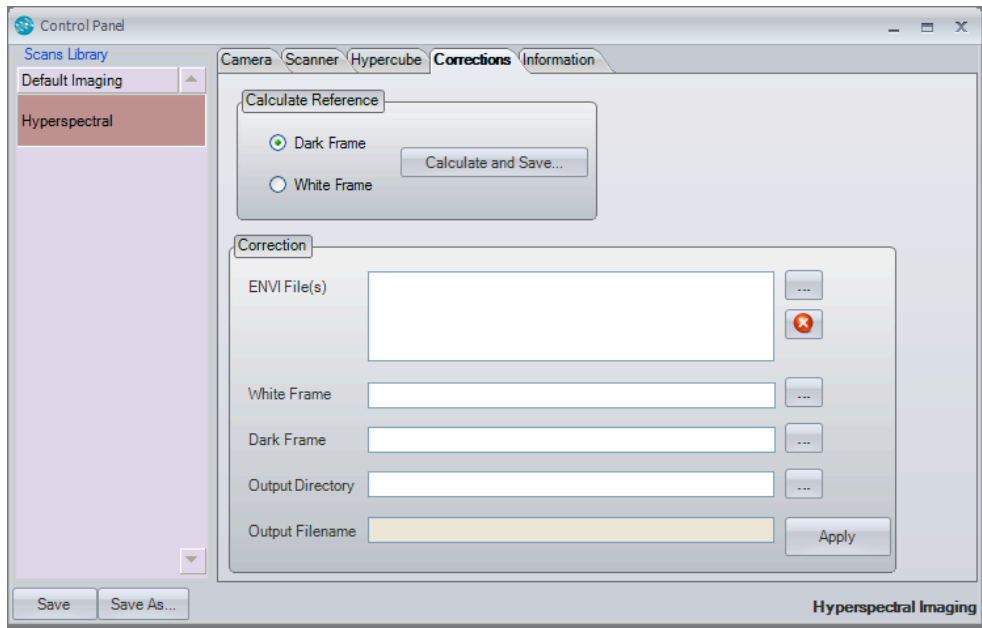


Figure 6.19 Screenshot of the corrections tab control



## **7. APPLICATIONS OF spectraSENS**

The spectraSENS system has been used in a range of applications from different fields for achieving many objectives. In this chapter, a detailed summary of three specific applications has been discussed. This includes food, agriculture and forensic sciences.

### **7.1 MEASUREMENT PROCEDURE**

This section discusses the measurement procedure adopted for the three applications of spectraSENS mentioned in this chapter. As a first step in the measurement process, the spectrograph in the Hyperspectral Imaging (HSI) system was aligned with respect to the horizontal and vertical axes of the CCD. The spectral camera was then focussed with the help of a test target. The test target is a black and white checker pattern located at the target distance. When focusing is correct, the black squares appear clear and sharp in the image. Spectral calibration was performed using a GPL LHM254 Mercury spectral calibration light and the calibration file, 'Calibration.txt' was generated. The whole configuration was performed under the guidance of the live acquisition display.

The target sample was placed on a ceramic plate, which was then mounted onto the platform of the linear scanner system. One sample was measured at a time. The exposure time was set depending on the illumination conditions. The criterion to achieve this is by inspecting the signal-to-noise ratio. The signal level is adjusted to be about 80% of the maximum value to ensure that signal saturation does not happen. The signal-to-noise ratio was then calculated from the spectral profile extracted from the waterfall display.

The horizontal and vertical binning parameters were set to desired values. The Region of interest (ROI) of the acquisition was set based on the physical size and shape of each target sample. The ROI, scan distance, zero offset distance values were configured under the guidance of the preview display. spectraSENS automatically calculates the optimum scan speed to form square pixels.

Once the acquisition and scan parameters were set, the data was then recorded to disk in ENVI format. The sample measurement time was typically a few seconds per scan.

A dark reference image was captured with the lens cap closed. A white reference image was captured with a calibration tile. A spectral correction was then performed on the saved data to convert the camera signal counts into reflectance values.

## **7.2 APPLICATION I – QUALITY ASSESSMENT OF MEAT**

### **Objective:**

The objective of this application was to discriminate the fat and meat and bone areas in three different meat samples, which include lamb, pork and beef using HSI. This experiment is carried out in the Near Infra red (NIR) region of the spectrum. This is because various chemical compounds show absorption bands in the NIR region of the spectrum and therefore is widely used for food analysis [8].

### **System specifications:**

A Gilden Photonics GPL-N17E-30-XV60-USB HSI system was used. This system consists of a NIR spectral camera module and a linear scanner system. The system specifications are provided in Table 7.1.

Spectral range	900 – 1700 nm
Spatial resolution (pixels/line)	320
Spectral bands	256
Spectral resolution	3.2 nm
Pixel size on sample	30 $\mu$ m
Illumination	Tungsten lamp – 150 W
Typical scan time	5 to 10 seconds

*Table 7.1 GPL-N17E-30-XV60-USB HSI system specifications*

**Acquisition parameters:**

The typical acquisition parameters configured for the hyperspectral data acquisition is provided in Table 7.2.

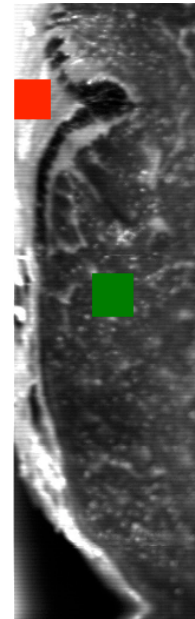
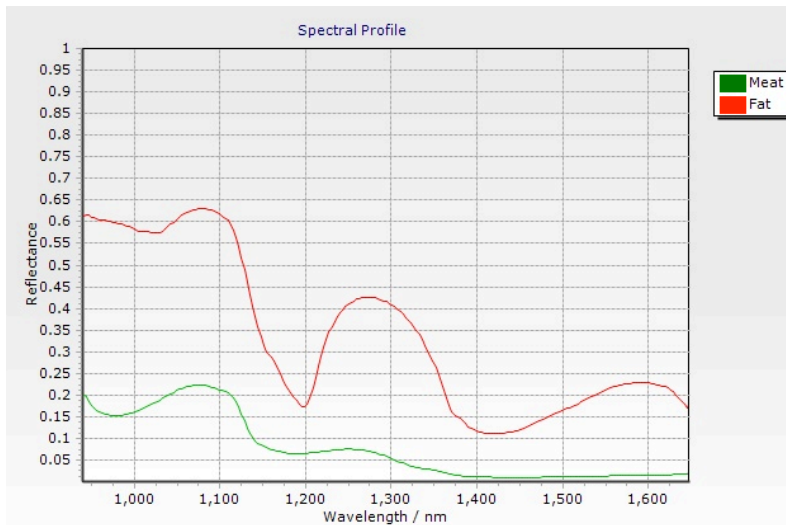
Exposure	2 ms
Binning (Spatial * Spectral)	1 * 1
Image dimensions (Spatial * Spectral)	320 * 256
Frame rate	80 Hz
Scanning range	80 to 100 mm
Scanning rate	6 mm/s
Measurement time	15 s

