

## ABSTRACT

Title of document:

CAMERA SPECTRAL SENSITIVITY  
CHARACTERIZATION USING A BLACKBODY  
SOURCE

Rucha Bedarkar, Master of Science (with thesis) in  
Mechanical Engineering, 2011.

Directed by:

Assistant Professor, Dr. Peter B. Sunderland,  
Department of Fire Protection Engineering

With digital cameras emerging as more effective tools for scientific research, there is increasing need for accurate and inexpensive ways to calibrate them. In particular, to date there has been no simple method to measure camera sensitivity as a function of wavelength. For example, narrow bandwidth monochromator beams are expensive and have calibration problems, while color chart method is unreliable owing to illumination dependence. This thesis presents a novel technique for spectral sensitivity calibration of a camera (or any black-and-white cameras or color sensors) using blackbody furnace operating at 650 – 1250 °C. Images recorded at 11 different temperatures are observed for red, green, and blue camera outputs. Using Planck's Law to calculate the incident light intensities, the three color sensitivities as functions of wavelength are computed using MATLAB function that optimizes the spectral sensitivities until the blackbody measurements are closely matched. The results are in reasonable agreement with published sensitivities.

CAMERA SPECTRAL SENSITIVITY CHARACTERIZATION USING A  
BLACKBODY SOURCE

By

Rucha S.Bedarkar

Thesis submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park, in partial fulfillment  
of the requirements for the degree of  
Master of Science  
2011

Advisory Committee:

Associate Professor Peter Sunderland, Chair

Professor James G. Quintiere

Professor Bongtae Han



©Copyright by

Rucha Bedarkar

2011

## Dedication

This thesis is dedicated to my parents and family for their relentless support and care .

## **Acknowledgements**

This research was a project sponsored by NASA and was conducted for ‘SPICE’ (Smoke Point Co-flow Experiment). I am immensely grateful to Dr.Peter Sunderland for giving me an opportunity to work on this thesis. He has been the greatest source of inspiration all along and the best mentor and guide one can ask for. I am also thankful to Dr.Zeng-Guang Yuan, who initiated this project together with Dr.Sunderland and has been a great support from time to time. Dr.Vivien Lecoustre has been of invaluable help through the various inputs he has given me from time to time. Dr. Mehran Mohammed, my colleague has a big contribution in the experimental part of the thesis with his technical expertise. I would like to mention Haiquing Guo and Jose Castillo for their tips on RAW file processing that made a huge difference. At this point, I also want to mention Dr.S.Balachandran who restored my confidence while I was going through a difficult phase with my sickness.

My parents, Mr.Sanjay Bedarkar and Ms.Sneha Bedarkar, and my brother Abhishek have forever been my pillars of support and it’s their love and blessings which carried me through life’s biggest hurdles. My second family, in the form all my close friends, is also very special to me and I owe a great deal of every success to them.

**List of Tables**

Table 1: Shutter times of response across the temperature range.....18

Table 2: Results of parameter variation on RGB sensitivities.....49

## Table of Contents

List of Tables .....	iv
List of Figures.....	viii
Chapter 1 INTRODUCTION.....	1
1.1. Motivation for color calibration .....	1
1.2. Working of a digital camera.....	2
1.2.1. Image Capture.....	3
1.2.2. Transmission of charges by CCD sensors.....	4
1.2.3. Internal Processing .....	4
1.2.4. Recording and storage .....	4
1.3. Literature review.....	5
1.3.1. Basic Camera Response model .....	6
1.3.2. Monochromator method .....	7
1.3.2.1. Simple estimate.....	7
1.3.2.2. Weiner Distribution .....	8
1.3.2.3. Pseudo Inverse method .....	8
1.3.3. Color chart based analytical methods.....	9
1.4. Need for a better method.....	11
1.5. Use of blackbody radiation for calibration .....	12
1.6. Objectives.....	12

<b>Chapter 2</b>	<b>EXPERIMENTAL METHODS .....</b>	<b>14</b>
2.1.	Design .....	14
2.2.	Set-up.....	15
2.2.1.	Blackbody furnace:.....	15
2.2.2.	Camera.....	16
2.2.3.	Surrounding lighting conditions and luminance.....	16
2.3.	Procedure .....	17
2.3.1.	Focusing .....	17
2.3.2.	Mounting.....	17
2.3.3.	Temperature control.....	17
2.3.4.	Computer controlled capture.....	17
<b>Chapter 3</b>	<b>ANALYTICAL METHODS .....</b>	<b>19</b>
3.1.	Physics of blackbody surfaces .....	19
3.2.	Planck's Law: .....	20
3.3.	Expression for total signal of a color:.....	21
3.4.	Mathematical model for camera response to blackbody radiation .....	21
3.5.	Nonlinear programming problem.....	22
3.5.1.	Statement of the problem .....	23
3.5.2.	Constraints on the algorithm .....	24
3.5.3.	Demonstration of the path followed by the proposed algorithm.....	24

<b>3.6. MATLAB Optimization Toolbox.....</b>	<b>26</b>
<b>3.6.1. Algorithms .....</b>	<b>27</b>
<b>3.6.2. Test on known sensitivity based ideal data.....</b>	<b>29</b>
<b>Chapter 4 RESULTS AND DISCUSSION .....</b>	<b>34</b>
<b>4.1. Analysis of experimental results .....</b>	<b>34</b>
<b>4.2. Results on optimization.....</b>	<b>43</b>
<b>4.2.1. Results for the three colors for basic settings.....</b>	<b>43</b>
<b>4.2.2. Impact of various factors on optimization results .....</b>	<b>49</b>
<b>4.2.3. Comparison with published Colorchart calibration results .....</b>	<b>56</b>
<b>Chapter 5 CONCLUSION AND FUTURE WORK .....</b>	<b>59</b>
<b>Appendices.....</b>	<b>61</b>

## **List of Figures**

Figure 1-1: Inside a CCD digital camera.....	2
Figure 1-2: Flowchart summarizing working of a digital camera.....	3
Figure 1-3: CCD sensor used in Nikon D100.....	5
Figure 1-4: Measurement of spectral intensities using monochromator.....	7
Figure 1-5: Experiment set-up for spectral calibration with a MacBeth Color.....	10
Figure 2-1: Set-up for the experiment.....	15
Figure 2-2: Commercial blackbody furnace used in the experiment.....	16
Figure 3-1: Intensity as a function of wavelength and temperature calculated using Planck's law.....	20
Figure 3-2 a.: Plot of the sensitivity function generated by every trial set of variables calculated in the function that will be optimized by search algorithms.	
Figure 3-2 b. Plot of the measured signal and calculated signal vs. Temperature.....	25
Figure 3-3: MATLAB Optimization Toolbox screenshot.....	26
Figure 3-4 Flowchart for the optimization function.....	27
Figure 3-5: Flowchart for Active Set Algorithm .....	28
Figure 3-6 a.: Calibration curves as published by Calibration Lab .....	29
Figure 3-6 b.: RGB Signal obtained with sensitivities from Fig 3-6a.....	29

Figure 3-7 :Optimization results for Red color .....	31
Figure 3-8 :Optimization results for Green color .....	32
Figure 3-9 Optimization results for Blue color .....	33
Figure 4-1.a: Photograph processed without correction.....	35
Figure 4-1b: Photograph processed with gamma correction but color transformation correction not applied.....	35
Figure 4-1c: Photograph processed with gamma correction and color transformation correction.....	35
Figure 4-2: Measurement Aoi sample size –screenshot from spotlight.....	36
Figure 4-3: Photographs of processed samples selected from the experiment.....	37--39
Figure 4-4: Figures showing pixel intensities across a line profile and noise scatter in the multiple samples for every temperature and particular shutter time selected .....	41
Figure 4-5: RGB signals for selected samples for 11 temperatures normalized over shutter speed.....	42
Figure 4-6: Sensitivity function for green color with 20 points across the range 400-1600 nanometers.....	44
Figure 4-7: Sensitivity function for green color with 30 points across the range 400-1100 nanometers.....	44

Figure 4-8: Sensitivity function for green color with 50 points optimized across the range 400-1000 nanometers.....	45
Figure 4-9: Sensitivity function for blue color with 20 points optimized across the range 400-1600 nanometers.....	45
Figure 4-10: Sensitivity function for blue color with 30 points optimized across the range 400-1100 nanometers.....	46
Figure 4-11: Sensitivity function for blue color with 50 points across the range 400-1000 nanometers.....	47
Figure 4-12: Sensitivity function for red color with 20 points across the range 400-1600 nanometers.....	47
Figure 4-13: Sensitivity function for red color with 30 points across the range 400-1600.....	48
Figure 4-14: Sensitivity function for red color with 100 points across the range 400-1600 nanometers.....	48
Figure 4-15: Sensitivity function for red color with 150 points across the range 400-1000 nanometers.....	49
Figure 4-16: Absolute error variation according to test parameters.....	51
Figure 4-17: Relative error variation according to test parameters.....	51
Figure 4-18 Red sensitivity curves for different number of control points.....	52

Figure 4-19 Green sensitivity curves for different number of control points.....53

Figure 4-20 Blue sensitivity curves for different number of control points.....54

Figure 4-21: Spectral sensitivities for red color as defined on different ranges to see the impact of this constraint on the results.....55

Figure 4-22: Comparison of results from blackbody source to those obtained.....55

Figure 4-23: Plots for comparison of the spectral sensitivities for each of the colors computed with published color chart values.....56

Figure 4-24: Comparison of results from blackbody source to those obtained from MacBeth Colorchart (100 points) .....57

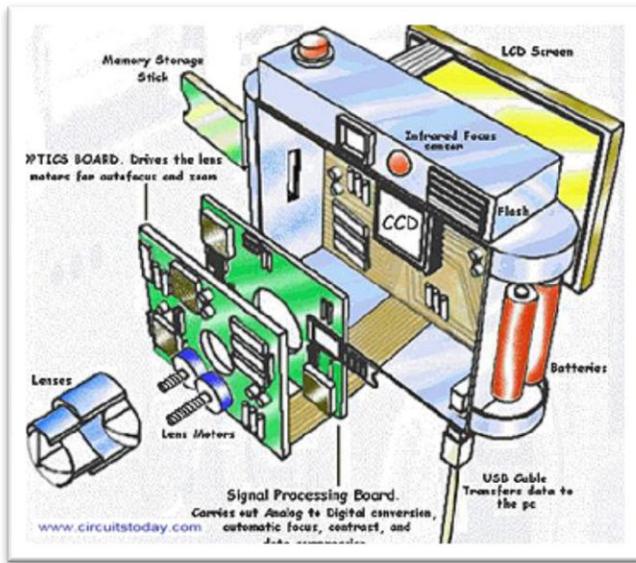
Figure 4-25: Plots for comparison of the spectral sensitivities for each of the colors computed with published color chart values.....58

## **Chapter 1    INTRODUCTION**

### **1.1. Motivation for color calibration**

The importance of information conveyed by visual media cannot be exaggerated. With the digitization of photography, images have grown to assume the role of more powerful tools of research than ever before. The overall progress of imaging science can occur only with simultaneous advancement in the image capturing device fabrication efficiency, performance of the electronic components and the accuracy of image processing. While the first two improve with the technological progress in the respective fields, the science of image processing has been blooming on its own in different directions as efforts are being directed towards obtaining the required version of the digital data collected. While photography as an art focuses on the aesthetic aspects of the digital image obtained and its enhancement, the scientific application of the images often first require a ‘true’ version of the image that helps one to extract information about the physical stimulus that was photographed for instance, animal coloration (Pike, 2011). The latter, thus, aims to rectify all the inconsistencies, errors and imperfections that are introduced in the original image as it is preprocessed, recorded and processed inside the camera. In other words, we need to ‘undo’ all the effects to be able to determine the exact image that the camera ‘sees’. After having crossed the optical system, the very first limitation that the image encounters is the performance capacity of the CCD sensors in the camera. These sensors are active in a finite bandwidth of the light spectrum and respond to every wavelength of every color (as the filters allow only specific color radiation to pass through). Hence, it is extremely important to apply that correction to be able to predict the original spectral intensities. As this data does not depend solely on the CCD sensors and is not made available by the manufacturers

of these cameras (or any of the color devices), it is imperative to calibrate this characteristic property of every specific camera for perfect interpretation of the picture data it records. Several studies have been conducted to that effect but a more reliable and inexpensive method still remains to be discovered. This research attempts to achieve a new more reliable technique for this spectral sensitivity calibration with the use of blackbody radiation.

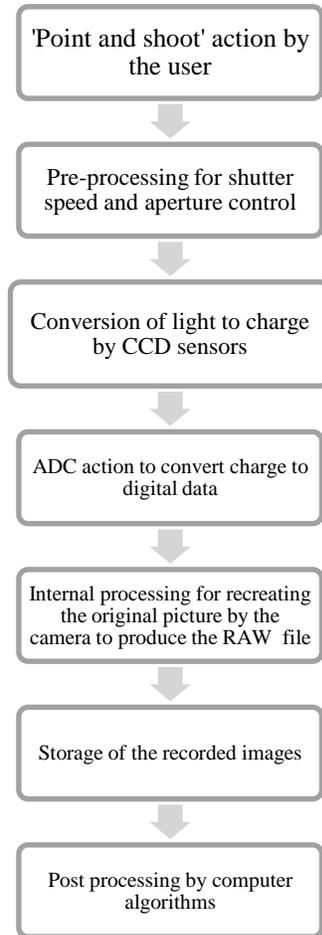


**Figure 1-1 Inside a CCD digital camera**  
(CircuitsFinder)

## 1.2. Working of a digital camera

To understand the role of color-wise spectral sensitivity in a device, a good place to begin would be a brief overview of the entire process of digital photography. The figure above shows the inside of a digital camera and its various basic parts. The image captured by the lens of the camera is focused on its CCD sensors, after which it is processed and stored in a digital RAW file. This file can then be processed to produce either the original image (identical to the real

stimulus that formed the input) or can be enhanced for various visual effects. The flowchart below (Figure 1-2) explains the order of the various operations in the complete process.



**Figure 1-2: Flowchart summarizing working of a digital camera**

It can be summarized as follows:

### **1.2.1. Image Capture**

As soon as the 'capture' button is triggered, the camera opens at the appropriate aperture for a reasonable shutter time, decided by the control systems that feed back the output of the sensors to these controls, to allow enough photons to pass through the lens of the camera. The lens, or system of lenses, of the camera direct photons from the object (illuminant source) to focus on an

array of photosensitive semiconductor cells – either CCDs (Charge Coupled Device) or CMOS (Complementary Metal Oxide Semiconductors). These respond to photons and generate a charge proportional to the intensity of the incident light. The light first passes through RGB filters, placed on each of the sensors, which are positioned next to each other in a definite pattern. Only one of the three colors can pass through each filter. The charges produced are conducted across the chip and converted to a value stored at the corner of the array.

### **1.2.2. Transmission of charges by CCD sensors**

The Analog-to Digital Converter digitizes the values stored by the CCD and creates an array of digital values ready to be processed inside the camera itself (Hainuat, 2006).

### **1.2.3. Internal Processing**

The array is still stored as raw values from the filter pattern. To get all three (R,G and B) values for each of the pixels, an algorithm processes the recorded single color values and reproduces the original color accurately through interpolation.

### **1.2.4. Recording and storage**

After the necessary primary processing the RAW file is recorded and stored on a card reader (Ref: Working of Digital Cameras - Basic circuit - CircuitsFinder-Free Electronic Circuit Diagram Design).

This, however, merely generates an input for the computer or processor, which is then processed in different ways as convenient to the particular application that the image is being used for. All in all , convolving the actual image from the camera output involves identification of all the added effects (brightness, gamma correction, geometric distortion, high and low pass

filters, edge enhancement ) , device and processing efficiencies, CCD sensor characteristics and their limitations, saturation limits, correction algorithms applied by default



**Figure 1-3: CCD sensor used in Nikon D100**

### **1.3. Literature review**

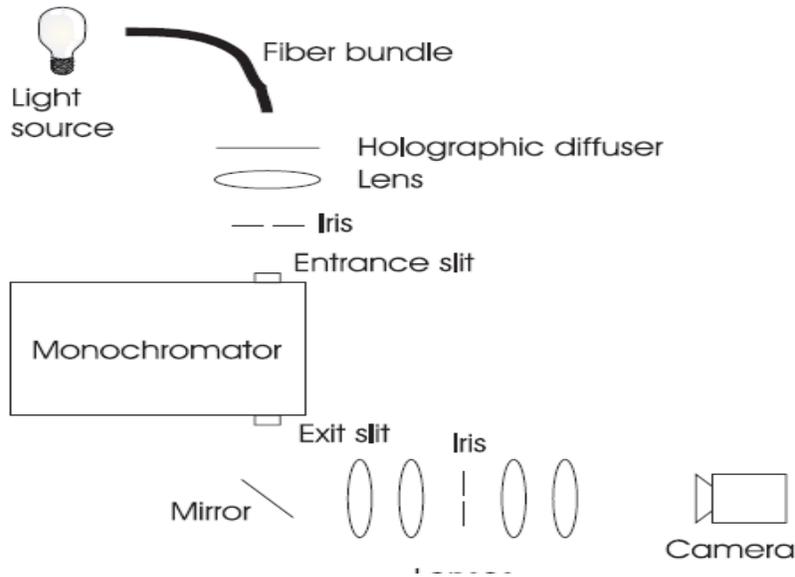
Of the filter-CCD characteristics, the single most important property that is essential to quantitative scientific measurements is the spectral sensitivity of these sensors. This function goes a long way in deciding the level of perfection to which reproduction of the true image is possible, i.e., it is a performance index of any color imaging system (Sharma & Trussell, 1997) . It directly affects the results of other colorimetric interpretations such as solving the color constancy problem i.e. determination of the exact spectral intensities and colors irrespective of the illuminant used to light up the subject (Finlayson & Funt, 1995, Foster, 1997). In order to calculate the incident spectral intensities, the inherent spectral sensitivity function of the CCD device used to read the image, across its entire bandwidth needs to be known.

### 1.3.1. Basic Camera Response model

The processing of color devices is generally done on the assumption that the response of the device is linear with respect to the intensity of the incident light. This assumption forms the basis for all the linear response models that have been developed to date, in order to estimate the spectral sensitivities. It has been found to be reasonably true from experiments conducted in earlier research (Vora P. , Farrell, Tiatz, & Brainard, 1997). However, some papers also provide examples of how the model can be appropriately augmented to include a known non-linearity in the RGB response to intensity. For instance, the gamma effect (Korsgaard & Andersen, 1998) affects the linear nature of camera response function with respect to intensity from improved picture quality but needs to be corrected for spectral analysis. These models use the following basic equation to generate a camera response model. The RGB response of the camera for the  $i^{\text{th}}$  pixel is given by

$$r_i = e \int_{\lambda_l}^{\lambda_h} s_i(\lambda) i(\lambda) d\lambda + n_i \dots\dots\dots(1.1)$$

Where  $s_i(\lambda)$  the spectral sensitivity of the  $i^{\text{th}}$  sensor type is,  $i(\lambda)$  is the incident power density per unit time at wavelength  $\lambda$ ,  $e$  is the exposure duration, and  $n_i$  is a normal random variable denoting noise. The wavelengths  $\lambda_h$ ,  $\lambda_l$  are the threshold values beyond which the spectral response of the sensor is negligible (Vora P. , Farrell, Tiatz, & Brainard, 1997). Based on this equation, several methods have been used to determine the spectral sensitivity values from the RGB signal values. These methods can be broadly divided into two categories. Some methods use light from a single narrowband source (with the help of a monochromater) and the responses recorded thus are used to build a look up table.



**Figure 1-4: Measurement of spectral intensities using monochromator**

(Lauziere, et al., 1999)

**1.3.2. Monochromator method**

The responses are measured at small finite equidistant intervals in the spectral space and for the space in between the measured responses, various interpolation methods are used to assign values to the wavelengths in the intervals (Sharma & Trussell, 1997, Vora P., Farrell, Tiatz, & Brainard, 1997, Lauziere, Gingras, & F., 1999). The more prominent ones are reviewed below.

1.3.2.1. Simple estimate

A ‘simple’ estimate of the values was used for the estimation in some papers (Hubel, Sherman, & Farrell, 1994), (Vora P. , Farrell, Tiatz, & Brainard, 1997). This method uses sampled form of equation (1.1) given as

$$r = F \left[ \begin{array}{l} e(1)c_r(\lambda_1) \sum_j s_1(\lambda_j + j\Delta\lambda)\Delta\lambda \\ e(i)c_r(\lambda_i) \sum_j s_i(\lambda_j + j\Delta\lambda)\Delta\lambda \\ e(K_r)c_r(\lambda_{K_r}) \sum_j s_{K_r}(\lambda_j + j\Delta\lambda)\Delta\lambda \end{array} \right] \dots\dots\dots(1.2)$$

This equation is used to calculate the value of spectral response  $c_i(\lambda_i)$  by measuring the total RGB response from a narrowband at equal intervals. Thus, the value of total signal itself becomes the value for sensitivity. The noise variation is ignored in this method and the following equation is used for calculation of spectral response at discrete wavelength values.

$$c_i(\lambda) = \frac{F^{-1}(r_i) - \bar{n}}{e(i) \sum_j s_i(\lambda_j + j\Delta\lambda) \Delta\lambda} \dots\dots\dots(1.3)$$

Interpolation for the wavelengths is generally done through simple averaging. Though this method provides a good starting point, it has some flaws. The variability of noise may greatly alter the predicted values as most cameras have a non-zero variation about the mean noise level. Secondly, the narrowband illuminant still has a certain finite width (of wavelength) across which the incident signal may vary. Hence, the distribution across this width needs to be considered.

### 1.3.2.2. Wiener Distribution

To eliminate these shortcomings, the Wiener estimation method was first used by W.K.Pratt (1978) for calculation of the response by weighting the spectral response distribution which was assumed to be a Gaussian distribution and noise was considered a vector, having a different value for every sensor.

### 1.3.2.3. Pseudo Inverse method

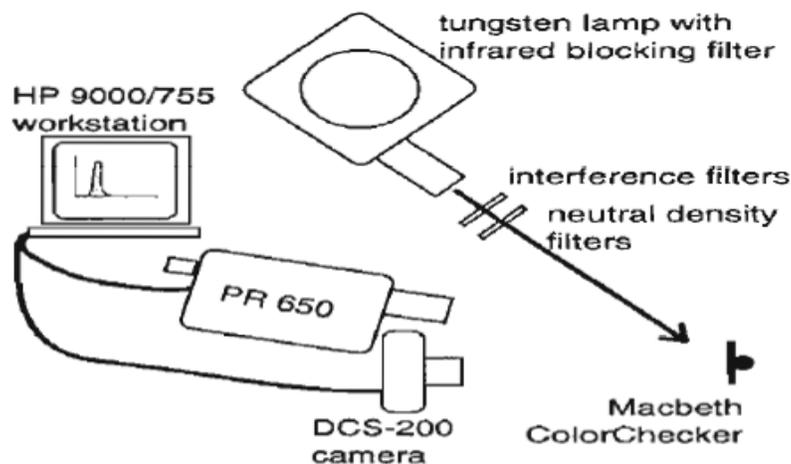
This method makes use of the pseudoinverse (A.Albert, 1972) of the reflectance matrix of the testing samples. It suffers from the lack of a smoothness constraint to compensate for spikiness and unrealistic fluctuations from the broadband data. It is extremely sensitive to noise. It thus fails to give a good reliable estimate of the sensitivity at any particular point. (A.Albert, 1972) . Later, a method was proposed to reduce the rank of the matrix to only include the more prominent values that formed a non-singular matrix along with a separate matrix to represent

noise in a vector form. This method called the Rank-deficient Pseudo Inverse Method (C.Reinsch, 1971, I.T.Jolliffe, 1986) and has been used by several authors for their analyses (Vora P., Farrell, Tiatz, & Brainard, 1997, H.J.Trussell, 1996, Farrell & Wandell, 1993, Trussell & Sharma, 1993).

While these methods are the most clear estimates of the sensitivity function, their accuracy completely depends on the camera (in general, device) efficiency, measurement efficiency. What make them even more impractical are the costs involved in using monochromators of such fine narrowband width. Even the state-of-the-art ones have a width of at least 8 nm.

### **1.3.3. Color chart based analytical methods**

The disadvantages associated with the monochromator method have encouraged research in the second category of methods for characterization of the spectral sensitivities of a color device. These methods simply use linear regression or other numerical methods on the spectral responses broken from the total RGB response in an image coming from a source divided into sections of known reflectance or transmittance values such as the Macbeth Color Checker<sup>TM</sup> chart. This chart is sectioned into squares of different colors which are surfaces of known reflective spectra (McCamy, Marcus, & Davidson, 1976). Thus, one single image can give values from known spectra. These, together with the basic assumption of linear response models (equation (i)) can be used to estimate the spectral sensitivities. Fig. 3 depicts such an experiment using MacBeth Color chart for calibration.



**Figure 1-5: Experiment set-up for spectral calibration with a MacBeth Color chart (Hubel, Sherman, & Farrell, 1994)**

Maloney & Wandell (1985) showed that the spectral response of a camera (or other color device) can be expressed as a finite-dimensional vector. This encouraged a completely new approach for calculation of the function numerically through linear regression, without use of any narrowband illumination requirement. The work of (Trussell & Sharma, 1993) suggests methods to smoothen the jagged sensitivities obtained by simple linear regression and imposed convexity constraint on the calculated function. Further, (Finlayson, Huble, & Hordley, 1998) obtained much better solutions by imposing constraints such as unimodality and positivity. They also represented the whole spectral sensitivity characterization empirical problem as a quadratic programming problem. The unimodality constraint, however, was implemented by making further estimations about the wavelength at which the peak sensitivity was observed from results of monochromator experiments. Aslam, Aslam and Finlayson (2002) further induced smoothness to the approximated function by use of half-sine basis function. Another turning point occurred when Barnard and Funt (2002) used a weighted sum of the relative error and smoothness factor

for the quadratic objective function which gave rise to curves with an optimal smoothness and error combination. Both this paper and Dyas (2000) independently used Tikhonov regularization (Hansen, 1998) in order to reduce the norm of the matrix and make the process of smoothing simpler. The latest work (Aslam & Hardeberg, 2006) in this area is an analysis that has made an attempt to remove the sharp peaks in the jagged curve makes use of a slightly different method. The authors have altered the measured response curve to make it smoother *before* the calculation for spectral sensitivity. The paper shows that the very same pseudoinverse matrix calculation together with TSVD (Truncated Singular Value Decomposition (Finlayson & Funt, 1995) can considerably improve result.

#### **1.4. Need for a better method**

Although there has been constant improvement over time in the algorithms that calculated spectral sensitivity function based on response to the MacBeth Colorchart, there are inherent flaws in the very use of color chart to provide reference values for the spectral responses against which the linear programming is performed. Firstly, although the reflectance values of the color patches are known, the surrounding illumination affects the amount of signal that the camera generates. The function calculated is applicable only to photographs taken in the same surrounding illumination. Not only does this limit the usability of the calibration chart but also makes it impossible to study photographs taken in outer space using the same function. Over the time, the nature and properties of the surface of the chart may change or depart from the values that the chart claims to have. Finally, due to finite number of patches of known reflectance, there are limitations over the representation of the matrix of sensitivity function values in the equation. This was partly solved by converting the problem to a quadratic linear program by Finlayson, Huble, & Hordley, 1998. However, we need to devise method where we could easily increase or

decrease the rank of the reference array as the camera model would demand. Keeping these drawbacks in mind, a new method is proposed to use computations to calculate the sensitivity function for a camera or any color device.

### **1.5. Use of blackbody radiation for calibration**

This method uses the fact that the intensity of the signal coming from a blackbody at a certain temperature and wavelength can be very accurately calculated from the Planck's law. Now referring back to the equation (1.1), if we can measure the RGB values for a blackbody at a temperature on a particular camera, the only unknown is the sensitivity function for that particular body. We can now divide the bandwidth into an optimum number of points and compute the best corresponding array of sensitivities such that the total signal generated at a particular temperature by the blackbody is same as the measured value. All we need is a set of photographs spread over a convenient temperature range taken with the blackbody in absence of any other source of light. The details of this method will be explained in the following chapters.

### **1.6. Objectives**

Thus motivated, the objectives of this study are to put forth a completely new method for camera spectral sensitivity calibration that would be universally applicable to all lighting condition, would be more reliable with respect to the reference signal values that it uses for the purpose of estimation and that would enable near perfect recovery of the actual object stimulus from the digital image obtained by a color capturing device. A statistical analysis for noise reduced sampling of data and performance characteristics of the optimization module on MATLAB has also been included to demonstrate its accuracy and sensitivity to different experimental and analytical factors. This project has been specially designed for calibration of a camera Nikon

D100 used by NASA in the SPICE ( Smoke Point In Co-flow Experiment) project to study temperature distribution in flames.

## **Chapter 2    EXPERIMENTAL METHODS**

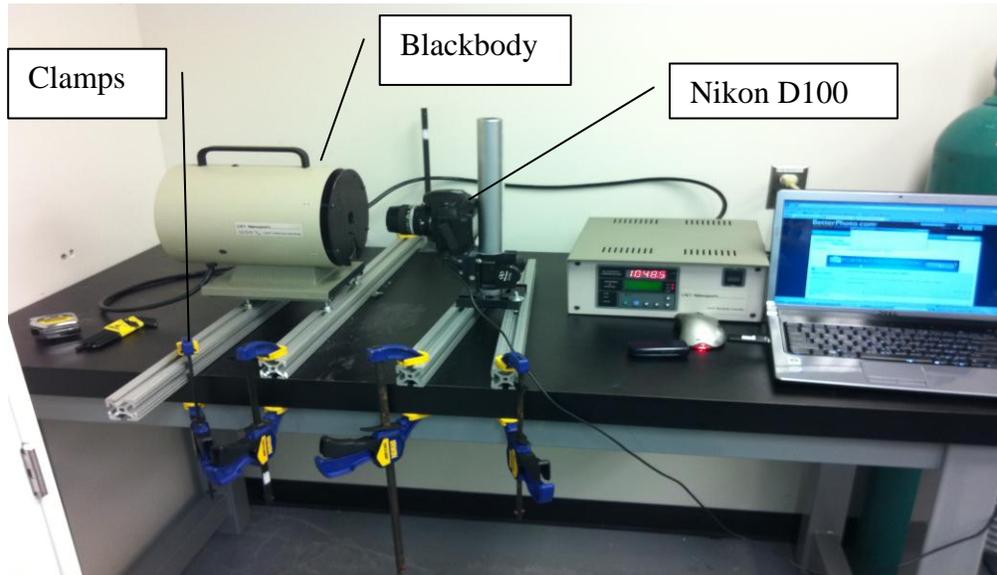
The experimental section of this research was aimed at the measurement of camera response to the blackbody source at various temperatures to obtain a set of values with minimal noise that could be used as input to the optimization algorithm and was carried out together with Mr.Mehran Mohammed.

### **2.1.Design**

The requirements of the mathematical model dictate the design of this experiment. To form an (experimentally) optimal array of signal values with the apparatus, the experiment was designed to capture images within a range of temperatures that is enough to exhibit the dependence of signal on the magnitude of absolute temperature. A wide range of shutter speeds was used to get strong signals for all three colors. The blackbody furnace pictures were taken on the way up the temperature scale and also on its way down to room temperature. This allowed a study of the hysteresis in the radiation from the original source as well as the camera so that the noise could be mitigated. Also, for the same settings, photographs were taken twice to average out the error in operation involved. This experiment was repeated twice again to examine the uniformity and reliability of the blackbody and camera both with respect to the values that the twain produce at various temperature-shutter speed combinations.

## 2.2. Set-up

The apparatus consisted of a blackbody radiator, a digital camera, a table and mounting components along with a computer to perform the processing and computations.



**Figure 2-1: Set-up for the experiment**

### 2.2.1. Blackbody furnace:

Although a perfect blackbody is a hypothetical concept, isothermal cavities that behave as near-perfect blackbodies with an emissivity of 1. In this experiment, the model ‘Oriel Instruments<sup>TM</sup> Blackbody’ with a 1 inch circular cavity opening that can operate between 50 °C-1200 °C. Other details have been listed in Appendix A. As emissivity is about 1 in the operating region we can justify our assumption of the cavity behaving as an ideal blackbody for all practical purposes.



**Figure 2-2: Commercial blackbody furnace used in the experiment**

### **2.2.2. Camera**

Any digital camera or electronic color image recording device could be calibrated with the method we propose. A Nikon D100 is chosen for our experiment as it's the exact same model that is used for photography from a space station installed by NASA and thus can be used for this specific application. Properties of the camera may be found in detail in Appendix B.

### **2.2.3. Surrounding lighting conditions and luminance**

All the light sources in the room were switched off and surfaces of high reflectance were kept away so that surroundings would not contribute to the image and the furnace was the sole illuminant.

## **2.3. Procedure**

### **2.3.1. Focusing**

The blackbody and camera were aligned and mounted so that the reflective plane of the blackbody was at the focal point of the camera (with the extension ring).

### **2.3.2. Mounting**

The blackbody was first screwed tightly on to metal bars over the table so we could have a fixed reference for the rest of the system. The camera was placed at different positions to check for the position that would yield sharpest images of the inner cone of the cavity of the blackbody. The camera is then screwed on to a stand which was also screwed down to the working table through channels. C-Clamps were used to fasten every component to the working table.

### **2.3.3. Temperature control**

The temperature of the blackbody is raised through the controller setpoint. The alarms control the heater and fan operation to bring the blackbody to a required temperature.

### **2.3.4. Computer controlled capture**

Once a thermally steady state was attained at a temperature, photographs were taken with the help of software, Camera Control Pro, which allows complete control of the camera through a computer. The image then directly gets transferred to the computer controlling it. Different shutter speeds were used for every temperature to ensure collection of relevant data that confirms linear characteristics of the camera and also creates a sample with strong signal from every color. The procedure was repeated with duplicate samples for every temperature and shutter speed. After having reached the highest temperature, the body was allowed to cool in the same number

of steps. The aperture was fixed throughout the experiment. The temperature range for the experiment was 960K to 1460K.

**Table 1: Shutter times of response across the temperature range**

Temperature(K)	Shutter time (seconds)		
910	15	10	3
960	6	4	1 1/2
1010	10	6	4 1 1/2 1/10
1060	6	4	1 1/2 1/10 1/15 1/30 1/60
1110	4	1	1/4 1/10 1/15 1/60 1/250 1/500
1160	1	1/2	1/4 1/10 1/30 1/60 1/125 1/250
1210	1	1/2	1/4 1/10 1/15 1/30 1/60 1/125 1/250
1260	1/2	1/4	1/10 1/30 1/60 1/125 1/250 1/500 1/1000
1310	1	1/2	1/10 1/4 1/10 1/30 1/60 1/125 1/250 1/500 1/1000
1360	1	1/4	1/10 1/60 1/125 1/500 1/1000
1410	1/4	1/10	1/30 1/60 1/125 1/250 1/500 1/1000
1460	1/4	1/10	1/30 1/60 1/125 1/250 1/500 1/1000

Multiple samples were collected from every combination of temperature and shutter speed. These facilitated removal of experimental error by averaging out the inconsistencies as we will see in Chapter 4.

### **Chapter 3    ANALYTICAL METHODS**

The method proposed involves the characterization of spectral sensitivity of a digital camera or other light sensor using the blackbody experiments described in Chapter 2. This source was chosen on account of the fact that blackbody radiation for a particular wavelength and temperature is known with extremely good accuracy. Consequently, there is much more reliable data for the actual input signal (blackbody intensity) as also the output signal (RGB values from the camera image). It is thus possible to eliminate the need to account for surface characteristics and the subsequent inconsistencies (the use of a color chart and the flaws associated with that method). The development of the analytical procedure is reviewed in this chapter.

#### **3.1.    Physics of blackbody surfaces**

An ideal blackbody can be defined as a surface with emissivity 1 i.e. a surface that absorbs the entire electromagnetic radiation incident on it (Gustav Kirchoff, 1860). Although perfect blackbodies are hypothetical, methods have been devised to construct isothermal cavities such that the emissivity to the small opening to this cavity is almost 1, after a series of internal reflections of electromagnetic rays in the main cavity. The blackbody radiator chosen for the experiment is one such artificial source.

### 3.2. Planck's Law:

Planck's Law (1900) gives a unique distribution function that determines the intensity of thermal power emitted by a blackbody surface at a given temperature, at any wavelength range under thermal equilibrium. According to this law,

$$I(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1} \dots\dots\dots(3.1)$$

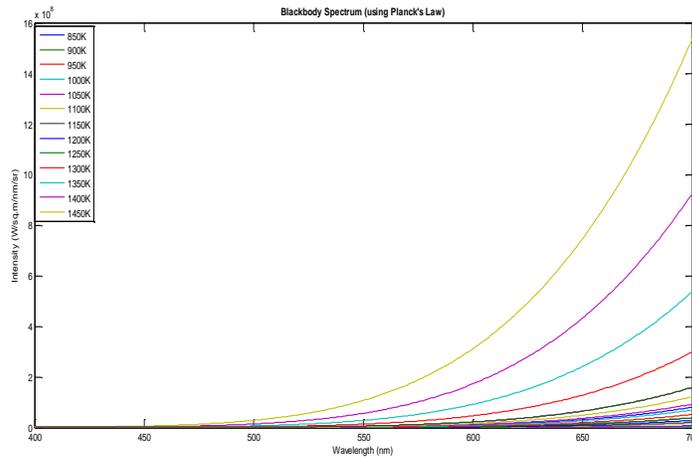
Where h->Planck's constant (6.62606896 x 10<sup>-34</sup> Js)

c-> Speed of light in vacuum (299,792,458 m/s)

k-> Boltzmann constant (1.3806504 x 10<sup>-23</sup> J/K)

λ-> Wavelength in nanometers (nm)

T-> Surface temperature in degree Kelvin (K)



**Figure 3-1: Intensity as a function of wavelength and temperature calculated using Planck's law.**

### **3.3. Expression for total signal of a color:**

Every device has a system to convert the incident source through sensors to an output image and this involves various factors which get multiplied to the actual intensity measured by the electronic sensors. The magnitude of this scaling effect is a characteristic of the particular device and its associated internal programming apart from the setting modified by the user. In the context of this thesis, the total product of these constants that appear during the conversion can be clubbed together into a constant, say  $c_1$ . Note that the effect of almost all of these constants (except for noise sources and low pass filters) is to ‘scale’ up or down the original data and not to alter its relative distribution effects as the transformation is performed on the whole color plane and not a particular bandwidth. The constant  $c_1$  is thus a dimensionless factor that modifies the amplitude of intensity across the entire bandwidth uniformly. That leaves the spectral sensitivity as the only non-random part of the whole transformation function. Let us call this function  $S_i(\lambda)$ . The subscript ‘i’ denotes the color plane with respect to which the sensitivity has been defined. This function here only accommodates for the variation in the sensor output based on the variation in incident spectrum. The sensors are also responsive to the change in intensity of the input. However, the response to intensity is either linear or determinate in nature for most of the cameras and color devices. This factor has been discussed at length in experimental section of this thesis. With the assumption of linearity in sensor response to intensity, we can consider the factor as a part of the group of constants  $c_1$ .

### **3.4. Mathematical model for camera response to blackbody radiation**

For the case of input from a blackbody source, the total camera response at a particular temperature depends on the spectral sensitivity function apart from the processing constant  $c_1$  and intensity which can be estimated. Let  $GS_i(T)$  denote the total grayscale level or output signal

value for a particular plane (Red, Green or Blue) with a subscript ‘i’ that stands for the color plane (from red, blue and green). From the above, it can be mathematically expressed as :

$$GS_i(T) = c1 \cdot \int_0^\infty S_i(\lambda) \cdot I(\lambda, T) d\lambda = \int_0^\infty \frac{2hc^2}{\lambda^5} \cdot \frac{1}{e^{\frac{hc}{\lambda kT}} - 1} \cdot S_i(\lambda) d\lambda \dots \dots \dots (3.2)$$

As the camera sensors respond to wavelengths in a finite interval, the limits of the integral can be replaced as  $\lambda_h$  and  $\lambda_l$ , the higher and lower limit of the bandwidth of response, respectively. Thus,

$$GS_i(T) = \int_{\lambda_l}^{\lambda_h} \frac{2hc^2}{\lambda^5} \cdot \frac{1}{e^{\frac{hc}{\lambda kT}} - 1} \cdot S_i(\lambda) d\lambda \dots \dots \dots (3.3)$$

As  $S_i(\lambda)$  is completely unknown, analytical solution to this problem becomes extremely difficult. The analytical intensity function derived from Planck’s equation is complicated to integrate. Also, as we can see in the expression above, what we have as known is the total value of the definite integral, not the exact analytical function that it obeys. These limitations rule out the possibility of solving this problem analytically through Laplace’s transforms or Homotopy analysis.

**3.5. Nonlinear programming problem**

In absence of analytical methods, a good alternative is nonlinear programming with use of numerical methods. We have already identified the mathematical problem statement. Nonlinear programming technique can obtain an optimum solution to the system of nonlinear equalities, inequalities and constraints that we have in this case. The next step is formulation of the problem for nonlinear programming.

### 3.5.1. Statement of the problem

Eventually, the problem requires finding the optimal set of spectral sensitivity function values that yield a total signal equal to the measured value. The problem can thus be structured as follows:

#### **Optimization problem formulation:**

- **Goal:** Achieving sensitivity values that generate signal that fits with the measured values as closely as possible.

- **Variables:** Array of sensitivity function values at regular intervals. This is essentially the array that we have set out to optimize.  $S[h]$  becomes our array of variable for 'h' divisions of the bandwidth.

- **Measured signal:** Values obtained through actual experimentation. This is the reference array against which the optimization is performed. 'GS[n]' is the reference array for 'n' readings obtained.

- **Calculated signal:** Numerical integration over a finite wavelength range, of the product of sensitivity and intensity at a given wavelength and temperature. Say 'calc[n]' is our calculated signal array.

- **Objective function:** Absolute difference in the logarithms of calculated and measured values normalized over the corresponding measured values added to the difference between actual and calculated values forms an objective function the sensitivity to which is very high for a fitting optimization algorithm. This function is very similar to the quadratic programming with the error function as described by Finlayson (1998). However, absolute value allows the function to use the real magnitude instead of using the square of the value of error which gets reduced for

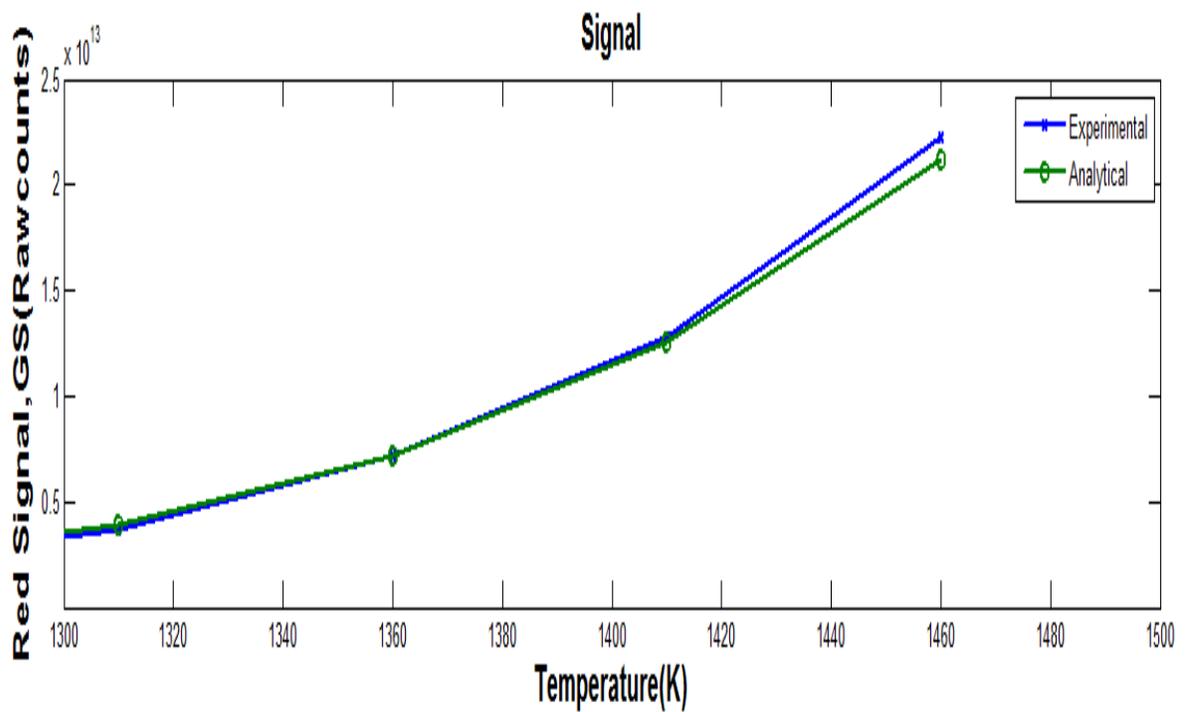
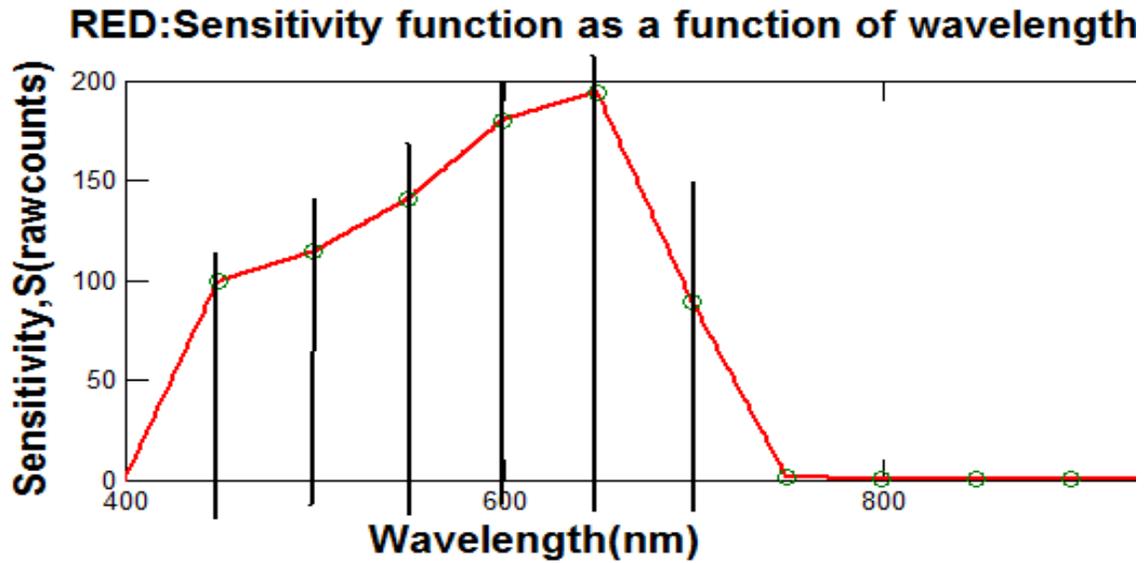
numbers between 0 and 1. The logarithms make use of the exponential nature of the signal function.

### 3.5.2. Constraints on the algorithm

- **Positivity:** All the variable values are positive by the very fact that they represent a count of photons that successfully get converted to charge in the camera sensors.
- **Band limits:** As the sensors of the digital camera have a finite bandwidth capacity within which they respond, we pick the most general range for all three planes (red, green and blue) as 400-700 nanometers

### 3.5.3. Demonstration of the path followed by the proposed algorithm

The following figures demonstrate the procedure that the algorithm should follow. Figure 3-2 a. shows the sensitivity function as generated by the function that is then optimized by the search algorithm to generate best possible value (minimum, in this case) of objective function through the set of control points which govern the nature of function generated by the algorithm. The figure 3-2 b. shows the calculated signal with function in fig. 3-2 a. and its fit with respect to the actual signal for the same color measured in the experiment over the range of temperatures.



**Figure 3-2 a. Plot of the sensitivity function generated by every trial set of variables calculated in the function that will be optimized by search algorithms.**

**Figure 3-2 b. Plot of the measured signal and calculated signal vs Temperature**

### 3.6. MATLAB Optimization Toolbox

There is a powerful mathematical analysis tool at our disposal, MATLAB for carrying out this optimization algorithm. MATLAB provides the perfect numerical computation environment necessary to carry out iterations sequentially for a problem like ours. The ‘Optimization Toolbox’ is specifically designed to have search functions designed to solve linear and nonlinear programming problems by computing the best possible values to attain a particular goal. The ‘fmincon’ algorithms in this toolbox are apt for numerical minimization problem that is to be solved. It finds the best value from a set constrained by the conditions of equalities, inequalities and bounds that the user can define. The following section describes in detail the algorithm that our code follows in the process of optimizing the solution set.

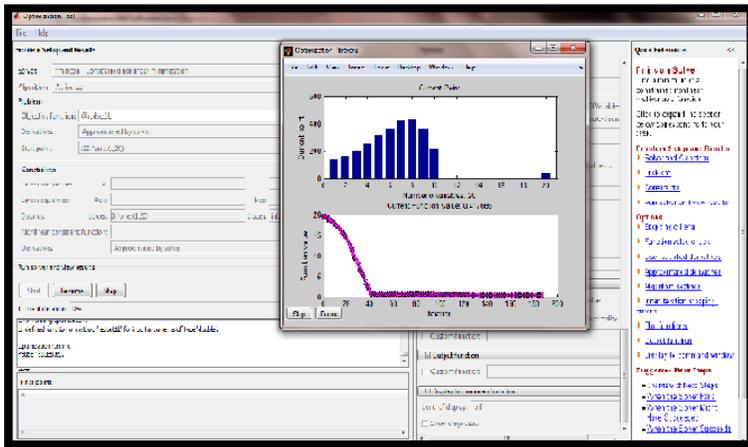
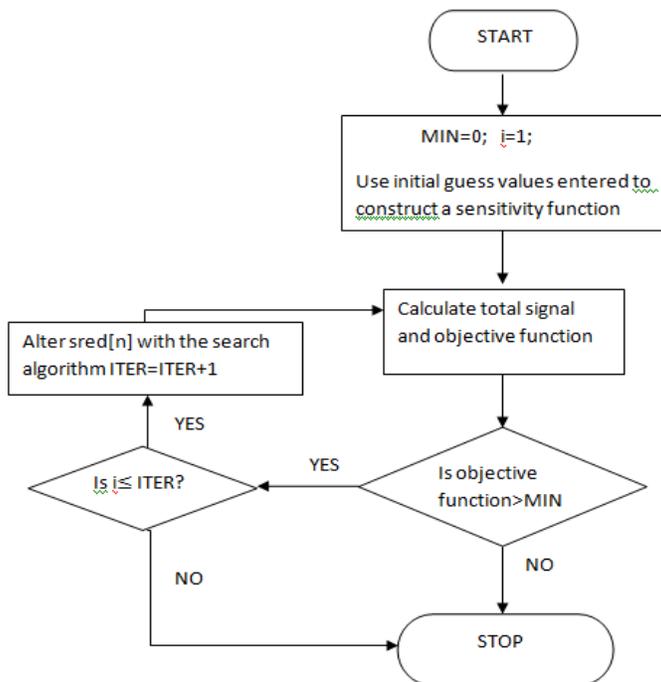


Figure 3-3: MATLAB Optimization Toolbox screenshot

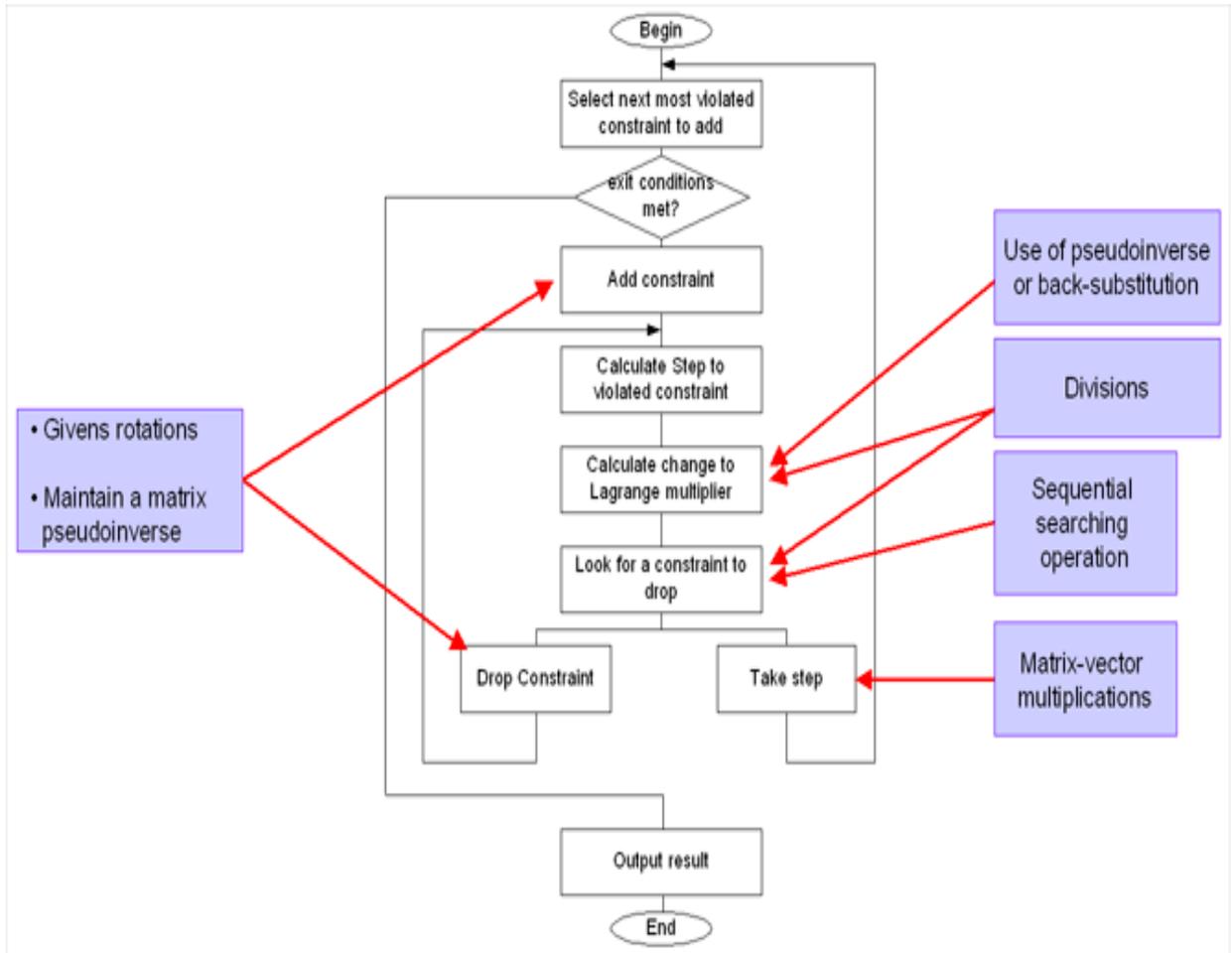
### 3.6.1. Algorithms

The main program calls the function to be optimized and the MATLAB Optimization toolbox function that contains the search algorithm. The optimized function accepts the array of sensitivity values at  $n$  points of a particular color, e.g.  $\text{optired}[n]$ . The bandwidth is divided into  $n+1$  parts. Now,  $\text{optired}$  supplies sensitivity value for last element of wavelength array division. Between two end points the function is treated as a straight line that connects these end points and a complete array  $\text{sred}[500]$  is calculated. An array of intensity at every wavelength  $\text{int}[500]$  is calculated for every temperature from the temperature array  $T[12]$ . Trapezoidal integration allows us to evaluate the total signal  $\text{calcred}[12]$  for each of the 12 temperatures. We have the array  $\text{expred}[12]$  from the experiments conducted. The objective function is then calculated with the absolute value of difference between the logarithmic values of these two arrays and is



**Figure 3-4 Flowchart for the optimization function**

returned by the function. 'Active set' algorithm is the actual search function generating program that computes a new value based on the change in objective function.