*Table 7.2 Typical acquisition parameters*

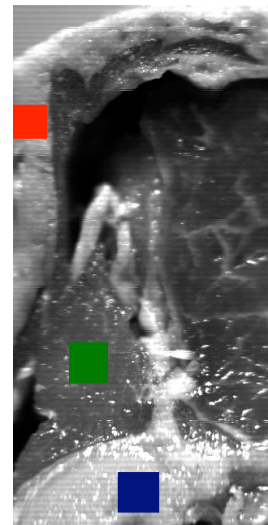
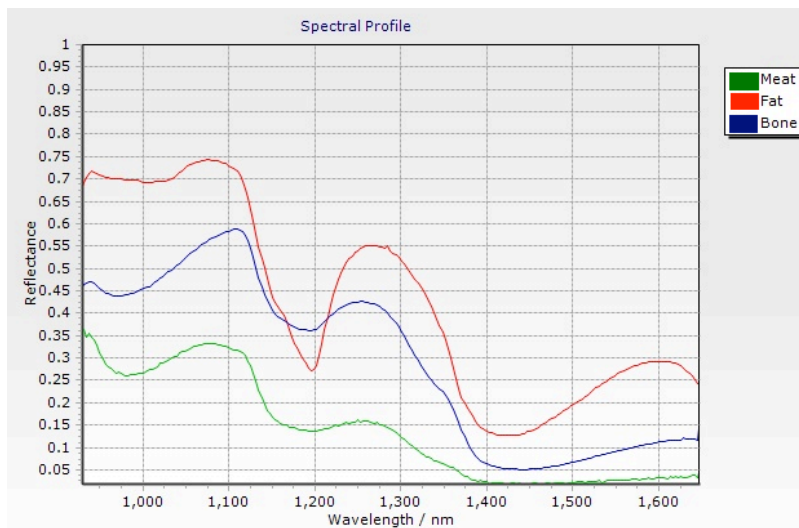
**Results and discussion:**

A square region of size 5 \* 5 pixels was chosen on different regions in each meat sample. Based on the subjective analysis of the meat, we had a prior knowledge of the locations of fat, meat and bone. All the three meat samples contained fat and meat, but bone region was located only in lamb. The mean reflectance spectra over each area in the square sample region was calculated and compared. The original spectra lie between 900 to 1700 nm. Due to low signal-to-noise ratio, the spectra below 940nm and above than 1670nm were excluded from the analysis.

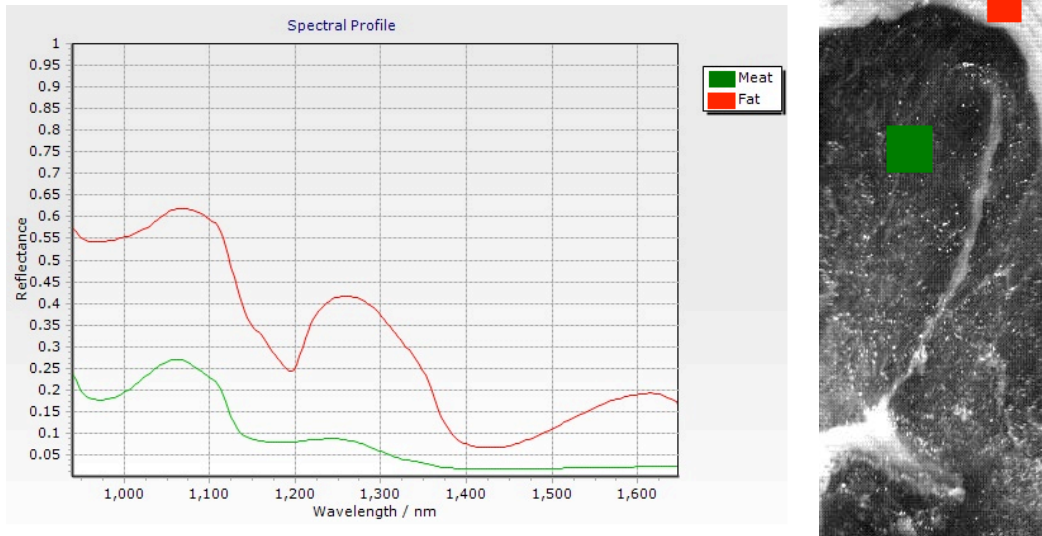
The spectral profiles of fat, meat and bone regions show significant variations. This is illustrated in figures 7.1, 7.2 and 7.3.



*Figure 7.1 Mean reflectance spectra of fat and meat in the beef sample. The red plot represents fat and green plot represents meat*

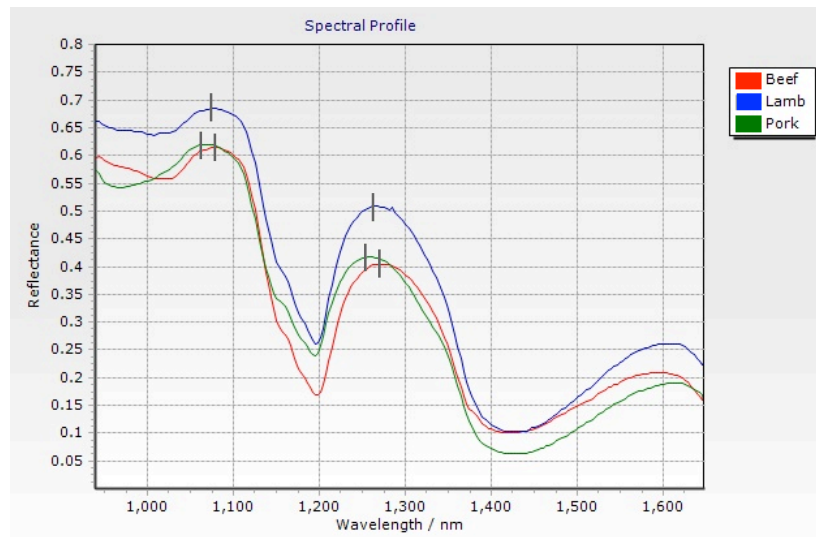


*Figure 7.2 Mean reflectance spectra of bone, fat and meat in the lamb sample. The red plot represents fat, green represents meat and blue represents the bone region*

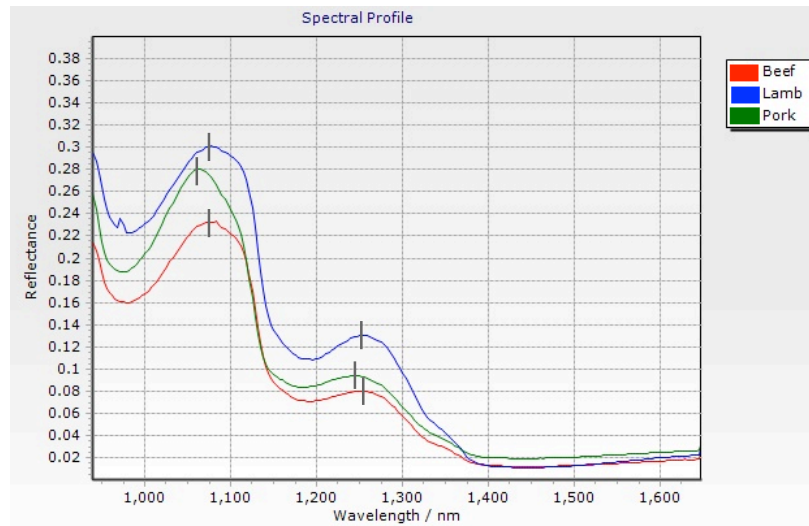


*Figure 7.3 Mean reflectance spectra of fat and meat in the pork sample. The red plot represents the fat and green plot represents meat*

From the figures 7.1, 7.2 and 7.3, it was observed that the fat, meat and bone regions in beef, lamb and pork showed unique spectral signatures. The reflectance spectra of fat and meat were compared across the three samples in the figures 7.4 and 7.5.



*Figure 7.4 Mean reflectance spectra of fat in beef, lamb and pork samples.*



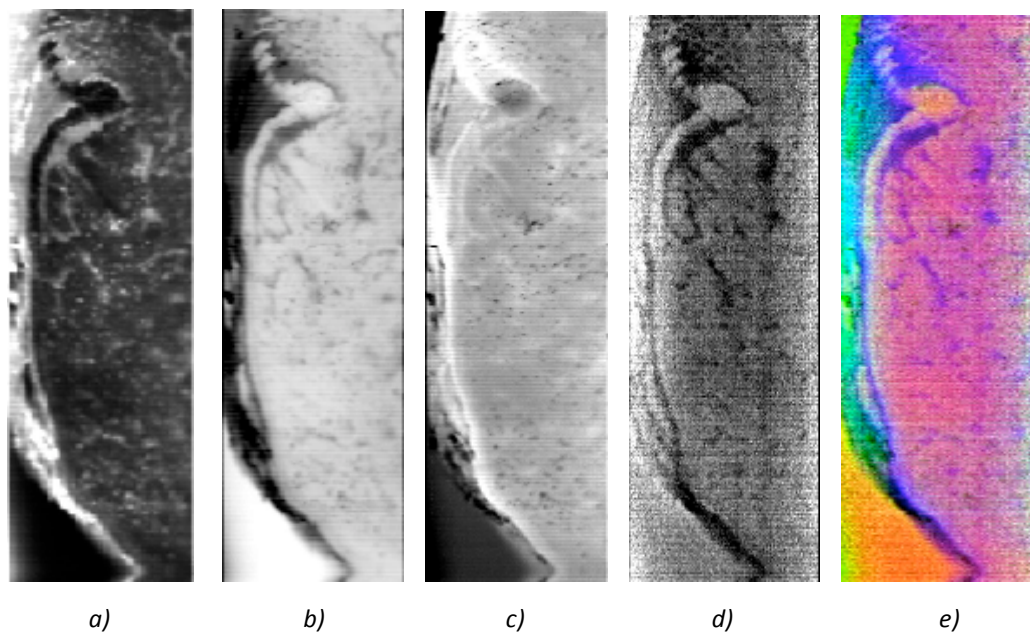
*Figure 7.5 Mean reflectance spectra of meat in beef, lamb and pork samples.*

From Figure 7.4, it was found that the spectral profiles of fat in the lamb, pork and beef samples showed similar pattern. The same was observed with meat spectra in the three meat samples [Figure 7.5]. These spectral profiles showed differences in the amount of reflectance. The reflectance of fat and meat in lamb was greater than that of beef and pork. The locations of the spectral peaks of each sample were different to one another. This property could be used for meat classification. Use of appropriate pattern classification techniques and tools will help to make a quantitative analysis of the data.

In Figure 7.4, the spectral profile of fat shows two high reflectance peaks at 1100nm and 1270nm. A high absorption is observed at 1200nm. This is related to the second overtone of the C-H bond Infrared (IR) spectrum [10]. In figures 7.2, 7.4 and 7.5, the fat, meat and bone regions show absorbance bands at 974nm and 1400nm, which are related to water absorption bands caused by O-H stretching in the second and first overtones respectively. To automate our objective for online process monitoring, a pattern discriminator can be built based on the above observed properties.