**Figure 3-5: Flowchart for Active Set Algorithm**

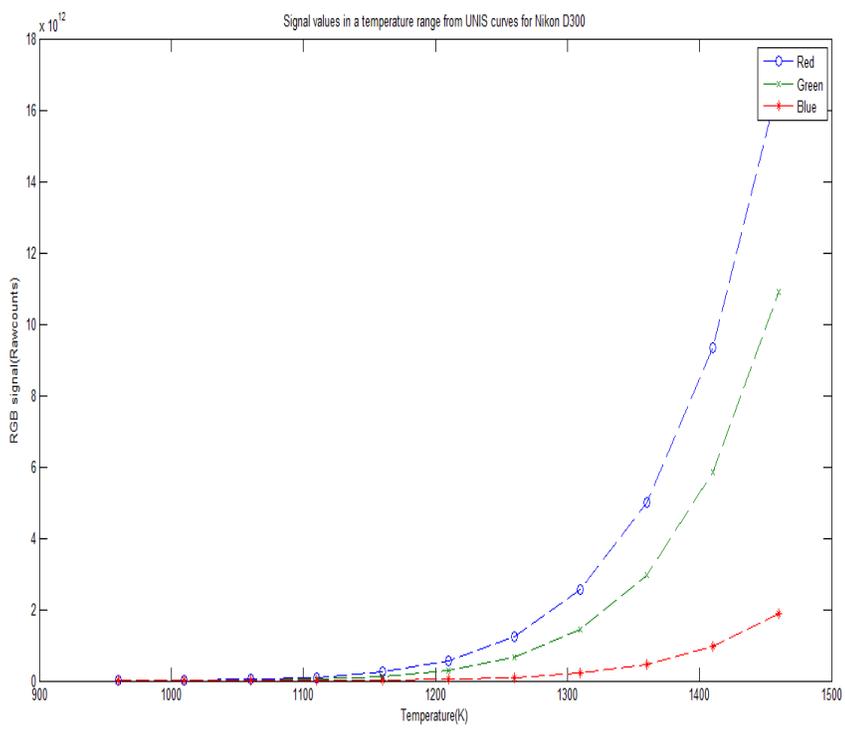
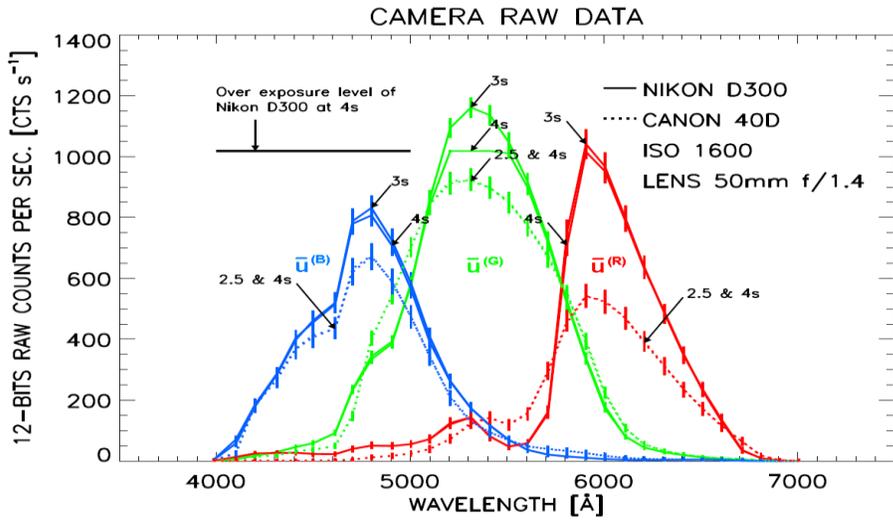
(Knagpe, 2009)

### 3.6.2. Test on known sensitivity based ideal data

This method was initially tested on data generated with reference values obtained from earlier publications and was found successful. The figure 3-5a. below shows a reference set of data that was obtained from Seigernes,et al.(1994). On using Planck's Law, the total RGB values of signal calculated as shown in figure 3-5 b.

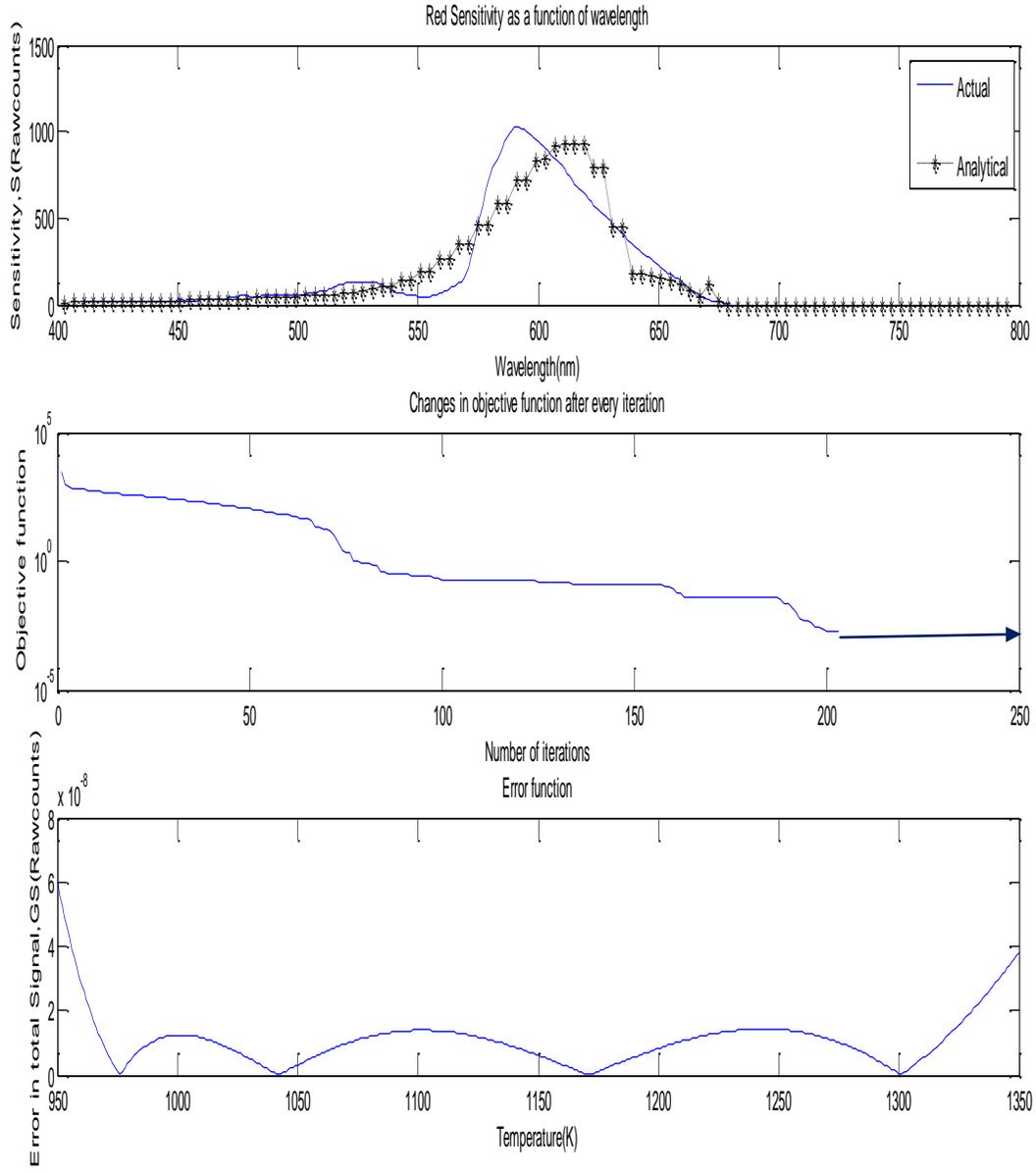
The optimization subroutine was then optimized to sharpen the accuracy further to obtain plots as shown in figures 3-6 – 3-8. These plots were results obtained on optimization of 99 points of each of the color's sensitivity curve that gave excellent convergence, evenly distributed error, when plotted across the temperature range within which it was minimized.

These results led to the inference that the method worked out well for all three curves with acceptable accuracy under ideal conditions with about 1% noise and confirmed its feasibility. The performance of the experiment was therefore justified as a creation of an actual prototype of the calibration system that was verified to have worked on an ideal set of data.

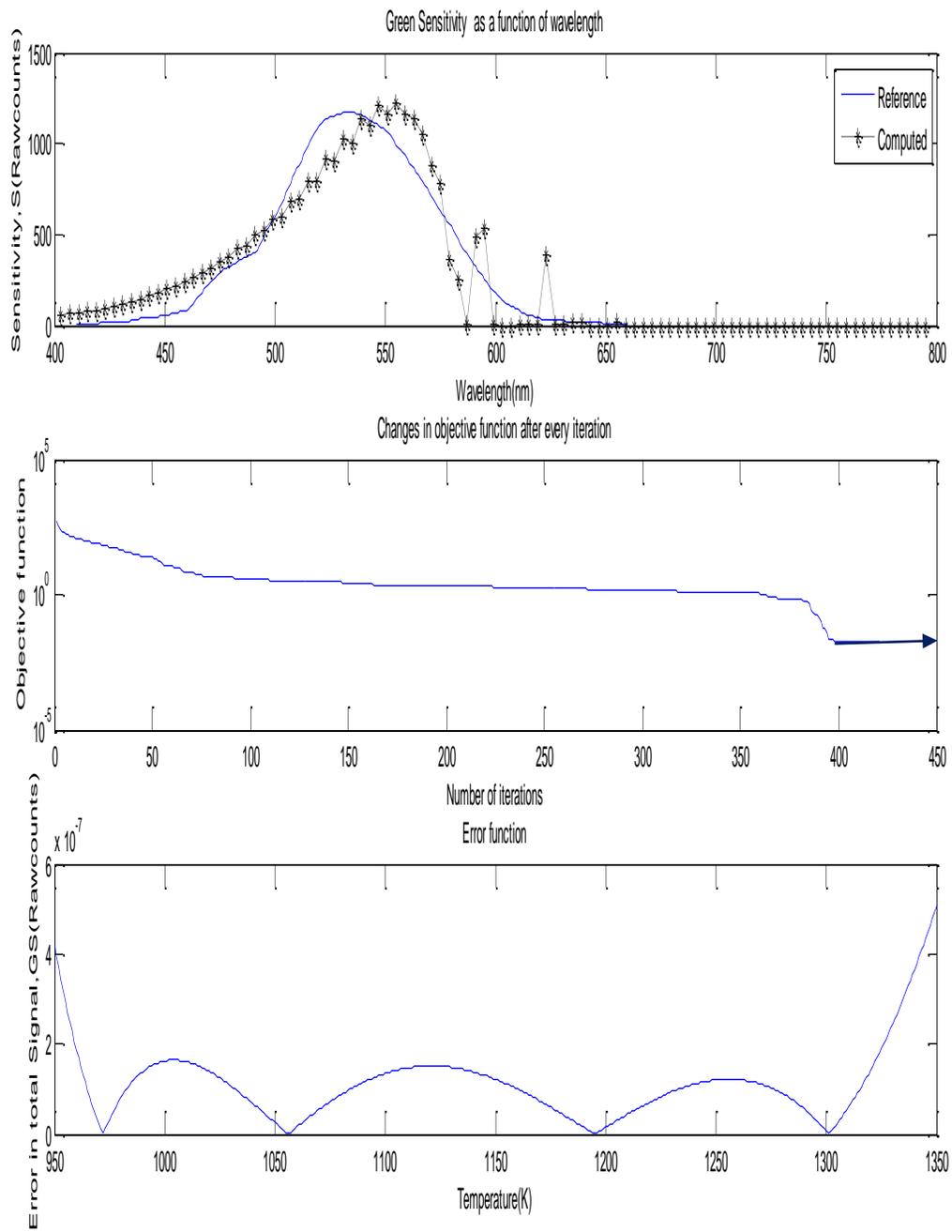


**Figure 3-6a. Calibration curves as published by Calibration Lab at UNIS (Seigernes, et al.,1994)**

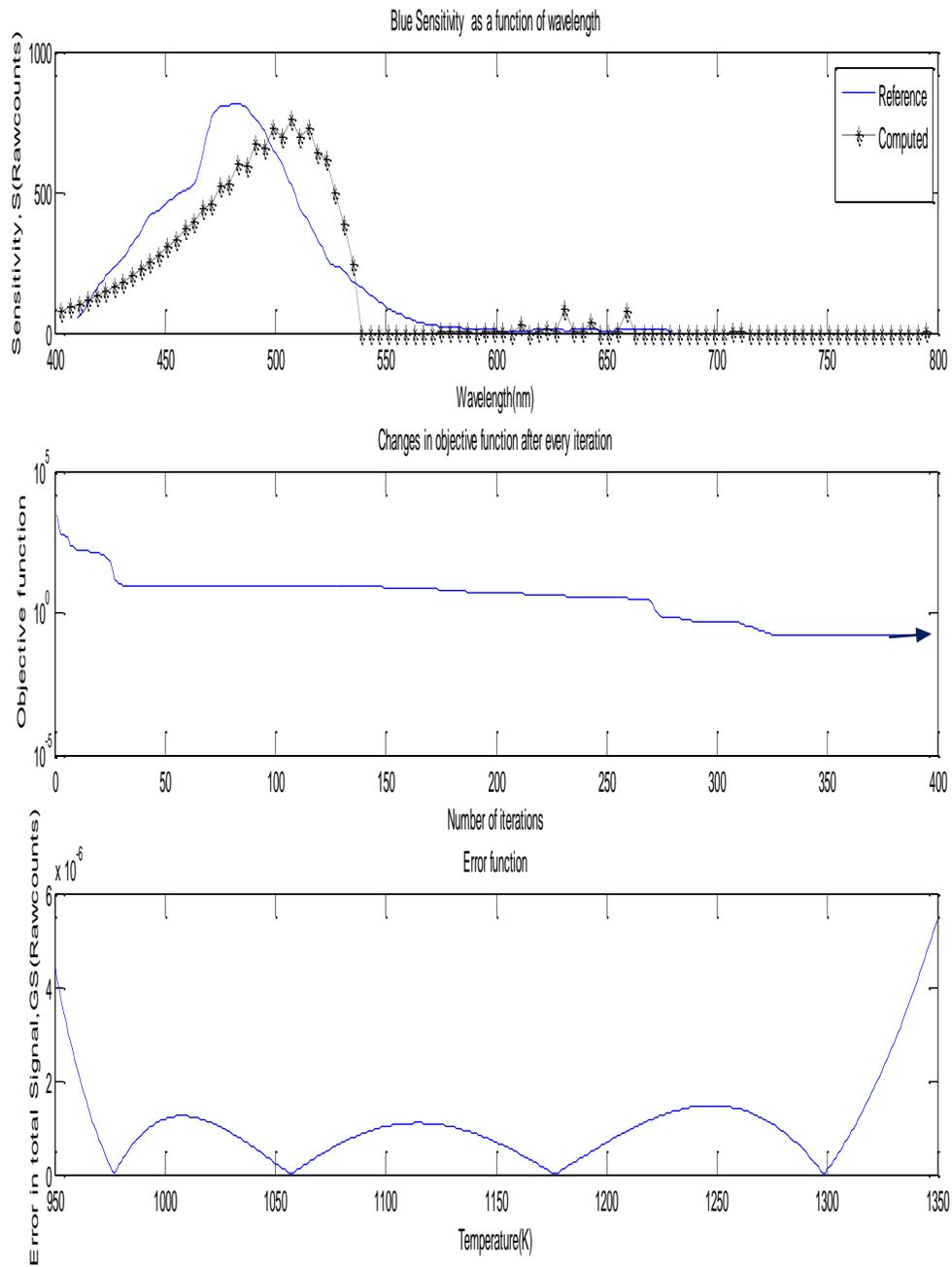
**Figure 3-6b. RGB Signal obtained with sensitivities from fig. 3-6a**



**Figure 3-7. Optimization results for Red color**



**Figure 3-8. Optimization results for Green color**



**Figure 3-9. Optimization results for Blue color**

## Chapter 4 RESULTS AND DISCUSSION

### 4.1. Analysis of experimental results

The experimental data being the source of input to our analytical method, constitutes an important component of the calibration method and controls the level of accuracy reached. Following procedure was followed in selection of samples to be used for the MATLAB optimization program.

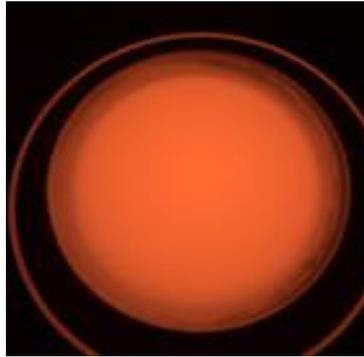


#### 4.1.1. Processing:

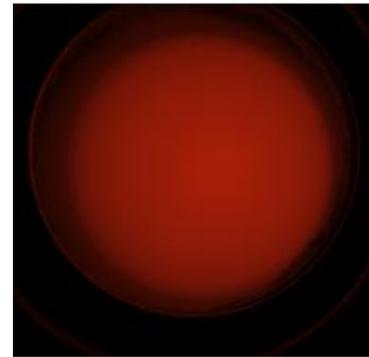
The camera yields files in RAW format that need to be converted to a format recognized by the measurement and analysis software. In our case, the RAW files generated by the camera were processed to '.tiff' format suitable for 'Spotlight'. The second objective in using DCRAW was to take care of the inevitable alteration of digital data of the picture that happens *before* conversion to RAW format. The following Figure 4-1 demonstrates the effect of application of special algorithms to nullify the gamma effect and color transformation effect that are induced in the internal processing of a digital camera.



**Figure 4.1a: Photograph  
processed without correction**



**Figure 4.1b: Photograph  
processed with gamma  
correction but color  
transformation correction not  
applied**

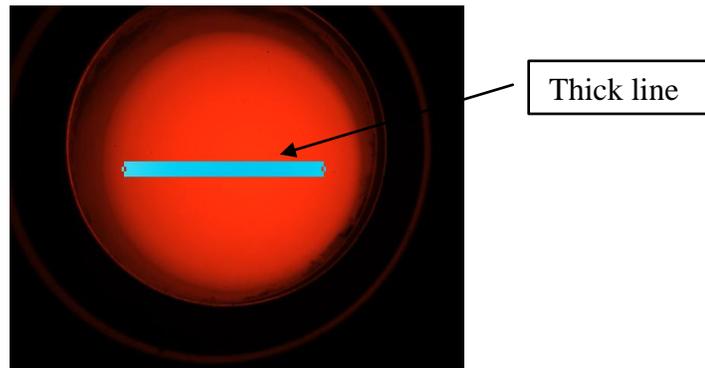


**Figure 4.1c: Photograph  
processed with gamma  
correction and color  
transformation correction**

#### **4.1.2. Signal measurement:**

Software called ‘Spotlight’ developed by NASA was chosen to obtain exact signal values at every pixel on every photograph. Several samples were taken for the purpose of statistical analysis and noise reduction. The ‘Aoi (Area of interest)’ feature in Spotlight allows analysis of a desired fragment of the picture which can be placed at constant position in multiple photographs. This brings about uniformity in all our samples with respect to pixel are picked for analysis. The final images were analyzed with a thick Aoi of line profile with pixel dimensions (20 X 1040) from the whole image (3037 X 2024) as shown in Figure 4.2 .The average of all the values across this linear Aoi was used to represent signal from the blackbody at that particular temperature and shutter speed (for a single pixel).

Good



**Figure 4.2: Measurement Aoi sample size –screenshot from spotlight**

#### 4.1.3. Filtering for removal of bad data

Data that was collected after obtaining the Aoi values for every picture had to be further filtered to select an ideal quality image at every temperature. All the pictures that collected a saturated signal for red were removed. Pictures with a very weak blue signal also had to be removed.

#### 4.1.4. Statistical Analysis

The photographs, after conversion, were analyzed to keep the ones that would make good samples for the final set to be extracted. After choosing an appropriate shutter speed at which non-saturated clear images were obtained , the photographs were processed to reduce noise. The multitude of samples obtained for the same settings gave us freedom to perform noise analysis and eventually dispose values with weak signal or high noise or both. Since, the code needs only about 11 points, noise reduction is crucial for better performance of the optimization algorithm. The following figures show the one sample (out of the four considered) for the final photographs selected for each of the eleven temperatures.