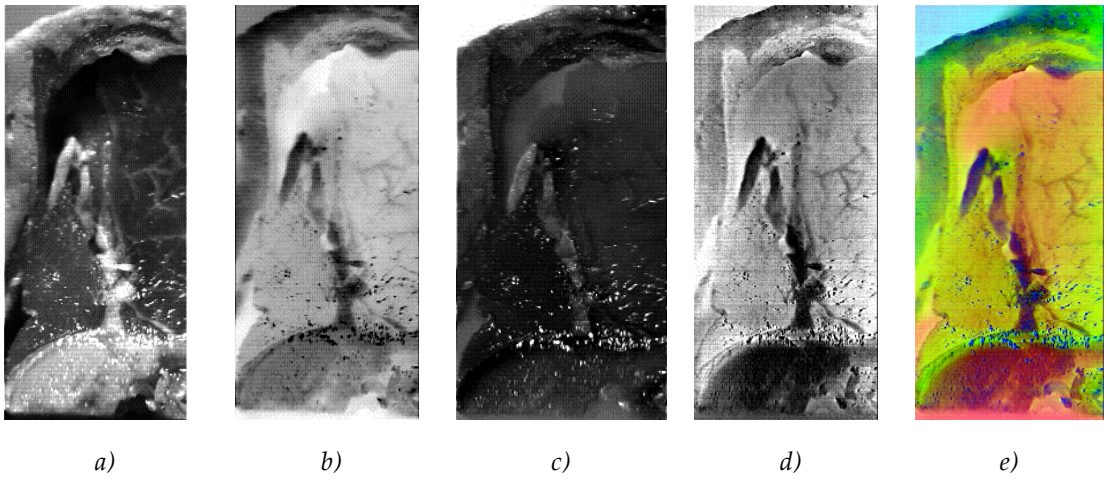
A dimensionality reduction technique, Principal Component analysis (PCA) was applied on the datasets of the three meat samples. This technique was

used to remove any redundant information in the hyperspectral data. PCA is a mathematical technique, which is used to identify patterns in a dataset. This technique converts N number of wavelength bands in a hyperspectral data into M principal components. The first principal component (PCA1) represents the maximum variability in the dataset and the rest of the components account for the remaining variability in this data. The images obtained by PCA are shown in figures 7.6, 7.7 and 7.8. The first three principal components accounted for the maximum spectral variation in the data. These images show the presence and distribution of various features in the samples, and how they are separated in the image. The colour composite image created by using images of PCA1, PCA2 and PCA3, showed significant contrast differences among fat, meat and bone areas. In order to make a quantitative identification, we need to prepare the correct model for the analysis.

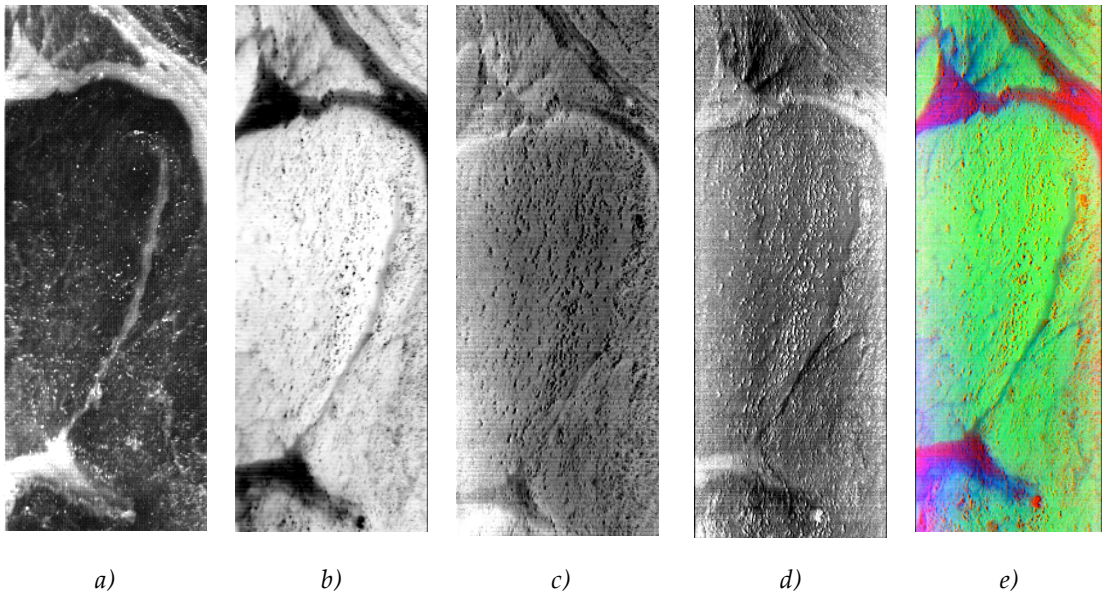


*Figure 7.6 Beef sample : a)Image at 1200nm, b)PCA 1, c)PCA 2, d)PCA 3, e)PCA Colour composite*





*Figure 7.7 Lamb sample : a)Image at 1200nm, b)PCA 1, c)PCA 2, d)PCA 3, e)PCA Colour composite*



*Figure 7.8 Pork sample : a)Image at 1200nm, b)PCA 1, c)PCA 2, d)PCA 3, e)PCA Colour composite*



### 7.3 APPLICATION II - MONITORING OF VEGETATION SAMPLES

#### Objective:

The objective of this study was to use hyperspectral imaging to,

1. Identify brown rust, a fungal disease in wheat leaf,
2. Identify yellow rust, a fungal disease in barley leaf,
3. Differentiate rye grass seeds from mustard seeds.

#### System specifications:

A Gilden Photonics GPL-V8E-30-LRUSB hyperspectral imaging (HSI) system, which consists of a visible range spectral camera module and a linear scanner system was used. The system specifications are provided in Table 7.3.

Spectral range	400 – 880 nm
Spatial resolution (pixels/line)	1004
Spectral bands	1002
Spectral resolution	1.4 nm
Pixel size on sample	8 $\mu$ m
Illumination	Tungsten lamp – 150 W
Typical scan time	5 to 10 seconds

*Table 7.3 GPL-V8E-30-LRUSB HSI system specifications*

#### Acquisition parameters:

The typical acquisition parameters configured for the hyperspectral data acquisition is provided in Table 7.4.