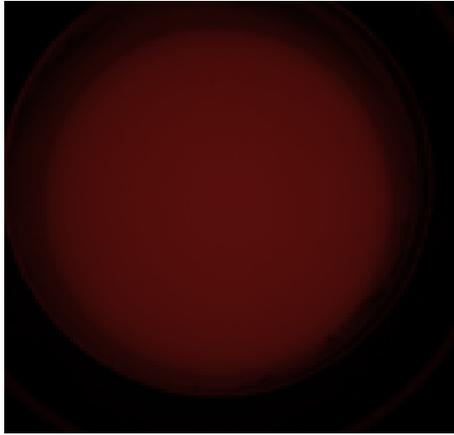
**Figure 4.3.a.:960 K (10 s)**



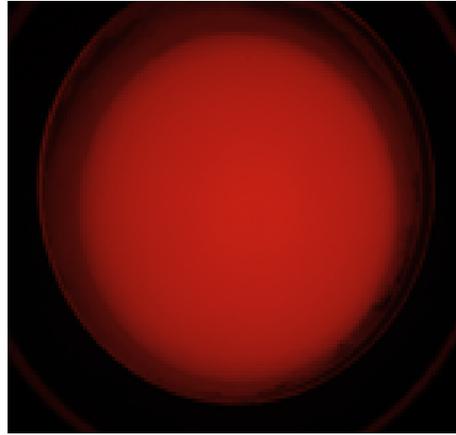
**Figure 4.3.b.:1010 K (10 s)**



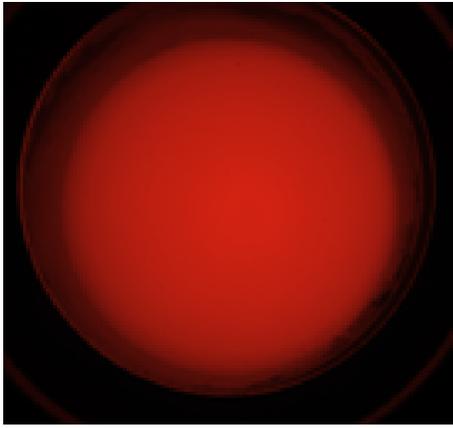
**Figure 4.3.c.:1060 K (6 s)**



**Figure 4.3.d.:1110 K (1 s)**



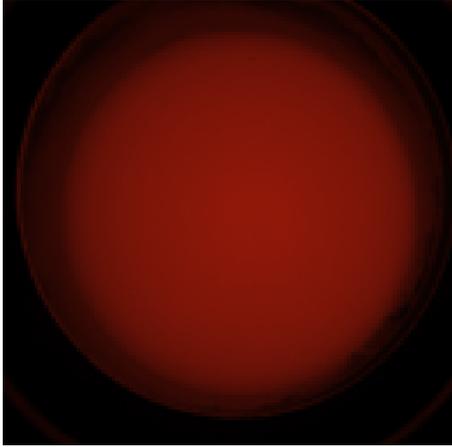
**Figure 4.3.e.:1160 K (1s)**



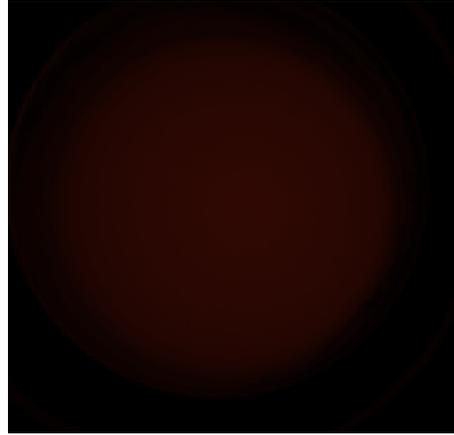
**Figure 4.3.g.:1210 K (0.5 s)**



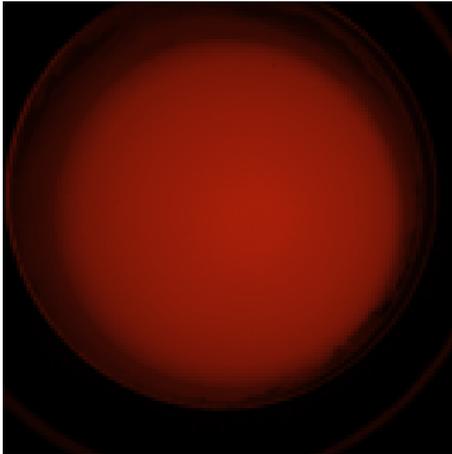
**Figure 4.3.f.:1260 K (0.25 s)**



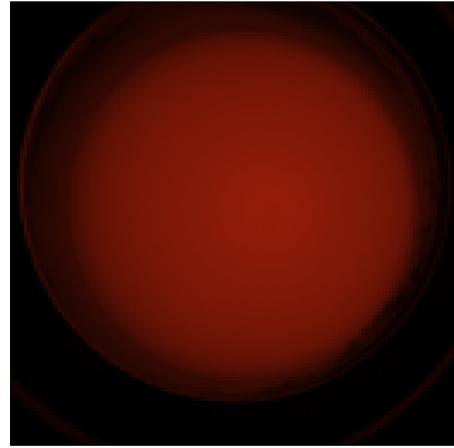
**Figure 4.3.i.:1310 K (0.1 s)**



**Figure 4.3.h.:1360 K (0.01666 s)**



**Figure 4.3.j.:1410 K (0.0333 s)**



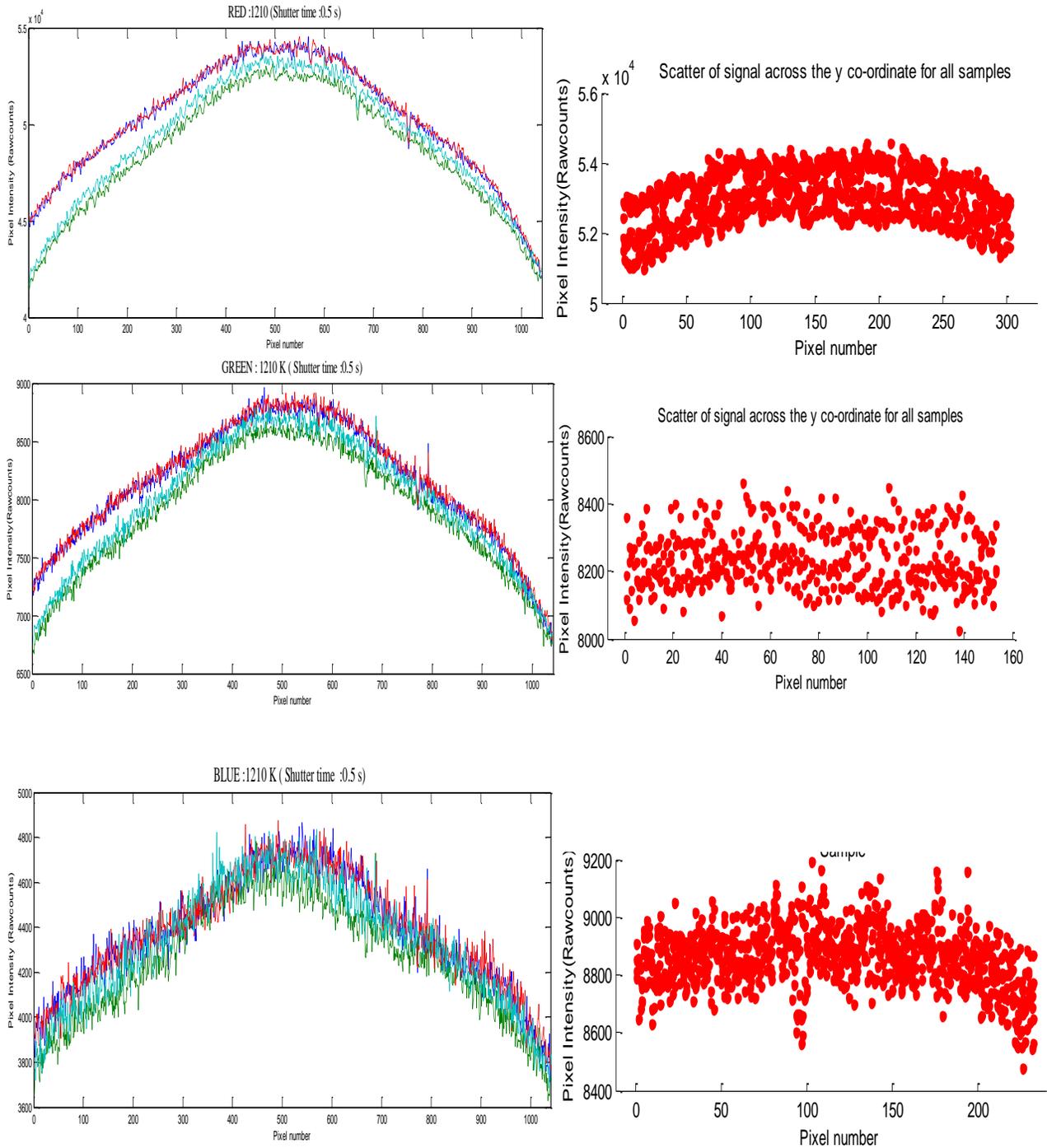
**Figure 4.3.k.:1460 K (0.01666 s)**

**Figures 4.3 (a-k): Photographs of processed samples selected from the experiment**

#### **4.1.4 Noise reduction in sampled data**

Across the Aoi, we observe noise as well as trailing of the signal towards the boundary of the red cavity which is caused by interference with the black edges of the cavity. To filter out this error, the central region with a standard deviation of less than 200 rawcounts was selected to give a noise reduced average value for a pixel for the blackbody at that particular temperature.

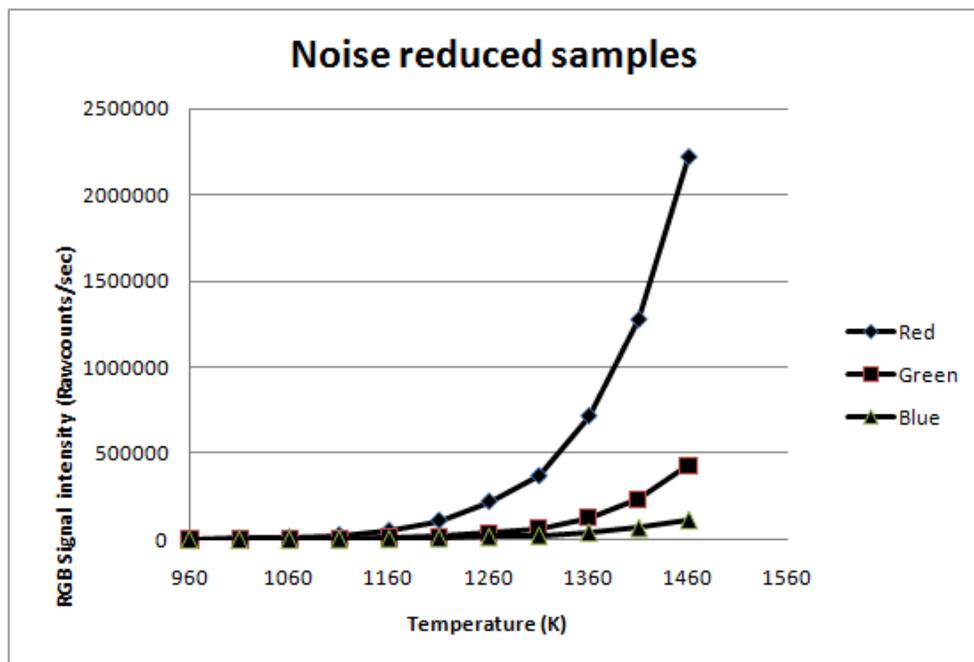
After examining the line profile, the noisy portions of a sample were removed from the pixels considered for estimation of the average value of signal intensity for a pixel. The standard deviation was kept around 1%. The scatter plot demonstrated the nature of noise and deviation in the samples. The best sample was selected from the bunch after evaluation of the standard deviation in each noise filtered. The Figure 4-4 shows the line profile plots of three colors for 1260 K temperature and 0.5 seconds shutter speed. This plot helped selection of an approximately flat centered section that would best represent the signal value at that temperature. This figure shows for an instance for this 1 temperature. The rest of these 33 plots can be elaborately studied from the Appendix D.



**Figure 4-4: Figures showing pixel intensities across a line profile and noise scatter in the multiple samples for 1210K**

#### 4.1.5. Final data after noise reduction:

The above analysis helped choice of the least noise samples for every temperature. Such samples were taken together and values were calculated for signal per second of shutter speed for every temperature. The resultant function of  $GS_i$  vs  $T$  was obtained for ever color plane 'i'. The figure below illustrates this function.



**Figure 4-5: RGB signals for selected samples for 11 temperatures normalized over shutter speed**

## **4.2. Results on optimization**

The three arrays obtained as above formed inputs to the MATLAB Optimization algorithm which is the final phase of our calibration process to generate spectral sensitivity values for the three colors. The plots shown ahead give an estimation of the level of convergence reached with the actual experiment.

### **4.2.1. Results for the three colors for basic settings**

The graphs obtained for different colors under different conditions have been shown. In every figure, the first graph shows the sensitivity function estimate while the second one shows the convergence of the solution in terms of its agreement with the experimentally calculated signal. Though the precision does change with the controlling parameters, in general, a very good agreement is observed in the experimentally measured and function generated values of total signal in the three planes. The errors (absolute and relative) were on the same scale and so was the objective function which proves the uniformity of the mathematical nature of the data collected.

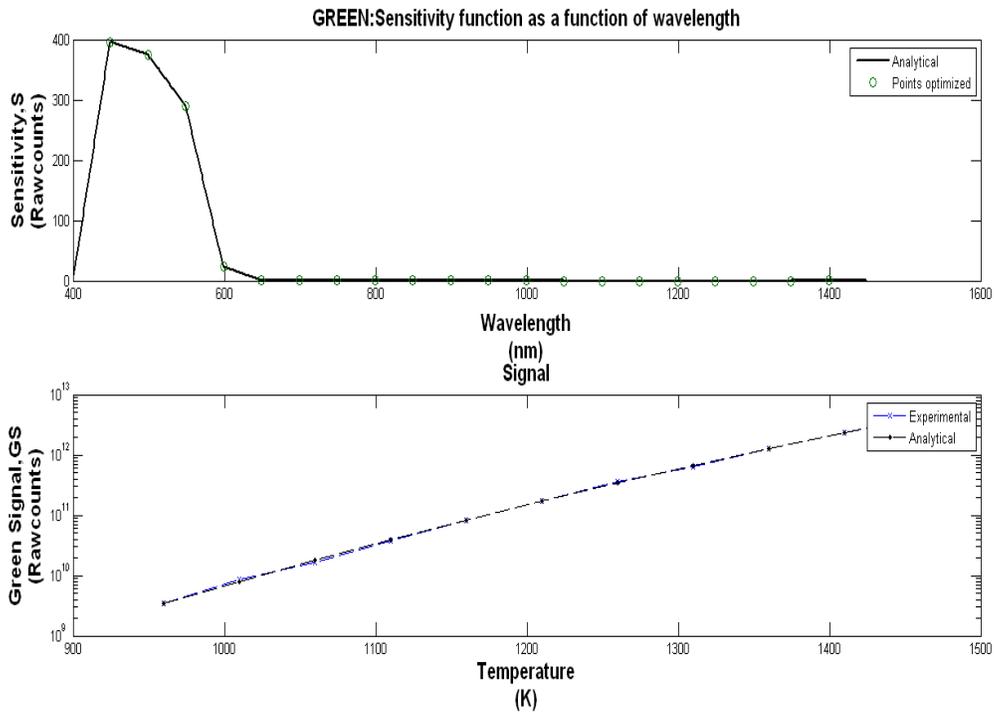


Figure 4-6: Sensitivity function for green color with 20 points across the range 400-1600 nanometers

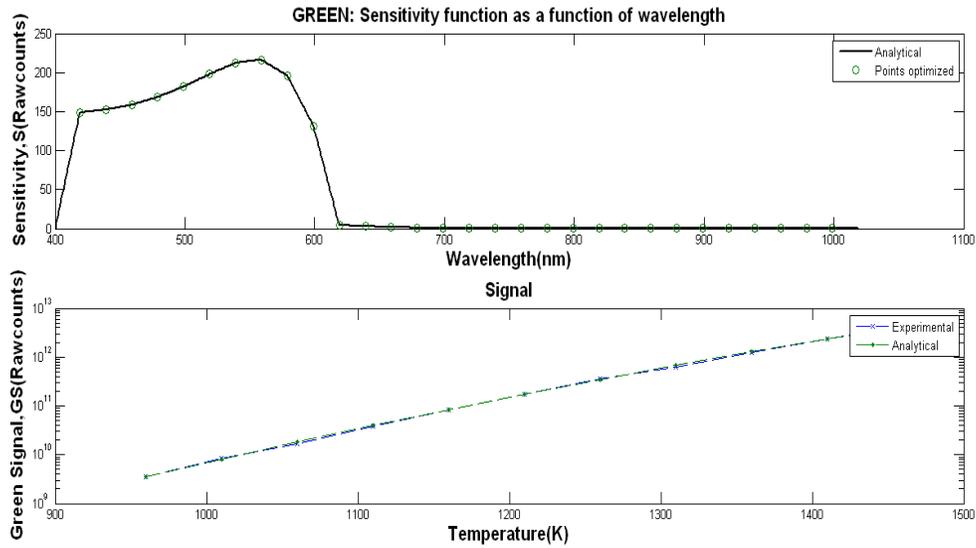
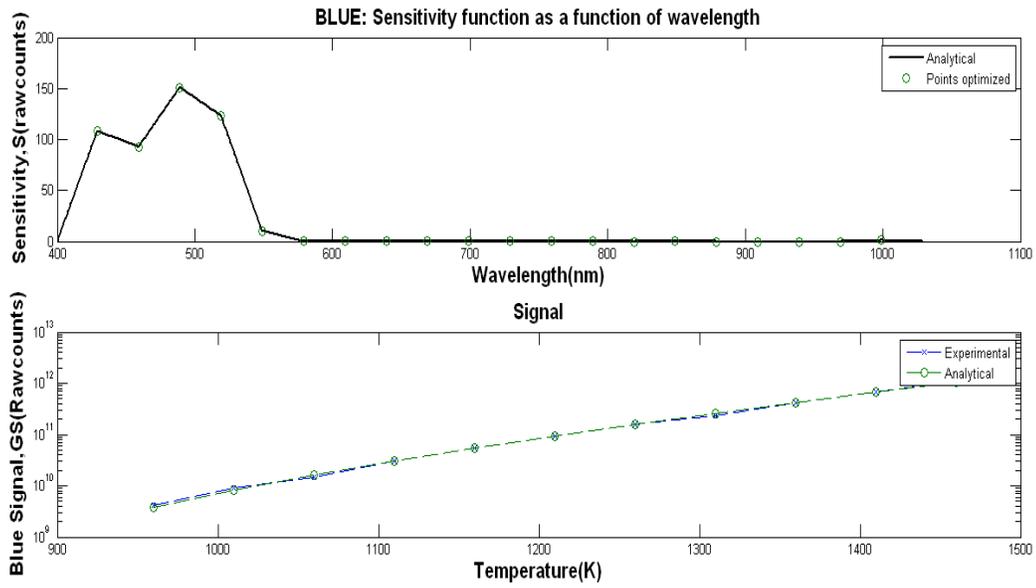
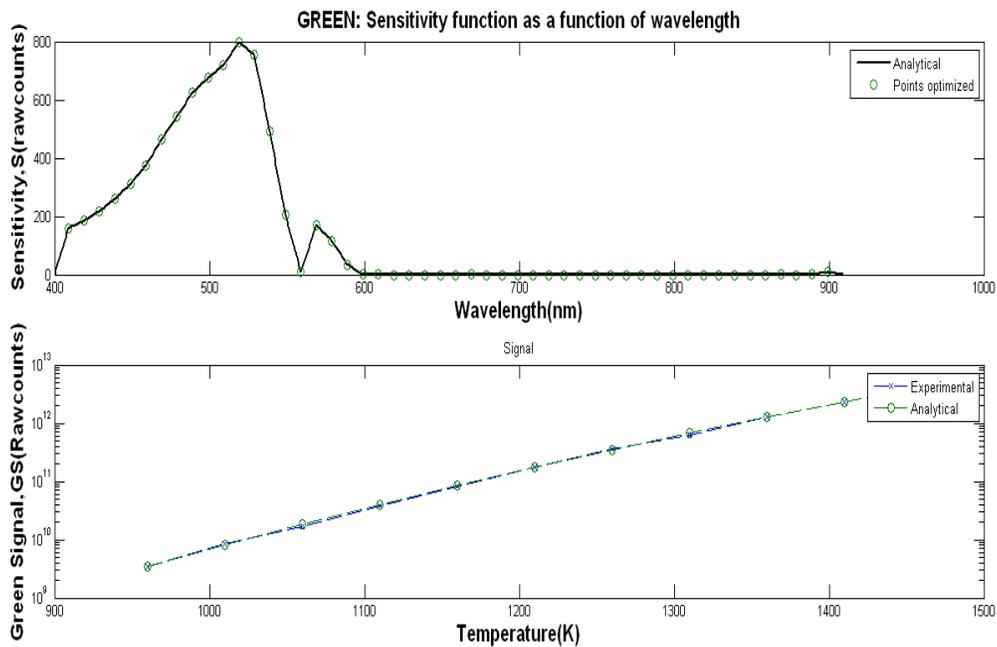


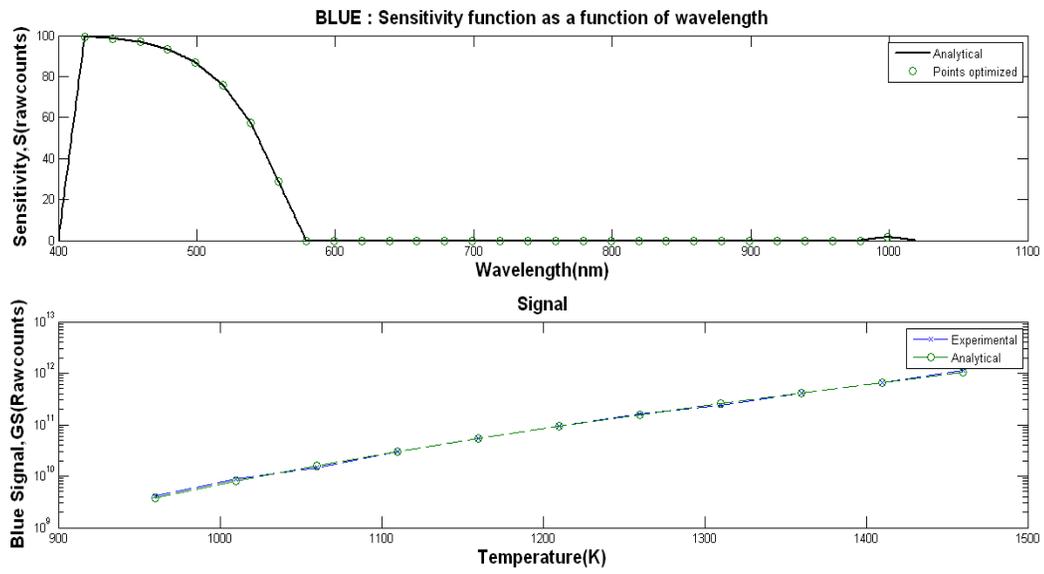
Figure 4-7: Sensitivity function for green color with 30 points across the range 400-1100



**Figure 4-8: Sensitivity function for blue color with 20 points optimized across the range 400-1600 nanometers**



**Figure 4-9: Sensitivity function for green color with 50 points optimized across the range 400-1000 nanometers**



**Figure 4-10: Sensitivity function for blue color with 30 points optimized across the range 400-1100 nanometers**

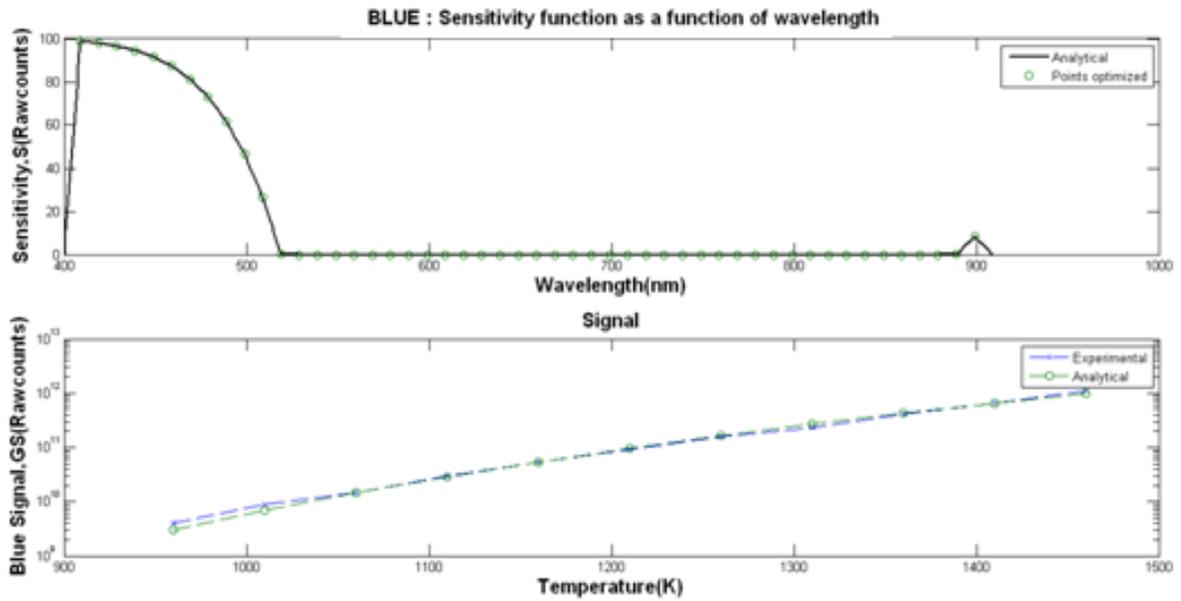


Figure 4-11: Sensitivity function for blue color with 50 points across the range 400-1000 nanometers

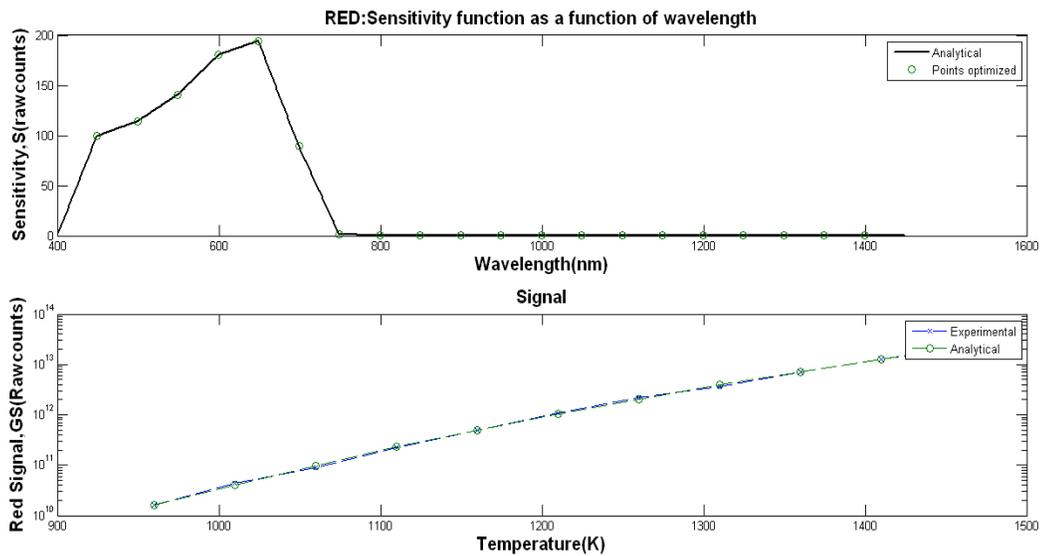


Figure 4-12: Sensitivity function for red color with 20 points across the range 400-1600 nanometers

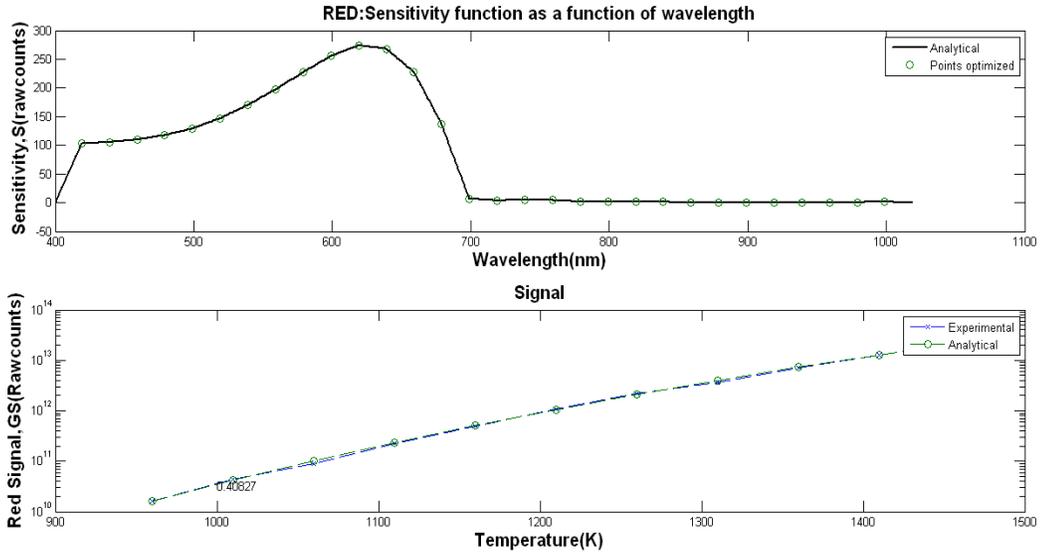


Figure 4-13: Sensitivity function for red color with 30 points across the range 400-1600

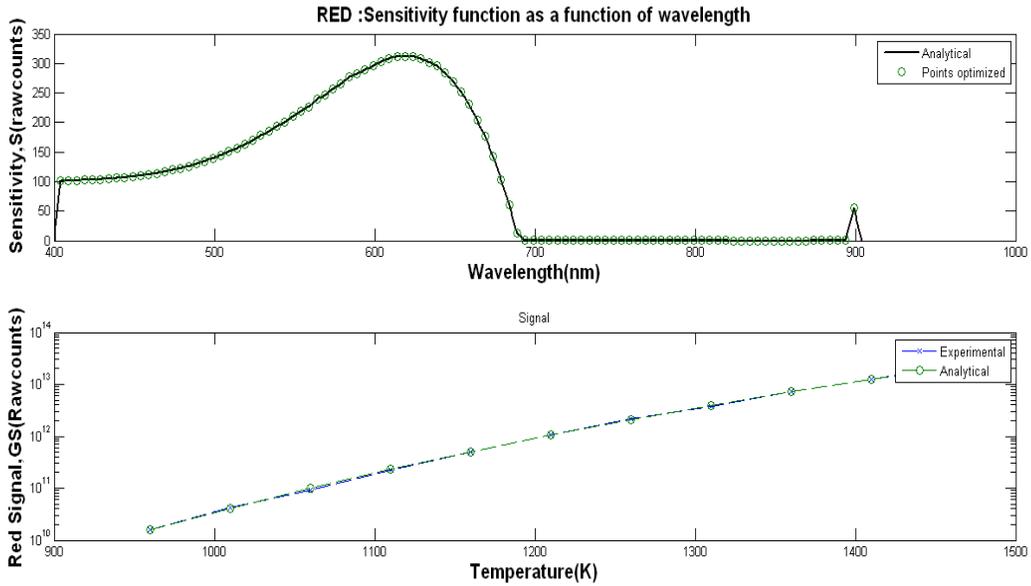
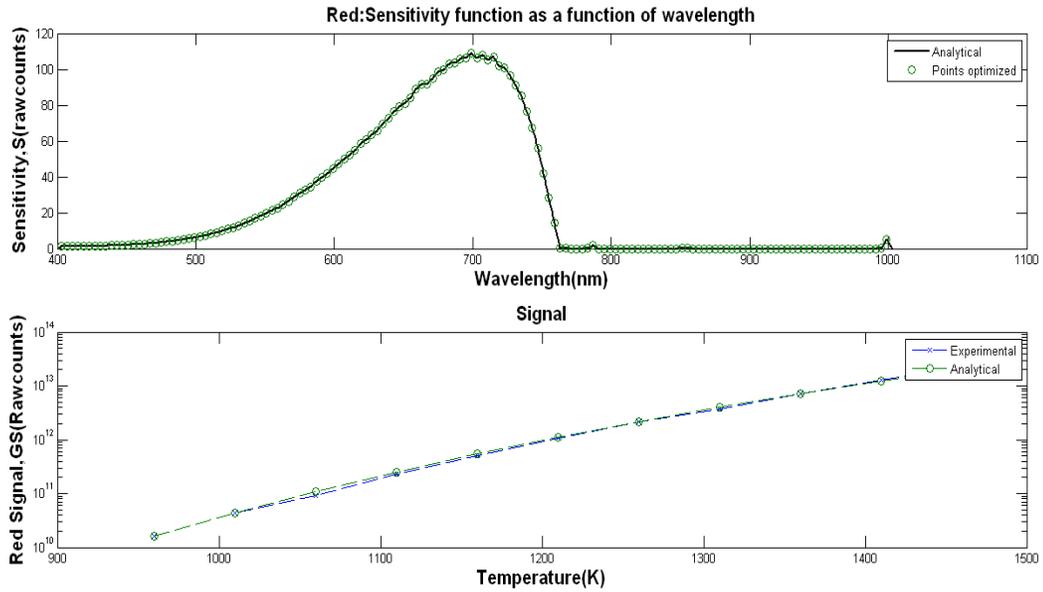


Figure 4-14: Sensitivity function for red color with 100 points across the range 400-1600



**Figure 4-15: Sensitivity function for red color with 150 points across the range 400-1000**

We observe that more points help better definition of the sensitivity curve at the cost of its overall efficiency in reproduction of the original signal values. To sum it up, based on the number of points that are controlled by the algorithm, we get different results for each case and those can be referred to together in the plots below.

#### **4.2.2. Impact of various factors on optimization results**

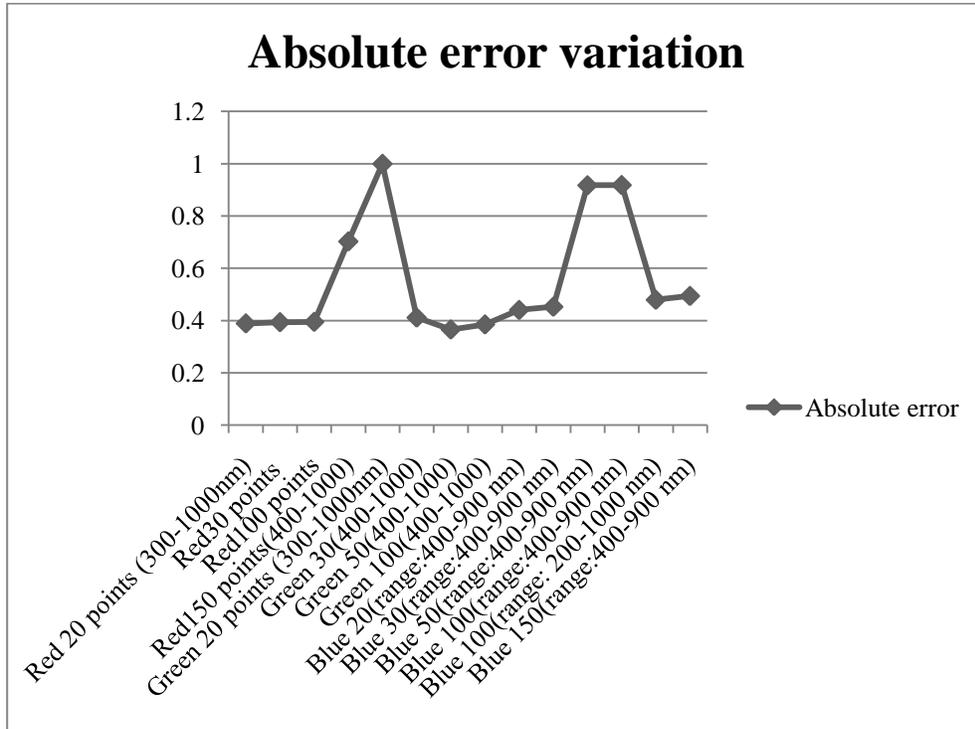
The structure of the program allows for various parameters of the problem to control its performance. For the given arrays, curves for each color were calculated by optimizing different number of points across the whole wavelength. The range of response of the CCD sensors assumed also had a visible effect on the results. Definition of the objective function for the program was another key factor in the nature of results obtained. Program was tested on relative

and absolute error respectively as well as on the sum of both. The results for such tests of every color have been summarized in the table below.

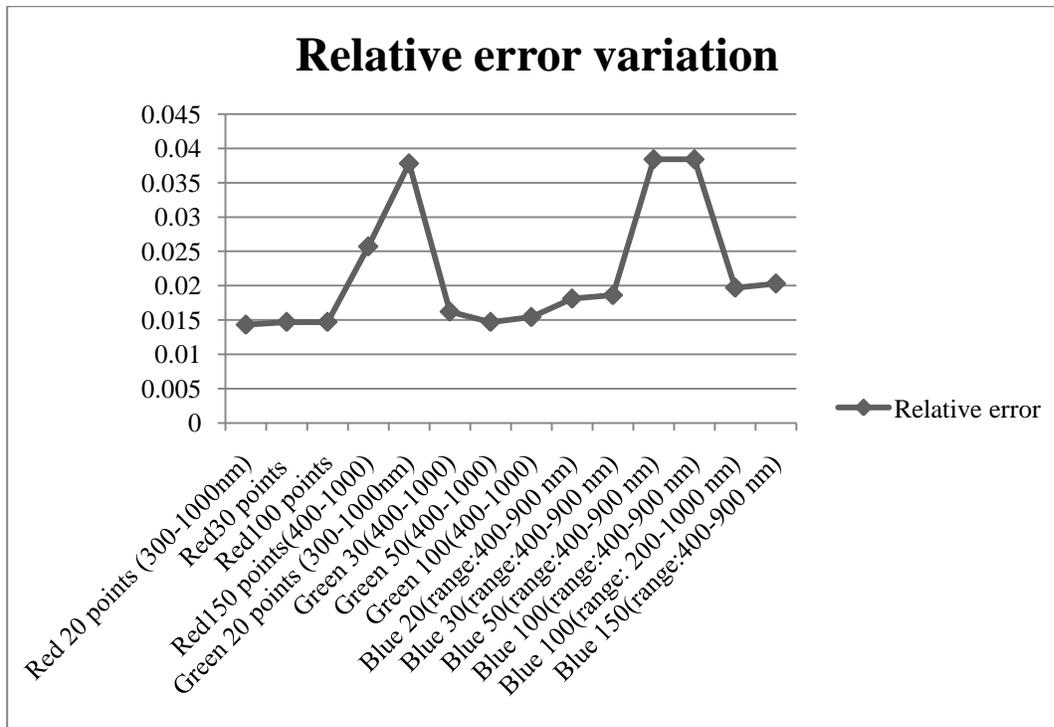
**Table 2: Results of parameter variation on RGB sensitivities**

<b>Color</b>	<b>Number of control points</b>	<b>Absolute Error</b>	<b>Relative Error</b>	<b>Objective function</b>	
				<b>Error</b>	<b>R<sup>2</sup></b>
Red	20	0.389	0.0143	0.4033	0.9996
Red	30	0.3936	0.0147	0.4083	0.9995
Red	50	0.3524	0.0131	0.3655	0.9996
Red	100	0.3948	0.0147	0.4095	0.9995
Green	20	0.9997	0.0378	0.0189	0.9997
Green	30	0.4113	0.0162	0.4275	0.9995
Green	50	0.3657	0.0147	0.3804	0.9995
Green	100	0.385	0.0154	0.4005	0.9995
Blue	20	4.40E-01	0.0181	0.4585	0.999
Blue	30	0.4527	0.0186	0.4713	0.999
Blue	50	0.9175	0.0384	0.9559	0.9944
Blue	100	0.4792	0.0197	0.4989	0.973

With the current set of variation in the Nedler-Mead algorithms available in the toolbox there is a fixed number of points beyond which the accuracy of function calculated goes on decreasing with increasing number of points. For the same parameters, optimization on the sum of absolute and relative error gave better results over those for absolute and relative error separately. Note that there is a lot of ambiguity in the implication of ‘accuracy’. Different researchers in imaging science have considered different error terms to determine accuracy. From Table 1, Figures 4-16 and 4-17, we see that both relative and absolute error show the same trend but their magnitudes being different, their relative impact may vary with the mathematical nature of the function. In this case of blackbody furnace data that we measured, equal weightage to both the errors gave better results in terms of convergence.

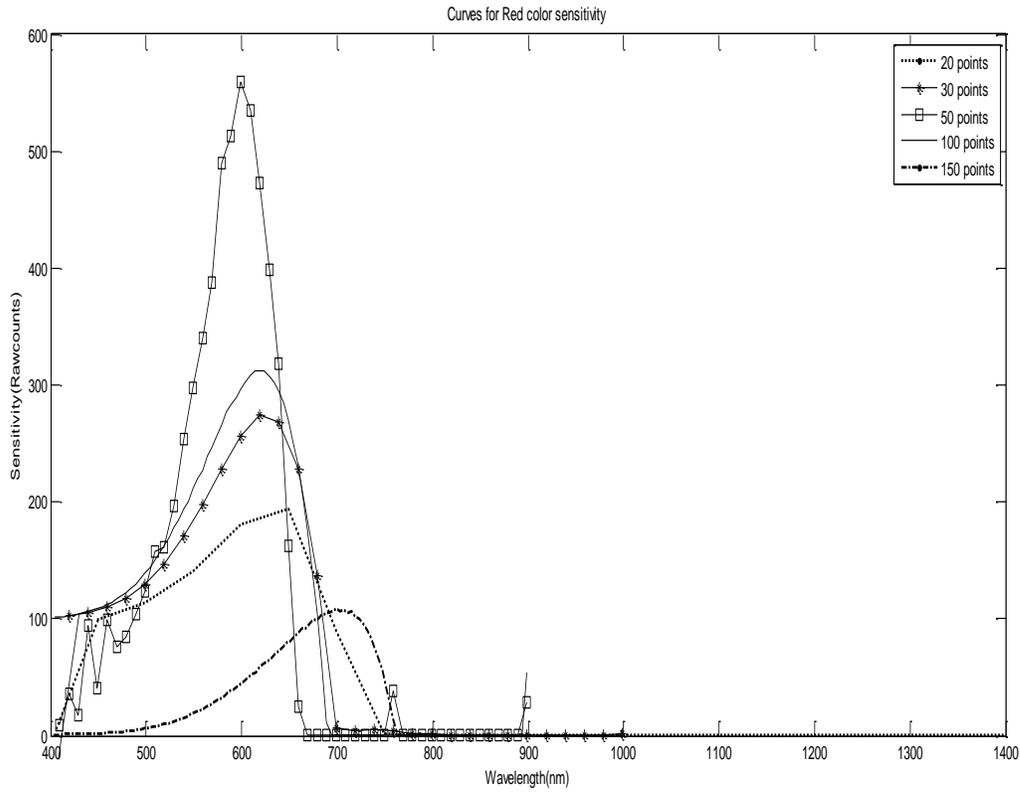


**Figure 4-16: Absolute error variation according to test parameters**

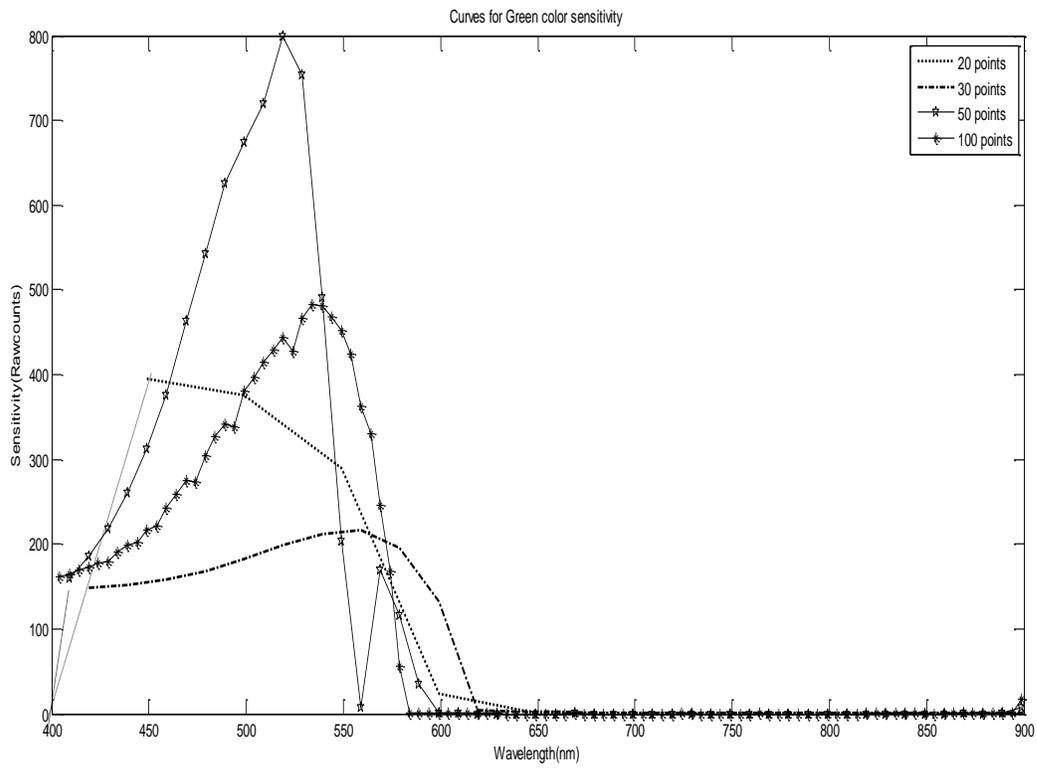


**Figure 4-17: Relative error variation according to test parameters**

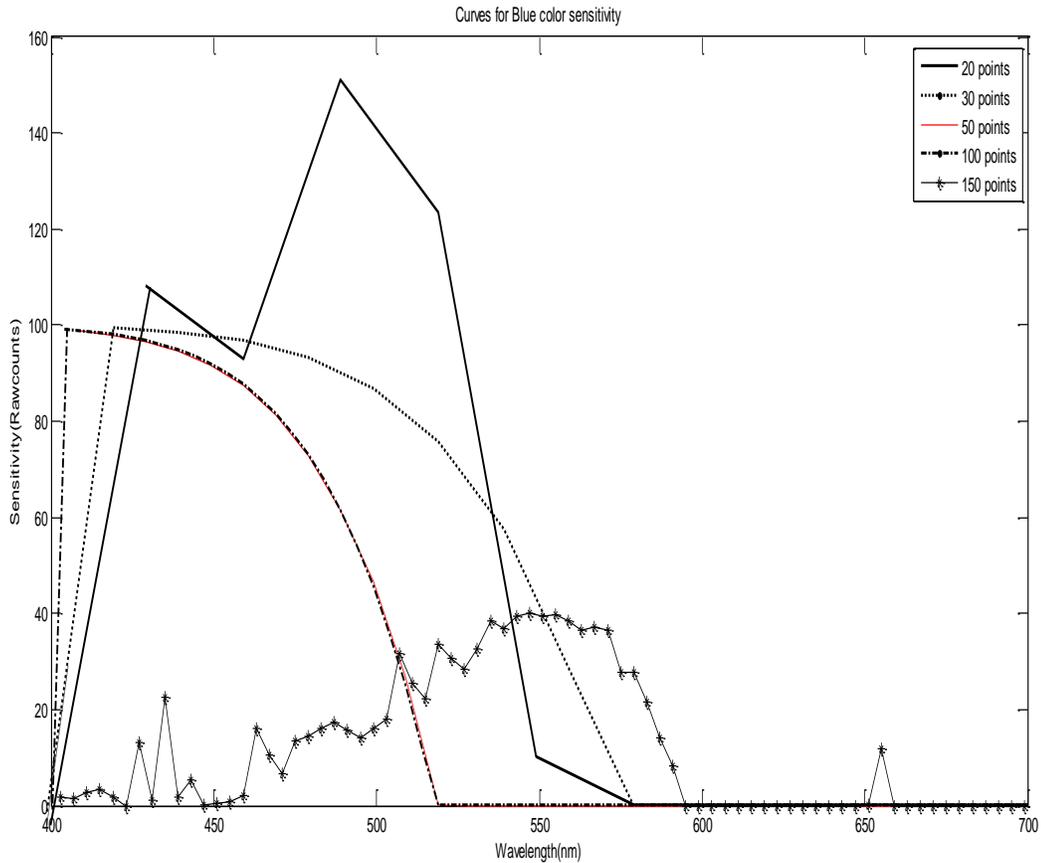
The dependence of number of points controlled by the program to optimize the function values is seen in the graphs below. The objective function reached goes on increasing and investigation needs to be done for the optimal number of points for each case.



**Figure 4-18 Red sensitivity curves for different number of control points**



**Figure 4-19 Green sensitivity curves for different number of control points**



**Figure 4-20 Blue sensitivity curves for different number of control points**

Next, tests were conducted to study the effect of the defined range on the results produced by the algorithm. The results as shown in Figure 4-21 clearly show a strong impact on the function nature with the proximity of range constraint to the actual constraint. The effect in terms of absolute and relative error as well as the objective function value can be seen in the Figure 4-22 that plots the variation in this function with range variation.

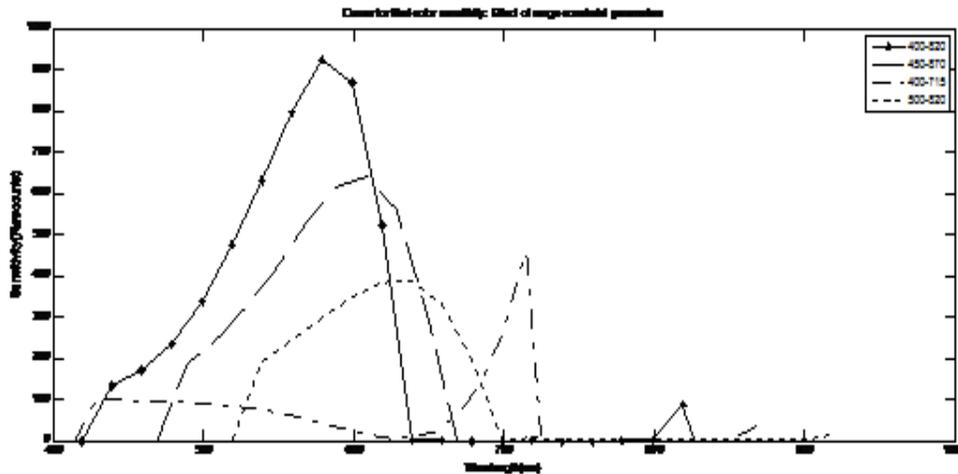


Figure 4-21: Spectral sensitivities for red color as defined on different ranges to see the impact of this constraint on the results

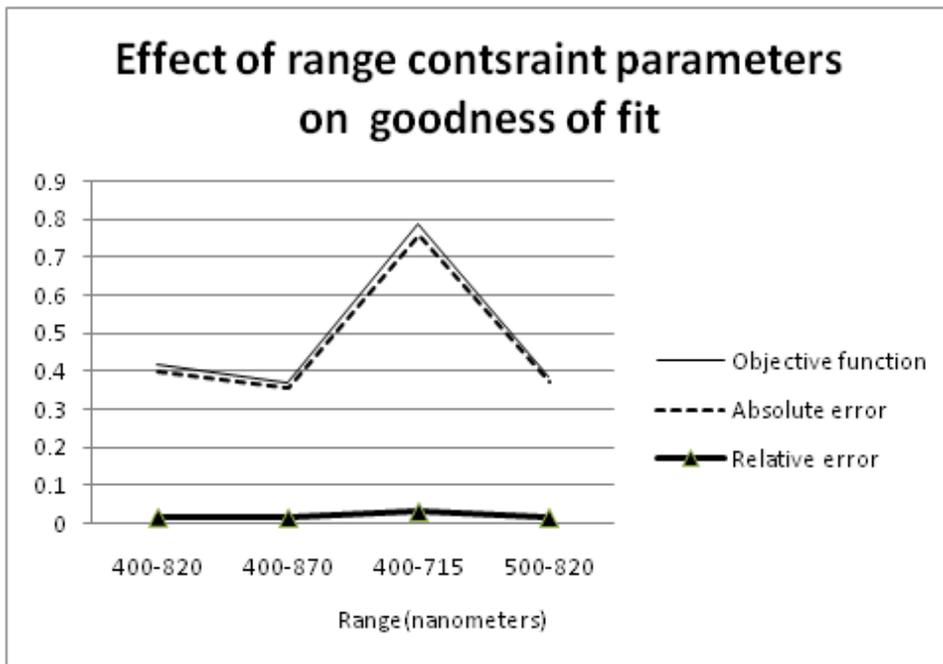
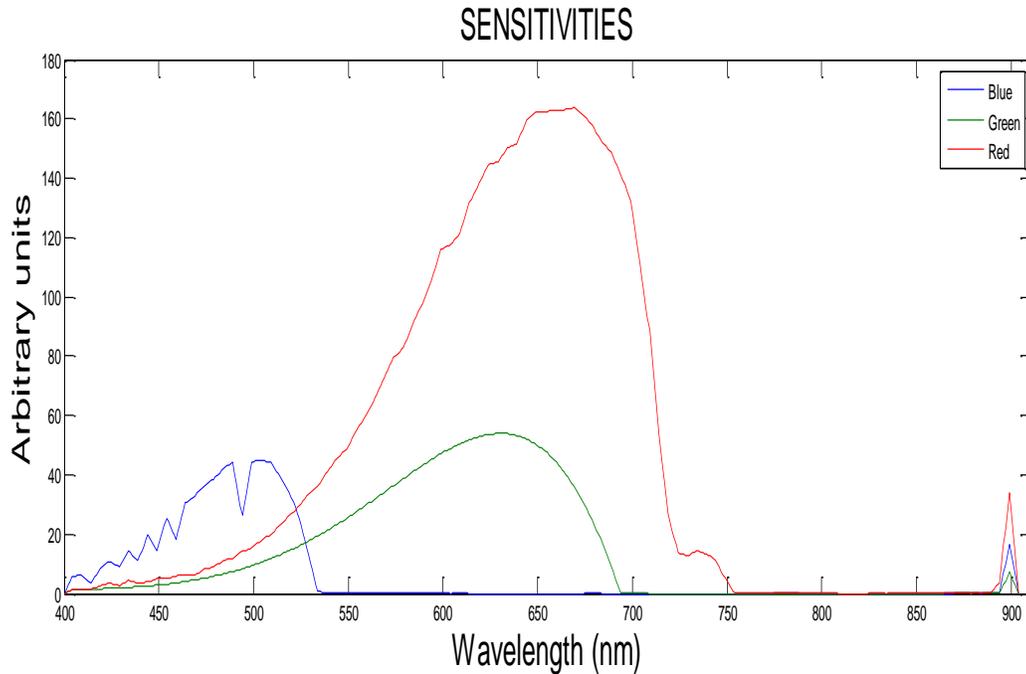


Figure 4-22: Spectral sensitivities for red color as defined on different ranges to see the impact of this constraint on the results

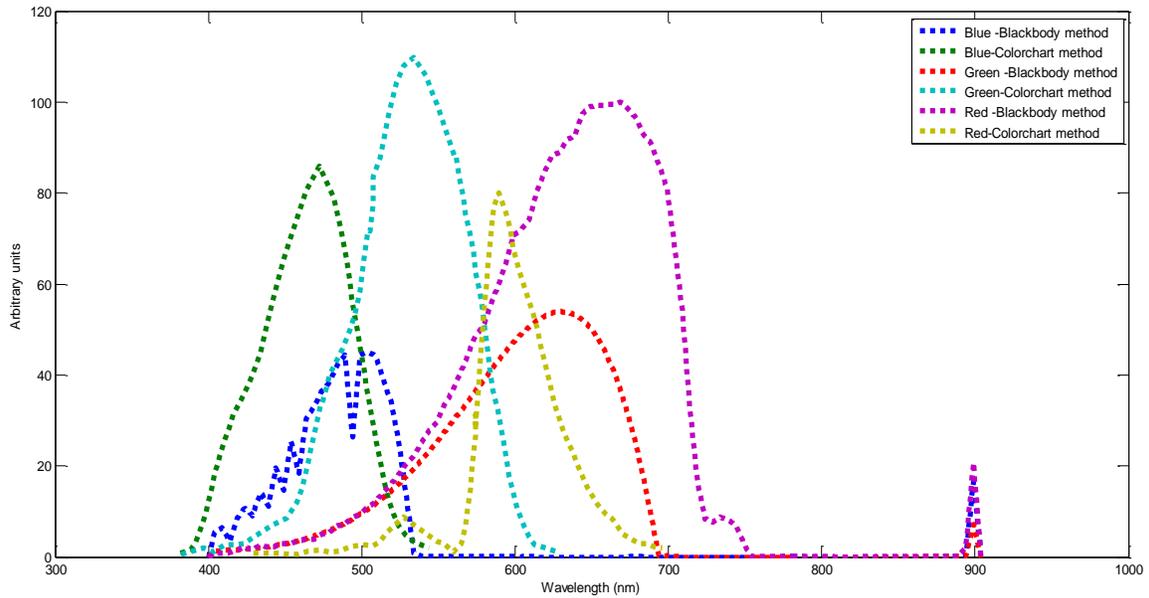


**Figure 4-23: Spectral sensitivities for the three colors as per the optimization function with 100 points defined on each**