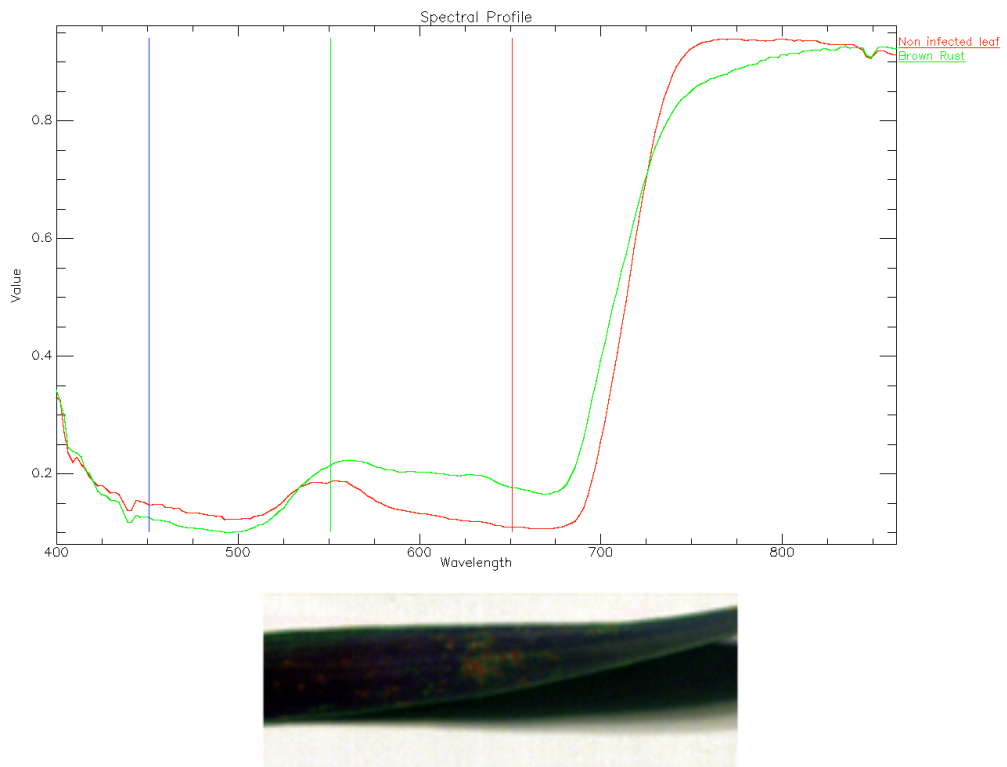
Exposure	50 ms
Frame rate	20 Hz
Binning (Spatial * Spectral)	4 * 4
Image dimensions (Spatial * Spectral)	251 * 250
Scanning range	30 to 50 mm
Scanning rate	6 mm/s
Measurement time	7 s

*Table 7.4 Typical acquisition parameters*

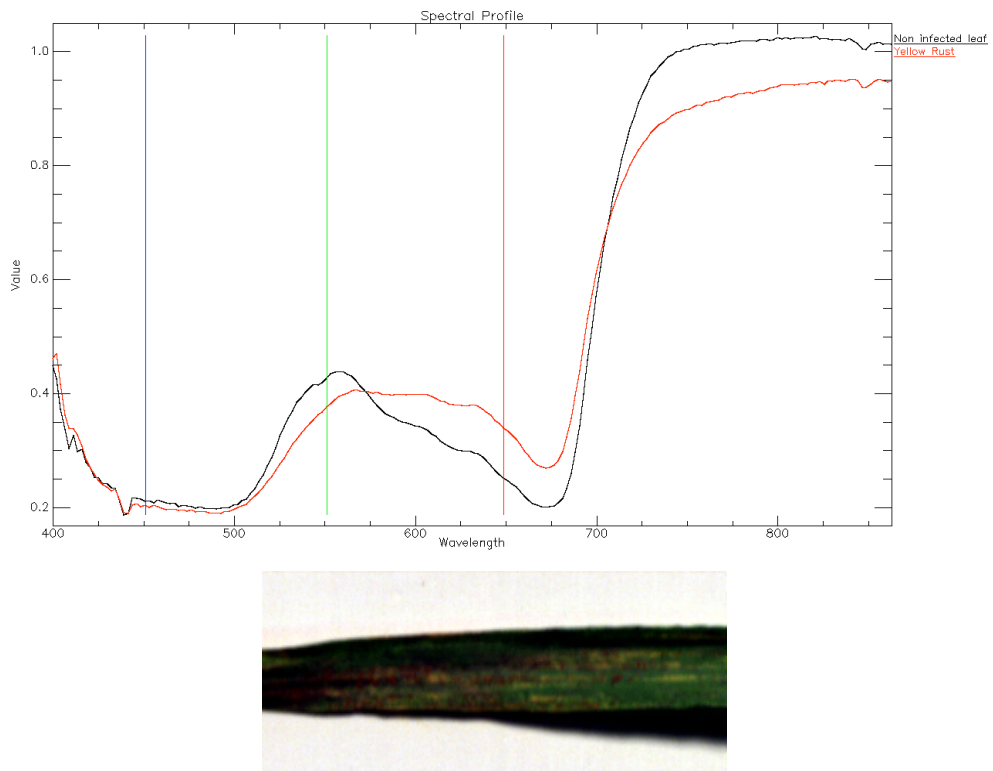
### **Results and discussion:**

The reflectance spectrum of brown rust on wheat leaf was compared to that of non-infected part of the leaf. These spectral profiles showed different patterns. Similar results were observed with yellow rust in barley leaf. This is illustrated in figures 7.9 and 7.10.