#### 4.2.3. Comparison with published Colorchart calibration results

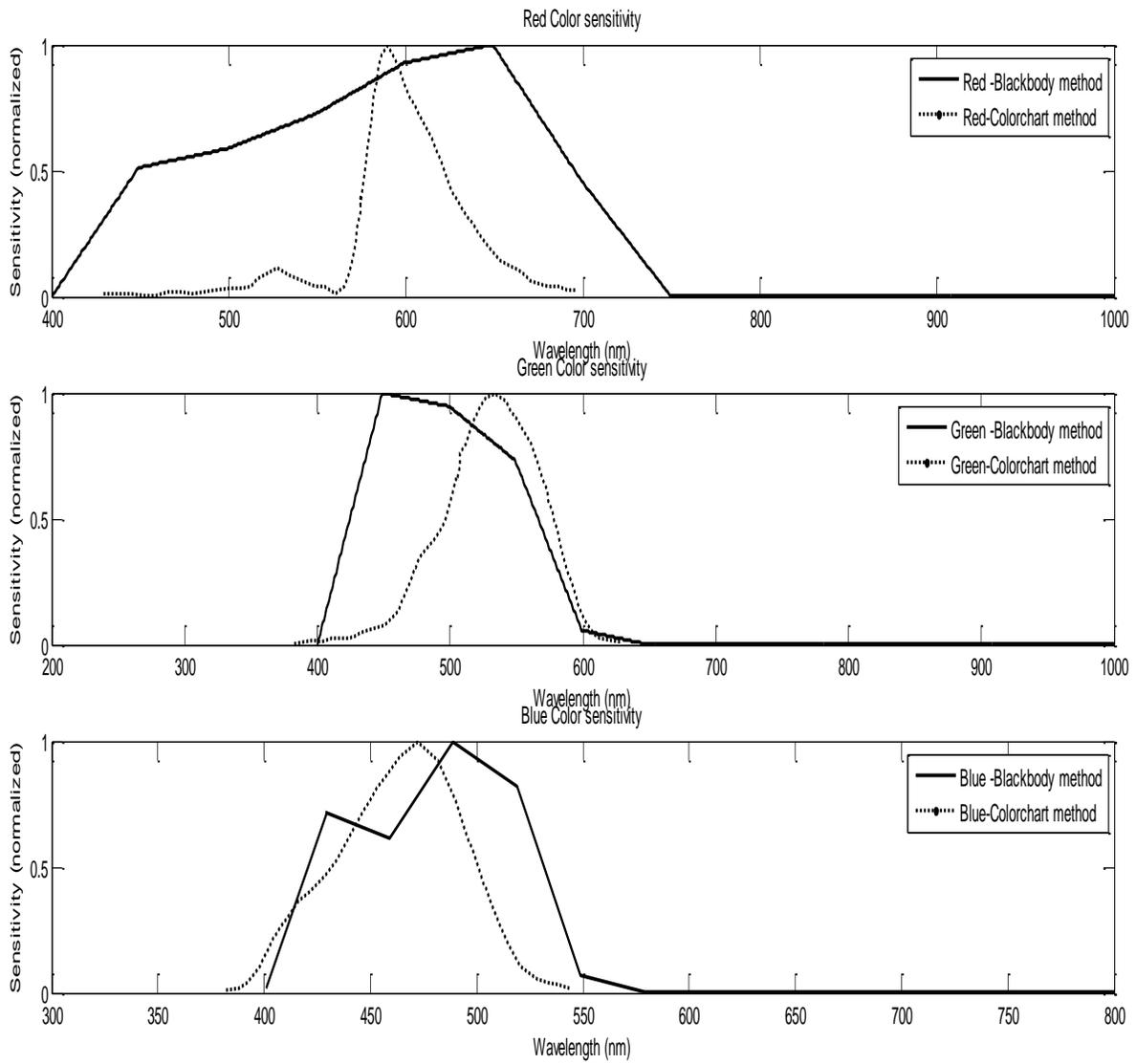
Earlier work with Macbeth color chart for Nikon D100 gives us a reference to compare our results with. On normalizing our results as also the values obtained from an internet source (Peterson & Heukelman, 2010) .

The following set of plots show a comparative analysis between color chart and blackbody calibration method.



**Figure 4-24: Comparison of results from blackbody source to those obtained from MacBeth Colorchart (100 points)**

However, from Table 1 we see that the best objective function values were obtained for 20 points for each of the curves. The plots from the blackbody calibration method were then compared to the published values for each color to get an idea of the agreement of the two sets.



**Figure 4-25: Plots for comparison of the spectral sensitivities for each of the colors computed with published color chart values.**

## **Chapter 5    CONCLUSION AND FUTURE WORK**

From the optimization results and their error analysis, it can be inferred that the calculated signal values obtained from the estimated sensitivity functions agree well with the experimentally observed values for a blackbody furnace with the same conditions and reduced noise. About 100 points can be successfully optimized to give a function (linear between every two of these 100 points) value at every nanometer. The trend has been verified with previous calibration values published which used a color chart.

The main advantage of this method lies in the fact that the values are defined for every nanometer and optimizes for the entire function through control of up to 100 points on the bandwidth for just 11 readings of temperature. This eliminates the need to use Matrix reduction methods. There are no constraints on the number of temperatures in the range of the furnace that could be used for this method. The color chart methods suffer from a major disadvantage of the dependency on illumination as also the agreement of used chart with the CIE standards. This new technique is unaffected by such factors. The only experimental factor is the efficiency of the furnace being used. For applications like space and other subnormal illumination conditions, the validity of sensitivities calculated under D65 illumination is doubtful.

Analytically, this new technique places minimal constraints on the optimization algorithm thus making the calculated values more reasonable mathematically. There are no bounds placed on the modality of the function thus increasing the degree of freedom for the program to choose from a set of feasible values.

On account of the immense progress in the field of mathematical optimization, the future holds a lot of promise for its applications such as the technique presented in this thesis. Thus, refinement in this particular calibration method is possible with more powerful algorithms for higher accuracy in optimization. Noise analysis shows that greater precision in the values obtained may naturally improve the quality of the results produced to some extent. Sophisticated state-of-the-art machinery may facilitate time-averaging of the noise to completely eliminate systematic error making it possible to obtain very high quality virtually noise free data. Although the sensitivity to noise is evidently not so much as it is in the case of matrix reduction methods, noise reduction can help estimate the real accuracy level achieved by particular numerical methods-search algorithm combination .

In the event of the results obtained by testing on the parameters of the optimization problem, there is a high scope for improvement in the analytical portion of this technique by conducting a further detailed study on the exact impact of initial guess choice, wavelength range estimation, objective function definition and search algorithm efficiency. Thus, it can be seen that this calibration method opens up a whole new path in reproduction of digital images that asymptotically reaches the highest level of convergence possible.

## Appendices

Appendix A: Details of the blackbody furnace used.

Model 67032	Type Blackbodies
Specifications	
Calibration	$\pm 0.2\text{ }^{\circ}\text{C} \pm 1\text{ digit}$
Stability	$\pm 0.02\%$ of full scale per 24-hour period
Resolution	$1\text{ }^{\circ}\text{C}$ or $0.1\text{ }^{\circ}\text{C}$ , selectable
Warm-up Time (1.0 inch cavity models)	35 minutes (ambient to $1050/1200\text{ }^{\circ}\text{C}$ )
Warm-up Time (0.25 and 0.4 inch cavity models)	15 minutes (ambient to $1050\text{ }^{\circ}\text{C}$ )
Sensing Element	Thermocouple, Type S (Plat/Plat 10% Rhodium) special 0.01% tolerance
Cavity Type	Recessed $20^{\circ}$ cone
Cavity Emissivity	$0.99 \pm 0.01\%$
Calibration T/C	Type S special 0.01% tolerance, matched to sensing T/C
Housing Temperature (1	$<15\text{ }^{\circ}\text{C}$ above ambient @ $1050\text{ }^{\circ}$

inch cavity models)

## Appendix B: Details of the camera used

Resolution	6.1 Megapixel
Color Support	Color Optical
Sensor Type	CCD
Total Pixels	6,310,000 pixels
Effective Sensor Resolution	6,100,000 pixels
Optical Sensor Size	15.6 x 23.7mm

## Appendix C: MATLAB Code

```
%Optimized function

function f=blueset(g)

lower=400;%Lower limit of the bandwidth considered and defining point for the

%range

num= acalc;% Wavelength of the first point where thefunction is caluclated
p=101;%Number of points optimized and used to set the rest of the sensitivity %function

d=5;%Number of points in every division of the range
pr=p-1;%Number of divisions of the range

for j=1:1:p;%Loop for every setting wavelength values in every division
    for i=1:1:d;%Loop for setting value for wavelength for each point in a division
        l(j,i)=num;% Value at first point defined from the range chosen
        num=num+1; %Incrementation for value to be set at the next point
    end %End of loop for same division points
end %End of loop for all divisions

%Catentation to create a continuous function of wavelengths

rl=l(1,1:end);%Initial value of the catenated variable set as the first division

for i=2:1:p;
    rl=horzcat(rl,l(i,1:end));
end
% rl is now a continuous wavelength function
%Slope for the first division (first point assumed to have 0 intensity)

slope(1)=g(1)/(l(1,d)-l(1,1));
%Slope for all other divisions except last
for k=2:1:pr
    slope(k)=(g(k)-g(k-1))/(l(k,d)-l((k-1),d));
end
%Slope for last division with last point assumed at 0 intensity
slope(p)=-1*(g(pr)/(l(p,d)-l(pr,d)));

%Setting the function for first division

for j=1:1:d
    s(1,j)=slope(1).*(l(1,j)-acalc);
end

%Setting the function for all divisions
for i=2:1:p
    for h=1:1:d
        s(i,h)=(slope(i).*(l(i,h)-l(i-1,+d)))+s(i-1,d);
    end
end
end
```

```

%Concatenation to create a continuous function
x=s(1,1:end);
for i=2:1:p;
x=horzcat(x,s(i,1:end));
end
%Experimental values for the 11 temperatures
rr=[410.150495 891.849505 1469.59571 2977.638614 5400.985149 9360.356436 16121.80198 23406.18812
41170.69307 66441.5346 111290.495];
rr=1e7.*rr;
T=[960 1010 1060 1110 1160 1210 1260 1310 1360 1410 1460];
arl=rl.*(10^-9);%Conversion of wavelength to metres
lme5=arl.^-5;
lre=arl.^-1;
w=size(T);
length=w(1,2);%Variable to store size of the temperature array

load c1 %Constants from Planck's Law stored elsewhere
load c2
%Calculation of total signal with estimated function

for count=1:1:length
t=(T(1,count)^-1);
pow=c2.*t.*lre;
den=(exp(pow))-1;
rden=den.^-1;
int=c1.*lme5.*rden;%Intensity calculated
pr1=int.*x;
ts(1,count)=trapz(rl,pr1);%Signal calculated with numerical integration of %Planck's Law expression
end

subtraction= abs(log(rr)-log(ts))./log(rr)+abs(log(rr)-log(ts));%Term for %objective function

%Objective function

f=(sum(subtraction));

%Plotting functions

%gg=l(1:pr,d);
%
% subplot(2,1,1);
% plot(rl,x,gg,g,'o');
% xlabel('Wavelength(nm)')
% ylabel('Sensitivity,S(rawcounts)')
% legend('Analytical','Points optimized')
% title('Sensitivity function as a function of wavelength')
% subplot(2,1,2);
% plot(T,rr,'--x',T,ts,'--o')
% xlabel('Temperature(K)')
% ylabel('Green Signal,GS(Rawcounts)')
% title('Signal')
% legend('Experimental','Analytical')
% vv= num2str(f);

```

```

% text(1000,9^11,vv)

MAIN FUNCTION TO CALL THE OPTIMIZATION TOOL
s0=ones(1,40)
options=optimset('Display','iter','MaxIter',10000,'MaxFunEvals'TolFun',1e-500,'TolX',1e-30)
%Setting options for the optimization
S=fminsearch(@blueset,s0, [],[],[],[],lb,ub,[],
options)

FMINSEARCH USED BY MATLAB
function [x,fval,exitflag,output] = fminsearch(funfcn,x,options,varargin)
%FMINSEARCH Multidimensional unconstrained nonlinear minimization (Nelder-Mead).
% X = FMINSEARCH(FUN,X0) starts at X0 and attempts to find a local minimizer
% X of the function FUN. FUN is a function handle. FUN accepts input X and
% returns a scalar function value F evaluated at X. X0 can be a scalar, vector
% or matrix.
%
% X = FMINSEARCH(FUN,X0,OPTIONS) minimizes with the default optimization
% parameters replaced by values in the structure OPTIONS, created
% with the OPTIMSET function. See OPTIMSET for details. FMINSEARCH uses
% these options: Display, TolX, TolFun, MaxFunEvals, MaxIter, FunValCheck,
% PlotFcns, and OutputFcn.
%
% X = FMINSEARCH(PROBLEM) finds the minimum for PROBLEM. PROBLEM is a
% structure with the function FUN in PROBLEM.objective, the start point
% in PROBLEM.x0, the options structure in PROBLEM.options, and solver
% name 'fminsearch' in PROBLEM.solver. The PROBLEM structure must have
% all the fields.
%
% [X,FVAL]= FMINSEARCH(...) returns the value of the objective function,
% described in FUN, at X.
%
% [X,FVAL,EXITFLAG] = FMINSEARCH(...) returns an EXITFLAG that describes
% the exit condition of FMINSEARCH. Possible values of EXITFLAG and the
% corresponding exit conditions are
%
% 1 Maximum coordinate difference between current best point and other
% points in simplex is less than or equal to TolX, and corresponding
% difference in function values is less than or equal to TolFun.
% 0 Maximum number of function evaluations or iterations reached.
% -1 Algorithm terminated by the output function.
%
% [X,FVAL,EXITFLAG,OUTPUT] = FMINSEARCH(...) returns a structure
% OUTPUT with the number of iterations taken in OUTPUT.iterations, the
% number of function evaluations in OUTPUT.funcCount, the algorithm name
% in OUTPUT.algorithm, and the exit message in OUTPUT.message.
%
% Examples
% FUN can be specified using @:
% X = fminsearch(@sin,3)
% finds a minimum of the SIN function near 3.
% In this case, SIN is a function that returns a scalar function value
% SIN evaluated at X.
%
% FUN can also be an anonymous function:
% X = fminsearch(@(x) norm(x),[1;2;3])

```

```

% returns a point near the minimizer [0;0;0].
%
% If FUN is parameterized, you can use anonymous functions to capture the
% problem-dependent parameters. Suppose you want to optimize the objective
% given in the function myfun, which is parameterized by its second argument c.
% Here myfun is an M-file function such as
%
% function f = myfun(x,c)
% f = x(1)^2 + c*x(2)^2;
%
% To optimize for a specific value of c, first assign the value to c. Then
% create a one-argument anonymous function that captures that value of c
% and calls myfun with two arguments. Finally, pass this anonymous function
% to FMINSEARCH:
%
% c = 1.5; % define parameter first
% x = fminsearch(@(x) myfun(x,c),[0.3;1])
%
% FMINSEARCH uses the Nelder-Mead simplex (direct search) method.
%
% See also OPTIMSET, FMINBND, FUNCTION_HANDLE.

% Reference: Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright,
% Paul E. Wright, "Convergence Properties of the Nelder-Mead Simplex
% Method in Low Dimensions", SIAM Journal of Optimization, 9(1):
% p.112-147, 1998.

% Copyright 1984-2007 The MathWorks, Inc.
% $Revision: 1.21.4.16 $ $Date: 2008/10/31 06:19:57 $

defaultopt = struct('Display','notify','MaxIter','200*numberOfVariables',...
    'MaxFunEvals','200*numberOfVariables','TolX',1e-4,'TolFun',1e-4, ...
    'FunValCheck','off','OutputFcn',[],'PlotFcns',[]);

% If just 'defaults' passed in, return the default options in X
if nargin==1 && nargin <= 1 && isequal(funcfn,'defaults')
    x = defaultopt;
    return
end

if nargin<3, options = []; end

% Detect problem structure input
if nargin == 1
    if isa(funcfn,'struct')
        [funcfn,x,options] = separateOptimStruct(funcfn);
    else % Single input and non-structure
        error('MATLAB:fminsearch:InputArg','The input to FMINSEARCH should be either a structure with valid
fields or consist of at least two arguments.');
```

```

    error('MATLAB:fminsearch:NotEnoughInputs',...
        'FMINSEARCH requires at least two input arguments');
end

% Check for non-double inputs
if ~isa(x,'double')
    error('MATLAB:fminsearch:NonDoubleInput', ...
        'FMINSEARCH only accepts inputs of data type double.')
end

n = numel(x);
numberOfVariables = n;

prntype = optimget(options,'Display',defaultopt,'fast');
tolx = optimget(options,'TolX',defaultopt,'fast');
tolf = optimget(options,'TolFun',defaultopt,'fast');
maxfun = optimget(options,'MaxFunEvals',defaultopt,'fast');
maxiter = optimget(options,'MaxIter',defaultopt,'fast');
funValCheck = strcmp(optimget(options,'FunValCheck',defaultopt,'fast'),'on');

% In case the defaults were gathered from calling: optimset('fminsearch'):
if ischar(maxfun)
    if isequal(lower(maxfun),'200*numberOfvariables')
        maxfun = 200*numberOfVariables;
    else
        error('MATLAB:fminsearch:OptMaxFunEvalsNotInteger',...
            'Option "MaxFunEvals" must be an integer value if not the default.')
    end
end
if ischar(maxiter)
    if isequal(lower(maxiter),'200*numberOfvariables')
        maxiter = 200*numberOfVariables;
    else
        error('MATLAB:fminsearch:OptMaxIterNotInteger',...
            'Option "MaxIter" must be an integer value if not the default.')
    end
end

switch prntype
    case {'notify','notify-detailed'}
        prnt = 1;
    case {'none','off'}
        prnt = 0;
    case {'iter','iter-detailed'}
        prnt = 3;
    case {'final','final-detailed'}
        prnt = 2;
    case 'simplex'
        prnt = 4;
    otherwise
        prnt = 1;
end
% Handle the output
outputfcn = optimget(options,'OutputFcn',defaultopt,'fast');

```

```

if isempty(outputfcn)
    haveoutputfcn = false;
else
    haveoutputfcn = true;
    xOutputfcn = x; % Last x passed to outputfcn; has the input x's shape
    % Parse OutputFcn which is needed to support cell array syntax for OutputFcn.
    outputfcn = createCellArrayOfFunctions(outputfcn,'OutputFcn');
end

% Handle the plot
plotfcns = optimget(options,'PlotFcns',defaultopt,'fast');
if isempty(plotfcns)
    haveplotfcn = false;
else
    haveplotfcn = true;
    xOutputfcn = x; % Last x passed to plotfcns; has the input x's shape
    % Parse PlotFcns which is needed to support cell array syntax for PlotFcns.
    plotfcns = createCellArrayOfFunctions(plotfcns,'PlotFcns');
end

header = 'Iteration  Func-count  min f(x)      Procedure';

% Convert to function handle as needed.
funfcn = fcnchk(funfcn,length(varargin));
% Add a wrapper function to check for Inf/NaN/complex values
if funValCheck
    % Add a wrapper function, CHECKFUN, to check for NaN/complex values without
    % having to change the calls that look like this:
    % f = funfcn(x,varargin{:});
    % x is the first argument to CHECKFUN, then the user's function,
    % then the elements of varargin. To accomplish this we need to add the
    % user's function to the beginning of varargin, and change funfcn to be
    % CHECKFUN.
    varargin = {funfcn, varargin{:}};
    funfcn = @checkfun;
end

n = numel(x);

% Initialize parameters
rho = 1; chi = 2; psi = 0.5; sigma = 0.5;
onesn = ones(1,n);
two2np1 = 2:n+1;
one2n = 1:n;

% Set up a simplex near the initial guess.
xin = x(:); % Force xin to be a column vector
v = zeros(n,n+1); fv = zeros(1,n+1);
v(:,1) = xin; % Place input guess in the simplex! (credit L.Pfeffer at Stanford)
x(:) = xin; % Change x to the form expected by funfcn
fv(:,1) = funfcn(x,varargin{:});
func_evals = 1;
itercount = 0;
how = ";
% Initial simplex setup continues later

```

```

% Initialize the output and plot functions.
if haveoutputfcn || haveplotfcn
    [xOutputfcn, optimValues, stop] = callOutputAndPlotFcns(outputfcn,plotfcns,v(:,1),xOutputfcn,'init',itercount, ...
        func_evals, how, fv(:,1),varargin{:});
    if stop
        [x,fval,exitflag,output] = cleanUpInterrupt(xOutputfcn,optimValues);
        if prnt > 0
            disp(output.message)
        end
        return;
    end
end

% Print out initial f(x) as 0th iteration
if prnt == 3
    disp(' ')
    disp(header)
    disp(sprintf(' %5.0f      %5.0f      %12.6g      %s', itercount, func_evals, fv(1), how));
elseif prnt == 4
    clc
    formatsave = get(0,{'format','formatspacing'});
    format compact
    format short e
    disp(' ')
    disp(how)
    v
    fv
    func_evals
end
% OutputFcn and PlotFcns call
if haveoutputfcn || haveplotfcn
    [xOutputfcn, optimValues, stop] = callOutputAndPlotFcns(outputfcn,plotfcns,v(:,1),xOutputfcn,'iter',itercount, ...
        func_evals, how, fv(:,1),varargin{:});
    if stop % Stop per user request.
        [x,fval,exitflag,output] = cleanUpInterrupt(xOutputfcn,optimValues);
        if prnt > 0
            disp(output.message)
        end
        return;
    end
end

% Continue setting up the initial simplex.
% Following improvement suggested by L.Pfeffer at Stanford
usual_delta = 0.05; % 5 percent deltas for non-zero terms
zero_term_delta = 0.00025; % Even smaller delta for zero elements of x
for j = 1:n
    y = xin;
    if y(j) ~= 0
        y(j) = (1 + usual_delta)*y(j);
    else
        y(j) = zero_term_delta;
    end
    v(:,j+1) = y;
end

```

```

    x(:) = y; f = funfcn(x,varargin{:});
    fv(1,j+1) = f;
end

% sort so v(1,:) has the lowest function value
[fv,j] = sort(fv);
v = v(:,j);

how = 'initial simplex';
itercount = itercount + 1;
func_evals = n+1;
if prnt == 3
    disp(sprintf(' %5.0f      %5.0f      %12.6g      %s', itercount, func_evals, fv(1), how))
elseif prnt == 4
    disp(' ')
    disp(how)
    v
    fv
    func_evals
end
% OutputFcn and PlotFcns call
if haveoutputfcn || haveplotfcn
    [xOutputfcn, optimValues, stop] = callOutputAndPlotFcns(outputfcn,plotfcns,v(:,1),xOutputfcn,'iter',itercount, ...
        func_evals, how, fv(:,1),varargin{:});
    if stop % Stop per user request.
        [x,fval,exitflag,output] = cleanUpInterrupt(xOutputfcn,optimValues);
        if prnt > 0
            disp(output.message)
        end
        return;
    end
end
exitflag = 1;

% Main algorithm: iterate until
% (a) the maximum coordinate difference between the current best point and the
% other points in the simplex is less than or equal to TolX. Specifically,
% until  $\max(\|v_2-v_1\|, \|v_2-v_1\|, \dots, \|v_{n+1}-v_1\|) \leq \text{TolX}$ ,
% where  $\|\cdot\|$  is the infinity-norm, and v1 holds the
% vertex with the current lowest value; AND
% (b) the corresponding difference in function values is less than or equal
% to TolFun. (Cannot use OR instead of AND.)
% The iteration stops if the maximum number of iterations or function evaluations
% are exceeded
while func_evals < maxfun && itercount < maxiter
    if max(abs(fv(1)-fv(two2np1))) <= max(tolf,10*eps(fv(1))) && ...
        max(max(abs(v(:,two2np1)-v(:,onesn)))) <= max(tolx,10*eps(max(v(:,1))))
        break
    end

    % Compute the reflection point

    % xbar = average of the n (NOT n+1) best points
    xbar = sum(v(:,one2n), 2)/n;
    xr = (1 + rho)*xbar - rho*v(:,end);

```

```

x(:) = xr; fxr = funfcn(x,varargin{:});
func_evals = func_evals+1;

if fxr < fv(:,1)
    % Calculate the expansion point
    xe = (1 + rho*chi)*xbar - rho*chi*v(:,end);
    x(:) = xe; fxe = funfcn(x,varargin{:});
    func_evals = func_evals+1;
    if fxe < fxr
        v(:,end) = xe;
        fv(:,end) = fxe;
        how = 'expand';
    else
        v(:,end) = xr;
        fv(:,end) = fxr;
        how = 'reflect';
    end
else % fv(:,1) <= fxr
    if fxr < fv(:,n)
        v(:,end) = xr;
        fv(:,end) = fxr;
        how = 'reflect';
    else % fxr >= fv(:,n)
        % Perform contraction
        if fxr < fv(:,end)
            % Perform an outside contraction
            xc = (1 + psi*rho)*xbar - psi*rho*v(:,end);
            x(:) = xc; fxc = funfcn(x,varargin{:});
            func_evals = func_evals+1;

            if fxc <= fxr
                v(:,end) = xc;
                fv(:,end) = fxc;
                how = 'contract outside';
            else
                % perform a shrink
                how = 'shrink';
            end
        else
            % Perform an inside contraction
            xcc = (1-psi)*xbar + psi*v(:,end);
            x(:) = xcc; fxcc = funfcn(x,varargin{:});
            func_evals = func_evals+1;

            if fxcc < fv(:,end)
                v(:,end) = xcc;
                fv(:,end) = fxcc;
                how = 'contract inside';
            else
                % perform a shrink
                how = 'shrink';
            end
        end
    end
end
if strcmp(how,'shrink')
    for j=two2np1