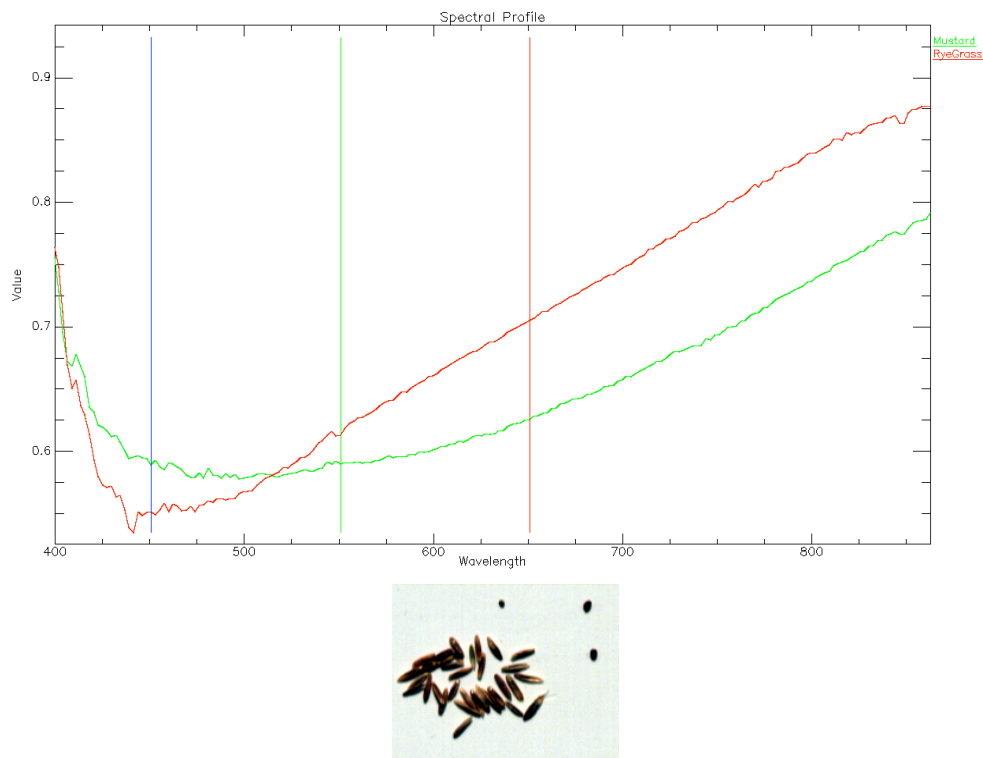
By inspecting the spectra of wheat and barley leaves shown in figures 7.9 and 7.10, it was observed that the reflectance of these leaves are lower in visible region (400-700 nm) compared to that of VNIR region (700 – 880nm). This is due to higher absorption of light energy by chlorophyll and other pigments such as carotenoids in the visible region. In contrast, the reflectance is higher in VNIR region due to lower absorption by pigments and higher scattering of light by plant tissues [11]. The reflectance of the diseased (brown/yellow rust) leaf is higher than that of the non-infected wheat/barley leaf in the bands 560nm – 670nm and lower in bands 730nm – 880nm. Plants with high chlorophyll content show high reflectance in green band (510 - 550nm) and high absorption in red band (~650nm) [12]. From figures 7.9 and 7.10, it was found that the absorption depth in red band and reflectance peak in green band are reduced in diseased (yellow/brown rust) leaf compared to that of the non-infected leaf. This indicates reduced chlorophyll content in the diseased leaf.



**Figure 7.9** Reflectance spectra of brown rust on wheat leaf (green plot) compared to that of non infected part of leaf (red plot)



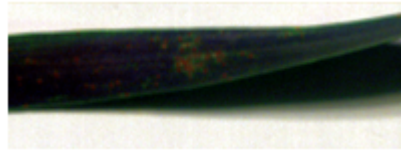
**Figure 7.10** Reflectance spectra of yellow rust on barley leaf (red plot) compared to that of non infected part of leaf (blue plot)



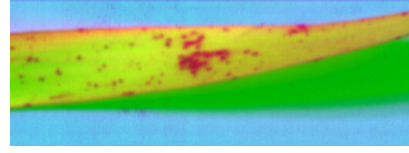
*Figure 7.11 Reflectance spectra of Ryegrass seeds (red plot) and mustard seeds (green plot)*

The reflectance spectra of ryegrass seeds and mustard seeds showed difference in their spectral properties. This is shown in Figure 7.11. It was observed that the ryegrass seeds showed higher absorption in wavelength region 400 nm – 500 nm compared to that of mustard seeds. The reflectance of the ryegrass seeds was relatively higher in bands 520 nm – 880 nm.

PCA was carried out on the individual hyperspectral datasets corresponding to wheat leaf, barley leaf and the seed samples using the ENVI software. A composite colour image was then created with images of three principal components. These images are shown in figures 7.12, 7.13 and 7.14. The PCA colour composite image showed the greatest contrast difference between the brown rust and the wheat leaf than shown in a normal colour image. The same was observed with the yellow rust on the barley leaf. The PCA colour composite image also highlighted mustard seeds from ryegrass seeds.



a)

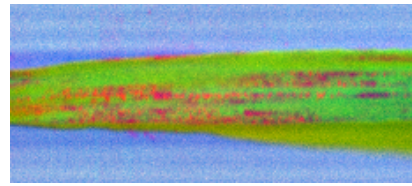


b)

*Figure 7.12 Wheat leaf infected with brown rust : a) Normal RGB image b) PCA colour composite image shows the locations of brown rust with higher contrast*



a)

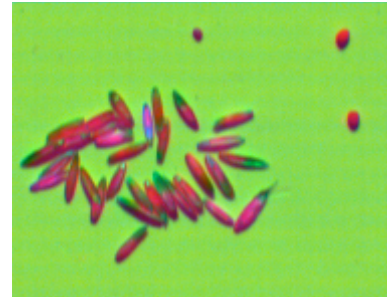


b)

*Figure 7.13 Barley leaf infected with yellow rust: a) Normal RGB image b) PCA colour composite image shows the locations of yellow rust with higher contrast*



a)



b)

*Figure 7.14 Ryegrass and mustard seeds : a) Normal RGB image b) PCA colour composite image highlights the difference in colour between ryegrass and mustard seeds*

## 7.4 APPLICATION III – COUNTERFEIT DETECTION AND INK ANALYSIS IN DOCUMENTS

The objective of this study was to use hyperspectral imaging to,

1. Identify alterations in a forged cheque,
2. Identify ink features in banknotes.

### System specifications:

A Gilден Photonics GPL-V8E-30-LRUSB hyperspectral imaging (HSI) system, which employs a visible spectral camera module and a linear scanner system was used. The system specifications are provided in Table 7.5.

Spectral range	400 – 880 nm
Spatial resolution (pixels/line)	1004
Spectral bands	1002
Spectral resolution	1.4 nm
Pixel size on sample	8 $\mu$ m
Illumination	Tungsten lamp – 150 W
Typical scan time	5 to 10 seconds

*Table 7.5 GPL-V8E-30-LRUSB HSI system specifications*

### Acquisition parameters:

The typical acquisition parameters configured for the hyperspectral data acquisition is provided in Table 7.6.

Exposure	50 ms
Frame rate	20 Hz
Binning (Spatial * Spectral)	4 * 4
Image Dimensions (Spatial * Spectral)	251 * 250
Scanning range	60 to 80 mm
Scanning rate	5 mm/s
Measurement time	14 s

*Table 7.6 Typical acquisition parameters*

### Results and discussion:

The images of the forged cheque and the banknote at different wavelength bands were analysed. Figure 7.15 shows the normal colour image of the forged cheque. In this image, the alterations made in the writings could not be identified by visual inspection. The false colour image of the cheque created with images at three different wavelength bands is shown in Figure 7.16. This image reveals the alteration made in the amount indicated on the cheque. In the false colour image, it can be clearly seen that the number '1' in the amount '\$10000' has been altered to number '4'. In this image, the alterations are shown in different colour compared to the original writing. This is because the alterations were made with a different type of black ink compared to the one used in original writing. Thus the black ink pair samples which look similar in the normal image was discriminated using HSI. The difference in reflectance spectra of the inks is presented in Figure 7.17.

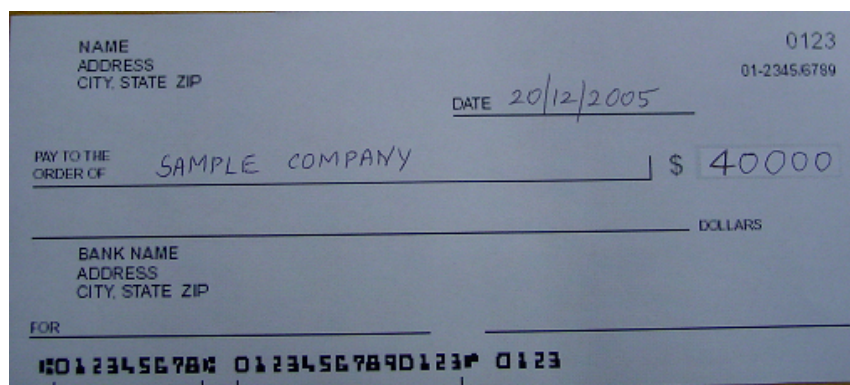
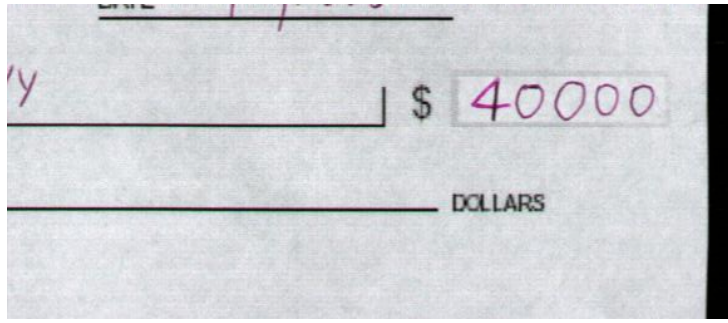
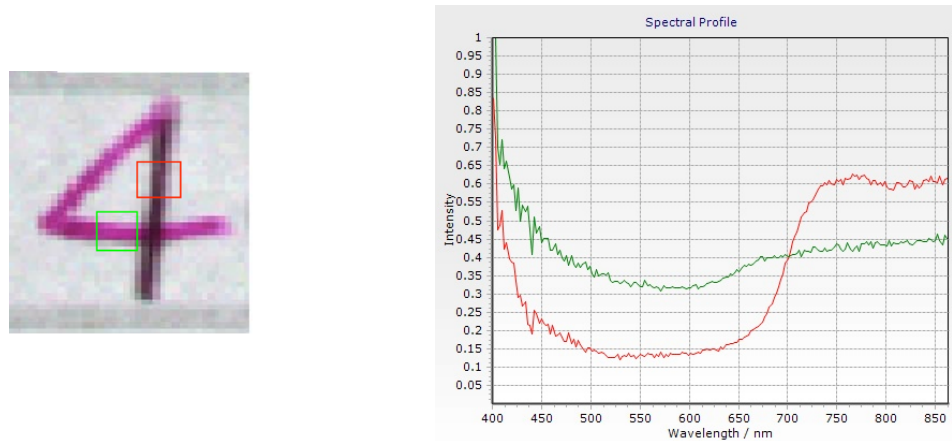


Figure 7.15 RGB image of the forged cheque. The alterations could not be seen here.

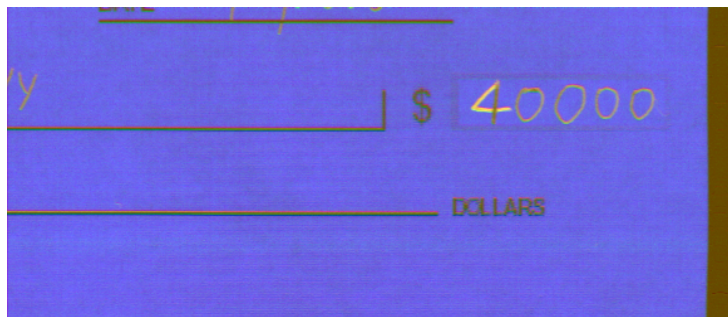


**Figure 7.16** False colour image of the cheque created with three different wavelength bands. This shows that the number '1' has been altered to number '4'



**Figure 7.17** Reflectance spectra of the horizontal line (green plot) and vertical line (red plot) in the number '4'. The difference in spectra shows that different inks were used.

PCA was performed on the hyperspectral data of the forged cheque to reduce its spectral dimension. A composite colour image created with images of three principal components showed significant contrast differences between the ink used in original writing and the one used in alteration. This image is presented in Figure 7.18.



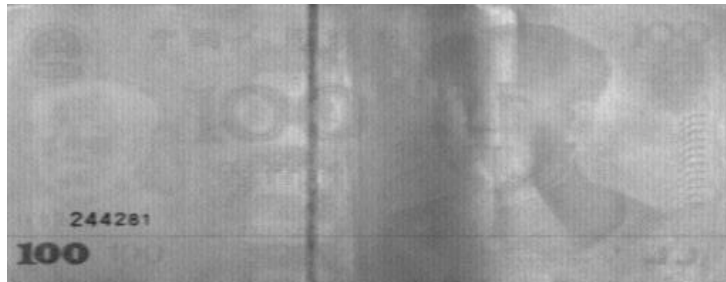
**Figure 7.18** PCA colour composite image of forged cheque. This shows significant contrast differences between the two types of inks used in number '4'



The hyperspectral images of the Chinese (Yuan) banknote showed significant variations across different wavelength bands. The different types of inks used in printing the banknote showed unique spectral features. Certain inks on the banknote became transparent in the Visible Near Infrared region (VNIR) of the spectrum. This is shown in Figure 7.20. The false colour image of the bank note formed with three different wavelength bands highlighted different types of inks which are not clearly visible in the normal colour image.