```

```

        v(:,j)=v(:,1)+sigma*(v(:,j) - v(:,1));
        x(:) = v(:,j); fv(:,j) = funfcn(x,varargin{:});
    end
    func_evals = func_evals + n;
end
end
end
[fv,j] = sort(fv);
v = v(:,j);
itercount = itercount + 1;
if prnt == 3
    disp(sprintf(' %5.0f    %5.0f    %12.6g    %s', itercount, func_evals, fv(1), how))
elseif prnt == 4
    disp(' ')
    disp(how)
    v
    fv
    func_evals
end
% OutputFcn and PlotFcns call
if haveoutputfcn || haveplotfcn
    [xOutputfcn, optimValues, stop] = callOutputAndPlotFcns(outputfcn,plotfcns,v(:,1),xOutputfcn,'iter',itercount,
...
    func_evals, how, fv(:,1),varargin{:});
    if stop % Stop per user request.
        [x,fval,exitflag,output] = cleanUpInterrupt(xOutputfcn,optimValues);
        if prnt > 0
            disp(output.message)
        end
        return;
    end
end
end % while

x(:) = v(:,1);
fval = fv(:,1);

if prnt == 4,
    % reset format
    set(0,{'format','formatspacing'},formatsave);
end
output.iterations = itercount;
output.funcCount = func_evals;
output.algorithm = 'Nelder-Mead simplex direct search';

% OutputFcn and PlotFcns call
if haveoutputfcn || haveplotfcn
    callOutputAndPlotFcns(outputfcn,plotfcns,x,xOutputfcn,'done',itercount, func_evals, how, fval, varargin{:});
end

if func_evals >= maxfun
    msg = sprintf(['Exiting: Maximum number of function evaluations has been exceeded\n' ...
        ' - increase MaxFunEvals option.\n' ...
        ' Current function value: %f \n'], fval);
    if prnt > 0

```

```

        disp(' ')
        disp(msg)
    end
    exitflag = 0;
elseif itercount >= maxiter
    msg = sprintf(['Exiting: Maximum number of iterations has been exceeded\n' ...
        ' - increase MaxIter option.\n' ...
        ' Current function value: %f \n'], fval);
    if prnt > 0
        disp(' ')
        disp(msg)
    end
    exitflag = 0;
else
    msg = ...
        sprintf(['Optimization terminated:\n', ...
            ' the current x satisfies the termination criteria using OPTIONS.TolX of %e \n' ...
            ' and F(X) satisfies the convergence criteria using OPTIONS.TolFun of %e \n'], ...
            tolX, tolF);
    if prnt > 1
        disp(' ')
        disp(msg)
    end
    exitflag = 1;
end

output.message = msg;

%-----
function [xOutputfcn, optimValues, stop] = callOutputAndPlotFcns(outputfcn,plotfcns,x,xOutputfcn,state,iter,...
    numf,how,f,varargin)
% CALLOUTPUTANDPLOTFCNS assigns values to the struct OptimValues and then calls the
% outputfcn/plotfcns.
%
% state - can have the values 'init','iter', or 'done'.

% For the 'done' state we do not check the value of 'stop' because the
% optimization is already done.
optimValues.iteration = iter;
optimValues.funccount = numf;
optimValues.fval = f;
optimValues.procedure = how;

xOutputfcn(:) = x; % Set x to have user expected size
stop = false;
% Call output functions
if ~isempty(outputfcn)
    switch state
        case {'iter','init'}
            stop = callAllOptimOutputFcns(outputfcn,xOutputfcn,optimValues,state,varargin{:}) || stop;
        case 'done'
            callAllOptimOutputFcns(outputfcn,xOutputfcn,optimValues,state,varargin{:});
        otherwise
            error('MATLAB:fminsearch:InvalidState', ...
                'Unknown state in CALLOUTPUTANDPLOTFCNS.')
    end
end

```

```

    end
end
% Call plot functions
if ~isempty(plotfns)
    switch state
        case {'iter','init'}
            stop = callAllOptimPlotFcns(plotfns,xOutputfcn,optimValues,state,varargin{:}) || stop;
        case 'done'
            callAllOptimPlotFcns(plotfns,xOutputfcn,optimValues,state,varargin{:});
        otherwise
            error('MATLAB:fminsearch:InvalidState', ...
                'Unknown state in CALLOUTPUTANDPLOTFCNS.')
    end
end

%-----
function [x,FVAL,EXITFLAG,OUTPUT] = cleanUpInterrupt(xOutputfcn,optimValues)
% CLEANUPINTERRUPT updates or sets all the output arguments of FMINBND when the optimization
% is interrupted.

x = xOutputfcn;
FVAL = optimValues.fval;
EXITFLAG = -1;
OUTPUT.iterations = optimValues.iteration;
OUTPUT.funcCount = optimValues.funccount;
OUTPUT.algorithm = 'golden section search, parabolic interpolation';
OUTPUT.message = 'Optimization terminated prematurely by user.';

%-----
function f = checkfun(x,userfcn,varargin)
% CHECKFUN checks for complex or NaN results from userfcn.

f = userfcn(x,varargin{:});
% Note: we do not check for Inf as FMINSEARCH handles it naturally.
if isnan(f)
    error('MATLAB:fminsearch:checkfun:NaNFval', ...
        'User function "%s" returned NaN when evaluated;\n FMINSEARCH cannot continue.', ...
        localChar(userfcn));
elseif ~isreal(f)
    error('MATLAB:fminsearch:checkfun:ComplexFval', ...
        'User function "%s" returned a complex value when evaluated;\n FMINSEARCH cannot continue.', ...
        localChar(userfcn));
end

%-----
function strfcn = localChar(fcn)
% Convert the fcn to a string for printing

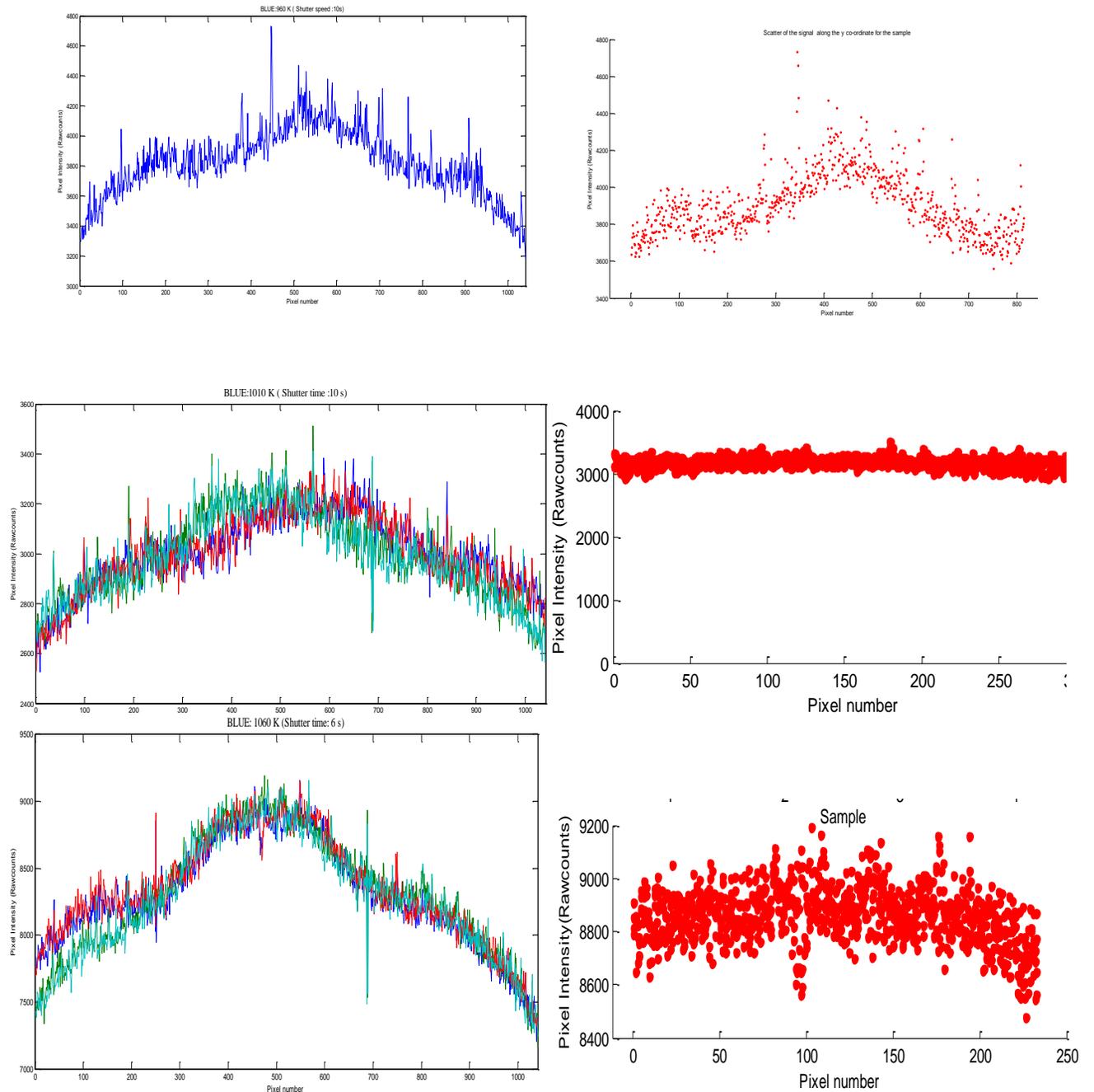
if ischar(fcn)
    strfcn = fcn;
elseif isa(fcn,'inline')
    strfcn = char(fcn);
elseif isa(fcn,'function_handle')
    strfcn = func2str(fcn);
else

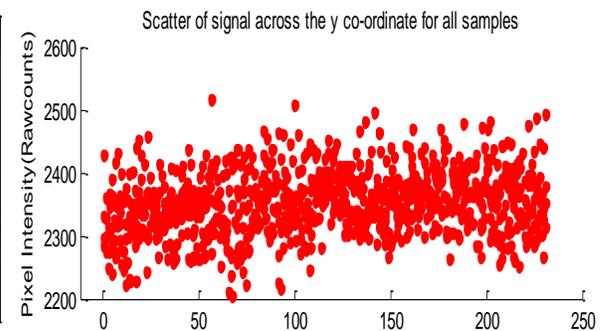
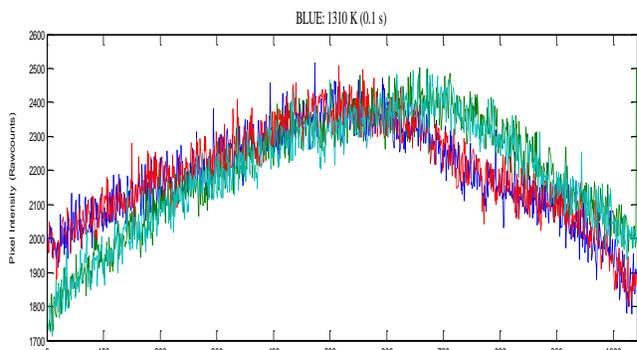
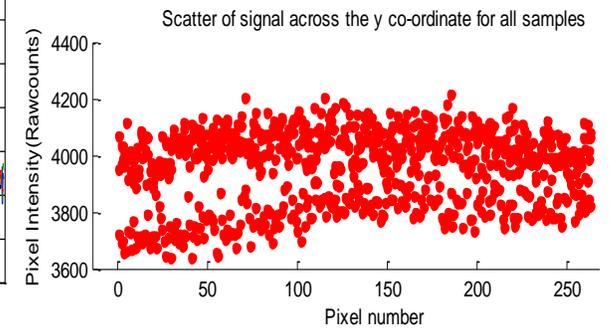
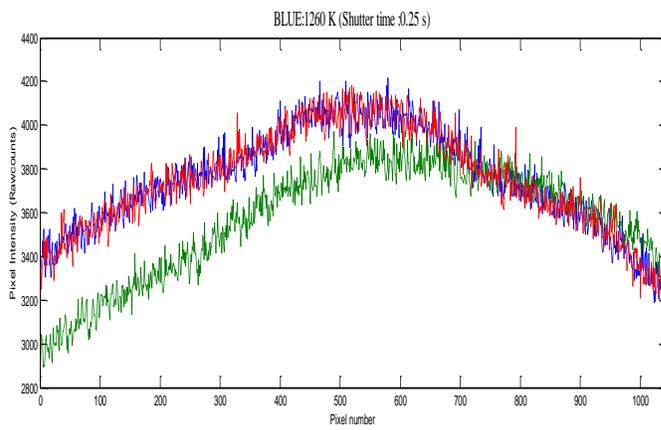
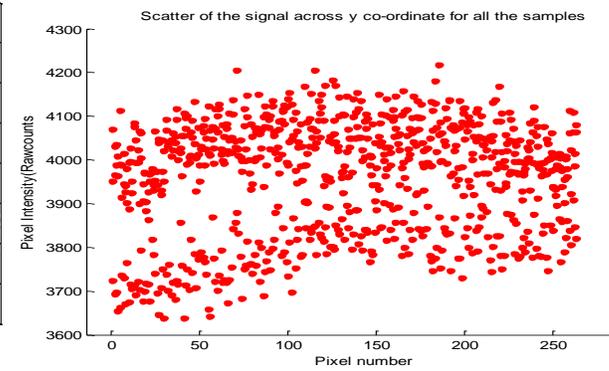
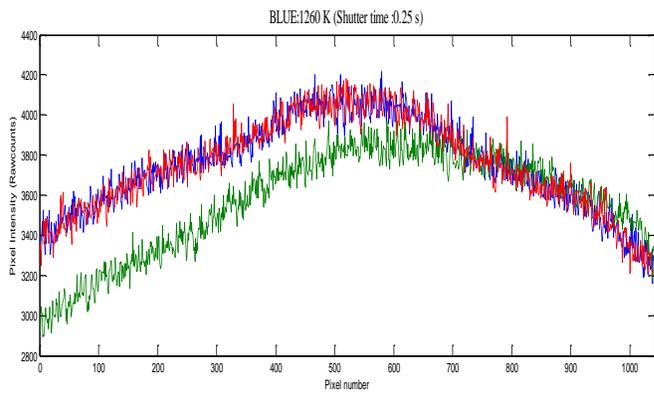
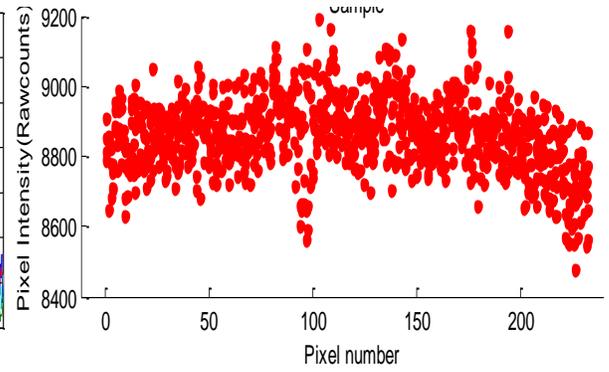
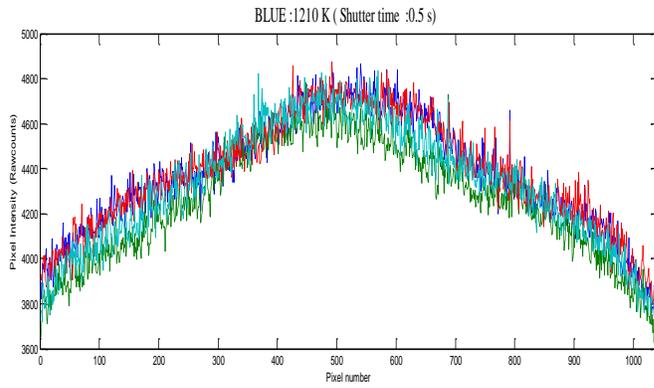
```

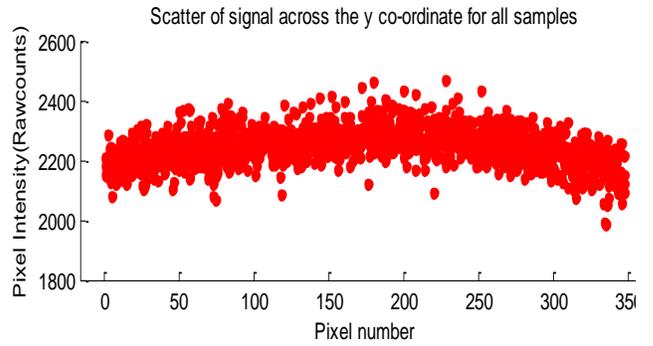
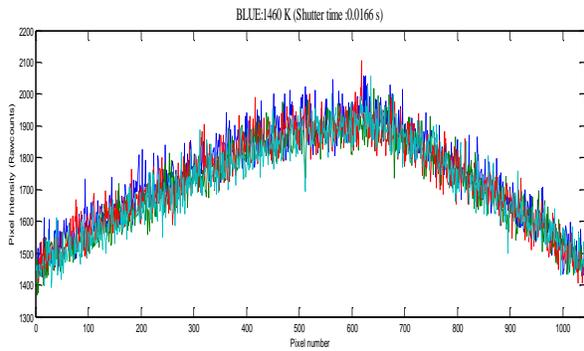
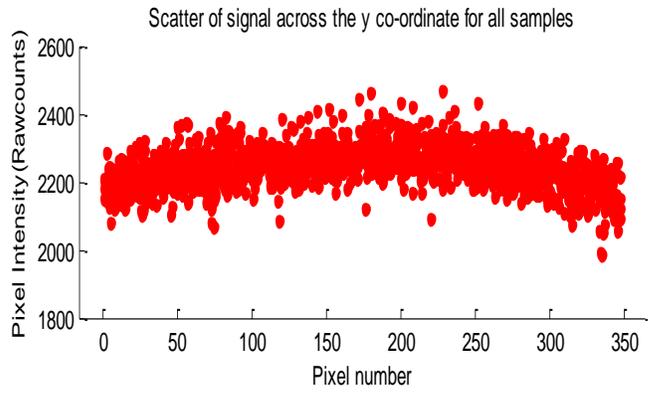
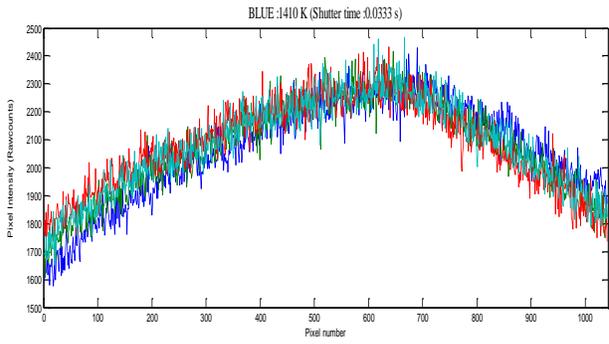
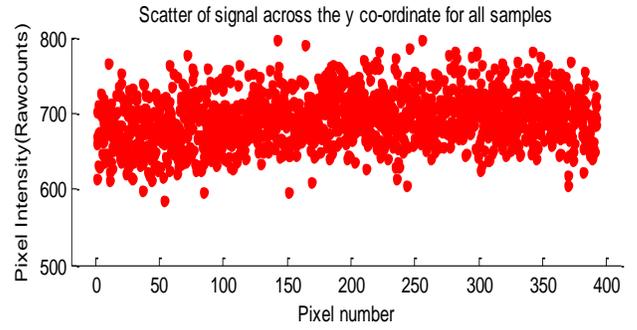
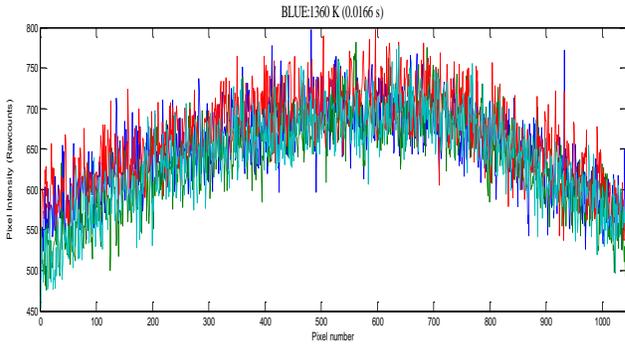
```
try
    strfcn = char(fcn);
catch
    strfcn = '(name not printable)';
end
end
```

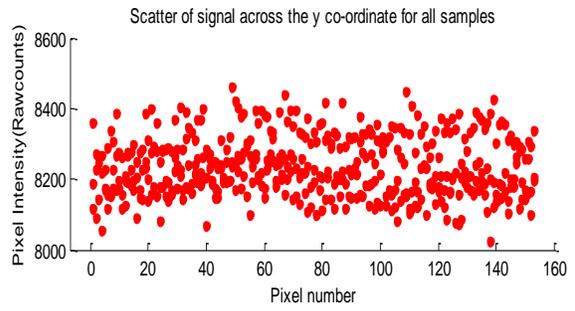
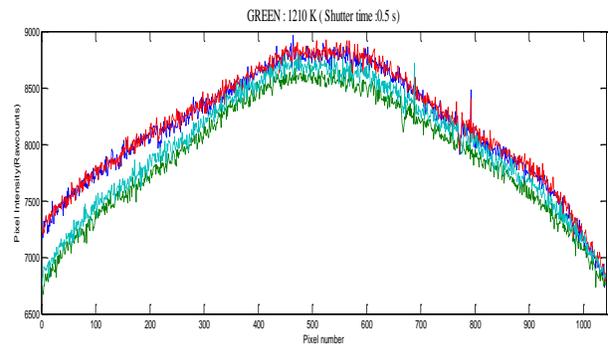
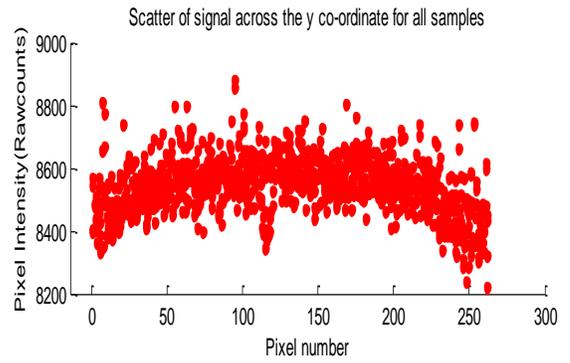
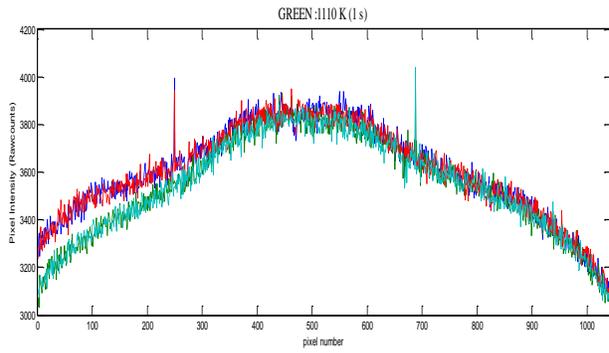
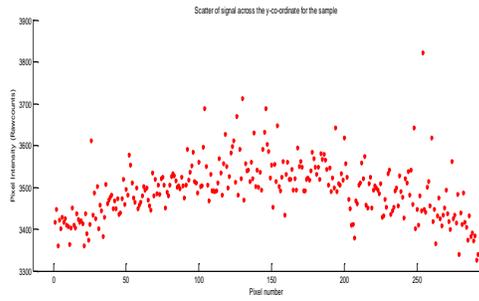
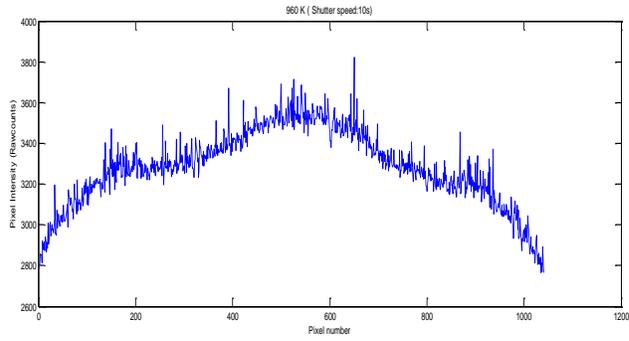
# APPENDIX D :STATISTICAL ANALYSIS FOR NOISE REDUCTION