*Figure 7.19* Colour image of the Yuan banknote



*Figure 7.20* Greyscale image of Yuan banknote at 800nm. Certain inks are transparent in this band.



*Figure 7.21* False colour image of Yuan banknote created with images at three wavelength bands. This highlights different inks used in printing the document.

PCA was applied to the hyperspectral data of the Yuan banknote to remove the redundant information. A composite colour image created with images of three principal components showed greatest contrast among various inks in the printed banknote. This is shown in Figure 7.22. One of the potential applications of hyperspectral analysis of inks and pigments is to reveal document forgery. The observed spectral features at various wavelength bands compared against unknown samples of Chinese Yuan banknotes will help to identify counterfeited banknotes.



*Figure 7.22 PCA colour composite image of Yuan banknote. This indicates significant contrast differences among the inks used in printing the document.*

## 8. CONCLUSION AND FUTURE WORK

Hyperspectral imaging (HSI) proves to be a breakthrough among imaging technologies. The potential of HSI has been realised in many applications. The scope of HSI has widened over the past decade. HSI has attracted huge research interests from both academia and industry. The main focus of this research has been to develop a system which will reduce the complexity involved in HSI and make this technology as simple and as easy to use.

A software system was developed which provides a user friendly interface to the HSI hardware, camera and scanner modules. This system generates hyperspectral data cube with automatic synchronisation between data acquisition from camera and the motion of scanner. This system supports a wide range of scanners and cameras facilitating for use in a wide range of applications. Plug and play design architecture allows various combinations to design a HSI system using various scanners and cameras.

Data acquisition techniques which allow capturing of relevant information in a multidimensional hyperspectral dataset were developed. In addition, the developed software tools allow users without any specialist knowledge in signal processing to process the hyperspectral data. The visualisation techniques developed also provide a real time display of hyperspectral data and facilitated easier user interpretation. A graphical user interface which allows data of many dimensions to be viewed on a standard computer monitor was implemented.

The design of the software architecture adopted provides far flexibility and allow dynamic configuration of different types of camera or scanner to be made within the application. The techniques developed in this research have

been successfully applied in real time applications revealing the huge potential of HSI. The applications include environmental monitoring, dental imaging for detection of gum diseases, fruit quality determination, imaging of body parts, fingerprint detection in forensic analysis and quality assessment of beef etc.

Future work will investigate implementing more post processing techniques for hyperspectral data. This will provide a single platform for both data acquisition and data analysis. Various data analysis techniques such as principal component analysis, spectral angle mapping and Fourier transform can be utilised to extract detailed information from hyperspectral imagery. There is opportunity to explore more on visualisation of hyperspectral data such as an interactive three dimensional view of hypercube in real time. There is also the opportunity to integrate new cameras and scanners. This development will be particularly useful for a more successful commercial HSI system.

Currently, the decision making process for the applications using spectraSENS is subjective. We need to develop quantitative methods to generate a knowledge based system which can be an expert system, pattern classifier, a neural network system etc enabling intelligent decision making capability. A support vector classifier can be implemented to perform classification of nonlinear data. In scenarios of a skewed dataset, where there is only one predominant class, a one class classifier will be very useful in decision making.

## REFERENCES

1. Hungate, W., Watkins, R., Borengasser, M. (2008). *Hyperspectral Remote Sensing: Principles and Applications*, Retrieved from <http://www.crcnetbase.com/doi/book/10.1201/9781420012606>
2. Salem, F., Kefatos, M. (2001). *Hyperspectral image analysis for oil spill mitigation*. Retrieved from [http://www.spectir.com/assets/Text%20Files/Hyperspectral\\_Image\\_Analysis\\_for\\_Oil\\_Spill\\_Mitigation-Salem.pdf](http://www.spectir.com/assets/Text%20Files/Hyperspectral_Image_Analysis_for_Oil_Spill_Mitigation-Salem.pdf)
3. Millar, S.J., Whitworth, M.B., Chau, A. (2008). Mapping food composition using NIR hyperspectral imaging. *New Food*, 3, pp. 36-39.
4. Yoon, S.C., Lawrence, K.C., Line, J.E., Siragusa, G.R., Feldner, P.W., Park, B., Windham, W.R. (2010, March 12). Detection of *Campylobacter* colonies using hyperspectral imaging, *Sensing and Instrumentation for food quality and safety*, 4, 35–49.
5. Padoan, R., Steemers, Th.A.G., Klein, M.E., Aalderink, B.J., Bruin, G.E. (May 2008). Quantitative hyperspectral imaging of historical documents: technique and applications, *9th International Conference on NDT of Art, Jerusalem Israel*, 25-30.
6. France, F.G., Toth, M.B., Hansen, E.F. (2010). Challenges of Linking Digital Heritage Scientific Data with Scholarly Research: From Navigation to Politics, Retrieved from <http://dh2010.cch.kcl.ac.uk/academicprogramme/abstracts/papers/html/ab-756.html>
7. Alabboud, I. (2009). *Human retinal oximetry using hyperspectral imaging*. PhD Thesis, Heriot-Watt University.

8. Myers, J.S. & Miller, R.L. (2005). Optical airborne remote sensing. In *Remote sensing of coastal aquatic environments: technologies, techniques and applications* (pp. 55-56). Netherlands: Springer.
9. Gat, N. (1999, March). Directions in environmental spectroscopy. *Spectroscopy showcase*, p. 1.
10. Kamruzzaman, M., ElMasry, G. (2011). Application of NIR hyperspectral imaging for discrimination of lamb muscles, *Journal of Food Engineering*, p. 4.
11. Babar, M.A., et al. (2006, March 27). Spectral Reflectance to Estimate Genetic Variation for In-Season Biomass, Leaf Chlorophyll, and Canopy Temperature in Wheat, *Crop Science*, 46, 1046–1057.
12. Gitelson, A.A, Gritz, Y., Merzlyak, M.N. (2003) Relationships between leaf chlorophyll content and spectral reflectance and algorithms for non-destructive chlorophyll assessment in higher plant leaves, *Journal of Plant Physiology*, 160, 271–282.