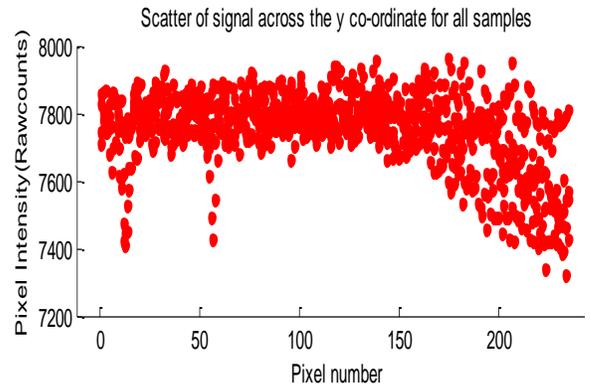
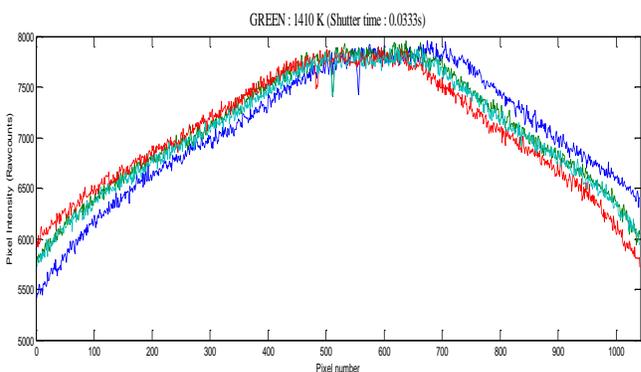
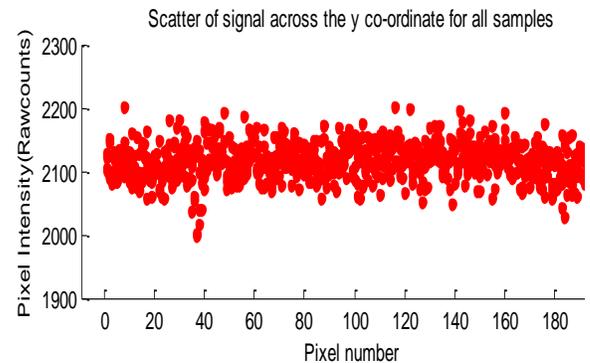
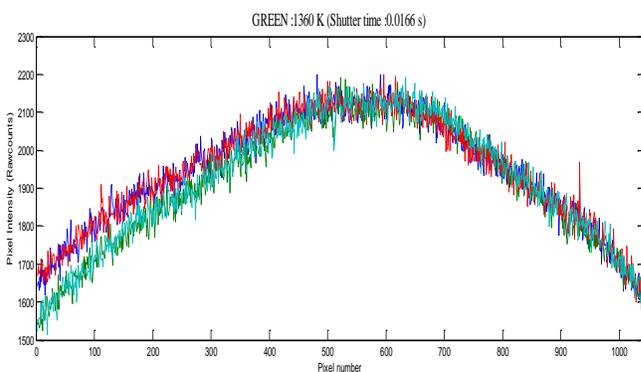
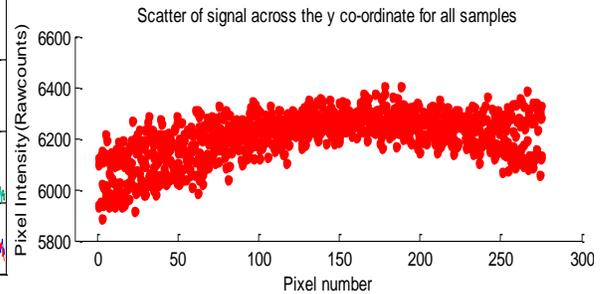
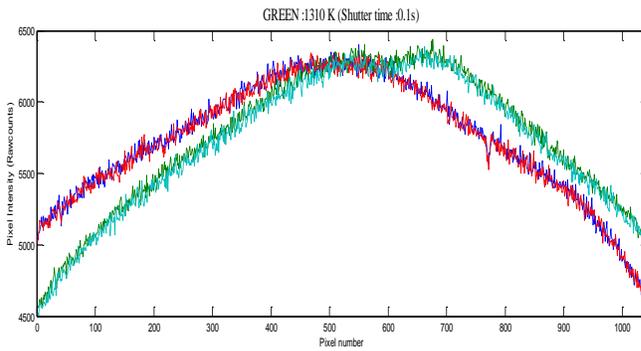
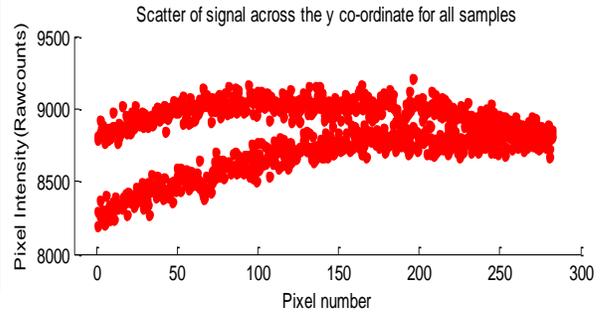
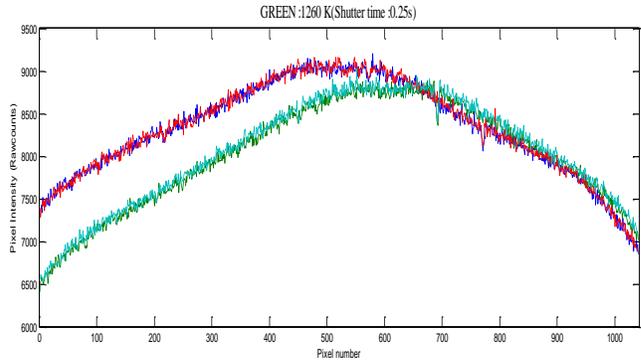
**Figures showing pixel intensities across a line profile and noise scatter in the multiple samples for every temperature and particular shutter time selected for all the 11 temperatures**

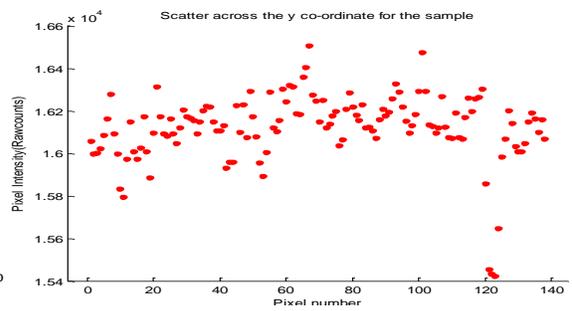
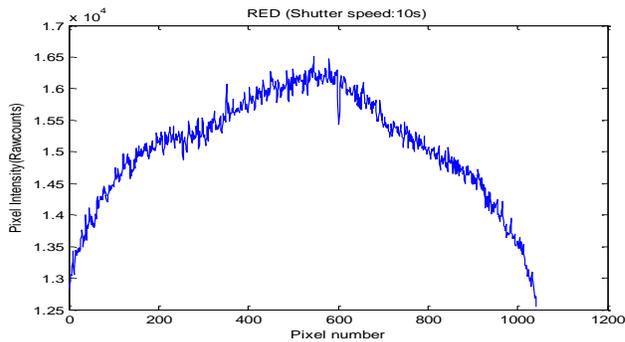
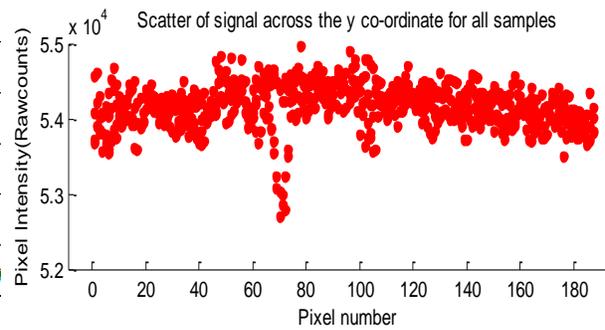
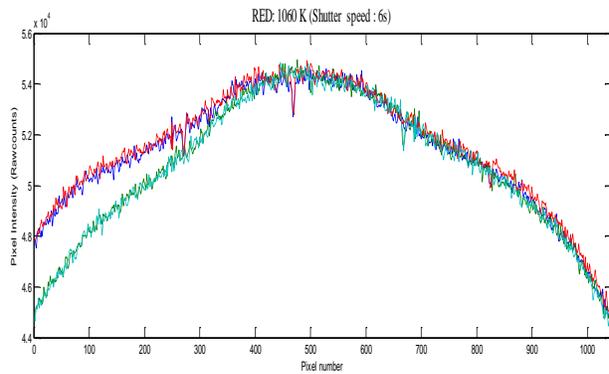
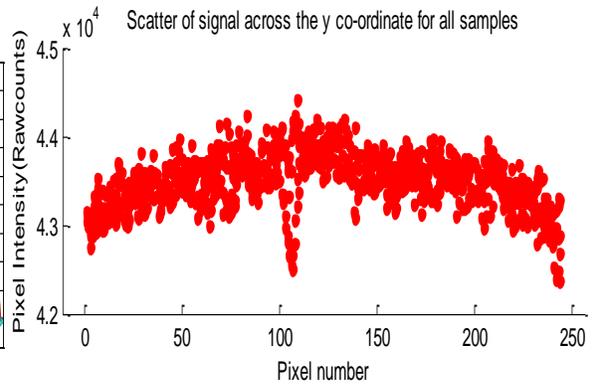
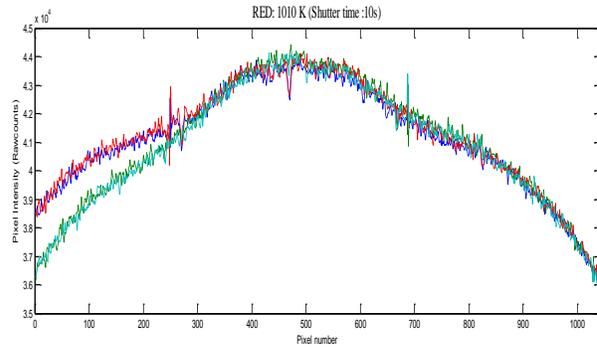
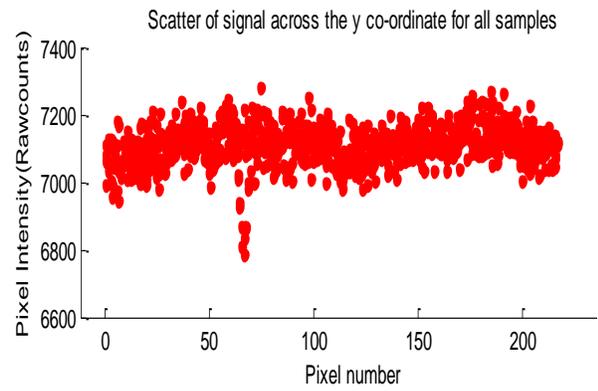
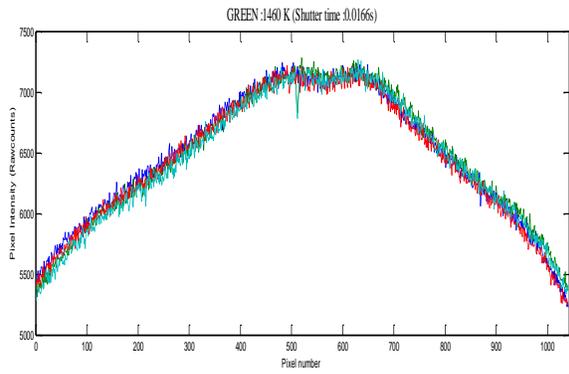


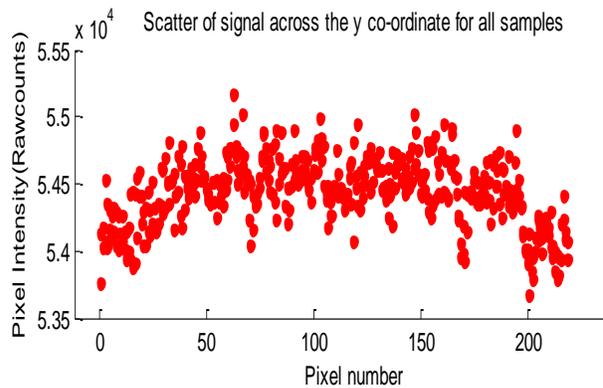
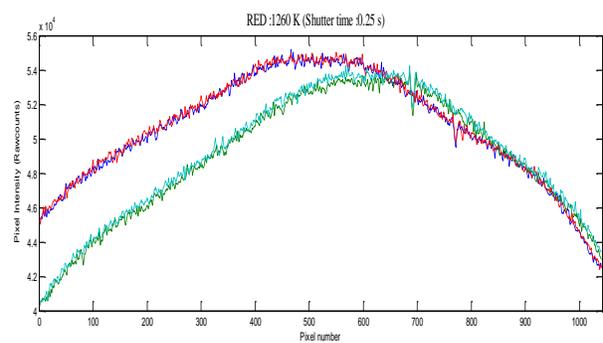
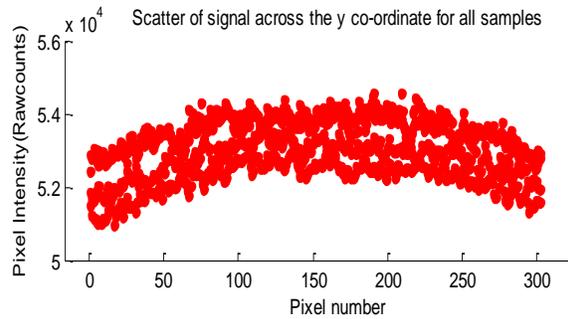
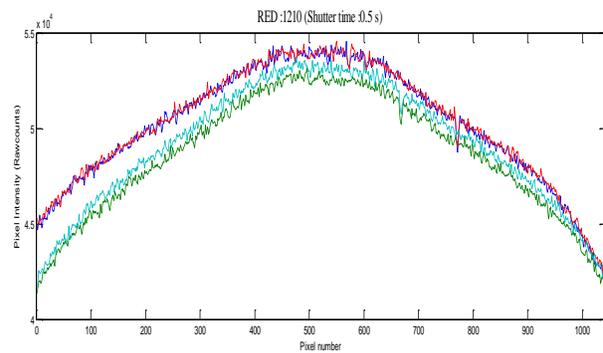
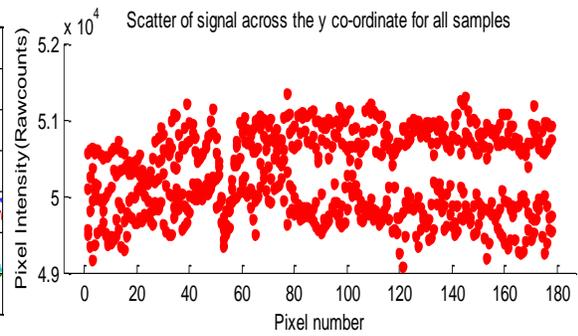
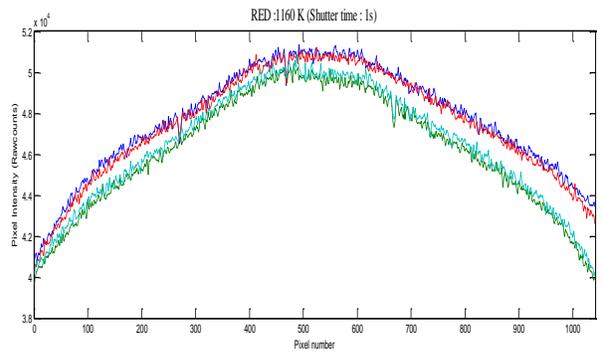
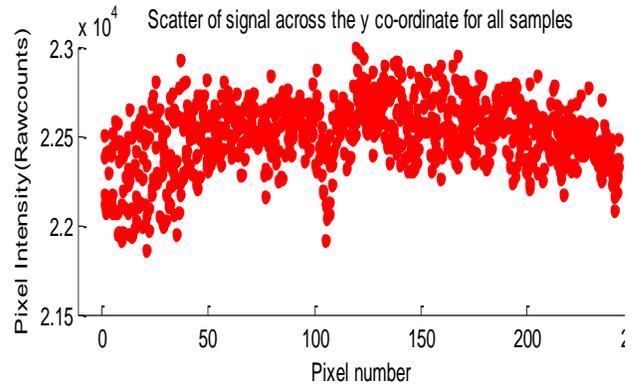
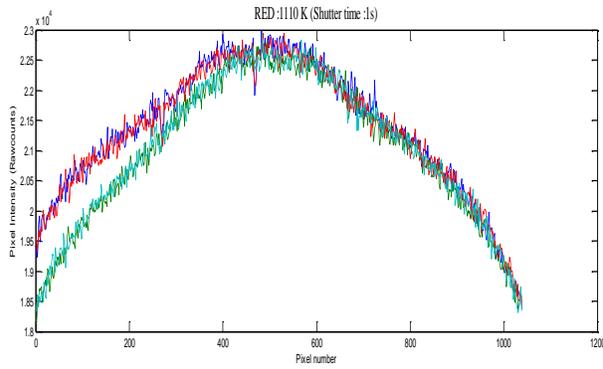


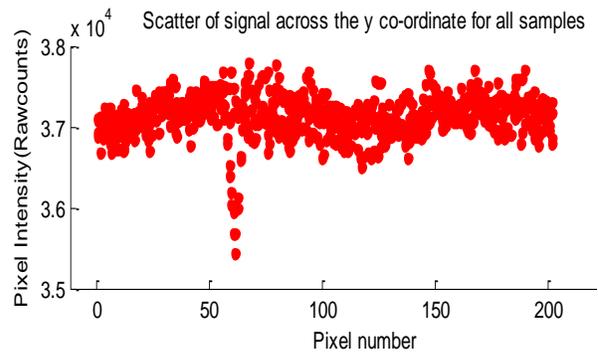
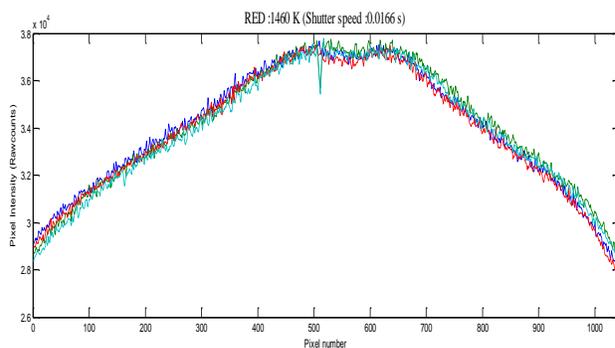
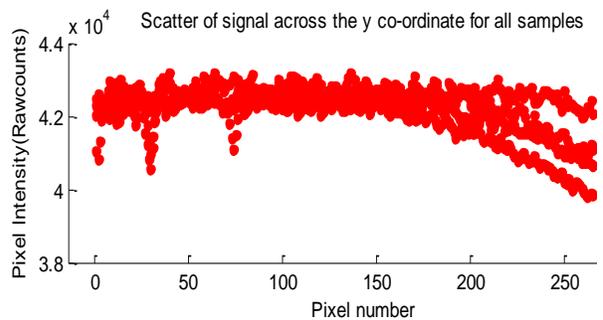
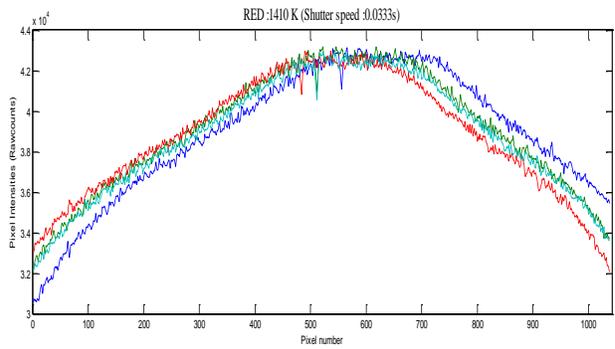
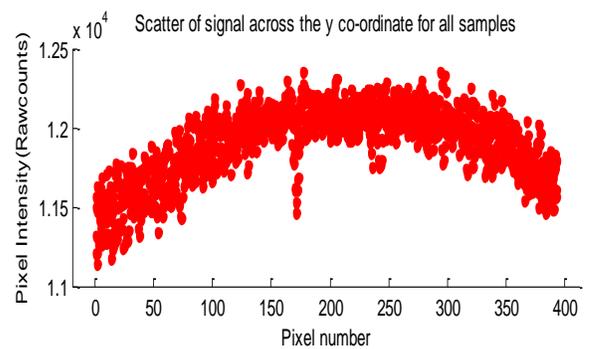
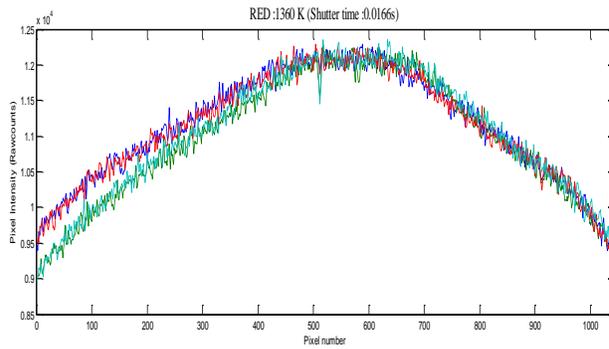
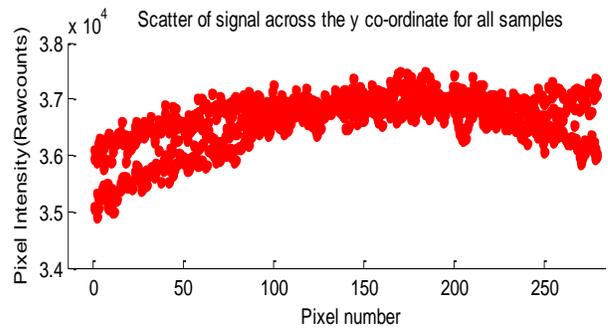
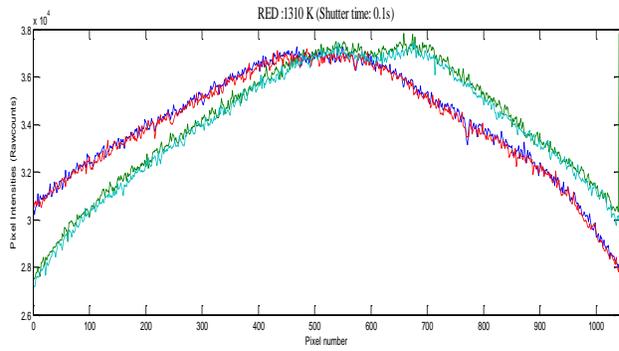












## Bibliography

- A. Albert. (1972). Regression and the Moore-Penrose pseudoinverse. *Academic Press* .
- Aslam, A., & Finlayson, G. (2002). Recovering spectral sensitivities with uncertainty. *The First European Conference on Color in Graphics, Imaging and Vision*, (pp. 22-26).
- Aslam, A., & Hardeberg, J. (2006). Smoothing Jagged Spectra for Accurate Spectral Sensitivities Recovery. *Computational Imaging and Vision* , 32, 259-266.
- Barnard, K., & Funt, B. (2002). Camera characterization for color research. *Color Research and Application* , 27 (3), 153-164.
- C. Reinsch, G. a. (1971). Singular value decomposition and least squares solutions in Linear Algebra. In J. a. C. Reinsch, *Handbook for Automatic Computation* (pp. 134-151). Springer-Verlag.
- Dyas, B. (2000). Robust sensor response characterization. *IS&T/SID Eight Color Imaging Conference* , (pp. 144-148). Arizona, USA.
- Farrell, J., & Wandell, B. A. (1993). Rethinking the white point. *Proceedings of IS&T and SID's Color Imaging Conference: Transforms and Transportability of Color* , (pp. 65-67). Scottsdale, Arizona.
- Finlayson, G., & Funt, B. (1995). Color constancy under varying illumination. *Computer Vision* .
- Finlayson, G., Huble, P., & Hordley, S. (1998). Recovering device sensitivities with quadratic programming. *IS&T/SID Sixth Color Imaging Conference : Color Science, Systems and Applications*, (pp. 90-95).
- Foster, D. S. (1997). Four issues concerning colour constancy and relational color constancy. *Vision research* .
- H. J. Trussell, G. (1996, October). Set theoretic estimation in color scanner characterization. *Journal of Electronic Imaging* , 479-489.
- Hainaut, O. (2006, December). *Basic CCD image processing*. Retrieved January 15, 2011, from <http://www.eso.org/~ohainaut/ccd/>
- Hansen, P. (1998). Rank-Deficient and Discrete III-Posed Problems: Numerical Aspects of Linear Inversion. *Siam Monographs on Mathematical Modeling Computation* .
- Hubel, P. M., Sherman, D., & Farrell, J. (1994). A Comparison of Methods of Sensor Spectral Sensitivity Estimation. *IS&T and SID's 2nd Color Imaging Conference: Color Science, Systems and Applications* , 45.
- I. T. Jolliffe. (1986). *Principal Component Analysis*. New York: Springer-Verlag.
- Jon Yngve Hardeberg, H. B. (1998). *Spectral characterisation of electronic cameras*. Paris, Francis: Ecole Nationale Supérieure des Telecommunications.

- Knagpe, D. (2009, October 2). *Model Predictive Control*. Retrieved July 26, 2011, from SPM Signal Processing Microelectronics: <http://sigpromu.org/mpc/qp.html>
- Korsgaard, I. R., & Andersen, A. H. (1998). The Additive Genetic Gamma Frailty Model. *Scandinavian Journal of Statistics* , 25, 255-269.
- Lauziere, Y., Gingras, D., & F., F. (1999). Color Camera Characterization with an Application to Detection under Daylight. *The EUROOTP Conference on Polarization and Color Techniques in Industrial Inspection*, 3826, pp. 86-100.
- Lauziere, Y., Gingras, D., & Ferrie, P. (1999). Color Camera Characterization with an Application to Detection under Daylight. *Vision Interface '99*. Trois Rivieres.
- Liebelt, P. (1967). An Introduction to Optimal Estimation. *Wesley Reading* .
- Maloney, L., & Wandell, B. (1985). Color constancy: a method for recovering surface spectral reflectance. *Journal of the Optical Society of America* , 29-33.
- Mancill, C. (1975). *Digital Color Image Restoration*. Los Angeles: Ph.D. Thesis, University of Southern California.
- Mancill, C., & Pratt, W. (1976). Spectral estimation techniques for the spectral calibration of a color iamge scanner. *Applied Optics* , 15(1), 73-75.
- McCamy, C., Marcus, H., & Davidson, J. (1976). A Color Rendition Chart. *Journal of Applied Photographic Engineering* , 2 (3).
- Nikon:Technology:Image Processing*. (2003, July). Retrieved July 23, 2011, from [http://www.nikon.com/about/technology/core/image/image\\_processing\\_e/index.htm](http://www.nikon.com/about/technology/core/image/image_processing_e/index.htm)
- P.D.Burns. (1997). *Analysis of multispectral acquisition systems*. Phd Thesis,Centre for Imaging Science,Rochester Institute of Technology.
- Peterson, J., & Heukelman, C. (2010, June). *Resolution in Color Filter Array images*. Retrieved July 25, 2011, from Resolution in Color Filter Array Images - Jon Peterson and Cobus Heukelman: <http://scien.stanford.edu/pages/labsite/2010/psych221/projects/2010/PetersonHeukelman/Website/index.html>
- Pike, T. (2011). Using digital cameras to investigate animal. *Behav Ecol Socioboiol* , 849-858.
- Sharma, G., & Trussell, H. (1997). Figures of merit for color scanners. *IEEE Transaction Image Processing* , 6, 990-1001.
- Sharma, G., & Trussell, H. (1994). Signal Processing Methods in Color Calibration. *Proceedings of SPIE:Device-Independent Color Imaging*, 2170, pp. 18-23.
- Trussell, H., & Sharma, G. (1993). Characterization of Scanner Sensitivity. *IS&T and SID's Color Imaging Conference:Transforms and Transportability of Color*, (pp. 103-107).

Vora, P. L., Farrell, J., Tietz, J. D., & Brainard, D. H. (1997). *Digital color cameras-2-Spectral response*. Hewlett Packard Company Technical Report.

Vora, P., Farrell, J., Tietz, J. D., & Brainard, D. (1997). *Digital color cameras-1-Response models*. Hewlett Packard Company Technical Report.

Vora, P., Farrell, J., Tietz, J., & Brainard, D. (1997). Linear Models For Digital Cameras. *Proceedings of the IS&T's 50th Annual Conference*: , 377-382.

Vora, P., Farrell, J., Tietz, J., & Brainard, D. (1997). Linear Models For Digital Cameras. *Proceedings of the IS&T's 50th Annual Conference*: , 377-382.

W.K.Pratt. (1978). *Digital Image Processing*. John Wiley and Sons .

Wandell, B., & Farrell, J. (1993). Scanner Linearity. *Journal of Electronic Imaging* , 225-230.

*Working of Digital Cameras - Basic circuit - CircuitsFinder-Free Electronic Circuit Diagram Design*.

(n.d.). Retrieved July 25, 2011, from

[http://www.circuitsfinder.com/basic/Working\\_of\\_Digital\\_Cameras\\_666.html](http://www.circuitsfinder.com/basic/Working_of_Digital_Cameras_666.html)

